



Facultad #6



Título: Sistema de Información Geográfica para la Universidad de las Ciencias Informáticas versión 2.0.

Trabajo de diploma para optar por el título de ingeniero en ciencias informáticas.

Autor: Alejandro Paret Fuentes.

Tutor: Ing. Héctor René Sánchez Falero.

Co-Tutor: Ing. Dania Mora Valdés.

Ciudad de La Habana, 28 de junio del 2011.

“Año 53 de la Revolución”

Agradecimientos.

... a mi mamá, por ser la persona que me ha forjado con mucho sacrificio y convertirme en lo que soy , por darme todo lo que siempre ha estado a su alcance y mucho más, Gracias por tu amor mamá. Para ti va dedicado este título, Ya somos Ingenieros...

... a mi hermano, el pelú quien siempre ha sido ¡un hermano! aunque nos fajemos por la PC...

... a Ely, por ser el padre que todos necesitan para conversar, querer y que te quiera...

...a Dania, por tutorarme, aguantarme, perdonarme y amarme sin condiciones a lo largo de este tiempo...

...a mi abuela Teresa y a Pipo por estar siempre dispuesto a ayudarme en cualquiera de mis locuras...

...a mis tíos, tías y primos...

... a Héctor, por tutorarme a lo largo de esta investigación, por estar siempre en el jabber cuando lo necesitaba, a pesar de protestar por la exposición, y siempre tener la cuenta bloqueada...

... al Compa por ser el Gran Capitán SIG y ayudarme desinteresadamente con el desarrollo del sistema...

... a los grandes amigos y amigas que he hecho durante estos 6 años, Humberto (La Taquilla), Eylena (La Pesqui), Yoandy (El Gordo), Aurelio (CR7), Inda (El Calvo), Leiber (El Tieso), El Viña, Alain, Gelsys, Lianet, Ariadna, David (el Cojo)...

... a mis compañeros de grupo, 9101, 9103, 9203, 9301, 9401 y 9501...

... a los compañeros de la brigada Ernesto Che Guevara, en especial a Brezo y Paco...

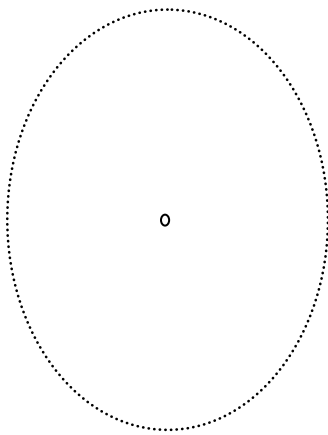
... a todos los integrantes del proyecto Aplicativos SIG que contribuyeron al desarrollo del sistema, en especial a Aliana, Alejandro, Adrián, Karel, Lilianne, Carlos...

... a todos los que de una forma u otra me ayudaron, no solo a que esta investigación se hiciera posible, también a los que me han ayudado a llegar donde estoy hoy...

Gracias a todos...

Dedicatoria.

...Este trabajo va dedicado a mi...



de parte de su



Declaración de Autoría.

Declaramos que somos los únicos autores del trabajo titulado:

“Sistema de Información Geográfica para la Universidad de las Ciencias Informáticas versión 2.0.”

Y autorizamos a la Universidad de las Ciencias Informáticas los derechos patrimoniales del mismo, con carácter exclusivo.

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

Alejandro Paret Fuentes.

Héctor René Sánchez Falero.

Datos de Contacto.

Síntesis del Tutor: Ing. Héctor René Sánchez Falero.

Profesión: Ingeniero en Ciencias Informáticas.

Categoría docente: Instructor.

Año de graduado: 2009.

Síntesis del Co-Tutor: Ing. Dania Mora Valdés.

Profesión: Ingeniero en Ciencias Informáticas.

Categoría docente: Instructor.

Año de graduado: 2008.

Opinión del Tutor.

Opinión del Oponente.

Resumen.

El Sistema de Información Geográfica (SIG) para la Universidad de las Ciencias Informáticas (UCI) en su segunda versión es una aplicación diseñada con el objetivo de permitir la representación geoespacial de la información asociada a la Universidad. Proporciona servicios de acceso a la información geográfica, para su consulta, análisis y visualización, mediante una interfaz de usuario sencilla y de fácil manejo que pueda ser utilizada por usuarios no especializados en tecnología de SIG. Para su desarrollo se considera el interés del país en lograr la soberanía tecnológica, por lo que se hace uso de herramientas y tecnologías libres, incluyendo la plataforma soberana GeneSIG, desarrollada en la misma Universidad.

Presenta funcionalidades que agilizan la búsqueda de personas, búsquedas por direcciones IP lugares de interés así como la localización de los mismos sobre un mapa de la Universidad. El SIGUCI 2.0 ha posibilitado un alivio para realizar análisis en la residencia, a la hora de hacer una tematización de edificios para localizar cuáles son de estudiantes y cuáles de profesores, con el objetivo de realizar una búsqueda más específica; para saber cuáles edificios pertenecen a qué facultades y de qué género son las personas que viven en los mismos. También permite consultar la cantidad de personas que habitan en un apartamento, para así realizar un control de las capacidades de los mismos, además permite modificar la información socio-económica de las estructuras para así mantener la información actualizada. Este sistema está desarrollado para mejorar la toma de decisiones en cuanto al manejo de los recursos y el personal de la Universidad.

Índice.

Agradecimientos	2
Declaración de autoría	4
Datos de contacto	5
Opinión del tutor	6
Opinión del oponente	7
Resumen	8
Índice	9
Índice de tablas	12
Índice de figuras	12
Introducción	13
Capítulo 1	18
Marco teórico de la investigación	18
1.1. INTRODUCCIÓN.....	18
1.2. CONCEPTOS ASOCIADOS AL DOMINIO DEL PROBLEMA.....	18
1.2.1. Sistema.....	18
1.2.2. Sistema Informático.....	18
1.2.3. Sistemas de Información Geográfica (SIG).....	18
1.2.4. Georreferenciación.....	19
1.2.5. Mapa.....	19
1.2.6. Cartografía.....	19
1.2.7. Datos geográficos.....	19
1.2.8. Escala.....	19
1.3. OBJETO DE ESTUDIO.....	20
1.3.1. Descripción General.....	20
1.3.2. Descripción actual del dominio del problema.....	20
1.3.3. Situación Problemática.....	21
1.3.4. Evolución de los Sistemas de Información Geográfica a nivel mundial.....	21
1.3.5. Evolución de los Sistemas de Información Geográfica en Cuba.....	23
1.3.6. Soluciones de Sistemas de Información Geográfica para universidades.....	24
1.4. CONCLUSIONES PARCIALES.....	25
Capítulo 2	26
Tendencias y tecnologías	26
2.1. INTRODUCCIÓN.....	26
2.2. SOFTWARE.....	26
2.2.1. Software Libre.....	26
2.3. SISTEMA OPERATIVO.....	27
2.4. APLICACIONES INFORMÁTICAS.....	28
2.4.1. Aplicación Web.....	28
2.5. ARQUITECTURA DE SOFTWARE.....	29
2.5.1. Arquitectura Cliente-Servidor.....	30

2.5.2.	Modelo-Vista-Controlador (MVC)	30
2.6.	WEB SERVICE (<i>SERVICIO WEB</i>)	31
2.7.	SERVIDORES.....	32
2.7.1.	Servidor Web.	32
2.7.1.1.	Apache.....	32
2.7.2.	Servidor de Mapas.....	32
2.7.2.1.	MapServer.	32
2.8.	SISTEMA GESTOR DE BASE DE DATOS.	33
2.8.1.	PostgreSQL.	33
2.9.	LENGUAJES Y TECNOLOGÍAS DE PROGRAMACIÓN.	34
2.9.1.	Lenguaje y tecnologías de programación del lado del cliente.....	34
2.9.1.1.	HTML: Hyper Text Markup Language (<i>Lenguaje de Marcado de Hipertexto</i>).	34
2.9.1.2.	JavaScript.	34
2.9.1.3.	AJAX: Asynchronous Java Script and XML.....	35
2.9.2.	Lenguaje y tecnologías de programación del lado del servidor.	36
2.9.2.1.	PHP: Hypertext Pre-processor.	36
2.9.2.2.	Framework (<i>Entorno de trabajo</i>).....	36
2.10.	XML: EXTENSIVE MARKUP LANGUAGE (<i>LENGUAJE EXTENSIBLE DE MARCAS</i>).	37
2.11.	IDE: INTEGRATED DEVELOPMENT ENVIROMENT (<i>ENTORNOS INTEGRADOS DE DESARROLLO</i>).	38
2.11.1.	Eclipse.	38
2.12.	LENGUAJE UNIFICADO DE MODELADO (UML).	38
2.13.	METODOLOGÍA DE DESARROLLO DE SOFTWARE.....	39
2.13.1.	RUP: Rational Unified Process (<i>Proceso Unificado de Desarrollo</i>).	39
2.14.	HERRAMIENTA CASE.....	42
2.14.1.	Visual Paradigm.....	42
2.15.	CONCLUSIONES PARCIALES.....	42
Capítulo 3	43
Presentación de la solución propuesta.	43
3.1.	INTRODUCCIÓN.	43
3.2.	ENTORNO DONDE TRABAJARÁ EL SISTEMA.	43
3.2.1.	Diagrama de clases del Modelo de Dominio.	43
3.3.	REQUISITOS FUNCIONALES.....	45
3.3.1.	Localizar Ubicación por IP.	45
3.3.2.	Tematizar.	45
3.3.3.	Gestionar Datos.....	46
3.3.4.	Localizar IP.	46
3.3.5.	Autenticar.	47
3.3.6.	Mostrar Ayuda.....	47
3.4.	REQUISITOS NO FUNCIONALES.	48
3.5.	DESCRIPCIÓN DEL SISTEMA PROPUESTO.	52
3.5.1.	Descripción de los Actores del Sistema.	52
3.5.2.	Descripción textual de los Casos de Uso del Sistema.	53
3.5.2.1.	Tematizar.	53

3.5.2.2. Gestionar Datos.....	55
3.6. CONCLUSIONES PARCIALES.....	58
Capítulo 4	59
Construcción de la solución propuesta.	59
4.1. INTRODUCCIÓN.	59
4.2. ARQUITECTURA PROPUESTA.....	59
4.3. MODELO DE DISEÑO.	60
4.3.1. Patrones de diseño.	62
4.3.2. Diagrama de clases del diseño.....	62
4.3.2.1. Diagrama de clases del diseño del Caso de Uso Localizar IP.....	63
4.4. PRINCIPIOS DE DISEÑO.	64
4.4.1. Estándares de interfaz de la aplicación.	64
4.5. DISEÑO DE LA BASE DE DATOS.	65
4.5.1. Diagrama Entidad-Relación.....	65
4.5.2. Diagrama de Clases Persistentes.	66
4.6. MODELO DE DESPLIEGUE.	68
4.7. MODELO DE IMPLEMENTACIÓN.	68
4.7.1. Diagramas de Componentes.....	69
4.8. PRUEBAS DEL SISTEMA PROPUESTO.....	70
4.8.1. Pruebas de Caja Negra.	70
4.8.2. Validación de Resultados.	71
4.9. CONCLUSIONES.	73
Conclusiones generales.	74
Recomendaciones.	75
Glosario de términos.	76
Trabajos citados.	79
Bibliografía consultada.	81

Índice de Tablas.

<i>Tabla 1 Requisito Funcional Localizar Ubicación por IP.</i>	45
<i>Tabla 2 Requisito Funcional Tematizar.</i>	45
<i>Tabla 3 Requisito Funcional Gestionar Datos.</i>	46
<i>Tabla 4 Requisito Funcional Localizar IP.</i>	46
<i>Tabla 5 Requisito Funcional Autenticar.</i>	47
<i>Tabla 6 Requisito Funcional Mostrar Ayuda.</i>	47
<i>Tabla 7 Requisito Funcional Identificar Estructura.</i>	48
<i>Tabla 8 Descripción de los Actores del Sistema.</i>	52
<i>Tabla 9 Descripción textual Caso de Uso Tematizar.</i>	55
<i>Tabla 10 Descripción textual Caso de Uso Gestionar Datos.</i>	58
<i>Tabla 11 Descripción de los paquetes de un Plugin.</i>	62
<i>Tabla 12 Caso de prueba # 1. Utilizando técnicas de caja negra.</i>	70
<i>Tabla 13 Caso de prueba # 2. Utilizando técnicas de caja negra.</i>	71
<i>Tabla 14 Caso de prueba # 3. Utilizando técnicas de caja negra.</i>	71
<i>Tabla 15 Validación del tiempo de respuesta.</i>	72

Índice de Figuras.

<i>Fig. 1 Fases y flujos de trabajo de RUP.</i>	40
<i>Fig. 2 Modelo del Dominio.</i>	43
<i>Fig. 3 Diagrama de Casos de Uso.</i>	52
<i>Fig. 4 Diseño de CartoWeb.</i>	60
<i>Fig. 5 Diseño de un Plugin de CartoWeb.</i>	61
<i>Fig. 6 Localizar IP.</i>	63
<i>Fig. 7 Diagrama Entidad-Relación.</i>	65
<i>Fig. 8 Diagrama Entidad-Relación. (Cont.)</i>	66
<i>Fig. 9 Diagrama de Clases Persistentes.</i>	67
<i>Fig. 10 Modelo de Despliegue.</i>	68
<i>Fig. 11 Diagrama de Componentes Localizar IP.</i>	69

Introducción.

El Hombre siempre ha tenido la necesidad de transmitir sus experiencias. Desde el año 2300 antes de Cristo, los mapas se elaboraron sobre pergaminos y fueron escritos utilizando plumas de aves. Con la invención de la brújula, la humanidad pudo desarrollar mapas de diferentes escalas y más exactos, posteriormente el surgimiento de la imprenta, el cuadrante¹ y el nonio² posibilitó el incremento de la producción de mapas. El telescopio; entre otros dispositivos que permiten hacer un estudio más exacto de la tierra; brindó a las personas encargadas de elaborar los mapas, encontrar la latitud midiendo ángulos a partir de la Estrella Polar, tanto de día, como de noche. Los mapas representan distintos tipos de información, que van desde el orden social y económico, hasta los fenómenos naturales que ocurren tanto en la Tierra, como en el espacio.

La Cartografía es la ciencia que estudia o se encarga de la elaboración de los mapas geográficos y territoriales, ha abarcado desde los primeros trazos realizados en la arena, hasta el uso de novedosas técnicas geodésicas, fotogramétricas, de teledetección y servicios de mapas en Internet.

Los avances en las tecnologías del Siglo XX y la creación de las modernas computadoras han llevado a una revolución en la rama de la Cartografía. Existe gran cantidad de equipos como monitores, impresoras, escáneres y trazadores estéreo-analíticos³, junto con los programas de computadora para la visualización, el procesamiento de imagen, el análisis espacial y la gerencia de bases de dato; estos equipos han brindado la posibilidad de revolucionar notablemente la forma de elaborar los mapas.

En la actualidad la mayoría de los mapas se construyen usando programas de computadoras, entre los que se destacan los Sistemas de Información Geográfica (SIG). La información espacial que es georreferenciada en estos sistemas puede almacenarse en bases de datos. Este tipo de sistema posibilita que los mapas sean más dinámicos e interactivos, permitiendo que sean manipulados de forma digital.

¹ **Cuadrante:** *Antiguo instrumento utilizado para medir ángulos en astronomía y navegación.*

² **Nonio:** *Segunda escala auxiliar que tienen algunos instrumentos de medición.*

³ **Trazadores estéreos analíticos:** *Periférico de salida de un ordenador con el que se realizan gráficos sobre papel.*

Un SIG está conformado por la integración entre el hardware⁴, software⁵ y los datos geográficos representados. Está diseñado para almacenar, capturar, analizar y desplegar la información geográfica referenciada con el fin de resolver problemas de investigación científica, gestión de recursos, gestión de activos, arqueología, evaluación del impacto ambiental, planificación urbana, cartografía, sociología, geografía histórica, marketing, logística, planificación y para la toma de decisiones sobre cualquier situación por compleja que sea.

Desde hace varios años se ha ido desarrollando un movimiento mundial a favor de lo que se denomina software libre, que no es más que el software, que una vez obtenido puede ser usado, copiado, estudiado, modificado y redistribuido libremente según Richard Stallman⁶. Cuba ha ido alcanzando un desarrollo informático de forma progresiva, lo cual ha conllevado a procesos de transformaciones desde el punto de vista tecnológico. El país se encuentra en una fase de estudio relacionado con el desarrollo de los Sistemas de Información Geográfica y su vinculación y adaptación con las alternativas libres existentes en el mundo. Paralelo a estas mejoras en el campo tecnológico, se han creado empresas e instituciones con el objetivo de contribuir al crecimiento informático cubano y en especial para aumentar el desarrollo y estudio de los SIG.

En nuestro país, la empresa GEOCUBA se dedica al trabajo con mapas y es la única patentada por las leyes del gobierno para la creación y manipulación de los mismos. Desde hace varios años, GEOCUBA, la Universidad de las Ciencias Informáticas (UCI) y las Fuerzas Armadas Revolucionarias (FAR), trabajan de conjunto en la creación de Sistemas de Información Geográfica sobre plataformas de software libre para el servicio a distintas empresas nacionales e internacionales.

Desde hace un año el SIGUCI se presentó en la universidad, como una herramienta alternativa de consulta. Este sistema brinda funcionalidades que agilizan la localización ágil de personas, lugares de interés y la localización de los mismos sobre un mapa de la universidad.

Con el crecimiento de la información que se debe representar, surgió la necesidad de incluir en el sistema nuevas funcionalidades para satisfacer la demanda de los usuarios:

⁴ **Hardware:** *Parte física de la computadora, dígase monitor, teclado, mouse y unidad central de procesamiento.*

⁵ **Software:** *Parte lógica de la computadora, dígase los programas que logran el funcionamiento de la misma.*

⁶ **Richard Stallman:** *Programador estadounidense y figura relevante del movimiento de software libre en el mundo.*

- ✓ Ubicar el lugar desde dónde se está utilizando la aplicación.
- ✓ Búsquedas por direcciones IP.
- ✓ Poder conocer la información de las estructuras referenciadas en el mapa.
- ✓ La opción de autenticarse en la aplicación ya sea por dominio uci o usuario local del sistema.
- ✓ Que se pueda modificar la información socio-económica de cada estructura.
- ✓ Poder crear tematizaciones del mapa según parámetros establecidos por el usuario.
- ✓ Crear una guía que muestre cómo se maneja la aplicación y que presente un glosario de términos propios de la rama de la geoinformática.

Algunas deficiencias actuales del sistema:

- ✓ Presenta mucho código que no se utiliza.
- ✓ La interfaz es poco amigable e intuitiva.
- ✓ Presenta funcionalidades propias de la rama de la geografía que dificulta el trabajo de cualquier usuario de la universidad.
- ✓ Funcionalidades que presentan problemas de funcionamiento.
- ✓ No es compatible con algunos navegadores.

Debido a lo que se ha planteado anteriormente surge el siguiente problema a resolver:

¿Cómo perfeccionar las funcionalidades del Sistema de Información Geográfica de la Universidad de las Ciencias Informáticas?

Al haberse hecho un análisis de esta problemática, se necesita encontrar una solución teniendo como objeto de estudio de la investigación: ***Proceso de desarrollo de Sistemas de Información Geográfica sobre la plataforma GeneSIG.***

Definiendo como objetivo general de la investigación: ***Ampliar y desarrollar funcionalidades del Sistema de Información Geográfica para la Universidad de las Ciencias Informáticas.***

Objetivos específicos trazados para el cumplimiento del objetivo general:

- ✓ Implementar la versión 2.0 del SIGUCI utilizando como base la nueva versión de la plataforma GeneSIG.
- ✓ Eliminar las funcionalidades innecesarias de la plataforma GeneSIG para la versión 2.0 del SIGUCI.
- ✓ Desarrollar una nueva interfaz para la versión 2.0 del SIGUCI.
- ✓ Desarrollar nuevas funcionalidades para el SIGUCI versión 2.0.

