

UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS  
FACULTAD 9



**TÍTULO:**

*Desarrollo de solución informática para Sistemas de  
Información Geográfica, en la Universidad de las Ciencias  
Informáticas.*

**TRABAJO DE DIPLOMA PARA OPTAR POR EL TÍTULO DE INGENIERO EN  
INFORMÁTICA**

**AUTORES:**

*Dayanelis Rodríguez Licea.*

*Alain Iglesia Herrera.*

**TUTOR:**

*Ing. Jesse Daniel Cano.*

**Ciudad de La Habana, 1 de Julio del 2010.  
“Año 52 de la Revolución”**

*Una máquina puede hacer el trabajo de 50 hombres corrientes. Pero no existe ninguna máquina que pueda hacer el trabajo de un hombre extraordinario.*

***Elbert Hubbard***

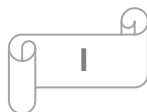
## DEDICATORIA

*A ti mamita, por hacer de mi lo que soy. No hay una sola línea de este trabajo donde tú no estés presente. Nunca me alcanzará el tiempo para agradecerte todo lo que has hecho para que hoy esté aquí. Confiando ciegamente en mí, lograste que mis objetivos se centraran solo en hacerte feliz. Eres la mejor madre del mundo y sabes que nunca te fallaré. Vive muchos años más que te necesito.*

**DAYANELIS**

*A mi abuela María,  
Dedico todos estos esfuerzos, y el poder completar hoy este maravilloso sueño.*

**ALAIN**



## AGRADECIMIENTOS

*A mi mamita, por ser lo más grande que tengo, por confiar siempre en mí y saber que yo podía llegar hasta donde estoy, por estar a mi lado en los buenos y en los malos momentos siendo madre, amiga y hermana. Sin ti no estuviera donde estoy. Te quiero más que a mí...*

*A mi hermanito, por verme como su ejemplo a seguir y porque no sé qué sería de mi vida sin su presencia...*

*A mi papito, por preocuparse y confiar siempre por mí...*

*A mi padrastro, porque ha sido como mi segundo papá...*

*A toda mi familia, por su apoyo incondicional y todo su amor...*

*A mis amistades de la universidad, por estar siempre conmigo, por formar parte de mi familia durante estos 5 años, que han sido los más lindos de mi vida. En especial a Elizabeth, Yulla, Ida, Suyen, Ari (gracias por salvarme la vida), Karu y Liu. Nunca las voy a olvidar, son de las cosas más lindas que me han pasado...*

*A Ani, Airen y Daimaris, por ser como mis hermanas. Las quiero mucho...*

*A Chuchy, por estar a mi lado dándome apoyo en los momentos en que pensé que no iba a poder realizar mi gran sueño. Por ser más que mi novio, un gran amigo, te quiero mucho y nunca te voy a olvidar...*

*A Jesse, por guiarme durante todo el desarrollo del trabajo de diploma...*

*A Alain, por ser un compañero de tesis sin igual...*

*A la Revolución, por haberme dado la posibilidad de formarme como profesional...*

*A todos los que de una forma u otra, más o menos, contribuyeron a que mis sueños hoy se estén haciendo realidad,*

GRACIAS...

DAYANELIS



## AGRADECIMIENTOS

*Cuantos momentos maravillosos debo agradecer*

*Para tantas personas maravillosas...*

*Quiero agradecer a mi madre por todo su amor, su apoyo, dedicación y comprensión,*

*Por todos sus desvelos y sus preocupaciones,*

*Para que mis sueños se hicieran realidad.*

*A mi hermana mi luz y mi guía, el ser que más quiero en el mundo y mi ejemplo a seguir.*

*A mi abuela Aida, porque siempre me ayudó y me aconsejó en todo momento.*

*A mi padre por dedicarme su esfuerzo y comprensión, por todos los consejos que*

*Siempre estuvo dispuesto a ofrecerme.*

*A mi hermano Maiquel que admiro tanto, quien siempre tuvo un oportuno y sencillo consejo a mano.*

*A mi novia a quien debo toda la fe que siempre ha tenido en mí,*

*Por ser y hacerme una mejor persona cada día,*

*Por todas las cosas buenas*

*Que compartimos.*

*A mi suegra, que me ha apoyado en todo momento, y*

*A mi suegro, la persona más tranquila que conozco.*

*A Javier, por sus constantes esfuerzos porque fuera "pa arriba del lío"...*

*A Machado un padre para mí, siempre me extendió su mano en los momentos más difíciles.*

*A Dayanelis, todas las mejores cosas del mundo para ti,*

*Por tu estrés, tu preocupación,*

*Y por ser una buena compañera,*

*Gracias.*

*A mi tutor Jesse, sin su ayuda no hubiera sido posible esta empresa.*

*A mis hermanos Osnel, Reinaldo, Vladimir, el Yiyo, Álvarez,*

*Al Cabilla, Al reinie, A yamil,*

*A Burgos quien me ayudó muchísimo en cada dificultad que se me presentó.*

*A Henry y Toledo que me ayudaron mucho con la aplicación.*

*A todos ellos gracias por haber formado parte de mi vida todo este tiempo,*

*Gracias por ser las personas maravillosas que son...*

*ALAIN*



## DECLARACIÓN DE AUTORÍA

Declaramos que (Dayanelis Rodríguez Licea y Alain Iglesia Herrera) somos los únicos autores de este trabajo de diploma titulado: **“Desarrollo de solución informática para Sistemas de Información Geográfica, en la Universidad de las Ciencias Informáticas”** y autorizamos a la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

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

\_\_\_\_\_  
Dayanelis Rodríguez Licea

Autor

\_\_\_\_\_  
Alain Iglesia Herrera.

Autor

\_\_\_\_\_  
Jesse Daniel Cano.

Tutor



## DATOS DE CONTACTO

**Tutor:** Ing. Jesse Daniel Cano.

**Formación Académica:** Ingeniero en Ciencias Informáticas (Julio/2008).

**Centro Laboral:** Universidad de las Ciencias Informáticas (UCI).

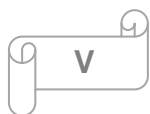
**Correo Electrónico:** [jdcano@uci.cu](mailto:jdcano@uci.cu)



## OPINIONES Y AVALES



## OPINIÓN DEL TUTOR



## **RESUMEN**

En la actualidad frecuentemente se encuentran problemas con el acceso a varios tipos de información. Estos solo pueden solucionarse con la utilización de datos espaciales. Los Sistemas de Información Geográfica (SIG) son la tecnología capaz de dar respuesta a todas estas interrogantes relacionadas con la ubicación o localización geográfica debido a que permiten manipular y almacenar datos geográficos ayudando a una mejor toma de decisiones.

Este trabajo de diploma propone desarrollar un SIG para la UCI que sea capaz de integrarse con las aplicaciones web existentes en la universidad. Luego de realizar un análisis profundo de algunas tendencias y tecnologías para la construcción del SIG se determinó que el sistema se debía realizar utilizando el marco de trabajo MapFish ya que responde a las características esenciales que debe poseer la solución.

En el documento se exponen los resultados de todo el proceso de investigación, se conceptualiza el sistema propuesto y se realiza el modelado del mismo. Finalmente, se muestra el estudio de factibilidad del producto y se plasman algunas recomendaciones con el objetivo de perfeccionar y darle continuidad al trabajo.

## **PALABRAS CLAVES**

Sistema de Información Geográfica, MapFish, Datos espaciales, Localización geográfica.

## ÍNDICE DE CONTENIDO

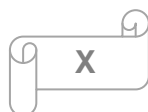
INTRODUCCIÓN.....	1
CAPÍTULO 1: “Fundamentación Teórica.” .....	5
1.1    Conceptos asociados al dominio del problema .....	5
1.1.1    Mapa .....	5
1.1.2    Software Libre.....	9
1.1.3    Aplicaciones web.....	10
1.1.4    SIG.....	10
1.1.5    SGBD.....	12
1.1.6    Marco de Trabajo .....	13
1.2    Proceso de referenciar la información geográficamente .....	13
1.2.1    Situación Problemática .....	15
1.3    Análisis de otras soluciones existentes .....	16
1.3.1    GvSIG.....	16
1.3.2    ArcGIS.....	18
1.3.3    Geomedia .....	20
1.3.4    MapFish.....	22
1.4    Metodologías de desarrollo de Software.....	24
1.4.1    Extreme Programming (XP) .....	25
1.4.2    Rational Unified Process (RUP) .....	25
1.4.3    Fundamentación de la metodología seleccionada (RUP) .....	26
1.5    Lenguaje de Modelación.....	26
1.5.1    Unified Modeling Language (UML) .....	26
1.6    Lenguaje de desarrollo Web - Hypertext Preprocessor (PHP) .....	27
1.7    Sistemas Gestores de Base de Datos .....	27
1.7.1    PostgreSQL .....	27
1.8    Herramientas propuestas para el desarrollo de aplicaciones .....	29
1.8.1    Visual Paradigm .....	29
1.8.2    PgAdmin III .....	30

1.9	Conclusiones .....	30
CAPÍTULO 2: “Presentación de la solución propuesta.” .....		31
2.1	Modelo de Dominio .....	31
2.1.1	Conceptos y eventos principales del entorno .....	31
2.2	Requerimientos Funcionales.....	33
2.3	Requerimientos No Funcionales .....	34
2.3.1	Usabilidad .....	34
2.3.2	Fiabilidad .....	34
2.3.3	Eficiencia .....	34
2.3.4	Soporte.....	34
2.3.5	Restricciones de diseño.....	34
2.3.6	Interfaz .....	35
2.3.7	Requisitos de Licencia.....	36
2.3.8	Requisitos Legales, de Derecho de Autor y otros.....	36
2.4	Descripción de la Solución Propuesta .....	37
2.4.1	Descripción de los actores del sistema.....	37
2.4.2	Descripción de los Casos de Uso del Sistema .....	38
2.5	Modelo de Diseño .....	39
2.5.1	Diagramas de Clases del Diseño .....	40
2.5.2	Estándares de la interfaz .....	41
2.6	Modelo de Datos .....	42
2.7	Conclusiones .....	43
CAPÍTULO 3: “Implementación, Prueba y Factibilidad.” .....		44
3.1	Modelo de Despliegue.....	44
3.2	Implementación. Modelo de Implementación.....	45
3.2.1	Diagrama de componentes .....	45
3.3	Prueba.....	47
3.3.1	Método de Caja Negra.....	47
3.4	Estudio de Factibilidad .....	48
3.4.1	Análisis de puntos de Casos de uso.....	48

3.5 Conclusiones .....	52
CONCLUSIONES GENERALES .....	54
RECOMENDACIONES .....	55
REFERENCIAS BIBLIOGRÁFICAS.....	56
BIBLIOGRAFÍA.....	58

## ÍNDICE DE TABLAS

Tabla 1 Estadísticas mundiales de Firefox. ....	2
Tabla 2 Actores de Sistema .....	38
Tabla 3 Descripción CU Mostrar Mapa .....	39
Tabla 4 Ejemplo de la capa edificios en el modelo de datos .....	42
Tabla 5 Mostrar Mapa.....	47
Tabla 6 Factor de peso de los actores sin ajustar.....	48
Tabla 7 Factor de peso de los CU sin ajustar .....	49
Tabla 8 Factor de complejidad técnica.....	50
Tabla 9 Factor Ambiente.....	51
Tabla 10 Relación Horas/Hombre en cada flujo de trabajo.....	52



## ÍNDICE DE FIGURAS

Figura 1 Estadísticas Mundiales de Apache. (FayerWarey, 2009).....	1
Figura 2 Arquitectura de GvSIG.....	18
Figura 3 ArcGIS.....	19
Figura 4 Arquitectura ArcGIS.....	20
Figura 5 Arquitectura de Geomedia .....	21
Figura 6 Arquitectura de MapFish.....	24
Figura 7 Diagrama de clases Del Modelo de dominio .....	32
Figura 8 Diagrama de Caso de Uso del Sistema .....	37
Figura 9 CU Mostrar Mapa.....	41
Figura 10 Diagrama de despliegue .....	44
Figura 11 Diagrama de Componentes .....	46

# INTRODUCCIÓN

Actualmente es considerable el auge tan amplio que han alcanzado las Tecnologías de la Información y de las Comunicaciones (TIC). Aunque su expansión se ha producido a una gran velocidad (entre países desarrollados), no cabe duda que existe aún una inmensa barrera entre las TIC y los países en desarrollo. Esto se debe a la posición político-económica que intentan imponer las más grandes potencias del orbe en todo momento y a la subyugación a los intereses de los “más fuertes”, del pensamiento y la posición de las demás naciones.

Dentro del continuo e incesante crecimiento de las TIC, se está desarrollando el Software Libre (SWL), el cual tiene una tendencia futura que fluye hacia un continuo desarrollo, provocando extraordinarias reacciones y polémicos debates entre sus seguidores y adversarios, trayendo como consecuencia, repercusiones positivas en la sociedad y sumando cada día más discípulos. A un paso agigantado el SWL se expande y posibilita una salida para la informatización y el comercio de los países.

Un cálido ejemplo de la rápida acogida que está teniendo, es el servidor web HTTP más popular, Apache, y Firefox es el segundo navegador más utilizado después de Internet Explorer (Microsoft), entre otros muchos casos que ponen al SWL en la cúspide por estos días. A continuación, en la Figura 1 y la Tabla 1 respectivamente se muestran una serie de estadísticas donde es posible observar y corroborar los aspectos antes mencionados.

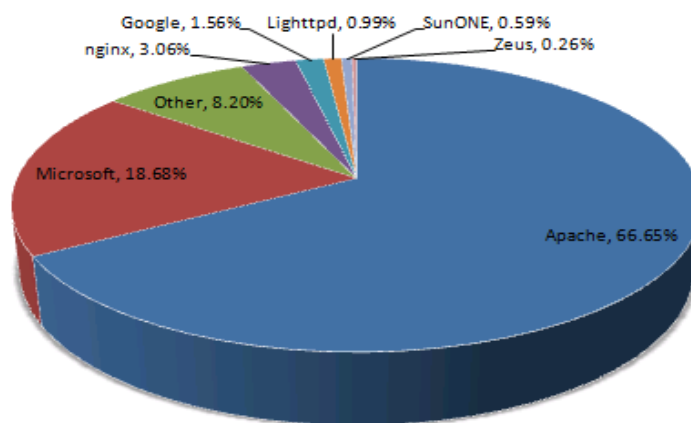


Figura 1 Estadísticas Mundiales de Apache. (FayerWarey, 2009)



Internet Explorer	63.62 %	- 1.02 %
Firefox	24.72 %	+ 0.65 %
Safari	4.36 %	-0.06 %
Chrome	3.93 %	+ 0.35 %
Opera	2.31 %	+ 0.14 %

Tabla 1 Estadísticas mundiales de Firefox<sup>1</sup>.

