

Universidad de las Ciencias Informáticas

Facultad 15



Título: Herramienta para la representación geoespacial de la información de los subsistemas de Reclutamiento y Reserva.

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

Autor: Alain Pérez Rodríguez.

Tutor: Ing. Yenisleidis Ayala Rosales.

Co-Tutor: Ing. Asdrubal Antonio Nicot García.

Ciudad de la Habana, Cuba

Junio 2010

“Año 52 de la Revolución”

DECLARACIÓN DE AUTORÍA

Declaro que soy el único autor de este trabajo y autorizo al Ministerio de las Fuerzas Armadas Revolucionarias (MINFAR) y a la Universidad de las Ciencias Informáticas (UCI) para que hagan el uso que estimen conveniente con este trabajo.

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

Alain Pérez Rodríguez.

Ing. Yenisleidis Ayala Rosales.

Firma del autor

Firma del tutor

Ing. Asdrúbal Antonio Nicot García.

Firma del Co-Tutor

DATOS DE CONTACTO

Tutor: Ing. Yenisleidis Ayala Rosales.

Breve Currículo:

Ingeniero en Ciencias Informáticas. Graduada en julio de 2007. Analista y Líder de gestión de la Línea Capital Humano de la Unidad de Compatibilización e Integración y Desarrollo de Software para la Defensa (UCID) - Universidad de la Ciencias Informáticas.

Correo Electrónico: yeayala@uci.cu

Años de graduado: 3

Años de experiencia en el tema: 4

Co-Tutor: Ing. Asdrúbal Antonio Nicot García.

Breve Currículo:

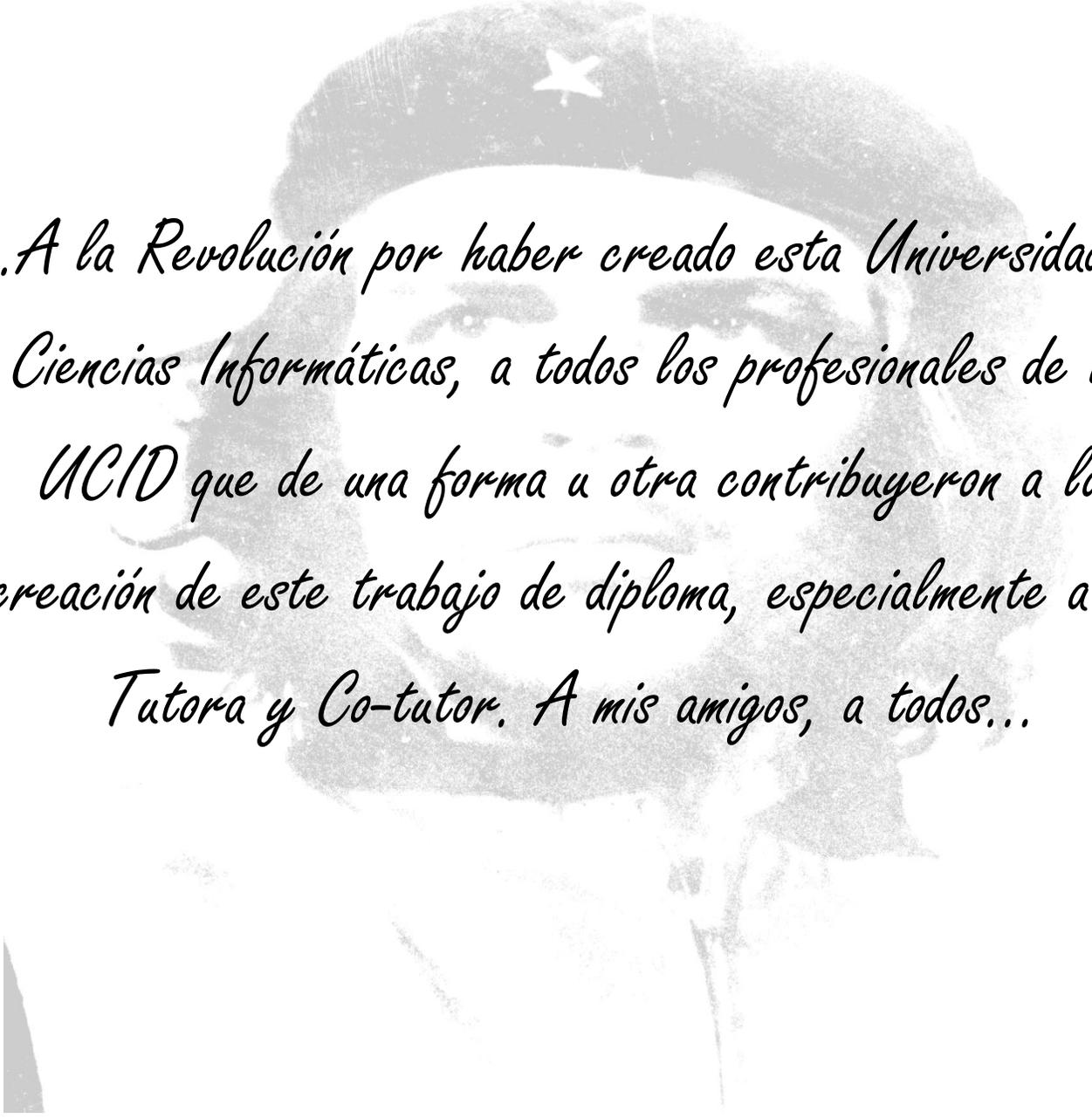
Ingeniero en Ciencias Informáticas. Graduado en julio de 2007. Especialista general del módulo 8 de la Unidad de Compatibilización e Integración y Desarrollo de Software para la Defensa (UCID), Jefe de calidad de la línea de capital humano, Jefe del módulo del Sistema para el registro, control y análisis de los medios y equipos de la reserva militar (SRCAMERM), pertenecientes ambos al proyecto Comités Militares (CMs), línea de recursos humanos, centro de mando y dirección.

Correo Electrónico: aanicot@uci.cu

Años de graduado: 3

Años de experiencia en el tema: 5

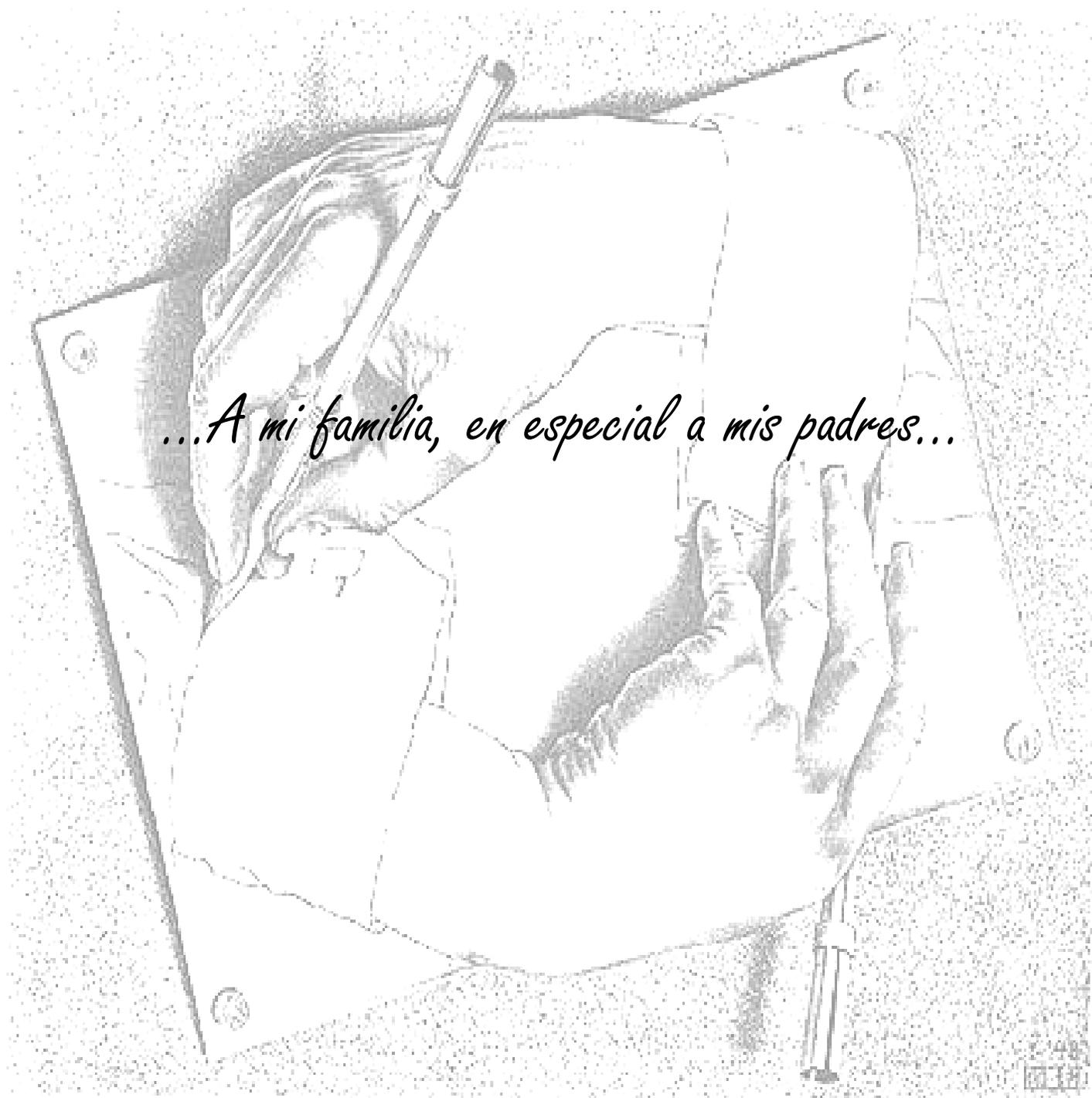
AGRADECIMIENTOS



...A la Revolución por haber creado esta Universidad de Ciencias Informáticas, a todos los profesionales de la UCID que de una forma u otra contribuyeron a la creación de este trabajo de diploma, especialmente a mi Tutora y Co-tutor. A mis amigos, a todos...

DEDICATORIA

...A mi familia, en especial a mis padres...



RESUMEN

El país se ha visto inmerso en la informatización de todas las esferas de la economía y la sociedad, el Ministerio de las Fuerzas Armadas Revolucionarias (MINFAR) se ha incorporado a la ardua tarea que es la de automatizar los procesos que se llevan a cabo en la organización. Actualmente las FAR, no cuenta con un sistema informático que facilite la representación geoespacial de los datos para los subsistemas Reclutamiento y Reserva, lo que provoca un manejo ineficiente de los mismos por los grandes volúmenes de información que se genera. Los altos directivos de las FAR hacen un análisis de información pobre y que esta es representada mediante 3 formas, tablas, gráficos y mapas, lo cual hace que este trabajo se torne engorroso y poco ágil.

EL presente trabajo de diploma desarrollará una herramienta que sea capaz de representar geoespacialmente la información de los subsistemas de Reclutamiento y Reserva. Para darle solución al problema existente se hará un estudio profundo de las herramientas de representación geoespacial que más se asemejan a la que se desea desarrollar.

Una vez desarrollado e instalado el sistema en todos los Comités Militares Municipales y Provinciales pertenecientes al MINFAR proporcionará una serie de beneficios a los usuarios ya sean civiles o militares de la institución que tendrán acceso a la información. El desarrollo de esta herramienta incorporará mejoras significativas en la calidad del trabajo que se realiza con estos datos como son:

- El manejo estadístico tendrá la calidad requerida.
- Se ganará en tiempo y mejorarán la toma de decisiones por parte de los jefes.
- El receptor tendrá las estadísticas centralizadas y podrá confiar plenamente sin temor a un mal manejo de los datos.
- Los jefes no dependerá de segundas ni terceras personas para el acceso a la información.

PALABRAS CLAVE

MINFAR, Reserva, Reclutamiento, Representación geoespacial

ÍNDICE DE CONTENIDO

INTRODUCCIÓN	1
CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA	4
INTRODUCCIÓN	4
1.1 Sistemas informáticos de representación geoespacial usados nacional e internacionalmente.....	4
1.2 Tecnologías actuales	9
1.3 Metodologías	22
1.4 Técnicas y métodos utilizados para el modelado de procesos y captura de requerimientos.....	27
CONCLUSIONES	29
CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA	31
INTRODUCCIÓN	31
2.1 Objeto de estudio.....	31
2.2 Problema	31
2.3 Situación problemática.....	31
2.4 Información que se maneja	32
2.5 Propuesta de sistema	32
2.6 Creación del modelo conceptual	32
2.7 Especificación de los requisitos de software	33
2.7.1 <i>Requerimientos Funcionales</i>	33
2.7.2 <i>Requerimientos no funcionales</i>	36
CONCLUSIONES	38
CAPÍTULO 3: ANÁLISIS Y DISEÑO DEL SISTEMA	39

INTRODUCCIÓN	39
3.1 Diseño.....	39
3.1.1 <i>Diagrama de componentes</i>	39
3.1.2 <i>Diagrama de clases web genérico</i>	41
3.1.3 <i>Diagrama de clases</i>	45
3.1.4 <i>Diagramas de interacción</i>	53
3.2 Diseño de la BD	54
3.2.2 <i>Diagrama Entidad Relación de la BD</i>	54
3.1.2 <i>Descripción de las tablas</i>	55
3.3 Definiciones de diseño que se apliquen.....	57
3.4 Tratamiento de errores.....	58
3.5 Seguridad.....	58
3.6 Interfaz.....	58
3.7 Concepción de la ayuda	58
CONCLUSIONES	59
CAPÍTULO 4: IMPLEMENTACIÓN Y PRUEBAS	60
INTRODUCCIÓN	60
4.1 Implementación.....	60
4.1.1 <i>Matriz de integración de componentes</i>	60
4.1.2 <i>Matriz de Integración de Componentes Interna</i>	60
4.1.3 <i>Matriz de Integración de Componentes Externa</i>	61
4.2 Prueba	61

4.2.1	<i>Descripción de los métodos empleados para la realización de las pruebas</i>	65
4.2.2	<i>Resultado de las pruebas</i>	65
	CONCLUSIONES	66
	CONCLUSIONES GENERALES	67
	RECOMENDACIONES	68
	BIBLIOGRAFÍAS	69
	REFERENCIAS BIBLIOGRÁFICAS	70
	GLOSARIO DE TÉRMINOS	73

INTRODUCCIÓN

El significativo desarrollo de las Tecnologías de la Información y las Comunicaciones (TIC) ha conllevado a la creación de nuevas herramientas en los más diversos campos, que permiten mejorar el proceso de toma de decisiones en toda la sociedad. Se ha demostrado que en dichos procesos entre el 80 y el 90% de la información que se maneja tiene una componente espacial. Los negocios a medida que pasa el tiempo exigen crear aplicaciones más complejas, con menos tiempo y presupuesto que antes. Crear estas aplicaciones, requiere en muchos casos de funcionalidades ya antes implementadas como parte de otros sistemas.

En las Fuerzas Armadas Revolucionarias (FAR) existe la necesidad de Representar geoespacialmente la información estadística de los subsistemas de Reclutamiento y Reserva. Debido a su gran extensión territorial y continuo crecimiento a medida que pasa el tiempo, cada vez se hace más complejo el manejo de dicha información debido a que no se cuenta con una herramienta que permita la representación geoespacial de los datos de los procesos Reclutamiento y Reserva.

Actualmente la dirección las FAR recibe anualmente una tabla de reportes regulares que contiene los datos necesarios para el proceso de toma de decisiones. Este trabajo se hace de forma manual representando los datos en un mapa de papel lo que provoca que el manejo estadístico no digitalizado no se haga de la forma y la calidad que lo requiere, lo cual se hace más engorroso y provoca pérdida de tiempo y malas decisiones.

Después de hacer un profundo análisis de las necesidades existentes surge el siguiente **problema a resolver**: ¿Cómo facilitar la representación geoespacial de la información de los subsistemas Reclutamiento y Reserva para contribuir al proceso de toma de decisiones?