Dada la relación existente entre el problema a resolver, el objeto de estudio y los objetivos anteriormente enunciados se define el campo de acción como: **Los Procesos de desarrollo que integran al Sistema de Información Geográfica de la UCI.**

Con vistas a desarrollar adecuadamente los objetivos antes mencionados, se trazan las siguientes tareas:

- ✓ Realizar un estudio de los Sistemas de Información Geográfica, ventajas y desventajas de las herramientas existentes.
- ✓ Realizar la modelación de los procesos de negocio, sistema, análisis, diseño e implementación.
- ✓ Implementar el sistema Web⁷ que dará solución a los objetivos de la investigación.
- ✓ Realizar las pruebas al sistema y analizar la factibilidad de la solución.

Como Idea a defender en la investigación:

Desarrollar y ampliar las funcionalidades del Sistema de Información Geográfica para la Universidad de las Ciencias Informáticas, lo cual contribuirá a mejorar el control de los recursos humanos y tecnológicos existentes en la universidad.

Para lograr el entendimiento de la situación existente, se utilizaron los métodos siguientes:

- ✓ **Métodos Empíricos:**
 - *Entrevistas:* Se realizaron entrevistas de forma no estructurada a los dirigentes del Centro de Desarrollo GEySED de la Facultad 6, para recoger toda la información necesaria sobre las nuevas funcionalidades a desarrollar, así como las que son innecesarias para su eliminación.

⁷ **Sistema Web:** Sistema de documentos (o páginas Web) interconectados por enlaces de hipertexto, disponibles en Internet.

✓ **Métodos Teóricos:**

- *Analítico-Sintético:* Se realizó un detallado estudio de toda la bibliografía, conformada por documentos, sitios Web y tesis, lo que permitió obtener una síntesis detallada de la misma.
- *Análisis Histórico-Lógico:* Se investigó y se hizo un análisis de los Sistemas de Información Geográfica existentes en otras universidades.
- *Modelación:* Se modelaron las actividades que se desarrollan en el transcurso de la elaboración del problema a través de diagramas.

Los resultados que se esperan alcanzar con este trabajo son:

- ✓ Una versión del SIGUCI que brinde nuevas funcionalidades.
- ✓ Eliminar los errores funcionales de la primera versión.
- ✓ Lograr que el sistema sea más rápido, atractivo y operativo.
- ✓ Una interfaz semejante a la línea que persigue la Intranet de la Universidad.

Estructura de la investigación:

La presente investigación científica está compuesta por 4 Capítulos, a continuación se describen cada uno de ellos de manera resumida:

Capítulo 1: “*Marco Teórico de la Investigación*”, es el capítulo donde se explican los conceptos asociados al dominio del problema y se realiza un análisis de las soluciones existentes.

Capítulo 2: “*Tendencias y Tecnologías Actuales a Desarrollar*”, es donde se analizan las tecnologías necesarias para el correcto desarrollo del sistema informático que se desea implementar.

Capítulo 3: “*Presentación de la Solución Propuesta*” muestra la descripción del negocio resultante de las entrevistas con los clientes y una presentación del conjunto de caso de usos del sistema resultante.

Capítulo 4: “*Construcción de la Solución Propuesta*” aborda todo el proceso de construcción del problema planteado.

Capítulo 1

Marco Teórico de la Investigación.

1.1. Introducción.

El objetivo principal del desarrollo del presente trabajo es la segunda versión del SIGUCI, en este capítulo se analizan y clasifican los conceptos fundamentales asociados al dominio del problema, se brinda una introducción al desarrollo de los SIG tanto a nivel nacional como internacional, se realiza el análisis de un conjunto de soluciones utilizadas en la actualidad para la construcción de los SIG. Se identifican los principales problemas que motivaron el desarrollo de la investigación y se realiza una breve descripción de la nueva versión del sistema.

1.2. Conceptos asociados al dominio del problema.

1.2.1. Sistema.

"Sistema es una colección organizada de hombres, máquinas y métodos necesaria para cumplir un objetivo específico". (1)

"Es un todo integrado, aunque compuesto de estructuras diversas, interactuantes y especializadas. Cualquier sistema tiene un número de objetivos, y los pesos asignados a cada uno de ellos pueden variar ampliamente de un sistema a otro. Un sistema ejecuta una función imposible de realizar por una cualquiera de las partes individuales. La complejidad de la combinación está implícita". (2)

1.2.2. Sistema Informático.

Un sistema informático resulta de la unión entre tres componentes principales, Hardware que es a lo que componentes físicos se refiere, Software a los lógicos y el Humanware al recurso humano. Este tipo de sistema usa para procesar datos, dispositivos para calcular, almacenar y difundir estos datos.

1.2.3. Sistemas de Información Geográfica (SIG).

Un Sistema de Información Geográfica o SIG, es un sistema de hardware, software y procedimientos elaborados para facilitar la obtención, gestión, manipulación, análisis, modelado, representación y salida de datos espacialmente referenciados, para resolver problemas complejos de planificación y gestión. (3)

1.2.4. Georreferenciación.

La georreferenciación es la asignación de coordenadas terrestres a objetos de interés, ya sean naturales o artificiales. Permite definir la ubicación de un objeto, de manera espacial, mediante el registro de su longitud y latitud en un sistema de coordenadas específico.

1.2.5. Mapa.

Es una representación geográfica de la Tierra o de parte de ella en una superficie plana o esférica. Un mapa con propiedades métricas, significa que ha de ser posible tomar medidas de distancias, superficies y ángulos sobre él y obtener un resultado casi exacto.

1.2.6. Cartografía.

"Técnica de representar en forma convencional parte o toda la superficie terrestre sobre un plano, utilizando para este fin un sistema de proyección y una relación de proporcionalidad (escala) entre terreno y mapa". (4)

Es la disciplina que se ocupa de la organización, representación, comunicación y utilización de la información espacial con vistas a la elaboración de mapas y otros tipos de representación convencional gráfica de la Tierra o cualquier parte del Universo.

1.2.7. Datos geográficos.

Los datos geográficos son la información que permite conocer lo que se encuentra en una determinada posición en el espacio, de qué manera y en qué tiempo. Los datos geográficos no son más que variables que almacenan una determinada información geoespacial de un evento, ubicándolo en tiempo y espacio.

1.2.8. Escala.

Es la relación matemática que existe entre las dimensiones reales y las del dibujo que representa la realidad sobre un plano o un mapa. La cantidad de unidades del terreno equivale a una unidad en el mapa. Como estándar se toma la unidad centímetro (cm). (5)

Versión 2.0 del SIGUCI.

1.3. Objeto de Estudio.

1.3.1. Descripción General.

La segunda versión del SIG de la UCI tiene como objetivo principal representar la información económica de la institución sobre un mapa. Así mismo se pretende brindar nuevas funcionalidades y mejorar las ya existentes, además de adaptarle una nueva interfaz para que el análisis sobre la información reflejada en el mismo sea más eficiente y seguro, lo que convierte a este sistema en una herramienta de grandes posibilidades de uso por la comunidad universitaria. Esta nueva entrega también persigue el cumplimiento de las especificaciones Opengis⁸ que establece la Open Geospatial Consortium⁹ (OGC) y en consecuencia con la política de migración al software libre y de soberanía tecnológica que impulsa el país.

1.3.2. Descripción actual del dominio del problema.

La Universidad de las Ciencias Informáticas (UCI), es la primera universidad surgida al calor de la Batalla de Ideas, comprometida con el desarrollo tecnológico, político y social de nuestra revolución. (6). La UCI es una universidad productiva, cuya misión es la de producir software y servicios informáticos a partir de la vinculación estudio-trabajo como modelo de formación y la de ejecutar las soluciones planificadas para incrementar el impacto de la universidad en la informatización del país y las exportaciones.

La producción se concentra en el desarrollo de proyectos en Centros de Desarrollo y se destacan resultados en las esferas de salud, educación, software libre, teleformación, sistemas legales, realidad virtual, automatización, bioinformática, procesamiento de imágenes y señales digitales, y geoinformática entre otras. (7)

La universidad promueve el desarrollo de productos y servicios informáticos en aquellas ramas donde Cuba tiene un reconocido prestigio en el mundo a través del concurso de los mejores especialistas del país para lograr una solución de calidad y de impacto internacional. Desarrolla programas de informatización para la sociedad cubana a través de la relación con entidades nacionales, estos resultados alcanzados se extienden por todo el país, incluida la propia universidad donde se han desarrollado

⁸ **OpenGIS:** *Es un estándar internacional orientado a Sistemas de Información Geográfica.*

⁹ **Open Geospatial Consortium:** *Organización internacional de estándares de consenso voluntario que está liderando el desarrollo de estándares para servicios basados en localización y geoespaciales.*

disímiles soluciones para lograr un mejor funcionamiento de la misma y así estar a la vanguardia en lo que a soluciones informáticas se requiere. El SIGUCI es un ejemplo de los tantos productos desarrollado por el Centro de Desarrollo de Geoinformática y Señales Digitales (GEySED) de la Facultad 6. Este sistema ha despertado un gran interés en empresas tanto nacionales como internacionales, las cuales desean obtener soluciones similares al SIGUCI.

1.3.3. Situación Problemática.

En la UCI existen varias vías de consultar información ya sea por el directorio de personas o la guía telefónica. También existe el Sistema de Información Geográfica para la localización y consulta de objetivos georreferenciados sobre un mapa de la universidad. En la actualidad los usuarios necesitan que el sistema brinde la posibilidad de realizar búsquedas de direcciones IP, además que el mismo permita identificar las estructuras referenciadas sobre el mapa con tan solo dar clic sobre estas, que se puedan tematizar áreas de la residencia por tipos de edificio, por género y por facultades, con operaciones simples. También se necesita una guía de consulta para los usuarios sobre el funcionamiento del sistema y que permita localizar el lugar desde donde se está utilizando el SIGUCI. Además actualizar la cartografía y la base de datos con las nuevas estructuras existentes en la universidad y migrar el sistema para una nueva y mejorada versión de GeneSIG.

El actual sistema presenta funcionalidades que casi no se utilizan o tienen problemas de funcionamiento, no están implementadas algunas formas de búsqueda de manera correcta. Resulta necesario además, agregarle al SIGUCI un módulo de administración que posibilite la gestión de la información socio-económica de la universidad y la seguridad del sistema. La versión actual de la aplicación no funciona en algunos navegadores, presenta mucho código que no se utiliza, la interfaz es poco amigable y los usuarios tienden a no usar el sistema por estas deficiencias o simplemente porque no conocen su existencia. Por lo anteriormente expuesto se decidió desarrollar la versión 2.0 del SIGUCI, con mejoras y nuevas funcionalidades.

Análisis de otras soluciones existentes.

1.3.4. Evolución de los Sistemas de Información Geográfica a nivel mundial.

Las tecnologías y las ciencias informáticas se encuentran en un continuo desarrollo diario y esto ha posibilitado su aplicación en la mayoría de las ramas de la sociedad. Los Sistemas de Información Geográfica se han convertido en la última década en herramientas de trabajo esenciales a la hora de

hacer algún planeamiento urbano y de gestión de recursos. Su capacidad para almacenar, recuperar, analizar, modelar y representar grandes extensiones de terreno con enormes volúmenes de datos espaciales les han situado a la cabeza de una gran cantidad de aplicaciones.

En la actualidad se pueden encontrar varios soportes para el desarrollo de Sistemas de Información Geográfica:

- ✓ **SIG Desktop:** Un SIG de Escritorio o SIG Desktop es una aplicación informática que permite la gestión, adquisición y tratamiento de información. Caracterizados por dar opciones de una gran capacidad de procesamiento y generalmente encargados de la edición de mapas y el manejo de la información espacial para el análisis de esta. Su gran capacidad de procesamiento está condicionada solo por las propiedades del ordenador donde se use. Existen en el mundo varios productos que se encuentran a la vanguardia de la rama de los SIG para escritorio.
- **ArcGIS Desktop:** Pertenece a la Environmental Systems Research Institute (ESRI)¹⁰, es una de las más usadas con un grupo de herramientas destinadas al análisis, manipulación, edición, visualización de mapas y de datos espaciales como ArcReader, ArcMap, ArcCatalog, ArcToolbox, ArcScene y ArcGlobe incluidas en las últimas versiones de ArcGIS Desktop. Este sistema es la principal herramienta SIG del software privado.
- **gvSIG Desktop:** Iniciado en el año 2004, es un proyecto de desarrollo informático impulsado inicialmente por la Consellería de Infraestructuras y Transportes de la Generalidad Valenciana y la Unión Europea mediante el Fondo Europeo de Desarrollo Regional (FEDER). Es un sistema informático que es distribuido bajo licencia GNU GPL v2¹¹ para manejar información geográfica con precisión cartográfica, implementa servicios OGC: WMS (Web Map Service), WFS (Web Feature Service), WCS (Web Coverage Service), Servicio de Catálogo y Servicio de Nomenclador.

¹⁰ **ESRI:** Empresa dedicada a desarrollar y comercializar software para Sistemas de Información Geográfica y es una de las compañías líderes en el sector a nivel mundial.

¹¹ **GNU GPL v2:** Es una licencia creada por la Fundación de Software Libre.

- ✓ **SIG Web:** Es una serie de aplicaciones que posibilitan la visualización de datos y acceso a funcionalidades de análisis y consultas de servidores a través de la Web. Hace uso de la arquitectura Cliente-Servidor, posibilita el uso de la herramienta por varios usuarios a la misma vez y desde lugares diferentes, requieren de solamente un navegador Web para su uso y son sistemas multiplataforma.
- **SIG AGROPECUARIO:** Es una herramienta desarrollada para la Secretaría de Agricultura, Ganadería, Pesca y Alimentos de Argentina. Este sistema muestra una eficiente información geoespacial generada por las distintas áreas de esta secretaría. Asimismo el sistema reúne información documental, catálogo de cartografía georreferenciada, catálogo de imágenes satelitales y productos especiales disponibles para los usuarios.
- ✓ **SIG Móviles:** Es un tipo nuevo de SIG adaptado a pantallas pequeñas para poder usarse en teléfonos móviles o Smartphone¹², con un visor SIG de datos vectoriales, rásters y remotos. Es una herramienta para medir distancias y áreas, para obtener información de datos locales desde un teléfono celular, con un receptor GPS¹³, y presenta un sencillo editor de datos vectoriales.

Dada la gran diversidad que presentan los SIG en cuanto a sus características, formas de despliegue, tipos de plataformas, formas de uso, y facilidades nunca antes vistas, brindan la opción a los usuarios de elegir el más adecuado para darle solución al problema que se le plantee.

1.3.5. Evolución de los Sistemas de Información Geográfica en Cuba.

En los últimos años el tratamiento de la información geográfica ha tomado un auge vertiginoso a escala mundial, cada día con mayores posibilidades de aplicación gracias al desarrollo de los Sistemas de Información Geográfica. Cuba también está inmersa en el desarrollo de este tipo de herramientas y crea en el año 1962 el Instituto de Geografía Tropical, encargándose este de la investigación de disímiles ramas de la geografía con el objetivo de hacer un uso más racional de los recursos naturales de la geografía cubana, así como de la conservación del medio ambiente. En 1987 esta institución crea el SIG de CUBA (SIGC) con el objetivo de actualizar el atlas nacional de Cuba, este SIG tenía una estructura

¹² **Smartphone:** Teléfono inteligente.

¹³ **GPS (Global Positioning System):** Sistema de posicionamiento global.

bastante compleja pero a pesar de esto sus 15 módulos funcionaban a la perfección, obteniéndose muy buenos resultados en el Instituto. (8)

En los años noventa el antiguo Instituto Cubano de Hidrografía deseaba tener su propio Sistema de Información Geográfica y crea el TELEMAT con el objetivo de evitar la tediosa y agotadora digitalización. En realidad este SIG es un módulo para el análisis espacial y se denominó Telemat/GIS como parte de un “Software para las Geociencias”.

En la actualidad y debido a la creación de la Universidad de las Ciencias Informáticas, en este centro y gracias al trabajo en conjunto con la empresa GEOCUBA y las FAR, se ha ido avanzando de manera significativa en el desarrollo de los SIG, el trabajo de estas entidades se puede ver reflejado en la plataforma soberana GeneSIG la cual tiene como objetivo principal desarrollar aplicaciones o ser la base para personalizaciones aplicables a diversos negocios o sectores sociales.

El SIGUCI es una de estas personalizaciones basadas en la plataforma GeneSIG, con una muy buena aceptación por parte de los usuarios de la universidad.

1.3.6. Soluciones de Sistemas de Información Geográfica para universidades.

- ✓ **SIGUA:** Un sistema creado para la comunidad universitaria y para la gestión de espacios en universidades, desarrollada en la Universidad de Alicante (UA) dicho proyecto, denominado SIGUA se inició en 1997. Está en estos momentos firmemente integrada con otras herramientas de gestión corporativa, tales como: gestión docente y académica, infraestructuras, distribución de espacios, patrimonio e inventario, señalización, simulaciones, prevención, seguridad e higiene.

Este sistema presenta como características principales la de ser un SIG cuyo núcleo reside en una base de datos espaciales o geodatabase. Agrupa un conjunto de normativas, protocolos y directivas de actuación para la gestión de espacios y desde el punto de vista del usuario final, SIGUA es un portal de Internet donde se consumen los servicios ofertados y se muestran los servicios de mapa. Por la parte tecnológica el sistema presenta una base de datos creada en PostgreSQL, al que se ha añadido un comportamiento espacial a través de la extensión PostGIS, dispone de un interfaz Web donde se puede llevar a cabo las tareas de mantenimiento del mismo.

- ✓ **SIGUCI v.1.0 Beta:** El Sistema de Información Geográfica de la Universidad (SIGUCI v.1.0 Beta), Está concebido estructuralmente como un SIG Web, el cual garantiza la representación geoespacial de objetos de varias categorías de negocio del entorno de la UCI, como la ubicación de los lugares de interés, tanto social como económico, además de brindar servicios de búsqueda y geolocalización de personas e instalaciones. (9).

El SIGUCI es un producto resultante del primer proyecto de personalización sobre la plataforma de desarrollo de sistemas de información geográfica GeneSIG, desarrollado por el proyecto de igual nombre del Polo Productivo Geoinformática de la Facultad 9, integrado además por especialistas del grupo empresarial GEOCUBA y el Centro UCID.

1.4. Conclusiones Parciales.

En el actual capítulo se ha visto la definición de los principales términos propios de la presente investigación, para así lograr un mejor entendimiento del problema a resolver. Se profundizo además en la problemática que presenta el SIGUCI, la cual ha dado lugar al desarrollo de este trabajo, también se ha visto un resumido análisis de la evolución de los Sistemas de Información Geográficos en el ámbito nacional e internacional, y de algunas soluciones ya existente. De tal modo que se ha llegado a la conclusión de que es necesario el desarrollo de un sistema que solucione los problemas que existen en la primera versión del SIGUCI, usando el mismo como punto de partida para el desarrollo de la nueva versión.

Capítulo 2

Tendencias y Tecnologías.

2.1. Introducción.

La Informatización de la sociedad es el proceso de utilización ordenada y masiva de las Tecnologías de la Información y las Comunicaciones (TIC) en la vida cotidiana, para satisfacer las necesidades de todas las esferas de la sociedad, en su esfuerzo por lograr cada vez mayor eficiencia en todos los procesos y por consiguiente mayor generación de riqueza y aumento en la calidad de vida de los ciudadanos. (10)

En Cuba se ha identificado la necesidad de dominar e introducir las Tecnologías de la Informática y las Comunicaciones en la sociedad para así lograr una cultura informática, lo que facilitaría a la sociedad cubana acercarse más hacia el objetivo de un desarrollo sostenible. El uso de modelos basados en las nuevas tecnologías por las empresas posibilitan una mejora de los procesos productivos internos al proporcionar herramientas que facilitan la clasificación, organización y manejo de la información, mejorando los procesos de interacción con clientes en el exterior, proveedores y socios, abriendo la posibilidad de nuevos negocios. En este capítulo se evidencian ejemplos de las tecnologías que se usan diariamente en nuestra universidad para el desarrollo de software.

