

Universidad de las Ciencias Informáticas

“Facultad 6”



Universidad de las Ciencias
Informáticas

Título: “Plataforma de Servicios de Mapas para la georreferenciación de objetivos socioeconómicos de Universidad de las Ciencias Informáticas”

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

Autor: “Antonio Gilbert Paneque Corchete”

Tutor: “Ing. Yoenis Pantoja Zaldívar”

“Ciudad de la Habana, Junio 2011”

“Año 53 de la Revolución”

DECLARACIÓN DE AUTORÍA

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

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

Antonio Gilbert Paneque Corchete

Firma del Autor

Ing. Yoenis Pantoja Zaldívar

Firma del Tutor

DATOS DE CONTACTO

Tutor: Ing. Yoenis Pantoja Zaldívar **E-mail:** ypantojaz@uci.cu

- ❖ Ingeniero en Informática, Universidad de Holguín, 2006.
- ❖ Jefe de Departamento de Geoinformática.
- ❖ Centro GEYSED.

DEDICATORIA

Dedico esta tesis a mi mamá, por todo el sacrificio, el apoyo y el amor que me ha dado a lo largo de mi vida.

A mi papá, que a pesar de que la vida no le concedió el sueño de verme entrar a la Universidad gran parte de lo que soy es gracias a él. Solo espero que esté donde esté pueda verme como su Ingeniero.

En general a toda mi familia y las personas que confiaron en mí y que se preocuparon por mí todos estos años.

AGRADECIMIENTOS

A mi mamá, por ser lo más grande que tengo y por su eterna preocupación por mí.

A mi papá, por ser mi guía, por ser un ejemplo y por haber hecho de mí un hombre mejor.

A mi tía Leslie y a mi tío Hermides que son mis segundos padres, por todo el apoyo y el amor que me han dado.

A mi tío Ruby y mi tía Iliana, por su incondicional ayuda y por todo el amor que me han dado en estos 5 años.

A mi tía Daisy por siempre estar pendiente de todo lo que me puede hacer falta, por todo lo que ha hecho por mí y por todo su amor.

A mi tía Hilde por toda la ayuda que me ha dado y por todo el amor de estos 5 años.

A mi tía Nene, a mi tía Xiomara, a mi tía Aní, a mi tía Chini, a mi tía Gladis, a mi prima Arelis y a muchos más que no me alcanza la hoja para escribir, por favor no se sientan dolidos. En general a toda mi familia por preocuparse por mí en todo este tiempo y por todo su cariño.

A todas mis amistades, Kenier, Yaumara, Carlos, Daer, Migue... en general a todos mis amigos de Buenaventura y los amigos de la UCI con los que he compartido todo este tiempo y que con su amistan han cambiado mi vida, por todo su apoyo y por estar siempre pendientes de mí.

A mi novia, por sus consejos y su preocupación.

A mi tutor Yoenis, porque fue el de la idea de mi tesis, por confiar en mí y por toda la ayuda que me ha dado.

A la Revolución, por haberme dado la posibilidad de ser parte de esta increíble Universidad que es la UCI.

RESUMEN

En los momentos actuales existe la tendencia a que la información geográfica sea manejada digitalmente y de esta forma facilitar la interacción con la información y la toma de decisiones. Con el surgimiento de los Sistemas de Información Geográfica, el proceso de manejar la información geográfica se puede personalizar para cualquier ambiente. Los Servicios Web también son usados para facilitar el intercambio de este tipo de información de una forma estandarizada y de esta manera poder usarse en aplicaciones personalizadas por el cliente.

La UCI como universidad de avanzada, encomendó al centro de GEYSED el desarrollo de un sistema informático que sea capaz de prestar un paquete de Servicios Web con la información geográfica disponible. Este sistema será capaz de brindar mapas en formato digital a la comunidad universitaria de forma estandarizada, también facilitará el uso de la información geoespacial disponible mediante funcionalidades de búsqueda y localización sobre el mapa digital de la institución.

PALABRAS CLAVE

Servicios Web, Geográfica, Universidad de las Ciencias Informáticas (UCI), Geoinformática y Señales Digitales (GEYSED), Sistemas de Información Geográfica.

TABLA DE CONTENIDOS

<i>Introducción</i>	1
CAPÍTULO 1: “Fundamentación Teórica”	5
1.1. <i>Introducción</i>	5
1.2. <i>Conceptos asociados al dominio del problema</i>	5
1.3. <i>Objeto de Estudio</i>	9
1.3.1. <i>Descripción General</i>	10
1.3.2. <i>Los Sistemas de Información Geográfica. Impacto en la sociedad</i> ...	10
1.3.3. <i>Componentes de un SIG</i>	11
1.3.4. <i>Servicios Web</i>	12
1.3.5. <i>Servicios de Mapas</i>	13
1.3.6. <i>UDDI</i>	14
1.4. <i>Descripción actual del dominio del problema</i>	14
1.5. <i>Situación Problemática</i>	14
1.6. <i>Análisis de otras soluciones existentes</i>	15
1.6.1. <i>Google Maps</i>	15
1.6.1.1. <i>Funciones de Google Maps</i>	16
1.6.2. <i>Infraestructuras de Datos Espaciales de la República de Cuba</i>	16
1.6.3. <i>Servicio Regional de Mapas del Programa GeoSUR</i>	17
1.7. <i>Conclusiones parciales</i>	17
CAPÍTULO 2:” Herramientas y tecnologías”	19
2.1. <i>Introducción</i>	19

2.2.	<i>Fundamentación de las tecnologías utilizadas.</i>	19
2.2.1.	<i>PHP.</i>	20
2.2.1.1.	<i>Características de PHP.</i>	20
2.2.2.	<i>OpenLayers.</i>	20
2.2.2.1.	<i>Características de OpenLayers.</i>	21
2.2.3.	<i>XML.</i>	21
2.2.3.1.	<i>Características de XML.</i>	21
2.2.4.	<i>Metodología de Desarrollo.</i>	22
2.2.4.1.	<i>Proceso Unificado de Desarrollo (RUP).</i>	22
2.2.4.2.	<i>Características de RUP.</i>	24
2.2.5.	<i>El Lenguaje Unificado de Modelado (UML) como soporte de la modelación de la solución propuesta.</i>	25
2.2.5.1.	<i>Características del UML.</i>	25
2.2.6.	<i>Visual Paradigm.</i>	26
2.2.5.1	<i>Características de Visual Paradigm.</i>	26
2.2.7.	<i>Servidor Web Apache.</i>	27
2.2.7.1.	<i>Características de Apache.</i>	27
2.2.8.	<i>Servidor de mapas MapServer.</i>	27
2.2.8.1.	<i>Características de MapServer.</i>	28
2.2.9.	<i>Entornos de Desarrollo NetBeans.</i>	28
2.3.	<i>Conclusiones.</i>	29
<i>Capítulo 3 “Presentación de la solución propuesta”</i>		30
3.1.	<i>Introducción.</i>	30

3.2.	<i>Modelo de Dominio.....</i>	30
3.2.1.	<i>Diagrama de clases del Modelo de Dominio.</i>	31
3.2.2.	<i>Glosario de Términos del Dominio.</i>	31
3.3.	<i>Levantamiento de requisitos.....</i>	33
3.3.1.	<i>Requerimientos Funcionales.</i>	33
3.3.2.	<i>Requerimientos No Funcionales.</i>	34
3.4.	<i>Descripción de los actores.</i>	36
3.5.	<i>Diagrama de Casos de Uso del Sistema.....</i>	36
3.5.1.	<i>Descripción de los Casos de Uso del Sistema.</i>	37
3.5.1.1.	<i>Descripción Textual del Caso de Uso Devolver Mapa.....</i>	37
3.5.1.2.	<i>Descripción Textual del Caso de Uso Localizar Objetivo.</i>	40
3.5.1.3.	<i>Descripción Textual del Caso de Uso Localizar Edificio Residencial.</i>	41
3.5.1.4.	<i>Descripción Textual del Caso de Uso Localizar Computadora.....</i>	43
3.5.1.5.	<i>Descripción Textual del Caso de Uso Localizar Persona.</i>	44
3.6.	<i>Conclusiones.....</i>	48
CAPÍTULO 4: “Construcción de la solución propuesta”		49
4.1.	<i>Introducción.....</i>	49
4.2.	<i>Descripción de la Arquitectura.....</i>	49
4.3.	<i>Patrones de diseño.....</i>	50
4.3.1.	<i>Patrones para asignar Responsabilidades (GRASP).</i>	50
4.3.2.	<i>Patrón Singleton.....</i>	53
4.4.	<i>Diseño del Sistema.....</i>	53

4.4.1.	<i>Diseño de la Base de Datos.</i>	54
4.5.	<i>Modelo de Implementación</i>	56
4.5.1.	<i>Diagrama de Componentes</i>	56
4.6.	<i>Diagrama de Despliegue.</i>	56
4.7.	<i>Validación de la solución propuesta.</i>	57
4.7.1.	<i>Diseño de prueba del Caso de Uso Localizar Persona.</i>	57
4.7.2.	<i>Pruebas de estrés y rendimiento</i>	63
4.8.	<i>Conclusiones</i>	63
	<i>Bibliografía</i>	67
	<i>GLOSARIO</i>	70

ÍNDICE DE TABLAS

<i>Tabla 1 Descripción de los actores.</i>	36
<i>Tabla 2 Descripción Textual del Caso de Uso Devolver Mapa.</i>	40
<i>Tabla 3 Descripción Textual del Caso de Uso Localizar Objetivo.</i>	41
<i>Tabla 4 Descripción Textual del Caso de Uso Localizar Edificio Residencial.</i>	43
<i>Tabla 5 Descripción Textual del Caso de Uso Localizar Computadora</i>	44
<i>Tabla 6 Descripción Textual del Caso de Uso Localizar Persona.</i>	48
<i>Tabla 7 Diseño de prueba del Caso de Uso Localizar Persona.</i>	58
<i>Tabla 8 Descripción de las variables.</i>	59
<i>Tabla 9 SC 1 Localizar Persona Por Solapín.</i>	60
<i>Tabla 10 SC 3 Localizar Persona Por Usuario.</i>	62
<i>Tabla 11 Pruebas de carga y estrés.</i>	63

ÍNDICE DE FIGURAS

<i>Figura 1 Descripción del acceso a datos espaciales por un usuario.</i>	<i>6</i>
<i>Figura 2 Componentes Básicos de un SIG.....</i>	<i>12</i>
<i>Figura 3 Imagen representativa de la estructura de RUP.....</i>	<i>23</i>
<i>Figura 4 Diagrama de Casos de Uso del Sistema.....</i>	<i>36</i>
<i>Figura 5 Diagrama de clases del diseño del Caso de Uso Localizar Persona.....</i>	<i>54</i>
<i>Figura 6 Modelo Entidad-Relación del SIG UCI Socioeconómico.</i>	<i>55</i>
<i>Figura 7 Modelo Entidad-Relación del SIG UCI.....</i>	<i>55</i>
<i>Figura 8 Vista General Del Diagrama de Componentes.....</i>	<i>56</i>
<i>Figura 9 Diagrama de Despliegue.</i>	<i>57</i>

INTRODUCCIÓN

Desde los inicios de la humanidad, el hombre antiguo se vio obligado a representar su manera de vivir y desarrollarse mediante bocetos en forma de dibujos en los que combinaba lugares de la geografía con situaciones de su hábitat; estos fueron los primeros mapas que se conocieron en la historia. En sus comienzos estos mapas eran representados de forma rudimentaria pero con el paso del tiempo estas técnicas se fueron perfeccionando y surgieron nuevos métodos para representarlos. En la actualidad se ha alcanzado un considerable desarrollo en lo relacionado con la representación de mapas y con la llegada de Internet y el avance de las Tecnologías de la Información y las Comunicaciones (TICs) se han logrado avances muy importantes. Las tendencias actuales a las que se exponen estas tecnologías están encaminadas a controlar o manejar las disímiles vías de información existentes en todos los sectores de la sociedad y la economía; datos geográficos que por su importancia y valor llegan a ser extensos, desorganizados y complejos en cuanto a su interpretación, por lo que la acción de gestionarlos digitalmente hoy en día se ha convertido en todo un reto para los desarrolladores de la rama de la Geomática.

Los conocidos servicios que brindan grandes compañías como Google, ESRI¹ y SuperMaps, líderes absolutos del compartimento de información geográfica sobre la Web, constituyen ejemplos claros de lo que se puede lograr con el buen uso de las nuevas tecnologías en función de las necesidades de los entes que desarrollan las ciencias geográficas.

Los Servicios Web surgieron por la necesidad de estandarizar la comunicación entre distintas plataformas y lenguajes de programación, permiten que servicios y software sin importar plataforma o geografía puedan ser combinados para cumplir un objetivo. El servicio Web Map Service (WMS), es una especificación de este tipo de tecnología, con el aporte de que produce mapas de datos referenciados espacialmente de forma dinámica a partir de información geográfica. Los mapas creados por WMS se generan normalmente en un formato de imagen como PNG, GIF o JPEG y constituyen componentes claves en la puesta en marcha de los llamados Sistemas de Información Geográfica (SIG) (**Arcancel, 2008**).

En nuestro país se han dado grandes pasos en este sentido, entidades como GEOCUBA, la Oficina Nacional de Hidrografía y Geodesia (ONHG) y el Instituto de Geografía Tropical (IGT) lideran los

¹ ESRI acrónimo en Inglés de Environmental Systems Research Institute.

desarrollos más notables de estas tecnologías con el objetivo de centralizar, estandarizar y publicar todo el patrimonio cartográfico de la isla en una infraestructura única, pero la diversidad de sistemas y de datos no ha posibilitado la fundación del mecanismo necesario para la compartimentación de servicios de mapas a toda la sociedad.

La Universidad de Ciencias Informáticas (UCI), fundada al calor de la Batalla de Ideas por el Comandante en Jefe Fidel Castro Ruz, emprendió la labor del desarrollo de software desde sus primeros años. Su organización productiva se destaca por la existencia de centros de desarrollo especializados que aprovechan la fuerza de trabajo que garantizan los más de 10 mil estudiantes y profesores que allí radican combinando los frentes de docencia, producción e investigación.

En la Facultad 6, se fundó a inicios del año 2010 el Centro de Geoinformática y Señales Digitales (GEYSED) que destina una de sus líneas a la producción de software y servicios para la Geomática. Uno de sus productos más significativos lo constituye el Sistema de Información Geográfica de la UCI (SIG-UCI), sistema ideado para la georreferenciación de los objetivos socioeconómicos del centro y la toma de decisiones sobre ellos mediante funcionalidades de representación, análisis temático y edición sobre el mapa digital de la instalación. Esta importante herramienta, sin embargo, no ha explotado suficientemente las ventajas que brinda la aplicación del concepto de Servicios de Mapas para ampliar su gama de funcionalidades y prestar mejores asistencias a los usuarios sobre la información geográfica disponible, de manera que éstos puedan utilizarlos en otras aplicaciones o subsistemas como complementos de mapificación o localización de cualquier recurso socioeconómico. Carece además de módulos o facilidades de uso para el consumo independiente de los datos espaciales a través de los protocolos estándares de intercambio de información entre sistemas.

Partiendo de la situación anteriormente expuesta, surge entonces como **problema** de la presente investigación: *La necesidad de proveer información geográfica a usuarios y sistemas de la Universidad de las Ciencias Informáticas para la georreferenciación de objetivos socioeconómicos* y como **objetivo general** se define: *Desarrollar una plataforma que brinde servicios de mapas a usuarios y sistemas de la Universidad de las Ciencias Informáticas.*

El **objeto de estudio** que enmarca la investigación es: los servicios de mapas sobre los estándares de Web Map Services (WMS), mientras que el **campo de acción** se define en los procesos de publicación de datos geográficos sobre servicios Web en la Universidad de las Ciencias Informáticas.

