

005.12  
Rey  
S  
TD-0172-06

TD-0172-06

**INSTITUTO SUPERIOR POLITÉCNICO "JOSÉ ANTONIO ECHEVERRÍA"  
FACULTAD DE INGENIERÍA INDUSTRIAL  
CENTRO DE ESTUDIOS DE INGENIERÍA DE SISTEMAS (CEIS)**



**SISTEMA DE CONTABILIDAD FINANCIERA PARA LA ACTIVIDAD  
PRESUPUESTADA EN LAS FAR.  
MODULO CONTABILIDAD.**



**TRABAJO PARA OPTAR POR EL TITULO DE INGENIERO EN  
INFORMATICA.**

**AUTOR**

Zénel Reyes Pérez

**TUTOR**

Ing. Yuneí López Lugo

**CONSULTANTES**

Ing. Igris Sevajanes Milian

Ing. Violena Hernández Aguilar

**CIUDAD DE LA HABANA  
JUNIO, 2006**

## RESUMEN

El manual de contabilidad financiera para la actividad presupuestada de las FAR, es el medio por el que se rigen los órganos financieros para registrar y controlar de todas sus actividades económicas, en la cual intervienen los órganos consumidores autorizados, los órganos administradores de primer nivel, de segundo nivel, el administrador principal y las entidades que ejecutan los gastos.

El proceso de registro y control de los hechos económicos, de forma general, comienza cuando un órgano consumidor solicita a través de un documento contable realizar alguna operación para una especialidad, que puede ser una asignación de presupuesto u otro tipo, convirtiéndose en un hecho económico que debe ser registrado para darle uso posteriormente; este proceso se realiza en cada uno de los niveles antes mencionados para varias entidades, acumulándose varios documentos a registrar y para lo cual se realizan un cúmulo de operaciones diarias, en su mayoría de forma manual, sujetas a datos erróneos y haciéndose más difícil en los órganos que manejan grandes volúmenes de información.

La automatización de los procesos descritos anteriormente en un sistema posibilita integrar toda la información, mejorando el registro y control de los hechos económicos entre los diferentes órganos financieros que intervienen en la contabilidad financiera para la actividad presupuestada en las FAR.

Los resultados más relevantes lo constituyen: la existencia de un producto informático para la contabilidad financiera en el MINFAR, con facilidades de migración hacia otra plataforma y un flujo más rápido y seguro de la información.

<b>INTRODUCCIÓN.....</b>	<b>1</b>
<b>Capítulo 1 FUNDAMENTACIÓN TEÓRICA.....</b>	<b>6</b>
1.1 Introducción .....	6
1.2 Contabilidad.....	6
1.3 Contabilidad Financiera.....	7
1.4 Objeto de estudio.....	7
1.4.1 Flujo actual del proceso.....	8
1.5 Sistemas automatizados vinculados al campo de acción.....	9
1.6 Tendencias y tecnologías actuales utilizadas.....	11
1.6.1 ¿Qué es Internet?.....	12
1.6.2 La información a través de Internet. La Web.....	12
1.6.3 Aplicaciones Web.....	13
1.6.4 Lenguajes de programación para la Web.....	14
1.6.5 Fundamentación de la selección del Gestor de Base de Datos.....	17
1.6.6 Fundamentación de la metodología a utilizar.....	21
1.6.7 Herramientas CASE.....	23
1.6.8 Otras herramientas necesarias.....	24
1.7 Propuesta Solución.....	25
1.8 Conclusiones.....	26
<b>Capítulo 2 DESCRIPCIÓN DE LA PROPUESTA SOLUCIÓN .....</b>	<b>28</b>
2.1 Introducción.....	28
2.2 Modelado del negocio.....	28
2.3 Diagrama de caso de uso del negocio.....	30
2.4 Descripción del caso de uso del negocio.....	30
2.5 Diagrama de actividades del caso de uso del negocio.....	32
2.6 Modelo de Objetos.....	34
2.7 Requerimientos funcionales.....	34
2.8 Requerimientos no funcionales.....	37
2.9 Descripción del sistema propuesto.....	39
2.10 Modelo del caso de uso del sistema.....	40
2.11 Expansión de los casos de uso.....	45
2.12 Conclusiones.....	56
<b>Capítulo 3 CONSTRUCCIÓN DE LA PROPUESTA SOLUCIÓN .....</b>	<b>58</b>
3.1 Introducción.....	58
3.2 Análisis y Diseño.....	58
3.3 Mecanismos de Diseño.....	58
3.3.1 Mecanismo de diseño de Seguridad.....	59
3.3.2 Mecanismo de diseño de Acceso a Datos.....	60
3.4 Diagrama de clases del diseño.....	62

## Índice

---

3.5 Diseño de la Base de Datos.....	66
3.6 Principios de Diseño.....	70
3.6.1 Estándares de diseño.....	71
3.6.2 Estándares de codificación.....	71
3.6.3 Estándares para la BD.....	74
3.6.4 Tratamiento de Errores.....	76
3.7 Modelo de despliegue.....	76
3.8 Modelo de implementación.....	78
3.9 Conclusiones.....	80
<b>Capítulo 4 ESTUDIO DE FACTIBILIDAD .....</b>	<b>83</b>
4.1 Introducción.....	83
4.2 Planificación.....	83
4.3 Estimación de esfuerzo y costos.....	84
4.4 Beneficios tangibles e intangibles.....	89
4.5 Análisis de costos y beneficios.....	90
4.6 Análisis de Sostenibilidad.....	90
4.7 Conclusiones.....	91
<b>CONCLUSIONES.....</b>	<b>92</b>
<b>RECOMENDACIONES .....</b>	<b>94</b>
<b>BIBLIOGRAFÍA.....</b>	<b>96</b>
<b>GLOSARIO DE TÉRMINOS .....</b>	<b>100</b>
<b>ANEXOS .....</b>	<b>103</b>
Anexo I.....	103
Los tipos de Operación son los siguientes.....	103
Anexo II.....	104
Anexo III.....	107
Anexo IV.....	108
Anexo V.....	109

### INTRODUCCIÓN

La contabilidad es una técnica que se ocupa de registrar, clasificar y resumir las operaciones de un negocio con el fin de interpretar sus resultados. Su origen data de hace muchos años, desde que el hombre se dio cuenta de que su memoria no era suficiente para guardar la información necesaria.

Desde el año 6000 a.c. ya existían los elementos necesarios para la actividad contable. Los primeros habitantes sobre la tierra necesitaban dejar constancia de determinados hechos, que se producían con demasiada frecuencia y eran muy complejos para ser memorizados.

Así, la contabilidad trascendió entre los siglos y cada civilización creó su propia forma de contabilizar en dependencia de sus necesidades hasta llegar a nuestros días, donde las empresas grandes o pequeñas poseen un sistema de contabilidad para registrar, clasificar y resumir todas sus operaciones económicas.

El hombre, en aras de simplificar todas sus actividades y con el devenido avance en el campo de la ciencia y técnica, principalmente en la Computación, comenzó a diseñar aplicaciones para agilizar el registro de los hechos económicos que sucedían en las empresas.

Cuba, es un país socialista, mas no por ello deja de utilizar la contabilidad como en otros países, aunque enfocada desde un punto de vista diferente; el Ministerio de la Fuerzas Armadas Revolucionarias (MINFAR), institución encargada de la defensa de nuestras conquistas es también una empresa y posee un Sistema de Contabilidad Financiera con particularidades.

El MINFAR en su estructura organizativa cuenta con varios niveles: un administrador principal, administradores de primer y segundo nivel, los órganos consumidores y las especialidades.

El ciclo contable del MINFAR comienza con el registro de la documentación primaria, presentada por un órgano consumidor, elaborándose un comprobante de operaciones, que asocia varios movimientos contables quedando registrados los hechos económicos ocurridos para una especialidad; existen órganos que

realizan dicho registro de forma manual, manipulando un volumen de información, que aumenta en dependencia del nivel, y que está sujeta a cálculos erróneos. En otros órganos financieros, el registro se realiza a través de tablas en Excel, pero esto no imposibilita que se comentan los mismos errores contables mencionados anteriormente; solo el administrador principal o MINFAR, cuenta con un sistema informático implementado en Visual FoxPro, el mismo podría utilizarse en cada uno de los órganos financieros para contabilizar sus hechos económicos, pero no es posible porque no es multiusuario, no tiene tecnología multiplataforma y no está certificado por las entidades correspondientes. En el país existen sistemas que permiten cumplir con las características antes mencionadas y trabajan la contabilidad financiera pero ninguno se adapta a la forma en que las FAR la realiza.

La presente investigación surge como necesidad de dar solución a las situaciones antes expuestas, por lo que el **problema científico** se describe como: El sistema existente, no abarca todos los procesos relacionados con el registro y control de los hechos económicos de la actividad presupuestada en las FAR y no responde a las nuevas concepciones de la institución.

Este problema se enmarca en el **objeto de estudio**: Contabilidad Financiera para la actividad presupuestada en las FAR, para ponerlo a disposición de los jefes, especialistas y técnicos que laboran en dicha actividad. Para resolver el problema planteado se propone el siguiente **objetivo general**: Diseñar e implementar una Aplicación Web que permita registrar y controlar los hechos económicos de un órgano financiero, en correspondencia con los requerimientos del usuario.

De acuerdo con la propuesta anterior se proponen los siguientes **objetivos específicos**:

- ✓ Modelar el negocio.
- ✓ Modelar los casos de uso del sistema del módulo correspondiente.
- ✓ Realizar el modelo de análisis y diseño.

- ✓ Iniciar la implementación de una aplicación Web para la automatización de los procesos de registro y control de los hechos económicos de un órgano financiero.

El objetivo delimita el **campo de acción**, que es, el proceso de automatización de la contabilidad para la actividad presupuestada de las FAR y el registro y control de los hechos económicos que ocurren en un órgano financiero.

Para guiar la investigación se plantea la siguiente **hipótesis**: El desarrollo de una aplicación Web, consistente y multiplataforma facilitará la automatización del proceso de registro y control de los hechos económicos de un órgano financiero para la Actividad Presupuestada en las FAR, que cumpla con los requisitos de los usuarios y que esté en correspondencia con las nuevas concepciones de automatización en las FAR.

Para cumplir con estos objetivos y resolver la situación problemática planteada, se proponen las siguientes **tareas**:

- ✓ Estudio del proceso contable en las FAR, para la actividad presupuestada.
- ✓ Estudio del proceso de registro de los hechos económicos para un órgano financiero.
- ✓ Selección de la metodología de Análisis y Diseño de sistemas informáticos, que facilite y garantice la creación con calidad del sistema.
- ✓ Selección de las herramientas para llevar a cabo el proyecto y la elección de la plataforma en la que se desarrollará la aplicación. Fundamentando su elección.
- ✓ Diseño de una base de datos que soporte las funcionalidades del sistema.
- ✓ Implementación e implantación de una aplicación Web que permita el registro y control de los hechos económicos de un órgano financiero para la Actividad Presupuestada en la FAR.

## Introducción

---

El presente documento consta de cuatro capítulos:

En el capítulo 1, se describen los aspectos fundamentales de la Contabilidad, las características de esta, los principales conceptos asociados al problema. Se identifican los principales problemas que fundamentan la propuesta de solución, y se marcan los objetivos generales y específicos.

En el capítulo 2, se describen las características de la propuesta solución a través de un modelo del negocio, y se hace el análisis del sistema a desarrollar. Se definen las funcionalidades del sistema y se describen detalladamente, utilizando herramientas de modelación.

En el capítulo 3, se enfoca la construcción de la solución mediante diagramas de clases y se plantean los principios para el diseño y la implementación. Aquí se construyen las funcionalidades que se definieron en el capítulo anterior.

En el capítulo 4 se realiza un estudio de factibilidad sobre el sistema, obteniendo los beneficios tangibles e intangibles y analizando los costos del desarrollo de esta propuesta.



# Capítulo 1

Introducción

Contabilidad

Contabilidad Financiera

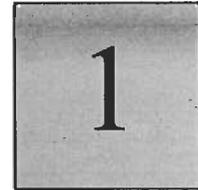
Objeto de Estudio

Sistemas Automatizados

Tendencias y Tecnologías

Propuesta Solución

Conclusiones



## CAPÍTULO FUNDAMENTACIÓN TEÓRICA

### 1.1 Introducción

En el presente capítulo se brindan los aspectos relacionados con la Contabilidad, sus características, y una descripción de los principales conceptos asociados al problema y que son necesarios para entender el negocio y la propuesta de solución.

Además se describen profundamente los procesos del negocio que se relacionan con el objeto de estudio de este trabajo. Se identifican los principales problemas que fundamentan la propuesta de solución, y se marcan los objetivos generales y específicos.

### 1.2 Contabilidad

Existen diversos criterios respecto a la contabilidad; algunos autores la definen como una ciencia y otros la suponen como una técnica.

De acuerdo a la definición de ciencia: *Conocimiento cierto de las cosas por sus principios y causas*, la Contabilidad es una ciencia porque es un conocimiento cierto y adquirido y no es una suposición de hechos sin relevancia alguna. Es el análisis de cada hecho y en cada uno se aplica un conocimiento adquirido y no supuesto, ni abstracto.

Sin embargo, al observar la definición de técnica: *Conjunto de procedimientos de que se sirve una ciencia o un arte, es la forma o artificio de que nos valemos para realizar una tarea* y en la Contabilidad la tarea es el registro, la teneduría de libros, y en esta, el análisis de los hechos contables, constituye la tarea científica.

Independientemente de estas definiciones y sin entrar en contradicción con ninguna de estas, existen conceptos, diferentes también, sobre que es la Contabilidad y el resultado indica que es el proceso sistemático de registrar, medir, clasificar, resumir e interpretar, en términos monetarios, las actividades económicas de cualquier empresa.

### 1.3 Contabilidad Financiera.

La contabilidad financiera es definida como la rama de la contabilidad, centrada en la preparación y presentación de un juego de estados contables con el propósito de suministrar información, que es útil en la toma de decisiones de naturaleza económica por parte de la mayoría de los usuarios, especialmente los externos (propietarios, proveedores, instituciones financieras, inversionistas, etc.). Si analizamos la contabilidad desde distintos puntos de vista, podemos establecer la que se prepara antes de que ocurran los hechos y que por tanto, recibe el nombre de contabilidad presupuestaria o predictiva, y la actividad económica realizada dentro de esta recibe el nombre de *actividad presupuestada* y se basa en hechos supuestos, que se analizan en base a presupuestos, realizados con técnicas y conocimientos científicos aplicados y que sirven como guía y pauta para alcanzar metas fijadas que en caso de no ser alcanzadas se realizan los análisis para conocer los desvíos y las causas que lo provocaron.

### 1.4 Objeto de estudio.

El Ministerio de las Fuerzas Armadas Revolucionarias (MINFAR), es la principal institución cubana encargada de defender las conquistas de la Revolución. Funciona económicamente como una empresa más en el país con características específicas que la diferencia del resto de las entidades. La actividad contable en el MINFAR se divide en dos: la actividad para las **empresas presupuestadas** (Unidades Militares, Hospitales y Escuelas), que

son generadoras de gastos y la actividad para las empresas autofinanciadas (Gaviota, TRD, GeoCuba, Talleres de Auto, UIM, UAM, etc.).

El registro de la contabilidad, en el ministerio, se divide en: contabilidad material y **Contabilidad Financiera**, para tener un mejor control, siendo esta última la que ocupará el objeto de estudio de este trabajo, definida específicamente para los órganos contables de las FAR.

#### 1.4.1 Flujo actual del proceso.

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.

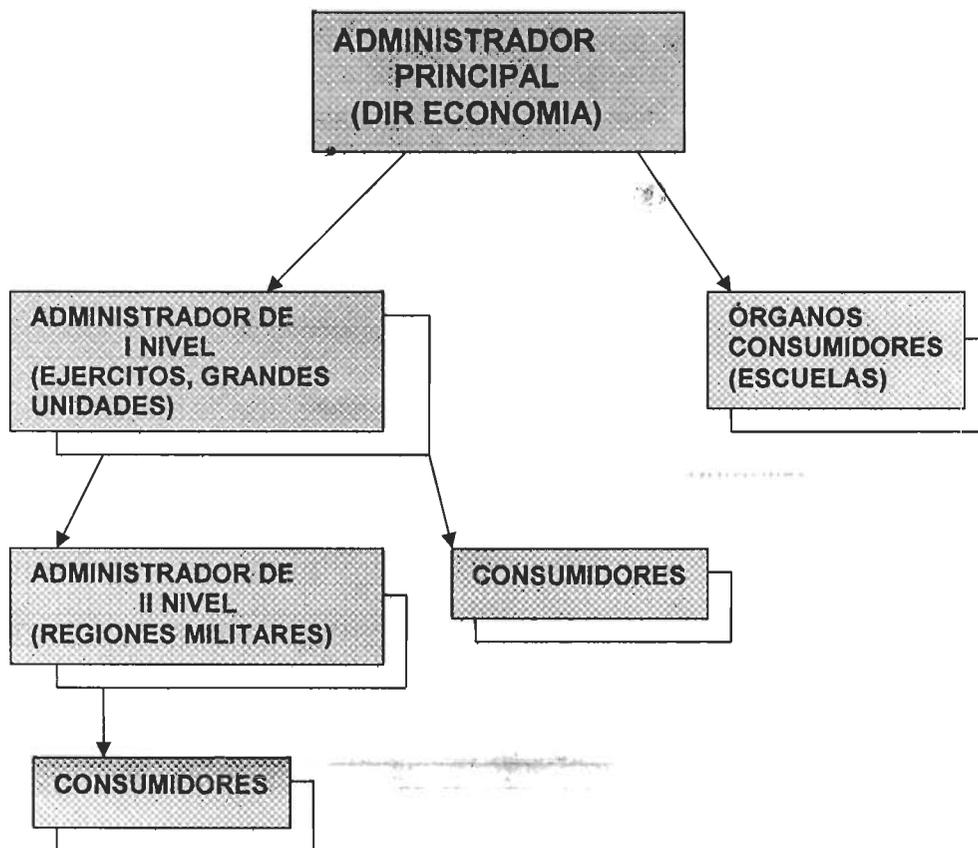


Figura 1. 1 Estructura organizativa y dependencias externas del MINFAR.

Cada uno de los órganos subordinados registra sus actividades por medio de comprobantes de operaciones que son llevados a los libros de registros (Mayor y Submayor).

El registro de los hechos económicos que ocurren en los órganos financieros se realiza de forma manual, y solo el ministerio, por el gran cúmulo de información que maneja, emplea un software implementado en VisualFoxPro, que presenta dificultades tales como:

- ✓ No está concebido para ser multiplataforma.
- ✓ No permite el trabajo de múltiples usuarios.
- ✓ 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 está acorde con los estándares establecidos lo que provoca que no esté certificado.

### 1.5 Sistemas automatizados vinculados al campo de acción.

Es amplia la gama de sistemas existentes en el mundo que registran y controlan los hechos económicos de una empresa y permiten con el uso de la información contenida tomar o no, una decisión; ellos son:

#### **ISIS.**

**Sistema de Contabilidad Clásica:** El tradicional sistema contable para carga de asientos, y emisión de diario, mayores y balances. Sus características destacadas son:

- ✓ Veloz pantalla de carga, con manejo vía teclado o Mouse, de acuerdo a lo que prefiera el usuario.
- ✓ Velocidad al manejar grandes volúmenes de datos / grabación de asientos / ordenamiento de las bases: gracias a su motor SQL Server, el

sistema contable maneja grandes cantidades de información en mínimo tiempo. Mucho más veloz que cualquier otro sistema contable de la competencia.

- ✓ Multiempresa, ideal para estudios contables que contabilizan a numerosos clientes.
- ✓ Bimonetaria.
- ✓ Exposición de sus saldos. Esto es particularmente útil cuando: Se desean emitir balances acordes a los requerimientos contables de una casa matriz, quizás extranjera, que se maneja con normas distintas a las nacionales. Se tiene un plan de cuentas con demasiados rubros y se desea mostrar saldos de manera resumida (por ejemplo, agrupar sueldos administración, sueldos comercialización, sueldos sucursal en una única cuenta de sueldos). Se desea tener una visión global de la empresa, cuando la contabilidad está abierta por proyectos.
- ✓ Asientos automáticos de apertura, cierre y resultados.
- ✓ Asientos modelo. Asientos resúmenes de ventas, compras, pagos y cobranzas.
- ✓ Emisión de informes contables clásicos (diario, mayores, balances) e informes adicionales.
- ✓ Previsualización de informes antes de impresión. Exportación de informes a utilitarios (Excel, Word, etc.).

### **VersatSoft.**

Es el sistema integral para automatizar la gestión económica - contable y financiera de cualquier empresa u organización.

Sus características más destacadas son:

- ✓ Registrar los hechos económicos a partir de los documentos primarios que le dieron origen.

- ✓ Trabaja con volúmenes e importes en las cuentas de costos, procesos, ingresos e inventarios.
- ✓ Emite un grupo de reportes predeterminados así como, una variedad de resultados configurables.
- ✓ Protección y seguridad en los datos, en operación y de forma física en las bases de datos.

El resultado de un estudio detallado de cada uno de estos sistemas arrojó la conclusión de que ninguno era capaz de darle solución al problema planteado porque no manejan la contabilidad como lo hacen en el MINFAR, ejemplo las cuentas, para la actividad presupuestada en FAR, se dividen en cuentas de patrimonio y de orden, siendo solo cuentas de orden para las otras entidades; y la Contabilidad Material tiene un análisis independiente de la Financiera.

### 1.6 Tendencias y tecnologías actuales utilizadas.

La revolución de las Tecnologías de la Información y las Comunicaciones (TIC), con la incorporación de la computadora a los medios electrónicos, los sistemas de comunicación por satélite, el teléfono, el fax y el celular, no acaban de asombrarnos. En el presente siglo otras novedades de comunicación e información se desarrollan y tendrán aplicación social. Se anuncian ya las redes de telecomunicación multimedia, que darán lugar al cambio más grande de todos los tiempos.

Entre las TIC tenemos: la realidad virtual, que puede catalogarse como la multimedia interactiva en su máxima expresión; la formación de redes que pudiéramos nombrarla como la tendencia fundamental de las nuevas tecnologías y donde se destaca la red de redes Internet. El número de computadoras que se venden cada año en todo el mundo es creciente por lo que el mercado en general de las TIC apunta a un crecimiento vertiginoso en el uso de las nuevas tecnologías.

La incorporación de las computadoras a la velocidad de los negocios y la posibilidad de que en segundos cualquier información de la vuelta al mundo varias veces obligan a un constante movimiento en la información para que sea válida, ya que cuando no está actualizada no sirve para la toma de decisiones.

El desarrollo de las TIC, ha llamado la atención de todas las empresas necesitadas de sistemas de procesamiento de información para acelerar sus procesos y elevar la calidad de los mismos.

### 1.6.1 ¿Qué es Internet?

El fenómeno social, cultural, sociológico y comercial de Internet tiene su origen en la década del 60 y se relaciona con un proyecto de defensa financiado por el gobierno de Estados Unidos. Gracias a esta iniciativa, hoy es posible buscar, crear y transferir información en tiempo real para 6 mil millones de personas.

Desde sus inicios, el crecimiento de Internet ha sido fenomenal, especialmente en la década del 90, época en que la red se convirtió en una herramienta fundamental de comunicación, información e integración, que permite a los usuarios ahorrar tiempo y dinero, además de tener a su alcance todos los productos y servicios que requieran sin fronteras de espacio o tiempo.

### 1.6.2 La información a través de Internet. La Web.

La Web, es el universo de información accesible a través de Internet. Es un sistema de información global, interactivo, dinámico, distribuido, gráfico, basado en Hipertexto<sup>1</sup>, con plataforma de enlaces cruzados, que se ejecuta en Internet.

El componente más usado en el Internet es definitivamente la Web. Su característica sobresaliente es el texto remarcado, un método para referencias

---

<sup>1</sup> Es una tecnología que organiza una base de información en bloques de contenidos, conectados a través de enlaces cuya activación o selección provoca la recuperación de información.

cruzadas instantáneas. Usando la Web, se tiene acceso a millones de páginas de información. La exploración se realiza por medio de un software especial denominado "Browser" o Explorador. El protocolo que se utiliza para la comunicación en la Web es el HTTP (Hypertext Transfer Protocol) y el formato que se utiliza para la transferencia es el HTML (Hypertext Markup Language).

### 1.6.3 Aplicaciones Web.

Las aplicaciones Web se desarrollan como una extensión de los Sistemas Web para agregar funcionalidad de negocio al proceso. En términos más simples, una aplicación Web es un Sistema Web que permite a los usuarios ejecutar lógica de negocio a través de un Navegador (Browser), o lo que es lo mismo: modificar el estado del negocio.

Las aplicaciones Web utilizan las tecnologías existentes para generar contenidos dinámicos y permitir a los usuarios del sistema modificar la lógica del negocio en el servidor. Si no existe lógica de negocios en el servidor, el sistema no puede ser considerado una aplicación Web, este es el caso de un sitio Web.

