

Universidad de las Ciencias Informáticas

Facultad 4



Título:

**“SISTEMA DE CONTABILIDAD FINANCIERA PARA LA ACTIVIDAD
PRESUPUESTADA EN LAS FUERZAS ARMADAS REVOLUCIONARIAS”
MODULO CONTABILIDAD**

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

Autores:

Yiset Pérez Rizo
Rogervis Pérez Cervantes

Tutor:

Ing. Yunei López Lugo

Consultante: Igris Sebjans

Asesores: Yordanis Milanés
Zenel Reyes

Ciudad de La Habana, Cuba

Julio, 2007

DATOS DE CONTACTO.

Tutor: Yunei Lopez Lugo.

Profesion: Ingeniero Informatico en el Centro de Calculo del MINFAR

Años de Experiencia: 3

Años de Graduado: 3

Asesor: Yordanis Milanes Zamora.

Profesion: Ingeniero Informatico

Categoria Docente: Profesor

Años de Experiencia: 1

Años de Graduado: 1

Asesor: Zenel Reyes Perez.

Profesion: Ingeniero Informatico

Categoria Docente: Profesor

Años de Experiencia: 1

Años de Graduado: 1

Consultante: Igris Sebjans.

Profesion: Ingeniero Informatico en el Centro de Calculo del MINFAR

Años de Experiencia: 12

Años de Graduado: 12

Agradecimientos:

Estrechar cálidamente la mano, ofrecer una amistosa palmada en la espalda o regalar el casi imperceptible parpadeo de los ojos, pudieran convertirse en el “Gracias” mas sincero que humanamente se pudiera dar. Para nosotros, combinar quizás estos gestos con las humildes palabras que tratásemos de redactar en este documento, pudiera dar a luz al intento mas inmediato de transformar esas “Gracias” sinceras y merecidas, en esta sarta de palabras, enorme, pero profunda. Quisiéramos agradecer a los que creyeron en nosotros cuando ni nosotros mismos creímos, a aquellos amigos que nos enseñaron el verdadero valor de la amistad cuando no alcanzamos a pensar lo invaluable que seria tenerlos ahora con nosotros. Gracias a nuestros hermanos y a nuestras parejas, por ser el sostén en la desesperanza y la bruma. Gracias a nuestros padres por representar la luz del faro que guio, en cada tormenta, las velas de este navío que por ellos, hoy comandamos. Gracias a cada irremplazable momento, a cada insuperable recuerdo, a cada insustituible anhelo que como universitarios grabamos en la memoria de lo inolvidable. Gracias también a las horas de duda, a los días grises y a la niebla densa, por hacernos fuertes y seguros, por entrenar y probar nuestra resistencia en cada ataque, por contribuir a la magnifica transformación del capullo en mariposa, del niño en hombre. Gracias a aquella sonrisa desconocida que nos entrego dicha una tarde, gracias al sol, al cielo, a los días que vivimos, a los años que tenemos. Gracias a la gente que nos conoce, y a la que no conocemos también, gracias a las cosas que nos hicieron reír, y a las que nos hicieron llorar, gracias a los días de verano y a los días de inviernos, gracias a la tierra por ser nuestra y por dejarnos vivir en ella. Gracias a Dios y a Fidel, uno por crear el mundo, y otro por transformar el nuestro y hacerlo mucho mejor. Gracias a lo que hoy tenemos, lo que no tuvimos y lo que tendremos, y gracias a nosotros mismos por encontrarnos algún día, por llegar hasta donde hoy estamos y por ser lo que, gracias a muchas cosas, finalmente somos.

Dedicatoria

A nuestros padres, por enseñarnos que en la vida, las grandes cosas se alcanzan a base de sacrificios y empeño. Por darnos la fuerza que hace falta en los momentos de debilidad. Por tendernos la mano firme en las caídas mas profundas. Por abrirnos los ojos cuando el cansancio y la derrota nos lo cerraron. Por ser el ejemplo a seguir..... Y por encima de todo: por ser nuestros padres.

Resumen

La informática y su continuo desarrollo han evolucionado el mundo de manera tal, que ha llegado incluso a sustituir la mano y el pensamiento humanos. En la actualidad, son disimiles los sistemas informatizados que se dedican a facilitar el trabajo del hombre en diferentes esferas y aspectos de la sociedad. Las empresas de gran alcance, debido al gran cumulo de información que manipulan y al gran flujo de acciones que ejecutan, incorporan el uso de las tecnologías de la informática y las comunicaciones a sus quehaceres diarios.

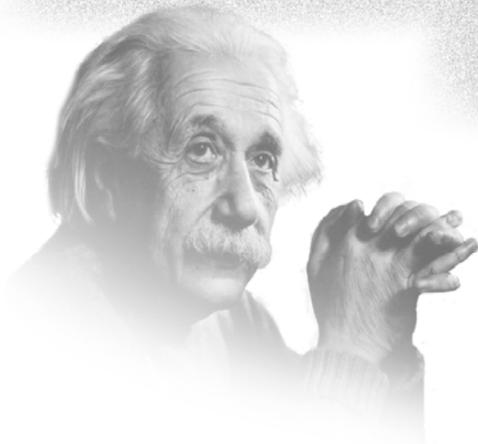
Las FAR, como una empresa mas de nuestro país y por la gran cantidad de documentos que genera la actividad contable financiera, así como por la falta de homogeneidad en cuanto a los procesos de obtención de recuperaciones e inicio de año, decidió, utilizar un sistema que se encargue de facilitar todo este tipo de acciones, emitiendo reportes digitales, fáciles de almacenar y de manipular.

El objetivo de este trabajo es informatizar los procesos de Recuperaciones y de Inicio del Año Contable, así como proponer la utilización del sistema para la ejecución de Traspasos, haciendo uso de las metodologías y herramientas seleccionadas por los usuarios, en conjunto con el equipo de desarrollo, para obtener, de este modo, una solución estándar de un sistema que facilite la actividad presupuestada de las FAR en cada uno de los órganos financieros que le corresponden.

Palabras Claves:

Sistemas informatizados
Actividad contable financiera
Recuperaciones
Reportes
Año Contable
Actividad presupuestada
Órganos financieros

HAY UNA FUERZA MOTRIZ
MÁS PODEROSA QUE EL VAPOR,
LA ELECTRICIDAD Y LA ENERGÍA ATÓMICA:
LA VOLUNTAD.



Índice

AGRADECIMIENTOS:	I
DEDICATORIA	II
RESUMEN	III
INTRODUCCION	1
CAPITULO 1: “FUNDAMENTACION TEORICA”	4
❖ Introducción	4
❖ Flujo actual de los procesos de negocio.	4
❖ Procesos objeto de automatización.	6
❖ Tendencias y tecnologías actuales.	6
❖ Tendencias en Tecnologías de Información.	8
❖ Las tecnologías más importantes para la AICPA (American of Certified Public Accountants)	9
❖ Las aplicaciones mas importantes para la AICPA	10
❖ Los sistemas de Información Integrados ENTERPRISE RESOURCE PLANNING (ERP)	11
❖ Conclusiones:.....	11
CAPITULO 2: “CARACTERISTICAS DEL SISTEMA”	16
❖ Introducción	16
❖ Modelado del Negocio	16
❖ Descripción de los Casos de Usos del Negocio.....	17
❖ Diagramas de Actividades para los Casos de Uso.....	20
❖ Modelo de Objetos	22
❖ Requerimientos Funcionales.	23
❖ Requerimientos no funcionales.	26
❖ Descripción del Sistema Propuesto.....	28
❖ Modelo de Casos de Uso del Sistema.....	30

❖ Descripción de los Casos de Uso del Sistema	32
❖ Conclusiones	37
CAPITULO 3: “ANALISIS Y DISEÑO DEL SISTEMA”	39
❖ Introducción	39
❖ Análisis y Diseño.	39
❖ Análisis:	40
❖ Diseño.	45
❖ Mecanismos de Diseño.	59
❖ Mecanismo de Diseño de Seguridad.	59
❖ Mecanismos de Diseño de Acceso a Datos.....	60
❖ Diagramas de Interacción.	72
❖ Diagramas de Secuencia	72
❖ Diseño de la Base de Datos.	77
❖ Conclusiones:.....	84
CAPITULO 4: “IMPLEMENTACION”	86
❖ Introduccion	86
❖ Diagrama de Componentes:	87
Fig. 58 Diagrama de Componentes II (Distribucion por capas de los componentes)	
.....	93
❖ Diagrama de Despliegue:.....	94
❖ Conclusiones:.....	96
CONCLUSIONES:	97
RECOMENDACIONES:.....	99
BIBLIOGRAFIA.....	100
GLOSARIO DE TERMINOS	101

INTRODUCCION

La contabilidad es el método para organizar los movimientos económicos y financieros de las empresas y personas, brindando una visión detallada y precisa de su situación patrimonial. Es un conjunto de técnicas que se utilizan para producir sistemática y estructuradamente la información cuantitativa expresada en unidades monetarias de las transacciones que realiza una entidad económica y de ciertos eventos económicos cuantificables que la afectan, con objeto de facilitar a los diversos interesados al tomar decisiones en relación con dicha entidad económica.

La contabilidad Financiera es una rama de la contabilidad, la cual presenta y presupone juegos contables con el propósito de suministrar información. Está regulada por normas imperativas de cara a dar una publicidad esencial del funcionamiento y estado financiero de la empresa a todos los agentes económicos interesados (clientes, inversores, proveedores, etc.).

Las Fuerzas Armadas de la Revolución (FAR) es la institución cubana encargada de la defensa del territorio cubano y de preservar las conquistas de la Patria. Además, las FAR funcionan como una empresa más del país aunque sus características la hacen única. Las FAR divide sus actividades contables para dos tipos de empresas: Empresas Presupuestadas (Unidades Militares, hospitales, escuelas, etc.) y Empresas Autofinanciadas (Gaviota, TRD, Geocuba, talleres de autos, etc.). Estas empresas dividen a su vez la contabilidad en dos tipos: Contabilidad Financiera y Contabilidad Material las cuales se rigen según el manual de normas y principios del MINFAR.

El proceso contable de la actividad presupuestada de las FAR genera un gran cúmulo de información. Para ello existe un sistema que se encarga del registro y emisión de la información mediante el uso de la computadora, pero basado en una tecnología obsoleta. El mismo no se explota en todas las empresas presupuestadas pertenecientes a las FAR, algunas realizan estas actividades manualmente o en tablas Excel, por lo que no existe una homogeneidad en el proceso contable financiero lo que dificulta que actividades como algunas recuperaciones y el inicio de año no se realicen de la forma más óptima, lo que propicia que se incurran en errores de cálculo, comprometiendo la veracidad de los datos.

Teniendo en cuenta lo antes expuesto y enmarcado en el objeto de estudio que proponen los procesos contables financieros automatizados en las empresas pertenecientes a las FAR, cabría preguntarse: ¿Cómo erradicar los problemas de los procesos contables de las recuperaciones e inicio de año mediante la automatización de un sistema de gestión?

A partir de esta interrogante formulada, se decidió trazar como objetivos fundamentales de este trabajo de tesis, realizar el análisis y diseño del sistema informático para la automatización de los procesos contables financieros, específicamente al paquete dedicado a las recuperaciones e inicio de año, y presentar el modulo de traspasos como una nueva funcionalidad de este sistema, constituyendo además, una mejora para la realización de la actividad económica en las empresas pertenecientes a las FAR.

Para guiar este trabajo se ha formulado la siguiente hipótesis: Si se desarrolla una solución de software estándar para la gestión de los procesos contables de las empresas pertenecientes a la FAR, entonces se logrará una mayor eficiencia de los procesos contables financieros de recuperaciones e inicio de año en las empresas pertenecientes a las FAR sin temor a errores de cálculo, permitiendo conocer la verdadera situación económica de la empresa.

Tras haber realizado un análisis del objetivo fundamental de este trabajo, se pueden desglosar los siguientes objetivos específicos:

- Describir los casos de uso del sistema de los paquetes de Traspasos (nueva funcionalidad que nuestro sistema propone), Recuperaciones e Inicio del Año Contable.
- Realizar los modelos de Análisis y Diseño correspondientes a cada uno de estos paquetes

Para alcanzar dichos objetivos, se han trazado las siguientes tareas de investigación:

1. Estudio de los principios económicos que rigen el negocio de la contabilidad presupuestada de las empresas pertenecientes a la FAR.
2. Indagación sobre una propuesta de arquitectura, acorde con las condiciones de las empresas pertenecientes a las FAR y que permitan la implantación del mismo al menor costo posible, para la automatización de procesos contables.

Capítulo



“Fundamentación Teórica”

Introducción

Flujo Actual de los procesos del negocio

Procesos objeto de automatización

Tendencias y Tecnologías actuales

Conclusiones

CAPITULO 1: “FUNDAMENTACION TEORICA”

❖ Introducción

La Contabilidad como la ciencia de la Información Empresarial a lo largo de su historia ha evolucionado buscando siempre satisfacer los requerimientos de los usuarios, sin embargo es notorio su estancamiento en las últimas décadas al no haber modificado la Información Financiera que reporta con relación a la evolución de la economía globalizada, la administración, la velocidad de los cambios tecnológicos y el incremento desmesurado del conocimiento.

En este último período, la contabilidad y la profesión contable ve cuestionada la Información Financiera que reporta a los usuarios, ya que estos estados financieros tradicionales que fueron diseñados para cumplir con los requerimientos de la era industrial, no exponen ni reflejan adecuadamente el valor real de las empresas, por lo que, este modelo tradicional que fue útil en su oportunidad muestra problemas fundamentales tales como:

- No mostrar el verdadero valor de las empresas que permita la toma de decisiones racionales de los usuarios.
- No identificar las fuentes de valor puestas a trabajar en una era virtual y de una economía del conocimiento.

En este capítulo se analizará las diferentes tecnologías de impacto que vienen afectando a la contabilidad, se plantearán los principales conceptos relacionados con el proceso de contabilidad financiera para la actividad presupuestada en las FAR, y se dará una breve panorámica de las tecnologías y tendencias actuales a tener en cuenta para la elaboración del proyecto con el propósito de decidir la tecnología a utilizar.

❖ Flujo actual de los procesos de negocio.

Actualmente el MINFAR cuenta con un sistema de contabilidad regido por el manual de normas y principios establecidos por el Ministerio. El administrador principal, anualmente realiza operaciones (asignación, distribución, etc.) a sus órganos subordinados.

- No cuenta con un entorno visual usuario-sistema. Al ser implementado en un software de programación bastante obsoleto, la calidad de la interfaz con el usuario no es la mejor y brinda muy pocas facilidades, lo que hace más engorroso el trabajo.
- Los formatos de salida de los reportes no están acorde con los estándares establecidos lo que provoca que no esté certificado.

Se hizo el estudio correspondiente a una primera versión que se elaboró en la unidad de servicio de las FAR UM 9001, que respondía a intereses del sistema de contabilidad anterior que existía en las FAR, este sistema se hizo en FoxPro 2.6 sobre MS-DOS. Entre los años 1999-2002 el Ejército Central continuó modificando y concluyendo esta versión.

A finales del año 2003, con el cambio de la nueva orden que tuvo el sistema de Contabilidad Financiera, el Ejército Central hizo una versión en Visual FoxPro, respetando el análisis y diseño del sistema anterior, solo haciendo algunas modificaciones en cuanto al clasificador de cuentas y grupos presupuestarios, así como la implementación de algunas recuperaciones acorde a la Orden 379/03.

❖ Procesos objeto de automatización.

- Inicio del Año
- Recuperaciones
- Traspasos

❖ Tendencias y tecnologías actuales.

El impacto de la tecnología en la profesión contable está fuera de toda duda. Las tecnologías de la información operan como motor del cambio que permite dar respuesta a las nuevas necesidades de información. Las tecnologías de la información son el motor del cambio que conduce a una nueva era postindustrial que amenaza con dejar obsoletas todas las estructuras empresariales que no sepan adaptarse.

En cuanto al impacto de las tecnologías de la información en las prácticas y sistemas de información contables es necesario distinguir entre aquellas mejoras que sólo afectan a aspectos cuantitativos y aquellas que permiten que nuevos métodos de trabajo contable puedan llevarse a cabo o permiten diseñar nuevos sistemas de información. Entre los primeros se puede citar la mayor velocidad que supone utilizar un soporte informático para obtener los balances o la posibilidad de depositar las cuentas en los registros mercantiles o liquidar los impuestos a través de procedimientos de intercambio electrónico de información. Esto es muy positivo por el ahorro en tiempo y dinero que supone pero no introduce cambios metodológicos.

Un ejemplo, Microsoft: el “sistema nervioso digital” y la Contabilidad en un artículo publicado por Scott M. Boggs, controlador de Microsoft, en mayo de 1999 en el Journal of Accountancy, comentan que hace cinco años, a Microsoft le costaba dos semanas de cada mes cerrar sus libros y ahora cuatro días. El objetivo previsto es lograr un cierre de libros continuo y diario. 350.000 copias en papel de informes de gestión han sido reemplazadas por documentos online. Hoy no generan ni una hoja de papel.

Ninguna de la tecnología que ellos usan esta fuera del alcance de una empresa, grande o pequeña. Utilizan una intranet, llamada Microsoft Fin Web (Financial Information Network) un data warehouse o almacén de datos que extrae información del diario y la pone accesible en una base de datos de Microsoft Access. Otras herramientas son tablas dinámicas de Excel (PivotTable) que permite profundizar en los detalles de cada división, producto, mercado, etc., que se complementa con consultas SQL mediante una herramienta llamada MS Report.

Otra herramienta interesante es MAP (Microsoft Accounting Policies), una parte de la intranet que archiva las políticas contables de la empresa, sobre 500 aspectos (por ejemplo, cómo contabilizar los gastos de automóvil, Cisco Systems el fabricante líder en Internet de componentes, redes, etc., cierra sus libros de Contabilidad en ¡1 día!, y publica los resultados en Internet seis días después.

❖ Tendencias en Tecnologías de Información.

Las tecnologías de la información están cambiando la forma en que los contadores trabajan y el marco en el que se desarrollan sus actividades, porque han transformado la forma en que las empresas se organizan y gestionan. De acuerdo con el IFAC (International Federation of Accountants), son particularmente dignas de mención las siguientes tendencias en Tecnologías de la Información (TI):

- La amplia disponibilidad de potentes pero económicos equipos informáticos.
- La amplia disponibilidad de programas de ordenadores potentes, económicos y relativamente sencillos de usar con interfaces de usuario gráficas.
- El paso de adquirir sistemas informáticos a medida en vez de utilizar software “llave en mano”
- El paso de grandes equipos centralizados o “mainframes” en lugar de ordenadores personales, utilizados solos o como parte de redes.
- La disponibilidad creciente de datos informatizados a los que se accede en tiempo real o diferido, mediante acceso local o remoto, incluido vía Internet.
- Las nuevas tecnologías de captura de datos y almacenamiento llevan a la informatización de datos e información en formato de texto, gráficos, audio y video y enfatizan el gestionar, presentar y comunicar información utilizando técnicas de multimedia.
- La convergencia de las tecnologías de la información y la comunicación, que afecta el cómo las personas trabajan y compran.
- El uso creciente de redes que conectan a los individuos y a las empresas- intra-empresas e inter-empresas con sistemas como el correo electrónico e Internet, incluyendo el World Wide Web.
- El uso creciente de Internet para el comercio entre empresas y la venta de productos finales a un consumidor, vía sistemas de comercio electrónico como el Intercambio Electrónico de Datos (EDI) y Sistemas de Transferencia Electrónica de Fondos (EFTS).
- El marketing y la distribución masiva de productos de tecnologías de la información y servicios tales como ordenadores, programas informáticos “llave en mano”, servicios de recuperación de datos on-line, correo electrónico y servicios financieros.

- La reducción de barreras al uso de sistemas, la prometedora y más amplia incorporación de sistemas de información en organizaciones lucrativas y no lucrativas de todo tipo para temas de contabilidad, de gestión y estratégicas, y el uso creciente de la informática de usuario final.
- Las nuevas técnicas de desarrollo de sistemas basadas en tecnologías de información como las herramientas de ingeniería del software (CASE), programación orientada a objetos y tecnologías work-flow.
- Desarrollo continuado de sistemas de ayuda inteligentes que incorporan sistemas expertos, redes neuronales, agentes inteligentes y otras técnicas de resolución de problemas.
- Nuevas metodologías de reingeniería de negocios basadas en la integración efectiva de tecnologías de información y procesos de negocio.

❖ Las tecnologías más importantes para la AICPA (American of Certified Public Accountants)

Otras instituciones también recomiendan que los contadores dominen ciertas tecnologías. Por ejemplo, The American Institute of Certified Public Accountants (AICPA) realiza un congreso anualmente al que asisten contadores públicos expertos en tecnologías y votan sobre cuales son las tecnologías de mayor impacto para la profesión contable.

Teniendo en cuenta los datos arrojados este ultimo año, podemos enumerar las siguientes tecnologías:

- Tecnologías relativas a la seguridad de los sistemas de información: Son actividades relacionadas con las copias de seguridad, encriptar un fichero u obtener una firma digital, es decir saber enfrentar los riesgos y amenazas de errores humanos, fallas en los equipos, robo, virus, sabotaje, fraude, etc.
- XML (Extensible Markup Language): Esta tecnología puede lograr que Internet se convierta en una gigantesca base de datos.
- Ancho de banda: Constituye la máxima cantidad de datos que pueden pasar por un medio de comunicación en un momento dado, medido en segundos.

- Tecnologías móviles: (como los teléfonos móviles con WAP o UMTS, ordenadores portátiles, ordenadores que caben en la palma de la mano). Esta tecnología está llamada a ser una de las tecnologías clave en el futuro próximo, con muchas aplicaciones para que los auditores, asesores, contables puedan trabajar o dar servicio a los clientes.
- Tecnología Inalámbrica: Trata de facilitar la comunicación entre dispositivos sin la utilización de cables, aprovechando la movilidad de los dispositivos inalámbricos.
- Verificación electrónica: Permite lograr la autenticación.
- Encriptación: Traduce los signos de un mensaje a otro sistema de signos cuya lectura no tenga ningún sentido para un desconocido que los intercepte.
- Autorización electrónica: Son soluciones técnicas que permiten que muchas tareas se aprueben electrónicamente, desde contratos a órdenes de compra.
- Herramientas de conectividad remota: Son tecnologías que permiten el acceso y control remoto de ordenadores. Un profesional puede acceder a un ordenador remoto desde su despacho o desde su casa y ejecutar un programa de gestión, contabilidad, etc.
- Bases de Datos: Tecnología usada para mantener las bases de datos con información relevante y actualizada.

❖ Las aplicaciones más importantes para la AICPA

En cuanto a aplicaciones, el AICPA se refiere a las 10 más entendidas como oportunidades de negocio que utilizan una o varias tecnologías de las que más impacto puede tener en la Contabilidad. A continuación se enumeran las siguientes aplicaciones:

- Aplicaciones basadas en el Web
- Gestión del conocimiento (los sistemas para gestionar el conocimiento incluyen aplicaciones de programas, diseñadas para optimizar el trabajo en grupos).
- Transmisión de información financiera (los nuevos estándares de Internet permitirán preparar, publicar, intercambiar y analizar informes financieros en una variedad de formatos).
- Aplicaciones Web-Enabled.
- Detección de intrusos

- Data Mining (consiste en explorar automáticamente grandes bases de datos para extraer patrones de comportamiento, relaciones ocultas entre las variables e identificar dependencias).
- Gestión de documentos (la mayor parte de las empresas siguen imprimiendo casi todos los documentos que generan. Para la empresa supone un coste muy elevado crear, gestionar y distribuir documentos. Estas herramientas permiten publicar, distribuir y gestionar grandes cantidades de información en la empresa utilizando varias tecnologías).
- Auditoria continua.
- Aprovisionamiento electrónico.
- Depósito electrónico de documentos contables o fiscales.

❖ Los sistemas de Información Integrados ENTERPRISE RESOURCE PLANNING (ERP)

Más que programas de ordenador son sistemas de información que integran aplicaciones informáticas para gestionar todos los departamentos y funciones de una empresa: contabilidad financiera y analítica, finanzas, producción, mantenimiento, logística, recursos humanos, materiales, gestión de activos, compras y pagos, ventas y cobros, bancos y efectos, tesorería, cartera, gestión de proyectos, etc.

❖ Conclusiones:

Según lo descrito en este capítulo y valorando las tecnologías y aplicaciones mas empleadas a nivel internacional en cuanto al mundo contable se refiere, se concluye que según las condiciones materiales que las FAR posee y las características específicas que como empresa tiene, se puede decir que de las 10 tecnologías mencionadas, las siguientes forman parte de la creación de este sistema automatizado:

- Verificación electrónica
- Encriptación
- Autorización electrónica
- Herramientas de conectividad remota
- Bases de Datos

Analizando de este mismo modo las condiciones que propone el MINFAR, se decidió que de las 10 aplicaciones descritas, las siguientes formaran parte de este sistema:

- Aplicaciones basadas en el Web
- Gestión del conocimiento
- Transmisión de información financiera
- Data Mining
- Gestión de documentos

En la implementación del sistema se utilizara, como lenguaje de programación el PHP 5 pues dentro de las ventajas que brinda se pueden mencionar las siguientes:

- Language.
- Alta capacidad de conexión con la mayoría de los manejadores de base de datos que se utilizan en la actualidad.
- Potente variedad de extensiones para el acceso a la mayoría de los sistemas de gestión de bases de datos.
- Es libre, por lo que se presenta como una alternativa de fácil acceso para todos.
- Permite las técnicas de Programación Orientada a Objetos.
- Permite crear formularios para la Web.
- Posee una biblioteca nativa de funciones sumamente amplia.

También, en la implementación del lado del cliente, se empleara Java Script que posee ventajas como:

- Se puede utilizar para construir sitios Web y para hacerlos más interactivos.
- Es un lenguaje interpretado, es decir, que no requiere compilación.
- Puede interactuar con el código HTML, permitiendo a los programadores Web utilizar contenido dinámico. Por ejemplo, hace fácil responder a los acontecimientos iniciados por usuarios (como introducción de datos en formularios) sin tener que utilizar CGI.

Como gestor de bases de datos se utilizara el PostgreSQL 8.0 porque tiene características ventajosas como:

- Es un servidor de base de datos relacional libre. Está considerado como la base de datos de código abierto más avanzada del mundo.
- Posee instalación ilimitada
- Tiene mucho mejor soporte que los proveedores comerciales
- Brinda la posibilidad de ahorros considerables en costos de operación
- Brinda estabilidad y confiabilidad
- Es extensible
- Es multiplataforma
- Fue diseñado para ambientes de alto volumen
- Presenta herramientas gráficas de diseño y administración de bases de datos

La metodología a seguir será RUP porque presenta ventajas como:

- Es el resultado de varios años de desarrollo y uso práctico en el que se han unificado técnicas de desarrollo a través del UML, y trabajo de muchas metodologías utilizadas por los clientes.
- Dirigido por casos de uso que reflejan lo que los usuarios futuros necesitan y desean, por lo que guían el proceso de desarrollo ya que los modelos que se obtienen, como resultado de los diferentes flujos de trabajo, representan la realización de los casos de uso.
- Centrado en la arquitectura que muestra la visión común del sistema completo en la que el equipo de proyecto y los usuarios deben estar de acuerdo, por lo que describe los elementos del modelo que son más importantes para su construcción, los cimientos del sistema que son necesarios como base para comprenderlo, desarrollarlo y producirlo económicamente.
- Es un proceso Iterativo e Incremental, que propone que cada fase se desarrolle en iteraciones que involucren actividades de todos los flujos de trabajo, aunque desarrollen fundamentalmente algunos flujos más que otros.

Los desarrolladores se apoyaran en el lenguaje de modelación UML para lograr un trabajo de calidad y en tiempo pues este lenguaje tiene características que se traducen en las siguientes ventajas:

- Es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad
- Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema de software.
- Ofrece un estándar para describir un "plano" del sistema (modelo)
- Incluye aspectos conceptuales tales como procesos de negocios y funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y componentes de software reutilizables.
- Se utiliza para definir un sistema de software, para detallar los artefactos en el sistema y para documentar y construir
- Se puede aplicar en una gran variedad de formas para soportar una metodología de desarrollo de software (tal como el Proceso Unificado de Rational).

Y, finalmente, como herramienta CASE de modelado se empleara el Visual Paradigm porque:

- Ofrece un entorno de creación de diagramas para UML, un diseño centrado en casos de uso y enfocado al negocio que generan un software de mayor calidad
- Hace uso de un lenguaje estándar común a todo el equipo de desarrollo que facilita la comunicación; capacidades de ingeniería directa (versión profesional) e inversa.
- Garantiza que el modelo y el código permanezcan sincronizados en todo el ciclo de desarrollo.
- Garantiza la disponibilidad de múltiples versiones, para cada necesidad.
- Garantiza la disponibilidad en múltiples plataformas
- Es muy útil para la generación de código fuente en PHP.

Capítulo



“Características del Sistema”

Introducción

Modelado del Negocio

Descripción de los Casos de Uso del Negocio

Diagramas de Actividades para los Casos de Uso del Negocio

Modelo de Objetos

Requisitos Funcionales

Requisitos No Funcionales

Descripción del Sistema Propuesto

Modelo de Casos de Uso del Sistema

Descripción de los Casos de Uso del Sistema

Conclusiones

❖ Introducción

En este capítulo se presenta una propuesta de lo que será el sistema a partir de análisis realizados previamente al negocio, detallando la descripción de los procesos que lo componen y analizando cuáles de estos procesos se automatizarán para conformar en su conjunto lo que sería el producto final.

En este capítulo aparecen también enumerados los requisitos funcionales y no funcionales del sistema a desarrollar, logrando de este modo las características imprescindibles que debe tener el mismo una vez que haya sido terminado.

