

Universidad de las Ciencias Informáticas

Facultad 4



Título: Sistema Informático para la gestión de la planificación de los
Centros de Balance.

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

Autoras: Marianela Tenrero Cabrera
Giselle Márquez Gimeno

Tutor: Ing. Alain Fernández Deronceré

CIUDAD DE LA HABANA, JUNIO DEL 2007

*Por el camino de la sabiduría te he encaminado,
Y por veredas derechas te he hecho andar.
Cuando anduvieres, no se estrecharán tus pasos,
Y si corrieres, no tropezarás.*



Proverbios 4:11-12



DECLARACIÓN DE AUTORÍA

DECLARACIÓN DE AUTORÍA

Declaramos que somos las únicas autoras de este trabajo y autorizamos al Ministerio de las Fuerzas Armadas Revolucionarias (MINFAR) y a la Universidad de las Ciencias Informáticas (UCI) para que hagan el uso que estimen pertinente con este trabajo.

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

Firma del Autor

Firma del Autor

Firma del Tutor



AGRADECIMIENTOS

De Giselle:

A diana por ser mi mejor amiga y haberme apoyado en las buenas y en las malas

A Maylin, Maria y Eugenia por su preocupación y ayuda incondicional a la hora de efectuar este trabajo.

Y a mi compañera de tesis Marianela, que más compañera, más bien fue una amiga, consejera y doctora.



AGRADECIMIENTOS

De Nela:

Le doy gracias a Dios por haberme concedido entrar a esta Universidad, y permitir que me mantuviera en ella acompañándome y guiándome hasta este momento.

A Arcel por estar a mi lado todos estos años, por su amor, paciencia, dedicación y no perder la confianza en mí.

A mis tíos Raully, Chichi, Paco, Gonzalo, por preocuparse siempre, por ayudarme en todo lo que han podido y hasta en las cosas que no han podido también, por sus oraciones y por considerarme una hija. A mis otros tíos y abuelos Leo, Lesvia, Onairis, Franco, Raúl, Xiomara, Guillermo, por siempre preocuparse por todos mis estudios, por sus consejos, aunque ahora estén muy lejos siempre me tienen presente y yo los llevo a todos en el corazón. A mis hermanas, aunque no lo son de sangre, Gresia, Angélica, Pilar, Nany, Cholo, por toda su preocupación todos estos años. A Papito, Mami Yeya, Maday, mis primos.

¡¡A toda mi familia, por quererme y orar tanto por mí!!

A mis amigos: Arodys, Karenia, Osdalme, Bazán, Dayana por estar cuando los he necesitado, en los buenos momentos y en los difíciles también. A mi amiga Yisel Pupo, por tanto apoyo, preocupación. A mis compañeros, Luis Manuel, Ilieva, Cristina, Osmar, Ivette, Ceállys, Jiselle, Damir, gracias a su ayuda pudimos terminar este trabajo con calidad. A Sarmiento por su preocupación y estar al tanto de este trabajo.

A Mercedes por ser un gran apoyo espiritual. A Gino y Teresa por su preocupación. Martha, Jorge, Robertico por su acogida y hacer con su ayuda más fácil todo este tiempo que he estado tan lejos de mi casa.

A nuestro tutor Alain, por su apoyo y darnos ánimo. A Rolando, Saumel, Yoandro por haber colaborado de una forma u otra al desarrollo de este trabajo.



DEDICATORIA

Les dedico este trabajo a mis padres por haberme apoyado en todos estos años de carrera, por luchar junto a mí por más que una carrera, por un futuro, sacrificando todo lo que tuvieran a su alcance.

Los quiero,

Su Yei.



DEDICATORIA

A mi mamá María Elena Cabrera Aria que es la mujer más grande y maravillosa que conozco, por ser el principal motivo por el cual cada mañana me levanto con deseos de ser mejor, por darme tanta tranquilidad, confianza y estabilidad.

A mi papá Carlos Andrés Estrada Romero, sé que no se conforma con cada logro que yo pueda alcanzar, sino que siempre está esperando mucho más de mí y eso me obliga a esforzarme y creer que siempre se puede un poco más.

A mi abuelita linda Adis Nelba Aria Labrada, mi otra mamá, por siempre estar luchando a mi lado, alentándome y aconsejándome para que alcance todas mis metas, es para mí un gran ejemplo de paciencia amor y dedicación.

A mi hermano Lázaro Carlos Estrada Cabrera, se que siempre está esperando lo mejor de mí, me apoya, se alegra con las cosas que me alegran, y sufre con las cosas que me angustian, por su preocupación y estar al tanto de cada una de mis situaciones y tenerme siempre presente aunque me encuentre muy lejos de él.

Los quiero mucho,

Nela



RESUMEN

El objetivo de este trabajo es diseñar una aplicación Web que cumpla con los requisitos necesarios para lograr un adecuado manejo de información en la planificación material y financiera de los Centros de Balance de las Fuerzas Armadas Revolucionarias.

Esta aplicación está diseñada basándose en un sistema ERP a la medida, a través de la cual se pueden realizar todas las operaciones y análisis necesarios para lograr un buen funcionamiento dentro de los Centros de Balance.

Este trabajo está encaminado, inicialmente, a la investigación de las herramientas y tecnologías que se emplean en la modelación del negocio y del sistema, para así desarrollar los requisitos funcionales que cumple la aplicación, así como también se utilizan en la realización del análisis y diseño, con el respectivo refinamiento de los diagramas de clases y las relaciones entre ellos, aplicando los patrones de diseño como un medio para facilitar la reutilización de buenos diseños y arquitecturas.



TABLA DE CONTENIDOS

AGRADECIMIENTOS	I
DEDICATORIA	III
RESUMEN	V
INTRODUCCIÓN	VI
CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA	7
1.1 INTRODUCCIÓN	7
1.2 ESTADO DEL ARTE	7
1.3 ¿POR QUÉ UNA WEB?	9
1.4 SITIOS WEB Y APLICACIONES WEB	9
1.5 PÁGINAS DINÁMICAS EN SERVIDOR.	10
1.6 LENGUAJE DEL LADO DEL CLIENTE.	11
1.6.1 JavaScript	11
1.6.2 HTML	12
1.7 LENGUAJES QUE CORREN EN EL SERVIDOR.	13
1.7.1 PHP	13
1.8 TECNOLOGÍA AJAX.	16
1.9 SISTEMAS DE GESTIÓN DE BASES DE DATOS.	20
1.9.1 PostgreSQL	20
1.10 HERRAMIENTAS CASE	21
1.10.1 Visual Paradigm.	22
1.11 CONCLUSIONES.	25
CAPÍTULO 2: MODELACIÓN DE LA PRPUESTA DE SOLUCIÓN	26
2.1 INTRODUCCIÓN.	26
2.2 MODELADO DEL NEGOCIO	26
2.3 ACTOR Y TRABAJADOR DEL NEGOCIO	27
2.4 DIAGRAMA DE CASOS DE USO DEL NEGOCIO (DCUN)	27
2.5 DESCRIPCIONES TEXTUALES	28



TABLA DE CONTENIDOS

2.6 DIAGRAMAS DE ACTIVIDADES.....	31
2.7 CLASES DEL MODELO DE OBJETOS.....	32
2.8 ¿QUÉ SON LOS REQUISITOS?	33
2.9 REQUISITOS FUNCIONALES.....	33
2.10 REQUISITOS NO FUNCIONALES.....	34
2.11 MODELO DEL SISTEMA.....	37
2.12 DIAGRAMA DE CASOS DE USO DEL SISTEMA.....	37
2.13 DESCRIPCIÓN DE CASOS DE USO DEL SISTEMA.....	39
2.14 CONCLUSIONES.....	58
CAPÍTULO 3: ANÁLISIS Y DISEÑO DEL SISTEMA	59
3.1 INTRODUCCIÓN.....	59
3.2 MODELO DE ANÁLISIS.....	59
3.3 DIAGRAMAS DEL ANÁLISIS.....	60
3.4 DIAGRAMAS DE SECUENCIA.....	62
3.5 MECANISMOS DE DISEÑO.....	66
3.5.1 Mecanismo de Diseño de Acceso a Datos	66
3.5.2 Mecanismo de Diseño de Seguridad	69
3.6 ESTÁNDARES DE DISEÑO.....	70
3.7 MODELO DE DISEÑO.....	70
3.8 DIAGRAMAS DEL DISEÑO.....	71
3.9 ARQUITECTURA.....	74
3.9.1 ¿Qué son los patrones?.....	74
3.10 DISEÑO DE LA BASE DE DATOS.....	78
3.11 DIAGRAMAS O MODELOS DE ENTIDAD RELACIÓN.....	78
3.12 DESCRIPCIÓN DE LAS TABLAS DE LA BASE DE DATOS.....	79
3.13 CONCLUSIONES.....	84
CAPÍTULO 4: IMPLEMENTACIÓN.....	85
4.1 INTRODUCCIÓN.....	85
4.2 DIAGRAMAS DE IMPLEMENTACIÓN.....	85
4.2.1 Diagrama de Despliegue.....	85



TABLA DE CONTENIDOS

4.2.2 Diagrama de Componentes	86
4.3 CONCLUSIONES	90
CONCLUSIONES	91
RECOMENDACIONES	92
REFERENCIAS BIBLIOGRÁFICAS	93
BIBLIOGRAFÍA	95
GLOSARIO	96

INTRODUCCIÓN

Internet es una red mundial de computadoras interconectadas con un conjunto de protocolos, el más destacado, el TCP/IP¹. Al contrario de lo que se piensa comúnmente, Internet no es sinónimo de World Wide Web². Esta es parte de Internet, siendo la World Wide Web uno de los muchos servicios ofertados en la red Internet. La Web es un sistema de información mucho más reciente (1995) que emplea Internet como medio de transmisión. [1]

El lanzamiento al espacio del primer satélite soviético, el Sputnik, en octubre de 1957 puede tomarse como un punto de referencia seguro para el comienzo de la historia de Internet. La puesta en órbita terrestre de aquella pequeña bola de metal galvanizó a los Estados Unidos. La precisión que requería el envío al espacio de un cohete y su transformación en un satélite artificial de la Tierra era superior a la que exigía el lanzamiento de una bomba nuclear a un objetivo. Aquella conmoción produjo diferentes resultados, que fueron desde los subsidios para desarrollar formas efectivas de entrenamiento de los científicos hasta la creación de una nueva agencia de defensa, la Advanced Research Project Agency (ARPA³ - siglas en inglés).

En 1961, el sabotaje a tres torres de microondas en Utah renovó, aunque a menor escala, la sensación de pánico que produjo el Sputnik unos años antes. La preocupación sobre la confiabilidad de los mecanismos militares de "comando y control", así como el temor a nuevos ataques terroristas que pudieran dejar al país sin medios de comunicación estimularon la búsqueda de infraestructuras más resistentes. El último fin de aquella búsqueda fue el diseño de un sistema de comunicación capaz de sobrevivir a una guerra nuclear.

Es precisamente en el año 1962 que surge el proyecto de Internet, producto del interés de los Estados Unidos por crear una red de militar capaz de soportar las comunicaciones de esta esfera bajo las condiciones de un ataque nuclear procedente de la entonces Unión Soviética y otros países del campo socialista.

Entre 1962 y 1964, Paul Baran de la Rand Corporation, elaboró una serie de informes en los que sugirió la creación de una estructura de comunicación en forma de tela de araña con el fin de impedir la selección de objetivos para cualquier forma de ataque. Para este proyecto, se empleó el protocolo de Internet. Probablemente era la única tecnología capaz de sobrevivir a un ataque nuclear. [2]

Para 1966/67 ARPA tenía suficiente material recopilado para publicar un plan para crear una red de computadoras que se llamaría ARPANET⁴. [3]

El objetivo original de ARPANET era ofrecer acceso remoto a computadoras distantes para compartir tanto ficheros remotos como las experiencias de investigación en materia de computación. [4]

El 2 de septiembre de 1969 se monta en la Universidad de California en Los Ángeles (UCLA) el primero de cuatro nodos, de ARPANET, llamada así en honor a su patrocinador. Los siguientes nodos fueron el Instituto De Investigación De Stanford "SRI) el 1 de octubre, la Universidad de California Santa Bárbara (UCSB) el 1 de noviembre y la Universidad de Utah, en diciembre.

El plan era sin precedente: Kleinrock, un profesor pionero de la informática en UCLA, y su grupo pequeño de estudiantes graduados entre los que se encontraba Charley Kline, esperaban lograr conectarse remotamente a la computadora de Stanford para intentar enviar algún dato. Comenzarían intentando lograr "firmarse" en la computadora en Stanford, éstos serían los primeros paquetes en ser enviados.

De cualquier manera una revolución había comenzado, esta permitió que los científicos compartieran información y recursos a través de largas distancias. ARPANET pronto se convirtió en un popular sistema de comunicación y de colaboración a distancia y creció, su estructura descentralizada y su capacidad para incorporar muchos tipos diferentes de computadoras, hicieron fácil su expansión. [3]

Con este progreso en el manejo de la información provoca un desarrollo creciente la industria del hardware unido con el desarrollo de la industria del software, en esta última el cambio de conceptos, paradigmas, y fundamentalmente la necesidad de relacionar a nuevos métodos y formas de comunicación e intercambio en general sin importar la distancia.

Con el surgimiento de Internet se dio lugar a que los medios para publicar y compartir información se evolucionaran hasta el punto de ser un factor importante en la economía mundial, por todos los servicios que ofrece.

Resulta difícil resumir en pocas palabras cuáles pueden ser todas las utilidades de Internet: puede servir para cualquier tarea que consista en intercambiar información. Permite realizar actividades como leer el periódico del día, comprar casi cualquier producto, jugar diferentes juegos con personas de todo el mundo, charlar con ellos (chatear), trabajar desde el propio domicilio, y muchas cosas más.

Para los universitarios, desde el punto de vista práctico, Internet puede ser una fuente de información: cada vez más universidades colocan información en Internet disponible para todo el mundo. En el ámbito de la investigación, se puede conectar con las universidades de primer nivel y obtener información sobre sus últimas investigaciones, conseguir artículos, consultar bases de datos bibliográficas para localizar referencias, etc. En el plano de la docencia, hay muchas universidades en la que se dispone de información sobre las asignaturas que se imparten: temarios, apuntes, información de referencia.... De este modo, se puede buscar en cualquier parte del mundo información. Además Internet es una importante fuente de información sobre organismos oficiales, finanzas, estadísticas, etc. [5]

Hoy en día, las empresas usan generalmente la Internet como medio para lograr que el uso compartido de la información sea más efectivo. La utilización de las aplicaciones Web, logran una mayor rentabilidad y eficiencia, pues gracias a la interconexión que permite la red, se logran optimizar esfuerzos, recursos, tiempo y dinero, así como la eliminación de las barreras de las distancias. Pero la forma de representar esa información, así de cómo, comunicárselo a la empresa es un paradigma hoy en día, debido a que con los rápidos avances de las tecnologías de información dejan viejas reglas de competencia y el duradero entendimiento de la relación cliente-proveedor obsoleto, por lo que se va creando la necesidad de un entorno de producción dinámico en el cual los productos y procesos pueden cambiar semanalmente y las agendas de la producción pueden cambiar diariamente.

Una de las muchas herramientas que existen para la ayuda en la toma de decisiones son los sistemas ERP⁵ (Enterprise Resource Planning, Planeación de Recursos Empresariales). Estos sistemas están diseñados para modelar y automatizar muchos de los procesos básicos, con el objetivo de integrar información a través de la empresa.

Ramesh (1998) citado por Alejandra Recio (1998) define un *ERP* como una "solución de software que trata las necesidades de la empresa tomando el punto de vista de proceso de la organización para alcanzar sus objetivos integrando todas las funciones de la misma". Recio menciona además que un sistema *ERP* facilita la integración de los sistemas de información de la empresa, ya que cubre todas las

áreas funcionales. Los sistemas que integra son bases de datos, aplicaciones, interfaces, herramientas y el rediseño en el proceso del negocio. Estos sistemas buscan como objetivos principales la optimización de los procesos empresariales, el acceso a información confiable, precisa y oportuna, la posibilidad de compartir información entre todos los componentes de la organización, la eliminación de datos y operaciones innecesarias y la reducción de tiempos y de los costes de los procesos.

Su **propósito** fundamental es otorgar apoyo a los clientes del negocio, tiempos de respuesta rápidos a sus problemas así como un eficiente manejo de información que permita la toma oportuna de decisiones y disminución de los costos totales de operación. Estos son sistemas integrales, modulares y adaptables.

Integrales, porque permiten controlar los diferentes procesos de la compañía entendiendo que todos los departamentos de una empresa se relacionan entre sí.

Modulares, porque en una empresa existen un conjunto de departamentos que se encuentran interrelacionados por la información que comparten y que se genera a partir de sus procesos. Por ejemplo: Materiales, Finanzas, logística, entre otros departamentos y Adaptables, porque están creados para adaptarse a la idiosincrasia de cada empresa. Esto se logra por medio de la configuración de los procesos de acuerdo con las salidas que se necesiten de cada uno.

Una vez que se haya implementado un sistema *ERP* deben haber logrado una base de datos centralizada, los componentes del *ERP* interactúan entre sí consolidando todas las operaciones y los datos del mismo se ingresan sólo una vez, dando consistencia y durabilidad.

El mundo empresarial de hoy no puede consistir sino se traza objetivos y metas que logren satisfacer las necesidades y demandas de una empresa, donde se efectúan paso a paso como un algoritmo, en otras palabras, **planificación**, no es más que un proceso racional de previsión, estructuración, diseño y asignación óptima de recursos de las organizaciones, para que alcancen resultados en un tiempo y espacio.

La Planificación Material y Financiera en el Ministerio de la Fuerzas Armadas Revolucionarias (MINFAR), se realiza para cumplir los objetivos trazados para el año de preparación para la defensa.

Esta Planificación se inicia en los Órganos Consumidores (**OC**), que puede ser un Órgano Consumidor Autorizado (**OCA**) o un Órgano Abastecedor Central (**OAC**), el mismo elabora su Plan de Demandas y lo envía a los respectivos Centros de Balances (**CB**). Una vez que el CB tiene en su poder el Plan de

Demandas, lo revisa, balancea lo solicitado y lo aprueba. Finalmente el Plan es enviado a la Dirección de Economía **(DE)** para su consolidación y presentación a los Organismos de la Administración Central del Estado **(OACE)**.

Sin embargo este trabajo crea grandes deficiencias en el proceso y es producto al poco uso de las tecnologías, los trámites se realizan de diferentes formas, ya sea manual o automatizadamente; en las tareas propias de cada órgano se emplean diversos formatos como las herramientas ofimáticas y otros; por lo que es muy difícil eliminar los archipiélagos de datos y lograr que la información esté disponible en todo momento con la objetividad y veracidad que se requiere, manifestándose constantemente la ineficacia para el cumplimiento con calidad de las tareas.

Es por ello que esta actividad no se ve ajena a la implantación de un sistema *ERP*, como un sistema que ayude a la toma de decisiones por parte de los directivos, debido a que constituye la base o fase inicial del ciclo logístico del MINFAR, donde se ven implicados diferentes órganos.

Por todo lo expresado anteriormente, se comprende que se está enfrentando a una **situación problemática** y es que en el Ministerio de las Fuerzas Armadas Revolucionarias el sistema de planificación material y financiera no está automatizado y todos los procesos se hacen manuales o por sistemas no profesionales, los datos recogidos son archivados en papel duro. Todas estas situaciones conllevan a que el nivel de errores de los cálculos sea considerable y las condiciones laborales no sean las ideales. Las tecnologías han ido avanzando y el sistema manual con el tiempo ha quedado obsoleto.

Por tal motivo el **problema científico** nos lleva a la siguiente interrogante: ¿Cómo llevar la informatización a la conformación de los planes de recursos materiales y financieros de los Centros de Balances, para eliminar así las barreras de distancia existentes entre estos y los Órganos Consumidores, con el mínimo costo de tiempo posible en la toma de decisiones?

Este problema se enmarca en el **objeto de investigación**: proceso de la Planificación Material y Financiera de los Centros de Balances del MINFAR.

El **objetivo general** o el **objetivo de investigación** es diseñar una aplicación Web que contribuya a eliminar las barreras de la distancia y una rápida toma de decisiones en la planificación material y financiera de los Centros de Balance.

Objetivos Específicos:

- ✓ Realizar el análisis y diseño del sistema utilizando una metodología de ingeniería de software.
- ✓ La informatización de las actividades vinculadas a la manipulación, y recuperación de la información en los Centros de Balances.

El **campo de acción** es el proceso de informatización vinculado a la toma de decisiones del MINFAR.

Para desarrollar la investigación se plantea la siguiente **hipótesis**: si se realiza una aplicación Web, 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, agilizará el proceso de planificación del MINFAR, favoreciendo una rápida toma de decisiones.

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

- ✓ Profundizar en el funcionamiento del proceso de planificación material y financiera específicamente lo que ocurre en los CB del MINFAR.
- ✓ Analizar el estado de la informatización dentro de las áreas vinculadas al proceso antes mencionado.
- ✓ Proponer las tecnologías y tendencias actuales a utilizar en el desarrollo de la aplicación.
- ✓ Analizar los procesos que se desarrollan dentro del sistema.
- ✓ Desarrollar el análisis y diseño del sistema para la conformación de los planes de recursos materiales y financieros de los Centros de Balances y su aprobación.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

1.1 Introducción

En este capítulo se abordarán algunos conceptos necesarios e importantes para comprender en qué consiste el funcionamiento y desarrollo de una aplicación Web; haciendo referencia básicamente a las diferentes herramientas que serán utilizadas para el desarrollo de la aplicación, así como su importancia y las diferentes prestaciones que estas ofrecen.

1.2 Estado del Arte

El llevar a cabo la administración de empresa desde sus inicios, es una tarea sumamente agotadora e importante, para su buen funcionamiento y desarrollo, estas tareas serán los cimientos de una posible gran organización en un futuro, siempre y cuando las actividades que se realicen ofrezcan una visión y fortalezas necesarias para lograrlo. Una de las herramientas con mayor potencial para lograr obtener fuertes bases para una empresa, es su información y su adecuado manejo.

La información es de suma importancia para cualquier organización, un sistema de información sería una herramienta muy útil para todas las empresas, con su aplicación se lograría obtener un mejor manejo de la toda la información que se genere para poder utilizarla cuando se crea necesario. Los sistemas de información cumplen tres objetivos básicos dentro de las organizaciones: automatizar procesos, proporcionar información que sirva de apoyo para la toma de decisiones y lograr ventajas competitivas a través de su implantación y uso.

Además, un sistema de información permite tener un control sobre todos los elementos de la empresa y todos los recursos con los que cuenta, así se evitarían desperdicios o pérdidas de recursos materiales y demás, y sobre el inventario con el que se cuenta y así saber en todo momento qué es lo que se tiene y qué es lo que se necesita. Con esto se obtendría una mejora en el servicio a los clientes, un incremento en las ventas y mejor manejo y administración de los recursos económicos de la empresa.

Uno de los principales sistemas de información es el que se conoce como Enterprise Resource Planning, por sus siglas en inglés ERP, dichos sistemas han ayudado enormemente a las compañías en su

crecimiento y expansión mundial ya que básicamente las apoya en su integración por medio de los diferentes módulos de que está compuesta. [6]

Algunas de las funciones principales del ERP son: Organizar y estandarizar procesos o datos internos de la empresa, administrar de manera integrada y eficiente la información de la empresa, comunicar diferentes áreas de las empresas mediante procesos y datos electrónicos, procesar la información convirtiéndola en conocimientos para su aplicación en la toma de decisiones y automatizar una gran parte de los procesos básicos de las empresas.

Actualmente, su implementación ha adquirido un apogeo significativo, ya que las empresas buscan maximizar sus beneficios y minimizar sus costos y entre ellas se encuentran: MEXLUB, SABRITAS, CIGATAM, HULES [7]

Para la Empresa Pública GIASA, era imprescindible, no sólo avanzar en eficacia y calidad, sino también en eficiencia. Por ello le resultó primordial disponer de un Sistema de Información capaz de soportar una moderna gestión de las infraestructuras. [8]

La empresa informática CTI aplicando sistemas ERP aumentó su producción, en el 2005 facturaron 4,2 millones, en 2006 llegaron a 4,6 y para este año se provee llegar a los 5,3 millones. [9]

Por todo lo planteado anteriormente el MINFAR optó por implementar un sistema ERP, ya que con la aplicación del mismo se puede obtener gran cantidad de información útil a través de los diferentes departamentos del área de Planificación Material y Financiera accediendo con gran eficiencia a esta, en el tiempo requerido para la administración de áreas funcionales como atención a clientes, recursos humanos, manejo de inventarios, proveedores, etc. Se optimizan y se benefician gracias al manejo y el uso oportuno de los datos convertidos en información relevante y se optimiza la toma de decisiones por parte de la directiva así como todo lo relacionado con los diferentes procesos, por tener la capacidad de contar con información confiable, oportuna y veraz.

A pesar de tener muchas funciones semejantes a las de cualquier sistema ERP en el mundo, hay aspectos que todavía hacen al MINFAR, tener un sistema a la medida, pues presenta una interfaz única de trabajo, para que la calidad de la información sea mejor, se desecha la competitividad, pues no se requiere de continuas optimizaciones de costos, ya sea de producción, comercialización o administración, la información no tiene por que integrarse en determinadas áreas vitales como finanzas, distribución y manufactura, y no tiene por que ser compartida del todo, pues se está tratando con un ministerio de carácter militar que parte de su información requiere ser confidencial.

1.3 ¿Por qué una Web?

