

005.12
San
S
TD-0179-06

TD-0179-06



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

**SISTEMA AUTOMATIZADO PARA LA CONTRATACION Y CONTROL
DEL SISTEMA DE PLANES DEL MINISTERIO DE LAS FUERZAS
ARMADAS REVOLUCIONARIAS**



**TRABAJO DE DIPLOMA PARA OPTAR POR
EL TÍTULO DE INGENIERO INFORMÁTICO**

**Autor: Franky Sánchez Barriel
TUTOR: ING. RAÚL TRIANA DÍAZ**

CUJAE, JUNIO DEL 2006

INDICE	
INTRODUCCION	1
Capítulo I Fundamentación Teórica	6
1.1 Objeto de estudio	6
1.1.1 Objetivos estratégicos de la organización	6
1.1.2 Flujo actual de los procesos	8
1.1.3 Análisis crítico de la ejecución de los procesos	10
1.2 Proceso de Objeto de Automatización	10
1.3 Fundamentación de los Objetivos	11
1.4 Conclusiones	13
Capítulo II Tecnologías actuales a considerar	15
2.1 Tendencias y Tecnologías Actuales	15
2.1.1 Introducción a Internet	15
2.1.2 La WWW como servicio de Internet	17
2.1.3 Historia de las aplicaciones Web	17
2.2 Lenguajes de programación WEB	19
2.2.1 Introducción	19
2.2.2 ¿Por qué PHP?	19
2.3 Sistemas Gestores de Bases de Datos	20
2.3.1 Introducción	20
2.3.2 ¿Por qué PostgreSQL?	21
2.4 Proceso de desarrollo de Software	23
2.4.1 ¿Qué metodología debo usar para el desarrollo de un Software?	23
2.4.2 ¿Qué es UML?	27
2.4.3 Qué son las Herramientas CASE	27
2.4.4 Porqué deberíamos usar herramientas CASE de modelado con UML	28
2.4.5 ¿Por qué Visual Paradigm como herramienta CASE, UML como lenguaje de modelado y RUP como metodología?	29
2.5 Otras Herramientas	30
2.6 Conclusiones	32
Capítulo III Descripción de la Solución Propuesta	34
3.1 Modelamiento del Negocio	34
3.1.1 Modelo de casos de uso	35
3.1.2 Especificaciones de los casos de uso	35
3.1.3 Actores y trabajadores del negocio	39
3.1.4 Modelo de Objetos	40
3.1.5 Diagrama de Actividades	41
3.2 Captura de Requisitos	42
3.3.3 Requisitos Funcionales	42
3.3.4 Requisitos No Funcionales	44
3.3 Modelado del Sistema	48
3.3.1 Actores del Sistema para automatización	49
3.3.2 Modelado de casos de uso del sistema	49
3.3.3 Especificaciones de los casos de usos	50
3.3.4 Expansión de los casos de Uso	50

3.4 Conclusiones	63
Capítulo IV Construcción de la solución propuesta	65
4.1 Diseño	65
4.1.1 Arquitectura	66
4.1.2 ¿Por qué patrón Capas?	68
4.1.3 Descripción de las Capas	69
4.1.4 Patrones de Diseño	74
4.1.5 Mecanismo de Diseño de Acceso a Datos	75
4.1.6 Mecanismo de Diseño de Acceso Seguridad	78
4.1.7 Diagramas de Clases del Diseño	79
4.1.8 Principios de Diseño	83
4.1.9 Estándares	85
4.2 Modelo Lógico y Físico de los Datos	93
4.3 Modelo de Despliegue.....	96
4.4 Modelo de Implementación.....	97
4.5 Conclusiones	99
Capítulo V Estudio de Factibilidad.....	101
5.1 Planificación	101
5.2 Planificación basada en casos de uso	105
5.2.1 Calcular el factor de peso de los actores sin ajustar (UAW).....	106
5.2.2 Calcular el factor de peso de los casos de uso sin ajustar (UUCW).....	106
5.2.3 Calcular el número de puntos de casos de uso no ajustados (UUCP) ...	107
5.2.4 Calcular factor que toma en cuenta complejidad del proyecto (TCF)	107
5.2.5 Calcular el factor que toma en cuenta el ambiente del proyecto (EF)	108
5.2.6 Calcular Puntos de casos de uso ajustados (UCP)	108
5.2.7 Determinar el Factor de conversión (CF)	109
5.2.8 Estimación de esfuerzo a través de los puntos de casos de uso (E).....	109
5.2.9 Calcular esfuerzo de todo el proyecto.	109
5.2.10 Análisis de esfuerzo	109
5.2.11 Análisis de costo del proyecto.	110
5.3 Beneficios Tangibles e Intangibles.....	110
5.4 Análisis de costos y beneficios.	111
5.5 Conclusiones	113
CONCLUSIONES	115
RECOMENDACIONES	117
REFERENCIAS BIBLIOGRAFICAS	119
GLOSARIO DE TERMINOS	123
ANEXOS	128
ANEXO I.....	128
ANEXO II.....	128
ANEXO III Descripción de alto nivel de casos uso.....	129
ANEXO IV Descripción de casos de uso expandidos	129
ANEXO V Diagrama de clases del diseño.	138
ANEXO VI.	140

INDICE DE FIGURAS

Fig. 3.1 Diagrama de casos de uso del negocio	35
Fig. 3.2 Modelo de objetos del negocio	40
Fig. 3.3 Diagrama de actividades CU Control y Ejecución de la Contratación	41
Fig. 3.4 Diagrama de casos de uso del sistema.....	49
Fig. 4.1 Diagrama de paquetes de la arquitectura	68
Fig. 4.2 Diagrama de subsistemas de Capa Presentación.....	69
Fig. 4.3 Diagrama de subsistemas del Paquete Dominio.....	70
Fig. 4.4 Diagrama de clases Subsistema Seguridad	71
Fig. 4.5 Diagrama de los subsistemas del Paquete Servicios.....	71
Fig. 4.6 Diagrama de clases Librería PHP	72
Fig. 4.7 Diagrama de clases Subsistema Acceso a Datos	73
Fig. 4.8 Vista estática del mecanismo de diseño para persistencia. Conectividad usando entorno desarrollo brindado por el lenguaje PHP.....	77
Fig. 4.9 Vista estática del mecanismo de seguridad usando entorno de desarrollo PHP.....	79
Fig. 4.10 Diagrama CU Actualizar 711.....	81
Fig. 4.11 Diagrama CU Actualizar Contratos.....	82
Fig. 4.12 Diagrama CU Buscar Documento.....	83
Fig. 4.13 Diagrama de clases persistentes.....	94
Fig. 4.14 Diagrama entidad-relación	95
Fig. 4.15 Diagrama de despliegue	97
Fig. 4.16 Diagrama de componentes	98
Fig. 5.1 Diagrama de GANTT	102
Fig. 5.2 Tabla de duración de tareas.....	103
Fig. 5.3 Estructura organizativa del equipo de desarrollo	104

INDICE DE TABLAS

Tabla 1	Etapas principales del proceso de planificación dentro del MINFAR ...	7
Tabla 2	Descripción del caso de uso del negocio Control y Ejecución de la Contratación	38
Tabla 3	Descripción de los actores del negocio.....	39
Tabla 4	Descripción de los trabajadores del negocio	40
Tabla 5	Descripción de los actores del sistema	49
Tabla 6	Descripción del caso de uso Actualizar 711	57
Tabla 7	Descripción del caso de uso Actualizar Contratos	61
Tabla 8	Descripción del caso Buscar Documento.....	63
Tabla 9	Estándar de codificación.....	90
Tabla 10	Estándar de diseño de la Base Datos	93
Tabla 11	Relación de órganos que participan en los procesos de planificación, aprobación y contratación y control.....	128

INTRODUCCION

Internet constituye uno de los fenómenos que más asombra al hombre promedio hoy en día. Su desarrollo en progresión geométrica, como es lógico, impresiona y desconcierta a muchos. Y no es menos cierto, que la industria de la computación y las telecomunicaciones experimentan un avance tan acelerado que pocas disciplinas pudieran igualárseles, pero también no es menos cierto, que fueron muchas las décadas en las que el hombre puso todo su empeño por obtener y perfeccionar los primeros componentes y dispositivos. Antes de los primeros logros, tuvieron que recorrer un largo camino las matemáticas y la física; después concurren la electrónica, la computación y muchas otras disciplinas.

Sin embargo, no es precisamente el desarrollo acelerado de esta gran red la causa del asombro principal, sino su poder de penetración en la vida humana, su capacidad de revolucionar desde el puesto de trabajo hasta las relaciones sociales: su capacidad de mediar en casi todos los actos de la vida moderna. Con la introducción de Internet y del Web en concreto, se han abierto infinitas posibilidades en cuanto al acceso a la información desde casi cualquier sitio. Esto representa un desafío a los desarrolladores de aplicaciones, ya que los avances en tecnología demandan cada vez aplicaciones más rápidas, ligeras y robustas que permitan utilizar el Web.

Obteniendo el máximo provecho de las capacidades ofrecidas por los nuevos avances y con la flexibilidad requerida por los sistemas de información, la evolución de aplicaciones Web empresariales hacia otros canales aumenta su alcance e impacto, aumentando su aportación de valor a las empresas

Primero fue Internet, más tarde aparecieron las Intranets, llevando todos los beneficios de la red al terreno de las empresas y organizaciones.

Este desarrollo en el mundo de las comunicaciones son una de las tecnologías más importantes para el desarrollo de las empresas al igual que en la década de los años setenta los grandes ordenadores centrales fueron los que soportaron el

INTRODUCCION

crecimiento comercial de las empresas y en los ochenta los ordenadores personales fueron los que automatizaron las tareas de las oficinas.

El uso creciente de Internet como instrumento de comunicación y de distribución de la información ha propiciado que las empresas se hayan planteado la posibilidad de utilizar los mismos servidores Web para difundir la información interna a través de las Intranets y para gestionar grandes volúmenes de datos.

Intranet es la aplicación de los estándares Internet dentro de un ámbito corporativo para mejorar la productividad, reducir costes y mantener los sistemas de información existentes.

Es una forma de poner al alcance de los trabajadores todo el potencial de la empresa, para resolver problemas, mejorar los procesos, construir nuevos recursos o mejorar los ya existentes, divulgar información de manera rápida y convertir a estos trabajadores en miembros activos de una red corporativa, o sea da al usuario la información que este necesita para su trabajo. Pretende que cada cual tenga la información necesaria en el momento oportuno sin que tenga que recurrir a terceros para conseguirla.

Estos avances tecnológicos continúan cambiando el estilo de vida de los consumidores, desde la manera de adquirir productos y servicios, hasta la manera de comunicárselo a la empresa. Por ello, las empresas que no adopten la tecnología como una parte importante de su estrategia y toma de decisiones, estarán fuera de la economía. En este sentido, los ERP (Enterprise Resource Planning, Planeamiento de Recursos para las Empresas) aparecen como una herramienta fundamental tanto para la integración de las diferentes áreas y sistemas de la empresa, como para la ayuda en la toma de decisiones estratégicas.

Son sistemas transaccionales, es decir, están diseñados para trabajar con procesos de la empresa, soportarlos, procesar los datos y obtener de ellos información específica. Así, puede haber un seguimiento y control de los procesos del negocio, como son: finanzas y contabilidad, ventas, compras, manufactura, logística, recursos humanos o mercadotecnia.

INTRODUCCION

El ERP gestiona de manera integrada y eficiente la información de la empresa, *comunicando las diferentes áreas del negocio mediante procesos electrónicos*. La función principal es organizar y estandarizar procesos y datos internos de la empresa, transformándolos en información útil para ser analizados para la toma de decisiones. Es importante recordar que finalmente, aunque estos sistemas apoyan en la toma de decisiones, no se quiere decir que ellos lo hagan, sino que los administradores (humanos) tienen el poder final para tomar las decisiones estratégicas y adecuadas en la empresa.

Como todos sabemos la actividad humana, cuando se realiza de una manera consciente, es casi siempre finalista. Es extraño que se cometa una determinada tarea con empeño y responsabilidad sin que se tenga presente una meta última propuesta con la suficiente antelación, ya que la racionalidad nos lleva a preparar de manera adecuada las actuaciones. Éste es, a nuestro entender, el sentido más profundo de la acción planificadora: *prever para acertar, disponer para evitar la confusión y el desorden, proyectar para resolver con desenvoltura y eficacia*.

Las aseveraciones, que vendrían a ser tres reglas de oro de un buen organizador, nos hacen reflexionar en la importancia que adquiere la planificación en cualquier ámbito de la vida cotidiana.

La actualidad reinante ha demostrado que una empresa u organización debe planificarse para poder funcionar bien.

Debido a este amplio desarrollo, todas las empresas a nivel internacional han incorporado las Intranets y los ERP como una alternativa para la gestión de su información vinculada a todos los procesos del negocio, dándole a las mismas fortaleza y supervivencia en un mundo tan competitivo.

Actualmente el MINFAR dado el proceso de informatización por el que está transcurriendo, cuenta con el soporte tecnológico, tanto desde el punto de vista de soporte de comunicaciones y capacidad de almacenamiento y procesamientos de información que le permiten proyectar la integración de la

INTRODUCCION

informativa de los diferentes procesos que componen el sistema de planificación, registro y control de los recursos.

En este trabajo nos concentremos y profundizaremos en el desarrollo de un sistema informático que abarca el *proceso de Control y Ejecución de la contratación* dentro del MINFAR.

El presente trabajo consta de introducción, cinco capítulos, conclusiones, recomendaciones, referencias bibliográficas y anexos:

En el capítulo 1, **Fundamentación Teórica**, se exponen los principales aspectos sobre el proceso de contratación y su estado actual dentro de la institución, así como se abordan sobre los puntos clave de la aplicación de la metodología de la investigación.

En el capítulo 2, **Tecnologías actuales a considerar**, se abordan los principales aspectos de la tecnología Web, así como las tecnologías fundamentales que sustentan el desarrollo de sistemas informáticos asociadas a la misma.

El capítulo 3, **Descripción de la solución propuesta**, se enfoca en los aspectos fundamentales de modelamiento del negocio, captura de requisitos y análisis del sistema.

En el capítulo 4, **Construcción de la solución propuesta**, centramos nuestra línea en los principales aspectos sobre la arquitectura del sistema, patrones, así como algunos mecanismos de diseño definidos para mejorar y optimizar el desarrollo e implementación del mismo.

En el capítulo 5, **Estudio de Factibilidad**, se explica como se gestionó y organizó la ejecución de nuestro sistema, así como la repercusión de los costos para desarrollarlo y el estimado de tiempo para llevar a cabo esta tarea.



Fundamentación Teórica

En este capítulo se abordan las principales actividades y conceptos involucrados dentro del proceso de contratación dentro del Ministerio de las Fuerzas Armadas.

Además se describen los procesos fundamentales del negocio que se relacionan con el objeto de estudio de este trabajo, así como se mencionan los principales problemas que fundamentan la propuesta de solución, y se marcan los objetivos generales y específicos. Todo lo anteriormente mencionado fomentará la base teórica principal de este trabajo.

1.1. Objeto de estudio

1.1.1 Objetivos estratégicos de la organización

Las Fuerzas Armadas Revolucionarias (FAR) cuyo objeto social es la dirección de la preparación del país para la lucha armada y para ello cuenta con una estructura y composición que le permiten el cumplimiento de las misiones planteadas, por nuestro partido y gobierno y que de forma general están refrendada en la ley 76 de la Defensa de nuestra constitución.

El MINFAR como órgano central de las FAR para asegurar el cumplimiento de sus funciones, planifica anualmente los objetivos generales a alcanzar a partir de los cuales cada uno de los órganos de dirección que componen al aparato central del Ministerio elabora sus objetivos particulares, los cuales para su

cumplimiento necesitan determinados niveles de aseguramientos materiales o financieros los cuales son planificados, aprobados, contratados y adquiridos.

El proceso de planificación del MINFAR se enmarca dentro del proceso de planificación del país como un organismo más de la administración central del estado (OACE) donde el Ministerio de Economía y Planificación actúa como órgano rector de dicha actividad. Constituyen **“Bases para el proceso de planificación del MINFAR”** las Directivas de Gobierno con relación a la Preparación del País para la Guerra aprobadas para el año X, las emitidas por nuestro organismo, relacionadas con la Preparación del País para la Defensa en el año X, así como las normas, regulaciones y calendarios que emiten los ministerios de Economía y Planificación, de Finanzas y Precios, de Trabajo y Seguridad Social y el Banco Central de Cuba teniendo en cuenta la búsqueda de la integridad del Plan Material y el Presupuesto.

Como podemos ver el MINFAR ejecuta de forma genérica las mismas actividades que cualquier OACE del país durante el proceso de planificación y contratación de los recursos materiales y financieros y para ello organiza dicho proceso por etapas de la forma siguiente:

ETAPAS	ACTIVIDADES
PRIMERA	Elaboración y aprobación de los objetivos de trabajo para la Planificación del MINFAR.
SEGUNDA	Elaboración de las propuestas del Sistema de Planes del MINFAR.
TERCERA	Aprobación del Sistema de Planes del MINFAR a presentar al Gobierno.
CUARTA	Evaluación integral del proceso de planificación y determinación de las insatisfacciones y vías de solución del aseguramiento al Plan. Ajuste de los niveles de actividad en correspondencia con los aseguramientos materiales y financieros que finalmente se aprueben.
QUINTA	Contratación y Control de la ejecución del Sistema de Planes del MINFAR.

Tabla 1. Etapas principales del proceso de planificación dentro del MINFAR

Nosotros nos centraremos en este trabajo en la quinta etapa dentro del proceso de planificación.

1.1.2 Flujo actual de los procesos

El proceso de contratación dentro del MINFAR revista una gran importancia ya que el mismo durante su ejecución se va asegurando las principales tareas y asegurando los objetivos fundamentales y específicos de las distintas áreas de la organización, en cuanto a medios e insumos. El mismo comienza cuando el Viceministro primero certifica el presupuesto asignado por el Ministerio de Economía y Planificación a las FAR para asegurar sus tareas principales. En este proceso el viceministro en cuanto a la solicitud hecha por el MINFAR y la enviada por el Ministerio de Economía, la reestructura y se lo informa a todos los órganos consumidores, órganos abastecedores centrales, órganos consumidores autorizados, los centros de balance y otros. Una vez recibidos estos informes comienza la distribución por los distintos especialistas de las áreas para la elaboración de los respectivos contratos SCM-711 que recogen un conjunto de medios e insumos para la respectiva contratación por la Empresa Importadora. Una vez contratados comienza la ejecución de estos contratos.

Control de la ejecución del Sistema de Planes que el MINFAR presenta a la Economía.

El control de la ejecución del Sistema de Planes que el MINFAR presenta a la Economía, se ejerce en el **ámbito externo** a través de los despachos periódicos que se realizan con los ministerios de Economía y Planificación, de Finanzas y Precios así como el resto de los OACE que balancean determinados medios o servicios presentes en el sistemas de planes del MINFAR.

En el **ámbito interno**: los despachos del Viceministro Primero con los jefes de los órganos de dirección del aparato central del MINFAR implicados en el proceso, los jefes de los diferentes mandos e instituciones de subordinación directa al Ministerio, Sistema Empresarial Militar, el Grupo Empresarial

GEOCUBA y la Empresa Importadora en lo referente a la producción, los servicios, los abastecimientos y las inversiones, lo cual está normado en las indicaciones No 32. del Ministro de las FAR y la Orden No. 54 del Viceministro Primero.

El control de la ejecución del Presupuesto del MINFAR, se ejerce mediante los sistemas de contabilidad material y financiera de la actividad presupuestada de las FAR.

Los mandos y unidades comunican, en correspondencia con el Sistema Informativo de la ejecución de los recursos financieros y de los gastos del Presupuesto y de los pagos a la Dirección de Economía; la que evalúa e informa mensualmente al Ministerio de Finanzas y Precios. Asimismo los mandos y unidades informan mensualmente la ejecución de los pagos, en correspondencia con el Sistema Informativo de la Ejecución Financiera, a la Dirección de Economía quien lo evalúa y comunica al Ministerio de Finanzas y Precios.

Los órganos abastecedores centrales y consumidores autorizados al cierre de cada semestre, informan la ejecución del presupuesto que les fue aprobado; así como el estado de las cuentas por cobrar y pagar y los daños materiales y perjuicios económicos causados a la propiedad estatal.

El control de la ejecución del Plan en Divisas para Operaciones no Comerciales y Transportaciones Aéreas Internacionales se ejerce de acuerdo a lo establecido en los documentos rectores que norman el proceso de ejecución, control e información de las operaciones no comerciales en divisas.

El control de estos Planes no sólo incluye el conocimiento del desarrollo de las actividades a partir de informes periódicos, sino que además comprende actividades de verificación y validación de la información, en interés de asegurar el cumplimiento de los objetivos planificados para el año y las medidas para garantizar dichos objetivos.

Durante este tiempo el Centro de Balance ejerce el control de la ejecución del proceso de contratación a partir de la información que le brinda la Empresa Importadora, el almacén donde se reciben los medios que se adquieren a partir de la ejecución de determinado contrato que responde a un (SCM-117) que está.

conformado por un conjunto de productos que pertenecen a una actividad de un objetivo.

1.1.3 Análisis crítico de la ejecución de los procesos

Este conjunto de procesos y actividades relacionadas con el proceso de planificación en su etapa de Contratación y Control de la Ejecución del Sistema de Planes del MINFAR se realiza de forma manual o con herramientas informáticas muy rudimentarias, por lo que la gestión de la información correspondiente a estos procesos no es la mejor y dada las demandas de hoy en día en cuanto a recuperación de la información, no se dispone de la misma cuando se requiere. Además el medio de envío y recepción de los modelos, informes y contratos varía constantemente, señalando que actualmente para los trabajadores enrolados en estas actividades se les presenta la dificultad de disponer en cualquier momento que sea necesario la información del estado de los contratos en cuanto a su ejecución, imposibilitando un buen proceso de toma de decisiones tanto para los mismos como para los jefes superiores. Por otra parte la Empresa Importadora cuenta con su propia solución informática diseñada para resolver sus problemas informativos que no son los mismos de los Centros de Balance, por lo que se necesita tener un mecanismo automatizado, estandarizado e interoperable de registro y control del estado de los contratos, así como el estado de su ejecución.

1.2 Proceso de Objeto de Automatización

Del análisis hecho a los Centros de Balance en cuanto a su situación actual, formas de gestión de la información correspondientes al proceso de contratación y las expectativas de los usuarios en cuanto mejoras del proceso en cuestión se identifico el siguiente **problema**: ¿Cómo favorecer la gestión de la información dentro del proceso de Contratación y Control de la ejecución del Sistema de Planes del MINFAR?

Dicho problema se enmarca dentro del **objeto de estudio**: proceso de Contratación y Control de la ejecución del Sistema de Planes del MINFAR.

1.3 Fundamentación de los Objetivos

Con el propósito de resolver el problema antes planteado se trazó el siguiente **objetivo**: diseñar e implementar un sistema informático que favorezca el proceso de Contratación y Control de la ejecución del Sistema de Planes del MINFAR.

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

- ✓ Estudio sobre el proceso Contratación y Control de la Ejecución del Sistema de Planes del MINFAR.
- ✓ Investigación sobre los principales sistemas existentes que abordan el proceso.
- ✓ Modelar los principales procesos del negocio y analizar la posibilidad de proponer mejoras a los mismos.
- ✓ Realizar el análisis del sistema.
- ✓ Diseñar el sistema.
- ✓ Realizar la implementación de las acciones principales del registro y actualización de los documentos principales dentro del proceso: SCM711 y los contratos.

El objetivo delimita el **campo de acción**: automatización del proceso de Contratación y Control de la ejecución del Sistema de Planes del MINFAR.

Para poder orientarnos dentro de esta investigación me planteo la siguiente **hipótesis**: si se desarrolla un sistema informático consistente y multiplataforma se favorecerá el *proceso de Contratación y Control de la ejecución del Sistema de Planes del MINFAR*.

Esta investigación estará guiada por las siguientes **tareas**:

- ✓ Elaboración de una base de conocimientos teóricos necesarios sobre el proceso de Contratación y Control de la ejecución del Sistema de Planes del MINFAR.
- ✓ Análisis del estado de la informatización dentro de las áreas vinculadas al proceso anteriormente mencionado.
- ✓ Análisis y valoración sobre el nivel de conocimiento de los usuarios en torno a la Informática y habilidades de trabajo con sistemas informáticos con características de ERP.
- ✓ Estudio sobre sistemas existentes que trabajan el proceso de contratación y estado de la misma; así como la filosofía de interacción con los usuarios y las principales heurísticas.
- ✓ Desarrollar el análisis, diseño e implementación de las tareas fundamentales vinculadas al registro y actualización en la gestión de los SCM711 y los contratos dentro del proceso de Contratación y Control de la ejecución del Sistema de Planes del MINFAR

Los métodos utilizados en este proceso de investigación son los siguientes:

- ✓ *Entrevistas* a los trabajadores de las áreas vinculadas al proceso de contratación y a los encargados del efectuar el proceso de Contratación y Control de la ejecución del Sistema de Planes del MINFAR
- ✓ *Análisis de documentos* con el objetivo de determinar los puntos esenciales del proceso.
- ✓ *Método Histórico* para determinar si actualmente están desarrollados sistemas informáticos que gestionan esta información y así determinar filosofías y heurísticas de trabajo e interacción con los usuarios potenciales.
- ✓ El *Método Causal* para determinar y analizar los factores que provocaron la necesidad de desarrollo de un sistema informático con estas características.

1.4 Conclusiones

Como se ha podido ver en este capítulo se han abordado los puntos principales del proceso de Contratación y Control de la ejecución del Sistema de Planes del MINFAR. Además se hizo énfasis en la forma en que se desarrolla actualmente este proceso, analizando su flujo, así como sus principales puntos críticos en cuanto a la ejecución del mismo. Además al concluir esta investigación se determinó algo muy esencial: llevar a cabo el desarrollo de un sistema informático para el Ministerio de las Fuerzas Armadas Revolucionarias que elevará la gestión dentro del proceso de Contratación y Control de la ejecución del Sistema de Planes del MINFAR.

A graphic element consisting of a dark gray square. At the top, the word "CAPÍTULO" is written in a white, serif, all-caps font. Below it, the Roman numeral "II" is written in a large, white, serif font.

Tecnologías actuales a considerar

En este capítulo se abordan las principales tecnologías que pueden sustentar el desarrollo de nuestro sistema, haciendo énfasis en las conocidas como código abierto (Open Source) y software libre, pero principalmente siguiendo la línea de utilización de herramientas e implementación de un sistema multiplataforma.

Además se aborda el tema de la metodología de desarrollo a seleccionar para llevar a cabo el proceso de análisis, diseño e implementación del sistema a desarrollar, dada la importancia que reviste la aplicación de la misma en la obtención de un producto software de calidad. También, se proponen las principales herramientas a utilizar para el desarrollo de aplicaciones con características similares a la nuestra, teniendo en cuenta los resultados obtenidos de la comparación entre las mismas.

2.1 Tendencias y Tecnologías Actuales

2.1.1. Introducción a Internet

Internet, la red de redes, nace a mediados de la década de los setenta, bajo los auspicios de DARPA, la Agencia de Proyectos Avanzados para la Defensa de Estados Unidos. DARPA inició un programa de investigación de técnicas y tecnologías para unir diversas redes de conmutación de paquetes, permitiendo así a los ordenadores conectados a estas redes comunicarse entre sí de forma

fácil y transparente.

De estos proyectos nació un protocolo de comunicaciones de datos, IP o Internet Protocol, que permitía a ordenadores diversos comunicarse a través de una red, Internet, formada por la interconexión de diversas redes.

A mediados de los ochenta la Fundación Nacional para la Ciencia norteamericana, la NSF, creó una red, la NSFNET, que se convirtió en el *backbone* (el troncal) de Internet junto con otras redes similares creadas por la NASA (NSINet) y el U.S. DoE (Departamento de Energía) con la ESNET. En Europa, la mayoría de países disponían de troncales nacionales (NORDUNET, RedIRIS, SWITCH, etc.) y de una serie de iniciativas paneuropeas (EARN y RARE). En esta época aparecen los primeros proveedores de acceso a Internet privados que ofrecen acceso pagado a Internet.

A partir de esta época, gracias entre otras cosas a la amplia disponibilidad de implementaciones de la *suite* de protocolos TCP/IP (formada por todos los protocolos de Internet y no sólo por TCP e IP), algunas de las cuales eran ya de código libre, Internet empezó lo que posteriormente se convertiría en una de sus características fundamentales, un ritmo de crecimiento exponencial.