2.2. Software.

2.2.1. Software Libre.

El software libre engloba a toda aplicación informática que se puede usar, copiar, modificar y distribuir; por lo tanto, debe estar acompañado de su código fuente para hacer efectivas sus características:

- ✓ Ejecutar el programa sea cual sea el propósito.
- ✓ Estudiar el funcionamiento del programa y adaptarlo a necesidades personales.
- ✓ Redistribuir copias.
- ✓ Mejorar el programa y luego publicarlo para el bien de toda la comunidad.

Software libre es cualquier programa cuyos usuarios gocen de estas libertades. Richard Stallman comenta que el software libre es una cuestión de libertad, no de precio. Para comprender este concepto, se debe pensar en la acepción de libre como en “libertad de expresión”. En términos del citado autor el software

libre se refiere a la libertad de los usuarios para ejecutar, copiar, distribuir, estudiar, cambiar y mejorar el software.

2.3. Sistema Operativo.

Programa o conjunto de programas más importante de una computadora, es el encargado de gestionar los procesos básicos y de mayor importancia de un sistema informático, y permite la normal ejecución del resto de las operaciones. Un Sistema Operativo provee de una interfaz entre el usuario y el resto de los programas de la computadora y los dispositivos de hardware, presenta la función de administrar los recursos de la máquina, coordinar el hardware y organizar archivos y directorios en dispositivos de almacenamiento. Cada computadora de uso cotidiano y general debe tener un Sistema Operativo. Los Sistemas Operativos más utilizados son MS-Dos, Windows, Linux y Macintosh.

Los sistemas operativos pueden ser clasificados de la siguiente forma:

- ✓ **Multiusuario:** Permite que dos o más usuarios utilicen sus programas al mismo tiempo.
- ✓ **Multiprocesador:** Soporta el ejecutar un mismo programa en más de un procesador.
- ✓ **Multitarea:** Permite que varios programas se ejecuten al mismo tiempo.
- ✓ **Multitramo:** Permite que diversas partes de un solo programa funcionen al mismo tiempo.
- ✓ **Tiempo Real:** Responde a las entradas inmediatamente.

Existen varias familias del sistema operativo UNIX que han evolucionado de manera independiente a lo largo de los años, y dentro de las más exitosas se pueden mencionar las familias BSD5¹⁴ y GNU/Linux¹⁵, que son los sistemas operativos libres con mayor reconocimiento a nivel mundial, los más usados son aquellos que han sido desarrollados sobre la plataforma GNU/Linux. Numerosa es la lista de los sistemas operativos basados en GNU/Linux pero entre los más utilizados se encuentran Red Hat, Suse Linux, Knoppix, Gentoo, Fedora, Debian y Ubuntu.

Debian es una distribución libre para utilizar y compartir, bajo los términos especificados por la Licencia Pública General de GNU; y constituye además el Sistema Operativo más potente para la construcción de aplicaciones informáticas. Ubuntu es una de las más importantes distribuciones de GNU/Linux a nivel mundial. El sistema operativo Ubuntu se basa en Debian GNU/Linux y concentra su objetivo en la

¹⁴ **BSD5:** *Es un sistema operativo derivado del sistema Unix nacido a partir de los aportes realizados a ese sistema por la Universidad de California en Berkeley.*

¹⁵ **GNU/Linux:** *Sistema operativo formado por las herramientas de GNU y el núcleo Linux en conjunto.*

facilidad y libertad de uso, la fluida instalación y los lanzamientos regulares. Ubuntu posee distribuciones que se adecuan al sector al cual van dirigidas y posee además soporte para todo tipo de sistemas.

2.4. Aplicaciones Informáticas.

Las aplicaciones informáticas son los programas que permiten la interacción entre el usuario y la computadora, están diseñados como herramientas para permitir al usuario realizar uno o diversos tipos de trabajo, por lo cual resultan ser soluciones informáticas para automatizar tareas generalmente complicadas. Las aplicaciones informáticas se pueden dividir en dos grandes tipos:

- ✓ Aplicaciones de Escritorio.
- ✓ Aplicaciones Web.

Con el transcurso de los años y el uso de las aplicaciones informáticas se ha demostrado que estas han tenido un impacto significativo en la economía del país. El auge y la rápida expansión de Internet, han condicionado la necesidad de que toda la información que se maneja en esta, sea accesible desde cualquier lugar y en el momento requerido, es por ello que en la actualidad se está produciendo una enorme inclinación de las aplicaciones Web sobre las aplicaciones de escritorio.

2.4.1. Aplicación Web.

Una aplicación Web es una aplicación informática que los usuarios utilizan accediendo a un servidor Web a través de Internet o de una Intranet. Es un sitio que se encuentra en la red, al cual los usuarios acceden con el fin de beneficiarse con el uso de la información que posee, los usuarios pueden interactuar con esta información, actualizarla, modificarla o eliminarla, según los privilegios de seguridad que posean dentro de la aplicación. Las aplicaciones Web son populares debido a lo práctico del navegador web como cliente ligero, a que son independientes del Sistema Operativo, así como a la facilidad para actualizar y mantener aplicaciones Web sin distribuir e instalar ningún software a miles de usuarios.

Entre las principales ventajas de las aplicaciones Web se tiene:

- ✓ **Multiplataforma:** Con un sólo programa y un único ejecutable, nuestras aplicaciones pueden ser utilizadas a través de múltiples plataformas, tanto de hardware como de software.
- ✓ **Actualización instantánea:** Debido a que todos los usuarios de la aplicación hacen uso de un sólo programa que radica en el servidor, los usuarios siempre utilizarán la versión más actualizada del sistema.

- ✓ **Suave curva de aprendizaje:** Los usuarios, al utilizar la aplicación a través de un navegador, hacen uso del sistema tal como si estuvieran navegando por Internet, por lo cual su acceso es más intuitivo.
- ✓ **Fácil de integrar con otros sistemas:** Debido a que se basa en protocolos estándares, la información manejada por el sistema puede ser accedida con mayor facilidad por otros sistemas.
- ✓ **Acceso móvil:** El usuario puede acceder a la aplicación con la única restricción de que cuente con un acceso a la red privada de la organización o a Internet, dependiendo de las políticas de dicha organización; puede hacerlo desde una computadora de escritorio o desde una agenda electrónica; desde su oficina, hogar u otra parte del mundo.

2.5. Arquitectura de software.

La Arquitectura de Software es la organización fundamental de un sistema representada en sus componentes, las relaciones entre ellos, el ambiente, los principios que orientan su diseño y evolución (11).

Consiste en un conjunto de patrones y abstracciones coherentes que proporcionan el marco de referencia necesario para guiar y establecer los fundamentos de analistas, diseñadores y programadores en la construcción de un software para un sistema de información.

Según Kruchten Philippe¹⁶: “La arquitectura de software, tiene que ver con el diseño y la implementación de estructuras de software de alto nivel. Es el resultado de ensamblar un cierto número de elementos arquitectónicos de forma adecuada para satisfacer la mayor funcionalidad y requerimientos de desempeño de un sistema, así como requerimientos no funcionales, como la confiabilidad, escalabilidad, portabilidad, y disponibilidad.”

Esta ha evolucionado como una disciplina de gran importancia, su adecuada representación y diseño permite entender el sistema, organizar su desarrollo, plantear la reutilización del software y hacerlo evolucionar, convirtiéndose de esa manera en lo que se conoce como el esqueleto de soporte del sistema.

¹⁶ **Kruchten Philippe:** *Ingeniero de software canadiense, y profesor de Ingeniería de Software en la Universidad de British Columbia, conocido como Director de Proceso de Desarrollo (RUP) en Rational Software.*

2.5.1. Arquitectura Cliente-Servidor.

En el modelo cliente - servidor, el cliente envía un mensaje solicitando un determinado servicio a un servidor haciendo una petición, y este envía uno o varios mensajes con la respuesta de la solicitud. La separación entre cliente y servidor es una separación de tipo lógico, donde el servidor no se ejecuta necesariamente sobre una sola máquina ni es un sólo programa. El cliente permite al usuario formular los requerimientos y pasarlos al servidor, también maneja las funciones relacionadas con el despliegue y manipulación de datos.

El servidor, es el encargado de atender a múltiples clientes que hacen peticiones de algún recurso administrado por él y normalmente maneja todas las funciones relacionadas con la mayoría de las reglas del negocio y los recursos de datos. El cliente y el servidor pueden actuar como una sola entidad y también como entidades separadas, realizando actividades o tareas independientes, permitiendo que los cambios ocurridos en el servidor generalmente no impliquen cambios en el cliente.

2.5.2. Modelo-Vista-Controlador (MVC).

El Modelo-Vista-Controlador (MVC) es un patrón de arquitectura que separa en tres componentes distintos a los datos de una aplicación, la interfaz de usuario y la lógica de control. Es hoy en día muy difundido en el uso de aplicaciones Web.

El MVC tiene tres piezas claves:

- ✓ **El modelo:** Responsable de toda la lógica y estado del dominio de negocio.
- ✓ **La vista:** Responsable de la presentación del dominio de negocio. Es la interfaz de usuario.
- ✓ **El controlador:** Responsable del flujo de control, la navegabilidad y el estado de la entrada del usuario.

La arquitectura MVC separa la lógica de negocio (el modelo) y la presentación (la vista) por lo que se consigue un mantenimiento más sencillo de las aplicaciones. El ciclo de vida del patrón básicamente empieza cuando el usuario hace una solicitud al controlador con información sobre lo que desea realizar.

El controlador decide a quien debe delegar la tarea y es aquí donde el modelo empieza su trabajo. En esta etapa el modelo se encarga de realizar operaciones sobre la información que maneja para cumplir con lo que le solicita el controlador. Una vez que termina su trabajo, le regresa al controlador la información resultante de sus operaciones, el cual a su vez dirige a la vista. La vista se encarga de transformar los datos en información visualmente entendible al usuario.

El MVC cumple el fin particular de cualquier framework, presenta una estructura bien definida que da soporte a un proyecto web, también nos ayuda a que nuestro proyecto sea más organizado y bien desarrollado.

2.6. Web Service (*Servicio Web*).

Los servicios Web son sistemas de software que permiten el intercambio de datos y funcionalidades entre aplicaciones sobre una red, y la intercomunicación entre sistemas de cualquier plataforma. Se utilizan además en una gran variedad de escenarios de integración, tanto dentro de las organizaciones como fuera de ellas. Esta soportado en diferentes estándares que garantizan la interoperabilidad de los servicios. Los servicios Web utilizan como su gran insumo el lenguaje extensible de marcado XML¹⁷ y se basa en una arquitectura en la que se define el servicio Web a través de uno de los lenguajes estándar.

La arquitectura que mejor se ha adaptado al mundo de los servicios Web es SOA¹⁸ brindando un enfoque que ha adoptado los negocios y ha incrementado el intercambio electrónico de datos y el comercio electrónico. La World Wide Web Consortium¹⁹ (W3C) define un servicio Web como: "...un sistema de software diseñado para soportar interacción interoperable máquina a máquina sobre una red. Este tiene una interfaz descrita en un formato procesable por una máquina. Otros sistemas interactúan con el servicio Web en una manera prescrita por su descripción usando mensajes SOAP²⁰, típicamente enviados usando HTTP con una serialización XML en relación con otros estándares relacionados con la Web". (12)

Los servicios Web son muy prácticos ya que pueden aportar gran independencia entre la aplicación que usa el servicio Web y el propio servicio. De esta forma, los cambios a lo largo del tiempo en uno no deben afectar al otro. Esta flexibilidad será cada vez más importante, dado que la tendencia a construir grandes aplicaciones a partir de componentes distribuidos más pequeños es cada día más utilizada.

¹⁷ **XML:** *Metalinguaje extensible de etiquetas.*

¹⁸ **SOA:** *Concepto de arquitectura de software, define la utilización de servicios para dar soporte a un negocio.*

¹⁹ **W3C:** *Comunidad internacional que trabaja para desarrollar estándares Web. Liderado por Tim Berners-Lee.*

²⁰ **SOAP:** *Protocolo estándar que define cómo dos objetos en diferentes procesos pueden comunicarse.*

2.7. Servidores.

2.7.1. Servidor Web.

Un servidor Web es un programa que se ejecuta en el servidor e implementa el protocolo HTTP, éste escucha las peticiones del cliente y dependiendo de ellas buscará una página web o ejecutará un programa en el servidor. El servidor responde al cliente enviándole el código HTML de la página, cuando éste lo recibe, lo interpreta y lo muestra en pantalla.

2.7.1.1. Apache.

El Servidor Apache HTTP es un servidor Web con tecnología de Código Abierto, para plataformas Unix, Windows y Macintosh, entre otras, que implementan el protocolo HTTP / 1.1, es un servidor sólido y para uso comercial desarrollado por la Apache Software Foundation. Apache tiene amplia aceptación en la red y desde 1996 es el servidor HTTP más usado y es considerado el líder mundial en cuanto a servidores web se refiere. No posee una interfaz de usuario gráfica para su administración, sino un archivo de configuración llamado *httpd.conf*, es uno de los primeros servidores Web en soportar tanto host basados en IP como host virtuales. Este servidor Web posee además soporte de servlets²¹ de Java y tiene servidor proxy integrado.

2.7.2. Servidor de Mapas.

Los servidores de mapas tienen como objetivo proveer servicios a usuarios que deseen acceder a información geoespacial existente, en diferentes formatos y servir dicha información a clientes de mapas a través de protocolos estándares. Presenta Servicios de mapas (WMS), Servicios de geometrías (WFS) y Servicios de coberturas (WCS). Todos estos servicios suelen llevar asociados estándares de interoperabilidad especificados generalmente por el OGC (Open Geospatial Consortium).

2.7.2.1. MapServer.

Es un entorno de desarrollo en código abierto para la creación de aplicaciones SIG con el fin de visualizar, consultar y analizar información geográfica a través de la red mediante la tecnología Internet MapServer (IMS). Se ejecuta bajo plataformas Linux y Windows, soporta los formatos raster soportados: JPG, PNG, GIF, TIFF y su extensión GeoTIFF, EPPL7 y otros vía GDAL²², MapServer presenta funcionalidades las

²¹ **Servelets:** Programa que se ejecuta en un servidor.

²² **GDAL:** Biblioteca de software para la lectura y escritura de formatos de datos geoespaciales.

cuales se acceden mediante MapScript el cual proporciona una API²³ para lenguajes de programación como PHP, Java, Perl, Python y C#.

2.8. Sistema Gestor de Base de Datos.

Un Sistema Gestor de Bases de Datos (SGBD) es una colección de programas cuyo objetivo es servir de interfaz entre la base de datos, el usuario y las aplicaciones. Se compone de un lenguaje de definición de datos, de un lenguaje de manipulación de datos y de un lenguaje de consulta. Un SGBD permite definir los datos a distintos niveles de abstracción y manipularlos, garantizando la seguridad e integridad de los mismos. Un SGBD permite crear y mantener una base de datos, asegurando su integridad, confidencialidad y seguridad.

2.8.1. PostgreSQL.

Es un Sistema Gestor de Bases de Datos Relacionales Orientadas a Objetos. Con cerca de una década de desarrollo, PostgreSQL es el gestor de bases de datos de código abierto más avanzado hoy en día, ofreciendo control de concurrencia multi-versión, soportando casi toda la sintaxis SQL incluyendo subconsultas, transacciones, y tipos y funciones definidas por el usuario, contando también con un amplio conjunto de enlaces con lenguajes de programación (13).

PostgreSQL proporciona un gran número de características que normalmente sólo se encontraban en las bases de datos comerciales tales como DB215 u Oracle16.

- ✓ **PostGIS:** Es un módulo que añade soporte de objetos geográficos a la base de datos objeto-relacional PostgreSQL, convirtiéndola en una base de datos espacial para su utilización en Sistema de Información Geográfica. Se publica bajo la Licencia Pública General de GNU. En la actualidad hay muchos productos informáticos que utilizan PostGIS como apoyo para su base de datos espaciales. Entre estos se incluyen: Quantum GIS, MapServer, gvSIG, ArcGIS, GeneSIG.

²³ **API:** Conjunto de funciones y procedimientos o métodos, en la programación orientada a objetos que ofrece cierta biblioteca para ser utilizado por otro software.

2.9. Lenguajes y tecnologías de programación.

2.9.1. Lenguaje y tecnologías de programación del lado del cliente.

2.9.1.1. HTML: Hyper Text Markup Language (*Lenguaje de Mercado de Hipertexto*).

Es el lenguaje de marcado predominante para la construcción de páginas web. HTML se escribe en forma de "etiquetas", rodeadas por corchetes angulares (<,>).

HTML también puede describir, hasta un cierto punto, la apariencia de un documento, y puede incluir un script, por ejemplo Javascript, el cual puede afectar el comportamiento de navegadores web y otros procesadores de HTML. Es un lenguaje de composición de documentos y especificación de ligas de hipertexto que define la sintaxis y coloca instrucciones especiales que no muestra el navegador, aunque si indica como desplegar el contenido del documento, incluyendo texto, imágenes y otros medios soportados. HTML también indica cómo hacer un documento interactivo a través de ligas especiales de hipertexto, las cuales conectan diferentes documentos. (14).

2.9.1.2. JavaScript.

Es un lenguaje de programación interpretado, es decir, que no requiere compilación, que se utiliza principalmente para crear páginas Web dinámicas. Una página Web dinámica es aquella que incorpora efectos como aparición y desaparición de texto, animaciones, acciones que se activan al pulsar botones u otros elementos y ventanas con mensajes de aviso al usuario. JavaScript es un lenguaje orientado a objetos que se puede incluir en cualquier documento HTML. Todos los navegadores modernos interpretan el código JavaScript integrado dentro de las páginas Web.

✓ **Ventajas de JavaScript:** (15)

- **Fácil de aprender, rápido y potente:** JavaScript es muy sencillo de aprender, se puede empezar a trabajar con él desde el principio. Es ideal para agregar ciertas funciones rápidas a una página Web. Una vez que se conocen las bases del lenguaje. Aunque suele ser un lenguaje muy potente y de alto nivel.
- **Alta usabilidad:** JavaScript es, con diferencia, el lenguaje de programación que más se utiliza en la Web. Hay publicadas millones de páginas Web que incorporan elementos que lo usan.

- **Reducción de la carga del servidor:** JavaScript se puede hacer cargo de gran parte de las funciones del cliente de las cuales se encargaba el servidor. Uno de los mejores ejemplos es la validación.
- ✓ **Ext JS:** Es una biblioteca de JavaScript utilizada para el desarrollo de aplicaciones web interactivas usando tecnologías como AJAX, DHTML²⁴ y DOM²⁵. Puede usarse como extensión para las bibliotecas jQuery y Prototype. Desde la versión 1.1 puede ejecutarse como una aplicación independiente. Dispone de un conjunto de componentes para incluir dentro de una aplicación Web, como cuadros y áreas de texto, campos para fechas, campos numéricos, radiobuttons²⁶ y checkboxes²⁷, editor HTML, elementos de datos, pestañas, barra de herramientas, menús al estilo de Windows, entre otros.

2.9.1.3. AJAX: Asynchronous Java Script and XML.

Es una técnica de desarrollo web para crear aplicaciones interactivas las cuales se ejecutan en el cliente, en el navegador de los usuarios mientras se mantiene la comunicación asíncrona con el servidor en segundo plano. Esto significa que es posible realizar cambios sobre las páginas sin necesidad de recargarlas, esto aumenta la interactividad, velocidad y usabilidad en las aplicaciones. AJAX es una combinación de cuatro tecnologías: XHTML, DOM, XML, y el objeto XMLHttpRequest. Las páginas con AJAX son más difíciles de desarrollar que las páginas estáticas, un sitio con AJAX usa más recursos en el servidor y hay que tener en cuenta que AJAX dependerá de las características que el navegador presente ya que no es permitido en algunos.