Con este trabajo de diploma se pretende permitir a los usuarios de las FAR la representación geoespacial de la información de los subsistemas Reclutamiento y Reserva a nivel de MINFAR, Ejército, Comités Militares Provinciales (CMP) o Comités Militares Municipales (CMM). Disminuyendo así los tiempos de obtención de la información y agilizando el trabajo.

En correspondencia con lo analizado el **objeto de estudio** está centrado en los procesos de gestión y procesamiento de la información geoespacial para su representación. Definiendo el **campo de acción** como: Representación geoespacial de los datos almacenados en los subsistemas de Reclutamiento y Reserva.

Objetivo General: Desarrollar una herramienta que permita la representación geoespacial de la información de los subsistemas de Reclutamiento y Reserva.

Idea a defender: Si se desarrolla una herramienta que sea capaz de representar geoespacialmente la información almacenada de los procesos de Reclutamiento y Reserva, entonces facilitará el proceso de análisis para la toma de decisiones.

Objetivos específicos:

- Realizar el diseño teórico de la investigación.
- Realizar el estudio del estado del arte.
- Modelar la solución.
- Realizar el diseño de la solución.
- Implementar una herramienta para la representación geoespacial de la información.
- Realizar pruebas de caja negra al software.

Capítulo 1. Fundamentación Teórica: Incluye un estado del arte del tema tratado, a nivel internacional, nacional de la Universidad y MINFAR, de las tendencias, técnicas, tecnologías, metodologías y software usados en la actualidad o en los que apoyarse para darle solución al problema enfrentado.

Capítulo 2. Características del sistema: Este capítulo hará un análisis del problema y situación problemática. Identificará los procesos que requieren automatización para su descripción. Así como también se describirán los sistemas automatizados que existen en la empresa. Se planteará la propuesta del

sistema y se modelarán los procesos de negocio especificando los requisitos del software y se describirán los requisitos funcionales.

Capítulo 3. Análisis y Diseño del sistema: Realizará un diseño donde se obtendrían los diagramas de clases e interacción. Hará una descripción de las clases así como el diseño de la base de datos donde se obtendría el diagrama de entidad relación de la Base de Datos y la descripción de las clases.

Capítulo 4. Implementación y Prueba: Refleja los aspectos relacionados con la construcción de la solución propuesta. Se obtienen los diagramas de componentes, su descripción y la matriz de integración de componentes ya sea interna como externa. Para las pruebas se describen los diseños de caso de prueba y la descripción de los métodos empleados para la realización de las pruebas y resultado de las mismas.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

INTRODUCCIÓN

En el presente capítulo se brinda una visión general de los aspectos relacionados con las tecnologías de información geoespacial, a través del análisis de sus disponibilidades a escala internacional, nacional y del MINFAR. También se muestra una breve panorámica de las características de cada tipo de herramientas y tecnologías actuales utilizados para la confección de la herramienta. Además, se ofrece una pequeña descripción del Proceso de desarrollo de software escogido.

1.1 Sistemas informáticos de representación geoespacial usados nacional e internacionalmente.

1.1.1 Sistemas informáticos para la representación geoespacial usados en el mundo

Cada país tiene necesidades diferentes, cada uno de ellos desarrollará el software según sus necesidades y políticas del gobierno que en ese momento esté presente. Por ejemplo en los países capitalistas el desarrollo de un Sistema de Información Geográfica (SIG) tiene como objetivo intereses propios, ya que existe la propiedad privada. A continuación se mencionan algunos de estos SIG.

1.1.1.1 GeoPista: Este software fue desarrollado con el objetivo de gestionar toda la información del ayuntamiento de Macael que es un municipio español de la provincia de Almería. Este software está compuesto por varios módulos dentro de los que se destacan el módulo de Guía Urbana, esta aplicación permite a través de un navegador las operaciones básicas de navegación a través del mapa del municipio, para mostrar información a escala, localizadores de vías y puntos de información (callejero y puntos de interés cultural o social) y operaciones de impresión para obtener dichos mapas en soporte de papel.

1.1.1.2 Maptitude: Maptitude es una poderosa combinación de software y datos geográficos que ofrecen múltiples beneficios en la generación de mapas y análisis espacial, en un único producto, fácil de usar y facilita crear y modificar mapas. Permiten ubicar diagramas de barras o circulares encima de mapas, como una característica temática de trazado además de que los archivos de imágenes,

tales como las fotos de satélite, se fusionen con mapas, brindando un gran nivel de detalle. Con Maptitude es posible:

- Crear mapas atractivos e informativos.
- Crear informes y presentaciones con mapas que ilustran claramente su mensaje.
- Encontrar patrones geográficos, ocultos en bases de datos y planillas de cálculo.
- Responder preguntas geográficas relativas a sus negocios.
- Compartir datos geográficos con su grupo de trabajo, departamento u organización. [1]

Algunas de las nuevas potencialidades de Maptitude 5:

- Base de datos mundial de sitios de interés.
- Lista mundial comprensiva de lugares poblados.
- Georeferenciador por ciudad y país.
- Integración con Google Earth.
- Conexiones a proveedores de datos libres.
- Videos de demostración del uso de Maptitude.

1.1.1.3 Google Earth: Te permite volar a cualquier parte de la Tierra para ver imágenes satelitales, mapas, relieves, edificios 3D, desde galaxias del espacio hasta los cañones del océano. Explora un rico contenido geográfico, guarda los lugares que se visitan y se pueden compartir con otras personas. Las funciones de Google Earth son las siguientes:

- Imágenes históricas de todo el mundo.
- Datos de los fondos marinos y la superficie oceánica proporcionados por expertos.

- Viajes simplificados con posibilidad de añadir audio y grabaciones de voz.[2]

1.1.2 Sistemas informáticos para la representación geoespacial usados en Cuba

En el mundo globalizado actual, en esta era de la creación de la sociedad de la información y el conocimiento, el gran dilema de los países subdesarrollados como lo es Cuba reside esencialmente en cómo gestionar adecuadamente y al menor costo posible la información que necesitan sus profesionales, científicos y ciudadanos en general con el propósito de potenciar la innovación a nivel de toda la sociedad. Para esto se han desarrollado varios sistemas informáticos de representación geoespacial de la información que sin duda alguna agiliza el proceso de toma de decisiones. A continuación se hace mención a algunos de estos sistemas utilizados en Cuba.

La Infraestructura de Datos Geoespacial de la República de Cuba (IDGRC) presenta disponible el Portal Geoespacial Nacional o GeoPortal (actualmente muy pobre) para brindar información geográfica disponible del país. Los datos previstos hasta el momento para su búsqueda y localización son Distribución Política Administrativa, Hidrografía, Puntos Poblados y asentamientos urbanos, Elevaciones, Viales y Nombres Geográficos de provincias y municipios.

La IDERC permitirá compartir la información geográfica en un ambiente cooperativo interinstitucional para soportar la toma de decisiones sociales, económicas y ambientales. [3]

La Universidad de las Ciencias Informática (UCI) no se ha quedado atrás en el desarrollo de estas tecnologías que son muy útiles y necesarias para el conocimiento geográfico y la toma de decisiones y se ha visto muy involucrada en el tema con el desarrollo de varios Trabajos de Diploma, como son:

1.1.2.1 Sistema de Información Geográfica sobre la WEB de la UCI: Permite mostrar las funcionalidades básicas de un SIG y hacer búsquedas por todos los edificios. [4]

1.1.2.2 Sistema de gráficos estadísticos y mapas temáticos: Ofrece la creación de gráficos estadísticos, por ejemplo: un gráfico de barras que muestre las llamadas telefónicas realizadas por cada mes. Además, permite crear mapas temáticos, como son: cantidad de estudiantes suspensos y por ciento de varones o hembras por edificios. [4]

1.1.2.3 Sistemas informáticos para la representación geoespacial usados en el MINFAR

Las Fuerzas Armadas Revolucionarias es una institución de mucha importancia en el país que controla recursos ya sean humanos o materiales, por lo que requiere de sistemas o herramientas informáticas que permitan agilizar el proceso de toma de decisiones y hacer el análisis requerido de la información que se manejan. Para esto se ha tenido que ver involucrada en el uso de algunos Softwares reconocidos a nivel mundial en la representación geoespacial de la información como lo es MapInfo. El centro UCID no ha estado ajeno a esta necesidad por lo que actualmente está en marcha el proyecto GeneSIG el cual brinda funcionalidades para la representación de la información en mapas. A continuación se realiza una breve descripción de los Softwares mencionados anteriormente.

MapInfo: Permiten ubicar diagramas de barras o circulares encima de mapas, como una característica temática de trazado además de que los archivos de imágenes, tales como las fotos de satélite, se fusionen con mapas, brindando un gran nivel de detalle. Permite a los usuarios elegir las siguientes opciones:

- Visualización de los Datos: Este es el punto fuerte de MapInfo. Da la posibilidad de ver los datos contenidos en las tablas de la base de datos de muchas formas diferentes, en forma de mapa o gráficos.
- Mapeado por Capas: Incorpora a una ventana de mapas varias capas de información que son manejables.
- Utilización del mapeado temático para el análisis: Poderosa herramienta para la representación de los datos de muchas formas diferentes.
- Zonificación: Permite al usuario asignar y manipular territorios para el análisis espacial.

- Análisis geográfico: Análisis potente de los datos geográficos en el mapa con guardado incluido en tampón. [5]

GeneSIG: Este proyecto se crea con la necesidad que tiene la FAR de representar información geoespacial en Softwares libres, debido a que mientras se esté usando softwares propietarios deberá pagar la licencia del mismo. GeneSIG es una plataforma para la representación geográfica de diferentes módulos del país. Esta plataforma usa entre otras herramientas libres PHP como lenguaje del lado del servidor, como servidor de mapa utiliza MapServer y en la programación del lado del cliente implementa EXT con JavaScripts. Tiene integrado las siguientes funcionalidades: Seleccionar objetos en el mapa, Calcular distancia, Redimensionar mapas, Localizador por coordenadas, Calculo de acimut, Exportación de los mapas a diferentes formatos.

Después de haber realizado un análisis profundo de los software de Sistemas de Información Geográfica usados a nivel internacional y nacional se llegó a la conclusión de que exceptuando GeneSIG ninguno de estos sistemas pueden ser usados pues los mismos no cumplen con las funcionalidades necesarias para la representación geoespacial de los subsistemas de Reclutamiento y Reserva de Cuba. GeneSIG es utilizado pues, además de ser desarrollado por el centro UCID con tecnologías y herramientas libres brinda el mapa cubano donde se representarán los datos geoespacialmente.

Software Libre

El software libre es una cuestión de la libertad de los usuarios de ejecutar, copiar, distribuir, estudiar, cambiar y mejorar el software, no de libertad de precios. Más precisamente, significa que los usuarios de programas tienen las cuatro libertades esenciales. [6]

- La libertad de usar el programa, con cualquier propósito.
- La libertad de estudiar cómo funciona el programa y modificarlo, adaptándolo a sus necesidades.
- La libertad de distribuir copias del programa, con lo cual puedes ayudar a tu prójimo.
- La libertad de mejorar el programa y hacer públicas esas mejoras a los demás, de modo que toda la comunidad se beneficie.

Las libertades 1 y 3 requieren acceso al código fuente porque estudiar y modificar software sin su código fuente es muy poco viable.

1.2 Tecnologías actuales

1.2.1 Arquitectura Cliente/Servidor

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 modelo soporta un medioambiente distribuido en el cual los requerimientos de servicio hechos por estaciones de trabajo inteligentes o "clientes", resultan en un trabajo realizado por otros computadores llamados servidores. [7] Véase la figura 1.1.

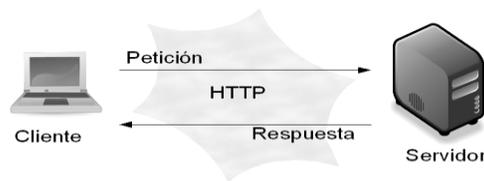


Figura: 1.1: Arquitectura Cliente/Servidor.

1.2.1.1 Principales características de la Arquitectura Cliente/Servidor:

- El Cliente y el Servidor pueden actuar como una sola entidad y también pueden actuar como entidades separadas, realizando actividades o tareas independientes.
- Las funciones de Cliente y Servidor pueden estar en plataformas separadas, o en la misma plataforma.
- Un servidor da servicio a múltiples clientes en forma concurrente.
- Cada plataforma puede ser escalable independientemente. Los cambios realizados en las plataformas de los Clientes o de los Servidores, ya sean por actualización o por reemplazo tecnológico, se realizan de una manera transparente para el usuario final.

- La interrelación entre el hardware y el software están basados en una infraestructura poderosa, de tal forma que el acceso a los recursos de la red no muestra la complejidad de los diferentes tipos de formatos de datos y de los protocolos.
- Las funciones Cliente/Servidor pueden ser dinámicas. Ejemplo, un servidor puede convertirse en cliente cuando realiza la solicitud de servicios a otras plataformas dentro de la red.
- Además, se constituye como el nexo de unión más adecuado para reconciliar los sistemas de información basados en mainframes o minicomputadores, con aquellos otros sustentados en entornos informáticos pequeños y estaciones de trabajo.[8]

1.2.1.2 Ventajas de la Arquitectura Cliente/Servidor:

- Centralización de los recursos: Mediante el servidor que es el centro de la red, se puede administrar los recursos comunes para todos los usuarios.
- Mejoramiento de la seguridad: Puesto que la cantidad de puntos de entrada permite el acceso a los datos.
- Administrar el nivel del servidor: Requieren menos administración debido a que los clientes no son muy importante en este modelo.
- Red escalable: Esta arquitectura, permite quitar o agregar clientes sin afectar el funcionamiento de la red sin mayores modificaciones.
- Centralización del control: Los accesos, recursos y la integridad de los datos son controlados por el servidor de forma que un programa cliente defectuoso o no autorizado no pueda dañar el sistema.
- Escalabilidad: Aumenta la capacidad de clientes y servidores por separado pudiendo ser los elementos aumentado o mejorado en cualquier circunstancia, además, de añadir nuevos nodos a la red clientes-servidores.

- Mantenimiento fácil: Distribuidas las funciones y responsabilidades en diferentes equipos independientes, se puede reemplazar, reparar, actualizar, o trasladar un servidor, mientras sus clientes no son afectados por el cambio.
- Las tecnologías son lo bastante desarrolladas, diseñadas en el entorno de Cliente Servidor para asegurar las transacciones, amigabilidad de la interfaz y la facilidad de empleo. [9]

1.2.1.3 Desventajas de la Arquitectura Cliente/Servidor:

- Alto Costo: Debido a lo complicado de la técnica del servidor.
- Punto débil: El servidor es el único eslabón débil en la red de cliente/servidor, puesto que toda la red está construida en torno a él.
- La congestión del tráfico ha sido siempre un problema en el paradigma de C/S. Cuando una gran cantidad de clientes envían peticiones simultaneas al mismo servidor, puede ser que cause muchos problemas para éste.
- El cliente no usa los recursos dentro del servidor. Por ejemplo, si la aplicación es una Web, no se puede escribir en el disco duro del cliente o imprimir directamente sobre las impresoras sin sacar antes la ventana previa de impresión de los navegadores. [9]

1.2.2 Servidor web

Básicamente, un servidor web sirve contenido estático a un navegador, carga un archivo y lo sirve a través de la red al navegador de un usuario. Este intercambio es mediado por el navegador y el servidor que hablan el uno con el otro mediante HTTP. Se pueden utilizar varias tecnologías en el servidor para aumentar su potencia más allá de su capacidad de entregar páginas HTML; estas incluyen scripts CGI, seguridad SSL y páginas activas del servidor. [10]

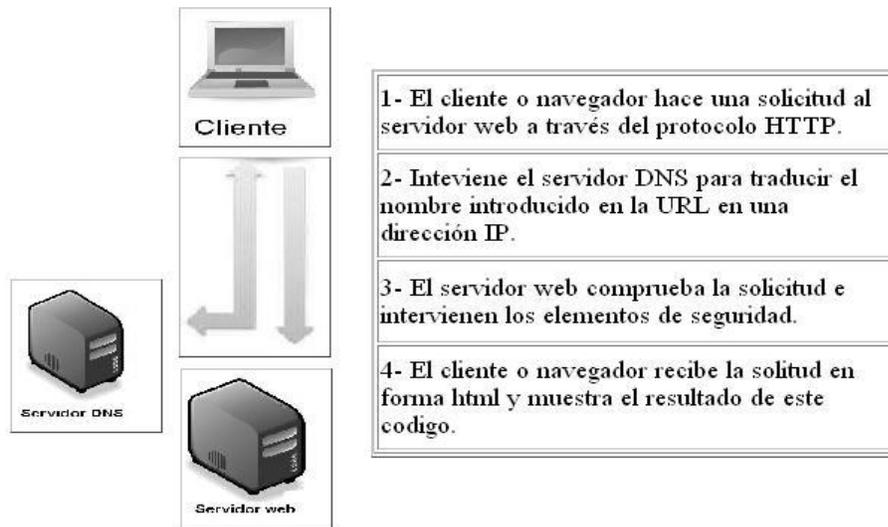


Figura 1.2: Servidor web.

