

# Análisis del Sistema de Información Geográfica genérico basado en la Tecnología TileCache

---

**AUTOR:** Silet Muñiz Guevara.

**TUTOR:** Ing. Alain Leon Companioni.

# Dedicatoria

---

A mi madre que es la principal protagonista en mi vida,

A mi abuelita que ha sido durante toda mi vida mi motor impulsor.

A mi hermano que es mi tesoro más valioso.

A mi esposo que gracias a su amor y apoyo he podido salir adelante.

A mi padre Enio que es el mejor padre que he tenido.

A mi abuelita Estela y mi abuelito Juan que aunque estén lejos me  
lleban presente.

A mis tíos José, Guille, Ernesto.

# Agradecimientos

---

Le doy gracias a Dios, por haberme dado la vida y mantenerme hasta estos momentos con todos mis seres queridos a mi lado.

A mi madre por haberme educado por un buen camino y haber hecho de mí la mujer que soy.

A mi abuela Margo que ha sido mi ángel de la guardia y constituye mi segunda madre.

A mi hermano que lo adoro con la vida y que me ha demostrado que yo soy su mayor ejemplo y trataré de no decepcionarlo nunca.

A mi esposo que ha estado en todos los momentos conmigo, en las buenas y en las malas.

A mi padre el chino que me ha demostrado que no hay que tener la misma sangre en las venas para ser un buen padre.

A mis tíos maternos a mi primita Glenda y su familia que son mi familia doble.

A mis abuelos paternos Estela y Juan y a toda mi familia.

A todas mis amiguitas Saili, Ana Iris, Sachie, Misbel, Mailin, Mary, Yara, Lucia y otras que se me quedan.

A todos los profesores, al tribunal a mi tutor Alain y a mi oponente.

A todo mi grupo viejo, a todo mi grupo nuevo.

A todas las personas de la antigua facultad diez a las amistades de mi novio y la mía en fin a todos los que me conocen.

# Declaración de Autoría

---

Declaro que soy la única autora del presente trabajo y autorizo a la Universidad de las Ciencias Informáticas (UCI) a que haga uso del mismo de la manera que mejor estime.

Para que así conste firmo la presente a los \_\_\_\_ días del mes de \_\_\_\_\_ del año \_\_\_\_\_.

Saillet Muñiz Guevara

Ing. Alain Leon Companioni

\_\_\_\_\_

\_\_\_\_\_

Firma de la autora

Firma del tutor

# Datos de Contacto

---

**TUTOR:** Ing. Alain Leon Companioni.

**Dirección de la institución:** Carretera de San Antonio. Boyeros. Ciudad Habana.

**Correo electrónico:** [acompanioni@uci.cu](mailto:acompanioni@uci.cu)

**Título de la especialidad de graduado:** Ingeniero en Ciencias Informáticas.

**Año de graduación:** 2010.

**Institución donde se graduó:** UCI.

# Resumen

---

En la actualidad los Sistemas de Información Geográfica (SIG) se han consolidado como una industria. Los SIG permiten combinar información básica para obtener información derivada, relacionando información de cualquier tipo (base de datos) con una localización geográfica (mapa). Estos sistemas pueden ser utilizados para actividades de planificación, distribución, logística, mercadeo o simplemente para ubicar algún dato de interés en un lugar determinado. La Universidad de las Ciencias Informáticas a lo largo de los años ha adquirido gran experiencia en el desarrollo de grandes proyectos contribuyendo de una forma u otra en la formación de profesionales con grandes conocimientos en áreas de la informática. En el Departamento Geoinformática se creó el proyecto Aplicativo SIG perteneciente al centro de Desarrollo y Producción de Software Geoinformática y Sistemas Digitales GEySED donde su principal objetivo es crear personalizaciones a GENESIG que es la plataforma soberana para el desarrollo de SIG en la universidad. Esta plataforma se basa en la tecnología MapServer (Servidor de Mapas), que realiza con eficacia la visualización de la información, pero es lento, debido al elevado grupo de datos a representa. Actualmente existen diversas tecnologías para agilizar el proceso de visualización de los mapas, entre ellas TileCache. Teniendo en cuenta las ventajas de esta tecnología y la necesidad de acelerar el proceso de visualización de mapas en los SIG desarrollados en el proyecto Aplicativo SIG, la presente investigación está enfocada en la realización de la documentación que contribuya al diseño e implementación de un Sistema de Información Geográfica que sirva de base para el desarrollo de otros SIG basado en la tecnología TileCache.

## **Palabras Claves:**

MapServer, mapas, Sistemas de Información Geográfica, TileCache.

# Resumen

---

Nowadays, Geographic Information Systems (GIS) have emerged as an industry. GIS combines basic information to obtain derived information, relating information of any kind (database) with a geographic location (map). These systems can be used for planning, distribution, logistics, marketing, or simply to locate any data of interest in a particular location. The University of Information Science over the years has acquired considerable experience in developing large projects contributed in one way or another in the training of professionals with vast knowledge in the areas of computing. In the Geoinformatics Department was created the Project Aplicativos SIG, belonging to the Center GEYSED, where his main goal is to create customizations to GENESIG, sovereign that is the platform for the development of GIS in the university. This platform technology is based on MapServer (Map Server), which effectively makes the visualization of information, but is slow due to the large group of data to represent. There are currently several technologies to streamline the process of viewing the maps, including TileCache. Taking into account the benefits of this technology and the need to accelerate the process of viewing GIS maps developed under the GIS applications, this investigation is focused on the completion of the documentation that contribute to the design and implementation of an Information System Geographic as a basis for the development of other technology-based GIS TileCache.

Keywords:

Geographic Information Systems, MapServer, maps, TileCache.

# Índice

---

Índice de tablas.....	I
Introducción.....	1
<b>1. Capítulo I: Fundamentación Teórica.....</b>	<b>5</b>
Introducción.....	5
<b>1.1. Conceptos asociados al dominio del problema.....</b>	<b>5</b>
1.1.1. Mapa.....	5
1.1.2. WMS.....	5
1.1.3. MapServer.....	6
1.1.4. Mapscript.....	6
1.1.5. TileCache.....	6
<b>1.2. Objeto de estudio: Los Sistema de Información Geográfica.....</b>	<b>7</b>
1.2.1. Desarrollo de los SIG.....	7
1.2.2. Desarrollo de los SIG en la UCI.....	8
<b>1.3. Soluciones existentes.....</b>	<b>9</b>
Especificaciones.....	10
1.3.1. WMS Tile Caching.....	10
1.3.2. Tile Map Service.....	10
1.3.3. Web MapTiling Service.....	11
Implementaciones de WMS Tile Caching.....	14
1.3.4. GeoWebCache.....	14
1.3.5. TileCache.....	15
<b>1.4. Comparación de soluciones existentes.....</b>	<b>17</b>
<b>1.5. Metodologías de Desarrollo de Software.....</b>	<b>18</b>
Metodologías Robustas.....	18
1.5.1. Rol de analista.....	19
<b>1.6. Herramientas CASE.....</b>	<b>21</b>
<b>1.7. Visual Paradigm.....</b>	<b>21</b>
<b>1.8. El Lenguaje Unificado de Modelado (UML).....</b>	<b>22</b>
1.8.1. Objetivos de UML. (29).....	22
Conclusiones.....	23
<b>2. Capítulo 2: Análisis del sistema propuesto.....</b>	<b>24</b>
Introducción.....	24

# Índice

---

2.1. Modelo de Dominio.....	24
2.1.1. Eventos principales del entorno.....	24
2.1.2. Diagrama de clases del Modelo de Dominio.....	25
2.2. Glosario de Términos del Dominio .....	25
2.3. Levantamiento de Requisitos .....	26
2.3.1. Requisitos Funcionales .....	27
2.3.2. Requisitos no funcionales.....	27
2.4. Descripción del Sistema Propuesto.....	28
2.4.1. Descripción de los actores.....	28
2.4.2. Diagrama de Casos de Uso del Sistema .....	28
2.5. Descripción Textual de Casos de Uso del sistema .....	29
2.5.1. Descripción del Caso de Uso Seleccionar capas.....	29
2.5.2. Descripción del Caso de Uso Localizar. ....	30
2.5.3. Descripción del Caso de Uso Enfocar mapa. ....	31
2.5.4. Descripción del Caso de Uso Mover mapa. ....	33
2.5.5. Descripción del Caso de Uso Visualizar coordenadas.....	34
2.5.6. Descripción del Caso de Uso Visualizar escala gráfica.....	34
2.6. Análisis.....	35
2.6.1. Diagrama de Clases del Análisis.....	35
2.7. Diagramas de Colaboración. ....	38
Conclusiones.....	42
3. Capítulo 3: Factibilidad y Validación del Sistema Propuesto .....	43
Introducción .....	43
3.1. Estudio de factibilidad del sistema propuesto.....	43
Estimación por Puntos de Caso de Uso (Karnar).....	44
3.1.1. Paso1: Cálculo de Puntos de Casos de Uso sin ajustar.....	44
3.1.2. Paso2: Cálculo de Puntos de Casos de Uso ajustados.....	46
3.1.3. Paso3: Estimación de esfuerzo a través de los puntos de casos de uso. ....	49
3.1.4. Paso4: Calcular el esfuerzo de todo el proyecto.....	50
3.2. Análisis Costo-Beneficio del proyecto. ....	51
3.3. Validación del sistema propuesto.....	52
3.3.1. Método de Delphi .....	53
3.3.2. Test de Turing .....	54

# Índice

---

3.3.3. Comparación de los resultados de salida.....	54
3.3.4. Prototipos .....	54
3.3.5. Matriz de trazabilidad.....	55
3.3.6. Revisión de Requisitos.....	55
3.3.7. Auditoria .....	56
3.4. Desarrollo de las técnicas de validación:.....	56
3.4.1. Desarrollo de la matriz de trazabilidad:.....	56
3.4.2. Desarrollo de la auditoria: .....	57
3.4.3. Desarrollo de las revisiones: .....	58
Conclusiones.....	59
Conclusiones .....	60
Bibliografía .....	62

# Índice

---

## Índice de figuras.

Figura 1 Funcionalidades de los SIG. ....	8
Figura 2 Malla de tiles en TMS. ....	11
Figura 3 Modelo piramidal de tiles de una capa de un servidor WMTS. ....	12
Figura 4 Espacio de una matriz de tiles (TileMatrix). ....	13
Figura 5 Diagrama UML de la interface WMTS. ....	14
Figura 6 Pirámide de tiles. ....	16
Figura 7 Diagrama de TileCache. ....	17
Figura 8 Diagrama del Modelo de Dominio. ....	25
Figura 9 Diagrama de Casos de Uso del Sistema. ....	29
Figura 10 La validación en el proceso de requisitos. ....	53
Figura 11 Prototipo en el proceso de validación de requisitos. ....	55

## Índice de tablas

Tabla 1 Actividades y Responsabilidades del Analista del sistema en RUP. ....	20
Tabla 2 Diagrama del Modelo de Dominio. ....	28
Tabla 3 Caso de Uso Seleccionar capas. ....	30
Tabla 4 Caso de Uso Localizar. ....	31
Tabla 5 Caso de Uso Enfocar mapa. ....	32
Tabla 6 Caso de Uso Mover mapa. ....	34
Tabla 7 Caso de Uso Visualizar coordenadas. ....	34
Tabla 8 Caso de Uso Visualizar escala gráfica. ....	35
Tabla 9 Enfocar Mapa. ....	36
Tabla 10 Escala Gráfica. ....	36
Tabla 11 Localizar. ....	37
Tabla 12 Seleccionar Capa. ....	38
Tabla 13 Visualizar Coordenadas. ....	38
Tabla 14 Mover Mapa. ....	38
Tabla 15 Enfocar mapa. ....	39
Tabla 16 Escala gráfica. ....	40
Tabla 17 Localizar. ....	40
Tabla 18 Seleccionar capa. ....	41

# Índice

---

Tabla 19 Mover mapa. ....	41
Tabla 20 Visualizar coordenadas. ....	42
Tabla 22 Factor de Peso de los Actores sin Ajustar. ....	45
Tabla 23 Factor de peso de los Casos de Uso sin Ajustar. ....	45
Tabla 24 Factor de Complejidad Técnica. ....	47
Tabla 25 Factor de Ambiente. ....	49
Tabla 26 Esfuerzo del Proyecto. ....	50
Tabla 27 Matriz de trazabilidad. ....	57
Tabla 28 Lista de chequeo de requerimientos. ....	58

## Introducción

Históricamente, uno de los primeros pasos en la elaboración de mapas, es la necesidad que ha tenido la humanidad de localización para recorrer una determinada región y tomar óptimas decisiones respecto a rutas de emigración o zonas de caza; después respecto a rutas comerciales, yacimientos minerales, entre otras; y ahora para realizar actividades de distribución, planificación, logística y mercadeo, o simplemente buscando temas de interés que se encuentran en una determinada región: monumentos históricos, museos, paisajes, y otros. En resumen, la elaboración de mapas está encaminada al manejo de información alfanumérica: las características de un lugar de interés junto con su posición espacial (¿dónde están?) y sus relaciones espaciales (¿qué cerca está, qué carreteras conectan, entre otras.?). **(1)**

Las actividades anteriores, se han venido realizando mediante la utilización de medios convencionales: mapas, guías, listados, inventarios, es decir, la combinación de dichas informaciones para extraer conclusiones se realiza en la mente de la persona que está realizando el trabajo. Sin embargo, en este terreno, como en tantos otros, la utilización de herramientas informáticas ha proporcionado nuevos medios para abordar el problema, específicamente la utilización de Sistemas de Información Geográfica (SIG). **(1)**

Aunque los Sistemas de Información Geográfica se han popularizado recientemente, la idea de utilizar herramientas informáticas para manejar entidades con componentes espaciales es bastante antigua. De hecho, está aceptado que el inventario forestal de Canadá, realizado en la segunda mitad de los años 60 bajo la dirección de Roger Tomlinson, fue la primera aplicación real, en la que se pusieron las bases de muchos de los conceptos que todavía se siguen aplicando.

Durante los años 70 se comenzó la aplicación de los SIG a problemas específicos, como el planeamiento o el impacto ambiental, pero fue a principios de los años 80, con la aparición de ordenadores con terminales gráficos, cuando cristalizó una industria. El paso de la década de los 80 a los 90, con la popularización de las estaciones de trabajo UNIX, proporcionó, por primera vez, una plataforma adecuada para desplegar toda la potencia de los SIG, aunque con barreras de precio que los han restringido a aplicaciones técnicas profesionales. **(1)**

# Introducción

---

En la actualidad existe una consolidación de los SIG como industria, caracterizada por una progresiva integración de sistemas ráster y vectoriales, además por el aumento de la importancia de las comunicaciones entre sistemas y la interfaz de usuario, así como por el uso de herramientas de programación tipo visual, basadas en la metodología de la programación de orientación a objetos (OO)<sup>1</sup>. Los nuevos campos de innovación de los SIG son la integración en sistemas de soporte de decisiones para divulgación de la cartografía y de la Información Geográfica. **(2)**

Las expectativas creadas sobre los SIG están también presentes en Cuba con sus correspondientes limitaciones y paradojas. Donde los usuarios de los SIG en Cuba no son los cibernéticos o especialistas informáticos, la mayoría son geólogos, cartógrafos, geógrafos, desarrolladores, arquitectos, ingenieros, quienes conocen y operan Sistemas de Información Geográfica en sus investigaciones y proyectos. Muchos de estos especialistas han aprendido a manejar los SIG de forma autodidacta, sin un adiestramiento previo y trabajan en organizaciones oficiales del Gobierno, tales como GEOCUBA, Ministerio de Ciencia, Tecnología y Medio Ambiente, Instituto de Planificación Física, Ministerio de las Fuerzas Armadas y las Universidades. **(3)**

En aras de ampliar su uso por otros usuarios y diversificar dichos sistemas, en la Universidad de las Ciencias Informáticas (UCI), se han creado varias soluciones dentro del Centro de Desarrollo y Producción de Software Geoinformática y Sistemas Digitales GEySED. Específicamente en el Departamento Geoinformática se creó el proyecto Aplicativo SIG para desarrollar personalizaciones a la plataforma GENESIG, la cual se basa en la tecnología MapServer (Servidor de Mapas). Esta tecnología, es un entorno de desarrollo en código abierto para la creación de aplicaciones SIG con el fin de visualizar, consultar y analizar información geográfica a través de la red. Actualmente MapServer realiza con eficacia la visualización de la información, pero es notablemente lento, debido al elevado grupo de datos a representar, además el proceso de generar los mapas consume tiempo que no permite al servicio responder adecuadamente a las peticiones realizadas por los usuarios.

---

<sup>1</sup>El tema de la programación OO dentro del campo de los SIG es recurrente en la última década. La consideración del objeto geográfico no es necesariamente equivalente a la aplicación de métodos de programación definidos como OO. En cualquier caso, no es evidente ni fácil de distinguir si se trata de una moda puramente informática o realmente de una nueva forma de aproximación a la realidad geográfica. Es necesariamente equivalente a la aplicación de métodos de programación definidos como OO. En cualquier caso, no es evidente ni fácil de distinguir si se trata de una moda puramente informática o realmente de una nueva forma de aproximación a la realidad geográfica.

# Introducción

---

Con el objetivo de agilizar el proceso de visualización de los mapas, han surgido varias tecnologías, entre ellas TileCache, la cual proporciona una base WMS-C/TMS<sup>2</sup> sobre un servidor compatible para Python.

Teniendo en cuenta las ventajas de esta tecnología y la necesidad de agilizar el proceso de visualización de mapas en los SIG desarrollados en el proyecto Aplicativo SIG, se plantea el siguiente **problema a resolver**: ¿Cómo obtener la documentación técnica que contribuya al diseño e implementación de un Sistema de Información Geográfica que sirva de base para el desarrollo de otros SIG basado en la tecnología TileCache?

Se define como **objeto de estudio** de la investigación: el proceso de desarrollo de Sistemas de Información Geográfica en la Universidad de las Ciencias Informáticas; enmarcado en el **campo de acción**: el proceso de visualización de los mapas utilizando TileCache.

Para dar solución al problema planteado se propone como **objetivo general**: desarrollar el análisis de un Sistema de Información Geográfica genérico basado en la tecnología TileCache.