La arquitectura de un Sitio Web es simple. Este contiene los componentes principales: el Servidor Web, una Red y un navegador o cliente. La arquitectura de una aplicación Web además incluye la aplicación en el servidor, que es la que permite al sistema manejar la lógica de negocio y tener un estado.

Las aplicaciones Web en sus inicios mostraban información de forma estática, esto implicaba que muchos sitios debían actualizarse manualmente, lo que provocaba que fuera muy costosa la actualización y entonces muchos estuvieran desactualizados e incluso algunos dejaron de existir por este problema. Pero esta situación cambió de estado, las aplicaciones dinámicas comenzaron a prevalecer sobre las estáticas y son hoy en día las que dominan Internet.

Ejemplo tenemos; sitios de noticia, investigación, publicaciones. Todo este dinamismo se logra mediante mecanismos de gestión de información, mediante los cuales intercambiamos con un gran “almacén de datos”.

### **1.6.4 Lenguajes de programación para la Web.**

Uno de los ejes fundamentales que diferencian a Internet de otros medios de comunicación es la interacción y personalización de la información con el usuario. Esto se logra por medio de algunos de los diferentes lenguajes de programación para Web que existen hoy en día. Dichos lenguajes se clasifican en dos partes fundamentales que reconocen la propia arquitectura Cliente/Servidor de esta plataforma de desarrollo: los lenguajes del lado del servidor y los lenguajes del lado del cliente.

Entre los lenguajes del lado del servidor podemos encontrar entre los más sobresalientes por el auge que estos han tenido, PERL, ASP, PHP, Java, JSP. Estos se caracterizan por desarrollar la lógica del negocio dentro del servidor, además de ser los encargados del acceso a Bases de Datos, tratamiento de la información, etc.

Del lado del cliente se encuentran principalmente el JavaScript (JScript) y el Visual Basic Script (VBScript), que son los encargados de aportar dinamismo a la aplicación en los navegadores. En el caso del VBScript este es prácticamente usado a la hora de programar en ASP del lado del servidor, ya que su mayor desventaja radica en que solo es soportado por el navegador Web de su fabricante, Microsoft. Por otro lado JScript es soportado por la mayoría de los navegadores existente actualmente.

Esta distinción en los lenguajes ha sido necesaria debido a que la Web funciona en modo “desconectado”, o sea, un usuario a través de un navegador hace una petición de una página Web a un servidor Web (Request), el servidor recepciona la petición, la procesa y le envía la respuesta al cliente (Response), este la recepciona y se desconecta.

### **Perl.**

Es un lenguaje de programación muy utilizado para construir aplicaciones CGI (Computer Graphic Interface) para la Web. Perl viene a indicar que se trata de un lenguaje de programación muy práctico para extraer información de archivos de texto y generar informes a partir del contenido de los ficheros.

Es un lenguaje libre de uso, eso quiere decir que es gratuito. Antes estaba muy asociado a la plataforma Unix, pero en la actualidad está disponible en otros sistemas operativos como Windows.

Perl está inspirado a partir de algunos lenguajes provenientes de los sistemas Unix, pero está enfocado a ser más práctico y fácil que estos últimos. Una diferencia fundamental de Perl con respecto a los otros lenguajes es que no limita el tamaño de los datos con los que trabaja, el límite lo pone la memoria que en ese momento se encuentre disponible.

### **Personal Home Page (PHP).**

Es un lenguaje de programación del lado del servidor gratuito e independiente de plataforma, rápido, con una gran librería de funciones y mucha documentación, interpretado y embebido en el HTML, creado en 1994 por Rasmus Lerdorf, pero al ser desarrollado con política de código abierto, a lo largo de su historia ha tenido muchas contribuciones de otros desarrolladores.

El PHP "corre" en 7 plataformas, funciona en 11 tipos de servidores, ofrece soporte sobre unas 20 Bases de Datos y contiene unas 40 extensiones estables sin contar las que se están experimentando, además de que:

- ✓ Es un software libre y de código abierto, lo que implica menos costes y servidores más baratos que otras alternativas.
- ✓ Es muy rápido. Su integración con los gestores de Base Datos y el servidor Apache, le permite constituirse como una de las alternativas más atractivas del mercado.

- ✓ Su sintaxis está inspirada en C, ligeramente modificada para adaptarlo al entorno en el que trabaja, de modo que si se está familiarizado con esta sintaxis, resultará muy fácil aprender PHP.
- ✓ PHP tiene una de las comunidades más grandes en Internet, por lo que no es complicado encontrar ayuda, documentación, artículos, noticias, y más recursos.
- ✓ Posee una potente variedad de extensiones para el acceso a la mayoría de los sistemas de gestión de bases de datos, por lo que una migración a otro sistema de gestión es mucho menos costosa que en otras plataformas.

### **Active Server Pages (ASP)**

ASP es la tecnología desarrollada por Microsoft para la creación de páginas dinámicas del servidor. ASP se escribe en la misma página Web, utilizando el lenguaje Visual Basic Script o JScript (JavaScript de Microsoft).

Actualmente se ha presentado ya la segunda versión de ASP, el ASP.NET, que comprende algunas mejoras en cuanto a posibilidades del lenguaje y rapidez con la que funciona.

ASP.NET tiene algunas diferencias en cuanto a sintaxis con el ASP, de modo que se ha de tratar de distintas maneras uno de otro.

### **Java Server Pages (JSP).**

JSP en castellano significa algo como Página de Servidor Java. Es una tecnología orientada a crear páginas Web con programación en Java.

Con JSP podemos crear aplicaciones Web que se ejecutan en variados servidores Web, de múltiples plataformas, ya que Java es en esencia un lenguaje multiplataforma. Las páginas JSP están compuestas de código HTML/XML mezclado con etiquetas especiales para programar Scripts de servidor en sintaxis Java. Por tanto, las JSP podremos escribirlas con nuestro editor HTML/XML habitual.

El motor de las páginas JSP está basado en los servlets de Java – programas en Java destinados a ejecutarse en el servidor –, aunque el número de desarrolladores que pueden afrontar la programación de JSP es mucho mayor, dado que resulta mucho más sencillo de aprender que los servlets.

### **Fundamentación de la selección del lenguaje a utilizar.**

Hasta el momento se han analizado las características fundamentales de los lenguajes de programación candidatos para la implementación de la propuesta de este trabajo, para fundamentar nuestra elección haremos una comparación teniendo en cuenta algunas características que influyen directamente en el ambiente de trabajo donde se va a desarrollar la propuesta. En cuanto a:

- ✓ Características multiplataformas: Menos el ASP, que es solamente soportado por la plataforma Windows, los demás lenguajes están soportados en múltiples plataformas.
- ✓ Velocidad de ejecución: la velocidad es mayor en PHP, seguido por JSP.
- ✓ Disponibilidad de recursos: actualmente el más utilizado en la Internet es el PHP, siendo más utilizado en la publicación de artículos y códigos de ejemplos. PHP tiene una de las comunidades más grandes en Internet.

De acuerdo a estas comparaciones, y al observar las características particulares con cada lenguaje, el PHP resulta mucho más favorecido, por tanto pensamos que es el adecuado para implementar la propuesta de sistema de este trabajo.

### **1.6.5 Fundamentación de la selección del Gestor de Base de Datos.**

Un Sistema de Gestión de Bases de Datos (SGBD) consiste en un conjunto de programas, procedimientos y lenguajes que nos proporcionan las herramientas necesarias para trabajar con una base de datos. Incorporar una

serie de funciones que nos permita definir los registros, sus campos, sus relaciones, insertar, suprimir, modificar y consultar los datos.

Para comunicarse con un SGBD, tanto para definir datos y estructuras como para hacer consultas sobre los datos se puede utilizar SQL (Structured Query Language), que no es más que un lenguaje de consultas estructurado compuesto por comandos, cláusulas, operadores y funciones de agregado.

Estos elementos se combinan en grupos de instrucciones (Consultas) para actualizar y manipular las bases de datos.

Su forma más común de utilización es a través de un lenguaje de programación anfitrión como, por ejemplo, los lenguajes de programación en la Web del lado del Servidor.

Entre los SGBD comúnmente utilizados en el mundo tenemos Oracle, MySQL, Microsoft SQL Server, PostgreSQL, InterBase, entre otros. Todos estos presentan un enfoque relacional con un buen basamento matemático centrado en el Álgebra Relacional.

Luego de un estudio detallado de los SGBD antes mencionados, hemos seleccionado el **PostgreSQL** como herramienta para almacenar y gestionar los datos de nuestro sistema.

*¿Qué características tiene PostgreSQL que lo hace nuestra elección?*

PostgreSQL ofrece muchas ventajas para su compañía o negocio, respecto a otros sistemas de bases de datos, en cuanto a:

### 1- Instalación ilimitada

Es frecuente que las bases de datos comerciales sean instaladas en más servidores de lo que permite la licencia. Algunos proveedores comerciales consideran a esto la principal fuente de incumplimiento de licencia. Con PostgreSQL, nadie puede demandarlo por violar acuerdos de licencia, puesto que no hay costo asociado a la licencia del software.

Esto trae consigo ventajas adicionales como:

- ✓ Modelos de negocios más rentables con instalaciones a gran escala.
- ✓ No existe la posibilidad de ser auditado para verificar cumplimiento de licencia en ningún momento.
- ✓ Flexibilidad para hacer investigación y desarrollo sin necesidad de incurrir en costos adicionales de licenciamiento.

### 2- Ahorros considerables en costos de operación

Nuestro software ha sido diseñado y creado para tener un mantenimiento y ajuste mucho menor que los productos de los proveedores comerciales, conservando todas las características, estabilidad y rendimiento.

Además de esto, nuestros programas de entrenamiento son reconocidamente mucho más costo-efectivos, manejables y prácticos en el mundo real que aquellos de los principales proveedores comerciales.

### 3- Estabilidad y confiabilidad legendarias

En contraste a muchos sistemas de bases de datos comerciales, es extremadamente común que compañías reporten que PostgreSQL nunca ha presentado caídas en varios años de operación de alta actividad.

### 4- Extensible

El código fuente está disponible para todos sin costo. Si su equipo necesita extender o personalizar PostgreSQL de alguna manera, pueden hacerlo con un mínimo esfuerzo, sin costos adicionales. Esto es complementado por la comunidad de profesionales y entusiastas de PostgreSQL alrededor del mundo que también extienden PostgreSQL todos los días.

### 5- Multiplataforma

PostgreSQL está disponible en casi cualquier Unix (34 plataformas en la última versión estable), y una versión nativa de Windows está actualmente en estado beta de pruebas.

### 6- Diseñado para ambientes de alto volumen

PostgreSQL usa una estrategia de almacenamiento de filas llamada MVCC para conseguir una mejor respuesta en ambientes de grandes volúmenes. Los principales proveedores de sistemas de bases de datos comerciales usan también esta tecnología, por las mismas razones.

### 7- Herramientas gráficas de diseño y administración de bases de datos

Existen varias herramientas gráficas de alta calidad para administrar las bases de datos (pgAdmin, pgAccess) y para hacer diseño de bases de datos (Tora, Data Architect).

Además PostgreSQL ofrece una serie de características técnicas, al igual que otros gestores, que permiten un mejor trabajo con las bases de datos; entre estas podemos encontrar:

- ✓ Replicación (soluciones comerciales y no comerciales) que permiten la duplicación de bases de datos maestras en múltiples sitios de replica.
- ✓ Interfaces nativas para ODBC, JDBC, C, C++, PHP, Perl, TCL, ECPG, Python y Ruby.
- ✓ Reglas
- ✓ Vistas
- ✓ Triggers
- ✓ Unicode
- ✓ Secuencias
- ✓ Herencia
- ✓ Outer Joins
- ✓ Sub-selects
- ✓ Una API abierta

- ✓ Procedimientos almacenados
- ✓ Soporte nativo SSL
- ✓ Lenguajes procedurales
- ✓ Índices parciales y funcionales
- ✓ Soporte para consultas con UNION, UNION ALL y EXCEPT
- ✓ Extensiones para SHA1, MD5, XML y otras funcionalidades
- ✓ Herramientas para generar SQL portable para compartir con otros sistemas compatibles con SQL
- ✓ Sistema de tipos de datos extensible para proveer tipos de datos definidos por el usuario, y rápido desarrollo de nuevos tipos.
- ✓ Funciones de compatibilidad para ayudar en la transición desde otros sistemas menos compatibles con SQL.

### **1.6.6 Fundamentación de la metodología a utilizar.**

Las metodologías imponen un proceso disciplinado sobre el desarrollo de software con el fin de hacerlo más predecible y eficiente. Lo hacen desarrollando un proceso detallado con un fuerte énfasis en planificar inspirado por otras disciplinas de la ingeniería.

El uso de una metodología para la elaboración de un producto informático, garantiza determinadas características en el mismo, dentro de ellas la calidad, factor clave tanto para el cliente como para el productor.

El tiempo es uno de los factores críticos que afecta todo producto, más si no se ha hecho un buen estudio de la aplicación que se va a realizar. La cantidad de personal; algunas veces en exceso, otras en déficit, los sistemas de organización, los métodos de control, el dominio sobre el tema y sobre las herramientas de desarrollo por parte de los analistas y programadores, la falta

de conocimientos sobre asuntos informáticos por el lado de los clientes, son otros de los factores que afectan todo el ciclo de desarrollo de una aplicación.

Sin duda alguna las metodologías que existen actualmente se adecuan al desarrollo de la mayoría de las aplicaciones, puesto que han surgido de la experiencia en la producción acumulada por varios años, incluso por varias décadas; muchas de ellas son el resultado de la evolución y desarrollo de otras unido al cambio producido en el mundo informático, influenciado por el cambio de paradigmas en la programación.

Nuestra investigación se basará en los fundamentos de la metodología del Proceso Unificado de Rational (RUP) utilizando como notación el Lenguaje Unificado de Modelado (UML).

### **Lenguaje Unificado de Modelado (UML)**

El UML es un lenguaje gráfico para visualizar, especificar, construir y documentar los artefactos de un sistema con gran cantidad de software. UML proporciona una forma estándar de escribir los planos de un sistema, cubriendo tanto las cosas conceptuales, tales como procesos del negocio y funciones del sistema, como las cosas concretas, tales como las clases escritas en un lenguaje de programación específico, esquemas de bases de datos y componentes software reutilizables.

### **Proceso Unificado de Rational (RUP)**

El objetivo final de cualquier aplicación, es un software robusto, flexible y escalable, por lo que es necesario tanto un lenguaje como un proceso para poder obtenerlo.

“El *Proceso Unificado de Rational (RUP)*, es un proceso de ingeniería de software planteado por Kruchten (1996) cuyo objetivo es producir software de alta calidad, es decir, que cumpla con los requerimientos de los usuarios dentro de una planificación y presupuesto establecido”.

“RUP toma en cuenta las mejores prácticas en el modelo de desarrollo de software en particular las siguientes:

- ✓ Desarrollo de software en forma iterativa (repite una acción).
- ✓ Manejo de requerimientos.
- ✓ Utiliza arquitectura basada en componentes.
- ✓ Modela el software visualmente (modela con UML).
- ✓ Verifica la calidad del software.
- ✓ Controla los cambios.

El Proceso Unificado de Rational (RUP) consta de cuatro fases o etapas:

- ✓ Fase de comienzo o inicio.
- ✓ Fase de Elaboración.
- ✓ Fase de Construcción.
- ✓ Fase de Transición.

### **1.6.7 Herramientas CASE.**

¿Por qué deberíamos usar herramientas CASE?

A medida que los sistemas que hoy se construyen se tornan más y más complejos, las herramientas de modelado, ofrecen muchos beneficios para todos los involucrados en un proyecto, por ejemplo, administrador del proyecto, analistas, arquitectos, desarrolladores y otros. Las herramientas CASE de modelado, nos permiten aplicar la metodología de análisis y diseño orientados a objetos y abstraernos del código fuente, en un nivel donde la arquitectura y el diseño se tornan más obvios y más fáciles de entender y modificar. Cuanto más grande es un proyecto, es más importante utilizar una herramienta CASE.

*¿Por qué Visual Paradigm como herramienta CASE para nuestro trabajo?*

Escogimos el Visual Paradigm porque es una herramienta CASE que ofrece un entorno de creación de diagramas para UML; diseño centrado en casos de uso y enfocado al negocio que generan un software de mayor calidad; uso de un lenguaje estándar común a todo el equipo de desarrollo que facilita la comunicación; capacidades de ingeniería directa (versión profesional) e inversa; modelo y código que permanece sincronizado en todo el ciclo de desarrollo; disponibilidad de múltiples versiones, para cada necesidad; disponibilidad de integrarse en los principales IDEs; disponibilidad en múltiples plataformas.

### **1.6.8 Otras herramientas necesarias.**

Como se pretende implementar una aplicación Web para confeccionar la propuesta de este trabajo, se hace necesario tener en cuenta la utilización de un editor de páginas Web, y una herramienta para el trabajo con las imágenes.

Para estas funciones la elección no ha sido muy difícil, ya que la herramienta de creación de sitios Web más utilizada en la actualidad es Macromedia Dreamweaver 8. Con esa herramienta se podrá desarrollar cualquier sitio Web personal con características de sitio profesional y utilizar casi todos los recursos de la Web, así como realizar aplicaciones que se ejecuten en servidor y vinculaciones dinámicas de datos; además de contar con un soporte para aplicaciones PHP. También cuenta con un amplio soporte para la creación y utilización de CSS (Cascading Style Sheets), para lograr un diseño fácil y óptimo.

Finalmente se escogió el Adobe Photoshop CS, como herramienta principal para crear las imágenes del Sistema, ya que se considera la aplicación estándar para el tratamiento digital de imágenes. Las continuas mejoras han hecho de este programa uno de los más profesionales para la edición y retoque fotográfico. Tiene un enfoque dirigido hacia los gráficos para la Web.

También se escogió SMARTY que es un motor de plantillas para PHP, cuya finalidad de trabajar con plantillas es la de separar el código PHP del código

HTML, con la ventaja de que un diseñador pueda trabajar en su ámbito sin tener que saber PHP. Por consiguiente, el programador puede hacer los cambios a la lógica de la aplicación sin la necesidad de reestructurar el diseño, y el diseñador puede hacer los cambios a las plantillas sin romper la lógica de la aplicación. Algunos de los principales aspectos de SMARTY son:

- ✓ Es sumamente rápido.
- ✓ Ninguna plantilla se analiza dos veces, sólo compila una vez.
- ✓ Tiene inteligencia para recompilar sólo los archivos de las plantillas que han cambiado.
- ✓ Se pueden hacer funciones personalizadas y personalizar las variables, por lo que el idioma de la plantilla es sumamente extensible.
- ✓ El uso arbitrario de las fuentes de la plantilla, o sea que una plantilla puede ser usada por varias pág. PHP, siempre que muestren el mismo contenido.

### 1.7 Propuesta Solución.

Después de realizar un análisis sobre el concepto general de la Contabilidad y detallar los conceptos de Contabilidad Financiera y Actividad Presupuestada, y determinar claramente cual es la situación actual sobre el objeto de estudio que tiene este trabajo, se concluye que se hace necesario diseñar e implementar un sistema que pueda registrar y controlar todos los hechos económicos que ocurren en un órgano financiero para la actividad presupuesta en la FAR,

De acuerdo a todo el análisis anterior sobre el registro y control de los hechos económicos, se puede señalar que tratándose de una empresa con una estructura organizativa (figura 1.1) con varios niveles, resultaría muy útil que el sistema a diseñar e implementar permita el trabajo simultáneo sobre la base de datos, que sea multiplataforma previendo cualquier cambio futuro y que este

certificado, es decir, que cumpla con los estándares establecidos para sus recuperaciones y que tenga una interfaz amigable en aras de lograr una mayor comunicación usuario-sistema, puesto que el lenguaje visual en que será implementado es ameno, potente y agradable

Basado en los argumentos antes expuestos y dado que se hará uso de tecnologías cliente/servidor sobre plataforma Web para implementar este trabajo, se ha escogido al PHP como lenguaje de programación producto de su portabilidad y eficiencia, como SGBD al PostgreSQL.

### 1.8 Conclusiones.

En este capítulo se detallaron las condiciones y problemas que rodean el objeto de estudio; y a través de los conceptos y definiciones planteadas, se determinaron las condiciones específicas que rodean al problema y en base a esto se obtuvieron los objetivos generales y específicos para este trabajo; se planteó una solución al problema, se realizó un análisis completo de las tecnologías que serán utilizadas a lo largo del desarrollo del sistema propuesto, y se fundamentaron las elecciones del lenguaje, el sistema gestor de bases de datos, y la metodología a utilizar. Una vez conocidas las herramientas optimas, y los conceptos a utilizar se puede empezar a desarrollar la propuesta de sistema.



# Capítulo 2

Introducción

Modelación del negocio

Diagrama de Caso de Uso

Diagrama de Actividades

Modelo de Objetos

Requerimientos Funcionales

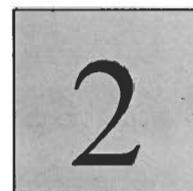
Requerimientos no Funcionales

Descripción del Sistema Propuesto

Modelo de Caso de Uso del Sistema

Expansión de los Casos de Uso

Conclusiones



## CAPÍTULO DESCRIPCIÓN DE LA PROPUESTA SOLUCIÓN

### 2.1 Introducción.

En el presente capítulo se hace la descripción de la propuesta que trae esta investigación, para ello se describen los procesos del negocio que tiene que ver con el objeto de estudio. Para poder entender el contexto que se investiga, definimos un Modelo de Negocio.

Además se enumeran los requisitos funcionales y no funcionales que debe tener el sistema que proponemos, lo que permite hacer una concepción general del sistema, e identificar mediante un Diagrama de Caso de Uso, las relaciones de los actores que interactúan con el sistema, y las secuencias de acciones con las que interactúan.

### 2.2 Modelado del negocio.

Para conseguir sus objetivos, una empresa organiza su actividad por medio de un conjunto de procesos de negocio. Cada uno de ellos se caracteriza por una colección de datos que son producidos y manipulados mediante un conjunto de tareas, en las que ciertos agentes (por ejemplo, trabajadores o departamentos) participan de acuerdo a un flujo de trabajo determinado. Además, estos procesos se hallan sujetos a un conjunto de reglas de negocio, que determinan la estructura de la información y las políticas de la empresa. Por tanto, la finalidad del modelado del negocio es describir cada proceso del negocio, especificando sus datos, actividades (o tareas), roles (o agentes) y reglas de negocio.

Una vez se han identificado los procesos de negocio, es preciso encontrar los agentes involucrados en su realización. Cada uno de estos agentes o actores del negocio desempeña cierto papel (juega un rol) cuando colabora con otros para llevar a cabo las actividades que conforman dicho caso de uso del negocio. De hecho, identificaremos los roles que son jugados por agentes de la propia empresa (que incluyen trabajadores, departamentos y dispositivos físicos) o agentes externos (como clientes u otros sistemas).

Por el momento nos centraremos en este último tipo de roles, con los que la organización interactúa para llevar a cabo sus procesos de negocio. En nuestro trabajo tenemos el rol Cliente que puede manifestarse en forma de Banco, Especialidad u Órgano Financiero, claramente externos al sistema.

Para tener una visión general de los diferentes procesos de negocio de la organización, puede construirse un diagrama de casos de uso del negocio, en el cual aparece cada proceso del negocio como un caso de uso. Este diagrama permite mostrar los límites y el entorno de la organización bajo estudio. Por esta razón, sólo aparecerán en este diagrama los actores del negocio correspondientes a los roles externos al sistema, de forma que los procesos de negocio en los que sólo tomen parte roles internos a la organización no estarán conectados a ningún actor. En la Figura 2.1 se muestra el diagrama de casos de uso del negocio para nuestro trabajo; es un diagrama de casos de uso UML formado por casos de uso del negocio y actores.

Un modelo de casos de usos del negocio describe los procesos de negocio de una empresa en términos de caso de uso del negocio y actores del negocio que se corresponden con los procesos del negocio y los clientes respectivamente; este presenta un negocio desde su perspectiva de uso, y esquematiza cómo proporciona valor a sus usuarios.

### 2.3 Diagrama de caso de uso del negocio.

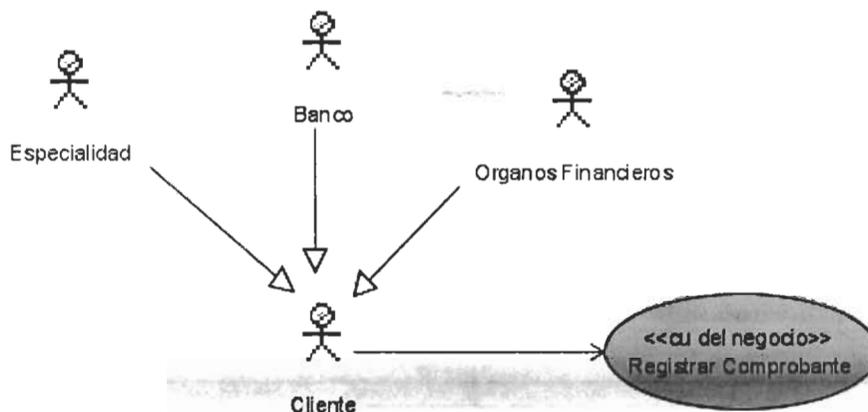


Figura 2.1 Diagrama de caso uso del negocio “Registrar Comprobante”

### 2.4 Descripción del caso de uso del negocio.

El siguiente paso dentro del modelado del negocio es introducirse en cada uno de los casos de uso del negocio identificados, para describirlo en detalle. Inicialmente se rellena una plantilla de descripción; y después, a partir de la información reflejada en dicha plantilla, se construye un conjunto de diagramas que describen completamente el caso de uso del negocio. En nuestro caso nos centraremos en el único caso de uso del negocio, “Registrar Comprobante”, cuya descripción se muestra a continuación.

