

**Universidad de las Ciencias Informáticas
Facultad 4**



Título: Análisis y Diseño del Módulo de Cobros y Pagos del sistema integral de gestión CEDRUX.

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

Autor(es): Leisniel Ignacio Cabrera Pereira.
Mailyn Hernández Gómez.

Tutor(es): Ing. Silvia María Llarch Leyva.
Ing. Anisleydi Céspedes Vega.

Ciudad de la Habana, Mayo del 2009
Año 50 de la Revolución

“La vida es aquello que te va sucediendo mientras tú te empeñas en hacer otros planes”

John Lennon

DATOS DE CONTACTO

Tutor(a): Ing. Anisleydi Céspedes Vega.

Correo electrónico: acespedes@uci.cu

Título de Graduado: Ingeniero en Ciencias Informáticas.

Categoría docente: Instructor Recién Graduado.

Tutor(a): Ing. Silvia María Llarch Leyva.

Correo electrónico: smallarch@uci.cu

Título de Graduado: Ingeniero en Ciencias Informáticas.

Categoría docente: Instructor Recién Graduado.

AGRADECIMIENTOS DE MAILYN



.....A mi mamá por ser mi guía, mi fuerza y mi apoyo durante estos cinco años.

.....A mi papá por apoyarme siempre en mis estudios y confiar en mí.

.....A mi hermana Maydelín que siempre me ha apoyado tanto, a ti también te debo este sueño.

.....A mi hermano Erito, que es mi vida, en los momentos difíciles ha sido mi luz, mi fuerza...

.....A mi novio Noel, que me ha apoyado tanto, a ti te debo parte de este sueño, gracias por darme la fuerza que siempre necesite en los momentos difíciles.

.....A mi abuela mima que siempre me ha apoyado tanto, con ese amor tan grande que profesa a todos sus nietos.

.....A mi abuelo Cardoso y a mi abuela Marina.

.....A mi tía Diana que aunque está lejos siempre me ha apoyado, y siempre ha confiado en mi capacidad para salir adelante y hacerlo bien.

.....A mi primo Cristian que es como si fuera otro hermano más, que lo quiero cantidad y siempre lo tengo presente.

.....A mi padrastro Erick, muchas gracias por siempre apoyarme y ayudarme.

.....A la Virtu, Saily, Leisis, Pedro, Annilie, Ismel, Alexey, Chirino, a la putty, en fin a todos mis compañeros de aula, que más que compañeros son amigos....

.....A mis amigos de siempre a Ania, Yudy, Lourdes, Jany, Evelyn, Kenia, Reidel.

.....A mis tutoras Anisleydi y Silvia, muchas gracias por todo.....

.....A la compañera María Elena por toda la ayuda que nos brindó.....

.....A la Revolución y a mi comandante Fidel por darme la oportunidad de estudiar en tan excelente escuela.

AGRADECIMIENTOS DE LEISNIEL



.....A mis padres y mi hermana por estar siempre para darme su apoyo y guiar mis pasos desde la distancia.

.....A mis abuelos maternos que son mis otros padres.

.....A mis abuelos que ya no están, pero si están.

.....A mi primo tío Ignacito, que ha sido de gran inspiración.

.....A mi familia por estar pendiente de mis estudios en todo momento.

.....A mis amigos de siempre.

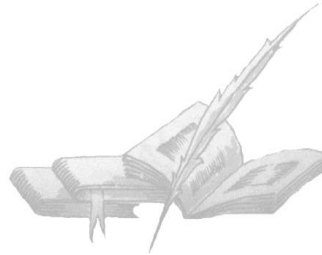
.....A las tutoras Anisleydi y Silvia, por su paciencia y dedicación.

.....A todos los amigos y compañeros con los que he compartido estos 5 años.

.....A la Revolución y a nuestro Comandante en jefe por brindarnos la posibilidad de cursar estudios y hacernos mejores personas.

DEDICATORIA DE MAILYN

Dedico este trabajo de diploma a mi mamá Mayda, a mi papá, a mis hermanos, a Noel y a mi abuela mima.....



DEDICATORIA DE LEISNIEL

Dedico este trabajo a mi papá, mi mamá, mi hermana y mis abuelos, por darme el motivo para seguir.

RESUMEN

Los procesos contables de Cobros y Pagos desempeñan un papel importante en las entidades cubanas, ya que permiten llevar a cabo un control eficiente de los derechos u obligaciones que tienen estas con sus clientes, proveedores y con el Estado.

En Cuba actualmente no existe un sistema contable que realice de forma genérica los procesos de cobros y pagos, ni que se ajuste a las particularidades de cada una de las entidades cubanas.

El presente trabajo de diploma contribuirá a la solución del problema antes planteado a través de la realización del análisis y diseño del módulo Cobros y Pagos del proyecto ERP Cubano, obteniéndose artefactos entendibles como: la especificación de los requisitos funcionales, los prototipos de interfaces de usuarios y los diagramas de clases del diseño, los cuales viabilizan a los implementadores la automatización de los procesos del módulo para que estos sean genéricos para todas las entidades cubanas.

Los artefactos generados se evaluaron a través de técnicas y métricas demostrando la calidad de los resultados obtenidos.

PALABRAS CLAVE: Cobros y Pagos, entidades, procesos, CEDRUX.

INDICE DE CONTENIDOS

AGRADECIMIENTOS DE MAILYN.....	II
AGRADECIMIENTOS DE LEISNIEL	III
RESUMEN	V
INTRODUCCIÓN:.....	1
CAPÍTULO I: FUNDAMENTACIÓN TEÓRICA.....	4
1.1. Introducción.	4
1.2. Sistemas ERP.....	4
1.3. La Contabilidad Financiera. Procesos contables de cobros y pagos.	5
1.4. Sistemas de Contabilidad Financiera existentes en las entidades cubanas.	7
1.5. Importancia de la automatización del módulo Cobros y Pagos a través del CEDRUX.	12
1.6. Modelo de desarrollo orientado a componentes.....	13
1.7. Actividades a desarrollar por el rol Analista y Diseñador de lógica del negocio.....	15
1.8. Notación de modelado de procesos de negocio BPMN.	18
1.9. Lenguaje unificado de modelado (UML).....	20
1.10. Herramienta Case a utilizar. Visual Paradigm.	21
1.11. Conclusiones parciales.	23
CAPÍTULO II: MODELADO DEL NEGOCIO Y REQUERIMIENTOS	24
2.1. Introducción.	24
2.2. Modelado de negocio.....	24
2.2.2. Descripción de procesos.	26
2.2.3. Modelo conceptual.	41
2.3. Requerimientos.....	47
2.4. Prototipos de interfaz de usuario.....	56
2.5. Validación de los requerimientos funcionales.....	120
2.6. Conclusiones parciales.	121

CAPÍTULO III: DISEÑO	122
3.1. Introducción:	122
3.2. Diseño.	122
3.3. Mecanismos de diseño.	123
3.4. Diagrama de clases del diseño.	125
3.6. Patrones.	190
3.7. Métricas para evaluar el diseño propuesto.....	194
3.8. Conclusiones parciales.	198
CONCLUSIONES.....	199
RECOMENDACIONES.....	200
BIBLIOGRAFÍA.....	201
GLOSARIO.....	205
ANEXOS	207

INTRODUCCIÓN:

En el mundo de hoy las actividades empresariales se encuentran en constantes cambios debido a los adelantos existentes, los cuales incluyen una mayor cantidad de personal vinculado a las entidades, gran cantidad de información a procesar, así como la necesidad de integrar los procesos contables, por lo que se hace indispensable la utilización de las nuevas tecnologías de la Informática y las Comunicaciones ya que las entidades deben contar con mayor agilidad y flexibilidad para adaptarse a los nuevos entornos de forma rápida y espontánea. Mediante las tecnologías informáticas se pueden aglutinar representaciones técnicas de Contabilidad que permiten mostrar de manera detallada las diferentes actividades contables en las entidades.

Dentro de los procesos contables de las entidades uno de los que más necesita ser automatizado es el de cobros y pagos debido a que desempeña un papel importante a la hora de relacionar a la entidad con sus clientes y proveedores. Mediante la gestión de cobros y pagos se organiza el control de las obligaciones o derechos que tiene la entidad tanto con el Presupuesto del Estado de acuerdo a las regulaciones financieras vigentes por concepto de impuestos y contribuciones, como con los clientes y proveedores en su relación de compra venta de mercancías, productos o servicios. Muchas de las grandes entidades existentes en el mundo han optado por realizar la informatización de sus actividades financieras y contables mediante los sistemas ERP¹, ya que estos sistemas de gestión empresarial son diseñados para automatizar y modelar la mayoría de los procesos de las entidades logrando una reducción de los costos de gerencia, incremento en el retorno de inversión y obtención de información financiera con las especificaciones de calidad y eficiencia que demandan los usuarios. Sin embargo, estos sistemas no están al alcance de todos, debido a que han obtenido precios muy elevados.

En varias de las entidades cubanas como Comercio y Gastronomía, actualmente no existe un sistema contable que gestione y controle su Contabilidad Financiera. Muchos de los procesos que engloban la Contabilidad Financiera, como el de cobros y pagos mencionado anteriormente, se realizan de forma manual, archivándose en papel duro todo lo relacionado con estos procesos, lo cual constituye un riesgo para el correcto desempeño y control de la gestión contable de la empresa, pues la probabilidad de cometer errores cuando se realizan de forma manual es muy alta, afectando esto la toma de decisiones de la entidad.

¹ Planificación de recursos empresariales del inglés Enterprise Resource Planning.

Cuba cuenta con 26 soluciones empresariales certificadas como CENTAI PROGRESS, RODAS XXI y SISCONT5. Existen entidades como Turismo, CITMATEL y TECNOMATICA (MINBAS) que sí llevan de forma automatizada mediante estas soluciones los procesos contables de cobros y pagos. Estas soluciones no son genéricas y son utilizadas en entidades específicas de acuerdo a las operaciones que las mismas realizan. Muchos de estos sistemas contables no son cubanos, en su mayoría están diseñados bajo principios capitalistas que tienen un modelo de gestión y procesos diferente al de las entidades cubanas, cuya economía es centralizada y tiene su mayor crecimiento en el aspecto social, además cuenta con la existencia de dos monedas que coexisten y comparten legalmente las funciones del dinero en la economía nacional, lo cual no ocurre en otro país. Otros son propietarios, por lo que requieren de licencias para su utilización, sus precios para la instalación, mantenimiento y actualización son elevados, los derechos de autor de dichos software en ocasiones pertenecen a entidades norteamericanas por lo que para Cuba es imposible utilizarlos debido al bloqueo impuesto por el gobierno de los Estados Unidos de América.

De forma general estos software existentes no se adaptan a las nuevas legislaciones emitidas por el Ministerio de Finanzas y Precios.

La situación problemática planteada permitió definir el **Problema Científico**: ¿Cómo mejorar los procesos contables de cobros y pagos, para que se realicen de forma genérica en todas las entidades cubanas?

Se plantea como **Objeto de Estudio**: Procesos contables de la Contabilidad Financiera.

Para la solución de este problema se plantea como **Objetivo General**: Realizar el análisis y diseño del módulo Cobros y Pagos del Cedrux para garantizar su posterior implementación.

Y el **Campo de Acción** se enmarca en: Procesos contables de Cobros y Pagos.

Para desarrollar el presente trabajo se llevaron a cabo las siguientes tareas:

- Elaborar el marco teórico de la investigación.
- Estudiar, identificar y describir los procesos que se realizan en cobros y pagos.
- Desarrollar el análisis y diseño del módulo.
- Evaluar los resultados obtenidos.

El presente trabajo de diploma se encuentra estructurado de la siguiente forma: presenta una introducción, tres capítulos, un glosario con los términos más empleados durante la realización del trabajo, un conjunto de anexos y la bibliografía consultada para el completamiento del marco teórico.

Capítulo 1. Fundamentación teórica: comienza con un análisis de los diferentes sistemas de Contabilidad Financiera existentes en las entidades cubanas. Se explica la importancia que tiene la automatización de los procesos contables de cobros y pagos en las entidades cubanas. Este capítulo finaliza con una justificación de la metodología de desarrollo de software, herramientas y técnicas a utilizar.

Capítulo 2. Modelado del negocio y requerimientos: en este capítulo se realiza un breve estudio del estado actual del negocio a considerar para automatizar el módulo Cobros y Pagos. Se presenta el modelo y la descripción de cada uno de los procesos identificados, así como el modelo conceptual correspondiente. Además, se especifican cada uno de los requerimientos funcionales definidos y se presentan los prototipos de interfaz propuestos para cada uno de ellos.

Capítulo 3. Diseño: en el presente capítulo se realiza el diseño del módulo Cobros y Pagos, presentando las clases del diseño realizadas utilizando los estereotipos web, así como una descripción de las mismas. Además serán estudiados los patrones de diseño empleados, así como los mecanismos de diseño elaborados por los diseñadores del módulo. Se evalúa a través de métricas la propuesta de diseño realizada para el módulo.

CAPÍTULO I: FUNDAMENTACIÓN TEÓRICA.

1.1. Introducción.

El presente capítulo comienza con un análisis de los diferentes sistemas de Contabilidad Financiera existentes en las entidades cubanas. Se explica la importancia que tiene la automatización de los procesos contables de cobros y pagos en dichas entidades. Esta sección finaliza con una justificación de la metodología de desarrollo de software, herramientas y técnicas a utilizar.

1.2. Sistemas ERP.

Los ERP son sistemas de información integral que incorporan los procesos operativos y de negocios, tienen como propósito fundamental otorgar apoyo a los clientes del negocio, tiempos rápidos de respuestas a sus problemas, así como un eficiente manejo de la información, lo cual permite la toma oportuna de decisiones y la disminución de los costos totales de operación.

Las características que distinguen a un ERP son las siguientes:

Integrales: Permiten controlar los diferentes procesos de la compañía entendiendo que todos los departamentos de una empresa se relacionan entre sí, es decir, que el resultado de un proceso es punto de inicio del siguiente. Si la empresa no utiliza un ERP, necesitará tener varios programas que controlen todos los procesos de cobros y pagos, caja e inventario, con la desventaja de que al no estar integrados, la información se duplica, crece el margen de contaminación de la misma (sobre todo por errores de captura) y se crea un escenario favorable para malversaciones. Con un ERP, el operador simplemente captura el pedido y el sistema se encarga de todo lo demás, por lo que la información no se manipula y se encuentra protegida. (Sánchez)

Modulares: Los ERP entienden que una empresa es un conjunto de departamentos que se encuentran interrelacionados por la información que comparten y que se genera a partir de sus procesos. Una ventaja de los ERP, tanto económica como técnica es que la funcionalidad se encuentra dividida en módulos, los cuales pueden instalarse de acuerdo con los requerimientos del cliente. Ejemplo: ventas, materiales, finanzas y control de almacén. (Sánchez)

Adaptables: Los ERP están creados para adaptarse a la idiosincrasia de cada empresa. Esto se logra por medio de la configuración o parametrización de los procesos de acuerdo con las salidas que se

necesiten de cada uno. Por ejemplo, para controlar inventarios, es posible que una empresa necesite manejar la partición de lotes pero otra empresa no. (Sánchez)

Las entidades utilizan los ERP por diferentes motivos:

- Ser competitivas.
- Mejorar la productividad.
- Mejorar la calidad.
- Mejorar los servicios prestados a los clientes.
- Mejorar la planificación y utilización de los recursos.

Las áreas típicas de un ERP son:

- Finanzas.
- Contabilidad.
- Planificación y control de la producción.
- Recursos humanos.
- Costos.
- Ventas.
- Marketing.

Dentro del área de Finanzas se encuentra todo lo referente al módulo Cobros y Pagos.

1.3. La Contabilidad Financiera. Procesos contables de cobros y pagos.

La Contabilidad es el arte de registrar, clasificar y resumir de manera significativa y en términos de dinero, las operaciones y hechos que son de carácter financiero y de interpretar sus resultados. Su objetivo esencial es que esté dirigida a satisfacer totalmente las necesidades de los diferentes usuarios de la información financiera. (Ministerio, 2005)

La Contabilidad según la función que realiza se divide en tres tipos: (Ministerio, 2005)

- Contabilidad Financiera: expresa en términos cuantitativos y monetarios las transacciones que realiza una entidad económica, así como ciertos acontecimientos económicos que le afectan, con el fin de proporcionar información útil y segura a usuarios externos a la organización para su toma de decisiones.
- Contabilidad Administrativa: se orienta a satisfacer las necesidades de información interna de la administración con un diseño destinado a facilitar las funciones administrativas internas de planeación y control, así como la toma de decisiones.
- Contabilidad Fiscal: está diseñada para dar cumplimiento a las obligaciones tributarias de las organizaciones respecto a un usuario específico: el fisco.

La Contabilidad constituye un servicio de excepcional importancia para facilitar: (Ministerio, 2005)

- La toma de decisiones gerenciales, ya que constituye una eficaz herramienta de dirección.
- La planeación y el control de las operaciones.
- El control por áreas de responsabilidad.
- Las coordinaciones de las actividades desarrolladas.
- La información a los interesados externos, al consejo de dirección y al colectivo de trabajadores.
- La exigencia por parte de la administración y el análisis de la eficiencia.
- La formulación, complementación y apreciación de normas administrativas.

La Contabilidad Financiera engloba una gran cantidad de procesos contables entre los que se encuentran los relacionados con cobros y pagos.

Dentro de los principales conceptos referentes a los procesos de cobros y pagos se encuentran:

- Obligación fiscal.
- Derecho fiscal.
- Derecho de cobro.
- Cobro anticipado.

- Pago anticipado.
- Obligación de pago.

1.4. Sistemas de Contabilidad Financiera existentes en las entidades cubanas.

Dentro de las 26 soluciones empresariales con las que cuenta Cuba se encuentra el sistema integral económico creado por CITMATEL llamado RODAS XXI, el cual entró en el mercado en el año 1999. RODAS XXI posibilita automatizar el funcionamiento de cualquier empresa o unidad presupuestada, es un sistema en constante desarrollo que tiene muy en cuenta la opinión de los clientes para perfeccionarse. Es un sistema multimonedado o sea pueden realizarse operaciones tanto en moneda nacional como extranjera.

Principales módulos de RODAS XXI: (CITMATEL)

Contabilidad: Incluye importación, realización y reversión de comprobantes de operaciones. Establece cinco niveles de análisis contables, contando a las cuentas principales, subcuentas y tres análisis dentro de cada subcuenta. Contiene una gran variedad de informes que permiten obtener desde las facturas por edades de un cliente en particular hasta el comprobante específico donde se contabilizó una factura de un proveedor, además permite la configuración de estos informes. Ofrece gran variedad de reportes como son balance general, balance de comprobación, estado de resultados, análisis de cuentas por pagar, cuentas por cobrar.

El módulo de contabilidad de RODAS XXI brinda la posibilidad de volver a períodos contables ya cerrados e incluso a períodos de años anteriores para visualizar informes correspondientes a dichos períodos, aunque por supuesto, en períodos ya cerrados no le permite realizar ninguna operación. Esta característica es compartida por todos los módulos de RODAS XXI.

Automatiza el proceso de cierre del año mediante una opción configurable que permite la realización del comprobante de cancelación de las cuentas de gastos e ingresos transfiriendo sus saldos a la cuenta resultado, quedando solamente por parte del usuario la realización del comprobante de traspaso de saldo de esta cuenta a las cuentas que correspondan antes de cerrar el año.

Admite el registro de cheques tanto emitidos, como recibidos, así como de otros instrumentos y efectúa operaciones de cobros y pagos con los mismos.

Inventario: Controla los movimientos de todos los productos. Lleva el historial de su ciclo de vida. Exporta hacia Contabilidad.

Activos fijos: Realiza las operaciones referentes a los activos fijos y obtiene la información actualizada a través de diversos reportes que brinda el subsistema.

Nómina: Facilita la definición y configuración de las nóminas necesarias para el pago según las características de la entidad.

Finanzas: Brinda información financiera actualizada. Se obtienen reportes de control de recibo de efectivo, de depósitos, de reintegros, dieta, vales de pagos menores.

Facturación: Permite la elaboración de facturas, cerrar la venta de un ciclo de un producto determinado a un cliente y justifica cómo el cliente paga el servicio que se le brinda.

Recursos Humanos: Diseñado e implementado en un entorno multiusuario para ambiente web. Permite proponer, controlar y evaluar la política para la movilidad y desarrollo del potencial humano. (CITMATEL)

Telecobranzas: Este módulo permite organizar estratégicamente la función de cobro y efectuar un seguimiento más estrecho de las gestiones con su cliente.

En la actualidad se encuentra en más de 400 entidades del país en las que hay aproximadamente 1000 computadoras con el sistema instalado.

Otra solución nacional es el SISCONT5, sistema integral de contabilidad elaborado por el MINBAS, el cual se aplica en cientos de entidades en el país. Se encuentra en su versión número 5 y cuenta con 8 módulos integrados.

Módulos de SISCONT5:

Contabilidad: Fue el primer módulo que se creó. Cuenta con fácil ingreso de datos (en una sola pantalla se ingresan todos los datos del documento). Actualiza en línea. Exporta al Excel 2003 para generar informes y formatos. Genera la diferencia de cambio y asientos automáticos. Al realizar el cierre del mes de los procesos contables de cobros y pagos si existe diferencia entre submayor, histórico y cuenta, SISCONT5 posibilita al cliente dar solución a este problema. Además cuando se

comienzan a realizar operaciones de Cobros anticipados y Pagos anticipados este sistema integral de Contabilidad emite un mensaje si la apertura no ha sido cerrada previamente.

Tesorería: Permite controlar y programar cuentas por pagar. Actualiza en línea el flujo de caja. Emite cheques vouchers en forma directa.

Caja chica: Este módulo permite controlar los gastos de caja chica, fondos fijos y/o fondos por rendir. Con sólo ingresar los datos del documento en una sola pantalla se genera el asiento contable.

No se requiere de conocimientos contables para el manejo de este módulo.

Créditos y cobranzas: Permite la gestión de cobranza de clientes con herramientas de mantenimiento de cartera, que actualiza en línea su flujo de caja. Tiene una pantalla que facilita la cancelación del cliente con diferentes tipos de pago, y registros automáticos de la cancelación de retenciones.

Gestión de negocios: Este módulo permite tener un balance de resultado detallado entre ingresos menos costos y gastos por cada área de su negocio, sucursales, líneas de producción, proyectos, servicios y otros.

Presupuestos: Este módulo es importante en la actualidad por la gran competencia entre las entidades, sus áreas de negocio deben proyectar las actividades por cada ciclo de negocio. Brinda un comparativo de los ingresos y gastos de lo estimado con lo real.

Informes gerenciales: Los informes están estructurados en Excel 2007 con tablas dinámicas que generan reportes a la medida de su negocio con un mínimo de esfuerzo en bancos, clientes, proveedores, unidades de negocio, presupuesto y balance.

Información compartida entre sistemas: Permite recibir información en archivo de texto desde sistemas externos o de terceros proveedores.

Con SISCONT5 se puede trabajar en forma monousuario, multiusuario y por Internet, dependiendo de el tipo de hardware (dispositivo).

El Versat Sarasola constituye otra de las soluciones nacionales con las que cuenta el país actualmente. Surge en el año 1998 debido a una necesidad del Ministerio Cubano del Azúcar de realizar una versión cubana que le permitiera obtener una mayor eficiencia en su gestión contable. (Cabrera)

Es el primer sistema de Contabilidad cubano certificado según las nuevas normas emitidas por el Ministerio de Finanzas y Precios y de la Informática y las Comunicaciones para este tipo de software.

Actualmente lo utilizan alrededor de 200 entidades de varias provincias y en lo adelante lo introducirán más de dos mil 500 unidades presupuestadas, entre las que figuran organismos de la Administración Central del Estado, las direcciones municipales de finanzas, tesorerías y la ONAT.

Cuenta con 10 módulos o subsistemas.

Módulos del Versat Sarasola:

Configuración: Se definen y controlan todas las actividades principales del resto de los subsistemas, así como los usuarios y permisos de cada uno de ellos.

Contabilidad general: Corazón del sistema. Hacia este módulo tributan todas las operaciones contables de los restantes subsistemas. En este subsistema, los comprobantes de operaciones a procesar manualmente son mínimos, pues la casi totalidad de los mismos serán generados por los restantes módulos.

Costos y procesos: Es un complemento del módulo Contabilidad general, pues además del registro contable de gastos, incluye dos actividades muy relevantes: El traspaso o distribución de los gastos indirectos hacia los centros de costo directo y el ajuste o costeo, según los volúmenes de producción o servicios y sus destinos.

Control de inventarios: Registra todos los documentos primarios que tradicionalmente generan las entradas y salidas en los almacenes; además tiene incorporada una gran cantidad de opciones que permiten ejecutar los conteos físicos y el tratamiento de los diferentes tipos de inventarios, que para el subsistema se llaman categorías.

Control de activos fijos: Recoge las operaciones normales que en esta actividad se realizan (altas, bajas, modificaciones de los medios).

Facturación: Subsistema que cierra el ciclo de lo que se pudiera considerar como las actividades de “control” dentro de cualquier entidad.

Planificación económica / productiva: Sustentada sobre los mecanismos o principios del Presupuesto Maestro (tecnología conocida mundialmente, pero poco difundida en Cuba).

Generador de reportes: También llamado “Complementos del Versat”, herramienta que permite obtener información sin mayores complejidades de uso, expresando sus resultados en una gama de funciones tipo y predeterminadas.

Nómina de salarios: El subsistema es lo suficientemente adecuado a los diversos sistemas de pago, estimulaciones y demás regulaciones laborales que existen en el país.

Finanzas, caja y banco: Es de los más abarcadores dentro del Versat Sarasola, pues dentro de las 5 actividades que posee, están recogidas la casi totalidad de las operaciones financieras que pueden generarse en cualquier entidad. Dentro de este módulo se encuentra todo lo referente a la gestión de los cobros y pagos. En esta actividad se procesan o muestran todos los documentos referidos a las operaciones que afectan directamente las diferentes cuentas que controla este subsistema y que tienen relación con las obligaciones de cobros y pagos. Se procesan las facturas emitidas y recibidas si no se instalan los subsistemas de Inventario y de Contratación o Facturación, se procesan además instrumentos de pagos emitidos, letras de cambios emitidas y recibidas que han sido aceptadas, así como transitarlas por los diferentes estados (descontadas en la fecha del vencimiento, antes del vencimiento y protestadas). Presenta una variedad de información relacionada con cuentas por cobrar y pagar con niveles de selección, cuentas por edades, registro de facturas emitidas, emitidas canceladas, recibidas, recibidas canceladas, facturas pagadas con detalles en cuanto a los documentos que originaron las operaciones. Controla los cobros y pagos anticipados para su posterior liquidación con las obligaciones correspondientes. Realiza una emisión de los registros de cheques emitidos, y emitidos cancelados.

Todas las soluciones nacionales mencionadas anteriormente son utilizadas en entidades específicas de acuerdo a las operaciones que las mismas realizan. Las nuevas tendencias del sistema empresarial cubano, con estructuras complejas y el uso extendido de las redes e Internet han provocado que los sistemas de gestión económicos, ERP, y otros relacionados con la gestión de las entidades necesiten cada vez más requerimientos, de ahí la necesidad de hacer un sistema de gestión integral cubano que sea genérico para todas las entidades, que sea altamente modular y que sea capaz de reajustarse a las peticiones específicas de cada una de las entidades o ministerios que los requieran.

1.5. Importancia de la automatización del módulo Cobros y Pagos a través del CEDRUX.

Los procesos contables de cobros y pagos tienen gran importancia para el funcionamiento de todas las entidades, por lo que mediante su informatización, se mantendrá de forma organizada y controlada todo lo referente a los mismos, posibilitando la toma de decisiones de la entidad.

En la actualidad muchas entidades realizan estos procesos de forma manual lo cual trae como consecuencia la ocurrencia de errores humanos y hechos delictivos. Estos problemas se solucionan mediante la realización de un sistema informático que gestione los procesos de cobros y pagos de las entidades cubanas, que contribuya a que la información fluya con claridad y mayor precisión permitiendo de esta forma que el sistema contable realizado sea más seguro.

La Universidad de las Ciencias Informáticas (UCI) en colaboración con el Ministerio de Finanzas y Precios (MFP) está desarrollando un sistema integral de gestión empresarial denominado CEDRUX, el cual se ajustará a las resoluciones emitidas por el MFP y permitirá una gestión estándar de los procesos contables en todas las entidades. CEDRUX cuenta con varios módulos dentro de los que se encuentra el de Cobros y Pagos.

La automatización del módulo Cobros y Pagos a través del CEDRUX permite obtener ciertas ventajas a la hora del manejo de estos procesos, por ejemplo a la hora de emitir los análisis por edades de las cuentas se tiene presente la fecha de pago pactada con el cliente o proveedor, lo cual permite obtener el vencimiento correcto de las cuentas. Mediante el módulo Cobros y Pagos se puede cambiar la moneda de registro establecida tanto para cobros como para pagos, conservándose de esta forma la moneda original. Además se emiten conciliaciones de todas las cuentas del módulo (pago anticipado, cobro anticipado, derecho de cobro, obligación de pago), permite registrar cualquier modificación que se necesite hacer y conservarlas en el sistema para cualquier consulta, opción esta totalmente exclusiva que se brinda en el módulo Cobros y Pagos a través del CEDRUX.

Una de las desventajas de los sistemas nacionales con los que cuenta Cuba es que no se adaptan a las nuevas legislaciones emitidas por el MFP, sin embargo en este módulo si se tienen en cuenta las más recientes resoluciones emitidas², incluso tiene presente lo establecido en cuanto a los datos de acceso obligatorio de los submayores analíticos de derechos y obligaciones de todas las cuentas, así

² Resolución 14/2007 del Ministerio de Finanzas y Precios.

como lo normado en materia de Cobros y Pagos por el Banco Central de Cuba en la Resolución 245-2008.

1.6. Modelo de desarrollo orientado a componentes.

Para el desarrollo de un proyecto de la magnitud del CEDRUX, se necesita de un modelo estándar para cada uno de los equipos de desarrollo, para de esta forma tener una definición clara y precisa de las responsabilidades de los roles que se ven involucrados en el desarrollo de la solución.

La propuesta realizada es un modelo de desarrollo orientado a componentes, desarrollado en colaboración con las Líneas de desarrollo del proyecto ERP, de acuerdo con las necesidades que han presentado cada una de ellas y teniendo en cuenta los principales riesgos con los que se cuentan en el proyecto. (2009)

Estructura del Modelo de desarrollo orientado a componentes:

- Estructura de las Líneas de desarrollo:

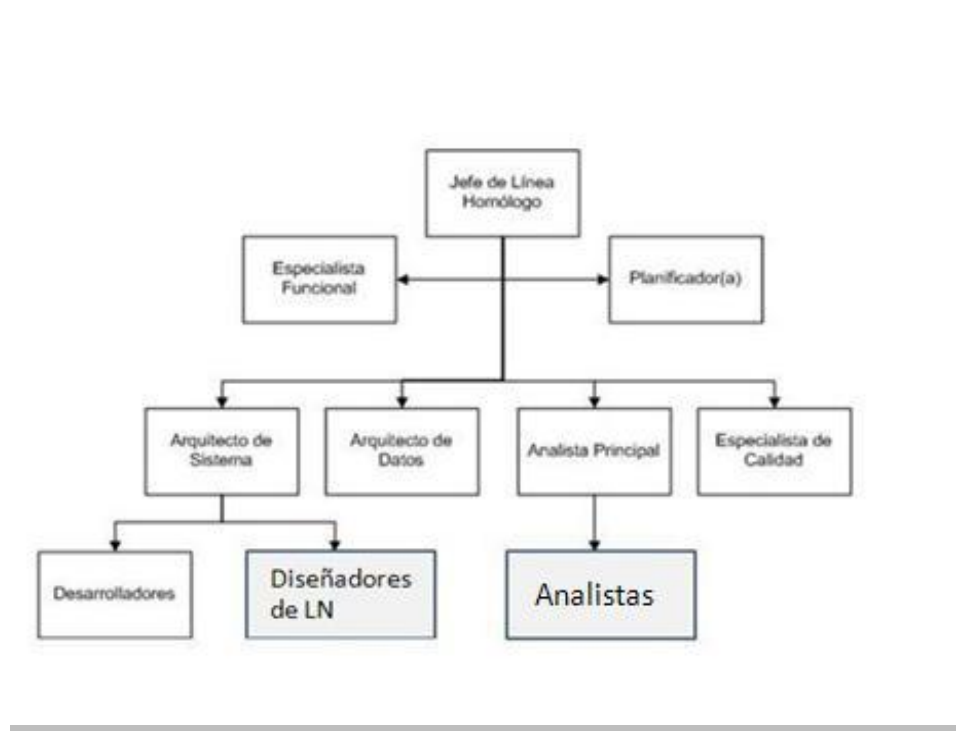


Figura 1. Estructura de las Líneas de desarrollo.

➤ Flujo de actividades del Modelo de desarrollo orientado a componentes.



Figura 2. Flujo de actividades.

➤ Principales Artefactos del modelo de desarrollo orientado a componentes.

Artefactos	Interno	Entregable
Plan de Iteración	X	
Plan de Gestión de Riesgos	X	
Plan de Trabajo individual de los integrantes de la línea	X	

Cronograma de desarrollo	X	
Diagrama de componentes	X	
Prioridad de los componentes	X	
Agrupación Requerimientos - Componentes	X	
Informe de Integración	X	
Modelo de Datos	X	
Descripción del Modelo de Dato	X	
Mapa de Procesos de la Línea	X	
Descripción de procesos de negocio	X	
Modelo Conceptual	X	
Especificación de requisitos		X
Casos de Prueba	X	
Plan de pruebas	X	
Registro de No Conformidades	X	
Descripción de los componentes	X	
Implementación de componentes	X	
Diagrama de Clases	X	
Descripción del Diseño de Clases	X	
Manual de Usuario		X

En el módulo Cobros y Pagos se realizó todo según lo definido en el modelo de desarrollo propuesto, logrando de esta forma una mayor organización en el trabajo realizado.

1.7. Actividades a desarrollar por el rol Analista y Diseñador de lógica del negocio.

La persona que desempeña el rol de analista del negocio, es aquella encargada de estudiar diversos problemas relacionados con el mercado de sus clientes y busca la forma de encontrar soluciones a los mismos. Los analistas del negocio realizan un análisis de las necesidades de la organización para la que trabajan, consiguiendo estrategias de solución apropiadas para las mismas. Un analista identifica necesidades y beneficia a la organización con la puesta en práctica de una solución técnica, o sea construye un puente entre el negocio y él. (2009)

Los autores del presente trabajo de diploma según el modelo de desarrollo propuesto los roles que desempeñan son los de analista y diseñador de lógica del negocio.

Principales actividades a desarrollar por el analista:

- Participar en las sesiones de trabajo para identificar, describir y validar los procesos de negocio y los requisitos de software
- Elaborar la descripción de procesos de negocio, especificación de requisitos y casos de prueba según los estándares establecidos para ello.
- Participar en el taller de Diseño.

Principales actividades a desarrollar por el diseñador de lógica del negocio:

- Participar en el taller de análisis.
- Diseño de lógica del negocio.
- Reunión de implementación.

Descripción de las actividades a desarrollar por el analista:

- Identificación de procesos: Debe identificar, analizar y describir los procesos que se llevan a cabo en el negocio que se desea automatizar, con el objetivo de organizar y documentar todas las acciones a tener en cuenta en el análisis para el desarrollo del software.
- Validación de procesos: Se aprueba que la identificación de los procesos se hizo correctamente y que el equipo tiene plena claridad del negocio a automatizar.
- Identificación de requerimientos: A partir de los procesos identificados se realiza la identificación de los requisitos o funcionalidades que deben cumplir, para que pueda ser realizado dicho proceso.
- Validación de requerimientos: Se validan todos los requisitos identificados para cada uno de los procesos que intervienen en la automatización del negocio deseado.
- Taller de análisis: Se evalúan cada uno de los requerimientos y procesos identificados y validados, a partir de los cuales se desarrolla el mapa de procesos a través del cual: se deben agrupar los requerimientos y procesos por componentes, identificar las dependencia entre los

componentes, así como los contratos de los mismo en el módulo y se establecen las prioridades de desarrollo de cada uno de los componentes, dependiendo de cuan críticos y complejos sean.

- Diseño de modelo de datos: Partiendo de un modelo lógico obtenido durante la captura de requisitos, se definen las estructuras de base de datos que darán soporte de persistencia a la solución de software orientada a los componentes identificados.
- Creación del modelo de datos: Creación de las estructuras y objetos de base de datos en el sistema de gestión seleccionado orientado a los componentes identificados.
- Casos de prueba: Construcción de todos los posibles caminos de ejecución, o escenarios, de cada componente desarrollado. Se obtiene como resultado un listado final con los casos de prueba identificados a partir de los posibles escenarios, los resultados esperados para cada caso y las condiciones o valores requeridos para la ejecución de los distintos escenarios.

Descripción de las actividades a desarrollar por el diseñador de lógica del negocio:

- Taller de análisis: Se evalúan cada uno de los requerimientos y procesos identificados y validados, a partir de los cuales se desarrolla el mapa de procesos a través del cual: se deben agrupar los requerimientos y procesos por componentes, identificar las dependencia entre los componentes, así como los contratos de los mismo en el módulo y se establecen las prioridades de desarrollo de cada uno de los componentes, dependiendo de cuan críticos y complejos sean.
- Diseño de lógica del negocio: Se diseñan los métodos y clases para dar solución a todas las necesidades detectadas durante la identificación de componentes, ajustándose a las funcionalidades previstas.
- Reunión de implementación: Se realiza una breve descripción con los implementadores explicándole las órdenes de desarrollo y la forma operativa de la lógica del negocio.

Artefactos que genera el analista:

- Plan de trabajo individual.
- Modelo de procesos de negocio.
- Descripción de procesos de negocio.
- Modelo conceptual.
- Prototipo de IU.
- Especificación de requisitos.

Artefactos que genera el diseñador de lógica del negocio:

- Diagrama de clases del diseño.
- Descripción de las clases del diseño.

1.8. Notación de modelado de procesos de negocio BPMN.

La Notación de Modelado de Procesos de Negocio (BPMN) fue desarrollada por la Iniciativa de Administración de Procesos de Negocio (BPMP) para Diagramas de Procesos de Negocio (BPD) con los siguientes objetivos: (2007)

- Proveer una notación entendible para analistas del negocio, desarrolladores técnicos y hasta personas propias del negocio.
- Crear un puente estandarizado entre el diseño de procesos de negocio y su implementación.
- Asegurar que los lenguajes para la ejecución de procesos de negocio puedan ser visualizados con una notación común.

BPMN es un nuevo estándar de modelado de procesos de negocio, en donde se presentan gráficamente las diferentes etapas del mismo. La notación ha sido diseñada específicamente para coordinar la secuencia de procesos y los mensajes que fluyen entre los diferentes procesos participantes. Está dirigido a personas de negocios, vendedores y proveedores de servicios que necesitan comunicar sus procesos de negocio en una forma estandarizada. (2009)

BPMN define diagramas de procesos de negocios basados en la técnica de diagramas de flujo, adaptados para graficar las operaciones de los procesos de la organización. Se compone de un conjunto de elementos gráficos que facilitan un diagrama entendible tanto por audiencias de negocios como técnicas. (2009)

El éxito del BPMN, ya en sus primeros años de vida ha sido tan importante, que se ha convertido rápidamente en el standard hegemónico en el mundo de la modelación de procesos de negocio hasta el punto de que las principales compañías que ofrecen herramientas de gestión por procesos están dirigiendo sus esfuerzos a la readaptación de la modelización de sus productos siguiendo dicho standard. (2009)

Tipos de BPD según BPMN:

- Procesos de negocio privados (internos): Desarrollados por una única entidad de negocio. Definen las actividades que desarrolla la entidad y que no son, por lo general, visibles a las otras entidades.
- Procesos abstractos (públicos): Representa la interacción entre un proceso privado y otro público. Las actividades internas del proceso público no son mostradas.
- Procesos colaborativos (globales): Describen la interacción entre dos o más entidades del negocio. No describen el proceso desde el punto de vista particular de algún participante.

Principales elementos de BPMN:

Los principales elementos de BPMN son:(Consultar Anexo 1.)

- Calles.
- Objetos de flujo.
- Eventos.
- Objetos de conexión.

Principales artefactos de BPMN:(Consultar Anexo 2.)

- Objetos de datos.
- Anotación.

- Grupo.

Ventajas de BPMN:

- Considera un único diagrama para la representación de los procesos (BPD).
- Pensado para ser asignado con naturalidad a lenguajes de ejecución.
- Fácil de entender para los analistas de negocios, desarrolladores técnicos y las personas de negocio que administrarán y controlarán esos procesos.

Desventajas de BPMN:

- Es un lenguaje nuevo que tiene poca documentación lo cual dificulta su estudio.

El modelado de procesos de negocio es de interés en diferentes campos tales como el empresarial y el de ingeniería de software. Esto es debido a que su importancia no solo radica en la descripción del proceso, sino que además generalmente representa una fase preparatoria para actividades tales como: la mejora de procesos de negocio, la reingeniería de procesos de negocio, la transferencia tecnológica y la estandarización del proceso.

1.9. Lenguaje unificado de modelado (UML).

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

Principales motivos por los que fue creado UML:

- La necesidad de modelar sistemas.
- Las tendencias en la industria de software.
- Unificar los distintos lenguajes y métodos existentes.
- Innovar los modelos para adaptarse a la arquitectura distribuida.

En la actualidad, UML está consolidado como el lenguaje estándar en el análisis y diseño de sistemas de cómputo. Entre más complejo es el sistema que se desea crear más beneficios presenta el uso de UML. (Tarazona & Gómez)

UML incluye los siguientes diagramas:

- Diagrama de casos de uso.
- Diagrama de clases.
- Diagrama de objetos.
- Diagrama de secuencia.
- Diagrama de colaboración.
- Diagrama de estados.
- Diagrama de actividades.
- Diagrama de componentes.
- Diagrama de despliegue.

Los analistas del módulo Cobros y Pagos emplearon BPMN para el modelado de los procesos y UML para el resto del modelado.

1.10. Herramienta Case a utilizar. Visual Paradigm.

Visual Paradigm es una herramienta Case³ que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue.

Su mayor éxito consiste en la capacidad de ejecutarse sobre diferentes sistemas operativos lo que le confiere la característica de ser multiplataforma. Visual Paradigm utiliza UML como lenguaje de modelado ofreciendo soluciones de software que permiten a las organizaciones desarrollar las aplicaciones de calidad más rápido, bien y más barato. Es muy fácil de usar y presenta un ambiente gráfico agradable para el usuario. Su notación es muy parecida a la estándar, permite configurar las líneas de redacción, el modelado de base de datos, el modelado de requerimientos, el modelado del proceso de negocio, la interoperabilidad, la generación de documentación y la generación de código base para diferentes lenguajes de programación como Java, C# y PHP.

³ Ingeniería de Software Asistida por Computadora del inglés Computer-Aided Software Engineering.

Visual Paradigm soporta hasta la fecha UML 2.1 completo y BPMN, permite realizar ingeniería tanto directa como inversa, es colaborativa, es decir, soporta múltiples usuarios trabajando sobre el mismo proyecto; genera la documentación del proyecto automáticamente en varios formatos como Web o .Pdf, y permite control de versiones.

Visual Paradigm ofrece:

- Entorno de creación de diagramas para UML 2.1
- Diseño centrado en casos de uso y enfocado al negocio que generan un software de mayor calidad.
- Uso de un lenguaje estándar común a todo el equipo de desarrollo que facilita la comunicación.
- Capacidades de ingeniería directa (versión profesional) e inversa.
- Modelo y código que permanece sincronizado en todo el ciclo de desarrollo.
- Disponibilidad de múltiples versiones, para cada necesidad.
- Disponibilidad en múltiples plataformas.

Diagramas más importantes que incluye Visual Paradigm:

- Diagramas de casos de uso.
- Diagramas de clases.
- Diagramas de secuencia.
- Diagramas de componentes.
- Diagramas de despliegue.
- Diagramas de objetos.
- Diagramas de interacción.
- Diagramas de procesos del negocio.

Debido a las funcionalidades brindadas por el Visual Paradigm y por ser una herramienta multiplataforma se decide utilizarlo como herramienta Case. Además permite la modelación de los

procesos de negocio y la Universidad de las Ciencias Informáticas cuenta con la licencia de esta herramienta, lo cual hace más acertada su selección.

1.11. Conclusiones parciales.

En este capítulo se realizó un estudio de algunos de los principales sistemas de Contabilidad Financiera existentes en las entidades cubanas, destacándose principalmente los principios que rigen las operaciones de cobros y pagos en los mismos. Se plasmó la importancia de la automatización del módulo Cobros y Pagos a través del CEDRUX. También se abordó sobre la herramienta Case a utilizar (Visual Paradigm) y las principales actividades a desarrollar por el analista.

