

Universidad de las Ciencias Informáticas.

Facultad 9



Modelado de una Aplicación WEB para el módulo de Activos Fijos del Sistema Económico-Contable ConDor



TRABAJO DE DIPLOMA PARA OPTAR POR EL TÍTULO DE INGENIERO EN CIENCIAS INFORMÁTICAS

Autores: Ricardo Medina Díaz.
Roberto Fernández Sánchez.

Tutora: MSc. Isabel María Martínez García.

La Habana, 28 de junio del 2007

“Año 49 de la Revolución”

“Nunca consideres al estudio como una obligación, sino como una oportunidad para penetrar en el bello y maravilloso mundo del saber”

Albert Einstein

DECLARACIÓN DE AUTORÍA

Declaramos que somos los únicos autores de este trabajo y autorizamos al SICS” Servicios Informáticos de Consultaría y Sistema”, y a la Universidad de las Ciencias Informáticas para que hagan el uso del mismo en su beneficio.

Para que así conste firmamos la presente a los _____ días del mes de _____ del 2007.

Ricardo Medina Díaz

Roberto Fernández Sánchez.

MSc. Isabel María Martínez García.

AGRADECIMIENTOS

Antes que todo quiero agradecer a mis padres por estar presentes en todo momento, y principalmente cuando más los he necesitado y confiar en mí todo el tiempo. A mi familia por ser la base del interés y esfuerzo que he dedicado al desarrollo de todo lo que he podido construir. Algo que siempre voy a agradecer en esta vida es el haber tenido la suerte de tener conmigo a una tropa de amigos y hermanos durante 5 años, que me han apoyado y ayudado mucho todo este tiempo, en especial a Leynier Díaz Navarro, Roberto Fernández Sánchez, Amado Zamora González y Leonardo Pérez Laurence, más que amigos son hermanos, todo mi agradecimiento para ellos. Un agradecimiento especial para mi novia que siempre estuvo allí donde más la necesitaba dándome apoyo en todo momento. Para todos los profesores que durante todo el transcurso como estudiante me han dado la formación e instrucción necesaria y hoy pueden ver el fruto de su trabajo. A una persona que no se puede quedar sin agradecer porque es el pilar principal de este resultado, es a mi tutora Isabel María Martínez García que con su dedicación y sus críticas ha ayudado a hacer las cosas de la mejor manera, gracias de todo corazón. En sentido general a todos aquellos que me quieren y me estiman, aquellos que me han enseñado que en la vida todo se puede con esfuerzo y sacrificio, a todos los que cada día desean que yo crezca como profesional y como amigo. ***Muchas Gracias a todos, los quiero mucho.***

Ricardo Medina Díaz

A mi familia en especial a mis padres que han recorrido junto a mí el camino hasta hoy con plena confianza en mí, a mi abuela María que aunque no llego a ver este día tuvo mucho que ver con el éxito del mismo, a mi tío Lazaro Fernández por estar siempre presente, a mis amigos que han estado siempre brindando su mano en los momento difíciles, a aquellos que confiaron en mí para salir adelante pese a las situaciones difíciles por las que atravesé en estos años de estudio alejado de mi familia pero cercado de amigos inmejorables, mi otra familia que estará por siempre en mi memoria, a mis compañeros de aula por soportarme este tiempo y en especial mi compañero de tesis Ricardo, que el primer día que se habló de tesis dijo que la íbamos a hacer juntos y aquí está el resultado de ese trabajo pues para él la mejor de las suertes, y por ultimo no menos importante a mis profesores que los considero mis amigos en especial a la tutora de este trabajo “*la profe de tele*” gracias a todos.

Roberto Fernández Sánchez

DEDICATORIA

A mis padres...

OPINIÓN DE LA TUTORA

Título: Modelado de una Aplicación WEB para el módulo de Activos Fijos del Sistema Económico-Contable ConDor.

Autores: Ricardo Medina Díaz

Roberto Fernández Sánchez

La tutora del presente Trabajo de Diploma considera que durante su ejecución los estudiantes mostraron, entre otras, las cualidades siguientes:

- Independencia y laboriosidad en la etapa de investigación y desarrollo de la tesis.
- Originalidad y creatividad en la búsqueda de soluciones, con un trabajo sostenido, serio y profundo.
- Alta responsabilidad en la tarea así como gran receptividad ante los señalamientos tanto de su tutora como de los profesores que de una u otra forma estuvieron vinculados en la etapa de desarrollo del trabajo.
- Facilidad de intercambio con los trabajadores que se convierten en usuarios de la aplicación.

El trabajo posee una alta calidad científico-técnica permitiendo, y está referido a la mejora de uno de los objetivos fundamentales de nuestro país en la actualidad el Desarrollo de Software cubano para la Gestión empresarial, entre ellos los Sistemas Económicos- Contables.

Por todo lo anteriormente expresado considero que los estudiantes están aptos para ejercer como Ingenieros Informáticos; y propongo que se le otorgue al Trabajo de Diploma la calificación de 5 puntos.

MSc. Isabel M. Martínez García
Profesor Asistente

Fecha: _____

RESUMEN

Con la evolución de la Informática en Cuba, nuestro país se ha visto inmerso en la preparación de estudiantes y condiciones para lograr un desarrollo sostenido de Software Cubanos que ayuden a una mejor Gestión Empresarial en todos y cada uno de los sectores económicos y sociales. La investigación desarrollada en este Trabajo de Diploma está vinculada con la empresa de Servicios Informáticos de Consultoría y Sistema (SICS), perteneciente al Ministerio de Transporte y dedicada, entre otras cosas, al desarrollo de software para el sector empresarial. Dicha empresa tiene como producto principal el Sistema Económico – Contable **ConDor**, instalado y en funcionamiento en más de 70 empresas, el sistema está desarrollado en MS-DOS y recientemente en una versión de escritorio que tiene como características fundamentales el trabajo en modo monousuario, es aquí donde surge la necesidad de modelar y diseñar una aplicación Web que permita el trabajo multiusuario y que además posea una base de datos única para todos los módulos que contempla, desarrollando en el Trabajo de Diploma específicamente el Módulo de Activos Fijos.

Al modelado de esta aplicación Web se le hicieron mejoras en sus funcionalidades, proponiendo un ambiente sencillo y amigable, que facilite la instalación y utilización del sistema, se sugiere un producto que contenga una alta seguridad y agilidad en el procesamiento de sus datos para la gestión de activos fijos. Para lograr de los objetivos se hace un estudio del sistema existente, de las tecnologías y herramientas, así como el entorno del negocio. Se describen las etapas de análisis y diseño del ciclo de desarrollo de la aplicación web de activos fijos y se realiza un estudio de la factibilidad del sistema para lograr tener una estimación del costo de la Aplicación WEB a desarrollar.

Palabras Claves

- ❖ ***ConDor***
- ❖ ***Activos Fijos***
- ❖ ***Aplicación Web***
- ❖ ***Modelado***

ÍNDICE

INTRODUCCIÓN	1
CAPÍTULO 1 FUNDAMENTACIÓN TEÓRICA	8
1.1 INTRODUCCIÓN.....	8
1.2 CONCEPTOS SISTEMA DE CONTABILIDAD.....	8
1.2.1 <i>Concepto general</i>	8
1.2.2 <i>Características de un sistema de información contable</i>	9
1.2.3 <i>Características generales Gestión contable de los Activos Fijos</i>	9
1.3 OBJETO DE ESTUDIO.....	9
1.3.1 <i>Descripción general</i>	10
1.3.2 <i>Descripción actual del dominio del problema</i>	12
1.3.3 <i>Situación Problemática</i>	12
1.4 ANÁLISIS DE OTRAS SOLUCIONES EXISTENTES.....	13
1.5 PROPUESTA DE SOLUCIÓN.....	15
1.6 CONCLUSIONES PARCIALES.....	15
CAPÍTULO 2 TENDENCIAS Y TECNOLOGÍAS ACTUALES A DESARROLLAR	16
2.1 INTRODUCCIÓN.....	16
2.2 SURGIMIENTO Y EVOLUCIÓN DE LA WEB.....	16
2.3 APLICACIÓN WEB.....	17
2.3.1 <i>Estructura</i>	17
2.3.2 <i>Uso en negocios</i>	17
2.3.3 <i>La tecnología .NET</i>	17
2.3.4 <i>Lenguajes de programación</i>	18
2.4 MODELO CLIENTE-SERVIDOR.....	19
2.4.1 <i>Programación del lado del cliente</i>	20
2.4.2 <i>Programación multicapas</i>	20
2.5 METODOLOGÍA.....	21
2.5.1 <i>Algunas metodologías</i>	21
2.5.2 <i>Fundamentación de la metodología utilizada. Rational Unified Process (RUP)</i>	22
2.6 LENGUAJE MODELADO OBJETO.....	23
2.6.1 <i>Lenguaje Unificado de Modelado (UML)</i>	24
2.7 HERRAMIENTAS A UTILIZAR.....	24
2.7.1 <i>Herramientas de modelado</i>	25
2.7.1.1 <i>Rational Rose Enterprise Edition</i>	25
2.7.1.2 <i>ERwin® Data Modeler</i>	25
2.7.2 <i>Herramientas de desarrollo</i>	26

2.7.2.1 Microsoft Visual Studio 2005.....	26
2.7.2.2 SQL Server 2005 Express	26
2.8 LENGUAJES DE PROGRAMACIÓN	27
2.8.1 C#.....	27
2.8.2 SQL (Structure Query Language).....	28
2.9 CONCLUSIONES PARCIALES.....	29
CAPÍTULO 3 PRESENTACIÓN DE LA SOLUCIÓN PROPUESTA.....	30
3.1 INTRODUCCIÓN.....	30
3.2 OBJETO DE AUTOMATIZACIÓN.....	30
3.3 INFORMACIÓN QUE SE MANEJA.....	30
3.4 PROPUESTA DE SISTEMA	31
3.5 MODELO DE NEGOCIO	31
3.5.1 Actores del negocio	31
3.5.2 Trabajadores del Negocio.....	32
3.5.3 Diagrama de casos de Uso del Negocio	34
3.5.4 Descripción de casos de uso del negocio.....	35
3.5.5 Modelo de objeto	39
3.6 ESPECIFICACIÓN DE LOS REQUERIMIENTOS DEL SOFTWARE.....	39
3.6.1 Actores del sistema a automatizar: módulo Activo Fijo	39
3.6.2 Requisitos funcionales	41
3.6.4 Diagrama de jerarquía de actores	47
3.6.5 Diagrama de relación entre paquetes del sistema a automatizar: Módulo de Activo Fijo.....	47
3.6.5 Diagrama de casos de uso del sistema a automatizar: Módulo de Activo Fijo.....	48
3.6.6 Descripción de los casos de uso del sistema a automatizar: Módulo de Activo Fijo	50
3.7 DEFINICIÓN DE LOS REQUISITOS NO FUNCIONALES.....	50
3.8 CONCLUSIONES PARCIALES.....	52
CAPÍTULO 4 CONSTRUCCIÓN DE LA SOLUCIÓN PROPUESTA.....	53
4.1 INTRODUCCIÓN.....	53
4.2 ANÁLISIS	53
4.3 PRINCIPIOS DE DISEÑO	54
4.3.1 Clases de la Capa Presentación.....	55
4.3.2 Clases de la Capa Lógica del Negocio.....	55
4.3.3 Clases de la Capa Acceso a Datos.....	56
4.4 PATRONES.....	56
4.4.1 Patrones del diseño.....	57
4.5 DISEÑO	57
4.5.1 Diagrama de clases del diseño	57
4.5.2 Diseño de la Base de Datos	57
4.5.2.1 Modelo lógico de datos.....	59

4.5.2 .2 Modelo físico de datos	60
4.5.3 <i>Diagrama de Despliegue</i>	61
4.6 GENERALIDADES DE LA IMPLEMENTACIÓN	62
4.6.1 PREVENCIÓN CONTRA INYECCIONES SQL.....	62
4.7 CONCLUSIONES PARCIALES	63
CAPÍTULO 5 ESTUDIO DE FACTIBILIDAD	64
5.1 INTRODUCCIÓN.....	64
5.2 PLANIFICACIÓN BASADA EN CASOS DE USO.....	64
5.3 BENEFICIOS TANGIBLES E INTANGIBLES	71
5.3.1 <i>Beneficios tangibles</i>	71
5.3.2 <i>Beneficios intangibles</i>	71
5.4 ANÁLISIS DE COSTO Y BENEFICIOS	72
5.5 CONCLUSIONES PARCIALES	72
CONCLUSIONES.....	73
RECOMENDACIONES	74
REFERENCIAS BIBLIOGRÁFICAS	75
Anexo 1. Modelo de Objetos.	78
Anexo 2. Descripciones textuales de los Casos de Uso del Sistema.....	82
Anexo 3. Diagrama de clases del diseño.....	93
Anexo 4. Tablas del Modelo de Datos.	102
GLOSARIO.....	105

ÍNDICE DE TABLAS

Tabla 1. Descripción de los actores del negocio.....	32
Tabla 2. Descripción de los trabajadores del negocio.....	33
Tabla 3. Descripción de los actores del sistema.....	40
Tabla 4. Factor de peso de los actores sin ajustar.....	65
Tabla 5. Factor de Peso de los Casos de Uso sin ajustar.....	65
Tabla 6. Factor de complejidad técnica.....	67
Tabla 7. Factor de Ambiente.....	68
Tabla 8. Distribución del Esfuerzo.....	70

ÍNDICE DE FIGURAS

Figura 1. Un proceso de desarrollo de software	23
Figura 2. Diagrama de casos de uso del negocio.....	34
Figura 3. Diagrama de jerarquía de actores	47
Figura 4. Diagrama de relación entre paquetes.....	47
Figura 5. Diagrama de casos de uso del sistema a automatizar.	48
Figura 6. Diagrama de casos de uso del paquete Gestión de la inicialización del sistema.	48
Figura 7. Diagrama de caso de uso del paquete Gestión de la explotación del sistema	49
Figura 8. Diagrama de caso de uso del paquete Administración de cierre contable.	49
Figura 9. Diagrama de caso de uso del paquete Administración de reportes e informes.....	50
Figura 10. Arquitectura WEB de tres niveles.	55
Figura 11. Capa de Presentación	55
Figura 12. Capa Lógica del Negocio.....	56
Figura 13. Capa de acceso a Datos.	56
Figura 14. Modelo lógico de datos.....	59
Figura 15. Modelo físico de datos.....	60
Figura 16. Diagrama de despliegue.....	61

INTRODUCCIÓN

Durante los primeros años de la informática, el *software* se consideraba como un añadido. La programación era un "arte" para el que no existían metodologías, era un proceso que se realizaba sin ninguna planificación. (*El por qué de la metodología 2003*)

En una segunda etapa (a partir de mitad de la década de 1960) se estableció el software como producto y aparecieron las empresas dedicadas al desarrollo y distribución masiva del mismo. El origen del término Ingeniería del Software, se atribuye a dos conferencias organizadas por la OTAN. (*El por qué de la metodología 2003*)

La tercera etapa comenzó a mediados de la década de 1970, época en la que los sistemas informáticos aumentaron mucho en su complejidad, y nacieron las redes de ordenadores. Esto supuso mucha presión para los desarrolladores, aunque los ordenadores para uso personal, apenas estaban difundidos. Esta época acabó con la aparición de los microprocesadores. (*El por qué de la metodología 2003*)

La cuarta era de la evolución de los sistemas informáticos, comienza hacia 1990 y se dirige al impacto colectivo de los ordenadores y el software, en todos los entornos. La industria del software tiene un gran peso en la economía mundial. Aparecen las técnicas de redes neuronales, junto con la lógica difusa. (*El por qué de la metodología 2003*)

A partir de todo este desarrollo se reconoce que las técnicas de los analistas no son eficientes, por el alto costo de realización y mantenimientos del software.

Para resolver este problema surge la **Ingeniería de Software**¹, que no es más que técnicas y herramientas de desarrollo. Luego surgen una gran variedad de metodologías. (ZAVALA 2000)

A lo largo de los años, el desarrollo de los proyectos de software ha causado bastantes confusiones y malas interpretaciones en los requerimientos de los clientes y usuarios, en parte debido a la abundancia

¹ Ingeniería de Software (Ver glosario)

de notaciones, metodologías y conceptos, haciendo que los desarrolladores de sistemas no se pongan de acuerdo en qué es lo que realmente están elaborando.

Al ver todo este caos, las principales empresas del mundo de la industria de la informática desarrollan una notación estándar **UML**² y un proceso de desarrollo **RUP**³. De forma paralela a la evolución de estas técnicas y herramientas, se desarrollan las tecnologías de las telecomunicaciones y con ello la necesidad de la transmisión de datos, la conectividad entre computadoras para compartir recursos e información, con independencia del lugar físico donde se encuentren, como producto del avance en ambos temas y el esfuerzo realizado por varias empresas surgen lo que se conoce hoy como **Internet**⁴, conocida como la Red de Redes.

Pero Internet tuvo también su evolución en el tiempo, en sus inicios la forma de obtener los datos era un poco primitiva, poco amigable para el usuario y dominada solamente por expertos en el tema, es por eso que se introducen nuevas técnicas y nociones, entre ellos el concepto **Web**⁵.

En Ingeniería del Software una **Aplicación Web** es aquella que los usuarios usan accediendo a un servidor Web a través de Internet o de una *intranet*. Las Aplicaciones Web son populares debido a la practicidad del *navegador Web* como cliente ligero. La habilidad para actualizar y mantener *Aplicaciones Web* sin distribuir e instalar software en miles de potenciales clientes es otra razón de su popularidad.

Cuba no ha estado ajena a esta evolución de la informática y las comunicaciones que se ha llevado a cabo desde la segunda mitad del pasado siglo. (*Manual de desarrollo web* 2004). Todos estos avances tecnológicos se han implementado en las diferentes Ministerios de nuestro país, tal es el caso del **MITRANS**⁶ que ha desarrollado Aplicaciones Web que facilitan la Gestión Empresarial y una de las áreas donde mayor interés cobra es en el Departamento Económico – Contable.

² lenguaje estándar para escribir planos de software.

³ Proceso de desarrollo de software configurable que se adapta a través de los proyectos variados en tamaños y complejidad.

⁴ Resultado de la interconexión de miles de computadoras de todo el mundo.

⁵ Posibilidad de lectura universal, que consiste en que una vez que la información esté disponible, se pueda acceder a ella desde cualquier *ordenador* desde cualquier país y por cualquier persona autorizada; fue un paso significativo en el desarrollo de **Internet**.

⁶ Ministerio del Transporte.

Todo esto pone de manifiesto el nuevo oriente que debe seguir la contabilidad y el profesional contable, pues la contabilidad es, quizás, una de las actividades más importante dentro del campo de los negocios, dada su naturaleza de informar acerca del incremento de la riqueza, la productividad y el posicionamiento de las empresas en los ambientes competitivos.

La importancia de los sistemas de información contable radica en la utilidad que tienen estos tanto para la toma de decisiones de los socios de las empresas, como por la información que brinda, a aquellos usuarios externos, sobre la salud de la empresa. (*LA CONTABILIDAD Y EL SISTEMA CONTABLE* 2003)

Los sistemas de información han desempeñado un importante lugar en el mercado, como elemento de control y organización, sin embargo no debe limitarse solo a esa función, dado que nos encontramos en la necesidad de buscar el poder explicativo de los sistemas de información en el comportamiento de los usuarios con la información empresarial, así no solo se busca un manejo técnico en la empresa sino que se tomen todos los componentes de la empresa tanto tangibles como intangibles, que nos permitan competir en un entorno que es cada vez más fuerte y más agresivo. Es entonces cuando surge la posibilidad de realizar un sistema que gestione los **activos fijos** de las empresas, con el objetivo de hacer más eficiente el control de estos.

SICS⁷ es una entidad subordinada al **MITRANS**, empresa que dedica sus acciones al desarrollo y comercialización de productos y servicios informáticos. Encontramos inmersos en un entorno económico cada vez más competitivo, hace necesario el uso de aplicaciones que cuenten con la funcionalidad de los sistemas multiusuario y las facilidades que brinda el ambiente Web, para de esta forma aportar un instrumento útil en el control de los activos fijos en las empresas y que proporcione un alto grado de visibilidad de las distintas actividades que sobre estos activos se realice.

Actualmente SICS ha desarrollado y comercializado un Sistema de Gestión Económico- Contable llamado ConDor, conformado fundamentalmente por 7 módulos que proporcionan el control Económico de la Empresa, entre ellos se encuentra el Módulo de Activos Fijos. Una de las mayores debilidades que posee

⁷ Servicios Informáticos de Consultoría y Sistema.

dicho Sistema es que la Gestión la hace de manera ineficiente y engorrosa, además de ser un Sistema Monousuario y no poseer una Base de datos única.

A partir de la necesidad de certificar dicho sistema gestión contable (**ConDor**)⁸ para cumplir con la normativa orientada por el **MEP**⁹, el **MEF**¹⁰ y el **MIC**¹¹, de que las aplicaciones deben ser multiusuario y contar con una base de datos única que pueda ser auditable en todo momento, trae como consecuencia la siguiente **Situación Problemática**: el sistema gestión contable (**ConDor**) no cuenta con la funcionalidad de un sistema multiusuario y una base de datos única, además de que el trabajo se ralentiza debido a dificultades en el diseño del mismo. Donde el **problema a resolver** sería la no existencia de una Aplicación Web que realice el control interno y la contabilización de los Activos Fijos Tangibles e Intangibles; de acuerdo a las normativas dictadas por el Ministerio de Finanzas y Precios y el Ministerio de Economía y Planificación.

De ahí surge la necesidad de desarrollar una aplicación Web, teniendo como base el Sistema ConDor, que permita el control de los activos fijos en las empresas de forma eficiente y agilice el trabajo con dicha aplicación a fin de economizar el máximo de tiempo de trabajo y donde la premisa fundamental sea la de buscar la mayor eficiencia, y sea capaz de realizar operaciones que agilicen el control de los recursos de los que dispone la empresa, en este caso, los **activos fijos**.

Para ello se propone una aplicación Web que facilite el trabajo de los ejecutivos de las empresas, ya que mediante la misma se dispondrá de datos y reportes referentes a todo lo relacionado con los activos fijos de las empresas.

Para la modelación del sistema se escogió el desarrollo de la aplicación Web que permita controlar los activos fijos en las empresas de forma eficiente, con un mínimo de esfuerzo, mediante el uso de una base de datos única y una interfaz amigable al usuario.

⁸ Sistema de gestión contable.

⁹ Ministerio de Economía y Planificación

¹⁰ Ministerio de Finanzas y Precios

¹¹ Ministerio de Informática y Comunicaciones.

Una de las principales metas propuestas es lograr una aplicación capaz de satisfacer las necesidades actuales del mercado y la de los clientes, mediante el intercambio de ideas en aras de lograr un mejor trabajo.

El **Objeto de Estudio** de este trabajo es el desarrollo de una aplicación WEB para el módulo de Activos Fijos del sistema Económico – Contable ConDor, siendo el **campo de acción** el entorno de desarrollo de las aplicaciones web.

Como **Hipótesis** se parte de la idea de que si se desarrolla una aplicación multiusuario con una interfaz amigable relacionada con una base de datos, es posible lograr llevar a cabo de forma rápida y eficiente un control de los activos fijos que forman parte del estado contable y financiero de la empresa, dando como resultado la creación de la aplicación Web para activos fijos.

Los **Objetivos Generales** de este trabajo son:

- ❖ Modelar una aplicación Web para la gestión de activos fijos.

De aquí se derivan los siguientes **Objetivos Específicos**:

- Realizar un estudio detallado del sistema contable de activos fijos (ya existente del **CONDOR**) para conocer su funcionamiento.
- Llevar a cabo un estudio de la contabilidad en **SICS**.
- Poner a disposición de los directivos la mayor cantidad de información, hechos y circunstancias sobre la actividad.
- Modelar la aplicación Web de control de activos fijos.

Para alcanzar los objetivos, se llevarán a cabo las siguientes **Tareas**:

- ✓ Realización de un estudio del entorno de trabajo.
- ✓ Declaración de los requisitos que debe cumplir el sistema de acuerdo con sus necesidades.
- ✓ Declaración de los ciclos de desarrollo.
- ✓ Identificación de las necesidades de la institución y del sistema en general.
- ✓ Diseño de una aplicación web y la BD.

Es necesario tener en cuenta la realización de la modelación de una aplicación Web para el módulo de Activos Fijos con una interfaz amigable, cómoda y eficiente para los usuarios del mismo, de allí que la **idea a defender** es realizar la modelación de una aplicación Web que garantice la eficiencia en sus procesos de gestión de Activos Fijos ganando en tiempo de respuesta a los usuarios y minimizando los errores.

Los **métodos de investigación científica** que se utilizaron en este trabajo son:

Métodos Teóricos:

- Inductivo – deductivo: Se analiza de lo general a lo particular el proceso.
- Analítico – sintético: analizar teóricamente el proceso de la contabilidad de activos fijos y sintetizarlo.
- Modelación: crear modelos a través de la metodología RUP.

Métodos Empíricos:

- Observación: Se realizaron visitas al departamento de contabilidad del SICS y de la UCI para observar los procesos, alcanzando el entendimiento de los procesos de gestión de activos fijos.

- Encuesta: El uso de la encuesta a las personas que interactúan con la aplicación se utilizan como diagnostico inicial del problema a resolver.
- Entrevista: Se realizan entrevistas al personal del departamento de contabilidad y a todo el grupo de trabajo que participó en la elaboración de la primera versión del software, con el objetivo de precisar el problema a resolver.
- Búsquedas en Internet y base de datos sobre temas relacionados con el problema.

Los **resultados esperados** son:

- Que SICS pueda tener la modelación del nuevo sistema **ConDor** para su futura implementación.
- Base de datos única para todos los módulos del sistema.

Este trabajo está formado por 5 capítulos:

Capítulo 1: Se realizó la Fundamentación teórica del trabajo, dentro del mismo se expone el objeto de estudio, se realiza la descripción del proceso de Activos Fijos, por otra parte se describen detalles de otros sistemas existentes, similares al **ConDor** y otros detalles fundamentales.

Capítulo 2: Se exponen las tendencias y tecnologías actuales necesarias para la modelación de la aplicación y para una futura implementación.

Capítulo 3: Se expone una solución del sistema, pero antes se describe el negocio de Activos Fijos, se definen las funcionalidades y se hace un análisis del sistema a desarrollar.

Capítulo 4: Se realizan todos los diagramas de clases del análisis y diseño, modelo de datos y diagrama de despliegue para así enfocar la construcción de la solución propuesta. Además se plantean los principios del diseño.

