

UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS

Facultad 6



**TÍTULO: Herramienta de representación de
información geográfica sobre tecnologías
libres**

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

AUTORES: Adrián Núñez Rondón

TUTORES: Ing. Yoandry Lazo Nodarse.

Ing. Adrián Gracia

Ciudad de La Habana, 23 de junio del 2011

“Año 53 de la Revolución”



“Los que pueden, lo hacen; los que no, sólo saben quejarse.”

Linus Torvalds



Dedicatoria

Dedico esta investigación en especial a mi madre Inés María Rondón Aliaga por estar siempre apoyándome en los momentos buenos y malos de mi vida, por demostrar dos veces consecutivas que estar sola y tener problemas económicos no es impedimento para que una madre lleve a sus hijos a un nivel de estudio universitario.

A mi hermano por ser un ejemplo a seguir durante mi carrera, por demostrarme que si se puede siempre que te lo propongas.

A mi novia Misbe por estar conmigo en los momentos buenos y malos y apoyarme cada segundo.

Adrián Núñez Rondón.



Agradecimientos

Agradecer a mi mamá y a mi hermano por ser tan especiales para mí, por impulsarme a escoger el camino correcto y no dejarse vencer jamás por las adversidades económicas.

A mi novia Misbe por estar siempre a mí lado en los momentos que más lo necesité, a ti mi corazón te agradezco tantas cosas que no me alcanzaría un año entero para decírtelas. Además por demostrarme que en el amor la suma de uno y uno siempre es uno.

A mis tíos: Henry, Carlo, Sergio y Cesar.

A mis tías que influyeron mucho en que realizara este sueño venciendo todas las adversidades económicas y ayudándonos como familia que somos a ustedes: Idania e Iliana gracias.

Agradezco a mis tutores Yoandri Lazo y Adrián Gracia por su apoyo incondicional y al tribunal de tesis por sus críticas constructivas para lograr que esta investigación saliera con calidad requerida.

Agradezco a mi familia de Pinar del Río que siempre estuvieron pendientes de mí, en especial a Martha, Ariel y Celina.

Agradecer a los tres fantásticos que vienen conmigo desde primaria ellos son: Yayo, Alberto y Tony, que de una forma u otra hicieron que esto fuera posible.

En fin a mis amigos que de una forma u otra hicieron que esto fuera posible entre ellos: Lisandra, Alejandro, Sergio, Elvis, El franio, Chapman, saillet, Rene entre otros que también aportaron su pedacito.



Declaración de Autoría

Declaración de Autoría

Ciudad de La Habana, junio, 2011

“Año 53 de la Revolución”

Yo: Adrián Nuñez Rondón declaro que soy el único autor de este trabajo y autorizo a la Universidad de las Ciencias Informáticas (UCI) a hacer uso del mismo en su beneficio.

Para que así conste firmo la presente a los ____ días del mes de _____ del 2010.

Adrián Nuñez Rondón.

Ing. Yoandry Lazo Nodarse.

FIRMA DEL AUTOR

FIRMA DEL TUTOR

Datos de Contacto

Autor

Nombre: Adrián

Apellidos: Nuñez Rondón.

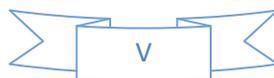
Correo electrónico: anrondon@estudiantes.uci.cu

Tutor

Nombre: Yoandry

Apellidos: Lazo Nodarse.

Correo electrónico: ylazo@uci.cu



Resumen

En la Universidad de las Ciencias Informáticas existen varios proyectos productivos dedicados al desarrollo de los Sistemas de Información Geográfica, destacándose entre ellos el proyecto de investigación y desarrollo llamado GeoQ perteneciente al departamento de Geoinformática, de la Facultad 6, el cual trabaja sobre la base del software libre. Otra de las líneas que existe en el departamento es la línea de SIG para dispositivos móviles, esta línea utiliza en una de sus soluciones el servidor de mapas MapServer al igual que otros productos desarrollados en Geoinformática. Actualmente en GeoQ existen dos aplicaciones SIG de escritorio: Grass (Geographic Resources Analysis Support System) y QuantumGis las cuales presentan dificultades para trabajar con el servidor de mapas MapServer, surgiendo así la dificultad de representar información georreferenciada en una aplicación de escritorio utilizando MapServer; para garantizar la homogeneidad en la representación de mapas en el departamento de Geoinformática es necesario desarrollar una aplicación de escritorio, sobre tecnología libre donde se represente información georreferenciada y que utilice como motor de mapas MapServer, siendo este el objetivo del trabajo de investigación. Para su elaboración se realizó un estudio sobre los Sistemas de Información Geográfica basados en plataformas de escritorio existentes actualmente a nivel mundial, así como las tecnologías actuales sobre el desarrollo de SIG utilizando como servidor de mapas MapServer.

Palabras claves: Sistema de Información Geográfica, MapServer, plataformas de escritorio.

Abstract

At the University of Information Sciences, there are several productive projects dedicated to the development of Geographic Information Systems, prominent among them the research and development project called GeoQ from the department of Geoinformatics, Faculty 6, which works on the basis of free software. Another line in the department there is a line of mobile GIS; this line uses one of its solutions MapServer map server as well as other products developed in Geoinformatics. GeoQ Currently there are two desktop GIS applications: Grass (Geographic Resources Analysis Support System) and QuantumGis which have difficulty working with the Map Server map server, thus resulting in the difficulty to represent geo-referenced information in a desktop application using MapServer. So to ensure uniformity in the representation of maps in the department of Geoinformatics is necessary to develop a desktop application on open technology which represents georeferenced information, which is the objective of this research. For its preparation, a study on Geographic Information System based on currently available desktop platforms worldwide and current technologies on the development of GIS maps as a server using MapServer.

Keywords: Geographic Information System, MapServer, desktop platforms.

Índice de Contenido

Introducción	1
Capítulo 1: Fundamentación Teórica de los Sistemas de Información Geográfica (SIG)	6
1.1 Conceptos asociados al dominio del problema	6
1.2 Proceso de desarrollo de los Sistemas de Información Geográfica.....	8
1.2.1 Antecedentes.....	8
1.2.2 Importancia de los SIG	9
1.2.3 Funcionalidades de los SIG	10
1.2.4 Componentes de un SIG.	11
1.2.5 Funcionamiento de los SIG.....	13
1.2.6 Análisis espacial con SIG	13
1.2.8 Situación Problemática	15
1.3 Análisis de soluciones existentes.....	17
1.4 Conclusiones Parciales.....	22
Capítulo 2: Tecnologías y herramientas a utilizar.	23
2.1 Introducción.	23
2.2 Lenguaje de Programación.....	23
2.3 Biblioteca PyQt.	24
2.4 Sistema Gestor de Bases de Datos	24
2.5 Servidor de Mapas.....	27
2.6 Librería Python-MapScript.	27
2.7 Entorno de desarrollo Integrado(IDE).....	28
2.8 Metodología ágil.....	29
2.9 Arquitectura de Software.....	33
2.10 Conclusiones Parciales.....	34
Capítulo 3: Presentación de la solución propuesta.	35
3.1 Introducción	35
3.2 Fase de Exploración	35

Índice de Contenido

3.2.1 Historias de Usuarios.....	35
3.3 Fase de planificación	41
3.3.1 Estimación de esfuerzos por historia de usuario	42
3.3.2 Plan de iteraciones	43
3.3.3 Plan de duración de las iteraciones.....	44
3.4 Conclusiones Parciales.....	45
Capítulo 4: Construcción de la solución propuesta.....	46
4.1 Introducción.....	46
4.2 Diseño de la solución propuesta.....	46
4.2.1 Tarjetas CRC.....	46
4.3 Desarrollo de las iteraciones.....	50
4.3.1 Iteración 1:.....	50
4.3.2 Iteración 2:.....	52
4.3.3 Iteración 3:.....	53
4.3.4 Iteración 4:.....	54
4.4. Pruebas.....	56
4.4.1 Pruebas unitarias.....	57
4.4.2 Pruebas de aceptación.....	57
4.5 Conclusiones parciales	62
Conclusiones Generales.....	63
Recomendaciones	64
Referencias Bibliográficas	65
Bibliografías Consultadas	67

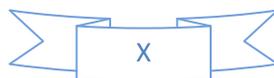
Índice de Figuras y Tablas

Índice de Figuras

Figura 1 Ejemplo de estructura en capas.....	13
Figura 2: Componentes en un sistema PostgreSQL.	25

Índice de Tablas

Tabla #1 Descripción de la HU	36
Tabla #2 Cargar Capas.....	37
Tabla #3 Visualizar Capas.	37
Tabla #4 Mostrar Mapa.	38
Tabla #5 Acercar Mapa.....	38
Tabla #6 Alejar Mapa.....	39
Tabla #7 Escala Normal.....	39
Tabla #8 Mover Mapa.	40
Tabla #9 Exportar a PDF.	40
Tabla #10 Imprimir Imagen.	41
Tabla #11 Estimación de esfuerzo por historia de usuario	43
Tabla #12 Plan de duración de las iteraciones.....	45
Tabla #13 Plantilla para Tarjetas CRC.....	46
Tabla #14 Conexion.....	47
Tabla #15 Principal.....	48
Tabla #16 RedMapa.	49
Tabla #17 Ui_Dialog	49
Tabla #18 edificio.....	50
Tabla #19 HU abordadas en la primera iteración.....	51
Tabla #20 Tarea de la HU #1	51
Tabla #21 Tarea de la HU #2.....	52
Tabla #23 Tarea de la HU #3.....	52
Tabla #24 HU abordadas en la tercera iteración.....	53
Tabla #25 Tarea de la HU #4.....	53
Tabla #26 Tarea de la HU #5.....	54
Tabla #27 Tarea de la HU #6.....	54
Tabla #28 HU abordadas en la cuarta iteración.....	55



Índice de Figuras y Tablas

Tabla #29 Tarea de la HU #7.....	55
Tabla #30 Tarea de la HU #8.....	56
Tabla #31 Tarea de la HU #9.....	56
Tabla #32 Prueba de aceptación para la HU “Cargar Capas.”	58
Tabla #33 Prueba de aceptación para la HU “Mostrar Capas.”	58
Tabla #34 Prueba de aceptación para la HU “Acercar Mapa.”	59
Tabla #35 Prueba de aceptación para la HU “Alejar Mapa.”	59
Tabla #36 Prueba de aceptación para la HU “Escala Normal.”	60
Tabla #37 Prueba de aceptación para la HU “Mover Mapa.”.....	60
Tabla #38 Prueba de aceptación para la HU “Exportar a pdf.”	61
Tabla #39 Prueba de aceptación para la HU “Imprimir Imagen.”	61

Introducción

Introducción

El hombre históricamente ha tenido la necesidad de explorar todo lo que le es desconocido, esta ambición de conocimiento lo indujo a buscar respuestas en lugares que estaban más allá de lo que era conocido hasta ese momento por él. La perseverancia por descubrir nuevas tierras, por tener experiencias inéditas provocó que se realizaran innumerables expediciones. Tantos viajes trajeron consigo un aumento de la información geográfica que dominaban, por lo que se hizo necesario documentar los nuevos conocimientos adquiridos, surgiendo así la cartografía, una de las ciencias más antiguas que existen, que desde sus orígenes fue considerada un arte y aún lo sigue siendo.

Junto al crecimiento de la cartografía, los mapas (principales objetos que genera esta ciencia) y los datos que estos simbolizan, fueron convirtiéndose en un obstáculo, haciéndose más difícil la lectura y comprensión de los mismos por personas que no son expertos en el trabajo con estos elementos. Para mitigar estos problemas actualmente se utilizan las Tecnologías de la Información y la Comunicación (TIC), mediante las cuales se pueden almacenar y manipular gran cantidad de información. Mediante el uso de las TIC la información puede ser modificada, cuantificada y cualificada de manera más eficiente, debido a la posibilidad que brindan los ordenadores para ejecutar varios procesos a la vez.

Dentro del continuo e incesante progreso de las TIC, ha ido evolucionando el Software Libre (SWL), como vía de informatización soberana y aportando una libertad tecnológica aplicable a todas las áreas de la sociedad, repercutiendo así de forma positiva y sumando seguidores que acogen este tipo de tecnología.

Cuba no está ajena a esta realidad y por ser un país bloqueado tiene que hacer un esfuerzo extraordinario para la informatización de la sociedad mediante las TIC. Actualmente las ventajas que ofrecen las tecnologías llegan a todos los rincones de la sociedad sin distinción de posición social o razas y con el uso extensivo del SWL incrementarían los resultados palpables en la economía de la isla.

Introducción

La Universidad de las Ciencias Informáticas (UCI) es un eslabón importante en la economía del país, que contribuye con el alcance de la soberanía tecnológica de la nación y forma parte de todo el proceso de informatización de la sociedad, usando y poniendo en práctica el desarrollo de productos basados en SWL, garantizando la libertad del comercio con empresas sin necesidad de tener que pagar una licencia de software privativo.

En la UCI, específicamente en la Facultad 6 y dentro de esta en el centro de desarrollo Geoinformática y Señales Digitales (GEySED), se ha creado un proyecto de investigación y desarrollo (I+D) llamado GeoQ, que se encarga del desarrollo de sistemas de información geográfica (SIG) de escritorio. El proyecto trabaja sobre la base del software libre, con el objetivo de capturar, manejar, manipular, analizar y modelar datos geográficos representados en disímiles cartografías.

Actualmente en el proyecto se trabajan con dos aplicaciones SIG de escritorio, una de ellas es conocida como Grass (***Geographic Resources Analysis Support System***), la cual no trabaja con el servidor de mapas MapServer¹ y presenta un conjunto de dificultades a la hora de interactuar con la aplicación, debido a que la comunicación con servidor de mapas es a través de comandos y se requiere de un elevado conocimiento, tampoco posee capacidad para cargar archivos con extensión *.map*. La otra aplicación es QuantumGis, una de las aplicaciones SIG más potentes que existe en cuanto a términos de edición de mapas, pero en la versión actual de este producto, no es posible cargar archivos con extensión *.map*; esta herramienta no trabaja con el servidor de mapas MapServer a pesar de la potencia que brinda para la graficación de mapas.

