

Universidad de las Ciencias Informáticas “Facultad 4”



**Título: “Sistema Informático de Gestión para la Planificación de los
Órganos Consumidores del MINFAR”**

**Trabajo de Diploma para Optar por el Título de
Ingeniero en Ciencias Informáticas.**

Autores: Ilieva Rodríguez Valido (irodriguezv@estudiantes.uci.cu).

Luis Manuel González Medel (lgonzalezm@estudiantes.uci.cu).

Tutor: Ing. Raúl Miguel Triana Díaz(rtrianad@uci.cu).

Ciudad de La Habana

Junio de 2007

**Sistema de Gestión
de Planes**



"Solo los débiles se enojan. El hombre fuerte, aún al caer, sonríe".
José Martí



DECLARACIÓN DE AUTORÍA

Declaramos que somos los únicos autores de este trabajo y autorizamos al MINFAR y a la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

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

Ilieva Rodríguez Valido.

Raúl Miguel Triana Diaz.

Luis Manuel González Medel



OPINIÓN DEL TUTOR DEL TRABAJO DE DIPLOMA

Título: Sistema Informático de Gestión para la Planificación de los Órganos Consumidores del MINFAR.

Autores: Iliava Rodríguez Valido.

Luis Manuel González Medel.

El tutor del presente Trabajo de Diploma considera que durante su ejecución los estudiantes mostraron las cualidades que a continuación se detallan.

- Independencia
- Originalidad
- Creatividad
- Laboriosidad
- Responsabilidad

Por todo lo anteriormente expresado considero que los estudiantes están aptos para ejercer como Ingeniero Informático; y propongo que se le otorgue al Trabajo de Diploma la calificación de _____.

Firma

Fecha



AGRADECIMIENTOS

“El agradecimiento es la memoria del corazón”

Toda obra humana por muy humilde que sea necesita el curso de varios, jamás el hombre alcanzó la meta por sí solo, siempre fue de alguna manera asistido en el empeño, y si algo lo enaltece es la gratitud.

Por eso quisiéramos agradecerles a todas aquellas personas que hicieron posible la realización de este trabajo.

A nuestros padres quienes nos infundieron la ética y el rigor que guían nuestro transitar por la vida.

A nuestros familiares que siempre juntos nos brindaron todo el universo de amor y apoyo incondicional que nos hacía falta para no claudicar en nuestro empeño.

A las amistades por su dedicación, cariño y por su enorme apoyo emocional. Por estar siempre en los momentos precisos.

A nuestro tutor por la confianza depositada en la realización de este trabajo.

A nuestro eterno Comandante en Jefe y a la Revolución, que han hecho posible que todos tengamos igualdad de derechos y posibilidades para superarnos profesionalmente.

A todos aquellos que no vacilaron en ofrecernos su mano amiga en los momentos más difíciles para lograr este objetivo.

A todos, nuestro eterno agradecimiento con la seguridad de que siempre estarán en nuestros corazones.



DEDICATORIA

A mi madre, Paula Valido infinitamente le dedico este triunfo en mi vida, por existir, por su cariño como madre, por estar cerca aún cuando estuvimos distantes, por haberme sabido guiar y confiar siempre en que yo podía llegar hasta aquí. Por ser lo que soy.

A mis hermanos y a Raúl por formar parte de mí y darme fuerzas para seguir.

A toda mi familia por su apoyo, por ser mi fuerza, mi inspiración, mis más fieles y constantes compañeros, mis más grandes amores y por haber dado todo para hacer realidad este sueño.

Ilieva.

A mi madre Erenia Medel dedico este triunfo por ser mi principal guía, por su gran corazón y capacidad de entrega, por darme la fuerza necesaria para salir adelante y enseñarme que la perseverancia y el esfuerzo son el camino para lograr objetivos.

A mi abuela por trasmitirme su experiencia y amor a lo largo de todos estos años y por darme ánimo en los momentos de flaqueza.

A toda mi familia por su cariño y dedicación, por su comprensión, por confiar en que yo podía llegar hasta aquí. Por enseñarme a que debemos tener la fortaleza de continuar hacia adelante no importa las circunstancias que la vida nos presenta. Por ser sencillamente una familia maravillosa pues sin el apoyo de ellos no hubiese sido posible alcanzar esta meta.

Luis Manuel.



RESUMEN

Las Fuerzas Armadas Revolucionarias (FAR) cuyo objeto social es la preparación del país para la lucha armada en defensa de la Revolución están estructuradas para su funcionamiento interno por departamentos que abarcan todas las esferas del ciclo logístico empresarial. Dentro de estos departamentos se encuentra el de Planificación Material y Financiera, en el cual se efectúan un sin número de procesos y se trabaja con grandes cantidades de información, las que sirven de apoyo al proceso de toma de decisiones.

Por todo esto, es de vital importancia lograr tiempos de procesamiento breves y una veracidad en los resultados obtenidos del flujo informativo para lograr el mejor funcionamiento.

El siguiente trabajo tiene como propósito explicar el desarrollo de una aplicación de gestión Web, la cual debe elevar el nivel de informatización en el proceso de gestión de los planes de los Órganos Consumidores del MINFAR, permitiendo la distribución de esta información y asegurando que la planificación se realice de forma óptima, rápida y segura.

El desarrollo de este sistema se sostiene de la aplicación de metodologías y el uso de herramientas actuales que aseguran que su resultado sea un producto de software confiable, con un alto grado de calidad.



ÍNDICE

INTRODUCCIÓN	1
CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA	4
1.1 INTRODUCCIÓN	4
1.2 ESTADO DEL ARTE	4
1.3 SOLUCIÓN A TRAVÉS DE UNA APLICACIÓN DE GESTIÓN WEB.	7
1.3.1 <i>Arquitectura Cliente-Servidor</i>	7
1.4 TÉCNICA Y TECNOLOGÍAS DEL LADO DEL CLIENTE.....	8
1.4.1 <i>Script del lado del cliente</i>	8
1.4.2 <i>AJAX: Una técnica de desarrollo Web</i>	9
1.4.3 <i>Hojas de Estilo en Cascada</i>	11
1.5 EXTENSIBLE MARKUP LANGUAGE (XML).....	12
1.6 TECNOLOGÍAS DEL LADO DEL SERVIDOR.....	12
1.6.1 <i>Módulo de los servidores Web</i>	13
1.6.2 <i>Script del lado del servidor</i>	13
1.7 GESTIÓN DE DATOS.	14
1.7.1 <i>Lenguaje de Consulta Estructurado</i>	15
1.7.2 <i>Sistema Gestor de Base de Datos</i>	15
1.8 PROCESO DE DESARROLLO DE SOFTWARE.	16
1.8.1 <i>Metodologías de desarrollo de software</i>	16
1.8.2 <i>Lenguaje de Modelado</i>	19
1.9 HERRAMIENTAS CASE.	20
1.10 HERRAMIENTAS DE APOYO.....	21
1.11 CONCLUSIONES.....	22
CAPÍTULO 2. CARACTERÍSTICAS DEL SISTEMA	23
2.1 INTRODUCCIÓN.	23
2.2 PROPUESTA DEL SISTEMA.....	23
2.3 MODELADO DEL NEGOCIO.	24
2.4 ACTOR Y TRABAJADORES DEL NEGOCIO.....	24
2.5 DIAGRAMA DE CASOS DE USO DEL NEGOCIO.....	25
2.6 DESCRIPCIONES TEXTUALES.	26
2.7 DIAGRAMA DE ACTIVIDADES.....	28



2.8 DIAGRAMAS DE CLASES DEL MODELO DE OBJETOS.....	30
2.9 REQUERIMIENTOS.	30
2.9.1 <i>Requerimientos Funcionales</i>	31
2.9.2 <i>Requerimientos No Funcionales</i>	32
2.10 MODELACIÓN DEL SISTEMA.....	35
2.11 ACTORES DEL SISTEMA.....	35
2.12 DIAGRAMA DE CASOS DE USO DEL SISTEMA.	36
2.13 ESPECIFICACIONES DE LOS CASOS DE USOS.....	37
2.14 CASOS DE USO EXPANDIDOS.	41
2.15 CONCLUSIONES.....	65
CAPÍTULO 3. ANÁLISIS Y DISEÑO DEL SISTEMA	66
3.1 INTRODUCCIÓN.....	66
3.2 MODELO DE ANÁLISIS.	66
3.3 MODELO DE DISEÑO.	69
3.4 ARQUITECTURA.	70
3.5 MECANISMO DE DISEÑO.	73
3.5.1 <i>Mecanismo de Seguridad</i>	74
3.5.2 <i>Mecanismo de Acceso a Datos</i>	75
3.6 DIAGRAMAS DE INTERACCIÓN.	78
3.7 DIAGRAMA DE CLASES DEL DISEÑO.	83
3.8 DISEÑO DE LA BASE DE DATOS.....	88
3.10 ESTÁNDARES DE DISEÑO.	89
3.11 ESTÁNDARES DE PROGRAMACIÓN.	90
3.12 TRATAMIENTO DE ERRORES.....	94
3.13 CONCLUSIONES.....	94
CAPÍTULO 4. IMPLEMENTACIÓN Y PRUEBA.....	95
4.1 INTRODUCCIÓN.....	95
4.2 MODELO DE IMPLEMENTACIÓN.....	95
4.3 MODELO DE DESPLIEGUE.....	97
4.4 MODELO DE PRUEBA.	98
4.5 CONCLUSIONES.....	103
CONCLUSIONES	104



RECOMENDACIONES	105
REFERENCIAS BIBLIOGRÁFICAS.....	106
BIBLIOGRAFÍA	107
GLOSARIO DE TÉRMINOS.....	109



INTRODUCCIÓN

La Planificación es el proceso mediante el cual se establecen metas organizacionales, se definen estrategias y políticas para lograr estas metas, se desarrollan planes detallados para asegurar la implantación de las estrategias y así obtener los fines buscados, no es más que diseñar un futuro deseado e identificar las formas y los recursos para lograrlo.

En la actualidad en cualquier proceso se requiere planificar, para lograr los objetivos y las metas propuestas. Los planes pueden ser a corto, mediano y largo plazo, según la amplitud y magnitud del proceso que se va a desarrollar.

Este proceso de planificación constituye un sistema que hace énfasis en el *qué lograr* (objetivos) y en el *qué hacer* (estrategias). Es decir, busca centrarse en aquellos objetivos factibles que te permitan obtener el éxito, definiendo para ello las acciones a realizar y los problemas a resolver, priorizando soluciones, estableciendo recursos y responsabilidades, y diseñando medidas de seguimiento y control que permitan la evaluación de los avances.

La esencia de la planificación consiste en la identificación sistemática de las oportunidades y peligros que surgen en el futuro, los cuales combinados con otros datos importantes proporcionan la base para que una empresa tome mejores decisiones.

En nuestro país el Ministerio de las Fuerzas Armadas Revolucionarias (en lo adelante MINFAR) realiza un proceso de previsión, estructuración y asignación de recursos a los diferentes organismos vinculados al mismo, para lograr una organización y optimización de los recursos materiales y financieros con el fin de cumplir los objetivos de trabajo y las metas trazadas en un tiempo determinado, proceso al que define como Planificación Material y Financiera en el MINFAR.

Dentro de este proceso se desarrolla la gestión de los Planes de Recursos Materiales y Financieros de los Órganos Consumidores (en lo adelante OC), donde estos basándose en los lineamientos generales definidos por la Dirección de Economía (en lo adelante DE) y teniendo en cuenta las experiencias adquiridas en años anteriores, así como los volúmenes de recursos materiales y presupuestarios existentes; crean sus indicaciones específicas para conformar los planes del año venidero. Una vez



conformados los planes son aprobados por el Jefe del Órgano Consumidor para enviarlo al Centro de Balance correspondiente donde son conciliados y a la Dirección de Economía para su aprobación final concluyendo así el proceso de conformación de los planes.

En el mismo participan gran variedad de entidades y el procesamiento, análisis e intercambio de información se realiza de forma manual o con herramientas ofimáticas. Esto hace que se genere gran cantidad de información dispersa y que el flujo de información sea lento. Además conlleva a que la información no esté disponible en todo momento y que en ocasiones no puedan tomarse decisiones con la objetividad, veracidad y rapidez que se requiere.

De lo expresado anteriormente se puede deducir que la no existencia de un sistema informático con un enfoque integral del problema, genera toda una gama de problemas que lastran la eficiencia del proceso de Planificación Material y Financiera. Por lo cual surge el siguiente **problema**: ¿Cómo centralizar los datos y garantizar la disponibilidad de la información, en el proceso de Gestión de los Planes de Demandas de los OC del MINFAR?

Este problema se enmarca en el **objeto de estudio**: Proceso de Planificación Material y Financiera de los Órganos Consumidores del MINFAR.

Para resolver el problema planteado se propone como **objetivo**: Diseñar e implementar un sistema informático que favorezca el proceso de gestión de los planes de los OC en el MINFAR.

Para lograr este objetivo se trazan los siguientes **objetivos específicos**:

1. Modelar los principales procesos del negocio y analizar la posibilidad de proponer mejoras a los mismos.
2. Realizar el análisis, diseño e implementación del sistema utilizando una metodología de ingeniería de software.
3. Mantener la estructura interna del proceso de gestión de los planes de demandas de los Órganos Consumidores con que cuenta hoy el MINFAR.
4. Mantener sus niveles de jerarquía y el flujo de información que hay entre ellos.
5. Informatizar las actividades vinculadas al almacenamiento, análisis e intercambio de información.



El objetivo que delimita el **campo de acción** es el proceso de Gestión de los Planes de de los Órganos Consumidores del MINFAR.

Para guiar la investigación se plantea la siguiente **hipótesis**: Si se cuenta con una aplicación de gestión Web multiplataforma para el proceso de gestión de los planes de los OC del MINFAR, que se identifique por la interconexión y la integración en el flujo de la información de los organismos en red, con un grado de uniformidad y consistencia, entonces se logrará centralizar y mantener disponible la información, permitiendo un eficiente proceso.

Para dar cumplimiento a los objetivos anteriormente planteados se definen las siguientes **tareas**:

1. Estudio del Manual de Normas y Procedimientos por los que se rigen los procesos de los OC del MINFAR.
2. Desarrollar análisis de los procesos del sistema.
3. Desarrollar diseño del sistema.
4. Implementar la aplicación para la Conformación de los Planes de Recursos Materiales y Financieros de los Órganos Consumidores del Ministerio de las FAR.



CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

1.1 Introducción

En el presente capítulo se ofrece una visión de las principales características de la Planificación Material y Financiera en el MINFAR, así como los principales problemas que afectan el campo de acción de este trabajo y la incidencia que puede tener la propuesta de solución.

Además se abordan importantes conceptos que son necesarios conocer para la creación de una aplicación de gestión web y para entender correctamente en qué consiste la misma. También se hace referencia a las principales características y funcionalidades de algunas tecnologías empleadas en la construcción de la misma.

1.2 Estado del Arte.

En la actualidad, las empresas enfrentan nuevos desafíos, entre ellos, cubrir necesidades de información mayores día a día y, aunque la estadística, la computación y la ingeniería están muy desarrolladas es un proceso que se produce constantemente. Casi nunca se posee toda la información que se necesita para tomar decisiones, esto es debido en gran parte a que los procesos son altamente variables. Esta dinámica, ligada a la necesidad de obtener utilidades, obliga a las empresas a requerir más y mejor información y a diseñar sistemas que les permitan conseguirla y administrarla de la mejor manera. Para hacer eficiente el procesamiento de la información, conseguirla, analizarla y tomar decisiones basados en ella, se desarrollan en las empresas los Sistemas de Información (**SI** [22]).

El conocimiento sobre sistemas de información es esencial porque la mayoría de las organizaciones requieren información, entre otras cosas, para controlar sus actividades, tomar decisiones, crear nuevos servicios, evaluar posibilidades de expansión, etc. y necesitan diseñar sistemas para cubrir los diferentes servicios y funciones de la empresa. La mejor práctica para lograr integrar estos es a través de Sistemas Empresariales (**ERP** [6]), que proporcionan una plataforma de tecnología en la que las organizaciones pueden integrar y coordinar sus principales procesos internos de negocios evitando las ineficiencias organizacionales creadas por el aislamiento de la información, procesos de negocios y tecnología, que tienen en muchas ocasiones, un impacto negativo en la eficiencia y desempeño de los negocios de la organización.



Los ERP están funcionando ampliamente en todo tipo de empresas modernas. Todos los departamentos funcionales que están involucrados en la operación o producción están integrados en un solo sistema.

En general, un ERP es una herramienta de apoyo que proporciona información en tiempo real con la cual se pueden tomar mejores decisiones.

Con la implementación de un ERP se obtienen, entre otros, los siguientes beneficios:

- Eliminación de las capturas repetitivas.
- Mayor oportunidad en la información.
- Consistencia, congruencia e integridad de la información.
- Eliminación de tiempos de espera para consulta de la información.
- Operaciones multi-usuario confiables y seguras.
- Reducción de los costos de operación.
- La posibilidad de compartir información entre todos los componentes de la organización. (1)

Varias empresas en el mundo han optado por la implementación de un ERP, y los resultados obtenidos son satisfactorios en varios casos, ejemplo:

- El ERP de la empresa Sellex, que permite tener todos los procesos de negocio, desde el ámbito financiero hasta las compras, pasando por los procesos de producción y calidad, en constante modernización.
- La corporación Dalton, quien vio la necesidad de invertir para conectar sus diferentes sucursales en la República Mexicana y se convirtió en líder en la venta de autos y motocicletas en la región occidental de México.
- SYC motors, que es ahora una importante y reconocida agencia de autos en Guadalajara, ya que cambiaron su estrategia de mercado, enfocándola al mejor servicio al cliente y fue con la implementación de un ERP que lograron su objetivo.
- Otro caso de éxito que se puede mencionar es el de la compañía Jaguar en España, que adoptó un sistema ERP para dar una mejor solución al rápido crecimiento que estaba teniendo.



Por todas las ventajas que proporcionan y las necesidades que satisfacen los ERP, el MINFAR ha decidido implementar un Sistema Empresarial para lograr la integridad de todas sus áreas de trabajo y facilitar la toma de decisiones de los directivos; pues el ERP gestiona de manera integrada y eficiente la información, comunicando las diferentes áreas del negocio mediante procesos electrónicos. Una de esas áreas es la de Planificación de los Recursos Materiales y Financieros.

Su función principal es organizar y estandarizar procesos y datos internos, transformándolos en información útil para ser analizados y tener en cuenta toma de decisiones.

El desempeño de dicha actividad posee ciertas características, entre las cuales se debe resaltar las siguientes:

1. Es un proceso dinámico: toda decisión tomada originará una cadena de eventos que a su vez requerirá de subsiguientes tomas de decisiones.
2. Debe ser continuo y cerrado; debe pasar de la planeación a la implementación, y luego al control. A medida que se dispone de nueva información, debe regresar a la planeación.
3. Las metas deben implicar un compromiso y ser factibles, específicas, cuantificables y verificables a lo largo del tiempo.
4. Debe estar siempre focalizada hacia los objetivos.

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 informativa de los diferentes procesos que componen el sistema de planificación, registro y control de los recursos. Dentro del mismo, el proceso de informatización de la Gestión de los Planes de Recursos Materiales y Financieros de los OC en el MINFAR aún se encuentran ajenos a muchos de los progresos que brindan las aplicaciones informáticas, impidiendo que la Planificación Material y Financiera sea todo lo eficiente que pudiera ser si se contara con un sistema informático que pueda darle solución a los problemas objetivos que enfrentan los OC a través de una aplicación de gestión.



1.3 Solución a través de una aplicación de gestión Web.

Una aplicación Web es aquella que los usuarios usan accediendo a un servidor Web a través de la red (**Internet** [10] o **intranet** [11]). Las aplicaciones Web son populares debido a la practicidad del navegador como cliente ligero. La habilidad para actualizar y mantener aplicaciones Web sin distribuir e instalar software en miles de potenciales clientes es otra razón de su popularidad.

En una aplicación Web la navegación y la entrada de datos por parte de un usuario, afectan el estado de la **lógica del negocio** [14]; empleando para ello tecnologías que generan contenidos dinámicos. Si no existe lógica del negocio en el servidor, el sistema no es considerado aplicación Web.

Las interfaces de las páginas Web poseen ciertas limitantes en la funcionalidad del cliente. Métodos comunes en las aplicaciones de escritorio como dibujar en la pantalla, arrastrar o soltar no están soportados por las tecnologías Web estándar. Los desarrolladores utilizan generalmente lenguajes interpretados que se ejecutan en el cliente Web para obtener una mayor funcionalidad, así como tecnologías que se ejecutan en el servidor para no tener que recargar la página en su totalidad, algo que molesta mucho a los usuarios. Recientemente se han desarrollado técnicas para coordinar estos lenguajes con tecnologías del lado del servidor, como por ejemplo PHP y AJAX que es una técnica de desarrollo web que usa una combinación de varias tecnologías como se verá más adelante.

1.3.1 Arquitectura Cliente-Servidor.

La arquitectura cliente-servidor es una forma de dividir las responsabilidades de un Sistema de Información separando la interfaz de usuario (Nivel de presentación) de la gestión de la información (Nivel de gestión de datos).

Esta arquitectura consiste básicamente en que un programa, el Cliente informático realiza peticiones a otro programa, el servidor, que les da respuesta.

Aunque esta idea se puede aplicar a programas que se ejecutan sobre una sola computadora es más ventajosa en un sistema multiusuario distribuido a través de una red de computadoras.



1.4 Técnica y tecnologías del lado del cliente.

Un cliente Web (navegador o browser) es una aplicación de software que permite al usuario recuperar y visualizar documentos de hipertexto. La parte cliente de las aplicaciones Web está formada por el código HTML que forma la página Web, con opción a código ejecutable mediante los lenguajes de scripting de los navegadores (JavaScript, Vbscript, etc.) o mediante pequeños programas en Java (applets). La programación del lado del cliente tiene como principal ventaja que la ejecución de la aplicación se delega al cliente, con lo cual se evita recargar al servidor de trabajo. Entre los navegadores más populares están, el Netscape, Internet Explorer, Mozilla. Los lenguajes más comunes que se ejecutan en el cliente son: *VBScript* y JavaScript.

1.4.1 Script del lado del cliente.

Una práctica muy difundida en el mundo de la programación web es emplear **lenguajes interpretados** [13] del lado del cliente para añadir más funcionalidad a las aplicaciones, especialmente para crear una experiencia interactiva que no requiera recargar la página cada vez.

JavaScript

JavaScript es un lenguaje interpretado, con una sintaxis semejante a la de los lenguajes Java y C. Su diseño no le permite considerarse un lenguaje puramente orientado a objetos como es el caso de Java. Fue desarrollado por la empresa Netscape Communications. Es muy utilizado para controlar la apariencia y manipular los eventos dentro de la ventana del navegador Web así como para validar datos de entrada en las interfaces de las aplicaciones.

¿Por qué JavaScript?

Gracias a su compatibilidad con la mayoría de los navegadores modernos, es el lenguaje de programación del lado del cliente más utilizado. Con Javascript se pueden crear efectos especiales en las páginas y definir interactividades con el usuario. El navegador del cliente es el encargado de interpretar las instrucciones Javascript y ejecutarlas para realizar estos efectos e interactividades, de modo que el mayor recurso, y tal vez el único, con que cuenta este lenguaje es el propio navegador.



Para interactuar con una página Web se provee al lenguaje JavaScript de una implementación del Modelo de Objetos de Documento, frecuentemente abreviado **DOM** [5], es una interfaz de programación de aplicaciones (en lo adelante API) para documentos HTML (paginas Web) y **XML** [25]. Define la estructura lógica de los documentos y el modo en que se accede y manipula un documento. En la especificación del DOM, el término "documento" se utiliza en un sentido amplio ya que es el documento el contenedor que soporta los demás elementos. A través del DOM los programadores pueden construir documentos, navegar por su estructura, añadir, modificar o eliminar elementos y contenido. Se puede acceder a cualquier elemento que se encuentre en un documento HTML o XML, y se puede modificar, eliminar o añadir usando DOM, salvo algunas excepciones. (2)

1.4.2 AJAX: Una técnica de desarrollo Web.

JavaScript y XML Asíncronos (*Asynchronous JavaScript And XML*, en lo adelante AJAX) es una técnica de desarrollo Web para crear aplicaciones interactivas. Éstas se ejecutan en el cliente y mantiene comunicación **asíncrona** [2] con el servidor en segundo plano. De esta forma es posible realizar cambios sobre la misma página sin necesidad de recargarla. Esto significa aumentar la interactividad, velocidad y usabilidad en la misma.

AJAX es una combinación de tres tecnologías ya existentes:

- **XHTML** [24] (o HTML) y CSS para el diseño que acompaña a la información.
- DOM accedido con un lenguaje de scripting por parte del usuario, como JavaScript y JScript, para mostrar e interactuar dinámicamente con la información presentada.
- El objeto **XMLHttpRequest** [26] para intercambiar datos asincrónicamente con el servidor web.
- XML es el formato usado comúnmente para la transferencia de vuelta al servidor, aunque cualquier formato puede funcionar, incluyendo HTML preformateado, texto plano.

Propiedades.

- Peticiones y respuestas **HTTP** [8].

- Código de lado del cliente usando JavaScript.



- Programación de lado del servidor en PHP.
- Transferir y procesar datos mediante el uso de XML.

Libertad del refresco de página:

Las aplicaciones Ajax no necesitan refrescar la página completa para actualizar la información, pueden simplemente actualizar parte de la página en cualquier momento, dándoles a los usuarios una respuesta instantánea a sus ingresos y consultas. Esto les permite a los usuarios ver continuamente con lo que están trabajando y reaccionar a cualquier cambio, error, o actualización que la interfaz les notifique.

Chequeo y guardado instantáneo del campo:

En la web, es fácil chequear los campos en el lado del cliente utilizando JavaScript. Esto produce un efecto inmediato, y replica el comportamiento de una aplicación de escritorio. Sin embargo, por razones de seguridad, es necesario chequear todos los campos también del lado del servidor. Afortunadamente, Ajax permite también que esto suceda.

Interfaz de pantalla única:

Una de las mayores razones para utilizar el abordaje basado en pantallas es la simplicidad de una interfaz de pantalla única. Una interfaz de pantalla única es más útil, brindando varias ventajas sobre las aplicaciones basadas en páginas.

Una de las ventajas de pantalla única es que el usuario puede ver el cuadro general de la aplicación, viendo todos los pasos necesarios para completar la aplicación. Esto le da al usuario una clara idea de lo que se espera de ellos durante una transacción.

La interfaz de pantalla única también le permite al usuario modificar y cambiar información en el orden que él elija.

Relativamente fácil de implementar:

Debido a que las aplicaciones Ajax están construidas utilizando nada más que estándares web actuales, ellas son relativamente sencillas de construir.