Para dar cumplimiento al objetivo planteado y en aras de dar solución al problema planteado, se definen las siguientes tareas de la investigación:

1. Caracterizar el estado del arte sobre servicios de mapas mediante las normativas de WMS y WFS.
2. Analizar los flujos arquitectónicos para la integración de los servicios de mapas con los sistemas actuales de la universidad que requieran su uso.
3. Identificar los requisitos que debe cumplir la plataforma de servicios de mapas para su futura implementación.
4. Vectorizar y migrar a formato espacial la información planimétrica existente en la universidad.
5. Implementar los servicios de localización de objetivos.
6. Realizar pruebas de servicios de mapas a usuarios.
7. Implementar los servicios de tematización.
8. Realizar pruebas de consumo de servicios de mapas a otros sistemas.
9. Validar los resultados del consumo de usuarios y aplicaciones mediante un sistema de prueba.
10. Desplegar el sistema en la infraestructura de servicios de la universidad.

En la investigación se recurrió al uso de métodos científicos tanto teóricos como empíricos para la validación teórica y conceptual de todo lo descrito. El método *Histórico-Lógico* se aplica para todo el trabajo recopilatorio sobre la existencia de alguna plataforma de servicios, la publicación de datos espaciales sobre WMS y su evolución en la era tecnológica. El método de *Análisis y Síntesis* se define con el objetivo de analizar el estado del arte sobre las tecnologías y herramientas para el desarrollo de WMS y las metodologías existentes para la construcción de los artefactos necesarios en la concesión ingenieril de este tipo de sistemas. El método de *Modelación* se emplea para mostrar los diferentes diagramas y componentes que se construyen como resultado del proceso de Ingeniería de Software.

La investigación se estructura en cuatro capítulos:

El Capítulo 1 contempla la fundamentación teórica en la cual se exponen algunos conceptos que contribuyen al mejor entendimiento del problema en cuestión, se recrean teóricamente los elementos del objeto de estudio y se referencian otras soluciones existentes. **El Capítulo 2** abarca el estudio de las tendencias y las tecnologías que se utilizan actualmente para el desarrollo de sistemas de servicios de mapas, se presentan las técnicas, herramientas y metodología que guiarán el proceso de implementación del producto. En el **Capítulo 3** se presenta la solución propuesta a partir de la descripción de un modelo de dominio, la descripción de los requisitos y se documentan las

generalidades de la construcción del sistema. Y en el **Capítulo 4** se validará la solución mediante pruebas de sistema que serán descritas a través de técnicas de caja negra, para evaluar el comportamiento del sistema bajo situaciones extremas se realizaran pruebas de carga y estrés.

CAPÍTULO 1: “Fundamentación Teórica”.

1.1. Introducción.

En este capítulo se abordan conceptos y aspectos fundamentales de los servicios de mapas y las tendencias actuales sobre los conceptos de sistemas de información geográfica, datos espaciales y plataformas. Se explica con profundidad el problema planteado y el objeto de estudio en el cual se enmarca la investigación.

1.2. Conceptos asociados al dominio del problema.

Definición de mapa:

Modelo gráfico de la superficie terrestre donde se representan objetos espaciales y sus propiedades métricas, topológicas y atributivas, un mapa puede ser analógico (impreso sobre papel, por ejemplo) o digital (codificado en cifras, almacenado en un ordenador y presentado en una pantalla), existen mapas métricos, diseñados para representar distancias, superficies o ángulos y mapa topológicos, diseñados para representar vecindad, inclusión, conectividad y orden en el contexto de los SIG, un mapa es la presentación de cualquier estructura de datos usada para reflejar cartográficamente una variable espaciales, nominal o cuantitativa, independientemente del modelo de datos utilizado que puede ser vectorial o ráster. **(Belmonte, Nuñez, 2009)**

Definición de Datos Espaciales

El punto de partida para la recopilación de datos es principalmente su definición, con ello se resuelven varias interrogantes que se producen al realizar la descripción de los datos básicos, para obtener cualquier información en el ámbito de los Sistemas de Información Geográfica y sus aplicaciones, entonces se **puede definir a los datos espaciales, como elementos que poseen posición, forma y atributos.**

Al nombrar posición relativa, se refiere a que la información posee como atributo su ubicación en algún sistema de coordenadas conocidos, estos dependen en el cual se encuentren, ya que pueden ser coordenadas Cartesianas o UTM (Universal Transverse Mercator).

En cuanto a la forma, existen tres tipos de ellas:

"**Los puntos** se encuentran determinados por las coordenadas terrestres de latitud y longitud, pudiendo relacionarse con individuos, empresas, ciudades, delitos cometidos o accidentes acontecidos"

"**Las líneas** son objetos abiertos que cubren una distancia dada y comunican varios puntos o nodos (dada la naturaleza esférica de la Tierra, las líneas son en realidad "arcos").

"**Los polígonos** son figuras planas conectadas por distintas líneas u objetos cerrados que cubren un área determinada, como es el caso de países, provincias, secciones censales o áreas comerciales".

En cuanto a los atributos, estos dependen del tipo de trabajo en el cual se requiere información, ya que las posibilidades de atributos son numerosas, con ello la cantidad de información que se adquiere, se puede actualizar diariamente, como por ejemplo, el estado de un camino, o de una carretera. (ENCINA, 2004)

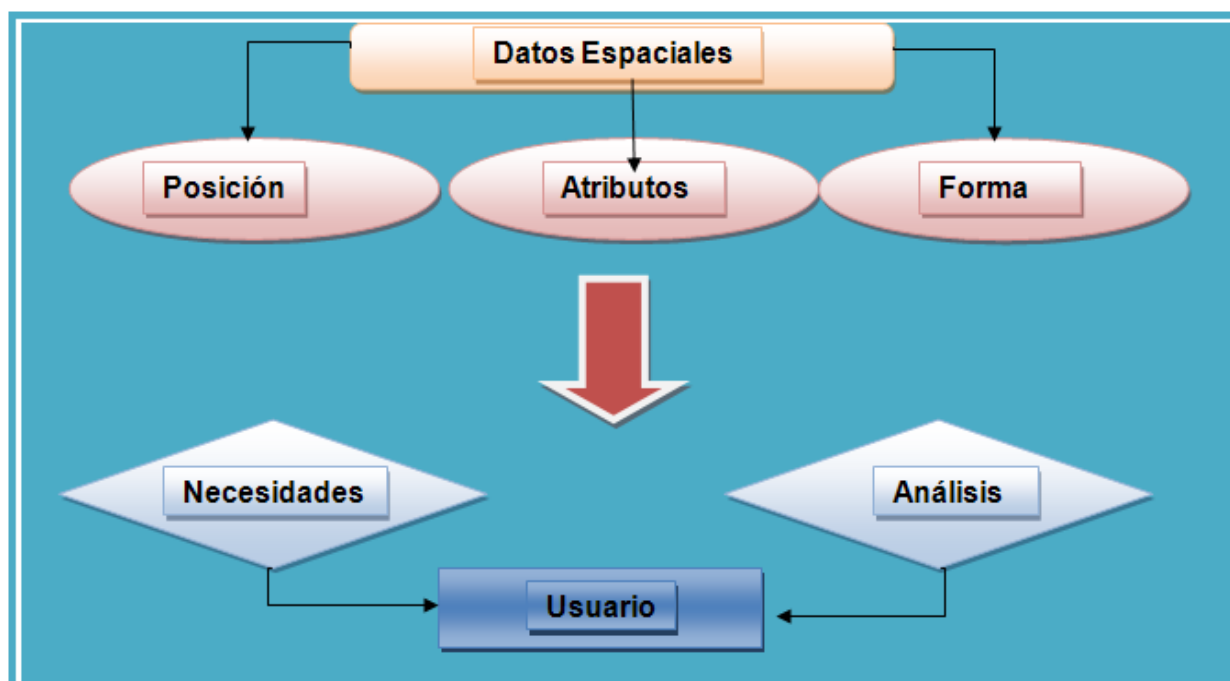


Figura 1 Descripción del acceso a datos espaciales por un usuario.

Definiciones de SIG:

Estos sistemas tienen varias definiciones:

- ❖ Un Sistema de Información Geográfica puede ser concebido como una especialización de un sistema de bases de datos, caracterizado por su capacidad de manejar datos geográficos, que

están georreferenciados y los cuales pueden ser visualizados como mapas. **(Webster and Bracken, 1992)**

- ❖ Es un conjunto de hardware, software y datos geográficos para capturar, manipular, analizar y mostrar información geográficamente referenciada. **(ESRI, 1995).**
- ❖ Un Sistema de Información Geográfica es una integración de hardware, software y datos geográficos diseñados para integrar, almacenar, editar, analizar, compartir y mostrar la información geográficamente referenciada con el objetivo de resolver problemas complejos de planificación y gestión. **(ECHEVERRIA, 2007).**

Según las definiciones expresadas anteriormente, se puede decir que los Sistemas de Información Geográfica son sistemas que permiten integrar, almacenar, editar, analizar, compartir y mostrar la información geográficamente referenciada, combinando el uso de hardware, software y procedimientos elaborados para facilitar la obtención, gestión, manipulación, análisis, modelado, representación y salida de datos espacialmente referenciados para resolver problemas complejos de planificación y gestión, con el fin de hacer más factible y puntual la toma de decisiones, dado el nivel de exactitud en cuanto a la ubicación geográfica de los objetos en el espacio. Son herramientas que permiten a los usuarios crear consultas interactivas, analizar la información espacial, editar datos, mapas y representar los resultados de todas estas operaciones. Es muy utilizado en muchos ámbitos, debido a que permite elaborar cartografía temática sobre cualquier aspecto ambiental, socioeconómico y políticos de la superficie terrestre.

Definición de Servicio Web:

Recurso de software que se ejecuta en un servidor Web remoto, en respuesta a la solicitud hecha por un cliente. Los servicios Web son equivalentes a cualquier aplicación que corre en un equipo local, sólo que la información necesaria para llevar a cabo una tarea específica es enviada al servidor y el resultado de esa tarea, devuelto al usuario, ambos en la forma de contenido Web. **(Puentes, 2005).**

Los Servicios Web surgieron por la necesidad de estandarizar la comunicación entre distintas plataformas y lenguajes de programación, básicamente es un conjunto de protocolos y estándares que sirven para intercambiar datos entre distintas aplicaciones de software desarrolladas en lenguajes de programación diferentes, y ejecutadas sobre cualquier plataforma, pueden utilizar los servicios web para intercambiar datos en cualquier tipo de redes como puede ser Internet. Las organizaciones

OASIS² y W3C³ son los comités responsables de la arquitectura y reglamentación de los servicios Web. Para la comunicación entre el servidor y el consumidor se implementan un conjunto de estándares como son el XML⁴ que es el formato estándar para el intercambio de datos y SOAP⁵ es el protocolo sobre el cual se establece el intercambio. El WSDL⁶ es el lenguaje de la interfaz pública para los servicios Web, es una descripción basada en XML de los requisitos funcionales necesarios para establecer una comunicación con los servicios Web. El protocolo UDDI es el encargado de publicar la información de los servicios Web, también permite comprobar qué el servicio web esté disponible.

Definición de Web Map Service (WMS):

Un servicio de mapas se define como un servicio que produce dinámicamente mapas georreferenciados a partir de información geográfica. **(Consortium, 2006)**

Open Geospatial Consortium, Inc. (OGC) es un consorcio internacional sin ánimo de lucro, formado por un conjunto de empresas, agencias gubernamentales y universidades, dedicado a desarrollar especificaciones de interfaces para promover y facilitar el uso global de la información espacial. De todos los servicios web de OGC, el más conocido y usado es el servicio de mapas o Web Map Service (WMS), un estándar internacional (ISO 19128) que define un servicio para producir mapas de forma dinámica a partir de datos georeferenciados. Según este estándar un mapa es una representación pictórica de la información geográfica en forma de archivo de imagen digital, apto para ser visualizado en la pantalla de un ordenador. Se definen tres operaciones básicas; una que permite conocer los metadatos del servicio, otra que permite obtener un mapa con un ámbito geográfico y dimensiones conocidas, y una última, opcional, que permite obtener información de los objetos particulares mostrados en el mapa. **(Consortium, 2006)**

Los Web Map Service (WMS) son servicios capaces de producir datos referenciados espacialmente, de forma dinámica a partir de información geográfica. Los mapas que son creados por WMS se generan normalmente en un formato de imagen como PNG, GIF o JPEG, y opcionalmente como gráficos vectoriales en formato SVG (Scalable Vector Graphics) o WebCGM (Web Computer Graphics Metafile).

Definición de Web Feature Service (WFS):

² **OASIS** acrónimo en Inglés de Organization for the Advancement of Structured Information Standards

³ **W3C** acrónimo en Inglés de World Wide Web Consortium

⁴ **XML** acrónimo en Inglés de Extensible Markup Language

⁵ **SOAP** acrónimo en Inglés de Simple Object Access Protocol

⁶ **WSDL** acrónimo en Inglés de Web Services Description Language

Los servicios conocidos como WFS ofrecen una interfaz de comunicación que permite interactuar con los mapas servidos por el estándar WMS, como por ejemplo, editar la imagen que nos ofrece el servicio WMS o analizar la imagen siguiendo criterios geográficos. Para realizar estas operaciones se utiliza el lenguaje GML que deriva del XML⁷, que es el estándar a través del que se transmiten las órdenes. **(Consortium, 2006)**

Definición de plataforma:

Es un término de carácter genérico que designa normalmente una arquitectura de hardware, aunque también se usa a veces para sistemas operativos o para el conjunto de ambos. **(Guglielmetti, 2005)**

La plataforma define un estándar alrededor el cual un sistema puede ser desarrollado. Una vez que la plataforma ha sido definida, se produce el software y el hardware apropiado para su uso. El término a menudo es usado como un sinónimo de sistema operativo. El término "plataforma cruzada" se refiere a la capacidad de software o hardware para correr de modo idéntico sobre plataformas diferentes. Muchas aplicaciones de Windows y Macintosh, por ejemplo, ahora producen archivos compatibles en lo binario, lo que quiere decir que los usuarios pueden cambiar de una plataforma a otra sin convertir sus datos a un nuevo formato.

Definición de Infraestructura de Datos Espaciales (IDE):

La Infraestructura de Datos Espaciales se define como "la acumulación importante de tecnologías, normas y planes institucionales que facilitan la disponibilidad y el acceso a datos espaciales". **(ENCINA, 2004)**

Una IDE es una iniciativa necesaria para almacenar, gestionar, permitir el acceso, entregar y fomentar la utilización de datos espaciales. Está completamente dirigida a facilitar y coordinar el hecho de compartir e intercambiar los datos espaciales; integra datos, metadatos, servicios e información de tipo geográfico para promover su uso. Es el conjunto "tecnologías, políticas, estándares y recursos humanos para adquirir, procesar, almacenar, distribuir y mejorar la utilización de la información geográfica". Al igual como las carreteras y autopistas facilitan el transporte vehicular, las IDE facilitan el transporte de información geoespacial.

1.3. Objeto de Estudio.

⁷ XML acrónimo en Inglés de Extensible Markup Language

1.3.1. Descripción General.

En este epígrafe se aborda más sobre lo que es un Sistema de Información Geográfica, cuales son los componentes que lo componen y en impacto que tienen en la sociedad. También se abunda sobre los servicios Web profundizando en los servicios de mapas. La UDDI como una especie de directorio en el cual podemos encontrar los servicios Web publicados y la descripción de sus servicios.

1.3.2. Los Sistemas de Información Geográfica. Impacto en la sociedad.

Gracias al creciente desarrollo de las tecnologías de la información y las comunicaciones se han logrado avances que hasta hace algunos años se creían imposibles para la Geomática. Debido al constante crecimiento de la información geográfica, manejarla con eficiencia es una prioridad. Los SIG surgen en el contexto de "sociedad de la información", en la que resulta esencial la disponibilidad rápida de la información. Con su surgimiento se le dieron solución al tratamiento manipulación y obtención de la información geográfica de manera más fácil y eficiente. Actualmente estos sistemas son usados en casi todas las esferas socioeconómicas, logrando con su vinculación un mayor control sobre los recursos.