A mediados de los noventa se inició el “boom” de Internet. En esa época el número de proveedores de acceso privado se disparó, permitiendo a millones de personas acceder a Internet, que a partir de ese momento ya se empezó a conocer como la Red, desbancado a las demás redes de comunicación existentes (CompuServe, FidoNet/BBS). El punto de inflexión vino marcado por la aparición de implementaciones de TCP/IP gratuitas (incluso de implementaciones que formaban parte del sistema operativo) así como por la popularización y abaratamiento de medios de acceso cada vez más rápidos (módems de mayor velocidad, RDSI, ADSL, cable, satélite). El efecto de todos estos cambios fue de “bola de nieve”: a medida que se conectaban más usuarios, los costes se reducían, aparecían más proveedores e Internet se hacía más atractivo y económico, con lo que se conectaban más usuarios.

En estos momentos disponer de una dirección de correo electrónico, de acceso a la Web, etc., ha dejado de ser una novedad para convertirse en algo normal en muchos países del mundo. Por eso las empresas, instituciones, administraciones y demás están migrando rápidamente todos sus servicios, aplicaciones, tiendas, etc., a un entorno Web que permita a sus clientes y usuarios acceder a todo ello por Internet. A pesar del ligero descenso experimentado en el ritmo de crecimiento, Internet está destinado a convertirse en una suerte de servicio universal de comunicaciones, permitiendo una comunicación universal.

2.1.2. La WWW como servicio de Internet

La **WWW** (World Wide Web) o, de forma más coloquial, la Web, se ha convertido, junto con el correo electrónico, en el principal caballo de batalla de Internet. Ésta ha dejado de ser una inmensa “biblioteca” de páginas estáticas para convertirse en un servicio que permite acceder a multitud de prestaciones y funciones, así como a infinidad de servicios, programas, tiendas, etc.

2.1.3. Historia de las aplicaciones Web

Inicialmente la Web era simplemente una colección de páginas estáticas, documentos, etc., que podían consultarse o descargarse.

El siguiente paso en su evolución fue la inclusión de un método para confeccionar páginas dinámicas que permitiesen que lo mostrado fuese dinámico (generado o calculado a partir de los datos de la petición). Dicho método fue conocido como CGI (*common gateway interface*) y definía un mecanismo mediante el cual podíamos pasar información entre el servidor HTTP y programas externos. Los CGI siguen siendo muy utilizados, puesto que la mayoría de los servidores Web los soportan debido a su sencillez. Además, nos proporcionan total libertad a la hora de escoger el lenguaje de programación para desarrollarlos.

El esquema de funcionamiento de los CGI tenía un punto débil: cada vez que recibíamos una petición, el servidor Web lanzaba un proceso que ejecutaba el programa CGI. Como, por otro lado, la mayoría de los CGI estaban escritos en algún lenguaje interpretado (Perl, Python, etc.) o en algún lenguaje que requería entorno de desarrollo (VisualBasic, Java, etc.), esto implicaba una gran carga para la máquina del servidor.

Además, si la Web tenía muchos accesos al CGI, esto suponía problemas graves. Por ello se empiezan a desarrollar alternativas a los CGI para solucionar este grave problema de rendimiento. Las soluciones vienen principalmente por dos vías. Por un lado se diseñan sistemas de ejecución de módulos más integrados con el servidor, que evitan que éste tenga que instanciar y ejecutar multitud de programas. La otra vía consiste en dotar al servidor de un intérprete de algún lenguaje de programación (RXML, PHP, VBScript, etc.) que nos permita incluir las páginas en el código de manera que el servidor sea quien lo ejecute, reduciendo así el tiempo de respuesta.

A partir de este momento, se vive una explosión del número de arquitecturas y lenguajes de programación que nos permiten desarrollar aplicaciones Web. Todas ellas siguen alguna de las dos vías ya mencionadas. De ellas, las más útiles y las que más se utilizan son aquellas que permiten mezclar los dos sistemas, es decir, un lenguaje de programación integrado que permita al servidor interpretar comandos que “incrustemos” en las páginas HTML y un sistema de ejecución de programas más enlazado con el servidor que no presente los problemas de rendimiento de los CGI.

Es por esto que a través de este tiempo se ha venido observando una evolución en los lenguajes de programación Web.

2.2 Lenguajes de programación WEB

2.2.1. Introducción

Los lenguajes de programación Web han surgido para dar dinamismo a las páginas Web contenidas en los sitios o aplicaciones Web. Estos se agrupan en los que se ejecutan del lado del cliente y los que se ejecutan del lado del servidor. Del lado cliente se encuentran principalmente Javascript y Visual Basic Script, del lado del servidor tenemos como los más difundidos PHP, ASP, JSP y otros.

2.2.2. ¿Por qué PHP?

Para el desarrollo de nuestra aplicación Web debemos tener en cuenta cuestiones esenciales como: garantizar que nuestro sistema sea multiplataforma y que pueda ser desarrollado con software libre por las características de nuestro país estar bloqueado y de las restricciones de las principales compañías de desarrollo informático. Analizando esto el más adecuado es el PHP por ser libre, multiplataforma, su flexibilidad de comunicación con los principales gestores de bases de datos y además por sus potencialidades en funcionalidades y rapidez; además por su costo de adquisición.

Las principales ventajas son:

- ✓ Se basa en ser un lenguaje multiplataforma.
- ✓ Capacidad de conexión con la mayoría de los manejadores de base de datos que se utilizan en la actualidad.
- ✓ Leer y manipular datos desde diversas fuentes, incluyendo datos que pueden ingresar los usuarios desde formularios HTML.
- ✓ Capacidad de expandir su potencial utilizando la enorme cantidad de módulos (llamados ext's o extensiones).
- ✓ Es libre, por lo que se presenta como una alternativa de fácil acceso para todos.

- ✓ Permite las técnicas de Programación Orientada a Objetos.

2.3 Sistemas Gestores de Bases de Datos

2.3.1. Introducción

Hoy en día, son muchas las aplicaciones que requieren del almacenamiento de los datos para su gestión. Estos datos se deben almacenar en algún soporte permanente, y las aplicaciones deben disponer de un medio para acceder a ellos. Normalmente, la forma en que un programa accede a un fichero es a través del sistema operativo. Este provee de funciones como *abrir archivo*, *leer información del archivo*, *guardar información*, etc. No obstante, este procedimiento de acceso a ficheros es altamente ineficaz cuando se trata con un volumen elevado de información. Es aquí donde aparecen los Sistemas Gestores de Bases de Datos: los cuales proporcionan un interfaz entre aplicaciones y el sistema operativo, consiguiendo, entre otras cosas, que el acceso a los datos se realice de una forma más eficiente, sea más fácil de implementar y, sobre todo, de la manera más segura.

Los Sistemas Gestores de Bases de Datos (SGBD) son un tipo de software muy específico, dedicado a servir de interfaz entre las bases de datos y las aplicaciones que la utilizan. Se compone de un lenguaje de definición de datos, de un lenguaje de manipulación de datos y de un lenguaje de consulta.

Los SGBD deben cumplir lo siguiente:

- ✓ Abstracción de la información
- ✓ Independencia.
- ✓ Redundancia mínima.
- ✓ Consistencia.
- ✓ Seguridad.
- ✓ Integridad.
- ✓ Respaldo y recuperación.
- ✓ Control de la concurrencia.
- ✓ Tiempo de respuesta.

En cuanto a las características anteriormente expuestas se diferencian los SGBD actuales, así como por la forma de adquirirlos. Ejemplos de los más difundidos y utilizados a nivel mundial en el mundo del código abierto y multiplataforma: tenemos a PostgreSQL y MySQL.

2.3.2. ¿Por qué PostgreSQL?

Hemos podido ver la importancia de tener la información almacenada y centralizada utilizando como puente un gestor de base de datos. Podemos ver que muchos de los gestores de base de datos más utilizados hoy en día son muy buenos en cuanto a la gestión de los datos.

PostgreSQL es un servidor de base de datos relacional libre, liberado bajo la licencia BSD. Es una alternativa a otros sistemas de bases de datos de código abierto (como MySQL, Firebird y MaxDB), así como sistemas propietarios como Oracle o DB2.

Pero nos inclinamos por el mismo por las potencialidades que me brinda como:

- ✓ **DBMS Objeto-Relacional**

PostgreSQL aproxima los datos a un modelo objeto-relacional, y es capaz de manejar complejas rutinas y reglas. Ejemplos de su avanzada funcionalidad son consultas SQL declarativas, control de concurrencia multi-versión, soporte multi-usuario, transacciones, optimización de consultas, herencia, y arreglos.

- ✓ **Altamente Extensible**

PostgreSQL soporta operadores, funcionales métodos de acceso y tipos de datos definidos por el usuario.

- ✓ **Soporte SQL Comprensivo**

PostgreSQL soporta la especificación SQL99 e incluye características avanzadas tales como las uniones (joins) SQL92.

- ✓ **Integridad Referencial**

PostgreSQL soporta integridad referencial, la cual es utilizada para garantizar la validez de los datos de la base de datos.

✓ **API Flexible**

La flexibilidad del API de PostgreSQL ha permitido a los vendedores proporcionar soporte al desarrollo fácilmente para el RDBMS PostgreSQL. Estas interfaces incluyen Object Pascal, Python, Perl, PHP, ODBC, Java/JDBC, Ruby, TCL, C/C++, y Pike.

✓ **Lenguajes Procedurales**

PostgreSQL tiene soporte para lenguajes procedurales internos, incluyendo un lenguaje nativo denominado PL/ pgSQL. Este lenguaje es comparable al lenguaje procedural de Oracle, PL/SQL. Otra ventaja de PostgreSQL es su habilidad para usar Perl, Python, o TCL como lenguaje procedural embebido.

✓ **MVCC**

MVCC, o Control de Concurrencia Multi-Versión (Multi-Version Concurrency Control), es la tecnología que PostgreSQL usa para evitar bloqueos innecesarios. Mediante el uso de MVCC, PostgreSQL evita este problema por completo. MVCC está considerado mejor que el bloqueo a nivel de fila porque un lector nunca es bloqueado por un escritor. En su lugar, PostgreSQL mantiene una ruta a todas las transacciones realizadas por los usuarios de la base de datos. PostgreSQL es capaz entonces de manejar los registros sin necesidad de que los usuarios tengan que esperar a que los registros estén disponibles.

✓ **Cliente/Servidor**

PostgreSQL usa una arquitectura proceso-por-usuario cliente/servidor. Esta es similar al método del Apache 1.3.x para manejar procesos. Hay un proceso maestro que se ramifica para proporcionar conexiones adicionales para cada cliente que intente conectar a PostgreSQL.

✓ **Write Ahead Logging (WAL)**

La característica de PostgreSQL conocida como *Write Ahead Logging* incrementa la dependencia de la base de datos al registro de cambios antes de que estos sean escritos en la base de datos. Esto garantiza que en el hipotético caso de que la base de datos se caiga, existirá un registro

de las transacciones a partir del cual podremos restaurar la base de datos. Esto puede ser enormemente beneficioso en el caso de caída, ya que cualesquiera cambios que no fueron escritos en la base de datos pueden ser recuperados usando el dato que fue previamente registrado. Una vez el sistema ha quedado restaurado, un usuario puede continuar trabajando desde el punto en que lo dejó cuando cayó la base de datos.

2.4 Proceso de desarrollo de Software

2.4.1. ¿Qué metodología debo usar para el desarrollo de un Software?

Todos en algún momento nos hemos hecho esta pregunta, cuando hemos tenido que desarrollar un software. Y de hecho esta pregunta se torna muy importante, pues como arquitectos de Software, debemos tener un plano en que apoyarnos.

Todo desarrollo de software es riesgoso y difícil de controlar, pero si no llevamos una metodología *de por medio, lo que obtenemos es clientes insatisfechos con el resultado y desarrolladores aún más insatisfechos.

Sin embargo, muchas veces no se toma en cuenta el utilizar una metodología adecuada, sobre todo cuando se trata de proyectos pequeños de dos o tres meses.

Lo que se hace con este tipo de proyectos es separar rápidamente el aplicativo en procesos, cada proceso en funciones, y por cada función determinar un tiempo aproximado de desarrollo. Cuando los proyectos que se van a desarrollar son de mayor envergadura, ahí sí toma sentido el basarnos en una metodología de desarrollo, y empezamos a buscar cual sería la más apropiada para nuestro caso. Lo cierto es que muchas veces no encontramos la más adecuada y terminamos por hacer o diseñar nuestra propia metodología, algo que por supuesto no está mal, siempre y cuando cumpla con el objetivo.

Para dar una idea de qué metodología podemos utilizar y cual se adapta más a nuestro medio, mencionaré dos de ellas la cual considero son las más

importantes hoy en día para el proceso de desarrollo de un software: Proceso Unificado de Desarrollo (PUD) y Programación Extrema (XP).

Proceso Unificado de Desarrollo (PUD)

El Proceso Unificado de Desarrollo de software (PUD) es una metodología para acometer el desarrollo de sistemas informáticos, propuesta por Ivar Jacobson, Grady Booch y James Rumbaugh.

La misma tiene las siguientes características:

- ✓ Se trata de un estándar avalado por OMG, Object Management Group, consorcio de alrededor de 800 miembros (compañías de la industria del software), que busca el desarrollo de especificaciones para la industria del software que sean técnicamente excelentes, comercialmente viables e independientes del vendedor. OMG define *object management* como el desarrollo software que modela el mundo real mediante su representación como objetos. Estos objetos no son más que el encapsulamiento de atributos, relaciones y métodos de componentes software identificables.
- ✓ Recoge la experiencia de tres grandes metodologías anteriores:
 - OMT (Object Modeling Technique) de Rumbaugh.
 - OOAD (Object-Oriented Analysis and Design) de Booch.
 - Objectory de Jacobson
- ✓ Actualmente se usa en la mayoría de las empresas para grandes proyectos y se imparte por muchas instituciones.

El PUD es un proceso de desarrollo de software. Podemos mencionar que un proceso de desarrollo de software es el conjunto de actividades necesarias para transformar los requisitos de un usuario en un sistema software. Sin embargo, el proceso unificado es más que un proceso de trabajo, es un marco de trabajo genérico que puede especializarse para una gran variedad de sistemas software, para diferentes áreas de aplicación, diferentes tipos de organizaciones y diferentes niveles de aptitud.

El PUD está basado en componentes y utiliza UML (lenguaje de modelado unificado (Unified Modeling Language) para visualizar, especificar y documentar cada una de las partes que comprende el desarrollo de software). Está dirigido por casos de uso (forma en como un actor opera con el sistema en desarrollo), porque con éstos se especifican las funcionalidades que el sistema proporciona al usuario. Los casos de uso representan los requisitos funcionales y fuerzan a pensar en términos de importancia para el usuario y no sólo en términos de qué funciones sería bueno tener. Los casos de uso no sólo son una herramienta para especificar los requisitos del sistema, también guían su diseño, implementación y pruebas, es decir, guían todo el desarrollo software.

El PUD está centrado en la arquitectura, pues la manera en que se organiza el sistema depende de los casos de uso clave y debe tener en cuenta la comprensibilidad, la facilidad de adaptación al cambio y la reutilización. Los casos de uso clave son aquellos que dotan al sistema con la funcionalidad fundamental para los usuarios y sin los cuales, los demás casos de uso no tienen sentido.

El PUD es iterativo e incremental. El trabajo se divide en partes más pequeñas, llamadas iteraciones. En cada iteración se recorren los flujos de trabajo (*requisitos, análisis y diseño, implementación y pruebas*) y se obtiene una versión interna. En síntesis, en cada iteración se amplía el sistema con nuevos casos de uso, se identifican nuevos riesgos y se mitigan los ya conocidos. Las iteraciones se agrupan en fases, que por orden secuencial son las siguientes: *inicio, elaboración, construcción y transición*.

Cada una centra sus esfuerzos más en unos flujos de trabajo que en otros. La etapa de inicio se centra en la captura de requisitos y el análisis. La etapa de elaboración lo hace con el análisis y el diseño. Las etapas de construcción y transición se centran en el diseño, implementación y pruebas.

El PUD es una metodología de desarrollo pensada para proyectos grandes, a largo plazo y con un equipo de desarrollo numeroso.

Programación Extrema (XP)

XP es una de las metodologías de desarrollo de software con más éxito en la actualidad. Se utiliza en proyectos con equipos de desarrollo pequeños y con plazos de entrega corto. La metodología consiste en una programación rápida o extrema. Una particularidad es tener como miembro del equipo al usuario final. Esta metodología tiene las siguientes características:

- ✓ **Pruebas Unitarias:** Las pruebas se realizan a los principales procesos sistemáticamente durante todo el desarrollo.
- ✓ **Refactorización:** El código se cambia constantemente para que sea lo más reutilizable y flexible posible. La refactorización consiste en el cambio del código para añadir más calidad al mismo pero sin cambiar la funcionalidad de la aplicación.
- ✓ **Programación en pares:** Propone la programación en pares, la cual consiste en que dos desarrolladores participen en un proyecto en un mismo puesto de trabajo. Cada miembro lleva a cabo la acción que el otro no está haciendo en ese momento.

XP propone que el desarrollo comienza dotando a la aplicación de poca funcionalidad que va siendo aumentada a medida que avanza el desarrollo con retroalimentación continua. La gestión del cambio se convierte en parte esencial del desarrollo. El coste del cambio no depende de la fase o etapa. Por último no se introducen funcionalidades antes de que sean necesarias.

XP incorpora activamente al cliente en el proceso de desarrollo. El cliente decide qué se implementa y sabe en todo momento el estado real y el progreso del proyecto. Puede añadir, cambiar o quitar requisitos en cualquier momento. Puede obtener un sistema funcionando en 3 ó 4 meses.

XP ofrece al desarrollador el poder de decidir como se implementan los procesos y crear el sistema con la mejor calidad posible. Puede disponer del cliente en cualquier momento para que le aclare algunos requisitos. Puede

estimar el esfuerzo necesario para implementar el sistema y puede cambiar los requisitos en base a nuevos descubrimientos.

Lo fundamental de XP es:

- ✓ **La comunicación:** Entre los usuarios y los desarrolladores.
- ✓ **La simplicidad:** Al desarrollar y codificar los módulos del sistema.
- ✓ **La retroalimentación:** Concreta y frecuente del equipo de desarrollo, el cliente y los usuarios finales.

2.4.2. ¿Qué es UML?

A mediados de los noventa existían muchos métodos de análisis y diseño Orientado a Objetos lo que suponía que los mismos conceptos tenían distinta notación según el método de que se tratara. Ante esta situación de confusión, en 1994 Booch, Rumbaugh y Jacobson decidieron unificar sus métodos dando lugar a *UML*. Esta unificación fue promovida por el OMG de tal manera que *UML* se convirtió en la notación estándar para la descripción de métodos software. Según su definición, *UML es un lenguaje para visualizar, especificar, construir y documentar los artefactos de un sistema que involucra una gran cantidad de software, desde una perspectiva Orientada a Objetos.*

Se caracteriza por:

- ✓ UML es un lenguaje.
- ✓ UML es un lenguaje para visualizar.
- ✓ UML es un lenguaje para especificar.
- ✓ UML es un lenguaje para construir.
- ✓ UML es un lenguaje para documentar.

2.4.3. Qué son las Herramientas CASE

Se puede definir a las Herramientas CASE como un conjunto de programas y ayudas que dan asistencia a los analistas, ingenieros de software y desarrolladores, durante todos los pasos del Ciclo de Vida de desarrollo de un Software.

2.4.4. Porqué deberíamos usar herramientas CASE de modelado con UML

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

- ✓ Los Analistas de Negocio/ Sistemas pueden capturar los requisitos del negocio/sistema con un modelo de casos de uso.
- ✓ Los Diseñadores/Arquitectos pueden producir el modelo de diseño para articular la interacción entre los objetos o los subsistemas de la misma o de diferentes capas (los diagramas UML típicos que se crean son los de clases y los de interacción).
- ✓ Los Desarrolladores pueden transformar rápidamente los modelos en una aplicación funcionando, y buscar un subconjunto de clases y métodos y asimilar el entendimiento de cómo lograr interfaces con ellos.

Por estas razones, las herramientas CASE de UML acompañadas con metodologías, nos brindan una forma de representar sistemas demasiados complejos para comprenderlos a través de su código fuente subyacente y nos permiten desarrollar la solución de software correcta más rápido y más económicamente.

Sin embargo, las herramientas CASE varían con respecto a las capacidades de modelado con UML, el soporte del ciclo de vida del proyecto, las ingenierías

directa y reversa, el modelado de datos, la performance, el precio, el soporte, la facilidad de uso, etc.

2.4.5 ¿Por qué Visual Paradigm como herramienta CASE, UML como lenguaje de modelado y RUP como metodología?

Hemos podido ver los principales aspectos en cuanto a la hora de seleccionar la metodología apropiada para el desarrollo de nuestro sistema informático y nos inclinamos por RUP ya que el mismo es recomendable a proyectos grandes, que es el caso nuestro, además es el ideal cuando se cuenta con equipos de desarrollo grandes y cuyos flujos dentro del proceso de desarrollo están bien especificados y determinado por roles, por ser un proceso iterativo e incremental y por su tratamiento de chequeos a las pruebas de producto informativo en desarrollo. Sobre lenguaje, si decidimos a RUP como metodología, la misma se basa en modelado con el lenguaje UML, además ya que el mismo nos permite:

- ✓ Modelar sistemas utilizando técnicas orientadas a objetos (OO).
- ✓ Especificar todas las decisiones de análisis, diseño e implementación, construyéndose así modelos precisos, no ambiguos y completos.
- ✓ Puede conectarse con lenguajes de programación (Ingeniería directa e inversa).
- ✓ Documentar todos los artefactos de un proceso de desarrollo (requisitos, arquitectura, pruebas, versiones, etc.).
- ✓ Cubre las cuestiones relacionadas con el tamaño propias de los sistemas complejos y críticos.
- ✓ Es un lenguaje muy expresivo que cubre todas las vistas necesarias para desarrollar y luego desplegar los sistemas.
- ✓ Existe un equilibrio entre expresividad y simplicidad, pues no es difícil de aprender ni de utilizar.

Nos inclinamos por el Visual Paradigm ya que me ofrece lo siguiente:

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

2.5 Otras Herramientas

Para cumplimentar nuestro trabajo, no son necesarias otras herramientas que me aseguren otros aspectos no menos importantes como:

- ✓ Publicación y edición de nuestra aplicación Web.
- ✓ Diseño de las imágenes principales a usar en nuestra aplicación.
- ✓ Codificación y depuración en el servidor.
- ✓ Control de versiones realizadas por cada uno de los integrantes del equipo de desarrolladores tanto para la codificación como para el modelado.

Las herramientas seleccionadas para asegurar lo anteriormente expuesto tenemos: **Macromedia Dreamweaver 8, Adobe PhotoShop CS, Zend Studio (Servidor-Desarrollo) y TortoiseSVN.**

A continuación algunas características de ¿Por qué? La selección de las mismas.

✓ **Macromedia Dreamweaver 8.**

Por ser la herramienta de desarrollo Web líder de su sector que permite que los usuarios diseñen, desarrollen y mantengan de forma eficaz sitios Web y aplicaciones basados en estándares. Brinda múltiples herramientas visuales de diseño y un entorno de codificación adaptables a lenguajes de programación Web (PHP), trabajo con hojas de estilos CSS, permite la comparación de archivos para determinar que ha cambiado, permite el trabajo directo del lado del servidor, etc.

Nota: quisiéramos hacer la salvedad de que esta es la única herramienta utilizada que nos abandona en cuanto al criterio de ser multiplataforma, por lo que se está trabajando y profundizando en otras herramientas que me aseguren esta línea base y que me brinden el soporte necesario que me brinda el Macromedia Dreamweaver 8.

✓ **Adobe PhotoShop CS.**

Adobe PhotoShop CS es el software estándar de edición de imágenes profesional.

Posee una alta productividad redefinida (conjunto de herramientas profesionales de pintura, dibujo y retoques, así como una extensa galería de filtros), además múltiples opciones para el diseño gráfico (trabajo con capas, pinceles artísticos), trabajo con fotografías (corrección de color), producción Web (partición de imágenes, soporte para datos de variables) y video (Soporte para píxeles no cuadrados, exportación de capas en archivos).

✓ **Zend Studio.**

Zend Studio es un editor de código PHP y para el desarrollo de aplicaciones PHP.

Posee un analizador de código bastante completo, permite la depuración del mismo y la detección de errores ocultos, además acceso directo a

gestores de bases de datos, manipulación de archivos, brinda una amplia gama de bloques de códigos con funcionalidades probadas por la comunidad de PHP, permite la gestión de proyectos, bastante documentación, soporte para servicios Web, así como controla versiones y otras.

✓ **TortoiseSVN.**

TortoiseSVN es un cliente gratuito de código abierto para el sistema de control de versiones *Subversión*.

Sus principales características son: integración con el shell de Windows, sobreimpresión de iconos, fácil acceso a los comandos de Subversión, versionado de carpetas, confirmaciones atómicas, metadatos versionados, elección de capas de red, manejo de datos consistente, etiquetado y creación de ramas eficiente y extensibilidad.

2.6 Conclusiones

Como se ha podido ver en este capítulo se han expuesto los puntos bases para dar cumplimiento al problema enunciado, explicándose las teorías fundamentales en cuanto al desarrollo tecnológico de la ciencia informática como tal en la actualidad.

Todos sabemos la revolución que el mundo de la Informática y su diversificación, así como el proceso de planificación en su constante cambiar a través del tiempo y cada vez más adaptable al mundo tan competitivo en que vivimos.

Múltiples investigaciones se han realizado en cuanto a metodologías de desarrollo, lenguajes de modelado y programación Web, gestores de base de datos, herramientas de edición Web, las cuales constituyen la línea base de este trabajo investigativo por lo que considero que las técnicas seleccionadas son las ideales para desarrollar la aplicación Web *“Control de la ejecución de presupuesto del MINFAR”*.



Descripción de la Solución Propuesta

En este capítulo se muestran un conjunto de artefactos resultantes de la aplicación de la metodología de desarrollo Proceso Unificado del Software de Rational (RUP) principalmente en las etapas: modelamiento del negocio, requisitos y análisis-diseño. En cada ciclo de vida que se da por cada iteración se ha llevado bajo la disciplina de desarrollo.

3.1 Modelamiento del Negocio.

Antes de adentrarnos en el desarrollo de un sistema el primer paso en el proceso de desarrollo de software es precisamente alcanzar cierto nivel de conocimientos sobre el problema en cuestión. Debemos ser capaces de comprender la estructura y la dinámica de la organización en la cual se va a implantar nuestro sistema, comprender los problemas actuales de la organización e identificar las mejoras potenciales, asegurar que los consumidores, usuarios finales y desarrolladores tengan un entendimiento común de la organización y derivar los requerimientos del sistema que va a soportar la organización. Para poder lograr esos propósitos, el proceso de modelamiento permite obtener una visión de la organización que permita definir los procesos, roles y responsabilidades de la organización en los modelos de casos de uso del negocio y de objetos. Debemos ser capaces de analizar el

estado del negocio, es decir, determinar si los procesos involucrados están claramente definidos que en nuestro caso es así, de ahí este modelado.