Capítulo 5: Se realiza un estudio de la factibilidad del sistema, logrando obtener un aproximado del costo del desarrollo del sistema, así como los beneficios tangibles e intangibles.

Capítulo 1

Fundamentación Teórica

1.1 Introducción

El presente capítulo brinda información de los temas sobre los que se desarrolla el sistema, se definen aspectos importantes de la teoría en que se basa la solución del problema, facilitando de esta manera la comprensión del mismo. Se hace referencia, además, a la actualidad internacional y nacional en el tema relacionado con el trabajo. Se describen los principales conceptos asociados al dominio del problema que son necesarios para entender la propuesta de solución.

1.2 Conceptos Sistema de Contabilidad

1.2.1 Concepto general

Un Sistema de Contabilidad es una estructura organizada mediante la cual se recogen las informaciones de una empresa como resultado de sus operaciones, valiéndose de recursos como formularios, reportes, libros, etc. y que presentados a la gerencia le permitirán a la misma tomar decisiones financieras.

Un Sistema de Contabilidad no es más que normas, pautas, procedimientos, etc., para controlar las operaciones y suministrar información financiera de una empresa, por medio de la organización, clasificación y cuantificación de las informaciones administrativas y financieras que se nos suministre. Para que un sistema de contabilidad funcione eficientemente es preciso que su estructura-configuración cumpla con los objetivos trazados. Esta red de procedimientos debe estar tan íntimamente ligada, de manera tal que integre el esquema general de la empresa y de esta forma pueda ser posible realizar cualquier actividad importante de la misma. (MIGUEL ANGEL MARTINEZ)

1.2.2 Características de un sistema de información contable

Un sistema de información bien diseñado ofrece control, compatibilidad, flexibilidad y una relación aceptable de costo / beneficio.

Control: un buen sistema de contabilidad le da a la administración control sobre las operaciones de la empresa. Los controles internos son los métodos y procedimientos que usa un negocio para autorizar las operaciones, proteger sus activos y asegurar la exactitud de sus registros contables.

Compatibilidad: un sistema de información cumple con la pauta de compatibilidad cuando opera sin problemas con la estructura, el personal, y las características especiales de un negocio en particular. (LA CONTABILIDAD Y EL SISTEMA CONTABLE 2003)

1.2.3 Características generales Gestión contable de los Activos Fijos

Los activos fijos están formados por las propiedades de naturaleza relativamente permanente, que se emplean en una empresa para sus operaciones sin la intención de venderlos.

Los activos fijos pueden ser: tangibles o Intangibles.

- **Tangibles:** si tiene sustancia corpórea, es decir físicos. Como por ejemplo:
 - **Computadoras** - sujetas a depreciación
 - **Bosques** - sujetos a agotamiento
- **Intangibles:** aquellos que no son físicos, por lo general resultan difíciles de controlar, por ejemplo: software, patentes, derecho de autor, donde su valor reside en los derechos que posee el propietario.

Los activos están sujetos a depreciación, lo cual no es más que la distribución equitativa del costo adquirido de un bien más las inversiones realizadas en este para usarse en un período estimado de vida.

1.3 Objeto de Estudio

El objeto de estudio de este trabajo se encuentra centrado en desarrollar la Aplicación WEB del Módulo de Activos Fijos del Sistema Económico – Contable ConDor a partir del módulo existente, el cual no

posee la funcionalidad de ser un sistema multiusuario, siendo este su principal inconveniente, ni contar con una base de datos única, donde las trabas en el diseño ralentizan el trabajo con el mismo según los usuarios del sistema actual. Para cumplir con el objeto de estudio antes expuesto, es necesario realizar una modelación de la aplicación, además de realizar un estudio detallado del sistema ya existente como base para la realización del nuevo sistema.

1.3.1 Descripción general

El sistema de gestión contable **CONDOR** es una suite integrada por módulos que pueden ser fácilmente adaptados a las particularidades de cualquier cliente, independientemente de las características que posea o la complejidad de sus operaciones.

Los módulos que posee el sistema son:

- ***Contabilidad general***
- ***Activos fijos***
- ***Inventario***
- ***Nomina***
- ***Prenómina***
- ***Condexce***
- ***Efectos***

El Módulo Control de Activos Fijos está constituido por dos etapas bien definidas que son: la ***Carga Inicial de los Datos*** y la ***Explotación del Sistema***.

La primera etapa culmina con el Cierre de Apertura y da paso a la segunda etapa, o sea, la Explotación del Sistema. La Carga Inicial de los Datos es muy importante, porque de la calidad de la misma depende la futura Explotación del Sistema.

Carga inicial de los datos.

La Carga Inicial de los datos incluye la creación del Tarjetero de Activos Fijos, a partir de los diferentes SubMayores de Activos Fijos, la revisión minuciosa de toda la información, la impresión de los Reportes por Cuenta y por Área, como constancia de la Carga Inicial, y la impresión de la Comprobación de Saldos Cuadrados del Tarjetero con Contabilidad General.

Apertura de datos

La operación de Apertura de los Datos consiste en crear el Tarjetero de Activos Fijos, a partir de la información que corresponde a cada uno de los SubMayores establecidos, en esta etapa se declaran los nomencladores que no son más que los que contienen los códigos que se usarán en las diferentes opciones del Sistema, los mismos pueden ser cambiados de acuerdo a los requerimientos de cada cliente del Sistema.

Nomencladores

Contienen los códigos que se usarán en las diferentes opciones del Sistema, los mismos pueden ser establecidos y cambiados de acuerdo a los requerimientos de cada cliente del Sistema.

Los Nomencladores se entregan con la codificación del **MITRANS**.

- ✓ ***Centro de costo.***
- ✓ ***Áreas.***
- ✓ ***Cuentas de movimiento.***
- ✓ ***Cuentas de valor y depreciación.***
- ✓ ***SubElemento de gasto.***
- ✓ ***SubCuenta de activo.***
- ✓ ***SubCuentas depreciación.***
- ✓ ***SubCuentas corrientes.***
- ✓ ***Situación de Activo Fijo.***
- ✓ ***Componente de obra.***
- ✓ ***Planilla de contabilización.***

Cierre de apertura.

Esta operación será realizada cuando todos los Activos Fijos han sido introducidos y además que la información está correcta después de una exhaustiva revisión, pues para que concluya con éxito no pueden existir incoherencias en los Activos Fijos que se han introducido.

Tarjetero

Formulario que permite registrar los datos que corresponden a los diferentes grupos de Activos Fijos para su adecuada identificación y control, tanto físico como contable.

Explotación del sistema

La explotación del sistema comienza con la entrada a la BD de cada uno de las tarjetas que dan la entrada y salida a los productos en el almacenen. Todo este movimiento de Activos Fijos incluyen todas las operaciones de altas, bajas, traslados y otras actualizaciones realizadas en el mes donde cada movimiento actualiza el tarjetero y genera los registros contables que deben ser revisados y en caso necesario corregidos antes de efectuar el Cierre de Mes.

1.3.2 Descripción actual del dominio del problema

Actualmente el trabajo contable en las empresas que utilizan el sistema gestión contable **CONDOR** se realiza sobre una aplicación de escritorio que ralentiza la gestión empresarial necesaria hoy día para mantener la competitividad en el mercado. Además no cumple con las normas orientadas por el **MIC** las cuales son imprescindibles para poder continuar la comercialización de **CONDOR** a nivel nacional con otras empresas, estas normativas incluyen un aspecto fundamental a cumplir en este tipo de sistemas, el cual sería la funcionalidad de que las aplicaciones deben ser multiusuario.

1.3.3 Situación Problemática

El sistema de gestión contable **CONDOR** no cuenta con la funcionalidad de un sistema multiusuario y se ve afectado su funcionamiento debido a problemas existentes en el diseño del mismo.

El no contar con la funcionalidad multiusuario implica inconvenientes con el funcionamiento del mismo que provocan pérdida de tiempo y de dinero, así como una mayor exigencia de hardware.

Entre estos inconvenientes se destacan por su relevancia los siguientes:

- El no contar con una base de datos única; aun cuando gran parte de la información es la misma cada módulo cuenta con su base de datos, lo que además de tener información repetida posibilita un amplio margen al error.
- El impedimento para realizar operaciones sobre el mismo módulo por varias personas; actualmente en cada módulo solo puede trabajar una sola persona sin poder recibir ayuda alguna, lo que se traduce en pérdida de tiempo que implica pérdida de dinero.
- Por otra parte para su uso se hace necesario instalarlo en todas las computadoras donde vaya a ser usado; además de exigir igual hardware en cada una de las máquinas donde va a ser usado, significa tiempo de instalación y mantenimiento, así como exige conocimientos de informática del personal que vaya a utilizar el software.

1.4 Análisis de otras soluciones existentes

Los **ERP** (*Sistemas de Planificación de Recursos*) de las empresas son sistemas de gestión de información que integran y automatizan muchas de las prácticas de negocio asociadas con los aspectos operativos o productivos de una empresa.

Un ERP es de obligación hoy en día para poder tomar mejor una decisión. La integración de la información de los procesos de negocio de una empresa es fundamental ya no para tener ventaja competitiva sino para simplemente competir. (*ERP Sistema de Gestión 2005*)

En Cuba

- ***Versat-Sarasola***

Sistema cubano de contabilidad confiable, que permite enviar información eficaz, de forma inmediata, desde lugares apartados, a la vez que ofrece mayor organización, control y disciplina en cada gestión.

Fue éste el primer sistema de contabilidad cubano certificado, en cuya evaluación participaron el Ministerio de Finanzas y Precios, Consultorías Internacionales y el Organismo encargado de la seguridad informática.

Sistema económico integrado. Constituido por 12 módulos que incluyen configuración y seguridad, contabilidad general y de gastos, costos y procesos, análisis económico empresarial y control de activos fijos, además, interviene finanzas y cajas, planificación y presupuestos, control de inventarios, de productos terminados, pago de salario, paquete de gestión, contratación y facturación.

Entre sus principales beneficios se pueden citar:

- Al evitarse la importación del sistema foráneo, el país se ahorró un millón 186 mil dólares que costaban las licencias, más la labor de consultoría, atenciones y otros gastos. Entre otras ventajas permitió la elaboración de un software similar o superior al que se pensaba adquirir y le agregamos nuevas tecnologías.
- Actualmente lo utilizan alrededor de 200 entidades de varias provincias y en lo adelante lo introducirán más de dos mil 500 unidades presupuestadas del país, entre las que figuran Organismos de la Administración Central del Estado, las Direcciones Municipales de Finanzas, tesorerías, la **ONAT** entre otros. (MARTIN 2005.)

- **SAP R/3**

El sistema SAP está compuesto de una serie de módulos funcionales que responden de forma completa a los procesos operativos de las compañías. Un resumen de módulos de SAP R/3son:

-FI (Gestión de Finanzas)

-CO (Control de Gestión)

-PS (Gestión de proyectos)

-MM (Gestión de Materiales)

-RE (Gestión de bienes Inmuebles)

-AM (Gestión de Activos fijos)

-SL (Gestión de Promociones)

Según consideraciones de usuarios del producto **SAP R/3**, para la gestión contable de las empresas, se considera como el mejor de su tipo ya que presenta como principales ventajas sobre otros software afines su funcionalidad, seguridad, apertura, reputación. Aunque reconocen que aun presenta como principal desventaja el alto precio del producto así como la complejidad que trae consigo su explotación.

1.5 Propuesta de solución

Tras analizar la situación actual sobre el objeto de estudio que tiene este trabajo, se llegó a la conclusión que es necesario elaborar un sistema que pueda garantizar las funciones de una aplicación de gestión contable. Para ello se ha decidido realizar un sistema Web que debe garantizar la seguridad y confiabilidad en todos los aspectos referente a la información contable de las empresas, disminuyendo el por ciento de errores, aspecto muy importante en la contabilidad de las mismas, brindando resultados específicos a los directores, auditores y consultores, también debe facilitar y agilizar el trabajo de las empresas y ayudar en un importante por ciento a la toma de decisiones por parte de los directivos de dichas empresas. La aplicación estará conformada por varios módulos que al integrarse formarán el sistema Web, que contará con base de datos única para todos los módulos y permitirá trabajar en una plataforma multiusuario.

1.6 Conclusiones parciales

En este capítulo se detallaron las condiciones y problemas que rodean al objeto de estudio, donde a través de los conceptos y definiciones planteadas, se determinaron las especificidades que rodean al problema del sistema Web para la gestión contable en la parte de activos fijos y en base a esto se obtuvieron los objetivos generales y específicos para este trabajo.

Además se abordaron temas acerca de la actualidad del sistema, también se analizaron soluciones existentes en nuestro país y en el mundo, dando así paso al problema científico que se presenta. Se aborda además una panorámica de la situación actual del **CONDOR**.

Capítulo **2**

Tendencias y Tecnologías actuales a desarrollar

2.1 Introducción

En este capítulo se hace un análisis de las tendencias y tecnologías a nivel internacional en el desarrollo de las aplicaciones Web. Se fundamentan las herramientas de diseño a utilizar como Rational Rose Enterprise Edition y ERwin® Data Modeler, así como la metodología a seguir y el lenguaje en que se va a implementar la propuesta.

2.2 Surgimiento y evolución de la Web

Los Servicios Web surgieron ante una necesidad de estandarizar la comunicación entre distintas plataformas. Ya instalados como protagonistas dentro del tercer milenio nos encontramos inmersos en un mundo plagado de información que circula a lo largo y ancho de nuestro planeta. La difusión de la información se ha visto favorecida enormemente por los avances tecnológicos que se han generado en cuanto a la velocidad de transmisión de datos, debido esto a la evolución tan importante que se ha dado en materia de la tecnología de la comunicación sobre todo en las dos últimas décadas.

Esta gran cantidad de datos ha encontrado un medio ideal de comunicación que con la unión de las tecnologías de la informática y los medios electrónicos ha generado nuevas tendencias en el ejercicio comunicativo, con lo que se ha venido a crear un nuevo medio de comunicación que viene a sumarse y a enriquecer a los medios ya conocidos. A este nuevo medio de comunicación se le ha denominado INTERNET, el cual ha venido a facilitar de manera instantánea la posibilidad de obtener datos precisos de los sucesos, acontecimientos e investigaciones del pasado y presente.

Internet se ha convertido en un medio de comunicación masivo superando el límite que lo calificaría únicamente como medio de información, siendo de esta manera un medio que ha rebasado fronteras, ya que ha llegado a los lugares más recónditos, entrelazados por medio de esta inmensa Red para obtener

datos, informaciones, incluso diálogos, imágenes y sonidos en tiempo real sobre el área que se solicite con una mayor efectividad de información. (SELLERS 2001)

2.3 Aplicación Web

Una aplicación Web es una aplicación informática que los usuarios utilizan accediendo a un servidor Web a través de Internet o de una intranet. Las aplicaciones Web son populares debido a la practicidad del navegador Web como cliente, siendo la utilidad que brinda para actualizar y mantener aplicaciones Web sin necesidad de distribuir e instalar software en los clientes, esta es precisamente una razón más de su popularidad. Aplicaciones como los webmails, wikis, weblogs, MMORPGs y las tiendas en línea son ejemplos bien conocidos de aplicaciones Web. (*Desarrollo web 2000*)

2.3.1 Estructura

Aunque muchas variaciones son posibles, una aplicación Web está comúnmente estructurada como una aplicación de tres-capas. En su forma más común, el navegador Web es la primera capa, un motor usando alguna tecnología Web dinámica (ejemplo: CGI, PHP, Java Servlets o ASP) es la capa media, y una base de datos como última capa. El navegador Web manda peticiones a la capa media, que la entrega valiéndose de consultas y actualizaciones a la base de datos generando una interfaz de usuario.

2.3.2 Uso en negocios

Una estrategia que está emergiendo para las empresas proveedoras de software, es proveer acceso vía Web al software. Para aplicaciones previamente distribuidas como de escritorio, esto puede requerir el desarrollo de una aplicación totalmente nueva o simplemente adaptar la aplicación para usar una interfaz Web. Estos programas permiten al usuario pagar una cuota mensual o anual para usar la aplicación, sin necesidad de instalarla en la computadora del usuario. Las compañías que siguen esta estrategia son llamadas Proveedores de Aplicaciones de Servicio (ASP por sus siglas en inglés), este modelo de negocios está atrayendo la atención de la industria del software.(SELLERS 2001)

2.3.3 La tecnología .NET

Microsoft.NET es el conjunto de nuevas tecnologías en las que Microsoft ha estado trabajando durante los últimos años, con el objetivo de obtener una plataforma sencilla y potente para distribuir el software en

forma de servicios, de forma tal que puedan ser suministrados remotamente, comunicándose y combinándose unos con otros totalmente independiente de la plataforma, lenguaje de programación y modelo de componentes con los que hayan sido desarrollados. (*Universidad .NET*) .NET ofrece un entorno de desarrollo de aplicaciones llamado *Visual Studio* .NET que consta de varios lenguajes de programación como Visual Basic .NET, Visual C#, Visual FoxPro y Visual C++ .NET. Estos lenguajes combinan las características de los lenguajes existentes con nuevas posibilidades para proporcionar un potente sistema de desarrollo. A continuación, se detallan algunas de las características de la Arquitectura .NET.

2.3.4 Lenguajes de programación

Existen numerosos lenguajes de programación utilizados para el desarrollo de Aplicaciones Web, entre los que destacan:

- PHP
- ASP/ASP.NET
- JSP
- Perl

2.3.5 Sistemas Gestores de Base de Datos (SGBD)

Un Sistema de Gestión de Bases de Datos (SGBD) no es más que un conjunto de programas, procedimientos y lenguajes que proporcionan las herramientas necesarias para trabajar con una base de datos, incorporar una serie de funciones que permitan definir los registros, sus campos, sus relaciones, insertar, suprimir, modificar y consultar los datos. Actualmente existen numerosos sistemas gestores de bases de datos, entre ellos el Microsoft Access, Oracle, MySQL, Visual Fox Pro y SQL Server.

En la elaboración de esta aplicación se hace uso del gestor de bases de datos relacionales SQL Server 2005 Express, es importante declarar que no es el mejor sistema gestor ya que es aventajado en los aspectos de disponibilidad de versiones para múltiples plataformas, por otros como Oracle y MySQL, pero SQL Server se ajusta a las necesidades de este proyecto. Podemos mencionar además que este gestor de bases de datos cuenta con un lenguaje para programar procedimientos almacenados y triggers, es más fácil de usar que el Oracle y más potente que MySQL, brinda una alta seguridad administrada según

perfiles configurables que puede ser sobre cuentas de usuarios propias integradas con las de Windows. Ofrece, además, la posibilidad de importar los datos en múltiples formatos, por otro lado Visual C# cuenta con un proveedor ADO.Net nativo para SQLServer, asimismo SQL Server ha sido desarrollado por Microsoft para explotar al máximo las posibilidades de los sistemas operativos Windows.

Por todas estas razones se decidió optar por la utilización del sistema gestor de bases de datos (**Microsoft SQL Server 2005 Express**). (MIGUEL CASTAÑO 1999)

2.4 Modelo Cliente-Servidor

La arquitectura cliente-servidor es una forma de dividir las responsabilidades de un Sistema de Información separando la interfaz de usuario (Nivel de presentación) de la gestión de la información (Nivel de gestión de datos). Esta arquitectura consiste básicamente en que un programa, el Cliente informático realiza peticiones a otro programa, el servidor, que les da respuesta. Aunque esta idea se puede aplicar a programas que se ejecutan sobre una sola computadora es más ventajosa en un sistema multiusuario distribuido a través de una red de computadoras.

En esta arquitectura, la capacidad de proceso está repartida entre los clientes y los servidores, aunque son más importantes las ventajas de tipo organizativo debidas a la centralización de la gestión de la información y la separación de responsabilidades, lo que facilita y clarifica el diseño del sistema. La separación entre cliente y servidor es una separación de tipo lógico, donde el servidor no se ejecuta necesariamente sobre una sola máquina ni es necesariamente un sólo programa.

Una disposición muy común son los sistemas multicapa en los que el servidor se descompone en diferentes programas que pueden ser ejecutados por diferentes computadoras aumentando así el grado de distribución del sistema. La arquitectura cliente-servidor sustituye a la arquitectura monolítica en la que no hay distribución, tanto a nivel físico como a nivel lógico. (*Cliente-servidor* 2006)

Ventajas de la arquitectura cliente-servidor.

Centralización del control: los accesos, recursos y la integridad de los datos son controlados por el servidor de forma que un programa cliente defectuoso o no autorizado no pueda dañar el sistema.

Escalabilidad: se puede aumentar la capacidad de clientes y servidores por separado. Se reduce el tráfico de red considerablemente. Idealmente, el cliente se comunica con el servidor utilizando un protocolo de alto nivel de abstracción como por ejemplo SQL.

2.4.1 Programación del lado del cliente

Cliente:

Es el encargado de los servicios mediante una aplicación llamada cliente. En el caso que ocupa el presente trabajo, se recomienda utilizar AJAX como programación del lado del cliente, ya que esta técnica de desarrollo Web para crear aplicaciones que responden a las acciones del usuario sin refrescar la página contra el servidor. Esto posibilita una mayor rapidez en las respuestas por parte de la aplicación a las peticiones del usuario.

Con la programación en el lado del cliente se validan los datos en el cliente antes de ser enviados al servidor, y facilita mostrar mensajes de error ante cualquier acción incorrecta del usuario. Para lograr este efecto, se utilizan la mayoría de las tecnologías disponibles para páginas Web tales como: HTML, CSS, XML y JavaScript.

Servidor

La programación del lado del servidor se considera un elemento agregado de suma importancia. Este tipo de programación nos brinda la posibilidad de ejecutar programas en un gran sistema, además permite el manejo de una gran cantidad de datos de forma dinámica. Actividad que queda fuera del alcance del HTML y la programación del lado del cliente.

2.4.2 Programación multicapas

La programación en múltiples capas es la técnica más efectiva en aplicaciones empresariales, debido a la fácil administración que implica el dividir los componentes de la aplicación en capas y la rapidez que esto implica en programas orientados a Cliente-Servidor. Este tipo de programación consiste en dividir los componentes primarios de la aplicación, programarlos por separado y luego unirlos sea en tiempo de ejecución o en el mismo código.

Se definen tres capas, tal es el caso de: Presentación, Reglas del Negocio y Acceso a Datos.

- **Presentación:** Esta capa constituye la interfaz gráfica y la interacción entre el usuario y el sistema.
- **Reglas del Negocio:** Contiene los métodos comunes para todas las clases con el propósito de regular las acciones del usuario.
- **Acceso a Datos:** En esta capa se programa todo lo que tiene que ver con el acceso a la base de datos. Esta capa queda encargada de tomar la información de la base de datos dada una petición de la capa de Reglas del Negocio, que a su vez es generada por la capa de presentación.

2.5 Metodología

La rama de la metodología, dentro de la ingeniería de software, se encarga de elaborar estrategias de desarrollo de software que promuevan prácticas adaptativas en vez de predictivas; centradas en las personas o los equipos, orientadas hacia la funcionalidad y la entrega, de comunicación intensiva y que requieren implicación directa del cliente.

2.5.1 Algunas metodologías

La metodología XP es una metodología exitosa porque enfatiza la satisfacción del cliente y promueve el trabajo en equipo. En XP, las actividades improductivas han sido eliminadas para reducir costos y frustraciones. Esta metodología ha sido diseñada para solucionar el eterno problema del desarrollo de software por encargo: entregar el resultado que el cliente necesita a tiempo.

CARACTERÍSTICAS

El desarrollo bajo XP tiene características que lo distinguen claramente de otras metodologías:

- Los diseñadores y programadores se comunican efectivamente con el cliente y entre ellos mismos.
- Los diseños del software se mantienen sencillos y libres de complejidad o pretensiones excesivas.
- Se obtiene retroalimentación de usuarios y clientes desde el primer día gracias a las baterías de pruebas
- El software es liberado en entregas frecuentes tan pronto como sea posible
- Los cambios se implementan rápidamente tal y como fueron sugeridos.

- Las metas en características, tiempos y costos son reajustadas permanentemente en función del avance real obtenido.

Con estas características no es sorprendente que XP sea la metodología más apropiada para un entorno caracterizado por requerimientos cambiantes originados por un mercado fluctuante y los propios avances de la tecnología y los negocios.

La metodología **Microsoft Solution Framework (MSF)** es una metodología flexible e interrelacionada con una serie de conceptos, modelos y prácticas de uso, que controlan la planificación, el desarrollo y la gestión de proyectos tecnológicos. MSF se centra en los modelos de proceso y de equipo dejando en un segundo plano las elecciones tecnológicas

MSF tiene las siguientes características:

- Adaptable.
- Escalable.
- Flexible.

2.5.2 Fundamentación de la metodología utilizada. Rational Unified Process (RUP)

Para la facilitación de la aplicación se utilizará la herramienta de la metodología RUP (Rational Unified Process) porque este proceso unificado del software contiene un conjunto de actividades necesarias para las necesidades o requisitos del usuario en un sistema de software (ver figura 1). Está basado en componentes, de allí que se deduce que un sistema software en su construcción y está formado por componentes interconectados a través de interfaces bien definidas. Utiliza el **Lenguaje Unificado de Modelado (UML)** para preparar todos los esquemas del sistema software.

Las características que definen de forma total a RUP son las siguientes:

➤ **Dirigidos por casos de uso.**

Los casos de usos expresan lo que los usuarios futuros necesitan, esto se logra cuando obtener cuando se modela el negocio y se representa a través de los requerimientos del sistema. De allí los casos de uso inclinan el proceso de desarrollo.

➤ **Centrado en arquitectura.**

La arquitectura origina una vista del sistema en la cual tanto los desarrolladores como el usuario pueden estar de acuerdo, por lo que detalla los elementos del modelo que son mas relevantes para la su desarrollo.

➤ **Iterativo e incremental.**

El proceso unificado de desarrollo propone que cada fase se desarrolle en iteraciones. Una iteración involucra actividades de todos los flujos de trabajo, aunque desarrolla fundamentalmente algunos más que otros. Este proceso unificado es repetitivo a lo largo de los ciclos de desarrollo del proceso. Cada ciclo está compuesto por cuatro fases tales como:

- ✓ Inicio.
- ✓ Elaboración.
- ✓ Construcción.
- ✓ Transición.