El surgimiento de la Web, así como el desarrollo de la World Wide Web o Telaraña Mundial, ha ayudado al crecimiento de Internet. Compañías pequeñas, empresas grandes, ayuntamientos, estados, gobiernos de distintos países, universidades y bibliotecas, están presentes en Internet. La World Wide Web permite, de una manera más organizada, acceder a la información disponible en Internet.

Internet es la gran red de redes, también llamada Supercarretera de la información. Es el resultado de la interconexión de miles de computadoras en todo el mundo. Todas ellas comparten los protocolos de comunicación, es decir que todos hablan el mismo lenguaje para ponerse en contacto unas con otras.

La Web, es un enorme universo de información que se accede a través de Internet, una fuente inagotable del conocimiento humano. Brinda servicios básicos como: correo electrónico, noticias en red, acceso a computadoras remotas y sistemas de adquisición de datos, así como la capacidad para transferir información entre computadoras remotas. Es un sistema de información global, interactivo, dinámico, distribuido, gráfico, basado en hipertexto, con plataforma de enlaces cruzados, que se ejecuta en Internet. Indiscutiblemente la Web es uno de los componentes más usados por el ínter nauta, es la forma más moderna de ofrecer información, es el medio más potente. La información se ofrece en forma de páginas electrónicas, que permiten ir de un lugar a otro. Una vez que el usuario está conectado a Internet, tiene que utilizar programas capaces de acceder a páginas Web y de llevarlo de unas a otras siguiendo los vínculos. Estos programas son los navegadores, uno de los más conocidos es el Explorer de Microsoft.

1.4 Sitios Web y Aplicaciones Web.

Un **sitio web** (en inglés: website) es un conjunto de páginas web, típicamente comunes a un dominio de Internet o subdominio en la World Wide Web en Internet. Hoy en día, hay más de 100 millones de sitios web en el mundo con dominios registrados. Una página web es un documento HTML⁶/XHTML⁷ accesible generalmente mediante el protocolo HTTP⁸ de Internet.

A las páginas de un sitio web se accede desde una URL⁹ raíz común llamada portada, que normalmente reside en el mismo servidor físico. Las URLs organizan las páginas en una jerarquía, aunque los hiperenlaces entre ellas controlan cómo el lector percibe la estructura general y cómo el tráfico web fluye entre las diferentes partes de los sitios. [10]

Su Sitio Web no necesariamente debe localizarse en el sistema de cómputo de su negocio. Los documentos que integran el Sitio Web pueden ubicarse en un equipo en otra localidad, inclusive en otro país. El único requisito es que el equipo en el que residan los documentos esté conectado a la red mundial de Internet. Este equipo de cómputo o Servidor Web, como se le denomina técnicamente, puede contener más de un sitio Web y atender concurrentemente a los visitantes de cada uno de los diferentes sitios. [11]

Una **aplicación web** es una aplicación informática que los usuarios utilizan accediendo a un servidor web a través de Internet o de una intranet. Las aplicaciones web son populares debido a la practicidad del navegador web como cliente 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. [6]

Las aplicaciones web generan dinámicamente una serie de páginas en un formato estándar, soportado por navegadores web comunes como HTML o XHTML. Se utilizan lenguajes interpretados del lado del cliente, tales como JavaScript¹⁰, para añadir elementos dinámicos a la interfaz de usuario. Generalmente cada página web individual es enviada al cliente como un documento estático, pero la secuencia de páginas provee de una experiencia interactiva.

1.5 Páginas dinámicas en servidor.

Las páginas dinámicas son páginas HTML generadas a partir de lenguajes de programación (scripts¹¹) que son ejecutados en el propio servidor web.

Las páginas dinámicas, pueden ayudar a gestionar más fácilmente los contenidos de un sitio web y a interactuar con Bases de Datos. [12]

Por ejemplo, si se tienen uno o varios menús en las páginas, y se desean modificarlos, no hay que ir página por página editándolos, sino que bastará hacerlo una sola vez. En el resto de las páginas bastará incluir una línea de código y el cambio habrá sido efectuado. [13]

Las páginas dinámicas son especialmente útiles en trabajos que se tiene que acceder a información centralizada, situada en una base de datos en el servidor, y cuando por razones de seguridad los cálculos no se pueden realizar en el ordenador del usuario.

Es importante destacar que las páginas dinámicas de servidor son necesarias porque para hacer la mayoría de las aplicaciones web se debe tener acceso a muchos recursos externos al ordenador del

cliente, principalmente bases de datos alojadas en servidores de Internet. Un caso claro es un banco: no tiene ningún sentido que el cliente tenga acceso a toda la base de datos, sólo a la información que le concierne.

Las ventajas de este tipo de programación son que el cliente no puede ver los scripts, ya que se ejecutan y transforman en HTML antes de enviarlos. Además son independientes del navegador del usuario, ya que el código que reciben es HTML fácilmente interpretable.

Como desventajas se puede señalar que será necesario un servidor más potente y con más capacidades que el necesario para las páginas cliente. Además, estos servidores podrán soportar menos usuarios concurrentes, porque se requerirá más tiempo de procesamiento para cada uno. [9]

1.6 Lenguaje del lado del cliente.

El Cliente es un ordenador que accede a recursos y servicios brindados por otro llamado Servidor, generalmente en forma remota. El cliente recibe los servicios que ofrece un servidor. El término se usó inicialmente para dispositivos que no eran capaces de ejecutar programas por sí mismos, pero podían interactuar con ordenadores remotos por red. Estos terminales eran clientes de los ordenadores centrales de tiempo compartido.

1.6.1 JavaScript

JavaScript es el lenguaje que permite interactuar con el navegador de manera dinámica y eficaz, proporcionando a las páginas web dinamismo y vida. Se trata de un lenguaje de tipo script compacto, basado en objetos y guiado por eventos diseñado específicamente para el desarrollo de aplicaciones cliente-servidor dentro del ámbito de Internet.

Los programas JavaScript van dentro de los documentos HTML, y se encargan de realizar acciones en el cliente, como pueden ser pedir datos, confirmaciones, mostrar mensajes, crear animaciones, comprobar campos. [6]

JavaScript es un lenguaje interpretado, es decir, que no requiere compilación, utilizado principalmente en páginas web, con una sintaxis semejante a la del lenguaje Java¹² y el lenguaje C¹³. Todos los navegadores interpretan el código JavaScript integrado dentro de las páginas web. [10]

La característica principal de Javascript, de hecho, es la de ser un lenguaje de scripting, pero, sobre todo, la de ser el lenguaje de scripting por excelencia y, sin lugar a dudas, el más usado. Esta particularidad conlleva una notable serie de ventajas y desventajas según el uso que se le deba dar y teniendo en cuenta la relación que se establece entre el mecanismo cliente-servidor. [10]

¿Cuáles son las ventajas y cuáles las desventajas respectivas de los lenguajes de scripting y los lenguajes compilados? Algunos ejemplos:

- ✓ Los script tienen capacidades limitadas, por razones de seguridad, por lo cual no es posible hacer todo con Javascript, sino que es necesario usarlo conjuntamente con otros lenguajes evolucionados, posiblemente más seguros, como Java.
- ✓ Un problema importante es que el código es visible y puede ser leído por cualquier persona, incluso si está protegido con las leyes del copyright.
- ✓ El código del script debe descargarse completamente antes de poderse ejecutar. Si los datos que un script utiliza son muchos (por ejemplo, una recopilación de citas que se mostrará de manera casual), el tiempo que tardará en descargarse será muy largo, mientras que la interrogación de la misma base de datos en el servidor sería más rápida.
- ✓ El código Javascript se ejecuta en el cliente por lo que el servidor no es solicitado más de lo debido; un script ejecutado en el servidor, sin embargo, sometería a éste a dura prueba y los servidores de capacidades más limitadas podrían resentirse por causa de una continua solicitud por un mayor número de usuarios.

Este fue el lenguaje seleccionado para validar los formularios en las páginas Web que conforman las Interfaces de la aplicación por su portabilidad y funcionalidad.

1.6.2 HTML

HTML (HyperText Markup Language) es un lenguaje muy sencillo que permite describir hipertexto, es decir, texto presentado de forma estructurada y agradable, con enlaces (hipervínculos) que conducen a otros documentos o fuentes de información relacionadas, y con inserciones multimedia (gráficos, sonido, entre otros). Este lenguaje es el que se utiliza para presentar información en el World Wide Web.

La descripción se basa en especificar en el texto la estructura lógica del contenido (títulos, párrafos de texto normal, enumeraciones, definiciones, citas, etc.), así como los diferentes efectos que se quieren dar

(cursiva, negrita, o un gráfico determinado) y dejar que luego la presentación final de dicho hipertexto se realice por un programa especializado. [14]

1.7 Lenguajes que corren en el servidor.

Un Servidor es una aplicación informática o programa que realiza algunas tareas en beneficio de otras aplicaciones llamadas clientes. Algunos servicios habituales son los servicios de archivos, que permiten a los usuarios almacenar y acceder a los archivos de un ordenador y los servicios de aplicaciones, que realizan tareas en beneficio directo del usuario final. Es posible que un ordenador cumpla simultáneamente las funciones de cliente y de servidor.

Un servidor no es necesariamente una máquina de última generación, grande y potente, no es necesariamente un superordenador; un servidor puede ser desde una computadora Pentium 2 hasta una máquina sumamente poderosa como: servidores web, bases de datos grandes, procesadores especiales y hasta varios gigas de memoria. Un servidor también puede ser un proceso que entrega información o sirve a otro proceso, el modelo cliente servidor no necesariamente implica tener dos ordenadores, ya que un proceso cliente puede solicitar algo como una impresión a un proceso servidor en un mismo ordenador. [8]

1.7.1 PHP

PHP¹⁴ (Hypertext Preprocessor), es un lenguaje interpretado de alto nivel impregnado en páginas HTML y ejecutado en el servidor. [15]

PHP, se conoce originalmente como Personal Home Pages, fue concebido en el otoño de 1994 por Rasmus Lerdorf. La primera versión salió en los comienzos de 1995, y fue ahí donde Rasmus se dio cuenta que haciendo el proyecto código-abierto, las personas arreglarían sus problemas. La primera versión fue muy precaria y solo reconocía unas pocas macros y brindaba algunas utilidades que se usaban comúnmente en sitios web.

Posteriormente fue reescrito a mediados de 1995 y se renombró PHP/FI versión 2. El FI en esta versión quería decir Interprete Formal. Lo que Rasmus había agregado a PHP fue de acuerdo a las necesidades crecientes de los sitios web. El soporte para MySQL¹⁵ fue agregado. PHP/FI tuvo un crecimiento masivo, y otras personas empezaron a contribuir, programando regularmente.

A mediados de 1997 Zeev Suraski y Andi Gutmans reescribieron el código principal, y PHP cambió de estar en manos de Rasmus a un grupo más orientado al proyecto. Esto formó las bases para que PHP3, fuese ahora llamado PHP.

La última versión, de PHP4, es otra reescritura de Suraski y Gutmans y está basada en el motor Zend¹⁶. PHP ahora tiene 200 contribuyentes regularmente trabajando en varias partes del proyecto. Tiene una cantidad muy grande extensiones, módulos y soporta todos los servidores más populares nativamente, y además tiene soporte para MySQL y ODBC¹⁷.

Las últimas estadísticas muestran que PHP es actualmente usado por más de 5.5 millones de dominios, y ha tenido un gran crecimiento durante el último año. [16]

PHP5 tiene las cuatro grandes características necesarias para ser un potente lenguaje de scripts: velocidad, estabilidad, simplicidad y seguridad:

Velocidad: PHP no requiere demasiados recursos de sistema. Por esta razón no crea demoras en la máquina.

Estabilidad: Con el respaldo de una increíble comunidad de programadores y usuarios es mucho más difícil para los errores sobrevivir. Se utiliza su propio sistema de administración de recursos y dispone de un sofisticado método de manejo de variables, conformando un sistema robusto y estable.

Seguridad: PHP provee diferentes niveles de seguridad, estos pueden ser configurados desde el archivo .INI

Simplicidad: PHP permite a los programadores generar código en el menor tiempo posible. Usuarios con experiencia en los lenguajes C y C++¹⁸ podrán aprender a utilizar PHP mucho más rápido.

Ventajas del uso de PHP.

Algunas de las principales ventajas de PHP están:

- ✓ Es un lenguaje muy fácil de aprender dada su similitud con Perl¹⁹ y C.
- ✓ Soporta la programación orientada a objetos.
- ✓ La validación de los parámetros que entra el usuario es realizada en el servidor y no en el cliente. Esto hace que el proceso de chequeo de los datos sea más eficaz y que se pueda apoyar en toda una serie de funciones de seguridad ya implementadas.

- ✓ En PHP no existen punteros.
- ✓ Constituye un software libre, pudiéndose obtener en Internet sin necesidad de comprar o adquirir alguna licencia.
- ✓ Presenta muchas interfaces diferentes para cada tipo de servidor y puede actualmente ser configurado para trabajar como CGI²⁰ ó se puede ejecutar sobre Apache²¹, IIS²², AOLSERVER²³, Roxen²⁴.
- ✓ Puede interactuar con muchos motores de bases de datos tales como MySQL, MS SQL²⁵, Oracle²⁶, Informix²⁷, PostgreSQL²⁸, entre otros.

Desventajas.

Algunas desventajas que tiene son:

- ✓ Entre sus mayores dificultades se encuentra el manejo de errores que implementa, pues no es tan sofisticado como en Cold fusion²⁹ o ASP³⁰, y no presenta un debugger³¹ o depurador, aunque actualmente hay software donde se puede depurar PHP.
- ✓ Todo el trabajo se realiza en el servidor sin relegar nada al cliente, por lo que se podría ser ineficiente si las solicitudes aumentan en número.

Ventajas adicionales de PHP.

PHP corre en (casi) cualquier plataforma utilizando el mismo código fuente, pudiendo ser compilado y ejecutado en 25 plataformas, incluyendo diferentes versiones de Unix, Windows³² (95, 98, NT, ME, 2000, XP) y Macs³³. Como en todos los sistemas se utiliza el mismo código base, los scripts pueden ser ejecutados de manera independiente al OS³⁴.

La sintaxis de PHP es similar a la del C, por esto cualquiera con experiencia en lenguajes del estilo C podrá entender rápidamente PHP. Entre los lenguajes del tipo C se incluye al Java y Javascript, de hecho mucha de la funcionalidad del PHP se la debe al C.

PHP es completamente expandible. Está compuesto de un sistema principal (escrito por Zend), un conjunto de módulos y una variedad de extensiones de código. Tiene diferentes interfaces para cada tipo de servidor. PHP actualmente se puede ejecutar bajo Apache, IIS, AOLServer, Roxen y THTTPD³⁵. Otra alternativa es configurarlo como módulo CGI.

Puede interactuar con muchos motores de bases de datos tales como MySQL, MS SQL, Oracle, Informix, PostgreSQL, y otros muchos. Siempre se podrá disponer de ODBC para situaciones que lo requieran.

Tiene una gran variedad de módulos, cuando un programador en PHP necesite una interfaz para una librería en particular, fácilmente podrá crear una API³⁶ para esta. Algunas de las que ya vienen implementadas permiten manejo de gráficos, archivos PDF³⁷, Flash³⁸, calendarios, XML³⁹, IMAP/POP⁴⁰. [16]

PHP generalmente es utilizado como módulo de Apache, lo que lo hace extremadamente veloz. Está completamente escrito en C, así que se ejecuta rápidamente utilizando poca memoria.

PHP es código abierto, lo cual significa que el usuario no depende de una compañía específica para arreglar cosas que no funcionan, además no está forzado a pagar actualizaciones anuales para tener una versión que funcione.

1.8 Tecnología AJAX.

AJAX⁴¹ (Asynchronous JavaScript And XML) Es una técnica de desarrollo Web que genera aplicaciones Web interactivas combinando:

- ✓ Document Object Model (DOM⁴²) para visualizar dinámicamente e interactuar con la información presentada.
- ✓ XML, XSLT⁴⁶ para intercambiar y manipular datos.
- ✓ CSS⁴³ para definir el aspecto del documento.
- ✓ JSON⁴⁵ y JSON-RPC pueden ser alternativas a XML/XSLT
- ✓ XMLHttpRequest⁴⁴ para recuperar datos de forma asincrónica.
- ✓ Javascript como nexos de unión de todas estas tecnologías.

AJAX es un patrón de diseño que propone un nuevo modelo de interacción Web combinando las tecnologías anteriores. [17]

¿Por qué AJAX?

Las aplicaciones web proliferan debido a su simplicidad, pero ofrecen una menor interactividad y usabilidad en comparación con las aplicaciones de escritorio, debido a que la interacción del usuario con una aplicación web se interrumpe cada vez que se necesita algo del servidor.

Varias tecnologías han sido diseñadas para resolver este problema, Java Applets, FLASH, AJAX es una nueva solución que no requiere plugins o capacidades específicas de ciertos navegadores y no constituye una tecnología en sí, por eso carece de un entorno de desarrollo propio, pero es un término que engloba a un grupo de éstas que trabajan conjuntamente.

El hecho de que el intercambio de datos se realice de forma asíncrona sirve para hacer que las aplicaciones Web funcionen de una manera casi transparente al usuario en términos de comunicación con el servidor.

El resultado es una interfaz con mayor respuesta, dado que la cantidad de datos intercambiados entre el navegador web y el servidor web es reducida enormemente. También se ahorra mucho tiempo de procesamiento en el servidor web, ya que una parte importante de dicho procesamiento se realiza en el lado del cliente.

Características AJAX.

- ✓ Las aplicaciones son más interactivas, responden a las interacciones del usuario más rápidamente, al estilo aplicaciones de escritorio.
- ✓ Estas aplicaciones tienen un aspecto muy similar a las aplicaciones de escritorio tradicionales sin depender de plugins o características específicas de los navegadores.
- ✓ Se reduce el tamaño de la información intercambiada. Se libera de procesamiento a la parte servidora (se realiza en la parte cliente)
- ✓ AJAX actualiza porciones de la página en vez de la página completa

¿Por qué AJAX es diferente?

Una aplicación AJAX elimina las demoras en la ejecución introduciendo un intermediario (el motor AJAX) entre el usuario y el servidor. Parecería que añadir una capa a la aplicación le conferiría una peor respuesta, pero en realidad ocurre lo contrario.

En lugar de cargar una página Web, al iniciar la sesión, lo que hace el navegador es cargar el motor AJAX (escrito en JavaScript y normalmente escondido en un marco oculto).

AJAX contra aplicaciones web tradicionales.

En las aplicaciones web tradicionales los usuarios interactúan mediante formularios, que al enviarse, realizan una petición al servidor web. El servidor se comporta según lo enviado en el formulario y contesta enviando una nueva página web. Se desperdicia mucho ancho de banda, ya que gran parte del HTML enviado en la segunda página web, ya estaba presente en la primera. Además, de esta manera no es posible crear aplicaciones con un grado de interacción similar al de las aplicaciones habituales.

Además en este tipo de aplicaciones web tradicionales mientras el servidor esta haciendo lo suyo, ¿qué esta haciendo el usuario? Exacto, esperando. Y, en cada paso de la tarea, el usuario espera por más. Obviamente, si estuviéramos diseñando la web desde cero para aplicaciones, no querríamos hacer esperar a los usuarios. Una vez que la interfaz esta cargada, ¿por qué la interacción del usuario debería detenerse cada vez que la aplicación necesita algo del servidor? De hecho, ¿por qué debería el usuario ver la aplicación yendo al servidor?

Por otro lado en aplicaciones AJAX se envían peticiones vía http(s) mediante eventos, scripts o rutinas al servidor web, para obtener únicamente la información necesaria, empleando SOAP o algún otro lenguaje para servicios web basado en XML, y usando JavaScript en el cliente para procesar la respuesta del servidor web. Esto redundo en una mayor interacción gracias a la reducción de información intercambiada entre servidor y cliente, y a que parte del proceso de la información se hace en el propio cliente, liberando al servidor de ese trabajo. Además esta petición se realiza como proceso de fondo (background), por lo que el usuario no tiene que esperar que el proceso concluya en su totalidad para continuar interactuando con la aplicación. La contrapartida es que la descarga inicial de la página es más lenta al tenerse que bajar todo el código JavaScript.

En fin, en el modelo clásico, cada vez que queremos cargar una nueva página Web con nuevos datos, hemos de enviar una petición al servidor Web, y éste nos devuelve la página entera, que incluye tanto los datos a mostrar como la presentación de la misma. Sin embargo utilizando el modelo AJAX, cuando queremos cargar datos nuevos lo que hacemos es enviar una petición HTTP al servidor Web que nos devuelve únicamente los datos a mostrar (en formato XML), no la presentación de la página. Con este sistema lo que conseguimos es reducir el volumen de tráfico entre cliente y servidor, y también conseguimos que no se tengan que cargar páginas HTML enteras cada vez que queremos representar nuevos datos.

El motor AJAX.

El motor AJAX (XMLHttpRequest) colocado entre el usuario y el servidor web evita el ciclo inicio-detener-inicio-detener característico de las aplicaciones web tradicionales y no es más que un fichero JavaScript que acompaña al HTML. El cual es cargado al inicio de la sesión y tiene una doble responsabilidad, primero generar la interfaz visualizada por el usuario y segundo comunicarse con el servidor en representación del usuario, lo cual ocurre de manera asíncrona evitando que el usuario vea una página blanca o el reloj de arena (de espera) cada vez que realice una acción.

Toda acción del usuario que normalmente generaría una petición HTTP, adquiere la forma de una llamada JavaScript al motor AJAX. Cualquier acción del usuario que no requiera un viaje al servidor (tales como validación de datos, edición de datos en memoria e incluso la propia navegación) es procesada por él mismo. Si el motor necesita algún tipo de proceso en el servidor para generar la respuesta (enviar datos para procesamiento, carga de código de interfaz adicional o petición de datos nuevos) realiza las peticiones de manera asíncrona, normalmente usando el objeto XMLHttpRequest, sin detener la interacción del usuario con la aplicación.

Seguridad en AJAX.

Por motivos de seguridad las invocaciones a recursos vía HTTP solamente pueden efectuarse a páginas alojadas en el mismo dominio que el de la página que contenía el script. Si queremos que la información de nuestras peticiones no sea visible deberemos utilizar un canal HTTPS. [17]

1.9 Sistemas de Gestión de Bases de Datos.

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

1.9.1 PostgreSQL.

PostgreSQL es un motor de base de datos, es servidor de base de datos relacional libre, liberado bajo la licencia BSD⁴⁷. [18]

PostgreSQL es el servidor de bases de datos de código abierto más potente que existe y es por tanto la alternativa a MySQL cuando se necesitan características avanzadas como transacciones, procedimientos almacenados, triggers, vistas, etc.

Es el servidor de bases de datos más utilizado por los programadores de servlets de Java y, en general, por todos aquellos que realizan aplicaciones cliente servidor complejas o críticas en el mundo Linux/Unix⁵³.

Desventajas:

- ✓ Consume gran cantidad de recursos.
- ✓ Tiene un límite de 8K por fila, aunque se puede aumentar a 32K, con una disminución considerable del rendimiento.
- ✓ Es lento.

Ventajas:

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

- ✓ Ahorros considerables en costos de operación
- ✓ Estabilidad y confiabilidad legendarias
- ✓ Extensible
- ✓ Multiplataforma
- ✓ Diseñado para ambientes de alto volumen
- ✓ Herramientas gráficas de diseño y administración de bases de datos

Por las características anteriores, el SGBD que se escogió para desarrollar el sistema fue el PostgreSQL, ya que el SQL Server no es multiplataforma y además hay que pagar un importe por su licencia, así mismo el Oracle, su coste es incluso mas elevado que el del SQL Server a parte de que el Oracle necesita una máquina con elevados recursos (memoria RAM, velocidad de microprocesador, capacidad de disco duro), en cuanto al MySQL podemos decir que cuando se almacena gran cantidad de datos, estos pueden perderse o dañarse, por lo que se decidió escoger el PostgreSQL que a pesar de ser el mas lento, aventaja en los demás aspectos a los otros SGBD comparados. [18]

1.10 Herramientas CASE.

Son las herramientas Case⁴⁸ el mejor método para el análisis y soluciones de software, ya que han venido a mejorar los aspectos claves en el desarrollo de los sistemas de información, han sido creadas para la automatización de procesos de análisis, diseño e implementación, brindando un sin número de componentes que hacen que los proyectos sean cada día más eficientes para los usuarios finales.

Desde que se crearon éstas herramientas (1984) hasta la actualidad, las CASE cuentan con una credibilidad y exactitud que tienen un reconocimiento universal, siendo usadas por cualquier analista y / o programador que busca un resultado óptimo y eficaz, para cada uno de sus procesos.

Las herramientas CASE brindan:

- ✓ Topologías de aplicación flexibles
- ✓ Aplicaciones portátiles
- ✓ Control de versión
- ✓ Crear código compilado en el servidor
- ✓ Dar un Soporte multiusuario
- ✓ Ofrecer Seguridad [19]

1.10.1 Visual Paradigm.

Visual Paradigm⁴⁹ es una herramienta CASE que utiliza “UML⁵⁰”: como lenguaje de modelado.

Visual Paradigm ofrece:

- ✓Entorno de creación de diagramas para UML 2.0
- ✓Diseño centrado en casos de uso y enfocado al negocio que generan un software de mayor calidad
- ✓Uso de un lenguaje estándar común a todo el equipo de desarrollo que facilita la comunicación
- ✓Capacidades de ingeniería directa (versión profesional) e inversa.
- ✓Modelo y código que permanece sincronizado en todo el ciclo de desarrollo
- ✓Disponibilidad de múltiples versiones, para cada necesidad.
- ✓Disponibilidad en múltiples plataformas.

