



Universidad de las Ciencias Informáticas

Facultad 6

Título: Desarrollo de un Sistema de Información Geográfica utilizando el framework Geomoose.

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

Autor: Saily Jorge Rodríguez.

Tutor: Ing. Lilianne Martínez Ledea.

La Habana, 30 de Junio de 2011

“Año 53 de la Revolución”

“La llegada de los Sistemas de Información Geográfica (SIG), son el mayor avance en el manejo de la información geográfica desde la invención del mapa”.

Peter Burrough

Declaro que soy el único autor de este trabajo y autorizo a la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

Para que así conste firmo la presente a los 30 días del mes de junio del año 2011.

.....
Saily Jorge Rodríguez.

(Autor)

.....
Ing. Lilianne Martínez Ledea.

(Tutor)

Autor: Saily Jorge Rodríguez.

Tutor: Ing. Lilianne Martínez Ledea.

Graduado de Ingeniería en Ciencias Informáticas en Julio del 2010.

Profesor recién graduado en adiestramiento.

Dirección: Universidad de las Ciencias Informáticas (UCI), Edificio: 94, Apto: 105.

Teléfono Oficina: +53 – 7 – 8373782. Teléfono.

Teléfono Apto: +53 – 7 – 8373095 E-mail: lledea@uci.cu.

Síntesis del Tutor:

Profesión: Profesor recién graduado en adiestramiento de la Facultad 6 de la UCI.

Años de graduado: 1

A mis padres, por ser mi motor impulsor, por su preocupación cariño y amor.

A mi hermanita Susana, porque la quiero mucho. Le dedico este trabajo para que el empeño puesto en él, lo vea como un ejemplo a seguir en su vida.

A mi abuelo, que aunque no pudo compartir este momento conmigo, se que estaría muy orgulloso por los logros que he obtenido en mi vida.

Al Comandante Fidel Castro Ruz, por ser el creador principal de esta obra en la que me formé durante cinco años. A él y a esta Revolución, les dedico este trabajo.

Les dedico todo mi esfuerzo a todas las personas merecedoras de mi cariño y que me tienen de igual manera, como alguien importante en sus vidas.

Muchas gracias a mis padres por la educación que me han dado y por hacer de mí lo que soy ahora y a mi hermanita, por compartir conmigo en todo momento: son lo más importante que tengo. Gracias por haber permanecido conmigo en los buenos y malos momentos de mi vida. Son lo mejor del mundo.

Al resto de mi familia, que me apoya y han colaborado de una u otra forma en la realización de este trabajo.

A Leony, que pasó de ser mi amigo, a ser mi amigo y además mi compañero. Gracias por la paciencia que se que has tenido durante este periodo. Has sido, junto a mi familia, una de las personas más importantes y que más me ha apoyado.

A mis negros de Ilú Aché, por haber sido unas de las mejores experiencias que viví en esta universidad. Gracias a Herson por ayudarme a sacar lo más especial de mí.

A Rubén (Yainier), muchas gracias por sacarme de un gran aprieto y ayudarme incondicionalmente.

A los profesores Frank David y Romanuel, por colaborar conmigo en la elaboración de este trabajo.

A las Kankas (Adaily, Cecilia, Krysna, Ivelin, Lisandra, Yamilé, Yisell y Glenda), un gran grupo de loquitas con las que tuve la oportunidad de compartir estos dos últimos años.

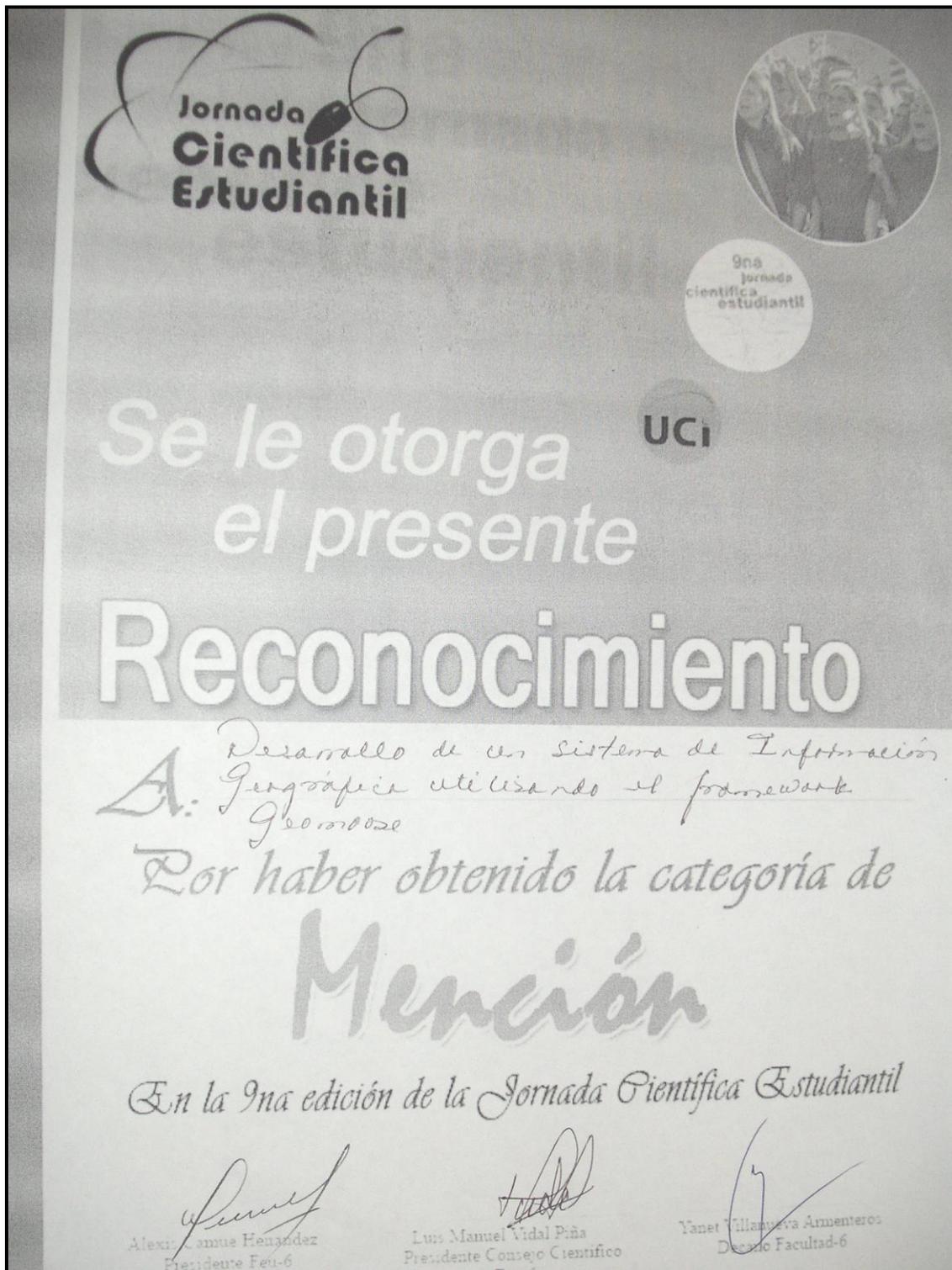
A Yari, que fue mi madre durante el tiempo que estuve en Venezuela. No sabes como te agradezco que hayas permanecido a mi lado y me hayas ayudado siempre que lo necesité. Eras el pedacito de Cuba que necesitaba tener estando tan lejos.

A mis amistades, que siempre me han apoyado a pesar de haber tomado rumbos diferentes en nuestras vidas.

A todas las personas que pude conocer durante todo este tiempo. Conocerlos a todos y tener una historia con cada uno de ellos, me hizo crecer como persona.

Al tribunal, oponente y tutora por ayudarme a la entrega de un documento con calidad.

Al Comandante y a la Revolución cubana, por darme la oportunidad de superarme y de formarme como profesional durante estos cinco años.



Resumen:

El presente trabajo de diploma, surge con el objetivo de que los usuarios interesados en aplicaciones de Sistemas de Información Geográfica (SIG) desarrolladas sobre el framework Geomoose, tengan a la mano la documentación técnica asociada al proceso de desarrollo de esta investigación. El documento fue elaborado utilizando toda la bibliografía disponible. Tanto artículos recientemente publicados, como sitios web y libros. Además, teniendo en cuenta las expectativas creadas actualmente sobre los SIGs y su utilización en Cuba y en el resto del mundo, es que la UCI se ha sumado al uso e implementación de estas tecnologías, por lo que es importante la elaboración de una investigación respecto al tema.

El departamento Geoinformática, que es donde se desarrolla este tipo de aplicaciones en la UCI, cuenta con la plataforma GeneSIG para la creación de SIG, desarrollada sobre el framework CartoWeb. Con el paso del tiempo y el surgimiento de nuevas tecnologías, han surgido diferentes problemas, entre ellos la necesidad de diversificar el desarrollo de las tecnologías empleadas en el departamento para la creación de SIG. Para llegar a una solución ante los inconvenientes planteados, se hace necesario desarrollar una investigación sobre el proceso de desarrollo de los SIG sobre el framework Geomoose. De esta manera se podrá agilizar la respuesta ante las peticiones de clientes interesados en soluciones SIG, sobre el framework Geomoose.

Con el presente trabajo, se obtuvo como resultado final, la documentación y artefactos generados durante la investigación, además de la aplicación, la cual se ajusta a un programa de pruebas después de ser implementada. Con estas pruebas realizadas, se verifica que se cumpla con los requisitos establecidos obteniendo una primera versión de un SIG genérico con funcionalidades básicas utilizando el framework Geomoose.

Palabras claves:

Sistema de Información Geográfica.

Aplicación Web.

Índice de Contenido:

Introducción	1
Capítulo 1: Fundamentación teórica	7
1.1 Introducción.....	7
1.2 Sistemas de Información.	7
1.2.1 Definición de Sistemas de Información.	7
1.2.2 Las actividades de los sistemas de Información.....	8
1.3 Sistema de Información Geográfica.	8
1.3.1 Definición.....	8
1.3.2 Conceptos básicos asociados a los Sistema de Información Geográfica....	9
1.3.3 Funcionalidades de los Sistema de Información Geográfica.	11
1.3.4 Componentes de los Sistemas de Información Geográfica.....	11
1.3.5 Tipos de programas gestores.	12
1.3.6 Técnicas utilizadas en los Sistemas de Información Geográfica	13
1.4 Análisis de las soluciones existentes.	15
1.5 Conclusiones parciales.....	17
Capítulo 2: Posibles tecnologías a utilizar	18
2.1 Introducción.....	18
2.2 Metodologías de desarrollo de software.	18
- Selección de la metodología de desarrollo de software a utilizar.	19
2.3 Lenguaje de Modelado.	19
- Selección del Lenguaje modelado (UML) a utilizar.	20
2.4 Herramientas de Ingeniería de Software Asistidas por Computadora.	20
- Selección de la Herramienta de Ingeniería de Software Asistida por Computadora a utilizar.....	21
2.5 Lenguajes de programación.	22
- Selección del lenguaje del lado del servidor a utilizar.	24
- Selección del lenguaje del lado del cliente a utilizar.....	25
2.6 Framework.....	25
2.6.1 Framework Geomoose.	25
- Selección del framework a utilizar.....	26

2.7	Sistemas de Gestión de Bases de Datos (SGBD).....	26
	- Selección del Sistema de Gestión de Bases de Datos (SGBD).	28
2.8	Entorno de Desarrollo Integrado.	28
	- Selección del Entorno de Desarrollo Integrado a utilizar.	29
2.9	Servidores de mapas.	29
	- Selección del servidor de mapas a utilizar.	30
2.10	Servidores Web.	30
	- Selección del servidor Web a utilizar.	31
2.11	Conclusiones parciales.	31
Capítulo 3: Análisis y diseño.		32
3.1	Introducción.	32
3.2	Descripción del Modelo de Negocio y del Modelo de Dominio.	32
3.3	Selección y explicación del modelo a utilizar.	33
3.4	Requerimientos funcionales y requerimientos no funcionales.	36
3.5	Diagrama de Casos de Usos del Sistema.	42
3.6	Prototipo de Interfaz de usuario.	45
3.7	Diagrama de Despliegue.	46
3.8	Arquitectura lógica.	50
3.9	Clases del diseño.	52
3.10	Diseño de la Base de Datos (BD).	56
3.11	Conclusiones parciales.	57
Capítulo 4: Implementación y pruebas.		58
4.1.	Introducción.	58
4.2.	Pruebas de software.	58
4.3.	Conclusiones parciales.	59
Conclusiones generales.		60
Recomendaciones.		62
Referencias bibliográficas.		63
Bibliografías consultadas.		65

Índice de figuras:

Ilustración 1 Modelo de Dominio.....	34
Ilustración 2 Diagrama de Casos de Uso del Sistema.....	43
Ilustración 3 Prototipo de interfaz de usuario.	45
Ilustración 4 Diagrama de Despliegue.	47
Ilustración 5 Servidor Web.....	48
Ilustración 6 Servidor de Base de Datos.	48
Ilustración 7 Servidor de Mapas.....	49
Ilustración 8 Cliente Web.	49
Ilustración 9 Arquitectura Cliente/Servidor.	51
Ilustración 10 Paquete de clases del diseño del framework Geomoose.....	53
Ilustración 11 Clases del diseño del CUS Realizar control de selección de capas.....	55

Índice de tablas:

Tabla 1 Comparación entre el Modelo de Dominio y el Modelo de Negocio.....	33
Tabla 2 Actores del sistema.....	42
Tabla 3 Descripción del Casos de Uso del Sistema Exportar mapa.....	44
Tabla 4 Resumen de los Casos de Uso del Sistema.....	45
Tabla 5 Secciones a probar en el Caso de Uso Realizar control de selección de capas.....	59
Tabla 6 Escenario 1 de la sección 1 del Caso de Uso Realizar control de selección de capas.....	59

Introducción

Desde los inicios de la humanidad, el hombre siempre ha evidenciado un gran interés por representar el lugar en el que habita. Destacando el afán de sentirse localizado en algún lugar en específico y la satisfacción de tener una idea exacta de hacia dónde iban o venían e incluso hacia dónde era más factible viajar. De esta manera, se convirtió en una necesidad el hecho de tener conocimiento de lo que lo rodeaba, la distancia que había entre diferentes lugares y se percató de que tener el conocimiento exacto de una ubicación, le facilitaba la vida en diferentes esferas de la sociedad. Poco a poco fueron surgiendo diferentes teorías, tales como que la tierra era plana o que el mundo tenía como límite hasta donde la vista alcanzaba ver una línea final en el mar. Todas estas teorías eran representadas en los dibujos y los mapas para poderse guiar. La necesidad de conocimiento del hombre fue más allá de las simples teorías y comenzó a trabajar en función de probarlas. Viajar por el mundo y utilizar las tecnologías que simultáneamente iban surgiendo en ese momento, era su forma de demostrar lo que tanto se especulaba. Finalmente, la afirmación más real era que la tierra, además de tener su forma esférica, no solo se podía representar con los simples mapas y dibujos, sino que también se podía representar a través de globos terráqueos por su forma redondeada y más adelante se podría representar en un mapa señalando todos sus detalles gracias a la utilización de las computadoras.

De esta manera, con el avance de las Tecnologías de la Información y la Comunicación (TIC)¹ y la revolución digital, surgen, a mediados de los años 1950 hasta los 70 los primeros esbozos de los Sistemas de Información Geográfica (SIG), que permiten que la información de este tipo sea más accesible para las personas con acceso a Internet; pero rápidamente sobre los años 70, se impone una actitud corporativa en su desarrollo. Luego, en la década de los 80, aparece la época comercial de los SIG, en la cual éstos se expanden a nivel internacional. Por último, ya en los 90, aparece la llamada etapa del usuario. Es en esta etapa donde, debido a la gran expansión y competencia de los SIG, las empresas comerciales comienzan a dar un lugar cada vez más importante al usuario, personalizando los SIG en dependencia de las demandas y requerimientos que estos pedían.

¹ Tecnologías de la Información y la Comunicación (TIC): Son el conjunto de tecnologías desarrolladas para gestionar información y enviarla de un lugar a otro. Incluyen las tecnologías para almacenar información y recuperarla después, enviar y recibir información de un sitio a otro, o procesar información para poder calcular resultados y elaborar informes.

Sería válido especificar que “un SIG se puede definir como aquel método o técnica de tratamiento de la información geográfica que permite combinar eficazmente información básica para obtener información derivada. Para ello, se contará tanto con las fuentes de información como con un conjunto de herramientas informáticas (hardware y software) que facilitarán esta tarea; todo ello enmarcado dentro de un proyecto que habrá sido definido por un conjunto de personas, y controlado, así mismo, por los técnicos responsables de su implantación y desarrollo” **(Comas, 2007)**.

En resumen, un SIG no es más que aquella integración entre elementos de hardware y software que permiten la gestión de la información geográficamente referenciada. Los SIG o GIS² han posibilitado un gran avance para el manejo de los temas de Geografía, debido a que son sistemas informáticos que brindan la posibilidad de almacenar y manipular datos que tengan un componente geográfico, permitiendo mayor eficiencia en el desarrollo de estos temas. En la actualidad son muy utilizados en distintas esferas de la sociedad debido al importante impacto social y económico que tienen, tanto en servicios que se le brindan a la población, como en controles referentes a la seguridad. Se considera que los SIG se han convertido en una herramienta imprescindible para todas aquellas personas que utilizan información geográfica, por lo que en los últimos decenios el tratamiento geográfico de la información ha cobrado un auge vertiginoso a escala mundial.