CAPÍTULO II: MODELADO DEL NEGOCIO Y REQUERIMIENTOS

2.1. Introducción.

En el presente capítulo se realiza un breve estudio del estado actual del negocio a considerar para automatizar el módulo Cobros y Pagos. Se presenta el modelo y la descripción de cada uno de los procesos identificados, así como el modelo conceptual correspondiente. Además, se especifican cada uno de los requerimientos funcionales definidos y se presentan los prototipos de interfaz propuestos para cada uno de ellos.

2.2. Modelado de negocio.

Si se pretende comprender y gestionar la complejidad de un sistema por muy pequeño que este sea, es necesario dividirlo en piezas, las cuales se representan a través de modelos que permiten abstraer sus características esenciales. Uno de estos modelos es el modelo de negocio, el cual es utilizado para comprender el conjunto de procesos de negocio que tienen lugar dentro de una organización como paso previo para establecer los requisitos del sistema a desarrollar. Sus objetivos son: (2009)

- Comprender la estructura y la dinámica de la organización en la cual se va a implantar el sistema.
- Comprender los problemas actuales de la organización e identificar las mejoras potenciales.
- Asegurar que los consumidores, usuarios finales y desarrolladores tengan un entendimiento común de la organización.
- Derivar los requerimientos del sistema que va a soportar la organización.

En la presente investigación se determinó que no es necesario realizar un modelo de negocio completo por lo que se realizó lo que se conoce como un modelo de dominio o modelo conceptual.

2.2.1. Modelo de procesos.

Un proceso de negocio es un conjunto estructurado de actividades, diseñado para producir una salida determinada o lograr un objetivo específico. Los procesos describen como es realizado el trabajo en la organización y se caracterizan por ser observables, medibles, mejorables y repetitivos. (Pressman, 2005)

Modelar procesos de negocio es una parte esencial de todos los procesos de desarrollo de software. Este modelo permite capturar el esquema general y los procedimientos que gobiernan el negocio, provee una descripción de dónde se va a ajustar el sistema de software considerado dentro de la estructura organizacional de las actividades habituales y la justificación para la construcción del sistema de software al capturar las actividades manuales y los procedimientos automatizados habituales que se incorporarán en el nuevo sistema, con costos y beneficios asociados. Además, posibilita capturar los eventos, los recursos y las salidas más importantes vinculadas con el proceso de negocio.

Un modelo de proceso de negocio define los siguientes elementos:

- El **objetivo** o el motivo del proceso.
- Las **entradas** específicas.
- Las **salidas** específicas.
- Los **recursos** consumidos.
- La secuencia de las **actividades**.
- Los **eventos** que dirigen el proceso.

Mapa de procesos del módulo Cobros y Pagos:

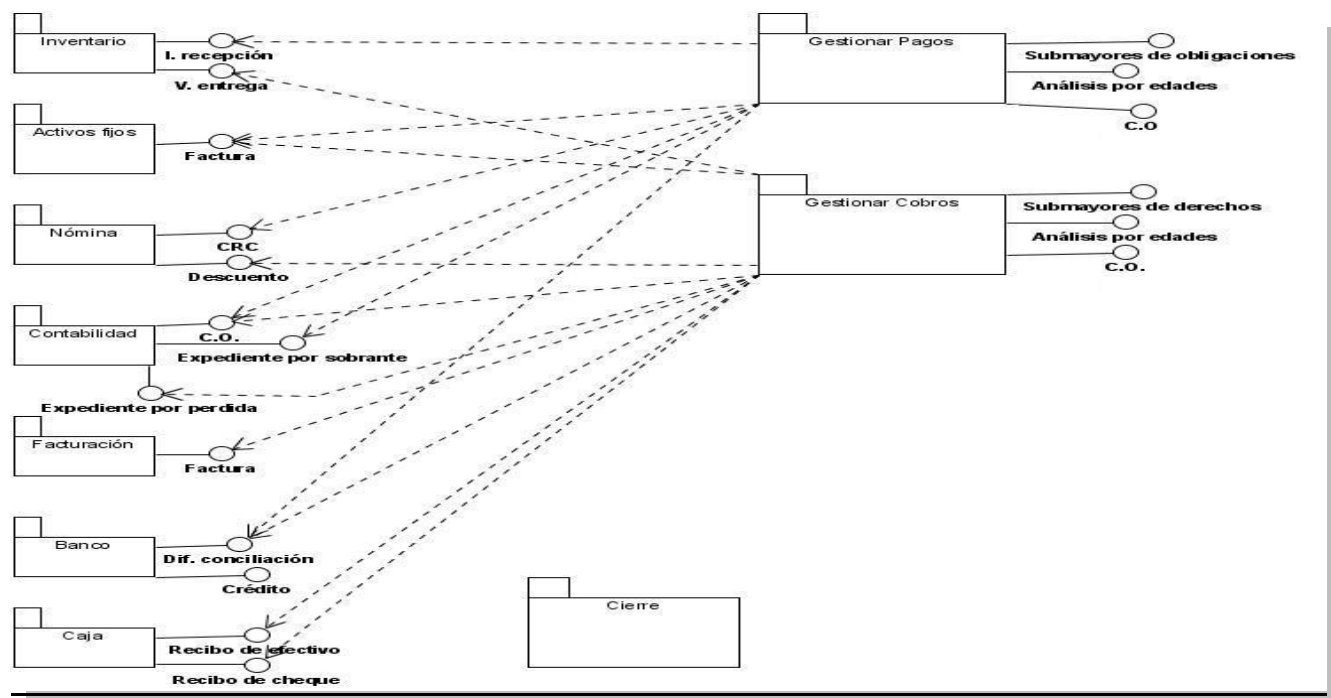


Figura 1. Mapa de procesos.

2.2.2. Descripción de procesos.

En el módulo Cobros y Pagos se definieron los siguientes procesos:

- Realizar cierre.
- Conciliar con cliente.
- Conciliar con proveedor.
- Liquidar cobro anticipado.
- Liquidar compra.
- Liquidar pago anticipado.
- Pagar anticipado.
- Pagar con cheque.

- Pagar con letra de cambio.
- Realizar cobro.
- Realizar compra.
- Realizar pago automático.
- Realizar venta.
- Recibir productos.

Descripción del proceso: Cobrar anticipado.

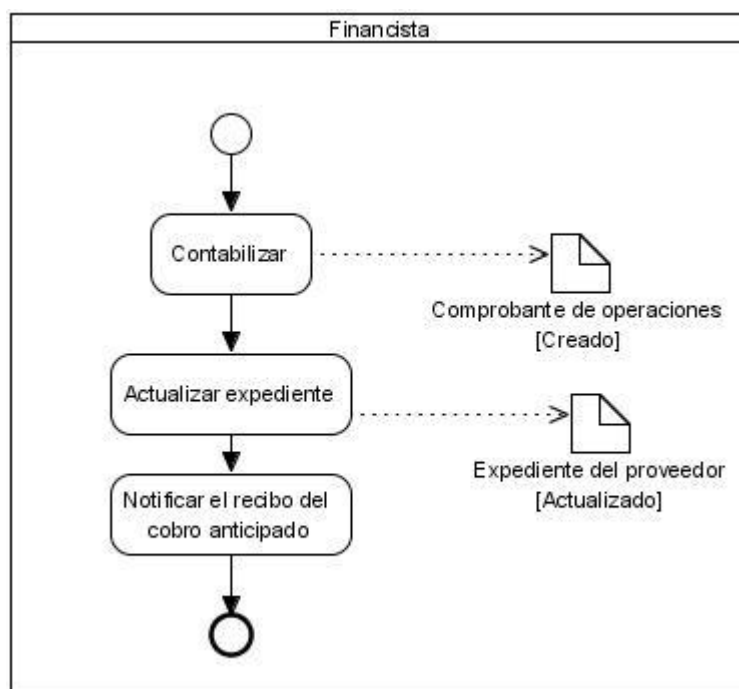


Figura 2. Cobrar anticipado.

Para obtener mayor información sobre la descripción de este proceso, consultar el documento de descripción de procesos realizado por los analistas del módulo Cobros y Pagos. (Analistas, 2009)

Descripción del proceso: Conciliar con cliente.

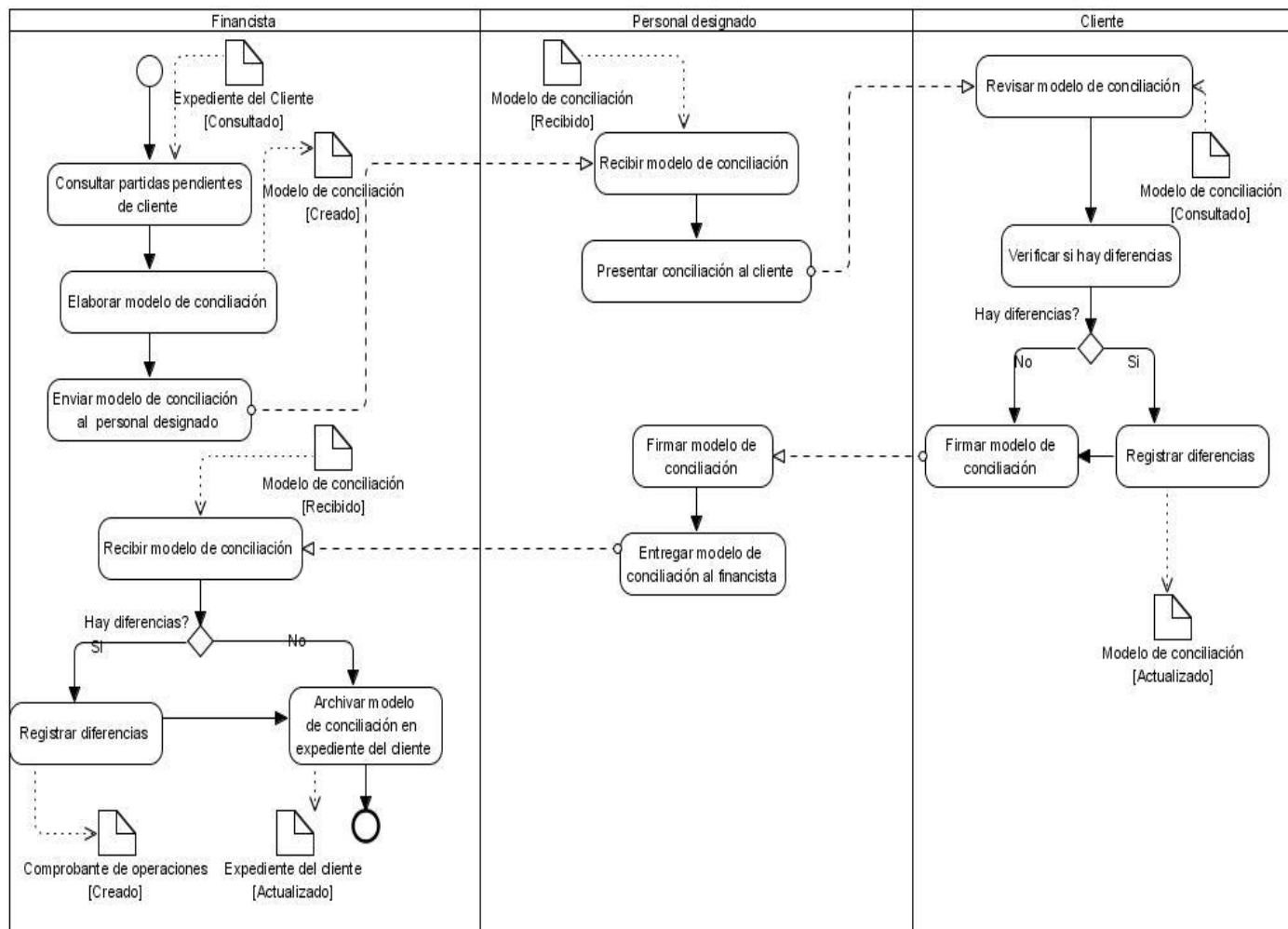


Figura 3. Conciliar con cliente.

Para obtener mayor información sobre la descripción de este proceso, consultar el documento de descripción de procesos realizado por los analistas del módulo Cobros y Pagos. (Analistas, 2009)

Descripción del proceso: Conciliar con proveedor.

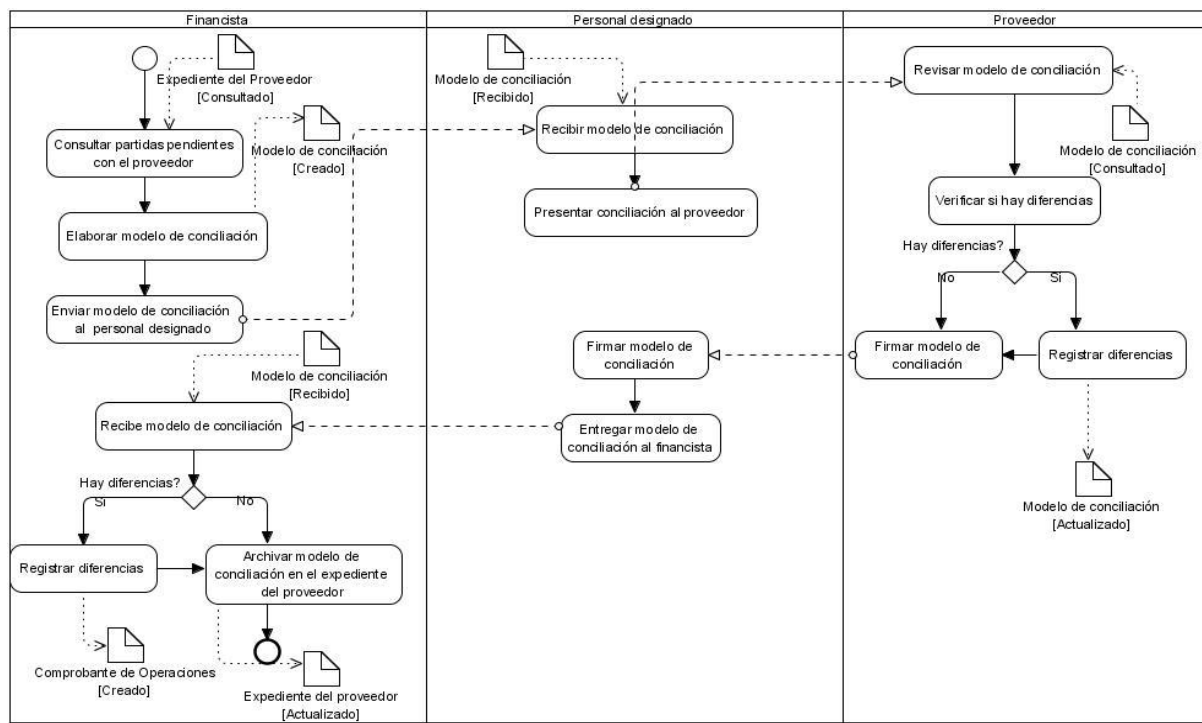


Figura 4. Conciliar con proveedor.

Para obtener mayor información sobre la descripción de este proceso, consultar el documento de descripción de procesos realizado por los analistas del módulo Cobros y Pagos. (Analistas, 2009)

Descripción del proceso: Liquidar cobro anticipado.

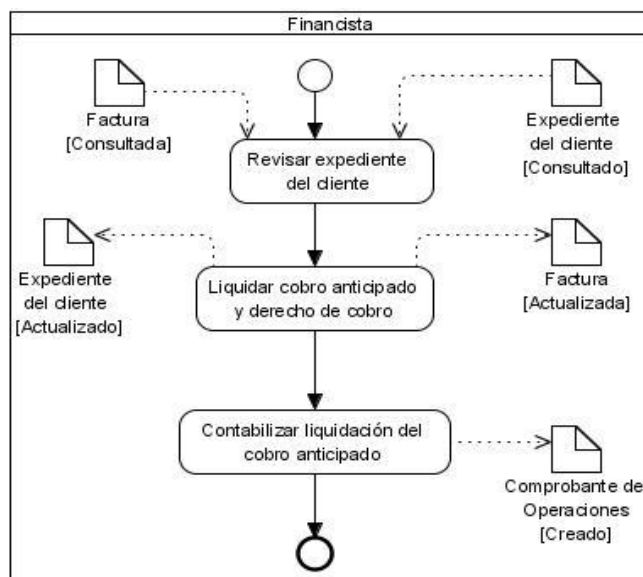


Figura 5. Liquidar cobro anticipado.

Para obtener mayor información sobre la descripción de este proceso, consultar el documento de descripción de procesos realizado por los analistas del módulo Cobros y Pagos. (Analistas, 2009)

Descripción del proceso: Liquidar compra.

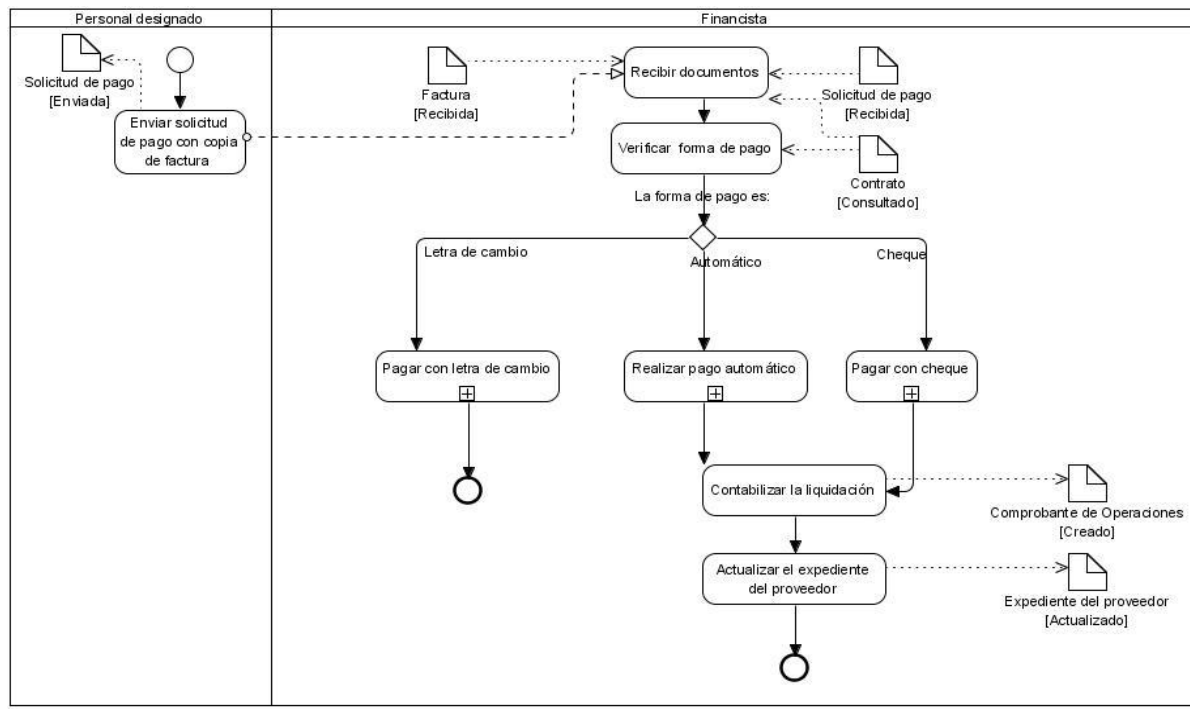


Figura 6. Liquidar compra.

Para obtener mayor información sobre la descripción de este proceso, consultar el documento de descripción de procesos realizado por los analistas del módulo Cobros y Pagos. (Analistas, 2009)

Descripción del proceso: Liquidar pago anticipado.

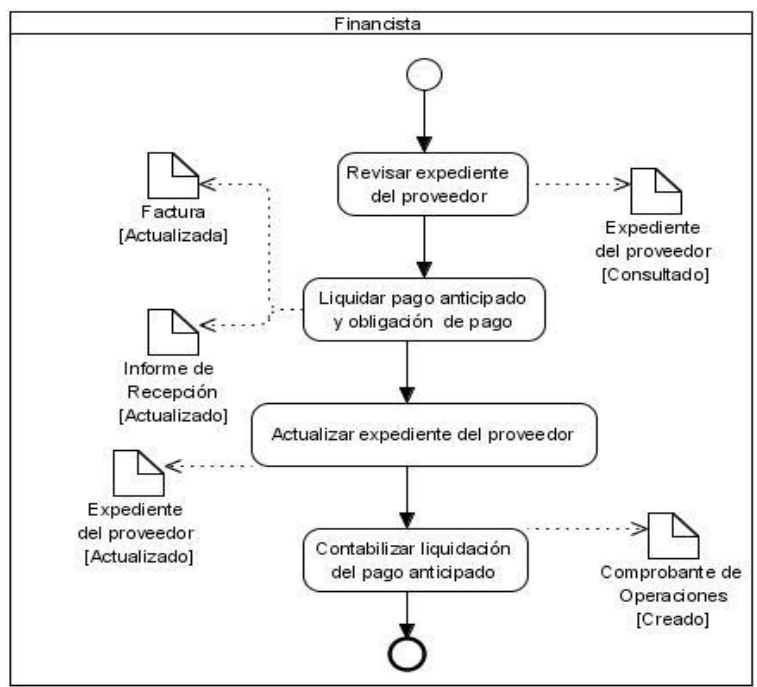


Figura 7. Liquidar pago anticipado.

Para obtener mayor información sobre la descripción de este proceso, consultar el documento de descripción de procesos realizado por los analistas del módulo Cobros y Pagos. (Analistas, 2009)

Descripción del proceso: Pagar anticipado.

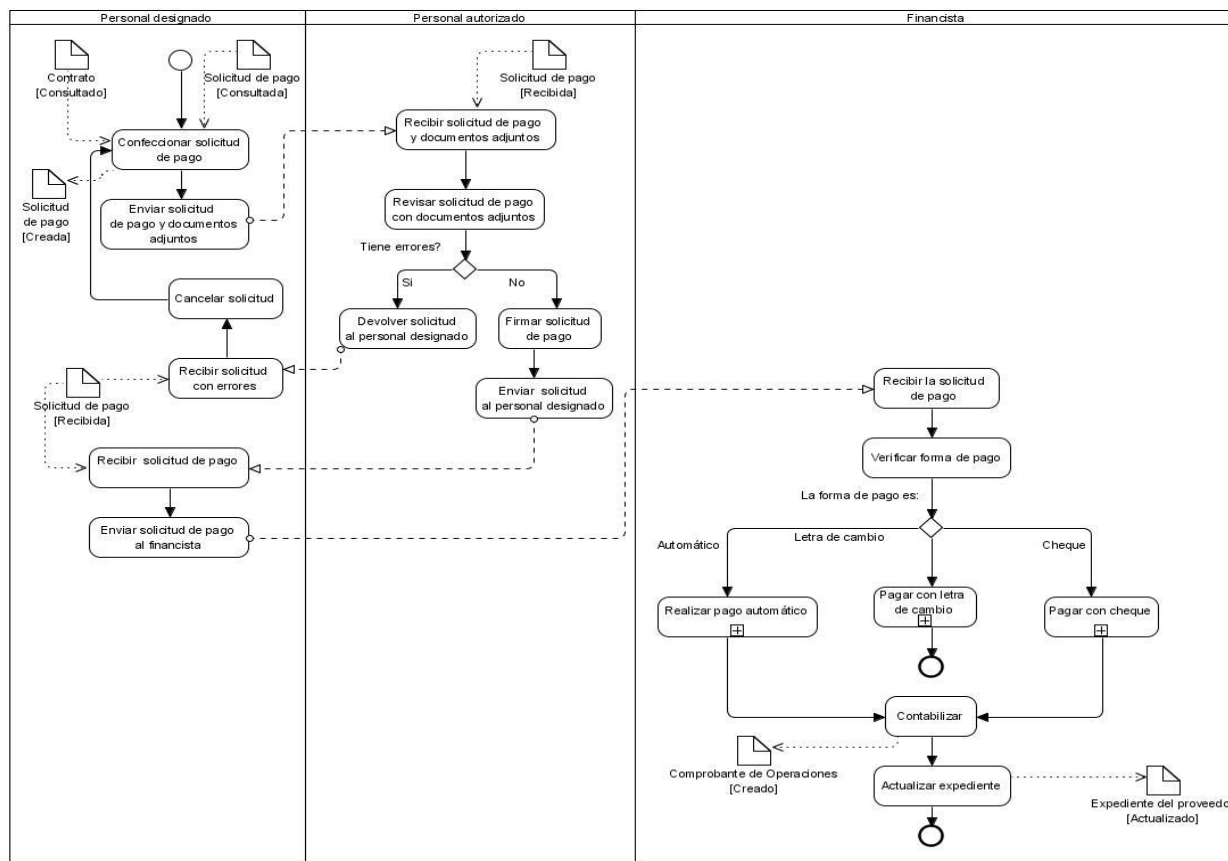


Figura 8. Pagar anticipado.

Para obtener mayor información sobre la descripción de este proceso, consultar el documento de descripción de procesos realizado por los analistas del módulo Cobros y Pagos. (Analistas, 2009)

Descripción del proceso: Pagar con cheque.

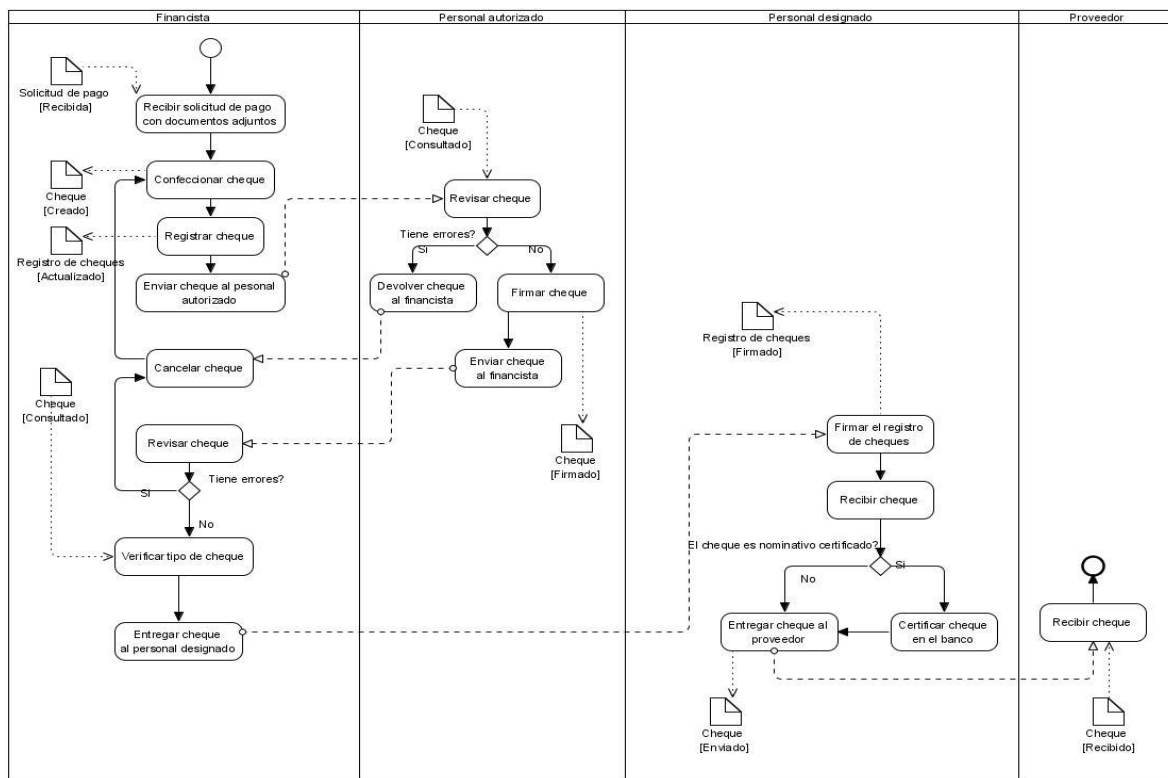


Figura 9. Pagar con cheque.

Para obtener mayor información sobre la descripción de este proceso, consultar el documento de descripción de procesos realizado por los analistas del módulo Cobros y Pagos. (Analistas, 2009)

Descripción del proceso: Pagar con letra de cambio.

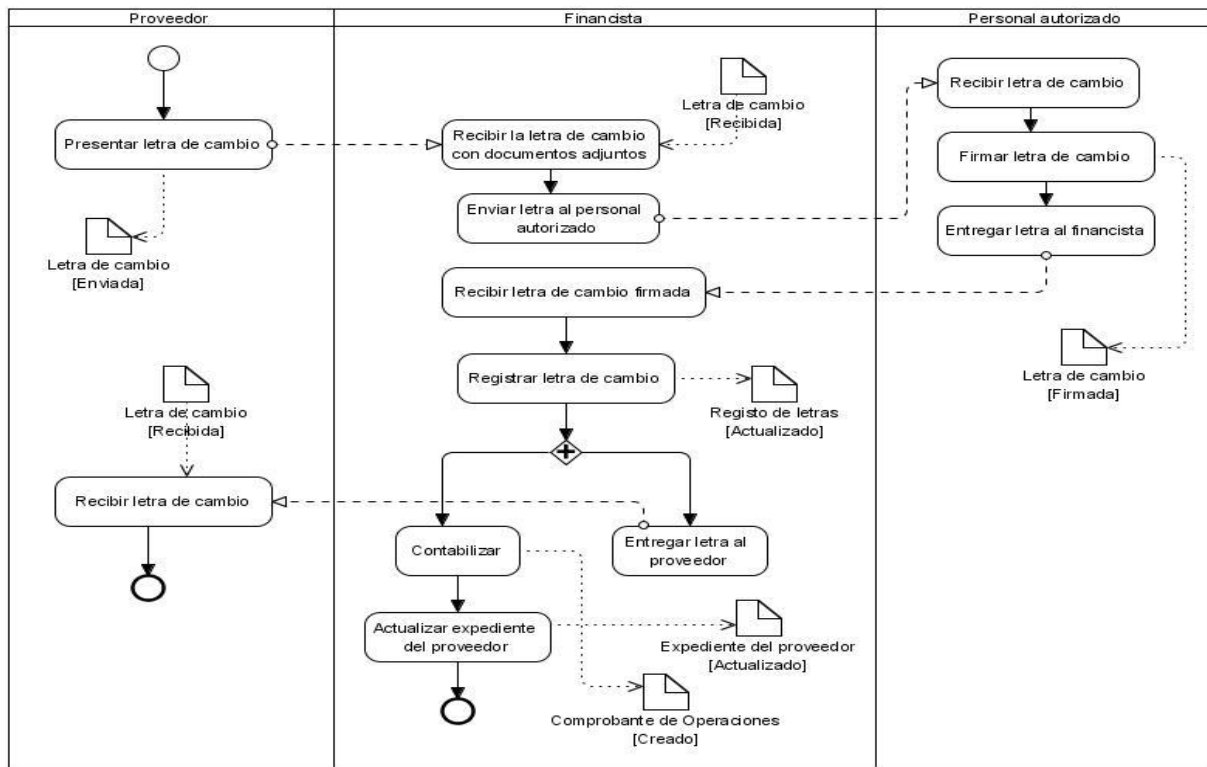


Figura 10. Pagar con letra de cambio.

Para obtener mayor información sobre la descripción de este proceso, consultar el documento de descripción de procesos realizado por los analistas del módulo Cobros y Pagos. (Analistas, 2009)

Descripción del proceso: Realizar cobro.

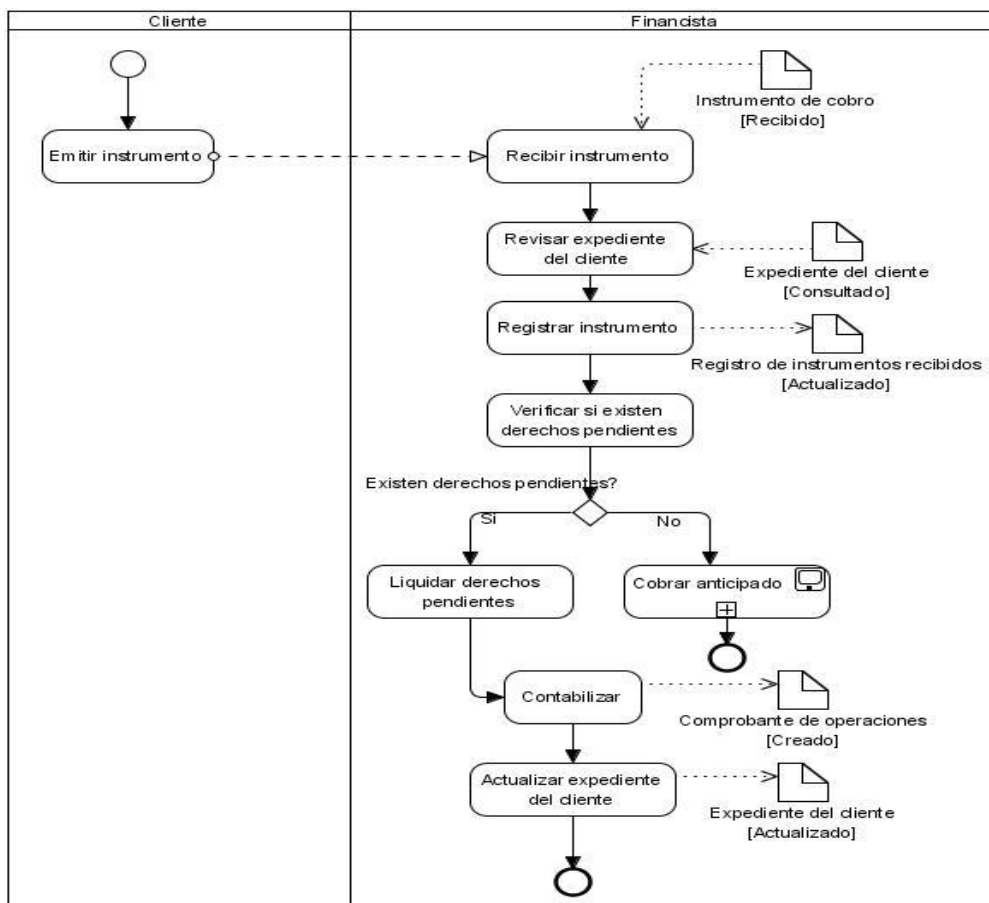


Figura 11. Realizar cobro.

Para obtener mayor información sobre la descripción de este proceso, consultar el documento de descripción de procesos realizado por los analistas del módulo Cobros y Pagos. (Analistas, 2009)

Descripción del proceso: Realizar compra.

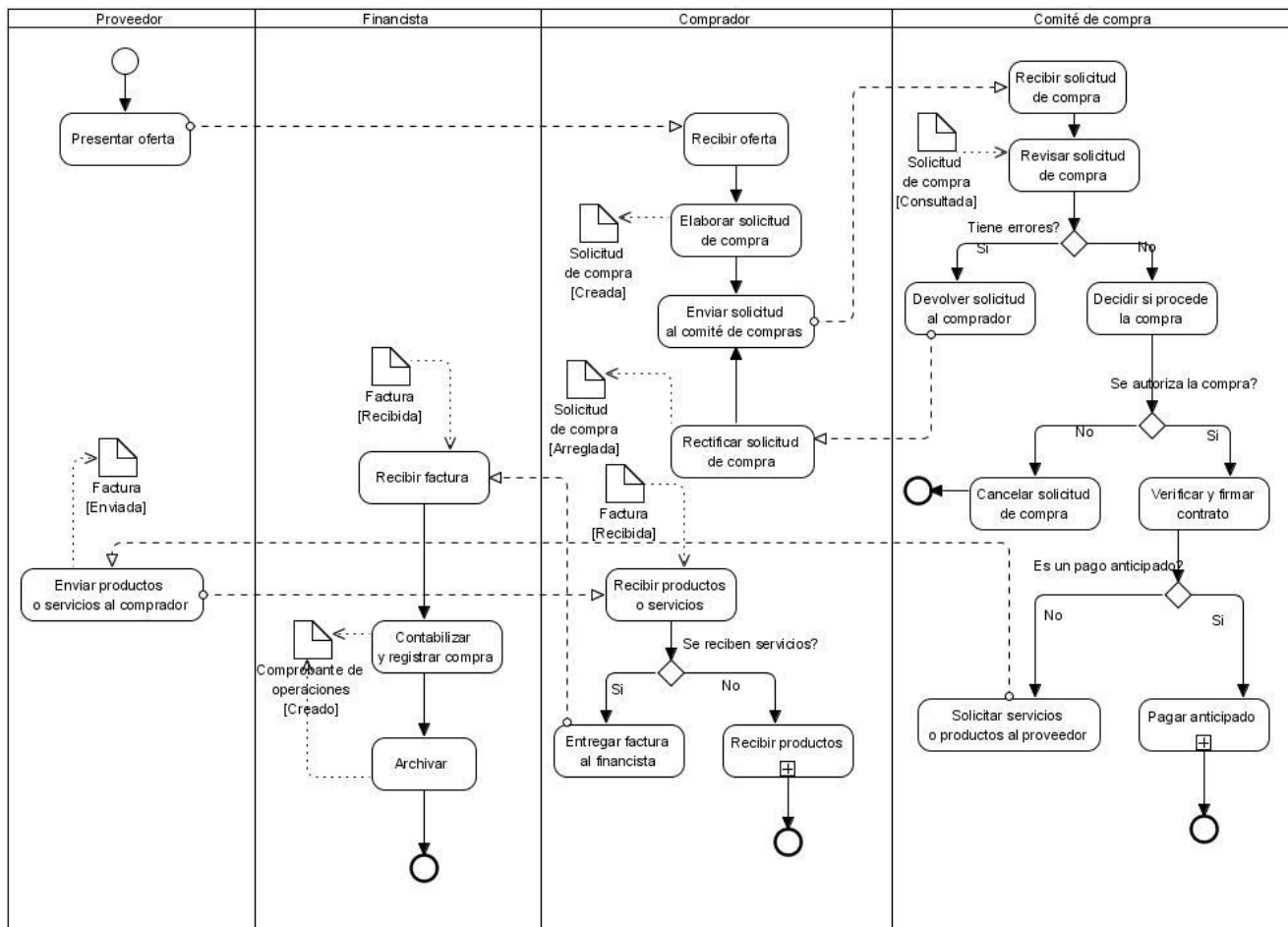


Figura 12. Realizar compra.

Para obtener mayor información sobre la descripción de este proceso, consultar el documento de descripción de procesos realizado por los analistas del módulo Cobros y Pagos. (Analistas, 2009)

Descripción del proceso: Realizar pago automático.

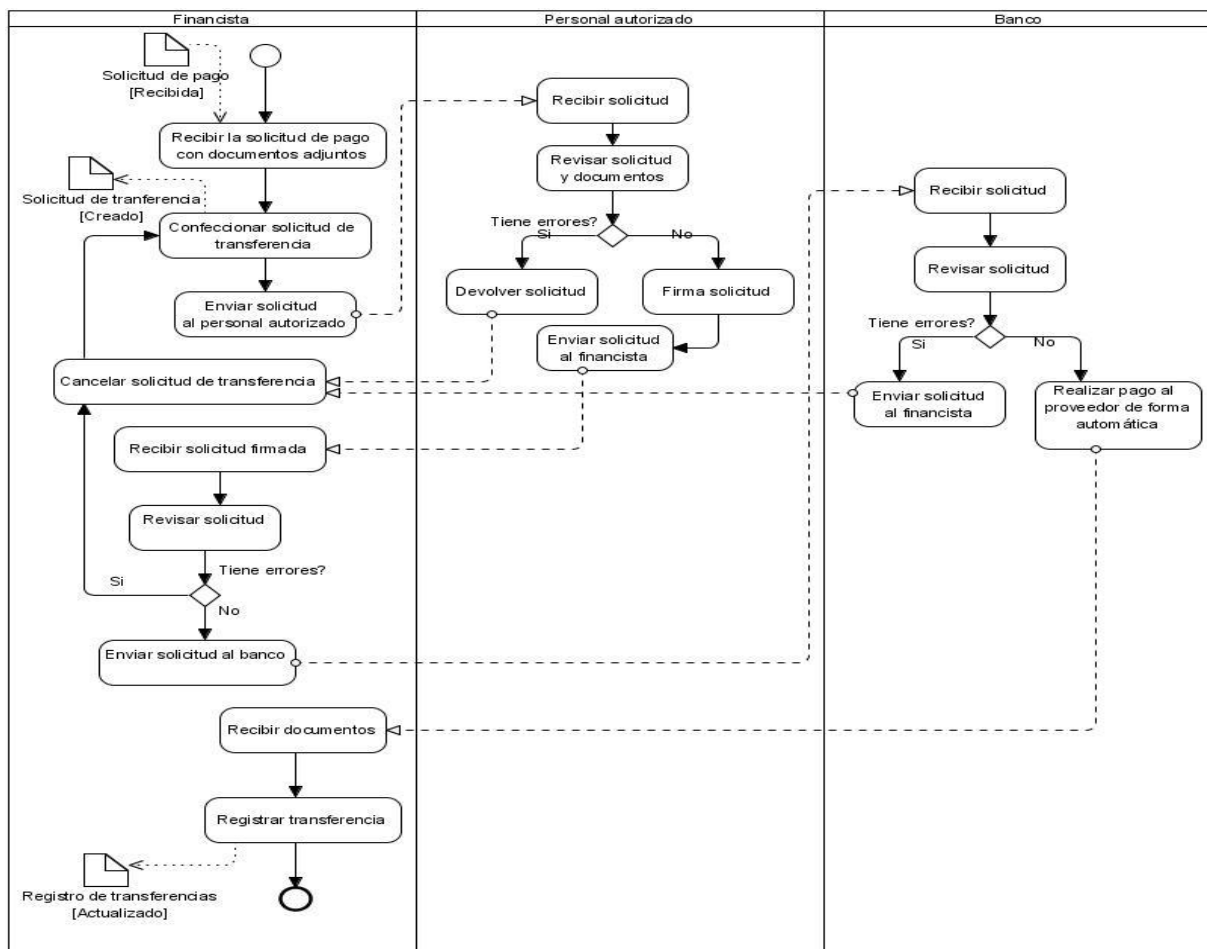


Figura 13. Realizar pago automático.

Para obtener mayor información sobre la descripción de este proceso, consultar el documento de descripción de procesos realizado por los analistas del módulo Cobros y Pagos. (Analistas, 2009)

Descripción del proceso: Realizar venta.

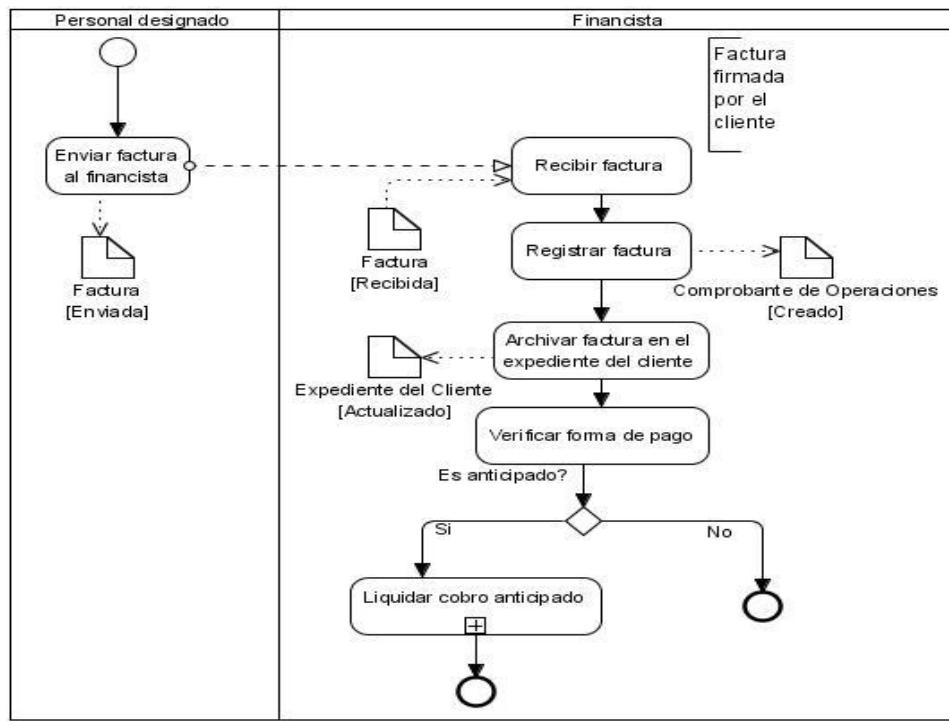


Figura 14. Realizar venta.

Para obtener mayor información sobre la descripción de este proceso, consultar el documento de descripción de procesos realizado por los analistas del módulo Cobros y Pagos. (Analistas, 2009)

Descripción del proceso: Recibir productos.