²⁴ **DHTML:** *HTML Dinámico, conjunto de técnicas que permiten crear sitios web interactivos.*

²⁵ **DOM:** *Es una interfaz de programación de aplicaciones (API) para documentos HTML y XML.*

²⁶ **Radiobuttons:** *Elemento de la interfaz gráfica de usuario que permite al usuario elegir sólo uno de un conjunto predefinido de opciones.*

²⁷ **Checkboxes:** *Elemento de la interfaz gráfica de usuario que permite al usuario hacer selecciones múltiples de un conjunto de opciones.*

2.9.2. Lenguaje y tecnologías de programación del lado del servidor.

2.9.2.1. PHP: Hypertext Pre-processor.

Es un lenguaje de programación interpretado, creado para el desarrollo de páginas Web dinámicas de alto nivel embebido en páginas HTML y ejecutado en el servidor, publicado bajo la PHP License, la fundación de software libre considera esta licencia como software libre. Puede interactuar con los servidores Web más populares ya que existe en versión CGI²⁸. Existe un Entorno de Desarrollo Integrado (IDE) comercial llamado Zend Studio desarrollado solo para PHP. También existen al menos un par de módulos para Eclipse, uno de los IDE más populares a nivel mundial.

✓ Principales características de PHP

- **Fácil de usar:** PHP es un lenguaje muy fácil de aprender con respecto a otros lenguajes utilizados para el mismo propósito.
- **Embebido en HTML:** Las páginas escritas en PHP son simples páginas en HTML que contienen, además de las etiquetas normales, el programa que queremos ejecutar.
- **Multiplataforma:** PHP se ejecuta en multitud de plataformas, Sistemas Operativos y Servidores existentes.
- **Licencia Open Source:** La licencia de Código Abierto implica que el código fuente de PHP es libre de ser descargado y completamente inspeccionado.
- **Multitud de Extensiones:** PHP se desarrolla para dar la mayor versatilidad y flexibilidad a los usuarios que lo utilizan.
- **Velocidad e incorporación de objetos:** El nuevo motor Zend 2.0 acelera los procesos de ejecución del código. Además, incorpora un nuevo modelo de objetos que permite crear clases y métodos privados, protegidos y públicos, clases abstractas e interfaces.

2.9.2.2. Framework (Entorno de trabajo).

Un framework simplifica el desarrollo de una aplicación mediante la automatización de algunos de los patrones utilizados para resolver las tareas comunes. Además, un framework proporciona estructura al código fuente, forzando al desarrollador a crear código más legible y más fácil de mantener. Por último, un

²⁸ CGI: Tecnología que se usa en los servidores web.

framework facilita la programación de aplicaciones, ya que encapsula operaciones complejas en instrucciones sencillas. (16)

- ✓ **CartoWeb:** CartoWeb es una herramienta SIG Web, que incluye funcionalidades muy robustas sin perder sus requerimientos de ser una aplicación modular²⁹ y escalable³⁰. Brinda un framework para que los desarrolladores puedan realiza aplicaciones avanzadas o personalizaciones de algunas aplicaciones ya desarrolladas. Es distribuido bajo licencia GNU GPL y su componente fundamental es MapServer. Presenta funcionalidades tales como: Despliegue de mapas, herramientas de navegación del mapa como zoom y paneo, árbol de capas, herramientas de geoinformación, herramientas de dibujo como puntos, líneas y polígonos, herramientas de medición de distancias y áreas, soporte para autenticar usuarios, entre otros. Para el usuario final ofrece funcionalidades básicas, pero la arquitectura de CartoWeb es la que hace a esta aplicación realmente modular y personalizada.

2.10. XML: Extensive Markup Language (*Lenguaje Extensible de Marcas*).

XML es un Lenguaje de Etiquetado Extensible muy simple, pero estricto, que juega un papel fundamental en el intercambio de una gran variedad de datos, desarrollado por el W3C. Es un lenguaje muy similar a HTML pero su función principal es describir datos y no mostrarlos como es el caso de HTML. XML es un formato que permite la lectura de datos a través de diferentes aplicaciones. Las tecnologías XML son un conjunto de módulos que ofrecen servicios útiles a las demandas más frecuentes por parte de los usuarios. XML sirve para estructurar, almacenar e intercambiar información.

✓ Principales características de XML:

- Integración de los datos de las fuentes más dispares. Puede hacer el intercambio de documentos entre las aplicaciones tanto en la propia PC como en una red local o extensa.
- Datos compuestos de múltiples aplicaciones. La extensibilidad y flexibilidad de este lenguaje permite agrupar una variedad amplia de aplicaciones, desde páginas Web hasta bases de datos.
- Gestión y manipulación de los datos desde el propio cliente Web.

²⁹ **Aplicación modular:** *Paradigma de programación que consiste en dividir un programa en módulos o subprogramas con el fin de hacerlo más legible y manejable.*

³⁰ **Aplicación escalable:** *Aplicación con la habilidad para extender el margen de operaciones sin perder calidad.*

- Se desarrollan de manera extensible las búsquedas personalizadas y subjetivas. También conlleva a que los clientes Web puedan ser más autónomos para desarrollar tareas que actualmente se ejecutan en el servidor.

2.11. IDE: Integrated Development Environment (*Entornos Integrados de Desarrollo*).

Un IDE es un entorno de programación que ha sido empaquetado como un programa de aplicación, es decir, consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica. Estos entornos pueden ser parte de una aplicación o pueden ser aplicaciones independientes. Los IDE proveen un marco de trabajo amigable para la mayoría de los lenguajes de programación.

2.11.1. Eclipse.

Eclipse es un IDE de código abierto multiplataforma para desarrollar lo que el proyecto llama "Aplicaciones de Cliente Enriquecido", opuesto a las aplicaciones "Cliente-liviano" basadas en navegadores. Esta plataforma, típicamente ha sido usada para desarrollar IDEs como el IDE de Java llamado *Java Development Toolkit* (JDT) y el compilador ECJ. Eclipse dispone de un editor de texto con resaltado de sintaxis. La compilación es en tiempo real. Tiene control de versiones con CVS³¹, asistentes (*wizards*) para creación de proyectos, clases, test, y refactorización.

2.12. Lenguaje Unificado de Modelado (UML).

Lenguaje Unificado de Modelado (UML) es un lenguaje que permite modelar, construir y documentar los elementos que forman un sistema de software orientado a objetos. UML es un estándar de la industria del software, pero no solo de la industria, sino de cualquier industria que requiera la construcción de modelos como condición previa para el diseño y posterior construcción de prototipos. (17)

UML ayuda a los usuarios a entender la realidad desde un punto de vista de la tecnología y la posibilidad de que reflexione antes de invertir y gastar grandes cantidades de dinero en proyectos que no estén seguros en su desarrollo, reduciendo el costo y el tiempo empleado en la construcción de los módulos que construirán el software. (18)

Por lo planteado anteriormente se decide aplicar este lenguaje de modelado en la vida del proyecto, ya que con la modelación de una serie de artefactos durante las primeras fases de ciclo de vida del software a desarrollar se brinda la posibilidad a que los implementadores tengan un mayor dominio y mejor

³¹ **CVS:** *Aplicación informática que implementa un sistema de control de versiones que mantiene el registro de todo el trabajo y los cambios en los ficheros.*

comprensión sobre qué es lo que se debe implementar en fases posteriores, además le permite al cliente y a los desarrolladores tener una representación real del alcance y la factibilidad que puede o no llegar a tener el producto.

2.13. Metodología de desarrollo de Software.

“Las metodologías de desarrollo de software definen quién está haciendo qué, cuándo y cómo para alcanzar un determinado objetivo”. (19)

Para obtener un producto final de calidad, y que se desarrolle en el tiempo establecido es necesario la aplicación de una de estas metodologías. Antes que todo se debe realizar un análisis de las características de cada proyecto a desarrollar para determinar cuál de las metodologías es la más factible para evitar insatisfacciones en el producto final.

Existen varios tipos de metodologías tales como: RUP, XP y SCRUM. Por la experiencia acumulada, su aplicación en el desarrollo de la plataforma GeneSIG base de los SIGs desarrollados en la universidad y particularmente en el desarrollo de la primera versión del SIGUCI, se dará continuidad a la utilización de RUP en esta segunda versión ya que esta metodología brinda buenos resultados y un desarrollo efectivo del software.

2.13.1. RUP: Rational Unified Process (*Proceso Unificado de Desarrollo*).

Es una metodología para la ingeniería de software que va más allá del mero análisis y diseño orientado a objetos para proporcionar una familia de técnicas que soportan el ciclo completo de desarrollo de software. El resultado es un proceso basado en componentes, dirigido por los casos de uso, centrado en la arquitectura, iterativo e incremental. (20)

Sus características permiten que esta metodología sea adaptable a una gran variedad de sistemas para diferentes áreas de aplicación, diferentes tipos de organización y diferentes tamaños de proyecto. La particularidad de que cada ciclo de iteración exige el uso de artefactos, este es el motivo que hace que sea una de las metodologías más importantes para alcanzar un alto grado de certificación en el desarrollo de un software.

✓ **El ciclo de vida de RUP:**

- **Dirigido por casos de uso (CU):** Los casos de uso reflejan lo que los usuarios futuros necesitan y desean, lo cual se capta cuando se modela el negocio y se representa a través de

los requerimientos. A partir de aquí los casos de uso guían el proceso de desarrollo ya que los modelos que se obtienen, como resultado de los diferentes flujos de trabajo, representan la realización de los casos de uso.

- **Centrado en la arquitectura:** La arquitectura muestra la visión común del sistema completo en la que el equipo de proyecto y los usuarios deben estar de acuerdo, por lo que describe los elementos del modelo que son más importantes para su construcción, los cimientos del sistema que son necesarios como base para comprenderlo, desarrollarlo y producirlo económicamente. RUP se desarrolla mediante iteraciones, comenzando por los CU relevantes desde el punto de vista de la arquitectura.
- **Iterativo e Incremental:** Aunque pueda parecer que los flujos de trabajo se desarrollan en cascada, RUP propone que cada fase se desarrolle en iteraciones. Una iteración involucra actividades de todos los flujos de trabajo, aunque desarrolla fundamentalmente algunos más que otros. Por ejemplo, una iteración de elaboración centra su atención en el análisis y diseño, aunque refina los requerimientos y obtiene un producto con un determinado nivel, pero que irá creciendo incrementalmente en cada iteración.

RUP divide su ciclo de vida en cuatro fases (*Fig. 1*) y nueve flujos de trabajo, de ellos seis de ingeniería y tres de soporte, tal como se muestra en la siguiente figura.

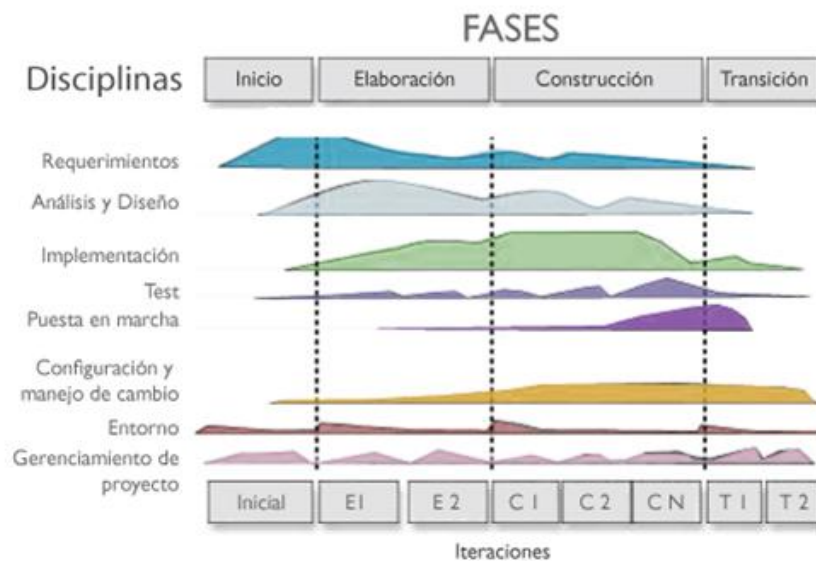


Fig. 1 Fases y flujos de trabajo de RUP.

✓ **Fases de RUP**

- **Inicio:** Establece la visión y el alcance del proyecto, y las partes interesadas deben realizar la estimación de tiempo y costo.
- **Elaboración:** Se analiza el dominio del problema para establecer una arquitectura base sólida para el desarrollo exitoso del proyecto.
- **Construcción:** Tiene como objetivo llegar a obtener la capacidad operacional inicial, así como desarrollar y probar el producto.
- **Transición:** Su objetivo fundamental es llegar a obtener la liberación o release del proyecto.

✓ **Flujos de Trabajo de Ingeniería**

- **Modelamiento del negocio:** Se realiza el entendimiento entre clientes y desarrolladores para concebir las necesidades del negocio que serán abarcadas.
- **Requerimientos:** Se realiza el acuerdo entre desarrolladores y clientes de lo que el sistema necesariamente debe hacer.
- **Análisis y Diseño:** Se realiza una descripción de cómo se implementará el sistema, centrándose en la noción que se tiene de la arquitectura.
- **Implementación:** Se crea el software, ajustándolo a la arquitectura y asegurando que tenga el comportamiento deseado.
- **Pruebas:** Se realizan las pruebas que aseguran que el comportamiento requerido es el correcto y que todo lo solicitado está presente.
- **Despliegue:** El producto final se hace llegar a sus usuarios finales.

✓ **Flujos de Trabajo de Soporte:**

- **Administración de configuración y cambios:** Controla los artefactos producidos y las versiones del producto.
- **Administración de proyectos:** Se manejan los riesgos y administran horarios y recursos.
- **Ambiente:** Se brinda una guía en la configuración de un ambiente apropiado para cada proyecto.

2.14. Herramienta CASE.

Las herramientas CASE³² son diversas aplicaciones informáticas destinadas a aumentar la productividad en el desarrollo de software reduciendo el coste de las mismas en términos de tiempo y de dinero. Estas herramientas nos pueden ayudar en todos los aspectos del ciclo de vida de desarrollo del software en tareas como el proceso de realizar un diseño del proyecto, cálculo de costes, implementación de parte del código automáticamente con el diseño dado, compilación automática, documentación o detección de errores entre otras.

2.14.1. Visual Paradigm.

Visual Paradigm es una herramienta CASE de diseño, para lenguaje UML, la que está diseñada para la ayuda a la hora de desarrollar un software. Esta herramienta CASE soporta estándares de la industria del desarrollo de software, tales como UML, SysML, Business Process Modeling Notation (BPMN), entre otros. Ofrece un conjunto de herramientas de ayuda a los equipos de desarrollo de software las cuales son necesarias para capturar los requisitos del sistema a desarrollar, para la planificación de programas, la planificación de controles, la clase de modelado, y modelado de datos entre otros. Ofrece generación de informes e incluye la generación de código. Es muy sencillo de usar y fácil de instalar. Soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue.

2.15. Conclusiones Parciales.

Con el objetivo de dar respuestas a los problemas planteados, se ha planteado la necesidad de desarrollar una nueva versión del SIGUCI. En este capítulo se realizó un nuevo análisis de las tecnologías para el desarrollo de la versión 2.0. Se seleccionó a Ubuntu en su distribución 10.10 como sistema operativo donde se desarrollará la solución. El servidor Web escogido fue Apache y de mapas a MapServer, como SGBD se escogió PostgreSQL 8.4 con el módulo PostGis para el trabajo con objetos espaciales. El lenguaje de programación de lado servidor que permitirá la implementación del sistema, continúa siendo PHP y por parte del cliente se continuará utilizando JavaScript con su librería ExtJS para darle un mayor dinamismo y atracción a las interfaces, para facilitar el entendimiento en lo que a la manipulación de los mapas y funcionalidades sobre los mismos se utilizará el framework CartoWeb, la metodología que guiará todo el proceso del software será RUP, y el lenguaje que permitirá el modelado del sistema, UML.

³² **CASE (Computer Aided Software Engineering):** *Ingeniería de Software Asistida por Computadora.*

Capítulo 3

Presentación de la solución propuesta.

3.1. Introducción.

En el presente capítulo se realizará el análisis de la modelación de los procesos que forman parte del modelo del dominio, también se definen los requisitos funcionales y no funcionales que el sistema debe cumplir así como la descripción del sistema que se tiene propuesto y se da una descripción textual de los casos de uso del sistema y se muestra como quedaría el diagrama del mismo.

3.2. Entorno donde trabajará el sistema.

3.2.1. Diagrama de clases del Modelo de Dominio.

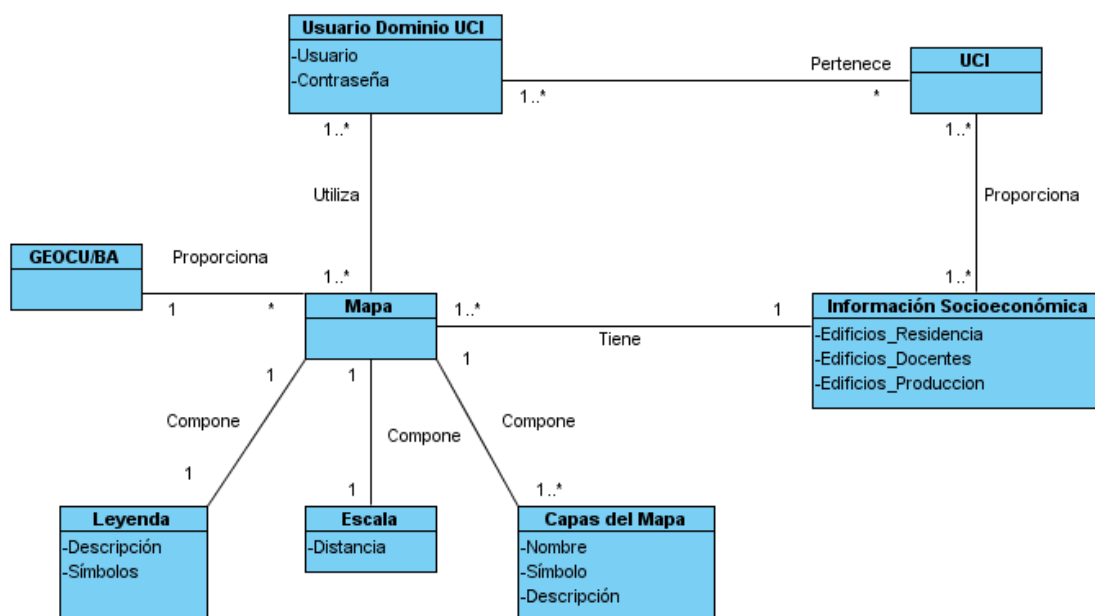


Fig. 2 Modelo del Dominio.

Descripción del diagrama: Los mapas se componen por la leyenda, la escala y las capas del mapa que no son más que aquellos fragmentos de mapas que conforman el mapa general. Estos mapas poseen la información socioeconómica, la cual es proporcionada por la UCI. Los mapas a utilizar los crea GEOCUBA, empresa que proporciona la cartografía para el desarrollo del SIGUCI.

