



UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS

FACULTAD 6

TRABAJO DE DIPLOMA PARA OPTAR POR EL TÍTULO DE INGENIERO EN CIENCIAS INFORMÁTICAS

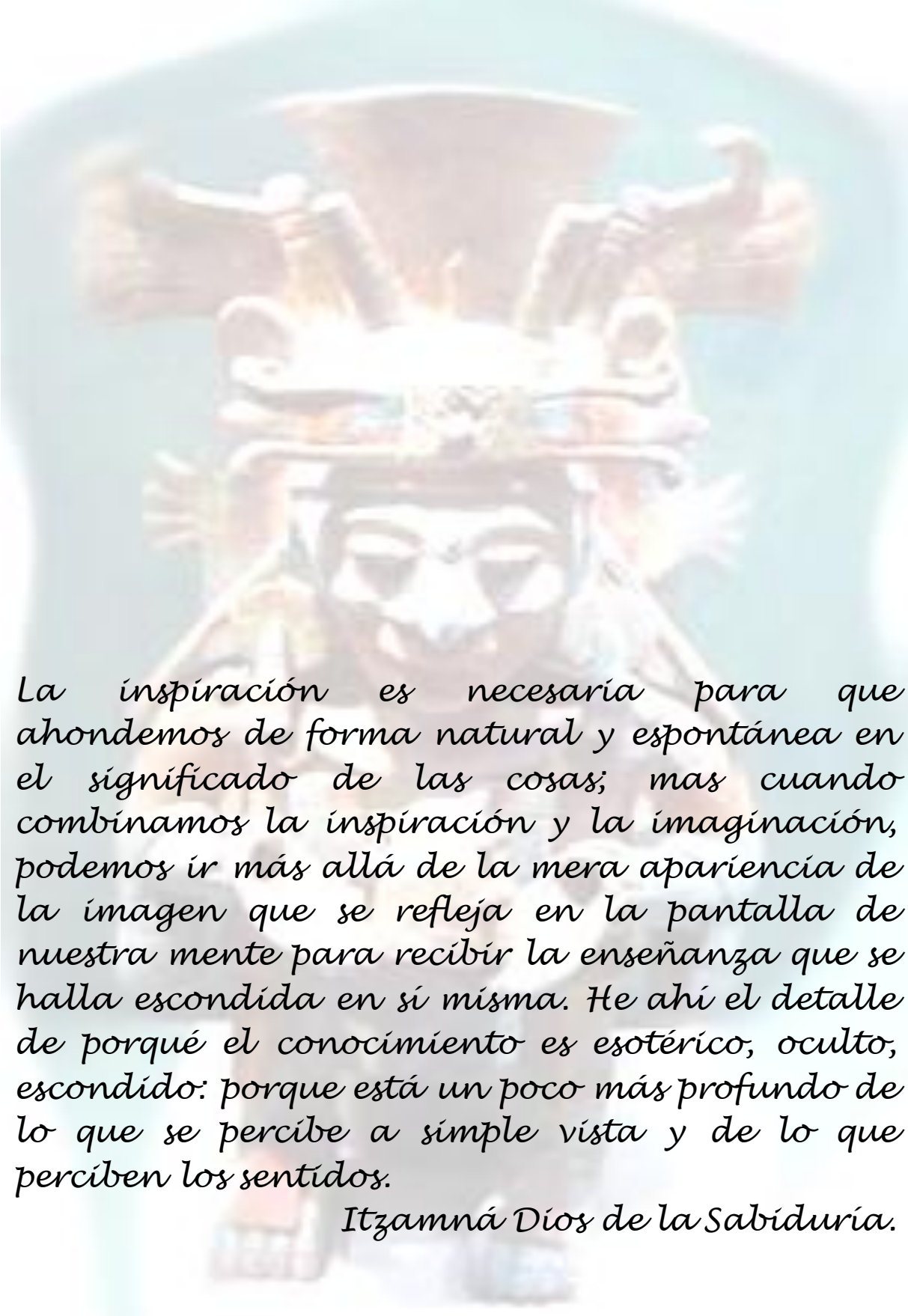
Tema: Propuesta del prototipo funcional de un Sistema de Información Geográfica Web utilizando el servidor de mapas Mapnik

Autor: Lazaro José Estupiñán Cutiño

Tutor: Ing. Yampier Medina Tarancón

Co-Tutores: Ing. Yudiel Rodríguez Larrazábal
Ing. Jesse Daniel Cano Otero

“Año 53 de la Revolución”
La Habana, julio, 1ro de 2011



La inspiración es necesaria para que ahondemos de forma natural y espontánea en el significado de las cosas; mas cuando combinamos la inspiración y la imaginación, podemos ir más allá de la mera apariencia de la imagen que se refleja en la pantalla de nuestra mente para recibir la enseñanza que se halla escondida en sí misma. He ahí el detalle de porqué el conocimiento es esotérico, oculto, escondido: porque está un poco más profundo de lo que se percibe a simple vista y de lo que perciben los sentidos.

Itzamná Dios de la Sabiduría.

DEDICATORIA

Dedico este trabajo de diploma a mi hija Isis Liliet Estupiñan de la Cruz por ser lo más lindo que tengo en la vida.

A mi padre, que dios lo tenga en la gloria y le haga llegar este mensaje de satisfacción.

A mi madre, la dueña absoluta por derecho de todo lo logrado hasta aquí.

AGRADECIMIENTOS

Agradezco a mi madre LÍndá Sucel Cutiño Cordero.

La máxima responsable de que hoy esté dando un paso tan importante en el escalón de la vida. La única persona que siempre creyó en mí incondicionalmente.

A mi hija Isis; por convertir el sacrificio en amor y hacer que todos los días haya un motivo demasiado fuerte para seguir luchando.

A mis hermanos Ernesto, Alexey e Isabel por el apoyo, la confianza y la voluntad inquebrantable de ayudarme siempre.

Agradezco a mi padre Ernesto Estupiñan Martínez, por haberme ayudado a forjar valores claves en la vida, y aún ausente sigue siendo imprescindible.

Agradezco a la persona más paciente y comprensiva del mundo: mi compañera, mi tutora y amiga Ariagnita.

A Jesse, Carlos Enrique, Pedro Enrique, Carlos Luis, Elizita, Osvaldo y David por compartir momentos tan inolvidables en la universidad.

Al tribunal, al tutor y al cotutor por el empeño puesto en este trabajo de diploma.

En general agradezco a todos mis compañeros y amigos. A los profesores desde la secundaria hasta la universidad. Y agradezco a Dios por darme el entendimiento y la sabiduría necesaria para salir adelante.

DECLARACIÓN DE AUTORÍA

Declaro ser autor de la tesis “Propuesta del prototipo funcional de un Sistema de Información Geográfica Web utilizando el servidor de mapas Mapnik” y reconozco los derechos patrimoniales de la misma, con carácter exclusivo, a la Universidad de las Ciencias Informáticas (UCI).

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

Lazaro José Estupiñán Cutiño

Ing. Yampier Medina Tarancón

Firma del Autor

Firma del Tutor

DATOS DE CONTACTO

Tutor: Ing. Yampier Medina Tarancón. Profesor adiestrado graduado de Ingeniero en Ciencias Informáticas en el año 2009.

Dirección de correo electrónico: ytarancon@uci.cu

Co-Tutor: Ing. Yudiel Rodríguez Larrazábal. Profesor adiestrado graduado de Ingeniero en Ciencias Informáticas en el año 2009.

Dirección de correo electrónico: ylarrazabal@uci.cu

Co-Tutor: Ing. Jesse Daniel Cano Otero. Profesor instructor graduado de Ingeniero en Ciencias Informáticas en el año 2008.

Dirección de correo electrónico: jdcano@uci.cu

RESUMEN

Los servidores de mapas son parte importante en el proceso de interacción con los datos espaciales, debido a que permiten a los usuarios visualizarlos y consultar su información geográfica por medio de un Sistema de Información Geográfica (SIG). Este último se ha convertido durante las últimas décadas en una de las más importantes herramientas de trabajo para investigadores, analistas y planificadores y son usados en todas las actividades que tienen como insumo el manejo de la información relacionada con diversos niveles de agregación espacial o territorial.

La presente investigación contempla el desarrollo del prototipo funcional de un Sistema de Información Geográfica Web utilizando el servidor de mapas Mapnik, como parte del proyecto GeneSIG que pertenece al Departamento Geoinformática del Centro de Desarrollo de Software para la Geoinformática y Señales Digitales (GEySED). Su objetivo es visualizar y manejar información geográfica que contribuya a la toma de decisiones con calidad, exactitud y actualidad en dicho proyecto, pretendiendo obtener como posibles resultados un prototipo funcional totalmente libre.

El trabajo de diploma está estructurado en tres capítulos. El primer capítulo recoge toda la teoría relacionada con el proceso a automatizar, así como las tecnologías y herramientas a utilizar. En el segundo capítulo se determinan las funcionalidades que brinda el sistema y se recoge el modelado y diseño del mismo; y en el tercer capítulo, se recoge el modelo de implementación, se muestran los resultados de los casos de prueba realizados, así como el resultado del estudio de factibilidad.

PALABRAS CLAVE

Mapnik, MapServer, servidor de mapa, Sistema de Información Geográfica, SIG, Web

ÍNDICE

| | |
|--|-----------|
| INTRODUCCIÓN | 10 |
| CAPÍTULO 1: PROCESO DE DESARROLLO DE LOS SIG. HERRAMIENTAS Y TECNOLOGÍAS..... | 14 |
| 1.1. INTRODUCCIÓN | 14 |
| 1.2. DESCRIPCIÓN DEL PROTOTIPO FUNCIONAL A DESARROLLAR..... | 14 |
| 1.3. ANTECEDENTES DEL DESARROLLO DE SIG EN EL MUNDO, CUBA Y LA UCI | 14 |
| 1.4. SERVIDORES DE MAPAS | 19 |
| 1.4.1. SERVIDOR DE MAPAS MAPSERVER | 20 |
| 1.4.2. SERVIDOR DE MAPAS MAPNIK..... | 21 |
| 1.5. LENGUAJES DE PROGRAMACIÓN..... | 24 |
| 1.5.1. PYTHON..... | 25 |
| 1.5.2. HTML | 25 |
| 1.5.3. JAVASCRIPT..... | 26 |
| 1.5.3.1. EXTJS..... | 26 |
| 1.5.3.2. JSON..... | 27 |
| 1.5.3.3. AJAX..... | 27 |
| 1.6. ENTORNO DE DESARROLLO INTEGRADO (IDE). ECLIPSE..... | 28 |
| 1.7. FRAMEWORK..... | 29 |
| 1.7.1. DJANGO 1.2 | 29 |
| 1.7.2. WEB2PY 1.87.3 | 30 |
| 1.8. SISTEMA GESTOR DE BASE DE DATOS. POSTGRESQL | 32 |
| 1.9. METODOLOGÍA DE DESARROLLO. AGILE UNIFIED PROCESS (AUP)..... | 33 |
| 1.10. LENGUAJE DE MODELADO. UML | 35 |
| 1.10.1. HERRAMIENTA CASE. VISUAL PARADIGM FOR UML..... | 36 |
| 1.11. CONCLUSIONES PARCIALES DEL CAPÍTULO | 36 |
| CAPÍTULO 2. MODELADO DEL PROTOTIPO FUNCIONAL SIG WEB..... | 37 |
| 2.1. INTRODUCCIÓN | 37 |
| 2.2. MODELO DE DOMINIO | 37 |
| 2.2.1. CONCEPTOS Y EVENTOS PRINCIPALES DEL ENTORNO..... | 37 |
| 2.2.2. DIAGRAMA DE CLASES DEL MODELO DE DOMINIO O CONCEPTUAL..... | 38 |
| 2.3. REQUISITOS | 38 |
| 2.3.1. REQUISITOS FUNCIONALES | 39 |
| 2.3.2. REQUISITOS NO FUNCIONALES..... | 40 |
| 2.4. DIAGRAMA DE CASO DE USO DEL SISTEMA | 42 |
| 2.4.1. DESCRIPCIÓN DE LOS ACTORES DEL SISTEMA | 42 |
| 2.4.2. DESCRIPCIÓN DE LOS CASOS DE USO DEL SISTEMA | 43 |
| 2.4.2.1. MOSTRAR MAPA | 43 |
| 2.4.2.2. REDIMENSIONAR MAPA | 44 |
| 2.4.2.3. MEDIR DISTANCIA | 46 |
| 2.4.2.4. RESTAURAR MAPA..... | 48 |
| 2.4.2.5. EXPORTAR MAPA..... | 50 |
| 2.5. MODELO DE DISEÑO | 52 |
| 2.5.1. DIAGRAMAS DE CLASES DEL DISEÑO..... | 53 |
| 2.5.2. ESTÁNDARES DE LA INTERFAZ..... | 56 |
| 2.6. CONCLUSIONES PARCIALES DEL CAPÍTULO..... | 57 |
| CAPÍTULO 3. IMPLEMENTACIÓN, PRUEBA Y FACTIBILIDAD..... | 58 |
| 3.1. INTRODUCCIÓN | 58 |
| 3.2. MODELO DE DESPLIEGUE..... | 58 |
| 3.3. MODELO DE IMPLEMENTACIÓN..... | 59 |
| 3.3.1. DIAGRAMA DE COMPONENTES..... | 59 |
| 3.4. PRUEBA | 60 |
| 3.4.1. MÉTODO DE CAJA NEGRA | 60 |
| 3.5. ESTUDIO DE FACTIBILIDAD | 62 |
| 3.5.1. ANÁLISIS DE PUNTOS DE CASOS DE USO | 62 |

| | | |
|--------|--|-----------|
| 3.5.2. | ESFUERZO TOTAL..... | 66 |
| 3.6. | CONCLUSIONES PARCIALES..... | 67 |
| | CONCLUSIONES GENERALES | 69 |
| | RECOMENDACIONES | 71 |
| | REFERENCIAS BIBLIOGRÁFICAS | 72 |
| | BIBLIOGRAFÍAS CONSULTADAS | 76 |
| | GLOSARIO DE TÉRMINOS | 79 |
| | ANEXOS | 81 |
| | ANEXO. 1. ENTREVISTA #. 1 REALIZADA AL PERSONAL DEL PROYECTO GENESIG..... | 81 |
| | ANEXO. 2. FIGURA DEL MECANISMO PARA EL PROCESAMIENTO DE DATOS DE LOS SIG | 82 |

Introducción

Los denominados Software libres han surgido como una alternativa de los Software propietarios. Hoy día existe prácticamente un homólogo libre para cada herramienta propietaria, esto no deja exento a los servidores de mapas. Tyler Mitchell define un servidor de mapas como el motor que permite la visualización de mapas en una página Web. Los mapas son generados a partir de datos espaciales que se encuentran almacenados de forma local o remota, es decir, los servidores de mapas tienen la capacidad de integrar datos espaciales de diversas fuentes en una aplicación espacial, la cual se define como una interfaz gráfica a través de un navegador Web que permite interactuar con datos espaciales.

Gran parte de la información tratada por instituciones y empresas tienen relación con datos espaciales, lo que demuestra que la toma de decisiones depende en gran parte de la calidad, exactitud y actualidad de esta información espacial; es aquí donde entran los Sistemas de Información Geográfica (SIG o GIS, en su acrónimo inglés (Geographic Information System)), los cuales hacen uso de los servidores de mapas y se han convertido durante las últimas décadas en una de las más importantes herramientas de trabajo para investigadores, analistas y planificadores, en todas las actividades que tienen como insumo el manejo de la información relacionada con diversos niveles de agregación espacial o territorial.

Según el profesor David Rhind (1989), un SIG no es más que un sistema de hardware, software y procedimientos, diseñados para soportar la captura, el manejo, la manipulación, el análisis, el modelado y el despliegue de datos espacialmente referenciados para la solución de los problemas complejos del manejo y planeamiento territorial.

Este tipo de sistema tiene gran importancia porque provee soluciones para muchos problemas frecuentes, como el acceso a varios tipos de información que sólo pueden ser relacionadas por geografía o distribución espacial. Además, esta tecnología permite almacenar y manipular información usando análisis de patrones, relaciones y tendencias en la información, con el objetivo de ayudar en la toma de mejores decisiones.

Los SIG, se están empleando en el mundo en casi todas las ramas de la ciencia y la tecnología, donde se destacan: las investigaciones científicas, la gestión de los recursos y

activos, la evaluación del impacto ambiental, el marketing, entre otras. Además, esta herramienta ha sido asimilada en universidades, gobiernos, empresas e instituciones y se ha aplicado a sectores como los bienes raíces, la salud pública, la criminología, la defensa nacional, el desarrollo sostenible, los recursos naturales, la ordenación del territorio, el urbanismo y el transporte.

En Cuba estos sistemas se han desarrollado para ser utilizados en áreas como salud, con el objetivo de facilitar la gestión de la estadística y realizar diferentes tipos de análisis con respecto a diversos aspectos como: morbilidad, mortalidad, población, recursos y servicios. También se han desarrollado algunos SIG como parte de proyectos relacionados con el planeamiento urbano y la gestión del medio ambiente, con el objetivo de dar seguimiento a las zonas de posibles inundaciones. Se han desarrollado otros SIG gobernados por ontologías como la *Herramienta para la interpretación semántica de la información espacial*, a partir de los resultados de los estudios geográficos multitemáticos. Dentro de los SIG que se han desarrollado en el país, existen algunos que permiten la gestión y ordenamiento de un preciado volumen de información imposible de procesar en breve tiempo por métodos tradicionales, para lograr incorporar datos de alto valor científico a los procesos de manejo, toma de decisiones, políticas y estrategia de desarrollo, tal es el caso del SIG basado en el ecosistema Sabana-Camagüey.

En la Universidad de las Ciencias Informáticas (UCI), también se desarrollan SIG para solucionar problemas de la rama del petróleo y los yacimientos minerales, por el proyecto GeneSIG. Este proyecto pertenece al Departamento GeoInformática del Centro de Desarrollo de Software para la GeoInformática y Señales Digitales (GEySED), de la UCI, Facultad 6. El objetivo de este proyecto es desarrollar SIG mediante una plataforma que lleva ese mismo nombre, haciendo uso de tecnologías libres que permita la conexión a servicios de cartografías digitales publicadas en la Web, así como el manejo de bases de datos de gran tamaño y la interacción del sistema-intérprete de manera flexible. Además de contribuir en la generación de la información dinámica válida, para la gestión de las diversas actividades y la toma de decisiones. La alta carga de transacciones provocadas por el peso de la información geográfica que se maneja por la Web y las limitadas capacidades de procesamiento en entorno del cliente (a diferencia de los SIG Desktop) conlleva a la necesidad de buscar alternativas para lograr un mejor rendimiento de estos tipos de sistemas (SIG Web); aparte de las cuestiones de optimización que puedan

tratarse, se ha considerado la alternativa de buscar si los servidores de mapas pudieran resolver esta dificultad. De ahí que se identificara, estudiar las capacidades y el rendimiento de Mapnik a través de su explotación en un SIG, con características (funcionalidades seleccionadas) similares a las que soporta el producto GeneSIG, con el fin de compararlo posteriormente con MapServer e identificar características que podrían extrapolarse a las soluciones que se explotan sobre este último.

Con la presente investigación se propone desarrollar el prototipo funcional de un Sistema de Información Geográfica Web utilizando el servidor de mapas Mapnik, que facilite la futura implementación de soluciones similares en el centro.

A partir de esta situación problemática se determinó el siguiente problema a resolver: ¿Cómo mejorar la visualización y el manejo de información geográfica para contribuir a la toma de decisiones con calidad, exactitud y actualidad en el Proyecto GeneSIG? Teniendo como objeto de estudio, el proceso de desarrollo de SIG y como campo de acción, el proceso de desarrollo de SIG utilizando el servidor de mapas Mapnik.

Para solucionar el problema expuesto anteriormente se plantea como objetivo general: desarrollar el prototipo funcional de un SIG Web utilizando el servidor de mapas Mapnik, haciendo uso además, de tecnologías libres.

Para dar cumplimiento al objetivo general se trazaron las siguientes tareas de la investigación:

1. Valorar los antecedentes y tendencias en el desarrollo de SIG usando el servidor de mapas Mapnik.
2. Identificar la metodología y tecnologías a utilizar en el desarrollo del prototipo funcional SIG Web sobre el servidor de mapas Mapnik.
3. Modelar los procesos del diseño del sistema.
4. Implementar el sistema.
5. Validar la solución propuesta.