3.1.1 Modelo de casos de uso.

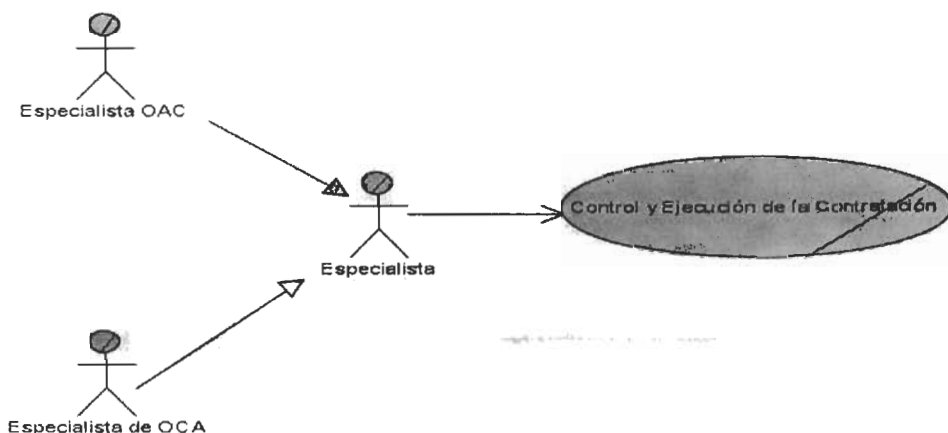


Fig. 3.1. Diagrama de casos de uso del negocio

3.1.2 Especificaciones de los casos de uso.

Caso de Uso:	Control y Ejecución de la Contratación
Actores:	Especialista (inicia)
Trabajadores:	Especialista Ejecutor CB, Especialista Controlador CB, Especialista Empresa Importadora
Resumen:	El caso de uso se inicia cuando el Especialista suministra el informe de surtidos aprobados en la planificación al Especialista Controlador CB para su revisión. Una vez revisado y aprobado es enviado al Especialista Ejecutor CB para la elaboración de los documentos 711. Estos documentos 711 son revisados por el Especialista

	<p>Controlador CB. Luego de ser aprobados, son enviados al Especialista Empresa Importadora para ver las posibles ofertas que existen para satisfacerlos. Después de ser encontrados los posibles suministradores de estos surtidos es informado al Especialista Controlador para su valoración y aprobación de contratación. Una vez aprobadas las ofertas los medios e insumos reflejados en los documentos 711 comienzan a contratarse. Luego de ser contratados todos los medios, comienza el proceso de control por parte del Especialista Controlador CB, cuya información es brindada por el Especialista Empresa Importadora finalizando así el caso de uso.</p>
Precondiciones:	Debe existir un informe de los surtidos aprobados.
Flujo Normal de Eventos	
Sección ""	
Acción del Actor	Respuesta del Negocio
<p>1. El Especialista presenta el informe de surtidos aprobados.</p>	<p>1.1. El Especialista Controlador CB revisa el informe de los surtidos para determinar si existen errores en él, como la falta de ortografía en el nombre de los medios e insumos, medios no existentes, etc. El informe de los surtidos está correcto.</p> <p>1.2. El Especialista Controlador CB entrega el informe de los surtidos al Especialista Ejecutor CB. El mismo elabora los documentos 711 correspondientes y se lo entrega al Especialista Controlador CB para su</p>

	<p>revisión y aprobación. Los documentos 711 están correctos.</p> <p>1.3. Los documentos 711 son presentados al Especialista Empresa Importadora para la búsqueda de las ofertas y los posibles suministradores a las mismas. Son revisados los documentos 711 por si tienen errores. Los documentos 711 están correctos.</p> <p>1.4. El Especialista Empresa Importadora informa al Especialista Controlador CB las ofertas existentes y los suministradores a las mismas para su valoración y aprobación.</p> <p>1.5. El Especialista Controlador CB revisa y aprueba las ofertas e informa su aprobación al Especialista Empresa Importadora.</p> <p>1.6. Se recibe notificación de aprobación y se elaboran los contratos correspondientes a los documentos 711 por el Especialista Empresa Importadora. Se presentan copias de los contratos elaborados al Especialista Controlador CB.</p> <p>1.7. El Especialista Controlador CB notifica al Especialista sobre los</p>
--	---

	contratos elaborados por los distintos documentos 711.
2. El Especialista recibe notificación de los contratos y solicita controlar la ejecución de los mismos.	<p>2.1. El Especialista Controlador CB solicita información sobre estado de ejecución de los contratos.</p> <p>2.2. El Especialista Empresa Importadora informa sobre estado de ejecución de los contratos al Especialista Controlador CB.</p>
Flujos Alternos	
Acción del Actor	Respuesta del Negocio
<p>Actividad 1.1. Si el informe de los surtidos aprobados presenta errores es devuelto al Especialista para su rectificación.</p> <p>Actividad 1.2. Si al menos un documento 711 presenta errores es devuelto al Especialista Ejecutor CB para su reelaboración.</p> <p>Actividad 1.3. Si al menos un documento 711 presenta errores es devuelto al Especialista Ejecutor CB para su rectificación.</p>	
Poscondiciones	El Especialista realiza contratación y chequea estado de ejecución.

Tabla 2. Descripción del caso de uso del negocio Control y Ejecución de la Contratación

3.1.3 Actores y trabajadores del negocio.

Actores del Negocio

Especialista	Este actor es una generalización de los actores Especialista OCA y Especialista OAC. Es el que inicia el caso de uso Control y Ejecución de la Contratación. Es el encargado de realizar y controlar el proceso de contratación.
Especialista OCA	Son aquellos órganos consumidores autorizados dentro del proceso de contratación. Solo controlan el estado de ejecución de sus planes correspondientes.
Especialista OAC	Son aquellos órganos abastecedores centrales dentro del proceso de contratación, los cuales se encargan de conformar y controlar el estado de ejecución de sus planes asociados.

Tabla 3. Descripción de los actores del negocio

Trabajadores del Negocio

Especialista Ejecutor CB	Es el encargado de realizar los documentos 711 después que el Especialista de Control de CB revisa los surtidos enviados por las OCA, cuando le es entregado el documento de la oferta, se encarga de asociarle al suministrador que le toca cada uno de los medios e insumos plasmado en el documento 711, después le da seguimiento a todo el proceso de contratación, controlando los cambios que ocurren en cada contrato hasta su fin.
	Es el encargado en el negocio de revisar los surtidos que vienen de cada OCA, se encarga de suministrar ese

<p>Especialista Control CB</p>	<p>surtido a el Especialista Ejecutor de CB para que realice los documentos 711, después este Especialista es el encargado de firmar el documento 711 que se dirigirá a la Empresa Importadora para su proceso de contratación.</p>
<p>Especialista Empresa Importadora</p>	<p>Es el encargado de revisar los documentos 711, en cuanto a precio y medios existentes en el mercado además de buscar las ofertas correspondientes para cada documento 711, realiza el proceso de contratación de cada documento 711 hasta que culmine este, e informa a los Centros de Balance correspondientes.</p>

Tabla 4. Descripción de los trabajadores del negocio

3.1.4 Modelo de Objetos

Este modelo describe la relación existente entre los trabajadores del negocio y las entidades del negocio. A continuación una muestra del mismo para nuestro sistema:

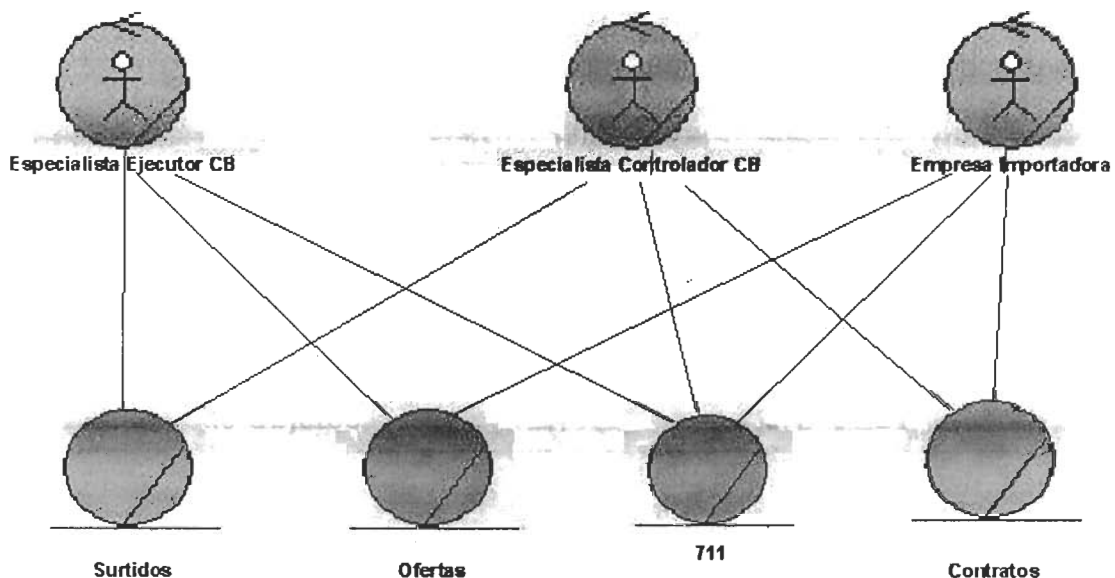


Fig. 3.2. Modelo de objetos del negocio

3.2 Captura de Requisitos

Como ya hemos podido apreciar el modelamiento del negocio brinda una vía natural para determinar los requerimientos del sistema de información. Lograr una comunicación efectiva entre los usuarios y el equipo de proyecto con el objetivo de llegar a un entendimiento de lo que hay que hacer, es la clave del éxito en la producción de un software. Nuestro objetivo es eliminar las incongruencias entre nosotros que somos el equipo de desarrollo y el cliente, para así lograr un sistema acorde y que nos satisfaga a ambos.

En la captura de requisitos debemos centrar nuestro esfuerzo en reconocer el problema tal y como lo ve el usuario, evaluación del problema y síntesis de la evaluación, un modelado que nos ayude a entender la entidad en cuestión, la construcción de un prototipo de alto nivel del sistema, revisión por parte del usuario y la firma de un contrato que refleje acuerdos entre ambas partes. Los requisitos se clasifican en funcionales y no funcionales. Todo lo antes mencionado se evidencia a continuación.

3.3.3 Requisitos Funcionales

Los requerimientos funcionales como bien nos sugiere, son capacidades o condiciones que el sistema debe cumplir, por lo que en nuestro caso este sistema que se está analizando debe tener la propiedad de satisfacer los requisitos siguientes:

RF1. Gestionar un modelo 711

- 1.1. Actualizar modelos 711.
 - 1.1.1. Elaborar nuevos modelos 711.
 - 1.1.2. Modificar un modelo 711 seleccionado.
 - 1.1.3. Guardar la información introducida o modificada en el modelo 711.
 - 1.1.4. Eliminar un modelo 711 seleccionado.
- 1.2. Recuperar modelos 711 elaborados.

- 1.2.1. Por los siguientes criterios: *Especialista, Cuenta Bancaria, Importador, No, Fecha Aprobado, Fecha Presentado, Tipo Medio y Medio.*
- 1.3. Mostrar listado de todos los modelos 711 recuperados mostrando *No. Modelo 711, Consumidor, Importador, Fecha Presentado y Fecha Aprobado.*
- 1.4. Mostrar reporte de modelo 711 seleccionado.
 - 1.4.1. Imprimir modelo 711 visualizado en el reporte

RF2. Gestionar Contratos.

- 2.1. Actualizar contratos.
 - 2.1.1. Elaborar contratos.
 - 2.1.2. Modificar un contrato seleccionado.
 - 2.1.3. Guardar la información introducida o modificada en el contrato.
 - 2.1.3. Eliminar un contrato seleccionado.
- 2.2. Recuperar contratos elaborados.
 - 2.2.1. Por los siguientes criterios: *No, Proveedor, Consumidor, Fecha Firma, Rate MIN, Rate MAX, General y Criterio.*
- 2.3. Mostrar listado de los contratos recuperados mostrando: *No, Proveedor y Fecha Firma.*
- 2.4. Mostrar reporte de contrato seleccionado.
 - 2.4.1. Imprimir el contrato visualizado en el reporte.

RF3. Consultar el estado de la contratación.

- 3.1. Mostrar listado de los contratos propios del especialista o usuario en ejecución.
- 3.2. Mostrar listado de los medios e insumos del contrato seleccionado.
- 3.3. Mostrar reporte sobre estado de contratación de los medios e insumos seleccionados.
- 3.4. Mostrar documentos 711 asociados a un contrato.

3.5. Notificar a los consultores sobre nuevo estado de ejecución de sus contratos.

RF4. Búsqueda de documentos

4.1. Por los siguientes criterios *Tipo Documento (General)*, *Genérico (Filtro)* y *Criterio de Búsqueda (Criterio)*.

3.3.4 Requisitos No Funcionales

Los requerimientos no funcionales como bien nos sugiere, son propiedades o cualidades que el producto debe tener. Los mismos forman una parte significativa de la especificación y son importantes para que clientes y usuarios puedan valorar las características no funcionales del producto, pues si se conoce que el mismo cumple con la toda la funcionalidad requerida, las propiedades no funcionales, como cuán usable, seguro, conveniente y agradable, pueden marcar la diferencia entre un producto bien aceptado y uno con poca aceptación. De todo esto pudimos identificar los siguientes:

Apariencia o interfaz externa

- ✓ La interfaz a implementar debe ser sencilla y con pocas entradas para que los usuarios que no son personas expertas en el uso de la WEB no necesiten tanto tiempo de adiestramiento.
- ✓ El diseño debe estar concebido para simular una aplicación de escritorio.
- ✓ Debe tener colores que le sean refrescantes para la vista y siguiendo los patrones de colores representativos de la entidad como: verde, gris, blanco y azul fundamentalmente.
- ✓ Debe utilizarse tecnología de frames (marcos).
- ✓ Para lograr que los usuarios y trabajadores se encuentren a gusto con el Sistema incorporándole lo antes descrito se le ponen otros elementos de diseño como son :
 - Gráficos de encabezamiento.
 - Estilos y formatos de texto.

- Estilo y paletas de color de los gráficos.

Usabilidad

- ✓ El sistema debe ser de fácil manejo para los usuarios que tengan niveles básicos sobre la computación o hallan trabajado con Web.
- ✓ Debe tener una opción de ayuda sobre las principales operaciones que se realizan y sus iconos respectivos para lograr un menor tiempo de aprendizaje.
- ✓ El sistema simulará tal y como es el proceso de contratación para lograr el menor tiempo en cuanto a la comprensión del sistema y del proceso.

Rendimiento

- ✓ La búsqueda de cualquier documento y su visualización en pantalla no debe exceder los 10 segundos.
- ✓ El almacenamiento de la información correspondiente a los modelos 711 y los contratos no debe exceder los 15 segundos.
- ✓ La aplicación debe estar concebida para el consumo mínimo de recursos.

Soporte

Para el servidor de aplicaciones:

- ✓ Se requiere que esté instalado un intérprete de ficheros PHP rápido y con las últimas actualizaciones del lenguaje.

Para el servidor de base de datos:

- ✓ Se requiere esté instalado un gestor de base de datos que soporte grandes volúmenes de datos, maneje la concurrencia y maneje transacciones.

Para el cliente:

- ✓ Se requiere esté instalado un navegador que interprete Javascript y versiones HTML 3.0 o superior.

Portabilidad

- ✓ El sistema deberá ser compatible con el sistema operativo UNIX (Linux y distribuciones como RedHat, Mandrake, Gentoo, Ubuntu, Debian y otras).
- ✓ El sistema deberá ser compatible con el sistema operativo Windows (versiones como 2000 y XP), siendo además accesible en la Intranet desde cualquier navegador compatible con HTML 3.0.

Hardware

Para las computadoras del cliente:

- ✓ Se requiere tengan tarjeta de red.
- ✓ Se requiere tengan al menos 64 MB de memoria RAM.
- ✓ Se requiere al menos 100MB de disco duro.
- ✓ Procesador 512 MHz como mínimo.

Para los servidores:

- ✓ Se requiere tarjeta de red.
- ✓ Se requiere tenga la menos 256MB de RAM.
- ✓ Se requiere al menos 1GB de disco duro.
- ✓ Procesador 1.2 GHz como mínimo.

Software

- ✓ El sistema se desarrollará con tecnología PHP version 5.0 o superior.
- ✓ Se utilizará un servidor con el sistema operativo instalado Windows 2000 o superior, o con sistema operativo UNIX (Linux) preferencialmente.
- ✓ Se utilizará tecnología Apache versión 2.0 o superior para el servidor Web.
- ✓ El sistema utilizará una base datos implementada en PostgreSQL version 8.0 o superior.

- ✓ En las computadoras de los clientes se garantizará versiones de Windows 98 o superior, así como Linux y sus correspondientes distribuciones.
- ✓ En las computadoras de los clientes solo se requiere de un navegador (Internet Explorer version 4.0 o superior, Mozilla Firefox version 1.5 o superior con las extensiones Web Developer, FireBug e InspectThis).
- ✓ La comunicación de las computadoras clientes con el servidor será a través de conexiones de fibra óptica, a una velocidad constante de 100 Mbps.

Seguridad

- ✓ El sistema debe comunicarse usando un protocolo seguro. (https).
- ✓ Los datos no pueden viajar de forma transparente por la red, deben ser encriptados.
- ✓ Chequear si el usuario que está accediendo al sistema esta autenticado y brindarle servicio de autenticación.
- ✓ Mostrar las operaciones de acuerdo al rol del usuario y no más.
- ✓ Mantener la integridad de la información, es decir que no se perderá durante su almacenamiento o transporte.
- ✓ Permitir que cuando se borre cualquier documento o información pueda existir una opción de advertencia antes realizar la acción.
- ✓ Realizar auditoría a los principales eventos dentro del sistema, registrando al usuario, el tipo de usuario y los eventos efectuados.
- ✓ Manejo de la inyección SQL.

Disponibilidad

- ✓ El sistema deberá estar disponible las 24 horas del día para todos los usuarios con derechos a utilizar el sistema.

Confiabilidad

- ✓ La información manejada por el sistema está protegida de acceso no autorizado y divulgación.

Integridad

- ✓ La información manejada por el sistema será objeto de cuidadosa protección contra la corrupción, de la misma forma será considerada igual a la fuente o autoridad de los datos.

Fiabilidad

- ✓ La tasa de fallos del sistema no debe exceder 2 fallos por semana.

Legales

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

3.3 Modelado del Sistema

Una de las situaciones más difíciles a la hora de construir un sistema es precisamente saber qué construir. Ninguna otra parte del trabajo conceptual es tan compleja como establecer los requerimientos técnicos detallados, incluyendo todas las interfaces con las personas, máquinas, y otros sistemas. Este establecimiento de los requisitos es el punto de partida para la identificación de los principales casos de uso y actores del sistema en cuestión.

Por lo tanto podemos plasmar todos los requerimientos en un modelo de casos de uso del sistema y a su vez identificar los principales actores que interactuarán con el mismo. Todo lo anteriormente mencionado se muestra a continuación.

3.3.1 Actores del Sistema para automatización

Ejecutor CB	En el sistema gestionará los documentos: 711 y contratos. Para mejor explicación asegura los RF1 y RF2.
Controlador CB	En el sistema se encarga de controlar el estado de ejecución de los contratos elaborados. Para mejor explicación asegura requerimiento funcional RF3.
Especialista CB	Es una generalización de los actores <i>Especialista Ejecutor CB</i> y <i>Especialista Controlador CB</i> .

Tabla 5. Descripción de los actores del sistema

3.3.2 Modelado de casos de uso del sistema

A continuación se muestra el diagrama de los casos de uso de nuestro sistema en cuestión:

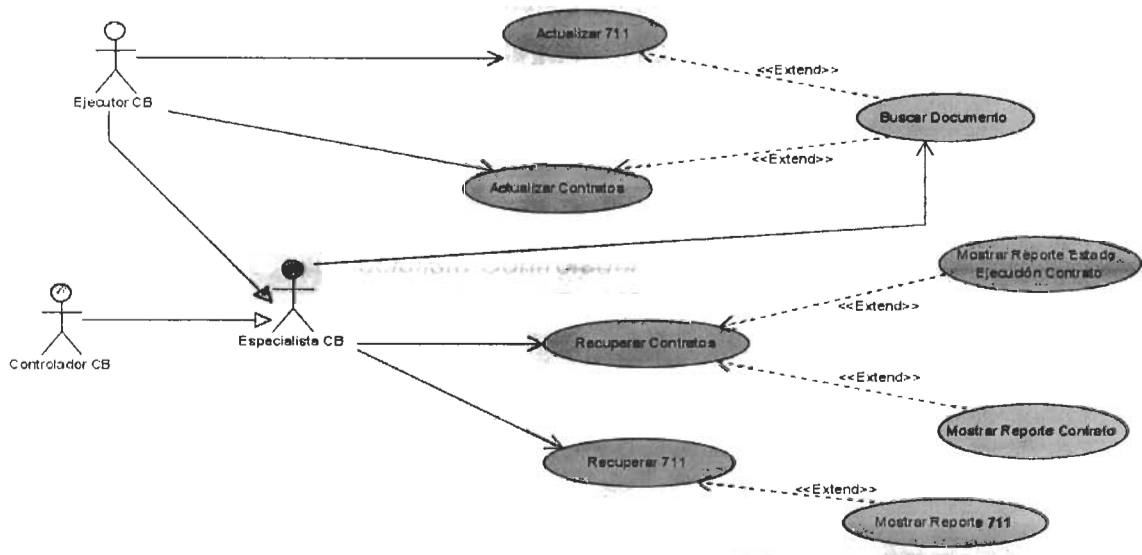


Fig.3.4. Diagrama de casos de uso del sistema

3.3.3 Especificaciones de los casos de usos

CU-1	Actualizar 711
Actor	Ejecutor CB.
Descripción	Un usuario solicita cualquier tipo de actualización (elaboración, modificación o eliminación) sobre un documento 711.
Referencia	RF1

CU-2	Actualizar Contrato
Actor	Ejecutor CB
Descripción	Un usuario solicita cualquier tipo de actualización (elaboración, modificación o eliminación) sobre un contrato.
Referencia	RF2

CU-3	Buscar Documento
Actor	Especialista CB (inicia).
Descripción	Un usuario solicita buscar un documento, ya sea documento 711 o un contrato.
Referencia	RF4

3.3.4 Expansión de los casos de Uso

A continuación se describen los principales casos de uso en los cuales centraremos nuestro trabajo, a continuación las descripciones expandidas de los casos de uso, las otras descripciones. **Ver Anexo III y Anexo IV.**

Caso de Uso:	Actualizar 711
Actor(es):	Ejecutor CB.
Propósito:	Actualizaciones de cualquier índole sobre los documentos 711.
Resumen:	Este caso de uso se inicia cuando el actor necesita actualizar información de algún documento 711. De acuerdo a su requerimiento funcional puede seleccionar tipo de actualización, si va elaborar, modificar o eliminar un documento 711.
Referencias:	RF1.
Caso de uso asociados	Buscar Documento (extensión)
Precondiciones:	El usuario levanta el sistema y se encuentra en la página principal. Debe haberse solicitado el servicio de autenticación. Para poder modificar o eliminar la información de documentos 711 debe existir al menos uno ya elaborado.
Pantalla 1	
Acción del Actor	Respuesta del Sistema
1. El Usuario elige la operación a realizar.	2. <ul style="list-style-type: none"> a) Si elige la opción Elaborar 711 (A), ver sección: "Elaborar 711". b) Si elige la opción Actualizar 711 (B), ver sección: "Actualizar 711"
Sección: "Elaborar 711"	

Elaborar 711

Detalles [Consumidor]		Detalles [Importador]	
Consumidor	<input type="text"/>	Importador	<input type="text"/>
Representante	<input type="text"/>	Código	<input type="text"/>
Centro de Pago	<input type="text"/>	Puerto Destino	<input type="text"/>
Cuenta Bancaria	<input type="text"/>	No. Aprobaciones	<input type="text"/>
Código	<input type="text"/>	No. de Control	<input type="text"/>
Presentado	<input type="text"/>	Consumidor	<input type="text"/>
Aprobado	<input type="text"/>	Suministrador	<input type="text"/>
Observaciones	<input type="text"/>	Medios	<input type="text"/>
		Actualizar	
		Aceptar	
		Cancelar	

Pantalla 2

1. El Sistema muestra Pantalla 2 con los filtros: consumidor(A), centro de pago (C), cuenta bancaria (D), importador (F), puerto destino (H), consumidor (L) y suministrador (M) con su primer valor, así como las cajas de texto representante (B), código consumidor (E), código importador (G), no. aprobaciones (I) y observaciones (N).

Nota: Estos elementos obtendrán los valores de un documento 711 seleccionado si esta pantalla 2 es llamada desde pantalla 4.

2. El Usuario para "Detalles [Consumidor]" escoge el consumidor

3. Ver sección "Actualizar Medios".

<p>(A), introduce el representante del consumidor (B), escoge el centro de pago (C), escoge la cuenta bancaria (D) e introduce el código del consumidor (E). Para "Detalles [Importador]" escoge el importador (F) e introduce el código (G). Escoge el puerto destino (H) e introduce no. de aprobaciones (I). Para "Fechas" introduce fecha de presentado (K) y fecha de aprobado (J). Para no. de control escoge el consumidor (L) y suministrador (M). Introduce observaciones si así lo desea (N) si no lo puede dejar en blanco. Presiona botón Actualizar (N).</p>	
<p>4. Al terminar de introducir los datos presiona botón Aceptar (O).</p>	<p>5. El sistema determina si la pantalla 2 no ha sido visualizada desde pantalla 4, si es así entonces registra nuevo documento 711 en el sistema sino actualiza información del documento 711 sobre el que se esta trabajando.</p>
<p>6. El usuario presiona botón Cancelar (P).</p>	<p>7. El sistema muestra pantalla 1.</p>
<p>Sección: "Actualizar Medios"</p>	

Actualizar Medios

Detalles Generales 711

Tipo Medio Código

Aprobación Proyecto

Medios

Medio Precio

Suministrador Cantidad

No. Pieza

Descripción

No	Suministrador	No. Pieza	Descripción	Precio	Cantidad	Importe

Pantalla 3

1. El sistema muestra listado de medios asignados al documento 711 en el grid (O). Además muestra para "Detalles Generales 711" el filtro tipo de medio (A) con el valor "Seleccione", así como la aprobación (B), el código (C) y el proyecto (D) de solo lectura, así como para "Medios" en los filtros medio (E) los medios pertenecientes al tipo mostrado en el filtro Tipo de medio (A), para el primer medio mostrado en el filtro medio (E) se muestran sus

	<p>detalles en las cajas de texto No. Pieza (G), descripción (H) y precio (I) de solo lectura, también muestra cantidad (J) en blanco.</p>
<p>2. El usuario escoge tipo de medio (A).</p>	<p>3. El sistema actualiza en "Medios" los filtros medio (E), estos distintos a los mostrados en el grid (O), suministrador (F), las cajas de texto no. pieza (G), descripción (H) y precio (I) de solo lectura todos con los detalles del primero medio mostrado en el filtro medio (E), además muestra cantidad (J) en blanco.</p>
<p>4. El usuario escoge medio (E).</p>	<p>5. El sistema actualiza en "Medios" el suministrador (F), las cajas de texto no. pieza (G), descripción (H) y precio (I) todos con los medio seleccionado en el filtro medio (E), además mantiene cantidad (J) en blanco.</p>
<p>6. El usuario escoge suministrador (F) e introduce cantidad (J). Presiona botón Agregar (K).</p>	<p>7. El sistema asigna nuevo medio al documento 711 en cuestión y actualiza los datos del grid (O).</p>
<p>8. El usuario selecciona una fila en el grid (O).</p>	<p>9. El sistema en "Medios" actualiza con detalles del medio seleccionado a medio (E) bloqueado, suministrador (F), no. pieza (G), descripción (H), precio (I) y cantidad (J).</p>
<p>10. Presiona botón Eliminar (M).</p>	<p>11. El sistema elimina medio del documento 711 en cuestión y actualiza los datos del grid (O).</p>

12. El usuario en "Medios" actualiza el suministrador (F), las cajas de texto no. pieza (G), descripción (H), precio (I) y cantidad (K) opcionalmente. Presiona Modificar (L).	13. El sistema actualiza datos del medio del documento 711 en cuestión y actualiza los datos del grid (O).
14. El usuario presiona botón Salir (N).	15. El sistema muestra Pantalla 2.

Sección: "Actualizar 711"

Actualizar 711

No	Modelo 711	Consumidor	Importador	Presentado	Aprobado

Pantalla 4

	1. El sistema muestra listado de todos los documentos 711 elaborados en el mes en curso en el grid (A).
2. El usuario presiona botón buscar (C).	3. Ver descripción del caso de uso extendido Buscar Documento.
4. El usuario selecciona documento 711 (B). Presiona botón Modificar (D).	5. El sistema muestra Pantalla 2 pasándole los datos del 711 seleccionado.
6. El usuario selecciona un documento 711 (B). Presiona botón Eliminar (E).	7. El sistema elimina el documento 711 y actualiza los datos del grid (A).
8. El usuario presiona botón Salir (F).	9. El sistema muestra Pantalla 1.
Poscondiciones:	La información referente a los documentos

	711 ha sido actualizada. Quedan auditados todos los eventos realizados por el usuario.
--	--

Tabla 6. Descripción del caso de uso Actualizar 711

Caso de Uso:	Actualizar Contratos
Actor(es):	Ejecutor CB.
Propósito:	Actualizaciones de cualquier índole sobre los contratos.
Resumen:	Este caso de uso se inicia cuando el actor necesita actualizar información sobre los contratos. De acuerdo a su requerimiento puede seleccionar tipo de actualización, si va registrar, modificar o eliminar un contrato.
Referencias:	RF2.
Casos de uso asociados	Buscar Documento (extensión)
Precondiciones:	El usuario levanta el sistema y se encuentra en la página principal. Debe haberse solicitado el servicio de autenticación. Para poder modificar o eliminar la información de los documentos contratos debe existir al menos uno ya elaborado.
Pantalla 1	
Acción del Actor	Respuesta del Sistema
1. El usuario elige la operación a realizar.	2. a) Si elige la opción Elaborar Contrato (A), ver sección: "Elaborar Contrato". b) Si elige la opción Actualizar Contrato (B), ver sección: "Actualizar Contrato"
Sección "Elaborar Contrato"	

The screenshot shows a web form titled "Elaborar Contrato" with a sub-header "Detalles [Contrato]". The form contains the following elements:

- No. Contrato:** A text input field with callout A.
- Fecha Firma:** A text input field with callout B.
- Descripción Medio:** A dropdown menu with callout C.
- Canje (Dolar US):** A text input field with callout D.
- Contratar Medios:** A section containing a "Contratar" button with callout E.
- Acceptar:** A button with callout F.
- Limplar:** A button with callout G.

