

Universidad de las Ciencias Informáticas

Facultad 1



Título: Modelación, diseño e implementación del Módulo de Mando y Dirección de la especialidad de Lucha Radio Electrónica para el Sistema de Mando y Estado Mayor.

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

Autor(es): Cadete Yaribey Rodríguez Pardo

Cadete Jorge Luis Pavón Acosta

Tutor(es): Ing. 1er Tte. Dobranchy Jiménez Quezada

Ciudad de La Habana, 2010.

Año del 52 de la Revolución

DECLARACIÓN DE AUTORÍA

Declaramos ser autores de la presente tesis y reconocemos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

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

Yaribey Rodríguez Pardo

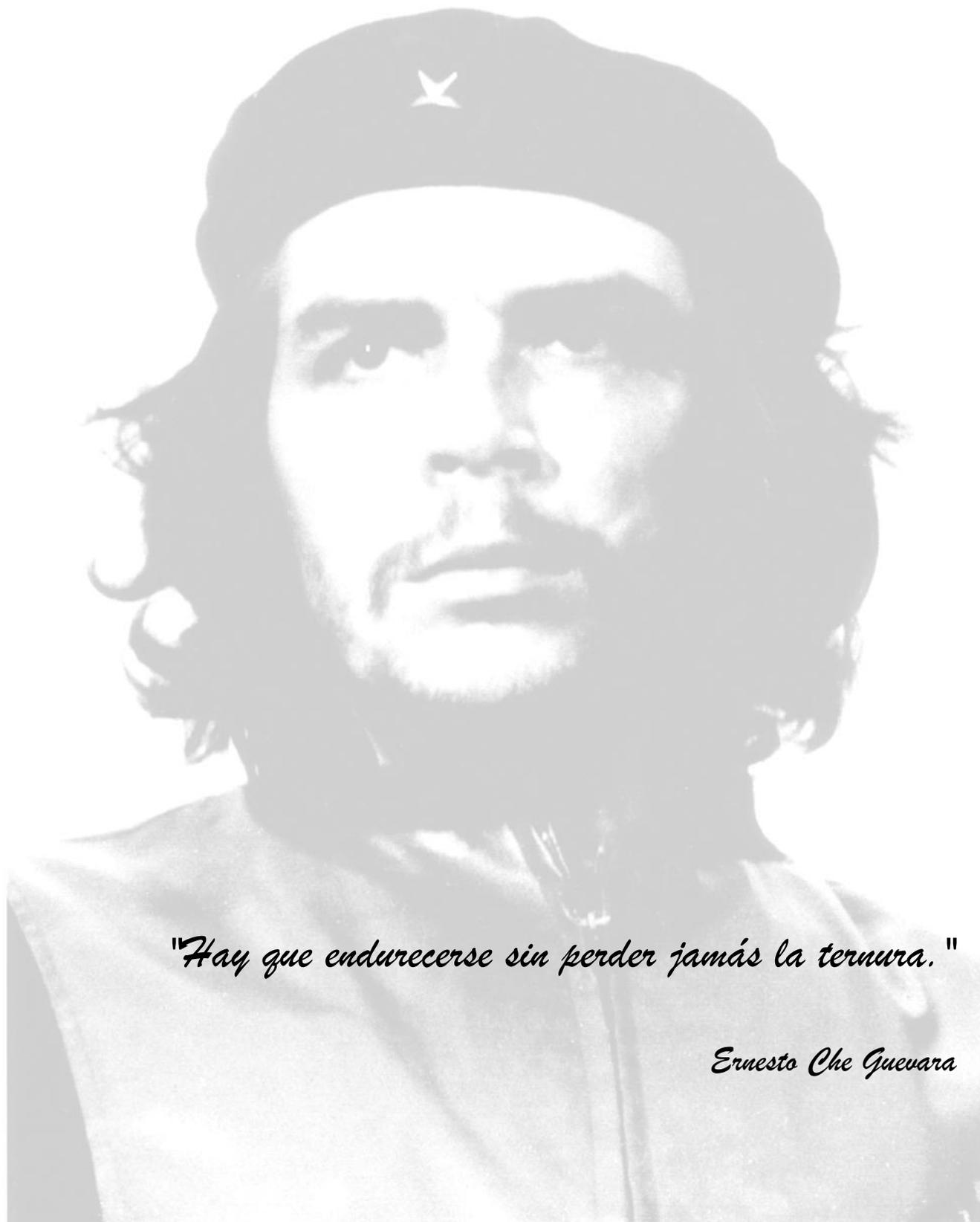
Autor

Jorge Luis Pavón Acosta

Autor

Tte. Dobranchy Jiménez Quezada

Tutor



"Hay que endurecerse sin perder jamás la ternura."

Ernesto Che Guevara

DATOS DE CONTACTO

1er Tte. Dobranchy Jiménez Quezada, Ingeniero en Informática graduado en el 2004 en el Instituto Técnico Militar José Martí Orden Antonio Maceo, se desempeña como Jefe de Centro de Apoyo a la Toma de Decisiones en el Centro de Compatibilización, Integración y Desarrollo de Productos para la Defensa (UCID).

AGRADECIMIENTOS

De Yary:

Un abrazo bien fuerte por apoyarme en lo que me hiciera falta:

Dulce María, mi primera maestra en preescolar, a Dulce María Cuevas, mi primera maestra de matemáticas que aunque no esté entre nosotros, siempre vislumbró un buen futuro; la maestra Mary, muy peleona, pero excelente en sus clases; a Iris y Berriel, mis maestros de sexto grado; a mi profe Noel de Español literatura y Noelvía de Matemáticas en la secundaria; a Miriam Amil la flaca de Matemáticas en la EMCC, la profe de Español Osmiriam, la capitán Yolexis y Yurielkis por enseñarme y acompañarme siempre; a Yene, que ha sido quien más me ha aguantado, Adita que casi se vuelve loca y fue sólo dos cursos; a toda la gente de los apartamentos; Migdelis, Iyatne, Sahilyn, Lianne, Tatiana y las otras, me cambiaron de apartamento muchas veces; a Ariel por caerme arriba para que estudiara y yo nunca hacerle caso. A mi gente de la k̄isomba, a todos. A mi tutor, a Laira, Ángel, Liván, Gersy, a los miembros del tribunal, a Oraldo y Javier que tanto los molestaba. Adrian que siempre me estaba cayendo arriba para revisarme la tesis o ayudarme con cualquier cosa y estuvo conmigo en buenos y malos momentos, a Lídice y Dariel, Arasay, Yorker y Andrés con los que siempre estaba conferenciando, a mi compañero de tesis por regañarme poco, y a todos los que de una forma u otra tuvieron que ver en mi formación. A todos gracias.

De Jorgito:

Para los que siempre me han brindado su ayuda:

Me gustaría agradecerle profundamente a mi familia por su apoyo incondicional. A todos mis profesores que he tenido durante todos estos años, que son los que han contribuido a formar la base de conocimientos que hoy poseo. A mi tutor por ayudarme en todo lo que hiciera falta. A Yorker y Andrés por contestar todas las dudas. A Liván, al que siempre estaba molestando, lo mismo con Ángel que también aportó conocimientos y apoyo. Laira no puede quedarse sin mencionar, que siempre me prestaba lo que me hiciera falta y por la que siento un profundo respeto.

Gracias por todo.

DEDICATORIA

De Yary:

A mi Nanín que desde que era pequeñita siempre me estaba empujando a que me esforzara y diera lo mejor de mí, y se ha ilusionado tanto con este fin de estudios.

A mi Guela, mi angelito de la guarda.

A tía Mary y mi prima Ney que son mis hermanas también.

A mis Titos, que son dos, el de Camagüey y el de Ciego, que sigan tan lindos.

A Maiquel, mi primo más inocente.

A mi madrina y el chino que han sido parte de mi vida también.

A mi gente del barrio que me tienen como una princesita.

A mi tía Elvira, y toda mi gran familia que sean de sangre o no, siempre me ayudan.

Un besote grande a todos...!!!

De Jorgito:

Le dedico este trabajo de diploma especialmente a mi mamá que siempre estuvo a mi lado apoyándome e intentando que diera siempre un paso al frente a pesar de todas las dificultades.

A mi familia en general que también estuvo siempre conmigo.

A mis amigos de toda la vida los presentes y los que están un tanto lejos este día pero que sé están pendientes de mis resultados.

A la Revolución Cubana por darme la oportunidad de estudiar en una escuela como esta, especialmente a nuestro querido Comandante en Jefe Fidel, guía de mis pasos.

Un abrazo...

RESUMEN

A pesar de los inconvenientes que se presentan a diario debido al escaso desarrollo que se presenta en el país, la sociedad se va superando con pasos agigantados dentro del campo de la informatización, inmersos en un proceso de utilización de las tecnologías de la información y las comunicaciones (TIC). Siendo este un punto decisivo para lograr el desarrollo tecnológico en la vida diaria, dígase las empresas, la economía o sociedad cubana, buscándose lograr mayor eficacia y eficiencia a la hora de satisfacer necesidades de información y conocimiento en todas las personas y esferas de la sociedad.

En todo este proceso de informatización y desarrollo también se encuentra sumergido el Ministerio de las Fuerzas Armadas Revolucionarias (MINFAR). Debido a que se procesa un gran volumen de información en este ministerio, es requerido el manejo y utilización de varias herramientas que permitan un tratamiento eficaz y seguro. Una de las necesidades actuales que se presenta, es la de automatizar los principales procesos de mando y dirección de la especialidad de Lucha Radio Electrónica (LRE) durante la preparación y realización de acciones combativas.

Este trabajo abarca el diseño e implementación de una aplicación informática que permita realizar los principales procesos de mando y dirección de la especialidad de Lucha Radio Electrónica (LRE) del MINFAR a través de la preparación y realización de acciones combativas.

PALABRAS CLAVE:

TIC, MINFAR, INFORMACIÓN, LUCHA RADIO ELECTRÓNICA, LRE, PROCESOS DE MANDO Y DIRECCIÓN, ACCIONES COMBATIVAS.

TABLA DE CONTENIDOS

INTRODUCCIÓN	5
CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA. ESTADO DEL ARTE.	9
1.1 INTRODUCCIÓN	9
1.2 TRABAJO SIMILARES	9
1.2.1 ÁMBITO INTERNACIONAL	10
1.2.2 ÁMBITO NACIONAL	11
1.3 ARQUITECTURA	12
1.4 TECNOLOGÍAS A UTILIZAR	15
1.4.1 LENGUAJES	15
1.4.1.1 LENGUAJES DEL LADO DEL SERVIDOR	16
1.4.1.2 LENGUAJES DEL LADO DEL CLIENTE	18
1.5 GESTOR DE BASE DE DATOS	20
1.6 METODOLOGÍA DE DESARROLLO	21
1.6.1 LENGUAJE DE MODELADO UML	22
1.6.2 NOTACIÓN DE MODELADOS DE PROCESOS DEL NEGOCIO (BPMN) [19]	23
1.7 HERRAMIENTAS DE PROGRAMACIÓN	23
1.8 HERRAMIENTAS BD	24
1.9 HERRAMIENTAS DE MODELADO	25
1.10 FRAMEWORK	26
1.11 PROPUESTA DE SOLUCIÓN	30
1.12 CONCLUSIONES	31
CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA	32
2.1 INTRODUCCIÓN	32
2.2 DESCRIPCIÓN DE LOS PROCESOS	32
2.3 FLUJO ACTUAL DE LOS PROCESOS	33
2.4 DESCRIPCIÓN DE LOS SISTEMAS VINCULADOS	33
2.5 DESCRIPCIÓN GENERAL DEL SISTEMA	34
2.6 ANÁLISIS COMPARATIVO CON OTRAS SOLUCIONES	34
2.7 MODELO DE LOS PROCESOS DE NEGOCIO	35
2.8 MAPA DE LOS PROCESOS	35
2.9 DESCRIPCIÓN DE LOS PROCESOS DE NEGOCIO	36
2.10 IDENTIFICACIÓN DE CONCEPTOS EN EL DOMINIO DEL PROBLEMA.	37
2.11 ESPECIFICACIÓN DE LOS REQUISITOS DE SOFTWARE.	38
2.12 REQUERIMIENTOS NO FUNCIONALES.	39
2.13 DESCRIPCIÓN DE REQUISITOS FUNCIONALES.	41
2.14 CONCLUSIONES	42

CAPÍTULO 3. DISEÑO DEL SISTEMA.	43
3.1 INTRODUCCIÓN	43
3.2 DISEÑO	43
3.3 DIAGRAMA DE CLASES	43
3.4 DIAGRAMA DE INTERACCIÓN (SECUENCIA)	44
3.5 DESCRIPCIÓN DE LAS CLASES	45
3.6 DIAGRAMA ENTIDAD RELACIÓN	47
3.7 DESCRIPCIÓN DE LAS TABLAS	48
3.8 MECANISMOS DE DISEÑO [25]	48
3.9 SEGURIDAD	51
3.10 INTERFAZ [26]	51
3.11 CONCEPCIÓN DE LA AYUDA	53
3.12 CONCLUSIONES	54
CAPÍTULO 4. IMPLEMENTACIÓN Y PRUEBA.	55
4.1 INTRODUCCIÓN	55
4.2 DIAGRAMAS DE COMPONENTES	55
4.3 MATRIZ DE INTEGRACIÓN DE COMPONENTES INTERNA.	56
4.4 MATRIZ DE INTEGRACIÓN DE COMPONENTES EXTERNA.	56
4.5 MODELO DE PRUEBA	57
4.6 PRUEBAS DE CAJA NEGRA	58
JUEGOS DE DATOS A PROBAR	61
4.7 CONCLUSIONES	62
CONCLUSIONES	63
RECOMENDACIONES	64
REFERENCIAS BIBLIOGRÁFICAS	65
ANEXO A	¡ERROR! MARCADOR NO DEFINIDO.
ANEXO B	¡ERROR! MARCADOR NO DEFINIDO.
ANEXO C	¡ERROR! MARCADOR NO DEFINIDO.
ANEXO D	¡ERROR! MARCADOR NO DEFINIDO.
ANEXO E	¡ERROR! MARCADOR NO DEFINIDO.
ANEXO F	¡ERROR! MARCADOR NO DEFINIDO.
GLOSARIO DE TÉRMINOS	¡ERROR! MARCADOR NO DEFINIDO.

Introducción

Actualmente en el mundo existen diversos puntos de vista respecto a la forma en la que se le debe dar tratamiento a la información. En los últimos años ha habido un incremento notable en el desarrollo de la informática y se han dado a conocer diversas formas de darle un tratamiento eficaz, ya sea a la hora de guardarla, protegerla o apoyarse en ella para escoger alguna vía que sirva de hilo conductor del trabajo que se realiza. Un desarrollo semejante en este campo permite idear caminos nuevos con la inspiración de utilizar los datos con los que se cuentan con el objetivo de mejorar los procesos que se llevan diariamente a cabo en el apoyo a la toma de decisiones. Se trabaja para idear maneras en las que la información pueda ser consultada, recuperada y analizada en tiempo real para darle uso de manera que, centrados en ella, se puedan manipular los conocimientos factiblemente acordes a las necesidades que se tengan.

Las personas que actúan o deciden racionalmente están intentando alcanzar alguna meta que no se puede lograr sin acción. Necesitan comprender en forma clara los cursos alternativos mediante los cuales se puede lograr una meta de acuerdo con las circunstancias y limitaciones existentes. Se necesita también la información y la capacidad para analizar y evaluar las alternativas de acuerdo con la meta deseada. Por último, necesitan tener el deseo de llegar a la mejor solución mediante la selección de la alternativa que satisfaga de un modo más efectivo el logro de la meta. [1]

En ocasiones los ingenieros consideran la toma de decisiones como su trabajo principal ya que tienen que seleccionar constantemente qué se hace, quien lo hace y cuándo, dónde e incluso como se hará. Sin embargo, estos caminos a escoger son sólo un paso de la planeación ya que forma la parte esencial de los procesos que se siguen para elaboración de los objetivos o metas trazadas a seguir. Rara vez se puede juzgar sólo un curso de acción, porque prácticamente cada alternativa tiene que estar engranada con otros planes. [1]

El proceso que conduce a la toma de decisión: [1]

1. Elaboración de premisas

2. Identificación de alternativas

3. Evaluación de las alternativas, partiendo de metas que se desean alcanzar

4. Selección de una alternativa, es decir, tomar una decisión

Dado el volumen de datos a procesar, el tiempo que transcurre, la posibilidad de pérdidas o errores que puede acarrear que estos sean transferidos de un lugar a otro, las decisiones particulares que por cuestiones obvias se tienen que tomar en el MINFAR, y que además actualmente no se cuenta con un módulo de mando y dirección especializado en el apoyo la toma de decisiones en actividades combativas de la Lucha Radio electrónica (LRE), se tiene la necesidad de un módulo que facilite el trabajo de los especialistas en este medio. Actualmente existen algunas aplicaciones aisladas que apoyan esta toma de decisiones en el módulo de mando y dirección de la LRE, pero se dificulta el trabajo con ellas ya que se necesita utilizarlas todas para resolver un mismo problema y esto conlleva a un desgaste de recursos y tiempo innecesario. Por lo antes expuesto el **problema a resolver** queda planteado de la siguiente manera: ¿Cómo facilitar los principales procesos de mando y dirección de la especialidad de Lucha Radio Electrónica durante la preparación y realización de acciones combativas? Para insertarse en el centro del conflicto, de manera que podamos darle solución a este planteamiento se tiene como **objeto de estudio**: los procesos de mando y dirección de la especialidad de Lucha Radio Electrónica y se abarcará como **campo de acción**: los procesos de mando y dirección de la especialidad de Lucha Radio Electrónica durante la preparación y realización de acciones combativas.