❖ Modelado del Negocio

El Modelo de Casos de Uso del Negocio implicará la determinación de los actores y casos de uso del negocio. Con esta actividad se pretende:

- Comprender la estructura y la dinámica de la organización en la cual se va a implantar un 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.
- Identificar los procesos en el negocio
- Definir las fronteras del negocio que va a modelarse
- Definir qué y quiénes interactuarán con el negocio
- Crear diagramas del modelo de casos de uso del negocio

En este trabajo se tiene como actor del negocio el Directivo, que sería la persona que solicitaría la emisión de un reporte determinado o la realización de la Apertura de un Año Contable, y se presentan 2 casos de uso principales denominados: Emitir Reporte y Apertura del Año Contable.

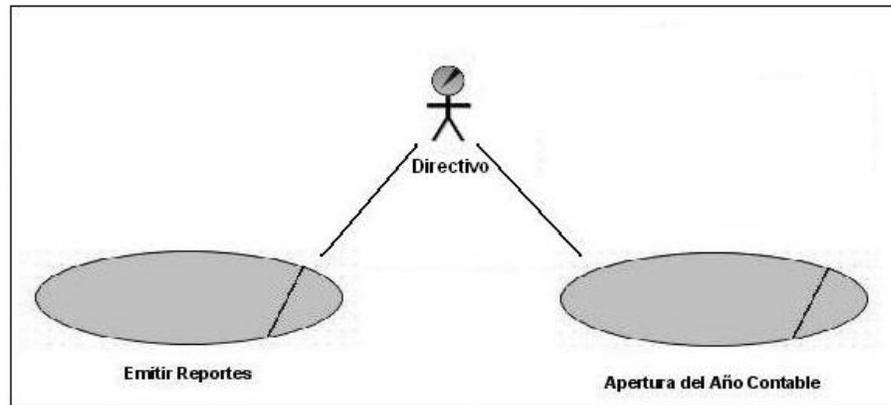


Fig. 2 Diagrama de Casos de Uso del Negocio.

❖ Descripción de los Casos de Usos del Negocio

La descripción textual de un caso de uso del negocio se formaliza en un documento generalmente llamado “Especificación del Caso de Uso del Negocio”.

A partir de la información reflejada en dicha plantilla, se construye un conjunto de diagramas que describen completamente el caso de uso del negocio. A continuación aparecen las descripciones de los casos de uso correspondientes al negocio que enmarca este trabajo.

Nombre del CU del Negocio:	<u>Emitir Reporte</u>
Actores del negocio:	Directivo
Trabajadores del negocio:	Contador Principal.
Resumen:	Mostrar los reportes a partir de las condiciones seleccionadas por el actor. Facilitar la información necesaria para registrar, de forma resumida, los hechos económicos que tienen lugar en un órgano financiero.
Acción del actor	Respuesta del negocio
<ol style="list-style-type: none"> 1. El cliente solicita el tipo de información que quiere obtener. 3. El actor comunica de las condiciones a tener en cuenta para procesar la información. 	<ol style="list-style-type: none"> 2. El contador le pide las condiciones de la información a procesar. 4. Utilizando las condiciones que el usuario solicita, se le hace un conjunto de operaciones a los comprobantes que cumplen con las condiciones del filtro. 5. El contador, a las cuentas contables afectadas en la información a generar, le realiza la operación clásica de la contabilidad: <ul style="list-style-type: none"> 5.1 Si la cuenta es deudora, (que aumenta por el debe y disminuye por el haber) el saldo será el acumulado del debe – haber. 5.2 Si el resultado es mayor que 0 se pone en la columna del debe y sino se pone al haber. 5.3 Si la cuenta es acreedora (que aumenta por el haber y disminuye por el debe), el saldo de la cuenta será el acumulado del haber – debe 5.4 Si el resultado es mayor que 0 se pone en la columna del haber y sino se pone al debe. 6. La información se muestra en el modelo correspondiente.
Mejoras:	Todas las operaciones realizadas a los comprobantes, se harán se forma automática según las condiciones del filtro, así como la generación de los modelos correspondientes.
Prioridad:	Crítica

Nombre del CU del Negocio:	Apertura del Año Contable
Actores del negocio:	Directivo
Trabajadores:	Contador Principal.
Resumen	Al iniciarse la actividad económica en una unidad determinada, se procede a la apertura de un conjunto de cuentas y libros de contabilidad con el objetivo de comenzar a registrar las operaciones en el año que comienza, se realiza el traspaso de los saldos de algunas cuentas que caducaron en el año que recién finalizó a cuentas que se habilitan para recoger esta información en el año nuevo y de este modo mantener el control sobre las operaciones económicas que realizara dicha unidad en el ejercicio económico que se va a iniciar. Esta actividad permite el conocimiento necesario sobre el estado económico de la unidad a la hora de iniciar la actividad económica en un nuevo período.
Acción del actor	Respuesta del negocio
<ol style="list-style-type: none"> 1. Se solicita hacer la apertura del año. 	<p>Si la unidad no es de nueva creación, o sea, que la apertura constituye la continuidad del ejercicio contable de un año a otro, se ejecutan los siguientes pasos.</p> <ol style="list-style-type: none"> 2. Se realiza la apertura de cuentas: <ol style="list-style-type: none"> 2.1 Se fijan los saldos con los cuales algunas cuentas reales cerraron las operaciones del año que termina. 2.2 Se fijan en las cuentas de orden, el financiamiento no utilizado del año anterior y los importes de los compromisos pendientes de pago. 2.3 Se fijan los saldos de los submayores. 2.4 Se traslada el modelo de "Ingresos Cobrados y Transferidos" al libro que se habilita para registrar las operaciones del nuevo año. 2.5 Se fijan algunas cuentas contra la Contrapartida de las Cuentas de Orden. 3. Se realiza la apertura de los libros de contabilidad. <ol style="list-style-type: none"> 2.1 Se abren algunas cuentas con saldos al cierre del ejercicio anterior contra la cuenta de Inversión Estatal. 2.2 Se confecciona una hoja de trabajo para relacionar los financiamientos no utilizados del año anterior. 2.3 El financiamiento no utilizado del año anterior se debita a las cuentas de orden y se acredita a las cuentas de financiamiento por recibir. 4. Informa al actor que las condiciones para la apertura del ejercicio económico para el nuevo año, han sido creadas correctamente y se registra un comprobante de operaciones con los detalles del cierre.
Mejoras:	Todas las actividades relacionadas con la apertura y acreditación de cuentas, así como el trabajo con los libros de contabilidad, se harán con la ayuda del sistema, que generará los modelos y notificaciones correspondientes de forma automática.
Prioridad:	Crítica.

❖ Diagramas de Actividades para los Casos de Uso

Estos diagramas de actividades describen los procesos que exploran el orden de las tareas o actividades que logran los objetivos del negocio. Son similares a los diagramas de estados en los cuales todos o la mayoría de los estados son estados de actividad y en los que todas o la mayoría de las transiciones se disparan al completarse las acciones en los estados fuentes precedentes.

A continuación aparecen los diagramas de actividades correspondientes a los casos de uso identificados en el negocio.

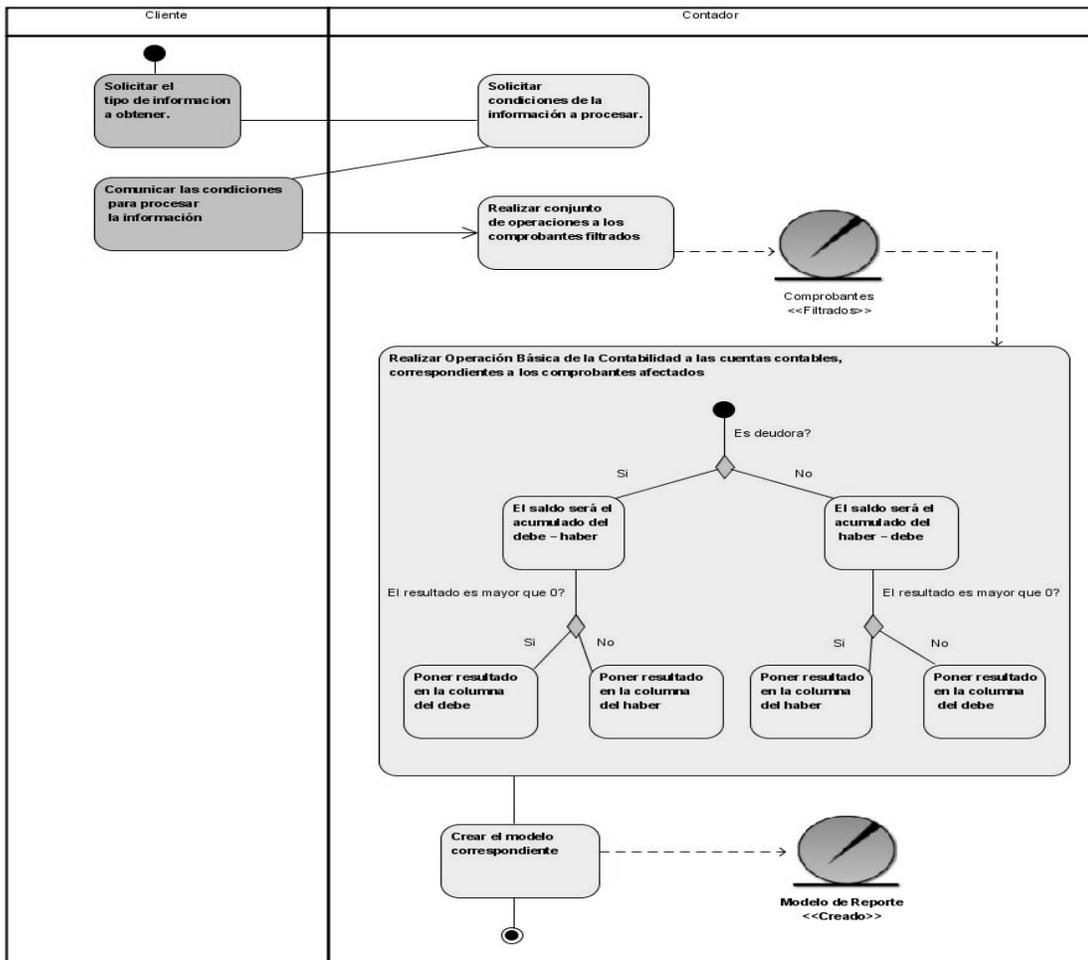


Fig. 3 Diagrama de Actividades para el Caso de Uso: "Emitir Reporte"

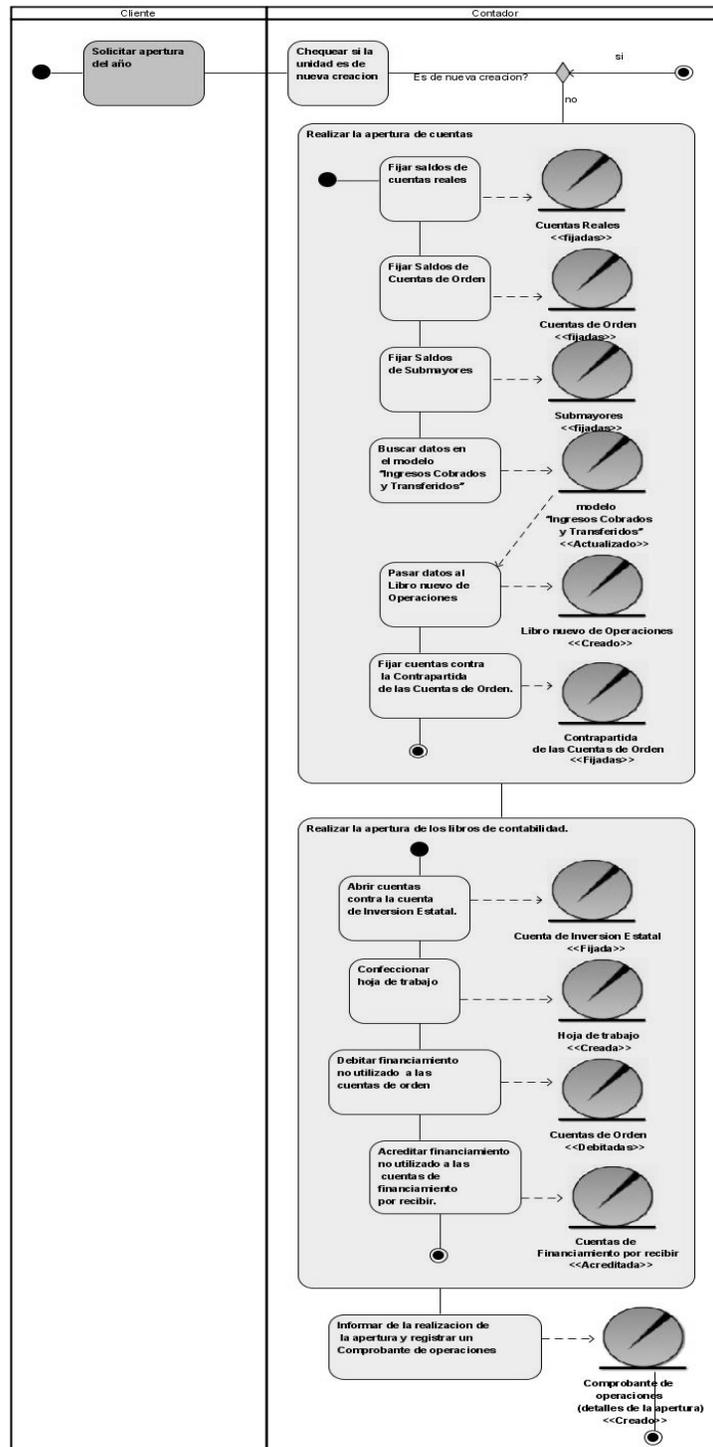


Fig. 4 Diagrama de Actividades para el Caso de Uso: “Apertura del Año Contable”

❖ Modelo de Objetos

Estos modelos de objetos muestran las relaciones que existen entre los trabajadores del negocio y las entidades que se manipulan en los procesos que se describen en el mismo. No es necesario realizar un Modelo de Objetos por caso de uso, estos diagramas se efectúan por cada escenario que se identifique en el negocio, y cada escenario está dado por los trabajadores y las entidades que intervienen en él.

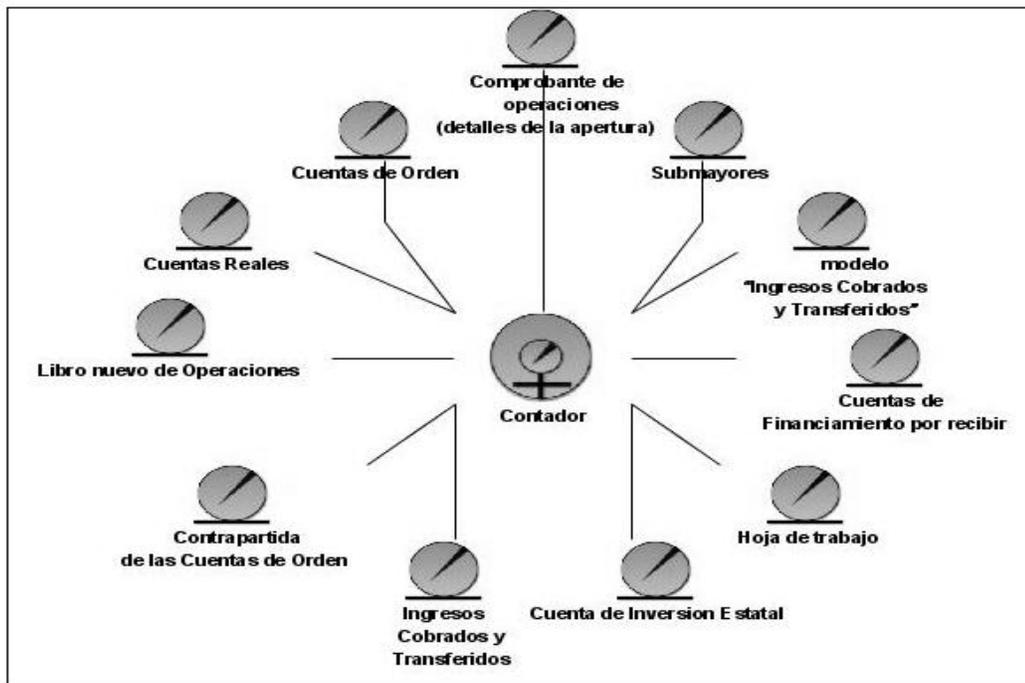


Fig.5 Modelo de Objetos. Escenario (Apertura del Año Contable)

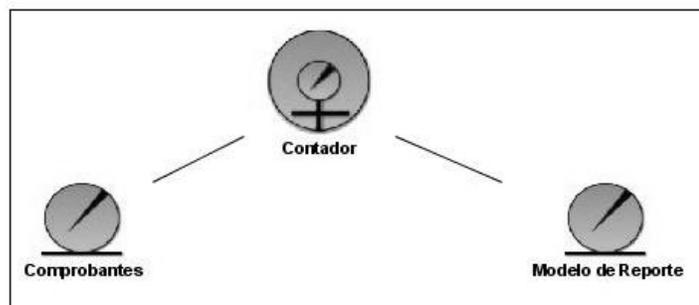


Fig.6 Modelo de Objetos. Escenario (Emitir Reporte)

❖ Requerimientos Funcionales.

Los requerimientos funcionales son capacidades o condiciones que el sistema debe cumplir. En la realización de los casos de uso del negocio, se obtienen las actividades que serán objeto de automatización. Estas actividades no son exactamente los requerimientos funcionales, pero si son el punto de partida para identificar qué debe hacer el sistema.

Los requerimientos funcionales no alteran la funcionalidad del producto, o sea, que se mantienen invariables sin importarle con qué propiedades o cualidades se relacionen.

Teniendo en cuenta lo antes planteado, podemos decir que los Requerimientos Funcionales que identificamos en este proyecto son:

- **R1.** Realizar Inicio del año contable.
 - ✓ R1.1 Realizar la apertura de cuentas.
 - R1.1.1 Fijar el saldo de cuentas
 - R1.1.2 Trasladar la información de un modelo a otro.
 - R1.1.3 Registrar la información en modelos específicos.
 - ✓ R1.2 Realizar la apertura de los libros de contabilidad.
 - R2.1.1 Abrir cuentas a partir de la información de otras cuentas cerradas en el ejercicio económico que finalizó.
 - R2.1.2 Confeccionar una hoja de trabajo para la documentación de las actividades económicas.
 - R2.1.3 Debitar y acreditar un conjunto de cuentas específicas.
- **R2.** Permitir al usuario autenticarse al ingresar al módulo.
- **R3.** Permitir al sistema la cancelación de las acciones
- **R4.** Permitir la realización de una salva de un conjunto de entidades.
- **R5.** No permitir que el usuario modifique la información que consulta para realizar el inicio.
- **R6.** Permitir escoger el rango de fecha para realizar el filtro de la información
- **R7.** Mostrar registro de operaciones por los siguientes criterios:
 - Numero de Comprobante de operaciones
 - Rango de fechas

- Tipo de operación
 - Grupo presupuestario
 - Especialidad
 - Partida de gasto
 - Ingresos (Si ha sido transferido o no)
- **R8.** Permitir seleccionar el tipo de registro contable para realizar el filtro de la información
 - **R9.** Permitir la especificación del tipo de operación a realizar sobre el registro contable filtrado.
 - **R10.** Permitir la creación de los modelos correspondientes a la operación seleccionada.
 - **R11.** Mostrar al usuario el modelo generado por la operación escogida por el.
 - **R12.** Realizar Traspaso (Anexo 1)
 - **R13.** Realizar traspaso (Estado A)
 - **R14.** Realizar traspaso (Estado B)
 - **R15.** Realizar traspaso (Estado C)
 - **R16.** Mostrar Comprobante de Operaciones vinculado al inicio del Año Contable.
 - **R17.** Mostrar Mayor
 - ✓ R17.1 Permitir seleccionar el tipo de cuenta para realizar el filtro de la información
 - ✓ R17.2 Realizar un acumulado general con las cifras pertenecientes al debe y al haber correspondientemente.
 - ✓ R17.3 Permitir al sistema mostrar el modelo del Libro del Mayor.
 - R18.** Mostrar Libro Control
 - ✓ R18.1 Permitir seleccionar el tipo de concepto para realizar el filtro de la información
 - ✓ R18.2 Permitir seleccionar el grupo presupuestario para realizar el filtro de la información
 - ✓ R18.3 Permitir seleccionar el tipo de especialidad para realizar el filtro de la información
 - ✓ R18.4 El sistema llevará un acumulado por mes y otro para el total general
 - ✓ R18.5 El sistema mostrará el modelo del libro de Registro y Control del Presupuesto
 - R19.** Realizar la operación clásica de la contabilidad en el caso que se especifique.
 - ✓ R25.1 Conocer la naturaleza de la cuenta.
 - ✓ R25.2 Calcular el saldo a partir de la naturaleza de la cuenta.

R20. Mostrar Submayor de Gasto

- ✓ R20.1 Permitir seleccionar el tipo de concepto para filtrar.
- ✓ R20.2 Permitir seleccionar el grupo presupuestario.
- ✓ R20.3 Permitir seleccionar el tipo de entidad.
- ✓ R20.4 Permitir seleccionar el tipo de suma.
- ✓ R20.5 El sistema mostrara el modelo del Submayor de Gasto.

R21. Mostrar Submayor de Ingreso

- ✓ R21.1 Permitir seleccionar el tipo de recursos financieros.
- ✓ R21.2 Permitir seleccionar el tipo de órgano.
- ✓ R21.3 Permitir seleccionar el tipo de suma.
- ✓ R21.4 El sistema mostrará el modelo del Submayor de Ingreso.

R22. Mostrar Submayor de Retención

- ✓ R22.1 Permitir seleccionar el tipo de retención.
- ✓ R22.2 Permitir seleccionar el tipo de órgano.
- ✓ R22.3 Permitir seleccionar el tipo de suma.
- ✓ R22.4 El sistema mostrará el modelo del Submayor de Retención.

R23. Mostrar Balance

- ✓ R23.1 Introducir inicio y fin del período
- ✓ R23.2 Mostrar el modelo del Balance.

R24. Mostrar Control Presupuesto

- ✓ R24.1 Consultar el Presupuesto Asignado
- ✓ R24.2 Consultar el Presupuesto Distribuido
- ✓ R24.3 Consultar el Presupuesto de Reserva
- ✓ R24.4 Consultar el Presupuesto por Especialidad
- ✓ R24.5 Análisis del Presupuesto por UM
- ✓ R24.6 Mostrar el modelo de libro de Registro y Control del Presupuesto

❖ Requerimientos no funcionales.

Los requerimientos no funcionales son propiedades o cualidades que el producto debe tener, son las características que hacen al producto atractivo, usable, rápido o confiable.

En muchos casos los requerimientos no funcionales son fundamentales en el éxito del producto. Normalmente están vinculados a requerimientos funcionales, es decir una vez se conozca lo que el sistema debe hacer se puede determinar cómo ha de comportarse, qué cualidades debe tener o cuán rápido o grande debe ser.

Los requerimientos no funcionales identificados forman una parte significativa de la especificación. Son importantes para que clientes y usuarios puedan valorar las características no funcionales del producto, pues si se conoce que el mismo cumple con la toda la funcionalidad requerida, las propiedades no funcionales, como cuán usable, seguro, conveniente y agradable, pueden marcar la diferencia entre un producto bien aceptado y uno con poca aceptación.

Para este sistema, se identificaron los siguientes requisitos no funcionales, especificando la categoría a la que pertenecen:

➤ Apariencia o interfaz externa:

- ✓ Diseño sencillo permitiendo que no sea necesario mucho entrenamiento para utilizar el sistema.
- ✓ Empleo de los colores: verde, gris, blanco y azul principalmente, que son los definidos en los estándares del proyecto.

➤ Usabilidad:

- ✓ El sistema podrá ser usado por cualquier persona que posea conocimientos básicos en el manejo de la computadora y de contabilidad financiera.
- ✓ El software tendrá siempre la posibilidad de ayuda disponible para cualquier tipo de usuario, lo que le permitirá un avance considerable en la explotación de la aplicación en todas sus funcionalidades.

➤ Rendimiento:

- ✓ Tiempos de respuestas rápidos al igual que la velocidad de procesamiento de la información, no mayor a los 5 segundos en las actualizaciones y no mayor de 20 para las recuperaciones.

➤ Soporte:

Se requiere un servidor de base de datos con las siguientes características:

- ✓ Soporte para grandes volúmenes de datos y velocidad de procesamiento.
- ✓ Tiempo de respuesta rápido en accesos concurrentes.
- ✓ Versión de PHP 5.0.
- ✓ Por parte del cliente se requiere un navegador capaz de interpretar JavaScript.

➤ Portabilidad:

- ✓ Necesidad de que el sistema sea multi-plataformas.

➤ Seguridad:

- ✓ Autenticación (contraseña de acceso)
- ✓ Garantizar que las funcionalidades del sistema se muestren de acuerdo al nivel de usuario que este activo.
- ✓ Protección contra acciones no autorizadas o que puedan afectar la integridad de los datos.
- ✓ Verificación sobre acciones irreversibles (eliminaciones).

➤ Legales:

- ✓ El sistema se basa en el manual de normas y principios establecidos por el MINFAR.

➤ Confiabilidad:

- ✓ La herramienta de implementación a utilizar tiene soporte para recuperación ante fallos y errores.

➤ Funcionalidad:

- ✓ Mínima cantidad de páginas para ejecutar todas las funciones posibles (preferentemente que estén relacionadas).

➤ Implantación

- ✓ Entregar toda la documentación asociada al proyecto.
- ✓ Organizar el adiestramiento de los usuarios.

➤ Software:

En secciones anteriores se ha mencionado que la construcción de nuestra aplicación funcionará bajo los conceptos de arquitectura cliente-servidor. Por tanto el servidor del usuario final debe tener como requerimientos mínimos de software:

- ✓ Una computadora personal con plataforma del sistema operativo Windows Advancer Server 2000 o superior; o Linux.

- ✓ Apache 2.0 o superior como servidor Web, con módulo PHP 5 disponible y debe estar configurado con la extensión pgsql incluida.
- ✓ PostgreSQL como Sistema Gestor de Base de Datos.

Y la máquina cliente del usuario debe tener como requerimiento mínimo:

- ✓ El navegador Mozilla FireFox.

➤ Hardware

Partiendo del mismo supuesto que los requerimientos de software, para los requerimientos mínimos de hardware, el usuario final debe tener un servidor con las siguientes características:

- ✓ Tarjeta de red.
- ✓ 128 MB de RAM o superior.
- ✓ 40 Gb. de disco duro o superior.
- ✓ Pentium II a 133 MHz de velocidad en su procesador o más.

Una computadora que sirva de cliente:

- ✓ Pentium a 200 MHz. de velocidad de procesamiento o superior.
- ✓ 32 MB de memoria RAM superior.
- ✓ Tarjeta de red.

❖ Descripción del Sistema Propuesto

Para llevar a su total cumplimiento, todos los objetivos propuestos al inicio de este trabajo, el sistema a desarrollar, con fines de lograr una mejor comprensión y facilitar el trabajo, se ha dividido en 3 paquetes (figura 7), un paquete que contiene todos los casos de usos relacionados con las Recuperaciones que se pueden obtener a partir de uno o más comprobantes, otro paquete que agrupa todos los casos de usos relacionados con los Traspasos y un tercero que reúne 1 caso de uso encargado de realizar el Inicio de un año contable. Se considera la existencia de un rol: el contador principal.

El subsistema “Realizar Inicio” puede ser utilizado por el contador principal y esta compuesto por un caso de uso que es el encargado de abrir cuentas nuevas para el inicio de un nuevo año contable y de hacer un traspaso, al libro que se habilita para registrar las operaciones del año que se inicia, de toda la información con la que cerraron algunas cuentas en el año que recién finalizó.

El subsistema “Realizar Traspasos” ha sido incorporado una vez creado y desarrollado el sistema, es decir, es una mejora que la automatización de un conjunto de procesos traería para la actividad económica de las unidades, el mismo puede ser utilizado por el contador principal, y pretende realizar una búsqueda de la información contable registrada en un período de tiempo determinado y, luego de aplicarle una lógica específica, se registra o “traspasa” esa información procesada a un modelo específico con el fin de facilitar su uso por parte de algunos módulos que como el de Estados Financieros, se remiten a nuestro módulo de Contabilidad para solicitar la información contenida en este tipo de modelo, y que, al existir el mismo, solamente necesita realizar la consulta deseada sin necesidad de volver a aplicarle la lógica específica que lleva.

El subsistema “Realizar Recuperaciones” sólo puede ser usado por usuarios cuyo rol sea el de contador principal, y comprende la confección de varias recuperaciones (mayor, submayores, balance de comprobación, etc.) a partir de la información contenida en uno o más comprobantes.

Vale destacar que en el sistema, aunque no se incluya el paquete “Comprobantes”, se utiliza, está previamente creado y aunque no entra en el ámbito del proyecto que se modela en este trabajo de diploma, presenta una relación muy fuerte con el sistema a desarrollar por las características que posee.

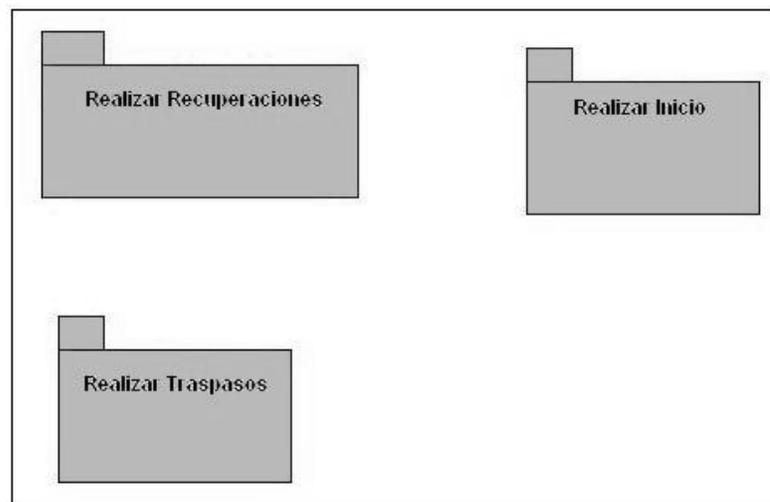


Fig. 7 Distribución del sistema por paquetes.