El uso de los SIG puede ser muy importante para el desarrollo socioeconómico de nuestro país, utilizando aplicaciones SIG dirigidas a la gestión de servicios de impacto social, tales como centros escolares, hospitales, centros deportivos, culturales, lugares de concentración en casos de emergencias, centros de recreo, entre otros. Incrementando gradualmente su uso, en los diversos sectores económicos de Cuba se puede lograr una gestión más eficiente de los recursos con que contamos y así elevar la producción. También pueden ser muy útiles para la defensa del país suministran información sobre los terrenos existentes en una determinada zona y ayudando en la planificación de determinada maniobra o haciendo rápidos cálculos de distancias en el terreno.

La combinación entre los SIG y los GPS⁸ se está convirtiendo en una poderosa herramienta, aplicable a una cantidad innumerable de campos como son la industria, administración, protección civil, medio ambiente, agricultura, transporte y sistemas de navegación. Un buen diseño y una buena implementación de estos SIG aumentan la productividad al optimizar recursos, ya que permiten asignar de forma adecuada y precisa los centros de atención a usuarios cubriendo de forma eficiente la totalidad de la zona de influencia.

⁸ **GPS** acrónimo de Sistemas de Posicionamiento Global

1.3.3. Componentes de un SIG.

Los SIG están compuestos por determinados componentes (*Ver Figura 3*) para hacer posible su funcionamiento, a continuación se explicará la funcionalidad de cada uno de los mismos.

Hardware

Equipos: Computadoras encargadas de operar el SIG. En la actualidad se pueden ejecutar en una gran variedad de plataformas: servidores, computadoras de escritorio y portátiles. (ECHEVERRIA, 2007).

El hardware no es más que el equipo o la computadora que operan con el SIG que pueden estar centralizadas, en configuraciones individuales o en red. Los SIG requieren de hardware para cumplir con las necesidades planteadas.

Software

El software no es más que las herramientas y funcionalidades para almacenar, analizar y mostrar información geográfica. Entre otras cosas permiten la entrada y manipulación de datos geográficos, operación de búsqueda, visualización de estos datos entre otras los principales software de SIG son:

- 1- Sistemas de manejo de base de datos.
- 2- Interfaces Gráficas de Usuario para lograr mejor acceso a las herramientas.
- 3- Herramientas para captura y manejo de información geográfica.
- 4- Herramientas para soporte de consultas, análisis y visualización de datos geográficos.

Red

La comunicación a través de la red permite al sistema conectarse a equipos y a sistemas que ya existen, organizaciones públicas y privadas que distribuyen software de SIG gratis que serían de gran provecho al SIG, además que permite que los usuarios obtengan los beneficios de éstos desde la red.

Conjunto de Datos Geográficos

Uno de los componentes más importantes de los SIG son los datos ya que se requiere de muy buenos datos para que el SIG pueda resolver el problema por el cual fue creado y responder

satisfactoriamente las preguntas de la manera más exitosa posible, es decir, de la calidad de estos dependerán los resultados de un SIG.

Recursos Humanos

Es el personal que trabaja en el sistema, este debe de tener un conocimiento previo de los datos que se manejan. Los SIG no son nada sin el personal que se encarga de manejar el sistema y desarrollar planes de implementación de la misma.

Organización y Procedimientos

Todo SIG debería operar de acuerdo a una organización y unos métodos de funcionamiento acordados. (ECHEVERRIA, 2007)

Para que un SIG tenga una implementación exitosa debe tener un buen diseño y reglas de actividades definidas para su desarrollo, que son los métodos y prácticas o procedimientos propios de cada organización.



Figura 2 Componentes Básicos de un SIG

1.3.4. Servicios Web.

Antes del surgimiento de los Servicios Web se habían realizado intentos de crear estándares para lograr la comunicación entre diferentes plataformas, pero fracasaron o no tuvieron el suficiente éxito, algunos de ellos son DCOM (Distributed Component Object Model) y CORBA. Tuvieron problemas con

el uso de RPC ⁹ para realizar la comunicación entre diferentes nodos. Esto, además de presentar ciertos problemas de seguridad, tiene la desventaja de que su implementación en un ambiente como es Internet, es casi imposible (muchos firewalls bloquean este tipo de mensajes, lo que hace prácticamente imposible a dos computadoras conectadas por Internet comunicarse).

Ante la necesidad de crear un estándar para la comunicación entre diferentes plataformas y los fracasos anteriores de DCOM y CORBA. Surgen Los Servicios Web para estandarizar la comunicación entre distintas plataformas y lenguajes de programación. Es por esto que en 1999 se comenzó a plantear un nuevo estándar, el cual terminaría utilizando XML, SOAP, WSDL, y UDDI.

Estándares empleados:

- XML (Extensible Markup Language): Es el formato estándar usado para el intercambio de datos.
- SOAP (Simple Object Access Protocol) o XML-RPC (XML Remote Procedure Call): Protocolos sobre los que se establece el intercambio.
- WSDL (Web Services Description Language): Es el lenguaje de la interfaz pública para los servicios Web. Es una descripción basada en XML de los requisitos funcionales necesarios para establecer una comunicación con los servicios Web.
- UDDI (Universal Description, Discovery and Integration): Protocolo para publicar la información de los servicios Web.

1.3.5. Servicios de Mapas.

Un servicio de mapas se define como un servicio que produce dinámicamente mapas georreferenciados a partir de información geográfica.

Los Web Map Service (WMS) son servicios capaces de producir datos referenciados espacialmente, de forma dinámica a partir de información geográfica. No se accede a los datos originales, a los datos en sí mismos, sino que lo que se obtiene es una representación gráfica de ellos, una imagen digital. Estos mapas pueden contener información de varias capas ráster y/o vectoriales, superpuestas en un determinado orden modificable. Se generan normalmente en un formato de imagen como PNG, GIF o JPEG.

⁹ **RPC** acrónimo en Inglés de Remote Procedure Call

Existen servicios conocidos como Web Feature Service (WFS) los cuales ofrecen una interfaz de comunicación que permite interactuar con los mapas servidos por el estándar WMS, como por ejemplo, editar la imagen que nos ofrece el servicio WMS o analizar la imagen siguiendo criterios geográficos.

1.3.6. UDDI.

Es un modelo de directorios para Web Services. Es una especificación para mantener directorios estandarizados de información acerca de los Web Services, sus capacidades, ubicación, y requerimientos en un formato reconocido universalmente. UDDI utiliza WSDL para describir las interfaces de los Web Services. Es un lugar en el cual podemos buscar cuales son los Servicios web disponibles, una especie de directorio en el cual podemos encontrar los Web Services publicados y la descripción de sus servicios.

1.4. Descripción actual del dominio del problema.

El Centro de Geoinformática y Señales Digitales (GEYSED) de la Facultad 6 destina una de sus líneas a la producción de software y servicios para la Geomática, uno de sus productos insignias es el Sistema de Información Geográfica de la UCI (SIG-UCI), que es una solución creada para la georreferenciación de los objetivos socioeconómicos del centro y la toma de decisiones sobre el entorno geográfico del mismo. Dispone para los usuarios un conjunto de funcionalidades de representación, análisis temático y edición sobre el mapa digital de la universidad, que se sirve desde una biblioteca de datos espaciales las cuales se combinan con consultas a los servicios Web de información atributiva disponibles en la infraestructura de red. Sin embargo, hoy en día no se han explotado suficientemente las ventajas que pueden brindar los servicios de mapas para ampliar su gama de funcionalidades y prestar mejores asistencias a los usuarios sobre la información geográfica propia, ya que con estas bondades se aprovechan estos resultados en otras aplicaciones o subsistemas como complementos de mapificación o localización de cualquier objetivo.

1.5. Situación Problemática.

Desde la puesta en marcha de la versión 1.0 del Sistema de Información Geográfica de la UCI (SIG-UCI), la comunidad universitaria cuenta con una cómoda herramienta para el trabajo de localización de objetivos, análisis de datos geográficos y demás procesos de toma de decisiones sobre el mapa de la ciudad. Con una totalidad de 20 funcionalidades accesibles desde la Web y en consonancia con el proceso de informatización que hoy en día se lleva a cabo en el centro, sin embargo existe una

dependencia funcional de sus servicios que limitan su explotación mediante aplicaciones o usuarios que requieran el uso de los mapas para procedimientos de georreferenciación particular de objetivos, así como de los requerimientos tecnológicos pautados (navegadores, recursos de hardware y espacio).

El Centro de Informatización de la UCI (CENIA) lleva desde hace varios años un proyecto de trabajo colaborativo, estandarizado basado en una sólida arquitectura orientada a servicios. Entre sus principales objetivos está la interoperabilidad entre los sistemas de software que actualmente están desplegados en el centro y para ello ha integrado un conjunto de servicios Web en la plataforma UDDI.UCI.CU que garantizan la centralización de los datos comunes que se manejan a diario en los procesos medulares. Hoy en día, la carencia de recursos de mapas está visible en la red global, dado por la poca explotación de los sistemas de mapificación que actualmente existen en la comunidad universitaria. Ejemplo de ello es el paquete de búsqueda por directorios de la intranet, que abarca solo resultados de ubicación social de los recursos pero sin incluir el componente espacial. Otros como los sistemas de control de acceso y seguridad, sistemas de gestión residencial, sistemas de contabilidad de activos fijos, entre otros, carecen de mecanismos de análisis de información geográfica, datos que son importantes para la ejecución de los procesos de toma de decisiones.

Se encomendó al Departamento de Geoinformática del Centro GEYSED disponer para la comunidad universitaria de un paquete de servicios de mapas que se nutran de los datos espaciales del SIG-UCI y enriquecer con nuevas funcionalidades el proyecto colaborativo de UDDI.UCI.CU. Se planteó como objetivo el desarrollo de una plataforma que brinde servicios de mapas mediante el estándar WMS e incluya funcionalidades de localización de edificios, localización de usuarios, localización de objetivos socioeconómicos, entre otros.

1.6. Análisis de otras soluciones existentes.

1.6.1. Google Maps.

Google Maps es uno de los servicio de mapas que brinda Google, este servidor de aplicaciones de mapas ofrece imágenes de mapas e incluso fotos satelitales del mundo entero, permite mover y hacer zoom a la imagen obtenida. El usuario puede crear una lista de puntos para saber el cómo llegar a su destino, calculando el tiempo necesario y la distancia recorrida entre las ubicaciones. También ofrece la posibilidad de que cualquier propietario de una página Web integre muchas de sus características a su sitio de forma gratuita, pero si se pretende hacer uso comercial de su servicio tiene que pagarlo, de esta forma se le incluyen un conjunto de nuevas funcionalidades. **(Google, 2010)**

Google Maps no funciona siguiendo el estándar WMS, sino que funciona a través de su propia API¹⁰, sirviendo las imágenes que ya tiene generadas y georreferenciadas, troceadas a las distintas escalas y almacenadas en sus gigantes almacenes de información, el proceso de es realmente rápido y esto lo logra por que monta la imagen por cuadrículas, cuando se desplaza el mapa solo tiene que añadir las cuadrículas adyacentes en vez de cargar toda la imagen entera.

1.6.1.1. Funciones de Google Maps.

Google Maps es un servicio de Google que ofrece tecnología cartográfica potente y muy fácil de usar, así como información sobre empresas locales (ubicación, información de contacto e indicaciones sobre cómo llegar). **(Google, 2010)**

Google Maps cuenta con las siguientes funciones:

- ❖ Paneo de los mapas: permite realizar paneo del mapa haciendo clic y arrastrando el puntero del ratón, obteniendo de esta forma los mapas adyacentes.
- ❖ Zoom: permite realizar zoom del mapa mediante la rueda de desplazamiento del ratón, ampliando o reduciendo el mapa.
- ❖ Imágenes por satélite: permite obtener imágenes satelitales con datos de mapa superpuestos, y realizar las funciones de zoom y paneo.
- ❖ Rutas detalladas: permite introducir una dirección y Google Maps se encargará de señalarla y de trazar el itinerario.
- ❖ Street View: permite ver imágenes de las calles y desplazarte por ellas.

1.6.2. Infraestructuras de Datos Espaciales de la República de Cuba.

La Infraestructura de Datos Espaciales de la República de Cuba (IDERC) abarca las políticas, tecnologías, estándares y recursos humanos necesarios para la efectiva recolección, administración, acceso, entrega y utilización de los datos espaciales a nivel nacional en función de la toma de decisiones económicas, políticas y sociales, y del desarrollo sostenible. **(IDERC,2004)**

La IDERC provee un conjunto de datos como son: Distribución Política Administrativa, Hidrografía, Puntos Poblados y asentamientos urbanos, Elevaciones, Viales y Nombres Geográficos de la república

¹⁰ **API** acrónimo en Inglés de Application Programming Interface

de Cuba. Cuenta con un conjunto de servicios como el Diccionario Geográfico, Nombres Geográficos y un Visor de Mapas accesible mediante mapas interactivos.

1.6.3. Servicio Regional de Mapas del Programa GeoSUR.

El Servicio Regional de Mapas del Programa GeoSUR es un servicio interactivo que ofrece acceso a mapas digitales regionales de Suramérica en diversos temas, entre los que están centros poblados, infraestructura, vialidad, relieve, puertos. También ofrece acceso a una base de datos espacial con información detallada de la cartera de proyectos de la Iniciativa para la Integración de la Infraestructura Regional Suramericana.

El servicio posee una funcionalidad similar a la ofrecida por los sistemas de información geográfica, no requiere de software especializado y es de libre acceso. La aplicación permite al usuario visualizar tanto mapas regionales como mapas nacionales o locales, acceder a sus bases de datos asociadas, realizar acercamientos a áreas de interés, realizar búsquedas detalladas o revisar fichas descriptivas de los mapas. No se requiere de conocimientos especiales o técnicos para utilizar la herramienta.

El sistema no requiere de ningún software especial para operar solo necesita un navegador y una conexión a internet. No tiene gran variedad de navegadores que lo soportan y en esto radica su principal desventaja, ya que solo pueden utilizarlo el Firefox 2.0, Internet Explorer 7.0 Netscape Communicator 8.1.

1.7. Conclusiones parciales.

La evolución marcada de los Sistemas de Información Geográfica en la sociedad es directamente proporcional a la necesidad creciente de los usuarios del uso de servicios de internet. El auge de estas tecnologías basadas en servicios hace menos dependientes a los clientes del propio desarrollo de sistemas y una mejor distribución de los resultados basada en formatos de datos estándares en la Web. Empresas como Google, ESRI y SuperMaps lideran el espectro de los servicios de mapas y aprovechan las ventajas de tecnologías como los WMS y WFS para la compartimentación de los datos espaciales propios y de terceros. La Universidad de las Ciencias Informáticas comenzó a dar sus primeros pasos en este sentido, su infraestructura basada en arquitecturas de servicios hace de ella una matriz ideal para desplegar soluciones en la Web con estas características. Soluciones como el SIG-UCI, con un despliegue que refleja casi un año de explotación y la interacción sistemática de miles de usuarios, forma parte del grupo de sistemas que debe adoptar la iniciativa de su funcionamiento

basado en servicios; es misión del proyecto Aplicativos SIG lograr tal reto, que garantizará la publicación de información geográfica hacia diversas variantes y objetivos.

La evolución de los sistemas de información geográfica en la sociedad y la necesidad de los usuarios por el uso de servicios impone encaminar el desarrollo de sistemas basados en estándares para la Web. Empresas como Google y SuperMaps lideran el compartimiento de información geográfica basada en estos estándares. La Universidad de las Ciencias Informáticas ya dio sus primeros pasos y cuenta con una infraestructura basada en arquitecturas de servicios, por lo que es misión del proyecto Aplicativos SIG garantizar la publicación de información geográfica con su cartografía.

CAPÍTULO 2:” Herramientas y tecnologías”.