Como **idea a defender** se plantea que la elaboración del módulo de mando y dirección de LRE facilitará el trabajo de los especialistas y jefes de la especialidad durante la realización y preparación de acciones combativas.

Como **objetivo general** se propone: Desarrollar el Módulo de Mando y Dirección para la especialidad de Lucha Radio Electrónica de forma que el especialista cuente con herramientas que le faciliten la realización de su trabajo durante la preparación y la realización de acciones combativas.

De acuerdo con esta propuesta se derivan los siguientes objetivos específicos:

- Elaborar la fundamentación teórica.
- Realizar el estado del arte.
- Realizar un estudio de las diferentes tecnologías y arquitectura de desarrollo así como de la base reglamentaria y los documentos rectores para la elaboración del sistema y conformación del marco conceptual.
- Definir los procesos de negocio así como su modelación.
- Definir los requisitos funcionales y no funcionales del sistema.
- Realizar el de análisis y diseño.
- Implementar el sistema

A continuación se da una breve explicación de la estructuración del contenido de la presente investigación:

El Capítulo I titulado “Fundamentación teórica. Estado del arte” ofrece los conceptos básicos asociados al dominio del problema. Además, brinda el estado del arte en cuanto a los antecedentes históricos del sistema, así como las técnicas, tecnologías, metodologías y *software* usados en la actualidad o en las que se apoya para la solución del problema que se enfrenta.

El Capítulo II denominado “Características del sistema”: se plantea una valoración crítica del diseño propuesto por el analista, un análisis de posibles implementaciones de componentes o módulos ya existentes que puedan ser rehusados y las estrategias de integración, una descripción de los algoritmos no triviales a implementar. Análisis de complejidad de los mismos y selección de las estructuras de datos apropiadas para la implementación de estos algoritmos, finalizando con la descripción de las nuevas clases u operaciones necesarias.

En el Capítulo III “Resultados y discusión”: se muestran los resultados en un sistema y las pruebas realizadas para validar dicha solución y se brindan una serie de recomendaciones a tener en cuenta en futuras iteraciones.

Capítulo 1: Fundamentación Teórica. Estado del Arte.

1.1 Introducción

En este capítulo se proporciona una idea general de las características y funcionalidades de algunos sistemas informáticos que permiten o ayudan a tomar decisiones nacionales e internacionalmente. Teniendo en cuenta las funcionalidades que pueden realizar cada uno de estos *software*, se analiza la propuesta de solución que se llevará a cabo. Se realiza además un estudio comparativo de los lenguajes de programación, las metodologías ya sean ligeras o pesadas, las herramientas de trabajo, y las tecnologías más propicias para la confección del *software* a construir. Este análisis contribuirá a desarrollar una selección eficaz de los elementos que se combinarán para desarrollar el sistema en cuestión y estará acorde con los requerimientos de la base reglamentaria y los documentos rectores para la elaboración del *software*. Estos elementos conformarán entonces el marco de trabajo a utilizar que brindará una propuesta de solución.

1.2 Trabajos similares

La toma de decisiones se define como la selección de un curso de acciones entre alternativas, es decir, que existe un plan, un compromiso de recursos de dirección o reputación. [1]

Los sistemas de apoyo a las decisiones (SAD) usan computadoras para el facilitar el proceso de toma de decisiones de tareas semiestructuradas. [1]

Estos sistemas están diseñados no para reemplazar el criterio administrativo, sino para apoyarlo y hacer mas efectivo el proceso de toma de decisiones. Los sistemas de respaldo a las decisiones ayudan también a los directivos a reaccionar rápidamente a los cambios de necesidades. Por lo tanto, queda claro que el diseño de un sistema efectivo requiere de un conocimiento profundo de cómo el personal encargado toma las decisiones y encontrándose necesario una interfaz web que consiga propiciar estos requisitos, se necesita entonces un estudio mundial de soluciones que presenten similitudes con el sistema a construir. [1]

1.2.1 Ámbito internacional

En el caso genérico de los sistemas de ayuda a la toma de decisiones son ampliamente empleados tanto en la medicina, la ingeniería, en la economía así como militarmente. A continuación se hace relación de algunos de estos sistemas en el plano militar en el mundo.

Sistema de Control del Espacio [2]

Propone un modelo capaz de responder a la problemática de asistencia a la toma de decisiones críticas en el Centro de Información y Control (CIC). Sistematiza y documenta con metodología de sistemas expertos el conocimiento requerido en un CIC. Determina las bases para la actuación de un Sistema Experto en ambientes de entrenamiento y asistencia a la toma de decisiones en un Centro de Información y Control Aéreo. Contribuye al entrenamiento de personal responsable del control, convirtiéndose en una potencial herramienta que permite evaluar las decisiones tomadas frente a una situación dada.

Concurrencia y Tiempo-Real en los Sistemas Distribuidos de Mando y Control [3]

Esta aplicación tiene como misión fundamental ayudar a la toma de decisiones de los mandos militares, realizando por ellos labores de planificación y de aplicación de la doctrina militar en entornos complejos y bajo condiciones extremas. Estas labores son delegadas en aplicaciones *software* que simulan los actuales papeles de trabajo de las secciones de la Plana Mayor de cada comandante.

Uno de los mayores problemas que actualmente tienen este tipo de Planas Mayores es la de sincronizar y coordinar toda la inteligencia militar que son capaces de acumular en un momento dado, seleccionando aquellos que son decisivos de los que no lo son. Al mismo tiempo, el mando militar dispone, en cada momento, de las soluciones parciales que se tengan en la cadena de producción de las mismas, con el fin de tomar decisiones, si fuese necesario, antes de alcanzar la solución óptima con el fin de ir previniendo posibles movimientos tácticos tanto ofensivos como defensivos.

La info-esfera conjunta del espacio de batalla (JBI) [4]

Consta de "espacio de información" globalmente interoperable que integre, añada y difunda de manera ingeniosa información relevante sobre el espacio de batalla para apoyar la toma de decisiones eficaz. La info-esfera forma parte de un sistema global de gestión de información de combate concebido para

proporcionarles a los usuarios individuales en todos los niveles de mando, información diseñada específicamente para sus responsabilidades funcionales.

Sistemas expertos en la planificación de operaciones tácticas militares (3GWG). [5]

El presente proyecto fue diseñado sobre la base del conocimiento y experiencia del experto militar quien proporcionó información sobre el desarrollo de estos planes, los mismos se encuentran dentro de la base de conocimiento del Sistema Experto Propuesto. Básicamente es un juego de guerra de tercera generación (3GWG), que al incorporar tres aspectos principales adicionales—ciclos de decisión, factores humanos y efectos operacionales—el 3GWG incrementa los juegos de guerra de segunda generación que aplican con éxito la reducción gradual de recursos, el movimiento y la logística. Además, los 3GWG ayudarán a educar a los encargados de tomar decisiones asistiéndoles a tomar mejores soluciones.

MELIS [6]

Es un marco de trabajo para la construcción de Sistemas de Ayuda a la Toma de Decisiones (DSS) en problemas de monitorización. Áreas de aplicación típica de los DSS son: la gestión y planificación de negocios, atención sanitaria, militar y cualquier área de gestión en el que se encuentren situaciones de toma de decisiones complejas. Se usa principalmente para decisiones estratégicas y tácticas en la gestión de nivel superior.

1.2.2 Ámbito nacional

En el contorno nacional este tipo de sistema, que es capaz de auxiliar en la toma de decisiones. Específicamente en el área militar, existe uno, pero que se encuentra sólo en estado de diseño y además generalizado para todos los puestos de mando y el Estado Mayor, no encontrándose ningún sistema que esté calificado para asistir en la toma de decisiones en un puesto de mando que requiera particularidades específicas individuales de él.

Sistema Informático de Mando y Estado Mayor, Módulo y Preparación de Acciones Combativas. [8]

Se diseñó un subsistema para la Planificación de Acciones Combativas tanto en tiempo de paz como en tiempo de guerra, para el apoyo del proceso docente educativo que llevan a cabo los Jefes que cursan estudios en la Academia de las FAR.

1.3 Arquitectura

La calidad de una aplicación web es considerablemente influenciada por su arquitectura subyacente. Aspectos arquitectónicos incompletos o perdidos hacen que sea difícil darse cuenta de los requisitos de la calidad de las aplicaciones web. Bajo rendimiento, insuficiente mantenimiento, y la baja disponibilidad de una aplicación web es a menudo causada por una inadecuada arquitectura. Además de las limitaciones técnicas, la arquitectura de las aplicaciones web también debe considerar el marco organizativo en el que están incorporados. El uso flexible de las arquitecturas multicapas, el examen de contenidos multimedia, y la integración de los repositorios de datos y aplicaciones son desafíos en el desarrollo exitoso de las arquitecturas de las aplicaciones web. [8]

Modelo Cliente Servidor

El modelo cliente-servidor, son un grupo de programas distribuidos que están corriendo en la red de computadoras, e interactúan a través de protocolos de comunicación conocidos, en lugar de funcionar todos en un único sistema, la aplicación cliente-servidor distribuye el procesamiento entre el servidor y los equipos clientes. [9]

Es la tecnología que proporciona al usuario final el acceso transparente a las aplicaciones, datos, servicios de cómputo o cualquier otro recurso del grupo de trabajo y/o a través de la organización, en múltiples plataformas. El Cliente y el Servidor pueden actuar como una sola entidad y también pueden actuar como entidades separadas, realizando actividades o tareas independientes. Este esquema facilita la integración entre sistemas diferentes y comparte información permitiendo que las máquinas ya existentes puedan ser utilizadas pero utilizando interfaces más amigables al usuario. [10]

Una aplicación web es una aplicación cliente-servidor (generalmente), utiliza los navegadores web como su cliente. Los navegadores envían las solicitudes a los servidores, los servidores generan una respuesta y se la regresan a los navegadores. [9]

La arquitectura cliente-servidor llamado modelo cliente-servidor es una forma de compartimentar y centrar los programas y las computadoras para ganar en eficiencia, simplificación, y objetividad. Las aplicaciones webs están basadas en varios componentes que se comunican entre sí mediante el principio petición-respuesta. Un componente envía una petición a otro componente y este envía una respuesta a dicha

solicitud a través de un canal de comunicación sincrónico. A continuación se muestra una breve descripción de los componentes principales: [9]

Cliente: generalmente es un navegador web que es utilizado por un usuario para acceder a la aplicación.

Servidor web: es un *software* específico que soporta varios protocolos entre los cuales están *HTTP* y *HyperText Transfer Protocol Security* (HTTPS). Permite además procesar y responder a las peticiones de los clientes.

Servidor de base de datos: es utilizado para almacenar información persistente. La figura 1.1 muestra la relación entre ellos.

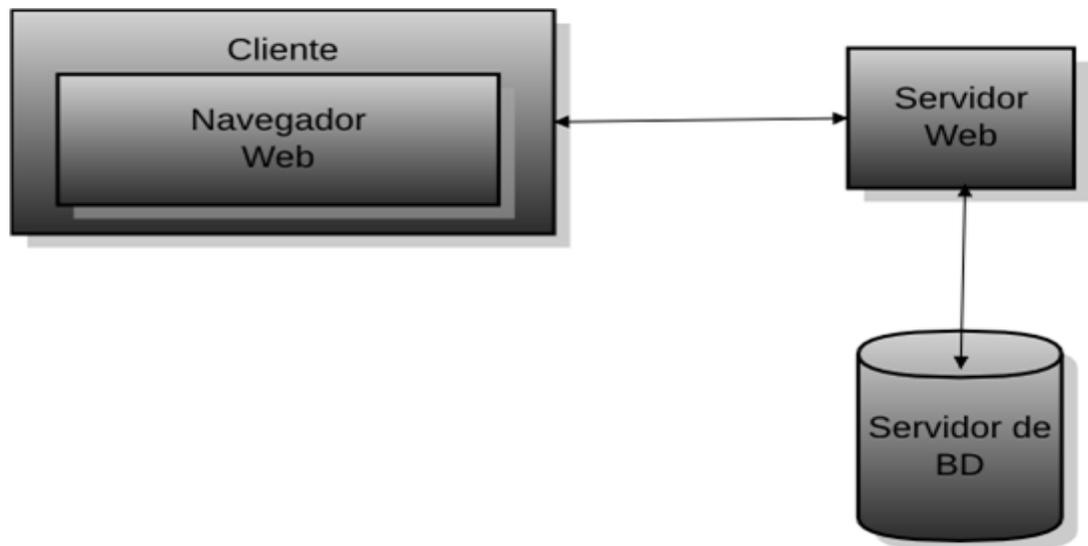


Figura 1.1: Componentes genéricos de una aplicación web.

Patrón de arquitectura Modelo Vista Controlador (MVC) [11]

La arquitectura MVC separa la lógica de negocio (el modelo) y la presentación (la vista) por lo que se consigue un mantenimiento más sencillo de las aplicaciones. Si por ejemplo una misma aplicación debe ejecutarse tanto en un navegador estándar como en un navegador de un dispositivo móvil, solamente es necesario crear una vista nueva para cada dispositivo; manteniendo el controlador y el modelo original. El

controlador se encarga de aislar al modelo y a la vista de los detalles del protocolo utilizado para las peticiones (HTTP, consola de comandos, *email*, etc.). El modelo se encarga de la abstracción de la lógica relacionada con los datos, haciendo que la vista y las acciones sean independientes de, por ejemplo, el tipo de gestor de bases de datos utilizado por la aplicación. [11] MVC divide una aplicación interactiva en 3 áreas: procesamiento, salida y entrada. Para esto, utiliza las siguientes abstracciones: [34]

- Modelo (Model): Encapsula los datos y las funcionalidades. El modelo es independiente de cualquier representación de salida y/o comportamiento de entrada.
- Vista (View): Muestra la información al usuario. Pueden existir múltiples vistas del modelo. Cada vista tiene asociado un componente controlador.
- Controlador (Controller): Reciben las entradas, usualmente como eventos que codifican los movimientos o pulsación de botones del ratón, pulsaciones de teclas, etc. Los eventos son traducidos a solicitudes de servicio ("*service requests*") para el modelo o la vista.

A continuación se presenta una imagen con el modo de funcionamiento de este patrón.



Figura 1.2 Funcionamiento del patrón MVC

Ventajas [12]

- La separación del modelo de la vista, es decir, separa los datos de la representación visual de los mismos.
- Es mucho más sencillo agregar múltiples representaciones de los mismos datos o información.
- Facilita agregar nuevos tipos de datos según sea requerido por la aplicación ya que son independientes del funcionamiento de las otras capas.
- Crea independencia de funcionamiento.
- Facilita el mantenimiento en caso de errores.
- Ofrece maneras más sencillas para probar el correcto funcionamiento del sistema.
- Permite el escalamiento de la aplicación en caso de ser requerido.

Desventajas: [12]

- La separación de conceptos en capas agrega complejidad al sistema.
- La cantidad de archivos a mantener y desarrollar se incrementa considerablemente.
- La curva de aprendizaje de este patrón de diseño es más alta que utilizando otros más sencillos.