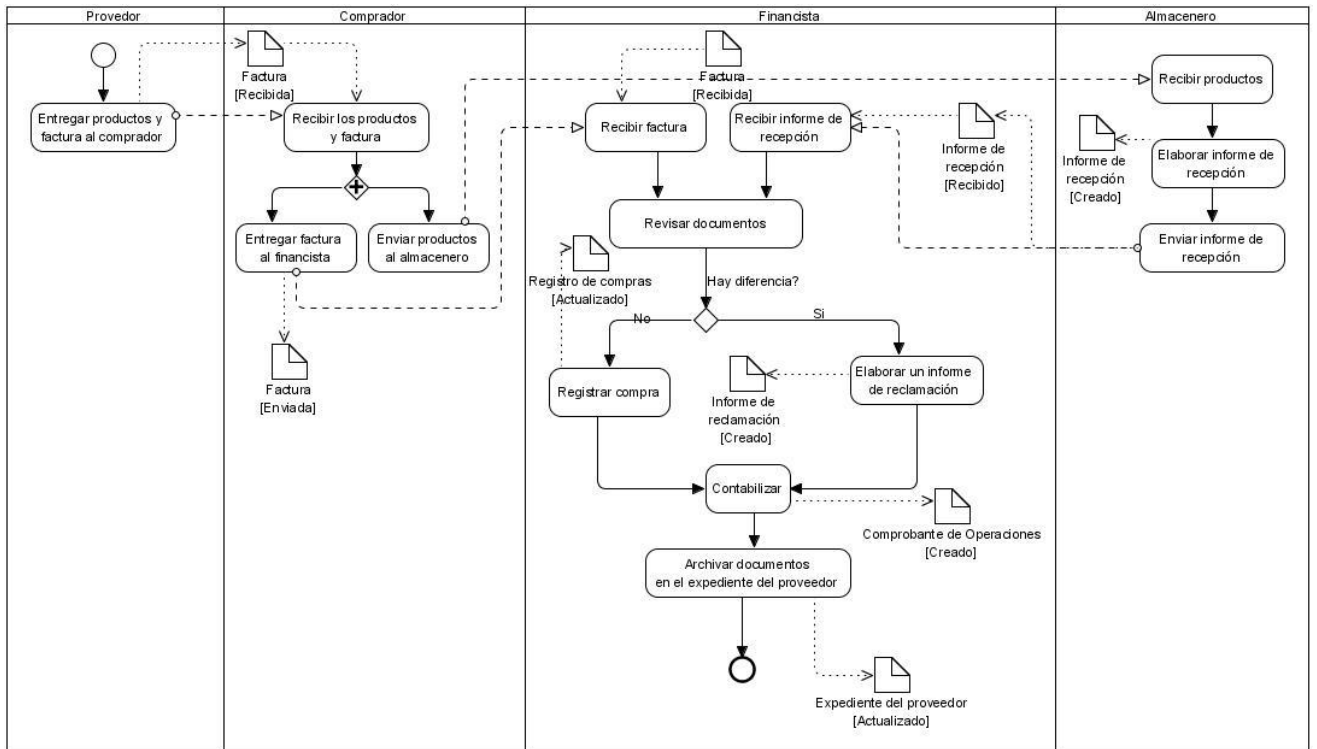


Figura 15. Recibir productos.

Para obtener mayor información sobre la descripción de este proceso, consultar el documento de descripción de procesos realizado por los analistas del módulo Cobros y Pagos. (Analistas, 2009)

2.2.3. Modelo conceptual.

El modelo de dominio o modelo conceptual es una representación visual de los conceptos u objetos del mundo real significativos para un problema o área de interés. Representa clases conceptuales del dominio del problema, conceptos del mundo real, no de los componentes del software. Este modelo se representa como un diagrama de clases en el que se muestran: (2009)

- Conceptos u objetos del dominio del problema: clases conceptuales.
- Asociaciones entre las clases conceptuales.
- Atributos de las clases conceptuales.

En este modelo no se muestra comportamiento, o sea las clases conceptuales que se definen pueden tener atributos pero no métodos. Para lograr una excelente realización de dicho modelo es necesario seguir la siguiente guía:

- Listar conceptos.
- Representar los conceptos en un diagrama.
- Agregar las asociaciones para registrar las relaciones entre conceptos.
- Agregar los atributos necesarios para cumplir los requerimientos de información.

En el módulo Cobros y Pagos se desarrolló un modelo conceptual general que abarca cada uno de los procesos generales definidos por los analistas. También se definieron modelos conceptuales para cada uno de estos procesos generales:

- Gestión de derechos de cobro.
- Gestión de obligaciones de pago.
- Gestión de derechos fiscales.
- Gestión de obligaciones fiscales.
- Configuración.

Cada uno de estos modelos cuenta con sus clases conceptuales correspondientes, los atributos que las identifican y las relaciones que se establecen entre las mismas.

Modelo conceptual general.

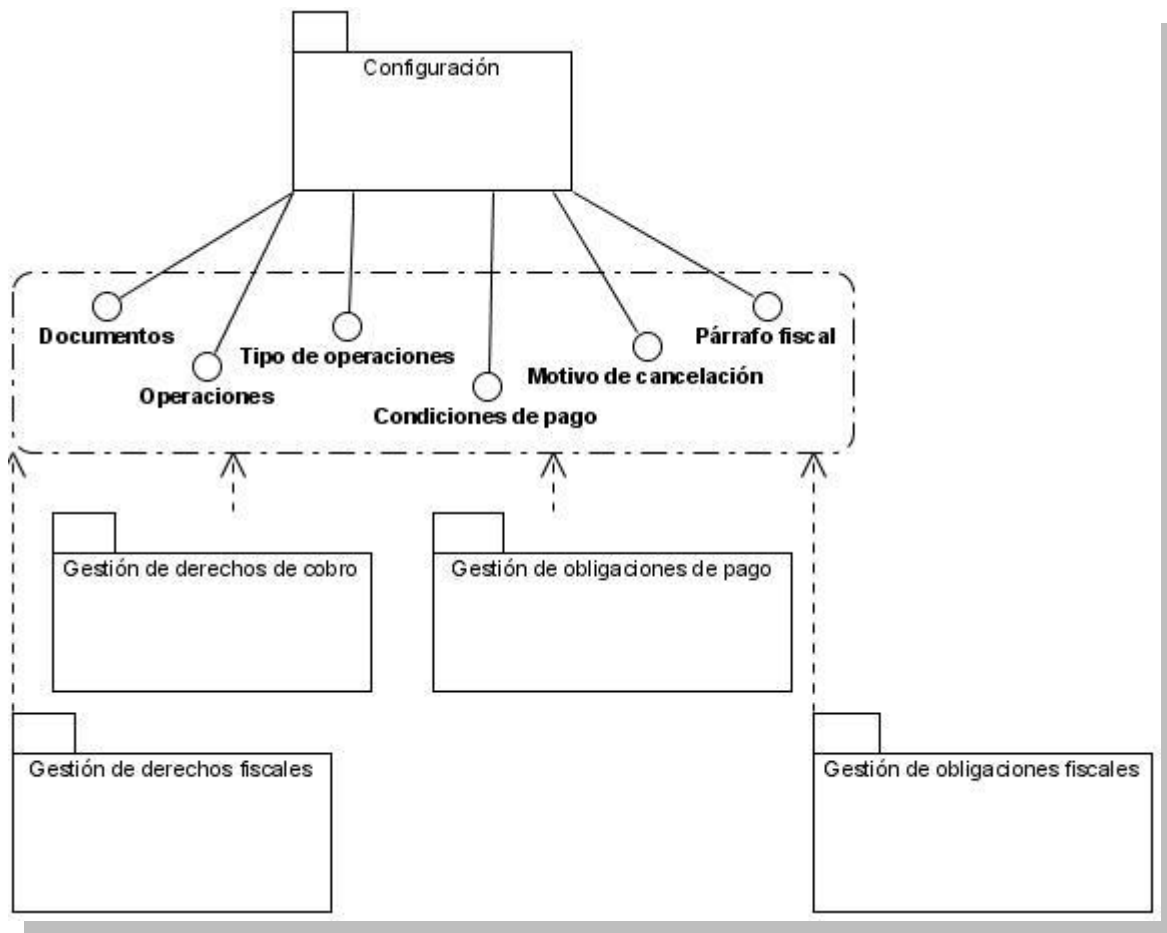


Figura 16. Modelo conceptual general del módulo Cobros y Pagos.

Modelo conceptual Gestión de derechos de cobro.

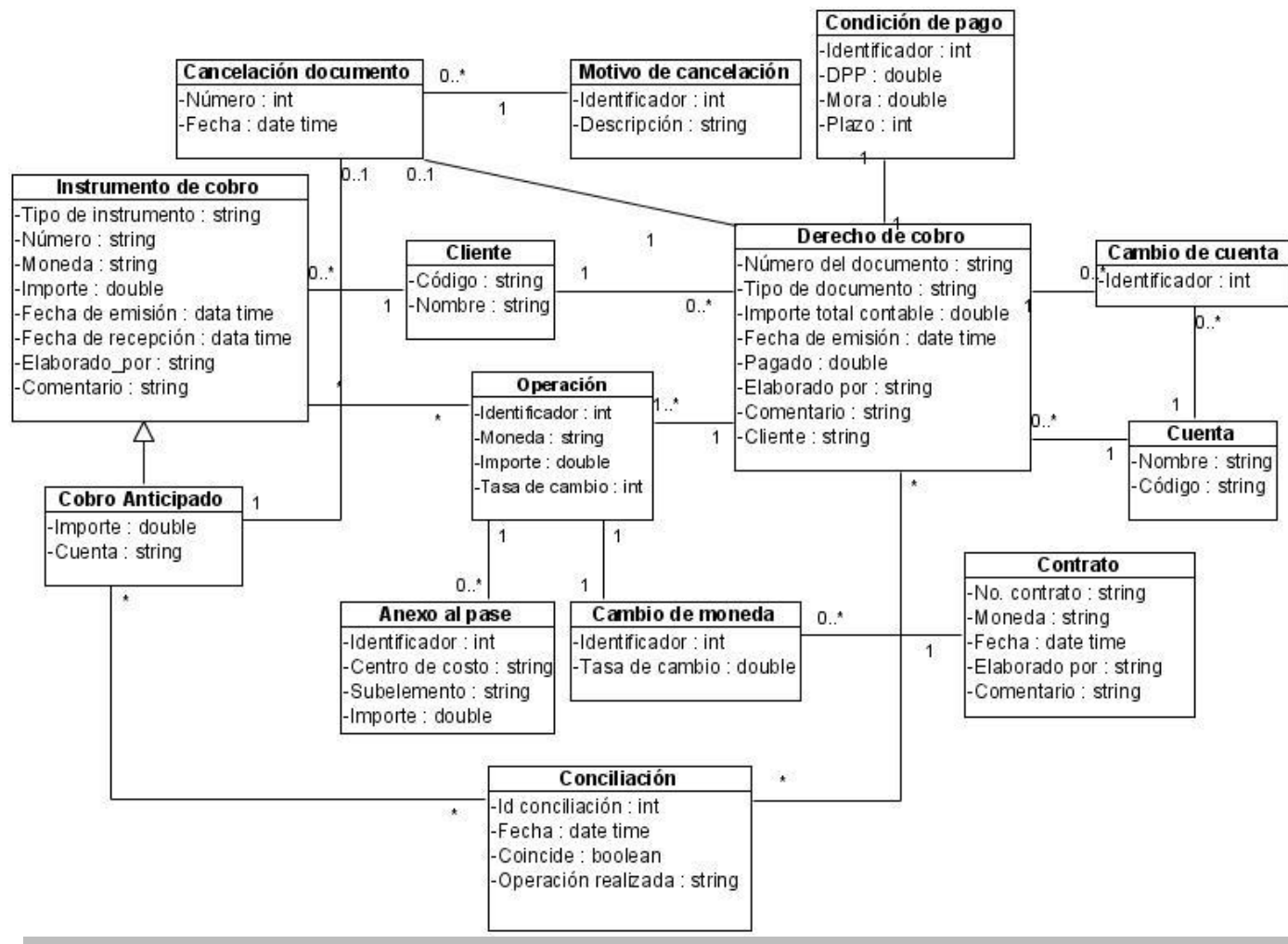


Figura 17. Modelo conceptual: Gestión de derechos de cobro.

Modelo conceptual Gestión de obligaciones de pago.

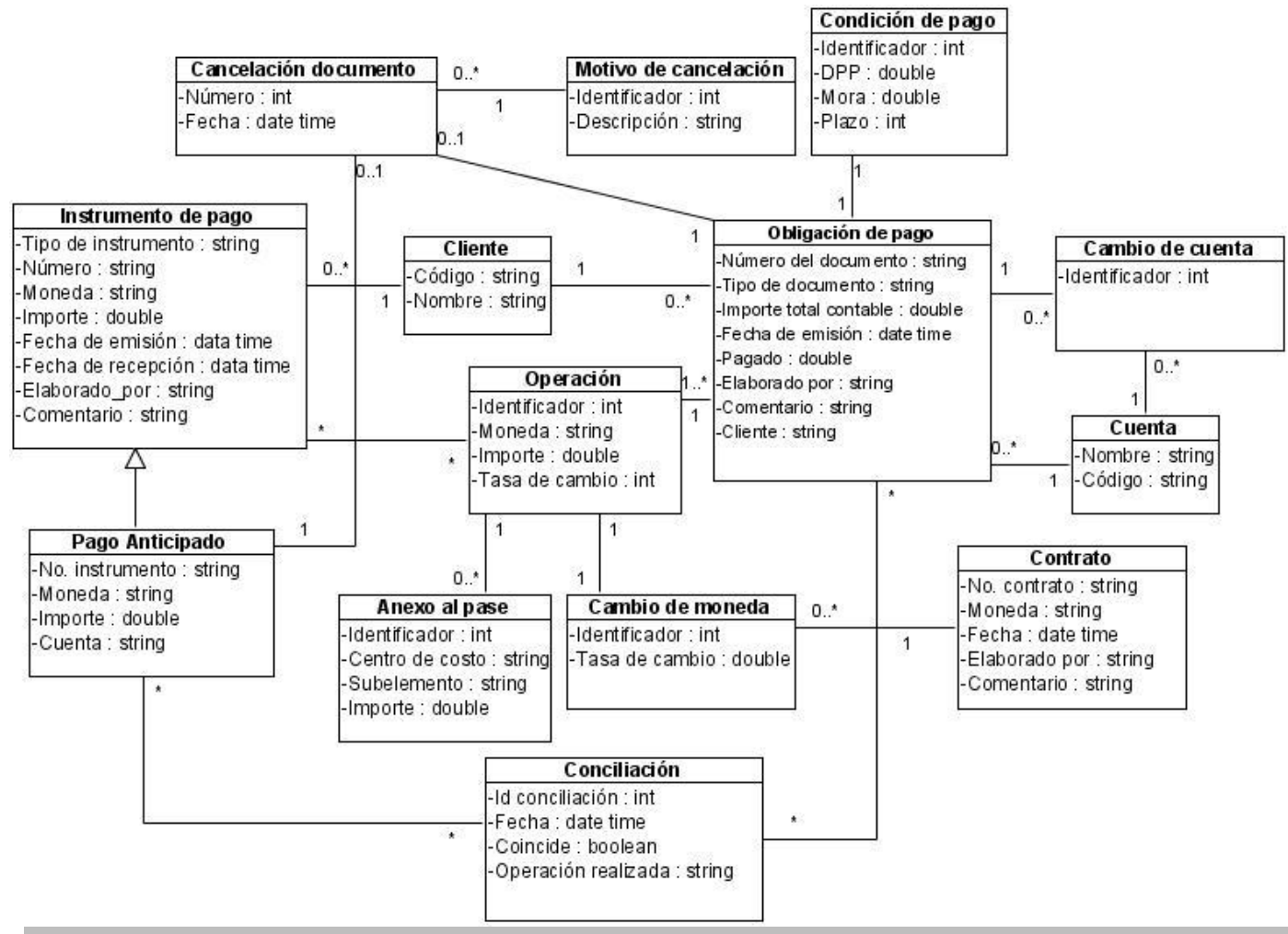


Figura 18. Modelo conceptual: Gestión de obligaciones de pago.

Modelo conceptual Gestión de derechos fiscales.

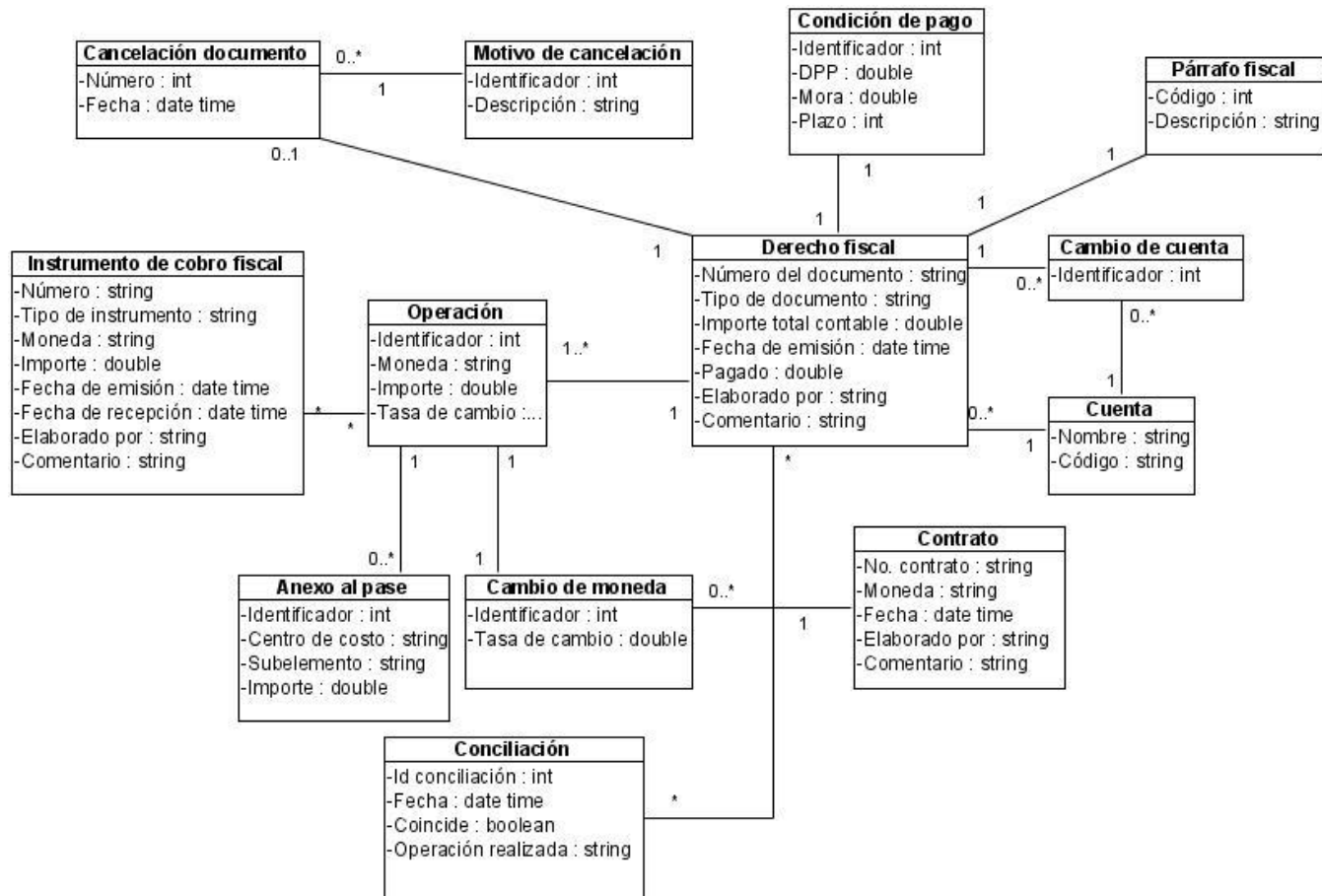


Figura 19. Modelo conceptual: Gestión de derechos fiscales.

Modelo conceptual Gestión de obligaciones fiscales.

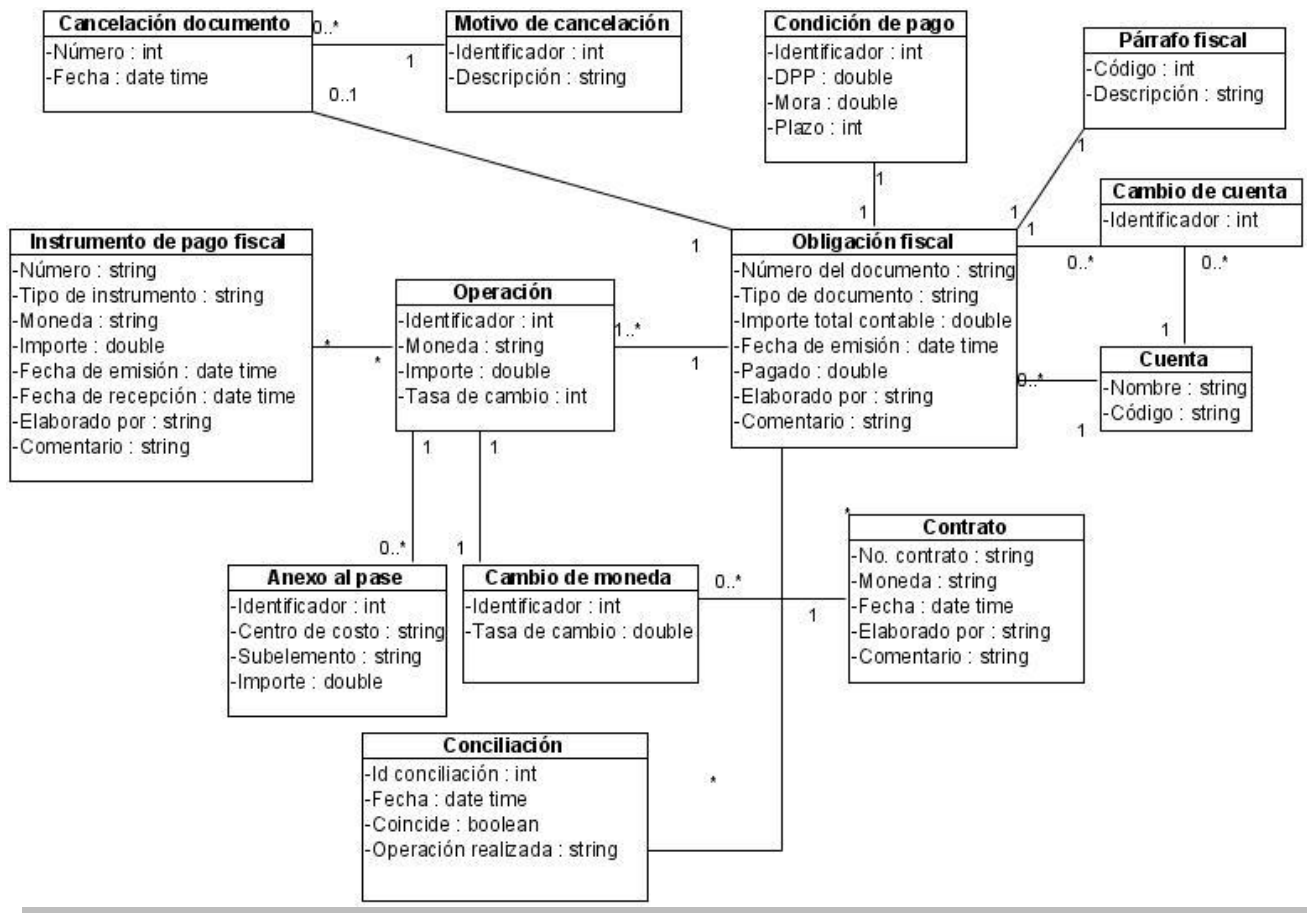


Figura 20. Modelo conceptual: Gestión de obligaciones fiscales.

Modelo Conceptual Configuración.

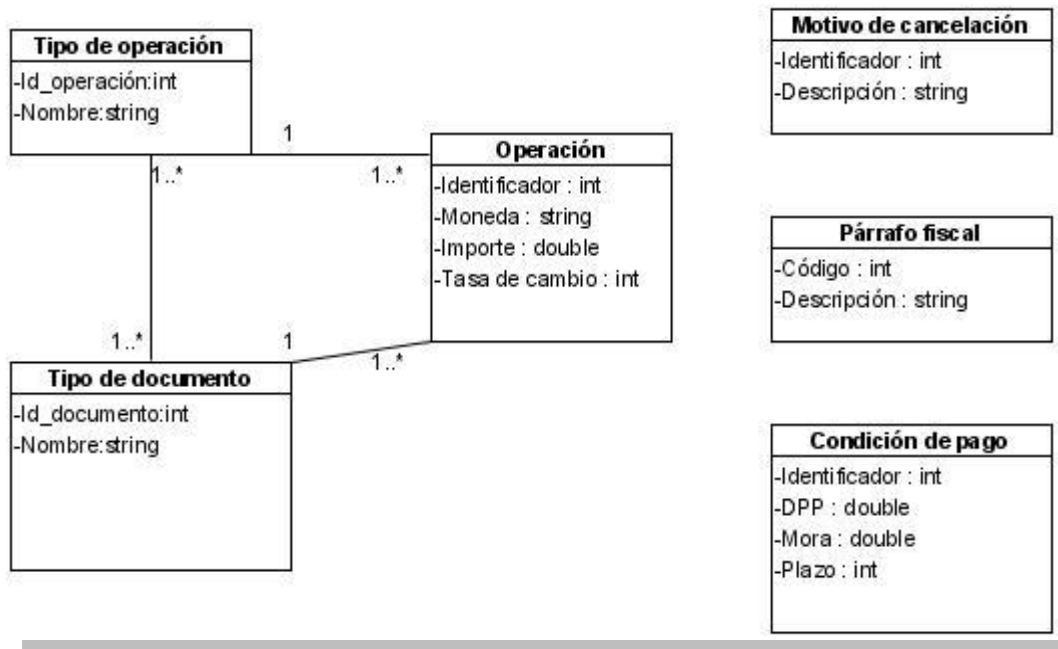


Figura 21. Modelo conceptual: Configuración.

Cada una de las entidades o clases conceptuales definidas en estos modelos tienen un diccionario de datos donde se expresa una breve descripción sobre las mismas y se describen características de sus atributos tales como: tipo de dato, si puede tomar valores nulos, si es único y las restricciones que se le imponen. (Analistas)

2.3. Requerimientos.

La IEEE⁴ define como requerimiento: (Arias, 2006)

- Condición o capacidad que necesita un usuario para resolver un problema o lograr un objetivo.
- Condición o capacidad que tiene que ser alcanzada o poseída por un sistema o componente de un sistema para satisfacer un contrato, estándar, u otro documento impuesto formalmente.

⁴ Glosario estándar de terminología de Ingeniería de Software del inglés Standard Glossary of Software Engineering Terminology.

La etapa de definición de requerimientos tiene gran importancia para el proceso de desarrollo de un software ya que mediante esta actividad el equipo de desarrollo de un sistema de software extrae las necesidades que debe cumplir dicho sistema, este proceso puede resultar complejo, principalmente si el entorno de trabajo es desconocido para el equipo de analistas y depende mucho de las personas que participen en él.

Los requerimientos se clasifican en: (Arias, 2006)

- Requerimientos funcionales: son capacidades o condiciones que el sistema debe cumplir.
- Requerimientos no funcionales: son cualidades o propiedades que el producto debe tener, las cuales hacen al producto atractivo, usable, rápido o confiable.

Los requerimientos deben cumplir las siguientes características: (Arias, 2006)

- Especificados por escrito. Como todo contrato o acuerdo entre dos partes.
- Posibles de probar o verificar. Si un requerimiento no se puede comprobar, entonces ¿cómo saber si se cumple con él o no?
- Descritos como una característica del sistema a entregar, o sea deben decir qué es lo que el sistema debe hacer, no cómo hacerlo.
- Lo más general y conciso posible. Para evitar malas interpretaciones.

Técnicas utilizadas para la captura de requisitos del módulo Cobros y Pagos.

La identificación de requisitos no es un proceso fácil, debido a que los datos son extraídos de personas y en ocasiones no fluye la comunicación entre las partes involucradas para lograr identificarlos, se han desarrollado técnicas con el objetivo de hacer este proceso de una forma más eficiente y segura.

Los analistas del módulo Cobros y Pagos emplearon para la identificación de los requisitos la técnica de las entrevistas y dentro de esta las siguientes:_(2009)

- Open ended interview: Estas entrevistas son del tipo que realizan los psicólogos. La idea es que los analistas permitan que el cliente le vaya platicando su problemática y ellos irlos guiando a través de la plática para ir determinando los requisitos del sistema.

- Entrevistas en grupos de desarrollo: Con este tipo de entrevistas se forman grupos específicos con el personal del cliente. Estos grupos tendrán en común algún área de trabajo o especialidad. El objetivo es poder contar con los expertos en cierta área de la empresa para poder llegar en conjunto a la especificación de requisitos.
- Discusiones: Este tipo de entrevistas pretende que los analistas sostengan una discusión con el cliente sobre su problemática para tratar de determinar en conjunto los requisitos del sistema.

En el módulo Cobros y Pagos se definieron los siguientes requisitos:

Requisitos Funcionales:

- Documento: Especificación de requisitos (Gestionar derechos de cobro y fiscales).

R1. Gestionar derecho de cobro.

- R1.1. Adicionar derecho de cobro.
- R1.2. Modificar derecho de cobro.
- R1.3. Cancelar derecho de cobro.
- R1.4. Liquidar derecho de cobro.
- R1.5. Confirmar derecho de cobro.

R2. Gestionar derechos fiscales.

- R2.1. Adicionar derecho fiscal.
- R2.2. Modificar un derecho fiscal.
- R2.3. Cancelar derecho fiscal.
- R2.4. Liquidar derecho fiscal.
- R2.5. Confirmar derecho fiscal.

R3. Cambiar de cuenta o efecto por cobrar a litigio y a proceso judicial.

R4. Adicionar instrumento de cobro.

R5. Adicionar instrumento de cobro para derechos fiscales.

R6. Gestionar cobros anticipados.

- R6.1. Adicionar un cobro anticipado.
- R6.2. Modificar un cobro anticipado.
- R6.3. Cancelar un cobro anticipado.
- R6.4. Liquidar un cobro anticipado.
- R6.5. Confirmar un cobro anticipado.

R7. Gestionar Conciliaciones.

- R7.1. Adicionar conciliación de derechos de cobro.
- R7.2. Adicionar conciliación de cobro anticipado.

➤ Documento: Especificación de requisitos (Gestionar obligaciones de pago y fiscales).

R1. Gestionar obligaciones de pago.

- R1.1. Adicionar obligación de pago.
- R1.2. Modificar obligación de pago.
- R1.3. Cancelar una obligación de pago.
- R1.4. Liquidar obligación de pago.
- R1.5. Confirmar obligación de pago.

R2. Gestionar obligaciones fiscales.

- R2.1. Adicionar obligación fiscal.
- R2.2. Modificar obligación fiscal.
- R2.3. Cancelar obligación fiscal.
- R2.4. Liquidar obligación fiscal.
- R2.5. Confirmar obligación fiscal.

R3. Adicionar instrumento de pago.

R4. Adicionar instrumento de pago para obligaciones fiscales.

R5. Gestionar pagos anticipados

R5.1. Adicionar un pago anticipado.

R5.2. Modificar un pago anticipado.

R5.3. Cancelar un pago anticipado.

R5.4. Liquidar un pago anticipado.

R5.5. Confirmar un pago anticipado.

R6. Gestionar conciliaciones.

R6.1. Adicionar conciliación de obligaciones de pago.

R6.2. Adicionar conciliación de pago anticipado.

➤ Documento: Especificación de requisitos comunes del módulo.

R1. Gestionar operaciones asociadas.

R1.1. Adicionar operación asociada.

R1.2. Modificar operación asociada.

R1.3. Eliminar operación asociada.

R2. Gestionar operaciones asociadas para carga inicial.

R2.1. Adicionar operación asociada para carga inicial.

R2.2. Modificar operación asociada para carga inicial.

R2.3. Eliminar operación asociada para carga inicial.

R3. Contabilizar operaciones.

R4. Generar comprobante de operaciones según la configuración del comprobante tipo.

R5. Carga inicial.

R5.1. Capturar obligaciones de pago de carga inicial.

R5.1. Modificar obligaciones de pago de carga inicial.

R5.2. Eliminar obligaciones de pago de carga inicial.

R5.3. Capturar obligaciones fiscales de carga inicial.

R5.4. Modificar obligaciones fiscales de carga inicial.

R5.5. Eliminar obligaciones fiscales de carga inicial.

R5.6. Capturar derechos de cobro de carga inicial.

R5.7. Modificar derechos de cobro de carga inicial.

R5.8. Eliminar derechos de cobro de carga inicial.

R5.9. Capturar derechos fiscales de carga inicial.

R5.10. Modificar derechos fiscales de carga inicial.

R5.11. Eliminar derechos fiscales de carga inicial.

R5.12. Capturar cobros anticipados de carga inicial.

R5.13. Modificar cobros anticipados de carga inicial.

R5.14. Eliminar cobros anticipados de carga inicial.

R5.15. Capturar pagos anticipados de carga inicial.

R5.16. Modificar pagos anticipados de carga inicial.

R5.17. Eliminar pagos anticipados de carga inicial.

R6. Gestionar contrato.

R6.1. Adicionar contrato.

R6.2. Modificar contrato.

R6.3. Cancelar contrato.

R7. Cambio de moneda original.

R8. Gestionar registro anexo al pase.

R8.1. Adicionar registro anexo al pase.

R8.2. Modificar registro anexo al pase.

R8.3. Eliminar registro anexo al pase.

R9. Realizar cierre.

➤ Documento: Especificación de requisitos de soporte.

R1. Recuperaciones.

R1.1. Mostrar importe de los derechos de cobro pendientes.

R1.2. Mostrar el submayor de todos los clientes.

R1.3. Mostrar el submayor de un cliente determinado.

R1.4. Mostrar el submayor de todos los proveedores.

R1.5. Mostrar el submayor de un proveedor determinado.

R1.6. Mostrar listado de instrumentos de cobro recibidos.

R1.7. Mostrar listado de instrumentos de cobro recibidos para derechos fiscales.

R1.8. Mostrar listado de cobros.

R1.9. Mostrar listado de cheques recibidos.

R1.10. Mostrar el importe total de las obligaciones de pago pendientes.

R1.11. Mostrar listado de los instrumentos de pago emitidos.

R1.12. Mostrar listado de los instrumentos de pago emitidos para obligaciones fiscales.

R1.13. Mostrar listado de pagos.

R1.14. Mostrar listado de cheques emitidos.

R1.15. Reportar las cuentas por pagar en pesos cubanos vencidas por más de seis meses.

R1.16. Imprimir conciliación de obligaciones de pago.

R1.17. Imprimir conciliación de derechos de cobro.

R1.18. Imprimir conciliación de obligaciones fiscales.

R1.19. Imprimir conciliación de derechos fiscales.

R1.20. Imprimir conciliación de cobros anticipados.

R1.21. Imprimir conciliación de pagos anticipados.

R1.22. Mostrar análisis de la antigüedad de los deudores.

R1.23. Mostrar análisis de la antigüedad de los acreedores.

R1.24. Mostrar análisis por edades de acreedores.

R1.25. Mostrar análisis por edades de deudores.

R1.26. Mostrar listado de los derechos de cobro.

R1.27. Mostrar listado de las obligaciones de pago.

➤ Documento: Especificación de requisitos (Configuración).

R1. Gestionar motivos de cancelación.

R1.1. Adicionar motivo de cancelación.

R1.2. Modificar motivo de cancelación.

R1.3. Eliminar motivo de cancelación.

R2. Gestionar párrafos fiscales.

R2.1. Adicionar párrafo fiscal

R2.2. Modificar párrafo fiscal.

R2.3. Eliminar párrafo fiscal.

R3. Adicionar condición de pago por defecto.

R4. Modificar condición de pago por defecto.

Cada uno de los requisitos mencionados anteriormente cuenta con su especificación correspondiente, donde se explica de forma detallada su descripción, así como las precondiciones, validaciones y post-

condiciones asociadas al mismo. En el módulo Cobros y Pagos se realizaron cinco documentos para la especificación de los requisitos funcionales: (Analistas, 2009)

- Documento: Especificación de requisitos Gestionar derecho de cobro y fiscales.
- Documento: Especificación de requisitos de Configuración. (Analistas, 2009)
- Documento: Especificación de requisitos Gestionar obligaciones de pago y fiscales.
- Documento: Especificación de requisitos comunes del módulo.
- Documento: Especificación de requisitos de soporte.

Los cinco documentos de especificación de requisitos realizados por los analistas del módulo Cobros y Pagos constituyen artefactos entregables del módulo y por ende del presente trabajo de diploma, por cuanto se propone no plasmar en el mismo las especificaciones de los requisitos identificados.

2.4. Prototipos de interfaz de usuario.

La creación de las interfaces de usuario ha sido un área en el desarrollo de software que ha evolucionado aceleradamente a partir de la década de los setenta. La interfaz de usuario es el vínculo entre el usuario y el programa de computadora. Una interfaz es un conjunto de comandos o menús a través de los cuales el usuario se comunica con el programa. (2006)

Esta es una de las partes más importantes de cualquier programa ya que determina que tan fácilmente es posible que el programa haga lo que el usuario quiere hacer. Un programa muy poderoso con una interfaz pobremente elaborada tiene poco valor para un usuario no experto.

La elaboración de una interfaz de usuario, bien diseñada, exige una gran dedicación pues generalmente las interfaces son grandes, complejas y difíciles de implementar, depurar y modificar. Hoy en día las interfaces de manipulación directa (también llamadas interfaces gráficas de usuario, GUI⁵) son prácticamente universales. Las interfaces que utilizan ventanas, íconos y menús se han convertido en estándar de los materiales computacionales. (2006)

⁵ Interfaz de entorno gráfico del inglés Graphic User Interface.

Prototipos de interfaz de usuario del módulo Cobros y Pagos:

Prototipo de interfaz de usuario: Gestionar derecho de cobro. (Analistas, 2008)

- Requisito funcional R1.5.
- Requisito funcional R3.

El prototipo de interfaz de usuario muestra una ventana con el título "Gestionar derecho de cobro". En la parte superior, hay una barra de herramientas con los botones "Adicionar", "Modificar", "Cancelar", "Confirmar" y "Contabilizar". Debajo de los botones, hay un formulario de búsqueda con los campos "Número de documento" (campo de texto), "Fecha" (campo de fecha con un icono de calendario), "Tipo de documento" (campo de lista desplegable) y "Estado" (campo de lista desplegable). En la parte inferior, hay una tabla con las siguientes columnas: "Número", "Fecha", "Tipo documento", "Estado" y "Comentario".

Figura 22. Gestionar derecho de cobro.

A través de esta interfaz el usuario puede realizar las siguientes funciones:

- Adicionar.
- Modificar.
- Cancelar.
- Confirmar.
- Contabilizar.

El usuario puede hacer su búsqueda más específica mediante un filtro con los siguientes datos (Número de documento, Fecha, Tipo de documento y Estado).

Para confirmar un derecho de cobro se debe seleccionar el derecho que se desea confirmar. El derecho seleccionado debe estar en estado "guardado".

Para contabilizar un derecho de cobro se debe seleccionar el derecho que se desea contabilizar. El derecho seleccionado debe estar en estado "confirmado". Se genera el asiento contable correspondiente al derecho elegido.

Prototipo de interfaz de usuario: Adicionar derecho de cobro. (Analistas, 2008)

- Requisito funcional R1.1.
- Requisito funcional R1.5.
- Requisito funcional R3.

Operación	Moneda	Importe	Tasa	Importe original
-----------	--------	---------	------	------------------

Figura 23. Adicionar derecho de cobro.

Mediante esta interfaz el usuario puede registrar un nuevo derecho de cobro, además puede confirmar o contabilizar el derecho una vez guardado.

Para confirmar el derecho de cobro mediante esta interfaz se debe haber guardado el derecho.

Para contabilizar el derecho de cobro mediante esta interfaz se debe haber guardado y confirmado el derecho.

Prototipo de interfaz de usuario: Modificar derecho de cobro. (Analistas, 2008)

- Requisito funcional R1.2.
- Requisito funcional R1.5.
- Requisito funcional R3.

El prototipo de interfaz de usuario para 'Modificar derecho de cobro' presenta un formulario con los siguientes campos y elementos:

- Campos de entrada:** Tipo documento (menú desplegable), Número de documento (campo de texto), Cliente (menú desplegable), Cuenta (menú desplegable), Fecha de emisión (menú desplegable), Plazo(Días) (campo de texto), D.P.P.(%) (campo de texto), Mora(%) (campo de texto).
- Operación:** Menú desplegable, Moneda (menú desplegable), Importe (campo de texto), Tasa (menú desplegable).
- Botones de acción:** +, M, -, X.
- Tabla de datos:** Una tabla con 5 columnas: Operación, Moneda, Importe, Tasa, Importe original. El cuerpo de la tabla está actualmente vacío.
- Resumen:** Importe total contable (campo de texto) y Pagado (campo de texto).
- Comentarios:** Campo de texto etiquetado como 'Comentario:'.
- Botones de control:** Guardar, Confirmar, Contabilizar, Cerrar.

Figura 24. Modificar derecho de cobro.

Para modificar un derecho de cobro primero se debe seleccionar el derecho que se desea modificar en la interfaz Gestionar derecho de cobro, además se debe verificar el estado del documento seleccionado, en caso de que sea “confirmado”, no se puede modificar.

Prototipo de interfaz de usuario: Cancelar derecho de cobro. (Analistas, 2008)

- Requisito funcional R1.3.

<input type="checkbox"/>	Tipo documento	No. documento	Cliente	Fecha
--------------------------	----------------	---------------	---------	-------

Figura 25. Cancelar derecho de cobro.

Para cancelar un derecho de cobro el usuario puede hacer la búsqueda más efectiva mediante un filtro con los siguientes datos (Cliente, No. documento). No se pueden cancelar aquellos derechos que hayan sido liquidados parcial o totalmente.

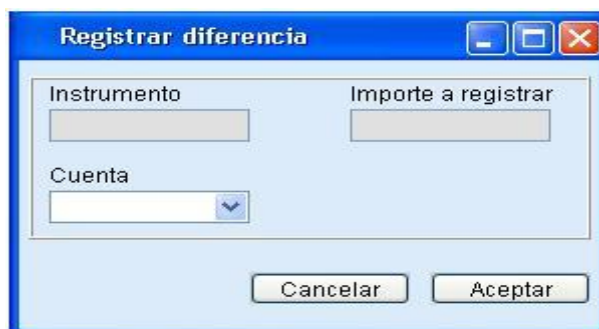
Prototipo de interfaz de usuario: Liquidar derecho de cobro. (Analistas, 2008)

- Requisito funcional R1.4.

Figura 26. Liquidar derecho de cobro.

Para liquidar un derecho de cobro se debe haber registrado un instrumento de cobro mediante la interfaz Registrar instrumento de cobro. Se puede liquidar parcial o totalmente un derecho de cobro. Solo se liquidan aquellos derechos del cliente seleccionado en el instrumento de cobro y cuyo importe esté en la moneda especificada en el instrumento.

Cuando se realiza la liquidación de los derechos seleccionados si queda importe disponible en el instrumento de cobro, el usuario debe registrar el mismo en una cuenta determinada, mediante la siguiente interfaz:

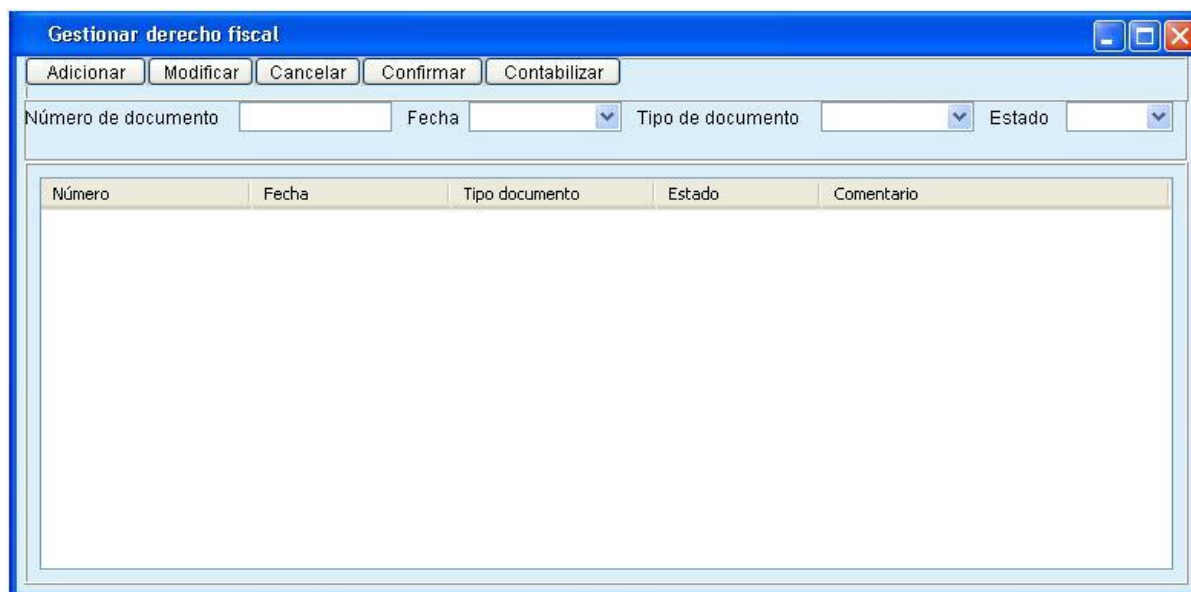


The image shows a Windows-style dialog box titled "Registrar diferencia". It contains three input fields: "Instrumento" (text), "Importe a registrar" (text), and "Cuenta" (dropdown menu). At the bottom, there are two buttons: "Cancelar" and "Aceptar".

