



Universidad de las Ciencias Informáticas
Facultad 8

Título: “Sistema informativo de la demanda para la lucha armada.”

(Módulo Planificar Demanda)

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

Autor: Mislenis Moreno Pérez

Tutor: Ing. Sergio Michel Rivera Rodríguez.

Co-Tutor: Ing. Meylin Martínez Chong.

“Año 50 de la Revolución”.

Ciudad de la Habana, Cuba

Junio, 2008

DECLARACIÓN DE AUTORÍA

Declaro que soy el único autor de la presente tesis, y reconozco a la Universidad de las Ciencias Informáticas los derechos de poder hacer uso del mismo en su beneficio.

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

Firma del Autor

Mislenis Moreno Pérez

Firma del Tutor

Ing. Sergio M. Rivera Rodríguez

DATOS DE CONTACTO

Síntesis del tutor: Sergio Michel Rivera Rodríguez.

Profesión: Ingeniero informático.

Años de graduado: 4.

Síntesis del Co-Tutor: Meylin Martínez Chong.

Profesión: Ingeniero informático.

Años de graduado: 2.

AGRADECIMIENTO

Quiero agradecer a la Revolución y a nuestro compañero Fidel Castro por hacer posible que exista la Universidad de las Ciencias Informáticas que significó una casa para mí en estos años de estudio y que me permite hoy graduarme como ingeniero en informática.

A mi tutor Sergio y a Meylín por su ayuda constante y por estar siempre cuando los necesitaba y en especial por su paciencia para conmigo, muchas gracias de corazón.

Muchas gracias a mi familia y a mi novio por estar conmigo en las buenas y en las malas y por confiar tanto en que podía lograr este sueño que hoy quiero compartir con ellos.

A mis amigos Elisa, Xonia, Perla Mailén, Raúl Carralero, Damir, Osiris, Eloy, Janiel, Pedro E Navia, Yosmel, Lisbey, Ricardo y Yoleidis por ser mis amigos, algunos lo más probable no los vuelva a ver pero donde quiera que estén siempre van a estar conmigo, de corazón los quiero mucho a todos.

Gracias a la Universidad y a mis profesores por hacerme la profesional que soy hoy.

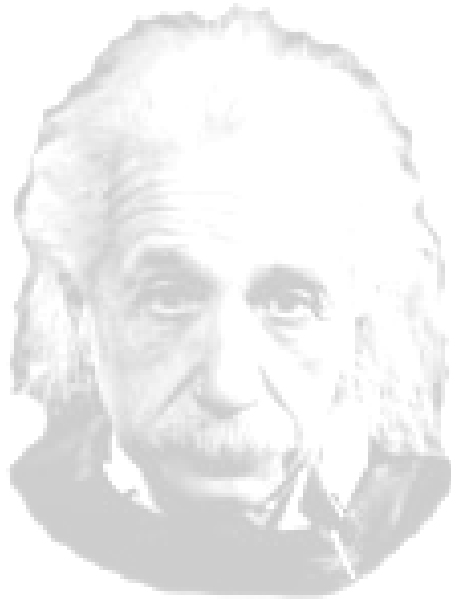
DEDICATORIA

Quiero dedicar este trabajo en especial a mi mamá Bárbara Pérez Delgado por ser todo para mí, mi apoyo, mi consuelo, mi fuerza a seguir a delante, también a mis hermanos Maileny y Oscar, a mis sobrinos Leonardo, Leodan, Christian y Cynthia.

También quiero dedicárselo a mi novio Dennis Molinet que gracias a él, hoy puedo hacer realidad este sueño, porque siempre creyó en mí cuando a veces hasta yo misma lo dudaba, me ayudó mucho en estos años de carrera que estuvimos juntos y siempre está conmigo para todo.

A todas esas personas que de una forma u otra me ayudaron a ser una mejor persona y me han guiado a ser lo que soy, algunos no se encuentran ya conmigo y otros si, para todos también le dedico este trabajo.

PENSAMIENTO



"Nunca consideres el estudio como una obligación sino como una oportunidad para penetrar en el bello y maravilloso mundo del saber."

Albert Einstein

RESUMEN

El proceso de la demanda para la lucha armada se realiza todos los años y en todo el país. La misma se elabora en todas las unidades militares (tropas regulares de las FAR, las MTT y los órganos del MININT), con el objetivo de solicitar todos los productos o servicios que estas necesitan para en caso de ocurrir alguna situación excepcional, entiéndase por esta una guerra, fenómenos naturales o alguna crisis por ejemplo un período especial, el país poder contar con todos los planes de producción y servicios que este necesita para su buen funcionamiento. Dicho proceso se enmarca en la “Metodología para el proceso de formulación, presentación y satisfacción de las demandas para la lucha armada”, la cual tiene como objetivos establecer los lineamientos generales y particulares de cada especialidad, las nomenclaturas, normas y procedimientos a emplear en la formulación de las demandas para la lucha armada, así como, definir los pasos a tenerse en cuenta para la realización del mismo.

En el presente trabajo se realizará el análisis y diseño de dicho proceso, realizando primero un estudio general de como es que funciona el mismo y exponiéndose posteriormente las características del sistema que se propone.

Palabras claves:

- Proceso de la demanda para la lucha armada.
- Metodología para el proceso de formulación, presentación y satisfacción de las demandas para la lucha armada.

AGRADECIMIENTO	III
DEDICATORIA	IV
PENSAMIENTO.....	V
RESUMEN.....	VI
INTRODUCCIÓN	1
CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA.....	4
1.1 Introducción	4
1.2 Tendencias y tecnologías actuales.....	4
1.2.1 Sistemas informáticos similares existentes vinculados al campo de acción:.....	4
1.2.2 Aplicación Web.....	5
1.2.3 Lenguajes de Programación.....	6
1.2.4 Servidor Web Apache.....	12
1.2.5 Sistema Gestor de Base de Datos (SGBD)	13
1.2.6 Metodologías de desarrollo de software	14
1.2.7 Lenguaje Unificado de Modelado (UML).....	17
1.2.8 Herramientas CASE de Modelado con UML.....	18
1.2.9 Notación Polaca Inversa.....	19
1.2.9.1 Convención de una cadena infija en una polaca postfija.....	20
1.2.9.2 Evaluación numérica de cadenas polacas mediante el uso de Pilas:.....	22
1.2.9.3 Evaluación numérica de cadenas polacas mediante el uso de Árbol:	22
1.3 Conclusiones	23
CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA	24
2.1 Introducción.	24
2.2 Objeto de estudio.	24
2.2.1 Problema y situación problemática.....	25
2.2.1.1 Objetivo estratégico de la organización.	25
2.2.1.2 Flujo actual del proceso involucrado en el campo de acción.....	25
2.2.2 Objeto de automatización.	26
2.2.3 Documentación que se maneja.	26
2.2.4 Propuesta de sistema.....	26
2.2.5 Modelación del negocio.....	27

2.2.5.1 Actores del negocio.	27
2.2.5.2 Trabajadores del negocio.	27
2.2.5.3 Diagrama de Casos de Usos del negocio.	28
2.2.5.4 Descripción del Caso de Uso del negocio.	28
2.2.5.5 Diagrama de actividades.	29
2.2.5.6 Modelo de objetos.	30
2.2.6 Especificación de los requisitos de software.	30
2.2.6.1 Definición de los requerimientos funcionales.	30
2.2.6.2 Definición de los requerimientos no funcionales.	31
2.2.7 Definición de los casos de uso.	33
2.2.7.1 Definición de los actores.	33
2.2.7.2 Listado de casos de usos.	33
2.2.7.3 Diagrama de Casos de Uso.	36
2.2.7.4 Caso de usos expandidos.	36
2.3 Conclusiones	52
CAPÍTULO 3: ANÁLISIS Y DISEÑO DEL SISTEMA	54
3.1 Introducción	54
3.2 Modelo de análisis	54
3.2.1 Diagramas de clases del análisis.	54
3.3 Arquitectura del sistema	59
3.4 Modelo de diseño	62
3.4.1 Diagramas de secuencia	62
3.4.2 Diagramas de clases de diseño Web	70
3.4.3 Diseño de la Base de Datos	76
3.4.3.1 Modelo lógico de datos	77
3.4.3.2 Modelo físico de datos	78
3.4.3.3 Descripción de las tablas	78
3.5 Principios de diseño	84
3.5.1 Tratamiento de errores.	85
3.5.2 Mecanismo de diseño	86
3.5.3 Seguridad	89

3.5.4 Interfaz de Usuario	92
3.5.5 Ayuda.....	92
3.6 Patrones de Diseño.....	92
3.7 Conclusiones	94
CONCLUSIONES	95
RECOMENDACIONES	96
REFERENCIAS BIBLIOGRÁFICAS	97
BIBLIOGRAFÍA.....	99
GLOSARIO DE TÉRMINOS.....	100
ANEXOS.....	102

INTRODUCCIÓN

Las demandas para la defensa se fundamentan en la Ley de la Defensa Nacional, la cual establece en su artículo 102 que la preparación de la economía nacional para la defensa se realiza mediante la ejecución de un conjunto de medidas de organización, y movilización, económicas, financieras, técnicos-materiales y de creación de las reservas necesarias para garantizar desde tiempos de paz el cumplimiento de los planes de producción y servicio establecidos para las situaciones excepcionales.

El proceso de formulación, presentación y satisfacción de las demandas para la lucha armada, como parte de las demandas para la defensa, constituye el conjunto de actividades que realizan las FAR junto a el MININT y las entidades económicas desde tiempo de paz, con el objetivo de planificar las necesidades materiales y financieras de la lucha armada y la posible satisfacción de estas por parte de la economía.[1]

El MINFAR, organiza y controla el proceso de formulación de las demandas para la lucha armada y es el encargado de determinar los niveles de actividad así como elaborar las propuestas de nomenclaturas a solicitar, sus normas y variantes alternativas.

Los recursos a planificar se clasifican en:

Nomenclatura territorial: cuya satisfacción se puede decidir territorialmente en el municipio o en la provincia de no existir en el municipio.

Nomenclatura central: cuya satisfacción se decide por los Órganos de Administración Central del Estado (OACE) conjuntamente por el MINFAR. [1]

Las demandas para la lucha armada son formuladas por: las tropas regulares de las FAR, las MTT y los órganos del MININT. Las mismas antes de su presentación deberán ser revisadas por las especialidades de las instancias superiores para comprobar la correcta utilización de las normas, empleo de de las nomenclaturas, y la utilización de los recursos y/o servicios que por los niveles superiores se le asignan para satisfacer sus necesidades.[1]

Cálculo y formulación de las demandas para la lucha armada:

Para el cálculo de las demandas, además de las normas de cada especialidad, debe contarse con las bases de cálculo, entre ellas las cifras del balance de personal por cada categoría; la existencia de

armamento, técnica y medios materiales de todo tipo; situación que presenta el acondicionamiento ingeniero del Teatro de Operaciones Militares (TOM) misiones que cumplen, estructuras y órganos que se crean para ello y otras cuestiones.

Los cálculos de las demandas para la lucha armada no pueden simplificarse a una operación aritmética de multiplicación de indicadores de consumo por niveles de actividades, pues aunque este paso es ineludible, conviene reiterar que antes, durante y después del mismo, se debe trabajar en todos los aspectos que depuren las demandas de elementos subjetivos y las conviertan en una solicitud necesaria y racional, que debe ser actualizada sistemáticamente a todos los niveles.

Para la determinación de las demandas de la lucha armada y su presentación a la economía, existen dos momentos independientes y con un mismo objetivo:

1. La determinación de las necesidades de recursos y/o servicios que necesitan para el aseguramiento multilateral de sus acciones.

A estas necesidades se les deberá descontar aquellos recursos y servicios que se aseguran con las existencias de consumo, las reservas creadas para las operaciones, los recursos que se obtienen mediante las producciones propias y los servicios que se garantizan por las instituciones de las FAR, así como los que se entregan por el sistema empresarial de las FAR en interés de las unidades.

2. Las unidades determinan el déficit de sus necesidades de recursos y/o servicios que no se pueden asegurar por las instituciones de las FAR y formulan las demandas a la economía a través de la Región Militar. [1]

Por lo antes expuesto se puede apreciar que en dicho proceso actualmente se trabaja con un gran volumen de documentación el cual se archivan en papeles y el manejo de los datos y los cálculos se realizan manualmente, haciéndose tedioso el trabajo con el mismo, pudiendo ocurrir que la información pueda extraviarse, o deteriorarse, además de ocurrir cualquier tipo de error a la hora de manejar tantos datos, por lo cual no va haber una confiabilidad, seguridad y calidad, además que el proceso será mas lento e inexacto.

Por lo antes expuesto el **problema a resolver** sería: el proceso actual de planificación (formulación) de la demanda no garantiza el control, la seguridad y calidad de la información, que fluye a través de él.

Enmarcando así el **objeto de estudio** como: Proceso que se dicta en la Metodología de formulación, presentación y satisfacción de las demandas para la lucha armada.

El **campo de acción** de este problema es: el proceso de planificación de la demanda para la lucha armada en las FAR.

Como **objetivo general** de este trabajo se tiene: Informatizar el proceso de planificación de la demanda para la lucha armada.

Para darle cumplimiento al objetivo general trazado se han decidido desarrollar las siguientes **tareas**:

- ✓ Hacer un estudio completo de cómo es que funciona el negocio, para lograr una mayor comprensión de la estructura y la dinámica del proceso de planificación de las demandas para la lucha armada.
- ✓ Estudiar las principales tecnologías y herramientas que se están empleando a nivel internacional.
- ✓ Realizar la modelación del proceso de planificación de las demandas para la lucha armada.
- ✓ Realizar el análisis y diseño del sistema.

Este Trabajo de Diploma está formado por tres capítulos cuyos objetivos son:

- Capítulo I. Fundamentación teórica: se expone el estado del arte del problema antes mencionado a nivel nacional, tendencias, técnicas, tecnologías, metodologías y software que se usan en la actualidad.
- Capítulo II. Características del sistema: se explica cual es la situación problémica y la propuesta del sistema que dará solución a la misma, especificando los requisitos funcionales y no funcionales que deberá cumplir el sistema.
- Capítulo III. Análisis y diseño del sistema: se crea el modelo de análisis, el modelo de clases de análisis, el modelo de clases del Diseño y los diagramas de interacción por cada realización de casos de uso, la descripción de las clases y el diseño de la Base de Datos, con la descripción de las tablas de la Base de Datos.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

1.1 Introducción

En el siguiente capítulo se persigue como principal objetivo, realizar un estudio de las herramientas y tecnologías previamente seleccionadas por el grupo de arquitectura del centro “Unidad de compatibilización, integración y desarrollo del software para la defensa”, de las mismas se tendrán en cuenta cuales son sus principales características y las ventajas que estas reportan. Además se analizarán aquellos sistemas informáticos que estén vinculados al proceso de planificación de la demanda, realizados hasta el momento a nivel nacional.

1.2 Tendencias y tecnologías actuales

A continuación se describen algunas de las tendencias y tecnologías actuales posibles a utilizar para darle solución a los problemas planteados anteriormente de manera eficiente, teniendo en cuenta que el sistema es interno de las FAR, se abordarán aquellas herramientas previamente definidas por el grupo de arquitectura del centro.

1.2.1 Sistemas informáticos similares existentes vinculados al campo de acción:

El sistema más cercano del que se tiene referencia es el realizado por el hoy Teniente Coronel Julio C. Cardoso Torres en la Región Militar de Ciudad Habana, hecho en lenguaje FOXPRO.

Dicho sistema resolvía la demanda por productos, la consolidación en todos los niveles y la satisfacción por las empresas.

Debido a cambios realizados en las metodologías por parte de la Dirección de Logística en el MINFAR, a la tarea se le planteaban misiones cada vez más complejas, como lo es la realización de la planificación a partir de variables definidas por la Logística.

Posteriormente a esta tarea, se realizó un sistema por el hoy Teniente Coronel Oliva de la Dirección de Logística en EXCEL, el cual fue aplicado en todas las unidades demandantes del país, resolviendo el problema de la planificación. Esta planificación se enviaba a la Región y allí, se resumía y comprobaba por el programa de FOXPRO.

La dificultad que presentaba la planificación en EXCEL es que las normas, fórmulas de cálculo por etapas y otras características de los productos se reflejaban en los ficheros y podían ser manipuladas con un mínimo conocimiento de EXCEL, además esta información viajaba en documentos impresos hasta la

Región, donde se pasaban los cálculos a mano para que el programa de FOXPRO los resumiera e hiciera su tratamiento con los datos.

Es apreciable lo incómodo, difícil e inseguro que resultaba realizar la planificación, comprobación y seguimiento de las demandas en todos los niveles.

Como es un sistema interno de las Fuerzas Armadas Revolucionarias de Cuba (FAR), este no tiene alcance a nivel internacional, aunque no se descarta la posibilidad de cooperación con otros países.

1.2.2 Aplicación Web

Las aplicaciones Web se encuadran dentro de la arquitectura cliente/servidor (conocida como arquitectura de dos capas): un ordenador solicita servicios (el cliente) y otro está a la espera de recibir solicitudes y responderlas (el servidor).

La comunicación entre los clientes web y los servidores web es realizada haciendo uso del HTTP (Hyper Text Transfer Protocol), este protocolo de transferencia de hipertexto es el sistema que permite enviar tanto las peticiones de un cliente a un servidor para acceder a una determinada página web como las páginas web que envía el servidor como respuesta al cliente.

En el caso de hacerse uso de servidores de bases de datos en la aplicación web, la arquitectura cliente – servidor sería de tres capas, siendo la tercera capa el servidor de bases de datos.

¿Qué características debe cumplir una aplicación Web?

- Para que una aplicación Web sea eficaz, la misma debe contener algunas cualidades indispensables: seguridad, escalabilidad, portabilidad y un diseño eficiente del interfaz y el programa. La escalabilidad y la portabilidad son las que con más frecuencia caen en el olvido. Escalabilidad significa que la aplicación tendrá la capacidad de crecer junto con las necesidades de sus usuarios. Portabilidad va ligado al concepto de escalabilidad, pues, se refiere a la compatibilidad de la aplicación con los múltiples sistemas operativos y plataformas existentes en el mercado. En general, cuando las demandas de los usuarios crecen, la aplicación debe cambiar de plataforma para pasar de un servidor pequeño a uno mediano o grande. Es entonces, cuando la portabilidad es crítica para poder “escalar” sin problemas, independientemente del entorno que se requiera. [2]

Entre las ventajas más importantes de las aplicaciones web se encuentran:[3]

- Varias tecnologías incluyendo Java, Flash, ASP y Ajax permiten un desarrollo efectivo de programas soportando todos los sistemas operativos principales.
- Las aplicaciones basadas en web no necesitan ser descargadas, instaladas y configuradas.
- Las aplicaciones basadas en web pueden ser utilizadas por múltiples usuarios al mismo tiempo, que pueden ver e incluso editar el mismo documento de manera conjunta.
- Una empresa puede migrar de sistema operativo o cambiar el hardware libremente sin afectar el funcionamiento de las aplicaciones de servidor.
- No se requieren de complicadas combinaciones de hardware/software para utilizar estas aplicaciones.
- Se facilita el trabajo a distancia, debido a que se puede trabajar desde cualquier PC o computador portátil con conexión a internet o a la intranet donde reside la aplicación.

1.2.3 Lenguajes de Programación

Un lenguaje de programación es un lenguaje que puede ser utilizado para controlar el comportamiento de una máquina, particularmente una computadora. Consiste en un conjunto de reglas sintácticas y semánticas que definen su estructura y el significado de sus elementos, respectivamente.

Un lenguaje de programación permite a un programador especificar de manera precisa: sobre qué datos una computadora debe operar, cómo deben ser estos almacenados y transmitidos y qué acciones debe tomar bajo una variada gama de circunstancias. Todo esto, a través de un lenguaje que intenta estar relativamente próximo al lenguaje humano o natural, tal como sucede con el lenguaje léxico.

Un programa escrito en un lenguaje de programación necesita pasar por un proceso de compilación, es decir, ser traducido al lenguaje de máquina, o ser interpretado para que pueda ser ejecutado por el ordenador.[4]

Lenguajes compilados: son aquellos cuya función es traducir un programa escrito en un determinado lenguaje a un idioma que la computadora entienda (lenguaje máquina con código binario).

Lenguajes interpretados: a diferencia de los lenguajes compilados en los lenguajes interpretados el código no necesita ser procesado por un compilador, eso significa que el ordenador es capaz de ejecutar la sucesión de instrucciones dadas por el programador sin necesidad de leer y traducir exhaustivamente todo el código. Para ello hace falta un intermediario (intérprete) el cual se encarga de leer una a una las

instrucciones textuales del programa, conforme éstas necesitan ser ejecutadas y descomponerlas en instrucciones del sistema.[4]

PHP:

PHP es el acrónimo de Hipertext Preprocesor, es un lenguaje de programación del lado del servidor, gratuito e independiente de plataforma, rápido, con una gran librería de funciones y mucha documentación que facilitan muchísimo el desarrollo de las aplicaciones. Es un lenguaje de script interpretado en el lado del servidor utilizado para la generación de páginas web dinámicas, embebidas en páginas HTML y ejecutadas en el servidor. PHP no necesita ser compilado para ejecutarse y algunas de las más importantes capacidades que tiene son: la compatibilidad con las bases de datos más comunes, MySQL, PostgreSQL, Oracle, Microsoft SQL Server, Firebird, lo cual facilita la creación de aplicaciones Web muy robustas, además incluye funciones para el envío de correo electrónico, descarga de archivos, crear dinámicamente en el servidor imágenes en formato GIF, incluso animadas.

El fácil uso y la similitud con los lenguajes de programación estructurada, como por ejemplo C y Perl, permiten a la mayoría de los programadores experimentados crear aplicaciones complejas con una curva de aprendizaje muy suave. También les permite involucrarse con aplicaciones de contenido dinámico sin tener que aprender todo un nuevo grupo de funciones y prácticas.

Su interpretación y ejecución se da en el servidor Web, en el cual se encuentra almacenado el script, y el cliente sólo recibe el resultado de la ejecución. Cuando el cliente hace una petición al servidor para que le envíe una página Web, generada por un script PHP, el servidor ejecuta el intérprete de PHP, el cual procesa el script solicitado que generará el contenido de manera dinámica, pudiendo modificar el contenido a enviar, y regresa el resultado al servidor, el cual se encarga de regresárselo al cliente. (Ver Anexo1)

Ventajas de PHP:

- Muy fácil de aprender.
- Se caracteriza por ser un lenguaje muy rápido.
- Es un lenguaje multiplataforma: Linux, Windows, entre otros.
- Capacidad de conexión con los manejadores de base de datos más comunes: MySQL, PostgreSQL, Oracle, Microsoft SQL Server.
- Capacidad de expandir su potencial utilizando módulos.

- Posee documentación en su página oficial la cual incluye descripción y ejemplos de cada una de sus funciones.
- Es libre, por lo que se presenta como una alternativa de fácil acceso para todos.
- Incluye gran cantidad de funciones.
- No requiere definición de tipos de variables. [5]

Lenguaje Javascript

JavaScript es un lenguaje de programación interpretado, es decir, que no requiere compilación, utilizado principalmente en páginas web, con una sintaxis semejante a la del lenguaje Java y el lenguaje C.

Se trata de un lenguaje de programación del lado del cliente, porque es el navegador el que soporta la carga de procesamiento. Gracias a su compatibilidad con la mayoría de los navegadores modernos, es el lenguaje de programación del lado del cliente más utilizado. [6]

Entre las acciones típicas que se pueden realizar en Javascript tenemos:

- efectos especiales sobre páginas Web, crear contenidos dinámicos y elementos de la página que tengan movimiento, cambiar de color o cualquier otro dinamismo;
- ejecutar instrucciones como respuesta a las acciones del usuario, con lo que se pueden crear páginas interactivas como calculadoras, agendas o tablas de cálculo.

El navegador del cliente es el encargado de interpretar y ejecutar las instrucciones Javascript para realizar estos efectos e interactividades. [6]

Además favorece muchas posibilidades: permite la programación de pequeños scripts, pero también de programas más grandes con funciones, estructuras de datos complejas, etc.; pone a disposición del programador todos los elementos que forman la página Web para que éste pueda acceder a ellos y modificarlos dinámicamente. [6]

Ventajas:

Entre las ventajas que presenta este lenguaje se encuentra que es **fácil de aprender, rápido y potente**: JavaScript es muy sencillo de aprender ya que incluye los elementos típicos de un lenguaje de programación: variables, sentencias, estructuras, operadores, etc. Es ideal para agregar ciertas funciones rápidas a una página Web, y no necesita compilarse ya que es interpretado por el navegador de cliente. El

hecho de que casi todos los navegadores lo soportan es otra de las grandes ventajas que presenta (**usabilidad**), además de, la **reducción de la carga del servidor**: JavaScript se puede hacer cargo de gran parte de las funciones del cliente de las cuales se encargaba el servidor un ejemplo es la validación, antiguamente el usuario introducía la información de la validación y se la enviaba en código HTML al servidor, que por medio de CGI¹ se aseguraba de que todo era correcto y ahora con el JavaScript es posible validar los elementos antes de que el usuario se los envíe al servidor reduciendo de esta forma la cantidad de transacciones que se efectúan a través de http² y las posibilidades de que se genere un error durante la inserción de los datos. [7]

XML (Extensible Markup Language – Lenguaje de marcado extensible)

Los documentos creados utilizando un lenguaje de marcado se compone de caracteres de marcado y el texto. Los caracteres de marcado definen la forma en que una aplicación que lea el documento debe interpretar el texto, sin embargo en XML el único elemento predefinido es el propio lenguaje de marcado, el diseñador de un documento XML define la estructura del documento y los caracteres de marcado. Esta característica dota a XML de flexibilidad y permite que los datos del documento interpretado puedan utilizarse para una gran variedad de fines.[8]

El XML se creó para que cumpliera varios objetivos.

- Que fuera idéntico a la hora de servir, recibir y procesar la información que el HTML, para aprovechar toda la tecnología implantada para este último.
- Que fuera formal y conciso desde el punto de vista de los datos y la manera de guardarlos.
- Que fuera extensible, para que lo puedan utilizar en todos los campos del conocimiento.
- Que fuese fácil de leer y editar.
- Que fuese fácil de implantar, programar y aplicar a los distintos sistemas. [9]

XML aporta muchas ventajas, algunas de ellas son:

¹ CGI: "Common Gateway Interface". Un script CGI es una pequeña aplicación que se ejecuta en el servidor, normalmente a petición de una página web alojada en el mismo. Los CGI proporcionan la posibilidad de acceder a bases de datos, intercambio de información a través de formularios HTML, gestión de claves de accesos, utilidades de búsqueda, etc.

² HTTP (HyperText Transfer Protocol). Protocolo estándar de transmisión de páginas Web en Internet.

- Comunicación de datos. Si la información se transfiere en XML, cualquier aplicación podría escribir un documento de texto plano con los datos que estaba manejando en formato XML y otra aplicación recibir esta información y trabajar con ella.
- Aplicaciones web. Hasta ahora cada navegador interpreta la información a su manera y los programadores de web deben trabajar en función del navegador del usuario. Con XML se tiene una sola aplicación que maneja los datos y para cada navegador o soporte se podrá tener una hoja de estilo o similar para aplicarle el estilo adecuado. [9]
- XML permite al programador dedicar sus esfuerzos a las tareas importantes cuando trabaja con los datos, ya que algunas tareas tediosas como la validación de estos o el recorrido de las estructuras corre a cargo del lenguaje y está especificado por el estándar, de modo que el programador no tiene que preocuparse por ello.

HTML son las siglas de HyperText Markup Language o, lo que es lo mismo, Lenguaje Marcador de hipertexto, está basado en el SGML (Standard Generalized Markup Language), que significa Lenguaje Marcador Estándar Generalizado y que se regularizó y recopiló en la norma ISO Estándar 8879:1986. [10]

HTML es un simple lenguaje de etiquetas ideado para crear documentos de hipertexto que se pueden transferir de una plataforma o sistema a otra plataforma o sistema. Consta de un conjunto de elementos, denominados etiquetas o marcas, que el manejo de las mismas permite:

- Definir la estructura lógica del documento HTML.
- Aplicar distintos estilos al texto (negrita, cursiva, etc.)
- La inclusión de hiperenlaces, que nos permitirán acceder a otros documentos relacionados con el actual.
- La inclusión de imágenes y ficheros multimedia (gráficos, vídeo, audio).