El servidor Web más utilizado en el mundo y a su vez el más popular y comprado es el Apache, fue el elegido para la realización de este trabajo por su libertad y capacidad. Esta popularidad que ha alcanzado se debe gracias a su sofisticación tecnológica. El éxito del servidor Apache también viene respaldado por la posibilidad de recogerlo a través de Internet. Los usuarios pueden descargar íntegramente el código fuente para compilarlo ellos mismos o bien descargar versiones precompiladas. Adicionalmente y debido a que el código fuente está disponible, los usuarios pueden aumentar y personalizar las funciones del servidor añadiendo módulos que extienden sus capacidades. Fue el primer servidor en implementar el protocolo HTTP/ 1.1. Para dar vigencia de la confiabilidad y seguridad que brinda este servidor le brindamos unas series de características:

1.2.2.1 Características del servidor web Apache:

- Multiplataforma.
- Modular: Puede ser adaptado a diferentes entornos y necesidades, con los diferentes módulos de apoyo que proporciona, y con la API de programación de módulos, para el desarrollo de módulos específicos.

- Incentiva la realimentación de los usuarios, obteniendo nuevas ideas, informes de fallos y parches para la solución de los mismos.
- Se desarrolla de forma abierta.
- Extensible: Gracias a ser modular se han desarrollado diversas extensiones entre las que destaca PHP, un lenguaje de programación del lado del servidor.[11]

1.2.3 Navegador o Browsers

Aplicación para visualizar documentos WWW y navegar por Internet. En su forma más básica son aplicaciones hipertexto que facilitan la navegación por los servidores de navegación de Internet. Los más avanzados, cuentan con funcionalidades plenamente multimedia y permiten indistintamente la navegación por servidores WWW, FTP, Gopher, acceso a grupos de noticias, la gestión del correo electrónico, etc. [12]

1.2.3.1 ¿Por qué utilizar Mozilla Firefox?

- Mozilla Firefox es compatible con varios estándares web.
- Mozilla Firefox es un navegador multiplataforma.
- Firefox es sumamente seguro.
- Presenta navegación por pestañas.
- Software libre.

1.2.4 Lenguajes de programación para la web

Debido a la arquitectura utilizada para la creación de este trabajo Cliente/Servidor se encontró los lenguajes de programación del lado del cliente y el lenguaje de programación del lado del servidor.

En el primer grupo que es el lenguaje del lado del cliente se tiene a: HTML, JAVASCRIPT, Técnicas AJAX, XML.

En el segundo grupo que son los lenguajes del lado del servidor se tiene a: PHP.

1.2.4.1 Del lado del Cliente:

HTML

Hyper Text Markup Language, o simplemente HTML, es un lenguaje de programación muy sencillo compuesto por etiquetas que se utiliza para crear los textos y páginas web. Si se hace la traducción de su nombre del inglés al castellano, sería “Lenguaje de Marca de Hipertextos”, ya que es justamente un lenguaje que se basa en las marcas para crear los hipertextos. [13]

AJAX

Es una técnica de desarrollo web para crear aplicaciones interactivas. Estas aplicaciones se ejecutan en el cliente, es decir, en el navegador de los usuarios mientras se mantiene la comunicación asíncrona con el servidor en segundo plano. De esta forma es posible realizar cambios sobre las páginas sin necesidad de recargarlas, lo que significa aumentar la interactividad, velocidad y usabilidad en las aplicaciones. [14]

¿Por qué utilizar Técnica Ajax?

- Recuperación asíncrona de datos, reduciendo el tiempo de espera del usuario.
- Código público.
- Portabilidad entre plataformas.

XML

XML: Más conocido como Extensible Markup Language. Es un meta-lenguaje que permite definir lenguajes de marcado adecuados a usos determinados. En la práctica corresponde a un estándar que permite a diferentes aplicaciones interactuar con facilidad a través de La Red, fue creado al amparo del World Wide Web Consortium (W3C) organismo que vela por el desarrollo de WWW partiendo de las amplias especificaciones de SGML. Su desarrollo se comenzó en 1996 y la primera versión salió a la luz

el 10 de febrero de 1998. La primera definición que apareció fue: Sistema para definir validar y compartir formatos de documentos en la web. [15]

1.2.4.2 Del lado del Servidor:

PHP

Es un lenguaje del lado del servidor es aquel que se ejecuta en el servidor web, justo antes de que se envíe la página a través de Internet al cliente. Las páginas que se ejecutan en el servidor pueden realizar accesos a bases de datos, conexiones en red, y otras tareas para crear la página final que verá el cliente. El cliente solamente recibe una página con el código HTML resultante de la ejecución de la PHP. Como la página resultante contiene únicamente código HTML, es compatible con todos los navegadores. [16] Su Funcionamiento es el siguiente. Véase figura 1.3



Figura 1.3: Funcionamiento de PHP.

Usos del lenguaje de programación PHP:

- Programación de páginas web dinámicas, habitualmente en combinación con el motor de base datos MySQL, aunque cuenta con soporte nativo para otros motores, incluyendo el estándar ODBC, lo que amplía en gran medida sus posibilidades de conexión.
- Programación en consola, al estilo de Perl o Shell scripting.

- Creación de aplicaciones gráficas independientes del navegador.

¿Por qué usar de PHP?

- Es un lenguaje multiplataforma.
- Capacidad de conexión con la mayoría de los manejadores de base de datos que se utilizan en la actualidad, destaca su conectividad con MySQL.
- Lee y manipula datos desde diversas fuentes, incluyendo datos que pueden ingresar los usuarios desde formularios HTML.
- Es libre, por lo que se presenta como una alternativa de fácil acceso para todos.
- Permite las técnicas de Programación Orientada a Objetos.
- Permite crear los formularios para la web.

1.2.5 Sistema Gestor de Bases de Datos (SGBD)

1.2.5.1 Base de Datos

Una Base de Datos es una colección de información organizada de forma que un programa de ordenador pueda seleccionar rápidamente los fragmentos de datos que necesite. Una base de datos es un sistema de archivos electrónico.

Las bases de datos tradicionales se organizan por campos, registros y archivos. Un campo es una pieza única de información; un registro es un sistema completo de campos; y un archivo es una colección de registros. Por ejemplo, una guía de teléfono es análoga a un archivo. Contiene una lista de registros, cada uno de los cuales consiste en tres campos: nombre, dirección, y número de teléfono. [17]

(Sistema de gestión de base de Datos) o en inglés Database management system (DBMS), es una agrupación de programas que sirven para definir, construir y manipular una base de Datos. [18] Los SGBD tienen las siguientes funciones:

- **Definición de los Datos:** El SGBD debe ser capaz de aceptar las definiciones de datos en versión fuente y convertirlas en la versión objeto, debe incluir componentes procesadores para cada uno de los lenguajes de definición de datos.
- **Manipulación de los Datos:** El SGBD debe atender las solicitudes de los usuarios para extraer, actualizar, adicionar o suprimir datos, debe incluir un componente procesador del Lenguaje de manipulación de datos.
- **Seguridad e Integridad de los Datos:** El SGBD debe supervisar las solicitudes de los usuarios y rechazar los intentos de violar las medidas de seguridad e integridad definidas por el Administrador de la Base de Datos.
- **Recuperación y concurrencia de los datos:** El principal objetivo de la implantación de una base de datos es poner a disposición de un gran número de usuarios en conjunto integrado de datos, estos datos podrán ser manipulados por los diferentes usuarios y es ahora cuando se debe garantizar la coherencia de los datos después de las diversas manipulaciones. [19]

1.2.5.2 PostgreSQL

PostgreSQL es un sistema de gestión de base de datos relacional orientada a objetos y libre, realizado para múltiples plataformas. Este puede distribuirse por todos y para todos con código abierto donde se puede modificar y usar para cualquier propósito. Es un motor de base de datos avanzado, que soporta consultas complejas, integridad referencia (llave foráneas), triggers, vistas, integridad transaccional y control de versiones concurrentes.

¿Por qué utilizar PostgreSQL?

- Ahorros considerables en costos de operación: El software ha sido diseñado y creado para tener un mantenimiento y ajuste, conservando todas las características, estabilidad y rendimiento.
- Estabilidad y confiabilidad legendarias: En contraste a muchos sistemas de bases de datos comerciales, es extremadamente común que compañías reporten que PostgreSQL nunca ha

presentado caídas en varios años de operación de alta actividad. Ni una sola vez. Simplemente funciona.

- Multiplataforma: PostgreSQL está disponible en casi cualquier Unix (34 plataformas en la última versión estable), también soporta Windows y Linux.
- Diseñado para ambientes de alto volumen: PostgreSQL usa una estrategia de almacenamiento de filas llamada MVCC para conseguir una mejor respuesta en ambientes de grandes volúmenes. [20]

1.2.6 Herramientas CASE

Son un conjunto de métodos, utilidades y técnicas que facilitan la automatización del ciclo de vida del desarrollo de sistemas de información, completamente o en alguna de sus fases. El empleo de herramientas CASE permiten integrar el proceso de ciclo de vida:

- Análisis de datos y procesos integrados mediante un repositorio.
- Generación de interfaces entre el análisis y el diseño.
- Generación del código a partir del diseño.
- Control de mantenimiento. [21]

1.2.6.1 ¿Por qué utilizar Herramientas CASE?

- Menor tiempo de mantenimiento.
- Mayor independencia entre análisis, diseño y programación.
- Mayor independencia del análisis y diseño con respecto a un entorno en particular.
- Mejora de la calidad del producto de software.
- Aplicaciones más productivas para las empresas.[22]

1.2.6.2 Visual Paradigm for UML

Visual Paradigm para UML es una herramienta que soporta el ciclo completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. El software de modelado UML ayuda a una más rápida construcción de aplicaciones de calidad, mejores y a un menor coste. Permite dibujar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación. Dentro de los tipos de diagramas que permite dibujar están Diagramas de Procesos de Negocio el cual permite identificar los procesos, actores del negocio, y toma de decisiones. [23]

1.2.6.3 ¿Por qué utilizar Visual Paradigm?

- Es multiplataforma. No importa el sistema operativo donde esté trabajando.
- Generación de código.
- Generación de bases de datos. Transformación de diagramas de Entidad-Relación en tablas de base de datos.
- Ingeniería inversa de bases de datos. Desde Sistemas Gestores de Bases de Datos (DBMS) existentes a diagramas de Entidad-Relación.
- Generador de informes para generación de documentación.
- Distribución automática de diagramas. Reorganización de las figuras y conectores de los diagramas UML.[24]

1.2.7 Sistema de Control de Versiones

Un Sistema de Control de Versiones es una herramienta que se utiliza para mantener distintas versiones de un fichero, ya sea código fuente de un proyecto, documentación o ficheros de configuración. Los Sistemas de Control de Versiones proporcionan grandes ventajas como es mantener las copias locales o de repositorio a salvo y así mantener la integridad de los archivos.

1.2.7.1 TortoiseSVN

Es un cliente gratuito de código abierto para el sistema de control de versiones Subversión. Maneja ficheros y directorios a lo largo del tiempo. Los ficheros se almacenan en un repositorio central. El repositorio es prácticamente lo mismo que un servidor de ficheros ordinario, salvo que recuerda todos los cambios que se hayan hecho a sus ficheros y directorios. Esto permite que puedan recuperar versiones antiguas de sus ficheros y examinar la historia de cuándo, cómo cambiaron sus datos, y quién hizo el cambio. Esta es la razón por la que mucha gente piensa que Subversion, y los sistemas de control de versiones en general, son una especie de “máquinas del tiempo”.

1.2.7.2 ¿Por qué utilizar TortoiseSVN?

- Puede ser usado sin un entorno de desarrollo.
- Pequeñas imágenes decoran los íconos de los archivos mostrando qué archivos o directorios necesitan ser enviados al repositorio.
- Disponible en 28 idiomas diferentes.
- Maneja el mostrar la diferencia de documentos de Office tales como los creados con Microsoft Word.

1.2.8 Doctrine

Doctrine es un mapeador de objetos relacionales (ORM) para PHP 5.2.3 + que se sienta encima de una potente capa de abstracción de bases de datos (dbal). Uno de sus principales características es la opción de escribir las consultas de base de datos en un objeto de propiedad orientada dialecto SQL llamada Doctrine Query Language (DQL), inspirado en Hibernate HQL. Esto le proporciona a los desarrolladores una poderosa alternativa para la programación SQL que mantiene la flexibilidad sin necesidad de la duplicación de código innecesaria. [32]

1.2.8.1 ¿Por qué utilizar Doctrine?

- **Reutilización:** La principal ventaja que aporta un ORM es la reutilización permitiendo llamar a los métodos de un objeto de datos desde distintas partes de la aplicación e incluso desde diferentes aplicaciones.
- **Encapsulación:** La capa ORM encapsula la lógica de los datos pudiendo hacer cambios que afectan a toda la aplicación únicamente modificando una función.
- **Portabilidad:** Utilizar una capa de abstracción permite cambiar en mitad de un proyecto de una base de datos MySQL a una Oracle sin ningún tipo de complicación. Esto es debido a que no utilizamos una sintaxis MySQL, Oracle o SQLite para acceder al modelo, sino una sintaxis propia del ORM utilizado que es capaz de traducir a diferentes tipos de bases de datos.
- **Seguridad:** Los ORM suelen implementar mecanismos de seguridad que protegen la aplicación de los ataques más comunes como SQL Injections.
- **Mantenimiento del código:** Gracias a la correcta ordenación de la capa de datos, modificar y mantener el código es una tarea sencilla. [25]

1.2.9 Zend Frameworks

Se trata de un framework o marco de trabajo 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 o de código abierto y trabaja con PHP 5. [26]

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.

1.2.9.1 ¿Por qué utilizar Zend Frameworks?

- Trabaja con MVC (Model View Controller).
- Reduce el tiempo de las aplicaciones, permitiendo ofrecer presupuestos más ajustados.

- Estandariza los procesos más frecuentes, dotándolos de gran robustez.
- Facilita el mantenimiento de las aplicaciones.
- Ofrece muchas facilidades para el acceso a recursos avanzados.
- Tiene el respaldo de la propia ZEND, creadora de PHP, lo que asegura su continuidad futura tanto como la del propio lenguaje PHP. [27]

1.2.10 EXTJS

ExtJS es una librería de JavaScript Cross-Browser para el desarrollo de Aplicaciones Ricas en Internet (RIA) usando tecnologías como AJAX, DHTML y DOM. Incluye widgets personalizables, buenos diseños, API extensa y fácil de utilizar, licencia libre y comercial disponible. Es muy compatible con los diferentes navegadores existentes en el mundo y fácil de usar. 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, menus y muchos otros.