Las expectativas creadas sobre los SIGs están también presentes en Cuba con sus correspondientes limitaciones y paradojas. “A pesar de que existen SIG creados y desarrollados por empresas españolas, holandesas, brasileñas, entre otras, que podrían adquirirse con sus licencias, una gran parte de los especialistas cubanos prefieren los productos de las compañías de EE.UU., que se dedican al desarrollo de software para SIG. De los cientos de SIG que existen en estos momentos a nivel mundial, solamente unos pocos son conocidos y utilizados en Cuba. Es conveniente aclarar que todos los SIG diseñados y creados en EE.UU., “adquiridos” en Cuba por distintas vías, no tienen licencia de utilización de usuario debido a que las compañías estadounidenses no están autorizadas a vender estos software a los cubanos” **(Silva, 2005)**.

Por tales motivos, es que la Universidad de las Ciencias Informáticas (UCI), que es la universidad prevista para la formación integral y continua de profesionales de la informática en Cuba, se ha sumado al uso e

² GIS: Geographic Information System; es el acrónimo en inglés de Sistema de Información Geográfica.

implementación de estas tecnologías, surgiendo en la Facultad 6, el Centro GEySED, el cual posee dos áreas de conocimiento muy bien definidas internacionalmente, el departamento de Geoinformática y el de Señales Digitales.

El departamento de Geoinformática persigue como objetivo poder contar con herramientas cubanas que permitan el manejo de la información georeferenciada. Además de contribuir con aquellas entidades que gestionen este tipo de información y que se encuentren inmersas en un proceso de informatización y migración hacia el software libre.

A pesar de que este departamento cuenta con la plataforma GeneSIG para la creación de SIG, desarrollada sobre el framework CartoWeb, en el departamento de Geoinformática se tiene en cuenta un aspecto importante. Por el avance tecnológico y el surgimiento de nuevos framework que brindan la posibilidad de un mejor rendimiento de los SIGs, se puede prever que los usuarios comiencen a solicitar SIG en esas nuevas tecnologías.

Por tanto, se ha hecho necesario diversificar el desarrollo de las tecnologías empleadas en el centro GEySED y para lograrlo se hace necesario el estudio de las mismas, para la creación de SIG, específicamente sobre el framework Geomoose.

Este framework, está tomando un gran auge, ya que entre una de sus principales características tiene que trae consigo muchas herramientas, las cuales funcionan como servicios. Cada vez que algún servicio es usado este se ejecuta de forma independiente en el servidor, lo que hace que la respuesta a las peticiones realizadas, sean mucho más rápidas. Geomoose tiene también como característica, que permite manejar un gran número de capas, de usuarios y de servicios, lo que posibilita gestionar gran cantidad de información. Estas características solo son algunas de las principales, que hacen de este framework una herramienta con novedosas funcionalidades. Además, GeneSIG, al ser la primera solución de este tipo que se implementa en la UCI y al no existir ninguna experiencia previa a su desarrollo, tiene un gran inconveniente: es una plataforma muy pesada en código. Esto implica que cada vez que se utiliza para el desarrollo de otros SIG, los desarrolladores tienen que cambiar mucho código. En cambio, al utilizar un SIG desarrollado sobre Geomoose, se permite la reutilización de algoritmos por lo que la implementación de los SIG es mucho más cómoda. Además teniendo en cuenta que no existen en la UCI estudios previos de esta nueva herramienta, se considera que la investigación, acompañada de la aplicación desarrollada, constituirá una fuente de apoyo para futuras implementaciones de este tipo. Se desea adquirir experiencia

en el empleo de las mismas, para que otros desarrolladores obtengan un producto eficaz de la manera más rápida posible para darles respuesta a las personas o entidades interesadas en el tema. Esto permitirá la implementación de productos SIG utilizando el framework Geomoose a usuarios que así lo soliciten.

Además teniendo en cuenta la situación en Cuba, siendo víctima de un bloqueo económico, resalta como una característica importante, que este framework es un software no privativo y las herramientas que se utilizan simultáneamente al trabajo con este, tienen licencia no privativa o tienen menos restricciones que las de las demás aplicaciones del mismo tipo.

Teniendo en cuenta lo expuesto anteriormente, el **problema a resolver** queda formulado por la siguiente interrogante: ¿Cómo agilizar la respuesta ante las peticiones de clientes interesados en soluciones SIG, sobre el framework Geomoose? Para dar solución a la problemática descrita se ha planteado como **objetivo general**: Desarrollar una solución SIG con el framework Geomoose que brinde las funcionalidades básicas de estos sistemas y pueda usarse como plataforma para la implementación de soluciones similares.

Seleccionando como **objeto de estudio**: El proceso de desarrollo de los Sistemas de Información Geográfica. Teniéndose como **campo de acción**: El desarrollo de los Sistemas de Información Geográfica sobre el framework Geomoose. Durante el transcurso de esta investigación se sostendrá la siguiente **idea a defender**: Con el desarrollo de un SIG empleando el framework Geomoose se logrará agilizar la implementación de soluciones similares en el Departamento de Geoinformática.

Para guiar la investigación y brindar una solución al problema planteado anteriormente, fueron planificadas diferentes tareas, las cuales se mencionan a continuación.

Tareas de la investigación:

1. Caracterizar los conceptos más importantes asociados a los Sistemas de Información Geográfica.
2. Caracterizar los Sistemas de Información Geográfica que hayan sido desarrollados sobre el framework Geomoose.
3. Fundamentar las tendencias actuales y las tecnologías asociadas al proceso de desarrollo de los Sistemas de Información Geográfica.
4. Fundamentar el método, las herramientas y las tecnologías a utilizar para el desarrollo del sistema.

5. Identificar las funcionalidades que debe brindar el sistema.
6. Elaborar el análisis y el diseño del sistema según los requerimientos identificados.
7. Realizar la implementación del sistema en correspondencia con las funcionalidades que debe brindar.
8. Realizar las pruebas del sistema mediante el uso de técnicas para este fin.

Esperando que esta investigación arroje como **posibles resultados**:

1. Aplicación SIG desarrollada con el framework Geomoose.
2. Documentación técnica asociada al proceso de desarrollo de la aplicación.

Para el cumplimiento de las tareas propuestas se utilizaron los métodos científicos de investigación, los cuales se clasifican en teóricos y empíricos.

Los **métodos teóricos** permiten estudiar las características del objeto de investigación que no son observables directamente. Además facilitan la construcción de modelos e hipótesis de investigación y crean las condiciones para ir más allá de las características fenomenológicas y superficiales de la realidad, contribuyendo al desarrollo de las teorías científicas y para su ejecución se apoyan en el proceso de análisis y síntesis (**Sayaz, 1995**). Estos participan en la etapa de asimilación de hechos, fenómenos y procesos y en la construcción del modelo e hipótesis de investigación. Dentro de los métodos teóricos, se utilizaron en esta investigación el método analítico-sintético y el método histórico-lógico.

- **Método analítico-sintético:** Este método se emplea para consultar la bibliografía especializada en cuanto a los Sistemas de Información Geográfica e identificar elementos claves que contribuyan a la solución del problema planteado. En la investigación este método se utiliza en el Capítulo 1, para sintetizar conceptos que no deben dejar de ser mencionados, pues son los que permitirán la mejor comprensión de toda la investigación que se realizará posteriormente. Además se utiliza éste método en el Capítulo 2 para plantear las características específicas y esenciales que justifican la solución propuesta.
- **Método histórico-lógico:** Permite estudiar la trayectoria histórica real de los fenómenos, su evolución y desarrollo. Se emplea en la investigación para identificar cómo han evolucionado

desde su surgimiento hasta la actualidad, los Sistemas de Información Geográfica. Estos elementos identificados se ven reflejados en la introducción de la investigación.

- **Método de modelación:** Este método se utiliza para reproducir el fenómeno que se ha investigado. Por lo que al utilizarlo se llega a la materialización de la investigación, obteniendo la aplicación como resultado de esta. Fue empleado en el capítulo 3 de la investigación.

Otro de los métodos científicos de investigación utilizados es el empírico, dentro del cual se utilizó la observación. Los **métodos empíricos** permiten revelar y explicar las características fenomenológicas del objeto. Este se emplea fundamentalmente en la primera etapa de acumulación de información empírica y en la de comprobación experimental de la hipótesis de trabajo (**Sayaz, 1995**). Dentro de los métodos empíricos se empleó la observación.

- **Observación:** Este método se emplea porque permitirá obtener información directa e inmediata en el proceso de desarrollo de los Sistemas de Información Geográfica, específicamente de los SIG desarrollados utilizando el framework Geomoose y de esta manera, plantear una mejor solución en el capítulo 2.

Capítulo 1: Fundamentación teórica.

1.1 Introducción.

En este capítulo se abordan los conceptos más importantes asociados al proceso de desarrollo de los Sistemas de Información Geográfica, que desde el punto de vista teórico, permiten un mejor entendimiento de la investigación en sentido general. Además se realiza un profundo análisis de las soluciones existentes en Cuba y en el mundo con relación a la investigación. Teniendo en cuenta el estudio del estado del arte y la caracterización los SIG desarrollados sobre el framework Geomoose, se podrá dar continuación a la investigación.

1.2 Sistemas de Información.

1.2.1 Definición de Sistemas de Información.

Un sistema de información es el conjunto de elementos que interactúan entre sí con el fin de apoyar las actividades de una empresa o negocio. Teniendo muy en cuenta el equipo computacional necesario para que el sistema de información pueda operar y el recurso humano que interactúa con el Sistema de Información, el cual está formado por las personas que utilizan el sistema (**Peralta, 2008**).

El estudio de los sistemas de información surgió como una subdisciplina de las ciencias de la computación, con el objetivo de racionalizar la administración de la tecnología dentro de las organizaciones. Los sistemas de información son un sistema de personas, registros de datos, actividades, en fin, elementos que procesan los datos y la información con el propósito de prestar atención a las demandas de información de una organización, para elevar el nivel de conocimientos que permitan un mejor apoyo a la toma de decisiones y desarrollo de acciones, incluyendo manuales de procesos o procesos automatizados.

Cumplen tres objetivos básicos dentro de las organizaciones, estos son:

- Automatizar procesos operativos.
- Proporcionar información que sirva de apoyo al proceso de toma de decisiones.
- Lograr ventajas competitivas a través de su implantación y uso.

1.2.2 Las actividades de los sistemas de Información.

Un sistema de información realiza cuatro actividades básicas: entrada, almacenamiento, procesamiento y salida de información.

- **Entrada de Información:** Es el proceso mediante el cual el Sistema de Información toma los datos que requiere para procesar la información. Las entradas pueden ser manuales o automáticas. Las unidades típicas de entrada de datos a las computadoras son las terminales, las cintas magnéticas, las unidades de diskette, los escáneres, el teclado, el mouse, entre otras.
- **Almacenamiento de información:** A través de ésta, el sistema puede recordar la información guardada en la sección o proceso anterior. Esta información suele ser almacenada en archivos.
- **Procesamiento de Información:** Es la capacidad del Sistema de Información para efectuar cálculos de acuerdo con una secuencia de operaciones preestablecida.
- **Salida de Información:** Es la capacidad de un Sistema de Información para sacar la información procesada al exterior. Las unidades típicas de salida son las impresoras, terminales, diskettes, cintas magnéticas, los plotters, entre otros.

1.3 Sistema de Información Geográfica.

1.3.1 Definición

Un SIG es una integración muy organizada de software, hardware y datos geográficos, diseñado para capturar, analizar, almacenar, manipular y desplegar la información geográficamente referenciada. Puede definirse también como un modelo de una parte de la realidad referido a un sistema de coordenadas terrestres, construido principalmente para satisfacer la necesidad de información y ubicación geográfica del mundo. Estos deben ser capaces de ubicar un objeto determinado en el espacio; de encontrar donde está un cuerpo con respecto a otro; de brindar información sobre su perímetro y área; de encontrar el camino mínimo de un punto a otro, así como la generación de modelos a partir de fenómenos o actuaciones simuladas **(Bravo Dominguez, Octubre 2000)**.

Un Sistema de Información Geográfico (SIG) particulariza un conjunto de procedimientos sobre una base de datos no gráfica o descriptiva de objetos del mundo real que tienen una representación gráfica y que son susceptibles de algún tipo de medición respecto a su tamaño y dimensión relativa a la superficie de la tierra **(A. d. Carmona, 2005)**.

En general, un SIG, es una herramienta de análisis de información diseñada para soportar la captura, administración, manipulación, análisis, modelamiento y graficación de datos u objetos. Existen diferentes puntos de vistas de muchos autores sobre la definición de SIG. La mayoría de los autores están de acuerdo en algunos términos, como que los SIG pueden ser aplicados en todas aquellas tareas y proyectos con una componente territorial, porque pueden utilizarse para investigaciones científicas, el marketing, la logística, la gestión de los recursos, entre otros. En definitiva, como una base de integración multidisciplinaria basada en el análisis de elementos.

1.3.2 Conceptos básicos asociados a los Sistema de Información Geográfica.

- **Cartografía:**

La cartografía es el arte de trazar mapas geográficos. Es la ciencia que los estudia y se encarga de la elaboración de estos (Lorenzo Martínez, Diciembre 2004). Como bien dice, es una ciencia, la cual abarca la creación y estudio de mapas territoriales y de diferentes dimensiones lineales y demás. También se denomina cartografía a un conjunto de documentos territoriales referidos a un ámbito concreto de estudio.

- **Mapa:**

Es una representación gráfica y métrica de una porción de territorio generalmente sobre una superficie bidimensional pero que puede ser también esférica como ocurre en los globos terráqueos. Incluye una síntesis de conjunto al igual que un detalle analítico que permita una lectura más profunda. El nivel en que se cumplan estas condiciones será igualmente el nivel de calidad cartográfica (**Florencia, 2009**). Representación geográfica de una parte de la superficie terrestre, en la que se da información relativa a una ciencia determinada.

- **Escala:**

Tamaño de un mapa, plano, diseño, etc., según la escala a que se ajusta. La relación matemática entre las dimensiones en el mapa, carta o plano y la superficie terrestre que representa (**Gopegui, 2009**). Cuanto mayor es la escala, más se aproxima al tamaño real de los elementos de la superficie terrestre. Los mapas a pequeña escala generalmente representan grandes porciones de la Tierra y, por tanto, son menos detallados que los mapas realizados con escalas más grandes.

- **Coordenadas:**

Cantidad usada para definir una posición en un sistema de referencia las coordenadas pueden ser lineales (cartesianas) o angulares (esféricas), según el sistema de referencia (**Felicísimo, 2003**). Se dice de las líneas que sirven para determinar la posición de un punto, y de los ejes o planos a que se refieren aquellas líneas.

- **Leyenda:**

Texto que acompaña a un plano, a un grabado, a un cuadro. Es un listado ordenado y estructurado de las relaciones símbolo/valor para las variables representadas en un mapa, la leyenda debe permitir interpretar los significados de los recursos gráficos usados en el mapa (**Palazón & Quiñonero, 2007**). La leyenda para los mapas es el esquema que viene debajo de éstos donde se indica el significado de los colores, los símbolos, los dibujos, etc.

- **Georreferencia:**

Asignar coordenadas geográficas a un objeto o estructura el concepto aplicado a una imagen digital implica un conjunto de operaciones geométricas que permiten asignar a cada pixel de la imagen un par de coordenadas (x, y) en un sistema de proyección (**Felicísimo, 2003**). En resumen, es asignar coordenadas geográficas a un objeto o estructura.

- **Nodo en un SIG:**

Vértice inicial o final de una línea se aplica por extensión a las entidades puntuales que están interconectadas en una estructura en red. El orden de los nodos (inicial/final) permite asignar a la línea un sentido y dejar definidos los conceptos topológicos de izquierda/derecha (**Felicísimo, 2003**).

- **Red:**

Modelo de datos formado por nodos y conexiones entre ellos tanto los nodos como las conexiones pueden tener atributos propios como, por ejemplo, longitud, resistencia, sinuosidad entre otros. El análisis de redes agrupa un conjunto de técnicas implicadas en la resolución de cuestiones que pueden ser modeladas mediante una red, por ejemplo, determinación del camino de mínimo coste entre dos puntos (**Felicísimo, 2003**).

1.3.3 Funcionalidades de los Sistema de Información Geográfica.

Existen al menos cinco argumentos básicos para la utilización de un SIG, estos motivos son:

- Permite realizar análisis vicariantes, es decir, permite realizar comparaciones entre escalas y perspectivas emulando una cierta capacidad de representación de diferentes lugares al mismo tiempo.
- Permite diferenciar entre cambios cualitativos y cuantitativos; aportando una gran capacidad de cálculo.
- Permite gestionar un gran volumen de información a diferentes escalas y proyecciones.
- Integra espacialmente datos tabulares y geográficos junto a cálculos sobre variables.
- Admite multiplicidad de aplicaciones y desarrollos, poniendo a disposición herramientas informáticas estandarizadas que pueden ir desde simples cajas de herramientas hasta paquetes llave en mano.