Rasgos:

Generales:

- ✓Todos los diagramas UML.
- ✓Soporta UML 2.0.
- ✓Mecanismos de extensión de UML.
- ✓Soporte robusto para la notación de análisis.
- ✓Repositorio de clases: clases y paquetes pueden ser compartidos.
- ✓Chequeador de Modelos.
- ✓Diagramas de capas automáticos.

Modelado de clases e ingeniería de código:

- ✓Control de visibilidades en los archivos compartidos.
- ✓Sincronización automática de código.
- ✓Generación de código.
- ✓Ingeniería inversa de código.

Reversa instantánea:

Genera modelos VP-UML instantáneamente a partir de código binario .NET.

Modelado básico:

- ✓Recursos del Visual Paradigm.
- ✓Colores para los elementos de los diagramas.

- ✓ Conectores 'conectables' y configurables.
- ✓ Estilos múltiples de movimiento del estado.
- ✓ Hitos visuales para acciones válidas o inválidas del usuario.
- ✓ Facilidades para copiar y pegar elementos.
- ✓ Edición de texto en diagramas UML.
- ✓ Edición de propiedades de diagramas fácilmente.
- ✓ Los paquetes pueden contener elementos UML.

Modelado Avanzado:

- ✓ Generador de Modelos.
- ✓ Generación automática de diagramas a partir de otros diagramas.

Interfaz de Usuario:

- ✓ Permite diferentes perspectivas del proyecto.
- ✓ Explorador de proyecto.

Análisis y modelado de Casos de Uso:

- ✓ Análisis de texto a nivel de proyecto y a nivel de casos de uso.
- ✓ Crea un modelo a partir de clases candidatas.
- ✓ Descripción de casos de uso.
- ✓ Flujo de eventos.
- ✓ Planificador de Casos de Uso (manual).
- ✓ Soporte para extensión de casos de uso.

Generación de documentación:

- ✓ Generación de HTML.
- ✓ Generación de PDF.

Exportaciones e importaciones:

- ✓ Exporta diagrama a imagen.
- ✓ Importa Racional Rose Project.

Diagramas en Visual Paradigm.

- ✓ Representación gráfica de una colección de elementos de modelado, a menudo dibujada como un grafo con vértices conectados por arcos.
- ✓ Diferentes tipos de diagrama permiten ver el sistema desde diferentes perspectivas.

Los tipos de diagramas que se pueden incluir son:

- ✓ Diagramas de Caso de Uso: Representan la funcionalidad del sistema.
- ✓ Diagramas de Clase: Describen la estructura (estática) del sistema.
- ✓ Diagrama de Actividad: Describen como se desarrolla un flujo de actividades entre elementos del sistema o del dominio.
- ✓ Diagramas de Secuencia: Describen la interacción entre elementos del sistema en el tiempo.
- ✓ Diagramas de Estado: Describen el estado, condiciones y respuesta de los elementos del sistema.
- ✓ Diagrama de Componentes: Describen la organización de los elementos físicos que implementan sistema.
- ✓ Diagrama de Despliegue: Describen la configuración del entorno de máquinas y redes sobre el que se distribuyen componentes y procesos del sistema. [20]

Se integra con las siguientes herramientas Java:

- ✓ Eclipse/IBM WebSphere
- ✓ JBuilder
- ✓ NetBeans IDE
- ✓ Oracle JDeveloper
- ✓ BEA Weblogic

Está disponible en varias ediciones, cada una destinada a unas necesidades: Enterprise, Professional, Community, Standard, Modeler y Personal. [21]

Ventajas:

- ✓ Apoya todo lo básico en cuanto a artefactos generados en las etapas de definición de requerimientos y de especificación de componentes.
- ✓ Tiene apoyo adicional en cuanto a generación de artefactos automáticamente.

Desventajas:

No está enfocado a CBD, por tanto herramientas como la generación automática de código no son aplicables. Cualquier diagrama generado automáticamente a partir de otro diagrama podría perder su utilidad.

1.11 Conclusiones.

Hasta aquí se han visto las características de diferentes herramientas, para la creación de un software o aplicación así como algunas metodologías. Para el desarrollo de la aplicación Web, este trabajo está enfocado hacia el desarrollo software libre, debido a las condiciones que enfrenta el país, además de las restricciones que impone el MINFAR como requisito en la elaboración de esta aplicación. Después de este análisis se quiere dejar plasmado que el lenguaje de programación a utilizar es el PHP por sus características antes mencionadas, así como el Javascript para el control de las diferentes funciones a realizar en el cliente y como gestor de base de datos se estableció el PosgreSql.

En la selección de la metodología se decidió por la modelación del proceso unificado (RUP)⁵¹ por ser un proceso iterativo e incremental, como herramienta CASE el VisualParadimg por las características antes mencionadas y como lenguaje de modelado UML.

CAPÍTULO 2: MODELACIÓN DE LA PRPUESTA DE SOLUCIÓN.

2.1 Introducción.

En la construcción y desarrollo de proyectos se aplican métodos y técnicas para resolver los problemas, la informática aporta herramientas y procedimientos sobre los que se apoya la ingeniería de software. Estos contribuyen a mejorar la calidad de los productos de software, aumentar la productividad y trabajo de los ingenieros del software, facilitar el control del proceso de desarrollo de software, suministrar a los desarrolladores las bases para construir software de alta calidad en una forma eficiente, definir una disciplina que garantice la producción y el mantenimiento de los productos software desarrollados, en el plazo fijado y dentro del costo estimado . [14]

La ingeniería de software es, por tanto, la aplicación de un método sistemático, disciplinado y cuantificable al desarrollo, operación y mantenimiento de software. [22]

Es por ello que para el logro exitoso de un producto software es preciso que se pongan en práctica estos métodos y técnicas. En el presente capítulo se obtiene el Modelo del Negocio como primer paso para entender y describir cada proceso del negocio, especificando sus datos, actividades (o tareas), roles (o agentes) y reglas del mismo. Posteriormente se derivan los requerimientos del sistema y ya con estos se realiza el Modelo del Sistema.

2.2 Modelado del Negocio.

Un modelo de negocio (también llamado diseño de negocio) es el mecanismo por el cual un negocio trata de generar ingresos y beneficios. Es un resumen de cómo una compañía planifica servir a sus clientes. [23]

Por tanto el propósito de la modelación del negocio es en primer lugar: entender la estructura y la dinámica de la organización para poder conocer los problemas actuales e identificar mejoras potenciales, luego se necesita estar seguros de que los clientes, usuarios finales y desarrolladores tengan una idea común de la organización y posteriormente derivar los requerimientos del sistema.

2.3 Actor y Trabajador del Negocio.

Una vez que se han identificado los procesos del negocio, es preciso encontrar los agentes involucrados en su realización. Cada uno de estos agentes o actores del negocio desempeñan cierto papel (juega un rol) cuando colaboran con otros para llevar a cabo las actividades que conforman dicho caso de uso del negocio. De hecho, se identifican los roles que son jugados por agentes de la propia empresa (que incluyen trabajadores, departamentos y dispositivos físicos) o agentes externos (como clientes u otros sistemas). Por el momento se centrarán en este último tipo de roles, con los que la organización interactúa para llevar a cabo sus procesos de negocio (ver tabla 1 y tabla 2 respectivamente).

Actores del negocio	Justificación
Dirección de Economía	Es la encargada de la puesta en marcha de la aprobación de los diferentes planes presentados por los CB y OC, y posterior confección del Sistema de Planes de Demandas del MINFAR.
Órganos Consumidores	Es el encargado de la confección de su plan de demandas su posterior envío a los distintos niveles superiores, en este caso CB y la DE.

Tabla 1. Descripción de los Actores del Negocio.

Trabajadores del negocio	Justificación
Especialista de Centro de Balance	Es el encargado de revisar y balancear los planes de demandas recibidos de los diferentes OC, y enviar los mismos a su respectivo Jefe para su revisión y posterior aprobación.
Jefe de Centro de Balance	Es el encargado de revisar y aprobar el plan elaborado por el Especialista del CB que le será enviado a la DE para su revisión y posterior aprobación.

Tabla 2. Descripción de los Trabajadores del Negocio

2.4 Diagrama de Casos de Uso del Negocio (DCUN)

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

El modelo de casos de uso del negocio describe los procesos de negocio de una empresa en términos de: Casos de uso del negocio y actores del negocio (clientes y socios)

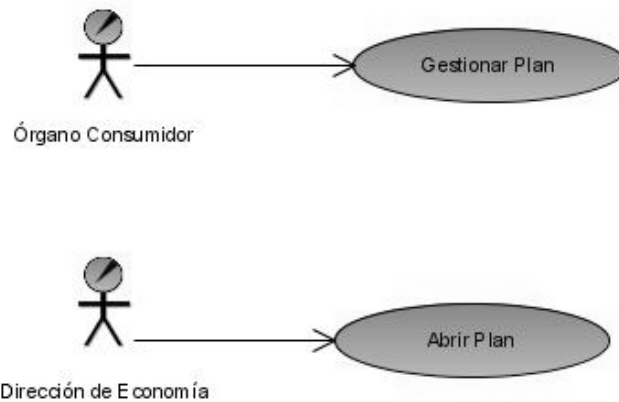


Figura 1. Diagrama de Casos de Uso del Negocio.

2.5 Descripciones Textuales.

El siguiente paso dentro del modelado del negocio es introducirse en cada uno de los casos de uso del negocio identificados, para describirlo en detalle. Inicialmente se rellena una plantilla de descripción, y después, a partir de la información reflejada en dicha plantilla, se construye un conjunto de diagramas (diagramas de actividades) que describen completamente el caso de uso del negocio (ver Tablas 3 y 4).

Caso de uso	Gestionar Plan
Actor del negocio	Órgano Consumidor (inicia)
Trabajadores del negocio	Especialista del CB, Jefe del CB
Resumen	El CU inicia cuando el OC envía su plan de demandas a su respectivo CB para su revisión y posterior aprobación. El CU termina cuando el Jefe del CB aprueba dicho plan.
Acción del actor	Respuesta del proceso de negocio
1-EI OC envía su plan de demandas al CB.	2-EI Especialista del CB recibe el plan de demandas.
	3-EI Especialista del CB realiza un estudio estadístico. -Compara el plan actual del OC con el de años anteriores verificando que no se hagan peticiones innecesarias.
	4-EI Especialista del CB consulta: Inventario Útil, Anticipo y Recursos en Contratación.
	5-EI Especialista del CB envía el plan al Jefe del CB.
	6-EI Jefe del CB recibe el plan del CB.
	7-EI Jefe del CB revisa el plan. -El Jefe del CB aprueba dicho plan y lo envía a la DE para su posterior aprobación.
Flujo Alternativo de Eventos	
Acción del Actor	Respuesta del Negocio
	3-Si el plan del OC no es correcto el Especialista del CB lo hace retornar.
	7-Si el plan del CB no es correcto el Jefe del mismo lo hace retornar.
Prioridad	Alta

Tabla 3. Descripción Textual del caso de uso “Gestionar Plan”

Caso de uso	Abrir Plan.
Actor del negocio	Dirección de Economía (inicia)
Trabajadores del negocio	Especialista del CB.
Resumen	El CU inicia cuando la DE aprueba el plan del CB y termina cuando se ha efectuado la apertura del mismo.
Acción del actor	Respuesta del proceso de negocio
1-La DE envía el plan aprobado al CB.	2- El Especialista del CB recibe el plan.
	3- El Especialista del CB toma cada producto demandado para realizar su apertura.
	4-El Especialista del Centro de Balance analiza los surtidos correspondientes a cada producto.
	5- El Especialista del CB analiza el código de cada surtido.
	6-El Especialista del CB especifica la cantidad en caso que lo requiera.
	7- El Especialista del CB indica la vía de obtención de los productos: 7.2 Si es mediante los OACE 7.2.1 Especificar el órgano 7.3 Si es otra vía especificarla
	8- El Especialista del CB especifica si es por crédito: 8.1 Si es Crédito Gubernamental 8.1.1 especificar país 8.2 Si es Crédito Comercial 8.2.1 especificar empresas
Prioridad	Alta

Tabla 4. Descripción Textual del caso de uso “Abrir Plan”

2.6 Diagramas de Actividades.

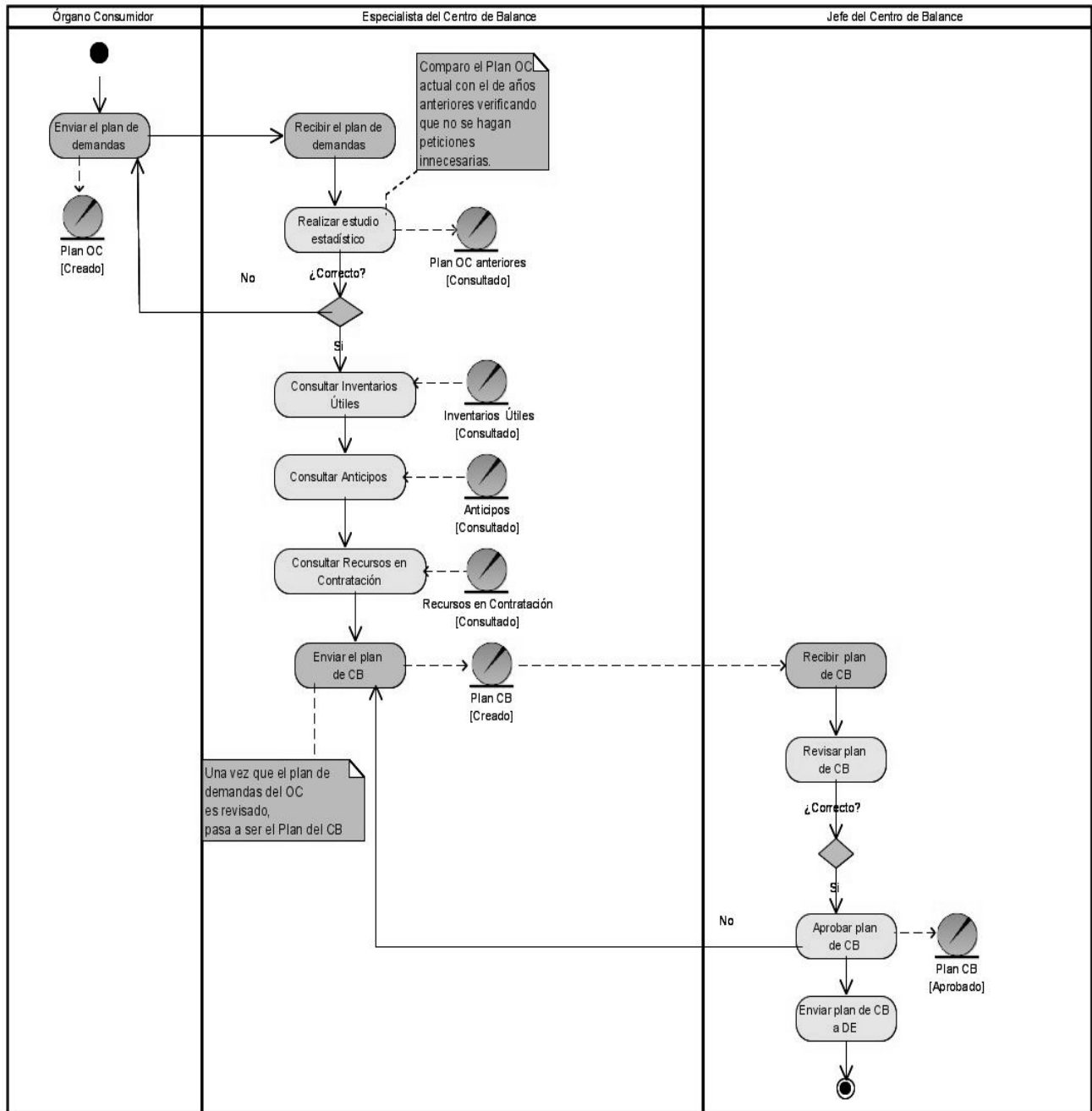


Figura 2. Diagrama de Actividades “Gestionar Plan”

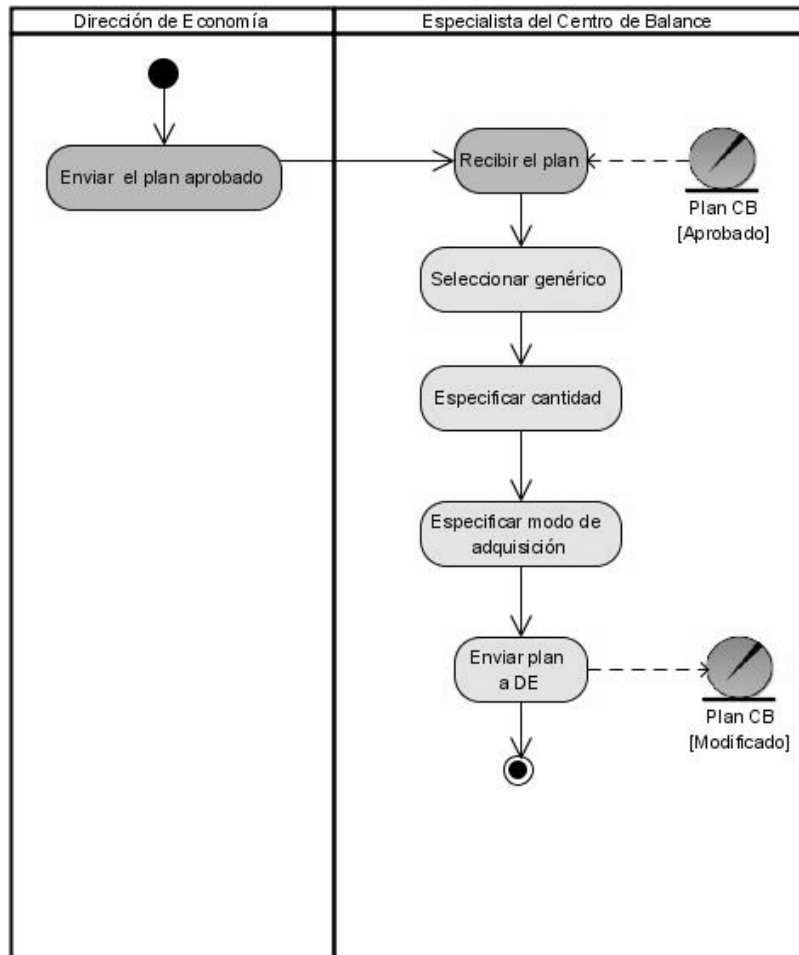


Figura 3. Diagrama de Actividades "Abrir Plan"

2.7 Clases del Modelo de Objetos.

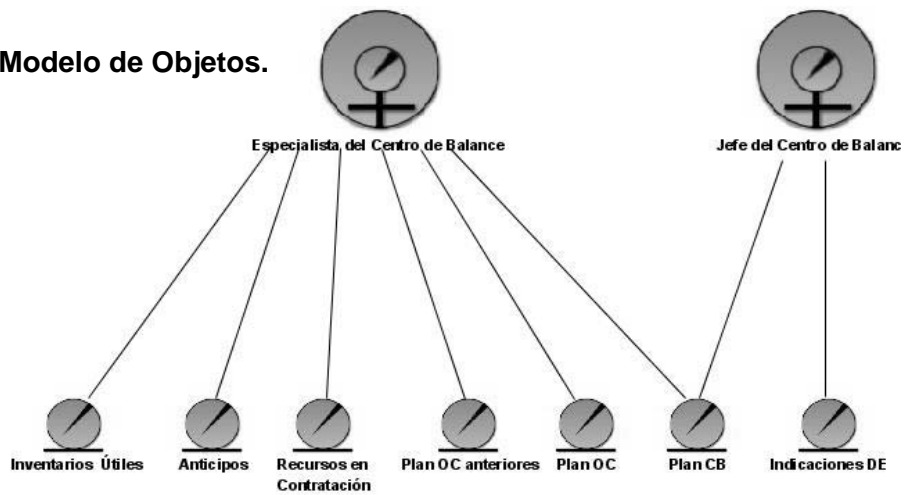


Figura 4. Modelo de Objeto

2.8 ¿Qué son los Requisitos?

Este es uno de los flujos de trabajo más importantes, porque en él se establece qué tiene que hacer exactamente el sistema que se construya. En esta línea los requisitos son el contrato que se debe cumplir, de modo que los usuarios finales tienen que comprender y aceptar los requisitos que se especifiquen.

Los objetivos del flujo de datos Requisitos son:

- ✓ Establecer y mantener un acuerdo entre clientes y otros stakeholders⁵⁴ sobre lo que el sistema podría hacer.
- ✓ Proveer a los desarrolladores un mejor entendimiento de los requisitos del sistema.
- ✓ Definir el ámbito del sistema.
- ✓ Proveer una base para la planeación de los contenidos técnicos de las iteraciones.
- ✓ Proveer una base para estimar costos y tiempo de desarrollo del sistema.
- ✓ Definir una interfaz de usuarios para el sistema, enfocada a las necesidades y metas del usuario.

Los requisitos se dividen en dos grupos. Los requisitos funcionales representan la funcionalidad del sistema. Se modelan mediante diagramas de Casos de Uso. Los requisitos no funcionales representan aquellos atributos que debe exhibir el sistema, pero que no son una funcionalidad específica. Por ejemplo requisitos de facilidad de uso, fiabilidad, eficiencia, portabilidad, etc.

2.9 Requisitos Funcionales

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

R1. Gestionar Plan

- 1.1 Mostrar plan por OC.
- 1.2 Mostrar plan de OC por año.
- 1.3 Consultar inventario útil.
- 1.4 Consultar anticipos.
- 1.5 Consultar recursos en contratación.

R2. Aprobar plan del CB.

R3. Abrir Plan

- 3.1 Seleccionar genérico.
- 3.2 Insertar cantidad.
- 3.3 Especificar forma de adquisición.

2.10 Requisitos no Funcionales

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

Para capturar los requisitos es preciso entrevistar, no sólo a los usuarios finales, sino también a todos los interesados en el proyecto, y anotar todas sus peticiones. A partir de ellas hay que descubrir lo que necesitan y expresarlo en forma de requisitos.

Apariencia o interfaz externa

La interfaz a diseñar 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, para lograr una respuesta rápida debe garantizarse una velocidad de conexión rápida, específicamente una Red de Área Local (LAN⁵²). El tiempo de respuesta y de búsqueda de información así como la generación de reportes y documentos debe ser en el tiempo más corto 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 rápido 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. Se utilizará un servidor con el sistema operativo UNIX (Linux). Se utilizará tecnología Apache versión 2.0 o superior para el servidor Web. El sistema utilizará una base datos implementada en PostgreSQL versión 8.0 o superior.

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

Disponibilidad

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

Confiabilidad

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

2.11 Modelo 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 (agentes externos) y los casos de uso (acciones) 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.

Breve descripción de los actores del sistema.

Actores del Sistema	Justificación
Especialista de Centro de Balance	Es el encargado de revisar y balancear los planes de demandas recibidos de los diferentes OC, y enviar los mismos a su respectivo Jefe para su revisión y posterior aprobación.
Jefe de Centro de Balance	Es el encargado de revisar y aprobar el plan elaborado por el Especialista del CB que le será enviado a la DE para su revisión y posterior aprobación.

Tabla 5. Descripción de los Trabajadores del Sistema

2.12 Diagrama de casos de uso del sistema.

Un diagrama de casos de uso es una representación gráfica de parte o el total de los actores y casos de uso del sistema, incluyendo sus interacciones. Un diagrama de casos de uso muestra, por tanto, los distintos requisitos funcionales que se esperan de una aplicación o sistema y cómo se relaciona con su entorno (usuarios u otras aplicaciones). [7]

En este caso el diagrama de casos de uso del sistema se dividió por paquetes para estructurar las funcionalidades de los mismos (Fig. 5).

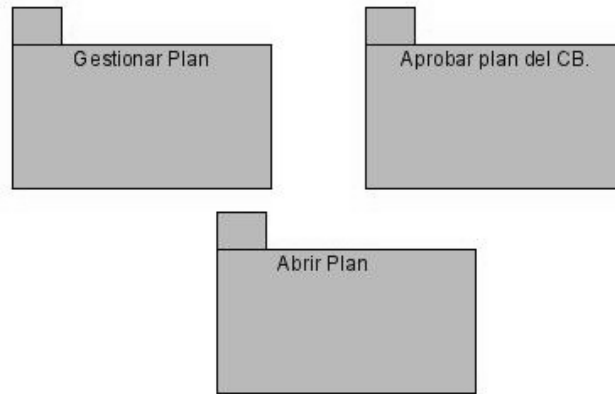


Figura 5. Estructuración del Sistema por paquetes.

