

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

**Título:** “Sistema de Información Geográfica para el control de los recursos  
hidráulicos cubanos”.

**Autor:** Osvaldo Garcia Gorgoy

**Tutor:** Ing. Pedro José Pérez González.

***Dedico especialmente este trabajo a mis padres que tanto se han esforzado por mí, a mi hermana por ser mi gran amiga, a mi esposa que ha estado a mi lado en todo momento y a mi futuro bebé que me inspira a seguir adelante.***

## **Agradecimientos**

A mis padres, por haber estado presentes en todas mis decisiones, por apoyarme y esforzarse tanto para que llegara hasta aquí.

A mi hermana por ser esa gran amiga con la que puedo contar en todo momento, por aconsejarme y animarme a seguir adelante.

A mi abuela Tita y a la tía Cuca, por ser tan buenas y preocuparse tanto por mí.

A mi sobrino que aunque es muy pequeño aún, su cariño ha significado muchísimo.

A mi esposa por estar conmigo a lo largo de estos 5 años, por ayudarme a hacer posible este momento tan importante de mi vida, y sobre todo por darme la alegría de esperar mi primer hijo.

A Gisel por estar muy preocupado siempre por mis problemas y por querer mucho a mi hermana.

A mis suegros y a mi cuñada, por ser tan buenos y cariñosos conmigo, por tratarme como un hijo y por hacerme parte de su familia.

A mis amigos David y Hi, por haber estado junto a mí en la mayor parte de la carrera.

A mi tutor Pedro, por ayudarme en todo y por estar siempre dispuesto.

A toda mi familia en general que de una forma u otra, todos han aportado un granito de arena para hacer este momento realidad.

Muchas gracias además a todas aquellas personas que me han apoyado para llegar hasta aquí.

## **Declaración de autoría**

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 \_\_\_\_ días del mes de \_\_\_\_\_ del año 2011.

Oswaldo Garcia Gorgoy

---

Firma del Autor

Pedro José Pérez González

---

Firma del Tutor

## **Datos de contacto**

**Síntesis del Autor:** Osvaldo Garcia Gorgoy

**Correo Electrónico:** ogorgoy@estudiantes.uci.cu

**Síntesis del Tutor:** Ing. Pedro José Pérez González

**Profesión:** Ingeniero en Ciencias Informáticas

**Años de graduado:** Adiestrado

**Correo Electrónico:** pjgonzalez@uci.cu

## Resumen

El Instituto Nacional de Recursos Hidráulicos (INRH) es el organismo encargado de dirigir, ejecutar y controlar la aplicación de la política del Estado y el Gobierno en cuanto a las actividades de los recursos hidráulicos del país. En la actualidad dicho instituto no cuenta con un sistema que le permita georreferenciar los recursos hidráulicos del país. Para solucionar este problema se pretende realizar un sistema que cubra las necesidades del instituto.

El presente trabajo que lleva por título: Sistema de Información Geográfica para el control de los recursos hidráulicos cubanos, tiene como objetivo desarrollar un Sistema de Información Geográfica (SIG), así como generar la documentación técnica asociada al mismo, definiendo las principales librerías, frameworks y herramientas utilizadas. Con el desarrollo del mismo se alcanzaron los resultados esperados, proporcionando una forma mucho más eficaz de visualizar los recursos hidráulicos y mostrar los datos relacionados con los mismos. El sistema disminuye el tiempo y el esfuerzo necesario para realizar cambios en la visualización de los mapas, así como la cantidad de recursos y personal involucrados.

**Palabras claves:** Sistema de Información Geográfica (SIG), Instituto Nacional de recursos hidráulicos (INRH).

# Índice

Resumen.....	5
Introducción .....	11
Capítulo 1: Elementos Teóricos de la investigación. ....	15
1.1 Introducción.....	15
1.2 Conceptos asociados al dominio del problema .....	15
1.3 Objeto de Estudio .....	16
1.3.1 Sistemas de Información Geográfica (SIG) .....	16
1.3.2 Descripción del dominio del problema .....	18
1.4 Análisis de otras soluciones existentes .....	18
1.5 Sistema web.....	19
1.5.1 Características de los Sistemas Web.....	19
1.5.2 Modelo Cliente – Servidor .....	20
1.6 Conclusiones parciales .....	21
Capítulo 2: Selección y descripción de herramientas, metodologías y lenguajes. ....	22
2.1 Introducción.....	22
2.2 Metodología de Desarrollo de Software .....	22
2.2.1 Proceso Unificado de Rational (RUP).....	22
2.2.2 Metodología ágil para el desarrollo de software eXtreme Programming (XP) .....	23
2.2.3 Metodología ágil de desarrollo SCRUM.....	23
2.2.4 Selección de la metodología a utilizar .....	24
2.3. Tecnologías web del lado del servidor .....	24
2.3.1 PHP.....	24
2.3.2 Python .....	26
2.3.3 Selección del lenguaje de programación para el servidor .....	27
2.3.4 Estándares de Codificación.....	27
2.3.5 Patrón Modelo – Vista – Controlador (MVC) .....	28

2.4 Tecnologías web del lado del cliente .....	29
2.4.1 ExtJS3.3 .....	29
2.4.2 OpenLayer .....	31
2.4.3 GeoExt .....	31
2.4.4 Estándares de Codificación .....	32
2.5 Servidores .....	32
2.5.1 Servidor Web Apache .....	32
2.5.2 Servidor web Internet Information Services (IIS) .....	33
2.5.3 Selección del servidor web .....	33
2.5.4 Servidor de Base de Datos PostgreSQL .....	34
2.5.5 MySQL .....	36
2.5.6 Oracle .....	37
2.5.7 Selección del servidor de base de datos a utilizar .....	37
2.6 Herramientas de Desarrollo .....	37
2.6.1 Visual Paradigm .....	38
2.6.2 Erwin .....	39
2.6.3 Rational Rose .....	39
2.6.4 Selección de la herramienta a utilizar .....	39
2.6.5 Zend Studio .....	39
2.7 Conclusiones Parciales .....	40
Capítulo 3: Análisis de la solución propuesta .....	41
3.1 Introducción .....	41
3.2 Descripción del Modelo de Dominio .....	41
3.3 Especificación de los requisitos de la aplicación .....	41
3.3.1 Requisitos Funcionales .....	42
3.3.2 Requisitos no funcionales .....	43
3.4 Definición de los casos de usos del sistema .....	44
3.4.1 Actores del sistema .....	44
3.4.2 Diagrama de casos de uso del sistema .....	45



3.4.3 Especificación de los casos de uso del sistema.....	45
3.5 Conclusiones parciales .....	55
Capítulo 4: Desarrollo de la solución y validación. ....	56
4.1 Introducción .....	56
4.2 Patrones de diseño .....	56
4.2 Diagramas de clases de diseño.....	56
4.3 Diseño de la Base de Datos.....	57
4.3.1 Diagrama Entidad-Relación.....	58
4.4 Modelo de despliegue .....	58
4.5 Diagrama de componentes .....	58
4.6 Pruebas funcionales .....	60
4.6.1 Diseño de las pruebas que permiten validar la solución propuesta .....	61
4.6.2 Pruebas de caja negra.....	67
4.7 Conclusiones parciales .....	68
Conclusiones Generales .....	69
Bibliografía Referenciada .....	71

## Índice de Tablas y Figuras:

Figura 1. Representación de los modelos de datos raster y vectorial .....	17
Figura 2. Modelo Cliente-Servidor.....	20
Figura 3. Roles Scrum.....	24
Figura 4. Patrón MVC .....	29
Figura 5. Servidores web más utilizados .....	34
Figura 6. Modelo de dominio. ....	41
Figura 7. Diagrama de Caso de Uso del Sistema .....	45
Figura 8. Diagrama de clases de diseño CU_Personalizar mapa.....	57
Figura 9. Diagrama Entidad-Relación .....	58
Figura 10. Diagrama de despliegue del sistema .....	58
Figura 11. Diagrama de componentes CU Personalizar mapa.....	59
Figura 12. Diagrama de componentes CU Mostrar Datos .....	59
Figura 13. Diagrama de componentes CU Acercar .....	60
Figura 14. Diagrama de componentes CU Recentrar mapa .....	60
Figura 15. Interfaz Gráfica Localizar por coordenadas. ....	67
Tabla 1. Descripción total del actor del sistema .....	45
Tabla 2. CUS Realizar Navegación .....	46
Tabla 3. CUS Personalizar mapa.....	49
Tabla 4. CUS Localizar Objetivo .....	50
Tabla 5. CUS Realizar Control de Selección .....	51
Tabla 6. CUS Localizar por Coordenadas .....	52
Tabla 7. CUS Medir Distancia .....	53
Tabla 8. Calcular área .....	54
Tabla 9. Secciones a probar en el caso de uso Realizar Navegación .....	61
Tabla 10. Acercar .....	62
Tabla 11. Alejar.....	62
Tabla 12. Recentrar mapa.....	63
Tabla 13. Mover mapa.....	63
Tabla 14. Secciones a probar en el caso de uso Localizar por Coordenadas.....	64
Tabla 15. Variables a probar .....	64
Tabla 16. Localizar por Coordenadas completo .....	65
Tabla 17. Secciones a probar en el caso de uso Medir distancia .....	65
Tabla 18. Medir distancia .....	66

Tabla 19. Calcular área .....	66
Tabla 20. Caso de prueba # 1 utilizando técnicas de caja negra.....	67
Tabla 21. Caso de prueba # 2 utilizando técnicas de caja negra.....	68
Tabla 22. Caso de prueba # 3 utilizando técnicas de caja negra.....	68

## Introducción

Con el transcurso de los años las Tecnologías de la Información y las Comunicaciones (TICs) se han desarrollado a medida que estas se han puesto al servicio de las crecientes necesidades sociales. Sin duda alguna, un sistema automatizado de los que existen actualmente con uno desarrollado años atrás, tienen una diferencia notable en cuanto a diseño, rapidez, eficacia y en la interfaz que permite tantas comodidades a los clientes de dichos sistemas.

En el mundo actual la sociedad ha desarrollado una estrecha relación con las nuevas tecnologías, esto asociado a elevar el nivel de vida, pero para lograr cubrir todas estas demandas se hace necesaria la búsqueda de nuevas soluciones. Apareciendo así los Sistemas de Información Geográfica, los que han dado una nueva perspectiva y se han convertido en una de las herramientas más aceptadas y utilizadas no solo por la población sino también por investigadores, analistas y planificadores. La gran aceptación de estos sistemas está dada porque la información con la que trabajan facilita la toma de decisiones en muchas áreas vitales para el desarrollo económico y social.

Un Sistema de Información Geográfica (SIG) no es más que: un conjunto de métodos, herramientas y datos que están diseñados para actuar coordinada y lógicamente para capturar, almacenar, analizar, transformar y presentar toda la información geográfica y de sus atributos con el fin de satisfacer múltiples propósitos. Los SIG son una tecnología que permite gestionar y analizar la información espacial, y que surgió como resultado de la necesidad de disponer rápidamente de información para resolver problemas y contestar a preguntas de modo inmediato (Brenes, 2008).

Un SIG es un sistema dinámico que permite seleccionar y eliminar cualquier criterio para mapear y analizar rápidamente cómo diferentes factores afectan a un modelo o análisis.

Cuba no se encuentra ajena al desarrollo de estos sistemas que a nivel mundial han dado prueba de sus utilidades. En la Universidad de las Ciencias Informáticas (UCI), específicamente en el departamento de Geoinformática de la Facultad 6, se desarrollan diferentes sistemas de información geográfica para todas las empresas y entidades que lo necesiten. A la misma se le ha planteado que el Instituto Nacional de Recursos Hidráulicos (INRH) cubano, presenta dificultades a la hora de mostrar sobre un mapa los recursos hidráulicos cubanos así como los datos específicos de cada uno

de ellos, confeccionar mapas temáticos partiendo de diferentes criterios de comparación, realizar cálculos de área y distancia sobre el terreno. Además una vez que los datos han sido dibujados sobre un mapa le resulta al personal del INRH muy difícil realizar el acercamiento de una zona determinada, así como el manejo y manipulación del mismo, debido a que la confección de un mapa tiene un alto costo tanto en esfuerzo como en tiempo dedicado.

De lo antes planteado se identifica como **problema a resolver**: ¿cómo mejorar la visualización de los datos de los recursos hidráulicos cubanos manejados por el INRH?

El **objeto de estudio** lo constituye el proceso de visualización de los datos de los recursos hidráulicos cubanos y el **campo de acción**, la automatización de la visualización de los datos de los recursos hidráulicos controlados por el INRH.

Como **objetivo general** se plantea desarrollar un Sistema de Información Geográfica que permita georreferenciar y visualizar datos de los recursos hidráulicos existentes en Cuba.

Como **idea a defender** se plantea que, el desarrollo de un Sistema de Información Geográfica permite mejorar la visualización de los datos de los recursos hidráulicos controlados por el INRH.

Para darle cumplimiento al objetivo general se plantean las siguientes **tareas de la investigación**:

- Caracterizar los procesos de visualización de los recursos hidráulicos controlados por el INRH.
- Seleccionar la Metodología de Desarrollo de Software a usar en el proceso.
- Identificar las principales actividades del Sistema de Información Geográfica para el control de los recursos hidráulicos cubanos.
- Diseñar el Sistema de Información Geográfica para el control de los recursos hidráulicos cubanos.
- Implementar el Sistema de Información Geográfica para el control de los recursos hidráulicos cubanos.
- Validar el Sistema de Información Geográfica para el control de los recursos hidráulicos cubanos.

Para garantizar la correcta solución del problema planteado se utilizaron los siguientes métodos teóricos de la investigación:

- **Analítico – Sintético:** Para llegar a conclusiones a partir de la documentación consultada, para entender fenómenos relacionados con los recursos hidráulicos cubanos y su representación visual.
- **Histórico – Lógico:** Para conocer las tendencias actuales en cuanto a la gestión de la información geográfica de los recursos hidráulicos cubanos.
- **Modelación:** Se modeló el proceso de diseño e implementación del sistema.

Dentro de los métodos empíricos se utilizó exactamente el de:

- **Observación:** Se utilizó en la investigación inicialmente para el diagnóstico del problema a investigar y fue de gran utilidad en el diseño de la investigación.

El trabajo de diploma cuyos posibles resultados son: el documento general del proceso de desarrollo y el Sistema de Información Geográfica para el control de los recursos hidráulicos cubanos, se estructura en cuatro capítulos que son descritos a continuación:

**Capítulo 1.** Elementos teóricos de la investigación: Contempla la fundamentación teórica de la investigación, donde se exponen conceptos que permitirán familiarizarse con términos utilizados en el tema en cuestión tales como: SIG, Georreferenciación, Información Geográfica. Además se especifican argumentos que muestran claramente el objeto de estudio y se mencionan otras soluciones existentes.

**Capítulo 2.** Selección y descripción de herramientas y metodologías: En este capítulo se muestran las características fundamentales de las principales tecnologías, herramientas y lenguajes que se utilizan en el mundo para la realización de este tipo de sistemas entre las que se destacan RUP, Visual Paradigm, Open Layer, Extjs y PHP. Se realizará también la elección de las que se utilizarán para la construcción de la solución propuesta.

**Capítulo 3.** Análisis y Diseño del sistema: Los requisitos funcionales y no funcionales del sistema, los casos de uso derivados de ellos y su descripción serán los temas abordados en este capítulo. Además se representa el modelo de dominio y el diagrama de caso de uso correspondiente al sistema.

**Capítulo 4.** Implementación y Prueba: Contempla los aspectos fundamentales sobre la construcción del sistema propuesta en el capítulo anterior, reflejando los principales

componentes y diagramas usados, además de la descripción de las pruebas realizadas al mismo.

# Capítulo 1: Elementos Teóricos de la investigación.

## 1.1 Introducción

En el presente capítulo se abordan los temas relacionados con la fundamentación teórica de la investigación. Se describen los principales conceptos que permiten un mayor entendimiento del dominio del problema, así como del objeto de estudio, analizando la situación problemática y las soluciones existentes tanto en Cuba como en el resto del mundo.

## 1.2 Conceptos asociados al dominio del problema

**Información geográfica:** Se denomina Información Geográfica (IG) a aquellos datos espaciales georreferenciados requeridos como parte de las operaciones científicas, administrativas o legales. Dichos datos espaciales suelen llevar una información alfanumérica asociada. Se estima que el 80% de los datos corporativos existentes en todo el mundo poseen este componente geográfico (Yagüez, 2002).

**Sistema de información:** Es un conjunto de elementos que interactúan entre sí con el fin de apoyar las actividades de una empresa o negocio (Burch, 1992).

En un SIG la información puede ser almacenada en formato raster o vectorial:

El modelo de SIG **raster** 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. Cuanto mayor sean las dimensiones de las celdas (resolución) menor es la precisión o detalle en la representación del espacio geográfico (Yagüez, 2002).

En el caso del modelo de SIG **vectorial**, el interés de las representaciones se centra en la precisión de localización de los elementos sobre el espacio y donde los fenómenos a representar son discretos, es decir, de límites definidos (Yagüez, 2002).

**Georreferenciación:** Es el posicionamiento en el que se define la localización de un objeto espacial en un sistema de coordenadas. Este proceso es utilizado frecuentemente en los Sistemas de Información Geográfica (Yagüez, 2002).

Además de estos conceptos antes mencionados es necesario plantear definiciones asociadas al tema:



**Recursos Hidráulicos:** Los recursos hidráulicos incluyen a los ríos, arroyos, lagos y lagunas, presas, así como los almacenamientos subterráneos y las grandes masas oceánicas (Puyol, 2004).