Se defiende la siguiente **idea**: El uso de la tecnología TileCache permitirá agilizar el proceso de visualización de los mapas en los Sistemas de Información Geográfica.

Para ello se especificaron las siguientes **tareas de investigación**:

- 1- Realizar el diseño teórico y metodológico de la investigación.
- 2- Describir los conceptos asociados al dominio del problema.
- 3- Seleccionar y argumentar las tendencias y tecnologías actuales a utilizar en el proceso.
- 4- Realizar la documentación técnica correspondiente al modelo de dominio.
- 5- Capturar los requisitos funcionales y no funcionales.
- 6- Confeccionar el modelo de casos de uso del sistema.
- 7- Elaborar la documentación técnica correspondiente al Modelo de Análisis.
- 8- Validar el análisis de la aplicación.

Los métodos científicos que se utilizarán en el desarrollo de la investigación son los siguientes:

---

<sup>2</sup> **TMS** por sus siglas en inglés es Transportation Management Systems.

# Introducción

---

## Métodos Teóricos

- **Análisis y síntesis:** Se usa para el estudio de la bibliografía utilizada y para realizar una recopilación de la misma.
- **Modelación:** Se utiliza para la modelación de diagramas, representar el proceso de desarrollo y propiciar mejor entendimiento de la solución a implementar.

Al concluir la investigación se esperan los siguientes **resultados**:

- Un Sistema de Información Geográfica genérico que funcione con la tecnología TileCache.
- Documentación de la arquitectura base.

El siguiente trabajo de diploma estará dividido en 3 capítulos:

**Capítulo 1: Fundamentación Teórica.** Se describen brevemente los conceptos fundamentales relacionados con el dominio del problema. Se realizará además un estudio acerca de las soluciones existentes, nacional e internacional y finalmente se presenta el análisis realizado para seleccionar la metodología de desarrollo de software, el lenguaje de modelado y la herramienta case que se utilizará para el desarrollo de la investigación.

**Capítulo 2: Análisis del sistema propuesto.** Se presenta la solución propuesta a partir de la descripción del modelo de dominio y la identificación de requisitos funcionales y no funcionales del sistema a construir, agrupándolos en casos de uso para conformar el Modelo de Casos de Uso del Sistema. Además se identifican las clases del análisis dando lugar a la realización del modelo de análisis.

**Capítulo 3: Estudio de factibilidad y validación del sistema propuesto.** Se hace un estudio de la factibilidad y validación del sistema propuesto.

## 1. Capítulo I: Fundamentación Teórica

### Introducción.

En el presente capítulo se abordan los temas relacionados con la fundamentación teórica de la investigación. Se describen los conceptos asociados al dominio del problema, el objeto de estudio con el fin de lograr un mayor conocimiento del problema a resolver. Se analizan algunas soluciones existentes nacionales e internacionales. Además se hace un detallado análisis de las tecnologías y herramientas a utilizar.

### 1.1. Conceptos asociados al dominio del problema.

La **información geográfica** es un conjunto de datos espaciales, los cuales brindan una información de algún hecho o fenómeno. **(4)**

Un **sistema de información** es un conjunto organizado de elementos que interactúan entre sí para procesar información. **(5)**

#### 1.1.1. Mapa.

El **mapa** es una representación gráfica y métrica de una porción de territorio sobre una superficie bidimensional, que por lo general suele ser plana. **(6)**

- Formato **ráster**, es una forma de representar la información digital por medio de cuadros llamados píxeles<sup>3</sup>, que es la unidad mínima de medida. **(7)**
- Formato **vectorial**, es una forma de representar la información digital por medio de puntos, líneas y polígonos o áreas. **(8)**

#### 1.1.2. WMS.

El servicio **Web Map Service (WMS)** definido por el OGC (Open Geospatial Consortium) produce mapas de datos referenciados espacialmente de forma dinámica a partir de información geográfica. Este estándar internacional define un "mapa" como una representación de la información geográfica en forma de un archivo de imagen digital, donde un mapa no consiste en los propios datos. Los mapas producidos por WMS se generan normalmente en un formato de imagen como PNG, GIF o

---

<sup>3</sup>**Pixel** es la menor unidad homogénea en color que forma una imagen digital.

JPEG y opcionalmente como gráficos vectoriales en formato SVG (Scalable Vector Graphics) o WebCGM (Web Computer Graphics Metafile). (9)

### 1.1.3. MapServer

**MapServer** es un servidor de mapas OpenSource (de código abierto), desarrollado actualmente por mapTools.org. Es un entorno sencillo que permite el desarrollo de aplicaciones SIG para Internet (o cualquier otra red donde pueda instalarse este servidor). Permite visualizar, consultar y analizar información geográfica a través de la red mediante la tecnología Web. Este puede ser utilizado como una aplicación CGI<sup>4</sup> o a través del acceso a la API<sup>5</sup> de MapServer que proveen las librerías Mapscript. (10)

### 1.1.4. Mapscript

**Mapscript** constituye la vía de comunicación de las aplicaciones SIG con el servidor de mapas MapServer y por otra parte rompe en cierta medida esta rigidez de la representación de mapas a través de los ficheros “.map”, dichos ficheros definen los recursos que serán utilizados usados en la aplicación CGI, también contiene información acerca de cómo se debe dibujar el mapa, la leyenda y el resultado de realizar una consulta. Mapscript además permite modificar los mismos en tiempo de ejecución, facilitando de esta forma la creación de aplicaciones con un grado de personalización mayor, eventualmente no alcanzado con aplicaciones del MapServer en modo CGI. (11)

### 1.1.5. TileCache.

**TileCache**<sup>6</sup> es una implementación de un WMS-C<sup>7</sup> disponible bajo la licencia BSD por MetaCarta<sup>8</sup>. TileCache proporciona una base WMS-C/TMS<sup>9</sup> sobre un servidor compatible para Python, con los mecanismos de almacenamiento en caché acoplables y de representación. En el caso de uso más simple, TileCache sólo requiere acceso de escritura al disco, la capacidad de ejecutar scripts CGI de Python y un WMS que se desea almacenar en caché<sup>10</sup>. Con estos recursos, se puede crear una cache

---

<sup>4</sup> **CGI** por sus siglas en inglés 'Common Gateway Interface' es de las primeras formas de programación web dinámica.

<sup>5</sup> **API** por sus siglas en inglés es 'Application Programming Interface'.

<sup>6</sup> **TileCache** se traduce como el almacenamiento en memoria de tiles (mosaicos o pequeñas imágenes).

<sup>7</sup> **WMS-C** por sus siglas en inglés es Web MapService – Cache de la especificación WMS Tile Caching.

<sup>8</sup> **MetaCarta** es un servicio que posibilitará a sus usuarios conocer el sitio exacto del planeta en el que una noticia ha tenido lugar.

<sup>9</sup> **TMS** por sus siglas en inglés es Transportation Management Systems.

<sup>10</sup> **Caché** se usa en informática, como adjetivo invariable, para referirse a la memoria de rápido acceso, situada entre el procesador y la memoria principal.

local de cualquier servidor WMS y utilizar el resultado en cualquier WMS-C del lado del cliente, como Open Layers. **(12)**

### 1.2. Objeto de estudio: Los Sistema de Información Geográfica.

Un Sistema de Información Geográfica se puede definir como aquel método o técnica de tratamiento de la información geográfica que permite combinar eficazmente información básica para obtener información derivada. Para ello, se cuenta tanto con las fuentes de información como con un conjunto de herramientas informáticas (*hardware* y *software*) que facilitan esta tarea; todo ello enmarcado dentro de un proyecto que habrá sido definido por un conjunto de personas y controlado, así mismo, por los técnicos responsables de su implantación y desarrollo. En definitiva, un SIG es una herramienta capaz de combinar información gráfica (mapas...) y alfanumérica (estadísticas...) para obtener una información derivada sobre el espacio. **(13)**

#### 1.2.1. Desarrollo de los SIG

La rápida evolución de la tecnología ha obligado a intentar estar al día de los acontecimientos para poder ser más productivos y hacer más fácil tareas que hace unos años eran meras utopías. Aunque los SIG se empezaron a generalizar a partir de la década de los 80, su gestación y desarrollo se remonta dos décadas atrás. Entre los años 1960 y 64 se desarrolló el Canadian Geographic Information System (C.G.I.S.), con el objeto de gestionar los bosques y superficies marginales de Canadá. Este sistema ha ido evolucionando y sigue en uso en la actualidad. **(2)**

Desde entonces hasta la actualidad, los SIG alcanzan un gran desarrollo y expansión gracias fundamentalmente a la evolución y ampliación de las capacidades de los ordenadores, el desarrollo de los lenguajes de programación y el avance del tratamiento gráfico. Los SIG permiten relacionar información de cualquier tipo (base de datos) con una localización geográfica (mapa). Con un SIG, instituciones gubernamentales o empresas de mercadeo pueden relacionar información demográfica de censos con mapas políticos; médicos y hospitales pueden relacionar mapas de enfermedades con condiciones de salubridad; autoridades y legisladores pueden relacionar mapas de lugares donde se cometieron delitos; personal de servicios de emergencia puede relacionar mapas de áreas de riesgo con información sobre inundaciones o incendios forestales. Otra característica poderosa de los SIG es la facilidad para relacionar simultáneamente, por medio de capas, diferentes tipos de información con

una localización geográfica. Esto permite comparar y analizar información con el fin de revelar causas y efectos muy difíciles de analizar con métodos cuantitativos tradicionales. A continuación se visualizan algunas de las funcionalidades de los componentes de un SIG. (Ver Figura 1) (14)



Figura 1 Funcionalidades de los SIG.

### 1.2.2. Desarrollo de los SIG en la UCI

La Universidad de las Ciencias Informáticas a lo largo de los años ha adquirido gran experiencia en el desarrollo de grandes proyectos, contribuyendo de una forma u otra en la formación de profesionales con grandes conocimientos en áreas de la informática. Con el desarrollo de la Plataforma GeneSIG, se persigue fortalecer la experiencia en la realización de Sistemas de Información Geográfica y abrir un espacio sólido en el mercado de aplicaciones de esta rama, reutilizando los componentes y funcionalidades para personalizar los productos en cualquier negocio que lo requiera. (15)

Dicha plataforma aspira a poseer los siguientes beneficios funcionales: (15)

- Representación geoespacial de la información asociada a negocios específicos.

# Capítulo 1

## Fundamentación Teórica

---

- Servicios de acceso a la información geográfica, para su consulta, análisis y visualización, mediante una interfaz de usuario sencilla y de fácil manejo que pueda ser utilizada por usuarios no especializados en tecnologías SIG.
- Actualización de las Bases de Datos a través de un mecanismo de réplica consistente.
- Acceso a Bases de Datos alfanuméricas externas para su representación geográfica y posterior análisis.
- Aislamiento de bases de datos geográficas y aplicaciones SIG mediante la adaptación de datos y aplicaciones a estándares (OpenGIS).
- Integración con la información ráster existente (imágenes de satélite, orto-fotos o mapas escaneados) con información vectorial.
- Publicación en Internet de servicios WMS (Web Map Services) de OGC (Open GIS Consortium) e integración de los mismos a geo-portales.

En general se puede lograr:

- Aumento de la capacidad de respuesta a la sociedad cubana por parte de estas entidades y a naciones interesadas en esta tecnología.
- Mayor satisfacción en los usuarios.
- Respuesta rápida para la toma de decisiones de otras áreas económicas, políticas y/o sociales.

De manera que el objetivo fundamental es lograr la representación geoespacial de la información asociada a negocios específicos, además debe permitirse realizar análisis sobre dicha información, que se realiza actualmente por medio de MapServer.

Para optimizar dicho objetivo se ha creado el Proyecto Aplicativos perteneciente al centro de Desarrollo y Producción de Software Geoinformática y Sistemas Digitales GEySED, del Departamento Geoinformática en la Facultad 6, para lograr personalizaciones de la Plataforma GeneSIG.

### 1.3. Soluciones existentes.

Hoy día se pueden encontrar muchas soluciones de proyectos que usen TileCache o procedimiento similares. A continuación se describen algunas de las recomendaciones y especificaciones actuales para el servicio de tiles<sup>11</sup> y algunas de las implementaciones más usadas por la comunidad de SIG.

---

<sup>11</sup> Tiles mosaicos.

### Especificaciones

Se han desarrollado las especificaciones WMS Tile Caching (conocida como WMS-C), Tile Map Service de OSGeo y Web MapTiling Service del Open Geospatial Consortium, de las cuales la primera es la más extendida.

#### 1.3.1. WMS Tile Caching

El objetivo de la propuesta WMS Tile Caching, es encontrar una manera de optimizar la entrega de las imágenes de mapa a través de Internet. La propuesta tiene que ofrecer algún medio ideal por el cual los clientes pueden buscar mapas de tiles de los servidores existentes, de tal manera que las imágenes puedan ser almacenadas en la caché del servidor, o en un lugar intermedio, o incluso ser completamente pre-generadas. Además, la propuesta debe ofrecer a modo de publicidad que un conjunto de tiles en caché está disponible para una determinada capa de un particular, WMS o proxy WMS. La propuesta debe aprovechar las inversiones existentes en servidores WMS OGC (Open Geospatial Consortium) compatibles en la red. Permite a los servidores optimizar la generación de imágenes, además que se almacenen los tiles en puntos intermedios. **(16)**

Esta recomendación define una extensión del servicio WMS como mejora para aumentar la escalabilidad. La misma se basa en dos conceptos básicos:

- a) Limitar las capas con tiles a configuraciones fijas (escalas, proyecciones, tamaño de tiles)
- b) Restringir los parámetros del WMS getMap a los mínimos requeridos.

#### 1.3.2. Tile Map Service

OSGeo (Open Source Geospatial Foundation) ofrece una especificación, TMS (Tile Map Service Specification – no es un estándar oficial pero sí un proyecto oficial de OSGeo), donde se describe una posible solución para la comunicación cliente-servidor utilizando un servicio de tiles. Esta especificación es el resultado o resumen de distintas personas de las comunidades SIG interesadas en el modelo de tiles **(18)**.

Se define cómo el servidor debe proporcionar la información relacionada con los mapas que contienen: las capas disponibles, conjunto de escalas permitidas, conjunto de tiles para cada escala, etc. y la forma de acceder a dichos tiles. Esta especificación tiene la ventaja de poder ofrecer capas con distintas configuraciones de SRS, tamaño de tile, escalas, etc. pero en su contra, requiere tener todas

los tiles pre-generados porque el acceso vía HTTP es directo a fichero siguiendo la interfaz REST<sup>12</sup> y la nomenclatura definida en la especificación.

En la figura 2 se muestra la malla definida por TMS. **(19)**

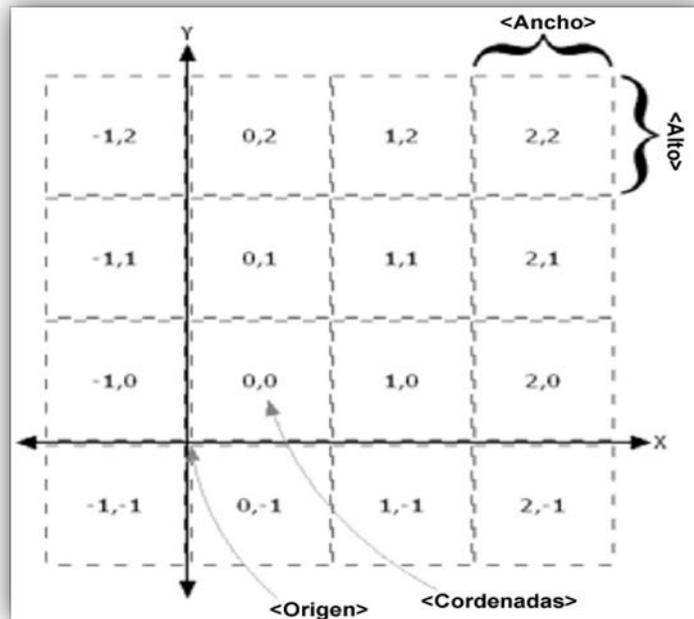


Figura 2 Malla de tiles en TMS.

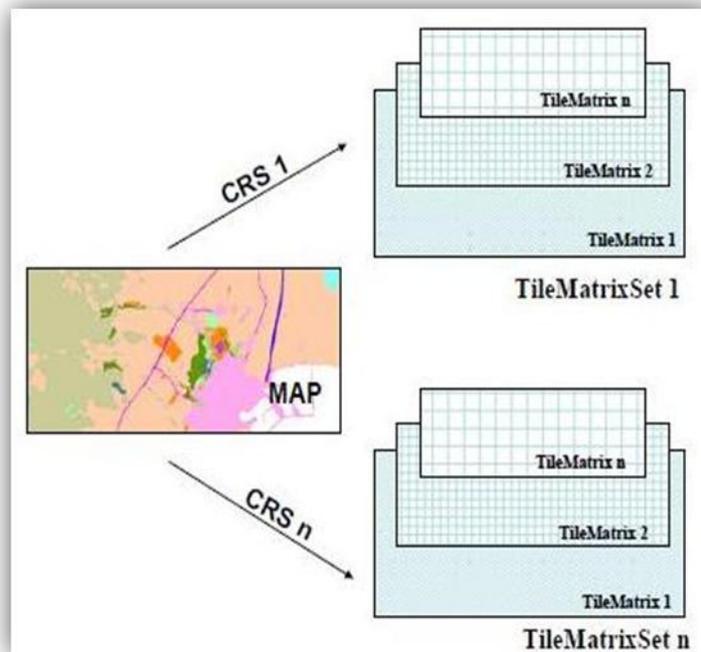
### 1.3.3. Web MapTiling Service

El WMTS esta basado en un modelo piramidal de tiles (tiling model), en el que los mapas son pre-renderizados y fragmentados a un conjunto predeterminado de escalas. Incorpora tres peticiones: GetCapabilities, GetTile y GetFeatureInfo. Este servicio acelera la respuesta del servidor, debido a que los datos están pre-renderizados y mejora el rendimiento de los clientes, porque estos conocen la estructuración de los datos y todas las peticiones son almacenadas en memoria. **(20)**

Un WMTS es un servicio escalable y cacheable (se puede almacenar en memoria), que usa un modelo de tiles parametrizado de tal manera que un cliente puede hacer peticiones de un conjunto discreto de

<sup>12</sup> **REST** Transferencia de Estado Representacional, o Representational State Transfer es una técnica para el acceso a recursos a través de una URI.

valores y recibir rápidamente del servidor fragmentos de imágenes prerenderizadas (tiles) que no requieren de ninguna manipulación posterior. Cada una de las capas de un servidor WMTS sigue estructuras piramidales de escalas, en la que cada escala o nivel de la pirámide es una rasterización y fragmentación de los datos geográficos a una escala o tamaño de píxel concreto. Una capa puede estar disponible en varios sistemas de coordenadas y tener diferente ámbito en función de éstos. Una TileMatrixSet es un conjunto de rasterizaciones y fragmentaciones de la capa a diferentes escalas predeterminadas para un sistema de referencia y ámbito concreto. Una capa tendrá tantos TileMatrixSet como sistemas de referencia y ámbitos para los cuales estén disponibles. Cada una de las rasterizaciones y fragmentaciones de la capa a una escala determinada, sistema de coordenadas y ámbito que conforman la TileMatrixSet, se conoce como TileMatrix (Ver Figura 3).