Cuba por ser un país subdesarrollado tiene que hacer un esfuerzo extraordinario para lograr informatizar su sociedad mediante el uso de las TIC. Los resultados que se han alcanzado a pesar de los inconvenientes que existen debido al bloqueo económico son múltiples y palpables. Hoy en día las tecnologías llegan a todos los rincones del territorio nacional y son accesibles para todas las personas sin distinción social. El desarrollo de la informática ha traído consigo el progreso de varios procesos vinculados a los diferentes sectores del país y con el uso generalizado del SWL aumentarán los resultados favorables para la economía de la isla. Con relación a lo planteado Richard Stallman en su visita a Cuba en el año 2007 expresó:

"Para eso el mayor obstáculo es la inercia social. Pero Cuba tiene experiencia en luchar contra fuertes obstáculos. Entonces puede hacerlo". (CubaSi, 2008)

"La UCI es fruto de la Batalla de Ideas, consecuencia de una coyuntura política y resultado del pensamiento de un hombre que en las más difíciles adversidades nunca ha perdido la confianza en el futuro."<sup>2</sup>

La Universidad de las Ciencias Informáticas (UCI) es un eslabón importante en este aspecto, debido a la fuerte vinculación que tienen sus estudiantes con las TIC. Además, se encuentra inmersa en el desarrollo de productos basados en SWL que sustentan la libertad del comercio con empresas sin necesidad de tener que pagar una licencia de software privativo, lo cual representa una considerable pérdida al país. En la actualidad sobre la base del SWL se trabaja en los Sistemas de Información Geográfica (SIG) con el objetivo de capturar, manejar, manipular, analizar y modelar datos geográficos espacialmente referenciados.

<sup>1</sup> Tomado de Abueloinformatico. [En línea] 2009. <http://www.fayerwayer.com/2009/04/apache-sigue-siendo-el-rey-entre-los-grandes-sitios>.

<sup>2</sup> Tomado de Diario Granma. 2007. Granma. [En línea] 2007. <http://www.granma.cubaweb.cu/2007/07/20/nacional/artic05.html>.

La principal razón para utilizar un SIG es la gestión de información espacial. Estos sistemas son capaces de separar la información en varias capas temáticas y almacenarlas independientemente, permitiendo el trabajo con ellas de forma rápida y sencilla; y además da la posibilidad al profesional de relacionar la información que ya existe mediante la topología de los objetos, con el fin de generar otra nueva que no se podría obtener de forma diferente. (GeoInfo, 2008)

Por lo anterior Cuba se ha interesado en la utilización de estos sistemas, justamente porque uno de sus principales objetivos radica en la independencia tecnológica que sea capaz de adquirir, ya no solo de la privatización del software como tal, sino, de cierta forma, para lograr sus propios productos y componentes de software, con los cuales puedan tener todo el control de la gestión que se realice en cualquier proyecto a desarrollar.

La **situación problemática** viene dada debido al gran esfuerzo que requiere adherir los Sistemas de Información Geográfica que existen en la UCI con las demás aplicaciones web. Actualmente el software se atrasa en su desarrollo, puesto que para poder integrarlos con las aplicaciones existentes, los desarrolladores tienen que estudiar y analizar dichas aplicaciones, lo cual conlleva a un consumo enorme de tiempo. Además, el código no es reutilizable a causa de su especificidad para determinadas tareas.

La situación antes expuesta genera el siguiente **problema a resolver**: ¿Cómo disminuir el esfuerzo que implica integrar los Sistemas de Información Geográfica con los que se cuentan en la universidad, con las aplicaciones web existentes?

La presente investigación tiene como **objeto de estudio** el proceso de referenciar la información geográficamente y como **campo de acción** la automatización del proceso de referenciar la información geográficamente en la UCI.

El **Objetivo General** del trabajo es desarrollar un SIG para la UCI, utilizando una plataforma que permita disminuir el esfuerzo necesario para integrarlo con las demás aplicaciones web existentes en la universidad.

Del objetivo antes planteado se derivan las siguientes **tareas de la investigación**:

1. Elaborar el marco metodológico y teórico en el que está enmarcado el sistema.

2. Identificar los principales aspectos a automatizar en el proceso obtención de información geográfica utilizando mapas.
3. Analizar y Diseñar la herramienta de manera que tribute a la implementación del producto.
4. Implementar un producto que cumpla con las especificaciones identificadas y de acuerdo con el diseño elaborado.
5. Validar el trabajo mediante el uso de técnicas para este fin.

### **Idea a defender:**

Si se desarrolla un SIG para la UCI usando una plataforma indicada, se disminuirá el esfuerzo que implica la integración de las aplicaciones web existentes con el Sistema de Información Geográfica.

### **Métodos de Investigación**

#### Métodos teóricos.

- Analítico–sintético: Se utilizará para analizar y comprender la teoría y documentación relacionada con el tema de investigación, permitiendo así, extraer los elementos más relacionados e importantes con el objeto de estudio.
- Análisis histórico–lógico: Se realizará un análisis histórico exhaustivo de la historia de los SIG, posibilitando el análisis de la trayectoria de estos sistemas para una mejor comprensión de los mismos.

#### Métodos Empíricos.

- Observación: Este método es de suma importancia ya que permitirá distinguir a partir de la situación real que se investiga cómo se desarrolla al proceso que constituye el objeto de estudio.

# **CAPÍTULO 1: “Fundamentación Teórica.”**

Este capítulo comprende las terminologías relacionadas con los SIG, los marcos de trabajo, el software libre y aquellos conceptos que permitan un mejor entendimiento de la situación problemática, en aras de lograr una visión general de la misma, así como la descripción del objeto de estudio, el análisis de otras soluciones existentes y de varias herramientas para el desarrollo del sistema.

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

Con el objetivo de lograr una mayor comprensión y una visión general de la investigación se hace necesario estudiar varios conceptos que se muestran en los siguientes tópicos. Todas estas unidades cognitivas son utilizadas directa o indirectamente durante el desarrollo del trabajo de diploma.

### **1.1.1 Mapa**

Los mapas geográficos son representaciones reducidas, generalizadas y matemáticamente determinadas, de la superficie terrestre sobre un plano, en las cuales se interpreta la distribución, el estado y los vínculos de los distintos fenómenos naturales y sociales, seleccionados y caracterizados de acuerdo con la asignación concreta del mapa. (Salitvech Konstantin, 1981)

#### Tipos de mapas

Los mapas pueden ser clasificados según diferentes aspectos, los más utilizados son:

- Carácter territorial.
- Especialización del contenido.
- Escala.
- Asignación.

Por su carácter territorial o magnitud espacial se distinguen mapas de:

- Todo el mundo: mapamundi o planisferio.
- Océanos y mares.

- Continentes.
- Partes de continentes: regiones.
- Países.
- Provincias, etc.

Esta clasificación va de lo general a lo particular y puede subdividirse más y orientarse a aspectos físico-geográficos, socioeconómicos y político-administrativos.

### ***Especialización del contenido***

Los mapas geográficos, de acuerdo con la especialización del contenido se dividen en mapas generales y especiales o temáticos.

### ***Mapas geográficos generales***

Los mapas generales o de referencia tienen como objetivo reflejar, de manera exacta y representativa, las relaciones entre una selección de diferentes accidentes geográficos (de la superficie terrestre). Pueden ser considerados la base sobre la que se construyen otros mapas o estudios relacionados. Se pueden subdividir en:

- Topográficos: representación exacta y detallada de la superficie del terreno y los objetos que contiene. Escalas 1:10 000 – 1:100 000.
- Planimétricos: situación horizontal.
- Catastrales: límites de las subdivisiones de la tierra.
- Batimétricos: profundidades.
- Generales o de conjunto: regiones, continentes, mundo. Escalas menores que 1:100 000.

Los mapas geográficos generales presentan una gran variedad y cantidad de información, con el objetivo de brindar una imagen, lo más fiel posible, de un territorio determinado. Su contenido abarca generalmente: hidrografía, relieve, poblaciones, red vial, límites y algunos elementos de vegetación y suelos. Sus características son determinadas fundamentalmente por la escala.

### ***Mapas especiales o temáticos***

Los mapas especiales o temáticos se realizan tomando los generales como referencia pero destacando, más que nada, la funcionalidad, rama o temática para la cual está confeccionada. Estos mapas están diseñados para satisfacer un objetivo específico. Pueden existir muchas divisiones en esta categoría; sin embargo se pueden agrupar en tres clases fundamentales:

- Físico – geográficos.
- Socio – económicos.
- Técnicos.

Estas clases a su vez se subdividen en conceptos de género y especie.

- Mapa físico – geográfico.
- Geológicos: estratigráficos, tectónicos, litológicos, hidrogeológicos, de yacimientos minerales.
- Geofísicos.
- Del relieve de la superficie terrestre: geomorfológicos, hipsométricos, de carso.
- Fenómenos atmosféricos: meteorológicos, climáticos.
- Hidrósfera: oceanográficos, hidrológicos, hidrográficos.
- Suelos.
- Botánicos.
- Zoo geográfico.
- Paisajes.

### ***Mapas socioeconómicos***

- Población: distribución de la población, composición de la población (sexo – etárea, profesional, etc.), movimiento natural (mortalidad, migraciones, etc.).
- Económicos: recursos naturales (para su evaluación económica), industria, agricultura y economía forestal, transporte y los medios de comunicación, comercio, intercambio y relaciones financieras, economía general.

- Turísticos: recursos naturales para el turismo, infraestructura hotelera y extra hotelera, indicadores turísticos, etc.
- Infraestructura social: servicios educacionales, recreativos, culturales, deportivos, médicos, etc.
- Culturales: religiones, lingüísticos, etnográficos, etc.
- Sociales: conflictos sociales, movimientos sociales, salud, etc.
- Históricos.
- Políticos.
- Medio Ambiente.

### ***Mapas Técnicos***

- Navegación marítima.
- Vuelos.
- Proyectos.
- Otros.

Esta clasificación tiene cierta condicionalidad por la estrecha interacción entre la sociedad y la naturaleza.

### ***Escala***

Según la escala los mapas geográficos se subdividen en:

- Escala pequeña - 1: 250 000 a 1: 1 000 000
- Escala mediana - 1: 50 000 a 1: 100 000
- Escala grande - 1:10 000 a 1:25 000

De acuerdo con la escala los mapas temáticos o especiales se dividen en:

- Grande: mayores que 1: 250 000
- Mediana: 1: 250 000 – 1: 1 000 000
- Pequeña: menores que 1: 1 000 000 (Oviedo Álvarez, 2005)

## 1.1.2 Software Libre

La Fundación de Software Libre lo define de la siguiente manera: “Es la libertad de los usuarios para ejecutar, copiar, distribuir, estudiar, cambiar y mejorar el software”. De manera más precisa, está resumido en cuatro libertades para los usuarios del software:

- Libertad 1: ejecutar el programa, con cualquier propósito.
- Libertad 2: estudiar cómo funciona el programa, y adaptarlo a sus necesidades. El acceso al código fuente es una precondition para esto.
- Libertad 3: distribuir copias, con lo que puede ayudar a su vecino.
- Libertad 4: mejorar el programa y hacer públicas las mejoras a los demás, de modo que toda la comunidad se beneficie. El acceso al código fuente es una precondition para esto.

Es muy habitual confundir los términos “Gratis” y “Libre”, debido en gran medida a la ambigüedad de la palabra inglesa "free". El software gratis puede incluir restricciones que no se adaptan a la definición de software libre, incumpliendo de esta forma con las cuatro libertades, por ejemplo, puede no incluir el código fuente, puede prohibir explícitamente a los distribuidores recibir una compensación a cambio, etc. Por tanto, el término “Software Libre” no significa 'no comercial'. Un programa libre debe estar disponible para uso productivo, desarrollo y distribución comercial. Dicho desarrollo del SWL ha dejado de ser inusual y es muy importante.

Por tal motivo la comunidad de software libre puso todos sus empeños en legalizar y hacer cumplir dichas libertades y con el objetivo de asegurar que el software GNU<sup>3</sup> permaneciera libre para que todos los usuarios pudieran "ejecutarlo, copiarlo, modificarlo y distribuirlo", el proyecto debía ser liberado bajo una licencia diseñada para garantizar esos derechos al tiempo que evitase restricciones posteriores de los mismos. La idea se conoce en inglés como copyleft (copia permitida) contenida en la Licencia General Pública de GNU, en clara oposición a copyright (derecho de copia).

---

<sup>3</sup> Proyecto iniciado por Richard M. Stallman, que tiene como objetivo desarrollar un sistema operativo compatible con UNIX 100% libre y operacional.



### **1.1.3 Aplicaciones web**

Jim Conallen define una aplicación web como un sitio donde la navegación, y la entrada de datos por parte de un usuario, afectan el estado de la lógica del negocio. En esencia, una aplicación web usa un sitio como entrada a una aplicación típica. Si no existe lógica del negocio en el servidor, el sistema no puede ser llamado aplicación web. (Conallen, 1999)

Las aplicaciones web se desarrollan como una extensión de los sitios para agregar funcionalidades al proceso; ejecutan la lógica del negocio, ya que lo más importante en ellas es su modelación y no los detalles de la presentación. Esto permite que cualquier modificación en el diseño de interfaz o la manera de mostrar los datos no alteren las funcionalidades del sistema.

Los usuarios de la WWW, hoy día, acceden a miles de aplicaciones web mediante un servidor web, ya sea directamente conectado a Internet o mediando una intranet. Son sorprendentes las estadísticas de la penetración internacional en el mundo digital a través de las aplicaciones web. Sublimesolutions.com tiene publicada una serie de estadísticas que se muestran en el Anexo #1 respecto a este tema tan concurrente. Los datos obtenidos, a pesar de ser del 2006, ya revelan resultados de cómo ha sido la incursión masiva en el ciber mundo en ese año, lo cual ha ido creciendo a grandes pasos con el transcurso de los siguientes hasta la actualidad.

### **1.1.4 SIG**

Un Sistema de Información Geográfica (SIG o GIS por su acrónimo inglés) es una integración organizada de hardware, software, datos geográficos y personal, diseñada para capturar, almacenar, manejar, analizar, modelar y representar en todas sus formas la información geográficamente referenciada con el fin de resolver problemas complejos de planificación y gestión. También puede definirse como un modelo de una parte de la realidad referido a un sistema de coordenadas terrestre y construido para satisfacer unas necesidades concretas de información. (GeoInfo, 2008)

Los SIG funcionan como una base de datos con información geográfica (datos alfanuméricos) que está asociada por un identificador común a los objetos gráficos de un mapa digital. Es por ello que señalando un objeto se conocen sus atributos o preguntando por un registro de la base de datos se puede saber su localización en la cartografía. Las características distintivas que posee un SIG respecto a otros sistemas

para dibujo, tratamiento de imágenes, diseño cartográfico, entre otros, son las operaciones de análisis de datos.