Una vez planteado el objetivo general se defiende la siguiente idea: el desarrollo del prototipo funcional de una aplicación SIG usando el servidor de mapas Mapnik con la correspondiente documentación técnica del proceso, contribuirá a mejorar la gestión de información geográfica en el proyecto GeneSIG; esperando obtener como posible resultado, un prototipo funcional de aplicación SIG en la Web con tecnologías libres

utilizando Mapnik como servidor de mapas y la documentación técnica del proceso de desarrollo.

Fue necesario guiar la investigación utilizando Métodos Científicos, tanto teóricos como empíricos. Los métodos científicos teóricos permitieron estudiar las características del objeto de investigación. Dentro de estos se empleó, el Analítico – sintético, para buscar la esencia de los fenómenos y caracterizar los elementos más importantes relacionados con el objeto de estudio. Además ayudó en el establecimiento de los métodos, herramientas y procedimientos más factibles para la implementación del prototipo funcional que se propone. También se empleó el método inductivo – deductivo para generalizar el conocimiento adquirido luego de aceptada la idea a defender y determinar que el desarrollo de un prototipo funcional SIG Web sobre el servidor de mapa Mapnik es una idea bien aceptada. Otro método utilizado es el histórico – lógico con el objetivo de actualizar los antecedentes y el estado actual del desarrollo de soluciones SIG sobre el servidor de mapa Mapnik, así como los logros y limitaciones de las mismas en Cuba y en el mundo. Y por último se utilizó la Modelación para dar cumplimiento a las tareas de investigación, diseñar e implementar un prototipo funcional SIG Web sobre el servidor de mapa Mapnik.

Los métodos científicos empíricos posibilitaron el conocimiento del estado del arte. Dentro de los métodos se empleó la Observación para recoger la información de cada uno de los conceptos asociados al dominio del problema e ir evaluando el avance y desarrollo del sistema. Otro método empírico usado fue la Entrevista para definir los requisitos funcionales y no funcionales con los que debe cumplir el prototipo funcional SIG Web, actividad que constituye el cimiento para el diseño y la implementación. Además este último ayudó a comprender la relación de las clases del dominio relacionado con el proceso que se desea informatizar.

Esta investigación está estructurada en tres capítulos. El primer capítulo recoge toda la teoría relacionada con el proceso a automatizar, así como las tecnologías y herramientas a utilizar. En el segundo capítulo se determinan las funcionalidades que brinda el sistema y se recoge el modelado y diseño del mismo; y el tercer capítulo recoge el modelo de implementación, muestra los resultados, de los casos de prueba realizados y del estudio de factibilidad.

Capítulo 1: Proceso de desarrollo de los SIG. Herramientas y tecnologías

1.1. Introducción

En este capítulo se recoge el fundamento teórico de la investigación, prestando especial atención al estudio de las características de los servidores de mapas Mapnik y MapServer; y además a la selección de las herramientas y tecnologías libres necesarias, tales como: framework, lenguajes de programación, metodología de desarrollo, lenguaje de modelado, Entorno de Desarrollo Integrado y Sistema Gestor de Base de Datos.

1.2. Descripción del prototipo funcional a desarrollar

En busca de alternativas para lograr un mayor rendimiento en los SIG, se propone desarrollar el prototipo funcional de un SIG Web haciendo uso del servidor de mapas Mapnik.

De acuerdo con los objetivos propuestos en el trabajo, el sistema debe ser capaz de mostrar mapas, permitir la selección de capas, realizar paneo, medir distancia, acercar (zoom in), alejar (zoom out), así como centrar, restaurar y exportar mapa. Además brindará la posibilidad de incluirle nuevas funcionalidades más complejas en futuras versiones.

1.3. Antecedentes del desarrollo de SIG en el mundo, Cuba y la UCI

El auge informático a nivel mundial ha permitido que en las últimas décadas los SIG hayan evolucionado rápidamente y se utilicen para resolver problemas de diversas áreas. A continuación se detallan algunos ejemplos de SIG desarrollados en Cuba y en otros países.

SIG desarrollados en el mundo

En el mundo existen empresas que se dedican específicamente al desarrollo de SIG, teniendo mayor connotación en el área de América Latina y el Caribe. En esta área se destacan las empresas: GEO SAT, de Argentina; SIG-DES LTDA, de Colombia; ACE TECHNOLOGIES, la cual opera en El Salvador y Guatemala; GEOTICA CORP S.A., de

Costa Rica y PIACHE de Centroamérica S.A., esta última, opera en Costa Rica, El Salvador, Guatemala, Honduras, Nicaragua y Panamá. Algunos de los sistemas que se especifican a continuación han sido desarrollados por algunas de estas empresas, como es el caso del software SIG Agropecuario e Identificación SIG Guadua, desarrollados por las empresas, argentina y colombiana respectivamente.

SIG Agropecuario

El SIG Agropecuario es un sistema desarrollado en Argentina, el cual permite obtener un documento cartográfico de gran utilidad que resulta relevante para tomar las decisiones del ámbito agropecuario. Este incluye datos estadísticos del uso de la tierra y datos sobre factores naturales (agro-meteorológicos, edafológicos, hidrológicos, etc.), además de otros datos importantes para la toma de decisiones. Para desarrollar este sistema se usó como servidor de mapas MapServer. [35]

QUANTUM GIS /GRASS

Este SIG es un software no-comercial de código abierto y libre que puede ser descargado gratuitamente. Este software es usado para el análisis petroglifos en México; incluye herramientas útiles para el registro sistemático, el análisis de la distribución y los contextos de petroglifos en el área de interés. Utiliza el servidor de mapas MapServer. [37]

Sistema de Información Geográfica (SIGPAC)

Este es un sistema utilizado en España. Brinda servicios de Mapas, acceso a mapas, ortofotos, sistema WMS, así como servicios de GPS, descarga de archivos de las bases de referencia para correcciones diferenciales GPS. Utiliza MapServer como servidor de mapas. [38]

Identificación SIG Guadua

El SIG Guadua es un sistema de información Web colombiano, que ofrece datos generales del recurso guadua. Es una herramienta que ayuda a la comunidad a resolver sus inquietudes entorno al recurso guadua y sirve de apoyo a las CAR´s que lo conforman para la toma de decisiones. [39]

Al analizar los sistemas desarrollados a nivel mundial que se describen anteriormente, se puede ultimar que es notable la tendencia hacia en el desarrollo de SIG utilizando MapServer como servidor de mapas. Más adelante se analizan las características de este servidor de mapas. (Ver epígrafe 1.4.1.)

SIG desarrollados en Cuba

En Cuba las aplicaciones SIG comenzaron a ser utilizadas desde mediados del año 1987 por especialistas del Instituto de Geografía Tropical de Cuba, para el estudio de mapas de riesgos de inundaciones. En la actualidad, las aplicaciones SIG son muy variadas por su uso, tales como: el reordenamiento territorial, la planificación física y la toma de decisiones en diferentes campos de la ciencia y la técnica. [11] A continuación se describen algunos SIG desarrollados en Cuba para solucionar problemas en áreas diversas.

El SIG “Manejo y gestión de la información que se genera en el sistema de salud cubano a nivel de la atención primaria de la salud, en los consultorios y policlínicos, para ser usado por la Dirección Municipal de Salud” permite cartografiar y realizar diferentes tipos de análisis con respecto a diversos aspectos de la salud como: morbilidad, mortalidad, población, recursos y servicios. [1,2] Este fue uno de los primeros SIG desarrollados en el país.

“Herramienta para la interpretación semántica de la información espacial y su integración a la Infraestructura de Datos Espaciales de la República de Cuba (IDERC)” es un SIG gobernado por ontologías desarrollado por los especialistas del Centro de Aplicaciones de Tecnologías de Avanzada (CENATAV); a partir de los resultados de los estudios geográficos multitemáticos realizados por los diferentes grupos de expertos del proyecto dirigido a la protección y uso sostenible de la Biodiversidad, patrocinado por el Programa de las Naciones Unidas para el Desarrollo (PNUD) y el Fondo para el Medio Ambiente Mundial (GEF), (GEF\PNUD) en el ecosistema Sabana-Camagüey. El software permite la gestión y ordenamiento de un preciado volumen de información que es prácticamente imposible procesar, en tan breve tiempo, por métodos tradicionales. [3,4]

Sig_inrh, este es un SIG para el Instituto de Recursos Hidráulicos, el cual permite obtener datos de los recursos hidráulicos existentes en Cuba. Permite la visualización de mapas y utilizando el servidor de mapas MapServer. [9]

SIG_insmet, SIG para el Instituto de Meteorología de la República de Cuba (INSMET). Permite conocer en tiempo real el comportamiento de un número de parámetros que se capturan en las estaciones meteorológicas automatizadas y utiliza como servidor de mapas a MapServer. [9]

En algunas Universidades cubanas también se desarrollan investigaciones referentes a los SIG, como es el caso de la Universidad Central de Las Villas (UCLV) y la Universidad de las Ciencias Informáticas (UCI). En la UCLV el Doctor en Ciencias Ramiro Pérez Vázquez presentó un estudio el 12 de noviembre de 2010 sobre “Los sistemas de información geográfica aplicados a los problemas medio ambientales”. Por su parte en la UCI desde el 2008 se investiga en la línea de los SIG, dentro del antiguo Polo Productivo Geoinformática, actualmente GEySED, iniciando con el proyecto “Geografía de la UCI (SIG-UCI)” (actualmente proyecto GeneSIG), con el afán de fortalecer la experiencia en la realización de SIG para la toma de decisiones en el trabajo organizacional del centro y así ampliar la infraestructura de servicios que exigen los usuarios de la entidad.

Es apreciable como en Cuba también ha existido una tendencia a desarrollar SIG utilizando el servidor de mapas MapServer y es precisamente por ello que en la UCI se fomenta el estudio de otros servidores de mapas. A continuación se detallan algunos SIG desarrollados por esta universidad y se mencionan otros que se encuentran en fase de ejecución.

SIG desarrollados en la UCI

En la UCI el desarrollo de SIG ha evolucionado en espiral desde el 2008 hasta la fecha, esto se demuestra con los resultados obtenidos:

“Aplicación Web que permite el manejo de datos geospaciales (mapas interactivos)”. Esta aplicación Web brinda al usuario la posibilidad de realizar una navegación básica por el mapa de la UCI. La misma está basada en el servidor de mapas opensource

MapServer y cuenta con un módulo que brinda al usuario la posibilidad de realizar análisis estadísticos con la información real de un área determinada. [9]

“Sistema de información geográfica de la UCI basado en tecnología opensource”. Este sistema cumple con las especificaciones del Consorcio OpenGis (OGC) en consonancia con la política de migración al software libre en la que se ve envuelta el país. El objetivo del mismo es ayudar en la toma de decisiones en cuanto a: seleccionar el mejor lugar para ubicar un nuevo edificio o un área recreativa dentro de la universidad. [9]

“Servicio de mapas temáticos” es un sistema que brinda las principales facilidades, igual que otros sistemas actuales. Permite ser usado y modificado libremente por cualquier usuario. Para la implementación del servicio se utilizó Mapnik como servidor de mapas y la librería Matplotlib, acompañado del lenguaje de programación Python, y otras herramientas libres. Este servicio permite generar mapas temáticos de datos puntuales, datos lineales, datos superficiales, puntos, datos proporcionales, isolíneas, flujo, coropletas y cartogramas. Además, permite obtener los datos de varias formas: desde un fichero Excel, desde una tabla en una Base de Datos Postgres o sencillamente datos existentes en bases de datos que están integradas al servicio (este sistema fue desarrollado para la UCI, por tanto se refiere a Base de Datos como Akademos u otra Base de Datos de la universidad). [36]

Como se puede apreciar en la UCI la balanza en cuanto al desarrollo de SIG, no se inclina solamente hacia el uso de MapServer como servidor de mapas, sino que también se ha considerado el uso de Mapnik, en busca de ampliar los conocimientos y generar alternativas más efectivas y eficientes. Cabe notar que además de los sistemas mencionados anteriormente, existen otros que se encuentran en fase de desarrollo, tales como:

- Algoritmos para el módulo de análisis del terreno de GeneSIG: análisis hidrológico y álgebra de mapa.
- Sistema de información geográfica de rutas de transporte obrero de la UCI
- Estrategia para implementar Sistemas de Información Geográficos petroleros sobre la base de la Plataforma GeneSIG.
- Desarrollo de una aplicación informática de Sistemas de información geográfica para la UCI sobre GVSIG Mobile.

Luego de haber analizado las tendencias en cuanto al uso de servidores de mapas, se hace necesario definir que es un servidor de mapas y cuales son las características fundamentales tanto de Mapnik como de MapServer.

1.4. Servidores de mapas

Los servidores de mapas tienen como objeto acceder a información geoespacial existente, normalmente en servidores, en diferentes formatos y servir dicha información a clientes de mapas a través de protocolos estándares. Permiten las funcionalidades de: visualización, se refiere al zoom para alejar o acercar los elementos cartográficos; identificación de atributos alfanuméricos en cada elemento cartográfico (“identify”); consultas de atributos alfanuméricos: sencillas, como la búsqueda de topónimos o más complejas; conexión de bases de datos locales a la base de datos remota del servidor de mapas (“data binding”), de cara a la creación de mapas temáticos con datos alfanuméricos propios, o para el volcado masivo o una a una de direcciones postales como puntos en una capa de ejes de calles (geocodificación de direcciones postales o “addressmatching”); selección de elementos por combinación de capas o análisis con operadores espaciales de superposición, contención, intersección, de dos capas (con la opción de creación de nuevas capas) y creación de zonas de influencia (“buffers”, señala la zona de riesgo a 2 Km de una industria de productos tóxicos); cálculo de rutas óptimas para la navegación de vehículos (“routing”); edición básica de líneas (“redlining”) por parte del cliente, de manera que el administrador del servidor de mapas puede recuperar esas líneas e incorporarlas a la cartografía y capacidad de imprimir el mapa manteniendo la escala. [28]

Arquitectura de los servidores de mapas

La arquitectura de los servidores de mapas es de tipo cliente/servidor. El cliente; en este caso, un “browser” o explorador de Internet, solicita los recursos del servidor. El servidor gestiona todas las peticiones y responde de manera ordenada a éstas. La red es la estructura física a través de la que cliente y servidor se comunican. El cliente, al recibir los datos del servidor (por ejemplo, código HTML) los interpreta y los presenta al usuario (en el “browser” como texto con un determinado estilo, tamaño de fuente, color, etc.). [28] (Ver Figura 1.1)



Figura 1.1. Esquema de la arquitectura de un servidor de mapas.

Existen servidores de mapas de código abierto y servidores de mapas comerciales. Los servidores de mapas de código abierto en comparación con los servidores de mapas comerciales pueden ser una buena elección para los usuarios, debido a que permiten el desarrollo tanto interno como externo, y generalmente no implican un coste económico considerable. Por otra parte los servidores de mapas comerciales son una alternativa para las empresas u organizaciones que están dispuestas a pagar un coste económico por un servidor de mapas a condición de que estos ofrezcan una solución rápida, sencilla de instalar y configurar, que no tenga defectos (bugs) y que ofrezca prestaciones sofisticadas o no disponibles en otros servidores de mapas de código abierto o de servicios gratuitos. Dentro de los servidores de mapas de código abierto se encuentran MapServer y Mapnik. [6]

1.4.1. Servidor de Mapas MapServer

MapServer es un software (servidor de mapas) de código abierto destinado a renderizar datos espaciales para su publicación a través de Internet. Fue originalmente desarrollado por la Universidad de Minnesota (UMN) en cooperación con la NASA y el Departamento de Recursos Naturales de Minnesota. Permite dibujar capas de información dependiendo de la escala, utilizando plantillas de salida totalmente personalizables. Se caracteriza por

tener un archivo de configuración denominado mapfile que tiene como extensión “.map”; que no es más que un fichero de texto donde se definen los datos a ser usados por la aplicación como: las capas, fuente de datos de origen y forma de servir los datos; leyenda, proyecciones, entre otras. [34]

MapServer soporta la proyección de las imágenes de salida de forma dinámica (On the fly), con aproximadamente mil proyecciones debido a la utilización de la librería Proj.4. Soporta varios lenguajes de programación, entre ellos Python. Es multiplataforma y capaz de operar sobre los sistemas operativos: Linux, Windows, Mac OS X y Solaris.

Entre las características más relevantes de este servidor se destaca la presencia de soporte para datos ráster (TIFF/GeoTIFF, EPPL7) y vectoriales (ESRI shapefiles, PostGIS, ESRI ArcSDE, Oracle Spatial, MySQL) y la posibilidad de integración con las bibliotecas GDAL y OGR, para aumentar el número de formatos soportados. Otra característica importante es su capacidad de trabajar de forma interoperable con diversos SIG y aplicaciones que siguen las especificaciones establecidas por el OGC. Esta última característica es la que permite la publicación de un fichero de mapa como servicio Web de mapas, posibilitando que puedan establecerse mecanismos de caché entre las aplicaciones clientes y el servidor de mapas. [34]

1.4.2. Servidor de Mapas Mapnik

Mapnik se puede definir como un servidor de mapas de código abierto para desarrollar aplicaciones cartográficas. Desarrollado en C++ y utiliza Python como lenguaje de programación para facilitar el desarrollo flexible y rápido de SIG. El mismo funciona sobre los sistemas operativos Windows, Linux y MacOS. Por esta razón se considera multiplataforma. Soporta la Base de Datos PostgreSQL y los estándares WMS. Permite crear y manejar mapas a partir de diferentes fuentes de datos. Se puede utilizar como motor gráfico para renderizar casillas de información geográfica. Se basa en el objeto MAP, el cual contiene varios atributos que permiten personalizarlo: background (establece el color de fondo del mapa), height (ancho del mapa), width (alto del mapa), style (define cómo se visualizan de los distintos componentes del mapa) y layers (capas que contiene el mapa).

Además esta clase permite manejar el mapa por medio de diversos métodos o funcionalidades: pan (centra el mapa en la posición indicada (x, y)), pan_and_zoom

(centra el mapa en la posición indicada y hace zoom sobre ella), zoom_all (realiza el zoom a todas las capas, es decir, visualiza el mapa entero), zoom_to_box (realiza el zoom a todas las capas en determinada área) y envelope (obtiene el área de las capas del mapa). Este servidor de mapas soporta fuente de datos de tipo vector, ráster, shapefile (archivos .shp y .tif), conexión a Base de Datos PostGIS, proyecciones cartográficas y extensiones basadas en Plugins. Es multi-idioma y permite los formatos GDAL y OGR (conjunto de librerías que dan soporte a diversos formatos ráster y vectoriales respectivamente). **[29,30]**

Todos los servidores de mapas presentan algunas ventajas y algunos inconvenientes. Mapnik se caracteriza por el renderizado de bonitos mapas, con bordes de geometrías limpios y suaves, provisto de un sistema gráfico con anti-aliasing de calidad, posicionamiento inteligente de etiquetas y simbolización SVG escalable. Sin embargo el hecho de soportar solamente los estándares WMS lo convierte en una herramienta inferior al servidor de mapas MapServer. También el rendimiento de Mapnik puede ser tan importante como su belleza, este es muy eficiente en el rendimiento bajo alta carga y la representación de grandes conjuntos de datos.

En un estudio comparativo en cuanto al rendimiento entre varios servidores de mapa de código abierto, realizado en septiembre de 2010 en Barcelona y administrado por el Cuerpo de Ingenieros del Ejército. Se obtuvo resultados muy alentadores para el servidor de mapas Mapnik. Cada servidor de mapas se evaluó bajo los mismos requisitos de hardware. En el resultado de la prueba, MapServer demostró ser el servidor más rápido baja carga moderada, mientras que Mapnik fue rápido en baja carga moderada e incluso más rápido en carga alta. Superando estos dos a otros servidores de mapas evaluados. Las pruebas más relevantes para Mapnik fueron los resultados de la utilización de archivos de forma de datos y PostGIS. **[29,30]** (Ver figuras 1.2 y 1.3).