❖ Modelo de Casos de Uso del Sistema

Una vez identificados los requisitos funcionales y no funcionales del sistema, se prosigue a la creación del Modelo del Sistema que está compuesto por varios artefactos dentro de los que podemos identificar al Diagrama de Casos de Uso del Sistema.

El Diagrama de Casos de Uso del Sistema representa gráficamente a los procesos y su interacción con los actores. Cada caso de uso debe comunicarse con al menos un actor, de no ser así, indicaría la existencia de algún tipo de error en el modelo de casos de uso o en los requerimientos planteados. Dada la complejidad de este sistema, se ha decidido realizar un diagrama de Casos de uso del Sistema por paquetes.

Generalmente, y apoyándose en los diagramas de actividades se creará un caso de uso del sistema por cada actividad del diagrama de actividades que hayan sido identificadas como críticas o imprescindibles a la hora de la automatización.

Por tanto, el rol que lleva a cabo la actividad será el actor principal del caso de uso. Nótese que, de acuerdo con la definición de caso de uso, no todas las actividades del diagrama serán consideradas como casos de uso, sino solamente aquellas que sean de valor para algún actor.

Vale destacar que algunos casos de uso no se obtendrán directamente a partir del diagrama de actividades sino que se detectarán al describir los casos de uso identificados y al adquirir un mayor conocimiento sobre los requisitos que deben ser soportados.

Estos nuevos casos de uso representarán funciones que debe llevar a cabo el sistema para lograr algún objetivo asociado con algún caso de uso ya existente.

A continuación aparece una tabla con el listado del actor del sistema y la justificación del mismo.

<u>Actores</u>	<u>Justificacion</u>
Contador Principal	Es el encargado de interactuar con el sistema para realizar cada una de las operaciones contables que describen los diferentes casos de uso del sistema.

Recuperaciones: Este subsistema agrupa todos los C.U que tengan funcionalidades parecidas, de forma que se facilite mucho la implementación de los mismos, dado que el flujo de trabajo que recogen estos C.U son muy similares, así como las tablas que utilizan de la Base de Datos.

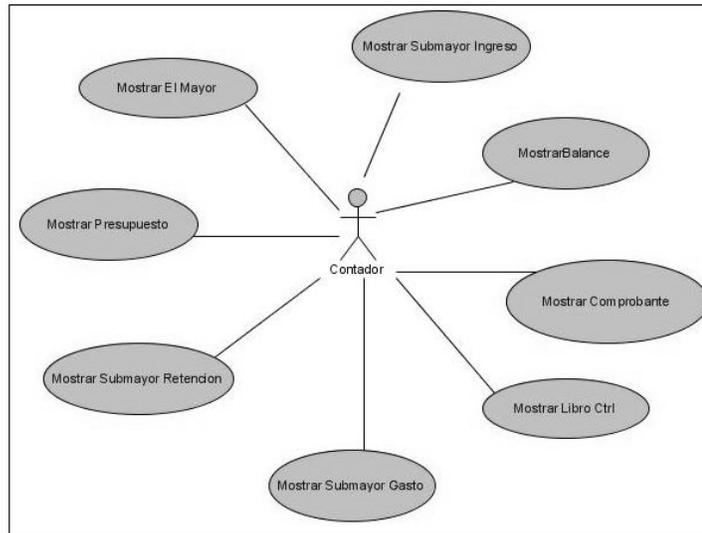


Fig. 8 Subsistema de Recuperaciones

Trasposos: Este subsistema se encarga de agrupar los C.U que se encargan de realizar cualquier tipo de traspaso, con el objetivo fundamental de economizar la programación y reutilizar los mismos recursos que emplean los Casos de Uso. A continuación aparece la descripción Global de cada uno de los Casos de Uso correspondientes a este subsistema.

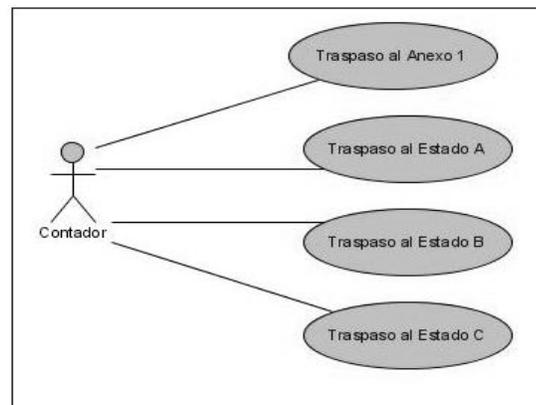


Fig. 9 Subsistema de Trasposos

Iniciar Año Contable: Este subsistema se encarga de dividir esta tarea de abrir un año contable en un Caso de Uso que desarrolla flujos de actividades que cumplen un mismo objetivo.

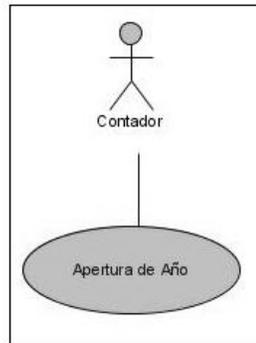


Fig. 10 Subsistema Iniciar Año Contable.

❖ Descripción de los Casos de Uso del Sistema

En este punto se describen de forma global el flujo de acciones llevadas a cabo por cada C.U., convirtiéndose este paso en el último a desarrollar antes de pasar a la construcción del sistema. La descripción detallada y expandida de cada Caso de Uso, con los modelos generados por el sistema, correspondientes a cada uno, se encuentran en los Anexos.

■ Paquete Recuperaciones

Este paquete se encuentra integrado por 8 Casos de Uso, que son los siguientes:

- 1) Mostrar Comprobante (SCF-56)
- 2) Mostrar el Mayor (SCF-50)
- 3) Mostrar Libro Registro y Control
- 4) Mostrar Submayor de Gasto
- 5) Mostrar Submayor de Ingreso
- 6) Mostrar Submayor de Retención
- 7) Mostrar Balance
- 8) Mostrar Control del Presupuesto

CU-1	<u>Mostrar Comprobante (SCF-56)</u>
Actor	Contador
CU asociado	Buscar Comprobante (incluye)
Descripción	Dentro de los objetivos fundamentales de este caso de uso están: -Servir de soporte al registro de las operaciones económicas que se realizan en las cuentas controles del Mayor y subcuentas afectadas. -Resumir las operaciones económicas anotadas en los diferentes registros, soportar los ajustes contables que se autorizan, recopilar mensualmente, dichas operaciones realizadas en los órganos financieros, con vistas a efectuar un solo pase a las cuentas del Mayor.
Referencia	R16

CU-2	<u>Mostrar el Mayor (SCF-50)</u>
Actor	Contador
CU asociado	
Descripción	Tiene como objetivo fundamental resumir las operaciones económicas que afectan las cuentas de control en los diferentes órganos financieros. Para obtener los saldos de las cuentas afectadas, partimos de que el sistema de contabilidad empleado se basa en el método de la partida doble, por lo que cada operación se registra por igual al Debe y al Haber de las cuentas de control o de orden que intervengan en dicha operación. Partiendo de la obtención de los saldos, se confecciona el Balance de Comprobación de Saldos (SICF- Estado A) y se realizan los cuadros de los submayores que han sido habilitados por la diferentes cuentas de control.
Referencia	R17

CU-3	<u>Mostrar Libro Registro y Control</u>
Actor	Contador
CU asociado	
Descripción	El CU se inicia cuando el usuario escoge la opción del Menú Principal: Mostrar Balance. Seguido a esto aparece un cuadro de diálogo donde deberá escoger el rango de fecha y el tipo de cuenta si es por Salario, Otros Gastos o Inversiones. El CU finaliza cuando el Sistema muestra el Balance. El objetivo fundamental de este caso de uso es que los órganos de economía administradores de créditos, a los diferentes niveles, puedan documentar los controles necesarios para administrar, distribuir y controlar el presupuesto y el financiamiento que por los diferentes conceptos desagrega, así como los recursos financieros (ingresos) que recibe de los órganos de economía o finanzas, consumidores de créditos que atiende financieramente.
Referencia	R18

CU-4	<u>Mostrar Submayor de Gasto</u>
Actor	Contador
CU asociado	
Descripción	Permite registrar y controlar los presupuestos aprobados, los gastos ejecutados por grupos presupuestarios, especialidades y partidas, así como los saldos del presupuesto. Registra y controla las cantidades de hombres que se corresponden con los importes contabilizados por cada una de las partidas de gastos de haberes y salarios. Sirve de base para la información a rendir sobre la base del presupuesto.
Referencia	R19

CU-5	<u>Mostrar Submayor de Ingreso</u>
Actor	Contador
CU asociado	
Descripción	Registra y controla, según su origen, la totalidad de los ingresos que cobran las unidades militares consumidoras de créditos. Sirve de base para transferir al órgano de economía del nivel superior los ingresos cobrados y controlar los que se encuentran pendientes de transferir. Sirve de base, además, para la información a rendir sobre la ejecución de los ingresos.
Referencia	R20

CU-6	<u>Mostrar Submayor de Retención</u>
Actor	Contador
CU asociado	
Descripción	Registra las pérdidas, los daños y faltantes de medios materiales y monetarios sujetos a investigación y las decisiones tomadas. Permite conocer, por cada especialidad, las causas que originaron las pérdidas, los daños y faltantes de medios materiales y monetarios. Sirve de Base para la información a rendir al órgano de economía del nivel superior sobre los daños materiales y perjuicios económicos causados a la Propiedad estatal.
Referencia	R21

CU-7	<u>Mostrar Balance (Estado A)</u>
Actor	Contador
CU asociado	
Descripción	Permite verificar si es igual el total de los saldos deudores y el total de los saldos acreedores de las cuentas contables. Permite comprobar además, la corrección de los saldos reflejados en las cuentas contables en relación con la naturaleza de estas.
Referencia	R22

CU-8	<u>Mostrar Control del Presupuesto</u>
Actor	Contador
CU asociado	
Descripción	Saber todo el control del Presupuesto que tiene ese órgano financiero, para eso se subdividió en diferentes opciones: - Presupuesto Asignado - Presupuesto Distribuido - Presupuesto Recibido - Libro que controla por cada registro el presupuesto por grupo presupuestario, especialidad y partida del gasto, ya sea el asignado, el distribuido y la reserva
Referencia	R23

■ Paquete de Traspasos

Este paquete se encuentra integrado por 4 Casos de Uso, que son los siguientes:

- 1) Traspaso (Anexo 1)
- 2) Traspaso (Estado A)
- 3) Traspaso (Estado B)
- 4) Traspaso (Estado C)

CU-9	<u>Realizar Traspaso (Anexo 1)</u>
Actor	Contador
CU asociado	
Descripción	Pasar el código de las Unidades Militares, su grupo presupuestario, su especialidad, la partida del gasto si lo lleva, el tipo del gasto más la asignación del presupuesto, y su ejecución en caso de ser un consumidor.
Referencia	R10

CU-10	<u>Realizar Traspaso (Estado A)</u>
Actor	Contador
CU asociado	
Descripción	El CU se inicia cuando el usuario escoge la opción del Menú Principal: Realizar Traspaso. Seguido a esto aparece un cuadro de diálogo donde deberá escoger el rango de fecha. El CU finaliza cuando el Sistema realice el traspaso. Este CU consiste en pasar los saldos de todas las cuentas y subcuentas contables desde una fecha hasta otra. Este pase sirve de base para el módulo de Estados Financieros ya que en este, se irán guardando de forma estadísticas los informes de Estado A de todos los órganos financieros incluyendo la del propio órgano.
Referencia	R11

CU-11	Realizar Traspaso (Estado B)
Actor	Contador
CU asociado	El CU se inicia cuando el usuario escoge la opción del Menú Principal: Traspaso/Estado. Seguido a esto aparece un cuadro de diálogo donde deberá escoger el rango de fecha. El CU finaliza cuando el Sistema realice el traspaso. Este caso de uso se encarga de pasar el código de las Unidades Militares, su especialidad, la partida del gasto si lo lleva, el tipo del gasto más la asignación del presupuesto, de la contabilidad propia del órgano.
Descripción	
Referencia	R12

CU-12	Realizar Traspaso (Estado C)
Actor	Contador
CU asociado	
Descripción	El CU se inicia cuando el usuario escoge la opción del Menú Principal: Traspaso/Estado. Seguido a esto aparece un cuadro de diálogo donde deberá escoger el rango de fecha. El CU finaliza cuando el Sistema realice el traspaso. Este Caso de Uso se encarga de pasar el código de las Unidades Militares más el grupo presupuestario de todas aquellas a las que se le distribuyeron presupuesto para que después el contador solamente tenga que poner el gasto ejecutado, para cada una de estas
Referencia	R13

■ Paquete Inicio de Año Contable

Este paquete esta compuesto por 1 caso de uso:

1) Apertrura del Año Contable

CU-12	Apertura del Año Contable
Actor	Contador
CU asociado	
Descripción	El CU se inicia cuando el usuario escoge la opción del Menú Principal: Realizar Apertura. Seguido a esto aparece una interfaz con un árbol que presenta las cuentas que deberá escoger cuyos saldos pasaran al año nuevo. El sistema se encargara de, según la clasificación de la cuenta, escoger contra que cuenta se abrirá y las acciones que deberá llevar sobre su saldo. Este Caso de Uso se encarga de pasar la información obtenida en el cierre para el comprobante que se habilita para recoger las operaciones contables del año que se inicia. El caso de uso termina cuando se muestra el comprobante para la apertura.
Referencia	R1

❖ Conclusiones

En este capítulo se comenzó a desarrollar la propuesta de solución, partiendo del análisis de los procesos del negocio, el listado de los requerimientos funcionales que debe tener el sistema, que se representaron, a su vez, mediante un Diagrama de Casos de Uso, y finalmente se describieron todas las acciones de los actores del sistema dentro de los procesos con los que se vinculan. Luego de haber desarrollado este conjunto de actividades, finalmente el equipo de trabajo se encuentra listo para empezar a construir el sistema, cuidando de garantizar el cumplimiento de las condiciones que se puntualizaron en este capítulo y que no pueden faltarle al producto finalmente acabado.

Capítulo



“Análisis y Diseño del Sistema”

Introducción

Análisis y Diseño

Análisis (Diagrama de Clases)

Diseño (Diagrama de Clases)

Mecanismos de Diseño

Diagramas de Interacción (Secuencia)

Diseño de la Base de Datos

Descripción de las tablas de la Base de Datos

Conclusiones

CAPITULO 3: “ANALISIS Y DISEÑO DEL SISTEMA”

❖ Introducción

Como resultado del flujo de trabajo de requisitos se obtiene una vista externa del sistema, que en el lenguaje del cliente, describe lo que se espera de él a través del Diagrama de Casos de Uso. A partir de aquí se profundiza en los casos de usos detallándolos de manera que permitan reflejar una vista interna del sistema descrita con el lenguaje de los desarrolladores. En esta vista interna se especifican mejor los casos de uso y se determinan las clases necesarias para llevar a cabo las funcionalidades en ellos contenidos.

Este proceso se desarrolla fundamentalmente dentro de la fase de Elaboración y se corresponde principalmente con el flujo de trabajo de Análisis y Diseño.

En este capítulo se comenzará con esta fase importantísima del proceso de desarrollo, pues en esta etapa, el sistema será estudiado para adquirir, finalmente, un conjunto de características en cuanto a su diseño, análisis e implementación que lograrán, en resumen, la calidad esperada. Serán estudiados además una serie de patrones que influirán en la construcción de la arquitectura y en los mecanismos de diseño de la misma.

❖ Análisis y Diseño.

En este flujo de trabajo se traducen los requisitos a una especificación que describe cómo implementar el sistema. El análisis específicamente, consiste en obtener una visión del sistema que se preocupa de ver QUÉ hace, de modo que sólo se interesa por los requisitos funcionales. Por otro lado, el diseño es un refinamiento del análisis que tiene en cuenta los requisitos no funcionales, en definitiva CÓMO cumple el sistema sus objetivos y debe ser suficiente para que el sistema pueda ser implementado sin ambigüedades.

❖ Análisis:

Las actividades contempladas en el análisis comienzan la profunda presentación del problema a resolver por lo que a través de ellas se representa una vista interna del sistema en la que, usando el lenguaje de los desarrolladores se refinan los requisitos y se estructuran en base a clases y paquetes. Este proceso continúa en el diseño hasta obtener los objetos que interactúan para cumplir los requisitos funcionales y no funcionales obtenidos. Se puede definir como propósitos del análisis:

- Conseguir una comprensión más precisa de los requisitos, refinarlos y estructurarlos.
- Utilizar el lenguaje de los desarrolladores para analizar con profundidad los requisitos funcionales.
- Proporcionar una visión general del sistema.

El Modelo de Análisis puede considerarse como una primera aproximación al Modelo de Diseño. Es una entrada fundamental cuando se da forma al sistema en el Diseño y en la Implementación.

A continuación aparecen los diagramas de Clases del Análisis por cada uno de los CU del sistema.

■ Paquete Recuperaciones.

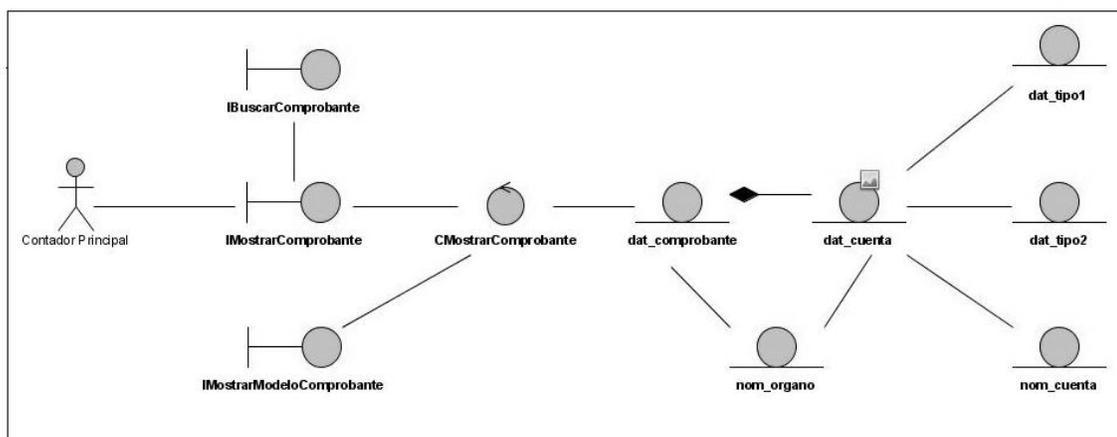


Fig. 11 Mostrar Comprobante.

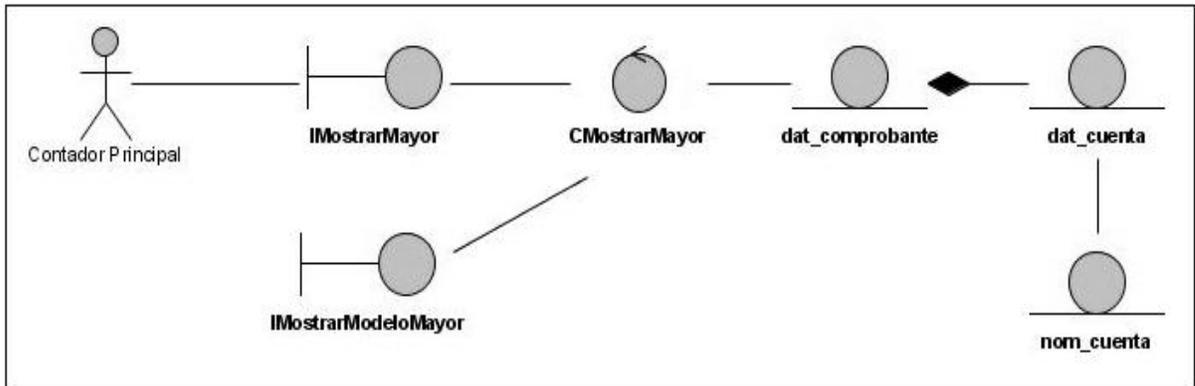


Fig. 12 Mostrar Mayor

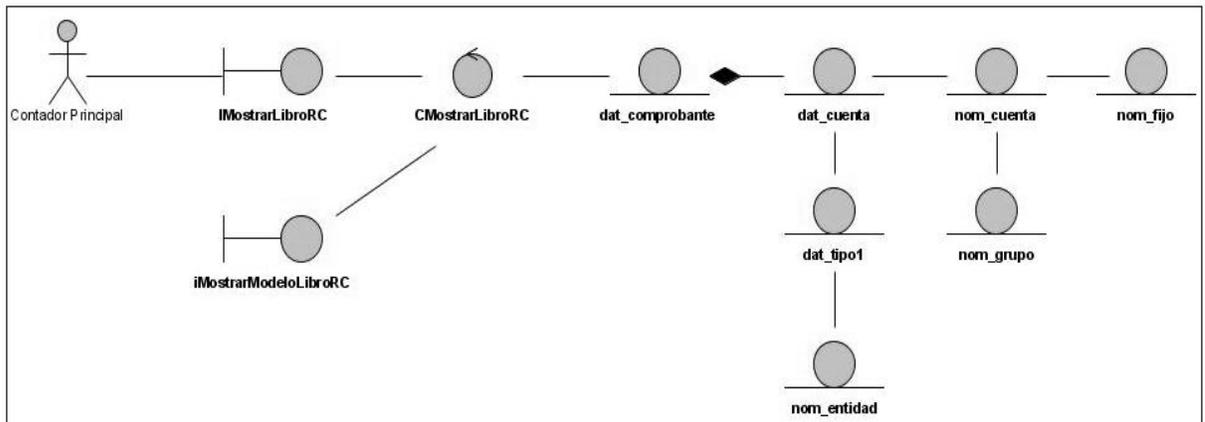


Fig. 13 Mostrar Libro de Registro y Control

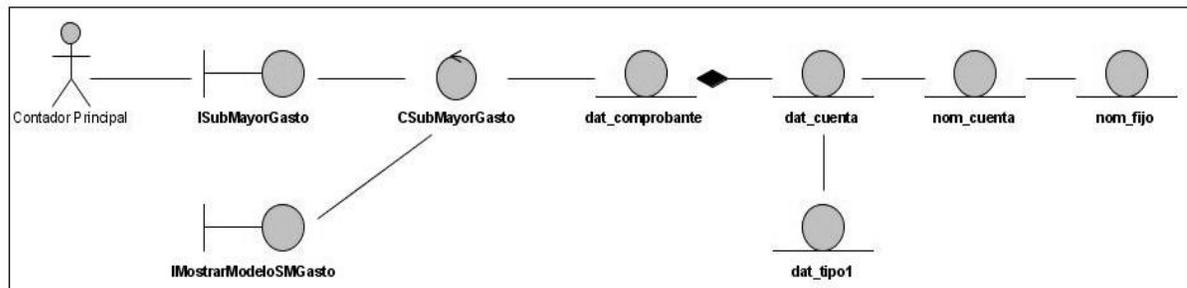


Fig. 14 Mostrar Submayor de Gasto.

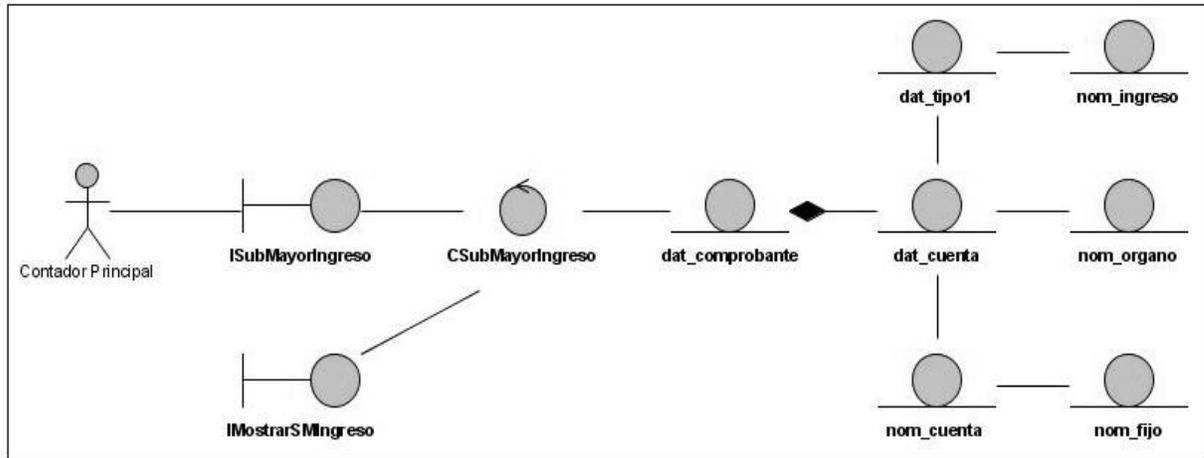


Fig. 15 Mostrar Submayor de Ingresos.

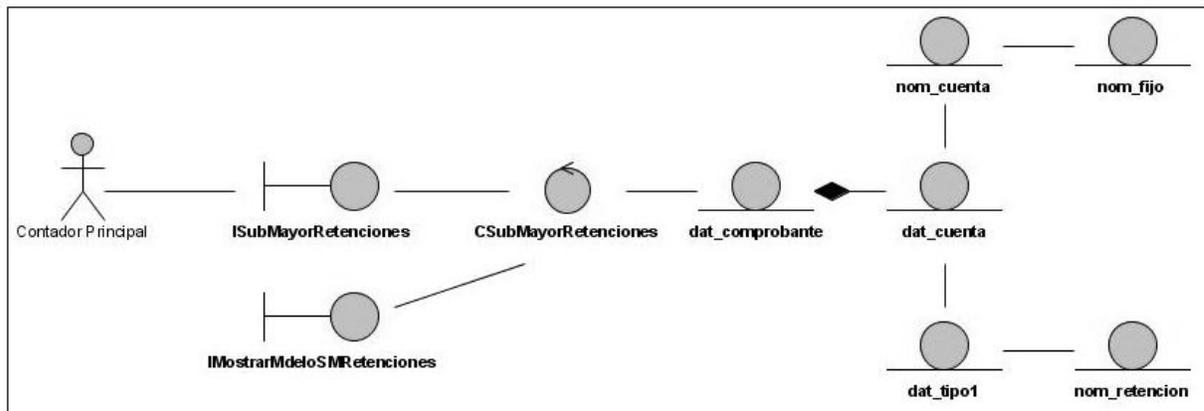


Fig. 16 Mostrar Submayor de Retenciones

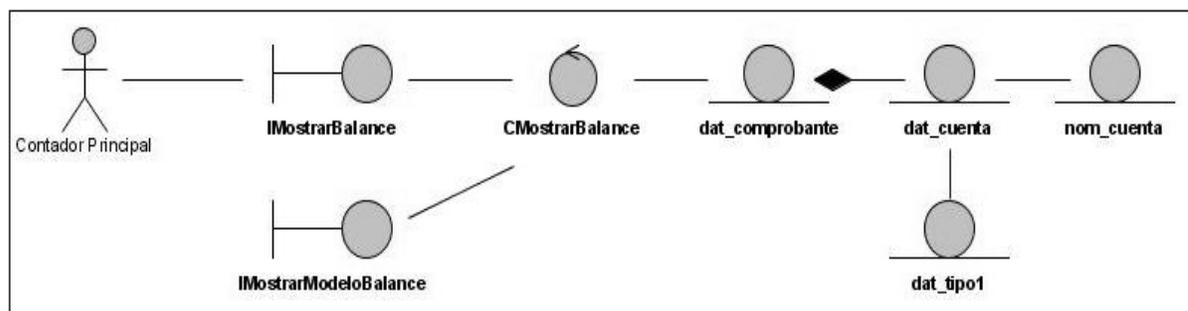


Fig. 17 Mostrar Balance.

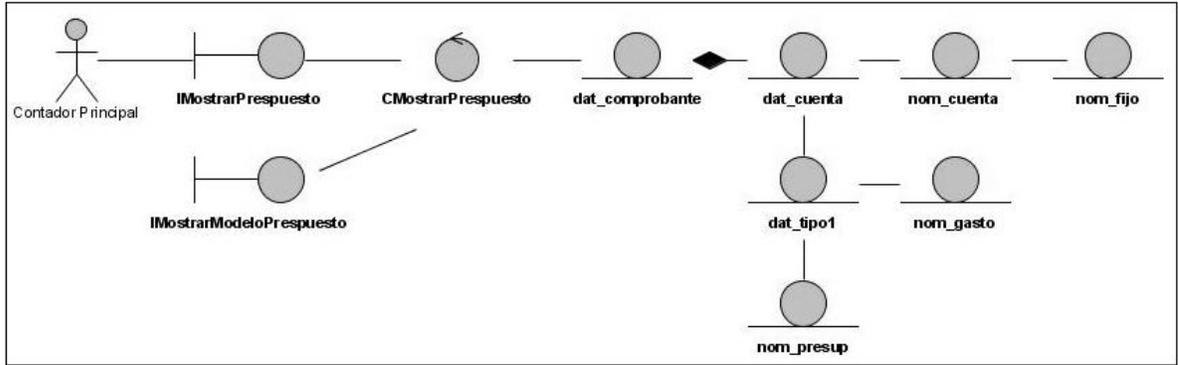


Fig. 18 Mostrar Presupuesto.

■ Paquete Traspasos.