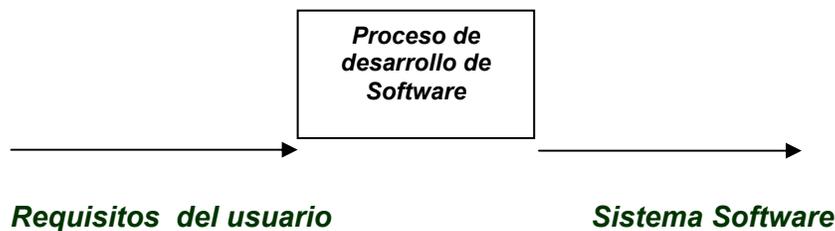


Figura 1. Un proceso de desarrollo de software

2.6 Lenguaje Modelado Objeto

El lenguaje de modelado de objetos es un conjunto estandarizado de símbolos y de modos de disponerlos para modelar (parte de) un diseño de software orientado a objetos. Algunas organizaciones los usan extensivamente en combinación con una metodología de desarrollo de software para avanzar de una especificación inicial a un plan de implementación y para comunicar dicho plan a todo un equipo de desarrolladores. El uso de un lenguaje de modelado es más sencillo que la auténtica programación, pues existen menos medios para verificar efectivamente el funcionamiento adecuado del modelo. Esto puede

suponer también que las interacciones entre partes del programa den lugar a sorpresas cuando el modelo ha sido convertido en un software funcional.

2.6.1 Lenguaje Unificado de Modelado (UML)

Lenguaje Unificado de Modelado (UML, por sus siglas en inglés, Unified Modeling Language) es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad; aún cuando todavía no es un estándar oficial, está apoyado en gran manera por el OMG (Object Management Group). Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema de software. UML ofrece un estándar para describir un "plano" del sistema (modelo), incluyendo aspectos conceptuales tales como procesos de negocios y funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y componentes de software reutilizables.(FOWLER 1999)

El punto importante para notar aquí es que UML es un "lenguaje" para especificar y no un método o un proceso. UML se usa para definir un sistema de software; para detallar los artefactos en el sistema; para documentar y construir -es el lenguaje en el que está descrito el modelo. UML se puede usar en una gran variedad de formas para soportar una metodología de desarrollo de software (tal como el Proceso Unificado de Rational) -pero no especifica en sí mismo qué metodología o proceso usar.

Se hace uso de UML ya que:

- Permite documentar todos los artefactos de un proceso de desarrollo.
- Puede conectarse con lenguajes de programación.
- Por la metodología que se utiliza para la realización de este trabajo.
- UML es independiente del proceso.
- Permite modelar aplicaciones o sistemas manejando técnicas orientas a objetos.

2.7 Herramientas a utilizar

La realización de la propuesta del sistema requiere la utilización de disímiles herramientas, dichas herramientas hacen que el trabajo se realice con una mayor rapidez y eficiencia. Para el análisis y fundamentación de este trabajo contamos con herramientas de modelado y de desarrollo.

2.7.1 Herramientas de modelado

Para el desarrollo del sistema es muy importante hacer uso de herramientas de modelado de objetos, ya que mediante ellas se realiza todo el análisis del sistema, por esta razón se hace uso de las herramientas de modelado el Rational Rose Enterprise Edition y el ERwin® Data Modeler.

2.7.1.1 Rational Rose Enterprise Edition

Es la herramienta Case desarrollada por los creadores de UML que cubren todo el ciclo de vida de un proyecto: concepción y formalización del modelo, construcción de los componentes y certificación de las distintas fases. Nos permite una trazabilidad real entre modelo (análisis y diseño) y el código ejecutable.

Rational Rose domina el mercado de herramientas para el análisis, modelado, diseño y construcción orientada a objetos, tiene todas las características que los desarrolladores, analistas, y arquitectos exigen, soporte UML incomparable, desarrollo basado en componentes con soporte para arquitecturas líderes en la industria y modelos de componentes, facilidad de uso e integración optimizada.

La corporación *Rational* ofrece el Proceso Unificado de *Rational* (RUP), que unifica las mejores prácticas de muchas disciplinas en un consistente y completo proceso del ciclo de vida, que permite al equipo de desarrollo disminuir los tiempos de liberación, además de hacer más predecible el software que ellos producen. Este proceso está basado en el Lenguaje Unificado de Modelación **UML** y únicamente integrado a herramientas líderes en el desarrollo de software de *Rational*, el Proceso Unificado de *Rational* apoya el equipo completo de desarrollo de software con guías detalladas e información crítica aplicable a la mayoría de las aplicaciones de la industria.

2.7.1.2 ERwin® Data Modeler

Es una herramienta que permite la visualización de elementos claves y optimiza el diseño de las bases de datos, esta herramienta permite diseñar rápida y fácilmente las aplicaciones de bases de datos, garantizando un alto rendimiento e integridad de las mismas. Ayuda además a generar y mantener una alta calidad desde un modelo lógico de los requerimientos de la información. Por lo que se recomienda en el proceso de creación de base de datos, utilizar modelos para ayudarse a visualizar y comunicar las reglas de su negocio. El producto clarifica los problemas complejos de diseño de datos, documenta las

bases de datos, y ayuda a la empresa a comprender mejor sus datos. El mismo le e ayudara a incorporar la calidad a sus diseños y a las bases de datos que genera, ya que refuerza automáticamente los principios fundamentales del diseño en la medida en que usted modela. Además de que puede ayudarle a tomar el mando de su proyecto.

2.7.2 Herramientas de desarrollo

Para el desarrollo de la aplicación es necesario utilizar una serie de herramientas que posibiliten la implementación de la misma así como otras para la gestión de los datos de la BD. En este caso se va a hacer uso del Microsoft Visual Studio 2005 como herramienta de programación y SQL Server 2005 Express como gestor de BD por sus facilidades y potencial, pero como principal requisito esta la decisión por parte del cliente para el uso de las mismas.

2.7.2.1 Microsoft Visual Studio 2005

Visual Studio 2005 es una herramienta diseñada para superar los retos en el desarrollo de software y proporciona una variedad de herramientas que ofrece amplios beneficios tanto para desarrolladores individuales como para equipos de desarrollo. (*Visual Studio 2005 2005*)

Entre las características más distintivas del Visual Studio 2005 podemos encontrar las siguientes:

- Herramienta muy potente.
- Crea aplicaciones dinámicas para Windows, la Web y otras.
- Desarrolla aplicaciones cliente/servidor.
- Desarrolla y depura aplicaciones multicapas del servidor.
- Tiene comunicación más eficaz en los equipos de software.
- Contiene una ayuda con refactorización.
- Garantiza calidad y rapidez en todo el desarrollo.

2.7.2.2 SQL Server 2005 Express

SQL Server 2005 Express es una versión gratuita de SQL Server 2005. Contiene muchas de las principales ventajas de SQL Server 2005 frente a su predecesor, SQL Server 2000, entre las que se incluye SQL Server Reporting Services (SSRS). Esta herramienta es fácil de utilizar basada en tecnología

SQL Server 2005. Está diseñada para construir aplicaciones robustas y fiables ofreciendo una sencilla pero potente base de datos que es además gratuita. Además, presenta la habilidad de instalar aplicaciones de bases de datos al moverlas como archivos Windows típicos (*Microsoft SQL Server 2005 Express 2005*).

Esta herramienta facilita nuevas funcionalidades como:

- Una base de datos robusta para crear aplicaciones dinámicas
- Fuerte soporte de XML
- Herramientas y características para mejorar la gestión y facilidad de uso
- Instalación y setup sencillos
- Escalabilidad y rendimientos
- Gran integración con Visual Studio 2005
- Robusta seguridad

2.8 Lenguajes de programación

Aunque ciertamente ASP no es un lenguaje de programación, sino una arquitectura de desarrollo Web en la que se pueden usar por debajo distintos lenguajes (por ejemplo VB.NET o C# para ASP.NET, o VB Script/JScript para ASP). Para el desarrollo de esta aplicación se utilizará el C# porque es el nuevo lenguaje de propósito general orientado a objetos creado por Microsoft para su nueva plataforma .NET. Aunque es posible escribir código para la plataforma .NET en muchos otros lenguajes, C# es el único que ha sido diseñado específicamente para ser utilizado en ella, pero como principal requisito por el que se toma en cuenta el C#, es por petición del cliente.

2.8.1 C#

Los lenguajes C y C++ son muy unos de los lenguajes más utilizados en el campo de la informática y la programación de sistemas ya que ofrecen mecanismos de bajo nivel que permiten a los programadores hacer uso de las características más avanzadas de las plataformas sobre la que trabajan, además proporcionan un nivel preciso para construir una aplicación compleja, por otro lado el lenguaje java está equilibrado con la inherente ineficiencia y la portabilidad. Dado esto Microsoft crea el lenguaje C#

combinando algunas de las características más avanzadas de java con algunas de las más potentes de C y C++ y construido especialmente para adaptarse de manera natural al *framework*.

Las principales características que identifican al lenguaje C# son:

Sencillez: Elimina muchos elementos que otros lenguajes incluyen y que son innecesarios en .NET.

Modernidad: Incorpora en el propio lenguaje elementos que son muy útiles para el desarrollo de aplicaciones y que en otros lenguajes hay que simular.

Orientación a objetos: Es más puro pues no admiten funciones ni variables globales, todo el código y datos han de especificarse dentro de definiciones de tipos de datos, lo que reduce problemas por conflictos de nombres y facilita la legibilidad del código.

Orientación a componentes: Su sintaxis incluye elementos propios del diseño de componentes que otros lenguajes tienen que simular mediante construcciones más o menos complejas.

Seguridad de tipos: Incluye mecanismos que permiten asegurar que los accesos a tipos de datos siempre se realicen correctamente.

Eficiencia: El código incluye numerosas restricciones para alcanzar su seguridad y no permite el uso de punteros. A diferencia de Java, en el mismo es posible saltarse dichas restricciones manipulando objetos a través de punteros.

2.8.2 SQL (Structure Query Language)

El lenguaje de consulta estructurado (**SQL**) es un lenguaje de base de datos normalizado, utilizado por el motor de base de datos de Microsoft Jet.

SQL se utiliza para crear objetos QueryDef, como el argumento de origen del método OpenRecordSet y como la propiedad RecordSource del control de datos. También se puede utilizar con el método Execute para crear y manipular directamente las bases de datos Jet y crear consultas **SQL** de paso a través para manipular bases de datos remotas cliente - servidor.

Existen dos tipos de comandos **SQL**:

- Los DDL que permiten crear y definir nuevas bases de datos, campos e índices.

- Los DML que permiten generar consultas para ordenar, filtrar y extraer datos de la base de datos.

Características del lenguaje SQL.

El SQL es un lenguaje de acceso a bases de datos que explota la flexibilidad y potencia de los sistemas relacionales permitiendo gran variedad de operaciones sobre los mismos. Es un lenguaje declarativo de **alto nivel** o **de no procedimiento**, que gracias a su fuerte base teórica y su orientación al manejo de conjuntos de registros, y no a registros individuales, permite una alta productividad en codificación. De esta forma una sola sentencia puede equivaler a uno o más programas que utilizaran un lenguaje de bajo nivel orientado a registro.

2.9 Conclusiones parciales

Después de realizarse un análisis profundo de todas las tecnologías y tendencias, como se ve a lo largo de todo el capítulo se puede concluir diciendo que SQL Server 2005 es el más adecuado para nosotros como desarrolladores teniendo en cuenta el sistema a desarrollar en este caso, ya que brinda muchas posibilidades por sus características, como metodología a utilizar **RUP** y el lenguaje de programación Visual C# y UML como lenguaje de modelado de objetos.

Capítulo

3

Presentación de la solución propuesta

3.1 Introducción

En este capítulo se dará una idea clara del negocio de este trabajo sobre los activos fijos en las empresas para lograr una mejor comprensión del problema, en general se hará una caracterización del sistema. Además se enumeran los requisitos funcionales y no funcionales que debe tener el sistema que proponemos, e identificar mediante un diagrama de casos de uso las relaciones de los actores que interactúan con el sistema y las secuencias de acciones con las que interactúan.

3.2 Objeto de automatización

Se quiere automatizar el módulo de activos fijos del proyecto **ConDor** ya existente en Cuba como una aplicación de escritorio. Por ello lo que se desea hacer es pasar dicho sistema a una aplicación Web con BD única, de esta manera se lograría un sistema multiusuario en el cual podrían trabajar varios usuarios al mismo tiempo y con la misma BD, logrando esto se evitaría también la utilización de ordenadores muy avanzados.

3.3 Información que se maneja

El sistema brinda la posibilidad de contar con una información detallada, cada una con varias características y que requieren de una alta seguridad. El módulo "**Control de Activos Fijos**", registra los procedimientos establecidos para el control interno y la contabilización de los Activos Fijos Tangibles e Intangibles. De esta forma el centro mantiene una correcta actualización y control de dicho activo, así como su estado y ubicación dentro del mismo. Se mantiene información referente a los proveedores de los productos que el centro solicita. Permite a su vez mantener un control sobre los activos que se reciben en el centro para su comercialización, realizando inventarios eventuales por cada tipo de activo tomando como objetivo el movimiento de los mismos dentro del centro y registrando las salidas del almacén hacia

las distintas áreas dentro del centro. Cheque la depreciación de los activos recibidos para así analizar su valor en uso.

El acceso a los datos que el sistema controla, es verificado para cada acción ejecutada por el actor, chequeando un correcto funcionamiento de la aplicación y una incorruptibilidad en el tráfico de la información, con el objetivo de garantizar la integridad de los datos. El volumen de la información que se registra esta en correspondencia directa con la cantidad de activos suministrados a la empresa.

3.4 Propuesta de sistema

La aplicación Web con BD única propuesta como sistema permite registrar y controlar la entrada y salida de los activos fijos de una empresa, así como sus movimientos y operaciones dentro de la misma, mantiene una correcta actualización de la disponibilidad de cada uno de ellos para así agilizar la asignación del mismo basándose en la previa solicitud de estos, efectuadas por las diferentes áreas. Permite una rápida atención a dichas solicitudes.

3.5 Modelo de negocio

Para este sistema de ha estudiado todo el negocio referente a toda la gestión contable de las empresas, pero la contabilidad está compuesta por varios módulos dentro de los cuales se encuentra el módulo de **Activos fijos**, que es en nuestro caso a los que hacemos referencia. En esta actividad se recepcionan todos los **activos fijos** que son ingresados a las empresas por varias vía, tanto compra como alquiler, así como se controla el movimiento de estos, además se vela por el buen uso y reparación de los mismos, los **activos fijos** pueden ser tangibles e intangibles. Estos activos se recepcionan en el almacén para después ser distribuidos por todos los departamentos de la empresa en cuestión, en dependencia de la demanda de los mismos.

3.5.1 Actores del negocio

Debe quedar claro quién o quiénes interactúan con el negocio para poder entender bien el problema, es decir quien envía o recibe algo en la empresa.

El término actor significa el papel que desempeña alguien o algo al interactuar con el negocio(KENDALL 2005). Los actores del negocio son:

Por lo general un actor responde a un usuario físico, pero en ocasiones un actor puede ser un sistema de información.

Nombre del actor	Descripción
Proveedor	Es el encargado de hacer llegar la mercancía a las empresas con la factura.
Director de la Entidad.	Es aquel que aprueba y firma todo movimiento y compra de activos fijos en la empresa.
Informante de alta.	Se encarga de informar el alta de cualquier activo fijo .
J' de Áreas.	Se encargan de hacer las solicitudes de movimiento de activos fijos entre áreas así como el pedido de estos para su asignación.
Organismo superior.	Es el encargado para aprobar y firmar los documentos de baja activos fijos .
Reloj.	Indica cuando realizar el chequeo mensual de los activos fijos . Es decir inicia actividades que se realizan con cierta periodicidad.
Taller.	Es el taller que decepciona los activos fijos que necesitan reparación para realizar dicha reparación.

Tabla 1. Descripción de los actores del negocio

3.5.2 Trabajadores del Negocio

En el caso del control de los **activos fijos** existen varios trabajadores del negocio ya que ejecutan dentro del mismo diferentes procesos y se mostraran continuación:

Nombre del trabajador	Descripción
Contador General de AFT	Es el encargado de ejecutar todos los procesos definidos en el negocio actual.
Director de Contabilidad y finanzas	Es el encargado de revisar y firmar todos documentos que llegan a sus manos.
Expertos (Especialista General)	Son los encargados de revisar, evaluar y aprobar todos documentos que llegan a sus manos.
Jefe del Almacén	Es el encargado de aprobar la entrada al almacén de los activos fijos que lleguen a la entidad.
Contabilidad	Es el Dpto. responsable de gestionar todos los procesos referentes a los Activos fijos.
Especialista superior de Inventario.	Es el encargado de revisar y aprobar los documentos relacionados con el Plan Mensual de Chequeos.

Tabla 2. Descripción de los trabajadores del negocio.

3.5.3 Diagrama de casos de Uso del Negocio

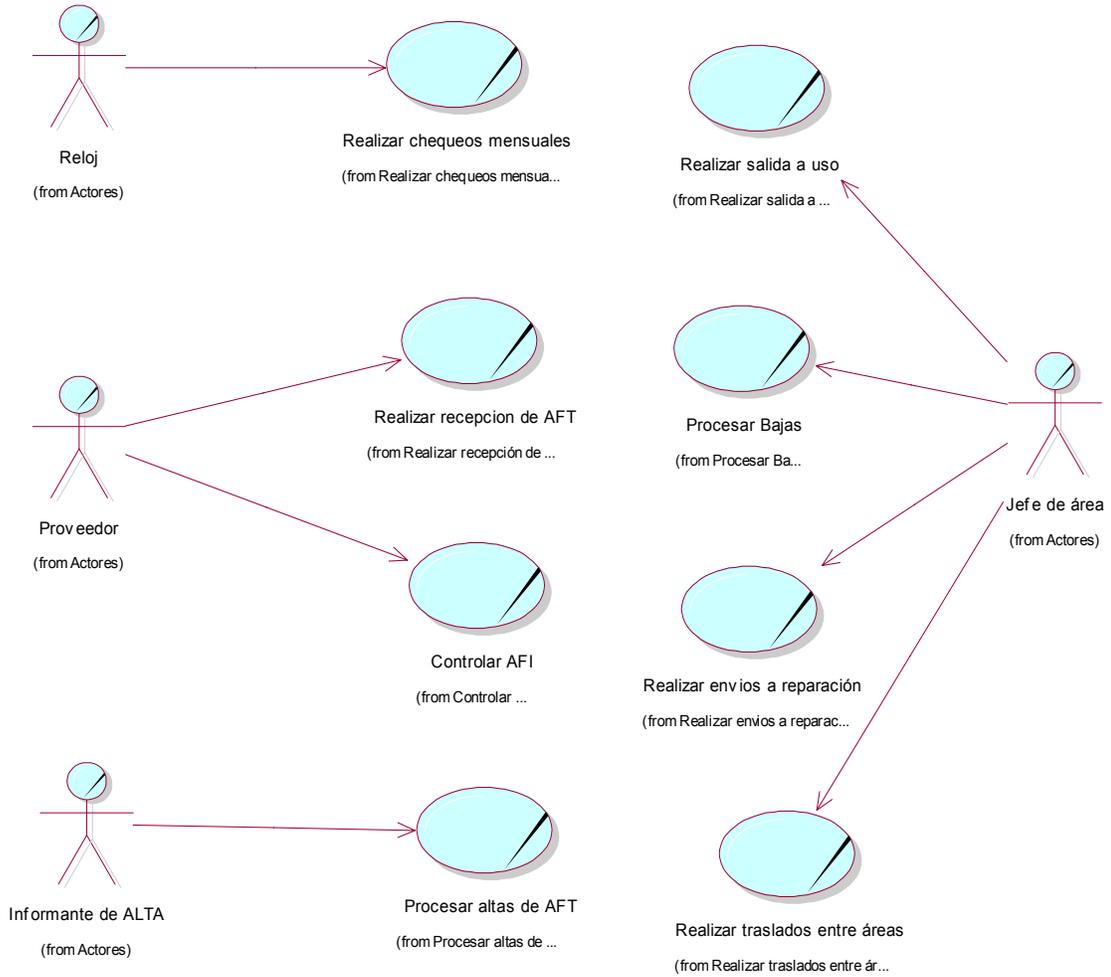


Figura 2. Diagrama de casos de uso del negocio.

3.5.4 Descripción de casos de uso del negocio

✓ Realizar recepción de AFT

Caso de Uso:	Realizar recepción de AFT
Actores:	Proveedor. (Inicia).
Trabajadores:	Jefe de Almacén, Contador General de AFT.
Resumen:	<p>El caso de uso inicia cuando el proveedor llega a la empresa con el cargamento. El contador general de activos fijos recibe la factura del cargamento que acaba de llegar, a su vez el cargamento es enviado al almacén y el jefe del mismo realiza una recepción sin la factura, que anteriormente ha sido entregada al Dpto. de Contabilidad. Luego de que el Dpto. de Contabilidad como el Almacén hayan hecho la conciliación del cargamento recibido, el Jefe de almacén se encarga de enviar el Informe de IR al Contador General que se encargará en caso de que no existan errores, de archivar la información en el Expediente de Informe de Recepción (IR), terminando así el CU.</p> <p>En caso contrario, es decir de encontrarse errores se realiza una reclamación, terminando así el CU.</p>
Precondiciones:	
Poscondiciones	
Mejoras	Informatizar: <ul style="list-style-type: none"> • Informe de IR • Expediente de IR • IR según factura
Prioridad	Alta.

✓ *Procesar altas de AFT*

Caso de Uso:	Procesar altas de AFT
Actores:	Informante de alta. (Inicia).
Trabajadores:	Contador General de AFT, Expertos
Resumen:	El caso de uso inicia cuando el informante de alta envía la factura del medio y el modelo SC 2-04 al contador general de activos fijos, este después analiza dichos documentos y realiza el informe de alta. Luego genera el modelo SC 1-01 y saca copias del mismo y de la factura, archiva el original de cada documento y a su vez envía las copias a los expertos. Después el propio contador general de activos fijos pide a los expertos que le faciliten el estado de la cuenta de control para conciliarlo con el modelo SC 1-01 y emitir el estado financiero, terminando así el CU.
Precondiciones:	
Poscondiciones	
Mejoras	Informatizar: <ul style="list-style-type: none"> • Informe de alta del medio • Modelo SC 1-01 • Carpeta para altas • Cuenta control
Prioridad	Alta.

 ✓ *Realizar Salidas a uso.*

Caso de Uso:	Realizar Salidas a uso.
Actores:	Jefe de Áreas. (Inicia).
Trabajadores:	Contador General de AFT, Director de la Entidad, Jefe del almacén.
Resumen:	El caso de uso inicia cuando el jefe de área envía la solicitud del movimiento al contador general de activos fijos, este analiza los datos de la solicitud y si están correctos procede a confeccionar el modelo SC 1-01 y a sacar copias del

	mismo, luego archiva la tercera copia y envía al director de la entidad las otras dos copias para que sea aprobado, al ser aprobado el modelo SC 1-01, el contador general de activos fijos realiza copias del modelo SC 1-01 aprobado por el director de la entidad y envía una copia al jefe de área solicitante y otra al jefe del almacén para proceder a poner en uso el medio, de allí contabiliza el medio y realiza el acta de responsabilidad material y realiza los ajusten necesarios en caso de que exista diferencia de precio, terminando así el CU.
Precondiciones:	
Poscondiciones	
Mejoras	Informatizar: <ul style="list-style-type: none"> • Modelo SC 1-01 • Carpeta de AFT pendientes a autorizar • Acta de responsabilidad material • Carpeta de movimientos realizados
Prioridad	Alta.

✓ Realizar traslados entre áreas

Caso de Uso:	Realizar traslados entre áreas
Actores:	Jefe de área. (Inicia), Director de la entidad
Trabajadores:	Contador General de AFT.
Resumen:	El caso de uso inicia una vez que el jefe de área envía una solicitud de movimiento de AFT al contador general de AFT éste confecciona el modelo SC 1-01 en original y dos copias y se los envía al director de la entidad quien es el encargado de firmar los modelos para aprobar la solicitud, una vez firmados son devueltos al contador general de AFT el cual al tener estos documentos aprobados envía las copias aprobadas a los jefes de área, confecciona el modelo SC 1-07 y archiva la información del traslado en una carpeta habilitada para este fin, terminando así el CU.

recondiciones:	
Poscondiciones	
Mejoras	Informatizar: <ul style="list-style-type: none"> • Modelo SC 1-01 • Carpeta de traslados
Prioridad	Alta.

✓ *Procesar Bajas.*

Caso de Uso:	Procesar Bajas
Actores:	Jefe de área. (Inicia), Organismo superior.
Trabajadores:	Contador General de AFT, Comisión para aprobar bajas
Resumen:	<p>El caso de uso inicia cuando el jefe de área informa y envía el informe técnico del activo en proceso de baja al contador general de activos fijos que existe un activo al cual se le debe dar baja de allí el propio contador general elabora un certificado técnico de baja y el modelo SC 1-01, después anexa ambos documentos en el expediente de baja y de allí saca copias del mismo y del modelo SC 1-01. Posteriormente envía la copia del expediente de baja a la comisión que aprueba las bajas, archiva una copia del modelo SC 1-01 en la carpeta pendientes a autorizar y envía original y otra copia del modelo SC 1-01 al organismo superior para que este lo apruebe y firme. Luego elabora el modelo SC 5-05 y el comprobante de operaciones y de allí anexa ambos documentos al proceso de baja y envía el modelo SC 1-01 al jefe de área, terminando así el CU.</p>
Precondiciones:	
Poscondiciones	
Mejoras	Informatizar: <ul style="list-style-type: none"> • Modelo SC 1-01 • Certificado técnico de baja • Expediente de baja

	<ul style="list-style-type: none">• Carpeta pendientes a autorizar• Acta de aprobación de baja• Comprobante de operaciones• Modelo SC 1-07
Prioridad	Alta.

3.5.5 Modelo de objeto

El diagrama de clases del Modelo de Objeto, es un artefacto que se construye para describir el modelo de objetos del negocio, muestra la participación de los trabajadores y entidades del negocio y la relación entre ellos. Para ver los modelos de objetos, consultar el **Anexo 1**.

3.6 Especificación de los requerimientos del software

“El propósito fundamental del flujo de trabajo de los requisitos es guiar el desarrollo hacia el sistema correcto. Esto se consigue mediante una descripción de los requisitos del sistema (es decir, las condiciones o capacidades que le sistema debe cumplir) suficientemente buena como para que pueda llegarse a un acuerdo entre el cliente (incluyendo los usuarios) y los desarrolladores sobre qué debe y que no debe hacer el sistema.(ZAVALA 2000)

3.6.1 Actores del sistema a automatizar: módulo Activo Fijo

Todo trabajador del negocio que interactúa con las actividades a automatizar es un posible actor del sistema. En caso de que exista algún actor del negocio que interactué con el sistema, entonces también será un actor del sistema.(ZAVALA 2000) Una vez analizados los actores y trabajadores del negocio en el módulo de activo fijo, se definieron los siguientes actores del sistema para este módulo:

Definición de actores del sistema a automatizar: Módulo de Activo Fijo

Nombre del actor	Descripción
Contador general de AF	Manipula todas las funcionalidades del módulo de Activo Fijo del Departamento de Contabilidad.
Técnico	Es un usuario genérico ya que permite la autenticación del: <ul style="list-style-type: none">➤ Contador general de AF.➤ Director de Contabilidad y Finanzas.
Director de Contabilidad y Finanzas	Además de ser el encargado de realizar las funciones como Director de Contabilidad y Finanzas, tiene funciones dentro del sistema que solo puede realizar su rol, como la revisión y aprobación de los documentos y Expedientes referentes a los Activos Fijos dentro del Dpto. de Contabilidad y Finanzas.
Informático de la Entidad	Es el encargado de realizar todas las operaciones de seguridad de la entidad, como el acceso de los usuarios y la configuración del sistema.