La evolución de los SIG se pueden ver a través de varias etapas: los primeros SIG desarrollados tenían el objetivo específico de resolver problemas de información referente a un proyecto específico; el número de usuarios era muy reducido y se conocían como SIG de escritorios debido a que su uso requería sólo de un ordenador. Luego aparecieron los SIG más desarrollados, que podían emitir la información manejada en una oficina o un departamento al interior de una organización, ampliándose además el número de usuarios. Después llegaron los SIG de empresas también llamados SIG corporativos. El último paso en la su evolución corresponde a los que están al servicio de la sociedad, que se caracterizan por soportarse en sólidas tecnologías informáticas y por el número ilimitado de usuarios.

Los SIG están compuestos por 5 componentes principales que permiten su funcionamiento. Estos son:

Hardware, Software, Datos, Personas y Métodos. (GeoInfo, 2008)

Hardware: Los SIG actuales se ejecutan sobre varias plataformas, estas pueden variar, desde servidores a computadores de escritorio. Los dispositivos y periféricos opcionales que poseen los ordenadores personales de hoy tienen su utilidad principalmente en la entrada y salida de los datos (módem, escáner, GPS, impresora en color, entre otros).

Software: Cada programa o paquete SIG entrega las funciones y las herramientas que se requieren para generar, almacenar, analizar y desplegar información geográfica. La elección del programa SIG por parte del usuario depende de las aplicaciones y análisis requeridos.

Base de Datos: Tienen vital importancia dentro de los componentes SIG y va a depender de ellas los resultados que se puedan obtener. El sistema integra datos espaciales con otras fuentes e incluso puede utilizar un sistema administrador de Bases de datos (DBMS) para organizar, mantener y manejar los datos espaciales.

Recursos humanos: Las personas son las encargadas de administrar el sistema y poner el SIG en funcionamiento. Sin el personal es imposible que el SIG trabaje y de estos depende su éxito.

Métodos: Para el trabajo con un SIG se necesita tener una estructura organizada que permita concebir un plan bien diseñado y de acuerdo con las reglas de la empresa, modelos de las actividades propias de cada organización.

Por lo general el Hardware y el Software son los componentes más caros, pero no los más importantes. Se pueden considerar la Base de Datos y los Recursos Humanos como los de suma prioridad ya que, por un lado, la disponibilidad de los datos sobre los que se realizan las operaciones, la precisión y actualidad son los que determinan la salida de los resultados, por el otro, es casi imposible que el personal aprenda con profundidad el manejo de un SIG en un corto período de tiempo; leyendo solo el manual del programa, es imprescindible la práctica e interacción con este.

Las funciones básicas de los SIG son: la captura de información, dentro de la cual se pueden encontrar la digitalización y el procesamiento de imágenes de satélite, fotografías, etc. y el análisis que se puede realizar con los datos gráficos y no gráficos.

Entre las razones fundamentales para utilizar un SIG se encuentra la gestión de información espacial. Además, con la utilización de los SIG se aprovechan e incorporan el avance constante de varias materias como la estadística, los computadores, los sensores remotos, los sistemas expertos y las redes neuronales, entre otros.

### **1.1.5 SGBD**

Sistema de gestión de base de datos o en inglés Database Management System (DBMS), es una agrupación de programas que sirven para definir, construir y manipular una base de datos.

Definir una base de datos: consiste en especificar los tipos de datos, estructuras y restricciones para los datos que se almacenarán.

Construir una base de datos: es el proceso de almacenar los datos sobre algún medio de almacenamiento.

Manipular una base de datos: incluye funciones como consulta, actualización, etc. de bases de datos. (Diccionario Informático, 2009)

### 1.1.6 Marco de Trabajo

Un marco de trabajo o framework (por su acrónimo en inglés) puede ser definido como: una estructura de soporte definida en la cual otro proyecto de software puede ser organizado y desarrollado. Típicamente, un framework puede incluir soporte de programas, bibliotecas y un lenguaje de scripting entre otros software para ayudar a desarrollar y unir los diferentes componentes de un proyecto. (Corrales Martínez, 2009)

Existen grandes cantidades de marcos de trabajos orientados a múltiples lenguajes y con disímiles funcionalidades. En ocasiones es muy difícil escoger un framework para la realización de un software y lo ideal es buscar el que más se adapta a las necesidades del desarrollador o de lo contrario hacer adaptaciones propias.

Con el uso de los marcos de trabajo se persigue que el proceso de desarrollo del software se realice con más agilidad, que se usen patrones y que aumente la reutilización del código existente garantizando que una parte o la totalidad de un programa informático puedan emplearse en la construcción de otro programa. Así se aprovecha el trabajo realizado con anterioridad, se ahorra tiempo y se garantiza la estabilidad de la arquitectura a cambios futuros.

## 1.2 Proceso de referenciar la información geográficamente

Para la comprensión del objeto de estudio es necesario analizar varios conceptos que conllevan a un mejor entendimiento del mismo. Los principales conceptos a analizar son:

Información: Conjunto de datos acerca de algún suceso, hecho o fenómeno, que organizados en un contexto determinado tienen su significado, cuyo propósito puede ser el de reducir la incertidumbre o incrementar el conocimiento acerca de algo. (Thompson, 2008)

Geografía: Ciencia que tiene por objeto el estudio de la superficie terrestre, la distribución espacial y las relaciones recíprocas de los fenómenos físicos, biológicos y sociales que en ella se manifiestan. (Enciclopedia Libre Universal en España, 2009)

Información Geográfica: Conjunto de datos espaciales que de una forma u otra están referenciados para brindar información de distintos fenómenos a los usuarios.

Cartografía Digital: Es la parte de la cartografía que se ocupa del tratamiento y la representación de datos mediante números o caracteres de un repertorio finito, con la que se puede, entre otras cosas, calcular, representar, interpretar y situar las cavidades en un formato digital.

Ventajas:

- Hacer mapas y actualizarlos más rápidamente.
- Más baratos, una vez instalados los nuevos sistemas.
- Permite hacer mapas para satisfacer necesidades específicas. Mapas a la carta.
- Permite la experimentación con diversas representaciones gráficas.
- Admite nuevas posibilidades de cartografía y mapas multimedia.

Inconvenientes:

- Se necesita hardware y software adecuado y conocimientos para efectuar el procesamiento y la lectura.
- Los elementos hardware y software se vuelven obsoletos muy rápidamente. Esto trae consigo problemas de almacenamiento.
- Corrupción de ficheros.
- Al actualizar la cartografía, en algunos casos no se almacenan los datos históricos.
- Se pierde la visión de conjunto en cuanto se hace un zoom, cosa que no ocurre el papel. (Dávila Martínez, 2004)

Dato Espacial: Un dato espacial puede entenderse como la representación de un objeto en dos o tres dimensiones, la cual tiene atributos inherentes al espacio; por lo que, por sí mismo, cuenta con los atributos de dimensión y de localización.

A través de los datos espaciales, tomando como ejemplo la movilidad, se pudiera representar con puntos las estaciones del tren ligero de la ciudad, con líneas se representarían las vías a lo largo de las cuales el tren transita y también mediante polígonos se pudieran representar las colonias en las que se encuentran cada una de las estaciones. (Revista Electrónica Tropical , 2008)

Análisis espacial: Comprende el conjunto de conceptos y procedimientos utilizados para abordar el estudio de la estructura y las relaciones territoriales a partir del conocimiento de la posición de las entidades geográficas y las características de las variables seleccionadas para su estudio. (Unzurrunzaga, y otros, 2007)

Para realizar un análisis espacial es indispensable tener un modelo de interacción visual, el cual permite definir la relación que existe entre los objetos.

Capa: Es en un software de edición de imágenes, un componente de una imagen que se puede manipular independientemente de las otras partes. (Glosario.net, 2006)

Permite tratar separadamente los objetos que componen el mapa, facilitando el trabajo con ellos. Pueden contener textos, imágenes, formularios e incluso otras capas y pueden ser colocadas unas delante de otras haciendo más fácil el trabajo de los programadores y diseñadores.

Formato vectorial: Esta es una forma de representar la información digital a través de la combinación de puntos, líneas y polígonos. Permite realizar ampliaciones a las imágenes sin que estas pierdan resolución posibilitando mayor calidad de la información mostrada. Es el formato que más se corresponde con la representación de objetos reales llevados al espacio y cuando se trabaja con datos promediados sobre una extensión territorial.

Formato ráster: Es una forma de representar la información digital a través de celdas, donde cada una tiene un número asignado según el atributo de la superficie al que pertenezca.

### **1.2.1 Situación Problemática**

Actualmente se requiere un gran esfuerzo para integrar las aplicaciones web con los SIG en la UCI. Esto se debe principalmente a que no existe un sistema de información geográfica basado en una plataforma que permita dicha integración. También trae como consecuencia un costo de tiempo enorme, pues el

software se atrasa en su desarrollo debido a que los desarrolladores tienen que estudiar y analizar las aplicaciones web que ya existen. El código resultante es poco reutilizable debido a su especificidad lo cual conlleva a que no se pueda utilizar en la construcción de otros programas.

La mayoría de los sistemas que hay hoy encargados de gestionar y analizar información espacial lamentablemente están basados en software propietario lo cual tiene grandes desventajas comparados con los basados en software libre, ya que no se puede redistribuir el producto libremente y además tras el fuerte bloqueo económico por parte de Estados Unidos el país se ve limitado al uso de diferentes tipos de software y tecnologías de suma importancia en el mundo informático.

Toda Cuba y en específico la UCI está migrando hacia el software libre puesto que brinda una alternativa en el desarrollo de soluciones informáticas para SIG basadas en la Web. Por esto el objetivo del trabajo de diploma es desarrollar un SIG para la UCI usando una plataforma que posibilite disminuir el esfuerzo necesario para integrarlo con las demás aplicaciones web existentes en la universidad.

### **1.3 Análisis de otras soluciones existentes**

Actualmente existen muchas plataformas SIG las cuales permiten el trabajo directo con los datos espaciales. Con su uso la información geográfica puede ser consultada, transferida, transformada, superpuesta, procesada y mostrada a los usuarios de forma correcta. A continuación se muestran algunas aplicaciones SIG muy utilizadas en estos días.

#### **1.3.1 GvSIG**

Surge como parte de un proyecto de migración a tecnologías abiertas gvPOINTS que se pone en marcha por la Conserjería de Infraestructuras y Transportes de la Generalidad de la Comunidad Valenciana.

Actualmente posee una mejora de la documentación técnica para desarrolladores y un sistema de control de versiones público donde los usuarios puedan obtener el código fuente actualizado del producto.

Posiciona como un potente visor (2D y 3D) de cartografía tanto local como remota (destacando en el soporte a estándares *Open Geospatial Consortium* (OGC)) así como una herramienta de publicación de cartografía tanto en papel como en servidores de mapas, y cuenta con gran variedad de herramientas de

análisis tanto vectorial como ráster. Además cuenta con funcionalidades de análisis aportadas por SEXTANTE. Es una herramienta open-source (código abierto) orientada al manejo de información geográfica.

El proyecto además del producto principal para escritorio, ha lanzado una versión para dispositivos móviles llamada GvSIG Mobile que permite realizar trabajos relacionados con la visualización y edición de cartografía en campo utilizando ordenadores de mano. (Sanz Salinas, 2009)

### **Arquitectura GvSIG**

La aplicación gvSIG está construida por capas, las cuales están relacionadas por mecanismos llamados extensiones. Esto permite a los desarrolladores añadir funcionalidades a la base de gvSIG.

La plataforma está conformada en su núcleo por tres subsistemas:

*Andami:* Es el esqueleto donde se sustenta gvSIG. Es como un soporte donde se van ensamblando todas las extensiones que conformarán el sistema. Es capaz de presentar un entorno de usuario MDI (Interfaz de múltiples documentos) y de incluir aspectos importantes en la ventana de la aplicación.

*Fmap:* Incluye todas las clases para manejar objetos SIG, drivers y adaptadores para facilitar el trabajo con los formatos más usados para el almacenamiento de datos cartográficos. Es en esta librería donde se encuentran las clases para leer y escribir los formatos soportados, asignar leyendas, realizar consultar, dibujar mapas y realizar búsquedas.

*gvSIG Extensión:* Contiene la parte de interfaz de usuario que representa los datos geográficos manejados por Fmap. Es en esta librería donde se encuentran los formularios para asignar leyendas, crear mapas y vistas, definir escalas, etc.

En la Figura 2 se representa la arquitectura de la plataforma.



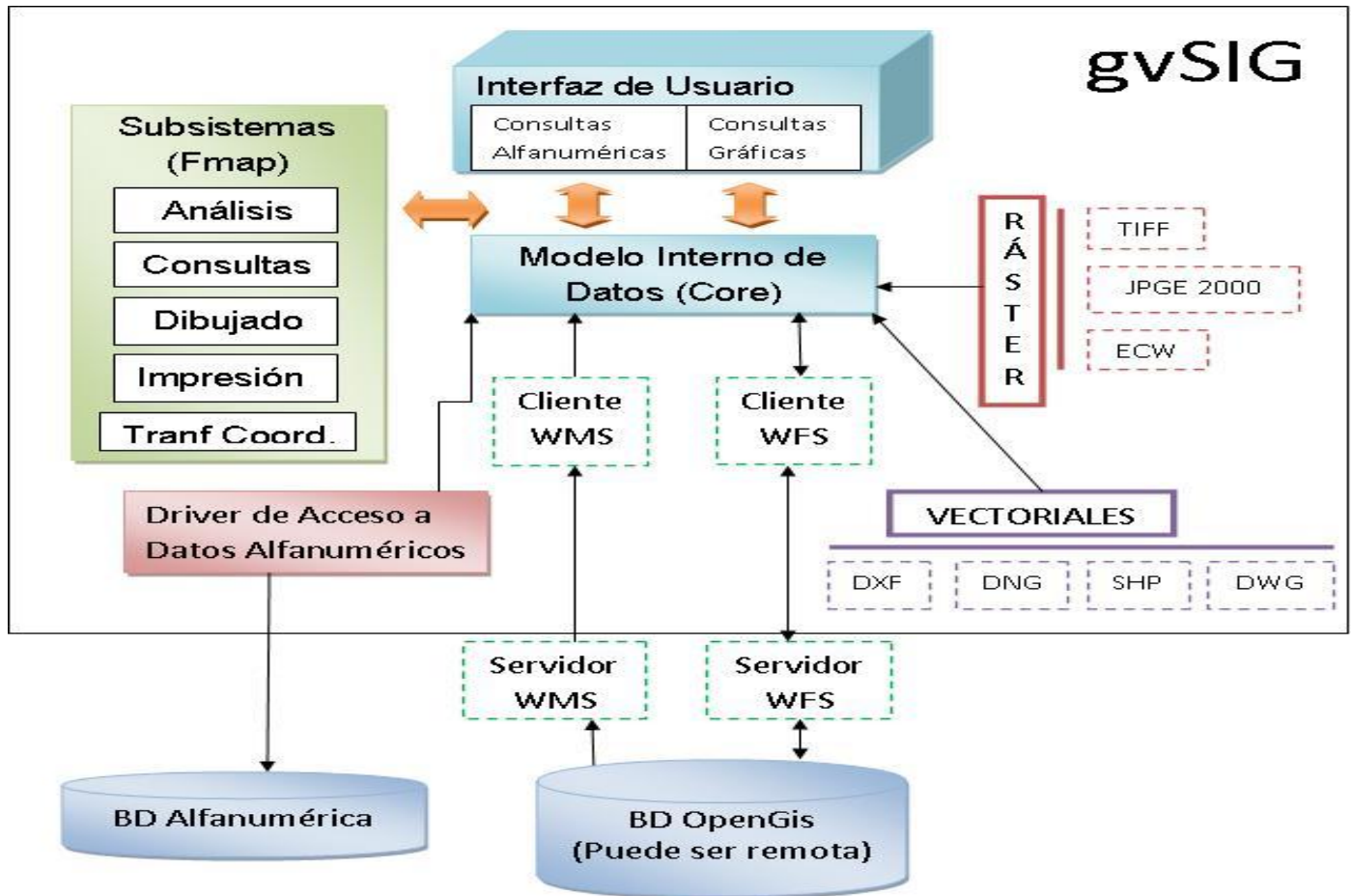


Figura 2 Arquitectura de GvSIG<sup>4</sup>

### 1.3.2 ArcGIS

Es una familia de productos de software, producidos y comercializados por ESRI, que forman un SIG completo construido sobre los estándares industriales que proveen capacidades excepcionales y aún más fáciles de usar, justo fuera de la caja. ArcGIS es un completo y sencillo sistema integrado de creación, administración y análisis de datos geográficos.