Muchos diseñadores web familiarizados con la construcción de aplicaciones basadas en páginas pueden migrar de una interfaz a Ajax más bien rápidamente.

También, desarrolladores emprendedores de Ajax crearon bloques de construcción fáciles de usar que les permiten a aquellos desarrolladores que no estén familiarizados con el abordaje migrar sus aplicaciones sin tener que escribir código desde cero.

Ajax: una buena alternativa:

Combinando la sofisticación de aplicaciones basadas en pantallas con la relativamente fácil de implementar de aplicaciones basadas en página, Ajax es una alternativa sólida para un nuevo desarrollo de interfaz.

A pesar de que nada construido con Ajax va a ser amigable para el usuario desde el comienzo, con cuidado las interfaces pueden sacar provecho de lo que se sabe y gusta de las aplicaciones de escritorio, y al mismo tiempo todavía sentir que se está usando la maravillosa web. (3)

1.4.3 Hojas de Estilo en Cascada.

Las hojas de estilo en cascada o Cascading Style Sheets (en lo adelante **CSS** [4]) son un lenguaje formal usado para definir la presentación de un documento estructurado escrito en HTML o XML (y por extensión en XHTML). La virtud del desarrollo de CSS es separar la estructura de un documento de su presentación. La información de estilo puede ser adjuntada tanto como un documento separado o en el mismo documento HTML. En este último podrían definirse estilos generales en la cabecera del documento o en cada etiqueta particular mediante el atributo "style".

Las ventajas de utilizar CSS (u otro lenguaje de estilo) son:

- Control centralizado de la presentación de un sitio Web completo con lo que se agiliza de forma considerable la actualización de la misma.



- Los navegadores permiten a los usuarios especificar su propia hoja de estilo local que será aplicada a un sitio Web remoto. Así por ejemplo, personas con deficiencias visuales pueden configurar su propia hoja de estilo para aumentar el tamaño del texto o remarcar más los enlaces.
- El documento HTML en sí mismo es más claro de entender y se consigue reducir considerablemente su tamaño.

1.5 Extensible Markup Language (XML).

XML es un metalenguaje, es decir, un lenguaje usado para hacer referencia a otros lenguajes que ofrece un formato para la descripción de datos estructurados. Esto facilita unas declaraciones de contenido más precisas y unos resultados de búsquedas más significativos en varias plataformas. Además, XML habilita una nueva generación de aplicaciones para ver y manipular datos basadas en la Web.

Se trata de una tecnología sencilla que tiene a su alrededor otras que la complementan y la hacen mucho más grande y con unas posibilidades mucho mayores. Tiene un papel muy importante en la actualidad ya que permite la compatibilidad entre sistemas para compartir la información de una manera segura, fiable y fácil.

1.6 Tecnologías del lado del servidor.

La parte del servidor está formada por un programa o script que es ejecutado por el servidor web, y cuya salida se envía al navegador del cliente. Los principales beneficios de la aplicación de la tecnología de servidores de aplicación son la centralización y la disminución de la complejidad en el desarrollo de aplicaciones.

Programar del lado del servidor tiene como gran ventaja que cualquier acción puede ejecutarse sin tener en cuenta el tipo de cliente, ya que la aplicación se ejecuta en el servidor que es un ambiente controlado. Una vez ejecutada la aplicación, el resultado que se envía al cliente.



1.6.1 Módulo de los servidores Web.

La tecnología más reciente para la ejecución de aplicaciones Web consiste en anexar a un servidor Web "módulos" que permiten al servidor interpretar un determinado lenguaje. De esta forma se logra eficiencia ya que el servidor no necesita crear un nuevo proceso por cada aplicación que ejecuta. Las aplicaciones son portables ya que son desarrolladas en un lenguaje estándar que no depende del servidor, y confiables ya que si bien pueden producir un error en el lenguaje en que están diseñadas si el módulo es sólido dichos errores no pueden comprometer al servidor Web. Por la potencialidad, rendimiento y seguridad que ofrece esta tecnología será la adoptada para el desarrollo de la aplicación que se propone como objetivo principal este trabajo.

1.6.2 Script del lado del servidor.

Los lenguajes de scripts que se ejecutan en el servidor Web son interpretados por alguna aplicación alojada en dicho servidor y que se envían al cliente en un formato comprensible para él, son independientes del cliente por lo que son mucho más flexibles en relación al cambio de un navegador a otro o con respecto a las versiones del mismo. El cliente solo verá el código HTML terminado.

PHP

PHP es un lenguaje de programación usado generalmente para la creación de contenido para sitios web. PHP es un acrónimo recurrente que significa "**PHP Hypertext Pre-processor**" (inicialmente PHP Tools, o, Personal Home Page Tools), y se trata de un lenguaje interpretado usado para la creación de aplicaciones para servidores, o creación de contenido dinámico para sitios web.

¿Por qué PHP?

Para el desarrollo de la aplicación Web se debe tener en cuenta cuestiones esenciales como: garantizar que el 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 por su potencialidad en funcionalidades y rapidez.



Dentro de las principales ventajas de PHP se encuentran:

- Es un lenguaje de programación multiplataforma.
- Capacidad de conexión con la mayoría de los manejadores de base de datos que se utilizan en la actualidad, destaca su conectividad con MySQL, PostgreSQL (soluciones libres).
- Lee y manipula datos desde diversas fuentes.
- Capacidad de expandir su potencial utilizando la enorme cantidad de módulos (extensiones).
- Se desarrolla activamente, cuenta con una amplia documentación en su página oficial.
- Es libre, por lo que se presenta como una alternativa de fácil acceso para todos.
- Permite las técnicas de **Programación Orientada a Objetos** [20].
- Permite crear los formularios para la web.
- Biblioteca nativa de funciones sumamente amplia e incluida.
- No requiere definición de tipos de variables ni manejo detallado del bajo nivel. (6)

1.7 Gestión de Datos.

Un Sistema Gestor de Bases de Datos (en lo adelante SGBD) o Sistema Gestor de Base de Datos Relacionales (RDBMS) es un conjunto de programas que permite a los usuarios crear y mantener una base de datos (en lo adelante BD), por lo tanto, el SGBD es un software de propósito general que facilita el proceso de definir, construir y manipular la BD para diversas aplicaciones así como la administración necesaria para mantenerlas operativas, mantener su integridad, consistencia, confidencialidad y seguridad.

El propósito general de los sistemas de gestión de base de datos es el de manejar de manera clara, sencilla y ordenada un conjunto de información.



1.7.1 Lenguaje de Consulta Estructurado (Structured Query Language, en lo adelante SQL).

Es un lenguaje de alto nivel, declarativo, optimizado y relacional ya que se emplea para manipular datos en BD relacionales, que permite realizar diversas operaciones sobre estas. En la actualidad el SQL se ha convertido en el estándar de la mayoría de los SGBD. Sus funcionalidades van más allá de ser un simple lenguaje de consultas a BD, estas comprenden además: lenguaje definición de datos (DDL), lenguaje de definición de vistas (VDL), y lenguaje de manipulación de datos (DML). Además permite la concesión y denegación de permisos, la implementación de restricciones de integridad y controles de transacción, y la alteración de esquemas.

El SQL trabaja con estructura cliente/servidor sobre una red de ordenadores, donde el ordenador cliente es el que inicia la consulta y el ordenador servidor es quien la atiende.

El SQL permite:

- Definir una base de datos mediante tablas.
- Almacenar información en tablas.
- Seleccionar la información que sea necesaria de la base de datos.
- Realizar cambios en la información y estructura de los datos.
- Combinar y calcular datos para conseguir la información necesaria.

1.7.2 Sistema Gestor de Base de Datos.

¿Por qué PostgreSQL?

PostgreSQL [19] 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.

Soporta tanto la programación orientada a objetos como la funcional. Las funciones tienen la particularidad de poder ejecutarse con los privilegios del usuario ejecutor o con los privilegios de un usuario definido previamente lo que supone un alto nivel de seguridad. Estas funciones son referidas en otros SGBD como procedimientos almacenados (stored procedures). PostgreSQL tiene la



extraordinaria potencialidad de permitir que mientras un proceso escribe en una tabla, otros accedan a la misma sin necesidad de bloqueos esto es posible gracias a un sistema denominado **Acceso Concurrente Multiversión (MVCC [15] en ingles)**. (4) Este gestor se identifica además por:

- Disparadores (triggers).
- Vistas.
- Integridad transaccional.
- Herencia de tablas.
- Tipos de datos y operaciones geométricas.

1.8 Proceso de Desarrollo de Software.

El proceso de desarrollo de software requiere por un lado un conjunto de conceptos, una metodología y un lenguaje propio. Este es el encargado de guiar el desarrollo por cada uno de los pasos que garanticen un producto final de calidad. Debe brindar la posibilidad al equipo de desarrollo de documentar lo aspectos esenciales para en un futuro dar soporte al software.

1.8.1 Metodologías de desarrollo de software.

Un proceso de software detallado y completo suele denominarse “Metodología”. Sin embargo existen numerosas propuestas metodológicas que inciden en distintas dimensiones del proceso de desarrollo. Vale destacar que todo desarrollo de software es riesgoso y difícil de controlar, por lo que es necesario aplicar una metodología que permita obtener como resultado un producto de calidad. Una definición acertada, en correspondencia con el objetivo de la investigación es la abordada por Maddison donde define metodología como “Conjunto de filosofías, fases, procedimientos, reglas, técnicas, herramientas, documentación y aspectos de formación para los desarrolladores de Sistemas de Información”.

Las metodologías guían el proceso de desarrollo y la clave del éxito de un proyecto de software es la elección correcta de la metodología, que puede conducir al programador a desarrollar un buen sistema de software, logrando el crecimiento de la calidad y la productividad del producto. La elección de la metodología adecuada es más importante que utilizar las mejores y más potentes herramientas.



Metodología para el desarrollo unificado, Rational Unified Process (RUP)

El Proceso Unificado de Desarrollo (**RUP**) [21] es un proceso de ingeniería de software que mejora la productividad del equipo de trabajo y entrega las mejores prácticas del software a todos los miembros del mismo. RUP, es una metodología basada en un pequeño grupo de principios claves: el equipo de un proyecto de software debe planificar el desarrollo; debe conocer hacia donde se dirige; debe documentar el proyecto de una manera perdurable y extensible.

RUP es un proceso de desarrollo de software donde durante el paso por las diferentes etapas de desarrollo se van a “transformar los requerimientos de los usuarios en un sistema software”. La metodología RUP, llamada así por sus siglas en inglés Rational Unified Process, divide en 4 fases el desarrollo del software.

1. **Fase de Conceptualización:** El Objetivo en esta etapa es determinar la visión del proyecto.
2. **Fase de Elaboración:** En esta etapa el objetivo es determinar la arquitectura óptima.
3. **Fase de Construcción:** En esta etapa el objetivo es llegar a obtener la capacidad operacional inicial.
4. **Fase de Transición:** El objetivo es llegar a obtener el release del proyecto.

Cada una de estas etapas es desarrollada mediante el ciclo de iteraciones. Los objetivos de una iteración se establecen en función de la evaluación de las iteraciones precedentes. Ver Figura 1

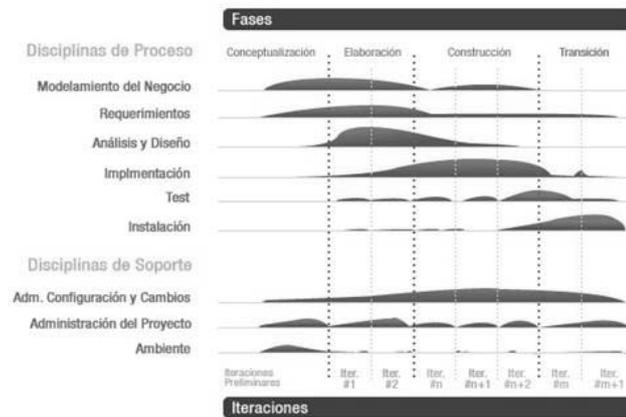


Figura 1. Fases e Iteraciones de la Metodología RUP.

Los elementos del RUP son:

1. Actividades, son los procesos que se llegan a determinar en cada iteración.
2. Trabajadores, vienen a ser las personas o entes involucrados en cada proceso.
3. Artefactos, un artefacto puede ser un documento, un modelo, o un elemento de modelo.

Características del Proceso Unificado

RUP es un proceso de desarrollo de software de forma tal que se asignan tareas y responsabilidades cuyos objetivos son asegurar la producción de software de calidad dentro de plazos y presupuestos predecibles. El ciclo de vida de RUP se caracteriza por ser: Dirigido por los casos de uso, Centrado en la arquitectura, Iterativo e incremental.

Dirigido por casos de uso: Los casos de uso reflejan lo que los usuarios futuros necesitan y desean, lo cual se capta cuando se modela el negocio y se representa a través de los requerimientos. Un caso de uso es un fragmento de funcionalidad del sistema que proporciona al usuario un resultado importante. Representan requerimientos funcionales. Los casos de uso guían la arquitectura del sistema y la arquitectura del sistema influye en la selección de los casos de uso.

Centrado en la arquitectura: La arquitectura muestra la visión común del sistema completo en la que el equipo de proyecto y los usuarios deben estar de acuerdo, por lo que describe los elementos del modelo



que son más importantes para su construcción, los cimientos del sistema que son necesarios como base para comprenderlo, desarrollarlo y producirlo económicamente. RUP se desarrolla mediante iteraciones, comenzando por los CU relevantes desde el punto de vista de la arquitectura. La arquitectura se refleja en los casos de uso pues cada producto tiene tanto una función como una forma, ninguna es suficiente por si sola.

Iterativo e Incremental: RUP propone dividir el trabajo en partes más pequeñas o miniproyectos, donde cada miniproyecto es una iteración que resulta en un incremento. Las iteraciones hacen referencia a pasos en los flujos de trabajo, y los incrementos, al crecimiento del producto. Cada iteración se realiza de forma planificada es por eso que se dice que son miniproyectos.

Una iteración es una secuencia de actividades con un plan establecido y criterios de evaluación las cuales están estrechamente relacionadas, cuyo resultado es una versión del software

Beneficios de la iteración:

1. Reduce el coste del riesgo al coste de un solo incremento.
2. Menos riesgo de no sacar el producto al mercado en fecha.
3. Acelera el ritmo de desarrollo.
4. Las necesidades del usuario y correspondientes requisitos no pueden definirse completamente al principio. Se requieren iteraciones sucesivas. (5)

1.8.2 Lenguaje de Modelado.

El Lenguaje Unificado de Modelado (**UML**) [23] prescribe un conjunto de notaciones y diagramas estándar para modelar sistemas orientados a objetos, y describe la semántica esencial de lo que estos diagramas y símbolos significan. UML se puede usar para modelar distintos tipos de sistemas: sistemas de software, sistemas de hardware, y organizaciones del mundo real. UML ofrece nueve diagramas en los cuales modelar sistemas:

1. Diagramas de Casos de Uso para modelar los procesos.
2. Diagramas de Secuencia para modelar el paso de mensajes entre objetos.
3. Diagramas de Colaboración para modelar interacciones entre objetos.



4. Diagramas de Estado para modelar el comportamiento de los objetos en el sistema.
5. Diagramas de Actividad para modelar el comportamiento de los Casos de Uso, objetos u operaciones.
6. Diagramas de Clases para modelar la estructura estática de las clases en el sistema.
7. Diagramas de Objetos para modelar la estructura estática de los objetos en el sistema.
8. Diagramas de Componentes para modelar componentes.
9. Diagramas de Implementación para modelar la distribución del sistema.

1.9 Herramientas CASE.

Las Herramientas **CASE** [3] son diversas aplicaciones informáticas destinadas a aumentar la productividad en el desarrollo de software reduciendo el coste de las mismas en términos de tiempo y de dinero. Estas herramientas pueden ayudar en todos los aspectos del ciclo de vida de desarrollo del software en tareas como el proceso de realizar un diseño del proyecto, calculo de costes, implementación de parte del código automáticamente con el diseño dado, compilación automática, documentación o detección de errores entre otros.

Visual Paradigm

El Visual Paradigm para UML es una herramienta CASE visual que ayuda a construir aplicaciones rápidamente, mejor y económicamente. Utiliza “UML”: como lenguaje de modelado, siguiendo el estándar UML 2.1. Esta herramienta tiene unas características gráficas muy cómodas que facilitan la realización de los diagramas de modelado que sigue el estándar de UML, que son: Diagramas de clases, Casos de Uso, Comunicación, Secuencia, Estado, Actividad, Componentes, etc.

La herramienta Visual-Paradigm incorpora una funcionalidad de análisis textual, facilitando la interacción directa con un enunciado escrito en lenguaje natural, es decir, permite relacionar elementos presentes en un enunciado con los diagramas UML correspondientes, aunque no contempla el tratamiento directo en el texto de la relación entre los componentes del enunciado.

Beneficios:

1. Navegación intuitiva entre el modelo visual y el código.



2. Poderosa herramienta de generación de **PDF** [18]/HTML a partir de diagramas UML.
3. Sincronización entre el código fuente y el modelo en tiempo real o bajo demanda.
4. Entorno visual de modelado superior.
5. Soporte para toda la notación UML.
6. Sofisticados y automáticos diagramas de capas.
7. Análisis de textos.
8. Diseño centrado en casos de uso y enfocado al negocio que generan un software de mayor calidad.
9. Uso de un lenguaje estándar común a todo el equipo de desarrollo que facilita la comunicación.
10. Capacidades de ingeniería directa (versión profesional) e inversa.
11. Modelo y código que permanece sincronizado en todo el ciclo de desarrollo.
12. Disponibilidad de integrarse en los principales **IDEs** [9].
13. Disponibilidad en múltiples plataformas. (6)

1.10 Herramientas de Apoyo.

Navegador

El navegador sobre el cual correrá la aplicación es el Mozilla Firefox este permite abrir por defecto las nuevas páginas web en pestañas, cada una de esas pestañas tiene su propio botón de cerrado.

Firefox incorpora bloqueo de ventanas emergentes, marcadores dinámicos, soporte para estándares abiertos, y un mecanismo para añadir funcionalidades mediante extensiones. Posee un corrector ortográfico para evitar que se cometan errores de ortografía en las entradas de información que se hagan. Tiene una sugerencia de búsqueda que se va desplazando a medida que se va introduciendo el texto que se desea buscar.

Mantiene a salvo a la aplicación de programas espías e impostores usando el poder de una comunidad de desarrollo que le da soporte. Estas y muchas otras posibilidades brindan este navegador del cual se hará uso para el sistema, aunque el sistema también correrá sobre el Internet Explorer.



Macromedia Dreamweaver 8.

Dreamweaver 8 es la herramienta de desarrollo Web líder del mercado y permite a sus usuarios diseñar, desarrollar y mantener de forma eficaz sitios y aplicaciones Web basadas en normas.

Con Dreamweaver 8, los desarrolladores Web lo abarcan todo, desde la creación y mantenimiento de sitios Web básicos hasta aplicaciones avanzadas compatibles con las mejores prácticas y las tecnologías más recientes.

Brinda múltiples herramientas visuales de diseño y un entorno de codificación adaptable a lenguajes de programación Web (PHP), trabaja con hojas de estilos CSS, permite la comparación de archivos para determinar que ha cambiado, permite el trabajo directo del lado del servidor, etc.

1.11 Conclusiones.

En este capítulo se detallaron las condiciones y problemas que rodean el objeto de estudio; y a través de los conceptos y definiciones planteadas, se determinaron las condiciones específicas que envuelven al mismo. En base a esto se obtuvieron los objetivos generales y específicos para este trabajo, se planteó una solución al problema, se analizaron las características de diferentes herramientas, para la creación de un software o aplicación así como algunas metodologías. Para el desarrollo de la aplicación Web se tuvo en cuenta las restricciones que impone el MINFAR como requisito en la elaboración de la misma.

Después de este análisis y la fundamentación realizada el lenguaje de programación que se utilizará es PHP y javascript para el control de las diferentes funciones a realizar en el cliente, como gestor de base de datos se estableció el PostgreSql.

Se empleará la metodología RUP por ser un proceso iterativo e incremental, que permite dirigir el proceso por casos de uso y tener un mejor rectoreo de la calidad en cada etapa del proceso, como lenguaje de modelado UML y la herramienta que se usará como entorno de creación de diagramas para UML el Visual Paradigm. Una vez conocidas las herramientas óptimas y los conceptos a utilizar se puede empezar a desarrollar la propuesta de sistema.



CAPÍTULO 2. CARACTERÍSTICAS DEL SISTEMA

2.1 Introducción.

En el campo del software también resulta útil la creación de modelos que organicen y presenten los detalles importantes de problemas reales que se vinculan con el sistema informático a construir.

Como se ha expresado anteriormente el objetivo del Proceso Unificado, es guiar a los desarrolladores de cualquier sistema software en la implementación y distribución eficiente de sistemas que se ajusten a las necesidades de los clientes.

En el presente capítulo, se dará una propuesta de solución de lo que funcionalmente debe realizar el sistema para erradicar lo problemas existentes anteriormente mencionados. Se transitará a lo largo de la fase de inicio por cada uno de los flujos de trabajo desde el negocio hasta el sistema. Todo esto se desarrollará por medio de un conjunto de artefactos y modelos resultantes de la aplicación de la metodología de desarrollo de software. Estos modelos deben cumplir una serie de propiedades, entre ellas la de ser coherentes y relacionados. Uno de los modelos útiles previo al desarrollo de un software es el modelo del negocio.

2.2 Propuesta del sistema.

Para desarrollar el proceso de gestión de los planes de los Órganos Consumidores del MINFAR, el sistema inicialmente debe permitir la elaboración de los mismos de acuerdo a las necesidades de los órganos. Una vez conformados que se verifica que están correctamente elaborados, se procede a la aprobación para posteriormente enviarlos a los Centros de Balances correspondientes y a la Dirección de Economía iniciando así el proceso de planificación.

Estos planes pueden ser actualizados dándose la posibilidad insertar, modificar y eliminar así como de realizar una búsqueda antes de efectuar las acciones anteriormente mencionadas. Además la aplicación permitirá obtener reportes de dichos planes por medio de un filtro de búsqueda, para realizar análisis pertinentes de estos, dando la posibilidad de imprimirlos.



2.3 Modelado del Negocio.

El modelado del negocio es una técnica para comprender y definir los procesos del negocio, roles y responsabilidades de la organización en los modelos de casos de uso del negocio y de objetos.

Los propósitos que se persiguen al realizarse el modelado del negocio, son:

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

Por tanto, la finalidad del modelado del negocio es describir cada proceso del negocio, especificando sus datos, actividades (o tareas), roles (o agentes) y reglas de negocio.

Este proceso está soportado por dos tipos de modelos de UML: modelos de casos de uso y modelo de objetos.

Un modelo de casos de uso del negocio describe los procesos de una empresa en términos de casos de uso y actores del negocio.

Un modelo de objetos del negocio es un modelo que describe cómo colaboran los trabajadores y las entidades del negocio dentro del flujo (realización).

Para lograr un mejor entendimiento del modelado del negocio, se presenta una descripción de cuales serán los actores y trabajadores del mismo.

2.4 Actor y Trabajadores del Negocio.

Un actor puede representar a una persona física, otro sistema, un dispositivo, siendo siempre un tercero fuera del sistema que colabora con él. La definición de actores, sirve para definir el contexto externo del sistema, esto es, delimitar los elementos que se encuentran fuera y dentro del mismo, y un actor juega un rol para cada caso de uso en el que colabora.



Tabla 1. Descripción de los actores del negocio.

Actor del negocio	Justificación
Dirección de Economía	Es quien especifica los planes que se van a conformar durante el proceso de planificación y crea las líneas generales para la planificación.

El trabajador, por otro lado, representa una posición que se puede asignar a una persona o equipo en una organización de desarrollo de software, y para cada trabajador se especifican las responsabilidades y habilidades requeridas, ya que este define el comportamiento y las responsabilidades de un individuo.

Tabla 2. Descripción de los trabajadores del negocio.

Trabajadores del negocio	Justificación
Especialista del Órgano Consumidor	Este trabajador será quien comienza la Planificación Material y Financiera en el MINFAR, conformará el o los distintos planes de demandas, para enviarlos luego a los niveles superiores, en este caso Centros de Balance y la Dirección de Economía.
Jefe del Órgano Consumidor	Es quien revisa y aprueba el plan de demanda que el especialista del OC ha elaborado.

2.5 Diagrama de Casos de Uso del Negocio.

Para lograr los propósitos del negocio y obtener una visión general de los diferentes procesos del mismo se construye el *diagrama de casos de uso del negocio*, donde cada proceso se representa como un caso de uso y solo aparecerán los actores del negocio correspondientes a los roles externos al sistema. Este diagrama permite mostrar los límites y el entorno de la organización (Ver Figura 2).



Figura 2. Diagrama de Casos de Uso del Negocio.

2.6 Descripciones Textuales.

La descripción textual de un caso de uso de negocio se formaliza en un documento generalmente llamado “Especificación del caso de uso de negocio”. Para realizar la misma, inicialmente se rellena una plantilla de descripción, y después, a partir de la información reflejada en dicha plantilla, se construye un conjunto de diagramas (*diagramas de actividades*) que describen completamente el caso de uso del negocio. En este caso se centrará en el único caso de uso del negocio, “Conformar Plan de Demandas”, cuya descripción se muestra a continuación. (Ver Tabla 3).

Tabla. 3 Descripción Textual Caso de Uso “Conformar Planes de Demandas”.

Caso de uso:	Conformar Plan de Demandas.
Actores del negocio	Dirección de Economía (inicia).
Trabajadores del negocio	Especialista del Órgano Consumidor (OC), Jefe del OC.
Resumen	El caso de uso comienza cuando la DE especifica los planes que se van a conformar en el año en curso, crea las líneas generales para dicha planificación y solicita la confección de los Planes de Demandas. El especialista del OC se estudia las Indicaciones Generales de la Planificación y a partir de ellas, realiza una



	selección de las actividades que les corresponden y confecciona sus objetivos en lo que se basa luego para conformar su plan de demandas. Una vez confeccionado el Plan el jefe del OC lo revisa y lo aprueba. El caso de uso termina cuando el OC le envía el plan al CB que lo balancea y también a la Dirección de Economía.
Precondiciones	Para que el caso de uso pueda realizarse tienen que existir las Indicaciones de la Planificación.
Acción del actor	Respuesta del proceso de negocio
<p>1. La DE crea las líneas generales para la planificación especificando:</p> <ul style="list-style-type: none"> • Planes a conformar para el próximo año. • Órganos que participan en la planificación. • Describe información que deben tributar los OC. 	
2. La DE solicita al Especialista del OC la confección del plan.	3. El Especialista del OC revisa las Indicaciones Generales para la Planificación.
	4. El Especialista del OC selecciona las indicaciones Generales propias.
	5. El Especialista del OC confecciona las indicaciones particulares
	6. El Especialista del OC conforma el Plan de Demandas.
	7. El Jefe del OC revisa el plan de demandas.
	<p>8. Si el Plan de Demandas está bien conformado lo aprueba y se hace entrega del mismo al CB correspondiente y a la Dirección de Economía.</p> <p>- Si el plan de Demandas presenta algún problema se procede a reconformar el nuevamente el Plan de Demandas.</p>
Pos-Condiciones	El Plan de Demandas General del OC queda conformado y en



	poder del CB correspondiente y de la Dirección de Economía.
Mejoras	La informatización de este caso de uso favorecerá a la rapidez y eficiencia en el proceso de planificación.
Prioridad	Crítico

2.7 Diagrama de Actividades.

Para mostrar de forma más detallada del flujo de trabajo que realiza cada proceso del negocio, se utilizará el diagrama de actividad este describe un proceso que explora el orden de las tareas o actividades que logran los objetivos del negocio. No es más que un diagrama de flujos (paralelos o secuenciales) que muestra el flujo de control de una actividad a otra.