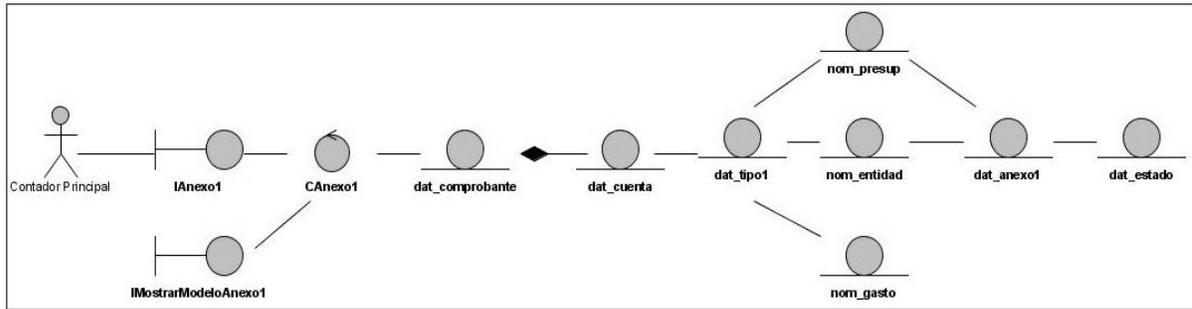


Fig. 19 Traspaso Anexo 1.

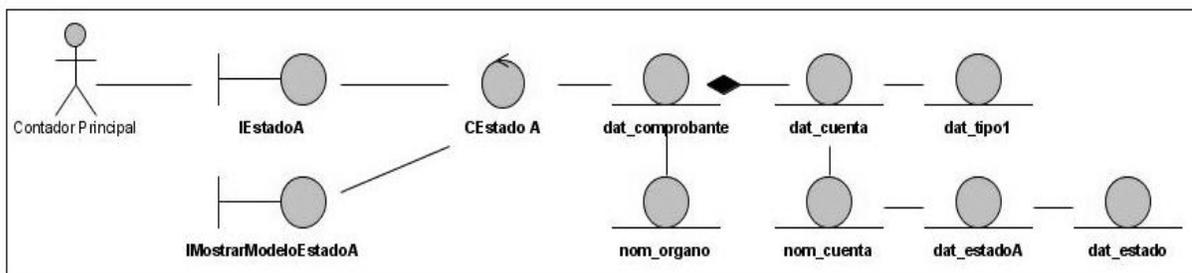


Fig. 20 Traspaso Estado A

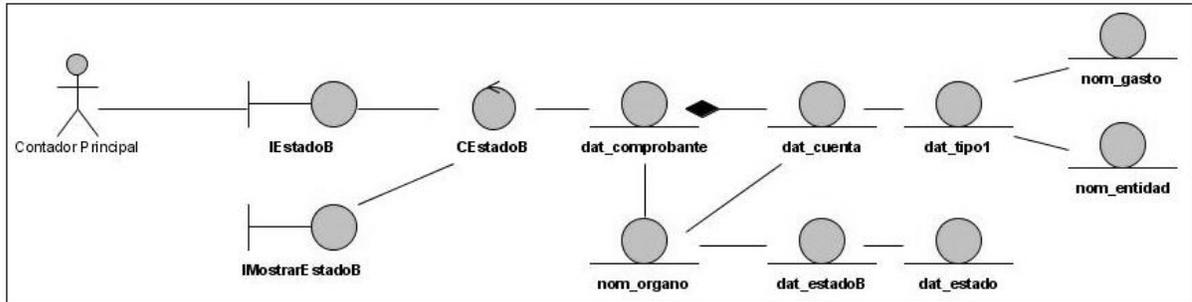


Fig. 21 Traspaso Estado B

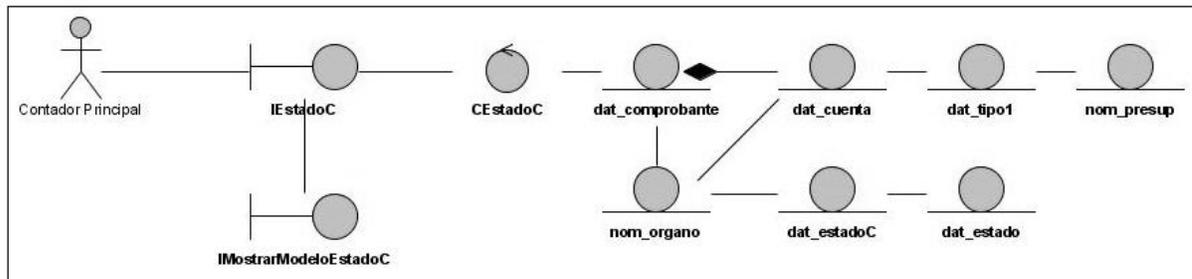


Fig. 22 Traspaso Estado C.

■ Paquete Inicio de Año Contable

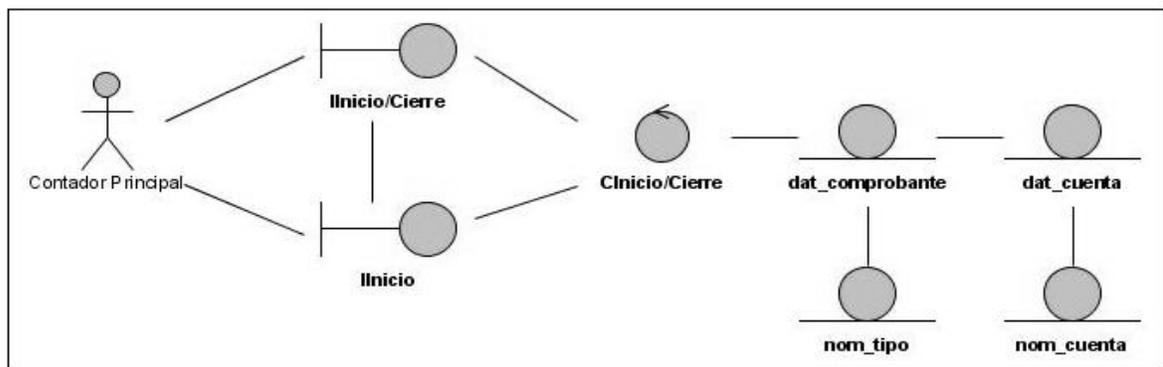


Fig. 23 Inicio del Año Contable

❖ Diseño.

El diseño es el centro de atención al final de la fase de Elaboración y el comienzo de las iteraciones de Construcción. Esto contribuye a una arquitectura estable y sólida, y crear un plano del Modelo de Implementación.

Este Modelo de Diseño está muy cercano al de Implementación, lo que es natural para guardar y mantener el Modelo de Diseño a través del ciclo de vida completo del software.

En el Diseño se modela el sistema y el mismo encuentra su forma (incluida la arquitectura) para que soporte todos los requisitos, incluyendo los no funcionales y las restricciones que le suponen.

Concretamente podemos definir como propósitos del diseño:

- Adquirir una comprensión de los aspectos relacionados con los requisitos no funcionales y restricciones relacionadas con los lenguajes de programación, componentes reutilizables, sistemas operativos, tecnologías de distribución y concurrencia y tecnologías de interfaz de usuario.
- Crear una entrada apropiada y un punto de partida para actividades de implementación, capturando los requisitos o subsistemas individuales, interfaces y clases.
- Descomponer los trabajos de implementación en partes más manejables que puedan ser llevadas a cabo por diferentes equipos de desarrollo.
- Capturar las interfaces entre los subsistemas en el ciclo de vida del software, lo cual es muy útil cuando utilizamos interfaces como elementos de sincronización entre diferentes equipos de desarrollo.

Este sistema cuenta, a su vez, con 3 subsistemas más, y teniendo en cuenta las actividades que ejecutan cada uno de ellos por separados, así como la homogeneidad de funcionamiento entre sus respectivos casos de uso, se ha decidido crear 3 lógicas de negocio correspondientes a cada uno de estos paquetes. Dichas lógicas de negocio recogen las funciones que son comunes a todos los casos de uso de ese paquete, y que son redefinidas después por cada caso de uso.

Esta redefinición se basa principalmente en los parámetros de entrada, pues el objetivo de estos métodos que se registran en las respectivas lógicas de negocio, son los mismos para cada caso de uso.

A continuación aparecen los diagramas genéricos de clases del diseño propios de cada subsistema y la explicación de la lógica de negocio correspondiente.

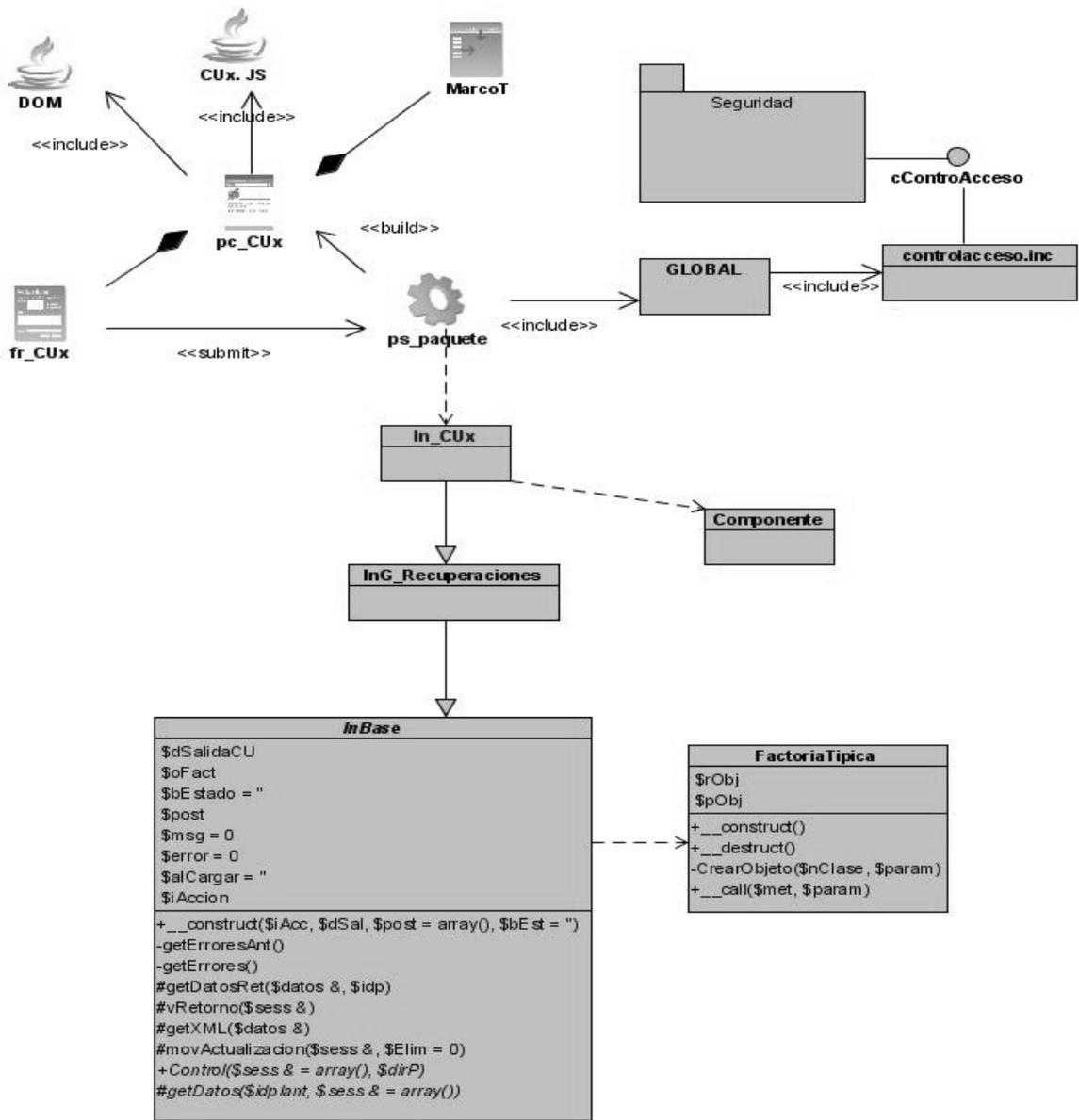


Fig. 24 Diagrama de Clases Genérico del Diseño. (Paquete Recuperaciones)

Nombre: Lógica de Negocio (Recuperaciones)	
Tipo de clase Lógica de Negocio	
Responsabilidades	
Nombre:	Buscar Cuentas()
Descripción:	Esta función se encarga de buscar el conjunto de cuentas, correspondientes a comprobantes, que cumplan con las características según los criterios de búsqueda seleccionados por el usuario. Los parámetros de entrada de esta función son variables en número y tipo, dado que cada recuperación tiene sus propias opciones de búsqueda de cuentas que le propone al usuario, este método se redefine en las Lógicas de Negocio correspondientes a cada Caso de Uso específico. Esta función devuelve un listado de cuentas.
Nombre:	Ordenar()
Descripción:	Esta función se encarga de ordenar un listado de cuentas, según un criterio determinado, en dependencia del Caso de Uso.
Nombre:	PasarDatos()
Descripción:	Esta función se encarga de recoger los datos, de las diferentes tablas de la Base de Datos, que servirán después para llenar los modelos respectivos de cada Caso de Uso. Los parámetros de esta función dependen también de cada Caso de Uso y del modelo que se genera en cada uno.
Nombre:	Acumular()
Descripción:	Esta función se encarga de sacar un acumulado de los importes de diferentes cuentas, determinando si el mismo se encuentra al debe o al haber. Este tipo de acumulado se realiza también mediante un criterio determinado que depende del Caso de Uso en el cual se este usando.
Nombre:	OperacionClasica()
Descripción:	Esta función se encarga de hacer la operación clásica de la contabilidad a diferentes cuentas. EL procedimiento es el mismo siempre, lo que varia es el tipo de cuenta al que se le realiza dicha operación.
Nombre:	MostrarModelo()
Descripción:	Esta función carga todos los datos recogidos en la función PasarDatos (), de ponerlos en los campos correspondientes en el modelo, y de mostrárselo al usuario. En esencia, la función es la misma para todos los casos de Uso, pero los atributos que carga son los que varían.

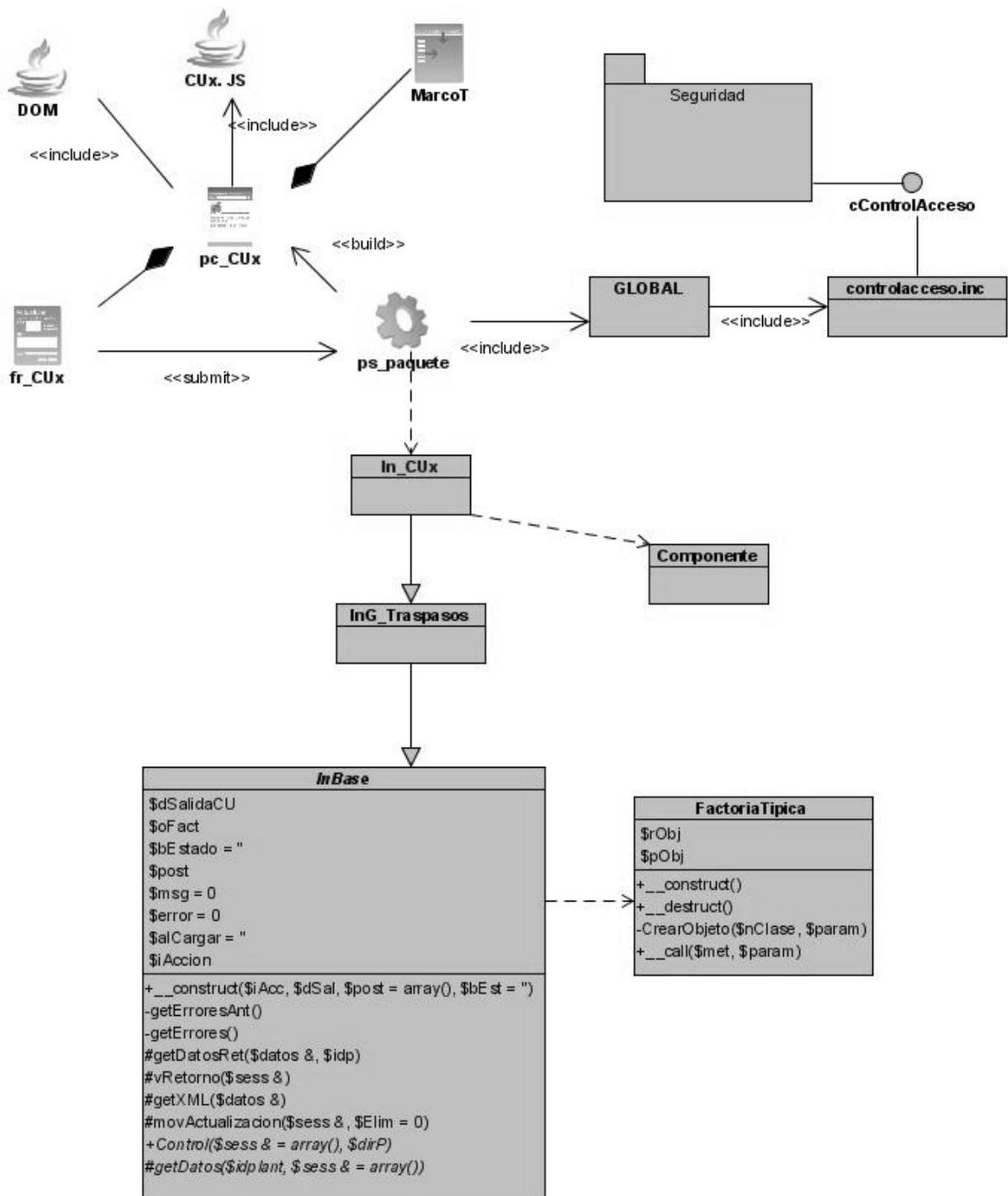


Fig. 25 Diagrama de Clases Genérico del Diseño. (Paquete Traspasos)

Nombre: Lógica de Negocio (Traspasos)	
Tipo de clase Lógica de Negocio	
Responsabilidades	
Nombre:	<u>BuscarComprobantes()</u>
Descripción:	Esta función se dedica a buscar los comprobantes que cumplen con las condiciones seleccionadas por el usuario, que en el caso de los Traspasos, es por el rango de fecha. La función es la misma en todos los Casos de Uso porque el criterio de búsqueda inicial para realizar un traspaso es a través de un filtro de fecha.
Nombre:	<u>PasarPresupuesto()</u>
Descripción:	Esta función se encarga de hacer el traspaso del presupuesto de algunas cuentas cumpliendo un criterio determinado, este criterio depende de cada Caso de Uso, incluso, esta función se puede utilizar en varios momentos en un mismo caso de uso, lo que varían son los criterios para hacer el traspaso del presupuesto y a que tipos de cuentas corresponde el mismo.
Nombre:	<u>TraspasoDatos()</u>
Descripción:	Esta función se dedica a realizar el traspaso de un conjunto de datos, es la función mas importante de esta lógica de negocio, pues cumple el objetivo fundamental de los traspasos, o sea, toma los datos que han sido previamente procesados y los mueve hacia el modelo destinado a recoger los datos de la operación, que seria el resultado final al que tendría acceso el usuario.
Nombre:	<u>CalcularSaldos()</u>
Descripción:	Esta función calcula los saldos de diferentes cuentas, estando sus importes ya sea al debe o al haber. Este calculo se hace siguiendo un conjunto de criterios que dependen totalmente del caso de uso en el que se este usando la función, incluso, se puede llamar en varias ocasiones al mismo procedimiento dentro de la lógica de negocio del Caso de Uso, pero puede que los criterios para emitir el saldo sean diferentes.
Nombre:	<u>FiltrarCuentas()</u>
Descripción:	Esta función se encarga de buscar un conjunto de cuentas que cumplen con determinadas características, definidas por el propio Caso de Uso.
Nombre:	<u>OrdenarCuentas()</u>
Descripción:	Este método ordena un listado de cuentas según un criterio determinado especificado por el propio caso de uso.
Nombre:	<u>MostrarTraspaso()</u>
Descripción:	Esta función carga los datos a los que previamente se les realizo el traspaso y los ubica en los campos que les corresponde en el modelo propio del traspaso para mostrárselo, luego, al usuario esta función es prácticamente la misma para todos los CU de uso, solamente varia en los datos que carga y en el modelo que muestra.

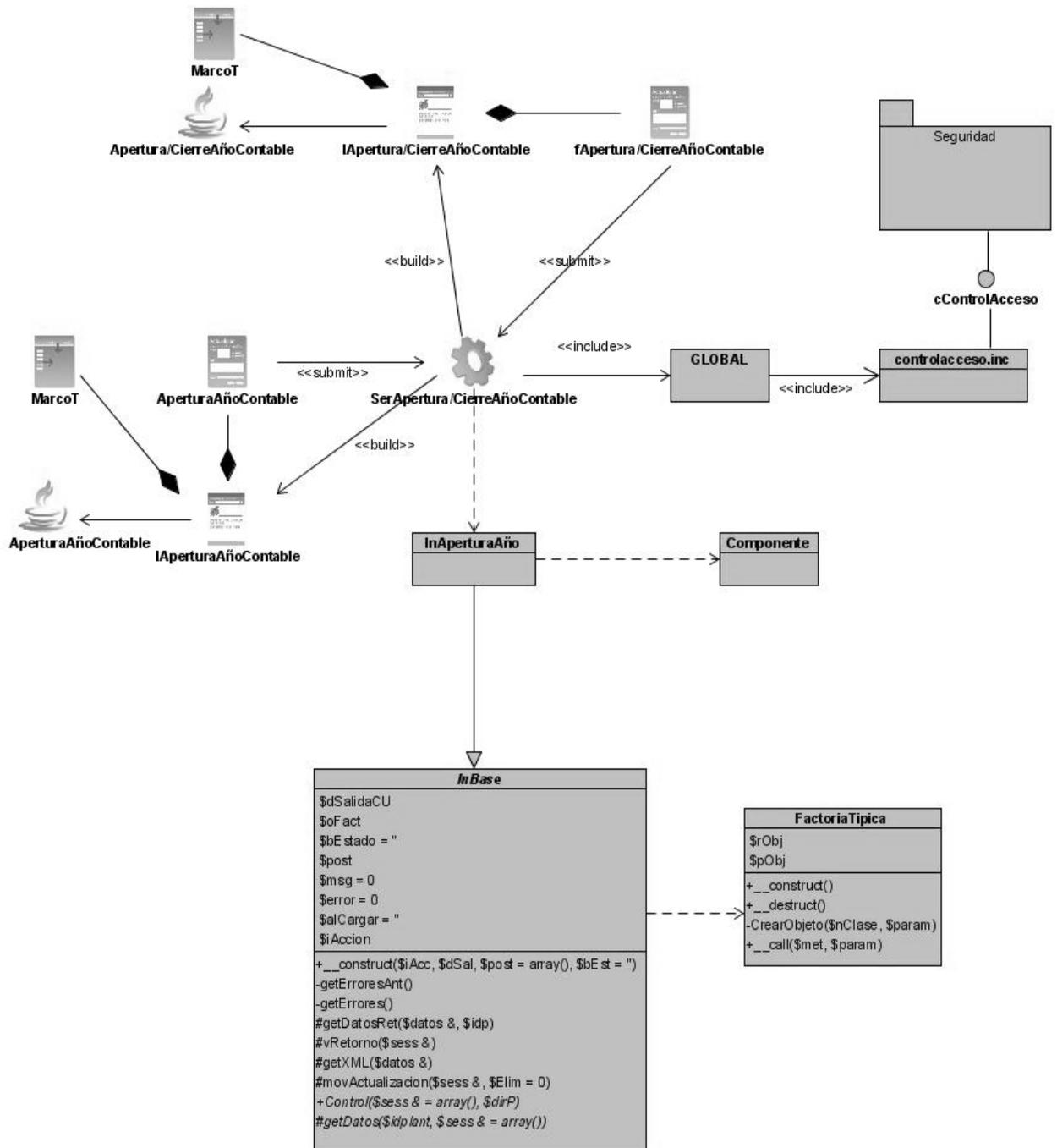


Fig. 26 Diagrama de Clases Genérico del Diseño. (Paquete Inicio del Año Contable)

Nombre: Lógica de Negocio (Inicio y Cierre de Año)	
Tipo de clase: Lógica de Negocio	
Responsabilidades	
Nombre:	BuscarComprobantes()
Descripción:	Esta función se dedica a buscar los comprobantes que cumplen con las condiciones seleccionadas por el usuario, que en el caso del Inicio/Cierre, es por el rango de fecha. La función es la misma en los dos Casos de Uso.
Nombre:	FijarSalDOS()
Descripción:	Este método fija el valor de los saldos de diferentes cuentas según criterios determinados por el caso de uso en el que se aplica y según el momento en el que se ha realizado la llamada del mismo dentro de la lógica de negocio propia del Caso de Uso.
Nombre:	TrasladarCifras()
Descripción:	Este método traslada un conjunto de cifras registradas en modelos específicos a otros modelos o libros que a partir de ese momento, recogerán estos datos con o sin previa modificación.
Nombre:	AcreditarSalDO()
Descripción:	Esta función se encarga de poner al haber en otras cuentas previamente definidas, el saldo que ha sido fijado algunas cuentas. Este procedimiento de acreditar depende de las características de la cuenta que se utiliza y de esta forma, la cuenta contra la que se acredita, también está especificada por las condiciones que impone el caso de uso.

A continuación aparecen los diagramas de clases del diseño de cada caso de uso. Estos diagramas presentan la lógica de negocio propia de cada caso de uso, que incluyen métodos que heredan de la lógica del negocio del subsistema al que pertenecen y los métodos propios del caso de uso en cuestión. Los métodos heredados se redefinen en cada caso de uso según las características que el mismo presente.