El primer gráfico representa el comportamiento con archivos de forma (Shapefile):

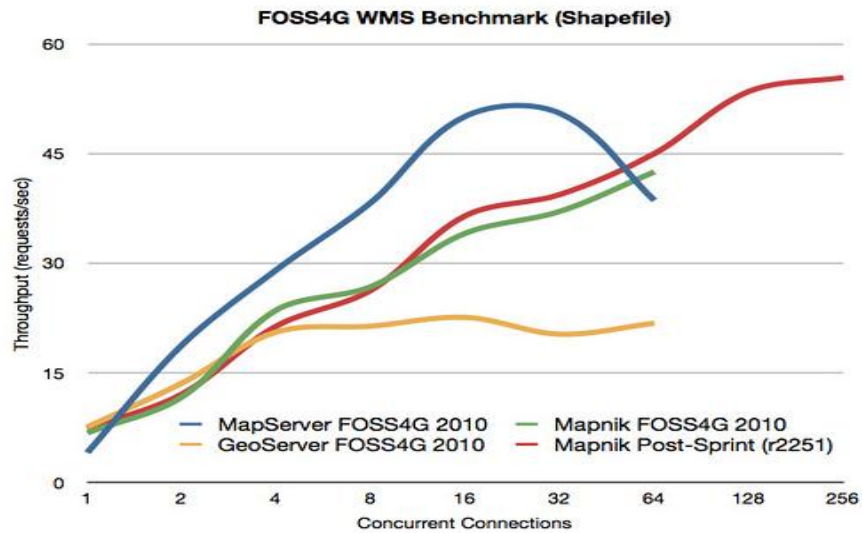


Figura 1.2. Rendimiento de los servidores de mapa con archivos Shapefile.

En la lectura de archivos de forma o shapefile, MapServer es bastante rápido, alcanzando más de 50 peticiones por segundo con carga de 16 conexiones en paralelo, pero luego disminuye el rendimiento en 64 conexiones en paralelo. Mapnik no es tan rápido como MapServer con carga baja, pero en carga alta su rendimiento es más consistente y a mayor carga Mapnik continúa rindiendo a escala lineal. Esto indica que el núcleo de la arquitectura Mapnik, apoyando la rápida prestación de subprocessos múltiples, compensa más lento los tiempos de representación absoluta, y se activa con carga alta. [29,30]

A continuación se muestra el gráfico que representa el comportamiento de PostGIS:

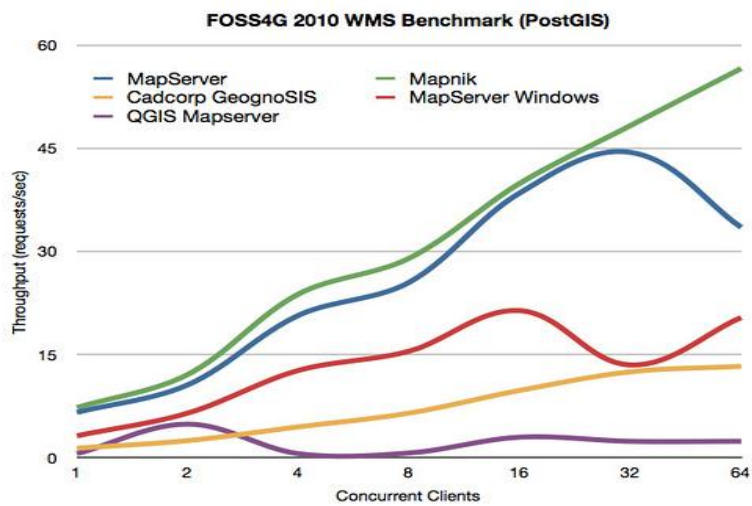


Figura 1.3. Rendimiento de los servidores de mapa con archivos PostGIS.

En esta prueba se destaca Mapnik por la velocidad pura en todos los niveles de carga, así como la escalabilidad con una gran carga. Se puede notar que los resultados de otros servidores variaron significativamente a partir del resultado de Mapnik. [29,30]

Luego de analizar estos resultados es notable que aunque Mapnik no es el servidor de mapas más utilizado a escala mundial y se ubica inferior a MapServer en algunos indicadores, en otros se mantiene muy estable ante el aumento de peticiones e incluso supera a MapServer en cuanto al rendimiento con archivos PostGIS. El comportamiento de Mapnik y sus características lo convierten en un servidor de mapas que puede ser explotado en un SIG Web, como alternativa para lograr un mayor rendimiento. Precisamente por esta razón es que el objetivo principal de esta investigación es desarrollar el prototipo funcional de un SIG Web utilizando el servidor de mapas Mapnik, luego de identificar y estudiar sus capacidades y rendimiento.

Como lenguaje de programación para desarrollar del prototipo SIG Web, es necesario el uso de un lenguaje compatible con el servidor de mapas Mapnik, en este caso se escoge Python. (Ver epígrafe 1.5.1.)

1.5. Lenguajes de programación

Un lenguaje de programación es un idioma artificial diseñado para expresar computaciones que pueden ser llevadas a cabo por computadoras; formado por un conjunto de símbolos y reglas sintácticas y semánticas que definen su estructura y el significado de sus elementos y expresiones. Estos pueden ser de alto o bajo nivel; los lenguajes de bajo nivel comprenden instrucciones simples y cercanas al funcionamiento de la máquina, como por ejemplo el código máquina y el lenguaje ensamblador, por su parte los lenguajes de alto nivel contemplan un elevado grado de abstracción y son más próximo a los humanos, como por ejemplo: Java, C#, C++, Python y otros. Estos son importantes para el desarrollo de software porque permiten crear programas que controlen el comportamiento físico y lógico de una máquina, para expresar algoritmos con precisión, o como modo de comunicación humana, en otras palabras, un lenguaje de programación es un modo práctico para darle instrucciones a un equipo.

1.5.1. Python

Se utiliza Python para el desarrollo del prototipo funcional SIG Web, porque es compatible con el servidor de mapas Mapnik. Este es un lenguaje interpretado, multiplataforma, orientado a objetos, además de ser un lenguaje de alto nivel, que usa tipado dinámico y conteo de referencias para la administración de memoria. También tiene como característica que ofrece un gran rango de funcionalidades y ventajas de programación sencilla, integrada y rápida. Este lenguaje de programación es multiparadigma (orientación a objetos, programación imperativa y, en menor medida, programación funcional). **[42]** Además de Python se utiliza HTML y JavaScript. (Ver epígrafes 1.5.2 y 1.5.3 respectivamente).

1.5.2. HTML

Se utilizó el lenguaje de marcado diseñado para estructurar textos y generar páginas Web, HTML, por sus siglas en inglés: HyperText Markup Language (Lenguaje de Marcas de Hipertexto); específicamente para describir la estructura y el contenido en forma de texto, así como completar el texto con objetos tales como imágenes.

La mayoría de las etiquetas del lenguaje HTML son semánticas. Este es extensible por lo que se pueden añadir características, etiquetas y funciones adicionales para el diseño de páginas Web, generando así un producto rápido y sencillo. **[19]**

Además de ser el lenguaje utilizado para la creación de páginas Web, se puede decir que los documentos HTML no son documentos de texto normal, sino documentos de hipertexto ya que en el propio documento aparecen enlaces a otros documentos. Es un estándar reconocido en todo el mundo. Este posibilita que una misma página HTML se visualice de forma muy similar en cualquier navegador de cualquier sistema operativo y es un lenguaje reconocido universalmente por lo que permite publicar información de forma global. Además puede ser utilizado en muchas aplicaciones electrónicas como buscadores, tiendas online y bancos electrónicos para dar estructura a una página Web. **[19]**

1.5.3. JavaScript

JavaScript es un lenguaje de scripts desarrollado por Netscape para incrementar las funcionalidades del lenguaje HTML, el cual permite crear acciones en sus páginas Web. Este lenguaje se usó para programar la parte del cliente. Se utilizaron además dos librerías de este lenguaje JSExt y JSON, la primera del lado del cliente, específicamente para la interfaz de usuario y la última se usó para la petición de los datos. También se utilizó la tecnología AJAX. (Ver epígrafe 1.5.3.3)

Al ser un lenguaje interpretado, no requiere compilación. El navegador del usuario se encarga de interpretar las sentencias JavaScript contenidas en una página HTML y ejecutarlas adecuadamente. Es un lenguaje dinámico, por lo que se puede cambiar totalmente el aspecto de la página a gusto del usuario y orientado a eventos (cuando un usuario pincha sobre un enlace o mueve el puntero sobre una imagen se produce un evento). Mediante JavaScript se pueden desarrollar scripts que ejecuten acciones en respuesta a estos eventos, además permite la programación orientada a objetos. El modelo de objetos de JavaScript incluye los elementos necesarios para que los scripts puedan acceder a la información de una página y puedan actuar sobre la interfaz del navegador. Es soportado por la gran mayoría de los navegadores como Internet Explorer, Netscape, Opera y Mozilla Firefox. Soporta cuatro tipos de datos, pero no es necesario declarar el tipo de las variables, argumentos de funciones ni valores de retorno de las funciones. [41]

1.5.3.1. ExtJS

JS Ext o ExtJS es una librería JavaScript que incluye: componentes UI del alto rendimiento y personalizables, modelo de componentes extensibles, un API fácil de usar y licencias OpenSource y comerciales. Permite crear aplicaciones complejas utilizando componentes predefinidos así como un manejador de layouts, que le posibilita proveer una experiencia consistente sobre cualquier navegador, evitando validar que el código escrito funcione bien en cada uno (Firefox, IE, Safari, etc.). La ventana flotante que provee JS Ext se considera excelente por la forma en la que funciona. Al moverla o redimensionarla solo se dibujan los bordes haciendo que el movimiento sea fluido. Además brinda los siguientes beneficios:

- Balance entre Cliente – Servidor: La carga de procesamiento se distribuye, permitiendo que el servidor, al tener menor carga, pueda manejar más clientes al mismo tiempo.
- Comunicación asíncrona: En este tipo de aplicación el motor de render puede comunicarse con el servidor sin necesidad de estar sujeta a un clic o una acción del usuario, dándole la libertad de cargar información sin que el cliente lo note.
- Eficiencia de la red: El tráfico de red puede disminuir al permitir que la aplicación elija que información desea transmitir al servidor y viceversa, sin embargo la aplicación que haga uso de la pre-carga de datos puede que revierta este beneficio por el incremento del tráfico. **[40]**

1.5.3.2. JSON

JSON, acrónimo de JavaScript Object Notation, es un formato ligero para el intercambio de datos. JSON es un subconjunto de la notación literal de objetos de JavaScript que no requiere el uso de XML. La simplicidad de JSON ha dado lugar a la generalización de su uso, especialmente como alternativa a XML en AJAX. JSON ha sido aceptado por parte de la comunidad de desarrolladores AJAX, debido a la ubicuidad de JavaScript en casi cualquier navegador Web. **[20]**

1.5.3.3. AJAX

AJAX, acrónimo de Asynchronous JavaScript And XML (JavaScript y XML asíncronos), es un conjunto de tecnologías aplicadas de forma concreta, que permite una técnica de desarrollo Web para crear aplicaciones interactivas. Estas se ejecutan en el cliente, es decir, en el navegador del usuario y mantiene comunicación asíncrona con el servidor en segundo plano. De esta forma es posible realizar cambios sobre la misma página sin necesidad de recargarla, esto significa aumentar la velocidad de interacción en la misma, siendo precisamente esta la característica que propició su uso en el desarrollo del sistema. Dentro de las ventajas de AJAX se tienen las siguientes:

- Las páginas son más responsivas: tiene la capacidad de obtener información para actualizar un mismo documento sin que el navegador modifique su estado; de esta forma la página no tiene que recargarse cada vez que se necesite un cambio. Esto

aumenta la interactividad, la velocidad (cuantificable en al menos el tiempo necesario para refrescar una página) y la usabilidad.

- Maximización del uso de las vías de comunicación lleva a una mayor eficiencia del sistema: la técnica permite un mayor uso de utilizar información por demanda sin que el usuario pague un costo por el mismo (no se tiene que refrescar la página) por lo que en realidad sólo tiene que enviarse al buscador la información que el usuario necesita.
- Posee gran rapidez en las operaciones que realiza.
- Se ahorra tiempo de procesamiento en el servidor Web ya que una gran parte del procesamiento se realiza en el lado del cliente.
- Las aplicaciones son más interactivas, responden a las interacciones del usuario más rápidamente, al estilo aplicaciones de escritorio.
- Actualiza porciones de la página en vez de la página completa. **[19]**

1.6. Entorno de Desarrollo Integrado (IDE). Eclipse

Una vez determinado Python como lenguaje principal de programación, es necesario el uso de un Entorno de Desarrollo Integrado (IDE) que sirva como editor del código expresado a través de este lenguaje de programación.

Un Entorno de Desarrollo Integrado o IDE (acrónimo en inglés de Integrated Development Environment), es un programa informático compuesto por un conjunto de herramientas de programación. Consiste básicamente en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica (GUI) y provee un marco de trabajo amigable para la mayoría de los lenguajes de programación tales como C++, Python, Java, C#, Delphi, Visual Basic, etc. Y por supuesto que su uso es obligatorio, sin un IDE es imposible editar el código expresado a través del lenguaje de programación. **[13]**

Se utilizó el IDE Eclipse para el desarrollo del presente sistema. Este es un IDE de código abierto y multiplataforma; desarrollado originalmente por IBM. Emplea módulos (en inglés *plug-in*) para proporcionar toda su funcionalidad al frente de la plataforma de cliente enriquecido. Este mecanismo de módulos es una plataforma ligera para componentes de software.

Además dispone de un Editor de texto con resaltado de sintaxis. La compilación es en tiempo real. Tiene pruebas unitarias con JUnit, control de versiones con CVS, integración

con Ant, asistentes (*wizards*) para creación de proyectos, clases, tests, etc., y refactorización. Asimismo, a través de "plugins" libremente disponibles es posible añadir control de versiones con Subversion e integración con Hibernate. Este IDE fue liberado originalmente bajo la licencia Common Public License, pero después se liberó bajo la licencia Eclipse Public License. La Free Software Foundation ha dicho que ambas licencias son licencias de software libre, pero son incompatibles con Licencia Pública General de GNU (GNU GPL). [13]

1.7. Framework

Al definir el uso de Python como lenguaje de programación y Eclipse como IDE, es necesario el empleo de un Framework que lo soporte.

En el desarrollo de software, un framework es una estructura conceptual y tecnológica de soporte definido, normalmente con artefactos o módulos de software concretos, con base en la cual otro proyecto de software puede ser organizado y desarrollado. Típicamente, puede incluir soporte de programas, bibliotecas y un lenguaje interpretado, entre otros programas para ayudar a desarrollar y unir los diferentes componentes de un proyecto.

Representa una arquitectura de software que modela las relaciones generales de las entidades del dominio. Provee una estructura y una metodología de trabajo la cual extiende o utiliza las aplicaciones del dominio. Para el desarrollo Web de código abierto existen varios framework como Django y Web2py. [12]

1.7.1. Django 1.2

Django es un framework de desarrollo Web de código abierto, escrito en Python, que cumple en cierta medida el paradigma del Modelo Vista Controlador. La meta fundamental de Django es facilitar la creación de sitios Web complejos. Django pone énfasis en la reutilización, la conectividad y extensibilidad de componentes, del desarrollo rápido y del principio de DRY (del inglés Don't Repeat Yourself). Python es usado en todas las partes del framework, incluso en configuraciones, archivos, y en los modelos de datos. Liberado al público bajo una licencia BSD. Otras características de Django son:

- Un mapeador objeto-relacional.
- Una API de base de datos robusta.
- Un sistema extensible de plantillas basado en etiquetas, con herencia de plantillas.

- Un despachador de URLs basado en expresiones regulares.
- Soporte de internacionalización, incluyendo traducciones incorporadas de la interfaz de administración.
- Documentación incorporada accesible a través de la aplicación administrativa (incluyendo documentación generada automáticamente de los modelos y las bibliotecas de plantillas añadidas por las aplicaciones). [33]

Soporte de bases de datos

Respecto a la base de datos, la recomendada es PostgreSQL, pero también son soportadas MySQL y SQLite 3. Django proporciona una abstracción de la base de datos a través de su API que permite crear, recuperar, actualizar y borrar objetos. También es posible que el usuario ejecute sus propias consultas SQL directamente. En el modelo de datos de Django, una clase representa una tabla en la base de datos y las instancias de esta serán las filas en la tabla. [33]

1.7.2. Web2py 1.87.3

Web2py es un framework de desarrollo Web de código abierto, escrito y programable en el lenguaje Python. Su objetivo principal es dar soporte al desarrollo ágil de aplicaciones Web escalables, seguras y portables enfocadas en bases de datos. Dado a que originalmente fue diseñado como una herramienta de enseñanza con énfasis en la facilidad de uso y despliegue, no tiene ningún archivo de configuración a nivel de proyecto. Se enfoca en desarrollo rápido, favoreciendo el enfoque convención sobre configuración y siguiendo el patrón Modelo Vista Controlador (MVC). A su vez, incluye bibliotecas para manejar protocolos. Este framework en cierta medida fue inspirado por el framework Django y al igual que este último, tiene la habilidad de generar formularios desde tablas de la base de datos e incluye un conjunto extensivo de validadores. [7]

Es un framework completo ya que tiene componentes incorporados para todas las funciones principales, incluyendo: pedidos y respuestas HTTP, cookies, sesiones; múltiples protocolos HTML/XML, REST, ATOM y RSS, RTF y CSV, JSON, JSON-RPC y XML-RPC, AMF-RPC (Flash/Flex), y SOAP; API para CRUD (Crear, Eliminar y Actualizar); mecanismos múltiples de autenticación y control de accesos basados en roles; capa de abstracción de base de datos (DAL) que genera dinámicamente SQL y lo ejecuta en múltiples servidores compatibles; caché para escalabilidad basadas en RAM,

disco y memoria caché; soporte para internacionalización; jQuery para programación en Ajax y efectos para la interfaz de usuario y ticket: registro automático de errores con contexto. Aplica buenas prácticas de la Ingeniería de Software: emplea el patrón arquitectónico Model-View-Controller, validación de formularios del lado del servidor, las devoluciones de datos que hacen que el código sea más legible, escalable y fácil de mantener. Se considera que gracias al lenguaje Python, este framework es menos verboso que los frameworks basados en Java, y su sintaxis tiende a ser más limpia que la de los frameworks basados en PHP. Esto hace que las aplicaciones sean más simples de desarrollar, leer y mantener. Corre en los Sistemas Operativos: Windows, Mac, Unix/Linux. [8]

Capa de abstracción de base de datos (DAL)

Web2py implementa una DAL, no un ORM. Un ORM mapea las tablas de una base de datos en clases y los registros en instancias de estas clases. La DAL en cambio mapea, de manera transparente, las tablas en instancias de una clase (*Table*) y los registros en instancias (*DAL Set*, *DAL Rows*, *DAL Fields*). Tiene una sintaxis muy similar a un ORM pero es más rápido, y puede manejar casi cualquier expresión SQL. [7,8]

SQLite está incluido en Python y es la base de datos por defecto de web2py. Un cambio en la cadena de conexión permite usar Firebird, IBM DB2, Informix, Ingres, Microsoft SQL Server, MySQL, Oracle, PostgreSQL, y la Google App Engine (GAE). Dentro de sus especialidades están: conexiones a múltiples bases de datos, creación y alteración automática de tablas, manejo automático de transacciones y transacciones distribuidas. [8]

Luego de analizar las características de los framework Web2py y Django, se escoge el primero, porque implementa una capa de abstracción a la base de datos, permitiendo la transparencia en el mapeo de las tablas de forma rápida. Además de estar basado en el propio Django, por tanto hereda muchas de las características de este último como cumplir con el patrón arquitectónico Modelo-Vista-Controlador. Ambos tienen la habilidad de generar formularios desde tablas de la base de datos e incluyen un conjunto extensible de validadores. Sin embargo, Web2py incluye nuevas facilidades que propician la simplicidad en el desarrollo de aplicaciones. Dentro de las nuevas facilidades se puede decir que se enfoca en el desarrollo rápido de aplicaciones Web. Además de ser más completo, sencillo, potente y compacto porque tiene componentes incorporados para todas las funciones principales, incluyendo pedidos y respuestas HTTP, cookies, sesiones

y múltiples protocolos, mecanismos múltiples de autenticación y control de acceso basado en roles y su aprendizaje resulta más fácil que el de Django. Cuenta con la librería jQuery para la programación en AJAX, efectos de la interfaz de usuario y ticket. También aplica buenas prácticas de la Ingeniería de Software y es multiplataforma.