1.3 Metodologías

1.3.1 Proceso Unificado de Desarrollo (RUP)

1.3.1.1 ¿Qué es una Metodología de Desarrollo de Software?

Las metodologías imponen un proceso disciplinado sobre el desarrollo de software con el fin de hacerlo más predecible y eficiente. Lo hacen desarrollando un proceso detallado con un fuerte énfasis en planificar y organizar los procesos de la ingeniería de software.

1.3.1.2 ¿Qué es RUP?

Es un proceso de ingeniería de software o forma disciplinada de asignar tareas y responsabilidades en una organización de desarrollo. Con el objetivo de asegurar la producción de software de calidad dentro de plazos y presupuestos predecibles. Es también es un producto desarrollado y mantenido por Rational, actualizado constantemente para tener en cuenta las mejores prácticas de acuerdo con la experiencia. RUP es una guía de cómo usar UML de la forma más efectiva. [28] Está constituido por 9 flujos de trabajo

fundamentales: modelado de negocio, requerimientos, análisis y diseño, implementación, prueba, despliegue, configuración y cambio, administración del proyecto y desarrollo los cuales tienen lugar sobre 4 etapas o fases: inicio, elaboración, construcción y transición. RUP es adaptable para proyectos a largo plazo y establece refinamientos sucesivos de una arquitectura ejecutable.

1.3.1.3 Ciclos y fases

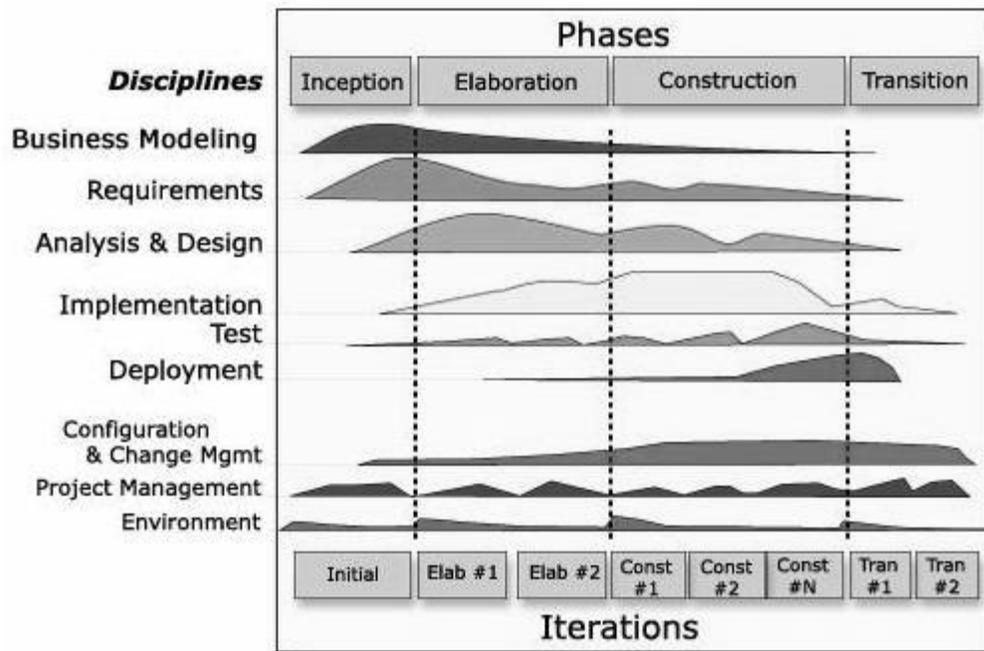


Figura 1.4: Esfuerzo en actividades según fase del proyecto.

1.3.1.4 Características de RUP

- **Guiado por los Casos de Uso:** Los Casos de Uso son el instrumento para validar la arquitectura del software y extraer los casos de prueba.

- **Centrado en la arquitectura:** Los modelos son proyecciones del análisis y el diseño constituye la arquitectura del producto a desarrollar.
- **Iterativo e incremental:** Durante todo el proceso de desarrollo se producen versiones incrementales (que se acercan al producto terminado) del producto en desarrollo.

Extreme Programming (XP) es una metodología ágil centrada en potenciar las relaciones interpersonales como clave para el éxito en desarrollo de software, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores, y propiciando un buen clima de trabajo. XP se basa en realimentación continua entre el cliente y el equipo de desarrollo, comunicación fluida entre todos los participantes, simplicidad en las soluciones implementadas y coraje para enfrentar los cambios. XP se define como especialmente adecuada para proyectos con requisitos imprecisos y muy cambiantes, y donde existe un alto riesgo técnico. [33]

XP fundamentalmente es utilizada para proyectos de corto plazo, con un equipo reducido. Consiste en una programación rápida o extrema donde el usuario final forma parte del equipo. XP también es iterativo e incremental, se basa en las pruebas unitarias continuas, en la reutilización de código, programación en parejas. Cuenta con cuatro fases: Planificación, Diseño, Desarrollo y Pruebas.

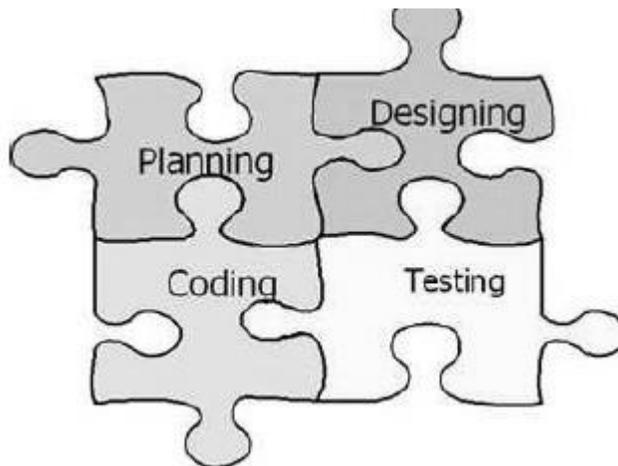


Figura 1.5: Fases de XP.

1.3.2 Proceso de desarrollo en la UCID

El modelo de desarrollo de software propuesto describe la secuencia de actividades de alto nivel para la construcción y desarrollo de soluciones. Se logra con la combinación entre los modelos basado en Componentes, el Iterativo y el Incremental. Se emplearán las técnicas de prototipado, si son requeridas, para los requerimientos del usuario de los que no existe una visión clara por parte de estos, con el objetivo de desarrollar una definición mejorada de los requisitos del usuario para el sistema.

Desarrollo iterativo e incremental: Es un enfoque en el que el ciclo de vida está compuesto por iteraciones, estas son pequeños procesos compuestos de varias actividades cuyo objetivo es entregar una parte del sistema parcialmente completo, probado, integrado y estable. Todo el software es integrado en cada entrega de cada iteración hasta obtener el producto de software completo en la última iteración. En cada iteración se obtiene como resultado un incremento.

Desarrollo basado en componentes: Lleva a alcanzar un mayor nivel de reutilización de software, aún en contextos distintos a aquellos para los que fue diseñado. Permite que las pruebas sean ejecutadas probando cada uno de los componentes antes de probar el conjunto completo de componentes ensamblados. Cuando existe un débil acoplamiento entre componentes, el desarrollador es libre de actualizar y/o agregar componentes según sea necesario, sin afectar otras partes del sistema. Dado que un componente puede ser construido y luego mejorado continuamente, la calidad de una aplicación basada en componentes mejorará con el paso del tiempo.

El ciclo de vida de un proyecto de software desarrollado en la entidad UCID se descompone en el tiempo en cinco fases secuenciales (ver figura 1.7) que son: Inicio, Modelación, Construcción, Explotación Experimental, Despliegue. [34]



Figura 1.6: Etapas del ciclo de vida del proyecto.

1.3.3 BPMN

1.3.3.1 BPMN (Business Process Modeling Notation) es un 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. [29]

1.3.3.2 ¿Por qué utilizar BPMN?

- Considera un único diagrama para la representación de los procesos.
- Fácil de aprender para los analistas del negocio, desarrolladores técnicos.
- Pensado para ser asignado con naturalidad a los lenguajes de ejecución (BPEL4WS).

1.3.4 UML

El Lenguaje de Modelado Unificado (UML: Unified Modeling Language) es la sucesión de una serie de métodos de análisis y diseño orientadas a objetos que aparecen a fines de los 80's y principios de los 90s. UML es llamado un lenguaje de modelado, no un método. Los métodos consisten de ambos de un lenguaje de modelado y de un proceso. UML incrementa la capacidad de lo que se puede hacer con otros métodos de análisis y diseño orientados a objetos. Los autores de UML apuntaron también al modelado de sistemas distribuidos y concurrentes para asegurar que el lenguaje maneje adecuadamente estos dominios. [30]

1.3.4.1 ¿Por qué utilizar UML?

- Permite la ingeniería directa. La ingeniería directa, va moviéndose desde los requerimientos, hacia el diseño (modelamiento, procesos) para llegar a la implementación.
- Documentación.
- Colaboración en el modelamiento. Para el diseño o rediseño de sistemas complejos, puede haber diferentes equipos participando el trabajo de diseño de diversos subsistemas en paralelo, debería realizarse sobre un solo ambiente.[31]

1.4 Técnicas y métodos utilizados para el modelado de procesos y captura de requerimientos

Con el significativo desarrollo de software en el mundo se han visto involucradas algunas técnicas que se utilizan para la captura de requisitos por parte de los analistas. La captura de requisitos es muy importante para el buen desarrollo de software. En el mundo existen varias técnicas para recabar la información dentro de las que se encuentran:

- Entrevistas.
- Sistemas existentes.
- JAD (Joint Application Development/ Desarrollo conjunto de aplicaciones).
- Brainstorming (Tormenta de ideas).

- Concept Mapping (Mapas Conceptuales).
- Sketches y Storyboards (Esbozos y Guiones Gráficos).
- Casos de Uso.
- Cuestionarios y Checklists (Listas de Chequeo).
- Comparación de terminología.
- Ingeniería Inversa.

En conjunto con la directiva del proyecto para la captura de requerimientos del sistema se han seleccionado las técnicas de la entrevista estructurada y no estructurada y sistemas existentes. La entrevista es utilizada para recaudar información de forma verbal a través de preguntas que propone el analista y los que responden pueden ser gerentes o empleados, los cuales son usuarios o clientes actuales del sistema existente. La entrevista es una de las técnicas más significativas para la recopilación de datos o información que se obtiene mediante un intercambio cara a cara entre el analista y el entrevistado. Es un canal de comunicación entre el analista y la organización que se utiliza para analizar información sobre las necesidades y la manera de satisfacerlas. La entrevista ofrece al analista una excelente oportunidad para establecer una corriente de simpatía con el personal usuario, lo cual es fundamental en transcurso del estudio. Para la preparación de una entrevista exitosa es necesario definir los siguientes aspectos antes de lanzarnos a entrevistar al usuario:

- Preparar las preguntas que van a plantearse, y los documentos necesarios (Organización).
- Fijar un límite de tiempo y preparar la agenda para la entrevista. (Sicología).
- Elegir un lugar donde se puede conducir la entrevista con la mayor comodidad (Sicología).
- Hacer la cita con la debida anticipación (Planeación).

Existen dos formas de realizar las entrevistas estructuradas y no estructuradas para las cuales existen ventajas y desventajas.

Ventajas de las entrevistas estructuradas:

- Asegura la elaboración uniforme de las preguntas para todos los que van a responder.
- Fácil de administrar y evaluar.
- Evaluación más objetiva tanto de quienes responden como de las respuestas a las preguntas.
- Se necesita un limitado entrenamiento del entrevistador.
- Resulta en entrevistas más pequeñas.

Ventajas de las entrevistas no estructuradas:

- El entrevistador tiene mayor flexibilidad al realizar las preguntas adecuadas a quien responde.
- El entrevistador puede explotar áreas que surgen espontáneamente durante la entrevista.
- Un alto nivel en la estructura puede no ser adecuado para todas las situaciones.
- Puede producir información sobre área que se minimizaron o en las que no se pensó que fueran importantes.

La técnica de sistemas existentes, la cual consiste en analizar los sistemas ya desarrollados como la plataforma GeneSIG y la Herramienta para la el análisis de la información actualmente en desarrollo que está relacionada con la aplicación que se va a construir. Analizándose las interfaces de usuarios y las entradas y salidas de información que los mismos proporcionan.

CONCLUSIONES

Como resultado del análisis realizado anteriormente se comparte el criterio general de la dirección del proyecto que para el desarrollo de las aplicaciones relacionadas con el perfil del proyecto se utilizarán las siguientes herramientas:

Servidor web: **Apache 2.0**

Navegador o Browsers: **Mozilla Firefox 3.0 o superior- Recomendado Firefox 3.6**

Lenguaje de programación del lado del cliente: **JavaScript, Técnicas AJAX, HTML, XML**

Lenguaje de programación del lado del servidor: **PHP v5.2.3 ó superior- recomendado PHP 5.2.6**

Sistema Gestor de Bases de Datos (SGBD): **PostgreSQL 8.3 o superior e inferior al 8.4 (Recomendado PostgreSQL 8.3.9).**

Herramienta CASE para modelado UML: **Visual Paradigm for UML 6.4 Enterprise Edition**

Sistema de Control de Versiones: **TortoiseSVN**

Para apoyar el proceso de implementación del software se utilizarán herramientas de desarrollo como **FrameWorks EXTJS, Zend FrameWorks, Doctrine v1.2.1.**

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

INTRODUCCIÓN

En este capítulo se abordará fundamentalmente el entorno del negocio y los requisitos, así como los procesos fundamentales que serán objeto de automatización. Se hará mención de los documentos procesados para la manipulación de la información. Se describirá la propuesta de la solución para dar una visión general de cómo funcionará el software Además de identificar los requisitos de software ya sean funcionales o no funcionales a través de entrevistas a los clientes.

2.1 Objeto de estudio

El objeto de estudio girará entorno a: Procesos de gestión y procesamiento de la información geoespacial para su representación.

2.2 Problema

¿Cómo facilitar la representación geoespacial de la información de los subsistemas Reclutamiento y Reserva para contribuir al proceso de toma de decisiones?

2.3 Situación problemática

Los subsistemas de Reclutamiento y Reserva instalados en todos los Comités Militares provinciales y municipales del país generan grandes volúmenes de información, la cual necesita ser analizada por parte de los jefes para la toma de decisiones. Actualmente las Fuerzas Armadas Revolucionarias no cuenta con una herramienta que facilite la representación de dichos datos lo que convierte el trabajo en engorroso y poco ágil generando un gasto innecesario de tiempo, debido a que los especialistas se convierten en estibadores de la información representándolas en mapas, tablas o graficas. Esto trae consigo un gasto excesivo de material debido a que los mapas, tablas y gráficas deben ser dibujados en papel convirtiéndose este en la principal materia prima para este trabajo. Además en algunas ocasiones el proceso de toma de decisiones podría ser ineficiente ya que el análisis de esta información generada por los subsistemas antes mencionados es pobre y carece de la calidad requerida.

2.4 Información que se maneja.

Tabla de reportes regulares: La tabla de reportes regulares es un documento digital Excel que se recibe en el MINFAR anualmente para tomar o leer datos de la misma y así poder hacer una representación en el mapa. No se exponen ninguno de los datos que aquí se representan por motivos de seguridad nacional.

2.5 Propuesta de sistema

Se propone desarrollar una herramienta que brinde una solución real a los problemas que existen actualmente en el MINFAR, más específicamente en los CMs de cada uno de los ejércitos del país. El mismo estará ubicado o instalado en todos los CMs del país y MINFAR, permitiendo a estos órganos realizar varias operaciones para la representación de los datos en el mapa de Cuba, con el objetivo de que los jefes obtengan una mayor efectividad en sus decisiones. Para ello el sistema contará con un nivel de seguridad que permita mostrar sólo los Comités Militares que pertenecen a la región territorial a la cual esté autorizado el usuario que acceda. Además de ser fácil de entender, de aplicar y de manejar en el menor tiempo posible.