1.3.4 Componentes de los Sistemas de Información Geográfica.

- **Hardware.**

El hardware es la computadora en la que opera el SIG. Actualmente, un SIG se ejecuta en un amplio rango de tipos de hardware, desde servidores de computadoras centralizados hasta computadoras de escritorio utilizadas en configuraciones individuales o de red. Una organización requiere de hardware suficientemente específico para cumplir las necesidades de la aplicación. Algunos aspectos a considerar incluyen: velocidad, costo, soporte, administración, escalabilidad y seguridad.

- **Software.**

El software de SIG provee las funciones y herramientas necesarias para almacenar, analizar y mostrar información geográfica. Los componentes clave del software son:

- Un sistema de manejo de base de datos (SMBD).
- Herramientas para el ingreso y manipulación de información geográfica.
- Herramientas de soporte para consultas, análisis y visualización geográficos.
- Una interfaz gráfica del usuario (IGU) para fácil acceso a herramientas.

- **Datos.**

El componente más importante de un SIG son los datos. Recolectar buenos datos de base es un proceso largo, que frecuentemente demora el desarrollo de productos que pueden utilizarse para justificar la inversión.

- **Personal.**

La tecnología de SIG es de valor limitado sin el personal que maneja el sistema. Frecuentemente subestimado, sin gente, los datos se desactualizan y se manejan equivocadamente. El hardware no se utiliza en todo su potencial. Los usuarios de SIG varían desde especialistas técnicos, que diseñan y mantienen el sistema, hasta aquellos que lo utilizan para ayudar a realizar sus tareas diarias.

- **Métodos.**

Un SIG exitoso opera de acuerdo a un plan bien diseñado y reglas de la actividad, que son los modelos y prácticas operativas únicas a cada organización.

1.3.5 Tipos de programas gestores.

Existen dos tipos de programas gestores, que son la base sobre la que se inspiran los diferentes programas. También se pueden tomar como dos formas de tratar la información, estas son:

- **Ráster.**

Un tipo de datos ráster es, en esencia, cualquier tipo de imagen digital representada en mallas. El modelo de SIG ráster o de retícula se centra en las propiedades del espacio más que en la precisión de la localización. Divide el espacio en celdas regulares donde cada una de ellas representa un único valor. Los datos ráster se compone de filas y columnas de celdas, cada celda almacena un valor único. Los datos ráster pueden ser imágenes (imágenes ráster), con un valor de color en cada celda (o píxel) (**Bosque, 1997**). Se puede decir entonces, que es una organización de base de datos donde se almacenan los datos espaciales como celdas de una matriz bidimensional o malla, que se ve como una imagen. El ejemplo más común es la información de satélites, fotos aéreas escaneadas y también cualquier otro tipo de imagen usada en diseño gráfico (tif, bmp, gif). Cada celda en un set de datos ráster tiene información descriptiva dada por el valor almacenado en cada celda.

- **Vectorial.**

En un SIG, las características geográficas se expresan con frecuencia como vectores, manteniendo las características geométricas de las figuras. En los datos vectoriales, el interés de las representaciones se centra en la precisión de localización de los elementos geográficos sobre el espacio y donde los fenómenos a representar son discretos, es decir, de límites definidos. Los lagos pueden tener un rango de colores en función del nivel de contaminación. Además, las diferentes geometrías de los elementos

también pueden ser comparados. Los datos vectoriales se pueden utilizar para representar variaciones continuas de fenómenos. Las líneas de contorno y las redes irregulares de triángulos (TIN) se utilizan para representar la altitud u otros valores en continua evolución. Los TIN son registros de valores en un punto localizado, que están conectados por líneas para formar una malla irregular de triángulos. La cara de los triángulos representa, por ejemplo, la superficie del terreno (**Bosque, 1997**).

Para modelar digitalmente las entidades del mundo real se utilizan tres elementos geométricos: el punto, la línea y el polígono.

- Puntos: Se utilizan para las entidades geográficas para una simple ubicación, también se pueden utilizar para representar zonas a una escala pequeña.
- Líneas o polilíneas: Son usadas para rasgos lineales como ríos, caminos, ferrocarriles, rastros, líneas topográficas o curvas de nivel.
- Polígonos: Se utilizan para representar elementos geográficos que cubren un área particular de la superficie de la tierra. Estas entidades pueden representar lagos, edificios, provincias, entre otros.

1.3.6 Técnicas utilizadas en los Sistemas de Información Geográfica

- **La creación de datos.**

La teledetección es una de las principales fuentes de datos para los SIG. Las modernas tecnologías SIG trabajan con información digital, para la cual existen varios métodos utilizados en la creación de datos digitales (**Bosque, 1997**). Entre los distintos métodos de creación de mapas para SIG, se destaca la digitalización, ya que es el método más utilizado. Este consiste en que a partir de un mapa impreso o con información tomada en campo, se transfiere a un medio digital.

- **La representación de los datos.**

Los datos SIG representan los objetos del mundo real (carreteras, el uso del suelo, altitudes). Los objetos del mundo real se pueden dividir en dos abstracciones: objetos discretos (una casa) y continuos (cantidad de lluvia caída, una elevación) (**Bosque, 1997**). Existen dos formas de almacenar los datos en un SIG, por ráster y vectorial, las cuales ya fueron expuestas en el punto 1.3.5 Tipos de programas gestores.

- **Datos no espaciales.**

Los datos no espaciales también pueden ser almacenados junto con los datos espaciales, aquellos representados por las coordenadas de la geometría de un vector o por la posición de una celda ráster (**Bosque, 1997**).

- **La captura de datos.**

La captura de datos y la introducción de información en el sistema consumen la mayor parte del tiempo de los profesionales de los SIG. Hay una amplia variedad de métodos utilizados para introducir datos en un SIG almacenados en un formato digital. Los datos impresos en papel o mapas en película pueden ser digitalizados o escaneados para producir datos digitales. Los datos obtenidos de mediciones topográficas pueden ser introducidos directamente en un SIG a través de instrumentos de captura de datos digitales mediante una técnica llamada geometría analítica (**Bosque, 1997**). Además, las coordenadas de posición tomadas a través de un Sistema de Posicionamiento Global (GPS) también pueden ser introducidas directamente en un SIG. Los sensores remotos también juegan un papel importante en la recolección de datos.

- **Cartografía automatizada.**

Tanto la cartografía digital como los SIG codifican relaciones espaciales en representaciones formales estructuradas. Los SIG son usados en la creación de cartografía digital como herramientas que permiten realizar un proceso automatizado o semiautomatizado de elaboración de mapas denominado cartografía automatizada (**Bosque, 1997**).

- **Geoestadística.**

La geoestadística analiza patrones espaciales con el fin de conseguir predicciones a partir de datos espaciales concretos. Se emplea el uso de la teoría de grafos y de matrices algebraicas para reducir el número de parámetros en los datos (**Bosque, 1997**). Esta es una forma de ver las propiedades estadísticas de los datos espaciales.

- **Geocodificación.**

Geocodificación es el proceso de asignar coordenadas geográficas (latitud-longitud) a puntos del mapa (direcciones, puntos de interés, entre otros) (**Bosque 1997**). La geocodificación puede realizarse también con datos reales más precisos.

1.4 Análisis de las soluciones existentes.

Actualmente en el mundo se ha hecho imprescindible el uso de los SIG. En casi todos los sectores de la sociedad se trabaja con SIG y se hace mucho más que solo graficar datos. En el mundo se han creado empresas especializadas en el trabajo con datos espaciales, por ejemplo:

- Geosystems, que es una empresa de IT (Tecnología de la Información) especializada en Sistemas de Información Geográfica, procesamiento de imágenes satelitales, colectores GPS y Estaciones Totales. Geosystems Ingeniería es una división de Geosystems, que se especializa también en la realización de proyectos de ingeniería civil, con énfasis en diseño, estudios topográficos, selección de ruta y determinación de características geotécnicas de suelos mediante prospección geofísica. En fin, toda una empresa de creación de SIG. Pero enmarcando sus características a la investigación que se realiza, se puede decir que hasta el momento no utilizan el framework Geomoose para el desarrollo de los SIG sino que emplean el framework Symphony.
- También está la familia de los productos ArcGIS que permiten crear arquitecturas de software que puedan soportar la implementación de Sistemas de Información Geográfica completos. Estos productos están clasificados en diferentes grupos según sus funcionalidades:
 - Productos de Escritorio (ArcGIS Desktop).
 - Productos Servidores (ArcGIS Server).
 - Productos Móviles (ArcGIS Mobile).

Pero de igual manera, esta familia para la creación de SIG no utiliza Geomoose. El framework que utiliza es Zend Framework, ya que este emplea código orientado a objetos.

- Otro de los ejemplos que se pueden exponer, es Kosmo. Este es un SIG que por sus características es el más cercano a la solución que se propone para resolver el problema planteado en la investigación. Kosmo está concebido para servir como plataforma en la implementación de soluciones SIG similares. Aunque en parte, luego de un análisis, es descartado, ya que este se encuentra

desarrollado sobre la antes mencionada familia de productos ArcGIS. Por tal motivo, el SIG Kosmo, tampoco utiliza el framework Geomoose, ya que también utiliza Zend Framework.

Geomoose es una tecnología que aún no está siendo utilizada por las grandes empresas para la creación de los SIG. Esto ocurre porque la mayoría de estas empresas, ya tiene experiencia con el uso de otros framework con los que han garantizado una buena comercialización de sus sistemas. Por lo que no han empleado a Geomoose para la implementación de estos. La percepción de que Geomoose puede garantizar el éxito al ser utilizado en el desarrollo de un SIG, se basa en el criterio de muchos desarrolladores alrededor del mundo. Estos implementadores han logrado soluciones de este tipo sobre el framework Geomoose siempre resaltando que es una herramienta fácil de usar. Aunque es válido aclarar que estos SIG que se han desarrollado y están implementándose, no son soluciones genéricas que puedan ser utilizados para el apoyo a otros desarrolladores interesados en el tema.

Cuba se suma también a la evolución de las TIC e implementa diferentes soluciones SIG para resolver los problemas de representación de datos espaciales. Con motivo de ganar experiencia en el tema, se preparan eventos de intercambio donde se invitan a los jóvenes de la Sociedad Cubana de Geografía (SCG) y a especialistas de instituciones como GEOCUBA, la Facultad de Geografía (³UH), la Universidad de Ciencias Informáticas, entre otras.

- Uno de los SIG existentes en Cuba es para representar el cultivo del tabaco en Pinar del Río. Este SIG fue creado para de esta forma, influir positivamente en la toma de decisiones sobre las estrategias de siembras del cultivo en el territorio. También para poder aminorar de alguna forma el ataque de plagas y enfermedades que tanto afectan al proceso productivo. Por lo que se puede concluir que este no es un SIG genérico, y no soluciona el problema planteado.
- El Sistema de Información Geográfica para la gestión de la estadística de salud de Cuba, previsto para llevar el control de higiene y Epidemiología, es otro ejemplo de SIG en Cuba. Este SIG al igual que el SIG para asistir la estrategia de conservación de las áreas protegidas de la provincia de Camagüey, es un ejemplo claro del auge que también toman los SIG en Cuba.

³ UH: Universidad de la Habana.

- Muchos de los SIG creados en Cuba, han nacido en la Universidad de las Ciencias Informáticas, como por ejemplo del SIGUCI. El SIGUCI es un producto creado específicamente para la representación de los datos espaciales de la UCI. Este tiene dos puntos importantes por los que se puede caracterizar. Primeramente, es un SIG personalizado, por lo que no se acerca mucho a la solución que se desea. En segundo lugar, por ser un SIG creado en la UCI, se utilizó para su desarrollo la plataforma GeneSIG, por lo que está implementado sobre el framework CartoWeb.

Después de realizado el anterior análisis, se puede determinar, que la solución que se plantea en la investigación, puede cumplir con la necesidad de resolver el problema planteado. Ya que no se cuenta con una herramienta que cumpla con las características necesarias que permita el desarrollo de los SIG de forma rápida y eficaz utilizando el framework Geomoose.

1.5 Conclusiones parciales.

Este primer capítulo, ha servido para obtener un mayor conocimiento de los principales conceptos asociados al proceso de desarrollo de los SIG. Al tener una mejor comprensión de estos elementos, se tiene una visión más clara de lo que hay mostrar en la investigación.

Además se detalló un estudio de las posibles soluciones existentes relacionadas con la solución que se plantea en la investigación. Se determinó que aunque, el análisis de estas soluciones existentes, arrojó como resultado, que muchas de las empresas en el mundo, que utilizan los SIG, no utilizan el framework Geomoose, el uso de estos, brinda una mayor organización en la gestión de sus datos espaciales. Por tanto después de realizado el estado del arte, se determinó, que en el mundo y en Cuba no existen soluciones que respondan a las necesidades que se enfrentan hoy día en el departamento de Geoinformática. Las soluciones de este tipo, se encuentran en proceso y están siendo implementadas por programadores de manera individual y no en empresas. A pesar de que estas aplicaciones existentes no son como las aplicaciones requeridas, el conocimiento adquirido durante el estudio de estas, se utiliza en gran medida para un mayor entendimiento de las características generales a tener en cuenta para la implementación de los SIG.

Capítulo 2: Posibles tecnologías a utilizar.

2.1 Introducción.

Después de realizado en el capítulo 1, un esbozo completo de los elementos fundamentales para el desarrollo de un SIG, se puede realizar un estudio de las posibles herramientas y tecnologías actuales a utilizar para la implementación de este. En el actual capítulo se realiza un análisis de los posibles lenguajes, herramientas y metodologías más conocidas, que son utilizadas para el desarrollo de este tipo de aplicaciones. Se expone el resultado de dicho estudio, haciendo una selección de las tecnologías que son más convenientes en el caso de esta investigación. Además en este capítulo se argumentan los motivos por los que esas herramientas son seleccionadas. Se tuvo en cuenta a la hora de realizar la selección, principalmente dos aspectos. En primer lugar que estas fueran las que mejor se ajustaran al proceso de desarrollo de un SIG sobre el framework Geomoose. En segundo lugar, que fueran herramientas no privativas, teniendo en cuenta el proceso de migración hacia software libre en el que se encuentran inmersas las empresas en Cuba.

2.2 Metodologías de desarrollo de software.

A lo largo del tiempo han surgido metodologías usadas para estructurar, planear y controlar el proceso de desarrollo en sistemas de información. A estas metodologías se les denomina metodologías de desarrollo de software, las cuales definen Quién hace Qué, Cuándo y Cómo lo hace. Una gran cantidad de métodos han sido desarrollados diferenciándose por su fortaleza y debilidad.

➤ **Metodologías robustas.**

• **Proceso Unificado de Desarrollo (RUP).**

La metodología RUP, llamada así por sus siglas en inglés Rational Unified Process, divide en 4 fases el desarrollo del software:

- Inicio: El objetivo en esta etapa es determinar la visión del proyecto.
- Elaboración: En esta etapa el objetivo es determinar la arquitectura óptima.
- Construcción: En esta etapa el objetivo es llevar a obtener la capacidad operacional inicial.
- Transición: El objetivo es llegar a obtener el release del proyecto.

Cada una de estas etapas es desarrollada mediante el ciclo de iteraciones, la cual consiste en reproducir el ciclo de vida en cascada a menor escala. Los objetivos de una iteración se establecen en función de la evaluación de las iteraciones precedentes **(Sánchez, Junio 2004)**.

RUP es una metodología orientada a objetos basada en el Lenguaje Unificado de Modelado (UML). En esta se han agrupado las actividades en 9 flujos de trabajo, siendo los 6 primeros flujos de ingeniería y los 3 últimos de apoyo. Sus principales elementos son los trabajadores (definen *quién*), las actividades (definen el *cómo*), los artefactos (definen *qué*) y el flujo de actividades (definen *cuándo*).

➤ **Metodologías ágiles.**

• **Programación Extrema (XP).**

Es una de las metodologías de desarrollo de software más exitosas utilizadas en la actualidad para proyectos de corto plazo. La metodología consiste en una programación rápida o extrema, cuya particularidad es tener como parte del equipo, al usuario final, pues es uno de los requisitos para llegar al éxito del proyecto **(Sánchez, Junio 2004)**.

XP es una metodología de desarrollo de software que se caracteriza por ser flexible, eficiente y poseer bajo riesgo. Se centra en potenciar las relaciones entre cliente y equipo de desarrollo. Así como potenciar al máximo el trabajo en equipo.

✓ **Selección de la metodología de desarrollo de software a utilizar.**

Proceso Unificado de Desarrollo (RUP).

Como Metodologías de desarrollo de software a utilizar se selecciona RUP, por la característica que este tiene. Permite soportar el desarrollo de grandes proyectos, además de la generación de toda la documentación asociada al mismo en cada una de sus fases. Como RUP es iterativo, da la posibilidad de organizar los proyectos por fases. Estas fases van a constar de una o más iteraciones, lo que ayuda a mitigar los riesgos tempranamente. Es por ello que esta metodología garantiza que se logre desde las primeras fases de creación un producto de calidad.