■ Paquete Recuperaciones:

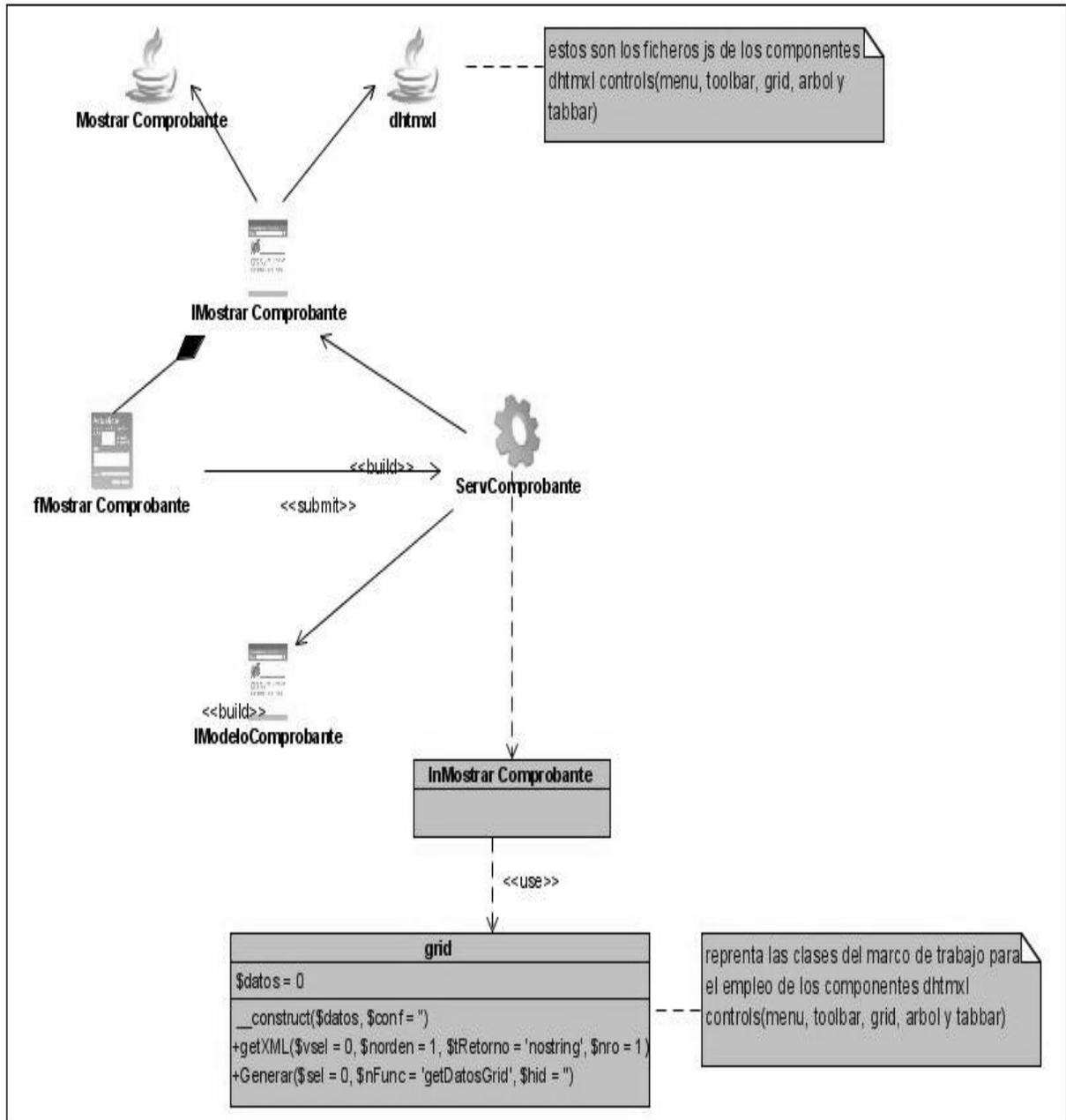


Fig. 27 Mostrar Comprobante

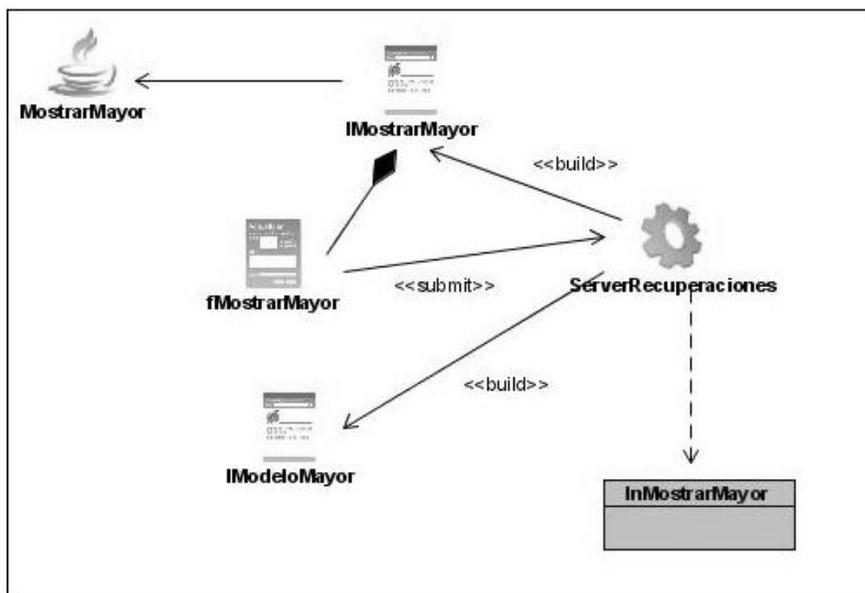


Fig. 28 Mostrar Mayor

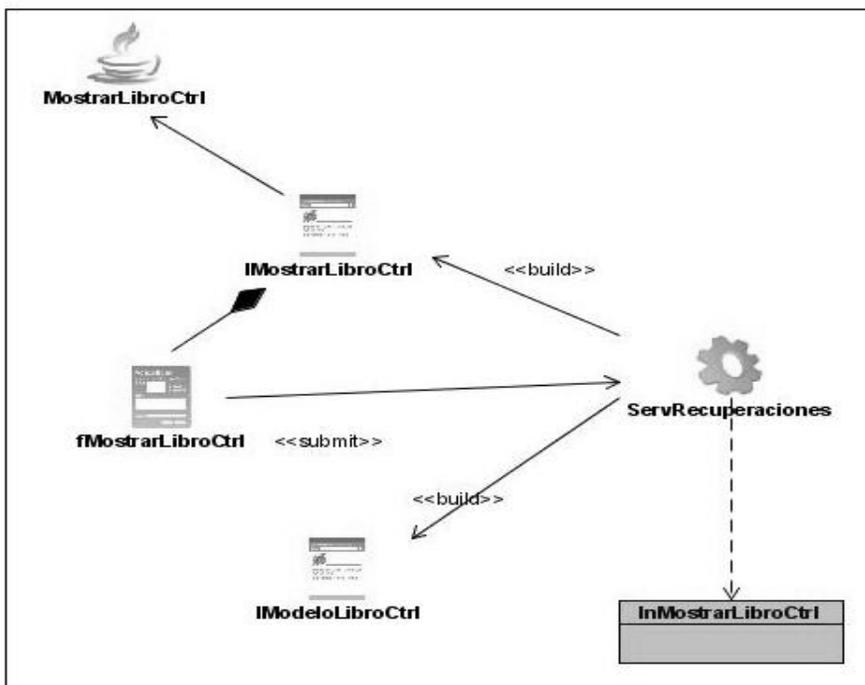


Fig. 29 Mostrar Libro de Registro y Control

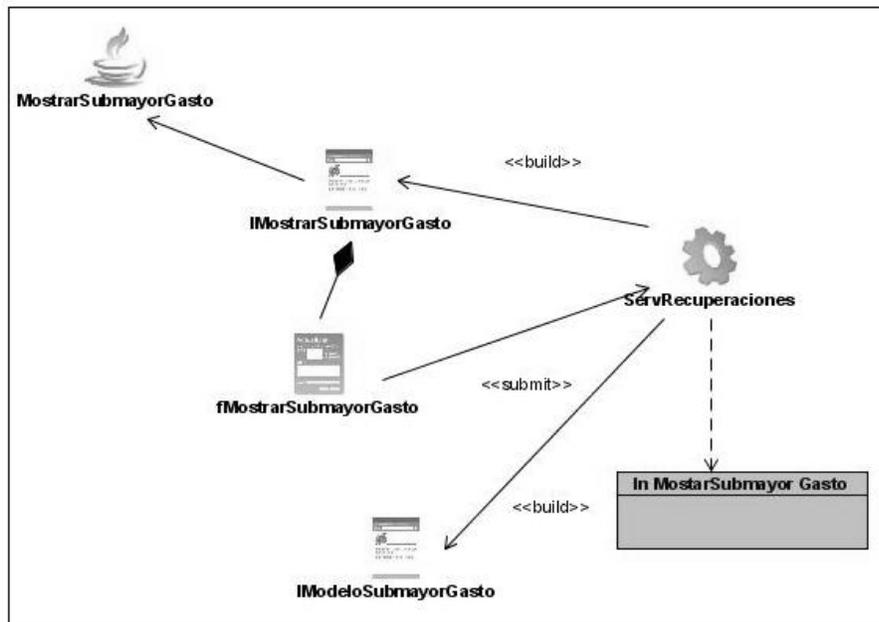


Fig. 30 Mostrar Submayor de Gasto

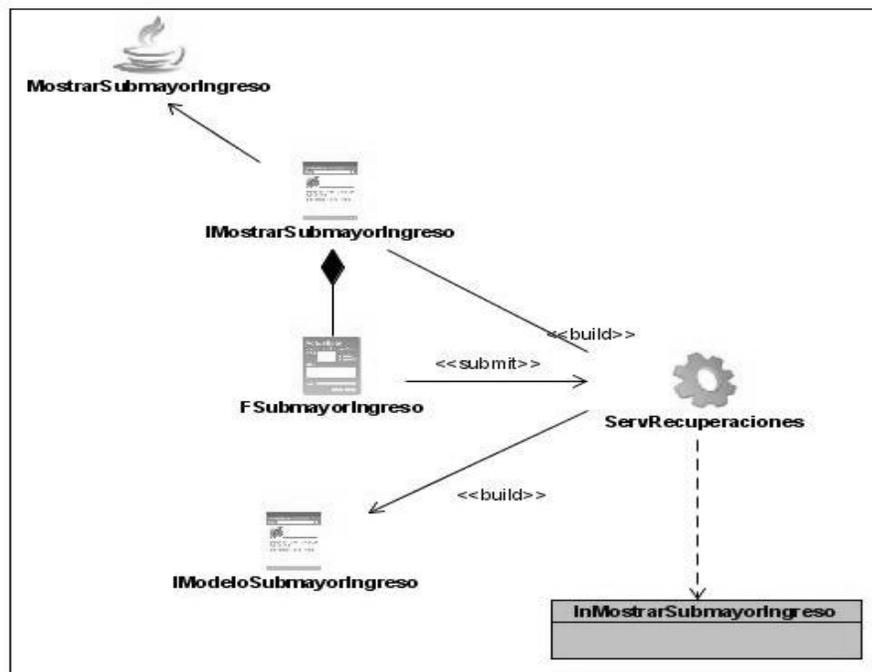


Fig. 31 Mostrar Submayor de Ingreso

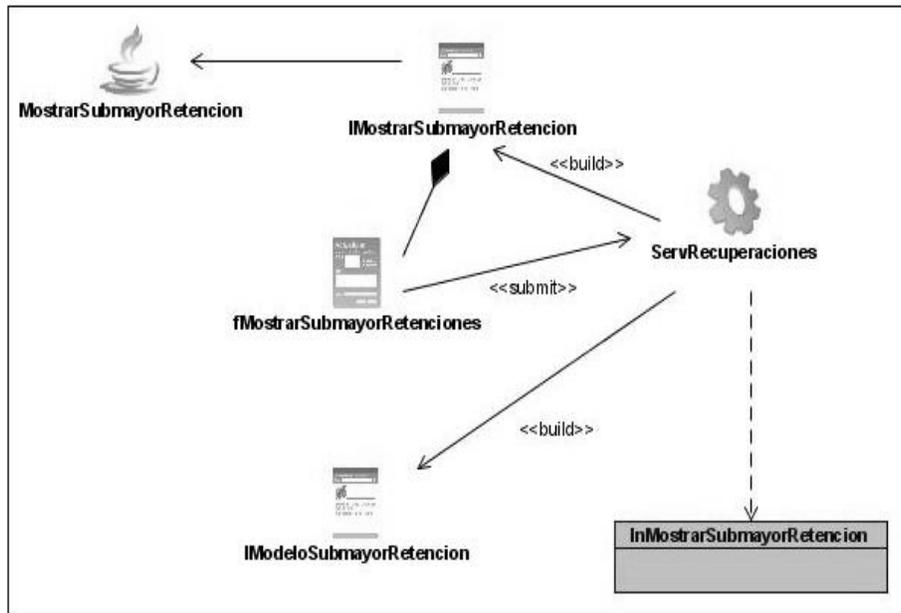


Fig. 32 Mostrar Submayor de Retención

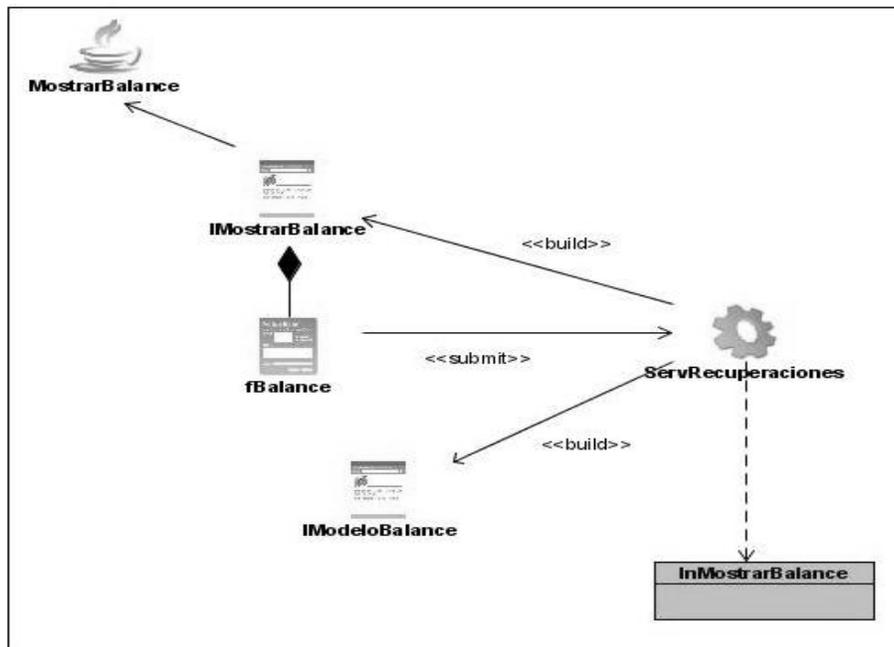


Fig. 33 Mostrar Balance

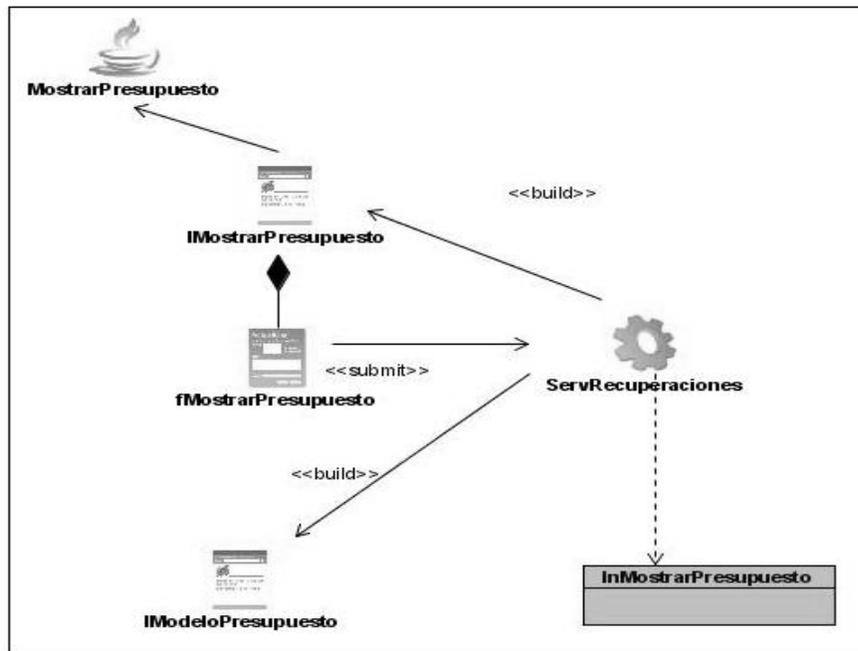


Fig. 34 Mostrar Presupuesto.

■ Paquete Traspasos

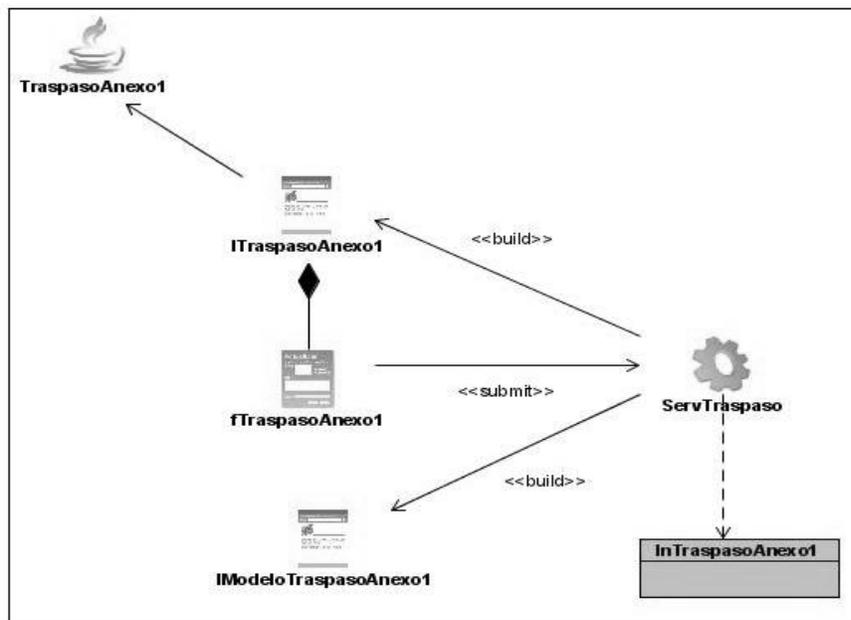


Fig. 35 Traspaso Anexo 1

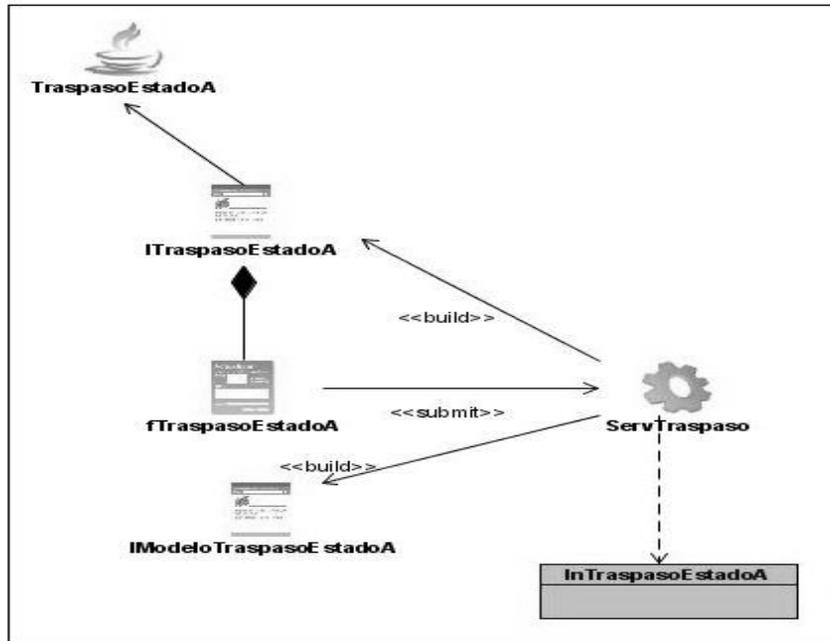


Fig. 36 Traspaso Estado A

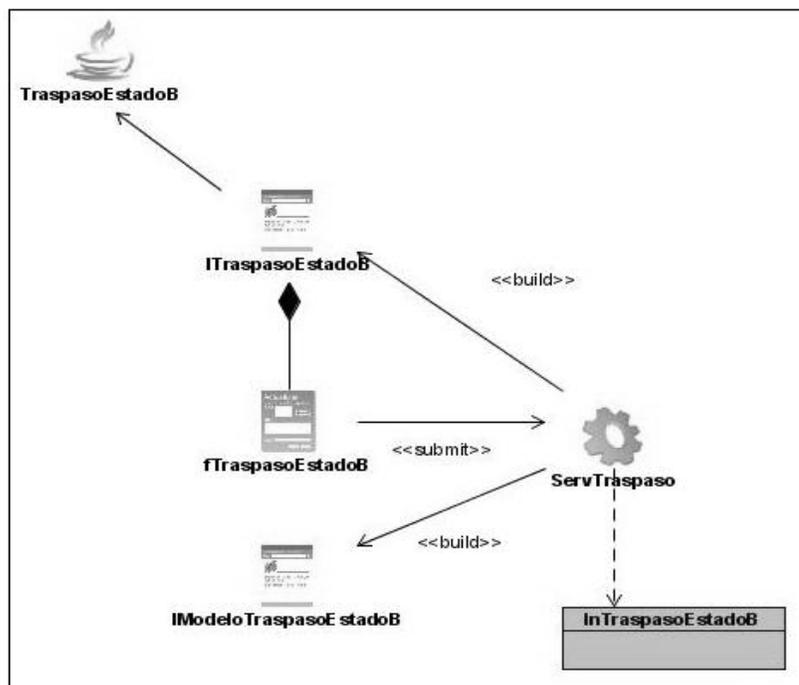


Fig. 37 Traspaso Estado B

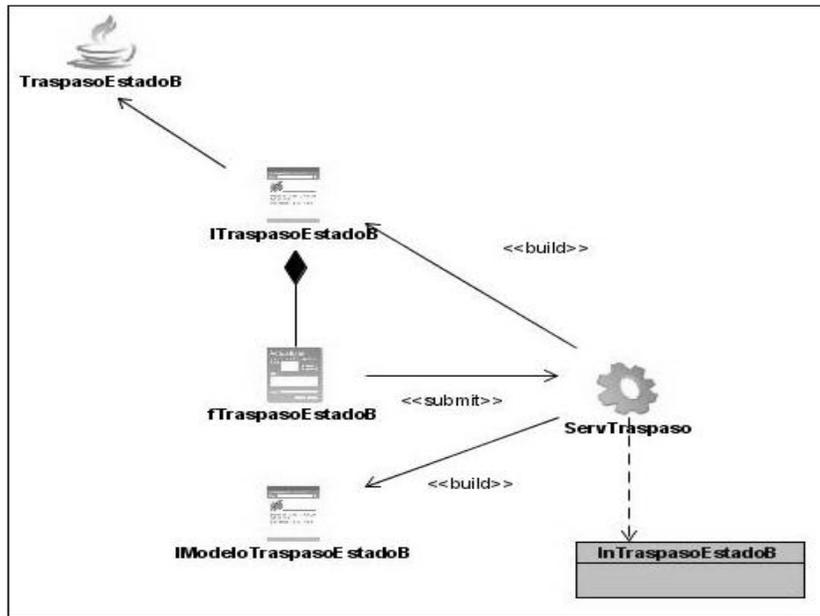


Fig. 38 Traspaso Estado C.

■ Paquete Inicio del Año Contable

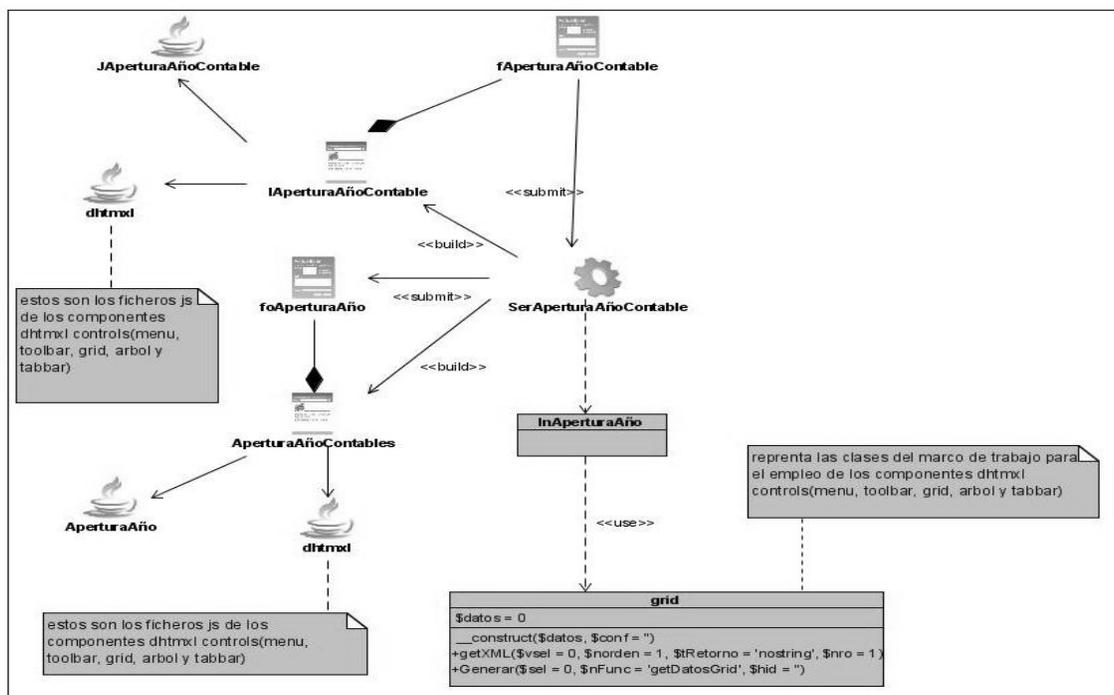


Fig. 39 Inicio de Año Contable

❖ Mecanismos de Diseño.

En este flujo de análisis y diseño se comienzan actividades desde la fase de Inicio, para que, finalmente, quede definida la solución del sistema. Los mecanismos de diseño reportan beneficios para mantener la homogeneidad en el diseño, reutilizar las soluciones anteriormente probadas, así como la documentación del proyecto.

❖ Mecanismo de Diseño de Seguridad.

Cuando utilizamos la Web para el modelamiento del sistema, es posible percatarse de que existen ciertas paginas que necesitan control para acceder a ellas, y para llevar a cabo estas acciones, lo mas recomendable es crear variables de sesión en las que se registran la identidad del usuario, una vez ejecutada la autenticación. Dichas variables se encargan de validar la existencia del usuario en el sistema y los permisos y privilegios que adjuntos a su perfil, el mismo ofrece, que a su vez, como modulo de un ERP, obtiene estas variables de sesión a partir de otro modulo de seguridad que, a través de un servicio WEB, facilita estas posibilidades.

El Sistema de Contabilidad Financiera es un modulo que se esta desarrollando dentro del ERP que para el MINFAR, se esta construyendo. Este ERP cuenta con un conjunto de subsistemas dentro de los que encontramos el destinado a la seguridad del proyecto. Este subsistema, que se comunica con una interfaz con la clase de control de Acceso, recibe las autenticaciones de los usuarios al ingresar al sistema, y chequea su existencia dentro del mismo, así como los privilegios que le han sido asignados, de ahí se derivan el conjunto de acciones que se pueden ejecutar dentro del modulo al que se ha accedido y al que lo han redireccionado tras haber confirmado la autenticidad de los datos entrados por el usuario.

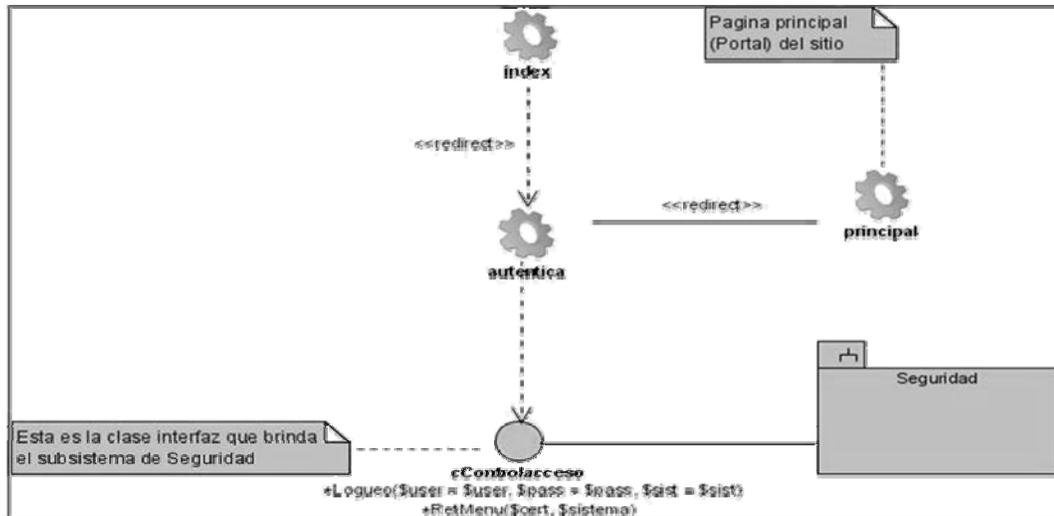


Fig. 40 Mecanismo de Diseño de Seguridad.

❖ Mecanismos de Diseño de Acceso a Datos

Este mecanismo de diseño se ha implementado para controlar el acceso a los datos, pues se han identificado objetos que se involucran siempre en este tipo de ejecución y se efectúan, además, un conjunto de operaciones comunes a la mayoría de los Casos de Uso.

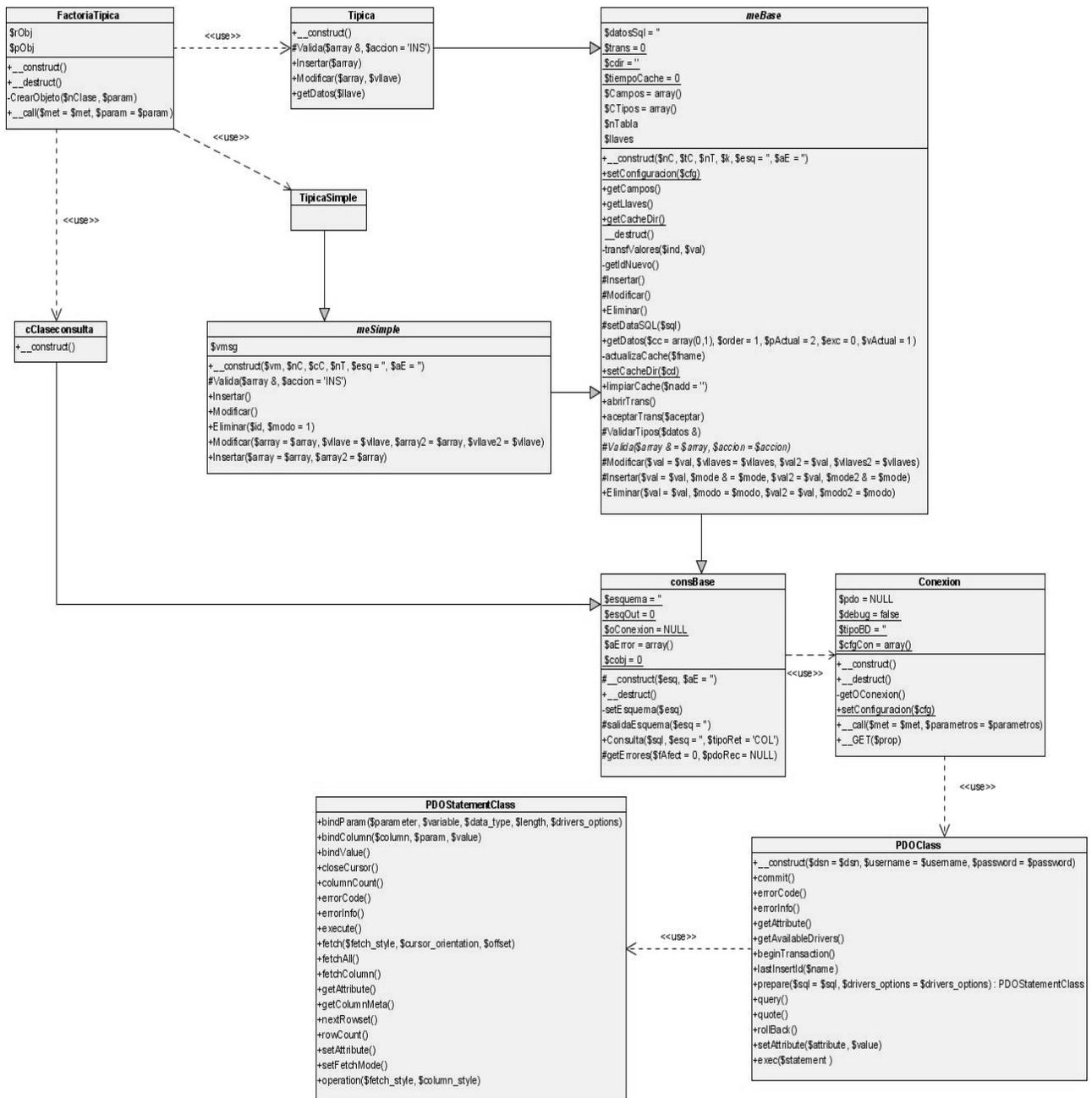


Fig. 41 Mecanismo de Acceso a Datos.