1.4 Tecnologías a utilizar

1.4.1 Lenguajes

Un lenguaje de programación es un lenguaje que puede ser utilizado para controlar el comportamiento de una computadora. Consiste en un conjunto de símbolos y reglas sintácticas y semánticas que definen su

estructura y el significado de sus elementos y expresiones. Un lenguaje de programación permite a uno o más programadores especificar de manera precisa: sobre qué datos una computadora debe operar, cómo deben ser estos almacenados, 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. [10]

1.4.1.1 Lenguajes del lado del servidor

Se les clasifica así a los lenguajes de programación en la tecnología cliente servidor que se ejecutan del lado del servidor y de los que los cuales los usuarios solo obtienen el beneficio del procesamiento de la información [10]

PHP

PHP es el acrónimo de PHP Hipertext Preprocesor[13]. Es un lenguaje de programación del lado del servidor gratuito e independiente de la plataforma, rápido, con una gran librería de funciones y mucha documentación. PHP es un lenguaje híbrido entre algunos de sus predecesores como C, shell, Perl, tomando lo mejor de estos lenguajes y creando un fácil lenguaje *script* para la programación de aplicaciones web [13]. PHP se escribe dentro del código HTML, lo que lo hace realmente fácil de utilizar. Cualquiera puede descargar a través de la página principal de PHP (<http://www.php.net/>) y de manera gratuita, un módulo que hace que el servidor web comprenda los *scripts* realizados en este lenguaje. Dado que es independiente de plataforma, puesto que existe un módulo de PHP para casi cualquier servidor web, hace que cualquier sistema pueda ser compatible con el lenguaje y significa una ventaja importante, ya que permite portar el sitio desarrollado en PHP de un sistema a otro sin mucho esfuerzo. [14]

Características. [15]

- Simplicidad: usuarios con experiencia en Perl, C o C++ podrán utilizarlo rápidamente.
- Velocidad: alta velocidad de ejecución sin introducir demoras en la máquina, bajo consumo de recursos y muy buena integración con Apache.

- Estabilidad: ninguna aplicación está completamente libre de errores (*bugs*), pero con una amplia comunidad de programadores y usuarios corregirlos es mucho más fácil. Posee un sofisticado manejo de variables que lo hacen muy robusto y estable.
- Seguridad: permite la protección contra diversos ataques a través de diferentes niveles de seguridad que pueden ser configurados desde el archivo `.ini`.

Ventajas. [15]

- PHP es *Open Source* ó código abierto, lo cual significa que el usuario no depende de una compañía específica para arreglar cosas que no funcionan, además no está forzado a pagar actualizaciones anuales para tener una versión que funcione.
- Gran número de funciones predefinidas. A diferencia de otros lenguajes de programación, PHP fue diseñado especialmente para el desarrollo de páginas web dinámicas. Por ello, está dotado de un gran número de funciones que simplificarán enormemente tareas habituales como descargar documentos, enviar correos, trabajar con *cookies* y sesiones.
- Puede interactuar con muchos motores de bases de datos tales como MySQL, Oracle, Informix, PostgreSQL ó cualquier otra base de datos compatible con ODBC. Actualmente se puede ejecutar bajo Apache, IIS, AOLServer, Roxen y THTTPD.
- PHP corre en "casi" cualquier plataforma utilizando el mismo código fuente, pudiendo ser compilado y ejecutado en aproximadamente 25 plataformas, incluyendo diferentes versiones de Unix, Windows (95, 98, NT, ME, 2000, XP, etc.) y Macs.
- En una entrevista, el creador de PHP Rasmus Lerdorf resumió de forma brillante porque usar PHP, PHP como lenguaje de programación no es mejor, probablemente es peor que otros lenguajes, pero lo que tiene PHP es que está orientado a un objetivo: la programación web en Internet.

1.4.1.2 Lenguajes del lado del cliente

Un lenguaje del lado cliente es totalmente independiente del servidor, lo cual permite que la página pueda ser albergada en cualquier sitio. Pero nuestra página no se verá bien si el ordenador cliente no tiene instalados los *plug-in* adecuados. El código, tanto del hipertexto como de los *scripts*, es accesible a cualquiera y ello puede afectar a la seguridad. [10]

HTML

HTML es el acrónimo inglés de HyperText Markup Language, que se traduce al español como Lenguaje de Marcas Hipertextuales. Es un lenguaje de marcación diseñado para estructurar textos y presentarlos en forma de hipertexto, que es el formato estándar de las páginas web. Gracias a Internet y a los navegadores como Internet Explorer, Opera, Firefox, Netscape o Safari, el HTML se ha convertido en uno de los formatos más populares y fáciles de aprender que existen para la elaboración de documentos para web. [10]

XML

XML, sigla en inglés de *eXtensible Markup Language* («lenguaje de marcas extensible»), es un metalenguaje extensible de etiquetas desarrollado por el *World Wide Web Consortium* (W3C). Permite definir la gramática de lenguajes específicos (de la misma manera que HTML). Por lo tanto, XML no es realmente un lenguaje en particular, sino una manera de definir lenguajes para diferentes necesidades. XML no ha nacido sólo para su aplicación en Internet, sino que se propone como un estándar para el intercambio de información estructurada entre diferentes plataformas. Se puede usar en bases de datos, editores de texto, hojas de cálculo y casi cualquier cosa imaginable. XML es una tecnología sencilla que tiene a su alrededor otras que la complementan y la hacen mucho más grande y con unas posibilidades mucho mayores. Tiene un papel muy importante en la actualidad ya que permite la compatibilidad entre sistemas para compartir la información de una manera segura, fiable y fácil. Es extensible, lo que quiere decir que una vez diseñado un lenguaje y puesto en producción, igual es posible extenderlo con la adición de nuevas etiquetas de manera de que los antiguos consumidores de la vieja versión todavía puedan entender el nuevo formato. El analizador es un componente estándar, no es necesario crear un analizador específico para cada lenguaje. Esto posibilita el empleo de uno de los tantos disponibles. De esta manera,

se evitan *bugs* y se acelera el desarrollo de la aplicación. Si un tercero decide usar un documento creado en XML, es sencillo entender su estructura y procesarlo. Mejora la compatibilidad entre aplicaciones. [10]

Java Script

JavaScript es un lenguaje 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. Al contrario que Java, JavaScript no es un lenguaje orientado a objetos propiamente dicho, ya que no dispone de Herencia, es más bien un lenguaje basado en prototipos, ya que las nuevas clases se generan clonando las clases base (prototipos) y extendiendo su funcionalidad. Todos los navegadores interpretan el código JavaScript integrado dentro de las páginas web. Para interactuar con una página web se provee al lenguaje JavaScript de una implementación del DOM. Javascript es un lenguaje con muchas posibilidades, utilizado para crear pequeños programas que luego son insertados en una página web y en programas más grandes, orientados a objetos mucho más complejos. Con Javascript se puede crear diferentes efectos e interactuar con nuestros usuarios. Javascript es soportado por la mayoría de los navegadores como Internet Explorer, Netscape, Opera, Mozilla Firefox, entre otros. [10]

Tecnología AJAX

AJAX, acrónimo de Asynchronous JavaScript And XML, es una técnica de desarrollo Web para crear aplicaciones interactivas. Éstas se ejecutan en el navegador de los usuarios y mantiene comunicación asíncrona con el servidor en segundo plano. De esta forma, es posible realizar cambios sobre la misma página sin necesidad de recargarla. AJAX es una combinación de tecnologías ya existentes como son: HTML y hojas de estilos CSS para generar estilos, implementaciones ECMAScript (uno de ellos es el lenguaje Javascript) y XMLHttpRequest que es una de las funciones más importantes que incluye, esta permite intercambiar datos asincrónicamente con el servidor Web, puede ser mediante PHP, ASP, JSP entre otros. [10]

Cascade Style Sheets (CSS)

Cascading Style Sheets, u Hoja de Estilo en Cascada es la tecnología desarrollada por el *World Wide Web Consortium* (W3C) con el fin de separar la estructura de la presentación. A pesar de que la recomendación oficial del grupo de trabajo de la W3C ya había alcanzado la estabilidad requerida para

que fuera soportada por los principales navegadores comerciales, como Netscape e Internet Explorer, tan tempranamente como en el año 1998, la situación de entonces, comúnmente conocida como la “guerra de los navegadores”, hacía que los intereses comerciales de las dos compañías en lucha por el mercado de usuarios de Internet se interpusieran en el camino de las CSS. [16]

¿Qué es CSS?

Hojas de Estilo en Cascada (Cascading Style Sheets), es un mecanismo simple que describe cómo se va a mostrar un documento en la pantalla, o cómo se va a imprimir, o incluso cómo va a ser pronunciada la información presente en ese documento a través de un dispositivo de lectura. Esta forma de descripción de estilos ofrece a los desarrolladores el control total sobre estilo y formato de sus documentos. [16]

¿Para qué sirve?

CSS se utiliza para dar estilo a documentos HTML y XML, separando el contenido de la presentación. Los Estilos definen la forma de mostrar los elementos HTML y XML. CSS permite a los desarrolladores web controlar el estilo y el formato de múltiples páginas web al mismo tiempo. Cualquier cambio en el estilo marcado para un elemento en la CSS afectará a todas las páginas vinculadas a esa CSS en las que aparezca ese elemento. [16]

1.5 Gestor de Base de Datos

PostgreSQL

PostgreSQL es un servidor libre, liberado bajo la licencia BSD. Que ofrece un potencial adicional al incorporar cuatro conceptos básicos mediante los cuales los usuarios pueden extender fácilmente los sistemas: clases, herencia, tipos y funciones. Nativamente provee soporte para: números de precisión arbitraria, texto de largo ilimitado, figuras geométricas (con una variedad de funciones asociadas), direcciones IP (IPv4 e IPv6), bloques de direcciones estilo CIDR, direcciones MAC y *Arrays*. A lo largo de su desarrollo se han implementado importantes características del motor de datos, incluyendo subconsultas, valores por defecto, restricciones a valores en los campos (*constraints*) y disparadores (*triggers*), se han añadido funcionalidades en línea con el estándar SQL92, incluyendo claves primarias,

identificadores entrecomillados, forzado de tipos cadena literal, conversión de tipos y entrada de enteros binarios y hexadecimales. Los tipos internos han sido mejorados, incluyendo nuevos tipos de fecha/hora de rango amplio y soporte para tipos geométricos adicionales. La velocidad del código del motor de datos ha sido incrementada aproximadamente en 40%, y su tiempo de arranque ha bajado el 80% desde que la versión 6.0 fuera lanzada, otra de las ventajas incorporada de PostgreSQL es el bloqueo de registros. Algunos de los lenguajes que se pueden usar con el PostgreSQL son los siguientes: un lenguaje propio llamado PL/PgSQL (similar al PL/SQL de Oracle), C, C++, Gambas, Java PL/Java Web, PL/Perl, pPHP, PL/Python, PL/Ruby, [10]

1.6 Metodología de desarrollo

Proceso de desarrollo de *software* UCID [17]

El desarrollo de cualquier tipo de proyecto consiste en una serie de fases, muchas veces secuenciales conocidas como el ciclo de vida del proyecto y generalmente definen: qué trabajo técnico se hará en cada fase, cuando se generarán los entregables, como se revisarán, verificarán y validarán en estas, quién está involucrado en cada fase y cómo controlar y aprobar cada una. Las principales características del ciclo de vida expuesto en nuestra organización son:

- Las fases son secuenciales y su transferencia debe ser precedida por un proceso de revisión o liberación del Centro de Calidad y su aprobación en Consejo Técnico Formal.
- Está basado en componentes.
- El nivel del personal es bajo al comienzo, alcanza su nivel máximo en la fase de construcción y decae rápidamente cuando el proyecto se aproxima a su conclusión.
- La participación de los interesados es alta en las etapas de Inicio y Modelación, baja en la etapa de Construcción y vuelve a subir en las etapas finales del proyecto.
- En nuestra unidad el ciclo de vida del proyecto se considera como parte del ciclo de vida del producto, el cual representa un marco de referencia que contiene los procesos, las actividades y las tareas involucradas en la adquisición, el desarrollo, la explotación y el mantenimiento de un producto.



Figura 1.3 Etapas del ciclo de vida del proyecto

1.6.1 Lenguaje de modelado UML

Lenguaje Unificado de Modelado UML, por sus siglas en inglés, es el lenguaje de modelado de sistemas de *software* más conocido y utilizado en la actualidad; está respaldado por el *Object Management Group* (OMG). Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema. UML ofrece un estándar para describir un “plano” del sistema (modelo), incluyendo aspectos conceptuales tales como procesos de negocio y funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y componentes reutilizables [18].

1.6.2 Notación de Modelados de Procesos del Negocio (BPMN) [19]

¿Qué es BPMN?

BPMN (*Business Process Modeling Notation*) es un nuevo estándar de modelado de procesos de negocio, en donde se presentan gráficamente las diferentes etapas del proceso del mismo. La notación ha sido diseñada específicamente para coordinar la secuencia de procesos y los mensajes que fluyen entre los diferentes procesos participantes.

¿Por qué es importante BPMN?

Imagina esta situación: Estás en las primeras entrevistas de requerimientos. ¿La sorpresa? Tus usuarios no están seguros cómo funcionan sus procesos de negocio y por lo tanto resulta imposible definir los requerimientos del sistema. En este momento tienes que ponerte la camiseta de un ingeniero de negocios y quitarte la del ingeniero de *software*. Pero, si no cuentas con las herramientas y técnicas adecuadas, las reuniones con tu cliente pueden ser eternas sin llegar a respuestas concretas.

¿La solución? Aprende el estándar BPMN y las extensiones de UML que te ayudarán a modelar la situación actual y deseada en los procesos de negocio de tu cliente. Ya tienes claro que si no partes de reglas de negocio claramente establecidas difícilmente podrás desarrollar el sistema adecuado que proporcione un valor real a tu cliente.

El mundo de los procesos de negocio ha cambiado dramáticamente en los últimos años. Un proceso de este tipo abarca múltiples participantes, y la coordinación puede ser compleja. Antes de BPMN no había una técnica de modelado estándar desarrollado para encargarse de estos asuntos. BPMN ha sido desarrollado para proveer a los usuarios de una notación de uso libre. Esto beneficiará a los usuarios de la misma forma que UML benefició el mundo de la ingeniería de *software*.

1.7 Herramientas de programación

Entorno de desarrollo integrado (*Integrated Development Environment (IDE)*)

Un entorno de desarrollo integrado o en inglés *Integrated Development Environment* (IDE) es un programa compuesto por un conjunto de herramientas para un programador. Puede dedicarse en exclusiva a un sólo lenguaje de programación o bien, poder utilizarse para varios. Un IDE es un entorno de programación que ha sido empaquetado como un programa de aplicación, es decir, consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica. Los IDEs pueden ser aplicaciones por si solas o pueden ser parte de aplicaciones existentes. Los IDEs proveen un marco de trabajo amigable para la mayoría de los lenguajes de programación tales como C++, Java, C#, Delphi, *Visual Basic*, *Object Pascal*, *Velneo*, etc. Es posible que un mismo IDE pueda funcionar con varios lenguajes de programación. [10] En la actualidad son varios los que están disponibles para el desarrollo de un proyecto utilizando PHP como Zend Studio¹, Open Komodo Project², PhpEdit⁷, Php Designer³, Netbeans⁴, Eclipse PDT⁵. En la prestigiosa revista Smashing Magazine⁶ nos muestran un estupendo artículo *The Big PHP IDE Test: Why Use One And Which To Cose*.

ZEND Studio

Zend Studio es una herramienta *software* profesional excelente para trabajar proyectos PHP. Es concebido con el fin de crear aplicaciones altamente fiables, proporciona una facilidad de uso inigualable, escalabilidad, fiabilidad, y la extensión que los programadores profesionales y de empresas requieren para desarrollar, distribuir, depurar y administrar aplicaciones PHP críticas de negocios.