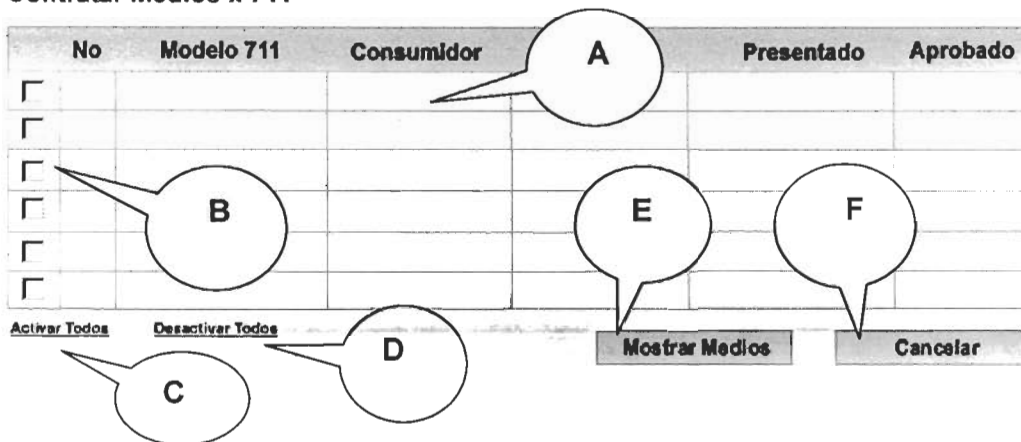
Pantalla 2

	<p>1. El sistema muestra las cajas de texto no. Contrato (A), fecha firma (B) y Canje (Dólar US) (D) vacías, así como el filtro descripción medio (C) con el primer valor de la lista ordenado alfabéticamente.</p> <p>Nota: Estos elementos obtendrán los valores de un documento contrato seleccionado si esta Pantalla 2 es llamada desde Pantalla 4.</p>
<p>2. El usuario especifica en las cajas de texto no. de contrato (A), fecha firma (B) y canje (Dólar US) (D) y escoge una opción en descripción medio (C). Presiona botón Contratar.</p>	<p>3. El sistema muestra Pantalla 3, ver sección "Contratar Medios 711".</p>
<p>4. El usuario especifica en las cajas de texto no. de contrato (A), fecha firma (B) y canje (Dólar US) (D) y escoge una opción en descripción medio (C). Presiona botón Aceptar.</p>	<p>5. El sistema registra nuevo documento contrato.</p>

<p>6. El usuario presiona botón Limpiar (F).</p>	<p>7. El sistema limpia las cajas de texto no. Contrato (A), fecha firma (B) y canje (Dólar US) (D) y posiciona el filtro descripción medio (C) en su primer valor.</p>
--	---

Sección "Contratar Medios 711"

Contratar Medios x 711



Pantalla 3

	<p>1. El sistema muestra en el grid (A) un listado con los documentos 711 elaborados en el mes en curso.</p>
<p>2. El usuario presiona botón Cancelar (F).</p>	<p>3. El sistema muestra Pantalla 2.</p>
<p>4. El usuario hace click en la acción Activar Todos (C).</p>	<p>5. El sistema selecciona todos los 711 mostrados en el grid (A) y activa las opciones (B), así como asigna color distinto al de por defecto.</p>
<p>6. El usuario hace click en la acción Desactivar Todos (D).</p>	<p>7. El sistema deselecciona todos los 711 mostrados en el grid (A) y desactiva todas las opciones (B) activadas, así como pone color de por defecto.</p>
<p>8. El usuario activa opción (B) en una fila determinada.</p>	<p>9. El sistema activa opción (B) en la fila determinada y pone color de fondo en la</p>

	misma distinto al de por defecto.
10. El usuario presiona botón Mostrar Medios (E).	11. El sistema muestra Pantalla 4 pasando los datos correspondientes, ver sección "Contratar Medios".

Sección "Contratar Medios"

Contratar Medios

Pantalla 4

	1. El sistema muestra en grid (A) los medios correspondientes a los documentos 711 seleccionados en Pantalla 3.
2. El usuario presiona botón Cancelar (F).	3. El sistema muestra Pantalla 3.
4. El usuario hace click en la acción Activar Todos (C).	5. El sistema selecciona todos los medios mostrados en el grid (A) y activa las opciones (B), así como asigna color distinto al de por defecto.
6. El usuario hace click en la acción Desactivar Todos (D).	7. El sistema deselecciona todos los medios mostrados en el grid (A) y desactiva todas las opciones (B) activadas, así como pone color de por defecto.
8. El usuario activa opción (B) en una fila determinada.	9. El sistema activa opción (B) en la fila determinada y pone color de fondo en la

	misma distinto al de por defecto.
10. El usuario presiona botón Contratar (E).	11. El sistema muestra Pantalla 2 y pasa los valores correspondientes.
Sección "Actualizar Contratos"	
<p>Actualizar Contratos</p> <p style="text-align: center;">Pantalla 5</p>	
	1. El sistema muestra listado de todos los contratos elaborados en el mes en curso en el grid (A).
2. El usuario presiona botón Buscar (C).	3. Ver descripción del caso de uso extendido Buscar Documento.
4. El usuario selecciona un contrato determinado (B). Presiona botón Modificar (D).	5. El sistema muestra Pantalla 2 pasándole los datos del contrato seleccionado.
6. El usuario selecciona un contrato determinado (B). Presiona botón Eliminar (E).	7. El sistema elimina contrato y actualiza los datos del grid (A).
8. El usuario presiona botón Salir (F).	9. El sistema muestra Pantalla 1.
Poscondiciones:	La información referente a los contratos ha sido actualizada. Quedan auditadas todas las operaciones realizadas por el usuario.

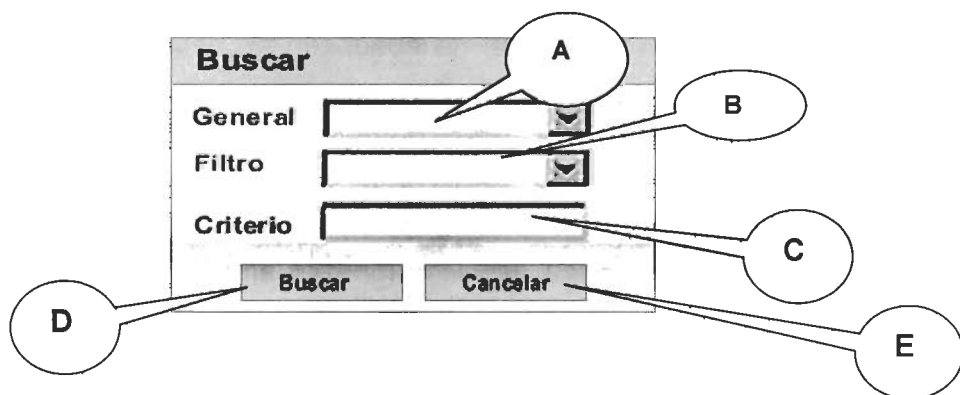
Tabla 7. Descripción del caso de uso Actualizar Contratos

Caso de Uso:	Buscar Documento
Actor(es):	Ejecutor CB
Propósito:	Realizar búsqueda de documentos (711 o contratos).
Resumen:	Este caso de uso se inicia cuando es el usuario elige la opción Buscar Documento desde el caso de uso de donde el esta extendido. El actor especifica los criterios de búsqueda definidos tipo de documento (General), Filtro y Criterio de Búsqueda. Se muestran los documentos obtenidos en el proceso de búsqueda en la interfaz que llama a este caso de uso.
Referencias:	RF4.
Precondiciones:	Debe haberse ejecutado el caso de uso del cual este caso de uso es una extensión . Debe haberse solicitado el servicio de autenticación.

Pantalla 1

Acción del Actor	Respuesta del Sistema
1. El usuario escoge operación a realizar.	2. Si elige opción Buscar Documento, ver sección "Buscar Documento".

Sección "Buscar Documento"



Pantalla 2

1. El sistema muestra la Pantalla 2 el filtro

	(A) deshabilitado con su valor por defecto en dependencia de que búsqueda ha sido activada: documentos 711 o contratos , el Filtro (B) con los parámetros de búsqueda correspondientes al filtro (A) mostrado y la caja de texto o filtro Criterio (C) en dependencia de el valor por defecto del filtro (B).
2. El usuario escoge opcionalmente en el filtro Filtro (B) y escribe en la caja de texto Criterio (C) o escoge en el filtro (C), esto en dependencia del control mostrado. Presiona botón Buscar (D).	3. El sistema busca resultados y le pasa los resultados a la Pantalla que la visualizó.
4. El usuario presiona botón Cancelar (E).	5. El sistema retorna a la Pantalla que la visualizo sin pasar resultados.
Poscondiciones:	

Tabla 8. Descripción del caso Buscar Documento

3.4. Conclusiones

En este capítulo comenzamos con el desarrollo de nuestra propuesta de solución, obteniéndose los principales artefactos resultantes del análisis de los procesos del negocio, captura de requisitos y la identificación de los principales elementos del sistema. Todo esto nos da pauta a comenzar con la construcción de nuestro sistema.



Construcción de la solución propuesta

En este capítulo se aborda sobre los principales aspectos del modelo del diseño. Dentro del mismo se hace referencia a la arquitectura aplicada, patrones de arquitectura, mecanismos de diseño y patrones de diseño, así como se muestra el resultado de los principales artefactos que genera este modelo. También se tocan aspectos sobre el modelo de datos, así como se tocan los puntos fundamentales del modelo de implementación y los principales artefactos resultantes. A continuación una muestra de lo anteriormente expuesto.

4.1 Diseño

Antes de dar un salto radical del análisis al diseño, flujo en el cual nos concentramos en obtener una visión del sistema, preocupándonos así por asegurar los requisitos funcionales; tenemos que decir que a partir de ahora el flujo de diseño definido por RUP en el cual nos encargaremos de refinar el análisis hecho concentrándonos en los requisitos no funcionales, para así obtener un producto de calidad. Por tanto en esta etapa trataremos de asegurar:

- ✓ Adquirir una comprensión de los aspectos relacionados con los requisitos no funcionales y restricciones relacionadas con los lenguajes de programación, componentes reutilizables, sistemas operativos,

tecnologías de distribución y concurrencia y tecnologías de interfaz de usuario.

- ✓ Crear una entrada apropiada y un punto de partida para actividades de implementación, capturando los requisitos o subsistemas individuales, interfaces y clases.
- ✓ Descomponer los trabajos de implementación en partes más manejables que puedan ser llevadas a cabo por diferentes equipos de desarrollo.
- ✓ Capturar las interfaces entre los subsistemas antes en el ciclo de vida del software.

Por tanto este flujo genera un conjunto de artefactos de los cuales abordaremos más adelante.

4.1.1 Arquitectura

En los inicios de la informática, la programación se consideraba un arte, debido a la dificultad que entrañaba para quienes la aplicaban, pero con el tiempo se han ido desarrollando metodologías y *fórmulas o trucos* para conseguir nuestros propósitos.

Todo lo anteriormente mencionado introduce un término indispensable para lograr la calidad de un producto software: arquitectura software.

La arquitectura software establece los fundamentos para que analistas, diseñadores, programadores, etc. trabajen en una línea común que permita alcanzar los objetivos y necesidades del sistema informático.

¿Por qué hablar de arquitectura software? En la medida que el tamaño de los sistemas crece, los algoritmos y las estructuras de datos dejan de convertirse en el mayor problema. El mayor reto es diseñar y especificar la estructura global del sistema.

Primeramente tenemos que saber *¿Qué queremos expresar con el término de arquitectura software?*

Muchos autores han tratado de definirlo, pero quiero resaltar que no tiene una definición común:

Pressman, "... descripción de subsistemas y componentes de un sistema informático y las relaciones entre ellos...").

Jerrold Grochow "... La arquitectura de un sistema constituye un amplio marco que describe su forma y su estructura, sus componentes y cómo estos interactúan..."

La arquitectura de software, tiene que ver con el diseño y la implementación de estructuras de software de alto nivel. Es el resultado de ensamblar un cierto número de elementos arquitectónicos de forma adecuada para satisfacer la mayor funcionalidad y requerimientos de desempeño de un sistema, así como requerimientos no funcionales, como la confiabilidad, escalabilidad, portabilidad, y disponibilidad.

Pero no solo al hablar de arquitectura podemos dejar de mencionar algo muy elemental: *los patrones de arquitectura*. Pero nos salta una primera pregunta ¿Que es un patrón?

Un patrón describe un problema que ocurre una y otra vez en nuestro entorno y describe también el núcleo de la solución al problema, de forma que puede reutilizarse continuamente.

¿A que responden los patrones de arquitectura?

- ✓ Son esquemas de organización de un sistema.
- ✓ Especifican una serie de subsistemas y sus responsabilidades.
- ✓ Incluyen reglas para organizar las relaciones entre ellos.

En la actualidad los patrones de arquitectura mas usados son:

- ✓ Patrón Modelo-Vista-Controlador (MVC).
- ✓ Patrón Capas (Layers).
- ✓ Patrón Distribución (Broker).

Para el caso nuestro aplicamos el patrón **Capas**, específicamente multicapas. A continuación una descripción detallada de cómo quedó estructurada la arquitectura.

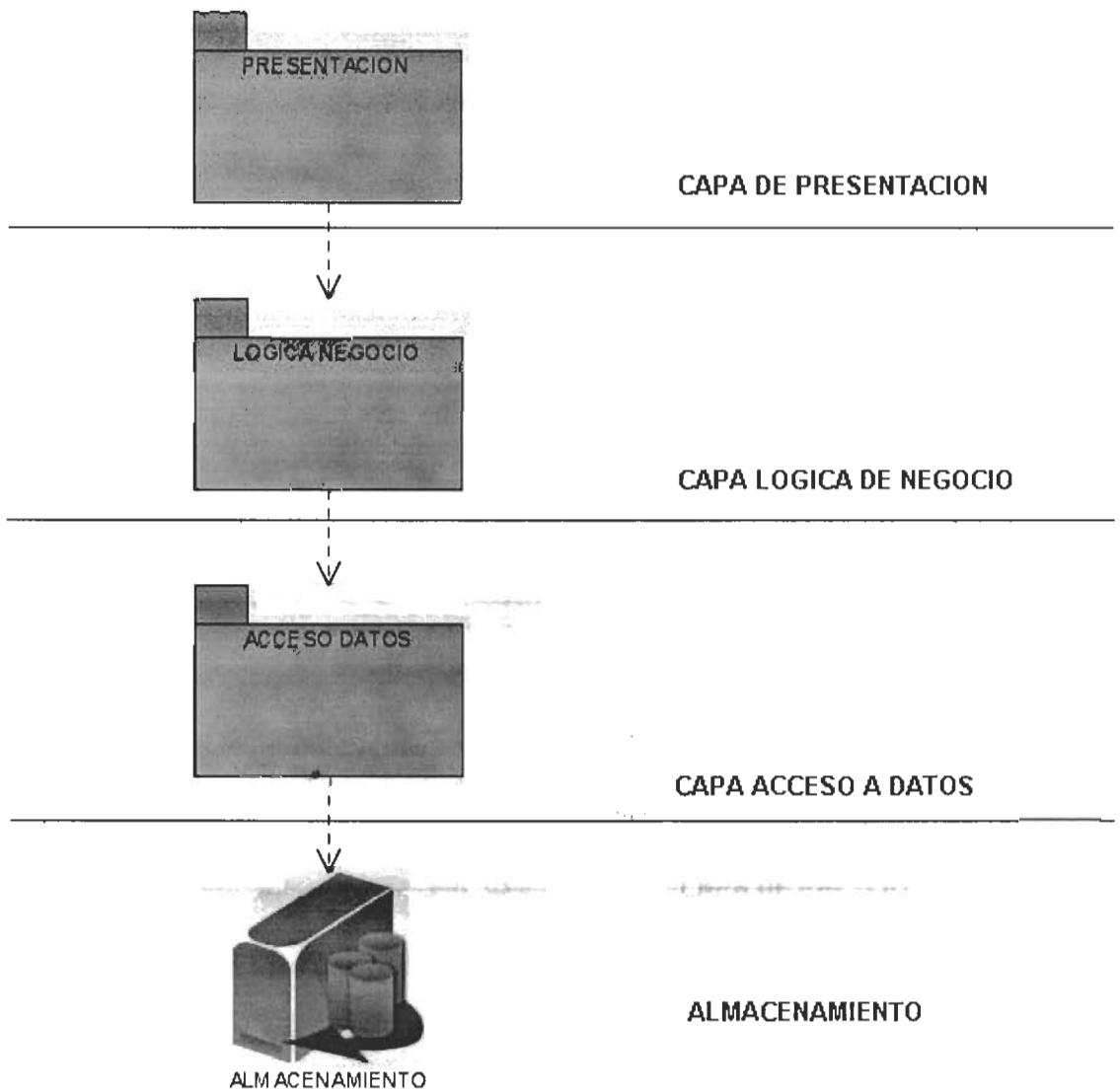


Fig. 4.1 Diagrama de paquetes de la arquitectura

4.1.2 ¿Por qué patrón Capas?

- ✓ Aislamiento de la lógica de la aplicación componentes separados reutilizables en otras aplicaciones.
- ✓ Distribución de capas en diferentes máquinas procesos, lo que puede mejorar el rendimiento, aumentar la coordinación y la comparación de información entre cliente y servidor.

- ✓ Dedicación de recursos a cada una de las y posibilidad de desarrollarlas en paralelo.

4.1.3 Descripción de las Capas

Capa Presentación

A la hora de adentrarnos en la descripción de esta capa debo decir que existen patrones definidos aplicables a la misma como:

- ✓ Vista de Plantillas (Template View)
- ✓ Transformación de Vistas (Transform View)
- ✓ Controlador de Páginas (Page Controller).
- ✓ Controlador al Frente (Front Controller).

El aplicado a este proyecto es el Vista Plantillas, el cual permite utilizar marcadores y separar (código script y HTML). Para facilitar esta separación y a su vez su procesamiento nos valemos del motor de plantillas php Smarty, el cual nos permite dar una mayor especialización del grupo de desarrollo.

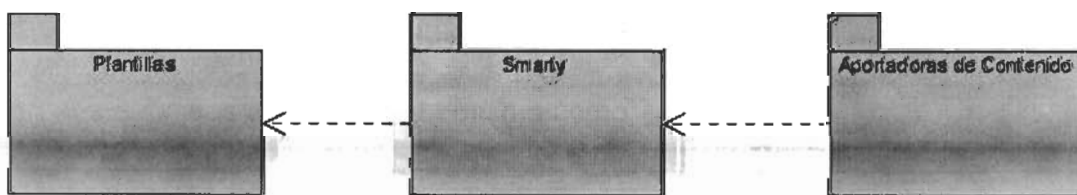


Fig. 4.2 Diagrama de subsistemas de Capa Presentación

En esta capa se evidencian los 3 paquetes:

- ✓ Plantillas (contiene todas las interfaces en código HTML y con las etiquetas para Smarty)
- ✓ Smarty (contiene las clases fundamentales brindadas por el motor de plantillas Smarty).

- ✓ Aportadoras de contenido (contienen todas las clases aportadoras de contenido a las plantillas).

Capa Lógica de Negocio

Para esta capa se aplican patrones definidos como:

- ✓ Modelo Dominio (Domain Module).
- ✓ Modulo Tabla (Table Module).
- ✓ Transaction Script .

El aplicado al caso nuestro es Modulo Tabla ya que recomendada al patrón Entrada Datos Tabla de la capa de Acceso Datos del cual abordaremos mas adelante.

Paquete Lógica Negocio

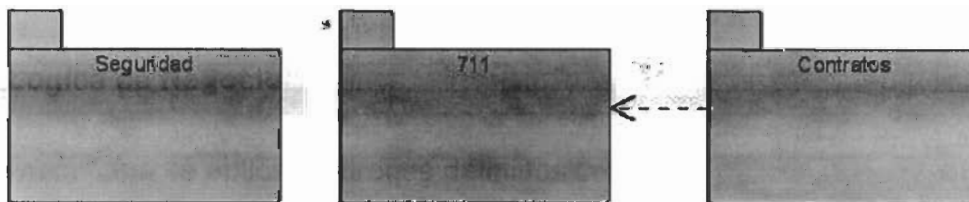


Fig. 4.3 Diagrama de subsistemas del Paquete Dominio

Este paquete contiene los siguientes subsistemas

- ✓ 711 (contiene las principales clases de lógica de negocio que aseguran la gestión de los documentos 711).
- ✓ Contratos ((contiene las principales clases de lógica de negocio que aseguran la gestión de los contratos).
- ✓ Seguridad (contiene todas las clases para brindar servicios de seguridad cliente al sistema).

Subsistema Seguridad

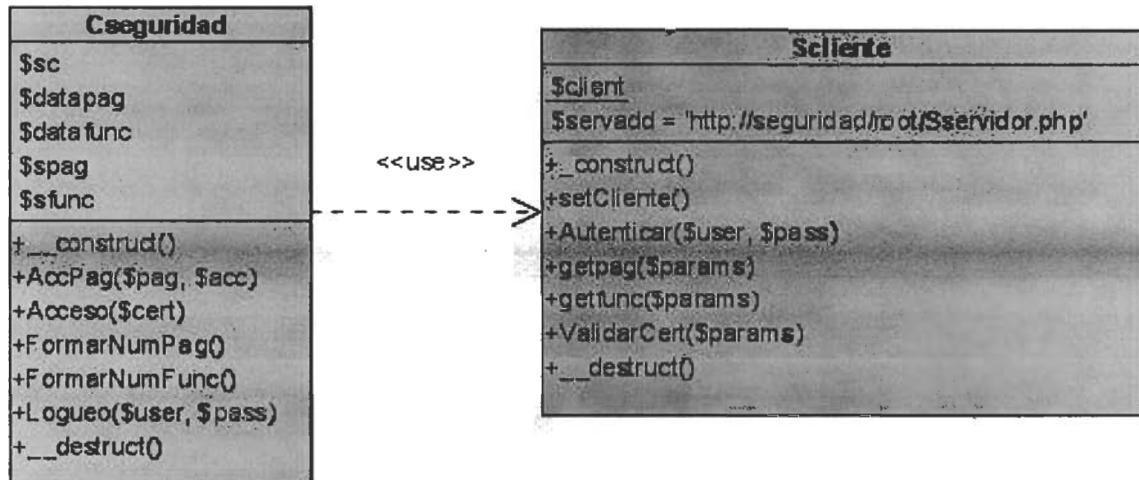


Fig. 4.4 Diagrama de clases Subsistema Seguridad

Paquete Acceso Datos



Fig. 4.5 Diagrama de los subsistemas del Paquete Servicios.

Este paquete contiene los siguientes subsistemas:

- ✓ Librería PHP (contiene las clases o bibliotecas brindadas por el entorno de desarrollo PHP).
- ✓ Acceso Datos (contiene todas las clases definidas para manejar la abstracción de la conexión y gestión con el gestor de base de datos).
- ✓ Típicas (contiene todas las clases típicas del sistema).

Para tener un mejor entendimiento de las clases involucradas en el paquete de Servicios, mostramos los diagramas de clases respectivos.

Subsistema Librería PHP

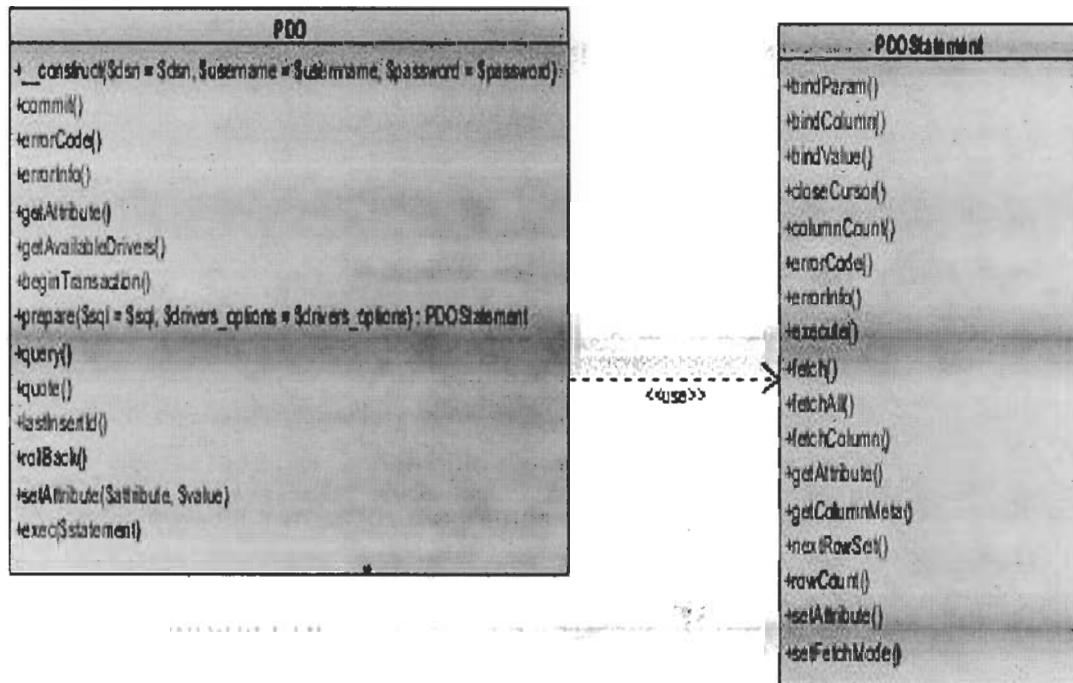


Fig. 4.6 Diagrama de clases Librería PHP

Subsistema Acceso a Datos

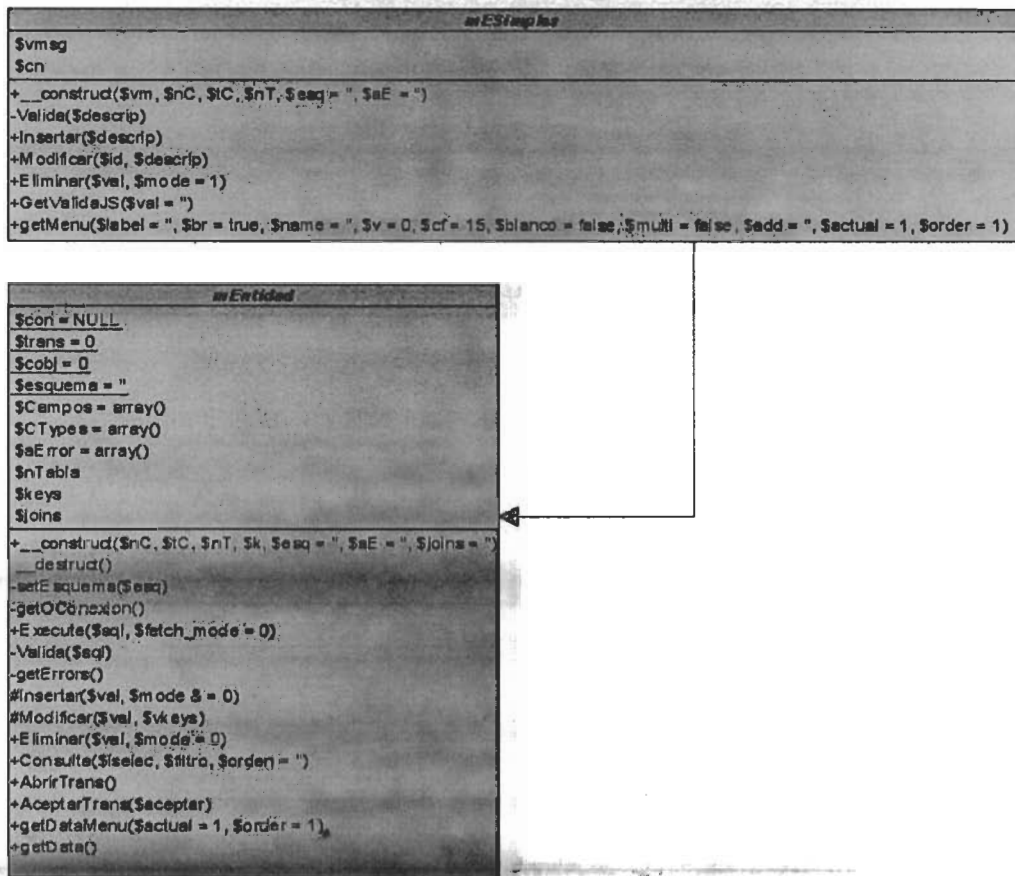


Fig. 4.7 Diagrama de clases Subsistema Acceso a Datos

Subsistema Típicas

Este subsistema encapsula todas las clases típicas, las cuales gestionan la manipulación directa con las tablas de la base de datos, así como la clase FactoriaTípicas. Para mas detalles ver **Anexos VI**.