En el departamento de Geoinformática, se ha desarrollado un producto que cumple con las exigencias de un SIG sobre plataforma web. Este sistema informático utiliza como servidor de mapas MapServer, que es sin lugar a dudas muy potente en la representación de mapas a partir de una cartografía existente. Otra de las líneas de investigación y desarrollo que existe en el departamento es la línea de SIG para dispositivos móviles, esta línea utiliza en una de sus soluciones el motor de mapas MapServer.

¹ MapServer: Es un entorno de desarrollo libre para crear aplicación web o escritorio con componente geográfico.

Introducción

Con la intención de lograr una línea de productos que tenga la mayor similitud posible en la representación de los mapas, se hace necesario buscar una alternativa para implementar una solución de escritorio que cumpla con las exigencias del departamento en este sentido.

Lo anteriormente expuesto genera el siguiente **problema a resolver**: Dificultad para representar información georreferenciada en una aplicación de escritorio utilizando MapServer, para garantizar la homogeneidad en la representación de mapas en el departamento de Geoinformática. El **objetivo general** del trabajo es desarrollar una aplicación de escritorio, sobre tecnología libre usando como servidor de mapas MapServer donde se represente información georreferenciada.

La presente investigación tiene como **objeto de estudio** el proceso de referenciar la información geográfica y como **campo de acción** la automatización del proceso de referenciar la información geográfica mediante el uso de MapServer.

Para darle cumplimiento al objetivo propuesto se definen las siguientes **tareas de la investigación**:

1. Elaborar el marco metodológico y teórico en el que está enmarcado el sistema.
2. Realizar propuesta de herramientas que permitan desarrollar la solución que se desea.
3. Diseñar una propuesta de solución.
4. Construir la propuesta de solución.
5. Probar la solución propuesta.

Es por ello que se considera como posibles resultados:

- ✚ Obtener una herramienta de representación de información geográfica sobre tecnologías libres.
- ✚ Documentación teórica del proceso de desarrollo.

Introducción

Definiéndose la siguiente **Idea a defender**: Con la implementación de un Sistema de Información Geográfica sobre plataforma escritorio, usando como servidor de mapas MapServer, se obtendrá un sistema que cumpla con los requerimientos de homogeneidad del departamento Geoinformática.

Métodos de investigación

Métodos teóricos

- ✚ **Analítico–Sintético**: Se realiza un estudio con profundidad de toda la información acerca de las tecnologías, metodologías y herramientas posibles a ser utilizadas en el desarrollo del SIG, logrando definir con mayor certeza las mismas, sintetizando sus características y analizando la viabilidad de cada una.
- ✚ **Histórico-Lógico**: Para llevar a cabo la realización de este trabajo de diploma, primeramente se analiza la historia de los SIG, los principales conceptos que giran alrededor de los mismos, los sistemas existentes en Cuba y el mundo, así como las diversas aplicaciones y funciones de estos.
- ✚ **Modelación**: Se utiliza para realizar los modelos correspondientes al ciclo de vida del sistema, esto permite facilidades a la hora de cumplir con las tareas de análisis y diseño de los procesos que intervienen en la aplicación.

El presente trabajo de diploma consta de 4 capítulos:

Capítulo 1. Fundamentación Teórica: En este capítulo se describen brevemente los conceptos fundamentales relacionados con el dominio del problema. Se realiza un estudio acerca de los SIG, además de hacer un análisis de otras soluciones existentes, ya sea nacional e internacionalmente.

Capítulo 2. Tecnologías y herramientas a utilizar: Se argumenta la selección de las tecnologías y herramientas que se utilizan en el proceso de desarrollo del software.

Capítulo 3. Presentación de la solución propuesta: Este capítulo aborda las dos primeras fases de la Programación Extrema (XP). En la fase de exploración se definen las

Introducción

historias de usuarios, para un mejor entendimiento del software y en la fase de planificación se realiza la estimación de esfuerzos por historia de usuario.

Capítulo 4. Construcción de la solución propuesta: Este capítulo aborda las restantes fases de la Programación Extrema (XP), construcción y prueba. Donde se describen las tarjetas CRC y se detallan las cuatro iteraciones llevadas a cabo durante la etapa de construcción de la aplicación, así como las tareas generadas por cada historia de usuario y las pruebas de aceptación efectuadas sobre el sistema.

CAPÍTULO I: *Fundamentación Teórica*

Capítulo 1: Fundamentación Teórica de los Sistemas de Información Geográfica (SIG).

En este capítulo se tratan los principales conceptos relacionados con el problema, buscando tener un mayor dominio de los aspectos relacionados en este, entre estos conceptos se destaca principalmente el de Sistema de Información Geográfica y otros como software libre. Además se aborda todo lo relacionado con la situación problemática analizando varias soluciones existentes que podrían solucionar el problema en cuestión.

1.1 Conceptos asociados al dominio del problema

Sistema de Información Geográfica (SIG)

El profesor David Rhind (1989) lo ha definido como 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 (georreferenciados), para la solución de los problemas complejos del manejo y planeamiento territorial (2).

Información geográfica

Es un conjunto de datos espaciales, los cuales brindan una información de algún hecho o fenómeno.

Información

Es un conjunto de datos acerca de algún suceso, hecho o fenómeno, que organizados en un contexto determinado tienen su significado, cuyo propósito puede ser el de reducir la incertidumbre o incrementar el conocimiento acerca de algo (1).

Mapa

El concepto de mapa proviene del término latino *mappa*. Se trata de una representación gráfica y métrica de una porción de territorio sobre una superficie bidimensional, que por

CAPÍTULO I: *Fundamentación Teórica*

lo general suele ser plana, aunque también puede ser esférica como en el caso de los globos terráqueos (3).

Mapfile

Archivo de definición del mapa. Contiene la información sobre el tipo de mapa de retorno, la extensión, la escala mínima y máxima, las definiciones de las capas, proyección. Es el corazón de MapServer (9).

Cartografía digital

Es el procedimiento que transforma la información geográfica de los mapas de papel a coordenadas digitales.

Formato ráster

Es una forma de representar la información digital por medio de cuadros llamados píxeles, que es la unidad mínima de medida.

Formato vectorial

Es una forma de representar la información digital por medio de puntos, líneas y polígonos o áreas.

Capa

Es una unidad básica de agrupación de varios mapas que comparten algunas características comunes en forma de temas relacionados con los objetos contenidos en un mapa (4).

Georreferenciación

Se basa en el posicionamiento en el que se concreta la localización de un objeto espacial en un sistema de coordenadas. La misma es muy utilizada en los Sistemas de Información Geográfica (5).

CAPÍTULO I: *Fundamentación Teórica*

Datos Espaciales

Un dato espacial puede entenderse como la representación de un objeto en dos o tres dimensiones, la cual tiene atributos inherentes a nuestro espacio; por lo que, por sí mismo, cuenta con los atributos de dimensión y de localización (6).

Software libre

Software libre se refiere a la libertad de los usuarios para ejecutar, copiar, distribuir, estudiar, cambiar y mejorar el software. De modo más preciso, se refiere a cuatro libertades de los usuarios del software:

1. La libertad de usar el programa, con cualquier propósito.
2. La libertad de estudiar cómo funciona el programa y adaptarlo a tus necesidades. El acceso al código fuente es una condición previa para esto.
3. La libertad de distribuir copias, con lo que puedes ayudar a tu vecino.
4. La libertad de mejorar el programa y hacer públicas las mejoras a los demás, de modo que toda la comunidad se beneficie. El acceso al código fuente es un requisito previo para esto (7).

Código abierto

(En inglés “open source”) es el término con el que se conoce al software distribuido y desarrollado libremente. El código abierto tiene un punto de vista más orientado a los beneficios prácticos de compartir el código que a las cuestiones morales y/o filosóficas las cuales se destacan en el llamado software libre (8).

1.2 Proceso de desarrollo de los Sistemas de Información Geográfica

1.2.1 Antecedentes

En los años 1960 y 1970 nacieron nuevas preferencias en la forma de utilizar los mapas para la valoración de recursos y planificación. Cuando se observaron las diferentes

CAPÍTULO I: Fundamentación Teórica

coberturas sobre la superficie de la tierra, los expertos en el tema se dan cuenta de que no eran independientes entre sí, si no que guardaban algún tipo de relación, se hacía necesario evaluarlas de una forma integrada y multidisciplinaria. Una manera sencilla de hacerlo era superponiendo copias transparentes de mapas de coberturas sobre mesas iluminadas y encontrar puntos de coincidencia en los distintos mapas de los diferentes datos descriptivos.

Como la informática nunca está ausente de cada acontecimiento que ocurre, esta técnica inmediatamente se aplicó a la tecnología informática, trazando mapas sencillos sobre una cuadrícula de papel ordinario, utilizando la sobreimpresión de los caracteres de la impresora por renglones para producir tonalidades de grises adecuadas a la representación de valores estadísticos, en lo que se conocía como sistema de cuadrícula (trama). Pero estos métodos no eran aceptados por los profesionales que usaban este tipo de información cartográfica.

A finales de los años 70`s con el uso de los ordenadores progresó rápidamente en el manejo de información cartográfica y se afinaron muchos de los sistemas informáticos para distintas aplicaciones cartográficas. De igual forma se estaba avanzando en una serie de sectores mezclados entre ellos, edafología, la topografía, la fotogrametría y la percepción remota.

En un principio, este rápido ritmo de desarrollo provocó una gran duplicación de esfuerzos en las distintas disciplinas relacionadas con la cartografía, pero a medida que se aumentaban los sistemas y se adquiría experiencia, surgió la posibilidad de articular los distintos tipos de elaboración automatizada de información espacial, reuniéndolos en verdaderos Sistemas de Información Geográfica para fines generales.

De ahí que el primer SIG desarrollado fue el Sistema de Información Geográfica Canadiense por Roger Tomlinson. En este bajo una estructura ráster y vectorial, combinó la cartografía con los datos necesarios para la gestión forestal.

1.2.2 Importancia de los SIG

CAPÍTULO I: Fundamentación Teórica

Las soluciones para muchos problemas frecuentemente requieren acceso a varios tipos de información que sólo pueden ser relacionadas por geografía o distribución espacial. Sólo la tecnología SIG permite almacenar y manipular información usando geografía, analizar patrones, relaciones y tendencias en la información, todo con el interés de contribuir a la toma de mejores decisiones (11).

1.2.3 Funcionalidades de los SIG

Funciones para la Entrada de Información

En un SIG, las funciones para la entrada de información son las que permiten la introducción, edición y visualización de datos geográficos.

Funciones de Almacenamiento

Las funciones de almacenamiento permitirán este proceso, que no implicará solamente al disco duro del ordenador, de tal forma que a medida que se esté desarrollando un SIG será necesario ir traspasando información a otros dispositivos, así como crear copias de seguridad.

Funciones de Gestión

La información capturada y almacenada en la base de datos (BD) se le podrá adicionar, eliminar y actualizar, además de consultar la BD y extraer las porciones de información espacial que interesen en cada momento. Para ello se utilizan las funciones de gestión, cuya finalidad esencial es permitir la independencia entre la organización física y lógica de los datos.

Funciones de Análisis Espacial

Las funciones de análisis espacial sin duda las más representativas del software de SIG, son las que en definitiva atribuyen valor a los datos geográficos, al revelarnos cosas que de otra forma no se lograra percibir. Conocer y comprender las operaciones espaciales es útil para planificar mejor y de forma más eficiente el trabajo con los SIG.

CAPÍTULO I: *Fundamentación Teórica*

Funciones de Salida o Representación Gráfica y Cartográfica de la Información

Finalmente, las funciones de salida o representación gráfica y cartográfica de la información mediante un SIG, son las que permiten transferir los datos, imágenes o mapas contenidos en él a otro medio o soporte.

De este modo se puede representar los datos almacenados a partir de una serie de criterios (criterios que pueden basarse en capas de información, simbología o el fenómeno que uno quiera representar) que permitirán visualizar la información en función de los objetivos establecidos (10).

1.2.4 Componentes de un SIG.

Los principales componentes de los SIG son:

Hardware.

Corren en todo tipo de computadoras desde equipos centralizados hasta individuales o de red. Permiten la entrada y salida de la información geográfica en diversos medios y formas.

Software.

Proveen las herramientas y las funcionalidades específicas para almacenar, analizar y poder mostrar información geográfica. Los componentes principales del software de un SIG son:

Herramientas para captura y manejo de la información geográfica.

- ✚ Sistema de manejo de Base de Datos.
- ✚ Herramientas para el soporte de consultas, análisis y visualización de los datos geográficos.
- ✚ Interfaz gráfica de usuario para el fácil acceso a las herramientas.

CAPÍTULO I: *Fundamentación Teórica*

Información.

Este es el componente más importante, debido a que se requiere de adecuados datos de soporte para que el SIG pueda resolver los problemas y tomar una mejor decisión.

Personal

Los especialistas que manipulan el sistema, buscan alternativas para mejorarlo desarrollando planes de implementación del mismo y para actualizar la información, necesita un personal experto y que además manipule su hardware y software en todo su potencial.

Métodos

Un SIG solo tendrá una buena implementación si se apoya en un buen diseño y reglas de actividades anteriormente definidas, que son modelos y prácticas operativas exclusivas en cada entidad.

Mapeo escritorio.