Está diseñado como un sistema escalable que puede ser desplegado en cada organización, desde un escritorio individual hasta una red de personas distribuidas globalmente.

<sup>4</sup> Tomado de TodoBI [En Línea] Noviembre 21, 2005. <http://todobi.blogspot.com/2005/11/gvsig-el-gis-open-source-espaol.html>

ArcGIS Desktop se distribuye comercialmente bajo tres niveles de licencias que son, en orden creciente de funcionalidades: ArcView, ArcEditor y ArcInfo. (IDAS, 2008)

Además, las aplicaciones de ArgGIS se engloban en familias temáticas como ArcGIS Desktop, ArcGIS Server para la gestión y publicación web y ArcGIS Móvil para capturar y gestionar información (Ver Figura 3).

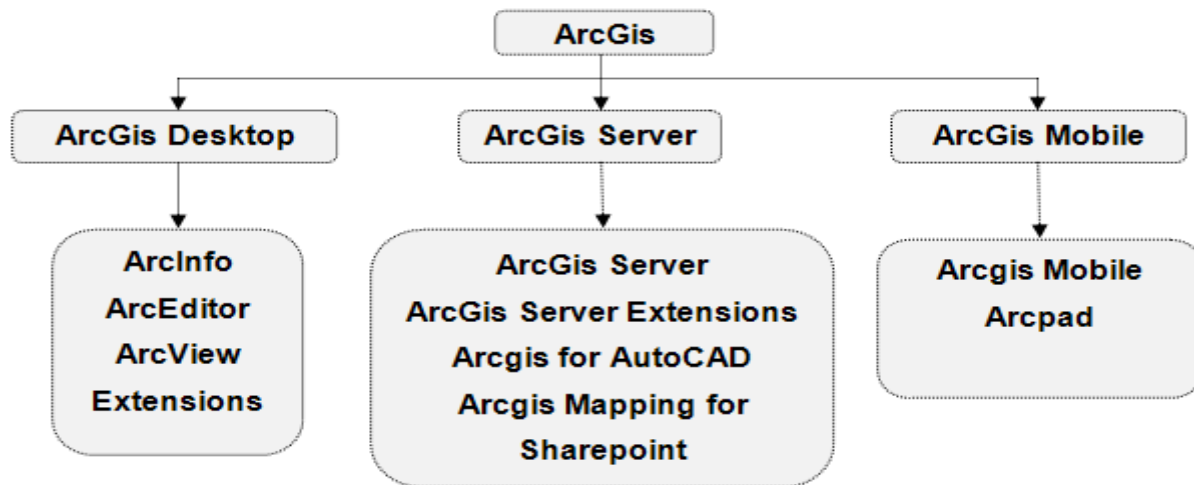


Figura 3 ArcGIS

ArcGIS Desktop como su nombre lo indica se ejecuta en un ambiente de escritorio. Es utilizado principalmente para crear, preguntar, trazar, importar, revisar, analizar, y publicar los datos geográficos. Por lo general, todas las aplicaciones ArcGIS Desktop poseen la misma arquitectura para facilitar el trabajo de los usuarios que las utilizan.

Por otra parte integra varias herramientas como: ArcMap, ArcCatalogTM, ArcToolboxTM y ArcGlobeTM, cada una de ellas con un sistema rico de funcionalidades y también puede ampliar sus extensiones opcionales para agregar capacidades especiales.

Este es un SIG que posee gran cantidad de funcionalidades implementadas que permiten a los usuarios desarrollar otras que les sean necesarias. Además, presenta un manejo de información geográfica potente

y con bastante exactitud respecto al plano real. Por todo ello es considerado actualmente el líder de los SIG a nivel mundial.

### Arquitectura ArcGis

ArcGIS posee una arquitectura que incluye una completa plataforma para el desarrollo de funcionalidad SIG tanto en el servidor (ArcGIS Server), como a nivel desktop (AerGIS Desktop). Soporta varios lenguajes de programación como Java, Visual Basic 6, Phyton, etc. A continuación en la Figura 4 como está estructurada la plataforma.

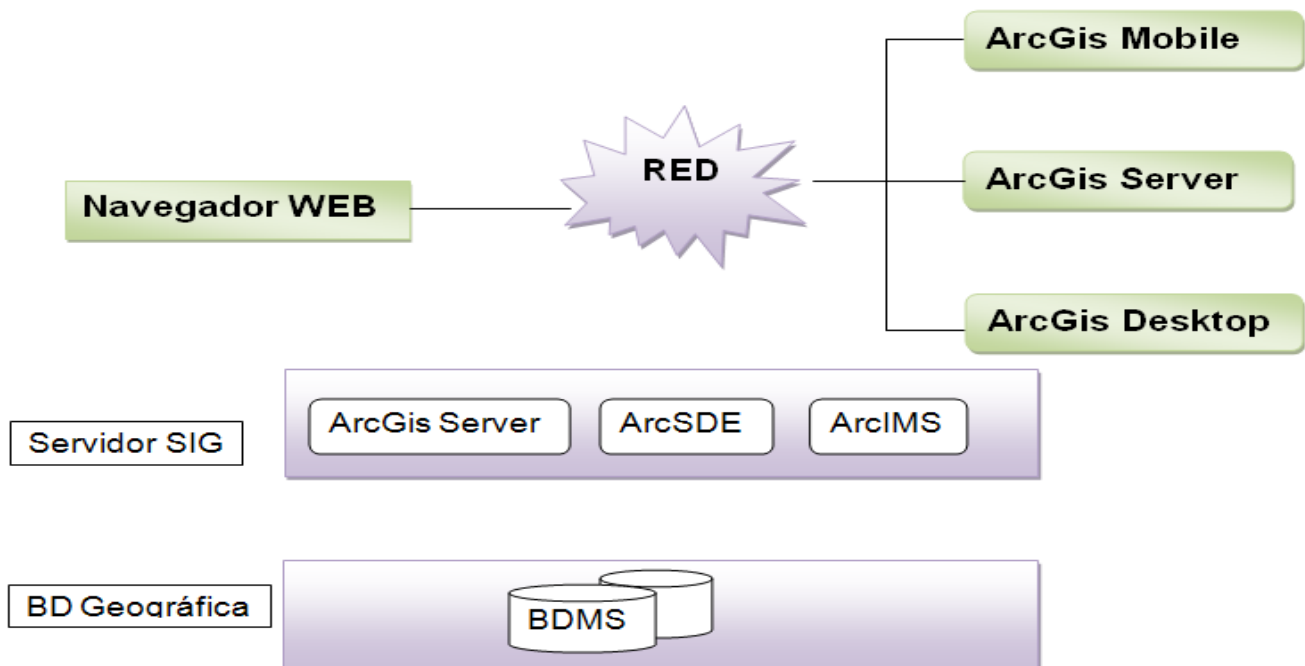


Figura 4 Arquitectura ArcGIS

### 1.3.3 Geomedia

Es una tecnología SIG de nueva generación creada en el año 1996 por Intergraph. Geomedia WebMap (versión actual del producto: 5.2) permite acceder a todos los datos de la empresa a través de Internet o Intranet, desde un entorno de Microsoft Windows. Incorpora toda la tecnología necesaria para construir aplicaciones que combinen, consulten, analicen y distribuyan información almacenada en un SIG por la

red. Posibilita compartir la información geográfica de la empresa mediante un simple explorador Web, de forma barata y sencilla. (Soluciones Gráficas S.A, 2005)

Geomedia se ejecuta bajo un ambiente Windows y permite el almacenamiento de información gráfica en bases de datos. A partir de Geomedia surge Geomedia Profesional la cual provee todas las funcionalidades de Geomedia e incluye herramientas para la edición de datos espaciales. Además, permite incrementar la productividad para modificar datos y acelerar la implementación de la base de datos SIG. Con esto último se pueden realizar conexiones en tiempo real a múltiples almacenes de datos SIG simultáneamente.

### Arquitectura de Geomedia

Presenta una arquitectura tres capas dividiendo la aplicación en Presentación, Lógica del Negocio y Datos (Ver figura 5) y es fácil de personalizar con herramientas estándares del mercado.



Figura 5 Arquitectura de Geomedia

### 1.3.4 MapFish

MapFish es un marco flexible y completo para la creación de aplicaciones Web con gran cantidad de mapas. Consta de dos partes: MapFish Cliente y MapFish Servidor. MapFish cliente es un JavaScript framework basado en OpenLayers para el cartografiado de mapas y en ExtJS para la parte GUI (widgets). MapFish Servidor permite llevar la composición y gestión de varios módulos que pueden ser implementados en lenguajes de programación tales como Python, Java, PHP. Ofrece una manera simple de configuración de parámetros para tener una herramienta de trabajo SIG-WEB que consume servicios Web Map Server (WMS) y Web Feature Server (WFS). No obstante MapFish puede también ser simplemente incluido en un sitio web ya existente, como un Sistema de gestión de contenidos (CMS).

#### ***MapFish del lado del cliente contiene:***

1. El panel de mapa para la representación de la información geográfica a través de servicios web de *Open Geospatial Consortium (OGC)* como WMS y WFS.
2. La barra de herramientas para el acceso a las funciones de la GUI como The layer tree para la organización y la gestión de las capas geoespacial.
3. El control de impresión para la creación de informes.
4. Las funciones de edición para la creación y actualización de conjuntos de datos.
5. La búsqueda y el control más reciente para la navegación avanzada en los datos.
6. El control de consulta para obtener información sobre los datos representados.
7. El sistema de plantillas para la definición de la interfaz gráfica de usuario reutilizables.

#### ***MapFish del lado del servidor permite:***

1. La creación de servicios RESTful (REpresentational State Transfer ful) para el acceso a datos y manipulación de datos geográficos (crear, leer, actualizar y eliminar objetos geográficos).
2. El uso de SQLAlchemy y Shapely es resaltado.

3. La configuración del componente de servidor de impresión para la definición de los informes.

**MapFish dispone de:**

1. Un tipo de geometría que utiliza el mapa de las tablas con PostGIS SQLAlchemy.
2. Un comando para generar el modelo y el modo de control correspondientes a las capas (tablas PostGIS) definida en un archivo de configuración.

Este sistema de información geográfica es de gran utilidad debido a la facilidad de integración con el resto de las aplicaciones web existentes. Sus características lo hacen exactamente funcional para el desarrollo del proyecto pues, entre otras características:

- Imprime los mapas en formato PDF, en ambos formatos, vector y ráster.
- Permite manejo del mapa utilizando ajax.
- Es multiplataforma (Windows, Linux, MacOS).
- Fácil de instalar en el servidor sin necesidad de instalar muchas dependencias.

**Arquitectura de la plataforma:**

Del lado del servidor, los datos pueden ser administrados dentro de la base de datos o sobre ficheros. Desde esos datos se puede utilizar un generador *tile* para almacenar en la cache el contenido geográfico con el objetivo de mejorar el rendimiento de la aplicación. El proceso de generar el *tile* se encuentra basado sobre un servidor cartográfico que es capaz de brindar el servicio de los *tiles* o sobre los ficheros (imágenes por ejemplo). El servidor de aplicaciones de MapFish provee servicios como buscar, imprimir o editar. Pueden realizarse complejos y personalizados análisis para el desarrollo de soluciones haciendo uso de la tecnología que provee este marco de trabajo. (Ver Figura 6)

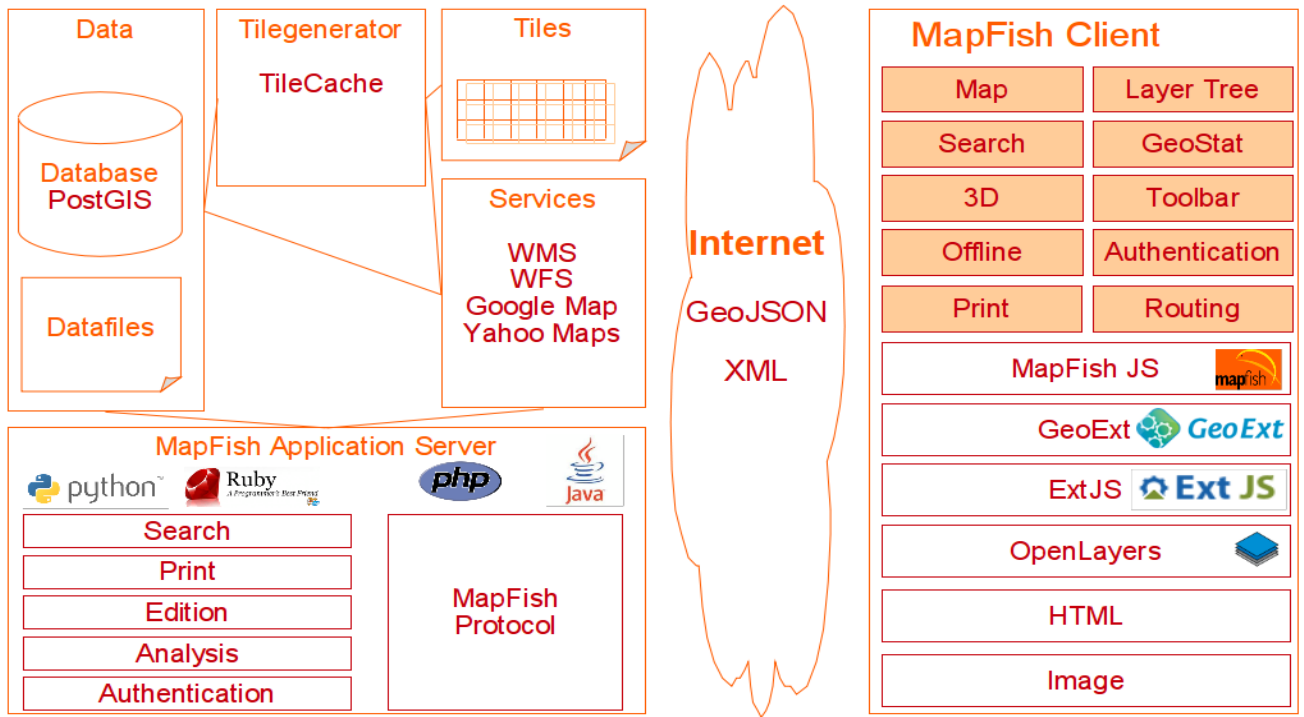


Figura 6 Arquitectura de MapFish

MapFish además está desarrollado sobre una plataforma libre, la cual permite disminuir el esfuerzo que implica integrar los SIG con otras aplicaciones web. Es por todas estas características que se decidió utilizar dicho framework como la herramienta para el desarrollo de la propuesta de solución.