Figura 27. Registrar diferencia.

Prototipo de interfaz de usuario: Gestionar derechos fiscales. (Analistas, 2008)

- Requisito funcional R2.5.
- Requisito funcional R3.



The image shows a Windows-style application window titled "Gestionar derecho fiscal". It features a toolbar with buttons for "Adicionar", "Modificar", "Cancelar", "Confirmar", and "Contabilizar". Below the toolbar are four input fields: "Número de documento" (text), "Fecha" (dropdown), "Tipo de documento" (dropdown), and "Estado" (dropdown). At the bottom, there is a table with the following columns: "Número", "Fecha", "Tipo documento", "Estado", and "Comentario". The table is currently empty.

Figura 28. Gestionar derechos fiscales.

A través de esta interfaz el usuario puede realizar las siguientes funciones:

- Adicionar.
- Modificar.
- Cancelar.
- Confirmar.
- Contabilizar.

El usuario puede hacer su búsqueda más específica mediante un filtro con los siguientes datos (Número de documento, fecha, tipo de documento y estado).

Para confirmar un derecho fiscal se debe seleccionar el derecho que se desea confirmar. El derecho seleccionado debe estar en estado "guardado".

Para contabilizar un derecho fiscal se debe seleccionar el derecho que se desea contabilizar. El derecho seleccionado debe estar en estado "confirmado". Se genera el asiento contable correspondiente al derecho elegido.

Prototipo de interfaz de usuario:Adicionar derecho fiscal. (Analistas, 2008)

- Requisito funcional R2.1.
- Requisito funcional R2.5.
- Requisito funcional R3.

The screenshot shows a software window titled "Adicionar derecho fiscal". It contains the following elements:

- Fields for "Tipo documento:", "Número de documento:", "Párrafo fiscal:", "Plazo(Días):", "Cuenta:", "Fecha de emisión:", "D.P.P(%):", and "Mora(%):".
- Fields for "Operación:", "Moneda:", "Importe:", and "Tasa:", along with buttons "+", "M", "-", and "X".
- A table with columns: "Operación", "Moneda", "Importe", "Tasa", and "Importe original".
- Fields for "Importe total contable:" and "Pagado:".
- A "Comentario:" text area.
- Buttons: "Guardar", "Confirmar", "Contabilizar", and "Cerrar".

Figura 29. Adicionar derecho fiscal.

Mediante esta interfaz el usuario puede registrar un nuevo derecho fiscal, además puede confirmar o contabilizar el derecho una vez guardado.

Para confirmar el derecho fiscal mediante esta interfaz se debe haber guardado el derecho.

Para contabilizar el derecho fiscal mediante esta interfaz se debe haber guardado y confirmado el derecho.

Prototipo de interfaz de usuario: Modificar derecho fiscal. (Analistas, 2008)

- Requisito funcional R2.2.
- Requisito funcional R2.5.
- Requisito funcional R3.

The screenshot shows a software window titled "Modificar derecho fiscal". The interface includes the following elements:

- Top Section:** Fields for "Tipo documento" (dropdown), "Número de documento" (text), "Párrafo fiscal" (dropdown), "Plazo(Días)" (text), "Cuenta" (dropdown), "Fecha de emisión" (dropdown), "D.P.P(%)" (text), and "Mora(%)" (text).
- Second Section:** Fields for "Operación" (dropdown), "Moneda" (dropdown), "Importe" (text), and "Tasa" (dropdown), accompanied by "+", "M", "-", and "X" buttons.
- Table:** A table with columns: "Operación", "Moneda", "Importe", "Tasa", and "Importe original".
- Bottom Section:** Fields for "Importe total contable" and "Pagado", a "Comen" field, and buttons for "Guardar", "Confirmar", "Contabilizar", and "Cerrar".

Figura 30. Modificar derecho fiscal.

Para modificar un derecho fiscal primero se debe seleccionar el derecho que se desea modificar en la interfaz Gestionar derecho fiscal, además se debe verificar el estado del documento seleccionado, en caso de que sea “confirmado”, no se puede modificar.

Prototipo de interfaz de usuario: Cancelar derecho fiscal. (Analistas, 2008)

- Requisito funcional R2.3.

The image shows a software dialog box titled "Cancelar derecho de cobro fiscal". The dialog has a light blue background and a blue title bar with standard window controls (minimize, maximize, close). The main area contains a search form with the following elements:

- A text input field labeled "No. documento:" followed by a "Filtrar" button.
- A dropdown menu labeled "Motivo de cancelación:".
- A dropdown menu labeled "Fecha:".

Below the search form is a table with the following columns:

<input type="checkbox"/>	Tipo de documento	No. documento	Fecha

At the bottom of the dialog are two buttons: "Aceptar" and "Cerrar".

Figura 31. Cancelar derecho fiscal.

Para cancelar un derecho fiscal el usuario puede hacer la búsqueda más efectiva mediante un filtro con los siguientes datos (No. documento). No se pueden cancelar aquellos derechos que hayan sido liquidados parcial o totalmente.

Prototipo de interfaz de usuario: Liquidar derecho fiscal. (Analistas, 2008)

- Requisito funcional R2.4.

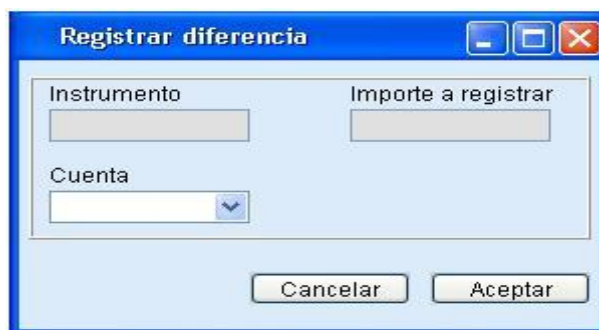
Instrumento de cobro		
No.instrumento:	Importe:	Importe total a cobrar
12	20.00	10.00

Derechos fiscales				
No. documento	Moneda	Importe pendiente	Operación	Importe a cobrar

Figura 32. Liquidar derecho fiscal.

Para liquidar un derecho fiscal se debe haber registrado un instrumento de cobro para derechos fiscales mediante la interfaz Registrar instrumento de cobro (operaciones fiscales). Se puede liquidar parcial o totalmente un derecho fiscal.

Cuando se realiza la liquidación de los derechos seleccionados si queda importe disponible en el instrumento de cobro, el usuario debe registrar el mismo en una cuenta determinada, mediante la siguiente interfaz:

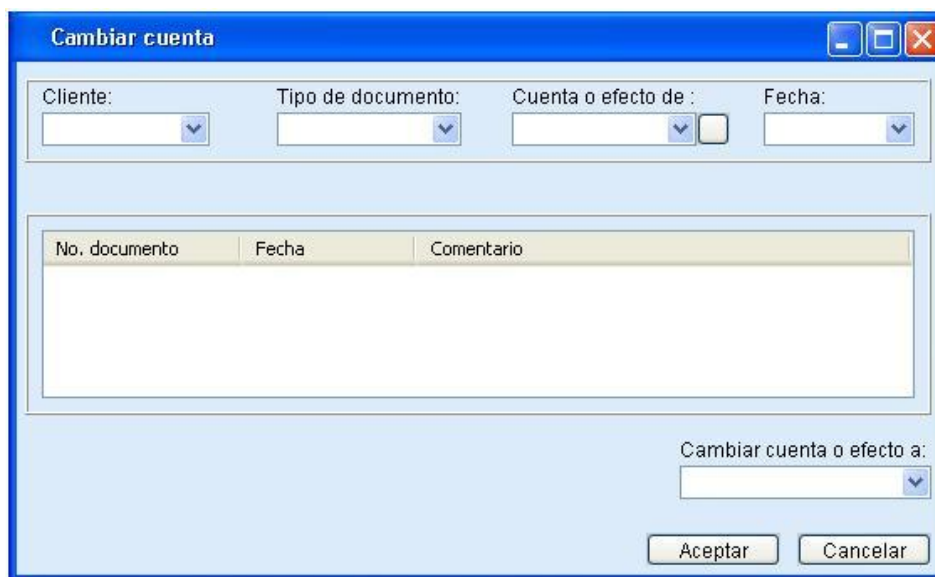


The image shows a Windows-style dialog box titled "Registrar diferencia". It has a blue title bar with standard window controls (minimize, maximize, close). The main area is light blue and contains two text input fields: "Instrumento" and "Importe a registrar". Below these is a dropdown menu labeled "Cuenta". At the bottom of the dialog are two buttons: "Cancelar" and "Aceptar".

Figura 33. Registrar diferencia.

Prototipo de interfaz de usuario: Cambiar de cuenta. (Analistas, 2008)

- Requisito funcional R3.
- Requisito funcional R3.



The image shows a Windows-style dialog box titled "Cambiar cuenta". It has a blue title bar with standard window controls. The main area is light blue and contains four dropdown menus: "Cliente:", "Tipo de documento:", "Cuenta o efecto de:", and "Fecha:". Below these is a table with columns "No. documento", "Fecha", and "Comentario". At the bottom right is a dropdown menu labeled "Cambiar cuenta o efecto a:" and two buttons: "Aceptar" and "Cancelar".

Figura 34. Cambiar de cuenta o efecto por cobrar a litigio y a proceso judicial.

El cambio de cuenta se le hace a los derechos u obligaciones que están pendientes a liquidar, se puede cambiar de cuenta o efecto a litigio y de cuenta o efecto en litigio a proceso judicial.

Prototipo de interfaz de usuario: Registrar instrumento de cobro.

- Requisito funcional R4. (Analistas, 2008)



El prototipo de interfaz de usuario, titulado "Registrar instrumento de cobro", presenta un diseño con un fondo azul claro y un borde azul. En la parte superior, hay un título y tres botones de control de ventana (minimizar, maximizar y cerrar). El formulario principal está dividido en varias secciones: "Cliente:" con un menú desplegable; "Tipo de instrumento:" con un menú desplegable; "Número:" con un campo de texto; "Moneda:" con un menú desplegable; "Importe:" con un campo de texto; "Fecha emisión:" con un menú desplegable; "Fecha recepción:" con un menú desplegable; "Elaborado por:" con un campo de texto; y "Comentario:" con un área de texto grande. En la parte inferior, hay tres botones: "Cancelar", "Liquidar derechos" y "Registrar cobro anticipado".

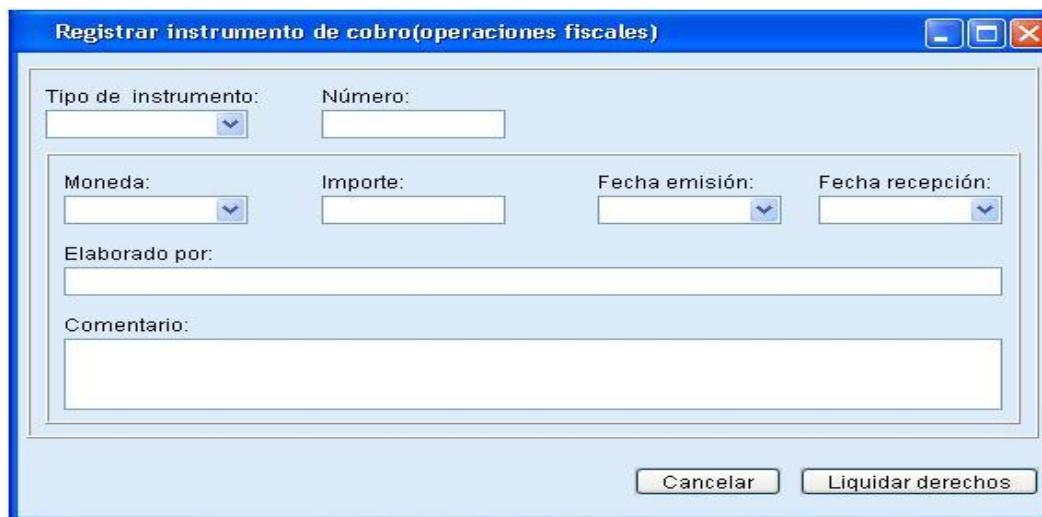
Figura 35. Adicionar instrumento de cobro.

Mediante este interfaz se puede registrar un instrumento de cobro para realizar las siguientes operaciones:

- Liquidar derechos.
- Registrar cobros anticipados.

Prototipo de interfaz de usuario: Registrar instrumento de cobro (operaciones fiscales). (Analistas, 2008)

- Requisito funcional R5.



Registrar instrumento de cobro(operaciones fiscales)

Tipo de instrumento: Número:

Moneda: Importe: Fecha emisión: Fecha recepción:

Elaborado por:

Comentario:

Cancelar Liquidar derechos

Figura 36. Registrar instrumento de cobro (operaciones fiscales).

Cuando se registra un instrumento de cobro para operaciones fiscales se pueden liquidar derechos fiscales.

Prototipo de interfaz de usuario: Gestionar cobro anticipado. (Analistas, 2008)

- Requisito funcional R6.5.
- Requisito funcional R3.

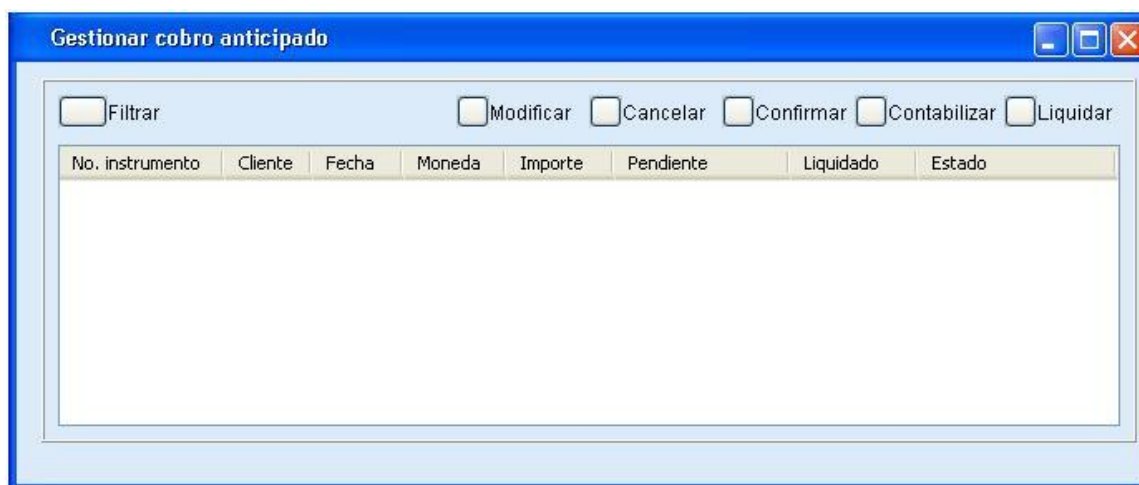


Figura 37. Gestionar cobro anticipado.

A través de esta interfaz el usuario puede realizar las siguientes funciones:

- Modificar.
- Cancelar.
- Confirmar.
- Contabilizar.

El usuario para realizar alguna de las funciones mencionadas, puede hacer su búsqueda más específica mediante el siguiente filtro:



The 'Filtrar' dialog box has a blue title bar with standard window controls. The main area is light blue and contains four input fields arranged in a 2x2 grid. The top-left field is labeled 'No. instrumento:' and is a text box. The top-right field is labeled 'Proveedor' and is a dropdown menu. The bottom-left field is labeled 'Fecha:' and is a dropdown menu. The bottom-right field is labeled 'Tipo de instrumento' and is a dropdown menu. At the bottom of the dialog are two buttons: 'Cancelar' and 'Aceptar'.

Figura 38. Filtrar para el gestionar cobro anticipado.

Para confirmar el cobro anticipado mediante esta interfaz se debe haber guardado el cobro anticipado.

Para contabilizar el cobro anticipado mediante esta interfaz se debe haber guardado y confirmado el cobro anticipado.

Prototipo de interfaz de usuario: Adicionar cobro anticipado. (Analistas, 2008)

- Requisito funcional R6.1.
- Requisito funcional R6.5.
- Requisito funcional R3.



The 'Adicionar cobro anticipado' dialog box has a blue title bar with standard window controls. The main area is light blue and contains three input fields in a row: 'No. instrumento:' (text box with value '12'), 'Moneda:' (text box with value 'CUP'), and 'Importe:' (text box with value '30.00'). Below these is a 'Cuenta:' dropdown menu. At the bottom of the dialog are four buttons: 'Guardar', 'Confirmar', 'Contabilizar', and 'Cerrar'.

Figura 39. Adicionar cobro anticipado.

Mediante esta interfaz el usuario puede registrar un nuevo cobro anticipado, además puede confirmar o contabilizar el cobro anticipado una vez guardado.

Para confirmar el cobro anticipado mediante esta interfaz se debe haber guardado el cobro anticipado.

Para contabilizar el cobro anticipado mediante esta interfaz se debe haber guardado y confirmado el cobro anticipado.

Prototipo de interfaz de usuario: Modificar cobro anticipado. (Analistas, 2008)

- Requisito funcional R6.2.
- Requisito funcional R6.5.
- Requisito funcional R3.



El prototipo de interfaz de usuario muestra una ventana con el título "Modificar cobro anticipado". Dentro de la ventana, hay tres campos de texto etiquetados "No.instrumento:", "Moneda:" y "Importe:". Debajo de estos, hay un menú desplegable etiquetado "Cuenta:". En la parte inferior de la ventana, hay cuatro botones: "Guardar", "Confirmar", "Contabilizar" y "Cerrar".

Figura 40. Modificar cobro anticipado.

Para modificar un cobro anticipado primero se debe seleccionar el cobro que se desea modificar en la interfaz Gestionar cobro anticipado, además se debe verificar el estado del documento seleccionado, en caso de que sea "confirmado", no se puede modificar.

Prototipo de interfaz de usuario: Cancelar cobro anticipado. (Analistas, 2008)

- Requisito funcional R6.3.

Tipo instrumento	No. instrumento	Cliente	Fecha
------------------	-----------------	---------	-------

Figura 41. Cancelar cobro anticipado.

Para cancelar un cobro anticipado el usuario puede hacer la búsqueda más efectiva mediante un filtro con los siguientes datos (No. Documento, Cliente). No se pueden cancelar aquellos cobros anticipados que hayan sido liquidados parcial o totalmente.

Prototipo de interfaz de usuario: Liquidar cobro anticipado. (Analistas, 2008)

- Requisito funcional R6.4.

Figura 42. Liquidar cobro anticipado.

Mediante esta interfaz se pueden liquidar varios derechos de cobro.

Cuando se realiza la liquidación de los derechos seleccionados si queda importe disponible en el instrumento de cobro, el usuario debe registrar el mismo en una cuenta determinada, mediante la siguiente interfaz:

Figura 43. Registrar diferencia.

Prototipo de interfaz de usuario: Adicionar conciliación de obligación o derecho. (Analistas, 2008)

- Requisito funcional R7.1.
- Requisito funcional R6.1.

El prototipo de interfaz de usuario muestra una ventana con el título "Adicionar conciliación obligación o derecho". En la parte superior izquierda hay un botón "Buscar" con un ícono de lupa. A la derecha de "Buscar" hay un campo de texto "Fecha" con un menú desplegable. Debajo de esto hay una tabla con las siguientes columnas: "Número", "Fecha", "Moneda", "Importe", "Coincide" y "Operación". El cuerpo de la tabla está vacío. En la parte inferior de la ventana, hay una sección "Datos" con dos campos de texto: "Número:" con el valor "1" y "Fecha:" con el valor "12/01/2009". A la derecha de "Datos" hay una sección "Coniliar" con tres opciones de radio: "Coincide", "Cambio de plazo" y "Cancelar". A la derecha de "Cambio de plazo" hay un campo de texto "Plazo". En la parte inferior derecha de la ventana hay tres botones: "Aplicar", "Cancelar" y "Aceptar".

Figura 44. Adicionar conciliación de obligación o derecho.

La funcionalidad que brinda esta interfaz es exclusiva del módulo Cobros y Pagos del CEDRUX, ya que mediante ella se pueden registrar las conciliaciones de derechos u obligaciones una vez efectuadas con la entidad correspondiente, para de esta forma tener un registro de las mismas en el sistema, cosa que no se hace en ninguna de las soluciones de software nacionales con las que cuenta Cuba actualmente.

Prototipo de interfaz de usuario: Adicionar conciliación de anticipos. (Analistas, 2008)

- Requisito funcional R7.2.
- Requisito funcional R6.2.

Adicionar conciliación de anticipo

Buscar Fecha

Número	Fecha	Moneda	Importe	Coincide	Operación
--------	-------	--------	---------	----------	-----------

Datos

Número: Fecha: Moneda: Importe:

Conciliar

Coincide Cancelar

Aplicar

Cancelar Aceptar

Figura 45. Adicionar conciliación de anticipos.

Esta interfaz brinda la misma funcionalidad de la interfaz descrita anteriormente pero en este caso para registrar las conciliaciones de cobro anticipado y pago anticipado.

El usuario puede realizar la búsqueda de los documentos más específica mediante el siguiente filtro:



The image shows a dialog box titled "Búsqueda por clientes". It contains four dropdown menus arranged in a 2x2 grid: "Cliente:", "Desde:", "Tipo de documento:", and "Hasta:". Below the dropdowns are two buttons: "Cancelar" and "Aceptar".

Figura 46. Filtro para la búsqueda por clientes.

Prototipo de interfaz de usuario: Gestionar obligaciones de pago. (Analistas, 2008)

- Requisito funcional R1.5.
- Requisito funcional R3.



The image shows a window titled "Gestionar obligación de pago". At the top, there is a menu bar with buttons: "Adicionar", "Modificar", "Cancelar", "Liquidar", "Contabilizar", and "Confirmar". Below the menu bar, there are four input fields: "Número de documento", "Fecha", "Tipo de documento", and "Estado". At the bottom, there is a table with the following columns: "Número", "Fecha", "Tipo documento", "Estado", and "Comentario".

Figura 47. Gestionar obligación de pago.

A través de esta interfaz el usuario puede realizar las siguientes funciones:

- Adicionar.
- Modificar.
- Cancelar.

- Confirmar.
- Contabilizar.

El usuario puede hacer su búsqueda más específica mediante un filtro con los siguientes datos (Número de documento, Fecha, Tipo de documento y Estado).

Para confirmar una obligación de pago se debe seleccionar la obligación que se desea confirmar. La obligación seleccionada debe estar en estado "guardado".

Para contabilizar una obligación de pago se debe seleccionar la obligación que se desea contabilizar. La obligación seleccionada debe estar en estado "confirmado". Se genera el asiento contable correspondiente a la obligación elegida.

Prototipo de interfaz de usuario: Adicionar obligación de pago. (Analistas, 2008)

- Requisito funcional R1.1.
- Requisito funcional R1.5.
- Requisito funcional R3.

El prototipo de interfaz de usuario muestra una ventana con el título "Adicionar obligación de pago". La interfaz está organizada en secciones:

- Encabezado:** Contiene campos para "Tipo documento:", "Número de documento:", "Proveedor:", "Cuenta:", "Fecha de emisión:", "Plazo(Días):", "D.P.P.(%)", y "Mora(%)".
- Operación:** Incluye un menú desplegable para "Operación:", un campo para "Moneda:", un campo para "Importe:", un menú desplegable para "Tasa:", y botones "+", "M", "-", y "X".
- Tabla:** Una tabla con los encabezados "Operación", "Moneda", "Importe", "Tasa", y "Importe original".
- Resumen:** Campos para "Importe total contable:" y "Pagado:".
- Comentarios:** Un campo de texto etiquetado como "Comentario:".
- Botones:** "Guardar", "Confirmar", "Contabilizar", y "Cerrar".

Figura 48. Adicionar obligación de pago.

Mediante esta interfaz el usuario puede registrar una nueva obligación de pago.

Prototipo de interfaz de usuario: Modificar obligación de pago.

- Requisito funcional R1.2. (Analistas, 2008)
- Requisito funcional R1.5. (Analistas, 2008)
- Requisito funcional R3. (Analistas, 2008)

The screenshot shows a software window titled "Modificar obligación de pago". The window contains the following elements:

- Top Section:** Fields for "Tipo documento:", "Número de documento:", "Proveedor:", "Plazo(Días):", "Cuenta:", "Fecha de emisión:", "D.P.P(%):", and "Mora(%):".
- Second Section:** Fields for "Operación:", "Moneda:", "Importe:", and "Tasa:", along with buttons for "+", "M", "-", and "X".
- Table:** A table with columns: "Operación", "Moneda", "Importe", "Tasa", and "Importe original".
- Bottom Section:** Fields for "Importe total contable:" and "Pagado:", a "Comentarios" field, and buttons for "Guardar", "Confirmar", "Contabilizar", and "Cerrar".

Figura 49. Modificar obligación de pago.

Para modificar una obligación de pago primero se debe seleccionar la obligación que se desea modificar en la interfaz Gestionar obligación de pago, además se debe verificar el estado del documento seleccionado, en caso de que sea "confirmado", no se puede modificar.

Prototipo de interfaz de usuario: Cancelar obligación de pago. (Analistas, 2008)

- Requisito funcional R1.3.

<input type="checkbox"/>	Tipo documento	No. documento	Proveedor	Fecha
--------------------------	----------------	---------------	-----------	-------

Figura 50. Cancelar obligación de pago.

Para cancelar una obligación de pago el usuario puede hacer la búsqueda más efectiva mediante un filtro con los siguientes datos (No. Documento, Proveedor). No se pueden cancelar aquellas obligaciones de pago que hayan sido liquidadas parcial o totalmente.

Prototipo de interfaz de usuario: Liquidar obligación de pago. (Analistas, 2008)

- Requisito funcional R1.4.

Figura 51. Liquidación de obligación.

Mediante esta interfaz se pueden liquidar varias obligaciones de pago. Para que se pueda realizar la liquidación de las obligaciones tiene que haberse registrado un instrumento de pago.

Cuando se realiza la liquidación de las obligaciones seleccionadas si queda importe disponible en el instrumento de pago, el usuario debe registrar el mismo en una cuenta determinada, mediante la siguiente interfaz:

Figura 52. Registrar diferencia.

Prototipo de interfaz de usuario: Gestionar obligaciones fiscales. (Analistas, 2008)

- Requisito funcional R2.5.
- Requisito funcional R3.

El prototipo de interfaz de usuario para 'Gestionar obligación fiscal' presenta un diseño con un encabezado azul que contiene el título de la ventana y botones de control (minimizar, maximizar, cerrar). Debajo del encabezado, hay una barra de herramientas con los botones 'Adicionar', 'Modificar', 'Cancelar', 'Contabilizar' y 'Confirmar'. A continuación, se encuentran los campos de búsqueda: 'Número de documento' (campo de texto), 'Fecha' (campo de texto con flecha hacia abajo), 'Tipo de documento' (campo de texto con flecha hacia abajo) y 'Estado' (campo de texto con flecha hacia abajo). El cuerpo principal de la interfaz es una tabla con cinco columnas: 'Número', 'Fecha', 'Tipo documento', 'Estado' y 'Comentario'. La tabla está actualmente vacía.

Figura 53. Gestionar obligación fiscal.

A través de esta interfaz el usuario puede realizar las siguientes funciones:

- Adicionar.
- Modificar.
- Cancelar.
- Confirmar.
- Contabilizar.

El usuario puede hacer su búsqueda más específica mediante un filtro con los siguientes datos (Número de documento, fecha, tipo de documento y estado).

Para confirmar una obligación fiscal se debe seleccionar la obligación que se desea confirmar. La obligación seleccionada debe estar en estado "guardado".

Para contabilizar una obligación fiscal se debe seleccionar la obligación que se desea contabilizar. La obligación seleccionada debe estar en estado "confirmado". Se genera el asiento contable correspondiente a la obligación elegida.

Prototipo de interfaz de usuario: Adicionar obligación fiscal. (Analistas, 2008)

- Requisito funcional R2.1.
- Requisito funcional R2.5.
- Requisito funcional R3.

The screenshot shows a software window titled "Adicionar obligación fiscal". It contains the following elements:

- Input fields for "Tipo documento:", "Número de documento:", "Párrafo fiscal:", "Cuenta:", and "Fecha de emisión:", each with a dropdown arrow.
- Input fields for "Plazo(Días):", "D.P.P(%)", and "Mora(%)".
- A row of controls: "Operación:" (dropdown), "Moneda:" (dropdown), "Importe:" (text box), "Tasa:" (dropdown), and buttons "+", "M", "-", and "X".
- A table with the following headers: "Operación", "Moneda", "Importe", "Tasa", and "Importe original". The table body is currently empty.
- Summary fields: "Importe total contable:" and "Pagado:".
- A "Comentario:" text area.
- Four buttons at the bottom: "Guardar", "Confirmar", "Contabilizar", and "Cerrar".

Figura 54. Adicionar obligación fiscal.

A través de esta interfaz se puede registrar una nueva obligación fiscal.

Prototipo de interfaz de usuario: Modificar obligación fiscal. (Analistas, 2008)

- Requisito funcional R2.2.
- Requisito funcional R2.5.
- Requisito funcional R3.

The screenshot shows a software window titled "Modificar obligación fiscal". The interface includes the following elements:

- Top Section:** Fields for "Tipo documento:" (dropdown), "Número de documento:" (text), "Párrafo fiscal:" (dropdown), "Plazo(Días):" (text), "Cuenta:" (dropdown), "Fecha de emisión:" (dropdown), "D.P.P(%)" (text), and "Mora(%)" (text).
- Second Section:** Fields for "Operación:" (dropdown), "Moneda:" (dropdown), "Importe:" (text), "Tasa:" (dropdown), and buttons for "+", "M", "-", and "X".
- Table:** A table with columns: "Operación", "Moneda", "Importe", "Tasa", and "Importe original". The table is currently empty.
- Bottom Section:** Fields for "Importe total contable:" and "Pagado:", a "Comentario:" text area, and buttons for "Guardar", "Confirmar", "Contabilizar", and "Cerrar".

Figura 55. Modificar obligación fiscal.

Para modificar una obligación fiscal primero se debe seleccionar la obligación que se desea modificar en la interfaz Gestionar obligación fiscal, además se debe verificar el estado del documento seleccionado, en caso de que sea “confirmado”, no se puede modificar.

Prototipo de interfaz de usuario: Cancelar obligación fiscal. (Analistas, 2008)

- Requisito funcional R2.3.



The screenshot shows a software window titled "Cancelar obligación de pago fiscal". It features a search interface with the following elements:

- No. documento:** A text input field followed by a "Filtrar" button.
- Motivo de cancelación:** A dropdown menu.
- Fecha:** A date selection dropdown menu.
- Table:** A table with a header row containing "Tipo de documento", "No. documento", and "Fecha". The table body is currently empty.
- Buttons:** "Aceptar" and "Cerrar" buttons located at the bottom right of the window.

Figura 56. Cancelar obligación fiscal.

Para cancelar una obligación fiscal el usuario puede hacer la búsqueda más efectiva mediante un filtro con el siguiente dato (No. documento). No se pueden cancelar aquellas obligaciones fiscales que hayan sido liquidadas parcial o totalmente.

Prototipo de interfaz de usuario: Liquidar obligación fiscal. (Analistas, 2008)

- Requisito funcional R2.4.

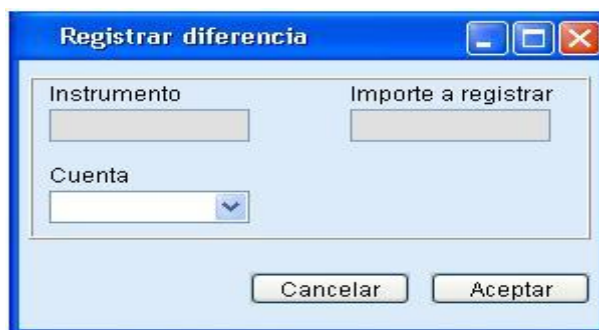
Instrumento de pago		
No. instrumento:	Importe:	Importe total a pagar:
12	20.00	10.00

No. documento	Moneda	Importe pendiente	Importe a pagar	Operación
---------------	--------	-------------------	-----------------	-----------

Figura 57. Liquidar obligación fiscal.

Mediante esta interfaz se pueden liquidar varias obligaciones fiscales. Para que se pueda realizar la liquidación de las obligaciones tiene que haberse registrado un instrumento de pago para operaciones fiscales.

Cuando se realiza la liquidación de las obligaciones seleccionadas si queda importe disponible en el instrumento de pago, el usuario debe registrar el mismo en una cuenta determinada, mediante la siguiente interfaz:



The screenshot shows a dialog box titled "Registrar diferencia". It contains two text input fields: "Instrumento" and "Importe a registrar". Below these is a dropdown menu labeled "Cuenta". At the bottom of the dialog are two buttons: "Cancelar" and "Aceptar".

Figura 58. Registrar diferencia.

Prototipo de interfaz de usuario: Adicionar instrumento de pago. (Analistas, 2008)

- Requisito funcional R3.



The screenshot shows a dialog box titled "Registrar instrumento de pago". It contains several fields: "Proveedor:" (dropdown), "Tipo de instrumento:" (dropdown), "Número:" (text input), "Moneda:" (dropdown), "Importe:" (text input), "Fecha emisión:" (dropdown), and "Fecha recepción:" (dropdown). Below these are two text input fields: "Elaborado por:" and "Comentario:". At the bottom are three buttons: "Cancelar", "Liquidar obligación", and "Registrar pago anticipado".

Figura 59. Adicionar instrumento de pago.

Cuando se registra un instrumento de pago se puede liquidar obligaciones de pago o registrar un pago anticipado.

Prototipo de interfaz de usuario: Adicionar instrumento de pago para obligaciones fiscales. (Analistas, 2008)

- Requisito funcional R4.



The image shows a software window titled "Registrar instrumento de pago (operaciones fiscales)". The window has a blue title bar with standard Windows window controls (minimize, maximize, close). The main area is light blue and contains the following fields:

- Tipo de instrumento:** A dropdown menu.
- Número:** A text input field.
- Moneda:** A dropdown menu.
- Importe:** A text input field.
- Fecha emisión:** A dropdown menu.
- Fecha recepción:** A dropdown menu.
- Elaborado por:** A text input field.
- Comentario:** A large text area.

At the bottom right of the window, there are two buttons: "Cancelar" and "Liquidar obligaciones".

Figura 60. Adicionar instrumento de pago (operaciones fiscales).

Mediante esta interfaz se registra un instrumento de pago, con el cual se pueden liquidar varias obligaciones fiscales.

Prototipo de interfaz de usuario: Gestionar pagos anticipados. (Analistas, 2008)

- Requisito funcional R5.5.
- Requisito funcional R3.



Figura 61. Gestionar pago anticipado.

A través de esta interfaz el usuario puede realizar las siguientes funciones:

- Modificar.
- Cancelar.
- Confirmar.
- Contabilizar.
- Liquidar.

El usuario para realizar alguna de las funciones mencionadas, puede hacer su búsqueda más específica mediante el siguiente filtro:



The image shows a window titled "Filtrar" with a blue header and standard Windows window controls. Inside the window, there are four input fields arranged in a 2x2 grid. The top-left field is labeled "No. instrumento:" and is a text box. The top-right field is labeled "Proveedor" and is a dropdown menu. The bottom-left field is labeled "Fecha:" and is a dropdown menu. The bottom-right field is labeled "Tipo de instrumento" and is a dropdown menu. At the bottom of the window, there are two buttons: "Cancelar" and "Aceptar".

Figura 62. Filtrar para el gestionar pago anticipado.

Para confirmar el pago anticipado mediante esta interfaz se debe haber guardado el pago anticipado.

Para contabilizar el pago anticipado mediante esta interfaz se debe haber guardado y confirmado el pago anticipado.

Prototipo de interfaz de usuario: Adicionar pago anticipado. (Analistas, 2008)

- Requisito funcional R5.1.
- Requisito funcional R5.5.
- Requisito funcional R3.



The image shows a window titled "Adicionar pago anticipado" with a blue header and standard Windows window controls. Inside the window, there are three input fields in a row: "No. instrumento:" with the value "12", "Moneda:" with the value "CUP", and "Importe:" with the value "30.00". Below these fields is a "Cuenta:" dropdown menu. At the bottom of the window, there are four buttons: "Guardar", "Confirmar", "Contabilizar", and "Cerrar".

Figura 63. Adicionar pago anticipado.

Mediante esta interfaz se puede registrar un pago anticipado, también permite confirmar o contabilizar el pago registrado.

Para confirmar el pago anticipado mediante esta interfaz se debe haber guardado el pago anticipado.

Para contabilizar el pago anticipado mediante esta interfaz se debe haber guardado y confirmado el pago anticipado.

Prototipo de interfaz de usuario: Modificar pago anticipado. (Analistas, 2008)

- Requisito funcional R5.2.
- Requisito funcional R5.5.
- Requisito funcional R3.



El prototipo de interfaz de usuario muestra una ventana con el título "Modificar pago anticipado". Dentro de la ventana, hay tres campos de texto etiquetados "No. instrumento:", "Moneda:" y "Importe:". Debajo de estos, hay un campo de lista desplegable etiquetado "Cuenta:". En la parte inferior de la ventana, hay cuatro botones: "Guardar", "Confirmar", "Contabilizar" y "Cerrar".

Figura 64. Modificar pago anticipado.

Para modificar un pago anticipado primero se debe seleccionar el pago que se desea modificar en la interfaz Gestionar pago anticipado, además se debe verificar el estado del documento seleccionado, en caso de que sea "confirmado", no se puede modificar.

Prototipo de interfaz de usuario: Cancelar un pago anticipado. (Analistas, 2008)

- Requisito funcional R5.3.

The image shows a software dialog box titled "Cancelar pago anticipado". It contains several input fields and buttons. At the top right are standard window control buttons (minimize, maximize, close). The main area has four search criteria: "Proveedor:" with a dropdown menu, "Motivo cancelación:" with a dropdown menu, "No. instrumen..." with a text input field, and "Fecha:" with a dropdown menu. A "Filtrar" button is positioned to the right of the "No. instrumen..." field. Below these fields is a table with four columns: "Tipo instrumento", "No. instrumento", "Cliente", and "Fecha". The table is currently empty. At the bottom right of the dialog are two buttons: "Aceptar" and "Cerrar".

Figura 65. Cancelar pago anticipado.

Para cancelar un pago anticipado el usuario puede hacer la búsqueda más efectiva mediante un filtro con los siguientes datos (No. instrumento, Cliente). No se pueden cancelar aquellos pagos anticipados que hayan sido liquidados parcial o totalmente.

Prototipo de interfaz de usuario: Liquidar pago anticipado. (Analistas, 2008)

- Requisito funcional R5.4.

Figura 66. Liquidar pago anticipado.

Mediante esta interfaz se pueden liquidar varias obligaciones de pago.

Cuando se realiza la liquidación de las obligaciones seleccionadas si queda importe disponible en el instrumento de pago, el usuario debe registrar el mismo en una cuenta determinada, mediante la siguiente interfaz:

Figura 67. Registrar diferencia.

Prototipo de interfaz de usuario: Gestionar obligación de pago, carga inicial. (Analistas, 2008)

- Requisito funcional R5.2.



Figura 68. Gestionar obligación de pago, carga inicial.

A través de esta interfaz se pueden realizar las siguientes funciones:

- Adicionar.
- Modificar.
- Eliminar.

Prototipo de interfaz de usuario: Adicionar obligación de pago, carga inicial. (Analistas, 2008)

- Requisito funcional R5.1.

The screenshot shows a software window titled "Adicionar obligación de pago". The window contains the following elements:

- Top Section:** Fields for "Tipo documento:" (dropdown), "Número documento:" (text), "Proveedor:" (dropdown), "Fecha de emisión:" (dropdown), "Cuenta:" (dropdown), "Plazo(Días):" (text), "D.P.P(%):" (text), and "Mora(%):" (text).
- Middle Section:** Fields for "Operación:" (dropdown), "Moneda:" (dropdown), "Importe original:" (text), and "Importe contable:" (text), followed by buttons "+", "M", "-", and "X".
- Table:** A table with columns "Operación", "Moneda", "Importe original", and "Importe contable". The table is currently empty.
- Bottom Section:** A field for "Importe total contable:" and a "Comentarios:" field.
- Buttons:** "Guardar" and "Cerrar" buttons at the bottom right.

Figura 69. Adicionar obligación de pago.

Mediante esta interfaz durante el proceso de carga inicial se registran las obligaciones de pago.

Prototipo de interfaz de usuario: Modificar obligación de pago, carga inicial. (Analistas, 2008)

- Requisito funcional R5.1.

The screenshot shows a software window titled "Modificar obligación de pago". The interface is organized into several sections:

- Header:** "Modificar obligación de pago" with standard window control buttons (minimize, maximize, close).
- Form Fields:**
 - Tipo documento:** A dropdown menu.
 - Número documento:** A text input field.
 - Proveedor:** A dropdown menu.
 - Fecha de emisión:** A date selection dropdown.
 - Cuenta:** A dropdown menu.
 - Plazo(Días):** A text input field.
 - D.P.P(%):** A text input field.
 - Mora(%):** A text input field.
- Operación Section:**
 - Operación:** A dropdown menu.
 - Moneda:** A dropdown menu.
 - Importe original:** A text input field.
 - Importe contable:** A text input field.
 - Buttons: "+", "M", "-", "X".
- Table:** A table with the following headers: "Operación", "Moneda", "Importe original", and "Importe contable". The table body is currently empty.
- Summary:** "Importe total contable:" followed by a text input field.
- Footer:** "Comentario:" followed by a large text area, and "Guardar" and "Cerrar" buttons.

Figura 70. Modificar obligación de pago, carga inicial.

Mediante esta interfaz se modifica una obligación de pago registrada durante el proceso de carga inicial.

Prototipo de interfaz de usuario: Gestionar obligaciones fiscales, carga inicial. (Analistas, 2008)

- Requisito funcional R5.5.



Figura 71. Gestionar obligación fiscal, carga inicial.

A través de esta interfaz se pueden realizar las siguientes funciones:

- Adicionar.
- Modificar.
- Eliminar.

Prototipo de interfaz de usuario: Adicionar obligación fiscal, carga inicial. (Analistas, 2008)

- Requisito funcional R5.3.

The screenshot shows a software window titled "Adicionar obligación fiscal". It contains several input fields and a table. At the top, there are fields for "Tipo documento:", "Número documento:", "Párrafo fiscal:", "Fecha de emisión:", and "Cuenta:". To the right, there are three input fields for "Plazo(Días):", "D.P.P(%)", and "Mora(%)". Below these, there are fields for "Operación:", "Moneda:", "Importe original:", and "Importe contable:", followed by buttons "+", "M", "-", and "X". A table with four columns: "Operación", "Moneda", "Importe original", and "Importe contable" is present. At the bottom right, there is a field for "Importe total contable:". At the very bottom, there is a "Comentario:" field and "Guardar" and "Cerrar" buttons.

Figura 72. Adicionar obligación fiscal, carga inicial.

Mediante esta interfaz durante el proceso de carga inicial se registran las obligaciones fiscales.

Prototipo de interfaz de usuario: Modificar obligación fiscal, carga inicial. (Analistas, 2008)

- Requisito funcional R5.4.

The screenshot shows a window titled "Modificar obligación fiscal" with the following fields and controls:

- Tipo documento:** Dropdown menu.
- Número documento:** Text input field.
- Párrafo fiscal:** Dropdown menu.
- Fecha de emisión:** Date selection field.
- Cuenta:** Dropdown menu.
- Plazo(Días):** Text input field.
- D.P.P(%):** Text input field.
- Mora(%):** Text input field.
- Operación:** Dropdown menu.
- Moneda:** Dropdown menu.
- Importe original:** Text input field.
- Importe contable:** Text input field.
- Buttons:** "+", "M", "-", "X" for calculations.
- Table:** A table with columns: Operación, Moneda, Importe original, Importe contable.
- Importe total contable:** Text input field.
- Comentario:** Text input field.
- Buttons:** "Guardar" and "Cerrar".

Figura 73. Modificar obligación fiscal, carga inicial.

Mediante esta interfaz se modifica una obligación fiscal registrada durante el proceso de carga inicial.

Prototipo de interfaz de usuario: Gestionar derecho de cobro, carga inicial. (Analistas, 2008)

- Requisito funcional R5.8.