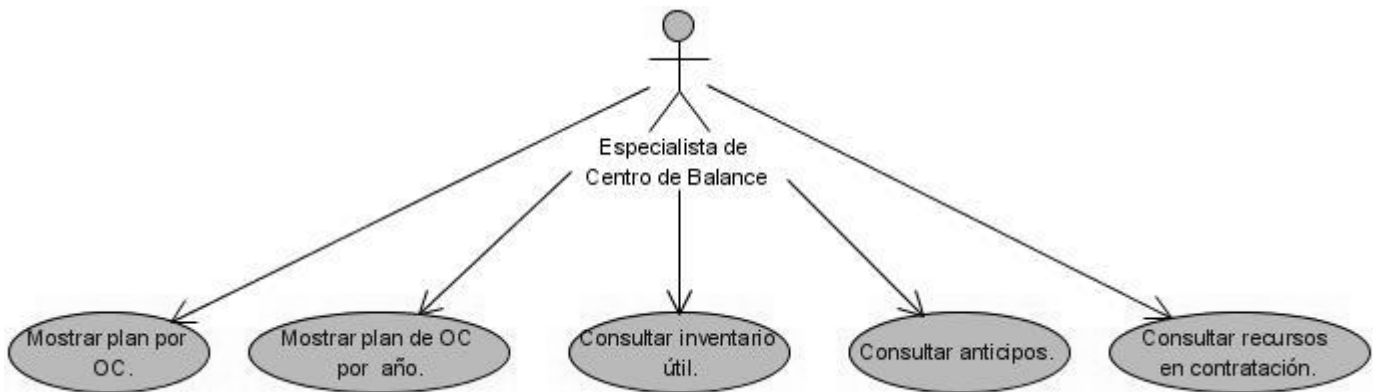


Figura 6. Diagrama de casos de uso del sistema “(Paquete Gestionar Plan)”

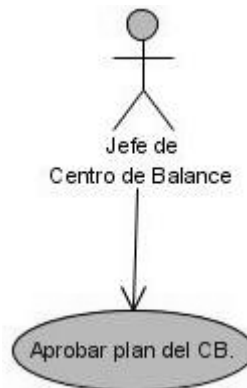


Figura 7. Diagrama de casos de uso del sistema “(Aprobar plan del CB)”



Figura 8. Diagrama de casos de uso del sistema “(Paquete Abrir Plan)”

2.13 Descripción de Casos de Uso del Sistema.

CU- 1	Mostrar plan por OC.
Actor	Especialista de Centro de Balance
Descripción	Este CU permite al Especialista de CB realizar una búsqueda de las diferentes demandas de los OC, además facilita este proceso a través de un filtro que permite insertar determinados parámetros.
Referencia	R1, R1.1

Tabla 6. Breve descripción del caso de uso “Mostrar plan por OC”.

CU- 2	Mostrar plan de OC por año.
Actor	Especialista de Centro de Balance
Descripción	Este CU permite al Especialista de CB acceder al plan de demandas de los diferentes OC del presente año así como de años anteriores, para verificar que las peticiones actuales no excedan en gran medida las anteriores.
Referencia	R1, R1.2

Tabla 7. Breve descripción del caso de uso “Mostrar plan de OC por año”.

CU- 3	Consultar inventario útil.
Actor	Especialista de Centro de Balance
Descripción	El CU inicia cuando el Especialista de CB desea conocer el estado del inventario útil existente en almacén.
Referencia	R1, R1.3

Tabla 8. Breve descripción del caso de uso “Consultar inventario útil”.

CU- 4	Consultar anticipos.
Actor	Especialista de Centro de Balance
Descripción	El CU inicia cuando el Especialista de CB desea conocer el anticipo que le ha sido solicitado por parte de cada OC.
Referencia	R1, R1.4

Tabla 9. Breve descripción del caso de uso “Consultar anticipos”.

CU- 5	Consultar recursos en contratación.
Actor	Especialista de Centro de Balance
Descripción	El CU inicia cuando el Especialista de CB desea conocer el estado de los contratos efectuados.
Referencia	R1, R1.5

Tabla 10. Breve descripción del caso de uso “Consultar recursos en contratación”.

CU- 6	Aprobar plan del CB.
Actor	Jefe de Centro de Balance
Descripción	El CU se inicia cuando el Jefe de CB decide aprobar el plan del CB, para su posterior envío a la DE.
Referencia	R2

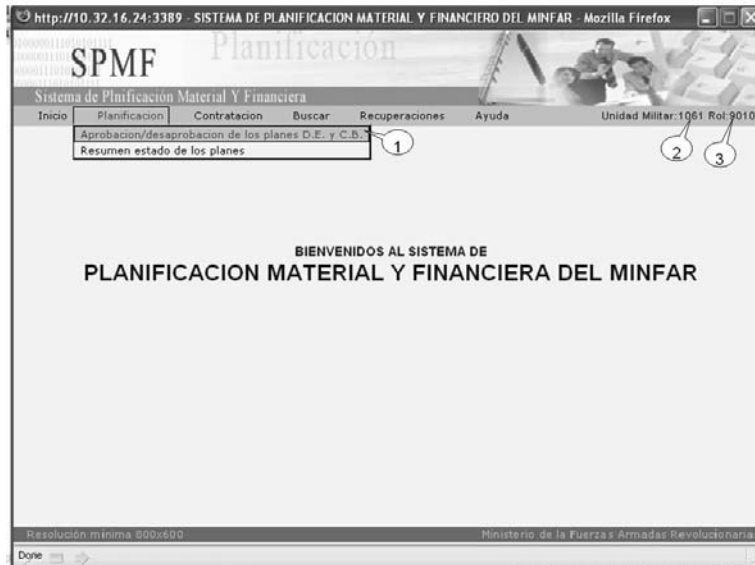
Tabla 11. Breve descripción del caso de uso “Aprobar plan del CB”.

CU- 7	Abrir Plan
Actor	Especialista de Centro de Balance
Descripción	El CU inicia cuando la DE aprueba el plan del CB, posteriormente se efectúa la apertura del mismo.
Referencia	R3, R3.1, R3.2, R3.3

Tabla 12. Breve descripción del caso de uso “Abrir Plan”.

Caso de uso	Mostrar plan por OC.
Actores:	Especialista de Centro de Balance (inicia).
Propósito:	Mostrar el plan de demandas de los diferentes Órganos Consumidores.
Resumen	El CU inicia cuando el Especialista de CB desea examinar las demandas de los diferentes OC analizando las peticiones de los mismos. El CU termina, una vez que el Especialista de CB tiene conocimiento total del plan de cada OC.
Precondiciones:	Existir el plan de demandas del OC.
Tipo:	Real y expandido
Responsabilidades:	R1, R1.1
Casos de uso relacionados:	No tiene

Interfaz I

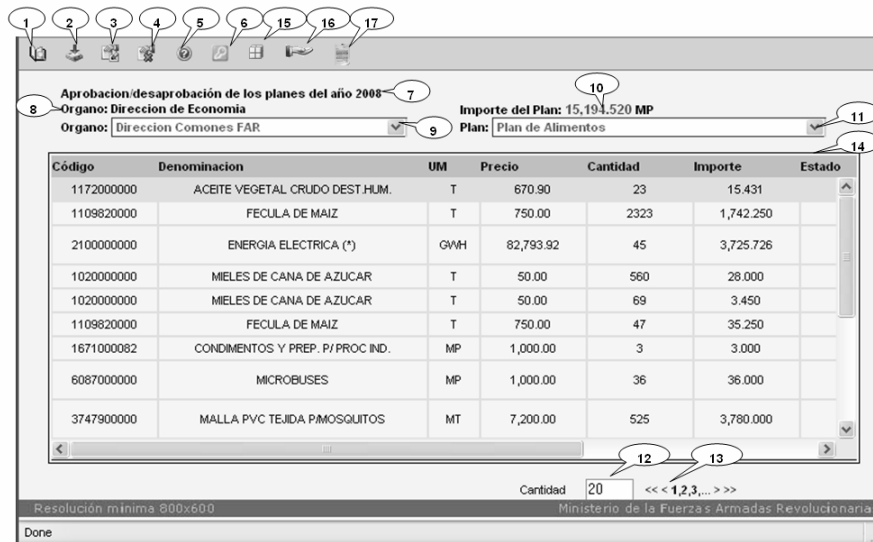


(1): nombre: (menú que permite revisar los planes de demandas)

(2): nombre: (label que muestra la unidad militar:7096)

(3): nombre: (label que muestra el rol:9010)

Interfaz II



(1): nombre: buscar (button que permite Realizar una búsqueda por parámetros de los datos del plan)

(2): nombre: cargar (button que permite cargar los datos del plan seleccionado)

- (3): **nombre: aprobar** (button que permite aprobar el plan seleccionado)
- (4): **nombre: desaprobar** (button que permite desaprobar el plan seleccionado)
- (5): **nombre : ayuda** (button que permite obtener ayuda del sistema)
- (6) **nombre: cerrarsesion** (button que permite cerrar sesión y salir del sistema)
- (7) **nombre: añot** (label que muestra el año de trabajo)
- (8) **nombre: oentrante** (órgano que entró)
- (9): **nombre: organos** (select con los nombres de los diferentes OC)
- (10): **nombre: importetot** (label que muestra el importe total del plan)
- (11): **nombre: plan** (select con los planes específicos)
- (12): **nombre: cantidad** (textfield que muestra la cantidad de elementos)
- (13): **nombre:** (muestra la cantidad de páginas)
- (14): **nombre: mygrid** (grid que muestra el plan que se selecciona)
- (15): **nombre: inventario** (button que permite consultar el Inventario Útil)
- (16): **nombre: inventario** (button que permite consultar el Anticipo)
- (17): **nombre: inventario** (button que permite consultar los Recursos en Contratación)

Curso normal de eventos para el caso de uso

Acción del actor	Respuesta del sistema
1. Seleccionar en el menú desplegable de la Interfaz I, la opción que permite consultar la Interfaz II, para buscar el plan de un OC determinado.	2. Mostrar la Interfaz II.
3. Seleccionar el OC (9) y el Plan (11) que desea examinar.	
4. Pulsar el botón “Cargar” (2).	5. Cargar el plan seleccionado (14). Mostrar los datos correspondientes a cada campo.


<p>6. Si desea realizar una búsqueda por parámetros de los datos de un plan ir a Sección Buscar (1).</p>	
<p>Sección Buscar</p>	
<p>Interfaz III</p>  <p>The screenshot shows a window titled 'Seleccionar recurso a buscar.....'. It contains several dropdown menus: 'Centro Balance' (1), 'Codigo' (2), 'Destino' (3), 'Actividad' (4), 'Denominacion' (5), and 'Sector - Rama' (6). There is a close button (7) in the top right and an 'Aceptar' button (8) at the bottom right.</p>	
<p>(1): nombre: cbalance (select con los nombres de los CB) (2): nombre: codigo (select con los códigos de los productos) (3): nombre : destino (select con los destinos u objetivos) (4): nombre: actividad (select con las actividades) (5): nombre: denominacion (select con los nombres de los productos) (6): nombre: destinof (select con los sectores-ramas o destinos finales) (7): nombre: close (permite cerrar la ventana) (8): nombre: Aceptar (submit)</p>	
<p>1. Pulsar en la Interfaz II el botón “Realizar una búsqueda por parámetros de los datos del plan” (1).</p>	<p>2. Mostrar la Interfaz III que permite realizar una búsqueda por parámetros de los datos de un plan.</p>
<p>3. Seleccionar los parámetros a filtrar.</p>	
<p>4. Pulsar el botón “Aceptar”.(8)</p>	<p>5. Mostrar el plan con los parámetros en la Interfaz II (14).</p>

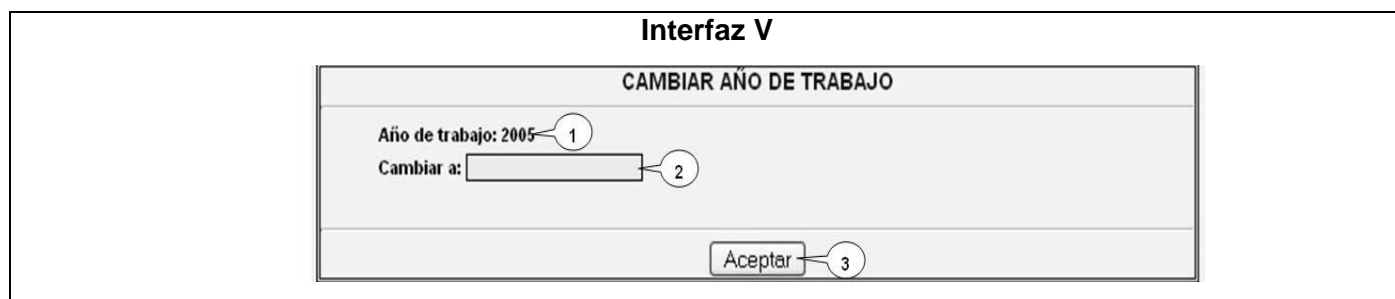
Tabla 13. Descripción Textual del caso de uso “Mostrar plan por OC.”

CAPÍTULO 2: MODELACIÓN DE LA PROPUESTA DE SOLUCIÓN

Caso de uso	Mostrar plan de OC por año.
Actores:	Especialista de Centro de Balance (inicia).
Propósito:	Mostrar el plan de demandas de años anteriores de los Órganos Consumidores.
Resumen	El CU cuando el Especialista de CB necesita conocer las demandas de los diferentes OC de años anteriores verificando que las peticiones actuales no excedan en gran medida las anteriores. El CU termina una vez que el Especialista de CB ha realizado la verificación
Precondiciones:	Existir registros de planes de demandas de OC de años anteriores.
Tipo:	Real y expandido.
Responsabilidades:	R1, R1.2
Casos de uso relacionados:	No tiene

Interfaz IV





(1): nombre: añot (label que muestra el año de trabajo)

(2): nombre: anot (text para introducir el año de trabajo)

(3): nombre: Aceptar (button para aceptar el año ingresado)

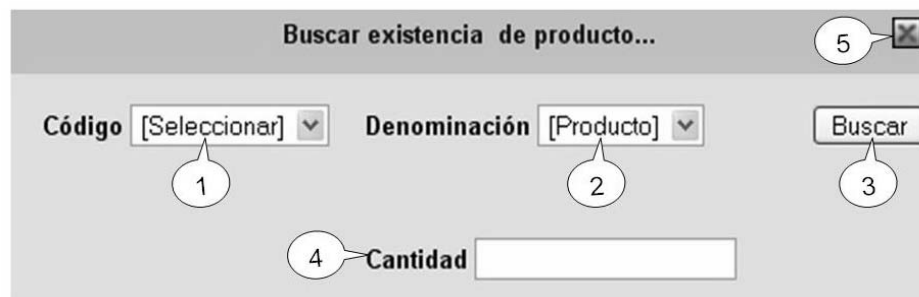
Curso normal de eventos para el caso de uso

Acción del actor	Respuesta del sistema
1. Seleccionar en el menú desplegable de la Interfaz IV, la opción que permite consultar la Interfaz V, para cambiar el año de trabajo.	2. Mostrar la Interfaz V.
3. Insertar en (2) el año.	
4. Pulsar el botón “Aceptar” (3).	5. Mostrar en (1) el año.
6. Seleccionar en el menú desplegable de la Interfaz I, la opción que permite consultar la Interfaz II, para examinar el plan de un Órgano Consumidor determinado.	
Repetir el mismo proceso del caso de uso Examinar plan de Órgano Consumidor .	

Tabla 14. Descripción Textual del caso de uso “Mostrar plan de OC por año.”

Caso de uso	Consultar inventario útil.
Actores:	Especialista de Centro de Balance (inicia).
Propósito:	Verificar el inventario útil en almacén.
Resumen	El CU se inicia cuando el Especialista de CB desea conocer el estado del inventario útil existente en almacén y termina cuando el mismo tiene noción de la existencia de los productos.
Precondiciones:	Existir registros actualizados del inventario útil en almacén.
Tipo:	Real y expandido.
Responsabilidades:	R1, R1.3
Casos de uso relacionados:	No tiene

Interfaz VI



(1): nombre: codigo (select con los códigos de los productos)

(2): nombre: denominacion (select con los nombres de los productos)

(3): nombre: buscar(submit)

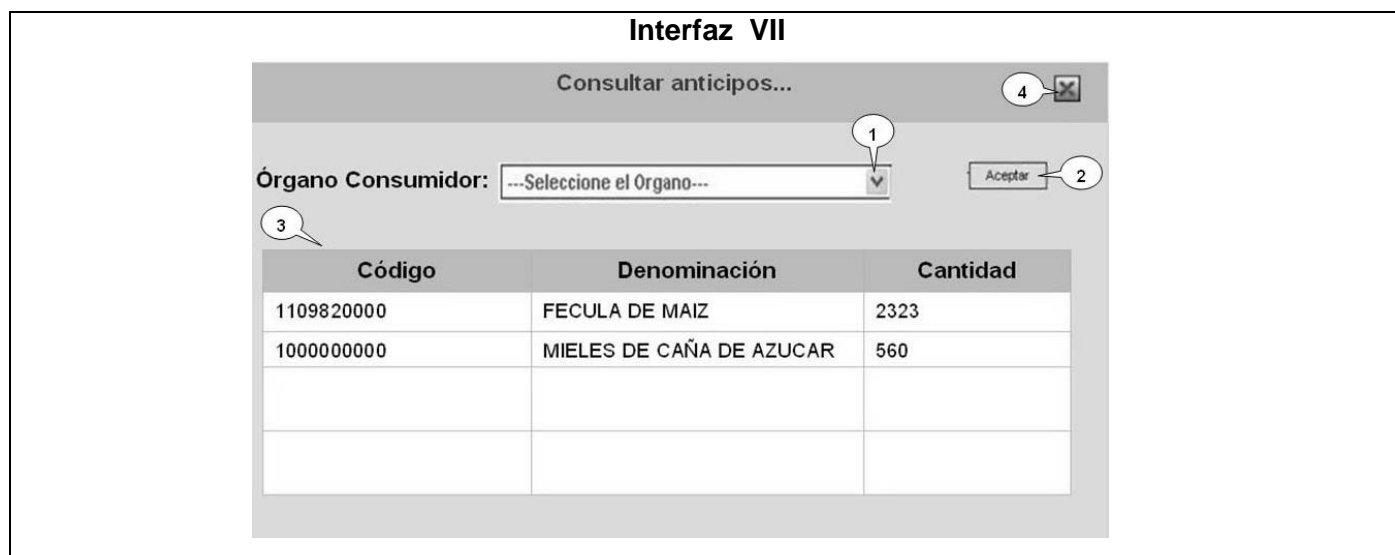
(4): nombre: cantidad (textfield que muestra la cantidad del producto)

(5): nombre: close (permite cerrar la ventana)

Curso normal de eventos para el caso de uso	
Acción del actor	Respuesta del sistema
1. Seleccionar en (15) de la Interfaz II, la opción que muestra la Interfaz VI, para consultar la existencia de producto en almacén.	2. Mostrar la Interfaz VI.
3. Seleccionar en (1) o en (2) el código del producto o su denominación, respectivamente.	4. Mostrar en (1) o en (2) según lo seleccionado.
5. Pulsar el botón “Buscar” (3).	6. Mostrar en (4) la cantidad del producto existente.

Tabla 15. Descripción Textual del caso de uso “Consultar inventario útil.”

Caso de uso	Consultar anticipo.
Actores:	Especialista de Centro de Balance (inicia).
Propósito:	Verificar Anticipo.
Resumen	El caso de uso inicia cuando el Especialista de CB desea conocer el anticipo que le ha sido solicitado por parte de cada OC, y termina cuando tiene conocimiento del mismo.
Precondiciones:	Existir registros actualizados de los anticipos en almacén.
Tipo:	Real y expandido.
Responsabilidades:	R1, R1.4
Casos de uso relacionados:	No tiene



- (1): nombre: **organos** (select con los nombres de los diferentes OC)
- (2): nombre: **Aceptar** (submit)
- (3): nombre: **mygrid** (grid que muestra los anticipos correspondientes al OC que se selecciona)
- (4): nombre: **close** (permite cerrar la ventana)

Curso normal de eventos para el caso de uso

Acción del actor	Respuesta del sistema
1. Seleccionar en (16) de la Interfaz II, la opción que muestra la Interfaz VII, para consultar el anticipo.	2. Mostrar la Interfaz VII.
3. Seleccionar en (1) el OC.	
4. Pulsar el botón "Aceptar" (2).	5. Mostrar en (3) los anticipos.

Tabla 16. Descripción Textual del caso de uso "Consultar anticipos."

Caso de uso	Consultar recursos en contratación.
Actores:	Especialista de Centro de Balance (inicia).
Propósito:	Verificar los recursos en contratación.
Resumen	El CU inicia cuando el Especialista de CB desea conocer el estado de los contratos efectuados.
Precondiciones:	Existir registros actualizados de los recursos contratados.

CAPÍTULO 2: MODELACIÓN DE LA PROPUESTA DE SOLUCIÓN

Tipo:	Real y expandido.
Responsabilidades:	R1, R1.5
Casos de uso relacionados:	No tiene

Interfaz VIII

- (1): **nombre: desde** (textfield que permite introducir fecha)
- (2): **nombre: hasta** (textfield que permite introducir fecha)
- (3): **nombre: contrato** (select con los números de los contratos efectuados)
- (4): **nombre: estado** (textfield que muestra el estado del contrato)
- (5): **nombre: provpais** (textfield que muestra el país proveedor del contrato)
- (6): **nombre: close** (permite cerrar la ventana)

Curso normal de eventos para el caso de uso

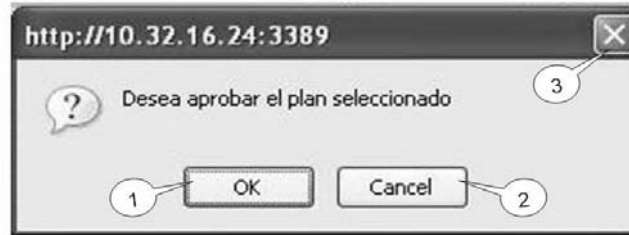
Acción del actor	Respuesta del sistema
1. Seleccionar en (17) de la Interfaz VIII, la opción que muestra la Interfaz XI, para consultar los recursos contratados.	2. Mostrar la Interfaz VIII.
3. Introducir en (1) la fecha a partir de la cual se desea conocer el estado de los contratos.	
3. Introducir en (2) la fecha hasta la cual se desea conocer el estado de los contratos.	
4. Seleccionar en (3) el número de contrato que se desea conocer.	5. Mostrar en (4) el estado del contrato.
	6. Mostrar en (5) el país proveedor del contrato

Tabla 17. Descripción Textual del caso de uso “Consultar recursos en contratación.”

Caso de uso	Aprobar plan del CB.
Actores:	Jefe de Centro de Balance (inicia).
Propósito:	Aprobar el plan del CB.
Resumen	El CU se inicia cuando el Jefe de CB decide aprobar el plan de su CB. El CU termina una vez que el plan ha sido enviado a la DE.
Precondiciones:	Existir el plan del CB.
Poscondiciones:	El plan del CB queda aprobado.
Tipo:	Real y expandido
Responsabilidades:	R2
Casos de uso relacionados:	No tiene
Curso normal de eventos para el caso de uso	
Acción del actor	Respuesta del sistema
1. Seleccionar en el menú desplegable de la Interfaz I, la opción que permite consultar la Interfaz II, para examinar el plan de un OC determinado.	2. Mostrar la Interfaz II.
3. Seleccionar el OC (9) y el Plan (11) que desea examinar.	
4. Pulsar el botón “Cargar” (2).	5. Cargar el plan seleccionado (14). Mostrar los datos correspondientes a cada campo.
Si desea aprobar el plan ir a Sección 1.	
Si desea desaprobar el plan ir a Sección 2.	

Sección 1

Mensaje 1



Mensaje 2



Acción del actor

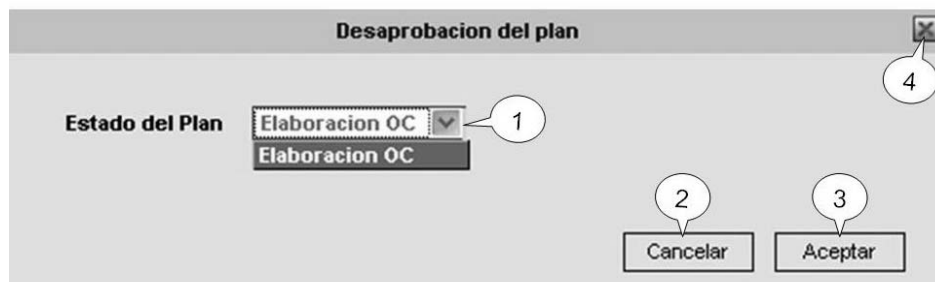
- 1. Pulsa en la Interfaz II el botón “Aprobar el plan seleccionado” (3).
- 3. Pulsa el botón “OK” (1).

Respuesta del sistema

- 2. Muestra el Mensaje 1: “¿Desea aprobar el plan seleccionado?”.
- 4. Muestra el Mensaje 2 de confirmación: “Plan aprobado satisfactoriamente”.

Sección 2

Interfaz IX



(1):nombre: estadoplan (select que permite cambiar el estado del plan)

(2):nombre: Cancelar (submit)

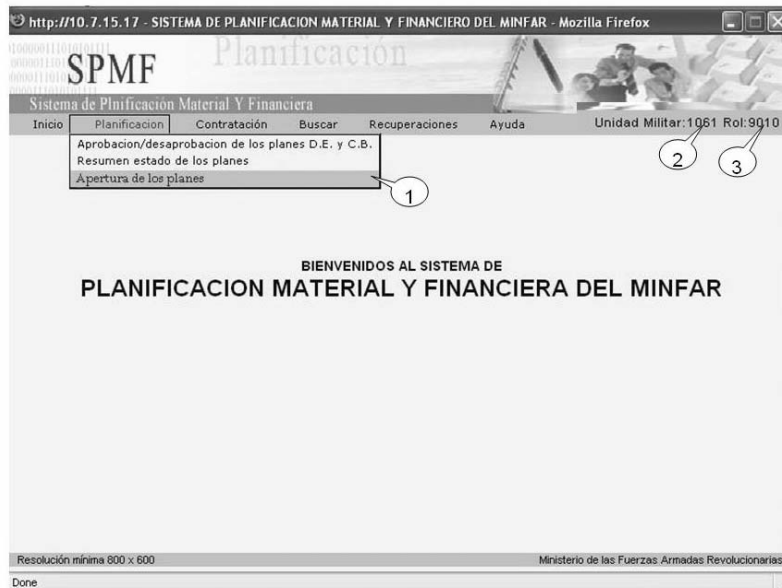
CAPÍTULO 2: MODELACIÓN DE LA PROPUESTA DE SOLUCIÓN

(3): nombre: Aceptar (submit)	
(4): nombre: close (button permite cerrar la ventana)	
Acción del actor	Respuesta del sistema
1. Pulsa el botón “Desaprobar el plan seleccionado”.(4)	2. Muestra la Interfaz IX.
3. Selecciona el estado al que desea pasar el plan.	
4. Pulsa el botón “Aceptar”(3)	6. Muestra un mensaje de confirmación con el estado al que pasó el plan.