Ventajas que aporta este lenguaje:

- los documentos HTML son de escaso tamaño, por lo que pueden transferirse por la red a la mayor brevedad.
- Los documentos HTML son independientes de los dispositivos, es decir, se despliegan en cualquier plataforma; todo lo que necesita es un visualizador para la plataforma en la que trabaje, que sea capaz de entender HTML.

- A nivel de su programación, es un lenguaje muy sencillo para aprender y de breve y fácil manejo. Son muy pocas las etiquetas e instrucciones que se tienen que recordar y además ya hay en el mercado editores que facilitan su manejo. [10]

AJAX, acrónimo de Asynchronous JavaScript And XML (JavaScript y XML asíncronos), es una técnica de desarrollo web para crear aplicaciones interactivas mediante la combinación de tres tecnologías ya existentes:[11]

- HTML y hojas de estilos en cascada (CSS) para presentar la información;
- Document Object Model (DOM) y JavaScript, para interactuar dinámicamente con los datos.
- XML para intercambiar y manipular datos con el servidor web (aunque las aplicaciones AJAX pueden usar otro tipo de tecnologías, incluyendo texto plano, para realizar esta labor).

Como el DHTML³ o LAMP⁴, AJAX no constituye una tecnología en sí, pero es un término que engloba a un grupo de éstas que trabajan conjuntamente.

Las aplicaciones AJAX usan navegadores web que soportan las tecnologías mencionadas más arriba. Entre estos se incluyen Mozilla Firefox, Internet Explorer, Opera, Konqueror y Safari.[11]

Razones para usar AJAX

- Basado en los estándares abiertos: Ajax está formado por las tecnologías Javascript, HTML, XML, CSS, y XML HTTP Request Object, siendo este último el único que "no es" estándar pero es soportado por los navegadores más utilizados de internet como son los basados en Mozilla Firefox, Internet Explorer, Safari y Opera.
- Usabilidad: Permite a las páginas hacer una pequeña petición de datos al servidor y recibirla sin necesidad de cargar la página entera. El incremento de las actualizaciones elimina el tener que refrescar el navegador, algo bastante apreciado a la hora de operar en una aplicación web.

³ Dynamic HTML. Una extensión de HTML que permite, entre otras, incluir pequeñas animaciones y menús dinámicos en páginas Web. El código DHTML utiliza hojas de estilo y JavaScript.

⁴ Se origina a finales del 2000 en Alemania para describir a las aplicaciones web creadas utilizando la siguiente combinación de herramientas:

- Linux, el sistema operativo;
- Apache, el servidor web;
- MySQL, el servidor de bases de datos;
- Perl, PHP, y/o Python, lenguajes de programación.

- Válido en cualquier plataforma y navegador: Es una de las razones más importantes por las que AJAX se ha vuelto tan popular. Aunque si bien muchos desarrolladores sabían que era posible usarse años atrás con Internet Explorer, no era viable realizarse. Ahora ya es posible su avance gracias a Mozilla Firefox.
- Beneficia las aplicaciones web: AJAX es la cara del presente en las aplicaciones web. Esto incluye un menor coste de creación, facilidad de soporte y mantenimiento, menores tiempos a la hora de desarrollarlas, y sin necesidad de instalaciones.
- Es independiente del tipo de tecnología de servidor que se utilice: Así como AJAX funciona en cualquier navegador, es perfectamente compatible con cualquier tipo de servidor estándar y lenguaje de programación Web. PHP, ASP. ASP.Net, Perl, JSP. El ser completamente compatible el desarrollo en estas tecnologías ha ayudado a AJAX a que vaya cada vez más en auge.
- Mejora la estética de la web: Con AJAX se puede interactuar la imaginación del desarrollador con la usabilidad de una aplicación web de forma que se pueda realizar una aplicación que si no estuviera dentro de un navegador, podría pasar por una aplicación de escritorio. [12]

1.2.4 Servidor Web Apache

Apache es el servidor web hecho por excelencia, su configurabilidad, robustez y estabilidad hacen que cada vez millones de servidores reiteren su confianza en este programa, y las razones de su popularidad se deben a:

- Corre en una multitud de Sistemas Operativos, lo que lo hace prácticamente universal.
- Apache es una tecnología gratuita de código fuente abierto. El hecho de ser gratuita es importante pero no tanto como que se trate de código fuente abierto. Esto le da una transparencia a este software de manera que si se quiere ver lo que se está instalando como servidor, se puede saber, sin ningún secreto, sin ninguna puerta trasera.
- Apache es un servidor altamente configurable de diseño modular. Es muy sencillo ampliar las capacidades del servidor Web Apache. Actualmente existen muchos módulos para Apache que son adaptables a este, y están ahí para que se instalen cuando se necesiten. Otra cosa importante es que cualquiera que posea una experiencia decente en la programación de C o Perl puede escribir un modulo para realizar una función determinada.

- Apache te permite personalizar la respuesta ante los posibles errores que se puedan dar en el servidor. Es posible configurar Apache para que ejecute un determinado script cuando ocurra un error en concreto.
- Tiene una alta configurabilidad en la creación y gestión de logs. Apache permite la creación de ficheros de log a medida del administrador, de este modo puedes tener un mayor control sobre lo que sucede en tu servidor. [13]

1.2.5 Sistema Gestor de Base de Datos (SGBD)

Las principales funciones que debe cumplir un SGBD se relacionan con la creación y mantenimiento de la base de datos, el control de accesos, la manipulación de datos de acuerdo con las necesidades del usuario, el cumplimiento de las normas de tratamiento de datos, evitar redundancias e inconsistencias y mantener la integridad. Algunos ejemplos de SGBD son Oracle, PostgreSQL, MySQL, Microsoft SQL Server, etc.

PostgreSQL: Es un sistema de gestión de bases de datos objeto-relacional (ORDBMS) basado en el proyecto POSTGRES, de la universidad de Berkeley. Está considerado como el gestor de base de datos de código abierto (Open Source) más avanzada del mundo, y es un sistema objeto-relacional, ya que incluye características de la orientación a objetos, como puede ser la herencia, tipos de datos, funciones, restricciones, disparadores, reglas e integridad transaccional.

Entre sus principales características se pueden mencionar: [14]

- DBMS Objeto-Relacional: PostgreSQL aproxima los datos a un modelo objeto-relacional, y es capaz de manejar complejas rutinas y reglas. Ejemplos de su avanzada funcionalidad son consultas SQL declarativas, control de concurrencia multi-versión, soporte multi-usuario, optimización de consultas, herencia, y arrays.
- Altamente Extensible: PostgreSQL soporta operadores, funciones, métodos de acceso y tipos de datos definidos por el usuario.
- Soporte SQL Comprensivo: PostgreSQL soporta la especificación SQL99 e incluye características avanzadas tales como las uniones (joins) SQL92.
- Integridad Referencial: PostgreSQL soporta integridad referencial, la cual es utilizada para garantizar la validez de los datos de la base de datos.

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

¿Por que usar PostgreSQL?

- Multiplataforma: Corre en casi todos los principales sistemas operativos: Linux, Unix, Mac OS, Windows, etc.
- Extensible: El código fuente está disponible para todos sin costo. Esto es complementado por la comunidad de profesionales y entusiastas de PostgreSQL alrededor del mundo que también extienden PostgreSQL constantemente.
- Estabilidad y confiabilidad legendarias: En contraste a muchos sistemas de bases de datos comerciales, es extremadamente común que compañías reporten que PostgreSQL nunca ha presentado caídas en varios años de operación de alta actividad.
- Instalación ilimitada: Con PostgreSQL, nadie puede demandarlo por violar acuerdos de licencia, puesto que no hay costo asociado a la licencia del software.
- Diseñado para ambientes de alto volumen: PostgreSQL usa una estrategia de almacenamiento de filas llamada MVCC⁵ para conseguir una mejor respuesta en ambientes de grandes volúmenes. Los principales proveedores de sistemas de bases de datos comerciales usan también esta tecnología, por las mismas razones. [15]

1.2.6 Metodologías de desarrollo de software

Las metodologías de desarrollo de software son un conjunto de procedimientos, técnicas, herramientas y un soporte documental que ayuda a los desarrolladores a construir un software. Por otra parte, define quién debe hacer qué, cuándo y cómo para alcanzar un determinado objetivo. También podemos decir que es un proceso, y en su modelación se definen como elementos principales los siguientes:

⁵ MVCC (Control de Concurrencia Multi Versión): es una técnica avanzada para mejorar las prestaciones de una base de datos en un entorno multiusuario.

- Trabajadores (quién): Define el comportamiento y responsabilidades (rol) de un individuo, grupo de individuos, sistema automatizado o máquina, que trabajan en conjunto como un equipo. Ellos realizan las actividades y son propietarios de elementos.
- Actividades (cómo): Es una tarea que tiene un propósito claro, es realizada por un trabajador y manipula elementos.
- Artefactos (qué): Productos tangibles del proyecto que son producidos, modificados y usados por las actividades. Pueden ser modelos, elementos dentro del modelo, código fuente y ejecutables.
- Flujo de actividades (cuándo): Secuencia de actividades realizadas por trabajadores y que producen un resultado de valor observable. [16]

Rational Unified Process (RUP)

En RUP (Proceso Unificado de Rational) se han agrupado las actividades en grupos lógicos definiéndose 9 flujos de trabajo principales. Los 6 primeros son conocidos como flujos de ingeniería y los tres últimos como de apoyo. En la siguiente figura se representa el proceso en el que se grafican los flujos de trabajo y las fases y muestra la dinámica expresada en iteraciones y puntos de control. (Ver Anexo2)

Flujos de trabajo:

- Business Modeling (Modelado del negocio): Describe los procesos de negocio, identificando quiénes participan y las actividades que requieren automatización.
- Requirements (Requerimientos): Define qué es lo que el sistema debe hacer, para lo cual se identifican las funcionalidades requeridas y las restricciones que se imponen.
- Analysis & Design (Análisis y Diseño): Describe cómo el sistema será realizado a partir de la funcionalidad prevista y las restricciones impuestas (requerimientos), por lo que indica con precisión lo que se debe programar.
- Implementation (Implementación): Define cómo se organizan las clases y objetos en componentes, cuáles nodos se utilizarán y la ubicación en ellos de los componentes y la estructura de capas de la aplicación.
- Test (Prueba): Busca los defectos a lo largo del ciclo de vida.
- Deployment (Instalación): Produce release del producto y realiza actividades (empaque, instalación, asistencia a usuarios, etc.) para entregar el software a los usuarios finales.

- Project Management (Administración del proyecto): Involucra actividades con las que se busca producir un producto que satisfaga las necesidades de los clientes.
- Configuration & Change Mgmt (Administración de configuración y cambios): Describe cómo controlar los elementos producidos por todos los integrantes del equipo de proyecto en cuanto a: utilización/actualización concurrente de elementos, control de versiones, etc.
- Environment (Ambiente): Contiene actividades que describen los procesos y herramientas que soportarán el equipo de trabajo del proyecto; así como el procedimiento para implementar el proceso en una organización. [16]

Fases:

- Inception (Concepción o Inicio): Se describe el negocio y se delimita el proyecto describiendo sus alcances con la identificación de los casos de uso del sistema.
- Elaboration (Elaboración): Se define la arquitectura del sistema y se obtiene una aplicación ejecutable que responde a los casos de uso que la comprometen.
- Construction (Construcción): Se obtiene un producto listo para su utilización que está documentado y tiene un manual de usuario. Se obtiene 1 o varios release del producto que han pasado las pruebas. Se ponen estos release a consideración de un subconjunto de usuarios.
- Transition (Transición): El release ya está listo para su instalación en las condiciones reales. Puede implicar reparación de errores. [16]

El ciclo de vida de RUP se caracteriza por:

1. Dirigido por casos de uso: Los casos de uso reflejan lo que los usuarios futuros necesitan y desean, lo cual se capta cuando se modela el negocio y se representa a través de los requerimientos. A partir de aquí los casos de uso guían el proceso de desarrollo ya que los modelos que se obtienen, como resultado de los diferentes flujos de trabajo, representan la realización de los casos de uso (cómo se llevan a cabo).

2. Centrado en la arquitectura: La arquitectura muestra la visión común del sistema completo en la que el equipo de proyecto y los usuarios deben estar de acuerdo, por lo que describe los elementos del modelo que son más importantes para su construcción, los cimientos del sistema que son necesarios como base para comprenderlo, desarrollarlo y producirlo económicamente. El modelo de arquitectura se representa a

través de vistas en las que se incluyen los diagramas de UML (Vista de Casos de Uso, Vista Lógica, Vista de Procesos, Vista de Implementación, Vista de Despliegue).

3. Iterativo e Incremental: RUP propone que cada fase se desarrolle en iteraciones. Una iteración involucra actividades de todos los flujos de trabajo, aunque desarrolla fundamentalmente algunos más que otros. Por ejemplo, una iteración de elaboración centra su atención en el análisis y diseño, aunque refina los requerimientos y obtiene un producto con un determinado nivel, pero que irá creciendo incrementalmente en cada iteración. [16]

1.2.7 Lenguaje Unificado de Modelado (UML)

El Lenguaje Unificado de Modelado (UML) es un lenguaje de modelado visual que se usa para especificar, visualizar, construir y documentar artefactos de un sistema de software. Captura decisiones y conocimiento sobre los sistemas que se deben construir. Se usa para entender, diseñar, hojear, configurar, mantener, y controlar la información sobre tales sistemas. Está pensado para usarse con todos los métodos de desarrollo, etapas del ciclo de vida, dominios de aplicación y medios. El lenguaje de modelado pretende unificar la experiencia pasada sobre técnicas de modelado e incorporar las mejores prácticas actuales en un acercamiento estándar. [17]

Algunas de las propiedades de UML como lenguaje de modelado estándar son:

- Concurrencia, es un lenguaje distribuido y adecuado a las necesidades de conectividad actuales y futuras.
- Modela estructuras complejas.
- Las estructuras más importantes que soportan tienen su fundamento en las tecnologías orientadas a objetos, tales como objetos, clase, componentes y nodos.
- Emplea operaciones abstractas como guía para variaciones futuras, añadiendo variables si es necesario.
- Comportamiento del sistema: casos de uso, diagramas de secuencia y de colaboraciones, que sirven para evaluar el estado de las máquinas. [17]

Ventajas de UML:

- Diseño y documentación.
- Descubrimiento de fallas.
- Ahorro de tiempo en el desarrollo del software.
- Mucho más fáciles las modificaciones.
- Fácil comunicación entre los programadores.

1.2.8 Herramientas CASE de Modelado con UML

A medida que los sistemas que hoy se construyen se tornan más y más complejos, las herramientas de modelado con UML ofrecen muchos beneficios para todos los involucrados en un proyecto, por ejemplo, administrador del proyecto, analistas, arquitectos, desarrolladores y otros. Las herramientas CASE (Computer Aided Software Engineering, Ingeniería de Software Asistida por Ordenador) de modelado con UML nos permiten aplicar la metodología de análisis y diseño orientados a objetos y abstraernos del código fuente, en un nivel donde la arquitectura y el diseño se tornan más obvios y más fáciles de entender y modificar. Cuanto más grande es un proyecto, es más importante utilizar una herramienta CASE. Por otro lado, al usar las herramientas CASE: [18]

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

El modelo actúa como el plano y guiará finalmente la construcción del sistema. De manera semejante, la administración es capaz de ver, puntualmente y desde un alto nivel, una representación del diseño y comprender lo que está sucediendo. Por estas razones, las herramientas CASE de UML acompañadas con metodologías, brindan una forma de representar sistemas demasiados complejos para comprenderlos a través de su código fuente subyacente y nos permiten desarrollar la solución de software correcta, más

rápido y más económicamente. Sin embargo, las herramientas CASE varían con respecto a las capacidades de modelado con UML, el soporte del ciclo de vida del proyecto, las ingenierías directa y reversa, el modelado de datos, el precio, el soporte, la facilidad de uso, etc. [18]

Visual Paradigm

Visual Paradigm es una herramienta CASE que utiliza UML como lenguaje de modelado. Entre sus principales características se encuentran que realiza un diseño centrado en casos de uso, hace posible el uso de un lenguaje estándar común a todo el equipo de desarrollo lo que facilita la comunicación entre los mismos, además tiene capacidad de realizar una ingeniería directa (versión profesional) así como inversa, el modelo y el código permanecen sincronizados en todo el ciclo de desarrollo y además corre en múltiples plataformas.

1.2.9 Notación Polaca Inversa

Tratamiento de expresiones:

Existen tres tipos de notación para expresiones matemáticas (atendiendo a la posición en que se sitúan los operadores respecto de sus operandos):

- ✓ Notación Infija: Se trata de la notación que utilizamos habitualmente. El operador se sitúa entre sus operandos (en el caso de los operadores binarios). En el caso de operadores unarios (como el coseno) cómo solo hay un operador se usa la misma forma que en la notación prefija.

Ejemplo: $(a + b) * c ^ d$

Notación prefija: En esta notación el operador precede a los operandos. El orden de operaciones queda claramente especificado por la posición de los elementos de la expresión (operadores y operandos). Se puede expresar como una lista de elementos o separando los operandos de cada operación con comas y entre paréntesis.

Ejemplo: Prefija (lista) $* + a b ^ c d$

Prefija $* (+ (a, b), ^ (c, d))$

- ✓ Notación postfija: Exactamente igual que en el caso de la notación prefija, aunque en este caso el operador se sitúa detrás de los operandos.

Ejemplo: $a b + c d ^ *$

Una de las desventajas que presenta la evaluación en notación prefija es: que se requiere conocer el número de operandos de cada operación, y se usa una o dos pilas de ejecución. Para eliminar esta dificultad se utilizará la postfija, en la cual los datos se evalúan directamente cuando se introducen y los mismos se manejan dentro de una estructura LIFO (Last In First Out), lo que optimiza los procesos a la hora de programar.

Principales ventajas de la notación postfija (polaca inversa):

- ✓ Los cálculos se realizan secuencialmente según se van introduciendo los operadores, en vez de tener que esperar a escribir la expresión al completo. Debido a esto, se cometen menos errores al procesar cálculos complejos.
- ✓ No requiere paréntesis ni reglas de preferencia, al contrario que la notación algebraica, ya que el proceso de apilamiento permite calcular la expresión por etapas.
- ✓ Un resultado intermedio permite al usuario comprobar los resultados y corregir los errores con mayor facilidad. Resulta más sencillo seguir el proceso del cálculo. El usuario define la prioridad de los operadores.
- ✓ La notación RPN es lógica, puesto que el usuario primero proporciona un número y luego indica qué debe hacerse con él.

1.2.9.1 Convención de una cadena infija en una polaca postfija

Para convertir una expresión infija en una postfija es necesario primero establecer la jerarquía de los operadores de tal manera que a cada operador se le asigna una prioridad la cual va a estar determinada según la prioridad de ejecución que tenga sobre los otros operadores quedando según el ejemplo siguiente:

<u>Operadores</u>	<u>Po</u>
Si entonces, SiSoloSi	1
Or, XOr	2
And	3
Lógicos Unarios	4
Asignación, Comparación	5
+, -	6
*, /	7
^	8
Unarios	9

Donde \rightarrow Po va a ser la “Prioridad absoluta de cada operador”.

Los operadores que tienen prioridad del 1 al 4 actúan sobre operandos lógicos y el resultado es un operando lógico.

Los operadores de comparación actúan sobre operandos numéricos y el resultado es un operando lógico. El operador de asignación puede actuar sobre operandos numéricos u operandos lógicos y el resultado es un operando lógico.

Los restantes actúan sobre operandos numéricos y el resultado es un operando numérico.

La forma de conversión consiste en crear dos cadenas, una de ellas de operadores con su jerarquía y la otra es la cadena postfija, aplicándose las siguientes reglas:

- Los operandos de la cadena infija original se vacían en la cadena postfija a medida que se leen.
- Los paréntesis, si son abiertos, incrementa el número de paréntesis abiertos en 1 (NPA+1); si son cerrados disminuyen el número de paréntesis abiertos en 1 (NPA-1).
- Si se lee un operador se le asigna una prioridad “P” (Prioridad relativa) dada por: $P = Po + NPA * (\text{Cantidad de operadores})$ y se almacena en la cadena de operadores. En caso de encontrarse un

operador con menor prioridad que del que se encuentre en la cadena de operadores, entonces el que tiene mayor prioridad se pasa a la cadena postfija.

- Cuando se lee el último elemento se agrega la cadena de operadores a la cadena postfija, empezando por el de mayor prioridad y terminando por el de menor.

Un ejemplo de la convención de cadena infija a postfija (Ver Anexo 3).

Todos los operadores tienen una acción definida lo que implica que la expresión tiene una sintaxis correcta, por otro lado: “una cadena polaca sintácticamente correcta comienza siempre con un operando y termina con un operador”.

La regla para la determinación del dominio de acción de un operador es simple, y consiste en:

- ✓ Para un operador unario: a partir de su posición determinar hacia la izquierda la primera cadena operando que exista.
- ✓ Para un operador binario: a partir de su posición a la izquierda determinar las dos cadenas operandos. [19]

1.2.9.2 Evaluación numérica de cadenas polacas mediante el uso de Pilas:

Una de las manipulaciones más frecuentes de las cadenas polacas es el cálculo numérico de las expresiones matemáticas que ellas representan, este problema puede parecer trivial, ya que la propia representación postfija de las cadenas, las hacen fáciles de evaluar, en efecto, basta con ir depositando el valor de los operandos en una lista, y cada vez que aparezca un operador, realizar la operación con los dos últimos valores de la lista si el operador es binario, o con el último si es unario, depositando a su vez el valor obtenido en dicha lista. (Ver Anexo 4)

1.2.9.3 Evaluación numérica de cadenas polacas mediante el uso de Árbol:

Traducción de notación postfija a un árbol binario:

Después de hacer la conversión a notación postfija de una expresión matemática, se pasará a representar la misma en un árbol binario, para realizar el mismo se tendrán en cuenta los siguientes pasos:

Antes de empezar vale aclarar que se utilizará una Pila la cual va a contener los operandos (números, variables y expresiones (subárboles)) que se vayan encontrando en la cadena postfija.

- Los operandos de la cadena postfija son adicionados directamente en la Pila a medida que son leídos.
- Cuando es encontrado un operador, se crea un nuevo árbol donde el operador pasa a ser la raíz del mismo, luego si el operador es binario se extraen dos elementos de la Pila el primero pasa a ser el hijo derecho del nuevo árbol y el segundo pasa a ser el hijo izquierdo, en caso de ser un operador unario se extrae un elemento de la Pila y este pasa a ser su hijo derecho, después de creado el nuevo árbol con sus nuevos hijos este es adicionado a la Pila.
- El resultado, es el árbol binario resultante de extraer el único término existente en la pila. (Ver Anexo 5)

Para realizar la evaluación de la cadena polaca representada en el árbol binario basta con recorrer el mismo en Entre-Orden el cual se realiza visitando primero el hijo izquierdo en Entre-Orden, luego la raíz y finalmente se visita el hijo derecho en Entre-Orden.

1.3 Conclusiones

Con la realización de este capítulo se definieron las principales características y ventajas de las herramientas predefinidas por el grupo de arquitectura del centro “Unidad de compatibilización, integración y desarrollo del software para la defensa”. Logrando con la realización del mismo una mayor comprensión del porque de su selección por parte de dicho grupo.

Gestor de base datos: PostgreSQL 8.0

Servidor Web: Apache 2.0 o superior.

Lenguaje de programación del lado del servidor: PHP 5.0

Lenguaje de programación del lado del cliente: JavaScript, técnicas de AJAX

Para el desarrollo de la documentación: Metodología RUP con notación UML

Herramienta CASE de Modelado UML: Visual Paradigm v2.3

Navegador: Mozilla Firefox.

Para el tratamiento de expresiones matemáticas se utilizará la notación postfija (polaca inversa) y para la evaluación de la misma se utilizará el TDA (Tipo de dato abstracto) Pila.

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

2.1 Introducción.

El presente capítulo tiene como propósito hacer un análisis de cómo es que funciona el proceso de planificación de la demanda para la lucha armada en las Fuerzas Armadas, ver quienes realizan la misma y de que manera le dan cumplimiento a lo establecido en la “Metodología para el proceso de formulación, presentación y satisfacción de las demandas para la lucha armada”, comprender la estructura y dinámica de la organización, los problemas actuales dentro de esta e identificar las mejoras potenciales.

2.2 Objeto de estudio.

El proceso de planificación de la demanda consiste en el conjunto de medidas y actividades que realizan desde el MINFAR hasta las Unidades Militares para dar cumplimiento a la formulación de la demanda para la lucha armada. El MINFAR es el encargado de determinar los niveles de actividad y el período que se requiere para la formulación, elabora las propuestas de nomenclaturas a solicitar, determina los volúmenes, sus normas y controla dicho proceso y elabora un cronograma de trabajo por donde sus niveles inferiores (Ejército, Región Militar y Unidades Demandantes) se guiarán para llevar a cabo el proceso de planificación de la demanda.

Las demandas para la lucha armada son formuladas por:

- ✓ Tropas Regulares de las FAR.
- ✓ Las MTT que garantizan la defensa territorial y local.
- ✓ Los órganos del MININT.

La formulación de la demanda es registrada en un informe, el cual contendrá toda la planificación realizada por una Unidad Demandante y antes de ser enviada a su Región Militar correspondiente deberá ser revisada por una persona calificada para el mismo, donde verificará el correcto manejo de las nomenclaturas y normas establecidas por el MINFAR además de que los datos presentados sean lo más correctos posibles.

2.2.1 Problema y situación problemática.

La planificación de la demanda para la lucha armada actualmente se realiza manualmente, el logístico debe especificar uno por uno los productos o servicios que va a demandar según la especialidad y operación, por lo que hace que este proceso se haga lento y tedioso a la hora de realizarlo y de existir cualquier error en los datos atrasaría todo el proceso. Todo esto trae como consecuencia que se haga necesario la informatización del mismo, para así poderse asegurar la calidad y seguridad del mismo.

2.2.1.1 Objetivo estratégico de la organización.

El objetivo estratégico que persigue la organización es garantizar desde tiempo de paz el cumplimiento de los planes de producción y servicios establecidos para las situaciones excepcionales. Para dar cumplimiento al mismo, es necesaria la informatización del proceso de planificación de la demanda para la lucha armada mediante un sistema basado en tecnología Web para así lograr:

- ✓ Agilizar el proceso de formulación de la demanda para la lucha armada.
- ✓ Asegurar la calidad y seguridad de la información que se manipulan en dicho proceso.
- ✓ Permitir que todas las Unidades encargadas de la planificación no importe su ubicación tenga el acceso a dicho sistema.

2.2.1.2 Flujo actual del proceso involucrado en el campo de acción.

El proceso de la planificación de la demanda para la lucha armada se inicia cuando el MINFAR determina los principales niveles de actividades a desarrollar, las normas, bases de cálculo a utilizar y elabora un cronograma de trabajo con el objetivo de mantener un control sobre la realización del mismo, en el que se especifica cuando las Unidades Militares deben comenzar con la planificación de la demanda y cuando deben entregarlas a sus correspondientes Regiones Militares. Sobre la base de las indicaciones emitidas por el MINFAR, del cronograma de trabajo y de los documentos normativos relacionados con esta actividad los Ejércitos organizan y planifican el trabajo de formulación, presentación y satisfacción de las demandas en su territorio y se lo hacen llegar a sus niveles inferiores (provincia-municipio-unidades).

A nivel de Región Militar se le informa a cada una de sus Unidades Militares dicho cronograma y documentos, y sobre las indicaciones recibidas es que las Unidades Militares (tropas regulares de las FAR, MTT y el MININT) puntualizan (calculan) las demandas a presentar para el año entrante y se las presentan a su Región Militar en un informe de planificación.

2.2.2 Objeto de automatización.

En el proceso de Planificar Demanda se desea automatizar:

- Calcular la demanda para la lucha armada.
- Realizar un informe con el cálculo de la demanda realizada.

2.2.3 Documentación que se maneja.

Cronograma de trabajo: Es un informe creado por el MINFAR en el cual se registran las fechas a tener en cuenta por las Unidades Militares para realizar la planificación, los principales niveles de actividades a desarrollar, las normas, bases de cálculo y las nomenclaturas a solicitar.

Informe de planificación: Es un informe que contiene los productos y/o servicios a demandar para la lucha armada del año entrante.