Tabla 3. Descripción de los actores del sistema.

3.6.2 Requisitos funcionales

Los requerimientos funcionales son capacidades o condiciones que el sistema debe cumplir. (CERIA 2003)

1. Autenticar usuario.

- 1.1. Usuario.
- 1.2. Contraseña.

2. Gestionar usuario.

- 2.1. Adicionar usuario.
- 2.2. Modificar usuario.
- 2.3. Eliminar usuario.
- 2.4. Buscar y visualizar reporte de usuarios.
 - 2.4.1. Imprimir reporte de usuarios.

3. Gestionar acceso de usuarios.

- 3.1. Adicionar registro de acceso de usuarios.
- 3.2. Modificar registro de acceso de usuarios.
 - 3.2.1. Registrar traza.
- 3.3. Eliminar registro de acceso de usuarios.
- 3.4. Buscar y visualizar registro de acceso de usuarios.
 - 3.4.1. Imprimir registros de acceso de usuarios.

4. Gestionar Tarjetas de Activos Fijos.

- 4.1. Insertar datos en Tarjetas de Activos Fijos.
- 4.2. Modificar datos de Tarjetas de Activos Fijos.
 - 4.2.1. Registrar traza.
- 4.3. Eliminar datos de Tarjetas de Activos Fijos.
- 4.4. Buscar y visualizar datos de Tarjetas de Activos Fijos.
 - 4.4.1. Imprimir Tarjetas de Activos Fijos.

5. Realizar cierre de apertura.

5.1. Confirmar el cierre de apertura.

6. Gestionar factura recibida.

6.1. Insertar datos en la factura.

6.2. Modificar datos en la factura.

6.2.1. Registrar traza.

6.3. Eliminar factura.

6.4. Buscar y visualizar factura.

6.4.1. Imprimir factura.

7. Gestionar modelo de movimiento.

7.1. Insertar datos en el modelo de movimiento.

7.2. Modificar datos en el modelo de movimiento.

7.2.1. Registrar traza.

7.3. Eliminar modelo de movimiento.

7.4. Buscar y visualizar modelo de movimiento.

7.4.1. Imprimir modelo de movimiento.

8. Gestionar submayor.

8.1. Insertar datos en el submayor.

8.2. Modificar datos en el submayor.

8.2.1. Registrar traza.

8.3. Eliminar submayor.

8.4. Buscar y visualizar submayor.

8.4.1. Imprimir en el submayor.

9. Gestionar Contabilización.

9.1. Realizar Contabilización

9.2. Modificar datos en contabilización.

9.2.1. Registrar traza.

9.3. Eliminar contabilización.

9.4. Buscar y visualizar contabilización.

9.4.1. Imprimir contabilización.

10. Gestionar expediente de depuración.

- 10.1. Insertar datos en el expediente de depuración.
- 10.2. Modificar datos en el expediente de depuración.
 - 10.2.1. Registrar traza.
- 10.3. Eliminar expediente de depuración.
- 10.4. Buscar y visualizar expediente de depuración.
 - 10.4.1. Imprimir expediente de depuración.

11. Gestionar contratos.

- 11.1. Insertar datos en el contrato.
- 11.2. Modificar datos en el contrato.
 - 11.2.1. Registrar traza.
- 11.3. Eliminar contrato.
- 11.4. Buscar y visualizar contrato.
 - 11.4.1. Imprimir contrato.

12. Gestionar facturas emitidas.

- 12.1. Insertar datos en las facturas por ventas.
- 12.2. Modificar datos en las facturas por ventas.
 - 12.2.1. Registrar traza.
- 12.3. Eliminar facturas por ventas.
- 12.4. Buscar y visualizar facturas por ventas.
 - 12.4.1. Imprimir facturas por ventas.

13. Realizar cierre de mes.

- 13.1. Generar cierre de mes.

14. Realizar comprobación de saldos.

- 14.1. Generar comprobación de saldos.
 - 14.1.1. Visualizar reporte de comprobación de saldos.
 - 14.1.2. Imprime.

15. Realizar cierre de fin de año e inicio del próximo.

- 15.1. Generar cierre de fin de año e inicio del próximo.

16. Gestionar reportes generales.

- 16.1. Crear reportes generales.

16.2. Visualizar reportes generales.

16.2.1. Imprimir reportes generales.

17. Gestionar centros de costo.

17.1. Insertar centros de costo.

17.2. Modificar centros de costo.

17.2.1. Registrar traza.

17.3. Eliminar centros de costo.

17.4. Buscar y visualizar centros de costo.

17.4.1. Imprimir centros de costo.

18. Gestionar Áreas.

18.1. Insertar áreas.

18.2. Modificar áreas.

18.2.1. Registrar traza.

18.3. Eliminar áreas.

18.4. Buscar y visualizar áreas.

18.4.1. Imprimir áreas.

19. Gestionar cuentas de movimientos.

19.1. Insertar cuentas de movimientos.

19.2. Modificar cuentas de movimientos.

19.2.1. Registrar traza.

19.3. Eliminar cuentas de movimientos.

19.4. Buscar y visualizar cuentas de movimientos.

19.4.1. Imprimir cuentas de movimientos.

20. Gestionar cuentas de valor y depreciación o grupos.

20.1. Insertar cuentas de valor y depreciación o grupos.

20.2. Modificar cuentas de valor y depreciación o grupos.

20.2.1. Registrar traza.

20.3. Eliminar cuentas de valor y depreciación o grupos.

20.4. Buscar y visualizar cuentas de valor y depreciación o grupos.

20.4.1. Imprimir cuentas de valor y depreciación o grupos.

21. Gestionar subelemento de gasto.

- 21.1. Insertar subelemento de gasto.
- 21.2. Modificar subelemento de gasto.
 - 21.2.1. Registrar traza.
- 21.3. Eliminar subelemento de gasto.
- 21.4. Buscar y visualizar subelemento de gasto.
 - 21.4.1. Imprimir subelemento de gasto.

22. Gestionar subcuenta de activo.

- 22.1. Insertar subcuenta de activo.
- 22.2. Modificar subcuenta de activo.
 - 22.2.1. Registrar traza.
- 22.3. Eliminar subcuenta de activo.
- 22.4. Buscar y visualizar subcuenta de activo.
 - 22.4.1. Imprimir subcuenta de activo.

23. Gestionar subcuenta de depreciación.

- 23.1. Insertar subcuenta de depreciación.
- 23.2. Modificar subcuenta de depreciación.
 - 23.2.1. Registrar traza.
- 23.3. Eliminar subcuenta de depreciación.
- 23.4. Buscar y visualizar subcuenta de depreciación.
 - 23.4.1. Imprimir subcuenta de depreciación.

24. Gestionar subcuenta corriente.

- 24.1. Insertar subcuenta corriente.
- 24.2. Modificar subcuenta corriente.
 - 24.2.1. Registrar traza.
- 24.3. Eliminar subcuenta corriente.
- 24.4. Buscar y visualizar subcuenta corriente.
 - 24.4.1. Imprimir subcuenta corriente.

25. Gestionar situación del activo.

- 25.1. Insertar situación del activo.

25.2. Modificar situación del activo.

25.2.1. Registrar traza.

25.3. Eliminar situación del activo.

25.4. Buscar y visualizar situación del activo.

25.4.1. Imprimir situación del activo.

26. Gestionar componente de obra.

26.1. Insertar componente de obra.

26.2. Modificar componente de obra.

26.2.1. Registrar traza.

26.3. Eliminar componente de obra.

26.4. Buscar y visualizar componente de obra.

26.4.1. Imprimir componente de obra.

27. Administrar plantilla de contabilización.

27.1. Modificar datos de la plantilla de contabilización.

27.1.1. Registrar traza.

27.2. Visualizar plantilla de contabilización.

27.2.1. Imprimir plantilla de contabilización.

28. Gestionar registros de operaciones.

28.1. Generar registros de operaciones.

28.2. Visualizar registros de operaciones.

28.2.1. Imprimir registros de operaciones

29. Gestionar comprobantes de operaciones.

29.1. Generar comprobantes de operaciones.

29.2. Visualizar comprobantes de operaciones.

29.2.1. Imprimir comprobantes de operaciones.

3.6.4 Diagrama de jerarquía de actores

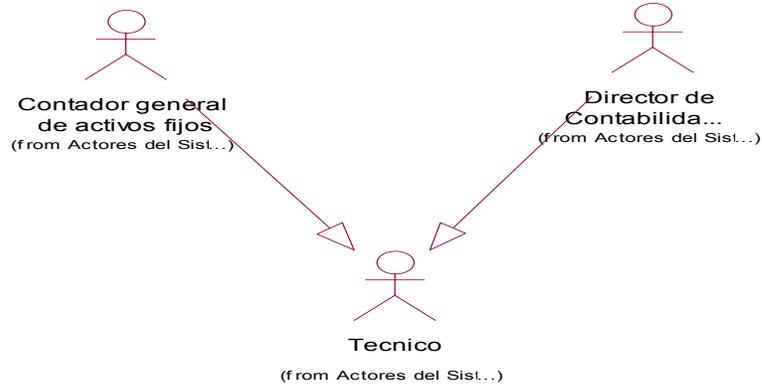


Figura 3. Diagrama de jerarquía de actores

3.6.5 Diagrama de relación entre paquetes del sistema a automatizar: Módulo de Activo Fijo

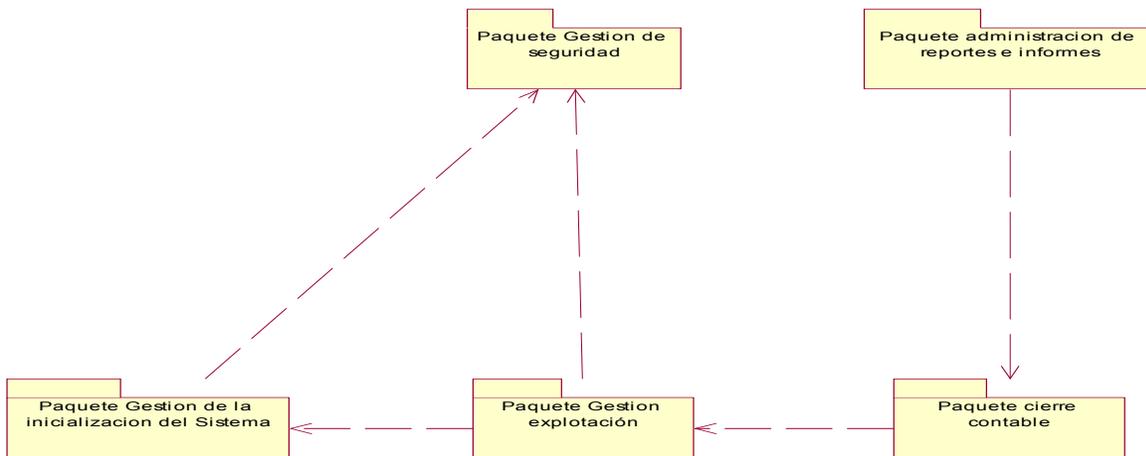


Figura 4. Diagrama de relación entre paquetes

3.6.5 Diagrama de casos de uso del sistema a automatizar: Módulo de Activo Fijo

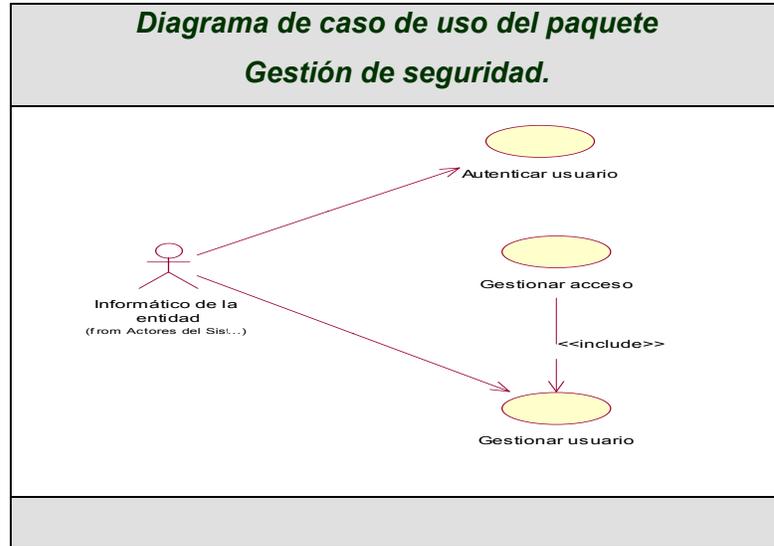


Figura 5. Diagrama de casos de uso del sistema a automatizar.

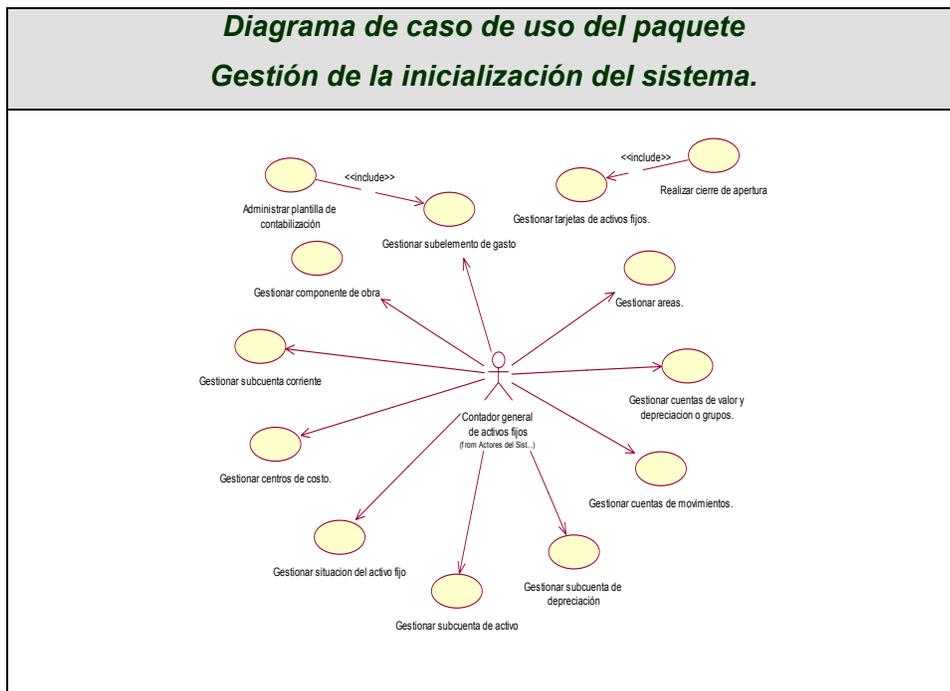


Figura 6. Diagrama de casos de uso del paquete Gestión de la inicialización del sistema.

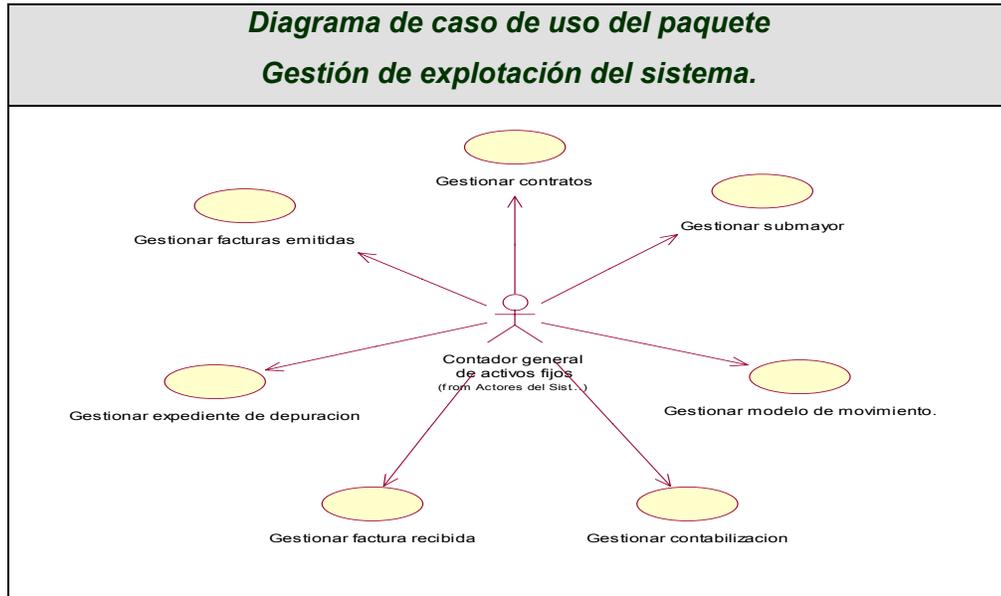


Figura 7. Diagrama de caso de uso del paquete Gestión de la explotación del sistema

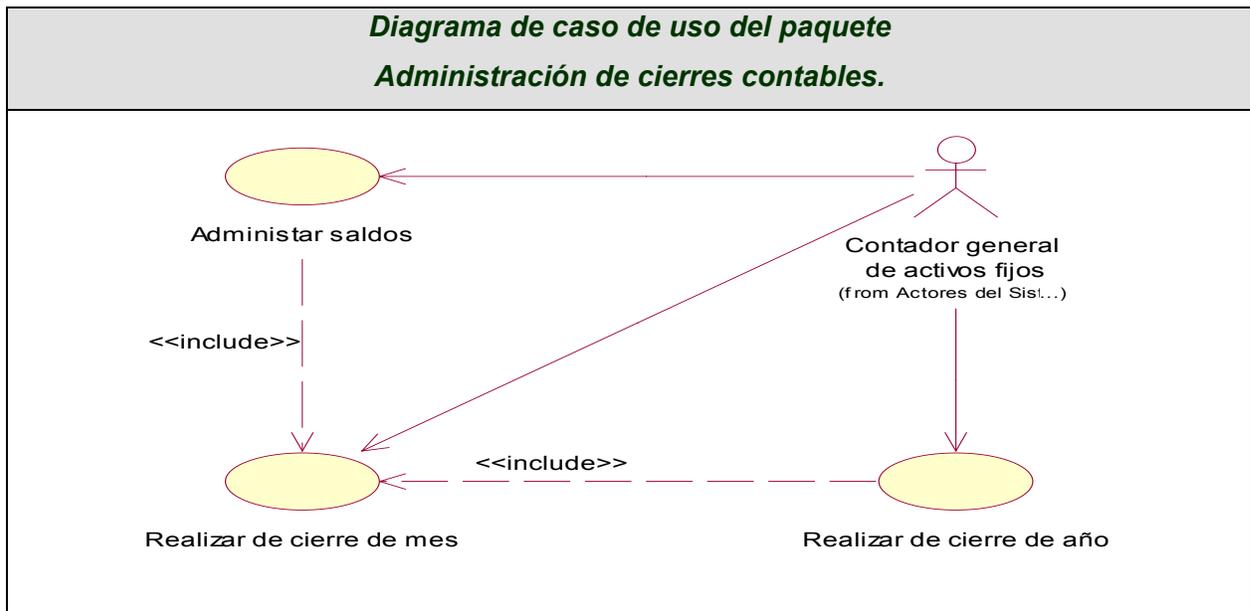


Figura 8. Diagrama de caso de uso del paquete Administración de cierre contable.

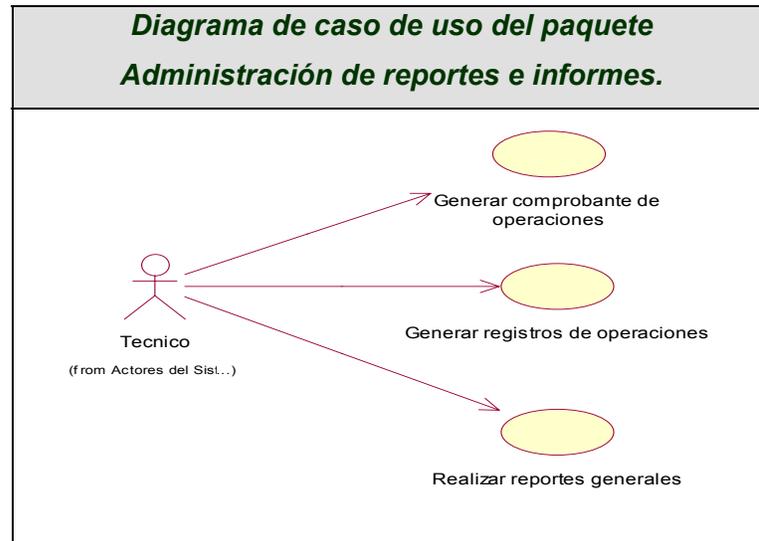


Figura 9. Diagrama de caso de uso del paquete Administración de reportes e informes.

3.6.6 Descripción de los casos de uso del sistema a automatizar: Módulo de Activo Fijo

Para ver las descripciones de los casos de uso del sistema, consultar el **Anexo 2**

3.7 Definición de los requisitos no funcionales

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

✓ **Apariencia e interfaz de usuario.**

La aplicación Web que se propone para la gestión de la contabilización de activos fijos debe ser sencilla, profesional y legible. En el caso del módulo **activos fijos** el color es el verde por petición del usuario.

✓ **Usabilidad.**

La aplicación podrá ser usada por cualquier persona que posea conocimientos básicos de computación y de contabilidad. Se debe garantizar un fácil y rápido acceso a los usuarios.

✓ **Soporte.**

La aplicación debe ser integrada ya que es un subsistema, de manera que se mantenga la colaboración entre todos los subsistemas que componen el sistema general. La base de datos que utilizará el sistema

como medio de almacenamiento de la información estará soportada sobre un gestor de bases de datos SQL Server Express, permitiéndole interactuar con otros subsistemas estableciendo vías de compatibilidad.

✓ **Portabilidad.**

El sistema podrá ser usado sobre el sistema operativo Windows

✓ **Seguridad.**

La aplicación debe contar con protección contra acciones no autorizadas o que puedan afectar la integridad de los datos. Además debe garantizar que la información solo pueda ser consultada por los usuarios con un nivel de acceso autorizado y estará prohibida su divulgación.

Solo se podrá acceder a la aplicación desde una computadora personal fuera de la empresa, en caso de que esta no tenga los recursos de hardware necesarios para montar el servidor en su misma empresa y necesite de los servicios del **SICS** para lograr trabajar con la aplicación **Condor**.

Debe contar con varios niveles de acceso para permitir el trabajo organizado con la aplicación.

✓ **Confiabilidad.**

La aplicación debe ser capaz de recuperarse de cualquier error en un tiempo mínimo. La aplicación será usada solamente por los trabajadores del departamento de contabilidad y finanzas, por lo que la información que fluya en el mismo será la oficial. La aplicación debe ser precisa en la información que le suministra al usuario para evitar cualquier tipo de error. Además la información manejada por el sistema será protegida contra corrupción y estados inconsistentes.

✓ **Instalación.**

Para la instalación se debe de crear dos instaladores, uno para todas páginas y otro para la BD.

✓ **Software en el cliente.**

Se deberá disponer de un Sistema Operativo Windows 98 o superior para su instalación, estas computadoras accederán al sistema usando un navegador compatible con Internet Explorer 5.5, Mozilla 1.7 o superior.

✓ **Software en el servidor.**

Se deberá disponer de un Sistema Operativo Windows Server 2003.

✓ **Hardware en el servidor.**

Un procesador de 3.0 GHz, memoria RAM de 2.0GB o superior, disco duro de 80 GB y 7200 RPM.

✓ **Hardware en el cliente.**

Se deberá contar con impresora y escáner en las computadoras personales clientes que interactúen con la aplicación y vayan a imprimir algún documento. Microprocesador superior a 1.0 GHz, 128 MB mínimo de RAM.

3.8 Conclusiones parciales

Se identificaron los actores y trabajadores del negocio, también se elaboró el diagrama de casos de usos del negocio, así como la descripción de los mismos. Se identificaron los actores y casos de usos del sistema, se obtuvieron los requisitos funcionales y no funcionales del sistema.

Se elaboró el diagrama de casos de uso del sistema, mostrando la relación entre éstos y los actores del sistema. Se describieron detalladamente los casos de uso del sistema, donde se reflejaron las funcionalidades recogidas en los requerimientos, proporcionando un fácil diseño de las clases y la elaboración de los diagramas correspondientes.

El desarrollo de todas estas actividades permitirá diseñar las clases con vista a la implementación.

Capítulo

4

Construcción de la solución propuesta

4.1 Introducción

En este capítulo se modelan los artefactos que conforman las clases de cada paquete así como las relaciones entre los mismos, además se muestra para un mejor entendimiento del manejo de los datos el modelo de datos donde aparece la información que en su momento será almacenada.

Se exponen los principios del diseño llevados a cabo durante el desarrollo de la capa de presentación del sistema, con detalles de la elaboración de la interfaz, de los estándares a seguir para la confección de los reportes generados por el sistema y de la estructura de los modelos utilizados en la contabilidad empresarial.

Como parte esencial para el funcionamiento del sistema se propone el diseño de la base de datos utilizada por el modulo de activos fijos del sistema, así como también se muestra el modelo de implementación donde se muestran los nodos y los componentes que forman la estructura de la aplicación.

4.2 Análisis

El análisis consiste en obtener una visión del sistema que se preocupa por ver ¿Qué hace?, de modo que sólo se interesa por los requisitos funcionales, transformar los requisitos funcionales en un diseño de clases viendo las relaciones e interacción que existe entre ellos, y teniendo en cuenta en el proceso una arquitectura robusta que permita adaptar el sistema al entorno de implementación que se está desarrollando.

4.3 Principios de diseño

Según Conallen UML cuenta con una extensión para modelar aplicaciones web, esta extensión es usada para el diseño de las clases. Los estereotipos que usa son:



<<Server Page>> Representa la parte de la pagina web que ejecuta el código en el servidor. Las operaciones representan las funciones del código y los atributos las variables dentro del alcance de la página. Esta clase sólo puede tener relaciones con objetos en el servidor.