**Presa:** Construcción con la que se retiene y almacena el agua en una región para conservarla y repartirla durante todo el año en las tierras que riega, y en muchas de ellas para generar electricidad; generalmente está compuesta por un muro alto, construido en el cauce de un río o al final de un cauce artificial, y unas compuertas que permiten el paso regulado del agua (Florencia, 2010).

Como otra definición se puede plantear que las presas son una barrera fabricada con piedra, hormigón o materiales sueltos, que se construye habitualmente en una cerrada o desfiladero sobre un río o arroyo con la finalidad de embalsar el agua en el cauce fluvial para su posterior aprovechamiento en abastecimiento o regadío, para elevar su nivel con el objetivo de derivarla a canalizaciones de riego, o para la producción de energía mecánica al transformar la energía potencial del almacenamiento en energía cinética, y ésta nuevamente en mecánica al accionar la fuerza del agua un elemento móvil (Puyol, 2004).

### **1.3 Objeto de Estudio**

Este trabajo de diploma se basa en el objeto de estudio los procesos de representación de los recursos hidráulicos cubanos. Para una mejor comprensión del mismo se hace necesario que se comience realizando una breve panorámica de lo que son los SIG de forma general y de forma particular.

#### **1.3.1 Sistemas de Información Geográfica (SIG)**

El desarrollo que la tecnología ha alcanzado y el gran valor de la información, exigen una mayor eficacia a la hora de tomar decisiones. En varias ramas de la sociedad los SIG se han convertido en una herramienta primordial para investigadores y especialistas, ya que están enfocados a visualizar geográficamente información o datos particulares de cada sistema. La popularidad de estos y su aceptación viene dada en gran medida por:

- Por ser un poderoso conjunto de herramientas para coleccionar, almacenar, recuperar, transformar y exhibir datos espaciales referenciados al mundo real (Burrough, 1986).

- Además se puede afirmar que es un sistema de hardware, software y procedimientos diseñados para realizar la captura, almacenamiento, manipulación, análisis, modelización y presentación de datos referenciados espacialmente para la resolución de problemas complejos de planificación y gestión (NCGIA, 1990).

## Componentes de un SIG

La composición de los SIG está dividida en cinco partes que son fundamentales, el Hardware, el Software, la Información, el Personal capacitado y los métodos de desarrollo.

Los componentes principales del software SIG son:

- Sistema de manejo de base de datos.
- Una Interfaz Gráfica de Usuarios (IGU) para el fácil acceso a las herramientas.
- Herramientas para captura y manejo de información geográfica.
- Herramientas para soporte de consultas, análisis y visualización de datos geográficos.

## Funcionamiento de los SIG

Los SIG funcionan con dos tipos diferentes de información geográfica: el modelo vector y el modelo raster como se muestra en la figura 1.

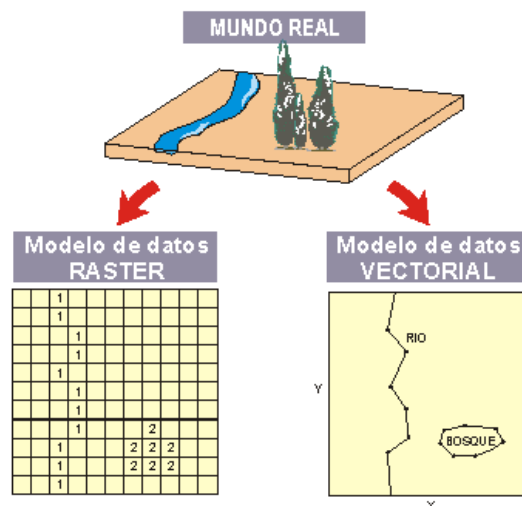


Figura 1. Representación de los modelos de datos raster y vectorial (Potencier, 2010)

Una imagen raster comprende una colección de celdas (píxel) de una grilla más como un mapa o una figura escaneada. Por otra parte en el modelo vector la información sobre puntos, líneas y polígonos se almacena como una colección de coordenadas x,y.

### **1.3.2 Descripción del dominio del problema**

En estos momentos el INRH presenta grandes dificultades a la hora de tratar todos los datos relacionados con los recursos hidráulicos, existe una gran deficiencia cuando se trata de mostrar los datos de dichos recursos en un mapa. La demora al tratar de obtener información actualizada sobre el estado de los ríos y embalses existentes en el país, afecta en la temprana y acertada toma de decisiones, esto unido con el esfuerzo necesario para realizar cálculos simples basados en la distancia y el estado actual de los niveles de los embalses, además de no contar con una forma real de visualización de los mismos teniendo en cuenta su capacidad en una escala de colores para una mejor interpretación de tan necesaria información.

El desarrollo de cambios en los recursos hidráulicos trae consigo semanas de observaciones de campo y dibujo de mapas, esto unido a la implicación de tiempo de trabajo de varios técnicos y especialistas, que en varias ocasiones al culminar su trabajo, que tiene un costo elevado tanto en recursos como en material humano, ya los datos representados son inexactos. Monitoreo de cambios es una de las situaciones más difíciles de enfrentar para los planificadores del instituto, mostrar los cambios que han surgido hace cinco años o un poco más para el análisis y la creación de políticas sobre el desarrollo socio-económico de áreas afectadas por estos cambios, puede ser tomar meses de interpretación de mapas y cálculos de probabilidad, tiempo que siempre es muy valioso cuando se habla de tomas de decisiones. Todas estas dificultades existentes en estos momentos, particularmente el alto costo y la gran implicación de tiempo que se necesita para cualquier situación no permiten el buen aprovechamiento de las tecnologías en esta rama de la economía que es fundamental para el desarrollo del país.

### **1.4 Análisis de otras soluciones existentes**

**Google Maps:** Es un servicio de mapas gratuito, al que puedes acceder por la web cualquier usuario que posea conexión a internet y posee varias funcionalidades entre las que sobresalen:

- Acercar y Alejar el mapa.
- Medición de distancia.

- Muestra el nivel de tráfico en la zona seleccionada
- Muestra la latitud y la longitud.

**Google Earth:** Es una aplicación de escritorio, que funciona conectándose a los servidores de Google y muestra mapas de todo el mundo, presenta varias funciones que son de gran utilidad para los usuarios:

- Acercar y Alejar el mapa.
- Buscar por nombre.
- Viaje por ruta.
- Lugares de interés.

**Dices:** Es un servicio de mapa de España que se puede acceder por la web y muestra a los usuarios información sobre detallada sobre el tema que se escoja y brinda utilidades como:

- Acercar y Alejar el mapa
- Muestra las coordenadas
- Búsqueda por provincias

**SIGUCI:** Es un servicio de mapas sobre una plataforma web que muestra información geográfica de la Universidad de las Ciencias Informáticas en Cuba, permitiéndole a sus usuarios:

- Acercar y Alejar el mapa.
- Localizar geográficamente a una persona.
- Muestra los lugares de interés.
- Medir distancia y calcular área.

## 1.5 Sistema web

Es un conjunto de páginas web estáticas o dinámicas, entiéndase por página web estática a aquella en que es mostrada siempre al cliente de la misma forma sin importar las veces que sea llamada y por página web dinámica a la página cuyo contenido cambia en dependencia de los permisos de acceso que tenga el usuario que la solicita, de la cantidad de veces que sea llamada por el usuario (Connalen, 1999).

### 1.5.1 Características de los Sistemas Web

Las características principales de una aplicación web son: (Sánchez, et al., 2007).

- Se encuentra alojada en un Servidor Web.
- Son accesibles mediante el internet, usando un navegador web.
- La lógica del sistema se ejecuta en el servidor, mientras que el cliente solamente representa los datos.
- El acceso al sistema puede ser público o restringido.
- La actualización del sistema no afecta ni depende del cliente.
- Los sistemas son multiplataforma, ya que pueden ejecutarse en cualquier Sistema Operativo que tenga un servidor web o un navegador web.

### 1.5.2 Modelo Cliente – Servidor

El uso de las aplicaciones que se encuentran publicadas en la Internet, es llamado modelo Cliente – Servidor. Cuando se hace uso de cualquier servicio que se encuentre en la Internet, ya sea visitar un sitio, consultar una base de datos, realizar una transferencia de ficheros; se establece un proceso en el cual intervienen dos integrantes. El cliente o usuario que ejecuta una aplicación en el ordenador local, el que se denomina programa cliente. Este cliente se encarga de comunicarse con el ordenador remoto para solicitar el servicio. El ordenador remoto atiende la solicitud del cliente mediante un programa que se denomina programa servidor. Los términos Cliente – Servidor son usados para referirse a los programas que cumplen con estas características (Sánchez, et al., 2007).

El modelo Cliente – Servidor (Figura 2) facilita la integración y comunicación entre diferentes sistemas. De esta manera se puede usar sistemas que usen máquinas con diferentes sistemas operativos o sistemas medianos o grandes (Alfaro, 2005).

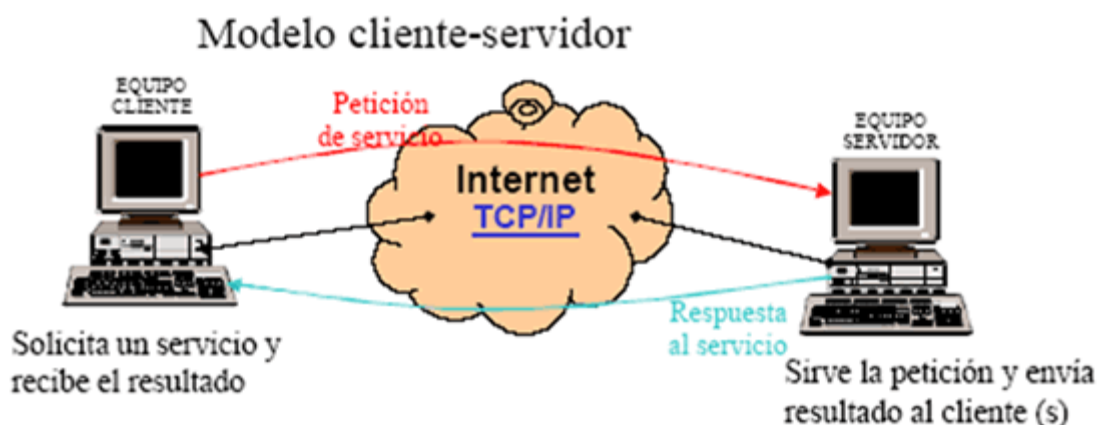


Figura 2. Modelo Cliente-Servidor (Alfaro, 2005)

## **1.6 Conclusiones parciales**

Como conclusión de este capítulo se puede plantear que luego de expuesto todo un conjunto de conceptos y analizados algunos sistemas similares que han permitido una mayor comprensión del problema en cuestión, se determino que se debía desarrollar un sistema web para disminuir el tiempo que se necesita actualmente para la actualización de los mapas y demás datos relacionados, asegurando de este modo una reducción del costo y el esfuerzo necesarios para llevar a cabo estas tareas.

# Capítulo 2: Selección y descripción de herramientas, metodologías y lenguajes.

## 2.1. Introducción

En este capítulo se describen las principales metodologías de desarrollo de software, las tecnologías web a usar del lado del servidor y del lado del cliente, además se mencionan los servidores utilizados y las herramientas de desarrollo que se utilizarán para la implementación del sistema.

## 2.2 Metodología de Desarrollo de Software

Las metodologías de desarrollo de software son utilizadas para estructurar, planear y controlar el proceso de desarrollo de software, estas ayudan a no cometer errores que otros en su momento han tenido, dando una mayor probabilidad de triunfo al complejo proceso de desarrollo.

### 2.2.1. Proceso Unificado de Rational (RUP)

RUP constituye la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas, tiene como objetivo garantizar un desarrollo eficiente y robusto, su uso brinda un gran apoyo si se quiere lograr un proceso eficiente, dirigido por casos de usos, centrado en la arquitectura, iterativo e incremental.

RUP es un proceso que define quién está haciendo, qué está haciendo, cuando lo hace y como lo hace, utiliza para ello el Lenguaje de Modelado Unificado (UML), para crear los esquemas del sistema de software (Jacobson, 2004).

**Roles según RUP:** (Process®, 2006).

- **Diseñador:** Identifica y define las responsabilidades, operaciones, atributos y relaciones de los elementos de diseño.
- **Diseñador de base de datos:** Es el responsable de definir el diseño detallado de la base de datos.
- **Diseñador de interfaz de usuario:** Este rol coordina el diseño de la interfaz de usuario.
- **Implementador:** Este rol desarrolla los componentes de software y efectúa las pruebas de desarrollador para la integración en subsistemas más grandes, de acuerdo con los estándares adoptados de proyecto.
- **Probador:** Este rol realiza pruebas y registra los resultados de las pruebas.

## 2.2.2 Metodología ágil para el desarrollo de software eXtreme Programming (XP)

XP es una metodología ágil centrada en potenciar las relaciones interpersonales como clave para el éxito en el desarrollo de software, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores, y propiciando un buen clima de trabajo.

XP se basa en realimentación continua entre el cliente y el equipo de desarrollo, comunicación fluida entre todos los participantes, simplicidad en las soluciones implementadas y coraje para enfrentar los cambios. XP se define como especialmente adecuada para proyectos con requisitos imprecisos y muy cambiantes, y donde existe un alto riesgo técnico (Penadés, 2004).

**Roles XP:** (Penadés, 2004).

- **Programador:** El programador escribe las pruebas unitarias y produce el código del sistema.
- **Cliente:** El cliente escribe las historias de usuario y las pruebas funcionales para validar su implementación.
- **Encargado de pruebas:** El encargado de pruebas ayuda al cliente a escribir las pruebas funcionales.
- **Encargado de seguimiento:** Su responsabilidad es verificar el grado de acierto entre las estimaciones realizadas y el tiempo real dedicado.
- **Entrenador:** Es responsable del proceso global.
- **Consultor:** Guía al equipo para resolver un problema específico.
- **Gestor:** Su labor esencial es de coordinación.

## 2.2.3 Metodología ágil de desarrollo SCRUM

Scrum es una metodología de desarrollo muy simple, que requiere trabajo duro porque no se basa en el seguimiento de un plan, sino en la adaptación continua a las circunstancias de la evolución del proyecto (Palacio, 2006).

**Los Roles Scrum:** (Palacio, 2006).

Scrum clasifica a todas las personas que intervienen o tienen interés en el desarrollo del proyecto (Ver figura 3).

- **Propietario del producto:** El responsable de obtener el mayor valor de producto para los clientes, usuarios y resto de implicados.



- **Equipo de desarrollo:** Grupo o grupos de trabajo que desarrollan el producto.
- **Encargado de Scrum (*ScrumManager*):** Gestor de los equipos que es responsable del funcionamiento de la metodología Scrum y de la productividad del equipo de desarrollo.

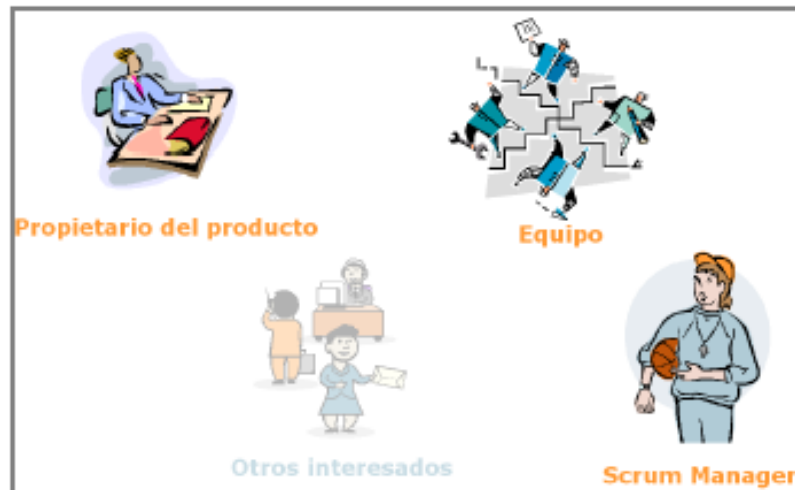


Figura 3. Roles Scrum (Palacio, 2006)

## 2.2.4 Selección de la metodología a utilizar

Las tres metodologías presentan diferentes formas de dirigir el ciclo de vida de un proyecto, para el desarrollo del SIG-INRH se determina RUP como metodología de desarrollo ya que, a diferencia de las metodologías ágiles, se lleva a cabo una documentación detallada de todas las fases, posibilitando un mayor entendimiento por parte del equipo de desarrollo y una gran ventaja a la hora de realizar futuras actualizaciones del producto.

## 2.3. Tecnologías web del lado del servidor

Se puede afirmar que los lenguajes de lado 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.

### 2.3.1. PHP

PHP (acrónimo de PHP: Hypertext Preprocessor). Es un lenguaje interpretado de alto nivel embebido en páginas HTML y ejecutado en el servidor. El PHP inicio como una modificación a Perl escrita por Rasmus Lerdorf a finales de 1994. Su primer uso fue el de mantener un control sobre quien visitaba su currículum en su web (Castillo, 2007).

En los siguientes tres años, se fue convirtiendo en lo que se conoce como PHP/FI 2.0. Esta forma de programar llegó a muchos usuarios, pero el lenguaje no tomó el peso actual hasta que Zeev Surasky y Andi Gutmans le incluyeron nuevas características en 1997, que dio por resultado el PHP 3.0. La versión 5 es la más reciente y popular en este momento debido a que presenta cambios muy beneficiosos para los desarrolladores (Castillo, 2007).

PHP es un lenguaje “Open Source<sup>1</sup>” interpretado de alto nivel, pensado para desarrollar sistemas web, el cual puede ser embebido en páginas HTML. La mayoría de su sintaxis es similar a C, Java y Perl, dándole la ventaja de la facilidad de aprendizaje. La meta de este lenguaje es permitir a los desarrolladores escribir páginas web dinámicas de manera fácil y rápida (PHP Group, 2010).