2.2.4 Propuesta de sistema.

Para llevar a cabo el proceso de planificación de la demanda para la lucha armada el sistema deberá validar que el usuario que desee trabajar en ella tiene el permiso para hacerlo, asegurando así la seguridad de la información que se maneja en la misma, además, mostrar las variables que se tendrán en cuenta en la planificación de la demanda para la lucha armada. Para realizar la planificación, el sistema deberá mostrar todos los municipios, las especialidades y las operaciones, según la selección del usuario el sistema visualizará todas las variables definidas para ese tipo de usuario, especialidad, operación y municipio, de ocurrir cualquier error en el cálculo de los productos y/o servicios, se podrá eliminar el mismo, como también verificarlo cuando es realizado; se podrá eliminar toda la demanda realizada hasta ese momento o que se encuentre almacenada en la base de datos, al igual que, revisar el informe final de la planificación o guardarlo en la máquina del usuario para su posterior envío a su Región Militar correspondiente.

2.2.5 Modelación del negocio.

El modelo del negocio describe el negocio en términos de casos de usos del negocio, que corresponde a lo que generalmente se le llama procesos.

Un proceso del negocio es el conjunto estructurado de las actividades que han sido diseñadas para producir un resultado específico para un cliente o el mercado.

Los objetivos del modelado del negocio son:

- ✓ Comprender la estructura y la dinámica de la organización en la cual se va a implantar un sistema.
- ✓ Comprender los problemas actuales de la organización e identificar las mejoras potenciales.
- ✓ Asegurar que los consumidores, usuarios finales y desarrolladores tengan un entendimiento común de la organización.
- ✓ Derivar los requerimientos del sistema que va a soportar la organización. [20]

2.2.5.1 Actores del negocio.

Actor del negocio	Descripción
Órgano de la Logística del MINFAR	Es el encargado de informar que ya se puede realizar la planificación de la demanda para la lucha armada.

2.2.5.2 Trabajadores del negocio.

Trabajador del negocio	Descripción
Ejército	Es el encargado de organizar y planificar el trabajo de formulación y presentación de las demandas en su territorio.
Región Militar	Es el encargado de informar a sus Unidades Militares que ya pueden comenzar el proceso de planificación.
Unidad Demandante (Unidad Militar, MININT, MTT)	Es el encargado de formular (calcular) la demanda.

2.2.5.3 Diagrama de Casos de Usos del negocio.

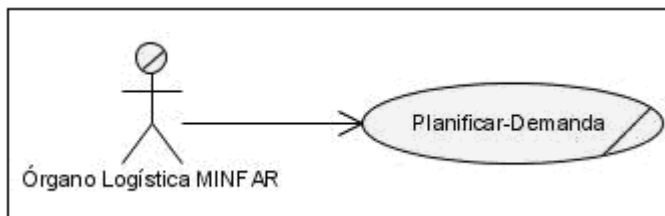


Figura 2.1- Diagrama de Caso de Uso del negocio.

2.2.5.4 Descripción del Caso de Uso del negocio.

Caso de Uso:	Planificar Demanda	
Actores:	Órgano de la Logística del MINFAR	
Trabajadores:	Ejército, Región Militar, Unidad Demandante	
Resumen:	El caso de uso se inicia cuando el Órgano de Logística del MINFAR indica a los niveles inferiores que ya pueden comenzar el proceso de formulación de la demanda para la lucha armada y este termina cuando la Unidad Demandante le envía el modelo de la planificación de la demanda a su Región Militar correspondiente.	
Precondiciones:	.	
Flujo Normal de Eventos		
Acción del Actor	Respuesta del Negocio	
1. El Órgano de la Logística del MINFAR elabora un cronograma de trabajo, a tener en cuenta para la formulación de las demandas para la lucha armada y se lo informa al Ejército.	2. El Ejército planifica el trabajo de formulación de las demandas en su territorio y se lo hace llegar a la Región Militar.	
	3. La Región Militar hace llegar esa información a las Unidades Demandantes pertenecientes en sus municipios.	
	4. Las Unidades Demandantes calculan las demandas a	

	presentar para el año entrante, y realizan un informe de planificación con esos datos. Después de revisado y aprobado el informe de planificación se lo presentan a su Región Militar, terminando el caso de uso.
Poscondiciones	<ul style="list-style-type: none"> Se envía un informe con la planificación de la demanda a la Región Militar.
Mejoras	
Prioridad	Alta.

2.2.5.5 Diagrama de actividades.

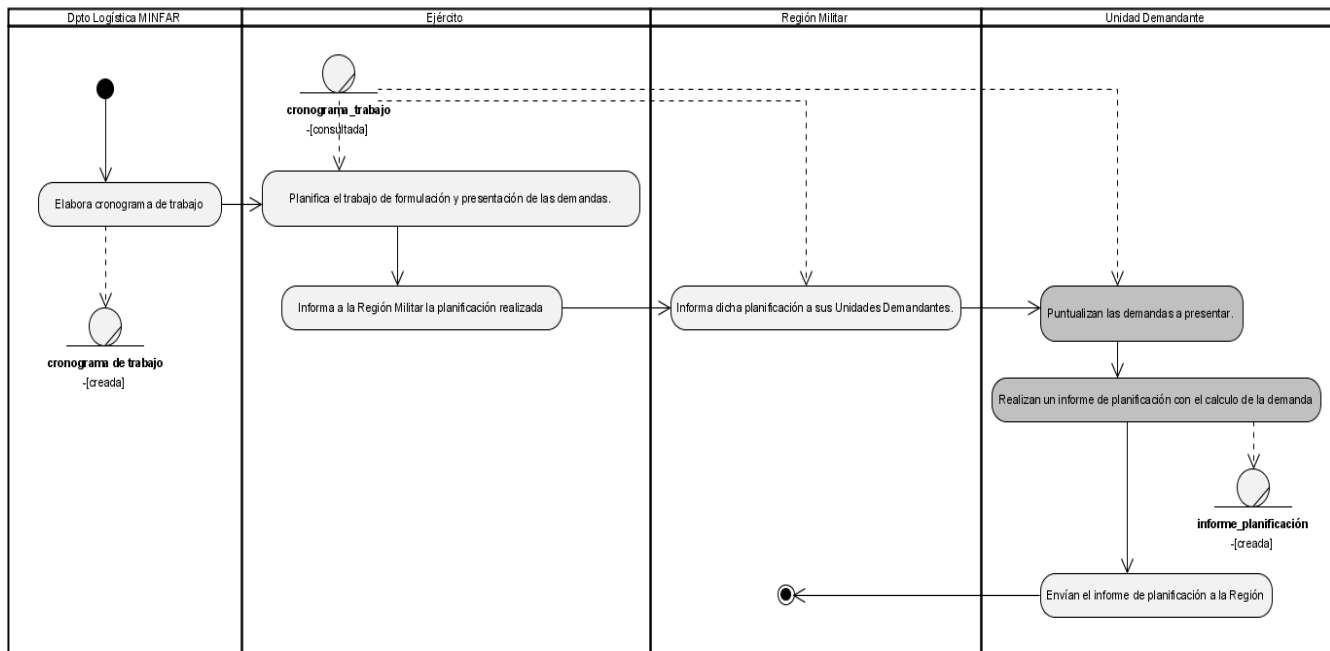


Figura 2.2- Diagrama de Actividades del Caso de Uso del negocio Planificar Demanda.

2.2.5.6 Modelo de objetos.

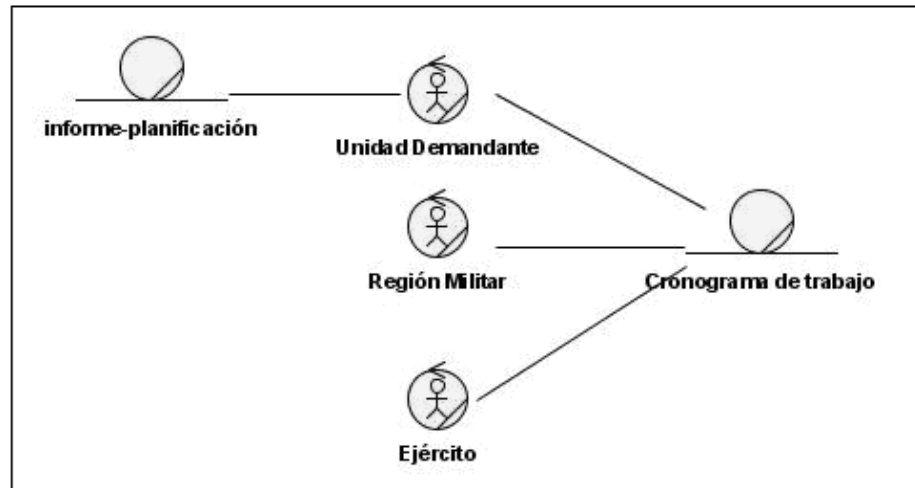


Figura 2.3- Modelo de Objetos del Caso de Uso del negocio Planificar Demanda.

2.2.6 Especificación de los requisitos de software.

¿Qué es un requerimiento?

Condición o capacidad que tiene que ser alcanzada o poseída por un sistema o componente de un sistema para satisfacer un contrato, estándar, u otro documento impuesto formalmente.

Los requisitos se pueden clasificar en:

- Funcionales
- No funcionales.

2.2.6.1 Definición de los requerimientos funcionales.

Son capacidades o condiciones que el sistema debe cumplir, los cuales son:

- R1 Calcular Demanda.
 - R1.1 Seleccionar municipio por donde se va a realizar la demanda.
 - R1.2 Seleccionar las especialidades a demandar.
 - R1.3. Seleccionar en que operación se va a realizar la demanda.
 - R1.4 Buscar las variables a calcular según el municipio, especialidad, usuario y operación.

- R1.5 Calcular los productos a demandar.
- R1.6 Mostrar los productos calculados.
- R1.7 Eliminar los productos calculados.
- R2 Mostrar las variables a planificar según el tipo de usuario.
- R3 Reiniciar la Demanda.
- R4 Autenticar Usuario
- R5 Modificar Contraseña
- R6 Trabajar con el informe de planificación.
 - R6.1 Mostrar informe de planificación.
 - R6.2 Exportar informe de planificación.

2.2.6.2 Definición de los requerimientos no funcionales.

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

- Apariencia o interfaz externa:
 - El sistema debe tener una interfaz fácil de usar y amigable para que pueda ser utilizada sin mucha preparación por el usuario.
 - Empleo de imágenes y colores identificados con el negocio donde se implantará el sistema.
 - Estará diseñado para resolución de 800x600, aunque deberá verse en cualquier resolución superior a esta.
- Usabilidad:
 - El sistema podrá ser usado por cualquier persona que posea conocimientos básicos en el manejo de la computadora.
- Rendimiento:
 - Los tiempos de respuesta y velocidad de procesamiento de la información serán rápidos, no mayores de 5 segundos para las actualizaciones y reportes y 10 minutos para la importación y exportación de ficheros de información a los distintos niveles.

- Soporte:
 - La aplicación contará antes de su puesta en marcha, con un período de pruebas, se le dará mantenimiento, configuración y se brindará el servicio de instalación.
- Portabilidad:
 - El sistema debe ser multiplataforma, haciendo énfasis en Linux y Windows.
- Seguridad:
 - Autenticación (Contraseña de acceso).
 - Autorización (Atribución a los usuarios respecto a sus funciones de trabajo).
 - La atención al sistema incluyendo, el mantenimiento de las bases de datos así como la salva de la información se realizarán de forma centralizada por el administrador.
- Políticos culturales:
 - El sistema solo podrá ser utilizado en territorio cubano y por las entidades autorizadas por el Ministerio de las FAR.
 - El producto no debe contener palabras en otros idiomas.
 - El producto debe respetar los términos empleados normalmente por los especialistas en el tema de la esfera que se automatiza.
- Software:
 - *Para el cliente:*
 - Navegador Mozilla Firefox.
 - Sistema operativo Windows 98 o superior o Linux.
 - *Para el servidor:*
 - Sistema operativo Windows Advancer Server (2000 o superior) o Linux en cualquiera de sus distribuciones.
 - Un servidor Apache 2.0 o superior con módulo PHP 5.0 disponible, este debe estar configurado con la extensión "pgsql" incluida.
 - Un servidor de base de datos PostgreSQL 8.0 o superior.
- Hardware:
 - *Para el cliente:*
 - Requerimientos mínimos: Procesador Pentium II a 133Mhz con 128 Mb de memoria RAM.
 - Tarjeta de red.

- *Para el servidor:*
 - Requerimientos mínimos: Procesador Pentium III a 1GHz de velocidad de procesamiento y 1Gb de memoria RAM.
 - Al menos 40Gb de espacio libre en disco duro.
 - Tarjeta de red.
- Restricciones para el diseño e implementación:
 - Utilizar los estándares establecidos (codificación, diseño, entre otros).
 - Emplear como servidores Web y de bases de datos Apache y PostgreSQL respectivamente.
 - Utilizar como lenguaje del lado del servidor al PHP 5.0 o superior y del lado del cliente el JavaScript.
- Ayuda y documentación en línea.
 - El sistema constará de una Ayuda que mostrará como está conformado la página con la cual se está trabajando, además de una opción de Misión y Objetivos que explica brevemente en que consiste la planificación.

2.2.7 Definición de los casos de uso.

2.2.7.1 Definición de los actores.

Actor	Descripción
Logístico por especialidades de la Unidad Demandante (tropas regulares de la FAR, MTT y el MININT).	Es el encargado de calcular la demanda según las especialidades, las operaciones y el municipio por el que va a planificar.

2.2.7.2 Listado de casos de usos.

CU - 1	Registrar Usuario
Actor	Unidad Demandante
Descripción	El Caso de Uso se inicia cuando el logístico de la Unidad Demandante se registra en la aplicación para realizar la planificación de la demanda para la lucha armada y este concluye cuando al registrarse correctamente este tiene acceso a la misma.
Referencia	R4

CU – 2	Obtener Variables
Actor	Unidad Demandante
Descripción	El Caso de Uso (CU) se inicia cuando el logístico selecciona la opción “Obtener Variables” el cual mostrará una ventana del navegador preguntándole al usuario que desea realizar con la misma: abrir la página, guardarla o cancelarlo todo, concluyendo así el mismo.
Referencia	R2

CU – 3	Planificar Demanda
Actor	Unidad Demandante
Descripción	El CU se inicia cuando el logístico selecciona la opción “Planificar Demanda” el cual mostrará una interfaz donde el logístico debe seleccionar porque municipio, especialidad y operación va a realizar la planificación, y después de calcular los productos planificados, en caso de ocurrir cualquier error al entrar los datos puede eliminar los mismos y después verificar si están correctos o no. El CU termina cuando se hallan calculados los productos a presentar para la demanda.
Referencia	R1

CU – 4	Reiniciar Demanda
Actor	Unidad Demandante
Descripción	El Caso de Uso se inicia cuando el logístico selecciona la opción “Reiniciar Demanda”, eliminando toda la planificación realizada, concluyendo así el CU.
Referencia	R3

CU – 5	Revisar informe _ planificación
Actor	Unidad Demandante
Descripción	El Caso de Uso se inicia cuando el logístico selecciona la opción “MD1/Revisar” mostrándose la demanda realizada, concluyendo así el CU.

Referencia	R6
-------------------	----

CU – 6	Exportar informe _ planificación
Actor	Unidad Demandante
Descripción	El Caso de Uso se inicia cuando el logístico selecciona la opción “MD1/Exportar”, el cual mostrará una ventana del navegador preguntándole al usuario que desea realizar con la misma: abrir la página, guardarla o cancelarlo todo, concluyendo así el CU.
Referencia	R6

CU – 7	Modificar Contraseña
Actor	Unidad Demandante
Descripción	El Caso de Uso se inicia cuando el logístico selecciona la opción “Cambiar Contraseña”, el cual mostrará una interfaz pidiéndole al usuario que introduzca su nueva contraseña y la repetición de la misma, concluyendo así el CU.
Referencia	R5

2.2.7.3 Diagrama de Casos de Uso.

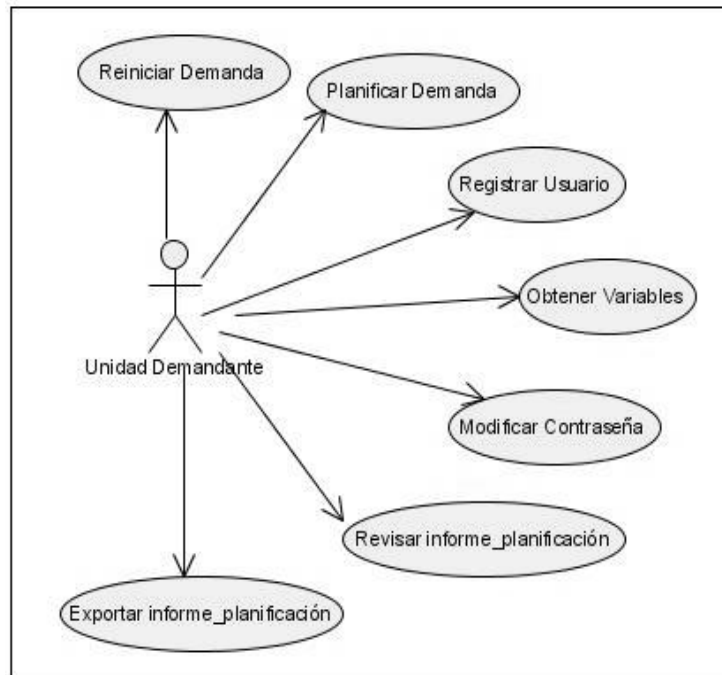


Figura 2.4- Diagrama de Caso de Uso de Planificar Demanda

2.2.7.4 Caso de usos expandidos

Nombre del Caso de Uso:	Registrar Usuario
Actores:	Logístico por especialidades de la Unidad Demandante
Propósito:	No permitir que la información sea accedida por un personal no autorizado para el mismo.
Resumen:	El CU se inicia cuando el logístico se registra en la aplicación, el mismo finaliza cuando el logístico es registrado correctamente y tiene acceso a la aplicación.
Responsabilidades:	R4
CU asociados:	-
Precondiciones:	
Interfaz I	



- (1) Usuario a registrarse para la aplicación.
- (2) Contraseña para validar que el usuario puede acceder a la aplicación.
- (3) Botón “Conectar”.

Flujo normal de los eventos

Acción del actor	Respuesta del Sistema
1. El logístico entra los datos (usuario (1), contraseña (2)) y selecciona la opción “Conectar” (3).	2. El sistema, si los datos introducidos son correctos le permite acceder a la aplicación.

Cursos Alternos

Línea 2:

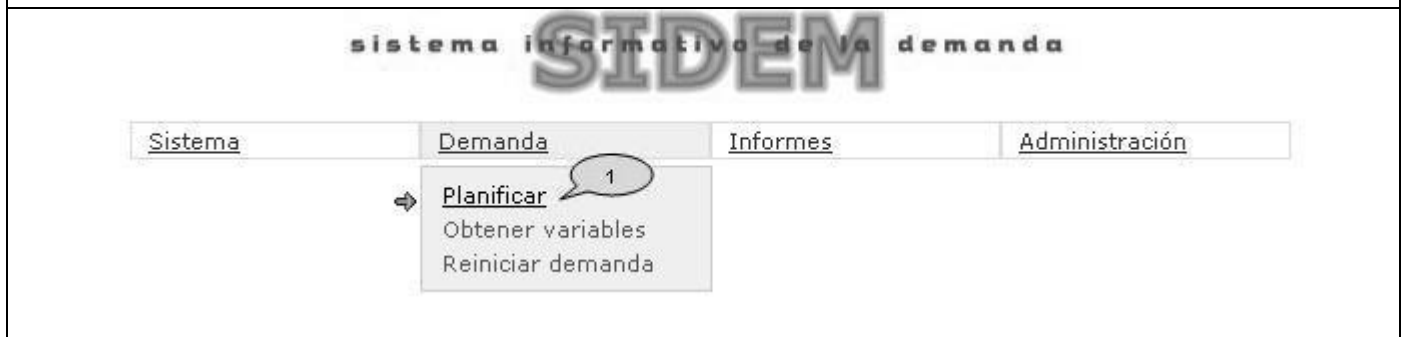
Si los datos introducidos no son correctos el sistema muestra nuevamente la interfaz de registrarse (Interfaz I).

Poscondiciones: Si el logístico entra los datos correctos, entra a la aplicación.

Nombre del Caso de Uso:	Planificar Demanda
Actores:	Logístico por especialidades de la Unidad Demandante
Propósito:	Calcular la demanda para la lucha armada, a presentar para el año entrante.
Resumen:	El caso de uso se inicia cuando el logístico selecciona en la aplicación la opción de planificar la demanda. El caso de uso finaliza cuando el usuario calcula los productos de la demanda para la lucha armada.

Responsabilidades:	R1, R1.1, R1.2, R1.3, R1.4, R1.5, R1.6, R1.7
CU asociados:	-
Precondiciones:	El logístico se haya autenticado en la aplicación.

Interfaz I

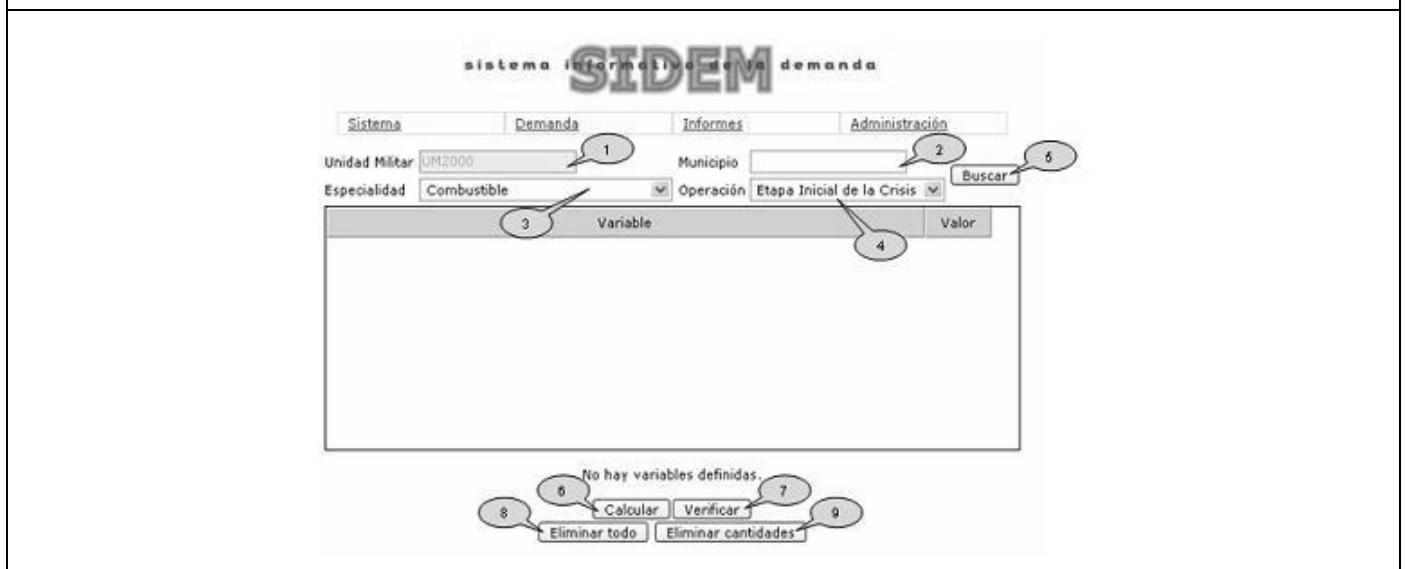


(1) Opción del Menú Principal “Planificar”.

Flujo normal de los eventos

Acción del actor	Respuesta del Sistema
	1. El sistema muestra la interfaz I
2. El logístico selecciona la opción “Planificar” (1).	
	3. El sistema muestra la interfaz II.

Interfaz II



(1) Número de la Unidad Militar (usuario).

(2) Árbol jerárquico con todos los municipios el mismo está conformado por:

- Región
 - Provincia
 - Municipio

(3) Listado de las especialidades.

(4) Listado de las operaciones.

(5) Botón que busca cuales son las variables a planificar según la especialidad, Unidad Militar, Municipio y Operación.

(6) Botón que calcula los productos de la de demanda para la lucha armada.

(7) Botón que muestra los productos calculados dado la especialidad, operación, municipio y usuario seleccionados.

(8) Botón que elimina los productos calculados y los datos se inicializan en cero.

(9) Botón que elimina los productos calculados y los datos permanece con los valores entrados anteriores.

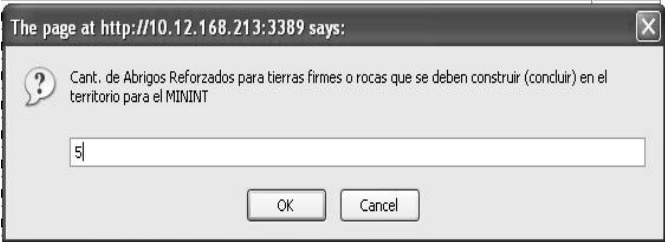

4. El logístico marca en la casilla del municipio (2).

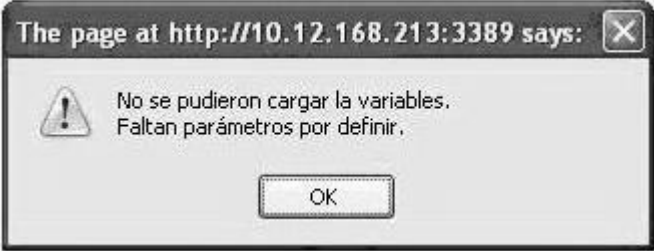
5. El sistema muestra un árbol jerárquico con todos los municipios.