Nombre del CU del Negocio:	Registrar Comprobante
Actores del negocio:	Especialistas, Banco, órganos financieros y Cliente
Propósito:	Contabilizar cualquier movimiento contable.
Resumen:	Se inicia cuando a un órgano financiero llega cualquier movimiento contable, este puede ser una Asignación del Presupuesto , una acumulación de nóminas, recibo de efectivo, vale para pagos menores, etc.

y finaliza cuando queda registrado en el comprobante de operaciones.	
<b>Flujo de Normal de Eventos</b>	
<b>Acción del actor</b>	<b>Respuesta del negocio</b>
1- El cliente entrega al órgano financiero el documento con el movimiento contable.	<p>2- El contador recepciona el documento que contiene el movimiento contable.</p> <p>3- El contador determina el tipo de operación y la cuenta contable a aplicar.</p> <p>4- Si es una asignación, ver la naturaleza de la cuenta, si es deudora se registra el importe al debe, sino se registra al haber. De lo contrario si es un retiro, pregunto si es deudora, registrar al haber, sino registrar al debe.</p> <p>5- Una vez contabilizado se genera el comprobante de operaciones.</p> <p>6- El Jefe de Finanzas lo recibe y aprueba.</p> <p>7.- Queda aprobado el comprobante de operaciones.</p>
<b>Prioridad:</b>	Este caso de uso es el centro de un sistema de contabilidad financiera a partir de esto se generan todos los reportes del sistema.
<b>Mejoras:</b>	Al automatizar este proceso se podrá realizar la contabilización de una forma más rápida, evitando la

	introducción de errores y cuadrando de forma automatizada todas las cuentas, así como el cómputo de todas las operaciones.
<b>Cursos alternos:</b> En el paso 2 si se entrega un documento con errores el contador se lo devuelve al cliente. En el paso 6 si el jefe detecta algún error se lo entrega al contador para su revisión	

## 2.5 Diagrama de actividades del caso de uso del negocio.

Para mostrar de forma más detallada el flujo de trabajo que realiza cada proceso del negocio, utilizaremos diagramas de actividades que no es más que un diagrama de flujos (paralelos o secuenciales) que muestra el flujo de control de una actividad a otra.

La figura 2.2 muestra el diagrama de proceso que incluye el escenario de la figura 2.1, existe una calle por cada rol participante en el escenario, que incluye las actividades que realiza dicho rol. El diagrama también muestra la información que necesita y produce cada actividad, y la sincronización requerida entre las diferentes actividades.

## Capítulo 2 Descripción de la propuesta solución

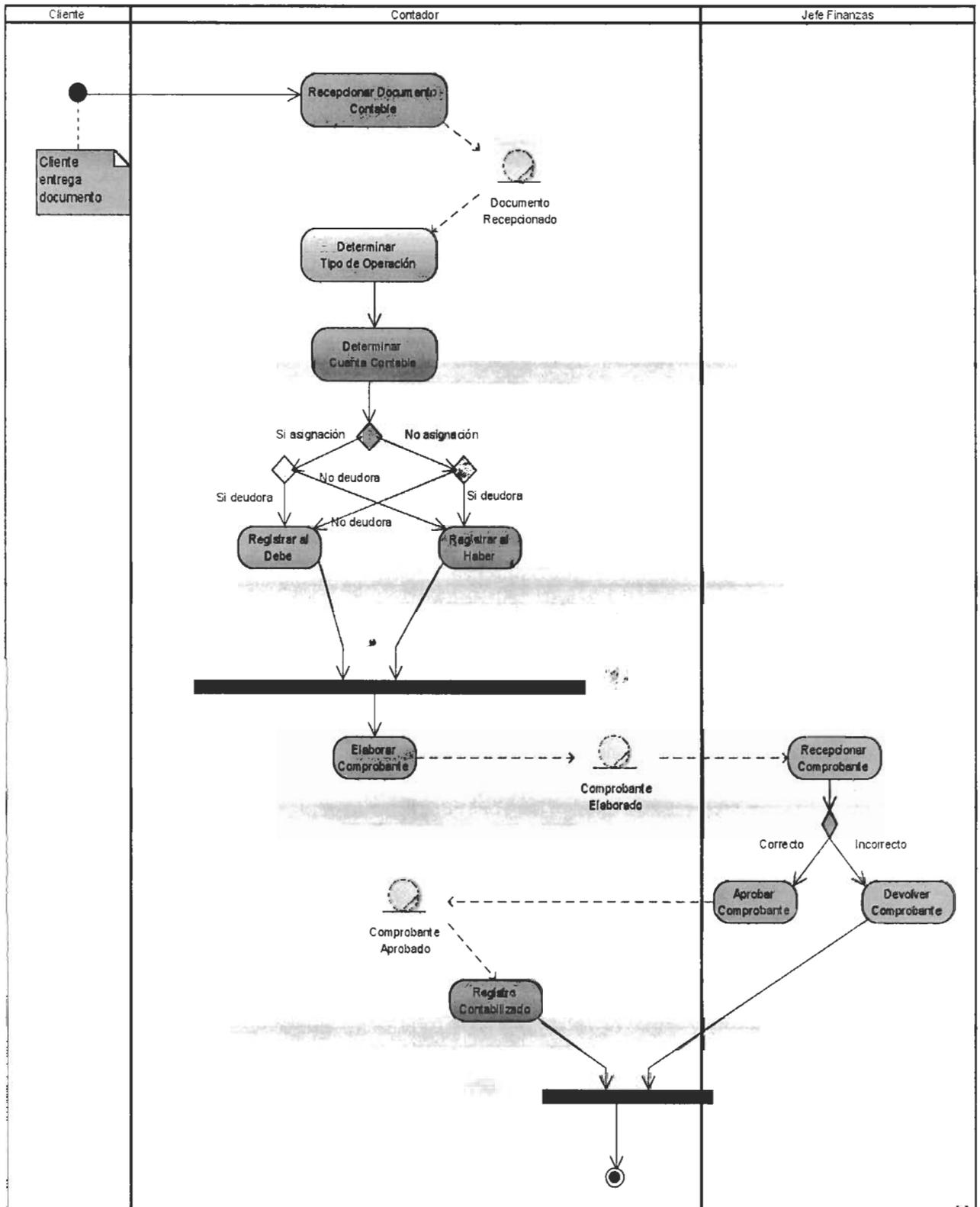


Figura 2. 2 Diagrama de Actividades del caso de uso "Registrar Comprobante".

## 2.6 Modelo de Objetos.

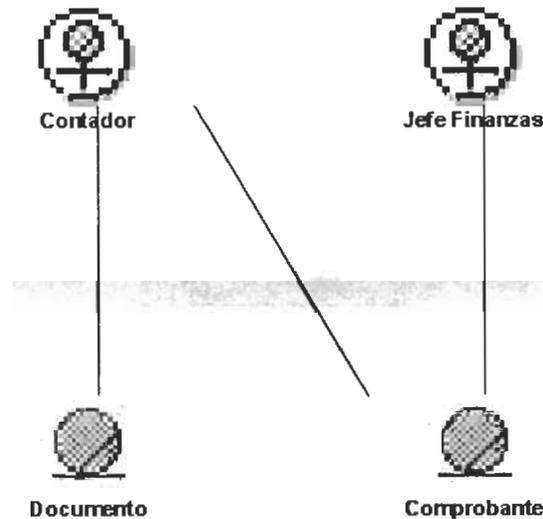


Figura 2.3 Modelo de Objeto

## 2.7 Requerimientos funcionales.

Los requerimientos son una descripción de las necesidades o deseos de un producto. Su meta principal es identificar y documentar lo que en realidad se necesita. Podríamos preguntarnos: ¿Qué acciones debe ejecutar el sistema para que cumpla con los objetivos planteados al inicio de este trabajo?

Esas acciones que el sistema debe ser capaz de realizar, se convierten en los requerimientos funcionales y de acuerdo a los objetivos propuestos el sistema debe ser capaz de:

### **R1. Gestionar Comprobante de Operación.**

1.1. Permitir que el contador actualice un comprobante.

1.1.1. Permitir que el contador elabore un comprobante de operaciones entrando al sistema el tipo de operación, la fecha, el órgano y el número del documento.

## Capítulo 2 Descripción de la propuesta solución

---

- 1.1.2. Permitir que el contador modifique un comprobante.
  - 1.2. Permitir que el contador actualice un movimiento contable asociado a un comprobante de operación.
    - 1.2.1. Permitir que el contador inserte un movimiento contable.
    - 1.2.2. Permitir que el contador modifique un movimiento contable.
    - 1.2.3. Permitir que el contador elimine un movimiento contable.
    - 1.2.4. Permitir al contador y al jefe de finanzas mostrar un comprobante de operaciones.
  - 1.3. Permitir que el jefe de finanzas controle un comprobante.
    - 1.3.1. Mostrar todos los registros de un órgano financiero.
    - 1.3.2. Mostrar la fecha en que fueron confeccionados los registros.
    - 1.3.3. Mostrar el estado de confirmación del comprobante.
    - 1.3.4. Mostrar el usuario que confeccionó el comprobante.
  - 1.4. Permitir que el contador confirme un comprobante de operación.
- R2. Desconfirmar un comprobante de operación.**
- 2.1. Permitir que el jefe de finanzas desconfirme un comprobante.
- R3. Confeccionar el Balance de Comprobación de Saldos (Estado A).**
- 3.1. Mostrar e imprimir el Estado A.
- R4. Confeccionar Mayor (SCF-50).**
- 4.1. Mostrar e imprimir el Mayor (SCF-50).
- R5. Confeccionar SubMayores.**
- 5.1. Mostrar e imprimir el submayor de Gastos del presupuesto (SCF-52).
  - 5.2. Mostrar e imprimir submayor de Ingresos (SCF-53).
    - 5.2.1. Submayor de Ingresos por consumidor.
    - 5.2.2. Submayor de ingresos por administrador.

5.3. Mostrar e imprimir Submayor de Cuentas por Cobrar y Pagar.

**R6. Preparar Presupuesto/Financiamiento.**

6.1. Permitir una asignación de presupuesto.

6.2. Permitir una distribución de presupuesto (P9).

6.3. Permitir realizar una situación de fondo.

6.4. Permitir controlar los presupuestos.

6.5. Permitir controlar los financiamientos.

**R7. Analizar Centro de Costo.**

7.1. Permitir al contador analizar por subcuentas y centro de costos.

7.2. Permitir al contador analizar por análisis y subcuentas.

7.3. Permitir al contador analizar por análisis y centro de costos.

**R8. Iniciar/Cerrar Operación Año.**

8.1. Permitir al contador iniciar y/o cerrar operación año.

**R9. Traspasar a Estados Financieros.**

9.1. Traspasar información (Estado A) al módulo de Estados Financieros.

**R10. Actualizar Clasificadores.**

10.1. Permitir al jefe de finanzas actualizar los clasificadores.

**R11. Buscar Comprobante.**

11.1. Permitir al contador o jefe de finanzas buscar un comprobante.

11.2. Mostrar un comprobante cuando la búsqueda es realizada por registro.

11.3. Mostrar una lista de comprobante cuando la búsqueda parte de un rango de fecha o un tipo de operación.

### 2.8 Requerimientos no funcionales.

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

#### Apariencia o interfaz externa:

- ✓ Diseño sencillo, con pocas entradas, 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 bases 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.

## Capítulo 2 Descripción de la propuesta solución

---

- ✓ Por parte del cliente se requiere un navegador capaz de interpretar JavaScript.

### Portabilidad:

- ✓ Necesidad de que el sistema sea multiplataformas.

### 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, nuestro modelo ideal (cliente/servidor), 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.

## 2.9 Descripción del sistema propuesto.

Para cumplimentar los objetivos propuestos al inicio de este trabajo, y teniendo en cuenta todos los requerimientos planteados, el sistema que se propone, para una mejor comprensión se ha dividido en tres paquetes (figura 2.4), un paquete que contiene todos los casos de usos relacionados con la gestión de

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, nuestro modelo ideal (cliente/servidor), 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.

## 2.9 Descripción del sistema propuesto.

Para cumplimentar los objetivos propuestos al inicio de este trabajo, y teniendo en cuenta todos los requerimientos planteados, el sistema que se propone, para una mejor comprensión se ha dividido en tres paquetes (figura 2.4), un paquete que contiene todos los casos de usos relacionados con la gestión de

comprobantes, otro paquete que agrupa todos los casos de usos relacionados con la recuperaciones que se pueden obtener a partir de uno o más comprobantes y un último paquete que agrupa todo lo relacionado con la actualización de los clasificadores.

Se considera la existencia de dos roles; un usuario se puede comportar como: contador o jefe de finanzas.

El paquete "Gestionar Comprobante" puede ser utilizado por todos los roles de usuario, y pretende realizar cualquier operación sobre un comprobante de operación, ya sea elaborarlo, modificarlo, eliminarlo, consultarlo; así como insertarle, modificarle o eliminarle un movimiento contable asociado a un comprobante y confirmarlo, desconfirmarlo o controlarlo.

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

Finalmente el paquete "Clasificadores" controlará la actualización de, como bien indica su nombre, los clasificadores, más conocidos como nomencladores, que no son más que objetos de información que permanecen constante frecuentemente.

### 2.10 Modelo del caso de uso del sistema.

Según la experiencia de muchos analistas las actividades del diagrama de actividades tienen el nivel de granularidad adecuado para ser asociadas a un caso de uso del sistema. De esta manera, crearemos un caso de uso del sistema por cada actividad del diagrama de actividades que deba ser soportada por el sistema software. 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.

Algunos casos de uso no se obtendrán directamente a partir del diagrama de actividades. Estos nuevos casos de uso se detectarían al describir los casos de uso identificados y adquirir un mayor conocimiento sobre los requisitos que deben ser soportados, y representarían funciones que debe llevar a cabo el sistema para lograr algún objetivo asociado con algún caso de uso ya existente.

Un caso de uso es un documento narrativo que describe la secuencia de un actor (agente externo) que utiliza un sistema para completar un proceso. Un actor no es parte del sistema, sino un rol que se juega dentro del sistema, que puede intercambiar información o puede ser un recipiente pasivo de información y representa a un ser humano, a un software o a una máquina que interactúa con el sistema. En esta investigación intervienen dos actores:

<b>Actores</b>	<b>Descripción</b>
Contador	Un contador tiene la función de contabilizar todas las operaciones contables de un órgano financiero.
Jefe de Finanzas	El Jefe de Finanzas tiene la función de controlar los registros contabilizados en comprobantes; es el encargado de actualizar los clasificadores y de desconfirmar los comprobantes en caso de error.

Los casos de uso se pueden organizar en varias formas, de acuerdo con la descomposición propuesta en la descripción del sistema, en la que planteamos agrupar los casos de uso por paquetes, el resultado sería el siguiente:

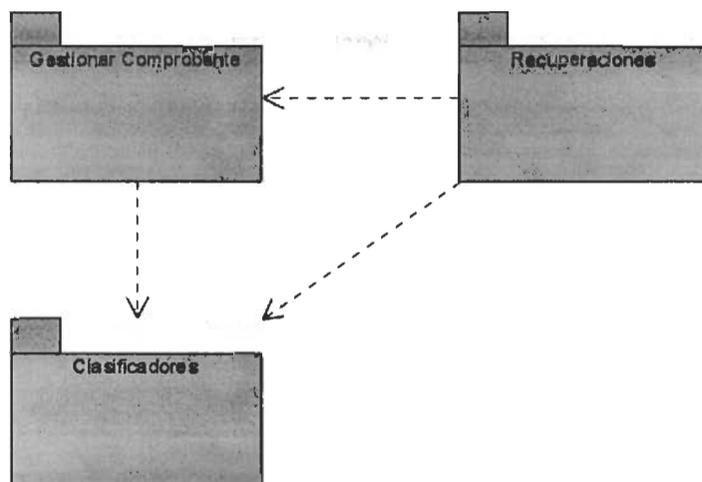


Figura 2.4 Diagrama de Paquetes que componen el sistema.

Los casos de uso determinados para satisfacer las necesidades del sistema en el paquete “Gestionar Comprobante” serían:

CU-1	Gestionar Comprobante.
Actor	Contador
CU asociado	Actualizar Movimiento Contable (include)
Descripción	El contador puede elaborar, modificar y/o mostrar un comprobante de operaciones, además de insertar, modificar o eliminar uno o varios movimientos contables asociados a un comprobante determinado.
Referencia	R1

CU-2	Desconfirmar Comprobante.
Actor	Jefe de Finanzas
CU asociado	Buscar comprobante (include)
Descripción	El jefe de finanzas puede estimar que un comprobante está mal elaborado y puede desconfirmarlo.
Referencia	R2

CU-3	Controlar Comprobante.
Actor	Jefe de Finanzas
Descripción	El jefe de finanzas puede manipular información referente a un comprobante que le permite llevar el control de este.
Referencia	R1

CU-4	Buscar Comprobante.
Actor	Contador, Jefe de Finanzas
CU asociado	Gestionar Comprobante (extend)
Descripción	El contador o jefe de finanzas solicitan buscar un comprobante para realizar alguna operación sobre este.

Referencia	R11
------------	-----

El diagrama de casos de uso perteneciente a dicho paquete es el que se muestra a continuación:

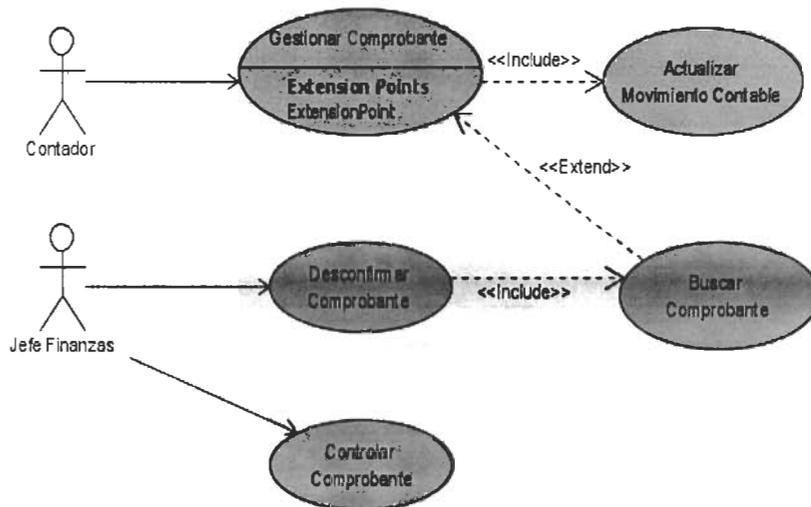


Figura 2.5 Diagrama de CU que conforman el paquete “Gestionar Comprobante”

Por otra parte, para el paquete “Recuperaciones” se han determinado los casos de uso:

CU-5	Confeccionar Balance de Comprobación de Saldos (Estado A)
Actor	Contador
Descripción	El contador puede mostrar e imprimir el Estado A de los comprobantes que el escoja en un rango de fecha determinado.
Referencia	R3

CU-6	Confeccionar SubMayores
Actor	Contador
Descripción	El contador puede mostrar e imprimir varios submayores.
Referencia	R5

CU-6	Analizar Centro de Costo.
Actor	Contador
Descripción	El contador puede analizar las cuentas por subcuentas, centro de costos o análisis.

## Capítulo 2 Descripción de la propuesta solución

Referencia	R7
------------	----

CU-7	Iniciar/Cerrar Operación Año.
Actor	Contador
Descripción	El contador puede registrar la cuenta que inicia y/o cierra el año
Referencia	R8

CU-8	Traspasar a Estados Financieros
Actor	Contador
Descripción	El contador puede traspasar información al módulo Estado Financiero.
Referencia	R9

CU-9	Preparar Presupuesto/Financiamiento.
Actor	Contador
Descripción	El contador puede asignar, distribuir, controlar presupuestos y financiamientos.
Referencia	R6

CU-10	Confeccionar Mayor (SCF- 50)
Actor	Contador
Descripción	El contador puede mostrar e imprimir el Mayor.
Referencia	R4

El diagrama de casos de uso correspondiente a este paquete queda como se muestra en la figura 2.6.

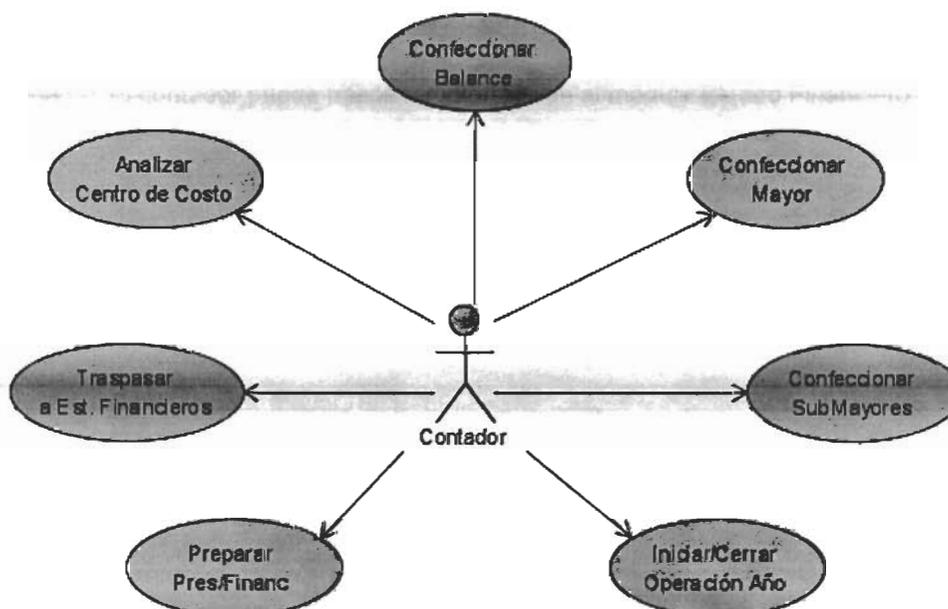


Figura 2.6 Diagrama de CU que conforman el paquete “Recuperaciones”.

En el paquete “Clasificadores” quedaría el siguiente caso de uso con su respectivo diagrama como muestra la figura 2.7:

<b>CU-11</b>	Actualizar Clasificadores
<b>Actor</b>	Jefe de Finanzas
<b>Descripción</b>	El jefe de finanzas puede en un momento determinado actualizar un nomenclador (clasificador).
<b>Referencia</b>	R10

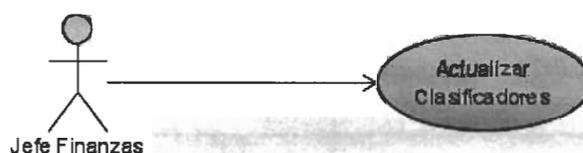


Figura 2.7 Diagrama de CU que componen el paquete “Clasificadores”.

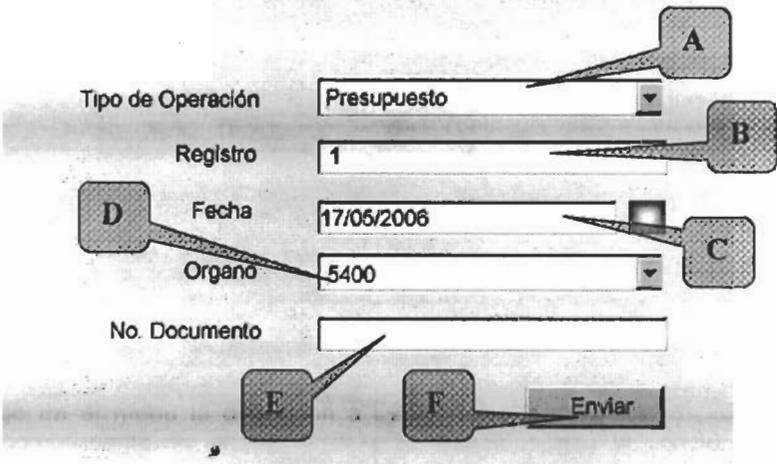
Una vez definidos los casos de usos que van a conformar nuestro sistema y explicados brevemente, centraremos nuestro trabajo sobre el paquete “Gestionar Comprobante” y sobre este aplicaremos el resto de los flujos de trabajo que establece RUP.