Un sistema de mapeo escritorio usa la metáfora del mapa para organizar los datos e interacción con el usuario. El foco de varios sistemas es la creación de mapas: el mapa es la base de datos. La mayoría de los sistemas de mapeo de escritorio tienen capacidades más limitadas de manejos de datos, de análisis espacial y de personalización. Los sistemas de mapeo operan en cómputo de escritorio tales como PC Macintosh y maquinas UNIX más pequeñas.

CAD (Sistemas Asistidos por Computadora).

Los sistemas CAD han evolucionado para crear diseños y planos de construcción e infraestructura. Esta actividad requirió que los componentes de características fijas se armaran para crear una sola estructura. Estos sistemas requieren pocas reglas para especificar como los componentes pueden armarse y muy limitada capacidad de análisis. Los sistemas CAD han sido ampliados como soportes de mapas, pero típicamente tienen una utilidad limitada para manejar o analizar grandes bases de datos geográficas (10).

CAPÍTULO I: *Fundamentación Teórica*

1.2.5 Funcionamiento de los SIG

Los SIG funcionan con dos tipos de información geográfica: el modelo vector y el modelo ráster.

El modelo ráster ha evolucionado para modelar tales características continuas. Una imagen ráster comprende una colección de celdas (píxel) de una grilla más como un mapa o una figura escaneada. Ambos modelos para almacenar datos geográficos tienen ventajas y desventajas únicas y los SIG modernos pueden manejar varios tipos.

En el modelo vector, la información sobre puntos, líneas y polígonos se almacena como una colección de coordenadas x, y . La ubicación de una característica puntual, pueden describirse con un sólo punto x, y . Las características lineales, pueden almacenarse como un conjunto de puntos de coordenadas x, y .

Las características poligonales, pueden almacenarse como un circuito cerrado de coordenadas. El modelo vector es extremadamente útil para describir características discretas, pero menos útil para describir características de variación continua (11).

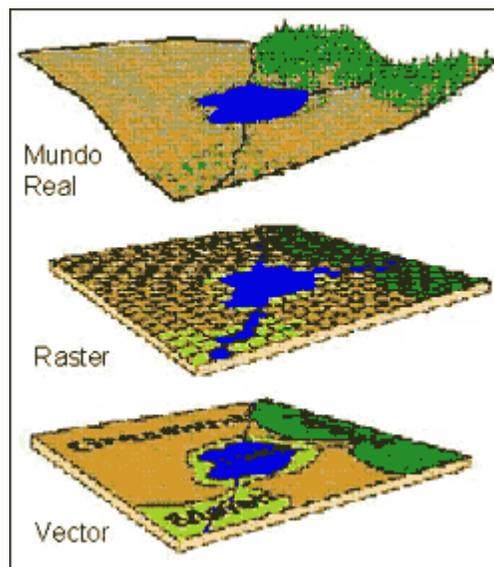


Figura 1 Ejemplo de estructura en capas

1.2.6 Análisis espacial con SIG

CAPÍTULO I: *Fundamentación Teórica*

Si alguna herramienta o potencialidad es intrínseca a la definición de un SIG ésta es el análisis espacial. Ésta está determinada por la existencia de relaciones topológicas entre los elementos y permite realizar cálculos entre variables y obtener así nuevos datos. Las principales herramientas de análisis espacial en un SIG son la superposición, la determinación de áreas de influencia, los análisis de vecindad, el análisis de redes y los modelos del terreno. Todo ello da a los SIG una enorme capacidad de modelización y prospectiva (16).

1.2.7 Tipos de software SIG²

Los SIG están orientados hacia un tipo de tarea establecida, íntegro a esto se puede decir que todos son diferentes, entre estos se encuentran cuatro tipos:

1. Los SIG de escritorio.
2. Los SIG web.
3. Componentes de desarrollo SIG.
4. SIG móviles.

Los **SIG de escritorios** son aquellos diseñados para ser utilizados en ordenadores personales, muchos de estos disponen de un mayor número de herramientas, para una gran variedad de ciencias a las que los SIG se auxilian, entre ellas la arqueología. Obviamente, no todos los software son iguales, los hay dedicados a un aspecto en concreto y otros enfocados a otros aspectos diferentes. Por ello debes de tener claro qué tipo de software usar y qué posibilidades tiene antes de comenzar el proyecto, para no llevaros una sorpresa desagradable a mitad proyecto.

Un **SIG web** permite la visualización de datos y el acceso a funcionalidades de análisis y consulta de servidores SIG a través de Internet o intranet.

² **software SIG**: El software SIG, es un tipo de programa informático, mediante el cual se podrá cargar la cartografía necesaria para nuestro proyecto, manipularla o procesarla y posteriormente darle salida en forma de plano en diferentes escalas según interese en cada momento. Básicamente esto es lo que hace un software SIG (12).

CAPÍTULO I: *Fundamentación Teórica*

Componentes de desarrollo, son herramientas de funciones SIG que se hallan empaquetados, para poder sacarles partido es necesario tener conocimientos de programación para poder implementar estas herramientas al software SIG. Son interesantes para desarrollar herramientas específicas para objetivos específicos y personalizar así el software.

Los **SIG móviles**, son aquellos que están diseñados para usarlos en dispositivos móviles para poder trabajar, las funcionalidades de estos son muy parecidas a las de los SIG de escritorio hace unos años. Ej.gvSIG Mobile (12).

1.2.8 Situación Problemática

Con el surgimiento de Internet en el año 1983, los logros alcanzados en la rama de la informática crecen considerablemente, el desarrollo de las entidades de trabajo generan cada día una enorme cantidad de información, por lo que las empresas están obligadas a utilizar herramientas que le permitan tomar decisiones de forma rápida y eficiente.

De esta forma surgen los Sistemas de Información Geográfica (SIG), permitiendo almacenar y manipular información usando cartografía, analizar patrones, relaciones y tendencias en la información, todo con el interés de contribuir a la toma de mejores decisiones.

En el mundo existen muchos países donde las instituciones hacen uso de los Sistemas de Información Geográfica (SIG), por la importancia que tienen en la manipulación de datos y la toma de decisiones. Las aplicaciones SIG en las esferas de la meteorología, la minería, los dispositivos móviles y en un sin número de esferas de la sociedad hacen que este tipo de soluciones informáticas sea cada vez más demandado por los usuarios a nivel mundial. La principal diferencia de este tipo de aplicación con respecto a otras que también realizan análisis es precisamente su capacidad de representación espacial, lo cual acerca aún más a la realidad estudiada a los especialistas.

Cuba a pesar de ser un país bloqueado tecnológicamente, no se ha quedado ajena a este tipo de tecnologías y con la ayuda del Software Libre (SWL) ha logrado crear instituciones que se dediquen a desarrollar este tipo de aplicaciones. Un ejemplo es la Universidad de

CAPÍTULO I: Fundamentación Teórica

Ciencias Informáticas (UCI); específicamente en la facultad 6, existe un proyecto llamado GeoQ, que está inmerso en el desarrollo de los SIG en plataformas de escritorio, dicho proyecto trabaja con dos aplicaciones SIG de escritorio: Quantum Gis (QGIS) y Grass.

QGIS, es una aplicación de gran potencia para la edición de mapas, es creada por la comunidad del (SWL), está programada en los lenguajes de programación C, C++, Python. Utiliza un servidor de mapas propio y posee funcionalidades para conectarse con determinadas aplicaciones como Grass y Postgres. QGIS no trabaja con el servidor de mapas MapServer y como consecuencia no permite cargar los ficheros de extensión *.map*.

Grass a pesar de permitir trabajar con MapServer, no carga ficheros de extensión *.map*, su estructura y comunicación es muy complicada, prácticamente a través de comandos, requiriéndose de un personal altamente especializado para el trabajo con esta aplicación.

En la UCI, específicamente el centro de desarrollo GEySED del departamento de Geoinformática de la Facultad 6, se ha desarrollado un producto que cumple con las exigencias de un SIG sobre plataforma web. Este sistema informático utiliza como servidor de mapas MapServer, que es sin lugar a dudas muy potente en la representación de mapas a partir de una cartografía existente. Otra de las líneas de (I+D³) que existe en el departamento es la línea de SIG para dispositivos móviles, esta línea utiliza en una de sus soluciones el motor de mapas MapServer.

Con la intención de lograr una línea de productos que tenga la mayor similitud posible en la representación de los mapas, se hace necesario buscar una alternativa para implementar una solución de escritorio que cumpla con las exigencias del departamento en este sentido.

³ Investigación y Desarrollo

CAPÍTULO I: *Fundamentación Teórica*

1.3 Análisis de soluciones existentes

Hoy día se pueden encontrar muchas soluciones que utilizan los SIG ya sea nacional e internacionalmente. A continuación se destacan como principales SIG de escritorios a nivel mundial:

ArcInfo

Es el más completo de los Sistemas de Información Geográfica de escritorio. Este incluye todas las funcionalidades de ArcEditor y ArcView y añade además el análisis espacial y la manipulación de datos. Las organizaciones usan el poder de ArcInfo cada día para crear, corregir y analizar sus datos, para tomar mejores decisiones y hacerlo con más rapidez.

Con ArcInfo se puede:

- ✚ Realizar operaciones avanzadas de análisis de datos SIG y la modelización. Disfruta de herramientas diseñadas para el análisis de superposición, análisis de proximidad, análisis de superficie y el procesamiento de la trama y la conversión.
- ✚ Publicar y convertir los datos en muchos formatos.
- ✚ Crear y administrar bases de datos geográficas.

ArcEditor

ArcEditor es un potente sistema de SIG de escritorio para la edición y gestión de datos geográficos. Se incluye toda la funcionalidad de ArcView, junto con otras herramientas de edición avanzada para garantizar la calidad de sus datos. ArcEditor soporta un único usuario y de edición multiusuario, lo que le permite desconectarse de la base de datos y editar en el campo.

Con ArcEditor se puede:

- ✚ Permitir que varios usuarios al mismo tiempo para modificar y editar datos.
- ✚ Construir y mantener las relaciones espaciales entre las características con las reglas de topología y una validación del proceso llamado.

CAPÍTULO I: *Fundamentación Teórica*

- ✚ Soporte de múltiples flujos de trabajo, gestionar el procesamiento de órdenes de trabajo y aplicar procedimientos de control de calidad para validar los cambios.
- ✚ Monitor de la base de datos a través del tiempo y evaluar escenarios hipotéticos.
- ✚ Realiza la conversión ráster a vectorial y crear datos de mapas escaneados.

ArcView

ArcView es el Sistema de Información Geográfica (SIG) para visualizar, gestionar, crear y analizar datos geográficos. El uso de ArcView, ayuda a entender el contexto geográfico de los datos, lo que le permite ver las relaciones e identificar patrones en formas nuevas.

Con ArcView se puede:

- ✚ Interactuar con los datos del mapa mediante la generación de informes y gráficos y de impresión e incrustación de sus mapas en otros documentos y aplicaciones.
- ✚ Construir modelos de procesos, secuencias de comandos y flujos de trabajo para visualizar y analizar los datos.
- ✚ Leer, importar y gestionar más de 70 diferentes tipos de datos y formatos, incluyendo datos demográficos, instalaciones, dibujos CAD, imágenes, servicios Web, multimedia y metadatos.
- ✚ Comunicar de manera más eficiente mediante la impresión, publicación y distribución de datos SIG y contenido dinámico con los demás.
- ✚ Utilizar herramientas como buscar, identificar, medir, e hipervínculo para descubrir información no disponible cuando se trabaja con mapas de papel estático.
- ✚ Tomar mejores decisiones y resolver problemas más rápidamente.

ArcReader

ArcReader es libre, es una aplicación de mapas de escritorio fácil de usar que permite a los usuarios visualizar, explorar e imprimir mapas y globos terráqueos. Cualquier persona con ArcReader puede consultar mapas interactivos de alta calidad escrito por un alto nivel de productos ArcGIS Desktop y publicado con la extensión de ArcGIS Editor.

CAPÍTULO I: *Fundamentación Teórica*

Con ArcReader se puede:

- ✚ Ver, navegar e imprimir los mapas publicados con ArcGIS ArcReader (archivos. mf).
- ✚ Implementación de los datos de SIG para principiantes y profesionales.
- ✚ Zoom, pan y cambiar entre mapas y puntos de vista de diseño de página.
- ✚ Comunicar de manera más eficiente con la posibilidad de marcar gráfica de mapas.
- ✚ Correspondencia documentos publicados y mundo, como todos los símbolos de la capa y elementos cartográficos del mapa en cualquier impresora compatible.
- ✚ Crear aplicaciones personalizadas de ArcReader e integrar las capacidades de ArcReader en las aplicaciones existentes (13).

Tanto ArcInfo, ArcReader, ArcEditor y ArcView pertenecen a la familia del software privativo, las cuales no cumplen con los requerimientos establecidos por el departamento.

Grass

GRASS es un SIG con capacidad ráster y vectorial. Además cuenta con sistemas integrados de visualización de datos y de procesamiento de imagen. Este incluye más de 350 módulos para gestión, procesamiento, análisis y visualización de datos georreferenciados.

GRASS es GNU GPL desde 1999. Esta licencia protege a los contribuidores de GRASS frente al uso de sus contribuciones en proyectos propietarios que no permiten el libre acceso a su código fuente. La GPL asegura que todo el código publicado basado en código GPL, solo pueda ser publicado a su vez bajo licencia GPL⁴.

GRASS es un SIG modular con datos organizados como coberturas ráster, vectoriales y de puntos. GRASS proporciona una gran variedad de herramientas que permiten clasificarlo como un SIG de altas prestaciones. A continuación se resumen las principales funcionalidades de GRASS:

- ✚ Integración de datos espaciales.

⁴ GPL: Licencia Pública General

CAPÍTULO I: *Fundamentación Teórica*

- ✚ Procesado de datos ráster.
- ✚ Procesado de datos vectoriales.
- ✚ Procesado de datos puntuales.
- ✚ Procesado de imágenes.
- ✚ Visualización.
- ✚ Modelado y simulaciones.
- ✚ Soporte para datos temporales.
- ✚ Procesado de datos 3D.
- ✚ Enlaces con otras herramientas (14).