2.3 Lenguaje de Modelado.

➤ **Lenguaje Unificado de Modelado (UML).**

UML (Unified Modeling Language) es un lenguaje que permite modelar, construir y documentar los elementos que forman un sistema software orientado a objetos. Se ha convertido en el estándar de la

industria, ya que crea una notación unificada en la que basa la construcción de sus herramientas CASE. En el proceso de creación de UML han participado, empresas de gran peso en la industria como Microsoft, Hewlett-Packard, Oracle o IBM, así como grupos de analistas y desarrolladores (G., J., & I., 1999). Para realizar un proyecto se debe realizar un esquema del mismo. El lenguaje UML permite mediante diagramas, plasmar de una forma detallada la solución al problema planteado. Además es necesario organizar el proceso de diseño de tal forma que los analistas, clientes, desarrolladores y otras personas involucradas en el desarrollo del sistema lo comprendan y convengan con él. El UML proporciona esta organización a través de la creación de diferentes diagramas (clases, objetos, casos de uso, secuencia, colaboración, estados, actividades, componentes, despliegue).

✓ Selección del Lenguaje modelado (UML) a utilizar.

Lenguaje Unificado de Modelado (UML) en su versión 2.1.

Si se va a emplear RUP como metodología de desarrollo de software, el lenguaje de modelado más eficiente y conveniente para utilizar es UML. Esta combinación de RUP con UML es la más utilizada para realizar el análisis, implementación y documentación de los sistemas orientados a objetos.

2.4 Herramientas de Ingeniería de Software Asistidas por Computadora.

Las herramientas CASE⁴ ayudan al ingeniero de software en la producción de resultados de alta calidad gracias a que facilitan la realización de prototipos y el desarrollo conjunto de aplicaciones, simplifican el mantenimiento del software, mejoran y estandarizan la documentación, aumentan la portabilidad de las aplicaciones u ofrecen la reutilización de componentes de software (Ramos, 2006). Las herramientas CASE permiten a los programadores trabajar en un alto nivel de abstracción en la definición de un sistema de software que entonces será construido. Este puede ser generalmente aplicado a cualquier sistema o colección de herramientas que ayuda a automatizar el proceso de diseño y desarrollo de software.

- **Rational Rose Enterprise Edition.**

Es la herramienta líder en el mundo de modelación visual para el proceso de modelación del negocio, análisis de requerimientos y diseño de arquitectura de componentes. Rational Rose es una herramienta de

⁴ CASE: Ingeniería de Software Asistida por Computadora. Con su acrónimo en inglés, Computer Aided Software Engineering.

desarrollo basada en modelos que se integra con las bases de datos y los IDE⁵ de las principales plataformas del sector. Todos los productos de Rational Rose dan soporte a UML, pero no son compatibles con las mismas tecnologías de implementación. Además tiene como inconveniente que necesita un buen equipo para que resulte ser eficiente.

- **Visual Paradigm para UML.**

Visual Paradigm es una herramienta CASE que utiliza UML como lenguaje de modelado. Soporta el ciclo de vida completo del desarrollo de software: Análisis y Diseño, Construcción, Pruebas y Despliegue. En Visual Paradigm for UML, el diseño está centrado en casos de uso. Es una herramienta diseñada para una amplia gama de usuarios. En adición al soporte de Modelado UML esta herramienta provee un generador de mapeo de objetos relacionales para los lenguajes de programación como PHP. Además se integra con NetBeans IDE entre otras herramientas. Tiene también conexión con Rational Rose en sus archivos de proyecto, por lo que los archivos pueden ser importados a Visual Paradigm UML (**Inc, 2005**). Visual Paradigm For UML Enterprise Edition, es una herramienta muy completa y fácil de emplear. Permite que varios desarrolladores trabajen en el mismo diagrama y vean los cambios que se hacen en tiempo real. Además ayuda a los desarrolladores a mejorar la construcción del modelo del software, lo que posibilita acelerar la producción y mejorar la calidad del trabajo.

- ✓ **Selección de la Herramienta de Ingeniería de Software Asistida por Computadora a utilizar.**

Visual Paradigm para UML en su versión 6.4.

La decisión de utilizar el Visual Paradigm for UML Enterprise Edition 6.4, está dada a que esta herramienta CASE se considera la herramienta de modelado más adecuada para trabajar en software libre. Fundamentalmente, se tiene en cuenta el proceso de migración hacia software libre en el que se encuentran las empresas que pueden llegar a solicitar productos SIG sobre la aplicación que se implementará durante esta investigación. Al ser seleccionado el lenguaje de modelado UML, es conveniente tener en cuenta su vinculación con Visual Paradigm ya que este soporta las últimas versiones del UML. Teniendo en cuenta este aspecto, es importante resaltar que Visual Paradigm tiene

⁵ IDE: Entorno de Desarrollo Integrado. Su acrónimo en inglés es Integrated Development Environment.

abundantes tutoriales de UML y demostraciones interactivas. Permite dibujar todos los tipos de diagramas fácilmente.

2.5 Lenguajes de programación.

Es un conjunto de símbolos junto a un conjunto de reglas para combinar dichos símbolos que se usan para expresar programas. Constan de un léxico⁶, una sintaxis⁷ y una semántica⁸. Atendiendo al número de instrucciones necesarias para realizar una tarea específica, se pueden clasificar los lenguajes informáticos en dos grandes bloques: bajo nivel y alto nivel (**Lobos, 2009**). En otras palabras un lenguaje de programación es cualquier lenguaje artificial, el cual, se utiliza para definir adecuadamente una secuencia de instrucciones que puedan ser interpretadas y ejecutadas en una computadora. Existen lenguajes de programación con los que se pueden implementar aplicaciones de escritorio y aplicaciones web. Si se desea desarrollar una aplicación web, como es el caso de esta investigación, se deben tener en cuenta los lenguajes del lado del servidor y los del lado del cliente.

➤ Lenguajes del lado del servidor.

Los lenguajes del lado del servidor son aquellos lenguajes que son reconocidos, ejecutados e interpretados por el propio servidor y que se envían al cliente en un formato comprensible para él (**Torre, 2006**). El navegador es una especie de aplicación capaz de interpretar las órdenes recibidas en forma de código HTML fundamentalmente y convertirlas en las páginas que son el resultado de dicha orden. Cuando se pulsa sobre un enlace hipertexto, en realidad lo que pasa es que se establece una petición de un archivo HTML residente en el servidor (un ordenador que se encuentra continuamente conectado a la red) el cual es enviado e interpretado por el navegador (el cliente). Existen varios lenguajes de programación del lado del servidor, tales como CGI (Common Gateway Interface, Pasarela de Interfaz Común), Perl, ASP.net, Java Server Pages (JSP), PHP y Ruby. En esta investigación se brindan características de algunos de estos:

⁶ Léxico: Conjunto de símbolos permitidos o vocabulario.

⁷ Sintaxis: Reglas que indican cómo realizar las construcciones del lenguaje.

⁸ Semántica: Reglas que permiten determinar el significado de cualquier construcción del lenguaje.

- **JSP.**

Con JSP se pueden crear aplicaciones web que se ejecuten en variados servidores web, de múltiples plataformas, ya que Java es en esencia un lenguaje multiplataforma. Por tanto, las JSP se pueden escribir con el editor HTML/XML habitual **(Torre, 2006)**.

Es decir, JSP es una tecnología orientada a crear páginas web con programación en Java.

- **Perl.**

Perl es un lenguaje de programación interpretado, al igual que muchos otros lenguajes de Internet como JavaScript o ASP. Esto quiere decir que el código de los scripts en Perl no se compila sino que cada vez que se quiere ejecutar se lee el código y se pone en marcha interpretando lo que hay escrito **(Torre, 2006)**.

- **PHP.**

PHP es el acrónimo de Hypertext Preprocesor. Es un lenguaje de programación del lado del servidor gratuito e independiente de plataforma, rápido, con una gran librería de funciones y mucha documentación. Se utiliza para la creación de páginas web dinámicas. Permite la conexión con la mayoría de los sistemas gestores de bases de dato, como son: MySQL, PostgreSQL, Oracle, MS SQL Server, entre otras. De la versión PHP5, la más reciente, la 5.3.1 presenta un mejor soporte para la Programación Orientada a Objetos, además de que posee mejoras de rendimiento y manejo de excepciones.

Este lenguaje de programación, tiene muchas ventajas que lo hacen resaltar. Algunas de estas ventajas son:

- Bajo coste.
- Facilidad de aprendizaje y uso.
- Portabilidad.
- Acceso a código abierto. **(Torre, 2006)**.

✓ Selección del lenguaje del lado del servidor a utilizar.

PHP5.

Dentro de los lenguajes de programación, el lenguaje de programación del lado del servidor seleccionado es PHP5. Después de estudiado este lenguaje, se considera que posee gran flexibilidad y rendimiento para la realización de páginas web dinámicas. Además de que cumple con ser gratuito y cuenta con abundante documentación, lo cual permite un mayor entendimiento del mismo.

➤ Lenguajes del lado del cliente.

Son aquellos que pueden ser directamente interpretados por el navegador. Así, por ejemplo, un lenguaje del lado del cliente es totalmente independiente del servidor. Esto permite que la página pueda ser albergada en cualquier sitio sin necesidad de pagar más ya que, por regla general, los servidores que aceptan páginas con scripts de lado del servidor, tienen en su mayoría prestaciones muy limitadas (**Torre, 2006**).

Resumiendo este aspecto, se puede decir, que los lenguajes del lado del cliente son independientes del servidor y esto posibilita que las páginas puedan guardarse en cualquier sitio. Algunos de los lenguajes de programación del lado del cliente son: JavaScript, HTML, CSS, entre otros.

• JavaScript.

JavaScript es un lenguaje del lado del cliente que no requiere de compilación, es decir, es interpretado. Fue diseñado con una sintaxis similar a la de Java. La principal diferencia radica en que Java es un lenguaje orientado a objetos y JavaScript está basado en prototipos. Es un lenguaje de programación utilizado para crear pequeños programitas encargados de realizar acciones dentro del ámbito de una página web. Su uso se basa fundamentalmente en la creación de efectos especiales en las páginas y la definición de interactividades con el usuario (**Torre, 2006**).

Es compatible con la mayoría de los navegadores como Netscape, Internet Explorer, Mozilla Firefox, entre otros. Para que interactúe con una página web evitando incompatibilidades, se diseñó el estándar DOM (Modelo de Objetos del Documento). Por último se debe señalar que este lenguaje es seguro y fiable.

- **HTML.**

HTML es un lenguaje de marcas hipertextuales, un lenguaje diseñado para estructurar textos para generar páginas web. La interpretación de las etiquetas es realizada por el navegador web. El lenguaje HTML es extensible, se le pueden añadir características, etiquetas y funciones adicionales para el diseño de páginas web, generando un producto vistoso, rápido y sencillo (**Torre, 2006**). Gracias a los navegadores web, el html se ha convertido en el formato más fácil para la creación de páginas web debido a su sencillez.

- ✓ **Selección del lenguaje del lado del cliente a utilizar.**

JavaScript.

Dentro de los lenguajes de programación, el lenguaje de programación del lado del cliente seleccionado es JavaScript. Uno de los puntos estudiados para su selección, es la característica que tiene de ser compatible con la mayoría de los navegadores. Este se utiliza del lado del cliente, implementado como parte de un navegador web permitiendo mejoras en la interfaz de usuario y páginas web dinámicas. Además es un lenguaje sencillo, del cual se puede obtener abundante documentación.

2.6 Framework.

Es un esquema, un esqueleto, un patrón para el desarrollo y/o la implementación de una aplicación. En términos generales, se puede decir que un framework es un conjunto de clases, que ofrece una guía arquitectónica y forman un diseño reutilizable para un tipo específico de software. (**Sys, 2006**).

Teniendo en cuenta lo afirmado en el concepto anterior, se puede concluir que los frameworks ofrecen estructuras básicas que se utilizan para construir aplicaciones web y aceleran todo el proceso de creación, pues permiten la reutilización de código y reducen también la cantidad de código repetitivo para los desarrolladores.

2.6.1 Framework Geomoose.

El framework Geomoose fue desarrollado en la Universidad de Minnesota y está basado totalmente en MapServer. Además, trabaja sobre OpenLayers, por lo que la interfaz de Geomoose llega a tener aspectos muy parecidos a este. Geomoose es un framework para uso del lado del cliente basado en JavaScript para visualizar información cartográfica distribuida. Es muy fácil de implementar y trae consigo muchas herramientas las cuales funcionan como servicios. Esto significa que cada vez que algún servicio

es usado, este se ejecuta de forma independiente en el servidor, el cual por medio de archivos .map de MapServer obtiene la información requerida o actualiza una capa de información geográfica según el tipo de solicitud hecha por el usuario. Es un framework que tiene una arquitectura modular y es configurable. Algunos de los aspectos más importantes referentes al framework Geomoose son:

- Licencia: MIT-style v.2+; City of Saint Paul Open Source License v.1.x
- País de origen: Estados Unidos.
- Documentación: En idioma inglés y en formato de HTML.
- Lenguaje en el que está escrito: JavaScript; PHP.
- Lenguaje de programación que admite su API: PHP.
- Dependencia de servidor de mapas: No.
- Requerimiento de plugins privativos: No.

✓ Selección del framework a utilizar.

Framework Geomoose en su versión 2.2.

El framework a utilizar fue seleccionado a comienzos de esta investigación. Su selección estuvo regida por la necesidad de diversificación de las tecnologías utilizadas en el departamento de Geoinformática para la creación de SIG. El framework con el que se trabaja es Geomoose.

2.7 Sistemas de Gestión de Bases de Datos (SGBD).

Pueden definirse como un paquete generalizado de software, que se ejecuta en un sistema computacional anfitrión, centralizando los accesos a los datos y actuando de interfaz entre los datos físicos y el usuario. Las principales funciones que debe cumplir un SGBD se relacionan con la creación y mantenimiento de la base de datos, el control de accesos, la manipulación de datos de acuerdo con las necesidades del usuario y mantener la integridad (**Tramullas y Kronos, 2000**). Son un tipo de software muy específico, dedicado a servir de interfaz entre la base de datos, el usuario y las aplicaciones que la utilizan. El propósito general de los SGBD es el de manejar de manera clara, sencilla y ordenada un conjunto de datos que posteriormente se convertirán en información relevante para una organización.

- **MySQL.**

MySQL surge con el objetivo de que cumpla el estándar SQL, Structure Query Language (Lenguaje de Consulta Estructurado). SQL es un lenguaje de programación para trabajar con base de datos relacionales. MySQL es un interpretador de SQL, es un servidor de base de datos, que permite crear base de datos y tablas, insertar datos, modificarlos, eliminarlos, ordenarlos, hacer consultas y realizar muchas operaciones (**Tramullas y Kronos, 2000**). A modo de resumen se puede plantear que MySQL administra bases de datos. Es multihilo por lo que puede realizar varias tareas a la vez y es también multiusuario, lo que le permite satisfacer de forma simultánea, las necesidades de varios usuarios que comparten los mismos recursos. Es un software de código abierto por lo que cualquier persona puede usarlo y modificarlo.

- **PostgreSQL.**

Es un proyecto con más de 15 años de vida. La licencia del PostgreSQL tiene menos restricciones en comparación con otras licencias de otros SGBD. Está orientado a objetos, es libre y multiplataforma. Incluye herencia entre tablas e incorpora una estructura de datos array. PostgreSQL es un gestor de base de datos objeto-relacional. Esta estrategia permite obtener una mejor respuesta en ambientes de grandes volúmenes. Tiene una serie de bibliotecas escritas en JavaScript para la construcción de aplicaciones interactivas. Brinda también, soporte nativo para los lenguajes más populares del medio: PHP, C, C++, Perl, Python, Java/JDBC, Ruby, entre otros. (**Montesinos & Sanz, 2003**).

Además se puede afirmar que el sistema insignia en el ámbito de las bases de datos del software libre es PostGIS, el módulo para PostgreSQL. Este módulo proporciona a PostgreSQL la capacidad no solo de almacenar información geoespacial, sino de realizar operaciones de análisis geográfico.

- **Oracle**

Oracle es un sistema de gestión de base de datos relacional (RDBMS por el acrónimo en inglés de Relational Data Base Management System). Se considera como uno de los sistemas de bases de datos más completos. La tecnología Oracle se encuentra prácticamente en todas las industrias alrededor del mundo. A continuación se señala información general sobre este SGBD:

- Género: RDBMS.
- Sistema Operativo: Multiplataforma.
- Licencia: Privativa.

✓ Selección del Sistema de Gestión de Bases de Datos (SGBD).

PostgreSQL en su versión 8.4.

El SGBD seleccionado fue PostgreSQL, el cual es más maduro que MySQL en cuanto al manejo de datos espaciales. Debido a cuestiones legales y sobre todo económicas, no se consideró el uso de Oracle, ya que su licencia es privativa. Además fue importante para esta selección, la valoración de la existencia de la extensión PostGIS que es la que permite el trabajo con datos espaciales en los SIG.

2.8 Entorno de Desarrollo Integrado.

Los IDEs son un conjunto de herramientas para el programador, que suelen incluir en una misma suite, un buen editor de código, administrador de proyectos y archivos, enlace transparente a compiladores e integración con sistemas controladores de versiones o repositorios (**Luciano, 2005**).