Los datos aparecen como objetos que fluyen entre las actividades y pueden tener un estado (Ver Figura 3).

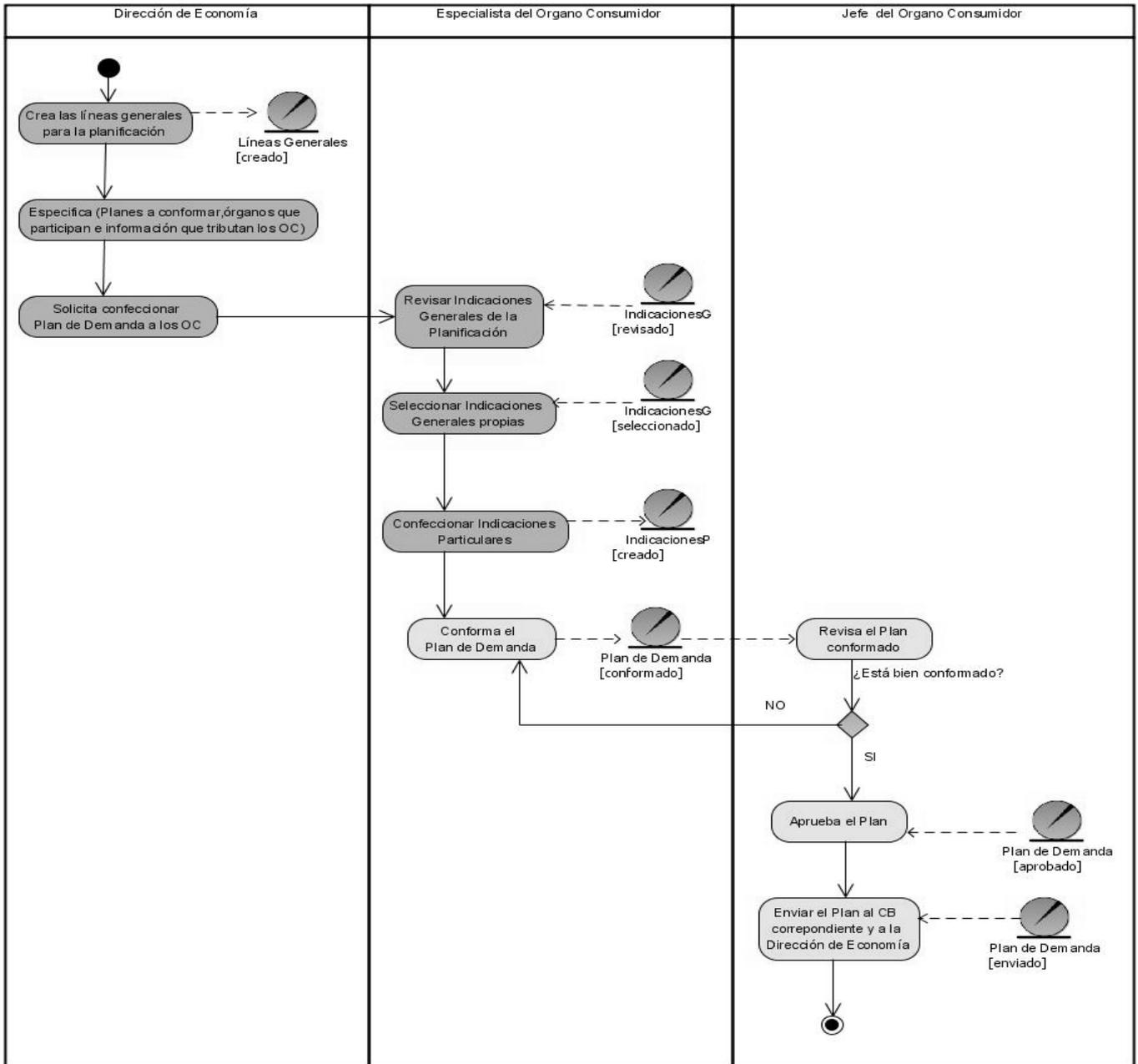


Figura 3. Diagrama de actividades "Conformar Plan de Demandas de los OC".



2.8 Diagramas de Clases del Modelo de Objetos.

Este diagrama muestra la relación existente entre los trabajadores y las entidades del negocio y proporciona un acercamiento a la identificación de los futuros actores y entidades del sistema (Ver Figura 4).

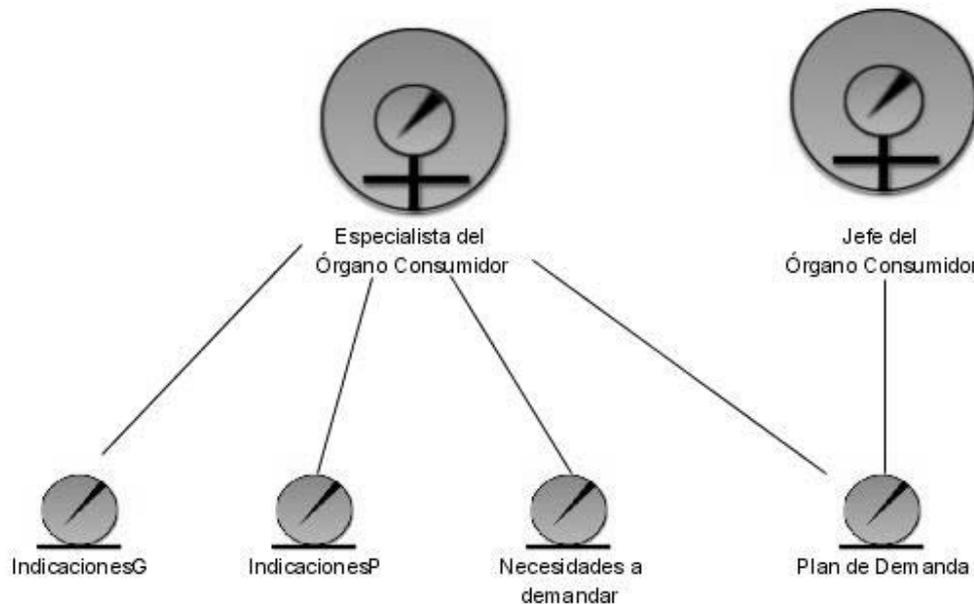


Figura 4. Modelo de Objeto.

2.9 Requerimientos.

Una de las principales tareas en el ciclo de desarrollo de un sistema es la de determinar los requerimientos del sistema de información (es decir, las condiciones o capacidades que el sistema debe cumplir). El propósito principal del flujo de trabajo de RUP captura de requisitos es guiar el desarrollo hacia el sistema correcto donde se describan con claridad y sin ambigüedades el comportamiento del mismo. Así como lograr una comunicación efectiva entre el cliente (incluyendo a los usuarios) y a los desarrolladores sobre qué debe y qué no debe hacer el sistema.

Los requisitos se pueden clasificar en: funcionales y no funcionales.



2.9.1 Requerimientos Funcionales.

Los requerimientos funcionales son capacidades o condiciones que el sistema debe cumplir. Su meta principal es identificar y documentar las acciones que en realidad debe ejecutar el sistema para que cumpla con los objetivos planteados al inicio de este trabajo.

Todas estas acciones se convierten en requisitos funcionales y de acuerdo a los objetivos propuestos el sistema debe ser capaz de:

R.1 Gestionar Plan de Demandas.

- 1.1. Actualizar Plan de Demandas del OC.
 - 1.1.1 Modificar un Plan seleccionado.
 - 1.1.2 Eliminar un producto del Plan seleccionado.
 - 1.1.3 Insertar un producto del Plan seleccionado.
- 1.2. Aprobar el Plan.

R.2. Gestionar Recuperaciones.

2.1 Mostar Reporte de Gastos.

- 2.1.1 Visualizar reporte por Gastos Corrientes.
- 2.1.2 Visualizar reporte por Gastos Capitales.
- 2.1.3 Imprimir el reporte visualizado.

2.2 Mostrar Reporte Plan Específico.

- 2.2.1 Mostar reporte por Plan Portadores Energéticos.
- 2.2.2 Mostar reporte por Plan Azúcar, Miel y Alcoholes.
- 2.2.3 Mostar reporte por Plan Producción Agropecuaria.
- 2.2.4 Mostar reporte por Plan Vestuario.
- 2.2.5 Mostar reporte por Plan Asistencia Médica.
- 2.2.6 Mostar reporte por Plan Aseguramiento Técnico.
- 2.2.7 Mostar reporte por Plan Mantenimiento Constructivo.
- 2.2.8 Mostar reporte por Plan Resto de Calidad de Vida.
- 2.2.9 Mostar reporte por Plan Operaciones Comerciales y No Comerciales.
- 2.2.10 Mostar reporte por Plan Otros Aseguramientos.



- 2.11 Mostar reporte por Plan Inversiones Constructivas.
- 2.12 Mostar reporte por Plan Equipos no Asociados a Inversiones Constructivas.
- 2.13 Mostar reporte por Plan Ciencia e Innovación Tecnológica.
- 2.14 Imprimir el reporte visualizado.

2.3 Mostrar Reporte Plan de Alimento.

- 2.3.1 Imprimir el reporte visualizado.

R.3 Buscar recursos en el Plan.

Por los siguientes criterios: *Centro de Balance, Código, Destino, Actividad, Denominación, Sector o Rama.*

R.4 Seleccionar Criterios de Selección.

2.9.2 Requerimientos No Funcionales.

Los requerimientos no funcionales son propiedades o cualidades que el producto debe tener. Debe pensarse en estas propiedades como las características que hacen al producto atractivo, usable, rápido o confiable. Los requerimientos no funcionales forman una parte significativa de la especificación. Son importantes para que clientes y usuarios puedan valorar las características no funcionales del producto y de esta forma poder decir si el producto tiene la calidad requerida.

Apariencia o interfaz externa.

La interfaz a implementar debe ser sencilla, de fácil uso y con rápida respuesta del sistema, de esta forma personas que no son expertas en el uso de la WEB no necesitarán tanto tiempo de adiestramiento. Debe tener diferentes opciones para mostrar y confeccionar documentos, así como para visualizar reportes y actualizaciones, contendrá un menú dinámico para que actúe con el usuario en dependencia de sus necesidades. Debe estar concebido para simular una aplicación de escritorio.

Los colores a usar deben ser refrescantes para la vista, colaborando con los requerimientos medio ambientales y siguiendo los patrones de colores representativos de la entidad como: gris, blanco y azul,



contendrá una ayuda para que el usuario que se encuentra trabajando en el sistema no se pierda y sepa lo que debe hacer.

Usabilidad

El sistema debe ser de fácil manejo para los usuarios que tengan niveles básicos sobre la computación o hayan realizado algún trabajo previo con Web.

Debe tener una opción de ayuda sobre las principales operaciones que se realizan y sus íconos respectivos para lograr un menor tiempo de aprendizaje.

Debe facilitar, principalmente, el ser manejado por usuarios que estén vinculados al proceso de Planificación Material y Financiera del MINFAR.

Rendimiento

Esta aplicación está concebida para un ambiente cliente/servidor así que los tiempos de respuestas deben ser generalmente rápidos (no debe exceder los 10 segundos) al igual que la velocidad de procesamiento de la información. El tiempo de búsqueda de información así como la generación de reportes y documentos debe ser en el menor tiempo posible pues se deben generar pantallas dinámicas, implicando esto que el acceso a la base de datos debe ser lo más disponible, rápido y consistente posible.

Soporte

Para el servidor de aplicaciones: Se requiere que esté instalado un intérprete de ficheros PHP 5.x y con las últimas actualizaciones del lenguaje.

Para el servidor de base de datos: Se requiere que esté instalado un gestor de base de datos que soporte grandes volúmenes de datos, maneje la concurrencia y transacciones.

Para el cliente: Se requiere que esté instalado un navegador que interprete Javascript y versiones HTML 3.0 o superior.

Portabilidad

Requiere de un sistema operativo el cual soporte el navegador Mozilla Firefox.



Requerimientos de 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 al menos 256MB de RAM.
- Se requiere al menos 1GB de disco duro.
- Procesador 1.2 GHz como mínimo.

Requerimientos de Software

El sistema se desarrollará con tecnología PHP versión 5.0. Correrá sobre un servidor con el sistema operativo UNIX (Linux). Se recurrirá a la tecnología Apache versión 2.0 o superior para el servidor Web. El sistema incluirá una base datos implementada en PostgreSQL versión 8.0 o superior.

En las computadoras de los clientes debe estar instalado el navegador Mozilla Firefox. Versión 1.5 o superior. La comunicación de las computadoras clientes con el servidor será a través de conexiones de fibra óptica, a una velocidad constante de 100 Mbps o superior.

Seguridad

El sistema debe comunicarse usando un protocolo http. 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.

Disponibilidad

El sistema deberá estar disponible las 24 horas del día para todos los usuarios con derechos a utilizarlo.



Confiabilidad

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

2.10 Modelación del Sistema.

El modelado del sistema representa la funcionalidad completa de un sistema mostrando su interacción con los agentes externos. Esta representación se hace a través de las relaciones entre los actores y los casos de uso dentro del sistema. Los diagramas de casos de uso definen conjuntos de funcionalidades afines que el sistema debe cumplir para satisfacer todos los requerimientos que tiene a su cargo. Esos conjuntos de funcionalidades son representados por los diferentes diagramas que darán solución a la aplicación.

2.11 Actores del Sistema.

Estos no son más que trabajadores que participan directamente en el negocio y/o nuevos actores que surgen para ejecutar funcionalidades que aparecen debido a la presencia de requisitos especiales (Ver Tabla 4).

Tabla 4. Descripción de los Actores del Sistema.

Actores del Sistema	Justificación
Especialista del Órgano Consumidor	Este actor será quien comienza la Planificación Material y Financiera en el MINFAR, conformará el o los distintos planes de demandas, para enviarlos



	luego a los niveles superiores, en este caso Centros de Balance y la Dirección de Economía.
Jefe del Órgano Consumidor	Es quien revisa y aprueba el o los planes de demandas que el especialista del OC ha elaborado.

2.12 Diagrama de Casos de Uso del Sistema.

Los diagramas de casos de uso sirven para especificar la funcionalidad y el comportamiento de un sistema mediante su interacción con los usuarios y/o otros sistemas. Es decir, un diagrama de casos de uso del sistema representa gráficamente a los procesos y su interacción con los actores.

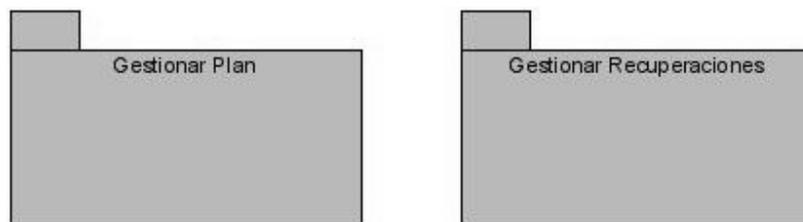


Figura 5. Estructuración del Sistema por Paquetes.

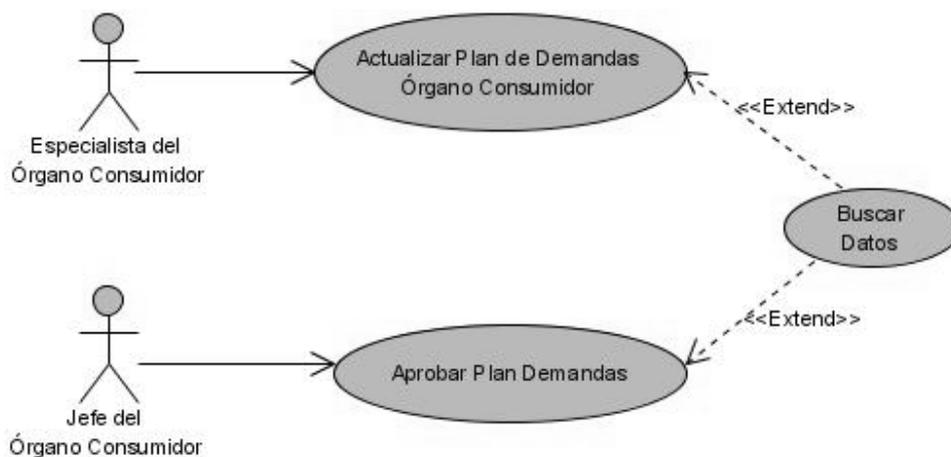


Figura 6. Diagrama de Casos de Uso del Sistema “Paquete Gestionar Plan”.

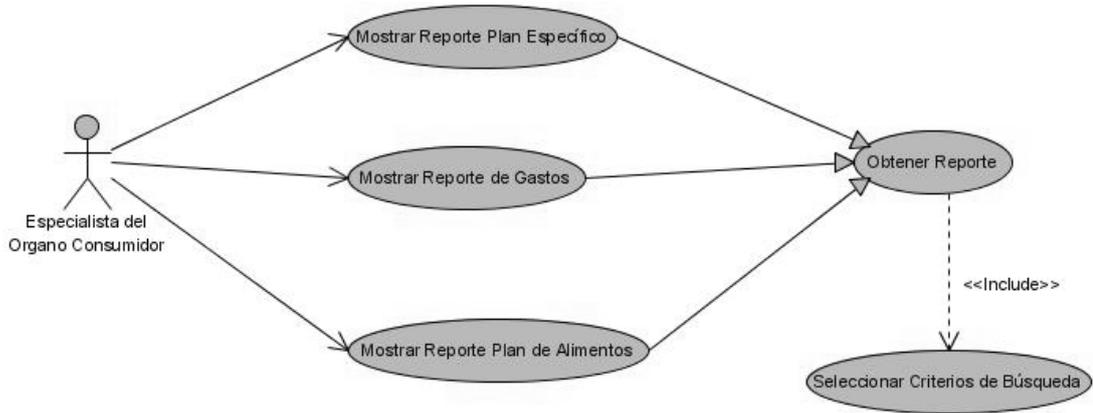


Figura 7. Diagrama de Casos de Uso del Sistema “Paquete Gestionar Recuperaciones”.

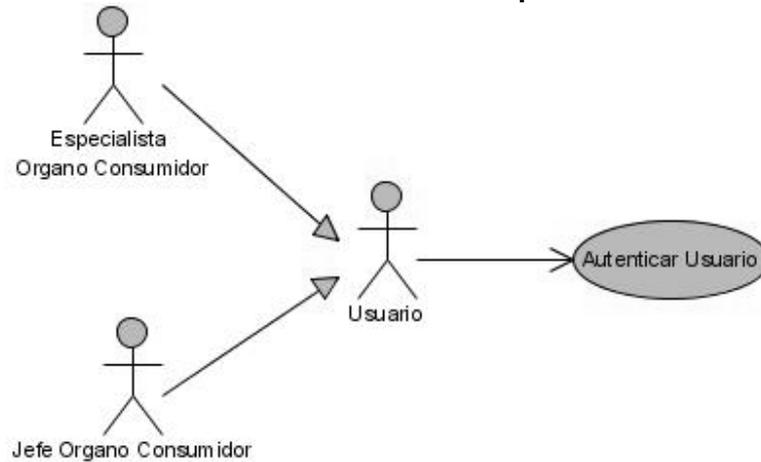


Figura 8. Diagrama de Casos de Uso del Sistema “Autenticar Usuario”.

2.13 Especificaciones de los Casos de Usos.

Tabla 5. Especificación del Caso de Uso “Actualizar Plan de Demandas”.

CU- 1	Actualizar Plan de Demandas Órgano Consumidor.
Actor	Especialista del Órgano Consumidor.(inicia)



Descripción	El caso de uso se inicia cuando el especialista del OC desea realizar actualizaciones en las demandas de productos y/o servicios que conforman el Plan de Demandas. El sistema muestra la interfaz de actualización del plan, donde el especialista puede agregar un nuevo producto o servicio, eliminar o modificar alguno de los datos ya existente. Finalmente se archivan los datos en el sistema y se devuelve el plan actualizado. Terminando así el caso de uso.
Referencia	R.1, 1.1
Prioridad	Crítico.

Tabla 6. Especificación del Caso de Uso “Aprobar Plan de Demandas”.

CU- 2	Aprobar Plan de Demandas Órgano Consumidor
Actor	Jefe del Órgano Consumidor.(inicia)
Descripción	El caso de uso se inicia cuando el Jefe del OC desea realizar un análisis del Plan de Demandas para aprobarlo. El sistema muestra el Plan de Demandas de forma que él pueda realizar el análisis pertinente y posteriormente aprobarlo. Terminando así el caso de uso.
Referencia	R.1, 1.2
Prioridad	Crítico.

Tabla 7. Especificación del Caso de Uso “Mostrar Reporte de Gastos”.

CU- 3	Mostrar Reporte de Gastos.
Actor	Especialista del Órgano Consumidor.(inicia)
Descripción	El caso de uso se inicia cuando el especialista del OC desea recuperar datos de Gastos Corrientes o de Gastos Capitales para su análisis o su posterior impresión. El sistema muestra una interfaz que le permite obtener el reporte haciendo una selección de diferentes criterios. Una



	vez efectuada la búsqueda muestra el reporte. El caso de uso termina cuando se visualiza el reporte.
Referencia	R.2, 2.1
Prioridad	Secundario

Tabla 8. Especificación del Caso de Uso “Mostrar Reporte Plan Específico”.

CU- 4	Mostrar Reporte Plan Específico.
Actor	Especialista del Órgano Consumidor.(inicia)
Descripción	El caso de uso se inicia cuando el especialista del OC desea recuperar datos de un Plan Específico para su análisis o su posterior impresión. El sistema muestra una interfaz que le permite obtener el reporte haciendo una selección de diferentes criterios. Una vez efectuada la búsqueda muestra el reporte. El caso de uso termina cuando se visualiza el reporte.
Referencia	R.2, 2.2
Prioridad	Secundario

Tabla 9. Especificación del Caso de Uso “Mostrar Reporte Plan de Alimentos”.

CU- 5	Mostrar Reporte Plan de Alimentos.
Actor	Especialista del Órgano Consumidor. (inicia)
Descripción	El caso de uso se inicia cuando el especialista del OC desea obtener un resumen del Plan de Alimentos para su análisis o Impresión del documento generado. El sistema muestra una interfaz que le permite obtener el plan haciendo una selección de diferentes criterios. Una vez



	efectuada la búsqueda muestra el Plan. El caso de uso termina cuando se visualiza el reporte.
Referencia	R.2, 2.3
Prioridad	Secundario

Tabla 10. Especificación del Caso de Uso “Buscar Datos”.

CU- 6	Buscar Datos.
Actor	Especialista del Órgano Consumidor, Jefe del OC
Descripción	<p>El caso de uso se inicia cuando se ejecuta el CU Actualizar Plan o Aprobar Plan.</p> <p>-El especialista del OC desea buscar datos del Plan de Demandas para realizar un análisis, modificar o eliminar. El sistema muestra los datos del Plan de Demandas los cuales fueron seleccionados por diferentes criterios. Terminando así el caso de uso.</p> <p>-El Jefe del OC desea realizar una búsqueda en el Plan de Demandas a aprobar. El sistema muestra los datos del Plan de Demandas los cuales fueron seleccionados por diferentes criterios. Terminando así el caso de uso.</p>
Referencia	R.3
Prioridad	Secundario

Tabla 11. Especificación del Caso de Uso “Seleccionar Criterios de Búsqueda”.

CU- 7	Seleccionar Criterios de Búsqueda. (Include)
Actor	Especialista del Órgano Consumidor.
Descripción	El caso de uso se inicia cuando el especialista del OC desea



	seleccionar algunos criterios para hacer más específica una recuperación determinada.
Referencia	R.4
Prioridad	Secundario

2.14 Casos de Uso expandidos.

La descripción puede ser elaborada de forma breve o extendida y debe ir acompañada del prototipo respectivo. El prototipo del sistema que se construye en este punto da una visión de las pantallas diseñadas para cada caso de uso, pero con comportamiento estático, que se presenta al usuario para verificar los requerimientos funcionales.

Ayuda a comprender los Casos de Uso del Sistema, además se hace referencia a los requerimientos consignados en el documento de Requerimientos, con los cuales tiene relación. Este sería el último paso en el análisis, para pasar a la construcción de la solución propuesta.

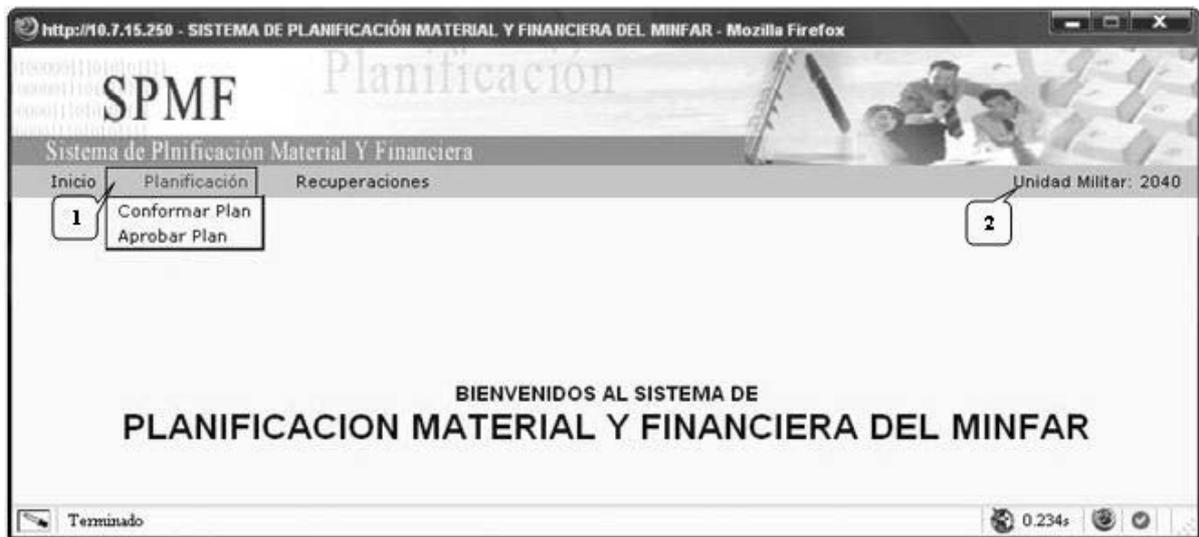
Tabla 12. Descripción Textual del caso de uso “Actualizar Plan de Demandas de los OC”.