1.8. Sistema gestor de Base de Datos. PostgreSQL

Un Sistema Gestor de Base de Datos es un tipo de software muy específico, dedicado a servir de interfaz entre la base de datos, el usuario y las aplicaciones que la utilizan, y controlar la entrada y salida de datos de la base de datos, manteniendo la integridad de la misma. La importancia de su uso es que proporcionan a los usuarios la capacidad de almacenar datos en la base de datos, acceder a ellos y actualizarlos. [14]

Aunque la base de datos por defecto del framework Web2py es SQLite, este permite ser configurado para usar otro Sistema Gestor de Base de Datos como PostgreSQL. Siendo este precisamente el utilizado en el prototipo funcional SIG Web, porque posee una extensión para el trabajo con datos espaciales denominada PostGIS, la cual le añade soporte para objetos geográficos a PostgreSQL. Es decir "habilita espacialmente" el servidor PostgreSQL, permitiendo que sea utilizado como una base de datos back-end espacial para los sistemas de información geográfica (SIG). Además tiene a su disposición el lenguaje Python. Lenguaje principal que se utilizó para programar el prototipo funcional SIG Web. Por tal motivo, para el desarrollo de la presente investigación se usó PostgreSQL con la extensión PostGIS.

PostgreSQL es un sistema objeto-relacional (ORDBMS), OpenSource, que utiliza el lenguaje SQL92/SQL99. Incluye características del paradigma orientado a objetos, como la herencia, tipos de datos, funciones, restricciones, disparadores, reglas e integridad transaccional. Soporta distintos tipos de datos como: datos de tipo fecha, monetarios, elementos gráficos, datos sobre redes y cadenas de bits; además de permitir la creación de tipos propios e incorporar una estructura de datos array, funciones como: manejo de fechas, funciones geométricas y funciones orientadas a operaciones con redes, y también permite la declaración de funciones propias, así como la definición de disparadores. Soporta el uso de índices, reglas y vistas e incluye herencia entre tablas (aunque no entre objetos, ya que no existen), por lo que a este gestor de bases de datos se le incluye entre

los gestores objeto-relacionales y por último permite la gestión de diferentes usuarios, como también los permisos asignados a cada uno de ellos. [14]

1.9. Metodología de desarrollo. Agile Unified Process (AUP)

Una metodología de desarrollo de software en ingeniería de software es un marco de trabajo usado para estructurar, planificar y controlar el proceso de desarrollo en sistemas informáticos.

Existen metodologías ágiles y robustas. Las metodologías ágiles son apropiadas para guiar proyectos de poco volumen que requieran una rápida implementación. Las metodologías robustas pueden ser empleadas para guiar el proceso de desarrollo de proyectos grandes o pequeños, aunque son más apropiadas para proyectos grandes que por su importancia requieren una fuerte planificación. [10]

El prototipo a implementar es sencillo en cuanto a su volumen y su implementación se debe realizar en un tiempo relativamente corto y limitado. Teniendo en cuenta estas características y la definición de metodología ágil y robusta que se detalla anteriormente, lo más indicado es utilizar una metodología ágil.

Extreme Programming (XP) es una metodología ágil, pero para trabajar con ella se recomienda tener experiencia, porque precisamente se basa en la capacidad y madurez del personal. Característica que descalifica su uso para guiar el desarrollo del prototipo funcional de un SIG Web, porque el personal no tiene ninguna experiencia en cuanto al trabajo con ella. Por tanto es necesario seleccionar una metodología ágil con la que el personal tenga experiencia o que se asemeje a otras metodologías que el personal domine.

Teniendo en cuenta lo que se expresa anteriormente, se selecciona para guiar el proceso de desarrollo del prototipo funcional de un SIG Web, la versión ágil de la metodología Rational Unified Process (RUP), denominada Agile Unified Process (AUP). A continuación se detallan sus características:

- AUP abarca siete flujos de trabajo, cuatro ingenieriles y tres de apoyo: Modelado, Implementación, Prueba, Despliegue, Gestión de configuración, Gestión de proyectos y Ambiente. El Modelado agrupa los tres primeros flujos de RUP (Modelamiento del negocio, Requisitos y Análisis y Diseño).

- Dispone de cuatro fases igual que RUP: Creación, Elaboración, Construcción y Transición.
- La metodología AUP se basa en los principios de simplicidad y agilidad: todo se describe concisamente utilizando poca documentación, no miles de ellas. Además de centrarse en actividades de alto valor, no en todo el proyecto. Se puede usar cualquier conjunto de herramientas para trabajar con ella, siempre que sean las más adecuadas.

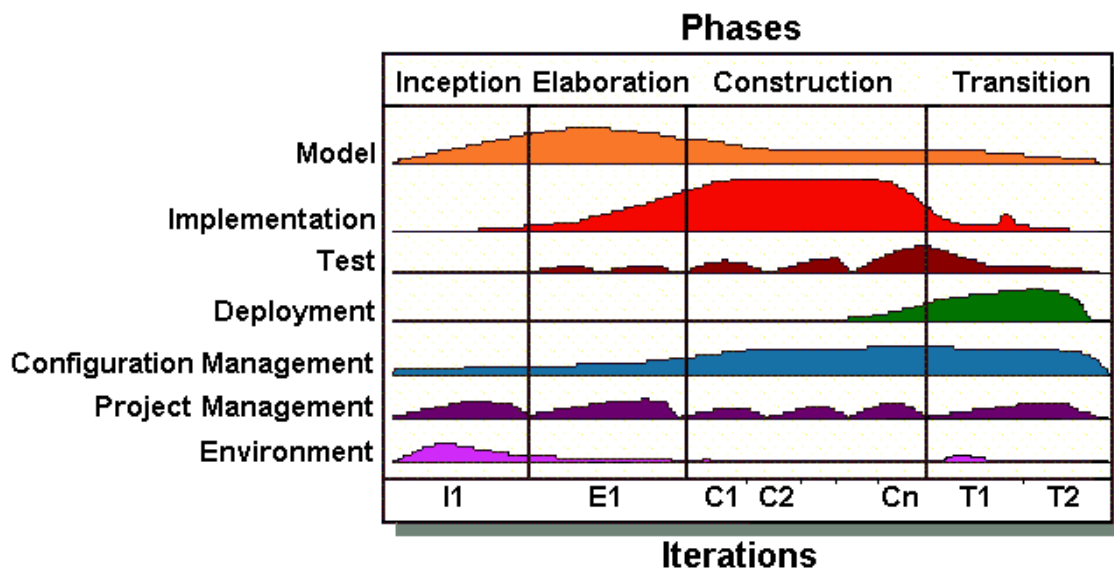


Figura. 1.4. Flujos de trabajo y las fases de AUP.

Descripción de los Flujos de Trabajo ingenieriles:

- Modelado: flujo de trabajo que tiene el objetivo de entender el negocio de la organización, el dominio del problema que se aborda en el proyecto y determinar una solución viable para resolver el problema.
- Implementación: tiene como objetivo transformar los modelos en código ejecutable y realizar un nivel básico de las pruebas, en particular, la unidad de pruebas.
- Prueba: tiene como objetivo realizar una evaluación objetiva para garantizar la calidad: incluye la búsqueda de defectos, validar que el sistema funciona tal como está establecido y verificar que se cumplan los requisitos.
- Despliegue: su objetivo es el plan para la prestación del sistema y la ejecución de dicho plan para que el sistema quede a disposición de los usuarios finales. **[10]**

1.10. Lenguaje de modelado. UML

Un lenguaje de modelado consta de vistas, diagramas, elementos de modelo, símbolos utilizados en los modelos y un conjunto de mecanismos generales o reglas que indican cómo utilizar los elementos.

En todas las disciplinas de la Ingeniería se hace evidente la importancia de los modelos ya que describen el aspecto y la conducta de "algo". Ese "algo" puede existir, estar en un estado de desarrollo o estar todavía en un estado de planeación. Para realizar los modelos del sistema propuesto se hará uso del Lenguaje Unificado de Modelado (UML, por sus siglas en inglés, Unified Modeling Language), porque permite “visualizar, especificar, construir y documentar los artefactos generados en el desarrollo de un sistema”. [5] La visualización, especificación, construcción y documentación de los artefactos generados se realiza a través de diagramas, así como las relaciones entre los diferentes componentes y objetos.

UML se puede aplicar en una gran variedad de formas para dar soporte a una metodología de desarrollo de software, sin especificar en sí mismo qué metodología o proceso usar. Intenta solucionar el problema de propiedad de código que se da con los desarrolladores, al implementar un lenguaje de modelado, y se crea una documentación también común, que cualquier desarrollador con conocimientos de UML será capaz de entender, independientemente del lenguaje utilizado para el desarrollo. Permite la modificación de todos sus miembros mediante estereotipos y restricciones. Su utilización es independiente del lenguaje de programación y de las características de los proyectos, ya que ha sido diseñado para modelar cualquier tipo de proyectos, tanto informáticos como de arquitectura, o de cualquier otra rama. Está consolidado como el lenguaje estándar en el análisis y diseño de sistemas de cómputo y se usa solo para modelar sistemas que usan tecnología orientada a objetos.

Este lenguaje de modelado es necesario para el prototipo funcional del SIG Web que se desea desarrollar. Porque es de fácil entendimiento para todo tipo de roles. Además este lenguaje de modelado genera la documentación de todo el ciclo de desarrollo, con vista a facilitar el mantenimiento, actualización y reutilización.

1.10.1. Herramienta CASE. Visual Paradigm for UML

Las herramientas CASE (Computer Aided Software Engineering, Ingeniería de Software Asistida por Computadora) 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. Mejoran la forma en que ocurre el desarrollo y tienen influencia sobre la calidad del resultado final.

Para apoyar el modelado de los diagramas se hará uso de la herramienta CASE Visual Paradigm for UML, la cual soporta hasta la versión 2.1 de UML y permite modelar los procesos del negocio, la base de datos y las clases del sistema de manera visual. También soporta patrones de diseño para lograr mejores prácticas y permite la conexión a repositorios como: el CVS (Concurrent Versions/Versioning System) y el Subversion.
[10]

Esta herramienta presenta gran usabilidad, porque los diagramas se agrupan por categorías, permitiendo al usuario una rápida localización de la información y configuración de los estilos y formatos de los diagramas incorporando imágenes y estereotipos. Así como la exportación de los diagramas en formato de imagen. Por último se puede decir que es una herramienta multiplataforma, siendo esta la característica que decidió su uso.

1.11. Conclusiones parciales del capítulo

En este capítulo quedaron reflejados los conceptos y la teoría necesaria para comprender todo lo relacionado con el problema que se plantea. Se obtuvo un mayor conocimiento acerca del proceso a automatizar. Se definieron las tecnologías y herramientas libres para desarrollar el prototipo funcional de un SIG Web sobre el servidor de mapas Mapnik. Se definió la metodología de desarrollo AUP para guiar todo el proceso, UML como lenguaje de modelado, Visual Paradigm como herramienta CASE.

Capítulo 2. Modelado del prototipo funcional SIG Web

2.1. Introducción

En este capítulo se exponen los artefactos generados que corresponden al flujo de trabajo Modelado. Estos artefactos ayudan a comprender mejor el desarrollo de la solución propuesta. Permiten ver desde varias perspectivas a través de diagramas y descripciones todo el diseño del prototipo funcional SIG Web. De esta forma se garantiza una buena base para el siguiente flujo de trabajo que es la implementación.

2.2. Modelo de dominio

Como parte del modelado del sistema se decidió realizar modelo de dominio o conceptual, porque no se logró determinar el proceso de negocio, las fronteras no están bien establecidas, no se logran ver claramente, quiénes son las personas que lo inician, quiénes son los beneficiados con cada uno de estos procesos, quiénes son las personas que desarrollan las actividades en cada uno de estos procesos. Pero si se identificaron conceptos y definiciones de estos.

2.2.1. Conceptos y eventos principales del entorno

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

Usuario: Toda persona que necesite trabajar o consultar algún tipo de información incluida en un mapa.

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

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

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

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

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

2.2.2. Diagrama de clases del Modelo de Dominio o Conceptual

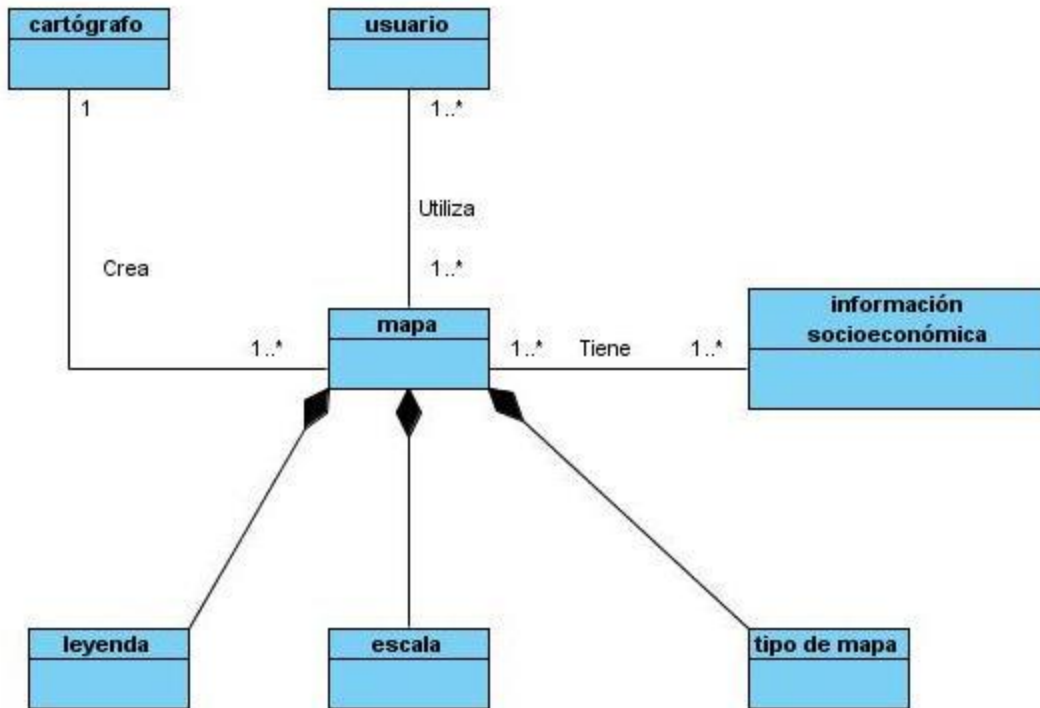


Figura 2.1 Diagrama de clases Del Modelo de dominio

2.3. Requisitos

Como parte del modelado, luego de identificar conceptos y sus definiciones, se definieron los requisitos, entiéndase por requisitos: “condición o capacidad que necesita un usuario para resolver un problema o lograr un objetivo” [5]. Los requisitos se clasifican en funcionales y no funcionales: los requisitos funcionales “son capacidades o condiciones que el sistema debe cumplir, se mantienen invariables sin importar con que propiedades o cualidades se relacionen por lo que no alteran la funcionalidad del producto” [5]. Y los requisitos no funcionales “son las propiedades o cualidades que

el sistema debe tener” [5]. Los requisitos no funcionales son aquellos requisitos que hacen que el sistema sea usable, rápido, confiable y agradable para los usuarios.

2.3.1. Requisitos Funcionales

A través de los requisitos funcionales, se puede expresar una especificación más detallada de las responsabilidades del sistema. Con ellos, se pretende determinar de manera clara y concisa lo que debe hacer el sistema siguiendo un enfoque funcional. Definiéndose los siguientes requisitos funcionales:

RF1. El sistema debe mostrar el mapa

Descripción RF 1

El sistema debe permitir mostrar el mapa con el que el usuario desee trabajar.

RF1.1. El sistema debe permitir la selección de capas

Descripción RF 1.1

El sistema debe permitir cambiar las capas seleccionables y visibles de un mapa.

RF2. El sistema debe permitir realizar paneo al mapa.

Descripción RF 2

El sistema debe permitir mover el mapa sin cambiar el campo visual del mismo.

RF3. El sistema debe permitir redimensionar el mapa.

Descripción RF 3

El sistema debe permitir cambiar la forma de visualización del mapa en la pantalla, mediante los efectos de zoom in y zoom out sobre el mapa

RF4. El sistema debe permitir centrar mapa

Descripción RF 4

El sistema debe permitir mover el mapa de forma tal que el centro del mismo quede en la posición que el usuario especifique.

RF5. El sistema debe permitir restaurar el mapa

Descripción RF 5

El sistema debe permitir mostrar el estado normal del mapa cuando el usuario lo desee, después de redimensionar el mapa.

RF 6. El sistema debe permitir medir distancias

Descripción RF 6

El sistema deber permitir medir las distancias entre puntos en el mapa que el usuario desee.

RF 7. El sistema debe permitir exportar el mapa

Descripción RF 7

El sistema debe permitir exportar la vista de un mapa a un fichero en formato PDF.

2.3.2. Requisitos No Funcionales

En muchos casos los requisitos no funcionales son fundamentales en el éxito del producto debido a que forman una parte significativa de la especificación.

Usabilidad

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

Fiabilidad

El sistema informa la presencia de algún error de forma inmediata luego de haberse producido. De esta manera se fortalece la confiabilidad del mismo y se tendrá en cuenta la recuperación ante fallos y errores.

Eficiencia

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

Soporte

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

Restricciones de diseño

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

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

Interfaz

Interfaces de usuario

El sistema debe tener una apariencia profesional y un diseño gráfico sencillo.

Interfaces de hardware

Para las PCs clientes se requiere tengan tarjeta de red. Al menos 256 MB de memoria RAM y 10 GB de disco duro. Procesador 512 MHz como mínimo. Para los servidores, se requiere tarjeta de red. El Servidor debe tener como mínimo 1GB de RAM y 80 GB de disco duro. Procesador 3 GHz como mínimo.

Interfaces de software

La construcción de la aplicación funcionará bajo los conceptos de arquitectura cliente/servidor. Por tanto, el servidor del usuario final debe tener como requisitos mínimos de software los siguientes: para las PCs clientes, un navegador como Mozilla Firefox, Safari u otro navegador que cumpla con los estándares W3C. Sistema operativo GNU/Linux, Windows y Mac OS. Para los Servidores, sistemas operativos GNU/Linux o Windows Server 2000 o superior. PostgreSQL como Sistema Gestor de Base de Datos. PostGIS como extensión de PostgreSQL para soporte de datos espaciales.

Interfaces de comunicación

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

Requisitos de Licencia

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

Requisitos Legales, de Derecho de Autor y otros

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

de las herramientas de desarrollo son libres y del resto, las licencias están avaladas. Como producto se distribuye amparado bajo las normativas legales establecidas en el registro comercial emitido por las entidades jurídicas de la Universidad de las Ciencias Informáticas.

2.4. Diagrama de Caso de Uso del Sistema

Los requisitos funcionales se agruparon en cinco casos de uso del sistema y se determinó como actor del sistema al usuario como se muestra en la siguiente figura.