En el modelo de diseño de este mecanismo se observan un conjunto de clases que interactúan para facilitar el acceso y la manipulación de los datos persistentes a los diferentes niveles, por ejemplo:

➤ Clase Factoría Típica:

En la creación de la misma se han aplicado diferentes patrones de diseño, que se traducen en una solución efectiva y reusable a problemas no triviales. Es una clase que ha sido implementada siguiendo los conceptos del patrón Factoría, que siendo un patrón creacional, propone soluciones acertadas a los problemas de creación de instancias de objetos. Este patrón centraliza el sitio donde se crean los objetos de componentes visuales, componentes de la lógica del negocio, u objetos concurrentes en el tiempo.

La clase Factoría basa la creación centralizada de las instancias de un objeto en los datos que se le pasan como parámetros. Para que este tipo de creación sea mas útil y eficaz, se busca que los objetos creados desciendan de la misma clase o implementen las mismas operaciones pero de distintas formas, de esta manera se logra el uso uniforme de estos objetos, con sus mismas operaciones, sin tener en cuenta la clase concreta con la que estaremos tratando en cada momento.

Factoría Típica está concebida para instanciar todas las típicas de nuestro sistema para utilizarlas de algún modo.

Las clases Típicas son clases que surgen también de la aplicación de patrones de arquitectura específicamente TABLE DATA GATEWAY que se encarga de realizar una clase para instanciar cada tabla de la base de datos.

Esta clase posee la función function __call (\$met, \$param) que implementa de forma explícita estos métodos base:

- ✓ Crear -> Permite la creación de instancias de clases. Se emplea solo cuando se desea ejecutar una única operación de las que implementa la clase instanciada (siempre se destruirá automáticamente la instancia después de la ejecución del método) y además su constructor debe recibir parámetros.
Empleo: [Nombre de la clase] _Crear ([parámetros,...]) ó simplemente [Nombre de la clase] ([parámetros,...])

- ✓ pCrear-> Permite la creación de instancias de clases. Se emplea cuando se desea ejecutar varias operaciones de las que implementa la clase instanciada. La destrucción del objeto creado será responsabilidad del programador, aunque al concluir la ejecución del script esta se destruye.

Empleo: [Nombre de la clase] _pCrear ([parámetros,...])

- ✓ Destruir-> Permite destruir una instancia creada.

Empleo: [Nombre de la clase] _Destruir () Implementa de forma implícita el resto de las operaciones que están contenidas dentro de las clases del modelo de acceso a datos, las más empleadas son por ejemplo:

- Insertar -> Para insertar datos en la entidad dada.

Empleo: [Nombre de la clase] _Insertar ([arreglo de campos=>valores])

- Modificar -> Para modificar datos en la entidad dada.

Empleo: [Nombre de la clase] _Modificar ([arreglo de campos=>valores], llave)

- Eliminar -> Para eliminar datos en la entidad dada.

Empleo: [Nombre de la clase] _Eliminar ([arreglo de llaves])

- getDatos -> Para consultar datos en las entidades de nomenclaturas.

Empleo: [Nombre de la clase] _getDatos ([arreglo de llaves])

Nombre: Factoría Típica	
Tipo de clase Interfaz (del modelo de persistencia con el resto de los subsistemas)	
Atributos	
Nombre	\$rObj
Descripción	Para el registro de objetos
Nombre	\$pObj
Descripción	Permite el chequeo de objeto persistentes
Responsabilidades	
Nombre:	<u>function construct()</u>
Descripción:	Constructor de la clase.
Nombre:	<u>function destruct()</u>
Descripción:	Destructor de la clase
Nombre:	<u>function CrearObjeto(\$nClase, \$param)</u>
Descripción:	Permite la creación de instancias de clases. Se emplea solo cuando se desea ejecutar una única operación de las que implementa la clase instanciada (siempre se destruirá automáticamente la instancia después de la ejecución del método) y además su constructor debe recibir parámetros.

	<u>Empleo:</u> [Nombre de la clase] _Crear ([parámetros,...]) ó simplemente [Nombre de la clase] ([parámetros,...])
Nombre:	function call(\$met, \$param)
Descripción:	Nos permite de forma transparente la interacción con el resto de las clases del modelo de persistencias.

➤ Clase meBase:

Es una clase abstracta que constituye la base para el resto de las que implementen funcionalidades para el trabajo con las entidades del sistema desarrollado. Implementa además, las operaciones básicas que pudieran realizarse a una entidad como pudieran ser: INSERT, DELETE, UPDATE así como un conjunto de consultas. Esta clase encapsula todo lo relacionado con la conexión al gestor de las bases de datos.

Nombre: meBase	
Tipo de clase Clase abstracta.	
Atributos	
Nombre	\$esqOut
Descripción	Bandera para ejecución de la sentencia SET PAHT para indicar el esquema a emplear.
Nombre	\$datosSql
Descripción	Solución a las consultas a los nomencladores complejos.
Nombre	\$con
Descripción	Objeto de conexión a la Base de datos. Se declara estático y privado para garantizar un único objeto de conexión y encapsular su empleo solo a esta clase. <u>Tipo:</u> PDO (objeto nativo), Valor por defecto: NULL.
Nombre	\$trans
Descripción	Control de transacciones. Bandera para indicar transacción en curso. <u>Tipo:</u> boolean, Valor por defecto: false (0)
Nombre	\$cobj
Descripción	Control de la cantidad de objetos creados empleando como clase base a esta. <u>Tipo:</u> entero, Valor por defecto: false (0)
Nombre	\$esquema
Descripción	Control de los esquemas que pertenecen las entidades con las que se trabaja. <u>Tipo:</u> cadena, Valor por defecto: Vacía (")
Nombre	\$cdir
Descripción	Configuración del directorio para la cache de resultados de consultas. Generalmente se emplea para las consultas a nomencladores. <u>Tipo:</u> cadena, Valor por defecto: Vacía (")

Nombre	\$tipoBD
Descripcion	Almacena el tipo de base de datos (Ejemp: pgsq, mysql, etc)
Nombre	\$debug
Descripcion	Para dar salida a la cadena de comandos SQL a ejecutar. Emplear solo en desarrollo. Tipo: boolean, Valor por defecto: false (0).
Nombre	\$Campos
Descripcion	Para guardar los nombres de columnas de la entidad a referenciar. Tipo: array()
Nombre	\$CTipos
Descripcion	Arreglo con los tipos de datos de los campos, estos pueden ser solo: <ul style="list-style-type: none"> o integer -> Para los campos numéricos enteros o double -> Para los campos numéricos con punto flotante (decimales) o string_TAM -> Para los campos de cadena. indicar en TAM la cantidad máxima de caracteres permitidos.
Nombre	\$aError
Descripcion	Arreglo para el tratamiento o personalización de los errores del gestor. Debe formarse como sigue: array ('nombre restricción'=>'Msg de error')
Nombre	\$nTabla
Descripcion	Nombre de la entidad
Nombre	\$llaves
Descripcion	Índice del campo llave, en caso de ser una llave compuesta, especificarlo en forma de arreglo.
Responsabilidades	
Nombre:	function construct(\$nC, \$tC, \$nT, \$k, \$esq="", \$aE=")
Descripcion:	Constructor de la clase. Parámetros: <ul style="list-style-type: none"> o \$nC-> array, de nombre de los campos de la entidad en cuestión o \$nT-> string, nombre de la entidad en cuestión o \$k ->integer (llaves simples) ó array (llaves compuestas), índice(s) del (los) campos llave de la entidad en cuestión o \$esq -> string, nombre del esquemas a que pertenece la entidad o \$aE -> Array, arreglo de errores personalizados. Se empleara fundamentalmente cuando la entidad o campos de esta posean restricciones de cualquier tipo
Nombre:	function destruct()
Descripcion:	Destructor de la clase. Controla el objeto de conexión, destruyéndole al destruir la última instancia de entidades creadas con anterioridad.
Nombre:	function getCampos()
Descripcion:	Permite obtener los campos.
Nombre:	function getLlaves()
Descripcion:	Permite obtener las llaves.
Nombre:	function getCacheDir()
Descripcion:	Permite obtener la dirección de cache.
Nombre:	function setConfiguracion(\$cfg)
Descripcion:	Permite modificar o insertar el valor de la configuración.
Nombre:	function setEsquema(\$esq)
Descripcion:	Concatenado de nombres de esquemas. Nos ahorrara tener que referirnos

	a estos en las consultas <u>Parámetros:</u> <ul style="list-style-type: none"> o \$esq -> string, nombre del esquema a que pertenece la entidad en cuestión
<u>Nombre:</u>	<u>function setCacheDir(\$cd)</u>
<u>Descripción:</u>	Permite especificar el directorio a emplear para hacer cache de recordsets <u>Parámetros:</u> <ul style="list-style-type: none"> o \$cd -> string, cadena con el path completo del directorio para caché <u>Return:</u> NADA
<u>Nombre:</u>	<u>function getOConexion()</u>
<u>Descripción:</u>	Obtención del objeto de conexión a la Base de datos. Obsérvese que la configuración total proviene de un fichero de cfg. <u>Configuración</u> -> tiposerverbd, nombre/ip, puerto, usuario, contraseña y nombre de la base de datos a emplear.
<u>Nombre:</u>	<u>function exec (\$sql)</u>
<u>Descripción:</u>	Ejecución de comandos sql. <u>Parámetros:</u> <ul style="list-style-type: none"> o \$sql -> string, cadena de comandos SQL a ejecutar
<u>Nombre:</u>	<u>function Consulta (\$sql, \$esq = "")</u>
<u>Descripción:</u>	Ejecución de comandos sql, consultas (SELECT) preferiblemente, aunque en determinadas ocasiones, dependiendo de la situación presentada, se podrá emplear para cualquier cadena de comandos SQL. <u>Parámetros:</u> <ul style="list-style-type: none"> o \$sql -> string, cadena de comandos SQL a ejecutar. o \$esq ->string, cadena de los esquemas a que pertenecen las tablas empleadas en la consulta <u>Retorno:</u> array de datos si la consulta devuelve valores, 0 si no es así.
<u>Nombre:</u>	<u>function getErrores(\$fAfect = 0)</u>
<u>Descripción:</u>	Obtener los errores personalizados o del gestor en caso de que existan <u>Retorno:</u> integer, 0, si no error, string, [mensaje], si error.
<u>Nombre:</u>	<u>function Insertar(\$val, &\$mode=0)</u>
<u>Descripción:</u>	Método de inserción de datos <u>Parámetros:</u> <ul style="list-style-type: none"> o \$val -> array, del tipo campo=>valor a insertar en la entidad. o \$mode ->entero, indica si retornara el ID autogenerado después de ejecutar la operación.
<u>Nombre:</u>	<u>function Modificar(\$val, \$vllaves)</u>
<u>Descripción:</u>	Método de modificación de datos <u>Parámetros:</u> <ul style="list-style-type: none"> o \$val -> array, del tipo campo=>valor a modificar en la entidad. o \$vllaves -> array, valores de las llaves de la fila a modificar. Su posición en el array debe coincidir con la posición del nombre del campo dentro del array de campos. Cuando la llave es simple, se puede pasar directamente el valor de la llave(es lo mas común).

Nombre:	function Eliminar(\$val, \$modo = 0)
Descripción:	Método de eliminación de datos. Su definición generalmente satisface las eliminaciones clásicas (filtrando por la(s) llave(s)), en los casos en que se realicen otros tipos, redefinir o crear un nuevo método en la típica de la entidad dada. GENERALMENTE no es necesaria de redefinición. <u>Parámetros:</u> <ul style="list-style-type: none"> o \$val -> string (llaves sencillas) ó array de strings (llaves compuestas), valores de llaves de tuplas a eliminar. Su índice en el array debe coincidir con el índice del campo llave del arreglo de nombres de campo. o \$mode -> Empleo de eliminación "VIRTUAL". *** Solo emplear en los casos de entidades con campo actual para el control de cambios.
Nombre:	function abrirTrans()
Descripción:	Comenzar un bloque de transacciones.
Nombre:	function aceptarTrans(\$aceptar)
Descripción:	Aceptar o no una transacción en curso. <u>Parámetros:</u> <ul style="list-style-type: none"> o \$aceptar -> boolean, indica si aceptar (commit) o cancelar (rollback) una transacción en curso. Normalmente algunos gestores (Ejemp: postgresql) automáticamente después de un error en una de las consultas del bloque, ignoran el resto de estas.
Nombre:	function setDataSQL (\$sql)
Descripción:	Nos permite redefinir la secuencia de comandos SQL a emplear en el método getData. Generalmente se emplea por las típicas que responden a nomencladores "complejos", cuando esta debe redefinir el método antes mencionado.
Nombre:	function getDatos(\$cc = array(0,1), \$order=1, \$pActual=2, \$exc = 0, \$vActual= 1)
Descripción:	Obtener los datos de un nomenclador para el clásico llenado de las listas en los formularios de actualización (consulta sobre la llave y un atributo descriptivo). Su implementación incluye el empleo de cache, en el caso de que esta haya sido especificada. Generalmente no tiene que ser redefinido en las típicas, solo cuando el nomenclador no es "simple". <u>Parámetros:</u> <ul style="list-style-type: none"> o \$cc -> array (), arreglo con los índices de los campos a consultar, por defecto el 0 y el 1. o \$order -> integer, índice del campo por el que se desea ordenar la consulta, por defecto, se realiza por el campo descriptivo del nomenclador. o \$actual -> integer, indica pos de cuando consultar los vigentes (>0) o todos (-1). o \$exc -> integer, bandera para excluir algún elemento en blanco que pudiera tener un nomenclador en un momento dado.
Nombre:	function limpiarCache(\$nadd = ")
Descripción:	Permite limpiar la cache en tiempo de ejecución. Muy empleado por los CU que tienen que ver con la actualización directa de las entidades que emplean cache para mejorar el rendimiento de la aplicación. IMPORTANTE tener esto en cuenta. <u>Parámetros:</u>

	<ul style="list-style-type: none"> o \$nadd: string, permite especificar algún elemento adicional para los nombres de los ficheros en cache, diferenciando así las diferentes consultas cacheadas realizadas sobre una misma entidad.
Nombre:	function Valida (& \$array, \$accion = 'INS')
Descripción:	<p>Declaración del método de validación de datos. En cada típica, antes de realizar las operaciones básicas debe implementarse un nivel de validación, este método se empleara en todos los casos para ello.</p> <p><u>Parámetros:</u></p> <ul style="list-style-type: none"> o \$array -> arreglo del tipo campo->valor que contiene la información a validar. o \$accion -> string de 3caracteres que indica si se llama al método desde la operación INSERTAR o desde MODIFICAR. Esto es necesario, pues en insertar se valida de forma obligatoria la actualización del 100% de los atributos de la entidad dada que lo requieran, lo contrario ocurre en modificar, solo se validan los atributos que se desean modificar. <p><u>Retorno:</u> Cadena de errores si los hubo, vacío en caso contrario.</p>
Nombre:	function ValidarTipos(& \$datos)
Descripción:	Validación de tipos de datos: (función actual: convertir las llaves del array de datos a strings)

✓ Clase meSimple

Es una clase abstracta que constituye la base para la implementación de las típicas que responderán a los nomencladores simples del modelo de persistencia dado. Redefine las operaciones básicas con la funcionalidad de la validación dada.

Representa entidades cuya estructura responde al siguiente patrón: idALGO, ALGO, actual. Donde 'ALGO' representa la descripción del atributo principal de los nomencladores clásicos.

Define las operaciones básicas que pudieran realizarse a una entidad (INSERT, DELETE, UPDATE) para los nomencladores simples.

Nombre: meSimple	
Tipo de clase Clase abstracta	
Atributos	
Nombre	\$vmsg
Descripción	Almacena el mensaje de validación que debe dársele al usuario en caso de un error en la actualización del atributo descripción de este tipo de entidad.
Nombre	\$pObj
Descripción	Permite el chequeo de objeto persistentes

Responsabilidades	
Nombre:	<u>function construct(\$vm, \$nC, \$tC, \$nT, \$esq="", \$aE=")</u>
Descripción:	Constructor de la clase. Parámetros: <ul style="list-style-type: none"> ○ \$vm -> string, mensaje de validación. ○ El resto son los mismos de meBase...
Nombre:	<u>function Valida(& \$array, \$accion = 'INS')</u>
Descripción:	Validación de datos Obligatorios(*) y No Obligatorios Parámetros: <ul style="list-style-type: none"> ○ \$descrip -> string, valor de la descripción del elemento.
Nombre:	<u>function Insertar(\$array)</u>
Descripción:	Inserción de valores. Parámetros: <ul style="list-style-type: none"> ○ \$descrip -> string, elemento a insertar*
Nombre:	<u>function Modificar(\$array, \$vllave)</u>
Descripción:	Modificar valores. Parámetros: <ul style="list-style-type: none"> ○ \$array -> array, valor del elemento a modificar. ○ \$vllave -> integer, valor del campo llave a modificar.
Nombre:	<u>function Eliminar(\$id, \$modo=1)</u>
Descripción:	Método para eliminaciones. Redefine el de la clase base meBase, cambiando el parámetro <i>modo</i> a valor 1 por defecto para indicar eliminación "virtual". Parámetros: <ul style="list-style-type: none"> ○ \$val -> string (llaves sencillas) ó array de strings (llaves compuestas), valores de llaves de tuplas a eliminar. Su índice en el array debe coincidir con el índice del campo llave del arreglo de nombres de campo. ○ \$modo -> Empleo de eliminación "VIRTUAL". *** Solo emplear en los casos de entidades con campo <i>actual</i> para el control de cambios.

✓ Clase Conexión

Esta clase, como su nombre lo indica, se encarga de establecer la conexión a la base de datos a través de objetos nativos brindados por el entorno de desarrollo PHP.

Nombre: Conexion	
Tipo de clase Clase para la conexión.	
Atributos	
Nombre	\$pdo = NULL
Descripción	Referencia a un objeto de tipo PDO, utilizado para establecer la conexión con la base de datos.
Nombre	\$debug = false
Descripción	Variable que se utiliza para el debug.

Nombre	\$tipoBD
Descripción	Tipo de la base de datos a la que se hace la conexión, este valor se toma del stringConexion del fichero de configuración.
Nombre	\$cfgCon = array()
Descripción	Arreglo con información de la configuración de la conexión. Contiene el stringConexion, el usuario y el password para conectarse a la base de datos.
Nombre	\$error = array()
Descripción	Variable que contiene el error, si ocurre, de realizar una consulta a la base de datos
Responsabilidades	
Nombre:	function __construct()
Descripción:	Constructor de la clase. Vacío.
Nombre:	public function __destruct()
Descripción:	Destructor de la clase. Aquí se destruye el objeto PDO
Nombre:	private function getOConexion()
Descripción:	Método encargado de crear el objeto PDO, es decir, trata de establecer la conexión con la base de datos.
Nombre:	function setConfiguracion(\$cfg)
Descripción:	Método encargado de establecer la configuración de la conexión. Parámetro: o \$cfg -> array: contiene los datos del fichero de configuración referentes a la conexión.
Nombre:	public function __call(\$met, \$parametros)
Descripción:	Función mágica que se ejecuta cuando se hace referencia a un método que no existe. En este caso es usada para ejecutar las funciones del objeto PDO.
Nombre:	function __GET(\$prop)
Descripción:	Función mágica para obtener los valores de los atributos privados. En este caso solo devuelve el valor de la variable \$tipoBD.

➤ Clase consBase:

Esta clase es la base en toda la jerarquía de Acceso a Datos y es empleada para aportar contenido dinámico a las plantillas. Encapsula el objeto creado para la conexión.

Nombre: consBase	
Tipo de clase Clase que implementa las operaciones de CONSULTA.	
Atributos	
Nombre	\$esquema
Descripción	Información sobre los esquemas a emplear en la conexión.
Nombre	\$esqOut
Descripción	Salida de ejecución para los esquemas.
Nombre	\$oConexion = NULL

<u>Descripción</u>	Referencia al objeto Conexion. Esta referencia es estática para tener una sola instancia del objeto.
<u>Nombre</u>	\$aError = array();
<u>Descripción</u>	Arreglo de conversión de errores (debe ser array('nombre restricción'=>'Msg de error'))
<u>Nombre</u>	\$cobj = 0;
<u>Descripción</u>	Contador de objetos que utilizan \$oConexion. Esta variable se incrementa en el constructor y se decrementa en el destructor.
Responsabilidades	
<u>Nombre:</u>	function __construct(\$esq, \$aE="")
<u>Descripción:</u>	Constructor de la clase. <u>Parámetros:</u> <ul style="list-style-type: none"> o \$esq -> string, nombre del esquema donde se realizará la consulta. o \$aE -> array, arreglo de errores personalizados. Incrementa el contador de objetos Conexion
<u>Nombre:</u>	function __destruct()
<u>Descripción:</u>	Destructor de la clase. Decrementa el contador de objetos Conexion
<u>Nombre:</u>	function setEsquema(\$esq)
<u>Descripción:</u>	Método privado encargado de concatenar todos los esquemas que se van a utilizar en una consulta. De esta forma no habrá que referirse a ellos en las mismas. <u>Parámetro:</u> <ul style="list-style-type: none"> o \$esq -> nombre del esquema a que pertenece la entidad en cuestión.
<u>Nombre:</u>	function salidaEsquema(\$esq = "")
<u>Descripción:</u>	Salida de search_path para automatizar la búsqueda de tablas en el gestor de base de datos.
<u>Nombre:</u>	function Consulta(\$sql, \$esq = "", \$tipoRet = 'AMBOS')
<u>Descripción:</u>	Método encargado de hacer una consulta a la base de datos. <u>Parámetros:</u> <ul style="list-style-type: none"> o \$sql: consulta a realizar en la base de Datos. o \$esq: cadena de los esquemas a los que pertenecen las tablas empleadas en la consulta. o \$tipoRet: indica la configuración del array de salida, por defecto <ul style="list-style-type: none"> - COL, índice con nombre de la columna; - NUM, índice numérico, consecutivo; - AMBOS, índice y valores dobles, por columnas y numérico. El retorno de este método es un array de datos si la consulta devuelve valores, 0 si no es así o una cadena en caso de error.
<u>Nombre:</u>	function getErrores(\$fAfect = 0, \$pdoRec = NULL)
<u>Descripción:</u>	Método encargado de obtener los errores personalizados o del gestor de base de datos en caso de que existan. Retorna el número de filas afectadas por la consulta si no hay errores o una cadena con el mensaje de error.
<u>Nombre:</u>	function getRetorno(\$sql, \$adicEsq = "")
<u>Descripción:</u>	Este método es para algunas clases que lo usan pero ya no está en la tercera iteración.

❖ Diagramas de Interacción.

Los diagramas de secuencia y los diagramas de colaboración (ambos llamados diagramas de interacción) son dos de los cinco tipos de diagramas de UML que se utilizan para modelar los aspectos dinámicos de los sistemas. Un diagrama de interacción muestra una interacción, que consiste en un conjunto de objetos y sus relaciones, incluyendo los mensajes que se pueden enviar entre ellos.

Estos diagramas de interacción fueron utilizados para modelar los aspectos dinámicos de este sistema. La mayoría de las veces, esto implica modelar instancias concretas o prototípicas de clases, interfaces, componentes y nodos, junto con los mensajes enviados entre ellos, todo en el contexto de un escenario que ilustra un comportamiento. Los diagramas de interacción pueden utilizarse para visualizar, especificar, construir y documentar la dinámica de una sociedad particular de objetos, o se pueden utilizar para modelar un flujo de control particular de un caso de uso.

Los diagramas de interacción no son sólo importantes para modelar los aspectos dinámicos de este Sistema, sino también para construir sistemas ejecutables por medio de ingeniería directa e inversa. Al igual que los demás diagramas, los diagramas de interacción pueden contener notas y restricciones.

❖ Diagramas de Secuencia

Los diagramas de secuencia desarrollados, destacan la ordenación temporal de los mensajes. Los diagramas de secuencia tienen dos características que los distinguen de los diagramas de colaboración: la línea de vida de un objeto y el foco de control. En nuestro sistema se han desarrollado diagramas de secuencia por cada realización de los Casos de Uso que componen el mismo.

A continuación, se muestra un diagrama de secuencia correspondiente a un Caso de Uso de Cada Subsistema, el resto de dichos diagramas de secuencia se encuentran documentados en los anexos de este trabajo de diploma.

■ Paquete Recuperaciones.

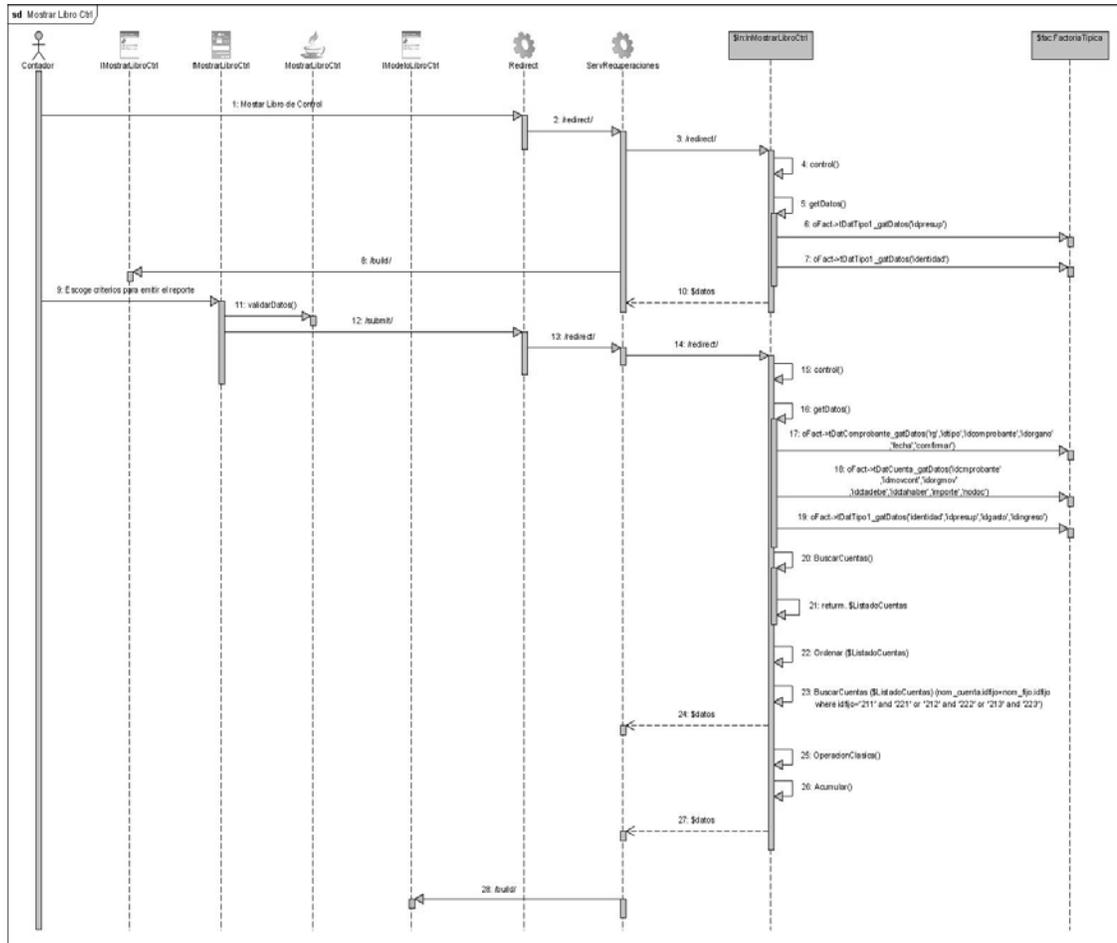


Fig. 42 Diagrama de Secuencia (Mostrar Libro de Registro y Control)

■ Paquete de Traspasos

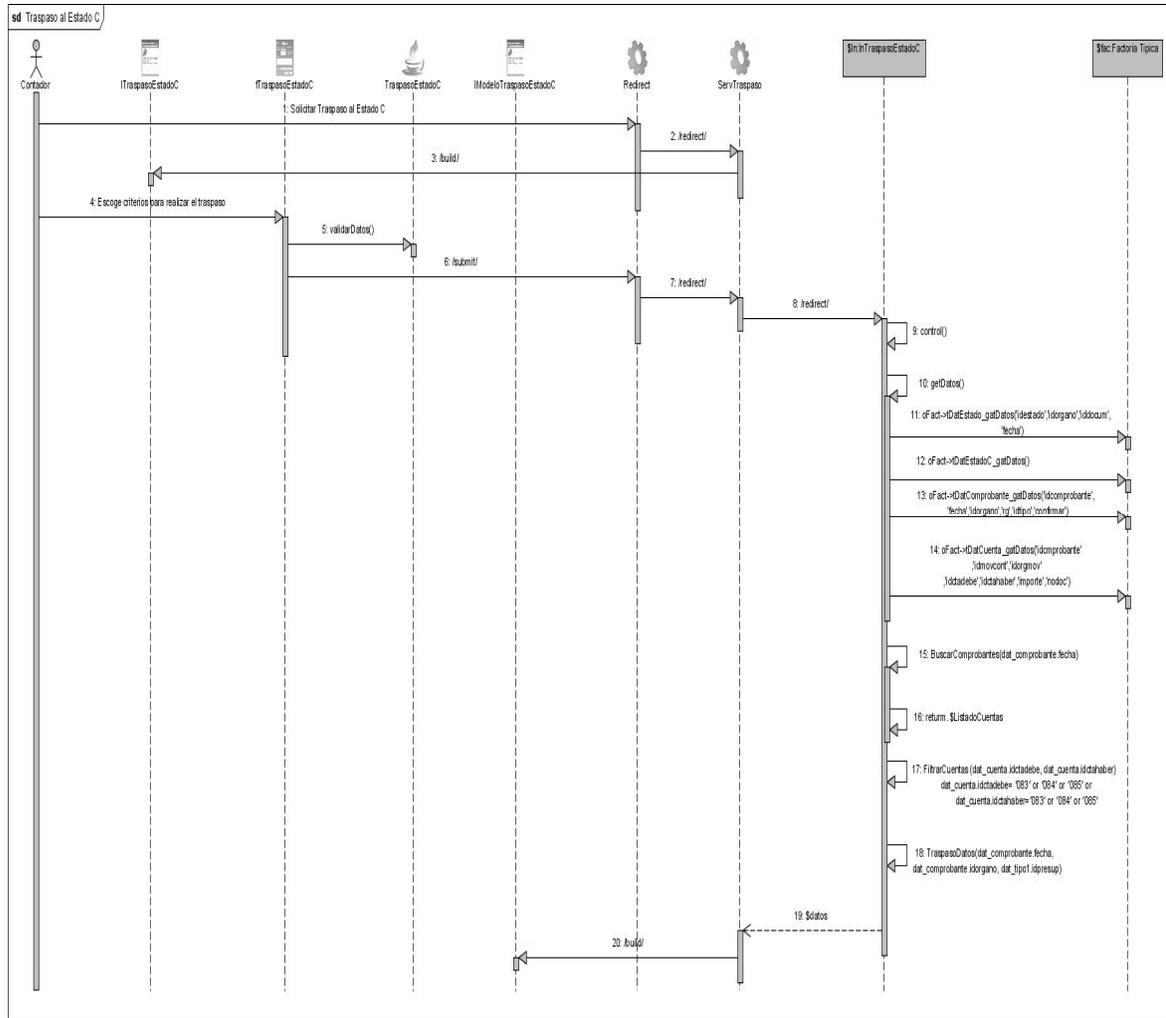


Fig. 43 Diagrama de Secuencia (Realizar Traspaso al Estado C)