Tabla 18. Descripción Textual del caso de uso “Aprobar plan del CB.”

Caso de uso	Abrir Plan.
Actores:	Especialista de Centro de Balance (inicia).
Propósito:	Realizar la apertura del plan y relacionar recurso con su respectivo proveedor.
Resumen	El CU inicia cuando el Especialista desea realizar la apertura del plan, posteriormente relaciona cada recurso con su respectivo proveedor. El CU termina cuando el Especialista ha hecho las distribuciones correspondientes.
Precondiciones:	Existir un registro de genéricos con los surtidos asociados y sus códigos. Existir un registro de los diferentes proveedores.
Poscondiciones:	Quedar relacionado cada genérico solicitado con el surtido que se necesita específicamente, así como la vía de obtención.
Tipo:	Real y expandido
Responsabilidades:	R3, R3.1, R3.2, R3.3
Casos de uso relacionados:	No tiene

Interfaz X

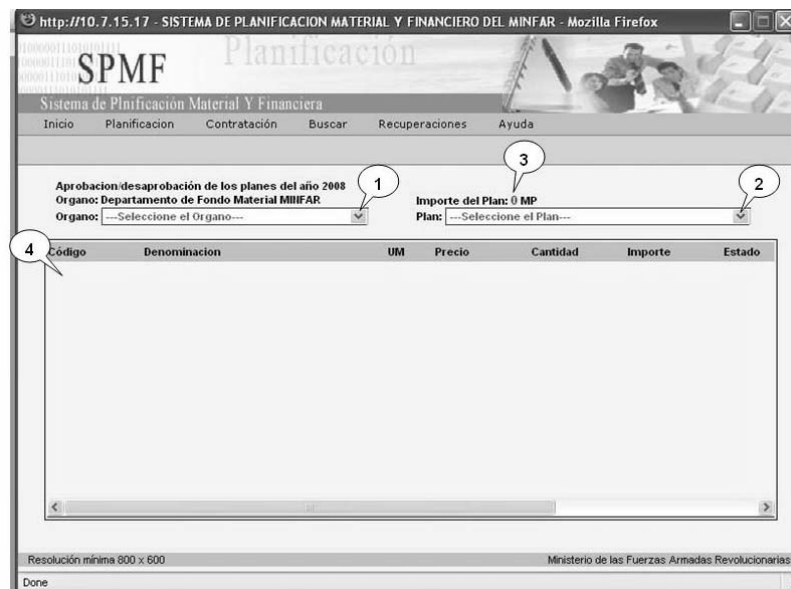


(1): nombre: (menú que permite realizar la apertura de los planes)

(2): nombre: (label que muestra la unidad militar:7096)

(3): nombre: (label que muestra el rol:9010)

Interfaz XI



- (1): nombre: **organos** (select con los nombres de los diferentes OC)
- (2): nombre: **plan** (select con los planes específicos)
- (3): nombre: **importetot** (label que muestra el importe total del plan)
- (4): nombre: **mygrid** (grid que muestra el plan que se selecciona)

Interfaz XII

Apertura del plan... 1 12

Importe: 100.000 MP

Surtido: [Seleccionar] Precio: [] Forma de Adquisición: [Seleccionar]

Cantidad: [] Importe: [] Tipo de Crédito: [Seleccionar]

Pais o Empresa: [Seleccionar] Aceptar Cancelar

Código	Productos	Valor	Divisas Líquidas				
			CHINA	RUSIA	CANADA	COMPRA EN FRONTERAS	OACE SIME MINAZ
644.1.00	PPA de UAZ	1,335,40		1,335,40			
644.101	PPA de motor de UAZ	105,30		105,30			
644.101	PPA de GACELA	188,40				188,40	
644.4.00	PPA de POLAKITO	83,70				83,70	

- (1): nombre: **importetot** (label que muestra el importe total del pedido por genérico)
- (2): nombre: **surtido** (select con los códigos de los surtidos)
- (3): nombre: **precio** (textfield que muestra el precio del surtido)
- (4): nombre: **cantidad** (textfield que permite insertar la cantidad)
- (5): nombre: **importe** (textfield que muestra el importe “cantidad * precio”)
- (6): nombre: **fadquisicion** (select con las formas de adquisición)
- (7): nombre: **tcredito** (select con los tipos de créditos)
- (8): nombre: **surtido** (select con los nombres de países o empresas)
- (9): nombre: **Cancelar** (submit)

(10): nombre: Aceptar (submit)	
(11): nombre: mygrid (grid que muestra los datos correspondientes con la apertura del plan)	
(12): nombre: close (permite cerrar la ventana)	
Curso normal de eventos para el caso de uso	
Acción del actor	Respuesta del sistema
1. Seleccionar en el menú desplegable de la Interfaz X, la opción que permite consultar la Interfaz XI, que muestra el plan que se desea abrir.	2. Mostrar la Interfaz XI.
3. Seleccionar el Órgano Consumidor (9) y el Plan (11) que desea examinar.	
4. Pulsar el botón "Cargar" (2).	5. Cargar el plan seleccionado (14). Mostrar los datos correspondientes a cada campo.
6. Seleccionar el recurso (genérico) del cual desea ver todos sus surtidos.	7. Mostrar la Interfaz XII.
8. Seleccionar en (2) el surtido que se desea.	9. Mostrar el precio de ese surtido (3)
10. Insertar la cantidad que se solicita (4).	11. Mostrar en (5) el importe.
12. Especificar en (6) la forma de adquisición: Crédito o Divisa Líquida.	
Si se escoge Crédito ir a Sección 1.	
Si se escoge Divisa Líquida ir a Sección 2.	

Sección 1	
Acción del actor	Respuesta del sistema
1. Especificar en (7) el tipo de crédito: Gubernamental o Empresarial.	2. Habilitar en (8) países o empresas según lo seleccionado en (7).
3. Seleccionar en (8).	
4. Pulsar el botón "Aceptar" (9).	5. Mostrar en (11) los datos insertados.
Sección 2	
Acción del actor	Respuesta del sistema
1. Especificar en (7) si es por OACE o Compra en Frontera.	2. Habilitar en (8) OACE o países según lo seleccionado en (7).
3. Seleccionar en (8).	
4. Pulsar el botón "Aceptar" (9).	5. Mostrar en (11) los datos insertados.

Tabla 19. Descripción Textual del caso de uso "Abrir Plan."

2.14 Conclusiones.

Este capítulo presenta una estrategia para abordar el modelado del negocio, y el análisis de requisitos, los cuales se obtienen de forma sencilla, a partir del modelo del negocio basado en el uso de diagramas de actividades UML.

A la vez que se realiza el modelado del negocio y de los requisitos, las reglas del negocio de la organización se recogen en un glosario, en forma de especificación de las actividades y de los casos de uso asociados, así como de las clases y objetos de información. Esto permite mantener las correspondientes relaciones de trazabilidad entre los diferentes artefactos del modelado.

En este trabajo se ha expuesto cómo el modelado del negocio puede facilitar la identificación de los requisitos tanto funcionales como no funcionales del sistema. Además, el hecho de que tales requisitos surjan de la descripción de los procesos del negocio, y que éstos sean el resultado del análisis de los objetivos de la organización, posibilita que los requisitos del sistema sean validados y verificados contra los objetivos del negocio.

CAPÍTULO 3: ANÁLISIS Y DISEÑO DEL SISTEMA

3.1 Introducción.

Los sistemas de información se desarrollan con diferentes propósitos, los cuales dependen de las necesidades de las empresas. Por ejemplo los sistemas de procesamiento de datos, los sistemas de información para la administración, y los sistemas de apoyo para la toma de decisiones, son diferentes tipos de sistemas de información computarizada. Estos se analizan y se diseñan mediante la aplicación de los conceptos y las técnicas del diseño y del análisis de sistemas

El Diagrama de Clases es el diagrama principal de diseño y análisis para un sistema. En él, la estructura de clases del sistema se especifica, con relaciones entre clases y estructuras de herencia. Durante el análisis del sistema, el diagrama se desarrolla buscando una solución ideal. Durante el diseño, se usa el mismo diagrama, y se modifica para satisfacer los detalles de las implementaciones.

3.2 Modelo de Análisis.

Analizar los requisitos en la forma de un modelo de análisis es importante por varios motivos.

Un modelo de análisis como resultado ofrece una especificación más precisa de los requisitos en la que se tienen como resultado de la captura de requisitos, incluyendo el modelo de casos de uso.

Un modelo de análisis se describe utilizando el lenguaje de los desarrolladores y puede por tanto introducir un mayor formalismo y ser utilizado para razonar sobre los funcionamientos internos del sistema.

Un modelo de análisis estructura los requisitos de un modo que facilita su comprensión, su preparación, su modificación, y en general, su mantenimiento.

Un modelo de análisis puede considerarse como una primera aproximación al modelo de diseño (aunque es un modelo por sí mismo), y es por tanto una entrada fundamental cuando se da forma al sistema en el diseño y en la implementación. Esto se debe a que se debería ser mantenible el sistema en su conjunto, y no solo la descripción de sus requisitos. [24]

El modelo de análisis ayuda a refinar los requisitos, permite razonar sobre los aspectos internos del sistema, ofrece un mayor poder expresivo y una mayor formalización, ej. Las que nos proporcionan los diagramas.

3.3 Diagramas del Análisis.

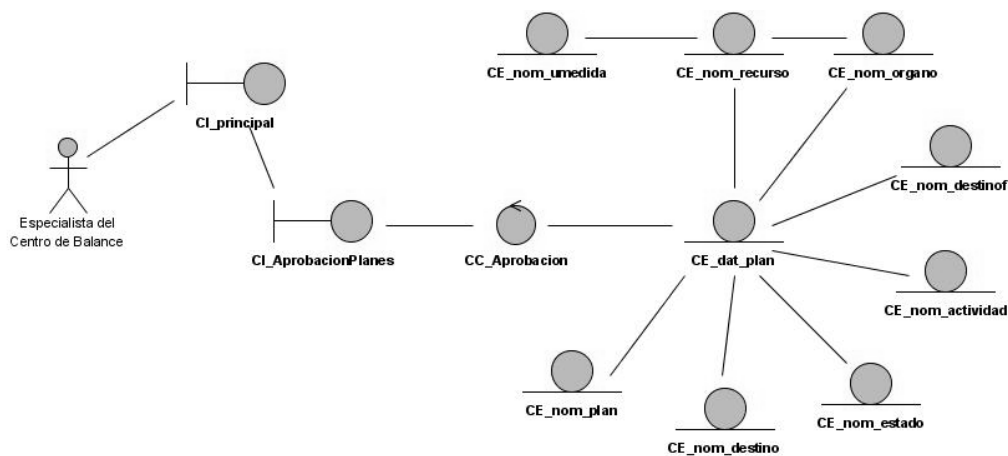


Figura 9. Diagrama de clases del análisis “Mostrar plan por OC (Paquete Gestionar Plan)”

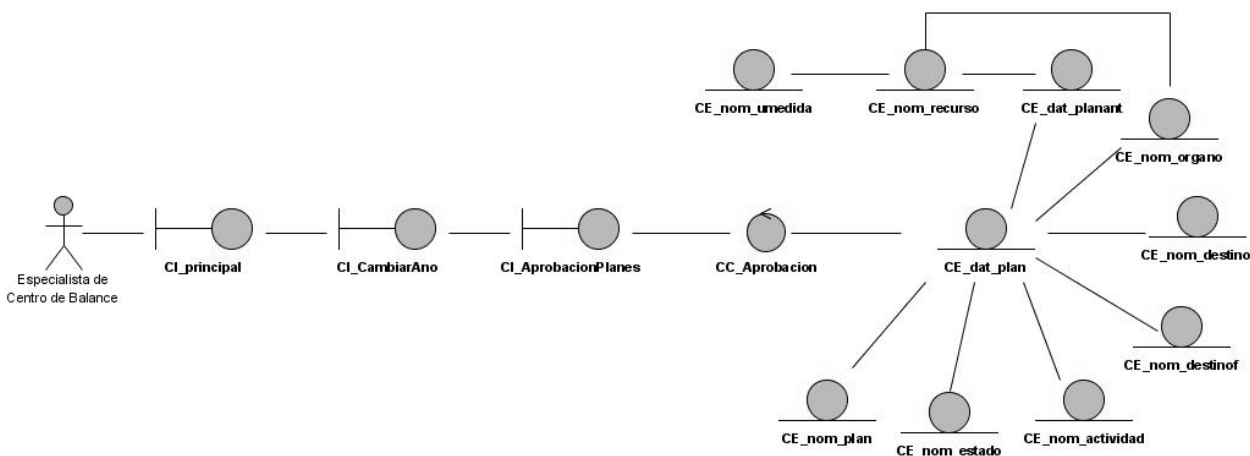


Figura 10. Diagrama de clases del análisis “Mostrar plan de OC por año (Paquete Gestionar Plan)”

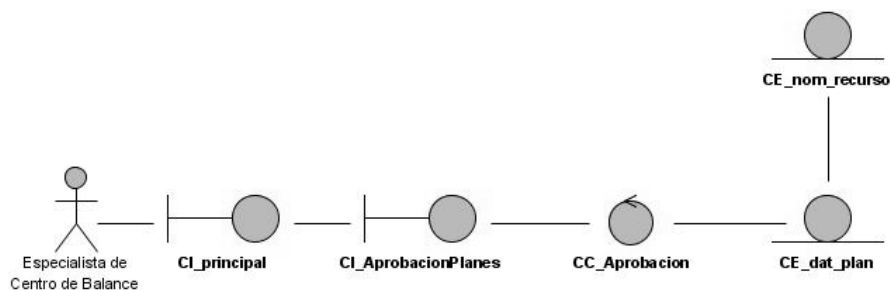


Figura 11. Diagrama de clases del análisis “Consultar inventario útil (Paquete Gestionar Plan)”

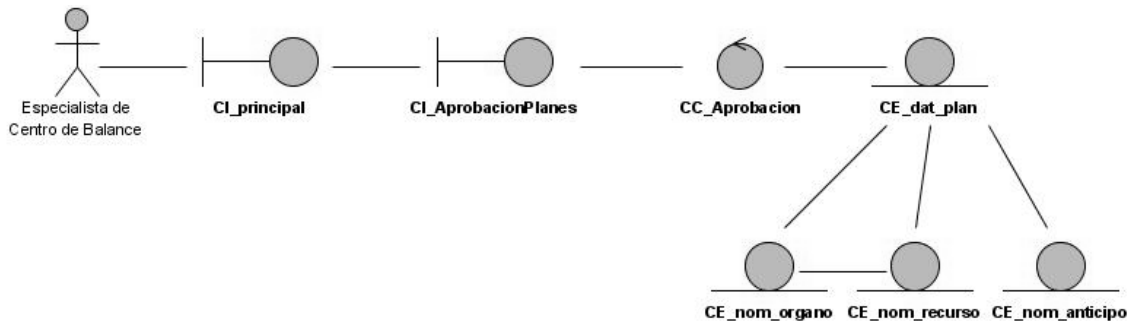


Figura 12. Diagrama de clases del análisis “Consultar anticipos (Paquete Gestionar Plan)”

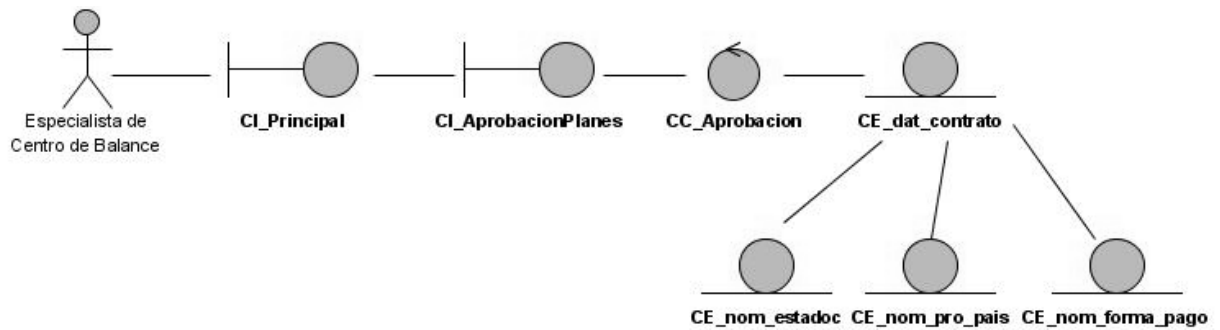


Figura 13. Diagrama de clases del análisis “Consultar recursos en contratación (Paquete Gestionar Plan)”

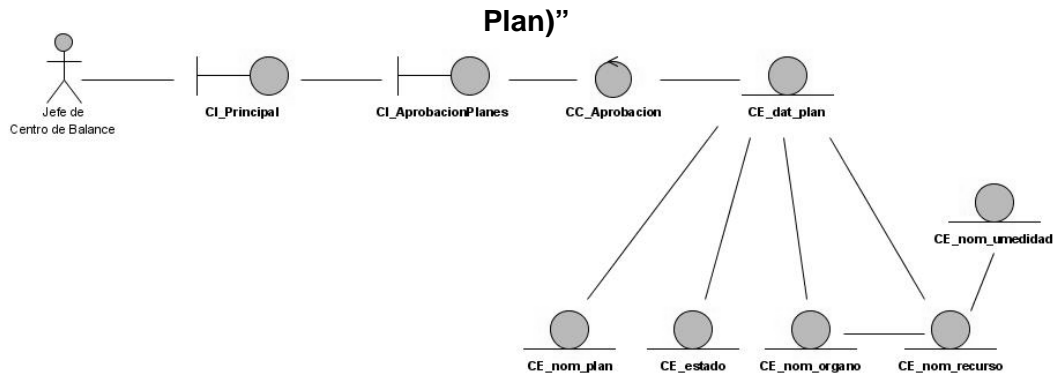


Figura 14. Diagrama de clases del análisis “Aprobar plan del CB (Paquete Aprobar plan del CB)”

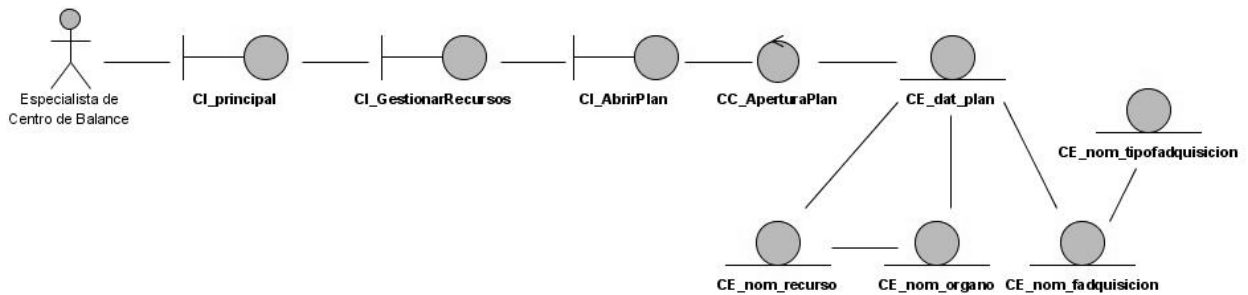


Figura 15. Diagrama de clases del análisis “Abrir Plan (Paquete Abrir Plan)”

3.4 Diagramas de Secuencia.

Es el diagrama que muestra las interacciones entre los objetos organizados en una secuencia temporal. En particular muestra los objetos participantes en la interacción y la secuencia de mensajes intercambiados.

Dentro del conjunto de mensajes representados dispuestos en una secuencia temporal, cada rol en la secuencia se muestra como una línea de vida, es decir, una línea vertical que representa el rol durante cierto plazo de tiempo, con la interacción completa. Los mensajes se muestran como flechas entre líneas de vida. Un diagrama de secuencia puede mostrar un escenario, es decir, una historia individual de transacción. Un uso de un diagrama de secuencia es mostrar la secuencia del comportamiento de un caso de uso. [14]

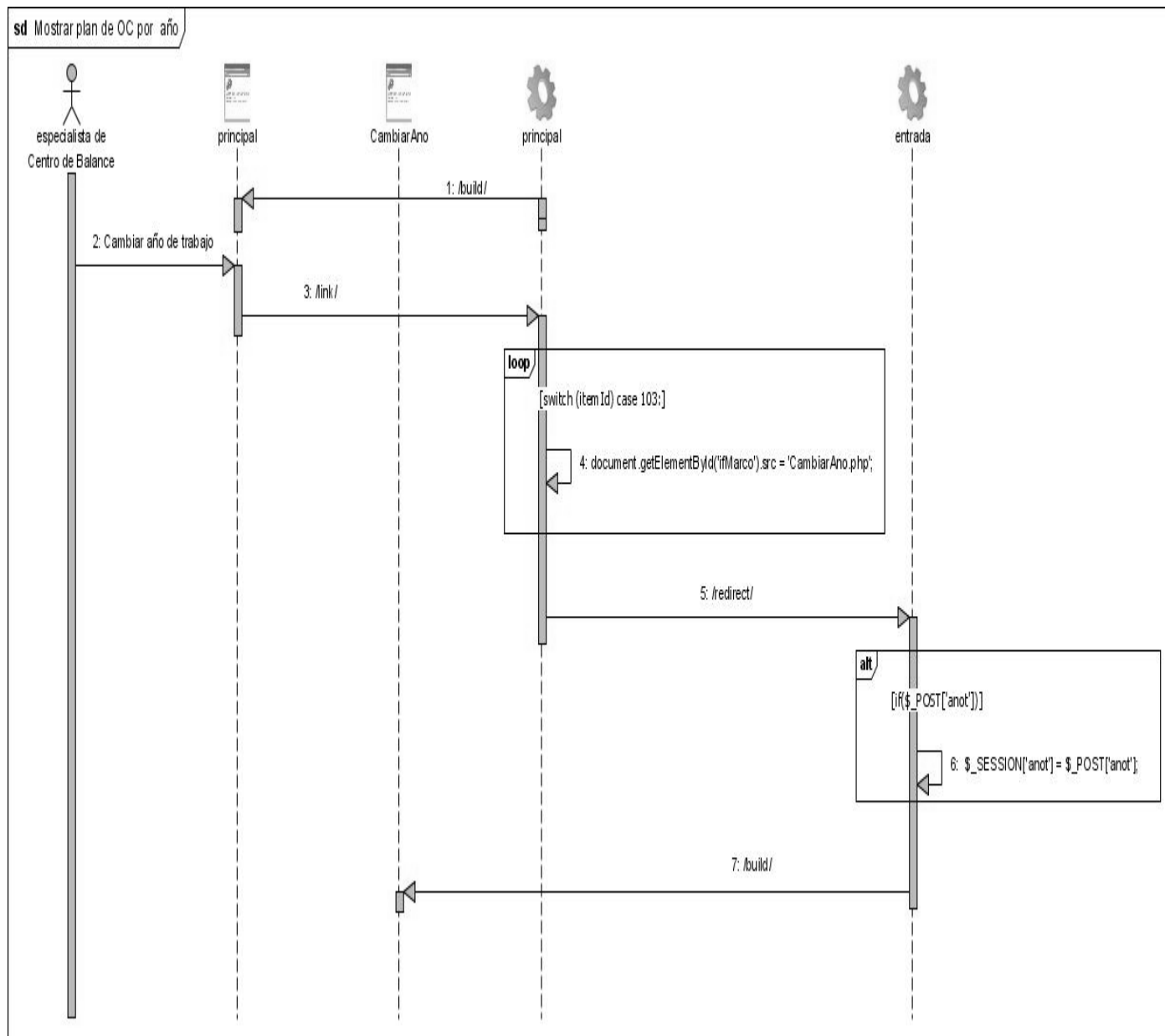


Figura 17. Diagrama de secuencia caso de uso “Mostrar plan de OC por año” realización construir página CambiarAño.

3.5 Mecanismos de Diseño.

Los mecanismos de diseño se modelan para comunicar la manera más óptima en que debe darse solución a problemas repetitivos en la aplicación. Aunque su desarrollo y mantenimiento es opcional, se recomienda su uso en entornos de desarrollo complejos.

Con su elaboración, se modelaría un conocimiento que ayudará tanto al desarrollo de la aplicación actual como a construcciones futuras, además de las labores de mantenimiento. En los mecanismos de diseño intervienen diversos elementos de la aplicación (clases, subsistemas). Un ejemplo típico de mecanismos son los patrones de diseño.

3.5.1 Mecanismo de Diseño de Acceso a Datos

Con el transcurso de los años buscar mecanismos para modelar el acceso a datos siempre ha sido un meta a lograr, es por eso que un mecanismo aplicado para resolver esta situación nunca será igual a otro definido. Muchos son los especialistas que buscan un modelo ideal que sirva para todos los casos y se pueda emplear sin pensarlo dos veces, pero es una realidad que el desarrollo de hoy en día es tan vertiginoso que permite buscar soluciones tan buenas como otras ya definidas, convirtiéndose a su vez en mecanismos aplicables bajo ciertas circunstancias. El acceso y la manipulación de los datos es algo realmente indispensable a la hora de desarrollar un sistema informático.