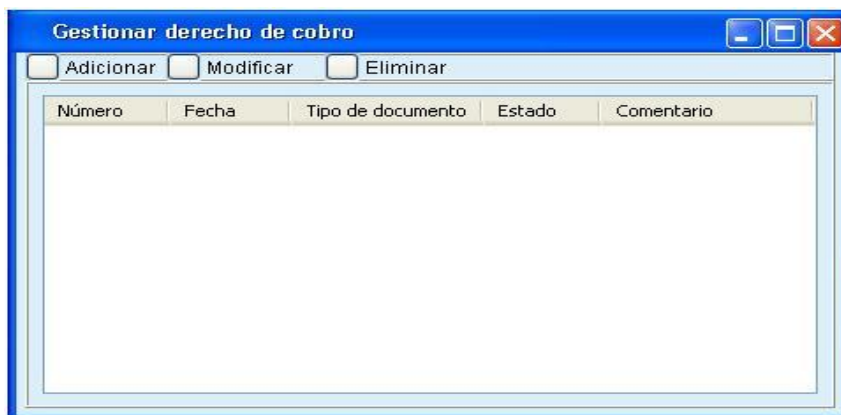


Figura 74. Gestionar derecho de cobro, carga inicial.

A través de esta interfaz se pueden realizar las siguientes funciones:

- Adicionar.
- Modificar.
- Eliminar.

Prototipo de interfaz de usuario: Adicionar derecho de cobro, carga inicial. (Analistas, 2008)

- Requisito funcional R5.6.

The screenshot shows a software window titled "Adicionar derecho de cobro". The interface is organized into several sections:

- Document Information:** Fields for "Tipo documento:" (dropdown), "Número documento:" (text), "Párrafo fiscal:" (dropdown), "Fecha de emisión:" (dropdown), and "Cuenta:" (dropdown).
- Financial Parameters:** A separate box containing "Plazo(Días):", "D.P.P(%)", and "Mora(%)" with corresponding input fields.
- Transaction Summary:** Fields for "Operación:" (dropdown), "Moneda:" (dropdown), "Importe original:" (text), and "Importe contable:" (text). To the right are buttons for "+", "M", "-", and "X".
- Table:** A table with four columns: "Operación", "Moneda", "Importe original", and "Importe contable". The table body is currently empty.
- Total:** A field labeled "Importe total contable:" with a text input box.
- Footer:** A "Comentario:" text area and two buttons: "Guardar" and "Cerrar".

Figura 75. Adicionar derecho de cobro, carga inicial.

Mediante esta interfaz durante el proceso de carga inicial se registran los derechos de cobro.

Prototipo de interfaz de usuario: Modificar derecho de cobro, carga inicial. (Analistas, 2008)

- Requisito funcional R5.7.

The screenshot shows a software window titled "Modificar derecho de cobro". The interface is organized into several sections:

- Top Section:** Contains fields for "Tipo documento:" (dropdown), "Número documento:" (text input), "Párrafo fiscal:" (dropdown), "Fecha de emisión:" (dropdown), "Cuenta:" (dropdown), "Plazo(Días):" (text input), "D.P.P(%):" (text input), and "Mora(%):" (text input).
- Second Section:** Contains "Operación:" (dropdown), "Moneda:" (dropdown), "Importe original:" (text input), "Importe contable:" (text input), and four buttons: "+", "M", "-", and "X".
- Table Section:** A table with four columns: "Operación", "Moneda", "Importe original", and "Importe contable". The table is currently empty.
- Bottom Section:** Contains a text input field for "Comentario:" and two buttons: "Guardar" and "Cerrar".

Figura 76. Modificar derecho de cobro, carga inicial.

Mediante esta interfaz se modifica un derecho de cobro registrado durante el proceso de carga inicial.

Prototipo de interfaz de usuario: Gestionar derechos fiscales, carga inicial. (Analistas, 2008)

- Requisito funcional R5.11.



Figura 77. Gestionar derecho fiscal.

A través de esta interfaz se pueden realizar las siguientes funciones:

- Adicionar.
- Modificar.
- Eliminar.

Prototipo de interfaz de usuario: Adicionar derechos fiscales, carga inicial. (Analistas, 2008)

- Requisito funcional R5.9.

Operación	Moneda	Importe original	Importe contable
-----------	--------	------------------	------------------

Figura 78. Adicionar derecho fiscal, carga inicial.

Mediante esta interfaz durante el proceso de carga inicial se registran los derechos fiscales.

Prototipo de interfaz de usuario: Modificar derechos fiscales, carga inicial. (Analistas, 2008)

- Requisito funcional R5.10.

The screenshot shows a software window titled "Modificar derecho fiscal". The interface includes several input fields and controls:

- Top Section:** Fields for "Tipo documento:" (dropdown), "Número documento:" (text), "Párrafo fiscal:" (dropdown), "Fecha de emisión:" (dropdown), and "Cuenta:" (dropdown). To the right are three input boxes for "Plazo(Días):", "D.P.P(%)", and "Mora(%)".
- Second Section:** Fields for "Operación:" (dropdown), "Moneda:" (dropdown), "Importe original:" (text), and "Importe contable:" (text). To the right are four buttons: "+", "M", "-", and "X".
- Table:** A table with four columns: "Operación", "Moneda", "Importe original", and "Importe contable". The table body is currently empty.
- Bottom Section:** A text field for "Comentario:" and a label "Importe total contable:" followed by a text input box.
- Footer:** Two buttons labeled "Guardar" and "Cerrar".

Figura 79. Modificar derechos fiscales, carga inicial.

Mediante esta interfaz se modifica un derecho fiscal registrado durante el proceso de carga inicial.

Prototipo de interfaz de usuario: Gestionar cobros anticipados, carga inicial. (Analistas, 2008)

- Requisito funcional R5.14.



Figura 80. Gestionar cobro anticipado, carga inicial.

A través de esta interfaz se pueden realizar las siguientes funciones:

- Adicionar.
- Modificar.
- Eliminar.

Prototipo de interfaz de usuario: Adicionar cobro anticipado, carga inicial. (Analistas, 2008)

- Requisito funcional R5.12.

El formulario 'Adicionar cobro anticipado' contiene los siguientes campos:

- Tipo de instrumento: menú desplegable.
- No. instrumento: campo de texto.
- Cliente: menú desplegable.
- Moneda: menú desplegable.
- Importe original: campo de texto.
- Importe contable: campo de texto.
- Fecha emisión: menú desplegable.
- Cuenta: menú desplegable.
- Fecha recepción: menú desplegable.
- Elaborador por: campo de texto.
- Comentario: campo de texto.

En la parte inferior del formulario se encuentran los botones 'Guardar' y 'Cerrar'.

Figura 81. Adicionar cobro anticipado, carga inicial.

Mediante esta interfaz durante el proceso de carga inicial se registran los cobros anticipados.

Prototipo de interfaz de usuario: Modificar cobro anticipado, carga inicial. (Analistas, 2008)

- Requisito funcional R5.13.

The screenshot shows a window titled "Modificar cobro anticipado". It contains the following fields and controls:

- Tipo de instrumento: [dropdown menu]
- No. instrumento: [text input]
- Cliente: [dropdown menu]
- Moneda: [dropdown menu]
- Importe original: [text input]
- Importe contable: [text input]
- Fecha emisión: [dropdown menu]
- Cuenta: [dropdown menu]
- Fecha recepción: [dropdown menu]
- Elaborador por: [text input]
- Comentario: [text area]
- Buttons: Guardar, Cerrar

Figura 82. Modificar cobro anticipado, carga inicial.

Mediante esta interfaz se modifica un cobro anticipado registrado durante el proceso de carga inicial.

Prototipo de interfaz de usuario: Gestionar pago anticipado, carga inicial. (Analistas, 2008)

- Requisito funcional R5.17.

The screenshot shows a window titled "Gestionar pago anticipado". It contains the following elements:

- Radio buttons: Adicionar, Modificar, Eliminar
- Table with columns: No. instrumento, Proveedor, Fecha, Importe original, Estado

Figura 83. Gestionar pago anticipado, carga inicial.

A través de esta interfaz se pueden realizar las siguientes funciones:

- Adicionar.
- Modificar.
- Eliminar.

Prototipo de interfaz de usuario: Adicionar pago anticipado, carga inicial. (Analistas, 2008)

- Requisito funcional R5.15.

El prototipo de interfaz de usuario, titulado "Adicionar pago anticipado", presenta un formulario con los siguientes campos:

- Tipo de instrumento: menú desplegable.
- No. instrumento: campo de texto.
- Proveedor: menú desplegable.
- Moneda: menú desplegable.
- Importe original: campo de texto.
- Importe contable: campo de texto.
- Fecha emisión: menú desplegable.
- Cuenta: menú desplegable.
- Fecha recepción: menú desplegable.
- Elaborador por: campo de texto.
- Comentario: campo de texto.

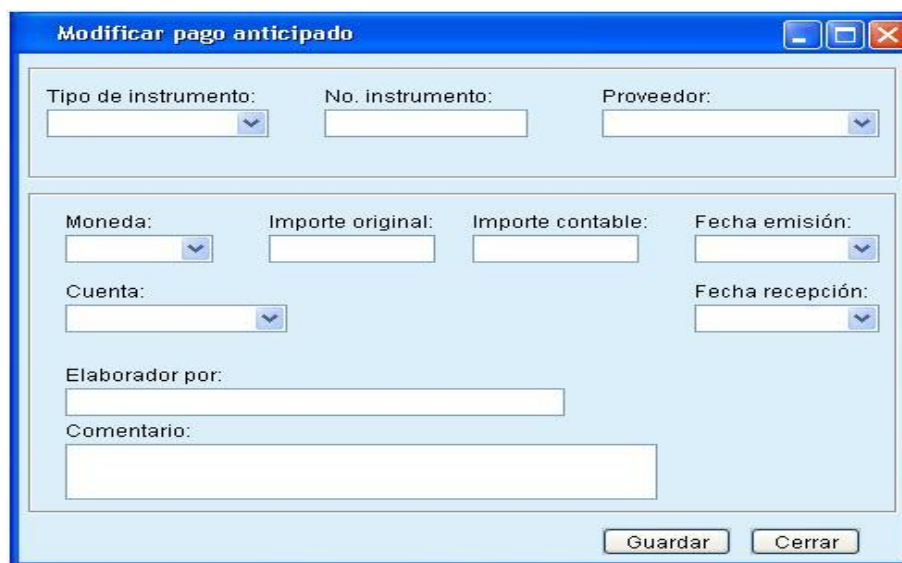
En la parte inferior derecha del formulario se encuentran dos botones: "Guardar" y "Cerrar".

Figura 84. Adicionar pago anticipado, carga inicial.

Mediante esta interfaz durante el proceso de carga inicial se registran los pagos anticipados.

Prototipo de interfaz de usuario: Modificar pago anticipado, carga inicial.

- Requisito funcional R5.16. (Analistas, 2008)

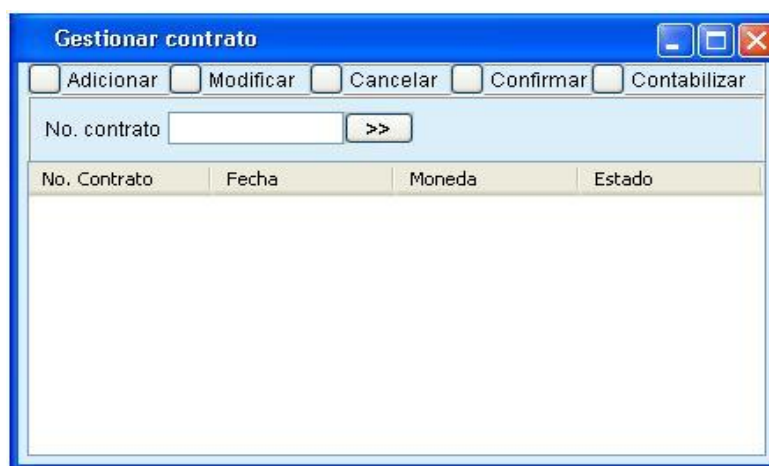


The screenshot shows a Windows-style window titled "Modificar pago anticipado". It contains several input fields and buttons. At the top, there are three fields: "Tipo de instrumento:" (a dropdown menu), "No. instrumento:" (a text box), and "Proveedor:" (a dropdown menu). Below these are four more fields: "Moneda:" (a dropdown menu), "Importe original:" (a text box), "Importe contable:" (a text box), and "Fecha emisión:" (a dropdown menu). Further down are "Cuenta:" (a dropdown menu) and "Fecha recepción:" (a dropdown menu). At the bottom of the form area are two text boxes: "Elaborador por:" and "Comentario:". At the very bottom of the window are two buttons: "Guardar" and "Cerrar".

Figura 85. Modificar pago anticipado, carga inicial.

Mediante esta interfaz se modifica un pago anticipado registrado durante el proceso de carga inicial.

Prototipo de interfaz de usuario: Gestionar contrato.



The screenshot shows a Windows-style window titled "Gestionar contrato". At the top, there are five buttons: "Adicionar", "Modificar", "Cancelar", "Confirmar", and "Contabilizar". Below these buttons is a text box labeled "No. contrato" followed by a ">>" button. Underneath is a table with four columns: "No. Contrato", "Fecha", "Moneda", and "Estado". The table is currently empty.

Figura 86. Gestionar contrato.

A través de esta interfaz se pueden realizar las siguientes funciones:

- Adicionar.
- Modificar.
- Cancelar.
- Confirmar.
- Contabilizar.

Prototipo de interfaz de usuario: Adicionar contrato. (Analistas, 2008)

- Requisito funcional R6.1.

El prototipo de interfaz de usuario para 'Adicionar contrato' presenta un diseño con un encabezado azul que contiene el título de la ventana y los botones de control de ventana. El contenido principal está dividido en secciones de configuración y de datos. En la parte superior izquierda, hay cuatro botones de radio para seleccionar el tipo de contrato: 'Derechos fiscales', 'Obligaciones fiscales', 'Derechos de cobro' y 'Obligaciones de pago'. Debajo de estos, hay un campo de texto para 'Cliente' con un menú desplegable y un botón '>>'. A continuación, se encuentran los campos para 'No. contrato', 'Fecha' (con un menú desplegable) y 'Moneda' (con un menú desplegable). Un campo de texto 'Elaborado por:' y un campo de texto más grande 'Comentario:' completan esta sección. A la derecha, hay dos cuadros de tabla. El primero tiene una columna 'No. documento' y una columna 'Importe'. El segundo tiene tres columnas: 'Operación', 'Moneda original' y 'TC'. En la parte inferior de la ventana, hay cuatro botones: 'Guardar', 'Confirmar', 'Contabilizar' y 'Cerrar'.

Figura 87. Adicionar contrato.

Mediante esta interfaz se puede registrar un nuevo contrato.

Prototipo de interfaz de usuario: Modificar contrato. (Analistas, 2008)

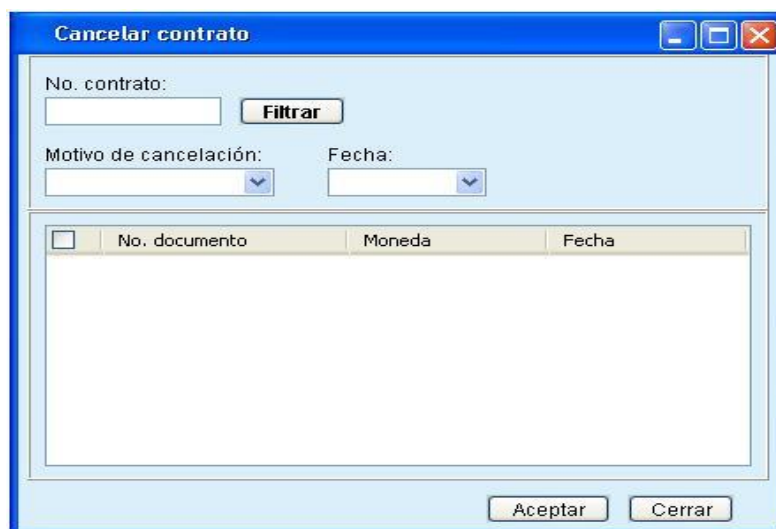
- Requisito funcional R6.2.

Figura 88. Modificar contrato.

Aquellos contratos en los cuales se hayan realizado operaciones con las obligaciones o derechos que tienen asociadas o que se hayan contabilizado, no se pueden modificar.

Prototipo de interfaz de usuario: Cancelar contrato. (Analistas, 2008)

- Requisito funcional R6.3.

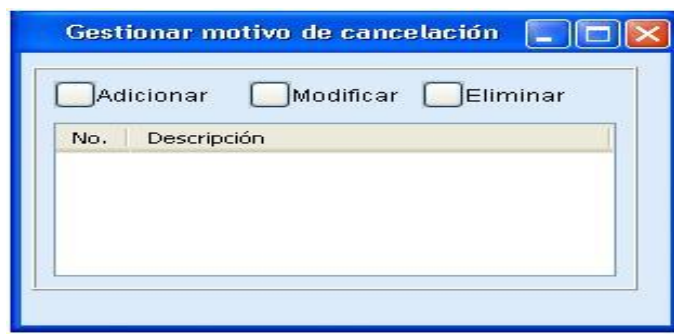


<input type="checkbox"/>	No. documento	Moneda	Fecha
--------------------------	---------------	--------	-------

Figura 89. Cancelar contrato.

Mediante esta interfaz se puede cancelar un contrato, la búsqueda se puede hacer mas especifica realizando un filtro por el número del contrato.No se pueden cancelar aquellos contratos en los que ya se hayan realizado operaciones con las obligaciones o derechos asociadas. Si la gestión del contrato ya fue contabilizada, esta cancelación revierte dicha operación y generará asientos contables.

Prototipo de interfaz de usuario: Gestionar motivos de cancelación.



No.	Descripción
-----	-------------

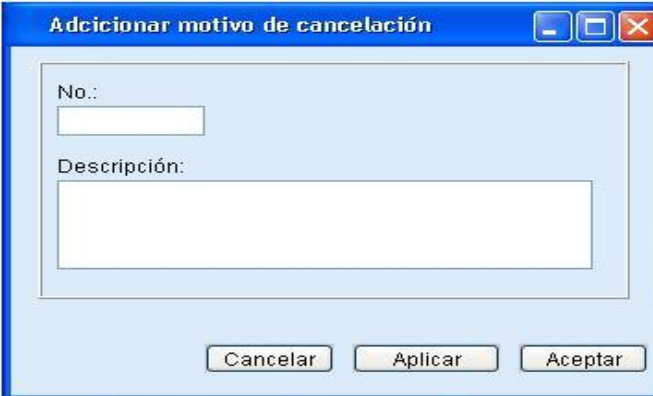
Figura 90. Gestionar motivos de cancelación.

A través de esta interfaz se pueden realizar las siguientes funciones:

- Adicionar.
- Modificar.
- Eliminar.

Prototipo de interfaz de usuario: Adicionar motivo de cancelación. (Analistas, 2008)

- Requisito funcional R1.1.



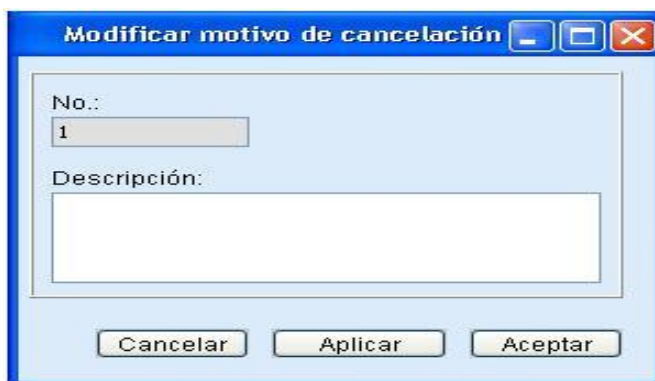
El prototipo de interfaz de usuario muestra una ventana con el título "Adicionar motivo de cancelación". Dentro de la ventana, hay un campo de texto etiquetado "No.:" y un campo de texto más grande etiquetado "Descripción:". En la parte inferior de la ventana, hay tres botones: "Cancelar", "Aplicar" y "Aceptar".

Figura 91. Adicionar motivo de cancelación.

Mediante esta interfaz se puede registrar un nuevo motivo de cancelación.

Prototipo de interfaz de usuario: Modificar motivo de cancelación. (Analistas, 2008)

- Requisito funcional R1.2.



Este es un prototipo de interfaz de usuario para una ventana titulada "Modificar motivo de cancelación". La ventana tiene un fondo azul claro y una barra de título azul con botones de control de ventana (minimizar, maximizar, cerrar). Dentro de la ventana, hay un campo de texto etiquetado "No." con el valor "1" ingresado. Debajo de esto, hay un campo de texto etiquetado "Descripción:" que está vacío. En la parte inferior de la ventana, hay tres botones: "Cancelar", "Aplicar" y "Aceptar".

Figura 92. Modificar motivo de cancelación.

Mediante esta interfaz se puede modificar un motivo de cancelación registrado. No se pueden modificar motivos de cancelación que tengan movimientos asociados.

Prototipo de interfaz de usuario: Gestionar párrafos fiscales.



Este es un prototipo de interfaz de usuario para una ventana titulada "Gestionar párrafo fiscal". La ventana tiene un fondo azul claro y una barra de título azul con botones de control de ventana (minimizar, maximizar, cerrar). Dentro de la ventana, hay tres botones de opción: "Adicionar", "Modificar" y "Eliminar", todos desactivados. Debajo de estos botones, hay una tabla con dos columnas: "Código" y "Descripción". La tabla está actualmente vacía.

Figura 93. Gestionar párrafo fiscal.

A través de esta interfaz se pueden realizar las siguientes funciones:

- Adicionar.
- Modificar.
- Eliminar.

Prototipo de interfaz de usuario: Adicionar párrafo fiscal. (Analistas, 2008)

- Requisito funcional R2.1.

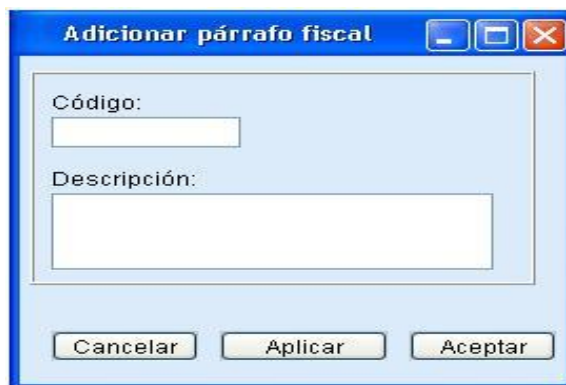
A screenshot of a Windows-style dialog box titled "Adicionar párrafo fiscal". The dialog has a blue title bar with standard minimize, maximize, and close buttons. Inside, there are two input fields: "Código:" with a small text box, and "Descripción:" with a larger text area. At the bottom, there are three buttons: "Cancelar", "Aplicar", and "Aceptar".

Figura 94. Adicionar párrafo fiscal.

Mediante esta interfaz se puede registrar un párrafo fiscal.

Prototipo de interfaz de usuario: Modificar párrafo fiscal. (Analistas, 2008)

- Requisito funcional R2.2.

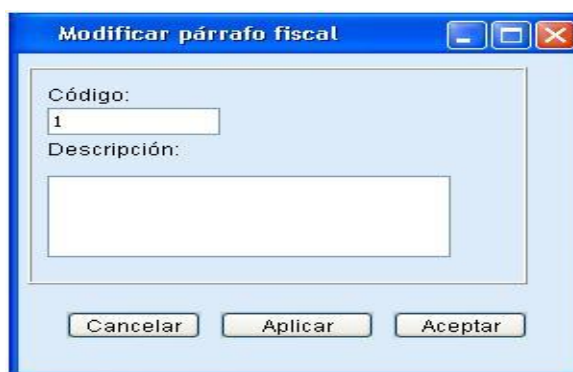
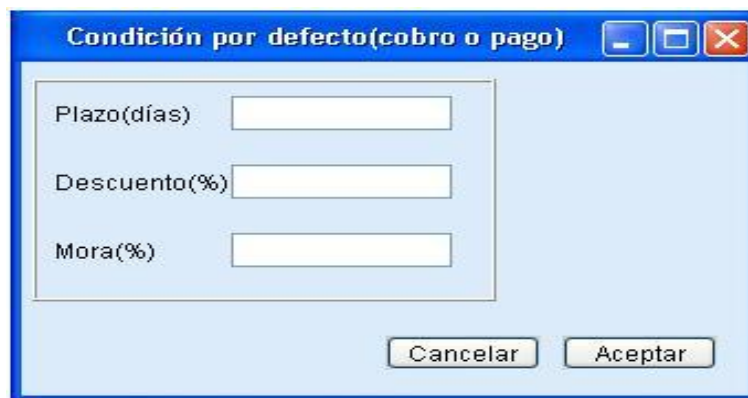
A screenshot of a Windows-style dialog box titled "Modificar párrafo fiscal". The dialog has a blue title bar with standard minimize, maximize, and close buttons. Inside, there are two input fields: "Código:" with a small text box containing the number "1", and "Descripción:" with a larger text area. At the bottom, there are three buttons: "Cancelar", "Aplicar", and "Aceptar".

Figura 95. Modificar párrafo fiscal.

Mediante esta interfaz se puede modificar un párrafo fiscal registrado. Solo se pueden modificar aquellos párrafos fiscales que tengan movimientos asociados.

Prototipo de interfaz de usuario: Adicionar condición de pago por defecto. (Analistas, 2008)

- Requisito funcional R3.



El prototipo muestra una ventana de diálogo con el título "Condición por defecto(cobro o pago)". Dentro de la ventana, hay un recuadro que contiene tres campos de entrada de texto, cada uno con una etiqueta a su izquierda: "Plazo(días)", "Descuento(%)" y "Mora(%)". Debajo de este recuadro, se encuentran dos botones: "Cancelar" y "Aceptar".

Figura 96. Adicionar condición de pago por defecto.

Mediante esta interfaz se puede registrar una nueva condición de pago por defecto para todos los documentos gestionados en el módulo Cobros y Pagos.

El resto de la información referente a los prototipos de interfaz del módulo Cobros y Pagos se encuentran en el manual de usuario realizado por los analistas del módulo. (Analistas, 2009)

Prototipo de interfaz de usuario: Submayor. (Analistas, 2008)

- Requisito funcional R1.2.
- Requisito funcional R1.3.
- Requisito funcional R1.4.
- Requisito funcional R1.5.

No. documento	Fecha	Comentario	Debe	Haber	Saldo
---------------	-------	------------	------	-------	-------

Figura 97. Submayor.

Mediante esta interfaz el usuario puede calcular el submayor de (Derecho de cobro, Obligación de pago, Cobros anticipados, Pagos anticipados) por varios criterios (Moneda, Cuentas, Cliente), además puede imprimir el submayor generado.

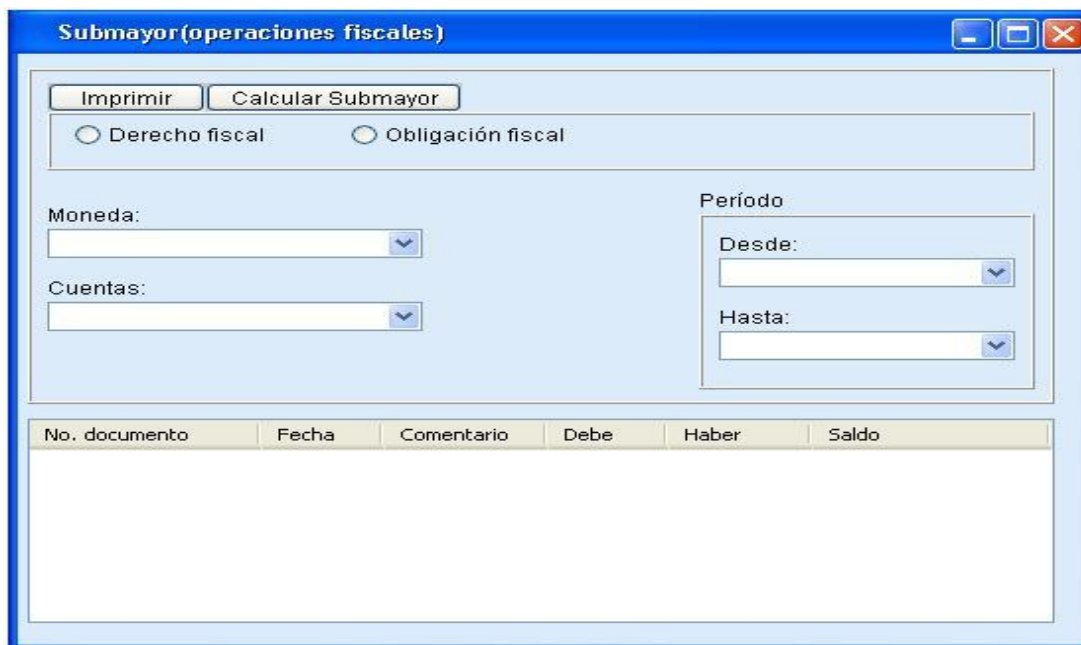


Figura 98. Submayor (operaciones fiscales).

Mediante esta interfaz el usuario puede calcular el submayor de (Derecho fiscal o Obligación fiscal) por varios criterios (Cuentas o por Moneda), además puede imprimir el submayor generado.

2.5. Validación de los requerimientos funcionales.

Los requisitos funcionales se validan con el objetivo de verificar que estos hayan sido especificados sin ambigüedades, inconsistencias, omisiones y que el resultado obtenido se ajuste a los estándares establecidos por el proyecto.

Los analistas del módulo Cobros y Pagos para validar los 109 requisitos funcionales especificados utilizaron las siguientes técnicas:

- Revisión técnica formal: incluye recorridos, inspecciones y revisiones cíclicas. Cada Revisión Técnica Formal (RTF) se lleva a cabo mediante una reunión y solo tiene éxito si esta es bien planificada, controlada y atendida. Los analistas del módulo una vez terminada las especificaciones de los requisitos realizaron la reunión de revisión, a la cual convocaron a la especialista funcional a cargo del módulo Cobros y Pagos Maria Elena Miranda Delgado, al Dr.

José Carlos Del Toro Ríos y al Ing. Rolando Ramírez Concepción representante del UCID, los cuales firmaron y aprobaron las especificaciones descritas sin posteriores modificaciones.

- **Prototipos:** Otra de las técnicas utilizadas fue a través de los prototipos de interfaz de usuario, ya que estos constituyen una visión inicial del sistema que favorece la comunicación entre clientes y desarrolladores. Un prototipo puede revelar errores u omisiones en los requisitos propuestos. Esta técnica tiene un pequeño problema y es que el cliente cree que lo que está viendo es el sistema final y no un prototipo, los analistas del módulo Cobros y Pagos se enfrentaron con este problema, al cual dieron solución de forma rápida a través de la plática, exponiéndole al cliente la diferencia entre uno y otro.

2.6. Conclusiones parciales.

Luego de varios encuentros realizados con la especialista funcional asignada por el proyecto para el módulo Cobros y Pagos se lograron identificar y describir cada uno de los procesos involucrados en el módulo. Se especificaron y validaron con los funcionales asignados los requisitos funcionales que el sistema debe cumplir para el módulo Cobros y Pagos. Además, se realizaron los prototipos de interfaz de usuario de acuerdo a las especificaciones de los requisitos planteados.

CAPÍTULO III: DISEÑO.

3.1. Introducción:

En el presente capítulo se realiza el diseño del módulo Cobros y Pagos, presentando las clases del diseño realizadas utilizando los estereotipos web, así como una descripción de las mismas. Además serán estudiados los patrones de diseño empleados, así como los mecanismos de diseño elaborados por los diseñadores del módulo. Se evalúa el diseño realizado a través de métricas.

3.2. Diseño.

El diseño es el centro de atención al final de la fase de elaboración y el comienzo de las iteraciones de construcción, lo cual contribuye al desarrollo de una arquitectura estable y sólida, y a crear un plano del modelo de implementación. Este flujo de trabajo es un refinamiento del análisis que tiene en cuenta los requisitos no funcionales, en definitiva cómo cumple el sistema sus objetivos.

Los objetivos del diseño son:

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

El módulo Cobros y Pagos se va a realizar a través de una aplicación web, por cuanto se decidió que para el desarrollo de las clases del diseño se utilizarán los siguientes estereotipos web:

- Páginas clientes o client page: es una página web con formato HTML. Mezcla de datos, presentación y lógica. Son interpretadas por el navegador. Sus atributos son las variables declaradas dentro del *script*.
- Build: Representa una asociación especial que relaciona las páginas clientes con las páginas servidoras, de forma general se expresa como que las páginas que se encuentran en el servidor construyen las páginas en el cliente.

- Submit: Es la relación que se crea siempre entre una página servidora y un formulario, a través de esta relación el formulario envía los valores de sus campos al servidor, para ser procesados por las páginas servidoras.

3.3. Mecanismos de diseño.

Los mecanismos de diseño se utilizan con el objetivo de simplificar los diagramas de clases. Cada diseñador de acorde a sus necesidades establece sus propios mecanismos de diseño, teniendo siempre en cuenta los patrones y estilos seleccionados. (2009)

Beneficios que ofrecen los mecanismos de diseño:

- Mantienen la homogeneidad en el diseño.
- Permiten reutilizar soluciones ya probadas.
- Permiten reutilizar documentación.

En el módulo Cobros y Pagos se definieron los siguientes mecanismos de diseño:

- Mecanismo de diseño para las páginas clientes.

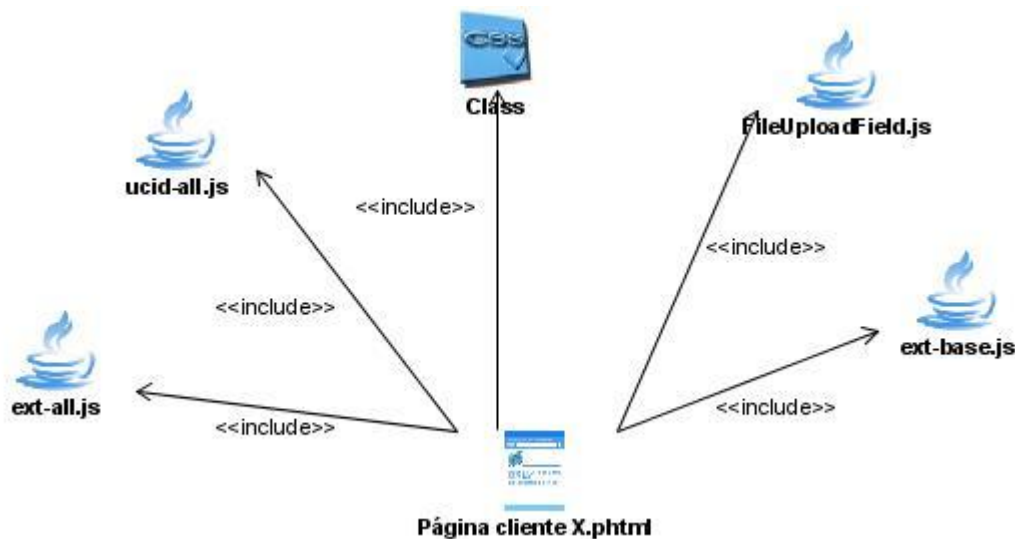


Figura 1. Mecanismo de diseño para las páginas clientes.

Todas las páginas clientes del módulo incluyen las siguientes clases:

- Ext-all.js: Es la encargada de la creación de los componentes visuales de la vista. Está incluida dentro de las clases que trae EXT JS.
- Ext-base.js: Encargada del manejo de las solicitudes y respuestas, manejo de componentes de EXT.
- Ucid-all.js: Encargada de mostrar la interfaz estándar de CEDRUX.
- FileUploadFieldSu función es cargar los ficheros.

Mecanismo de diseño para las clases controladoras.

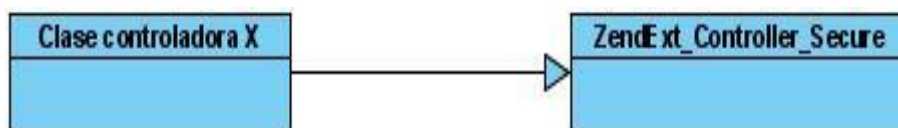


Figura 2. Mecanismo de diseño para las clases controladoras.

Todas las clases controladoras definidas en el diseño del módulo Cobros y Pagos heredan de la clase ZendExt_Controller_Secure, ya que en ella se incluyen numerosas funcionalidades comunes.

Mecanismo de diseño para las clases modelos.



Figura 3. Mecanismo de diseño para las clases modelos.

Todas las clases modelos o model definidas en el diseño heredan de la clase ZendExtModel, ya que esta incluye las principales funciones para el manejo de los datos.

3.4. Diagrama de clases del diseño.

Los diagramas de clases del diseño: (Canaviri, y otros)

- Se realizan con el objetivo de tener en cuenta los detalles concretos de la implementación del sistema.
- Son la base para los diagramas de componentes y los diagramas de despliegue.
- Los diagramas de clases del diseño son importantes no sólo para visualizar, especificar y documentar modelos estructurales, sino también para construir sistemas ejecutables, aplicando ingeniería directa e inversa.

Los diagramas de clases del diseño del módulo Cobros y Pagos se realizaron según los componentes definidos en el módulo:

- Conciliación.
- Configuración.
- Cierre.
- Carga inicial.
- Submayor.
- Funcionalidades: En este componente se encuentran las siguientes funcionalidades: liquidación, cambio de moneda y cambio de cuenta.
- Derecho_Obligación.

Diagrama de clases del diseño del componente Conciliación.

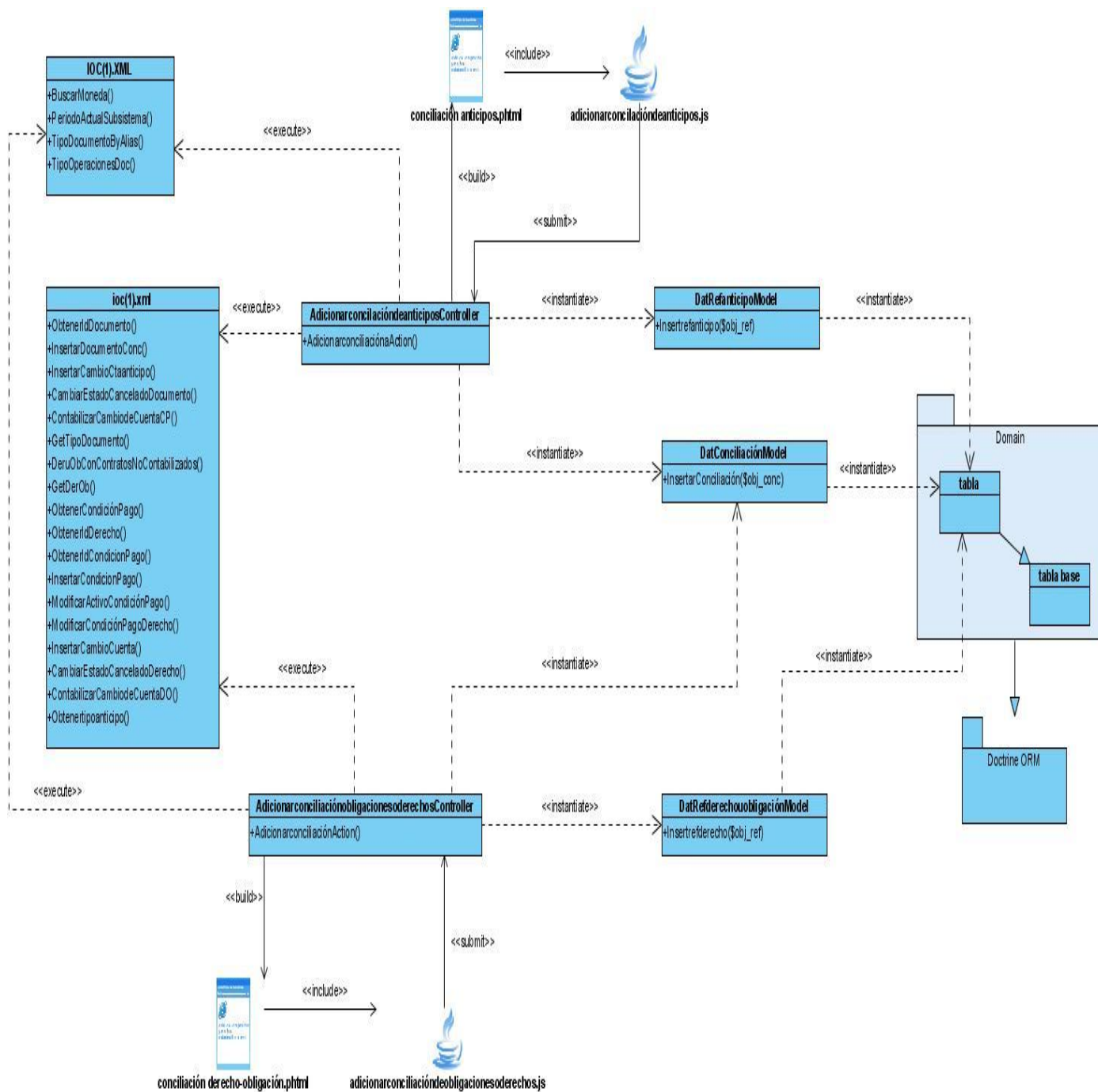


Figura 1. Diagrama de clases del diseño del componente Conciliación

Diagrama de clases del paquete de dominio del componente Conciliación.

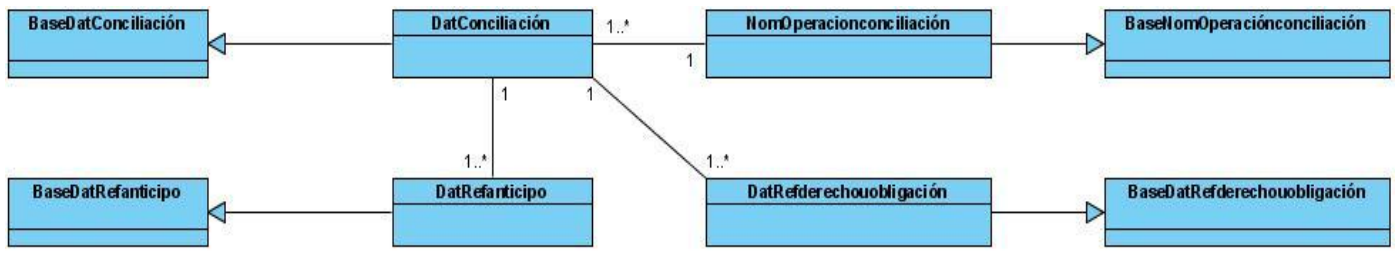


Figura 2. Diagrama de clases del paquete de dominio del componente Conciliación.

Diagrama de clases del diseño del componente Configuración.

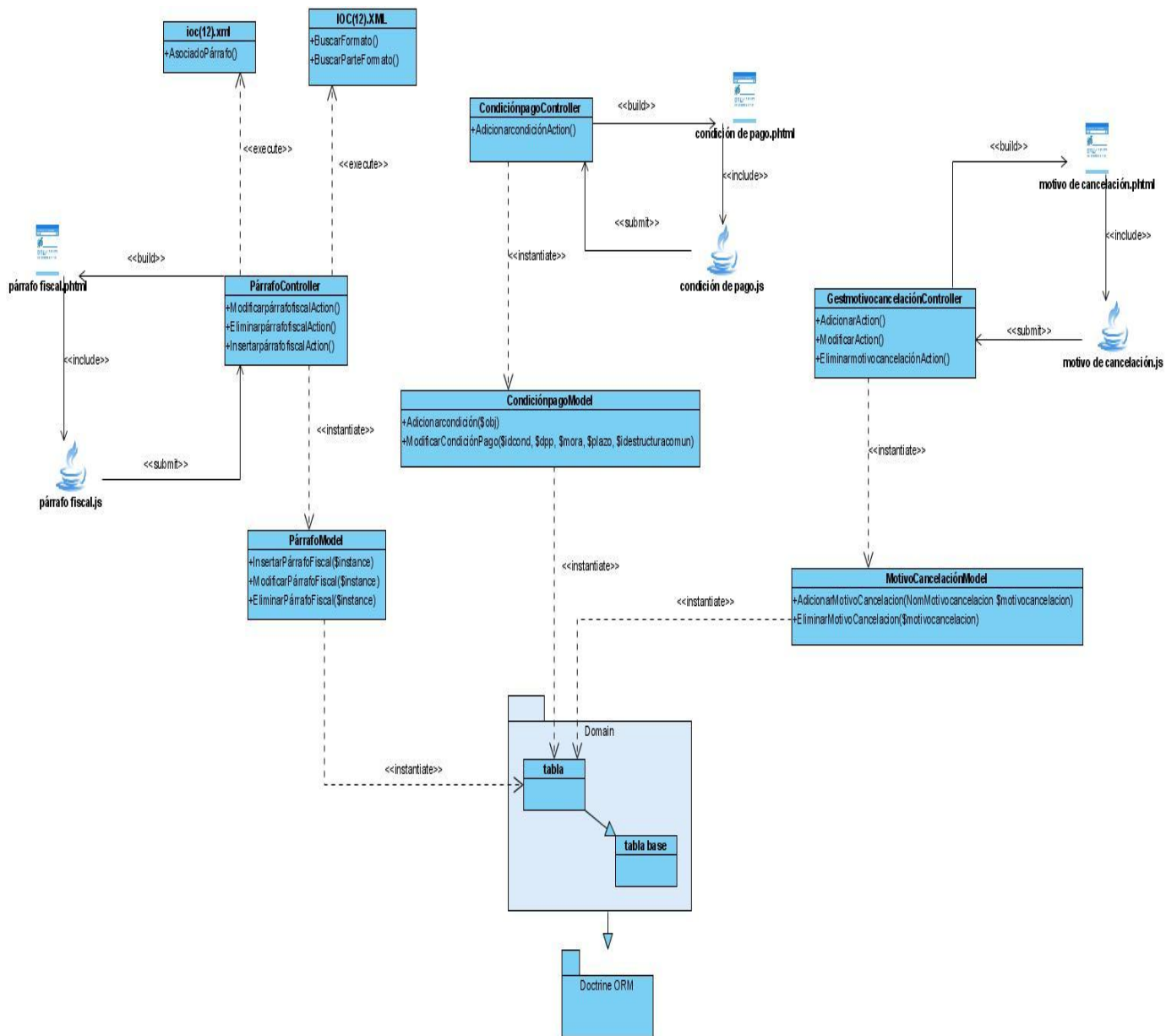


Figura 3. Diagrama de clases del diseño del componente Configuración.