Caso de Uso	Actualizar Plan Demandas Órgano Consumidor
Actores	Especialista del Órgano Consumidor (inicia).
Propósito	El propósito de este caso de uso es que el especialista del OC pueda realizar actualizaciones en el Plan de Demandas de productos y/o servicios del Órgano Consumidor.
Resumen	El caso de uso se inicia cuando el especialista del OC desea realizar actualizaciones en las demandas de productos y/o servicios que conforman el Plan de Demandas. El sistema muestra la interfaz de actualización del plan, donde el especialista puede agregar un nuevo producto o servicio, eliminar o modificar alguno de los datos ya existente. Finalmente se archivan los datos en el sistema y se devuelve el plan actualizado. Finalizando así el Caso de Uso.
Tipo	Real y expandido.



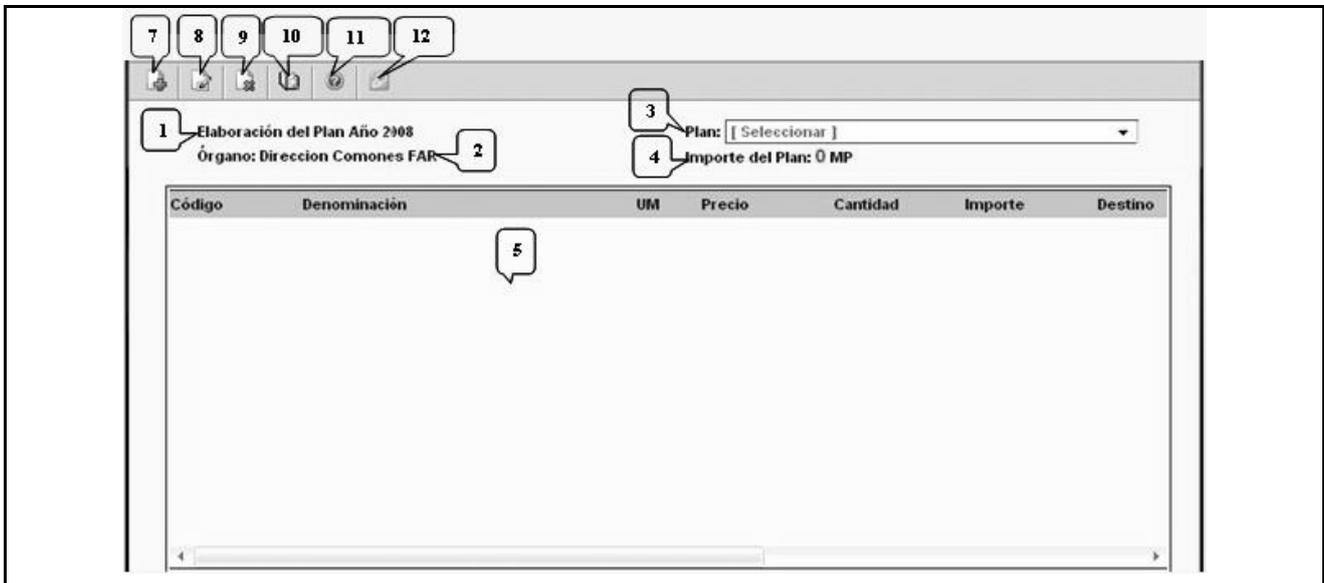
Precondiciones.	El actor se debe haber autenticado, como tal encontrarse trabajando con el sistema y tener acceso a actualizar el Plan de Demandas. Al registrarse el actor, el sistema lo identifica como OC (rol).
Referencias	R.1, R1.1, R1.1.1, R1.1.2, R1.1.3
Casos de Uso relacionados:	Buscar Datos (extendido).

Interfaz I



- (1) Submenú para seleccionar la acción que desea realizar.
- (2) Número público de la Unidad Militar. Su valor es de una variable de sesión.

Interfaz II



- (1) Label que contiene el año para el que se está planificando. Su valor es de una variable de sesión.
- (2) Label que contiene el OC que entró al sistema. Su valor es de una variable de sesión .
- (3) ListBox para seleccionar el Plan de Demanda. **Nombre:** Plan, **tipo:** string
- (4) Campo calculado. **Nombre:** Importe, **tipo:** float
- (5) Grid para mostrar los datos (si existen) del Plan de Demandas, inicialmente vacío, hasta que se seleccione el Plan. **Nombre:** GridDatos, **tipo:** objeto
- (6) Botón para Agregar productos o servicios al Plan. **Nombre:** Agregar, **tipo:** button
- (7) Botón para Modificar el Plan. **Nombre:** Modificar, **tipo:** button
- (8) Botón para Eliminar productos o servicios del Plan. **Nombre:** Eliminar, **tipo:** button
- (9) Botón para Buscar Plan. **Nombre:** Buscar, **tipo:** button
- (10) Botón para la Ayuda del sistema. **Nombre:** Ayuda, **tipo:** button
- (11) Botón para cerrar la ventana. **Nombre:** Cerrar, **tipo:** button

Flujo normal de eventos

Acción del actor	Respuesta del sistema
------------------	-----------------------



<p>1. El especialista del OC desea actualizar el Plan de Demandas y selecciona la opción <Conformar Plan> del menú principal Interfaz I</p>	<p>2. El sistema visualiza la Interfaz II.</p>
<p>3. El Especialista selecciona el Plan que desea actualizar. Interfaz II (4).</p>	<p>4. El sistema visualiza en el grid Interfaz II (6) los datos del Plan de Demandas (si existen), sino lo muestra vacío, hasta que adicionen productos o servicios en el Plan.</p>
<p>5. El especialista decide:</p> <ul style="list-style-type: none">• Agregar productos (6) (ver sección Insertar).• Modificar productos (7) (ver sección Modificar).• Eliminar productos (9) (ver sección Eliminar).• Buscar productos (10) (ver sección Buscar).	

Sección Insertar

Interfaz III

The screenshot shows a software interface titled "Está planificando: Plan de Alimentos." with a close button in the top right corner. The interface is divided into two columns of input fields. The left column contains: "Centro Balance" with a dropdown menu labeled "1"; "Código" with a dropdown menu labeled "2"; "Unidad" with a text input field labeled "3"; "Medida" with a text input field labeled "4"; "Cantidad" with a text input field labeled "5"; "Destino" with a dropdown menu labeled "6"; and "Actividad" with a dropdown menu labeled "6". The right column contains: "Denominación" with a dropdown menu labeled "7"; "Precio" with a text input field labeled "8"; "Importe" with a text input field labeled "9"; "Sector - Rama" with a dropdown menu labeled "10"; and "Satisface" with a dropdown menu labeled "11". At the bottom right, there are two buttons: "Aceptar" labeled "12" and "Cancelar" labeled "13". A small "X" icon in a box is labeled "14" in the top right corner.



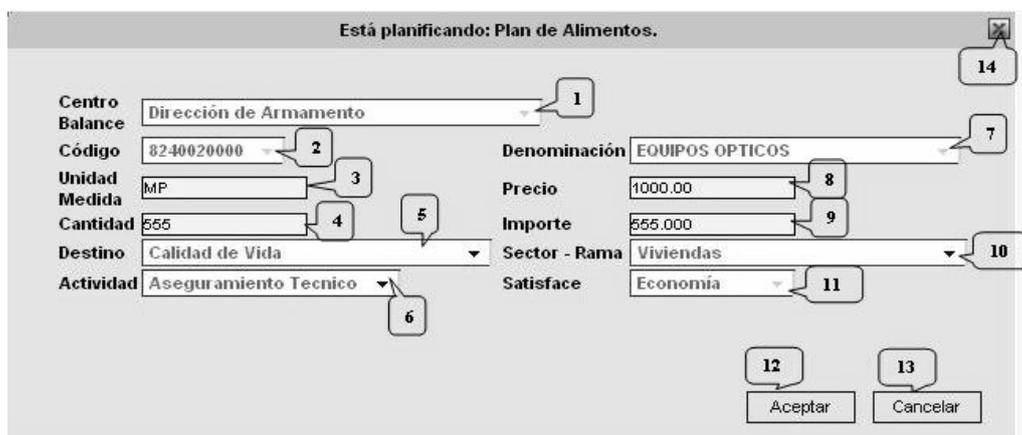
<p>(1) ListBox seleccionar el Centro de Balance. Nombre: CentBalan, tipo: string</p> <p>(2) ListBox seleccionar el Código del Producto o Servicio. Nombre: CodProd, tipo: entero</p> <p>(3) Edit Unidad de Medida. Nombre: UniMed, tipo: string.</p> <p>(4) Edit Cantidad del Producto a Demandar. Nombre: CantProd, tipo: entero</p> <p>(5) ListBox seleccionar el Destino. Nombre: Dest, tipo: string.</p> <p>(6) ListBox seleccionar opcionalmente la Actividad. Nombre: Activ, tipo: string.</p> <p>(7) ListBox Denominación. Nombre: Denom, tipo: string</p> <p>(8) Edit Precio. Nombre: Precio, tipo: double.</p> <p>(9) Edit Importe. Nombre: Importe, tipo: double</p> <p>(10) ListBox seleccionar el Sector_ Rama. Nombre: Sector, tipo: string</p> <p>(11) ListBox seleccionar OACE que Satisface el recurso. Nombre: OACE, tipo: string</p> <p>(12) Botón Aceptar. Nombre: Aceptar, tipo: button</p> <p>(13) Botón Cancelar la operación. Nombre: Cancelar, tipo: button.</p> <p>(14) Button Cerrar ventana. Nombre: Cerrar, tipo: button.</p>	
Acción del actor	Respuesta del sistema
<p>1. El especialista desea introducir un nuevo producto o servicio en el Plan y selecciona la opción <Insertar>. Interfaz II (6)</p>	<p>2. El sistema le muestra el formulario (Interfaz III) en blanco para dar la posibilidad de agregar un nuevo producto y muestra las opciones Aceptar y Cancelar activas.</p>
<p>3. El especialista introduce los datos del nuevo producto o servicio por teclado. En la opción:</p> <p>(1) Selecciona el Centro de Balance.</p> <p>(2) Selecciona el Código del Producto o Servicio.</p> <p>(4) Cantidad del Producto a</p>	<p>4. El sistema auto completa :</p> <p>(3) Unidad de Medida.</p> <p>(7) Denominación.</p> <p>(8) Precio.</p> <p>(9) Importe.</p>



<p>Demandar.</p> <p>(5) Selecciona el Destino.</p> <p>(6) Selecciona opcionalmente la actividad.</p> <p>(10) Selecciona el Sector_ Rama.</p> <p>(11) Selecciona OACE que Satisface el recurso.</p>	
<p>5. El especialista selecciona <Aceptar> Interfaz III (12).</p>	<p>6. El sistema verifica que los datos introducidos sean correctos. Guarda los cambios en sistema. Devuelve el Plan de Demandas actualizado visualizándolo en el Grid Interfaz II (5) y envía mensaje al actor.</p> 

Sección Modificar

Interfaz IV



Está planificando: Plan de Alimentos.

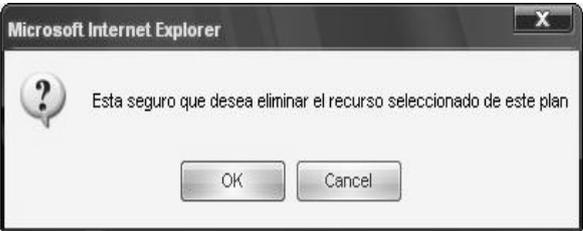
Centro	Dirección de Armamento	1	Denominación	EQUIPOS OPTICOS	7
Balance			Precio	1000.00	8
Código	8240020000	2	Importe	555.000	9
Unidad	MP	3	Sector - Rama	Viviendas	10
Medida			Satisface	Economía	11
Cantidad	555	4			
Destino	Calidad de Vida	5			
Actividad	Aseguramiento Tecnico	6			

12 Aceptar 13 Cancelar



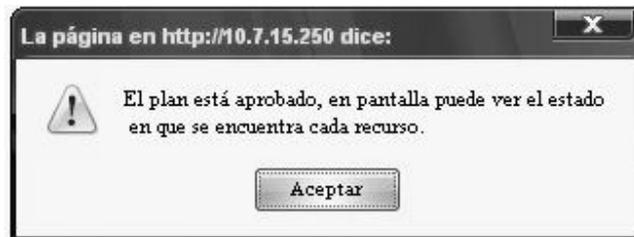
<p>(1) ListBox seleccionar el Centro de Balance. Nombre: CentBalan, tipo: string</p> <p>(2) ListBox seleccionar el Código del Producto o Servicio. Nombre: CodProd, tipo: entero</p> <p>(3) Edit Unidad de Medida. Nombre: UniMed, tipo: string.</p> <p>(4) Edit Cantidad del Producto a Demandar. Nombre: CantProd, tipo: entero</p> <p>(5) ListBox seleccionar el Destino. Nombre: Dest, tipo: string.</p> <p>(6) ListBox seleccionar opcionalmente la Actividad. Nombre: Activ, tipo: string.</p> <p>(7) ListBox Denominación. Nombre: Denom, tipo: string</p> <p>(8) Edit Precio. Nombre: Precio, tipo: double.</p> <p>(9) Edit Importe. Nombre: Importe, tipo: double</p> <p>(10) ListBox seleccionar el Sector_ Rama. Nombre: Sector, tipo: string</p> <p>(11) ListBox seleccionar OACE que Satisface el recurso. Nombre: OACE, tipo: string</p> <p>(12) Botón Aceptar. Nombre: Aceptar, tipo: button</p> <p>(13) Botón Cancelar la operación. Nombre: Cancelar, tipo: button.</p> <p>(14) Botón Cerrar ventana. Nombre: Cerrar, tipo: button.</p>	
Acción del actor	Respuesta del sistema
<p>1. El especialista selecciona el producto o servicio que desea modificar del Plan y selecciona la opción <Modificar> en la Interfaz II (7).</p>	<p>2. El sistema muestra la Interfaz IV con los datos del producto o servicio que no cambian inhabilitados (1), (2), (7), (11). Auto completa los campos de UM (3), Precio (8) e Importe (9) y muestra las opciones Aceptar y Cancelar activas.</p>
<p>3. El especialista realiza los cambios necesarios por teclado, solo puede cambiar las opciones (5), (6) y (10) y selecciona <Aceptar> (12).</p>	<ul style="list-style-type: none"> El sistema verifica que los datos modificados sean correctos. Guarda los cambios en el sistema. Devuelve el Plan de Demandas actualizado y envía mensaje al actor.



	
Sección Eliminar	
Acción del actor	Respuesta del sistema
1. El especialista selecciona el producto o servicio que desea eliminar del Plan Interfaz II (5) y selecciona la opción <Eliminar> en la Interfaz II (8).	2. El sistema pide confirmación al actor 
3. El especialista responde "OK"	4. El sistema elimina el producto o servicio señalado por el usuario. Guarda los datos en el sistema. Devuelve el Plan de Demandas actualizado.
Sección Buscar	
Acción del actor	Respuesta del sistema
El especialista desea buscar un producto o servicio y selecciona la opción <Buscar> Interfaz II (10).	El sistema ejecuta CU Buscar Datos.
Cursos Alternos	
Línea 3. Si el Plan seleccionado no contiene datos el sistema envía mensaje al actor: 	



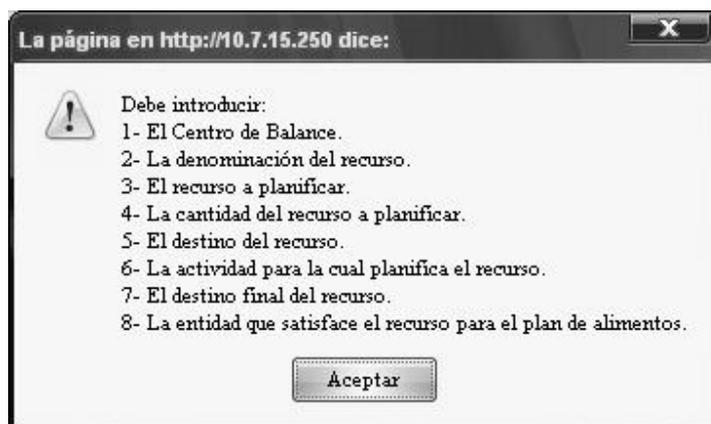
Línea 3. Si el Plan está aprobado el sistema envía mensaje al actor:



Línea 4. Si el Plan está aprobado el sistema visualiza la **Interfaz II** con los datos del Plan de Demandas, y la opciones de Insertar (7), Modificar (8) y Eliminar (9) deshabilitadas.

Sección Insertar.

Línea 5. En caso que el especialista no seleccione criterios de búsqueda y oprima <Aceptar> **Interfaz III** (12), el sistema envía mensaje al actor.



Línea 5. Si el usuario introduce un producto o un servicio que ya existe, el sistema envía mensaje al actor.

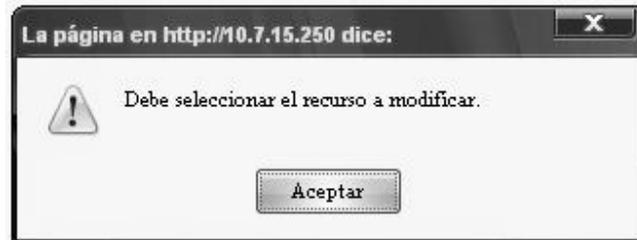


Línea 5. Si el actor decide <Cancelar> el sistema no registra los cambios y el sistema regresa a la interfaz inicial.

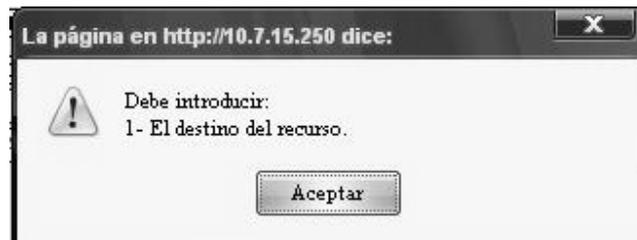


Sección Modificar.

Línea 1. Si el especialista no ha seleccionado el recurso a modificar y selecciona <Modificar> el sistema le envía un mensaje.



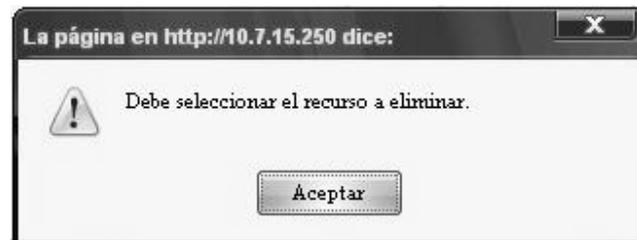
Línea 3. En caso que el especialista no seleccione criterios de búsqueda y oprima <Aceptar> **Interfaz IV** (12), el sistema envía mensaje al actor.



Línea 3. Si el actor decide <Cancelar> el sistema regresa a la interfaz inicial.

Sección Eliminar.

Línea 1. Si el especialista no ha seleccionado el recurso a eliminar y selecciona <Eliminar> el sistema le envía un mensaje.



Línea 4. Si el actor decide <Cancel> el sistema regresa a la interfaz inicial.

Pos-condiciones	El Plan de Demandas queda Actualizado.
Prioridad	Crítico



Tabla 13. Descripción Textual del caso de uso “Buscar Datos en el Plan de Demandas OC”.

Caso de Uso	Buscar Datos en el Plan de Demandas Órgano Consumidor
Actores	Especialista del Órgano Consumidor, Jefe del Órgano Consumidor
Propósito	El propósito de este caso de uso es que el especialista del OC o el Jefe del Órgano Consumidor puedan buscar datos del Plan de Demandas.
Resumen	
El caso de uso se inicia cuando el especialista del OC desea buscar datos del Plan de Demandas para realizar un análisis, modificar o eliminar. El sistema muestra los datos del Plan de Demandas los cuales fueron seleccionados por diferentes criterios. Finalizando así el caso de uso.	
Tipo	Real y expandido.
Referencias	R 3
Interfaz V	
<p>(1) ListBox seleccionar Centro de Balance. Nombre: CentBaln, tipo: string</p> <p>(2) ListBox seleccionar Código del producto o servicio. Nombre: CodProd, tipo: double</p> <p>(3) ListBox seleccionar Destino. Nombre: Dest, tipo: string</p>	



(4) ListBox seleccionar Actividad. Nombre: Activ, tipo: texto	
(5) ListBox seleccionar Denominación. Nombre: Fund, tipo: string	
(6) ListBox seleccionar Sector o Rama. Nombre: Fund, tipo: string	
Flujo normal de eventos	
Acción del actor	Respuesta del sistema
1. El actor desea buscar un producto en el Plan y selecciona la opción <Buscar>. Interfaz II (9)	2. El sistema muestra la Interfaz V.
3. El actor selecciona los criterios por los que desea buscar, que pueden ser: (1) Centro de Balance. (2) Código del producto o servicio. (3) Destino. (4) Actividad. (5) Denominación. (6) Sector o Rama.	
4. El actor selecciona <Aceptar> (7)	5. El sistema muestra en el Grid Interfaz II (5) ó Interfaz VII (9) el resultado de la búsqueda de los datos del Plan de Demandas.
Cursos Alternos	
Línea 5. Si el sistema no encuentra datos en el Plan de Demandas muestra en el Grid Interfaz II (5) un mensaje:	



<table border="1"> <thead> <tr> <th>Código</th> <th>Denominación</th> <th>UM</th> <th>Precio</th> <th>Cantidad</th> <th>Importe</th> </tr> </thead> <tbody> <tr> <td>0</td> <td colspan="5">No hay resultados que mostrar</td> </tr> </tbody> </table>		Código	Denominación	UM	Precio	Cantidad	Importe	0	No hay resultados que mostrar				
Código	Denominación	UM	Precio	Cantidad	Importe								
0	No hay resultados que mostrar												
Pos-condiciones	El usuario obtiene una respuesta del sistema.												
Prioridad	Secundario												

Tabla 14. Descripción Textual del caso de uso “Seleccionar Criterios de Búsqueda.”.

Nombre del caso de uso	Seleccionar Criterios de Búsqueda. (Include)
Actores	Especialista del Órgano Consumidor (OC).
Propósito	El propósito de este caso de uso es que el especialista del OC pueda seleccionar los diferentes criterios por los cuales desea generar un reporte determinado.
Resumen	
El caso de uso se inicia cuando el especialista del OC desea seleccionar algunos criterios para hacer más específica una recuperación. El caso de uso termina cuando el especialista oprime el botón aceptar.	
Tipo	Real y expandido.
Referencias	R.5
Caso de Uso relacionado:	
Interfaz VI	



FILTRO REPORTES

C.Balance: --Seleccione el Centro de Balance-- OACE: --Seleccione el OACE--

O.Consumidor: --Seleccione Organo Consumidor-- Satisface: --Seleccione quien satisface--

Tipo Gasto: [Seleccionar] Concepto: [Seleccionar]

Destino: -- Sector - Rama: [Seleccionar]

Codigo: [Seleccionar] Denominacion: [Seleccionar recurso]

Actividad: --Seleccione-- Lineas por Hoja: 30

- (1) ListBox seleccionar Centro de Balance. **Nombre:** CentBalanc, **tipo:** string
- (2) ListBox seleccionar Órgano Consumidor. **Nombre:** OrgConsum, **tipo:** string.
- (3) ListBox seleccionar Tipo de Gasto. **Nombre:** TGasto, **tipo:** string.
- (4) ListBox seleccionar Destino. **Nombre:** Dest, **tipo:** string.
- (5) ListBox seleccionar Código. **Nombre:** Cod, **tipo:** entero
- (6) ListBox seleccionar Actividad. **Nombre:** Actv, **tipo:** string.
- (7) ListBox seleccionar el OACE. **Nombre:** OACE, **tipo:** string.
- (8) Listbox seleccionar a quien Satisface. **Nombre:** Satisf, **tipo:** string
- (9) ListBox seleccionar Concepto de Gasto. **Nombre:** ConcepGasto., **tipo:** string
- (10) ListBox seleccionar Sector o Rama. **Nombre:** Sector, **tipo:** string.
- (11) ListBox seleccionar la Denominación. **Nombre:** Denom, **tipo:** string.
- (12) Edit para entrar la cantidad Líneas por Hojas. **Nombre:** CantLineas, **tipo:** entero.

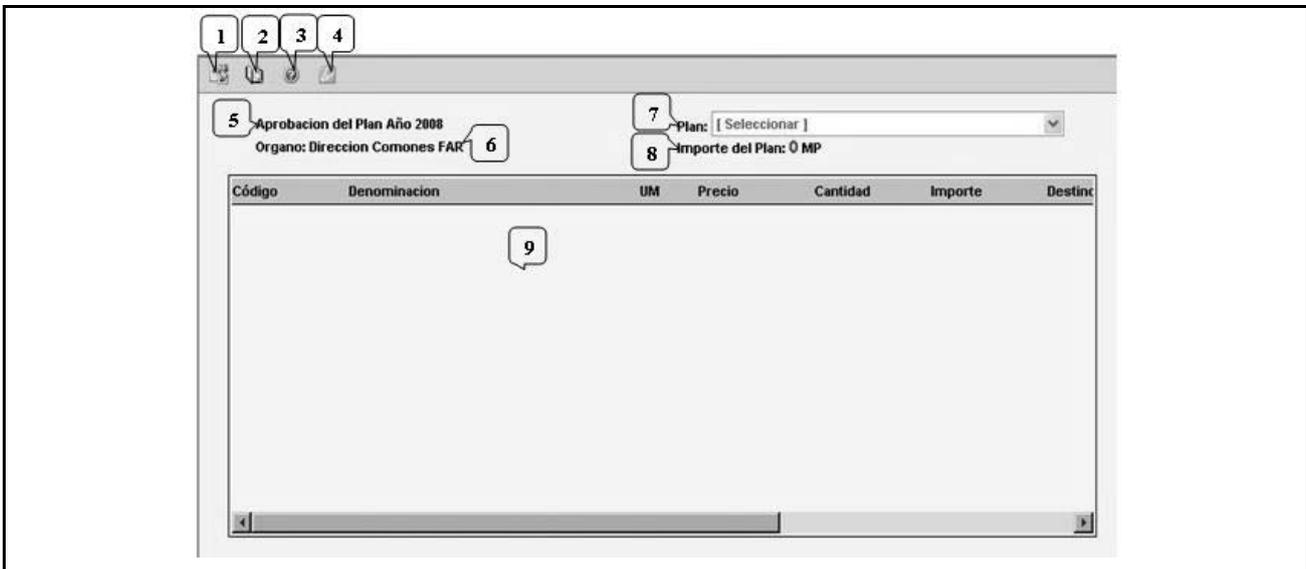
Flujo normal de eventos	
Acción del actor	Respuesta del sistema
	1. El sistema muestra la Interfaz VI con los diferentes criterios por los cuales puede generarse un reporte.
2. El especialista selecciona en los diferentes listbox los criterios de búsqueda por los que va a filtrar el reporte.	