## 1.4 Metodologías de desarrollo de Software

Siempre que se construye una solución informática es de vital importancia la selección de la metodología de software a utilizar. Esta es la encargada de especificar quién realiza una tarea, cuáles son las tareas que se realizan, qué documentación se generará, etc. Su correcta elección origina un producto de software que satisface las necesidades de los clientes.

### 1.4.1 Extreme Programming (XP)

Conocido en español como Programación Extrema, se encuentra dentro de las metodologías ágiles o ligeras. Está centrada principalmente en las relaciones interpersonales como clave para el éxito del software. Se enfoca en el trabajo en equipo donde clientes y desarrolladores trabajan de la mano persiguiendo un único objetivo, el desarrollo de un software con calidad. Además, existe una retroalimentación continua entre ellos.

XP es utilizada en proyectos cortos donde los requisitos son inestables e imprecisos y las soluciones son poco complejas. El equipo de trabajo suele ser pequeño y existen pocos roles. Está enfocado en tres características principales:

Comunicación: Los desarrolladores se comunican continuamente con los clientes y con el resto de los desarrolladores.

Retroalimentación: Se realizan pruebas desde los primeros días de desarrollo y el producto es entregado a sus clientes tan pronto como sea posible.

Simplicidad: El diseño del software es tan simple como sea posible. No se realiza mucho énfasis en la arquitectura.

### 1.4.2 Rational Unified Process (RUP)

El Proceso Unificado de Software es una metodología tradicional o pesada. Es utilizada principalmente en proyectos grandes y con requisitos poco variables. Tiene como objetivo lograr un producto de máxima calidad que cumpla con las necesidades planteadas por el usuario en un tiempo y con un presupuesto acordado con anterioridad. Realiza un modelado visual del software y permite gestionar una potente documentación y control de cambios.

RUP es una metodología adaptable a las necesidades de la empresa. Los roles están bien definidos y en caso de ser un proyecto muy grande participan varios equipos con una comunicación bien fluida. El proceso de desarrollo está dividido en cuatro fases dentro de las cuales se van realizando varias iteraciones según el proyecto y la importancia de cada actividad que se realiza.



Posee tres características esenciales que lo distinguen de las restantes metodologías y lo hacen único:

Dirigido por casos de uso: Los casos de uso son el claro reflejo de lo que el cliente planteó que deseaba en la modelación del negocio a través de los requerimientos. A partir de que aparecen los casos de usos, estos pasan a guiar el proceso de desarrollo puesto que los modelos que se obtienen como resultado de la realización de los flujos de trabajo están representados en los casos de uso.

Centrado en la arquitectura: La arquitectura muestra una visión común del sistema completo. RUP se desarrolla mediante iteraciones, dando prioridad a los casos de uso arquitectónicamente significativos.

Iterativo e incremental: Todas las fases se desarrollan mediante iteraciones. Una iteración incluye actividades de todos los flujos de trabajo, las cuales va refinando en cada iteración.

### **1.4.3 Fundamentación de la metodología seleccionada (RUP)**

El pasar de los años ha demostrado la necesidad de la utilización de una metodología de software para que el producto tenga una mayor calidad, exista más organización en el trabajo y el sistema sea puesto en práctica en el tiempo acordado.

Luego de analizar lo que se propone en el trabajo de diploma y viendo los posibles procesos a automatizar se escoge la metodología RUP. Es efectivo para un proyecto de gran magnitud y adaptable a las necesidades del equipo de desarrollo. Además, genera una documentación de gran utilidad para la continuidad del proyecto.

## **1.5 Lenguaje de Modelación**

La modelación es de suma importancia no solo en grandes proyectos sino también en medianos y pequeños proyectos. Esto se debe principalmente a que con el modelado se logra una notación que es de entendimiento para todo el equipo de desarrollo sin importar el rol que desempeñe.

### **1.5.1 Unified Modeling Language (UML)**

El lenguaje unificado de modelación (UML por sus siglas en inglés) es un lenguaje que permite visualizar, especificar, construir y documentar todo un software. Las combinaciones de sus elementos forman

diagramas. Este lenguaje permite la modelación de sistemas con tecnología orientada a objetos y su uso ayuda al equipo a comunicarse y a explorar diseños.

UML está compuesto por elementos, relaciones y diagramas y su uso posibilita el establecimiento de requerimientos y estructuras necesarias en un sistema de software antes de escribir el código. Es importante tener en cuenta que cuanto más complejo es el sistema que se quiere realizar, más importante es el uso de UML.

## **1.6 Lenguaje de desarrollo Web - Hypertext Preprocessor (PHP)**

PHP es un lenguaje de programación, interpretado y de alto nivel del lado del servidor, incrustado en lenguaje HTML. Con su utilización se pueden leer y escribir archivos, crear imágenes, realizar consultas a base de datos y comunicarse con servidores remotos. Dispone de librerías que facilitan el desarrollo de aplicaciones y es compatible con varios servidores web y gestores de base de datos. Es un lenguaje sencillo, cómodo a la hora de usarlo y trabajar con su sintaxis. A pesar de ser interpretado es bastante rápido y además es multiplataforma. PHP está basado en herramientas con licencia de software libre por lo que no hay que pagar licencias.

## **1.7 Sistemas Gestores de Base de Datos**

Un Sistema de Gestión de Bases de Datos (SGBD) es un conjunto de programas, procedimientos y lenguajes que facilitan las herramientas necesarias para el trabajo con una base de datos. Actualmente existe una gran competencia a nivel mundial, muchos compiten por ser la presentación más rápida, segura, confiable y robusta. Entre los más usados se encuentran el PostgreSQL, MySQL, Microsoft SQL Server y Oracle.

### **1.7.1 PostgreSQL**

PostgreSQL es un sistema gestor de base de datos relacional. Está basado en el proyecto Postgres y fue uno de los primeros que utilizó los conceptos existentes actualmente en el sistema objeto- relacional.

Incluye características de la orientación a objetos como la herencia, tipos de datos, funciones, restricciones, disparadores, reglas e integridad transaccional. Es capaz de realizar consultas complejas, manejo de visitas, control concurrente multiversión, entre otros.

Tiene un soporte completo para subconsultas. Funciona en los principales y más usados sistemas operativos. Es compatible con el almacenamiento de objetos binarios, incluyendo imágenes, sonidos o vídeos.

Entre sus principales características se encuentran las siguientes:

- Está implementado sobre el estándar SQL92/SQL99.
- Soporta varios tipos de datos como: tipo fecha, monetarios, elementos gráficos, datos sobre redes (MAC, IP...), cadenas de bits, permite la creación de tipos propios y además soporta los tipos base.
- Incorpora funciones de diversas índoles: manejo de fechas, geométricas, orientadas a operaciones con redes, etc.
- Soporta el uso de índices, reglas y vistas.
- Incluye herencia entre tablas (aunque no entre objetos, ya que no existen), por lo que a este gestor de bases de datos se le incluye entre los gestores objeto-relacionales.
- Permite la gestión de diferentes usuarios, como también los permisos asignados a cada uno de ellos.

PostgreSQL tiene varios módulos dentro de los cuales se encuentra PostGIS el cual es de vital importancia para la realización de bases de datos espaciales. Las bases de datos espaciales contienen además de los datos propios de cualquier base de datos como: texto y número, información referente a la localización espacial de objetos geográficos. Estas contienen elementos que tienen forma y ubicación en el espacio ya sea bidimensional o tridimensional.

## **PostGIS**

Es el módulo espacial de PostgreSQL que permite la manipulación y almacenamiento de datos espaciales. Además, es capaz de implementar metadatos y funciones geométricas y topológicas para el trabajo con datos espaciales basados en el estándar del *OpenGis Consortium*. Es open-source, es decir, código abierto, soporta funciones básicas para el trabajo con objetos que contienen información geográfica.

Es capaz de tratar grandes volúmenes de datos con escalabilidad. Puede usarse, con adaptaciones, en cualquier plataforma. Entre sus funciones podemos encontrar:

- Analizar si dos geometrías cuentan con un punto en común (Disjoint ()).
- Analizar si dos geometrías poseen alguna intersección (Intersects ()).
- Analizar si dos geometrías se cruzan (Crosses ()).
- Analizar si una geometría contiene a la otra (Contains ()).
- Mostrar una geometría en su representación textual (AsText ()).
- Devuelve la diferencia entre dos geometrías dadas (Difference ()).

## **1.8 Herramientas propuestas para el desarrollo de aplicaciones**

Cuando se desarrolla una aplicación es imprescindible la selección de las herramientas que posibilitarán la construcción adecuada del sistema. Es importante tener en cuenta que mientras mejor sea la elección realizada mayor será la rapidez y la calidad que poseerá la solución informática.

### **1.8.1 Visual Paradigm**

Visual Paradigm es una herramienta CASE (Ingeniería de Software Asistida por Ordenadores o CASE por sus acrónimos en inglés) que da soporte al modelado UML. Permite generar artefactos de gran importancia para la creación de un software dando la posibilidad de documentar todo el producto sin la

necesidad de utilizar una herramienta externa. Es fácil de usar y está disponible en varios idiomas. Es compatible con varias plataformas como Microsoft Windows (98, 2000, XP, o Vista), Linux, MacOS X, Solaris o Java. Reduce el tiempo de desarrollo y mantenimiento del producto y se puede utilizar para realizar diferentes aspectos en todo el ciclo de vida del software.

### **1.8.2 PgAdmin III**

Es una aplicación gráfica con licencia de código abierto que posibilita gestionar el SGBD PostgreSQL. Se encuentra disponible en más de una docena de lenguajes. Su interfaz gráfica soporta las características de PostgreSQL y provee su administración. No solo es capaz de responder a consultas sencillas de los usuarios sino además puede desarrollar complejas bases de datos. Posee herramientas de consultas SQL, editor de código procedural y agente de planificación SQL y puede ser ejecutado bajo el ambiente de Linux, Windows, Mac OS, entre otros.

## **1.9 Conclusiones**

En este capítulo se realizó un análisis de los principales conceptos asociados al dominio del problema y otros que permiten un total entendimiento del Proceso de referenciar la información geográficamente (objeto de estudio). Además, se comentó más profundamente la situación problemática que originó la investigación. Se analizaron varias plataformas para el desarrollo de Sistemas de Información Geográfica y se seleccionó MapFish por ser la que más cumplía con las necesidades existentes.

Se analizaron las tendencias tecnológicas actuales y se concluyó que la solución hará uso de la metodología RUP para su desarrollo, se utilizará el lenguaje de modelado UML, Visual Paradigm como herramienta CASE, PHP como lenguaje de programación y PostgreSQL como gestor de base de datos con su módulo espacial PostGIS.

Considerando las tendencias actuales y las políticas de migración hacia el software libre del país, se puede concluir que las herramientas seleccionadas son las óptimas para la realización de la solución que se propone en el trabajo. Además una vez realizado el capítulo ya están creadas las bases para comenzar el desarrollar la solución propuesta.

## **CAPÍTULO 2: “Presentación de la solución propuesta.”**

Es este capítulo se realizará un análisis de la solución propuesta. Al no ser identificados claramente los procesos de negocio se plantea la conceptualización del entorno mediante un modelo de dominio. Se analizan y relacionan los conceptos y entidades que están presentes donde funcionará el sistema. Se identifican los requisitos funcionales y no funcionales con los que contará el sistema. Se analizan y detallan los casos de uso que tendrá el sistema y se describen los actores. Además, se realiza el análisis y el diseño del sistema y de la base de datos.

### **2.1 Modelo de Dominio**

RUP considera el modelo de dominio como un subconjunto del modelo de negocio. El modelo de dominio se realiza cuando los procesos del negocio no están claros debido a que no se conocen sus orígenes o simplemente son sucesos o eventos. Además, no es posible identificar los trabajadores del negocio debido a que existe una sobrecarga de responsabilidades y es muy difícil establecer las reglas de funcionamiento del sistema a implementar.

El modelo de dominio no es más que la representación visual de los objetos y conceptos de la entidad donde se realizará el sistema. Cuando se utiliza este modelo se capturan los objetos y eventos más importantes y no importa quién es el responsable de realizar las actividades debido a que los procesos no están bien definidos.

Se representa usando el del lenguaje de modelado UML, a través de un Diagrama de Clases donde se muestran conceptos u objetos del dominio del problema (clases conceptuales), sus relaciones y atributos.

#### **2.1.1 Conceptos y eventos principales del entorno**

Cartógrafo: Persona cuya profesión está basada en la realización y el estudio de mapas. Es el encargado de elaborar los mapas que luego se utilizan en la entidad donde se implementará el sistema.

Personal UCI: Estudiantes, profesores y trabajadores de la UCI que necesiten trabajar o consultar algún tipo de información incluida en un mapa.

UCI: Entidad que solicita un servicio determinado utilizando un mapa y proporcionando la información

socioeconómica referente al mismo.

Mapa: Es una representación gráfica de una porción territorial sobre una superficie que generalmente es plana. Se clasifican por su finalidad y por su extensión.

Información socioeconómica: Datos referentes al aspecto social y económico de un territorio determinado.

Escala: Es la relación que existe entre la distancia de dos puntos en un mapa y la distancia de esos dos puntos en la superficie terrestre. Cuanto mayor sea la escala de en un mapa, mayor será la aproximación que tiene este en correspondencia con la superficie terrestre que representa.

Leyenda: Es la explicación de la simbología utilizada en los mapas. Suele encontrarse en la parte inferior de los mapas. Es de vital importancia puesto que un mapa sin una leyenda es casi imposible de entender.

Tipo de mapa: Es la clasificación que se le da a los mapas.

En la Figura 7 se muestra el **Diagrama de Clases del Dominio** el cual representa los conceptos y eventos principales capturados en la entidad.