### 2.11 Expansión de los casos de uso.

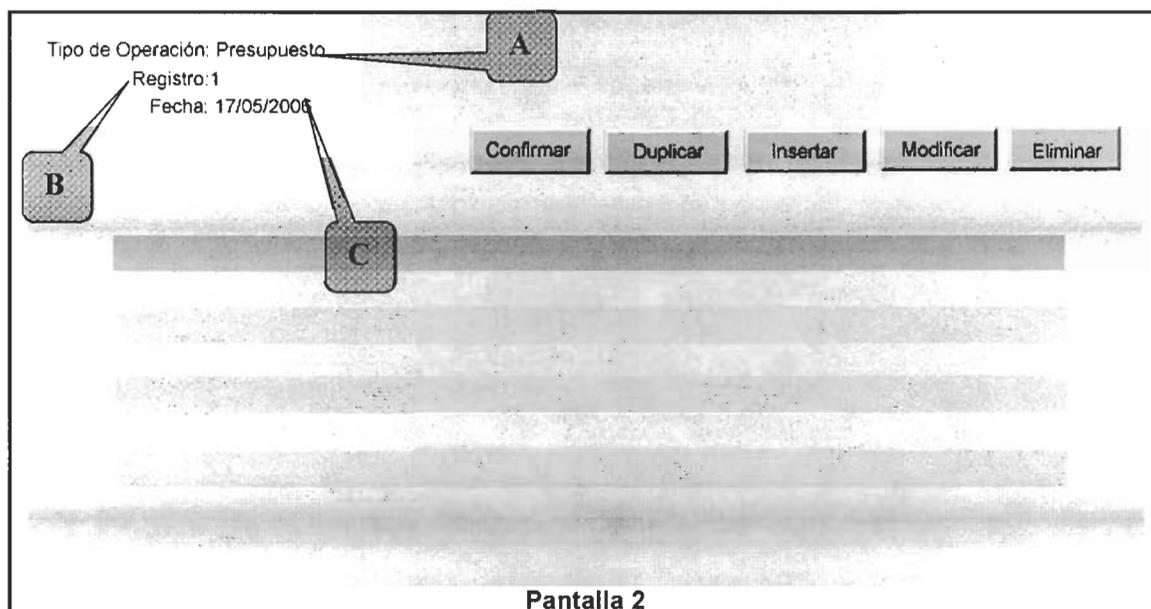
Mediante los casos de uso expandidos se describe paso a paso la secuencia de eventos que los actores utilizan para completar un proceso a través del sistema. Este sería el último paso en el análisis, para pasar a la construcción de la solución propuesta.

<b>Caso de Uso:</b>	<b>Gestionar Comprobante</b>
<b>Actor(es):</b>	Contador (inicia)
<b>Propósito:</b>	Realizar una acción sobre un comprobante de operación, que puede ser imprimirlo, modificarlo o elaborarlo.
<b>Resumen:</b>	El caso de uso inicia cuando el contador necesita realizar una acción determinada sobre un comprobante. De acuerdo a su requerimiento puede elaborar un comprobante, modificarlo, eliminarlo o visualizarlo.
<b>Referencias:</b>	R1.
<b>CU asociado:</b>	Buscar Comprobante(exclude), Actualizar Movimiento Contable(include)
<b>Precondiciones:</b>	El usuario, que debe autenticarse como contador, levanta el sistema y se encuentra con la página principal.

## Capítulo 2 Descripción de la propuesta solución

Acción del Actor	Respuesta del Sistema
1. El Usuario elige en el menú la operación a realizar sobre un comprobante.	2. <ul style="list-style-type: none"> <li>a) Si elige la opción Elaborar, ver sección: "Elaborar Comprobante".</li> <li>b) Si elige la opción Modificar, ver sección: "Modificar Comprobante"</li> <li>c) Si elige la opción Imprimir, ver sección: "Mostrar Comprobante".</li> </ul>
<b>Sección: "Elaborar Comprobante"</b>	
 <p style="text-align: center;"><b>Pantalla 1</b></p>	
1. El usuario escoge en el menú la opción "Elaborar Comprobante".	2. El Sistema muestra la Pantalla 1, con los datos a introducir, donde Registro (B), es autonumérico.
3. El usuario obligatoriamente:	
3.1 Escoge un Tipo de Operación (A)	3.2 El sistema guarda el valor del atributo escogido, según la selección del usuario (ver Anexo 1).
3.3 Selecciona la fecha (C)	3.4 Se propone por defecto la fecha actual.
4. Opcionalmente decide según los movimientos contables asociados a ese comprobante:	
4.1 Escoge o Introduce el órgano (D).	4.2 Permite escoger o introducir el valor, en caso de introducir, valida que ese órgano exista en la base datos.
4.3 Introduce el No. Documento (E)	4.4 Valida que sean números (longitud máxima 10 dígitos) puede comenzar con cero Ej:0000034
4.5 Hace clic en el botón Enviar (F)	4.6 Se muestra un mensaje de confirmación informando los datos que introdujo en la Pantalla 1 y muestra la Pantalla 2 donde aparece el tipo de operación escogida (A), el registro asignado por el sistema (B) y la fecha (C).

## Capítulo 2 Descripción de la propuesta solución

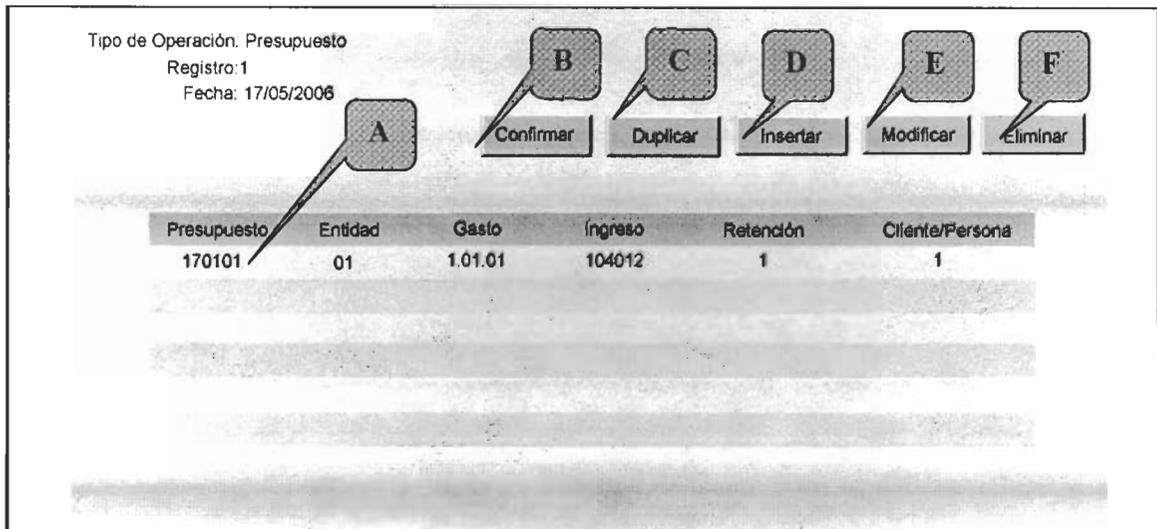


Pantalla 2

### Sección: "Modificar Comprobante"

1. El usuario selecciona en el menú la opción "Modificar Comprobante".	2. El sistema: 2.1. (Ver CU: Buscar Comprobante).
	3. Determinar según el Tipo de Operación escogido los campos a mostrar en la pantalla 3, estos pueden ser: <ul style="list-style-type: none"> <li>✓ <u>Tabla 1</u>, es para cualquier tipo de operaciones escogida, excepto "Acumulación de Nóminas" y "Centro de Costo"</li> <li>✓ <u>Tabla 2</u>, solo para el tipo de operación "Acumulación de Nóminas".</li> <li>✓ <u>Tabla 3</u>, solo para el tipo de operación "Centro de Costo".</li> </ul> 3.1 Muestra la pantalla 3, con un grid (A), que contiene todos los movimientos contables asociados al comprobante seleccionado.
	4. Brinda la posibilidad de realizar cualquier operación sobre un movimiento contable, puede ser: Insertar, Modificar, Eliminar.
5. Hace clic en Confirmar (B).	5.1 Ver CU: Actualizar Movimiento Contable
6. Hace clic en Duplicar (C).	6.1 Ver CU: Actualizar Movimiento Contable
7. Hace clic en Insertar (D).	7.1 Ver CU: Actualizar Movimiento Contable (sección "Insertar Movimiento Contable").
8. Hace clic en Modificar (E).	8.1 Ver CU: Actualizar Movimiento Contable (sección "Modificar Movimiento Contable").
9. Hace clic en Eliminar (F).	9.1 Ver CU: Actualizar Movimiento Contable (sección "Eliminar Movimiento Contable").

Capítulo 2 Descripción de la propuesta solución



Pantalla 3

Tabla 1

Presup.	Entidad	Gasto	Ingreso	Retención	Cliente/Persona

Tabla 2

Presup.	Entidad	Gasto	Hombres

Tabla 3

Costo	Análisis

Poscondiciones:	Se visualiza el comprobante (SCF-56).
Cursos alternos:	<p>Sección "Elaborar Comprobante"</p> <p>4.6.1 Si presionar F en la pantalla 2 y los campos obligatorios (A, B, C) no tiene datos, mostrar un mensaje de error advirtiendo esto al usuario.</p> <p>Sección "Modificar Comprobante"</p> <p>3.2.3 Si se presiona buscar (D) en la pantalla 4 y no existe al menos un campo con datos el sistema mostrará un mensaje de error informando esto al usuario.</p> <p>Sección "Mostrar Comprobante"</p> <p>Si el usuario no selecciona ningún comprobante de la pantalla 5 el sistema mostrará un error informando que debe seleccionar al menos uno.</p>

<b>Caso de Uso:</b>	<b>Actualizar Movimientos Contables.</b>
Actor(es):	Contador (inicia).
Propósito:	Manipular los movimientos contables asociados a un comprobante de operaciones de un Órgano Financiero.

## Capítulo 2 Descripción de la propuesta solución

Resumen:	El caso de uso se inicia cuando el contador decide manipular información referente a un comprobante y termina cuando realiza una acción sobre este.
Referencias:	R1
Precondiciones:	El usuario debe haberse autenticado como contador y debe existir al menos un comprobante.
Acción del Actor	Respuesta del Sistema
	<p>1. El sistema muestra la pantalla 5 para insertar los movimientos contables asociados al comprobante de operaciones que está elaborando.</p> <p>1.1 La interfaz muestra:</p> <ul style="list-style-type: none"> <li>✓ los datos asociados a ese comprobante de manera informativa (A1).</li> <li>✓ DbGrid (A2), estado inicial vacío</li> <li>✓ Controles de edición (A3).</li> <li>✓ Botones de acción (1, 2, A, B, C).</li> </ul>
<b>Sección: "Insertar Movimiento Contable"</b>	
	<p>2- En dependencia del tipo de operación el sistema muestra la pantalla 5 donde, para cada una, A3 contiene los controles de edición.</p> <p>2.1 - A3 puede estar compuesto por la tabla 1, tabla 2 y tabla 3 (CU: Gestionar Comprobante, sección Modificar Comprobante)</p>
	<b>3- Si tabla 1</b>
3.1 – El usuario escoge o introduce el grupo presupuestario	3.1.1 - El sistema permite escoger o introducir el valor del grupo presupuestario.
3.2 - Escoge o introduce el código de la entidad.	3.2.1 - El sistema permite escoger o introducir el valor de la entidad.
3.3 - Escoge o introduce el código del gasto.	3.3.1 – Permite escoger o introducir el valor del gasto.
3.4 - Escoge o introduce el código del ingreso	3.4.1 - Permite escoger o introducir el valor del ingreso.
3.5 - Escoge la retención.	3.5.1 - Permite escoger el valor de la retención.
3.6 - Escoge o introduce el cliente civil o militar.	3.6.1 - Permite escoger o introducir el valor del cliente.
3.7 - Escoge o introduce la persona.	3.7.1 - Permite escoger o introducir el valor de la persona.
	<b>4- Si tabla 2</b>
4.1 – El usuario escoge o introduce el grupo presupuestario.	4.1.1 - Permite escoger o introducir el valor del grupo presupuestario.
4.2 - Escoge o introduce el código de la entidad.	4.2.1 - Permite escoger o introducir el valor de la entidad.
4.3 - Escoge o introduce el código del gasto.	4.3.1 - Permite escoger o introducir el valor del gasto.
4.4 - Introduce la cantidad de hombres.	4.4.1 - Permite introducir la cantidad de hombre.
	<b>5 – Si tabla 3</b>
5.1 - Escoge el costo.	5.1.1 - Permite escoger o introducir el valor del costo.
5.2 – Escoge o introduce el código del análisis.	5.2.1 - Permite escoger o introducir el valor del análisis.
6- Escoge la cuenta al DEBE o introduce el código de la cuenta (puede llenarse o no)	6.1 - Permite escoger o introducir el valor del debe.
7- Escoge la cuenta al HABER o introduce el	7.1 - Permite escoger o introducir el valor del

## Capítulo 2 Descripción de la propuesta solución

código de la cuenta (puede llenarse o no.)	haber.
8- Introduce el importe	8.1 - Permite escoger o introducir el valor del importe.
<b>Pantalla 5</b>	
11. Hace clic en Aceptar (E)	11.1 Chequea que estén llenos los valores que son obligatorios, guarda los datos en la BD y muestra el movimiento insertado en el grid.
12. Hace clic en Cancelar (F)	12.1 Cancela la inserción de datos y muestra la pantalla 2.
<b>Sección: "Modificar Movimiento Contable"</b>	
1. Hace clic en Modificar (A)	1.1 Muestra la pantalla 5 mostrando todos los movimientos contables de un comprobante.
2. Selecciona en el grid (D) el movimiento que quiere modificar.	2.1 Muestra la pantalla 6 con las mismas características de la pantalla 5 pero en el área (A3) muestra los datos del movimiento.
3. Modifica los datos en el área A3 y hace clic en Aceptar (B).	3.1 Guarda los datos en la BD y muestra en el grid en movimiento con los nuevos datos.
4. Hace clic en Cancelar (C).	4.1 Cancela la modificación de datos y muestra la pantalla 1.

## Capítulo 2 Descripción de la propuesta solución

Tipo de Operación: Presupuesto  
 Registro: 1  
 Fecha: 17/05/2006

A

Presupuesto	Entidad	Gasto	Ingreso	Retención	Cliente/Persona
170101	01	1.01.01	104012	1	1
170101	02	1.01.01	104015	1	0

D

A3

Grupo Presupuestario	Especialidad	Partida de Gasto	Cuenta de Ingreso
170101	01	1.01.01	104012
Cliente	Retención		
1	1		

C

Pantalla 6

### Sección: "Eliminar Movimiento Contable"

- |   |  |
|---|--|
| 1. Selecciona un movimiento contable de la pantalla 2 y hace clic en eliminar (A) | 1.1 Muestra la pantalla 7.                         |
| 2. Hace clic en Aceptar (B)   | 2.1 Muestra un mensaje de confirmación de borrado. |
| 3. Confirma borrado del movimiento  | 3.1 Elimina movimiento de la BD.                   |
| 4. Hace clic en Cancelar (C).   | 4.1 Muestra un mensaje de confirmación.            |
| 5. Confirma cancelar operación.   | 5.1 Cancela la acción y muestra pantalla 2.        |

Tipo de Operación: Presupuesto  
 Registro: 1  
 Fecha: 17/05/2006

A

Presupuesto	Entidad	Gasto	Ingreso	Retención	Cliente/Persona
170101	01	1.01.01	104012	1	1
170101	02	1.01.01	104015	1	0

C

Pantalla 7

### Sección: "Mostrar Comprobante"

- |   |                  |
|---|------------------|
| 1- El usuario escoge en el menú la opción | El sistema debe: |
|---|------------------|

## Capítulo 2 Descripción de la propuesta solución

"Imprimir Comprobante".	<p>1.1 Si está en curso el CU: Buscar v Comprobante.</p> <ul style="list-style-type: none"> <li>○ Mostrar pantalla 4 con los datos del comprobante (SCF-56) que se seleccione.</li> </ul> <p>1.2 Si no está en curso el CU: Buscar Comprobante.</p> <ul style="list-style-type: none"> <li>○ Debe buscar el comprobante que quiere visualizar (Ver CU: Buscar Comprobante)</li> </ul>
	<p>1.3 El sistema para cada comprobante debe:</p> <p>1.3.1 Seleccionar todos los atributos de una cuenta en la BD.</p> <p>1.3.2 Ordenar los movimientos contables:</p> <ul style="list-style-type: none"> <li>✓ Por la cuentas que están al debe y dentro de estas: <ul style="list-style-type: none"> <li>○ Las cuentas de patrimonio.</li> <li>○ Las cuentas de orden.</li> </ul> </li> <li>✓ Las cuentas que están al haber y dentro de estas: <ul style="list-style-type: none"> <li>○ Las cuentas de patrimonio.</li> <li>○ Las cuentas de orden.</li> </ul> </li> </ul>
	<p>1.4 Para todas las cuentas al debe:</p> <p>1.4.1 Acumular el valor total de todos sus movimientos contables al debe.</p> <p>1.5 Registrar en la tabla de salida de la pantalla 6.</p>
	<p>1.6 Para las cuentas al debe que llevan análisis:</p> <ul style="list-style-type: none"> <li>✓ Agrupar para cada uno de los movimientos contables la información: <ul style="list-style-type: none"> <li>○ Por grupo presupuestario</li> <li>○ Ingreso</li> <li>○ Retención</li> </ul> </li> </ul> <p>Registrar en la tabla de salida de la pantalla 6.</p>
	<p>1.7 Para las subcuentas de las cuentas al debe.</p> <ul style="list-style-type: none"> <li>✓ Determinar los padres de las subcuentas.</li> </ul> <p>Determinar el importe total de la cuenta padre a partir de la suma de los importes de las subcuentas.</p>
	<p>1.8 Para todas las cuentas al haber:</p> <ul style="list-style-type: none"> <li>✓ Acumular el valor de todos sus movimientos contables al haber.</li> </ul> <p>Registrar en la tabla de salida correspondiente a la pantalla 6.</p>
	<p>1.9 Para todas las cuentas que llevan análisis al haber:</p> <p>Lo mismo que 1.5</p>
	<p>1.10 Para las subcuentas de las cuentas al haber.</p> <p>Lo mismo que 1.6</p>
<p><b>Ver Anexo 2 (Tabla SCF-56)</b> <b>Pantalla 4</b></p>	

## Capítulo 2 Descripción de la propuesta solución

2. Hace clic en el botón Confirmar (1).	2.1 Envía un mensaje indicado la confirmación del comprobante. 2.2 Activa el estado confirmar del comprobante en la BD.
3. Hace clic en el botón Duplicar (2).	3.1 El sistema muestra en A3 (pantalla 5) los valores guardados en las variables temporales, así evita que tenga que teclear una serie de valores que pueden repetirse y modificar lo que necesite.
<b>Cursos Alternos</b>	
Si se desea insertar y no hay datos en los campos del área A1, se muestra un mensaje de error. Si no se modifica ningún campo, no se guardan los datos y se mantienen los mismos. Si no se selecciona ningún campo del grid para modificar o eliminar, se muestra un mensaje indicando la acción. Si clic en Confirmar y no hay movimiento contable asociado al comprobante, se muestra mensaje de error. Ej.: "No puede confirmar un comprobante que no tenga movimiento contable".	
Poscondiciones:	Son actualizados los datos de los movimientos contables.

<b>Caso de Uso:</b>	<b>Desconfirmar Comprobante</b>
Actor(es):	Jefe Finanzas (inicia).
Propósito:	Desconfirmar un comprobante de operaciones para realizar una operación sobre el mismo.
Resumen:	El caso de uso inicia cuando el jefe de finanzas desea desconfirmar un comprobante y termina cuando lo desconfirma.
Referencias:	R2.
Precondiciones:	El comprobante debe estar confirmado previamente y el usuario debe estar autenticado como jefe de finanzas.
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
1. El jefe de finanzas decide desconfirmar un comprobante. 2. El jefe de finanzas entra los datos del comprobante en la pantalla 8.	1.1 El sistema muestra la pantalla 8 para que se busque el comprobante (ver CU: Buscar Comprobante). 2.1 El sistema comprueba que el comprobante a desconfirmar: ✓ Exista en la BD. ✓ Esté confirmado. 3. Desconfirma el comprobante. 4. Muestra mensaje de desconfirmación al usuario.
<b>Cursos Alternos</b>	
Si el comprobante seleccionado para desconfirmar no ha sido confirmado, el sistema muestra un mensaje de error.	
Poscondiciones:	El comprobante queda desconformado.

<b>Caso de Uso:</b>	<b>Buscar Comprobante</b>
Actor(es):	Jefe Finanzas, Contador (inicia).
Propósito:	Buscar uno o varios comprobantes de operaciones existentes pertenecientes a un Órgano Financiero.
Resumen:	El caso de uso inicia cuando el contador o jefe de finanzas desean

Capítulo 2 Descripción de la propuesta solución

	buscar un comprobante para realizar una acción y termina cuando se muestran los resultados de la búsqueda.
Referencias:	R11
Precondiciones:	El comprobante debe existir en la base de datos.
<b>Pantalla 8</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
	<ol style="list-style-type: none"> <li>1. El sistema muestra la pantalla 1</li> <li>2. El sistema le permite al actor realizar la búsqueda del comprobante por: <ul style="list-style-type: none"> <li>✓ Registro (valor único en la BD para un año y un Órgano financiero)</li> <li>✓ Tipo de operación</li> <li>✓ Fecha inicial</li> <li>✓ Fecha final</li> </ul> </li> </ol>
3. Introduce uno o todos los datos correspondientes a los campos (A), (B), (C), (D) de la pantalla 8 y presiona (E).	<ol style="list-style-type: none"> <li>3.1 El sistema debe: <ol style="list-style-type: none"> <li>3.1.1 Encontrar el o los valores en la BD.</li> <li>3.1.2 Muestra un grid(A), pantalla 2, con el o los valores encontrados.</li> <li>3.1.3 Si el usuario escoge en el menú (CU: Actualizar comprobante) la opción "Modificar Comprobante": <ul style="list-style-type: none"> <li>✓ El usuario puede seleccionar nada más un registro.(Chequear que si selecciona un registro y quiere seleccionar otro se desmarca el anterior)</li> </ul> </li> </ol> <p>Si el usuario escoge en el menú "Imprimir Comprobante"</p> <ul style="list-style-type: none"> <li>✓ El usuario puede seleccionar uno o varios registros.</li> </ul> </li> <li>3.2 Verificar que el comprobante no este confirmado porque una vez en este estado no se puede realizar ninguna acción sobre este.</li> <li>3.3 Si el dato corresponde al campo (C) verificar que corresponda al año en curso y si es del campo (D), verificar si corresponde al año en curso y si es menor que la fecha actual.</li> </ol>



## Capítulo 2 Descripción de la propuesta solución

---

	órgano a que pertenece el usuario en la pantalla 10: 2. No. de registro del comprobante (A). 3. Fecha del registro (B). 4. Estado de confirmación del registro (C). 5. Fecha en que se realizó la acción 'Elaborar Comprobante' (D). 6. Nombre del usuario que confeccionó el comprobante (E).
<b>Cursos Alternos</b> Si no hay comprobante seleccionado el sistema muestra un mensaje de error informando que debe seleccionar un comprobante para controlarlo.	
Poscondiciones:	Los datos del comprobante quedan controlados.

### 2.12 Conclusiones.

En este capítulo se comenzó a desarrollar la propuesta de solución, obteniéndose a partir del análisis de los procesos del negocio, un listado con las funciones que debe tener el sistema, que se representaron mediante un Diagrama de Casos de Uso, y finalmente se describieron paso a paso todas las acciones de los actores del sistema con los casos de uso con los que interactúan. Gracias a esto ahora se puede empezar a construir el sistema, tratando de que se cumplan todos los requerimientos y las funciones que han sido consideradas necesarias en este capítulo.



# Capítulo 3

Introducción

Análisis y Diseño

Mecanismos de Diseño

Diagramas de Clases del Diseño

Diseño de la Base de Datos

Modelo de Despliegue

Modelo de Implementación

Conclusiones



## CAPÍTULO CONSTRUCCIÓN DE LA PROPUESTA SOLUCIÓN

### 3.1 Introducción.

En este capítulo se detallaran los artefactos necesarios para la construcción de la propuesta solución, se transitará por una fase importantísima de la ingeniería del software en la cual se termina el análisis, el diseño, implementación del sistema, además de establecer patrones de arquitectura, de diseño y mecanismos de diseño que permiten obtener un producto de calidad.

### 3.2 Análisis y Diseño.

El objetivo del análisis y diseño en la construcción de un sistema es traducir los requisitos a una especificación que describe cómo implementar el sistema. El análisis 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. El diseño debe ser suficiente para que el sistema pueda ser implementado sin ambigüedades.

Antes de comenzar a realizar los diagramas que nos muestran la solución del sistema, urge definir algunos conceptos que nos ayudarán a obtener un producto de mejor calidad.

### 3.3 Mecanismos de Diseño.

RUP, define el flujo de análisis y diseño como el único en el que se comienzan actividades desde la fase de inicio, es un flujo en el que queda

definida la solución del sistema; indudablemente, esto provoca una inversión de esfuerzo en la modelación de los artefactos de esta fase, ya que se trata de hacer lo más eficiente posible, de manera que los modelos obtenidos sean comprensibles y permitan una comunicación efectiva.