Capa Almacenamiento

Esta capa solo representa a la base de datos relacional, es la capa mas baja y solo tiene las tablas de la base de datos con sus relaciones. **Ver Anexo VI**.

4.1.4 Patrones de Diseño

Como mismo abordamos los patrones de arquitectura también se define patrones de diseño enfocados directamente a los objetos principales resultantes del flujo de trabajo de análisis y que sobreviven en esta etapa de diseño.

Los patrones de diseño han contribuido a dar flexibilidad y extensibilidad a nuestros diseños. Además han demostrado ser una forma muy útil y exitosa de reutilizar diseño ya definidos, ya que ellos no sólo nombran, abstraen e identifican aspectos claves de estructuras comunes de diseño, sino que generalmente son descritos en una forma específica documental, haciendo su comprensión y aplicación fácil para el conjunto de desarrolladores.

Ahora podemos decir, un *patrón de diseño* es una descripción de clases y objetos comunicándose entre sí que resuelve un problema de diseño general en un contexto particular.

Un patrón de diseño identifica: *Clases, Instancias, Roles, Colaboraciones* y la *distribución de responsabilidades*.

En base a las responsabilidades asignadas se clasifican en

- ✓ Experto.
- ✓ Creador.
- ✓ Alta Cohesión.
- ✓ Bajo Acoplamiento.
- ✓ Controlador.

Dentro de los mismos se encuentran definidos una gran variedad de patrones, pero realmente cual usar, esa es nuestra principal polémica:

Problema: en nuestra capa de acceso a datos hemos hecho una abstracción de la misma para el manejo de las clases típicas de cada tabla de la base de datos. Necesitamos tener un objeto que se encargue de instanciar objetos de la misma clase, en este caso las propias típicas para lograr una centralización y simplificación de algunos artefactos resultantes del flujo de diseño como los diagramas de clases y diagramas de secuencia.

Solución: por las características antes descritas el patrón mas adecuado y por el que nos inclinamos es dentro de los creadores el patrón Factoría.

Patrón Factoría

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

La clase factoría devuelve una instancia de un objeto según los datos que se le pasan como parámetros.

Para el caso nuestro hicimos varias modificaciones, es decir esta factoría funciona recibiendo argumentos que determina que clase típica crear y a su vez registra un atributo publico en la misma con la instancia de la clase típica especificada.

A continuación un ejemplo de cómo quedo codificada esta clase:

```
class FactoriaTipicas
```

```
{
    public function __construct ()
    public function __destruct ()
    public function InsTip ($nameT, $nameI = 0, $argscT = 0)
}
```

La misma contiene un método InsTip el cual registra dinámicamente un atributo publico que hace referencia al clase típica especificada en \$nameT.

4.1.5 Mecanismo de Diseño de Acceso a Datos

Con el decursar de los años buscar mecanismos para modelar el acceso a datos siempre ha sido un meta a lograr, es por eso que un mecanismo aplicado para resolver esta situación nunca será igual a otro definido. Muchos son los especialistas que buscan un modelo ideal que sirva para todos los casos y se pueda emplear sin pensarlo dos veces, pero es una realidad que el desarrollo de hoy en día es tan vertiginoso que nos permite buscar soluciones tan buenas

como otras ya definidas, convirtiéndose a su vez en mecanismos aplicables bajo ciertas circunstancias. Como todos sabemos el acceso y la manipulación de los datos es algo realmente indispensable a la hora de desarrollar un sistema informático.

Por todo esto a raíz de la aparición del concepto de patrones, se han definido algunos que abordan esta problemática., por lo cual nos resulta un punto de partida a la hora de modelar el “acceso a datos”.

¿Por qué definir un mecanismo para el acceso a datos?

Para acceder a los datos siempre están involucrados los mismos objetos y se efectúan un conjunto de operaciones comunes en las realizaciones de algunos casos de uso. De todo esto la necesidad de documentar un mecanismo que simplifique el modelado y que quede como punto de referencia para los desarrolladores. Todo esto nos permitirá obtener diagramas más entendibles, que nos permita una mayor comunicación con nuestro equipo de desarrollo. Pero lo más importante es que nos trazara una línea común, una política a seguir, fomentando algo muy indispensable para lograr eficiencia, la *reutilización*.

Para nuestro sistema en cuestión se plantea el siguiente mecanismo de diseño para modelar el “acceso a datos”.

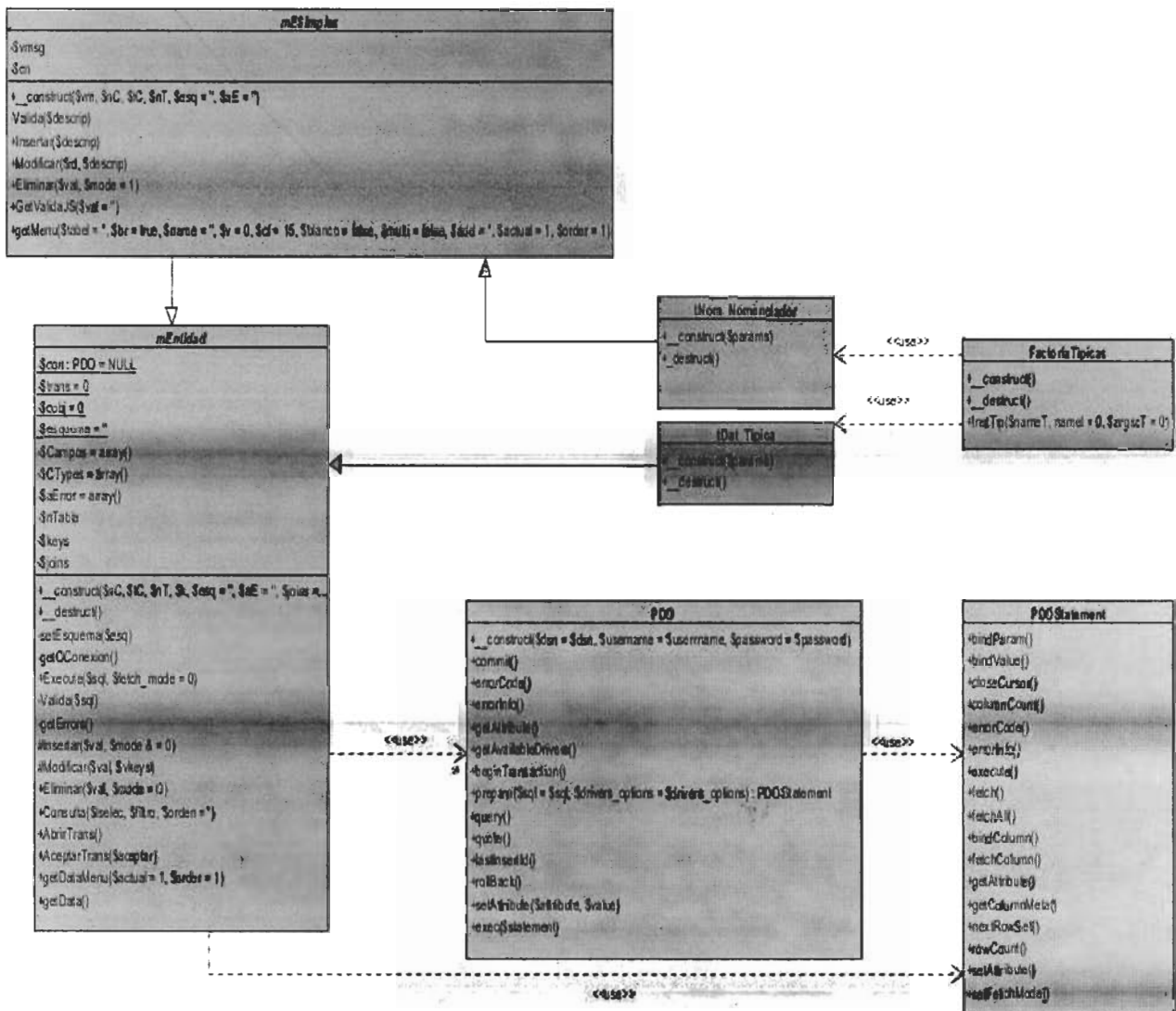


Fig. 4.8 Vista estática del mecanismo de diseño para persistencia. Conectividad usando entorno desarrollo brindado por el lenguaje PHP.

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

Para dar la responsabilidad a una clase que encapsulará las instancias de estos objetos se definió la clase *FactoriaTipicas* ya antes mencionada en el patrón de diseño aplicado a este funcionamiento.

4.1.6 Mecanismo de Diseño de Acceso Seguridad

La seguridad es un aspecto crítico de las aplicaciones Web. Las aplicaciones Web, por definición, permiten el acceso de usuarios a recursos centrales, el servidor Web y, a través de éste, a otros como los servidores de base de datos. Con los conocimientos y la implementación correcta de medidas de seguridad, puede proteger sus propios recursos así como proporcionar un entorno seguro donde los usuarios trabajen cómodos con su aplicación.

El proceso administrativo centralizado de la seguridad en los sistemas es un elemento fundamental para un control riguroso de los accesos a las aplicaciones Web, dado que es más fácil actual ante cualquier violación.

Los sistemas realizan el control de la seguridad de manera semejante, usando un servicio Web encargado del control de los accesos, autenticación y registro de los eventos que ocurren, por lo cual se propone un mecanismo de diseño que sirva de manera general a todas las aplicaciones que usan dicho servicio, garantizando así los requerimientos necesarios para su correcto funcionamiento.

Básicamente los servicios Web permiten que diferentes aplicaciones, realizadas con diferentes tecnologías, y ejecutándose en toda una variedad de entornos, puedan comunicarse e integrarse, lo cual es muy importante.

Por lo explicado anteriormente se propone el siguiente mecanismo de diseño para seguridad basado en el uso de servicios Web.

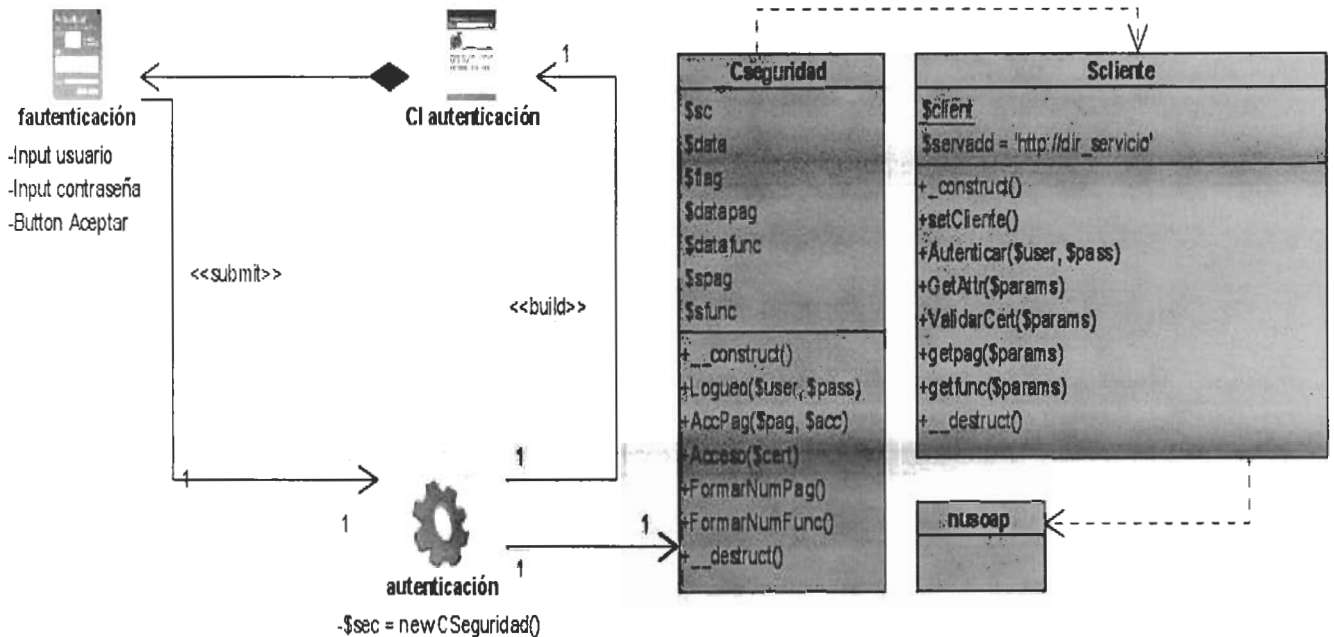


Fig. 4.9 Vista estática del mecanismo de seguridad usando entorno de desarrollo PHP.

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

4.1.7 Diagramas de Clases del Diseño

Para poder llevar a cabo una realización de caso de uso del diseño, es necesario plasmar una descripción de **los principales** flujos de eventos de

manera textual, así como diagramas de clases que muestren sus clases de diseño participantes y diagramas de interacción que muestran la realización de un flujo o escenarios concretos de un caso de uso en términos de interacción entre objetos del diseño.

Antes de adentrarnos en el tema tenemos que decir que lo que estamos modelando es una aplicación Web por lo que la concepción de modelado de las clases del diseño están extendidas por el lenguaje UML. Para una mejor comprensión se concentra en detalles importantes como la separación de que parte se ejecuta del lado cliente y la que esta del lado del servidor.

Podemos mencionar que esto diagramas son el resultado de la aplicación de los mecanismos de diseño (**Acceso a Datos y Seguridad**) antes mencionados.

Explicación de un diagrama de clases del diseño:

Para nuestro sistema por cada caso de uso tenemos concebido lo siguiente:

- ✓ Una clase de lógica de negocio (**InNombreCU**) por cada caso de uso.
- ✓ Una clase aportadora de contenido para todas las páginas clientes involucradas (**acNombreCU**). La misma tendrá un método (**getIntPaginaCliente1, getIntPaginaCliente2... getIntPaginaClienteN**) donde n es la cantidad de páginas clientes asociadas al caso de uso
- ✓ Una clase **GLOBAL** que centraliza al objeto principal para manejar la seguridad : **Cseguridad**.
- ✓ Un paquete **Smarty** que gestiona a las plantillas, donde tendrá asociado tantas plantillas como páginas clientes tenga asociado el caso de uso.
- ✓ La página servidora tendrá las instancias de los objetos **InNombreCU** y **acNombreCU**.

Para entender el funcionamiento de la seguridad y el acceso a datos ver los mecanismos antes mencionados.

A continuación solo mostraremos los diagramas de clases del diseño para los casos de uso principales Actualizar 711, Actualizar Contratos y Buscar Documento.

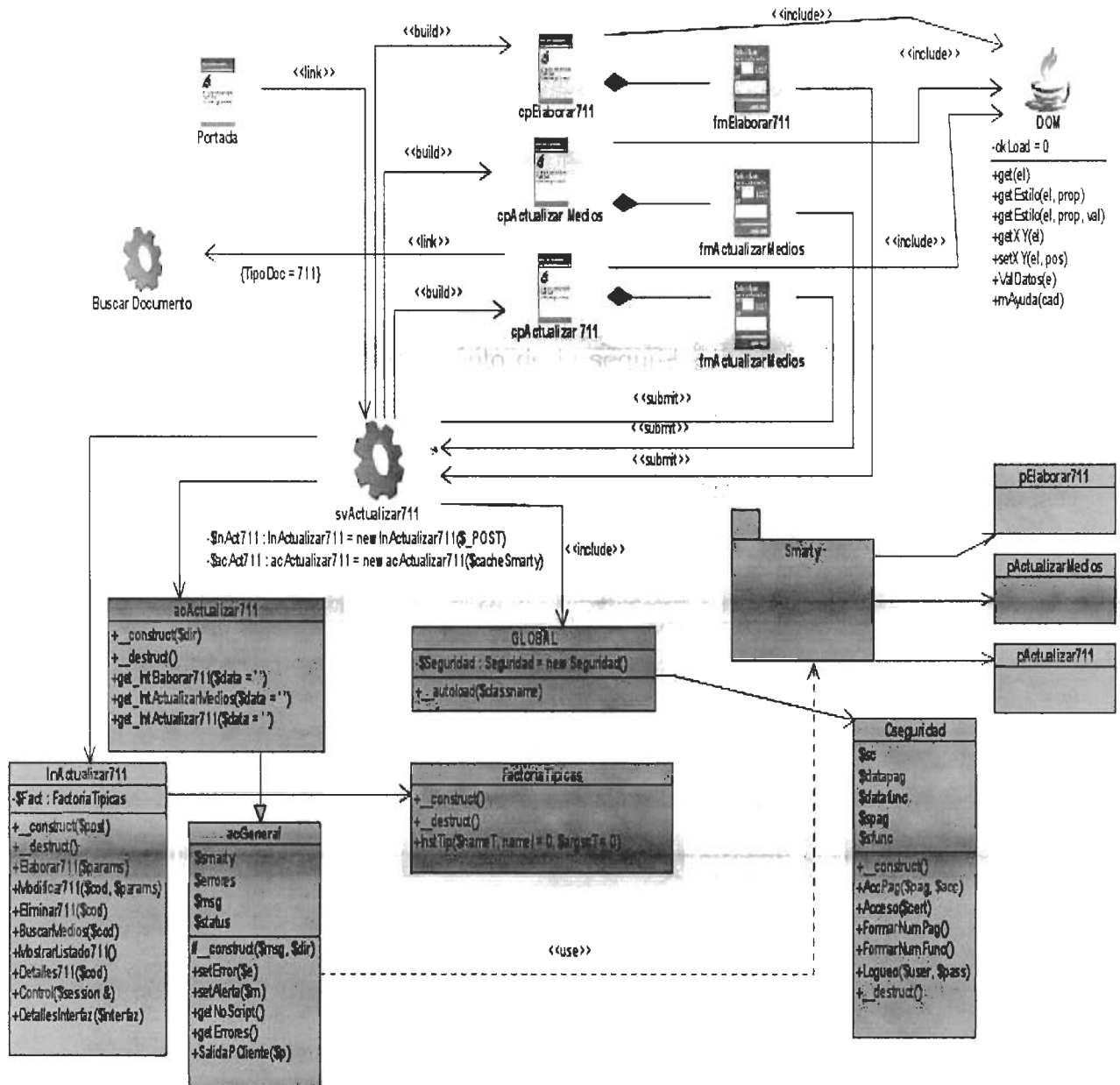


Fig. 4.10 Diagrama CU Actualizar 711

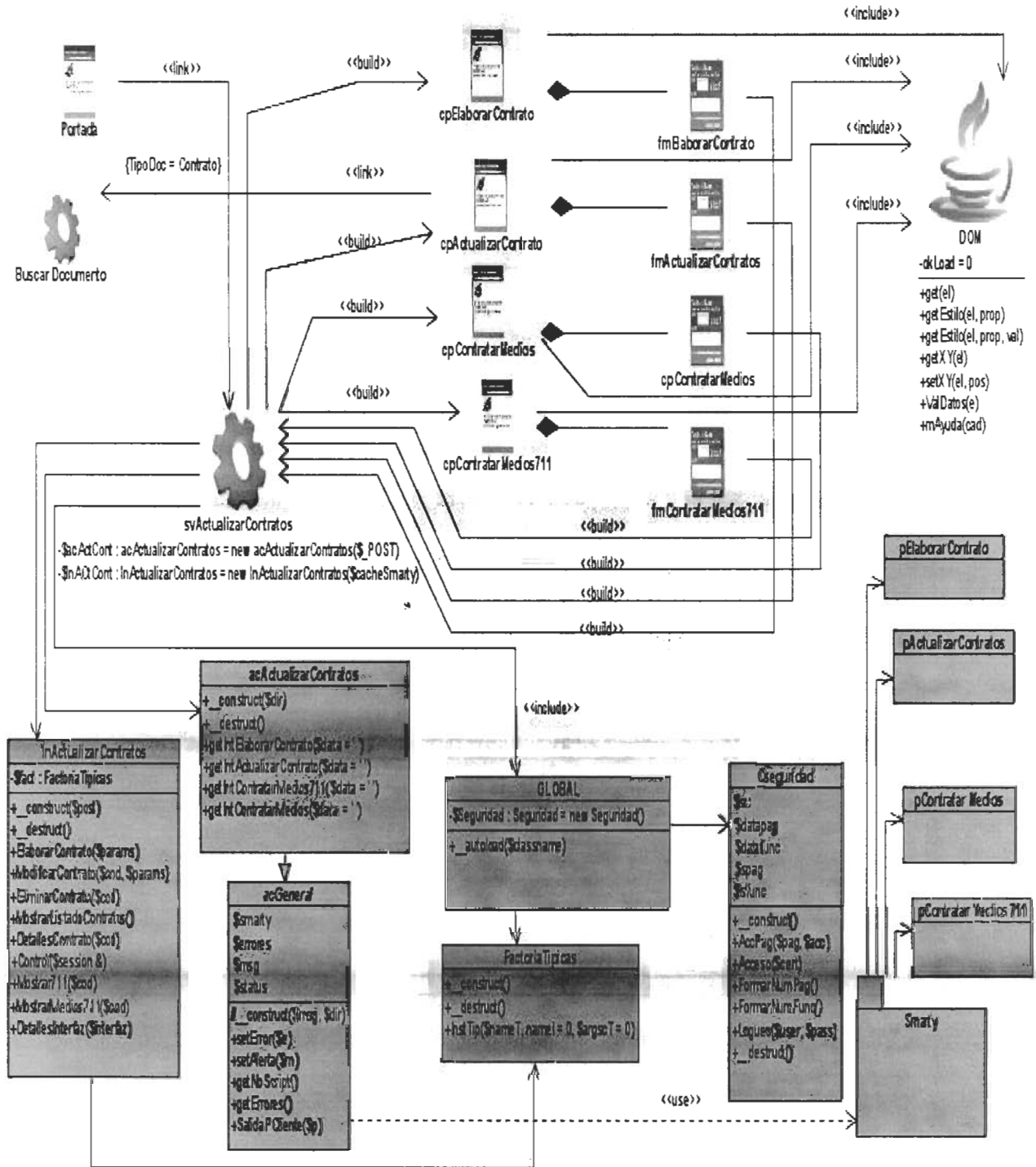


Fig. 4.11 Diagrama CU Actualizar Contratos

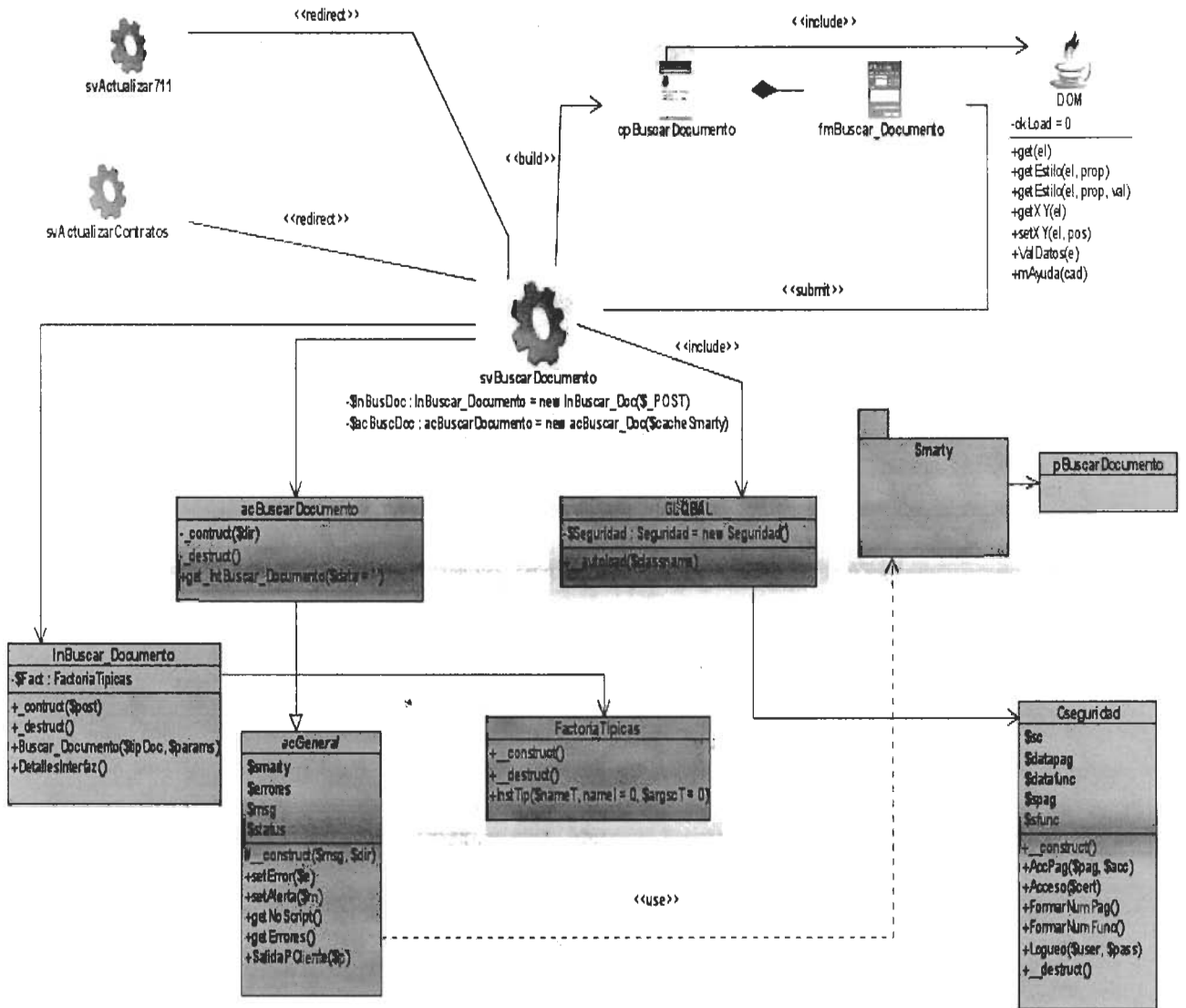


Fig. 4.12 Diagrama CU Buscar Documento

4.1.8 Principios de Diseño

Antes de adentrarnos en los principios de diseño tenemos que tener bien claros para quien están dirigidos los mismos: para los usuarios.

Los principios generales de diseño aplicables a las aplicaciones Web y los que tuvimos en cuenta para el desarrollo de nuestra aplicación son los siguientes:

✓ **Uso equiparable**

Pautas para este principio: Que proporcione las mismas maneras de uso para todos los usuarios. Que evite segregar o estigmatizar a cualquier usuario. Las características de privacidad, garantía y seguridad deben estar igualmente disponibles para todos los usuarios. Que el diseño sea atractivo para todos los usuarios.

✓ **Uso flexible**

Pautas para este principio: Que ofrezca posibilidades de elección en los métodos de uso. Que pueda accederse y usarse tanto con la mano derecha como con la izquierda. Que facilite al usuario la exactitud y precisión. Que se adapte al paso o ritmo del usuario.

✓ **Simple e intuitivo**

Pautas para el principio: Que elimine la complejidad innecesaria. Que sea consistente con las expectativas e intuición del usuario. Que se acomode a un amplio rango de alfabetización y habilidades lingüísticas. Que dispense la información de manera consistente con su importancia. Que proporcione avisos eficaces y métodos de respuesta durante y tras la finalización de la tarea.

✓ **Información perceptible**

Pautas para el principio: Que use diferentes modos para presentar de manera redundante la información esencial (gráfica, verbal o táctilmente). Que proporcione contraste suficiente entre la información esencial y sus

alrededores. Que amplíe la legibilidad de la información esencial. Que diferencie los elementos en formas que puedan ser descritas (por ejemplo, que haga fácil dar instrucciones o direcciones). Que proporcione compatibilidad con varias técnicas o dispositivos usados por personas con limitaciones sensoriales.

✓ **Con tolerancia al error**

Pautas para el principio: Que disponga los elementos para minimizar los riesgos y errores: elementos más usados, más accesibles; y los elementos peligrosos eliminados, aislados o tapados. Que proporcione advertencias sobre peligros y errores. Que proporcione características seguras de interrupción. Que desaliente acciones inconscientes en tareas que requieren vigilancia.

✓ **Que exija poco esfuerzo físico.**

Pautas para el principio: Que permita que el usuario mantenga una posición corporal neutra. Que utilice de manera razonable las fuerzas necesarias para operar. Que minimice las acciones repetitivas. Que minimice el esfuerzo físico continuado.

✓ **Tamaño y espacio para el acceso y uso**

Pautas para el principio: Que proporcione una línea de visión clara hacia los elementos importantes tanto para un usuario sentado como de pie. Que el alcance de cualquier componente sea cómodo para cualquier usuario sentado o de pie. Que se acomode a variaciones de tamaño de la mano o del agarre. Que proporcione el espacio necesario para el uso de ayudas técnicas o de asistencia personal.