Permite conectarse directamente con las bases de datos profesionales más utilizadas tales como IBM DB2/Cloudscape/ Derby/, MySQL, Oracle, Microsoft SQL Server, PostgreSQL y SQLite. Tiene características de depuración avanzadas, incluyendo condiciones límite, visualización de errores, vistas avanzadas, variables y *buffer* de salida. Asegura la protección máxima de ubicaciones de proyectos o en Internet con depuradores remotos. Facilita el desarrollo y colaboración en equipo mediante la administración efectiva de su código fuente utilizando CVS o Subversión directamente desde Zend Studio. Tiene soporte para PHP 5 completo, analizador de código, carpeta de código, completado de código, coloreado de sintaxis, administrador de proyecto, editor de código, depurador de gráficos y asistentes. [10]

1.8 Herramientas BD

PG adminIII

pgAdmin III es una aplicación gráfica para gestionar el gestor de bases de datos PostgreSQL, teniendo licencia Open Source. Está escrita en C++ usando la librería gráfica multiplataforma wxWidgets, lo que permite que se pueda usar en Linux, FreeBSD, Solaris, Mac OS X y Windows. Es capaz de gestionar versiones a partir de la PostgreSQL 7.3 ejecutándose en cualquier plataforma, así como versiones comerciales de PostgreSQL como Pervasive Postgres, EnterpriseDB, Mammoth Replicator y SRA PowerGres.[22]

pgAdmin III está diseñado para responder a las necesidades de todos los usuarios, desde escribir consultas SQL simples hasta desarrollar bases de datos complejas. El interfaz gráfico soporta todas las características de PostgreSQL y facilita enormemente la administración. La aplicación también incluye un editor SQL con resaltado de sintaxis, un editor de código de la parte del servidor, un agente para lanzar *scripts* programados, soporte para el motor de replicación Slony-I y mucho más. La conexión al servidor puede hacerse mediante conexión TCP/IP o Unix Domain Sockets (en plataformas *nix), y puede encriptarse mediante SSL para mayor seguridad. [22]

1.9 Herramientas de modelado

Herramienta Case: Visual Paradigm

Visual Paradigm es una herramienta CASE profesional que soporta la última versión de UML 2.1 así como el ciclo de vida completo del desarrollo de *software*, análisis y diseño orientados a objetos, construcción, pruebas y despliegue, exportación desde Rational Rose, exportación/importación XML, generación de informes y edición de figuras. Visual Paradigm tiene disponible distintas versiones: *Enterprise*, *Professional*, *Standard*, *Modeler*, *Personal* y *Community* (que es gratuita). [23] Visual Paradigm ofrece además:

- Uso de un lenguaje estándar común a todo el equipo de desarrollo que facilita la comunicación.
- Capacidades de ingeniería directa (versión profesional) e inversa
- Modelo y código que permanece sincronizado en todo el ciclo de desarrollo
- Disponibilidad de múltiples versiones, para cada necesidad

- Disponibilidad de integrarse en los principales IDE [1](#)
- Disponibilidad en múltiples plataformas
- Ingeniería Inversa Java, C++, Esquemas XML, XML5, NET exe/dll, CORBA IDL
- Ingeniería de ida y vuelta.

1.10 Framework

En general, el término *framework*, se refiere a una estructura de *software* compuesta de componentes personalizables e intercambiables para el desarrollo de una aplicación. En otras palabras, un *framework* se puede considerar como una aplicación genérica incompleta y configurable a la que se le puede añadir las últimas piezas para desarrollar una aplicación concreta. Por lo que resumiendo, un *framework* es el esqueleto de una aplicación que debe ser adaptado por el programador según sus necesidades específicas para desarrollar una aplicación. [20]

Los objetivos principales que persigue un *framework* son: acelerar el proceso de desarrollo, reutilizar código ya existente y promover buenas prácticas de desarrollo como el uso de patrones. Un *framework* web, por tanto, podemos definirlo como un conjunto de componentes que componen un diseño reutilizable que facilita y agiliza el desarrollo de sistemas web. [20]

Framework SAUXE

El framework SAUXE constituye un híbrido, de diferentes marcos de trabajo que se encuentra siendo utilizado en la Unidad de Compatibilización e Integración para la Defensa (unidad 1722) con mucho éxito. Reúne en él las principales y mejores características de estos *frameworks* como son: Doctrine, Zend Framework y ExtJs, proporcionando un entorno de trabajo agradable para el que desee utilizarlo.

Framework de Persistencia de Objetos Relacionales Doctrine

Las bases de datos siguen una estructura relacional. PHP 5 por el contrario es orientado a objetos. Por este motivo, para acceder a la base de datos como si fuera orientada a objetos, es necesario contar con

una interfaz que traduzca la lógica de los objetos a la lógica relacional. Esta interfaz es lo que se conoce como ORM. Un ORM consiste en una serie de objetos que permiten acceder a los datos y que contienen en su interior cierta lógica de negocio

¿Qué es Doctrine?

Doctrine es un ORM que posee una poderosa capa de abstracción de BD. Una de sus características es la opción de escribir consultas de BD en un objeto apropiado orientado al dialecto SQL (Structured Query Language) y que se le denomina Lenguaje de Consulta de Doctrine o DQL del inglés Doctrine Query Language, inspirado por el Lenguaje de Consulta de Hibernate (HQL por sus siglas en inglés). Este proporciona a los desarrolladores una poderosa alternativa al SQL que mantiene la flexibilidad sin requerir duplicación de código innecesario. [21]

Principales características de Doctrine [21]

Doctrine es un framework para el mapeo objeto – relacional para PHP que está dividido en dos capas principales, la DBAL del inglés *Database Abstraction Layer* y el ORM. La imagen que se muestra a continuación refleja cómo las capas de Doctrine trabajan juntas.

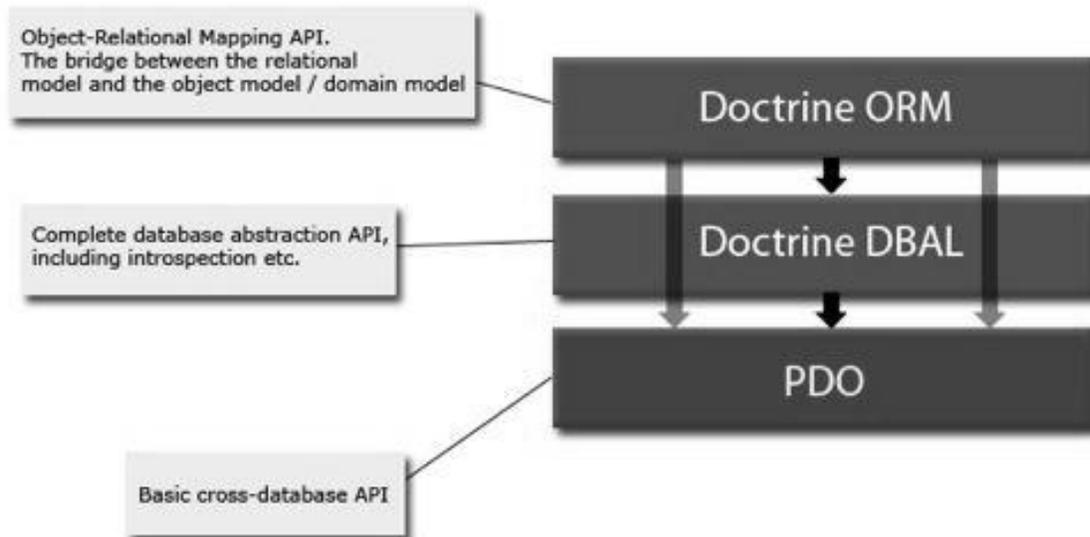


Figura 1.4 Estructura del framework Doctrine

Zend Framework [24]

Zend Framework se trata de un *framework* para desarrollo de aplicaciones Web y servicios Web con PHP, te brinda soluciones para construir sitios web modernos, robustos y seguros. Además, es *Open Source* y trabaja con PHP 5.

Principales características:

- Trabaja con MVC (*Model View Controller*)
- Cuenta con módulos para manejar archivos PDF, canales RSS, Web Services (Amazon, Flickr, Yahoo), etc.
- El Marco de Zend también incluye objetos de las diferentes bases de datos, por lo que es extremadamente simple para consultar su base de datos, sin tener que escribir ninguna consulta SQL.

- Una solución para el acceso a base de datos que balancea el ORM con eficiencia y simplicidad. Podemos ver esta función en el futuro.
- Completa documentación y tests de alta calidad.
- Soporte avanzado para i18n (internacionalización).
- Un buscador compatible con Lucene.
- Robustas clases para autenticación y filtrado de entrada.
- Clientes para servicios web, incluidos Google Data APIs y Strikelron.
- Muchas otras clases útiles para hacerlo tan productivo como sea posible.

Ext JS [7]

Framework, Componentes o Librerías. Amparado bajo GPLv3, Ext. 2.0 es un *framework* completo y extremadamente avanzado, que contiene casi todo lo que te puedas imaginar y nace como solución a tareas comunes -pero complejas. Este framework te hará recordar la programación de entornos visuales, ya que es completamente basado en la Programación Orientada a Objetos (POO). Cada objeto contiene lo típico: propiedades, métodos, eventos..., etc. Ext. basa toda su funcionalidad en JavaScript a través de librerías ya muy conocidas: YUI, jQuery y Prototype/Script.aculo.us y un núcleo interno poderoso. Así, en tiempo de ejecución carga y crea todos los objetos html a través del uso intenso de DOM.

Ventajas:

- La orientación intensa a objetos te hará modular todos tus *scripts*.
- El diseño está completamente separado de la funcionalidad.
- Funciones comunes como validación, *comboxes* editables, ventanas arrastables (con minimizar y maximizar), grillas editables, son muy fáciles de implementar.
- Buena y amplia documentación, así como también su comunidad.

1.11 Propuesta de solución

Teniendo en cuenta que un sistema de apoyo toma de decisiones es un *software* capaz de auxiliar a alguna persona en particular proporcionándole datos o elementos con los cuales el usuario pueda valorar una situación y esté en condiciones de tomar una decisión respecto a algún tema en particular el módulo que se pretende realizar tiene como objetivo principal proporcionar esta funcionalidad. Después de estudiar varias soluciones existentes a nivel internacional de interfaces web de sistemas de apoyo a la toma de decisiones, incluido un *software* nacional de planificación de acciones combativas tanto en tiempo de paz como de guerra para apoyar el proceso docente educativo en la academia de las Fuerzas Armadas Revolucionarias, pero dado que estos sistemas son tan genéricos se ha ideado realizar un módulo que sea específicamente para la especialidad de Lucha Radio Electrónica.

Confeccionar una aplicación con los requisitos necesarios de seguridad, portabilidad y escalabilidad, entre otros, suponen una arquitectura, tecnología y marcos de trabajo apropiados así como un diseño de base de datos rica en virtudes que proporcionen las cualidades que se desean obtener del *software*. Los lenguajes de programación a escoger representan una amplia gama de cualidades en cuanto a las ventajas que tiene cada uno. Teniéndose en cuenta la base reglamentaria que es utilizada en la unidad, se utilizará PHP como lenguaje del lado del servidor. Del lado del cliente se utilizará JavaScript, HTML y XML por su sencillez para ser comprendidos. Como *framework* se escogió el marco de trabajo que se realizó en la UCID, SAUXE, dado que representa una combinación de las funcionalidades más efectivas de los *framework* *DOCTRINE*, *Extjs* y *ZendFramework* Para ello se puso atención en el IDE *ZendStudio* por ser fácil de trabajar, su escalabilidad y fiabilidad entre otros aspectos que se engranan perfectamente con los lenguajes anteriormente expuestos. Se utilizará a PostgreSQL como gestor de base de datos, así como Pg AdminIII como interfaz para manejar la base de datos.

El proceso de desarrollo de *software* que se tiene en la UCID, es una metodología efectiva para utilizar, además de que se encuentra dentro del marco de trabajo de la unidad. Para realizar el análisis y diseño de la aplicación a realizar se fijó principalmente la atención en la herramienta de modelado Visual Paradigm ya que recorre el ciclo de vida completo de un proyecto y soporta la última versión de UML y BPMN.

1.12 Conclusiones

En este capítulo se realizó un breve recorrido por los principales sistemas de apoyo a la toma de decisiones que existen en el mundo. Tras delimitar las fronteras, características y ventajas de estos *softwares* se expusieron los lenguajes de programación, herramientas, metodologías en el diseño y desarrollo del *software*, mostrando algunas de sus funcionalidades más relevantes para el desarrollo de esta aplicación en particular.

Capítulo 2: Características del Sistema

2.1 Introducción

El desarrollo de un *software* parte de entender la estructura y la dinámica de la organización para la cual el sistema va a ser elaborado, comprender el problema y asegurar que todas las partes involucradas tengan un entendimiento común de los objetivos. El proceso de desarrollo de *software* que se lleva a cabo en la unidad, permite lograr la comprensión del problema desde su fase de inicio, a través del levantamiento de requisitos se identifican los principales procesos, y con la realización del Modelo conceptual, que forma parte del Modelo de Negocio, se logra un mayor entendimiento por todos los implicados. En el caso de la aplicación los sistemas de apoyo a las decisiones (SAD) para LRE, se basa en la descripción de los procesos, lo cual permite mediante un mapa de procesos analizar con más profundidad el mismo y se presentarán los diagramas asociados a estos. Por último, se hará el levantamiento de los requisitos funcionales y no funcionales que debe cumplir el sistema.

2.2 Descripción de los procesos

Georreferenciar nuestras tropas: Para georreferenciar nuestras tropas se necesita definir los niveles estructurales, precisar las estructuras y especificar la composición para las mismas. Cuando la estructura y composición están creadas, se le da a la tropa una ubicación en el terreno.

Controlar la guardia combativa: El jefe del mando superior se informa acerca del personal que se encuentra de guardia y del estado del que consta cada unidad, ya sea cumpliendo algún cometido o no y el tipo de misión en ese instante.

Plantear la misión: Después que se ha dado una situación de combate y se ha analizado la vía más eficiente para combatir la eventualidad, el jefe elabora la misión con la descripción y los datos de lo que debe realizarse, acto seguido se le informa a la unidad combativa su misión.

Cumplir Misión: Para cumplir la misión el Jefe de turno procesa los datos que le son enviados, debe seleccionar los medios con los que se va a cumplir, realiza los cálculos pertinentes, perpetra la tarea encomendada y expide un informe de respuesta con los resultados de la misma.

2.3 Flujo actual de los procesos

Georreferenciar nuestras tropas: Para georreferenciar las unidades, el jefe toma la plantilla de la tropa, se hace de las características de esta y las ubica manualmente en un mapa que tenga destinado para estas funciones.

Controlar la guardia combativa: El jefe del mando superior se informa acerca del personal que se encuentra de guardia en cada unidad y del estado en que se halla a través del teléfono.

Planteamiento de misión: El jefe elabora la misión y la envía a la unidad mediante el correo electrónico de las FAR, o cualquier otra vía confiable para que esta sea entregada.

Cumplir la misión: El Jefe de turno, cuando recibe la misión, realiza los cálculos pertinentes, selecciona los medios, cumple la tarea e informa al jefe superior. Estas acciones son realizadas manualmente, lo que conlleva un gasto de tiempo innecesario.

2.4 Descripción de los sistemas vinculados

Plataforma GENESIG: Plataforma tecnológica para el desarrollo de sistemas de información geográfica desarrollado en la Unidad de Compatibilización e Integración para la Defensa con el objetivo de suplir las necesidades existentes en cuanto a este sentido.

SIDEC: Sistema de información de organización y personal.

2.5 Descripción general del sistema

La preparación, organización y realización de acciones combativas apoyadas en una base de conocimiento que permita a los jefes y especialistas tomar las decisiones más acertadas en la dirección del Ministerio de las Fuerzas Armadas Revolucionarias, posee gran importancia para el devenir exitoso de una misión frente a una situación que así lo requiera. Necesariamente en la especialidad de LRE de las FAR se llevan a cabo estos procesos de preparación y realización de misiones, por tal motivo se hace necesario elaborar un módulo que sea capaz de ayudar a tomar decisiones en un caso dado; suscitándose así que el objetivo principal del presente trabajo sean los procesos de mando y dirección de la especialidad de Lucha Radio Electrónica tanto en tiempo de paz como de guerra.

El Sistema de Apoyo a la Toma de Decisiones es un módulo web, que facilitará el trabajo de los especialistas de LRE que asignarán las misiones. Debe permitir la gestión de las tropas así como su ubicación en el mapa, facilitará igualmente el trabajo de los encargados de cumplir la misión. Debe brindar un conjunto de salidas entre las que se encuentran la Ubicación de las tropas, el Estado de la Guardia en las unidades, la Misión que se va a enviar, y el Reporte de estas misiones que indica el estado de realización de la misma.

2.6 Análisis comparativo con otras soluciones

El análisis comparativo con otras soluciones en el mundo es difícil dado el nivel de seguridad que poseen estos sistemas y la información que se maneja de ellos es muy superficial; y en Cuba no se cuenta con ningún otro sistema de un alcance parecido. No obstante, según el estudio del estado del arte realizado, otras soluciones estudiadas anteriormente son bastante generales en cuanto al tema del apoyo a la toma de decisiones. El presente módulo será capaz de brindar información pertinente a un contexto y el oficial especialista podrá determinar qué hacer. Sin embargo, otros sistemas similares sólo son juegos que evalúan las decisiones que se toman, o dicen directamente lo que se debe hacer, que no es el objetivo que se inquiriere ya que una máquina no puede decidir por encima del ser humano; o bien son marcos de trabajo para realizar Sistemas de Apoyo a las Decisiones (SAD), que en resumen, no cumplen con las especificaciones que se necesitan.

2.7 Modelo de los procesos de negocio

Identificación de los procesos de negocio

- Georeferenciar nuestras tropas.
- Controlar la guardia combativa
- Plantear la misión.
- Cumplir la misión.