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

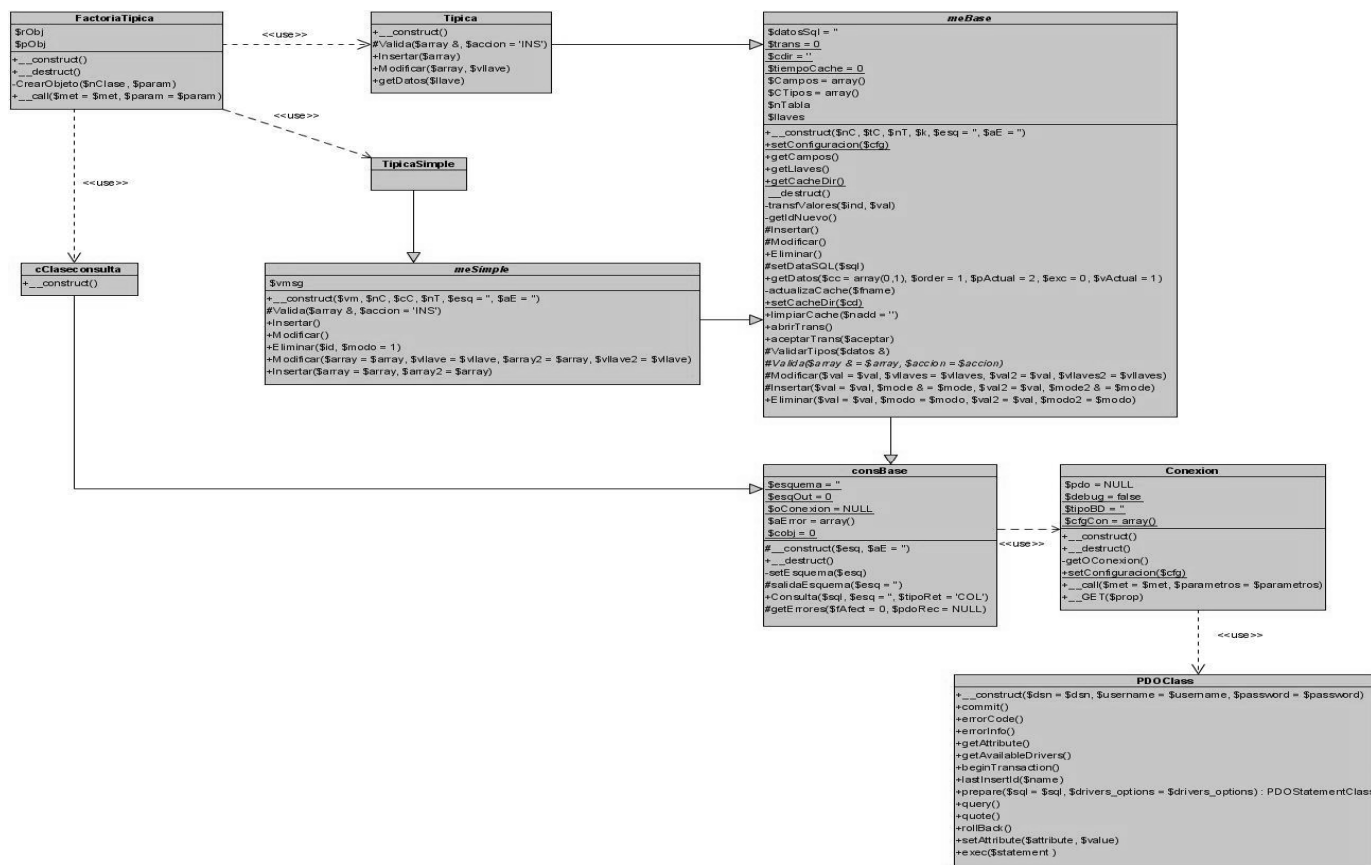


Figura 19. Diagrama general del “Mecanismo de Diseño de Acceso a Datos”.

En el diagrama representado se muestra la vista estática de la solución propuesta para manejar el acceso a datos del sistema. En él se encuentran una serie de clases las cuales cumplen con determinadas responsabilidades para asegurar un buen funcionamiento de este mecanismo.

Factoría Típica:

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.

Típicas:

Clase que representa a las clases típicas en general de la aplicación. Existe una típica para cada entidad de la base de datos. Para la implementación de esta clase se aplicará 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 meBase.

Típica Simple:

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 habían que codificar quedaron encapsuladas en la clase base de las mismas. Para la implementación de esta clase se aplicará 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

cClaseconsulta:

Clase que representa a las clases consultas en general de la aplicación. Existe una clase consulta para cada entidad de la base de datos. Hereda de la clase abstracta consBase.

meSimple:

Clase abstracta que heredera de meBase, y a 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:

Clase abstracta que hereda de consBase la operación de Consulta, es la base para el resto de las que implementan funcionalidades para el trabajo con las entidades del sistema a implementar. Implementa las operaciones básicas que pudieran realizarse a una entidad (insertar, eliminar y modificar).

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:

Clase 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:

Modelo de acceso a datos para php que brinda una capa de abstracción para el acceso a BD desde php.

3.5.2 Mecanismo de Diseño de Seguridad

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

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

Los sistemas realizan el control de la seguridad de manera semejante, usando un servicio Web encargado del control de los accesos, autenticación y registro de los eventos que ocurren, por lo cual se propone un mecanismo de diseño que sirva de manera general a todas las aplicaciones que usan dicho servicio, garantizando así los requerimientos necesarios para su correcto funcionamiento. Básicamente los servicios Web permiten que diferentes aplicaciones, realizadas con diferentes tecnologías, y ejecutándose en toda una variedad de entornos, puedan comunicarse e integrarse, lo cual es muy importante.

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

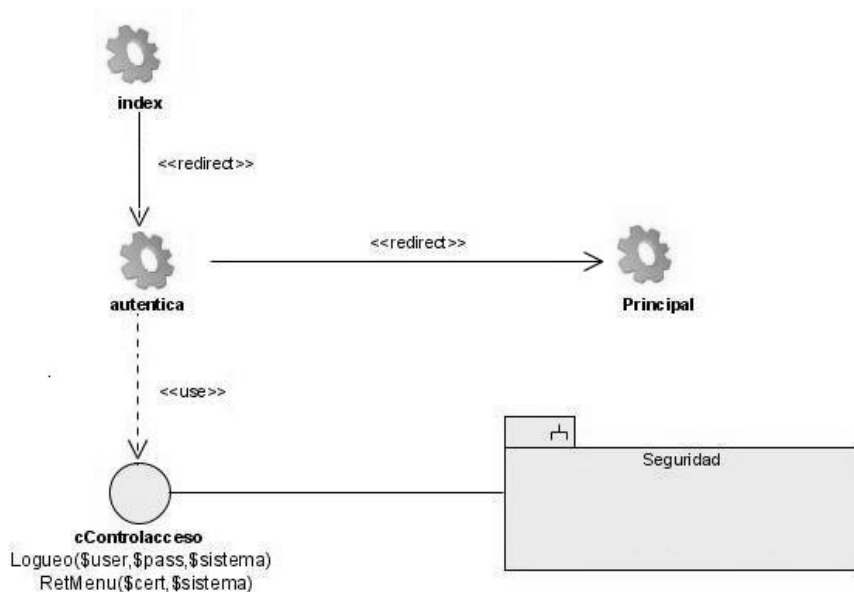


Figura 20. Diagrama general del “Mecanismo de Diseño de Seguridad”.

3.6 Estándares de Diseño.

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

Las páginas deben tener una cabecera (banner) representativa, un área de trabajo y una barra menú con las opciones, además tener una hoja de estilo en común para lograr la uniformidad, es decir, se trabajará con la familia de fuentes Arial, 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 y/o gris.

3.7 Modelo de Diseño.

El modelo de diseño es un modelo de objetos que describe la realización física de los casos de uso centrándose en cómo los requisitos funcionales y no funcionales, junto con otras restricciones relacionadas con el entorno de implementación, tienen impacto en el sistema a considerar. Además, el modelo de diseño sirve de abstracción de la implementación del sistema y es, de ese modo, utilizada como una entrada fundamental de las actividades de implementación. [24]

3.8 Diagramas del Diseño.

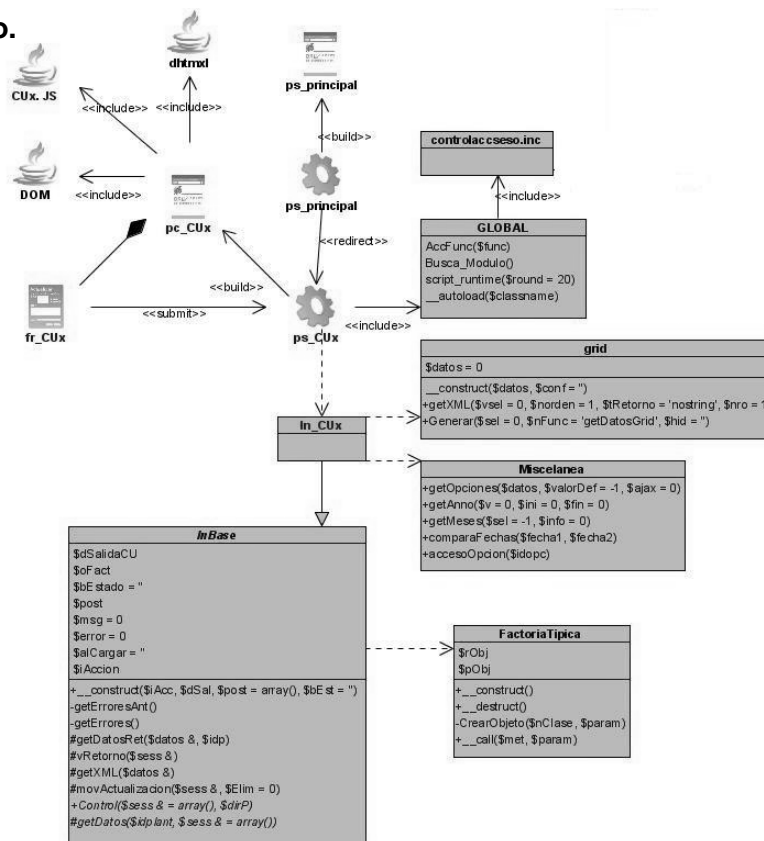


Figura 21. Diagrama de clases del diseño “Genérico”

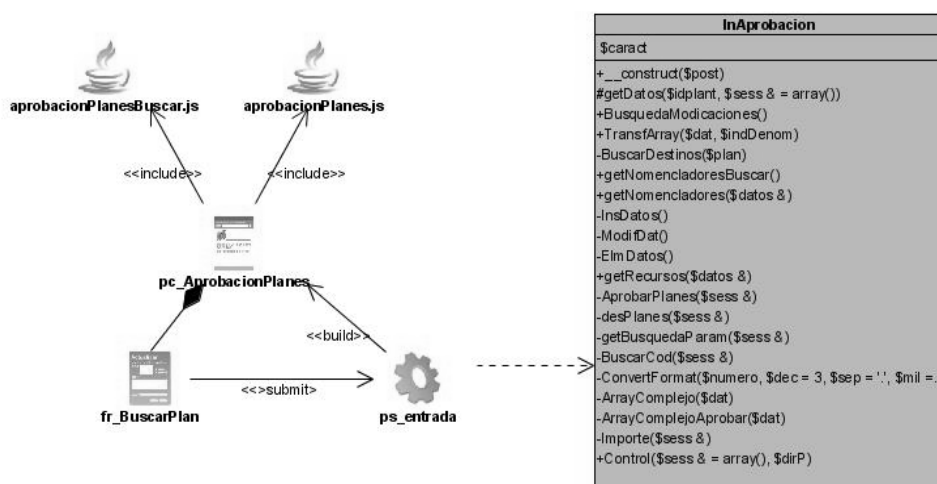


Figura 22. Diagrama de clases del diseño “Mostrar plan por OC (Paquete Gestionar Plan)”

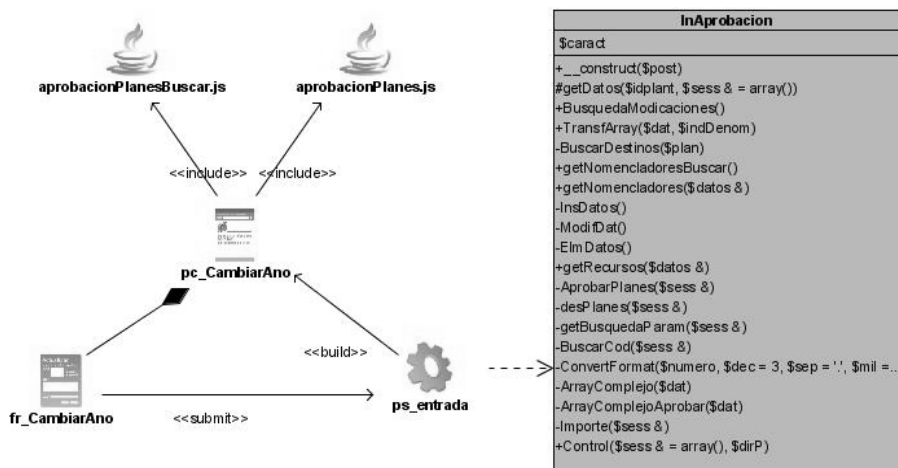


Figura 23. Diagrama de clases del diseño “Mostrar plan de OC por año (Paquete Gestionar Plan)”

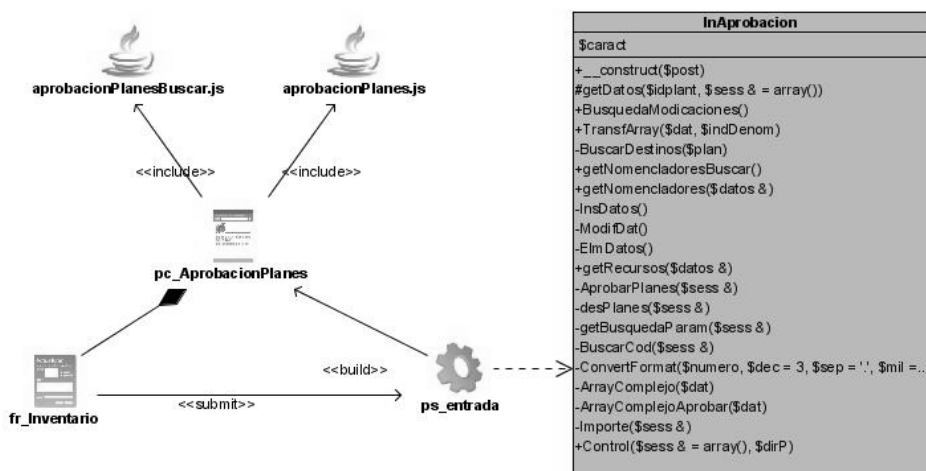


Figura 24. Diagrama de clases del diseño “Consultar inventario útil (Paquete Gestionar Plan)”

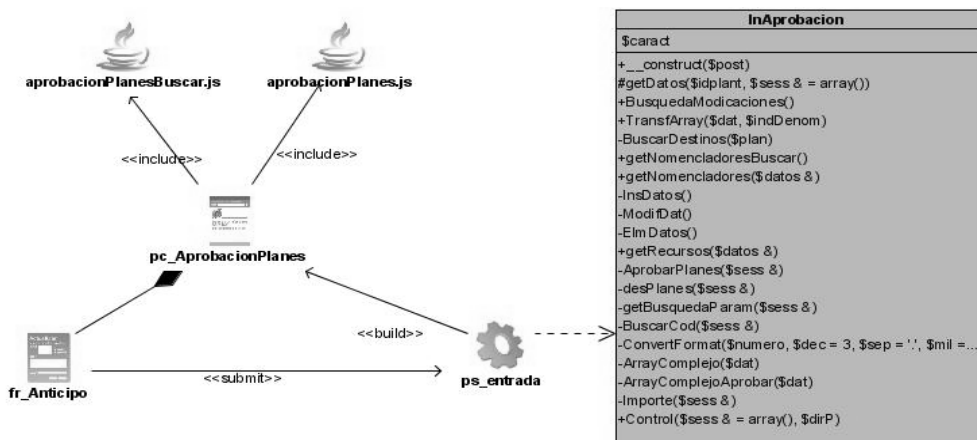


Figura 25. Diagrama de clases del diseño “Consultar anticipos (Paquete Gestionar Plan)”

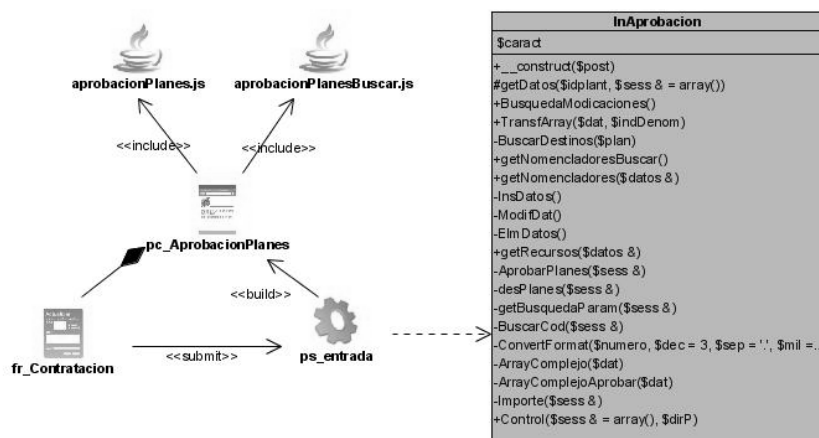


Figura 26. Diagrama de clases del diseño “Consultar recursos en contratación (Paquete Gestionar Plan)”

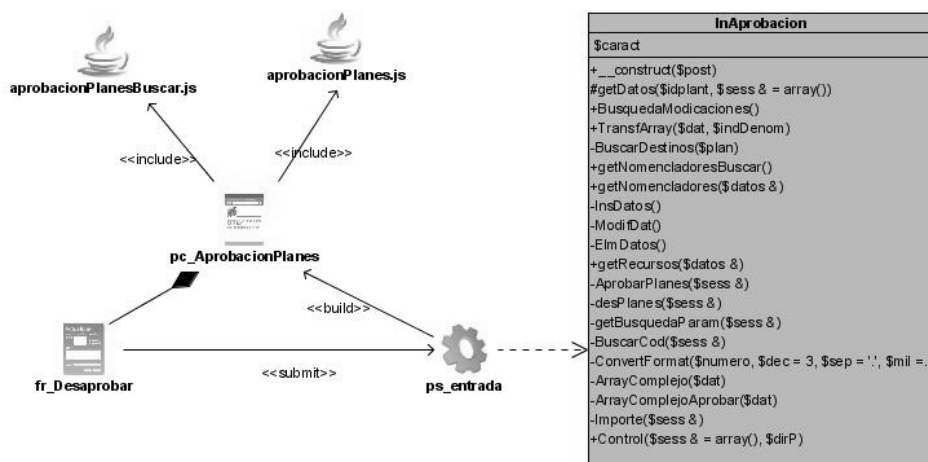


Figura 27. Diagrama de clases del diseño “Aprobar plan del CB (Paquete Gestionar Plan)”

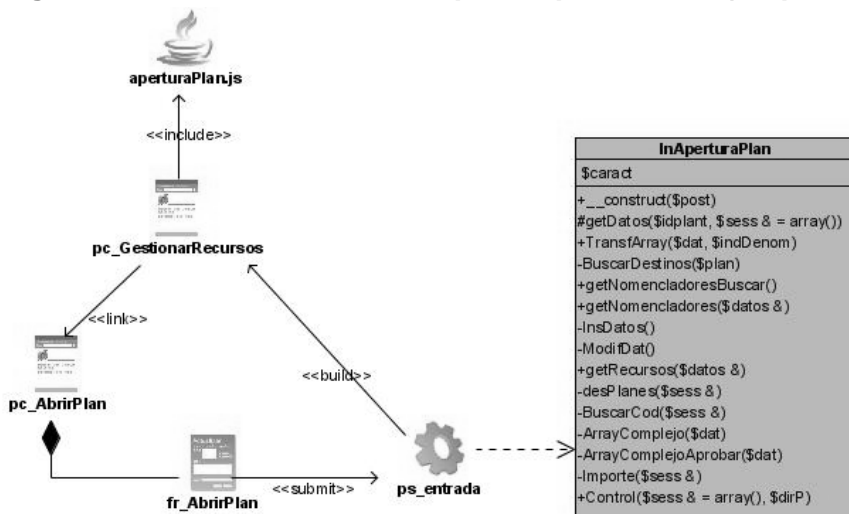


Figura 28. Diagrama de clases del diseño “Abrir Plan (Paquete Abrir Plan)”

3.9 Arquitectura.

La Arquitectura de Software es la organización fundamental de un sistema encarnada en sus componentes, las relaciones entre ellos y el ambiente y los principios que orientan su diseño y evolución.

Es a grandes rasgos, una vista del sistema que incluye los componentes principales del mismo, la conducta de esos componentes según se le percibe desde el resto del sistema y las formas en que los componentes interactúan y se coordinan para alcanzar la misión del sistema. La vista arquitectónica es una vista abstracta, aportando el más alto nivel de comprensión y la supresión o diferencia del detalle inherente a la mayor parte de las abstracciones.

3.9.1 ¿Qué son los patrones?

Los patrones son soluciones listas para usar, aplicables a problemas que se repiten con frecuencia en contextos acotados. Ayudan a construir la experiencia colectiva de Ingeniería de Software, son una abstracción de "problema – solución", ocupándose de problemas recurrentes, identificando y especificando abstracciones de niveles más altos que componentes o clases individuales y proporcionando vocabulario y entendimiento común. Algunos tipos de patrones:

- ✓ Patrones de Arquitectura. Formas de descomponer, conectar y relacionar sistemas, trata conceptos como: niveles, tuberías y filtros. Es un nivel de abstracción mayor que el de los Patrones de Diseño.
- ✓ Patrones de Programación (Idioms Patterns). Patrones de bajo nivel acerca de un lenguaje de programación concreto, describen como implementar cuestiones concretas.
- ✓ Patrones de Análisis. Conjunto de reglas que permiten modelar un sistema de forma satisfactoria.
- ✓ Patrones de Organizacionales. Describen como organizar grupos humanos, generalmente relacionados con el software.
- ✓ Otros Patrones de Software. Se puede hablar de patrones de Programación concurrente, de Interfaz Gráfica, de Organización de Código, de Optimización de Código, de Robustez de Código, de Fase de Prueba.

¿Qué es un patrón de Arquitectura?

Un patrón de arquitectura de software describe un problema particular y recurrente del diseño, que surge en un contexto específico, y presenta un esquema genérico y probado de su solución.

Patrón de Arquitectura en Capas

Capa Presentación: En la capa de presentación se establece la composición de los contenidos que se presentan finalmente al usuario, como resultado de su petición. Se añade información de presentación a los contenidos: dónde tienen que aparecer, cuál es el tamaño de letra más adecuado, dónde va el título, dónde la fotografía, dónde el gráfico, etc.

Capa Lógica de aplicación: La lógica de aplicación es la parte del portal encargada de atender las peticiones de los usuarios. Esta lógica proporciona valor añadido al contenido almacenado, combinando la información y adaptándola a las necesidades de cada usuario. En caso de tratarse de contenido externo, esta lógica es la encargada de interpretar qué tipo de conexión es necesario establecer con el proveedor y si es o no necesario aplicar transformaciones antes de servirlo al usuario final. [11]

Capa Datos: En la capa de datos es donde se almacena de forma persistente toda la información necesaria para facilitar los servicios ofrecidos por el portal. El perfil de los usuarios del portal, índices de los motores de búsqueda, contenidos de agregación, información sobre la publicación, así como información sobre las terceras partes que proveen contenido o servicios al portal. [12]

La capa que se agrega es la que surge de separar definitivamente las reglas de negocio de la de Datos. Esta arquitectura nos 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 tocar un módulo específico, así como al momento de plantear la escalabilidad de nuestro sistema, si hemos respetado las reglas básicas de diseño no deberíamos afrontar grandes modificaciones.

Ventajas y beneficios:

El uso de un modelo de n capas combinado con XML permite que los desarrollos realizados cumplan una serie de ventajas muy importantes:

- ✓ Acceso a la información en tiempo real.
- ✓ Indexación y organización de la información accesible desde una misma interfaz.
- ✓ Obtener y distribuir datos en varias aplicaciones desde un mismo programa.
- ✓ Facilidad de uso utilizando interfaces web ágiles y dinámicos.
- ✓ Ahorro de tiempo y costes en el desarrollo de nuevas aplicaciones y la integración en el resto de los procesos d gestión de la empresa.
- ✓ Desarrollos abiertos y escalables que permiten ampliar funcionalidades a medida que se necesiten e integrarlas fácilmente.
- ✓ Portabilidad de la interfaces de presentación a otras plataformas (móviles, PDA...). [13]

Patrones de creación

Además del patrón arquitectura se hace uso de algunos patrones de creación: Método Factoría y Singleton que son los que abstraen la forma en que se crean los objetos, de forma que permite tratar las clases a crear de forma genérica apartando la decisión de qué clases crear o como crearlas.

- ✓ Método Factoría: Tiene como intención abstraer la instanciación de clases relegando esta responsabilidad a las mismas clases. 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.
- ✓ Singleton: Tiene como intención ofrecer una instancia de una clase y un punto de acceso a la misma. Es un modelo que garantiza que solo hay una instancia y que se puede acceder a ella por todos. Para ello en lugar de tener una variable global, la instancia se almacena un atributo estático de la clase y se accede a ella por el método getInstance. Se aplica en aquellos casos en que hay que compartir recursos únicos como una memoria compartida por varios Threads, un Spool, etc.

Patrones de Acceso a Datos.