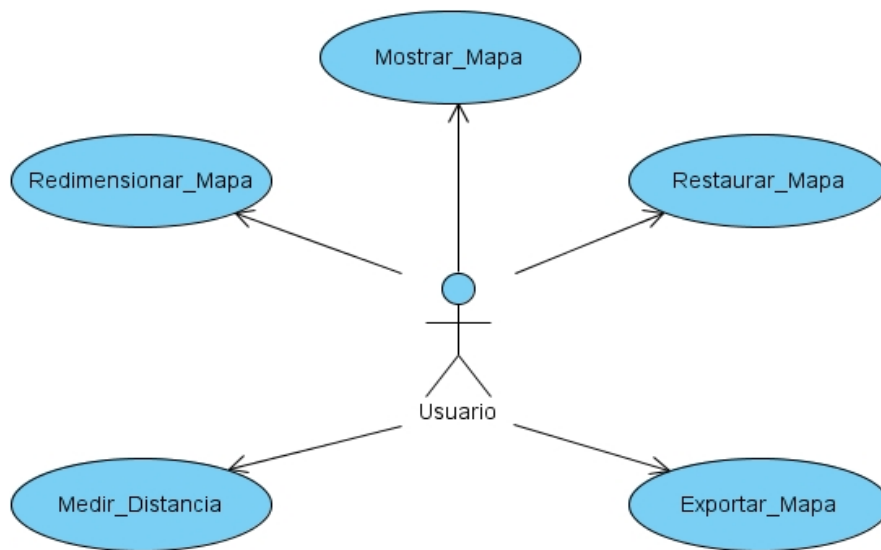


Figura 2.2. Diagrama de Caso de Uso del Sistema

2.4.1. Descripción de los actores del sistema

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

Tabla 2.1. Actores del Sistema

2.4.2. Descripción de los Casos de Uso del sistema

Luego de determinados los requisitos funcionales, éstos se agruparon en casos de uso del sistema. Los casos de uso del sistema se definen como: “fragmentos de funcionalidad del sistema que proporciona al usuario un resultado importante”. [5]

2.4.2.1. Mostrar Mapa

| | |
|--|--|
| Caso de Uso | Mostrar Mapa |
| Actores | Usuario |
| Propósito | Este caso de uso tiene como objetivo mostrar el mapa con el que se desea trabajar y cambiar las capas seleccionables y visibles de un mapa. |
| Resumen | El caso de uso se inicia cuando el usuario desea mostrar el mapa o seleccionar alguna(s) capa(s) en dependencia de su interés y termina cuando el sistema muestra el mapa o modifica el estado de las capas visibles en el mapa. |
| Precondiciones | - |
| Referencias | RF 1, RF1.1 |
| Prioridad | Crítico |
| Flujo Normal de Eventos | |
| Acción del Actor | Respuesta del Sistema |
| 1. El usuario selecciona las capas que desea que se muestren en el mapa mediante la opción “Control de Capas” contenida en el panel izquierdo de la interfaz. (A). | 2. El sistema activa el evento “Control de Capas”. |
| 3. El usuario actualiza su pedido mediante el botón “Actualizar”, contenido en la parte superior del listado de las capas. (B). | 4. El sistema procesa la información y muestra el mapa con el correspondiente pedido del usuario. |

| | |
|---|---|
| Flujos Alternos | |
| Acción del Actor | Respuesta del Sistema |
| 1. El usuario no hace clic en la opción que le brinda el sistema. | 4. El sistema muestra el mapa sin preferencias. |
| Interfaz | |
| | |
| Poscondiciones | |

Tabla 2.2. Descripción CU Mostrar Mapa

2.4.2.2. Redimensionar Mapa

| | |
|----------------|--|
| Caso de Uso | Redimensionar Mapa |
| Actores | Usuario |
| Propósito | Este caso de uso tiene como objetivo cambiar la forma de visualización del mapa en la pantalla. |
| Resumen | El caso de uso se inicia cuando el usuario decide ampliar, minimizar, mover o cambiar el centro del mapa y termina cuando el sistema muestra el resultado en el monitor. |
| Precondiciones | Debe haberse ejecutado el caso de uso "Mostrar Mapa". |
| Referencias | RF 2, RF 3, RF 4 |
| Prioridad | Crítico |

| Flujo Normal de Eventos | |
|--|---|
| Acción del Actor | Respuesta del Sistema |
| <p>1. El usuario selecciona las opciones de redimensionar el mapa:</p> <ul style="list-style-type: none"> • Acercar. (A). • Alejar. (B). • Mover. (C). • Centrar. (D). | <p>2. El sistema realiza la operación según la opción seleccionada por el usuario.</p> <ol style="list-style-type: none"> a. Si seleccionó “Acercar”, ver sección “Acercar”. b. Si seleccionó “Alejar”, ver sección “Alejar”. c. Si seleccionó “Mover”, ver sección “Mover”. d. Si seleccionó “Centrar”, ver sección “Centrar”. |
| Sección “Acercar” | |
| Acción del Actor | Respuesta del Sistema |
| <p>1. El usuario selecciona la opción “Acercar” y da clic en un punto del mapa.</p> | <p>2. El sistema aumentará el zoom y disminuye la escala puntualmente, coge las coordenadas (x, y) y se mueve al centro del mapa el punto donde se dio clic disminuyendo más la escala.</p> |
| | <p>3. El sistema procesa la información y devuelve la visualización del resultado.</p> |
| Sección “Alejar” | |
| Acción del Actor | Respuesta del Sistema |
| <p>1. El usuario selecciona la opción “Alejar” y da clic en un punto del mapa.</p> | <p>2. El sistema disminuirá el zoom y aumenta la escala puntualmente cogiendo las coordenadas (x, y) y aumentando la escala.</p> |
| | <p>3. El sistema procesa la información y devuelve la visualización del resultado.</p> |
| Sección “Mover” | |
| Acción del Actor | Respuesta del Sistema |
| <p>1. El usuario da clic en un punto determinado y arrastra el dispositivo manual.</p> | <p>2. El sistema calcula un ΔX y un ΔY a partir del centro de pantalla y se obtiene un nuevo centro de pantalla que sería a donde se movería el mapa. No varía la escala de representación del mapa.</p> |
| | <p>3. El sistema realiza lo que se describe en el paso 3 del flujo normal de los eventos.</p> |
| Sección “Centrar” | |
| Acción del Actor | Respuesta del Sistema |
| <p>1. El usuario selecciona la opción “Centrar” y da clic</p> | <p>2. El sistema mueve el mapa de forma tal</p> |

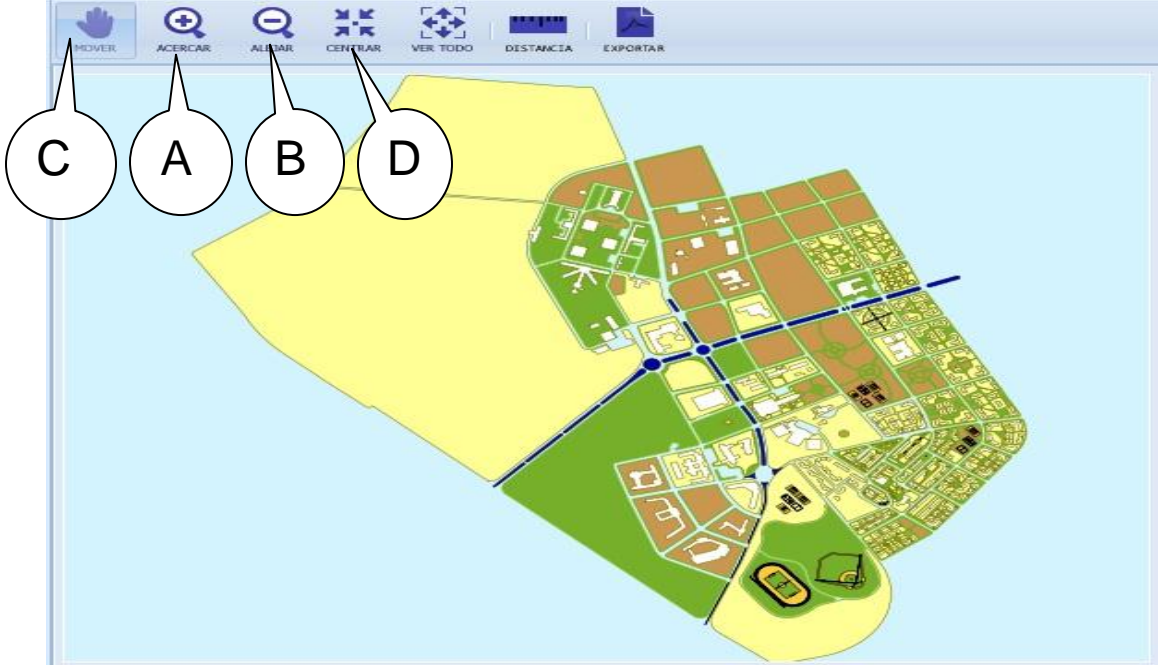
| | |
|--|---|
| en la posición hacia donde desea mover el centro del mapa. | que el centro del mismo quede en la posición que el usuario especificó. |
| Flujos Alternos | |
| Acción del Actor | Respuesta del Sistema |
| Interfaz | |
|  <p>The screenshot shows a map interface with a toolbar at the top. The toolbar contains icons for MOVER (hand), ACERCAR (magnifying glass with plus), ALEJAR (magnifying glass with minus), CENTRAR (crosshair), VER TODO (square with arrows), DISTANCIA (ruler), and EXPORTAR (document). Below the toolbar, four callout bubbles labeled C, A, B, and D point to the MOVER, ACERCAR, ALEJAR, and CENTRAR icons respectively. The map itself shows a city layout with a blue path and a yellow highlighted area.</p> | |
| Poscondiciones | |

Tabla 2.3 Descripción CU Redimensionar Mapa

2.4.2.3. Medir Distancia

| | |
|-------------|---|
| Caso de Uso | Medir Distancia |
| Actores | Usuario |
| Propósito | Este caso de uso tiene como objetivo medir las distancias entre puntos en el mapa. |
| Resumen | El caso de uso se inicia cuando el usuario quiere conocer la distancia existente entre 2 o más puntos situados en el mapa y termina cuando el sistema muestra el resultado de la distancia acumulada. |

| | | |
|---|---|--|
| Precondiciones | Debe haberse ejecutado el caso de uso "Mostrar Mapa". | |
| Referencias | RF 6 | |
| Prioridad | Crítico | |
| Flujo Normal de Eventos | | |
| Acción del Actor | Respuesta del Sistema | |
| 1. El usuario selecciona en la barra de herramientas la opción "Distancia" (A). | 2. El sistema activa el evento "Distancia". | |
| 3. El usuario da clic en el punto inicial de la trayectoria que desea trazar. | 4. El sistema procesa la petición del usuario y activa una barra de información para mostrar la distancia que va acumulando el usuario en su trayectoria. | |
| 5. El usuario continúa la trayectoria dando clic en tantos puntos como desee. | 6. El sistema muestra la trayectoria que va realizando el usuario mediante una línea (B) y actualiza la barra de información con la distancia acumulada entre los puntos consecutivos y desde el punto inicial hasta el último punto especificado por el usuario (C). | |
| 7. El usuario da doble clic indicando que ha terminado de especificar la trayectoria deseada. | 8. El sistema muestra en la barra de información la distancia total acumulada en la trayectoria deseada por el usuario. | |
| Flujos Alternos | | |
| Acción del Actor | Respuesta del Sistema | |
| | | |
| Interfaz | | |

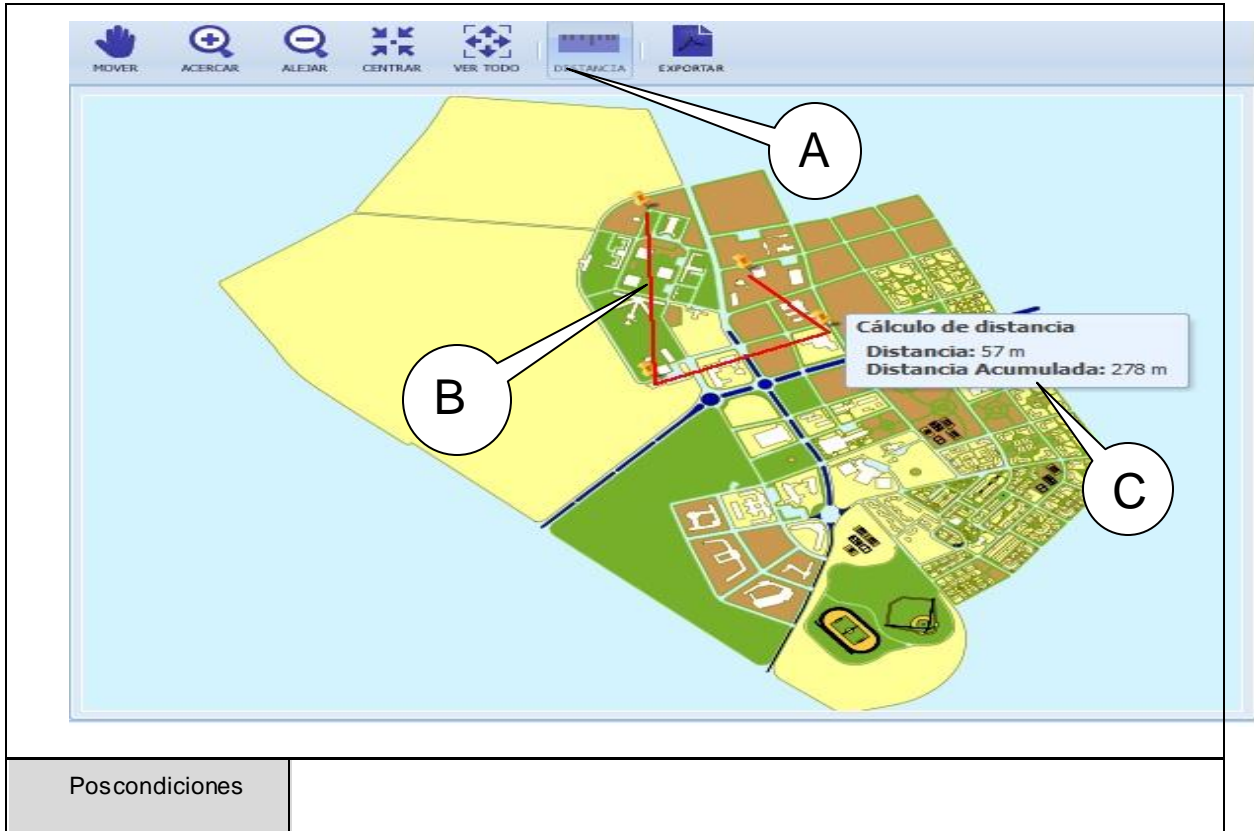
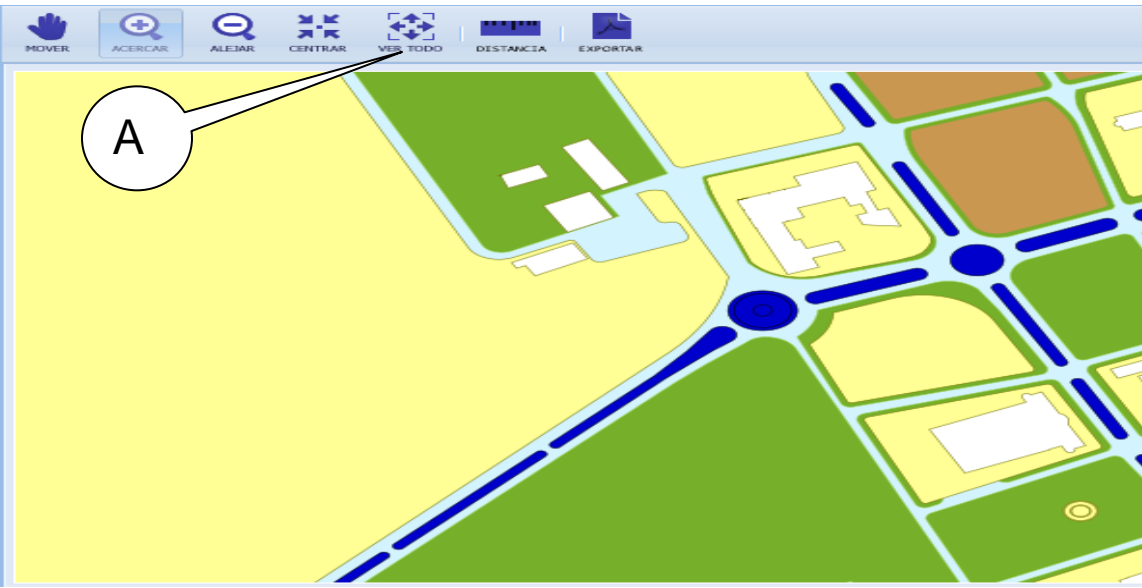


Tabla 2.4. Descripción CU Medir Distancia

2.4.2.4. Restaurar Mapa

| | |
|-------------------------|--|
| Caso de Uso | Restaurar Mapa |
| Actores | Usuario |
| Propósito | Este caso de uso tiene como objetivo mostrar el estado normal del mapa. |
| Resumen | El caso de uso se inicia cuando el usuario quiere ver el estado normal del mapa después de haber realizado diferentes acciones sobre el mismo y termina cuando el sistema muestra el mapa en su condición inicial. |
| Precondiciones | Debe haberse ejecutado el caso de uso "Mostrar Mapa". |
| Referencias | RF 5 |
| Prioridad | Crítico |
| Flujo Normal de Eventos | |

| Acción del Actor | Respuesta del Sistema |
|--|--|
| 1. El usuario selecciona en la barra de herramientas la opción "Ver Todo" (A). | 2. El sistema activa el evento "Ver Todo". |
| | 3. El sistema muestra el mapa en su tamaño normal (B). |
| Flujos Alternos | |
| Acción del Actor | Respuesta del Sistema |
| | |
| Interfaz | |
|  <p>The screenshot shows a map application interface. At the top, there is a toolbar with several icons and labels: 'MOVER' (hand icon), 'ACERCAR' (plus icon), 'ALEJAR' (minus icon), 'CENTRAR' (crosshair icon), 'VER TODO' (four arrows icon), 'DISTANCIA' (ruler icon), and 'EXPORTAR' (document icon). Below the toolbar is a map view showing a street layout with green and yellow areas. A callout bubble labeled 'A' points to the 'VER TODO' button in the toolbar.</p> | |


| | |
|---|--|
|  | |
| Poscondiciones | |

Tabla 2.5. Descripción CU Restaurar Mapa

2.4.2.5. Exportar Mapa

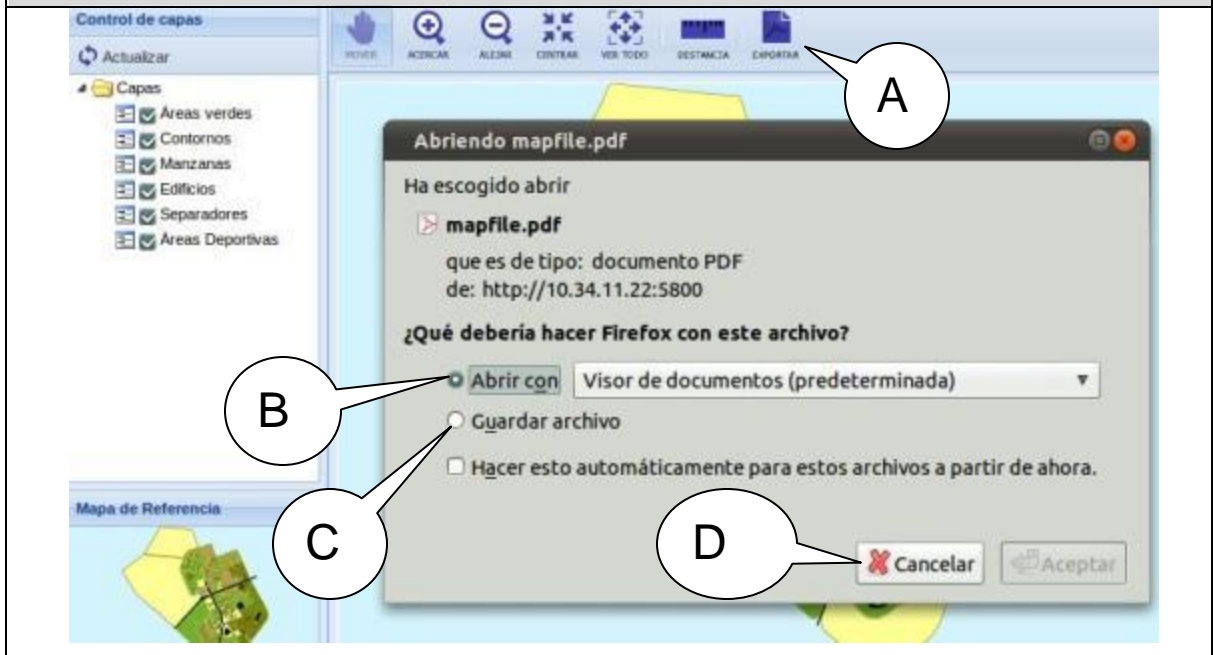
| | |
|-------------------------|--|
| Caso de Uso | Exportar Mapa |
| Actores | Usuario |
| Propósito | Este caso de uso tiene como exportar la vista de un mapa a un fichero en formato PDF. |
| Resumen | El caso de uso se inicia cuando el usuario desea obtener una selección del mapa en un formato PDF y termina con la devolución del fichero en el formato deseado. |
| Precondiciones | Debe haberse ejecutado el caso de uso "Mostrar Mapa". |
| Referencias | RF 7 |
| Prioridad | Crítico |
| Flujo Normal de Eventos | |

| Acción del Actor | Respuesta del Sistema |
|--|---|
| 1. El actor selecciona en la barra de herramientas la opción "Exportar" (A). | 2. El sistema activa el evento "Exportar" y muestra una ventana posibilitando al actor "Abrir" (B) o "Guardar" (C). |
| 3. El actor selecciona la opción que le brinda el sistema | 4. El sistema exporta el mapa a un fichero PDF y brinda la opción de "Abrir" o "Salvar" el mismo. |
| 5. El actor selecciona la opción deseada en dependencia de su necesidad. | 6. El sistema realiza la operación teniendo en cuenta la opción del actor. <ul style="list-style-type: none"> • Si seleccionó "Abrir": El sistema muestra en una nueva ventana el resultado de la operación. • Si seleccionó "Salvar": El sistema salva el fichero PDF listo para imprimirse. |

Flujos Alternos

| Acción del Actor | Respuesta del Sistema |
|---|--|
| 1. El actor hace clic en el botón "Cancelar" (D). | 2. El sistema abandona todo tipo de operación realizada y cierra la ventana. |

Interfaz



| | |
|----------------|--|
| Poscondiciones | |
|----------------|--|

Tabla 2.6. Descripción CU Exportar Mapa

Se definieron todos los casos de uso del sistema como críticos. Los casos de uso críticos “son los más importantes para los usuarios porque cubren las principales tareas o funciones que el sistema ha de realizar y definen la arquitectura básica” [5].