3. El especialista selecciona <Aceptar> (10).	
Cursos alternos	
<p>Línea 2 En caso que el especialista desee generar el reporte sin seleccionar criterios de búsqueda, oprime el botón <Aceptar> (10) de la Interfaz VI, finalizando así el CU.</p>	
Pos-condiciones	El sistema le muestra el Plan seleccionado.
Prioridad	Secundario

Tabla 15. Descripción Textual del caso de uso “Aprobar Plan de Demandas OC.”

Caso de Uso	Aprobar Plan de Demandas Órgano Consumidor
Actores	Jefe del Órgano Consumidor (inicia)
Propósito	El propósito de este caso de uso es que el jefe del OC pueda aprobar el Plan de Demandas.
Resumen	El caso de uso se inicia cuando el jefe del OC desea realizar un análisis del Plan de Demandas para aprobarlo. El sistema muestra el Plan de Demandas de forma que él pueda realizar el análisis pertinente y posteriormente aprobarlo. El caso de uso termina cuando el Plan es aprobado.
Tipo	Real y expandido.
Precondiciones.	El plan de Demandas debe estar conformado.
Referencias cruzadas.	R.1, R.1.2
Interfaz VII	



- (1) Botón Aprobar Plan de Demandas. **Nombre:** Aprobar **tipo:** button.
- 2) Botón buscar un determinado producto por diferentes criterios de selección. **Nombre:** Buscar **tipo:** button.
- (3) Botón ayuda del sistema. **Nombre:** Ayuda **tipo:** button.
- (4) Botón cerrar sección. **Nombre:** Cerrar **tipo:** button
- (5) Label que contiene la aprobación del año para el cual se está realizando la misma. Su valor es una variable de sesión.
- (6) Label que contiene el órgano para el cual se va realizar la probación del Plan. Su valor es una variable de sesión.
- (7) ListBox seleccionar el Plan de Demandas que desea aprobar. **Nombre:** Plan **tipo:** string.
- (9) Grid mostrar el Plan Conformado. **Nombre:** GridPlan **tipo:** objeto.

Flujo normal de eventos

Acción del actor	Respuesta del sistema
1. El jefe del OC desea revisar el Plan para aprobarlo y selecciona la opción Aprobar	2. El sistema muestra la interfaz VII.



Plan del menú principal Interfaz I.	
3. El Jefe del OC selecciona el Plan de Demandas que desea aprobar Interfaz VII (7).	4. El sistema le muestra en el Grid Interfaz VII (9) los datos del Plan seleccionado.
5. El Jefe selecciona <Aprobar Plan> Interfaz VI (1).	6. El sistema muestra un mensaje de confirmación al actor: 
7. El Jefe selecciona <Aceptar>.	8. El sistema guarda los datos en el sistema, el plan queda aprobado.
Cursos Alternos	
Línea 5. Si el Jefe del OC desea realizar primero una búsqueda antes de aprobar el Plan y selecciona <Buscar> Interfaz VII (2) (Ejecutar CU Buscar Datos).	
Línea 7. Si el actor decide <Cancelar> el sistema regresa a la interfaz inicial.	
Pos-condiciones	El Plan no puede ser modificado por el especialista del OC.
Prioridad	Crítico

Tabla 16. Descripción Textual del caso de uso “Obtener Reporte.”

Caso de Uso	Obtener Reporte (Caso de uso base generalizado)
Actores	
Propósito	El propósito de este caso de uso es que el especialista del OC pueda obtener reporte.
Resumen	El caso de uso se inicia cuando el Caso Uso “Mostrar Reporte de Gasto” o “Mostrar Reporte Plan Específico” o “Mostrar Reporte Plan de Alimento”, invoca al comportamiento del caso de usos en



	cuestión, que generaliza acciones comunes de ambos casos.
Tipo	Real y expandido
Referencias	R.2

Flujo normal de eventos

Sección “ Seleccionar Reporte”

Interfaz VIII



- (1) Submenú para seleccionar la acción que desea realizar.
- (2) Número público de la Unidad Militar. Su valor es de una variable de sesión..

Acción del actor	Respuesta del sistema
1. El especialista del OC desea obtener un reporte y selecciona la opción <Recuperaciones> del menú principal Interfaz VIII (1) .	2. El sistema ejecuta el Caso de Uso: “Seleccionar Criterios de Búsqueda”

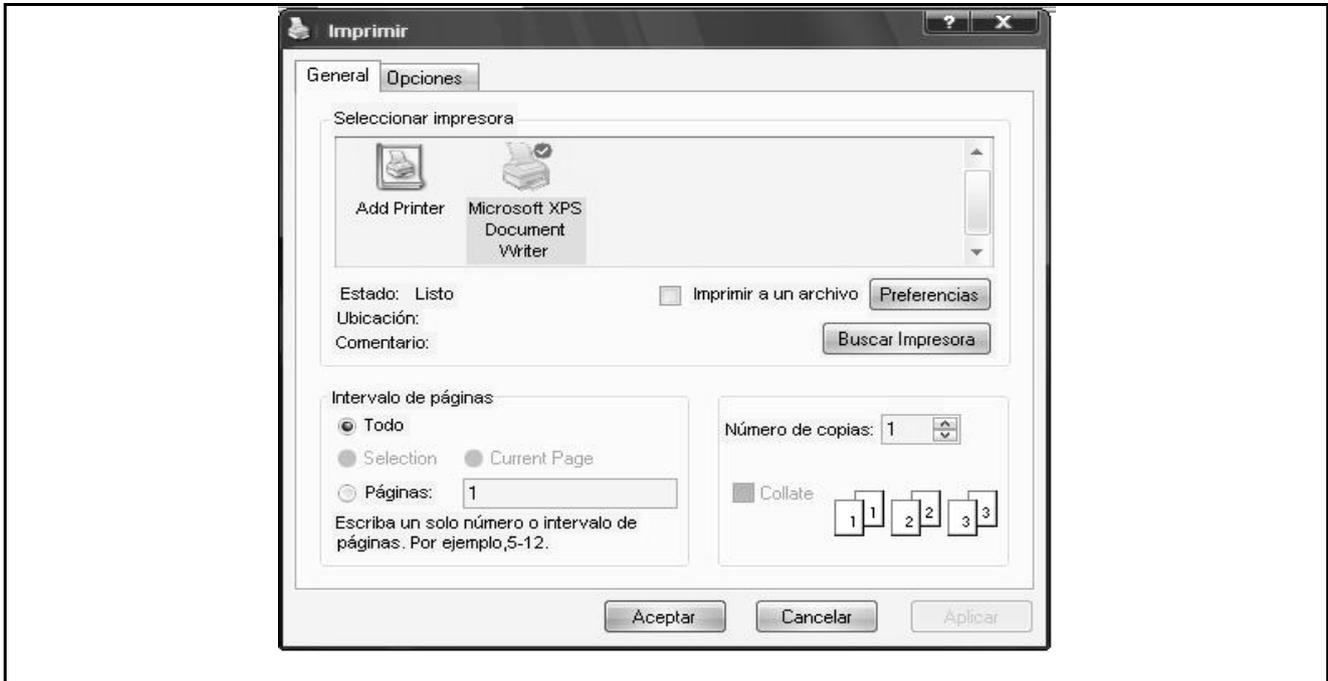
Sección “ Visualizar Reporte”



3. Se invoca la sección “Visualizar Reporte” del caso de uso:

- “Mostrar Reporte Plan de Alimento” en caso que el Reporte sea: Plan de Alimento.
- “Mostar Reporte de Gasto”. En caso que el reporte sea: Resumen de Gastos Corrientes o Resumen de Gastos Capitales.
- “Mostrar Reporte Plan Específico” en caso de que sea:
 - “Plan de Portadores Energéticos”
 - " Plan de Azúcar, Miel y Alcoholes"
 - "Plan de Producción Agropecuaria"
 - "Plan de Vestuario"
 - "Plan de Asistencia Médica"
 - "Plan de Aseguramiento Técnico"
 - “Plan de Mantenimiento Constructivo"
 - " Plan de Resto Calidad de Vida"
 - " Plan de Operaciones Comerciales y No comerciales"
 - " Plan de Otros Aseguramientos"
 - " Plan de Inversiones Constructivas"
 - " Plan de Equipos No Asociados a Inversiones Constructivas"
 - “Plan de Ciencia e Innovación Tecnológica"

Sección “ Imprimir”



Flujo normal de eventos

Acción del actor	Respuesta del sistema
1. El especialista selecciona <Imprimir>.	2. El sistema le muestra la Interfaz IX para que configure el reporte a imprimir.
3. El especialista configura el reporte selecciona <Aplicar> y posteriormente selecciona <Aceptar>.	4. El sistema imprime el reporte.

Cursos Alternos

Sección Imprimir

Línea 3. Si el actor decide <Cancelar> el sistema regresa a la interfaz inicial.

Línea 3. Si la impresora no está conectada el sistema envía mensaje al actor:



Pos-condiciones	Plan conformado

Tabla 17. Descripción Textual del caso de uso “Mostrar Reporte Tipo de Gasto.”

Caso de Uso	Mostrar Reporte Tipo de Gasto (Especialización)	
Actores	Especialista del Órgano Consumidor.(inicia)	
Propósito	El propósito de este caso de uso es que el especialista del OC pueda obtener reporte.	
Resumen	El caso de uso se inicia cuando el especialista del OC desea recuperar datos del Plan de Gastos Corrientes o de Gastos Capitales para su análisis o su posterior impresión. El sistema muestra una interfaz que le permite obtener el reporte haciendo una selección de diferentes criterios. Una vez efectuada la búsqueda muestra el reporte. El caso de uso termina cuando se visualiza el reporte.	
Tipo	Real y expandido.	
Precondiciones	El Plan de Demandas debe estar conformado.	
Referencias	R.2, R 2.1, R 2.1.1, R 2.1.2, R 2.1.3	
Flujo normal de eventos		
Acción del actor	Respuesta del sistema	
Sección “ Seleccionar Reporte”		
Se invoca a la sección: “Seleccionar Reporte” del caso de uso generalizado :“Obtener Reporte”		
Sección “ Visualizar Reporte”		
Acción del actor	Respuesta del sistema	



4. El sistema visualiza el resumen, en caso que sea:
 Gastos Capitales ir a la sección “Visualizar Gastos Capitales”.
 Castos Corrientes ir a la sección “Visualizar Gastos Corrientes”.

Sección “ Visualizar Gastos Capitales”

CÓDIGO DEL PRODUCTO	DESCRIPCIÓN	UM	PRECIO	TOTAL GASTOS CAPITALES		Equipos no asociados a inversiones constructivas		Ciencia e innovación tecnológica	
				CANT.	VALOR	CANT.	VALOR	CANT.	VALOR
Centro de Investig. Cient. Defensa Civil									
3372001022	MEDIOS DE DIAGNOSTICO	MP	1.000	143	143.000	87	87.000	56	56.000
4889991022	ENVASES PLÁSTICOS DE LABORATORIO	MP	1.000	253	253.000			253	253.000
	SUBTOTAL				396.000		87.000		309.000
Dirección de Armamento									
8240041014	EQUIPOS ESP CON ACCESORIOS	MP	1.000	256	256.000	256	256.000		
	SUBTOTAL				256.000		256.000		
Dirección de Inteligencia Militar									
8100110000	EQUIPOS PMADREPERLA (DIM)	MP	1.000	365	365.000	365	365.000		
	SUBTOTAL				365.000		365.000		
Dirección de Intendencia									
1001000000	AZUCAR CRUDO	T	0.252	264	66.620			264	66.620
1101000000	ARROZ CONSUMO	T	0.569	56	31.847	56	31.847		
1171000000	ACEITE VEG.REFINADO Y MANT CER	T	0.860	584	502.240			584	502.240
	SUBTOTAL				600.708		31.847		568.860
Dpto. de Fondos Materiales									
1012000000	TORULA	T	0.612	41	25.100			41	25.100

- (1) Botón Imprimir. **Nombre:** Imprimir **tipo:** button
- (2) Reporte del Plan seleccionado.
- (3) Plan visualizado. Su valor es una variable de sesión.

Acción del actor	Respuesta del sistema
5. El especialista analiza el reporte. En caso que desee: Imprimir ir sección “Imprimir “.	

Sección “ Visualizar Gastos Corrientes”



1

2

3

RESUMEN CONCEPTO DE GASTOS CORRIENTES AÑO 2008

CÓDIGO DEL PRODUCTO	DESCRIPCIÓN	UM	PRECIO	TOTAL GASTOS CORRIENTES		Plan de Portadores Energeticos		Operaciones comerciales, No Comerciales y TAJ		Otros Aseguramientos		CAN
				CANT.	VALOR	CANT.	VALOR	CANT.	VALOR	CANT.	VALOR	
Centro de Investig. Cient. Defensa Civil												
4889991022	ENVASES PLÁSTICOS DE LABORATORIO	MP	1.000	45	45.000			45	45.000			
9021001022	PPA P/FILTRO- HEPA	MP	1.000	127777777777	127.777.777.777.000							1277777
	SUBTOTAL				127.777.777.822.000				45.000			
Departamento de Fondo Material MIHFAI												
2700000000	CONEXIONES DE ACERO	MP	1.000	1	1.000			1	1.000			
	SUBTOTAL				1.000				1.000			
Dirección de Armamento												
8242010002	PPA P/ARMAMENTO (ECONOMÍA)	MP	1.000	55	55.000							
9202001014	CONTINGENCIA 3 %	MP	1.000	414	414.000					414	414.000	
	SUBTOTAL				469.000						414.000	
Dirección de Construcción y Alojamiento												
5360001010	OTROS PROD. DE LA INDUSTRIA	MP	1.000	24	24.000			14	14.000			
	SUBTOTAL				24.000				14.000			

- (1) Botón Imprimir. **Nombre:** Imprimir **tipo:** button
- (2) Reporte del Plan seleccionado.
- (3) Plan visualizado. Su valor es una variable de sesión.

Acción del actor	Respuesta del sistema
<p>5. El especialista analiza el reporte. En caso que desee: Imprimir ir sección "Imprimir".</p>	

Sección " Imprimir"

Se invoca a la sección: "Imprimir" del caso de uso generalizado: "Obtener Reporte".

Cursos Alternos

Línea 4. En caso que no contengas datos el Plan selecciona el sistema envía el mensaje:

NO SE ENCONTRARON DATOS.

Pos-condiciones	Obtiene un resumen por un tipo de gasto.
Prioridad	Secundario



Tabla 18. Descripción Textual del caso de uso “Mostrar Reporte Plan de Alimentos.”

Caso de Uso	Mostrar Reporte Plan de Alimentos (Especialización)																																																									
Actores	Especialista del Órgano Consumidor. (inicia)																																																									
Propósito	El propósito de este caso de uso es que el especialista del OC pueda obtener reporte.																																																									
Resumen	El caso de uso se inicia cuando el especialista del OC desea obtener un resumen del Plan de Alimentos para su análisis o Impresión del documento generado. El sistema muestra una interfaz que le permite obtener el plan haciendo una selección de diferentes criterios. Una vez efectuada la búsqueda muestra el Plan. El caso de uso termina cuando se visualiza el reporte.																																																									
Precondiciones	El Plan de Demandas debe estar conformado.																																																									
Referencias	R.2, R 2.3, R 2.3.1																																																									
Flujo normal de eventos																																																										
Acción del actor	Respuesta del sistema																																																									
Sección “ Seleccionar Reporte”																																																										
Se invoca a la sección: “Seleccionar Reporte” del caso de uso generalizado :“Obtener Reporte”																																																										
Sección “ Visualizar Reporte”																																																										
<table border="1"> <thead> <tr> <th rowspan="2">CÓDIGO DEL PRODUCTO</th> <th rowspan="2">DESCRIPCIÓN</th> <th rowspan="2">UM</th> <th rowspan="2">PRECIO</th> <th colspan="2">TOTAL ALIMENTOS</th> <th colspan="4">UAM</th> </tr> <tr> <th>CANTIDAD</th> <th>VALOR</th> <th colspan="2">TOTAL</th> <th colspan="2">Fondos mercantiles</th> </tr> <tr> <th></th> <th></th> <th></th> <th></th> <th>CANTIDAD</th> <th>VALOR</th> <th>CANTIDAD</th> <th>VALOR</th> <th>CANTIDAD</th> <th>VALOR</th> </tr> </thead> <tbody> <tr> <td>3294600082</td> <td>PROD. QUM AUX P/SIST.ORDENO</td> <td>MP</td> <td>1,000.000</td> <td>100</td> <td>100.000</td> <td>100</td> <td>100.000</td> <td>100</td> <td>100.000</td> </tr> <tr> <td>9021001022</td> <td>PPA P/FILTRO-HEPA</td> <td>MP</td> <td>1,000.000</td> <td>127777777777</td> <td>127,777,777,777.000</td> <td>127777777777</td> <td>127,777,777,777.000</td> <td></td> <td></td> </tr> <tr> <td colspan="4">Total:</td> <td></td> <td>127,777,777,877.000</td> <td></td> <td>127,777,777,877.000</td> <td></td> <td>100.000</td> </tr> </tbody> </table>			CÓDIGO DEL PRODUCTO	DESCRIPCIÓN	UM	PRECIO	TOTAL ALIMENTOS		UAM				CANTIDAD	VALOR	TOTAL		Fondos mercantiles						CANTIDAD	VALOR	CANTIDAD	VALOR	CANTIDAD	VALOR	3294600082	PROD. QUM AUX P/SIST.ORDENO	MP	1,000.000	100	100.000	100	100.000	100	100.000	9021001022	PPA P/FILTRO-HEPA	MP	1,000.000	127777777777	127,777,777,777.000	127777777777	127,777,777,777.000			Total:					127,777,777,877.000		127,777,777,877.000		100.000
CÓDIGO DEL PRODUCTO	DESCRIPCIÓN	UM					PRECIO	TOTAL ALIMENTOS		UAM																																																
			CANTIDAD	VALOR	TOTAL			Fondos mercantiles																																																		
				CANTIDAD	VALOR	CANTIDAD	VALOR	CANTIDAD	VALOR																																																	
3294600082	PROD. QUM AUX P/SIST.ORDENO	MP	1,000.000	100	100.000	100	100.000	100	100.000																																																	
9021001022	PPA P/FILTRO-HEPA	MP	1,000.000	127777777777	127,777,777,777.000	127777777777	127,777,777,777.000																																																			
Total:					127,777,777,877.000		127,777,777,877.000		100.000																																																	



(1) Botón Imprimir. Nombre: Imprimir tipo: button	
(2) Plan visualizado. Su valor es una variable de sesión.	
(3) Reporte del Plan seleccionado.	
Acción del actor	Respuesta del sistema
	3. El sistema visualiza el reporte.
5. El especialista analiza el reporte. En caso que desee: Imprimir ir sección "Imprimir".	
Sección " Imprimir"	
Se invoca a la sección: "Imprimir" del caso de uso generalizado: "Obtener Reporte".	
Pos-condiciones	Obtiene el reporte del Plan de Alimentos.
Prioridad	Secundario

2.15 Conclusiones

Hasta el momento se ha visto como a partir del análisis de los procesos del negocio y de los artefactos obtenidos en el modelamiento del mismo se identificaron los requerimientos, los cuales ayudarán a desarrollar las funcionalidades que el software a construir debe cumplir. Estas se representaron mediante el Diagrama de Casos de Uso y finalmente se describieron paso a paso todas las acciones de los actores del sistema con los casos de uso con los que interactúan. Todo esto constituye la base para empezar la construcción del sistema. La descripción textual del caso de uso "Mostrar Reporte Plan Específico" se puede observar en el Anexo1. La descripción textual del Caso de Uso Autenticar Usuario no se desarrollará porque no es objetivo del trabajo.



CAPÍTULO 3. ANÁLISIS Y DISEÑO DEL SISTEMA

3.1 Introducción.

En este capítulo se procede a la confección de la propuesta de solución, basándose en el análisis hecho anteriormente. Para el desarrollo de este capítulo se tuvieron en cuenta los requisitos funcionales, como no funcionales obtenidos durante el flujo de la captura de requisitos. El mismo constará con varios artefactos que tienen como objetivos mostrar el diseño y la implementación del sistema propuestos como solución.

3.2 Modelo de Análisis.

El modelo de análisis es la primera representación técnica del sistema, por lo que puede considerarse como una primera aproximación al modelo de diseño, y es por tanto una entrada fundamental para las actividades de diseño e implementación subsiguientes. A partir de su realización se refinan y estructuran los requisitos obtenidos con anterioridad proporcionando una visión general del sistema.

El modelo de análisis debe lograr tres objetivos primarios:

1. Describir lo que requiere el cliente.
2. Establecer una base para la creación de un diseño de software.
3. Definir un conjunto de requisitos que se pueda validar una vez que se construye el software.

A continuación se muestran los diagramas de clases del análisis correspondiente a cada uno de los casos de usos que se describieron en el capítulo anterior.

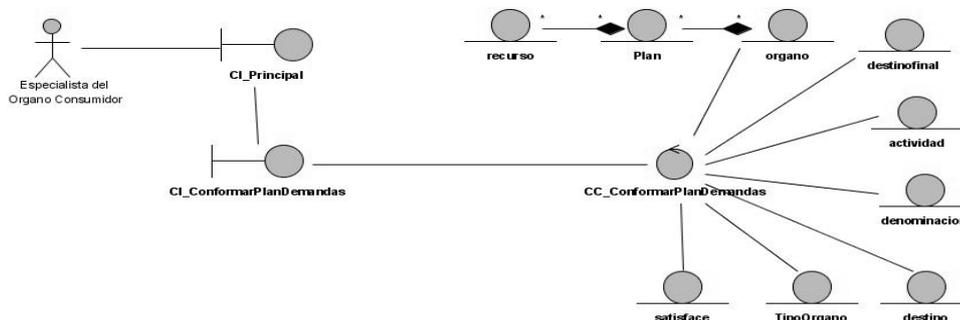


Figura 9. Diagrama de clases del análisis “Actualizar Plan de Demandas”.

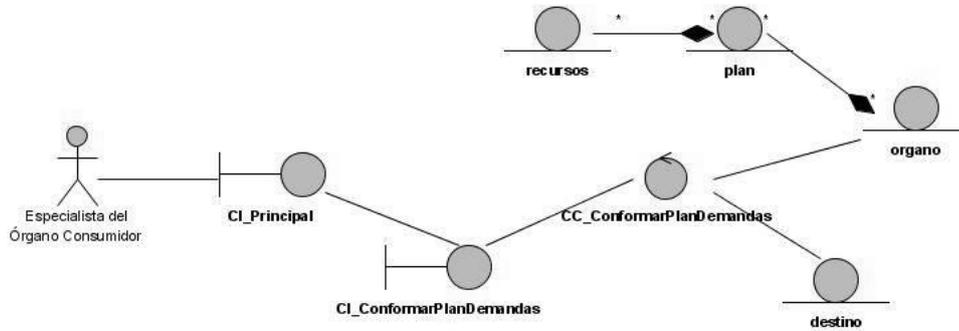


Figura 10. Diagrama de clases del análisis “Aprobar Plan de Demandas”.

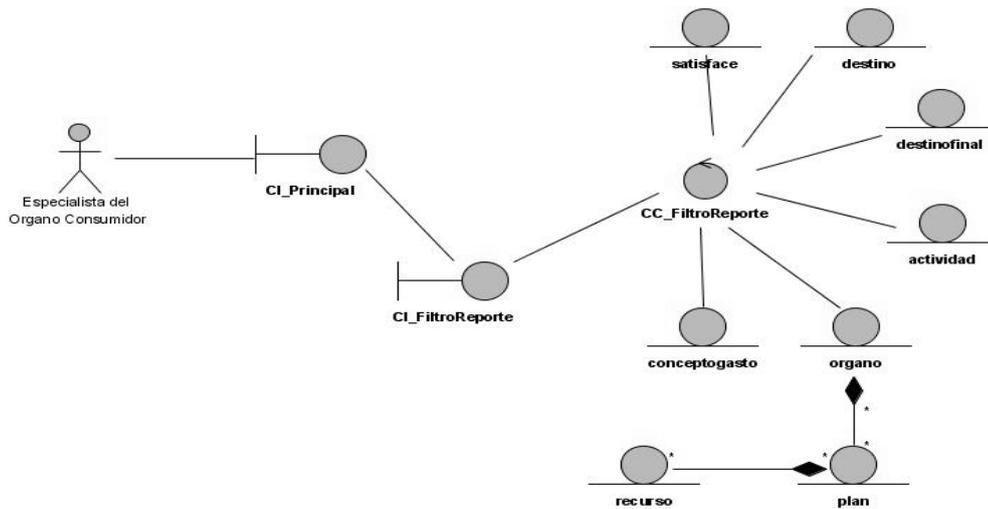


Figura 11. Diagrama de clases del análisis “Seleccionar Criterios de Búsqueda”.

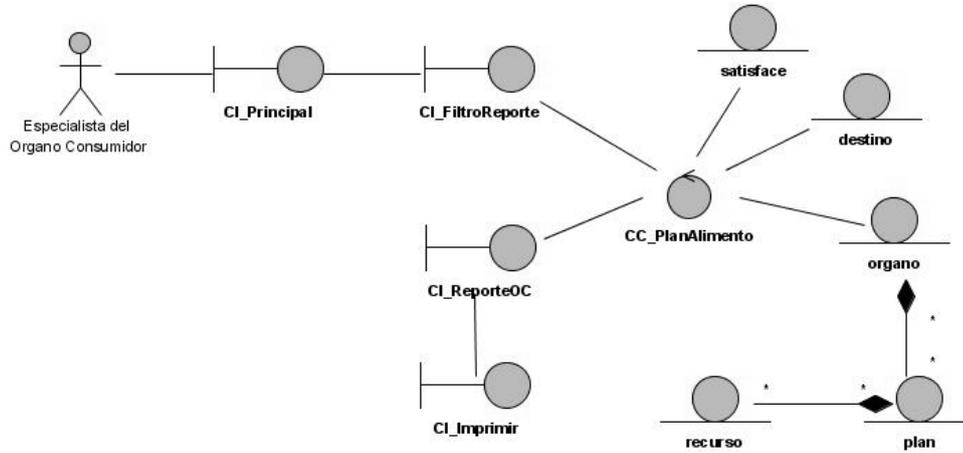


Figura 12. Diagrama de clases del análisis “Mostar Reporte Plan de Alimento”.