2.1. Introducción.

El creciente desarrollo de la Informática actual es posible por el perfeccionamiento de las herramientas y tecnologías para el desarrollo de software, que constituyen hoy instrumentos fundamentales para garantizar la calidad de los procesos de desarrollo.

En este capítulo se explica de forma breve el uso de las herramientas y tecnologías a utilizar para el desarrollo del sistema. Se abordan temas relacionados con el uso de metodologías de desarrollo, lenguajes de programación, así como los sistemas para la realización del análisis y diseño del mismo. El propósito del estudio que se brinda a continuación está basado en dar a conocer cuáles serán las tecnologías que guiarán el proceso de implementación.

2.2. Fundamentación de las tecnologías utilizadas.

De acuerdo con la política de la utilización “Software Libre” para la construcción de sistemas informáticos, uno de los valores más importantes de esta solución es enfocar las herramientas a utilizar para su desarrollo a software libre.

Las tecnologías son:

- ❖ MapServer
- ❖ XML
- ❖ PHP 5
- ❖ Apache
- ❖ OpenLayers

Para la implementación de este sistema se ha realizado un estudio previo del estado del arte de las tecnologías empleadas en el mundo en la construcción de aplicaciones de este tipo, lo que ha permitido en todo momento el uso de las más modernas técnicas, teniendo en cuenta además la estabilidad y empleando para ello patrones de diseño y de construcción que reúnen la experiencia de muchos años de buenas prácticas de programación.

2.2.1. PHP.

PHP (Hypertext Preprocessor), es un lenguaje de programación interpretado de alto nivel, embebido en páginas HTML y ejecutado en el servidor, diseñado originalmente para la creación de páginas Web dinámicas. (**Van Der Henst, 2001**).

PHP está orientado al desarrollo Web, es de gran velocidad por lo que no requiere de muchos recursos del sistema y además se integra perfectamente con muchos servidores. Es libre y está disponible bajo la licencia GPL, es multiplataforma por lo que no tendrá ningún inconveniente al usarlo en cualquier computadora de la Universidad. Se caracteriza por la simplicidad de su código y por la amplia documentación que brinda.

Se decidió utilizar PHP como lenguaje de programación del lado del servidor por sus amplias ventajas y características.

2.2.1.1. Características de PHP.

- ❖ Es un lenguaje multiplataforma tanto para diversos Sistemas Operativos, como servidores HTTP y bases de datos.
- ❖ Es libre, por lo que se presenta como una alternativa de fácil acceso para todos.
- ❖ El código se actualiza continuamente con mejoras y extensiones de lenguaje para ampliar sus capacidades.
- ❖ Capacidad de conexión con una gran cantidad de bases de datos como MySQL, PostgreSQL, Oracle, MS SQL Server, Sybase mSQL, Informix, entre otras.
- ❖ Permite aplicar técnicas de programación orientada a objetos.
- ❖ Orientado completamente al desarrollo de aplicaciones Web dinámicas con acceso a información almacenada en una base de datos.

- ❖ Permite la integración con varias bibliotecas externas, generar documentos en PDF y analizar código XML.

2.2.2. OpenLayers.

OpenLayers es una biblioteca de JavaScript de código abierto que posibilita mostrar mapas interactivos en los navegadores web. OpenLayers ofrece un API para acceder a diferentes fuentes de

información cartográfica como pueden ser los WMS. Además cuenta con una interface muy amigable muy parecida a la de Google Maps.

2.2.2.1. Características de OpenLayers.

- ❖ Soporta Capas vectoriales, Simple Interface, OGC WMS, OGC WFS, GeoRSS, Canvas.
- ❖ Pose un conjunto de controles para interactuar con el mapa como: Zoom panorámico, Barra zoom, Controls Mouse, Scale Ratio, Marcadores, Popups, Barra escala.
- ❖ Es de código libre lo que permite utilizarlo o modificarlo en cualquier momento.
- ❖ Carga muy poco al servidor.
- ❖ Puede utilizar múltiples servidores de datos.

2.2.3. XML.

XML (Extensible Markup Language), no es más que un metalenguaje que define la sintaxis utilizada para definir otros lenguajes de etiquetas estructurados. Es un subconjunto de SGML y contiene no del todo a HTML, especializado en la gestión de información para la Web.

2.2.3.1. Características de XML.

- ❖ Permite definir lenguajes de marcas para cualquier fin y tiene capacidades de validación de datos.
- ❖ Garantiza la gestión y manipulación de los datos desde el cliente Web.
- ❖ Facilita la exportación a otros formatos de publicación.
- ❖ Es extensible y flexible, lo que permite agrupar una amplia variedad de aplicaciones, desde páginas Web hasta bases de datos.
- ❖ No necesita versiones para que puedan funcionar en futuros navegadores, por lo que los identificadores pueden ser creados de manera simple y ser adaptados por medio de un validador de documentos (parser).
- ❖ Tiene un comportamiento más estable y actualizable de las aplicaciones Web, incluyendo enlaces bidireccionales y almacenados de forma externa.
- ❖ La codificación del contenido Web en XML posibilita que la estructura de la información resulte más accesible, por lo que los motores de búsqueda devolverán respuestas más precisas y adecuadas.

2.2.4. Metodología de Desarrollo.

Una metodología de desarrollo es un proceso que engloba procedimientos, técnicas, documentación y herramienta que se utilizan en la creación de un producto de software.

Se quiere construir un software de alta calidad, desarrollado en el tiempo planificado y con los costes establecidos, pero que además satisfaga la necesidad de ser elaborado de una forma más acelerada y que exista una reducción del costo del producto, es necesario enfocarse en trabajar de forma organizada, donde se controle y documente todo lo relacionado con el proyecto en cuestión y puedan eliminarse los riesgos que podrían presentarse durante el desarrollo del mismo, lo cual no podría lograrse sin el empleo de una metodología eficaz que se adapte a las características propias del software que se esté desarrollando.

2.2.4.1. Proceso Unificado de Desarrollo (RUP).

El Proceso Unificado de Desarrollo (RUP) es una metodología que se utiliza para la Ingeniería de Software, va más allá del análisis y el diseño orientado a objetos, para proporcionar técnicas que soportan un ciclo de desarrollo de software. Las 4 P¹¹ conocidas sustentan el proceso de desarrollo de software. RUP define para cada etapa: el flujo de trabajo, los trabajadores que intervienen, las actividades que realizan y los artefactos que se necesitan o producen. Su meta es asegurar la producción de software con la más alta calidad, que cumpla con las necesidades de los usuarios dentro del cronograma planeado y la inversión prevista.

El Proceso Unificado está basado en componentes, lo cual quiere decir que el sistema software en construcción está formado por componentes software interconectados a través de interfaces bien definidas. Además, el Proceso Unificado utiliza el Lenguaje Unificado de Modelado (UML: Unified Modeling Language) para expresar gráficamente todos los esquemas de un sistema software.

Los aspectos más importantes que definen este Proceso Unificado son tres: es iterativo e incremental, dirigido por casos de uso y centrado en la arquitectura **(Ibar Jacobson, 2000)**

En su modelación define como sus principales elementos:

¹¹ 4 P: (*proyecto, personas, producto y proceso*).

- ❖ **Trabajadores** (Quién): define el comportamiento y responsabilidades de un individuo, grupo de individuos, sistema automatizado o máquina, los cuales se encargan de realizar las actividades y son los responsables de una serie de artefactos.
- ❖ **Actividades** (Cómo): es una tarea que tiene un propósito claro, es realizada por un trabajador y manipula elementos.
- ❖ **Artefactos** (Qué): productos tangibles del proyecto que son producidos, modificados y usados por las actividades.
- ❖ **Flujo de actividades** (Cuándo): secuencia de actividades realizadas por trabajadores y que produce un resultado de valor observable.

RUP se divide en 4 fases:

- ❖ Inicio: el objetivo de esta etapa es determinar la visión del proyecto.
- ❖ Elaboración: el objetivo es determinar la estructura óptima.
- ❖ Construcción: el objetivo es obtener la capacidad operacional inicial.
- ❖ Transición: el objetivo es obtener la primera versión del proyecto.

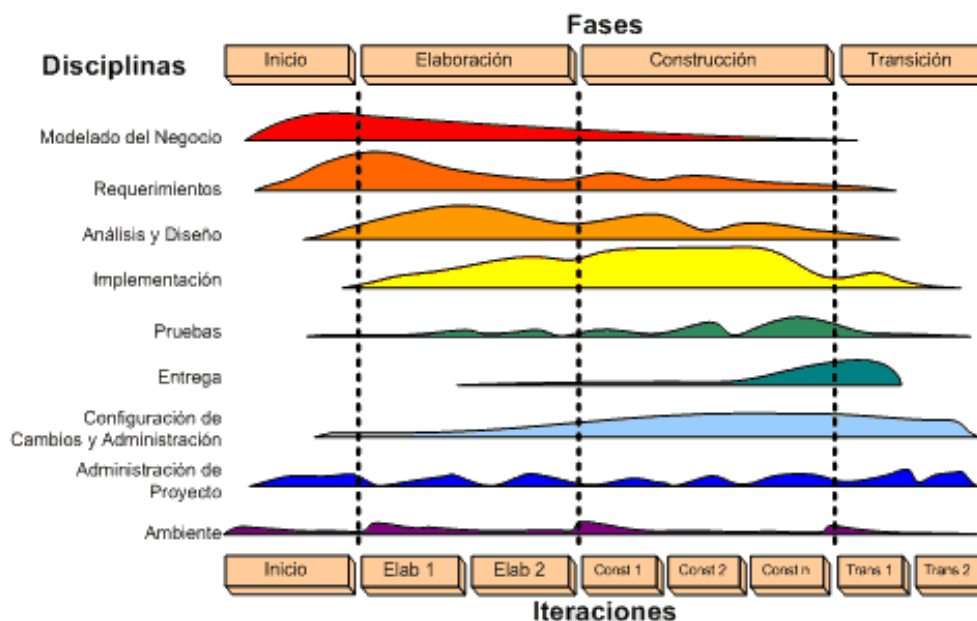


Figura 3 Imagen representativa de la estructura de RUP.

Flujos de Trabajo

- ❖ **Modelamiento del negocio:** Describe los procesos de negocio, identificando quiénes participan y las actividades que requieren automatización.

- ❖ **Requerimientos:** Define qué es lo que el sistema debe hacer, para lo cual se identifican las funcionalidades requeridas y las restricciones que se imponen.
- ❖ **Análisis y diseño:** Describe cómo el sistema será realizado a partir de las funcionalidades previstas y las restricciones impuestas, por lo que indica con precisión lo que se debe programar.
- ❖ **Implementación:** Define cómo se organizan las clases y objetos en componentes, cuáles nodos se utilizarán y la ubicación en ellos de los componentes y la estructura de capas de la aplicación.
- ❖ **Prueba (Testeo):** Busca identificar los defectos y corregirlos antes de la instalación final del sistema, se verifica la integración apropiada de los componentes y que se satisfacen los requerimientos de los clientes.
- ❖ **Instalación (Despliegue):** Produce release del producto y realiza actividades (empaquete, instalación, asistencia a usuarios, y otras) para entregar el software a los usuarios finales.
- ❖ **Administración del proyecto:** Involucra actividades con las que se busca producir un producto que satisfaga las necesidades de los clientes.
- ❖ **Administración de configuración y cambios:** Describe cómo controlar los elementos producidos por todos los integrantes del equipo de proyecto.
- ❖ **Ambiente:** Contiene actividades que describen los procesos y herramientas que soportarán el equipo de trabajo del proyecto; así como el procedimiento para implementar el proceso en una organización.

El Proceso Unificado de Modelado RUP ofrece una alta capacidad organizativa para el proyecto. La unificación de los mejores elementos de otras metodologías hace que RUP sea de primer nivel. Está preparada para desarrollar proyectos de grandes envergaduras pero posee alta flexibilidad, lo que permite que sea ajustado a nuestro entorno de desarrollo.

2.2.4.2. Características de RUP.

El ciclo de vida de RUP se caracteriza por ser: dirigido por casos de uso, centrado en la arquitectura y además por ser iterativo e incremental.

Los casos de uso reflejan lo que los usuarios futuros necesitan y desean, lo cual se capta cuando se modela el negocio y se representa a través de los requerimientos. A partir de aquí los casos de uso

guían el proceso de desarrollo, obteniéndose un producto final que cuenta con la calidad requerida por el cliente.

La arquitectura muestra la visión común del sistema completo, que describe los elementos del modelo que son importantes para la construcción del software, creándose de esta manera los cimientos del sistema que son necesarios como base para comprenderlo, desarrollarlo y producirlo. RUP propone que se desarrolle el software mediante iteraciones, comenzando por los casos de uso relevantes desde el punto de vista para la arquitectura y además que cada fase se desarrolle en iteraciones, donde en cada iteración se involucran actividades de todos los flujos de trabajo, aunque se desarrollan fundamentalmente algunos flujos más que otros.

- ❖ Creado por Jacobson, Rumbaugh y Booch.
- ❖ Unifica los mejores elementos de metodologías anteriores.
- ❖ Preparado para desarrollar grandes y complejos proyectos.
- ❖ Orientado a objetos.
- ❖ Utiliza UML como lenguaje de representación visual.

2.2.5. El Lenguaje Unificado de Modelado (UML) como soporte de la modelación de la solución propuesta.

El Proceso Unificado de Desarrollo utiliza Lenguaje Unificado de Modelado para preparar todos los esquemas de un sistema de software. Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema de software que ayuda a capturar la idea de un sistema para comunicarla posteriormente a quien está involucrado en su proceso de desarrollo; esto se lleva a cabo mediante un conjunto de símbolos y diagramas. Cada diagrama tiene fines distintos dentro del proceso de desarrollo (**Schmuller, 2000**).

Se ha seleccionado UML como soporte de la modelación de la solución propuesta, ya que sirve para modelar todas las fases del proceso de desarrollo del sistema, tanto en el diseño de los sistemas de software como para la arquitectura hardware donde se ejecuten. El lenguaje UML tiene una notación gráfica muy expresiva que permite representar en mayor o menor medida todas las fases de un proyecto informático: desde el análisis con los casos de uso, el diseño con los diagramas de clases, objetos, etc., hasta la implementación y configuración con los diagramas de despliegue.

2.2.5.1. Características del UML.

- ❖ Permite modelar sistemas utilizando técnicas orientadas a objetos (OO).

- ❖ Permite especificar todas las decisiones de análisis, diseño e implementación, construyéndose así modelos precisos, no ambiguos y completos.
- ❖ Puede conectarse con los lenguajes de programación (Ingeniería directa e inversa).
- ❖ Permite documentar todos los artefactos de un proceso de desarrollo (requisitos, arquitectura, pruebas, versiones, entre otras.).
- ❖ Cubre las cuestiones relacionadas con el tamaño propio de los sistemas complejos y críticos.
- ❖ Es un lenguaje muy expresivo que cubre todas las vistas necesarias para desarrollar y luego desplegar los sistemas.
- ❖ Existe un equilibrio entre expresividad y simplicidad, pues no es difícil de aprender ni de utilizar.
- ❖ UML es independiente del proceso, aunque para utilizarlo óptimamente se debería usar en un proceso que fuese dirigido por los casos de uso, centrado en la arquitectura, iterativo e incremental. **(Joaquín, 2005)**

2.2.6. Visual Paradigm.

Visual Paradigm para UML es una herramienta profesional multiplataforma que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. 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, código inverso, generar código desde diagramas y generar documentación. La herramienta UML CASE también proporciona abundantes tutoriales de UML, demostraciones interactivas de UML y proyectos UML.

Teniendo en cuenta que Visual Paradigm es una herramienta multiplataforma, que brinda importantes prestaciones para el diseño de la aplicación. Se decidió usarlo para la modelación del sistema.