✓ **Glosario de términos del diagrama del modelo del dominio.**

- **GEOCUBA:** Es un grupo empresarial que se dedica a la elaboración, producción y venta de planos, mapas y cartas náuticas con diversos fines, así como a la realización de Estudios Geográficos, de Impacto Ambiental, e investigaciones científicas en ramas del campo de las geociencias, entregando a sus clientes, productos informativos terminados con una alta calidad y fiabilidad.
- **Usuario dominio uci:** Persona que habita en las instalaciones de la Universidad de las Ciencias Informáticas y que necesite trabajar o consultar algún tipo de información.
- **UCI:** La universidad es la encargada de solicitar un servicio determinado utilizando un mapa y que proporcione la información socioeconómica referente a la misma.
- **Mapa:** Es una representación gráfica y métrica de una porción de territorio sobre una superficie bidimensional, generalmente plana, pero que puede ser también esférica.
- **Escala:** Relación entre la distancia que separa dos puntos en un mapa y la distancia real de esos dos puntos en la superficie terrestre.
- **Leyenda:** Explicación de los símbolos, colores, tramas y los sombreados empleados en un mapa; suele encontrarse a pie de página o en un recuadro, situado en sus márgenes o en su dorso.
- **Capas del mapa:** Mapas que serán utilizados como capas para la formación del mapa topográfico a utilizar por el sistema.
- **Información socioeconómica:** Es un conjunto organizado de datos procesados referentes al aspecto social y económico de cualquier lugar de interés del país en este caso de la UCI.

3.3. Requisitos Funcionales.

3.3.1. Localizar Ubicación por IP.

Nombre	RF 1 Localizar Ubicación por IP.
Asociados	CUS Autenticar.
Descripción	Con este requerimiento se desea que el usuario pueda ubicar en el mapa el lugar desde donde se está usando el SIGUCI.
Seguimiento	CUS Localizar Ubicación por IP.
Estado	Aprobado
Prioridad	No Crítico

Tabla 1 Requisito Funcional Localizar Ubicación por IP.

3.3.2. Tematizar.

Nombre	RF 2 Tematizar.
Asociados	CUS Autenticar.
Descripción	Con este requerimiento se quiere que el usuario pueda crear un mapa temático entrando los valores asociados a los Criterios de análisis: “Edificios por Facultad”, “Edificios por Género”, “Edificios por Tipo de Residencia”.
Seguimiento	CUS Tematizar.
Estado	Aprobado.
Prioridad	Crítico

Tabla 2 Requisito Funcional Tematizar.

3.3.3. Gestionar Datos.

Nombre	RF 3 Gestionar Datos.
Asociados	CUS Autenticar.
Descripción	Con este requerimiento se desea que el usuario con permiso de administración pueda agregar, eliminar y actualizar la información socioeconómica.
Seguimiento	CUS Gestionar Datos.
Estado	Aprobado.
Prioridad	Crítico

Tabla 3 Requisito Funcional Gestionar Datos.

3.3.4. Localizar IP.

Nombre	RF 4 Localizar IP.
Asociados	CUS Autenticar.
Descripción	Con este requerimiento se desea que el usuario pueda mediante un número de IP conocer la información referente al lugar al que pertenece el mismo y su ubicación en el mapa.
Seguimiento	CUS Localizar IP.
Estado	Aprobado.
Prioridad	No Crítico.

Tabla 4 Requisito Funcional Localizar IP.

3.3.5. Autenticar.

Nombre	RF 5 Autenticar.
Asociados	
Descripción	Con este requerimiento se quiere que el usuario pueda acceder mediante su contraseña del dominio UCI al sistema.
Seguimiento	CUS Autenticar.
Estado	Aprobado.
Prioridad	Crítico

Tabla 5 Requisito Funcional Autenticar.

3.3.6. Mostrar Ayuda.

Nombre	RF 6 Mostrar Ayuda.
Asociados	CUS Autenticar.
Descripción	Con este requerimiento se quiere que el usuario pueda conocer el sistema y cómo utilizarlo, además de tener la posibilidad de conocer términos propios de la geoinformática.
Seguimiento	CUS Ayuda.
Estado	Aprobado.
Prioridad	No Crítico

Tabla 6 Requisito Funcional Mostrar Ayuda.

3.3.7. Identificar Estructura.

Nombre	RF 7 Identificar Estructura.
Asociados	CUS Autenticar.
Descripción	Con este requerimiento se quiere que el usuario pueda conocer la información de las estructuras referenciadas en el sistema.
Seguimiento	CUS Identificar Estructura.
Estado	Aprobado.
Prioridad	Crítico

Tabla 7 Requisito Funcional Identificar Estructura.

3.4. Requisitos no funcionales.

No son más que las propiedades y características que el sistema debe tener. A continuación se enumerarán las mismas.

✓ Usabilidad.

- El sistema podrá ser usado por el personal de la universidad con conocimientos básicos en el manejo de computadoras. Se emplearán componentes que indiquen al usuario el estado de los procesos que por su complejidad requieran de un tiempo de procesamiento apreciable.
- El software presentará la opción de Ayuda, lo que posibilitará un mejor aprovechamiento por parte de los usuarios de las funcionalidades del sistema.

✓ Fiabilidad.

- Disponibilidad: El SIGUCI estará disponible las 24 horas del día para su uso por el personal de la universidad. Se dispondrá de un equipo de profesionales para su mantenimiento, soporte y recuperación ante fallos y errores. La información manejada por el sistema estará protegida de acceso no autorizado y divulgación.
- Debido a la arquitectura que presenta el sistema, siendo más robusto al no tratarse de un sistema de gestión que requiera mantenimiento y optimización en el almacenamiento, se estima un tiempo promedio de 6 meses entre posibles fallas.

- El tiempo medio de reparación depende de la magnitud con que sea el fallo.
- ✓ **Eficiencia.**
- El tiempo de respuesta estará dado por la cantidad de información a procesar, y la cantidad de veces que se tenga que dibujar un nuevo mapa a la hora de hacer alguna consulta. Si el sistema presenta 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 el sistema.
- ✓ **Soporte.**
- El SIGUCI recibirá mantenimiento cada vez que el cliente sienta la necesidad de desarrollar una nueva versión o cada vez que el sistema presente un problema de funcionamiento.
- ✓ **Restricciones de diseño.**
- El SIG de la UCI debe presentar un diseño sencillo, con una iconografía e interfaz amigable y llamativa, que el usuario se sienta a gusto con el sistema, sus ventanas deben presentar pocas entradas, donde no sea necesario mucho entrenamiento para utilizar el SIGUCI.
 - El producto de software final debe diseñarse sobre una arquitectura cliente-servidor.
 - Se deben emplear los estándares establecidos por la OGC.
- ✓ **Requisitos para la documentación de usuarios en línea y ayuda del sistema.**
- El sistema tendrá siempre la posibilidad de una ayuda, la cual servirá como guía a los usuarios que se dispongan a usar el SIGUCI por primera vez, esta ayuda siempre estará disponible y visible en la barra superior, lo que le permitirá un avance considerable en la explotación de la aplicación y sus funcionalidades.
- ✓ **Componentes Comprados.**
- La Universidad no se ha visto en la necesidad de comprar ningún componente o software para el desarrollo del SIGUCI.

✓ **Interfaz.**

- Interfaces de usuario

El sistema debe:

- Tener una apariencia profesional y un diseño gráfico sencillo y atractivo, ser intuitivo, presentar los colores predominantes de la intranet ya que el SIGUCI forma parte de este portal.

- Interfaces de hardware

Para las PCs clientes:

- Se requiere tengan tarjeta de red.
- Al menos 128 MB de memoria RAM.
- Se requiere al menos 10 GB de disco duro.
- Procesador 1.90 MHz como mínimo.

Para los servidores:

- Se requiere tarjeta de red.
- El Servidor de Mapas tenga como mínimo 4 GB de RAM y 40 GB de disco duro.
- El Servidor de BD tenga como mínimo 4 GB de RAM y 80 GB de disco duro.
- Procesador 3 GHz como mínimo para cada uno de los servidores.

✓ **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 versión 3.6, Zafari u otro navegador que cumpla con los estándares de la W3C.
- Sistema operativo: GNU/Linux, Windows o Mac OS.

Para los Servidores:

- Sistemas operativos GNU/Linux.
- Servidor Web Apache 2.0 o superior, con módulo PHP 5 configurado con la extensión pgsql incluida.
- PostgreSQL 8.4 como Sistema Gestor de Base de Datos.

- PostGis como extensión de PostgreSQL para el soporte de datos espaciales.
- MapServer 5.2.2 o superior, con extensión PHP mapscript.

✓ **Interfaces de comunicación.**

El SIGUCI garantizará mediante su interfaz la configuración del entorno de trabajo mediante funcionalidades propias como ocultar y mostrar paneles, marcar y desmarcar las capas deseadas.

✓ **Requisitos de licencia.**

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

✓ **Requisitos Legales, de Derecho de Autor y otros.**

- ✓ El sistema debe ajustarse y regirse por la ley, decretos leyes, decretos, 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, el SIGUCI se distribuye amparado bajo las normativas legales establecidas en el registro comercial emitido por las entidades jurídicas de la Universidad de las Ciencias Informáticas.

✓ **Estándares Aplicables.**

El sistema será desarrollado bajo estándares OpenGIS como aseguramiento de la parte científica y en el desarrollo se codificará y modelará siguiendo los patrones de las normativas ISO, tanto de codificación como de diseño de bases de datos.

3.5. Descripción del sistema propuesto.

3.5.1. Descripción de los Actores del Sistema.

Actor	Descripción
Usuario	Personas que trabaja y estudian en instalaciones de la Universidad y que necesite consultar algún tipo de información incluida en un mapa del sistema.
Administrador	Personal especializado, encargado de mantener actualizada la información manejada por el sistema.

Tabla 8 Descripción de los Actores del Sistema.

Diagrama de Casos de Uso del Sistema.

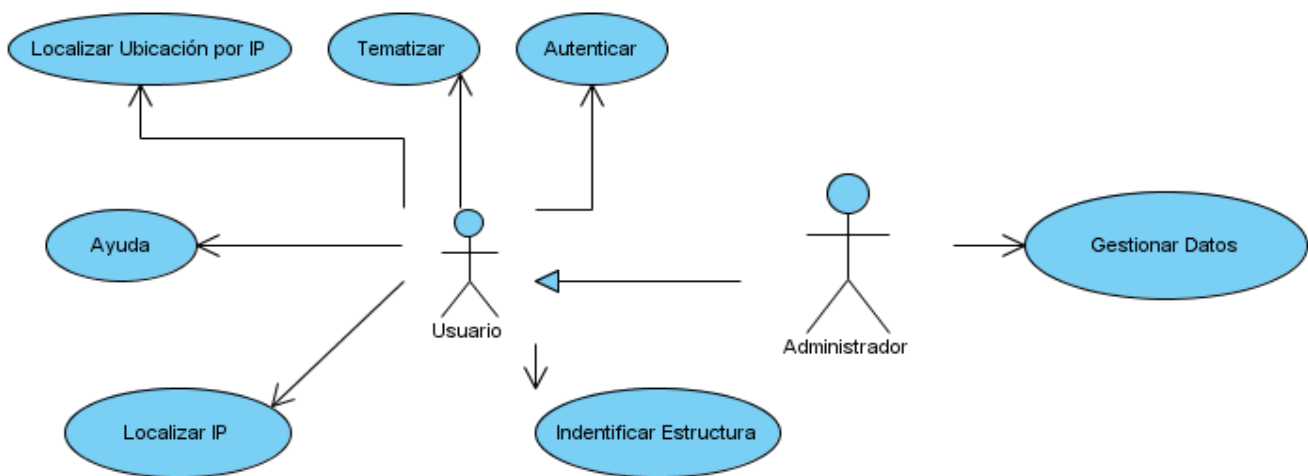


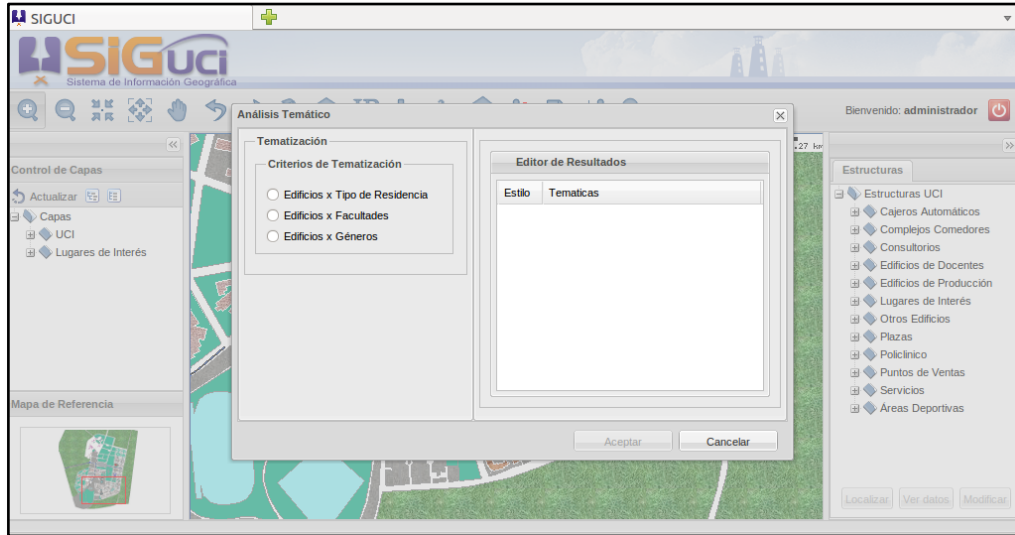
Fig. 3 Diagrama de Casos de Uso.

3.5.2. Descripción textual de los Casos de Uso del Sistema.

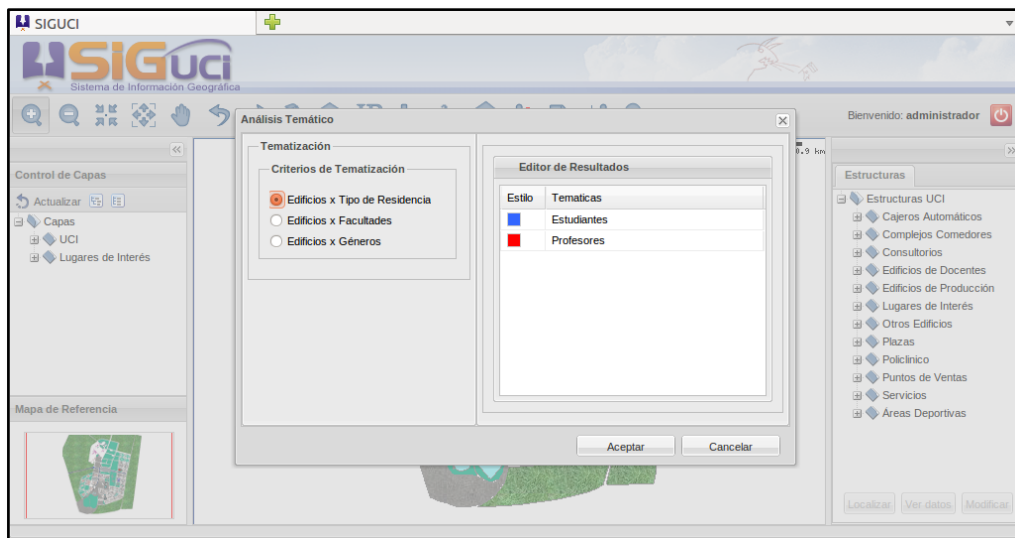
3.5.2.1. Tematizar.

Caso de Uso:	Tematizar.
Actores:	Usuarios, Administrador.
Propósito	Este caso de uso se lleva a cabo con el propósito de poder visualizar un mapa temático según la opción de análisis que desee el usuario.
Resumen:	El caso de uso se inicia cuando el usuario desea realizar una tematización de una parte del mapa de la universidad correspondiente a la opción de análisis, y termina visualizándose el mapa temático.
Precondiciones:	El usuario debe estar autenticado en el sistema.
Referencias	RF 2.
Prioridad	Crítico
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. El usuario selecciona la opción de Tematizar.	2. El sistema muestra la ventana “Análisis Temático”. (Fig.1)
3. El usuario selecciona el criterio de Tematización deseado.	4. El sistema llena el editor de resultados en la parte derecha según el criterio seleccionado. (Fig.2)
5. El usuario acepta la Tematización.	6. El sistema procesa la información y muestra en el mapa la tematización. (Fig.3)

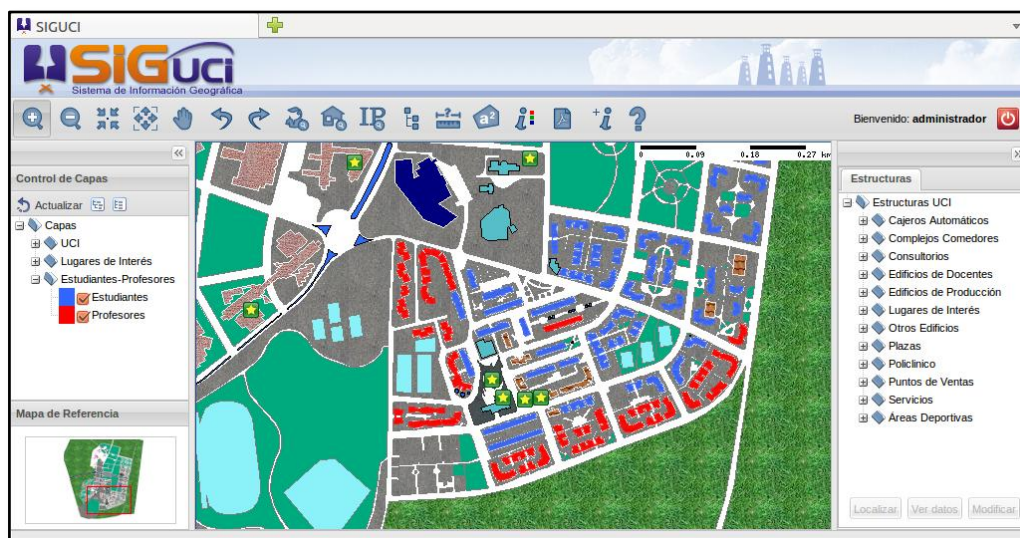
Prototipo de Interfaz



Prototipo de Interfaz Fig. 1



Prototipo de Interfaz Fig. 2



Prototipo de Interfaz Fig. 3

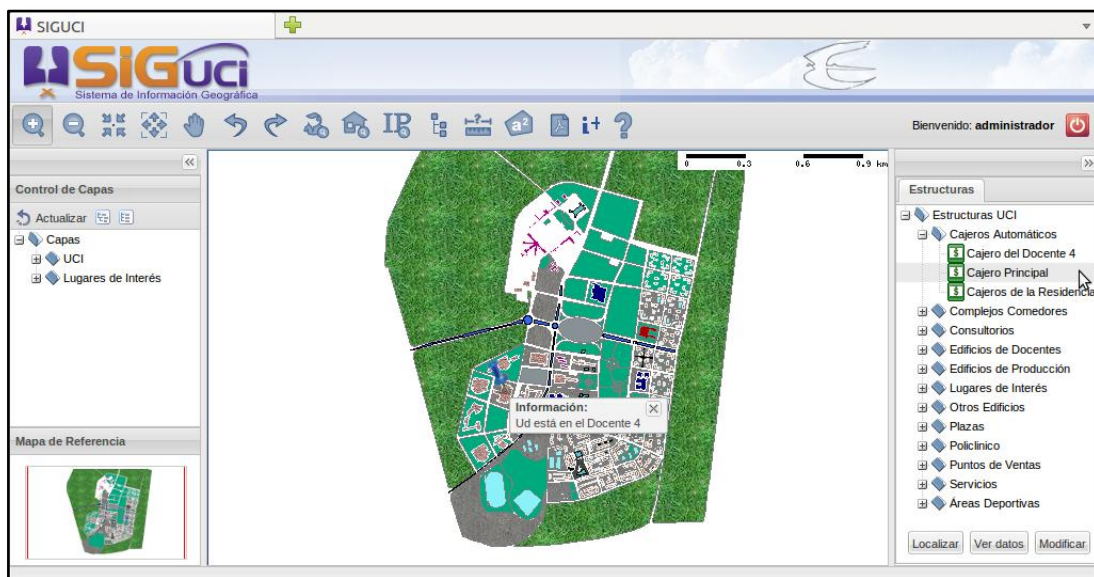
Tabla 9 Descripción textual Caso de Uso Tematizar.