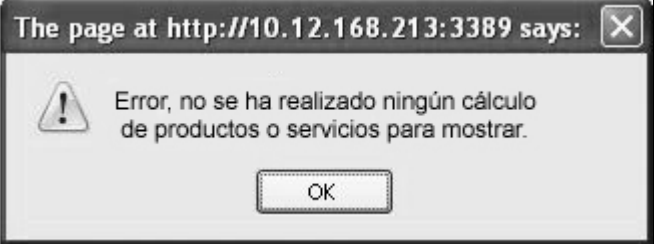
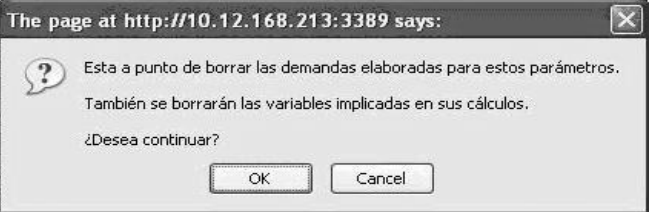
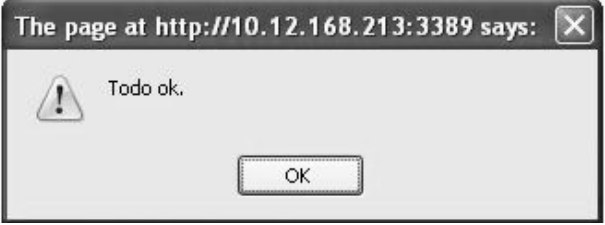


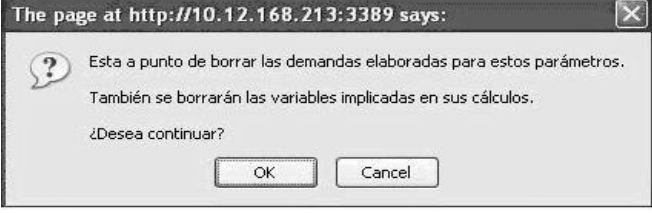
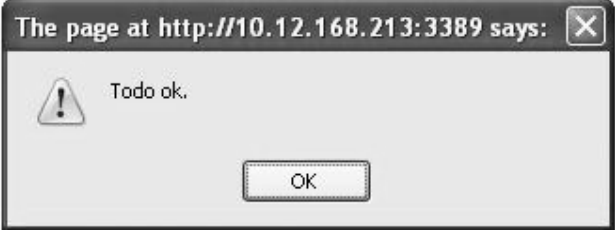
6. El logístico selecciona por que municipio va a planificar y después selecciona por que operación (4) y en que especialidad (3) calculará los productos. Después de

7. El sistema muestra todas las variables definidas para ese tipo de usuario, especialidad, operación y municipio.

<p>definidos estos datos el logístico selecciona la opción “Buscar” (5).</p>	
<p>8. El logístico selecciona con doble clic la casilla donde va a introducir los datos pertenecientes a cada variable.</p>	<p>9. El sistema muestra una ventana con la opción de asignar dichas variables.</p> 
<p>10. El logístico introduce los datos de cada producto a calcular y da OK.</p>	
<p>11. El logístico selecciona la opción de “Calcular” (6).</p>	<p>12. El sistema calcula la demanda, y muestra la cantidad de productos que fueron calculados e introducidos en el sistema y los que tuvieron error.</p> 
<p>13. El logístico después de haber calculado los productos de la demanda para lucha armada, él puede seleccionar las siguientes opciones:</p> <ul style="list-style-type: none"> ✓ Verificar Productos (ver sección “Verificar Productos”) ✓ Eliminar Cantidades (ver sección “Eliminar 	

<p>Cantidades”) ✓ Eliminar todo (ver sección “Eliminar todo”)</p>																										
Flujo alterno																										
Acción del actor	Respuesta del Sistema																									
<p>4.1. Si el logístico no selecciona porque municipio va a planificar.</p>	<p>4.2. El sistema muestra un mensaje de error.</p> 																									
Sección “Verificar Productos”																										
Interfaz III																										
<p style="text-align: center;">Esta es una muestra de los productos calculados. En ella se puede apreciar la generalización con que cumplen determinados productos.</p> <table border="1" data-bbox="285 1094 1406 1276"> <thead> <tr> <th>Producto</th> <th>Cantidad</th> <th>U.M.</th> <th>Importe</th> <th>Nom.</th> </tr> </thead> <tbody> <tr> <td>Elementos Prefab. de HA del tipo C-67 p/l Ptos. de</td> <td>12.00</td> <td>Uno</td> <td>1 292.40</td> <td>C</td> </tr> <tr> <td>Elementos Prefab. de HA del tipo C-67 p/l Zanjas C</td> <td>5.00</td> <td>Uno</td> <td>538.50</td> <td>T</td> </tr> <tr> <td>Elementos Prefab. de HA del tipo C-67 p/l Zanjas C</td> <td>1.00</td> <td>Uno</td> <td>107.70</td> <td>C</td> </tr> <tr> <td>Elementos Prefab. de HA del tipo C-67-T p/l Ptos.</td> <td>6.00</td> <td>Uno</td> <td>240.48</td> <td>C</td> </tr> </tbody> </table>		Producto	Cantidad	U.M.	Importe	Nom.	Elementos Prefab. de HA del tipo C-67 p/l Ptos. de	12.00	Uno	1 292.40	C	Elementos Prefab. de HA del tipo C-67 p/l Zanjas C	5.00	Uno	538.50	T	Elementos Prefab. de HA del tipo C-67 p/l Zanjas C	1.00	Uno	107.70	C	Elementos Prefab. de HA del tipo C-67-T p/l Ptos.	6.00	Uno	240.48	C
Producto	Cantidad	U.M.	Importe	Nom.																						
Elementos Prefab. de HA del tipo C-67 p/l Ptos. de	12.00	Uno	1 292.40	C																						
Elementos Prefab. de HA del tipo C-67 p/l Zanjas C	5.00	Uno	538.50	T																						
Elementos Prefab. de HA del tipo C-67 p/l Zanjas C	1.00	Uno	107.70	C																						
Elementos Prefab. de HA del tipo C-67-T p/l Ptos.	6.00	Uno	240.48	C																						
Acción del actor	Respuesta del Sistema																									
<p>1. El logístico selecciona la opción “Verificar” (7).</p>	<p>2. El sistema muestra la interfaz III con todos los productos calculados (este resultado se da por especialidad, operación, usuario y municipio).</p>																									
Flujo alterno Sección “Verificar Productos”																										
Acción del actor	Respuesta del Sistema																									
<p>1.1. Si el logístico selecciona la opción “Verificar” sin haber calculado los productos previamente.</p>	<p>1.2. El sistema muestra un mensaje de error.</p>																									

	
Sección “Eliminar Cantidades”	
Acción del actor	Respuesta del Sistema
<p>1. El logístico selecciona la opción “Eliminar cantidades” (9).</p>	<p>2. El sistema muestra una ventana preguntando al usuario si está seguro de eliminar las demandas elaboradas.</p> 
<p>3. El logístico selecciona el botón “OK”.</p>	<p>4. El sistema elimina la planificación que se hizo con esas variables, pero da la oportunidad de corregir los valores de las variables, (es decir las variables mantienen sus valores), y muestra una ventana informando que se eliminó correctamente la demanda.</p> 
Sección “Eliminar Todo”	
Acción del actor	Respuesta del Sistema
<p>1. El logístico selecciona la opción “Eliminar todo”.</p>	<p>2. El sistema muestra una ventana preguntando al usuario si está seguro de eliminar las demandas elaboradas.</p>


	
<p>3. El logístico selecciona el botón "OK".</p>	<p>4. El sistema elimina toda la planificación que se hizo con esas variables, y no da la oportunidad de corregir los valores de las variables, (es decir las variables se inician en cero), y muestra una ventana informando que se eliminó correctamente la demanda.</p> 
<p>Poscondiciones:</p>	<p>El sistema calcula los productos o servicios a demandar para las fuerzas armadas.</p>

<p>Nombre del Caso de Uso:</p>	<p>Obtener Variables</p>
<p>Uso:</p>	
<p>Actores:</p>	<p>Logístico por especialidades de la Unidad Demandante</p>
<p>Propósito:</p>	<p>Mostrar las variables (según especialidad, operación y usuario), que el logístico va a tener en cuenta para la planificación.</p>
<p>Resumen:</p>	<p>El caso de uso se inicia cuando el logístico selecciona "Obtener variables", el mismo finaliza cuando se muestra una ventana del navegador preguntándole al usuario que desea realizar con la misma: abrir la página, guardarla o cancelarlo todo.</p>
<p>Responsabilidades:</p>	<p>R2.</p>
<p>CU asociados:</p>	<p>-</p>
<p>Precondiciones:</p>	<p>El logístico se haya autenticado en la aplicación.</p>
<p>Interfaz I</p>	



(1) Opción del Menú Principal “Obtener variables”.

Flujo normal de los eventos

Acción del actor	Respuesta del Sistema
	1. El sistema muestra la interfaz I.
2. El logístico selecciona la opción “Obtener variables” (1).	
	3. El sistema muestra la ventana del navegador preguntándole al usuario que desea realizar: si abrirla, guardarla o cancelar todo. 
Poscondiciones:	

Nombre del Caso de	Reiniciar Demanda
Uso:	

Actores:	Logístico por especialidades de la Unidad Demandante
Propósito:	Eliminar la planificación realizada (por municipios, especialidad y operación).
Resumen:	El caso de uso se inicia cuando el logístico selecciona “Reiniciar Demanda”, para eliminar toda la planificación realizada hasta ese momento, el mismo finaliza cuando el sistema elimina toda la planificación de la demanda.
Responsabilidades:	R3.
CU asociados:	
Precondiciones:	El logístico se haya autenticado en la aplicación y haya realizado alguna planificación.

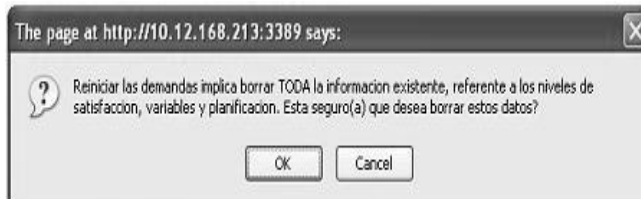
Interfaz I

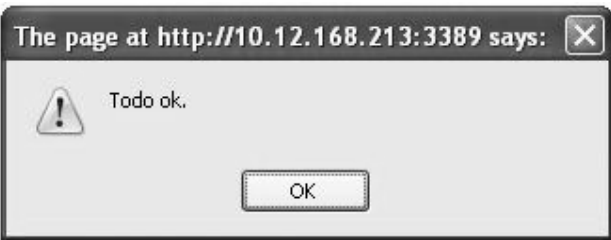


(1) Opción del Menú Principal “Reiniciar demanda”.

Flujo normal de los eventos

Acción del actor	Respuesta del Sistema
	1. El sistema muestra la interfaz I.
2. El logístico selecciona la opción “Reiniciar Demanda” (1).	
	3. El sistema muestra una ventana preguntándole al usuario si está de acuerdo con eliminar todos los datos registrados en la aplicación.



<p>4. El logístico selecciona la opción "OK".</p>	<p>5. El sistema elimina todas las planificaciones de la demanda realizadas y muestra una ventana informando que se eliminó correctamente la demanda.</p> 
<p>Poscondiciones:</p>	<p>La aplicación se queda sin ninguna planificación guardada en sus bases de datos.</p>

<p>Nombre del Caso de Uso:</p>	<p>Revisar informe _ planificación (MD1)</p>
<p>Actores:</p>	<p>Logístico por especialidades de la Unidad Demandante</p>
<p>Propósito:</p>	<p>Representar las demandas realizadas por el usuario que está trabajando en el sistema.</p>
<p>Resumen</p>	<p>El caso de uso se inicia cuando el logístico selecciona "Revisar MD1", el mismo finaliza cuando el sistema muestra una interfaz con las demandas realizadas.</p>
<p>Responsabilidades:</p>	<p>R6.</p>
<p>CU asociados:</p>	<p>-</p>
<p>Precondiciones:</p>	<p>El logístico se haya autenticado en la aplicación y haya realizado alguna planificación.</p>
<p>Interfaz I</p>	



(1) Opción del Menú Principal “MD-1/Revisar”.

Flujo normal de los eventos

Acción del actor	Respuesta del Sistema
	1. El sistema muestra la interfaz I.
2. El logístico selecciona la opción “MD1/Revisar” (1).	
	3. El sistema muestra la interfaz II en donde se muestra el árbol jerárquico con las demandas realizadas.

Interfaz II

The screenshot shows the SIDEM system interface with the title 'sistema informativo de la demanda'. The main menu includes 'Sistema', 'Demanda', 'Informes', and 'Administración'. A tree view is displayed under 'MD 1' showing a hierarchy of demands. To the right of the tree, there is explanatory text.

Este árbol representa las demandas realizadas por el usuario que esta trabajando en el sistema.

Está formado por niveles que particularizan las demandas realizadas. Estos son:

- Tipo de Tropa
- + Unidad Planificadora
- + Ejército
- + Región Militar
- + Municipio
- + Especialidad
- + Operación
- Nomenclatura

<p>4. El logístico va ir desplegando el árbol jerárquico, seleccionando las variables (Tipo de Tropa, Unidad Planificadora, Ejército, Región Militar (Provincia), Municipio, Especialidad, Operación, Nomenclatura) según corresponda con la planificación que realizó.</p>	<p>5. El sistema muestra al final la interfaz III que es el informe de planificación (MD1) correspondiente a los datos seleccionados por el usuario.</p>
---	--

Interfaz III

NECESIDADES A DEMANDAR PARA LA LUCHA ARMADA (MD-1)					
TROPA	Ministerio del Interior	UNIDAD	UM2000	MUNICIPIO	San Jose las Lajas
OPERACIÓN	Etapa Inicial de la Crisis	ESPECIALIDAD	Ingenieria	NOMENCLATURA	Central
NO.	PRODUCTOS Y/O SERVICIOS	UM	TOTAL A DEMANDAR		OBSERVACIONES
			CANTIDAD	IMPORTE	
1	Elementos Prefab. de HA del tipo P-2	Uno	765.00	21 519.00	
TOTAL DEMANDADO				21 519.00	
FECHA		ELABORADO		APROBADO	
29-05-2008					

Poscondiciones: El sistema muestra el MD1.

<p>Nombre del Caso de Uso:</p>	<p>Exportar informe _ planificación (MD1)</p>
<p>Actores:</p>	<p>Logístico por especialidades de la Unidad Demandante</p>
<p>Propósito:</p>	<p>Descargar el informe de la planificación (MD1), para su posterior envío a la Región Militar correspondiente.</p>
<p>Resumen:</p>	<p>El CU se inicia cuando el logístico selecciona "Exportar MD1", el mismo finaliza cuando se muestra una ventana del navegador preguntándole al usuario que desea realizar con la misma: abrir la página, guardarla o cancelarlo todo.</p>
<p>Responsabilidades:</p>	<p>R6.</p>
<p>CU asociados:</p>	<p></p>

Precondiciones:	El logístico se haya autenticado en la aplicación y haya realizado alguna planificación.
------------------------	--

Interfaz I



(1) Opción del Menú Principal “MD-1/Exportar”.

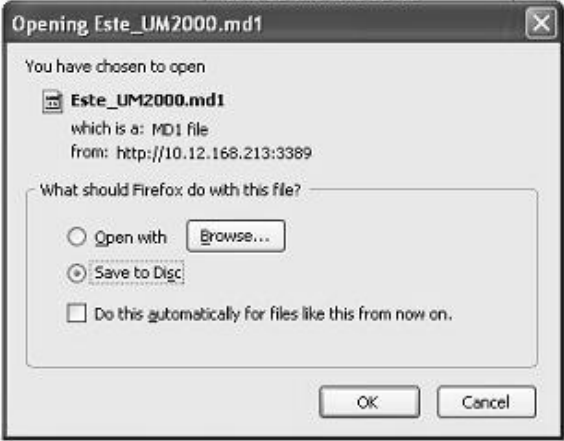
Flujo normal de los eventos

Acción del actor	Respuesta del Sistema
	1. El sistema muestra la interfaz I.
2. El logístico selecciona la opción “MD1/Exportar” (1).	
	3. El sistema muestra una interfaz II con una lista de las regiones por las que el logístico ha planificado.

Interfaz II



- (1) Listado de regiones.
- (2) Botón “Exportar” permite guardar el informe MD1 en la máquina del usuario.

<p>4. El logístico selecciona la Región Militar (1) perteneciente a la planificación que desea archivar y da "Exportar" (2).</p>	
	<p>5. El sistema muestra una ventana del navegador preguntándole al usuario que desea realizar con la misma: abrir la página, guardarla o cancelarlo todo.</p> 
<p>Poscondiciones</p>	

<p>Nombre del Caso de Uso:</p>	<p>Modificar Contraseña</p>
<p>Actores:</p>	<p>Logístico por especialidades de la Unidad Demandante</p>
<p>Propósito:</p>	<p>Permitirle al usuario la posibilidad de poder modificar su contraseña para acceder al sistema.</p>
<p>Resumen:</p>	<p>El CU se inicia cuando el logístico selecciona "Cambiar contraseña", el mismo finaliza cuando la contraseña del logístico ha sido modificada correctamente.</p>
<p>Responsabilidades:</p>	<p>R5.</p>
<p>CU asociados:</p>	
<p>Precondiciones:</p>	<p>El logístico se haya autenticado en la aplicación.</p>
<p>Interfaz I</p>	



(1) Opción del Menú Principal “Cambiar contraseña”.

Flujo normal de los eventos

Acción del actor	Respuesta del Sistema
	1. El sistema muestra la interfaz I.
2. El logístico selecciona la opción “Cambiar contraseña” (1).	
	3. El sistema muestra una interfaz II para que el usuario pueda introducir su nueva contraseña.

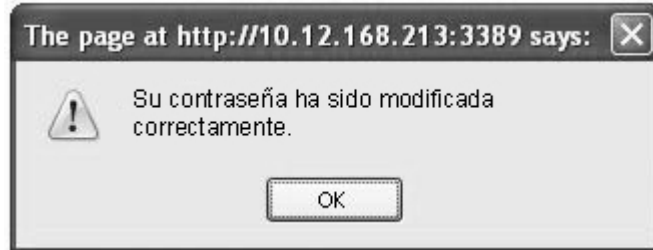
Interfaz II



- (1) Nueva contraseña de identificación del usuario.
- (2) Repetir la contraseña introducida en el campo de texto (1).
- (3) Botón “Aceptar” que invoca dicho proceso.

4. El logístico introduce su nueva contraseña (1) y la vuelve a poner en el campo de texto “Repetir” (2) y selecciona el botón “Aceptar”.

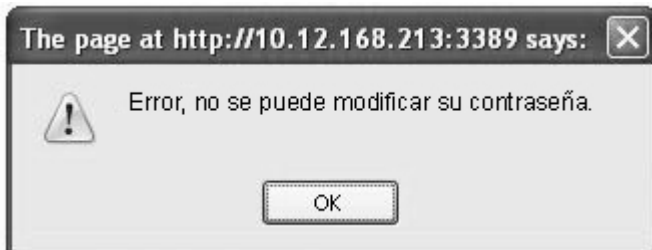
5. Si la nueva contraseña no coincide con la anterior que tenía el usuario, y a la hora de volver a ponerla como una forma de verificación escribe la que introdujo como su nueva contraseña. El sistema muestra un mensaje indicándole al usuario que su contraseña ha sido modificada correctamente.



Cursos Alternos

Línea 2:

Si la nueva contraseña coincide con la que tenía anteriormente el usuario o a la hora de repetirla no vuelve a escribir la contraseña nueva, el sistema mostrará un mensaje de error indicando al usuario que no puede modificar su antigua contraseña.



Poscondiciones

Se modifica la contraseña del usuario.

2.3 Conclusiones

Durante este capítulo fueron expuestas las características que contendrá el sistema, apoyándose para ello en el análisis de los actuales procesos de negocio, y habiendo identificado, quiénes son los actores y trabajadores que intervienen en el mismo y con cuáles actividades y entidades interactúan estos. Se identificaron además, los requisitos funcionales y no funcionales que debe cumplir el sistema en cuestión, y con ello fueron expuestos los casos de uso a tratar durante el desarrollo del mismo, con la correspondiente descripción textual de cada uno, lo cual provee de una visión general de qué es lo que el

sistema debe hacer, por lo que se está en condiciones de pasar a ver cómo es que el mismo va a realizar las operaciones antes descritas y con ello, darle solución a los problemas planteados.

CAPÍTULO 3: ANÁLISIS Y DISEÑO DEL SISTEMA

3.1 Introducción

En este capítulo se expondrá lo referente al análisis y diseño del sistema propuesto, cuyo objetivo principal será traducir los requisitos a una especificación que describa cómo implementar la misma. El análisis consiste en obtener una visión del sistema que se preocupa de ver QUÉ hace, de modo que sólo se interesa por los requisitos funcionales. Por otro lado, el diseño es un refinamiento del análisis que tiene en cuenta los requisitos no funcionales, en definitiva CÓMO cumple el sistema sus objetivos. El diseño debe ser suficiente para que el sistema pueda ser implementado sin ambigüedades. Para ello se presentarán el modelo de análisis de los casos de usos, el modelo de diseño con sus diagramas de interacción correspondientes, los diagramas de clases persistentes de la base de datos del sistema y para simplificar los diagramas de diseño se presentarán mecanismos de diseño.

3.2 Modelo de análisis

En la construcción del modelo de análisis se tienen que identificar las clases que describen la realización de los casos de uso y las relaciones entre ellas. Con esta información se construye el Diagrama de clases del análisis, que por lo general se descompone para agrupar las clases en paquetes. Esta descomposición tiene impacto por lo general en el diseño e implementación de la solución.

3.2.1 Diagramas de clases del análisis

RUP propone clasificar a las clases en:

1. Clases Interfaz: Modelan la interacción entre el sistema y sus actores.
2. Clases entidad: Modelan información que posee larga vida y que es a menudo persistente.
3. Clases de control: Coordinan los eventos necesarios para la realización o especificación del caso de uso, es decir son las que ejecutan el caso de uso.

Se realizó un diagrama de clases del análisis por cada caso de uso las cuales se muestran a continuación:

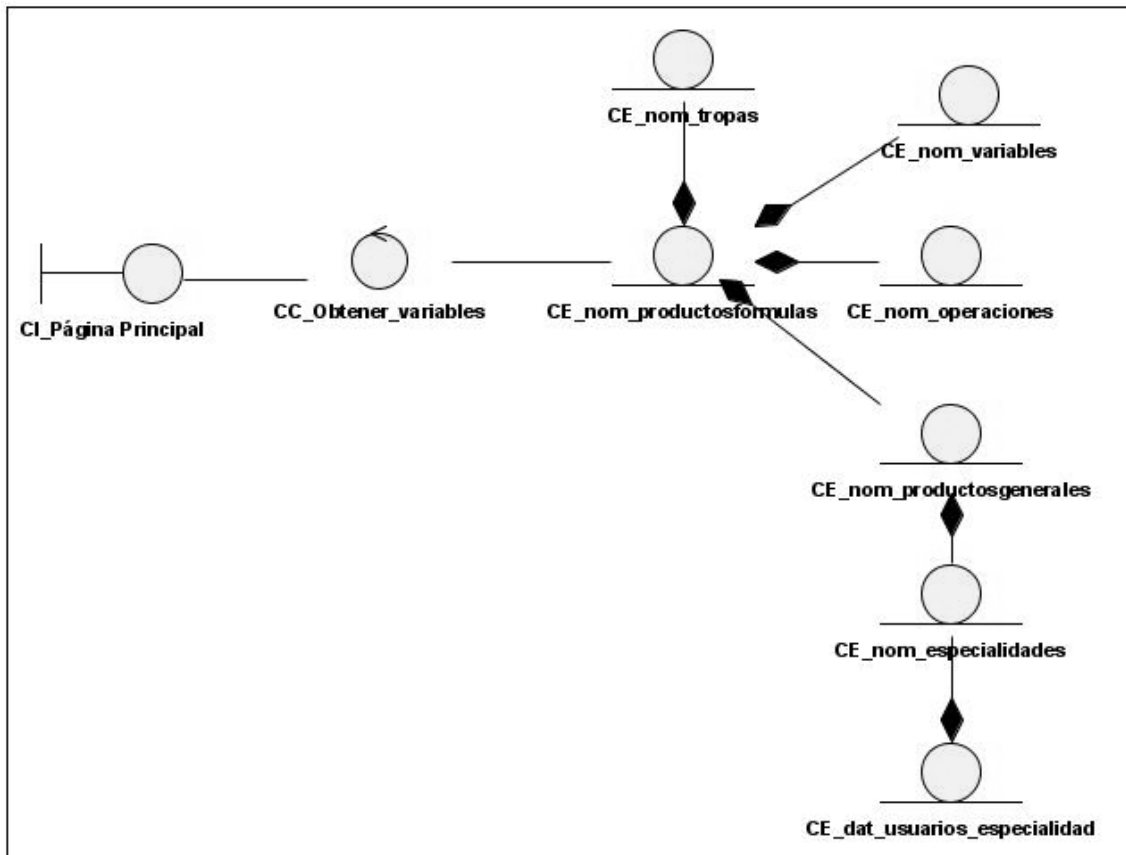


Figura 3.1- Diagrama de clases del análisis CU Obtener Variables

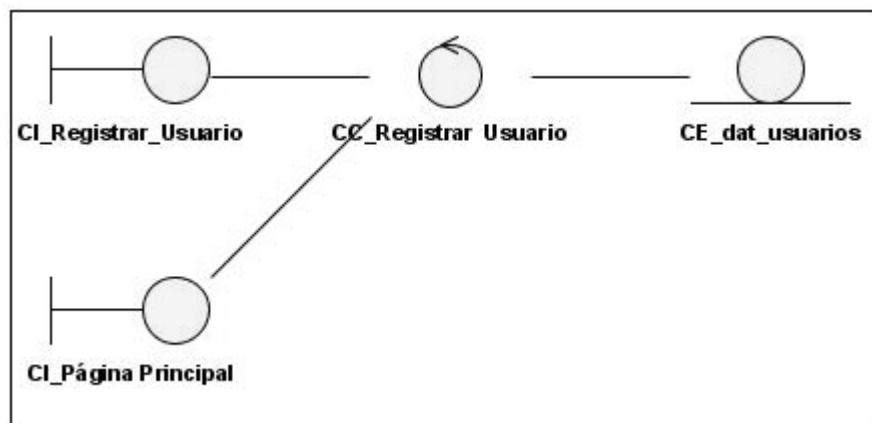


Figura 3.2- Diagrama de clases del análisis CU Registrar Usuario

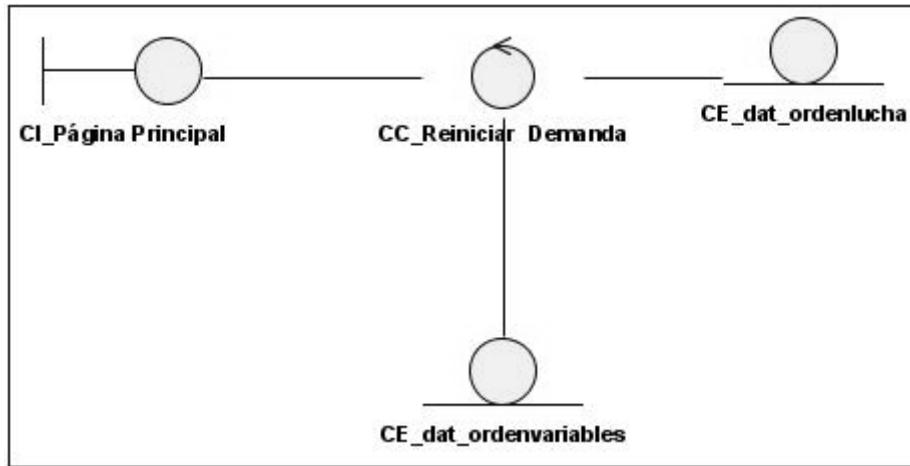


Figura 3.3- Diagrama de clases del análisis CU Reiniciar Demanda

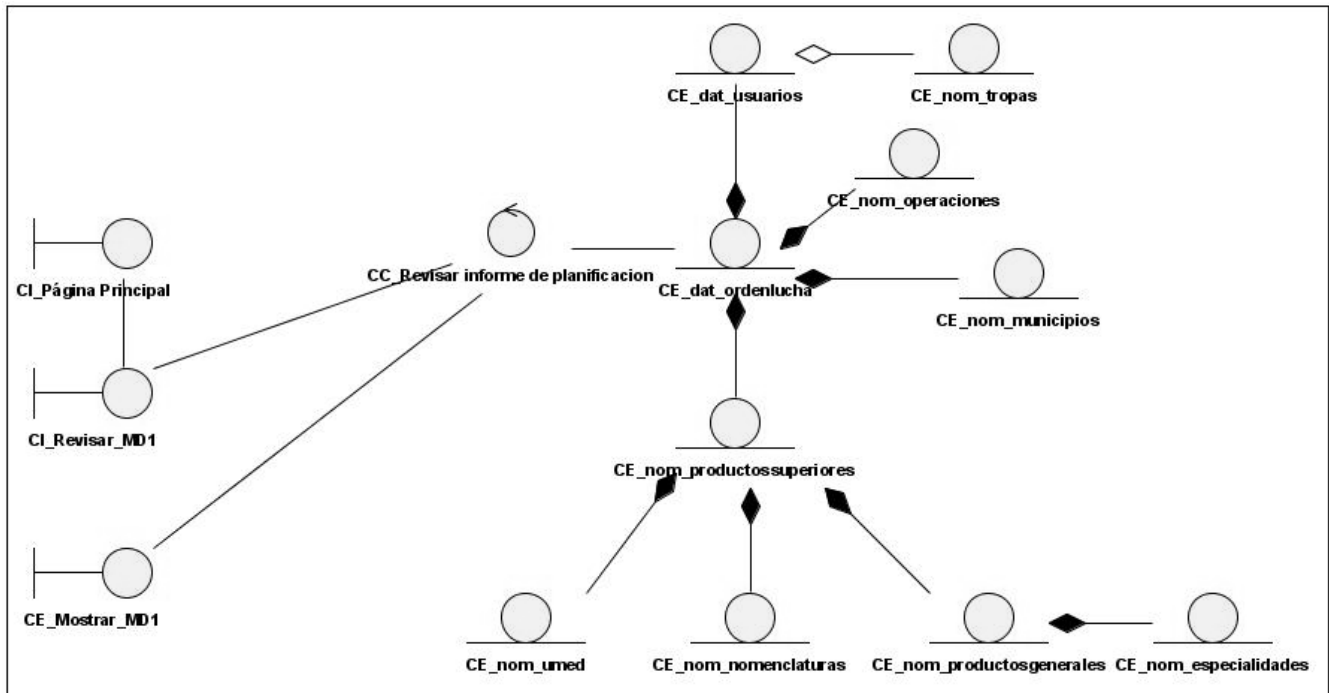


Figura 3.4- Diagrama de clases del análisis CU Revisar informe _ planificación

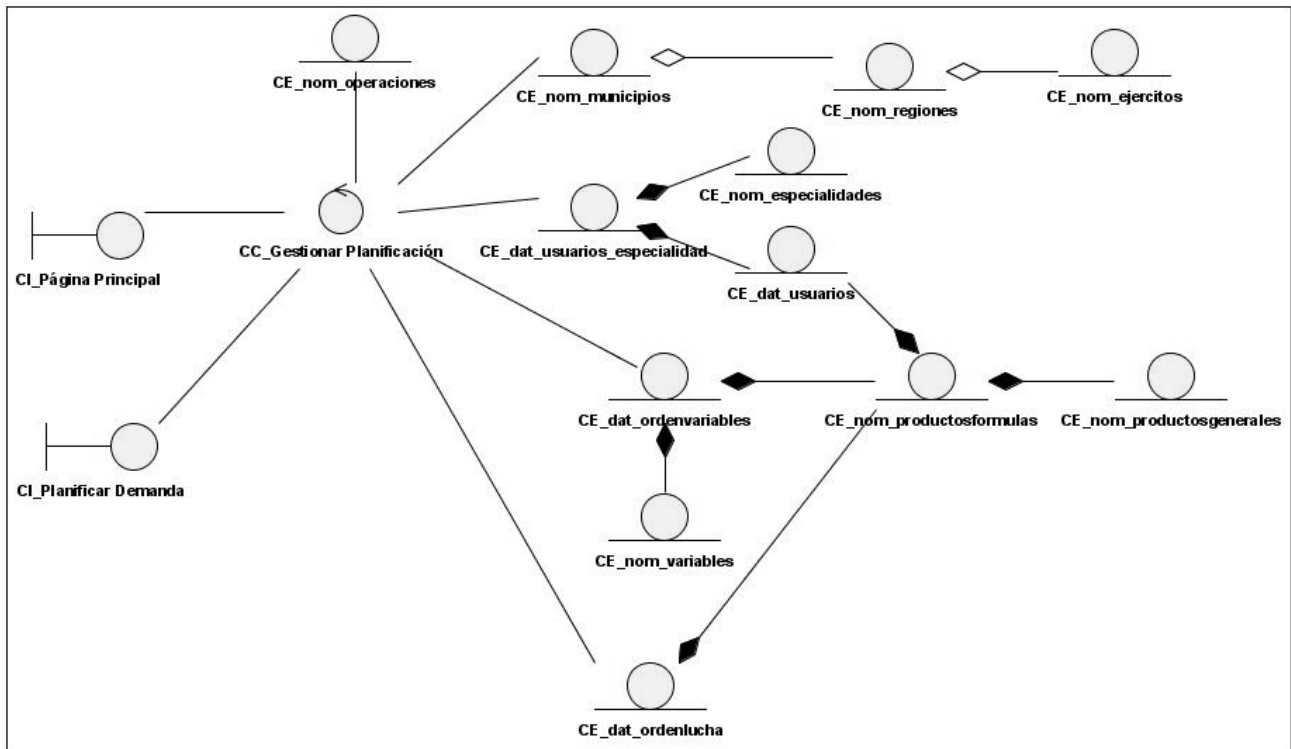


Figura 3.5- Diagrama de clases del análisis CU Planificar Demanda

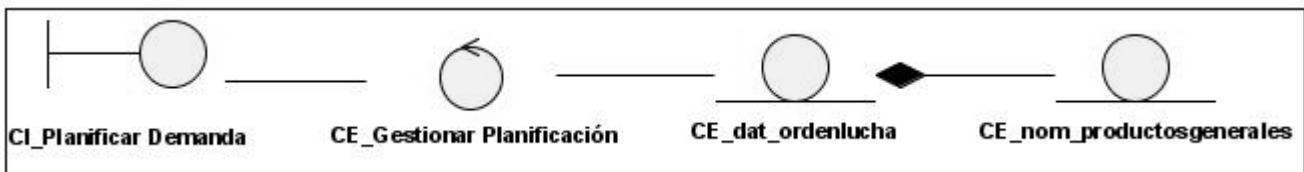


Figura 3.6- Diagrama de clases del análisis CU Planificar Demanda (Sección Eliminar Cantidades)