Los IDEs son editores de código que pueden servir además, para depurar y facilitar las diferentes tareas necesarias en el desarrollo de cualquier tipo de aplicación. Esto se debe a que incorporan un conjunto de herramientas como editores contextuales, visuales, paletas de componentes de interfaz de usuario, consola de salidas, entre otros, que facilitan el desarrollo de aplicaciones. Existen diferentes IDEs con los que se puede trabajar, algunos de estos se mencionan a continuación:

- **Zend Studio**

Utilizado para desarrollar aplicaciones Web con PHP, es posiblemente uno de los mejores IDE del momento para PHP. Es un editor de texto para páginas en código PHP que proporciona ayudas a medida que se va desarrollando el proyecto, desde la creación y gestión de proyectos hasta la depuración del código. La integración de Zend Studio con Zend Framework está también en el contexto de las mejores prácticas de este IDE.

- **Netbeans**

Es multilenguaje y modular con soporte para Java. Tiene una gran comunidad de usuarios y desarrolladores. Crea ventanas, menús y barras de herramientas fácilmente. Es una herramienta para que los programadores puedan escribir, compilar, depurar y ejecutar programas (**Cerda, 2007**). Está escrito en Java pero puede servir para cualquier otro lenguaje de programación. Con este IDE se pueden crear

aplicaciones de escritorio, aplicaciones web, entre otras. Netbeans IDE es un producto libre y gratuito sin restricciones de uso, con instalación y actualización simple.

✓ Selección del Entorno de Desarrollo Integrado a utilizar.

Netbeans en su versión 6.9.

Se selecciona Netbeans, pues brinda la posibilidad de desarrollar aplicaciones web utilizando PHP 5. Permite llamar a sus comandos y construcciones, posibilitando que sea más confortable su uso. También se pueden agregar paquetes de clases y usar sus funcionalidades. Tiene completamiento y corrección para PHP, JavaScript, entre otros.

2.9 Servidores de mapas.

Los servidores de mapas permiten al usuario la máxima interacción con la información geográfica. Por un lado el usuario o cliente accede a información en su formato original, de manera que es posible realizar consultas tan complejas como las que haría un SIG. El servidor de mapas es personalizable, es decir, se pueden preparar o programar las herramientas (los iconos de la aplicación) de manera que sean intuitivas para el usuario no experto en SIG (Rozo, 2002).

- **MapServer.**

Es un entorno de desarrollo en código abierto (Open Source Initiative) para la creación de aplicaciones SIG en Internet con el fin de visualizar, consultar y analizar información geográfica a través de la red mediante la tecnología Internet Map Server (IMS). Su funcionamiento básico está configurado en un fichero de texto, que generalmente tiene la extensión .map (Fawcett & Butler, 2010).

Presenta las siguientes características:

- Adaptable.
- Corre bajo plataformas Linux y Windows.
- Formatos ráster soportados: TIFF/GeoTIFF, GIF, PNG, ERDAS, JPEG y EPPL7.
- Formatos vectoriales soportados.
- Permite dibujar sobrecargas en datos tanto ráster como vectoriales.

Además está compuesto por un fichero de configuración el cual permite guardar datos espaciales que pueden ser cargados en el servidor como archivos de mapa. Este fichero es el MapFile.

- **GeoServer.**

GeoServer es un servidor de software de código abierto escrito en Java que permite a los usuarios compartir y editar los datos geoespaciales. GeoServer es desarrollado, probado y apoyado por un grupo diverso de individuos y organizaciones de todo el mundo. GeoServer forma un componente básico de la Web Geoespacial.

- ✓ **Selección del servidor de mapas a utilizar.**

MapServer en su versión 5.6.

Como servidor de mapas se seleccionó a MapServer, por las características que presenta en cuanto a la sencillez de configuración y administración. Además, el framework Geomoose obtiene la información requerida o actualiza una capa de información geográfica, según el tipo de solicitud hecha por el usuario del SIG Web, por medio de archivos .map de MapServer.

2.10 Servidores Web.

Un servidor web es un programa que sirve para atender y responder a las diferentes peticiones de los navegadores, proporcionando los recursos que soliciten usando el protocolo HTTP o el protocolo HTTPS.

- **Apache.**

Apache es el servidor Web más utilizado en todo el mundo. Permite que se le incorporen extensiones, debido a que sus capacidades pueden ser ampliadas incluyéndole nuevos módulos, por ejemplo PHP. Apache es compatible con varios Sistemas Operativos, además tienen a su favor que es una tecnología libre, ya que su licencia no es privativa (**Cooper Bowen & Coar, 2004**).

- **IIS.**

Servicios de Internet Information Server (IIS) es un servidor web, creado por Microsoft para su uso con Microsoft Windows. Es el servidor web más utilizado después del Servidor Apache. Es una parte integral de la familia de productos Windows Server, así como todas las ediciones de Windows Vista y Windows 7. Aunque algunas funciones no son compatibles con las versiones cliente de Windows.

✓ Selección del servidor Web a utilizar.

Apache en su versión 2.

Como servidor web, se seleccionó Apache, principalmente por la característica que tiene de ser una tecnología libre y de ser compatible con el lenguaje de desarrollo utilizado en esta investigación, PHP. Además, actualmente es el servidor web más utilizado en todo el mundo, debido a que es el más adecuado para PHP.

2.11 Conclusiones parciales.

Durante la confección del presente capítulo, se ha realizado un estudio y caracterización de las herramientas que pueden ser utilizadas para desarrollar un SIG. Se ha determinado que existen muchas herramientas que se pueden aplicar para la implementación del mismo. Además se ha establecido, que de la correcta selección de las herramientas a utilizar, dependen las decisiones a tomar en el proyecto, logrando soluciones que cubran las necesidades iniciales del mismo. Esto posibilita la continuidad de la investigación, permitiendo el modelamiento e implementación de la aplicación mediante el uso de las herramientas previamente seleccionadas. Se ha tenido en cuenta para la selección de las herramientas, las que son más factibles para el trabajo con Geomoose. Además se tuvo en cuenta, que dentro de lo posible, fueran las más utilizadas comúnmente en el proyecto, para garantizar un mayor éxito en el desarrollo de la aplicación.

Capítulo 3: Análisis y diseño.

3.1 Introducción.

En el capítulo anterior, se realizó un análisis sobre las herramientas y tecnologías existentes para la resolución del problema planteado. Parte de este estudio, permitió realizar una correcta selección de las mismas para garantizar la continuidad de esta investigación. Teniendo en cuenta los resultados obtenidos, se prosigue en el capítulo 3, con el análisis y diseño de la aplicación que se va a desarrollar. En el actual capítulo se presentará el Modelo de Negocio o el Modelo de Dominio, según corresponda en este caso. Se definirán los requisitos funcionales y no funcionales, así como el diagrama de casos de usos del sistema con la descripción detallada del mismo. Entre los puntos tratados en este capítulo se mostrará el prototipo de interfaz de usuario, el diagrama de despliegue, entre otros aspectos.

3.2 Descripción del Modelo de Negocio y del Modelo de Dominio.

➤ Modelado de Negocio.

El flujo de trabajo Modelo de Negocio es una labor de los analistas de procesos de negocio, quienes tienen la misión de entender cómo funciona el negocio. Estos modelan el mismo a través de diagramas de actividades donde se refleja la secuencia de pasos que se llevan a cabo, las personas beneficiadas con las acciones realizadas y las que realizan las actividades. El Modelado del Negocio tiene como objetivos:

- Comprender la estructura y la dinámica de la organización en la cual se va a implantar el sistema.
- Comprender los problemas actuales de la organización e identificar las mejoras potenciales.
- Asegurar que los consumidores, usuarios finales y desarrolladores tengan un entendimiento común de la organización.
- Derivar los requerimientos del sistema que va a soportar la organización **(Pressman, 2001)**.

En resumen, el objetivo del Modelo de Negocio (Model of Business) es describir los procesos existentes u observados, con el propósito de comprenderlos. Se especifican aquí qué procesos del negocio soportará el sistema. Además de identificar los objetos del negocio implicados, este modelo establece las competencias que se requieren de cada proceso, sus trabajadores, sus responsabilidades y las operaciones que llevan a cabo.

➤ **Modelado del Dominio.**

El Modelo de Negocio de RUP incluye toda la organización, sus relaciones, sus procesos, todo. Sin embargo, el Modelo de Dominio se centra en una parte del negocio, la relacionada con el ámbito del proyecto. En este contexto el término dominio representa una parte del negocio. El Modelo de Dominio ayuda a comprender los conceptos que utilizan los usuarios, los conceptos con los que trabajan y con los que deberá trabajar la aplicación. El proceso para su elaboración tiene tres pasos. En primer lugar identificar las Clases Conceptuales, después dibujarlas en un Diagrama de Clases y finalmente añadir Relaciones y Atributos (**Pressman, 2001**).

Resumiendo, se llama “Modelo de Dominio” a la representación visual de los conceptos u objetos que nos interesa, sus características y las relaciones entre dichos conceptos. Es el mecanismo fundamental para comprender el dominio del problema y para establecer conceptos comunes. Es un diccionario visual del dominio del problema.

3.3 Selección y explicación del modelo a utilizar.

A continuación se ha elaborado una tabla comparativa para poder proporcionar un mayor entendimiento, sobre el por qué de la utilización de uno de estos dos modelos para la continuación de esta aplicación.

Modelo de Dominio	Modelo de Negocio
Las clases se obtienen de la base de del conocimiento de unos pocos expertos del dominio o posiblemente del conocimiento de asociado con sistemas similares al que estamos desarrollando.	Las entidades se derivan a partir de los clientes del negocio, identificando los casos de uso del negocio y después buscando las entidades.
Las clases tienen atributos pero normalmente ninguna o muy pocas operaciones.	Se identifican las entidades y los trabajadores que participarán en la realización de los casos de uso. Además se identifica cómo utilizarán esos trabajadores las entidades a través de operaciones que debe ofrecer cada entidad.
Puede hacer la traza de las clases hasta la experiencia de los expertos del dominio. No hay forma evidente de hacer la traza entre el Modelo de Dominio y los casos de usos del sistema.	Puede hacer la traza de la necesidad de cada elemento del modelo hasta los clientes, es decir, permite hacer el seguimiento de las necesidades del cliente a lo largo del camino completo.

Tabla 1 Comparación entre el Modelo de Dominio y el Modelo de Negocio.

Teniendo en cuenta todo lo explicado anteriormente y debido a que no se tienen bien definidas las fronteras del negocio para la aplicación a desarrollar, se decide realizar un Modelo de Dominio. El Modelo

de Dominio es empleado principalmente cuando la información tiene múltiples orígenes. Esto se realiza con el fin de lograr una mejor comprensión de la estructura de la organización. En este modelo se representan los tipos de objetos más importantes que existen en el entorno donde estará el sistema. Se representa en UML con un Diagrama de Clases en el que se representan las clases conceptuales, asociaciones entre las clases conceptuales y atributos de la clase conceptuales.

Seguidamente se presenta el diagrama de clases del dominio para la aplicación a desarrollar en esta investigación. Además se procederá a explicar cada uno de los conceptos que conforman al mismo.

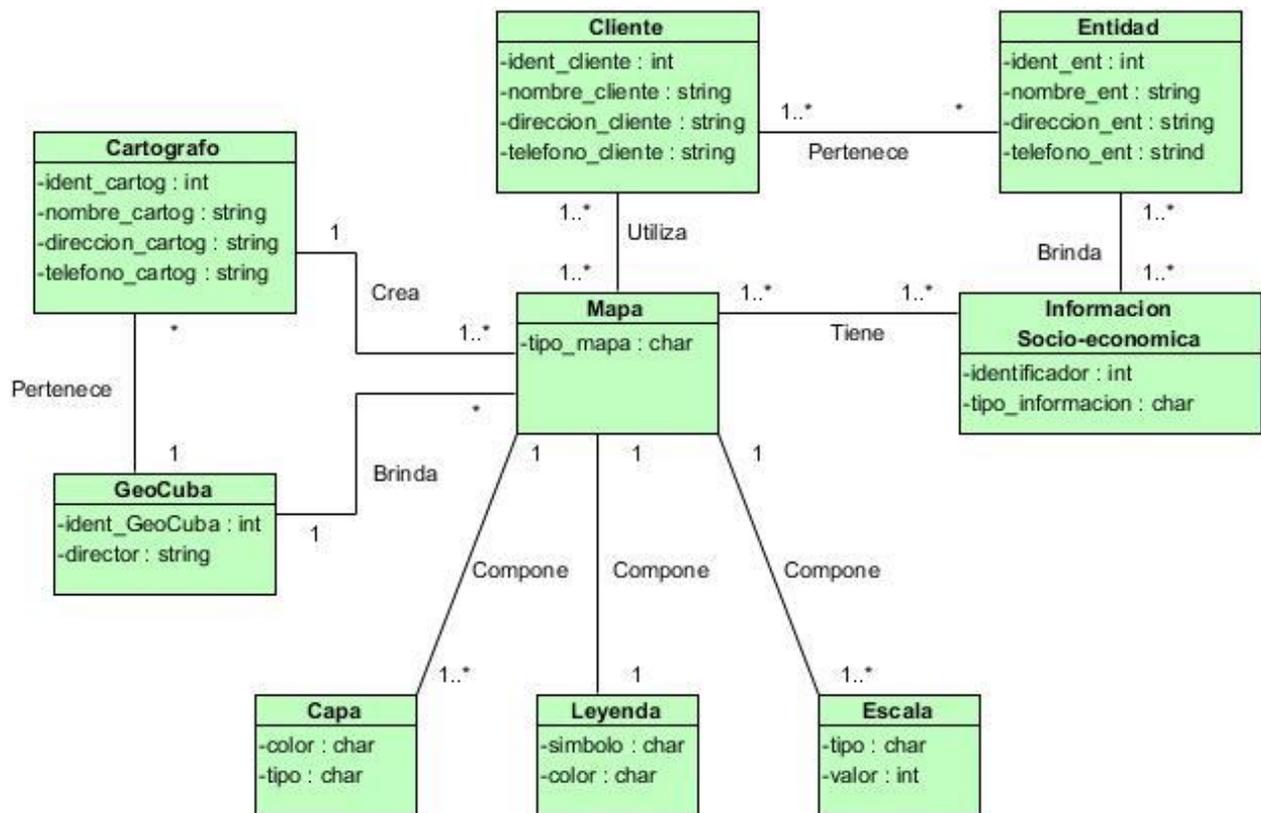


Ilustración 1 Modelo de Dominio.

- **Cartógrafo.**

Es la persona que se encarga del estudio y de la elaboración de los mapas geográficos, territoriales y de diferentes dimensiones lineales y demás.

- **GeoCuba.**

Es un grupo empresarial cubano que se dedica a la elaboración, producción y venta de mapas y planos con diversos fines. Realiza estudios geográficos e investigaciones científicas en ramas del campo de las ciencias geográficas. GeoCuba es muy reconocido por entregarle a sus clientes, productos informativos con una alta calidad y fiabilidad.

- **Ciente.**

Es la persona que trabaja en una entidad, empresa u organismo y que necesita consultar algún tipo de información incluida en un mapa.

- **Entidad.**

Una entidad es la empresa, institución u organismo que solicita un servicio determinado. En este caso sería un servicio mediante la utilización de un mapa. La entidad debe de brindar la información socioeconómica referente a él mismo.

- **Mapa.**

Representación geográfica de la Tierra o parte de ella en una superficie bidimensional, generalmente plana.

- **Información socioeconómica.**

Se presenta un conjunto organizado de datos, representando información demográfica, social y económica de cualquier lugar de interés del país.

- **Escala.**

Relación entre la distancia que separa dos puntos en un mapa y la distancia real de esos dos puntos en la superficie terrestre.

- **Leyenda.**

Explicación de los símbolos, los colores, las tramas y los sombreados empleados en el mapa. Suele encontrarse a pie de página o en un recuadro, situado en sus márgenes o bien en su dorso.

- **Capas.**

Las capas son una forma de organizar la información temática para la elaboración de los SIG. Además el sistema permite activar o no las capas disponibles.

➤ **Descripción del diagrama.**

En este diagrama se representa la interacción entre las clases conceptuales identificadas. GeoCuba es un grupo empresarial donde se determinan todos los procesos geológicos desarrollados en el país. A esta pertenecen varios cartógrafos que son los encargados de crear los mapas, que se le proporcionan a las entidades. A los mapas se le agrega toda la información socio-económica proporcionada por las entidades. Además están compuestos por varias escalas representativas, leyendas que permiten un mejor entendimiento de los mismos y las capas. Son utilizados por los clientes pertenecientes a una entidad específica.

3.4 Requerimientos funcionales y requerimientos no funcionales.

De acuerdo con RUP, el término requerimiento se refiere a una condición que un sistema puede cumplir. Un requerimiento, fundamentalmente debe cumplir con ser consistente y no ambiguo, preciso, verificable, que pueda ser leído y modificable. Se pueden separar los requerimientos en grandes cantidades que faciliten su manejo, lo que conlleva en general a una mejora del manejo del proyecto en su totalidad (**Pressman, 2005**). Los requerimientos pueden ser clasificados en 2 grandes categorías, requerimientos funcionales y requerimientos no funcionales.

➤ **Requerimientos funcionales.**

La plataforma contará con un conjunto de funciones necesarias para la representación, modelación y análisis de la información geográfica. A estas funciones se les llamará requerimientos funcionales. A continuación se identifican y describen los Requerimientos Funcionales (RF) para la aplicación a desarrollar.