3.5.2.2. Gestionar Datos.

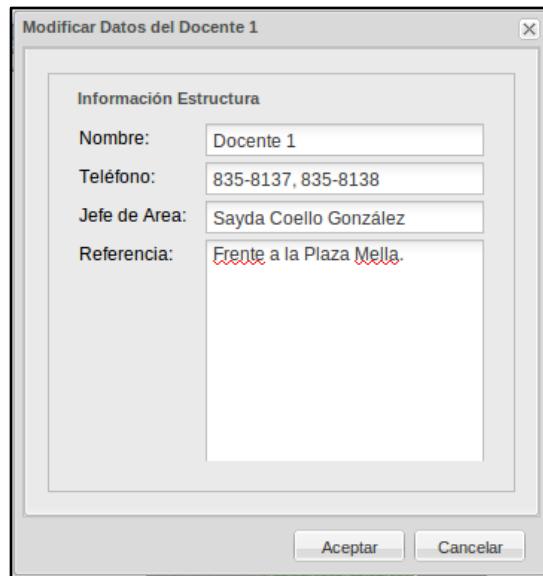
Caso de Uso:	Gestionar Datos.
Actores:	Administrador.
Propósito	Con este caso de uso se pretende que el administrador pueda modificar los campos de la base de datos socioeconómica.
Resumen:	El caso de uso se inicia cuando el administrador se autentica en el sistema para modificar algún dato y termina cuando el sistema muestra una ventana que informa que la modificación ha sido satisfactoria.
Precondiciones:	El usuario deberá autenticarse en el sistema como administrador.
Referencias	RF 3.
Prioridad	Crítico
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema

1. El usuario comienza a utilizar el SIGUCI.	2. El sistema muestra una ventana de autenticación.
3. El usuario introduce el usuario y contraseña del administrador.	3. El sistema procesa la información y actualiza el mapa con un indicador en el lugar donde se encuentra el usuario.
4. El usuario busca en el panel derecho la estructura a la que le desea realizar los cambios.(Fig.1)	5. El sistema habilita los botones Localizar, Ver Datos y Modificar.
6. El usuario selecciona el botón Modificar.	7. El sistema despliega una nueva ventana donde se encuentra la información de la estructura seleccionada.(Fig.2)
8. El usuario modifica la información.	9. El sistema muestra un mensaje de confirmación.(Fig.3)
10. El usuario acepta el mensaje de confirmación.	11. El sistema procesa la información y muestra un mensaje de que los datos se han guardado satisfactoriamente. (Fig.4)

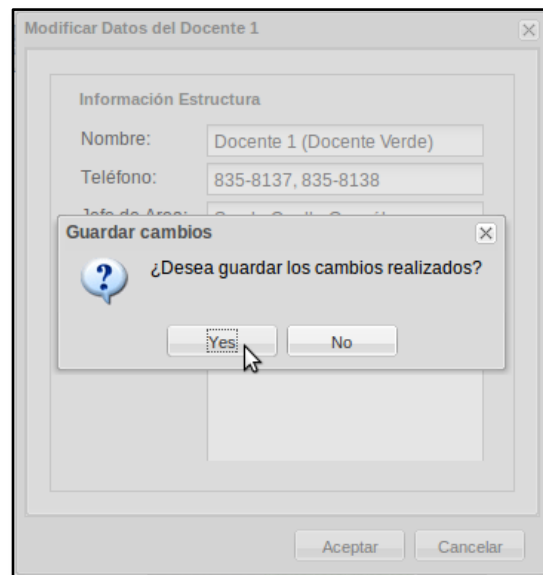
Prototipo de Interfaz



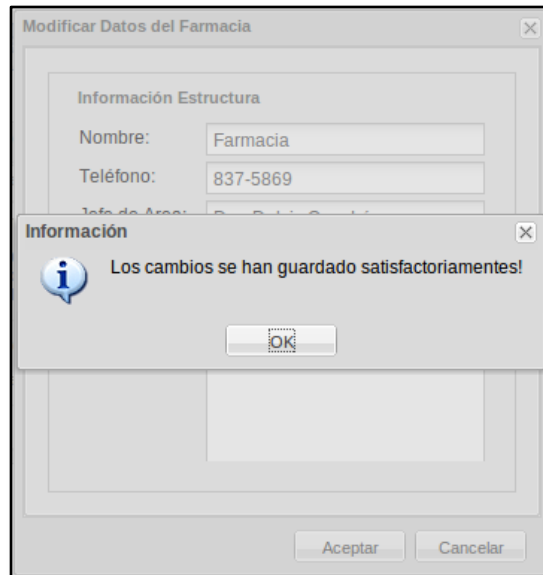
Prototipo de Interfaz Fig. 1



Prototipo de Interfaz Fig. 2



Prototipo de Interfaz Fig. 3



Prototipo de Interfaz Fig. 4

Tabla 10 Descripción textual Caso de Uso Gestionar Datos.

3.6. Conclusiones Parciales.

Se ha presentado en el capítulo que culmina toda la información perteneciente al Modelo del Dominio, por tanto se realizó el análisis de la modelación de los procesos con la especificación de sus datos, actividades y personas involucradas, así como los casos de uso del sistema, también se definieron los requisitos funcionales y no funcionales que el sistema debe cumplir así como la descripción textual de los casos de uso del sistema que se ha propuesto.

Capítulo 4

Construcción de la solución propuesta.

4.1. Introducción.

En el siguiente capítulo se aborda todo lo relacionado con el diseño del sistema. Se hace una descripción de los distintos patrones que se usan en el proceso de implementación, ya sea tanto arquitectónico como de diseño. Se detallará una gran parte del proceso de implementación del sistema, mostrando los diagramas correspondientes al diseño, dando una muestra de lo que es la implementación. Se presentará el modelo de despliegue del sistema, el cual mostrará a través de nodos, cómo estarán distribuidos los recursos que interactúan con el sistema. También en el capítulo se especificarán las técnicas que se utilizarán para realizar las pruebas al sistema.

4.2. Arquitectura Propuesta.

Durante el proceso de desarrollo de un software, el arquitecto de software es quien tiene una responsabilidad global sobre el proyecto, es quien diseña los aspectos más significativos del sistema para que funcionen en su conjunto.

Sin embargo, la arquitectura de software está afectada no sólo por la estructura y el comportamiento, sino también por el uso, la funcionalidad, el rendimiento, la flexibilidad, la reutilización, factibilidad de compresión, las restricciones y compromisos económicos y tecnológicos, y la estética. (21)

✓ Patrón de arquitectura orientada a objetos.

Resumiendo las características de las arquitecturas Orientadas a Objetos (OO), se podría decir que los componentes del estilo se basan en principios OO: encapsulamiento, herencia y polimorfismo. Son las unidades de modelado, diseño e implementación, y los objetos y sus interacciones son el centro de las incumbencias en el diseño de la arquitectura y en la estructura de la aplicación. En cuanto a las restricciones, puede admitirse o no que una interfaz pueda ser implementada por múltiples clases.

El sistema será modelado haciendo uso de este patrón arquitectónico, ya que el framework CartoWeb, que será utilizado para su desarrollo, posee una arquitectura orientada a objetos.

✓ **Patrón de arquitectura basada en componentes.**

Las características principales de este patrón son la modularidad, la reusabilidad y compatibilidad. En la arquitectura basada en componentes también se requiere robustez ya que los componentes han de operar en entornos mucho más heterogéneos y diversos. Su premisa es que los componentes cumplan con alta cohesión y bajo acoplamiento.

El sistema será diseñado sobre la arquitectura basada en componentes, pues el framework CartoWeb también cumple con dicho patrón arquitectónico. La estructura del framework CartoWeb está compuesta por plugins, los cuales se dividen en dos grupos, los core-plugins y los plugins, los primeros son de obligatoria presencia ya que son los utilizados por el sistema y los segundos serán creados por los desarrolladores. Esta posibilidad que brinda el framework permite que la arquitectura del sistema sea flexible y fácil de personalizar.

4.3. Modelo de diseño.

Teniendo en cuenta que el sistema será desarrollado haciendo uso del framework CartoWeb, se hace necesario realizar una descripción de la arquitectura del mismo, para lograr una mejor comprensión de su estructura. El diseño de CartoWeb está conformado por paquetes (*Fig.4*), donde cada uno de ellos realiza una función determinante y la relación interactiva entre los mismos, esto conlleva al funcionamiento óptimo del framework. Entre los paquetes que contiene CartoWeb se encuentra el “Project”, donde se encuentran guardadas todas las aplicaciones que se construyan con esta herramienta, y es precisamente donde se encontrará ubicado el sistema.

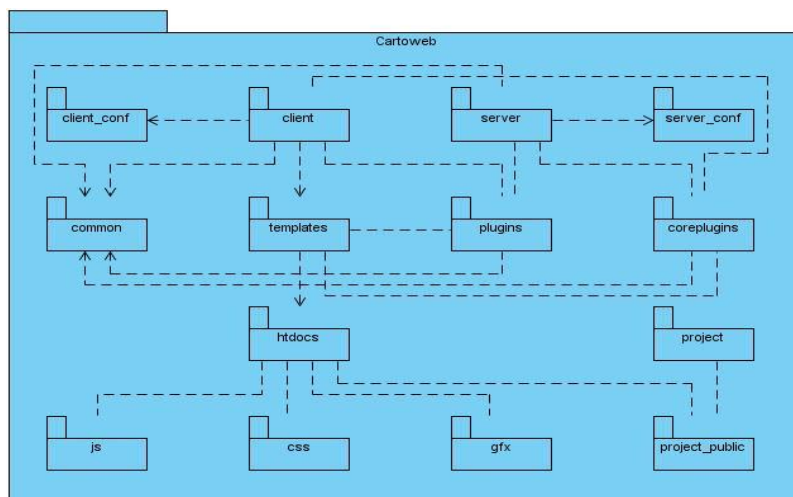


Fig. 4 Diseño de CartoWeb.

CartoWeb, presenta características propias de un geoportal, con la posibilidad de añadir o desarrollar nuevos plugins. Es precisamente a través de estos plugins, que el sistema contendrá un conjunto de funcionalidades, que actúan como herramientas del propio sistema y le brindan la posibilidad de ser altamente modular y escalable. Todos los plugins integrados al sistema tendrán una estructura interna (Fig. 5), estructura que se interrelaciona con los otros paquetes del sistema, solo que cada plugin constituye una funcionalidad específica y entre ellos mismos no existe alguna relación común.

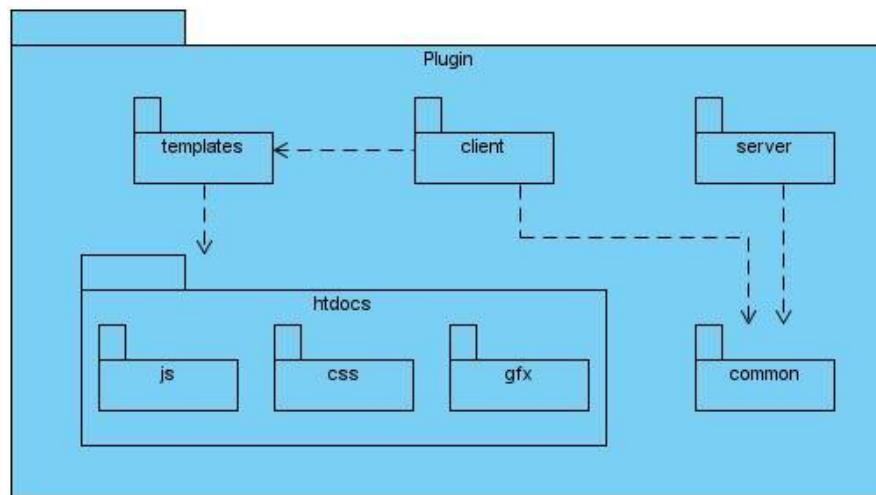


Fig. 5 Diseño de un Plugin de CartoWeb.

Descripción de los paquetes de un plugin.

Módulos	Descripción
client	Contiene todos los componentes de extensión .php del plugins que implementan el módulo correspondiente al lado del cliente.
server	Contiene todos los componentes de extensión .php del plugins que implementan el módulo correspondiente al lado del servidor.
common	Contiene todos los componentes que implementan aquellas clases que servirán de puente para la comunicación entre cliente-servidor, incluyendo el fichero *.wsdl.inc, donde se especifica como acceder a estas vía SOAP.
templates	Contiene específicamente el fichero *.tpl basado en el generador de plantillas Smarty para el procesamiento de datos.

htdocs/js	Contiene todos los componentes de extensión .js del plugins que implementan el módulo cliente en javascript.
htdocs/css	Contiene las hojas de estilo que serán utilizadas en el plugins.
htdocs/gfx	Contiene las imágenes que serán utilizadas en el plugins.

Tabla 11 Descripción de los paquetes de un Plugin.

4.3.1. Patrones de diseño.

A lo largo de los años y a partir de experiencias adquiridas durante el desarrollo de sistemas informáticos, se han ido documentando una serie de patrones y buenas prácticas de programación. Con el objetivo de construir un software basándose en la experiencia colectivas de ingenieros de software. Un patrón define la solución a un problema determinado que pudiera presentarse comúnmente durante el desarrollo de un software. Brindando sugerencias de cómo podría ser solucionado aplicando el patrón en otros contextos.

✓ **Singleton (Solitario):**

En el diseño de clases es necesario aplicar la solución del patrón Singleton que no es más que garantizar el acceso único a una clase mediante una única instancia. De esta forma se controla el acceso a las clases. Se utiliza para modificar el framework CartoWeb, el objetivo del mismo es crear el objeto “mapa” para que no se cree cada vez que se hace un submit en la aplicación.

✓ **Command (Acción):**

Encapsula una petición en un objeto, permitiendo así parametrizar a los clientes con distintas peticiones, encolar o llevar un registro de las peticiones y poder deshacer la operaciones.

Se utiliza en el proceso de petición mediante la Interfaz Gráfica de Usuario (GUI) al sistema de una información cualquiera por un cliente. Uno de los aspectos más importantes en el sistema son las GUI, ya que el usuario interactúa constantemente con ellas y por eso se aplica la solución propuesta por este patrón de diseño.

4.3.2. Diagrama de clases del diseño.

Los diagramas de clases del diseño presentados a continuación están basados en la implementación que se realiza con el framework CartoWeb, además de estar diseñados bajo una arquitectura de componentes. Debido a la complejidad y la extensión de la solución, incluyendo específicamente todas las

clases contenidas en el framework, y siguiendo la premisa de ajustarse al flujo propio de la solución y no a los detalles transparentes al equipo de desarrollo, se decidió representar únicamente las clases y extensiones web que tuvieran que ver directamente con la construcción de la solución, sin dejar de incluir las partes fundamentales del framework que ayuden a su comprensión. Los diagramas de clases presentados a continuación corresponden a algunos de los casos de uso del sistema.

4.3.2.1. Diagrama de clases del diseño del Caso de Uso Localizar IP.

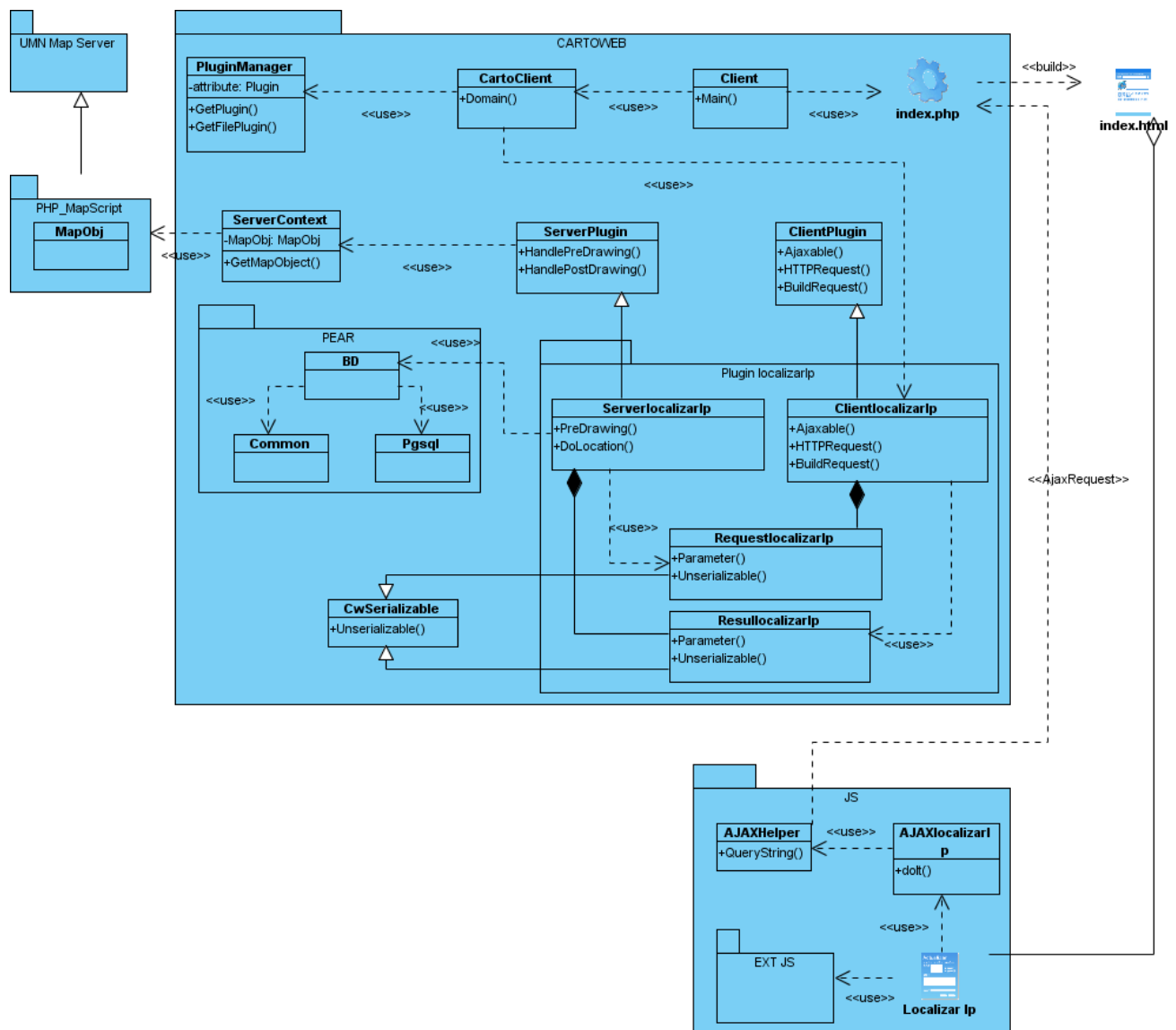


Fig. 6 Localizar IP.

4.4. Principios de diseño.

A la hora de desarrollar un sistema informático pensado para que sea usado por determinados usuarios, sin tener información previa sobre los conocimientos informáticos de los mismos, es conveniente definir algunos principios a la hora de diseñar la apariencia estética de la interfaz de usuario. Con ello se garantiza que el sistema se convierta en una herramienta atractiva para el usuario, y a su vez contribuye a una mejor interacción entre el sistema y quien hace uso del mismo.

Persiguiendo el objetivo de realizar un correcto diseño de la interfaz, así como una mejor navegabilidad en el mismo, el sistema deberá contar con una barra de herramientas ubicada en la parte superior, donde se encontrarán todas las funcionalidades. Contará con un panel ubicado a la izquierda, donde se podrá acceder al área de control de capas, donde el usuario podrá seleccionar las capas que desea visualizar del mapa, al lado derecho se encontrará un panel con la información de las estructuras de la universidad y más abajo se encontrará el mapa de referencia. Se deberá dedicar la mayor área posible para la visualización de los mapas y la información mostrada en estos, esta área estará ubicada en el centro de la pantalla.

4.4.1. Estándares de interfaz de la aplicación.

Para el correcto diseño del SIGUCI se proponen seguir una serie de estándares que van encaminados a garantizar la consistencia del sistema.