4.1.9 Estándares

A continuación se muestra los principales estándares aplicados:

Estándar de Codificación

<p>Apariencia de clases y objetos</p>	<p>Primera letra en mayúscula</p>	<p>Los nombres de las clases y las instancias de las mismas deben comenzar con la primera letra en mayúscula y el resto en minúscula, en caso de que sea un nombre compuesto se empleará notación PascalCasing*. Ejemplo: MiClase ().</p>
<p>Nombre de clases y objetos</p>	<p>Relacionados al propósito</p>	<p>El nombre empleado, debe permitir que con sólo leerlo se conozca el propósito de la clase o instancia de la misma. Para el caso de las instancias es recomendable que se denoten así: Para la clase: Nomcliente su instancia será \$Ocliente, de forma tal que la primera letra indique que es un objeto y el resto, la clase a la que pertenece.</p>
<p>Apariencia de atributos</p>	<p>Primera letra en minúscula</p>	<p>El nombre que se le da a los atributos de las clases debe comenzar con la primera letra en minúscula, en caso de que sea un nombre compuesto se empleará notación CamelCasing**.</p>
<p>Nombre de atributos</p>	<p>Nemotécnicos</p>	<p>El nombre empleado, debe permitir que con sólo leerlo se conozca el propósito del mismo dentro de la clase. Ejemplo: \$nTabla, este atributo denota el nombre de una tabla.</p>

Apariencia de las funciones	Primera letra en mayúscula	Los nombres de las funciones deben comenzar con la primera letra en mayúscula y el resto en minúscula, en caso de que sea un nombre compuesto se empleará notación PascalCasing*. Si son funciones que obtienen un dato se emplea el prefijo get y si fijan algún valor se emplea el prefijo set .
Nombre de las funciones	Nemotécnicos	El nombre empleado, debe permitir que con sólo leerlo se conozca el propósito de la misma dentro de la clase.
Declaración de parámetro en funciones	Agrupados por tipos primero los string, los numéricos y valores por defecto.	Los parámetros que se le pasan a las funciones se recomienda sean declarados de forma tal que estén agrupados por el tipo de dato que contienen. Ejemplo: BuscaUnidad (\$nTabla(string), \$nCampos(string), \$kIndice(entero)).
Variables y constantes		
Apariencia de constantes	Todas sus letras en mayúscula	Se deben declarar las constantes con todas sus letras en mayúscula.
Nombres de las variables y constantes	Nemotécnicos	El nombre empleado, debe permitir que con sólo leerlo se conozca el propósito de la misma. Ejemplo: \$nFields.
Declaración de constantes y asignación a variables	Una por cada línea	Se recomienda declarar una constante por cada línea y con las asignaciones a las variables sucede lo mismo. Ejemplo: define("CONSTANT1", "value1"); define("CONSTANT2", "value2");

		\$nTabla='nomproducto'; \$kIndice=0;
Identación		
Objetivo: Lograr una estructura uniforme para los bloques de código así como para los diferentes niveles de anidamiento.		
0 espacios en blanco desde la izquierda en	Require Include Class	No se empleará ningún espacio en blanco desde la izquierda para las instrucciones antes mencionadas. Se tomará como inicio de la página el tag PHP <?
2 espacio en blanco desde la izquierda en	Function Define	Se dejarán dos espacios en blanco desde la izquierda en las instrucciones antes mencionadas.
2 espacio en blanco desde la referencia en	Inicio y fin de bloque	Se recomienda dejar dos espacios en blanco desde la instrucción anterior para el inicio y fin de bloque {}. Lo mismo sucede para el caso de las instrucciones If, else, For, While, Do While, Switch, Foreach.
Niveles de anidación	Hasta 5 niveles	Se recomienda emplear hasta 5 niveles de anidación en instrucciones If, For, While.
Ejemplo de identación		
<pre> <? require ('class/Interface.php'); class MiClase { function BuscaUnidad(\$nTabla, \$nFields, \$kIndice) { if (\$nTabla) { ... } } } </pre>		

<pre> } for (...) { ... } } } ?> </pre>		
Comentarios, separadores, líneas y espacios en blanco		
Objetivo: Establecer un modo común para comentar el código de forma tal que sea comprensible con sólo leerlo una vez.		
Ubicación de comentarios	Al inicio de cada clase o función y al final de cada bloque de código.	Se recomienda comentar al inicio de la clase o función especificando el objetivo de la misma así como los parámetros que usa (especificar tipos de dato, y objetivo del parámetro) entre otras cosas. Y se comenta también cuando se cierran los ciclos, clases, instrucciones if y otras.
Separador de instrucciones	Se emplea el punto y coma.	Se recomienda usar el separador al final de cada instrucción y no en la línea de abajo. Ejemplo: define ("CONSTANT", "value1");
Líneas en blanco	Se emplean antes de cada función.	Se recomienda dejar una línea en blanco antes de la definición de cada función para dar claridad al código.
Espacios en blanco	Entre operadores lógicos y aritméticos.	Se recomienda usar espacios en blanco entre estos operadores para lograr una mayor legibilidad en el código. Ejemplo: \$nTabla = 'nomproducto'; if ((\$nTabla) && (\$nFields))

Tabla 9. Estándar de codificación

Estándares para la BD

Apariencia de la BD	Primera letra en mayúscula	Los nombres de las BDs deben comenzar con la primera letra en mayúscula y el resto en minúscula, en caso de que sea un nombre compuesto se empleará notación PascalCasing*.
Nombres de las BDs	Nemotécnicos y relacionados al propósito.	El nombre empleado, debe permitir que con sólo leerlo se conozca el propósito de la misma.
Apariencia de los esquemas	Todas las letras en minúscula.	El nombre a emplear para los esquemas debe escribirse con todas las letras en minúscula para evitar problemas con el Case Sensitive del gestor. Ejemplo: create schema 'finanzas';
Nombres de los esquemas	Nemotécnicos y relacionados al propósito.	El nombre empleado, debe permitir que con sólo leerlo se conozca el propósito del mismo.
Apariencia de las tablas	Todas las letras en minúscula.	El nombre a emplear para las tablas debe escribirse con todas las letras en minúscula para evitar problemas con el Case Sensitive del gestor.

		<p>Ejemplo: create table 'nom_producto';</p>
Nombres de las tablas	<p>Nemotécnicos y relacionados al propósito. Además clasificando las tablas por su tipo.</p>	<p>El nombre empleado, debe permitir que con sólo leerlo se conozca el propósito del mismo. Se deben clasificar las tablas por su tipo, es decir por los datos que contienen se le coloca un prefijo, que se puede clasificar en:</p> <p>Ejemplo: Nomencladores nom_... Auxiliares aux_... Datos dat_... Históricas his_... Seguridad seg_... Temporales tmp_... Configuración cfg_...</p>
Apariencia de los campos	<p>Todas las letras en minúscula.</p>	<p>El nombre a emplear para los campos debe escribirse con todas las letras en minúscula para evitar problemas con el Case Sensitive del gestor.</p> <p>Ejemplo: add field 'idproducto';</p>
Nombre de los campos	<p>Nemotécnicos En caso de identificadores,</p>	<p>El nombre empleado, debe permitir que con sólo leerlo se conozca el propósito del</p>

	emplear id, este sería igual en la tabla de datos que lo emplea.	mismo. Además se debe incluir un comentario en la descripción del mismo.
Nombre de las llaves primarias	Nemotécnicos empleando prefijos.	Se nombrarán las llaves primarias de forma que se vea de qué tabla es y que es primaria. Ejemplo: pk_cuenta. (Llave primaria de la tabla cuenta). Si es una llave compuesta se coloca el prefijo y en nemotécnico los campos que la forman.
Nombre de las llaves foráneas.	Nemotécnicos empleando prefijos.	Se nombrarán las llaves foráneas de forma que se vea de qué tabla es y que es foránea. Ejemplo: fk_cuenta. (Llave foránea de la tabla cuenta). Si es una llave compuesta se coloca el prefijo y en nemotécnico los campos que la forman.
Nombre de las secuencias	Nemotécnicos empleando prefijos.	Se nombrarán las secuencias de forma que se vea de qué campo es y que es una secuencia. Ejemplo: seq_idcuenta. (Secuencia del campo idcuenta).

Restricciones Únicas y de Chequeo	Nemo­técnicos empleando prefijos.	Ejemplo: (u_ o c_) + nombre del campo que la emplea.
Nombres de las funciones, triggers, y vistas	Prefijos + Nemo­técnicos	El nombre empleado, debe permitir que con sólo leerlo se conozca el propósito del mismo. Ejemplo: ft_ Funciones de triggers.

Tabla 10. Estándar de diseño de la Base Datos

4.2. Modelo Lógico y Físico de los Datos

Para llevar a cabo el diseño de la base de datos, se identificaron las principales clases persistentes y se establecieron las relaciones entre las mismas, todo esto nos dio la posibilidad de establecer el modelo lógico de la base de datos y su vez por las facilidades del Visual Paradigm general el modelo físico o diagrama entidad-relación. Tenemos que decir que estas clases persistentes tienen un gran peso en la aplicación ya que nos permiten hacer persistir la información manipulada por la lógica de negocio. A continuación una muestra de lo antes mencionado:

Modelo Lógico

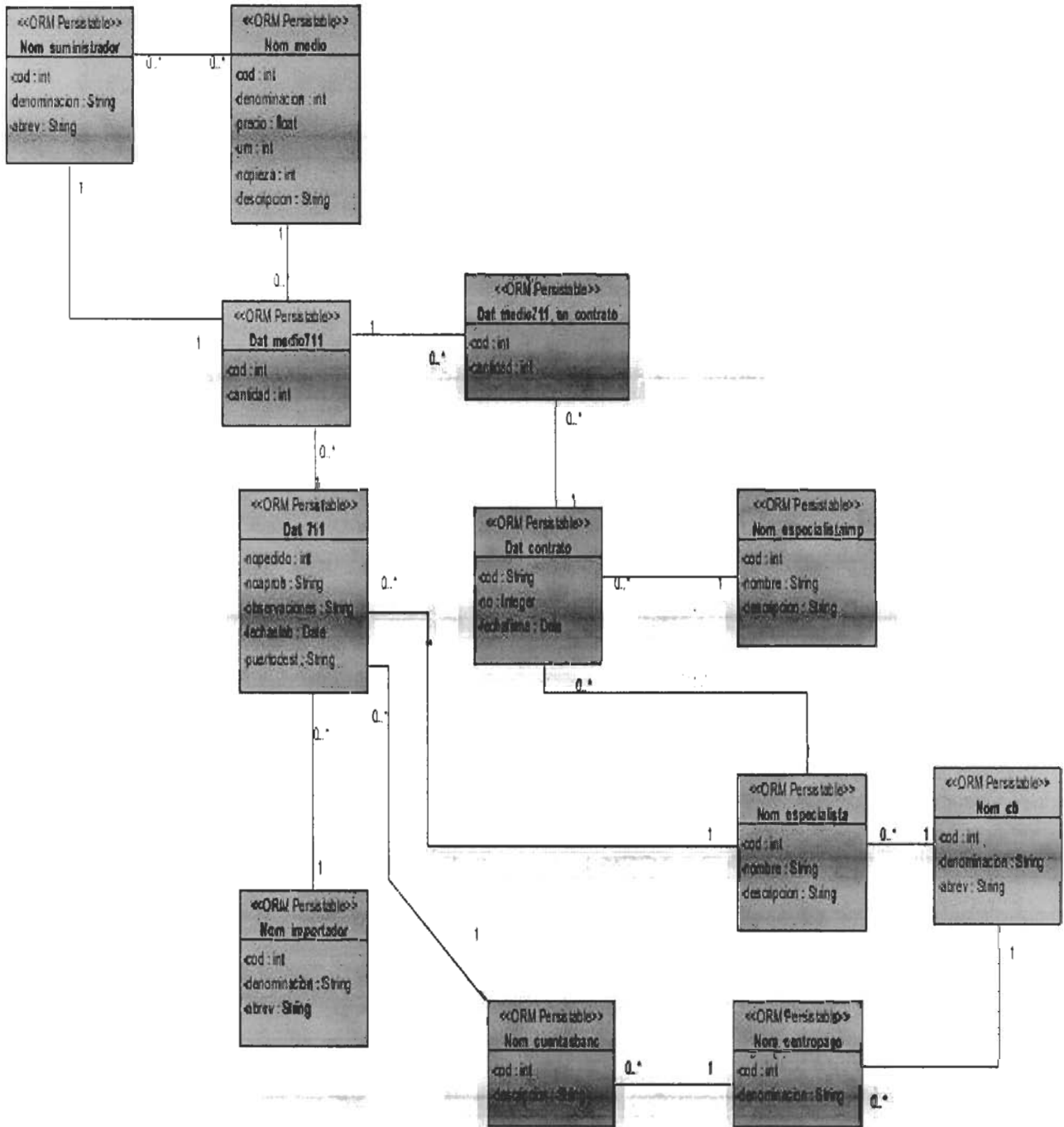


Fig. 4.13 Diagrama de clases persistentes

Modelo Físico

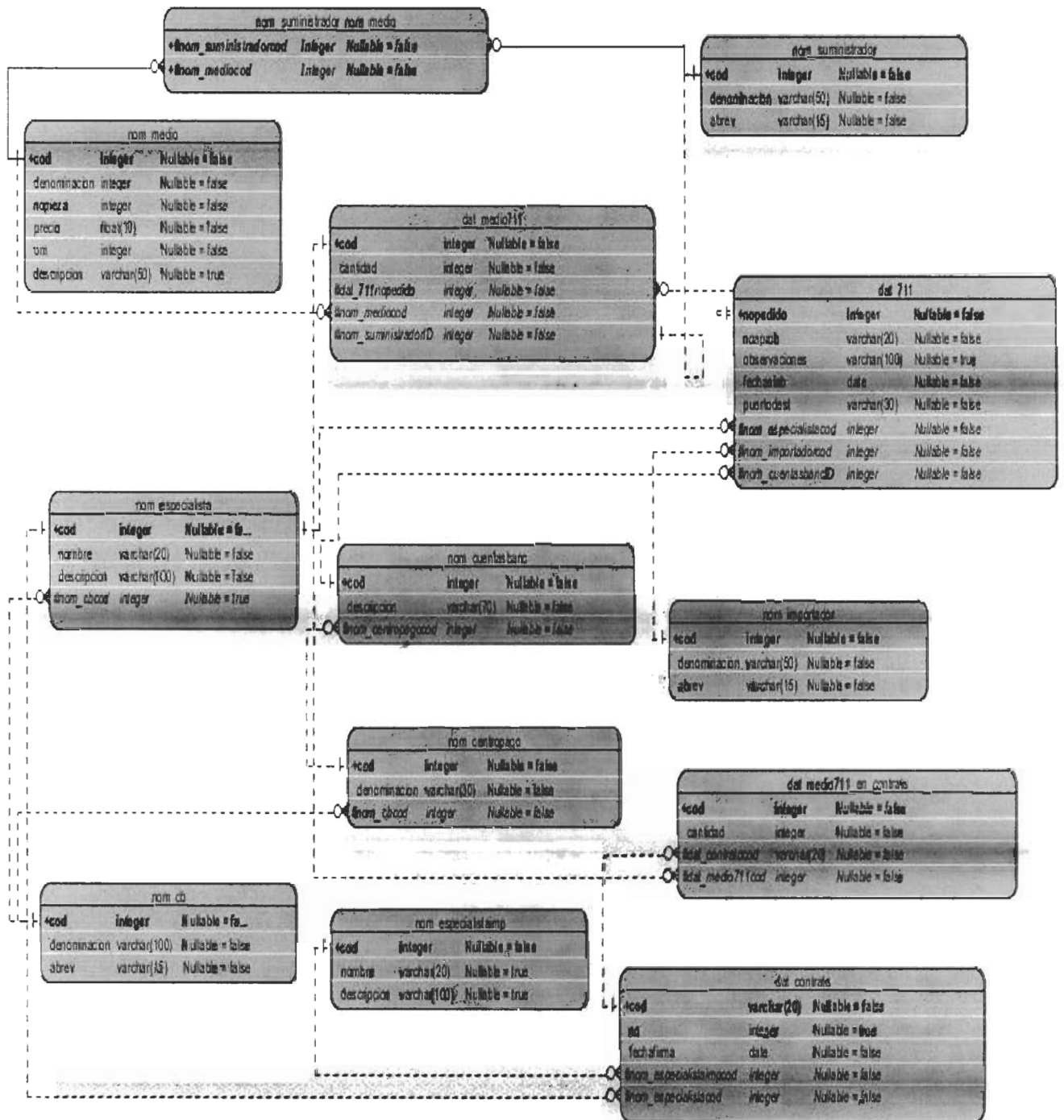


Fig. 4.14 Diagrama entidad-relación

4.3 Modelo de Despliegue.

Podemos mencionar que el diagrama de despliegue presenta la arquitectura de tiempo de ejecución de los procesadores, dispositivos, y los componentes de software que se ejecutan en esa arquitectura. Es la descripción física última de la topología del sistema, y describe la estructura de las unidades del hardware y el software que se ejecuta en cada unidad. En una arquitectura así, debe ser posible mirar un nodo específico de la topología, ver que componentes se ejecutan en el nodo, ver que elementos lógicos están implementados en el componente, y finalmente trazar esos elementos al análisis de requerimientos inicial del sistema.

Para nuestro caso existen tres escenarios en que puede desplegarse nuestro sistema:

- ✓ **Modelo Ideal:** tener un servidor Web de aplicaciones, un servidor de Base de Datos y muchos nodos clientes.
- ✓ **Modelo Semi-Ideal:** tener un mismo servidor Web y de Base de Datos y muchos nodos clientes.
- ✓ **Modelo Crítico:** tener la aplicación y la base de datos instalada en el mismo nodo cliente.

A continuación se muestra lo anteriormente mencionado:

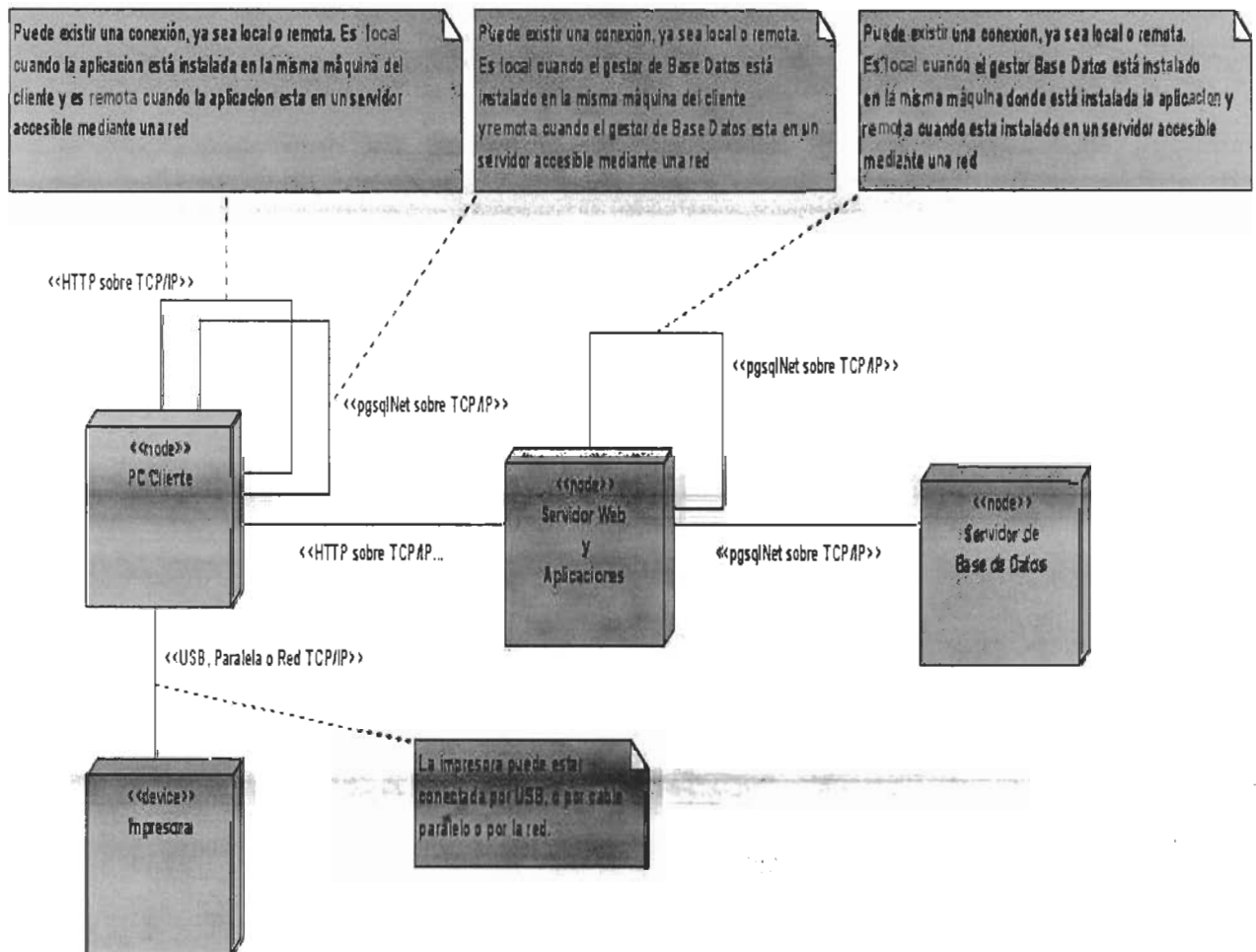


Fig. 4.15 Diagrama de despliegue

4.4 Modelo de Implementación

Podemos decir que el diagrama de componentes es el artefacto principal resultante del modelo de implementación, describe los componentes de software y sus dependencias entre si, representando la estructura del código. Los componentes son la implementación en la arquitectura física de los conceptos y la funcionalidad descrita en la arquitectura lógica (clases, objetos, sus relaciones, y las colaboraciones). Los componentes son típicamente los archivos de implementación en el ambiente de desarrollo.

A continuación una muestra de lo anteriormente mencionado enfocado a los componentes fuentes, componentes de tiempo de ejecución y componentes de tiempo de enlace:

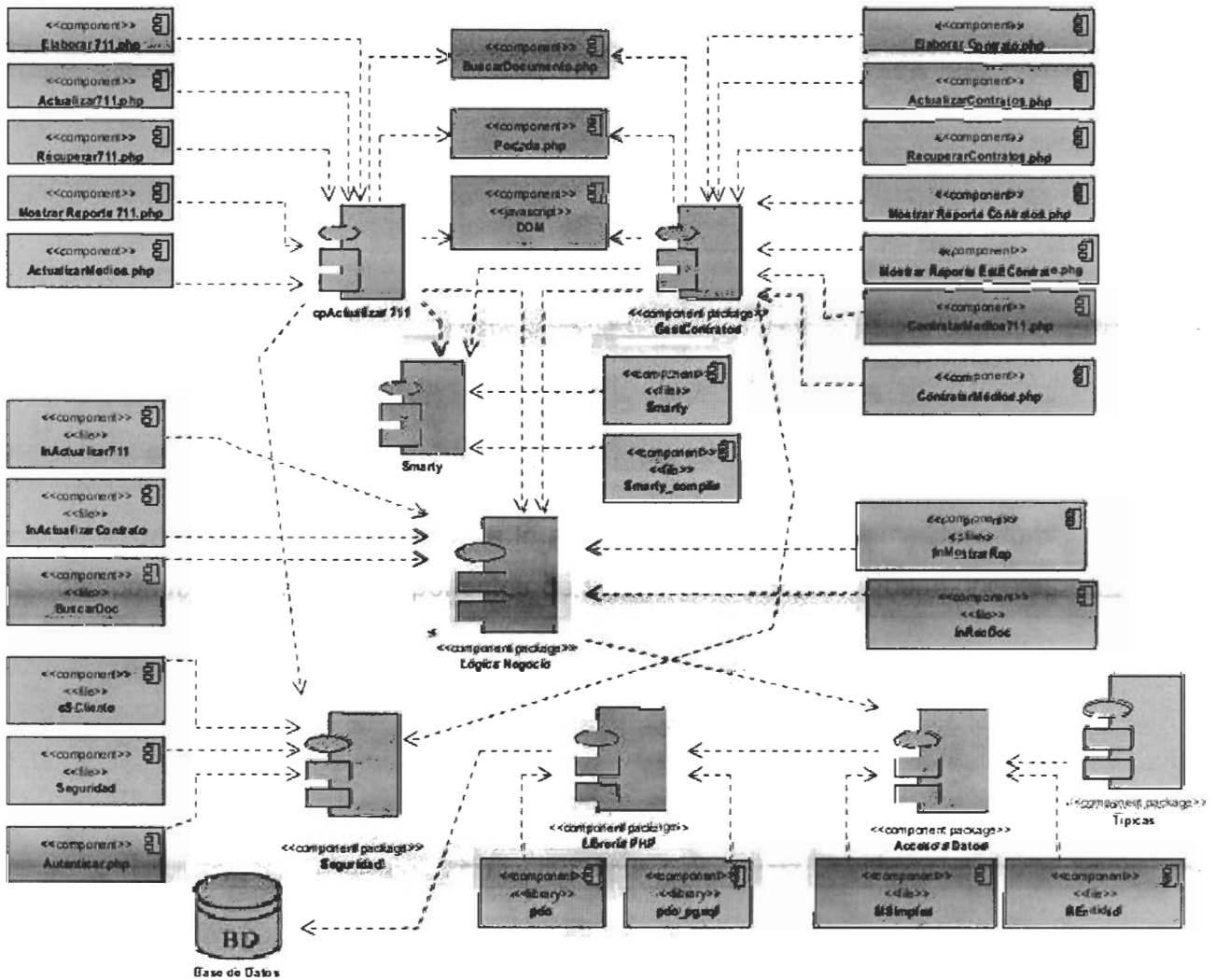


Fig. 4.16 Diagrama de componentes

En este diagrama de componentes se representan las relaciones entre los principales paquetes de componentes, así como los componentes que tienen contenidos. Quisiera mencionar que en la capa de presentación se utilizó el motor de Plantillas Smarty por lo que cada componente (página php) tiene asociada un componente que es una plantilla .tpl.

4.5 Conclusiones

Al término de este capítulo hemos podido apreciar realmente como será implementado el sistema, así como quedará desplegado el mismo. También obtuvimos los principales artefactos del flujo de diseño e implementación y se obtuvo el modelo físico de la base de datos para su montaje en el gestor de base de datos seleccionado. Por tanto podemos concluir que ya tenemos todas las herramientas para comenzar a implementarlo.



Estudio de Factibilidad

En este capítulo se abordan los principales aspectos a tener en cuenta para planificar un sistema informático, así como un análisis de los costos de llevar a cabo el desarrollo previo del sistema. Además se hace una valoración de los principales beneficios que nos reportará. A continuación se exponen las ideas antes expuestas.

5.1 Planificación

Para poder completar con éxito un proyecto de software, se necesita tener un control riguroso sobre el tiempo, las personas o los imprevistos que puedan surgir, como por ejemplo cambios en el software; de toda esta problemática surge un término muy importante y que no debemos dejar pasar por alto, ya que podría provocar el fracaso del desarrollo de nuestro sistema informático, este término es la *planificación*.

Para llevar a cabo la misma es de suma importancia primeramente tener en cuenta:

- ✓ Definición y seguimiento de cómo se van a lograr los distintos objetivos.
- ✓ Organización temporal de las tareas.

- ✓ Asignación y distribución de recursos a tareas.

Ahora existen técnicas que nos puede ayudar a llevar la planificación de un proyecto informático como es el **Diagrama de GANTT**.