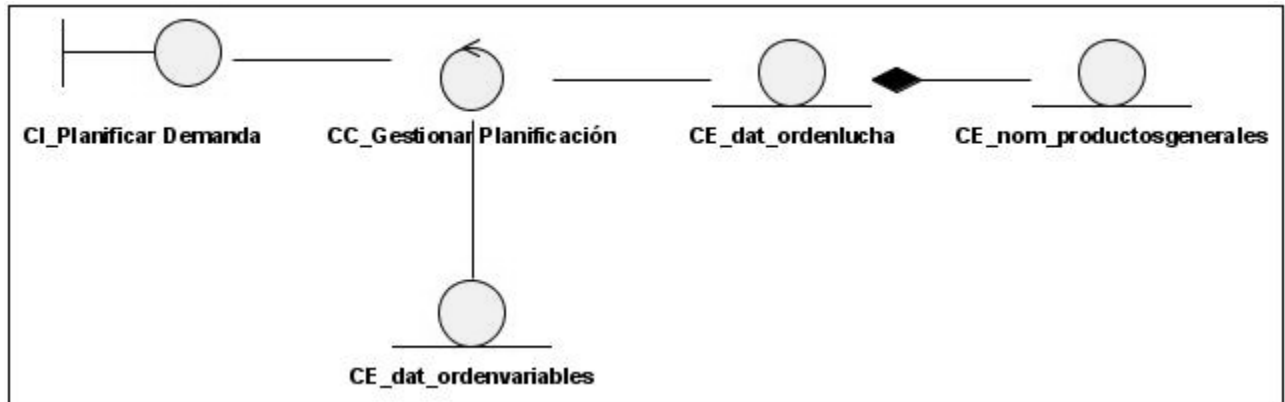


Figura 3.7- Diagrama de clases del análisis CU Planificar Demanda (Sección Eliminar Todo)

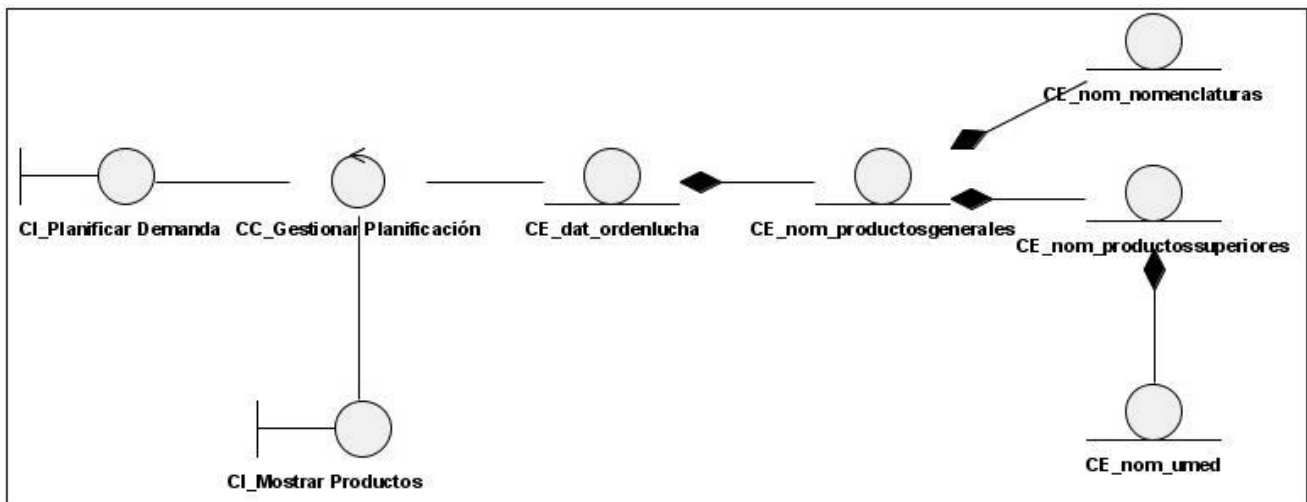


Figura 3.8- Diagrama de clases del análisis CU Planificar Demanda (Sección Verificar)

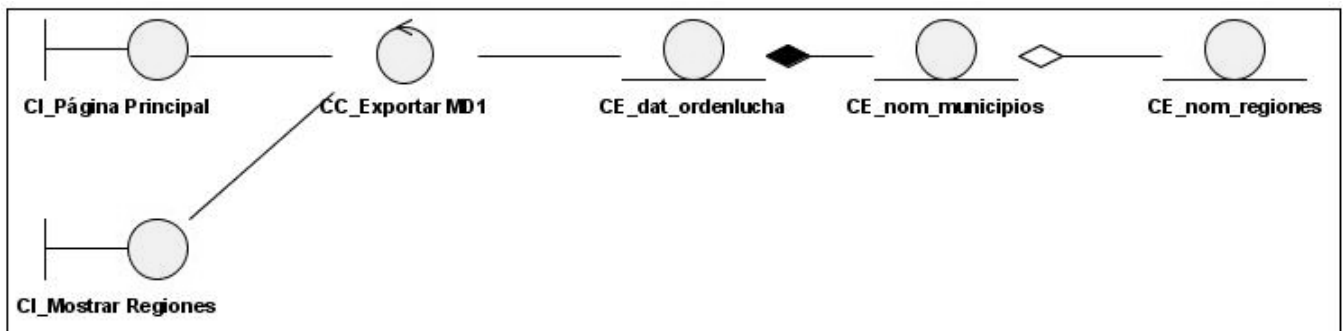


Figura 3.9- Diagrama de clases del análisis CU Exportar informe _ planificación

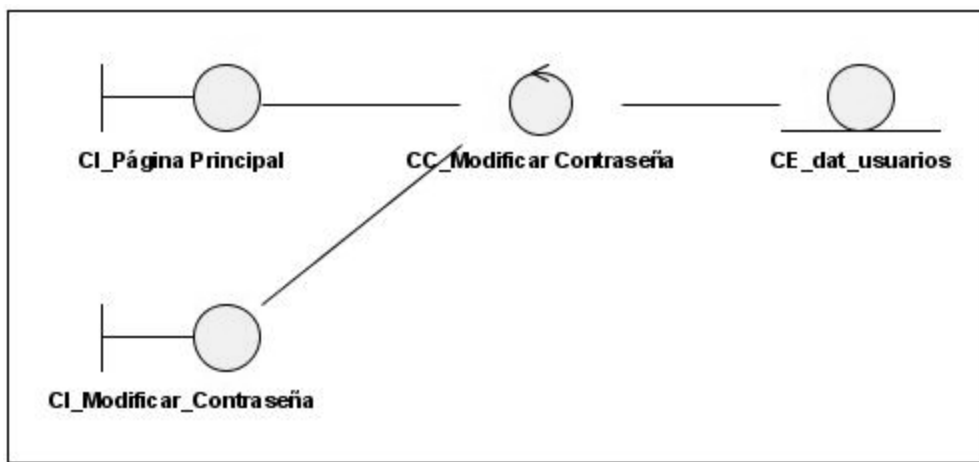


Figura 3.10- Diagrama de clases del análisis CU Modificar Contraseña

3.3 Arquitectura del sistema

“Una Arquitectura Software, también denominada *Arquitectura lógica*, consiste en un conjunto de patrones y abstracciones coherentes que proporcionan el marco de referencia necesario para guiar la construcción del software para un sistema de información. [21]

La arquitectura de software, tiene que ver con el diseño y la implementación de estructuras de software de alto nivel. Es el resultado de ensamblar un cierto número de elementos arquitectónicos de forma adecuada para satisfacer la mayor funcionalidad y requerimientos de desempeño de un sistema, así como requerimientos no funcionales, como la confiabilidad, escalabilidad, portabilidad, y disponibilidad. [21]

Generalmente, no es necesario inventar una nueva arquitectura software para cada sistema de información. Lo habitual es adoptar una arquitectura conocida en función de sus ventajas e inconvenientes para cada caso en concreto. Ejemplo de arquitecturas tenemos:

Cliente-servidor: Donde el software reparte su carga de cómputo en dos partes independientes pero sin reparto claro de funciones.

Arquitectura de tres niveles. Generalización de la arquitectura cliente-servidor donde la carga se divide en tres partes con un reparto claro de funciones: una capa para la presentación, otra para la lógica del negocio y otra para el almacenamiento.

- Capa de presentación: En ella se representan las clases interfaces del sistema (páginas de la aplicación con que el usuario interactúa), es la que presenta la información y captura los datos introducidos por el usuario. Esta capa solo se comunica con la capa de negocio.
- Capa de negocio: Es la que maneja y controla toda la lógica del proceso. En ella se encuentran todos los métodos, las clases, que se necesiten para la ejecución del mismo, es la encargada de recibir los datos que son enviados de la capa de presentación, y pedir de la capa de acceso a datos lo necesario para poder dar respuesta a una petición realizada por el usuario.
- Capa de acceso a datos: Aquí es donde se encuentran todas las clases persistentes con que cuenta el caso de uso, para obtener el acceso a los datos se utilizan los componentes lógicos de acceso a los datos. Estos componentes abstraen la semántica del almacén de datos subyacente y la tecnología de acceso a datos (como PDO) y proporcionan una interfaz simple de programación para la recuperación y realización de operaciones con datos.

Modelo Vista Controlador (MVC): es un patrón de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos. El patrón MVC se ve frecuentemente en aplicaciones Web, donde la vista es la página HTML y el código que provee de datos dinámicos a la página.

- Modelo: Esta es la representación específica del dominio de la información sobre la cual funciona la aplicación. El modelo es otra forma de llamar a la capa de dominio. La lógica de dominio añade significado a los datos; por ejemplo, calculando si hoy es el cumpleaños del usuario o los totales impuestos en un carrito de compra.
- Vista: Este presenta el modelo en un formato adecuado para interactuar, usualmente un elemento de interfaz de usuario.
- Controlador: Este responde a eventos, usualmente acciones del usuario e invoca cambios en el modelo y probablemente en la vista.

Arquitectura n-Capas: La que más comúnmente se usa es la de cuatro capas, la capa que se agrega es la que surge de separar definitivamente las reglas de negocio de la de Acceso a Datos. Esta arquitectura

trae consigo 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 este sistema, si se han respetado las reglas básicas de diseño, entonces no se deberán afrontar grandes modificaciones. [21]

Descripción de la arquitectura que se propone:

En cuanto a la arquitectura a utilizar en el sistema, se decidió la Arquitectura de 3 capas, debido a que permite trabajar de una forma más organizada, lo cual trae grandes beneficios a la hora de hacer cambios y corregir errores.

Ventajas de una arquitectura de tres capas

- Las llamadas de la interfaz del usuario, en la estación de trabajo, al servidor de capa intermedia, son más flexibles que en el diseño de dos capas, ya que la estación sólo necesita transferir parámetros a la capa intermedia.
- Con la arquitectura de tres capas, la interfaz del cliente no es requerida para comprender o comunicarse con el receptor de los datos. Por lo tanto, esa estructura de los datos puede ser modificada sin cambiar la interfaz del usuario en la PC.
- El código de la capa intermedia puede ser reutilizado por múltiples aplicaciones si está diseñado en formato modular. Esto puede reducir los esfuerzos de desarrollo y mantenimiento, así como los costos de migración.
- La separación de roles en tres capas, hace más fácil reemplazar o modificar una capa sin afectar a los módulos restantes.
- Separando la aplicación de la base de datos, hace más fácil utilizar nuevas tecnologías de agrupamiento y balance de cargas.
- Separando la interfaz del usuario de la aplicación, libera gran procesamiento a la estación de trabajo y permite que las actualizaciones de la aplicación sean centralizadas en el servidor de aplicaciones. [22]

3.4 Modelo de diseño

En el diseño modelamos el sistema y encontramos su forma (incluida la arquitectura) para que soporte todos los requisitos, incluyendo los no funcionales y las restricciones que se le suponen. Una entrada esencial en el diseño es el resultado del análisis, o sea el modelo de análisis, que proporciona una comprensión detallada de los requisitos, además impone una estructura del sistema que se debe tratar de conservar lo más fielmente posible cuando se da forma al sistema. Definiendo así como sus principales propósitos:

- Transformar los requerimientos en un diseño de como el sistema debe ser.
- Desarrollar una robusta arquitectura del sistema.
- Adaptar el diseño para que se corresponda con el entorno de implementación, diseñando sus funcionalidades.

3.4.1 Diagramas de secuencia

Los diagramas de secuencia (llamados diagramas de interacción) se utilizan para modelar los aspectos dinámicos de un sistema. Un diagrama de interacción consiste en un conjunto de objetos y sus relaciones, incluyendo los mensajes que se pueden enviar entre ellos. Los diagramas de secuencia destacan el orden temporal de los mensajes. Se realizó un diagrama de secuencia por cada caso de uso, las cuales se muestran a continuación (los diagramas de secuencia de las secciones “Verificar”, “Eliminar todo”, “Eliminar cantidades” del CU Planificar Demanda se encuentran en el Anexo 7 de este documento).