<<Client Page>> Una instancia de Página Cliente es una página Web, con formato HTML. Son interpretadas por el browser. Cada página cliente solo puede ser construida por una página servidor.



<<Form>> Es el grupo de elementos de entrada que forman parte de una página cliente. Sus atributos son los elementos de entrada del formulario (Text Field, Text Area, Button, Label, Radio Button, Radio Group, Select, Check Box y Hidden Fields), y se relaciona con la etiqueta de igual nombre del HTML.

Arquitectura Web de tres Capas:

Es muy común en las aplicaciones Web usar una arquitectura en tres capas (Ver Fig. 1). La primera capa consiste en la capa de presentación que aparte del navegador incluye el servidor Web, este último se encarga de darle formato a los datos. La segunda es la capa de lógica del negocio que habitualmente está referido a algún tipo de programa, y la tercera capa es la de acceso a datos que aporta a la segunda capa datos para su ejecución.

El funcionamiento de cualquier aplicación web seria recoger datos del usuario, enviarlos al servidor que ejecuta el programa y el resultado será presentado al usuario en el navegador.

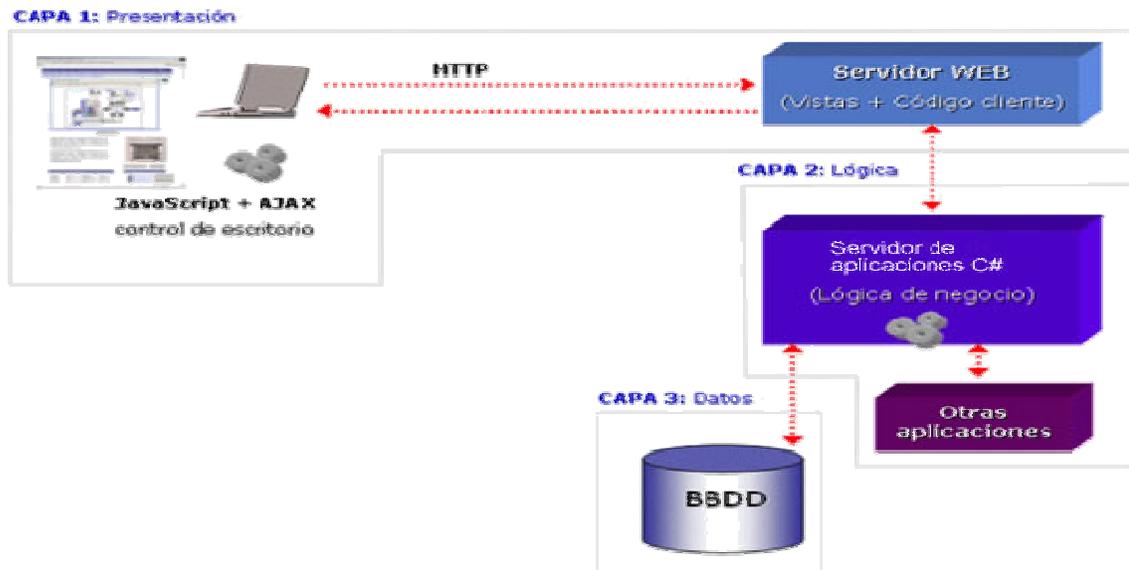


Figura 10. Arquitectura WEB de tres niveles.

Luego de analizado todo estos principios de la arquitectura, se propone el siguiente diseño para las clases de la aplicación propuesta.

4.3.1 Clases de la Capa Presentación

Esta capa incluye todas las clases cliente y sus formularios, así como un código JavaScript para la validación de los datos introducidos en el mismo, y además contiene un método imprimir para la impresión de los datos.



Figura 11. Capa de Presentación

4.3.2 Clases de la Capa Lógica del Negocio

Las clases que tienen diferentes procedimientos que dan respuesta la funcionalidad de la aplicación, estas clases se muestran en esta capa que contiene la lógica del negocio. Las operaciones que realizan

los modelos de movimientos, submayor, factura, etc., se pueden agrupar en una clase ya que son comunes. Observe que esta clase contiene un atributo objAcceso que es una instancia de la clase My_AccesoDato. Esta instancia accede a todas los procedimientos de la clase My_AccesoDato, la cual accederá a la Base de Datos.



Figura 12. Capa Lógica del Negocio

4.3.3 Clases de la Capa Acceso a Datos

Visual Studio incluye el uso de la capa de abstracción de acceso a datos ADO.NET, la misma sirvió como base para el diseño de una clase con la funcionalidad necesaria para el manejo de las clases entidad.

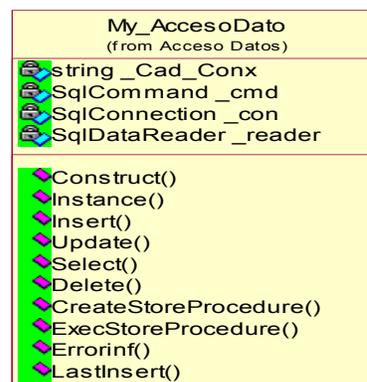


Figura 13. Capa de acceso a Datos.

4.4 Patrones

Los **Patrones** son la base para la búsqueda de soluciones a problemas comunes en el desarrollo de software y otros ámbitos referentes al diseño de interacción o interfaces. Es una solución a un problema

de diseño no trivial que es efectiva (ya se resolvió el problema satisfactoriamente en ocasiones anteriores) y reusable (se puede aplicar a diferentes problemas de diseño en distintas circunstancias).(LARMAN 2002)

4.4.1 Patrones del diseño

Para el diseño se utilizan los siguientes patrones del diseño:

Creador: Posibilita la creación de instancia de cada clase por sí misma.

Experto: Le asigna la responsabilidad a la clase que tiene mayor información sobre la tarea a realizar.

Controlador: Evita el acceso directo a las clases entidades, evita que la capa de presentación no maneje los eventos del sistema.

Alta cohesión: Reduce la dependencia, aumenta la complejidad pero es una complejidad manejable y eficiente. La responsabilidad se reparte en varias clases y existe una fuerte colaboración entre ellas.

4.5 Diseño

El **diseño** consiste en el refinamiento de los modelos de análisis para crear especificaciones adicionales que enriquecen el modelo de análisis con detalles próximos a la implementación. Una solución lógica, de forma que se cumplan los requerimientos (asignación de responsabilidades, interacciones entre objetos, etc.)(KENDALL)

4.5.1 Diagrama de clases del diseño

Para ver los diagramas de clases del diseño, consultar el **Anexo 3**

4.5.2 Diseño de la Base de Datos

Para que las bases de datos puedan almacenar información, reconocer datos y recuperar contenido necesitan de una definición de su estructura, dicha estructura debe ser desarrollada para la necesidad de las aplicaciones que la usarán.

Para el diseño de la base de datos se realizó el diagrama de clases persistentes y el modelo de datos generado por la herramienta Racional Rose.

Una clase persistente (persistent) es una clase entidad que tiene la capacidad de mantener su valor en el espacio y en el tiempo. En el diagrama de clases persistentes se muestran dichas clases y las relaciones entre ellas (asociación, agregación/composición y generalización/especialización).

En el diagrama del Modelo físico de datos, se muestran las tablas (sin atributos) y sus relaciones. Para ver las tablas con sus atributos consultar el **Anexo. Tablas del Modelo de Datos**.

4.5.2.1 Modelo lógico de datos

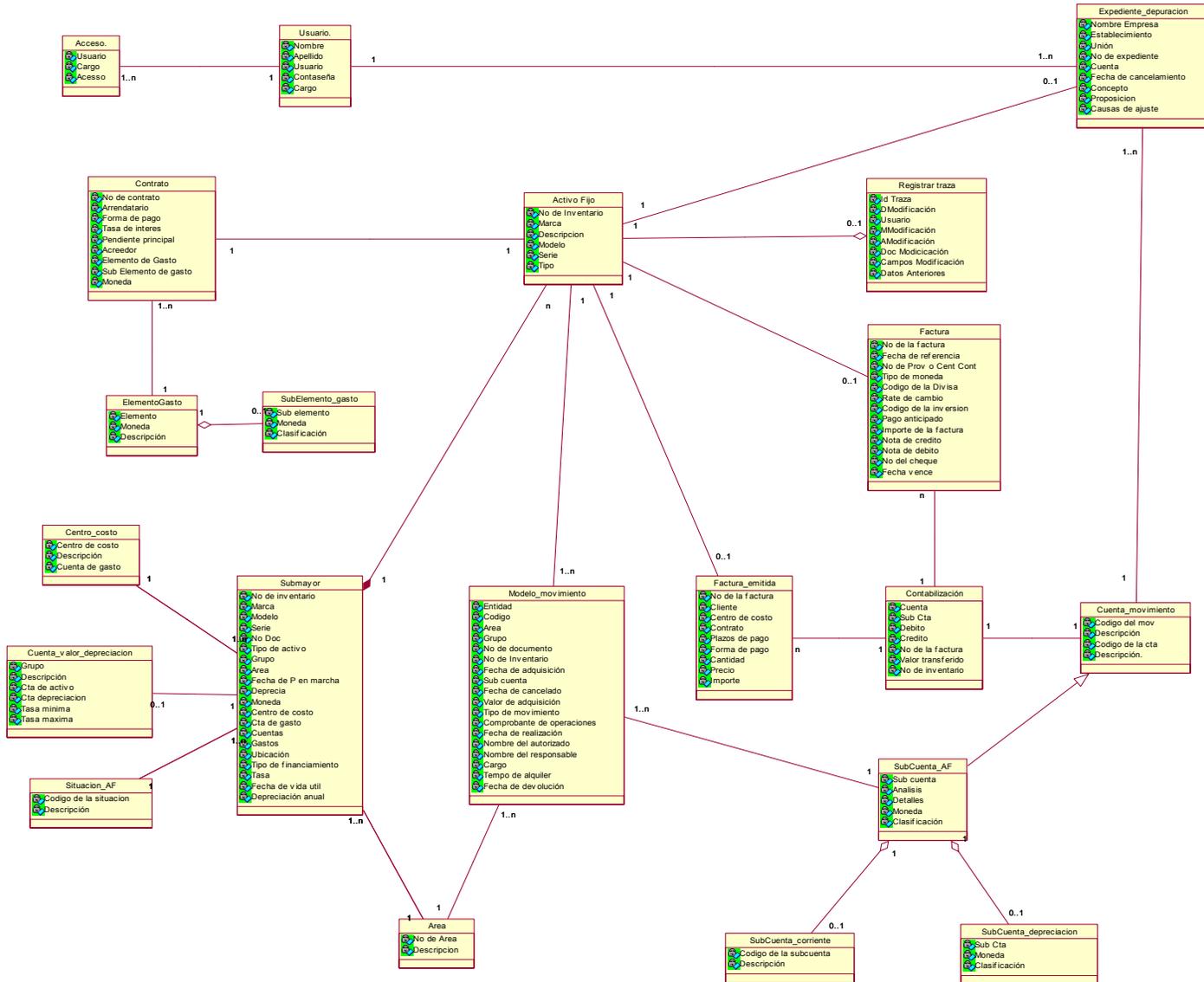


Figura 14. Modelo lógico de datos

4.5.3 Diagrama de Despliegue

El modelo que describe la distribución física del sistema es el **modelo de despliegue**, además de que describe la funcionalidad entre los nodos de cómputo. De allí que podemos observar los siguientes datos del mismo:

Los nodos tienen relaciones que representan la comunicación entre ellos, como HTTP, USB, TCP/IP y otras, La funcionalidad de un nodo se especifica por los componentes que se marcan sobre ese nodo. Se pueden describir varias configuraciones de red. Y además cada recurso representa un medio de cómputo.

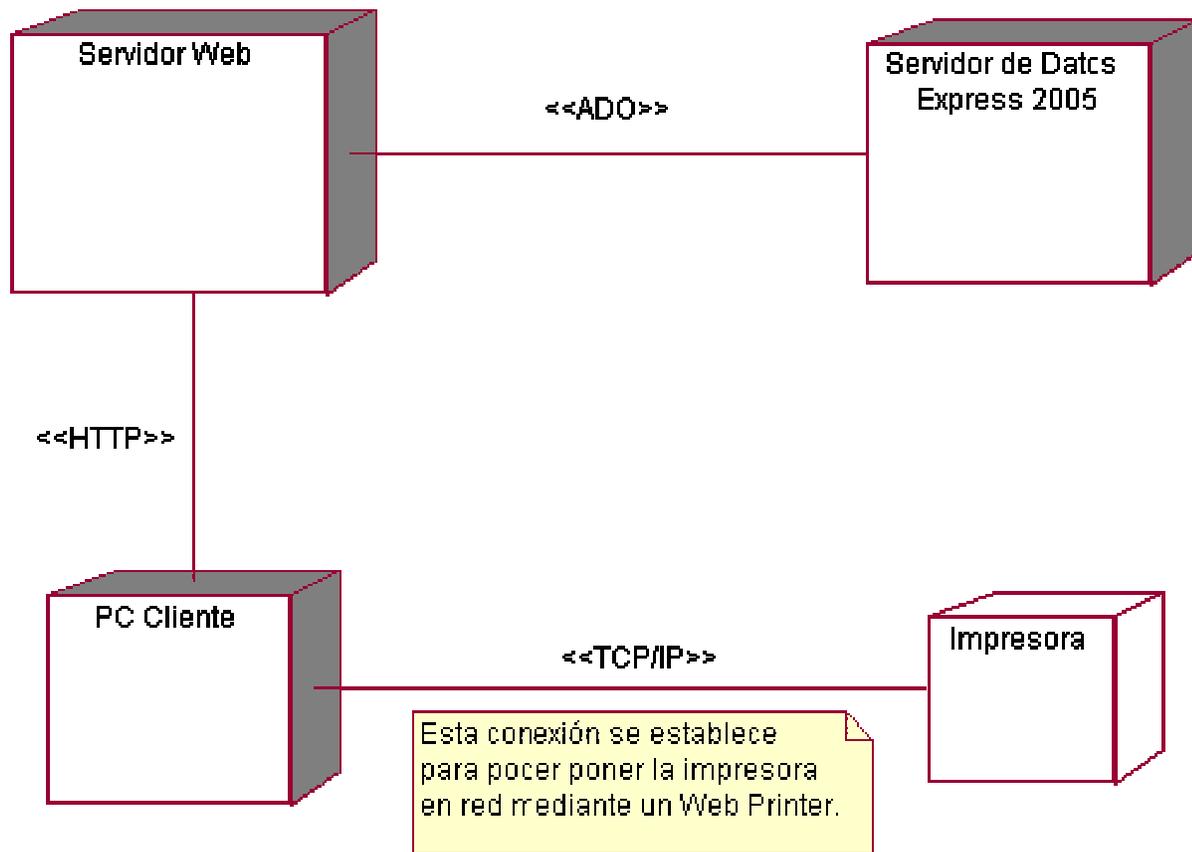


Figura 16. Diagrama de despliegue.

4.6 Generalidades de la implementación

Todos los campos de textos deben de ser validados en la interfaz antes de ser enviados a la BD y además también deben ser validados en el servidor porque en caso de una ataque a la aplicación y modifiquen las validaciones en el cliente todavía existan validaciones en el servidor y no influya en la autenticidad de los datos en la BD. El algoritmo de encriptación a usar será uno creado por el programador ya que el Ministerio del Interior no permite usar ningún algoritmo extranjero y las validaciones en el cliente se realizará usando el lenguaje AJAX+JavaScript. Se deben aplicar estándares para la codificación. Se debe implementar un mecanismo de seguridad para bloquear la BD en caso de ataques. Se debe hacer una salva diaria de la BD para en caso de incendio u otro fenómeno.

4.6.1 Prevención contra inyecciones SQL

En el lenguaje de programación .NET (o el lenguaje de programación Mono) "C#", tenemos ADO.NET SqlCommand (para Microsoft SQL Server) OR OracleCommand (para servidores de bases de datos Oracle). El ejemplo de abajo muestra cómo prevenir los ataques de inyección de código usando el objeto SqlCommand. El código para ADO.NET se programa de forma similar, pero puede variar levemente según la implementación específica de cada proveedor.

En vez de

```
using( SqlConnection con = (acquire connection) ) {
    con. Open();
    using( SqlCommand cmd = new SqlCommand("SELECT * FROM usuarios WHERE nombre = " +
nombreUsuario + "'", con) ) {
        using( SqlDataReader rdr = cmd.ExecuteReader() ){
            ...
        }
    }
}
```

hay que usar lo siguiente

```
using ( SqlConnection con = (acquire connection) ) {
    con. Open();
    using( SqlCommand cmd = new SqlCommand("SELECT * FROM usuarios WHERE nombre =
@nombreUsuario", con) ) {

        cmd.Parameters.AddWithValue("@nombreUsuario", nombreUsuario);

        using( SqlDataReader rdr = cmd.ExecuteReader() ){
            ...
        }
    }
}
```

4.7 Conclusiones parciales

Al hacer uso de la clase My_AccesoDato se elimina la necesidad de implementar métodos de acceso a datos, logrando manejar cualquier clase entidad. El uso de la arquitectura en tres capas y el paradigma de POO han facilitado considerablemente el diseño de las clases que se implementarán.

Capítulo

Estudio de factibilidad

5

5.1 Introducción

Para la obtención un aproximado del tiempo de duración de un proyecto se utilizan metodologías tradicionales para la estimación de proyectos software, las estimaciones están asociadas con el esfuerzo y el tiempo y este con las actividades identificadas en el proyecto.

Se deben estimar las respuestas a las siguientes preguntas:

- ¿Cuánto esfuerzo se requiere para realizar una actividad?
- ¿Cuánto tiempo se necesita para realizar una actividad?
- ¿Cuál es el costo total de una actividad?

5.2 Planificación basada en casos de uso

La estimación mediante el análisis de Puntos de Casos de Uso es un método de estimación del tiempo de desarrollo de un proyecto mediante la asignación de “pesos” a un cierto número de factores que lo afectan, para finalmente, contabilizar el tiempo total estimado para el proyecto a partir de esos factores.

Paso 1. Identificar los Puntos de casos de uso Desajustados.

$$UUCP = UAW + UUCW$$

Donde:

UUCP: Puntos de Casos de Uso sin ajustar.

UAW: Factor de Peso de los Actores sin ajustar.

UUCW: Factor de Peso de los Casos de Uso sin ajustar.

- Cálculo del Factor de Peso de los Actores sin ajustar (UAW):

Se calcula mediante un análisis de la cantidad de actores del sistema y la complejidad de cada uno de ellos.

Tipo de actor	Factor de peso	Actores	Total
Simple	1	0	0
Medio	2	0	0
Complejo	3	3	9
UAW = S(Factor * Actores)			9

Tabla 4. Factor de peso de los actores sin ajustar

- Cálculo del Factor de Peso de los Casos de Uso sin ajustar (UUCW):

Se calcula mediante un análisis de la cantidad de CU presentes en el sistema y la complejidad de cada uno de ellos.

Tipo de CU	Peso	Cantidad de CU	Total
Simple	5	9	45
Medio	10	20	200
Complejo	15	0	0
UUCW = Sumatoria(Factor * CantCU)			245

Tabla 5. Factor de Peso de los Casos de Uso sin ajustar.

$$UUCP = 9 + 245$$

$$UUCP = 254$$

Paso 2. Ajustar los Puntos de casos de uso.

Luego de tener los Puntos de Casos de Uso sin ajustar, se debe ajustar éste valor mediante la siguiente ecuación:

$$UCP = UUCP * TCF * EF$$

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.

- Cálculo de Puntos de Casos de Uso sin ajustar (**UUCP**).

UUCP=254

- Cálculo del Factor de complejidad técnica (**TCF**).

$$TCF = 0.6 + 0.01 \times \Sigma (\text{Peso}_i \times \text{Valor asignado}_i)$$

Se calcula mediante la cuantificación de un conjunto de factores que determinan la complejidad técnica del sistema. Cada uno de los factores se cuantifica con un valor de 0 a 5 (0 No presente o sin influencia, 1 Influencia incidental, 2 Influencia moderada, 3 Influencia media, 4 Influencia significativa y 5 Fuerte influencia)

Factor	Descripción	Peso	Valor	Peso * Valor
T1	Sistema distribuido.	2	0	4
T2	Objetivos del performance o tiempo de respuesta.	1	4	3
T3	Eficiencia del usuario final.	1	3	2

T4	Procesamiento interno complejo.	1	3	2
T5	El código debe ser reutilizable.	1	3	3
T6	Facilidad de instalación.	0.5	4	1.5
T7	Facilidad de uso.	0.5	5	2.5
T8	Portabilidad.	2	3	0
T9	Facilidad de cambio.	1	4	3
T10	Concurrencia.	1	4	4
T11	Incluye objetivos especiales de seguridad.	1	5	4
T12	Provee acceso directo a terceras partes.	1	0	0
T13	Se requieren facilidades especiales de entrenamiento a los usuarios.	1	1	1
TOTAL Σ (Peso_i * Valor_i)				37.5

Tabla 6. Factor de complejidad técnica.

$$TCF = 0.6 + 0.01 * \text{Sumatoria (Peso * Valor)}$$

$$TCF = 0.6 + 0.01 * 37.5$$

$$TCF = 0,975$$

- Cálculo del Factor de Ambiente (EF):

$$EF = 1.4 - 0.03 * \Sigma (\text{Peso}_i * \text{Valor}_i) \text{ (Donde Valor es un número del 0 al 5)}$$

Las habilidades y el entrenamiento del grupo involucrado en el desarrollo del sistema informático tienen un gran impacto en las estimaciones de tiempo (son los que se contemplan en este cálculo).

Factor	Descripción	Peso	Valor	$\Sigma (\text{Peso}_i * \text{Valor}_i)$
E1	Familiaridad con el modelo de proyecto utilizado.	1.5	3	6
E2	Experiencia en la aplicación.	0.5	2	0
E3	Experiencia en orientación a objetos.	1	4	4
E4	Capacidad del analista líder.	0.5	2	1.5
E5	Motivación.	1	5	5
E6	Estabilidad de los requerimientos	2	4	8
E7	Personal part-time	-1	5	-5
E8	Dificultad del lenguaje de programación	-1	2	-2
TOTAL $\Sigma (\text{Peso}_i * \text{Valor}_i)$				17.5

Tabla 7. Factor de Ambiente.

$$EF = 1.4 - 0.03 * 17.5$$

$$EF = 0.875$$

Luego

Los Puntos de casos de uso ajustados (UCP) es de:

$$UCP = UUCP \times TCF \times EF$$

$$UCP = 254 \times 0.975 \times 0.875$$

$$UCP = 216.69375$$

Paso 3. Calcular esfuerzo a través de los puntos de casos de uso.

El esfuerzo en horas-hombre viene dado por:

$$E = UCP * CF$$

Donde

- **E:** esfuerzo estimado en horas-hombre.
- **UCP:** Puntos de Casos de Uso ajustados.
- **CF:** factor de conversión.

Para calcular CF:

$$CF = 20 \text{ horas-hombre (si Total}_{EF} \leq 2)$$

$$CF = 28 \text{ horas-hombre (si Total}_{EF} = 3 \text{ ó Total}_{EF} = 4)$$

$$CF = \text{abandonar o cambiar proyecto (si Total}_{EF} \geq 5)$$

$$\text{Total}_{EF} = \text{Cant EF} < 3 \text{ (entre E1 –E6)} + \text{Cant EF} > 3 \text{ (entre E7, E8)}$$

$$\text{Como Total}_{EF} = 1 + 1$$

$$\text{Total}_{EF} = 2$$

$$CF = 20 \text{ horas-hombre (porque Total}_{EF} \leq 2)$$

Luego

$$E = 216.69375 * 20 \text{ horas-hombre}$$

$$E = 4333.875 \text{ horas-hombre}$$

Paso 4. Calcular esfuerzo de todo el proyecto.

El procedimiento aporta una estimación del esfuerzo en horas-hombre observando solo el desarrollo de la funcionalidad especificada en los casos de usos. Para lograr obtener una estimación más exacta de la duración total del proyecto se debe agregar a la estimación del esfuerzo lograda por los puntos de casos de uso, las estimaciones de esfuerzo de las demás actividades relacionadas con el desarrollo de software. Para lograr esto podemos tener en cuenta el siguiente criterio:

El criterio plantea la distribución del esfuerzo entre las diferentes actividades de un proyecto.

Actividad	% esfuerzo	Valor esfuerzo
Análisis	10%	1083.46875 horas-hombre
Diseño	20%	2166.9375 horas-hombre
Implementación	40%	4333.875 horas-hombre
Prueba	15%	1625.20313 horas-hombre
Sobrecarga	15%	1625.20313 horas-hombre
Total	100%	10834.6875 horas-hombre

Tabla 8. Distribución del Esfuerzo

El esfuerzo total (ET1) es de 10834.6875 horas-hombre y el esfuerzo total (ET2) es de 61.6 mes – hombre. Esto significa que 1 persona puede realizar el sistema en aproximadamente 62 meses.

Paso 5. Costo del proyecto.

CHM: Costo Hombre Mes.

ET: Esfuerzo total (mes-hombre).

SBM: Salario básico mensual.

Se asume un SBM de \$100.00.

$CHM = 2 \times SBM$

CHM = 200.00 pesos/mes

Costo = CHM x ET

Costo = 200 x 61.6

Costo = \$ 6156

5.3 Beneficios tangibles e intangibles

5.3.1 Beneficios tangibles

- Ahorro de factura telefónica, por conceptos de llamadas para la obtención de datos necesarios para realizar las operaciones con los activos.
- Un buen ahorro de papeles, por conceptos de informes relacionados con la contabilidad de activos.
- Ganancias económicas, ahorro de tiempo, costes y otros.

5.3.2 Beneficios intangibles

- Interfaz gráfica sencilla.
- Impresión de las páginas y de los informes desde la aplicación.
- Integridad de los datos en la base de datos.
- Información legible.
- Búsquedas rápidas y eficientes.
- Rápida gestión de la información.
- Seguridad de la información.

5.4 Análisis de costo y beneficios

Es sistema será desarrollado con pocos recursos, pero en dependencia de la cantidad de trabajadores será el tiempo de desarrollo; el servidor puede ser cualquier máquina del departamento de Contabilidad y finanzas de la empresa donde se utilice. La aplicación se desarrollada con tecnología .NET.

5.5 Conclusiones parciales

Luego de haber realizado un estudio profundo de la factibilidad a lo largo de este capítulo podemos decir que existen dos factores fundamentales en la estimación de un proyecto software, su duración y costo, aunque podemos decir que la duración es primordial ya que la información toma un papel cada vez más importante en su desarrollo. Podemos decir además que el proyecto proporciona resultados favorables para el departamento de contabilidad de cualquier empresa que lo utilice.