2.8 Mapa de los procesos

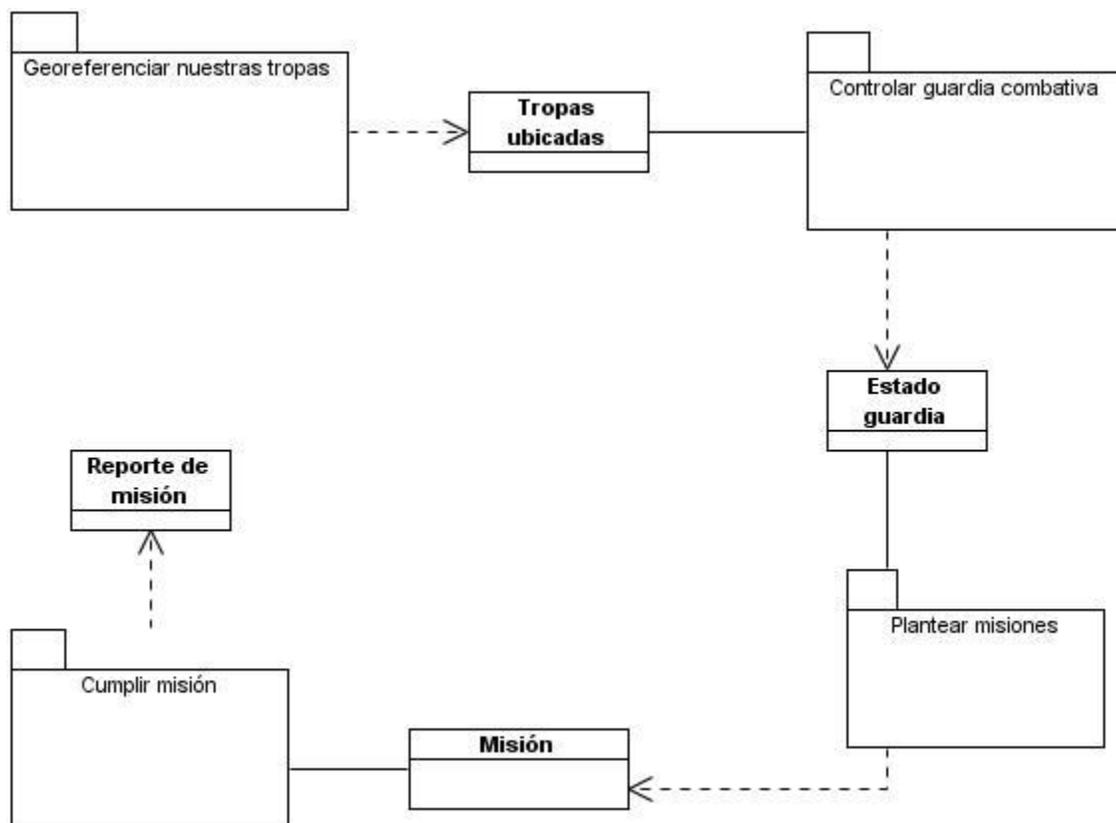


Figura 2.1 Mapa de procesos

2.9 Descripción de los procesos de negocio

Descripción del proceso Georreferenciar

Objetivos	Darle una composición a una tropa y ubicarla en el terreno.
Evento(s) que lo generan	Necesidad de ubicar una tropa.
Precondiciones	Debe existir una tropa.
Pos condiciones	La tropa debe estar creada y ubicada.
Marco jurídico	El sistema se basará en el manual de normas y principios establecidos por el MINFAR.
Clientes internos	-
Clientes externos	Unidades subordinadas.
Entradas	Plantilla de la tropa.
Salidas	Tropa ubicación.

Las restantes tablas de descripción de los procesos del negocio se encuentran en el [Anexo A](#).

Proceso de negocio Georreferenciar nuestras tropas

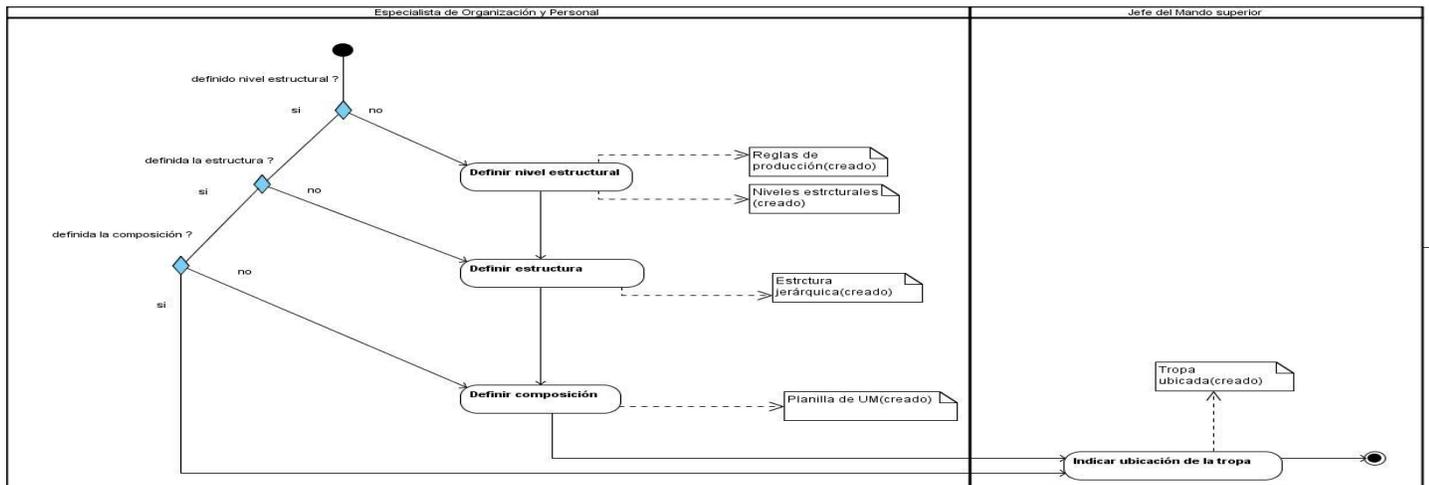


Figura 2.2 Descripción del subproceso Georreferenciar nuestras tropas.

2.10 Identificación de conceptos en el dominio del problema.

Niveles estructurales: Define los conceptos con los que se trabaja.

Reglas de producción: Define la relación entre los niveles.

Estructura jerárquica: Define relación entre los conceptos reales.

Plantilla de UM: Se encargará de guardar el nivel estructural, la composición, el número público y los cargos.

Tropas ubicadas: Se encargará de guardar las coordenadas en las que se encuentra la unidad.

Parte del servicio de guardia: Informará el estado del servicio de guardia en la unidad combativa.

Parte del estado de las unidades combativas: Informe del estado de la guardia en las unidades combativas.

Guardia combativa: Informe general del estado de la guardia combativa en todas las unidades.

Misión: Contendrá todos los datos de la misión acordada por los jefes que debe dársele cumplimiento en la unidad.

Medios a emplear: Medios con los que se llevarán a cabo las misiones.

Reporte de misión: Se guardará el reporte de la misión, si se realizó o no, con una breve descripción.

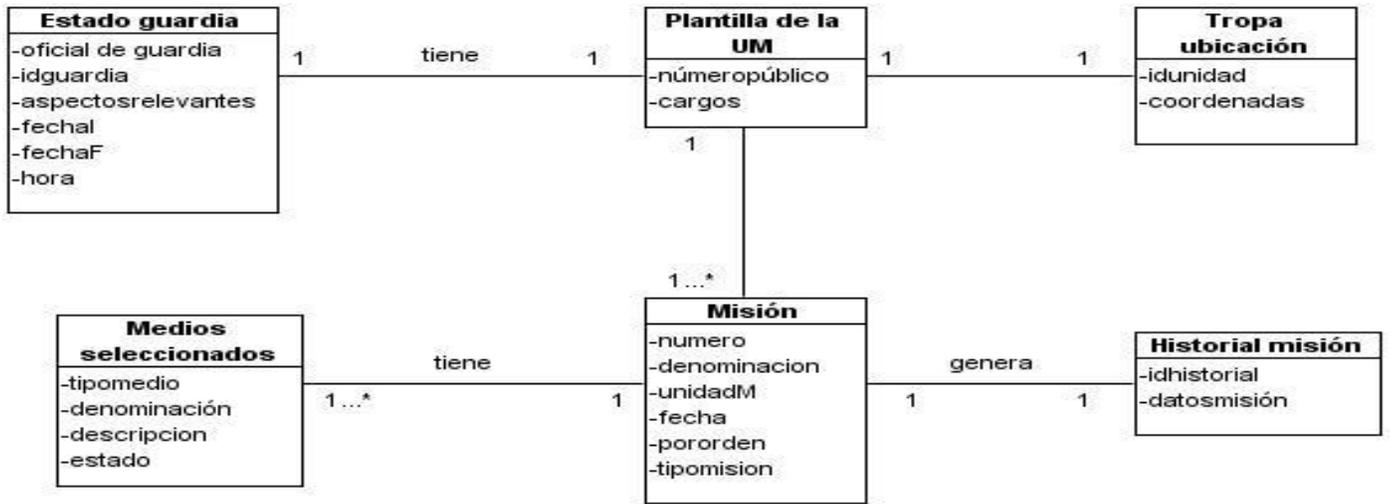


Fig. 2.3 Mapa conceptual del proceso

2.11 Especificación de los requisitos de software.

- **Gestionar plantilla UM**

R1. Adicionar plantilla UM

R2. Eliminar plantilla UM

R3. Modificar plantilla UM

- **Gestionar ubicación de la tropa**

R4. Crear ubicación de una tropa

R5. Modificar ubicación de una tropa

- **Consultar guardia combativa**

- **Gestionar misiones**

R6. Adicionar misiones

R7. Modificar misiones

R8. Editar misiones

R9. Duplicar Misiones

- **Calcular datos de la misión**

- **Gestionar Enviar Misiones**

R10. Adicionar Misiones a Enviar

R11. Modificar Misiones a Enviar

R12. Configurar Misiones a Enviar

R13. Eliminar Misiones a Enviar

- **Gestionar Nomencladores**

R10. Adicionar Nomencladores

R10. Modificar Nomencladores

R10. Cancelar Nomencladores

- **Recibir misión**

2.12 Requerimientos no funcionales.

Usabilidad

El sistema debe poder ser usado por cualquier persona que tenga conocimientos básicos de informática.

El *software* contará con una ayuda, que se mantendrá visible para apoyar el empleo del sistema.

Confiabilidad

La información manejada por el sistema debe estar protegida de acceso no autorizado y de divulgación.

Rendimiento

Los tiempos de respuesta y velocidad de procesamiento de la información serán relativamente rápidos.

Soporte.

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

Para el cliente: Se requiere que esté instalado un navegador que interprete Javascript.

Restricciones de diseño

El producto de *software* final debe diseñarse sobre una arquitectura modelo-vista-controlador.

Emplear los estándares establecidos (diseño de interfaces, base de datos y codificación)

Requerimientos de ayuda y documentación

Debe contar con una ayuda que describa cómo funciona el sistema, también estará disponible en cada una de las interfaces, de esta forma, los usuarios tendrán conocimiento de las funcionalidades del mismo y hacer un mejor uso de estas.

Interfaz

La interfaz debe ser lo más sencilla posible, para que pueda ser manejada por cualquier tipo de usuario

Debe presentar una buena combinación de colores

Emplear los estándares establecidos (diseño de interfaces, base de datos y codificación)

Portabilidad

El sistema debe ser multiplataforma haciéndose énfasis en la plataforma Linux.

Políticos Culturales

El sistema solo podrá ser usado en territorio cubano y por las entidades del MINFAR.

El *software* debe ser en idioma español.

Debe respetar los términos usados en la especialidad.

Legales

El sistema se basará en el manual de normas y principios establecidos por el MINFAR.

Seguridad

Cada usuario podrá autenticarse, se le otorgarán atribuciones según sus funciones de trabajo.

Software

Navegador Mozilla Firefox.

Sistema operativo Windows XP o superior o Linux.

Hardware

En la computadora donde trabaje el cliente se necesita una Pentium III o superior con 128 MB de memoria RAM.

Donde se ubique el servidor es necesario que soporte grandes volúmenes de datos.

Como dispositivo externo es necesaria una impresora.

Aplicación de Estándares

Se aplicarán los estándares de ingeniería, codificación, diseño para la base de datos y mecanismos de diseño definidos por UCID.

2.13 Descripción de requisitos funcionales.

Requisito funcional

El requisito permite que se adicionen, modifiquen y elimine la plantilla UM.

Especificación del requisito Adicionar la plantilla UM.

Conceptos tratados	Conceptos	Atributos
Precondiciones	plantilla UM	número público, cargos
Descripción	<p data-bbox="495 1157 1539 1696">Precondiciones Debe existir la tropa.</p> <p data-bbox="495 1157 1539 1696">Pre-requisito No procede.</p> <ol data-bbox="544 1157 1539 1696" style="list-style-type: none">1. En la página principal de Estructura y composición se muestra un mapa de Cuba y un árbol en el que aparecen las estructuras creadas.2. Se selecciona la opción "Adicionar".3. Se muestra la interfaz correspondiente.4. Se introducen los datos que se solicitan.5. Se presiona el botón Aplicar para introducir más de una plantilla UM.6. Se presiona el botón Aceptar.7. Se verifica que los campos obligatorios estén llenos.8. Se guardan los datos y se muestra el mensaje de información: "Se ha insertado satisfactoriamente".9. Se presiona el botón Aceptar.	
Validaciones	Verificar que los campos obligatorios estén llenos.	

Post-condiciones	Queda adicionada una plantilla UM de una unidad.
Post-requisito	No procede.

Las restantes descripciones de los requisitos funcionales se encuentran en el [Anexo B](#)

2.14 Conclusiones

Con la realización del presente capítulo se ha realizado un estudio de los procesos de negocio existentes, para llevar a cabo una descripción de los mismos, así como un examen de los conceptos del dominio. Se efectuó un análisis de los requisitos funcionales y no funcionales del sistema para poder pasar a determinar la realización del análisis y diseño del sistema que se llevará a cabo en el próximo capítulo.

Capítulo 3. Diseño del sistema.

3.1 Introducción

En este capítulo se describe de manera general la solución que se presenta. Para ello se construirá el modelo de diseño, que constituye un plano bien cercano a la implementación, y contribuirá a una arquitectura estable y sólida. Como parte del modelo del diseño se hará una propuesta de los diagramas de clases del diseño basándose en la implementación que realiza SAUXE del patrón de arquitectura MVC.

3.2 Diseño

El Modelo de Diseño, es un modelo de objetos que describe la realización física de los procesos, centrándose en como los requisitos funcionales y no funcionales tienen impacto en el sistema a desarrollar. El diseño es el centro de atención al final de la fase de elaboración y el comienzo de las iteraciones de construcción.

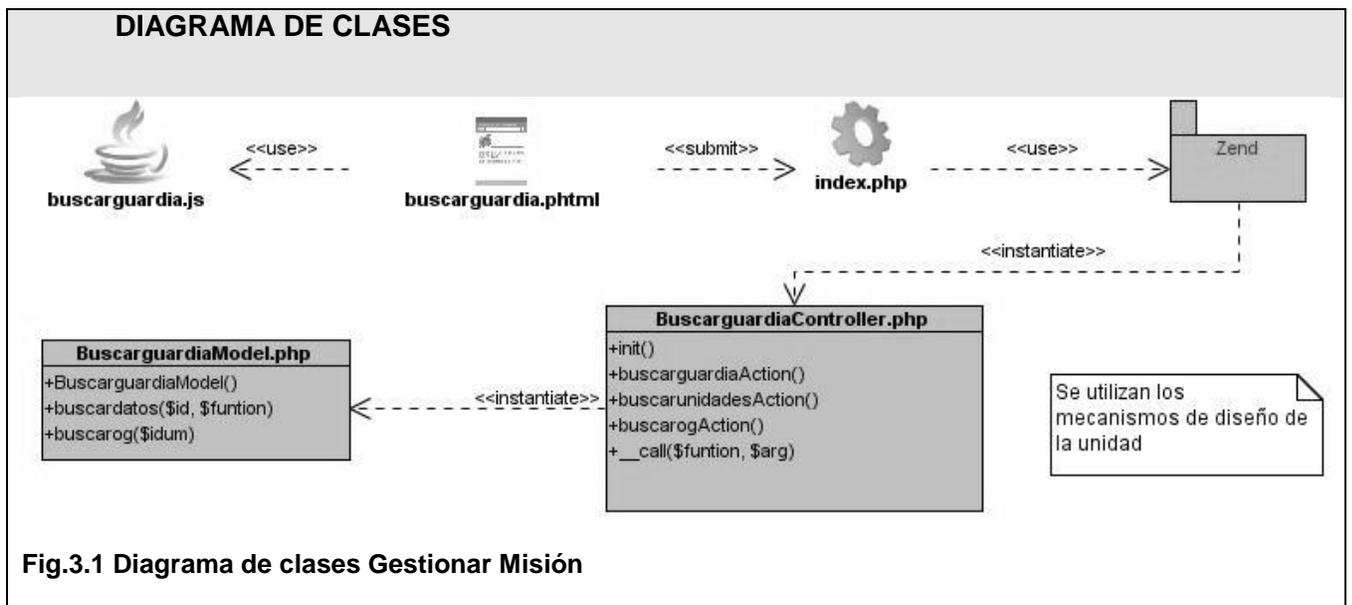


Fig.3.1 Diagrama de clases Gestionar Misión

Los restantes diagramas de clases se encuentran en [Anexo C](#).