Ventajas: (Marley, 2011).

- Es un lenguaje multiplataforma.
- Completamente orientado a la web.
- Capacidad de conexión con la mayoría de los motores de base de datos que se utilizan en la actualidad, destaca su conectividad con MySQL y PostgreSQL.
- Capacidad de expandir su potencial utilizando la enorme cantidad de módulos.
- Posee una amplia documentación, entre la cual se destaca que todas las funciones del sistema están explicadas y ejemplificadas en un único archivo de ayuda.
- Permite las técnicas de Programación Orientada a Objetos.
- Biblioteca nativa de funciones sumamente amplia e incluida.
- No requiere definición de tipos de variables aunque sus variables se pueden evaluar también por el tipo que estén manejando en tiempo de ejecución.
- Tiene manejo de excepciones.
- Rapidez. PHP generalmente es utilizado como modulo de Apache, lo que lo hace extremadamente veloz. Esta completamente escrito en C, así que se ejecuta rápidamente utilizando poca memoria.

---

<sup>1</sup>**Open Source:** Término usado para denotar al Código Libre, la característica fundamental es que el código de una aplicación no puede ser privado, sino que todos tienen que tener acceso a ver, analizar o modificar este código.

Desventajas: (Marley, 2011).

Si bien PHP no obliga a quien lo usa a seguir una determinada metodología a la hora de programar, aún estando dirigido a alguna en particular, el programador puede aplicar en su trabajo cualquier técnica de programación y/o desarrollo que le permita escribir código ordenado, estructurado y manejable. Un ejemplo de esto son los desarrollos que en PHP se han hecho del patrón de diseño Modelo Vista Controlador (o MVC), que permiten separar el tratamiento y acceso a los datos, la lógica de control y la interfaz de usuario en tres componentes independientes. La ofuscación de código es la única forma de ocultar las fuentes.

### 2.3.2 Python

Python es un lenguaje de scripting independiente de plataforma y orientado a objetos, preparado para realizar cualquier tipo de programa, desde aplicaciones Windows a servidores de red o incluso, páginas web. Es un lenguaje interpretado, lo que significa que no se necesita compilar el código fuente para poder ejecutarlo, lo que ofrece ventajas como la rapidez de desarrollo e inconvenientes como una menor velocidad (Alvarez, 2007).

**Características del lenguaje:** (Alvarez, 2007).

- **Propósito general:** Se pueden crear todo tipo de programas. No es un lenguaje creado específicamente para la web, aunque entre sus posibilidades se encuentra el desarrollo de páginas.
- **Multiplataforma:** Hay versiones disponibles en muchos sistemas informáticos. Originalmente se desarrolló para Unix, aunque cualquier sistema es compatible con el lenguaje siempre y cuando exista un intérprete programado para él.
- **Interpretado:** No se debe compilar el código antes de su ejecución. En realidad sí que se realiza una compilación, pero esta se realiza de manera transparente para el programador.
- **Interactivo:** Dispone de un intérprete por línea de comandos en el que se pueden introducir sentencias.
- **Orientado a Objetos:** La programación orientada a objetos está soportada en Python y ofrece en muchos casos una manera sencilla de crear programas con componentes reutilizables.
- **Funciones y librerías:** Dispone de muchas funciones incorporadas en el propio lenguaje, para el tratamiento de strings, números, archivos, etc.

Además, existen muchas librerías que se pueden importar en los programas para tratar temas específicos.

- **Sintaxis clara:** Tiene una sintaxis visual, gracias a una notación indentada de obligado cumplimiento, esto ayuda a que todos los programadores adopten unas mismas notaciones y que los programas de cualquier persona tengan un aspecto muy similar.

### 2.3.3 Selección del lenguaje de programación para el servidor

El lenguaje que se utilizará para el desarrollo del lado del servidor será PHP, debido a las ventajas que presenta ya que es un lenguaje desarrollado específicamente para la web, además que brindará una mayor rapidez al sistema por ser utilizado como un módulo de Apache, elementos en los que Python presenta dificultades.

### 2.3.4 Estándares de Codificación

Los estándares de código se definen como las reglas o las directivas que se usan cuando se escribe el código de un programa. Estos estándares ayudan a que otros programadores puedan entender, usar o corregir errores de un fragmento de código escrito por otro programador. Estos patrones son generalmente creados para un lenguaje de programación específico, de forma que un estándar escrito para C++ no pueda ser aplicado a Python, aunque muchas reglas puedan ser aplicadas a distintos lenguajes de programación (Sensio Lab, 2009).

Estándar de código propuesto para el desarrollo del sistema:

- No usar el tab para indentar el código, use 2 espacios en blanco.
- No usar los espacios después de abrir y antes de cerrar los paréntesis.
- Use la notación camello o camelCase para las variables, métodos y funciones, esta notación define que los nombres empiezan por minúscula, en caso de ser un nombre compuesto el primer nombre empieza con minúscula y el segundo seguido empieza con mayúscula.
- Las llaves {} siempre se definen en una línea, úselas para definir el cuerpo de las estructuras de control y ordenar las expresiones definidas en el cuerpo de estas.
- No termine los ficheros .php con la etiqueta de cierre ?>, esto no es necesario y puede provocar errores de difícil detección, en caso de que exista un espacio después de esta etiqueta.

- En el cuerpo de una función la declaración de retorno debe de estar separada por una línea en blanco, de esta forma se aumenta la legibilidad del código.
- Todo comentario que solamente tenga una línea, tiene que ser escrito usando la etiqueta de comentario de línea seguido por un espacio y el comentario en minúscula.
- Evite evaluar variables en conjunto con un string, en lugar de eso use la concatenación.
- Para comprobar si una variable es nula no use la función nativa de php `is_null`, use el operador de comparación exactamente distinto (`!==`).
- Cuando compare una variable con un string, ponga primero el string y después la variable, en caso de poderse usar el operador de exactamente igual (`===`).

### 2.3.5 Patrón Modelo – Vista – Controlador (MVC)

El Modelo-Vista-Controlador es el patrón clásico para el diseño web, el cual está formado por 3 niveles: (Ver figura 4).

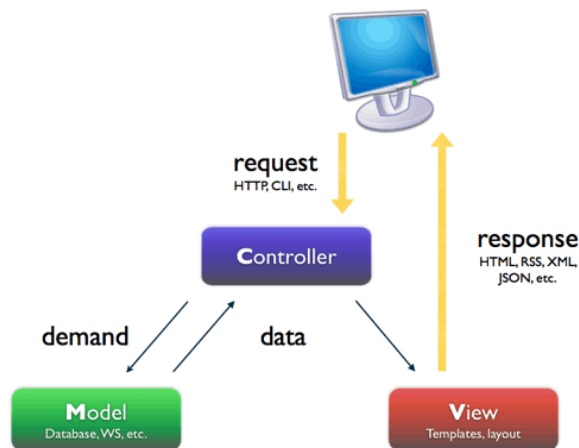
- El modelo representa la información con que trabaja la aplicación, la lógica del negocio.
- La vista transforma el modelo en una página web que permite al usuario interactuar con ella.
- El controlador se encarga de procesar las iteraciones del usuario y realizar los cambios necesarios en el modelo o la vista, también controla la lógica del sistema.

La arquitectura MVC separa la lógica del negocio y la representación de la información, por lo que se consigue un mantenimiento más sencillo de las aplicaciones.

El controlador se encarga de aislar al modelo y a la vista de los detalles del protocolo utilizado por la petición (HTTP<sup>2</sup>, Consola de Comandos, Correo Electrónico). El modelo se encarga de la abstracción de la lógica relacionada con los datos, haciendo que la vista y los datos sea independiente del tipo de gestor de base de datos utilizado en la aplicación (Potencier, et al., 2009).

---

<sup>2</sup>**HTTP**: Hypertext Transfer Protocol o Protocolo de Transferencia de Hipertexto, protocolo usado en la comunicación de los sistemas web mediante el uso de la Internet.



**Figura 4. Patrón MVC (Potencier, y otros, 2009)**

Este modelo de arquitectura presenta varias ventajas: (Potencier, y otros, 2009).

- Hay una clara separación entre los componentes de un programa; lo cual permite implementarlos por separado
- Hay un API muy bien definido; cualquiera que use el API, podrá reemplazar el Modelo, la Vista o el Controlador, sin aparente dificultad.
- La conexión entre el Modelo y sus Vistas es dinámica; se produce en tiempo de ejecución, no en tiempo de compilación.

## 2.4 Tecnologías web del lado del cliente

Las tecnologías utilizadas en el lado del cliente están insertadas en el código HTML de la página del cliente y son interpretadas y ejecutadas por el navegador web. Su funcionamiento depende del navegador y la versión utilizada por el cliente (Sánchez, et al., 2007).

### 2.4.1 ExtJS3.3

Tras el éxito de sus versiones beta Sencha<sup>3</sup> lanzó finalmente Ext JS 3.3, la más importante actualización de la línea 3.x. Con nuevas características, componentes y mejoras de rendimiento, reemplaza por completo a Ext JS 3.2, sin requerir cambios considerables para la mayoría de las aplicaciones.

Esta nueva versión suma oportunidades y ventajas, destacándose entre los componentes agregados: (Baires, 2010).

<sup>3</sup>**Sencha:** es un proveedor líder de los marcos de código abierto de aplicaciones web y herramientas para las grandes empresas y desarrolladores independientes.

- PivotGrid: herramientas de resúmenes de datos, ideales para llevar extensos conjuntos de datos. Permite reordenar y agrupar información por cualquier parámetro fijado en una base de datos.
- Componentes de calendario: con Ext JS 3.3 pueden mostrarse eventos por día, semana y mes, todos combinados en una única aplicación.
- Más de 200 mejoras y arreglos respecto a la versión 3.2, originados en los reportes de bugs y nuevos requerimientos.

ExtJS es un framework basado en JavaScript Orientado a Objeto, uno de los más usados en el desarrollo de aplicaciones web en la parte del cliente, debido a que cuenta con una gran cantidad de componentes, clases y validaciones que pueden ser configuradas según las necesidades de cada sistema o del cliente.

Cuenta con una ayuda en inglés, la que tiene diversos ejemplos, de los cuales se pueden auxiliar los desarrolladores para realizar las aplicaciones. Tiene un diseñador de interfaces en línea, lo que facilita la creación de ventanas y componentes, garantizando un rápido desarrollo de sistemas web (ExtJS, 2010).

**Ventajas:** (ExtJS development Group, 2010).

- Existe un balance entre Cliente – Servidor. La carga de procesamiento se distribuye, permitiendo que el servidor, al tener menor carga, pueda manejar más clientes al mismo tiempo.
- Comunicación asíncrona. En este tipo de aplicación el motor de render puede comunicarse con el servidor sin necesidad de estar sujeta a un clic o una acción del usuario, dándole la libertad de cargar información sin que el cliente note esta acción.
- Eficiencia de la red. El tráfico de red puede disminuir al permitir que la aplicación elija que información desea transmitir al servidor y viceversa, sin embargo la aplicación que haga uso de la pre-carga de datos puede que revierta este beneficio por el incremento del tráfico.

**Desventajas:** (ExtJS development Group, 2010).

- Necesita una plataforma para mostrar los componentes y hacer el render de la aplicación.

- Descargas lentas. Al ser aplicaciones grandes, especialmente porque cargan todo al inicio, hace que el tiempo de descarga sea mayor al de una aplicación web tradicional.
- Problemas con los motores de búsqueda. Los motores de búsqueda indexan el contenido web estático por lo que los textos cargados de manera dinámica no serán encontrados.
- Accesibilidad. Estas aplicaciones tienen problemas con los programas de accesibilidad pues, al igual que los motores de búsqueda, no trabajan bien con texto cargado dinámicamente.
- No se pueden usar fuera de línea. Por su naturaleza web estas aplicaciones no pueden ser usadas en el cliente como cualquier otra aplicación.

### 2.4.2 OpenLayer

OpenLayer es un API para construir mapas en aplicaciones web, completamente libre, abierto a todos los desarrolladores, esta aplicación está hecha en JavaScript.

**Ventajas:** (OpenLayers, 2009).

- OpenLayers no requiere instalación.
- Menor procesamiento en el servidor.
- Puede ampliar fácilmente el código para su aplicación en particular.
- Puede utilizar múltiples Servidores de datos.

OpenLayers es una librería JavaScript puro para la visualización de los datos del mapa en la mayoría de navegadores modernos. Implementa un (aún en desarrollo) API de JavaScript para la construcción de aplicaciones web basadas en la representación geográfica, similar a la de Google Maps y MSN Virtual Earth API, con una diferencia importante OpenLayers es software libre, desarrollado por y para la comunidad Open Source Software (OpenLayers, 2009).

### 2.4.3 GeoExt

La librería GeoExt es una herramienta utilizada para construcción de aplicaciones web de mapas basada en JavaScript. GeoExt reúne a los usuarios geoespaciales en cómo saber de OpenLayers con la interfaz de usuario de ExtJS para ayudarle a crear aplicaciones SIG en la web con JavaScript. GeoExt es herramienta de código abierto y se apoya en una creciente comunidad de individuos, empresas y organizaciones (GeoExt, 2010).



## 2.4.4 Estándares de Codificación

Los ficheros javascript no cuentan con un estándar de codificación definido, debido a que al publicar una aplicación en modo de producción estos ficheros son comprimidos usando un compresor de código lo que convierte al código javascript ilegible, imposibilitando su modificación, pero acelerando la carga y ejecución de estos.

Teniendo en cuenta lo anteriormente planteado se propone el siguiente estándar de codificación a usar en los ficheros con javascript vinculados al frameworks ExtJS:

- Use 4 espacios para indentar el código.
- Defina las variables a usar dentro de la configuración de la propia variable.
- El comentario de línea defínalo en una sola línea comenzando por el signo de comentario seguido de un espacio y el comentario en letra minúscula.
- Las variables string defínalas entre comilla simple.
- En caso de que tenga que escribir un comentario de varias líneas, utilice varios comentarios de línea.

## 2.5 Servidores

En el presente epígrafe se describen los servidores usados en el desarrollo de la aplicación web, presentando primeramente el servidor web Apache el cual será el encargado de publicar en línea la aplicación desarrollada, y por otra parte el servidor de base datos PostgreSQL, específicamente la extensión PostGIS que permite el manejo de datos espaciales.

### 2.5.1 Servidor Web Apache

Apache es un servidor web que se encarga de atender las solicitudes realizadas por los usuarios mediante la Internet, brinda la funcionalidad de publicar sitios en línea. El proyecto de Apache es creado y actualizado por la Apache Software Foundation (ASF). Es una organización no lucrativa, ya que distribuyen el servidor apache de forma libre para todo aquel que desee usarlo. ASF es una comunidad descentralizada de desarrolladores que trabajan cada uno en sus propios proyectos de código abierto. Se fundó en 1999 a partir de la creación del Grupo Apache, el cual tenía como objetivo la creación de un servidor Http de código abierto. Entre los objetivos principales de la ASF está la de dar protección legal a todos los voluntarios que trabajan en “Proyectos Apaches”. El Proyecto Apache es el origen de un grupo de licencias de código abierto, las cuales se denominan “Estilo Apache” (ASF, 2010).

La utilización de apache ayuda en la mejora del posicionamiento. El servidor web Apache puede convertirse en una herramienta muy jugosa para crear páginas con enlaces amigables para los buscadores.

Ventajas: (Foundation, 2011).

- Modular
- Código abierto
- Multi-plataforma
- Extensible
- Popular (fácil de conseguir ayuda y soporte)

### **2.5.2 Servidor web Internet Information Services (IIS)**

Windows XP Pro posee un Servicio de publicación Web gratuito, el que incluye servicios de WWW, FTP y Correo SMTP, pero mucha gente ignora este hecho, ya que la instalación por defecto de Windows no los deja instalados (Microsoft, 2011).

Los pilares clave de IIS 7.0 son: (Microsoft, 2011).

- Modelo de extensibilidad flexible para una eficaz personalización.
- Herramientas eficaces de diagnóstico y solución de problemas.
- Administración delegada.
- Seguridad mejorada y superficie expuesta a ataques reducida mediante personalización.
- Administración de mantenimiento y aplicación integrada para servicios de Windows Communication Foundation (WCF).
- Herramientas de administración mejoradas.

### **2.5.3 Selección del servidor web**

Apache será el servidor web que se utilizará debido a que es multiplataforma y de código abierto mientras que IIS de Microsoft es propietario y funciona solamente en Windows. Por otra parte Apache es configurable y su forma modular permite el aumento de sus funcionalidades, cuenta con soporte para varios lenguajes y una magnífica gestión de los logs de seguridad dándole una mayor seguridad al servidor, todas estas características hacen de Apache el servidor mas utilizado a nivel mundial (Ver figura 5).