La aplicación que se propone tiene relación con la Herramienta de análisis de la información actualmente en desarrollo, la plataforma GeneSIG, dichas aplicaciones brindan las recuperaciones y el mapa respectivamente. Se desea presentar una capa intermediaria a través de una interfaz que sea capaz de representar en el mapa cada uno de los reportes o recuperaciones seleccionadas por los usuarios haciendo una representación de los datos obtenidos para cada CMs que se seleccione a través de una ventana.

2.6 Creación del modelo conceptual

¿Qué es un Modelo conceptual?

Un Modelo conceptual es una serie de objetos significativos del negocio y se representan mediante clases o entidades de las cuales se representan sus atributos. Cada una de estas clases estará relacionada como una asociación y se especifica el tipo de relación que existe en lenguaje natural. Un modelo conceptual representa cosas del mundo real, no componentes del software. El modelo conceptual es una herramienta de comunicación, con la cual se intenta comprender los conceptos importantes y sus relaciones.

Para obtener un buen modelo conceptual es necesario:

- Identificar los conceptos (Objetos) significativos del negocio y se representan mediante clases.
- Relacionar los conceptos representándose como una asociación y se especifica el tipo de relación que existe en lenguaje natural. En caso que desee se puede representar la multiplicidad en las asociaciones.
- Identificar los atributos de cada concepto y se incluyen en la entidad correspondiente.
- Identificar las posibles relaciones con conceptos que no pertenezcan al negocio y se representan dentro de un paquete que identifique dicho negocio. [35]

La figura ([ver anexo 1](#)) muestra el modelo conceptual asociado al dominio del problema existente en los CMs. Las personas que pertenecen a los CMs presentan una tarjeta de registro y forman parte de un listado único. Las Personas son Prerreclutas y Reservistas. Los Prerreclutas antes tuvieron que ser Preseleccionados para los cuales se realiza el Reclutamiento para ser llamados al servicio militar activo, tienen una evaluación del Órgano de Control y un examen médico. Los Reservistas tienen o no uno o muchos exámenes médicos además de tener una evaluación del Órgano de Control. El cliente o usuario será el encargado de utilizar el mapa digital para la representación geoespacial de los datos de estas personas.

2.7 Especificación de los requisitos de software

Un requisito define un requerimiento como Condición o capacidad que necesita un usuario para resolver un problema o lograr un objetivo.

2.7.1 Requerimientos Funcionales

Los requerimientos funcionales son capacidades o condiciones que el sistema debe cumplir. El sistema deberá ser capaz de:

R 1. Gestionar Recuperaciones de la Reserva.

R 1.1 Cargar recuperaciones guardadas.

R 1.2 Guardar recuperación.

R 1.5 Representar recuperaciones.

R 2. Gestionar Recuperaciones de Reclutamiento.

R 2.1 Cargar recuperaciones guardadas.

R 2.2 Guardar recuperación.

R 2.5 Representar recuperaciones.

2.7.1.1 Descripción de los requisitos funcionales.

2.7.1.1.1 Gestionar Recuperaciones de la Reserva.

R 1.1 Cargar Recuperaciones guardadas.

Conceptos tratados	Conceptos	Atributos
	No procede.	No procede.
Precondiciones	Precondiciones	Pre-requisito
	Se han guardado las recuperaciones.	No procede.
Descripción	<ol style="list-style-type: none"> 1. Se selecciona la opción Análisis de la información/ Reserva/ Representación geoespacial del menú principal. 2. Se muestra la interfaz Representación geoespacial. 3. Se oprime el botón Seleccionar reportes. 4. Se muestra la ventana Reportes. <ol style="list-style-type: none"> 4.1 Si existen recuperaciones que se hayan representado 	

	<p>anteriormente estas se mostrarán marcadas.</p> <p>5. Se seleccionan las recuperaciones a mostrar en el mapa.</p> <p>5.1 Si se oprime el botón Cancelar se cierra la ventana Reportes.</p> <p>5.2 Si oprime el botón Aceptar se guardan las recuperaciones que se desean mostrar.</p>
Validaciones	No procede.
Post-condiciones	No procede.
Post-requisito	No procede.

Prototipo Interfaz de usuario Cargar Recuperaciones guardadas.



Figura 2.2: Cargar Recuperaciones guardadas.

R 1.2 Guardar recuperación: El siguiente requisito funcional permite a los usuarios seleccionar la o las recuperaciones que desea representar geoespacialmente. Para mostrar su descripción ([ver anexo 2](#)).

R1.5 Representar recuperaciones: El siguiente requisito funcional permite a los usuarios representar en el mapa las recuperaciones seleccionadas. Para mostrar su descripción ([ver anexo 3](#)).

Cuando el usuario hace la representación sobre el mapa se muestra el **Prototipo Interfaz de usuario** Representar recuperaciones ([ver anexo 4](#)).

2.7.2 Requerimientos no funcionales

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

- **Apariencia o interfaz externa:** El sistema debe contar con una interfaz fácil, amigable, sencilla, permitiendo que los usuarios finales del mismo sean capaces de interactuar con este aún teniendo conocimientos básicos. Se empleará imágenes y colores identificando la funcionalidad, el nombre de los botones evidencian su acción o responsabilidad.
- **Usabilidad:** La herramienta podrá ser usada por personas que cuenten con un conocimiento básico de informática. Se emplearán barras de progreso para indicar el estado de los procesos que por su complejidad requieran de un tiempo de procesamiento apreciable por los usuarios. El software tendrá siempre visible la opción de Ayuda, lo que posibilitará un mejor aprovechamiento por parte de los usuarios de sus funcionalidades.
- **Rendimiento:** Los tiempos de respuesta y velocidad de procesamiento de la información serán no tan rápidos, con una espera de 20 a 30 segundos para representar el mapa y de 5 a 10 segundos para la representación de la información.
- **Soporte:** Se necesita un servidor de base de datos que soporte grandes volúmenes de información.
- **Portabilidad:** La herramienta es multiplataforma. (Linux y Windows).

- **Seguridad:** El sistema estará integrado al modulo seguridad definido por la arquitectura del sistema que se encargará de la asignación de los roles a cada uno de los usuarios. Además se encargará de la autenticación de los mismos brindando acceso a las funcionalidades y nivel que le pertenece.
- **Políticos-culturales:** La herramienta será utilizada solamente en el país y la podrán utilizar entidades autorizadas por el MINFAR. El producto no debe contener palabras en otros idiomas. Debe respetar los términos empleados normalmente por los especialistas en el tema de la esfera que se informatiza.
- **Legales:** El sistema debe regirse y cumplir las leyes y decretos estipulados, en los procesos que se automatizarán. Ley No.75 de la Defensa Nacional capitulo XV, establece los principios generales que rigen la creación de la Reserva Militar de Medios y Equipos de Economía Nacional para ser utilizada en interés de la defensa. Decreto-Ley No. 223 del Servicio Militar, de la Reserva Militar de Medios y Equipos de la Economía Nacional. Resolución No.48, para la organización del registro militar, empleo en la defensa y movilización de los medios y equipos de la reserva militar de la economía nacional.
- **Confiabilidad:** Deben existir mecanismos de restablecimiento de la herramienta en caso de fallos o problemas de comunicación. Deben instalarse sistemas de respaldo eléctrico en los servidores para el mantenimiento de los servicios en caso de fallos eléctricos.
- **Ayuda y documentación en línea:** Estará disponible en cada una de las interfaces, la ayuda correspondiente a dicha interfaz para que así los usuarios tengan un conocimiento de las funcionalidades de la misma y puedan hacer un mejor uso de estas.
- **Requerimientos de software**
 - ✓ **Para el cliente:** Navegador Mozilla Firefox y cualquiera de los sistemas operativos en los que se puede ejecutar Mozilla Firefox (Windows o Linux).
 - ✓ **Para el servidor:** Sistema operativo Windows o Linux. Un servidor WWW Apache v2.0 o superior con módulo PHP5 disponible, este debe estar configurado con las extensiones PDO y PDO_pgsq. Para servidor de base de datos PostgreSQL v4.1.0.1 o superior.
- **Requerimientos de hardware**

- ✓ **Para el cliente:** Requerimientos mínimos: Procesador Pentium IV a 3.00Ghz con 512 MB de memoria RAM. Tarjeta de red e impresora.
- ✓ **Para el servidor:** Requerimientos mínimos: Procesador Pentium IV a 3.00GHz y 1Gb de memoria RAM. Tarjeta de red.
- **Requerimientos del diseño y la implementación**
 - ✓ Como lenguaje del lado del cliente se uso el FrameWorks EXTJS con JavaScripts y del lado del servidor PHP v5.0 o superior. Para el servidor de base de datos PostgreSQL v4.1.0.1 o superior y servidor web Apache v2.0 o superior.

CONCLUSIONES

Con la realización de este capítulo se ha obtenido artefactos que ayudará a un mejor entendimiento del problema. Se realizó el modelo conceptual para lograr un mejor entendimiento del dominio del problema. Se han identificado los requisitos funcionales y no funcionales por los que el software podrá alcanzar la confiabilidad, robustez y la máxima conformidad a los clientes.

CAPÍTULO 3: ANÁLISIS Y DISEÑO DEL SISTEMA

INTRODUCCIÓN

En el presente capítulo se obtendrá una serie de artefactos como son los diagramas de secuencia y de clases web, así como la descripción de cada una de las clases. Para el diseño de la base de datos el diagrama de entidad relación será modelado mediante el Visual Paradigm, describiendo cada una de las tablas existentes en el diseño.

3.1 Diseño

El diseño del software permite traducir los requisitos analizados de un sistema, tanto funcionales como no funcionales, en una representación del software, que inicialmente da una visión del mismo y tras posteriores refinamientos sirve como esquema para la implementación del sistema y es, de ese modo, utilizado como una entrada fundamental de las actividades de implementación. Los objetivos del diseño son los siguientes:

- Crear una entrada apropiada y un punto de partida para actividades de implementación subsiguientes, capturando los requisitos o subsistemas individuales, interfaces y clases.
- Descomponer los trabajos de implementación en partes más manejables.
- Visualizar y reflexionar sobre el diseño utilizando una notación común.

3.1.1 *Diagrama de componentes*

Los diagramas de componentes describen los elementos físicos del sistema y sus relaciones. Muestran las opciones de realización incluyendo código fuente. Los componentes representan todos los tipos de elementos del software que entran en la fabricación de aplicaciones informáticas.

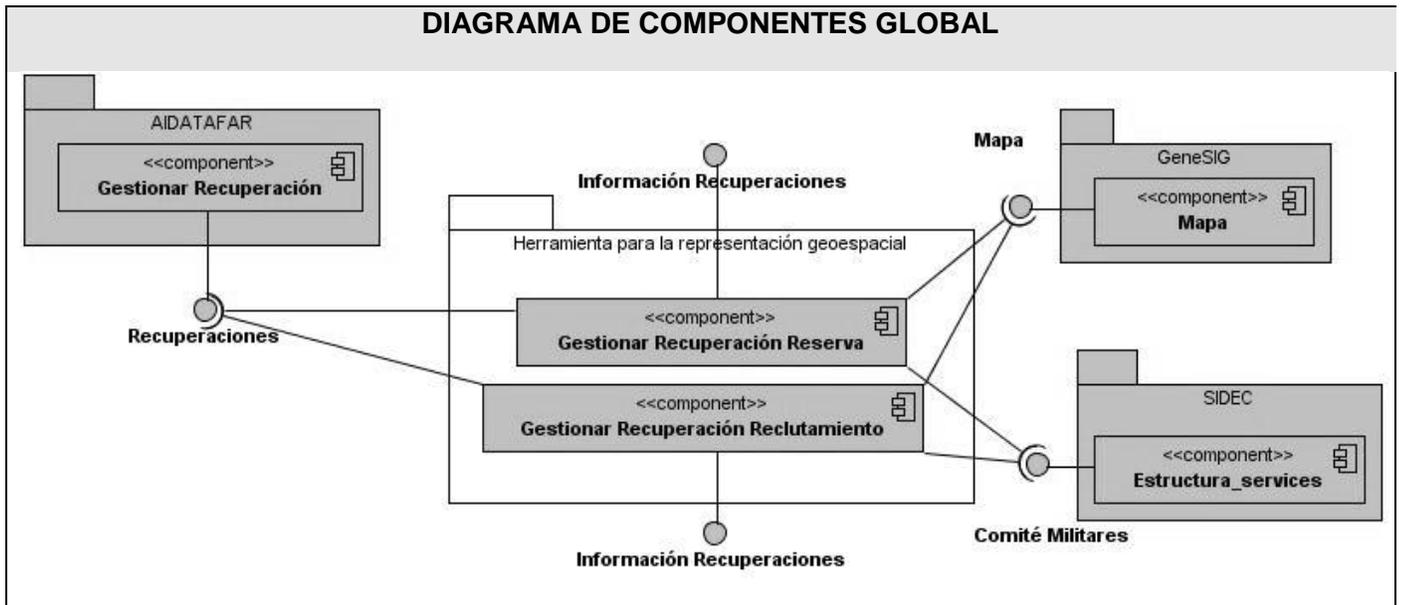


Figura 4.1: Diagrama de componentes.

4.1.1.1 Descripción del diagrama de componentes Global.

Como se muestra en la figura 4.1 los componentes internos Gestionar Recuperación de la Reserva y Gestionar Recuperación de Reclutamiento se relacionan con los componentes externos Estructura_services, Gestionar Recuperación y Mapa para cargar o los Comités Militares, Recuperaciones y Mapa para la representación respectivamente. La salida de los componentes internos sería la información a representar a través de una ventana.

3.1.2 Diagrama de clases web genérico

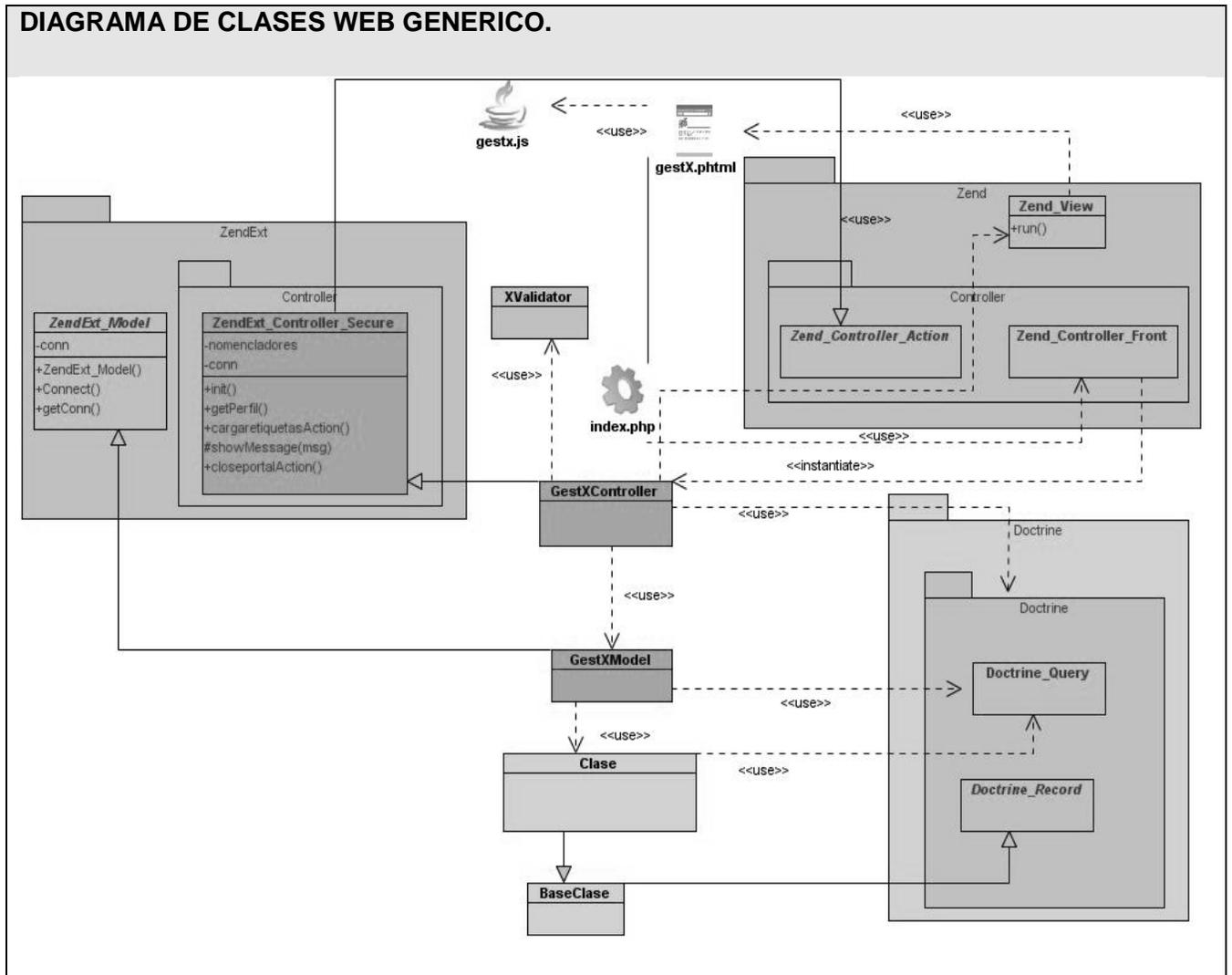


Figura 3.1: Diagrama de clases web genérico.