DIAGRAMA DE INTERACCIÓN (Secuencia)

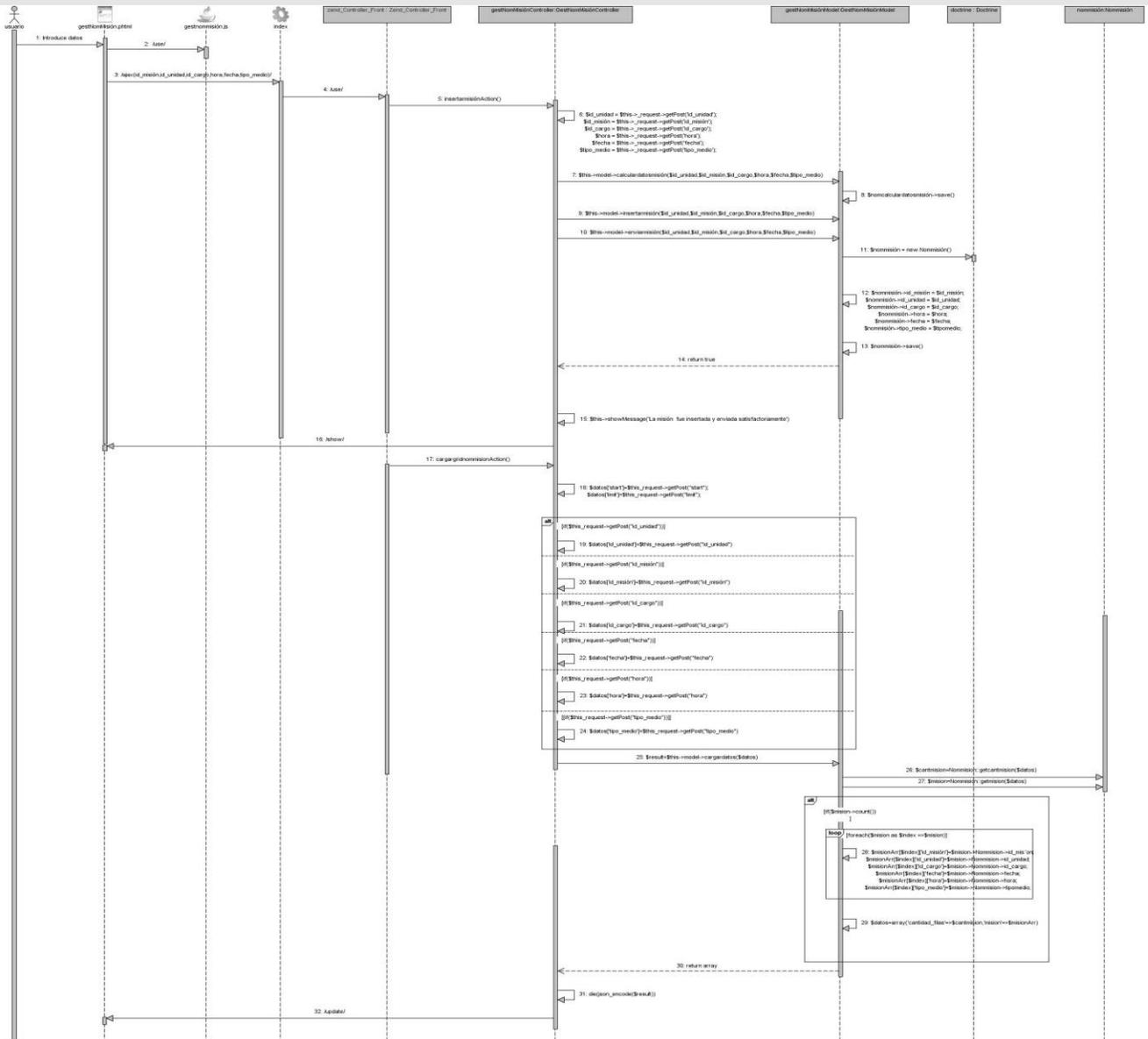


Fig. 3.2 Diagrama de Secuencia Insertar Misión

Los restantes diagramas de secuencia se encuentran en el [Anexo D](#)

3.5 Descripción de las clases

Controladoras

Las clases “Controller” son aquellas que dirigen el flujo del control de la aplicación. Estas se usan para encapsular el control de un requisito en concreto, regulando las actividades de los objetos que implementan la funcionalidad del proceso, por lo que definen el flujo de control y las transacciones dentro de un proceso delegando el trabajo a otros objetos.

En la siguiente tabla se muestra la clase controladora GestMisiónController que tiene como función capturar los eventos relacionados con la gestión de las misiones y la comunicación con la clase del negocio GestMisionModel.

Nombre: GestMisionController	
Tipo de clase: Controladora	
Atributo	Tipo
Para cada responsabilidad:	
Nombre:	Descripción:
init()	Constructor de la clase.
gestionarmisionAction()	Gestionar una misión
insertarmisionAction()	Insertar una misión
enviarmisionAction()	Enviar una misión.
recibirmisionAction()	Recibir una misión
listarmisionAction()	Listar las misiones

Modelo

Las clases modelo de la aplicación no guardan mucha información del estado por sí mismas, pero trabajan en la ejecución de tareas complejas.

En la siguiente tabla se muestra la clase modelo GestMisionModel que tiene como función el proceso que se realiza para manejar una misión como bien su nombre lo indica, que consiste en crear una misión enviarla y recibirla.

Nombre: GestMisionModel	
Tipo de clase: Modelo de la aplicación	
Atributo	Tipo
Para cada responsabilidad:	
Nombre:	Descripción:
Insertarmision(id_unidad,id:cargo,idmision,fecha,hora, tipo_medio)	Inserta una misión
Enviarmision(id_unidad,id:cargo,idmision,fecha,hora, tipo_medio)	Enviar una misión
Recibirmision(id_unidad,id:cargo,idmision,fecha,hora, tipo_medio)	Recibe una misión
Listarmisión(id_unidad,id:cargo,idmision,fecha,hora, tipo_medio)	Lista las misiones

Entidad

Estas clases modelan información que poseen una larga vida y que a menudo son persistentes. La fuente principal de obtención son las clases entidades del negocio y el glosario de términos elaborado a lo largo de la investigación. Se encargan de modelar la información del sistema y el comportamiento asociado a una información.

En la siguiente tabla se muestra la clase modelo del dominio que tiene como función el acceso a la base de datos para la gestión de las misiones

Nombre: GestMisionModel	
Tipo de clase: Modelo de la aplicación	
Atributo	Tipo
Para cada responsabilidad:	
Nombre:	Descripción:
Insertarmision(id_unidad,id:cargo,idmision,fecha,hora, tipo_medio)	Inserta una misión
Enviarmision(id_unidad,id:cargo,idmision,fecha,hora, tipo_medio)	Enviar una misión

Recibirmisión(id_unidad,id:cargo,idmision,fecha,hora,tipo_medio)	Recibe una misión
Listarmisión(id_unidad,id:cargo,idmision,fecha,hora,tipo_medio)	Lista las misiones

Las restantes descripciones de clases se encuentran en el [Anexo E](#)

3.6 DIAGRAMA ENTIDAD RELACIÓN

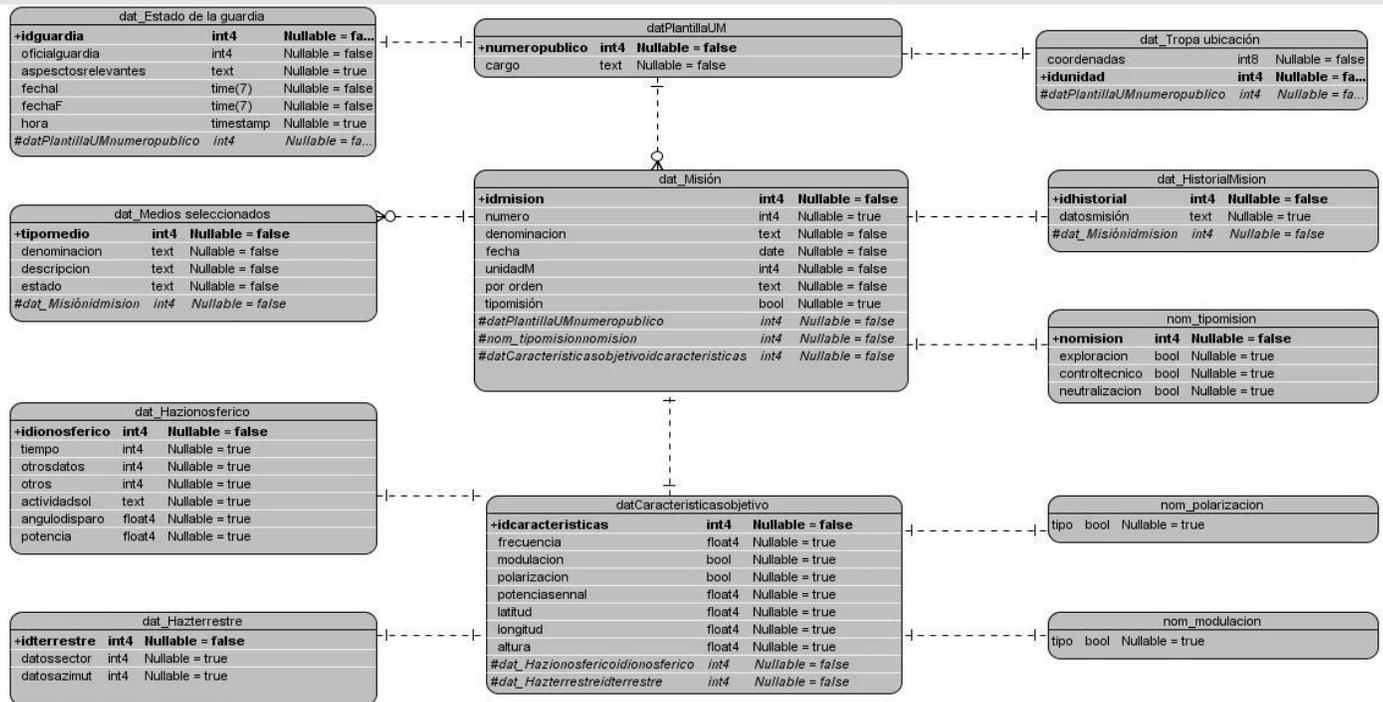


Fig.3.3 Diagrama entidad-relación

Teniendo en cuenta que la base de datos que se utiliza en la aplicación es uno de los elementos principales se hará una descripción de las tablas que esta posee y sus atributos. A pesar de esto se ha

presentado anteriormente el Diagrama Entidad Relación (DER) para una mejor comprensión del módulo y su BD.

3.7 Descripción de las tablas

Nombre: Misión		
Descripción: Datos que se deben requerir para crear y enviar una misión		
Atributo	Tipo	Descripción
Id_mision	entero	identifica a la misión
numeropublico	entero	identifica a la unidad
cargo	entero	identifica el cargo
fecha	date	Fecha en que se va a realizar la misión
pororden	char	Identifica el que envió la orden
estado	char	Estado en el que se encuentra

Las restantes descripciones de clases se encuentran en el [Anexo F](#)

3.8 Mecanismos de Diseño [25]

Una de las principales preocupaciones de los diseñadores de hoy día es como reutilizar modelos ya elaborados, al igual que los desarrolladores desean siempre reutilizar código. El termino de Mecanismos de Diseño fue introducido en (Documentación del Proceso Unificado, 2003) y ha sido utilizado con el objetivo de simplificar los diagramas de clases. Cada diseñador establece sus propios mecanismos de diseño y es el total responsable de sus modelos y su forma siempre y cuando no viole los patrones y estilos seleccionados. Los mismos reportan beneficios para al menos 3 propósitos:

- Mantener la homogeneidad en el diseño.
- Reutilizar soluciones anteriormente probadas.
- Reutilizar documentación.

A continuación, haciendo uso de este artefacto, se detallan los componentes a utilizar en cada parte de MVC.

Arquitectura. Vista. Uso de EXTJS [25]

Es un *framework* para *JavaScript* muy utilizado en el desarrollo de aplicaciones Web con AJAX. Tiene una librería inmensa que permite configurar las interfaces Web de manera semejante a aplicaciones desktop. Tiene incluidos la mayoría de los controles de los formularios Web incluyendo Grids para mostrar datos y elementos semejantes a la programación desktop como los formularios, paneles, barras de herramientas, menús y muchos otros. Dentro de su librería de componentes incluye componentes para el manejo de datos, lectura de XML, lectura de datos JSON e implementaciones basadas en AJAX. Presenta el uso de JavaScript con una programación orientada a objetos.

Clase	Descripción [35]
ext-base	Encargada del manejo de las solicitudes y respuestas, trabajo con ajax y manejo de componentes de EXT. Esta incluida en el paquete original.
ext-all	Es la encargada de la creación de los componentes visuales de la vista. Esta incluida dentro de las clases que trae EXT JS.
Vista	Representa la vista que se muestra al usuario
js_vista	Fichero js con las funciones Java Script asociadas a la vista. Aquí se establece la referencia a las clases de EXT.

Arquitectura. Controlador. Uso de EXTJS [25]

Son las clases que gestionan el manejo de la lógica del negocio. Por lo general incluyen las restricciones y validaciones fundamentales determinadas por las reglas. Para un caso de uso puede representarse una clase controladora o más.

Uso de Zend Framework [25]

Zend Framework es uno de los más utilizados para PHP y utiliza el estilo MVC como base de su funcionamiento. Es fácilmente integrable a las aplicaciones debido a su composición y a que contiene diferentes clases de gran utilidad, como por ejemplo en la búsqueda dinámica de ficheros a incluir o utilizar. Cuenta con un importante mecanismo de manejo de controladores y vistas por lo que se propone tenerlo en cuenta para el diseño de estos dos componentes de la arquitectura.

Elementos Arquitectónicamente significativos de Zend Framework [25]

Los frameworks, como aplicaciones de *software* que son, deben cumplir con los requerimientos de arquitectura semejante al resto. En este caso se encuentran los patrones de diseño y Zend Framework incluye en su implementación algunos de ellos:

- **Vista:** Implementa el patrón Decorator en la clase `Zend_View`, encargada de asignarle responsabilidades a objetos de manera dinámica y configurarlos con nuevos atributos.
- **Controlador:** *Zend Framework* tiene implementado el patrón *Front Controller* que implica que todas las solicitudes son dirigidas a un único script PHP que se encarga de instanciar al controlador frontal y redirigir las llamadas. Además, tiene una instancia única del controlador frontal disponible mediante el patrón *Singleton* para lograr una vía de entrada única a las solicitudes.
- **Modelo:** ZF provee una API para el acceso a dato conformada por un conjunto de clases que implementan los patrones *Factory*, *Table Data Gateway* y *Row Data Gateway*.

Por supuesto todos los patrones de diseño anteriormente mencionados implican características y comportamiento específico en los componentes de la arquitectura por lo que deben ser tenidos muy en cuenta.

Modelo. Uso de Doctrine PHP [25]

Doctrine es un potente y completo sistema ORM (*object relational mapper*) para PHP 5.2+ con un DBAL (*database abstraction layer*) incorporado. Entre muchas otras cosas tienes la posibilidad de exportar una base de datos existente a sus clases correspondientes y también a la inversa, es decir, convertir clases (convenientemente creadas siguiendo las pautas del ORM) a tablas de una base de datos. Su principal ventaja radica en poder acceder a la base de datos utilizando la programación orientada a objetos (POO)

debido a que doctrine utiliza el patrón *Active Record* para manejar la base de datos, tiene su propio lenguaje de consultas y trabaja de manera rápida y eficiente. Es fácilmente integrado a los principales *frameworks* de desarrollo utilizados actualmente, por lo que se propone su uso.