QuantumGis

Es una aplicación SIG de escritorio, con grandes potencialidades para la edición de mapas, de código abierto. El proyecto nació en mayo del 2002 y se estableció como un proyecto dentro del Sourceforge en junio del mismo año. Actualmente QGis⁵ corre en la mayoría de las plataformas como Unix, Windows y OSx. Sus interfaces gráficas están desarrolladas en Qt y la parte funcional del sistema en los lenguajes C, C++ y Python. Esto hace que QGIS sea muy rápido y tenga una interfaz muy amigable para los usuarios.

QGIS contiene características comunes a todos los SIG divididas en elementos del núcleo y complementos.

Elementos de núcleo:

- ✚ Soporte ráster y vectorial.
- ✚ Soporte para PostgreSQL con tablas espaciales utilizado PostGIS.
- ✚ Integración con Grass incluida visualización edición y análisis.
- ✚ Diseño de mapas.
- ✚ Soporte OGC⁶.
- ✚ Panel de vista general.
- ✚ Marcadores espaciales.

⁵ QGis: Quantum GIS, es un Sistema de Información Geográfica de ambiente de escritorio.

⁶ OGC: Open Geospatial Consortium, su fin es la definición de estándares abiertos e interoperables dentro de los Sistemas de Información Geográfica

CAPÍTULO I: *Fundamentación Teórica*

- ✚ Identificar y seleccionar elementos.
- ✚ Editar, visualizar y buscar atributos.
- ✚ Etiquetado de elementos.
- ✚ Proyecciones al vuelo.
- ✚ Guardar y recuperar proyectos.
- ✚ Exportar ficheros con extensión .map a MapServer.
- ✚ Cambiar simbología vectorial y ráster.
- ✚ Arquitectura extensible con complementos.

Complementos:

- ✚ Añadir capas WFS⁷.
- ✚ Añadir capas de texto delimitado.
- ✚ Decoración.
- ✚ Herramienta GPS8Grass.
- ✚ Generador de mallas.
- ✚ Funciones de geprocesamiento PostgreSQL.
- ✚ Consola Python (15).

Gracias a la solidaridad y ayuda de amigos en otros países y a la creatividad y desarrollo de los especialistas cubanos, estos siempre han estado muy actualizados en cuanto a la aplicación de los Sistemas de Información Geográfica desde que surgió esta tecnología, existen varias esferas que utilizan aplicaciones SIG web, ejemplos de esto es el Sistema de Información Geográfica en Salud (SIGSA) y desarrollo de los mismos para las áreas de epidemiología como SIG-EPI.

Existen otros SIG como el Sistema de Información Geográfica para la Gestión integrada de la Cuenca Almendrares (2004), de los autores Luís Peña, Rafael Ojeda, Floralb Rosado, Urbano Ferias, Luís Hernández, Justo Orihuela, el cual fue creado para la gestión integral de la Cuenca hidrogeológica Almendrares en función del desarrollo local

⁷ WFS: Web Feature Service o Servicio de entidades vectoriales que proporciona la información relativa a la entidad almacenada en una capa vectorial (cobertura) que reúnen las características formuladas en la consulta.

⁸ GPS: Son las siglas de Sistema de Posicionamiento Global (*Global Positioning System*). Este sistema permite la localización de algún objeto (persona, nave, vehículo) en cualquier punto del planeta

CAPÍTULO I: *Fundamentación Teórica*

utilizando las herramientas SIG, en dicho trabajo los autores exponen como el vínculo del conocimiento de los datos geográficos y su relación con los factores sociales de la comunidad ayudan para la toma de decisiones relacionada con la protección medio ambiental de la Cuenca. También ha utilizado SIGs de escritorio como ArcView. La primera versión obtenida en Cuba fue la 2.1 para Windows soportada en 14 disquetes de 3½". Posteriormente surgieron otras versiones en la década de los noventa.

1.4 Conclusiones Parciales

En este capítulo se han abordado diferentes conceptos relacionados con el dominio del problema que ayudan a la comprensión de la investigación. Conjuntamente se realizó una descripción del objeto de estudio, tratando los aspectos más importantes. Se puede afirmar además que de las soluciones existentes estudiadas, arcInfo, arcReader y arcView no son factibles para países bloqueados como Cuba por pertenecer a la familia del software privativo. En el caso de Grass y QuantumGis a pesar de que muchas funcionalidades pueden servir de ayuda a la hora de implementar el sistema, no cumple con los necesidades del Departamento debido a que dificulta el trabajo con el servidor de mapas MapServer, quedando demostrado de esta manera la necesidad de crear una herramienta de representación de información geográfica sobre tecnologías libres que permita la comunicación con el servidor de mapas MapServer.

CAPÍTULO II: Tecnologías y Herramientas a utilizar.

Capítulo 2: Tecnologías y herramientas a utilizar.

2.1 Introducción.

En este capítulo se explica mediante una descripción detallada el uso de las herramientas y tecnologías a utilizar, para desarrollar la herramienta de representación de información geográfica sobre tecnologías libres. Además se definirá la metodología para guiar el proceso de desarrollo del sistema, el lenguaje de programación, el gestor de base de datos a utilizar, las principales librerías así como el entorno de desarrollo donde se realizará la aplicación.

2.2 Lenguaje de Programación.

Se escogió el lenguaje de programación Python para el desarrollo de la aplicación, debido a que proporciona capacidades de Programación Orientada a Objetos (P.O.O) con el fin de incrementar la productividad, calidad y reutilización del software. Se trata de un lenguaje interpretado o de script, con tipado dinámico, fuertemente tipado, multiplataforma, con una sintaxis muy limpia y que favorece un código legible. Además de lo expuesto anteriormente se eligió este lenguaje por su estrecha comunicación con el servidor de mapas MapServer a través de la librería Python-MapScript, que este último brinda.

El intérprete de Python está disponible en multitud de plataformas (UNIX, Solaris, Linux, DOS, Windows, OS/2, Mac OS, etc.) por lo que sino se trabaja con librerías específicas los programas desarrollados en este, podran correr en todos estos sistemas sin grandes cambios. También permite la programación imperativa, programación funcional y programación orientada a aspectos. Python tiene, no obstante, muchas de las características de los lenguajes compilados, por lo que se podría decir que es semi interpretado. En Python, como en Java y muchos otros lenguajes, el código fuente se traduce a un pseudo código máquina intermedio llamado bytecode la primera vez que se ejecuta, generando archivos .pyc⁹ o .pyo¹⁰(bytecode optimizado), que son los que se ejecutarán en sucesivas ocasiones (17).

⁹.pyc: Extensión de algunos archivos de Python después que es ejecutado.

CAPÍTULO II: Tecnologías y Herramientas a utilizar.

2.3 Biblioteca PyQt.

Se escogió para el desarrollo de interfaces gráficas de usuarios la librería multiplataforma Qt. Esta librería a pesar de ser creada inicialmente para el lenguaje C++, por su fortaleza en apariencia y portabilidad se fueron extendiendo hacia otros lenguajes de programación, ejemplo en Python esta librería es conocida como PyQt, siendo ella el objetivo principal de la selección realizada.

PyQt una librería totalmente orientada a objetos, es por ello que las API (*Application Programming Interface*) cuenta con diferentes métodos, soporta el uso de diferentes motores de Bases de Datos y de archivos XML, además de otras estructuras de datos tradicionales. Igualmente gana en portabilidad con respecto a la librería wxPython y PyGTK, logrando también excelente apariencia en los sistemas operativos Linux como en Windows.

Estas librerías de desarrollo se encuentran disponibles para las siguientes plataformas:

- 🚦 X11: Para X Windows System (GNU/Linux, UNIX, BSD).
- 🚦 Mac: Para los sistemas Mac OS X.
- 🚦 Windows: Para las diferentes versiones del Sistema Operativo
- 🚦 Windows.
- 🚦 PDA - Para los dispositivos empotrados (21).

2.4 Sistema Gestor de Bases de Datos

Se utilizó como Sistema Gestor de Base de Datos PostgreSQL con extensión PostGIS, por la gran cantidad de ventajas que tiene, pero principalmente porque existe un módulo integrado en este gestor (PostGIS), que permite almacenar y manejar objetos geográficos, objetivo clave para el desarrollo de la herramienta.

PostgreSQL es el gestor de bases de datos de código abierto más avanzado hoy en día. Este ofrece el control de concurrencia multi-versión, soportando casi toda la sintaxis SQL (Incluyendo subconsultas, transacciones y tipos y funciones definidas por el usuario),

¹⁰.pyo: Extensión de algunos archivos de Python después que es ejecutado.

CAPÍTULO II: Tecnologías y Herramientas a utilizar.

contando también con un amplio conjunto de enlaces con lenguajes de programación (incluyendo C, C++, Java, perl, tcl y python).

Además de lo antes planteado se seleccionó porque con el gestor de bases de datos PostgreSQL, se pueden incorporar los siguientes cuatro conceptos básicos de tal forma que los usuarios pueden extender fácilmente el sistema, clases, herencia tipos y funciones entre otras características que aportan flexibilidad, los conceptos son:

- ✚ Restricciones, (Constraints).
- ✚ Disparadores (triggers).
- ✚ Reglas (rules).
- ✚ Integridad transaccional.

Estas características colocan a Postgres en la categoría de las Bases de Datos identificadas como *objeto-relacionales* (18).

El siguiente gráfico (Fig.1) ilustra de manera general los componentes más importantes en un sistema PostgreSQL.

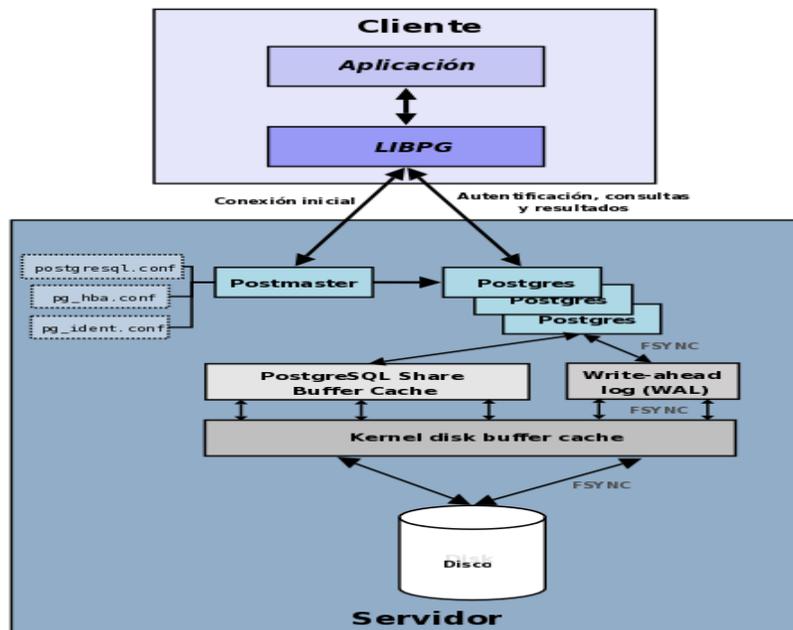


Figura 2: Componentes en un sistema PostgreSQL(18).

CAPÍTULO II: Tecnologías y Herramientas a utilizar.

Aplicación cliente: Esta es la aplicación cliente que utiliza PostgreSQL como administrador de bases de datos. La conexión puede ocurrir vía TCP/IP ó sockets locales.

Demonio postmaster: Este es el proceso principal de PostgreSQL. Es el encargado de escuchar por un puerto/socket por conexiones entrantes de clientes. También es el encargado de crear los procesos hijos que se encargaran de autenticar estas peticiones, gestionar las consultas y mandar los resultados a las aplicaciones clientes.

Ficheros de configuración: Los 3 ficheros principales de configuración utilizados por PostgreSQL son: postgresql.conf, pg_hba.conf y pg_ident.conf.

Procesos hijos postgres: Procesos hijos que se encargan de autenticar a los clientes, de gestionar las consultas y mandar los resultados a las aplicaciones clientes.

PostgreSQL share buffer cache: Memoria compartida usada por PostgreSQL para almacenar datos en caché.

Write-Ahead Log (WAL): Componente del sistema encargado de asegurar la integridad de los datos (recuperación de tipo REDO).

Kernel disk buffer cache: Caché de disco del sistema operativo.

Disco: Disco físico donde se almacenan los datos y toda la información necesaria para que PostgreSQL funcione (19).

Después de haber visto las características de PostgreSQL, se puede deducir que posee numerosas ventajas pero para trabajar con SIG es necesario instalarlo con extensión **PostGIS** permitiendo el uso de objetos SIG (Geographic Information Systems). Además PostGIS incluye soporte para índices GiST basados en R-Tree y funciones básicas para el análisis de objetos SIG.

CAPÍTULO II: Tecnologías y Herramientas a utilizar.

Está creado por Refrations Research Inc, como un proyecto de investigación de tecnologías de bases de datos espaciales y está publicado bajo licencia GNU¹¹.

Al utilizar PostGIS se pueden usar todos los objetos que aparecen en la especificación OpenGIS como puntos, líneas, polígonos, multilíneas, multipuntos y colecciones geométricas.

2.5 Servidor de Mapas.

Se eligió como servidor de mapas MapServer por la fortaleza que posee para la graficación de imágenes y además proporciona una librería MapScript para la comunicación de él con otros lenguajes de programación entre ellos Python.

MapServer es un entorno de desarrollo en código abierto (*Open Source Initiative*). Permite la creación de aplicaciones SIG en Internet/Intranet con el fin de visualizar, consultar y analizar información geográfica a través de la red mediante la tecnología *Internet Map Server* (IMS).