Se definió un diagrama de clases genérico con el objetivo de minimizar el trabajo y hacerlo de forma más eficiente, mantiene la comprensión del modelo. El mismo representa las clases del diseño fundamentales que participan en los procesos de negocio del sistema, siendo las clases GestXController, GestXModel, XValidator y gestx las que se modifican en dependencia procesos de negocio con el que se esté trabajando. La clase index constituye el único punto de acceso a la aplicación, conjuntamente con la clase

Zend_Loader y Zend_Controller_Front se encarga del funcionamiento de la aplicación, atención a solicitudes y respuestas.

Nombre: <i>ext-base.js</i>	
Tipo de clase: <i>Interfaz</i>	
Descripción:	Encargada del manejo de las solicitudes y respuestas, trabajo con ajax y manejo de componentes de EXT. Esta incluida en el paquete original.

Tabla 3.1: Ext-Base.

Nombre: <i>ext-all.js</i>	
Tipo de clase: <i>Interfaz</i>	
Descripción:	Es la encargada de la creación de los componentes visuales de la vista. Esta incluida dentro de las clases que trae EXT JS.

Tabla 3.2: Ext-All.

Nombre: <i>vista.phtml</i>	
Tipo de clase: <i>Interfaz</i>	
Descripción:	Representa la vista que se muestra al usuario.

Tabla 3.3: Vista.

Nombre: <i>vista.js</i>	
Tipo de clase: <i>Interfaz</i>	
Descripción:	Fichero js con las funciones Java Script asociadas a la vista. Aquí se establece la referencia a las clases de EXT.

Tabla 3.4: Vista-JS.

Nombre: <i>ucid.js</i>	
Tipo de clase: <i>Interfaz</i>	
Descripción:	Fichero js con las funciones Java Script comunes para todo el sistema.

Tabla 3.5: Ucid.

Nombre: <i>Zend_Controller_Action.php</i>	
Tipo de clase: <i>Controladora</i>	
Descripción:	De esta clase deben heredar todos los controladores de la aplicación, en ella se incluyen numerosas funcionalidades comunes.

Tabla 3.6: Zend_Controller_Action.

Nombre: <i>Zend_Controller_Secure.php</i>	
Tipo de clase: <i>Controladora</i>	
Descripción:	De esta clase deben heredar todas las controladoras. Es un controlador de acciones personalizado e integrado a seguridad.

Tabla 3.7: Zend_Controller_Secure.

Nombre: <i>vistaController.php</i>	
Tipo de clase: <i>Controladora</i>	
Descripción:	Representa el controlador del Caso de Uso en cuestión.

Tabla 3.8: Vista-Controller.

Nombre: <i>Zend_Loader.php</i>	
Tipo de clase: <i>Controladora</i>	

Descripción:	Se encarga de la carga dinámica de clases y ficheros. Es una utilidad de Zend Framework.
---------------------	--

Tabla 3.9: Zend_Loader.

Nombre: <i>Zend_View.php</i>	
Tipo de clase: <i>Controladora.</i>	
Descripción:	Es la clase de Zend Framework encargada del manejo de las vistas.

Tabla 3.10: Zend_View.

Nombre: <i>Zend_Controller_Front.php</i>	
Tipo de clase: <i>Controladora.</i>	
Descripción:	Representa el controlador frontal de la aplicación. Se encarga de manejar las solicitudes y respuestas. Es manejado por index.

Tabla 3.11: Zend_Controller_Front.

Nombre: <i>Index.php</i>	
Tipo de clase: <i>Servidora</i>	
Descripción:	Constituye el único punto de acceso a la aplicación, conjuntamente con la clase Zend_Loader y Zend_Controller_Front se encarga del funcionamiento de la aplicación, atención a solicitudes y respuestas.

Tabla 3.12: Index.

Nombre: <i>ZendExt_model.php</i>	
Tipo de clase: <i>Modelo</i>	

Descripción:	Se encarga de gestionar el modelo y el negocio.
---------------------	---

Tabla 3.13: ZendExt_model.

Nombre: <i>claseModel.php</i>	
Tipo de clase: <i>Modelo</i>	
Descripción:	En ella se implementan las funciones básicas para el trabajo con la base de dato como son insertar, modificar y eliminar.

Tabla 3.14: Clase-Model.

Nombre: <i>BaseClase.php</i>	
Tipo de clase: <i>Modelo</i>	
Descripción:	Representa la clase modelo del Caso de Uso, es generada dinámicamente por Doctrine, por lo general lleva el mismo nombre de la tabla asociada. Hereda de doctrine_record.

Tabla 3.15: Base-Clase.

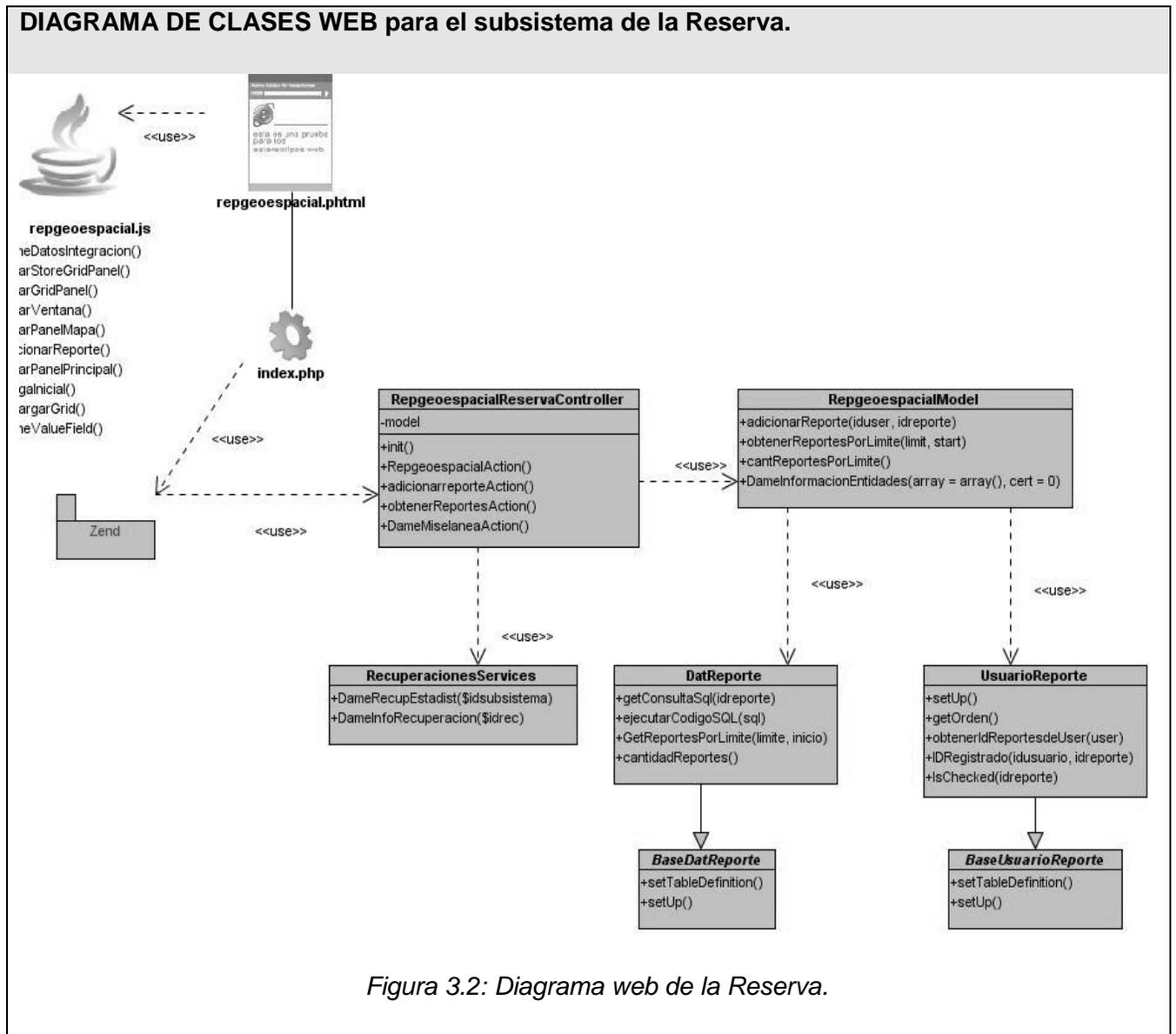
Nombre: <i>Clase.php</i>	
Tipo de clase: <i>Modelo</i>	
Descripción:	En ella se implementan todas las funcionalidades para obtener datos de la base de datos. Hereda de Base Clase

Tabla 3.16: Clase.

3.1.3 Diagrama de clases

En el diseño de clases se resume la definición de las clases que se pueden implementar en el software, se visualizan las relaciones entre ellas, y se muestra gráficamente la interacción de los objetos para comunicarse entre sí. [34]

A continuación se muestra una figura que explica el diseño de clase que se tomará como guía para la implementación de la aplicación. Dicho diseño se corresponde con el diagrama de clase web para la representación del subsistema de la reserva. Para el subsistema de Reclutamiento ([ver anexo 5](#)).



3.1.3.1 Descripción de las clases

Clases Interfaces:

Nombre: <i>repgeoespacial.js</i> .	
Tipo de clase: <i>Interfaz</i> .	
Descripción:	En la <i>repgeoespacial.js</i> se implementan todas las funciones Java Script asociadas a la vista. Aquí se establece la referencia a las clases de EXT. Esta clase es la encargada del diseño final de las interfaces, así como en la ejecución de operaciones básicas a través de la controladora para el funcionamiento de la aplicación.

Tabla 3.17: *Repgeoespacial.js*.

Nombre: <i>repgeoespacial.phtml</i> .	
Tipo de clase: <i>Interfaz</i> .	
Descripción:	Esta clase es la encargada de llamar a través de una dirección a <i>repgeoespacial.js</i> para así poder mostrar la vista final a los usuarios.

Tabla 3.18: *Repgeoespacial.phtml*.

Clases Controladoras:

Nombre: <i>RepgeoespacialReservaController</i> .	
Tipo de clase: <i>Controladora</i> .	
Atributo	Tipo

\$model	private
Para cada responsabilidad	
Nombre	Descripción
RepgeoespacialAction()	Es el encargado de mostrar la vista.
adicionarreporteAction()	Es el encargado de hacer uso de la operación implementada en la modelo para adicionar en una tabla de la Base Datos las recuperaciones que se desean representar en un momento dado.
obtenerReportesAction()	Es el encargado de mostrar en la interfaz de usuario todas las recuperaciones que fueron guardadas para su futura representación.
DameMiselaneaAction()	Es el encargado de devolverle al mapa los nombres de los CMM y CMP que se representarán en él.

Tabla 3.19: *RepgeoespacialReservaController*.

Clases Modelo:

Nombre: <i>RepgeoespacialModel</i> .	
Tipo de clase: <i>Modelo</i> .	
Atributo	Tipo
No Procede	No Procede
Para cada responsabilidad	

Nombre	Descripción
adicionarReporte(\$idUser, \$idreporte)	Es el encargado de adicionar en la base de datos la recuperación que se desea representar en el mapa.
obtenerReportesPorLimite(\$limit, \$start)	Es el encargado de devolver de la Base de datos un límite finito de recuperaciones para así poder mostrarlas en el grid (tabla en la interfaz) en diferentes paginados.
cantReportesPorLimite()	Es el encargado de devolver la cantidad existente de recuperaciones para el paginado.
DameInformacionEntidades(\$array = array(), \$cert = 0)	Esta operación es la encargada de que dada una cantidad de Comités Militares y un certificado de seguridad debe ser capaz conformar un resultado que es el que debe ser representado por parte de los mapas.

Tabla 3.20: RepgeoespacialModel.

Clases Entidad:

Nombre: <i>UsuarioReporte.</i>	
Tipo de clase: <i>Entidad.</i>	
Atributo	Tipo
No procede	No procede
Para cada responsabilidad	
Nombre (4)	Descripción (5)
setUp()	Constructor de la clase.

IDRegistrado(\$idusuario, \$idreporte)	Es el encargado de verificar si existe un usuario con la misma recuperación registrado.
IsChecked(\$idreporte)	Es el encargado de devolver la cantidad de recuperaciones dado un identificador.
getOrden()	Es el encargado de devolver en orden las recuperaciones representadas.
obtenerIdReportesdeUser(\$user)	Devolver el identificador de las recuperaciones dado un usuario.

Tabla 3.21: UsuarioReporte.

Nombre: <i>DatReporte</i> .	
Tipo de clase: <i>Entidad</i> .	
Atributo	Tipo
No procede	No procede
Para cada responsabilidad	
Nombre	Descripción
setUp()	Constructor de la clase.
getConsultaSql(\$idreporte)	Es la encargado de devolver el valor de los campo donde se encuentran las sentencias select y where.
ejecutarCodigoSQL(\$sql)	Es el encargado de ejecutar el código sql.
GetReportesPorLimite(\$limite, \$inicio)	Es el encargado de devolver las recuperaciones por límites.

cantidadReportes()	Es el encargado de devolver la cantidad existente de las recuperaciones.
--------------------	--

Tabla 3.22: *DatReporte*.

Clases Bases:

Nombre: <i>BaseUsuarioReporte</i> .	
Tipo de clase: <i>Base</i> .	
Atributo	Tipo
No procede	No procede
Para cada responsabilidad	
Nombre (4)	Descripción (5)
setUp()	Constructor de la clase.
setTableDefinition()	Es generado dinámicamente por Doctrine. En el se definen tantas variables como tenga la tabla asociada de la base de datos.

Tabla 3.23: *BaseUsuarioReporte*.

Nombre: <i>BaseDatReporte</i> .	
Tipo de clase: <i>Base</i> .	
Atributo	Tipo
No procede	No procede

Para cada responsabilidad	
Nombre (4)	Descripción (5)
setUp()	Constructor de la clase.
setTableDefinition()	Es generado dinámicamente por Doctrine. En el se definen tantas variables como tenga la tabla asociada de la base de datos.

Tabla 3.24: BaseDatReporte.

Clases Servicios:

Nombre: <i>RecuperacionesServices.</i>	
Tipo de clase: <i>Servicios.</i>	
Atributo	Tipo
No procede	No procede
Para cada responsabilidad	
Nombre (4)	Descripción (5)
DameRecupEstadist(\$idsistema)	Servicio que brinda todas las recuperaciones a representar.
DameInfoRecuperacion(\$idrec)	Servicio que devuelve el valor de la consulta realizada a cada recuperación para su posterior representación.