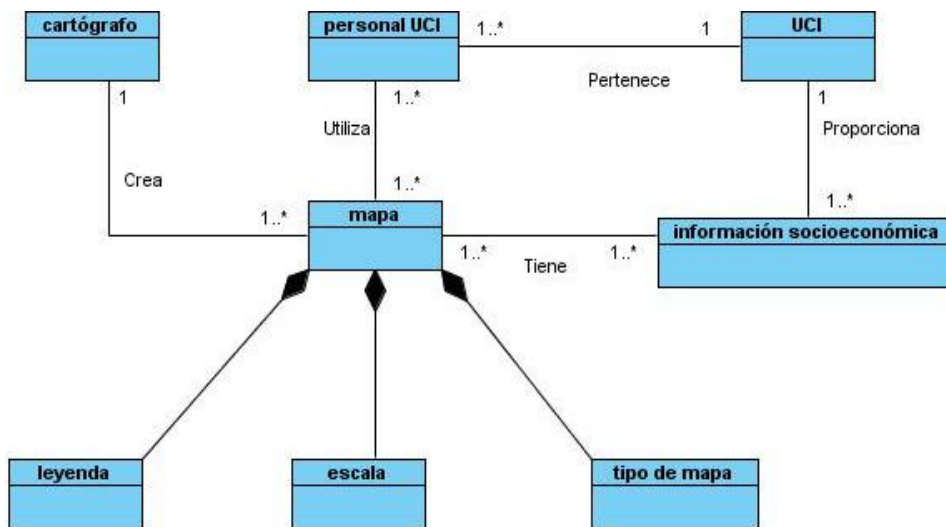


Figura 7 Diagrama de clases Del Modelo de dominio

## 2.2 Requerimientos Funcionales

De acuerdo con los objetivos propuestos en el trabajo, el sistema debe ser capaz de:

**RF1.** Mostrar el mapa de La Universidad.

**RF1.1.** Permitir la selección de las capas que desean mostrarse.

**RF1.2.** Mostrar un mapa más pequeño de forma que se vea completo donde se señale la porción del mapa grande que está siendo visualizada por el usuario.

**RF2.** Realizar paneos a los mapas (grande y chiquito).

**RF3.** Realizar los efectos de aumentar y disminuir la escala del mapa.

**RF4.** Realizar búsquedas y señalar en el mapa la ubicación de los objetos de dichas búsquedas.

**RF4.1.** Dado un usuario o un número de solapín señalar en el mapa la ubicación del edificio de la persona.

**RF4.1.1.** Mostrar el nombre, número de apto y número de edificio de la persona buscada.

**RF4.2.** Dado un número de edificio señalar en el mapa la ubicación del mismo.

**RF5.** Mostrar la información de cualquiera de los edificios que se muestren en el mapa.

**RF6.** Realizar mediciones de distancias en el mapa.

**RF7.** Realizar cálculos de áreas en el mapa.

**RF8.** Visualizar las coordenadas del mapa.

**RF9.** Efectuar, dado dos puntos, el cálculo de la ruta mínima entre esos dos puntos y mostrarlo sobre el mapa.



## **2.3 Requerimientos No Funcionales**

### **2.3.1 Usabilidad**

El sistema se podrá usar por personas con conocimientos de computación básicos. Se emplearán componentes que indiquen al usuario el estado de los procesos que por su complejidad requieran de un tiempo de procesamiento apreciable.

### **2.3.2 Fiabilidad**

Se tendrá en cuenta la recuperación ante fallos y errores.

### **2.3.3 Eficiencia**

El tiempo de respuesta dependerá de la cantidad de información a procesar, entre mayor cantidad de información mayor será el tiempo de procesamiento. Al igual que el tiempo de respuesta, la velocidad de procesamiento de la información, la actualización y la recuperación dependerán de la cantidad de información que tenga que procesar la aplicación.

### **2.3.4 Soporte**

La aplicación recibirá mantenimiento en el período de tiempo determinado por el equipo de desarrollo y los clientes.

### **2.3.5 Restricciones de diseño**

Poseerá un diseño sencillo donde no sea necesario mucho entrenamiento para utilizar el sistema. El producto de software final debe diseñarse sobre una arquitectura cliente-servidor.

Se debe lograr un producto altamente configurable y extensible, teniendo en cuenta que se desarrollará sobre el framework MapFish y que constituye una plataforma de desarrollo para ser personalizada como aplicaciones a la medida, pudiéndose incorporar a ésta nuevas funcionalidades.

## 2.3.6 Interfaz

### Interfaces de usuario

El sistema debe:

- Tener una apariencia profesional y un diseño gráfico sencillo.
- Ser intuitivo.

### Interfaces de hardware

Para las PCs clientes:

- Se requiere tengan tarjeta de red.
- Al menos 128 MB de memoria RAM.
- Se requiere al menos 100 MB de disco duro.
- Procesador 512 MHz como mínimo.

Para los servidores:

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

### Interfaces de software

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

Para las PCs clientes:

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

- Sistema operativo: GNU/Linux, Windows y Mac OS.

Para los Servidores:

- Sistemas operativos GNU/Linux o Windows Server 2000 o superior.
- Servidor Web Apache 2.0 o superior, con módulo PHP 5 configurado con la extensión pgsql incluida.
- PostgreSQL como Sistema Gestor de Base de Datos.
- PostGIS como extensión de PostgreSQL como soporte de datos espaciales.
- MapServer 5.2.2 o superior, con extensión PHP mapscript.

#### Interfaces de comunicación

El producto garantizará mediante su interfaz la configuración del entorno de trabajo mediante funcionalidades propias como ocultar y mostrar paneles, así como elementos para cambiar las vistas y las capas que serán visibles en la interacción.

### **2.3.7 Requisitos de Licencia**

De acuerdo con los tipos de licencias de los componentes y herramientas que se proponen a utilizar para el desarrollo del producto se puede catalogar legalmente esta arquitectura de modelo libre, permitiendo la utilización, modificación y distribución de las mismas por terceros sin necesidad de obtener la autorización de sus respectivos titulares.

### **2.3.8 Requisitos Legales, de Derecho de Autor y otros**

El sistema debe ajustarse y regirse por la ley, decretos leyes, resoluciones y manuales (órdenes) establecidos, que norman los procesos que serán automatizados.

La mayoría de las herramientas de desarrollo son libres y del resto, las licencias están avaladas.

Como producto se distribuye amparado bajo las normativas legales establecidas en el registro comercial emitido por las entidades jurídicas de la Universidad de las Ciencias Informáticas.

## 2.4 Descripción de la Solución Propuesta

Una vez realizado el levantamiento de requisitos en la entidad se identificaron las condiciones que el sistema debe tener y debe cumplir. A partir de los requisitos funcionales (condiciones que el sistema debe tener) se realizó el **Diagrama de Casos de Uso del Sistema** (Ver Figura 8) el cual muestra la relación que existe entre el usuario final de la aplicación y las funcionalidades que esta poseerá.

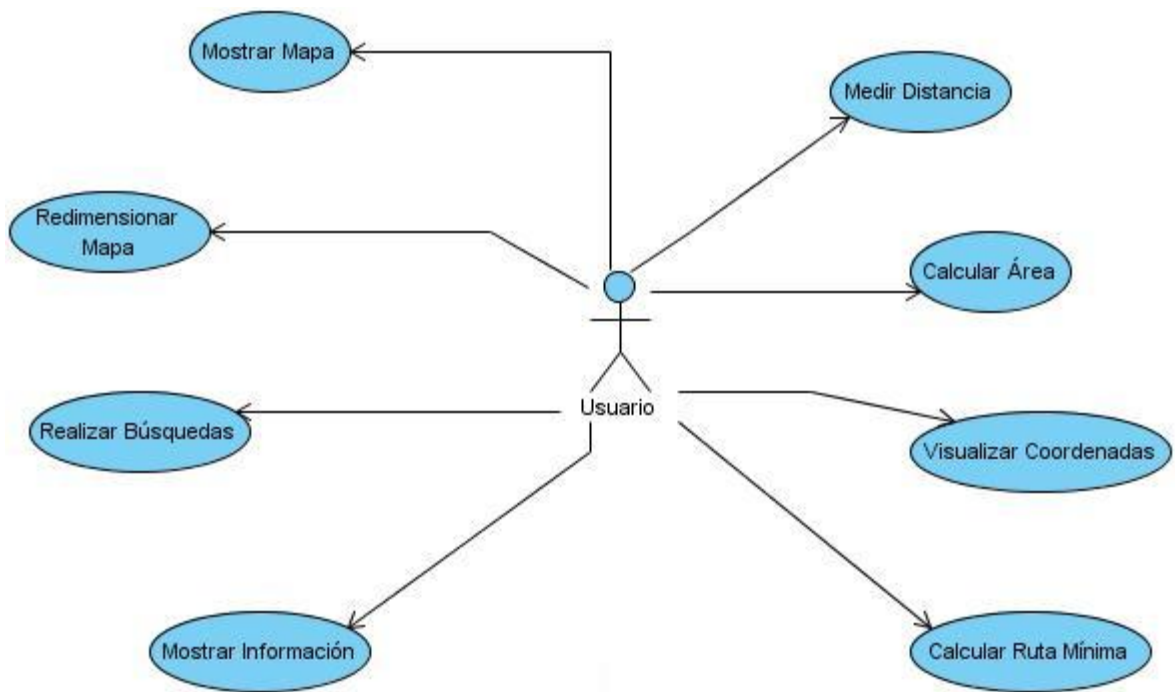


Figura 8 Diagrama de Caso de Uso del Sistema

### 2.4.1 Descripción de los actores del sistema

En la Tabla 2 se muestra una descripción del actor final del sistema, que en este caso es el personal UCI que tendrá acceso a la aplicación.

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

Tabla 2 Actores de Sistema

## 2.4.2 Descripción de los Casos de Uso del Sistema

En la Tabla 3 se muestra la descripción del CU Mostrar Mapa, donde se explican paso a paso las acciones del usuario y las respuestas del sistema. Para acceder a las descripciones de los restantes casos de uso ver Anexo #2.

### Mostrar Mapa

<b>Caso de Uso</b>	Mostrar Mapa	
<b>Actores</b>	Usuario	
<b>Propósito</b>	Este caso de uso tiene como objetivo mostrar el mapa UCI y cambiar las capas seleccionables y visibles de un mapa	
<b>Resumen</b>	El caso de uso se inicia cuando el actor desea mostrar el mapa o seleccionar alguna(s) capa(s) en dependencia de su interés y termina cuando el sistema muestra la capa base o modifica el estado de las capas visibles en el mapa.	
<b>Precondiciones</b>		
<b>Referencias</b>	RF 1, RF1.1, RF 1.2, RF 1.3	
<b>Prioridad</b>	Crítico	
<b>Flujo Normal de Eventos</b>		
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>	

1. El actor selecciona la o las capas que desea visualizar.	2. El sistema mantiene visible en la parte derecha todas las capas del mapa.
3. El actor selecciona las capas que serán visibles.	4. El sistema muestra la(s) capa(s) escogida(s) por el usuario.
<b>Flujos Alternos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
<b>Poscondiciones</b>	

Tabla 3 Descripción CU Mostrar Mapa

## 2.5 Modelo de Diseño

Según Ivar Jacobson, Grady Booch y Jame Rumbaugh en el libro “El Proceso Unificado del Software”, la fase de análisis en la metodología de desarrollo RUP es opcional. El equipo de desarrollo es el encargado de decidir si se va a realizar o no el análisis. Esta fase es la encargada de describir con más claridad y detalladamente los requisitos funcionales para una mejor comprensión a la hora de realizar el diseño de la aplicación y se utiliza en sistemas con grandes cantidades de casos de uso, donde estos últimos presentan una complejidad elevada.

La aplicación que se está desarrollando en el presente trabajo de diploma posee pocos casos de uso (8 específicamente) con un promedio medio de complejidad y un equipo de desarrollo pequeño (2 personas), el cual desempeñará todos los roles. Por lo tanto, se ha decidido no realizar la fase de análisis y desarrollar directamente el diseño de la aplicación.

El diseño del sistema sirve de base para los pasos siguientes de soporte e ingeniería del software. Sin un buen diseño o con la ausencia completa de este se obtiene como resultado un producto débil a cambios, inestable y muy difícil de probar.

Con la realización del diseño se logra aumentar la calidad del software, además de describir la realización física de los casos de uso. Posibilita representar el software de forma que se pueda evaluar en cuanto a calidad y es la única forma de convertir exactamente los requisitos de un cliente en un producto o sistema de software finalizado.

### **2.5.1 Diagramas de Clases del Diseño**

Los diagramas de clases del diseño muestran la relación que existe entre dichas clases en el sistema. Son diagramas que están considerados como uno de los pilares básicos en el modelado UML, son de estructura estática y muestran de forma abstracta cómo se implementará el sistema.

El presente sistema se ha diseñado con una arquitectura Modelo Vista Controlador (MVC). Esta arquitectura es muy usada actualmente en sistemas informáticos ya que reduce el esfuerzo necesario en la implementación. En la vista se define la interfaz del usuario, el controlador responde a eventos y modifica el modelo y la vista, y el modelo posee componentes encargados del acceso a datos. Entre sus principales ventajas se encuentra la posibilidad de realizar cambios en la vista, el controlador y el modelo de forma independiente.

La vista es la encargada de interactuar con el usuario. El controlador gestiona las entradas del usuario y el modelo se encarga del trabajo con los datos y las reglas del negocio.

El MVC se usa para el diseño de aplicaciones con sofisticadas interfaces. Ayuda a mejorar la reusabilidad del código y además hace que la decodificación en la vista impacte en menor escala en la lógica del negocio o de los datos.

A continuación en la Figura 9, se muestra el diagrama de clases del diseño del CU Mostrar Mapa. Para acceder a los restantes diagramas ver Anexo #3.