Es multiplataforma por lo cual se ejecuta en Linux y Windows, soporta formatos vectoriales como: ESRI¹², shapefiles, PostGIS, ESRI ArcSDE, GML y otros muchos vía OGR. Los formatos ráster soportados son: JPG, PNG, GIF, TIFF/, EPPL7 y otros vía GDAL(22).

2.6 Librería Python-MapScript.

Se utilizará la librería Python-MapScript porque permite la comunicación entre el lenguaje de programación Python y las funciones del servidor de mapas MapServer. Permite utilizar funciones como MapObj para crear objetos de tipo mapa dada una cartografía con extensión .map, así como trabajar con las diferentes funciones que este brinda, como el extent, fullextent, rectobj, entre otras y generar imágenes para mostrarlas a través de un visor.

¹¹ GNU: Licencia pública general(siglas en inglés)

¹² ESRI: Es un tipo de formato que soporta las aplicaciones SIG.

CAPÍTULO II: Tecnologías y Herramientas a utilizar.

MapScript expone las funcionalidades de MapServer a varios lenguajes de scripting; esto reduce el tiempo de programación para los desarrolladores que quieren añadir capacidades de mapeo a una aplicación. En lugar de crear un método personalizado para el mapeo, la API MapScript proporciona algunas herramientas de gran alcance que son robustas y listas para usar.

También proporciona una forma cómoda de interactuar con el mapeo de datos sin dejar de utilizar el lenguaje de programación favorito. Permite cargar, manipular y crear archivos de mapas. Por ejemplo, puede cambiar la configuración de la capa, manipular clases mapa de archivos, obtener imágenes de la producción, la exportación de datos espaciales y mucho más. Debido a que los usos comunes entornos de secuencias de comandos, funciones MapScript se puede combinar con otras funciones de ese idioma (22).

2.7 Entorno de desarrollo Integrado(IDE).

Se ha seleccionado como entorno de desarrollo Eclipse, a causa de que ofrece numerosas ventajas, pero antes de verlas es necesario conocer en qué consiste. Eclipse es un entorno de desarrollo integrado (IDE, Integrated Development Environment) que facilita enormemente las tareas de edición, compilación y ejecución de programas durante su fase de desarrollo. Además es una aplicación gratuita y de código abierto, disponible en la red para su descarga e incluida ya en muchas distribuciones de Linux.

En sí mismo Eclipse es un marco y un conjunto de servicios para construir un entorno de desarrollo a partir de componentes conectados (plugin). Dentro de estos plugin se encuentra eclipse PyDev, el cual permite hacer configuraciones al eclipse para programar en lenguaje Python, siendo este es uno de los principales motivos por el que se utiliza Eclipse, conjuntamente con las siguientes ventajas que ofrece este plugin:

- ✚ Permite Acceso a todos los ficheros del proyecto.
- ✚ Ofrece una ventana de depuración y errores.
- ✚ Información de los parámetros que usa una función.
- ✚ Subrayado de errores automáticos.
- ✚ Autocompletado.

CAPÍTULO II: Tecnologías y Herramientas a utilizar.

- ✚ Ventana de consola donde se muestran todos los “prints” que hace nuestro script.
- ✚ Ventana con un completo listado de todas las variables, funciones, objetos, métodos, del fichero que se está editando.
- ✚ Acceso rápido a varios proyectos a la vez.

Además de las ventajas anteriormente expuestas incluye una serie de características únicas como la refactorización de código, el código de actualizaciones automáticas e instalaciones (a través del Update Manager). A pesar del gran número de características estándar, Eclipse se diferencia de IDE tradicional en varios aspectos fundamentales. A consideración de varios autores una de las características más interesantes de Eclipse es que es completamente independiente de la plataforma y del idioma. Además de la mezcla ecléctica de lenguajes soportados por el Consorcio Eclipse (Java, C / C + +, COBOL), también hay proyectos en marcha para agregar soporte para idiomas tan diversos como Python, Eiffel, PHP, Ruby y C # para Eclipse (23).

2.8 Metodología ágil.

Para el desarrollo del software se necesitó de una metodología que fuera capaz de cumplir con una serie de requisitos que proporcionaran el progreso de la aplicación, para esto se eligió **Programación Extrema (XP)**.

Esta es una metodología para el desarrollo de software y consiste básicamente en ajustarse estrictamente a una serie de reglas que se centran en las necesidades del cliente para lograr un producto de buena calidad en poco tiempo.

La Programación Extrema es una metodología ágil centrada en potenciar las relaciones interpersonales como clave para el éxito en el desarrollo de software. Este tipo de método se basa en una realimentación continuada entre el cliente y el equipo de desarrollo con una comunicación fluida entre todos los participantes, también busca la simplicidad en las soluciones implementadas y el coraje para enfrentar los cambios. Este tipo de programación es la adecuada para los proyectos con requisitos imprecisos, muy cambiantes y con un riesgo técnico excesivo.

CAPÍTULO II: Tecnologías y Herramientas a utilizar.

Básicamente, la metodología XP se basa en cinco valores:

- ✚ **Comunicación:** La comunicación permanente es fundamental en XP. Dado que la documentación es escasa, el diálogo frontal, cara a cara, entre desarrolladores, gerentes y el cliente es el medio básico de comunicación. Una buena comunicación tiene que estar presente durante todo el proyecto.
- ✚ **Simplicidad:** La sencillez es esencial para que todos puedan entender el código y se trata de mejorar mediante recodificaciones continuas.
- ✚ **Feedback:** Básicamente el continuo contacto con el usuario, al irle entregando las sucesivas versiones, en funcionamiento del producto, permite que este de su valoración y comunique, cada vez mejor, lo que realmente quiere en el producto.
- ✚ **Coraje:** Básicamente es trabajar muy duro durante las horas dedicadas a ello.
- ✚ **Respeto:** Si los miembros de un equipo no se preocupan por sí mismos y por su trabajo, la metodología no puede funcionar. Es necesario ser respetuoso con sus colegas, sus contribuciones, su organización y con las personas cuya vida se toca por el sistema que está escribiendo (25).

El ciclo de vida ideal consta de 6 fases:

1. **Exploración:** En esta fase, los clientes plantean a grandes rasgos las historias de usuario que son de interés para la primera entrega del producto. Al mismo tiempo el equipo de desarrollo se familiariza con las herramientas, tecnologías y prácticas que se utilizarán en el proyecto. Se prueba la tecnología y se exploran las posibilidades de la arquitectura del sistema construyendo un prototipo.
2. **Planificación de Entregas:** En esta fase el cliente establece la prioridad de cada historia de usuario y correspondientemente, los programadores realizan una estimación del esfuerzo necesario de cada una de ellas. Se toman acuerdos sobre el contenido de la primera entrega y se determina un cronograma en conjunto con el cliente.

CAPÍTULO II: Tecnologías y Herramientas a utilizar.

- 3. Iteraciones:** Esta fase incluye varias iteraciones sobre el sistema antes de ser entregado. El Plan de Entrega está compuesto por iteraciones de no más de tres semanas.
- 4. Producción:** La fase de producción requiere de pruebas adicionales y revisiones de rendimiento antes de que el sistema sea trasladado al entorno del cliente.
- 5. Mantenimiento:** Mientras la primera versión se encuentra en producción, el proyecto XP debe mantener el sistema en funcionamiento al mismo tiempo que desarrolla nuevas iteraciones. Para realizar esto se requiere de tareas de soporte para el cliente.
- 6. Muerte:** Es cuando el cliente no tiene más historias para ser incluidas en el sistema. Esto requiere que se satisfagan las necesidades del cliente en otros aspectos como rendimiento y confiabilidad del sistema. Se genera la documentación final del sistema y no se realizan más cambios en la arquitectura (25).

Se optó por la metodología XP debido a que el software que se realizó es pequeño, así como su equipo de trabajo (una persona) y que existe una sistemática comunicación entre el equipo de trabajo y el cliente. Se tuvieron en cuenta sus características fundamentales:

- ✚ Desarrollo iterativo e incremental: pequeñas mejoras, unas tras otras.
- ✚ Pruebas unitarias continuas, frecuentemente repetidas y automatizadas, incluyendo pruebas de regresión. Se aconseja escribir el código de la prueba antes de la codificación.
- ✚ Frecuente interacción del equipo de programación con el cliente o usuario. Se recomienda que un representante del cliente trabaje junto al equipo de desarrollo.
- ✚ Corrección de todos los errores antes de añadir nueva funcionalidad. Hacer entregas frecuentes.

CAPÍTULO II: Tecnologías y Herramientas a utilizar.

- ✚ Refactorización del código, es decir, reescribir ciertas partes del código para aumentar su legibilidad y mantenibilidad pero sin modificar su comportamiento. Las pruebas han de garantizar que en la refactorización no se ha introducido ningún fallo.
- ✚ Propiedad del código compartida: en vez de dividir la responsabilidad en el desarrollo de cada módulo en grupos de trabajo distintos, este método promueve el que todo el personal pueda.
- ✚ Corregir y extender cualquier parte del proyecto. Las frecuentes pruebas de regresión garantizan que los posibles errores serán detectados.
- ✚ Simplicidad en el código: Es la mejor manera de que las cosas funcionen. Cuando todo funcione se podrá añadir funcionalidad si es necesario. La programación extrema apuesta que es más sencillo hacer algo simple y tener un poco de trabajo extra para cambiarlo si se requiere, que realizar algo complicado y quizás nunca utilizarlo (24).

Fundamentalmente se determinó seleccionar la metodología **XP**, porque responde a la necesidades de entregar la herramienta de representación y análisis de información geográfica sobre tecnologías libres en un corto período de tiempo y porque no se cuenta con un equipo de desarrollo para generar todos los artefactos que pueda generar otra metodología, siendo XP la que se ajusta a que se logre con eficiencia el desarrollo del sistema.

Además utilizarla tiene como ventaja que el cliente se encuentra dentro del equipo de desarrollo, es decir trabajan en conjunto permitiendo la retroalimentación o una mejor comunicación entre el cliente y el programador.

También se consigue tener un proceso de desarrollo motivado, debido a que XP no permite excesos de trabajos y se consigue una mayor integración entre los miembros por la comunicación.

CAPÍTULO II: Tecnologías y Herramientas a utilizar.

Otros de los motivos de su uso es que una de las prácticas más importante de XP y que está muy ligada al proceso de formación actual de la universidad es la de Paso Sostenible. Aquí se plantea que en la semana se debe trabajar solo 40 horas, lo que permite que el proyecto se adapte a las horas de producción que se necesitan tener por cada estudiante de la UCI.

2.9 Arquitectura de Software.

La herramienta de representación de información geográfica sobre tecnologías libres está basada en el patrón de arquitectura Modelo Vista Controlador (MVC). La arquitectura MVC consiste en separar los datos de una aplicación, la interfaz de usuario y la lógica de control en tres componentes distintos.

- ✚ **Modelo:** Encapsula los datos y las funcionalidades. El modelo es independiente de cualquier representación de salida y/o comportamiento de entrada.
- ✚ **Vista:** Muestra la información al usuario. Obtiene los datos del modelo. Pueden existir múltiples vistas del modelo. Cada vista tiene asociado un componente controlador.
- ✚ **Controlador:** Reciben las entradas, usualmente como eventos que codifican los movimientos o pulsación de botones del ratón, pulsaciones de teclas, etc. Los eventos son traducidos a solicitudes de servicio ("**service requests**" en el texto original) para el modelo o la vista. El usuario interactúa con el sistema a través de los controladores (27).

Ventajas del MVC (26):

- ✚ Clara separación entre interfaz, lógica de negocio y presentación.
- ✚ Sencillez para crear distintas representaciones de los mismos datos.
- ✚ Reutilización de los componentes.

CAPÍTULO II: Tecnologías y Herramientas a utilizar.

2.10 Conclusiones Parciales.

En este capítulo se efectuó un estudio de las herramientas, el lenguaje y la metodología que se utilizaron para el desarrollo del software que permita la representación y análisis de información geográfica sobre tecnologías libres. Gracias a este estudio se permitió conocer más acerca de las debilidades, fortalezas de las herramientas y tecnologías adecuadas para la construcción de la aplicación, seleccionando así a Python como lenguaje de programación, el cual permite el desarrollo de las funcionalidades básicas de la herramienta, MapServer v5.2 como servidor de mapas proporcionando gran calidad en la graficación de los mapas que se utilizarán, Eclipse como entorno de desarrollo integrado, sirviendo de ayuda a la hora de integrar con el plugins Eclipse_PyDev para vinculación con el lenguaje de programación Python, PostgreSQL con extensión PostGIS como sistema gestor de base de datos permitiendo el manejo con objetos espaciales, y como metodología utilizada para guiar el proceso de desarrollo XP, colaborando a la necesidades de entregar la herramienta en un corto período de tiempo.

CAPÍTULO III: Presentación de la Solución Propuesta.

Capítulo 3: Presentación de la solución propuesta.

3.1 Introducción

En el presente capítulo se describen las Historias de Usuarios, lo que permite lograr de forma general, una mejor comprensión acerca del sistema, ya que son escritas por los propios clientes tal y como ven ellos las necesidades del sistema.

3.2 Fase de Exploración

La primera fase de la metodología de desarrollo XP es exploración, en esta fase los clientes plantean a grandes rasgos las historias de usuarios que son de interés para la primera entrega del producto. Así mismo el equipo de trabajo se familiariza con la tecnología, las herramientas y prácticas a utilizar en el proyecto. La fase de exploración toma de pocas semanas a pocos meses, dependiendo del tamaño y familiaridad que tengan los programadores con la tecnología. Finalmente se prueba la tecnología y se exploran las posibilidades de la arquitectura del sistema construyendo un prototipo. Es la fase en la que se define el alcance general del proyecto (24).