Diagrama de clases del paquete de dominio del componente Configuración.

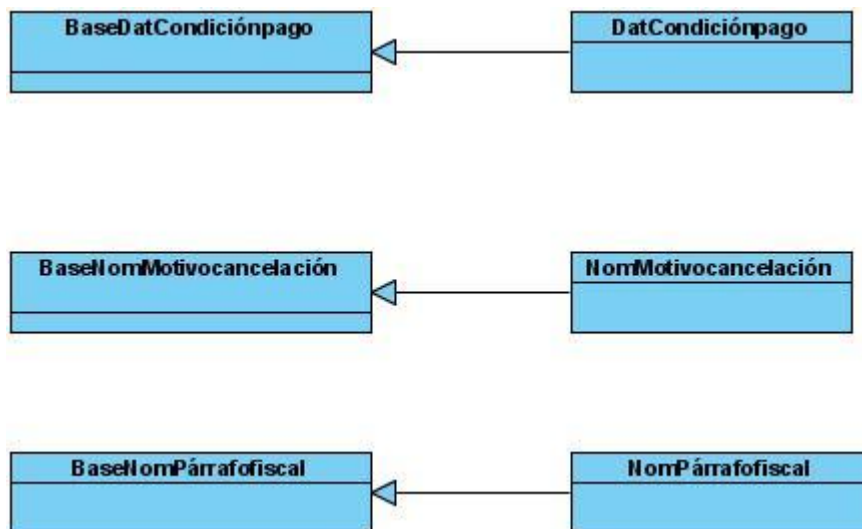


Figura 4. Diagrama de clases del paquete de dominio del componente Configuración.

Diagrama de clases del diseño del componente Cierre.

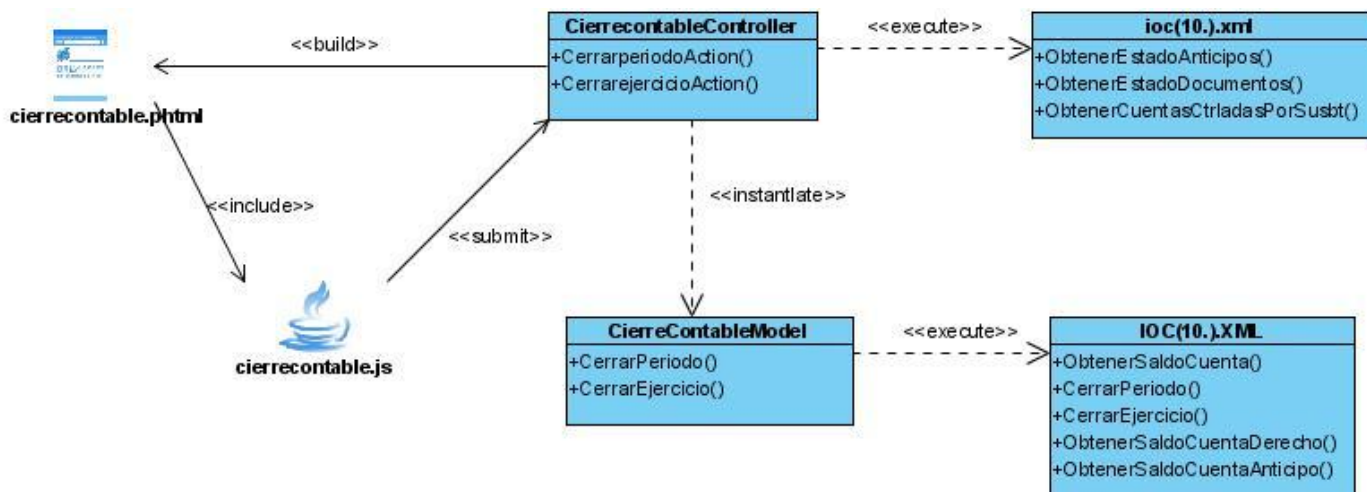


Figura 5. Diagrama de clases del diseño del componente Cierre.

Diagrama de clases del diseño del componente Carga inicial.

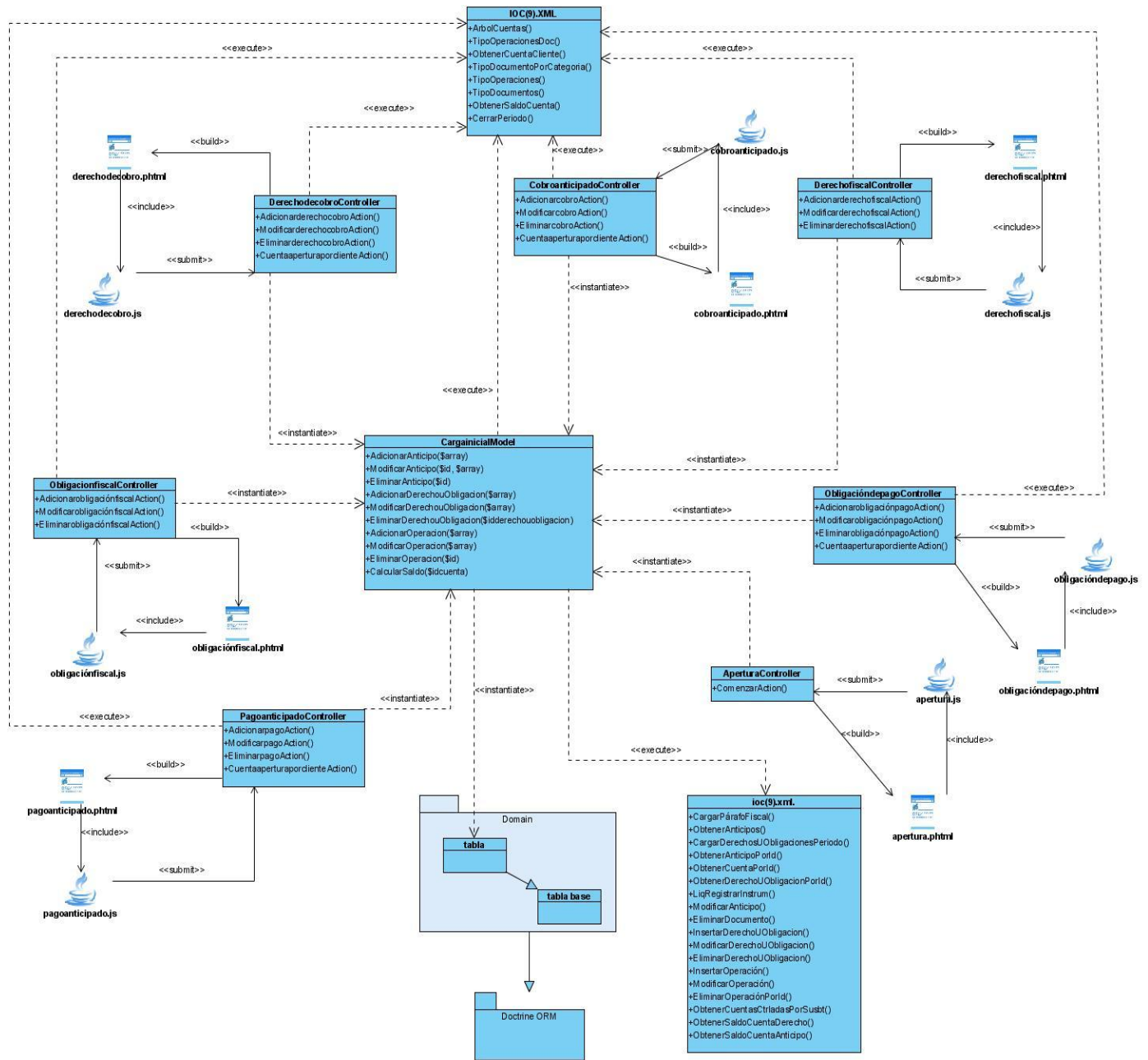


Figura 6. Diagrama de clases del diseño del componente carga inicial.

Diagrama de clases del paquete de dominio del componente Carga inicial.

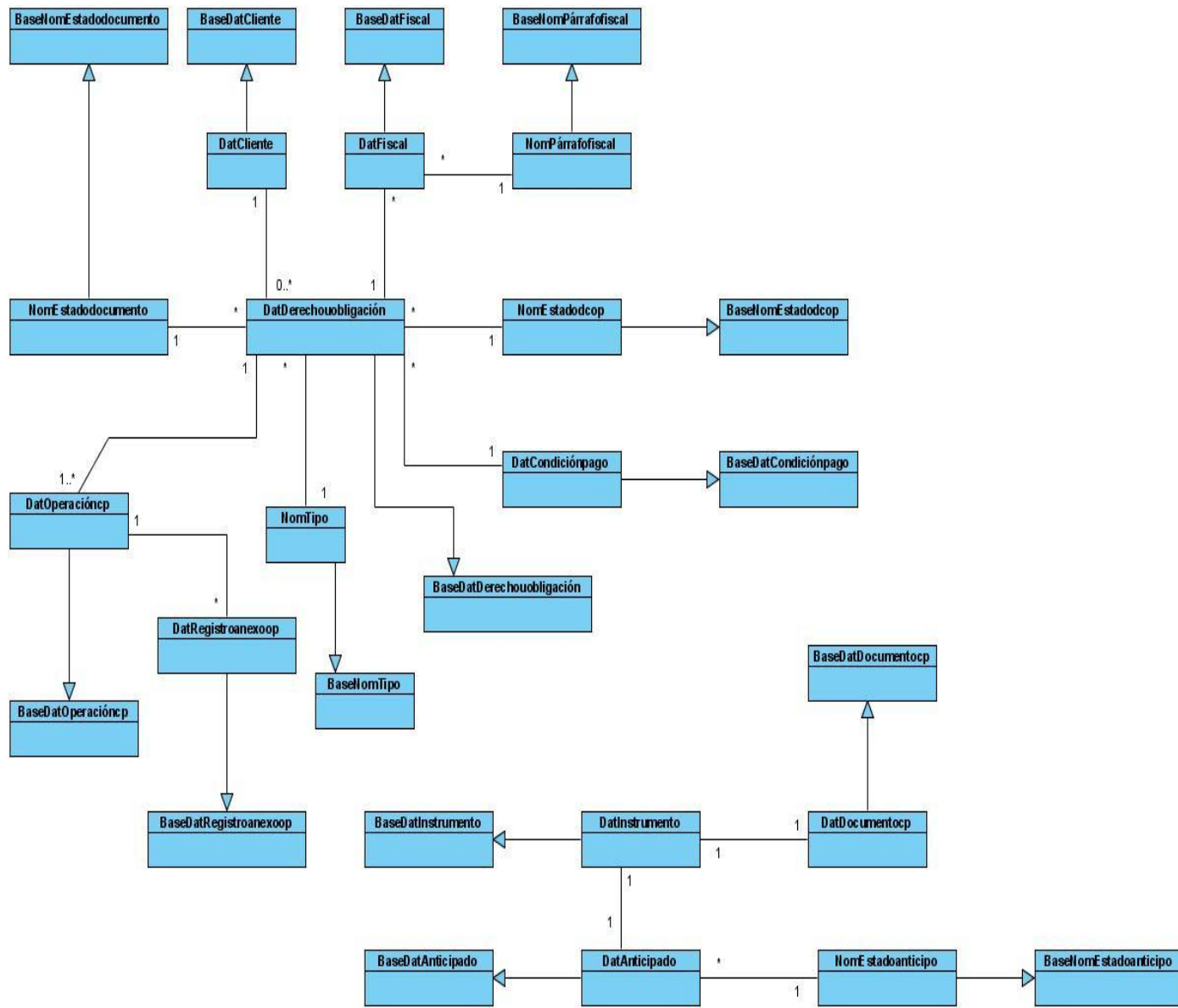


Figura 7. Diagrama de clases del paquete de dominio del componente Carga inicial

Diagrama de clases del diseño del componente Submayor.

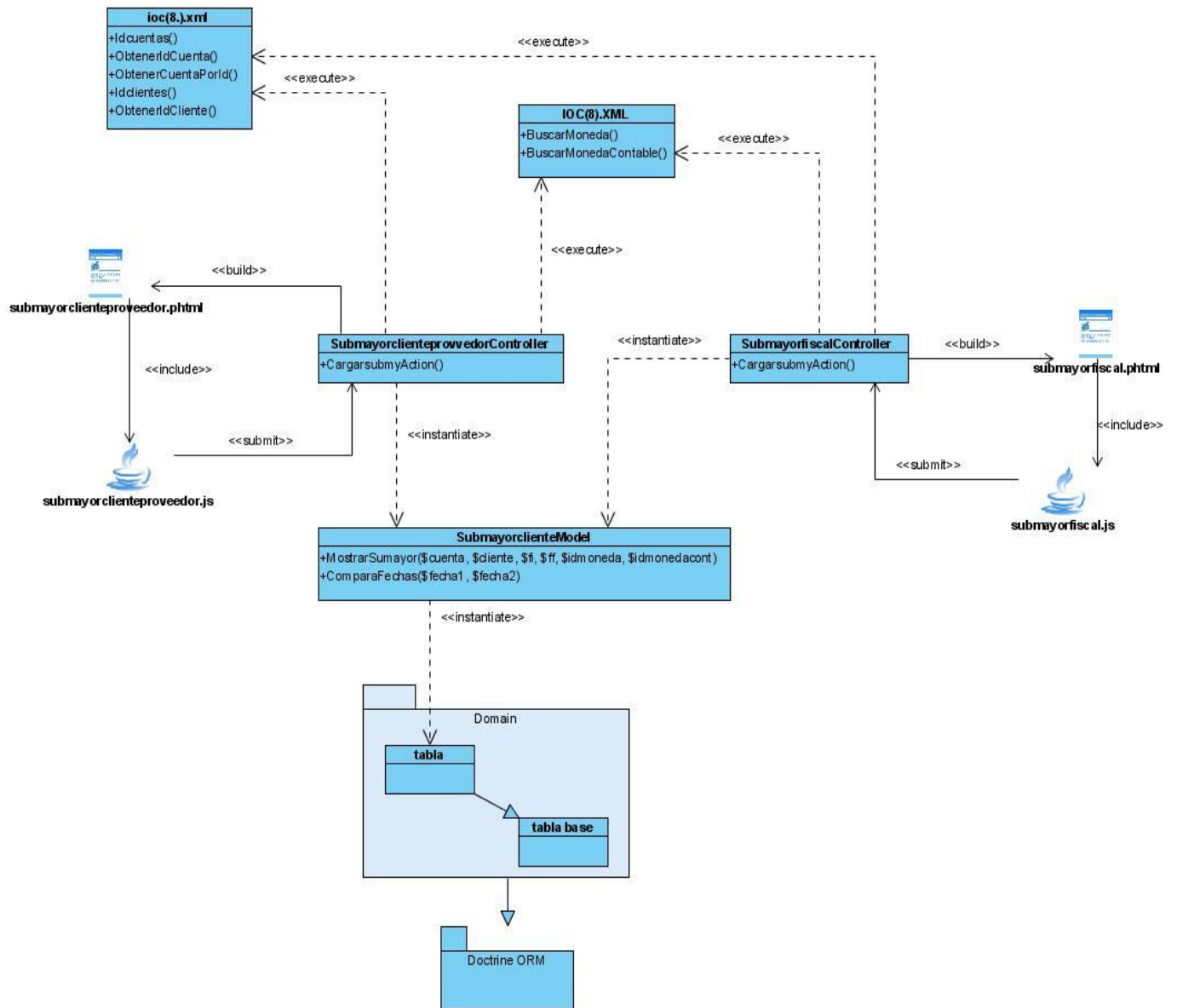


Figura 8. Diagrama de clases del diseño del componente Submayor.

Diagrama de clases del paquete de dominio del componente Submayor.



Figura 9. Diagrama de clases del paquete de dominio del componente Submayor.

Diagrama de clases del diseño Cambio de cuenta del componente Funcionalidades.

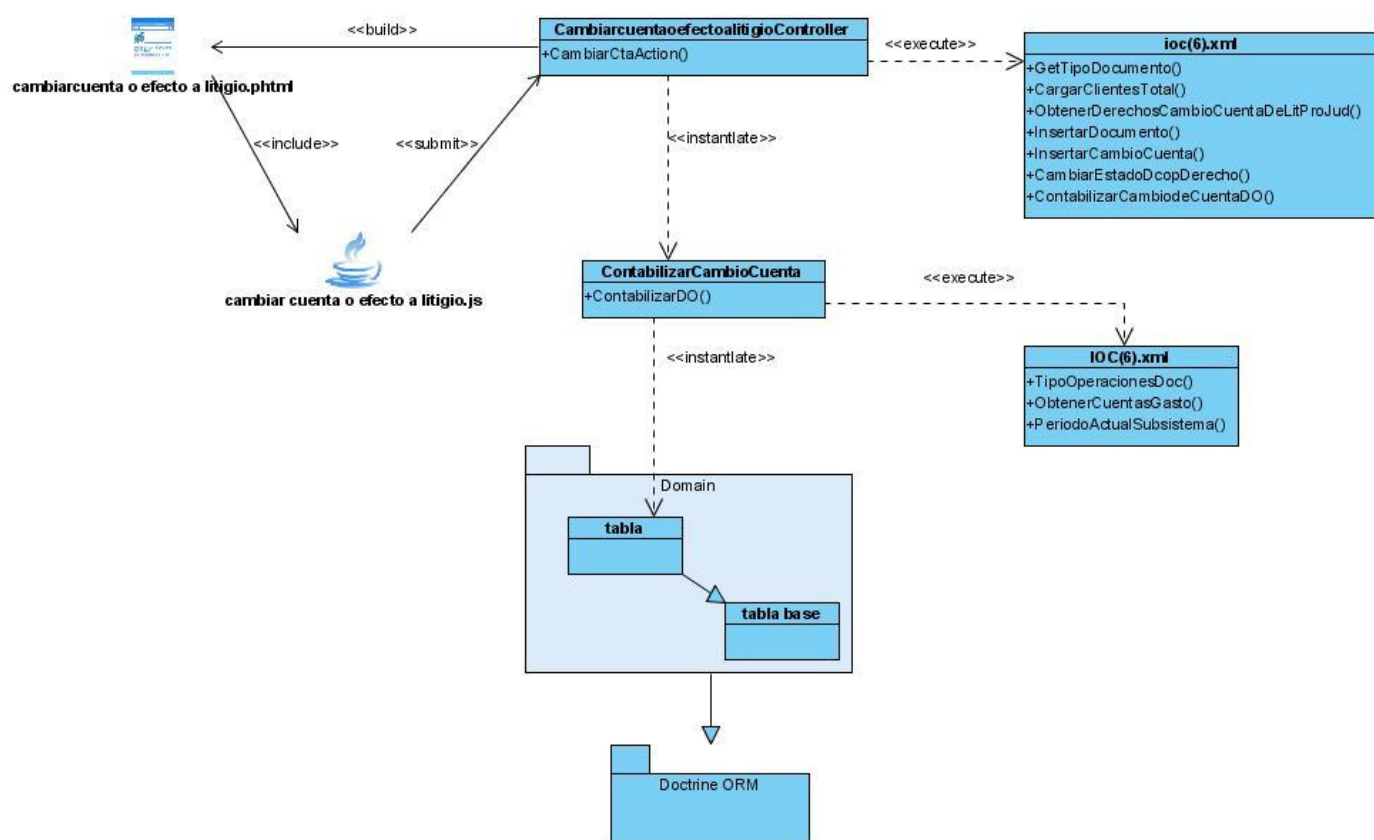


Figura 10. Diagrama de clases Cambio de cuenta del componente Funcionalidades.

Diagrama de clases del paquete de dominio Cambio de cuenta del componente Funcionalidades.

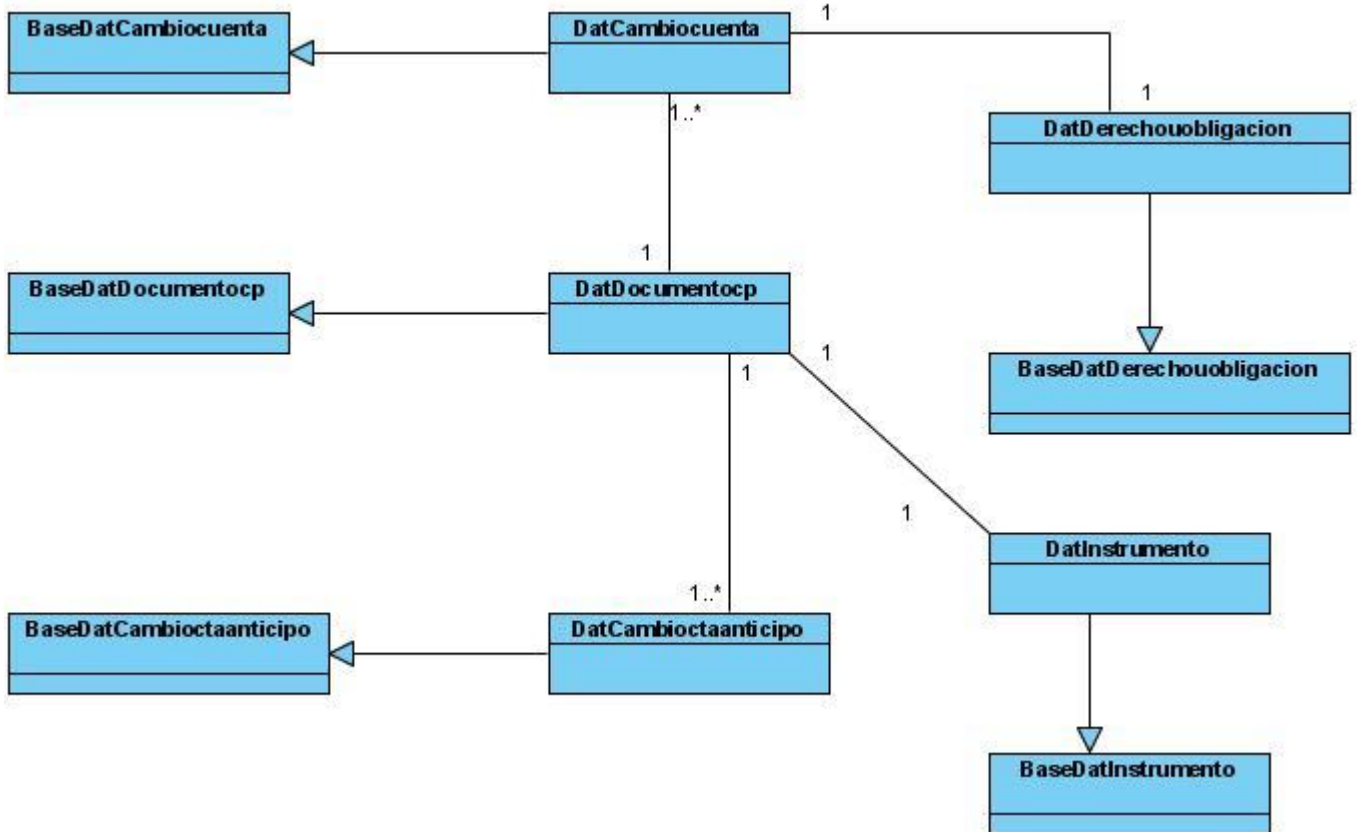


Figura 11. Diagrama del paquete de dominio Cambio de cuenta del componente Funcionalidades.

Diagrama de clases del diseño Cambio de moneda del componente Funcionalidades.

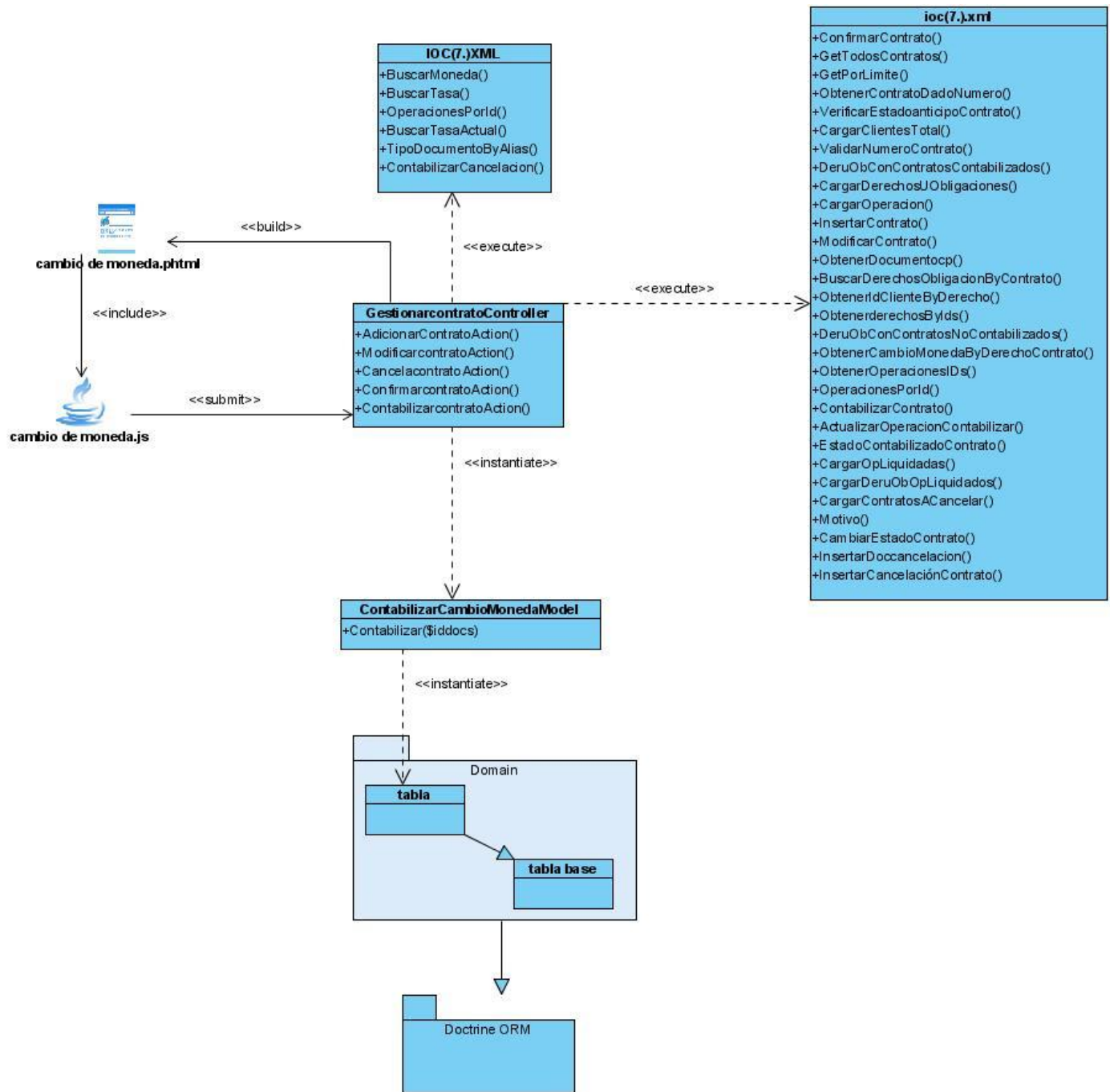


Figura 12. Diagrama de clases del diseño Cambio de moneda del componente de Funcionalidades.

Diagrama de clases del paquete de dominio Cambio de moneda del componente Funcionalidades.

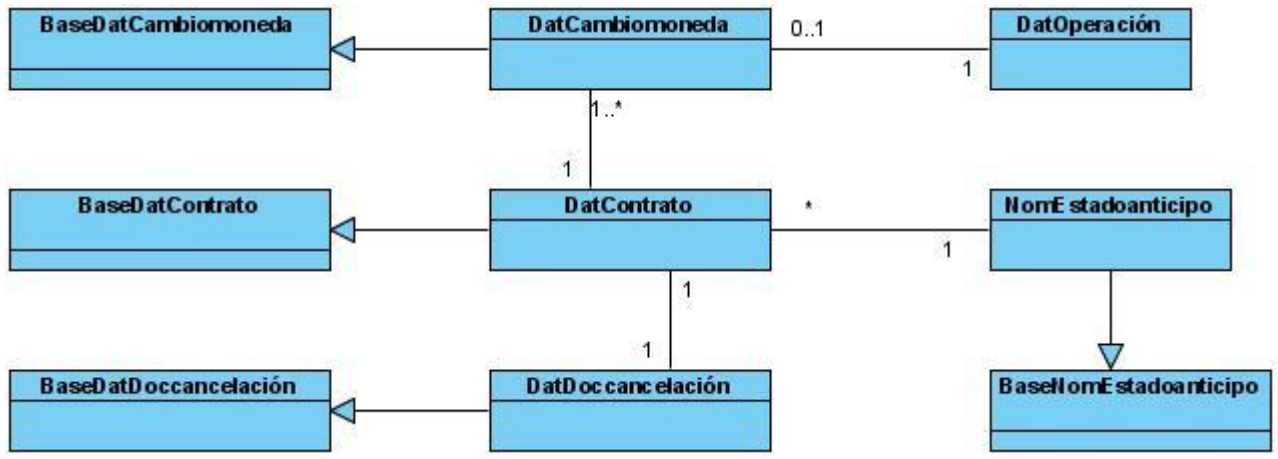


Figura 13. Diagrama de clases del paquete de dominio Cambio de moneda del componente Funcionalidades.

Diagrama de clases del diseño Liquidación del componente Funcionalidades.

El diagrama de clases de Liquidación del componente Funcionalidades se dividió en dos diagramas debido a la complejidad que este presente, con el objetivo de lograr un mayor entendimiento del mismo.

➤ Liquidación de anticipos.

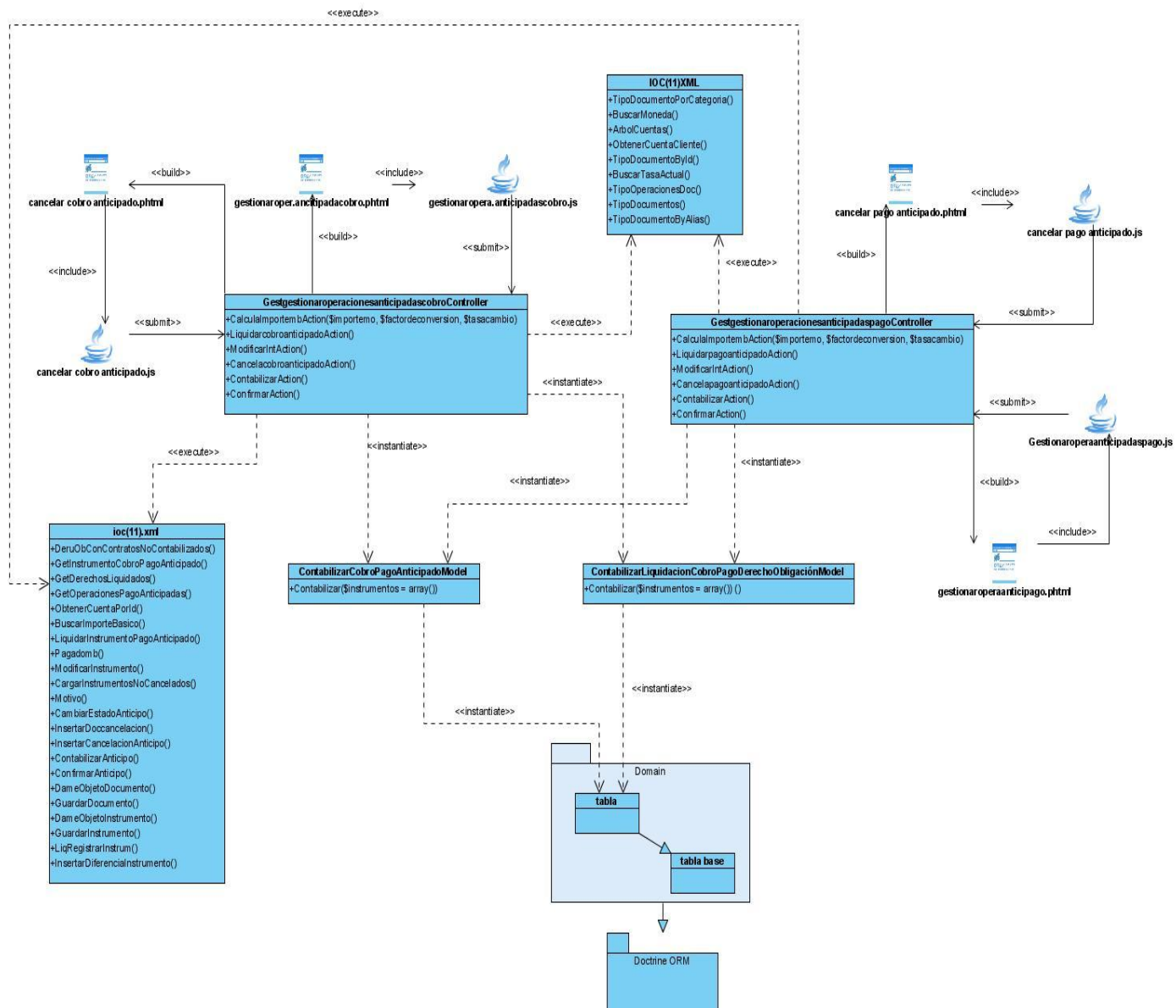


Figura 14. Diagrama de clases del diseño Liquidación de anticipos del componente Funcionalidades.

➤ Liquidación de derecho_obligación.

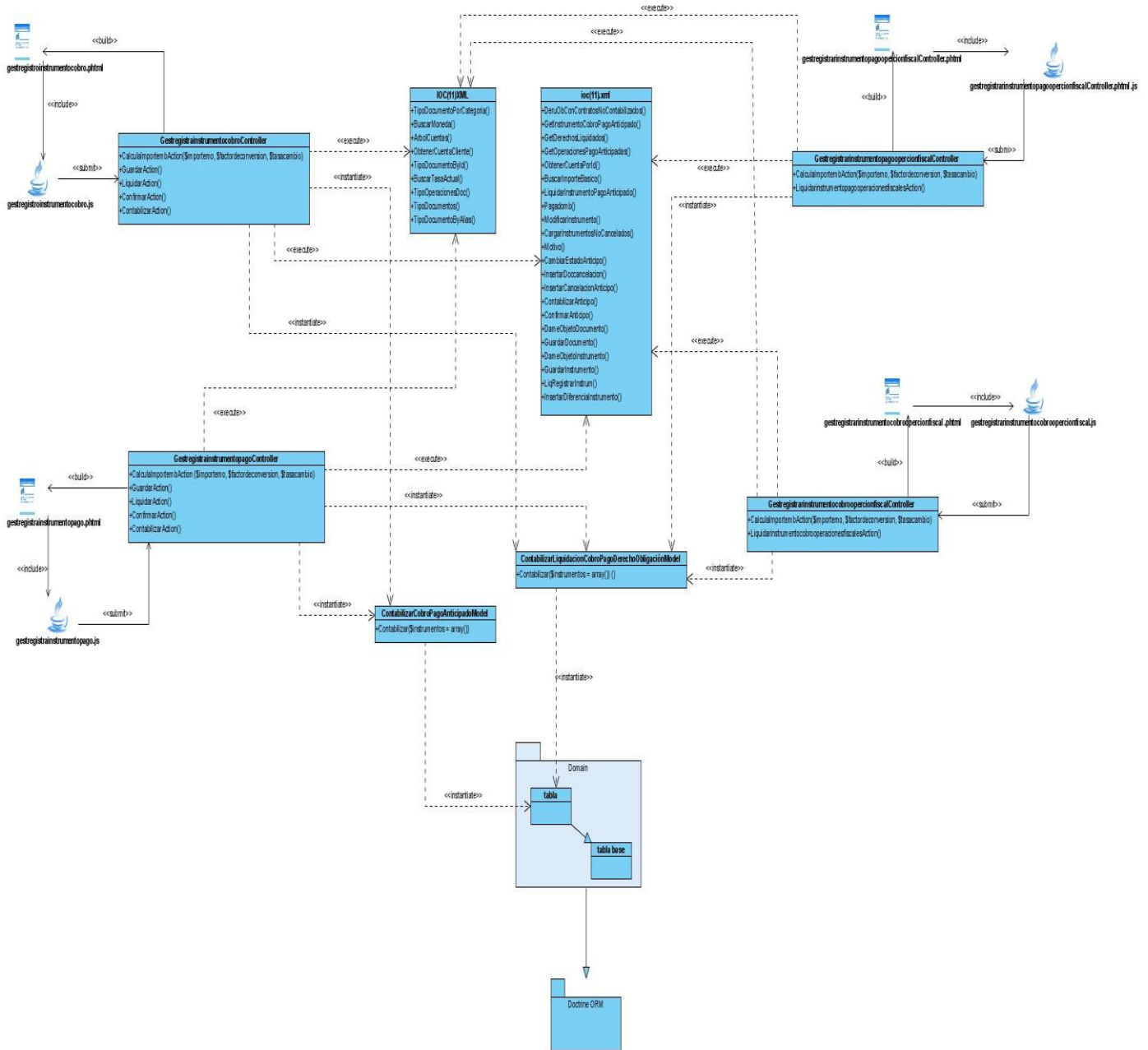


Figura 15. Diagrama de clases del diseño Liquidación de derecho_obligación del componente Funcionalidades.

Diagrama de clases del paquete de dominio Liquidación del componente Funcionalidades.

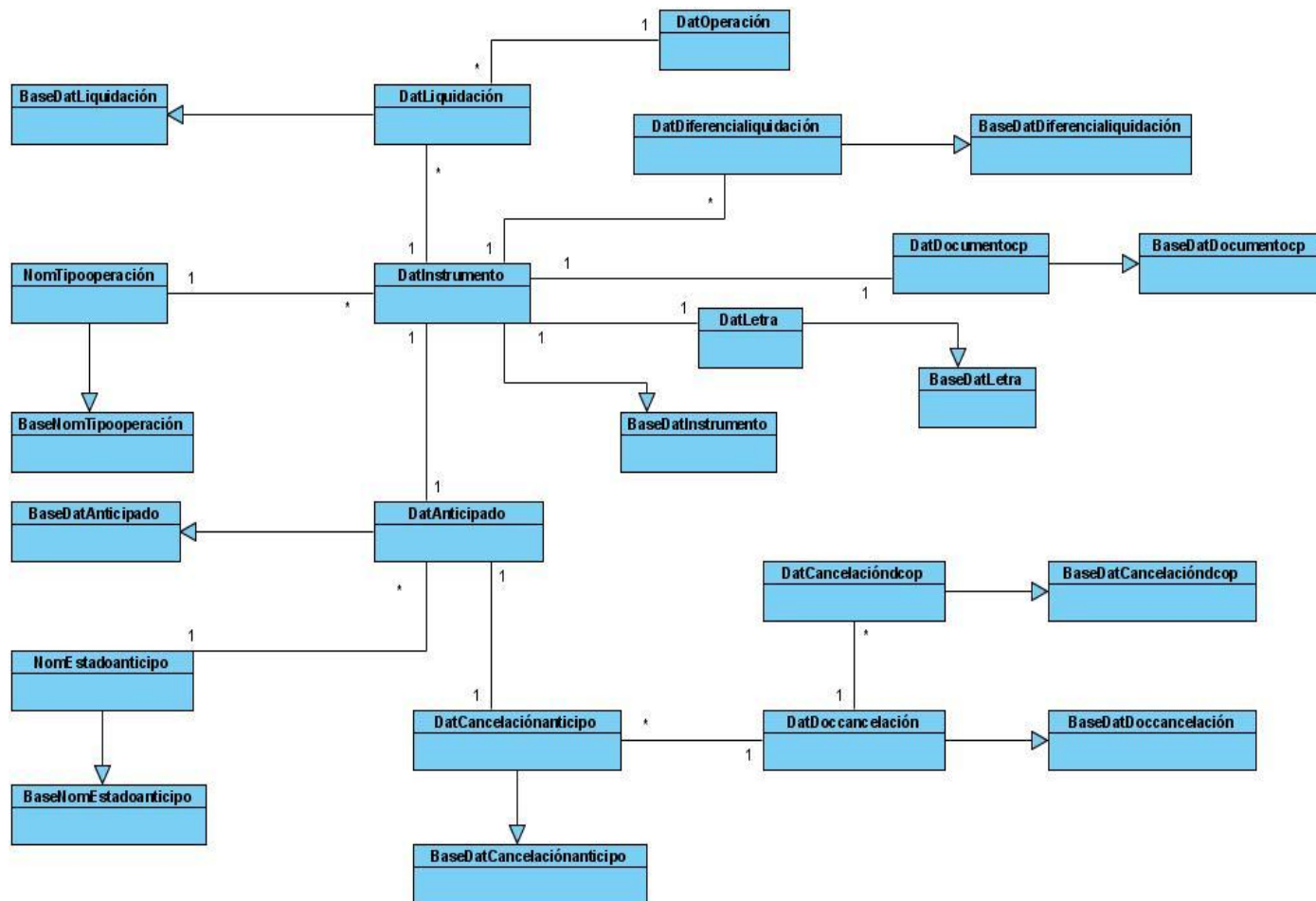


Figura 16. Diagrama de clases del paquete de dominio Liquidación del componente Funcionalidades.

Diagrama de clases del diseño del componente Derecho Obligación.

Los analistas del módulo Cobros y Pagos debido a la complejidad de este componente dividieron el diagrama de clases de diseño del mismo en los siguientes diagramas:

- Diagrama de clases del diseño de derecho de cobro.
- Diagrama de clases del diseño de derecho fiscal.
- Diagrama de clases del diseño de obligación de pago.
- Diagrama de clases del diseño de obligación fiscal.

Diagrama de clases del diseño de Derecho de cobro del componente Derecho_Obligación.