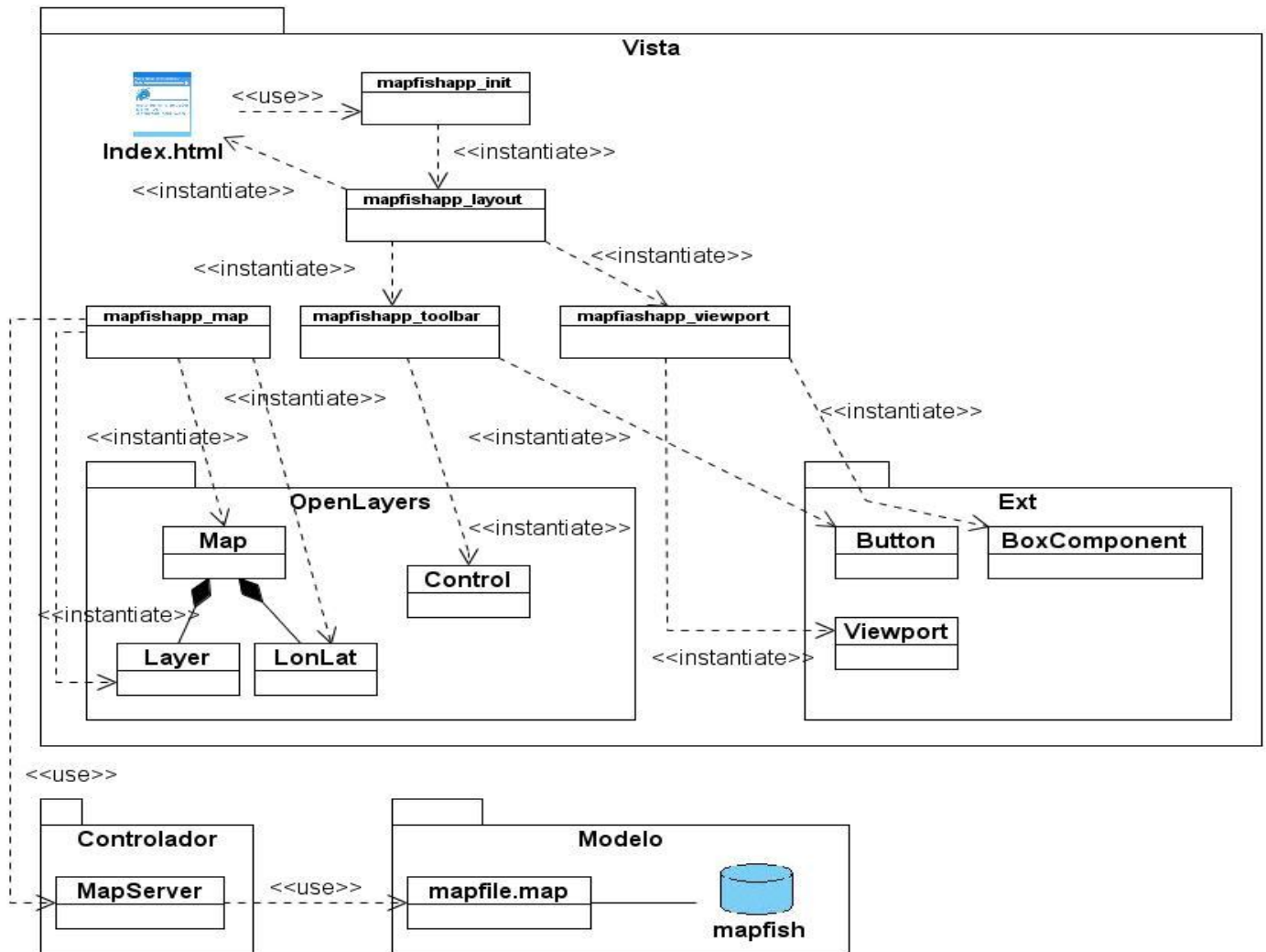


Figura 9 CU Mostrar Mapa

## 2.5.2 Estándares de la interfaz

El diseño de la interfaz es de gran importancia en el desarrollo de un sistema y requiere calidad debido a que es la parte que interactúa con el usuario. Debe ser amena, poco cargada de información y muy relacionada con el tema que se trata en el software. El éxito o el fracaso de un sistema pueden depender de la calidad de la interfaz y es por esto que se presta gran atención a la hora de su realización.



Para el diseño del sistema se utilizó la biblioteca Ext.js de la cual se utiliza su función ext.viewport para dividir la interfaz, permitiendo que esta se ajuste automáticamente al tamaño de la ventana del navegador que se esté usando. Además, se utilizó ext.toolbar para realizar la barra de herramientas y que el contenido esté organizado de una forma más agradable. Se utiliza una sola página principal y la introducción de la información que brinda el usuario se realiza a través de formularios construidos en Ext.js con la función ext.form.

La interfaz no está sobrecargada con elementos innecesarios. Los objetos con los que debe interactuar el usuario están claramente visibles. En la parte superior de la interfaz aparece el nombre del sistema para que el usuario sepa dónde está ubicado. Posee buen diseño gráfico posibilitando el intercambio con la aplicación de forma sencilla y rápida.

## 2.6 Modelo de Datos

Campos	Descripción
gid	Identificador de de los polígonos en la tabla.
número	Número del edificio.
tipo	Tipo de edificación.
the_geom	Geometría de cada polígono.
Id_servicio	Identificador de cada edificación en el catálogo de servicios.

Tabla 4 Ejemplo de la capa edificios en el modelo de datos

Un aspecto fundamental dentro de los sistemas SIG es la forma de almacenar la información. El mapa que se muestra en la aplicación está formado por capas, donde cada una de estas representa un área de la Universidad. Todas están guardadas en la Base de Datos y formadas por un identificador, un nombre y un objeto espacial. Cuando se quiere hacer referencia alguna capa almacenada en la Base de Datos se realiza mediante el identificador. Un ejemplo se muestra en la realización del CU Realizar Búsqueda, específicamente en el escenario “Buscar Persona”, donde el usuario introduce el número de solapín o el usuario UCI, luego mediante el servicio SOAP se accede a un Catálogo de Servicios y se obtienen sus datos además del identificador del edificio donde vive, después este se hace coincidir mediante una consulta con el identificador que tiene dicho edificio en la Base de Datos y se devuelven las coordenadas donde está ubicado en el mapa para poder posicionarlo.

## **2.7 Conclusiones**

En este capítulo se realizó un análisis del negocio existente y se concluyó que lo que se debía realizar era un modelo de dominio debido a que no estaban claros los procesos del negocio. Esto sirvió de entrada para identificar las condiciones que el sistema debía tener y debía cumplir y a partir. A partir de ahí se definieron los requisitos funcionales y no funcionales permitiendo realizar el diagrama de casos de uso del sistema para comprender las funcionalidades que la aplicación posee.

Se acordó no realizar la fase de análisis debido a que se está desarrollando un sistema con pocos casos de uso y no muy complejo. Se realizó el diseño de la aplicación el cual permite evaluar la solución informática en cuanto a calidad y se puntualizó la arquitectura a utilizar dando una visión común de lo que será el sistema.

## CAPÍTULO 3: “Implementación, Prueba y Factibilidad.”

Es este capítulo se describe la implementación de la solución propuesta siguiendo el diseño realizado. Además se validan las funcionalidades del sistema efectuando las pruebas del mismo y se muestra cómo se debe realizar el despliegue de la solución informática para que esta se ejecute correctamente.

### 3.1 Modelo de Despliegue

El modelo de despliegue describe la distribución física del sistema en término de cómo están distribuidas las funcionalidades en nodos de cómputo. Permite comprender y representar la relación entre la arquitectura de software y la arquitectura de hardware del sistema.

A continuación en la Figura 10, se muestra el diagrama de despliegue ideal para el sistema:

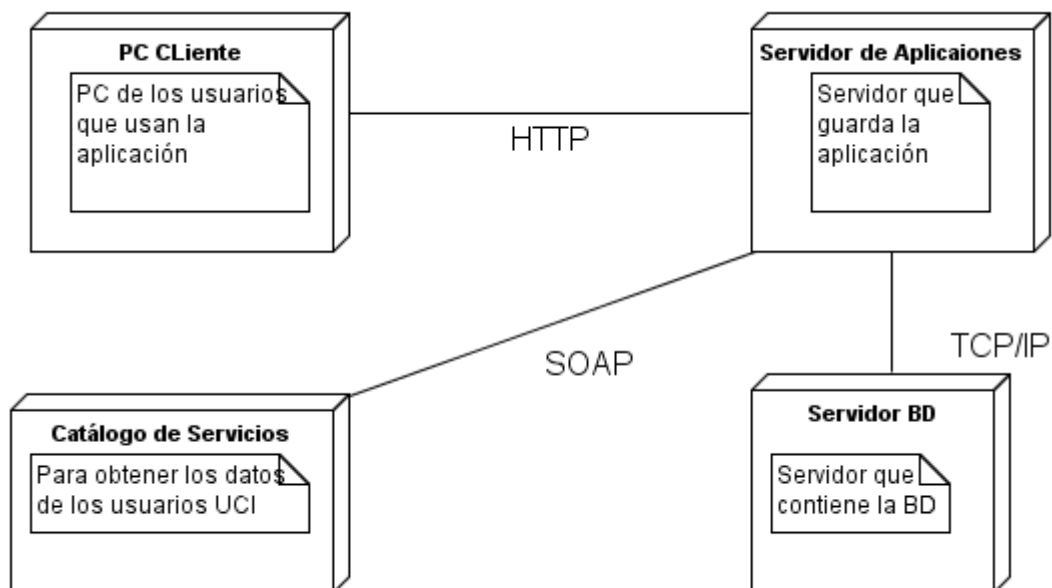


Figura 10 Diagrama de despliegue

## **3.2 Implementación. Modelo de Implementación.**

Este flujo de trabajo comienza con los resultados del diseño y describe cómo se implementarán las clases en términos de componentes, como ficheros de código fuente, ejecutables, etc. En el siguiente subepígrafe se especificará el diagrama de componentes relacionado a la aplicación que se está desarrollando.

### **3.2.1 Diagrama de componentes**

Los diagramas de componentes son usados para modelar los elementos que se utilizan en la fabricación de un sistema informático y las relaciones existentes entre dichos elementos de implementación. Es parte de la vista física de un sistema, la cual es capaz de establecer correspondencia entre las clases y los componentes de implementación. En la Figura 11 se muestra el diagrama de componentes de la aplicación.

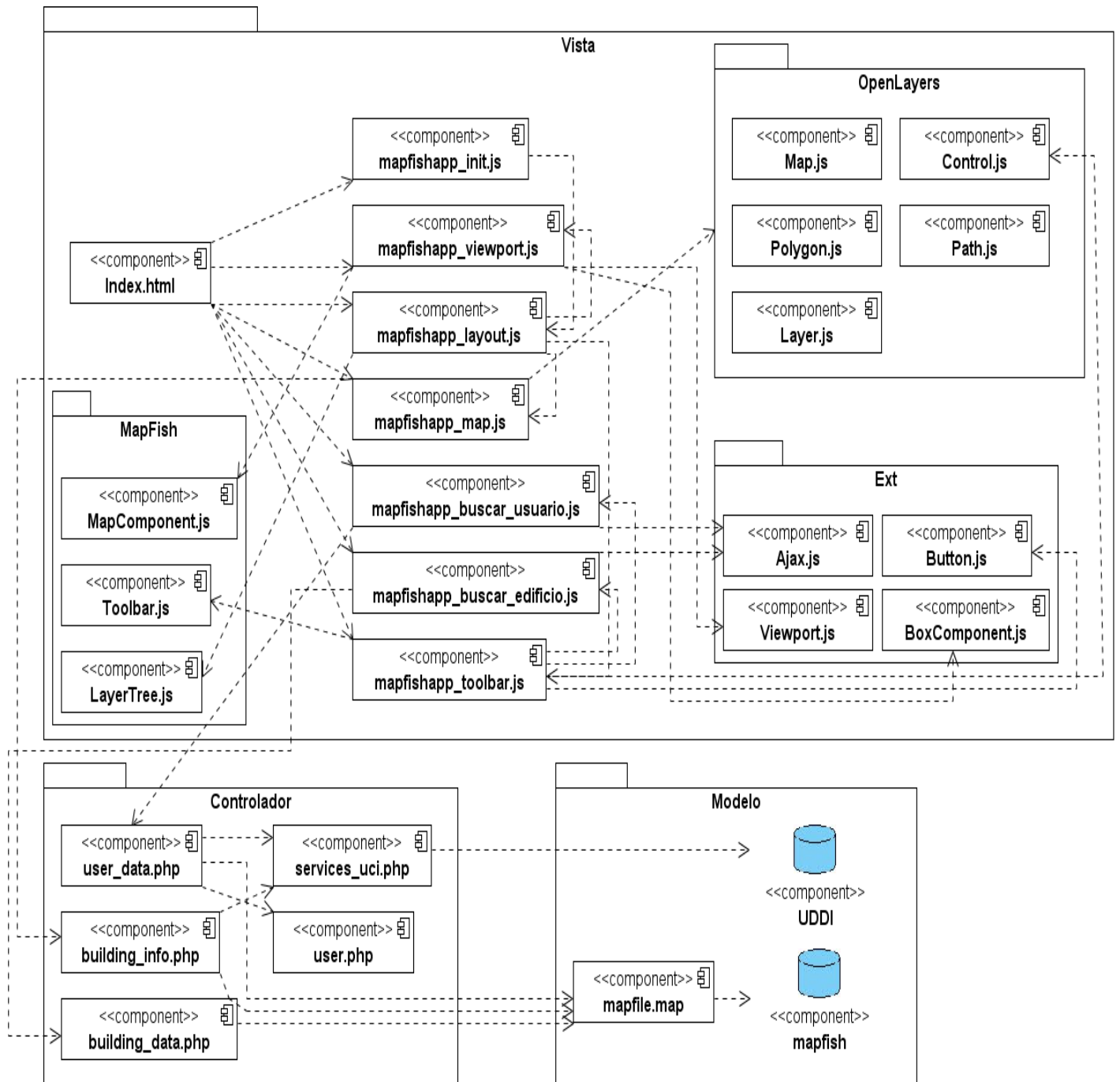


Figura 11 Diagrama de Componentes

### 3.3 Prueba

El desarrollo de software implica gran cantidad de actividades en la que la presencia del hombre es indispensable y con ella la posibilidad de que aparezcan errores en los sistemas debido a que es imposible que el comportamiento humano sea perfecto. Es por ello que surge la necesidad de realizar pruebas a la aplicación luego de construida para detectar y corregir posibles errores.

Las pruebas de software son un elemento crítico para ganar en calidad y fortaleza del producto y son capaces de verificar completamente las especificaciones tanto de diseño como de codificación. Entre mejor sea la prueba que se realice, más alta será la probabilidad de encontrar errores. Existen varios métodos de prueba como el de Caja Blanca y el de Caja Negra, siendo este último el que se aplicará al sistema propuesto.

#### 3.3.1 Método de Caja Negra

Las pruebas de caja negra se llevan a cabo en la interfaz de la aplicación. Lo que se pretende probar con este método es que las funciones del software son operativas, que los datos insertados son correctos y que se obtiene la respuesta esperada.

La realización de esta prueba se basa en los requisitos funcionales y los errores que principalmente encuentran son de interfaz, de tipos de datos y de funciones incorrectas o ausentes. En la aplicación se aplicará el método de caja negra utilizando la técnica de partición equivalente que consiste en dividir el dominio de entrada de un programa en clases de datos de los que se pueden derivar casos de prueba.

A continuación en la Tabla 4, se muestra el caso de prueba para el CU Mostrar Mapa. El resto de los casos de prueba se encuentran en el Anexo #4.

Tabla 5 Mostrar Mapa

Entrada	Resultado	Evaluación de la prueba
El usuario selecciona la o las capas que desea que se muestren en pantalla.	Se muestra la o las capas seleccionadas por el usuario.	Prueba Satisfactoria.

### 3.4 Estudio de Factibilidad

La planificación de proyectos es un proceso necesario antes de comenzar a desarrollar el mismo, no se puede hablar de esta como una etapa independiente en un momento determinado sino que durante el avance del proyecto se podrán modificar tareas, reasignar recursos, etc.

Uno de los objetivos de la planificación es realizar la estimación del proyecto, donde se pueden definir la mejor y peor situación, de modo que los resultados del proyecto pueden limitarse. Con la estimación de los recursos, el tiempo y los resultados del proyecto se prevén pérdidas innecesarias durante el desarrollo del sistema.