- RF 1** Acercar determinada región del mapa.
- RF 2** Alejar determinada región del mapa.
- RF 3** Visualizar todo el mapa.
- RF 4** Visualizar diferentes regiones del mapa.
- RF 5** Utilizar mapa de referencia.
- RF 6** Recentrar mapa.
- RF 7** Dibujar un punto en el mapa.
- RF 8** Dibujar una línea en el mapa.
- RF 9** Dibujar un polígono en el mapa.
- RF 10** Dibujar un polígono regular en el mapa.
- RF 11** Mover el dibujo en el mapa.
- RF 12** Borrar el dibujo del mapa.
- RF 13** Medir la distancia entre dos o más puntos.
- RF 14** Visualizar ventana Control de Selección de Capas.
- RF 15** Visualizar ventana Mapa de Referencia.
- RF 16** Realizar control de selección de capas.
- RF 17** Ver las coordenadas de la posición del cursor en el mapa.
- RF 18** Utilizar navegador virtual.

- **Descripción de los requerimientos funcionales:**

RF 1 Acercar determinada región del mapa: Con este requerimiento se quiere que el usuario pueda realizar la función de zoom aumentar para aumentar una visualización sobre el mapa.

RF 2 Alejar determinada región del mapa: Con este requerimiento se quiere que el usuario pueda realizar la función de zoom disminuir para disminuir una visualización sobre el mapa.

RF 3 Visualizar todo el mapa: Con este requerimiento se quiere que el usuario pueda realizar la función de zoom extenso para volver a ver el mapa completo.

RF 4 Visualizar diferentes regiones del mapa: Con este requerimiento se quiere que el usuario, con el puntero del mouse, pueda mover el mapa de la posición de la vista que se presenta.

RF 5 Utilizar mapa de referencia: Permite que el usuario pueda moverse por el mapa, en un rango de escala determinado a partir de un rectángulo equivalente al extend⁹ visualizado.

RF 6 Recentrar mapa: Con este requerimiento se quiere que el usuario pueda recentrar el mapa según la posición puntual que realizó sin modificar la escala del mapa.

RF 7 Dibujar un punto en el mapa: Con este requerimiento se quiere que el usuario pueda insertar un objeto geográfico en el mapa. Este objeto se agregará a la capa temporal y va a ser un punto.

RF 8 Dibujar una línea en el mapa: Con este requerimiento se quiere que el usuario pueda insertar un objeto geográfico en el mapa. Este objeto se agregará a la capa temporal y va a ser una línea.

RF 9 Dibujar un polígono en el mapa: Con este requerimiento se quiere que el usuario pueda insertar un objeto geográfico en el mapa. Este objeto se agregará a la capa temporal y va a ser un polígono.

RF 10 Dibujar un polígono regular en el mapa: Con este requerimiento se quiere que el usuario pueda insertar un objeto geográfico en el mapa. Este objeto se agregará a la capa temporal y va a ser un polígono regular.

⁹ Extend: Es la dimensión de representación de un mapa en la pantalla.

RF 11 Mover el dibujo en el mapa: Con este requerimiento se quiere que el usuario pueda mover un objeto geográfico en la capa temporal del mapa.

RF 12 Borrar el dibujo del mapa: Con este requerimiento se quiere que el usuario pueda eliminar un objeto geográfico en la capa temporal del mapa.

RF 13 Medir la distancia entre dos o más puntos: Con este requerimiento se quiere que el usuario pueda realizar trazos por el mapa y visualice la distancia total acumulada así como la existente entre los dos últimos vértices trazados.

RF 14 Visualizar ventana Control de Selección de Capas. Con este requerimiento se quiere que el usuario pueda visualizar y ocultar la ventana correspondiente al Control de Selección.

RF 15 Visualizar ventana Mapa de Referencia. Con este requerimiento se quiere que el usuario pueda visualizar y ocultar la ventana correspondiente al Mapa de Referencia.

RF 16 Realizar control de selección de capas: Con este requerimiento se quiere que el usuario pueda identificar las capas que son seleccionables o que se pueden consultar en un mapa.

RF 17 Ver las coordenadas de la posición del cursor en el mapa: Con este requerimiento se quiere que al posicionar el mouse en cualquier área del mapa, se visualice en la parte inferior derecha del mismo, sus coordenadas x, y.

RF 18 Utilizar navegador virtual: Permite que el usuario pueda mover el mapa en las cuatro direcciones principales: norte, sur, este y oeste.

➤ **Requerimientos no funcionales.**

Los Requerimientos no funcionales (RNF) son propiedades o cualidades que el producto debe tener. Normalmente están vinculados a requerimientos funcionales, es decir una vez se conozca lo que el sistema debe hacer se puede determinar cómo ha de comportarse, qué cualidades debe tener o cuán rápido o grande debe ser (**Pressman, 2005**).

Por tanto, se resume que los requerimientos no funcionales se refieren a todos los requisitos que ni describen información a guardar, ni funciones a realizar.

- **Usabilidad.**
 - El sistema debe poder ser utilizado por cualquier persona con conocimiento básico en SIG.
 - La información deberá estar disponible en todo momento.
 - Se debe de garantizar que siempre está visible la opción de Ayuda, para posibilitar que el usuario la consulte ante cualquier duda.
 - La aplicación deberá correr tanto en plataformas Linux, como en plataforma Windows 2000 o superior para garantizar que pueda ser utilizada por usuarios que trabajen con cualquiera de los dos sistemas.
- **Fiabilidad.**
 - El tiempo medio de reparación, en caso de un fallo es de 7 días.
- **Eficiencia.**
 - El tiempo de respuesta del sistema ante las peticiones de los usuarios estará dado por la cantidad de información a procesar. Aunque se garantiza rapidez en la respuesta ante estas peticiones por las características propias del framework Geomoose.
- **Soporte.**
 - La aplicación recibirá mantenimiento en el período de tiempo determinado por el equipo de desarrollo y los clientes.
- **Restricciones de diseño.**
 - El sistema debe tener una apariencia profesional, sencilla y amigable para el usuario, que lo guíe paso a paso hasta alcanzar el objetivo deseado.
 - La aplicación debe de diseñarse sobre una arquitectura cliente-servidor.
 - Se deben emplear los estándares establecidos (diseño de interfaces, base de datos y codificación).
- **Software.**
 - Las computadoras que utilizarán el software deberán tener instalado cualquier distribución de Linux, así como el sistema operativo Windows 2000 o una versión superior.

- **Interfaces de hardware**

Para las PCs clientes:

- Se requiere tengan tarjeta de red.
- Al menos 64 MB de memoria RAM.
- Se requiere al menos 100 MB de disco duro.
- Procesador 512 MHz como mínimo.

Para los servidores:

- Se requiere tarjeta de red.
- El Servidor de Mapas tenga como mínimo 2GB de RAM y 2GB de disco duro.
- El Servidor de BD tenga como mínimo 2GB de RAM y 10GB de disco duro.
- Procesador 3 GHz como mínimo.

- **Interfaces de software**

La construcción de la aplicación funcionará bajo los conceptos de arquitectura cliente/servidor. Por tanto, el servidor del usuario final debe tener como requerimientos mínimos de software:

Para las PCs clientes:

- Un Navegador como Mozilla Firefox, Zafari u otro navegador que cumpla con los estándares W3C.
- Sistema operativo: GNU/Linux o Windows.

Para los Servidores:

- Sistema operativo GNU/Linux.
- Servidor Web Apache 2.0 o superior, con módulo PHP 5 configurado con la extensión pgsql incluida.
- PostgreSQL como Sistema Gestor de Base de Datos.
- PostGIS como extensión de PostgreSQL como soporte de datos espaciales.
- MapServer 5.2.2 o superior, con extensión PHP mapscript.

3.5 Diagrama de Casos de Usos del Sistema.

➤ **Actores del sistema.**

Los actores representan papeles que las personas o dispositivos juegan como impulsores del sistema. Un actor es algo que se comunica con el sistema o producto y que es externo a un sistema en sí mismo **(Pressman, 2005)**.

Un actor es una agrupación uniforme de personas, sistemas o máquinas que interactúan con el sistema que se está construyendo de la misma forma **(Santiago, 2002)**.

En resumen, un actor no es más que un conjunto de roles que los usuarios de Casos de Uso desempeñan cuando interaccionan con estos Casos de Uso. Los actores representan a terceros fuera del sistema que colaboran con el mismo. Una vez identificado los actores del sistema, queda identificado el entorno externo del sistema. En la aplicación se cuenta con los actores del sistema que se describen a continuación:

Actor	Descripción
	Es la persona, sistema o máquina que interactúa con el sistema.

Tabla 2 Actores del sistema.

➤ **Casos de Uso del Sistema (CUS).**

Un caso de uso proporciona a los desarrolladores una visión de lo que quieren los usuarios. Se enfoca en lo que hace el sistema más que en la forma como lo hace. Además en un caso de uso, un actor que utiliza el sistema inicia un evento que desencadena una serie de interacciones relacionadas en el sistema **(Kendall & Kendall, 2005)**.

La forma en que los actores usan el sistema se representa con un caso de uso. Los casos de uso son como una parte de las funcionalidades que tendrá la aplicación. Esta parte brindará resultados concretos a los actores de este sistema. De manera más precisa, un caso de uso del sistema especifica una

secuencia de acciones que el sistema puede llevar a cabo interactuando con sus actores, incluyendo alternativas dentro de la secuencia.

Diagrama de Casos de Uso del Sistema.

Diagrama de Casos de Uso: Conjunto de casos de uso, actores y sus relaciones. Son importantes para modelar el comportamiento de un sistema, un subsistema o una clase. Cada uno muestra un conjunto de casos de uso, actores y sus relaciones. A continuación se muestra el DCUS correspondiente a esta investigación.

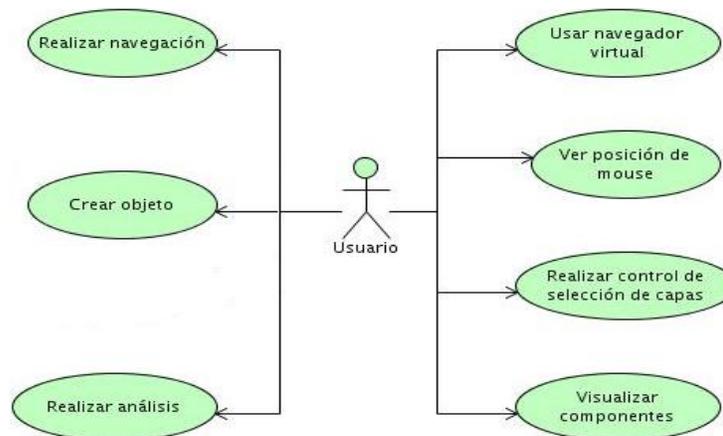


Ilustración 2 Diagrama de Casos de Uso del Sistema.

➤ **Especificación de los Casos de Uso del Sistema.**

Uno de los puntos fundamentales para llevar a cabo la descripción de los CUS, es la identificación del interés de los actores por los Casos de Uso. Priorizar el desarrollo de unos casos de usos de otros, permite determinar los que son críticos (o más importantes) para la aplicación. A continuación se describe uno de los CUS que fueron identificados, este es Realizar control de selección de capas.

Realizar control de selección de capas.

Caso de Uso:	Realizar control de selección de capas
Actores:	Usuario
Propósito	Este caso de uso se lleva a cabo con el objetivo de activar o desactivar las capas disponibles en la aplicación.

Resumen:	El caso de uso se inicia cuando el actor desea activar o desactivar una capa en la ventana de control de selección de capas y termina cuando se muestra en el mapa la capa activada o se deja de ver la capa desactivada.	
Precondiciones:	-	
Referencias	RF 18	
Prioridad	Crítico	
Flujo Normal de Eventos		
Acción del Actor	Respuesta del Sistema	
1. El actor selecciona en la ventana de control de selección de capas la capa que desea activar o desactivar.	2. El sistema muestra en el mapa la capa activada o no visualiza en el mapa la capa desactivada.	
Poscondiciones	Se activan o desactivan las capas disponibles en la aplicación.	

Tabla 3 Descripción del Casos de Uso del Sistema Exportar mapa.

Seguidamente se facilita una tabla con una breve descripción de los demás CUS. La especificación de estos CUS, se encuentra en el documento “Descripción de CUS” de la carpeta Ingeniería adjuntada a esta investigación.

Actor	Caso de Uso	Resumen	Prioridad
Usuario	Realizar navegación	El caso de uso se inicia cuando el usuario desea mover, ampliar, disminuir, utilizar el mapa de referencia, visualizar todo el mapa o recentar el mapa en dependencia de su selección. Este es el CUS que trabaja directamente sobre el mapa, mostrando lo que el actor desea y termina cuando el sistema muestra el resultado en el monitor.	Crítico
Usuario	Visualizar Componentes	El caso de uso se inicia cuando el actor desea visualizar o no la ventana del mapa de referencia y la de control de selección de capas y termina cuando realiza la operación.	Secundario
Usuario	Análisis	El caso de uso se inicia cuando el actor desea medir la distancia existente entre 2 o más puntos marcados en un mapa y termina cuando el sistema muestra los valores correspondientes.	Crítico
Usuario	Crear objeto	El caso de uso se inicia cuando el actor desea dibujar un objeto de forma puntual, lineal, poligonal o poligonal regular o bien cuando desee mover o eliminar alguno de los objetos creados. Termina cuando el sistema muestra el(los) objeto(s) dibujado(s) en el mapa, o lo(s) mueve o elimina.	Auxiliar

Usuario	Ver posición del mouse	El caso de uso se inicia cuando el actor mueve el cursor sobre el mapa y termina al cerrar la aplicación.	Secundario
Usuario	Usar navegador virtual	El caso de uso se inicia cuando el actor pulsa sobre alguno de los botones de norte, sur este u oeste y termina cuando se redimensiona el mapa.	Secundario

Tabla 4 Resumen de los Casos de Uso del Sistema.

3.6 Prototipo de Interfaz de usuario.

La interfaz de usuario es el medio por que el usuario puede comunicarse con una máquina, un equipo o una computadora. Además comprende todos los puntos de contacto entre el usuario y el equipo. Normalmente suelen ser fáciles de entender y fáciles de accionar.



Ilustración 3 Prototipo de interfaz de usuario.

➤ Descripción del prototipo de Interfaz de usuario.

Seguidamente se realiza una explicación del prototipo de Interfaz de usuario, para un mayor entendimiento del entorno de trabajo.

1. En esta posición va el banner de la aplicación.
2. Esta es la barra de menú.

A. El menú Archivo: Donde se tendrá la opción de "Cerrar" y de la "Ayuda".

- B. El menú Navegación: Donde se tienen las opciones de “Aumentar”, “Disminuir”, “Ver todo”, “Recentrar mapa” y “Paneo”.
 - C. El menú Dibujo: Donde se tiene las opciones “Dibujar punto”, “Dibujar línea”, “Dibujar polígono”, “Dibujar polígono regular”, “Mover dibujo”, “Borrar dibujo”.
 - D. El menú Análisis: Donde se tienen las opciones de “Medir distancia” y “Calcular área y perímetro”.
3. Esta es la barra de herramientas. En esta barra se representan los iconos que permiten ejecutar las funcionalidades mencionadas en la barra de menú como la Ayuda (E).
 4. En este espacio es donde se lleva el control de las capas que están seleccionadas.
 5. En este espacio se visualiza la ventana de “Mapa de referencia”.
 6. En esta pequeña barra se tiene el aviso de copyright, ya que este aviso de autorización, se incluirá en todas las copias o partes sustanciales del software que utilice Geomoose.
 7. En este espacio se tiene el mapa con el que se va a trabajar. Es donde se visualizan todos los cambios del mismo.

3.7 Diagrama de Despliegue.

El diagrama UML de despliegue representa una vista estática de la configuración en tiempo de ejecución de los nodos que intervienen en el proceso y de los componentes que se ejecutan en esos nodos (**Aycart Pérez, Gibert Ginestà, Hernández Matías, & Mas Hernández, 2007**). Es decir, en este modelo se representan las conexiones entre los elementos de procesamiento en el sistema. Se conforma por la representación de uno o más nodos, dispositivos y conexiones entre ellos. Seguidamente se presenta el Diagrama de Despliegue de la aplicación:

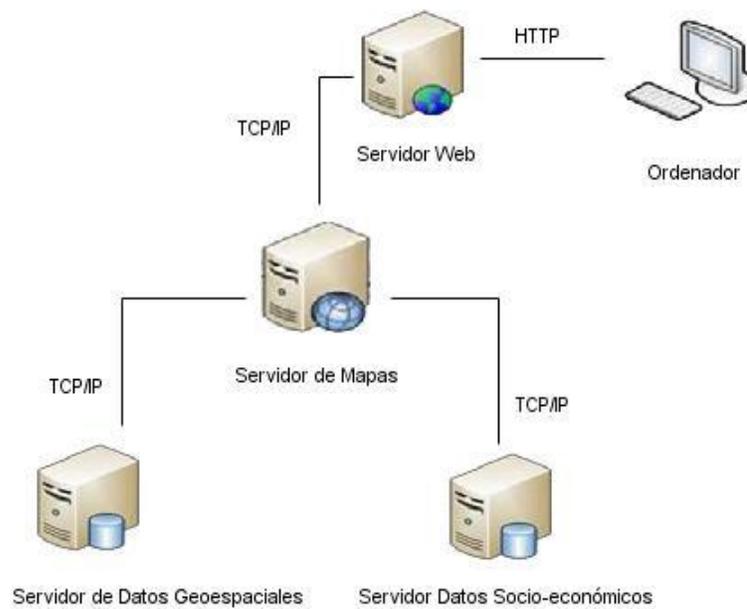


Ilustración 4 Diagrama de Despliegue.