De los casos de uso críticos se consideran arquitectónicamente significativos los siguientes: Mostrar Mapa, Medir Distancia y Exportar Mapa. Los casos de uso arquitectónicamente significativos son: “aquellos que nos ayudan a mitigar los riesgos más importantes, deben ser los más importantes para los usuarios del sistema y nos ayudan a cubrir las funcionalidades significativas” [5]. Para determinar los casos de uso arquitectónicamente significativos, aquellos que como se plantea en su definición, son necesarios desarrollar en las primeras iteraciones, se tuvo en cuenta que cumplieran con los siguientes criterios:

- Casos de uso con dificultad de desarrollo.
- Casos de uso imprescindibles para la puesta en marcha del sistema.
- Organización del desarrollo incremental.
- Disponibilidad de equipo de desarrollo.

2.5. Modelo de Diseño

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

La aplicación que se está desarrollando en el presente trabajo de diploma posee pocos casos de uso (cinco específicamente) con complejidad media y desarrollado por una persona, desempeñando todos los roles. Por lo tanto, se ha decidido no realizar la fase de análisis y desarrollar directamente el diseño de la aplicación.

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

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

2.5.1. Diagramas de Clases del Diseño

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

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

La vista es la encargada de interactuar con el usuario, muestra la información del modelo a los que usan del sistema. El controlador gestiona las entradas del usuario y el modelo se encarga del trabajo con los datos y las reglas del negocio.

El MVC se usa para el diseño de aplicaciones con sofisticadas interfaces. Ayuda a mejorar la reusabilidad del código y además hace que la decodificación en la vista impacte en menor escala en la lógica del negocio o de datos. A continuación se muestran los diagramas de clases divididos por casos de uso:

Diagrama de clases del diseño CU Mostrar Mapa

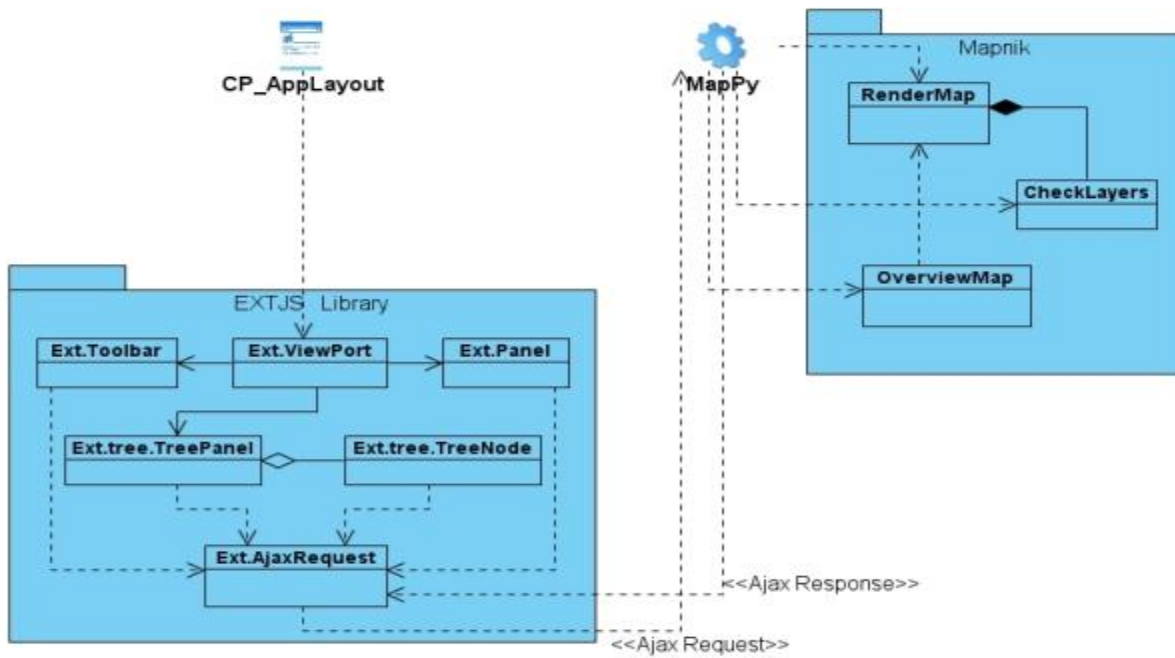


Figura 2.3 Diagrama de clases del diseño del CU Mostrar Mapa

Diagrama de clases del diseño CU Redimensionar Mapa

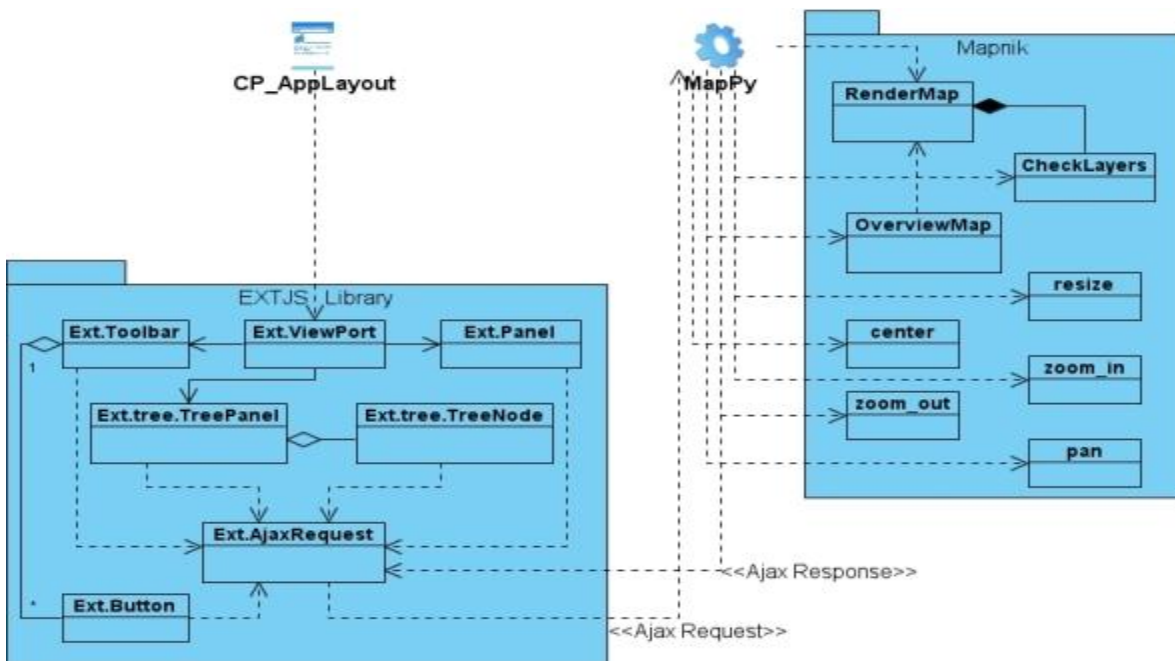


Figura 2.4 Diagrama de clases del diseño del CU Redimensionar Mapa

Diagrama de clases del diseño CU Medir Distancia

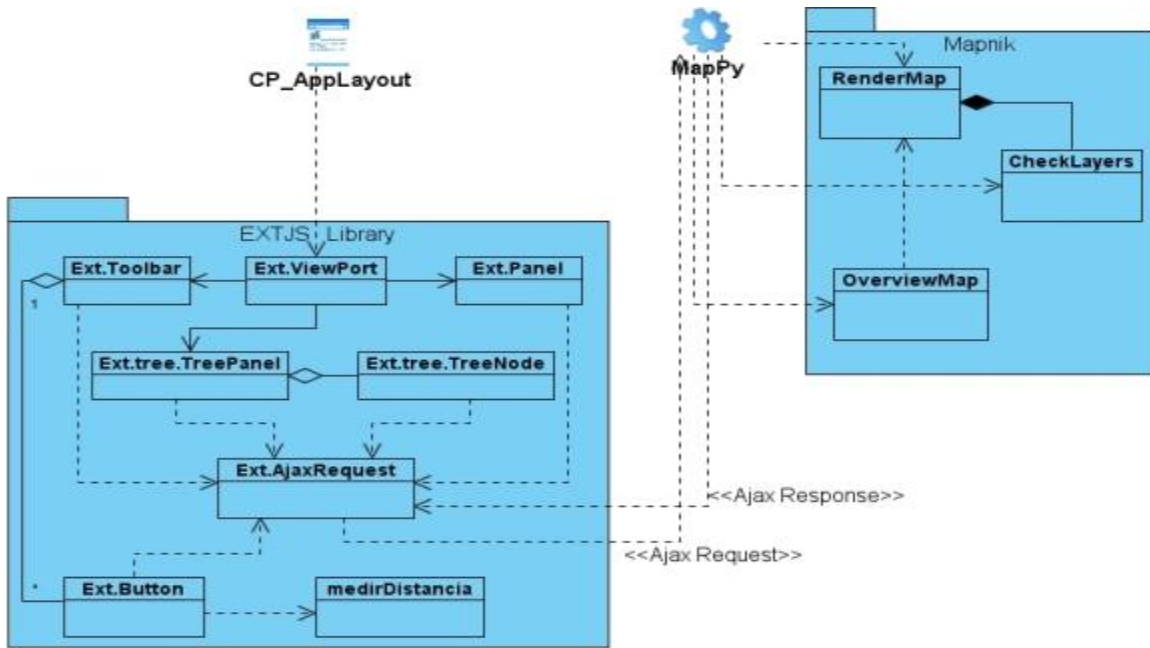


Figura 2.5 Diagrama de clases del diseño del CU Medir Distancia

Diagrama de clases del diseño CU Restaurar Mapa

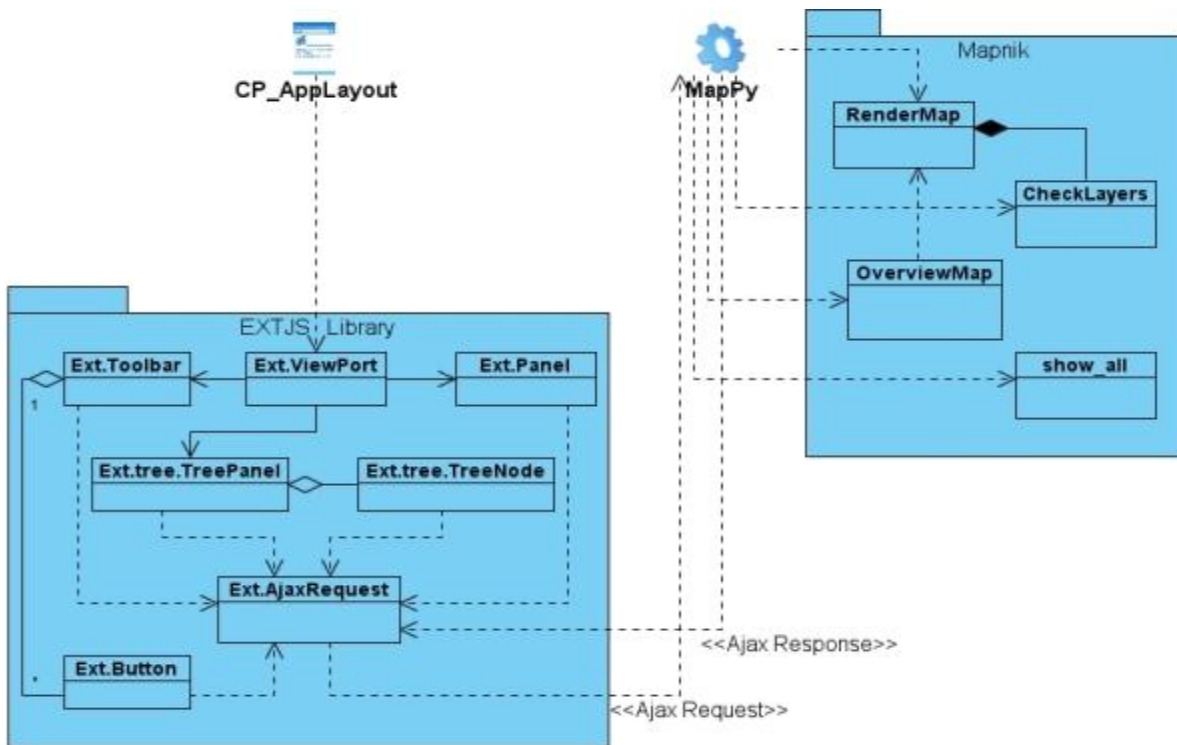


Figura 2.6 Diagrama de clases del diseño del CU Restaurar Mapa

Diagrama de clases del diseño CU Exportar Mapa

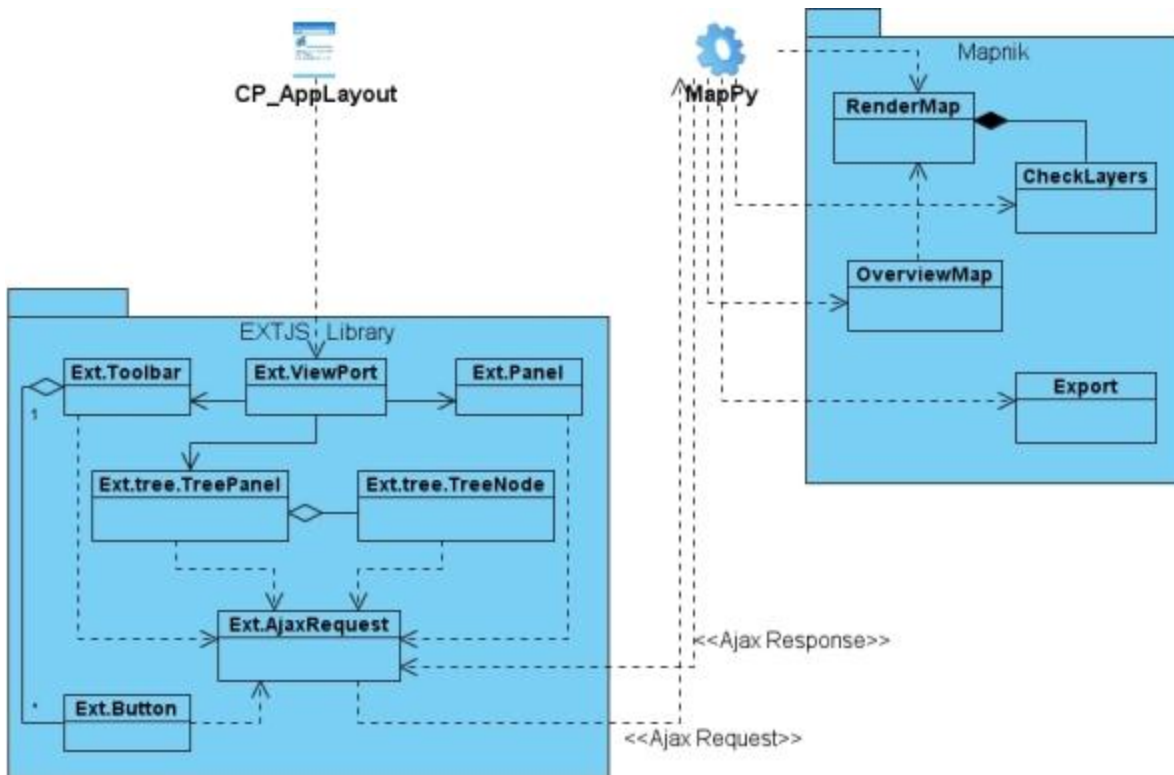


Figura 2.7 Diagrama de clases del diseño del CU Exportar Mapa

2.5.2. Estándares de la interfaz

El diseño de la interfaz es de gran importancia en el desarrollo de un sistema y requiere calidad debido a que es la parte que interactúa con el usuario. Debe ser agradable, poco cargada de información y muy relacionada con el tema que se trata en la aplicación. La calidad de la interfaz contribuye de manera esencial al éxito o el fracaso de un sistema y es por eso que a la hora de su diseño se procede con gran cautela.

Para el diseñar la aplicación se utilizaron algunas funciones de la librería Ext.js, como la función ext.viewport para dividir la interfaz y permitir que esta se ajuste automáticamente al tamaño de la ventana del navegador que se esté usando. También se utilizó ext.toolbar para realizar la barra de herramientas y que el contenido esté organizado de una forma más amena.

2.6. Conclusiones Parciales del capítulo

En este capítulo quedó expresado el modelado del sistema. El modelado del sistema sirvió para comprender mejor la complejidad del prototipo funcional SIG Web que se propone, a través de modelos, diagramas y descripciones, como: el modelo de dominio, descripción de los requisitos funcionales y no funcionales. De esta forma quedó sentada la base para comenzar la implementación.

Capítulo 3. Implementación, Prueba y Factibilidad

3.1. Introducción

Luego de un completo entendimiento de las funcionalidades y procesos del sistema, se procede a la implementación de dichas funcionalidades. En este capítulo también se recogerán algunos artefactos generados en este flujo de trabajo. Estos artefactos ayudan a entender las relaciones entre los elementos de la implementación.

3.2. Modelo de Despliegue

Para entender y representar la relación entre la arquitectura de software y la arquitectura de hardware del sistema se construye el diagrama de despliegue. El modelo o diagrama de despliegue muestra las relaciones físicas de los distintos nodos que componen un sistema y el reparto de los componentes sobre dichos nodos. [31] A continuación se muestra este diagrama.