Un elemento que puede ser favorable en este sentido son los *Mecanismos de Diseño*. Los mismos reportan beneficios para al menos 3 propósitos:

- ✓ Mantener la homogeneidad en el diseño.
- ✓ Reutilizar soluciones anteriormente probadas.
- ✓ Reutilizar documentación.

### 3.3.1 Mecanismo de diseño de Seguridad.

En las aplicaciones Web, es común que ciertos recursos (páginas), requieran de un control en el acceso, para lo cual, se emplea por lo general el mecanismo de crear variables de sesión, registrando en esta una variable indicando la identidad del usuario, una vez ejecutada satisfactoriamente la autenticación de este en el sistema.

Es bueno aclarar que nuestro sistema, como módulo de un ERP, obtiene las variables de sesión mencionadas a partir de otro módulo encargado de la seguridad y que ofrece esta posibilidad a través de un servicio Web.

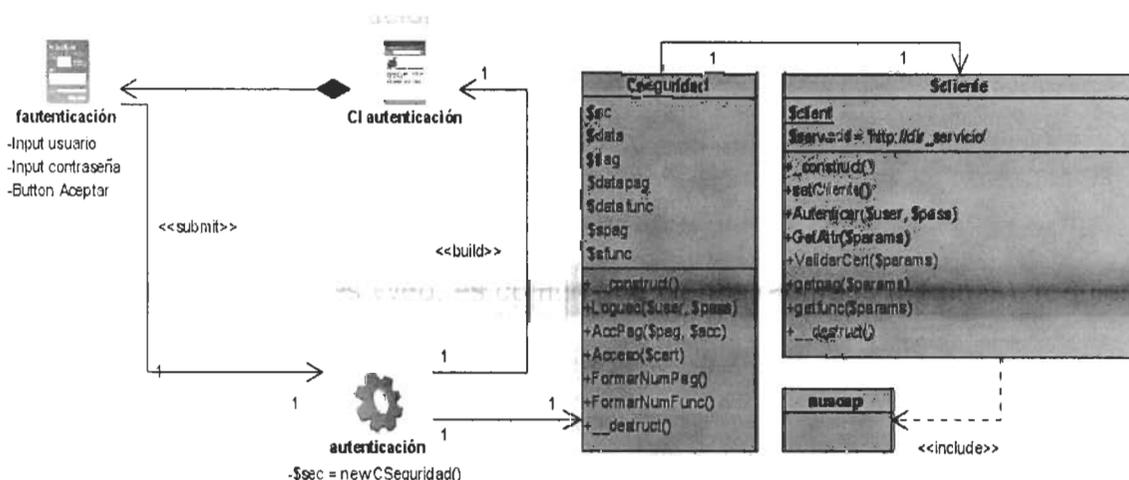


Figura 3.1 Diagrama de Clases del Mecanismo de Diseño de Seguridad

El mecanismo anterior constituye la parte cliente del servicio, con la que contarán todas las aplicaciones y mediante la cual se hará el acceso al componente servidor del servicio Web. La clase Scliente es la encargada de la comunicación con la parte servidora del servicio Web de seguridad. La clase Cseguridad es la intermediaria entre los sistemas y la clase Scliente, siendo transparente el servicio Web a los sistemas que lo usan. La clase nusoap incluye todas las clases necesarias para el funcionamiento del servicio en la parte cliente. Además de las clases principales se brinda la interfaz de autenticación, que es el elemento fundamental e inicio del mecanismo, así como la clase autenticación que es la que regula el proceso e instancia a la clase Cseguridad.

### **3.3.2 Mecanismo de diseño de Acceso a Datos.**

Este mecanismo surge por el mismo motivo que el de Seguridad, para acceder a los datos siempre estaban involucrados los mismos objetos y se efectuaban un conjunto de operaciones comunes en la mayoría de los casos de uso.

La vista estática de este mecanismo de acceso a datos muestra un conjunto de clases que interactúan para dar acceso y manipulación de los datos de la persistencia desde el nivel más bajo, es decir, utilizando los objetos nativos brindados por el entorno de desarrollo PHP como son PDO y PDOStatement, siguiendo así hasta la abstracción del acceso a datos a través de mEntidad de la cual heredan las clases particulares de nuestro sistema como Típicas y mSimples.

Para dar la responsabilidad a una clase que encapsulará las instancias de estos objetos se definió la clase FactoríaTípicas.

¿Qué es FactoríaTípicas?

Es el resultado de aplicar los patrones de diseño, los cuales son una solución a un problema de diseño no trivial que es efectiva y reusable; efectiva porque ya ha resuelto un problema similar anteriormente de manera satisfactoria y reusable

porque se puede aplicar a muchos problemas de diseño en diferentes circunstancias.

FactoríaTípicas, es una clase que está implementada siguiendo los conceptos del patrón Factoría, que es un patrón de diseño dentro del grupo de los patrones creacionales, que son soluciones aceptadas como "buena" a los problemas de creación de instancias de objetos.

¿Qué es el Patrón Factoría?

El patrón factoría es uno de los varios patrones creadores definidos por la GoF (Gans of Four). La idea que se esconde detrás de este patrón es la de centralizar el sitio donde se crean los objetos, normalmente donde se crean objetos de una misma "familia", sin dar una definición clara de lo que nuestro software puede entender como familia, como podría ser componentes visuales, componentes de la lógica del negocio, o objetos concurrentes en el tiempo.

La clase factoría devuelve una instancia de un objeto según los datos que se le pasan como parámetros. Para que la creación centralizada de objetos sea lo más "útil y eficaz" posible, es de esperar que todos los objetos creados descendan de la misma clase o implementen el mismo interface (es decir, hagan una operación similar pero de distintas formas), así podemos usarlos todos de la misma manera, con los mismos métodos (gracias al polimorfismo), sin importarnos que clase concreta estamos tratando en cada momento.

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

¿Qué es Típicas?

Son clases que surgen también de la aplicación de patrones, en este caso, patrones de arquitectura y dentro de estos el patrón TABLE DATA GATEWAY, que tiene como supuesto realizar una clase para instanciar cada tabla de la BD.



## Capítulo 3 Construcción de la propuesta solución

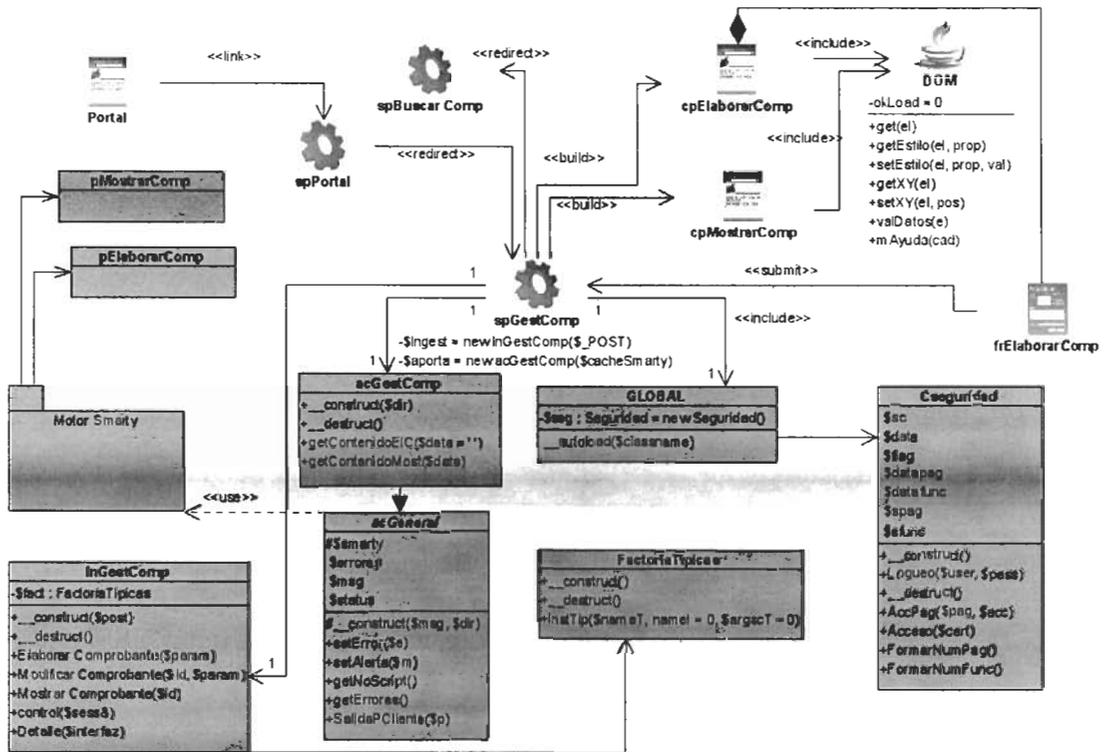


Figura 3.3 Diagrama de Clases del CU: Gestionar Comprobante.

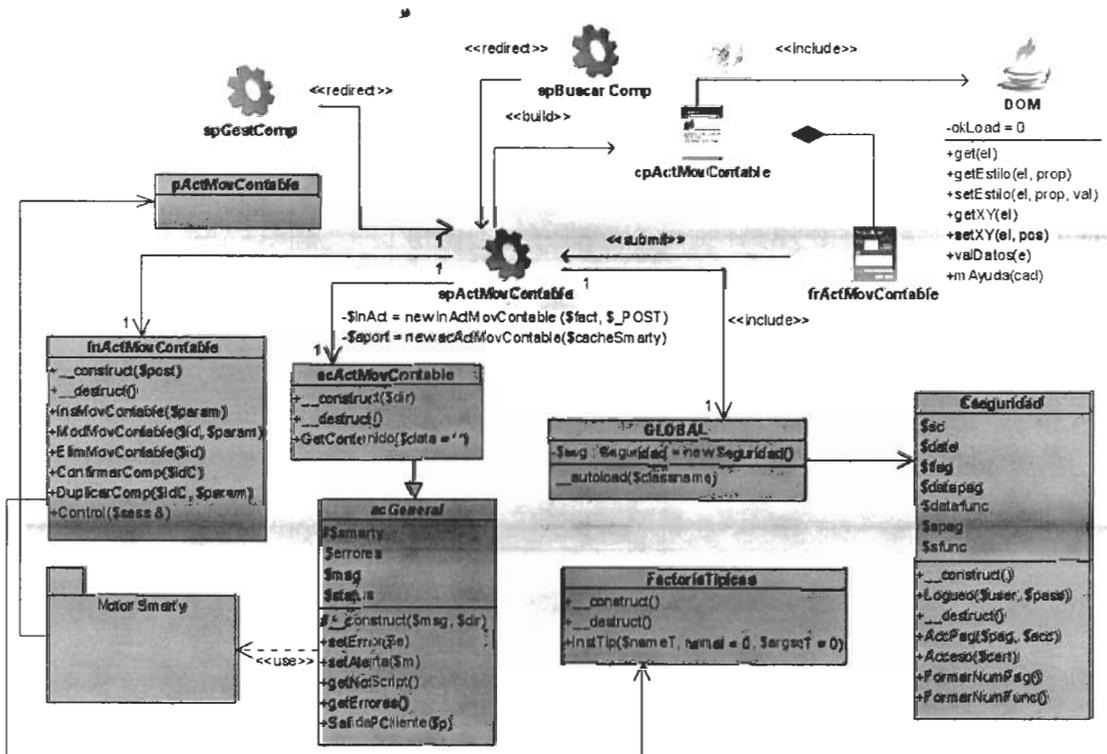


Figura 3.4 Diagrama de Clases del CU: Actualizar Movimientos Contables

## Capítulo 3 Construcción de la propuesta solución

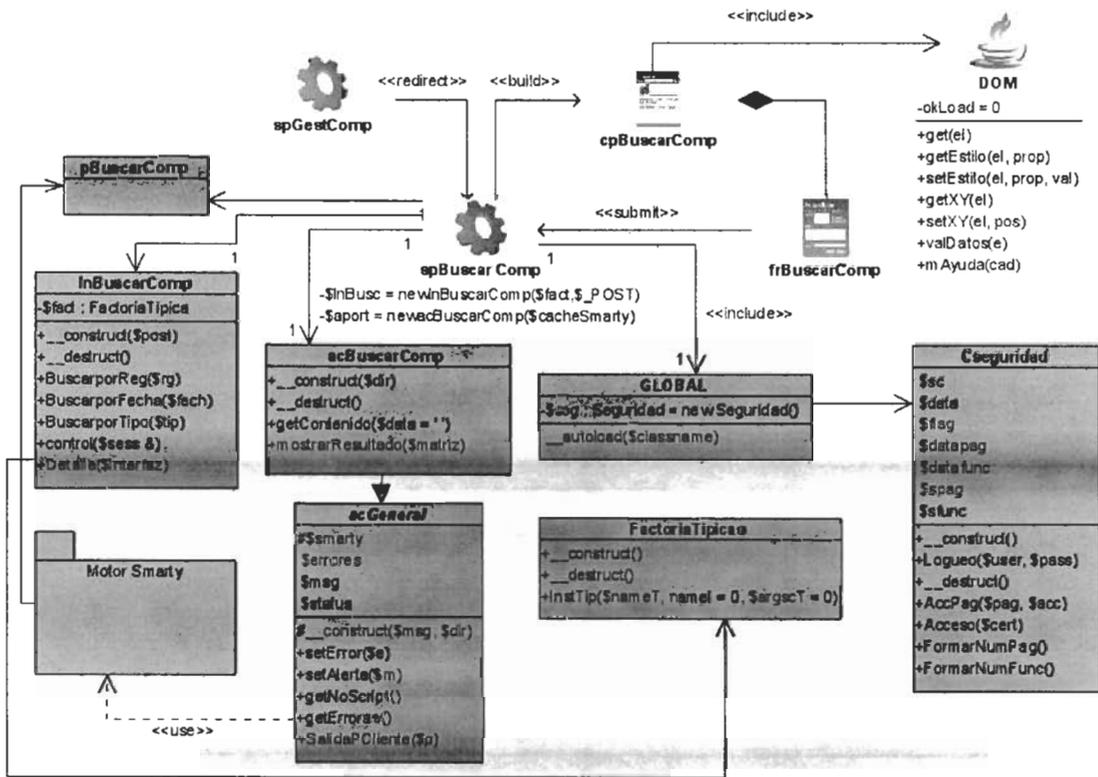


Figura 3.5 Diagrama de Clases del CU: Buscar Comprobante

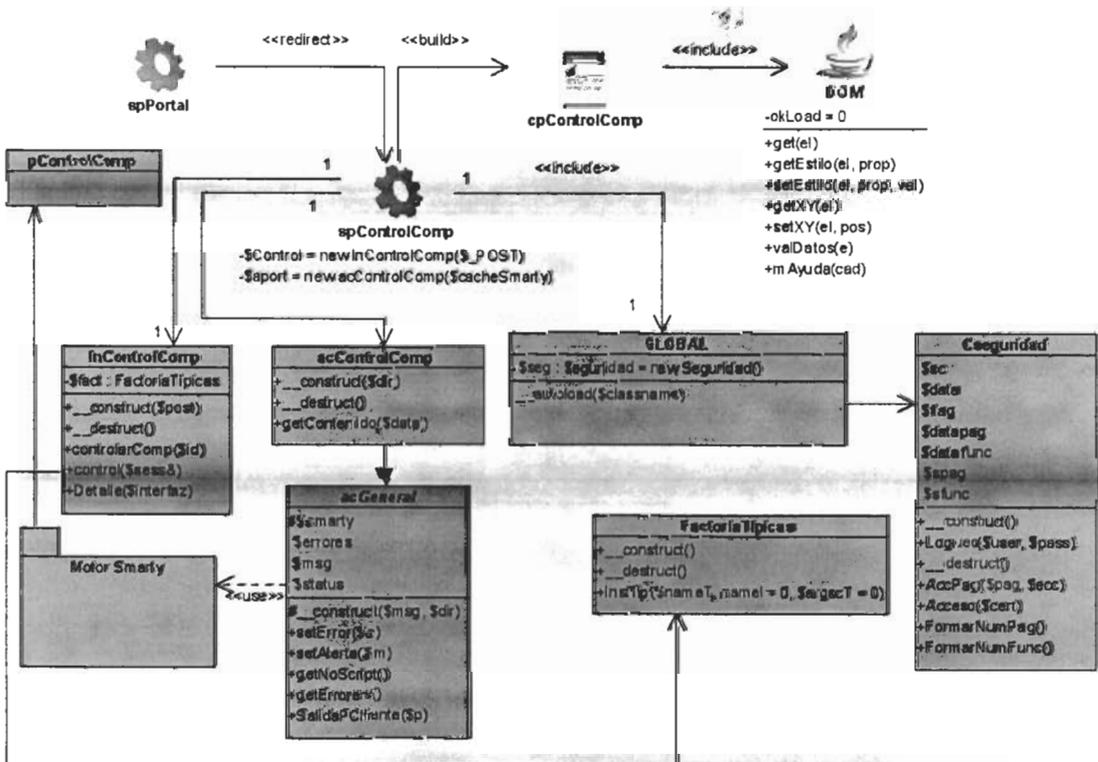


Figura 3.6 Diagrama de Clases del CU: Control Comprobante

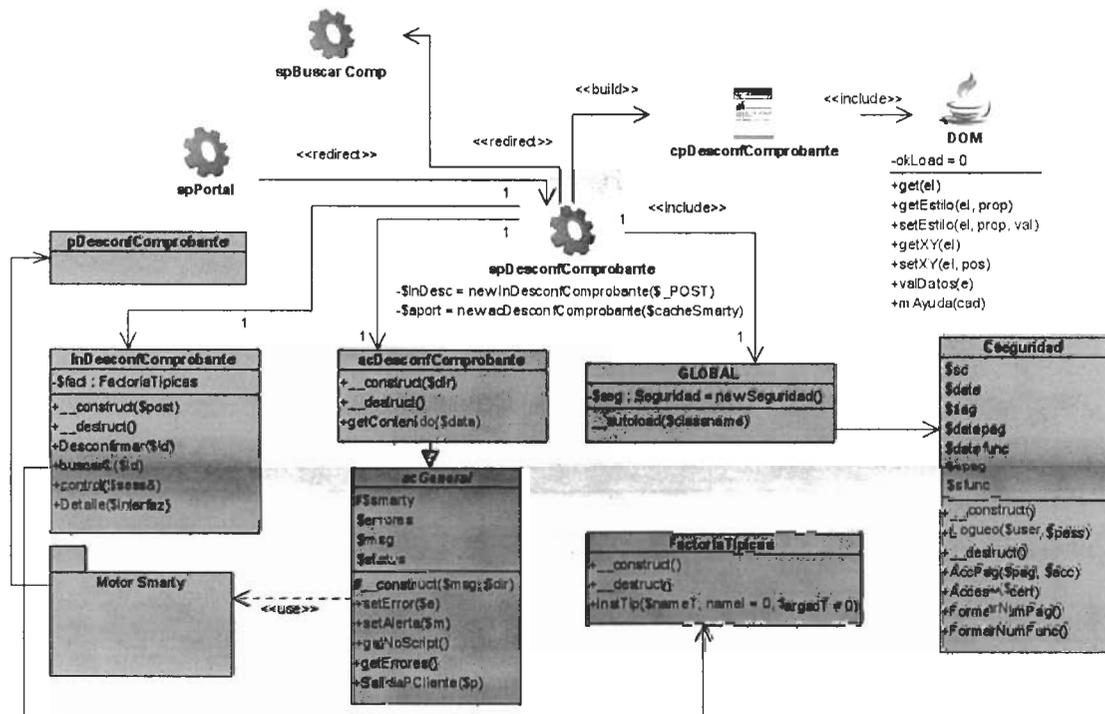


Figura 3.7 Diagrama de Clases del CU: Desconfirmar Comprobante

¿Cómo está compuesto un diagrama de clases de diseño de nuestro sistema?

Por cada caso de uso tenemos:

- ✓ Una clase de lógica de negocio (*InNombreCU*) por cada caso de uso.
- ✓ Una clase aportadora de contenido para todas las páginas clientes involucradas (*acNombreCU*), que contiene métodos como *getInPaginaCliente1*, *getInPaginaClienteN*, donde N es la cantidad de páginas clientes asociadas al CU.
- ✓ Una clase *global* que centraliza a los objetos principales para manejar la seguridad y el acceso a datos a través de las clases *Cseguridad* y *FactoriaTipicas*.
- ✓ Un paquete *Smarty* que gestiona a las plantillas, donde tendrá asociado tantas plantillas como páginas clientes tenga el CU.

## Capítulo 3 Construcción de la propuesta solución

---

- ✓ La página servidora tendrá las ideas de los objetos *InNombreCU* (lógica de negocio) y *acNombreCU* (aportadora de contenido).
- ✓ Las clases *Cseguridad* y *FactoríaTípicas* son representadas en estos diagramas como parte del mecanismo de diseño y están explicadas en sus respectivos epígrafes.

### 3.5 Diseño de la Base de Datos.

Uno de los pasos cruciales en la construcción de una aplicación que maneje una base de datos, es sin duda, el diseño de la base de datos. Si las tablas no son definidas apropiadamente, podemos tener muchos dolores de cabeza al momento de ejecutar consultas a la base de datos para tratar de obtener algún tipo de información.

Para diseñar la base de datos del sistema, se utiliza el diagrama de clases persistentes y el modelo de datos, que están basados en la modelación de las clases del epígrafe anterior. Algunas de las clases representaban los datos que se obtienen y almacenan durante los procesos de la aplicación, estos son lo que pueden modelarse a través de un diagrama de clases persistentes, lo que permitirá ver la relación entre los datos, y completará la modelación de la lógica de negocio de la aplicación.

El Diagrama de Clase Persistentes se puede usar para modelar la estructura lógica de la base de datos, con clases representando tablas, y atributos de clase representando columnas.

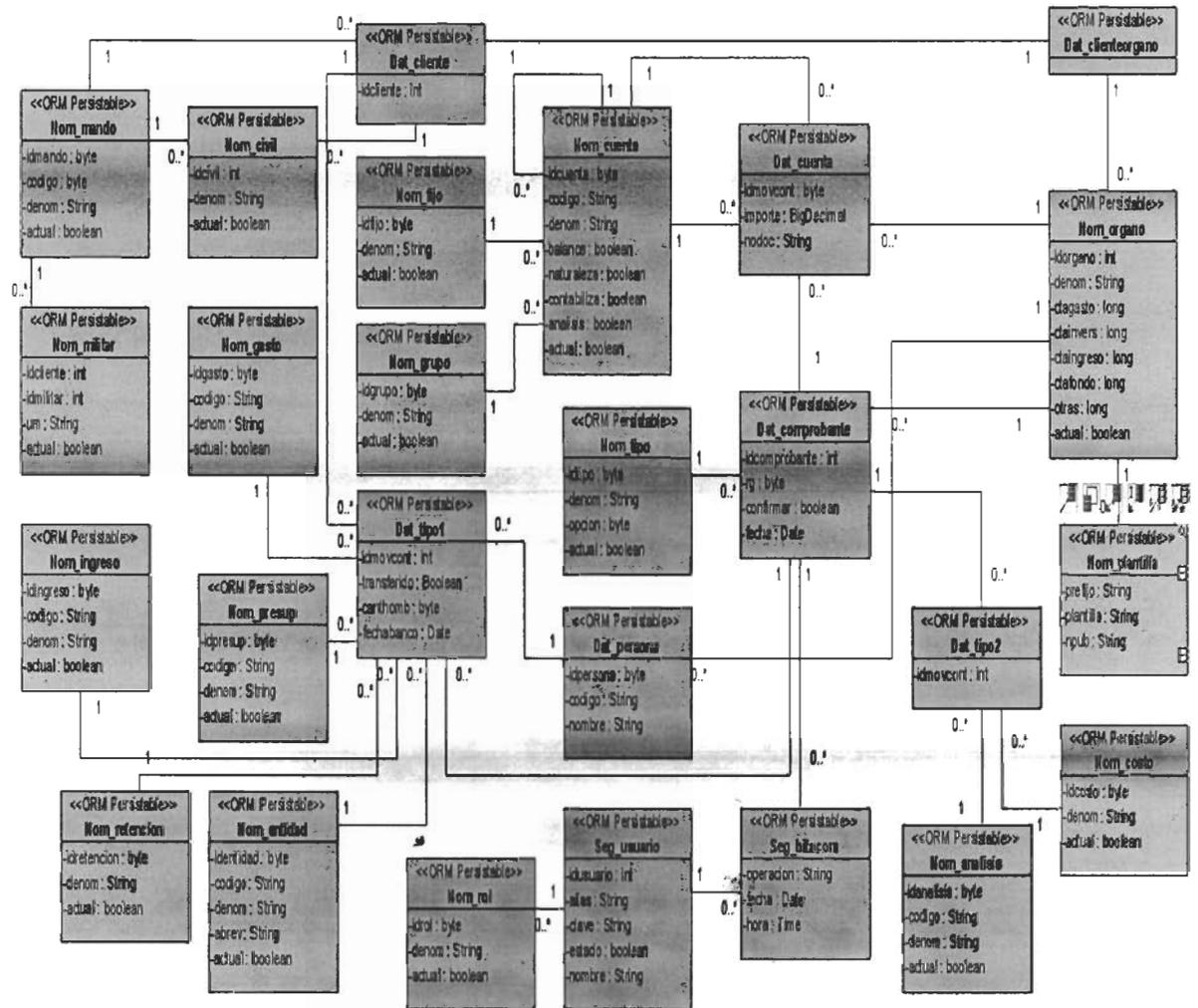


Figura 3.8 Diagrama de Clases Persistentes

El modelo de datos tiene como objetivo “dotar de significado” a las estructuras de datos, carentes del mismo, en el modelo de persistencia.