**Figura 3** Modelo piramidal de tiles de una capa de un servidor WMTS.

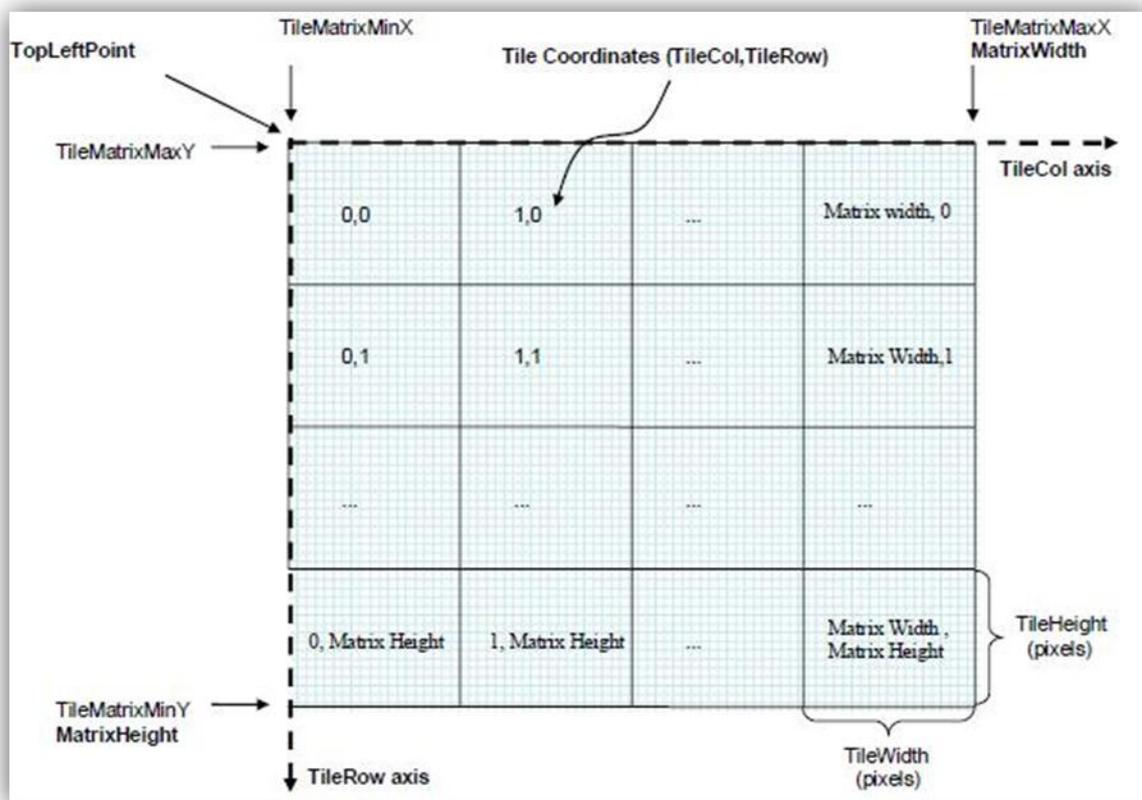
Cada TileMatrix define el patrón de corte de la capa en tiles para un sistema de referencia concreto, un ámbito y una escala; indicando el origen de la matriz en el sistema de coordenadas, el tamaño de la matriz en número de fragmentos (MatrixWidth y MatrixHeight) y el tamaño de cada uno de sus fragmentos o tiles en número de píxeles (TileWidth y TileHeight) (Ver figura 4).

# Capítulo 1

## Fundamentación Teórica

La interfaz WMTS especifica dos operaciones básicas que pueden ser solicitadas por un cliente: la operación de GetCapabilities y la de GetTile (Ver figura 5). La operación de GetCapabilities, obligatoria en todos los servicios web OGC, permite al cliente solicitar y recibir del servidor los metadatos del servicio; que en este caso son una descripción detallada de las capas disponible en el servidor y de los patrones de matrices de fragmentación que siguen.

Una vez conocidas las capacidades de un servidor, la operación de GetTile permite al cliente solicitar y recibir una tesela de una de sus capas, a una escala, estilo de visualización y formato determinado.



**Figura 4** Espacio de una matriz de tiles (TileMatrix).

La especificación del WMTS, también define otra operación, en este caso opcional, el GetFeatureInfo, que permite obtener la información sobre el contenido de un píxel particular del mapa. Las peticiones del servicio WMTS pueden realizarse en codificación GET y POST con KVP<sup>13</sup> o en codificación RESTful. La codificación KVP, fácil de especificar y muy usada, permite realizar cualquiera de las

<sup>13</sup> KVP por sus siglas en inglés es Key Value Par (clave de valor nominal)

peticiones del servicio de forma eficaz, pero tiene el inconveniente de que si se altera el orden de los parámetros, la petición ya no se almacena en memoria. La codificación RESTful, permite almacenar en memoria e identificar de forma unívoca cualquiera de los recursos del servicio WMTS, el documento de los metadatos del servicio, cada tile y cada píxel de un tile son identificados como recursos, correspondiendo una URL a cada uno.

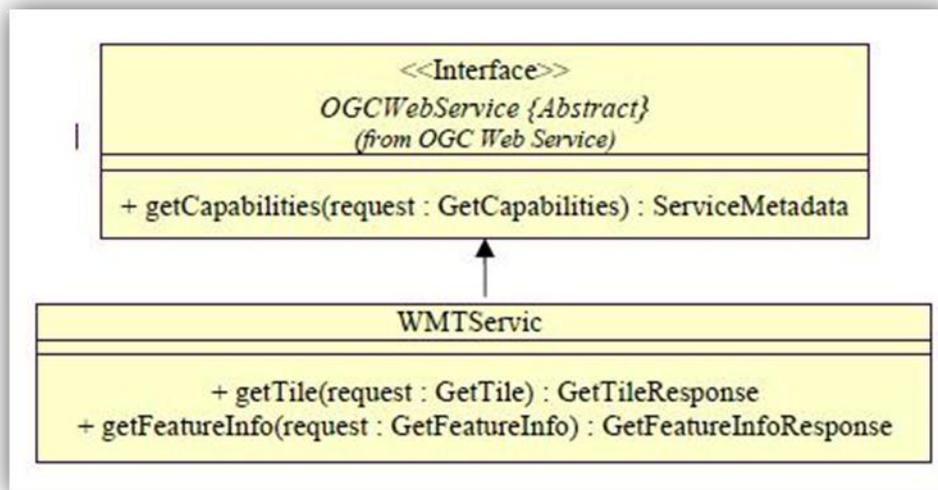


Figura 5 Diagrama UML de la interface WMTS.

### Implementaciones de WMS Tile Caching

De entre las implementaciones de WMS-C, del lado del servidor se destacan TileCache y GeoWebcache que a continuación profundizaremos, para luego establecer una breve comparación.

#### 1.3.4. GeoWebCache

GeoWebCache es un caché para servicios WMS implementados en Java. Optimiza el renderizado de Web Map Services (WMS), guardando los mapas renderizados cuando son solicitados, actuando como un proxy entre el cliente (Open Layers o Google Maps) y el servidor (GeoServer u otro servidor WMS). En la medida en que nuevos mapas son solicitados, GeoWebCache intercepta estas llamadas y envía los tiles pre-renderizados, si están almacenados, o llama al servidor para que los renderice si es

necesario. De esta manera, una vez que un mapa está guardado, la velocidad de renderizado se incrementa varias veces, mejorando la experiencia del usuario. **(21)**

Tiene la ventaja de que al ser una aplicación J2EE(Java 2 Platform Enterprise Edition) es fácilmente integrable en entornos empresariales donde se utilice el stack de aplicaciones Java. Actualmente GeoWebCache se distribuye tanto como aplicación web independiente como integrada con GeoServer. **(22)**

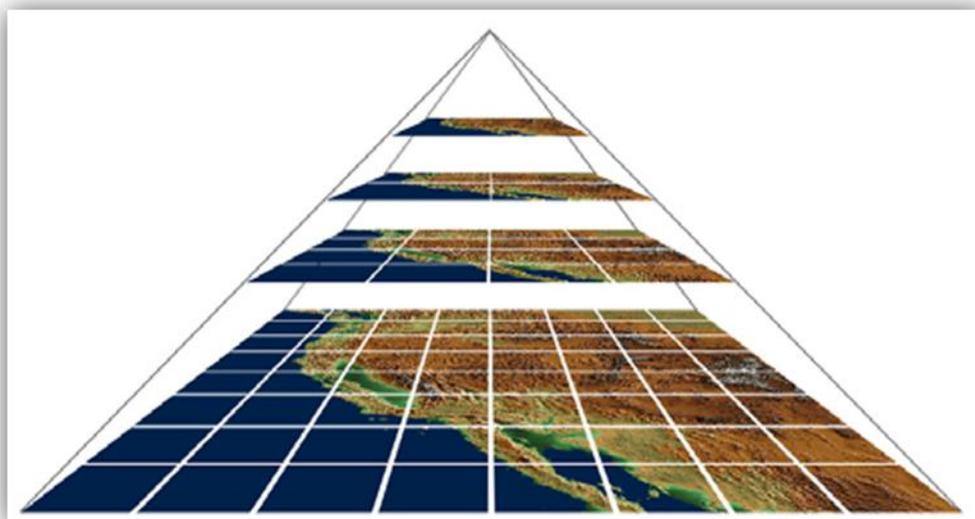
### Características **(23)**

- OGC compatibles Web Map Tiling Service (WMTS), Web Map Service-Caching (WMS-C), Tiled Map Service (TMS).
- Capacidad de servir como un WMS compatible mediante la recombinación de remuestreo y tiles para responder a peticiones arbitrarias de imagen.
- Salida nativa de Google Maps (incluyendo Google para móviles), Google Earth super superposiciones (vectorial y ráster), Bing Maps y Yahoo Maps.
- Buen control sobre la expiración de los tiles por API REST o notificación GeoRSS.
- Las cuotas de disco con menos uso frecuente y menos usadas recientemente, algoritmos para gestionar con eficacia el espacio en disco.
- Maneja peticiones WMS con el tiempo, la elevación, los estilos alternativos y filtros.

### 1.3.5. TileCache

La salida de información de un SIG puede ser de tipo textual o de tipo gráfico. Ambos tipos de información pueden ser presentadas en forma digital o analógica. El medio analógico es el que se presenta al usuario como respuesta a una interrogante de él mismo. La información textual analógica consiste normalmente en un conjunto de tablas que representan la información almacenada en la base de datos o representan el resultado de algún tipo de análisis efectuado sobre esta. La información analógica gráfica consiste en mapas, gráficos o diagramas. El sistema debe proveer la capacidad de complementar la información gráfica, antes de su presentación definitiva, por medio de una simbología adecuada y manejar la posibilidad de adicionar elementos geométricos que permitan una calidad y una visualización fáciles de entender por el usuario.

Para una mejor representación de la información geográfica surge la idea de tener las imágenes pre-generadas para eliminar el costo de renderizar (es el proceso de generar una imagen desde un modelo.) basándose en un modelo de tiles. Este modelo consiste en dividir el mapa cartográfico, ofrecido por el servidor, en una pirámide de mallas donde cada nivel de la pirámide corresponde con una resolución o escala del mapa y todas las celdas de la pirámide tienen un tamaño fijo en pixel, como muestra la figura 6.



**Figura 6** Pirámide de tiles.

### **Funcionamiento**

Este nuevo servicio, con el modelo de tiles, requiere limitar las posibilidades de configuración en la petición HTTP frente a las que ofrece el servicio WMS y definir una especificación donde describir la comunicación cliente-servidor.

Para tal fin se han desarrollado especificaciones como WMS Tile Caching (conocida como WMS-C) y de sus implementaciones se destaca el proyecto TileCache creado en MetaCartaLabs, para la gestión de una solicitud de un cliente, pedir a un servicio Web Map Service para la respuesta y la memoria caché, así que para las solicitudes junto a los mismos datos que el propio proxy podría utilizar su copia en caché y evitar las peticiones al servidor web de mapas. **(24)**

El proyecto TileCache está desarrollado en Python y funciona como proxy entre el cliente y el servidor WMS. Este proxy se encarga de atender las peticiones WMS y almacenar las tiles en caché (el

almacenamiento en caché sigue una estructura de carpetas denominadas tile cache). La ventaja de este proyecto es la generación de tiles de forma dinámica. Si se realiza una petición a un tile no almacenado, le pasa la petición al servidor WMS y guarda (cachea) la respuesta de este para una posterior petición. (19)

En la siguiente figura se muestra los distintos componentes que intervienen en el proyecto.

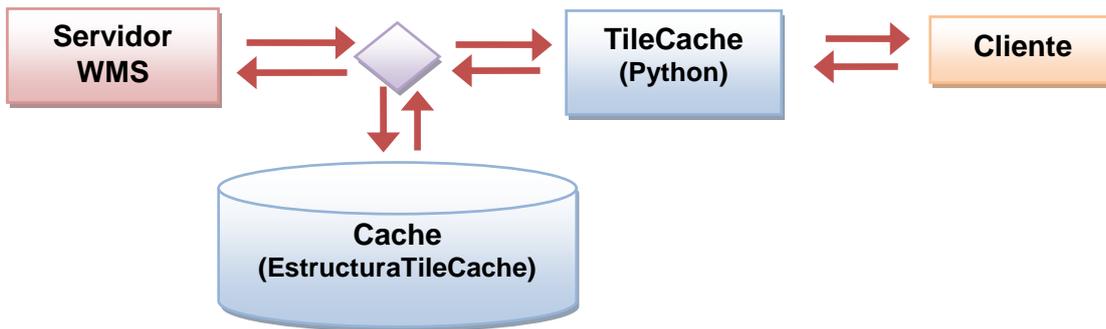


Figura 7 Diagrama de TileCache.

#### 1.4. Comparación de soluciones existentes.

Existen múltiples estudios comparativos acerca del rendimiento de servidores de mapas. Sin embargo no se han constatado estudios de rendimiento comparativo entre sistemas de caché de mapas. A continuación se presentan algunas características comparativas entre las implementaciones existentes.

##### GeoWebCache

- Se implementa utilizando Java en lugar de Python. Se distribuye bajo la GNU General Public License.
- Como una aplicación Java, GeoWebCache requiere un tiempo de ejecución de Java (v1.5 o v1.6) y Java Servlet Server (por ejemplo, Apache Tomcat) para ser instalado. GeoWebCache afirma que es una caché rápida capaz de saturar una conexión de 100 Mbps.
- A diferencia de TileCache, GeoWebCache normalmente sólo se ejecuta en una configuración independiente, lectura de archivos WMS. Servido por un servidor WMS compatible. Sin embargo, también se suministra con GeoServer en una forma más fácil de usar y tiene un menor consumo de memoria.

### TileCache

- Tiene dos modalidades de funcionamiento. El primero hace el tile usando MapServer o Mapnick como una copia de la representación final. El segundo puede almacenar en caché tiles WMS descargados desde un servicio WMS remoto.
- Soporta dos mecanismos de caché diferentes. El mecanismo de almacenamiento DiskCache almacena en memoria tiles como archivos de imagen. Además, el mecanismo MemoryCache que almacena datos en una instancia memcache<sup>14</sup> o clúster.
- Está distribuido bajo licencia BSD. Se requiere Python para ser instalado en el servidor host.
- Fácil configuración.

Por lo antes presentado se puede apreciar que el TileCache ofrece el mejor rendimiento, comodidad y facilidad de uso en las implementaciones probadas.

### 1.5. Metodologías de Desarrollo de Software.

Un proceso de software detallado y completo suele denominarse “Metodología”. Las metodologías se basan en una combinación de los modelos de proceso genéricos (cascada, evolutivo, incremental, entre otros). Adicionalmente una metodología debería definir con precisión los artefactos, roles y actividades involucrados, junto con prácticas y técnicas recomendadas, guías de adaptación de la metodología al proyecto, guías para uso de herramientas de apoyo, entre otros. **(25)**

La comparación o clasificación de las mismas no es una tarea sencilla debido a la diversidad de propuestas y diferencias en el grado de detalles, alcance de cada una y la información disponible. Se pueden clasificar en dos grupos: Metodologías Estructuradas y Metodologías Orientadas a Objetos si se toman como criterio las notaciones utilizadas para especificar artefactos producidos en actividades de análisis y diseño. Según su filosofía se clasifican en Ágiles o Robustas. A continuación se presentará un breve resumen de las Robustas.

#### Metodologías Robustas.

Las metodologías no ágiles son aquellas que están guiadas por una fuerte planificación durante todo el proceso de desarrollo; llamadas también metodologías tradicionales o pesadas, donde se realiza una intensa etapa de análisis y diseño antes de la construcción del sistema. **(25)**

---

<sup>14</sup>**Memcached** es una memoria de clave y valor para almacenar en pequeños trozos de datos arbitrarios (cadenas, objetos).

Entre las principales metodologías robustas se encuentran RUP (Rational Unified Process). Su principal objetivo está centrado en la creación detallada y exhaustiva de la documentación y en el cumplimiento de un plan de proyecto, donde todo esto se define en la fase inicial del desarrollo del mismo.