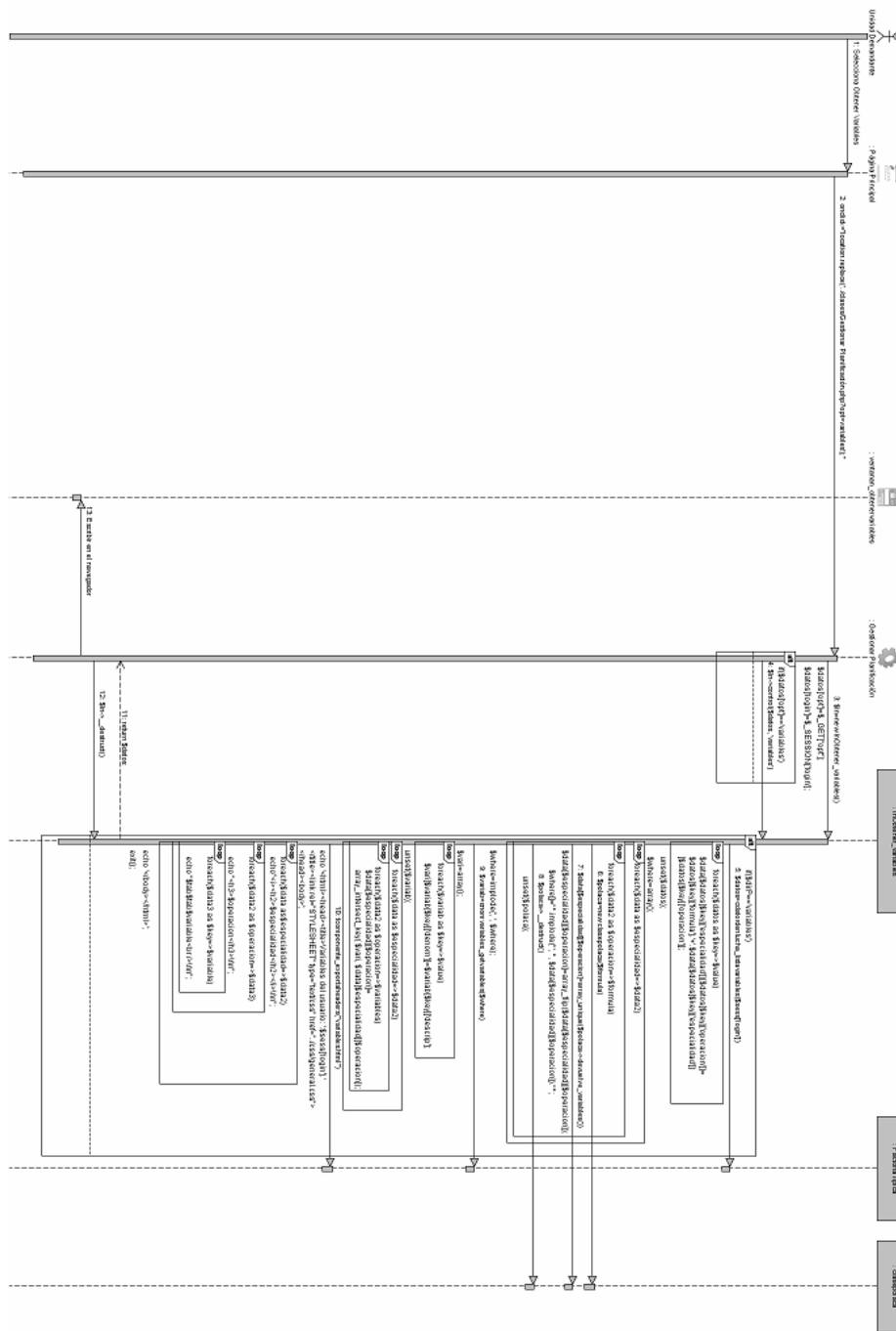


Figura 3.11- Diagrama de secuencia CU Obtener Variables

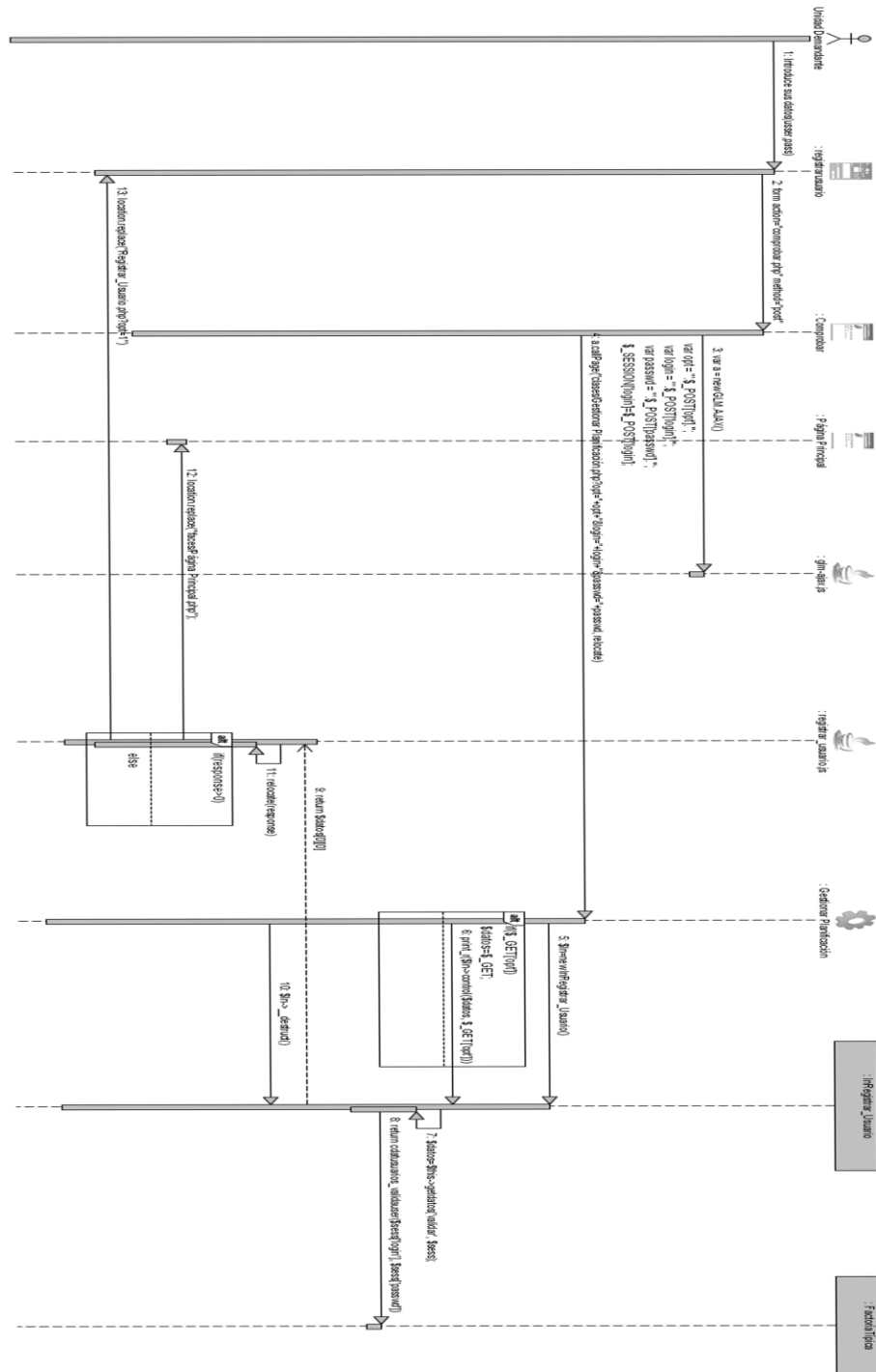


Figura 3.12- Diagrama de secuencia CU Registrar Usuario

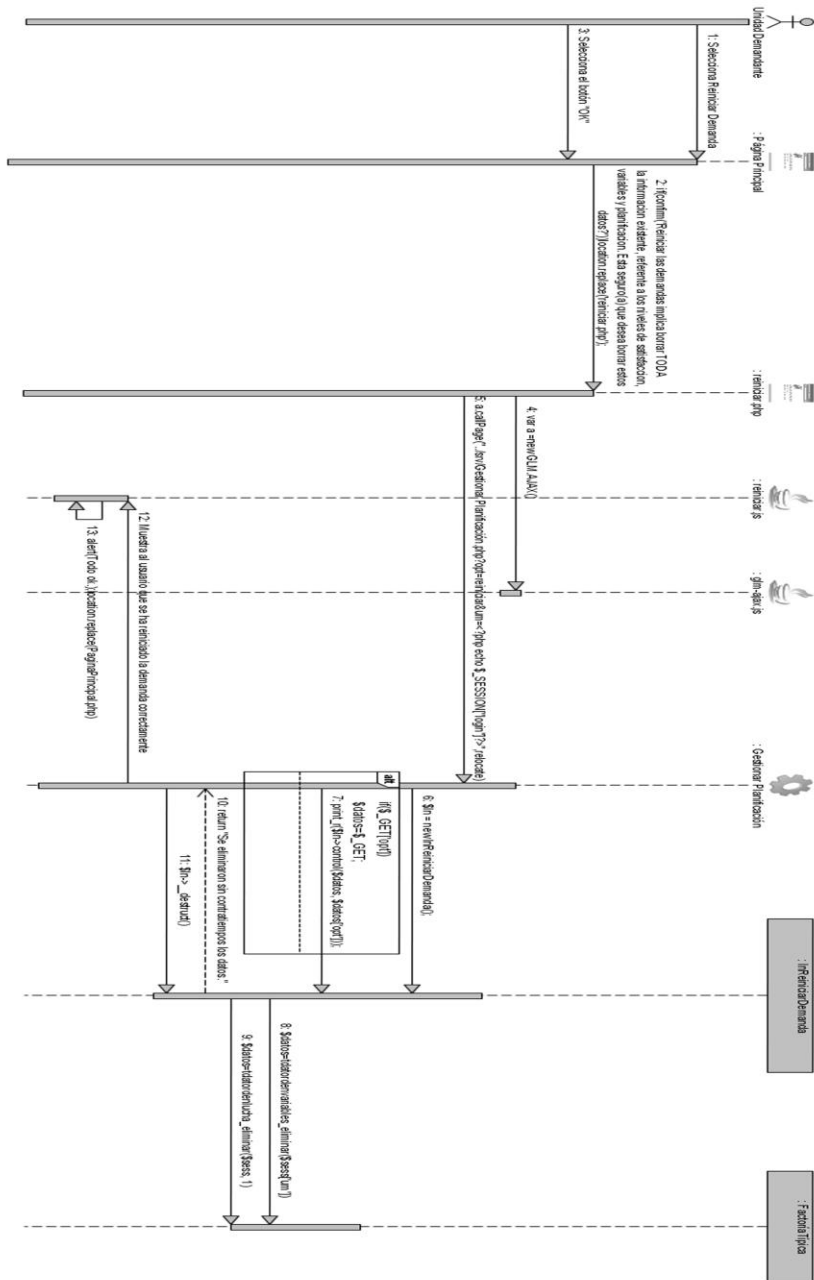


Figura 3.13- Diagrama de secuencia CU Reiniciar Demanda

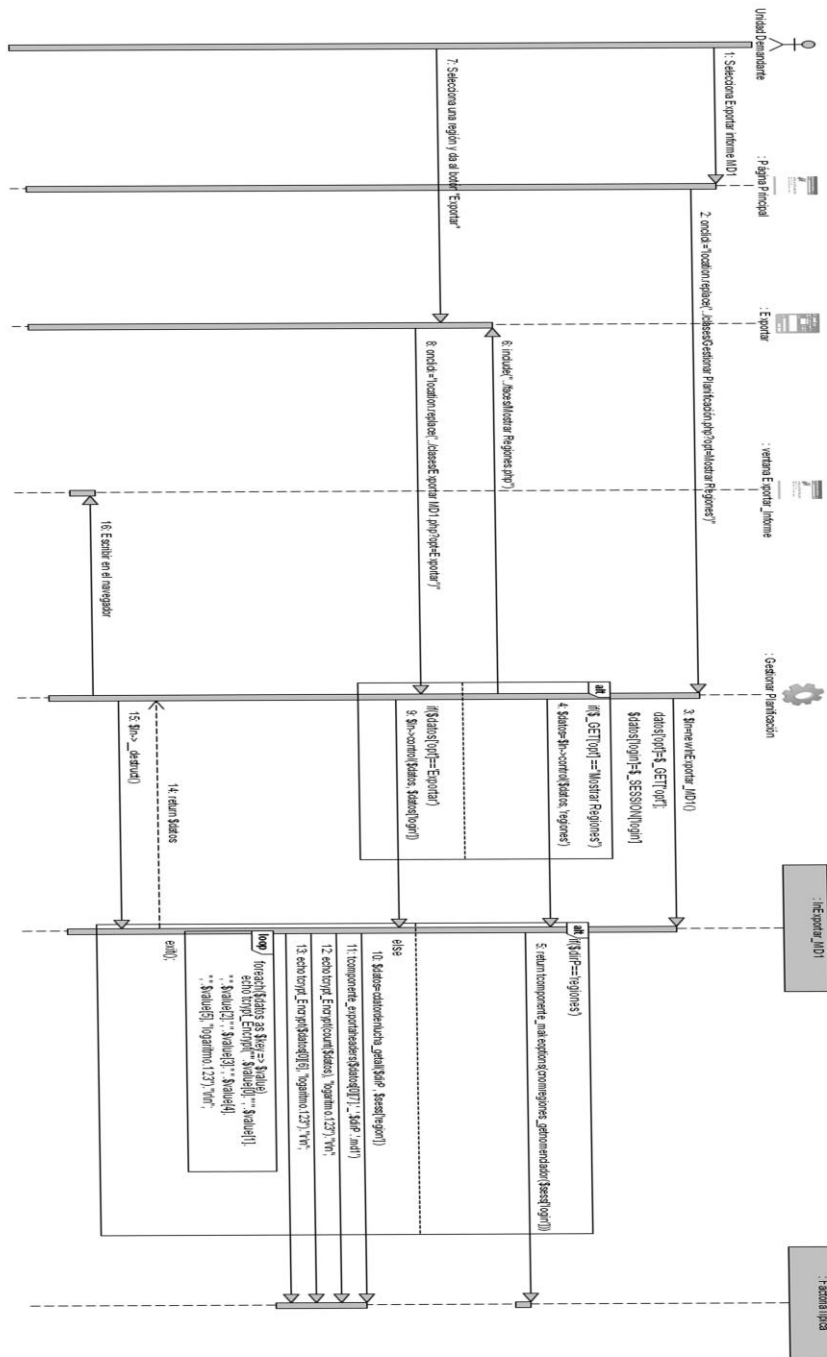


Figura 3.15 - Diagrama de secuencia CU Exportar informe _ planificación

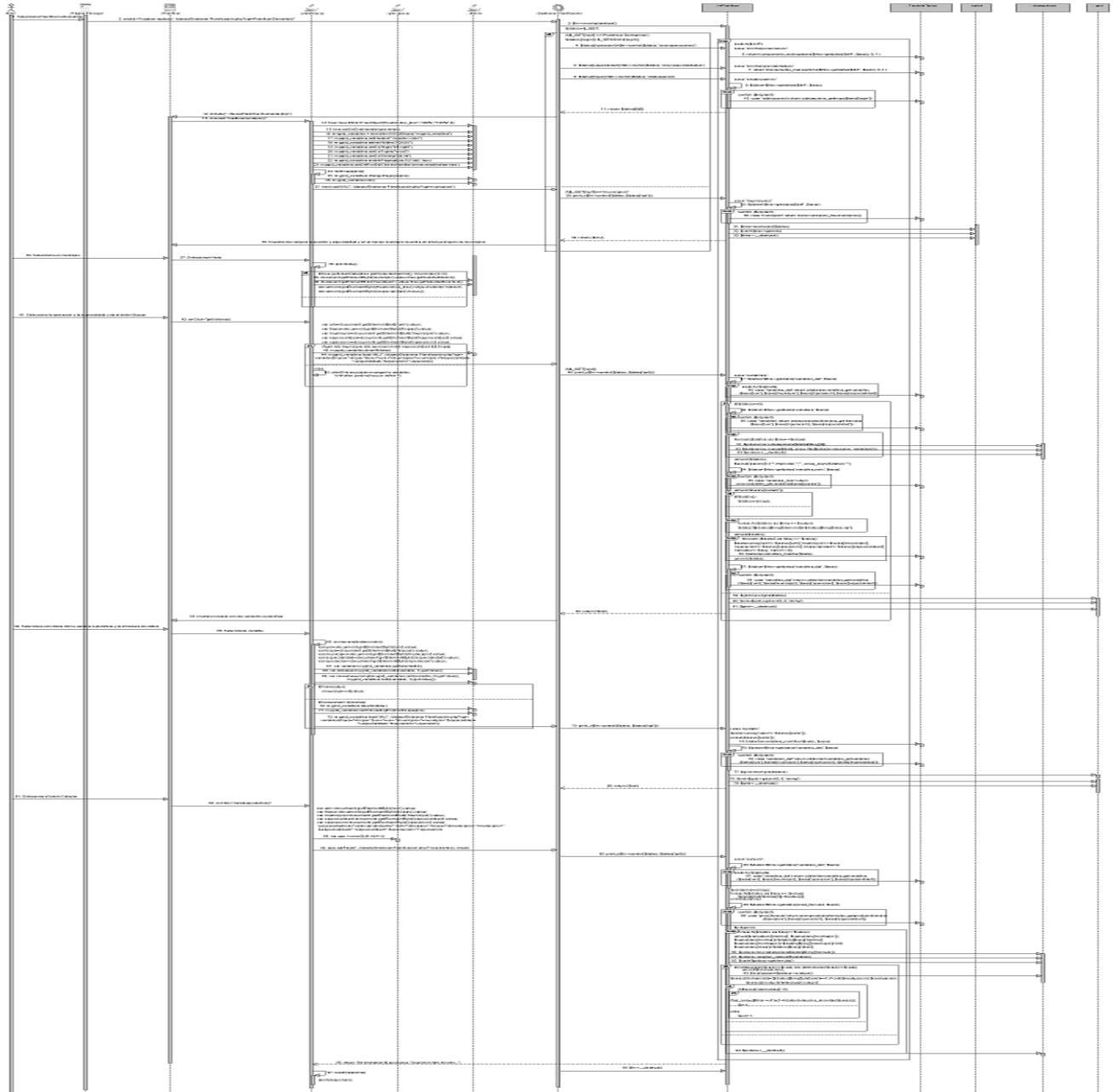


Figura 3.16 - Diagrama de secuencia CU Planificar Demanda

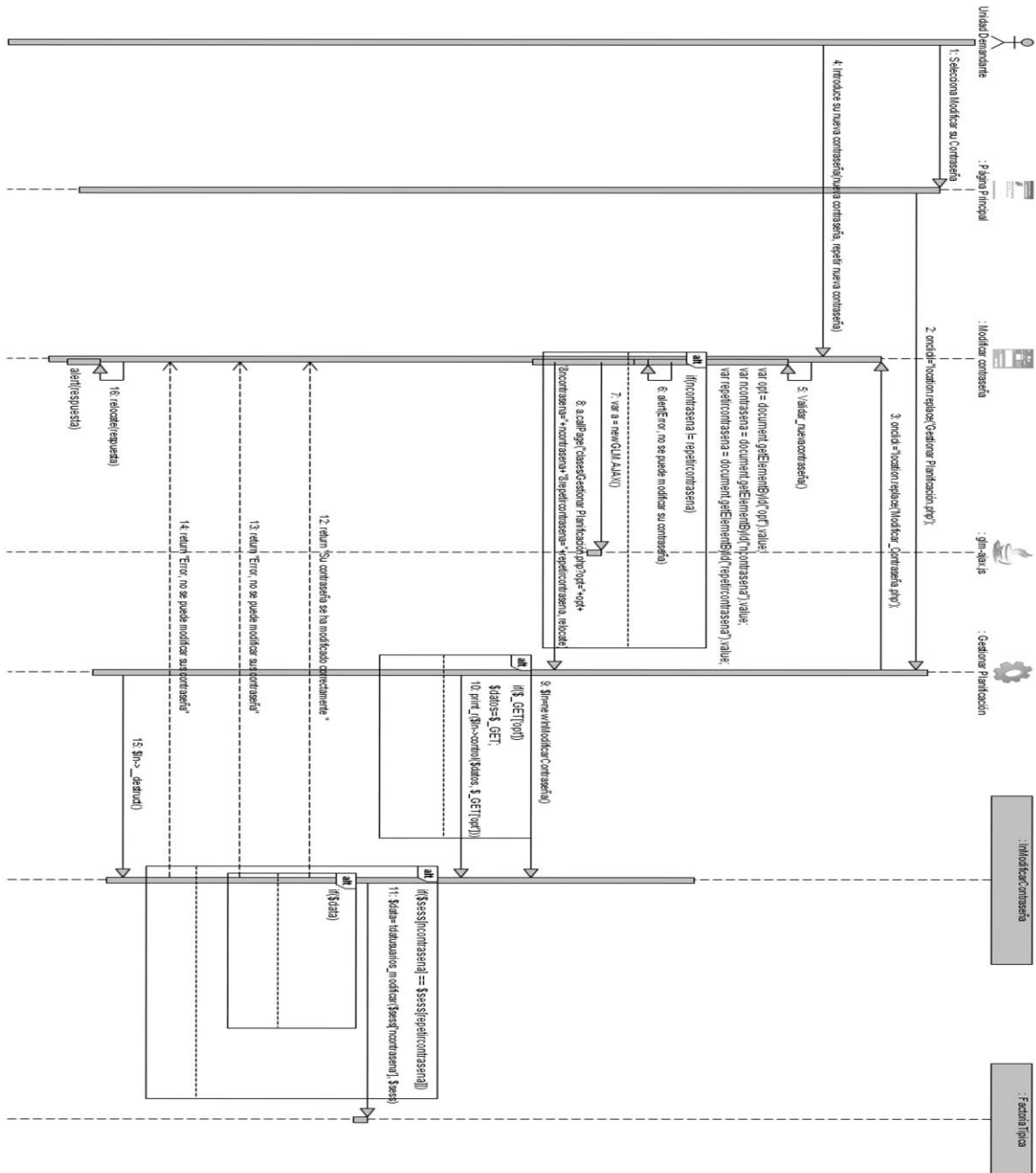


Figura 3.17 - Diagrama de secuencia CU Modificar Contraseña

3.4.2 Diagramas de clases de diseño Web

Los **diagramas de clases** son diagramas de estructura estática que muestran las clases del sistema y sus interrelaciones. Los diagramas de clase son el pilar básico del modelado con UML, siendo utilizados tanto para mostrar lo que el sistema puede hacer (análisis), como para mostrar cómo puede ser construido (diseño). Se realizó un diagrama de clases del diseño por cada caso de uso las cuales se muestran a continuación:

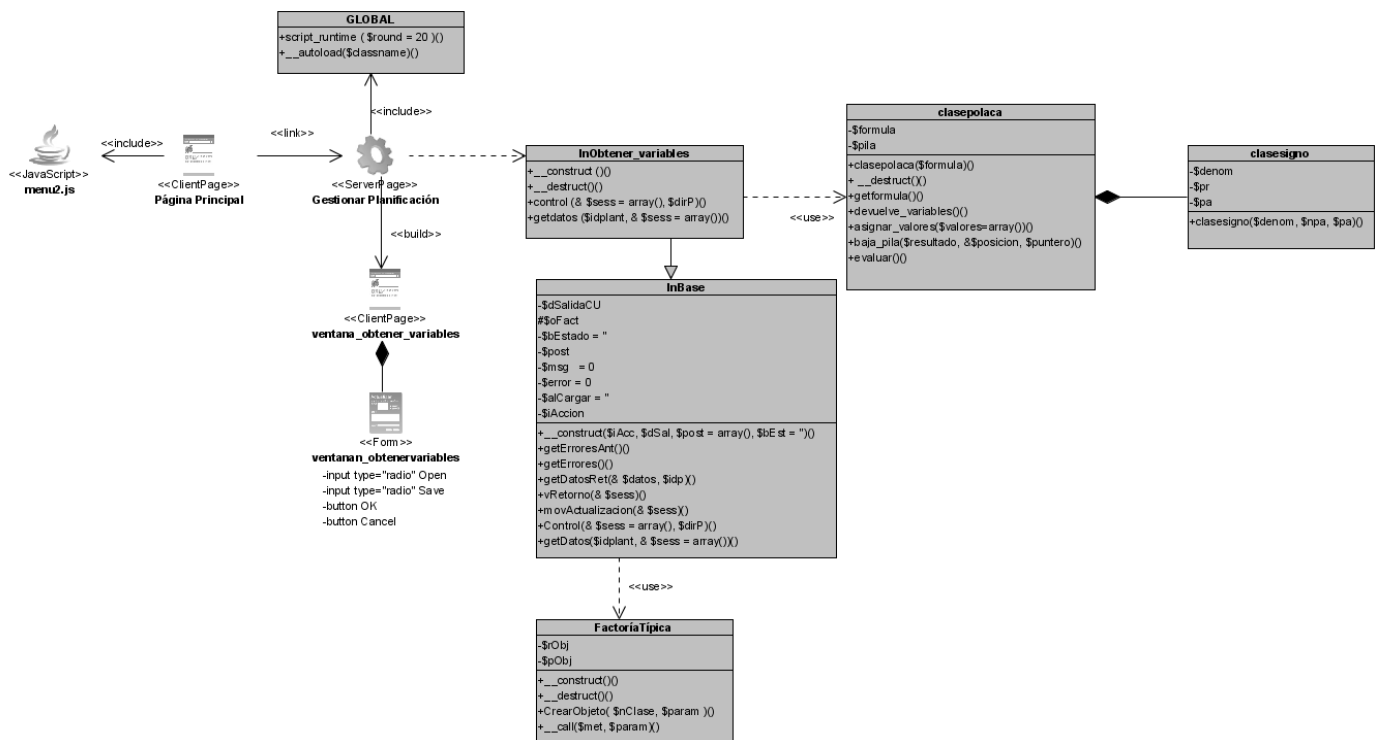


Figura 3.18- Diagrama de clases del diseño CU Obtener Variables

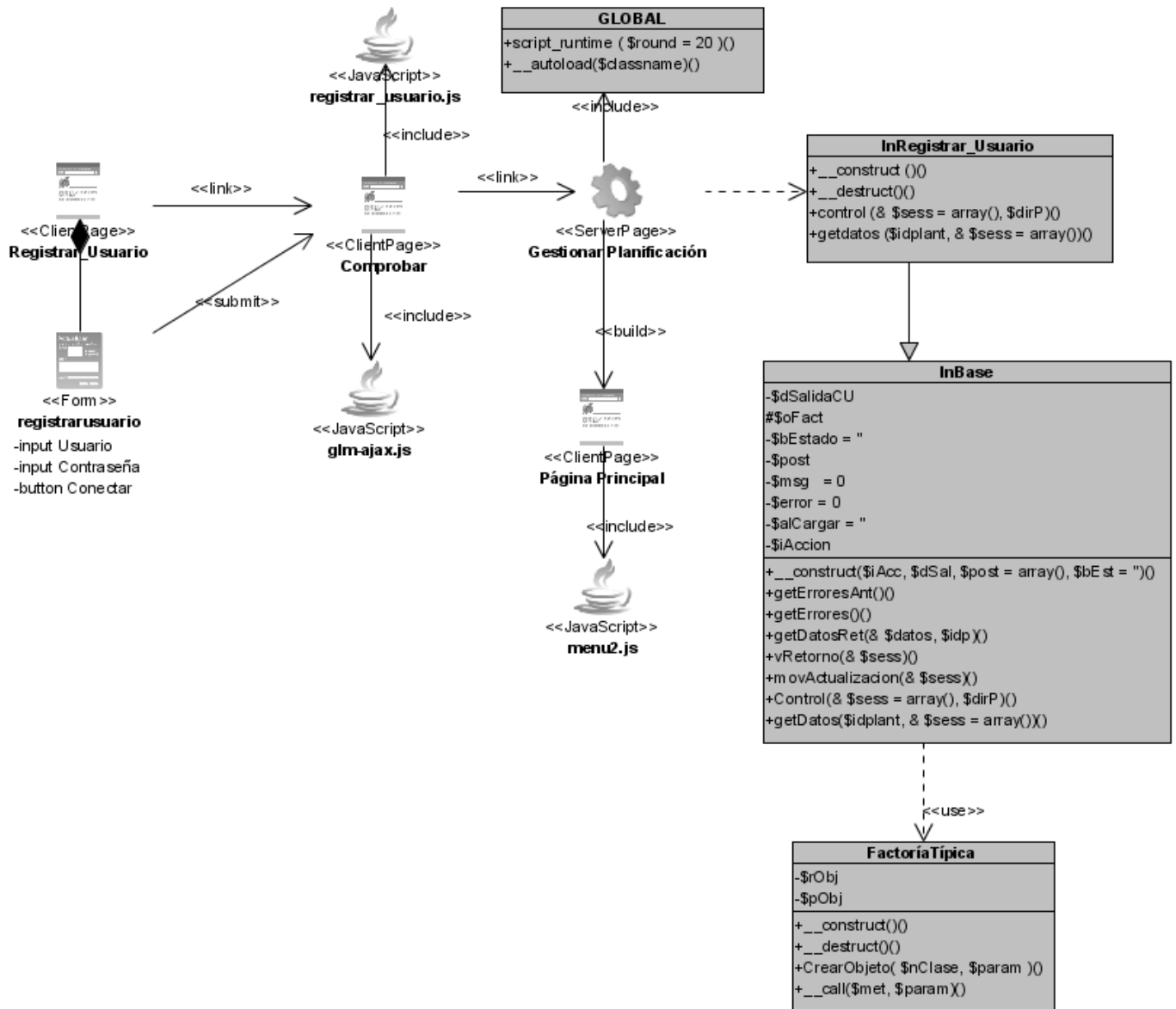


Figura 3.19- Diagrama de clases del diseño CU Registrar Usuario

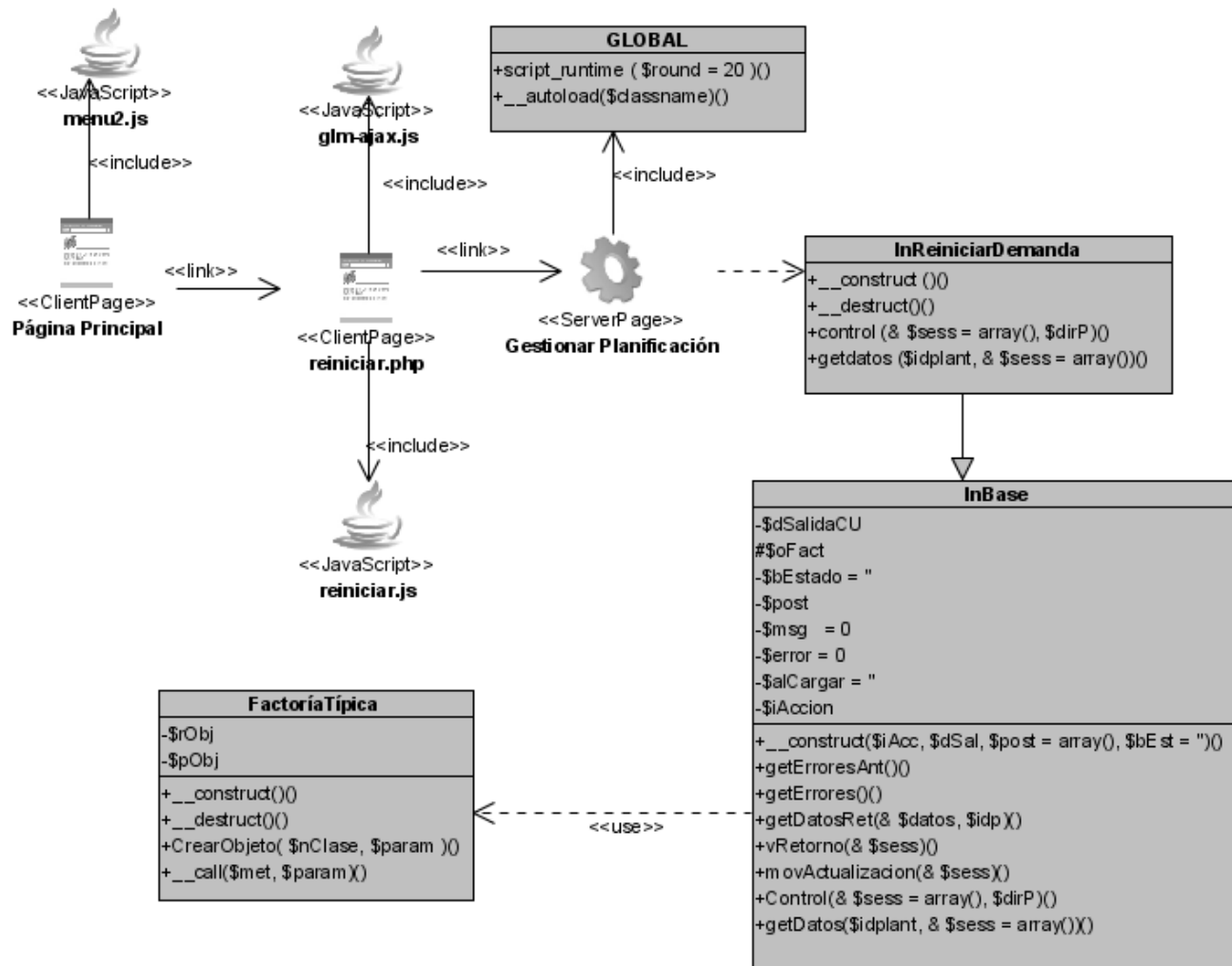


Figura 3.20- Diagrama de clases del diseño CU Reiniciar Demanda

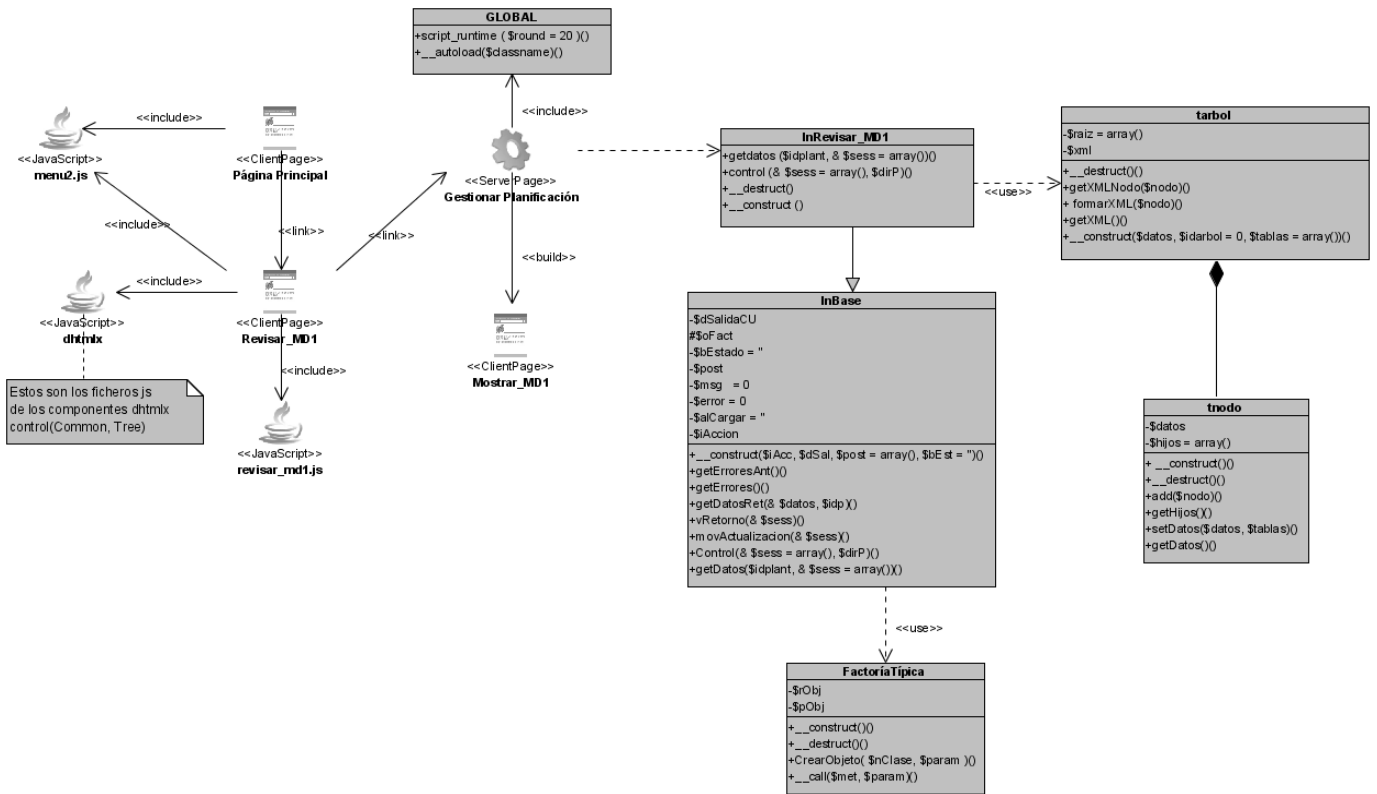


Figura 3.21- Diagrama de clases del diseño CU Revisar informe _ planificación

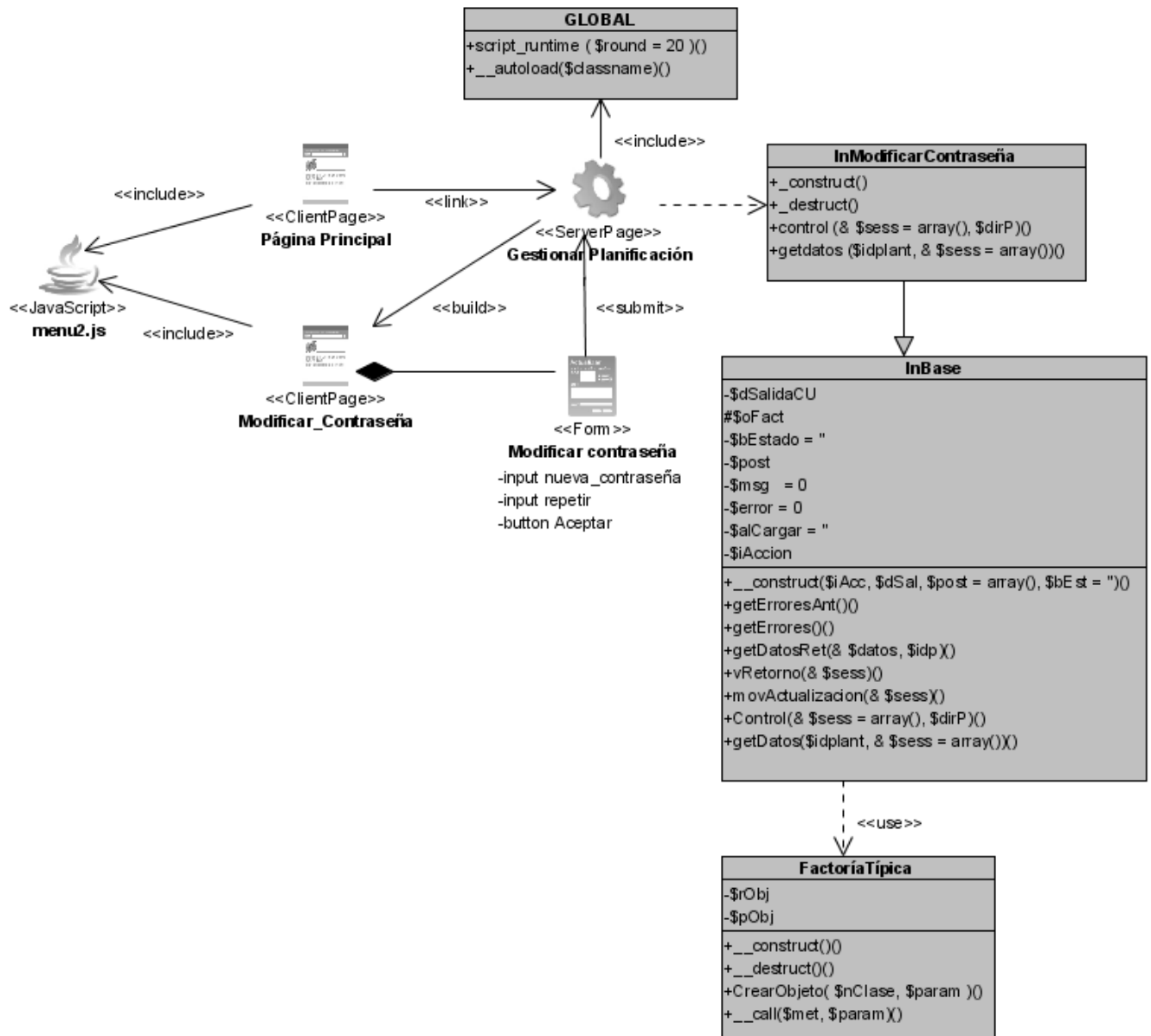


Figura 3.24- Diagrama de clases del diseño CU Modificar Contraseña

3.4.3 Diseño de la Base de Datos

La puesta en práctica de la base de datos es el paso final en el desarrollo de aplicaciones de soporte del negocio. Tiene que conformarse con los requisitos del proceso del negocio, que es la primera abstracción de la vista de la base de datos. Las Bases de datos necesitan de una definición de su estructura que le