- ✓ La mayoría de las funcionalidades deberán estar al alcance de un clic, brindando una interfaz sencilla de manera tal que cualquier persona con un mínimo dominio de computación pueda aprender a trabajar con el sistema.
- ✓ Garantizar la legibilidad de manera que exista contraste de los colores de los textos con el fondo y el tamaño de la fuente sea lo suficientemente adecuado a la vista del usuario.
- ✓ Mostrar al usuario, siempre que vaya a realizar una acción relevante sobre el sistema, un mensaje de confirmación que le permita asegurarse que es correcta la opción seleccionada.
- ✓ Mostrar al usuario solamente aquellas opciones a las que, dado su rol en el negocio, tiene derecho a acceder.

4.5. Diseño de la Base de Datos.

La base de datos necesita de una definición de su estructura, de manera que permita almacenar datos, reconocer el contenido, y recuperar la información. Para diseñar una base de datos se necesita seguir un conjunto de pasos que comienzan con definir las clases persistentes, luego refinarlas y clasificarlas junto con sus atributos, para más tarde realizar el diagrama de clases persistentes. Realizar el diagrama de transición de estado es el siguiente paso para diseñar la base de datos, y el último es la conversión de las clases al medio de almacenamiento. (22)

4.5.1. Diagrama Entidad-Relación.

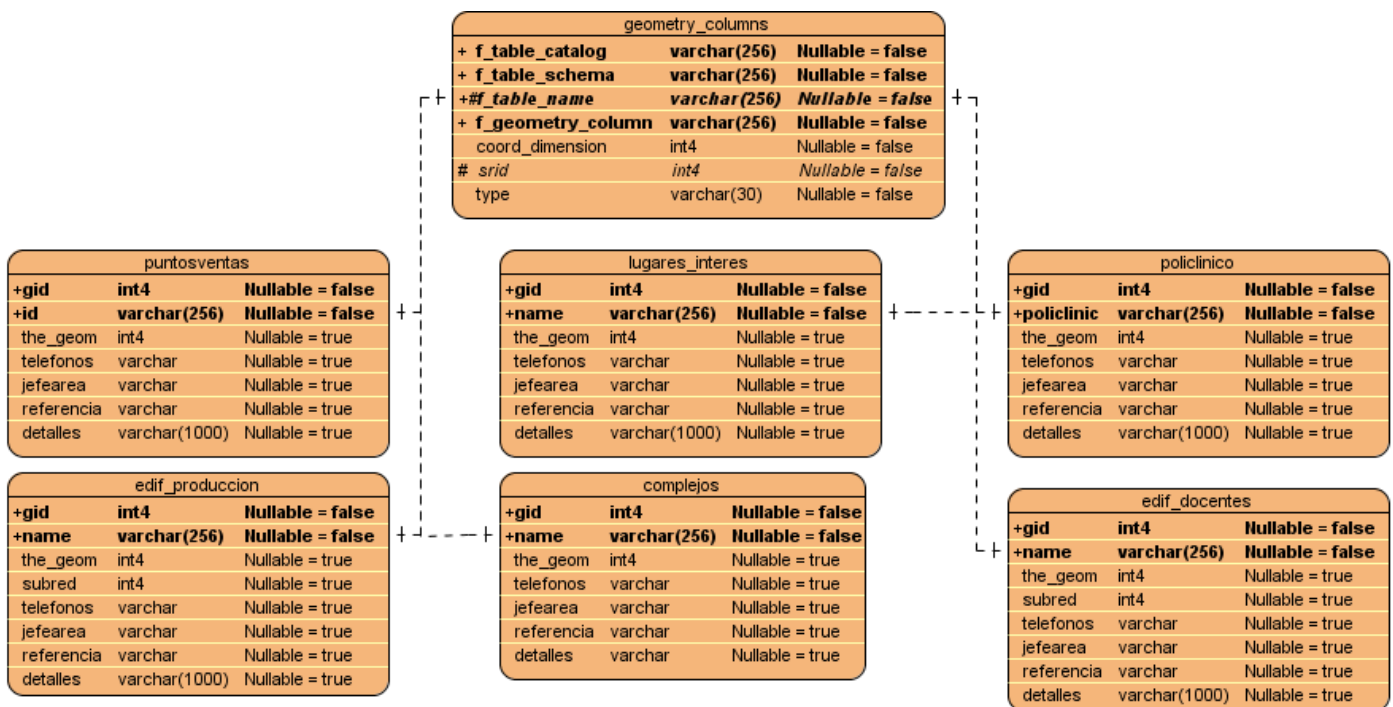


Fig. 7 Diagrama Entidad-Relación.

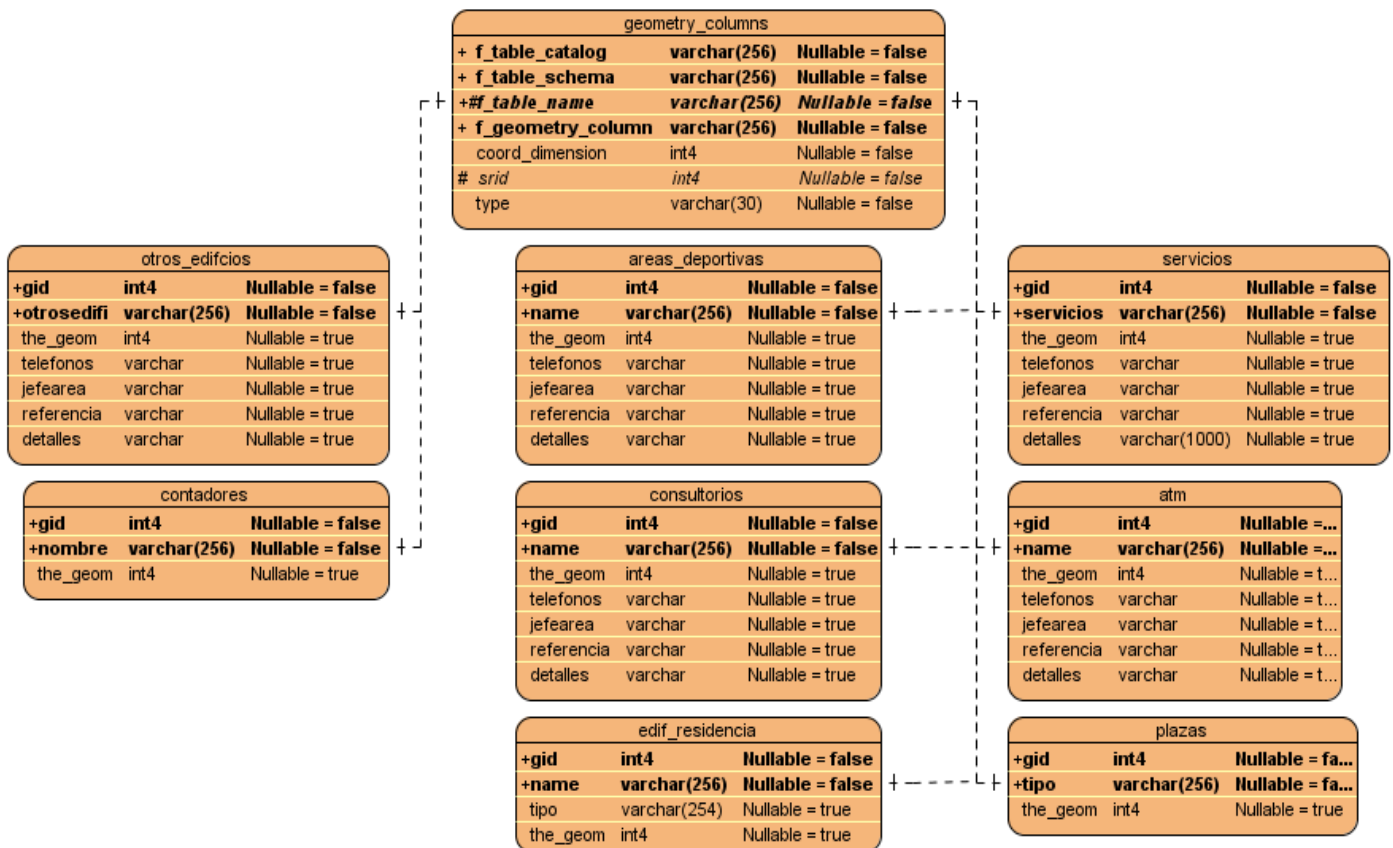


Fig. 8 Diagrama Entidad-Relación. (Cont.)

4.5.2. Diagrama de Clases Persistentes.

La persistencia es la capacidad de un objeto de mantener su valor en el espacio y en el tiempo. Las clases persistentes, sus atributos, y sus relaciones pueden ser implementados directamente en una base de datos orientada a objetos. Lo contrario son las clases temporales que son manejadas y almacenadas por el sistema en tiempo de ejecución por lo que dejan de existir cuando termina el programa.

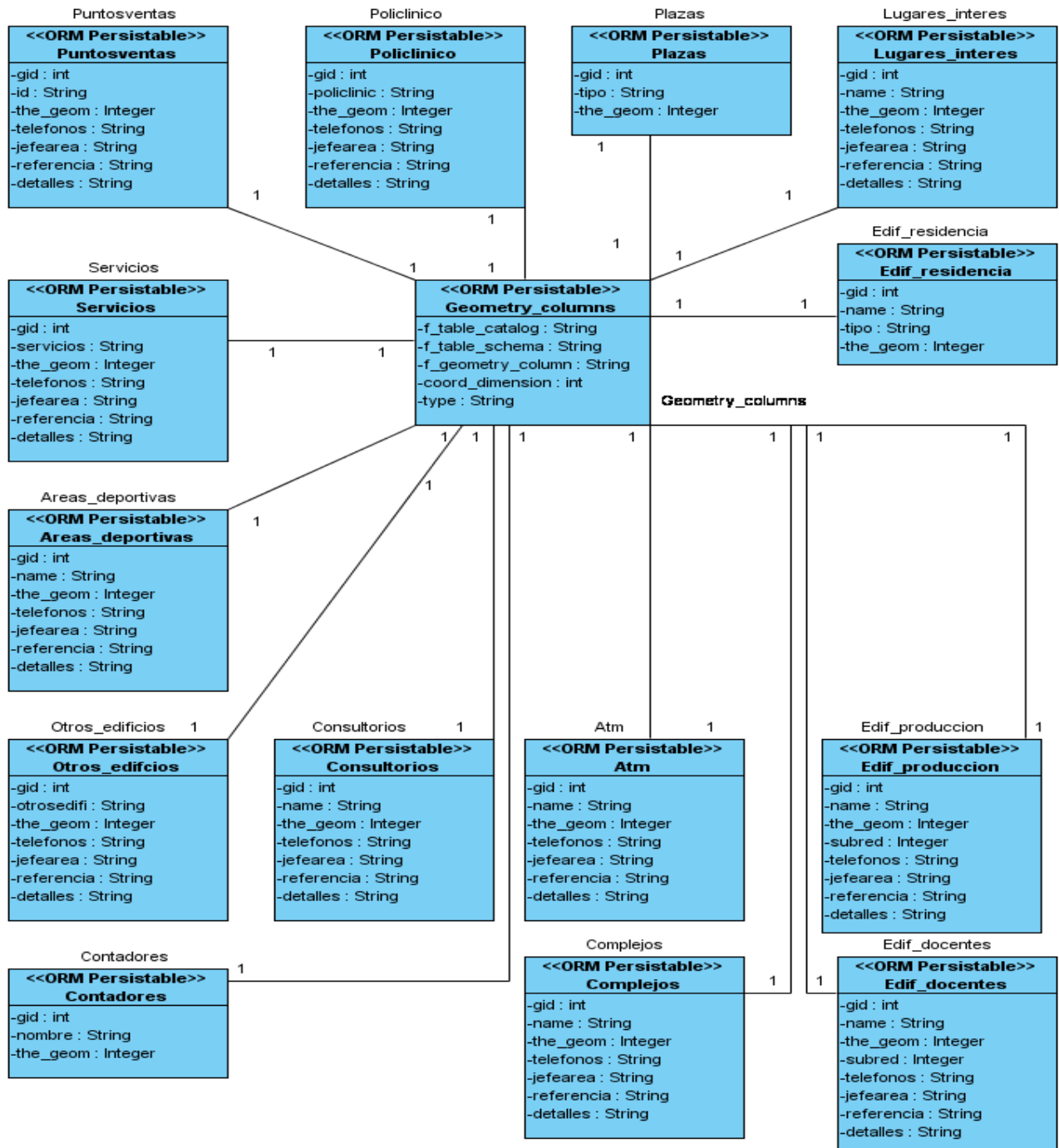


Fig. 9 Diagrama de Clases Persistentes.

4.6. Modelo de Despliegue.

Un diagrama de despliegue muestra las relaciones físicas entre los componentes hardware y software en el sistema. También mapea procesos dentro de estos elementos de procesamiento, permitiendo la distribución del comportamiento a través de los nodos que son representados, estos nodos están unidos por conexiones de comunicación. Un nodo puede contener instancias de componentes software, objetos y procesos.

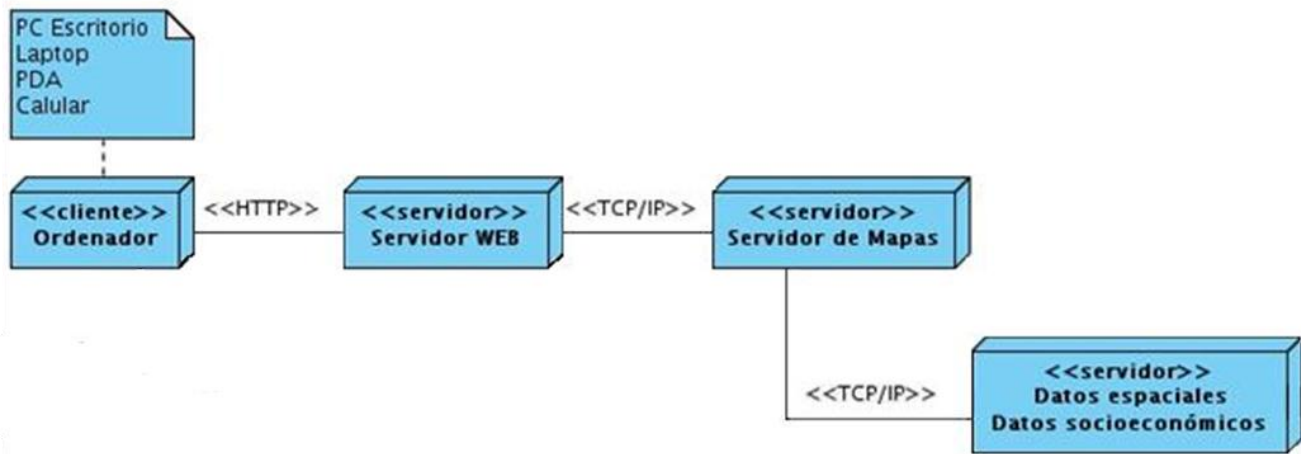


Fig. 10 Modelo de Despliegue.

4.7. Modelo de Implementación.

El modelo de implementación está compuesto por una colección de componentes, subsistemas de implementación, y paquetes utilizados para agrupar elementos del modelo. Los componentes son la parte modular del sistema, desplegable y reemplazable que encapsula implementación y un conjunto de interfaces, y proporciona la realización de los mismos. Un componente en su normalidad contiene clases y puede ser implementado por uno o más artefactos, estos pueden ser un fichero de código fuente, scripts, ficheros de código binario, ejecutables o similares. Los diagramas de componentes son utilizados para modelar la vista estática del sistema, mostrando la organización y las dependencias lógicas entre los componentes.

4.7.1. Diagramas de Componentes.

Estos diagramas han sido confeccionados agrupando los casos de uso por funcionalidad, de la misma manera que fueron agrupados para confeccionar los diagramas de clases del diseño. A continuación se muestra el diagrama correspondiente a la funcionalidad Buscar Número Telefónico, los demás diagramas de componentes se pueden encontrar en el anexo # 2.

4.7.1.1. Diagrama de Componentes Localizar IP.

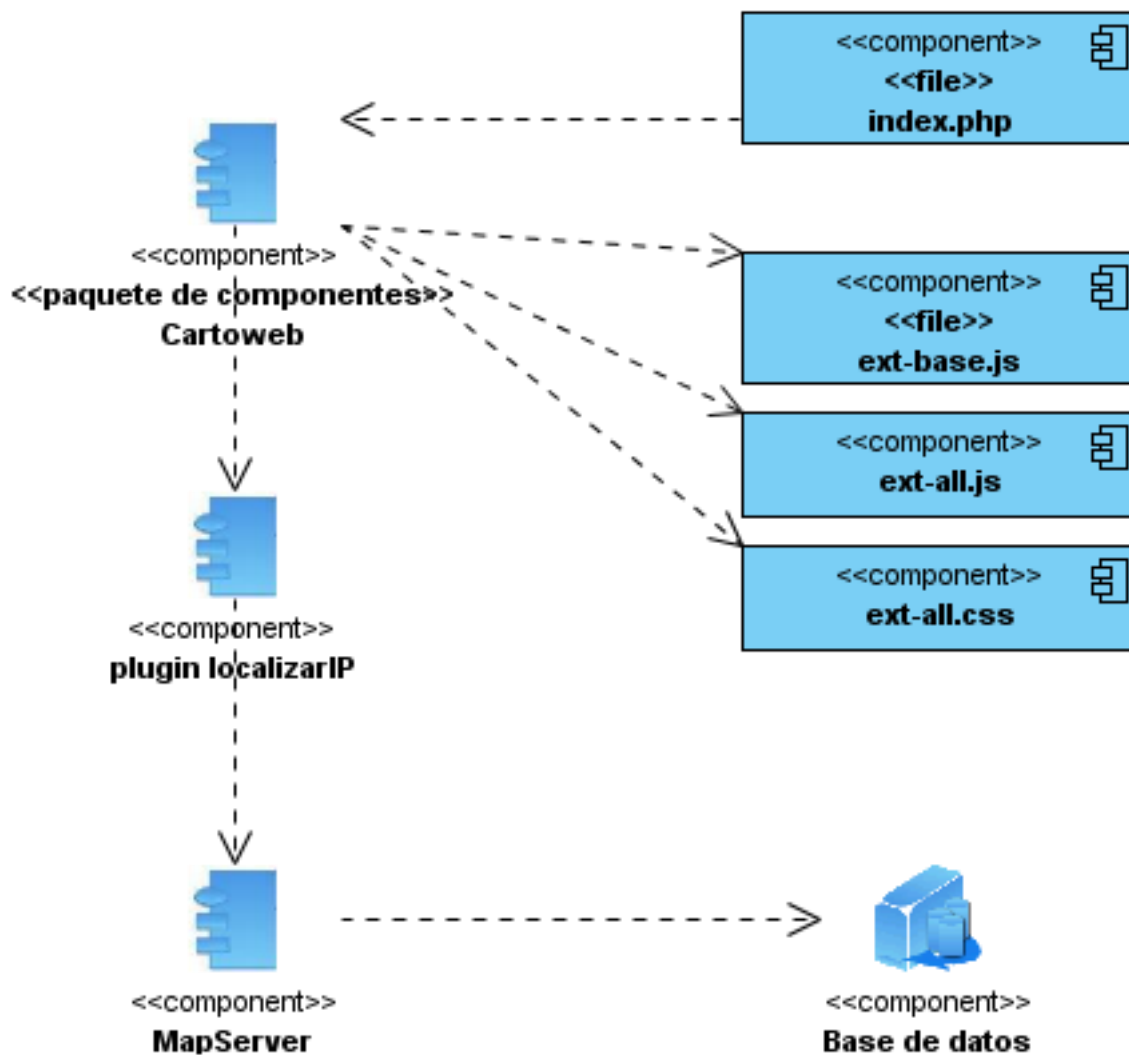


Fig. 11 Diagrama de Componentes Localizar IP.