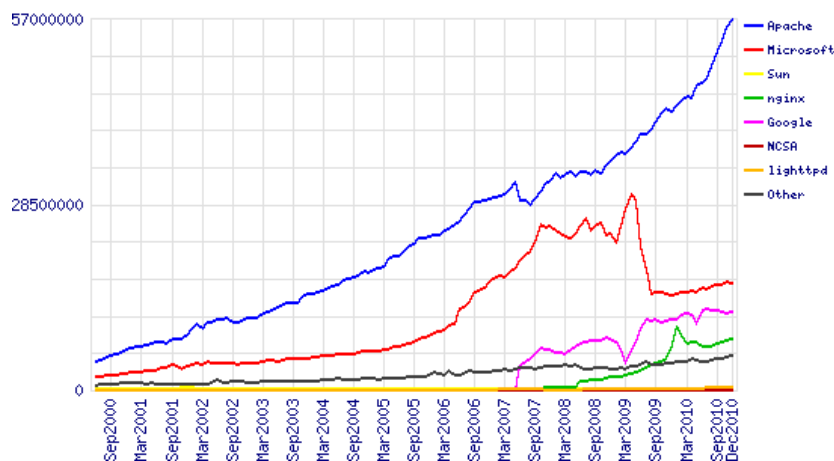


Figura 5. Servidores web más utilizados (Copca, 2004)

#### 2.5.4 Servidor de Base de Datos PostgreSQL

PostgreSQL es un potente sistema de base de datos objeto – relacional de código abierto. Cuenta con un historial de 15 años de activo desarrollo, en los cuales se ha ganado una gran reputación por ser confiable, correcta integración de datos y alto rendimiento. Se puede ejecutar sobre la mayoría de los Sistemas Operativos (SO) que existen en la actualidad, como Linux, Microsoft Windows, Mac OS X, Solaris, Fedora, etc. Cuenta con un completo soporte para llaves, llaves foráneas, uniones, vistas, disparadores y procedimientos almacenados. Tiene todos los tipos de datos que son usados por SQL Server 2008 y cuenta con interfaces para su comunicación en la mayoría de los lenguajes existentes como C, C++, C#, Perl, PHP, Java, Python, Ruby, ODBC (PostgreSQL Foundation, 2009).

El desarrollo de PostgreSQL es realizado por un equipo de desarrolladores (voluntarios en su mayoría) dispersos alrededor del mundo y comunicados vía Internet. Este es un proyecto de la comunidad y no es controlado por ninguna compañía (Medina, 2006).

**Principales características:** (Medina, 2006).

- Soporta casi todas las sintaxis.
- Integridad transaccional.
- Acceso concurrente multiversión, no se bloquean las tablas, ni siquiera las filas, cuando un proceso escribe. Es la tecnología que PostgreSQL usa para evitar bloqueos innecesarios.
- PostgreSQL usa una arquitectura cliente/servidor.

- La característica de PostgreSQL conocida como Write Ahead Logging incrementa la dependencia de la base de datos al registro de cambios antes de que estos sean escritos en la base de datos. Esto garantiza que en el hipotético caso de que la base de datos se caiga, existirá un registro de las transacciones a partir del cual se puede restaurar la base de datos.
- Interfaces con lenguajes de programación. Estas interfaces incluyen Object Pascal, Python, Perl, PHP, ODBC, Java/JDBC, Ruby, TCL, C/C++, etc.
- Herencia de tablas.
- Incluye la mayoría de los tipos de datos SQL92 y SQL99, soporta almacenamiento de objetos grandes binarios, además de tipos de datos y operaciones geométricas.
- Cuenta con un sofisticado analizador/optimizador de consultas.

PostgreSQL ofrece muchas ventajas respecto a otros sistemas de bases de datos:

- Instalación ilimitada: Es frecuente que las bases de datos comerciales sean instaladas en más servidores de lo que permite la licencia. Algunos proveedores comerciales consideran a esto la principal fuente de incumplimiento de licencia. Con PostgreSQL, nadie puede demandarlo por violar acuerdos de licencia, puesto que no hay costo asociado a la licencia del software.
- Mejor soporte que los proveedores comerciales.
- Ahorros considerables en costos de operación: El software ha sido diseñado y creado para tener un mantenimiento y ajuste mucho menor que los productos de los proveedores comerciales, conservando todas las características, estabilidad y rendimiento.
- Estabilidad y confiabilidad legendarias: En contraste a muchos sistemas de bases de datos comerciales, es extremadamente común que compañías reporten que PostgreSQL nunca ha presentado caídas en varios años de operación de alta actividad.
- Multiplataforma: PostgreSQL está disponible en casi cualquier Unix (34 plataformas en la última versión estable), y una versión nativa de Windows está actualmente en estado beta de pruebas.
- Diseñado para ambientes de alto volumen: PostgreSQL usa una estrategia de almacenamiento de filas para conseguir una mejor respuesta en ambientes de grandes volúmenes.

Es importante tener en cuenta las limitaciones que tiene PostgreSQL: (Medina, 2006).

- Máximo tamaño de base de datos ilimitado.
- Máximo tamaño de tabla 32 TB.
- Máximo tamaño de tupla 1.6 TB.
- Máximo tamaño de campo 1 GB.
- Máximo tuplas por tabla ilimitado.
- Máximo columnas por tabla 250 - 1600 dependiendo de los tipos de columnas.
- Máximo de índices por tabla ilimitado.

## **PostGIS**

PostGIS es una extensión al sistema de base de datos objeto-relacional PostgreSQL. Permite el uso de objetos GIS (Geographic Information Systems). Esta creado por Refrations Research Inc, como un proyecto de investigación de tecnologías de bases de datos espaciales. Está publicado bajo licencia GNU (Martín, 2008).

Características de PostGIS: (Martín, 2008).

- Permite usar todos los objetos que aparecen en las especificaciones OpenGIS (puntos, líneas, polígonos, multilineas, multipuntos y colecciones geométricas).
- Puede funcionar con GEOS (Geometry Engine Open Source) como motor de comprobación topológica.

### **2.5.5 MySQL**

MySQL es un sistema de gestión de bases de datos relacional, presenta un diseño multihilo que le permite soportar una gran carga de forma muy eficiente. Aunque es software libre, también se distribuye una versión comercial que no se diferencia de la versión libre más que en el soporte técnico que se ofrece y la posibilidad de integrar este gestor en un software propietario (Oracle, 2010).

Se plantea que este gestor de bases de datos es, posiblemente, el gestor más usado en el mundo del software libre, ya que es sumamente rápido y con mucha facilidad de uso, además de su sencilla instalación y configuración

### 2.5.6 Oracle

Oracle es un sistema de gestión de base de datos relacional que se considera como uno de los más completos, destacando: el soporte de transacciones, la estabilidad, escalabilidad y soporte multiplataforma (Oracle, 2010).

Características: (Copca, 2004).

- Es una herramienta de administración gráfica muy intuitiva y cómoda de utilizar.
- Ayuda a analizar datos y efectuar recomendaciones concernientes a mejorar el rendimiento y la eficiencia en el manejo de aquellos datos que se encuentran almacenados.
- Apoya el diseño y optimización de modelos de datos.
- Asiste a los desarrolladores con sus conocimientos de SQL y de construcción de procedimientos almacenados y *triggers*, entre otros.
- Apoya en la definición de estándares de diseño y nomenclatura de objetos.
- Permite documentar y mantener un registro periódico de las mantenciones, actualizaciones de hardware y software, cambios en las aplicaciones y, en general, todos aquellos eventos relacionados con cambios en el entorno de utilización de una base de datos.

### 2.5.7 Selección del servidor de base de datos a utilizar

El servidor de base de datos que se utilizará en el desarrollo del sistema será Postgres debido principalmente a que no es necesario una licencia para incluirlo en el producto, además de que el trabajo con los datos espaciales es muy superior tanto a Oracle como a MySQL.

## 2.6 Herramientas de Desarrollo

Las herramientas de desarrollo permiten el uso de técnicas y herramientas conocidas para crear controles y contenido web. Estas herramientas incluyen Visual Basic, Visual C++, Macromedia ShockWave, Adobe Photoshop, Visual Paradigm, Rational Rose, Zend Studio, Borland Delphi, entre otras.

Se escogió Visual Paradigm para el modelado del ciclo de vida completo del desarrollo de software y Zend Studio como IDE de desarrollo, son las herramientas descritas en este epígrafe.

## 2.6.1 Visual Paradigm

Visual Paradigm es una herramienta CASE<sup>4</sup> para UML (Lenguaje de Modelado Unificado) es una herramienta que apoya las actividades que tienen lugar a todo lo largo del ciclo de vida del desarrollo de software. Además facilita la diagramación visual y el diseño de proyectos y una fácil integración con la mayoría de los IDEs<sup>5</sup> de desarrollo. Para desarrolladores independientes existe una versión llamada Community Edition que es de uso no comercial.

El software de modelado UML ayuda a una más rápida construcción de aplicaciones de calidad, mejores y a un menor costo. Permite dibujar todos los tipos de diagramas de clases, ingeniería inversa, generar código desde diagramas y generar documentación. Esta herramienta soporta UML 2.1 completo, notación para el Modelado de Procesos de Negocio (BPMN) y permite realizar ingeniería tanto directa como inversa (Visual Paradigm Design Group, 2009).

A partir de un modelo relacional en SQL Server, MySQL, etc. Es capaz de desplegar todas las clases asociadas a las tablas (siguiendo el patrón de diseño Una Clase-Una Tabla). Para gestionar la persistencia y el mapeo de estas clases con la base de datos utiliza Hibernate para Java y NHibernate en el caso de un proyecto .Net. Es colaborativa, soporta múltiples usuarios trabajando sobre el mismo proyecto; genera la documentación del proyecto automáticamente en varios formatos como web o PDF, y permite control de versiones.

Es posible generar código desde Visual Paradigm para plataformas como .NET, Java y PHP, así como obtener diagramas a partir del código, esto es de gran utilidad pues ahorra tiempo a los desarrolladores y reduce las posibilidades de cometer errores. Brinda la posibilidad de obtener una base de datos relacional y el código necesario para acceder a esta a partir de un Diagrama Entidad Relación, se puede conectar de forma fácil a diversos servidores de base de datos. Se integra con varios ambientes de desarrollo integrados (IDE) lo cual permite pasar del código al modelado y viceversa. Establece interoperabilidad con otras aplicaciones como el Visio y el Rational Rose. Disponible en múltiples lenguajes y plataformas: Microsoft Windows (98, 2000, XP, o Vista), Linux, Mac OS X, Solaris o Java (Varona, 2009).

---

<sup>4</sup> **CASE:** Ingeniería de Software Asistida por Computadora (Computer Aided Software Engineering).

<sup>5</sup> **IDE** (Entorno de desarrollo integrado): Es un programa informático compuesto por un conjunto de herramientas de programación.

## **2.6.2 Erwin**

Erwin es otra de las herramientas CASE, cuyo mayor diferenciador es su simplicidad por generar código para la mayoría de los manejadores de base de datos ya que es completamente abierta y la rapidez para el desarrollo de bases de datos complejas (Community, 2010).

Esta herramienta ofrece una metodología para realizar diagramas entidad-relación y cuenta con una interfaz gráfica altamente intuitiva.

## **2.6.3 Rational Rose**

Es una herramienta de producción y comercialización, es un instrumento operativo que utiliza el lenguaje unificado (UML) como medio para facilitar la captura de dominio de la semántica, la arquitectura y el diseño (Alonso, 2009).

Esta herramienta no es gratuita por lo que se debe hacer un previo pago para poder adquirir el producto.

## **2.6.4 Selección de la herramienta a utilizar**

Las herramientas CASE son de vital importancia, ya que permiten el modelado del sistema antes de comenzar a desarrollarlo, previendo así posibles errores. Los tres software anteriormente descritos están entre los más potentes y completos a nivel mundial. Para el desarrollo del sistema se determina que la opción más adecuada es usar Visual Paradigm ya que se puede utilizar sobre plataformas Unix.

## **2.6.5 Zend Studio**

Zend Studio es un IDE usado para el desarrollo de aplicaciones web, tanto para lenguaje PHP como para Java. Al ser creado y mantenido por el mismo equipo que desarrolla PHP, está optimizado para la integración con este lenguaje de programación. Está diseñado para aumentar la productividad, velocidad y rendimiento en el trabajo colectivo e individual (Zend GDE, 2010).

Zend Studio consta de dos partes en las que se dividen las funcionalidades de parte del cliente y las del servidor. Las dos partes se instalan por separado, la del cliente contiene el interfaz de edición y la ayuda. Permite además hacer depuraciones simples de scripts, aunque para disfrutar de toda la potencia de la herramienta de depuración habrá que disponer de la parte del servidor,



que instala Apache y el módulo PHP o, en caso de que estén instalados, los configura para trabajar juntos en depuración (Aroche, 2003).

- **Características:** excelente completamiento de código, coloreado en la sintaxis del código, administración avanzada de proyectos, múltiples lenguajes, incorpora el Framework de Zend, PHP Documentor, manual de PHP. Integración con subversión, los navegadores, integración avanzada con FTP. Soporte para Web Services, PHP4, PHP5 y SQL (Valdés, 2008).
- **Ventajas:** agiliza el trabajo de los desarrolladores, cuenta con un buen Depurador, infinitas opciones que permiten un desarrollo profesional de las aplicaciones (Valdés, 2008).

Cuenta con la posibilidad de añadir más funcionalidades, mediante la inclusión de plugins, uno de los más usados en el desarrollo en equipo es el subclipse, el cual permite la conexión con un repositorio o control de versiones para asegurar un correcto trabajo en equipo. Cuenta con soporte y reconocimiento de la última versión de PHP y Eclipse Galileo usado para la programación en ficheros javascript, hojas de estilo y páginas clientes. Tiene un amplio completamiento de código para la web y la posibilidad de integrarles las librerías que sean usadas en el desarrollo de proyectos (Zend GDE, 2010).

Zend Studio es el IDE seleccionado para el desarrollo del sistema debido a las grandes ventajas que presenta. Este IDE fue desarrollado específicamente para PHP por lo tanto cuenta con un completamiento de código muy superior, cuenta además con detección de errores de sintaxis en tiempo real y permite la depuración en tiempo de ejecución. Todas estas características lo convierten en una opción muy adecuada para el desarrollo de este tipo de sistema.

## **2.7 Conclusiones Parciales**

El desarrollo de la investigación referente al presente capítulo dio lugar a un mayor entendimiento de las herramientas necesarias para el desarrollo, permitiendo la selección de las más adecuadas para el diseño e implementación del sistema. La información recopilada en este capítulo servirá de base para desarrollar un producto de forma rápida y con una elevada calidad.

# Capítulo 3: Análisis de la solución propuesta

## 3.1 Introducción

En el presente capítulo se describen los requisitos funcionales y no funcionales con los que debe cumplir el sistema, lo que permite comprender e identificar mediante un diagrama de caso de uso las relaciones de los actores que interactúan con el sistema, y las secuencias de acciones con las que interactúan.

## 3.2 Descripción del Modelo de Dominio

Todos los usuarios serán estándar y podrán acceder al sistema sin necesidad de registrarse, podrán acceder a todas las funcionalidades y visualizar el mapa que mostrará los datos representados y espacialmente referenciados (Ver Figura 6).

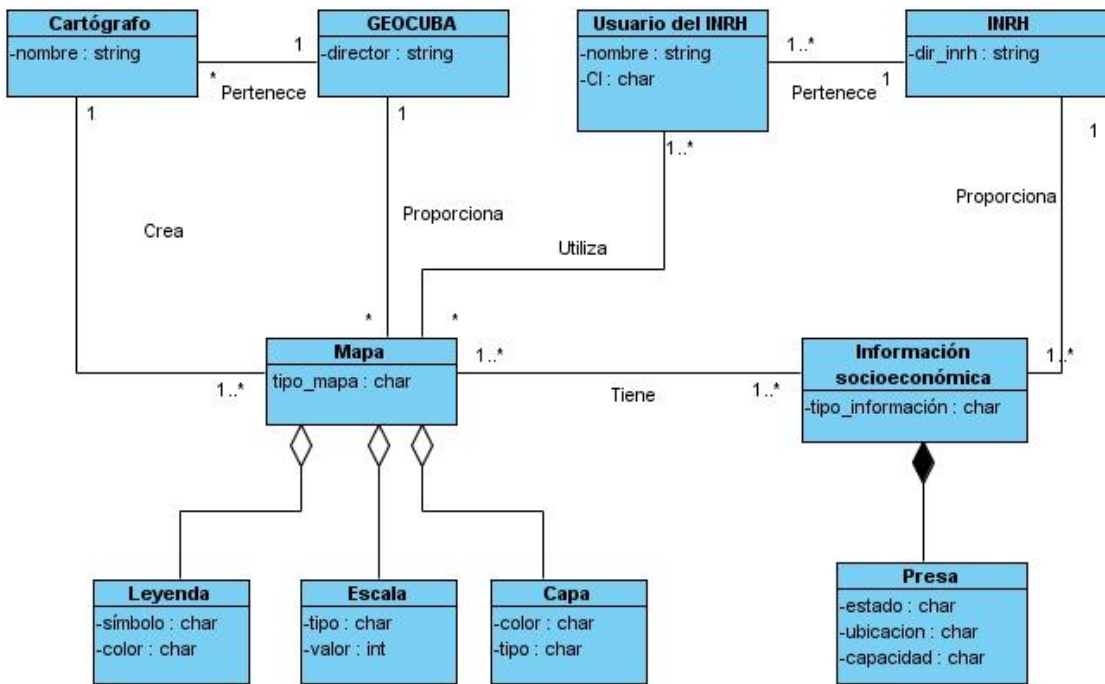


Figura 6. Modelo de dominio.

## 3.3 Especificación de los requisitos de la aplicación

El éxito de un proyecto se debe medir por varios factores, pero indiscutiblemente la satisfacción del cliente es uno de los principales. La comunicación fluida entre el cliente y el equipo de desarrollo con el objetivo de lograr un entendimiento común de lo que se desea constituye la clave del éxito en la producción de software. Los requisitos funcionales y no funcionales del sistema, son el producto de este entendimiento entre

clientes y equipo de desarrollo, además de ser una de las forma de comprobar la culminación del sistema.

Los requisitos funcionales son capacidades o condiciones que el sistema debe cumplir, los cuales constituyen un paso esencial para la implementación del sistema propuesto (Jacobson, 2000).

Se puede plantear que un requisito no funcional es un requisito de software que describe no lo que el software hará, sino como lo hará. Los requisitos no funcionales son difíciles de verificar/testear, y por ello son evaluados subjetivamente (Thayer, 1990).

### **3.3.1 Requisitos Funcionales**

A continuación se exponen los requisitos funcionales (RF) del sistema:

**RF 1. Realizar Acercamiento:** Con este requerimiento se permite aumentar el tamaño del mapa y disminuye la escala, ubicando en el centro del mapa el punto en el que el usuario realizó la operación de Acercar.