3.9 Seguridad en el Módulo

Autenticación y autorización

Cuando se diseña una aplicación Web es importante definir los aspectos relacionados con la seguridad de la misma. En este contexto es importante prestar atención a la autenticación y gestión de credenciales. Una vez que un usuario se autentica, se le asignan los privilegios necesarios (credenciales) para que pueda utilizar las funcionalidades que brinda la aplicación según el rol que desempeñe.

A partir de los requisitos funcionales y no funcionales, y tal como se había planteado, la IW poseerá un tipo de usuario que será el encargado de chequear el estado de la guardia en cada pequeña unidad y que será capaz de, auxiliado por el módulo, elaborar y enviar una misión. De este modo solo será necesario definir la seguridad y la credencial o credenciales que se requieran para la ejecución de cada una de las acciones, pues la responsabilidad de chequear que un usuario esté correctamente autenticado y posea los privilegios necesarios será del marco de trabajo

3.10 Interfaz [26]

- La interfaz corresponderá con las características, metas y nivel de experiencia de los usuarios de la aplicación, siempre que no se atente contra las pautas para el diseño.
- Todos los formularios y componentes visuales se optimizarán para una resolución de pantalla de 800 * 600.
- La información estará libre de errores gramaticales, ortográficos y tipográficos.
- La taxonomía de los títulos encima de los controles estará según la estipulada, primera letra con mayúscula, el resto con minúscula.
- En controles de tipo [*input*, *select*, *text-área*] cambiará el color de fondo indicando que el cursor está situado en el para modificar los datos. Una vez que el cursor deja de estar situado sobre ellos, regresarán a su estado normal. (Garantizado por Ext.).

- En controles de tipo *button* cambiará el color de fondo indicando que el cursor está situado en el para presionar. Una vez que el cursor deja de estar situado sobre ellos, regresarán a su estado normal. El cursor mantendrá la forma de puntero. (Garantizado por Ext.).
- Los iconos se corresponderán con las tareas a realizar.
- Los botones en todas las barras de herramientas se corresponderán con los comandos de un menú. (Garantizado por Ext.).
- En caso necesario un comando del menú tendrá una secuencia de “hot-keys” (teclas de acceso rápido), que lo invocarán cuando sean activadas por el usuario.
- En las cajas de diálogo "tabuladas", no se emplearán abreviaturas en los nombres de las lengüetas o pestañas.
- En cajas de diálogo "tabuladas", las lengüetas o pestañas se podrán identificar con combinaciones de “hot-keys” apropiadas para su tarea. (Opcional).
- En las cajas dialogadas no existirán “hot-keys” duplicadas.
- Las lengüetas serán colocadas horizontalmente.
- Se propiciará el uso apropiado de la tecla Escape. (opcional).
- Un botón de comando se habilitará cuando la acción se corresponda con su uso y no en otras ocasiones, donde se pone en peligro la información por un posible error de utilización.
- Los nombres de los botones de comandos no serán abreviaturas y tendrán un significado claro para los usuarios del sistema.
- Los botones de comando serán de tamaño y forma similares. (Garantizado por Ext.).
- Los nombres de los botones de opción (botón de radio o *radiobutton*) y de las cajas de opción no serán abreviaturas.
- El usuario siempre tendrá a la mano la posibilidad de "Ayuda" de la interfaz con la que interactúa.
- El sistema asistirá al usuario de forma efectiva para hacer las tareas en caso de que no haya una única manera de hacerlas.
- Solo se presentará al usuario la información que realmente necesita.
- Existirán diferentes niveles de ayuda.

- En un formulario los controles estarán a 10 px de margen en la horizontal uno con otro, y a 10 px de margen en la vertical. Los tamaños variarán según la especificación del tipo de dato y la cantidad de caracteres requeridos, además, se establecerá una alineación uniforme.
- El *label* siempre se colocará encima del control, excepto en pantallas de autenticación, en barras de herramientas, y otros como el *radiobutton*, y el *checkbox* que se colocará seguido o delante del control a la parte derecha o izquierda.
- Pantallas de autenticación: Alineado a la izquierda.
- Para modificar valores que se muestran en un *Grid*, se empleará la barra de herramientas, donde al presionar algún botón se abrirá una ventana (empleando el componente que corresponda), y se realizarán los cambios pertinentes. Siempre que el usuario estime conveniente, se brindará la posibilidad de editar en el propio *grid*.
- Para editar en un *grid* utilizara el doble clic, si el *grid* no es editable, al dar doble clic sobre una *tupla*, habilitara una ventana para modificar el registro siempre que este lo requiera.
- En todos los casos el idioma definido por el sistema se empleará en todos los textos. Si es español no se muestran palabras en inglés u otro, y de igual forma para los demás.
- La ventana que se emplea para adicionar un elemento siempre comenzará el título con la palabra “Adicionar seguida por el elemento específico” Ej. Adicionar un estudiante.
- La ventana que se emplea para modificar un elemento siempre comenzará el título con la palabra “Modificar seguida por el elemento específico” Ej. Modificar un estudiante.
- La fecha siempre se utilizará en formato de d/m/Y (01/10/2008).

3.11 Concepción de la ayuda

Un elemento importante de cualquier aplicación es la ayuda que los usuarios necesitan para hacer uso de la misma. En este sentido, se presenta un manual de usuario que aparecerá en la aplicación con las situaciones que pueden ocurrir y la explicación pertinente para que el usuario sea capaz de entenderla y llevar a cabo la acción que se proponía.

3.12 Conclusiones

La generación de algunos artefactos relacionados con el flujo de análisis y diseño, teniendo en cuenta la arquitectura MVC que el marco de trabajo establece, permitió obtener una mayor comprensión de la aplicación y definir los principios que guiarán la implementación y organización de la misma. Además, se analizaron las estrategias a seguir para dar respuesta a los requisitos no funcionales relacionados con la seguridad, se obtuvo un prototipo de interfaz y las pautas de diseño para la misma, así como la concepción de la ayuda de la aplicación.

Capítulo 4. Implementación y prueba.

4.1 Introducción

En el presente capítulo se trata el flujo de trabajo de implementación y pruebas, este flujo es en el que se construye el modelo necesario para desarrollar el proceso de implementación del sistema con los diagramas de componentes definidos. Además, se abordarán los temas relacionados con las pruebas aplicadas al *software*.

4.2 Diagramas de Componentes

El diagrama de Componentes muestra las organizaciones y dependencias lógicas entre componentes de *Software*, sean estos ficheros de código fuente, binarios o ejecutables. Los elementos de modelado que lo conforman son los componentes y paquetes y muestran la estructura del sistema en términos de implementación a un alto nivel [27].

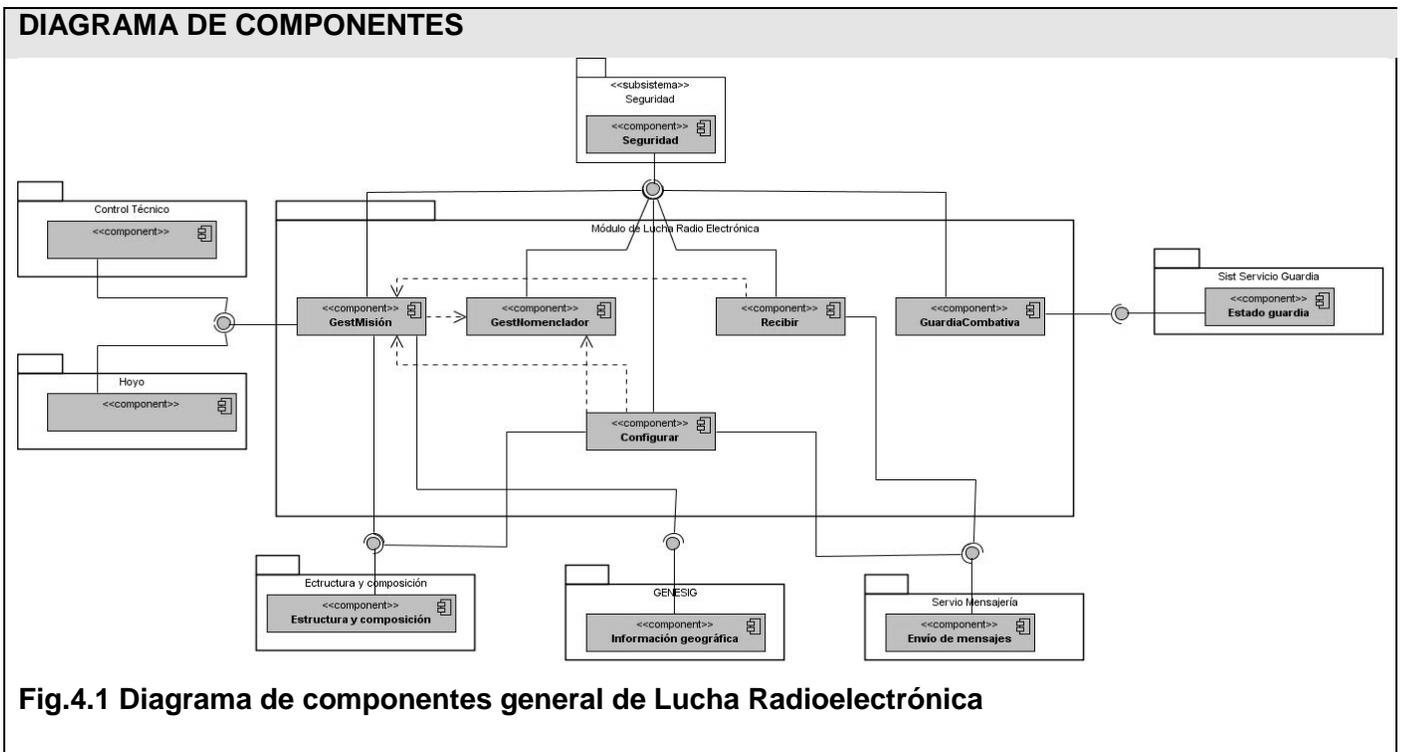


Fig.4.1 Diagrama de componentes general de Lucha Radioelectrónica

4.3 Matriz de Integración de Componentes Interna.

Componentes Internos					
Componentes Internos	Gestionar Misión	Guardia Combativa	Gestionar Nomenclador	Recibir Misión	Configurar Misión
Gestionar Misión	-	-	Tipo de nomenclador	Datos de la misión	Datos de la misión
Guardia Combativa	-	-	-	-	-
Gestionar Nomenclador	Tipo de nomenclador	-	-	-	-
Recibir Misión	Datos de la misión	-	-	-	-
Configurar Misión	Datos de la misión	-	-	-	-

4.4 Matriz de Integración de Componentes Externa.

Componentes externos							
Componentes Externos	Estructura y composición	GENESIG	Sistema servicio guardia	Servicio mensajería	Sistema seguridad	Control técnico	Hoyo
Estructura y composición	-	Estructura y ubicación de	-	Estructura y	Nivel de acceso	-	-

		las tropas		ubicación de las tropas			
GENESIG	Ubicación de las tropas	-	-	-	Nivel de acceso	-	-
Sistema servicio guardia	-	-	-	-	Nivel de acceso	-	-
Servicio mensajería	Estructura y ubicación de las tropas	-	-	-	Nivel de acceso	-	-
Sistema seguridad	Nivel de acceso	Nivel de acceso	Nivel de acceso	Nivel de acceso		Nivel de acceso	Nivel de acceso
Control técnico	-	-	-	-	Nivel de acceso	-	-
Hoyo	-	-	-	-	Nivel de acceso	-	-

4.5 Modelo de prueba

El desarrollo de sistemas de *software* implica una serie de actividades de producción en las que las posibilidades de que aparezca el fallo humano son enormes. Los errores pueden empezar a darse desde el primer momento del proceso en el que los objetivos pueden estar especificados de forma errónea o imperfecta así como en posteriores pasos de diseño y desarrollo. Debido a la imposibilidad humana de trabajar y comunicarse de forma perfecta el desarrollo del *software* ha de ir acompañado de una actividad que garantice la calidad [28].

Las pruebas del *software* son un elemento crítico para la garantía de la calidad del *software* y representa una revisión final de las especificaciones, del diseño y de la codificación [28]. Existen diferentes estrategias, niveles, tipos y métodos de pruebas. Para realizar estas pruebas se diseñan casos de pruebas. Las estrategias que se pueden seguir son varias, entre las que está incluir técnicas de detección de errores aplicados a los modelos de Análisis y Diseño, cambiar la estrategia para las pruebas de unidad e integración.

El diseño de casos de prueba debe tener en cuenta las características propias del *software*. Los niveles se aplican para diferentes objetivos, entre ellos están las pruebas de desarrollador, independiente, de Unidad, de Integración, de sistema de aceptación. Los tipos de pruebas son varios, pueden ser de funcionalidad, usabilidad, fiabilidad, rendimiento, y soportabilidad. Los métodos pueden ser pruebas de caja negra y pruebas de caja blanca.

Cualquier producto de ingeniería puede probarse de una de estas formas:

(1) conociendo la función específica para la que fue diseñado el producto, se pueden llevar a cabo pruebas que demuestren que cada función es completamente operativa y al mismo tiempo buscando errores en cada función,

(2) conociendo el funcionamiento del producto, se pueden desarrollar pruebas que aseguren que todas las piezas encajan, o sea, que la operación interna se ajusta a las especificaciones y que todos los componentes internos se han probado de forma adecuada. El primer enfoque de pruebas se denomina prueba de caja negra y el segundo, prueba de caja blanca [28]. Para la validación del sistema propuesto se llevará el método de caja negra.

4.6 Pruebas de caja negra

La prueba de caja negra se refiere a las pruebas que se llevan a cabo sobre la interfaz del *software*. O sea, los casos de pruebas pretenden demostrar que las funciones del *software* son operativas, que la entrada se acepta de forma adecuada y que se produce un resultado correcto [28].

Permite obtener conjuntos de condiciones de entrada que ejerciten completamente todos los requisitos funcionales de un programa. Intenta encontrar errores de las siguientes categorías: funciones incorrectas o ausentes; errores de interfaz, en estructuras de datos o en acceso a bases de datos externas; errores de rendimiento, de inicialización y de terminación [28].

Gestionar Misión

Condiciones de ejecución

- El usuario debe tener los permisos necesarios para realizar las operaciones.

Nombre del requisito	Descripción general	Escenarios de pruebas	Flujo del escenario
Gestionar Misión	Permitir que el Jefe de Turno pueda insertar los datos de las misiones.	EP 1.1: Adicionar Datos de la misión correctamente	<ul style="list-style-type: none">• Se da clic en el botón “Inicio”, se selecciona la opción de “Misiones”, después se da clic en la opción “Adicionar”.• El sistema muestra una interfaz con los datos a llenar de la misión.• Se llenan los campos correctamente y selecciona la opción de “Haz terrestre o Haz ionosférico”.• El sistema muestra una interfaz con los datos restantes a llenar de la misión.• Se llenan los campos correctamente y selecciona la opción de “Aceptar”.• El sistema muestra un mensaje indicando que la operación se realizó de manera correcta.

		<p>EP 1.1: Adicionar Datos de la misión incorrectamente</p>	<ul style="list-style-type: none"> • Se da clic en el botón “Inicio”, se selecciona la opción de “Misiones”, después se da clic en la opción “Adicionar”. • El sistema muestra una interfaz con los datos a llenar de la aplicación. • Se llenan los campos incorrectamente y/o deja campos obligatorios vacíos. • El sistema marca los datos incorrectos y/o vacíos. • El usuario presiona el botón Aceptar. • El sistema no permite insertar los datos de la aplicación.
		<p>EP 1.3: Cancelar operaciones.</p>	<ul style="list-style-type: none"> ▪ Se da clic en el botón “Inicio”, se selecciona la opción de “Misión”, después se da clic en la opción “Adicionar”. ▪ El sistema muestra una interfaz con los datos a llenar de la aplicación. <p>El usuario llena o no los campos y presiona el botón “Cancelar”.</p>

No	Nombre de campo	Clasificación	Puede ser nulo	Descripción
1	frecuencia	float	NO	El campo acepta números.

2	modulación	bool	NO	El campo acepta un valor de la lista.
3	polarización	bool	NO	El campo acepta un valor de la lista.
4	latitud	float	NO	El campo acepta números.
5	longitud	float	NO	El campo acepta números.
6	altura	float	NO	El campo acepta números.
7	potencia	float	NO	El campo acepta números.