El Proceso Unificado de Desarrollo de Software conocido como RUP es una metodología de software que permite el desarrollo de aplicaciones a gran escala, mediante un proceso continuo de pruebas y retroalimentación, garantizando el cumplimiento de ciertos estándares de calidad, con el inconveniente de generar mayor complejidad en los controles de administración del mismo.

### **Características de la tecnología**

RUP presenta entre sus características ser un proceso de desarrollo orientado a objetos y utiliza el Lenguaje Unificado de Modelado (UML) como lenguaje de representación visual. Este proceso unificado define “Quién”, “Cómo”, “Cuándo” y “Qué” debe hacerse en el proyecto. Tiene tres características fundamentales: es iterativo e incremental, centrado en la arquitectura y dirigido por casos de usos **(25)**

El período de vida del software está particionado en ciclos, cada ciclo consta de cuatro fases: concepción o inicio, elaboración, construcción y transición y cada vez que termina un ciclo se produce una versión del sistema. Es ideal para proyectos cuyos requisitos no son variables y para grandes equipos de desarrollo. Sin embargo puede adaptarse a diferentes condiciones. En RUP se han agrupado las actividades en grupos lógicos definiéndose 9 flujos de trabajo, los 6 primeros son flujos de ingeniería y los tres últimos de apoyo.

Se decide escoger esta metodología luego de haberla analizado profundamente. Se escoge porque consiste en una metodología que implementa buenas prácticas para el desarrollo de software, permite reducir los riesgos y hace que el producto se torne más predecible, además de posibilitar la realización de pruebas continuas e iterativas que promueven una mejor evaluación del estado del proyecto. La presente investigación desarrolla las fases Inicio y Elaboración, específicamente, los flujos de trabajo Modelo de negocio, Requisitos y parte de flujo de trabajo Análisis y Diseño.

#### **1.5.1. Rol de analista**

Dentro de los flujos de trabajo llevado a cabo en la investigación se desarrollará el rol de analista. El cual consiste en un individuo que es el responsable de llevar a cabo el análisis del sistema, con el

# Capítulo 1

## Fundamentación Teórica

objetivo de comprender el problema a resolver, definir el alcance y brindar una visión más clara de lo que el sistema debe hacer para cumplir los requerimientos del usuario.

Para desempeñar un buen rol, como analista del sistema se debe poseer un amplio rango de habilidades o niveles de competencias los cuales son: **(26)**

- Debe ser por encima de todo, un experto en la identificación y comprensión de problemas.
- Buen modelador y tener habilidades de comunicación por encima de todo.
- Debe tener dominio del proyecto y de las tecnologías.
- Capacidad de absorber y comprender rápidamente la información.
- Capaz de colaborar de forma efectiva con otros miembros del proyecto.

Entre las actividades que realiza y responsabilidades (Ver Tabla 1) que debe llevar a cabo el analista del sistema se encuentran: **(26)**

Actividades	Responsabilidades
Buscar actores, trabajadores y Casos de uso	Atributos de requisitos
Crear un vocabulario común	Especificaciones suplementarias
Definir el contexto del sistema	Glosario
Desarrollar el plan de gestión de requisitos	Guión gráfico
Desarrollar especificaciones suplementarias	Modelos de Caso de uso
Desarrollar la visión.	Plan de gestión de requisitos
Gestionar las dependencias.	Solicitudes de los interesados
Obtener solicitudes del interesado.	Visión.

**Tabla 1** Actividades y Responsabilidades del Analista del sistema en RUP.

En el desarrollo de la presente investigación, se realizarán las siguientes tareas:

- Realizar el modelo de dominio.
- Identificar roles, trabajadores y casos de uso.

- Capturar, especificar y validar los requisitos funcionales y no funcionales del sistema.
- Elaborar el Modelo de clases del Análisis.
- Realizar el Diagrama del modelo de Objeto.

### 1.6. Herramientas CASE.

Las herramientas de Ingeniería de Software Asistida por Ordenador (CASE, por sus siglas en Inglés, Computer Aided Software Engineering) son diversas aplicaciones informáticas destinadas a aumentar la productividad en el desarrollo de software reduciendo el costo de las mismas en términos de tiempo y de dinero. Estas herramientas pueden ayudar en todos los aspectos del ciclo de vida de desarrollo del software en tareas como el proceso de realizar un diseño del proyecto, cálculo de costos, implementación de parte del código automáticamente con el diseño dado, compilación automática, documentación o detección de errores entre otras. **(27)**

### 1.7. Visual Paradigm.

Visual Paradigm es una potente plataforma, la cual está diseñada para una amplia gama de usuarios, incluidos los analistas de sistemas, ingenieros de software, entre otros. Esta herramienta facilita la interoperabilidad con otras herramientas de modelado de UML, además de permitir la transición de análisis para el diseño. **(28)**

Es la herramienta por excelencia para ser utilizada en un ambiente de software libre. Permite crear tipos diferentes de diagramas en un ambiente totalmente visual. Es muy sencillo de usar, fácil de instalar y actualizar. Soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue.

Visual Paradigm ofrece: **(28)**

- Un entorno de creación de diagramas para UML 2.0.
- Diseño centrado en casos de uso y enfocado al negocio que genera un software de mayor calidad.
- Uso de un lenguaje estándar común a todo el equipo de desarrollo que facilita la comunicación.
- Capacidades de ingeniería directa e inversa.
- Modelo y código que permanece sincronizado en todo el ciclo de desarrollo.

- Disponibilidad de múltiples versiones, para cada necesidad.
- Disponibilidad de integrarse en los principales entornos de desarrollo.
- Disponibilidad en múltiples plataformas.

Las principales características del Visual Paradigm son: **(28)**

- Producto de calidad.
- Soporta aplicaciones web.
- Varios idiomas.
- Generación de código para Java y exportación como HTML.
- Fácil de instalar y actualizar.
- Compatibilidad entre ediciones.

### 1.8. El Lenguaje Unificado de Modelado (UML).

Lenguaje Unificado de Modelado (*Unified Modeling Language*) UML, por sus siglas en inglés, es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad. Es un lenguaje gráfico para visualizar, especificar, construir y documentar los artefactos que se crean durante el proceso de desarrollo de software. **(29)**

Se utiliza para definir un sistema de software, para detallar los artefactos en el sistema y para documentar y construir el mismo. Se puede aplicar en una gran variedad de formas para dar soporte a una metodología de desarrollo de software (como el Proceso Unificado Racional), pero no especifica en sí mismo qué metodología o proceso usar. Intenta solucionar el problema de propiedad de código que se da con los desarrolladores, al implementar un lenguaje de modelado, y se crea una documentación también común, que cualquier desarrollador con conocimientos de UML será capaz de entender, independientemente del lenguaje utilizado para el desarrollo. **(29)**

#### 1.8.1. Objetivos de UML. (29)

- Visualizar: UML permite expresar de una forma gráfica un sistema de forma que otro lo puedan entender.
- Especificar: UML permite especificar cuáles son las características de un sistema antes de su construcción.
- Construir: A partir de los modelos especificados se pueden construir los sistemas diseñados.

- Documentar: los propios elementos gráficos sirven como documentación del sistema desarrollado que pueden servir para su futura revisión. Un modelo UML está compuesto por tres clases de bloques de construcción:

- Elementos: los elementos son abstracciones de cosas reales o ficticias (objetos, acciones, entre otros).
- Relaciones: relacionan los elementos entre sí.
- Diagramas: son colecciones de elementos con sus relaciones.

Dentro de sus características se encuentra: **(29)**

- Es un lenguaje de representación visual.
- Permite combinar diversos elementos gráficos y crear diagramas.

### Conclusiones

En la elaboración de este capítulo se abordaron conceptos generales que permitieron desarrollar el marco conceptual de la investigación. Además se realizó una caracterización de las soluciones existentes, para analizar y comparar detalladamente cada una de ellas, y se definió la metodología y herramientas necesarias para el desarrollo de la solución. Al finalizar el capítulo se arribaron a las siguientes conclusiones:

- De la especificación seleccionada, se analizaron y compararon las implementaciones TileCache y GeoWebCache; seleccionándose la primera por ofrecer un mejor rendimiento, comodidad y facilidad de uso.
- Se demostró la importancia de usar TileCache en los SIG para lograr una mejor visión del análisis de la información geográfica como fuente principal para la gestión y manipulación de datos geográficos.
- Se determina el uso de la Metodología de Desarrollo de Software RUP por ser una metodología robusta que brinda un conjunto procedimientos, técnicas, herramientas y un soporte documental que ayuda a los desarrolladores en la realización de un nuevo software.
- Se seleccionó como herramienta CASE Visual Paradigm para el modelado del sistema a través de la realización de diagramas en el lenguaje UML.

## 2. Capítulo 2: Análisis del sistema propuesto

### Introducción.

Para el desarrollo exitoso de la aplicación es necesario tener conocimiento de los procesos que ocurren en la actualidad acerca de la creación de los Sistemas de Información Geográfica, mediante el modelo de dominio se analizan todas las entidades y conceptos presentes en el entorno donde se aplica el sistema. En el presente capítulo se identifican los principales requerimientos funcionales y no funcionales para posteriormente puntualizar sus principales procesos. Además se genera y representa el análisis que se le efectuará al sistema.

### 2.1. Modelo de Dominio

Debido a que un Sistema de Información Geográfica puede ser personalizado y comercializado por cualquier cliente o empresa interesada, es difícil encontrar procesos de negocio bien definidos que permitan realizar un Modelo del Negocio; por consiguiente se debe realizar un modelo de dominio y un glosario de términos para describir los términos empleados en el modelo.

El Modelo de Dominio o Modelo Conceptual, permite de manera visual mostrar al usuario los principales conceptos que se manejan en el dominio del problema. Representa las clases conceptuales del mundo real, no de componentes de software. Puede considerarse como un diccionario visual de las abstracciones relevantes, vocabulario e información del dominio. Aprovechando las oportunidades 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. **(30)**

#### 2.1.1. Eventos principales del entorno

Los cartógrafos son los encargados de crear o construir los mapas. Los mapas están compuestos por varias escalas representativas, leyendas que permiten un mejor entendimiento de los mismos y un sistema de coordenadas que permite localizar y medir elementos geográficos. También estos mapas proporcionan información, dicha información puede ser económica, social o científica. Los mapas son utilizados, además, por los diferentes clientes pertenecientes a una entidad específica, la misma proporciona toda la información que se le agregará posteriormente a los mapas en dependencia de las necesidades de los clientes. (Ver figura 8)

### 2.1.2. Diagrama de clases del Modelo de Dominio

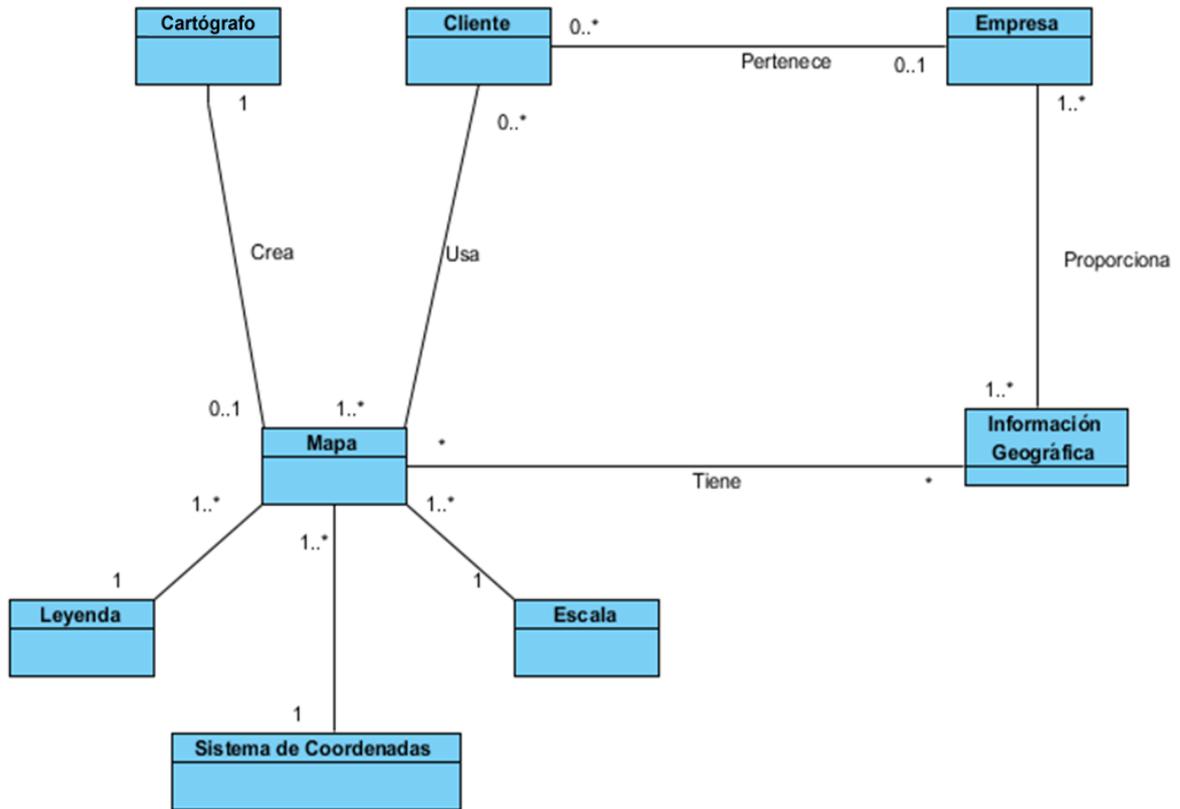


Figura 8 Diagrama del Modelo de Dominio.

### 2.2. Glosario de Términos del Dominio

Para una mejor comprensión del Modelo de Dominio a continuación se listan algunos conceptos que forman parte del entorno donde se modelará la solución informática. Estos son:

- **Cliente:** persona que trabaja en una entidad, organismo u organización que necesite trabajar o consultar algún tipo de información incluida en un mapa.
- **Cartógrafo:** persona que se asigna para la realización de cartas geográficas, al estudio y elaboración de mapas.

- **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.
- **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 por lo tanto son menos detallados que los mapas realizados con escalas más grandes.
- **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 está dado por las magnitudes de latitud y longitud en unidades de grados, minutos y segundos.
- **Empresa:** entidad, organismo u organización que solicita un servicio determinado utilizando un mapa y que proporciona la información socioeconómica referente al mismo.

### 2.3. Levantamiento de Requisitos

Los requisitos son una “condición o capacidad que necesita un usuario para resolver un problema o lograr un objetivo”. Los requisitos se clasifican en funcionales y no funcionales. Los requisitos funcionales “son capacidades o condiciones que el sistema debe cumplir, se mantienen invariables sin importar con qué propiedades o cualidades se relacionen por lo que no alteran la funcionalidad del producto”. Los requisitos no funcionales “son las propiedades o cualidades que el sistema debe tener”. Los requisitos no funcionales son aquellos requisitos que hacen que el sistema sea usable, rápido, confiable y agradable para los usuarios. **(31)**

### 2.3.1. Requisitos Funcionales

- RF1. **Seleccionar capas:** El sistema debe permitir que el usuario pueda seleccionar cualquier capa del mapa para ser visualizado en el mismo.
- RF2. **Enfocar mapa:** El sistema debe permitir que el usuario pueda realizar funciones para acercar o alejar el mapa.
- RF3. **Mover mapa:** El sistema debe permitir que el usuario pueda mover el mapa variando con el puntero del mouse o con un Navegador virtual, la posición de la vista que se presenta.
- RF4. **Localizar:** El sistema debe permitir mostrar un punto específico en el mapa.
- RF5. **Visualizar coordenadas:** El sistema debe permitir que el usuario pueda visualizar según la posición puntual en la que se encuentra el mapa, las coordenadas correspondientes a ese punto.
- RF6. **Visualizar escala gráfica:** El sistema debe permitir que el usuario pueda visualizar la escala gráfica del mapa.

### 2.3.2. Requisitos no funcionales

- Usabilidad.

El sistema podrá ser usado por personas con conocimientos básicos en el manejo de computadoras.

- Confiabilidad.

La herramienta de implementación a utilizar debe tener soporte para recuperación ante fallos y errores.

- Apariencia o interfaz externa.

Debe brindar una interfaz amigable, interactiva, intuitiva y de fácil comprensión para el usuario, facilitando en todo momento la interacción de este con el sistema.

- Rendimiento.

El tiempo de respuesta para visualizar el mapa en la pantalla será mínimo, para procesar, actualizar y recuperar información debe operar con igual rendimiento, en dependencia de la cantidad de información.

- Restricciones de diseño

El diseño debe ser sencillo, con pocas entradas, donde no es necesario mucho entrenamiento para utilizar el sistema. Se logrará un producto altamente configurable y extensible.

- Hardware

Para las computadoras clientes:

- Se requiere tengan tarjeta de red.
- Al menos 128 MB de memoria RAM.
- Procesador 512 MHz como mínimo.

Para los servidores:

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

- Software.

Para las computadoras clientes:

- Cualquier navegador web que cumpla con los estándares de la W3C.