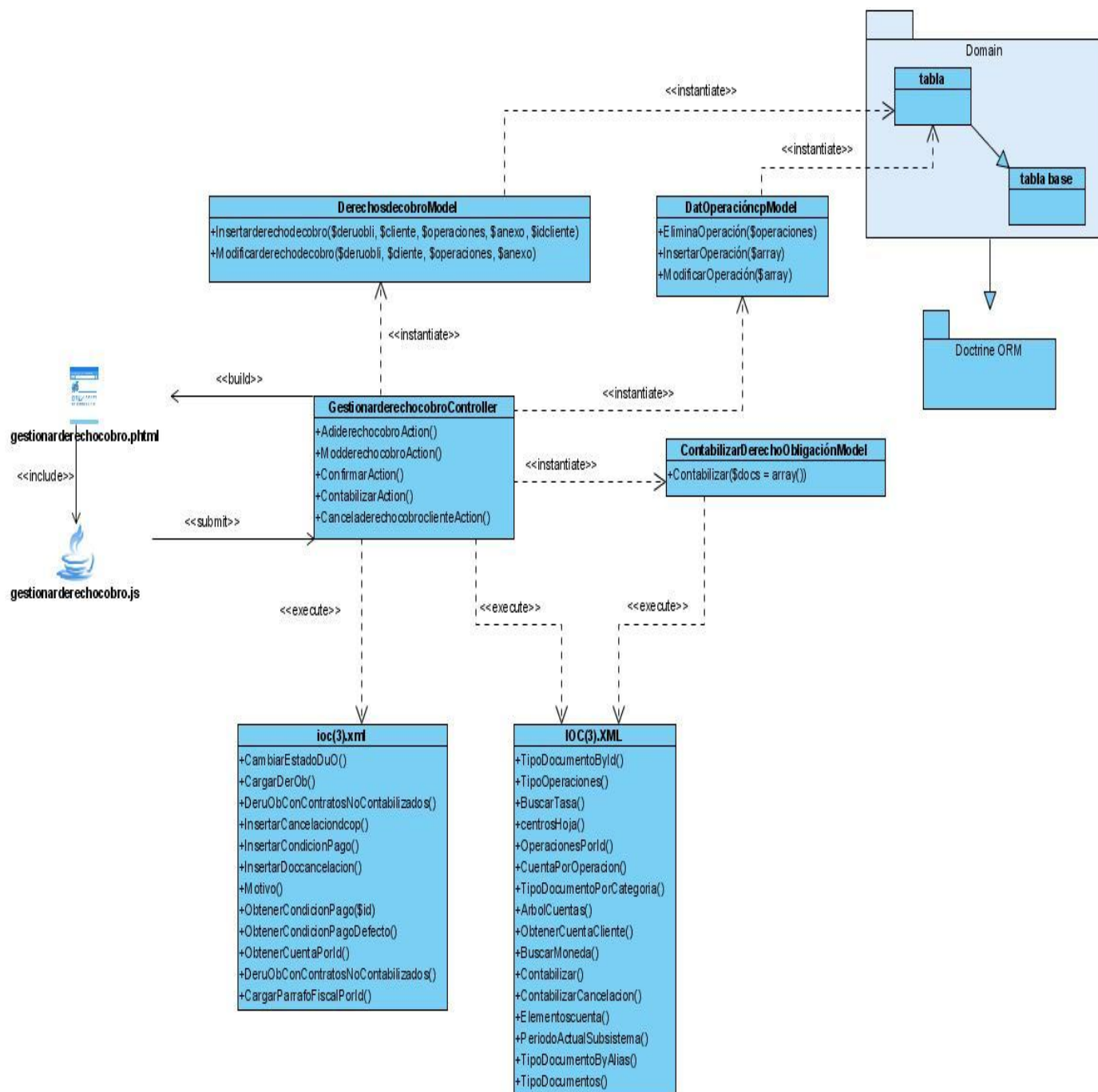


Figura 17. Diagrama de clases del diseño de Derecho de cobro del componente Derecho_Obligación.

Diagrama de clases del diseño de Derecho fiscal del componente Derecho_Obligación

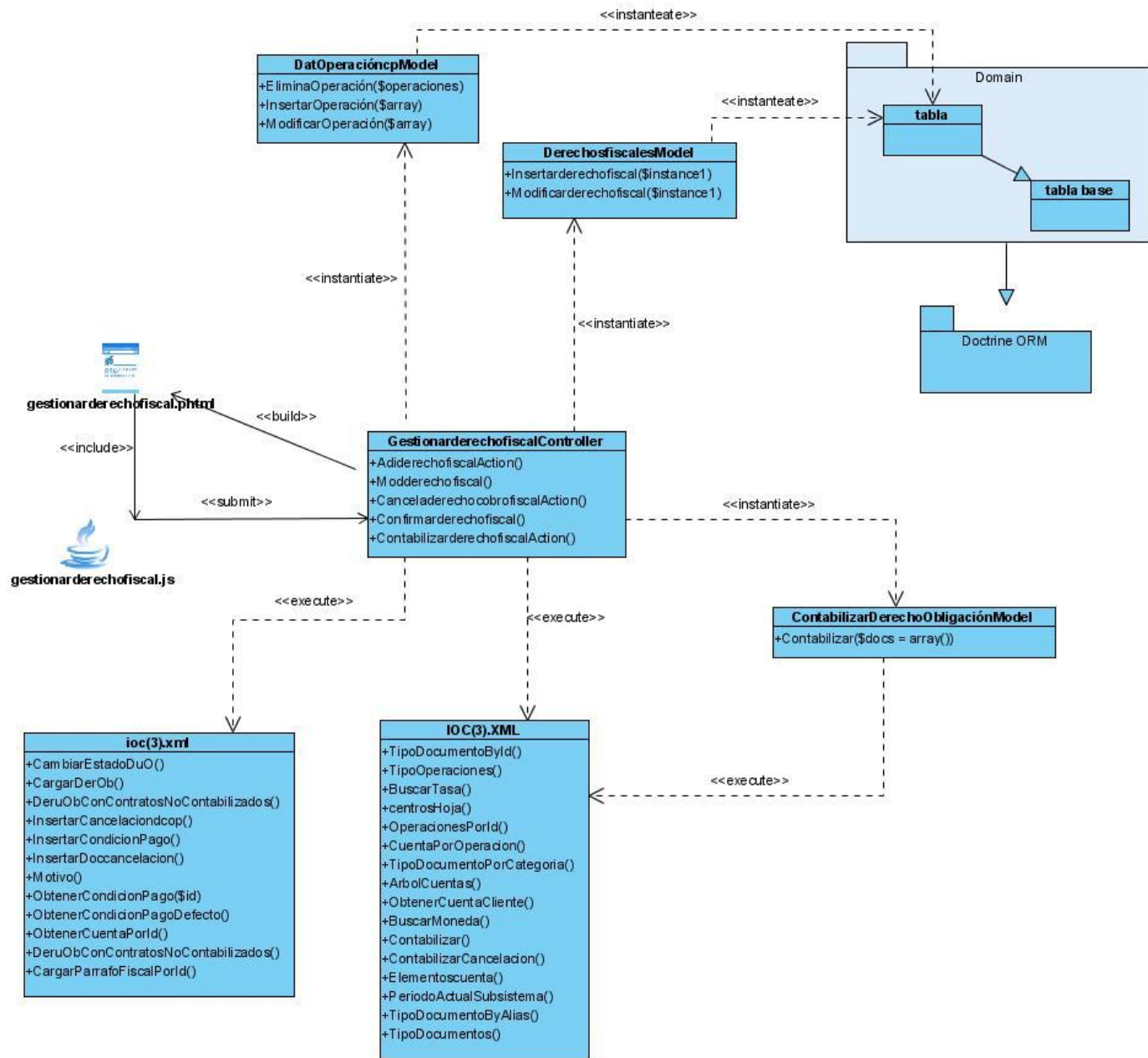


Figura 18. Diagrama de clases del diseño de Derecho fiscal del componente Derecho_Obligación

Diagrama de clases del diseño de Obligación de pago del componente Derecho_Obligación

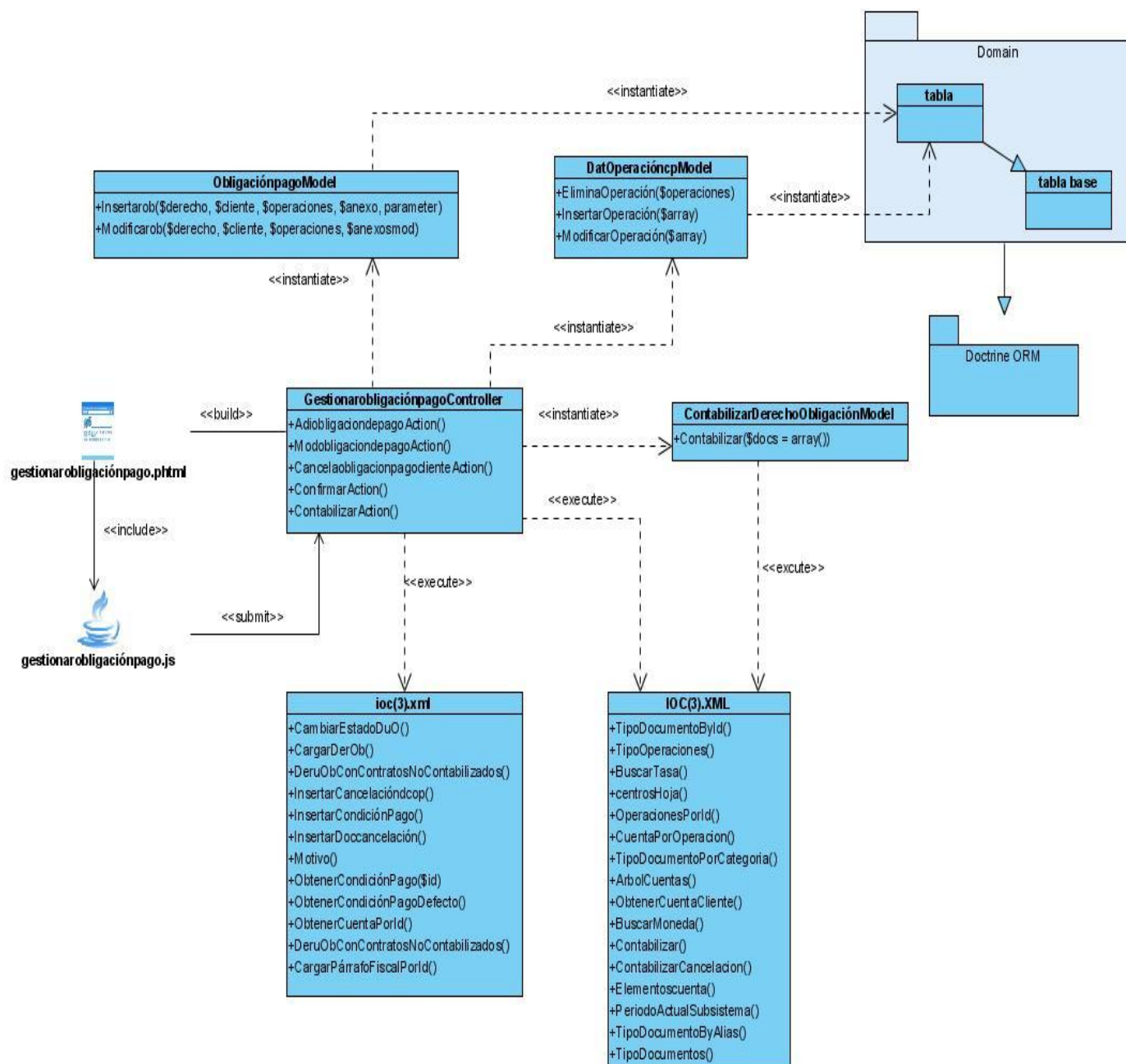


Figura 19. Diagrama de clases del diseño de Obligación de pago del componente Derecho_Obligación.

Diagrama de clases del diseño de Obligación fiscal del componente Derecho_Obligación

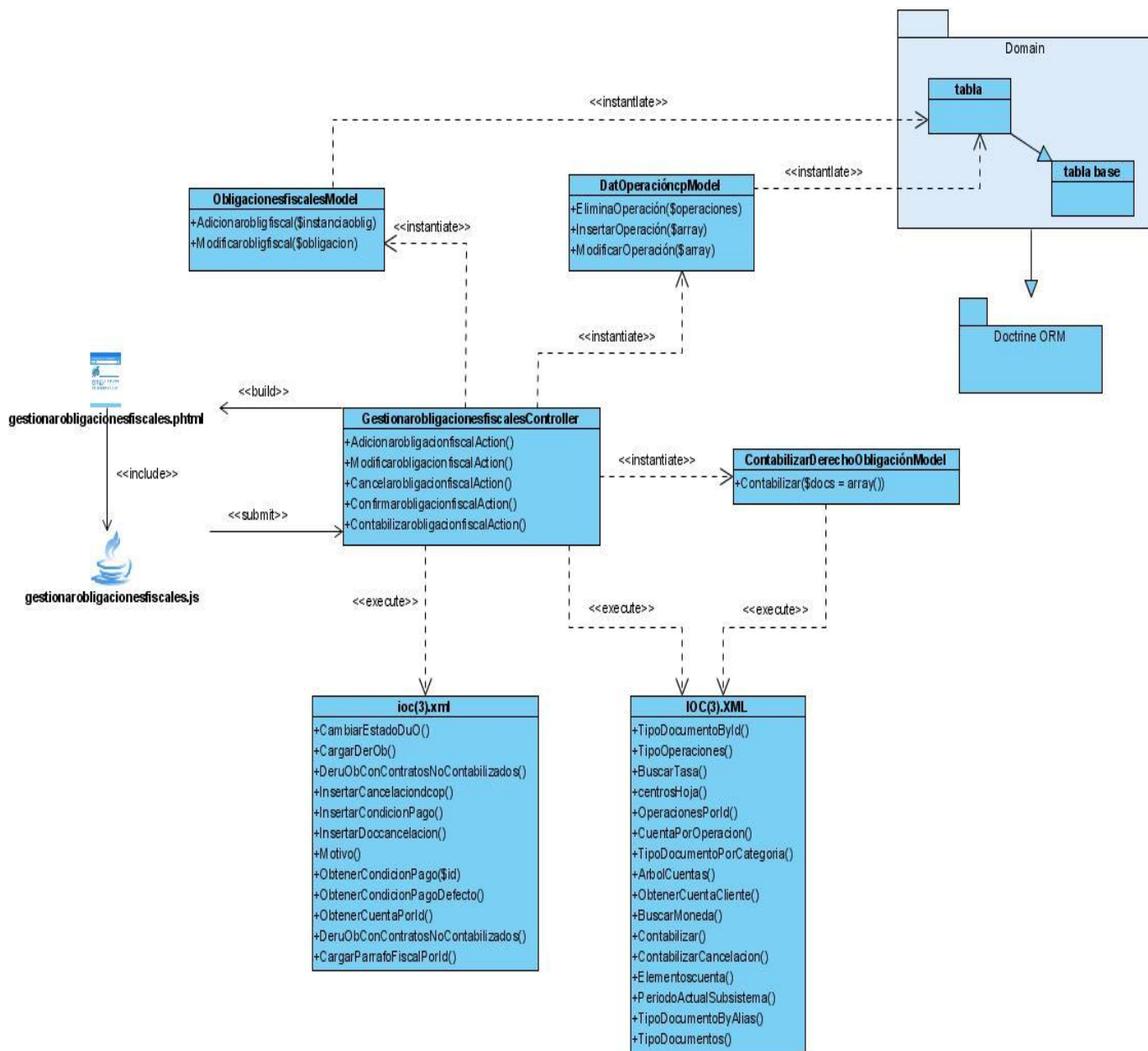


Figura 20. Diagrama de clases del diseño de Obligación fiscal del componente Derecho_Obligación.

Diagrama de clases del paquete de dominio del componente Derecho_Obligación.

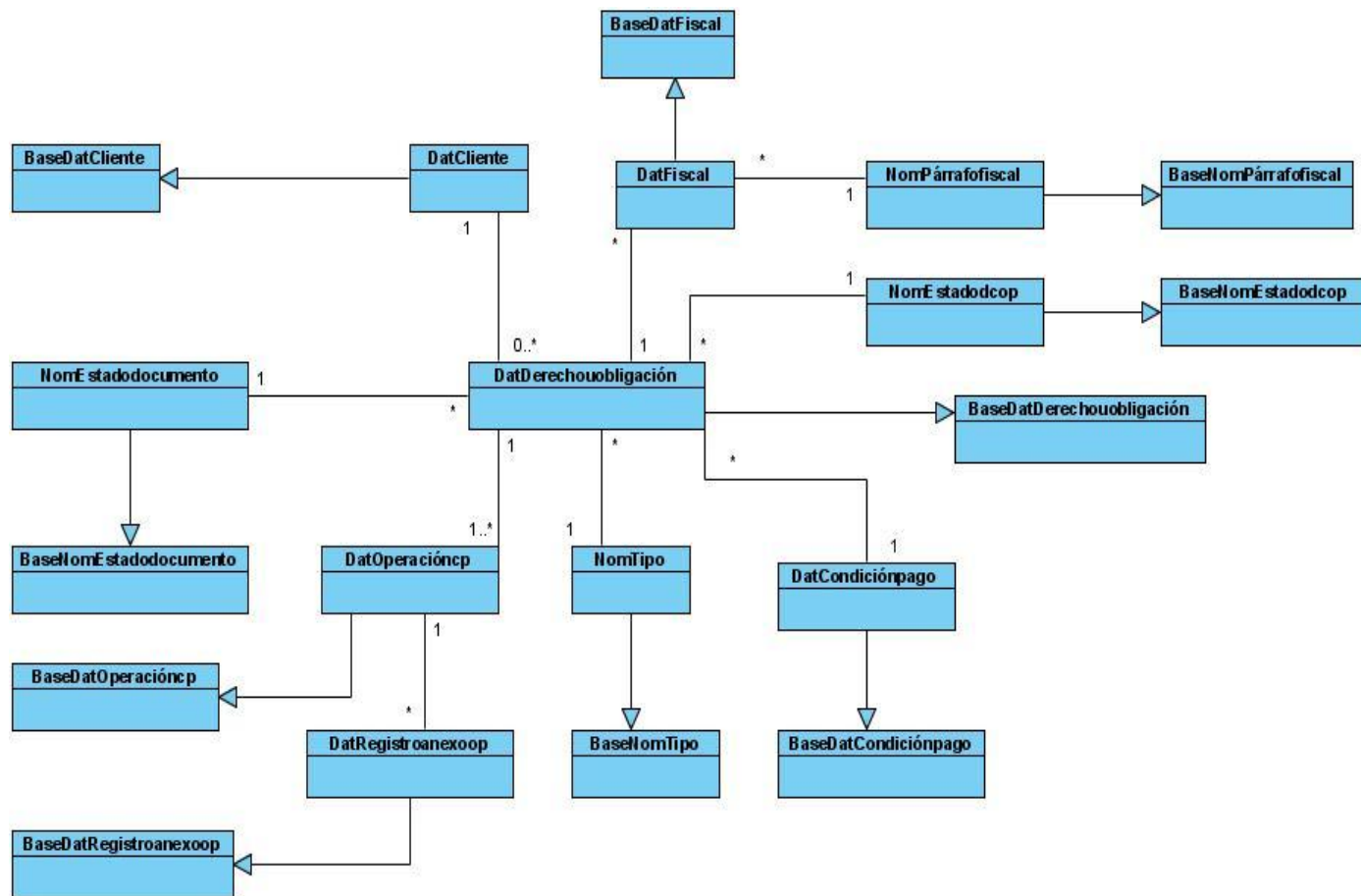


Figura 21. Diagrama de clases del paquete de dominio del componente Derecho_Obligación.

3.5. Descripción de las clases del diseño.

En el módulo Cobros y Pagos se describieron todas las clases del diseño identificadas, con el objetivo de lograr un mayor entendimiento del funcionamiento de las mismas.

Descripción de las clases del diseño del componente Conciliación.

Clases controladoras del componente Conciliación:

Nombre:	AdicionarconciliacióndeanticiposController
Tipo de clase:	Controladora
Atributo	Tipo
Para cada responsabilidad:	
Nombre:	Descripción
AdicionarconciliaciónAction()	Permite adicionar una conciliación de anticipo obteniendo mediante POST los parámetros de la conciliación y enviándolos a las clases del negocio DatRefanticipoModel y DatConciliaciónModel

Nombre:	AdicionarconciliacióndeobligacionesoderechosController
Tipo de clase:	Controladora
Atributo	Tipo
Para cada responsabilidad:	
Nombre:	Descripción
AdicionarconciliaciónAction()	Permite adicionar una conciliación de derecho u obligación obteniendo mediante POST los parámetros de la conciliación y enviándolos a las clases del negocio DatRefderechouobligaciónModel y DatConciliaciónModel.

Clases del negocio del componente Conciliación:

Nombre:	DatConciliaciónModel
Tipo de clase:	Modelo
Atributo	Tipo
Para cada responsabilidad:	
Nombre:	Descripción
InsertarConciliación(\$obj_conc)	Permite guardar en la base datos la nueva conciliación pasada como parámetro.

Nombre:	DatRefantipicoModel
Tipo de clase:	Modelo
Atributo	Tipo
Para cada responsabilidad:	
Nombre:	Descripción
Insertrefantipico(\$obj_ref)	Permite guardar en la base de datos una conciliación de anticipo que es pasada como parámetro en el método

Nombre:	DatRefderechouobligacionModel
Tipo de clase:	Modelo
Atributo	Tipo
Para cada responsabilidad:	
Nombre:	Descripción
Insertrefderecho(\$obj_ref)	Permite guardar en la base de datos la conciliación de derecho u obligación pasada como parámetro.

Clases del paquete de dominio del componente Conciliación:

Nombre:	DatConciliación
Tipo de clase:	Entidad
Descripción	
Permite el acceso a la tabla de la base de datos DatConciliación.	

Nombre:	DatRefderechouobligación
Tipo de clase:	Entidad
Descripción	
Permite el acceso a la tabla de la base de datos DatRefderechouobligación, la cual contiene información referente a las conciliaciones de derechos u obligaciones registradas.	

Nombre:	NomOperaciónconciliación
Tipo de clase:	Entidad
Descripción	
Permite el acceso a la tabla de la base de datos NomOperaciónconciliación, que es un nomenclador de las operaciones de conciliación.	

Nombre:	DatRefanticipo
Tipo de clase:	Entidad
Descripción	
Permite el acceso a la tabla de la base de datos DatRefanticipo, que contiene la información correspondiente a una conciliación de anticipo.	

Descripción de las clases del diseño del componente Configuración.

Clases controladoras del componente Configuración.

Nombre:	CondiciónpagoController
Tipo de clase:	Controladora
Atributo	Tipo
Para cada responsabilidad:	
Nombre:	Descripción
AdicionarcondiciónAction()	Permite adicionar una nueva condición de pago, obteniendo los datos de la misma por POST y enviándolos a la clase del negocio CondiciónpagoModel.

Nombre:	GestmotivocancelaciónController
Tipo de clase:	Controladora
Atributo	Tipo
Para cada responsabilidad:	
Nombre:	Descripción
AdicionarAction()	Permite obtener los parámetros por POST para crear un objeto de la clase NomMotivocancelación y enviarlo para la clase del negocio GestmotivocancelaciónModel
ModificarAction()	Permite obtener los parámetros por POST para cambiar el valor de la

	descripción en un objeto de NomMotivocancelación y enviarlo para la clase del negocio GestmotivocancelaciónModel
EliminarmotivocancelaciónAction()	Permite obtener por POST el id del motivo de cancelación, el cual se envía para la clase del negocio GestmotivocancelaciónModel y se elimina si no tiene dependencia relacional.

Nombre:	PárrafoController
Tipo de clase:	Controladora
Atributo	Tipo
Para cada responsabilidad:	
Nombre:	Descripción
InsertarpárrafofiscalAction()	Permite obtener los parámetros por POST para crear un objeto de la clase NomPárrafofiscal y enviarlo para la clase del negocio PárrafoModel que se encarga de insertar el nuevo párrafo fiscal.
ModificarpárrafofiscalAction()	Permite obtener los parámetros por POST para cambiar el valor de la descripción y el código de un párrafo fiscal y enviar los nuevos valores para la clase del negocio PárrafoModel
EliminarpárrafofiscalAction()	Permite obtener el id del párrafo fiscal a eliminar, mediante POST, se verifica que el párrafo no este asociado a otras operaciones y se envía el id del párrafo a la clase PárrafoModel para ser eliminado.

Clases del negocio del componente Configuración.

Nombre:	CondiciónpagoModel
Tipo de clase:	Modelo
Atributo	Tipo
Para cada responsabilidad:	

Nombre:	Descripción
Adicionarcondición(\$obj)	Permite adicionar una condición de pago pasando como parámetro el objeto a guardar.
ModificarCondiciónPago (\$idcond,\$dpp, \$mora, \$plazo, \$idestructuracomun)	Permite modificar una condición de pago pasando como parámetros los nuevos datos y el identificador de la condición que se va a modificar.

Nombre:	MotivoCancelaciónModel
Tipo de clase:	Modelo
Atributo	Tipo
Para cada responsabilidad:	
Nombre:	Descripción
AdicionarMotivoCancelación(NomMotivocancelación \$motivocancelación)	Permite adicionar un nuevo motivo de cancelación pasando como parámetro un objeto del mismo que es guardado en la base de datos
EliminarMotivoCancelación(\$motivocancelación)	Permite eliminar un motivo de cancelación, recibiendo como parámetro un objeto para ser eliminado de la base de datos

Nombre:	parrafoModel
Tipo de clase:	
Atributo	Tipo
Para cada responsabilidad:	
Nombre:	Descripción
InsertarPárrafoFiscal(\$instance)	Permite insertar un nuevo párrafo fiscal recibiendo como parámetro un objeto que es guardado en la base de datos.
ModificarPárrafoFiscal(\$instance)	Permite modificar un párrafo fiscal pasando como parámetro el objeto con los datos modificados y guardando el mismo en la base de datos.
EliminarPárrafoFiscal(\$instance)	Permite eliminar un párrafo fiscal de la base de datos, pasando como parámetro una instancia del párrafo fiscal que se desea eliminar.

Clases del paquete de dominio del componente Configuración.

Nombre:	DatCondiciónpago
Tipo de clase:	Entidad
Descripción	
Permite el acceso a la tabla de la base de datos DatCondiciónpago, que contiene la información correspondiente a la condición de pago establecida para los derechos y las obligaciones.	

Nombre:	NomMotivocancelación
Tipo de clase:	Entidad
Descripción	
Permite el acceso a la tabla de la base de datos NomMotivocancelación, que es un nomenclador que contiene los posibles motivos por los cuales se puede cancelar un documento.	

Nombre:	NomPárrafofiscal
Tipo de clase:	Entidad
Descripción	
Permite el acceso a la tabla de la base de datos NomPárrafofiscal, que es un nomenclador que contiene los párrafos fiscales definidos.	

Descripción de las clases del diseño del componente Cierre.

Clases Controladores del componente Cierre.

Nombre:	CierrecontableController
Tipo de clase:	Controladora
Atributo	Tipo
Para cada responsabilidad:	
Nombre:	Descripción
CerrarperiodoAction()	Permite realizar el cierre de periodo encargándole a la clase del negocio CierreContableModel realizar el cierre de periodo y envía a la vista cierrecontable.phtml los reportes correspondientes del cierre de periodo para ser mostrados.
CerrarejercicioAction ()	Permite realizar el cierre de ejercicio encargándole a la clase del negocio CierreContableModel realizar el cierre de ejercicio y envía a la vista cierrecontable.phtml los reportes correspondientes al cierre de ejercicio para ser mostrados

Clases del negocio del componente Cierre.

Nombre:	CierreContableModel
Tipo de clase:	Modelo
Atributo	Tipo
Para cada responsabilidad:	
Nombre:	Descripción
CerrarPeriodo()	Se encarga de realizar el cierre de periodo utilizando servicios del ioc.xml
CerrarEjercicio()	Se encarga de realizar el cierre de ejercicio utilizando servicios del ioc.xml

Descripción de las clases del diseño del componente Carga inicial.

Clases controladoras del componente Carga inicial.

Nombre:	AperturaController
Tipo de clase:	Controladora
Atributo	Tipo
Para cada responsabilidad:	
Nombre:	Descripción
ComenzarAction()	Permite enviar la petición de comenzar el ejercicio a la clase del negocio CargainicialModel y captura los reportes de la apertura para ser mostrados en la vista apertura.phtml.

Nombre:	ObligacióndepagController
Tipo de clase:	Controladora
Atributo	Tipo
Para cada responsabilidad:	
Nombre:	Descripción
AdicionarobligacionpagoAction()	Permite obtener mediante POST los atributos de la obligación de pago y los envía a la clase del negocio CargainicialModel para adicionar una nueva obligación de pago, y luego mostrar un mensaje con los resultados de la operación.
ModificarobligaciónpagoAction()	Permite obtener mediante POST los nuevos atributos de la obligación de pago y los envía a la clase del negocio CargainicialModel para modificar una obligación de pago, y luego mostrar un mensaje con los resultados de la operación.
EliminarobligaciónpagoAction()	Permite obtener mediante POST el identificador de la obligación de pago que se desea eliminar y lo envía a la clase CargainicialModel para eliminar la obligación de pago correspondiente.
CuentaaperturaclienteAction()	Permite obtener mediante POST los campos cuenta y cliente y verifica si el cliente tiene apertura para esa cuenta.

Nombre:	DerechofiscalController
Tipo de clase:	Controladora
Atributo	Tipo
Para cada responsabilidad:	
Nombre:	Descripción
AdicionarderechofiscalAction()	Permite obtener mediante POST los atributos del derecho fiscal y los envía a la clase del negocio CargainicialModel para adicionar un nuevo

	derecho fiscal, y luego mostrar un mensaje con los resultados de la operación.
ModificarobligaciónfiscalAction()	Permite obtener mediante POST los nuevos atributos del derecho fiscal y los envía a la clase del negocio CargainicialModel para modificar un derecho fiscal, y luego mostrar un mensaje con los resultados de la operación.
eliminarobligacionfiscalAction() ()	Permite obtener mediante POST el identificador del derecho fiscal que se desea eliminar y lo envía a la clase CargainicialModel para eliminar el derecho fiscal correspondiente.

Nombre:	DerechodecobroController
Tipo de clase:	Controladora
Atributo	Tipo
Para cada responsabilidad:	
Nombre:	Descripción
AdicionarderechocobroAction()	Permite obtener mediante POST los atributos del derecho de cobro y los envía a la clase del negocio CargainicialModel para adicionarlo, y luego mostrar un mensaje con los resultados de la operación.
ModificarderechocobroAction()	Permite obtener mediante POST los nuevos atributos del derecho de cobro y los envía a la clase del negocio CargainicialModel para modificarlo, y luego mostrar un mensaje con los resultados de la operación.
EliminarderechocobroAction()	Permite obtener mediante POST el identificador del derecho de cobro que se desea eliminar y lo envía a la clase CargainicialModel para eliminarlo.
CuentaaperturaaporclienteAction()	Permite obtener mediante POST los campos cuenta y cliente y verifica si el cliente tiene apertura para esa cuenta.

Nombre:	PagoanticipadoController
Tipo de clase:	Controladora
Atributo	Tipo
Para cada responsabilidad:	
Nombre:	Descripción
AdicionarPagoAction()	Permite obtener mediante POST los atributos del pago anticipado y los envía a la clase del negocio CargainicialModel para adicionarlo y luego mostrar un mensaje con los resultados de la operación.
ModificarPagoAction()	Permite obtener mediante POST los nuevos atributos del pago anticipado y los envía a la clase del negocio CargainicialModel para modificarlo, y luego mostrar un mensaje con los resultados de la operación.
EliminarPagoAction()	Permite obtener mediante POST el identificador del pago anticipado que se desea eliminar y lo envía a la clase CargainicialModel para eliminarlo.
CuentaAperturaPorClienteAction()	Permite obtener mediante POST los campos cuenta y cliente y verifica si el cliente tiene apertura para esa cuenta.

Nombre	ObligaciónfiscalController
Tipo de clase:	Controladora
Atributo	Tipo
Para cada responsabilidad:	
Nombre:	Descripción
AdicionarobligaciónfiscalAction()	Permite obtener mediante POST los atributos de la obligación fiscal y los envía a la clase del negocio CargainicialModel para adicionarla, y luego mostrar un mensaje con los resultados de la operación.
ModificarobligaciónfiscalAction()	Permite obtener mediante POST los nuevos atributos de la obligación fiscal y los envía a la clase del negocio CargainicialModel para modificarla, y luego mostrar un mensaje con los resultados de la operación.
EliminarobligaciónfiscalAction()	Permite obtener mediante POST el identificador de la obligación fiscal que se desea eliminar y lo envía a la clase CargainicialModel para eliminarlo.

Nombre:	CobroanticipadoController
Tipo de clase:	Controladora
Atributo	Tipo
Para cada responsabilidad:	
Nombre:	Descripción
AdicionarcobroAction()	Permite obtener mediante POST los atributos del cobro anticipado y los envía a la clase del negocio CargainicialModel para adicionarlo, y luego mostrar un mensaje con los resultados de la operación.
ModificarcobroAction()	Permite obtener mediante POST los nuevos atributos del cobro anticipado y los envía a la clase del negocio CargainicialModel para modificarlo, y luego mostrar un mensaje con los resultados de la

	operación.
EliminarcobroAction()	Permite obtener mediante POST el identificador del cobro anticipado que se desea eliminar y lo envía a la clase CargainicialModel para eliminarlo.
CuentaaperturaaporclienteAction()	Permite obtener mediante POST los campos cuenta y cliente y verifica si el cliente tiene apertura para esa cuenta.

Clases del negocio del componente Carga inicial.

Nombre:	CargainicialModel
Tipo de clase:	Modelo
Atributo	Tipo
Para cada responsabilidad:	
Nombre:	Descripción
AdicionarAnticipo(\$array)	Permite guardar en la base de datos un cobro o pago anticipado, cuyos datos son recibidos en un arreglo que es pasado como parámetro en el método.
ModificarAnticipo(\$id,\$array)	Permite guardar en la base de datos los cambios realizados a un cobro o pago anticipado, cuyos identificadores y nuevos datos son recibidos como parámetros en el método.
EliminarAnticipo(\$id)	Permite eliminar de la base de datos un pago o cobro anticipado cuyo identificador es pasado como parámetro en el método.
AdicionarDerechouObligacion(\$array)	Permite guardar en la base de datos un derecho de cobro o una obligación de pago, cuyos datos son recibidos en un arreglo que es pasado como parámetro en el método.

ModificarDerechouObligacion(\$array)	Permite guardar en la base de datos los cambios realizados de un derecho de cobro o a una obligación de pago, cuyos identificadores y nuevos datos son recibidos como parámetros en el método.
EliminarDerechouObligacion(\$idderechouobligacion)	Permite eliminar de la base de datos un derecho de cobro o una obligación de pago cuyo identificador es pasado como parámetro en el método.
AdicionarOperacion (\$array)	Permite adicionar una nueva operación a la base de datos recibiendo los datos de la operación en un arreglo que es pasado como parámetro.
ModificarOperacion(\$array)	Permite modificar una operación en base de datos recibiendo los datos de la operación en un arreglo que es pasado como parámetro.
EliminarOperacion(\$id)	Permite eliminar una operación de la base de datos recibiendo el identificador de la operación como parámetro del método.
CalcularSaldo(\$idcuenta)	Permite calcular el saldo de una cuenta que es pasada como parámetro del método.

Clases del dominio del componente Carga inicial.

Nombre:	DatCondiciónpago
Tipo de clase:	Entidad
Descripción	
Permite el acceso a la tabla de la base de datos DatCondiciónpago, que contiene la información correspondiente a la condición de pago de los derechos y las obligaciones.	

Nombre:	NomPárrafofiscal
Tipo de clase:	Entidad
Descripción	
Permite el acceso a la tabla de la base de datos NomPárrafofiscal, que es un nomenclador que contiene los párrafos fiscales definidos.	

Nombre:	NomEstadodcop
Tipo de clase:	Entidad
Descripción	
Permite el acceso a la tabla de la base de datos NomEstadodcop, que es un nomenclador que contiene los estados que puede tener un documento de cobros y pagos.	

Nombre:	NomTipo
Tipo de clase:	Entidad
Descripción	
Permite el acceso a la tabla de la base de datos NomTipo, que es un nomenclador que contiene los tipos de derechos y obligaciones que existen.	

Nombre:	DatOperación
Tipo de clase:	Entidad
Descripción	
Permite el acceso a la tabla de la base de datos DatOperación, que contiene la información referente a las operaciones asociadas a los derechos y obligaciones.	

Nombre:	DatCliente
Tipo de clase:	Entidad
Descripción	
Permite el acceso a la tabla de la base de datos DatCliente, que contiene la información referente a los clientes que realizan operaciones de cobros y pagos.	

Nombre:	DatFiscal
Tipo de clase:	Entidad
Descripción	
Permite el acceso a la tabla de la base de datos DatFiscal, que contiene la información referente a las obligaciones y derechos fiscales.	

Nombre:	DatDerechoubligación
Tipo de clase:	Entidad
Descripción	
Permite el acceso a la tabla de la base de datos DatDerechoubligación, que contiene la información referente a las obligaciones y derechos.	

Nombre:	DatRegistroanexoop
Tipo de clase:	Entidad
Descripción	
Permite el acceso a la tabla de la base de datos DatRegistroanexoop, que contiene la información referente a los registros de anexos al pase.	

Nombre:	NomEstadodocumento
Tipo de clase:	Entidad
Descripción	
Permite el acceso a la tabla de la base de datos NomEstadodocumento, que es un nomenclador que contiene los estados que puede tener un documento.	

Nombre:	DatInstrumento
Tipo de clase:	Entidad
Descripción	
Permite el acceso a la tabla de la base de datos DatInstrumento, que contiene la información referente a los instrumentos.	

Nombre:	DatAnticipado
Tipo de clase:	Entidad
Descripción	
Permite el acceso a la tabla de la base de datos DatAnticipado, que contiene la información referente a las operaciones anticipadas.	

Nombre:	NomEstadoantipico
Tipo de clase:	Entidad
Descripción	
Permite el acceso a la tabla de la base de datos NomEstadoantipico, que es un nomenclador que contiene los estados en que se puede encontrar un anticipo.	

Nombre:	DatDocumentocp
Tipo de clase:	Entidad

Descripción
Permite el acceso a la tabla de la base de datos DatDocumentocp, que contiene la información referente a los documentos de cobros y pagos.

Nombre:	SubmayorclienteproveedorController
Tipo de clase:	Controladora
Atributo	Tipo
Para cada responsabilidad:	
Nombre:	Descripción
CargarsubmyAction()	Este método obtiene mediante Post los parámetros para mostrar el submayor y le pide el submayor a la clase del negocio SubmayorclienteModel para luego ser mostrado en la vista submayorclienteproveedor.phtml.

Descripción de las clases del diseño del componente Submayor.

Clases controladores del componente Submayor.

Nombre:	SubmayorfiscalController
Tipo de clase:	Controladora
Atributo	Tipo
Para cada responsabilidad:	
Nombre:	Descripción
CargarsubmyAction()	Este método obtiene mediante Post los parámetros para mostrar el submayor y le pide el submayor a la clase del negocio SubmayorclienteModel para luego ser mostrado en la vista submayorfiscal.phtml.phtml.

Clases del negocio del componente Submayor.

Nombre:	SubmayorclienteModel
Tipo de clase:	Modelo
Atributo	Tipo
Para cada responsabilidad:	
Nombre:	Descripción
MostrarSumayor(\$cuenta, \$cliente, \$fi, \$ff, \$idmoneda, \$idmonedacont)	Recibe los parámetros del submayor y devuelve una variable con los datos del mismo.
ComparaFechas(\$fecha1, \$fecha2)	Permite comprar dos fechas pasadas como parámetros.

Clases del dominio del componente Submayor.

Nombre:	BaseRes-Resumencuentacyp
Tipo de clase:	Entidad
Descripción	
Permite el acceso a la tabla de la base de datos BaseRes-Resumencuentacyp, que contiene información de las afectaciones realizadas a las cuentas de cobros y pagos.	

Descripción de las clases del diseño de Cambio de cuenta del componente Funcionalidades.

Clases controladoras de Cambio de cuenta del componente Funcionalidades.

Nombre:	CambiarcuentaoefectoalitigioController
Tipo de clase:	Controladora
Atributo	Tipo
Para cada responsabilidad:	
Nombre:	Descripción

CambiarCtaAction()	Permite realizar el cambio de cuenta obteniendo los parámetros por POST y utilizando servicios del ioc.xml. Contabiliza la operación mediante la clase del negocio ContabilizarCambioCuentaModel.
--------------------	---

Clases del negocio de Cambio de cuenta del componente Funcionalidades.

Nombre:	ContabilizarCambioCuentaModel
Tipo de clase:	Modelo
Atributo	Tipo
Para cada responsabilidad:	
Nombre:	Descripción
ContabilizarDO(\$iddocs)	Permite contabilizar el cambio de cuenta utilizando servicios del ioc.xml.

Clases del paquete de dominio de Cambio de cuenta del componente Funcionalidades.

Nombre:	DatCambiocuenta
Tipo de clase:	Entidad
Descripción	
Permite el acceso a la tabla de la base de datos DatCambiocuenta, que contiene la información referente a los cambios de cuenta.	

Nombre:	DatDocumentocp
Tipo de clase:	Entidad
Descripción	
Permite el acceso a la tabla de la base de datos DatDocumentocp, que contiene la información referente a los documentos de cobros y pagos.	

Nombre:	DatCambioctaantipico
Tipo de clase:	Entidad
Descripción	
Permite el acceso a la tabla de la base de datos DatCambioctaantipico, que contiene la información referente a los cambios de cuenta a los anticipos.	

Nombre:	DatDerechouobligacion
Tipo de clase:	Entidad
Descripción	
Permite el acceso a la tabla de la base de datos DatDerechouobligacion, que contiene la información referente a los derechos y las obligaciones.	

Nombre:	DatInstrumento
Tipo de clase:	Entidad
Descripción	
Permite el acceso a la tabla de la base de datos DatInstrumento, que contiene la información referente a los instrumentos.	

Descripción de las clases del diseño de Cambio de moneda del componente Funcionalidades.

Clases controladoras de Cambio de moneda del componente Funcionalidades.

Nombre:	GestionarcontratoController
Tipo de clase:	Controladora
Atributo	Tipo
Para cada responsabilidad:	
Nombre:	Descripción
AdicionarContratoAction()	Permite adicionar un nuevo contrato obteniendo los parámetros por POST y enviándolos a guardar mediante servicios del ioc.xml.
ModificarcontratoAction()	Permite modificar un contrato obteniendo los parámetros mediante POST y enviándolos a guardar mediante servicios del ioc.xml.
CancelacontratoAction()	Permite cancelar un contrato obteniendo los parámetros mediante POST y enviándolos a guardar mediante servicios del ioc.xml.
ConfirmarcontratoAction()	Permite confirmar uno o varios contratos obteniendo los identificadores de los mismos mediante POST y utilizando servicios del ioc.xml para realizar la confirmación.
ContabilizarcontratoAction()	Permite contabilizar uno o varios contratos obteniendo los identificadores de los mismos mediante POST y enviando los datos a la clase del negocio ContabilizarCambioMoneda.

Clases del negocio de Cambio de moneda del componente Funcionalidades.

Nombre:	ContabilizarCambioMoneda
Tipo de clase:	Modelo
Atributo	Tipo
Para cada responsabilidad:	
Nombre:	Descripción
Contabilizar(\$iddocs)	Permite contabilizar un conjunto de contratos recibiendo un arreglo con el id de los mismos. Utiliza servicios del ioc.xml.

Clases del paquete de dominio de Cambio de moneda del componente Funcionalidades.

Nombre:	DatCambiomoneda
Tipo de clase:	Entidad
Descripción	
Permite el acceso a la tabla de la base de datos DatCambiomoneda, que contiene la información referente a los cambios de moneda.	

Nombre:	DatContrato
Tipo de clase:	Entidad
Descripción	
Permite el acceso a la tabla de la base de datos DatContrato, que contiene la información referente a los contratos.	

Nombre:	DatDoccancelación
Tipo de clase:	Entidad
Descripción	
Permite el acceso a la tabla de la base de datos DatDoccancelación, que contiene la información referente las cancelaciones.	

Nombre:	DatOperación
Tipo de clase:	Entidad
Descripción	
Permite el acceso a la tabla de la base de datos DatOperación que contiene la información de las operaciones asociadas a los derechos y las obligaciones.	

Nombre:	NomEstadoantipico
Tipo de clase:	Entidad
Descripción	
Permite el acceso a la tabla de la base de datos NomEstadoantipico, que es un nomenclador que contiene los estados que puede tener un anticipo.	

Descripción de las clases del diseño de Liquidación del componente Funcionalidades.

Clases controladoras de Liquidación del componente Funcionalidades.

Nombre:	GestregistrainstrumentocobroController
Tipo de clase:	Controladora
Atributo	Tipo
Para cada responsabilidad:	
Nombre:	Descripción

CalculaImportembAction(\$importemo, \$factordeconversion, \$tasacambio)	Este método permite calcular el importe en moneda contable dados los parámetros importe en moneda original, factor de conversión y tasa de cambio.
GuardarAction()	Este método obtiene mediante POST los datos un instrumento de cobro y verifica si es para liquidar o para registrar un anticipo, enviando mediante servicios del ioc.xml los datos para ser guardado el instrumento según la operación que se vaya a realizar con el mismo.
LiquidarAction()	Permite obtener mediante POST los datos de un instrumento de cobro para liquidar y los datos del derecho de cobro que liquida, envía estos datos utilizando servicios para realizar la liquidación del instrumento contra el derecho. Envía los datos del instrumento y del derecho a la clase del negocio ContabilizarLiquidaciónCobroPagoDerechoObligaciónModel para contabilizar la operación.
ConfirmarAction()	Permite obtener mediante POST el identificador de un instrumento de cobro que se desea confirmar y utiliza un servicio del ioc.xml para confirmarlo.
ContabilizarAction()	Permite enviar a la clase del negocio ContabilizarCobroPagoAnticipadoModel los datos de un instrumento de cobro para ser contabilizado como un anticipo. Los datos del instrumento los obtiene mediante POST.