➤ Algunos conceptos importantes:

- **Nodos:** Elementos de procesamiento con al menos un procesador, memoria y posiblemente otros dispositivos.
- **Dispositivos:** Nodos estereotipados sin capacidad de procesamiento en el nivel de abstracción que se modela.
- **Conectores:** Expresa el tipo de conector o protocolo utilizado entre el resto de los elementos del modelo.
- **TCP:** Protocolo de control de trasmisión.
- **IP:** Protocolo de internet.
- **HTTP:** Protocolo de transferencia de hipertexto.

3.7.1 Descripción de los nodos:

➤ **Servidor Web.**



Ilustración 5 Servidor Web.

- **Requerimientos de Hardware.**

Procesador Pentium IV 2x2 caché. 3 GHz como mínimo.

Memoria 2GB de RAM.

- **Subsistemas de Implementación.**

En este nodo se ejecutarán todas las funcionalidades del Servidor Web.

➤ **Servidor de Base de Datos.**



Ilustración 6 Servidor de Base de Datos.

- **Requerimientos de Hardware.**

Procesador Pentium IV 2x2 caché. 3 GHz como mínimo.

Memoria 2GB de RAM.

Almacenamiento en disco con capacidad igual o superior a los 10 GB (como mínimo). Tecnología de respaldo de datos históricos.

- **Subsistemas de Implementación.**

En este nodo se estará ejecutando el Gestor de Base de Datos PostgreSQL, con su extensión PostGIS.

➤ Servidor de Mapas.



Ilustración 7 Servidor de Mapas.

- **Requerimientos de Hardware.**

Procesador Pentium IV ó superior. 3 GHz como mínimo.

Memoria 2GB de RAM.

Disco duro de 5GB mínimo libre.

- **Subsistemas de Implementación.**

Este nodo contendrá el servidor de mapas MapServer con su fichero de configuración MapFile.

➤ Cliente Web.



Ilustración 8 Cliente Web.

- **Requerimientos de Hardware.**

Procesador Pentium 3.

256 MB de memoria RAM.

Se requiere al menos 100 MB de disco duro.

Procesador 512 MHz como mínimo.

- **Requerimientos de Software.**

Un Navegador como Mozilla Firefox, Zafari u otro navegador que cumpla con los estándares W3C.

Sistema operativo: GNU/Linux y Windows.

3.8 Arquitectura lógica.

➤ Patrones.

Los patrones son un esqueleto básico que cada diseñador adapta a las peculiaridades de su aplicación. Además permiten realizar soluciones a problemas comunes. Los patrones se pueden clasificar según la escala o nivel de abstracción en patrones de diseño, patrones de arquitectura, patrones de distribución, entre otros.

• Patrones de distribución.

Definen una estructura para el modelo de despliegue. Existen 3 tipos de patrones de distribución de los cuales se selecciona uno para el desarrollo de esta aplicación. Estos son: Arquitectura Cliente/Servidor, Arquitectura Tres capas y Arquitectura Orientada a Servicios.

Arquitectura: Se considera la arquitectura, como una estructura de los componentes de un programa o sistema, sus interrelaciones, los principios y reglas que gobiernan su diseño y evolución en el tiempo.

• Patrón de distribución a utilizar.

Se utiliza en la aplicación a desarrollar, la Arquitectura Cliente/Servidor, ya que la solución a implementar es una aplicación web. Por tal motivo se requiere de una arquitectura que permita la interacción entre la aplicación y el cliente. Se divide en dos partes claramente diferenciadas, la primera es la parte del servidor y la segunda la de un conjunto de clientes.

Servidor: Normalmente el servidor es una máquina potente que actúa de depósito de datos y funciona como un sistema gestor de base de datos. Son los encargados de proporcionar un servicio al cliente y devolver los resultados en forma de una respuesta. Generalmente un servidor puede tratar múltiples peticiones, es decir varios clientes al mismo tiempo.

Cliente: Por otro lado los clientes suelen ser estaciones de trabajo que inician el diálogo o solicitan varios servicios al servidor. Ambas partes deben estar conectadas entre sí mediante una red.

Algunas aplicaciones clientes son Mozilla Firefox, Microsoft Outlook e Internet Explorer.

Entre las principales características de la arquitectura Cliente-Servidor, se destacan las siguientes:

- El cliente y el servidor pueden actuar como una sola entidad y también como entidades separadas, realizando actividades o tareas independientes.
- El cliente no necesita conocer la lógica del servidor, sólo su interfaz externa.

- El cliente no depende de la ubicación física del servidor, ni del tipo de equipo físico en el que se encuentra, ni de su sistema operativo.
- Los cambios en el servidor implican pocos o ningún cambio en el cliente.
- Las funciones de cliente y servidor pueden estar en plataformas separadas o en la misma plataforma.

Una representación gráfica de este tipo de arquitectura sería la siguiente:



Ilustración 9 Arquitectura Cliente/Servidor.

- **Patrones de diseño.**

Un patrón de diseño provee un esquema para refinar los subsistemas o componentes de un sistema de software, o las relaciones entre ellos. Describe la estructura comúnmente recurrente de los componentes en comunicación, que resuelve un problema general de diseño en un contexto particular (**Buschmann, 1996**).

Estos están más próximos a la implementación y su uso no se refleja en la estructura global del sistema. Principalmente se clasifican en estructurales, de comportamiento y de creación. Estos patrones están previamente definidos en la arquitectura del framework Geomoose.

Se utilizan los patrones GRASP¹⁰, que son patrones generales de software para asignación de responsabilidades. Aunque se considera que más que patrones propiamente dichos, son una serie de

¹⁰ GRASP: General Responsibility Assignment Software Patterns.

buenas prácticas de aplicación recomendable en el diseño de software. Algunos de los patrones GRASP más conocidos son:

- Creador: El patrón creador ayuda a identificar quién debe ser el responsable de la creación (o instalación) de nuevos objetos o clases.
- Controlador: El patrón controlador es un patrón que sirve como intermediario entre una determinada interfaz y el algoritmo que la implementa. Es la que recibe los datos del usuario y la que los envía a las distintas clases según el método llamado.
- Alta cohesión y bajo acoplamiento: Los conceptos de cohesión y acoplamiento están íntimamente relacionados. Un mayor grado de cohesión implica uno menor de acoplamiento.

- **Patrones de arquitectura.**

Son esquemas de organización general de un sistema. Especifican una serie de subsistemas y sus responsabilidades. Además incluyen reglas para organizar las relaciones entre ellas los patrones arquitectónicos. Como patrones arquitectónicos se utilizan:

- Patrón de Arquitecturas Orientadas a Objetos (AOO).

Los componentes del estilo se basan en principios Orientados a Objetos: encapsulamiento, herencia y polimorfismo. En cuanto a las restricciones, puede admitirse o no que una interfaz pueda ser implementada por múltiples clases.

3.9 Clases del diseño.

En el lenguaje UML, se modelan diferentes diagramas para presentar el diseño de la aplicación. Uno de estos diagramas es el de paquete. Un diagrama de paquetes muestra cómo un sistema está dividido en agrupaciones lógicas mostrando las dependencias entre esas agrupaciones (**Pressman, 2005**). Los diagramas de paquetes suministran una descomposición de la jerarquía lógica de un sistema. A continuación se anexa una imagen con el diagrama de paquetes del framework Geomoose.

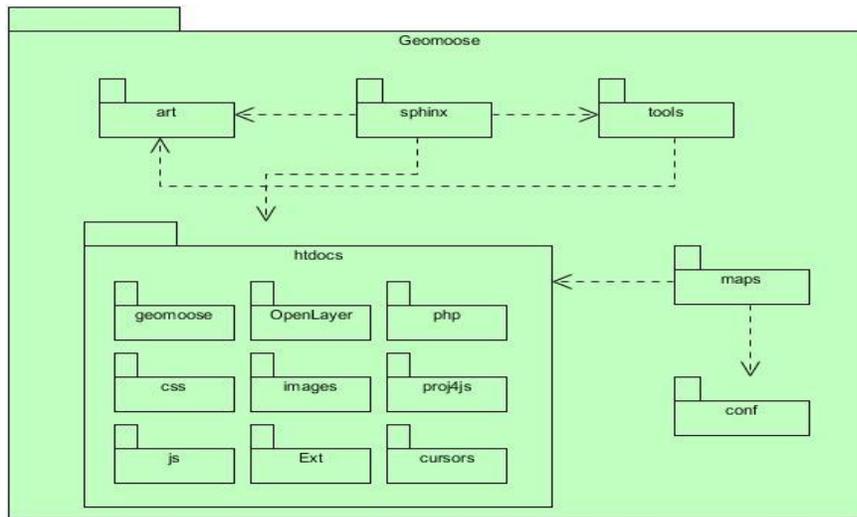


Ilustración 10 Paquete de clases del diseño del framework Geomoose.

Los paquetes relacionados contienen clases que intervienen en el funcionamiento de la aplicación. Seguidamente se muestra una breve explicación de cada uno de estos paquetes.

- art: Esta carpeta contiene Gráficos Vectoriales Escalables (SVG) y versiones de los gráficos utilizados en las interfaces.
- htdocs: Contiene los paquetes relacionados con el diseño del framework.
- maps: Contiene todo lo relacionado con los archivos .map, además incluye ejemplos de mapfile para probar la aplicación.
- conf: En este paquete se encuentran todos los ficheros relacionados con las configuraciones del framework.
- css: Contiene todos los estilos que se pueden llegar a utilizar en la aplicación.
- php: En este paquete se encuentran todas las páginas php que se pueden utilizar para el desarrollo del sistema.
- geomoose: En este se encuentran todos los ficheros .js que se puedan utilizar en la aplicación.
- proj4js: En este se encuentran los archivos que contienen las proyecciones que se pueden utilizar.
- tools: Este paquete contiene scripts escritos en Python que combinan iconos e imágenes, así como scripts.

- sphinx: Contiene documentos que guían la configuración del sistema.
- cursors: Están las imágenes para el cursor que utiliza Geomoose.
- images: Están las imágenes que utiliza Geomoose.
- OpenLayers: Se encuentra en esta carpeta el framework OpenLayers que es el que utiliza Geomoose para algunas de sus funciones.
- js: En esta carpeta se encuentra una página js, llamada principal que es para trabajar con las barras.
- Ext: En esta carpeta se encuentra la librería de Ext que se utiliza para la creación de las barras de menú y de herramientas.

Otro de los diagramas es el de clases del diseño. Un diagrama de clases es un tipo de diagrama estático que describe la estructura de un sistema mostrando sus clases, atributos y las relaciones entre ellos (**Pressman, 2005**). Estos se representan en diagramas de clases UML. Con ellos se identifican las clases estereotipadas que se van a utilizar.

Se presenta a continuación el diagrama de clases del diseño del CUS Realizar control de selección de capas. Los demás diagramas se encuentran en el documento “Diagramas de clases del diseño” que se encuentra en la carpeta Ingeniería adjunta a esta investigación.

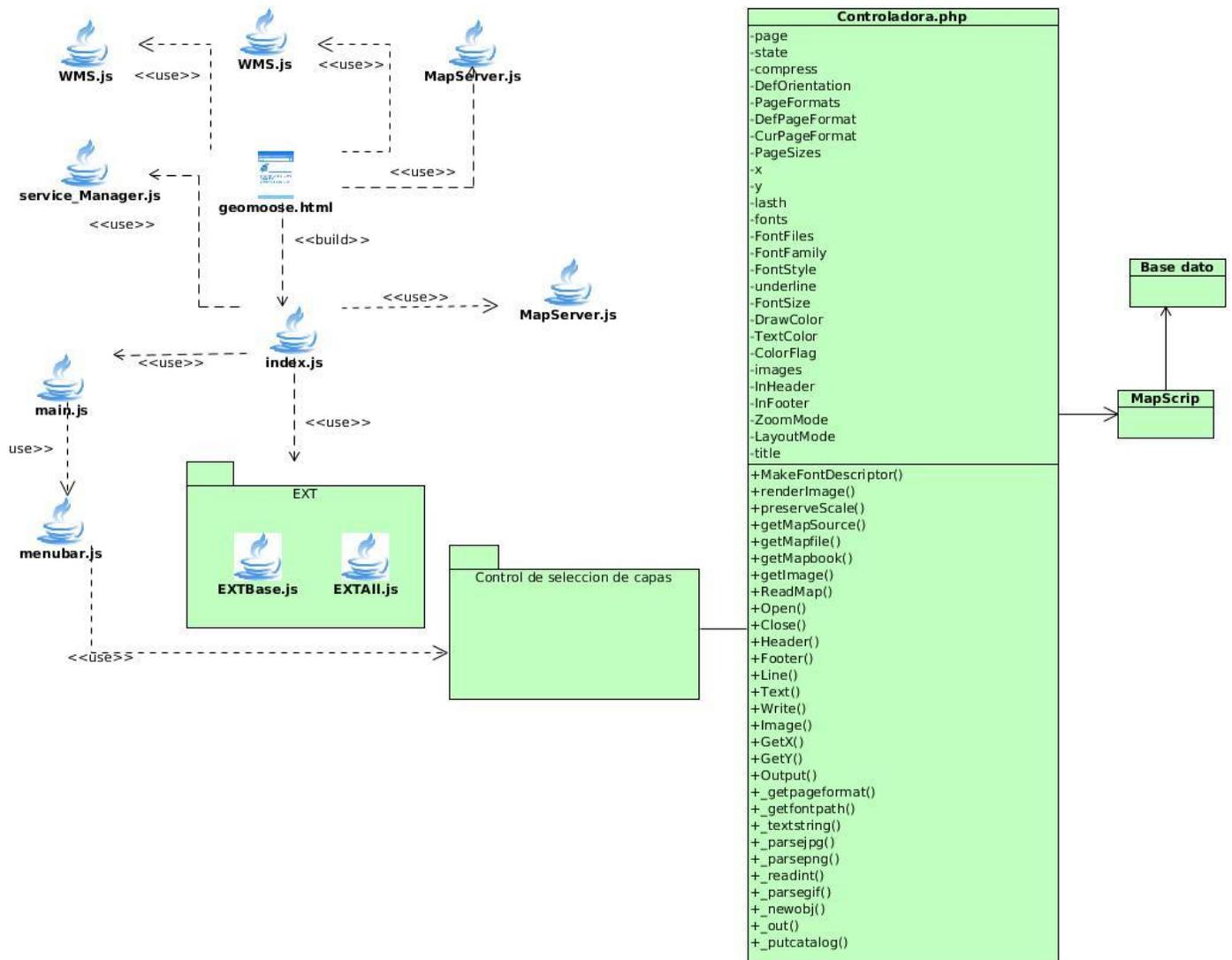


Ilustración 11 Clases del diseño del CUS Realizar control de selección de capas.

Cada uno de los diagramas de Clases de Diseño posee clases que son comunes para la mayoría de ellos. Estas clases serán descritas a continuación.

- geomoose.html: Integra y recoge todos los datos y funciones realizadas por cada una de las .js que intervienen en el Caso de Uso.
- geomoose.js: Donde se encuentran definidas las funciones para la navegación.

- index.js: Es la que tiene que ver con la presentación de la aplicación. Es donde se crean todos los componentes visuales de dicha aplicación.
- main.js: Es la que controla las funciones presentadas en la barra de herramientas. Esta es una vía mucho más rápida de acceder a las funcionalidades de la aplicación.
- menubar.js: Es donde se definen todas las funciones correspondientes a los eventos necesarios para el funcionamiento de las funcionalidades en la main.js.
- service_Manager.js: Es la clase manejadora de los envíos y pedidos realizados.
- EXTAll.js y EXTBase.js: Son ficheros que deben de aparecer, ya que definen los componentes visuales que son presentados en la interfaz y gestionados posteriormente por la index.js.
- Controladora: Contiene todas las variables de entorno de la aplicación.

3.10 Diseño de la Base de Datos (BD).

Para lograr el funcionamiento del SIG que se desea implementar durante esta investigación, se ha seleccionado como SGBD a PostgreSQL. Los datos que se cargaran en dicho gestor se clasifican según el tipo de información (datos espaciales y socioeconómicos).

Datos Espaciales: En la cual se almacenan los datos espaciales.

Datos Socioeconómicos: En la cual se almacena la información referente a los objetos que aparecen en dicho mapa.

En esta aplicación no se presenta el modelo de diseño de BD. Teniendo en cuenta que no se encuentran definidas las fronteras del negocio, se establece que el SGDB se utilizará para almacenar la información de la cartografía que se vaya a utilizar en el momento de cargar un mapa. Esto está dado por la necesidad de utilizar esta aplicación para la elaboración de diferentes SIG. Por tal motivo, se considera que no es necesario restringir el uso de la BD a tablas previamente definidas, cuando se conoce que la BD va a variar en dependencia de la cartografía que se utilice.