### Capítulo 3 Construcción de la propuesta solución

No.	Nombre de la tabla	Clasificación	Descripción
1	dat_cliente	Datos	datos comunes para los clientes
2	dat_clienteorgano	Datos	Tabla para relacionar los clientes que tiene un órgano financiero
3	dat_comprobante	Datos	Se guarda para un registro y el órgano a que pertenece, la fecha y el tipo de operación.
4	dat_cuenta	Datos	se registran los movimientos contables asociados a un registro y a un órgano financiero
5	dat_persona	Datos	listado de las personas que tiene retención al banco
6	dat_tipo1	Datos	Tabla que se utiliza para tipo 1
7	dat_tipo2	Datos	Tabla que se utiliza para un análisis de los hospitales militares
8	nom_análisis	nomenclador	Nomenclador de análisis de gasto para hospitales militares
9	nom_civil	nomenclador	Cientes Civiles acreedores o proveedores
10	nom_costo	nomenclador	Nomenclador de Centros de Costo
11	nom_cuenta	nomenclador	nomenclador de cuentas contables
12	nom_entidad	nomenclador	Nomenclador de Entidad o Especialidad
13	nom_fijo	nomenclador	Nomenclador de Activos Fijo
14	nom_gasto	nomenclador	Nomenclador de Gastos en las FAR
15	nom_grupo	nomenclador	Nomenclador de grupo de cuentas
16	nom_ingreso	nomenclador	Nomenclador de ingresos al presupuesto del estado
17	nom_mando	nomenclador	Nomenclador de mandos
18	nom_militar	nomenclador	Nomenclador de militar
19	nom_organo	nomenclador	Nomenclador de Organos Financieros
20	nom_plantilla	nomenclador	Nomenclador de plantilla
21	nom_presup	nomenclador	Nomenclador de Grupos Presupuestarios
22	nom_retencion	nomenclador	nomenclador de retención bancaria
23	nom_rol	nomenclador	roles de los usuarios
24	nom_tipo	nomenclador	Nomenclador de Tipo de Operaciones
25	seg_bitacora	seguridad	Tabla para controlar comprobantes
26	seg_usuario	seguridad	Tabla para controlar usuarios.

**Relación de las tablas que componen la BD**

### 3.6 Principios de Diseño.

El diseño, sea cual sea el objeto del mismo, tiene que basarse en el usuario final del sistema, muchos de ellos no poseen una preparación en las cuestiones de la informática. Para contrarrestar esto existen los llamados Principios de Diseño, que son aplicables en la arquitectura, la ingeniería y, por supuesto, las páginas y aplicaciones Web, entre otros campos de aplicación.

Este sistema utiliza ciertos principios generales entre los que encontramos:

1. Principio de uso equiparable: donde las características de privacidad, garantía y seguridad estén igualmente disponibles para todos los usuarios, y que el diseño sea atractivo para todos los usuarios.
2. Principio de la flexibilidad: donde se ofrezcan posibilidades de elección en los métodos de uso, que facilite al usuario la exactitud y precisión, y se adapte al paso o ritmo del usuario.
3. Principio de la Información perceptible: donde se usen diferentes modos para presentar de manera redundante la información esencial (gráfica y verbal), se proporcione contraste suficiente entre la información esencial y sus alrededores, se amplíe la legibilidad de la información esencial, y que diferencie los elementos en formas que puedan ser descritas (por ejemplo, para las funciones de catalogación).
4. Principio de tolerancia al error: donde se dispongan los elementos para minimizar los riesgos y errores, por ejemplo utilizando elementos comunes; y los elementos peligrosos eliminados, aislados o tapados, que se proporcionen advertencias sobre peligros y errores. Hay que posibilitar el descubrimiento interactivo y el aprendizaje ensayo-error, y posibilitar la reversibilidad y la recuperabilidad de las acciones.
5. Principio de esfuerzo de acceso y uso: que minimicen las acciones repetitivas, y que proporcione una línea de visión clara hacia los elementos importantes tanto para un usuario sentado como de pie.

En aras de alcanzar un software agradable al usuario se definieron estándares a seguir durante la construcción del mismo:

### 3.6.1 Estándares de diseño.

La página principal de la aplicación, se concibe como un portal, con un menú, que no debe exceder de 3 niveles de profundidad, donde se agrupan las funcionalidades del sistema.

Las páginas deben tener una cabecera (banner) representativa, un área de trabajo y una barra menú con las opciones, además tener una hoja de estilo en común para lograr la uniformidad, es decir, se trabajará con la familia de fuentes Arial Helvética, Sans-Sheriff, el tamaño de la misma no debe diferir mucho de 11Px.

Los colores con los que se trabajarán serán tonalidades claras basadas en el verde, azul y amarillo, combinados con el color blanco o gris.

### 3.6.2 Estándares de codificación.

A continuación se muestra el estándar de codificación a seguir para implementar el sistema.

<b>Apariencia de clases y objetos</b>	Primera letra en mayúscula	Los nombres de las clases y las instancias de las mismas deben comenzar con la primera letra en mayúscula y el resto en minúscula, en caso de que sea un nombre compuesto se empleará notación PascalCasing*. Ejemplo: MiClase ().
<b>Nombre de clases y objetos</b>	Relacionados al propósito	El nombre empleado, debe permitir que con sólo leerlo se conozca el propósito de la clase o instancia de la misma. Para el caso de las instancias es recomendable que se denoten así: Para la clase: Nomcliente su instancia será \$Ocliente, de forma tal que la primera letra

### Capítulo 3 Construcción de la propuesta solución

		indique que es un objeto y el resto, la clase a la que pertenece.
<b>Apariencia de atributos</b>	Primera letra en minúscula	El nombre que se le da a los atributos de las clases debe comenzar con la primera letra en minúscula, en caso de que sea un nombre compuesto se empleará notación CamelCasing**.
<b>Nombre de atributos</b>	Nemotécnicos	El nombre empleado, debe permitir que con sólo leerlo se conozca el propósito del mismo dentro de la clase. Ejemplo: \$nTabla, este atributo denota el nombre de una tabla.
<b>Apariencia de las funciones</b>	Primera letra en mayúscula	Los nombres de las funciones deben comenzar con la primera letra en mayúscula y el resto en minúscula, en caso de que sea un nombre compuesto se empleará notación PascalCasing*. Si son funciones que obtienen un dato se emplea el prefijo <code>get</code> y si fijan algún valor se emplea el prefijo <code>set</code> .
<b>Nombre de las funciones</b>	Nemotécnicos	El nombre empleado, debe permitir que con sólo leerlo se conozca el propósito de la misma dentro de la clase.
<b>Declaración de parámetro en funciones</b>	Agrupados por tipos primero los string, los numéricos y valores por defecto.	Los parámetros que se le pasan a las funciones se recomienda sean declarados de forma tal que estén agrupados por el tipo de dato que contienen. Ejemplo: BuscaUnidad (\$nTabla(string), \$nCampos(string), \$kIndice(entero)).
<b>Variables y constantes</b>		
<b>Apariencia de constantes</b>	Todas sus letras en mayúscula	Se deben declarar las constantes con todas sus letras en mayúscula.
<b>Nombres de las variables y constantes</b>	Nemotécnicos	El nombre empleado, debe permitir que con sólo leerlo se conozca el propósito de la misma. Ejemplo: \$nFields.
<b>Declaración de constantes y asignación a variables</b>	Una por cada línea	Se recomienda declarar una constante por cada línea y con las asignaciones a las variables sucede lo mismo. Ejemplo: define("CONSTANT1", "value1"); define("CONSTANT2", "value2"); \$nTabla='nomproducto'; \$kIndice=0;
<b>Indentación</b>		
<b>Objetivo:</b> Lograr una estructura uniforme para los bloques de código así como para los diferentes niveles de anidamiento.		
<b>0 espacios en</b>	Require	No se empleará ningún espacio en blanco

<b>blanco desde la izquierda en</b>	Include Class	desde la izquierda para las instrucciones antes mencionadas. Se tomará como inicio de la página el tag PHP <?
<b>2 espacio en blanco desde la izquierda en</b>	Function Define	Se dejarán dos espacios en blanco desde la izquierda en las instrucciones antes mencionadas.
<b>2 espacio en blanco desde la referencia en</b>	Inicio y fin de bloque	Se recomienda dejar dos espacios en blanco desde la instrucción anterior para el inicio y fin de bloque {}. Lo mismo sucede para el caso de las instrucciones If, else, For, While, Do While, Switch, Foreach.
<b>Niveles de anidación</b>	Hasta 5 niveles	Se recomienda emplear hasta 5 niveles de anidación en instrucciones If, For, While.
<b>Ejemplo de indentación</b>		
<pre> &lt;? require ('class/Interface.php');  class MiClase {     function BuscaUnidad(\$nTabla, \$nFields, \$kIndice)     {         if (\$nTabla)         {             ...         }         for (...)         {             ...         }     } } ?&gt; </pre>		
<b>Comentarios, separadores, líneas y espacios en blanco</b>		
<b>Ubicación de comentarios</b>	Al inicio de cada clase o función y al final de cada bloque de código.	Se recomienda comentar al inicio de la clase o función especificando el objetivo de la misma así como los parámetros que usa (especificar tipos de dato, y objetivo del parámetro) entre otras cosas. Y se comenta también cuando se cierran los ciclos, clases, instrucciones if y otras.
<b>Separador de instrucciones</b>	Se emplea el punto y coma.	Se recomienda usar el separador al final de cada instrucción y no en la línea de abajo. Ejemplo: define ("CONSTANT", "value1");
<b>Líneas en blanco</b>	Se emplean antes de cada función.	Se recomienda dejar una línea en blanco antes de la definición de cada función para

		dar claridad al código.
<b>Espacios en blanco</b>	Entre operadores lógicos y aritméticos.	Se recomienda usar espacios en blanco entre estos operadores para lograr una mayor legibilidad en el código. Ejemplo: \$nTabla = 'nomproducto'; if ((\$nTabla) && (\$nFields))

### 3.6.3 Estándares para la BD.

<b>Apariencia de la BD</b>	Primera letra en mayúscula	Los nombres de las BDs deben comenzar con la primera letra en mayúscula y el resto en minúscula, en caso de que sea un nombre compuesto se empleará notación PascalCasing*.
<b>Nombres de las BDs</b>	Nemotécnicos y relacionados al propósito.	El nombre empleado, debe permitir que con sólo leerlo se conozca el propósito de la misma.
<b>Apariencia de los esquemas</b>	Todas las letras en minúscula.	El nombre a emplear para los esquemas debe escribirse con todas las letras en minúscula para evitar problemas con el Case Sensitive del gestor. Ejemplo: create schema 'finanzas';
<b>Nombres de los esquemas</b>	Nemotécnicos y relacionados al propósito.	El nombre empleado, debe permitir que con sólo leerlo se conozca el propósito del mismo.
<b>Apariencia de las tablas</b>	Todas las letras en minúscula.	El nombre a emplear para las tablas debe escribirse con todas las letras en minúscula para evitar problemas con el Case Sensitive del gestor. Ejemplo: create table 'nom_producto';
<b>Nombres de las tablas</b>	Nemotécnicos y relacionados al propósito. Además clasificando las tablas por su tipo.	El nombre empleado, debe permitir que con sólo leerlo se conozca el propósito del mismo. Se deben clasificar las tablas por su tipo, es decir por los datos que contienen se le coloca un prefijo, que se puede clasificar en: Ejemplo: Nomencladores nom ...

		<p>Auxiliares aux_...  Datos dat_...  Históricas his_...  Seguridad seg_...  Temporales tmp_...  Configuración cfg_...</p>
<b>Apariencia de los campos</b>	Todas las letras en minúscula.	<p>El nombre a emplear para los campos debe escribirse con todas las letras en minúscula para evitar problemas con el Case Sensitive del gestor.  Ejemplo:  add field 'idproducto';</p>
<b>Nombre de los campos</b>	Nemotécnicos En caso de identificadores, emplear id, este sería igual en la tabla de datos que lo emplea.	<p>El nombre empleado, debe permitir que con sólo leerlo se conozca el propósito del mismo. Además se debe incluir un comentario en la descripción del mismo.</p>
<b>Nombre de las llaves primarias</b>	Nemotécnicos empleando prefijos.	<p>Se nombrarán las llaves primarias de forma que se vea de qué tabla es y que es primaria. Ejemplo:  pk_cuenta. (Llave primaria de la tabla cuenta). Si es una llave compuesta se coloca el prefijo y en nemotécnico los campos que la forman.</p>
<b>Nombre de las llaves foráneas.</b>	Nemotécnicos empleando prefijos.	<p>Se nombrarán las llaves foráneas de forma que se vea de qué tabla es y que es foránea. Ejemplo:  fk_cuenta. (Llave foránea de la tabla cuenta). Si es una llave compuesta se coloca el prefijo y en nemotécnico los campos que la forman.</p>
<b>Nombre de las secuencias</b>	Nemotécnicos empleando prefijos.	<p>Se nombrarán las secuencias de forma que se vea de qué campo es y que es una secuencia. Ejemplo:  seq_idcuenta. (Secuencia del campo idcuenta).</p>
<b>Restricciones Únicas y de Chequeo</b>	Nemotécnicos empleando prefijos.	<p>Ejemplo:  (u_ o c_) + nombre del campo que la emplea.</p>

<b>Nombres de las funciones, triggers, y vistas</b>	Prefijos + Nemotécnicos	El nombre empleado, debe permitir que con sólo leerlo se conozca el propósito del mismo. Ejemplo: ft_ Funciones de triggers.
---	----------------------------	--

### 3.6.4 Tratamiento de Errores.

Los errores se tratan en una página especial que incluye el fichero de configuración general, y está preparada para recoger el número del error y presentar la pantalla con el error que le corresponde a ese código. En algunos casos incluye la forma de solucionar el error, como es el caso de la sesión, y la autenticación de los usuarios.

Algunos errores serán generados por funciones JavaScript para evitar la ejecución de la página en vano. Este es el caso de los formularios de inserción/actualización, y las eliminaciones, se utilizan los errores en forma de mensajes de texto, como alerts de JS, en la misma página donde se ejecutó la acción, de forma que el usuario pueda corregir más fácilmente y continuar, estos mensajes se generan donde se controla el error, solo que viajan entre las clases hasta llegar a la aportadora de contenido que es la que los mezcla finalmente con la salida del usuario. En operaciones muy largas o complicadas, se permite volver atrás, para revisar o modificar la información. Y se utilizan mensajes de confirmación, para acciones que son irreversibles como es el caso de las eliminaciones.

### 3.7 Modelo de despliegue.

El diagrama de despliegue nos representa la arquitectura de tiempo de ejecución de los procesadores, dispositivos y los componentes de software que se ejecutan en esa arquitectura. Es la descripción física última de la topología del sistema y describe la estructura de las unidades de hardware y el software que se ejecuta en cada unidad.

## Capítulo 3 Construcción de la propuesta solución

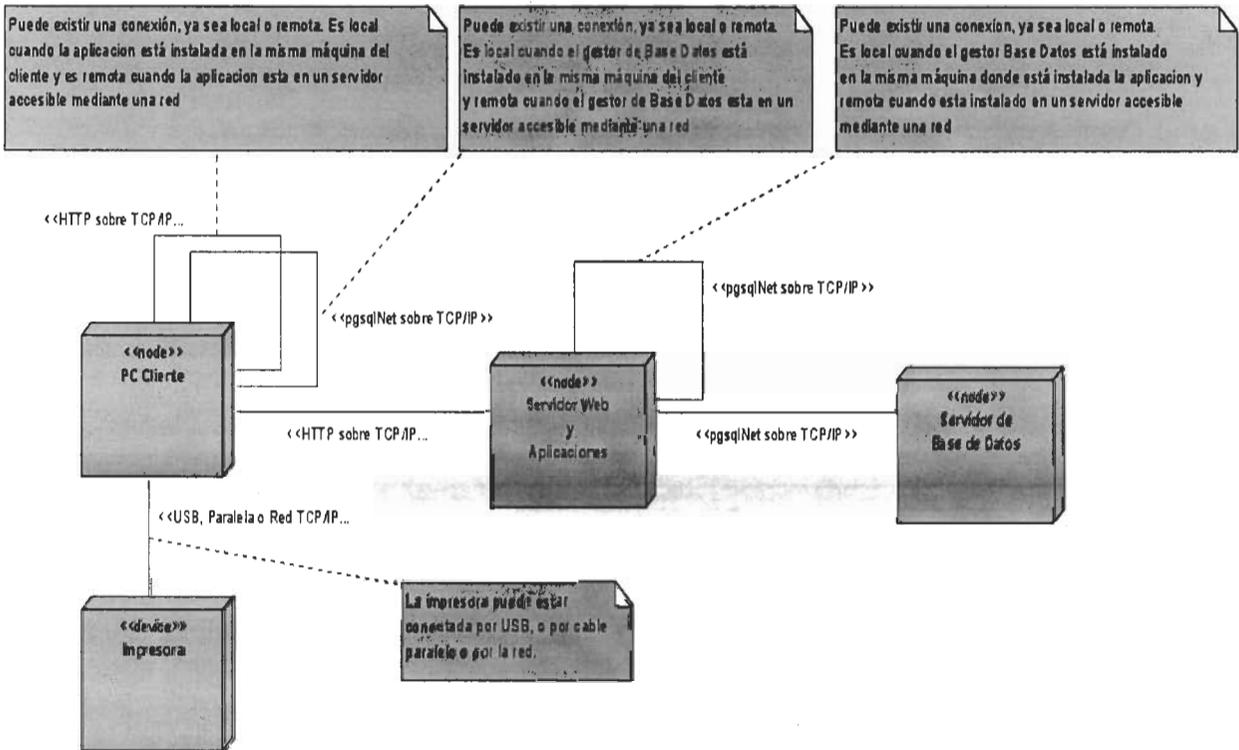


Figura 3.10 Diagrama de Despliegue del sistema

El diagrama mostrado en la figura anterior representa un modelo ideal que se corresponde con la arquitectura cliente/servidor en tres capas concebida para el sistema, es decir, el usuario final tendrá a su disposición un servidor para las aplicaciones Web, otro que sirva de base de datos y una computadora para visualizar las operaciones sobre ambos servidores.

¿Por qué entonces las líneas del primer nodo y del segundo incidiendo sobre ellos mismos?

Supongamos que la entidad a la que va destinada el producto solo tenga un servidor, entonces en este estarían la BD y la aplicación juntos, lo cual provoca que el segundo nodo del diagrama tenga una línea sobre sí, indicando esta situación; también puede suceder que el cliente no tenga la posibilidad de poseer servidores y que solo tenga una máquina, en ese caso, sucede lo

representado por el primer nodo con dos líneas sobre si mismo, es decir, que la computadora del cliente cumpla las tres funciones simultáneamente.

### 3.8 Modelo de implementación.

El diagrama de componentes describe los componentes de software y sus dependencias entre sí, representando la estructura del código. Los componentes son la implementación en la arquitectura física de los conceptos y la funcionalidad descrita en la arquitectura lógica (clases, objetos, relaciones y colaboraciones), son típicamente los archivos de implementación en el ambiente de desarrollo.

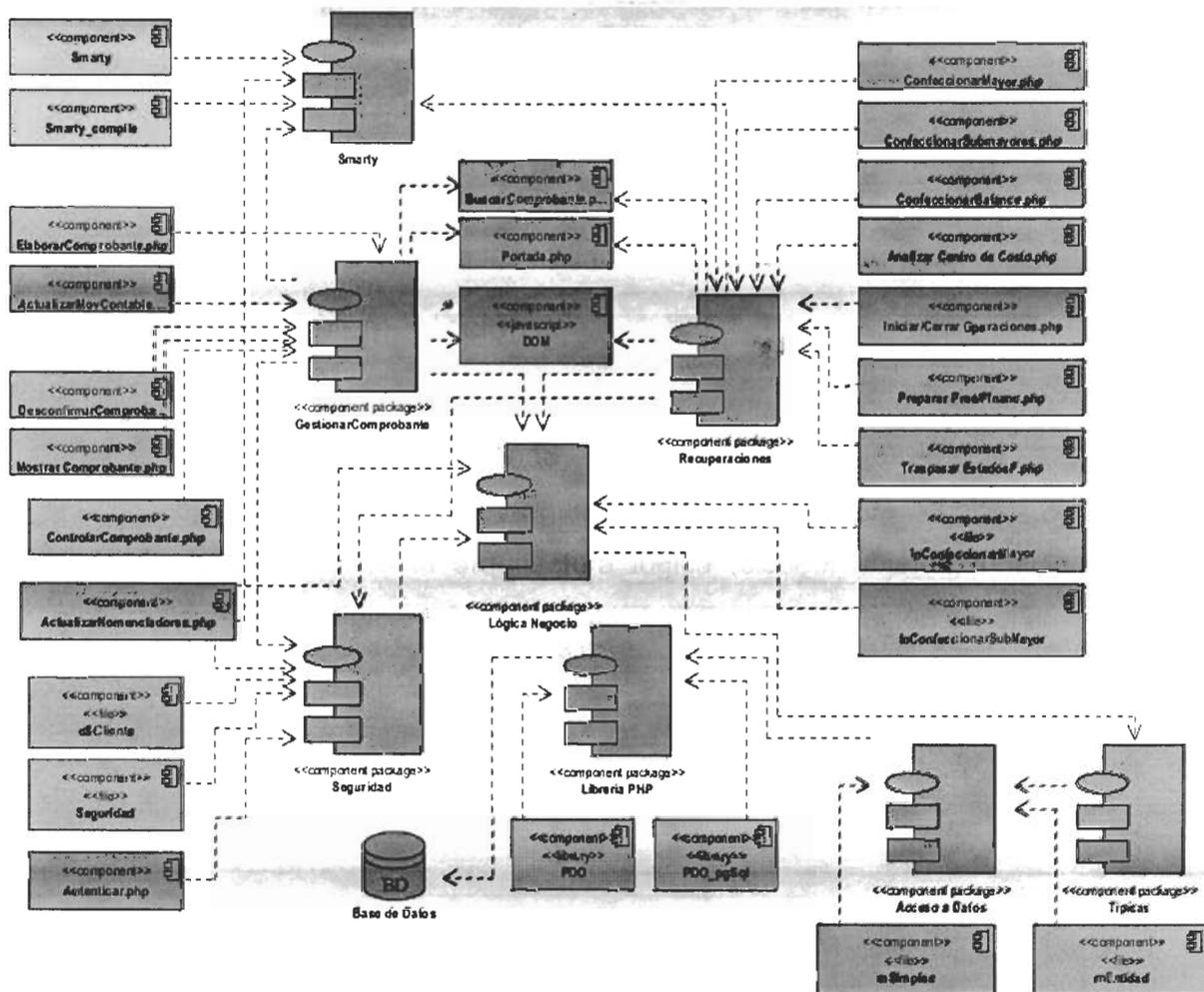


Figura 3.11 Diagrama de componentes del sistema

Es importante destacar que como se utilizó Smarty, cada página .php, tiene asociado una plantilla .tpl; cada caso de uso tiene 1 o más plantillas para su solución y en dependencia de la cantidad existente serán los componentes que lo compongan, la figura es el resultado de lo expuesto.

PDO y PDO\_pgSql, son componentes de tiempo de enlace, es decir, son librerías que surgen como resultado de compilar uno o más componentes de tiempo de compilación.

En la siguiente figura se muestran los componentes asociados al paquete “Lógica de Negocio”, estos no fueron incluidos en el diagrama anterior con el objetivo de lograr una mejor visibilidad y entendimiento del mismo.