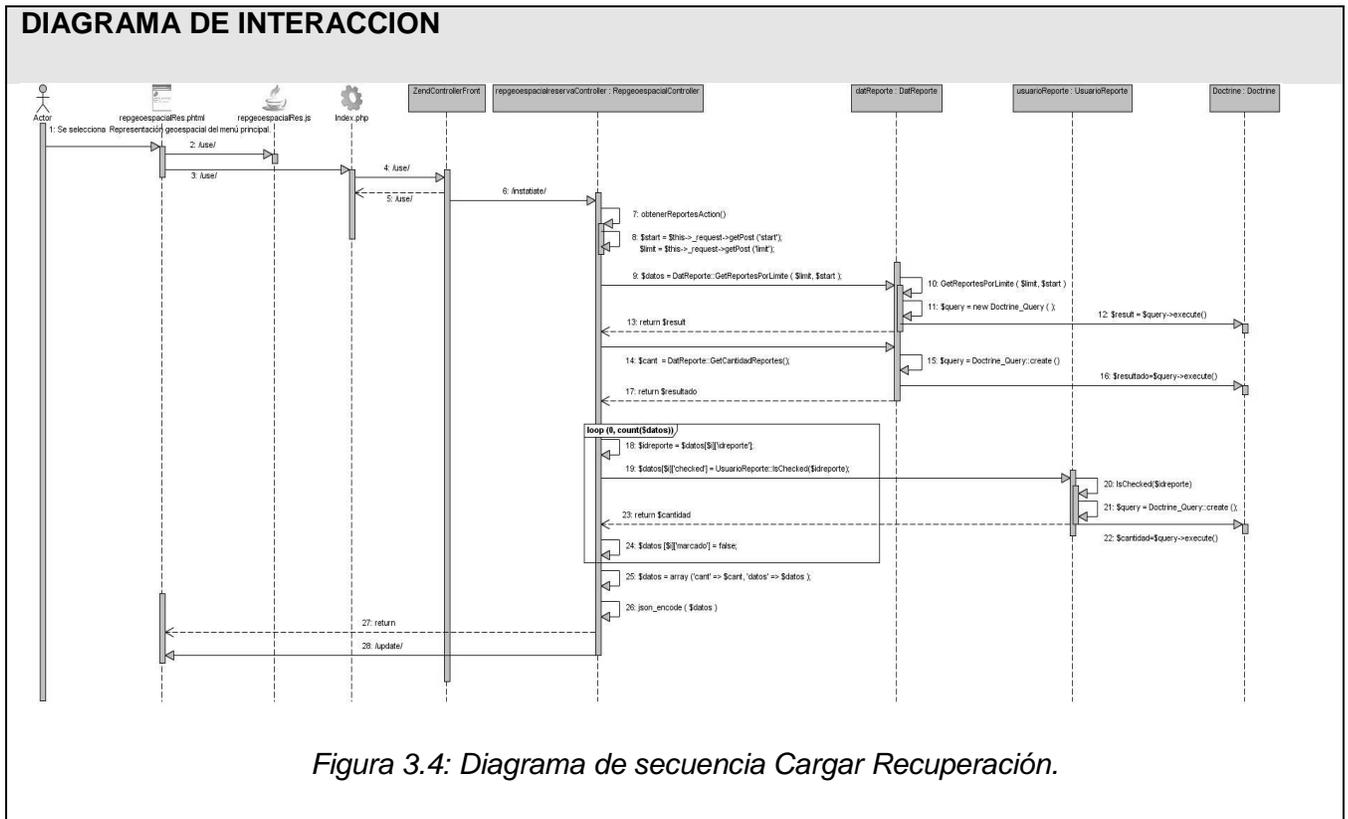
Tabla 3.25: RecuperacionesServices.

3.1.4 Diagramas de interacción

El objetivo de un diagrama de interacción es mostrar la interacción entre los usuarios, la interfaz y las instancias de los objetos en el sistema. Provee un mapa secuencial del paso de los mensajes entre los objetos a lo largo del tiempo.

Los diagramas de secuencia para la representación geoespacial del subsistema Reclutamiento y Reserva son los siguientes:

El siguiente diagrama de secuencia que se muestra a continuación muestra una serie de pasos que se llevan a cabo para Cargar las Recuperaciones que se desean mostrar. Así como la interrelación entre las clases. El usuario primeramente debe hacer uso de la clase `repgeoespacial.phtml` y esta a su vez hará uso de la `repgeoespacial.js` que será la encargada de poner en marcha una serie de operaciones necesarias e indispensables para la funcionalidad.



Para obtener más información sobre los diagramas secuencia para cada requisito ([ver anexos 6, 7](#)).

3.2 Diseño de la BD

3.2.2 Diagrama Entidad Relación de la BD

Los diagramas entidad relación son un lenguaje gráfico para describir conceptos. Son simples dibujos o gráficos que describen la información que trata un sistema de información y el software que lo automatiza. Es una herramienta para el modelado de datos de un sistema de información. Estos modelos expresan entidades relevantes para un sistema de información así como sus interrelaciones y propiedades.

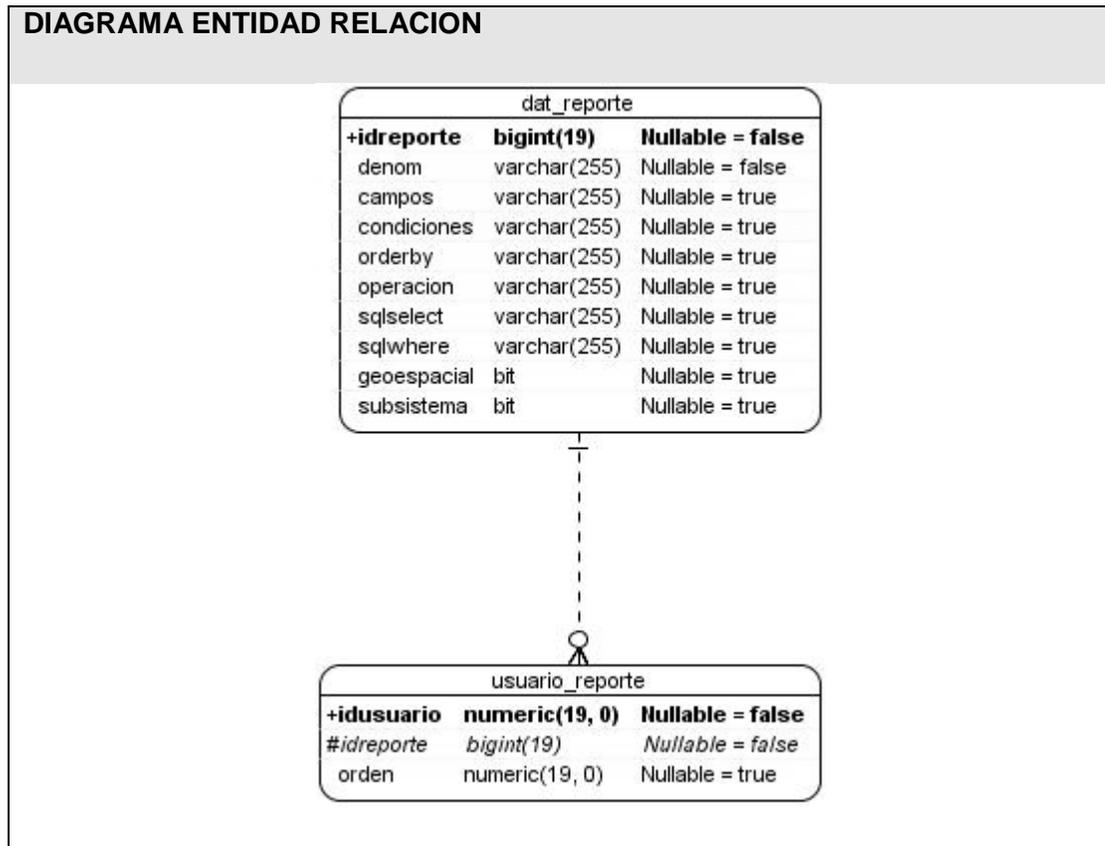


Figura 3.9: Diagrama Entidad Relación.

3.1.2 Descripción de las tablas

Nombre: *dat_reporte*.

Descripción: La tabla es la encargada de guardar los campos correspondientes de una recuperación que se desea mostrar en el futuro en el mapa de Cuba.

Atributo	Tipo	Descripción
Id_reporte	bigint	Es la llave primaria de cada reporte para su identificación.

denom	varchar	Denominación o nombre de la recuperación.
campos	varchar	Campos a mostrar de cada recuperación.
condiciones	varchar	Condiciones que deben cumplir cada recuperación ya sea igualdad o desigualdad de cualquier atributo, mayor que o menor que.
orderby	varchar	Para ordenar el resultado de la recuperación por lo deseado.
operacion	varchar	Este campo guarda cada el código sql utilizado en la recuperación para el realizar alguna operación como es por ejemplo contar.
sqlselect	varchar	Este campo guarda cada el código sql utilizado en la recuperación para el select.
sqlwhere	varchar	Este campo guarda cada el código sql utilizado en la recuperación para el where.
geoespacial	bit	Este campo guarda 0 o 1 para saber que recuperación será representada en el mapa. En caso de 0 es cuando único no se representará en el mapa.
subsistema	bit	Este campo guarda 0 o 1 para saber que recuperación pertenece a la reserva y cual a reclutamiento. En caso de 0 es reclutamiento y 1 es la reserva.

Tabla 3.26: dat_reporte.

Nombre: <i>usuario_reporte</i> .		
Descripción: La tabla es la encargada de guardar los campos correspondientes de la recuperación que ya se ha representado en el mapa y las que se desean representar en el instante.		
Atributo	Tipo	Descripción
Id_reporte	bigint	Es una llave foránea en esta tabla heredada de la tabla dat_reporte para identificar los reportes que le pertenecen a cada usuario.
Id_usuario	numeric	Identificador del usuario.
orden	numeric	Proporciona el orden de los reportes guardados.

Tabla 3.27: usuario_reporte.

3.3 Definiciones de diseño que se apliquen

El diseño está muy cercano a la fase de implementación, lo que es natural que se guarde y mantenga el modelo de diseño a través de todo el ciclo de vida del software. En el diseño se modela el sistema y se encuentra su forma (incluida la arquitectura) para que soporte todos los requisitos, incluyendo los no funcionales y las restricciones que se le suponen.

Dentro de los propósitos del diseño se puede encontrar:

- Adquirir una comprensión de los aspectos relacionados con los requisitos no funcionales y restricciones relacionadas con los lenguajes de programación, componentes reutilizables, sistemas operativos, tecnologías de distribución y concurrencia y tecnologías de interfaz de usuario.
- Crear una entrada apropiada y un punto de partida para actividades de implementación, capturando los requisitos o subsistemas individuales, interfaces y clases.

- Descomponer los trabajos de implementación en partes más manejables que puedan ser llevadas a cabo por diferentes equipos de desarrollo.
- Capturar las interfaces entre los subsistemas antes en el ciclo de vida del software, lo cual es muy útil cuando utilizamos interfaces como elementos de sincronización entre diferentes equipos de desarrollo.

3.4 Tratamiento de errores

En cada una de las funcionalidades del sistema se ha realizado un tratamiento de errores en caso de que lo requiera para poder tener constancia de la validación de los datos. Estos tratamientos de errores serán mostrados en las páginas clientes a través de mensajes de información.

3.5 Seguridad

El sistema debe ser capaz darle acceso a los usuarios según el nivel al que pertenecen, para esto se cuenta con el módulo seguridad que es el encargado de gestionar toda la seguridad que debe tener la aplicación. Para esto se le asignan a cada uno de los usuarios las funcionalidades a las que tendrá permiso a acceder. En cuanto a la seguridad del código se cuenta con el sistema de control de versiones Subversion que será el encargado de mantener una copia local y remota de los ficheros. El servidor de base de datos tiene instalado un firewall que permite reducir los riesgos de un atacante o hacker, se realizarán copias de seguridad de la base de datos para que en casos de desastres poder restablecer la información.

3.6 Interfaz

Las interfaces deben ser diseñadas a petición de los clientes, según el estándar de diseño predefinido por el Comité de rol. El resultado final de cada una de las interfaces debe ser fácil, amigable y sencillo, para una mejor interacción por parte de los clientes finales. Los botones estarán acompañados por íconos y nombres que identificarán la acción. Se usarán colores adecuados para lograr un mejor diseño.

3.7 Concepción de la ayuda

La concepción de la ayuda será de vital importancia para los usuarios que tendrán una interacción directa con el software. En todas las interfaces se dispondrá de un botón ayuda para un mejor aprovechamiento

de todas las funcionalidades que el sistema brinda. El sistema podrá ser usado por cualquier persona con conocimiento básico de informática porque la ayuda le servirá de guía a través de imágenes y números consecutivos que describirán claramente la acción de un botón o funcionalidad.

CONCLUSIONES

En este capítulo se concluyó que la obtención del diseño propuesto fue muy importante desde el punto de vista que ayudó en gran medida a la correcta conformación de los componentes a implementar, así como todo lo relacionado con los requisitos funcionales y no funcionales. La arquitectura escogida fue la más apropiada, que se basa en la integración de Zend Framework, Doctrine y el Framework EXTJS, puesto que implementa el patrón arquitectónico modelo, vista, controlador (MVC). Es un estilo basado en un patrón de diseño que plantea la separación de diferentes clases en dependencia de la función que realizan de modo tal que sea posible manejar dinámicamente la forma en que se procesan solicitudes y se gestiona la manera en que se muestran resultados al usuario final.

CAPÍTULO 4: IMPLEMENTACIÓN Y PRUEBAS

INTRODUCCIÓN

En el presente capítulo se obtendrá a partir de la implementación una serie de artefactos que serán los encargados de darle continuidad y cumplimiento al mismo. Dentro de los artefactos se encuentran el diagrama de componentes, además de la obtención de matriz de integración de componentes interna y externa. Para la realización de pruebas de calidad se describen los diseños de casos de prueba empleados para la realización de las pruebas y resultado de las mismas.

4.1 Implementación

4.1.1 Matriz de integración de componentes

La matriz de integración de componentes contiene todos los componentes definidos en el subsistema, de forma matricial, y en las intercepciones se especificarán los servicios que consume el componente en la vertical del horizontal. Existen dos matrices de integración de componentes, Interna y Externa.

4.1.2 Matriz de Integración de Componentes Interna

	Componentes Internos que consumen servicios.	
Componentes Internos que brindan.	Gestionar Recuperación Reserva	Gestionar Recuperación Reclutamiento
AIDATAFAR	DameRecupEstadist(\$idsistema) DameInfoRecuperacion(\$idrec)	DameRecupEstadist(\$idsistema) DameInfoRecuperacion(\$idrec)

Tabla 4.1: Matriz de Integración de Componentes Interna.

4.1.3 Matriz de Integración de Componentes Externa

	Componentes externos que brindan el servicio
Componentes Internos	Estructura y Composición
Gestionar Recuperación Reserva	getArbol(\$idestructura)
Gestionar Recuperación Reclutamiento	getArbol(\$idestructura)

Tabla 4.2: Matriz de Integración de Componentes Externa.

4.2 Prueba

Desde el inicio del desarrollo de los softwares los errores pueden empezar a darse desde el primer momento del proceso en el que los objetivos pueden estar especificados de forma errónea e imperfecta; así en los posteriores pasos del 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.

Las pruebas son una actividad en la cual un sistema o componente es ejecutado bajo unas condiciones o requerimientos especificados, los resultados son observados y registrados, y una evaluación es hecha de algún aspecto del sistema o componente. La prueba de software es 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.

La creciente inclusión del software como un elemento más de muchos sistemas y la importancia de los costos asociados a un fallo del mismo, han motivado la creación de pruebas más minuciosas y bien planificadas. No es raro que una organización invierta el 40% del esfuerzo de un proyecto en la prueba.