**RF 2. Realizar Alejamiento:** Con este requerimiento se disminuye el tamaño del mapa y aumenta la escala.

**RF 3. Recentrar mapa:** Con este requerimiento se permite que el usuario pueda recentrar una región previamente seleccionada al dar clic en el mapa, sin modificar su escala.

**RF 4. Mover el mapa:** Con este requerimiento se permite que el usuario pueda mover el mapa variando con el puntero del ratón la posición de la vista que se presenta.

**RF 5. Localizar Objetivo:** Con este requerimiento se permite la localización de objetivos. En este requerimiento se activa otra ventana, donde el usuario puede seleccionar lo que se desea localizar, dando clic sobre ello, en el mapa se ubica el lugar de interés de forma puntual.

**RF 6. Identificar Geometría Puntual:** Con este requerimiento se quiere que el usuario pueda consultar la información asociada a los objetos que forman parte de las capas del mapa que fueron seleccionadas previamente; esta consulta puede ser de tipo puntual.

**RF 7. Localizar por Coordenadas:** Con este requerimiento se activa una ventana donde el usuario puede introducir la longitud y latitud, y se ubica en pantalla el lugar de interés.

**RF 8. Medir distancia:** Con este requerimiento se quiere que el usuario pueda calcular la distancia entre dos o más puntos a partir del trazado de los mismos con el ratón y visualizar el valor de la distancia entre los dos últimos vértices dibujados y la distancia acumulada (del primer al último vértice) en un panel lateral.

**RF 9. Calcular área:** Con este requerimiento se quiere que el usuario pueda calcular el área de una región a partir del trazado de la misma con el ratón sobre el mapa, se visualizan los valores del área en un panel lateral.

**RF 10. Personalizar mapa:** Con este requerimiento se quiere que el usuario pueda crear un mapa temático entrando los valores asociados a los criterios de búsqueda regionales y a los criterios de búsqueda estadística.

### **3.3.2 Requisitos no funcionales**

Los requerimientos no funcionales son propiedades o cualidades que el producto debe tener y que harán del producto un sistema confiable y seguro. El sistema cuenta con los siguientes requisitos no funcionales:

Apariencia o interfaz externa:

El acceso a la aplicación y su presentación será sencillo, fácil de manejar por el usuario. Tendrá una interfaz amigable y similar al ambiente de desarrollo del sistema operativo Windows. Poseerá el color azul fundamentalmente.

Interfaces de usuario:

El sistema debe:

- Tener una apariencia profesional y un diseño gráfico sencillo.
- Posibilitarle al usuario la configuración del entorno de trabajo.
- Ser intuitivo.

Interfaces de hardware:

Para las PCs clientes:

- Se requiere tengan tarjeta de red.

- Al menos 128 MB de memoria RAM.
- 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 Web 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.

Usabilidad:

- El sistema podrá ser usado por personas con conocimientos básicos en el manejo de computadoras. Se emplearán componentes que indiquen al usuario el estado de los procesos que por su complejidad requieran de un tiempo de procesamiento apreciable.
- El software tendrá siempre visible la opción de Ayuda, lo que posibilitará un mejor aprovechamiento por parte de los usuarios de sus funcionalidades.
- El sistema tendrá una correcta Arquitectura de la Información, a partir de un estudio de usuarios para su etiquetado e indexado.
- Las funcionalidades principales del sistema estarán orientadas a iconos para un mayor reconocimiento por parte del usuario.

Fiabilidad:

- El sistema debe tener soporte para recuperación ante fallos y errores. La información manejada por el mismo estará protegida de acceso no autorizado y divulgación.
- Debido a la arquitectura que presenta el sistema, siendo más robusto al no tratarse de un sistema de gestión que requiera mantenimiento y optimización en el almacenamiento, se estima un tiempo promedio de 6 meses entre posibles fallas. El tiempo medio de reparación, en caso de un fallo es de 7 días.

### **3.4 Definición de los casos de usos del sistema**

#### **3.4.1 Actores del sistema**

Un actor es una idealización de una persona externa, de un proceso, o de un objeto que interactúa con un sistema, un subsistema, o una clase. Un actor caracteriza las

interacciones que los usuarios exteriores pueden tener con el sistema. En tiempo de ejecución, un usuario físico puede estar limitado a los actores múltiples dentro del sistema. Diferentes usuarios pueden estar ligados al mismo actor y por lo tanto pueden representar casos múltiples de la misma definición de actor (Pressman, 2000).

Se puede decir que un actor del sistema va a hacer todo aquel trabajador del negocio que posea actividades a automatizar (Ver tabla 1).

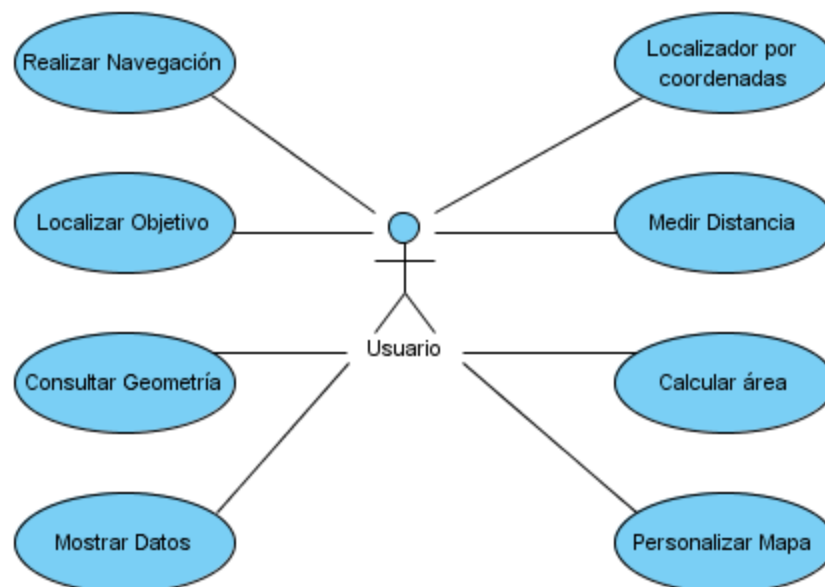
**Tabla 1. Descripción total del actor del sistema**

Actor	Descripción
Usuario	Cualquier usuario que utiliza las funcionalidades del sistema.

### 3.4.2 Diagrama de casos de uso del sistema

Un diagrama de Casos de Uso del Sistema describe parte del modelo de casos de uso y muestra un conjunto de casos de uso y actores con una asociación entre cada par actor/caso de uso que interactúan (Jacobson, 2000).

A continuación se muestra el diagrama de casos de uso del sistema para el sistema de información geográfica para el INRH: (Ver figura 7).



**Figura 7. Diagrama de Caso de Uso del Sistema**

### 3.4.3 Especificación de los casos de uso del sistema

La definición de un caso de uso incluye todo el comportamiento que implica: las líneas principales, las diferentes variaciones sobre el comportamiento normal, y todas las

condiciones excepcionales, que pueden ocurrir con tal comportamiento, junto con la respuesta deseada. Desde el punto de vista de los usuarios, éstas pueden ser situaciones anormales. Desde el punto de vista de los sistemas, son las variaciones adicionales que deben ser descritas y manejadas (Rumbaugh, 2000).

**Tabla 2. CUS Realizar Navegación**

<b>Caso de Uso:</b>	Realizar Navegación.	
<b>Actores:</b>	Usuario	
<b>Propósito</b>	Este caso de uso se lleva a cabo con el objetivo de modificar la visualización inicial del mapa en la pantalla.	
<b>Resumen:</b>	El caso de uso se inicia cuando el usuario selecciona el botón “Realizar Navegación” y termina cuando el sistema muestra el mapa resultante en la pantalla.	
<b>Referencias</b>	RF 1, RF 2, RF 3, RF 4.	
<b>Prioridad</b>	Crítico	
<b>Flujo Normal de Eventos</b>		
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>	
<p>1. El usuario selecciona una de las opciones de navegación :</p> <ul style="list-style-type: none"> <li>- Acercar</li> <li>- Alejar.</li> <li>- Zoom Extenso.</li> <li>- Mover Mapa.</li> <li>- Recentrar Mapa</li> <li>- Navegar a través del Mapa de Referencia</li> </ul>	<p>2. El sistema realiza la operación según la opción seleccionada por el usuario.</p> <ul style="list-style-type: none"> <li>- Se seleccionó “Acercar”, ver sección “Acercar”.</li> <li>- Se seleccionó “Alejar”, ver sección “Alejar”.</li> <li>- Se seleccionó “Zoom Extenso”, ver sección “Zoom Extenso”.</li> <li>- Se seleccionó “Mover Mapa”, ver sección “Mover Mapa”.</li> <li>- Se seleccionó “Recentrar Mapa”, ver sección “Recentrar Mapa”.</li> <li>- Se seleccionó “Navegar a través del Mapa de Referencia”, ver sección “Navegar a través del Mapa de Referencia”.</li> </ul>	
	3. El sistema procesa la	

	información según la acción realizada por el usuario y actualiza la visualización del mapa.
<b>Sección “Acercar”</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
1. El usuario selecciona la opción Acercar y dibuja un punto con un clic en la región específica del mapa que desea visualizar.	2. El sistema realiza la operación de Acercar seleccionada por el usuario aumentando el zoom y disminuyendo la escala, es decir, toma las coordenadas (x, y) y mueve al centro del mapa el punto donde dio clic.
	3. El sistema procesa la información a partir de la acción realizada por el usuario y actualiza la visualización del mapa.
<b>Sección “Alejar”</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
1. El usuario selecciona la opción Alejar y da clic en un punto del mapa.	2. El sistema realiza la operación de Alejar seleccionada por el usuario disminuyendo el zoom y aumentando la escala puntualmente y mueve al centro del mapa el punto donde dio clic.
	3. El sistema procesa la información a partir de la acción realizada por el usuario y actualiza la visualización del mapa.
<b>Sección “Recentrar Mapa”</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
1. El usuario selecciona la opción Recentrar el mapa y marca un punto en el mapa de la región que desea recentrar.	2. El sistema obtiene las coordenadas (x, y) del punto donde el usuario hizo clic y lo traslada al centro de la pantalla sin cambiar la escala del mapa.

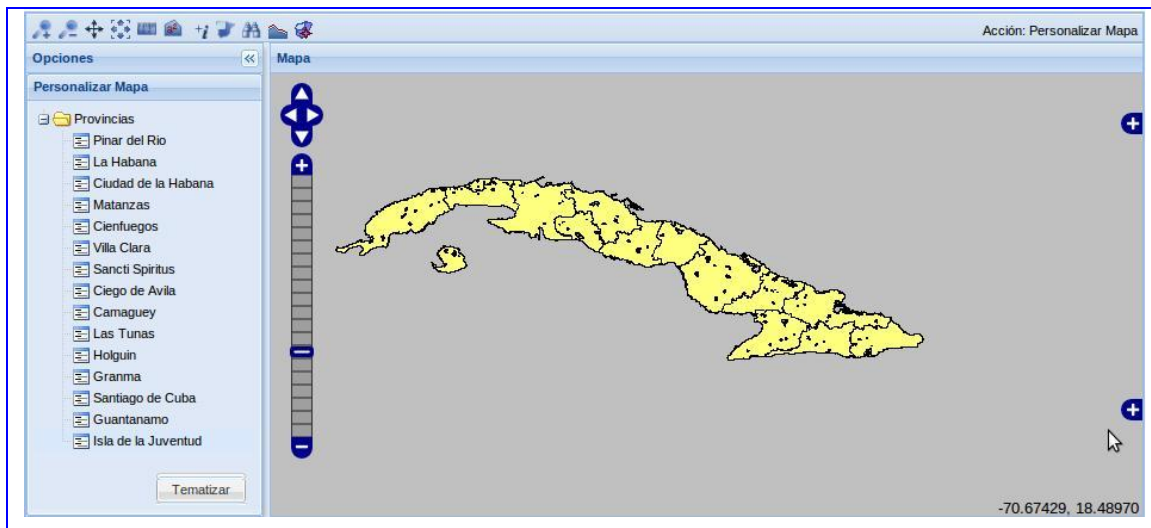


	<p><b>3.</b> El sistema procesa la información a partir de la acción realizada por el usuario y actualiza la visualización del mapa.</p>
<p><b>Sección “Zoom Extenso”</b></p>	
<p><b>Acción del Actor</b></p>	<p><b>Respuesta del Sistema</b></p>
<p>1. El usuario selecciona la opción Zoom extenso.</p>	<p>2. El sistema visualiza el mapa según el extend inicial de la aplicación.</p>
<p><b>Sección “Mover Mapa”</b></p>	
<p><b>Acción del Actor</b></p>	<p><b>Respuesta del Sistema</b></p>
<p>1. El usuario da clic en un punto determinado y arrastra el mouse hasta obtener la visualización deseada.</p>	<p>2. El sistema calcula un <math>\Delta X</math> y un <math>\Delta Y</math> a partir del centro de pantalla y se obtiene un nuevo centro de pantalla a donde se movería el mapa. No varía la escala ni el zoom de representación del mapa.</p>
	<p>3. El sistema procesa la información a partir de la acción realizada por el usuario y actualiza la visualización del mapa.</p>
<p><b>Sección “Navegar a través del Mapa de Referencia”</b></p>	
<p><b>Acción del Actor</b></p>	<p><b>Respuesta del Sistema</b></p>
<p>1. El usuario hace clic en un punto del mapa de referencia.</p>	<p>2. El sistema toma las coordenadas del punto y centra el mapa sin modificar la escala. En el mapa de referencia el sistema va mostrando en un rango de escala determinado un rectángulo equivalente al extend visualizado y, muestra el área de visualización con un rectángulo.</p>
	<p>3. El sistema procesa la información a partir de la acción realizada por el usuario y actualiza la visualización del mapa.</p>

<b>Poscondiciones</b>	El sistema visualiza el mapa a partir de la acción realizada por el usuario.
-----------------------	--

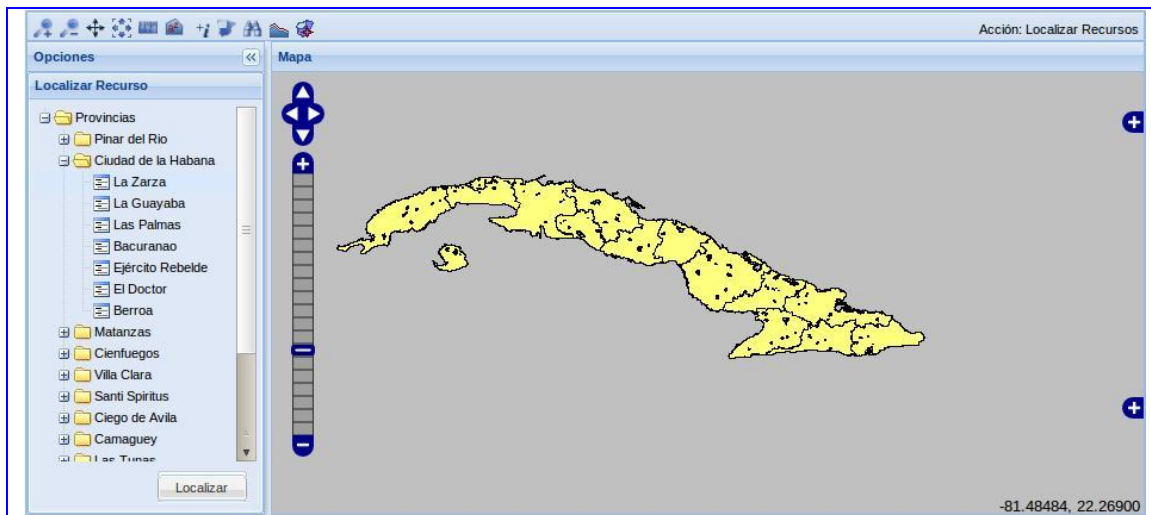
**Tabla 3. CUS Personalizar mapa**

<b>Caso de Uso:</b>	Personalizar mapa.	
<b>Actores:</b>	Usuario	
<b>Propósito</b>	Con este requerimiento se quiere que el usuario pueda crear un mapa temático entrando los valores asociados a los criterios de búsqueda regionales como provincias y municipios, y a los criterios de búsqueda estadística como el nivel de llenado, la cantidad de presas y el estado en que se encuentran.	
<b>Resumen:</b>	El caso de uso se inicia cuando el usuario selecciona el botón “Personalizar Mapa” y termina cuando se muestra el mapa personalizado de acuerdo a los parámetros seleccionados.	
<b>Referencias</b>	RF 10	
<b>Prioridad</b>	Crítico	
<b>Flujo Normal de Eventos</b>		
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>	
1. El usuario selecciona la opción personalizar mapa.	2. El sistema realiza la operación según la opción seleccionada por el usuario. - Se seleccionó personalizar mapa.	
	3. El sistema procesa la información a partir de la acción realizada por el usuario y actualiza la visualización del mapa.	
<b>Prototipo de Interfaz</b>		



**Tabla 4. CUS Localizar Objetivo**

<b>Caso de Uso:</b>	Localizar Objetivo.	
<b>Actores:</b>	Usuario	
<b>Propósito</b>	Este caso de uso se lleva a cabo con el objetivo de obtener información asociada a los objetos seleccionados por el usuario así como su georeferenciación.	
<b>Resumen:</b>	El caso de uso se inicia cuando el usuario selecciona el botón “Localizar Objetivo” y termina cuando el objetivo es localizado en el mapa.	
<b>Referencias</b>	RF 5	
<b>Prioridad</b>	Crítico	
<b>Flujo Normal de Eventos</b>		
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>	
1. El usuario selecciona la opción “Localizar objetivo”.	2. El sistema realiza la operación según la opción seleccionada por el usuario. - Se seleccionó Localizar Objetivo.	
	3. El sistema procesa la información a partir de la acción realizada por el usuario y actualiza la visualización del mapa.	
<b>Prototipo de Interfaz</b>		

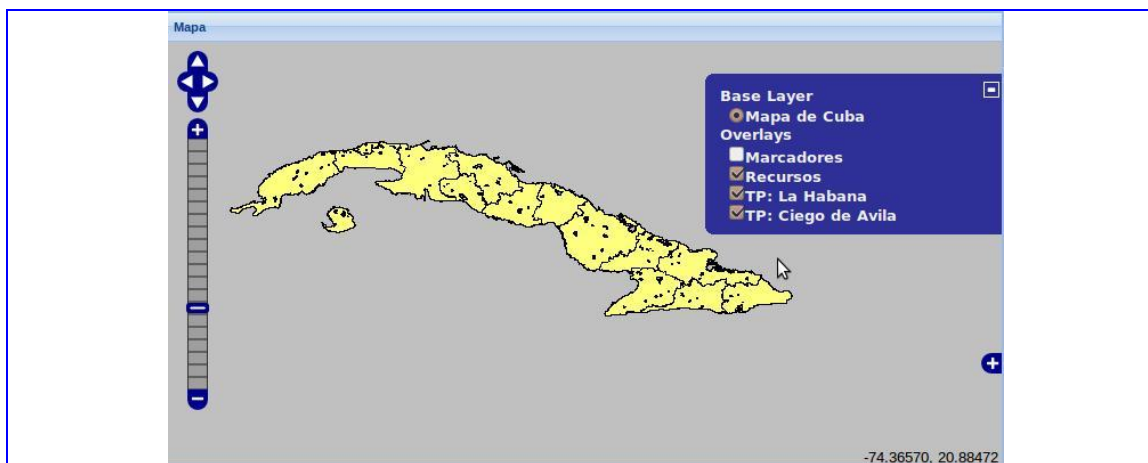


### Flujos Alternos

Acción del Actor	Respuesta del Sistema
	5.1 El sistema muestra en el panel lateral el mensaje “No existe objetivo asociado a esos datos”.