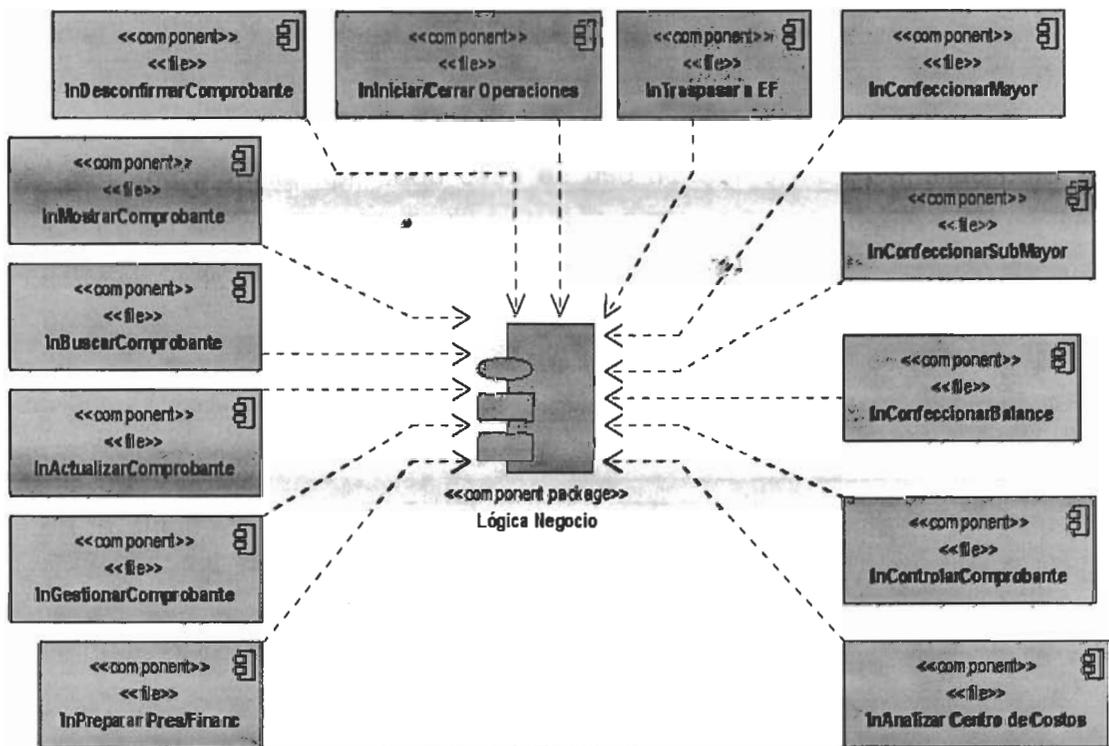


Figura 3.12 Componentes que pertenecen al Paquete Lógica de Negocio

La figura 3.13 muestra los componentes asociados al paquete “Típicas”; en secciones anteriores se mencionaba que por cada tabla existente en la BD existe una clase típica que manipula sus datos; en este paquete también encontramos la clase FactoríaTípicas, explicada previamente en este capítulo.

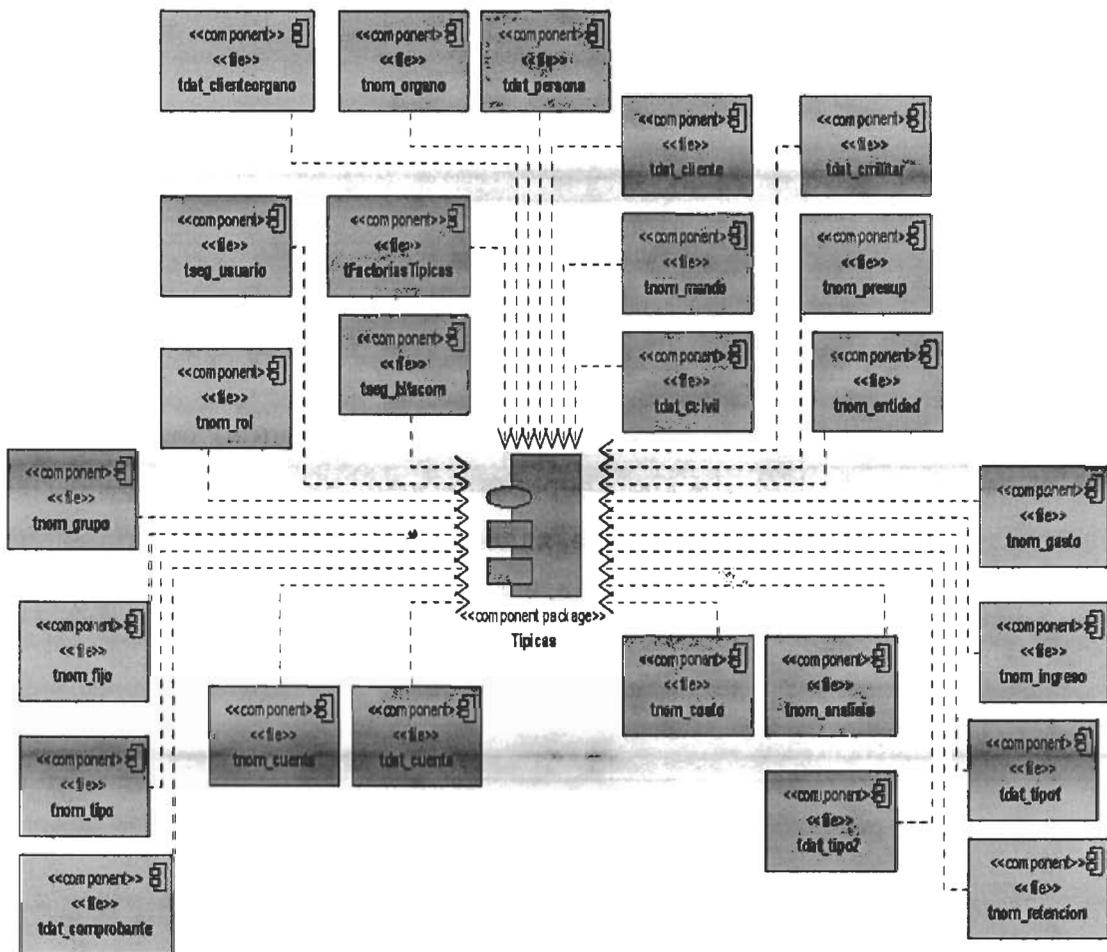


Figura 3.13 Componentes que pertenecen al paquete Típicas

### 3.9 Conclusiones.

En este capítulo se mostraron varios artefactos para llevar a cabo el proceso de construcción del sistema. Igualmente se identificaron otras funcionalidades que se pueden tener en cuenta para futuras versiones del sistema. Se utilizaron diagramas de clases Web para explicar la lógica del negocio del sistema, se crearon mecanismos de

### Capítulo 3 Construcción de la propuesta solución

---

diseño que simplifican el modelado y se diseñaron las clases persistentes que permiten hacer el diagrama de entidad-relación, en el sistema de gestión de bases de datos que se utilizará. En este momento, ya se tiene confeccionada completamente la propuesta que trae este trabajo.



# Capítulo 4

Introducción

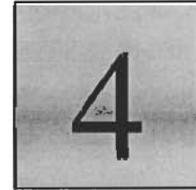
Planificación

Estimación de Esfuerzo y Costos

Beneficios Tangibles e Intangibles

Análisis de Costos y Beneficios

Análisis de Sostenibilidad



## CAPÍTULO ESTUDIO DE FACTIBILIDAD

### 4.1 Introducción.

Todo producto informático está sometido a una variable que constantemente lo amenaza: “el tiempo”, mas existen otros de igual importancia sin los cuales no se podría pensar en un producto y entre estos encontramos “los recursos”.

Si los analistas y desarrolladores de sistemas contaran con tiempo y recursos indefinidos, los productos saldrían con una calidad envidiable y serían “perfectos” funcionalmente porque las tendrían todas consigo para corregir errores que suelen cometerse durante la construcción de un software; resulta lamentable que no pueda ser así. ¿Por qué? Todo proyecto tiene definido una fecha de entrega y en aras de cumplir con esta, los analistas y desarrolladores, regidos por una planificación que abarca todas las etapas por las que transita un software, ponen todo su empeño.

En este capítulo se muestra la estimación de tiempo y costo para nuestro sistema al utilizar las técnicas de estimación por casos de uso, así como la planificación concebida para el desarrollo del mismo.

### 4.2 Planificación.

La planificación es el primer paso en la realización de un proyecto informático y en gran parte el responsable del éxito o fracaso del mismo; es el momento que tiene el jefe de proyecto para determinar los objetivos o metas del producto, seleccionar las estrategias de trabajo, determinar el curso de acción, tomar decisiones, preparar presupuestos, identificar y agrupar las tareas requeridas,

seleccionar y establecer la estructura organizativa, definir responsabilidades y autoridad, entre otras actividades que permiten definir la planificación como el proceso donde se planifica, organiza y controla un proyecto.

Nuestro equipo de trabajo está estructurado jerárquicamente como muestra la figura 4.1, compuesta por un jefe general de proyecto, jefes intermedios que tienen asignados un personal de trabajo, en este equipo la comunicación es informal, todos están involucrados en las decisiones pero es el jefe principal quien toma la decisión final.

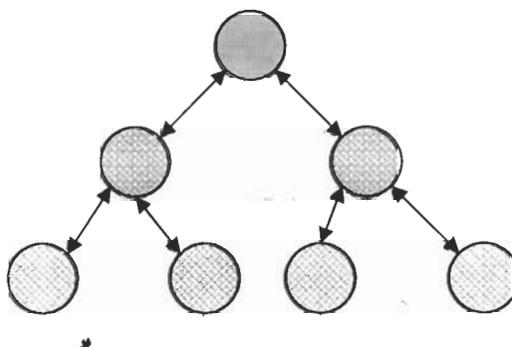


Figura 4.1 Estructura jerárquica del equipo de trabajo

La asignación de tareas y recursos en relación con el tiempo planificado previamente se puede constatar en la figura 4.2 y 4.3 (ver Anexos 3 y 4), elaborada con el Microsoft Project.

### 4.3 Estimación de esfuerzo y costos.

La utilización de metodologías tradicionales para la estimación de proyectos software ha resuelto la necesidad de conocer la duración de un proyecto como una variable dependiente de los recursos a utilizar. Se propone en este trabajo el uso de la técnica de estimación basada en casos de usos.

**Paso 1: Cálculo de los puntos de casos de uso desajustados.**

La fórmula propuesta es  $UUCP = UAW + UUCW$

Donde:

UUCP: Puntos de caso de uso desajustados.

UUCW: Factor de peso de los casos de uso sin ajustar.

UAW: Factor de peso de los actores sin ajustar.

Para calcular UAW, utilizaremos la siguiente tabla y nos apoyaremos en la fórmula que continúa  $UAW = \sum cant \text{ actores} * peso \text{ (1)}$

Tipo de actor	Descripción	Factor de peso	Actores	Total
Simple	Sistema con sistema a través de interfaz de programación.	1	0	0
Medio	Sistema con sistema mediante protocolo de interfaz basada en texto.	2	0	0
Complejo	Persona que interactúa con el sistema mediante interfaz gráfica.	3	2	6

El resultado obtenido de aplicar (1) sería **UAW = 6**.

Para calcular UUCW, nos basamos en la tabla que sigue y utilizamos la fórmula  $UUCW = \sum cant \text{ CU} * Peso \text{ (2)}$

Tipo de CU	Descripción	Peso	Cantidad de CU	Total
Simple	El caso de uso tiene de 1 a 3 transacciones.	5	1	5
Medio	El caso de uso tiene de 4 a 7 transacciones.	10	5	50
Complejo	El caso de uso tiene 8 o más transacciones.	15	7	105

El resultado obtenido de aplicar (2) es **UUCW = 160**

Una vez obtenidos los resultados de (1) y (2) podemos calcular UUCP.

$$UUCP = UUCW + UAW$$

$$UUCP = 160 + 6$$

**UUCP = 166.**

**Paso 2: Cálculo de los puntos de los casos de uso ajustados.**

Este paso propone la fórmula  $UCP = UUCP * TCF * EF$  donde:

UCP: Puntos de casos de uso ajustados.

UUCP: Puntos de casos de uso sin ajustar.

TCF: Factor de complejidad técnica.

EF: Factor de ambiente.

El factor de complejidad técnica (TCF) se calcula mediante la cuantificación de un conjunto de factores que determinan la complejidad técnica del sistema. Cada factor se cuantifica en un valor desde 0 (aporte irrelevante) hasta 5 (aporte muy relevante).

Factor	Descripción	Peso	Valor asignado	Total
T1	Sistema distribuido	2	4	8
T2	Tiempo de respuesta	1	4	4
T3	Eficiencia del usuario final	1	4	4
T4	Funcionamiento Interno complejo	1	4	4
T5	El código debe ser reutilizable	1	3	3
T6	Facilidad de instalación	0,5	4	2
T7	Facilidad de uso	0,5	5	2,5
T8	Portabilidad	2	4	8
T9	Facilidad de cambio	1	4	4
T10	Concurrencia	1	5	5
T11	Incluye objetivos especiales de seguridad	1	5	5
T12	Provee acceso directo a terceras partes	1	5	5
T13	Se requieren facilidades especiales de entrenamiento de usuarios	1	4	4

Una vez completada la tabla podemos calcular el TCF a través de:

$$TCF = 0.6 + 0.01 * \sum (peso * valor asignado)$$

**TCF = 1.185**

El factor de ambiente (EF) está relacionado con las habilidades y entrenamiento del grupo de desarrollo que realiza el sistema. Cada factor se cuantifica con un valor desde 0 (aporte irrelevante) hasta 5 (aporte muy relevante).

Factor	Descripción	Peso	Valor asignado	Total
E1	Familiaridad con el modelo de proyecto utilizado	1,5	4	6
E2	Experiencia en la aplicación	0,5	3	1.5
E3	Experiencia en la orientación a objetivos.	1	4	4
E4	Capacidad del analista líder.	0,5	3	1.5
E5	Motivación.	1	2	2
E6	Estabilidad de requerimientos	-2	3	6
E7	Personal Part-Time	-1	2	-2
E8	Dificultad del lenguaje de programación	-1	4	-4

Para calcular EF necesitamos utilizar:

$$EF = 1.4 - 0.03 * \sum ( peso * valor asignado )$$

Entonces el resultado es **EF = 0.95**

Hemos obtenido hasta el momento UUCP, TFC, EF dejándonos en condiciones de calcular **UCP = 190.252**

### Paso 3: Estimación de esfuerzo a través de los puntos de casos de uso.

El cálculo del esfuerzo está determinado por la fórmula  $E = UCP * CF$

Donde:

E: Esfuerzo estimado en horas hombres.

UCP: Punto de casos de usos ajustados.

CF: Factor de conversión.

Para obtener el factor de conversión (CF) se cuentan cuantos valores de los que afectan el factor ambiente (E1...E6) están por debajo de la media (3), y los que están por arriba de la media para los restantes (E7, E8). Si el total es 2 o menos se utiliza el factor de conversión 20 Horas-Hombre / Punto de Casos de uso. Si el total es 3 o 4 se utiliza el factor de conversión 28 Horas-Hombre / Punto de Casos de uso. Si el total es mayor o igual que 5 se recomienda efectuar cambios en el proyecto ya que se considera que el riesgo de fracaso del mismo es demasiado alto.

En este caso se puede decir que  $E = 3805.035$

**Paso 4: Calcular esfuerzo de todo el proyecto**

Actividad	% esfuerzo	Horas-Hombres
Análisis	10	951.258
Diseño	20	1902.517
Implementación	40	3805.035
Pruebas	15	1426.888
Sobrecarga	15	1426.888
<b>Total</b>	<b>100</b>	<b>9512.587</b>

Si  $E_T = 9512.587$  horas-hombre y cada mes tiene como promedio 240 horas, eso daría un  $E_T = 39.636$  mes-hombre.

Esto quiere decir que 1 persona puede realizar el problema analizado en aproximadamente 40 meses.

Podríamos decir que si una persona puede realizar el trabajo en 40 meses aproximadamente y el equipo está compuesto por 7 personas se concluiría diciendo que el trabajo se puede realizar en aproximadamente 5 meses y medio aunque esta deducción matemática no es recomendable.

### **Costo del Proyecto.**

Se asume como salario promedio mensual \$50, debido a que los integrantes del equipo son estudiantes.

$$\text{CHM} = 1 * \text{Salario Promedio}$$

$$\text{CHM} = 350 \text{ \$/mes}$$

$$\text{Costo} = \text{CHM} * E_T$$

$$\text{Costo} = \$ 1750.52$$

### **4.4 Beneficios tangibles e intangibles.**

El Sistema de Contabilidad Financiera para la Actividad Presupuestada en las FAR no es un producto con fines comerciales, su principal objetivo y beneficio es la existencia de una aplicación Web agradable, dinámica y flexible que permite registrar y controlar los hechos económicos que ocurren en un órgano financiero.

Que no tenga beneficios tangibles desde el punto de vista monetario, no significa que no los tenga y podemos encontrar:

- ✓ Disminución del tiempo y esfuerzo que se invierte en la realización esta tarea, hasta ahora, de forma manual.
- ✓ Disminución de los gastos, pues resulta menos costoso crear y procesar información digital que copias duras.
- ✓ Fácil procesamiento de la información y obtención, dinámica, de reportes de la situación de la misión en cualquier momento.
- ✓ Disminución de la acumulación de materiales impresos relacionados con los procesos de registro y control de los movimientos económicos.
- ✓ Fácil detección de problemas.

Los beneficios intangibles son:

- ✓ Mejoramiento en la toma de decisiones.
- ✓ Fácil y rápido acceso a la información.
- ✓ Procesos de aprendizaje más rápido.
- ✓ Mejoramiento de la productividad.

### 4.5 Análisis de costo y beneficios.

Al inicio de este capítulo mencionábamos las distintas variables que se involucran en la construcción de un software, hablábamos de tiempo y recursos, mas ahora podemos introducir el costo, variable calculada en secciones anteriores.

El realizar un análisis de costo y beneficios de un producto informático nos resulta útil para determinar si construir el sistema es factible o no; los beneficios pueden ser económicos o de orden social.

Como mencionamos anteriormente el sistema que se propone no es un producto con fines comerciales está dirigido fundamentalmente a la contabilidad financiera en las FAR; y esta entidad la encarga de asumir el costo tanto directo como indirecto en los que se incurra, por tanto es factible el desarrollo de la aplicación.

### 4.6 Análisis de Sostenibilidad

Todo producto tiene repercusión social, económica, política, ambiental, etc., el nuestro no está exento de este tipo de análisis y podemos decir que:

**Social:** La aplicación no produce desempleos, problema común que suele ocurrir con otros softwares, más bien proporciona al personal de una herramienta más eficiente para realizar con calidad su trabajo por lo que resulta sostenible.

Económico: La aplicación como se mencionaba en los beneficios tangibles e intangibles proporciona una serie de bienes que de una forma u otra resultan un ahorro económico al país, el hecho de procesar la información de manera digital y no en copia dura representa una disminución monetaria en cuanto a la compra de insumos de oficina; disminuir el tiempo y esfuerzo requerido para realizar la actividad por la que fue concebido el sistema también representa una mejora para la economía ya que se produciría más en una misma jornada laboral, estos elementos lo hacen sostenible económicamente.

Político: El sistema basa su funcionalidad en normas y principios estipuladas en órdenes firmadas por el Ministerio, es por ello que los resultados que este proporciona son totalmente legales y no constituyen fraudes, manteniendo transparente el sistema económico del país y haciendo del software un sistema sostenible.

Ambiental: El sistema no daña el medio ambiente, no provoca ruidos, sus colores fueron concebidos para que no causaran daños visuales a los clientes por lo que podemos decir que es sostenible.

### 4.7 Conclusiones

Durante este capítulo hemos detallado de una forma clara los costos a incurrir, los recursos materiales necesarios, la importancia de la variable tiempo para el desarrollo de una aplicación, la planificación llevada a cabo para poner fin a las metas propuestas y los beneficios del producto en cuestión.

También analizamos cuán factible y sostenible nos resultaría desarrollar la aplicación y arribamos a dos conclusiones fundamentales:

1. Desarrollar el sistema es factible porque los desarrolladores no incurrir en costo alguno.
2. La aplicación es sostenible.



## CONCLUSIONES

Al finalizar las fases de elaboración y construcción seguidas en la producción de SCF; podemos arribar a las siguientes conclusiones:

- ✓ El estudio de las necesidades del usuario y los procesos que se realizan en la entidad, son imprescindibles para el desarrollo de un sistema informático.
- ✓ El uso de una metodología para el desarrollo de una aplicación informática, es vital para lograr el objetivo final, el producto, y en él rasgos como la calidad.
- ✓ La metodología utilizada para el análisis, diseño y desarrollo de la aplicación resultó eficiente y queda disponible para su utilización en sistemas similares.
- ✓ Valorados los impactos social, económico, político y ambiental del proceso de desarrollo e implantación del SCF, se puede afirmar que el producto es sostenible.
- ✓ Se logró informatizar los procesos necesarios para el registro y control de los hechos económicos de un órgano financiero.
- ✓ Se comenzó a desarrollar el Sistema de Contabilidad Financiera (SCF) para el MINFAR, el cual mejorará el registro y control de los hechos económicos de un órgano financiero y el tiempo de procesamiento, por lo tanto, favorecerá la toma de decisiones de los directivos.

Se puede plantear en este momento, que concluido este trabajo de diploma y desarrollado los temas que en él se exponen y que describen el proceso de análisis, modelación, elaboración y producción de SCF; se ha cumplido la hipótesis planteada en el mismo.



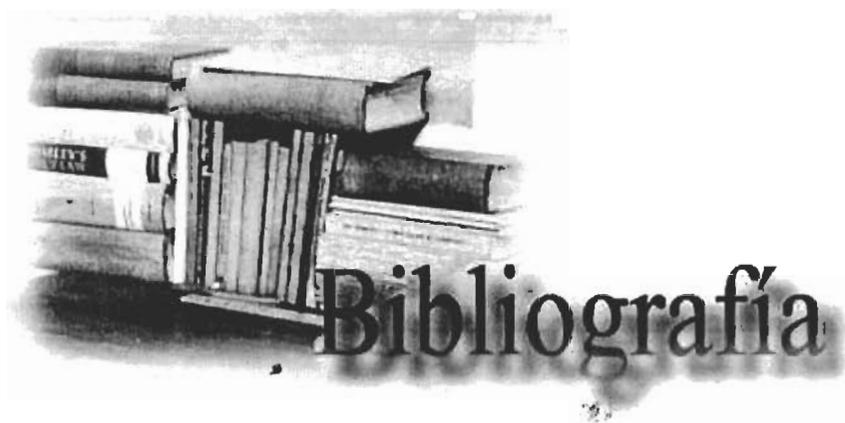
# Recomendaciones

---

## RECOMENDACIONES

Como resumen de todo lo discutido hasta este momento, y luego de la presentación de todo lo trabajado para concretar el Sistema de Contabilidad Financiera (SCF), se listan a continuación unas series de recomendaciones tanto teóricas como prácticas para la ampliación, modificación, mejora y construcción de nuevas iteraciones del sistema.

- ✓ Analizar el tema de los estándares de diseño para próximas iteraciones en aras de obtener un producto más amigable.
- ✓ Continuar la implementación de todas las funcionalidades diseñadas para SCF, hasta completar el funcionamiento de la aplicación completa para el usuario final.
- ✓ Valorar por parte del usuario final, la utilización de la modelación e implementación del SCF para el montaje de nuevos sistemas asociados al objeto de estudio.
- ✓ Perfeccionar en el tema del tratamiento de errores.
- ✓ Profundizar en la utilización de las herramientas de análisis y diseño que incorpora la Visual Paradigm, específicamente las relacionadas con la arquitectura y la generación de código.