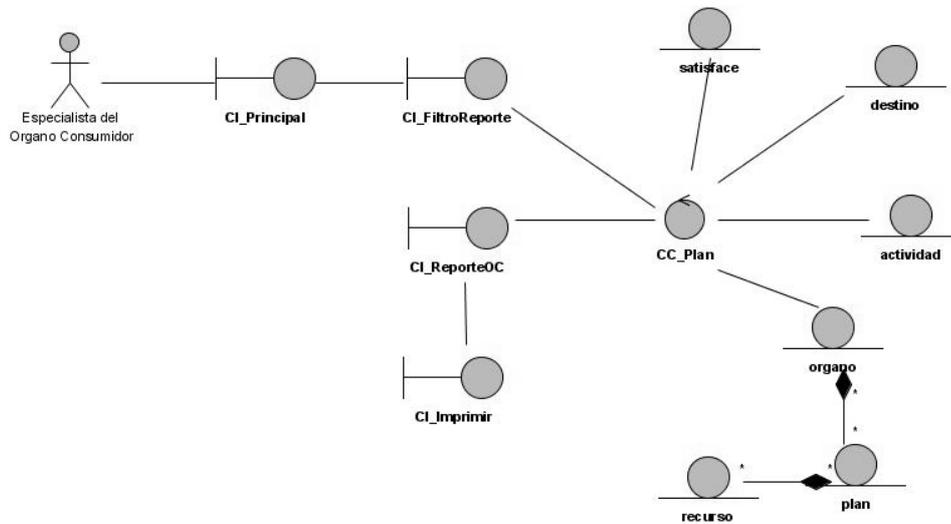


Figura 13. Diagrama de clases del análisis “Mostar Reporte Plan Especifico”.

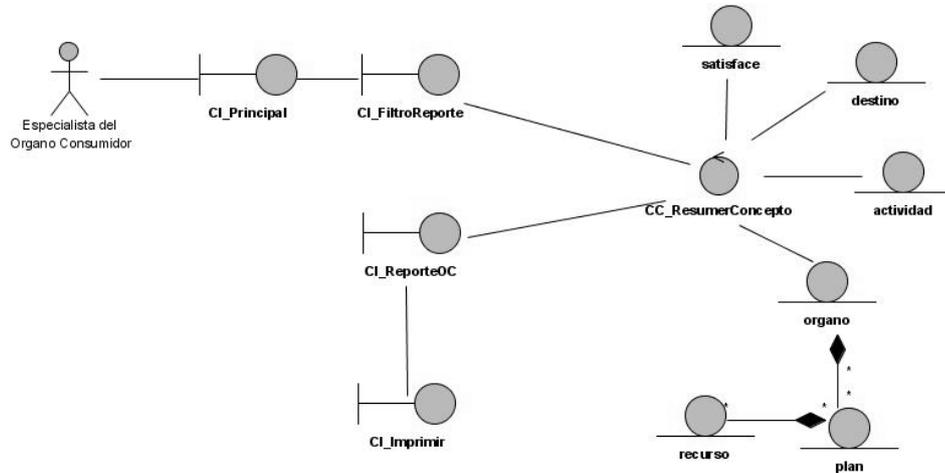


Figura 14. Diagrama de clases del análisis “Mostar Reporte Plan Concepto Gasto”.

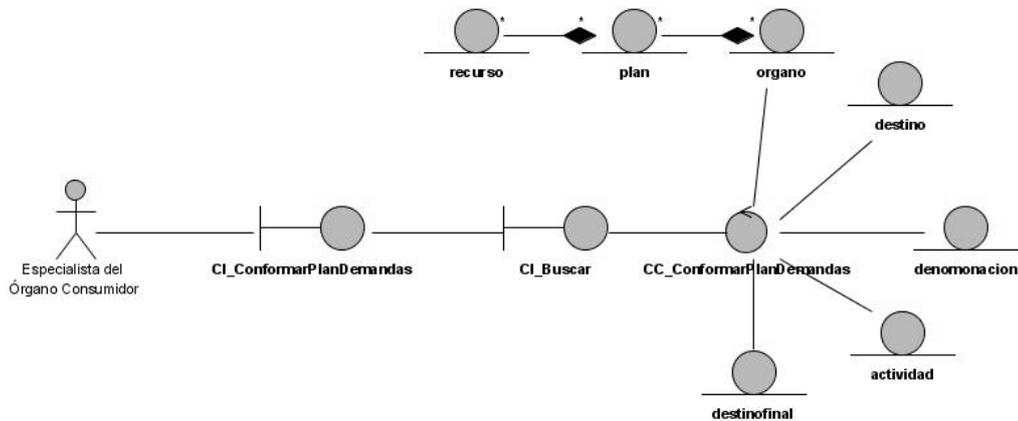


Figura 15. Diagrama de clases del análisis “Buscar Datos”.

3.3 Modelo de Diseño.

El diseño constituye el centro de atención de la propuesta de solución, el mismo contribuye a una arquitectura estable y sólida. En el diseño se modela el sistema y de forma incluida la arquitectura para que soporte todos los requisitos, incluyendo los no funcionales y las restricciones que se le suponen. Una entrada esencial en el diseño es el resultado del análisis, o sea el modelo de análisis, que proporciona una



comprensión detallada de los requisitos. Además impone una estructura del sistema que se debe conservar lo más fielmente posible cuando se de forma al sistema.

Antes de realizar los diagramas que muestran la solución del sistema, es necesario definir algunos conceptos que ayudan a obtener un producto de mejor calidad.

3.4 Arquitectura.

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

Dentro de la arquitectura hay elementos de vital importancia que no se puede dejar de mencionar: *los patrones de arquitectura*.

¿Qué es un patrón?

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

¿A qué responden los patrones de arquitectura?

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

Para el desarrollo de la aplicación se propone utilizar el patón de Capas.

¿Por qué patrón Capas?

Este patrón define cómo organizar el modelo de diseño en capas, que pueden estar físicamente distribuidas, lo cual quiere decir que los componentes de una capa sólo pueden hacer referencia a componentes en capas inmediatamente inferiores. Este patrón es importante porque simplifica la comprensión y la organización del desarrollo de sistemas complejos, reduciendo las dependencias de forma que las capas más bajas no son conscientes de ningún detalle o interfaz de las superiores. Además,



ayuda a identificar qué puede reutilizarse. A continuación una descripción detallada de la arquitectura que se propone.

Arquitectura de n Capas.

Esta arquitectura brinda la ventaja de aislar definitivamente la lógica de negocios de todo lo que tenga que ver con el origen de datos, ya que desde el manejo de la conexión, hasta la ejecución de una consulta, la manejará la capa de Acceso a Datos. De este modo, ante cualquier eventual cambio, solo se deberá cambiar una capa en específico.

- 1. Capa de presentación o capa de interfaz de usuario:** es la forma de ofrecer al usuario un modo de interactuar con la aplicación. Las interfaces de usuario se implementan utilizando formularios, controles u otro tipo de tecnología que permita procesar y dar formato a los datos de los usuarios, así como adquirir y validar los datos entrantes procedentes de éstos.
- 2. Capa de lógica de negocio:** es la parte más importante de una aplicación debido a que encapsula las responsabilidades y los procesos que en ella se comprenden, su funcionamiento basado en el uso de clases o componentes y a su vez compuestos estos por métodos que de una forma u otra establecen la relación entre los componentes de presentación y los componentes de acceso a datos. Para el acceso al subsistema de la capa de acceso a datos se presenta una interfaz denominada “Factoría típica”, clase que implementa el patrón para clases con un objetivo y funcionalidades similares, o sea todo flujo de información entre estas dos capas es a través de esta clase.
- 3. Capa de Acceso a Datos:** permite conocer el formato de los datos que se intercambian a través de los componentes de acceso a datos y la capa lógica de negocio, permiten el acceso a los datos a través de los mismos. Además usan la tecnología de acceso a datos PDO e implementa un interfaz de programación para la gestión de los datos.
- 4. Fuente de Datos:** Contiene el almacenamiento de los datos, es decir todas las tablas de la Base de Datos.



¿Por qué una Arquitectura de n Capas?

1. **Abstracción total acerca del origen de datos:** las distintas capas se especializan absolutamente en la funcionalidad que deben brindar sin importar cual es el origen de los datos procesados.
2. **Bajo costo de desarrollo y mantenimiento de las aplicaciones:** esta arquitectura brinda un control más cercano de cada componente, así como también la posibilidad de una verdadera reutilización del código.
3. **Mejor calidad en las aplicaciones:** como las aplicaciones son construidas en unidades separadas, estas pueden ser probadas independientemente y con mucho más detalle, esto conduce a obtener un producto mucho más sólido.
4. **Reutilización de código:** la concepción natural de un sistema desarrollado con esta arquitectura, promueve la reutilización de sus componentes en varias partes del propio desarrollo y de futuros sistemas.

En la figura 16 que se muestra a continuación se puede ver el patrón propuesto para la solución.

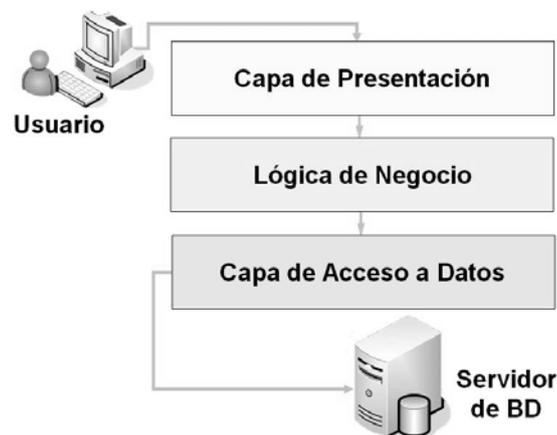


Figura 16. Arquitectura de n Capas.



3.5 Mecanismo de Diseño.

Los mecanismos de diseño se modelan para describir una sola vez un conjunto de operaciones y clases que siempre participan en las realizaciones de los casos de uso. Con su elaboración, se modelaría un conocimiento que ayudará tanto al desarrollo de la aplicación actual como a futuras iteraciones, además de las labores de mantenimiento. En los mecanismos de diseño intervienen diversos elementos de la aplicación (clases, subsistemas).

Los mismos reportan beneficios como:

1. Mantener la homogeneidad en el diseño.
2. Reutilizar soluciones anteriormente probadas.
3. Fomentar la eficiencia y la reutilización.

Un ejemplo típico de mecanismos son los patrones de diseño.

¿Qué es un Patrón de Diseño?

Un patrón de diseño es una solución a un problema de diseño no trivial que es efectiva y reusable, igualmente es una descripción de clases y objetos comunicándose entre sí que resuelve un problema de diseño general en un contexto particular.

Los patrones de diseño han contribuido a dar flexibilidad y extensibilidad a los 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.

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

En la actualidad son muchos los patrones de diseños utilizados en la construcción de una aplicación y en base a las responsabilidades asignadas se clasifican en:



1. Experto.
2. Creador.
3. Alta Cohesión.
4. Bajo Acoplamiento.
5. Controlador.

Dentro de patrón creadores definidos por la GoF (Gang of Four) se encuentra el patrón de Factoría.

Este patrón es un modelo que utiliza abstracción de clases para crear y relacionar objetos sin conocer de qué clase son. Se utiliza cuando la aplicación no sabe de antemano el tipo de objeto que se va a crear, es en tiempo de ejecución cuando toma la decisión.

3.5.1 Mecanismo de Seguridad.

Las aplicaciones Web permiten el acceso de usuarios a recursos centrales como el servidor Web y, a través de éste, a otros como los servidores de base de datos. Con la implementación correcta de medidas de seguridad, pueden protegerse los recursos con los que se cuenta, así como proporcionar un entorno seguro a los usuarios que trabajen con la aplicación.

Administrar centralmente la seguridad de la aplicación es un factor esencial para controlar el acceso a la misma, esto se logra a través del empleo de variables de sesión, en la cual se registra la identidad del usuario, una vez se autentica en el sistema.

Al constituir esta aplicación como módulo de un ERP, las variables de sesión necesarias se obtienen a través de otro módulo encargado de la seguridad, que brinda esta posibilidad por medio de un servicio Web, el cual proporciona una interfaz cControlacceso para la autenticación de los usuarios, esta interfaz contiene un método público llamado logueo, que recibe como parámetro el usuario, la contraseña y el módulo al que desea entrar, este método devolverá un mensaje de error en caso de que exista algún problema.

En caso de éxito en la autenticación el sistema devuelve además de las variables de sesión necesarias para tratar al usuario en el sistema, el menú y la barra de herramienta que será diferente para cada usuario en dependencia del rol. El rol del usuario es una de las variables de sesión de mayor importancia ya que esto indica al sistema los privilegios que tiene el usuario autenticado.



A continuación se muestra en la Figura 17 el mecanismo de diseño para la seguridad del sistema basado en el uso de servicios Web.

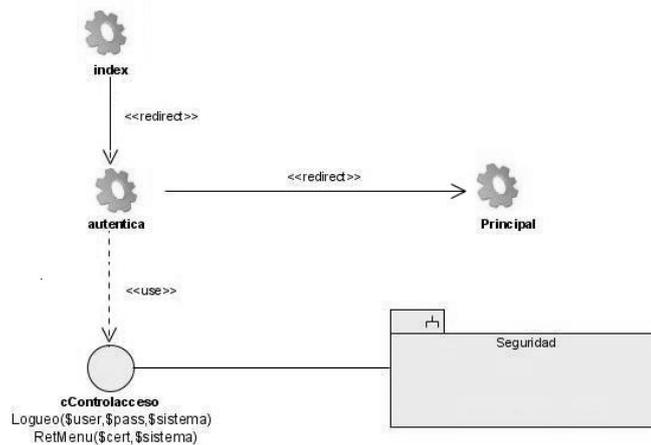


Figura 17. Mecanismo de Seguridad.

3.5.2 Mecanismo de Acceso a Datos.

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

Este mecanismo surge con el objetivo de facilitar el acceso a datos del sistema, creándose un punto de referencia para los desarrolladores, permitiendo obtener diagramas más entendibles, que posibiliten una mayor comunicación con el equipo de desarrollo, trazando una línea común, una política a seguir, fomentando algo muy indispensable para lograr eficiencia, la reutilización.

La vista estática de este mecanismo de acceso a datos (Figura 18) visualiza un conjunto de clases que interactúan para dar acceso y manipulación de los datos de la persistencia desde el nivel más bajo, es decir, utilizando los objetos nativos brindados por el entorno de desarrollo PHP como son la extensión PDO que define una ligera interfaz, para el acceso a bases de datos., siguiendo así hasta la abstracción del acceso a datos a través de meBase de la cual heredan las clases particulares del sistema como Típicas y meSimples.



Para dar la responsabilidad a una clase que encapsulara las instancias de estos objetos se definió la clase FactoríaTípica, que está concebida para instanciar todas las típicas del sistema para utilizarlas de algún modo.

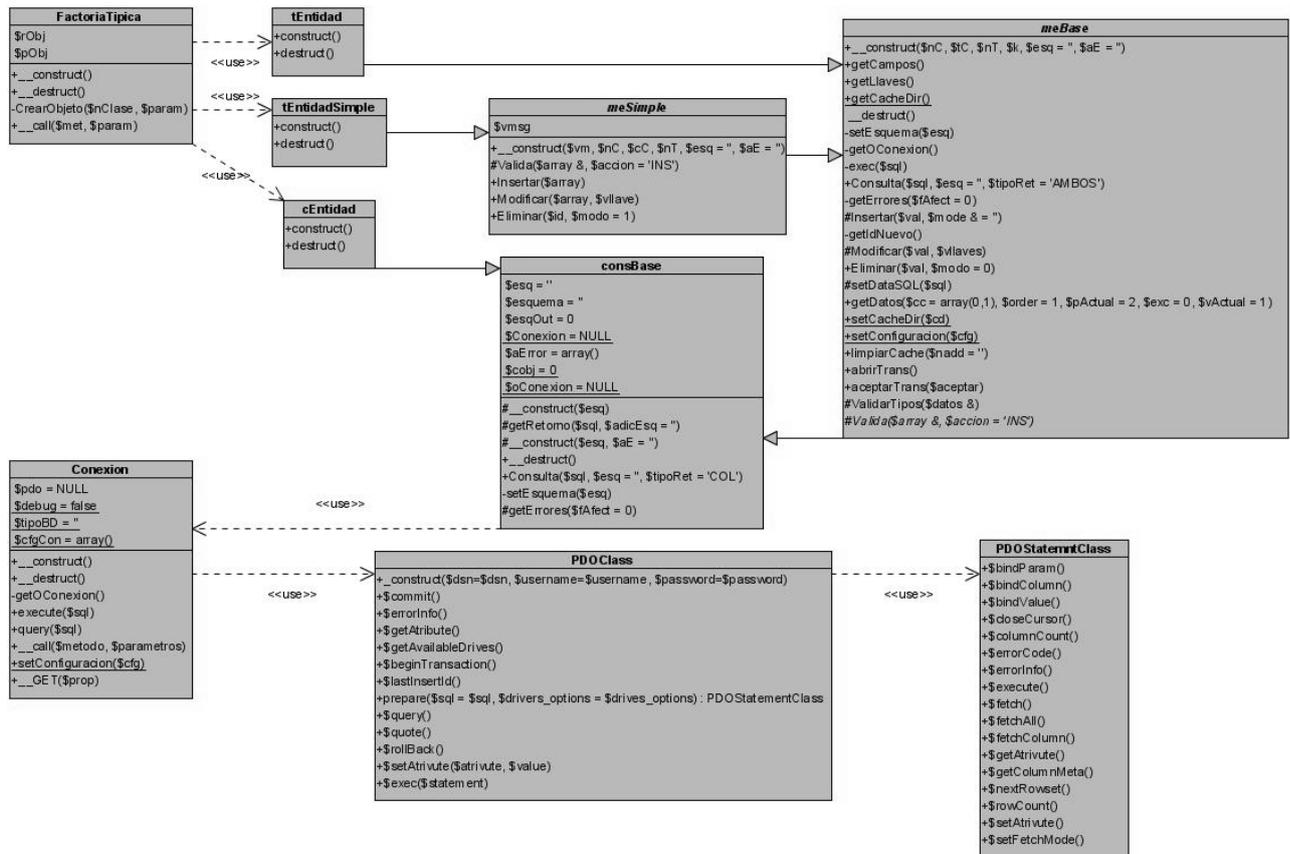


Figura 18. Diagrama general del “Mecanismo de Acceso a Datos”.

A continuación se muestra una descripción más detallada de estas clases.

FactoríaTípica: Clase que implementa la interfaz del modelo de persistencia con el resto de los subsistemas. A través de esta clase se crean y se manipulan los objetos de las típicas simples, los



nomencladores y las demás típicas. Es una puerta entre la capa de Acceso a Datos y la capa de Lógica de Negocio. Implementa un método de instancia de clases típicas.

tEntidadSimple: Es una clase que representa a las clases típicas (nomencladores simples) en general de la aplicación. Estas típicas son de una implementación muy sencilla, pues la mayoría de las líneas que normalmente habrían que codificar quedaron encapsuladas en la clase base de las mismas. Para la implementación de esta clase se decidió aplicar el patrón de diseño Table Data Gateway, que consiste en crear una instancia por cada tabla existente en la BD. Sus métodos consisten en las operaciones básicas que se realizan sobre estas tablas, insertar, modificar y eliminar. Hereda de la clase abstracta meSimple.

meSimple: Clase abstracta heredera de meBase, y la vez base para la implementación de las típicas que responderán a los nomencladores simples del modelo de persistencia dado. Redefine las operaciones básicas con la funcionalidad de Validación dada. Redefine las operaciones básicas que pudieran realizarse a una entidad (insertar, eliminar, modificar) para los nomencladores simples.

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

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

Conexión: Esta clase es la encargada de establecer la conexión con el servidor de la base de datos a través de un objeto PDO de la librería de PHP.

PDO: Es un modelo de acceso a datos para PHP que brinda una capa de abstracción para el acceso a BD desde PHP.



3.6 Diagramas de Interacción.

Los diagramas de interacción muestran las interacciones entre objetos mediante transferencia de mensajes entre objetos o subsistemas.

Los diagramas de interacción se clasifican en Diagramas de Colaboración y Diagramas de Secuencias.

Un *diagrama de colaboración* muestra las interacciones entre objetos creando enlaces entre ellos y añadiendo mensajes a esos enlaces.

Un *diagrama de secuencia* muestra las interacciones entre objetos, ordenadas en secuencia temporal durante un escenario concreto.

En el diseño, se decidió representar estas interacciones mediante diagramas de secuencias ya que el centro de atención principal es encontrar secuencia de interacciones detalladas y ordenadas en el tiempo.

A continuación se muestran los diagramas de interacción (secuencia) de los distintos casos de uso.

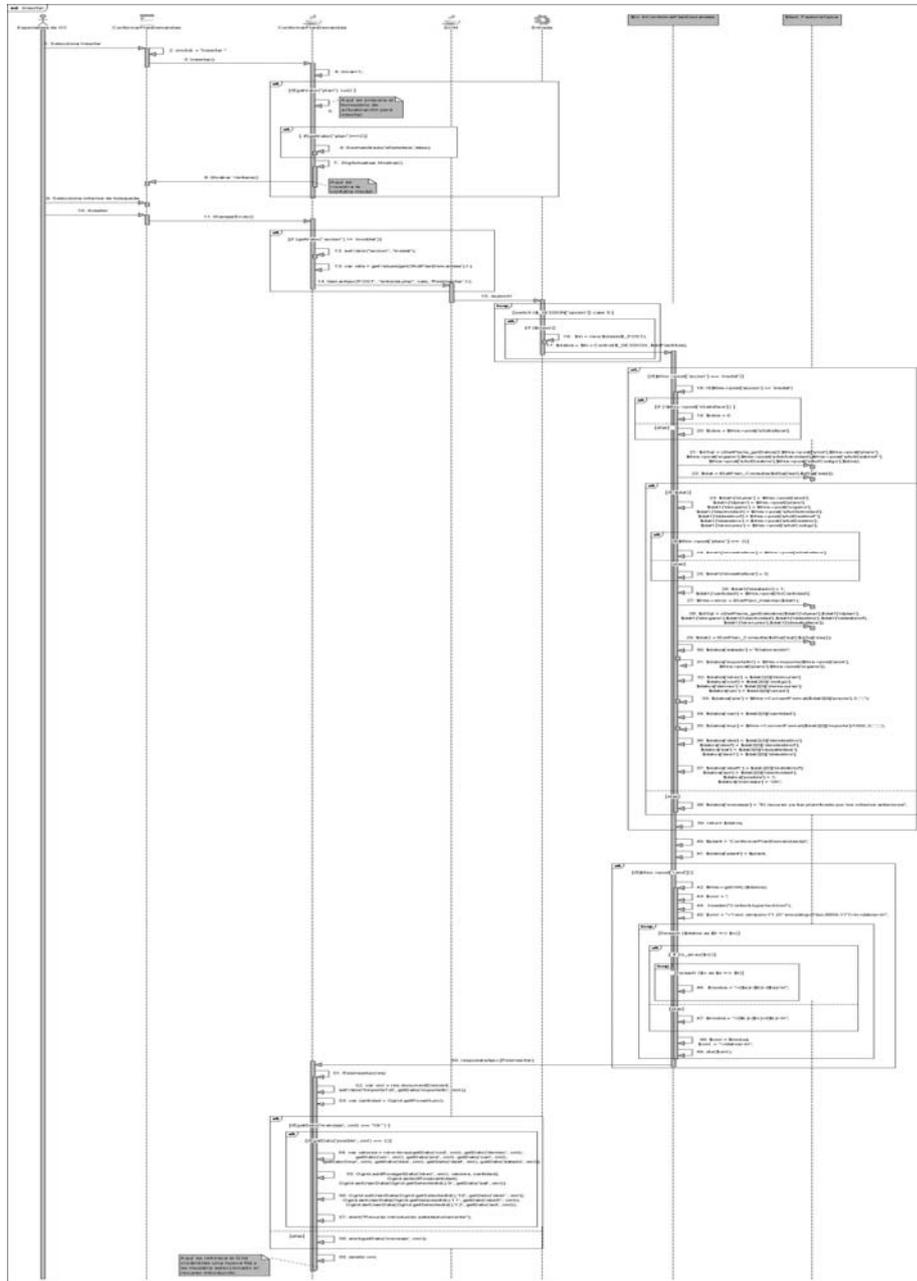


Figura 20. Realización Insertar (Caso de Uso Actualizar Plan)”.



3.7 Diagrama de Clases del Diseño.

Para obtener un nivel correcto de abstracción y detalle que permita obtener un resultado final, es mejor modelar los artefactos del sistema, es decir, modelar las páginas, los enlaces entre ellas, todo el código que irá creando las páginas, así como el contenido dinámico de estas una vez que estén en el navegador del cliente, para ello se realizarán los diagramas de clases.

Su objetivo fundamental es crear una entrada apropiada y un punto de partida para la implementación, capturando los requisitos o subsistemas individuales, interfaces y clases. El diseño convierte el modelo de análisis en diseños de datos, arquitectónicos, de interfaz y a nivel de componentes del software.

Este sistema tiene concebido por cada caso de uso lo siguiente:

- Una clase de lógica de negocio (In_**NombreCU**) por cada caso de uso.
- Un fichero **GLOBAL** para la configuración de las direcciones de las clases a través de la función __autoload y algún elemento de configuración.
- La página servidora tendrá las instancias de los objetos In_**NombreCU**.
- Una página servidora **principal** que redireccionará a servidora de cada Caso de Uso por módulo.
- Una interfaz **cControlacceso** que permite la autenticación de los usuarios.
- Una clase **InBase**: Es la clase base de lógica de negocio, de la cual heredarán todas las clases "In_**NombreCU**". En ella se define el objeto de la clase Factoría Típica.
- Una clase **Miscelanea** donde se encuentran métodos que pueden ser empleados en diferentes Casos de Uso, como son métodos para formateo de fechas, cantidades, etc.
- Una clase **FactoríaTípica** que a través de la cual se crean y se manipulan los objetos de acceso a datos como las tablas y consultas. Es una interfaz entre la Lógica de Negocio y la capa de Acceso a Datos. Basada en el patrón Factoría.
- El fichero **DOM** que es un repositorio de funciones que son comunes en la mayoría de los CU.
- El fichero **dhtmlX** que son los ficheros dhtmlX controls (menú, toolbar).