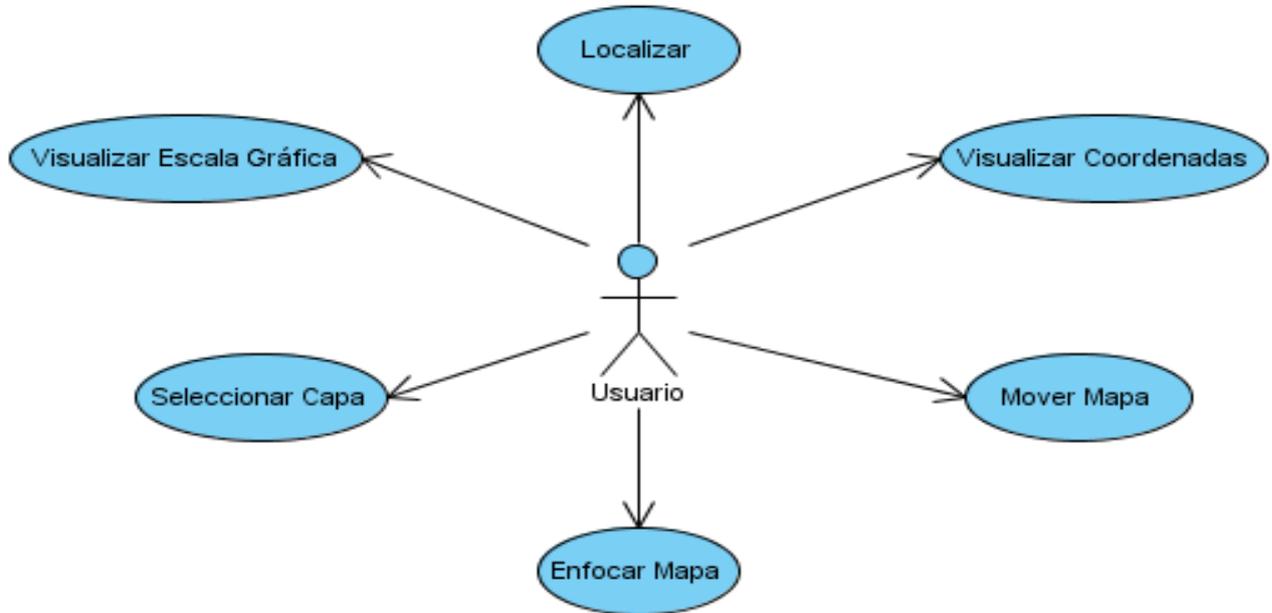
## 2.4. Descripción del Sistema Propuesto

### 2.4.1. Descripción de los actores

Actores	Descripción
Usuario	Es el encargado de inicializar todos los CU, de acuerdo a sus permisos tendrá acceso o no a las funcionalidades del sistema.

**Tabla 2** Diagrama del Modelo de Dominio.

### 2.4.2. Diagrama de Casos de Uso del Sistema



**Figura 9** Diagrama de Casos de Uso del Sistema.

### 2.5. Descripción Textual de Casos de Uso del sistema

#### 2.5.1. Descripción del Caso de Uso Seleccionar capas.

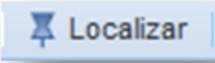
Caso de Uso	Seleccionar capas	
Actores	Usuario	
Propósito	Cambiar las capas seleccionables y visibles de un mapa.	
Resumen	El caso de uso se inicia cuando el usuario desea seleccionar o deseleccionar alguna capa y termina cuando el sistema muestra las capas seleccionadas por el usuario.	
Precondiciones		
Referencias	RF1	
Prioridad		
<b>Flujo Normal de Eventos</b>		
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>	
1. Selecciona las capas que serán visibles y seleccionables del SIG.	2. Habilitará o desactivará las opciones de visualización o de selección de las capas.	

Prototipo de Interfaz	
	
<b>Poscondiciones</b>	Queda conformado el mapa con las capas seleccionadas.

**Tabla 3** Caso de Uso Seleccionar capas.

### 2.5.2. Descripción del Caso de Uso Localizar.

Caso de Uso	Localizar
Actores	Usuario
Propósito	Mostrar la posición de un punto en el mapa mediante un criterio de búsqueda.
Resumen	El caso de uso se inicia cuando el usuario desea buscar un lugar determinado y termina cuando el sistema muestra en el mapa el punto deseado.
Precondiciones	
Referencias	RF4
Prioridad	

Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. Selecciona la opción de localizar.	2. Muestra los criterios de búsquedas disponibles para la localización seleccionada.
3. Introduce los datos del objetivo a localizar.	4. Muestra el resultado de la búsqueda en el mapa.
Prototipo de Interfaz	
	
<b>Poscondiciones</b>	Queda localizado el objetivo en el mapa.

**Tabla 4** Caso de Uso Localizar.

### 2.5.3. Descripción del Caso de Uso Enfocar mapa.

Caso de Uso	Enfocar mapa
Actores	Usuario
Propósito	Cambiar la forma de visualización del mapa.
Resumen	El caso de uso se inicia cuando el usuario desea ampliar o contraer el mapa y termina cuando el sistema muestra el mapa como el usuario desea.
Precondiciones	
Referencias	RF2
Prioridad	
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. Selecciona las opciones de enfocar el mapa. <ul style="list-style-type: none"> <li>• Zum +.</li> <li>• Zum -.</li> </ul>	2. Realiza la operación según la opción seleccionada por el actor. <ul style="list-style-type: none"> <li>• Si seleccionó “Zum +”, ver sección “Zum +”.</li> <li>• Si seleccionó “Zum -”, ver sección “Zum -”.</li> </ul>

Prototipo de Interfaz	
Sección Zum +	
Acción del Actor	Respuesta del Sistema
1. Selecciona la opción “zum +”.	2. Aumenta el zum y disminuye la escala puntualmente.
Flujo Alternativo de Eventos	
Acción del Actor	Respuesta del Sistema
	2. No aumenta el zum del mapa si la resolución que presenta hasta el momento es la última disponible.
Sección Zum -	
Acción del Actor	Respuesta del Sistema
1. Selecciona el “zum -”.	2. Aumenta la escala y disminuye el zum del mapa.
Poscondiciones	Que enfocado el mapa.

Tabla 5 Caso de Uso Enfocar mapa.

### 2.5.4. Descripción del Caso de Uso Mover mapa.

Caso de Uso	Mover mapa
Actores	Usuario
Propósito	Cambiar la forma de visualización del mapa.
Resumen	El caso de uso se inicia cuando el usuario desea mover el mapa para visualizar otras regiones y termina cuando el sistema muestra el mapa según la dirección seleccionada por el usuario.
Precondiciones	El mapa debe estar visualizado.
Referencias	RF3
Prioridad	
<b>Flujo Normal de Eventos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
1. Selecciona las opciones de mover el mapa. <ul style="list-style-type: none"> <li>• Mover arriba.</li> <li>• Mover abajo.</li> <li>• Mover derecha.</li> <li>• Mover Izquierda.</li> </ul>	2. Realiza la operación según la opción seleccionada por el actor, mostrando el mapa según las direcciones indicadas. 2.1. Si el mapa está enfocado entonces muestra los tiles restantes del mapa.
<b>Flujo Alternativo de Eventos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
	2. No hace nada si el actor llega a los bordes del mapa.
<b>Prototipo de Interfaz</b>	
	
<b>Poscondiciones</b>	Se muestra el mapa según el tipo de movimiento seleccionado.

**Tabla 6** Caso de Uso Mover mapa.

### 2.5.5. Descripción del Caso de Uso Visualizar coordenadas.

Caso de Uso	Visualizar coordenadas	
Actores	Usuario	
Propósito	Mostrar las coordenadas "X" y "Y".	
Resumen	El caso de uso se inicia cuando el actor realiza movimientos del cursor sobre el mapa y termina cuando el sistema muestra los valores de las coordenadas "X" y "Y" correspondientes al punto donde se encuentra ubicado el cursor.	
Precondiciones	El mapa debe estar visualizado.	
Referencias	RF5	
Prioridad		
<b>Flujo Normal de Eventos</b>		
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>	
1. Realiza movimientos del cursor sobre el mapa.	2. Visualiza las coordenadas X y Y correspondientes al punto donde se encuentra ubicado el cursor.	
<b>Prototipo de Interfaz</b>		
<div style="border: 2px solid black; padding: 5px; display: inline-block;">             Coordenadas(degree): -62.458701° / 12.201°           </div>		
<b>Poscondiciones</b>	Quedan visualizadas las coordenadas.	

**Tabla 7** Caso de Uso Visualizar coordenadas.

### 2.5.6. Descripción del Caso de Uso Visualizar escala gráfica.

Caso de Uso	Visualizar escala gráfica
Actores	Usuario
Propósito	Mostrar la escala gráfica del mapa.
Resumen	El caso de uso se inicia cuando el actor realiza funciones de zoom sobre el mapa y termina cuando el sistema muestra la escala gráfica correspondiente al mapa

	visualizado.
Precondiciones	
Referencias	RF6
Prioridad	
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. Realiza funciones de zum sobre el mapa.	2. Visualiza la escala gráfica del mapa.
Prototipo de Interfaz	
	
Poscondiciones	Queda visualizada la escala gráfica del mapa.

**Tabla 8** Caso de Uso Visualizar escala gráfica.

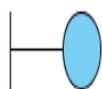
## 2.6. Análisis.

Durante el análisis, se examinan los requisitos que se describieron en la captura de requisitos, refinándolos y estructurándolos. El objetivo de hacerlo es conseguir una comprensión más precisa de los requisitos y una descripción de los mismos que sea fácil de mantener y que contribuya a estructurar el sistema completo.

### 2.6.1. Diagrama de Clases del Análisis.

Uno de los principales artefactos del análisis es el Diagrama de Clases del Análisis. Estos representan los conceptos fundamentales del dominio del problema, y las relaciones entre las clases.

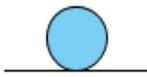
Clasificación de las clases del análisis:



Clase Interfaz: Modela la interacción entre el sistema y sus actores.



Clase Entidad: Modela información que posee una larga vida y es a menudo persistente.



Clase Control: Representa coordinación, secuencia, transacciones y control de otros objetos.

A continuación se muestra los diagramas de clases de análisis correspondiente a los casos de usos descritos:

### Diagrama de Clases del Análisis: Enfocar Mapa.

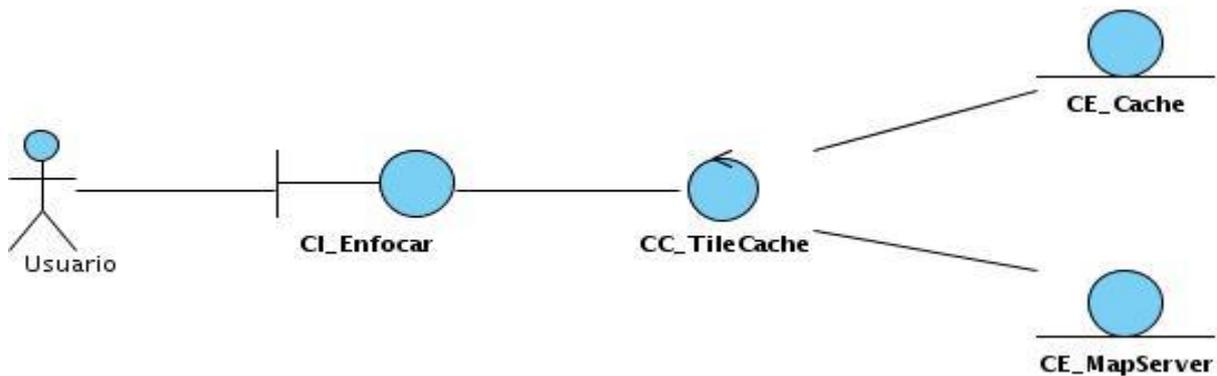


Tabla 9 Enfocar Mapa.

### Diagrama de Clases del Análisis: Escala Gráfica.

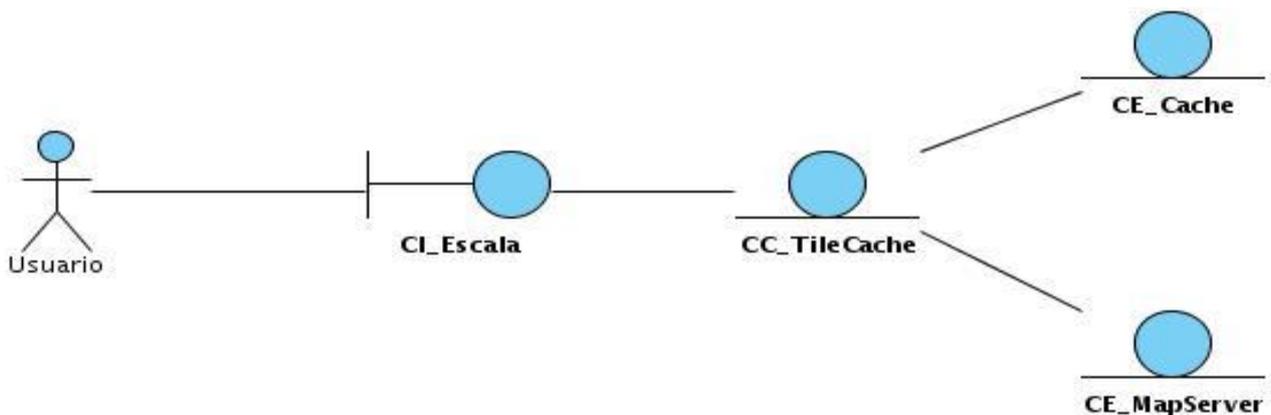


Tabla 10 Escala Gráfica.

Diagrama de Clases del Análisis:  
Localizar.

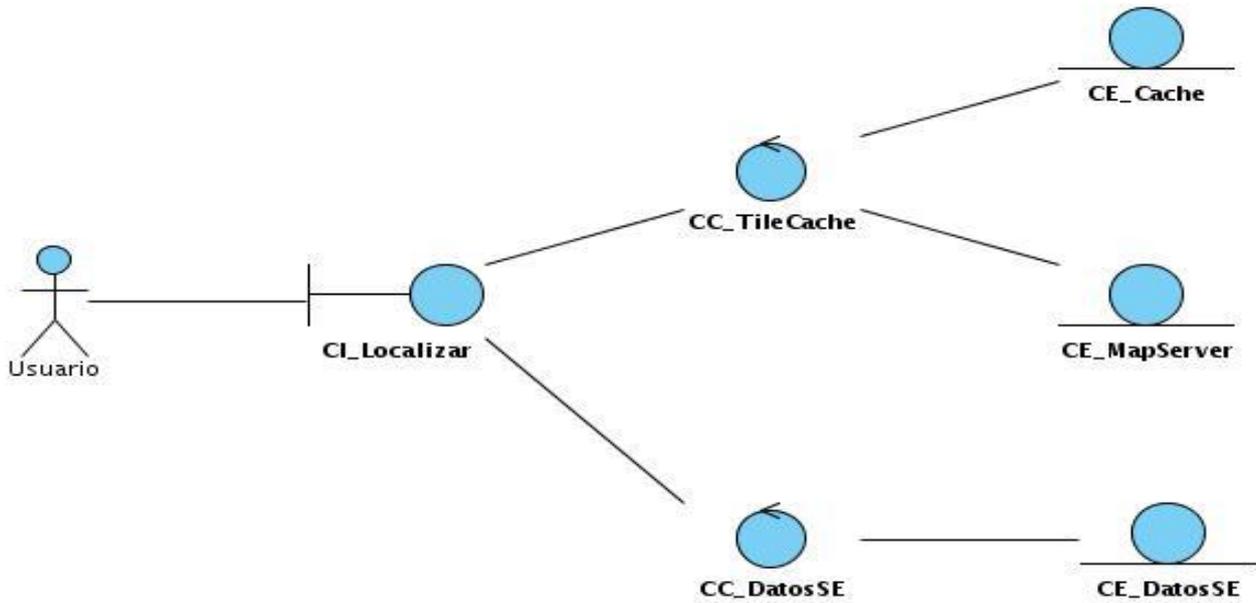


Tabla 11 Localizar.

Diagrama de Clases del Análisis: Seleccionar Capa.

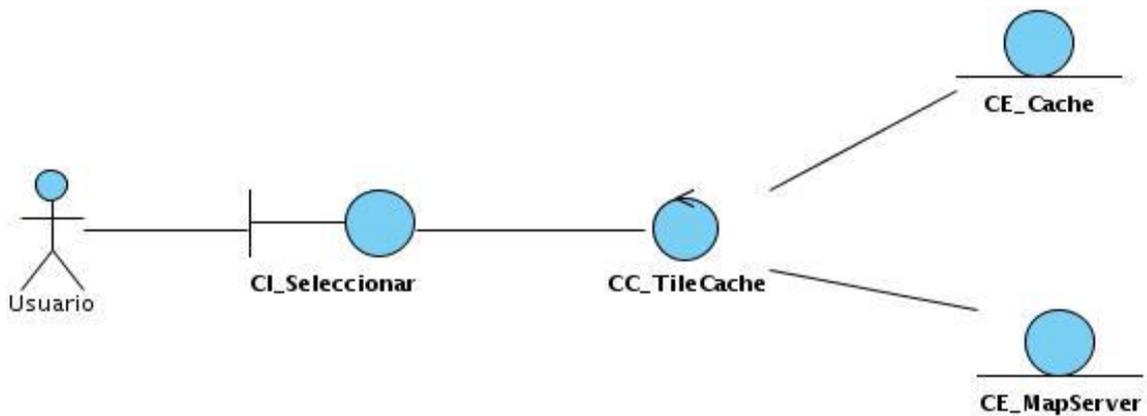


Tabla 12 Seleccionar Capa.

### Diagrama de Clases del Análisis: Visualizar Coordenadas.

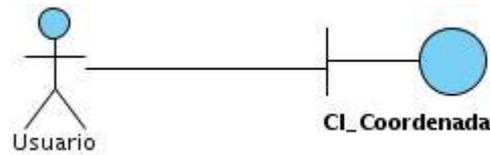


Tabla 13 Visualizar Coordenadas.

### Diagrama de Clases del Análisis: Mover Mapa.

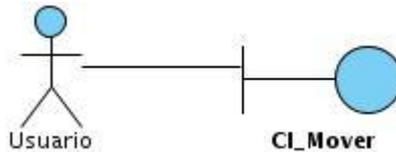


Tabla 14 Mover Mapa.