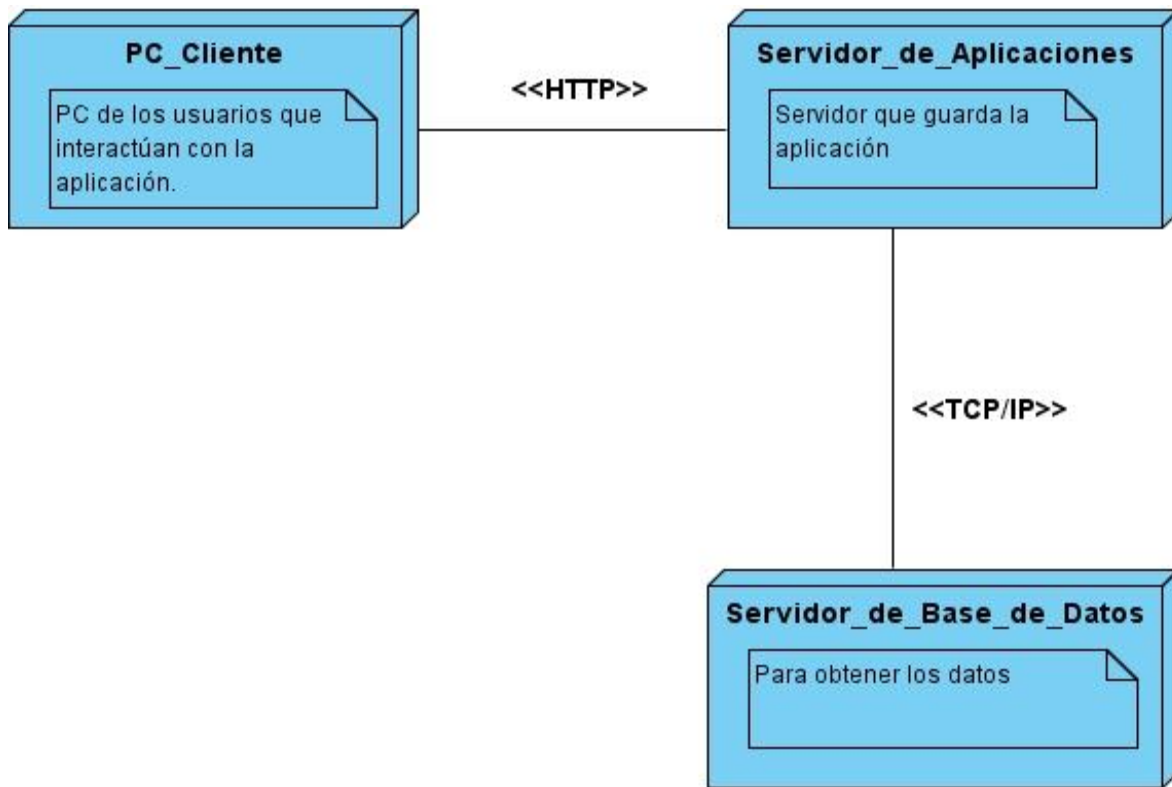


Figura 3.1. Diagrama de despliegue del sistema

3.3. Modelo de Implementación

El modelo de implementación describe cómo los elementos del modelo de diseño se implementan en términos de componentes, como ficheros de código fuentes, ejecutables, etc. En este modelo se realiza el diagrama de componentes, que como su nombre lo indica describe los componentes a construir. A continuación se especifica este modelo:

3.3.1. Diagrama de Componentes

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

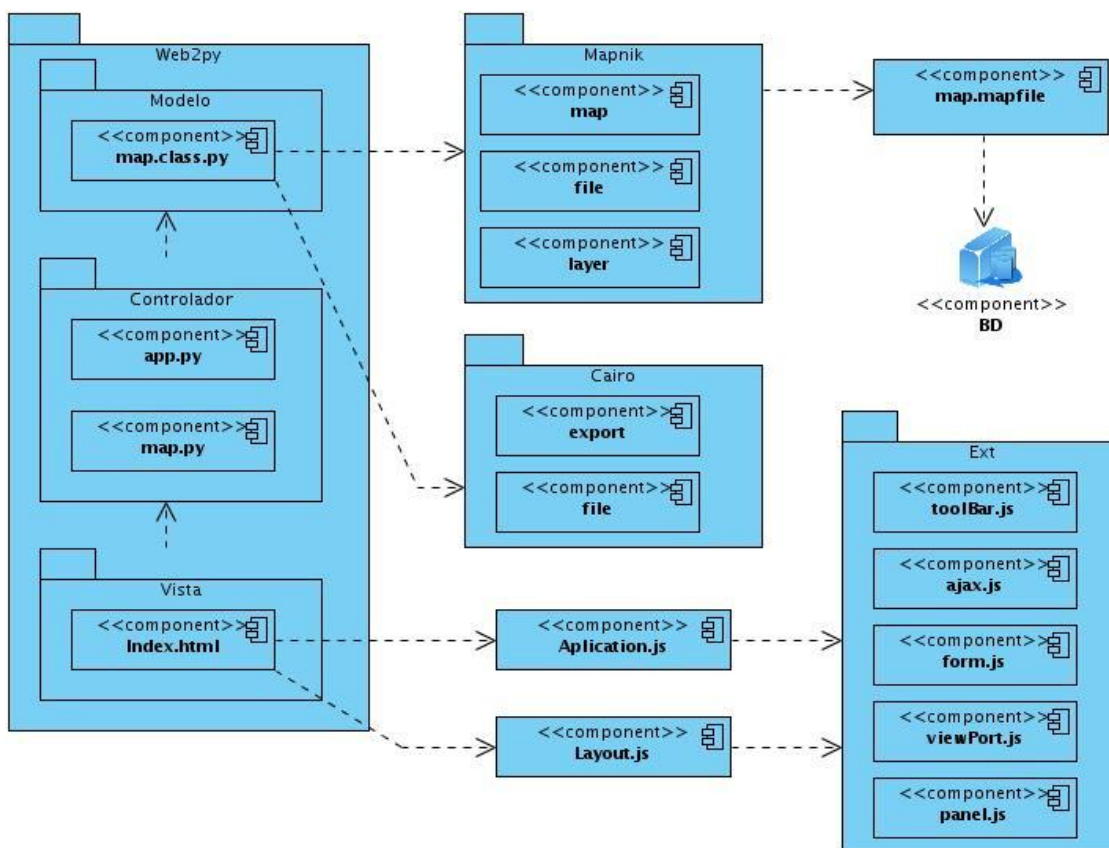


Figura 3.2. Modelo de componentes del sistema

3.4. Prueba

El comportamiento humano no es perfecto y debido a que el desarrollo de un software está sujeto a una gran cantidad de actividades protagonizadas por el hombre, existe la posibilidad de que aparezcan errores en los sistemas. Por tanto surge la necesidad de realizar pruebas para detectar y corregir posibles errores, llamadas pruebas de software.

Las pruebas de software son un elemento crítico para que el producto gane en calidad y fortaleza, capaces de verificar completamente las especificaciones tanto de diseño como de codificación. Estas aunque pertenecen al penúltimo flujo de trabajo ingenieril de AUP, no quiere decir que sea lo último que se realiza. Se puede ir realizando pruebas desde la fase de inicio del software hasta la fase de construcción, siendo esta última fase donde tiene mayor volumen este flujo de trabajo. Existen pruebas de caja blanca y pruebas de caja negra.

Las pruebas de caja blanca son: el tipo de pruebas de software que se realiza sobre las funciones internas de un módulo. Las pruebas de caja blanca están dirigidas a las funciones internas.

Las pruebas de caja negra son el tipo de pruebas de software que ejercitan los requisitos funcionales desde el exterior del módulo. En este tipo de prueba solo interesa su forma de interactuar con el medio que le rodea, entender “qué es lo que hace”, pero sin dar importancia a “cómo lo hace”. Por tanto, deben estar muy bien definidas sus entradas y salidas.

Al prototipo funcional SIG Web propuesto, se le realizaron pruebas de caja negra. A continuación se muestran cinco casos de pruebas realizados a cada uno de los casos de uso del sistema. Se entiende por caso de prueba, según la Ingeniería de software, al conjunto de condiciones o variables bajo las cuáles el analista determinará si el requisito de una aplicación es parcial o completamente satisfactorio.

3.4.1. Método de caja negra

Para probar que las funciones del software son operativas, que los datos insertados son correctos y que se obtiene la respuesta esperada se realizan las pruebas de caja negra. Estas pruebas se llevan a cabo en la interfaz de la aplicación. La realización de esta

prueba se basa en los requisitos funcionales y los errores que principalmente encuentra son de interfaz, de rendimiento, de estructuras de datos y de funciones incorrectas o ausentes.

Caso de prueba: CUS Mostrar Mapa

| Entrada | Resultado | Evaluación de la prueba |
|---|--|------------------------------|
| El usuario selecciona las capas que desea que se muestren en pantalla o selecciona volver a mostrar el mapa original. | Se muestran las capas seleccionadas por el usuario o el mapa original. | <i>Prueba Satisfactoria.</i> |

Caso de prueba: CUS Redimensionar Mapa

| Entrada | Resultado | Evaluación de la prueba |
|--|--|-----------------------------|
| El usuario selecciona la opción Zoom In, Zoom Out o Paneo. | Si el usuario escoge la opción Zoom In, aumenta la resolución del mapa; en el caso de escoger Zoom Out, disminuye la resolución del mapa, en el caso de la opción Paneo permite mover el mapa y si decide Centrar en un punto, el centro del mapa se mueve según su preferencia. | <i>Prueba Satisfactoria</i> |

Caso de prueba: CUS Medir Distancia

| Entrada | Resultado | Evaluación de la prueba |
|--|---|------------------------------|
| El usuario selecciona la opción "Distancia" en la barra de herramientas. | Se muestra una barra de información con el resultado de la distancia. | <i>Prueba Satisfactoria.</i> |

Caso de prueba: CUS Restaurar Mapa

| Entrada | Resultado | Evaluación de la prueba |
|---|--|------------------------------|
| El usuario selecciona la opción "Ver Todo" en la barra de herramientas. | El mapa se muestra en su estado inicial. | <i>Prueba Satisfactoria.</i> |

Caso de prueba: CUS Exportar Mapa

| Entrada | Resultado | Evaluación de la prueba |
|--|--|------------------------------|
| El usuario selecciona la opción "Exportar", en la barra de herramientas. | Se muestra la vista del mapa en un fichero en formato PDF, listo para abrirse o guardarse. | <i>Prueba Satisfactoria.</i> |

3.5. Estudio de factibilidad

No se puede hablar de la planificación de un proyecto como una etapa independiente en un momento determinado, porque aunque es un proceso que se hace necesariamente antes de comenzar a desarrollar el software, durante el avance del proyecto se pueden modificar tareas y reasignar recursos.

El objetivo de la Planificación del proyecto de Software es proporcionar un marco de trabajo que permita al gestor hacer estimaciones razonables de recursos, costos y planificación temporal. Estas estimaciones se hacen dentro de un marco de tiempo limitado al comienzo de un proyecto de software, y deben actualizarse regularmente a medida que progresa el proyecto. Además las estimaciones deben definir los escenarios del mejor caso, y peor caso, de modo que los resultados del proyecto pueden limitarse.

Es muy importante realizar la factibilidad del proyecto para sujetar futuros cambios dentro del software. Por tal motivo es necesario realizarla con gran cautela y con una calidad apropiada para evitar gastos innecesarios de tiempo y de personal. [32]

3.5.1. Análisis de puntos de Casos de uso

El análisis de puntos de CU es un método de estimación del esfuerzo de desarrollo del proyecto que está basado, valga la redundancia, en los CU y en algunas complejidades de ambiente que influyen en el tiempo de progreso del sistema. Se basa en la asignación

de "pesos" a varios factores que afectan el sistema para determinar el tiempo total estimado para el proyecto a partir de dichos factores. [32]

Factor de peso de los actores sin ajustar (UAW)

Este factor consiste en la evaluación de la complejidad de los actores con los que tendrá que interactuar el sistema. Este puntaje se calcula determinando si cada actor es una persona u otro sistema, además evalúa la forma en la que este interactúa con el caso de uso, y la cantidad de actores de cada tipo.

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

Tabla 1. Factor de peso de los actores sin ajustar

$$UAW = \sum (\text{Actores} * \text{Peso})$$

$$UAW = 1 * 3 = 3$$

Factor de peso de los CU sin ajustar (UUCW)

Este punto funciona muy similar al UAW, pero para determinar el nivel de complejidad se puede realizar mediante dos métodos: basado en transacciones o basado en clases de análisis.

El UUCW basado en transacciones toma en cuenta la cantidad de actividades atómicas o transacciones que se pueden realizar en un caso de uso dando un factor de peso correspondiente.

A continuación se muestra el resultado del análisis por UUCW basado en transacciones.

| Tipo de Caso de Uso | Descripción | Factor de Peso |
|---------------------|---|----------------|
| Mostrar Mapa | El Caso de Uso contiene de 4 a 7 Transacciones. | 10 |
| Redimensionar Mapa | El Caso de Uso contiene de 1 a 3 Transacciones. | 5 |

| | | |
|-----------------|---|----|
| Restaurar Mapa | El Caso de Uso contiene de 1 a 3 Transacciones. | 5 |
| Medir Distancia | El Caso de Uso contiene de 4 a 7 Transacciones. | 10 |
| Exportar Mapa | El Caso de Uso contiene de 4 a 7 Transacciones. | 10 |

Tabla 3.2. Factor de peso de los CU sin ajustar

$$UUCW = \sum CU * \text{Peso}$$

$$UUCW = 2 * 5 + 3 * 10 = 40$$

Puntos de casos de uso sin ajustar (UUCP)

El UUCP son los puntos de casos de uso sin ajustar, esto nos puede servir para tener una idea un poco más precisa de la dificultad de los casos de uso e interfaces, tomando en cuenta los pesos de los actores (UAW) y los pesos de los casos de uso (UUCW).

Después de obtener el UAW y el UUCW se puede calcular el UUCP.

$$UUCP = UAW + UUCW$$

$$UUCP = 3 + 40 = 43$$

Una vez calculado el UUCP se deben ajustar los CU a una serie de características técnicas y ambientales.

Cálculo de Puntos de CU ajustados

$$UCP = UUCP \times TCF \times EF.$$

UCP: Puntos de Casos de Uso ajustados.

UUCP: Puntos de Casos de Uso sin ajustar.

TCF: Factor de complejidad técnica.

EF: Factor de ambiente.

Factor de Complejidad Técnica (TCF)

Este factor se compone de 13 puntos que evalúan la complejidad de los módulos del sistema que se desarrolla, cada uno de estos factores tienen un peso definido con los cuales se obtendrán puntos ponderados por cada uno de ellos, según la valoración que se le asigne.

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

Tabla 3.3. Factor de complejidad técnica

$$TCF = 0.6 + 0.01 \times \sum (\text{Peso } i \times \text{Valor asignado } i)$$

$$TCF = 0.6 + 0.01 \times 32$$

$$TFC = 0.6 + 0.032$$

$$TFC = 0.92$$

Factor Ambiente

Estos son los factores sobre los cuales se realiza la evaluación de 8 puntos, que están relacionados con las habilidades y experiencia del grupo de personas involucradas con el desarrollo del proyecto. Estos factores se muestran a continuación:

| Factor | Descripción | Peso | Valor | Total |
|--------|--|------|-------|-------|
| E1 | Familiaridad con el modelo de proyecto utilizado | 1.5 | 3 | 4.5 |
| E2 | Experiencia en la aplicación | 0.5 | 2 | 1 |
| E3 | Experiencia en orientación a objetos | 1 | 5 | 5 |
| E4 | Capacidad del analista líder | 0.5 | 4 | 2 |
| E5 | Motivación | 1 | 5 | 5 |
| E6 | Estabilidad de los requisitos | 2 | 5 | 10 |

| | | | | |
|----|---|----|---|----|
| E7 | Personal part-time | -1 | 3 | -3 |
| E8 | Dificultad del lenguaje de programación | -1 | 3 | -3 |

Tabla 3.4. Factor Ambiente

$$EF = 1.4 - 0.03 \times \Sigma (\text{Peso } i \times \text{Valor asignado } i)$$

$$EF = 1.4 - 0.03 \times 21.5$$

$$EF = 1.4 - 0.645$$

$$EF = 0.76$$

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

$$UCP = UUCP \times TCF \times EF$$

$$UCP = 43 \times 0.92 \times 0.76$$

$$UCP = 30$$

3.5.2. Esfuerzo Total

Este cálculo se realiza con el fin de tener una aproximación del esfuerzo, pensando solo en el desarrollo según las funcionalidades de los casos de uso. Anteriormente, se sugería utilizar 20 horas persona por UCP, pero a través del tiempo se ha ido mejorando. Está basado en los factores ambientales y se calcula de la siguiente manera:

$$E = UCP \times CF$$

E: Esfuerzo estimado en horas-hombre

UCP: Puntos de Casos de Uso ajustados

CF: Factor de conversión

$$E = 30 \times 20$$

$$E = 600 \text{ horas/hombre}$$

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

| Actividad | % esfuerzo | Valor esfuerzo |
|-----------|------------|----------------|
| | | |

| | | |
|----------------|------|-------------------|
| Análisis | 10% | 150 horas-hombre |
| Diseño | 20% | 300 horas-hombre |
| Implementación | 40% | 600 horas-hombre |
| Prueba | 15% | 225 horas-hombre |
| Sobrecarga | 15% | 225 horas-hombre |
| Total | 100% | 1500 horas-hombre |

Tabla 3.5. Horas/Hombre

Teniendo en cuenta que una persona trabaje 8 horas por día, y un mes tiene como promedio 30 días; la cantidad de horas que puede trabajar una persona en un mes es 240 horas. Si el valor del esfuerzo total es de 1500 horas-hombre y por cada 240 horas se tiene un mes eso daría un ET = 6.25 mes-hombre.

Teniendo en cuenta que el proyecto fue realizado por una persona, se puede concluir que el sistema puede desarrollarse en aproximadamente 6 meses, sin tener en cuenta los inconvenientes que ocurran y afecten directamente la realización del producto.

3.6. Conclusiones parciales

Como resultado de este capítulo se obtuvo el modelo de implementación, dando a conocer la forma en se implementan los elementos del modelo de diseño en cuestión de componentes. Esto permitió entender cómo se relacionan entre sí. También se especificó el modelo de despliegue representando la relación entre las arquitecturas de software y hardware. Además se obtuvieron los resultados de los casos de pruebas realizados. Así como la estimación del esfuerzo y los costos del proyecto, guiada por el análisis por puntos de casos de uso. Con la culminación del tercer capítulo se tiene un sistema listo para ser desplegado.

Conclusiones generales

El estudio de las tendencias actuales a nivel mundial en cuanto al desarrollo de SIG y la utilización de servidores de mapas, arrojó lo siguiente: la utilización de SIG se ha extendido a todas las áreas y ramas para solucionar todo tipo de problemas. A medida que han evolucionado los SIG, su utilización es mayor. Las empresas que desarrollan SIG tienden a utilizar MapServer como servidor de mapas.

Luego de estudiar las características de Mapserver y Mapnik, se puede decir que Mapnik constituye una alternativa que puede ser explotado en un SIG generando eficiencia. Mapnik es un servidor de mapas con un excelente rendimiento en cuanto a las peticiones PostGIS.

Se escogieron herramientas libres que faciliten la distribución y evolución del prototipo funcional SIG Web, al dar la posibilidad de modificarlo e incluirle nuevas funcionalidades en futuras versiones.

Se logró modelar y construir un sistema amigable, sencillo, de fácil manipulación para los usuarios, de buena apariencia y con una estructura de interfaz clara y bien distribuida. Eficiente y capaz de brindar respuestas en el menor tiempo posible preservando la calidad requerida. Se obtuvo un producto fiable, capaz de informar al usuario los errores, inmediatamente que son producidos.

Tener en cuenta el patrón arquitectónico Modelo-Vista-Controlador, facilitó una buena organización de las funcionalidades del sistema, contribuyendo a garantizar un mayor tiempo de vida útil del mismo, reutilización y facilidad para el mantenimiento.

Se logró implementar el prototipo funcional SIG Web en el tiempo estimado y según la planificación realizada.

El producto software obtenido da solución al problema planteado porque se demostró con la explotación de Mapnik y el estudio de sus características, que los servidores de mapas pueden garantizar un buen rendimiento en cuanto a la alta carga de transacciones provocadas por el peso de la información geográfica que se maneja por la Web.

El prototipo funcional SIG Web desarrollado sobre el servidor de mapas Mapnik sirve como punto de partida para realizar comparaciones con el resultado obtenido de la explotación de otros servidores de mapas como MapServer en un SIG.

Este prototipo funcional SIG WEB también constituye el punto de partida para crear una tendencia dentro del Proyecto GeneSIG hacia el uso de Mapnik en futuras soluciones informáticas, generando mayor rendimiento de las mismas.

Recomendaciones

La solución propuesta resuelve el problema, aunque existen funcionalidades que no se implementaron en esta primera versión, por esta razón se recomienda que se desarrolle una segunda versión, que incluya otras funcionalidades, con vista a enriquecer la propuesta.

Se recomienda que se preparen desarrolladores en el uso de servidores de mapas como Mapnik para que sean capaces de realizar una segunda versión.

Se recomienda realizar pruebas de caja blanca al prototipo funcional SIG Web. Se le realizaron únicamente pruebas de caja negra.