Con el objetivo de optimizar la modelación del diseño de la aplicación en desarrollo, se construyó un diagrama genérico de clases del diseño (Figura 23), que tiene como misión la representación de los elementos comunes para todos los diagramas de clases a desarrollar, destacando en color azul el nombre de las clases que varían en dependencia del diagrama que se esté realizando, por tanto los posteriores diagramas solamente contendrán las páginas clientes, servidoras, las lógicas de negocio, y en caso de tener formularios y ficheros java script también serán incluidos.

A continuación se muestran los diagramas de clases del diseño.

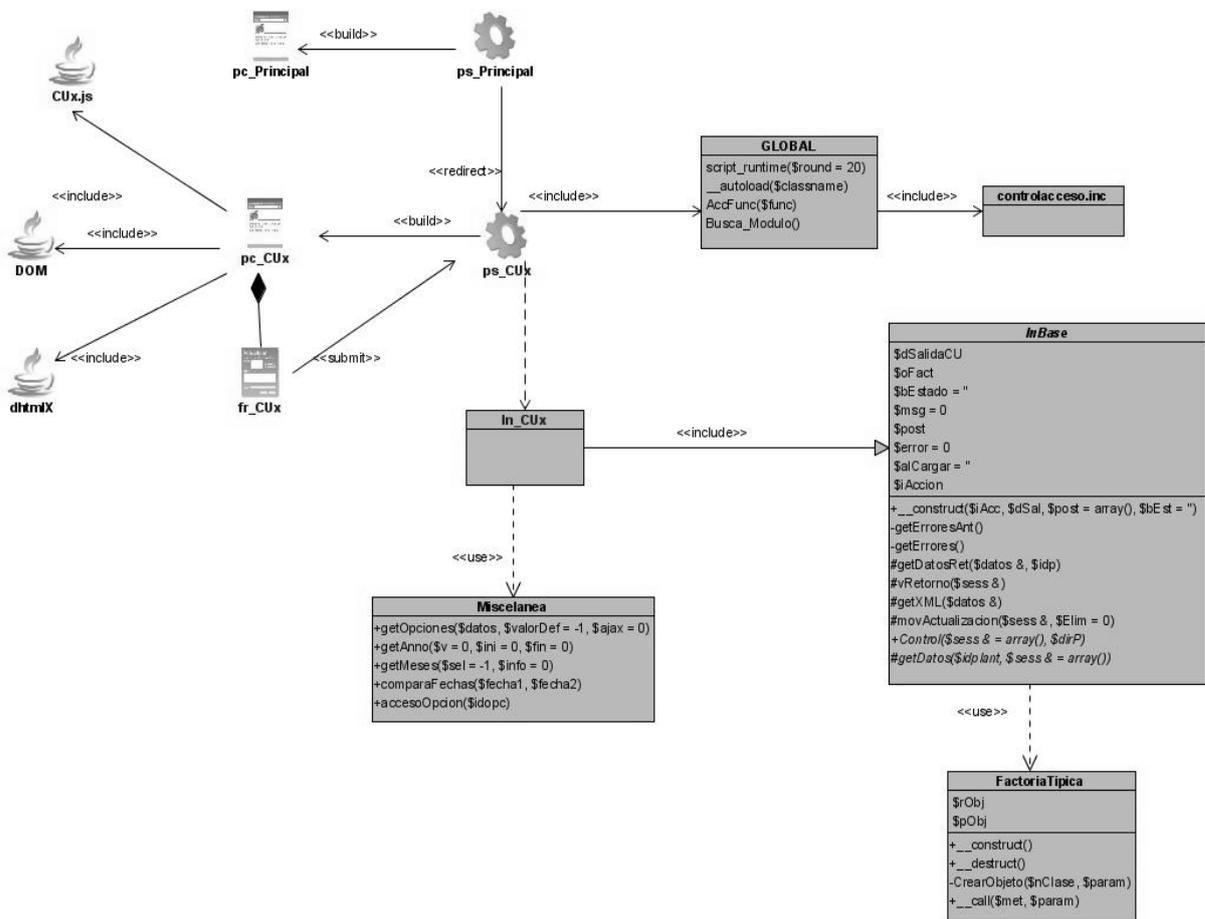


Figura 23. Diagrama de clases del diseño "Genérico".

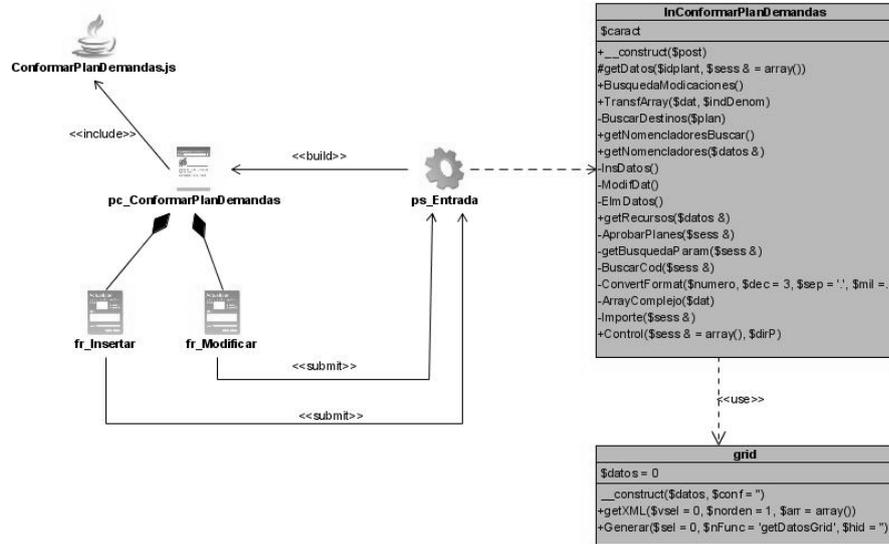


Figura 24. Diagrama de clases del diseño “Actualizar Plan de Demandas”.

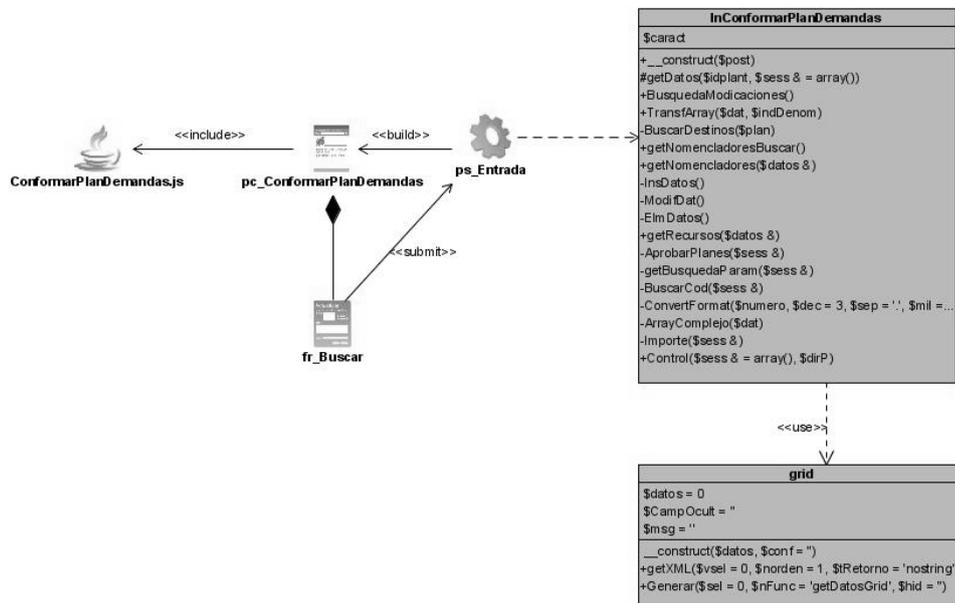


Figura 25. Diagrama de clases del diseño “Aprobar Plan de Demandas”.

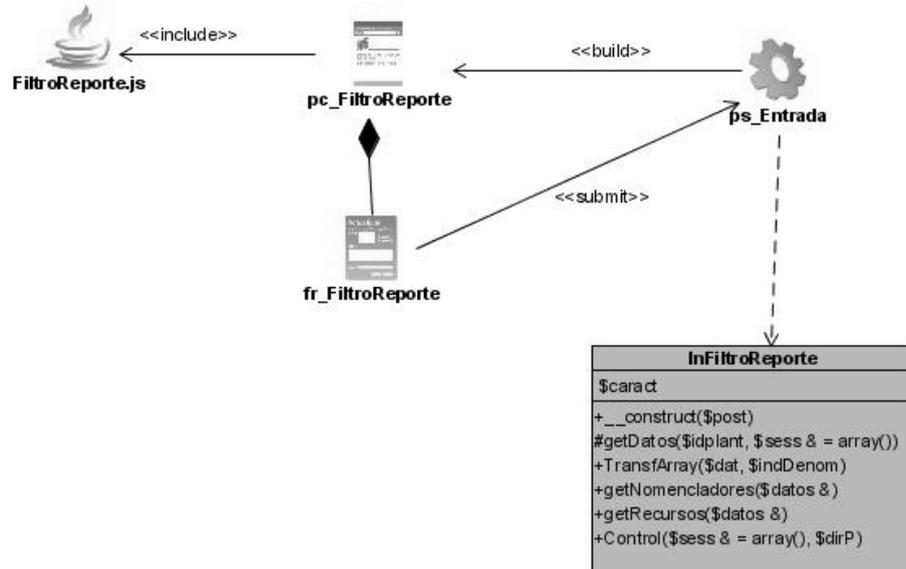


Figura 26. Diagrama de clases del diseño “Seleccionar Criterios de Búsqueda”.

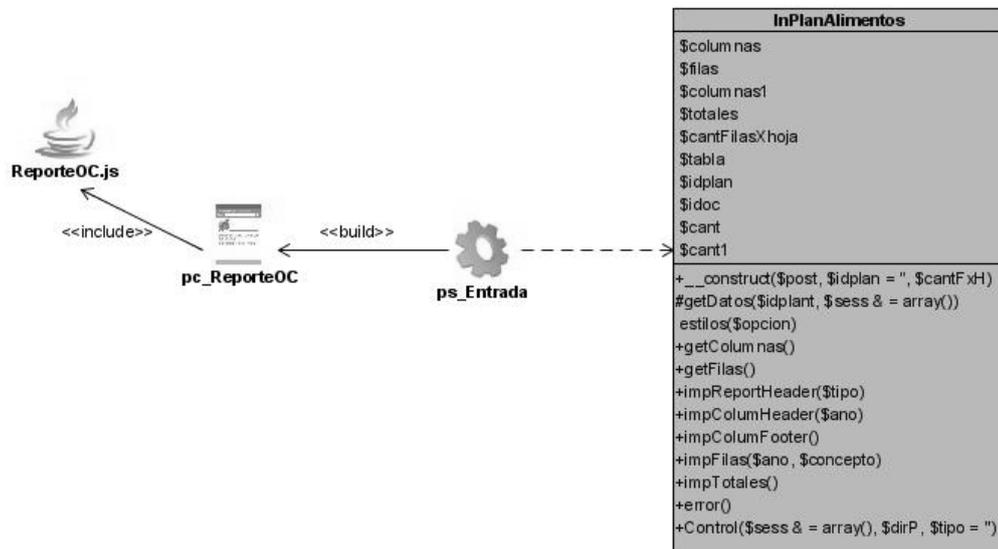


Figura 27. Diagrama de clases del diseño “Mostrar Reporte Plan de Alimento”.

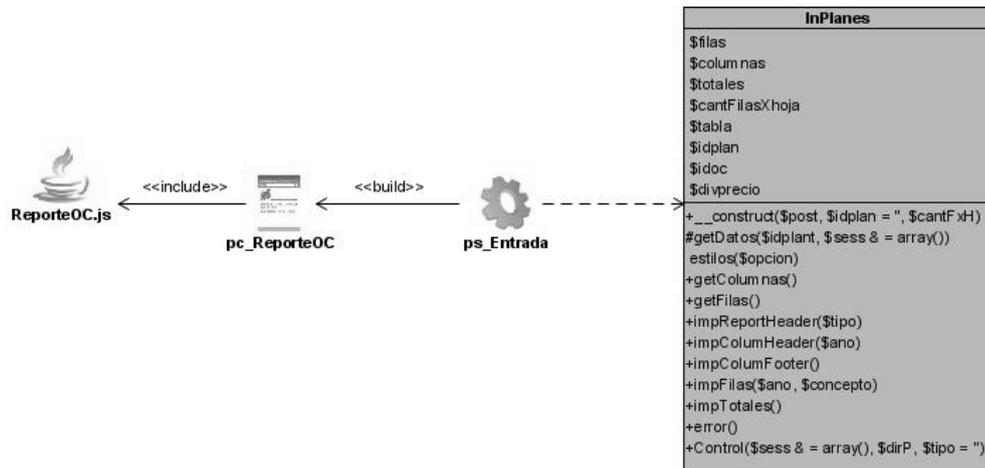


Figura 28. Diagrama de clases del diseño “Mostrar Reporte Plan Específico”.

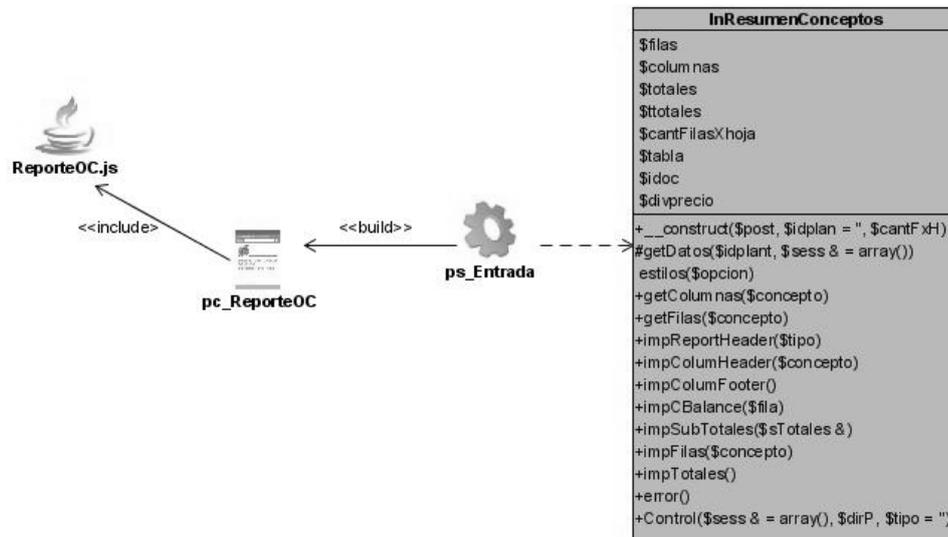


Figura 29. Diagrama de clases del diseño “Mostrar Reporte Plan Concepto Gasto”.

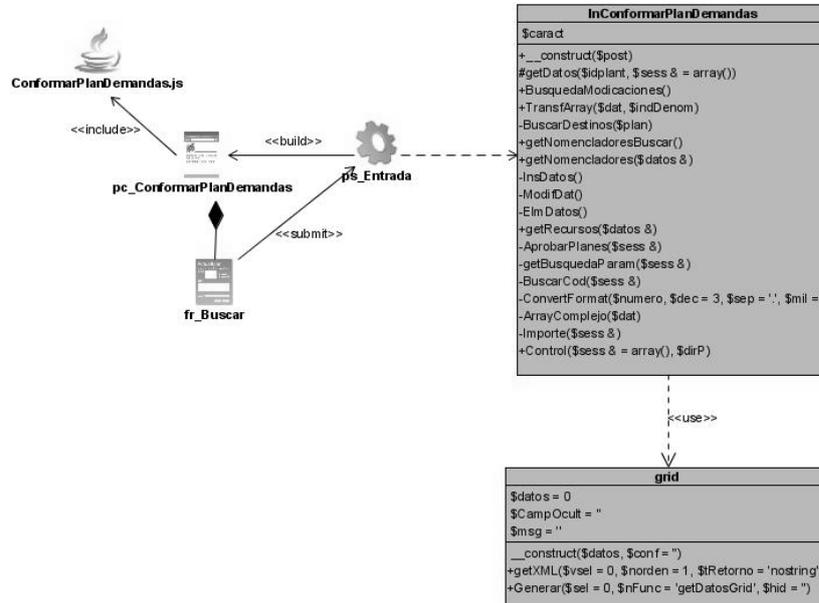


Figura 30. Diagrama de clases del análisis “Buscar Datos”.

3.8 Diseño de la Base de Datos.

Diagrama Entidad Relación de la BD.

Los diagramas o modelos entidad-relación son una herramienta para el modelado de datos de un sistema de información. Estos modelos expresan entidades relevantes para dicho sistema, sus interrelaciones y propiedades.

Además en él se modela el tratamiento de la información con carácter persistente dentro del sistema.



con la familia de fuentes Arial, Geneva, Helvetica, Sans-Serif, el tamaño de la misma no debe diferir mucho de 11px. Los colores con los que se trabajarán serán tonalidades claras basadas en el azul combinado con el color blanco o gris.

Gracias a los aspectos anteriormente se garantiza que el sistema sea agradable al usuario y muy fácil de usar, pues le permite adaptarse más fácilmente al área de trabajo que ante él se despliega.

3.11 Estándares de Programación.

Resulta ventajoso utilizar un estándar para escribir código, entre dichas ventajas, tenemos las siguientes:

1. Reducir errores
2. Escribir un código comprensible y fácil de leer
3. Garantizar una buena comunicación entre los programadores del equipo
4. Facilitar el mantenimiento del software

A continuación se muestra los principales estándares aplicados:

Tabla 19. Estándares de Código.

Legibilidad Objetivo: Nombrar las clases e instancias de las mismas de forma estándar para todas las aplicaciones.		
Apariencia de clases y objetos	Primera letra en mayúscula y serán en singular.	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 (). Los prefijos a utilizar serán los siguientes: t----- Clases Típicas In----- Lógica de Negocio. c-----Consulta



		me-----Manejadora de Entidad
Nombre de clases y objetos	Relacionados al propósito	El nombre empleado, debe permitir que con sólo leerlo se conozca el propósito de la clase o instancia de la misma. Para el caso de las instancias es recomendable que se denoten así: Para la clase: tNomumedida su instancia será \$Oumedida, de forma tal que la primera letra indique que es un objeto y el resto, la clase a la que pertenece.
Apariencia de atributos	Primera letra en minúscula	El nombre que se le da a los atributos de las clases debe comenzar con la primera letra en minúscula, en caso de que sea un nombre compuesto se empleará notación CamelCasing [16].
Nombre de atributos	Nemotécnicos	El nombre empleado, debe permitir que con sólo leerlo se conozca el propósito del mismo dentro de la clase. Ejemplo: \$nTabla, este atributo denota el nombre de una tabla.
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 [17]. 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	Agrupados por tipos primero los string, los	Los parámetros que se le pasan a las funciones se recomienda sean declarados de forma tal que estén



parámetro en funciones	numéricos y valores por defecto.	agrupados por el tipo de dato que contienen. Ejemplo: BuscaUnidad ($\$n$ Tabla(string), $\$n$ Campos(string), $\$k$ Indice (entero)).
Variables y constantes		
Apariencia de constantes	Todas sus letras en mayúscula	Se deben declarar las constantes con todas sus letras en mayúscula.
Apariencia de variables:	Primera letra en minúscula	El nombre que se le da a las variables debe comenzar con la primera letra en minúscula, en caso de que sea un nombre compuesto se empleará notación CamelCasing.
Nombres de las variables y constantes	Nemotécnicos	El nombre empleado, debe permitir que con sólo leerlo se conozca el propósito de la misma. Ejemplo: $\$n$ Campos.
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"); $\$n$ Tabla='nomproducto'; $\$k$ Indice=0;
Indentación		
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 espacios en	Function	Se dejarán dos espacios en blanco desde la izquierda



blanco desde la izquierda en	Define	en las instrucciones antes mencionadas.
2 espacios 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.
<p>Comentarios, separadores, líneas y espacios en blanco</p> <p>Objetivo: Establecer un modo común para comentar el código de forma tal que sea comprensible con sólo leerlo una vez.</p>		
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))



3.12 Tratamiento de Errores.

Los errores se tratan en los diferentes niveles desde la entrada de datos de los usuarios hasta los más complicados, que se pueden generar en la actualización de los datos.

Para prevenir errores por parte del usuario, a la hora de efectuar cualquier operación se deshabilitan ciertos botones si el usuario no tiene que utilizarlos en ese momento.

Mediante una combinación de validación en el lado del cliente y en el lado del servidor, se garantiza que los datos suministrados por los usuarios, se almacenen íntegros y no existan inconsistencias.

Algunos errores serán generados por funciones JavaScript para evitar la ejecución de la página en vano. Este es el caso de los formularios de inserción/actualización, y las eliminaciones, se utilizan los errores en forma de mensajes de texto, como alertas de JavaScript, en la misma página donde se ejecutó la acción, de forma que el usuario pueda corregir más fácilmente y continuar. En operaciones muy largas o complicadas, se permite volver atrás, para revisar o modificar la información. Y se utilizan mensajes de confirmación, para acciones que son irreversibles como es el caso de las eliminaciones y las aprobaciones.

3.13 Conclusiones.

En este capítulo se mostró como a través del análisis y diseño, se transformó el modelo de análisis en un modelo de diseño, es decir, en una estructura de clasificadores y realizaciones de casos de uso. Además se construyeron varios artefactos para llevar a cabo el proceso de implementación del sistema. También se identificaron otras funcionalidades que se deben tener en cuenta para futuras versiones del sistema. Se utilizaron diagramas de clases Web para explicar la lógica del negocio del sistema, se crearon mecanismos de diseño que simplifican el modelado y se modeló la base de datos de la aplicación. En este momento, ya se tiene confeccionada completamente la propuesta que trae este trabajo.



CAPÍTULO 4. IMPLEMENTACIÓN Y PRUEBA

4.1 Introducción.

En este capítulo se procede a la implementación del sistema y a la realización de pruebas del software para a través de las mismas obtener una revisión final de las especificaciones, del diseño y de la codificación.

Para el desarrollo de este capítulo se tuvieron en cuenta algunos artefactos que fueron generados en el flujo de análisis y diseño. A partir de los cuales se desarrolló la implementación del sistema en términos de componentes, es decir, ficheros de código fuente, scripts, ficheros de código binario, ejecutables y similares.

4.2 Modelo de Implementación.

El modelo de implementación constituye la vista de Implementación de la arquitectura, y como tal guía las labores de construcción del sistema. Este contiene fundamentalmente los subsistemas de implementación, incluyendo las dependencias y otras informaciones necesarias para su utilización.

Para lograr una mejor comprensión de los componentes que forman el sistema, se presentarán los diagramas de componentes que describen los elementos físicos y lógicos del sistema y sus relaciones.

Los componentes representan todos los tipos de elementos software que entran en la fabricación de aplicaciones informáticas. Pueden ser simples archivos, paquetes, bibliotecas cargadas dinámicamente. Las relaciones de dependencia se utilizan en los diagramas de componentes para indicar que un componente utiliza los servicios ofrecidos por otro componente, las mismas pueden ser de **import** entre subsistemas y de **compilación** entre paquetes de componentes o componentes.

El diagrama que se presenta a continuación tiene como objetivo figurar la estructura general de la aplicación en desarrollo, en términos de componentes, el mismo está estructurado por capas, representando así el estilo arquitectónico utilizado en el sistema. Los subsistemas que se incluyen en el mismo son agrupados por el criterio de funcionalidad y están compuestos por sus respectivos paquetes de componentes, dígame: Gestionar Plan (Actualizar Plan, Aprobar Plan, Buscar Datos), Gestionar Recuperaciones (Seleccionar Criterios de Búsqueda, Obtener Reportes), y Sistema Planificación. A



continuación se muestra en la Figura 32 el diagrama de componentes general y el de los paquetes de componentes que conforman los subsistemas se podrán observar en el Anexo 3.

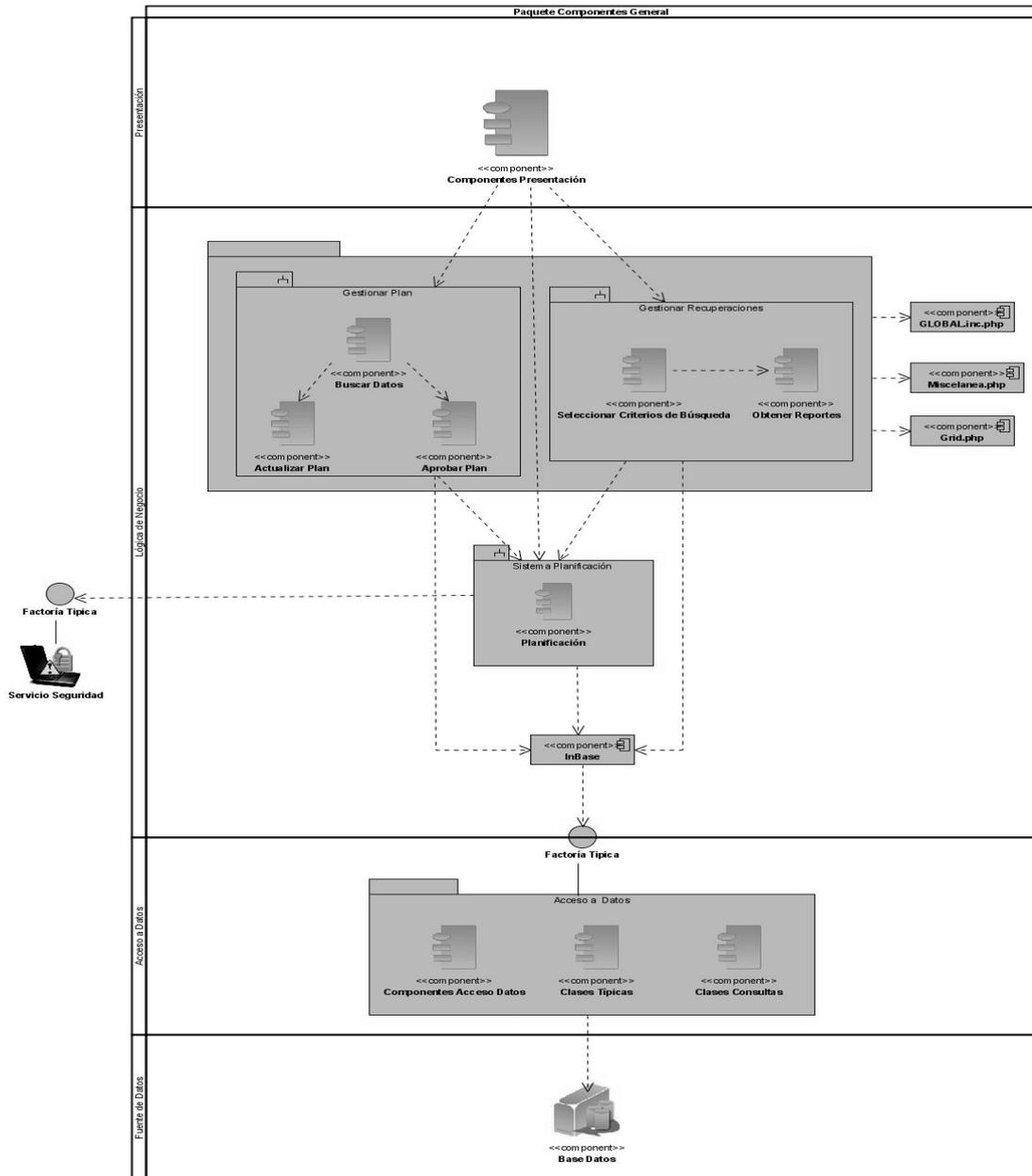


Figura 32. Diagrama de Componentes General.