3.4.3.2 Modelo físico de datos

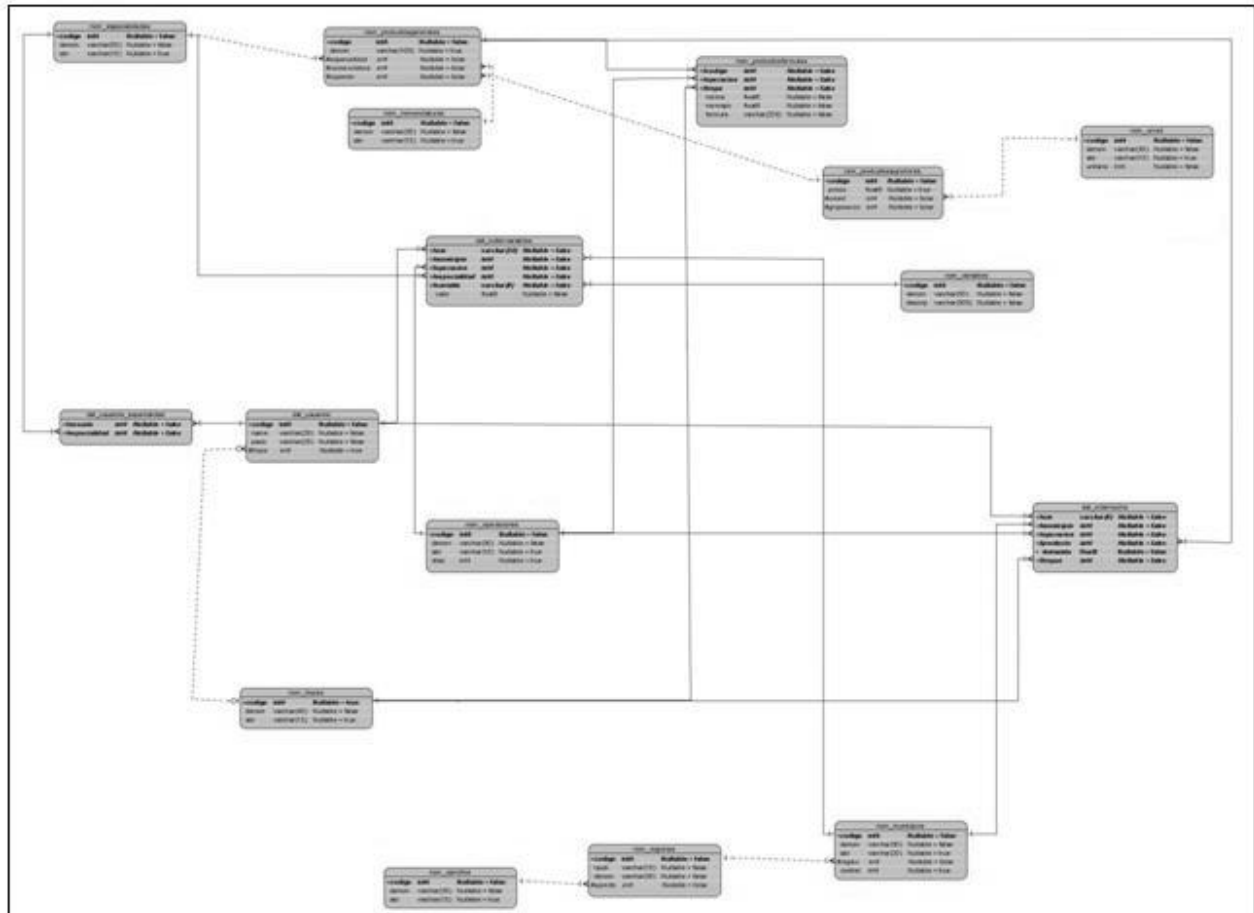


Figura 3.26- Modelo físico de datos

3.4.3.3 Descripción de las tablas

Nombre: dat_ordenlucha		
Descripción: Es la encargada del almacenamiento de los datos de las demandas, de los productos y/o servicios a planificar		
Atributo	Tipo	Descripción
Llave primaria (um, municipio, operacion, producto, tropas)		

um	varchar	Este viene de dat_usuarios. Identificador de la unidad militar que realiza la planificación.
municipio	integer	Viene de nom_municipios. Identificador del municipio donde se realiza la planificación.
operacion	integer	Viene de nom_operaciones. Identificador de la operación que se le realiza la planificación.
producto	integer	Producto que se demanda.
demanda	double	Cantidad de productos y/o servicios que se demandan.
tropas	integer	Viene de nom_tropas, se corresponde al tipo de tropa que pertenece la unidad militar.

Nombre: nom_productosgenerales		
Descripción: En ella se definen todos los productos y/o servicios y sus características generales.		
Atributo	Tipo	Descripción
codigo (llave primaria)	integer	Código del producto general.
denom	varchar	Denominación del producto y/o servicios.
especialidad	integer	Viene de nom_especialidades, identificador de la especialidad que se le hace la planificación.
nomenclatura	integer	Viene de nom_nomenclaturas. Identificador de la nomenclatura que se le calcula los productos.
superior	integer	Viene de nom_productosuperiores. Identificador del producto padre.

Nombre: nom_productosuperiores		
Descripción: En ella se definen los productos superiores, forman parte de los productos generales pero con características más específicas		
Atributo	Tipo	Descripción

codigo (llave primaria)	integer	Coincide con el código de productos generales.
precio	double	Precio que tiene el producto.
umed	integer	Viene del nomenclador nom_umed

Nombre: nom_productosformulas		
Descripción: En esta tabla se encuentran definidas las fórmulas que rigen el cálculo de los productos y/o servicios a demandar por especialidades, operaciones y tropas.		
Atributo	Tipo	Descripción
Llave primaria (codigo, operacion, tropa)		
codigo	integer	Es el código de la tabla nom_productosgenerales.
operacion	integer	Viene de nom_operaciones. Identificador de la operación que se le realiza la planificación.
tropa	integer	Viene de nom_tropas, se corresponde al tipo de tropa que pertenece la unidad militar.
norma	double	Norma del cual se rigen para el cálculo de los productos y/o servicios.
normapc	double	Es la norma en por ciento.
formula	varchar	Fórmula para calcular el producto.

Nombre: dat_ordenvariables		
Descripción: Esta es la tabla principal que define los valores que adopta la variable en cada una de las especificaciones del campo llave, para luego planificar con ellas.		
Atributo	Tipo	Descripción
Llave primaria (um, municipio, operacion, especialidad, variable)		
um	varchar	Viene de la tabla dat_usuarios.
operacion	integer	Viene de la tabla nom_operaciones.
municipio	integer	Viene de la tabla nom_municipios.
especialidad	integer	Viene de la tabla nom_especialidades.

variable	varchar	Viene de nom_variables. Identificador de la variable.
valor	double	Valor de la variable que se planifica.

Nombre: dat_usuarios_especialidad.		
Descripción: Esta tabla es una relación entre las tablas nom_especialidades y dat_usuarios, la cual tiene los usuarios que pueden planificar dada las especialidades.		
Atributo	Tipo	Descripción
Llave primaria (usuario, especialidad)		
usuario	integer	Viene de la tabla dat_usuarios. Identificador del usuario que realiza la planificación.
especialidad	integer	Viene de nom_especialidades. Identificador de la especialidad que se le hace la planificación.

Nombre: dat_usuarios		
Descripción: Esta tabla contiene todos los datos del usuario.		
Atributo	Tipo	Descripción
codigo (llave primaria)	integer	Código del usuario.
name (único)	varchar	Nombre del usuario.
pass	varchar	Contraseña del usuario.
tropa	integer	Viene de nom_tropas. Identificador de la tropa al que pertenece el usuario.

Nombre: nom_regiones		
Descripción: Esta tabla contiene todos los datos de la Región Militar.		
Atributo	Tipo	Descripción
codigo (llave primaria)	integer	Código de la región.

npub	varchar	Número público.
denom	varchar	Denominación del producto y/o servicios.
ejercito	integer	Viene de nom_ejercitos. Identificador del ejército al que pertenece la región.

Nombre: nom_ejercitos		
Descripción: Esta tabla contiene todos los datos del Ejército.		
Atributo	Tipo	Descripción
codigo (llave primaria)	integer	Código del ejército.
abr	varchar	Abreviatura de la denominación.
denom	varchar	Denominación del producto y/o servicios.

Nombre: nom_nomenclaturas		
Descripción: Esta tabla contiene todos los datos de las nomenclaturas a planificar.		
Atributo	Tipo	Descripción
codigo (llave primaria)	integer	Código de la nomenclatura.
abr	varchar	Abreviatura de la denominación.
denom	varchar	Denominación del producto y/o servicios.

Nombre: nom_operaciones		
Descripción: Esta tabla contiene todos los datos de las operaciones que se le realizan la planificación.		
Atributo	Tipo	Descripción
codigo (llave primaria)	integer	Código de la operación.
abr	varchar	Abreviatura de la denominación.
denom	varchar	Denominación del producto y/o servicios.
dias	integer	Identificador de los días que planificarán para esa operación.

Nombre: nom_especialidades		
Descripción: Esta tabla contiene todos los datos de las especialidades que se le realizan la planificación.		
Atributo	Tipo	Descripción
codigo (llave primaria)	integer	Código de la especialidad.
abr	varchar	Abreviatura de la denominación.
denom	varchar	Denominación del producto y/o servicios.

Nombre: nom_municipios		
Descripción: Esta tabla contiene todos los datos de los municipios que se le realizan la planificación.		
Atributo	Tipo	Descripción
codigo (llave primaria)	integer	Código del municipio.
abr	varchar	Abreviatura de la denominación.
denom	varchar	Denominación del producto y/o servicios.
region	integer	Viene de nom_regiones. Identificador de la región perteneciente al municipio.

Nombre: nom_umed		
Descripción: Esta tabla contiene los nomencladores de unidades de medida.		
Atributo	Tipo	Descripción
codigo (llave primaria)	integer	Código de la unidad de medida.
abr	varchar	Abreviatura de la denominación.
denom	varchar	Denominación del producto y/o servicios.
unitario	integer	Especifica ¿qué unidad de medida al calcularse un producto asociado a ella se debe redondear al valor superior?

Nombre: nom_tropas		
Descripción: Esta tabla contiene todos los datos de las tropas que realizan la planificación.		
Atributo	Tipo	Descripción

codigo (llave primaria)	integer	Código de la tropa.
abr	varchar	Abreviatura de la denominación.
denom	varchar	Denominación del producto y/o servicios.

Nombre: nom_variables		
Descripción: Esta tabla contiene todos los datos de las variables a planificar.		
Atributo	Tipo	Descripción
codigo (llave primaria)	integer	Código de las variables.
descrip	varchar	Es la descripción textual que contiene cada variable.
denom	varchar	Denominación del producto y/o servicios.

3.5 Principios de diseño

Para lograr un diseño del sistema que este acorde con lo que se pide por parte del cliente, se tuvieron en cuenta los siguientes principios de diseño para el desarrollo del mismo:

- Tratamiento de errores
- Mecanismo de diseño
- Seguridad
- Interfaz de usuario
- Ayuda

Antes de comenzar con una breve descripción de cada principio, es importante puntualizar que:

- Para facilitar que el usuario pueda acceder a cada funcionalidad del sistema se hará uso de un menú principal que en el se encontrarán todas las funcionalidades que habrán de la aplicación, y este estará presente en todas las páginas que sean accedidas.
- Las páginas no estarán cargada de mucha información y serán lo mas sencillas posibles.
- Las páginas deberán tener colores claros.

- Se deberán hacer uso de componentes de dhtmlx como son: árboles y tablas para así optimizar la visualización y manejo de la información relacionada al mismo.

3.5.1 Tratamiento de errores.

El correcto tratamiento de los errores en un sistema garantiza la seguridad e integridad de la información que en él se maneja, asegurando así el buen funcionamiento de este. Para ello se tuvieron en cuenta todos los posibles errores que pudieran ocurrir al interactuar el usuario con el sistema, como son: se realizaron las debidas validaciones para en caso de algún descuido del usuario al entrar cualquier dato incorrecto como es el uso de letras, operadores matemáticos (+, -, *, / etc.) y signos especiales (@, %, \$, & etc.) el sistema le informará de su error. Además de esto, se muestran mensajes de confirmación ante acciones que son irreversibles, como es el caso de la eliminación de datos, a la vez que se muestran mensajes para indicar cuándo una acción fue realizada con éxito. De no entrarse por parte del usuario todos los datos necesarios para la ejecución de un proceso o no haber realizado los procesos que se necesitan para la realización de otros, el sistema mostrará un mensaje de error informándole al usuario que no puede continuar y que se pide por parte del sistema para que se ejecute. Ver figura 3.27



Figura 3.27- Mensaje de error durante la acción de entrar los datos para buscar las variables a calcular.

3.5.2 Mecanismo de diseño.

Mecanismos de Diseño, es un artefacto de RUP que agrupa un conjunto de clases del diseño, colaboraciones, e incluso subsistemas del modelo de diseño que llevan a cabo requisitos comunes como persistencia, distribución, seguridad, y funcionamiento.

Permiten la reutilización de la documentación en los modelos, alcanzando así una mayor claridad en estos y propiciando por tanto una mejor comunicación entre los especialistas involucrados en el desarrollo de la aplicación, además de, mantener la homogeneidad en el diseño y la reutilización de soluciones anteriormente probadas.

En este sistema se propone un mecanismo de diseño para manejar la persistencia.

Se realizaron diagramas que muestran la vista estática (diagrama de clases) y diagramas de interacción (diagramas de secuencia) que muestran las distintas colaboraciones que dan solución al problema, estos diagramas se pueden ver en el (Anexo 8).

El diagrama de clases que se muestra a continuación es la propuesta que se propone al sistema para el manejo de acceso a los datos.

métodos consisten en las operaciones básicas que se realizan sobre estas tablas, (INSERT, DELETE, UPDATE). Hereda de la clase abstracta meBase.

TípicaSimple: es una clase que representa a las clases típicas para nomencladores simples. Para la implementación de esta clase se decidió aplicar el patrón de diseño Table Data Gateway. Sus métodos consisten en las operaciones básicas que se realizan sobre estas tablas, (INSERT, DELETE, UPDATE). Hereda de la clase abstracta meSimple.

cClaseconsulta: es una 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, 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.

** Entidades cuya estructura responde al siguiente patrón: idALGO, ALGO, actual. Donde ALGO representa la descripción del atributo principal de los nomencladores clásicos.

meSimple define las operaciones básicas que pudieran realizarse a una entidad (INSERT, DELETE, UPDATE) para los nomencladores simples. Hereda de la clase abstracta meBase.

meBase: clase abstracta, base para el resto de las que implementen funcionalidades para el trabajo con las entidades del sistema a implementar. Implementa las operaciones básicas que pudieran realizarse a una entidad (INSERT, DELETE, UPDATE). Hereda de consBase la operación de CONSULTA.

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 BD a través de un objeto PDO de la librería de PHP. Se concibió aplicando el patrón Singleton el cual garantiza una única instancia para una clase y la creación de un mecanismo global (único) de acceso a dicha instancia.

PDO: es un modelo de acceso a bases de datos para PHP. PDO nos brinda una capa de abstracción para el acceso a bases de datos desde PHP.

3.5.3 Seguridad

Seguridad Informática: Conjunto de métodos y herramientas destinados a proteger los bienes informáticos de una institución. El término seguridad informática está estrechamente relacionado con tres aspectos fundamentales de cualquier sistema de computación:

- Confidencialidad: La información o los activos informáticos son accedidos solo por las personas autorizadas.
- Integridad: Los activos o la información solo pueden ser modificados por las personas autorizadas y de la forma autorizada.
- Disponibilidad: Los activos informáticos son accedidos por las personas autorizadas en el momento requerido. [23]

Para lograr un sistema seguro es necesario tratar de establecer un balance correcto entre estos tres aspectos.

Un sistema de computación tiene tres componentes fundamentales: hardware, software y datos. Los datos son generalmente el activo máspreciado para cualquier institución.

Principios de seguridad informática

Existen algunos principios y estrategias a seguir para mantener adecuadamente la seguridad informática de una institución. Dichos principios son:

- Mínimo privilegio: se deben otorgar los permisos estrictamente necesarios para efectuar las acciones que se requieran, ni más ni menos de lo solicitado.
- Eslabón más débil: la seguridad de un sistema es tan fuerte como su parte más débil. Un atacante primero analiza cual es el punto más débil del sistema y concentra sus esfuerzos en ese lugar.
- Proporcionalidad: las medidas de seguridad deben estar en correspondencia con lo que se protege y con el nivel de riesgo existente. No sería lógico proteger con múltiples recursos un activo informático que no posee valor que la probabilidad de ataque sobre el mismo es muy baja.
- Dinamismo: la seguridad informática no es un producto, es un proceso. No se termina con la implementación de los medios tecnológicos, se requiere permanentemente monitoreo y mantenimiento.

- Participación universal: la gestión de seguridad informática necesita de la participación de todo el personal de una institución. [23]

Control de acceso

Control de acceso es el proceso de conceder permisos a usuarios o grupos de acceder a objetos, tales como ficheros o impresoras en la red, el mismo está basado en tres conceptos fundamentales: identificación, autenticación y autorización. Los controles de accesos son necesarios para proteger la confidencialidad, integridad y disponibilidad de los objetos, y por extensión la información que contienen, pues permiten que los usuarios autorizados accedan solo a los recursos que ellos requieren para realizar sus tareas.[24]

✓ Técnicas de Identificación y Autenticación

Identificación: acción del usuario de presentar su identidad a un sistema, usualmente se usa un identificador de usuario. Establece además que el usuario es responsable de las acciones que lleve a cabo en el sistema. Esto está relacionado con los registros de auditorias que permiten guardar las acciones realizadas dentro del sistema y rastrearlos hasta el usuario autenticado.

Autenticación: verificación de que el usuario que trata de identificarse es válido, usualmente se implementa con una contraseña en el momento de iniciar una sesión. Existen cuatro tipos de técnicas que permiten realizar la autenticación de la identidad del usuario, las cuales pueden ser utilizadas individualmente o combinadas (autenticación de varios factores):

1. Algo que solamente el individuo conoce: por ejemplo una contraseña.
2. Algo que la persona posee: por ejemplo una tarjeta magnética.
3. Algo que el individuo es y que lo identifica unívocamente: por ejemplo las huellas digitales.
4. Algo que solo el individuo es capaz de hacer: por ejemplo los patrones de escritura. [24]

La técnica de autenticación a utilizar en el sistema será el uso de la contraseña, cada usuario del sistema deberá identificarse en el sistema con su nombre y contraseña, los cuales serán verificados y si alguno de estos datos son incorrectos, se denegará dicho acceso. Para proteger que la contraseña de un usuario solo sea de conocimiento del mismo la misma se le realizará un proceso de encriptación, garantizando la integridad de la misma.

✓ Autorización. Modelos de Control de Acceso

La autorización es el procedimiento para determinar si el usuario o proceso previamente identificado y autenticado tiene permitido el acceso a los recursos. Se implementa con uno de los siguientes modelos de control de acceso. [24]

Control de Acceso Obligatorio (MAC)

La autorización para que un sujeto acceda a un objeto depende de los niveles de seguridad que tengan, ya que estos indican que permiso de seguridad tiene el sujeto y el nivel de sensibilidad del objeto. Un individuo puede acceder a documentos clasificados con un nivel igual o inferior a su nivel de seguridad.

Todos los sujetos y objetos del sistema tienen una etiqueta de seguridad que se compone de:

- Una clasificación o nivel de seguridad como un número en un rango, o un conjunto de clasificaciones discretas (Desclasificado, Confidencial, Secreto y Sumamente Secreto)

Control de acceso Discrecional (DAC)

En este modelo un usuario bien identificado, típicamente el creador o propietario del recurso, decide cómo protegerlo estableciendo como compartirlo, mediante controles de acceso impuestos por el sistema.

Control de acceso basado en Roles (RBAC)

En este modelo a los usuarios le son asignados uno o varios roles mientras que los permisos y privilegios se asignan a estos roles. Por tanto, las políticas de control de accesos basado en roles regulan el acceso de los usuarios a la información en términos de sus actividades y funciones de trabajo (roles).

Dada la alta integración entre los roles y las responsabilidades de los usuarios, se pueden seguir los principios del mínimo privilegio y de la separación de responsabilidades. Estos principios son vitales para alcanzar el objetivo de integridad, al requerir que a un usuario no se le otorguen mayores privilegios que los necesarios para efectuar su trabajo. [24]

El modelo de control de acceso que se aplicará al sistema será el MAC, debido a que, el usuario del sistema (Unidad Demandante) solo tendrá acceso a las especialidades que contendrá la tropa a la que él pertenece, es decir, una tropa tiene definida un rango de especialidades por las que puede realizar la planificación y la unidad militar que pertenezca a esa tropa solo puede planificar por las especialidades definidas en dicho rango.

3.5.4 Interfaz de Usuario

Para el diseño de la interfaz de usuario del sistema se deberán tener en cuenta primeramente los requerimientos no funcionales definidos por el usuario, quien es el que va a hacer uso del mismo y quien determina el éxito o fracaso del mismo. Por eso la misma deberá tener una interfaz fácil de usar y que a la vez sea amigable para que pueda ser utilizada sin mucha preparación por el usuario, se utilizará un mismo formato y estilo en cada una de las páginas y los colores utilizados serán claros preferentemente blanco, gris y azul. Cada interfaz deberá llevar una cabecera que estará definida por una imagen que presentará el logotipo de la aplicación, el diseño de las interfaces tendrán una resolución de 800x600px y las mismas contarán con una barra menú horizontal en cada una de ellas que contendrá todas las funcionalidades que brindará el sistema. Se hará uso de una hoja de estilo para guardar la configuración del diseño de todas las páginas del sistema, tanto para los botones como para el tipo y tamaño de letra, para lograr así una uniformidad en todas estas.

3.5.5 Ayuda

La Ayuda siempre estará presente en cada interfaz del sistema ubicada en la barra menú para ser usada por el usuario cada vez que lo crea conveniente, la misma mostrará de forma clara y sencilla como está conformado la página con la cual se está trabajando, lo que permitirá que el usuario siempre este informado en lo que tiene que hacer y como hacerlo, para llevar a cabo el proceso de planificación de la demanda para la lucha armada correctamente.

3.6 Patrones de Diseño

Un patrón de diseño es:

- una solución estándar para un problema común de programación
- una técnica para flexibilizar el código haciéndolo satisfacer ciertos criterios
- un proyecto o estructura de implementación que logra una finalidad determinada
- un lenguaje de programación de alto nivel
- una manera más práctica de describir ciertos aspectos de la organización de un programa
- conexiones entre componentes de programas
- la forma de un diagrama de objeto o de un modelo de objeto.

Podemos 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.

Patrones de creación

Los patrones de creación abstraen la forma en la que se crean los objetos, permitiendo tratar las clases a crear de forma genérica dejando para más tarde la decisión de qué clases crear o cómo crearlas.

Según donde se tome dicha decisión podemos clasificar a los patrones de creación en patrones de creación de clase (la decisión se toma en los constructores de las clases y usan la herencia para determinar la creación de las instancias) y patrones de creación de objeto (se modifica la clase desde el objeto).

Patrones estructurales

Tratan de conseguir que cambios en los requisitos de la aplicación no ocasionen cambios en las relaciones entre los objetos. Lo fundamental son las relaciones de uso entre los objetos, y, éstas están determinadas por las interfaces que soportan los objetos. Estudian como se relacionan los objetos en tiempo de ejecución. Sirven para diseñar las interconexiones entre los objetos.

Patrones de comportamiento

Los patrones de comportamiento estudian las relaciones entre llamadas entre los diferentes objetos, normalmente ligados con la dimensión temporal.

Patrón Factoría: La clase factoría devuelve una instancia de un objeto según los datos que se le pasan como parámetros. Para que la creación centralizada de objetos sea lo más útil y eficaz posible, es de esperar que todos los objetos creados descendan de la misma clase o implementen el mismo interface (es decir, hagan una operación similar pero de distintas formas), así poderlos usar todos de la misma manera y con los mismos métodos (gracias al polimorfismo), sin importarnos que clase concreta se está tratando en cada momento. [25]

Patrón Singleton: Garantiza que una clase sólo tenga una instancia y proporciona un punto de acceso global a ella. El trabajo con este patrón permite:

- Acceso controlado a la única instancia. Puede tener un control estricto sobre cómo y cuando acceden los clientes a la instancia.
- Espacio de nombres reducido. El patrón Singleton es una mejora sobre las variables globales.
- Permite el refinamiento de operaciones y la representación. Se puede crear una subclase de Singleton.
- Permite un número variable de instancias. El patrón hace que sea fácil cambiar de opinión y permitir más de una instancia de la clase Singleton.
- Más flexible que las operaciones de clase (static en C#, Shared en VB .NET). [26]

Patrón Table Data Gateway: Es un patrón de acceso a datos que se encarga de generar todo el SQL para acceder a una sola tabla: select, insert, update y delete. También puede tener otros métodos para interactuar con la base de datos.

3.7 Conclusiones

Con la realización de este capítulo se logró realizar una visualización y especificación de como es que estará conformado nuestro sistema y como se ven reflejados los requisitos funcionales y no funcionales definidos anteriormente por el usuario, en el desarrollo de la misma. Para ello se modelaron los diagramas de clases del análisis y los de diseño Web junto con sus diagramas de interacción, para cada caso de uso, Además, se definió la arquitectura necesaria para la construcción del sistema, se presentaron los modelos físico y lógico de la base de datos y por último fueron expuestos los principios de diseño tales como: Tratamiento de errores, Mecanismo de diseño, Seguridad, Interfaz de usuario y Ayuda.

CONCLUSIONES

Después de realizado un estudio de la estructura y dinámica del proceso de planificación de la demanda para la lucha armada y haberse investigado las herramientas y tecnologías existentes a nivel internacional, además de, realizarse el modelado de dicho proceso y el análisis y diseño del sistema a proponer, se puede arribar a la conclusión de que se logró un diseño del proceso de planificación de la demanda para la lucha armada según lo establecido en la “Metodología para el proceso de formulación, presentación y satisfacción de la demanda para la lucha armada”. Traduciendo los requisitos en una especificación que describe como se deberá implementar el sistema. Cumpliéndose entonces la informatización del proceso de planificación de la demanda.

RECOMENDACIONES

1. Integrar en un futuro al “Sistema de Planificación Material y Financiera”.
2. Integrar las interfaces con EXT.JS.
3. Proponer realizar el manejo de la seguridad del sistema haciendo uso del mecanismo de seguridad utilizado en la “Unidad de compatibilización, integración y desarrollo del software para la defensa”.
4. Implementar el manejo de la auditoría (Registrar la confirmación de cada operación efectuada por el usuario que afecte los registros contables) en el sistema.