Referencias bibliográficas

1. PDHL/Cuba. [En línea][Consultado: 20 de abril de 2010]. Disponible en: <http://www.undp.org/cu/pdhl/Documentacion/tesis /procesamiento.html>
2. Fernández Núñez HM. SIG-ESAC: Sistema de Información Geográfica para la gestión de la estadística de salud de Cuba. Rev Cubana Hig Epidemiol 2006; 44(3). [En línea][Consultado: 20 de abril de 2010]. Disponible en: http://bvs.sld.cu/revistas/hie/vol44_3_06/hie03306.htm
3. Páez Moro M, Martínez Fernández P. Metodología para la digitalización de los mapas elaborados en el marco del Proyecto PNUD "Sabana Camagüey». [En línea][Consultado: 20 de abril de 2010]. Disponible en: <http://www.bibliociencias.cu/gsd/collect/eventos/index/assoc/HASH34e9.dir/doc.doc>
4. Universidad de Cienfuegos. Cetas. Contribución al desarrollo agrario y rural con criterios de sostenibilidad en el municipio Aguada de Pasajeros. [En línea][Consultado: 20 de abril de 2010]. Disponible en: <http://cetas.ucf.edu.cu/Proyecto4.htm>
5. Jacobson, Ivar. Booch, Grady. Rumbaugh, James. El Proceso Unificado de Desarrollo. Vol I, Vol II. Editorial Félix Varela. La Habana, Cuba. 2004.
6. Monge de la Cruz, L., Torres Herrera, J., López Chico, L., Navarro Cota, C. (2010): "Análisis comparativo de servidores de mapas", GeoFocus (Informes y comentarios), nº 10, p. 1-10. ISSN: 1578-5157
7. Web2py. [En línea][Consultado el: 29 de octubre de 2010]. Disponible en: <http://es.wikipedia.org/wiki/Web2py>
8. Web2py. [En línea] [Consultado el: 29 de octubre de 2010]. Disponible en: <http://www.web2py.com.ar/examples/default/index>
9. XIV Convención y Feria Internacional. Informática 2011. Tecnologías convergentes: integración e independencia. Póster Electrónico [En línea] [Consultado el: 14 de febrero de 2011] Disponible en: <http://www.informaticahabana.cu/taxonomy/term/2?page=7>
10. Ramírez Martín, Carlos E. y Rodríguez Donatien, Ariagna, junio 2009. Sistema para la Identificación de Aguas en Pozos Petroleros (SIAPP). Ciudad de La Habana: Universidad de las Ciencias Informáticas. Facultad 9.
11. Hernández Sigler, Beatriz, septiembre 2009. Aplicación de un Sistema Información Geográfica para el soporte de la toma de decisiones en el Sistema de Salud a nivel

- primario. CENPALAB. Cuba. [En línea] [Consultado el: 19 de febrero de 2011]
Disponible en: http://www.mappinginteractivo.com/plantilla-egeo.asp?id_articulo=1618
12. Framework. ECURED. [En línea] [Consultado el: 19 de febrero de 2011] Disponible en: <http://www.ecured.cu/index.php/Framework>
 13. Eclipse (software). [En línea] [Consultado el: 19 de febrero de 2011] Disponible en: <http://es.scribd.com/doc/55151939/Taller-de-Rails>
 14. PostgreSQL vs. MySQL. [En línea] [Consultado el: 19 de febrero de 2011] Disponible en: http://danielpecos.com/docs/mysql_postgres/x15.html
 15. Definiciones de dato en la Web. Glosario de Términos Computacionales. [En línea] [Consultado: 01 de junio de 2010]. Disponible en: <http://members.fortunecity.com/miprofe/archivos/TerminosComputacionales.htm>
 16. Definiciones de dato espacial en la Web. Diccionario Geográfico. [En línea][Consultado: 01 de junio de 2010]. Disponible en: http://infoteca.semarnat.gob.mx/website/diccionario/diccionario_d.html
 17. Definiciones de distribución en la Web. [En línea][Consultado: 01 de junio de 2010]. Disponible en: http://www.buzoneo.info/diccionario_marketing/diccionario_marketing_b.php
 18. Definiciones de sistema en la Web. Sistema informático. [En línea][Consultado: 02 de junio de 2010]. Disponible en: [http://es.wikipedia.org/wiki/Sistema_\(informática\)](http://es.wikipedia.org/wiki/Sistema_(informática))
 19. Breve descripción de terminología usada en esta página. CodeBox: Glosario. [En línea][Consultado: 30 de mayo de 2011]. Disponible en: <http://www.codebox.es/glosario>
 20. JSON. [En línea][Consultado: 30 de mayo de 2011]. Disponible en: <http://es.wikipedia.org/wiki/JSON>
 21. Interfaz de programación de aplicaciones. [En línea][Consultado: 30 de mayo de 2011]. Disponible en: http://es.wikipedia.org/wiki/Interfaz_de_programaci%C3%B3n_de_aplicaciones
 22. GNU/Linux. [En línea][Consultado: 30 de mayo de 2011]. Disponible en: <http://es.wikipedia.org/wiki/GNU/Linux>
 23. Mac Os X. [En línea][Consultado: 30 de mayo de 2011]. Disponible en: http://es.wikipedia.org/wiki/Mac_OS_X
 24. Unix. [En línea][Consultado: 30 de mayo de 2011]. Disponible en: <http://es.wikipedia.org/wiki/Unix>

25. Microsoft Windows. [En línea][Consultado: 30 de mayo de 2011]. Disponible en: http://es.wikipedia.org/wiki/Microsoft_Windows
26. Solaris (sistema operativo). [En línea][Consultado: 30 de mayo de 2011]. Disponible en: http://es.wikipedia.org/wiki/Solaris_%28sistema_operativo%29
27. OGC. [En línea][Consultado: 30 de mayo de 2011]. Disponible en: <http://www.opengeospatial.org/&ei=ENXjTfXKHejk0QHRpMS5Bw&sa=X&oi=translate&ct=result&resnum=1&ved=0CCoQ7gEwAA&prev=/search%3Fq%3DOGC%26hl%3Des%26biw%3D1024%26bih%3D574%26prmd%3Divns>
28. Revista internacional de Ciencias de la Tierra. [Especial - Octubre de 2002] [En línea][Consultado: 30 de mayo de 2011]. Disponible en: www.mappinginteractivo.com
29. Presentación final propuesta por el equipo de FOSS4G 2010) [En línea][Consultado: 30 de mayo de 2011]. Disponible en: <http://svn.osgeo/foss4g/benchmarking/scripts/results/benchmarking2010.odp>
30. Entrevista con el Dr.Dane Springmeyer de la rapidez y escalabilidad de Mapnik [En línea] [Consultado: 30 de mayo de 2011]. Disponible en: <http://developmentseed.org/blog/2010/>
31. Weeler, Veronika, 2008. Linux en Español. Entorno de Desarrollo Integrado.
32. Puntos de Caso de Uso. [En línea] [Consultado: 30 de mayo de 2011]. Disponible en: <http://es.wikipedia.org>
33. Django. [En línea] [Consultado: 16 de junio de 2011]. Disponible en: <http://es.wikipedia.org/wiki/Django>
34. MapServer. [En línea] [Consultado: 17 de junio de 2011]. Disponible en: http://www.google.com/cu/url?sa=t&source=web&cd=1&ved=0CCgQFjAA&url=http%3A%2F%2Fes.wikipedia.org%2Fwiki%2FMapServer&rct=j&q=MapServer&tbs=lr%3Alang_1es&ei=QYP7TefoBZKqtweLu83DDg&usq=AFQjCNHZGWYAdO6_1ZK0zixt_-omQo9gsg&cad=rja
35. SIG Agropecuario de la República de Argentina. [En línea] [Consultado: 17 de junio de 2011]. Disponible en: <http://www.sigagropecuario.gov.ar/>
36. Rodriguez Torres, Alexander y Rodriguez Puente, Rafael. Enero 2010. Servicio de mapas temáticos. La Habana, Cuba. Universidad de las Ciencias Informáticas, Facultad 4.
37. QUANTUM GIS /GRASS. [En línea] [Consultado: 17 de junio de 2011]. Disponible en: [http://www.google.com/cu/url?sa=t&source=web&cd=10&ved=0CHMQFjAJ&url=http%](http://www.google.com/cu/url?sa=t&source=web&cd=10&ved=0CHMQFjAJ&url=http%3A%2F%2Fwww.quantum-gis.org%2Fen%2Findex.html)

Bibliografías consultadas

Wikipedia. Cholera. [En línea] [Consultado: 20 de abril de 2010]. Disponible en: http://es.wikipedia.org/wiki/John_Snow

Una epidemia de cólera y los mapas de John Snow. [En línea] [Consultado: 20 de abril de 2010]. Disponible en: <http://www.buscadordenoticias.com.ar/165437/una-epidemia-de-colera-y-los-mapas-de-john-snow.html>

Radio Guáimaro. Sistema de Información Geográfica. Foro. [En línea] [Consultado: 20 de abril de 2010]. Disponible en: <http://www.radioguaimaro.co.cu/index.php/miscelaneas/foro.html?func=view&catid=21&id=119>

PDHL/Cuba. [En línea][Consultado: 20 de abril de 2010]. Disponible en: <http://www.undp.org/cu/pdhl/Documentacion/tesis/procesamiento.html>

Fernández Núñez HM. SIG-ESAC: Sistema de Información Geográfica para la gestión de la estadística de salud de Cuba. Rev Cubana Hig Epidemiol 2006; 44(3). [En línea][Consultado: 20 de abril de 2010]. Disponible en: http://bvs.sld.cu/revistas/hie/vol44_3_06/hie03306.htm

Páez Moro M, Martínez Fernández P. Metodología para la digitalización de los mapas elaborados en el marco del Proyecto PNUD "Sabana Camagüey». [En línea][Consultado: 20 de abril de 2010]. Disponible en: <http://www.bibliociencias.cu/gsd/collect/eventos/index/assoc/HASH34e9.dir/doc.doc>

Universidad de Cienfuegos. Cetas. Contribución al desarrollo agrario y rural con criterios de sostenibilidad en el municipio Aguada de Pasajeros. [En línea] [Consultado: 20 de abril de 2010]. Disponible en: <http://cetas.ucf.edu.cu/Proyecto4.htm>

Hernández, L. y otros. 2005. El paradigma cuantitativo de la investigación científica. Ciudad de la Habana, Cuba. Editorial Universitaria (Eduniv).

Castellanos, S. 1998. La Planificación de la investigación educativa. Ciudad de la Habana, Cuba.

Bayarre, H; Hersford, R. 2004. Metodología de la Investigación. Ciudad de La Habana, Cuba. Editorial Ciencias Médicas.

Álvarez de Zayas, C. 1995. Metodología de la Investigación Científica. Universidad de Oriente, Sgto. De Cuba, Cuba.

Hernández Sampieri, Roberto y otros 2003. Metodología de la Investigación, Ciudad de La Habana, Cuba. Editorial Félix Varela.

Definiciones de dato en la Web. Glosario de Términos Computacionales. [En línea] [Consultado: 01 de junio de 2010]. Disponible en:

<http://members.fortunecity.com/miprofe/archivos/TerminosComputacionales.htm>

Definiciones de dato espacial en la Web. Diccionario Geográfico. [En línea][Consultado: 01 de junio de 2010]. Disponible en:

http://infoteca.semarnat.gob.mx/website/diccionario/diccionario_d.html

Definiciones de distribución en la Web. [En línea][Consultado: 01 de junio de 2010]. Disponible en:

http://www.buzoneo.info/diccionario_marketing/diccionario_marketing_b.php

Definiciones de sistema en la Web. Sistema informático. [En línea][Consultado: 02 de junio de 2010]. Disponible en: [http://es.wikipedia.org/wiki/Sistema_\(informática\)](http://es.wikipedia.org/wiki/Sistema_(informática))

Jacobson, Ivar. Booch, Grady. Rumbaugh, James. El Proceso Unificado de Desarrollo. Vol I, Vol II. Editorial Félix Varela. La Habana, Cuba. 2004.

Monge de la Cruz, L., Torres Herrera, J., López Chico, L., Navarro Cota, C. (2010): "Análisis comparativo de servidores de mapas", GeoFocus (Informes y comentarios), nº 10, p. 1-10. ISSN: 1578-5157

Fernando Mesa Giraldo, Samuel (2009). GeoPython: Soluciones Geoinformáticas Libres basadas en Python

GDAL (2009): Geospatial data abstraction library. [Consulta: 29 de octubre de 2010]. Disponible en <http://www.gdal.org/>

Google (2009): Condiciones de uso del servidor de mapas de servicios gratuitos Google Maps. [Consulta: 29 de octubre de 2010]. Disponible en http://maps.google.com/help/terms_maps.html

Horak, R. (2008): Webster's New World Telecom Dictionary. Editorial Wiley. Indianapolis ISBN: 978-0-471-77457-0, 568 pp.

Mitchell, T. (2005): Web Mapping Illustrated. Editorial O'Really, California. ISBN 0-596-00865-1, 368 pp.

Neumann, A. (2008): Encyclopedia of GIS -Web Mapping and web cartography. Editorial Springer, New York, ISBN: 978-0-387-35975-5, 1263 pp.

OGC (2009): Open Geospatial Consortium. [Consulta: 20-04-2009]. Disponible en <http://www.opengeospatial.org/>

Padrón, D., Prado, E., Chuvieco, E. (2003): Empleo de servidores cartográficos en Internet para la gestión y manejo de desastres. Foro Euro Mediterráneo sobre prevención de catástrofes. Recurso digital

http://www.proteccioncivil.org/es/DGPCE/Informacion_y_documentacion/catalogo/ carpeta_04/foroeuromediterraneo2003/cd/docs/seminario/sesion2/S2-4.pdf

Ramsey, P. (2006): Mashing up the Enterprise [Consultado: 29 de octubre de 2010].

Disponible en: <http://www.refractions.net/expertise/whitepapers/mashups/mashups/>

Web2py. [En línea][Consultado el: 29 de octubre de 2010]. Disponible en:

<http://es.wikipedia.org/wiki/Web2py>

Web2py. [En línea] [Consultado el: 29 de octubre de 2010]. Disponible en:

<http://www.web2py.com.ar/examples/default/index>

XIV Convención y Feria Internacional. Informática 2011. Tecnologías convergentes: integración e independencia. Póster Electrónico. [En línea] [Consultado el: 14 de febrero de 2011] Disponible en: <http://www.informaticahabana.cu/taxonomy/term/2?page=7>

Ramírez Martín, Carlos E. y Rodríguez Donatien, Ariagna, junio 2009. Sistema para la Identificación de Aguas en Pozos Petroleros (SIAPP). Ciudad de La Habana: Universidad de las Ciencias Informáticas. Facultad 9.

Hernández Sigler, Beatriz, septiembre 2009. Aplicación de un Sistema Información Geográfica para el soporte de la toma de decisiones en el Sistema de Salud a nivel primario. CENPALAB. Cuba. [En línea] [Consultado el: 19 de febrero de 2011] Disponible en: http://www.mappinginteractivo.com/plantilla-egeo.asp?id_articulo=1618

Glosario de términos

Datos: Unidad mínima de información, que por si misma no tiene sentido para quien la posee; sin embargo adquiere mayor relevancia cuando conforma un conjunto coherente para el que la interpreta. [15]

Datos espaciales: Objeto o entidad que resulta de una abstracción del espacio geográfico real y que puede ser un rasgo natural, obra humana o alguna abstracción numérica derivada del tratamiento de cifras relacionadas con tal objeto o entidad. Su característica definitiva es la referencia espacial, en dos o tres dimensiones. En algunos casos se consideran sinónimos de dato espacial los siguientes términos: dato geográfico, dato geoespacial, entidad geográfica, objeto espacial, rasgo geográfico y demás que correspondan con esta definición. [16]

Distribución: Es la clasificación y distribución de un grupo de datos incluidos en una categoría definida. [17]

Sistema: Un sistema informático como todo sistema, es el conjunto de partes interrelacionadas, hardware, software y de Recurso Humano (humanware). Un sistema informático típico emplea una computadora que usa dispositivos programables para capturar, almacenar y procesar datos. [18]

Requisitos: Condición o capacidad que necesita un usuario para resolver un problema o lograr un objetivo. [5]

Requisitos funcionales: Son capacidades o condiciones que el sistema debe cumplir, se mantienen invariables sin importar con que propiedades o cualidades se relacionen por lo que no alteran la funcionalidad del producto. [5]

Requisitos no funcionales: Son las propiedades o cualidades que el sistema debe tener. [5]

APIs: interfaz de programación de aplicaciones es el conjunto de funciones y procedimientos (o métodos, en la programación orientada a objetos) que ofrece cierta biblioteca para ser utilizado por otro software como una capa de abstracción. Son usados generalmente en las bibliotecas. [21]

Linux: es uno de los términos empleados para referirse a la combinación del núcleo o *kernel* libre similar a Unix, que es usado con herramientas de sistema GNU. [22]

Solaris: sistema operativo de tipo Unix desarrollado desde 1992 inicialmente por Sun Microsystems y actualmente por Oracle Corporation. [26]

MacOsX: sistema operativo desarrollado y comercializado por Apple Inc. [23]

Unix: sistema operativo portable, multitarea y multiusuario; desarrollado, en principio, en 1969 por un grupo de empleados de los laboratorios Bell de AT&T. [24]

Windows: sistemas operativos desarrollados por Microsoft desde 1981. [25]

OGC: Open Geospatial Consortium, Inc. ® (OGC) es una organización sin fines de lucro, internacional, organización de estándares de consenso voluntario que está liderando el desarrollo de estándares para servicios basados en localización y geoespaciales. [27]

PC: Computadora Personal o Personal Computer

ArcView: nivel de entrada de concesión de licencias de ArcGIS Desktop, un sistema de información geográfica producto de software producido por Esri. [27]

ArcExplore: se usa para la lectura de diversos formatos ArcGIS. [27]

SVG: Gráficos Vectoriales Escalables (del inglés Scalable Vector Graphics) o SVG es una especificación para describir gráficos vectoriales bidimensionales, tanto estáticos como animados (estos últimos con ayuda de SMIL), en formato XML.

Antialiasing: en el área del procesamiento digital de señales en general, se le llama antialiasing a los procesos que permiten minimizar el aliasing cuando se desea representar una señal de alta resolución en un sustrato de más baja resolución.

Aliasing: el aliasing es el efecto que causa que señales continuas distintas se tornen indistinguibles cuando se les muestrea digitalmente. Cuando esto sucede, la señal original no puede ser reconstruida de forma unívoca a partir de la señal digital.

Anexos

Anexo. 1. Entrevista #. 1 realizada al Personal del Proyecto GeneSIG

Esta entrevista se realizó con el objetivo de entender el proceso de desarrollo de los SIG. Ganar en claridad en cuanto a cómo se desarrolla actualmente en el proyecto GeneSIG este tipo de sistema. Así como comprender mejor la necesidad existente para proponer una buena solución y obtener aportes prácticos.

1. ¿El personal del proyecto tiene experiencia en el desarrollo de SIG?
2. ¿Qué servidores de mapas usan en el desarrollo de SIG?
3. ¿Es factible el uso de esos servidores de mapas?
4. ¿Han trabajado en el proyecto con el servidor de mapas Mapnik?
5. ¿Presentan algunas deficiencias los servidores de mapas utilizados? ¿Cuáles?

El resultado de la entrevista ayudó a fundamentar teóricamente la investigación. Se pudo determinar que la tendencia hasta el momento ha sido el uso de MapServer como servidor de mapas en los SIG. Aunque actualmente se exploran y consideran otros servidores de mapas para ser usados como es el caso de Mapnik.

Desde el 2008 que se comenzó a trabajar en esta línea se han obtenidos buenos resultados y experiencias, pero aun faltan cosas por explorar en esta línea.

El uso de servidores de mapas es factible porque optimiza el rendimiento de este tipo de sistema cuando se realizan altas cargas de transacciones en la Web.

MapServer es el servidor de mapas comúnmente utilizado, este es un buen servidor de mapas, pero es necesario estudiar otros servidores de mapas para establecer comparaciones. Se ha trabajado muy poco con Mapnik.

Anexo. 2. Figura del mecanismo para el procesamiento de datos de los SIG

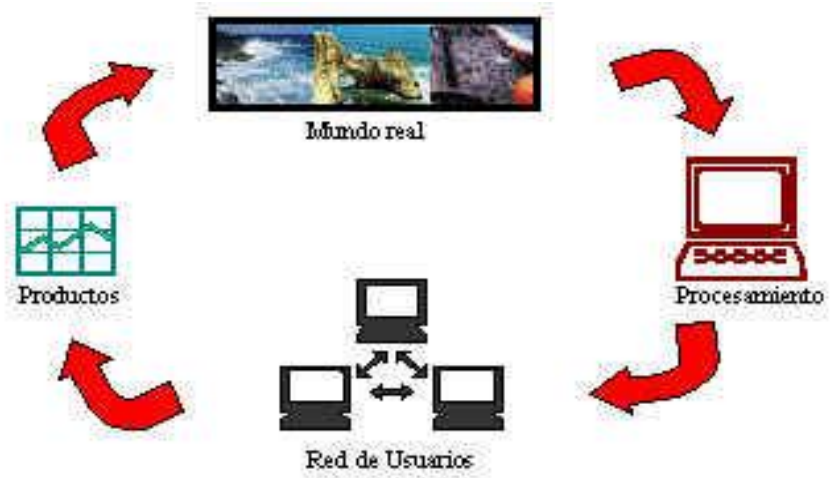


Fig. 1. Mecanismo para el procesamiento de datos de los SIG.