Juegos de datos a probar

Id del escenario	Escenario	Frecuencia	Modulación	Polarización	Potencia	Latitud	Longitud	Altura	Respuesta del sistema	Resultado de la prueba
-------------------------	------------------	-------------------	-------------------	---------------------	-----------------	----------------	-----------------	---------------	------------------------------	-------------------------------

EP 1.1	Adicionar datos de la misión correctamente	63.1	modulación1	Polarización2	50.63	124.6	145.98	3254.12	El sistema muestra el siguiente mensaje: Los datos se insertaron correctamente.	Satisfactorio
EP 1.2	Adicionar datos de la misión incorrectamente	vacío	seleccionar	seleccionar	máxima	norte	este	vacío	No permite adicionar aplicaciones	Insatisfactorio
EP 1.3	Cancelar operaciones.	-	-	-	-	-	-	-	Se cancela la operación y se cierra la ventana	Insatisfactorio

4.7 Conclusiones

Tras el flujo de implementación y prueba el sistema quedó desarrollado. En los diagramas generados pudo ilustrarse la relación entre los principales componentes del sistema y como estará distribuido este. Las pruebas realizadas permitieron aumentar la calidad final de la solución. La aplicación final aprovechó las ventajas del marco de trabajo y quedó lista para integrarse con la especialidad.

Conclusiones

Con el desarrollo del módulo de apoyo a la toma de decisiones de Lucha Radio Electrónica se llegó a las siguientes conclusiones:

- Se elaboró la fundamentación teórica y se definió toda la problemática de la situación, lo que permitió insertarse en el problema que se planteaba, así como su objeto de estudio y campo de acción, aspectos fundamentales a la hora de limitar el trabajo a realizar y hasta qué punto consumarlo.
- El análisis de los principales sistemas de apoyo a la toma de decisiones, como parte del estudio realizado al estado del arte, permitió reformar la visión que se poseía acerca de los sistemas de apoyo a la toma de decisiones en Cuba y el mundo, aseverando la no existencia de un módulo capaz de aportar las funcionalidades necesarias requeridas en la especialidad de LRE.
- A partir de la descripción del problema se definieron y describieron los procesos del negocio, así como su modelación, dando una vista global del Módulo de Apoyo a la Toma de Decisiones de la especialidad de lucha radioelectrónica.
- Posteriormente de realizarse la definición de los requisitos funcionales y no funcionales del sistema, se obtuvo una perspectiva más completa de las funcionalidades con que debía cumplir el Módulo así como de las cualidades o condiciones que se comprometían a estar presentes en la realización del mismo.
- La generación de artefactos relacionados con el flujo de análisis y diseño, teniendo en cuenta la arquitectura MVC que el marco de trabajo establece, permitió obtener una mayor comprensión de la aplicación y definió los principios que guiaron la implementación del sistema.
- Como conclusión general, puede afirmarse que se le dieron cumplimiento a todos los objetivos planteados al inicio del trabajo, incluyendo la implementación del módulo de Apoyo a la Toma de Decisiones del Sistema de Mando y Dirección de la especialidad de Lucha Radioelectrónica y se verificó además la validez de la idea a defender materializada en la solución obtenida.

Recomendaciones

Por la propia naturaleza cambiante de las aplicaciones Web y de los sistemas informáticos en general, y para tratar de aumentar las ventajas de uso del módulo de LRE se recomienda:

Estudiar la posibilidad de que el oficial de guardia que tiene la responsabilidad de tomar las decisiones respecto a las misiones, apoyado en el módulo de LRE, sea capaz de enviarle misiones directamente a los medios de lucha, dígase antenas u otro, de modo que se cumpla la tarea de forma automática.

Trabajar en la integración del sistema HOYO, REALIZACIÓN, Servicio de guardia y Control Técnico con el módulo de apoyo a la toma de decisiones de LRE.

Bibliografía

Maestros de la web. ¿Qué es CSS? Ponencia sobre Gentoo Linux en MadridWeb Revolution en Rep. Dominicana con Christian Van Der Henst S. Consultado enero 2010. Disponible: <http://www.maestrosdelweb.com/editorial/introcss/>

Proceso de desarrollo de *software* de la UCID.

UML Resource Page [Internet]. Disponible en: <<http://www.uml.org/>> [Citado 11 febrero 2010].

Curso práctico de Modelado de Negocios con UML y BPMN. Consultado: enero 2010. Disponible: <http://www.milestone.com.mx/CursoModeladoNegociosBPMN.htm>

Javier J. Gutiérrez. ¿qué es un *framework* web? 2006. Consultado enero2010.

Disponible: http://www.lsi.us.es/~javierj/investigacion_ficheros/Framework.pdf

Doctrine - PHP Object Relational Mapper. [En línea] [Citado: enero de 2010.] <http://www.doctrine-project.org/>. Doctrine ORM for PHP. Doctrine 1.2 License: Creative Commons Attribution-Share Alike 3.0 Unported License Version: manual-1.2-en-2010-02-16

PG AdminII, Guia Ubuntu, 10-3-2010.

Disponible: http://www.guia-ubuntu.org/index.php?title=PgAdmin_III.

Agil Unified Process. [En línea] <http://www.ambyssoft.com/unifiedprocess/agileUP.html>.

DebShop.ZendFramework. Consultado enero 2010.

Disponible: <http://softwareparaubuntu.blogspot.com/2009/10/zend-framework-193pl1.html>

Proenza Arias Yuniel. profesorProgramación.Atix08.2008. Diseño Avanzado de Aplicaciones Web. EXT – Zend Framework y Doctrine.

Estándar de diseño de interfaces para las aplicaciones de gestión. UCID. 2010

Ana Fernández Vilas (2001). Diagrama de Componentes [Internet]. Disponible en: <<http://www-gris.det.uvigo.es/~avilas/UML/node49.html>> [Citado 15 mayo 2010].

Roger S. Pressman. Ingeniería de *software*, un enfoque práctico, *chapter* Técnicas de prueba del *software*, pages 281–299. McGraw-Hill, 2001

TOVAR Adriana María. TOMA DE DECISIONES. INGENIERÍA DE SISTEMAS III. TOMA DE DECISIONES. Onceava edición, 1999. HAROLD KOONTZ y HEINZ WEIHRICH “Administración una Perspectiva Global” MCGRAW- HILL INTERAMERICANA DE EDITORES, S.A.1999.

Salvador Lerache Jorge, García Martínez Ramón. Sistema Experto Aplicado al Control del Espacio Aéreo. Ingeniería del *Software* e Ingeniería del Conocimiento. Google académico. 2010, 17.

R.Juanes², A. Viña, F. Bellas, N. Rodríguez, A.Molano³. Concurrencia y Tiempo-Real en los Sistemas Distribuidos de Mando y Control. Departamento de Electrónica y Sistemas. Google académico. 2010

Paul W. Phister Jr., PE, PhD Igor G. Plonisch*. Empleos de la Informática en el Ámbito Militar. Dirección de Información del Laboratorio de Investigaciones de la Fuerza Aérea, Rome, New Cork. Google académico, 23 August 04. 2010.

Cadena Sosa, Yovani Fabio, Sistemas expertos en la planificación de operaciones tácticas militares. Caso: Fuerza Naval. Servicios I digital Catálogo digital. <http://hdl.handle.net/123456789/122>.2010 Google académico.

Picazza.J.M. MELISIS, un marco de trabajo para la construcción de sistemas de ayuda a la toma de decisiones en problemas de monitorización. Departamento de lenguajes y sistemas informáticos. Google académico, 2010. 11 páginas

Gómez López, Laira; Machado Rodríguez Leyanny. Sistema Informático de Mando y Estado Mayor. Módulo Preparación de Acciones Combativas. Análisis y diseño. UCI, 2009.

Gerti Kappel et al. Web Engineering: The Discipline of Systematic Development of Web Applications,chapter An Introduction to Web Engineering, pages 65–67. John Wiley & Sons Ltd., 2006.

Richard Rosen Leon Shklar. Web Application Architecture: Principles, protocols and practices, chapter Introduction, pages 201–205. John Wiley & Sons Ltd., 2003.

Montesino Alfonso Yasmany, Parra Lubi Orestes. Implementación del submódulo de recuperaciones de información del módulo de recuperaciones dinámicas. Cadete, investigación, UCI, 2008.

Librosweb.es. Patrón MVC. Consultado enero 2010.

Disponible: http://www.librosweb.es/symfony_1_0/capitulo2/el_patron_mvc.html.

Capítulo II. Arquitectura de *software*. Consultado: enero 2010. Disponible: http://catarina.udlap.mx/u_dl_a/tales/documentos/lis/rivera_l_a/capitulo2.pdf

Derick Rethans Andi Gutmans, Stig Sather Bakken. PHP 5 Power Programming, chapter PHP 5 BasicLanguage, pages 10–13. PRENTICE HALL, Professional Technical Reference, 2005

M. A. Álvarez. ¿Qué es PHP?, page 87. 2004.

Agustín Dondo. ¿Por qué elegir PHP? [Internet]. Disponible en: <http://www.programacion.com/php/articulo/porquephp/> [Citado 7 febrero 2010].

Referencias bibliográficas

- 1) TOVAR Adriana María. TOMA DE DECISIONES. INGENIERÍA DE SISTEMAS III. TOMA DE DECISIONES. Onceava edición, 1999. HAROLD KOONTZ y HEINZ WEIHRICH “Administración una Perspectiva Global” McGRAW- HILL INTERAMERICANA DE EDITORES, S.A.1999.
- 2) Salvador Lerache Jorge, García Martínez Ramón. Sistema Experto Aplicado al Control del Espacio Aéreo. Ingeniería del *Software* e Ingeniería del Conocimiento. Google académico. 2010, 17.
- 3) R.Juanes2, A. Viña, F. Bellas, N. Rodríguez, A.Molano3. Concurrencia y Tiempo-Real en los Sistemas Distribuidos de Mando y Control. Departamento de Electrónica y Sistemas. Google académico. 2010
- 4) Paul W. Phister Jr., PE, PhD Igor G. Plonisch*. Empleos de la Informática en el Ámbito Militar. Dirección de Información del Laboratorio de Investigaciones de la Fuerza Aérea, Rome, New Cork. Google académico, 23 August 04. 2010.
- 5) Cadena Sosa, Yovani Fabio, Sistemas expertos en la planificación de operaciones tácticas militares. Caso: Fuerza Naval. Servicios I digital Catálogo digital. <http://hdl.handle.net/123456789/122>.2010 Google académico.
- 6) Picazza.J.M. MELISIS, un marco de trabajo para la construcción de sistemas de ayuda a la toma de decisiones en problemas de monitorización. Departamento de lenguajes y sistemas informáticos. Google académico, 2010. 11 páginas
- 7) Gómez López, Laura; Machado Rodríguez Leyanny. Sistema Informático de Mando y Estado Mayor. Módulo Preparación de Acciones Combativas. Análisis y diseño. UCI, 2009.
- 8) Gerti Kappel et al. Web Engineering: The Discipline of Systematic Development of Web Applications,chapter An Introduction to Web Engineering, pages 65–67. John Wiley & Sons Ltd., 2006.
- 9) Richard Rosen Leon Shklar. Web Application Architecture: Principles, protocols and practices, chapter Introduction, pages 201–205. John Wiley & Sons Ltd., 2003.

10) Montesino Alfonso Yasmany, Parra Lubi Orestes. Implementación del submódulo de recuperaciones de información del módulo de recuperaciones dinámicas. Cadete, investigación, UCI, 2008.

11) Librosweb.es. Patrón MVC. Consultado enero 2010.

Disponible: http://www.librosweb.es/symfony_1_0/capitulo2/el_patron_mvc.html.

12) Capítulo II. Arquitectura de *software*. Consultado: enero 2010. Disponible: http://catarina.udlap.mx/u_dl_a/tales/documentos/lis/rivera_l_a/capitulo2.pdf

13) Derick Rethans Andi Gutmans, Stig Sather Bakken. PHP 5 Power Programming, chapter PHP 5 BasicLanguage, pages 10–13. PRENTICE HALL, Professional Technical Reference, 2005

14) M. A. Álvarez. ¿Qué es PHP?, page 87. 2004.

15) Agustín Dondo. ¿Por qué elegir PHP? [Internet]. Disponible en: <http://www.programacion.com/php/articulo/porquephp/> [Citado 7 febrero 2010].

16) Maestros de la web. ¿Qué es CSS? Ponencia sobre Gentoo Linux en MadridWeb Revolution en Rep. Dominicana con Christian Van Der Henst S. Consultado enero 2010. Disponible: <http://www.maestrosdelweb.com/editorial/introcss/>

17) Proceso de desarrollo de *software* de la UCID.

18) UML Resource Page [Internet]. Disponible en: <http://www.uml.org/> [Citado 11 febrero 2010].

19) Curso práctico de Modelado de Negocios con UML y BPMN. Consultado: enero 2010. Disponible: <http://www.milestone.com.mx/CursoModeladoNegociosBPMN.htm>

20) Javier J. Gutiérrez. ¿qué es un *framework* web? 2006. Consultado enero2010.

Disponible: http://www.lsi.us.es/~javierj/investigacion_ficheros/Framework.pdf

- 21) Doctrine - PHP Object Relational Mapper. [En línea] [Citado: enero de 2010.] <http://www.doctrine-project.org/>. Doctrine ORM for PHP. Doctrine 1.2 License: Creative Commons Attribution-Share Alike 3.0 Unported License Version: manual-1.2-en-2010-02-16
- 22) PG AdminII, Guía Ubuntu, 10-3-2010. Disponible: http://www.guia-ubuntu.org/index.php?title=PgAdmin_III.
- 23) Agil Unified Process. [En línea] <http://www.ambyssoft.com/unifiedprocess/agileUP.html>.
- 24) DebShop.ZendFramework. Consultado enero 2010.
Disponible: <http://softwareparaubuntu.blogspot.com/2009/10/zend-framework-193pl1.html>
- 25) Proenza Arias Yuniel. profesor Programación. Atix08.2008. Diseño Avanzado de Aplicaciones Web. EXT – Zend Framework y Doctrine.
- 26) Estándar de diseño de interfaces para las aplicaciones de gestión. UCID. 2010
- 27) Ana Fernández Vilas (2001). Diagrama de Componentes [Internet]. Disponible en: <http://www-gris.det.uvigo.es/~avilas/UML/node49.html> [Citado 15 mayo 2010].
- 28) Roger S. Pressman. Ingeniería de *software*, un enfoque práctico, *chapter* Técnicas de prueba del *software*, pages 281–299. McGraw-Hill, 2001

Glosario de términos

TIC

Tecnología para la información y las comunicaciones

MINFAR

Ministerio de las Fuerzas Armadas revolucionarias

Lucha radioelectrónica

Enfrentamiento o combate que se realiza mediante la radioelectrónica

Radioelectrónica

Tecnología de comunicación entre dos dispositivos electrónicos.

LRE

Lucha radioelectrónica

SAD

Sistemas de apoyo a la toma de decisiones

Interfaz web

Es la parte de una aplicación que se encarga de interactuar con el usuario

Concurrencia

Que se realiza al mismo tiempo

Arquitectura web

La arquitectura de la información de un sitio web, como resultado de la actividad, comprende los sistemas de organización y estructuración de los contenidos, los sistemas de rotulado o etiquetado de dichos contenidos, y los sistemas de recuperación de información y navegación que provea el sitio web.

Tecnología

Tecnología es el conjunto de conocimientos técnicos, ordenados científicamente, que permiten construir objetos y máquinas para adaptar el medio y satisfacer las necesidades de las personas.

Lenguaje del lado del servidor

Lenguajes de programación que se enfocan en funcionalidades que no tienen que ver con la interfaz de usuario, están hechos pero el cliente no es capaz de ver el código ni como este influye en lo que está haciendo.

Lenguaje del lado del cliente

Lenguajes que su función principal es mejorar la interfaz del cliente, ayudar al usuario a navegar y realizar operaciones sobre la IW sin que sea engorroso.

Gestor de base de datos

Es como un almacén donde puedes guardar datos importantes que te hagan falta para realizar otras operaciones con posterioridad.

Metodologías de desarrollo

Plan que se lleva a cabo basado en un modelo o patrón por el que regirse para darle forma a la ingeniería de un software

Metodologías robustas o tradicionales

Metodología que requiere una gran cantidad de esfuerzo por parte del personal y entregables, se usa principalmente para proyectos grandes de larga duración.

Metodologías ágiles o ligeras

Metodología que no requiere mucho personal, aunque este debe estar bien experimentado en lo que se va a tratar, se usa para proyectos cortos.

Ciclo de vida de un proyecto

Todo el recorrido de un proyecto, dónde comienza y donde termina.

Herramientas

Es lo que se va a usar para realizar determinado objetivo, es en lo que se apoya para llevar a cabo un producto terminado.

Marco teórico

Todo lo que se va a usar para la confección y realización del software en cuestión.

Mapa de procesos

Esbozos dónde se puede ver el flujo que tendrán las actividades de cada proceso.

Descripción de los procesos

Detalles descritos específicos de cómo va funcionar el proceso.