3.2.1 Historias de Usuarios

Las Historias de Usuario (HU) son utilizadas en la metodología de desarrollo ágil XP para representar una breve descripción del comportamiento del sistema. Se trata de tarjetas de papel en las cuales el cliente describe brevemente las características que el sistema debe poseer, sean requisitos funcionales o no funcionales. Emplea terminología del cliente sin lenguaje técnico, se realiza una por cada funcionalidad del sistema, se emplean para hacer estimaciones de tiempo y para el plan de lanzamientos, reemplazan un gran documento de requisitos y presiden la creación de las pruebas de aceptación.

Cuando llega la hora de implementar una historia de usuario, el cliente y los desarrolladores se reúnen para concretar y detallar lo que tiene que hacer dicha historia. El tiempo de desarrollo ideal para una historia de usuario es entre 1 y 3 semanas, estas permiten responder rápidamente a los requerimientos cambiantes (24).

CAPÍTULO III: Presentación de la Solución Propuesta.

Las Historias de Usuarios tienen tres aspectos:

- ✚ **Tarjeta:** En ella se almacena suficiente información para identificar y detallar la historia.
- ✚ **Conversación:** Cliente y programadores discuten la historia para ampliar los detalles (verbalmente cuando sea posible, pero documentada cuando se requiera confirmación).
- ✚ **Pruebas de Aceptación:** Permite confirmar que la historia ha sido implementada correctamente.

A continuación se realiza una explicación de la plantilla de HU que se utiliza para el desarrollo del sistema:

Historias de Usuarios	
Número: Se ponen los números sucesivos a partir de 1	Nombre: Se identifica la HU.
Usuario: Aquí se colocan los usuarios que interactúan con la HU	
Prioridad en Negocio: Define el impacto de la HU para el negocio, de acuerdo con las necesidades del usuario. Y se define como (Alta / Media / Baja)	Riesgo en Desarrollo: Define la dificultad técnica al desarrollar la HU, desde el punto de vista del programador. Y se define como (Alto / Medio / Bajo)
Puntos de Estimación: Permiten estimar duración de implementación.	Iteración Asignada: Precisa la iteración a la que pertenece la historia de usuario.
Descripción: Explica en qué consiste las HU.	
Observaciones: Información que se estime agregar para hacer más entendible la HU.	

Tabla #1 Descripción de la HU

CAPÍTULO III: Presentación de la Solución Propuesta.

Historias de Usuarios	
Número: 1	Nombre: Cargar Capas.
Usuario: Cliente	
Prioridad en Negocio: Alta	Riesgo en Desarrollo: Alto
Puntos de Estimación: 2	Iteración Asignada: 1
Descripción: Permite al usuario visualizar todas las capas del mapa.	
Observaciones: La aplicación debe estar conectada al MapServer a través de la librería MapScrip.	

Tabla #2 Cargar Capas.

Historias de Usuarios	
Número: 2	Nombre: Visualizar Capas.
Usuario: Cliente	
Prioridad en Negocio: Alta	Riesgo en Desarrollo: Alto
Puntos de Estimación: 1	Iteración Asignada: 1
Descripción: Permite al usuario activar o desactivar las capas que el estime conveniente de acuerdo a sus necesidades.	
Observaciones: Debe haberse cargado al menos una capa para permitir el manejo sobre ella.	

Tabla #3 Visualizar Capas.

CAPÍTULO III: Presentación de la Solución Propuesta.

Historias de Usuarios	
Número: 3	Nombre: Mostrar Capas.
Usuario: Cliente	
Prioridad en Negocio: Alta	Riesgo en Desarrollo: Medio
Puntos de Estimación: 3	Iteración Asignada: 2
Descripción: Mostrar Capas le permite al usuario escoger la capa que desea mostrar, logrando una mejor visión del mapa.	
Observaciones: La aplicación debe tener varias capas cargadas, para poder seleccionar la que se desee.	

Tabla #4 Mostrar Mapa.

Historias de Usuarios	
Número: 4	Nombre: Acercar Mapa.
Usuario: Cliente	
Prioridad en Negocio: Media	Riesgo en Desarrollo: Medio
Puntos de Estimación: 1	Iteración Asignada: 3
Descripción: Acercar Mapa le permite al usuario acercar la imagen, logrando una mejor visión del elemento seleccionado.	
Observaciones: La aplicación debe tener una capa cargada para poder realizar el desplazamiento en ella.	

Tabla #5 Acercar Mapa.

CAPÍTULO III: Presentación de la Solución Propuesta.

Historias de Usuarios	
Número: 5	Nombre: Alejar Mapa.
Usuario: Cliente	
Prioridad en Negocio: Media	Riesgo en Desarrollo: Medio
Puntos de Estimación: 1	Iteración Asignada: 3
Descripción: Alejar mapa le permite al usuario alejar la imagen, logrando una mejor visión del elemento seleccionado.	
Observaciones: La aplicación debe tener una capa cargada para poder realizar el desplazamiento en ella.	

Tabla #6 Alejar Mapa.

Historias de Usuarios	
Número: 6	Nombre: Escala Normal.
Usuario: Cliente	
Prioridad en Negocio: Media	Riesgo en Desarrollo: Medio
Puntos de Estimación: 1	Iteración Asignada: 3
Descripción: Escala Normal le permite al usuario volver a ver el mapa en su proporción original.	
Observaciones: La aplicación debe tener una capa cargada para poder trabajar sobre ella.	

Tabla #7 Escala Normal.

CAPÍTULO III: Presentación de la Solución Propuesta.

Historias de Usuarios	
Número: 7	Nombre: Mover Mapa.
Usuario: Cliente	
Prioridad en Negocio: Alta	Riesgo en Desarrollo: Alto
Puntos de Estimación: 1	Iteración Asignada: 4
Descripción: Mover mapa le permite al usuario mover todo el lienzo o área de trabajo según sea necesario y cómodo para él.	
Observaciones: La aplicación debe tener una capa cargada para poder realizar el desplazamiento en ella.	

Tabla #8 Mover Mapa.

Historias de Usuarios	
Número: 8	Nombre: Exportar a PDF.
Usuario: Cliente	
Prioridad en Negocio: Media	Riesgo en Desarrollo: Medio
Puntos de Estimación: 1	Iteración Asignada: 4
Descripción: Permite al usuario exportar una imagen del mapa.	
Observaciones: Debe estar agregada la librería ReporLab que permite al usuario exportar imágenes del mapa en archivos pdf, dando la opción de guardar la imagen.	

Tabla #9 Exportar a PDF.

CAPÍTULO III: Presentación de la Solución Propuesta.

Historias de Usuarios	
Número: 9	Nombre: Imprimir Imagen.
Usuario: Cliente	
Prioridad en Negocio: Media	Riesgo en Desarrollo: Bajo
Puntos de Estimación: 1	Iteración Asignada: 4
Descripción: Permite al usuario imprimir una imagen del mapa.	
Observaciones: La aplicación debe tener una capa cargada para poder imprimir la imagen.	

Tabla #10 Imprimir Imagen.

3.3 Fase de planificación

En esta fase el cliente establece la prioridad de cada historia de usuario y correspondientemente los programadores realizan una estimación del esfuerzo necesario de cada una de ellas. Se toman acuerdos sobre el contenido de la primera entrega y se determina un cronograma en conjunto con el cliente. Una entrega debería obtenerse en no más de tres meses. Esta fase dura unos pocos días.

Las estimaciones de esfuerzo asociado a la implementación de las historias la establecen los programadores utilizando como medida el punto. Un punto, equivale a una semana ideal de programación. Las historias generalmente valen de 1 a 3 puntos.

Por otra parte, el equipo de desarrollo mantiene un registro de la velocidad de desarrollo, establecida en puntos por iteración, basándose principalmente en la suma de puntos correspondientes a las historias de usuario que fueron terminadas en la última iteración. La planificación se puede realizar basándose en el tiempo o el alcance.

CAPÍTULO III: *Presentación de la Solución Propuesta.*

La velocidad del proyecto es utilizada para establecer cuántas historias se pueden implementar antes de una fecha determinada o cuánto tiempo tomará implementar un conjunto de historias.

Al planificar por tiempo, se multiplica el número de iteraciones por la velocidad del proyecto, determinándose cuántos puntos se pueden completar. Al planificar según alcance del sistema, se divide la suma de puntos de las historias de usuario seleccionadas entre la velocidad del proyecto, obteniendo el número de iteraciones necesarias para su implementación. (24).

3.3.1 Estimación de esfuerzos por historia de usuario

Para el buen desarrollo del sistema propuesto, se efectúa una estimación para cada una de las historias de usuario identificadas, llegando a los resultados que se muestran a continuación.

Historias de Usuarios	Puntos de estimación
Cargar Capas	2 semana
Visualizar Capas	1 semana
Mostrar Capas	3 semana
Acercar Mapa	1 semana
Alejar Mapa	1 semana
Escala Normal	1 semana
Mover Mapa	1 semana
Exportar a pdf	1 semana
Imprimir Imagen	1 semana

CAPÍTULO III: *Presentación de la Solución Propuesta.*

Tabla #11 Estimación de esfuerzo por historia de usuario

3.3.2 Plan de iteraciones

Esta fase incluye varias iteraciones sobre el sistema antes de ser entregado. El Plan de entrega está compuesto por iteraciones de no más de tres semanas. En la primera iteración se puede intentar establecer una arquitectura del sistema que pueda ser utilizada durante el resto del proyecto. Esto se logra escogiendo las historias que fueren la creación de esta arquitectura, sin embargo, esto no siempre es posible ya que es el cliente quien decide que historias de usuarios se deben implementar en cada iteración (para maximizar el valor de negocio). Al final de la última iteración el sistema estará listo para entrar en producción. (24).

Después de haber definido las HU y estimado el esfuerzo propuesto para la realización de cada una de ellas, se tomó la decisión de realizar el sistema en 4 iteraciones, las cuales se especifican a continuación:

Iteración 1:

El objetivo fundamental de esta iteración es dar cumplimiento a las HU que tienen mayor prioridad para el desarrollo del software. Una vez concluida esta iteración se cuenta con la funcionalidad descrita en la HU # 1 y 2, las cuales hace alusión a todo lo relativo con cargar y visualizar capas.

Iteración 2:

La presente iteración tiene como objetivo desarrollar la HU#3. Una vez concluida esta iteración se cuenta con la funcionalidad descrita en la HU # 3, la cual hace mención a todo lo referente con mostrar las capas que el usuario desee.

Iteración 3:

En esta iteración se cumple con las HU que corresponden a los números #4, #5 y #6. Estas son referentes al zoom del mapa, le permiten al cliente acercar y alejar la parte

CAPÍTULO III: Presentación de la Solución Propuesta.

seleccionada del mapa, así como lograr su escala normal, ofreciendo al usuario una mejor visualización.

Iteración 4:

Esta iteración tiene como propósito desarrollar la HU #7, HU #8 y #9. Esta historia es la que se encarga de desplazar el mapa, esto le permite al usuario mover todo el lienzo o área de trabajo según le sea necesario, además de exportar a formato pdf e imprimir la imagen respectivamente.

3.3.3 Plan de duración de las iteraciones.

Para un excelente entendimiento y como parte del ciclo de vida de la metodología de desarrollo de software XP, se establece el plan de duración de cada una de las iteraciones que se llevarán a cabo durante el desarrollo del proyecto. Este plan tiene como finalidad mostrar la duración de cada iteración, así como el orden en que deben ser implementadas las HU en cada una de ellas.

Iteración	Historias de Usuario	Duración total de Iteraciones
Iteración 1	Cargar Capa.	3 semanas
	Visualizar Capa.	
Iteración 2	Mostrar Capas.	3 semanas
Iteración 3	Acercar Mapa.	3 semanas.
	Alejar Mapa.	
	Escala Normal.	
	Mover Mapa.	

CAPÍTULO III: *Presentación de la Solución Propuesta.*

Iteración 4	Exportar a pdf.	3 semanas.
	Imprimir Imagen.	

Tabla #12 Plan de duración de las iteraciones.

3.4 Conclusiones Parciales.

En este capítulo se ha descrito la propuesta de solución para la herramienta de representación de información geográfica sobre tecnologías libres, abordando las dos primeras fases del ciclo de vida de la metodología XP. La primera fase de esta metodología es exploración, pues aquí se define el alcance del proyecto y el cliente describe a grandes rasgos las historias de usuario. Estas son de gran importancia para los desarrolladores, debido a que se realiza una breve descripción del comportamiento del sistema y el equipo de desarrollo se familiariza con las herramientas, tecnologías y prácticas que se utilizan en el proyecto, resultando de gran importancia para las fases posteriores. La fase de planificación, demostró que el desarrollo del software tendrá 4 iteraciones concluyendo que el proyecto demorará 12 semanas proporcionando rapidez en la entrega del producto y de esta forma ofreciendo satisfacción al cliente. También se documentaron todos los artefactos generados en el transcurso de las fases permitiendo al finalizar el proyecto contar con una exhaustiva información del mismo.

CAPÍTULO IV: Construcción de la solución propuesta.

Capítulo 4: Construcción de la solución propuesta.

4.1 Introducción.

En el presente capítulo se abordan las fases de construcción y prueba de la metodología XP, en el que se describen las tarjetas CRC (Contenido, Responsabilidad y Colaboración), para lograr un mejor entendimiento del sistema. Además se detallan las tareas de ingeniería por cada iteración, así como las pruebas unitarias y pruebas de aceptación que se le hacen a la aplicación, para conocer si estas trabajan correctamente.

4.2 Diseño de la solución propuesta.

La metodología ágil XP para el diseño de sus aplicaciones, no requiere la representación del sistema mediante diagramas de clase utilizando notación UML, sino que utiliza otras técnicas como las llamadas tarjetas CRC. Sin embargo la utilización de estos diagramas puede aplicarse siempre y cuando influya en el mejoramiento de la comunicación entre el equipo de desarrollo y se enfoquen en la información importante.