3.11 Conclusiones parciales.

Haciendo uso de las herramientas y tecnologías seleccionadas anteriormente, se realizó en el presente capítulo la ingeniería del sistema. Tras concluir la etapa de análisis y diseño de la aplicación, quedan documentadas todas las pautas necesarias para la continuidad de la investigación. Con la finalización de este capítulo, se generaron una serie de diagramas que son fundamentales para garantizar la continuidad de la investigación. Además, con la confección de este capítulo, se hizo más sencillo comprender la estructura del framework y el trabajo con éste. Por tales motivos se puede comenzar la implementación del sistema para cumplir con los objetivos planteados desde un inicio y darle respuesta a la situación problemática surgida.

Capítulo 4: Implementación y pruebas.

4.1. Introducción.

El presente capítulo es el último de esta investigación. Durante la confección del mismo se estará desarrollando simultáneamente la aplicación SIG sobre el framework Geomoose. Esto se debe a que la investigación ya se encuentra en la etapa de implementación, para posteriormente pasar a la etapa final. Esta etapa final se centra en realizarle las pruebas al sistema para identificar y corregir los errores en el mismo. Esto se hace con el objetivo de garantizar una mayor calidad del producto y mayor aceptación por parte del usuario.

4.2. Pruebas de software.

Las pruebas del software proporcionan directrices sistemáticas para pruebas de diseño que:

- Comprueben la lógica interna y las interfaces de todo componente del software
- Comprueben los dominios de entrada y salida del programa para descubrir errores en su función, comportamiento y desempeño.

Las pruebas de caja negra son las que se aplican a la interfaz del software. Una prueba de este tipo examina algún aspecto funcional de un sistema.

La prueba de caja blanca del software se basa en un examen cercano al detalle procedimental. Se prueban rutas lógicas del software y la colaboración entre componentes, al proporcionar casos de prueba que ejerciten conjuntos específicos de condiciones, bucles o ambos (**Acuña, 2002**).

La prueba del software es un elemento crítico para la garantía de la calidad del software. El objetivo de la etapa de pruebas es garantizar la calidad del producto desarrollado. Para esta aplicación se decidió utilizar la técnica de caja negra. A continuación se muestra un caso de prueba del CUS Realizar control de selección de capas que se ha ido mostrando desde inicios del análisis y diseño de la aplicación. El resto de los casos de prueba se encuentran en el documento "Casos de prueba" que se encuentra contenido en la carpeta Ingeniería adjunta a la investigación.

- **Realizar control de selección de capas.**

Secciones a probar en el Caso de Uso.

Nombre de la sección	Escenario de la sección	Descripción de la funcionalidad
SC 1: Realizar control de selección de capas	EC 1.1: Realizar control de selección de capas	Se muestra en el mapa las capas seleccionadas.

Tabla 5 Secciones a probar en el Caso de Uso Realizar control de selección de capas.

SC 1. Realizar control de selección de capas

Escenario	Respuesta del sistema	Resultado de la prueba	Flujo central
EC 1.1: Realizar control de selección de capas	Se muestra en el mapa las capas seleccionadas.	Satisfactoria	Se selecciona de la ventana de control de selección de capas, la capa que se desee visualizar.

Tabla 6 Escenario 1 de la sección 1 del Caso de Uso Realizar control de selección de capas.

4.3. Conclusiones parciales.

Con el desarrollo de este capítulo, finaliza satisfactoriamente la etapa de implementación y pruebas de la investigación. Se logró utilizar los diagramas generados en el capítulo anterior para implementar la aplicación que daría respuesta a la problemática planteada a inicios de la investigación.

Después de desarrollado el sistema y realizadas las pruebas de caja negra, se puede concluir que el mismo cumple con todas las funcionalidades establecidas en la etapa de análisis y diseño. La aplicación cuenta con interfaces sencillas y mantiene visible la ayuda para posibilitar un mejor desenvolvimiento del usuario en el entorno de trabajo. Además cumple con los principales atributos de calidad que debe de tener un sistema en su primera versión, los cuales son, funcionalidad, eficiencia y usabilidad.

Conclusiones generales.

El desarrollo de los SIGs se ha convertido actualmente en un reto en la UCI. Tras la culminación de aplicaciones de este tipo, han ido surgiendo diferentes deficiencias que han provocado la problemática que fue el punto de partida para esta investigación. Durante el periodo de realización de este trabajo de diploma, se tuvieron en cuenta diferentes aspectos de los SIG que eran imprescindibles para el desarrollo del mismo.

Primero se analizaron los principales conceptos de los SIG y las diferentes características de las posibles tecnologías de este tipo ya existentes en el mundo. Con los resultados obtenidos se lograron establecer factores en común que se utilizaban para el desarrollo de este tipo de aplicaciones.

Seguidamente se realizó un estudio de las herramientas y tecnologías utilizadas para el desarrollo de los SIGs, seleccionando las más adecuadas para la implementación de los SIGs sobre el framework Geomoose.

Posteriormente se identificaron las funcionalidades que debía de tener el sistema y se generaron los artefactos necesarios para garantizar la continuidad de la investigación y el comienzo de la implementación.

Finalmente, al concluir el desarrollo de la aplicación, se le realizaron las pruebas pertinentes al sistema, seleccionando las pruebas de caja negra para poder detectar la existencia de errores de funcionalidad en el mismo.

Después de haber sido llevadas a cabo todas las tareas propuestas para el cumplimiento de los objetivos planteados, se ha arribado a las siguientes conclusiones generales:

- La aplicación desarrollada presenta un importante valor social. Este planteamiento se basa en que el sistema se ha implementado para mejorar las condiciones de trabajo de los programadores que laboran en el proyecto encargado del desarrollo de SIGs web en del departamento de Geoinformática, minimizando el tiempo de desarrollo de estos sistemas.

- El empleo de herramientas y tecnologías libres contribuyó a que se pueda contar con herramientas cubanas para el manejo de información geográfica sin necesidad de invertir en licencias privativas, por lo que representa un aporte económico para el país.
- Esta herramienta contribuyó además con el proceso de informatización y migración hacia el software libre en el que se ve envuelta Cuba actualmente.
- La documentación que genera esta investigación servirá de apoyo para otros desarrolladores del departamento de Geoinformática y personas interesadas en el tema que deseen emplear el framework Geomoose.

Recomendaciones.

Se recomienda para futuras versiones:

- Ampliar la cantidad de requerimientos que ofrece el sistema para aumentar la funcionalidad del mismo.
- La creación de una base de datos que permita guardar los objetos que se crean (el punto, la línea, el polígono y el polígono regular). Esto permitiría que los nuevos objetos creados en un momento determinado, se puedan visualizar cada vez que se inicie la aplicación.
- Establecer la autenticación de usuario para llevar a cabo el control de los usuarios que interactúen con el sistema.

Referencias bibliográficas:

- Acua Cesar. Pruebas de software. Ingeniería del software 2002
- Bosque, Sandra J. «Sistemas de Información Geográfica.» 451. Rialp, 1997.
- Buschmann, F. (1996). Pattern – Oriented Software Architecture. A System of Patterns. Inglaterra.
- Bravo, Javier Domínguez. "Breve Introducción a la Cartografía y a los Sistemas de Información Geográfica (SIG)". Octubre de 2000.
- Carmona, Alvaro de J. Sistemas de Información Geográfica. 2005. http://www.mappinginteractivo.com/plantilla-ante.asp?id_articulo=733.
- Cerda, F. (2007). Obtenido de http://www.techblog.com/talks/netbeans65es_cl.pdf
- Cooper Bowen, R., & Coar, K. (2004). *Apache Práctico*. Estudio Rojas.
- Comas, David. Fundamento de los Sistemas de Información Geográfica. Ariel Geografía, 2007.
- Fawcett, D., & Butler, H. (25 de Noviembre de 2010). Obtenido de <http://nyksadsi.blogspot.com/2010/08/mapserver.html>
- FELICÍSIMO, A. M. (2003). Glosario de términos usados en el trabajo con SIG.
- FLORENCIA. (14 de Abril de 2009). Obtenido de <http://www.definicionabc.com/general/mapa.php>
- G., B., J., R., & I., J. (1999). El Lenguaje Unificado de Modelado. G. Booch, J. Rumbaugh, I. Jacobson.
- Gopegui, B. (2009). La Escala De Los Mapas - 3ª Ed. Anagrama.
- Inc, S. (2005). *Visual Paradigm*. Recuperado el 20 de Febrero de 2011, de <http://www.slideshare.net/vanquishdarkenigma/visual-paradigm-for-uml>.
- Kendall, K., & Kendall, j. (2005). Análisis y diseño de sistemas. México: Atlacomulco.
- Kevin Sánchez, Junio 2004. Bases de la Ingeniería. Análisis y Diseño.
- Larman, C. (1999). UML y patrones. Introducción al análisis y diseño orientado a objetos. P.Hall.
- Laman, C. (2004). *Appling UML and Patterns*.
- Lobos, M. E. (8 de marzo de 2009). Aprender a programar. Obtenido de <http://www.mailxmail.com/curso-aprende-programar/concepto-lenguaje-programacion>
- Lorenzo Martínez, R. M. (Diciembre 2004). Cartografía. CIE Inversiones Editoriales Dossat-2000, S.L.

- Luciano. Entorno de Desarrollo Integrado para Java. 2005. <http://luauf.com/2008/05/13/entornos-de-desarrollo-integrado-para-java/>
- Montesinos, M. L., & Sanz, J. G. 2003. Panorama actual del ecosistema de software libre para SIG. Valencia: Prodevelop SL, Conde Salvatierra de Ávila .
- Palazón, J. A., & Quiñonero, J. M. (2007). Sistemas de Información Geográfica: mapas, variables y leyendas. Murcia: TAYGA.
- Paradigma visual para UML. (5 de Marzo de 2007). Recuperado el 15 de Enero de 2011, de http://www.freedownloadmanager.org/es/downloads/Paradigma_Visual_para_UML_%5Bcuenta_de_Plataforma_de_Java_14715_p/
- Peralta, Manuel. Sistema de Información. 2008. <http://www.monografias.com/trabajos7/sisinf/sisinf.shtml>
- Pozo, Pau Serra del. «Cinco servidores de mapas.» Revista internacionl de ciencias de la tierra, 2002.
- Pressman, R. .. (2001). Ingeniería de Software.
- Pressman, R. (2005). Ingeniería de Software.
- Ramos, M. J. (2006). Desarrollo aplicaciones entorno 4ª generacion herramientas CASE . McGraw-Hill / Interamericana de España, S.A.
- Santiago, C. (2002). Casos de Uso: Un método Práctico para Explorar Requisitos .
- Sayaz, Carlos Álvarez de. «Metodología de la Investigación Científica.» Santiago de Cuba, 1995.
- Silva, Dr. José Luis Batista. «Aplicación de Sistemas de Información Geográfica en Cuba .» 2005.
- Sys, Ram. ¿Qué es un framework? 29 de Septirmbre de 2006. http://www.mappinginteractivo.com/plantilla-ante.asp?id_articulo=733.
- Torre, Anival de la. Lenguajes de programacin. 2006. http://www.adelat.org/media/docum/nuke_publico/lenguajes_del_lado_servidor_o_cliente.html.
- Tramullas, Jesus, y Kronos. Los sistemas de bases de datos y los SGBD. 2000. <http://tramullas.com/documatica/2-4.html>.

Bibliografías consultadas:

- A, Ernesto Quiñones. Postgre. 2006. http://postgresql.org.pe/articles/introduccion_a_postgresql.pdf.
- Antonio, José. «SIG-ESAC: Sistema de Información Geográfica para la gestión de estadística de Cuba.» Revista Cubana de Higiene y Epidemiología, 2006.
- Bosque, Sandra J. «Sistemas de Información Geográfica.» 451. Rialp, 1997.
- Bravo. Componentes SIG. 2009. http://www.google.com/cu/images?hl=es&biw=1024&bih=592&q=componentes+SIG&um=1&ie=UTF-8&source=univ&ei=qF76TK-VMpvknQeWhrDJCG&sa=X&oi=image_result_group&ct=title&resnum=2&ved=0CC4QsAQwAQ.
- Bravo, Javier Dominguez. Breve introducción a la Cartografía y a los Sistemas de Información Geográfica (SIG) . CIEMAT, 2008.
- Buschmann, F. (1996). Pattern – Oriented Software Architecture. A System of Patterns. Inglaterra.
- Carmona, Alvaro de J. Sistemas de Información Geográfica. 2005. http://www.mappinginteractivo.com/plantilla-ante.asp?id_articulo=733.
- Cerda, F. (2007). Obtenido de http://www.techblog.com/talks/netbeans65es_cl.pdf
- Comas, David. Fundamento de los Sistemas de Información Geográfica. Ariel Geografía, 2007.
- Cooper Bowen, R., & Coar, K. (2004). *Apache Práctico*. Estudio Rojas.
- Fariñas, Lic. Joceta Primelles. «SIG para asistir la estrategia de conservación de las Áreas Fawcett, D., & Butler, H. (25 de Noviembre de 2010). Obtenido de <http://nyksadsi.blogspot.com/2010/08/mapserver.html>
- FELICÍSIMO, A. M. (2003). Glosario de términos usados en el trabajo con SIG.
- FLORENCIA. (14 de Abril de 2009). Obtenido de <http://www.definicionabc.com/general/mapa.php>
- G., B., J., R., & I., J. (1999). El Lenguaje Unificado de Modelado. G. Booch, J. Rumbaugh, I. Jacobson.
- Gopegui, B. (2009). La Escala De Los Mapas - 3ª Ed. Anagrama.
- Guevara, Roberto Ticono. Definición y algunas aplicaciones de los SIG. 2005. <http://www.monografias.com/trabajos14/informageogra/informageogra.shtml>.
- Hernández, Yoana. Geosystems. 2010. <http://www.geosystems.cc/?mod=categoriaProducto&codigo=sig>.

- Inc, S. (2005). Visual Paradigm. Recuperado el 20 de Febrero de 2011, de <http://www.slideshare.net/vanquishdarkenigma/visual-paradigm-for-uml>.
- Kendall, K., & Kendall, J. (2005). Análisis y diseño de sistemas. México: Atlacomulco.
- Kevin Sánchez, Junio 2004. Bases de la Ingeniería. Análisis y Diseño.
- Larman, C. (1999). UML y patrones. Introducción al análisis y diseño orientado a objetos. P.Hall.
- Laman, C. (2004). Applying UML and Patterns.
- Leal, Ing. Yosleny Galvez. Tabaco en Cuba. SIG. 2007. <http://www.monografias.com/trabajos58/sig-tabacalera-cubana/sig-tabacalera-cubana2.shtml>
- Lobos, M. E. (8 de marzo de 2009). Aprender a programar. Obtenido de <http://www.mailxmail.com/curso-aprende-programar/concepto-lenguaje-programacion>
- Lorenzo Martínez, R. M. (Diciembre 2004). Cartografía. CIE Inversiones Editoriales Dossat-2000, S.L.
- Luciano. Entorno de Desarrollo Integrado para Java. 2005. <http://luauf.com/2008/05/13/entornos-de-desarrollo-integrado-para-java/>.
- Luján. «Geografía y Sistemas de información Geográfica (GeoSIG).» Revista digital del Grupo de Estudios sobre Geografía y Análisis espacial con Sistemas de Información Geográfica, 2009.
- Montesinos, M. L., & Sanz, J. G. 2003. Panorama actual del ecosistema de software libre para SIG. Valencia: Prodevelop SL, Conde Salvatierra de Ávila.
- Palazón, J. A., & Quiñonero, J. M. (2007). Sistemas de Información Geográfica: mapas, variables y leyendas. Murcia: TayGA.
- Paradigma visual para UML. (5 de Marzo de 2007). Recuperado el 15 de Enero de 2011, de http://www.freedownloadmanager.org/es/downloads/Paradigma_Visual_para_UML_%5Bcuenta_de_Plataforma_de_Java_14715_p/
- Peralta, Manuel. Sistema de Información. 2008. <http://www.monografias.com/trabajos7/sisinf/sisinf.shtml> --- Manuel Peralta 2008.
- Pressman, R. .. (2001). Ingeniería de Software.
- Pressman, R. (2005). Ingeniería de Software.
- Protegidas de la provincia de Camaguey.» Revista electrónica de la Agencia de Medio Ambiente, 2002.

- Ramos, M. J. (2006). Desarrollo aplicaciones entorno 4ª generación herramientas CASE . McGraw-Hill / Interamericana de España, S.A.
- Rozo, Pau Serra del. «Cinco servidores de mapas.» Revista internacional de ciencias de la tierra, 2002.
- Santiago, C. (2002). Casos de Uso: Un método Práctico para Explorar Requisitos.
- Sayaz, Carlos Álvarez de. «Metodología de la Investigación Científica.» Santiago de Cuba, 1995.
- Silva, Dr. José Luis Batista. «Aplicación de Sistemas de Información Geográfica en Cuba .» 2005.
- Sys, Ram. MapServer en Geomoose. 2010.
http://foro.gabrielortiz.com/pop_printer_friendly.asp?TOPIC_ID=25252.
- Sys, Ram. ¿Qué es un framework? 29 de Septiembre de 2006.
http://www.mappinginteractivo.com/plantilla-ante.asp?id_articulo=733.
- Torre, Anival de la. Lenguajes de programación. 2006.
http://www.adelat.org/media/docum/nuke_publico/lenguajes_del_lado_servidor_o_cliente.html.
- Tramullas, Jesus, y Kronos. Los sistemas de bases de datos y los SGBD. 2000.
<http://tramullas.com/documatica/2-4.html>.