4.8. Pruebas del Sistema Propuesto.

La prueba de software es un conjunto de herramientas, técnicas y métodos que se le hacen al desempeño de un programa. Las técnicas para encontrar problemas en un programa son extensamente variadas y van desde el uso del ingenio por parte del personal de prueba, hasta herramientas automatizadas que ayudan a aliviar el peso y el costo de tiempo de esta actividad. Las pruebas de caja negra y caja blanca han tomado un lugar muy importante en el desarrollo de sistemas de cualquier tipo, tanto que sin estas pruebas un sistema desarrollado carece de garantía y credibilidad en sus resultados.

4.8.1. Pruebas de Caja Negra.

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

Para la realización de las pruebas utilizando las técnicas de caja negra se escoge el caso de uso: “Localizar Dirección IP”. En la siguiente figura se muestra la interfaz correspondiente al caso de uso.

El siguiente listado son las posibles variantes que pueden existir durante la búsqueda de una dirección IP.

1. Numero de la dirección IP correcto.
2. Campo de texto Buscar IP vacío.
3. Campo de texto Buscar IP con caracteres no válidos.

Caso de prueba # 1. Utilizando técnicas de caja negra.

Caso de Uso:	“Localizar Dirección IP”
Caso de prueba:	1
Entrada:	10.34.12.100
Salida:	Como los datos insertados en el campo Buscar IP son válidos y el número de la dirección IP existe, se localiza en el mapa y se visualiza una pequeña información del lugar al que corresponde la dirección IP.

Tabla 12 Caso de prueba # 1. Utilizando técnicas de caja negra.

Caso de prueba # 2. Utilizando técnicas de caja negra.

Caso de Uso:	“Localizar Dirección IP”
Caso de prueba:	2
Entrada:	10.8.
Salida:	Como los datos insertados en el campo Buscar IP no están completos muestra un mensaje de error.

Tabla 13 Caso de prueba # 2. Utilizando técnicas de caja negra.

Caso de prueba # 3. Utilizando técnicas de caja negra.

Caso de Uso:	“Localizar Dirección IP”
Caso de prueba:	3
Entrada:	10.8.A.*
Salida:	Como el dato insertado en el campo Buscar IP presenta caracteres no válidos el sistema muestra un mensaje de error.

Tabla 14 Caso de prueba # 3. Utilizando técnicas de caja negra.

4.8.2. Validación de Resultados.

Una validación es una verificación documentada que proporciona un alto grado de confianza de que el sistema integral o proceso funciona de la manera prevista, en el ambiente de operación normal. En otras palabras, proporcionar un alto grado de confianza de que un proceso o sistema específico producirá en forma consistente un resultado que cumpla con sus especificaciones predefinidas.

Validación del tiempo de respuesta

Funcionalidades	SIGUCI v1.0	SIGUCI v2.0
Autenticar	No presenta	El sistema tuvo un tiempo de respuesta de 2.06 seg.
Zoom +	El sistema tuvo un tiempo de respuesta de 3.9 seg.	El sistema tuvo un tiempo de respuesta de 4.01 seg.

Zoom -	El sistema tuvo un tiempo de respuesta de 1.80 seg.	El sistema tuvo un tiempo de respuesta de 1.86 seg.
Paneo	El sistema tuvo un tiempo de respuesta de 412 mseg.	El sistema tuvo un tiempo de respuesta de 1.86 seg.
Recentrar Mapa	El sistema tuvo un tiempo de respuesta de 1.90 seg.	El sistema tuvo un tiempo de respuesta de 1.87 seg.
Buscar Usuario	El sistema tuvo un tiempo de respuesta de 3.00 seg.	El sistema tuvo un tiempo de respuesta de 3.41 seg.
Buscar Ip	No tiene	El sistema tuvo un tiempo de respuesta de 2.43 seg.
Buscar Edificio	El sistema tuvo un tiempo de respuesta de 2.35 seg.	El sistema tuvo un tiempo de respuesta de 2.45 seg.
Ubicar Estructura	El sistema tuvo un tiempo de respuesta de 3.21 seg.	El sistema tuvo un tiempo de respuesta de 3.31 seg.
Ubicar IP	No presenta	El sistema tuvo un tiempo de respuesta de 2.04 seg.

Tabla 15 Validación del tiempo de respuesta.

Las diferencias de tiempo después de haberles realizado las pruebas arrojaron que el SIGUCI en su primera versión presenta ventajas ya que tiene menos información a mostrar que la segunda versión, además presenta menor cantidad de funcionalidades a la hora de inicializar el sistema, además la primera versión está alojada en un servidor de mejores prestaciones lo que posibilita que el tiempo de respuesta sea menor.

El SIGUCI en su segunda versión presenta una interfaz más accesible y atractiva para los usuarios, la iconografía es más visible y representativa de las funcionalidades del sistema lo cual posibilita a los usuarios una mejor navegabilidad por el mismo, los colores de la interfaz están acorde con los de la intranet y visualmente se encuentra listo para formar parte del principal sitio de la universidad. En cuanto al mapa, fue mejorado aplicándole texturas a diferentes capas para que la representación de las áreas de la universidad fuera más real, también se agregaron capas nuevas que la antigua versión no tenía para lograr un mayor nivel de especificación de las estructuras de la UCI. Además presenta componentes que

indican al usuario el estado de los procesos de una funcionalidad en uso, a diferencia de la primera versión que presenta una interfaz incómoda, la iconografía es poco visible y no es representativa de las funcionalidades del sistema. En esta nueva entrega se incluye una ayuda, la cual da la posibilidad al personal de la universidad que hace uso del SIGUCI, a tener una guía del sistema y un glosario de términos para la comprensión de las palabras propias de la rama de la Geoinformática.

La segunda versión incorpora funcionalidades las cuales no presentan casi entrada de datos y están al alcance de un clic. En la anterior versión existían pocas funcionalidades y algunas pedían tantos datos que llegaba a ser agobiante para el usuario y al final no mostraba los resultados esperados.

El SIGUCI v1.0 era una aplicación la cual no podía ser usada desde navegadores que no cumplieran con los estándares de la W3C y esto dificultaba grandemente su usabilidad. La nueva versión en desarrollo brinda la posibilidad de consultar el sistema desde los navegadores que violan estos estándares como el Internet Explorer, a la vez que mantiene la compatibilidad con los navegadores Mozilla Firefox, Chrome y Zafari, entre otros.

4.9. Conclusiones.

Con la conclusión de este capítulo, se ha obtenido un sistema completamente diseñado y construido, en términos de clases del diseño se refiere. También se ha obtenido una vista lógica del SIGUCI en componentes y subsistemas de implementación. Se señalaron los principios de diseños a tener en cuenta para el desarrollo de la interfaz y como punto final, se realizaron pruebas al sistema de caja negra y caja blanca, las cuales demuestran que la aplicación y las nuevas funcionalidades que dan respuesta a los problemas identificados en la primera versión del SIGUCI son factibles.

Conclusiones Generales.

Luego del desarrollo de la investigación la cual trae consigo el desarrollo de la segunda versión del Sistema de Información Geográfica para la Universidad de las Ciencias Informáticas, para dar respuestas a los problemas identificados en la primera versión de este sistema. Se considera que los objetivos que dieron inicio al desarrollo de esta investigación se han cumplido.

Haciendo un uso correcto de los métodos teóricos puestos en práctica,

- ✓ Se ampliaron los conocimientos acerca del objeto de estudio, así como el entorno que rodeaba el problema en cuestión.
- ✓ Se desarrolló un sistema informático capaz de dar respuesta a las necesidades identificadas en la primera versión del SIGUCI.
- ✓ Se logró que el sistema cumpla con los requisitos planteados para su desarrollo, por lo que se convierte en una herramienta capaz de brindar los resultados esperados.
- ✓ Se detallaron todos los artefactos generados a partir de la puesta en práctica del Proceso Unificado de Desarrollo como Metodología de Desarrollo de Software.
- ✓ Se logró que la nueva versión del sistema funcione en navegadores que no cumplen con las especificaciones de la W3C.
- ✓ Se crearon nuevas funcionalidades y se mejoraron otras que presentaban problemas de funcionamiento.
- ✓ Se creó una nueva interfaz más agradable y cómoda para la navegación de los usuarios.
- ✓ Se logró un sistema con mayor rendimiento y velocidad a la hora de mostrar la información a pesar de contar con mayor información a procesar que la anterior versión.

El sistema SIGUCI versión 2.0, constituye un aporte importante a la Universidad, brindando además un conjunto de funcionalidades para el manejo y el análisis de los datos. Funciones tales como ubicar geográficamente direcciones IP, mostrar los datos de las estructuras, también permite gestionar los datos de las mismas y la autenticación de usuarios ya sea por dominio uci o locales del sistema, además cuenta con una nueva interfaz más llamativa e intuitiva, así como una gran mejora en el procesamiento interno de la información, lo que brinda una mayor rapidez al sistema.

Recomendaciones.

A partir de los resultados o beneficios que proporciona este trabajo de diploma, se proponen las siguientes recomendaciones:

- ✓ Que se haga una exhaustiva revisión y se despliegue la segunda versión del sistema junto con sus tecnologías correspondientes.
- ✓ Investigar y desarrollar otras posibles funcionalidades, tales como localización de usuarios conectados al dominio uci, búsqueda telefónica y la localización de las postas de guardia por facultades.
- ✓ Que sea consultada la bibliografía de la investigación, por parte de todas aquellas personas que se apoyen en este trabajo, para que amplíen así la información obtenida en el mismo.
- ✓ Realizarle mejoras a la cartografía e incluirle una capa ráster al mapa de la universidad.
- ✓ Revisar y desplegar el sistema para su uso por el personal de universidad.

Glosario de Términos.

Sistemas de Información Geográfica (SIG): Un Sistema de Información Geográfica o SIG, es un sistema de hardware, software y procedimientos elaborados para facilitar la obtención, gestión, manipulación, análisis, modelado, representación y salida de datos espacialmente referenciados, para resolver problemas complejos de planificación y gestión.

Cartografía: Del griego *chartis* = mapa y *graphein* = escrito; es la ciencia que se encarga del estudio y de la elaboración de los mapas.

Cartografía base: Es la colección de mapas y cartas empleadas como fuentes principales de un SIG.

CartoWeb: Framework para el desarrollo de aplicaciones GIS. Complejizado para el trabajo con información geográfica, trabajo con mapas y funcionalidades de análisis geoespacial.

GeoCuba: Grupo empresarial que se dedica a la elaboración, producción y venta de planos, mapas y cartas náuticas con diversos fines, así como a la realización de Estudios Geográficos, de Impacto Ambiental, e investigaciones científicas en ramas del campo de las geociencias, entregando a sus clientes, productos informativos terminados con una alta calidad y fiabilidad.

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

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

Escala: Relación entre la distancia que separa dos puntos en un mapa y la distancia real de esos dos puntos en la superficie terrestre. En los mapas, la escala puede expresarse de tres modos distintos: en forma de proporción o fracción, con una escala gráfica o con una expresión en palabras y cifras. Cuanto mayor es la escala, más se aproxima al tamaño real de los elementos de la superficie terrestre. Los

mapas a pequeña escala generalmente representan grandes porciones de la Tierra y, por tanto, son menos detallados que los mapas realizados con escalas más grandes.

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

Coordenadas geográficas: Es un sistema de coordenadas curvas definido sobre el elipsoide de referencia. Se expresan como Longitud (long.), Latitud (lat.) y Altura (h) donde *long.* y *lat.* son medidas angulares desde el meridiano origen y el ecuador respectivamente; *h* es la altura sobre el elipsoide de referencia.

Georreferencia: Son las coordenadas de un punto que permiten su ubicación sobre el mapa.

Latitud: Es la distancia angular entre el ecuador y un punto determinado del planeta. La latitud se mide en grados, entre 0 y 90; y puede representarse de dos formas:

1. Indicando a qué hemisferio pertenece la coordenada.
2. Valores positivos y negativos.

Tipo de Mapa: Clasificación que se le da a los mapas de acuerdo con su especificación.

Información socioeconómica: Es un conjunto organizado de datos procesados referentes al aspecto social y económico de cualquier lugar de interés del país.

Cartografía: Ciencia que estudia o se encarga de la elaboración de los mapas geográficos y territoriales.

Zoom: Distancia expresada en unidades de medida en el mapa que corresponde con la longitud de la representación actual en el eje X. Entre más grande sea el valor, menor será la escala.

Ráster: Malla, cuadrícula. Imagen formada por los colores o tonos de gris de una cuadrícula, en particular los píxeles del monitor.

Vectorial: Es una estructura de datos que se basa en la definición de puntos, líneas, y áreas.

Datos geográficos: Los datos geográficos son la información que permite conocer lo que se encuentra en una determinada posición en el espacio, de qué manera y en qué tiempo. Los datos geográficos no son más que variables que almacenan una determinada información geoespacial de un evento, ubicándolo en tiempo y espacio.

GeneSIG: Plataforma soberana la cual tiene como objetivo principal desarrollar SIGs o ser la base para personalizaciones aplicables a diversos negocios o sectores sociales.

Tematización: Procedimiento el cual agrupa a todos los elementos que se deseen modificar según nuestras necesidades, se les asigna un color aleatorio y una simbología a una capa del mapa.

Paneo: Acción de mover el mapa.

Zoom: Es la acción de agrandar la parte seleccionada del mapa, lo que implica el uso de una cantidad menor de pixeles y por ello una reducción en la calidad de la imagen.

Plugin: Son las funcionalidades del sistema que serán creados por los desarrolladores

Coreplugin: Son las funcionalidades del sistema que son básicas del sistema como la navegación por el mapa, la gestión de capas, las propiedades de las imágenes entre otros.

Web Coverage Service (WCS): Ofrece una interfaz que permite las solicitudes de coberturas geográficas a través de la web con independencia de la plataforma que ha sido llamada Las coberturas son objetos (o imágenes) en un área geográfica.

Web Map Service (WMS): Produce mapas de datos referenciados espacialmente, de forma dinámica a partir de información geográfica.

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

Trabajos Citados.

1. **DAEDALUS.** [En línea] [Citado el: 12 de 11 de 2010.] <http://www.daedalus.es/inteligencia-de-negocio/sistemas-complejos/ciencia-de-sistemas/que-es-un-sistema/>
2. **Electronic, IEEE Standard Dictionary of Electrical and.** [En línea] [Citado el: 1 de 12 de 2010.] <http://www.daedalus.es/inteligencia-de-negocio/sistemas-complejos/ciencia-de-sistemas/que-es-un-sistema/>
3. **Analysis, National Center for Geographic Information and.** Departamento de Geografía - Universidad de Alcalá. [En línea] [Citado el: 1 de 12 de 2010.] <http://www.geogra.uah.es/gisweb/1modulosespanyol/IntroduccionSIG/GISModule/GISTheory.htm>
4. **Académico, Sitio Web del.** Sitio Web del Académico. [En línea] [Citado el: 4 de 12 de 2010.] <http://grupo.unavirtual.una.ac.cr/mahara/view/artefact.php?artefact=1626&view=147>
5. **Polo Productivo Geoinformática. Facultad 6. Proyecto GeneSIG.SIGUCI.** [En línea] [Citado el: 3 de 12 de 2010.] <http://siguci.uci.cu/htdocs/geoweb/AyudaSIGUCI/>
6. **UCI.** Portal UCI. [En línea] [Citado el: 4 de 12 de 2010.] <http://www.uci.cu/?q=node/59>
7. **UCI.** Portal UCI. [En línea] [Citado el: 4 de 12 de 2010.] <http://www.uci.cu/?q=node/46>
8. **Silva, Investigador Titular Dr. José Luis Batista.** Mapping Interactivo. [En línea] [Citado el: 4 de 12 de 2010.] http://www.mappinginteractivo.com/plantilla-ante.asp?id_articulo=1051
9. **UCI.** Revista GeoNews. [En línea] [Citado el: 1 de 12 de 2010.] <http://revistageonews.wordpress.com/2009/02/15/proyecto-sig-uci/>
10. **Ministerio de la Informática y las Comunicaciones.** Ministerio de la Informática y las Comunicaciones. [En línea] [Citado el: 21 de 2 de 2011.] <http://www.mic.gov.cu/>
11. **Reynoso, Carlos Billy.** *Introducción a la Arquitectura de Software.* BUENOS AIRES : s.n., 2004.

12. **Architecture, WebServices.** W3C Consortium. [En línea] [Citado el: 21 de 2 de 2011.] <http://www.w3.org/TR/2004/NOTE-ws-arch-20040211/#whatis>
13. **Potencier, François Zaninotto Fabien.** *Symfony 1.1, la guía definitiva.* s.l. : Apress, 2008.
14. **Bill Kenedy, Chuck Musciano.** *HTML la guía completa 2da edición.* s.l. : O'Reilly&Associates.
15. **Brandendaugh, Jerry.** *Aplicaciones JavaScript.* s.l. : Anaya Multimedia. ISBN: 84-41 5-1 070-9.
16. **Fabien Potencier, François Zaninotto.** *Symfony 1.1, la guía definitiva.* s.l. : E. Apress.
17. **Camacho, Sunamy Ruiz.** *Personalización de GeneSIG para la Universidad de las Ciencias Informáticas: Rol Analista de Sistemas.* Habana : UCI.
18. **Camacho., Sunamy Ruiz.** *Personalización de GeneSIG para la Universidad de las Ciencias Informáticas: Rol Analista de Sistemas.* Habana : UCI.
19. **Camacho., Sunamy Ruiz.** *Personalización de GeneSIG para la Universidad de las Ciencias Informáticas: Rol Analista de Sistemas.* Habana : UCI.
20. **G. Rumbaugh, J.Jacobson, I. Booch.** *El Proceso Unificado de Desarrollo de Software.* Madrid : Addison – Wesley (Edición en español por la Pearson Educación S.A, 2000).
21. **G. Rumbaugh, J.Jacobson, I. Booch.** *El Proceso Unificado de Desarrollo de Software.* Madrid : Addison – Wesley (Edición en español por la Pearson Educación S.A, 2000).
22. **Yadira Hernández Inza, Héctor René Sánchez Falero.** *“Gestión de Nomencladores para el Programa Nacional de Informatización del Conocimiento Geológico.* Habana : UCI.

Bibliografía Consultada.

Vmapas. <http://www.vmapas.com/>

Cano, José Ignacio Barredo. *Sistemas de Información Geográfica y evolución multicriterio en la ordenación del territorio*. 1996.

DAEDALUS. <http://www.daedalus.es/inteligencia-de-negocio/sistemas-complejos/ciencia-de-sistemas/que-es-un-sistema/> (último acceso: 12 de 11 de 2010).

Definición de cartografía - Qué es, Significado y Concepto. 2008. <http://definicion.de/cartografia/>

El Proceso Unificado de Desarrollo de Software.

ExtJS Framework. <http://groupbit.com/curso-extjs-framework>

Instituto Nacional de Estadística y Geografía. 2010.

<http://mapserver.inegi.gob.mx/geografia/espanol/prodyserv/marcoteo/mundo/mundo.cfm?c=234>

José H. Canós, Patricio Letelier y M^a Carmen Penadés. *Métodologías Ágiles en el Desarrollo de Software*.

Lenguajes del lado del cliente o servidor. 2006.

http://www.adelat.org/media/docum/nuke_publico/lenguajes_del_lado_servidor_o_cliente.html

Lorenzo, Dra. Carmen Mosquera. «LA INFORMACIÓN EN UN SIG.»

Martín, Manuel Martín. *Manual PostGIS*.

Silva, Dr. José Luis Batista. *Aplicación de sistemas de información geográfica en Cuba*. 11 de 2005.

http://www.mappinginteractivo.com/plantilla-ante.asp?id_articulo=1051

Wikipedia. *Aplicación web - Wikipedia, la enciclopedia libre*. 08 de 11 de 2010.

http://es.wikipedia.org/wiki/Aplicaci%C3%B3n_web#cite_ref-ayudaappweb_0-0

MapServer - Wikipedia, la enciclopedia libre. 30 de 09 de 2010. <http://es.wikipedia.org/wiki/MapServer>