Tabla 5. CUS Realizar Control de Selección

<b>Caso de Uso:</b>	Realizar Control de Selección.
<b>Actores:</b>	Usuario
<b>Propósito</b>	Este caso de uso se lleva a cabo con el objetivo de cambiar las capas seleccionables y visibles de un mapa.
<b>Resumen:</b>	El caso de uso se inicia cuando el usuario selecciona la opción “Control de Selección” y termina cuando el sistema modifica el estado de visibilidad de las capas.
<b>Referencias</b>	
<b>Prioridad</b>	Crítico
<b>Flujo Normal de Eventos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
1. El usuario selecciona la opción “Control de Selección”	2. El sistema visualiza una ventana donde muestra las capas que puede seleccionar el usuario.
3. El usuario selecciona las capas que serán visibles.	4. El sistema habilitará o deshabilitará las opciones de visualización de las capas.
<b>Prototipo de Interfaz</b>	



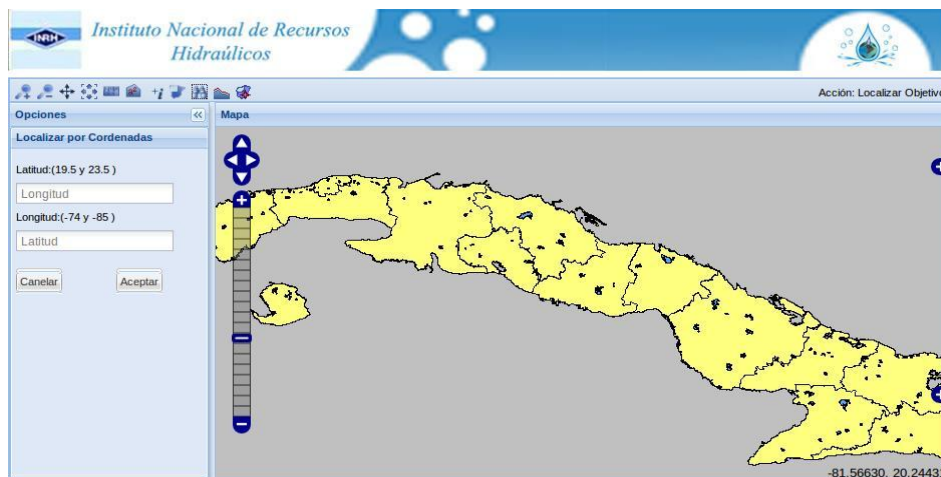
Flujos Alternos	
Acción del Actor	Respuesta del Sistema
3.1 El usuario desmarca las capas que estén activas por una selección anterior que no desee visualizar y selecciona las capas que serán visibles.	4.1 El sistema actualizará la visualización las capas.
<b>Poscondiciones</b>	El sistema guarda todos los valores asociados a las capas seleccionadas por el usuario.

Tabla 6. CUS Localizar por Coordenadas

<b>Caso de Uso:</b>	Localizar por Coordenadas.
<b>Actores:</b>	Usuario
<b>Propósito</b>	Este caso de uso se lleva a cabo con el objetivo de localizar cualquier objeto del mapa de acuerdo a la latitud y longitud proporcionadas por el usuario.
<b>Resumen:</b>	El caso de uso se inicia cuando el usuario selecciona el botón “Localizar por coordenadas” y termina cuando el sistema muestra un marcador y recentra el mapa a las coordenadas introducidas por el usuario.
<b>Referencias</b>	RF 7.
<b>Prioridad</b>	Secundario
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. El usuario selecciona la opción “Localizar por Coordenadas”.	2. El sistema muestra la ventana Localizar por Coordenadas en el panel lateral.

	Para los valores de las coordenadas se tienen dos cajas de texto, uno para la “Latitud” y otro para la “Longitud”.
3. El usuario introduce los datos asociados a los valores de coordenadas. Al finalizar da clic en el botón “Aceptar”.	4. El sistema muestra un marcador y mueve el centro del mapa a las coordenadas introducidas por el usuario.

### Prototipo de Interfaz



### Flujos Alternos

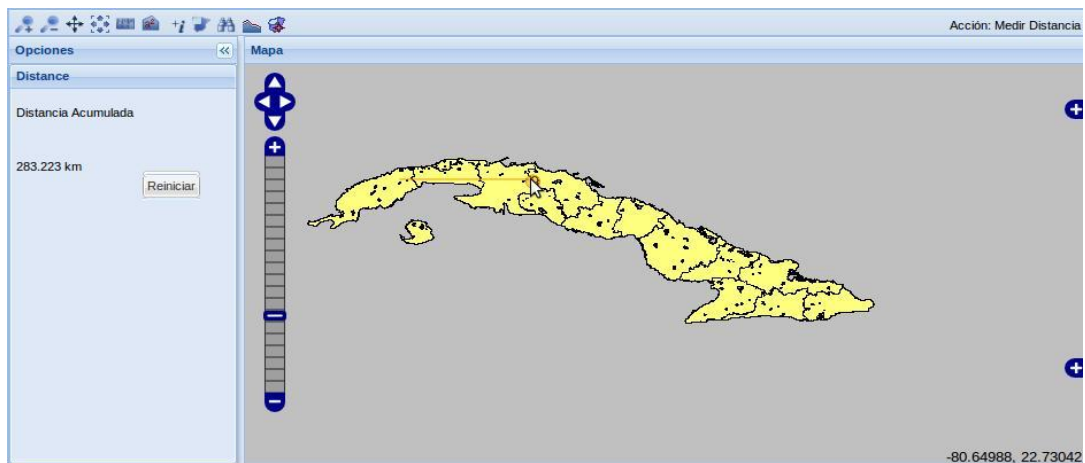
Acción del Actor	Respuesta del Sistema
3.1 El usuario hace clic en el botón “Cancelar”.	4.1 El sistema cancela todo tipo de operación realizada y cierra la ventana.
<b>Poscondiciones</b>	El sistema visualiza un marcador en las coordenadas introducidas por el usuario.

Tabla 7. CUS Medir Distancia

<b>Caso de Uso:</b>	Medir Distancia.
<b>Actores:</b>	Usuario
<b>Propósito</b>	Este caso de uso se lleva a cabo con el objetivo de medir las distancias entre puntos en el mapa.
<b>Resumen:</b>	El caso de uso se inicia cuando el usuario selecciona el botón “Medir Distancia” y termina cuando el sistema muestra los valores correspondientes.
<b>Referencias</b>	RF 8
<b>Prioridad</b>	Crítico

Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. El usuario selecciona la opción "Medir Distancia".	2. El sistema actualiza la barra de información.
3. El usuario realiza el trazado de una polilínea dando clic izquierdo y arrastrando el cursor y termina con un clic en el mapa cuando haya realizado el trazo deseado.	4. El sistema crea una capa temporal donde dibuja la polilínea trazada por el usuario. Las distancias se expresan en la unidad de medida que muestra la aplicación por defecto.
	5. El sistema muestra en la barra de izquierda la distancia total del primer vértice al actual.

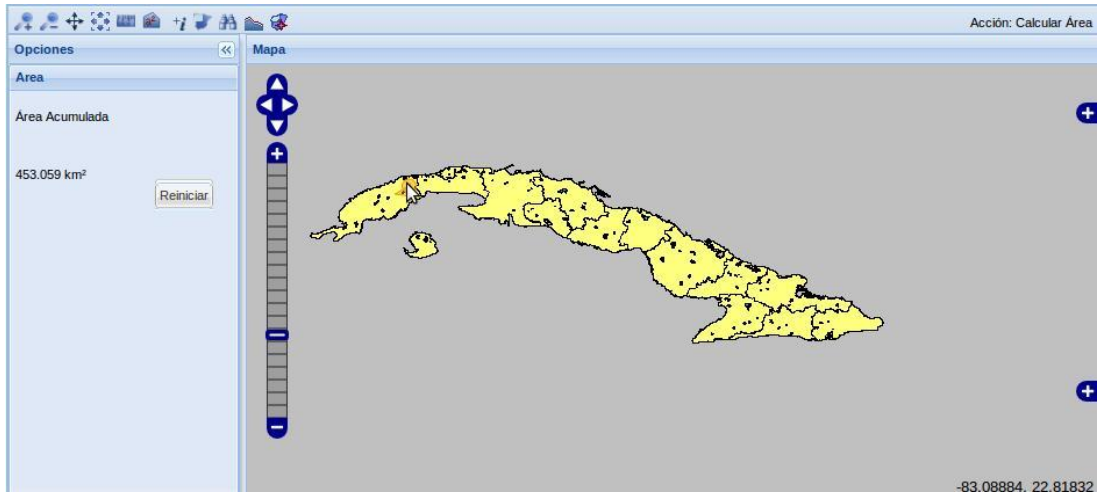
### Prototipo de Interfaz



<b>Poscondiciones</b>	El sistema muestra el valor correspondiente a la distancia según la polilínea trazada por el usuario.
-----------------------	---

**Tabla 8. Calcular área**

<b>Caso de Uso:</b>	Calcular área
<b>Actores:</b>	Usuario
<b>Propósito</b>	Este caso de uso se lleva a cabo con el objetivo de calcular un área seleccionada en el mapa.
<b>Resumen:</b>	El caso de uso se inicia cuando el usuario selecciona el botón "Calcular área" y termina cuando el sistema muestra los valores correspondientes.
<b>Referencias</b>	RF 9

<b>Prioridad</b>	Crítico	
<b>Flujo Normal de Eventos</b>		
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>	
1. El usuario selecciona la opción “Calcular área”.	2. El sistema actualiza la barra de información.	
3. El usuario realiza el trazado de un polígono dando clic izquierdo y arrastrando el cursor y termina con un clic en el mapa cuando haya realizado el trazo deseado.	4. El sistema crea una capa temporal donde dibuja el polígono trazado por el usuario. El área se expresa en la unidad de medida que muestra la aplicación por defecto.	
	5. El sistema muestra en la barra de izquierda el área total seleccionada.	
<b>Prototipo de Interfaz</b>		
		
<b>Poscondiciones</b>	El sistema muestra el valor correspondiente al área según el polígono trazado por el usuario.	

### 3.5 Conclusiones parciales

Un mejor entendimiento de los casos de uso mediante una descripción detallada de cada uno de ellos y la definición de los requisitos tanto funcionales como no funcionales del sistema han permitido un entendimiento mucho más amplio del sistema. Además apoyándose en el modelo de dominio y el diagrama de caso de uso del sistema se han creado las bases necesarias para dar paso a la implementación del producto.



## Capítulo 4. Desarrollo de la solución y validación.

### 4.1 Introducción

A través de la realización de este capítulo se describe cómo se realizó la implementación de la aplicación en términos de componentes. Se describen algunos fragmentos de códigos, se exponen los principales resultados obtenidos a lo largo del trabajo, a la vez que se discute sobre la importancia y relevancia de los mismos.

### 4.2 Patrones de diseño

Un patrón define la solución a un problema determinado que pudiera presentarse comúnmente durante el desarrollo de un software. Brindando sugerencias de cómo podría ser solucionado aplicando el patrón en otros contextos.

- **Observer (Observador):** Define una dependencia del tipo uno-a-muchos entre objetos, de manera que cuando uno de los objetos cambia su estado, el observador se encarga de notificar este cambio a todos los otros dependientes. Se utiliza para actualizar listas mostradas que pueden cambiar dinámicamente.
- **Singleton (Solitario):** Asegura que una clase sólo tendrá una instancia y provee un punto global de acceso a la misma. De esta forma se controla el acceso a las clases. Se utiliza con el objetivo de crear el objeto “mapa” para luego utilizarlo sin la necesidad de construirlo nuevamente.
- **Command (Acción):** Permite solicitar una operación a un objeto sin conocer realmente el contenido de esta operación, ni el receptor real de la misma. Para ello se encapsula la petición como un objeto. Se utiliza en el proceso de petición mediante la Interfaz Gráfica de Usuario al sistema de una información cualquiera por un cliente.
- **Factory Method:** Define una interfaz para crear un objeto, pero deja que las subclases decidan qué clase instanciar, delega la instanciación de una clase a sus subclases.

### 4.2 Diagramas de clases de diseño

Los diagramas de clases del diseño presentados a continuación están basados en la implementación del sistema, debido a la complejidad y la extensión de la solución se decidió representar únicamente las clases y extensiones web que tuvieran que ver directamente con la construcción de la misma (Ver Figura 8).

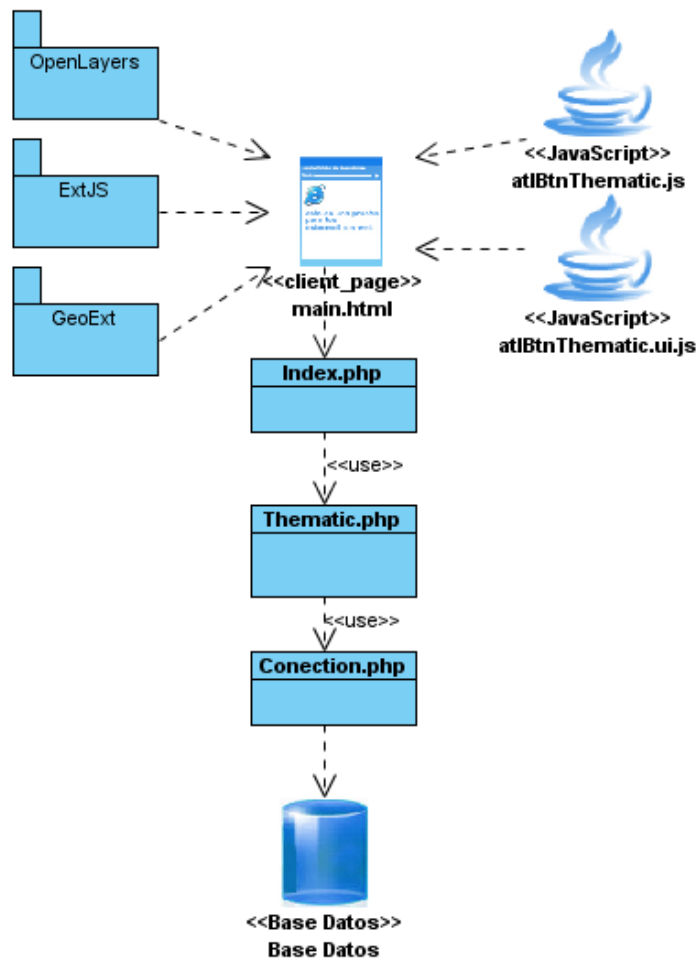


Figura 8. Diagrama de clases de diseño CU\_Personalizar mapa

#### 4.3 Diseño de la Base de Datos

La base de datos necesita de una definición de su estructura, de manera que permita almacenar datos, reconocer el contenido y recuperar la información. Para diseñar una base de datos se necesita seguir un conjunto de pasos que comienzan con definir las clases persistentes, luego refinarlas y clasificarlas junto con sus atributos, para más tarde realizar el diagrama de clases persistentes. Realizar el diagrama de transición de estado es el siguiente paso para diseñar la base de datos, y el último es la conversión de las clases al medio de almacenamiento (Falero, 2010) (Ver Figura 9).