El mismo es muy eficaz en las etapas iniciales de la planificación, además nos sirve para llevar una planificación temporal de las tareas principales a ejecutar. Aquí una muestra del mismo para nuestro proyecto:

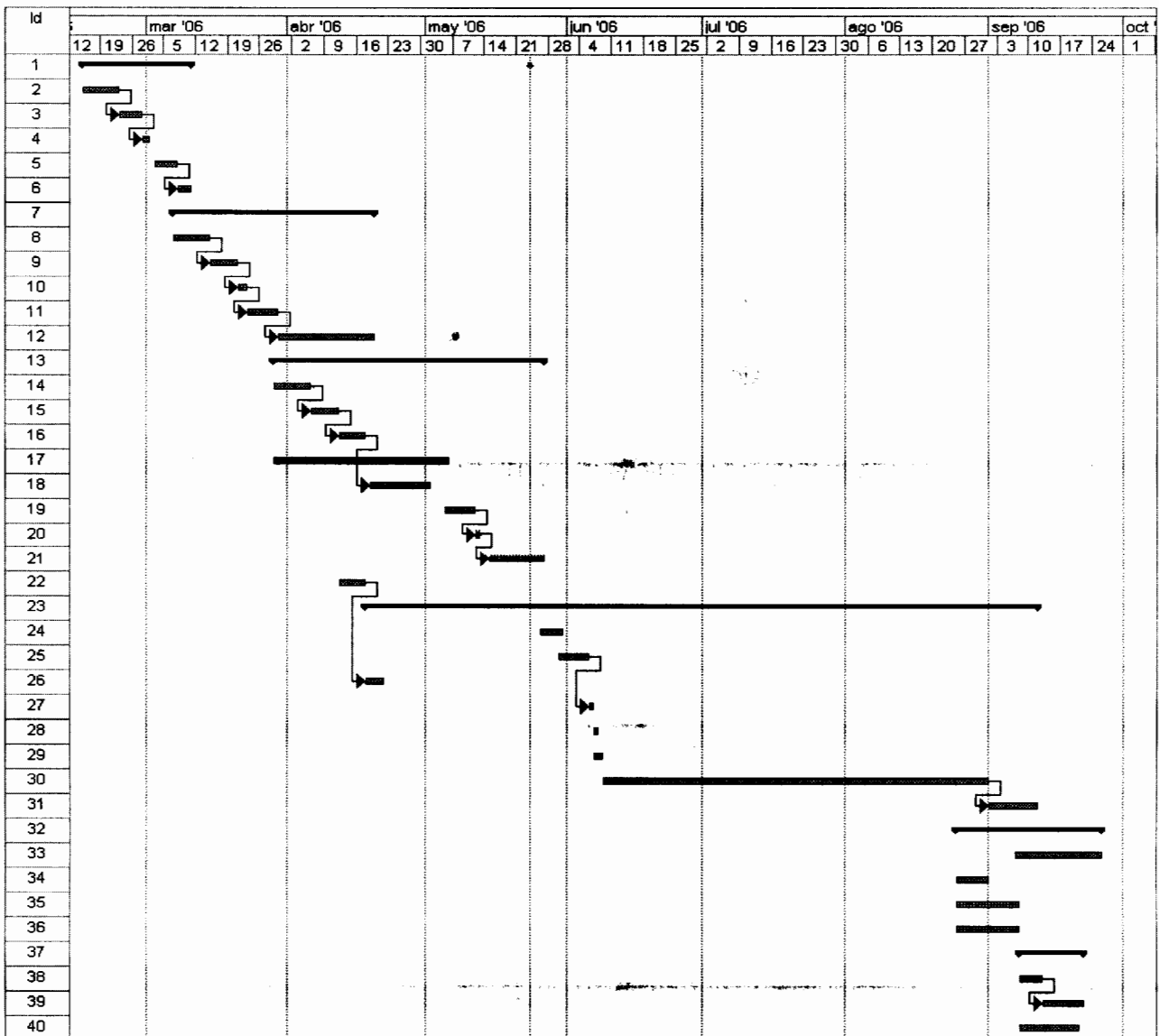


Fig. 5.1 Diagrama de GANTT

En el siguiente diagrama se evidencia la asignación de los tiempos para cumplimentar las siguientes tareas y que fueron reflejados en el diagrama anterior:

Id	Nombre de tarea	Comienzo real	Fin real
1	Modelamiento del Negocio	mié 2/15/06	NA
2	Identificación Procesos del Negocio	mié 2/15/06	mié 2/22/06
3	Identificación de Roles del Entorno del Negocio	jue 2/23/06	lun 2/27/06
4	Especificación de Reglas del Negocio	mar 2/28/06	mié 3/1/06
5	Obtención de los modelos principales del negocio	vie 3/3/06	mar 3/7/06
6	Descripción de los Casos de Uso del Negocio	mié 3/8/06	vie 3/10/06
7	Captura de requisitos	mar 3/7/06	mié 4/19/06
8	Obtención requerimientos funcionales del sistema	mar 3/7/06	mar 3/14/06
9	Obtención requerimientos funcionales del sistema	mié 3/15/06	lun 3/20/06
10	Encontrar los actores y casos de uso	mar 3/21/06	mié 3/22/06
11	Obtención del modelo de caso de uso del sistema	jue 3/23/06	mié 3/29/06
12	Descripción de los casos de usos	jue 3/30/06	mié 4/19/06
13	Análisis y Diseño	mié 3/29/06	NA
14	Identificación de clases del análisis por CU	mié 3/29/06	mié 4/5/06
15	Obtención del diagrama de clases del análisis por CU	jue 4/6/06	mar 4/11/06
16	Obtención del diagrama de colaboración por CU	mié 4/12/06	lun 4/17/06
17	Obtención de los mecanismos de diseño	mié 3/29/06	vie 5/5/06
18	Identificación de las clases del diseño por CU	mié 4/19/06	lun 5/1/06
19	Obtención del diagrama de clase del diseño por CU	vie 5/5/06	jue 5/11/06
20	Diseñar interfaces	vie 5/12/06	vie 5/12/06
21	Obtención del diagrama de secuencia por realizaciones de CU	lun 5/15/06	NA
22	Obtención del diagrama de clases persistentes	mié 4/12/06	lun 4/17/06
23	Implementación	mar 4/18/06	NA
24	Definir arquitectura	NA	NA
25	Diseñar componentes	NA	NA
26	Diseñar Base de Datos	mar 4/18/06	vie 4/21/06
27	Obtener diagrama de componentes	NA	NA
28	Obtener diagrama de despliegue	NA	NA
29	Generación de códigos a partir de diagrama de clases	NA	NA
30	Implementar los elementos de diseño	NA	NA
31	Integrar los resultados en un sistema ejecutable	NA	NA
32	Prueba	NA	NA
33	Encontrar y documentar los defectos del software	NA	NA
34	Probar que el software trabaje como fue diseñado	NA	NA
35	Probar los requisitos que debe cumplir el software	NA	NA
36	Probar que los requisitos fueron implementados correctamente	NA	NA
37	Implantación y Evaluación	NA	NA
38	Montaje del Sistema	NA	NA
39	Preparación del personal	NA	NA
40	Evaluación de funcionamiento del sistema	NA	NA

Fig. 5.2 Tabla de duración de tareas

Para llevar a cabo todas estas tareas, debemos conformar un equipo de desarrollo y asignar responsabilidades. Primeramente definimos la estructura y su organización y definir los canales de comunicación. Para el caso nuestro quedo estructurado por **Jefe de proyecto**:

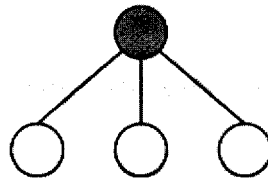


Fig. 5.3. Estructura organizativa del equipo de desarrollo

Para chequear el cumplimiento continuo de esta planificación en cada iteración de la metodología de desarrollo usada por nosotros (RUP) se reelabora el plan y se continua con su plena revisión.

Ahora, en todo proceso de planificación se declaran tareas con tiempos de cumplimiento, pero su vez para cumplimentar las mismas se asignan recursos y personal lo cual implica adentrarnos en un término fundamental *costes*. El mismo se analiza desde el punto de vista: *costos*, *costos del personal* y *costos globales*. Todo esto se evidencia en el análisis de lo siguiente:

Costos

- ✓ Alcance del proyecto.
- ✓ Duración.
- ✓ Estimación de recursos hardware.
- ✓ Estimación de recursos software.
- ✓ Estimación de recursos humanos.

Costos del personal

- ✓ Establecer un coste unitario Categoría/Unidad de esfuerzo (persona/día, persona/mes, persona/año) llamado *coeficiente de trabajo*.
- ✓ Calcular el esfuerzo estimado para cada categoría.
- ✓ Calcular el coste de cada categoría.
- ✓ Calcular el coste total.

Costos Globales

- ✓ Hardware.
- ✓ Software.
- ✓ Personal.
- ✓ Viajes.
- ✓ Servicios.
- ✓ Gastos Comunes.

En la sección siguiente se evidencia un análisis de lo anteriormente expuesto usando una técnica para la estimación de costes.

5.2 Planificación basada en casos de uso

La especificación de los requerimientos mediante Casos de Uso ha probado ser uno de los métodos más efectivos para capturar la funcionalidad de un sistema. Este hecho se puede apreciar en algunas metodologías actuales ampliamente difundidas, como el Proceso Unificado de Rational (Rational Unified Process), en la cual se propone especificar la funcionalidad de los sistemas mediante la utilización de Casos de Uso.

El método de Casos de Uso permite documentar los requerimientos de un sistema en términos de Actores y Casos de Uso. Si bien los Casos de Uso permiten especificar la funcionalidad de un sistema bajo análisis, no permiten por sí mismos efectuar una estimación del tamaño que tendrá el sistema o del esfuerzo que tomaría implementarlo.

Para la estimación del tamaño de un sistema a partir de sus requerimientos, una de las técnicas más difundidas es el Análisis de Puntos de Función. Ésta técnica permite cuantificar el tamaño de un sistema en unidades independientes del lenguaje de programación, las metodologías, plataformas y/o tecnologías utilizadas, denominadas Puntos de Función.

5.2.1 Calcular el factor de peso de los actores sin ajustar (UAW)

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

UAW = Sumatoria (Factor.Peso Actores.*Cantidad Actores)

UAW = 6

5.2.2 Calcular el factor de peso de los casos de uso sin ajustar (UUCW)

Tipo de CU	Descripción	Peso	Cantidad de CU	Total
Simple	El caso de uso tiene de 1 a 3 transacciones.	5	4	10
Medio	El caso de uso tiene de 4 a 7 transacciones.	10	2	0
Complejo	El caso de uso tiene más de 8	15	2	30

transacciones.			
----------------	--	--	--

UUCW = Sumatoria (Factor Peso CU * Cantidad CU)

UUCW = 70

5.2.3 Calcular el número de puntos de casos de uso no ajustados (UUCP)

UUCP = UAW + UUCW

UUCP = 76

5.2.4 Calcular factor que toma en cuenta complejidad del proyecto (TCF)

Factor	Descripción	Peso	Valor asignado	Total
T1	Sistema distribuido	2	5	10
T2	Tiempo de respuesta	1	4	4
T3	Eficiencia del usuario final	1	4	4
T4	Funcionamiento Interno complejo	1	4	4
T5	El código debe ser reutilizable	1	3	3
T6	Facilidad de instalación	0.5	2	1
T7	Facilidad de uso	0.5	5	2.5
T8	Portabilidad	2	5	10
T9	Facilidad de cambio	1	4	4
T10	Concurrencia	1	5	5
T11	Incluye objetivos especiales de seguridad	1	4	4
T12	Provee acceso directo a terceras partes	1	0	0

T13	Se requieren facilidades especiales de entrenamiento de usuarios	1	3	3
-----	--	---	---	---

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

TCF = 1.155

5.2.5 Calcular el factor que toma en cuenta el ambiente del proyecto (EF)

Factor	Descripción	Peso	Valor asignado	Total
E1	Familiaridad con el modelo de proyecto utilizado	1.5	5	7.5
E2	Experiencia en la aplicación	0.5	4	2
E3	Experiencia en la orientación a objetivos.	1	4	4
E4	Capacidad del analista líder.	0.5	5	2.5
E5	Motivación.	1	4	4
E6	Estabilidad de requerimientos	2	4	8
E7	Personal Part-Time	-1	2	-2
E8	Dificultad del lenguaje de programación	-1	3	-3

$$EF = 1.4 - 0.03 * \text{Sumatoria (Factor Peso E * Valor Asignado)}$$

EF = 0.71

5.2.6 Calcular Puntos de casos de uso ajustados (UCP)

$$UCP = UUCP * TCF * EF$$

UCP = 62.3238

5.2.7 Determinar el Factor de conversión (CF)

Para determinar este factor se debe tener en cuenta lo siguiente

$$CF = 20$$

5.2.8 Estimación de esfuerzo a través de los puntos de casos de uso (E)

$$E = UCP * CF$$

$$E = 1246.476$$

5.2.9 Calcular esfuerzo de todo el proyecto.

Actividad	Porcentaje %	Horas-Hombres
Análisis	10	311.619
Diseño	20	623.238
Implementación	40	1246.476
Pruebas	15	467.4285
Sobrecarga (otras actividades)	15	467.4285
Total	100	3116.19

5.2.10 Análisis de esfuerzo

Podemos decir, como el esfuerzo total es de 3116.19 horas-hombre y cada mes tiene como aproximadamente 240 horas, esto daría un esfuerzo total

$ET = 12.984125$ mes-hombre, por lo que podemos arribar a la conclusión de que una persona puede desarrollar este sistema informático en 12 meses y 24 días aproximadamente.

Ahora el tiempo de desarrollo del equipo será el esfuerzo total entre la cantidad de integrantes del equipo que es 3, por tanto será lo que significa que podría desarrollarse en 4 meses y 10 días.

$$\text{TDE} = 4.32804167$$

5.2.11 Análisis de costo del proyecto.

Podemos decir que nuestro equipo de desarrollo esta conformado por 3 integrantes cuyo salario promedio mensual es de \$50.00.

Costo Hombre- Mes

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

$$\text{CHM} = 50.00 \text{ \$/mes}$$

Costo Total

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

$$\text{Costo} = 12.984125 * 50$$

$$\text{Costo} = \$ 649.20625$$

5.3 Beneficios Tangibles e Intangibles

A la hora de adentrarnos en el desarrollo de un sistema informático lo primero que nos podemos preguntar es **¿que beneficios nos reportará el mismo?**, es por eso que una de las formas más comunes para observar estos beneficios es utilizando los términos de beneficios *tangibles* e *intangibles*.

Los **beneficios tangibles** son las ventajas económicas (*cuantificables*) que obtiene la organización, es decir aquellos que los directivos pueden tomar en sus manos.

Algunos de los beneficios que la organización obtiene a través de un sistema de información son difíciles de cuantificar, pero no por ello dejan de ser importantes. A éstos se les conoce como **beneficios intangibles** (*cualitativos*).

Para el caso nuestro objetivo principal es desarrollar una aplicación WEB que automatice el proceso de control y ejecución de la contratación, por lo que nuestros beneficios serán:

Beneficios tangibles

- ✓ Logro de una implementación económica.
- ✓ Facilidades de uso.
- ✓ Desarrollo basado en estándares abiertos.
- ✓ Ahorro de tiempo y dinero.
- ✓ Mejor gestión de la información, más rapidez.
- ✓ Escalabilidad y flexibilidad.
- ✓ Conexión entre distintas plataformas.

Beneficios Intangibles

- ✓ Mejoramiento y apoyo del proceso de toma de decisiones.
- ✓ Construcción de una cultura de colaboración.
- ✓ Obtención de procesos de aprendizaje más fáciles.
- ✓ Mejoramiento de la productividad dentro del proceso de desarrollo.

5.4 Análisis de costos y beneficios.

Incorporar nueva tecnología dentro de una organización puede incrementar los costos o disminuirlos, dependiendo del costo de la nueva tecnología en comparación con la del sistema al que reemplazará o si es la primera vez que se implantará dentro de la organización. La nueva tecnología también puede acarrear beneficios no monetarios, como mejoras en la prestación del servicio o

en la transparencia. Es conveniente realizar una evaluación en términos de costo-efectividad y de los potenciales beneficios de la nueva tecnología antes de comprometerse con su introducción.

Para llevar a cabo un análisis de costo beneficio a propósito de la introducción de nueva tecnología debemos considerar los siguientes elementos:

- ✓ Aclarar el propósito de la adquisición.
- ✓ Describir el proceso que será reemplazado.
- ✓ Señalar los riesgos o problemas que pueden surgir o agudizarse si se sigue utilizando el proceso actual.
- ✓ Describir como funcionaría la nueva aplicación tecnológica.
- ✓ Señalar cuales son los beneficios que reportará.
- ✓ Mencionar que problemas o riesgos que puede representar su adopción.
- ✓ Analizar si la nueva tecnología mejorará o no la transparencia del proceso.
- ✓ Estimar todos los costos implicados en la adquisición de nueva tecnología, incluyendo equipo, programas, infraestructura, comunicaciones, transporte, personal, consultores externos y mantenimiento.
- ✓ Calcular los costos que representaría una falla de la nueva tecnología.
- ✓ Mostrar los costos asociados con el procedimiento actual.
- ✓ Identificar las economías que podría generar la nueva tecnología.
- ✓ Detallar un cronograma para su implantación.
- ✓ Examinar su sustentabilidad e indicar su ciclo estimado de vida
- ✓ Identificar cualquier factor externo necesario para la viabilidad del proyecto, como la aprobación de los interesados o la promulgación de reformas legislativas.

Para el caso nuestro debemos decir que el desarrollo de este software surge con el proceso de mejorar la gestión dentro del proceso de contratación en el MINFAR, de ahí la tarea de desarrollar la aplicación Web. Llevar a cabo este desarrollo no influirá directamente en gastos elevados de recursos, además su implantación deberá reducir los tiempos de gestión en los principales procesos vinculados a la contratación. El montaje de las herramientas necesarias que aseguran su funcionamiento no incurrirá en gastos elevados. La misma no requiere la adquisición de tecnologías de desarrollo costosas, ni con el riesgo de ser propietarias. El sistema es portable ya que está concebido en su análisis que debía desarrollarse para su funcionamiento en varias plataformas (UNIX y Windows), por lo que un cambio entre estas plataformas no afectará su funcionamiento, por lo que deberá ser viable y factible. Podemos decir además que el sistema será sustentable ya que se han concebido etapas de mantenimiento, actualizaciones de componentes indispensables en su funcionamiento, así como reducción en los costos de operación, por lo que aseguramos un largo tiempo de vida del software.

5.5 Conclusiones

Podemos concluir que después de haber realizado el estudio de factibilidad del producto informático en desarrollo, que le mismo no provocará grandes gastos a las organizaciones implicadas en su desarrollo, en este caso la UCI y el MIFAR.

Es muy importante recalcar que a pesar de que el producto todavía está en pleno desarrollo, suponemos que el mismo tendrá un impacto social relevante, así como del punto de vista ambiental y político. En cuanto a los gastos para llevar a cabo este desarrollo podemos resumir que es factible y económico desarrollar el sistema en cuestión.

CONCLUSIONES

Hoy en día llevar a cabo el proceso de contratación dentro del Ministerio de las Fuerzas Armadas Revolucionarias es algo complejo por la gran cantidad de información que se maneja, así como los documentos que se registran. Es por esto que se evidencia la necesidad de desarrollar un sistema informático que automatice este proceso.

En este trabajo se presenta una parte esencial de este proceso tan complejo, y en el cual esperamos se mejore la gestión de la información, así como los tiempos de respuesta de las principales recuperaciones necesitadas a diario en dicha institución para apoyar el proceso de toma de decisiones.

Esta parte del sistema se desarrollo sin violar ninguna etapa indispensable de la metodología RUP, la cual nos acompañó en todo el proceso. Se trabajó en conjunto siguiendo los principales estándares establecidos para lograr un sistema de alta calidad.

Por todo lo antes mencionado se concluye que los objetivos propuestos para el presente proyecto han sido cumplidos en su generalidad. A continuación incluyo ciertas recomendaciones que pueden contribuir a mejorar el funcionamiento y desarrollo del sistema.

RECOMENDACIONES

Solamente quisiéramos que para mejorar el desarrollo del sistema en próximas iteraciones recomendar lo siguiente:

- ✓ Profundizar en los estándares definidos de diseño para redefinirlos y lograr interfaces más claras al usuario final.
- ✓ Ampliar en el tema de las recuperaciones, manejando el término de que puedan ser dinámicas para dar apoyo al proceso de toma de decisiones.
- ✓ Buscar un mecanismo óptimo y factible para dar tratamiento a los posibles errores dentro del sistema.
- ✓ No perder el vínculo con el cliente principal para posibles actualizaciones del sistema.
- ✓ Obtener las opiniones finales del uso del sistema por parte del personal que lo utilizará en búsqueda de nuevas mejoras.

REFERENCIAS BIBLIOGRAFICAS

- Pressman, R. *Software Engineering. A Practitioner's Approach*. Fourth Edition. McGraw – Hill. USA, 1999.
- Booch, G., Rumbaugh, J., Jacobson, I. *El Lenguaje Unificado de Modelado*. Addison-Wesley. 1999.
- Larman, C. *UML Y PATRONES, Introducción al análisis y diseño orientado a objetos*. La Habana. Cuba 2004.
- Booch, G., Rumbaugh, J., Jacobson, I. *El Proceso Unificado de Desarrollo de Software*. La Habana. Cuba 2004.
- Franco, J A. *UML en acción. Modelando Aplicaciones Web*. La Habana. Cuba 2006.
- Muñoz, J. *Un Framework basado en OSGi para el Desarrollo de Sistemas Pervasivos*. mayo 2006.
- Méndez G. *Construcción de Aplicaciones Web con UML. Diseño*. Mayo 2006.
- Méndez G. *Construcción de Aplicaciones Web con UML. Conceptos Generales*. Mayo 2006.
- Mexica J. *Un patrón arquitectónico para la creación de cursos WBT*. mayo 2006.

REFERENCIAS BIBLIOGRAFICAS

Herrera R A., Caldera R J., Martínez M. tema: *Análisis y Diseño de Sistemas con el Lenguaje de Modelaje Unificado (UML)*. Univeridad Catolica *Redemptoris Mater*". Proyecto Monografico. Managua, abril de 1999.

Soto Lopez N M., Saborit Ramirez Y. *Sistema de Catalogación y Recuperación de Recursos de Información, HUBBLE*, trabajo de diploma para optar por el titulo de Ingenieria en Informatica. Instituto Superior Politecnico "Jose Antonio Hecheverria". Ciudad de la Habana, junio del 2004.

Espinosa Hidalgo A. *Sistema para la Administración Unificada de usuarios*. Trabajo de diploma para optar por el titulo de Ingenieria en Informatica. Instituto Superior Politecnico "Jose Antonio Hecheverria". Ciudad de la Habana, junio del 2004.

Franco Navarro, J. *Acceso a Datos Objeto – Relacional; Patrones y Mecanismos. EJB - CMP, una solución "Alguien lo hace por mi" sobre J2EE*. Consultado en mayo del 2006.

Grupo de Ingeniería del software, Universidad de Sevilla, *Introducción al Análisis de Requisitos*. Publicado en Octubre del 2005. Consultado en mayo del 2006.

Grupo de Ingeniería del software, Universidad de Sevilla, *Elicitación de Requisitos: modelado del Negocio. (Diagramas de Actividades)*. Publicado en Octubre del 2005. Consultado en mayo del 2006.

Grupo de Ingeniería del software, Universidad de Sevilla, *Documentación de Casos de Uso*. Publicado en Octubre del 2005. Consultado en mayo del 2006.

REFERENCIAS BIBLIOGRAFICAS

Grupo de investigación Ingeniería de Software. Universidad EAFIT. *Tertulia de Ingeniería de Software. La Importancia de la Arquitectura en el desarrollo de software de calidad*. Publicado febrero 17 de 2005. Consultado mayo 26, 2006.

Isla Monte, j. *Modelado Estructural de Patrones de Diseño*. Cadiz, España, 2003.

Peralta M. *Estimación del esfuerzo basada en casos de usos.pdf*.

<http://www.itba.edu.ar/capis/webcapis/planma.html> , mayo del 2006.

<http://msdn.microsoft.com/architecture/> (mayo 10, 2006)

<http://www.dm.univaq.it/~muccini> (mayo 10, 2006)

<http://www.sparxsystems.cl/uml-tutorial.html> (mayo 28, 2006)

http://www.sparxsystems.cl/resources/uml_datamodel.html (mayo 28, 2006)

http://www.sparxsystems.cl/resources/developers/uml_patterns.html (mayo 28 2006)

<http://www.mozilla-europe.org/es/about/> (mayo 22, 2006)

<http://kursor.net/article/584/control-de-versiones-con-subversion> (mayo 22, 2006)

<http://www.capisol.net/formacion/mod/book/view.php?id=52&chapterid=25> (mayo 22,2006)

<http://www.lug.fi.uba.ar/sistemas-de-control-versiones.mht> (mayo 22, 2006)

<http://www.monografias.com/trabajos15/estimacion-hipermedia/estimacion-hipermedia.shtml#INTRO> (mayo 23, 2006)

http://es.wikipedia.org/wiki/Patr%C3%B3n_de_dise%C3%B1o (mayo 29, 2006)

<http://es.tldp.org/Tutoriales/doc-modelado-sistemas-UML/multiple-html/x219.html> (mayo 29, 2006)

REFERENCIAS BIBLIOGRAFICAS

<http://www.elrincondelprogramador.com/articulos/puntuar.asp?puntos=4&id=45>>4ptos. (mayo 29, 2006)

<http://www.fi.uba.ar/~dmontal/> (mayo 29, 2006)

<http://www3.uji.es/~mmarques/f47/apun/node79.html> (junio 4, 2006)

<http://www.monografias.com/trabajos10/recped/recped.shtml#intro> (marzo 14, 2006)

http://www.php_net-pgsql.htm/ (marzo15, 2006)

GLOSARIO DE TERMINOS

SCM-711

Es un documento oficial donde se solicita a la Empresa Importadora los posibles suministradores a un conjunto de medios aprobados en la planificación para su contratación. Es el documento principal dentro del proceso de contratación dentro del MINFAR.

ADMINISTRADOR SUBORDINADO DE CREDITOS DE PRIMER NIVEL

Es el órgano de Economía al que se le desagregan recurso presupuestarios y asignaciones de fondos para su distribución a los órganos Administradores o Consumidores de Créditos que se le subordinan a los que se le asegura financieramente, realizando su notificación, el control de la ejecución y liquidación del presupuesto a su nivel. Lo constituyen la Jefatura de la Logística de las FAR, la Jefatura de Tropas Especiales, la Dirección de Contrainteligencia Militar, los ejércitos, la División de Tanques y la Unidad de Servicios de las FAR.

CONSUMIDOR DE CREDITO ATENDIDO POR LA DIRECCION DE ECONOMIA

Es el órgano de Economía o de Finanzas donde se ejecutan y registra los gastos de presupuesto y se realizan los pagos a partir de las asignaciones de fondos situadas por el órgano de Economía de nivel superior. De igual forma en el se captan y cobran los recursos financieros (ingresos).

GASTOS CORRIENTES

Contiene los gastos de funcionamiento, entre los que se incluyen las retribuciones monetarias, las prestaciones de la seguridad social a corto plazo y la estimulación al personal de la empresa MINFAR; la compra de medios

GLOSARIO DE TERMINOS

materiales y demás bienes que se consumen; el pago por servicios recibidos; así como otros gastos en que incurren las entidades de la empresa MINFAR.

GASTOS DE CAPITAL

Abarca todos los gastos necesarios para la ejecución y puesta en explotación de las inversiones, tanto en el equipamiento tecnológico y su montaje como la construcción de edificaciones y su rehabilitación, la documentación técnica, gastos vinculados con la administración desde el inicio hasta el final de la inversión, gastos de prueba y puesta en explotación, capital de trabajo y otros gastos necesarios.

ORGANO ABASTECEDOR CENTRAL

Son los órganos de dirección del Aparato Central del MINFAR que de acuerdo a las misiones que tiene asignadas responden por la planificación, adquisición, abastecimiento, ejecución y consumo del aseguramiento de los recursos materiales y presupuestarios necesarios para el funcionamiento de la actividad o actividades a todos los niveles de las FAR.

ORGANO CENTRO DE BALANCE

Son órganos de dirección del Aparato Central del MINFAR que responden por el balance de un conjunto de nomenclaturas de productos que le son asignadas, centralizando bajo su responsabilidad la adquisición en el país o en el exterior y la distribución de las demandas de recursos materiales o servicios mediante la compra venta de mercancías; las cuales fueron planificadas por los órganos abastecedores centrales, consumidores autorizados, uniones y otras entidades del sistema empresarial militar, con el objetivo de dar cumplimiento a las políticas aprobadas y abaratar los costos.

ORGANO CONSUMIDOR AUTORIZADO

Son los órganos de dirección del Aparato Central del MINFAR autorizados para planificar, adquirir, ejecutar y consumir determinados recursos materiales y

GLOSARIO DE TERMINOS

presupuestarios, que por su característica son específicos del trabajo que ejecutan y que no son abastecidos por ningún órgano abastecedor central.

ORGANO EVALUADOR

Son órganos de dirección del Aparato Central del MINFAR que por decisión de la jefatura del MINFAR responde por la evaluación de determinadas actividades o demandas de recursos materiales, que le son presentadas por los órganos y uniones que participan en el proceso de planificación, emitiendo sobre el particular un dictamen o aval para su inclusión en los planes.

ORGANO INVERSIONISTA DIRECTO

Son los órganos que están directamente responsabilizados con la administración de las inversiones constructivas. Los constituyen los ejércitos, la División de Tanques, la Jefatura de Tropas Especiales, la Dilección de Contrainteligencia Militar y otros que estén nominalizados en la Directiva de Inversiones que aprueba el Ministro de las FAR anualmente.

ORGANO INVERSIONISTA PRINCIPAL

Son órganos de dirección del Aparato Central del MINFAR, las uniones, el Grupo Empresarial GEOCUBA y la Unidad Administrativa Comercial Central que tienen la responsabilidad de dirigir integralmente el proceso inversionista de su radio de acción y son los encargados de determinar los niveles de actividad del plan de inversiones en correspondencia con las misiones asignadas.