2.2.6.1. Características de Visual Paradigm.

- ❖ Ofrece un entorno de creación de diagramas para UML versión 2.1
- ❖ Generación de código.
- ❖ Generación de bases de datos.
- ❖ Transformación de diagramas de entidad.
- ❖ Relación en tablas de base de datos.
- ❖ Es disponible en múltiples plataformas.

- ❖ Diseño centrado en casos de usos y enfocado al negocio que generan un software de mayor calidad.
- ❖ Capacidad de ingería directa e inversa.
- ❖ Soporta aplicaciones Web.

2.2.7. Servidor Web Apache.

Apache es uno de los servidores Web más empleado en el mundo, cada día millones de servidores reiteran su confianza por su estabilidad. Se decidió usar Apache como nuestro servidor Web debido a que es de código abierto y es compatible con diversas plataformas como son GNU/Linux, Microsoft Windows y Macintosh.

2.2.7.1. Características de Apache.

- ❖ Puede ser adaptado a diferentes entornos y necesidades, con los diferentes módulos de apoyo y con la API (application programming interface) de programación de módulos.
- ❖ Es un servidor altamente configurable de diseño modular. Es muy sencillo ampliar las capacidades del servidor Web Apache.
- ❖ Permite personalizar la respuesta ante los posibles errores que se puedan dar en el servidor. Es posible configurar Apache para que ejecute un determinado script cuando ocurra un error en concreto.
- ❖ Tiene diversas formas de configurar la creación y gestión de log. Apache permite la creación de ficheros de log a medida del administrador, de este modo puedes tener un mayor control sobre lo que sucede en tu servidor.
- ❖ Es multiplataforma: corre en varios Sistemas Operativos, lo que lo hace prácticamente universal.

2.2.8. Servidor de mapas MapServer.

Estudios realizados revelan que este proyecto “nació como unos scripts para ArcINFO que generaban de forma dinámica impresiones de cartografía para publicar web. Inicialmente fue un proyecto financiado por la NASA, la Universidad de Minnesota y el departamento de recursos forestales de Minnesota (MNDNR)” (Méndez, 2009).

Es calificado como el producto de software libre más maduro y popular para el desarrollo de aplicaciones Web espaciales. Programado en el lenguaje C, accede a todos los formatos soportados por OGR y GDAL, y permite programación con PHP y otros muchos lenguajes.

2.2.8.1. Características de MapServer.

- ❖ Sencillez de configuración y administración.
- ❖ Plataformas sobre las que puede operar.
- ❖ Velocidad de acceso a datos.
- ❖ Cantidad de formatos tanto vectoriales como raster soportados.
- ❖ CGI o biblioteca de componentes (MapScript).
- ❖ Obtiene un performance significativamente superior cuando se utiliza en combinación con PostGIS/PostgreSQL.
- ❖ Accesible desde PHP, Python, Java, Perl, C#.
- ❖ Servidor de mapas rápido y ligero.

2.2.9. Entornos de Desarrollo NetBeans.

NetBeans es una herramienta de desarrollo para aplicaciones, escrita puramente sobre la base de la tecnología Java, de modo que puede ejecutarse en cualquier ambiente que ejecute Java. Es un producto de código abierto, con todos los beneficios del software disponible en forma gratuita, el cual ha sido examinado por una comunidad de desarrolladores. Aparte de la filosofía de distribución y desarrollo que respalda a NetBeans, el IDE ofrece a los desarrolladores numerosas ventajas, en la creación de nuevas aplicaciones multiplataforma.

En una era en la cual la arquitectura orientada al servicio (SOA) requiere servicios con cierta relación que manejen procesos específicos del negocio, este satisface los requisitos con conjuntos de herramientas independientes de la plataforma, modulares y orientadas al objeto (**Sun Microsystem, 2009**).

Es un IDE de desarrollo maduro y con soporte a gran cantidad de librerías, lo cual lo convierte en un entorno de desarrollo ideal, por lo que permite a los programadores el uso de las mismas de una forma muy fácil e intuitiva. Es muy sencillo de utilizar y exige muy poco conocimiento por parte del usuario para su uso. Es completamente software libre y tiene versiones para los principales sistemas operativos como: Windows, MacOS y LINUX que es el sistema operativo donde se va a implementar.

2.3. Conclusiones.

A partir del análisis y estudio de diferentes herramientas de acuerdo a la tendencia mundial, se determinaron las más idóneas para garantizar el buen desarrollo del sistema. En la selección de éstas se tuvieron en cuenta varios criterios tecnológicos como su flexibilidad, compatibilidad (Código Abierto) y facilidades de empleo. Como lenguaje de programación se seleccionó PHP, que en conjunto con Ajax, XML, HTML y JavaScript convierten la plataforma en un entorno de trabajo cliente-servidor adecuado. La preferencia por MapServer como servidor de mapas estuvo dada por su potencialidad en la compartimentación de los recursos de mapas en la Web, así como su arquitectura para la orquestación de Web Map Services (WMS).

CAPÍTULO 3 “Presentación de la solución propuesta”

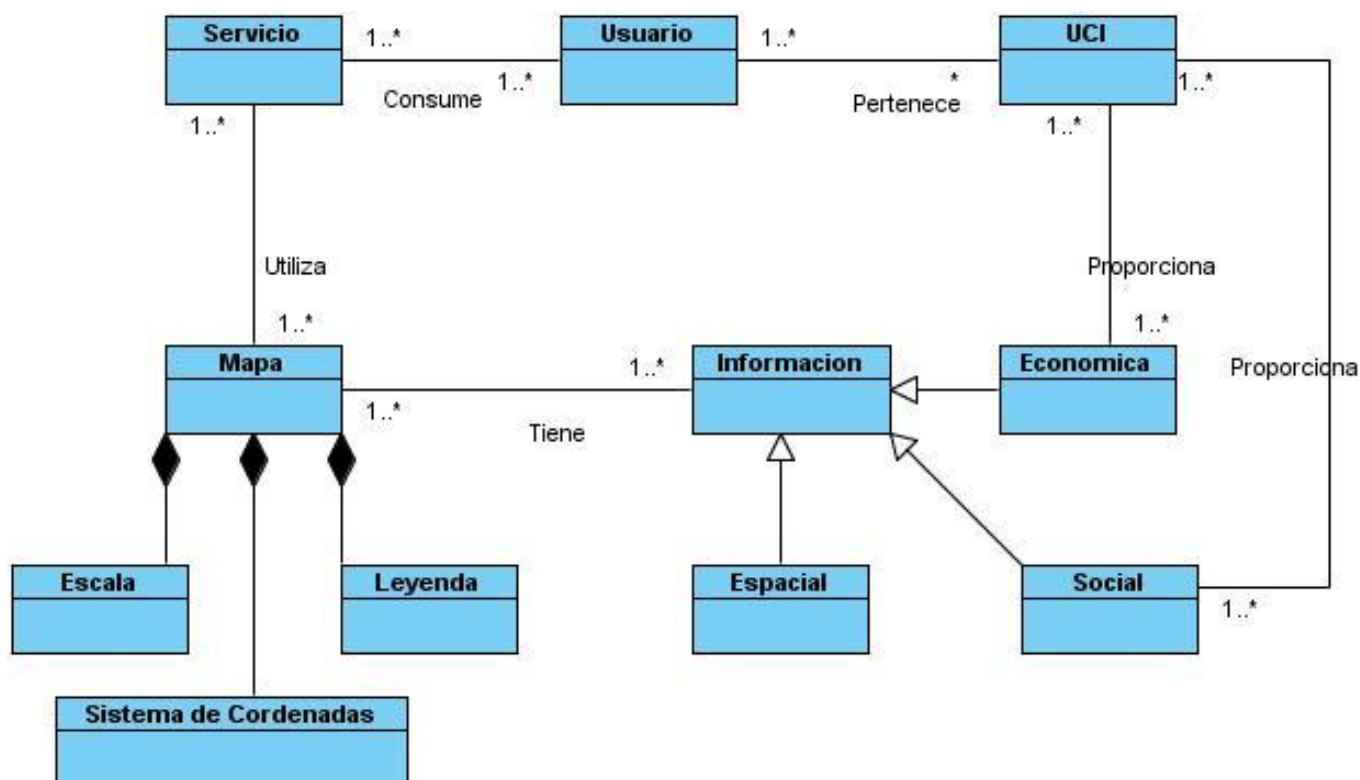
3.1. Introducción.

En éste capítulo se exponen los artefactos generados correspondientes al flujo de trabajo de modelado. Debido a la poca estructuración de los procesos de negocio y para poder comprender el contexto en el cual se desarrollan los componentes se determinó desarrollar un Modelo de Dominio, donde se expone un marco conceptual y las relaciones entre estas definiciones. Por otra parte, se enumeran los requerimientos funcionales y no funcionales, agrupándose los primeros en Casos de Uso, con el fin de estructurar el Diagrama de Casos de Uso del Sistema.

3.2. Modelo de Dominio.

El Modelo de Dominio, permite de manera visual mostrar al usuario los principales conceptos que se manejan en el dominio del problema. Un Modelo del Dominio es una representación de las clases conceptuales del mundo real, no de componentes de software. El modelo desarrollado no se trata de un conjunto de diagramas que describen clases de software u objetos de software con responsabilidades, sino que puede considerarse como un diccionario visual de las abstracciones relevantes, vocabulario e información del dominio. Aprovechando las bondades de los diagramas UML para representar conceptos, el Modelo de Dominio se presenta en forma de diagrama de clases donde figuran los principales conceptos y roles del sistema en cuestión.

3.2.1. Diagrama de clases del Modelo de Dominio.



3.2.2. Glosario de Términos del Dominio.

Para una mejor comprensión del Diagrama del Modelo de Dominio se proporciona una descripción de las clases que se encuentran en el Diagrama.

❖ **Usuario.**

Usuario o sistema de la Universidad de las Ciencias Informáticas, que necesite obtener o consultar algún tipo de mapa.

❖ **Servicio.**

Es el servicio Web el cual es el encargado de publicar los mapas para que cualquier usuario lo consuma.

❖ **Mapa.**

Es una representación gráfica y métrica de una porción de territorio sobre una superficie bidimensional, generalmente plana, pero que puede ser también esférica como ocurre en los globos terráqueos. El que el mapa tenga propiedades métricas significa que ha de ser posible tomar medidas de distancia, ángulos o superficies sobre él y obtener un resultado aproximadamente exacto.

❖ **Sistema de Coordenadas.**

Es un sistema de referencia usado para localizar y medir elementos geográficos. Para representar el mundo real se utiliza el sistema de coordenadas en el cual la localización de un elemento esta dado por las magnitudes de latitud y longitud en unidades de grados, minutos y segundo.

❖ **Escala.**

Relación entre la distancia que separa dos puntos en un mapa y la distancia real de esos dos puntos en la superficie terrestre. En los mapas, la escala puede expresarse de tres modos distintos: en forma de proporción o fracción, con una escala gráfica o con una expresión en palabras y cifras. Cuanto mayor es la escala, más se aproxima al tamaño real de los elementos de la superficie terrestre. Los mapas a pequeña escala generalmente representan grandes porciones de la Tierra y, por tanto, son menos detallados que los mapas realizados con escalas más grandes.

La relación matemática entre las dimensiones en el mapa, carta o plano y la superficie terrestre que representa. Por extensión puede referirse a la mayor o menor profundidad del enfoque en un tema geográfico.

❖ **Leyenda.**

Explicación de los símbolos, los colores, las tramas y los sombreados empleados en un mapa; suele encontrarse a pie de página o en un recuadro, situado en sus márgenes o bien en su dorso. Los símbolos empleados en los mapas pueden llegar a contener un gran volumen de información, que por su facilidad de lectura permiten una rápida interpretación.

❖ **Información.**

Conjunto de datos que están organizados y que tienen un significado. En este caso pueden ser de origen científico, social o espacial.

❖ Científica.

Se refiere al conjunto de métodos y técnicas que organizan la información adquirida mediante la experiencia o la introspección.

❖ Espacial.

Es el tipo de información referente los datos espaciales, a los puntos y su ubicación en el mapa.

❖ Social.

Es aquello perteneciente o relativo a la sociedad.

3.3. Levantamiento de requisitos.

Como parte del modelado se definieron los requerimientos, los cuales son una “condición o capacidad que necesita un usuario para resolver un problema o lograr un objetivo” (**Jacobson, Booch ,2004**). Los requerimientos se clasifican en funcionales y no funcionales. Los requerimientos funcionales “son capacidades o condiciones que el sistema debe cumplir, se mantienen invariables sin importar con que propiedades o cualidades se relacionen por lo que no alteran la funcionalidad del producto” (**Jacobson, Booch ,2004**). Y los requerimientos no funcionales “son las propiedades o cualidades que el sistema debe tener” (**Jacobson, Booch ,2004**). Los requerimientos no funcionales son aquellos requisitos que hacen que el sistema sea usable, rápido, confiable y agradable para los usuarios.

3.3.1. Requerimientos Funcionales.

Los requerimientos funcionales son capacidades o condiciones que el sistema debe cumplir. Se mantienen invariables sin importar con qué propiedades o cualidades se relacionen.

RF1 Devolver Mapa: El sistema debe permitir que el usuario pueda obtener el mapa, permitiéndole realizar las operaciones de zoom, mover el mapa, recentrar el mapa, también debe mostrar la escala gráfica y las coordenadas de la posición marcada por el puntero del mouse.

RF 2 Localizar Edificio Residencial: El sistema debe permitir localizar un edificio residencial dado.

RF 4 Localizar Persona Por Solapín: El sistema debe permitir localizar la residencia de una persona de la universidad dado el solapín.

RF 5 Localizar Persona Por Carne de Identidad: El sistema debe permitir localizar la residencia de una persona de la universidad dado el carné de identidad.

RF 6 Localizar Persona Por Usuario: El sistema debe permitir localizar la residencia de una persona de la universidad dado el usuario.

RF7 Localizar Computadora: El sistema debe permitir buscar la ubicación de una computadora de la universidad por su número TCP/IP.

RF 8 Localizar Objetivo: El sistema debe permitir localizar un objetivo socioeconómico dado.

3.3.2. Requerimientos No Funcionales.

Los requerimientos no funcionales son fundamentales en el éxito del producto, debido a que forman una parte significativa del mismo. Estos requerimientos constituyen rasgos fundamentales que hacen que el sistema sea usable, rápido, confiable y agradable para los usuarios. De ahí la importancia de su correcta descripción para el funcionamiento óptimo de la aplicación.

Usabilidad:

La usabilidad del sistema PLASMA sera elevada, o sea: contara con un alto nivel de aceptación por los usuarios finales, porque será sencillo, agradable y bien organizado. Las técnicas empleadas en su confección permiten que el sistema sea utilizado por usuarios con conocimientos mínimos de computación.

Apariencia o interfaz externa:

El sistema PLASMA debe brindar una interfaz amigable, intuitiva y de fácil comprensión para el usuario, facilitando en todo momento la interacción de este con el sistema. Se garantiza una interfaz de fácil uso y se combinará los colores de manera que sea agradable a la vista del usuario sin dejar que se pierda la seriedad y profesionalidad del sistema.

Rendimiento:

El tiempo de respuesta para visualizar el mapa en la pantalla será en menos de 2 segundos, para realizar las funciones de zoom y paneo entre 500ms y 1000ms en dependencia de la cantidad de información que tenga que actualizar.

Restricciones de diseño:

El diseño de PLASMA debe ser sencillo, donde no es necesario mucho entrenamiento para utilizarlo, el usuario simplemente usando su intuición se puede familiarizar fácilmente con el sistema.

Portabilidad:

El sistema será multiplataforma.

Hardware:

Para la PC cliente:

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

Para el servidor:

- Se requiere tarjeta de red.
- El Servidor tenga como mínimo 512MB de RAM
- Se requiere al menos 1GB de disco duro.
- Procesador 3 GHz o superior.

Software:

Para las PCs clientes:

- Navegador Web: Mozilla FireFox, Internet Explorer, Opera, Net Scape.
- PHP Script language 5.2.5.