Para facilitar el acceso a datos desde la lógica de dominio, independientemente de la base de datos, se cuenta, con el patrón de acceso a datos:

✓Table Data Gateway (TDG) propone brindar un objeto que actúe como Gateway a una tabla, vista o conjunto de tablas, en una base de datos manejando todos los registros de esa tabla. El Gateway contiene un conjunto de operaciones CRUD ¹ con el código SQL necesario para manipular los datos en la tabla correspondiente. Generalmente no tiene estado ya que solamente extrae y actualiza información de la base de datos

Además de este existen otros patrones de acceso a datos como:

- ✓Row Data Gateway,
- ✓Active Record (AR)
- ✓Data Mapper (DM)
- ✓Record Set (RS)
- ✓Metadata Mapping

¿Qué es un Patrón de Diseño?

Son soluciones simples y elegantes a problemas específicos y comunes del diseño orientado a objetos, una técnica para flexibilizar el código haciéndolo satisfacer ciertos criterios y una manera más práctica de describir ciertos aspectos de la organización de un programa. Son soluciones basadas en la experiencia y que se ha demostrado que funcionan

Los patrones de diseño pueden incrementar o disminuir la capacidad de comprensión de un diseño o de una implementación, disminuirla al añadir accesos indirectos o aumentar la cantidad de código, disminuirla al regular la modularidad, separar mejor los conceptos y simplificar la descripción. Por otro lado, los patrones de diseño, facilitan el aprendizaje al programador inexperto, pudiendo establecer parejas problema-solución.

En la programación orientada a objetos resulta complicado descomponer el sistema en objetos (encapsulación, granularidad, dependencias, flexibilidad, reusabilidad, etc.), los patrones de diseño nos permitirán identificar a los objetos apropiados de una manera mucho más sencilla. También nos permitirán determinar la granularidad de los objetos.

Además, los patrones de diseño, también nos ayudarán a especificar las interfaces, identificando los elementos claves en las interfaces y las relaciones existentes entre distintas interfaces. También, y de forma casi automática, nos ayudan a reutilizar código, facilitando la decisión entre "herencia o composición" (favorece la composición sobre la herencia y hace uso de la delegación), relacionando estructuras en tiempo de compilación y en tiempo de ejecución y permitiendo hacer un diseño preparado para el cambio.

Se pueden clasificar a los patrones según su propósito:

- ✓ Patrones de creación: para creación de instancias.
- ✓ Patrones estructurales: relaciones entre clases, combinación y formación de estructuras mayores.
- ✓ Patrones de comportamiento: interacción y cooperación entre clases. [12]

3.10 Diseño de la Base de Datos.

Muchos han sido los argumentos a tomar en cuenta a la hora de crear el diseño de la base de datos, y entre las más destacadas se encuentran las estructuras de almacenamiento y de acceso para alcanzar un rendimiento óptimo de las aplicaciones de la base de datos, siendo los criterios adoptados el tiempo de respuesta. [15]

Uno de los pasos más importantes en el diseño de una base de datos consiste en asegurarse de que los datos se distribuyan correctamente entre las tablas. Existen distintos modos de organizar la información y representar las relaciones entre los datos en una base de datos, si las estructuras de datos son correctas, el resto de la aplicación (las consultas, los formularios, los informes, el código, etc.) se verá simplificada en gran medida.

Los Sistemas administradores de bases de datos convencionales usan uno de los tres modelos lógicos de bases de datos para hacer seguimiento de las entidades, atributos y relaciones. Los tres modelos lógicos principalmente de bases de datos son el jerárquico, de redes y el relacional. Cada modelo lógico tiene ciertas ventajas de procesamiento y también ciertas ventajas de negocios. Esta etapa consta de tres fases: diseño conceptual, diseño lógico y diseño físico de la base de datos.

3.11 Diagramas o modelos de entidad relación.

El modelo entidad-relación (E-R) es uno de los varios modelos conceptuales existentes para el diseño de bases de datos. Los elementos esenciales del modelo son las *entidades*, los *atributos* y las *relaciones*

entre las entidades. El propósito de este modelo es simplificar el diseño de bases de datos a partir de descripciones textuales de los requerimientos.

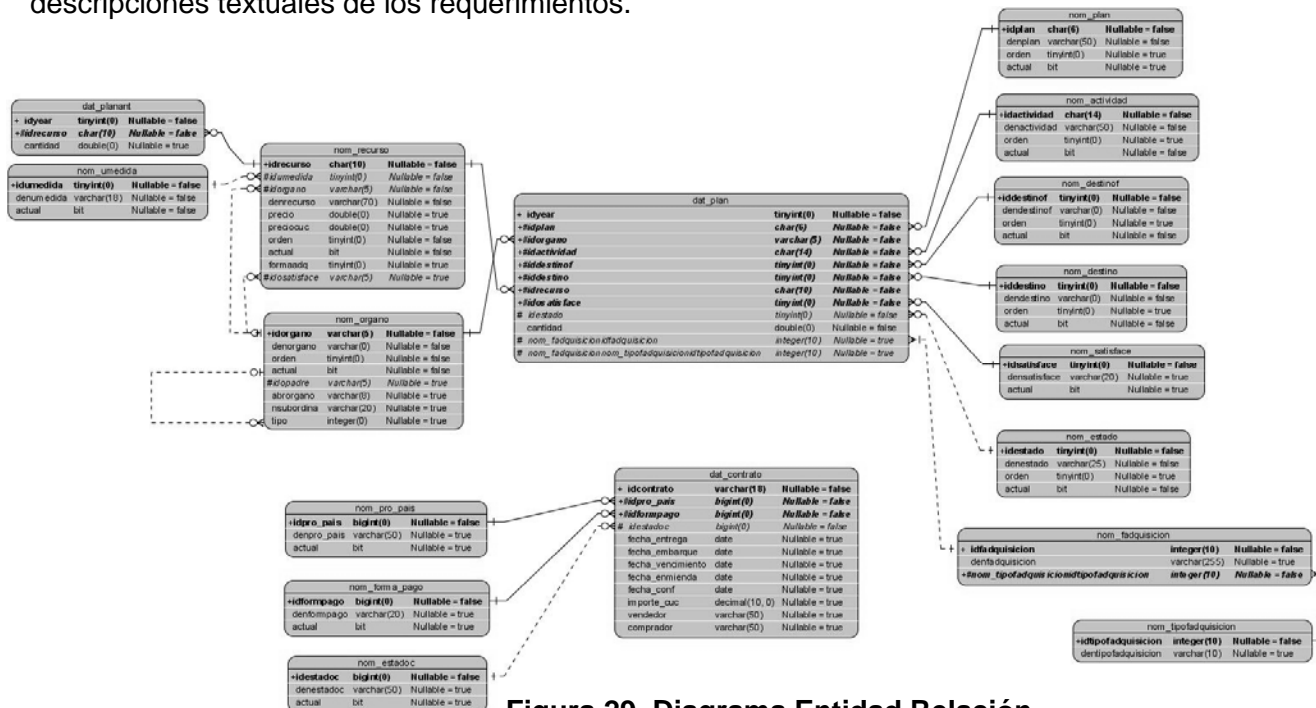


Figura 29. Diagrama Entidad Relación.

3.12 Descripción de las tablas de la Base de Datos.

bre: dat_plan		
Descripción: Contiene todos los datos de un plan determinado.		
Atributo	Tipo	Descripción
idyear	smallint	Año en el que se planifica.
idplan	char(6)	Identificador del plan.
idorgano	varchar(5)	Identificador del Órgano relacionado con el plan.
idactividad	char(14)	Identificador de la Actividad a que está vinculado el plan.
iddestinof	smallint	Identificador del Destino Final al que está encaminado el plan.
iddestino	smallint	Identificador del Destino al que está encaminado el plan dentro de su Destino Final.
idrecurso	char(10)	Identificador del recurso.
idosatisface	smallint	Identificador del Órgano que satisface el plan.
idestado	smallint	Identificador del estado del plan.
cantidad	double precision	Cantidad a planificar.

Tabla 20. Descripción de la tabla “dat_plan”

Nombre: dat_planant		
Descripción: Contiene datos de un plan determinado.		
Atributo	Tipo	Descripción
idyear	smallint	Año en el que se planifica.
idrecurso	char(10)	Identificador del recurso (Servicio, Equipo o Producto).
cantidad	double precision	Cantidad a planificar.

Tabla 21. Descripción de la tabla “dat_planant”

Nombre: nom_destino		
Descripción: Contiene todos los atributos que caracterizan al nomenclador Destino.		
Atributo	Tipo	Descripción
iddestino	smallint	Identificador del Destino al que está encaminado el plan dentro de su Destino Final.
dendestino	varchar	Denominación del Destino al que está encaminado el plan dentro de su Destino Final.
orden	smallint	Orden del Destino al que está encaminado el plan dentro de su Destino Final.
actual	bit(1)	Estado del atributo (activo o inactivo).

Tabla 22. Descripción de la tabla “nom_destino”

Nombre: nom_destinof		
Descripción: Contiene todos los atributos que caracterizan al nomenclador Destino Final.		
Atributo	Tipo	Descripción
iddestinof	smallint	Identificador del Destino Final al que está encaminado el plan.
dendestinof	varchar	Denominación del Destino Final al que está encaminado el plan.
orden	smallint	Orden del Destino Final al que está encaminado el plan.
actual	bit(1)	Estado del atributo (activo o inactivo).

Tabla 23. Descripción de la tabla “nom_destinof”

Nombre: nom_estado		
Descripción: Contiene todos los atributos que caracterizan al nomenclador Estado.		
Atributo	Tipo	Descripción
idestado	smallint	Identificador del estado del plan.
denestado	varchar(25)	Denominación del estado del plan.
orden	smallint	Orden del estado del plan.
actual	bit(1)	Estado del atributo (activo o inactivo).

Tabla 24. Descripción de la tabla “nom_estado”

Nombre: nom_organ		
Descripción: Contiene todos los atributos que caracterizan al nomenclador Órgano.		
Atributo	Tipo	Descripción
idorgano	varchar(5)	Identificador del Órgano relacionado con el plan.
denorgano	varchar	Denominación del Órgano relacionado con el plan.
orden	smallint	Orden del Órgano relacionado con el plan.
actual	bit(1)	Estado del atributo (activo o inactivo).
abroorgano	varchar(8)	Abreviatura del Órgano relacionado con el plan.
nsubordina	varchar(20)	Nivel de subordinación de un Órgano determinado.

Tabla 25. Descripción de la tabla “nom_organ”

Nombre: nom_plan		
Descripción: Contiene todos los atributos que caracterizan al nomenclador Plan.		
Atributo	Tipo	Descripción
idplan	char(6)	Identificador del plan.
idconceptogasto	varchar(4)	Identificador del concepto de gasto.
denplan	varchar(50)	Denominación o nombre del plan.
orden	smallint	Orden del plan.
actual	bit(1)	Estado del atributo (activo o inactivo).

Tabla 26. Descripción de la tabla “nom_plan”

Nombre: nom_recurso		
Descripción: Contiene todos los atributos que caracterizan al nomenclador Recurso.		
Atributo	Tipo	Descripción
idrecurso	char(10)	Identificador del recurso (Servicio, Equipo o Producto).
idumedia	smallint	Identificador de la unidad con que se mide el recurso.
idorgano	varchar(5)	Identificador del Órgano.
idnatrecurso	smallint	Identificador de la naturaleza del recurso.
denrecurso	varchar(50)	Denominación o nombre del recurso.
precio	double precision	Precio del recurso.
preciocuc	double precision	Precio del recurso en CUC.
orden	smallint	Orden del recurso.
actual	bit(1)	Estado del atributo (activo o inactivo).
formaadq	smallint	Forma de adquisición (Importación directa, Sustitución de importación)
idosatisface	varchar(5)	Identificador del Órgano que satisface el plan.

Tabla 27. Descripción de la tabla “nom_recurso”

Nombre: nom_umedida		
Descripción: Contiene todos los atributos que caracterizan al nomenclador Unidad de Medida.		
Atributo	Tipo	Descripción
idumedida	smallint	Identificador de la unidad con que se mide el recurso.
denumedida	varchar(18)	Denominación de la unidad con que se mide el recurso.
actual	bit(1)	Estado del atributo (activo o inactivo).

Tabla 28. Descripción de la tabla “nom_umedida”

Nombre: nom_actividad		
Descripción: Contiene todos los atributos que caracterizan al nomenclador Actividad.		
Atributo	Tipo	Descripción
idactividad	char(14)	Identificador de la Actividad a que está vinculado el plan.
denactividad	varchar(50)	Denominación de la Actividad a que está vinculado el plan.
orden	smallint	Orden de la Actividad a que está vinculado el plan.
actual	bit(1)	Estado del atributo (activo o inactivo).

Tabla 29. Descripción de la tabla “nom_actividad”

Nombre: nom_satisface		
Descripción: Contiene quien satisface el producto.		
Atributo	Tipo	Descripción
idsatisface	smallint	Identificador del que satisface.
densatisface	varchar(20)	Denominación del que satisface
actual	bit(1)	Estado del atributo (activo si está en 1 o inactivo si está en 0).

Tabla 30. Descripción de la tabla “nom_satisface”

Nombre: dat_contrato		
Descripción: Contiene todos los datos de un Modelo SCM 711.		
Atributo	Tipo	Descripción
idcontrato	varchar(18)	Identificador del Contrato.
fecha_entrega	date	Fecha de entrega de los recursos contratados
fecha_embarque	date	Fecha de embarque de los recursos contratados
fecha_vencimiento	date	Fecha de vencimiento de los recursos contratados
fecha_enmienda	date	Fecha de enmienda del contrato
fecha_conf	date	Fecha de confección de la enmienda del contrato
importe_cuc	decimal(10,0)	Importe en moneda CUC.
vendedor	varchar(50)	Nombre del vendedor.
comprador	varchar(50)	Nombre del comprador.
idpro_pais	bigint(0)	Identificador del proveedor y del país del proveedor.
idform_pago	bigint(0)	Identificador de la forma de pago.

Idestadoc	bigint(0)	Identificador del estado del contrato.
idimportador	bigint(0)	Identificador del importador
idconsumidor	varchar(10)	Identificador del consumidor
idaprobacion	char(10)	Identificador de la aprobación.
idrecurso	varchar(10)	Identificador del recurso
idoferta	varchar(10)	Identificador de la Oferta

Tabla 31. Descripción de la tabla “dat_contrato”

Nombre: nom_pro_pais		
Descripción: Contiene todos los atributos que caracterizan al nomenclador de Proveedor/País.		
Atributo	Tipo	Descripción
idpro_pais	bigint(0)	Identificador del proveedor y del país del proveedor.
denpro_pais	varchar(50)	Denominación del proveedor y del país del proveedor.
actual	bit(0)	Estado del atributo (activo si está en 1 o inactivo si está en 0).

Tabla 32. Descripción de la tabla “nom_pro_pais”

Nombre: nom_estadoc		
Descripción: Contiene todos los atributos que caracterizan al nomenclador de Estado de la Aprobación.		
Atributo	Tipo	Descripción
idestadoc	bigint(0)	Identificador del estado del Contrato.
denestadoc	varchar(50)	Denominación del estado del Contrato.
actual	bit(0)	Estado del atributo (activo si está en 1 o inactivo si está en 0).

Tabla 33. Descripción de la tabla “nom_estadoc”

Nombre: nom_forma_pago		
Descripción Contiene todos los atributos que caracterizan al nomenclador de Forma de Pago.		
Atributo	Tipo	Descripción
idform_pago	bigint(0)	Identificador de la forma de pago.
denform_pago	varchar(20)	Denominación de la forma de pago.
actual	bit(0)	Estado del atributo (activo si está en 1 o inactivo si está en 0).

Tabla 34. Descripción de la tabla “nom_forma_pago”

Nombre: nom_fadquisicion		
Descripción: Contiene todos los atributos que caracterizan el tipo de adquisición del producto.		
Atributo	Tipo	Descripción
idfadquisicion	Integer(10)	Identificador del tipo de adquisición.
denfadquisicion	varchar(255)	Denominación del tipo de adquisición.

Tabla 35. Descripción de la tabla “nom_fadquisicion”

Nombre: nom_tipofadquisicion		
Descripción: Contiene todos los atributos que caracterizan la forma de adquisición del producto.		
Atributo	Tipo	Descripción
idtipofadquisicion	Integer(10)	Identificador de la forma de adquisición.
dentipofadquisicion	varchar(10)	Denominación de la forma de adquisición.

Tabla 36. Descripción de la tabla “nom_tipofadquisicion”

3.13 Conclusiones.

Durante el presente capítulo se efectuó la modelación de los diagramas de clases de análisis y de diseño analizándose el dominio del problema para construir un modelo del mundo real utilizando objetos, refinando así los modelos de análisis para crear especificaciones adicionales que enriquecen al mismo con detalles próximos para una futura implementación.

CAPÍTULO 4: IMPLEMENTACIÓN

4.1 Introducción.

A continuación se presentan los modelos definidos en RUP como diagrama de despliegue y diagrama de componentes como objetivo primordial de la fase implementación, en los cuales se muestra la descripción física de la topología del sistema y la estructura de las unidades de hardware y el software que se ejecuta en cada unidad como la disposición de las partes integrantes de la aplicación y las dependencias entre los distintos módulos de la aplicación.

4.2 Diagramas de Implementación.

Un diagrama de implementación muestra las dependencias entre las partes de código del sistema (diagrama de componentes) o la estructura del sistema en ejecución (diagrama de despliegue): los diagramas de componentes se utilizan para modelar la vista de implementación estática de un sistema, mientras que los diagramas de despliegue se utilizan para modelar la vista de despliegue estática.

4.2.1 Diagrama de Despliegue.

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

Un diagrama de despliegue es un grafo de nodos unidos por conexiones de comunicación. Un nodo puede contener instancias de componentes software, objetos, procesos (caso particular de un objeto). En general un nodo será una unidad de computación de algún tipo, desde un sensor a un mainframe. Las instancias de componentes software pueden estar unidas por relaciones de dependencia, posiblemente a interfaces (ya que un componente puede tener más de una interfaz). [15]

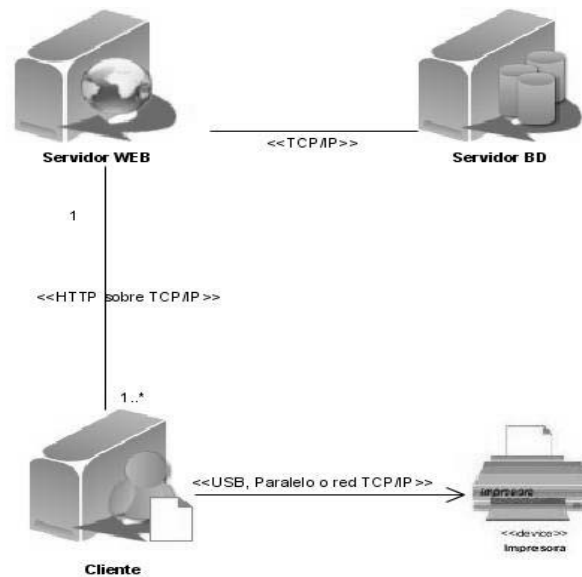


Figura 29. Diagrama de Despliegue.

4.2.2 Diagrama de Componentes.

Un diagrama de componentes muestra las organizaciones y dependencias lógicas entre componentes *software*, sean éstos componentes de código fuente, binarios o ejecutables. Desde el punto de vista del diagrama de componentes se tienen en consideración los requisitos relacionados con la facilidad de desarrollo, la gestión del *software*, la reutilización, y las restricciones impuestas por los lenguajes de programación y las herramientas utilizadas en el desarrollo. Los elementos de modelado dentro de un diagrama de componentes serán componentes y paquetes. En cuanto a los componentes, sólo aparecen tipos de componentes, ya que las instancias específicas de cada tipo se encuentran en el diagrama de despliegue. [25]

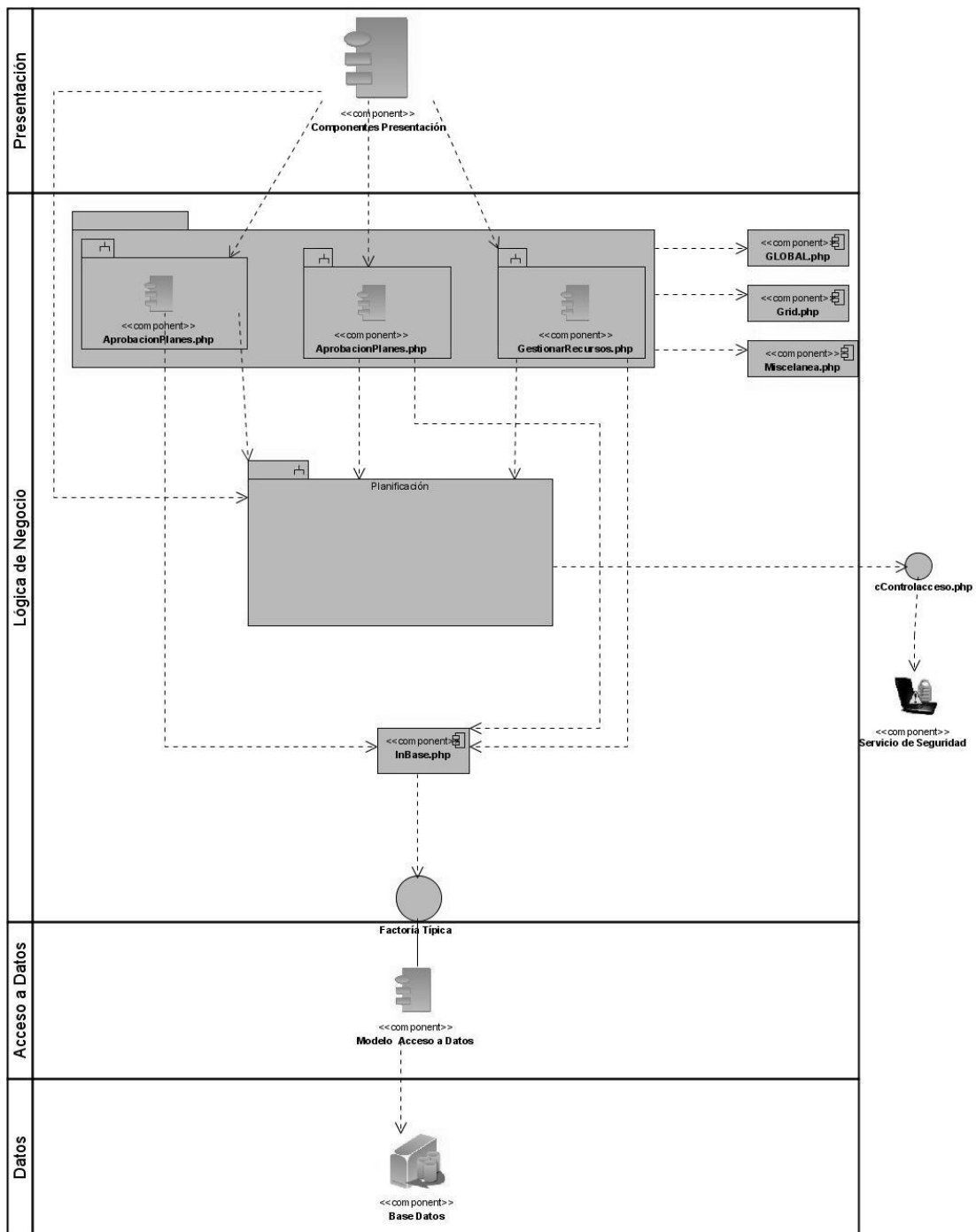


Figura 30. Diagrama de Componentes General.

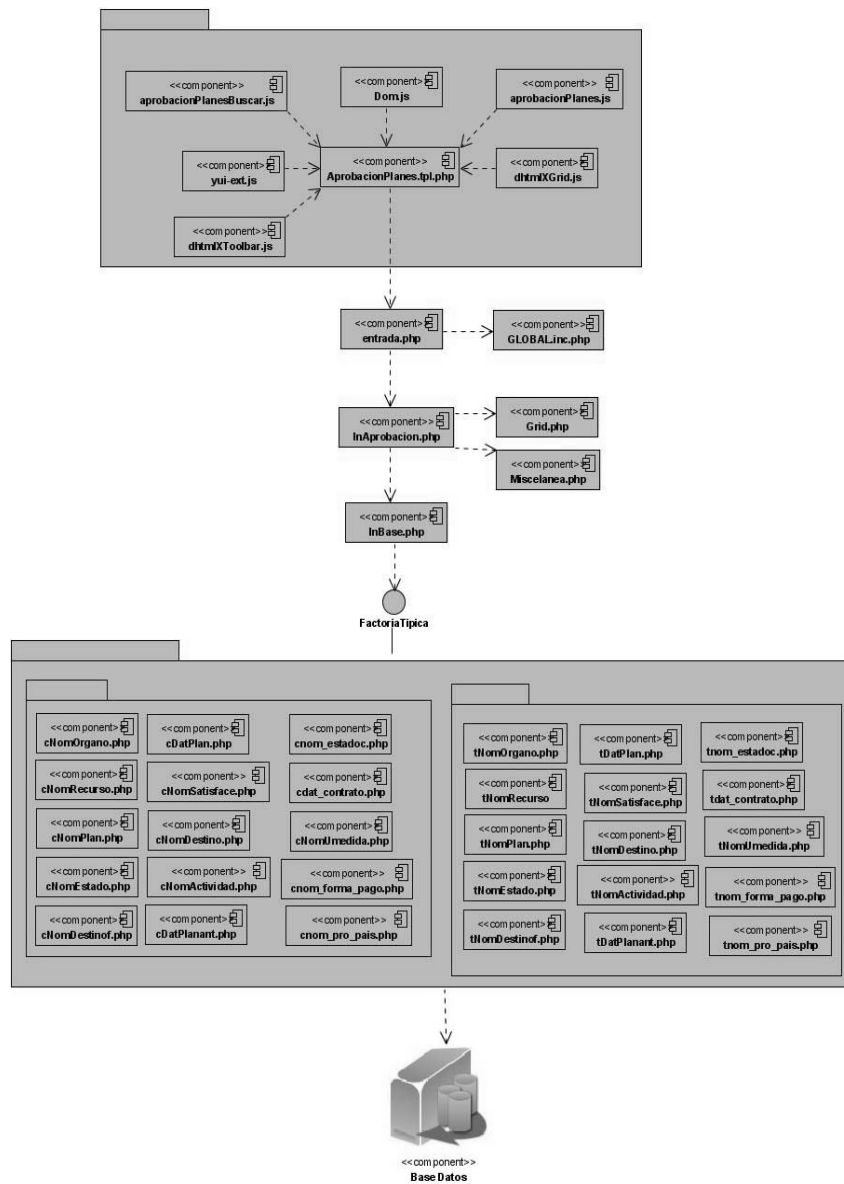


Figura 32. Diagrama de Componentes Gestionar Plan y Aprobar plan de CB.