ORGANO RECTOR

Son aquellos órganos de dirección del Aparato Central del MINFAR que actúan como rectores en el proceso de planificación, determinando los niveles de actividades fundamentales.

UNION (EMPRESA)

Son las organizaciones superiores de dirección del sistema empresarial militar, las cuales actúan bajo el principio de autofinanciamiento empresarial. Las comprenden las uniones, la Unidad Administrativa Comercial Central y el Grupo Empresarial GEOCUBA. En este grupo hemos incluido a la Empresa TECNOIMPORT, la cual responde en el MINFAR por la importación directa de los recursos materiales para una gran parte de nuestros requerimientos en divisas.

UNIDAD PRESUPUESTADA DE TRATAMIENTO DIFERENCIADO

Es la unidad presupuestada por la gestión que realiza, obtiene un nivel de ingresos que se le autoriza retener para financiar total o parcialmente sus gastos.

CONTRATO

Es un documento elaborado por la Empresa Importadora donde se contratan los medios solicitados en los documentos 711.

PHP

Es un lenguaje de programación usado generalmente para la creación de contenido para sitios Web. PHP es el (Personal Home Page, Preprocesador de Hipertexto) es un lenguaje interpretado usado para la creación de aplicaciones para servidores, o creación de contenido dinámico para sitios Web, y últimamente también para la creación de otro tipo de programas incluyendo aplicaciones con interfaz gráfica usando la librería

POSTGRESQL

Es un servidor de base de datos relacional libre, liberado bajo la licencia BSD. Es una alternativa a otros sistemas de bases de datos de código abierto (como MySQL, Firebird y MaxDB), así como sistemas propietarios como Oracle o DB2.

GLOSARIO DE TERMINOS

TCP/IP

Se refieren a dos protocolos de red: Transmission Control Protocol (Protocolo de Control de Transmisión) e Internet Protocol (Protocolo de Internet). Los diferentes protocolos de la suite TCP/IP trabajan conjuntamente para proporcionar el transporte de datos dentro de Internet (o Intranet). En otras palabras, hacen posible que accedamos a los distintos servicios de la Red.

UNIX

Sistema operativo atribuido a Ken Thompson y comercializado por la empresa ATT en la década de los 70s. Entre sus principales características tenemos que es: portable, robusto, flexible y abierto, actualmente goza de gran popularidad dentro de la tecnología de Internet.

ANEXOS

ANEXO I.

Órganos
Órganos Rectores
Centros de Balance
Órganos Abastecedores Centrales
Órganos Consumidores Autorizados
Uniones (Empresa)
Órganos Inversionistas Principales
Órganos Inversionistas Directos
Órganos Evaluadores
Administradores Subordinados de Crédito de Primer Nivel
Unidades Consumidoras de Crédito atendidas directamente por la Dirección de Economía
Entidades Presupuestadas de Tratamiento Diferenciado

Tabla 11. Relación de órganos que participan en los procesos de planificación, aprobación y contratación y control.

ANEXO II.



Figura 1. Flujo informativo entre los procesos de planificación, aprobación y contratación y control.

ANEXO III. Descripción de alto nivel de casos uso.

Nombre del caso de uso	Mostrar Reporte Control de Estado de Contrato
Actor(es)	Especialista CB (inicia)
Propósito	Mostrar un reporte del estado de ejecución del documento contrato recuperado.
Resumen	Este caso de uso se inicia cuando el especialista CB decide mostrar reporte estado de ejecución de documento contrato recuperado. El sistema muestra interfaz con el reporte del mismo.
Precondiciones	El caso de uso base Recuperar Contratos debe haberse ejecutado. El actor debe haberse autenticado.
Poscondiciones	

Quiero mencionar que este caso de uso por la falta de información sobre el mismo, está en fase de estudio sobre como deberá salir al usuario, por lo que no se ha definido un prototipo de interfaz para el mismo, por lo que queda para una próxima iteración su análisis previo.

ANEXO IV. Descripción de casos de uso expandidos

Caso de Uso:	Recuperar 711
Actor(es):	Especialista CB (inicia)
Propósito:	Recuperar documentos 711 usando un criterio determinado de búsqueda.
Resumen:	El caso de uso inicia cuando el actor decide recuperar información de documentos 711. Se le mostrara la información de los mismos de acuerdo a diversos criterios especificados por el actor.
Referencias:	RF1
Casos de uso	Mostrar Reporte 711 (extensión).

asociados	
Precondiciones:	El usuario levanta el sistema y se encuentra en la página principal. Debe haber solicitado el servicio de autenticación.

Pantalla 1

Acción del Actor	Respuesta del Sistema
1. El usuario elige operación a realizar.	2. Si elige opción Recuperar 711, ver sección "Recuperar 711".

Sección "Recuperar 711"

The screenshot shows a web interface titled "Recuperar 711". It features several dropdown menus for filtering: "Especialista", "Cuenta Bancaria", "Importador", "Consumidor", "No", "Aprobado", "Presentado", and "Tipo Medio". Below these is a table with columns: "No", "Módulo 711", "sumido", "Importador", "Presentado", and "Aprobado". At the bottom, there are three buttons: "Recuperar", "Mostrar Reporte", and "Cancelar".

Pantalla 2

	1. El sistema muestra la Pantalla 2, con los filtros: especialista(A), cuenta bancaria (B), importador(C), consumidor (D), No. (E), tipo medio (H) y medio (I),
--	---

ANEXOS

	así como las cajas de texto aprobado (F) y presentado (G) vacías, así como el grid (L) vacío.
2. El usuario selecciona opcionalmente una opción en cualquier filtro o introduce fechas en las cajas de texto aprobado (G) o presentado (G). Presiona botón Recuperar (J).	3. El sistema busca documentos 711 analizando los criterios especificados por el actor y muestra los resultados en el grid (L).
4. El usuario presiona botón Cancelar (N).	5. El sistema devuelve a Pantalla que la visualizo a ella.
6. El usuario selecciona fila (M) del grid (L) y presiona botón Mostrar Reporte (K).	7. Ver descripción del caso de uso extendido Mostrar Reporte 711.
Poscondiciones:	

Caso de Uso:	Mostrar Reporte 711
Actor(es):	Especialista CB (inicia).
Propósito:	Mostrar reporte del documento 711 recuperado.
Resumen:	Este caso de uso se inicia cuando el especialista CB decide mostrar reporte de documento 711 recuperado. El sistema muestra interfaz con el reporte del mismo.
Referencias:	RF1
Precondiciones:	El usuario ha recuperado documentos 711. Debe haber solicitado servicio de autenticación.

ANEXOS

	así como las cajas de texto aprobado (F) y presentado (G) vacías, así como el grid (L) vacío.
2. El usuario selecciona opcionalmente una opción en cualquier filtro o introduce fechas en las cajas de texto aprobado (G) o presentado (G). Presiona botón Recuperar (J).	3. El sistema busca documentos 711 analizando los criterios especificados por el actor y muestra los resultados en el grid (L).
4. El usuario presiona botón Cancelar (N).	5. El sistema devuelve a Pantalla que la visualizo a ella.
6. El usuario selecciona fila (M) del grid (L) y presiona botón Mostrar Reporte (K).	7. Ver descripción del caso de uso extendido Mostrar Reporte 711.
Poscondiciones:	

Caso de Uso:	Mostrar Reporte 711
Actor(es):	Especialista CB (inicia).
Propósito:	Mostrar reporte del documento 711 recuperado.
Resumen:	Este caso de uso se inicia cuando el especialista CB decide mostrar reporte de documento 711 recuperado. El sistema muestra interfaz con el reporte del mismo.
Referencias:	RF1
Precondiciones:	El usuario ha recuperado documentos 711. Debe haber solicitado servicio de autenticación.

ANEXOS

SOLICITUD DE ABASTECIMIENTO DE PRODUCTOS IMPORTADOS

DLRE (Caney) Modelo 711 FAR

Consumidor: DIR. COMUNICACIONES Centro de Pago: UM-5878 Importador: TECNOIMPORT Solicitud Puerto de destino: Ciudad Habana No. de Aprobaciones: 08.MEP.04

Código: 271.0.1056 Cuenta Bancaria: 403381101123007 Código: 271.0.0000

No	Descripción del Producto	U/M	Código del Producto	Para el consumidor Solicitado		Para el uso del importador	
				Cantidad	Valor Aprobado	Precio Proyecto	Valor Proyecto (importe)
	I + D	U	600B				1894.00
	TOTAL						1894.00

Observaciones:

Representante del Suministrador		Representante del Consumidor		Fechas		No. de Control	
Nombre	Firma	Nombre	Firma	Presentado	Aprobado	Presentado	Aprobado

Pantalla 1

Acción del Actor	Respuesta del Sistema
	1. El sistema muestra la Pantalla 1 con el reporte del 711 mostrando Especialista(A), Proyecto(B), Consumidor(C), Código del Consumidor(D), Centro de Pago(E), Cuenta bancaria(F), Importador(G), Código Importador(H), Puerto destino(I), No. de Aprobación(J), No. orden de los productos en el modelo(K), Descripción del producto(L), Unidad de Medida(M), Código del producto(N), Cantidad de producto(O),

ANEXOS

DLRE (Caney) Modelo 711 FAR

SOLICITUD DE ABASTECIMIENTO IMPORTADOS

Consumidor: DIR. COMUNICACIONES Centro de Pago: UM-5878 Importador: TECNOIMPORT Solicitud Puerto de destino: Ciudad Habana No. de Aprobaciones: 08.MEP.04

Código: 271.0.1056 Cuenta Bancaria: 403381101123007 Código: 271.0.0000

No	Descripción del Producto	U/M	Código del Producto	Para el consumidor Solicitado		Para el uso del importador	
				Cantidad	Valor Aprobado	Precio Proyecto	Valor Proyecto (Importe)
	I + D	U	600B				1894.00
	TOTAL						1894.00

Observaciones:

Representante del Suministrador	Representante del Consumidor		Fechas		No. de Control	
	Nombre	Firma	Presentado	Aprobado	Presentado	Aprobado

Pantalla 1

Acción del Actor	Respuesta del Sistema
	<p>1. El sistema muestra la Pantalla 1 con el reporte del 711 mostrando Especialista(A), Proyecto(B), Consumidor(C), Código del Consumidor(D), Centro de Pago(E), Cuenta bancaria(F), Importador(G), Código Importador(H), Puerto destino(I), No. de Aprobación(J), No. orden de los productos en el modelo(K), Descripción del producto(L), Unidad de Medida(M), Código del producto(N), Cantidad de producto(O),</p>

ANEXOS

	Valor aprobado(P), Precio de Proyecto(Q), Valor proyecto(R), Observaciones(S), Nombre y firma del suministrador (T), Nombre y Firma de consumidor(U), NO. Control (V), presentado por el consumidor (W), aprobado por el consumidor(X), presentado por el suministrador (Y), aprobado por el suministrador (Z).
Poscondiciones:	

Caso de Uso:	Recuperar Contratos
Actor(es):	Especialista CB (inicia).
Propósito:	Recuperar documentos contratos usando criterios determinado de búsqueda.
Resumen:	El caso de uso inicia cuando el actor decide recuperar información de contratos. Se le mostrara la información de los mismos de acuerdo a los datos especificados por el actor.
Referencias:	RF2
Casos de uso asociados	Mostrar Reporte Contrato (extensión), Mostrar Reporte estado Ejecución Contrato (extensión)
Precondiciones:	El usuario levanta el sistema y se encuentra en la página principal. Debe haber solicitado el servicio de autenticación.
Pantalla 1	
Acción del Actor	Respuesta del sistema
1. El usuario operación a realizar.	2. Si elige opción Recuperar Contratos, ver sección "Recuperar Contratos".
Sección "Recuperar Contratos"	

The screenshot shows a web application interface for recovering contracts. It features several input fields and buttons:

- Form Fields:**
 - No. Contrato (A):** A text input field.
 - Proveedor (B):** A dropdown menu.
 - Consumidor (C):** A dropdown menu.
 - Fecha Firma (D):** A date input field.
 - Rate MIN (F):** A text input field.
 - Rate MAX (G):** A text input field.
 - General (H):** A dropdown menu.
 - Criterio (I):** A dropdown menu.
- Table (N):** A table with columns for 'No', 'Proveedor', and 'Fecha de Firma'. The table is currently empty.
- Buttons (J, K, M):** Three buttons at the bottom: 'Recuperar' (J), 'Mostrar Reporte' (K), and 'Cancelar' (M).
- Other Elements:** A grey rectangular area (L) is located at the bottom left of the interface.

Pantalla 2

	<p>1. El sistema muestra la Pantalla 2, con los filtros: No. Contrato(A), proveedor (B), consumidor (C), Fecha Firma (D), General (H) y Criterio (I), así como las cajas de texto Rate MIN (F) Rate MAX (G) vacías, así como el gris (L) vacío.</p>
<p>2. El usuario selecciona opcionalmente una opción en cualquier filtro o introduce valores específicos en las cajas de texto Rate MIN (F) Rate MAX (G). Presiona botón Recuperar (J).</p>	<p>3. El sistema busca contratos analizando los criterios especificados por el actor y muestra los resultados en (L).</p>
<p>4. El usuario presiona botón</p>	<p>5. El sistema devuelve a Pantalla que la</p>

ANEXOS

Cancelar (K).	visualizo a ella.
6. El usuario selecciona fila (N) del grid (L) y presiona botón Mostrar Reporte (K).	7. Ver descripción del caso de uso extendido Mostrar Reporte Contrato.
Poscondiciones:	

Caso de Uso:	Mostrar Reporte Contrato
Actor(es):	Especialista CB (inicia).
Propósito:	Mostrar reporte del contrato recuperado.
Resumen:	Este caso de uso se inicia cuando el especialista CB decide mostrar reporte de documento contrato recuperado. El sistema muestra interfaz con el reporte del mismo.
Referencias:	RF2
Precondiciones:	El usuario ha recuperado documentos contratos. Debe haber solicitado servicio de autenticación.

tecnoimport
EMPRESA CUBANA IMPORTADORA Y EXPORTADORA DE PRODUCTOS TECNICOS

[Visualizar Anexos](#)

CONTRATO DE COMPRA VENTA

No. de Contrato: 17-9998-53-1056

Fecha de la firma del contrato: 31/02/2006

Descripción del producto: Medios de Redes

Nombre del usuario: DIRECCION COMUNICACIONES (DC)

No. del pedido	Aprobación	Código de la aprobación	Importe CUC
20060300	CR-P02-247	S/N	9.907.40

Proveedor: GRAN KAIMAN TELECO SA / CUBA

Importe (moneda de ctto): 10,700.00 (Dólar Estadounidense)

Rate (Fecha): 1.08000 (13/02/2006)

Importe (CUC): 9,907.40

Rate (Fecha): 1.08000 (13/02/2006)

Aprobado

JULIO GARCIA HERRADA
GÉRENTE COMERCIAL

Pantalla 1

Acción del Actor	Respuesta del Sistema
	<p>1. El sistema muestra la Pantalla 1 el reporte del documento contrato y visualiza vinculo Visualizar Anexos si el contrato tiene anexos asociados (A), No. Contrato(B), Fecha de la firma del contrato(C), Descripción del producto(D), Nombre del usuario(E), No. de pedidos(F), Aprobación(G), Código de la aprobación(H), Importe en CUC(I), Proveedor(J), Importe(K), Rate(L), Importe CUC(M), Rate(N), Gerente comercial(O).</p>
<p>2. El usuario presiona vinculo</p>	<p>3. Ver sección "Visualizar Anexos".</p>

Visualizar Anexos (A).

Sección "Visualizar Anexos"



EMPRESA CUBANA IMPORTADORA Y EXPORTADORA DE PRODUCTOS TECNICOS

ANEXO

CONTRATO DE COMPRA VENTA

No. de Contrato: 17-9998-53-1056

Moneda: Dólar Estadounidense

Anexo: 1

No	Código	Descripción	UM	Precio	Cantidad	Impte Mon.	Impte Mon. con Desc (y/o) Aum.
1	RT-SIC-1E1-F	1-PORT E1 INTERFACE CARD FRACTIONAL	U	500.0000	4	2,000.00	2,000.00
2	RT-4E1-F	4-PORT E1 MODULE FRACTIONAL	U	1,600.0000	1	1,600.00	1,600.00
3	CAB-E1-75-3M	E1 CABLE 3M-DB15 MALE/2*BNC (75 OHM)	U	45.0000	8	360.00	360.00
4	RT-1FE	1-PORT 10/100BASE-TX MODULE RJ45	U	902.5000	1	902.50	902.50
5	AR-28-90	QUIDWAY AR 28-80 ROUTER HOST 8 MIM MODULAR SLOTS AC 220V	U	1,847.5000	1	1,847.50	1,847.50
6	AR-28-10	QUIDWAY AR 28-10 ROUTER HOST 1 MIM SLOT 2 SIC SLOTS, 1 10/100 BASE-TX PORT 1 SYNC/ASYNC SERIAL PORT AC 220V	U	997.5000	4	3,999.50	3,999.50

Total Anexo: 1 10,700.00 10,700.00

Total Impte Moneda: 10,700.00 10,700.00

Descuento General:

Aumento General:

Total con descuento (y/o aumento): 10,700.00

Otros Gastos

Total general: 10,700.00

VENDEDOR

COMPRADOR

Pantalla 2

1. El sistema muestra pantalla 2 con el anexo del contrato mostrado anteriormente, muestra No. de contrato(A), Moneda(B),

	<p>No. de orden de los productos(C), Código de producto(D), Descripción(E), Unidad Medidas(F), Precio(G), Cantidad(H), Importe Monetario(I), Importe monetario con Descuento(J), Total de Anexos(K), Total Importe Moneda(L), Descuento General (M), Aumento General(N), Total con descuento(O), Total General(P), Vendedor(Q), Comprador(R).</p>
<p>Poscondiciones:</p>	

ANEXO V. Diagrama de clases del diseño.

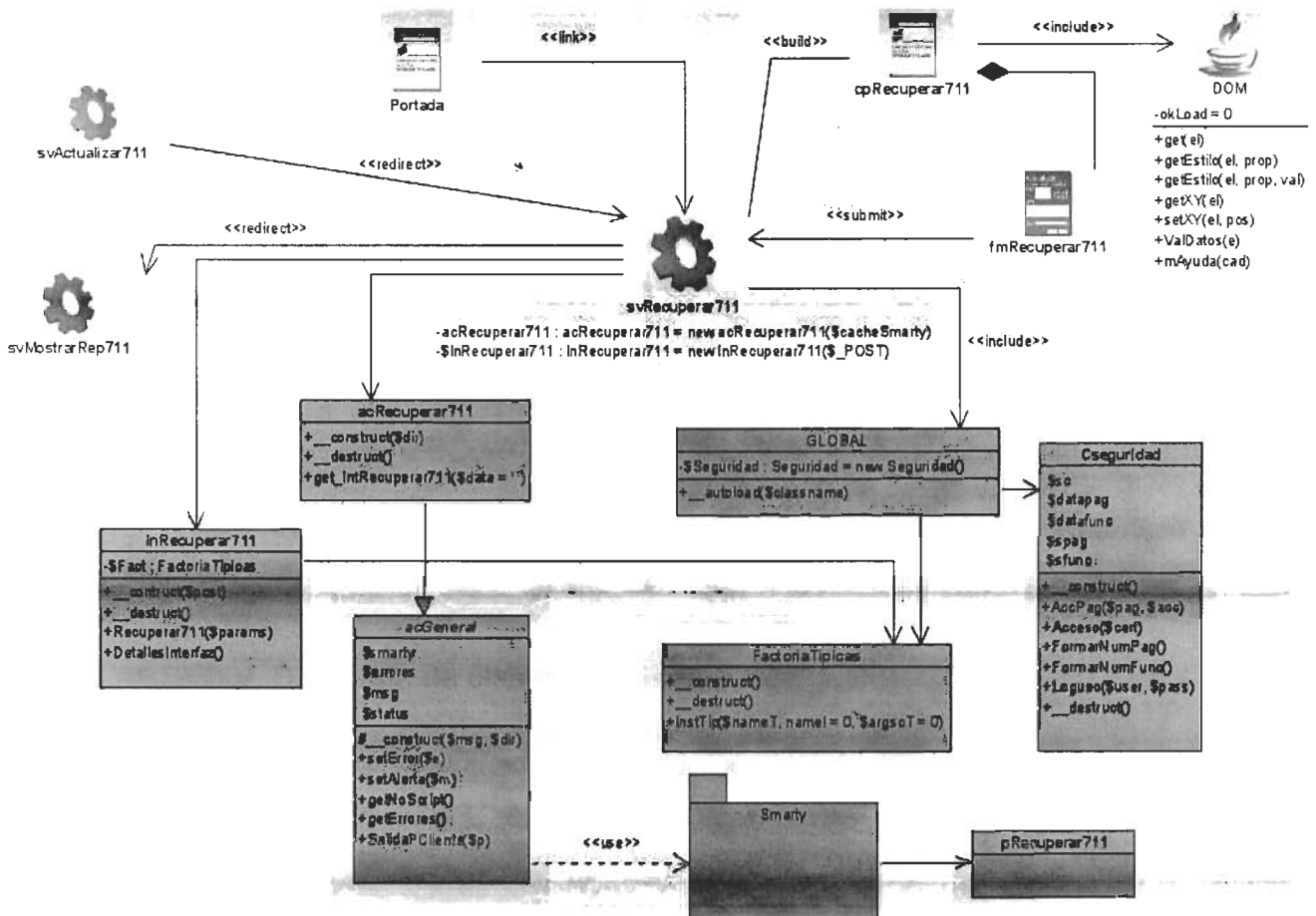


Fig. 2. Caso de uso Recuperar 711

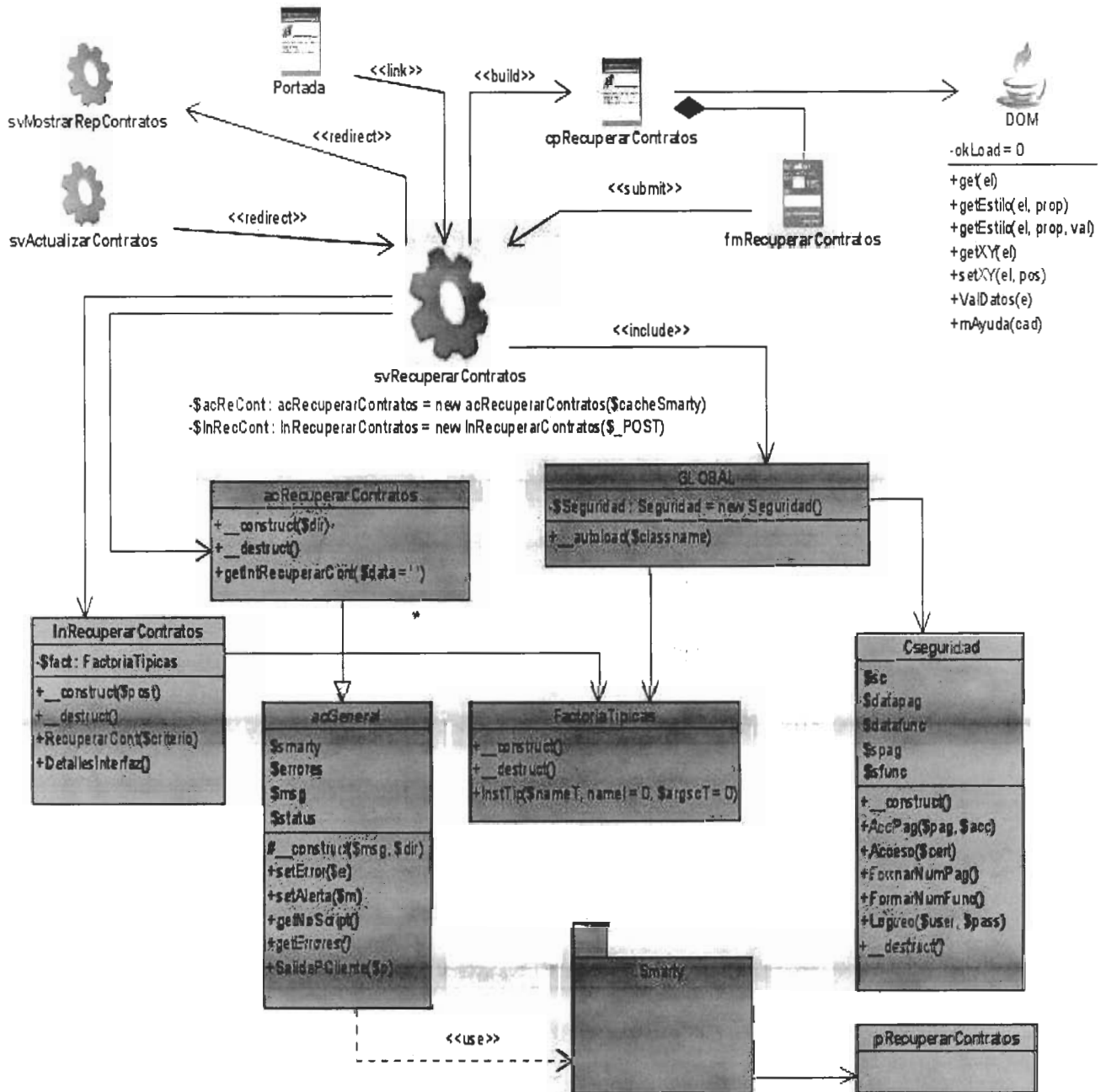


Fig. 3. Caso de uso Recuperar Contratos

ANEXO VI.

Tabla BD	Clasificación	Clase Asociada	Descripción
nom_especialista	Nomenclador	tnomEspec.php	Esta tabla almacena la información de los especialistas del CB.
nom_cuentasbanc	Nomenclador	tnomCuentB.php	Esta tabla almacena la información de las cuentas bancarias de cada centro de pago
nom_medio	Nomenclador	tnomMedio.php	Esta tabla almacena la información de los medios que intervienen en el Proceso de planificación material y financiera.
nom_cb	Nomenclador	tnomCB.php	Esta tabla almacena los Centros de Balances que intervienen en el proceso.
nom_centropago	Nomenclador	tnomCentroP.php	Esta tabla almacena toda la información de los distintos centros de pagos que existen en el MINFAR.
nom_especialistaimp	Nomenclador	tnomEsplImport.php	Esta tabla almacena la relación de los especialistas de la Empresa Importadora.
nom_importador	Nomenclador	tnomImport.php	Esta tabla almacena la relación de los Importadores existente en el proceso.
nom_suministrador	Nomenclador	tnomSumin.php	Esta tabla almacena los datos de los suministradores que intervienen en el proceso.
tmedios711	Típicas	t711.php	Esta tabla almacena todos los medios correspondientes a un 711 determinado.
tmedios711_en_contratos	Típicas	t711Med711Cont.php	Esta tabla almacena los datos de los medios de los 711 que contiene un contrato.
t711	Típica	t711.php	Esta tabla almacena los datos relacionados con los 711 existentes.
tcontrato	Típicas	tContrato.php	Esta tabla almacena los datos relacionados con los contratos.

Tabla 2. Relación de las tablas que conforman la Base de Datos y que clase tiene asociada.

ANEXO VI.

Tabla BD	Clasificación	Clase Asociada	Descripción
nom_especialista	Nomenclador	tnomEspec.php	Esta tabla almacena la información de los especialistas del CB.
nom_cuentasbanc	Nomenclador	tnomCuentB.php	Esta tabla almacena la información de las cuentas bancarias de cada centro de pago
nom_medio	Nomenclador	tnomMedio.php	Esta tabla almacena la información de los medios que intervienen en el Proceso de planificación material y financiera.
nom_cb	Nomenclador	tnomCB.php	Esta tabla almacena los Centros de Balances que intervienen en el proceso.
nom_centropago	Nomenclador	tnomCentroP.php	Esta tabla almacena toda la información de los distintos centros de pagos que existen en el MINFAR.
nom_especialistaimp	Nomenclador	tnomEsplImport.php	Esta tabla almacena la relación de los especialistas de la Empresa Importadora.
nom_importador	Nomenclador	tnomImport.php	Esta tabla almacena la relación de los Importadores existente en el proceso.
nom_suministrador	Nomenclador	tnomSumin.php	Esta tabla almacena los datos de los suministradores que intervienen en el proceso.
tmedios711	Típicas	tdat711.php	Esta tabla almacena todos los medios correspondientes a un 711 determinado.
tmedios711_en_contratos	Típicas	tdatMed711Cont.php	Esta tabla almacena los datos de los medios de los 711 que contiene un contrato.
t711	Típica	tdat711.php	Esta tabla almacena los datos relacionados con los 711 existentes.
tcontrato	Típicas	tdatContrato.php	Esta tabla almacena los datos relacionados con los contratos.

Tabla 2. Relación de las tablas que conforman la Base de Datos y que clase tiene asociada.