REFERENCIAS BIBLIOGRÁFICAS

1. JLFAR, *Metodología para el proceso de formulación, presentación y satisfacción de las demandas para la lucha armada*. 2003.
2. Monmany, J., *Aplicaciones web: Rentabilizar Internet*. 2007: p. 1.
3. Corsi, M., *Beneficios De Las Aplicaciones Basadas En Web Y El Anuncio De Microsoft De La Era "En Vivo"*. 2005.
4. *Lenguajes de Programación*. [cited 2008 6 febrero]; Available from: <http://jorgesaavedra.wordpress.com/2007/05/05/lenguajes-de-programacion/>.
5. Valdés, D.P. *Los diferentes lenguajes de programación para la web*. 2007 [cited 2008 6 febrero]; Available from: <http://www.maestrosdelweb.com/principiantes/los-diferentes-lenguajes-de-programacion-para-la-web/>.
6. Alvarez, M.A. *Qué es Javascript*. [cited 2008 9 febrero]; Available from: <http://www.desarrolloweb.com/articulos/25.php>.
7. Tena, J.I.L.d., ed. *Aplicaciones JavaScript*. ANAYA MULTIMEDIA (GRUPO ANAYA, S.A) ed., ed. 15.28027. 2000: Madrid. 532.
8. Sturm, J., *Desarrollo de soluciones XML*.
9. Alvarez, M.A. *Objetivos y usos del XML*. [cited 2008 13 febrero]; Available from: <http://www.desarrolloweb.com/articulos/460.php>.
10. *Introducción al lenguaje HTML*. 2000.
11. Mellado, J. *Que es AJAX*. 2006 [cited 2008 18 febrero]; Available from: <http://www.ajaxhispano.com/que-es-ajax.html>.
12. Mellado, J. *Diez razones para usar AJAX*. 2006 [cited 2008 18 febrero]; Available from: <http://www.ajaxhispano.com/diez-razones-para-usar-AJAX.html>.
13. *Una Introducción a APACHE*. 2006: Madrid.
14. González, C.D. *¿Qué es PostgreSQL?* mayo 2008 [cited 2008 22 febrero]; Available from: <http://www.usabilidadweb.com.ar/postgre.php>.
15. Espinosa, H., *PostgreSQL una alternativa de DBMS Open Source*. 2005.
16. *Introducción a la Ingeniería de Software*, in *Ingeniería de Software*. 2008.
17. Jacobson, I.B., G. y Rumbaugh, J., *El Lenguaje Unificado de Modelado. Manual de Referencia*. p. 528.
18. *Comparación de Herramientas de modelado UML: Enterprise Architect y Rational Rose*. 2005 [cited 2008 25 febrero]; Available from: <http://www.apexnet.com.ar/index.php/news/main/38/event=view>.
19. González, Y.M., *Simulación Generas de Sistemas* 2004: Ciudad de La Habana. p. 171.
20. *Fase de Inicio. Modelo del Negocio*, in *Ingeniería de Software*. 2008.
21. Ricote, A.a.M.F., *SGIF: Sistema de Gestión de la Información para la facultad*. 2006, Instituto Superior Politécnico "José Antonio Echeverría".
22. Technology, J. *Aplicaciones basadas en Web*. 2007 [cited 2008 21 abril]; Available from: <http://www.janium.com/page2/page1/page6/page7/page7.html>.
23. *Introducción a la Seguridad Informática*, in *Seguridad Informática*. 2008.

24. *Control de acceso. Identificación y autenticación*, in *Seguridad Informática*. 2008.
25. *Diseño de software con patrones (parte 2)*. 2001 [cited 2008 24 abril]; Available from: http://www.javahispano.org/contenidos/es/patrones_de_software_parte_2/.
26. *El Patrón Singleton*. [cited 2008 24 abril]; Available from: http://www.microsoft.com/spanish/msdn/comunidad/mtj.net/voices/MTJ_4081/default.aspx.

BIBLIOGRAFÍA

- [1] Alvarez, A. Miguel, *¿Qué es PHP?* [cited 2008 6 febrero]; Available from: <http://www.desarrolloweb.com/articulos/392.php>
- [2] Larman. Craig, *UML y Patrones. Introducción al análisis y diseño orientado a objetos*. México; PRENTICE HALL; 1999; p. 507.
- [3] González, Y.M., *Simulación Generas de Sistemas;* ITM José Martí” 2004: Ciudad de La Habana. p. 171.
- [4] Kennedy.Bill y M.Chuck, *HTML. La Guía Completa*. México; Litográfica Ingramex; 1999; p. 531.
- [5] Schmuller. Joseph. *UML en 24 en Horas;* México; PEARSON EDUCACION; 2000; p. 423.
- [6] Sturm. Jack. *Desarrollo de soluciones XML;* p. 393.
- [7] *Características de PHP.* 2007 [cited 2008 6 febrero]; Available from: <http://www.linuxcentro.net/linux/staticpages/index.php?page=CaracteristicasPHP>.
- [8] Entorno Virtual de Aprendizaje; 2008; <http://teleformacion.uci.cu/>
- [9] *¿Qué es un Sistema Gestor de Base de Datos o SGBD?;*
<http://www.cavsi.com/preguntasrespuestas/que-es-un-sistema-gestor-de-bases-de-datos-o-sgbd>
- [10] JLFAR, *Metodología para el proceso de formulación, presentación y satisfacción de las demandas para la lucha armada.* 2003.

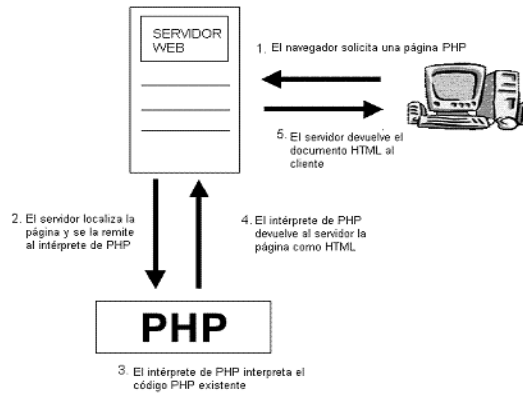
GLOSARIO DE TÉRMINOS

- FAR: Fuerzas Armadas Revolucionarias.
- MTT: Milicias de Tropas Regulares.
- MININT: Ministerio del Interior.
- Lineamientos generales: Indicaciones generales a tenerse en cuenta para realizar el proceso de la demanda para la lucha armada.
- Lineamientos particulares: Indicaciones particulares a tenerse en cuenta para realizar el proceso de la demanda.
- Operación: Se refiere a las operaciones por las que las unidades militares debe planificar la demanda para la lucha armada para en caso de situaciones excepcionales, (estas operaciones son: Etapa Inicial de la Crisis, Segunda operación, Primera operación.
- Nomenclaturas: Rige el tipo de satisfacción que se le puede dar a los productos que se planifican.
- Normas: Es por donde se van a regir las Unidades Militares para realizar el cálculo de la demanda.
- MINFAR: Ministerio de las Fueras Armadas Revolucionarias.
- Proceso de demanda: Es el proceso que se realiza en tiempo de paz, donde las unidades militares demandan (solicitan) los productos o servicios que necesitan para en caso de ocurrir alguna situación excepcional.
- HTTP: *HyperText Transfer Protocol*. Protocolo de Transferencia de Hipertextos. Modo de comunicación para solicitar páginas Web.
- MySQL: Es un sistema de administración de bases de datos relacional (RDBMS). Es la base de datos de código abierto más popular del mundo. Su ingeniosa arquitectura lo hace extremadamente rápido y fácil de personalizar.
- Oracle: Es un sistema gestor de base de datos robusto, tiene muchas características que garantizan la seguridad e integridad de los datos; que las transacciones se ejecuten de forma correcta, sin causar inconsistencias; ayuda a administrar y almacenar grandes volúmenes de datos; estabilidad, escalabilidad y es multiplataforma.
- Microsoft SQL Server: Constituye la alternativa de Microsoft a otros potentes sistemas gestores de bases de datos como son Oracle, PostgreSQL o MySQL.

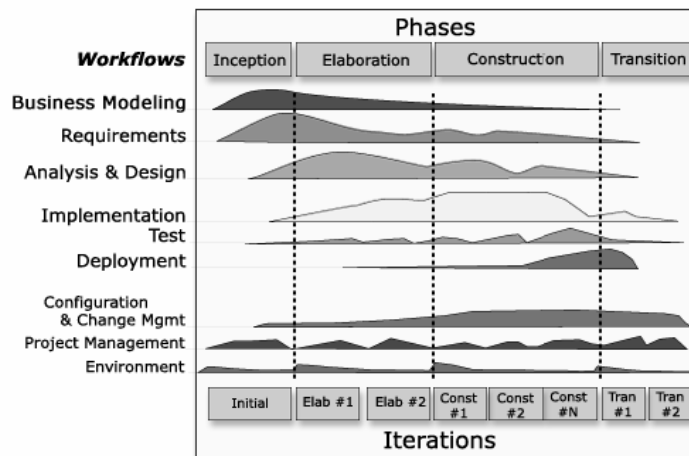
- Firebird: Es una base de datos relacional multiplataforma. Firebird ofrece una concurrencia excelente, alto rendimiento y un poderoso lenguaje de procedimientos almacenados y disparadores.
- Perl: Es un lenguaje de programación interpretado muy utilizado para construir aplicaciones CGI para el web. Perl es un acrónimo de Practical Extracting and Reporting Language, que viene a indicar que se trata de un lenguaje de programación muy práctico para extraer información de archivos de texto y generar informes a partir del contenido de los ficheros. Es un lenguaje libre de uso.
- Script: Es un guión o conjunto de instrucciones. Permiten la automatización de tareas creando pequeñas utilidades. Es muy utilizado para la administración de sistemas UNIX. Son ejecutados por un intérprete de línea de órdenes y usualmente son archivos de texto.
- Java: Es un lenguaje de programación orientado a objetos, independiente de la plataforma, con Java podemos programar páginas web dinámicas, con accesos a bases de datos, utilizando XML, con cualquier tipo de conexión de red entre cualquier sistema.
- CSS: Acrónimo de Cascading Style Sheets, lenguaje de programación que describe el estilo gráfico de un documento web (colores, tipografías y composición).
- DOM: (Document Object Model o Modelo de objetos en documentos). Es una interface independiente de la plataforma y del lenguaje que permite que los programas y scripts tengan acceso dinámicamente y actualicen el contenido, la estructura y estilo de los documentos.
- Plug-in: Programa que se agrega al Navegador para manejar en éste cierto tipo de archivos.
- ASP: Active Server Pages. Una especificación que permite crear dinámicamente páginas Web mediante HTML, scripts, y componentes de servidor ActiveX reutilizables.
- ASP.Net: Es un framework para aplicaciones web desarrollado y comercializado por Microsoft. Es usado para construir sitios web dinámicos, aplicaciones web y servicios web XML.
- JSP: (Java Server Pages) Páginas de Servidor Java. Es una tecnología orientada a crear páginas web con programación en Java.
- Logs: Archivo creado por un servidor que contiene toda la información relativa al acceso a un sitio.
- RPN: Notación polaca inversa (postfija).

ANEXOS

Anexo 1: Proceso de comunicación petición/respuesta de una pagina PHP por parte de un cliente a un servidor de HTTP como APACHE.



Anexo 2: RUP en dos dimensiones



Anexo 3: Ejemplo de convertir una cadena infija a postfija.

Cadena infija $(a + b) * c ^ d$

Cadena polaca

Cadena operadores

Prioridad

NPA 0

- se encuentra un "("

Cadena infija a + b) * c ^ d

Cadena polaca

Cadena operadores

Prioridad

NPA 1

- se encuentra un "a"

Cadena infija + b) * c ^ d

Cadena polaca a

Cadena operadores

Prioridad

NPA 1

- se encuentra un "+" para calcular la prioridad se utiliza la formula $P = P_0 + NPA * (\text{cantidad de operadores})$

Cadena infija b) * c ^ d

Cadena polaca a

Cadena operadores +

Prioridad 15

NPA 1

- se encuentra un "b"

Cadena infija) * c ^ d

Cadena polaca a b

Cadena operadores +

Prioridad 15

NPA 1

- se encuentra un ")"

Cadena infija * c ^ d

Cadena polaca a b

Cadena operadores +

Prioridad 15

NPA 0

- se encuentra un “*”

Cadena infija $c \wedge d$

Cadena polaca $a b$

Cadena operadores $+ *$

Prioridad 15 7

NPA 0

En este caso al ser la prioridad del operador “*” es menor que la del “+”, entonces el operador “+” pasa a la cadena polaca.

Cadena infija $c \wedge d$

Cadena polaca $a b +$

Cadena operadores $*$

Prioridad 7

NPA 0

- se encuentra un “c”

Cadena infija $\wedge d$

Cadena polaca $a b + c$

Cadena operadores $*$

Prioridad 7

NPA 0

- se encuentra un “^”

Cadena infija d

Cadena polaca $a b + c$

Cadena operadores $* \wedge$

Prioridad 7 8

NPA 0

- se encuentra un “d”

Cadena infija

Cadena polaca $a b + c d$

Cadena operadores $* \wedge$

Prioridad 7 8

NPA 0

Como en la cadena infija ya se analizaron todos los elementos que la componían y todavía quedan elementos en la cadena de operadores, entonces de derecha a izquierda se van pasando a la cadena polaca.

Cadena infija

Cadena polaca a b + c d ^ *

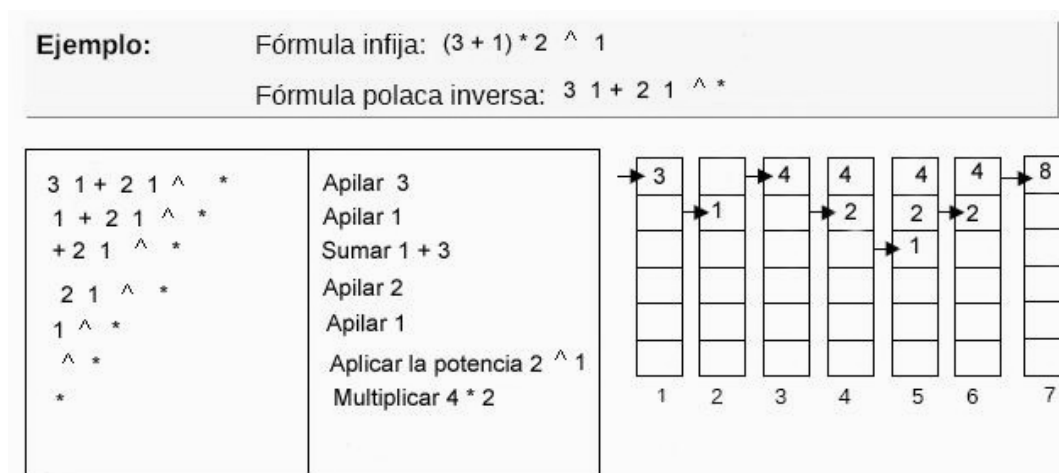
Cadena operadores

Prioridad

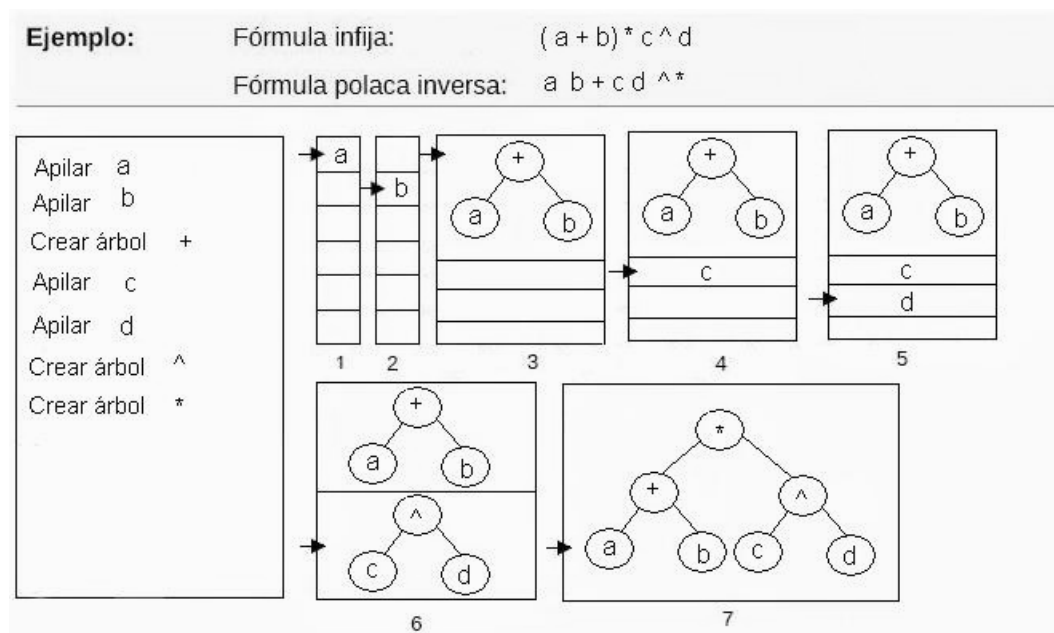
NPA 0

Anexo 4 Ejemplo de evaluación de una cadena postfija utilizando Pilas. Para la misma a la fórmula $(a + b) * c ^ d$ se le van a pasar valores ficticios para una mejor comprensión y claridad de cómo es que se realiza la misma.

a = 3, b = 1, c = 2, d = 1



Anexo 5: Representación de una cadena polaca utilizando el TDA Árbol.



Anexo 6: Informe de planificación de la demanda para la lucha armada (MD1).



Diagrama de secuencia CU Planificar Demanda (Sección Eliminar Cantidades)



Diagrama de secuencia CU Planificar Demanda (Sección Eliminar Todo)

Anexo8

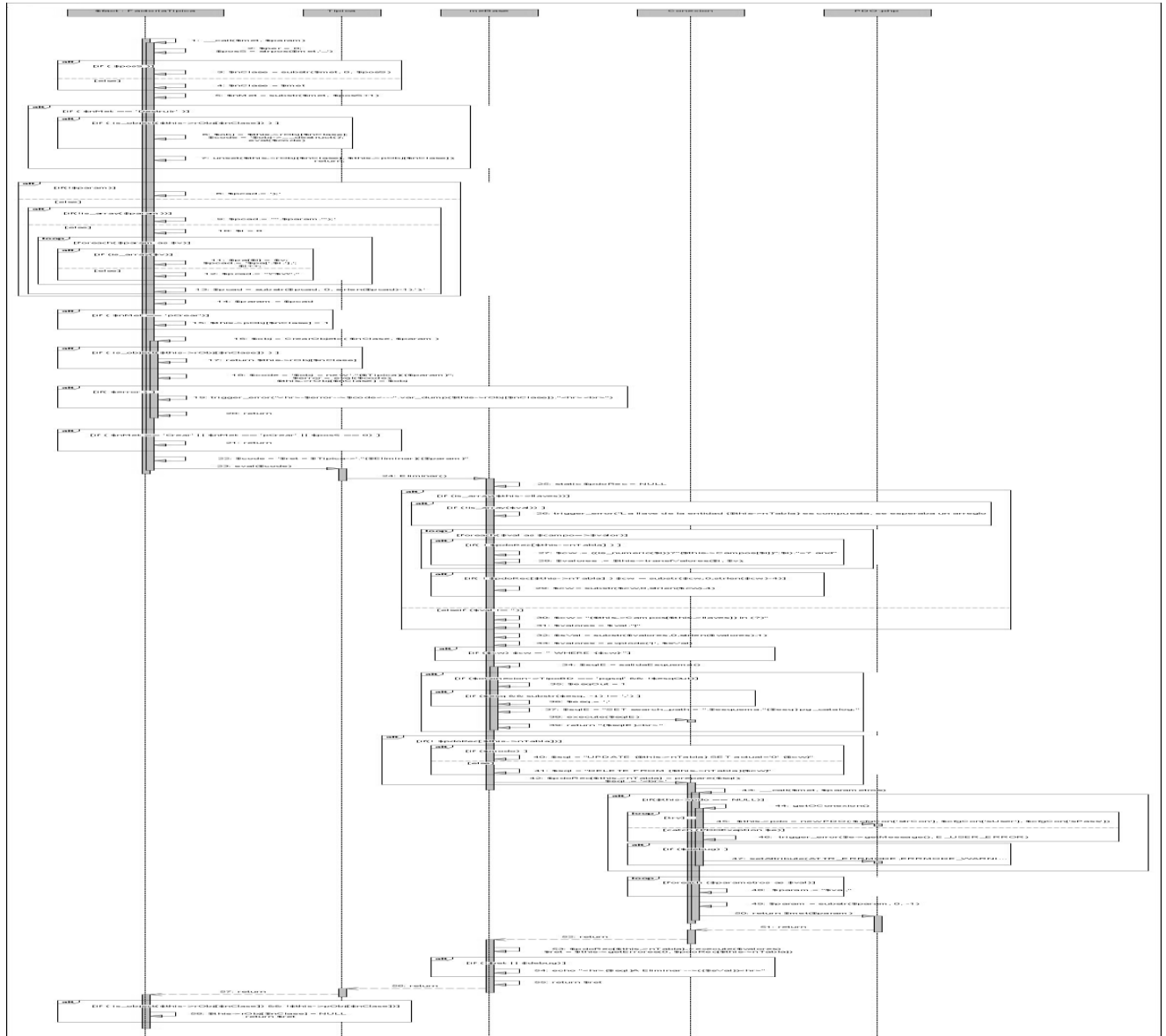


Diagrama de interacción Eliminar para nomencladores que no son simples.

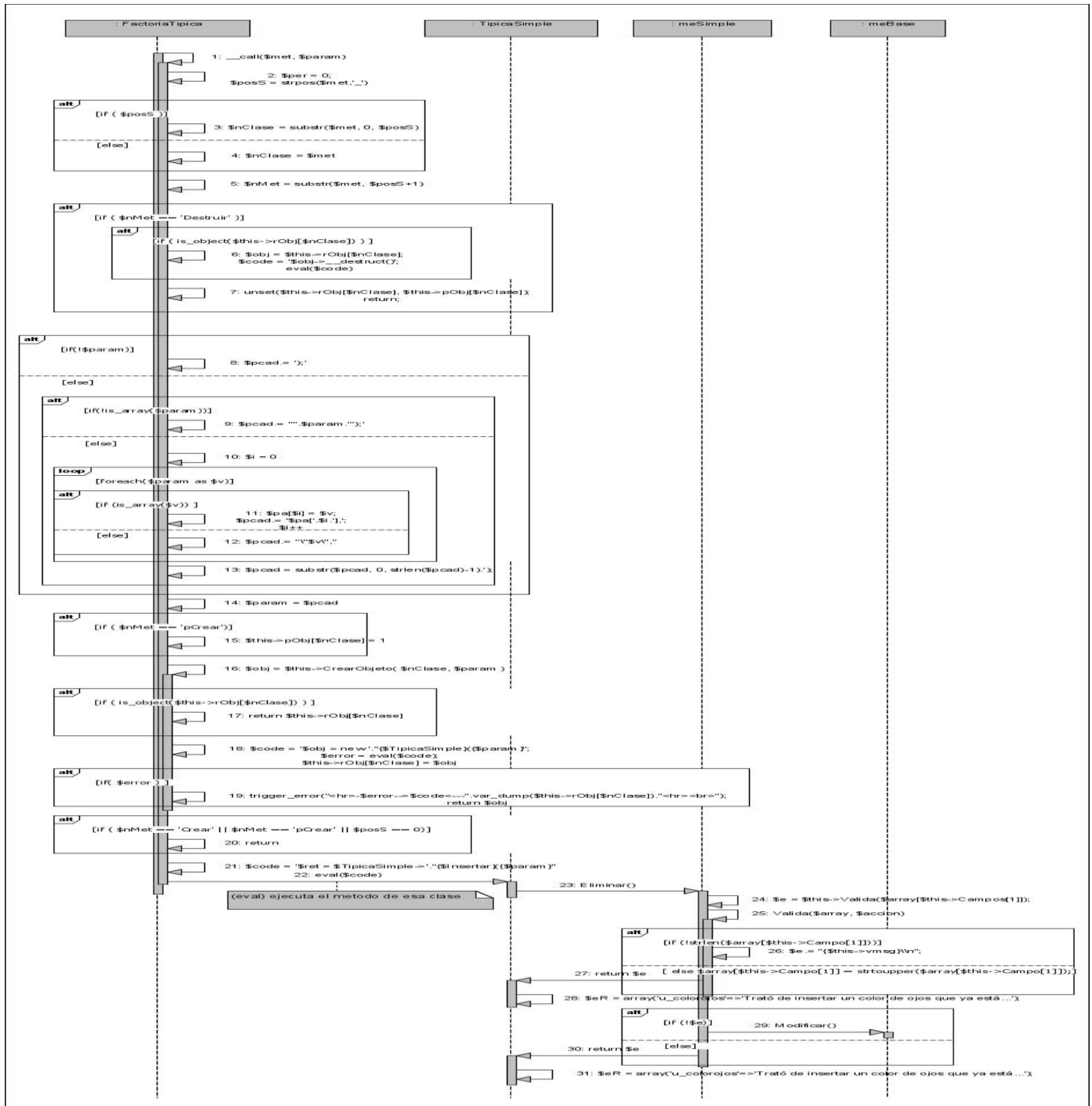


Diagrama de interacción Eliminar para nomencladores simples.

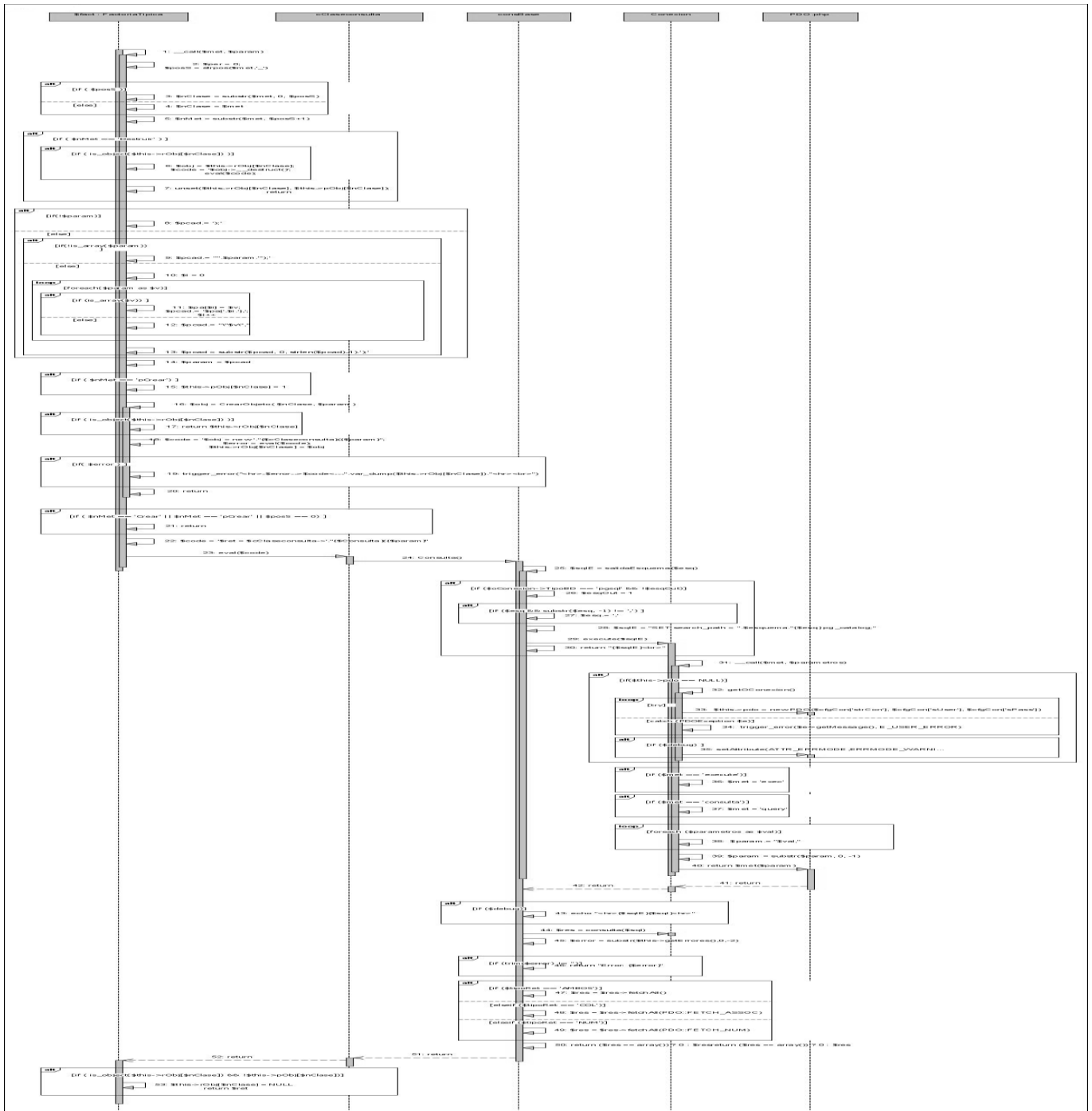


Diagrama de interacción Consulta.