## 2.7. Diagramas de Colaboración.

Una colaboración modela los objetos y los enlaces significativos dentro de una interacción. Los objetos y los enlaces son significativos solamente en el contexto proporcionado por la interacción. Un rol describe un objeto, y un rol en la asociación describe un enlace dentro de una colaboración. Un diagrama de colaboración muestra los roles en la interacción en una disposición geométrica. Los mensajes se muestran como flechas, ligadas a las líneas de la relación, que conectan a los roles. La secuencia de mensajes, se indica con los números secuenciales que preceden a las descripciones del mensaje. **(32)**

Un uso de un diagrama de colaboración es mostrar la implementación de una operación. La colaboración muestra los parámetros y las variables locales de la operación, así como asociaciones más permanentes. Cuando se implementa el comportamiento, la secuencia de los mensajes corresponde a la estructura de llamadas anidadas y el paso de señales del programa. **(32)**

A continuación se muestra los diagramas de colaboración correspondiente a los casos de usos descritos:

### Diagrama de Colaboración: Enfocar Mapa.

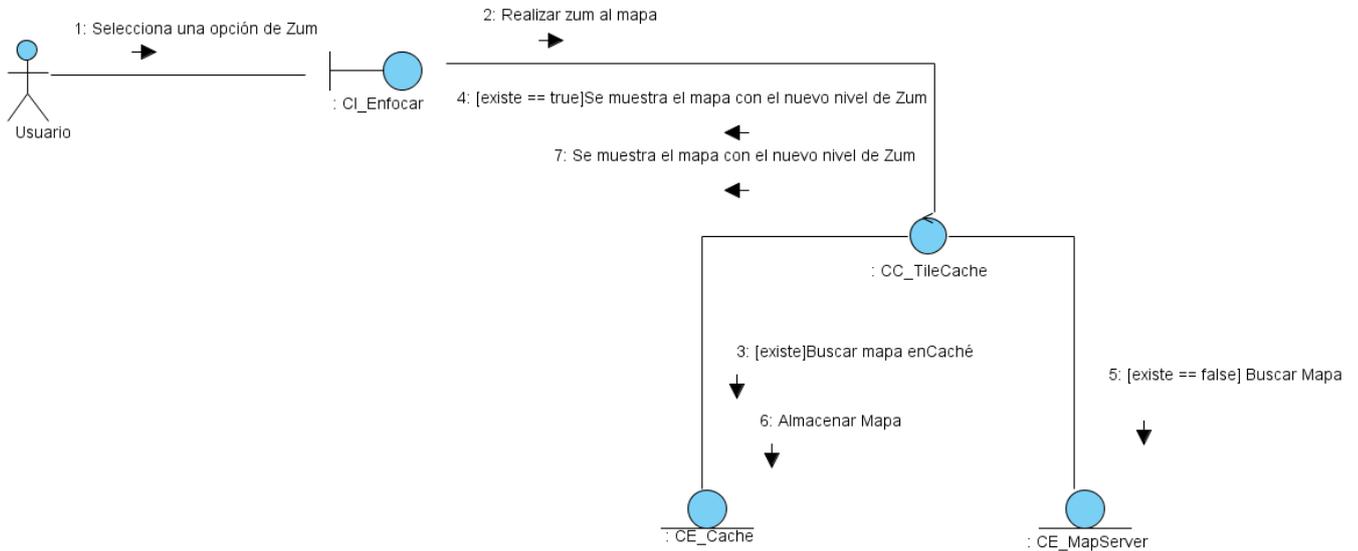


Tabla 15 Enfocar mapa.

### Diagrama de Colaboración: Escala gráfica

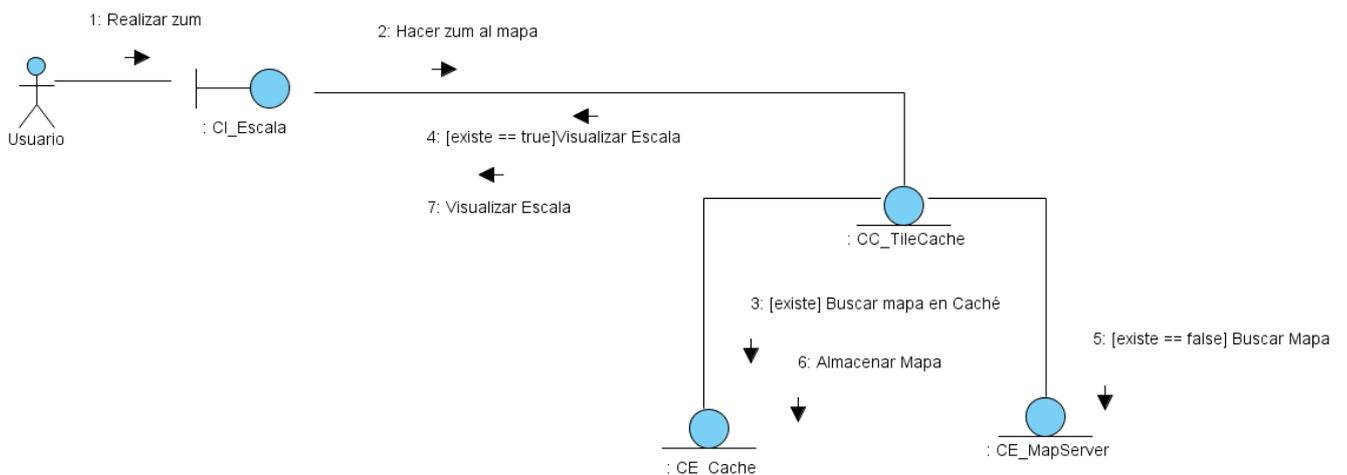


Tabla 16 Escala gráfica.

### Diagrama de Colaboración: Localizar.

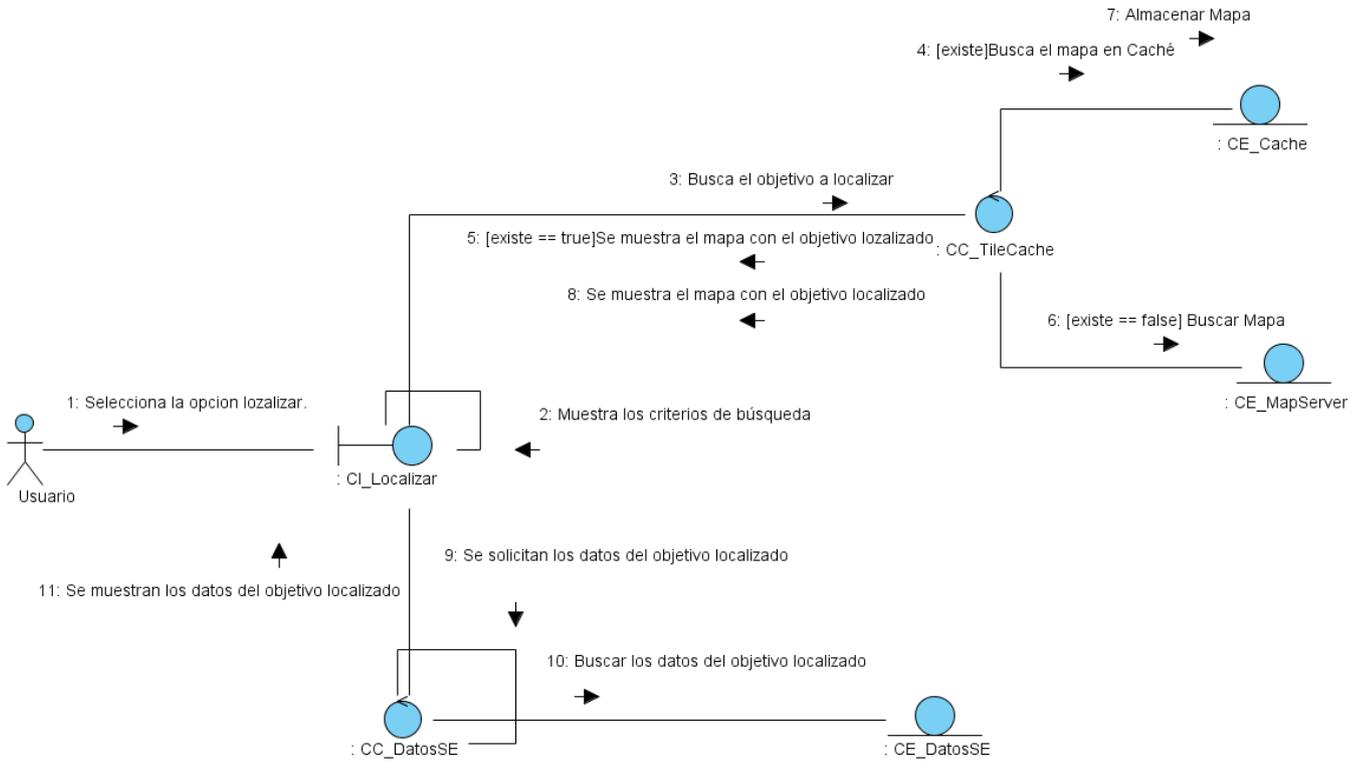
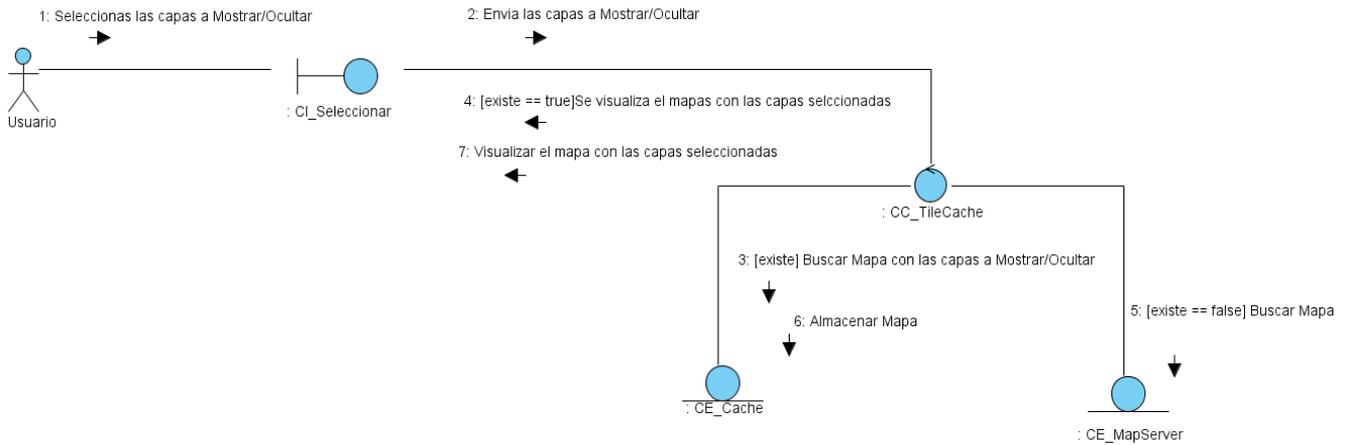


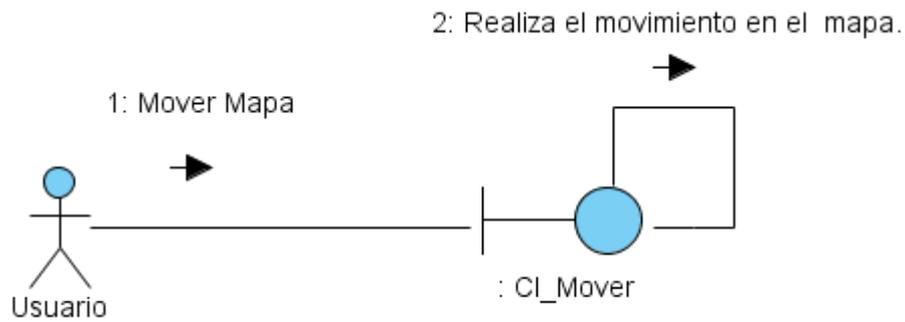
Tabla 17 Localizar.

### Diagrama de Colaboración: Seleccionar Capa.



**Tabla 18** Seleccionar capa.

### Diagrama de Colaboración: Mover Mapa.



**Tabla 19** Mover mapa.

### Diagrama de Colaboración: Visualizar Coordenadas.

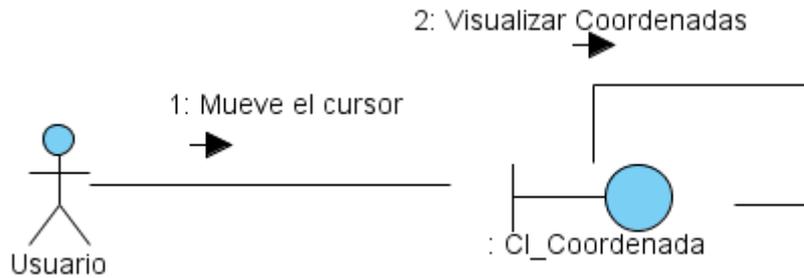


Tabla 20 Visualizar coordenadas.

### Conclusiones

En el presente capítulo se modeló básicamente lo referente a los diferentes procesos de un Sistema de Información Geográfica haciendo uso de la Metodología RUP, se identificaron además los requisitos funcionales como no funcionales existentes y posteriormente se modeló el sistema a desarrollar basado en sus casos de usos. Al finalizar el capítulo se arribaron a las siguientes conclusiones:

- La realización del Modelo de Dominio permitió comprender de una mejor forma los conceptos que forman parte del entorno donde se modela la solución informática.
- La correcta definición de los requisitos funcionales, contribuyó a identificar y describir cada una de las funcionalidades que el sistema debe permitir.
- La identificación de los requerimientos no funcionales, garantizará la estructura, presencia y eficiencia del sistema.
- A partir de la especificación de los requisitos funcionales se identificaron los casos de uso del sistema y la descripción de los mismos. Además se desarrolló el Modelo de Casos de Uso, el cual permitió tener una visión general de las funcionalidades del sistema relacionadas con los actores del mismo.
- La realización de los diagramas de clases del análisis permitió modelar el sistema en términos de clases, por cada caso de uso identificado, logrando un mejor entendimiento de las relaciones entre ellas.
- Finalmente se obtuvieron los diagramas de colaboración del análisis, los cuales permitieron lograr una mejor visión de la secuencia de mensajes que definen la relación entre las clases.

### 3. Capítulo 3: Factibilidad y Validación del Sistema Propuesto

#### Introducción

En el presente capítulo para llevar a cabo el análisis de factibilidad del sistema, se emplea el método Análisis de Puntos de Casos de Uso para la estimación del esfuerzo mediante la asignación de "pesos" a un cierto número de factores que lo afectan, para finalmente, contabilizar el tiempo total estimado para el proyecto a partir de esos factores. Se describe la estimación de costo de un Sistema de Información Geográfico y los beneficios que trae consigo. Además se valida el sistema mediante la técnica de validación, que es un proceso que tiene como objetivo demostrar que la definición de los requisitos detalla realmente lo que el cliente necesita.

#### 3.1. Estudio de factibilidad del sistema propuesto.

La factibilidad se refiere a la disponibilidad de los recursos necesarios para llevar a cabo los objetivos o metas señalados. Estudiar la factibilidad permitirá recopilar datos relevantes para el desarrollo del sistema, en base a ello se toman las mejores decisiones, para proceder a su estudio, desarrollo o implementación. El éxito del proyecto estará determinado por el grado de factibilidad que se presente en cada uno de los aspectos que a continuación se exponen:

La factibilidad se apoya en 4 aspectos básicos: **(33)**

- Factibilidad Operativa: Se refiere a que debe existir el personal capacitado requerido para llevar a cabo el proyecto y así mismo, deben existir usuarios finales dispuestos a emplear los productos o servicios generados por el proyecto o sistema desarrollado. Durante esta etapa se identifican todas aquellas actividades que son necesarias para lograr los objetivos planteados y se evalúa y determina todo lo necesario para llevarlas a cabo.
- Factibilidad Técnica o Tecnológica: Indica si se dispone de los conocimientos y habilidades en el manejo de métodos, procedimientos y funciones requeridas para el desarrollo e implantación del proyecto. Además indica si se dispone del equipo y herramientas para llevarlo a cabo, de no ser así, si existe la posibilidad de generarlos o crearlos en el tiempo requerido por el proyecto.
- Factibilidad Económica: Se refiere a que se dispone del capital en efectivo o de los créditos de financiamiento necesario para invertir en el desarrollo del proyecto.

- Factibilidad de Tiempo: En ella se verifica que se cumplan los plazos entre lo planeado y lo real, para poder llevar a cabo el software.

### Estimación por Puntos de Caso de Uso (Karner)

La estimación mediante el análisis de Puntos de Casos de Uso es un método propuesto originalmente por Gustav Karner, y posteriormente refinado por muchos otros autores. La planificación basada en casos de usos es un método de estimación del tiempo de desarrollo de un proyecto mediante la asignación de "pesos" a un cierto número de factores que lo afectan, para contabilizar el tiempo total estimado para el proyecto a partir de esos factores. (34)

A continuación, se detallan los pasos a seguir para la aplicación de éste método. (34)

#### 3.1.1. Paso1: Cálculo de Puntos de Casos de Uso sin ajustar.

El primer paso para la estimación consiste en el cálculo de los Puntos de Casos de Uso sin ajustar. Este valor, se calcula a partir de la siguiente ecuación:

$$UUCP = UAW + UUCW$$

Donde,

- **UUCP:** Puntos de Casos de Uso sin ajustar
- **UAW:** Factor de Peso de los Actores sin ajustar
- **UUCW:** Factor de Peso de los Casos de Uso sin ajustar

El valor de Factor de Peso de los Actores sin ajustar (UAW) se calcula mediante un análisis de la cantidad de Actores presentes en el sistema y la complejidad de cada uno de ellos. La complejidad de los Actores se establece teniendo en cuenta en primer lugar si se trata de una persona o de otro sistema, y en segundo lugar, la forma en la que el actor interactúa con el sistema (Ver Tabla 22).

Donde los criterios son:

Tipo de Actor	Descripción	Factor de Peso	Actores	Total
Simple	Sistema con sistema a través de interfaz de programación.	1	0	0
Medio	Sistema con sistema mediante protocolo de	2	0	0

# Capítulo 111

## Factibilidad y Validación

	interfaz basada en texto.			
Complejo	Persona que interactúa con el sistema mediante interfaz gráfica.	3	1	3
	Total		1	3

**Tabla 21** Factor de Peso de los Actores sin Ajustar.

Por tanto:

$$\mathbf{UAW} = \Sigma \text{cant actores} * \text{peso}$$

$$\mathbf{UAW} = 1 * 3$$

$$\mathbf{UAW}=3$$

El valor de Factor de Peso de los Casos de Uso sin ajustar (UUCW) se calcula mediante un análisis de la cantidad de Casos de Uso presentes en el sistema y la complejidad de cada uno de ellos. La complejidad de los Casos de Uso se establece teniendo en cuenta la cantidad de transacciones efectuadas en el mismo, donde una transacción se entiende como una secuencia de actividades atómica, es decir, se efectúa la secuencia de actividades completa, o no se efectúa ninguna de las actividades de la secuencia.