Para los servidores:

- Apache Server 2.2.4.
- MapServer 5.6.1.
- Postgres 8.4.8.

3.4. Descripción de los actores.

Un actor del sistema no es parte del sistema en desarrollo, es un agente externo que intercambia información con el mismo en pos de obtener un resultado esperado. Este sistema cuenta con el siguiente actor:

Actor	Descripción
Usuario	Es el cliente que utiliza las funcionalidades de el sistema, puede ser un sistema informatico o un usuario.

Tabla 1 Descripción de los actores.

3.5. Diagrama de Casos de Uso del Sistema.

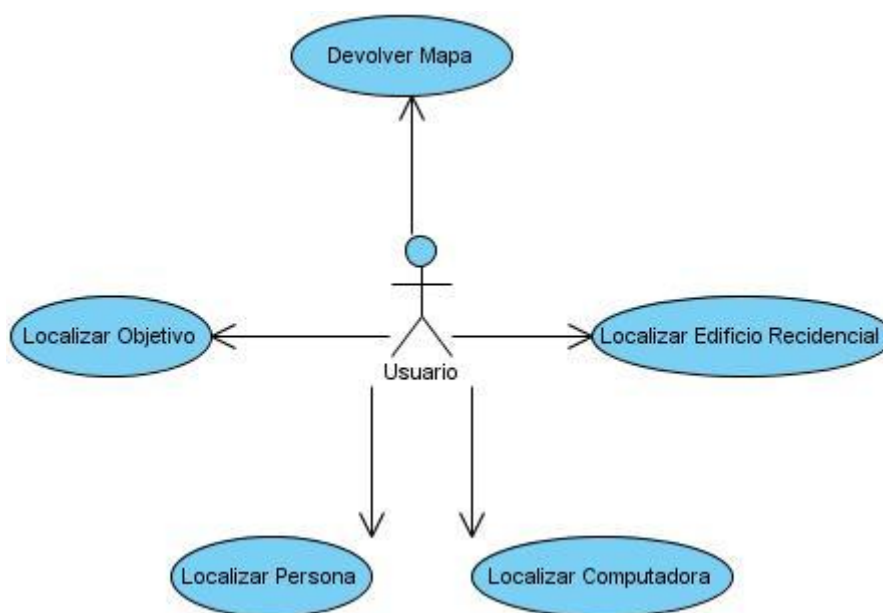


Figura 4 Diagrama de Casos de Uso del Sistema.

3.5.1. Descripción de los Casos de Uso del Sistema.

3.5.1.1. Descripción Textual del Caso de Uso Devolver Mapa.

Caso de Uso:	Devolver Mapa.
Actores:	Usuario.
Propósito	Este caso de uso se lleva a cabo con el objetivo de que el cliente obtenga el mapa y pueda interactuar con sus funcionalidades.
Resumen:	El caso de uso se inicia cuando el cliente consume uno de los servicios disponibles en el sistema y termina cuando el sistema devuelve el mapa resultante.
Precondiciones:	
Referencias	RF 1
Prioridad	Crítico
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. El cliente consume uno de los servicios disponibles.	2. El sistema devuelve el mapa centrado en el objetivo que se localizó.
3. El cliente utiliza una de las opciones de navegación: -Zoom In. -Zoom Out. -Mover Mapa. -El cliente deja presionado el clic izquierdo	4. El sistema realiza la operación dependiendo de la opción que se seleccionó: - Si seleccionó "Zoom In", ver sección "Zoom In". - Si seleccionó "Zoom Out", ver sección "Zoom Out".

<p>sobre el mapa y lo desplaza para mover el mapa.</p> <p>-El usuario hace doble clic en el mapa.</p>	<p>- Si seleccionó “Mover Mapa” o movió el mapa, ver sección “Mover Mapa”.</p>
	<p>5. El sistema procesa la información según la acción realizada por el usuario y actualiza la visualización del mapa.</p>

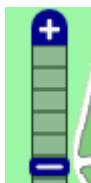
Prototipo de Interfaz



Sección “Zoom In”

Acción del Actor	Respuesta del Sistema
<p>1. El usuario selecciona la opción “Zoom In”.</p>	<p>2. El sistema realiza la operación de Zoom In seleccionada por el usuario, acercando el mapa y disminuyendo la escala en dependencia del nivel de zoom que escogió.</p>
	<p>3. El sistema procesa la información a partir de la acción realizada por el usuario y actualiza la visualización del mapa.</p>

Prototipo de Interfaz



Sección "Zoom Out"

Acción del Actor	Respuesta del Sistema
1. El usuario selecciona la opción del "Zoom Out".	2. El sistema realiza la operación de Zoom Out seleccionada por el usuario, alejando el mapa y aumentando la escala en dependencia del nivel de zoom que escogió.
	3. El sistema procesa la información a partir de la acción realizada por el usuario y actualiza la visualización del mapa.

Prototipo de Interfaz



Sección "Mover Mapa"

Acción del Actor	Respuesta del Sistema
1. El usuario clic en un punto determinado y arrastra el mouse.	2. El sistema calcula un ΔX y un ΔY a partir del centro de pantalla y se obtiene un nuevo centro de pantalla a donde se movería el mapa. No varía la escala de representación del mapa.


3. El usuario hace clic en el panel de navegación para mover el mapa.	4. El sistema desplaza el mapa en la dirección que selecciono el usuario. No varía la escala de representación del mapa.
	4. El sistema procesa la información a partir de la acción realizada por el usuario y actualiza la visualización del mapa.
<i>Prototipo de Interfaz</i>	
	
Poscondiciones	

Tabla 2 Descripción Textual del Caso de Uso Devolver Mapa.

3.5.1.2. Descripción Textual del Caso de Uso Localizar Objetivo.

Caso de Uso:	Localizar Objetivo.
Actores:	Usuario
Propósito	Este caso de uso se lleva a cabo con el objetivo de localizar y visualizar cualquier objetivo socioeconómico de la UCI.
Resumen:	El caso de uso se inicia cuando el cliente desea localizar cualquier objetivo socioeconómico de la UCI y termina cuando se devuelve el mapa centrado en el objetivo.
Precondiciones:	
Referencias	RF 1, RF 8
Prioridad	Crítico


Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. El cliente consume el servicio de “Localizar Objetivo”.	2. El sistema realiza la búsqueda del Objetivo deseado, devuelve el mapa centrado en la localización.
	2.1 El sistema devuelve un error, Objetivo no encontrado.
	3. El sistema actualiza la visualización del mapa.
<i>Prototipo de Interfaz</i>	
	
Poscondiciones	

Tabla 3 Descripción Textual del Caso de Uso Localizar Objetivo.

3.5.1.3. Descripción Textual del Caso de Uso Localizar Edificio Residencial.

Caso de Uso:	Localizar Edificio Residencial.
Actores:	Usuario
Propósito	Este caso de uso se lleva a cabo con el objetivo de localizar y visualizar un edificio del área residencial de la UCI.

Resumen:	El caso de uso se inicia cuando el cliente desea localizar un edificio residencial de la universidad y termina cuando se devuelve el mapa centrado en el objetivo.	
Precondiciones:		
Referencias	RF 1, RF 2	
Prioridad	Crítico	
Flujo Normal de Eventos		
Acción del Actor	Respuesta del Sistema	
1. El cliente consume el servicio de “Localizar Edificio Residencial”.	2. El sistema realiza la búsqueda del Edificio deseado, devuelve el mapa centrado en el Edificio.	
	2.1 El sistema devuelve un error, Edificio no encontrado.	
	3. El sistema actualiza la visualización del mapa.	
<i>Prototipo de Interfaz</i>		



Poscondiciones	
----------------	--

Tabla 4 Descripción Textual del Caso de Uso Localizar Edificio Residencial.

3.5.1.4. Descripción Textual del Caso de Uso Localizar Computadora.

Caso de Uso:	Localizar Computadora.
Actores:	Usuario
Propósito	Este caso de uso se lleva a cabo con el objetivo de localizar y visualizar la ubicación de una computadora teniendo su número de TCP/IP.
Resumen:	El caso de uso se inicia cuando el cliente desea localizar cualquier computadora de la UCI y termina cuando se devuelve el mapa centrado en el objetivo.
Precondiciones:	
Referencias	RF 1, RF 7
Prioridad	Crítico
Flujo Normal de Eventos	


Acción del Actor	Respuesta del Sistema
1. El cliente consume el servicio de “Localizar Computadora”.	2. El sistema realiza la búsqueda del Objetivo deseado, devuelve el mapa centrado en la localización.
	2.1 El sistema devuelve un error, Objetivo no encontrado.
	3. El sistema actualiza la visualización del mapa.
<i>Prototipo de Interfaz</i>	
	
Poscondiciones	

Tabla 5 Descripción Textual del Caso de Uso Localizar Computadora

3.5.1.5. Descripción Textual del Caso de Uso Localizar Persona.

Caso de Uso:	Localizar Persona.
Actores:	Usuario.
Propósito	Este caso de uso se lleva a cabo con el objetivo de localizar y visualizar el edificio donde reside una persona de la Universidad.

Resumen:	El caso de uso se inicia cuando el cliente desea localizar la residencia donde radica una persona de la UCI y termina cuando se devuelve el mapa centrado en el objetivo.	
Precondiciones:		
Referencias	RF 1, RF 4, RF 5, RF 6	
Prioridad	Crítico	
Flujo Normal de Eventos		
Acción del Actor	Respuesta del Sistema	
1. El cliente consume uno de los siguientes servicios: - “Localizar PersonaPorCI”. - “Localizar PersonaPorSolapin”. - “Localizar PersonaPorUsuario”.	2. El sistema realiza la búsqueda según el servicio seleccionado. - Si seleccionó “Localizar PersonaPorCI “, ver sección “Localizar PersonaPorCI”. - Si seleccionó “Localizar PersonaPorSolapin “, ver sección “Localizar PersonaPorSolapin”. - Si seleccionó “Localizar PersonaPorUsuario “, ver sección “Localizar PersonaPorUsuario”.	
	2.1 El sistema devuelve un error, Usuario no encontrado.	
	3. El sistema actualiza la visualización del mapa.	
<i>Prototipo de Interfaz</i>		



Poscondiciones	
----------------	--

Sección “Localizar PersonaPorCI”

Acción del Actor	Respuesta del Sistema
1. El cliente consume el servicio de “Localizar PersonaPorCI”.	2. El sistema realiza la búsqueda de la persona dada y devuelve el mapa centrado y con una marca en el edificio donde radica la persona.
	2.1. El sistema devuelve un mensaje de error en caso de que el usuario no sea encontrado.

Prototipo de Interfaz



Sección “Localizar PersonaPorSolapin”

Acción del Actor	Respuesta del Sistema

<p>1. El cliente consume el servicio de “Localizar PersonaPorSolapin”.</p>	<p>2. El sistema realiza la búsqueda de la persona dada y devuelve el mapa centrado y con una marca en el edificio donde radica la persona.</p>
	<p>2.1. El sistema devuelve un mensaje de error en caso de que el usuario no sea encontrado.</p>

Prototipo de Interfaz



Sección “Localizar PersonaPorUsuario”

Acción del Actor	Respuesta del Sistema
<p>1. El cliente consume el servicio de “Localizar PersonaPorUsuario”.</p>	<p>2. El sistema realiza la búsqueda de la persona dada y devuelve el mapa centrado y con una marca en el edificio donde radica la persona.</p>
	<p>2.1. El sistema devuelve un mensaje de error en caso de que el usuario no sea encontrado.</p>

Prototipo de Interfaz

	
Poscondiciones	

Tabla 6 Descripción Textual del Caso de Uso Localizar Persona.

3.6. Conclusiones.

Como no se pudo identificar fácilmente procesos de negocios bien estructurados, se tomó la decisión de realizar el modelo de dominio, el cual permitió conocer con mayor profundidad los conceptos presentes en el entorno donde coexiste el problema. Se determinaron los requerimientos funcionales que el sistema debe cumplir y los no funcionales que debe tener. Luego de establecidos los requerimientos funcionales, se pudo identificar fácilmente el actor del sistema y los casos de usos, los cuales fueron plasmados en el diagrama de casos de uso del sistema. La realización de la descripción textual de los casos de uso, posibilitó un mejor entendimiento de cómo funciona el sistema.

CAPÍTULO 4: “Construcción de la solución propuesta”.

4.1. Introducción.

En este capítulo se presentan los principales artefactos para lograr la construcción del sistema como los diagramas de clases del diseño, el modelo de despliegue y el modelo de componentes. Para un mejor entendimiento se explican además los patrones de diseño que se emplean. Se realiza el diseño de caso de prueba de cada caso de uso.

4.2. Descripción de la Arquitectura.

Según Barry Boehm, en el libro “Engineering Context (for Software Architecture)”, en 1995: “Si un proyecto no ha logrado una arquitectura del sistema, incluyendo su justificación, el proyecto no debe empezar el desarrollo en gran escala. Si se especifica la arquitectura como un elemento a entregar, se puede usar a lo largo de los procesos de desarrollo y mantenimiento.”

La arquitectura es una vista estructural de alto nivel, que define estilo o combinación de estilos para una solución. Se puede decir que la arquitectura es esencial para éxito o fracaso de un proyecto. Además la arquitectura de un software es necesaria para comprender el sistema, organizar el desarrollo del mismo, fomentar la reutilización y controlar la evolución del proyecto (**Pressman, 2005**).

Para la realización de la plataforma se definió utilizar la arquitectura en tres capas. Es un estilo de programación, su objetivo primordial es la separación de la capa de presentación, capa de negocio y capa de acceso a datos.

Capa de presentación: es la encargada de presentarle el sistema al usuario, le comunica la información y captura los valores introducidos por el. También es conocida como interfaz gráfica y debe tener la característica de ser "amigable" para el usuario. Esta capa se comunica únicamente con la capa de negocio.

Capa de negocio: es donde se ejecuta la lógica del negocio, se reciben las peticiones del usuario y se envían las respuestas tras el proceso. Esta capa se comunica con la capa de presentación, para recibir

las solicitudes y presentar los resultados, y con la capa de datos, para solicitar al gestor de base de datos la información que se necesite.

Capa de datos: es donde residen los datos y es la encargada de acceder a los mismos. Está formada por el gestor de base de datos PostgreSQL, esta recibe las solicitudes de información desde la capa de negocio y es la encargada de devolvérsela.

4.3. Patrones de diseño.

Los patrones de diseño son soluciones simples y elegantes a problemas específicos y comunes del diseño orientado a objetos. Son soluciones basadas en la experiencia y que se han demostrado que funcionan. No son fáciles de entender, pero una vez entendido su funcionamiento, los diseños son más flexibles, modulares y reutilizables. **(Gracia, 2005)**

La utilización de patrones de diseño es una práctica recomendada en la ingeniería de software, pues ayuda a la estandarización, flexibilidad y robustez en la arquitectura. Cada patrón resuelve un problema, pero es importante saber combinarlos de manera adecuada cuando es necesario.

4.3.1. Patrones para asignar Responsabilidades (GRASP).

Los patrones de diseño GRASP (General Responsibility Assignment Software Patterns), describen los principios esenciales de la asignación de responsabilidades a objetos, expresados en forma de patrones.

Patrón Experto:

Asigna una responsabilidad al experto en información. Está enfocado en resolver el problema de ¿Cuál es el principio fundamental en virtud del cual se asignan las responsabilidades en el diseño orientado a objetos? La solución se basa en asignar una responsabilidad al experto en información: la clase que cuenta con la información necesaria para cumplir la responsabilidad.

Beneficios:

- ❖ Se conserva el encapsulamiento, ya que los objetos se valen de su propia información para hacer lo que se les pide.

- ❖ El comportamiento se distribuye entre las clases que cuentan con la información requerida alentando con ello definiciones de clases sencillas y más cohesivas que son fáciles de comprender y mantener.

Ejemplo:

El patrón se emplea en la clase MapControl, es la encargada de realizar todas las funciones de localización ya que es la única que cuenta con los atributos necesarios para resolver el problema.