Un diseño de casos de prueba es el método de prueba que se aplicará a nuestro software, en este caso se realizará el método de Caja Negra que se refieren a las pruebas que se llevan a cabo sobre la interfaz del software. O sea, pretenden demostrar que las funciones del software son operativas, que la entrada se acepta de forma adecuada y que se produce una salida correcta.

A continuación se muestra la descripción del diseño de caso de prueba del requisito funcional **Guardar**

Recuperación:

Descripción de las variables

No	Nombre de campo	Clasificación	Puede ser nulo	Descripción
1.	NA	NA	NA	NA

Tabla 4.3: Descripción de las variables.

Requisitos a probar

Nombre del requisito	Descripción general	Escenarios de pruebas	Flujo del escenario

<p>Guardar Recuperación.</p>	<p>El objetivo de este requisito es guardar Recuperaciones.</p>	<p>EP 1.1 Guardar Recuperación</p>	<ol style="list-style-type: none"> 1. Se selecciona la opción Análisis de la información/ Reserva/ Representación geoespacial del menú principal. 2. Se muestra la interfaz Representación geoespacial. 3. Se oprime el botón Seleccionar reportes. 4. Se muestra la ventana Reportes. 5. Se selecciona en la lista las recuperaciones que se desea representar. 6. Se habilita el botón Aceptar. 7. Se oprime el botón Aceptar. 8. Se guarda la recuperación seleccionada y se muestra un mensaje de información indicando que se guardó satisfactoriamente.
------------------------------	---	------------------------------------	---

		EP 1.2 Cancelar la operación.	<ol style="list-style-type: none"> 1. Se selecciona la opción Análisis de la información/ Reserva/ Representación geoespacial del menú principal. 2. Se muestra la interfaz Representación geoespacial. 3. Se oprime el botón Seleccionar reportes. 4. Se muestra la ventana Reportes. 5. Se oprime el botón Cancelar. 6. El sistema cancela la operación.
--	--	-------------------------------	---

Tabla 4.4: Requisitos a probar.

Juegos de datos a probar

Id del escenario	Escenario	Nombre	Respuesta del sistema	Resultado de la prueba
EP 1.1	Guardar recuperaciones.	NA	Guarda las recuperaciones seleccionadas y se muestra un mensaje de información indicando que se guardó satisfactoriamente.	Satisfactoria.

EP 1.2	Cancelar la operación.	NA	El sistema cancela las operaciones y cierra la ventana reportes.	Satisfactoria.
--------	------------------------	----	--	----------------

Tabla 4.5: Juegos de datos a probar.

Para la descripción del diseño de caso de prueba del requisito funcional **Cargar Recuperación** ([ver anexo 8](#)).

Para la descripción del diseño de caso de prueba del requisito funcional **Representar Recuperación** ([ver anexo 9](#)).

4.2.1 Descripción de los métodos empleados para la realización de las pruebas

El método empleado para la realización de las pruebas del software fue el método de Caja Negra el cual se refiere a las pruebas que se llevan a cabo sobre la interfaz del software, por lo que los casos de prueba pretenden demostrar que las funciones del software son operativas, que la entrada se acepta de forma adecuada y que se produce una salida correcta, así como que la integridad de la información externa se mantiene. Esta prueba examina algunos aspectos del modelo fundamentalmente del sistema sin tener mucho en cuenta la estructura interna del software.

Se centran principalmente en los requisitos funcionales del *software*. Estas pruebas permiten obtener un conjunto de condiciones de entrada que ejerciten completamente todos los requisitos funcionales de un programa. En ellas se ignora la estructura de control, concentrándose en los requisitos funcionales del sistema y ejercitándolos.

4.2.2 Resultado de las pruebas

Después de haber identificado e implementados los requisitos funcionales se diseñaron los casos de pruebas correspondientes para cada escenario de cada requisito. Se obtuvieron tres errores no significativos como por ejemplo en el diseño de las interfaces se encontraron dos faltas de ortografías y una falta en un mensaje de información.

CONCLUSIONES

Se realizó la implementación de los requisitos anteriormente identificado logrando el funcionamiento correcto de cada uno. Los artefactos desarrollados en este flujo de trabajo fueron el diagrama de componentes y una breve muestra de las pruebas realizadas al sistema. Al concluir este capítulo, se han cumplido satisfactoriamente los objetivos trazados. Se le r las pruebas de caja negra a los procesos implementados y se termina con el placer de obtener nuevas experiencias, además de adquirir nuevos conocimientos.

CONCLUSIONES GENERALES

A partir del estudio realizado de las necesidades existentes en los Comités s Militares y el MINFAR, se ha llegado a la conclusión de que la principal causa de los problemas, es la carencia de una herramienta que sea capaz de representar la información geoespacialmente de los subsistemas de Reclutamiento y Reserva.

Se realizó el diseño teórico de la investigación así como el estudio del estado del arte.

Se realizó el diseño de un producto informático que sea eficiente y capaz de crear una entrada apropiada y un punto de partida para actividades de implementación, capturando los requisitos o subsistemas individuales, interfaces y clases.

Se implementó la herramienta diseñada obteniendo como resultado un producto confiable que implementa políticas de seguridad para su acceso. Los requisitos funcionales cumplen con las expectativas logrando conformidad en los clientes.

Finalmente se realizan un conjunto de pruebas de caja negra al software en busca de posibles vulnerabilidades del mismo, con el objetivo de minimizar los posibles errores luego de la implantación del sistema en las entidades nacionales.

En resumen, se cumplió el objetivo propuesto de realizar la implementación de la herramienta para la representación geoespacial de la información de los subsistemas de Reclutamiento y Reserva, basado en los requisitos funcionales, obteniendo como resultado un producto confiable y eficientemente que ayude al proceso de toma de decisiones.

RECOMENDACIONES

Una vez vencidos todos los objetivos del presente trabajo de diploma se recomienda:

- Continuar el estudio de los SIG para un mejor entendimiento del funcionamiento y objetivos principales de los mismos.
- Aumentar las propiedades físicas de los ordenadores clientes y servidores para así aumentar rendimiento de la aplicación para que el tiempo de respuesta de las consultas a la base de datos se reduzca al máximo.
- Aumentar la integración de la aplicación a otros sistemas informáticos de la FAR, para así ayudar al proceso de toma de decisiones en otras esferas de la entidad. Un ejemplo de integración con otros sistemas informáticos podría ser el sistema de Medios y Equipos actualmente en desarrollo y Movilización que está por desarrollarse.
- Realizar pruebas experimentales al software en los Comités Militares para así encontrar no conformidades con el producto en explotación.

BIBLIOGRAFÍAS

- 1. UCID. 2009.** *Proceso de Desarrollo y Gestión de Proyectos de Software (1ra Versión)*. Ciudad de La Habana : s.n., 2009.
- 2. UCID. 2009.** *Estándar de Ingeniería de Software (1ra Versión)*. Ciudad de La Habana : s.n., 2009.
- 3. Larman, Craing.** *Uml y Patrones. Introducción al análisis y diseño orientado a objetos*. México: Prentice Hall, 2005.
- 4. S. Presuman, Roger.** *Ingeniería del software, Un enfoque práctico parte I y II*. La Habana: Félix Varela, 2005.
- 5. Javyser.** Formación y desarrollo. Manual de Microsoft Project 98.

REFERENCIAS BIBLIOGRÁFICAS

Trabajos citados

- [1]. Maptitude Sistema de Información Geográfica. *Maptitude Sistema de Información Geográfica*. [En línea] 2010. [Citado el: 3 de 9 de 2009.] http://www.caliper.com/Maptitude/InternationalEspa%C3%B1ol/software_de_cartograf%C3%ADa.htm.
- [2]. Google Earth. *Google Earth*. [En línea] 9 de 2009. [Citado el: 3 de 9 de 2009.] <http://earth.google.com/intl/es/>.
- [3]. Portal Geoespacial de la República de Cuba. [En línea] 2009. [Citado el: 4 de 9 de 2009.] <http://www.iderc.co.cu/>.
- [4]. Plataforma de servicios web para el procesamiento y gestión de la información geoespacial. [En línea] 2009. [Citado el: 4 de 9 de 2009.] http://bibliodoc.uci.cu/TD/TD_1494_08.pdf.
- [5]. Aprendizaje de MapInfo Profesional. [En línea] 2010. [Citado el: 4 de 9 de 2009.] <http://www.servimaps.com/ProgramasGratis/Tutorial/STARTTUT.HTM>.
- [6]. La Definición de Software Libre. [En línea] 2010. [Citado el: 4 de 9 de 2009.] <http://www.gnu.org/philosophy/free-sw.es.html>.
- [7]. Definición de Cliente/Servidor. [En línea] 2009. [Citado el: 4 de 9 de 2009.] http://alumno.ucol.mx/sadanary/public_html/bd/cs.htm.
- [8]. Definición Arquitectura Cliente Servidor. [En línea] 2009. [Citado el: 5 de 9 de 2009.] <http://www.monografias.com/trabajos24/arquitectura-cliente-servidor/arquitectura-cliente-servidor.shtml>.
- [9]. Ventajas y Desventajas de la Arquitectura Cliente Servidor. [En línea] 2010. [Citado el: 5 de 9 de 2009.] http://www.gratisblog.com/tecnologia_de_internet_lopez/i125057-ventajas_y_desventajas_de_la_arquitectura_clienteservidor.htm.
- [10]. ¿Qué es un Servidor Web (Web Servers)? *Definición de Servidor Web*. [En línea] 2010. [Citado el: 6 de 9 de 2009.] <http://www.masadelante.com/faqs/servidor-web>.
- [11]. Características. *Características*. [En línea] 2009. [Citado el: 6 de 9 de 2009.] <http://acsblog.es/articulos/trunk/LinuxActual/Apache/html/x31.html>.
- [12]. Definición de Browsers. *Definición de Browsers*. [En línea] 2010. [Citado el: 6 de 9 de 2009.] <http://www.mastermagazine.info/termino/4098.php>.

- [13]. Definición de HTML. *Definición de HTML*. [En línea] 2010. [Citado el: 6 de 9 de 2009.] <http://www.mastermagazine.info/termino/5286.php>.
- [14]. Definición Ajax. [En línea] 2009. [Citado el: 6 de 9 de 2009.]
- [15]. Definición de XML. [En línea] 2009. [Citado el: 7 de 9 de 2009.] <http://www.mastermagazine.info/termino/7292.php>.
- [16]. Que es PHP. [En línea] 2009. [Citado el: 7 de 9 de 2009.] <http://www.desarrolloweb.com/articulos/392.php>.
- [17]. ¿Qué es una Base de Datos? *Definición de Base de Datos*. [En línea] 2009. [Citado el: 7 de 9 de 2009.] <http://www.masadelante.com/faqs/base-de-datos>.
- [18]. Definición de SGBD. [En línea] 2010. [Citado el: 7 de 9 de 2009.] <http://www.alegsa.com.ar/Dic/sghd.php>.
- [19]. Sistema de Gestión de Base de Datos. [En línea] 2009. [Citado el: 8 de 9 de 2009.] http://www.igac.gov.co:8080/igac_web/UserFiles/File/ciaf/TutorialSIG_2005_26_02/paginas/ctr_sistemasdegestiondebasededatos.htm.
- [20]. ¿Por qué utilizar PostgreSQL? [En línea] 2010. [Citado el: 10 de 9 de 2009.] http://soporte.tiendalinux.com/portal/Portfolio/postgresql_ventajas_html.
- [21]. Herramientas CASE (Computer Aided Software Engineering). [En línea] 2009. [Citado el: 10 de 9 de 2009.] <http://html.rincondelvago.com/herramientas-case.html>.
- [22]. Herramientas CASE. [En línea] 2009. [Citado el: 10 de 9 de 2009.] Disponible en: http://html.rincondelvago.com/herramientas-case_5.html.
- [23]. Visual Paradigm for UML. [En línea] 2010. [Citado el: 10 de 9 de 2009.] [http://www.freedownloadmanager.org/es/downloads/Paradigma_Visual_para_UML_\(M%C3%8D\)_14720_p/](http://www.freedownloadmanager.org/es/downloads/Paradigma_Visual_para_UML_(M%C3%8D)_14720_p/).
- [24]. Visual Paradigm for UML. [En línea] 2010. [Citado el: 10 de 9 de 2009.] [http://www.freedownloadmanager.org/es/downloads/Paradigma_Visual_para_UML_\(M%C3%8D\)_14720_p/](http://www.freedownloadmanager.org/es/downloads/Paradigma_Visual_para_UML_(M%C3%8D)_14720_p/).
- [25]. Qué es Doctrine ORM. [En línea] 2010. [Citado el: 16 de 9 de 2009.] <http://www.tecnoretas.com/programacion/que-es-doctrine-orm/>.

- [26]. Zend Framework, una introducción. [En línea] 2010. [Citado el: 16 de 9 de 2009.] <http://techtastico.com/post/zend-framework-una-introduccion/>.
- [27]. Más sobre Zend Framework. [En línea] 2010. [Citado el: 16 de 9 de 2009.] <http://www.proyectosbds.com/?id=121>.
- [28]. RUP . [En línea] 2009. [Citado el: 16 de 9 de 2009.] <http://www.scribd.com/doc/12812403/Que-es-RUP>.
- [29]. Curso práctico de Modelado de Negocios con UML y BPMN. [En línea] 2009. [Citado el: 20 de 9 de 2009.] <http://www.milestone.com.mx/CursoModeladoNegociosBPMN.htm>.
- [30]. Definición de UML . [En línea] 2009. [Citado el: 20 de 9 de 2009.] <http://www.docirs.cl/uml.htm>.
- [31]. ¿Cuáles son las características que debe tener una herramienta UML? [En línea] 2009. [Citado el: 20 de 9 de 2009.] http://www.docirs.cl/caracteristica_herramienta_uml.htm.
- [32]. ¿What is Doctrine? [En línea] 2010. [Citado el: 22 de 9 de 2009.] <http://www.doctrine-project.org/>.
- [33]. Metodologías ágiles para el desarrollo de software: eXtreme Programming (XP). [En línea] Letelier, Patricio y Penadés, M^a Carmen, 2010. [Citado el: 23 de 9 de 2009.] <http://www.willydev.net/descargas/masyxp.pdf..>
- [34]. UCID. 2009. Proceso de Desarrollo y Gestión de Proyectos de Software (1ra Versión). Ciudad de La Habana : s.n., 2009.
- [35]. UCID. 2009 Estándar de Ingeniería de Software (Versión 1.1). Ciudad de la Habana: s.n., 2009.

GLOSARIO DE TÉRMINOS

API: interfaz de programación de aplicaciones, es el conjunto de funciones y procedimientos que ofrece cierta biblioteca para ser utilizado por otro software como una capa de abstracción.

C/S: Arquitectura cliente – servidor.

CGI: Es el medio de comunicación que emplea un servidor Web para enviar información útil en ambos sentidos, entre el visualizador (browser) y su propio programa de cómputo.

CMM: Comités s Militar Municipal.

CMP: Comités Militar Provincial.

CMs: Comités Militares.

Código abierto: (en inglés open source) es el término con el que se conoce al software distribuido y desarrollado libremente.

Frameworks: ayudan en el desarrollo de software, proporcionan una estructura definida la cual ayuda a crear aplicaciones con mayor rapidez. Los frameworks son desarrollados con el objetivo de brindarles a los programadores y diseñadores una mejor organización y estructura a sus proyectos.

MINFAR: Ministerio de las Fuerzas Armadas Revolucionarias.

Protocolo: Es un conjunto de reglas usadas por computadoras para comunicarse unas con otras a través de una red.

Representación geoespacial: Representación de la información de forma digital a través de mapas, gráficas o tablas.

SQL: lenguaje de consulta estructurado, es un lenguaje declarativo de acceso a bases de datos relacionales.

SSL: Implementa un protocolo de negociación para establecer una comunicación segura a nivel de socket (nombre de máquina más puerto), de forma transparente al usuario y a las aplicaciones que lo usan.

WWW: World Wide Web, cuya traducción podría ser Red Global.