Nombre:	GestregistrainstrumentopagoController
Tipo de clase:	Controladora
Atributo	Tipo
Para cada responsabilidad:	
Nombre:	Descripción

CalculaImportembAction(\$importo, \$factordeconversion, \$tasacambio)	Este método permite calcular el importe en moneda contable dados los parámetros importe en moneda original, factor de conversión y tasa de cambio.
GuardarAction()	Este método obtiene mediante POST los datos un instrumento de pago y verifica si es para liquidar o para un anticipo, enviando mediante servicios del ioc.xml los datos para ser guardado el instrumento según la operación que se vaya a realizar en el mismo.
LiquidarAction()	Permite obtener mediante POST los datos de un instrumento de pago para liquidar y los datos de la obligación de pago que liquida, envía estos datos utilizando servicios para realizar la liquidación del instrumento contra la obligación. Envía los datos del instrumento y de la obligación a la clase del negocio ContabilizarLiquidacionCobroPagoDerechoObligaciónModel para contabilizar la operación.
ConfirmarAction()	Permite obtener mediante POST el identificador de un instrumento de pago que se desea confirmar y utiliza un servicio del ioc.xml para confirmarlo.
ContabilizarAction()	Permite enviar a la clase del negocio ContabilizarCobroPagoAnticipadoModel los datos de un instrumento de pago para ser contabilizado como un anticipo. Los datos del instrumento los obtiene mediante POST.

Nombre:	GestregistrarinstrumentopagofiscalController
Tipo de clase:	Controladora
Atributo	Tipo
Para cada responsabilidad:	

Nombre:	Descripción
CalculaImportembAction(\$importemo, \$factordeconversion, \$tasacambio)	Este método permite calcular el importe en moneda contable dados los parámetros importe en moneda original, factor de conversión y tasa de cambio.
LiquidarAction()	Permite obtener mediante POST los datos de un instrumento de pago fiscal y los datos de la obligación de pago que liquida, envía estos datos utilizando servicios para realizar la liquidación del instrumento contra la obligación fiscal. Envía los datos del instrumento y de la obligación a la clase del negocio ContabilizarLiquidacionCobroPagoDerechoObligaciónModel para contabilizar la operación.
ConfirmarAction()	Permite obtener mediante POST el identificador de un instrumento de pago fiscal que se desea confirmar y utiliza un servicio del ioc.xml para confirmarlo.

Nombre:	GestregistrarinstrumentocobrofiscalController
Tipo de clase:	Controladora
Atributo	Tipo
Para cada responsabilidad:	
Nombre:	Descripción
CalculaImportembAction(\$importemo, \$factordeconversion, \$tasacambio)	Este método permite calcular el importe en moneda contable dados los parámetros importe en moneda original, factor de conversión y tasa de cambio.
LiquidarAction()	Permite obtener mediante POST los datos de un instrumento de cobro fiscal y los datos del derecho de cobro fiscal que liquida, envía estos datos utilizando servicios para realizar la liquidación del instrumento contra el derecho fiscal. Envía los datos del

	instrumento y del derecho a la clase del negocio ContabilizarLiquidacionCobroPagoDerechoObligacion para contabilizar la operación.
ConfirmarAction()	Permite obtener mediante POST el identificador de un instrumento de cobro fiscal que se desea confirmar y utiliza un servicio del ioc.xml para confirmarlo.

Nombre:	GestgestionaroperacionesanticipadascobroController
Tipo de clase:	Controladora
Atributo	Tipo
Para cada responsabilidad:	
Nombre:	Descripción
CalculaImportembAction(\$importemo, \$factordeconversion, \$tasacambio)	Este método permite calcular el importe en moneda contable dados los parámetros importe en moneda original, factor de conversión y tasa de cambio.
LiquidarcobroanticipadoAction()	Permite obtener mediante POST el identificador del instrumento asociado al cobro anticipado y los datos de las operaciones asociadas al derecho de cobro con que se liquida, estos datos son enviados(mediante servicios del ioc.xml) para liquidar el instrumento de cobro contra el derecho de cobro. El método también envía los datos del instrumento y del derecho a la clase del negocio ContabilizarLiquidaciónCobroPagoDerechoObligaciónModel para contabilizar la operación.
ModificarIntAction()	Permite obtener mediante POST los datos del instrumento asociado a un cobro anticipado que se desea modificar, los nuevos datos son enviados para ser modificados mediante un

	servicio del ioc.xml.
CancelacobroanticipadoAction()	Permite obtener mediante POST los datos de la cancelación de uno o varios cobros anticipados, estos datos son enviados mediante un servicio para realizar la cancelación de los cobros anticipados.
ConfirmarAction()	Permite obtener mediante POST los identificadores de los cobros anticipados que se desean confirmar, verifica que estos se encuentren en estado <i>guardado</i> y utiliza un servicio del ioc.xml para realizar la confirmación.
ContabilizarAction()	Permite enviar a la clase del negocio ContabilizarCobroPagoAnticipadoModel los datos de uno varios instrumentos de cobro para ser contabilizados como anticipos. Los datos de los instrumentos los obtiene mediante POST.

Nombre:	GestgestionaroperacionesanticipadascobroController
Tipo de clase:	Controladora
Atributo	Tipo
Para cada responsabilidad:	
Nombre:	Descripción
CalculaImportembAction(\$importe, \$factordeconversion, \$tasacambio)	Este método permite calcular el importe en moneda contable dados los parámetros importe en moneda original, factor de conversión y tasa de cambio.
LiquidarpagoanticipadoAction()	Permite obtener mediante POST el identificador del instrumento asociado al pago anticipado y los datos de las operaciones asociadas a la obligación con que se liquida, estos datos son enviados (mediante servicios del ioc.xml) para liquidar el instrumento de pago contra la obligación. El método también envía los datos del instrumento y de la obligación a la clase del

	<p>negocio</p> <p>ContabilizarLiquidaciónCobroPagoDerechoObligaciónModel para contabilizar la operación.</p>
ModificarIntAction()	<p>Permite obtener mediante POST los datos del instrumento asociado a un pago anticipado que se desea modificar, los nuevos datos son enviados para ser modificados mediante un servicio del ioc.xml.</p>
CancelapagoanticipadoAction()	<p>Permite obtener mediante POST los datos de la cancelación de uno o varios pagos anticipados, estos datos son enviados mediante un servicio para realizar la cancelación de los pagos anticipados.</p>
ConfirmarAction()	<p>Permite obtener mediante POST los identificadores de los pagos anticipados que se desean confirmar, verifica que estos se encuentren en estado <i>guardado</i> y utiliza un servicio del ioc.xml para realizar la confirmación.</p>
ContabilizarAction()	<p>Permite enviar a la clase del negocio ContabilizarCobroPagoAnticipadoModel los datos de uno varios instrumentos de pago para ser contabilizados como anticipos. Los datos de los instrumentos los obtiene mediante POST.</p>

Clases del negocio de Liquidación del componente Funcionalidades.

Nombre:	ContabilizarCobroPagoAnticipadoModel
Tipo de clase:	Modelo
Atributo	Tipo
Para cada responsabilidad:	
Nombre:	Descripción
Contabilizar(\$instrumentos = array()) ()	Permite contabilizar uno o varios instrumentos como operaciones anticipadas, recibiendo como parámetro un arreglo con los datos de los instrumentos.

Nombre:	ContabilizarLiquidaciónCobroPagoDerechoObligaciónModel
Tipo de clase:	Modelo
Atributo	Tipo
Para cada responsabilidad:	
Nombre:	Descripción
Contabilizar(\$instrumentos = array()) ()	Permite contabilizar uno o varios instrumentos contra obligaciones o derechos, recibiendo como parámetro un arreglo con los datos de los instrumentos y las obligaciones o derechos.

Clases del paquete de dominio de Liquidación del componente Funcionalidades.

Nombre:	DatLiquidación
Tipo de clase:	Entidad
Descripción	
Permite el acceso a la tabla de la base de datos DatLiquidación, que contiene la información referente a las liquidaciones.	

Nombre:	DatInstrumento
Tipo de clase:	Entidad
Descripción	
Permite el acceso a la tabla de la base de datos DatInstrumento, que contiene la información referente a los instrumentos.	

Nombre:	DatAnticipado
Tipo de clase:	Entidad
Descripción	
Permite el acceso a la tabla de la base de datos DatAnticipado, que contiene la información referente a las operaciones anticipadas.	

Nombre:	NomTipooperación
Tipo de clase:	Entidad
Descripción	
Permite el acceso a la tabla de la base de datos NomTipooperación, que es un nomenclador que contiene las operaciones que se puede realizar con un instrumento.	

Nombre:	NomEstadoanticipo
Tipo de clase:	Entidad
Descripción	
Permite el acceso a la tabla de la base de datos NomEstadoanticipo, que es un nomenclador que contiene los estados en que se puede encontrar un anticipo.	

Nombre:	DatCancelaciónantipico
Tipo de clase:	Entidad
Descripción	
Permite el acceso a la tabla de la base de datos DatCancelaciónantipico, que contiene la información referente a la cancelación de anticipo.	

Nombre:	DatDoccancelación
Tipo de clase:	Entidad
Descripción	
Permite el acceso a la tabla de la base de datos DatDoccancelación, que contiene la información referente a las cancelaciones.	

Nombre:	DatCancelacióndcop
Tipo de clase:	Entidad
Descripción	
Permite el acceso a la tabla de la base de datos DatCancelacióndcop, que contiene la información referente a las cancelaciones de derechos y obligaciones.	

Nombre:	DatOperación
Tipo de clase:	Entidad
Descripción	
Permite el acceso a la tabla de la base de datos DatOperación, que contiene la información referente a las operaciones asociadas a los derechos y obligaciones.	

Nombre:	DatDiferencialiquidación
Tipo de clase:	Entidad
Descripción	
Permite el acceso a la tabla de la base de datos DatDiferencialiquidación, que contiene la información referente a la diferencia que se genera al realizar una liquidación.	

Nombre:	DatLetra
Tipo de clase:	Entidad
Descripción	
Permite el acceso a la tabla de la base de datos DatLetra, que contiene la información referente a las letras de cambio.	

Nombre:	DatDocumentocp
Tipo de clase:	Entidad
Descripción	
Permite el acceso a la tabla de la base de datos DatDocumentocp, que contiene la información referente a los documentos de cobros y pagos.	

Descripción de las clases del diseño del componente Derecho Obligación.

Clases controladoras.

Nombre:	GestionarderechofiscalController
Tipo de clase:	Controladora
Atributo	Tipo
Para cada responsabilidad:	
Nombre:	Descripción
AdiderechofiscalAction()	Permite adicionar un nuevo derecho fiscal obteniendo los parámetro mediante POST y creando un nuevo objeto con estos parámetros y enviándolos a la clase del negocio derechosfiscalesModel para ser guardados.
Modderechofiscal()	Permite modificar un derecho fiscal obteniendo por POST los valores de los campos del derecho fiscal, para ser enviados a la clase del negocio derechosfiscalesModel que se encarga de guardar los cambios.
CanceladerechocobrofiscalAction()	Permite cancelar uno o varios derechos fiscales obteniendo los datos de la cancelación mediante POST. Utiliza servicios del ioc.xml para realizar la cancelación.
Confirmarderechofiscal()	Permite confirmar uno o mas derechos fiscales obteniéndolos por POST y cambiando el estado de estos a “confirmado”.
ContabilizarderechofiscalAction()	Permite contabilizar uno o mas derechos fiscales obteniéndolos por POST y enviándolos a la clase del negocio ContabilizarDerechoObligaciónModel mediante el método Contabilizar(\$docs = array()).

Nombre:	GestionarderechocobroController
Tipo de clase:	Controladora
Atributo	Tipo
Para cada responsabilidad:	
Nombre:	Descripción
AdiderechocobroAction()	Permite adicionar un nuevo derecho de cobro obteniendo los parámetro mediante POST y creando un nuevo objeto con estos parámetros y enviándolos a la clase del negocio derechosdecobroModel para ser guardados.
ModderechocobroAction()	Permite modificar un derecho de cobro obteniendo por POST los valores de los campos del derecho para ser enviados a la clase del negocio derechosdecobroModel que se encarga de guardar los cambios.
CanceladerechocobroclienteAction()	Permite cancelar uno o varios derechos de cobro obteniendo los datos de la cancelación mediante POST. Utiliza servicios del ioc.xml para realizar la cancelación.
ConfirmarAction()	Permite confirmar uno o más derechos de cobro obteniéndolos por POST y cambiando el estado de ellos mediante servicios del ioc.xml.
ContabilizarderechofiscalAction()	Permite contabilizar uno o mas derechos de cobro obteniéndolos por POST y enviándolos a la clase del negocio ContabilizarDerechoObligaciónModel mediante el método Contabilizar(\$docs = array()).

Nombre:	GestionarobligaciónpagoController
Tipo de clase:	Controladora
Atributo	Tipo
Para cada responsabilidad:	
Nombre:	Descripción
AdiobligacióndepagoAction()	Permite adicionar una nueva obligación de pago obteniendo los parámetros mediante POST y creando un nuevo objeto con estos parámetros y enviándolos a la clase del negocio ObligaciónpagoModel para ser guardados.
ModobligacióndepagoAction()	Permite modificar una obligación de pago obteniendo por POST los valores de los campos de la obligación, para ser enviados a la clase del negocio ObligaciónpagoModel que se encarga de guardar los cambios.
CancelaobligaciónpagoclienteAction()	Permite cancelar una o varias obligaciones de pago obteniendo los datos de la cancelación mediante POST. Utiliza servicios del ioc.xml para realizar la cancelación.
ConfirmarAction()	Permite confirmar una o más obligaciones de pago obteniéndolas por POST y cambiando el estado de ellas mediante servicios.
ContabilizarAction()	Permite contabilizar una o mas obligaciones de pagos obteniéndolas por POST y enviándolas a la clase del negocio ContabilizarDerechoObligaciónModel mediante el método Contabilizar(\$docs = array()).

Nombre	GestionarobligacionesfiscalesController
Tipo de clase:	Controladora
Atributo	Tipo
Para cada responsabilidad:	
Nombre:	Descripción
AdicionarobligaciónfiscalAction()	Permite adicionar una nueva obligación fiscal obteniendo los parámetros mediante POST y creando un nuevo objeto con estos parámetros y enviándolos a la clase del negocio obligacionesfiscalesModel para ser guardados.
ModificarobligaciónfiscalAction()	Permite modificar una obligación fiscal obteniendo por POST los valores de los campos de la obligación, para ser enviados a la clase del negocio ObligacionesfiscalesModel que se encarga de guardar los cambios.
CancelarobligaciónfiscalAction()	Permite cancelar una o varias obligaciones fiscales obteniendo los datos de la cancelación mediante POST. Utiliza servicios del ioc.xml para realizar la cancelación.
ConfirmarobligaciónfiscalAction()	Permite confirmar una o más obligaciones fiscales obteniéndolas por POST y cambiando el estado de ellas mediante servicios.
ContabilizarobligaciónfiscalAction()	Permite contabilizar una o mas obligaciones fiscales obteniéndolas por POST y enviándolas a la clase del negocio ContabilizarDerechoObligaciónModel mediante el método Contabilizar (\$docs = array()).

Clases del negocio del componente Derecho Obligación.

Nombre:	ContabilizarDerechoObligaciónModel
Tipo de clase:	Modelo
Atributo	Tipo
Para cada responsabilidad:	
Nombre:	Descripción
Contabilizar(\$docs= array())	Este método permite contabilizar uno o más documentos pasados como parámetros, el método utiliza el servicio Contabilizar para cada uno de los documentos.

Nombre:	DatOperacióncpModel
Tipo de clase:	Modelo
Atributo	Tipo
Para cada responsabilidad:	
Nombre:	Descripción
EliminaOperación(\$operaciones)	Este método permite eliminar de la base de datos la operación que tiene como identificador el parámetro del método.
InsertarOperación(\$array)	Este método permite guardar en la base de datos una operación pasando como parámetro un arreglo con los datos de la misma.
ModificarOperación(\$array)	Este método permite guardar en la base de datos los cambios realizado a una operación existente, los nuevos datos de la operación el método los recibe como parámetro en un arreglo.

Nombre:	DerechosfiscalesModel
Tipo de clase:	Modelo
Atributo	Tipo
Para cada responsabilidad:	
Nombre:	Descripción
Insertarderechofiscal(\$instance1)	Este método permite guardar en la base datos un derecho fiscal que es pasado como parámetro.
Modificarderechofiscal(\$instance1)	Este método permite modificar guardar en la base de datos los cambios realizados a un derecho fiscal, el que es pasado como parámetro.

Nombre:	DerechosdecobroModel
Tipo de clase:	Modelo
Atributo	Tipo
Para cada responsabilidad:	
Nombre:	Descripción
Insertarderechodecobro(\$deruobli, \$cliente, \$operaciones, \$anexo, \$idcliente)	Permite guardar en la base de datos un nuevo derecho de cobro recibiendo los datos como parámetros.
Modificarderechodecobro(\$deruobli, \$cliente, \$operaciones, \$anexo)	Permite guardar en la base de datos los cambios realizados a un derecho de cobro recibiendo los datos como parámetros.

Nombre:	ObligaciónpagoModel
Tipo de clase:	Modelo
Atributo	Tipo
Para cada responsabilidad:	
Nombre:	Descripción
Insertarob(\$derecho, \$cliente, \$operaciones, \$anexo, parameter)	Permite guardar en la base de datos una nueva obligación de pago recibiendo los datos como parámetros.
Modificarob(\$derecho, \$cliente, \$operaciones, \$anexosmod)	Permite guardar en la base de datos los cambios realizados a una obligación de pago recibiendo los datos como parámetros.

Nombre:	ObligacionesfiscalesModel
Tipo de clase:	Modelo
Atributo	Tipo
Para cada responsabilidad:	
Nombre:	Descripción
Adicionarobligfiscal(\$instanciaoblig)	Permite guardar en la base de datos una nueva obligación fiscal recibiendo la instancia de la misma como parámetro.
Modificarobligfiscal(\$obligacion)	Permite guardar en la base de datos los cambios realizados a una obligación fiscal recibiendo la instancia de la misma como parámetro.

Clases del paquete de dominio del componente Derecgo Obligación.

Nombre:	DatCondiciónpago
Tipo de clase:	Entidad
Descripción	
Permite el acceso a la tabla de la base de datos DatCondiciónpago, que contiene la información correspondiente a la condición de pago de los derechos y las obligaciones.	

Nombre:	NomPárrafofiscal
Tipo de clase:	entidad
Descripción	
Permite el acceso a la tabla de la base de datos NomPárrafofiscal, que es un nomenclador que contiene los párrafos fiscales definidos.	

Nombre:	NomEstadodcop
Tipo de clase:	Entidad
Descripción	
Permite el acceso a la tabla de la base de datos NomEstadodcop, que es un nomenclador que contiene los estados que puede tener un documento de cobros y pagos.	

Nombre:	NomTipo
Tipo de clase:	Entidad
Descripción	
Permite el acceso a la tabla de la base de datos NomTipo, que es un nomenclador que contiene los tipos de derechos y obligaciones que existen.	

Nombre:	DatOperación
Tipo de clase:	Entidad
Descripción	
Permite el acceso a la tabla de la base de datos DatOperación, que contiene la información referente a las operaciones asociadas a los derechos y obligaciones.	

Nombre:	DatCliente
Tipo de clase:	Entidad
Descripción	
Permite el acceso a la tabla de la base de datos DatCliente, que contiene la información referente a los clientes que realizan operaciones de cobros y pagos.	

Nombre:	DatFiscal
Tipo de clase:	Entidad
Descripción	
Permite el acceso a la tabla de la base de datos DatFiscal, que contiene la información referente a las obligaciones y derechos fiscales.	

Nombre:	DatDerechoubligación
Tipo de clase:	Entidad
Descripción	
Permite el acceso a la tabla de la base de datos DatDerechoubligación, que contiene la información referente a las obligaciones y derechos.	

Nombre:	DatRegistroanexoop
Tipo de clase:	Entidad
Descripción	
Permite el acceso a la tabla de la base de datos DatRegistroanexoop, que contiene la información referente a los registros de anexos al pase.	

Nombre:	NomEstadodocumento
Tipo de clase:	Entidad
Descripción	

Permite el acceso a la tabla de la base de datos NomEstadodocumento, que es un nomenclador que contiene los estados que puede tener un documento.

3.6. Patrones.

Un patrón es una unidad de información nombrada, instructiva e intuitiva que captura la esencia de una familia exitosa de soluciones probadas a un problema recurrente dentro de cierto contexto. Su objetivo es crear un lenguaje común para la comunidad de desarrolladores.

Ventajas del uso de los patrones:

- Ayudan a construir la experiencia colectiva de Ingeniería de Software.
- Son una abstracción de "problema – solución".
- Se ocupan de problemas recurrentes.
- Identifican y especifican abstracciones de niveles más altos que componentes o clases individuales.
- Proporcionan vocabulario y entendimiento común.

Patrones de diseño:

Un patrón de diseño es una abstracción de una solución en un nivel alto. Los patrones solucionan problemas que existen en muchos niveles de abstracción. Hay patrones que abarcan las distintas etapas del desarrollo; desde el análisis hasta el diseño y desde la arquitectura hasta la implementación.

Clasificación de los patrones de diseño: (Larman, 1999)

- Creacionales: Patrones creacionales tratan con las formas de crear instancias de objetos. El objetivo de estos patrones es de abstraer el proceso de instanciación y ocultar los detalles de cómo los objetos son creados o inicializados.
- Estructurales: Los patrones estructurales describen como las clases y objetos pueden ser combinados para formar grandes estructuras y proporcionar nuevas funcionalidades. Estos objetos adicionales pueden ser incluso objetos simples u objetos compuestos.

- Comportamiento: Los patrones de comportamiento nos ayudan a definir la comunicación e iteración entre los objetos de un sistema. El propósito de este patrón es reducir el acoplamiento entre los objetos.

Los frameworks EXT, Zend Frameworks (ZF) y Doctrine propuestos por la línea de Arquitectura del Cedrux para la implementación de cada uno de sus módulos incorporan algunos patrones de diseño, por cuanto se propone no volver a especificar los mismos durante la realización del diseño del módulo Cobros y Pagos.

Patrones de diseño empleados:

- Patrón Decorator:

El patrón Decorator también conocido como Wrapper, es un patrón estructural.

Funcionamiento:

En ocasiones se desea adicionar responsabilidades a un objeto, pero no a toda la clase, estas se pueden adicionar por medio de los mecanismos de herencia, pero este mecanismo no es flexible porque la responsabilidad es adicionada estáticamente. La solución flexible es la de rodear el objeto con otro objeto que es el que adiciona la nueva responsabilidad. Este nuevo objeto es el Decorator.

Aplicaciones:

El Decorator se debe usar para:

- Adicionar responsabilidades a objetos individuales dinámicamente sin afectar otros objetos.
- Para agregar responsabilidades que pueden ser retiradas.
- Cuando no es práctico adicionar responsabilidades por medio de la herencia.

El framework ZF implementa el patrón Decorator en su clase Zend View, la cual es la encargada de asignarle responsabilidades a objetos de manera dinámica y configurarlos con nuevos atributos.

- Patrón MVC (Modelo Vista Controlador).

Es un patrón de diseño que plantea la separación de diferentes clases en dependencia de la función que realizan, con el objetivo de manejar dinámicamente la forma en que se procesan solicitudes y se muestran resultados al usuario final. En otras palabras separa la presentación del dominio de la aplicación.

El objetivo principal del MVC es realizar un diseño que desacople la vista del modelo, con la finalidad de mejorar la reusabilidad, permitiendo esto que las modificaciones realizadas en las vistas influyan en menor medida en la lógica de negocio o de datos.

Los frameworks utilizados para la implementación del módulo implementan el patrón MVC, lo cual posibilita que se puedan utilizar de forma sencilla mediante la implementación de interfaces.

Elementos del patrón MVC:

- Modelo: Cuenta con los datos y las reglas del negocio.
- Vista: Muestra la información del modelo al usuario.
- Controlador: Gestiona las entradas del usuario.

Responsabilidades de los elementos:

El modelo es el responsable de:

- Acceder a la capa de almacenamiento de datos.
- Define las reglas de negocio (la funcionalidad del sistema).
- Lleva un registro de las vistas y controladores del sistema.

La vista es responsable de:

- Recibir datos del modelo y mostrarlos al usuario.
- Tiene un registro de su controlador asociado.

El controlador es el responsable de:

- Recibir los eventos de entrada (un clic, un cambio en un campo de texto, etc.).
- Contiene reglas de gestión de eventos.

Para la realización del diseño del módulo Cobros y Pagos los diseñadores emplearon los patrones GRASP⁶ ya que asignar correctamente las responsabilidades de las clases es muy importante en el diseño orientado a objetos.

Patrones GRASP:

- Experto: La función de este patrón es asignar una responsabilidad al experto en información, o sea la clase que cuenta con la información necesaria para cumplir la responsabilidad.
- Controlador: Asignar la responsabilidad del manejo de un mensaje de los eventos de un sistema a una clase que presente una de las siguientes opciones:
 - El “sistema” global (controlador de fachada).
 - La empresa u organización global (Controlador de fachada).
 - Algo en el mundo real que es activo (por ejemplo, el papel de una persona) y que pueda participar en la tarea (controlador de tareas).
 - Un manejador artificial de todos los eventos del sistema de un caso de uso, generalmente denominado “Manejador<NombreCasosdeUso” (controlador de casos de uso).
- Creador: El patrón Creador guía la asignación de responsabilidades relacionadas con la creación de objetos, tarea muy frecuente en los sistemas orientados a objetos.
- Bajo acoplamiento: Asignar una responsabilidad para mantener bajo acoplamiento. El acoplamiento es una medida de la fuerza con que una clase está conectada a otras clases. Una clase con bajo (o débil) acoplamiento no depende de muchas otras
- Alta cohesión: El objetivo de este patrón es asignar una responsabilidad de modo que la cohesión siga siendo alta. En la perspectiva del diseño orientado a objetos, la cohesión es una medida de cuan relacionadas y enfocadas están las responsabilidades de una clase. Una alta cohesión caracteriza a las clases con responsabilidades estrechamente relacionadas que no realicen un trabajo enorme.

⁶ Patrones de Software para la Asignación General de Responsabilidades del inglés General Responsibility Assignment Software Patterns.

3.7. Métricas para evaluar el diseño propuesto.

Las métricas son un buen medio para entender, monitorizar, controlar, predecir y probar el desarrollo de softwares y de proyectos de mantenimiento. (Calero Muñoz, 2006)

Objetivos de las métricas de diseño:

- Llevar un control de la calidad del producto que se está desarrollando.
- Estimar el impacto que ese producto tendrá en las fases posteriores del proceso de desarrollo.
- Ayudan a que el diseño evolucione a un nivel superior de calidad.

Existen varias métricas para medir la calidad del diseño realizado, estas se estructuran en:

- Métricas de diseño de alto nivel.
- Métricas de diseño de componentes.
- Métricas de diseño de interfaz.
- Métricas de diseño Orientado a Objetos (OO).

Para evaluar el diseño propuesto para el módulo Cobros y Pagos se utilizaron las métricas OO, específicamente las orientadas a clases, debido a que en un sistema OO la clase es la unidad fundamental, por cuanto las medidas y métricas para una clase individual, la jerarquía de clases y las colaboraciones de estas permiten al ingeniero del software evaluar la calidad del diseño propuesto. (Pressman)

Métricas empleadas para evaluar el diseño del módulo Cobros y Pagos:

Métricas propuestas por Lorenz y Kidd: son métricas basadas en clases divididas en las siguientes categorías: (Pressman, 2005)

- Tamaño.
- Herencia.
- Valores internos.
- Valores externos.

La métrica del tamaño de clase (TC) fue la que se empleó en el módulo Cobros y Pagos, debido a que es una métrica fácil de usar y se ajusta más para evaluar el diseño realizado. Esta métrica no se le aplicó a todas las clases del diseño, debido a la complejidad del módulo y al gran cúmulo de clases identificadas, por cuanto se propuso solo aplicarla a las clases modelos de los diagramas realizados.

El tamaño general de una clase se mide a través de las siguientes medidas:

- El total de operaciones (operaciones tanto heredadas como privadas de la instancia), que se encapsulan dentro de la clase.
- El número de atributos (atributos tanto heredados como privados de la instancia), encapsulados por la clase.
- Promedio general de las dos métricas anteriores para el sistema en general.

Para evaluar estas métricas se utilizaron los umbrales siguientes:

- Umbrales para el tamaño general de las clases.

Clasificación	Valores de los umbrales
Pequeño	\leq Promedio de operaciones(PO)
Medio	$>$ PO y $\leq 2 * PO$
Grande	$> 2 * PO$

- Umbrales para la responsabilidad de las clases.

Categoría	Valores de los umbrales
Baja	\leq Promedio de operaciones(PO)
Media	$>$ PO y $\leq 2 * PO$
Alta	$> 2 * PO$

- Umbrales para la complejidad de implementación de las clases.

Categoría	Valores de los umbrales
Baja	\leq Promedio de operaciones(PO)

Media	>PO y <=2*PO
Alta	>2*PO

➤ Umbrales para la reutilización de las clases.

Categoría	Valores de los umbrales
Baja	< =Promedio de operaciones(PO)
Media	>PO y <=2*PO
Alta	>2*PO

No.	Nombre	Cantidad de atributos	Cantidad de operaciones	Tamaño
1	DatRefanticipoModel	0	1	Pequeño
2	DatConciliaciónModel	0	1	Pequeño
3	DatRefderechouobligaciónModel	0	1	Pequeño
4	DatOperacióncpModel	0	3	Medio
5	DerechosfiscalesModel	0	2	Pequeño
6	ContabilizarDerechoObligaciónModel	0	1	Pequeño
7	DerechosdecobroModel	0	2	Pequeño
8	ObligaciónpagoModel	0	2	Pequeño

9	ObligacionesfiscalesModel	0	2	Pequeño
10	ContabilizarCambioCuentaModel	0	1	Pequeño
11	ContabilizarCambioMonedaModel	0	1	Pequeño
12	SubmayorclienteModel	0	2	Pequeño
13	CargainicialModel	0	10	Grande
14	CierreContableModel	0	2	Pequeño
15	ContabilizarLiquidacionCobroPagoDerechoObligaciónModel	0	1	Pequeño
16	ContabilizarCobroPagoAnticipadoModel	0	1	Pequeño
17	PárrafoModel	0	3	Medio
18	CondiciónpagoModel	0	2	Pequeño
19	MotivoCancelaciónModel	0	2	Pequeño

Cantidad de clases por clasificación.

Clasificación	Cantidad Clases	Responsabilidad de las clases	Complejidad de implementación de las clases	Reutilización de las clases
Pequeño	16	Baja	Baja	Alta
Medio	2	Media	Media	Media
Grande	1	Alta	Alta	Baja

Resultado general de la métrica.

Cantidad de clases	Cantidad de clases pequeñas	Cantidad de clases grandes	Cantidad de clases medias	Promedio de atributos	Promedio de operaciones
19	16	1	2	0	2.0

La mayoría de las clases modelos de los diagramas de clases del diseño del módulo están dentro de la categoría de pequeñas lo cual demuestra que el diseño realizado es simple, por cuanto según esta técnica los diseñadores del módulo han obtenido resultados satisfactorios, ya que los valores pequeños de TC alcanzados demuestran que no se afectan los valores de calidad definidos por la métrica empleada, la implementación de forma general es sencilla, se pueden realizar pruebas con facilidad y disminuye en gran medida la responsabilidad de las clases.

3.8. Conclusiones parciales.

En este capítulo se realizó el diseño del módulo Cobros y Pagos, exponiéndose los diagramas de clases del diseño de cada uno de los componentes identificados, así como la descripción de las clases de los mismos. Establecer mecanismos de diseños es sumamente importante para mantener la homogeneidad en el diseño, los diseñadores del módulo Cobros y Pagos realizaron sus propios mecanismos, los cuales facilitaron en gran medida la realización de los diagramas de clases del módulo. Se realizó un análisis de los patrones de diseño que tienen incorporados los frameworks empleados. Además se evaluó el diseño propuesto a través de métricas.

CONCLUSIONES

Al concluir el presente trabajo de diploma y haber logrado el cumplimiento del objetivo general propuesto al inicio del mismo, se arribaron a las siguientes conclusiones:

- La obtención de la información necesaria referente a los procesos contables de cobros y pagos favoreció en gran medida la realización del análisis y diseño del módulo Cobros y Pagos.
- La utilización de técnicas para la captura de requisitos le permitió a los analistas del módulo Cobros y Pagos identificar los requisitos funcionales de forma eficiente y segura.
- Se evaluó la calidad de los principales artefactos generados: los requisitos y el diseño a través de métricas como por ejemplo las de Lorenz y Kidd para el caso del diseño y las técnicas de prototipos y Revisión Técnica Formal para los requisitos.
- Se garantiza la implementación del módulo Cobros y Pagos, a través de los artefactos obtenidos durante este trabajo de diploma.

RECOMENDACIONES

Se recomienda:

- Realizar la implementación del Módulo Cobros y Pagos a partir de los artefactos generados en el presente trabajo de diploma.
- Utilizar el presente trabajo de diploma como documentación para el desarrollo de la fase # 2 del módulo Cobros y Pagos.

BIBLIOGRAFÍA

Analistas del módulo Cobros y Pagos. 2009. Descripción Negocio-Realizar compra. 2009.

—. **2009.** ERP-CF-COPA Especificación de Requisitos (Gestionar derechos de cobro y fiscales)v1.4. 2009.

—. **2009.** ERP-CF-COPA Especificación de Requisitos (Configuración)v1.1. 2008. 2009.

—. **2009.** ERP-CF-COPA Especificación de Requisitos (Gestionar obligaciones de pago y fiscales)v1.4. 2009.

—. **2009.** ERP-CF-COPA Especificación de Requisitos Comunes del módulo (Cierre) v1.1. 2009.

—. **2009.** ERP-CF-COPA Especificación de requisitos de soporte v1.3. 2009.

—. **2008.** *ERP-CF-COPA Modelo Conceptual.* 2008.

—. **2009.** ERP-CF-COPA-Descripción Negocio-Conciliar con cliente. 2009. 2009.

—. **2009.** ERP-CF-COPA-Descripción Negocio-Conciliar con proveedor. 2009.

—. **2009.** ERP-CF-COPA-Descripción Negocio-Liquidar cobro anticipado. 2009.

—. **2009.** ERP-CF-COPA-Descripción Negocio-Liquidar compra. 2009.

—. **2009.** ERP-CF-COPA-Descripción Negocio-Pagar anticipado. 2009.

—. **2009.** ERP-CF-COPA-Descripción Negocio-Pagar con cheque. 2009.

—. **2009.** ERP-CF-COPA-Descripción Negocio-Pagar con letra de cambio. 2009.

—. **2009.** ERP-CF-COPA-Descripción Negocio-Realizar cobro. 2009.

—. **2009.** ERP-CF-COPA-Descripción Negocio-Realizar venta. 2009.

—. **2009.** ERP-CF-COPA-Descripción Negocio-Recibir productos. 2009.

Analistas dle módulo Cobros y Pagos. 2009. ERP-CF-COPA-Descripción Negocio-Liquidar pago anticipado. 2009.

Anlistas del módulo Cobros y Pagos. 2009. ERP-CF-COPA-Descripción Negocio-Realizar pago automático. 2009.

Article Garden. [En línea] [Citado el: 20 de Febrero de 2009.] <http://www.articlegarden.com/es/Article/Business-Analyst-Jobs/41123..>

Ayaviri I García, Daniel. *Contabilidad Básica y Documentos Mercantiles.* .

Boter Maupi, Fernando. 2008. *Curso de Contabilidad. Barcelona : s.n., 1923. (Boter, 1923).* s.l. : Gaceta, 2008.

Brito, Henry Raúl. 2006. ERP cubano, un paso estratégico para la consolidación del Software Libre en Cuba. [En línea] 2006 de Octubre de 2006. [Citado el: 10 de Diciembre de 2008.] http://www.informaticahabana.com/evento_vi.

Cabrera González, Lic.Miguel P. Forum de ciencia y técnica. [En línea] [Citado el: 23 de Febrero de 2009.] <http://www.forum.villaclara.cu/ponencias/trabajo/71..>

Canaviri, Erick, y otros. [En línea] [Citado el: 12 de Abril de 2008 .] <http://virtual.usalesiana.edu.bo/web/practica/archiv/clases22.doc..>

CITMATEL. Rodas XXI. [En línea] [Citado el: 23 de Febrero de 2009.] [http://www.rodasxxi.cu/..](http://www.rodasxxi.cu/)

2009. Curso práctico de Modelado de Negocios con UML y BPMN. [En línea] 2009. [Citado el: 27 de Febrero de 2009.] <http://www.milestone.com.mx/CursoModeladoNegociosBPMN.htm..>

Chaves Arias, Michael. 2006. La ingeniería de requerimientos y su importancia en el desarrollo de proyectos de software. [En línea] 7 de Julio de 2006. [Citado el: 4 de Marzo de 2009.] http://intersedes.ucr.ac.cr/pdfs_10/10-art_11.pdf..

El modelo de procesos de negocio. [En línea] [Citado el: 17 de Febrero de 2009.] http://www.sparxsystems.com.ar/downloads/whitepapers/El_Modelo_de_Proceso_de_Negocio.pdf..

El modelo de procesos de negocio. [En línea] [Citado el: 17 de Febrero de 2009.]
http://www.sparxsystems.com.ar/downloads/whitepapers/El_Modelo_de_Proceso_de_Negocio.pdf..

2009. *ERP-PROD Modelo de desarrollo orientado a componentes.* . 2009.

Expendedora:Modelo de Dominio. [En línea] [Citado el: 17 de Febrero de 2009.]
http://iie.fing.edu.uy/ense/assign/desasoftware/practico/hoja8/ejemplos_clase2..

Gaceta Oficial. 2008. 2008. No.035.

2008. Gestión Empresarial. [En línea] 1 de Octubre de 2008. [Citado el: 12 de Diciembre de 2008.]
<http://www.abartiateam.com/gestion-empresarial..>

2006. Interfaz de usuario. [En línea] 2006. [Citado el: 5 de Marzo de 2009.]
<http://www.fismat.umich.mx/~crivera/tesis/node6.html..>

2009. Las Bitacoras GWAI. [En línea] 2009. [Citado el: 7 de Abril de 2009.]
http://jcgcm.ei.uvigo.es/jcmoreno/?page_id=2..

Martin, E.I. 2005. Un efectivo sistema cubano de Contabilidad. [En línea] 2005. [Citado el: 12 de Diciembre de 2008.]
<http://www.eleconomista.cubaweb.cu/2005/edicionimpresa/premioeconomiamiguel.html..>

Ministerio de Finanzas y Precios. *La Resolución 235/05 y 294/05.*

2009. Modelado del negocio. [En línea] 2009. [Citado el: 24 de Febrero de 2009.]
http://merinde.rinde.gob.ve/index.php?option=com_content&task=view&id=19&Itemid=103..

Ochoa, E.S. 2007. Ventajas y Desventajas de ERP. [En línea] 2007. [Citado el: 13 de Diciembre de 2008.]
<http://secretosenred.com/articles/2753/2751/VENTAJAS-Y-DESVENTAJAS-DE-ERP/Paacutegina2751.html..>

Pressman, R.S. 2005. *Ingengería del Software:Un enfoque práctico.* 2005.

Sánchez Plancarte, Ing. Federico. [En línea] [Citado el: 22 de Febrero de 2009.]
<http://www.gestiopolis.com/recursos4/docs/ger/planerp.htm..>

Sosa Porteiro, Marisel. 2007. El VERSAT-Sarasola: Sistema cubano de Gestión Contable-Financiero. [En línea] Enero de 2007. [Citado el: 12 de Diciembre de 2008.] http://www.betsime.disaic.cu/secciones/eco_enemar_07.htm#2..

Canaviri, Erik, y otros. [En línea] [Citado el: 13 de Abril de 2009.] <http://virtual.usalesiana.edu.bo/web/practica/archiv/clases22>.

2009. Actualidad. [En línea] 9 de Marzo de 2009. [Citado el: 2 de Abril 0 de 2009.] <http://actualidad.espaciolinux.com/2009/03/09/2745/>. No.08.

Pressman, Roger S. Ingeniería del Software.Un enfoque práctico. Quinta Edición..

Negro, Pablo y Giandini, Roxana. UMBRALES PARA METRICAS ORIENTADAS A OBJETOS. [En línea] [Citado el: 01 de Mayo de 2009.] <http://www.lifia.info.unlp.edu.ar/papers/2007/Pablo2007.pdf>.

Calero Muñoz, Dr.Coral. 2006. Conceptos básicos, definición y formalización.Calidad de Software. [En línea] Octubre de 2006. [Citado el: 05 de Mayo de 2009.]

Larman, C. (1999). *Patrones.Introducción al análisis y diseño orientado a objetos*.

GLOSARIO

- Cobro Anticipado: Es el efectivo recibido de clientes sin que haya mediado la contraprestación del servicio o la entrega del producto o de la mercancía.
- Derechos Fiscales: Representan los importes pendientes de recibir del presupuesto del Estado por los pagos en exceso por concepto de impuestos, contribuciones, y de pagos por concepto de seguridad social a corto plazo, pendientes de reintegrar.
- Derecho de cobro: Es un derecho que tiene el acreedor frente al deudor de cobrar lo debido por venta de mercancías o prestación de servicios.
- Entidad: Toda organización administrativa, comercial, económica, productiva y de servicios de carácter estatal, cooperativa, privada o mixta, residentes en el territorio nacional; así como las organizaciones sociales y de masas del país.
- Obligaciones Fiscales: Representan los importes pendientes de recibir del presupuesto del Estado por concepto de impuestos, contribuciones y pagos por concepto de seguridad social a corto plazo pendientes de reintegrar.
- Obligaciones de Pago: Representan los importes de pago a proveedores por operaciones corrientes, independientemente que su pago se efectúe previa o posteriormente a la recepción o aceptación de las mercancías, materiales y servicios recibidos.
- Pago Anticipado: Son los pagos efectuados a los proveedores por la recepción futura de los productos o mercancías, así como por la aceptación posterior de los trabajos o servicios, en virtud de los contratos suscritos o pactos aceptados.
- Umbrales: Valores heurísticas usados para fijar rangos de valores deseables y no deseables de métricas, para el software medido. Se utilizan para identificar anomalías.
- Conciliación: La conciliación es un mecanismo de solución de conflictos a través del cual, dos o más personas gestionan por sí mismas la solución de sus diferencias, con la ayuda de un tercero neutral y calificado, denominado conciliador. La conciliación es un acto jurídico en el cual intervienen sujetos con capacidad jurídica y distintos intereses y en donde su consentimiento y voluntad están dirigidos directamente a dar por terminada una obligación o una relación jurídica,

a modificar un acuerdo existente o a crear situaciones o relaciones jurídicas nuevas que beneficien a ambas partes.

- Submayor: Libro auxiliar de contabilidad que se utiliza para registrar el detalle de los movimientos de las cuentas contables que serán resumidos en el libro Mayor.
- Liquidación: es la acción y efecto de liquidar (hacer el ajuste formal de una cuenta o pagar enteramente una cuenta). Liquidar es saldar una cuenta, pagar una deuda.
- Procesos: Un proceso se define como un conjunto de tareas, actividades o acciones interrelacionadas entre sí que, a partir de una o varias entradas de información, materiales o de salidas de otros procesos, dan lugar a una o varias salidas también de materiales (productos) o información con un valor añadido.

ANEXOS

Anexo 1.Elementos de BPMN.

- Calles:

Piscina: Representa un participante en el proceso.



Carril: Es una subpartición dentro de una piscina, son utilizadas para organizar la información dentro de una piscina.



- Objetos de flujo.

- Actividad.

- Proceso: Es un conjunto de actividades estructuradas para lograr un objetivo de la organización, que convierte un conjunto de elementos de entrada en salidas, según las reglas de negocio.
- Subproceso: Es una actividad compuesta definida como un flujo de otras actividades.
- Tarea: Es una unidad atómica de ejecución que se desarrolla por un único recurso de la entidad.

- Eventos:

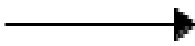


Inicio Intermedio Fin

- Puertas: Son usadas para dividir y unificar el flujo del proceso.

- Objetos de conexión.

- Secuencia: Es usada para mostrar el orden en que las actividades ocurren.

Normal: 

Condicional: 

Por defecto: 

- Mensaje: Es usado para representar el flujo de mensajes entre dos entidades.



- Asociación: Es utilizada para asociar información a los Objetos de flujo.

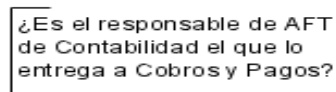


Anexo 2.Artefactos de BPMN.

- Objetos de datos: Objetos de datos. Representan las entradas y salidas a los Objetos de flujo.



- Anotación.: Permiten añadir documentación adicional al proceso de negocio.



- Grupo. Es un grupo de actividades que no afecta el proceso. Los grupos pueden ser usados para documentar o realizar análisis, así como para mostrar una transacción distribuida entre varias piscinas.