En la presente investigación, se usó otro método para calcular el Factor de Peso de los Casos de Uso sin ajustar (UUCW), método basado en clases de análisis donde se toma en cuenta el número de clases de análisis que tiene un caso de uso y se evalúa según la siguiente tabla:

Tipo de CU	Descripción	Peso	Cantidad de CU	Total
Simple	Menos de 5 clases.	5	5	25
Medio	5 a 10 clases.	10	1	10
Complejo	Más de 10 clases.	15	0	0
	Total		6	35

**Tabla 22** Factor de peso de los Casos de Uso sin Ajustar.

Por tanto:

$$\mathbf{UUCW} = \Sigma \text{cant CU} * \text{Peso}$$

$$\mathbf{UUCW} = 5 * 5 + 1 * 10$$

$$\mathbf{UUCW} = 35$$

Una vez calculado el valor del Factor de Peso de los Actores sin ajustar y el Peso de los Casos de Usos sin ajustar se puede calcular los Puntos de Casos de Uso sin ajustar:

$$\mathbf{UUCP} = \mathbf{UAW} + \mathbf{UUCW}$$

$$\mathbf{UUCP} = 3 + 35$$

$$\mathbf{UUCP} = 38$$

### **3.1.2. Paso2: Cálculo de Puntos de Casos de Uso ajustados**

Una vez que se tienen los Puntos de Casos de Uso sin ajustar se debe ajustar este valor mediante la siguiente ecuación:

$$\mathbf{UCP} = \mathbf{UUCP} \times \mathbf{TCF} \times \mathbf{EF}$$

Donde,

- **UCP:** Puntos de Casos de Uso ajustados
- **UUCP:** Puntos de Casos de Uso sin ajustar
- **TCF:** Factor de Complejidad Técnica
- **EF:** Factor de ambiente

El Factor de Complejidad Técnica (TCF) se calcula mediante la cuantificación de un conjunto de factores que determinan la complejidad técnica del sistema. Cada uno de los factores se cuantifica con un valor de 0 a 5, donde 0 significa un aporte irrelevante y 5 un aporte muy importante. En la tabla 24 se muestra el significado y el peso de cada uno de estos factores:

#### **Descripción de los factores de complejidad técnica:**

**T1:** El sistema no es distribuido.

**T2:** Los tiempos de respuesta deben ser muy altos, debe ofrecer un buen rendimiento.

**T3:** El usuario final requiere la adecuada preparación para usar el sistema.

**T4:** El procesamiento interno es medio, puesto que para la gestión de documentos se utiliza un framework como subsistema base.

**T5:** El código es reutilizable, ya que puede ser utilizado en desarrollos similares.

**T6:** Aunque son varios los módulos a instalar, el trabajo de instalación es muy sencillo.

**T7:** El sistema cuenta con una interfaz gráfica y amigable, muy sencilla de utilizar.

**T8:** El sistema debe ser multiplataforma, siendo usado bajo los Sistemas Operativos Windows y Linux.

**T9:** Se le podrá realizar algún cambio específico, no todos los cambios de forma general.

# Capítulo 111

## Factibilidad y Validación

**T10:** Algunos procesos a los que accede el usuario dependen de que esté funcionando otro.

**T11:** Se crean diferentes cuentas de usuario y se asignan a cada uno los permisos pertinentes, mostrándose a cada usuario sólo las funcionalidades del sistema sobre las cuales tiene permiso de acceso.

**T12:** No provee acceso a terceras partes.

**T13:** No es tan necesario darles preparación a los usuarios para poder usar el sistema.

Factor	Descripción	Peso	Valor asignado	Total
T1	Sistema distribuido.	2	0	0
T2	Objetivo de performance o tiempo de respuesta.	1	5	5
T3	Eficiencia del usuario final.	1	2	2
T4	Procesamiento interno complejo.	1	4	4
T5	El código debe ser reutilizable.	1	5	5
T6	Facilidad de instalación.	0.5	5	2.5
T7	Facilidad de uso.	0.5	5	2.5
T8	Portabilidad.	2	5	10
T9	Facilidad de cambio.	1	5	5
T10	Concurrencia.	1	5	5
T11	Incluye objetivos especiales de seguridad.	1	0	0
T12	Provee acceso directo a terceras partes.	1	0	0
T13	Se requieren facilidades especiales de entrenamiento a usuarios.	1	2	2
Total			43	43

**Tabla 23** Factor de Complejidad Técnica.

Por lo que:

$$\mathbf{TCF} = 0.6 + 0.01 * \Sigma (\text{peso} * \text{valor asignado})$$

$$\mathbf{TCF} = 0.6 + 0.01 * 43$$

$$\mathbf{TCF} = 0.6 + 0.43$$

$$\mathbf{TCF} = 1.03$$

# Capítulo 111

## Factibilidad y Validación

El Factor de Ambiente (EF) está relacionado con las habilidades y entrenamiento del grupo de desarrollo que realiza el sistema (Ver tabla 25). Cada factor se cuantifica con un valor desde 0 (aporte irrelevante) hasta 5 (aporte muy relevante). Es importante tener en cuenta:

- Para los factores E1 al E4, un valor asignado de 0 significa sin experiencia, 3 experiencia media y 5 amplia experiencia (experto).
- Para el factor E5, 0 significa sin motivación para el proyecto, 3 motivación media y 5 alta motivación.
- Para el factor E6, 0 significa requerimientos extremadamente inestables, 3 estabilidad media y 5 requerimientos estables sin posibilidad de cambios.
- Para el factor E7, 0 significa que no hay personal part-time (es decir todos son full-time), 3 significa mitad y mitad y 5 significa que todo el personal es part-time (nadie es full-time).
- Para el factor E8, 0 significa que el lenguaje de programación es fácil de usar, 3 medio y 5 que el lenguaje es extremadamente difícil.

### Descripción de los factores de ambiente:

**E1:** El equipo se encuentra familiarizado con el modelo utilizado.

**E2:** No se tiene experiencia en el trabajo con aplicaciones similares.

**E3:** Hay una experiencia moderada con la programación orientada a objetos.

**E4:** El analista líder es una persona capacitada.

**E5:** Hay una alta motivación en el equipo de trabajo para la construcción del producto final.

**E6:** Los requerimientos no cambian, aún cuando se implementa el sistema.

**E7:** Los miembros no trabajan a tiempo completo debido a que parte del grupo de trabajo son estudiantes.

**E8:** Se programa con PHP, un lenguaje de poca complejidad.

Factor	Descripción	Peso	Valor asignado	Total
E1	Familiaridad con el modelo de proyecto utilizado.	1.5	0	0
E2	Experiencia en la aplicación.	0.5	3	1.5
E3	Experiencia en orientación a objetos.	1	5	5
E4	Capacidad del analista líder.	0.5	5	2.5
E5	Motivación.	1	5	5

# Capítulo 111

## Factibilidad y Validación

E6	Estabilidad de los requerimientos.	2	3	6
E7	Personal part-time.	-1	3	-3
E8	Dificultad del lenguaje de programación.	-1	3	-3
Total			27	14

Tabla 24 Factor de Ambiente.

Por lo que:

$$EF = 1.4 - 0.03 * \Sigma (\text{peso} * \text{valor asignado})$$

$$EF = 1.4 - 0,03 * 14$$

$$EF = 1.4 - 0.42$$

$$EF = 0.98$$

Una vez obtenido el EF y el TCF, conjuntamente con los Puntos de Casos de Uso sin ajustar calculado anteriormente, los Puntos de Casos de Uso ajustados resultan:

$$UCP = UUCP * TCF * EF$$

$$UCP = 38 * 1.03 * 0.98$$

$$UCP = 48.41$$

### 3.1.3. Paso3: Estimación de esfuerzo a través de los puntos de casos de uso.

El esfuerzo en horas-hombre viene dado por:

$$E = UCP * CF$$

Donde,

- **E:** Esfuerzo estimado en horas-hombre
- **UCP:** Puntos de Casos de Uso ajustados
- **CF:** Factor de Conversión

Para obtener el Factor de Conversión (CF) se cuentan cuántos valores de los que afectan el factor ambiente (E1...E6) están por debajo de la media (3) y los que están por arriba de la media para los restantes (E7, E8).

- Si el total es 2 ó menos se utiliza el factor de conversión 20 Horas-Hombre / Punto de Casos de uso.
- Si el total es 3 ó 4 se utiliza el factor de conversión 28 Horas-Hombre / Punto de Casos de uso.

# Capítulo 111

## Factibilidad y Validación

- Si el total es mayor ó igual que 5 se recomienda efectuar cambios en el proyecto ya que se considera que el riesgo de fracaso del mismo es demasiado alto.

$$\text{Total EF} = (\text{Cantidad EF (entre E1 –E6)} < 3) + (\text{Cantidad EF (entre E7, E8)} > 3)$$

$$\text{Total EF} = 1 + 0$$

$$\text{Total EF} = 1$$

Por lo que:

$$\text{CF} = 20 \text{ Horas-Hombre} / \text{Punto de Casos de uso}$$

Por tanto:

$$\text{E} = \text{UCP} * \text{CF}$$

$$\text{E} = 48.41 * 20$$

$$\text{E} = 968,2 \text{ Horas-Hombre}$$

### 3.1.4. Paso4: Calcular el esfuerzo de todo el proyecto.

Para lograr una completa estimación de la duración total del proyecto, hay que agregar las estimaciones de esfuerzo de las demás actividades a la estimación del esfuerzo obtenida por los Puntos de Casos de Uso. Teniendo en cuenta el siguiente criterio, que estadísticamente se considera aceptable. El criterio plantea la distribución del esfuerzo entre las diferentes actividades de un proyecto, según la siguiente aproximación expuesta en la tabla anterior:

Actividad	Porcentaje	Horas-Hombre
Análisis	10.00 %	96,82
Diseño	20.00 %	193,64
Programación	40.00 %	387,28
Pruebas	15.00 %	145,23
Sobrecarga(otras actividades)	15.00 %	145,23
Total	100 %	986,2

**Tabla 25** Esfuerzo del Proyecto.

Suponiendo que 1 persona trabaja 24 días del mes y 8 horas por día tiene como promedio 192 horas de trabajo. Si  $E = 986,2$  horas-hombre y se divide por 192 horas se tiene un  $ET = 5,0427$ mes-hombre. Esto quiere decir que 1 persona puede dar solución al problema analizado aproximadamente en 5

meses. Teniendo en cuenta entonces que actualmente en el desarrollo del software laboran 2 personas, entonces el mismo se debe desarrollar aproximadamente en 3 meses.

### 3.2. Análisis Costo-Beneficio del proyecto.

Una vez desarrollada la técnica Planificación basada en Casos de Uso se procede a llevar a cabo el cálculo de los costos.

**CH:** Cantidad de hombres.

**TTP:** Tiempo total del proyecto.

**SM:** Salario mensual.

**TEF:** Total de esfuerzo.

**CP:** Costo del proyecto.

En el caso que trabajen estudiantes:

Asumiendo que el salario básico de un estudiante es de \$ 100.00 mensual (estipendio).

CH = 2 hombres

SM = \$100

TEF = 5

CP = SM \* CH \* TEF

CP = 100 \* 2 \* 5

CP = \$1000

TTP = TEF/CH

TTP = 5/2

TTP = 2, 5

TTP = 3 meses

Se estima que el proyecto tenga un costo total de \$1000, trabajando 2 estudiantes, durante 3 meses.

En caso que trabajen profesores:

Asumiendo que el salario básico de un profesor es de \$ 400.00 mensual.

CH = 2 personas

$$SM = \$400$$

$$TEF = 5$$

$$CP = SM * CH * TEF$$

$$CP = 400 * 2 * 5$$

$$CP = \$4000$$

$$TTP = TEF/CH$$

$$TTP = 5/2$$

$$TTP = 2.5$$

$$TTP = 3 \text{ meses}$$

Se estima que el proyecto tenga un costo total de \$4000, trabajando 2 profesores, durante 3 meses.

Por lo que:

Producto a las características propias de la Universidad de las Ciencias Informáticas de desarrollar proyectos utilizando herramientas de software libre, los gastos en cuanto a las licencias no son necesarios. Además la universidad posee los recursos tecnológicos para la puesta en marcha del desarrollo del sistema, así como el personal calificado. Es por ello que luego de realizar los cálculos y el análisis pertinente de los costos del proyecto, se llega a la conclusión de que la implementación del Sistema de Información Geográfica genérico basado en la tecnología de TileCache es factible.

### 3.3. Validación del sistema propuesto

La validación de requisitos es un proceso que tiene como misión demostrar que los requisitos definidos son realmente lo que el cliente quiere. Se revisa que sean consistentes, completos, precisos, realistas y verificables, con esto se logra que los requisitos propuestos no sean ambiguos y redundantes. La realización de estas actividades pretende evitar los altos costos que significaría el tener que corregir una vez avanzado el desarrollo.

La actividad de validación tiene como entrada el documento de requisitos, los estándares relacionados y el conocimiento de la organización. Como salida se obtiene una lista de problemas y una lista de acciones, tal y como muestra la Figura 10 (35)



**Figura 10** La validación en el proceso de requisitos.

Existen diferentes técnicas o métodos de validación entre los que se encuentran:

- Revisión de Requisitos
- Auditorías
- Prototipos
- Matrices de trazabilidad.
- Comparación de los resultados de salida del modelo con los del sistema real.
- Método Delphi.
- Test de Turing.

Los métodos que se estudiaron en la investigación fueron:

### 3.3.1. Método de Delphi

El método de validación Delphi consiste en seleccionar un grupo de expertos en el tema y circular entre ellos de manera individual un cuestionario que contenga las preguntas que se consideren más importantes sobre el tema en cuestión. Basándose en las respuestas dadas a las cuestiones planteadas, se elaboran nuevos cuestionarios que van centrándose en temas más específicos, estos nuevos cuestionarios, son enviados a los expertos junto con las respuestas obtenidas en rondas anteriores. Estos pasos pueden ir repitiéndose hasta conseguir por parte del equipo de expertos una predicción de la respuesta del sistema. En este método se excluyen las discusiones cara a cara de los miembros del equipo de expertos y su mayor desventaja es que consume mucho tiempo, por eso, en la

mayoría de los casos no se realizan muchas iteraciones en cuanto a la elaboración de nuevos test y su análisis. **(36)**

### 3.3.2. Test de Turing

Alan Turing sugirió este método como un test de inteligencia artificial. En este test, a un experto, o grupo de expertos, se le presentan resúmenes o informes de resultados de ejecución del sistema y del modelo, a los que se les ha dado el mismo formato. Estos informes se reparten aleatoriamente a los ingenieros y administradores del sistema, para ver si son capaces de discernir cuáles son los reales del sistema y cuales la imitación resultado de la simulación. Si los expertos no son capaces de distinguir entre ambos, se puede concluir que no hay evidencias para considerar inadecuado al modelo. Si descubren diferencias las respuestas sobre lo que encuentran inconsistente se puede utilizar para realizar mejoras en el modelo.

Se puede considerar que este método es el inverso al método de Delphi. En el test de Turing se consulta a los expertos para ver si son capaces de identificar las respuestas del sistema, mientras que en el de Delphi se pregunta a los expertos para que predigan las respuestas del sistema. Aunque este test parece muy intuitivo, hay muy pocos informes de su uso, ya que requiere un esfuerzo considerable para formatear las medidas de ejecución del sistema a la hora de crear el informe que se da a los expertos. Otra dificultad está en ajustar las medias del sistema real ya que en ellas intervienen elementos que no se han considerado en el modelo. Por último, este test requiere un análisis estadístico por parte del grupo de expertos para determinar si hay diferencias significativas entre el informe real y el simulado. **(37)**

### 3.3.3. Comparación de los resultados de salida.

Para poder hacer uso de este método, el sistema real debe existir, además se comparan las salidas del modelo con las del sistema, mediante algún método estadístico. El mayor inconveniente que tiene este método es que la mayoría de los procesos de salida no son estacionarios, provocando que estos test no sean directamente aplicables. **(37)**

### 3.3.4. Prototipos

Los prototipos son un método de validación ampliamente utilizado en muchas disciplinas, y en todos los casos, los principios subyacentes son los mismos: el prototipado consiste en la creación de una

maqueta o versión del producto final. Los objetivos de los prototipos varían en función de la disciplina. En el caso de la actividad de requisitos, los prototipos se utilizan, fundamentalmente, para comprobar la corrección y completitud de la especificación de requisitos.

Existen varios tipos de prototipos, cada uno de los cuales permite la realización de un tipo determinado de pruebas y con un determinado nivel de realismo. En ingeniería de requisitos, el prototipo más común es la maqueta. Una maqueta es una versión simplificada del sistema software deseado típicamente, una maqueta representa únicamente el interfaz del sistema y opcionalmente, las conexiones entre pantallas mediante la utilización de elementos activos como los botones. Si fuera necesaria mayor fidelidad, podrían codificarse partes del sistema, de tal modo que además, del interfaz, el software pudiera ofrecer algunos resultados reales. Ello es lo que se conoce como “prototipo funcional”. (35)

Construir el prototipo es únicamente una parte y quizás la más sencilla de la tarea de validación. Para validar los requisitos, es necesario realizar adicionalmente las tareas indicadas en la Figura 11. (35)

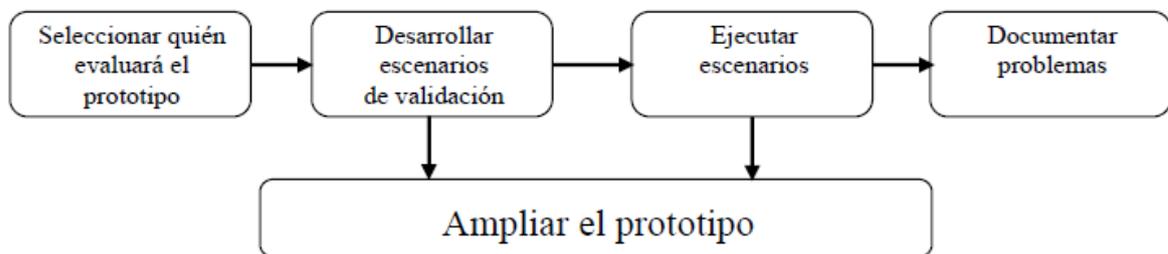


Figura 11 Prototipo en el proceso de validación de requisitos.