Patrón Creador:

Permite asignar quién debería ser el responsable de la creación de una nueva instancia de alguna clase. Guía la asignación de responsabilidades relacionadas con la creación de objetos.

El propósito general de este patrón es encontrar un creador que debemos conectar con el objeto producido en cualquier evento.

Beneficios:

- ❖ Brinda soporte de bajo acoplamiento, lo cual admite menos dependencias con respecto al mantenimiento y mejores oportunidades de reutilización. Es probable que el acoplamiento no aumente, pues la clase creada tiende a ser visible a la clase creador, debido a las asociaciones actuales que llevaron a elegirla como el parámetro adecuado.

Ejemplo:

El patrón se emplea en la clase BeginEndControl, esta clase es la encargada de crear y de inicializar la instancia de PostgresDB, en este caso BeginEndControl almacena a PostgresDB, tiene todos los datos para inicializar su constructor y es la encargada de contenerlo y usarlo.

Patrón Controlador:

El propósito de este patrón es asignar la responsabilidad de controlar el flujo de eventos del sistema, a clases específicas. El controlador no realiza estas actividades, las delega en otras clases con las que mantiene un modelo de alta cohesión.

Beneficios.

- ❖ Mayor potencial de los componentes reutilizables.
- ❖ Razonamiento sobre el estado de los casos de uso:
- ❖ Asegurar que la secuencia es válida.
- ❖ Razonar sobre el estado actual.

Ejemplo:

El patrón se pone de manifiesto en la clase Servicio, esta clase es la encargada de recibir los datos del usuario y enviarlos a las clases ServiceControl y MapControl, estas son las encargadas de la lógica del negocio.

Patrón Bajo Acoplamiento:

Este patrón especifica cómo dar soporte a una dependencia escasa y a un aumento de la reutilización, enfocándose en asignar una responsabilidad para mantener bajo acoplamiento.

El bajo acoplamiento estimula la asignación de responsabilidades de forma tal que la inclusión de éstas no incremente el acoplamiento, creando clases más independientes y con mayor resistencia al impacto de los campos, que aumentan la productividad y la posibilidad de reutilización.

El bajo acoplamiento soporta el diseño de clases más independientes, que reducen el impacto de los cambios. Es válido precisar que este patrón no puede verse de forma independiente a los patrones Experto y Alta Cohesión, sino más bien incluirse como otro de los principios del diseño que influyen de forma determinante a la hora de la asignación de responsabilidades.

Beneficios.

- ❖ No afectan los cambios en otros componentes.
- ❖ Conveniente para reutilizar.
- ❖ Fácil de entender de manera aislada.

Ejemplo:

De manera general el patrón se pone de manifiesto en todo el diseño de clases, no se implementa ningún tipo de herencia o se abusa de la dependencia entre clases, de esta forma se logra la reutilización de clases como PostgresDB y Servicios.

Patrón Alta Cohesión:

El objetivo de este patrón es asignar responsabilidades de tal forma que la cohesión siga siendo alta. Una alta cohesión caracteriza a las clases con responsabilidades estrechamente relacionadas que no realicen un trabajo enorme.

Beneficios.

- ❖ Incrementa la claridad y facilidad de comprensión del diseño.
- ❖ Simplifica el mantenimiento.

- ❖ Soporta bajo acoplamiento. Una mala cohesión lleva a mal acoplamiento y viceversa.
- ❖ El grano fino de funcionalidad altamente relacionada incrementa la reutilización.

Ejemplo:

El patrón se pone de manifiesto en conjunto con el de Bajo Acoplamiento, las tareas se distribuyen en varias clases y permiten la reutilización de las funciones, las clases Servicios y GlobalManager son ejemplo de esto.

4.3.2. Patrón Singleton.

Garantiza que una clase sólo tenga una instancia y proporciona un punto de acceso global a esta instancia, característica que tributa al óptimo funcionamiento del sistema. De esta forma se controla el acceso a las clases. Para la puesta en práctica de este patrón se creó una única instancia del objeto mapa para que pueda ser utilizado de una manera global para cada una de las conexiones.

4.4. Diseño del Sistema.

Un diagrama de clases del diseño Web permite describir gráficamente un conjunto de clases, las relaciones entre ellas, logrando de esta forma una muestra del sistema muy importante para la implementación.

A continuación en la figura 5 se muestra el diagrama de clases del diseño del Caso de Uso DCD Localizar Persona. Los diagramas de diseño correspondientes a los demás Casos de Uso pueden ser consultados en el [Anexo2](#).

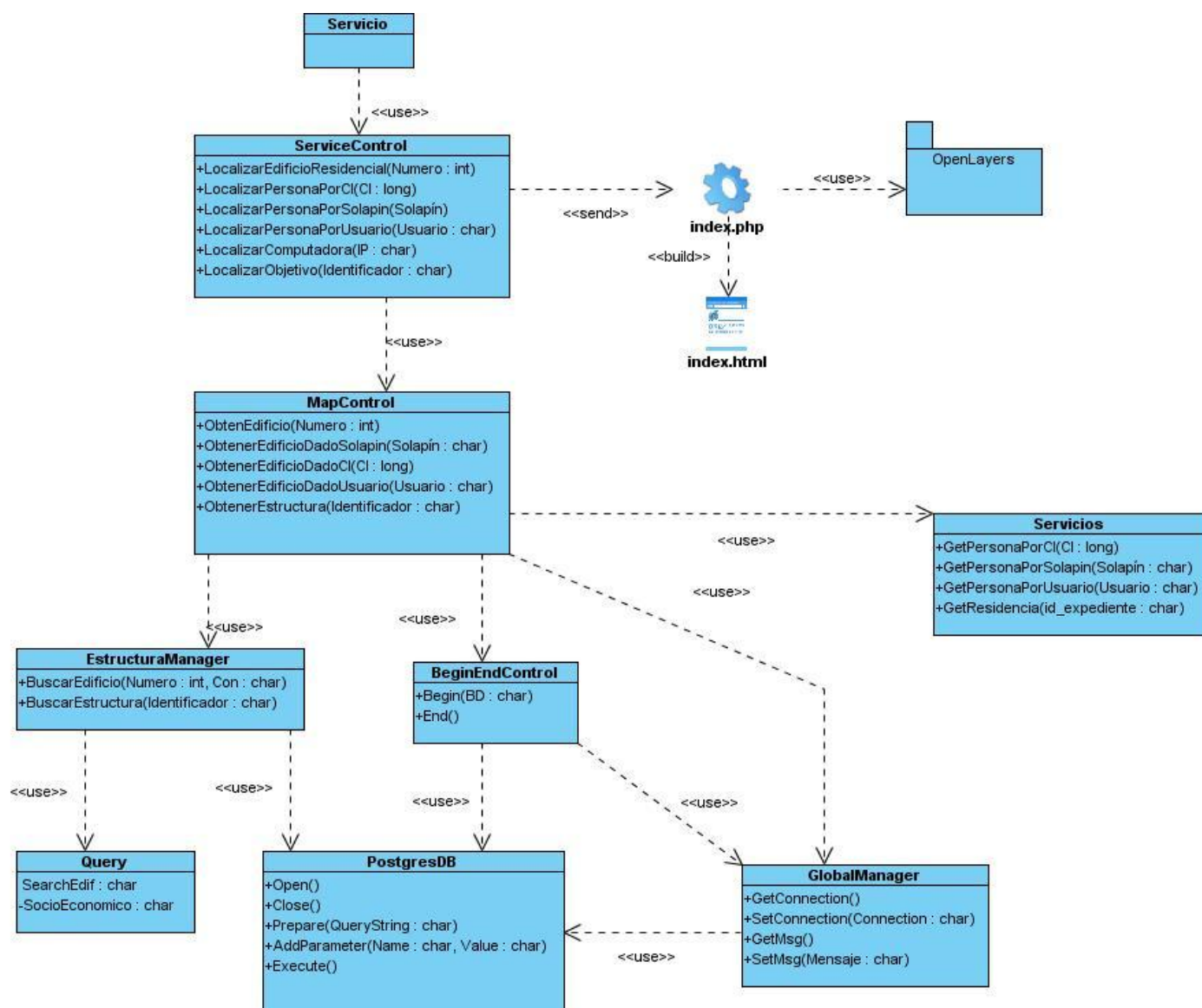


Figura 5 Diagrama de clases del diseño del Caso de Uso Localizar Persona.

4.4.1. Diseño de la Base de Datos.

Para el óptimo funcionamiento del PLASMA se trabaja con dos Base de Datos: la base de datos espacial del SIG-UCI (Figura 6) y la base de datos socioeconómica (Figura 5). Las bases de datos se encuentran normalizadas con el objetivo de evitar ambigüedades, e inconsistencia en los datos que provoquen resultados no esperados por el sistema una vez interactuando con el mismo.

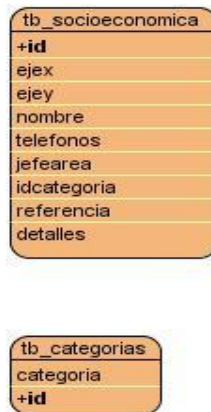


Figura 6 Modelo Entidad-Relación del SIG UCI Socioeconómico.

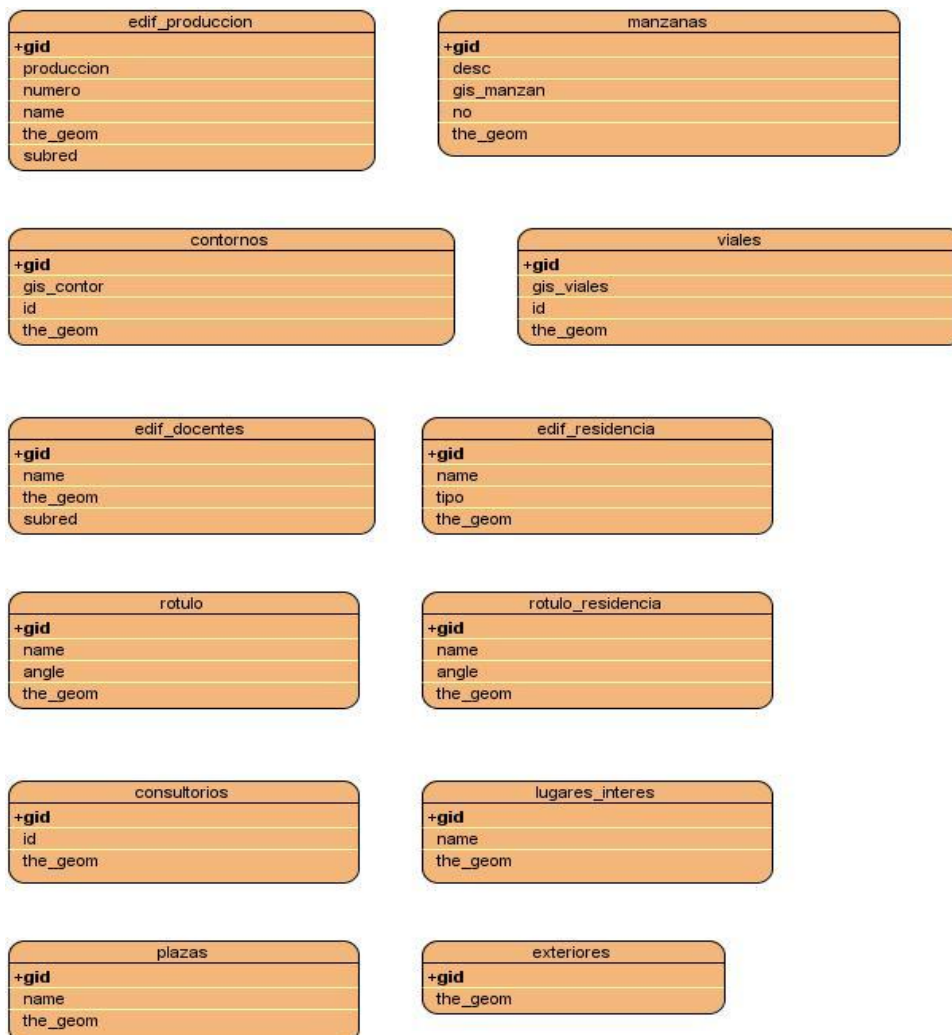


Figura 7 Modelo Entidad-Relación del SIG UCI

4.5. Modelo de Implementación

Un modelo de implementación consiste es una visión general de lo que tiene que ser implementado, y un apartado para cada iteración con los componentes y subsistemas a implementar durante esa iteración, así como de los resultados software que se han de obtener y el testeo que se ha de realizar sobre ellos. (UCI, 2010)

4.5.1. Diagrama de Componentes

En la figura 8 se muestra la vista general del diagrama de componentes correspondiente al sistema desarrollado. Para un mejor entendimiento de dicho diagrama se expone en el [Anexo1](#) una vista detalla de los paquetes Presentación, Negocio y Recursos, así como los componentes que contiene cada paquete en específico.

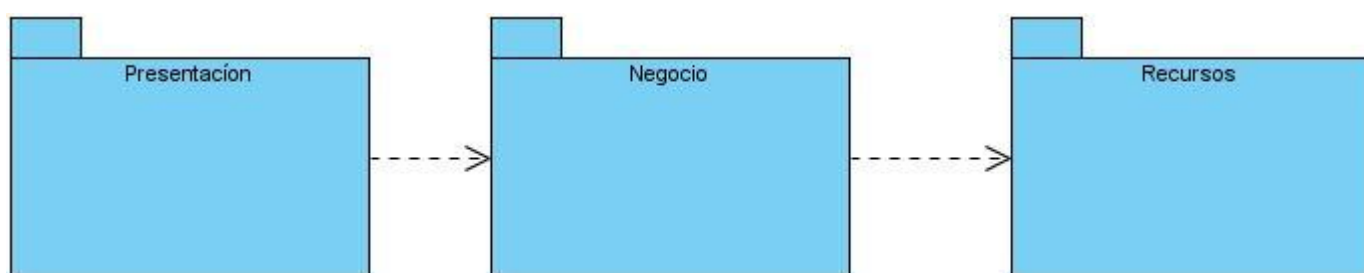


Figura 8 Vista General Del Diagrama de Componentes.

4.6. Diagrama de Despliegue.

Para lograr un entendimiento de la arquitectura de la aplicación en tiempo de ejecución se desarrolló el Diagrama de Despliegue. En este se evidencia la configuración de los elementos de hardware (nodos) y muestra cómo los elementos y artefactos del software se trazan en esos nodos. Describe el camino en el cual los componentes específicos corresponden al despliegue de la aplicación.

Dicho diagrama está representado por cuatro nodos. Una PC Cliente, la cual se comunica con el servidor Web a través del protocolo HTTP, el mismo, haciendo uso del protocolo ADO y TCP/IP, establece comunicación con el servidor de Base de Datos y el servidor de Mapas respectivamente. En la siguiente figura se muestra el Diagrama de Despliegue.

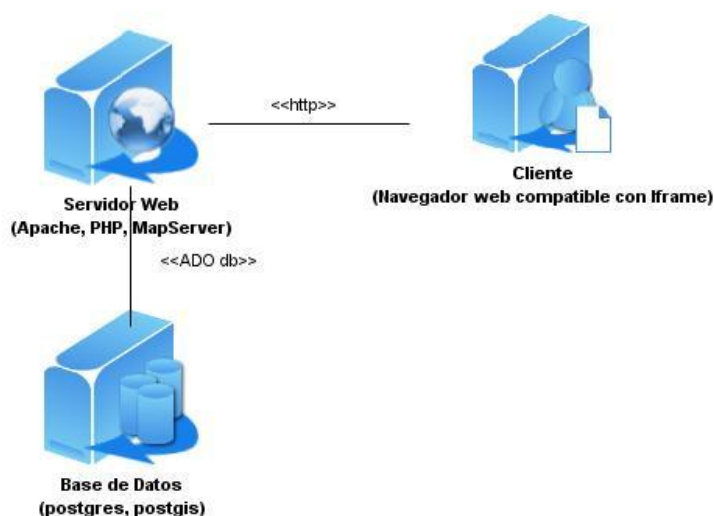


Figura 9 Diagrama de Despliegue.