■ Paquete Inicio de Año Contable

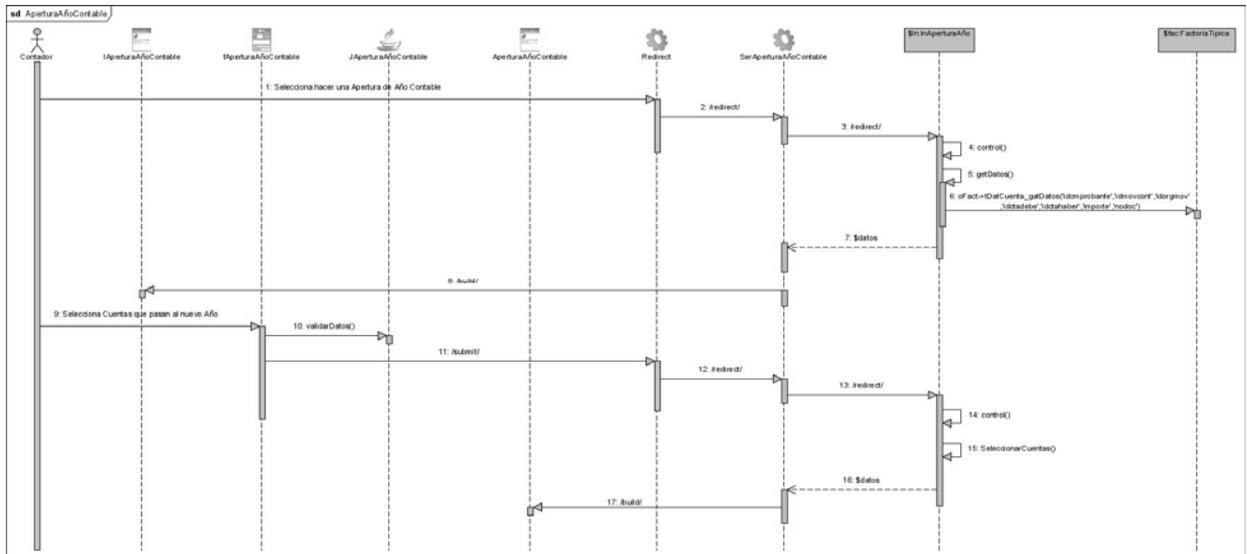


Fig. 44 Inicio del Año Contable (Realización Principal)

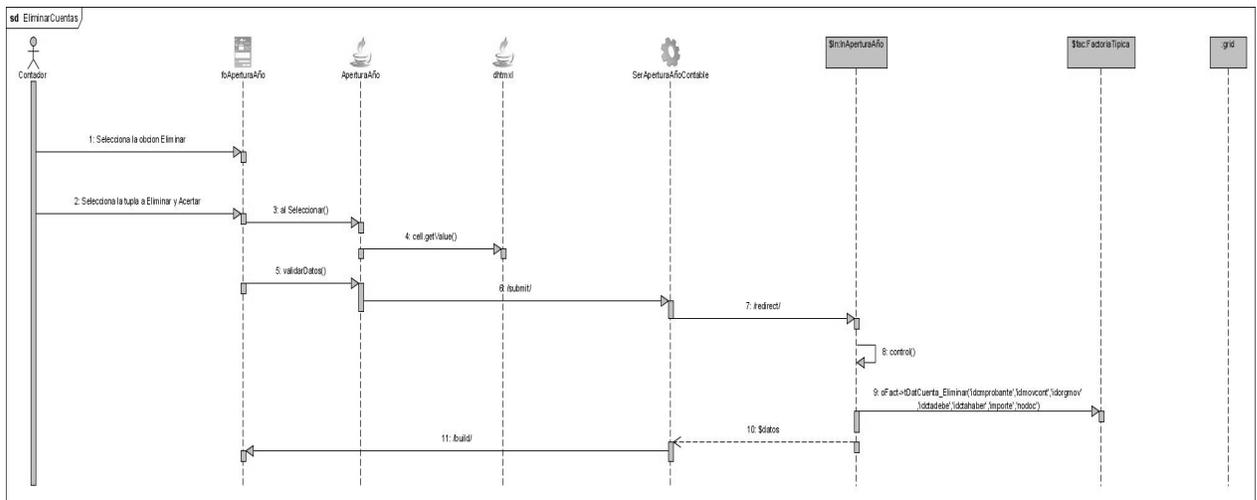


Fig. 45 Inicio del Año Contable (Realización Eliminar)

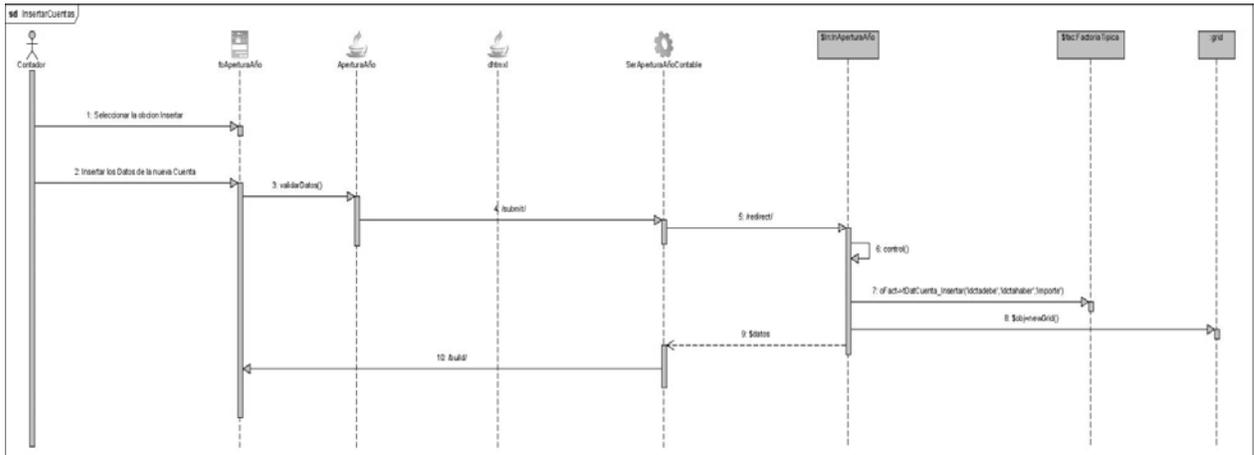


Fig. 46 Inicio del Año Contable (Realización Insertar)

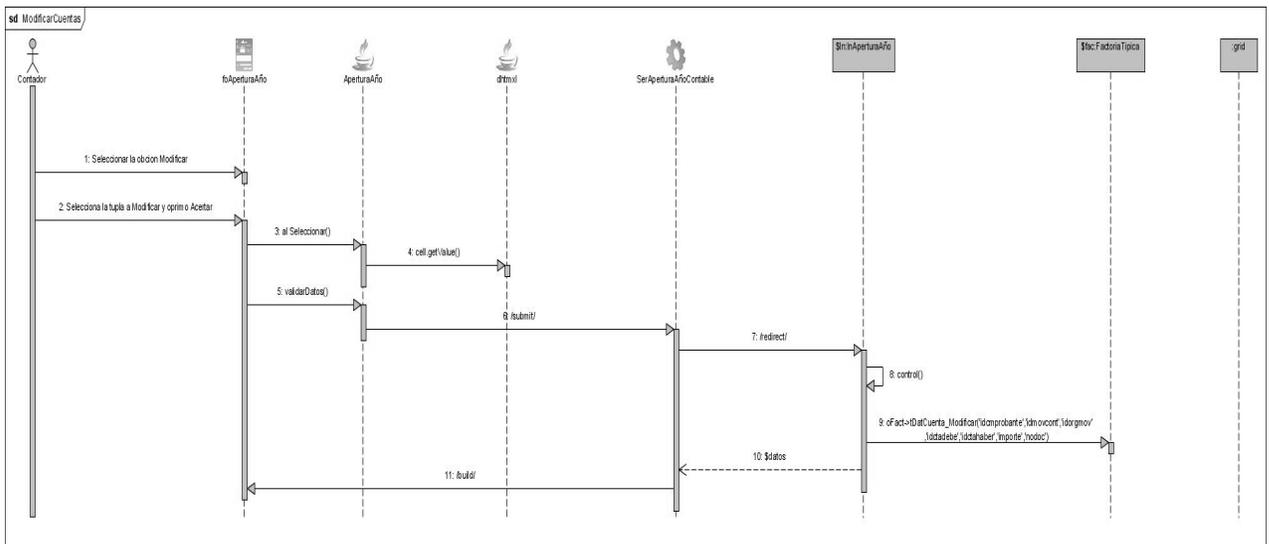


Fig. 47 Inicio del Año Contable (Realización Modificar)

Nombre: dat_comprobante		
Descripción: Se guarda por comprobante y el número de registro, el órgano a que pertenece, la fecha y el tipo de operación.		
Atributo	Tipo	Descripción
idcomprobante	varchar(12)	identificador del comprobante (se forma concatenando los valores fecha(año)+idorgano+rg)
idorgano	integer	órgano propio, viene del nom_organo
rg	smallint	registro del comprobante se inicializa por año en 1
fecha	date	fecha de realización del comprobante(menor o igual a la fecha actual)
idtipo	smallint	viene de nom_tipo
confirmar	bit(1)	marca de confirmación del comprobante(1: no se puede realizar ninguna modificación 0:permite modificar)

Nombre: dat_Anexo 1		
Descripción: Corresponde al modulo de Estados Financieros, se guardan los movimientos del anexo1 para un órgano y una fecha.		
Atributo	Tipo	Descripción
idestado	varchar(13)	Identificador
mov	smallint	contador para saber a cantidad de movimientos asociados a un anexo1
idorgmov	integer	órgano al que se le realiza el mov contable, viene de nom_organo
idpresup	smallint	viene de nom_presup
identidad	smallint	viene de nom_entidad
idgasto	smallint	viene de nom_gasto
canthomb	integer	Cantidad de hombres
impasig	numeric(12,2)	importe asignado
impgast	numeric(12,2)	importe gasto

Nombre: dat_cuenta		
Descripción: se registran los movimientos contables asociados a un registro y a un órgano financiero		
Atributo	Tipo	Descripción
idcomprobante	varchar(12)	viene de dat_comprobante
idmovcont	smallint	identificador del movimiento contable se incrementa por la función ft_cuenta_idmovcont
idorgmov	integer	identificador del órgano al que se le hace el movimiento(los órganos a mostrar se selecciona de nom_organo a través de una relación con nom_plantilla)
idctadebe	smallint	viene de nom_cuenta
idctahaber	smallint	viene de nom_cuenta
importe	numeric(12,2)	importe de la cuenta
nodoc	varchar(10)	numero del documento: solo # hasta 10 puede empezar con 0

Nombre: dat_estado		
Descripción: Corresponde al modulo de Estados Financieros, se guardan los documentos por un órgano y una fecha		
Atributo	Tipo	Descripción
idestado	varchar(13)	identificador de estados se forma concatenando los valores fecha+idorgano+iddocum (Ej:0220060001001) *** fecha (los primeros 6 dígitos correspondiente al mes y al año 022006)
idorgano	integer	órgano propio, viene de nom_organo
iddocum	smallint	viene de nom_docum
fecha	date	fecha en que se crea el estado

Nombre: dat_estadoa		
Descripción: Corresponde al modulo de Estados Financieros, se guardan los datos correspondientes al Estado A		
Atributo	Tipo	Descripción
idestado	varchar(13)	viene de dat_estado
mov	smallint	contador para saber a cantidad de movimientos asociados a un anexo1
idorgmov	integer	identificador del órgano al que se le realiza el movimiento viene de nom_organo
idcuenta	smallint	viene de nom_cuenta
impdebe	numeric(12,2)	importe al debe
imphaber	numeric(12,2)	importe al haber

Nombre: dat_estadob		
Descripción: Corresponde al modulo de Estados Financieros, referente a los datos del estado B sección 1.		
Atributo	Tipo	Descripción
idestado	varchar(13)	identificador
mov	smallint	contador para saber a cantidad de movimientos asociados a un estadob
idorgmov	integer	órgano al que se le realiza el mov contable, viene de nom_organo
idingreso	smallint	viene de nom_ingreso
impcapt	numeric(12,2)	importe captado
imptransf	numeric(12,2)	importe transferido

Nombre: dat_tipo1		
Descripción:		
Atributo	Tipo	Descripción
idcomprobante	varchar(12)	viene de dat_cuenta
idmovcont	integer	viene de dat_cuenta
idpresup	smallint	viene del nom_presup
identidad	smallint	viene del nom_entidad
idgasto	smallint	viene del nom_gasto
idingreso	smallint	viene del nom_ingreso
transferido	bit(1)	marcar si es transferido o no el comprobante
canthomb	smallint	cantidad de personas que se le paga por nominas
idcliente	smallint	viene del nom_cliente
idpersona	smallint	viene de la tabla persona
idretencion	smallint	viene de nomretencion
fechabanco	date	fecha banco

Nombre: dat_tipo2		
Descripción: Tabla que se utiliza para un análisis de los hospitales militares		
Atributo	Tipo	Descripción
idcomprobante	varchar(12)	viene de dat_cuenta
idmovcont	integer	viene de dat_cuenta
idcosto	smallint	viene de nom_costo
idanalisis	smallint	viene de nom_analisis

Nombre: nom_cuenta		
Descripción: nomenclador de cuentas contables		
Atributo	Tipo	Descripción
idcuenta	smallint	identificador de cuenta(autonumerico)
codigo	varchar(7)	Codigo de la cuenta (###.#.#)
denom	varchar(200)	denominacion de la cuenta
balance	bit(1)	para saber si es una cuenta de patrimonio (1) o de orden(0)
naturaleza	bit(1)	si es una cuenta deudora(0) o acreedora(1)
contabiliza	bit(1)	si por esa cuenta se contabiliza(1), no se contabiliza por ella(0)
analisis	bit(1)	Si esa cuenta lleva análisis (1)(grupo,entidad,gasto,ingreso,costo,retenciones...etc., no lleva análisis(0)
idgrupo	smallint	viene de nom_grupo
padre	smallint	codigo del padre de la cuenta
idfijo	smallint	viene de nom_fijo
actual	bit(1)	indicador de uso

Nombre: nom_entidad		
Descripción: Nomenclador de Entidad o Especialidad		
Atributo	Tipo	Descripción
identidad	smallint	identificador de entidad
codigo	varchar(2)	codigo de la entidad o especialidad
denom	varchar(70)	denominacion de la entidad o especialidad
abrev	varchar(20)	abreviatura de la entidad
actual	bit(1)	indicador de uso

Nombre: nom_fijo		
Descripción:		
Atributo	Tipo	Descripción
idfijo	smallint	identificador de categoría de las cuentas por grupos
codigo	varchar(3)	codigo de la categoría de la cuenta
denom	varchar(60)	denominacion de la categoría de la cuenta
actual	bit(1)	indicador de uso

Nombre: nom_gasto		
Descripción: Nomenclador de Gastos en las FAR		
Atributo	Tipo	Descripción
idgasto	smallint	identificador del gasto o partida
codigo	varchar(10)	codigo del gasto
denom	varchar(40)	denominacion del gasto
actual	bit(1)	indicador de uso

Nombre: nom_grupo		
Descripción: Nomenclador de grupo de cuentas		
Atributo	Tipo	Descripción
idgrupo	smallint	identificador del grupo de cuentas
denom	varchar(80)	denominacion del grupo de cuentas
padre	smallint	padre del grupo de cuentas
actual	bit(1)	indicador de uso

Nombre: nom_ingreso		
Descripción: Nomenclador de ingresos al presupuesto del estado		
Atributo	Tipo	Descripción
idingreso	smallint	identificador de ingreso
codigo	varchar(6)	codigo del ingreso (solo números)
denom	varchar(60)	denominacion del ingreso
actual	bit(1)	indicador de uso

Nombre: nom_organo		
Descripción: Nomenclador de Órganos Financieros		
Atributo	Tipo	Descripción
idorgano	integer	identificador del organo
denom	varchar(30)	denominacion del organo
ctagasto	bigint	cuenta bancaria para gastos
ctainvers	bigint	cuenta bancaria para inversiones
ctaingreso	bigint	cuenta bancaria para ingresos
ctafondo	bigint	cuenta bancaria para fondos de operaciones
otras	bigint	otras cuentas bancarias
padre	integer	determina si el órgano es un consumidor o administrador
actual	bit(1)	Indicador de uso

Nombre: nom_presup		
Descripción: Nomenclador de Grupos Presupuestarios		
Atributo	Tipo	Descripción
idpresup	smallint	identificador del grupo presupuestario
codigo	varchar(8)	codigo del presupuesto solo números y punto(111111.2)
denom	varchar(40)	denominacion del grupo presupuestario
actual	bit(1)	indicador de uso

Nombre: nom_retencion		
Descripción: nomenclador de retención bancaria		
Atributo	Tipo	Descripción
idretencion	smallint	identificador de retención al banco
denom	varchar(30)	denominacion de retención al banco
actual	bit(1)	indicador de uso

De todas las tablas que el sistema utiliza, existen 4 tablas principales, que son tablas de datos, las demás, son tablas referidas a los nomencladores que se relacionan con las mismas. Existen también tablas de datos pero correspondientes a los diferentes estados que intervienen en los traspasos, pero son tablas temporales que se crean para contener la información de cada traspaso y no constituyen la base de datos principal del Módulo de Contabilidad, el mismo las utilice en una que otra ocasión.

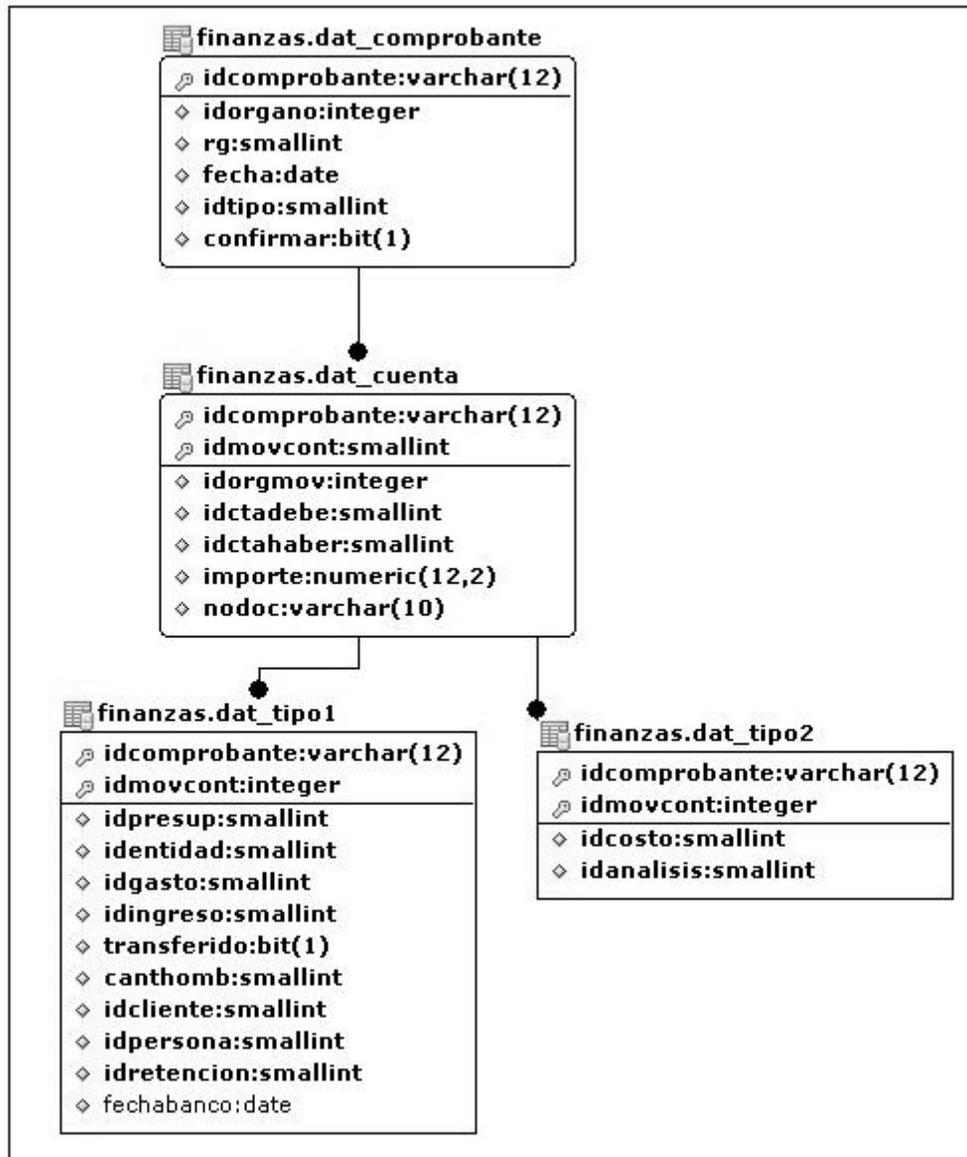


Fig. 49 Tablas principales de la Base de Datos.

❖ Conclusiones:

En este capítulo se comenzó a desarrollar la propuesta solución para el sistema sugerido. Este desarrollo se enmarcó en el análisis profundo de los procesos del negocio, obteniendo un listado de las principales funcionalidades que debe tener el sistema cuando se comiencen a automatizar los procesos más importantes que lo componen. Este capítulo sirvió de espacio para describir, detalladamente, todas las acciones de los actores del sistema en los casos de uso con los que interactúan, de modo que quedan claros los objetivos fundamentales a la hora de comenzar la construcción de la propuesta solución.

Capítulo



“Implementación”

Introduccion

Diagrama de Componentes

Diagrama de Despliegue

Conclusiones

❖ Introduccion

En la implementación se comienza con el resultado del diseño y se implementa el sistema en términos de componentes, es decir, ficheros de código fuente, scripts, ficheros de código binario, ejecutables y similares. El flujo de trabajo de diseño se propone crear un plano del modelo de implementación, por lo que sus últimas actividades están vinculadas a la creación del modelo de despliegue. El flujo de trabajo de implementación describe cómo los elementos del modelo del diseño se implementan en términos de componentes y cómo estos se organizan de acuerdo a los nodos específicos en el modelo de despliegue.

Los diagramas de despliegue y componentes que se muestran, son artefactos generados en este flujo de trabajo y que conforman lo que se conoce como un Modelo de Implementación al describir los componentes a construir y su organización y la dependencia entre nodos físicos en los que funcionará a aplicación. Afortunadamente, la mayor parte de la arquitectura del sistema es capturada durante el diseño, siendo el propósito principal de la implementación el desarrollar la arquitectura y el sistema como un todo. De forma más específica, los propósitos de la implementación son:

- Planificar las integraciones de sistema necesarias en cada iteración. Se sigue para ello un enfoque incremental, lo que da lugar a un sistema que se implementa en una sucesión de pasos pequeños y manejables.
- Distribuir el sistema asignando componentes ejecutables a nodos en el diagrama de despliegue. Esto se basa fundamentalmente en las clases activas encontradas durante el diseño.
- Implementar las clases y subsistemas encontrados durante el diseño. En particular, las clases se implementan como componentes de fichero que contienen código fuente.
- Probar los componentes individualmente, y a continuación integrarlos compilándolos y enlazándolos en uno o más ejecutables, antes de ser enviados para ser integrados y llevar a cabo las comprobaciones de sistema.

❖ Diagrama de Componentes:

Los diagramas de componentes mostrados, son usados para estructurar el modelo de implementación en términos de subsistemas de implementación y mostrar las relaciones entre los elementos de implementación.

El uso más importante de estos diagramas es mostrar la estructura de alto nivel del modelo de implementación, especificando:

- Los subsistemas de implementación y sus dependencias a la hora de importar código.
- Organizar los subsistemas de implementación en capas.

La descripción de los componentes del software propuesto y sus dependencias entre si , se ven reflejados en el diagrama de componentes. Este diagrama muestra la estructura física de los conceptos y las funcionalidades descritas por clases, objetos, relaciones y colaboraciones. Se representa como un grafo de componentes software unidos por medio de relaciones de dependencia. Muestra la organización y las dependencias lógicas entre un conjunto de componentes software, sean éstos componentes de código fuente, librerías, binarios o ejecutables. No es necesario que un diagrama incluya todos los componentes del sistema, normalmente se realizan por partes.

Lo que distingue a un diagrama de componentes de otros tipos de diagramas es su contenido.

Normalmente contienen componentes, interfaces y relaciones entre ellos. Y como todos los diagramas, también puede contener paquetes utilizados para agrupar elementos del modelo.

Los paquetes en estos diagramas de componentes representan una división física del sistema. Los paquetes se organizan en una jerarquía de capas donde cada capa tiene una interfaz bien definida.

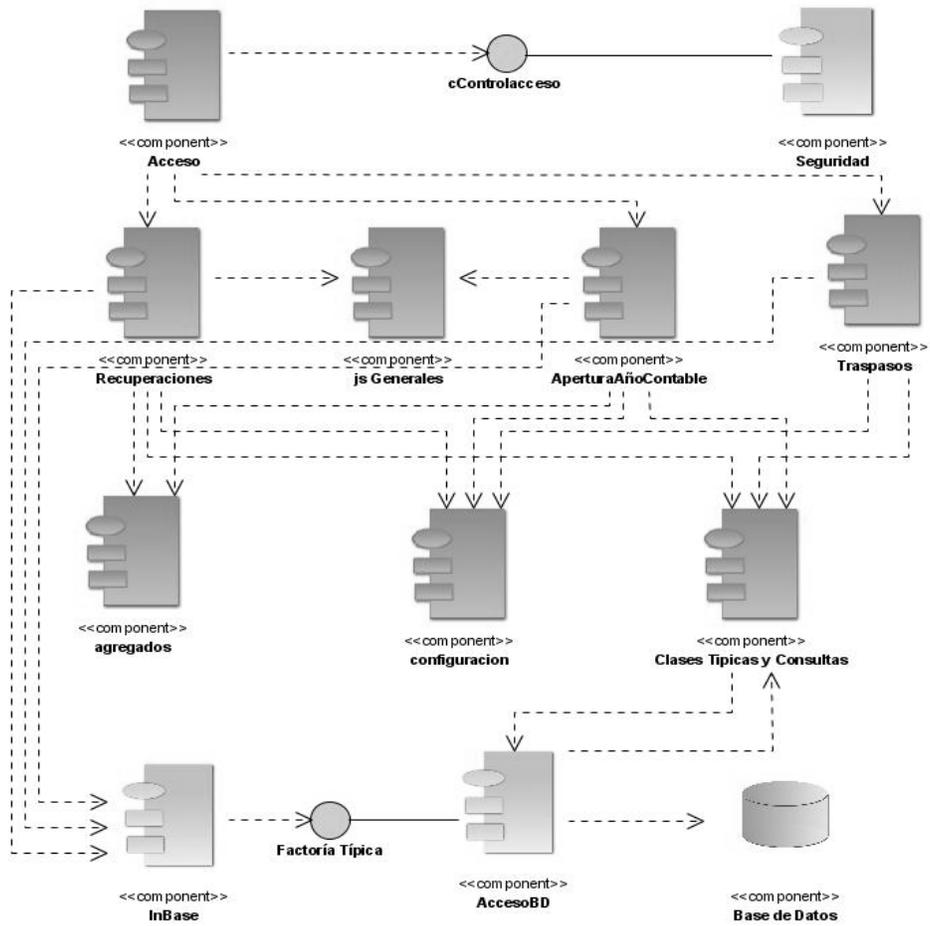


Fig. 50 Diagrama de Componentes.

A partir de este diagrama general de componentes de este sistema, se especifica el contenido de los principales paquetes de componentes reflejados en el mismo.

- Paquetes de Componentes de Recuperaciones y Traspasos:

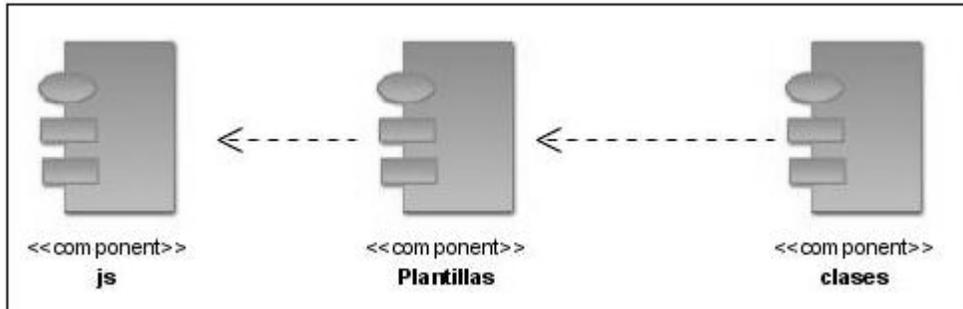


Fig. 51 Paquetes de Recuperaciones y Traspasos

Estos paquetes, dentro contienen, a su vez, tres tipos de subpaquetes de componentes: uno dedicado a las clases propias (lógica de negocio) de los casos de uso correspondientes, otro para las plantillas de dichas clases, y un tercero para agrupar los archivos js pertenecientes a las plantillas usadas. La estructura de estos paquetes de Traspasos y Recuperaciones es prácticamente la misma, solamente difieren en el contenido de los subpaquetes que lo integran.