### 4.3.1 Diagrama Entidad-Relación

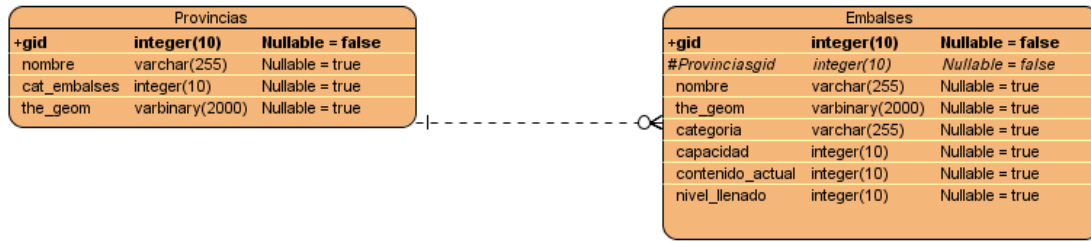


Figura 9. Diagrama Entidad-Relación

### 4.4 Modelo de despliegue

El modelo de despliegue describe la distribución física del sistema en términos de cómo se distribuye la funcionalidad entre los nodos de cómputo (Ver Figura 10).

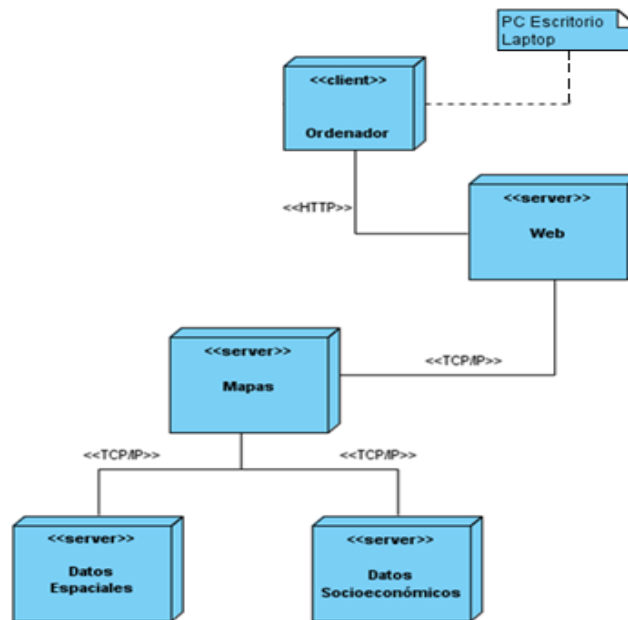


Figura 10. Diagrama de despliegue del sistema

### 4.5 Diagrama de componentes

El flujo de trabajo de implementación describe cómo los elementos del modelo del diseño se implementan en términos de componentes. Un diagrama de componentes muestra las organizaciones y dependencias lógicas entre los componentes del software.

A continuación se muestran los diagramas de componentes referentes a cada caso de uso del sistema:

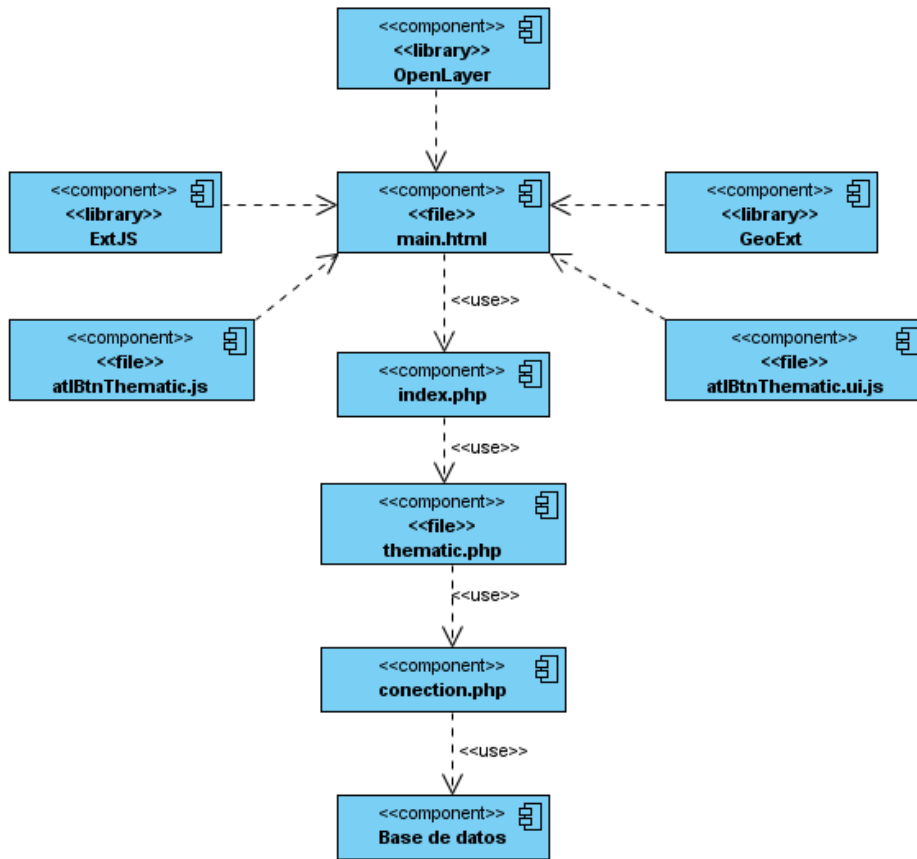


Figura 11. Diagrama de componentes CU Personalizar mapa

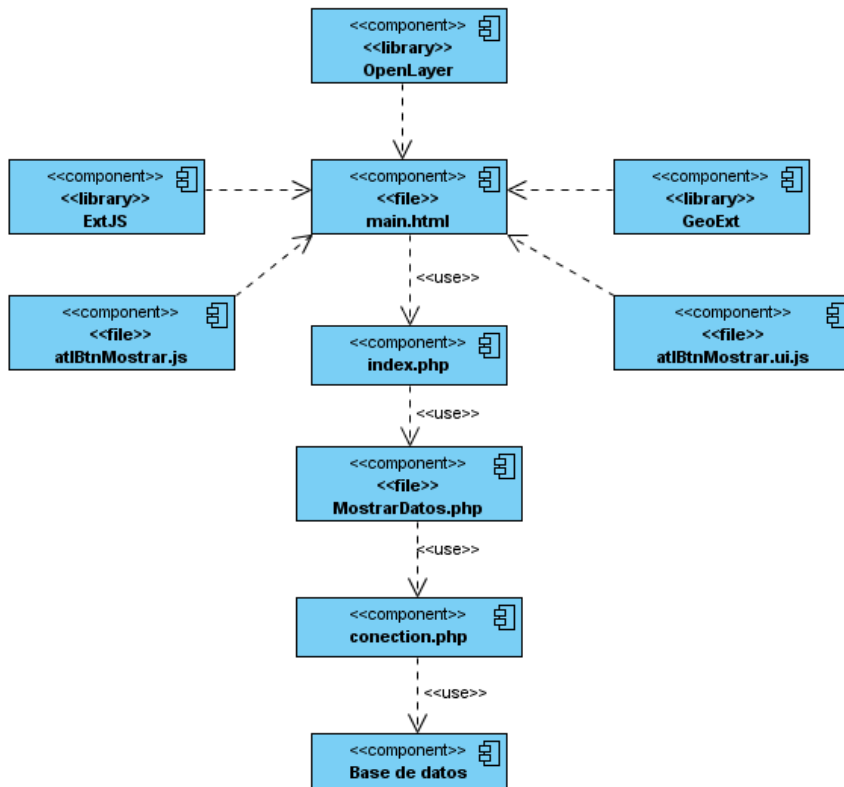


Figura 12. Diagrama de componentes CU Mostrar Datos

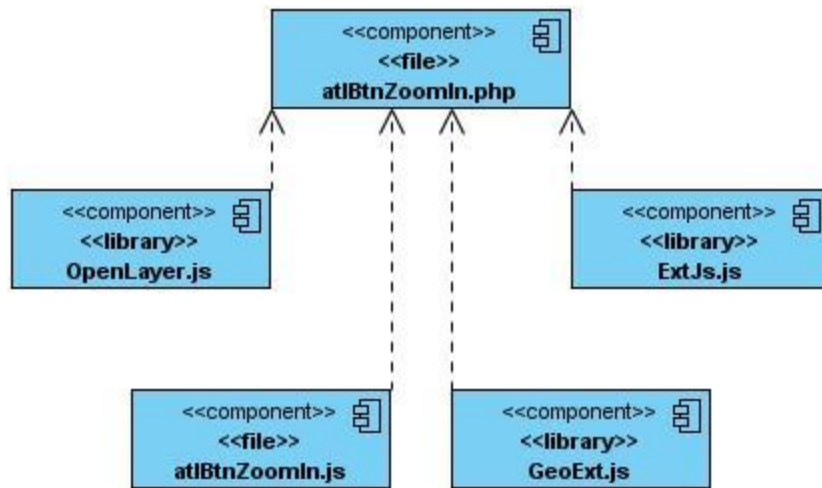


Figura 13. Diagrama de componentes CU Acercar

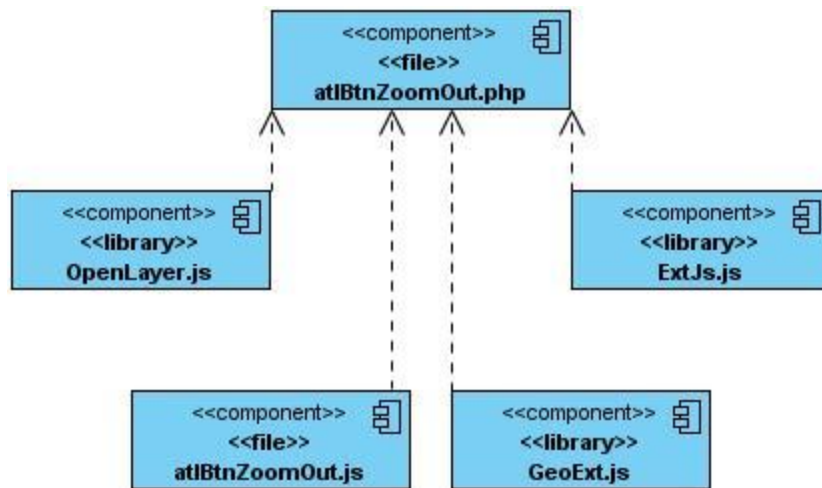


Figura 14. Diagrama de componentes CU Recentrar mapa

#### 4.6 Pruebas funcionales

Una vez generado el código de la aplicación es necesario realizar pruebas que determinen funcionalidad y calidad del sistema antes de ser liberado a los usuarios finales.

Las pruebas funcionales o de comportamiento están diseñadas para probar el funcionamiento de un sistema, ya que se no se enfoca en como se genera la respuesta sino en que sea el resultado esperado por el usuario.

Para realizar este tipo de pruebas es necesario usar el sistema como un usuario final pero más bien con una misión destructiva. El probador realiza las pruebas con el objetivo de encontrar alguna debilidad en el sistema. Cada error encontrado por el probador es un éxito y debe ser visto como tal.

Las pruebas funcionales miden todo el desarrollo de la aplicación, desde la petición realizada por un navegador hasta la respuesta enviada por el servidor. Verifican que todo funcione correctamente al pulsar sobre todos los enlaces y botones y obtener la respuesta deseada.

#### 4.6.1 Diseño de las pruebas que permiten validar la solución propuesta

##### Realizar navegación

###### 1. Descripción general

El caso de uso Realizar Navegación se inicia cuando el usuario desea mover, ampliar o recentrar 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 mapa resultante en la pantalla.

###### 2. Condición de ejecución

Ninguna.

**Tabla 9. Secciones a probar en el caso de uso Realizar Navegación**

Nombre de la sección	Escenarios de la sección	Descripción de la funcionalidad
SC 1: Acercar	EC 1.1: Acercar cuando hace clic en un punto del mapa.	El sistema actualiza la visualización del mapa.
SC 2: Alejar	EC 2.1: Alejar cuando hace clic en un punto del mapa.	El sistema actualiza la visualización del mapa.
SC 3: Recentrar mapa	EC 3.1: Recentrar mapa cuando hace clic en un punto del mapa.	El sistema actualiza la visualización del mapa.
SC 4: Mover Mapa.	EC 4.1: Mover Mapa cuando hace clic en un punto del mapa.	El sistema actualiza la visualización del mapa.

###### 3. Descripción de variable

No existen variables para este Caso de Prueba.

4. Matriz de datos.

**Tabla 10. Acercar**

<b>Escenario</b>	<b>Respuesta del sistema</b>	<b>Resultado de la prueba</b>	<b>Flujo Central</b>
C 1.1: Acercar cuando hace clic en un punto del mapa.	El sistema actualiza la visualización del mapa.	El sistema actualiza la visualización del mapa. Exitosa, cumplida.	<ol style="list-style-type: none"> <li>1. Seleccionar la opción Acercar.</li> <li>2. Dar clic en el mapa.</li> <li>3. El sistema aumentará el zoom, coge las coordenadas (x, y) del lugar donde se dio clic y se moverá al centro del mapa hacia ese punto.</li> </ol>

**Tabla 11. Alejar**

<b>Escenario</b>	<b>Respuesta del sistema</b>	<b>Resultado de la prueba</b>	<b>Flujo Central</b>
EC 1.1: Alejar cuando hace clic en un punto del mapa.	El sistema actualiza la visualización del mapa.	El sistema actualiza la visualización del mapa. Exitosa, cumplida.	<ol style="list-style-type: none"> <li>1. Seleccionar la opción Alejar.</li> <li>2. Dar clic en el mapa en la región que se desea visualizar.</li> <li>3. El sistema disminuirá el zoom, coge las coordenadas (x, y) del lugar donde se dio clic y se moverá al centro del mapa hacia ese punto.</li> </ol>

Tabla 12. Recentrar mapa

Escenario	Respuesta del sistema	Resultado de la prueba	Flujo Central
EC 1.1: Recentrar mapa cuando hace clic en un punto del mapa.	El sistema actualiza la visualización del mapa.	El sistema actualiza la visualización del mapa. Exitosa, cumplida.	<ol style="list-style-type: none"> <li>1. Seleccionar la opción Recentrar Mapa.</li> <li>2. Dar clic en el mapa en la región que se desea recentrar.</li> <li>3. El sistema obtiene las coordenadas del punto donde el usuario hizo clic y traslada al centro de la pantalla el punto seleccionado sin modificarse la escala del mapa.</li> </ol>

Tabla 13. Mover mapa

Escenario	Respuesta del sistema	Resultado de la prueba	Flujo Central
EC 1.1: Mover mapa cuando hace clic en un punto del mapa.	El sistema actualiza la visualización del mapa.	El sistema actualiza la visualización del mapa. Exitosa, cumplida.	<ol style="list-style-type: none"> <li>1. Seleccionar la opción Mover Mapa.</li> <li>2. Dar clic y arrastrar hasta obtener la región del mapa que se desea visualizar.</li> <li>3. El sistema calcula un <math>\Delta X</math> y un <math>\Delta Y</math> a partir del centro de pantalla y se obtiene un nuevo centro de pantalla a donde se movería el mapa.</li> </ol>



## Localizar por coordenadas

### 1. Descripción general

El caso de uso se inicia cuando el usuario desea localizar un objeto del mapa por sus coordenadas, introducidas por él y termina cuando el sistema muestra el objeto redimensionando el mapa a las coordenadas introducidas.

### 2. Condición de ejecución

Ninguna.

### 3. Descripción de variable

No existen variables para este Caso de Prueba.

**Tabla 14. Secciones a probar en el caso de uso Localizar por Coordenadas.**

Nombre de la sección	Escenarios de la sección	Descripción de la funcionalidad
C 1: Localizar por Coordenadas	EC 1.1: Localizador por Coordenadas completo.	El sistema pone un marcador y recentra el mapa en el punto creado a partir de las coordenadas introducidas por el usuario

**Tabla 15. Variables a probar**

Nombre de campo	Clasificación	Valor Nulo	Descripción
Longitud	Campo de texto	No	Se deben introducir datos y no dejar vacío el campo.
Latitud	Campo de texto	No	Se deben introducir datos y no dejar vacío el campo.

### 4. Matriz de datos.

**Tabla 16. Localizar por Coordenadas completo**

Escenario	Respuesta del sistema	Resultado de la prueba	Flujo Central
<p>EC 1.1: Localizador por Coordenadas completo.</p>	<p>El sistema pone un marcador y recentra el mapa a partir de las coordenadas introducidas por el usuario.</p>	<p>El sistema actualiza la visualización del mapa y pone un marcador. Exitosa, cumplida.</p>	<ol style="list-style-type: none"> <li>1. Seleccionar la opción Localizador por Coordenadas.</li> <li>2. El sistema muestra una ventana para insertar los valores correspondientes. Para los valores de las "Coordenadas se tienen dos cuadros de texto uno para la "Latitud" y otro para la "Longitud".</li> <li>3. Llenar los valores de cada uno de los campos.</li> <li>4. Dar clic en el botón "Aceptar".</li> </ol>

**Medir distancia**

1. Descripción general

El caso de uso se inicia cuando el usuario desea medir la distancia existente entre 2 o más puntos marcados en un mapa, y termina cuando el sistema muestra los valores correspondientes.

2. Condición de ejecución

Ninguno.

3. Descripción de variable

No existen variables para este Caso de Prueba.

**Tabla 17. Secciones a probar en el caso de uso Medir distancia**

Nombre de la sección	Escenarios de la sección	Descripción de la funcionalidad
SC 1: Medir distancia.	EC 1.1: Medir distancia.	El sistema muestra el valor correspondiente a la distancia según la polilínea trazada por el usuario.
	EC 1.2: Calcular área.	El sistema muestra el valor correspondiente al área trazada por el usuario.