Conclusiones

Dado que el trabajo se encuentra finalizado, podemos decir que se han cumplido exitosamente los objetivos perfilados, se logró modelar una aplicación Web cumpliendo los requisitos implantados por el cliente, aplicación capaz de realizar todo el proceso de control de los activos fijos de cualquier empresa.

Para lograr una excelente modelado se realizaron estudios y análisis de software ya existentes, como el desarrollado por SICS, el **ConDor**, además de este nos sirvieron de base otros ya existente en Cuba. El sistema se modeló orientado por la metodología RUP, y se alcanzó llevar la misma hasta el análisis y el diseño. Para su implementación se propone utilizar la plataforma.NET además de ser una petición del cliente. Como gestor de BD se propone SQL Server 2005 Express ya que su principal característica es que es libre y brinda buena seguridad. Como lenguaje de programación se propone el C# por la gran gama de ventajas que ofrece.

Se hace un análisis del negocio de la gestión de activos fijos en las empresas y de allí se hace una propuesta del sistema a futura implementación, la aplicación propuesta consta de una ambiente amigable, fácil de usar y cumple con los requisitos expuestos.

No obstante al tiempo y costo de desarrollo calculado, después de un profundo estudio de la factibilidad de la aplicación, podemos concluir que la aplicación es factible para ser implementada.

Recomendaciones

Con el objetivo de una futura conclusión de la aplicación y mejora de la misma, se recomienda:

- ✓ Continuar el modelado de las operaciones y comprobantes de activos, funcionalidades que no se investigaron por el alcance impuesto en la misma, debido a su tiempo de desarrollo.
- ✓ Realizar la implementación de la aplicación propuesta, para su futura utilización y puesta en marcha en los servicios del departamento de contabilidad y finanzas de las empresas que la adquieran.
- ✓ Realizar la continuidad de la investigación sobre la Gestión Empresarial en general para lograr hacer un ERP totalmente cubano.
- ✓ Realizar en la implementación de la aplicación, validación en el cliente y en el servidor, en el cliente con AJAX+JavaScript.
- ✓ Se utilice un algoritmo de encriptación creado por el propio programador, a partir de las políticas existentes en el país para la utilización de algoritmos de encriptación.
- ✓ Tener en cuenta la política de seguridad presentada en el capítulo 2, epígrafe 2.9.

Referencias Bibliográficas

- CERIA, S. *Ingeniería de Software I. Casos de Uso. Un Método Práctico para Explorar Requerimientos*. 2003. p.
- Cliente-servidor*. 2006. [Disponible en: <http://es.wikipedia.org/wiki/Cliente-servidor>]
- Desarrollo web*. 2000. [Disponible en: <http://www.desarrolloweb.com>]
- El por qué de la metodología*. 2003. [Disponible en: <http://www.um.es/docencia/barzana/IAGP/Iagp1.html>]
- ERP Sistema de Gestión*. 2005. [Disponible en: http://www.adpime.com/ERP/Es_ERP_intro.htm]
- FOWLER, M. *UML Gota a Gota*. 1999. p.
- KENDALL, K. *Análisis Y Diseño De Sistemas*. 3ª. Edición, 2005.
- . *Análisis Y Diseño De Sistemas*. 3ª. Edición.
- LA CONTABILIDAD Y EL SISTEMA CONTABLE* 2003. [Disponible en: <http://www.gestiopolis.com/recursos/documentos/fulldocs/fin/contaysiscontable.htm>]
- LARMAN, C. *UML y Patrones, Introducción al análisis y diseño orientado a objetos.*, Prentice-Hall, 2002. p.
- Manual de desarrollo web* 2004. [Disponible en: <http://www.desarrolloweb.com/manuales/58/>]
- MARTIN, E. L. *Un efectivo sistema cubano de contabilidad*, 2005. [Disponible en: <http://www.economista.cubaweb.cu/2005/edicionimpresa/premioeconomiamiguel.html>]
- Microsoft SQL Server 2005 Express*. 2005. [Disponible en: <http://www.microsoft.com/spanish/msdn/vstudio/express/SQL/default.msp>]
- MIGUEL ANGEL MARTINEZ, M. V. M. *Contabilidad y finanzas para la formación económica*.
- MIGUEL CASTAÑO, M. G. P. V. *Diseño de Bases de datos*. 1999. p. Editorial Alfaomega S.A.
- SELLERS, D. C. Y. D. *Inicie su negocio en Web*. 2001. p.
- Universidad .NET*. Disponible en: <http://www.microsoft.com/spanish/msdn/comunidad/uni.net/>
- Visual Studio 2005*. 2005. [Disponible en: <http://www.microsoft.com/emea/msdn/visualstudio/eses/getthetrials/>]
- ZAVALA, R. *Ingeniería de Software*. Mexico, 2000. p.

Bibliografías

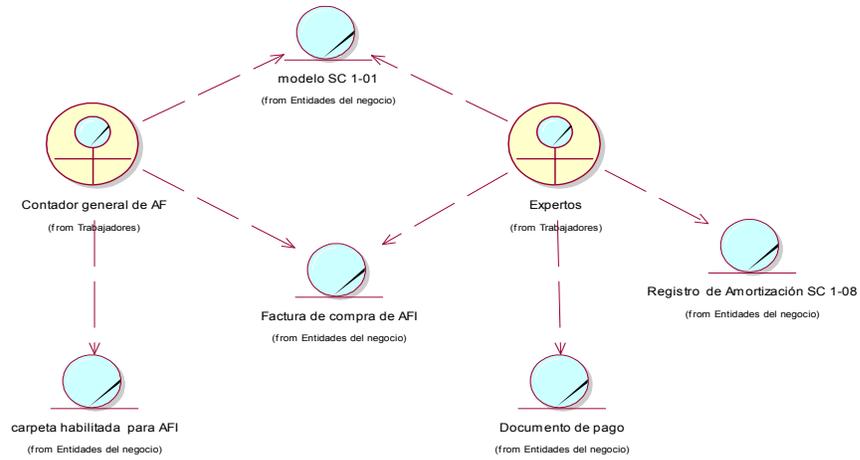
Sitios visitados

- http://bibliodoc.uci.cu/pdf/cont_avanzada.pdf
- <http://bibliodoc.uci.cu/pdf/reg03319.pdf>
- <http://bibliodoc.uci.cu/pdf/reg02901.pdf>
- http://www.cii-murcia.es/informas/ene05/articulos/Arquitectura_y_diseño_de_sistemas_web_modernos.html
- http://www.programacion.com/articulo/lenguajes_patrones
- <http://www.microsoft.com/spanish/msdn/vcsharp/default.asp>
- http://es.wikipedia.org/wiki/C_Sostenido
- http://es.wikipedia.org/wiki/Microsoft_SQL_Server
- <http://www.microsoft.com/latam/sql/>
- http://www.microsoft.com/spanish/msdn/comunidad/mtj.net/voices/MTJ_3317.asp
- <http://es.wikipedia.org/wiki/Cliente-servidor>
- <http://www.desarrolloweb.com>

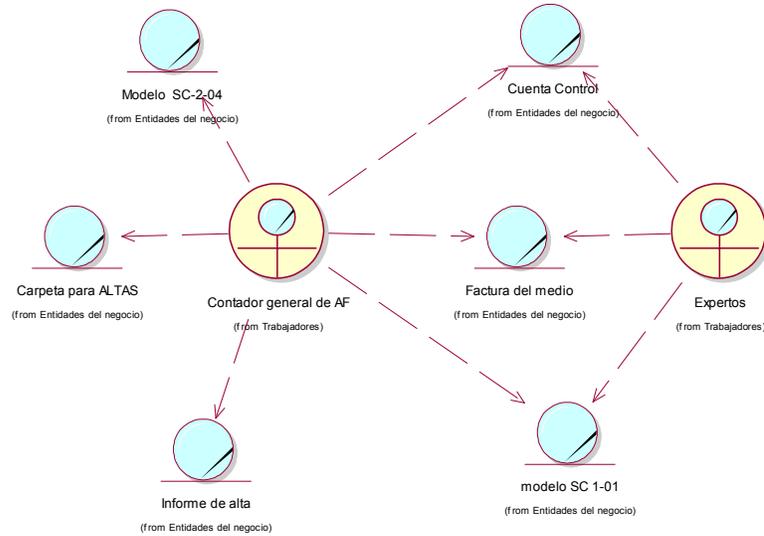
- <http://www.um.es/docencia/barzana/IAGP/Iagp1.html>
- <http://www.microsoft.com/emea/msdn/visualstudio/eses/getthetrials/>
- <http://www.microsoft.com/spanish/msdn/comunidad/uni.net/>
- <http://www.microsoft.com/spanish/msdn/vstudio/express/SQL/default.mspix>
- <http://www.eleconomista.cubaweb.cu/2005/edicionimpresa/premioeconomiamiguel.html>
- <http://www.desarrolloweb.com/manuales/58/>
- <http://www.gestiopolis.com/recursos/documentos/fulldocs/fin/contaysiscontable.htm>
- **Guía de estilo de codificación**, Copyright © 2006 Ignasi Fosch Alonso, Javier Arellano Roig.

Anexo 1. Modelo de Objetos.

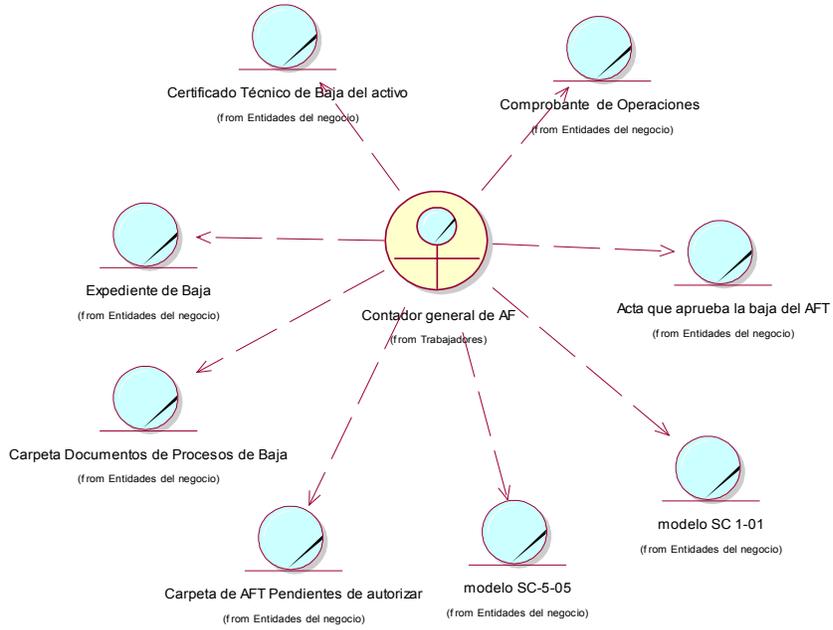
✓ Controlar AFI.



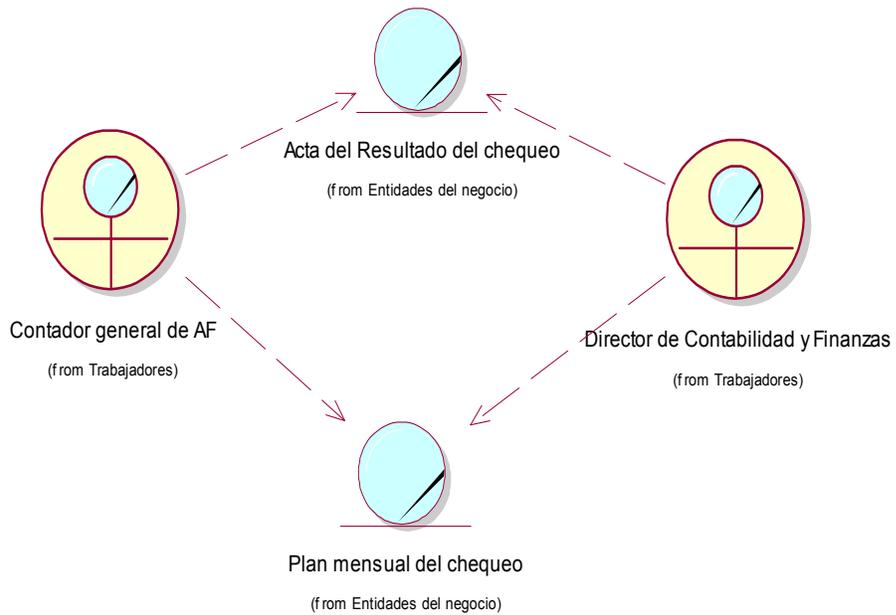
✓ Procesar Altas de AF.



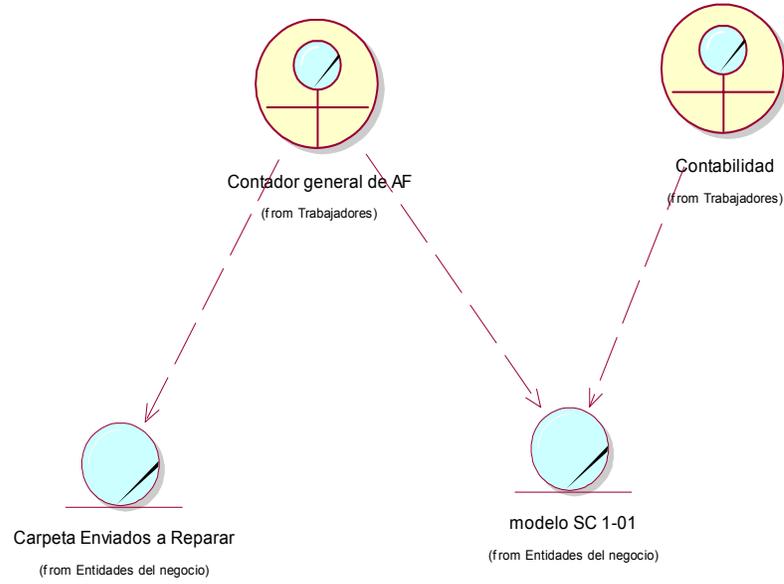
✓ Procesar bajas de AF.



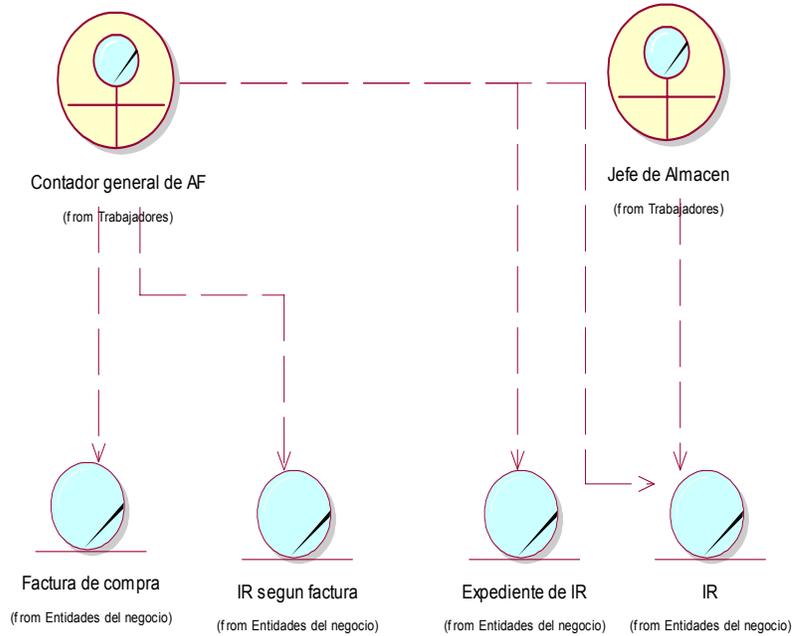
✓ Realizar Chequeos mensuales.



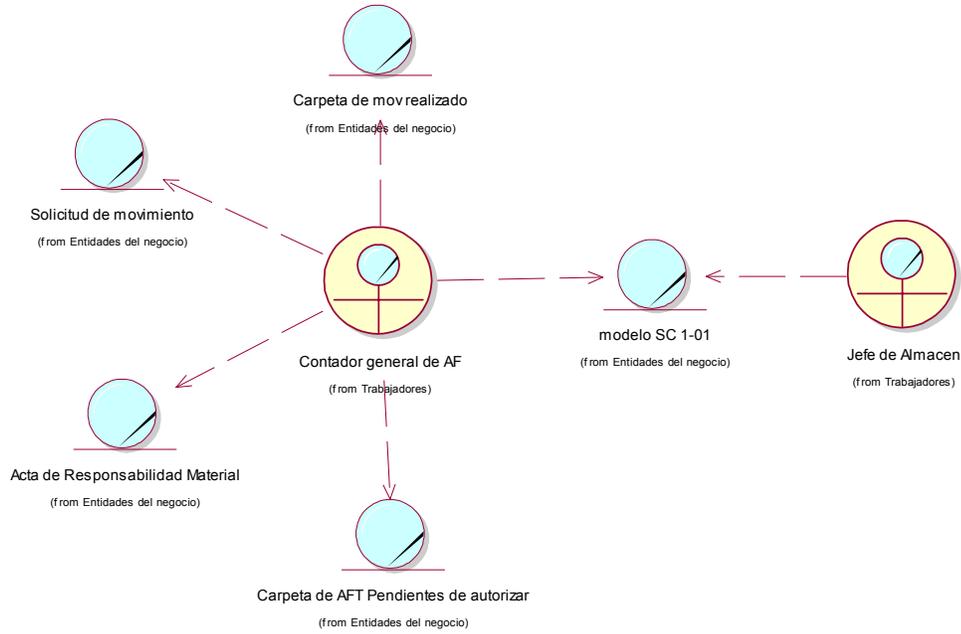
✓ Realizar envíos a reparar.



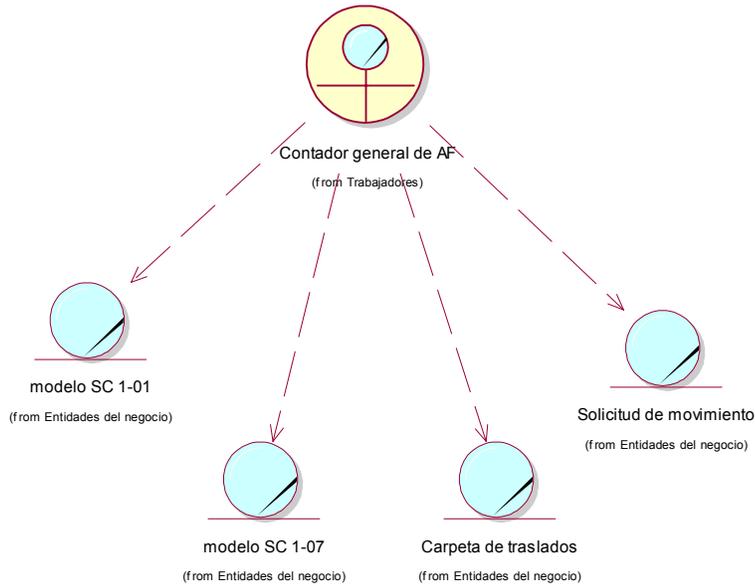
✓ Realizar recepción.



✓ Realizar salidas a uso.



✓ Realizar traslados entre áreas.



Anexo 2. Descripciones textuales de los Casos de Uso del Sistema.

Nombre del Caso de Uso	
Gestionar tarjetas de activos fijos.	
Actores	Contador general de activos fijos (Inicia).
Propósito	Gestionar (insertar, modificar, eliminar, buscar y visualizar y imprimir) datos de apertura.
Resumen	El caso de uso se inicia cuando el contador general de AF selecciona la opción insertar, modificar, eliminar, Buscar y visualizar datos de tarjetas de activos fijos.
Referencias	R 5.1, R 5.2, R5.3 y R 5.4
Precondiciones	
Requerimientos especiales	
Curso normal de los eventos:	
Acción del actor:	Respuesta del proceso del Sistema:
1. El contador general de AF selecciona la opción gestionar datos de apertura.	1.1. El sistema muestra las opciones: insertar, modificar, eliminar, buscar y visualizar e imprimir apertura de datos.
Escenario 1: Insertar datos de apertura.	
1. El contador general de AF selecciona la opción insertar apertura de datos.	1.1 El sistema muestra el formulario completo para insertar los datos de apertura.
2. El contador general de AF introduce los datos solicitados por el sistema.	2.1 El sistema verifica que los datos introducidos sean correctos
	2.3 Si los datos son correctos el sistema adiciona los datos introducidos en la base de datos correspondiente, terminado así el caso de uso.
Curso alterno de los eventos:	
Acción 2.3	2.3 Si los datos introducidos por el contador general de AF son incorrectos el sistema muestra un mensaje de error indicando donde se encuentra el dato erróneo e indica al usuario retornar a la acción 2.
Escenario 2: Modificar datos de apertura.	

1. El contador general de AF de la entidad selecciona la opción modificar datos de apertura.	1.1 El sistema muestra un listado de los datos de apertura existentes en la base de datos.
2. El contador general de AF selecciona los datos para modificarlo.	2.1 El sistema localiza los datos de apertura seleccionado, los muestra listo para modificarlos.
3. El contador general de AF realiza los cambios necesarios.	3.1 El sistema verifica los datos modificados por el contador general de AF
	3.2 Si los datos modificados son correctos, el sistema actualiza dichos datos en la base de datos correspondiente.
Curso alterno de los eventos:	
Acción 3.2	Si los datos introducidos por el contador general de AF son incorrectos el sistema muestra un mensaje de error indicando donde se encuentra el dato erróneo e indica al usuario retornar a la acción 3.
Escenario 3: Eliminar datos de apertura.	
1. El contador general de AF selecciona la opción eliminar datos de apertura.	1.1 El sistema muestra un listado de los datos de apertura existentes en la base de datos.
2. El contador general de AF selecciona los datos para eliminarlo.	2.1 El sistema localiza los datos de apertura, los muestra listos para eliminarlos.
3. El contador general de AF selecciona la opción de eliminar.	3.1 El sistema muestra un mensaje de advertencia para la acción a realizar.
4. El contador general de AF confirma si quiere o no eliminar el acceso.	4.1 Si el contador general de AF acepta el sistema elimina los datos del acceso seleccionado y culmina así el caso de uso.,
Curso alterno de los eventos:	
Acción 4.1	Si el contador general de AF cancela la acción de eliminar culmina el caso de uso sin realizar ninguna acción.
Escenario 4: Buscar y visualizar datos de apertura.	

1. El contador general de AF selecciona la opción buscar y visualizar datos de apertura.	1.1 El sistema muestra un listado de los datos de apertura existentes en la base de datos.
2. El contador general de AF selecciona la opción imprimir datos de apertura.	2.1 Si el contador general de AF selecciona la opción imprimir el sistema manda a imprimir los datos de apertura y culmina así el caso de uso.
Curso alterno de los eventos:	
Acción 2.1	Si no selecciona la opción de imprimir el caso de uso culmina con la visualización de los datos de apertura.
Prioridad :	Critico.

Nombre del Caso de Uso		Gestionar factura recibida.
Actores	Contador general de activos fijos (Inicia).	
Propósito	Permite al contador general de AF gestionar (Insertar, modificar, eliminar, buscar y visualizar) factura recibida.	
Resumen	El caso de uso se inicia cuando el contador general de AF selecciona la opción insertar, modificar, eliminar, buscar y visualizar datos de la factura recibida.	
Referencias	R 8.1, R 8.2, R 8.3 y R 8.4	
Precondiciones		
Requerimientos especiales		
Curso normal de los eventos:		
Acción del actor:	Respuesta del proceso del Sistema:	
1. El contador general de AF selecciona la opción gestionar factura recibida.	El sistema muestra la opción: insertar, modificar, eliminar, buscar y visualizar factura recibida.	
Escenario 1: Insertar datos de la factura recibida.		
1. El contador general de AF selecciona la opción insertar datos de la factura	1.1 El sistema muestra el formulario correspondiente para la inserción de los datos de una nueva factura.	

recibida.	
2. El contador general de AF introduce los datos solicitados por el sistema.	2.1 El sistema verifica que los datos introducidos por el contador general de AF sean correctos.
	2.2 Si los datos introducidos por el contador general de AF son correctos, el sistema inserta los datos en la base de datos correspondiente y culmina así el caso de uso.
Curso alterno de los eventos:	
Acción 2.2	Si los datos introducidos por el contador general de AF son incorrectos, el sistema muestra un mensaje de error indicando donde se encuentra el dato erróneo e indica al usuario retornar a la acción 2.
Escenario 2: Modificar datos de la factura recibida.	
1. El contador general de AF selecciona la opción de modificar datos de la factura recibida.	1.1 El sistema muestra un listado de las facturas recibidas existente en la base de datos.
2. El contador general de AF selecciona la factura a la que le va a hacer modificaciones.	2.1 El sistema localiza los datos de la factura seleccionada por el contador general de AF y los muestra listos para modificarlos.
3. El contador general de AF realiza los cambios necesarios.	3.1 El sistema verifica que los datos introducidos por el contador general de AF sean correctos.
	3.2 Si los datos introducidos por el contador general de AF son correctos, el sistema actualiza los datos de la factura en la base de datos.
Curso alterno de los eventos:	
Acción 3.2	Si los datos introducidos por el contador general de AF son incorrectos, el sistema muestra un mensaje de error indicando donde se encuentran los datos erróneos e indica al usuario retornar a la acción 3
Escenario 3: Eliminar factura recibida.	

1. El contador general de AF selecciona la opción de eliminar factura recibida.	1.1 El sistema muestra un listado de las facturas recibidas existente en la base de datos.
2. El contador general de AF selecciona la factura que va a eliminar.	2.1 El sistema localiza la factura seleccionada por el contador general de AF y la muestra lista para eliminarla.
3. El contador general de AF selecciona la opción de eliminar.	3.2 El sistema muestra un mensaje de advertencia para la acción a realizar.
4. El contador general de AF confirma si quiere o no eliminar la factura.	4.1 Si el contador general de AF acepta, el sistema elimina la factura selecciona y culmina así el caso de uso.
Curso alternativo de los eventos:	
Acción 4.1	Si el contador general de AF cancela, culmina el caso de uso sin realizarse ninguna acción.
Escenario 4: Buscar y visualizar datos de la factura recibida.	
1. El contador general de AF selecciona la opción de buscar y visualizar factura recibida.	1.1 El sistema muestra una ventana para introducir el No de la factura deseada.
2. El contador general de AF introduce el No de la factura.	2.1 El sistema verifica que la factura exista en la base de datos.
	2.2 Si la factura existe, el sistema muestra los datos de la factura deseada y permite la opción de imprimir.
3. El contador general de AF confirma si quiere imprimir o no.	3.1 Si el contador general de activos fijos acepta, el sistema manda a imprimir los datos de la factura y culmina así el caso de uso.
Curso alternativo de los eventos:	
Acción 2.2	Si la factura no existe, el sistema muestra un mensaje de error e indica al usuario retornar a la acción 2
Acción 3.1	Si el contador general cancela culmina el caso de uso sin realizar la acción de imprimir.
Prioridad :	Critico.