4.7. Validación de la solución propuesta.

Las pruebas de un software están constituidas por un conjunto de herramientas, técnicas y métodos que tributan a la excelencia del desempeño de un programa. Técnicamente consisten en inspeccionar manualmente el código o hacer pruebas de ensayos (ejecutar el software y ver los resultados) y el objetivo principal es encontrar errores o defectos en el software.

A continuación se muestra el diseño de casos de prueba de caja negra del Casos de Usos Localizar persona. Los diseños de caso de pruebas correspondientes a los demás Casos de Uso pueden ser consultados en el [Anexo3](#).

4.7.1. Diseño de prueba del Caso de Uso Localizar Persona.

1. **Descripción General:** el caso de uso se inicia cuando el usuario utiliza uno de los métodos del servicio para localizar persona (Localizar Persona Por Solapín, Localizar Persona Por Carne Identidad, Localizar Persona Por Usuario) y finaliza cuando devuelve el mapa con la localización.

2. Secciones a probar en el Caso de Uso.

Nombre de la sección	Escenarios de la sección	Descripción de la funcionalidad
SC 1: Localizar Persona Por Solapín	EC 1.1: Localizar Persona Por Solapín	El sistema brinda un servicio Web que ofrece la posibilidad de localizar una persona en el mapa y visualizar su residencia dado su solapín.
	EC 1.2: Marcar localización	El sistema marca con una tachuela en el mapa el edificio de residencia de la persona localizada.
	EC 1.3: Tratamiento de errores	El sistema devuelve un SoapFault “No existe Persona con ese solapín en la Base de Datos”.
SC 2: Localizar Persona Por Carne Identidad	EC 2.1: Localizar Persona Por Carne Identidad	El sistema brinda un servicio Web que ofrece la posibilidad de localizar una persona en el mapa y visualizar su residencia dado su carne de identidad.
	EC 2.2: Marcar localización	El sistema marca con una tachuela en el mapa el edificio de residencia de la persona localizada.
	EC 2.3: Tratamiento de errores	El sistema devuelve un SoapFault “No existe Persona con ese Carné de Identidad en la Base de Datos”.
SC 3: Localizar Persona Por Usuario	EC 3.1: Localizar Persona Por Usuario	El sistema brinda un servicio Web que ofrece la posibilidad de localizar una persona en el mapa y visualizar su residencia dado su usuario.
	EC 3.2: Marcar localización	El sistema marca con una tachuela en el mapa el edificio de residencia de la persona localizada.
	EC 3.3: Tratamiento de errores	El sistema devuelve un SoapFault “No existe Persona con ese usuario en la Base de Datos”.

Tabla 7 Diseño de prueba del Caso de Uso Localizar Persona.

3. Descripción de las variables.

No	Nombre de campo	Clasificación	Valor Nulo	Descripción
1.	Solapín	Parámetro	No	Se tiene que seleccionar el servicio de “Localizar Persona Por Solapín”
2.	CI	Parámetro	No	Se tiene que seleccionar el servicio de “Localizar Persona Por Carne Identidad”
3.	Usuario	Parámetro	No	Se tiene que seleccionar el servicio de “Localizar Persona Por Usuario”

Tabla 8 Descripción de las variables.

SC 1 Localizar Persona Por Solapín.

Escenario	Variable 1 Solapín	Respuesta del Sistema	Flujo Central
EC 1.1: Localizar Persona Por Solapín	V	El sistema localiza la persona en el mapa.	1- Seleccionar el método de Localizar Persona Por Solapín. 2- Introduce como parámetro el solapín de la persona. 3- Devuelve el mapa en un <iframe> centrado en la localización de la persona.

Escenario	Variable 1 Solapín	Respuesta del Sistema	Flujo Central
EC 1.2: Marcar localización	V	Marca la localización	1- Seleccionar el método de Localizar Persona Por Solapín. 2- Introduce como parámetro el solapín de la persona. 3- Devuelve el mapa en un <iframe> centrado en la localización de la persona. 4- Marca en el mapa la localización con una tachuela.
EC 1.3: Tratamiento de errores	V	El sistema devuelve un SoapFault “No existe usuario asociado a ese dato”.	1- Seleccionar el método de Localizar Persona Por Solapín. 2- Introduce como parámetro el solapín de la persona. 3- En el caso de que el usuario no exista o que el valor introducido no sea válido el sistema devuelve una Excepción de tipo SoapFault.

Tabla 9 SC 1 Localizar Persona Por Solapín.

4. SC 2 Localizar Persona Por Carne Identidad.

Escenario	Variable 1 Solapín	Respuesta del Sistema	Flujo Central
EC 1.1: Localizar Persona Por Carne Identidad	V	El sistema localiza la persona en el mapa.	<ol style="list-style-type: none"> 1- Seleccionar el método de Localizar Persona Por Carne Identidad. 2- Introduce como parámetro el CI de la persona. 3- Devuelve el mapa en un <iframe> centrado en la localización de la persona.
EC 1.2: Marcar localización	V	Marca la localización	<ol style="list-style-type: none"> 1- Seleccionar el método de Localizar Persona Por Carne Identidad. 2- Introduce como parámetro el CI de la persona. 3- Devuelve el mapa en un <iframe> centrado en la localización de la persona. 4- Marca en el mapa la localización con una tachuela.
EC 1.3: Tratamiento de errores	V	El sistema devuelve un SoapFault "No existe usuario asociado a ese dato".	<ol style="list-style-type: none"> 1- Seleccionar el método de Localizar Persona Por Carne Identidad. 2- Introduce como parámetro el CI de la persona. 3- En el caso de que el usuario no exista o que el valor introducido no sea válido el sistema devuelve una Excepción de tipo SoapFault.

5. SC 3 Localizar Persona Por Usuario.

Escenario	Variable 1 Solapín	Respuesta del Sistema	Flujo Central
EC 1.1: Localizar Persona Por Usuario	V	El sistema localiza la persona en el mapa.	<ol style="list-style-type: none"> 1- Seleccionar el método de Localizar Persona Por Usuario. 2- Introduce como parámetro el Usuario de la persona. 3- Devuelve el mapa en un <iframe> centrado en la localización de la persona.
EC 1.2: Marcar localización	V	Marca la localización	<ol style="list-style-type: none"> 1- Seleccionar el método de Localizar Persona Por Usuario. 2- Introduce como parámetro el Usuario de la persona. 3- Devuelve el mapa en un <iframe> centrado en la localización de la persona. 4- Marca en el mapa la localización con una tachuela.
EC 1.3: Tratamiento de errores	V	El sistema devuelve un SoapFault "No existe usuario asociado a ese dato".	<ol style="list-style-type: none"> 1- Seleccionar el método de Localizar Persona Por Usuario. 2- Introduce como parámetro el Usuario de la persona. 3- En el caso de que el usuario no exista o que el valor introducido no sea válido el sistema devuelve una Excepción de tipo SoapFault.

Tabla 10 SC 3 Localizar Persona Por Usuario.

4.7.2. Pruebas de estrés y rendimiento

Para realizar las pruebas de carga y estrés a la aplicación se usó el JMeter, es una herramienta Java desarrollada dentro del proyecto Jakarta, que permite realizar Pruebas de Rendimiento y Pruebas Funcionales sobre Aplicaciones Web. Utilizar JMeter en aplicaciones Web para la comprobación de los recursos del sistema, supone una mayor efectividad en el proceso y en la fiabilidad de los resultados. JMeter como herramienta de prueba dispone de varios componentes que facilitan la elaboración de los escenarios de prueba con la ventaja de simular para cada uno de esos escenarios miles de usuarios.

Realizando un resumen de los datos brindados por las pruebas, en un ambiente con las características de software y de hardware como las descritas en los requisitos no funcionales, se obtuvo que para un total de 50 usuarios (hilos) en el escenario de cargar el mapa inicial la aplicación generó un total de 360 kb/seg de transferencia de datos lo que incurrió en un rendimiento de 1.4segundos. Se demuestra que la aplicación es estable ya que para estos 50 usuarios se mantuvo prestando servicios todo el tiempo sin incurrir en fallos. El tiempo de respuestas del servidor se encuentra por debajo del tiempo que se especificaba en los requerimientos del sistema por lo que se concluye que la aplicación esta lista para el despliegue.

A continuación se muestra el comportamiento que tuvo la aplicación para diferentes números de usuarios.

Cantidad de Usuarios	Rendimiento	Tranferencia de datos
25	1.3seg	320 kb/seg
50	1.6seg	360 kb/seg
100	3.1seg	430 kb/seg

Tabla 11 Pruebas de carga y estrés.

4.8. Conclusiones.

Mediante la realización del Modelo de Despliegue, se describe como está distribuida física y lógicamente la arquitectura del sistema y sus conexiones. La realización del Modelo de Implementación, permitió detallar los componentes creados para el desarrollo de la aplicación y la

relación entre ellos. Se llevó a cabo la elaboración de los diseños de pruebas de Caja Negra de las principales funcionalidades del sistema y se le realizaron pruebas de balance y de estrés las cuales fueron satisfactorias, las cuales dieron como resultado que el sistema está preparado tecnológicamente para su puesta en marcha de acuerdo a los requerimientos funcionales y no funcionales anteriormente definidos.

Conclusiones

Luego de haber realizado un estudio de los sistemas que existen para la compartición de la información geográfica, se pudo reconocer la importancia de incluir en nuestra solución el uso de OpenLayers. Estas librerías permiten soluciones rápidas y eficientes, debido a que su arquitectura es flexible en cuanto a diferentes plantillas, permite la personalización, ya que se puede adaptar a cualquier negocio con mucha facilidad y ayuda a tomar mejores decisiones en menos tiempo.

El hecho de que las herramientas y tecnologías empleadas para el desarrollo de PLASMA sean totalmente libres, posibilita que pueda ser distribuido a cualquier empresa interesada, modificado a sus necesidades y contar con un grupo de desarrollo cubano, encargado de manejar el sistema y desarrollar planes para su mejora e implementación y para actualizar la información.

Se obtuvo un sistema, que además de localizar permite realizar una navegación básica sobre los objetos georreferenciados en la cartografía. Además se puede contar con la documentación técnica del proceso de desarrollo, la cual permite un mejor entendimiento del sistema desarrollado, y puede ser utilizada como guía para futuras actualizaciones o para continuar el desarrollo del sistema, llevando a cabo la implementación de nuevas funcionalidades.

Recomendaciones

Luego de haber concluido la plataforma y cumplido los objetivos trazados, se plantean las siguientes recomendaciones:

- ❖ Incrementar el número de servicios de mapas que se ofertan con PLASMA.
- ❖ Desplegar el sistema en entidades con recursos geográficos más complejos (mayor cartografía e información socioeconómica).
- ❖ Desarrollar versiones de PLASMA que manipulen objetos de mapas más complejos como los Web Features Services (WFS).

Bibliografía:

1. Carlos E. Arcancel: **Web Map Services: La nueva era de información geográfica digital.** Revista Ciencia e Innovación Tecnológica. Número 126. **2008.**
2. Belmonte, Silvina y Núñez, Virgilio: **Desarrollo de modelos hidrológicos con herramientas SIG.** Instituto de Recursos Naturales y Ecodesarrollo. Facultad de Ciencias Naturales. Universidad Nacional de Salta. **2009.**
3. Consortium, Open Geospatial: **Web Map Server Implementation Specification.** **2006.**
4. Encina, Eduardo Felipe Parraguez: **Infraestructura de Datos Espaciales para Sistemas de Universidad de Santiago de Chile.** **2004.**
5. Webster C J, Bracken I J: **Exploring the discriminating power of texture in urban image analysis.** **1992.**
6. ESRI: **1995. Disponible en: <http://www.gis.com/content/what-gis>.**
7. Samuel Puentes Fernández: Facultad de Biología, Universidad de La Habana. **2005. Disponible en: <http://fbio.uh.cu/sites/bioinfo/glosario.html>.**
8. Secretaria Ejecutiva de la IDERC : Infraestructura de Datos Espaciales de la República de Cuba. **2004.**
9. Ayuda de Google: **Acerca de Google Maps. 2010. Disponible en: <http://maps.google.es/support/bin/answer.py?hl=es&answer=7060>**
10. Guglielmetti Marcos, Yanover David, Lanzillotta Analia, Alsina Guillem: **Definición de Plataforma. 2005. Disponible en: <http://www.mastermagazine.info/termino/6325.php>**
11. Ibar Jacobson, Grady Booch, James Rumbaugh: **El Proceso Unificado de Desarrollo de Software.** Madrid: Pearson Educación.S.A. **2000.**
12. Schmuller, Joseph: **Aprendiendo UML en 24 Horas.** México: PEARSON EDUCACION. **2000**
13. Eclipse Fundation :[Citado el: 2 de febrero de 2010.] **Disponible en: <http://www.eclipse.org/>.**

14. Gracia Joaquín: ***Desarrollo de Software Orientado a Objetos***. 2005. Disponible en:
<http://www.ingenierosoftware.com/analisisydiseno/uml.php>.
15. Sun Microsystem. Netbeans.org [Citado el: 01 de 03 de 2009.] **Disponible en:**
www.netbeans.org
16. Méndez Roldán, Isuel : ***El servidor de mapas MapServer, una solución recomendada para la representación de Información Geoespacial***. 2009. Disponible en:
http://www.mappinginteractivo.com/plantilla-egeo.asp?id_articulo=1600
17. Jacobson,G. Booch: ***El Proceso Unificado de Desarrollo de Software***. La Habana, Editorial Félix Varela. **2004**.
18. Pressman, Roger: ***Ingeniería de Software. Un enfoque práctico. Vol I, vol II***. Editorial Félix Varela. La Habana, Cuba. **2005**.
19. Serrano Rosales, Carlos Luis, and Solangel Rodríguez Vásquez: ***Desarrollo de Biblioteca de Métodos Numéricos (BMN), referente a Sistemas de Ecuaciones Lineales, Sistemas de Gran Dimensión y Poco Densos e Integración Numérica***. Trabajo de diploma, Habana: UCI. **2008**.

Webgrafía:

- 1- <http://www.gis.com/whatisgis/index.html>
- 2- <http://docs.sun.com/app/docs/doc/820-3091/6nel9nuoa?l=es&a=view> .
- 3- <http://fbio.uh.cu/sites/bioinfo/glosario.html>.
- 4- <http://www.opengeospatial.org/standards/wms>.
- 5- <http://www.opengeospatial.org/standards/wfs>.
- 6- <http://www.navactiva.com/web/es/descargas/pdf/atic/sig1.pdf>
- 7- <http://www.scourdesign.com/articulos/tutoriales/php/tutoriales-php-mysql-servicios-web-soap.php>
- 8- <http://www.desarrolloweb.com/articulos/1883.php>
- 9- <http://www.iderc.co.cu/>
- 10- <http://www.geosur.info/geosur/iirsa/>

GLOSARIO

Software: conjunto de programas, instrucciones y reglas informáticas para ejecutar ciertas tareas en una computadora.

Hardware: conjuntos de componentes que integran la parte material de una computadora. Componentes físicos de una tecnología.

API: una API (acrónimo en inglés de Application Programming Interface) o Interface de Programación de Aplicaciones (en español) es un conjunto de funciones que facilitan el intercambio de mensajes o datos entre dos aplicaciones.

IDE: un entorno de desarrollo integrado (acrónimo en inglés de Integrated Development Environment), es un programa informático compuesto por un conjunto de herramientas de programación.

Open Source (en español Código Abierto): es el término con el que se conoce al software distribuido y desarrollado libremente.

Requerimiento: condición o cualidad que debe cumplir algo o alguien.

SIG: Sistema de Información Geográfica.

Web Service: Servicio Web.

WMS: Servidor Web de Mapas.

UCI: Universidad de las Ciencias Informáticas.