4. Matriz de datos.

**Tabla 18. Medir distancia**

Escenario	Respuesta del sistema	Resultado de la prueba	Flujo Central
EC 1.1: Medir distancia.	El sistema muestra el valor correspondiente a la distancia según la polilínea trazada por el usuario.	El sistema muestra el resultado. Exitosa, cumplida.	<ol style="list-style-type: none"> <li>1. Seleccionar la opción Medir Distancia.</li> <li>2. Dibujar la polilínea.</li> <li>3. Mostrar la distancia acumulada en metros.</li> </ol>

**Tabla 19. Calcular área**

Escenario	Respuesta del sistema	Resultado de la prueba	Flujo Central
EC 1.1: Calcular área.	El sistema muestra el valor correspondiente al área según el polígono trazado por el usuario.	El sistema muestra el resultado. Exitosa, cumplida.	<ol style="list-style-type: none"> <li>1. Seleccionar la opción Calcular área.</li> <li>2. Dibujar el polígono.</li> <li>3. Mostrar el área acumulada en metros.</li> </ol>

#### 4.6.2. Pruebas de caja negra

Las pruebas de caja negra son las que se enfocan directamente en el exterior del sistema, sin importar el código, son pruebas funcionales en las que se trata de encontrar fallas en la interfaz, en la apariencia de los menús, entre otros. En estas pruebas no es necesario conocer la lógica del programa, sino únicamente la funcionalidad que debe realizar.

Para la realización de las pruebas utilizando las técnicas de caja negra se escoge el caso de uso: "Localizar por coordenadas". En la figura 15 se muestra la interfaz correspondiente al caso de uso.



Figura 15. Interfaz Gráfica Localizar por coordenadas.

El siguiente listado son las posibles variantes que pueden existir durante la localización de un punto.

1. Valor Latitud y Longitud correcto.
2. Valor Latitud y Longitud vacío.
3. Valor Latitud y Longitud con números no válidos.

Tabla 20. Caso de prueba # 1 utilizando técnicas de caja negra

<b>Caso de Uso:</b>	"Localizar por coordenadas"
<b>Caso de prueba:</b>	1

<b>Entrada:</b>	Valor para latitud 22.67 y valor para longitud -80.15
<b>Salida:</b>	Como los datos insertados en los campos son válidos, se localiza en el mapa poniendo un marcador sobre el punto seleccionado.

**Tabla 21. Caso de prueba # 2 utilizando técnicas de caja negra**

<b>Caso de Uso:</b>	“Localizar por coordenadas”
<b>Caso de prueba:</b>	2
<b>Entrada:</b>	Ambos campos vacíos
<b>Salida:</b>	Como no se han insertado datos en los campos sale una ventana de error, diciendo que debe insertar datos válidos.

**Tabla 22. Caso de prueba # 3 utilizando técnicas de caja negra**

<b>Caso de Uso:</b>	“Localizar por coordenadas”
<b>Caso de prueba:</b>	3
<b>Entrada:</b>	Valor de latitud 25.4 y valor de longitud -70.6
<b>Salida:</b>	Como los datos insertados están fuera del rango, los campos se ponen en color rojo y al dar aceptar el sistema muestra un mensaje de error.

Durante el desarrollo de las pruebas funcionales realizadas al sistema se detectaron 7 no conformidades las cuales no representaban errores graves, aunque si afectaban el buen funcionamiento de la aplicación. Las dificultades detectadas fueron resultas en el período de los 4 días posteriores al desarrollo de las pruebas.

#### **4.7 Conclusiones parciales**

Los diagramas de componentes muestran de una forma mucho más detallada el funcionamiento del sistema. La realización de las pruebas funcionales y los excelentes resultados obtenidos validan la mayoría de las funcionalidades de la aplicación aunque el mismo aún se encuentra en fase de pruebas.

## Conclusiones Generales

Al concluir la investigación se le dio cumplimiento al objetivo general de la misma y se comprobó la idea a defender como solución al problema científico. Como resultado se obtuvo una aplicación SIG para el manejo de los recursos hidráulicos cubanos.

Luego del desarrollo de la investigación:

- Se logró ampliar los conocimientos acerca del objeto de estudio, así como el entorno que rodeaba el problema en cuestión.
- Se fueron dando cumplimiento a las tareas investigativas trazadas al comienzo de la misma, logrando una mayor organización durante el desarrollo del trabajo.
- Se logró el desarrollo de un sistema informático capaz de dar respuesta a las necesidades identificadas.
- Se logró que el sistema cumpla con los requisitos planteados para su desarrollo.
- Se generó una documentación técnica asociada al sistema y a los procesos que se llevaron a cabo durante su desarrollo.

El sistema, constituye un aporte importante para el INRH, brindando además un conjunto de funcionalidades para el manejo y el análisis de los datos, tales como, medir distancia en el mapa, calcular el área, mostrar los datos de los embalses deseados, localizar por coordenadas un lugar en específico, la personalización del mapa, además de contar con una interfaz más llamativa e intuitiva.

## Recomendaciones

Se recomienda para posteriores actualizaciones de la aplicación:

- Mejorar la base de datos del producto la cual no tiene los datos socioeconómicos necesarios para llevar a cabo el despliegue de la aplicación y presenta algunas incoherencias en los datos geográficos.
- Usar TileCache para disminuir el tiempo de respuesta del sistema, ante las peticiones de los usuarios.
- Implementar una ayuda para facilitar a los usuarios la interacción con el sistema.

## Bibliografía Referenciada

- Real Academia. 2009.** Diccionario de la Real Academia de la Lengua Española. *Diccionario de la Real Academia de la Lengua Española*. [En línea]. [Citado el: 18 de enero de 2011.] [http:// www.rae.es](http://www.rae.es).
- Alfaro, F.M. 2005.** *Tecnología Cliente - Servidor*.
- Alfaro, Mg. Emigdio. 2007.** *Desarrollo de Sistemas Web*.
- Alonso, Evelín. 2009.** *Herramientas CASE para el proceso de desarrollo de Software*.
- Alvarez, Miguel Angel. 2007.** *Python*.
- . 2004.** *Qué es la Programacion Orientada a Objeto*. [En línea] 24 de Julio de 2004. [Citado el: 3 de Marzo de 2010.] <http://www.desarrolloweb.com/articulos/499.php>.
- Aroche, Christian Van Der Henst y Stephanie Falla. 2003.** *Maestros del Web. Evaluando Zend Studio*. [En línea] 3 de noviembre de 2003. [Citado el: 18 de febrero de 2011.] <http://www.maestrosdelweb.com/editorial/zendstudio/>.
- ASF. 2010.** Apache - Web Server. [En línea] 05 de 01 de 2010. [Citado el: 10 de 02 de 2010.] <http://www.apache.org/>.
- Autoridad Aeronautica Civil.** Implementacion del sistema de facturación y cobro FYC4. [En línea] [Citado el: 12 de 10 de 2009.] [http://www.aeronautica.gob.pa/index.php?option=com\\_content&task=view&id=522&Itemid=2](http://www.aeronautica.gob.pa/index.php?option=com_content&task=view&id=522&Itemid=2).
- Baires, IT - Desarrollo de Software. 2010.** Bairesit, Information Technology Engineering. [En línea] 7 de diciembre de 2010. [Citado el: 7 de febrero de 2011.] <http://www.bairesit.com/2010/12/lanzamiento-de-nueva-version-de-sencha-ext-js/>.
- Barreda, Jose. 2007.** *OpenLayer*. Perú.
- Ben-Kiki, Oren, Evans, Clark y Ingerson, Brian. 2006.** *YAML Ain't Markup Language*.
- Bocher, Steiniger. 2009.** *Capaware: Plataforma de desarrollo de software, 3ra Jornadas de SIG Libre*. Girona.
- Brenes, Carlos. 2008.** *Sistemas de información geográfica*.
- Burch, John G. 1992.** *Diseño de sistemas de información : teoría y práctica*.
- Burrough, P. A. 1986.** *Principles of Geographic Information Systems for Land Resource Assessment*.
- Castillo, Alvaro del. 2007.** *Manual PHP*.
- Codima, Marc. 2004.** *Plataformas para desarrollo de aplicaciones SIT en Internet por personal no especializado*. Brasil.
- CommTrack.** Sistema de Facturación CommTrack. [En línea] [Citado el: 10 de 10 de 2009.] [http://www.techmarksoftware.com/media/tsi\\_pdf/cbs4\\_espanol.pdf](http://www.techmarksoftware.com/media/tsi_pdf/cbs4_espanol.pdf).



**Community, Erwin. 2010.** Erwin. *ERwin® Data Modeler*. [En línea] 2010. [Citado el: 27 de marzo de 2011.] [www.erwin.com](http://www.erwin.com).

**Connalen, J. 1999.** *Building Web Applications with UML*. s.l. : Adison Wesley, 1999.

**Copca, Agustín. 2004.** *Oracle*.

**Corbyn, Chris. 2010.** SwiftMailer. [En línea] 12 de Febrero de 2010. [Citado el: 3 de Marzo de 2010.] <http://www.swiftmailer.org/>.

**Enmanuel. 2011.** *Sistemas de Información Geográfica*.

**ExtJS development Group. 2010.** ExtJS Javascript Framework and RIA Plataform. [En línea] 03 de Marzo de 2010. [Citado el: 08 de Marzo de 2010.] <http://www.extjs.com/>.

**ExtJS, development Group. 2010.** ExtJS Javascript Framework and RIA Plataform. *ExtJS Javascript Framework and RIA Plataform*. [En línea] 03 de marzo de 2010. [Citado el: 19 de febrero de 2011.] <http://www.extjs.com/>.

**Falero, Yadira Hernández Inza & Héctor René Sánchez. 2010.** *Gestión de Nomencladores para el Programa Nacional de Informatización del Conocimiento Geológico*.

**Florencia. 2010.** Definición ABC una guía única en la red. *Definición ABC una guía única en la red*. [En línea] 23 de junio de 2010. [Citado el: 8 de febrero de 2011.] <http://www.definicionabc.com/general/pozo.php>.

**Foundation, Apache Software. 2011.** The Apache Software Foundation. *The Apache Software Foundation*. [En línea] 2011. [Citado el: 18 de febrero de 2011.] <http://www.apache.org/>.

**Frederick, Steve. 2008.** *Learning ExtJS*. ISBN 978-1-847195 .

**Garrido, John. 2004.** *Arquitectura y diseño de sistemas web modernos*.

**GeoExt, Comunidad. 2010.** GeoExt. *GeoExt*. [En línea] 2010. [Citado el: 21 de febrero de 2011.] <http://www.geoext.org/>.

**Gironés, Jordi. 2007.** *Puro Software*. Argentina.

**González, Pedro. 2010.** *Implementación de los módulos de Facturación y Cobro del Sistema de Facturación y Cobro para la Empresa de Gas Manufacturado de Ciudad de la Habana*. Habana.

**Group, PHP. 2010.** *PHP Hipertext Preprocesor*.

**Guardado, Iván. 2010.** *Doctrine*.

**Guglielmetti, Marcos. 2005.** *Definición de Sistema web*.

**Jacobson, Ivar. 2004.** *El Proceso Unificado de Desarrollo de Software*. s.l. : Addison Wesley.

**Lellelid, Hans. 2005.** *Propel - Guía de usuario*.

**Lobo, Armando Robert. 2006.** *Tutorial Básico de ExtJS*.

**Marley, Jimi. 2011.** Programación en castellano. *¿Por qué elegir PHP?* [En línea] 2011. [Citado el: 21 de febrero de 2011.] [http://www.programacion.com/articulo/por\\_que\\_elegir\\_php\\_143](http://www.programacion.com/articulo/por_que_elegir_php_143).

**Martín, Manuel. 2008.** *Manual PostGIS*.

**Martínez, Jose. 2010.** *PostGIS. Bases de datos espaciales*. España.

**Maturana, Sergio. 2009.** *Gescoop*.

**Medina, José. 2006.** *Administración SGBD PostgreSQL*.

**Microsoft. 2011.** Microsoft TechNet. *Windows Server TechCenter*. [En línea] 2011. [Citado el: 26 de marzo de 2011.] <http://technet.microsoft.com/es-es/library/cc730981%28WS.10%29.aspx>.

—. **2009.** MSDN Home Page. [En línea] <http://msdn.microsoft.com/es-es/library/aa291591%28VS.71%29.aspx>. [En línea]

**Minas, Facultad de ingeniería de. 2009.** *Manuales ArcView*. Piura.

**Monsalve, Carmona. 2009.** *Sistemas de Información Geográfica*.

**Montalvo, Carlos. 2011.** *TCPDF*. Lima. Perú.

**Montesino, Noraisi. 2009.** *Frameworks*. MSDN Home Page. [En línea] <http://msdn.microsoft.com/es-es/library/aa291591%28VS.71%29.aspx>. [En línea]

**NCGIA, National Center for Geographic Information and Analysis. 1990.**

**Openbravo, S.L.U. 2010.** OpenBravo ERP. [En línea] 2010. [Citado el: 05 de Marzo de 2010.] <http://www.openbravo.com/>.

**OpenLayers, Group. 2009.** OpenLayers. *OpenLayers*. [En línea] 2009. [Citado el: 21 de enero de 2011.] <http://www.openlayers.org/>.

**Oracle, Inc. 2010.** MySQL. *Panorámica del sistema de gestión de base de datos MySQL*. [En línea] 2010. [Citado el: 27 de marzo de 2011.] <http://dev.mysql.com/doc/refman/5.0/es/what-is.html>.

**Oviedo, Iván. 2006.** *Introducción al Agile UP*. Costa Rica.

**Palacio, Juan. 2006.** *El modelo Scrum*.

**Park, Amelia Industrial. 2008.** *GSInnova*. Puerto Rico : s.n.,

**Penadés, María. 2004.** *Métodologías ágiles para el desarrollo de software: eXtreme Programming (XP)*. s.l. : Universidad Politécnica de Valencia.

**PHP Group. 2010.** PHP Hipertext Preprocesor. [En línea] 17 de Marzo de 2010. [Citado el: 17 de Marzo de 2010.] <http://www.php.net/>.

**php.net. 2010.** *PHP Development*. s.l. : Philip Olson.

**Pino, José. 2010.** Servidores web más usados. [En línea] 30 de julio de 2010. [Citado el: 31 de marzo de 2011.] <http://lopezpino.es/2010/07/30/servidores-web-mas-usados/>. *Plataforma Informática*.

- PostgreSQL Foundation. 2009.** PostgreSQL - Database Server. [En línea] 15 de 12 de 2009. <http://www.postgresql.org/>.
- Potencier, Fabien. 2010.** *El Tutorial Jobbet*.
- **2010** Practical Symfony. *Practical Symfony*. [En línea] [Citado el: 21 de febrero de 2011.] [http://www.symfony-project.org/jobeeet/1\\_4/Propel/es/04](http://www.symfony-project.org/jobeeet/1_4/Propel/es/04).
- **2009.** *Symfony, guía definitiva*.
- Prat, Noraisi. 2009.** *Diseño de una Arquitectura de Software para el Sistema de Facturación y Cobro de la Empresa de Gas Manufacturado*.
- Pressman, R.S. 2000.** *Ingeniería del Software. Un enfoque práctico*.
- Process®, IBM® Rational Unified. 2006.** *Rational Software Architect, Ayuda de RUP*. s.l. : IBM Corp.
- Puyol, Dr. Agustín. 2004.** *Precipitación y recursos hidráulicos en México*.
- Rico, Gabriel. 2006.** Topografía Global. *Topografía Global*. [En línea] 2006. [Citado el: 17 de 11 de 2010.] <http://www.topografiaglobal.com.ar/archivos/teoria/sig.html>.
- Rumbaugh, G. Booch. 2000.** *El Lenguaje Unificado de Modelado. Manual de Referencia*.
- Sánchez, Darlenis. 2007.** *Sistema de gestión para el Programa Nacional de Grupos Electrogenos DENYO*. Ciudad de la Habana.
- Sendra, Bosque. 1992.** *Sistemas de Información Geográfica*. Madrid : Rialp, 1992.
- Sensio Lab. 2009.** *Askeet - symfony*.
- **2009.** *More with symfony*.
- **2009.** *Practical symfony "Jobeeet" + Propel*.
- **2009.** *symfony Forms in Action*.
- **2009.** *The symfony Cookbook*.
- **2009.** *The symfony Reference Book*.
- Sensio Labs. 2010.** Symfony | Web PHP Framework. [En línea] 10 de Marzo de 2010. [Citado el: 12 de Marzo de 2010.] <http://www.symfony-project.org/>.
- Thayer, Dorfman. 1990.** *"Standards, Guidelines and Examples on System and Software Requirements Engineering"*. s.l. : IEEE Computer Society Press.
- Valdés, Damián. 2008.** Noticias de Informática. *Editores web que facilitan tu trabajo*. [En línea] 2008. [Citado el: 25 de enero de 2011.] <http://www.radiocaribe.co.cu/secundaria/informatica/342.htm>.
- Varios.** Sensagent. [En línea] [Citado el: 24 de noviembre de 2010.] <http://diccionario.sensagent.com/presa/es-es/>.
- Varona, Helimay. 2009.** *Análisis y diseño del sistema de facturación y cobro*. Ciudad de la Habana.

**Visual Paradigm Design Group. 2009.** Visual Paradigm. [En línea] 12 de Noviembre de 2009. [Citado el: 18 de 01 de 2010.] <http://www.visual-paradigm.com/>.

**Wordware Publishing, Inc. 2001.** *Advanced Javascript*. ISBN 1-55622-852-X.

**Yagüez, Julio. 2002.** Instituto Nacional de tecnología Agropecuaria. [En línea] 2002. [Citado el: 20 de noviembre de 2010.] [http://www.inta.gov.ar/barrow/info/documentos/SIG/que\\_es\\_sig.htm](http://www.inta.gov.ar/barrow/info/documentos/SIG/que_es_sig.htm).

**Pantoja, Yoenis. 2009.** *Módulo de análisis de la Plataforma GENESIG*. Habana.

**Zend GDE. 2010.** Zend Studio Integration Development Enviroment. [En línea] 22 de Febrero de 2010. [Citado el: 25 de Febrero de 2010.] <http://www.zend.com/>.