Realizar la factibilidad del proyecto es de gran importancia para cambios futuros dentro de la aplicación. Es por ello que debe efectuarse con gran cuidado y debe tener la calidad requerida para que no existan gastos innecesarios de personal y tiempo.

#### 3.4.1 Análisis de puntos de Casos de uso

El análisis de puntos de CU es un método de estimación del esfuerzo de desarrollo del proyecto que está establecido en los CU y en algunas complejidades de ambiente que influyen en el tiempo de progreso del sistema. Se basa en la asignación de "pesos" a varios factores que afectan al sistema para determinar el tiempo total estimado para el proyecto a partir de dichos factores.

En la Tabla 5 se muestra el **Factor de peso del actores sin ajustar (UAW)**

Tipo de Actor	Descripción	Factor de Peso
Usuario	Una persona que interactúa con el sistema mediante una interfaz gráfica.	3

Tabla 6 Factor de peso de los actores sin ajustar

$$UAW = \sum_{1}^{n} (\text{Actor} * \text{Peso})$$

$$UAW = 1 * 3$$

$$UAW = 3$$

En la Tabla 6 se muestra el **Factor de peso de los CU sin ajustar (UUCW)**

Tipo de Caso de Uso	Descripción	Factor de Peso
Mostrar Mapa	El Caso de Uso contiene de 4 a 7 Transacciones.	10
Redimensionar Mapa	El Caso de Uso contiene de 1 a 3 Transacciones.	5
Realizar Búsquedas	El Caso de Uso contiene de 4 a 7 Transacciones.	10
Mostrar Información	El Caso de Uso contiene de 4 a 7 Transacciones.	10
Calcular Ruta Mínima	El Caso de Uso contiene de 4 a 7 Transacciones.	10
Visualizar Coordenadas	El Caso de Uso contiene de 1 a 3 Transacciones.	5
Calcular Área	El Caso de Uso contiene de 4 a 7 Transacciones.	10
Medir Distancia	El Caso de Uso contiene de 4 a 7 Transacciones.	10

Tabla 7 Factor de peso de los CU sin ajustar

$$UUCW = \sum_{1}^{n} (CU * Peso)$$

$$UUCW = 2 * 5 + 6 * 10$$

$$UUCW = 70$$

Una vez obtenidos el UAW y el UUCW se puede calcular el Factor de Peso de los CU sin ajustar (UUCP)

$$UUCP = UAW + UUCW$$

$$UUCP = 3 + 70$$

Luego de haber calculado los UUCP se deben ajustar los CU a una serie de características técnicas y ambientales.

### 3.4.1.1 Cálculo de Puntos de CU ajustados

$$UCP = UUCP * TCF * EF$$



UCP: Puntos de Casos de Uso ajustados.

UUCP: Puntos de Casos de Uso sin ajustar.

TCF: Factor de complejidad técnica.

EF: Factor de ambiente.

### Factor de Complejidad Técnica (TCF)

$$\text{TCF} = 0.6 + 0.01 * \sum_{1}^n (\text{Peso} * \text{Valor Asignado})$$

En la Tabla 7 se muestra una lista con varios factores y una relación peso, valor y total para cada uno de ellos.

Factor	Descripción	Peso	Valor	Total
T1	Sistema distribuido	2	3	6
T2	Tiempo de respuesta	1	2	2
T3	Eficiencia del usuario final	1	1	1
T4	Procesamiento interno complejo	1	2	2
T5	El código debe ser reutilizable	1	3	3
T6	Facilidad de instalación	0.5	3	1.5
T7	Facilidad de uso	0.5	3	1.5
T8	Portabilidad	2	2	4
T9	Facilidad de cambio	1	2	2
T10	Concurrencia	1	1	1
T11	Incluye objetivos especiales de seguridad	1	1	1
T12	Provee acceso directo a terceras partes	1	5	5
T13	Se requieren facilidades especiales de entrenamiento a usuarios	1	2	2

Tabla 8 Factor de complejidad técnica

$$\text{TCF} = 0.6 + 0.01 * 32$$

$$\text{TCF} = 0.6 + 0.032$$

$$\mathbf{TCF = 0.92}$$

### Factor Ambiente

$$\mathbf{EF = 1.4 - 0.03 * \sum_1^n (Peso * Valor Asignado)}$$

En la Tabla 8 se muestran varios factores contabilizados en cuanto a peso y valores.

Factor	Descripción	Peso	Valor	Total
E1	Familiaridad con el modelo de proyecto utilizado	1.5	3	4.5
E2	Experiencia en la aplicación	0.5	2	1
E3	Experiencia en orientación a objetos	1	5	5
E4	Capacidad del analista líder	0.5	4	2
E5	Motivación	1	5	5
E6	Estabilidad de los requerimientos	2	5	10
E7	Personal a tiempo parcial.	-1	3	-3
E8	Dificultad del lenguaje de programación	-1	3	-3

Tabla 9 Factor Ambiente

$$\mathbf{EF = 0.03 * 21.5}$$

$$\mathbf{EF = 1.4 - 0.645}$$

$$\mathbf{EF = 0.76}$$

Una vez obtenidos el UUCP, el TFC y el EF se puede calcular el Factor de Peso de los CU ajustados (UCP).

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

$$\mathbf{UCP = 73 * 0.92 * 0.76}$$

$$\mathbf{UCP = 51}$$

#### 3.3.1.4 Esfuerzo Total

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

E: Esfuerzo estimado en horas-hombre

UCP: Puntos de Casos de Uso ajustados

CF: Factor de conversión

$$E = 51 * 20$$

$$E = 1020 \text{ horas/hombre}$$

Este esfuerzo es solo para el flujo de trabajo de implementación y a continuación en la Tabla 9, se procede a calcular el esfuerzo para todo el ciclo de vida del proyecto.

Actividad	% esfuerzo	Valor esfuerzo
Análisis	10%	255 horas-hombre
Diseño	20%	510 horas-hombre
Implementación	40%	1020 horas-hombre
Prueba	15%	382.5 horas-hombre
Sobrecarga	15%	382.5 horas-hombre
Total	100%	2550 horas-hombre

Tabla 10 Relación Horas/Hombre en cada flujo de trabajo

Suponiendo que una persona trabaje 8 horas por día, y un mes tiene como promedio 30 días; la cantidad de horas que puede trabajar una persona en 1 mes es 240 horas. Si el valor del esfuerzo total es de 2550 horas-hombre y la cantidad de horas se puede trabajar en 1 mes es 240 horas, entonces  $ET = 10.6$  mes-hombre

Teniendo en cuenta que el proyecto fue realizado por 2 personas que emplearon el mismo esfuerzo y tiempo, se puede concluir que el sistema puede desarrollarse en aproximadamente 5 meses, sin tener en cuenta los inconvenientes que ocurrieron y afectaron directamente en la realización del producto.

### 3.5 Conclusiones

En este capítulo se realizó el modelo de implementación, plasmando el diagrama de componentes para mostrar la relación de los principales elementos de la solución y como base para obtener un producto con la máxima fortaleza posible. Se cumplieron las pruebas al sistema utilizando el método de caja negra para comprobar la calidad y validez de las funcionalidades implementadas y demostrando que la solución cumple con todos los requisitos funcionales planteados en el tópico 2.2 y se realizó el estudio de

factibilidad demostrando que el proyecto era factible y no era una pérdida en cuanto a consumo de recursos y de tiempo.

## CONCLUSIONES GENERALES

En el presente trabajo de diploma se realizó un estudio profundo de los Sistemas de Información Geográfica, se analizaron rigurosamente varias metodologías y tecnologías usadas en la actualidad para la construcción de aplicaciones y se le solucionó la interrogante planteada al comienzo de la investigación.

Finalmente, se concluye que:

- ✓ Se utilizó el marco de trabajo MapFish permitiendo construir un SIG para la UCI que disminuyera el esfuerzo requerido para integrarse con aplicaciones web.
- ✓ Se mejoró la toma de decisiones por parte de los usuarios de la aplicación a la hora de saber la distancia exacta de un punto a otro para caminar por la Universidad.
- ✓ Con su implantación en la Universidad se favorecerá al ahorro monetario debido a que todas las herramientas y el marco de trabajo utilizado se encuentran bajo una licencia de software libre, lo cual es de gran beneficio en la actual lucha hacia la independencia tecnológica en la cual está inmerso el país.
- ✓ Se implementó un producto cumpliendo con el diseño indicado y posibilitando seguir conociendo sobre ese amplio mundo de los SIG.
- ✓ El uso de métodos de investigación facilitó el estudio profundo del proceso de referenciar la información geográficamente.
- ✓ Como principal aporte a la investigación se desarrollo el CU Calcular Ruta Mínima el cual facilita a los usuarios conocer la distancia más cercana de un punto a otro.

## RECOMENDACIONES

- ✓ Continuar el ciclo de desarrollo de la solución informática incluyendo nuevas funcionalidades para obtener una aplicación con mayor competitividad.
- ✓ Ampliar el campo de acción del trabajo de diploma para que no solamente se quede enmarcado en la UCI, sino que redefiniendo las funcionalidades se aplique en todo el país.
- ✓ Utilizar el producto resultante en la universidad.

## REFERENCIAS BIBLIOGRÁFICAS

- Aplicaciones de Sistemas de Información Geográfica en Cuba, 2005.** Aplicaciones de Sistemas de Información Geográfica en Cuba. Batista Silva, José Luis. 2005. ISSN: 1.131-9.100, 2005.
- Conallen, Jim. 1999.** Modeling Web Application with UML, Communication of the ACM. 1999.
- Corrales Martínez, Yormis. 2009.** Componentes para la configuración de la gestión del multilinguaje y la gestión de trazas del sistema Cedrux. Ciudad de la Habana : Universidad de las Ciencias Informáticas, 2009.
- CubaSi. 2008.** CubaSi. [En línea] 2008.  
<http://www.cubasi.cu/DesktopDefault.aspx?SPK=160&CLK=195138&LK=2&CK=100184&SPKA=35>.
- Dávila Martínez, Francisco J. 2004.** Sociedad Geográfica Española. [En línea] 2004.  
<http://www.sge.org/cartografia/sig2.pdf>.
- Diario Granma. 2007.** Granma. [En línea] 2007.  
<http://www.granma.cubaweb.cu/2007/07/20/nacional/artic05.html>.
- Diccionario Informático. 2009.** Alegsá. [En línea] 2009. <http://www.alegsa.com.ar/Dic/sgdg.php>.
- Enciclopedia Libre Universal en España. 2009.** Enciclopedia. [En línea] 25 de 6 de 2009.  
<http://enciclopedia.us.es/index.php/Geograf%C3%ADa>.
- FayerWarey. 2009.** FayerWarey. [En línea] 2009. <http://www.fayerwayer.com/2009/04/apache-sigue-siendo-el-rey-entre-los-grandes-sitios>.
- GeoInfo. 2008.** GeoInfo. [En línea] 2008. <http://www.geoinfo-int.com/htmls/sig.html>.
- Glosario.net. 2006.** Glosario.Término. [En línea] 27 de 10 de 2006. <http://tecnologia.glosario.net/terminos-tecnicos-internet/capa-%28layer%29-258.html>.
- IDAS. 2008.** IDAS. [En línea] 2008. [http://www.idasnet.com/idas\\_site/idasnet\\_esp/noticias/arcgis\\_esp.htm](http://www.idasnet.com/idas_site/idasnet_esp/noticias/arcgis_esp.htm).

**Oviedo Álvarez, Vivian. 2005.** *Cartografía Temática*. Ciudad de la Habana : Facultad de Geografía, Universidad de la Habana, 2005. 117.

**Prodevelop. 2009.** Clientes Ligeros de Mapas. [En línea] 2009.  
<http://www.prodelop.es/es/tecs/geo/clienteligero>.

**Revista Electrónica Tropical . 2008.** Revista Electrónica Tropical . [En línea] 6 de 3 de 2008.  
[http://www.iit.jalisco.gob.mx/RET/E6/RET6\\_postg.pdf](http://www.iit.jalisco.gob.mx/RET/E6/RET6_postg.pdf).

**Salitvech Konstantin, A. 1981.** *Cartografía*. Ciudad de la Habana : Pueblo y Educación, 1981. 255.

**Sanz Salinas, Jorge Gaspar. 2009.** Prodevelop. [En línea] 23 de 3 de 2009.  
<https://confluence.prodevelop.es/display/pan/Panorama+del+ecosistema+de+Software+Libre+para+SIG+-+completo>.

**Soluciones Gráficas S.A. 2005.** SolGraf. [Online] 11 2005.  
[http://www.solgrafperu.com/productos/geomedia\\_web/geomedia\\_webmap.html](http://www.solgrafperu.com/productos/geomedia_web/geomedia_webmap.html).

**Thompson, Ivan. 2008.** PromonegocioS.net. [En línea] 10 de 2008  
<http://promonegocios.net/mercadotecnia/definicion-informacion.html>.

**Unzurrunzaga, Carolina, Genoves, Patricia y Petruccelli, Mabel. 2007.** FaHCE. [En línea] 5 de 2007.  
<http://www.fahce.unlp.edu.ar/academica/Areas/geografa/Catedras/tnicasdeanalisisespacial>.



## BIBLIOGRAFÍA

1. **Hernan, Jesús.** Software Libre.Historia, ética y praxis de una ciencia en red. [En línea] Mayo 30, 2009. <<http://knol.google.com/k/jesus-hernan/software-libre/1gpqch708n8iz/4#>>
2. **MapFish.** Página Oficial de Mapfish. [En Línea] 2009. <[www.mapfish.org](http://www.mapfish.org)>
3. **Panamá, Fraternidad de Software Libre.** Historia del Software Libre. [En línea] Diciembre 11, 2009. <<http://www.fslpma.org>>
4. **Conallen, Jim.** Modeling Web application with UML, Communication of the ACM. 10,1999, Vol 42, páginas 63-70.
5. **Free Software Foundation. 2008.** EL Sistema Operativo GNU. EL Sistema Operativo GNU. [En Línea] 2008. <http://www.gnu.org/home.es.html>
6. **GeoMedia. 2002.** The GeoMedia Architecture Advantage. [En Línea] 2002. <[http://spatialnews.geocomm.com/whitepapers/GeoMedia\\_Architecture\\_Advantage.pdf](http://spatialnews.geocomm.com/whitepapers/GeoMedia_Architecture_Advantage.pdf)>
7. **Revista Electrónica Tropical.** PosgreSQL: Una solución para el almacenamiento y el análisis de los datos espaciales. [En Línea] Marzo 6, 2008 <[http://www.iit.jalisco.gob.mx/RET/E6/RET6\\_postg.pdf](http://www.iit.jalisco.gob.mx/RET/E6/RET6_postg.pdf)>
8. **Wells, Don.** Extreme Programming: A gentle introduction. [En Línea] Febrero 17, 2006. <[www.extremeprogramming.org](http://www.extremeprogramming.org)>