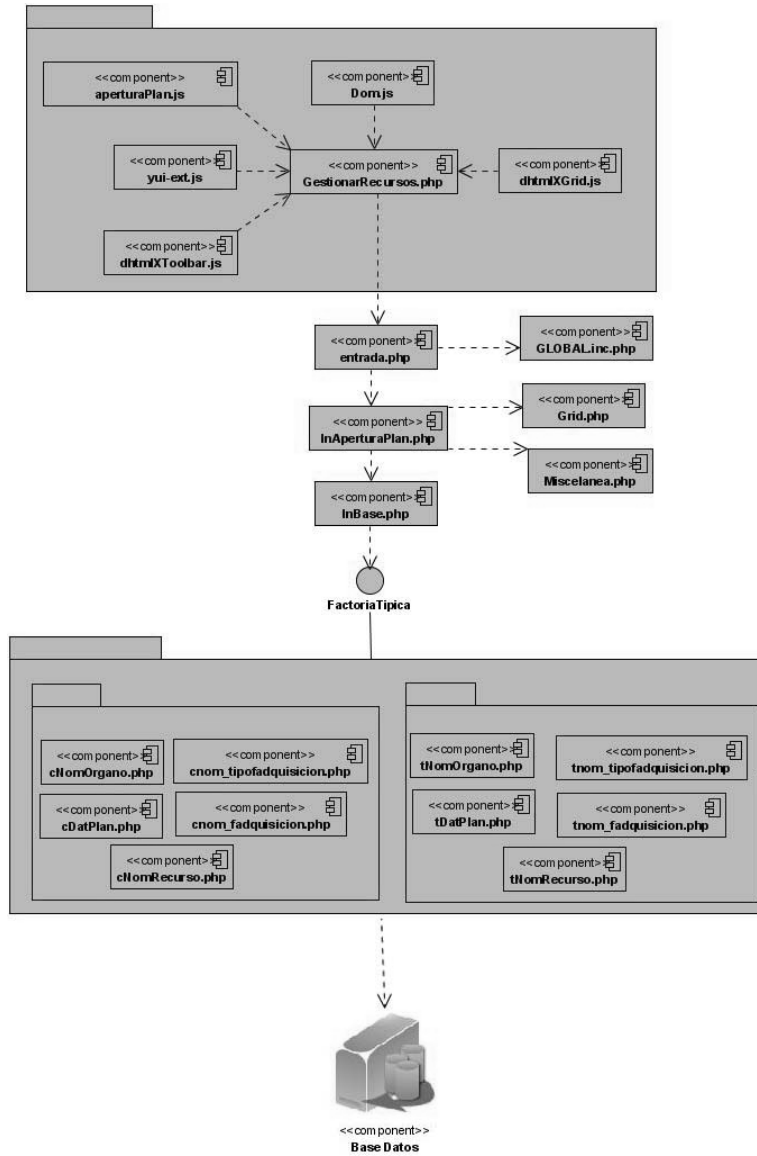


Figura 33. Diagrama de Componentes Abrir Plan.

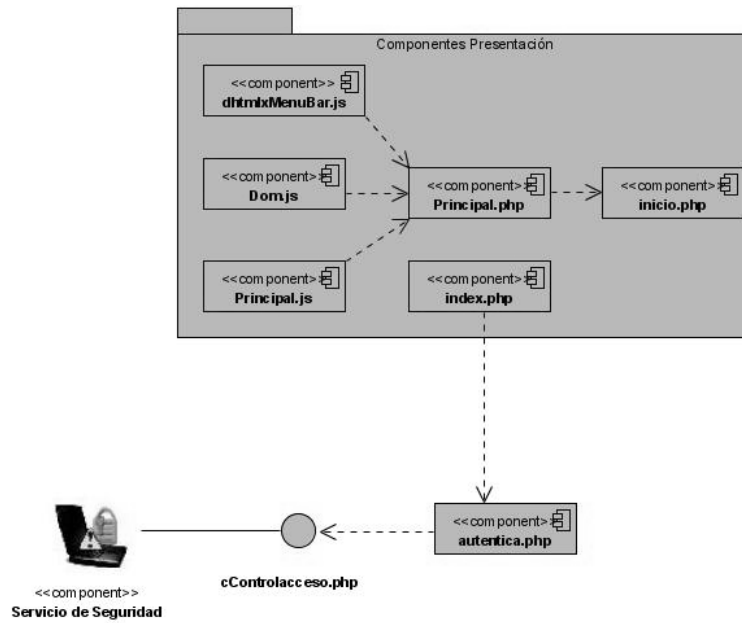


Figura 31. Diagrama de Componentes Sistema Planificación.

4.3 Conclusiones.

En este capítulo se abordaron los objetivos de la fase implementación del sistema verificando que el comportamiento externo del sistema satisfaga los requisitos establecidos por los clientes y futuros usuarios del mismo., y todo esto a través de dos formas de representación como la del diagrama de componentes que mostró la organización y la dependencia entre un conjunto de componentes, la vista estática del sistema y el diagramas de despliegue que son los complementos de los diagramas de componentes que unidos proveen la vista de implementación del sistema, dando paso así a la implementación del mismo.

CONCLUSIONES

El presente trabajo una vez terminado y certificado se pondrá en explotación en el MINFAR como parte del sistema de planificación que actualmente se desarrolla de forma manual o semiautomática. En el mismo se reflejó el estudio realizado, el cual permitió obtener las bases para la generación de una propuesta de solución que permitiera realizar los procesos de gestión de una forma rápida y eficiente.

Se considera que la hipótesis planteada al inicio ha sido debidamente demostrada aportando a través del análisis y el diseño las bases para el futuro desarrollo de una aplicación Web que se identifique por la interconexión y la integración en el flujo de la información de los organismos en red.

Las metas propuestas fueron alcanzadas y se logró el objetivo fundamental de la investigación, el análisis, y diseño de una aplicación Web que contribuya a eliminar las barreras de la distancia y una rápida toma de decisiones en la planificación material y financiera de los Centros de Balance.

RECOMENDACIONES

Este trabajo fue concebido inicialmente para darle solución a un problema existente en el MINFAR durante el proceso de planificación material y financiera, pero debido a la gran complejidad de dicho proceso, se acordó hacer una primera iteración en la cual se le dio una solución a las situaciones que más apremiaban. Sin embargo esta solución se puede mejorar para así lograr que el usuario tenga mayor facilidad en el trabajo, por tanto se recomienda:

- ✓ Continuar trabajando sobre la base de lo diseñado para finalizar así todas las funcionalidades que requiere un Centro de Balance.
- ✓ Analizar la posibilidad de realizar el cambio de año en la misma interfaz en la que se efectúan todas las operaciones.
- ✓ Obtener un análisis estadístico por parte de la aplicación cuando se realice la comparación de los planes de los OC de años anteriores y actuales, para así facilitarle al cliente su trabajo ahorrarle tiempo y lograr una rápida toma de decisiones.

REFERENCIAS BIBLIOGRÁFICAS

1. Marín, J.C. [cited 2007 1 de febrero del 2007]; Available from: <http://www.uco.es/ccg/glosario/glosario.html#INTERNET>.
2. Andalia, R.C. [cited 2007; Available from: http://bvs.sld.cu/revistas/aci/vol12_1_04/aci05104.htm
3. Gilardi, L. and B. Rodríguez. 1996-2004 [cited 2007 1 de febrero del 2007]; Available from: <http://www.soho.com.mx/content/dfdcfad2-83d4-4304-9c4d-4d41f49a8f13>.
4. Andalia, R.C. [cited 2007 1 de febrero del 2007]; Available from: http://bvs.sld.cu/revistas/aci/vol12_1_04/aci05104.htm
5. Line, E.O. [cited 2007 1 de febrero del 2007]; Available from: <http://www.economicas-online.com/parainter.htm>.
6. Gracia, J. 29 de enero del 2007 [cited 2007; Available from: <http://www.webestilo.com/javascript/>
7. Vilas, A.F. 29 de enero del 2007 [cited 2007 30 de enero del 2007]; Available from: <http://www-gris.det.uvigo.es/~avilas/UML/node25.html>.
8. Morea, L. 30 de enero del 2007 [cited 2007; Available from: <http://www.wordreference.com/definicion/servidor>.
9. Alvarez, M.A. 29 de enero del 2007 [cited 2007 29 de enero del 2007]; Available from: <http://www.desarrolloweb.com/articulos/715.php>
10. Alvarez, M.A. 29 de enero del 2007 [cited 2007; Available from: <http://www.desarrolloweb.com/javascript>.
11. Informática Milenium, S.A.d.C.V. 1998-2006 30 de enero del 2007 [cited 2007; Available from: <http://www.informaticamilenium.com.mx/Paginas/espanol/sitioweb.htm>.
12. Dirson. 30 de enero del 2007 [cited 2007 30 de enero del 2007]; Available from: <http://google.dirson.com/posicionamiento.net/paginas-dinamicas/>
13. Dirson. 30 de enero del 2007 [cited 2007; Available from: <http://google.dirson.com/posicionamiento.net/beneficios-paginas-dinamicas/>.
14. Alvarez, S. 29 de enero del 2007 [cited; Available from: <http://www.monografias.com/trabajos5/inso/inso.shtml#intro>
15. Henst, C.V.D. 29 de enero del 2007 [cited 2007; Available from: <http://www.unav.es/cti/manuales/HTML/indice.html>.
16. [cited 2007; Available from: <http://gtk.php.net/manual1/es/html/intro.whatis.php.history.html>.

17. Aguila, Y.P. [cited 2007; Available from:
http://www.sappiens.com/castellano/articulos.nsf/Inform%C3%A1tica/AJAX_un_nuevo_acercamiento_a_las_aplicaciones_web/1308E8D01F5F704BC125727E00830B62!opendocument.
18. Simón, M. [cited 2007 30 de marzo del 2007]; Available from: <http://es.tldp.org/Postgresql-es/web/navegable/tutorial/x56.html#AEN136>.
19. MESA, A.A. and A.R. TAMAYO. 1997 [cited 2007; Available from:
<http://www.monografias.com/trabajos24/herramientas-case/>
20. Vizcaíno, A., F. ÓscarGarcía, and I. Caballero. [cited 2007 1 de febrero del 2007]; Available from:
http://alarcos.inf-cr.uclm.es/per/fgarcia/isoftware/doc/LabTr1_VP.pdf.
21. Navarro, J. [cited 2007 1 de febrero del 2007]; Available from:
<http://www.versioncero.com/noticia/210/visual-paradigm-for-uml>
22. Bagnato, L. 30 de enero del 2007 [cited 2007 29 de enero del 2007]; Available from:
<http://www.monografias.com/trabajos5/inso/inso.shtml#intro>
23. Bagnato, L. [cited 2007 2 de abril del 2007]; Available from:
<http://dialnet.unirioja.es/servlet/articulo?codigo=2097256>.
24. Jacobson, I., G. Booch, and J. Rumbaugh. [cited 2007; Available from:
<http://bibliodoc.uci.cu/pdf/reg00060.pdf>.
25. Porter, J. 2005 [cited 2007; Available from:
http://www.masternewmedia.org/es/2005/07/28/ajax_el_nuevo_abordaje_de.htm

BIBLIOGRAFÍA

- ✓ [cited 2007 30 de enero del 2007]; Available from: http://es.wikipedia.org/wiki/Sitio_web
- ✓ [cited 2007 30 de enero del 2007]; Available from: <http://www.desarrolloweb.com/articulos/541.php>
- ✓ [cited 2007 30 de enero del 2007]; Available from: <http://es.wikipedia.org/wiki/Cliente>
- ✓ [cited 2007 30 de enero del 2007]; Available from: <http://es.wikipedia.org/wiki/Servidor>
- ✓ [cited 2007 30 de enero del 2007]; Available from:
<http://www.informaticamilenium.com.mx/Paginas/espanol/sitioweb.htm>
- ✓ [cited 2007 30 de enero del 2007]; Available from: <http://es.wikipedia.org/wiki/JavaScript>
- ✓ [cited 2007 30 de enero del 2007]; Available from: <http://www.webestilo.com/javascript>
- ✓ [cited 2007 30 de enero del 2007]; Available from: http://es.wikipedia.org/wiki/Active_Server_Pages
- ✓ [cited 30 de enero del 2007]; Available from: <http://www.desarrolloweb.com/articulos/831.php>
- ✓ [cited 30 de enero del 2007]; Available from: http://es.wikipedia.org/wiki/Rational_Unified_Process
- ✓ [cited 30 de enero del 2007]; Available from: http://www.indudata.com/1rational_rose.htm#6
- ✓ [cited 30 de enero del 2007]; Available from:
<http://www.lawebdelprogramador.com/diccionario/mostrar.php?letra=A&pagina=3>
- ✓ [cited 2007 6 de enero del 2007]; Available from:
http://www.htmlpoint.com/javascript/corso/js_02.htm
- ✓ [cited 30 de marzo del 2007]; Available from: <http://es.tldp.org/Postgresql-es/web/navegable/tutorial/x56.html#AEN136>
- ✓ [cited 2 de abril del 2007]; Available from:
<https://pid.dsic.upv.es/C1/Material/Documentos%20Disponibles/Introducci%C3%B3n%20a%20RUP.doc>
- ✓ [cited 2007 30 de enero del 2007]; Available from:
http://es.wikipedia.org/wiki/Aplicaci%C3%B3n_web

GLOSARIO

1. **TCP/IP:** **TCP** (Transmission Control Protocol) y el **IP** (Internet Protocol) son los que dan nombre al conjunto TCP/IP, el protocolo común utilizado por todos los ordenadores conectados a Internet, de manera que éstos puedan comunicarse entre sí
2. **World Wide Web:** (O la "Web"): es un sistema de documentos de hipertexto enlazados y accesibles a través de Internet. Con un navegador Web, un usuario visualiza páginas Web que pueden contener texto, imágenes u otros contenidos multimedia, y navega a través de ellas usando hiperenlaces.
3. **ARPA:** (Advanced Research Projects Agency) En 1958 se organiza en EE.UU. la agencia gubernamental de investigación creada en respuesta a los desafíos tecnológicos y militares de Rusia, durante la Guerra Fría, de la cual surgiran una década más tarde los fundamentos de la futura red global de computadores Internet. Es la respuesta al lanzamiento realizado un año antes por Rusia del primer satélite artificial, el Sputnik I, el suceso, de fuertes implicancias estratégicas y militares ha impactado a los EE.UU.
4. **ARPANET:** (Advanced Research Projects Agency Network): fue creada por encargo del Departamento de Defensa de los Estados Unidos como medio de comunicación para los diferentes organismos del país. El primer nodo se creó en la Universidad de California y fue la espina dorsal de Internet hasta 1990, tras finalizar la transición al protocolo TCP/IP en 1983.
5. **ERP:** Planeación de Recursos Empresariales
6. **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.
7. **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.
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. **URL:** (Uniform Resource Locator): Es el modo estándar de proporcionar la dirección de cualquier recurso en Internet. No solo representa una dirección de Internet sino que apunta aun recurso concreto dentro de esa dirección.
 10. **JavaScript:** Lenguaje de programación para realizar scripts, usado para la creación de acciones y procesos de automatización para páginas web. Puede ser insertado entre el código HTML e interpretado y ejecutado por el navegador
 11. **Scripts:** Del inglés, guión. Instrucciones internas de una aplicación.
 12. **Java:** Es un lenguaje de programación que ofrece la potencia del diseño orientado a objetos con una sintaxis fácilmente accesible y un entorno robusto y agradable.
 13. **C:** Lenguaje de programación orientado a la implementación de Sistemas Operativos (los sistemas operativos Linux y UNIX están escritos mayormente en C, pero se ha convertido en un lenguaje de propósito general de los más usados.
 14. **PHP:** Hypertext Pre-processor (Preprocesador de Hipertexto) es un lenguaje de programación tipo script para entornos Web utilizado, sobre todo, en servidores Linux con el fin de personalizar la información que se envía a los usuarios que acceden a un sitio web.
 15. **mySQL:** Es un sistema de gestión de base de datos, multihilo y multiusuario
 16. **Zend:** Se trata de un programa de la casa Zend, impulsores de la tecnología de servidor PHP, orientada a desarrollar aplicaciones web, en lenguaje PHP. El programa, además de servir de editor de texto para páginas PHP, proporciona una serie de ayudas que pasan desde la creación y gestión de proyectos hasta la depuración de código.
 17. **ODBC:** Siglas de Open DataBase Connectivity, un estándar de acceso a Bases de Datos desarrollado por Microsoft Corporation, el objetivo de ODBC es hacer posible el acceder a cualquier dato de cualquier aplicación, sin importar qué Sistema Gestor de Bases de Datos (DBMS por sus siglas en Ingles) almacene los datos, ODBC logra esto al insertar una capa intermedia llamada
 18. **C++:** Es un lenguaje de programación, es el lenguaje que abarca tres paradigmas de la programación: la programación estructurada, la programación genérica y la programación orientada a objetos.

19. **Perl:** (Lenguaje Práctico para la Extracción e Informe), es un lenguaje interpretado que tiene varias utilidades, pero está principalmente orientado a la búsqueda, extracción y formateado de ficheros de tipo texto.
20. **CGI:** Common Gateway Interface (en inglés «Pasarela de Interfaz Común», abreviado CGI) es una importante tecnología de la World Wide Web que permite a un cliente (explorador web) solicitar datos de un programa ejecutado en un servidor web. CGI especifica un estándar para transferir datos entre el cliente y el programa. Toda pieza de software puede ser un programa CGI si esta maneja entradas y salidas (input, output) de acuerdo a los estándares CGI.
21. **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.
22. **IIS:** Internet Information Services (o Server), IIS, es una serie de servicios para los ordenadores que funcionan con Windows.
23. **AOLSERVER:** Es un servidor diseñado a partir de las deficiencias que existían en el modelo inicial utilizado por Apache.
24. **Roxen:** Este programa realiza las funciones de servidor web de alto rendimiento con características en cuanto a rendimiento y estabilidad similares a otros servidores del mercado como Apache o Microsoft IIS.
25. **MS SQL:** Servidor de base datos
26. **Oracle:** Es un sistema de gestión de base de datos relacional (o RDBMS por el acrónimo en inglés de Relational Data Base Management System)
27. **Informix:** Información en Unix, 'es una familia de productos RDBMS de IBM
28. **PostgreSQL:** Es un servidor de base de datos relacional libre, liberado bajo la licencia BSD.
29. **Cold fusion:** Servidor de páginas webs desarrollado originalmente por Allaire y luego por Macromedia (actualmente de Adobe Systems) que genera de manera rápida contenido dinámico por medio de tags especiales embebidos en código HTML, incluso su programación es mediante tags similares al HTML.
30. **ASP:** Acrónimo de Active Server Page. Tecnología creada por Microsoft, con el fin de que el usuario de Internet pueda recibir páginas generadas dinámicamente en el "Servidor". Una página asp es una página HTML que incluye en su código un script que es procesado por el IIS de Microsoft antes de enviar la página al navegador. Las páginas ASP se pueden escribir en VBScript, JScript, etc.

31. **Debugger:** Depurador. Aplicación o herramienta que permite la ejecución controlada de un programa o un código, para seguir cada instrucción ejecutada y localizar así bugs (depuración), códigos de protección, etc.
32. **Windows:** Es un sistema operativo con interfaz gráfica para computadoras personales
33. **Macs:** Es la última versión de Mac OS, el sistema operativo de la familia de ordenadores Macintosh. Está basado en BSD y cualquier persona puede aportar contribuciones encaminadas a mejorar la plataforma.
34. **OS:** Sistema Operativo.
35. **THHTTPD:** Programa que corre de fondo en un servidor web y espera peticiones de entrada para responderles.
36. **API:** (Application Program Interface). Conjunto de convenciones internacionales que definen cómo debe invocarse una determinada función de un programa desde una aplicación. Cuando se intenta estandarizar una plataforma, se estipulan unos APIs comunes a los que deben ajustarse todos los desarrolladores de aplicaciones.
37. **PDF:** Es la Extensión que corresponde a un tipo de fichero (un libro electrónico) creado con Adobe Acrobat.
38. **Flash:** Software de Macromedia para crear pequeñas animaciones vectoriales reproducidas en la Web. El navegador de un usuario necesita el plug-in Flash Player para interpretar las animaciones Flash.
39. **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. Es más amplio, más rico y más dinámico que HTML. 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. El formato de XML es muy parecido al del HTML aunque no es una extensión ni un componente de éste
40. **IMAP/ POP:** (Internet Message Access Protocol): Son dos formas de recuperación de mensajes que operan con clientes como Eudora, Netscape, etcétera. Una forma sencilla de entender el funcionamiento de POP (Post Office Protocol) e IMAP (Internet Message Access Protocol) es compararlos con los carteros del correo convencional, aunque sólo se puede usar uno de ellos a la vez en el cliente de correo electrónico

41. **AJAX:** Acrónimo de Asynchronous JavaScript And XML (JavaScript y XML asíncronos), es una técnica de desarrollo web para crear aplicaciones interactivas
42. **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.
43. **CSS:** (Hoja de Estilo en Cascada) Dentro del diseño de páginas de Internet se presenta esta como la vanguardia en cuanto a definición de estilos dentro de las plantillas de diseño. A través de instrucciones en código HTML se definen los estándares del conjunto de páginas que conforman el proyecto.
44. **XMLHttpRequest:** 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 XMLHttpRequest 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.
45. **JSON:** Acrónimo de "JavaScript Object Notation", es un formato ligero para el intercambio de datos.
46. **XSLT:** (Extensible Stylesheet Language Transformation) Lenguaje de transformación basado en hojas de estilo. Es un lenguaje que nos permite dar al XML un formato de salida comprensible ya sea: Transformar un documento XML en otro XML o convertirlo en otro formato de salida como puede ser RTF, PDF, etc.
47. **BSD:** Son las iniciales de Berkeley Software Distribution (en español, Versión de Software Berkeley) y se utilizan para identificar un sistema operativo derivado del sistema Unix nacido a partir de las aportaciones realizadas a ese sistema por la Universidad de California en Berkeley.
48. **Case:** Acrónimo inglés de Computer Aided Software Engineering, que viene a significar Ingeniería de Software Asistida por Ordenador.
49. **Visual Paradigm:** Es una herramienta CASE visual
50. **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).

-
51. **RUP:** El Proceso Racional Unificado o RUP (Rational Unified Process), es un proceso de desarrollo de software.
 52. **LAN:** Local Area Network («Red de área local»). Nombre genérico dado a una red de máquinas conectadas al mismo cable físico en un espacio relativamente reducido (por ejemplo, misma oficina, mismo edificio,...).
 53. **UNIX /Linux:** Sistema operativo abierto utilizado originariamente en grandes sistemas informáticos a los que tienen acceso simultáneo gran cantidad de usuarios. Hoy en día existen multitud de variantes que se adaptan a todo tipo de equipos informáticos. Linux es su variante gratuita y de código abierto. Debido a que fue diseñado para funcionar en Red, es el sistema operativo más difundido en Servidores conectados a Internet. RedHat Una de las más grandes compañías dedicadas al software de código abierto y el más grande distribuidor de sistemas operativos Linux.
 54. **Stakeholders:** Es un individuo o grupo de individuos que tiene intereses directos e indirectos en una empresa que puede ser afectado en el logro de sus objetivos por las acciones, decisiones, políticas o prácticas empresariales, ya que estas tienen obligación moral con la sociedad y estas obligaciones se conoce como responsabilidad social empresarial.
 55. **Mandrake:** Es un sistema operativo potente que está disponible para Intel Pentium® o superior, AMD K6, K6II, Athlon®.
 56. **Gentoo:** Software de Sistema Operativo que automáticamente configura y optimiza la ejecución del hardware subyacente y se adapta para muchas situaciones y aplicaciones, a saber, servidores seguros, estaciones de desarrollo, escritorios para uso profesional, estaciones de videojuegos, y soluciones embebidas
 57. **Ubuntu:** Es una distribución Linux que ofrece un sistema operativo predominantemente enfocado a ordenadores de escritorio aunque también proporciona soporte para servidores. Basada en Debian GNU/Linux
 58. **Debian:** Sistema operativo.
 59. **JDBC:** Acrónimo de Java Database Connectivity, un API que permite la ejecución de operaciones sobre bases de datos desde el lenguaje de programación Java independientemente del sistema de operación donde se ejecute o de la base de datos a la cual se accede utilizando el dialecto SQL del modelo de base de datos que se utilice.