4.2.1 Tarjetas CRC.

Las tarjetas CRC permiten desprenderse del método de trabajo basado en procedimientos y trabajar con una metodología basada en objetos. Además permiten que el equipo completo ayude en la tarea del diseño.

Debido a la facilidad de su uso y entendimiento, se decidió utilizarlas para diseñar la aplicación que se desea desarrollar.

Tarjetas CRC	
Clase: Nombre de la clase que se está modelando.	
Responsabilidades: Es una descripción de alto nivel del propósito de la clase.	Colaboraciones: Indica con cuáles otras clases se requiere relación para cumplir la responsabilidad.

Tabla #13 Plantilla para Tarjetas CRC.

CAPÍTULO IV: Construcción de la solución propuesta.

Tarjetas CRC	
Clase: Conexion	
Responsabilidades: Open: Abre la conexión a la base de datos. Ejec_Obt_Consulta: Ejecuta y obtiene la consulta de la base de datos. CconvertToFloat: Quita los caracteres extraños devueltos por la consulta dejando solo los puntos (x, y). get_X: Retorna X geográfica. get_Y: Retorna Y geográfica.	Colaboraciones:

Tabla #14 Conexion.

Tarjetas CRC	
Clase: Principal	
Responsabilidades: Generar: Carga la cartografía y la muestra en el visor. print_: Imprime el mapa. MousePressEvent: Evento para recoger las coordenadas cuando es presionado el mouse. MouseReleaseEvent: Evento para recoger las coordenadas cuando es liberado el mouse. FitToWindow: Actualiza los actions luego de cargar la cartografía.	Colaboraciones: RedMapa

CAPÍTULO IV: Construcción de la solución propuesta.

<p>mostar_Capas: Muestra todas la capas que se encuentran dentro del archivo .map.</p> <p>Pintar_Imagen: Se encarga de graficar la imagen con funcionalidades de Mapserver.</p> <p>About: Información acerca de su creador.</p> <p>CreateActions: Crea todos los actions de la intefaz.</p> <p>Referencia: Grafica la referencia.</p> <p>exportar_pdf: Exporta la imagen del mapa en un pdf.</p> <p>CreateMenus: Se crean los menús de la interfaz.</p> <p>UpdateActions: Actualiza los actions.</p> <p>CreateToolBars: Crea el toolbar de la interfaz.</p> <p>CloseEvent: Indica cuando se ha cerrado la aplicación.</p>	
---	--

Tabla #15 Principal

Tarjetas CRC	
Clase: RedMapa	
<p>Responsabilidades:</p> <p>GenImagen: Gráfica la imagen del mapa perteneciente a la cartografía.</p> <p>ZoomPoint: Aumenta la visibilidad del mapa.</p> <p>FullExtent: Lleva el mapa a escala normal.</p>	<p>Colaboraciones: -</p>

CAPÍTULO IV: Construcción de la solución propuesta.

<p>Pan: Desplaza el mapa sobre el lienzo buscando el centro.</p> <p>NumCapas: Lleva el control de número de capas.</p> <p>GetLayerName: Retorna el nombre de la capa.</p> <p>GetLayer: Retorna el objeto capa.</p> <p>Act_Desc_Capas: Activa y desactiva las capas.</p> <p>PaneoImagen: Desplaza el mapa hacia cualquier posición del lienzo.</p> <p>mostrar_Leyenda: Muestra la leyenda del mapa, dando como resultado una imagen.</p> <p>Referencia: Muestra la referencia del mapa dando como resultado una imagen.</p> <p>convetirTopix_X: Convierte el pixel X en coordenadas geográficas.</p> <p>convetirTopix_Y: Convierte el pixel Y en coordenadas geográficas.</p>	
--	--

Tabla #16 RedMapa.

Tarjetas CRC	
Clase: Ui_Dialog	
Responsabilidades: setupUi retranslateUi	Colaboraciones: -

Tabla #17 Ui_Dialog

Tarjetas CRC	
Clase: edificio	
Responsabilidades:	Colaboraciones: -

CAPÍTULO IV: Construcción de la solución propuesta.

ObtEdificio: Devuelve las coordenadas geográficas del punto (x, y). Ed_X: Convierte el punto geográfico X a x pixel. Ed_Y: Convierte el punto geográfico Y a y pixel.	
---	--

Tabla #18 edificio

4.3 Desarrollo de las iteraciones.

En la segunda fase de la metodología XP, se describen las HU correspondientes a cada iteración a desarrollar, teniendo en cuenta las necesidades requeridas por el cliente. Durante el transcurso de las iteraciones se lleva a cabo una revisión del plan de iteraciones y se modifica en caso de ser necesario. Como parte de este plan, se descomponen las HU en tareas de programación o ingeniería, las mismas pueden ser escritas en lenguaje técnico debido a que son para uso estricto de los programadores, donde el usuario no tiene que necesariamente comprenderlas.

Teniendo en cuenta la planificación realizada se llevó a cabo el desarrollo del sistema en cuatro iteraciones, obteniéndose como finalidad una aplicación con todas las condiciones y características propuestas por el cliente. A continuación se definen cada una de las iteraciones.

4.3.1 Iteración 1:

El objetivo fundamental de esta iteración es dar cumplimiento a las HU que tienen mayor prioridad para el desarrollo del software. Una vez concluida esta iteración se contará con la funcionalidad descrita en la HU #1 y #2, las cuales hace alusión a todo lo referente con cargar y visualizar las capas.

Historias de Usuario	Tiempo de Implementación (semanas)
----------------------	------------------------------------

CAPÍTULO IV: Construcción de la solución propuesta.

	Estimación	Real
Cargar Capas.	2	2
Visualizar Capas.	1	1

Tabla #19 HU abordadas en la primera iteración.

A continuación mediante tablas se evidencian las tareas de programación o ingeniería donde la Historia de Usuario mencionada anteriormente fue desglosada, para un mejor funcionamiento de la aplicación.

Tareas de la Ingeniería	
No. de la tarea: 1	No. de la HU: 1
Nombre de la tarea: Cargar la cartografía del mapa a utilizar	
Tipo de tarea: Desarrollo.	Puntos estimados: 2
Fecha inicio: 7/1/2011	Fecha fin: 12/1/2011
Programador responsable: Adrian Núñez Rondón.	
Descripción: Pretende cargar toda la cartografía del sistema, para luego poder trabajar en la lógica del negocio.	

Tabla #20 Tarea de la HU #1

Tareas de la Ingeniería	
No. de la tarea: 1	No. de la HU: 1
Nombre de la tarea: Visualizar todas las capas de la cartografía.	
Tipo de tarea: Desarrollo.	Puntos estimados: 1.
Fecha inicio: 13/1/2011	Fecha fin: 19/1/2011
Programador responsable: Adrian Núñez Rondón.	
Descripción: Pretende visualizar todas las capas de la cartografía a trabajar en la aplicación.	

CAPÍTULO IV: Construcción de la solución propuesta.

Tabla #21 Tarea de la HU #2

4.3.2 Iteración 2:

La presente iteración tiene como objetivo desarrollar la HU#3. Una vez concluida esta iteración se cuenta con la funcionalidad descrita en la HU # 3, la cual hace alusión a todo lo referente con mostrar las capas que el usuario desee.

Historias de Usuario	Tiempo de Implementación (semanas)	
	Estimación	Real
Mostrar Capas.	3	3

Tabla #22 HU abordadas en la segunda iteración.

A continuación mediante tablas se demuestran las tareas de programación o ingeniería en las que la Historia de Usuario mencionada anteriormente fue desglosada, para un mejor funcionamiento de la aplicación.

Tareas de la Ingeniería	
No. de la tarea: 1	No. de la HU: 3
Nombre de la tarea: Mostrar todas las capas de la cartografía.	
Tipo de tarea: Desarrollo.	Puntos estimados: 3.
Fecha inicio: 20/1/2011	Fecha fin: 27/1/2011
Programador responsable: Adrian Núñez Rondón.	
Descripción: Pretende mostrar todas las capas de la cartografía a trabajar en la aplicación de acuerdo a las necesidades del usuario.	

Tabla #23 Tarea de la HU #3

CAPÍTULO IV: Construcción de la solución propuesta.

4.3.3 Iteración 3:

En esta iteración se cumplió con las HU que corresponden a los números #4, #5 y #6. Estas son referentes al zoom del mapa, le permiten al cliente acercar y alejar la parte seleccionada del mapa, así como permitir que vuelva a su escala normal, ofreciendo al usuario una mejor visualización.

Historias de Usuario	Tiempo de Implementación (semanas)	
	Estimación	Real
Acercar Mapa.	1	1
Alejar Mapa.	1	1
Escala Normal	1	1

Tabla #24 HU abordadas en la tercera iteración.

A continuación se evidencian las tareas de ingeniería en las que la Historia de Usuario mencionada anteriormente fue desglosada, para lograr un mejor funcionamiento del sistema.

Tareas de la Ingeniería	
No. de la tarea: 1	No. de la HU: 4
Nombre de la tarea: Aumentar referencia del mapa.	
Tipo de tarea: Desarrollo.	Puntos estimados: 1
Fecha inicio: 2/2/2011	Fecha fin: 10/2/2011
Programador responsable: Adrian Núñez Rondón.	
Descripción: Consiste en lograr una mayor visibilidad de los detalles del mapa.	

Tabla #25 Tarea de la HU #4

CAPÍTULO IV: Construcción de la solución propuesta.

Tareas de la Ingeniería	
No. de la tarea: 1	No. de la HU: 5
Nombre de la tarea: Disminuir referencia del mapa.	
Tipo de tarea: Desarrollo.	Puntos estimados: 1
Fecha inicio: 11/2/2011	Fecha fin: 19/2/2011
Programador responsable: Adrian Núñez Rondón.	
Descripción: Pretende disminuir visibilidad de los detalles del mapa.	

Tabla #26 Tarea de la HU #5

Tareas de la Ingeniería	
No. de la tarea: 1	No. de la HU: 6
Nombre de la tarea: Escala normal del mapa.	
Tipo de tarea: Desarrollo.	Puntos estimados: 1
Fecha inicio: 20/2/2011	Fecha fin: 28/2/2011
Programador responsable: Adrian Núñez Rondón.	
Descripción: Consiste en volver a mostrar el mapa en su escala normal.	

Tabla #27 Tarea de la HU #6

4.3.4 Iteración 4:

Esta iteración tiene como propósito desarrollar la HU #7, #8 y #9. Esta historia es la que se encarga de desplazar el mapa, esto le permite al usuario mover todo el lienzo o área de trabajo según le sea necesario, así como exportar e imprimir la imagen que el usuario desee.

CAPÍTULO IV: Construcción de la solución propuesta.

Historias de Usuario	Tiempo de Implementación (semanas)	
	Estimación	Real
Mover Mapa.	1	1
Exportar a pdf.	1	1
Imprimir Imagen	1	1

Tabla #28 HU abordadas en la cuarta iteración.

A continuación se evidencian las tareas de ingeniería en las que la Historia de Usuario mencionada anteriormente fue desglosada, para un lograr un mejor funcionamiento del sistema.

Tareas de la Ingeniería	
No. de la tarea: 1	No. de la HU: 7
Nombre de la tarea: Mover Mapa.	
Tipo de tarea: Desarrollo.	Puntos estimados: 1
Fecha inicio: 1/3/2011	Fecha fin: 8/3/2011
Programador responsable: Adrian Núñez Rondón.	
Descripción: Permite lograr una mejor visibilidad en detalles del mapa sobre cualquier punto que el usuario referencie en el lienzo.	

Tabla #29 Tarea de la HU #7

Tareas de la Ingeniería	
No. de la tarea: 1	No. de la HU: 8
Nombre de la tarea: Exportar imagen en formato pdf.	

CAPÍTULO IV: Construcción de la solución propuesta.

Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 9/3/2011	Fecha fin: 16/3/2011
Programador responsable: Adrian Núñez Rondón.	
Descripción: Consiste en permitir al usuario poder exportar la imagen a un formato pdf en la escala que este desee.	

Tabla #30 Tarea de la HU #8

Tareas de la Ingeniería	
No. de la tarea: 1	No. de la HU: 9
Nombre de la tarea: Imprimir Imagen.	
Tipo de tarea: Desarrollo.	Puntos estimados: 1
Fecha inicio: 18/3/2011	Fecha fin: 25/3/2011
Programador responsable: Adrian Núñez Rondón.	
Descripción: Permite lograr imprimir la imagen del mapa que el usuario desee.	

Tabla #31 Tarea de la HU #9

4.4. Pruebas.

Uno de los pilares de la metodología XP es el uso de las pruebas para comprobar el funcionamiento de los códigos que se vayan implementando. Esto permite aumentar la calidad de los sistemas reduciendo el número de errores no detectados y disminuyendo el tiempo transcurrido entre la aparición de un error y su detección. También permite aumentar la seguridad de evitar efectos colaterales no deseados a la hora de realizar modificaciones y refactorizaciones.

La metodología ágil XP divide las pruebas en dos grupos: pruebas unitarias y pruebas de aceptación. Las pruebas unitarias son desarrolladas por los programadores, los cuales se encargan de verificar el código automáticamente y las pruebas de aceptación están destinadas a verificar que al final de cada iteración las Historias de Usuarios cumplen con la funcionalidad asignada además de satisfacer las necesidades del cliente. (24).

CAPÍTULO IV: Construcción de la solución propuesta.

4.4.1 Pruebas unitarias

Las pruebas unitarias o pruebas de unidad consisten en comprobaciones (manuales o automatizadas) desarrolladas por los programadores. Las cuales se realizan para verificar que el código correspondiente a un módulo concreto se comporta de manera esperada.