Nombre del Caso de Uso		Gestionar modelo de movimiento.
Actores	Contador general de activos fijos (Inicia).	
Propósito	Permite al contador general de AF gestionar (insertar, modificar, eliminar, buscar y visualizar) modelo de movimiento.	
Resumen	El caso de uso se inicia cuando el contador general de AF selecciona la opción Insertar, modificar, eliminar, buscar y visualizar datos de movimiento.	
Referencias	R 9.1, R 9.2, R9.3 y R 9.4	
Precondiciones		
Requerimientos especiales		
Curso normal de los eventos:		
Acción del actor:		Respuesta del proceso del Sistema:
1. El contador general de AF selecciona la opción gestionar modelo de movimiento.		1.1. El sistema muestra la opción: insertar, modificar, eliminar, buscar y visualizar modelo de movimiento.
Escenario 1: Insertar modelo de movimiento.		
1. El contador general de AF selecciona la opción insertar modelo de movimiento.		1.1 El sistema muestra el formulario completo para insertar los datos de modelo de movimiento.
2. El contador general de AF introduce los datos solicitados por el sistema.		2.1 El sistema verifica que los datos introducidos sean correctos
		2.2 Si los datos son correctos el sistema adiciona los datos introducidos en la base de datos correspondiente, terminado así el caso de uso.
Curso alterno de los eventos:		
Acción 2.3		2.3 Si los datos introducidos por el contador general de AF son incorrectos el sistema muestra un mensaje de error indicando donde se encuentra el dato erróneo e indica al usuario retornar a la acción 2.
Escenario 2: Modificar modelo de movimiento.		

1. El contador general de AF de la entidad selecciona la opción modificar datos de modelo de movimiento.	1.1 El sistema muestra un listado de todos los modelos de movimientos existente en la base de datos.
2. El contador general de AF selecciona el modelo de movimiento para modificarlo.	2.1 El sistema localiza los datos de modelo de movimiento seleccionado y los muestra listo para modificarlos.
3. El contador general de AF realiza los cambios necesarios.	3.1 El sistema verifica los datos modificados por el contador general de AF
	3.2 Si los datos modificados son correctos, el sistema actualiza dichos datos en la base de datos correspondiente.
Curso alterno de los eventos:	
Acción 3.2	Si los datos introducidos por el contador general de AF son incorrectos el sistema muestra un mensaje de error indicando donde se encuentra el dato erróneo e indica al usuario retornar a la acción 3.
Escenario 3: Eliminar modelo de movimiento.	
1. El contador general de AF selecciona la opción eliminar modelo de movimiento.	1.1 El sistema muestra un listado de todos los modelo de movimiento existentes en la base de datos.
2. El contador general de AF selecciona el modelo de movimiento para eliminarlo.	2.1 El sistema localiza los datos del modelo de movimiento seleccionado y los muestra listos para eliminarlos.
3. El contador general de AF selecciona la opción de eliminar.	3.1 El sistema muestra un mensaje de advertencia para la acciona a realizar.
4. El contador general de AF confirma si quiere o no eliminar el modelo de movimiento.	4.1 Si el contador general de AF acepta el sistema elimina el modelo de movimiento seleccionado y culmina así el caso de uso.,
Curso alterno de los eventos:	
Acción 4.1	Si el contador general de AF cancela la acción de eliminar, culmina el caso de uso sin realizar ninguna acción.
Escenario 4: Buscar y visualizar datos del modelo de movimiento.	

1. El contador general de AF selecciona la opción de buscar y visualizar modelo de movimiento.	1.1 El sistema muestra una ventana para introducir el No de la factura y el código del movimiento del modelo de movimiento deseado.
2. El contador general de AF introduce los datos solicitados por el sistema.	2.1 El sistema verifica que los datos existan en la base de datos.
	2.2 Si los datos son correctos, el sistema muestra los datos del modelo de movimiento deseado y permite la opción de imprimir.
3. El contador general de AF confirma si quiere imprimir o no.	3.1 Si el contador general de activos fijos acepta, el sistema manda a imprimir los datos del modelo de movimiento y culmina así el caso de uso.
Curso alterno de los eventos:	
Acción 2.2	Si los datos son no existen en la base de datos, el sistema muestra un mensaje de error e indica al usuario retornar a la acción 2
Acción 3.1	Si el contador general cancela, culmina el caso de uso sin realizar la acción de imprimir.
Prioridad :	Critico.

Nombre del Caso de Uso		Gestionar submayor.
Actores	Contador general de activos fijos (Inicia).	
Propósito	Permite al contador general de AF gestionar (insertar, modificar, eliminar, buscar y visualizar) submayor.	
Resumen	El caso de uso se inicia cuando el contador general de AF selecciona la opción Insertar, modificar, eliminar, buscar y visualizar submayor.	
Referencias	R 10.1, R 10.2, R 10.3 y R 10.4	
Precondiciones		
Requerimientos especiales		

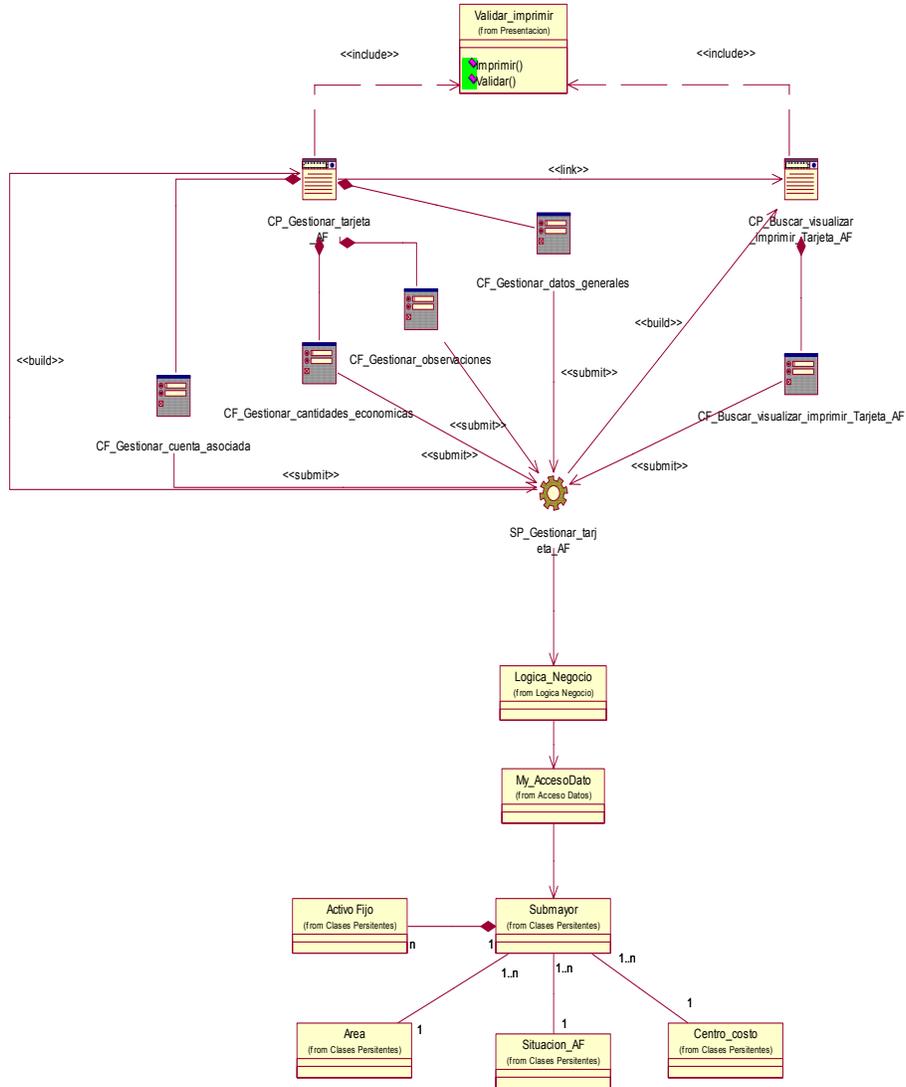
Curso normal de los eventos:	
Acción del actor:	Respuesta del proceso del Sistema:
1. El contador general de AF selecciona la opción gestionar submayor.	1.1. El sistema muestra la opción: insertar, modificar, eliminar, buscar y visualizar submayor.
Escenario 1: Insertar submayor.	
1. El contador general de AF selecciona la opción insertar submayor.	1.1 El sistema muestra el formulario completo para insertar los datos del submayor.
2. El contador general de AF introduce los datos solicitados por el sistema.	2.1 El sistema verifica que los datos introducidos sean correctos
	2.2 Si los datos son correctos el sistema adiciona los datos introducidos en la base de datos correspondiente, terminado así el caso de uso.
Curso alterno de los eventos:	
Acción 2.3	2.3 Si los datos introducidos por el contador general de AF son incorrectos el sistema muestra un mensaje de error indicando donde se encuentra el dato erróneo e indica al usuario retornar a la acción 2.
Escenario 2: Modificar submayor.	
1. El contador general de AF de la entidad selecciona la opción modificar datos del submayor.	1.1 El sistema muestra un listado de todos los submayores existentes en la base de datos.
2. El contador general de AF selecciona el submayor para modificarlo.	2.1 El sistema localiza los datos del submayor seleccionado y los muestra listo para modificarlos.
3. El contador general de AF realiza los cambios necesarios.	3.1 El sistema verifica los datos modificados por el contador general de AF
	3.2 Si los datos modificados son correctos, el sistema actualiza dichos datos en la base de datos correspondiente.
Curso alterno de los eventos:	
Acción 3.2	Si los datos introducidos por el contador general de AF son

	incorrectos el sistema muestra un mensaje de error indicando donde se encuentra el dato erróneo e indica al usuario retornar a la acción 3.
Escenario 3: Eliminar submayor.	
1. El contador general de AF selecciona la opción eliminar submayor.	1.1 El sistema muestra un listado de todos los submayores existentes en la base de datos.
2. El contador general de AF selecciona el submayor para eliminarlo.	2.1 El sistema localiza los datos del submayor seleccionado y los muestra listos para eliminarlos.
3. El contador general de AF selecciona la opción de eliminar.	3.1 El sistema muestra un mensaje de advertencia para la acción a realizar.
4. El contador general de AF confirma si quiere o no eliminar el submayor.	4.1 Si el contador general de AF acepta el sistema elimina el submayor seleccionado y culmina así el caso de uso.,
Curso alterno de los eventos:	
Acción 4.1	Si el contador general de AF cancela la acción de eliminar, culmina el caso de uso sin realizar ninguna acción.
Escenario 4: Buscar y visualizar datos del submayor.	
1. El contador general de AF selecciona la opción de buscar y visualizar submayor.	1.1 El sistema muestra una ventana para introducir el No de la factura del submayor deseado.
2. El contador general de AF introduce los datos solicitados por el sistema.	2.1 El sistema verifica que los datos existan en la base de datos.
	2.2 Si los datos son correctos, el sistema muestra los datos del submayor deseado y permite la opción de imprimir.
3. El contador general de AF confirma si quiere imprimir o no.	3.1 Si el contador general de activos fijos acepta, el sistema manda a imprimir los datos del submayor y culmina así el caso de uso.
Curso alterno de los eventos:	
Acción 2.2	Si los datos son no existen en la base de datos, el sistema muestra un mensaje de error e indica al usuario retornar a

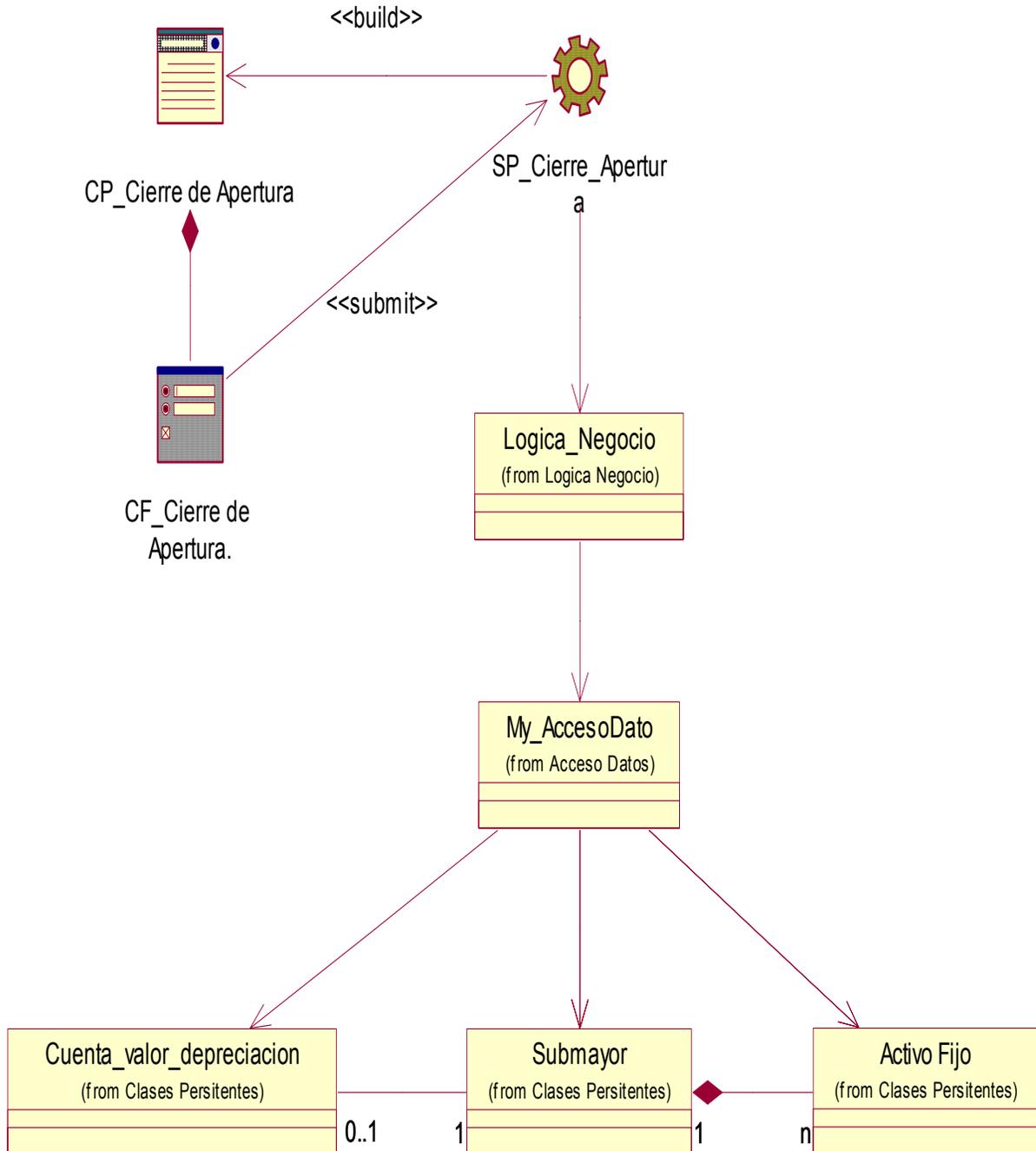
	la acción 2
Acción 3.1	Si el contador general cancela, culmina el caso de uso sin realizar la acción de imprimir.
Prioridad :	Critico.

Anexo 3. Diagrama de clases del diseño.

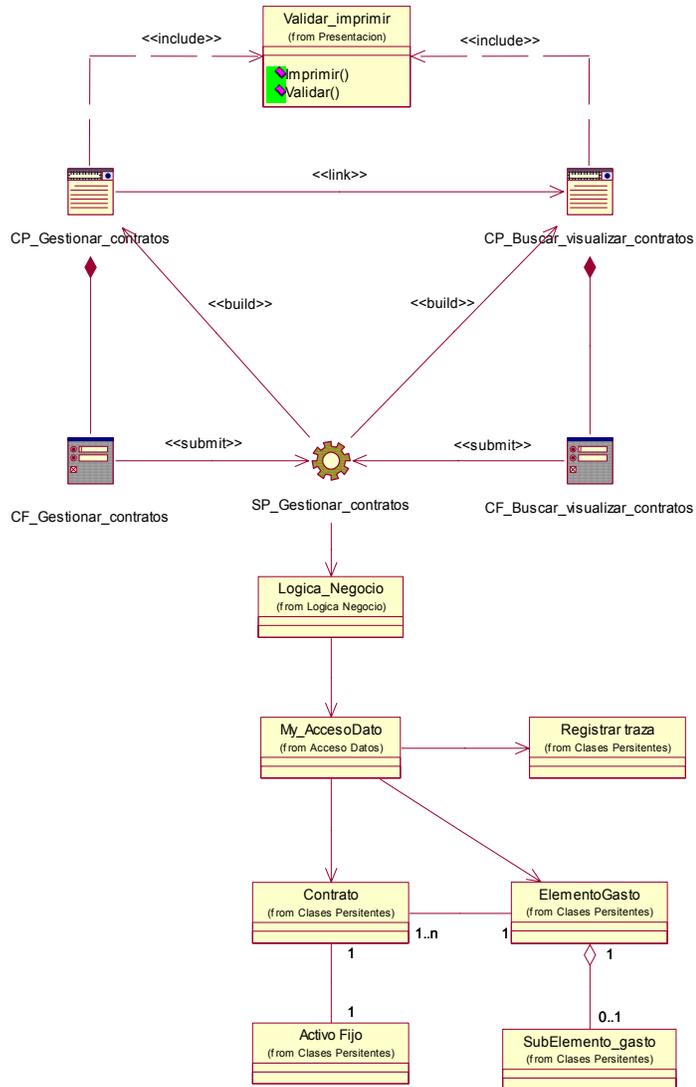
➤ Gestionar tarjeta de activos fijos.



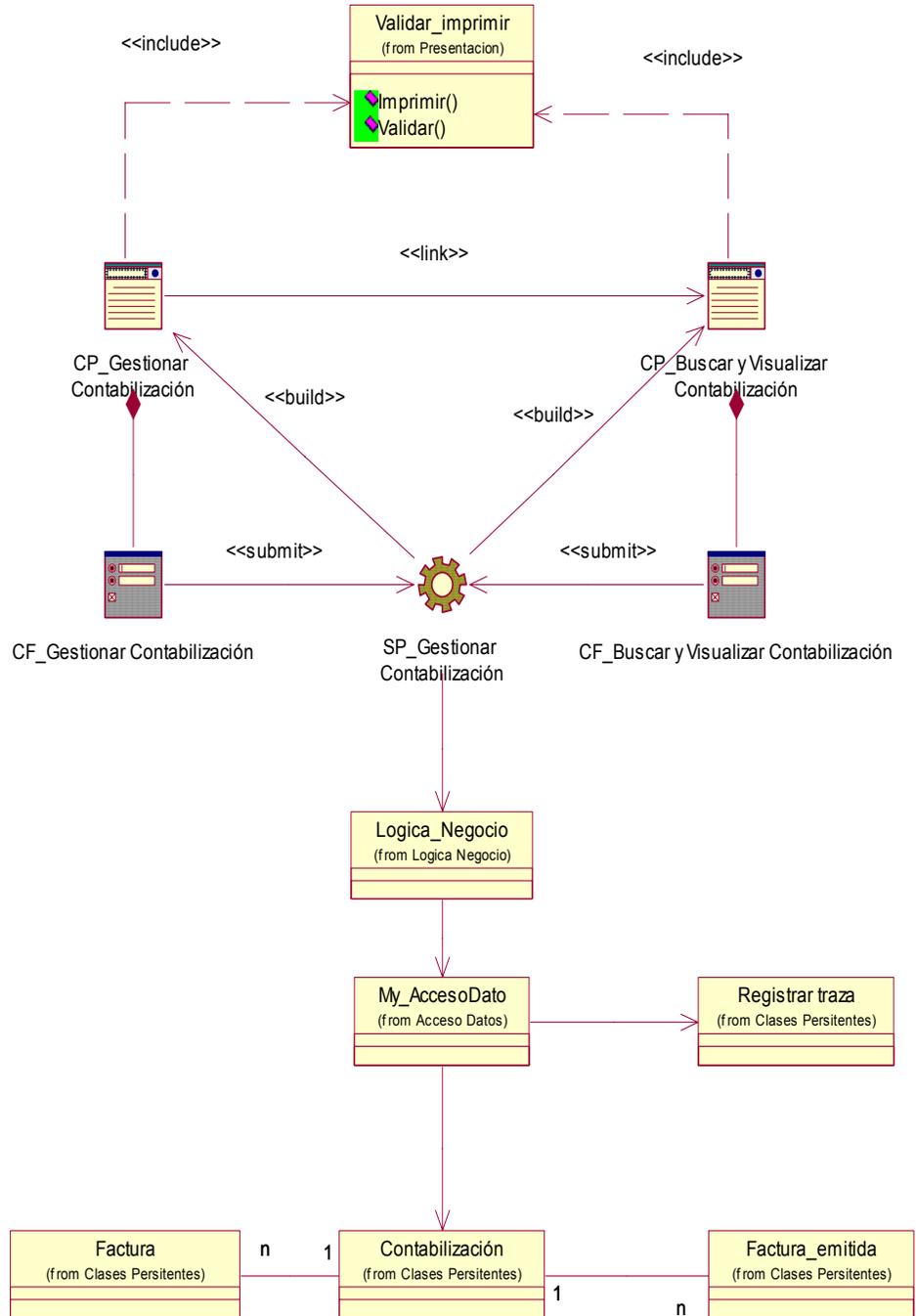
- Realizar cierre de apertura.