## BIBLIOGRAFÍA

- Pressman, R. *Software Engineering. A Practitioner's Approach*. Fourth Edition. McGraw – Hill. USA, 1999.
- Booch, G., Rumbaugh, J., Jacobson, I. *El Lenguaje Unificado de Modelado*. Addison-Wesley. 1999.
- Larman, C. *UML Y PATRONES, Introducción al análisis y diseño orientado a objetos*. La Habana. Cuba 2004.
- Booch, G., Rumbaugh, J., Jacobson, I. *El Proceso Unificado de Desarrollo de Software*. La Habana. Cuba 2004.
- Franco, J A. *UML en acción. Modelando Aplicaciones Web*. La Habana. Cuba 2006.
- Muñoz, J. *Un Framework basado en OSGi para el Desarrollo de Sistemas Pervasivos*. mayo 2006.
- Méndez G. *Construcción de Aplicaciones Web con UML. Diseño*. Mayo 2006.
- Méndez G. *Construcción de Aplicaciones Web con UML. Conceptos Generales*. Mayo 2006.
- Mexica J. *Un patrón arquitectónico para la creación de cursos WBT*. mayo 2006.
- Herrera R A., Caldera R J., Martínez M. tema: *Análisis y Diseño de Sistemas con el Lenguaje de Modelaje Unificado (UML)*. Universidad Católica Redemptoris Mater". Proyecto Monográfico. Managua, abril de 1999.
- Soto Lopez N M., Saborit Ramírez Y. *Sistema de Catalogación y Recuperación de Recursos de Información, HUBBLE*, trabajo de diploma para optar por el título de Ingeniería en Informática. Instituto Superior

## Bibliografía

---

- Politécnico “José Antonio Echeverría”. Ciudad de la Habana, junio del 2004.
- Espinosa Hidalgo A. *Sistema para la Administración Unificada de usuarios*. Trabajo de diploma para optar por el título de Ingeniería en Informática. Instituto Superior Politécnico “José Antonio Echeverría”. Ciudad de la Habana, junio del 2004.
  - Franco Navarro, J. *Acceso a Datos Objeto – Relacional; Patrones y Mecanismos. EJB - CMP, una solución “Alguien lo hace por mí” sobre J2EE*. Consultado en mayo del 2006.
  - Grupo de Ingeniería del software, Universidad de Sevilla, *Introducción al Análisis de Requisitos*. Publicado en Octubre del 2005. Consultado en mayo del 2006.
  - Grupo de Ingeniería del software, Universidad de Sevilla, *Elicitación de Requisitos: modelado del Negocio. (Diagramas de Actividades)*. Publicado en Octubre del 2005. Consultado en mayo del 2006.
  - Grupo de Ingeniería del software, Universidad de Sevilla, *Documentación de Casos de Uso*. Publicado en Octubre del 2005. Consultado en mayo del 2006.
  - Grupo de investigación Ingeniería de Software. Universidad EAFIT. *Tertulia de Ingeniería de Software. La Importancia de la Arquitectura en el desarrollo de software de calidad*. Publicado febrero 17 de 2005. Consultado mayo 26, 2006.
  - Isla Monte, j. **Modelado Estructural de Patrones de Diseño**. Cádiz, España, 2003.
  - Peralta M. *Estimación del esfuerzo basada en casos de usos.pdf*. <http://www.itba.edu.ar/capis/webcapis/planma.html> , mayo del 2006.
  - <http://msdn.microsoft.com/architecture/> (mayo 10, 2006)
  - <http://www.dm.univaq.it/~muccini> (mayo 10, 2006)

## Bibliografía

---

- <http://www.sparxsystems.cl/uml-tutorial.html> (mayo 28, 2006)
- [http://www.sparxsystems.cl/resources/uml\\_datamodel.html](http://www.sparxsystems.cl/resources/uml_datamodel.html) (mayo 28, 2006)
- [http://www.sparxsystems.cl/resources/developers/uml\\_patterns.html](http://www.sparxsystems.cl/resources/developers/uml_patterns.html) (mayo 28 2006)
- <http://www.mozilla-europe.org/es/about/> (mayo 22, 2006)
- <http://kursor.net/article/584/control-de-versiones-con-subversion> (mayo 22, 2006)
- <http://www.capisol.net/formacion/mod/book/view.php?id=52&chapterid=25> (mayo 22,2006)
- <http://www.lug.fi.uba.ar/sistemas-de-control-versiones.mht> (mayo 22, 2006)
- <http://www.monografias.com/trabajos15/estimacion-hipermedia/estimacion-hipermedia.shtml#INTRO> (mayo 23, 2006)
- [http://es.wikipedia.org/wiki/Patr%C3%B3n\\_de\\_dise%C3%B1o](http://es.wikipedia.org/wiki/Patr%C3%B3n_de_dise%C3%B1o) (mayo 29. 2006)
- <http://es.tldp.org/Tutoriales/doc-modelado-sistemas-UML/multiple-html/x219.html> (mayo 29, 2006)
- <http://www.fi.uba.ar/~dmontal/> (mayo 29, 2006)
- <http://www3.uji.es/~mmarques/f47/apun/node79.html> (junio 4, 2006)
- <http://www.monografias.com/trabajos10/recped/recped.shtml#intro> (marzo 14, 2006)
- [http://www\\_php\\_net-pgsql.htm/](http://www_php_net-pgsql.htm/) (marzo15, 2006)

## GLOSARIO DE TÉRMINOS

**Artefacto**: Pieza de información utilizada o producida por un proceso de desarrollo de software como un documento externo o el producto de un trabajo. Un artefacto puede ser un modelo, una descripción o el software.

**Balance de Comprobación de Saldos**: Listado de saldos de todas las cuentas del balance, para comprobar si los saldos deudores son iguales a los saldos acreedores.

**Ciclo contable**: Serie de actividades contables que tienen lugar desde el comienzo hasta el final de un período contable. En cada período se repite la misma secuencia de procedimientos contables que comienza con el registro de los hechos económicos que tienen lugar durante el período y concluye con la elaboración de los estados financieros correspondientes.

**Cierre de operación**: Acciones y anotaciones contables realizadas al final del ejercicio económico, destinadas a cerrar los libros de contabilidad al efectuar los últimos asientos del año.

**Conciliación bancaria**: Proceso sistemático de cuadro entre la información que brinda el Banco sobre las operaciones y el saldo de una cuenta bancaria y las cuentas contables de banco correspondientes a una unidad, con una explicación de las diferencias encontradas.

**Cuenta**: Registro que contiene las operaciones de igual naturaleza y la fecha de cada una de ellas en orden cronológico. Las anotaciones en una cuenta se expresan en débitos y créditos, evaluados en términos monetarios, y muestran el saldo actual en caso de existir.

**Cuentas nominales**: Reflejan las operaciones que generan recursos financieros (ingresos) o gastos de la unidad. Comprenden el importe de los gastos corrientes ejecutados durante el ejercicio económico por las unidades,

con cargo al presupuesto anual aprobado, analizadas por conceptos, así como el importe de los recursos financieros (ingresos) cobrados en el caso de las unidades que se autoricen a realizar esta función.

Estas cuentas tienen como características que se dividen en Deudoras y Acreedoras. Las Deudoras se identifican con los gastos y las Acreedoras, con los recursos financieros (ingresos).

**Cuentas de orden:** Permiten controlar en las unidades el presupuesto anual aprobado, el financiamiento situado o recibido para Gastos Corrientes y de Capital, así como el comportamiento de otros hechos económicos específicos.

**CU:** Caso de Uso.

**ERP:** Enterprise Resource Planning (Sistema de Planificación de Recursos) son sistemas de gestión de información que integran y automatizan muchas de las prácticas del negocio asociadas con los aspectos operativos o productivos de una empresa.

**Estados financieros:** Resumen de la información al cierre del período o ejercicio económico que refleja los elementos primordiales que sintetizan la situación económica de una unidad. Estos constituyen fuentes de información para la propia unidad y para el nivel superior que la atiende financieramente.

**Financiamiento:** Acción de situar los fondos en efectivo a las unidades para garantizar su funcionamiento, sobre la base de la proyección de los pagos que se realiza de acuerdo con las actividades previstas en el período, hasta el límite del presupuesto aprobado.

**GoF:** Gans of Four, grupo de los cuatros, son patrones de asignación de responsabilidades, más conocidos como GRASP.

**HTTP:** Es el conjunto de reglas para intercambiar archivos (texto, gráfica, imágenes, sonido, video y otros archivos multimedia) en la World Wide Web.

**Mayor:** Registro de los asientos contables originales mediante el cual se agrupan las operaciones contables utilizando cuentas al efecto.

Adicionalmente al registro mayor, se habilitan los submayores, libros de cuentas que proporcionan detalles del respaldo de los saldos individuales, cuyo total aparece en el Mayor.

**Paquete**: Término que denota un mecanismo de propósito general para organizar en grupos los elementos. Se pueden anidar dentro de otros paquetes, y en el pueden aparecer tanto elementos del modelo como diagramas.

**Presupuesto**: Disponibilidad de créditos o recursos presupuestarios que se aprueban anualmente por diferentes conceptos y especialidades, para garantizar el funcionamiento de las unidades presupuestadas de las FAR. También incluye los recursos financieros (ingresos) que se prevén captar durante este período.

**Recurso financiero (ingreso)**: denominación que se le da a los ingresos en efectivo o equivalente monetario que se reciben como contrapartida por la venta de bienes y servicios.

**Requisito o Requerimiento**: Una característica, propiedad o comportamiento que se desea para el sistema.

**Saldo contable**: Saldo neto deudor o acreedor de una cuenta. Un exceso de los débitos totales sobre los créditos totales da un saldo deudor, mientras que los excesos de los créditos totales sobre los débitos totales dan lugar a un saldo acreedor.

**SCF**: Sistema de Contabilidad Financiera.

## ANEXOS

### Anexo I

Los tipos de Operación son los siguientes:

- |  |                              |
|--|------------------------------|
| 01: Asiento Apertura                   | 09: Liquidación de Anticipos |
| 02: Asignación/Cancelación Presupuesto | 10: Nóminas Pagos Especiales |
| 03: Acumulación de Nóminas             | 11: Cheques.                 |
| 04: Aviso Crédito                      | 12: Situación de Fondos      |
| 05: Aviso Débito                       | 13: Recursos Financieros     |
| 06: Recibo Efectivo                    | 14: Cuentas por Cobrar       |
| 07: Vales Pagos menores                | 15: Cuentas por Pagar        |
| 08: Entrega Anticipos                  | 16: Ajustes.                 |
|  | 17: Centros de Costos        |

Los nomencladores:

- |                                       |                          |
|---------------------------------------|--------------------------|
| 01: Usuarios.                         | 09: Mandos.              |
| 02: Cuentas Contables.                | 10: Centro de costos.    |
| 03: Órganos Financieros.              | 11: Análisis del gasto.  |
| 04: Objeto de gasto.                  | 12: Grupos de cuentas.   |
| 05: Entidad de gasto.                 | 13: Tipo de operaciones. |
| 06: Grupo presupuestario.             | 14: Rol                  |
| 07: Recursos Financieros.             | 15: Retenciones.         |
| 08: Clientes (Acreedores/Proveedores) | 16: Personas.            |



Anexos

MINFAR	MAYOR SCF-50				No (2)
	NOMBRE DE LA CUENTA				CODIGO
UM: (1)					
AÑO	CO	DETALLE	DEBE	HABER	SALDO
2006	(6)	(7)	(8)	(9)	(10)
		<b>Total</b>			

Reporte generado por el CU: Confeccionar Mayor por cuentas desglosadas

MINFAR	BALANCE DE COMPROBACIÓN DE SALDOS				
	SICF - ESTADO A				
Um (1)	PERIODO TERMINADO EN: (2) D / M / A		UNIDAD DE MEDIDA (PESOS Y CENTAVOS)		
CODIGO (3)	NOMBRE DE LA CUENTA (4)	PARCIAL (5)	DEBE (6)	HABER (7)	
	<b>CUENTAS PATRIMONIALES</b>				
	SUBTOTAL	////////////////////	(8)	(9)	
	<b>CUENTAS DE ORDEN</b>				
	SUBTOTAL	////////////////////	(10)	(11)	
	TOTAL GENERAL	////////////////////	(12)	(13)	
CERTIFICAMOS QUE LOS DATOS CONTENIDOS EN ESTE ESTADO SE CORRESPONDEN CON LAS ANOTACIONES CONTABLES			APROBADO JEFE UNIDAD MILITAR (16)	FECHA (17)	
CONFECCIONADO POR (14)	JEFE ORGANO ECONOMIA/FINANZAS (15)			D	M

Reporte generado por el CU: Confeccionar Balance

**Anexos**

Rg	Fecha	Presupuesto			UM	UM .....
		Aprobado	Distribuido	No Distrib		
Suma Acumulada						
Total Gral						

**Reporte generado por el CU: Confeccionar Submayores**

MINFAR		SUBMAYOR DE GASTOS DEL PRESUPUESTO					No (2)		
UM: (1)		SCF-52							
GRUPO PRESUPUESTARIO (3)				ESPECIALIDAD (3)		CLASIFICACION DEL GASTO (4)			
						Salarios _____ O.G. _____			
						Inversiones _____			
AÑO (5)		CO (6)	DETALLE (7)	PRESUP. APROBADO (8)	GASTOS (REINTEGRO) (9)	SALDO (10)	PARTIDAS (11)		
D	M						H		H
Suma Acumulada mes Ant.									
Total General									

**Reporte generado por el CU: Confeccionar Submayores**

MINFAR		SUBMAYOR DE RECURSOS FINANCIEROS COBRADOS Y TRANSFERIDOS						No. (2)		
UM: (1)		SCF-53								
CODIGO DE INGRESO (3)				DENOMINACIÓN (3)			PRESUPUESTO APROBADO (4)			
							UM (12)			
AÑO (5)		CO (6)	DETALLE (7)	CO (8)	DEBE (9)	HABER (10)	SALDO (11)			
D	M									
Suma Acumulada Mes anterior										
03	01	001				5.00	5.00			
04	01			002	5.00		0.00			
Subtotal										

**Reporte generado por el CU: Confeccionar Submayores**

# Anexos

## Anexo III

Id		Nombre de tarea	Duración	Comienzo	Fin	Predecesoras	Nombres de los recursos
1		<b>Modelamiento del negocio</b>	<b>16 días?</b>	<b>mié 2/1/06</b>	<b>mié 2/22/06</b>		<b>Insumos de oficina, Computad</b>
2	■	Identificación de Procesos de Negocio	6 días	mié 2/1/06	mié 2/8/06		
3	■	Identificación de Roles del Entorno del Negocio	3 días	jue 2/9/06	lun 2/13/06	2	
4		Especificación de Reglas del Negocio	1 día?	mar 2/14/06	mar 2/14/06	3	
5		Obtención de los modelos principales del negocio	3 días	mié 2/15/06	vie 2/17/06	4	
6		Descripción de los Casos de Uso del Negocio	3 días	lun 2/20/06	mié 2/22/06	5	
7		<b>Captura de requisitos</b>	<b>30 días</b>	<b>jue 2/16/06</b>	<b>mar 3/28/06</b>		
8	■	Obtención requerimientos funcionales del sistema	5 días	jue 2/16/06	mié 2/22/06		
9	■	Obtención requerimientos no funcionales del sistema	5 días	mié 2/22/06	mar 2/28/06		
10	■	Encontrar los actores y casos de uso	2 días	mar 2/28/06	mié 3/1/06		
11		Obtención del modelo de caso de uso del sistema	5 días	jue 3/2/06	mié 3/8/06	10	
12	■	Descripción de los casos de usos	15 días	jue 3/9/06	mar 3/28/06		
13		<b>Análisis y Diseño</b>	<b>48 días</b>	<b>sáb 3/18/06</b>	<b>mar 5/23/06</b>		
14	■	Identificación de clases del análisis por CU	6 días	jue 3/30/06	jue 4/6/06		
15		Obtención del diagrama de clases del análisis por CU	4 días	vie 4/7/06	mié 4/12/06	14	
16		Obtención del diagrama de colaboración por CU	4 días	jue 4/13/06	mar 4/18/06	15	
17	■	Obtención de los mecanismos de diseño	30 días	mié 4/5/06	mar 5/16/06		
18	■	Identificación de las clases del diseño por CU	10 días	mié 4/19/06	mar 5/2/06	16	
19		Obtención del diagrama de clase del diseño por CU	5 días	mié 5/3/06	mar 5/9/06	18	
20		Obtención del diagrama de secuencia por realizaciones de CU	10 días	mié 5/10/06	mar 5/23/06	19	
21	■	Obtención del diagrama de clases persistentes	4 días	sáb 3/18/06	mié 3/22/06		
22		<b>Implementación</b>	<b>115 días</b>	<b>sáb 3/18/06</b>	<b>jue 8/24/06</b>		
23	■	Definir arquitectura	5 días	sáb 3/18/06	jue 3/23/06		
24		Diseñar componentes	5 días	mié 4/19/06	mar 4/25/06	16	
25		Diseñar Base de Datos	5 días	jue 3/23/06	mié 3/29/06	21	
26		Obtener diagrama de componentes	1 día	mié 4/26/06	mié 4/26/06	24	
27		Obtener diagrama de despliegue	1 día	vie 3/24/06	vie 3/24/06	23	
28		Generación de códigos a partir de diagrama de clases	1 día	mié 5/3/06	mié 5/3/06	18	
29		Implementar los elementos de diseño	60 días	mié 5/24/06	mar 8/15/06	20	
30		Integrar los resultados en un sistema ejecutable	7 días	mié 8/16/06	jue 8/24/06	29	
31		<b>Prueba</b>	<b>10 días</b>	<b>vie 8/25/06</b>	<b>jue 9/7/06</b>		
32	■	Encontrar y documentar los defectos del software	10 días	vie 8/25/06	jue 9/7/06		
33	■	Probar que el software trabaje como fue diseñado	10 días	vie 8/25/06	jue 9/7/06		
34	■	Probar los requisitos que debe cumplir el software	10 días	vie 8/25/06	jue 9/7/06		
35	■	Probar que los requisitos fueron implementados correctamente	10 días	vie 8/25/06	jue 9/7/06		

Figura 4.2 Tabla de asignación de tareas y recursos elaborada con el Microsoft Project.

Anexo IV

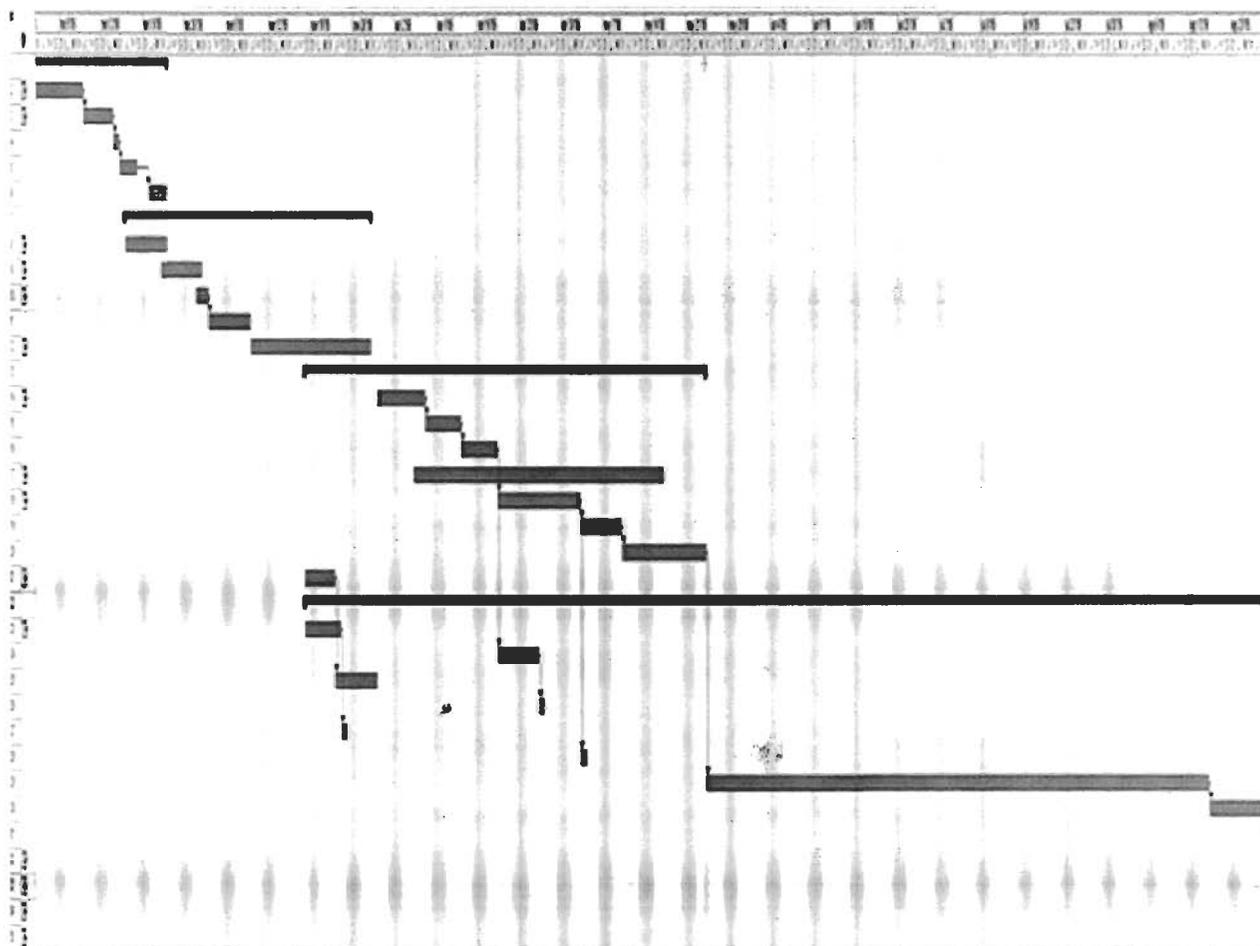


Figura 4.3 Diagrama de Gant correspondiente a la tabla de asignación de tareas y recursos

Anexo V

A continuación se muestran las descripciones de alto nivel de los demás CU que componen al sistema.

<b>Caso de Uso:</b>	<b>Confeccionar Mayor</b>
Actor(es):	Contador (inicia).
Propósito:	Visualizar y/o imprimir el resumen del mayor (scf - 50)
Resumen:	El CU se inicia cuando el contador principal solicita recuperar el mayor de una o varias cuentas entre una fecha determinada
Referencias:	R4
Precondiciones:	Debe existir un comprobante de operaciones y el usuario debe haberse autenticado como Contador.
Poscondiciones:	Se recupera el Mayor (SCF-50)

El CU: Confeccionar Mayor genera dos reportes SCF-50, que pueden verse en el anexo II.

<b>Caso de Uso:</b>	<b>Confeccionar Submayores</b>
Actor(es):	Contador (inicia).
Propósito:	Visualizar y/o imprimir los submayores
Resumen:	El caso de uso inicia cuando el contador solicita recuperar uno de los submayores de una o varias cuentas que pueden ser: Submayor de Control de presupuesto. Submayor de Gasto de presupuesto. Submayor de Ingresos. Submayor de Pendiente por situar. y termina cuando obtiene uno de los submayores
Referencias:	R5
Precondiciones:	Debe haberse actualizado al menos un comprobante de presupuesto, uno de gasto, de ingreso o pendiente y el usuario debe autenticarse como contador.
Poscondiciones:	Se recupera el o los submayores.

Este CU genera varios reportes (SCF-52, SCF-53) que pueden verse en el anexo II.

## Anexos

<b>Caso de Uso:</b>	<b>Confeccionar Balance</b>
Actor(es):	Contador (inicia).
Propósito:	Editar o imprimir el Balance de Operación de Saldos (Estado A)
Resumen:	El caso de uso inicia cuando el contador desea obtener el balance de un comprobante y termina cuando lo recupera.
Referencias:	R3
Precondiciones:	Debe haberse gestionado al menos un comprobante.
Poscondiciones:	El comprobante queda desconformado.

Este CU genera un reporte SICF (estado-A) que puede verse en el anexo II

<b>Caso de Uso:</b>	<b>Analizar Centro de Costo</b>
Actor(es):	Contador (inicia).
Propósito:	Realizar un análisis de los centros de costo.
Resumen:	El caso de uso inicia cuando el usuario solicita hacer un análisis de los centros de costo y termina cuando hace el análisis.
Referencias:	R7.
Precondiciones:	El usuario debe autenticarse como contador.
Poscondiciones:	Se realiza el análisis.

<b>Caso de Uso:</b>	<b>Iniciar/Cerrar operación</b>
Actor(es):	Contador (inicia).
Propósito:	Desconfirmar un comprobante de operaciones para realizar una operación sobre el mismo.
Resumen:	El caso de uso se inicia cuando el usuario escoge cerrar una operación de año y pasa la fecha y el registro y termina cuando se borra las tablas en curso y se pasará de la auxiliar a estas con el registro que dio el contador y la fecha hacia el nuevo año.
Referencias:	R8.
Precondiciones:	El usuario debe autenticarse como contador y deben existir cuentas que pertenezcan al año que se acaba, en caso de que se vaya a cerrar.
Poscondiciones:	Se inicia o cierra una operación de año

<b>Caso de Uso:</b>	<b>Traspasar a Estados Financieros.</b>
Actor(es):	Contador (inicia).
Propósito:	Traspasar los saldos de las cuentas contables.
Resumen:	El caso de uso inicia cuando el contador decide traspasar los saldos de las cuentas contables, estos pueden ser: 1- Traspaso de la contabilidad al estado A 2- Traspaso de la contabilidad al estado C. 3- Traspaso de la contabilidad al estado B. 4- traspaso de la contabilidad al anexo 1. Y termina cuando se traspasan los saldos a los EF.
Referencias:	R9.
Precondiciones:	El usuario ha de autenticarse como contador
Poscondiciones:	Los saldos son transferidos a los EF.

## Anexos

<b>Caso de Uso:</b>	<b>Preparar Presupuesto/Financiamiento</b>
Actor(es):	Contador (inicia).
Propósito:	Asignar presupuestos o financiamientos.
Resumen:	El caso de uso inicia cuando el contador desea preparar un presupuesto o un financiamiento y termina cuando realiza una operación con estos.
Referencias:	R6.
Precondiciones:	Debe existir presupuesto o financiamiento y el usuario debe autenticarse como contador.
Poscondiciones:	Quedan preparados los presupuestos o financiamiento.

<b>Caso de Uso:</b>	<b>Actualizar Nomencladores.</b>
Actor(es):	Jefe Finanzas (inicia).
Propósito:	Actualizar los valores de los nomencladores.
Resumen:	El caso de uso inicia cuando el jefe de finanzas desea actualizar los datos de uno de los nomencladores y termina cuando el nomenclador o los nomencladores actualizados guardan su nuevo valor.
Referencias:	R10.
Precondiciones:	Deben existir previamente los nomencladores y el usuario tiene que autenticarse como jefe de finanzas.
Poscondiciones:	Los nomencladores quedan actualizados.