- Subpaquete de Clases de Recuperaciones:

Este Subpaquete contiene todas las clases de lógica de negocio pertenecientes a cada caso de uso del paquete de Recuperaciones.

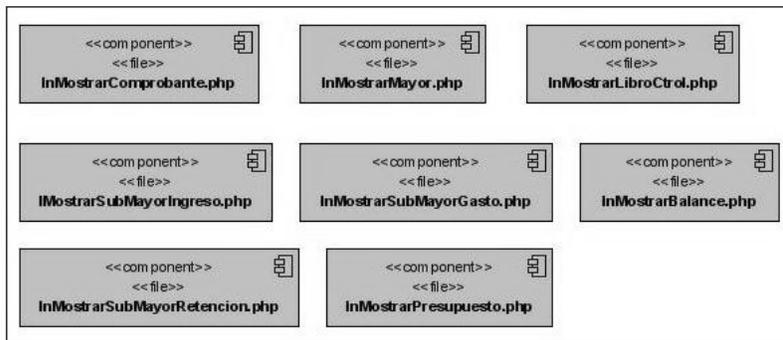


Fig. 52 Subpaquete de Clases de Recuperaciones.

■ Subpaquetes de Plantillas de Recuperaciones:

Este Subpaquete incluye todos los componentes correspondientes a los archivos php por cada caso de uso, que son dos pues en cada caso de uso existen 2 interfaces: una para la realización de la operación y otra para mostrar el modelo generado al final de la realización de cada caso de uso.

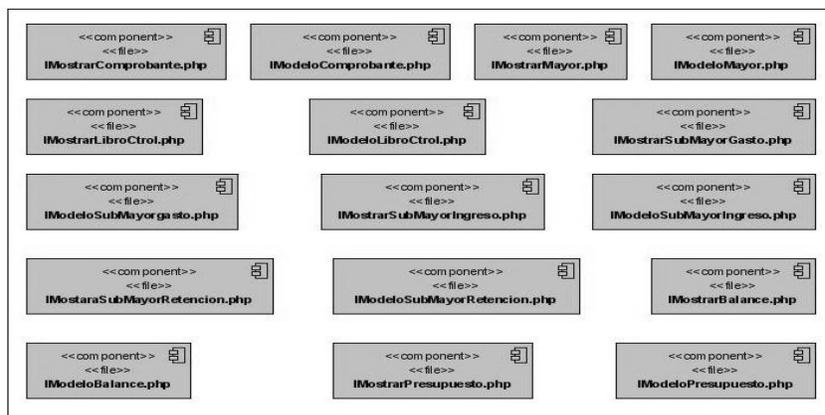


Fig. 53 Subpaquete de plantillas de Recuperaciones.

■ Subpaquete de JS de Recuperaciones:

Como su nombre lo indica, este Subpaquete recoge los archivos js pertenecientes a cada recuperación.

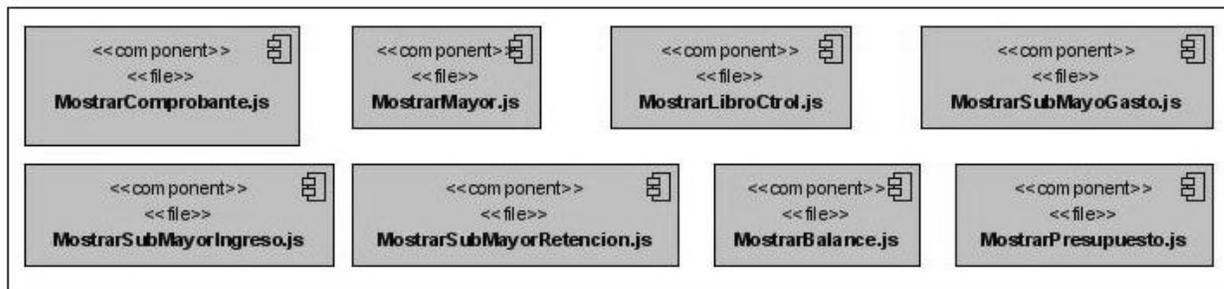


Fig. 54 Subpaquete de JS de Recuperaciones

- Subpaquete de Clases de Traspasos:

Este Subpaquete, al igual que el de Recuperaciones, contiene las clases lógicas de negocio por cada caso de uso correspondiente al Subpaquete de Traspasos.

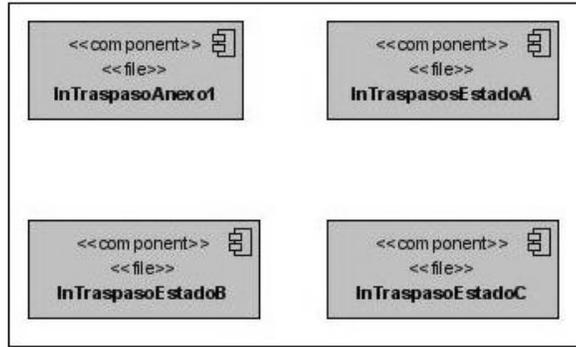


Fig. 55 Subpaquete de Clases de Traspaso.

- Subpaquete de Plantillas de Traspasos:

Este Subpaquete, al igual que su homologo para las recuperaciones, incluye los archivos php correspondientes a las dos interfaces usadas en cada caso de uso.

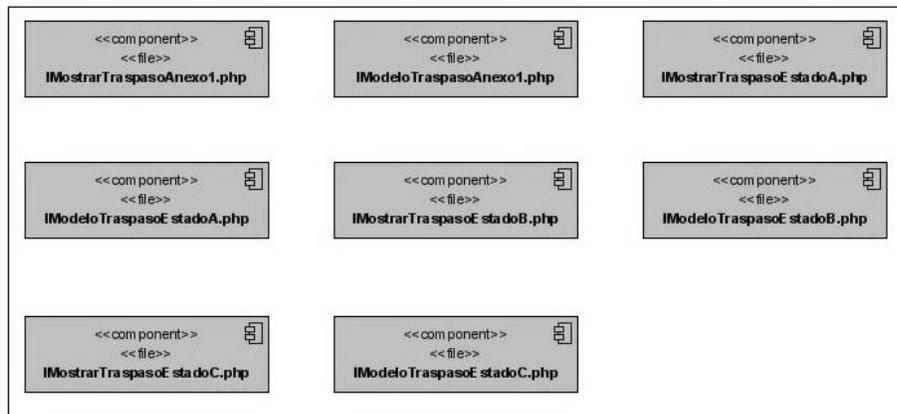


Fig. 56 Subpaquete de Plantillas de Traspasos.

- Subpaquete de JS de Traspasos.

Estos componentes que incluye este Subpaquete corresponden a cada archivo JS por cada traspaso.

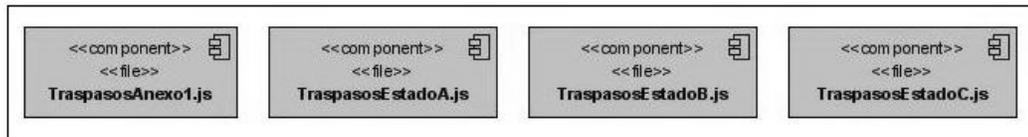


Fig. 57 Subpaquete de JS de Traspasos.

El diagrama que se presenta en a continuación tiene como objetivo figurar también la estructura general de la aplicación en desarrollo, en términos de componentes, el mismo está estructurado por capas, representando así el estilo arquitectónico utilizado en el sistema. Los subsistemas que se incluyen en el mismo son agrupados por el criterio de funcionalidad. Estos están compuestos por sus respectivos paquetes de componentes que se conforman en dependencia de cada sistema que pueden ser por casos de usos y representan una división física del sistema, dígame: Recuperaciones, Traspasos e Inicio de Año Contable.

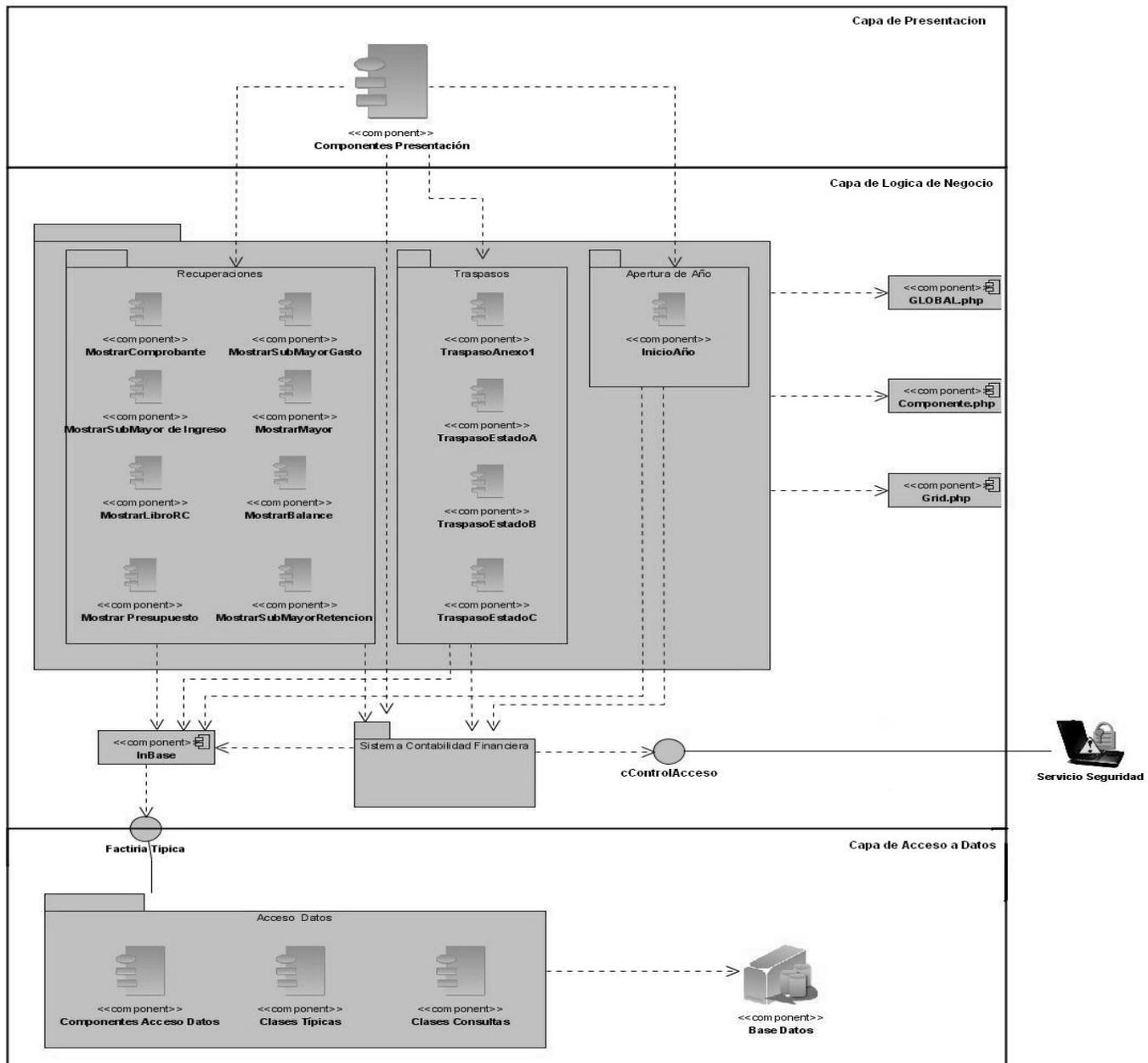


Fig. 58 Diagrama de Componentes II (Distribucion por capas de los componentes)

❖ Diagrama de Despliegue:

El diagrama de despliegue reflejado muestra las relaciones físicas entre los componentes hardware y software en el sistema final, es decir, la configuración de los elementos de procesamiento en tiempo de ejecución y los componentes software (procesos y objetos que se ejecutan en ellos). Estarán formados por instancias de los componentes software que representan manifestaciones del código en tiempo de ejecución (los componentes que sólo sean utilizados en tiempo de compilación deben mostrarse en el diagrama de componentes).

Este diagrama de despliegue es un grafo de nodos unidos por conexiones de comunicación. Un nodo puede contener instancias de componentes software, objetos, procesos (caso particular de un objeto). En general un nodo será una unidad de computación de algún tipo, desde un sensor a un mainframe. Las instancias de componentes software pueden estar unidas por relaciones de dependencia, posiblemente a interfaces (ya que un componente puede tener más de una interfaz).

La mayoría de las veces el modelado de la vista de despliegue estática implica modelar la topología del hardware sobre el que se ejecuta el sistema. Los diagramas de despliegue son fundamentalmente diagramas de clases que se ocupan de modelar los nodos de un sistema. Aunque UML no es un lenguaje de especificación hardware de propósito general, se ha diseñado para modelar muchos de los aspectos hardware de un sistema a un nivel suficiente para que un ingeniero software pueda especificar la plataforma sobre la que se ejecuta el software del sistema y para que un ingeniero de sistemas pueda manejar la frontera entre el hardware y el software.

Cuando se trata de la relación entre hardware y software se utilizan los diagramas de despliegue para razonar sobre la topología de procesadores y dispositivos sobre los que se ejecuta el software.

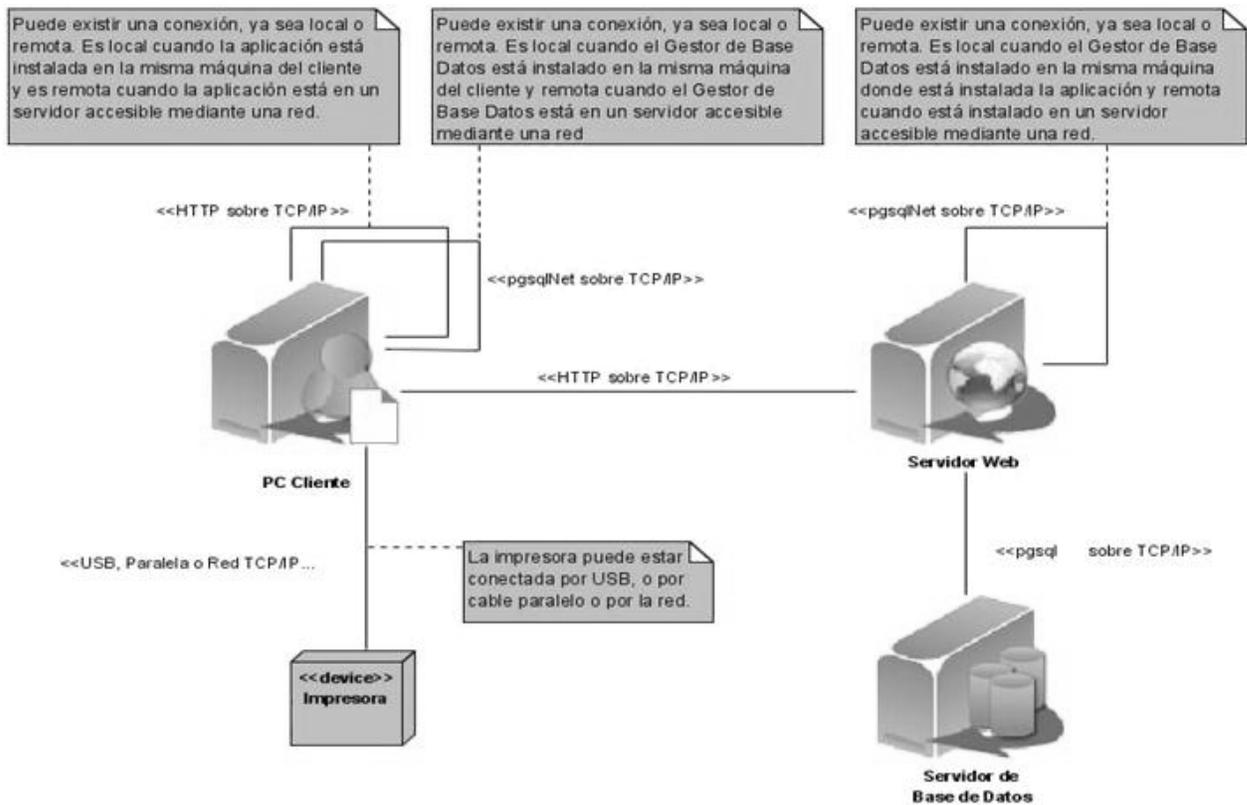


Fig. 59 Diagrama de Despliegue.

El sistema propuesto, muestra en su diagrama de despliegue una PC cliente con conexión local o remota hacia el servidor WEB. Esta PC cliente tendrá a su servicio un dispositivo para la impresión de los modelos generados por el sistema. El servidor WEB utilizado, tendrá conexión con el servidor de Base de Datos, y la misma podrá ser local o remota, en dependencia de donde se encuentre instalada la aplicación.

❖ Conclusiones:

A manera de resumen y conclusiones se define como el resultado principal de la implementación el Modelo de Implementación, el cual incluye los siguientes elementos:

- Subsistemas de implementación y sus dependencias, interfaces y contenidos.
- Componentes, incluyendo componentes fichero y ejecutables, y las dependencias entre ellos. Los componentes son sometidos a pruebas de unidad.
- La vista de arquitectura del modelo de implementación, incluyendo sus elementos arquitectónicamente significativos.

La implementación también produce como resultado un refinamiento de la vista de la arquitectura del modelo de despliegue, donde los componentes ejecutables son asignados a nodos.

El modelo de implementación es la entrada principal de las etapas de prueba que siguen a la implementación.

Más concretamente, durante la etapa de prueba cada construcción generada durante la implementación es sometida a pruebas de integración, y posiblemente también a pruebas de sistema.

Conclusiones:

En esta Era Virtual y del Conocimiento, las nuevas necesidades informativas de los usuarios deben ser satisfechas, tomando en cuenta las nuevas características que tiene la Contabilidad para cumplir con esas necesidades de los usuarios, se pueden destacar a continuación los principales factores sociales que influyen al desarrollo continuo e informatizado de la actividad contable a nivel mundial:

- El aumento progresivo del conocimiento y educación humanos.
- La eficiencia y eficacia dependientes en gran medida de la velocidad de procesamiento y obtención de la información.
- La globalización de las Tecnologías de la Informática y las Comunicaciones y su aplicación a todas las grandes e importantes empresas a nivel mundial.

El contador debe prepararse para conocer a plenitud las tecnologías de la Información emergente en esta Era del Conocimiento ya que la implementación de las Tecnologías de la Información no es simple, pero al ejecutarlas en los negocios debe sentarse en: los procesos y en la estructura y cultura organizacionales de la empresa. El contador de hoy en día tiene que prepararse tanto mental, intelectual como también en cuanto a los cambios mecánicos que surgen con el pasar del tiempo. Es importante reconocer también, que la implementación de sistemas contables esta cambiando el trabajo manual por las maquinas es por eso que cada uno de los que se encuentran ejerciendo la contabilidad deben preocuparse por conocer mas sobre los sistemas avanzados que facilitan estas actividades con el uso de la tecnología.

El implemento de sistemas informáticos es de gran importancia pues ayuda a una revelación eficiente de los estados financieros de cualquier empresa, brindando información valiosa a la hora de la toma de decisiones pues el objetivo primordial de dichos estados financieros es suministrar información acerca de la situación financiera como también el desempeño de la entidad y revelarlos o darlos a conocer a los usuarios de una forma eficiente y eficaz y que estos sean confiables y comprensibles para terceros.

El impacto de las TIC en la Contabilidad ha sido de gran importancia para las empresas debido a que ha venido a minimizar en gran manera tanto los costos, gastos como el elemento más importante para ella que es el tiempo.

Con la realización de este trabajo de diploma, se han cumplido todos los objetivos trazados dando terminación a todos y cada uno de los objetivos específicos que se han descrito en el mismo. Se ha desarrollado los flujos de trabajo completos correspondientes a Inicio y Elaboración, documentándose correctamente los artefactos generados en cada uno de ellos.

En el modelado del sistema que hemos propuesto para mejorar la actividad contable para las empresas presupuestadas de las FAR, hemos desarrollado los conocimientos relacionados con la investigación metodológica y la Ingeniería de Software, adquiridos durante este periodo de 5 años que duro la carrera de Ingeniería en Ciencias Informáticas. Se ha cumplido con la labor asignada por el proyecto productivo al que poerteneceemos y hemos contruido la documentación de uno de los modulos del ERP que para las Fuerzas Armadas Revolucionarias se esta desarrollando en la Universidad de las Ciencias Informáticas.

Recomendaciones:

Al dar por concluida la realización de este trabajo de diploma, y al considerarse cumplidos los objetivos trazados en el mismo, se recomienda la continuidad del mismo, procediendo esta vez a las fases de Construcción y Transición del modulo dedicado a la contabilidad del Sistema para la Gestión de la actividad Presupuestada de las FAR.

Se sugiere, que teniendo en cuenta el análisis y diseño realizado a los subsistemas de Recuperaciones, Inicio del Año Contable y Traspasos, se comience la implementación de los casos de uso correspondientes a cada uno, a fin de completar el trabajo iniciado en este proyecto.

Se espera que se tengan en cuenta la distribución lógica y física aplicada al sistema propuesto, así como los mecanismos y patrones de diseño empleados, para lograr, de esta forma, una concordancia clara y fiel entre lo documentado en este trabajo de diploma y lo que obtendrán los programadores una vez comenzada la implementación del sistema.

Se quiere la continua evaluación de los análisis aplicados a esta propuesta del sistema a fin de obtener, al final de cada iteración, la calidad que se espera en cada fase o flujo de trabajo por el que trascienda este proyecto.

BIBLIOGRAFIA

Consultada:

García Menéndez, Kristal. "El impacto que ha tenido la tecnología de información en las áreas de Contabilidad y Fiscal", 2007.

[Disponible en: <http://www.gestiopolis.com/canales2/finanzas/1/tecinfocontable.htm>]

Correa, Diana. "*Federación Internacional de Contadores Públicos (IFAC)*", 2007.

[Disponible en: <http://www.gestiopolis.com/Canales4/fin/ifac.htm>]

Serrano Cinca C. "*La Contabilidad en la Era del Conocimiento*", 2006.

[Disponible en: <http://ciberconta.unizar.es/leccion/INTRODUC/350.HTM>]

"*Top 10 Technology Initiatives*", 2007. [Disponible en:

<https://infotech.aicpa.org/Resources/Top+Technology+Initiatives/2007+Top+10+Technology+Initiatives/>]

Fernández Vilas, Ana. "*Diagrama de Despliegue*", 2007.

[Disponible en: <http://www-gris.det.uvigo.es/~avilas/UML/node50.html>]

Díaz Inchicaqui, Miguel. "*Impacto de las TIC en la carrera de contaduría*", 2002.

[Disponible en: <http://ciberia.ya.com/reynakaren/IMPACTO DE LAS TIC EN LA CARRERA DE CONTADURIA P.doc>]

"*Contabilidad*", 2007. [Disponible en: <http://es.wikipedia.org/wiki/Contabilidad>]

GLOSARIO DE TERMINOS

- Sistemas informatizados: conjunto de elementos o subsistemas, que mediante el uso de las TIC, aparecen interrelacionados entre sí tratando de alcanzar “ciertos objetivos”.
- Contabilidad: Ciencia del Patrimonio, que estudia la representación del patrimonio empresarial, tanto desde un punto de vista estático como dinámico.
- Contabilidad Financiera: técnica mediante la cual se recolectan, se clasifican, se registran, se sumarizan y se informa de las operaciones cuantificables en dinero, realizadas por una entidad económica.
- Estados Financieros o Contables: constituyen el producto final de la contabilidad. Básicamente se trata de cuadros y notas aclaratorias que resumen la situación económica y financiera de la empresa. Esta información resulta útil para gestores, reguladores y otros tipos de interesados como los accionistas, acreedores o propietarios.
- Cuentas: Instrumento de representación y medida de cada elemento patrimonial.
- Saldos: diferencia entre su Debe y su Haber. El saldo será deudor cuando el Debe es mayor que el Haber, en caso contrario se llamará saldo acreedor. El saldo es el valor residual que posee una cuenta, dependiendo del movimiento contable que tenga su saldo será deudor o acreedor.
- Libros contables:
 - ✓ El Diario. Donde se recogen, día a día, los hechos económicos. La anotación de un hecho económico en el libro Diario se llama "asiento". Registro de transacciones suscitadas del giro de la empresa.
 - ✓ El Mayor. En él se recogen todas las cuentas, con todos los cargos y abonos realizados en las mismas. Libro resumen del registro del movimiento de una cuenta específica.

- ✓ El Balance. En este libro se elaboran periódicamente los balances, que serían un resumen ordenado por masas patrimoniales de las distintas cuentas con sus saldos, así como los estados de "pérdidas y ganancias", donde se resumen y organizan las cuentas que presentan resultados de la empresa al objeto de hallar el posible beneficio o pérdida en cada uno de los ejercicios económicos.
- Finanzas: es la disciplina que estudia el flujo del dinero entre individuos, empresas o Estados. Las finanzas son una rama de la economía que estudia la obtención y gestión, por parte de una compañía, individuo o del Estado, de los fondos que necesita para sus objetivos y de los criterios con que dispone de sus activos. Estudia lo relativo a la obtención y gestión del dinero y de otros valores como títulos, bonos, etc.
- Crédito: obtención de recursos en el presente sin efectuar un pago inmediato, bajo la promesa de restituirlos en el futuro con las condiciones previamente establecidas. En el caso de un crédito bancario, es un contrato por el cual la entidad financiera pone a disposición del cliente una cantidad de dinero que el cliente deberá devolver con intereses y sumándole las comisiones pactadas.
- Deposito: contrato por el cual se entrega a una entidad financiera una cantidad de dinero para que lo guarde y custodie con obligación de restituirlo posteriormente, cumpliendo las condiciones pactadas.
- Acreedor: es aquella persona (física o jurídica) legítimamente facultada para exigir el pago o cumplimiento de una obligación contraída por dos partes con anterioridad. Es decir, que a pesar de que una de las partes se quede sin medios para cumplir con su obligación, ésta persiste. Por ejemplo, en caso de la quiebra de una empresa sus acreedores mantienen la facultad de requerir el cumplimiento de dicha obligación.
- Salario o remuneración: es el pago que recibe en forma periódica un trabajador de mano de su empleador a cambio del trabajo para el que fue contratado. El empleado recibe un salario a cambio de poner su trabajo a disposición del empleador, siendo éstas las obligaciones principales de su relación contractual.

El salario es el elemento monetario principal en la negociación de un contrato de trabajo. Es la contraprestación en la relación bilateral, aunque en algunas ocasiones se tienen también en cuenta otras condiciones laborales como vacaciones, jornada, etc.

- Propiedad: es el poder directo e inmediato sobre una cosa, por la que se atribuye a su titular la capacidad de disponer de ella, sin más limitaciones que las que imponga la ley.

- Año contable: es un periodo de 12 meses usado para calcular informes financieros anuales en negocios y otras organizaciones. En la mayoría de jurisdicciones hay leyes que regulan la contabilidad y requieren estos informes una vez cada doce meses. Este periodo normalmente empieza el 1 de enero y termina el 31 de diciembre del mismo año. Hay empresas que tienen ciertas características y formas de administración y su año fiscal dentro de la empresa empieza el 1 de julio y termina el 30 de junio del siguiente año.

- Órgano financiero: entidad sobre la que se ejecuta toda la actividad contable en cuestión.

- Órgano administrador de Crédito: entidad que se encarga de evaluar, distribuir y controlar el presupuesto de las unidades subordinadas que tiene bajo su mando, con fin de garantizar una utilización fiable y correcta del mismo.

- Actividad Presupuestada: Conjunto de acciones y movimientos contables que se realizan utilizando el presupuesto asignado al órgano financiero, por el órgano administrador de créditos correspondiente.

- Movimiento: Las transacciones comerciales que diariamente realiza una empresa deben estar respaldadas por un soporte que permite su registro en el comprobante diario y traslado a los libros principales y auxiliares de contabilidad. El registro del movimiento operativo de la empresa, de acuerdo con sus necesidades, se puede realizar en forma diaria, semanal, quincenal o mensual, en comprobante diario y libros. Al final se elabora el Balance de prueba.

- Movimiento Contable: Conjunto de acciones relacionadas con la economía, llevadas a cabo en un órgano financiero y que pueden ser registradas y documentadas en los modelos correspondientes a dicho movimiento.

- Presupuesto: es la previsión de gastos e ingresos para un determinado lapso, por lo general un año. Permite a las empresas, los gobiernos, las organizaciones privadas y las familias establecer prioridades y evaluar la consecución de sus objetivos. Para alcanzar estos fines puede ser necesario incurrir en déficit (que los gastos superen a los ingresos) o, por el contrario, puede ser posible ahorrar, en cuyo caso el presupuesto presentará un superávit (los ingresos superan a los gastos).

- Comprobante: Documento que registra los movimientos contables realizados en el órgano financieros así como las cuentas afectadas por los mismos.

- Apertura: Al constituirse la empresa, se inicia el proceso contable con los aportes del dueño o socios registrados en el inventario general inicial. Con el inventario general inicial se elaboran en el Balance General inicial y el comprobante diario de contabilidad, denominado comprobante de apertura que sirve para abrir los libros principales y auxiliares.