### 3.3.5. Matriz de trazabilidad

La trazabilidad de requisitos es clave para conseguir una exitosa gestión de los requisitos. Se utilizan las matrices de trazabilidad para validar los requisitos que se identifican en el sistema. Su principal objetivo es comprobar que los casos de usos obtenidos satisfacen todos los requisitos identificados en el sistema, verificando así que no existan inconsistencias en el proceso de incluir requisitos en los casos de usos. (Ver tabla 27)

### 3.3.6. Revisión de Requisitos

La revisión de requisitos es uno de los mejores métodos de validación de requisitos. Generalmente hablando, las revisiones de requisitos permiten: **(35)**

- Descubrir una gran cantidad de defectos en los requisitos.
- Reducir los costos de desarrollo entre un 25% y un 30%.
- Reducir el tiempo de pruebas entre un 50% y un 90%.

### 3.3.7. Auditoría

Las auditorías son un proceso sistemático que consiste principalmente en estudiar los mecanismos de control que están implantados en una empresa u organización, determinando si los mismos son adecuados y cumplen determinados objetivos. Permite detectar qué información es crítica para el cumplimiento de su misión y objetivos, identificando necesidades, duplicidades, costos, valor y barreras, que dañan la calidad de la información.

La revisión de la documentación con esta técnica consiste en un chequeo de los resultados contra una Lista de chequeo. Estas listas de chequeos consisten en una serie de preguntas a realizar, o una serie de aspectos a considerar, los cuales pretenden apoyar al analista durante la identificación de defectos en la especificación (Ver tabla 28).

### 3.4. Desarrollo de las técnicas de validación:

Todas estas técnicas anteriormente mencionadas ofrecen grandes ventajas, su diferencia radica en que unas brindan información más detalladas que otras, permitiendo que la actividad de validación sea lo más eficaz y eficiente posible. Se han seleccionado como técnicas para validar los requisitos del sistema, la revisión de requisitos, la matriz de trazabilidad y la auditoría ya que son las técnicas que más se ajustaban a las necesidades y características del sistema. A continuación serán desarrolladas estas técnicas.

#### 3.4.1. Desarrollo de la matriz de trazabilidad:

En la tabla 27 se muestra una matriz de trazabilidad donde todos los casos de usos del sistema se muestran de forma horizontal y los requisitos funcionales están ubicados de forma vertical, de manera que puedan relacionarse entre ellos.

## Capítulo 111

### Factibilidad y Validación

CUS RF	Seleccionar Capas	Localizar	Enfocar Mapa	Mover Mapa
Seleccionar capas	X			
Enfocar mapa			X	
Mover Mapa				X
Localizar		X		
CUS RF	Visualizar Coordenadas	Visualizar Escala Gráfica		
Visualizar coordenadas	X			
Visualizar escala gráfica		X		

**Tabla 26** Matriz de trazabilidad.

Tras haber realizado la tabla anterior, se puede concluir que se implementarán todos los requisitos funcionales, ya que están relacionados con todos los casos de uso del sistema. No se quedará ningún requisito sin implementar, evitándose así la inconsistencia.

#### 3.4.2. Desarrollo de la auditoría:

La tabla 28 pertenece a las listas de chequeo correspondiente a los requisitos donde su principal propósito es comprobar que estén correctos.

**Descripción de los niveles de importancia:** A alto, B bajo, M medio

Evidencia	Nivel de importancia	Procedimiento	Respuesta
1. ¿Están incluidas todas las funciones requeridas por el cliente?	A	Revisión del documento	Sí

# Capítulo 111

## Factibilidad y Validación

2. ¿Existen conflictos en los requisitos?	A	Revisión del documento	No
3. ¿Tiene alguno de los requisitos más de una interpretación?	A	Revisión del documento	No
4. ¿Está cada requisito claramente representado?	A	Revisión del documento	Sí
5. ¿Pueden los requisitos ser implementados con la tecnología y el presupuesto disponible?	A	Revisión del documento	Sí
6. ¿Está la especificación de los requisitos escrita en un lenguaje apropiado?	A	Revisión del documento	Sí
7. ¿Existe facilidad para hacer cambios en los requisitos?	A	Revisión del documento	No
8. ¿Está claramente definido el origen de cada requisito?	A	Revisión del documento	No

**Tabla 27** Lista de chequeo de requerimientos.

Una vez terminada esta lista de chequeo, se puede concluir que las funcionalidades de la aplicación evaluadas en su totalidad presentan una buena calidad, pues las mismas cumplen con todas las reglas por las que fueron evaluadas, además no se identificaron errores.

### 3.4.3. Desarrollo de las revisiones:

Con la revisión de los requisitos en este sistema se puede validar que la información transmitida esté expresada de forma clara y concisa. Esta técnica consiste en una o varias reuniones planificadas por el analista y un conjunto de colegas que, no están relacionados con el proyecto, además son competentes en la actividad de requisitos. En la misma se pretende realizar una lectura y corrección de la completa documentación de la definición de requisitos. Donde su principal objetivo es que en un futuro la aplicación pueda brindarle a las personas que en un inicio no estaban en la concepción del software un mayor entendimiento a través de la documentación generada. El resultado final de las reuniones de revisión es un documento que contiene la lista de defectos localizados y una lista de acciones recomendadas.

### Conclusiones

En este capítulo se realizó un análisis de la factibilidad del sistema donde se estimó el esfuerzo que requiere el mismo, se calculó el costo beneficio del sistema, se realizó un estudio de algunas técnicas de validación de requisitos existentes, y se arribó a siguientes conclusiones:

- Luego del análisis del costo-beneficio se concluye que es factible la puesta en marcha de la aplicación en el proyecto Aplicativos SIG.
- Una vez realizado un estudio de las técnicas de validación se determinó que en la investigación los métodos Delphi y Test de Turing no pueden ser desarrollados, debido que es necesario un grupo de experto en el tema. El sistema no cuenta con el personal capacitado para este tipo de tarea. En el test de Turing se consulta a los expertos para ver si son capaces de identificar las respuestas del sistema, mientras en Delphi se pregunta a los expertos para que predigan las respuestas del sistema.
- La Comparación de los resultados de salida y Prototipos forman parte de las técnicas que no fueron desarrolladas en la investigación. La técnica Comparación de los resultados no se pudo aplicar en el análisis del Sistema de Información Geográfica Genérico porque no se cuenta con la versión final del producto. En el caso de Prototipos es imprescindible la participación de clientes para que evalúen el producto y el sistema no posee la presencia de los mismos.
- La aplicación de las técnicas de validación de requisitos seleccionadas, permitió confirmar que los requisitos funcionales definidos están descritos con claridad, son medibles y cumplen con los estándares de calidad requeridos además queda demostrado que cumple con las necesidades del proyecto.

## Conclusiones

- En la elaboración del presente trabajo, se realizó un estudio del arte de la tecnología TileCache y se analizaron los beneficios que puede brindar el uso de esta tecnología en los Sistemas de Información Geográfica ya que permite agilizar el proceso de visualización de los mapas.
- Se plasmaron todas las actividades referentes al análisis de una posible solución para la futura implementación de un Sistema de Información Geográfica genérico basado en la tecnología TileCache, obteniéndose como resultado los artefactos necesarios para una amplia y organizada documentación y un mayor entendimiento del proceso del negocio.
- Mediante el estudio de la factibilidad del Sistema de Información Geográfica genérico se llegó a conclusiones positivas porque mediante los cálculos realizados se estableció que es factible la realización del mismo. Además se aplicaron técnicas de validación para evitar y corregir los posibles errores que podrían ocurrir, logrando como resultado la calidad exigida de los requisitos para su aprobación.

Al concluir el siguiente trabajo de diploma se proponen las siguientes recomendaciones que deberán tenerse en cuenta una vez realizado el análisis del sistema propuesto:

- Realizar la implementación y prueba para obtener así una versión del producto.
- Consultar la bibliografía con el fin de ampliar los conocimientos con respecto al tema abordado en la presente investigación.
- Realizar una ayuda para facilitar la comprensión del sistema por parte de los usuarios.

## Bibliografía

1. Barroso, Alfonso Runío y Gutiérrez, Javier Puebla. *Los Sistemas de Información Geográficos: Origen y Perspectivas*. 1997.
2. Bravo, Javier Domínguez. *Breve Introducción a la Cartografía y a los Sistemas de Información Geográfica (SIG)*. s.l. : CIEMAT, 2000.
3. Silva, r. José Luis Batista. Aplicación de Sistemas de Información Geográfica en Cuba. [En línea] 11 de 2005. [Citado el: 3 de 12 de 2010.] [http://www.mappinginteractivo.com/plantilla-ante.asp?id\\_articulo=1051](http://www.mappinginteractivo.com/plantilla-ante.asp?id_articulo=1051).
4. Gianfelici, Esteban. El dato geográfico y la información geográfica. Definiciones y diferencias. [En línea] 7 de 2008. [Citado el: 3 de 12 de 2010.] <http://www.mapasymapas.com.ar/el%20dato%20geografico.php>.
5. Definición de sistema de información - Qué es - Significado y Concepto. [En línea] 2008. [Citado el: 3 de 12 de 2010.] <http://definicion.de/sistema-de-informacion/>.
6. Definición de mapa - Qué es, Significado y Concepto. [En línea] 2008. [Citado el: 3 de 12 de 2010.] <http://definicion.de/mapa/>.
7. Páginas de SIG y Teledetección. [En línea] 2005-2006. [Citado el: 3 de 12 de 2010.] [http://www.um.es/geograf/sigmur/sigpdf/temario\\_8.pdf](http://www.um.es/geograf/sigmur/sigpdf/temario_8.pdf).
8. Páginas de SIG y Teledetección. [En línea] 2005-2006. [Citado el: 3 de 12 de 2010.] [http://www.um.es/geograf/sigmur/sigpdf/temario\\_4.pdf](http://www.um.es/geograf/sigmur/sigpdf/temario_4.pdf).
9. Consortium, Open Geospatial. Web Map Service|OGC®. [En línea] 1994-2010. [Citado el: 3 de 12 de 2010.] <http://www.opengeospatial.org/standards/wms>.
10. Gianfelici, Esteban. Instalar MapServer y ms4w. Guía de instalación fácil y en castellano. [En línea] 7 de 2008. [Citado el: 3 de 12 de 2010.] <http://www.mapasymapas.com.ar/mapserver.php>.
11. Espinosa, Antonio Membrides. *Fundamentos del Mapserver, Mapscript, PostGIS y su integración con el Cartoweb*. 2000.
12. TileCache, from MetaCarta Labs. [En línea] MetaCarta Labs, 2010. [Citado el: 3 de 12 de 2010.] [tilecache.org](http://tilecache.org).
13. Bravo., Javier Domínguez. *Breve Introducción a la Cartografía y a los Sistemas de Información Geográfica (SIG)*. s.l. : CIEMAT, 2000.
14. Santos Sanz, Miguel Angel. *Conceptos básicos sobre Sistemas de Información Geográfica (SIG)*.

15. Zaldívar, Yoenis Pantoja. *Documento Visión*. La Habana : Dirección de Calidad de la Infraestructura Productiva, 2008.
16. WMS Tile Caching - OSGeo Wiki. [En línea] 25 de 2 de 2010. [Citado el: 3 de 12 de 2010.] [http://wiki.osgeo.org/wiki/WMS\\_Tile\\_Caching](http://wiki.osgeo.org/wiki/WMS_Tile_Caching).
17. WMS Tiling Client Recommendation (OSGeo). [En línea] [Citado el: 2 de 12 de 2010.] [http://wiki.osgeo.org/wiki/WMS\\_Tiling\\_Client\\_Recommendation](http://wiki.osgeo.org/wiki/WMS_Tiling_Client_Recommendation).
18. Tile Map Service Specification (OSGeo). [En línea] [Citado el: 2 de 12 de 2010.] [http://wiki.osgeo.org/wiki/Tile\\_Map\\_Service\\_Specification](http://wiki.osgeo.org/wiki/Tile_Map_Service_Specification).
19. Ramírez, Ignacio Gámez. *Escalabilidad en servicios de mapas. Modelo de teselas en cache con OpenLayers*. s.l. : Geograma S.L.
20. Maso, Joan, Julia, Nuria y Pons, Xavier. *Historia y estado actual del futuro estándar Web Map Tiling Service del OGC*. Barcelona : s.n.
21. GeoWebCache - GeoBolivia. [En línea] 22 de 9 de 2010. [Citado el: 3 de 12 de 2010.] <http://geobolivia.abc.gob.bo/?GeoWebCache>.
22. Salinas, Jorge Gaspar Sanz y Lajara, Miguel Montesinos. Panorama actual del ecosistema de SIG libre. [En línea] 23 de 3 de 2009. [Citado el: 3 de 12 de 2010.] <https://confluence.prodevelop.es/display/pan/Panorama+del+ecosistema+de+Software+Libre+para+SIG+completo?showComments=true&showCommentArea=true>.
23. Roldan, Gabriel. OpenGeo: GeoWebCache. [En línea] [Citado el: 3 de 12 de 2010.] <http://opengeo.org/community/geowebcache/>.
24. Corti, Paolo. A day with TileCache: generating KML Super-Overlays - Thinking in GIS. [En línea] 6 de 8 de 2008. [Citado el: 4 de 12 de 2010.] <http://www.paolocorti.net/2008/08/06/a-day-with-tilecache-generating-kml-super-overlays/>.
25. "Desarrollo de Software". " *Desarrollo de Software*". [En línea] Computación, 2004. <https://pid.dsic.upv.es/C1/Material/Documentos%20Disponibles/Forms/DispForm.aspx?ID=6&Source=https%3A%2F%2Fpid.dsic.upv.es%2FC1%2FMaterial%2Fdefault.aspx&RootFolder>.
26. Guerrero, Hanz Cocchi. h@nz...el Geek. [En línea] 19 de Abril de 2007. [Citado el: 29 de Marzo de 2011.] <http://www.hanzcocchi.net..>
27. Herramienta CASE. . [En línea] 5 de febrero de 2011. <://www.slideshare.net/guest15d257/herramientas-case-508428..>
28. Visual Paradigm for UML (ME). [En línea] 5 febrero de 2011.

- [http://www.freedownloadmanager.org/es/downloads/Paradigma\\_Visual\\_para\\_UML\\_%28M%C3%8D%29\\_14720\\_p/](http://www.freedownloadmanager.org/es/downloads/Paradigma_Visual_para_UML_%28M%C3%8D%29_14720_p/)
29. Introducción al Proceso Unificado de Desarrollo de Software (RUP) y al Lenguaje Unificado de Modelado (UML). [En línea] uci. [Citado el: 26 de abril de 2011.]  
[http://eva.uci.cu/file.php/102/Curso\\_2010-2011/Clases/Semana\\_02/Conferencia\\_3/Materiales\\_complementarios/Introduccion\\_a\\_RUP\\_y\\_UML.pdf](http://eva.uci.cu/file.php/102/Curso_2010-2011/Clases/Semana_02/Conferencia_3/Materiales_complementarios/Introduccion_a_RUP_y_UML.pdf)
30. Modelo de Dominio-Supcripción Automática. [En línea] [Citado el: 2011 de 2 de 18.]  
<http://bdigital.eafit.edu.co/bdigital/PROYECTO/P005.12CDT629/capitulo6.pdf>.
31. Jacobson, I. Booch, G. Rumbaugh, J. *El Proceso Unificado de Desarrollo de Software*. Madrid : Edición en español por la Pearson Educación S.A., 2000.
32. Introducción a la Disciplina de análisis y diseño. [En línea] uci. [Citado el: 8 de abril de 2011.]  
[http://eva.uci.cu/file.php/102/Curso\\_2010-2011/Clases/Semana\\_10/Conferencia\\_10/Materiales\\_complementarios/Introduccion\\_a\\_la\\_Disciplina\\_Analisis\\_y\\_Diseño.pdf](http://eva.uci.cu/file.php/102/Curso_2010-2011/Clases/Semana_10/Conferencia_10/Materiales_complementarios/Introduccion_a_la_Disciplina_Analisis_y_Diseño.pdf)
33. Lagran, Profesor:Yussef Faran. *Estudio de Factividad*. Chile : s.n., 2000.
34. Ibarra, Marcia y Fernández, Perez. *Análisis y Diseño de los servicios de Información para la Intranet Corporativa PDVSA*. Universidad de las Ciencias Informáticas. : s.n., 2007.
35. Validación de Requisitos. [aut. libro] Master de Ingeniería de Software. *Requisitos de Sistemas de Software*.
36. Betancourt Quintanal, Danay y Abreus Ruiz, Dairon. *Desarrollo de Plug-ins de soporte para la herramienta AuditBD*. Habana : Universidad de las Ciencias Infomáticas., 2010.
37. Miles, Russ y Pilone, Dan. *Head First Software Development*.
38. Martín, R. García y Fernández, J.P. de Castro. *Benchmarking de implementaciones WMS-C de software libre*. Valladolid : s.n.

**Python:** Es un lenguaje de programación de alto nivel cuya filosofía hace hincapié en una sintaxis muy limpia y que favorezca un código legible.

**CGI:** Por sus siglas en inglés Common Gateway Interface es de las primeras formas de programación web dinámica. Es un método para la transmisión de información hacia un compilador instalado en el servidor.

**BSD:** La Licencia de Distribución de Software de Berkeley (inglés: Berkeley Software Distribution o BSD) no impone ninguna restricción a los desarrolladores de software en lo referente a la utilización posterior del código en derivados y licencias de estos programas. Este tipo de licencia permite a los programadores utilizar, modificar y distribuir a terceros el código fuente y el código binario del programa de software original con o sin modificaciones.

**OpenLayers:** Es una biblioteca de JavaScript de código abierto bajo una derivación de la licencia BSD para mostrar mapas interactivos en los navegadores web.

**Google Maps:** Es el nombre de un servicio gratuito de Google. Es un servidor de aplicaciones de mapas en la Web. Ofrece imágenes de mapas desplazables, así como fotos satelitales del mundo entero e incluso la ruta entre diferentes ubicaciones.

**Proxy:** En una red informática, es un programa o dispositivo que realiza una acción en representación de otro.

**HTTP:** Por sus siglas en inglés Hyper Text Transfer Protocol (Protocolo de transferencia de hipertexto) es el método más común de intercambio de información en la World Wide Web, el método mediante el cual se transfieren las páginas web a un ordenador

**Java:** Es un lenguaje de programación orientado a objetos.

**GeoServer:** Es un servidor de código abierto escrito en Java permite a los usuarios compartir y editar datos geospaciales.

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