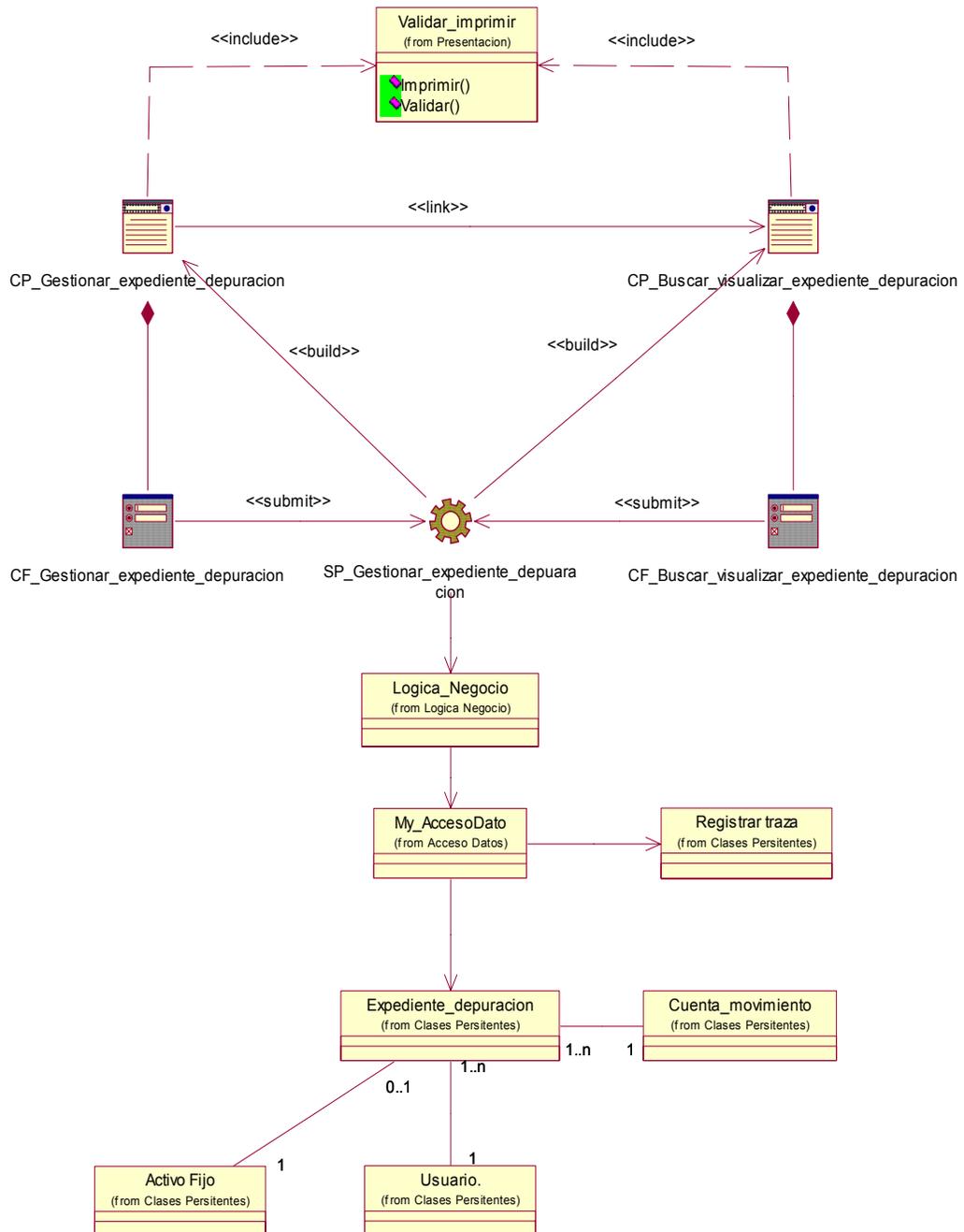
➤ *Gestionar contrato.*



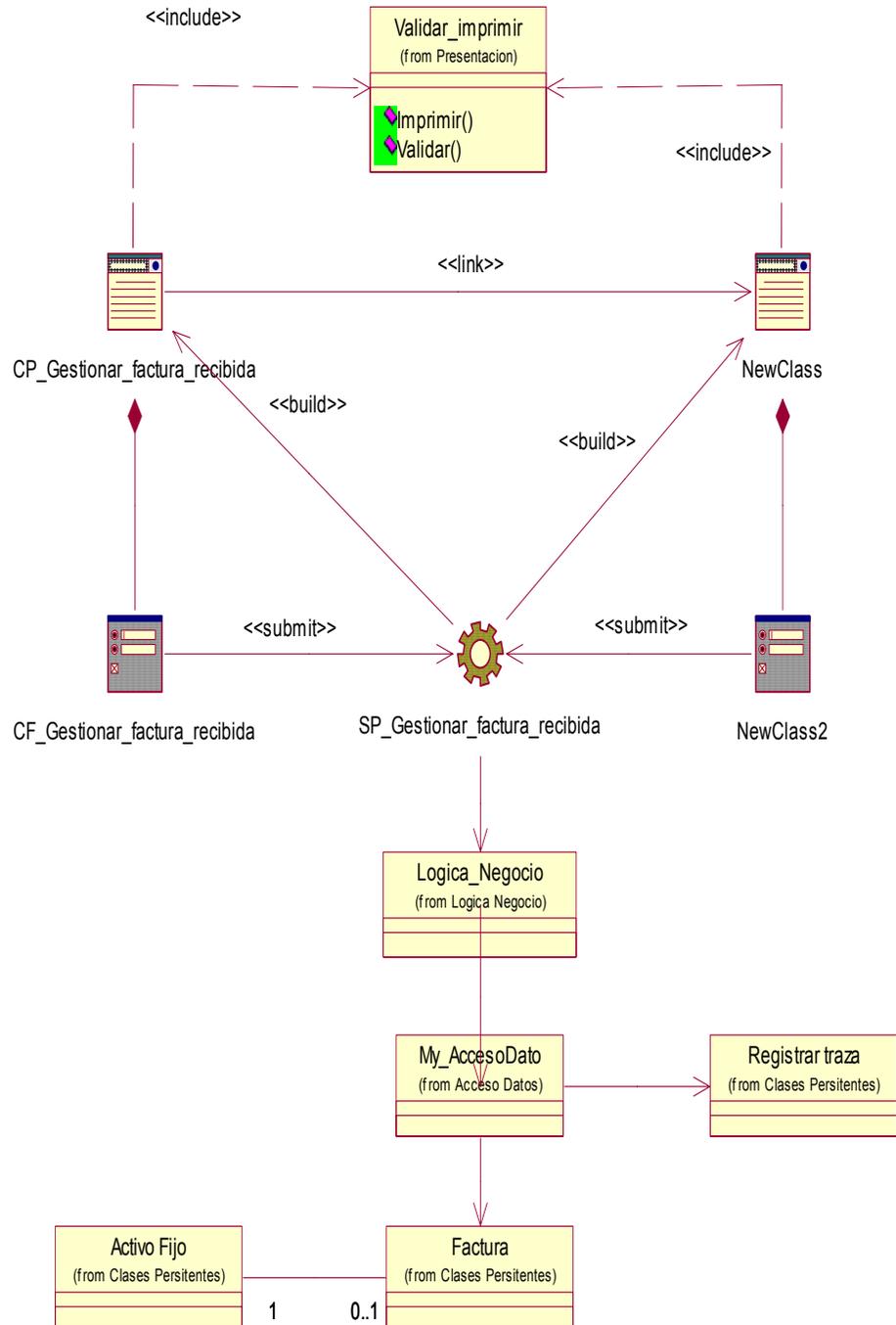
➤ *Gestionar contabilización.*



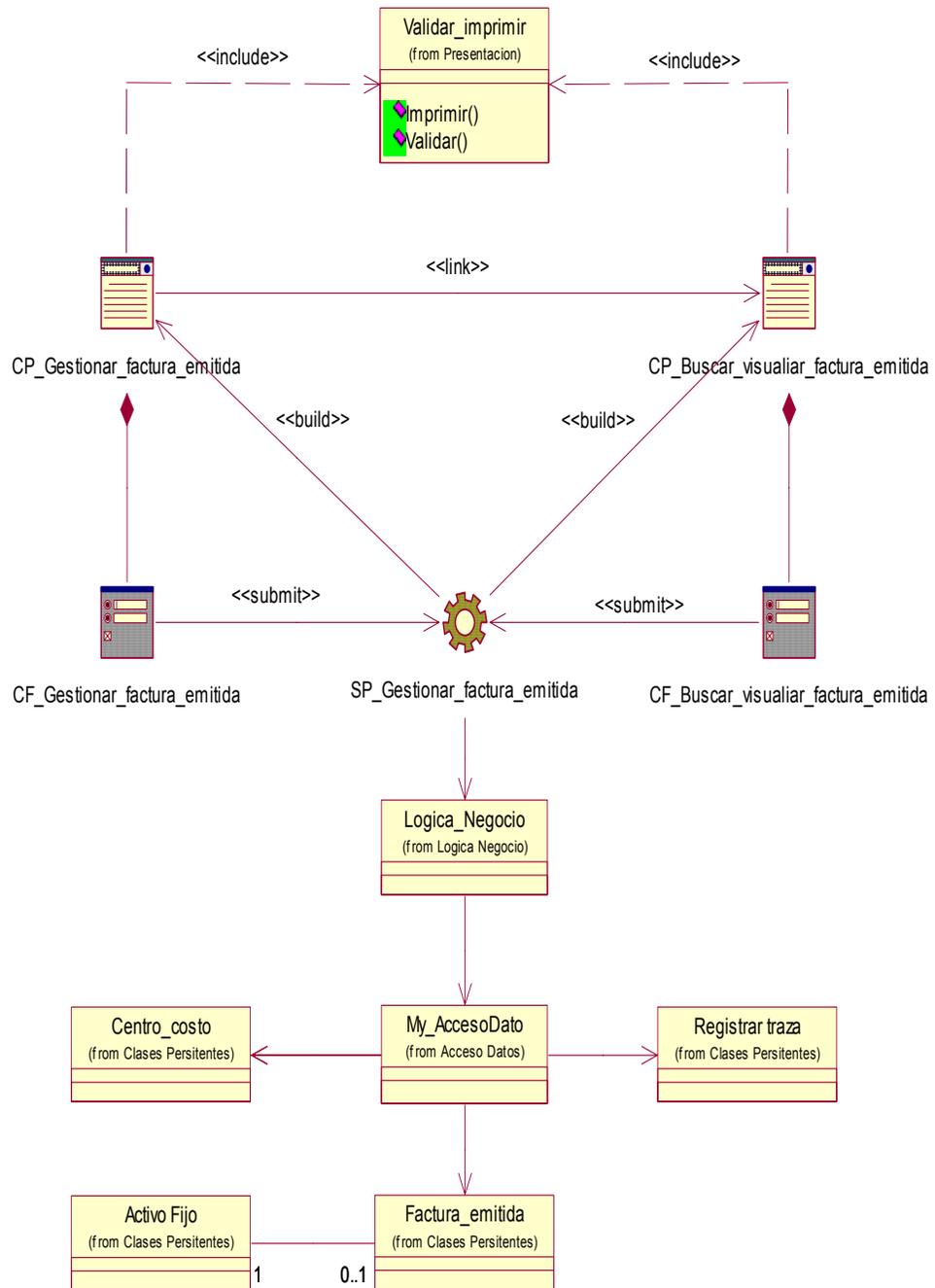
➤ *Gestionar expediente de depuración.*



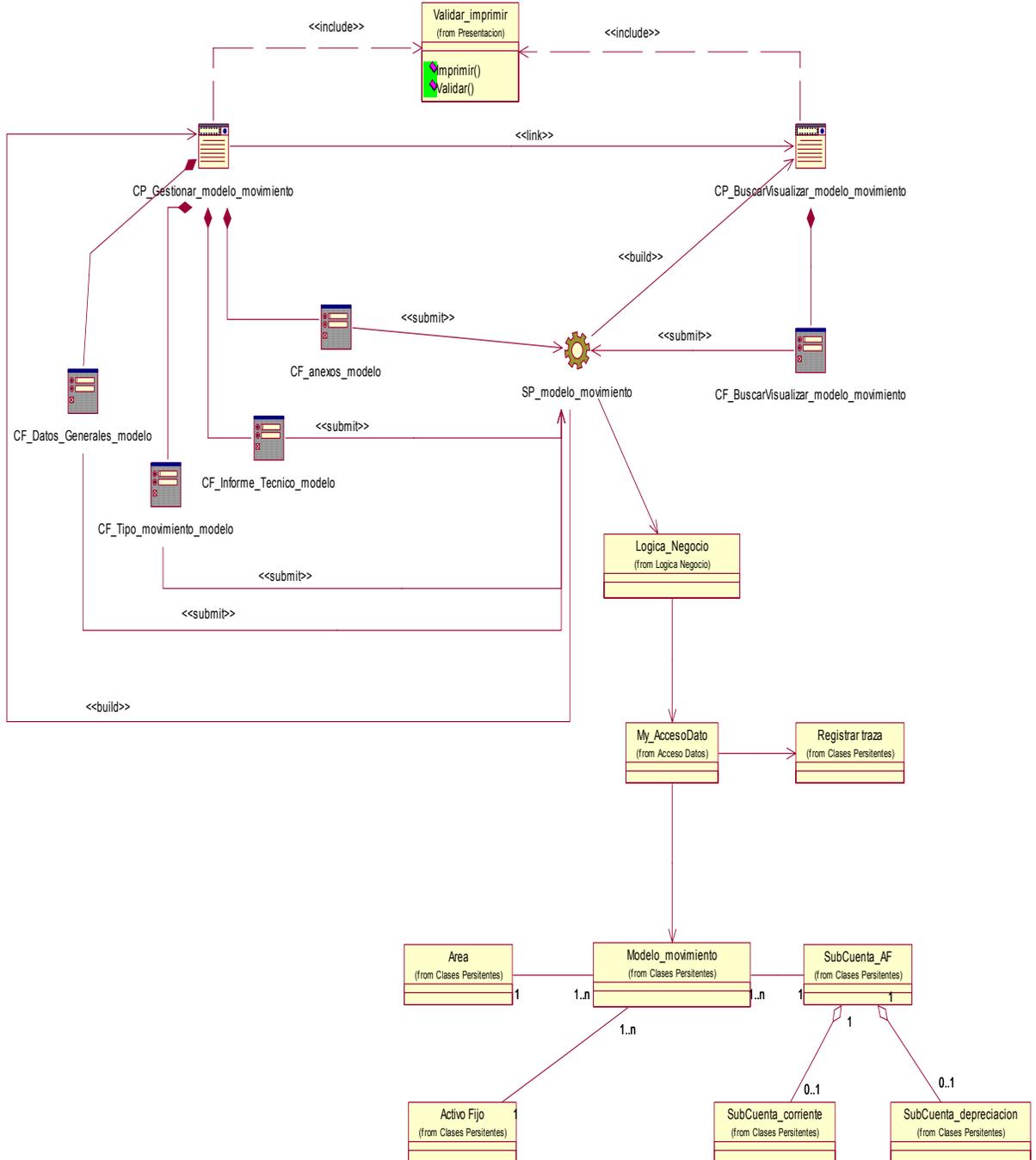
➤ *Gestionar factura recibida.*



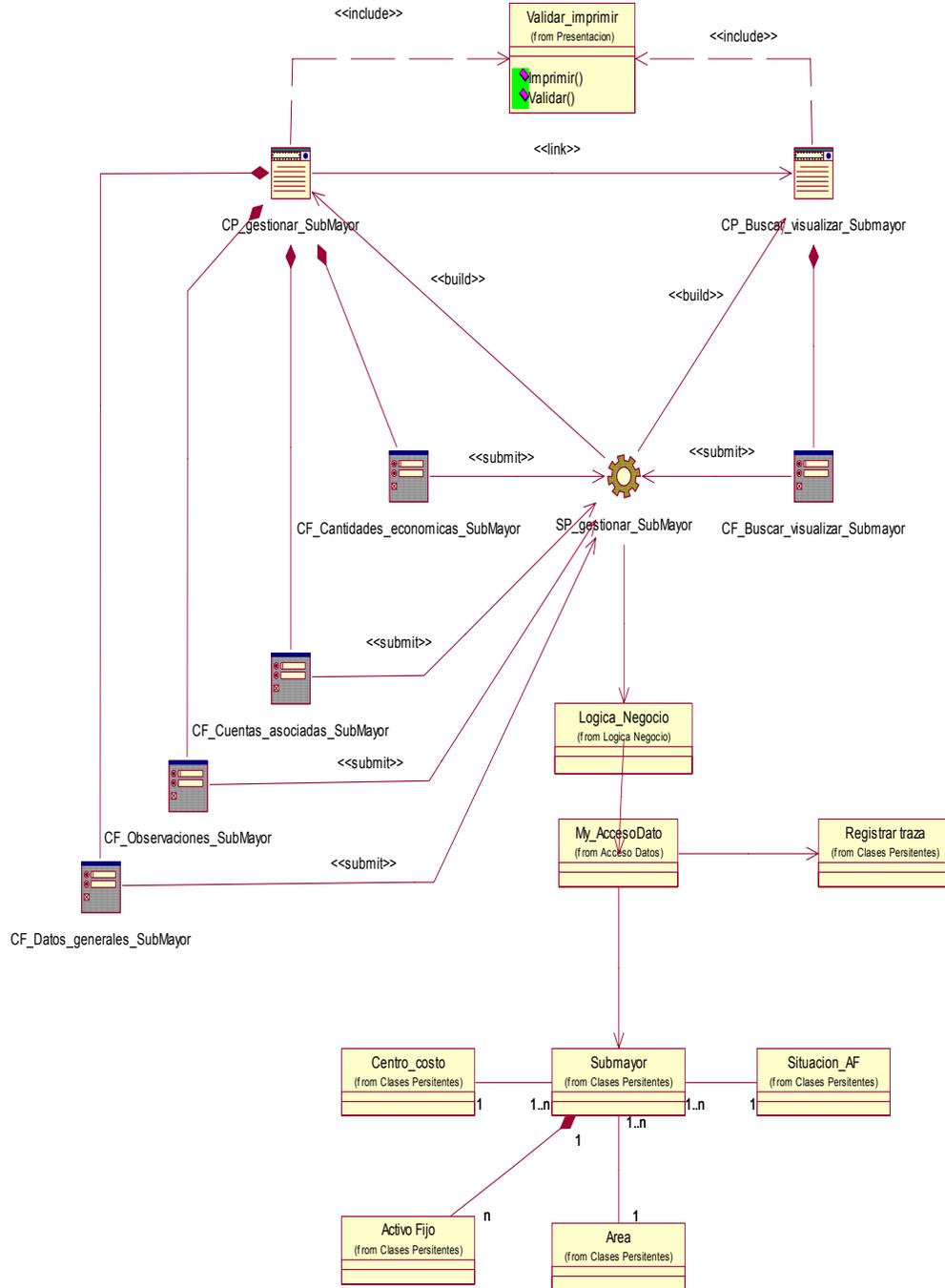
➤ *Gestionar factura emitida.*



➤ *Gestionar modelo de movimiento.*



➤ *Gestionar submayor.*



Anexo 4. Tablas del Modelo de Datos.

T_Acceso. (from S_0 ok)
Usuario : SMALLINT Cargo : SMALLINT PK Acceso : SMALLINT PK Nombre : SMALLINT
<<PK>> PK_T_Acceso.19() <<FK>> FK_T_Acceso.22() <<Index>> TC_T_Acceso.55()

T_Activo Fijo (from S_0 ok)
PK No de Inventario : SMALLINT Marca : SMALLINT Descripción : SMALLINT Modelo : SMALLINT Serie : SMALLINT Tipo : SMALLINT T_Submayor_No de inventario : SMALLINT PK No de contrato : SMALLINT PK Id Traza : SMALLINT PK Codigo : SMALLINT
<<PK>> PK_T_Activo Fijo15() <<Unique>> TC_T_Activo Fijo47() <<Unique>> TC_T_Activo Fijo4() <<FK>> FK_T_Activo Fijo18() <<FK>> FK_T_Activo Fijo1() <<FK>> FK_T_Activo Fijo0() <<Index>> TC_T_Activo Fijo46() <<Index>> TC_T_Activo Fijo3() <<Index>> TC_T_Activo Fijo1() <<FK>> FK_T_Activo Fijo52()

T_Area (from S_0 ok)
PK No de Area : SMALLINT Descripción : SMALLINT PK Codigo : SMALLINT
<<PK>> PK_T_Area5() <<FK>> FK_T_Area53()

T_Centro_costo (from S_0 ok)
PK Centro de costo : SMALLINT Descripción : SMALLINT Cuenta de gasto : SMALLINT
<<PK>> PK_T_Centro_costo6()

T_Contabilización (from S_0 ok)
Cuenta : SMALLINT Sub Cta : SMALLINT Debito : SMALLINT Credito : SMALLINT PK No de la factura : SMALLINT Valor transferido : SMALLINT No de inventario : SMALLINT PK Codigo del mov : SMALLINT
<<Unique>> TC_T_Contabilización17() <<PK>> PK_T_Contabilización17() <<FK>> FK_T_Contabilización7() <<Index>> TC_T_Contabilización16()

T_Contrato (from S_0 ok)
PK No de contrato : SMALLINT Arrendatario : SMALLINT Forma de pago : SMALLINT Tasa de interes : SMALLINT Pendiente principal : SMALLINT Acreedor : SMALLINT Elemento de Gasto : SMALLINT Sub Elemento de gasto : SMALLINT Moneda : SMALLINT PK Elemento : SMALLINT
<<PK>> PK_T_Contrato2() <<FK>> FK_T_Contrato9() <<Index>> TC_T_Contrato22()

T_Cuenta_movimiento (from S_0 ok)
PK Codigo del mov : SMALLINT Descripción : SMALLINT Codigo de la cta : SMALLINT Descripción. : SMALLINT
<<PK>> PK_T_Cuenta_movimiento7()

T_Cuenta_valor_depreciacion (from S_0 ok)	
Grupo	: SMALLINT
Descripción	: SMALLINT
Cta de activo	: SMALLINT
PK Cta depreciacion	: SMALLINT
Tasa minima	: SMALLINT
Tasa maxima	: SMALLINT
<<PK>> PK_T_Cuenta_valor_depreciacion8()	

T_ElementoGasto (from S_0 ok)	
PK Elemento	: SMALLINT
Moneda	: SMALLINT
Descripción	: SMALLINT
<<PK>> PK_T_ElementoGasto9()	

T_Expediente_depuracion (from S_0 ok)	
PK Nombre Empresa	: SMALLINT
Establecimiento	: SMALLINT
Unión	: SMALLINT
PK No de expediente	: SMALLINT
Cuenta	: SMALLINT
Fecha de cancelamiento	: SMALLINT
Concepto	: SMALLINT
Proposición	: SMALLINT
Causas de ajuste	: SMALLINT
FK Codigo del mov	: SMALLINT
FK T_Submayor_No de inventario	: SMALLINT
FK No de Inventario	: SMALLINT
FK Nombre	: SMALLINT
<<PK>> PK_T_Expediente_depuracion3() <<Unique>> TC_T_Expediente_depuracion38() <<FK>> FK_T_Expediente_depuracion21() <<FK>> FK_T_Expediente_depuracion15() <<FK>> FK_T_Expediente_depuracion6() <<Index>> TC_T_Expediente_depuracion37() <<Index>> TC_T_Expediente_depuracion14() <<Index>> TC_T_Expediente_depuracion53()	

T_Factura (from S_0 ok)	
PK No de la factura	: SMALLINT
Fecha de referencia	: SMALLINT
No de Provo Cent Cont	: SMALLINT
Tipo de moneda	: SMALLINT
Codigo de la Divisa	: SMALLINT
Rate de cambio	: SMALLINT
Codigo de la inversion	: SMALLINT
Pago anticipado	: SMALLINT
Importe de la factura	: SMALLINT
Nota de credito	: SMALLINT
Nota de debito	: SMALLINT
No del cheque	: SMALLINT
Fecha vence	: SMALLINT
PK T_Submayor_No de inventario	: SMALLINT
PK No de Inventario	: SMALLINT
PK T_Contabilización_No de la factura	: SMALLINT
<<PK>> PK_T_Factura0() <<Unique>> TC_T_Factura41() <<FK>> FK_T_Factura19() <<FK>> FK_T_Factura16() <<Index>> TC_T_Factura49() <<Index>> TC_T_Factura40()	

T_Factura_emitida (from S_0 ok)	
PK No de la factura	: SMALLINT
Cliente	: SMALLINT
Centro de costo	: SMALLINT
Contrato	: SMALLINT
Plazos de pago	: SMALLINT
Forma de pago	: SMALLINT
Cantidad	: SMALLINT
Precio	: SMALLINT
Importe	: SMALLINT
FK T_Submayor_No de inventario	: SMALLINT
FK No de Inventario	: SMALLINT
FK T_Contabilización_No de la factura	: SMALLINT
<<PK>> PK_T_Factura_emitida4() <<Unique>> TC_T_Factura_emitida44() <<FK>> FK_T_Factura_emitida20() <<FK>> FK_T_Factura_emitida17() <<Index>> TC_T_Factura_emitida43() <<Index>> TC_T_Factura_emitida51()	

T_Modelo_movimiento (from S_0 ok)	
Entidad	: SMALLINT
PK Codigo	: SMALLINT
Area	: SMALLINT
Grupo	: SMALLINT
No de documento	: SMALLINT
No de Inventario	: SMALLINT
Fecha de adquisición	: SMALLINT
Sub cuenta	: SMALLINT
Fecha de cancelado	: SMALLINT
Valor de adquisición	: SMALLINT
Tipo de movimiento	: SMALLINT
Comprobante de operaciones	: SMALLINT
Fecha de realización	: SMALLINT
Nombre del autorizado	: SMALLINT
Nombre del responsable	: SMALLINT
Cargo	: SMALLINT
Tempo de alquiler	: SMALLINT
Fecha de devolución	: SMALLINT
FK No de Area	: SMALLINT
FK T_SubCuenta_AF_Sub cuenta	: SMALLINT
FK T_Submayor_No de inventario	: SMALLINT
FK T_Activo Fijo_No de Inventario	: SMALLINT
FK Codigo del mov	: SMALLINT
<<PK>> PK_T_Modelo_movimiento40() <<FK>> FK_T_Modelo_movimiento41() <<FK>> FK_T_Modelo_movimiento27() <<FK>> FK_T_Modelo_movimiento38() <<Index>> TC_T_Modelo_movimiento98() <<Index>> TC_T_Modelo_movimiento65() <<Index>> TC_T_Modelo_movimiento90()	

T_Registrar traza (from S_0 ok)	
PK Id Traza	: SMALLINT
DModificación	: SMALLINT
Usuario	: SMALLINT
MModificación	: SMALLINT
AModificación	: SMALLINT
Doc Modificación	: SMALLINT
Campos Modificación	: SMALLINT
Datos Anteriores	: SMALLINT
<<PK>> PK_T_Registrar traza16()	



Glosario

➤ ***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 un software.

➤ ***Aplicación Web***

Sitio Web que contiene páginas con contenido sin determinar parcialmente o en su totalidad. El contenido final de estas páginas se determina sólo cuando un visitante solicita una página del servidor Web.

➤ ***Activos Fijos***

Término contable-financiero con el que se denomina a los recursos económicos bienes materiales, créditos y derechos de una persona, sociedad, corporación, entidad o empresa; son los recursos que se administran en el desarrollo de las actividades, independientemente de que sean o no propiedad de la misma empresa.

➤ ***Activo Fijo tangible***

Lo integran los bienes adquiridos, construidos o en tránsito de importación, construcción y montaje, con la intención de emplearlos en forma permanente, para la producción o suministro de otros bienes y servicios, arrendarlos o de utilizarlos en la administración del público, y que no están destinados para la venta, siempre que su vida útil probable exceda de un año.

➤ ***Activo Fijo intangible***

Lo integran las cosas que no pueden ser tocadas materialmente, tales como los derechos de patente, los de vía, el crédito mercantil, el valor de ciertas concesiones, etc.

➤ **Base de datos**

Conjunto no redundante de información almacenada en memoria organizada independientemente de su utilización y su implementación en máquinas accesibles en tiempo real y compatible con usuarios concurrentes con necesidad de información diferente y no predicable en tiempo.

➤ **Browser o Navegador**

Un **navegador Web** o **explorador Web** (del inglés, *navigator* o *browser*) es una aplicación software que permite al usuario recuperar y visualizar documentos de hipertexto, comúnmente descritos en HTML, desde servidores Web de todo el mundo a través de Intranet. Esta red de documentos es denominada World Wide Web (WWW). Cualquier navegador actual permite mostrar o ejecutar gráficos, secuencias de vídeo, sonido, animaciones y programas diversos además del texto y los hipervínculos o enlaces.

➤ **Caso de uso**

Especificación de las secuencias de acciones, incluyendo secuencias variantes y una descripción de un conjunto de secuencias de acciones, incluyendo variaciones, que un sistema lleva a cabo y que conduce a un resultado observable de interés para un actor determinado.

➤ **ConDor**

Proyecto económico contable para todas las gestiones contables de una empresa que está compuesto por 7 módulos de contabilidad, este proyecto fue desarrollado por el SICS, empresa que se dedica a realizar software para el ministerio del Transporte.

➤ **ERP** (Enterprise Resource Planning o Planeación de Recursos de la Empresa).

Un ERP es un sistema de información integral que incorpora los procesos operativos y de negocio. El propósito fundamental de un ERP es otorgar apoyo a los clientes del negocio, tiempos rápidos de respuesta a sus problemas así como un eficiente manejo de información que permita la toma oportuna de decisiones y disminución de los costos totales de operación.

➤ ***Expediente de depuración de cuentas.***

Expediente que se crea cuando se detecta un faltante o sobrante, ya sea por la realización de un inventario físico y por otras circunstancias.

➤ ***Hardware***

Componentes físicos que constituyen las Computadoras y demás dispositivos periféricos.

➤ ***Ingeniería de Software***

Es el estudio de los principios y metodologías para desarrollo de sistemas de software. Es la aplicación práctica del conocimiento científico en el diseño y construcción de programas de ordenador y la documentación adecuada para desarrollar, operar y mantenerlos.

➤ ***Lenguaje de programación***

Conjunto de normas «lingüísticas» que permiten escribir un programa y que éste sea entendido por el ordenador y pueda ser trasladado a ordenadores similares para su funcionamiento en otros sistemas. Cualquier lenguaje artificial que puede utilizarse para definir una secuencia de instrucciones para su procesamiento por un ordenador o computadora.

➤ ***Metodología***

Un sistema de principios y normas generales de organización y estructuración teórico-práctica de actividades.

➤ ***Modelo de movimientos***

Modelo en el cual se formaliza y se deja en evidencia todos los movimientos de los activos fijos, impliquen o no su movimiento físico.

➤ ***Software***

Conjunto de programas escritos para un ordenador.

Software o programa se aplica a aquellos componentes de un sistema informático que no son tangibles, es decir, que físicamente no se pueden tocar.

➤ **Sub. Mayor**

Es el documento que permite registrar los datos que corresponden a los diferentes grupos de Activos Fijos para su adecuada identificación y control, tanto físico como contable.

➤ **SGBD**

Colección de datos interrelacionados y un conjunto de programas para acceder dichos datos. Son productos de software fabricados por diferentes proveedores y con diferentes características que los diferencian unos de otros: Dbase, Fox, Access, Informix, Unify, Oracle, Interbase.

➤ **SQL**

Server Query Language. Lenguaje de consulta que permite la recuperación de información desde un servidor de bases de datos.