4.3 Modelo de Despliegue.

El diagrama de despliegue representa la arquitectura de tiempo de ejecución de los procesadores, dispositivos y los componentes de software que se ejecutan en esa arquitectura. Es la última descripción física de la topología del sistema y describe la estructura de las unidades de hardware. Además, representa la disposición de las instancias de componentes de ejecución en instancias de nodos conectados por enlaces de comunicación.

Un nodo es un recurso de ejecución tal como un procesador, un dispositivo o memoria. En los procesadores es donde se encuentran alojados los componentes.

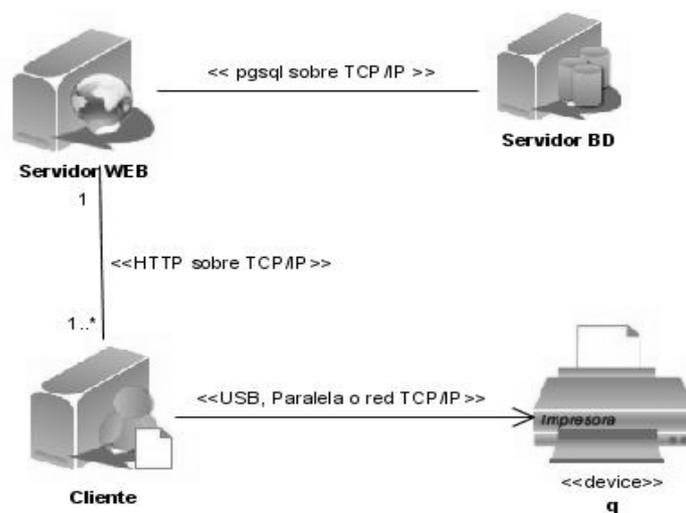


Figura 33. Diagrama de Despliegue.

El diagrama de despliegue mostrado anteriormente representa un modelo ideal que está en correspondencia con la arquitectura cliente-servidor en capas, compuesto por un Servidor Web, un Servidor de Base de Datos, una computadora para visualizar las operaciones sobre ambos servidores, un dispositivo que en este caso es una Impresora y las conexiones entre ellos.



4.4 Modelo de Prueba.

Las pruebas del software son un elemento crítico para la garantía de la calidad del software y representa una revisión final de las especificaciones, del diseño y de la codificación. Dentro de las pruebas se encuentran dos fundamentales: *prueba de caja negra* y *de caja blanca*.

La prueba de la **caja blanca** del software comprueba los caminos lógicos del software proponiendo casos de prueba que se ejerciten conjuntos específicos de condiciones. Se puede examinar el estado del programa en varios puntos para determinar si el estado real coinciden con el esperado o mencionado.

La prueba de **caja negra** se refiere a las pruebas que se llevan a cabo sobre la interfaz del software. O sea, a través de los casos de prueba se demuestra que las funciones del software son operativas, que la entrada se acepta de forma adecuada y que se produce un resultado correcto, así como que la integridad de la información externa se mantiene. La prueba de caja negra es la que se le aplicó al sistema. A continuación se muestran los casos de prueba.

Caso de Uso: Actualizar Plan de Demandas.

1. Seleccionar un Plan y modificar un Recurso.
2. Seleccionar un Plan y que el Recurso no pueda ser eliminado.
3. Seleccionar un Recurso y modificarlo.
4. Insertar un Recurso en el Plan satisfactoriamente.
5. No pueda insertar un Recurso.

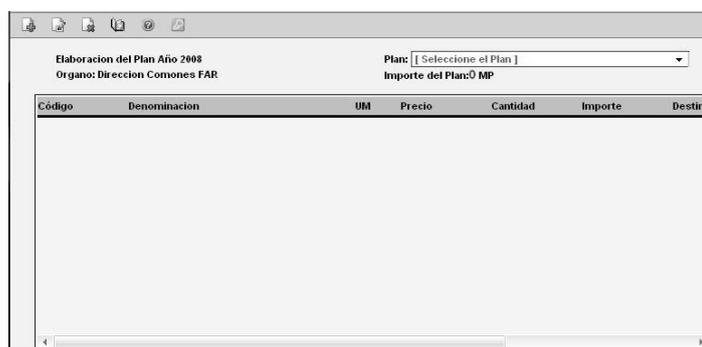


Figura 34. Caso de Prueba “Actualizar Plan de Demandas”

Tabla 20. Caso de prueba “Actualizar Plan de Demandas”.



Caso de Uso	Actualizar Plan de Demandas.		
Caso de Prueba	Entrada	Condiciones	Resultados
1	Plan seleccionado = "Aseguramiento Técnico". Recurso = "Flete y Seguros"	Como seleccionó el Plan y no seleccionó el recurso a modificar, no puede modificar ningún recurso.	El sistema le muestra un mensaje al usuario indicándole que debe seleccionar el recurso a modificar.
2	Plan seleccionado = "Asistencia Médica" Recurso = "Azúcar Crudo"	Como el recurso seleccionado ya se encuentra aprobado, no puede ser eliminado.	El sistema le muestra un mensaje al usuario que el recurso seleccionado no puede ser eliminado porque ya se encuentra aprobado
3	Plan seleccionado = "Plan Alimento" Recurso = "Torula"	Como el recurso seleccionado ya se encuentra aprobado, no puede ser modificado.	El sistema le muestra un mensaje al usuario que el recurso seleccionado no puede ser modificado porque ya se encuentra aprobado
4	Recurso a Insertar = "Aviones (INC 603/605)"	Como todos los criterios fueron seleccionados y el recurso no se encuentra aprobado, se pudo insertar satisfactoriamente.	El sistema le muestra un mensaje al usuario informándole que el recurso fue introducido satisfactoriamente.
	Recurso a Insertar = "Equipos Ópticos"	Como no seleccionó todos los criterios el recurso no puedo se	El sistema le muestra un mensaje indicándole



5		insertado.	al usuario que debe introducir los diferentes criterios de selección que no seleccionó.
---	--	------------	---

Caso de Uso: Aprobar Plan de Demandas.

1. Seleccionar el Plan y aprobarlo.

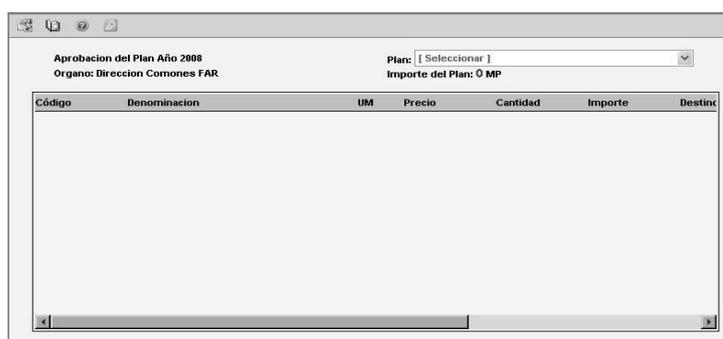


Figura 35. Caso de Prueba “Aprobar Plan”

Tabla 21. Caso de prueba “Aprobar Plan”.

Caso de Uso	Aprobar Plan de Demandas.		
Caso de Prueba	Entrada	Condiciones	Resultados
1	Plan seleccionado = “Inversiones Constructivas”	Como el plan seleccionado no presenta ninguna dificultad el plan puede ser aprobado.	El sistema le muestra un mensaje al usuario informando que el plan fue aprobado satisfactoriamente.

Caso de Uso: Mostrar reporte Plan Alimento.

1. Obtener el reporte por los criterios seleccionados.
2. Obtener el reporte con el paginado que fue especificado.



3. Introducir letras en el campo de “Líneas por hojas”.

PLAN DE ALIMENTOS AÑO 2008									
CÓDIGO DEL PRODUCTO	DESCRIPCIÓN	UM	PRECIO	TOTAL ALIMENTOS		UAM			
				CANTIDAD	VALOR	TOTAL		Fondos mercantiles	
						CANTIDAD	VALOR	CANTIDAD	VALOR
3294600082	PROD. QUIM AUX P/SIST.ORDENO	MP	1,000.000	100	100.000	100	100.000	100	100.000
9021001022	PPA P/FILTRO-HEPA	MP	1,000.000	127777777777	127,777,777,777.000	127777777777	127,777,777,777.000		
Total:					127,777,777,877.000		127,777,777,877.000		100.000

Figura 36. Caso de Prueba “Mostrar Reporte Plan de Alimento”

Tabla 22. Caso de Prueba “Mostrar Reporte Plan de Alimento”.

Caso de Uso	Mostrar Reporte Plan Alimento.		
Caso de Prueba	Entrada	Caso de Prueba	Resultados
1	Criterios seleccionados: Centro de Balance = DAAFAR. Órgano Consumidor = Dirección de Comones FAR. Tipo Gasto = Capitales Código = 6010000000	Como en la base de datos no se encuentra ningún dato que cumpla con los criterios especificados no se puede mostrar el reporte.	El sistema le muestra un mensaje al usuario informando que no se encontraron datos.
2	Reporte = Plan Vestuario Paginado = 3	Como el reporte a mostrar tiene más de tres tuplas de datos, devuelve el reporte paginado en segmentos de tres filas. Pudiendo devolver el último segmento con menos cantidad que la especificada	El sistema le muestra el reporte con el paginado especificado.



3	Líneas por hojas = "dos"	Como ese campo es numérico no permite que se introduzcan otros caracteres. Ejemplo letras.	El sistema no permite escribir caracteres que no sean números enteros.
---	--------------------------	--	--

Caso de Uso: Buscar Datos.

1. Buscar por diferentes criterios de búsqueda.
2. Buscar por un criterio de búsqueda

Figura 37. Caso de Prueba "Buscar Datos".

Tabla 23. Caso de prueba "Buscar Datos".

Caso de Uso	Buscar.		
Caso de Prueba	Entrada	Caso de Prueba	Resultados
1	Criterios de búsqueda: Centro Balance = Dpto. NMCC Código = 337982000	Como ese recurso no se encuentra en el plan que se está efectuando la búsqueda no se obtienen resultados.	El sistema le muestra un mensaje al usuario indicándole que no hay resultados para mostrar
2	Criterio de búsqueda: Plan = Azúcar, Miel y Alcoholes Centro de Balance = Unión Agropecuaria Militar	Como el Centro de Balance seleccionado durante la búsqueda balancea recursos del Plan especificado se obtiene el/los recursos del	El sistema le muestra los recursos del plan especificado que son balanceados por el Centro de Balance



		Plan que son balanceados por ese Centro de Balance.	seleccionado en la búsqueda.
--	--	---	------------------------------

4.5 Conclusiones.

En este capítulo se mostró como a través de la implementación, se produjo un refinamiento de la vista de la arquitectura del modelo de despliegue, donde los componentes ejecutables fueron asignados a nodos. Además cómo el modelo de implementación fue la entrada principal de las etapas de prueba que se realizan seguido de la implementación. Donde se verificó el resultado de esta probando cada construcción, incluyendo las versiones finales del sistema.

Se utilizaron diagramas de componentes para representar a través de un grafo los componentes de software unidos por medio de relaciones de dependencia; con los cuales se modeló la vista estática de un sistema. Además sirvieron para mostrar la organización y las dependencias lógicas entre un conjunto de componentes software. En este momento, ya se tiene el producto de software.



CONCLUSIONES

1. Para el desarrollo de este sistema se realizó una investigación de forma exhaustiva de los procesos relacionados con el funcionamiento interno de la institución para realizar la planificación.
2. Se tuvieron en cuenta los requerimientos obtenidos de los usuarios.
3. Se mantuvo la estructura interna del proceso de gestión de los planes de demandas de los Órganos Consumidores, así como sus niveles de jerarquía y asegurando un mejor flujo de la información de los órganos en red.
4. Se logró una especificación de los requisitos fundamentales para el diseño del Sistema.
5. Se definieron los casos de uso que satisfacen los requisitos especificados.
6. Se obtuvo un diseño de la base de datos.
7. Se documentó, de acuerdo con RUP, lo desarrollado.
8. Se implementó el sistema propuesto.
9. Se lograron mejores condiciones laborales del personal involucrado en el proceso, contribuyendo a su eficiente desarrollo.
10. Los efectos de este trabajo ya se evidencian en el MINFAR, pues la aplicación está siendo utilizada en el proceso de Planificación para el próximo año con resultados satisfactorios.



RECOMENDACIONES

Los objetivos de este trabajo han sido cumplidos, pero durante el desarrollo del mismo surgieron ideas que se recomiendan implementarse en el futuro, de forma que se logre una mayor eficiencia en la aplicación. A continuación se citan algunas de ellas:

1. Brindar la posibilidad al usuario de exportar los reportes a otro formato.
2. Incrementar la navegabilidad del sistema a través de la inserción de un botón en la interfaz de cada reporte, con la finalidad de hacer una nueva búsqueda en el mismo.
3. Reutilizar la metodología, tecnologías y herramientas empleadas en la construcción de la aplicación, para el desarrollo de futuras iteraciones de la misma.

**REFERENCIAS BIBLIOGRÁFICAS**

1. **de León, Maria Oralía.** ERP: su presencia en Latinoamérica. de Gerencia.com. *ERP: su presencia en Latinoamérica. de Gerencia.com.* [En línea] Meltom Technologies, 2001-2007. [Citado el: 11 de Enero de 2007.] http://www.degerencia.com/articulo/erp_su_presencia_en_latinoamerica.
2. **de la Torre, Anibal.** Lenguajes del lado servidor o cliente. *Lenguajes del lado servidor o cliente.* [En línea] 2006. [Citado el: 15 de Enero de 2007.] http://www.adelat.org/media/docum/nuke_publico/lenguajes_del_lado_servidor_o_cliente.html.
3. [En línea] 28 de Marzo de 2005. [Citado el: 15 de Abril de 2007.] <http://www.uberbin.net/archivos/internet/ajax-un-nuevo-acercamiento-a-aplicaciones-web.php>.
4. El lenguaje de Gestión de Base de Datos. *El lenguaje de Gestión de Base de Datos.* [En línea] [Citado el: 10 de Enero de 2007.] <http://www.utem.cl/web/lengsql.htm>.
5. Metodología de Desarrollo de Software (MSD). *Metodología de Desarrollo de Software (MSD).* [En línea] Reynox, Servicios Informáticos, 2005. [Citado el: 29 de Enero de 2007.] <http://reynox.com/sistemas/metodologia.php>.
6. Visual Paradigm for UML. *Visual Paradigm for UML.* [En línea] Visual Paradigm. [Citado el: 02 de Febrero de 2007.] <http://www.visual-paradigm.com/product/vpuml>.



BIBLIOGRAFÍA

1. **Teruel, Alejandro.** Arquitectura de Capas . *Arquitectura de Capas* . [En línea] 21 de Mayo de 2001. [Citado el: 03 de Mayo de 2007.] <http://www ldc.usb.ve/~teruel/ci3715/clases/arcapas.html>.
2. **Valdez Altamirano, Ing. Alfonso.** Comparativo de Entornos de Desarrollo Integrados(IDE's). *Comparativo de Entornos de Desarrollo Integrados(IDE's)*. [Online] [Cited: Enero 10, 2007.] <http://www.ubicuos.com/files/downloads/ComparativoIDES.pdf>.
3. **Jacobson, Ivan, Booch, Grady and Rumbaugh, James.** “*El Lenguaje Unificado de Modelado*”. 1999.
4. **Larman, Craig.** UML y Patrones Introducción al análisis y diseño orientado a objetos. *UML y Patrones Introducción al análisis y diseño orientado a objetos*. [Online] 1999. [Cited: Marzo 17, 2007.] http://alarcos.inf-cr.uclm.es/per/fgarcia/isoftware/doc/LabTr5_Multicapa.pdf.
5. **Garrigós Fernández, María Cristina.** Una herramienta para la enseñanza del modelado orientado a objetos. *Una herramienta para la enseñanza del modelado orientado a objetos*. [Online] 2004-2005. [Cited: Enero 10, 2007.] <http://vido.escet.urjc.es/textoo/Textoo-beta1.pdf>.
6. **Mendoza Sánchez, María.** [En línea] 07 de Julio de 2004. [Citado el: 27 de Enero de 2007.] http://www.informatizate.net/articulos/metodologias_de_desarrollo_de_software_07062004.html.
7. Herramientas CASE. *Herramientas CASE*. [Online] [Cited: Enero 18, 2007.] <http://www.cyta.com.ar/biblioteca/bddoc/bdlibros/proyectoinformatico/libro/c5/c5.htm>.
8. **Loyola Vera, Karol.** Ingeniería de Software. *Ingeniería de Software*. [Online] Noviembre 06, 2006. [Cited: Enero 29, 2007.] <http://www.mmug.cl/articulos.php?id=287&tod=1>.
9. **López, Carlos.** IMPACTO DE LOS ERP'S EN LAS EMPRESAS. *IMPACTO DE LOS ERP'S EN LAS EMPRESAS*. [Online] GestiónPolis. [Cited: Enero 11, 2007.] <http://www.gestiopolis.com/canales2/gerencia/erpbetty.htm>.
10. **Jacobson, Ivar, Booch, Grady and Rumbaugh, James.** *El Proceso Unificado de Desarrollo de Software. (Vol. I)*. La Habana : Félix Varela, 2004.



11. **Álvarez, Miguel Angel.** Lenguajes de lado servidor o cliente. *Lenguajes de lado servidor o cliente*. [Online] desarrolloweb.com. [Cited: Diciembre 22, 2006.] <http://www.desarrolloweb.com/articulos/239.php>.
12. **Lagos Torres, Manuel.** Introducción al diseño con patrones. *Introducción al diseño con patrones*. [Online] [Cited: Marzo 15, 2007.] <http://www.elrincondelprogramador.com/default.asp?pag=articulos/leer.asp&id=29>.
13. **Figueroa, Pablo.** Introducción a Patrones. *Introducción a Patrones*. [Online] [Cited: Marzo 15, 2007.] http://agamenon.uniandes.edu.co/~pfiguero/soo/Magister_Patrones/intropatrones.html.
14. **Eriksson, Hans-Erik and Penker, Magnus.** *Business Modeling with UML: Business Patterns at Work*. United States of America : s.n., 2000.
15. PHPCODE. *PHPCODE*. [Online] [Cited: Abril 15, 2007.] <http://www.phpcode.es>.
16. **Pressman, Roger S.** *Ingeniería del Software . Un Enfoque Práctico (Edición 5)*.
17. **Adolph, Steve, Bramble, Paul and Cockburn, Alistair.** *Patterns for Effective Use Cases*. s.l. : Cockburn-Highsmith Series, 2001.
18. **de la Torre, Aníbal.** Lenguajes del lado servidor o cliente. *Lenguajes del lado servidor o cliente*. [Online] [Cited: Enero 15, 2007.] http://www.adelat.org/media/docum/nuke_publico/lenguajes_del_lado_servidor_o_cliente.html.
19. Ventajas de PostgreSQL. *Ventajas de PostgreSQL*. [Online] TiendaLinux.com. [Cited: Enero 20, 2007.] http://soporte.tiendalinux.com/portal/Portfolio/postgresql_ventajas_html.
20. **Hernandis, José Alberto.** Visual Paradigm for UML. *Visual Paradigm for UML*. [Online] Versioncero, Julio 04, 2005. [Cited: Enero 20, 2007.] <http://www.versioncero.com/noticia/210/visual-paradigm-for-uml>.
21. **Tramullas y Kronos, Jesús.** Introducción a la Documática. *Introducción a la Documática*. [En línea] 1997-2000. [Citado el: 10 de Diciembre de 2006.] <http://tramullas.com/documatica/2-4.html>.



GLOSARIO DE TÉRMINOS

1. **Apache:** Servidor web de distribución libre. Fue desarrollado en 1995 y ha llegado a ser el más usado de Internet. Soporta los lenguajes PERL y PHP.
2. **Asíncrona:** referencia al suceso que no tiene lugar en total correspondencia temporal con otro suceso. Por ejemplo podemos hablar de motor asíncrono a aquel cuya velocidad de rotación no corresponde con la frecuencia de corriente alterna que lo hace funcionar
3. **CASE:** Acrónimo inglés de Computer Aided Software Engineering, que viene a significar Ingeniería de Software Asistida por Ordenador.
4. **CSS:** (Cascade Style Sheet) Hoja de Estilo en Cascada. Conjunto de instrucciones escritas en HTML que definen las apariencias de una página web con el objetivo de que sus estilos se parezcan.
5. **DOM:** El DOM o Document Object Model (Modelo de Objetos de Documento) es una forma de representar documentos estructurados (tales como una página web HTML o un documento XML) que es independiente de cualquier lenguaje orientado a objetos.
6. **ERP:** *Enterprise Resource Planning*, sistema de información gerencial que integra muchas de las prácticas de los negocios asociados con las operaciones de producción.
7. **HTML:** Acrónimo inglés de Hyper Text Markup Language (lenguaje de marcación de hipertexto), es un lenguaje de marcas diseñado para estructurar textos y presentarlos en forma de hipertexto, que es el formato estándar de las páginas web. Este lenguaje se basa en tags (instrucciones que le dicen al texto como deben mostrarse) y atributos (parámetros que dan valor al tag). Es el estándar usado en el World Wide Web.
8. **HTTP:** Es el protocolo de la Web (WWW), usado en cada transacción. Las letras significan Hyper Text Transfer Protocol, es decir, protocolo de transferencia de hipertexto. El hipertexto es el contenido de las páginas web, y el protocolo de transferencia es el sistema mediante el cual se envían las peticiones de acceder a una página web, y la respuesta de esa web, remitiendo la información que se verá en pantalla.
9. **IDE:** Siglas de: Integrated Drive Electronics ó Integrated development environment, es decir, un entorno integrado de desarrollo.



10. **Internet:** Es una red de redes a escala mundial de millones de computadoras interconectadas con el conjunto de protocolos TCP/IP. También se usa este nombre como sustantivo común y por tanto en minúsculas para designar a cualquier red de redes que use las mismas tecnologías que la Internet, independientemente de su extensión o de que sea pública o privada.
11. **Intranet:** Una intranet es una red de ordenadores de una red de área local.
12. **LAN:** es la abreviatura de **Local Area Network** (Red de Área Local o simplemente Red Local). Una red local es la interconexión de varios ordenadores y periféricos. Su extensión esta limitada físicamente a un edificio o a un entorno de unos pocos kilómetros.
13. **Lenguajes Interpretados:** Los lenguajes de programación, que incluye a aquellos lenguajes cuyos programas son habitualmente lenguajes interpretados (o lenguajes de script) forman un subconjunto de los ejecutados en un intérprete en vez de compilados. Sin embargo, la definición de un lenguaje de programación es independiente de cómo se ejecuten los programas en él escritos, ya sea mediante una compilación previa o a través de un intérprete.
14. **Lógica de negocio:** Se denomina capa de negocio (e incluso de lógica del negocio) pues es aquí donde se establecen todas las reglas que deben cumplirse. Recibiendo las peticiones del usuario y enviando las respuestas tras el proceso.
15. **MVCC:** Método concurrente utilizado en las bases de datos relacionales.
16. **Notación CamelCasing:** Los identificadores y nombres de variables, métodos y funciones están compuestos por múltiples palabras juntas iniciando cada palabra con letra mayúscula excepto la primera palabra que debe iniciar con minúscula. Ejemplo: notacionCamelCasing.
17. **Notación PascalCasing:** Los identificadores y nombres de variables, métodos y funciones están compuestos por múltiples palabras juntas iniciando cada palabra con letra mayúscula. Ejemplo: NotacionPascalCasing.
18. **PDF:** Es la Extensión que corresponde a un tipo de fichero (un libro electrónico) creado con Adobe Acrobat.
19. **PostgreSQL:** Es un servidor de base de datos relacional libre, liberado bajo la licencia BSD.
20. **Programación Orientada A Objetos:** La Programación Orientada a Objetos (POO u OOP según siglas en inglés) es un paradigma de programación que define los programas en términos de "clases de objetos", objetos que son entidades que



combinan estado (es decir, datos), comportamiento (esto es, procedimientos o métodos) e identidad (propiedad del objeto que lo diferencia del resto). La programación orientada a objetos expresa un programa como un conjunto de estos objetos, que colaboran entre ellos para realizar tareas. Esto permite hacer los programas y módulos más fáciles de escribir, mantener y reutilizar.

21. **RUP:** El Proceso Racional Unificado o RUP (Rational Unified Process), es un proceso de desarrollo de software.
22. **SI:** es un conjunto de componentes interrelacionados que reúne (u obtiene), procesa, almacena y distribuye información para apoyar la toma de decisiones y el control en una organización.
23. **UML:** Lenguaje Unificado de Modelado (UML, por sus siglas en inglés, Unified Modelling Language) es el lenguaje de modelado de sistemas de software más conocido en la actualidad; aún cuando todavía no es un estándar oficial, está apoyado en gran manera por la OMG (Grupo dedicado a la promoción de la tecnología orientada a objetos y su estandarización).
24. **XHTML:** Acrónimo inglés de eXtensible Hyper Text Markup Language (lenguaje extensible de marcado de hipertexto), es el lenguaje de marcado pensado para sustituir a HTML como estándar para las páginas web. XHTML es la versión XML de HTML, por lo que tiene, básicamente, las mismas funcionalidades, pero cumple las especificaciones, más estrictas, de XML.
25. **XML:** Acrónimo de "EXtensible Markup Language". Es decir, lenguaje de marcas extensible, es de reciente creación (febrero de 1998). XML es un metalenguaje, o sea, sirve para crear lenguajes. Fue diseñado para permitir la descripción de información contenida en el WWW a través de estándares y formatos comunes, de manera que tanto los usuarios de Internet como programas específicos (agentes), puedan buscar, comparar y compartir información en la red.
26. **XMLHttpRequest (XHR):** Es un conjunto de APIs que pueden ser usadas por JavaScript y otros lenguajes para transferir XML u otra información textual hacia o desde un servidor web usando HTTP, mediante el establecimiento de un canal de comunicación independiente entre páginas web del lado del cliente y del lado del servidor. El mayor avance de XMLHTTP es la habilidad de actualizar dinámicamente una página web sin recargar la página entera o usando plugins. Es usado por diversos sitios web para implementar aplicaciones web dinámicas.