Las pruebas unitarias proporcionan beneficios tales como:

- ✚ El programador puede realizar cambios de forma segura respaldado por efectivos casos de prueba.
- ✚ Permite saber si una determinada funcionalidad se puede agregar al sistema existente sin alterar el funcionamiento actual del mismo.
- ✚ Brindan al programador una inmediata retroalimentación de cómo está realizando su trabajo. (24)

4.4.2 Pruebas de aceptación.

Las pruebas de aceptación son consideradas como pruebas de caja negra. Los clientes son responsables de verificar que los resultados de estas pruebas sean correctos. Así mismo, en caso de que fallen varias pruebas, deben indicar el orden de prioridad de resolución. Una historia de usuario no se puede considerar terminada hasta tanto pase correctamente todas las pruebas de aceptación.

Las pruebas de aceptación son más importantes que las pruebas unitarias dado que significan la satisfacción del cliente con el producto desarrollado, el final de una iteración y el comienzo de la siguiente. Las pruebas de aceptación se elaboran a lo largo de la iteración, en paralelo con el desarrollo del sistema y adaptándose a los cambios que el sistema sufra. (24)

Caso de prueba de aceptación	
Código: HU1_P1	No. de la HU: 1

CAPÍTULO IV: Construcción de la solución propuesta.

Nombre: Comprobar que el mapa se cargue.
Descripción: Evaluar que el mapa se cargue.
Condiciones de Ejecución: El cliente debe probar que el mapa se cargue correctamente.
Entrada/ Pasos de ejecución: Se solicita la opción cargar para probar que se carga el mapa seleccionado.
Resultado Esperado: Se cargó correctamente el mapa.
Evaluación de la Prueba: Prueba satisfactoria.

Tabla #32 Prueba de aceptación para la HU "Cargar Capas."

Caso de prueba de aceptación	
Código: HU3_P1	No. de la HU: 3
Nombre: Comprobar que las capas del mapa se muestren.	
Descripción: Evaluar que las capas del mapa se muestren de acuerdo a las necesidades del usuario.	
Condiciones de Ejecución: El cliente debe probar que las capas del mapa se muestren correctamente.	
Entrada/ Pasos de ejecución: Se solicita la opción mostrar capas para probar que el cliente pueda elegir que capa mostrar.	
Resultado Esperado: Se mostraron correctamente las capas que el cliente eligió.	
Evaluación de la Prueba: Prueba satisfactoria.	

Tabla #33 Prueba de aceptación para la HU "Mostrar Capas."

Caso de prueba de aceptación

CAPÍTULO IV: Construcción de la solución propuesta.

Código: HU4_P1	No. de la HU: 4
Nombre: Comprobar que el mapa aumente de tamaño.	
Descripción: Evaluar que el mapa aumente de tamaño.	
Condiciones de Ejecución: El cliente debe probar que el mapa se aumenta correctamente.	
Entrada/ Pasos de ejecución: Se solicita la opción zoomIn para probar que el mapa aumenta de tamaño.	
Resultado Esperado: Aumentó correctamente el mapa de tamaño.	
Evaluación de la Prueba: Prueba satisfactoria.	

Tabla #34 Prueba de aceptación para la HU "Acercar Mapa."

Caso de prueba de aceptación	
Código: HU5_P1	No. de la HU: 5
Nombre: Comprobar que el mapa disminuye de tamaño.	
Descripción: Evaluar que el mapa disminuye de tamaño	
Condiciones de Ejecución: El cliente debe probar que el mapa se disminuya correctamente de tamaño.	
Entrada/ Pasos de ejecución: Se solicita la opción zoom out para probar que el mapa disminuye de tamaño.	
Resultado Esperado: Disminuyó correctamente el mapa de tamaño.	
Evaluación de la Prueba: Prueba satisfactoria.	

Tabla #35 Prueba de aceptación para la HU "Alejar Mapa."

CAPÍTULO IV: Construcción de la solución propuesta.

Caso de prueba de aceptación	
Código: HU6_P1	No. de la HU: 6
Nombre: Escala Normal.	
Descripción: Evaluar que el mapa vuelve a su escala normal.	
Condiciones de Ejecución: El cliente debe probar que el mapa vuelva a su escala normal correctamente.	
Entrada/ Pasos de ejecución: Se solicita la opción escala normal para probar que el mapa volvió a su proporción original.	
Resultado Esperado: El mapa volvió a su escala normal.	
Evaluación de la Prueba: Prueba satisfactoria.	

Tabla #36 Prueba de aceptación para la HU "Escala Normal."

Caso de prueba de aceptación	
Código: HU7_P1	No. de la HU: 7
Nombre: Comprobar que el mapa se desliza por el lienzo.	
Descripción: Evaluar que el mapa se desliza por todo el lienzo.	
Condiciones de Ejecución: El cliente debe probar que el mapa se deslice correctamente por todo el lienzo.	
Entrada/ Pasos de ejecución: Se solicita la opción paneo para probar que el mapa se desliza.	
Resultado Esperado: Se deslizó correctamente el mapa.	
Evaluación de la Prueba: Prueba satisfactoria.	

Tabla #37 Prueba de aceptación para la HU "Mover Mapa."

CAPÍTULO IV: Construcción de la solución propuesta.

Caso de prueba de aceptación	
Código: HU8_P1	No. de la HU: 8
Nombre: Exportar pdf.	
Descripción: Evaluar que la imagen se exporta en pdf.	
Condiciones de Ejecución: El cliente debe probar que la imagen del mapa se exporta correctamente en pdf	
Entrada/ Pasos de ejecución: Se solicita la opción exportar a pdf para probar que la imagen del mapa se exportó.	
Resultado Esperado: Se exportó correctamente la imagen a pdf.	
Evaluación de la Prueba: Prueba satisfactoria.	

Tabla #38 Prueba de aceptación para la HU "Exportar a Pdf."

Caso de prueba de aceptación	
Código: HU9_P1	No. de la HU: 9
Nombre: Imprimir imagen.	
Descripción: Evaluar que la imagen se imprima.	
Condiciones de Ejecución: El cliente debe probar que la imagen del mapa que el usuario desee se imprima correctamente.	
Entrada/ Pasos de ejecución: Se solicita la opción imprimir para probar la imagen del mapa se imprimió.	
Resultado Esperado: Se imprimió correctamente la imagen del mapa.	
Evaluación de la Prueba: Prueba satisfactoria.	

Tabla #39 Prueba de aceptación para la HU "Imprimir Imagen."

CAPÍTULO IV: Construcción de la solución propuesta.

4.5 Conclusiones parciales

Con la culminación de este capítulo se le da cumplimiento al objetivo del presente trabajo, debido a que se desarrolla una herramienta informática sobre tecnologías libres que posibilita la integración con el servidor de mapas MapServer. Además permitió integrarse como núcleo a un plugin desarrollado para QuantumGis, logrando una mejor visualización de los disímiles mapas y permitir cargar archivos de extensión .map. Además se detallaron las tareas de ingeniería por cada iteración, así como se realizó un estudio profundo entre las pruebas unitarias y las pruebas de aceptación demostrando que las pruebas de aceptación son las indicadas para comprobar el funcionamiento del sistema, conociendo si este trabaja correctamente, es decir proporcionando la satisfacción del cliente con el producto desarrollado.

Conclusiones Generales

Conclusiones Generales

Concluyendo el desarrollo del sistema para la representación de información geográfica sobre tecnologías libres, utilizando como servidor de mapas MapServer, se ha arribado a las siguientes conclusiones:

- ✚ La solución propuesta constituye la primera aplicación de escritorio desarrollada desde cero sobre tecnologías libres capaz de integrarse con el servidor de mapas MapServer.
- ✚ Se insistió en todo momento en la utilización de software y herramientas libres atendiendo a una sana política de independencia y libertad de acción futura previendo, una vez implementado el sistema, la incorporación paulatina de mejoras en posteriores versiones del mismo.
- ✚ Las herramientas, la metodología de desarrollo del software y el lenguaje utilizado, brindaron el soporte necesario para lograr un producto con los requerimientos deseados, además de proporcionarle al mismo una calidad y un rendimiento acorde a las exigencias planteadas por el cliente.
- ✚ En la metodología XP las Historias de Usuario representan una breve descripción del comportamiento del sistema, estas fueron debidamente implementadas.
- ✚ Las pruebas de aceptación permitieron la validación del sistema propuesto para verificar el cumplimiento del funcionamiento del sistema.

Se cumplieron los objetivos de la investigación, pues se obtuvo un Sistema de Información Geográfica que cumple con las exigencias del departamento Geoinformática, demostrando que se puede lograr mayor similitud tecnológica, entre los productos de plataformas web y de escritorio consiguiendo la integración con el servidor de mapas MapServer.

Recomendaciones

Recomendaciones

Las metas planteadas con este trabajo se han cumplido y en base a los resultados obtenidos se recomienda:

-  Agregar un módulo de seguridad.
-  Adicionar un módulo de estructura y composición.
-  Complementar un módulo de edición.

Referencias Bibliográficas

Referencias Bibliográficas

1. Información. [Online] [Cited: 11 1, 2010.] <http://www.promonegocios.net/mercadotecnia/definición-información.html>.
2. Ssistema de información geográfica. [Online] [Cited: 11 5, 2010.] (http://www.mappinginteractivo.com/plantilla-ante.asp?id_articulo=1184).
3. mapa. [Online] [Cited: 10 15, 2010.] <http://definicion.de/mapa/>.
4. Gis. [Online] [Cited: 10 20, 2010.] <http://www.concla.net/Geografia/GIS.html>.
5. Diccionario Babylon. [Online] 2009. [Cited: 10 30, 2010.]
6. Sig. [Online] [Cited: 10 16, 2010.] http://app.jalisco.gob.mx/enlaceit.nsf/Ed6_Posg?OpenPage.
7. [Online] [Cited: 11 12, 2010.] <http://gsyc.escet.urjc.es/~grex/sobre-libre/definicion-software-libre.html>.
8. [Online] [Cited: 12 1, 2010.] <http://www.slideshare.net/coleaspheres/tecnologa-open-source-en-la-empresa-moderna>.
9. [Online] [Cited: 10 3, 2010.] www.sigte.udg.edu/.
10. Funcionalidades de los SIG. [Online] [Cited: 11 29, 2010.] www.construmatica.com.
11. [Online] [Cited: 10 29, 2010.] www.humboldt.org.co.
12. [Online] [Cited: 11 29, 2010.] (<http://arqueologiasig.blogspot.com/>).
13. Esri. [Online] [Cited: 12 1, 2010.] <http://mastergis.esri.es/productos.html#>.
14. Grass. [Online] [Cited: 11 29, 2010.] www.scribd.com/doc/2096954/SIG-Libres.
15. Qgis. [Online] [Cited: 11 29, 2010.] <http://www.qgis.org>.
16. Dominguez, javier. *Breve Introducción a la Cartografía y a los Sistemas de Información Geografica*. Mexico : s.n., 2002.
17. Duque, Raul Gonzalez. *Python para todos*. España : s.n.
18. Lockhart, Thomas. *Manual del usuario de PostgreSQL*. California : s.n., 1996.
19. http://www.postgresql-es.org/sobre_postgresql.
20. Arquba.com.ar. [En línea] [Citado el: 15 de 1 de 2011.] www.arquba.com.ar/software-gratis/quantum-gis/.

Referencias Bibliográficas

21. PyQt. [En línea] [Citado el: 12 de 1 de 2011.] <http://www.slideshare.net/juancabicho/py-qt-desarrollando-aplicaciones-de-escritorio-el-co-di-go-k-presentation>.
22. Servidor de mapa. [En línea] [Citado el: 15 de 2 de 2011.] metadatos.ingemmet.gob.pe/files/CURSO/5.-%20mapserver.ppt.
23. Eclipse. [En línea] [Citado el: 2 de 2 de 2011.] blog.elartedeprogramar.cl/page.
24. Pérez, Isaías Carrillo. *METODOLOGIA DE DESARROLLO de software*. 2005. 2.
25. Candeaux, Luís R. Díaz Cisneros y Rafael. *Revista Internacional de Ciencias de la Tierra*. 1994.
26. Buenmaster. [En línea] [Citado el: 15 de 3 de 2010.] <http://www.buenmaster.com/?a=536>.
27. MVC. [En línea] [Citado el: 10 de 3 de 2010.] <http://msdn.microsoft.com/es-es/library/bb972251.aspx#M19>.

Bibliografías Consultadas

Bibliografías Consultadas

1. Sig. [Online] [Cited: 10 16, 2010.] http://app.jalisco.gob.mx/enlaceit.nsf/Ed6_Posg?OpenPage.
2. [Online] [Cited: 11 12, 2010.] <http://gsyc.escet.urjc.es/~grex/sobre-libre/definicion-software-libre.html>.
3. [Online] [Cited: 12 1, 2010.] <http://www.slideshare.net/coleaspheres/tecnologia-open-source-en-la-empresa-moderna>.
4. [Online] [Cited: 10 3, 2010.] www.sigte.udg.edu/.
5. Funcionalidades de los SIG. [Online] [Cited: 11 29, 2010.] www.construmatica.com.
6. [Online] [Cited: 10 29, 2010.] www.humboldt.org.co.
7. [Online] [Cited: 11 29, 2010.] (<http://arqueologiasig.blogspot.com/>).
8. Esri. [Online] [Cited: 12 1, 2010.] <http://mestergis.esri.es/productos.html#>.
9. Eclipse. [En línea] [Citado el: 2 de 2 de 2011.] blog.elartedeprogramar.cl/page.
10. Pérez, Isaías Carrillo. *METODOLOGIA DE DESARROLLO de software*. 2005. 2.
11. Candeaux, Luís R. Díaz Cisneros y Rafael. *Revista Internacional de Ciencias de la Tierra*. 1994.