

UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS

Facultad 6



**TÍTULO: Representación de información geográfica en 3D
usando la herramienta de escritorio QGIS.**

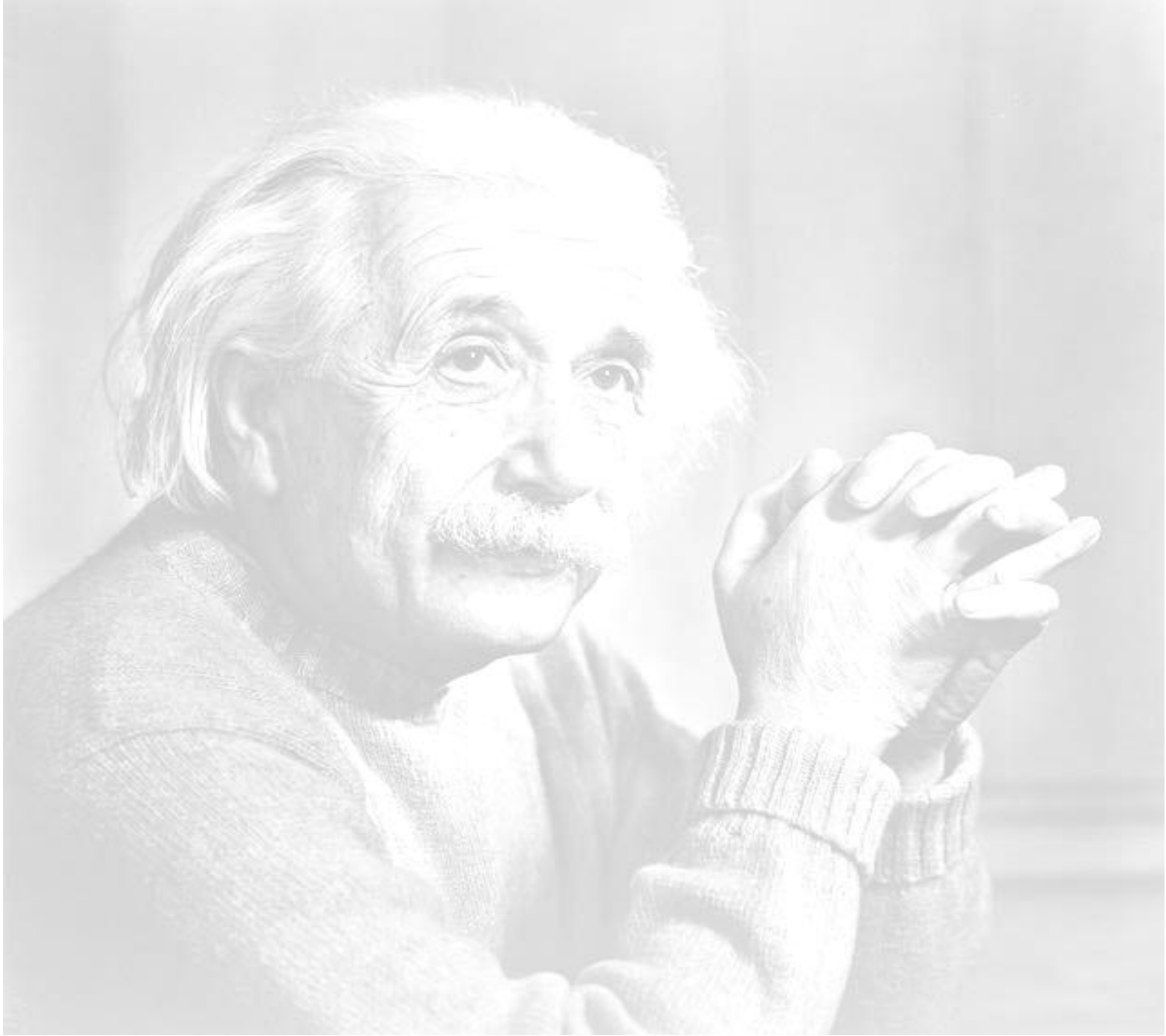
**TRABAJO DE DIPLOMA PARA OPTAR POR EL TÍTULO DE
INGENIERO EN CIENCIAS INFORMÁTICAS**

AUTOR: Pavel Raúl Romero Roldós.

TUTOR: Lic. Yaima Fernández Segredo.

Ciudad de La Habana Abril 2011.

Año 53 de la Revolución



Nunca consideres el estudio como una obligación, sino como una oportunidad para penetrar en el bello y maravilloso mundo del saber.

Declaración de Autoría

DECLARACIÓN DE AUTORÍA

Declaro que soy el único autor de este trabajo y autorizo al Departamento de Geoinformática de la Facultad 6 de la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

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

Autor: Pavel Raúl Romero Roldós

Tutor: Lic. Yaima Fernández Segredo.

Datos de Contacto

Dedicatoria

Dedicatoria

A mi madre lo más grande de este mundo, sin ti nada sería posible.

A mi hermanita que es el brillo de mis ojos.

A toda mi familia.

A todos mis amigos, esos hermanos que te va dando la vida.



Agradecimientos

Agradecimientos

A mi mamá por nunca rendirse, por sacrificarse, por buscar siempre la forma de salir adelante.

A mi hermanita por hacerme reír incluso en los momentos difíciles.

A mi papa, a mis hermanos, a mis abuelos.

A mis tíos, a mis primos.

A mis viejas amistades de Guantánamo, a los vecinos del barrio.

A todos los amigos con los que he compartido a lo largo de los años, la vieja guardia.

A las amistades de la universidad por estos magníficos años.

A los Pentium boys, lo último que trajo el barco.

A mis amigos Adrian, Daniel y todos los compañeros del proyecto que ayudaron en el desarrollo de esta investigación.

A todos los profesores que han contribuido en mi formación, a los miembros del tribunal, en especial a mi tutora Yaima

A todas aquellas personas que de una forma u otra me han ayudado.

A todos ustedes, gracias.





Resumen

La cartografía digital está experimentando en los últimos tiempos grandes avances debido a la popularización de los Sistemas de Información Geográfica (SIG). Estos sistemas permiten dar solución a un problema tan antiguo como la representación de la superficie terrestre utilizando las nuevas tecnologías de la información y las telecomunicaciones. Otra área que ha recibido un notable impulso con el advenimiento de las nuevas tecnologías es la visualización de objetos en tres dimensiones. Algunos de los SIG brindan la opción de visualizar la superficie terrestre de esta forma.

La herramienta QGIS es la base de varias aplicaciones SIG desarrolladas en la Universidad de las Ciencias Informáticas; esta no cuenta con la funcionalidad de mostrar la información geográfica en 3D. Para dar solución a esta problemática se decidió desarrollar una extensión que permita la representación de la información geográfica en 3D para dicha herramienta.

Para lo cual se investigaron soluciones existentes, determinado que ninguna constituye de forma total o parcial a la solución de la problemática planteada. Se prosiguió con el estudio de los métodos de representación de la información geográfica en tres dimensiones, las redes irregulares trianguladas y el algoritmo de triangulación de Delaunay. Concluyendo con la aplicación de estos a la implementación de la extensión propuesta.

Además, se determinó la metodología XP para guiar el proceso de desarrollo de la extensión. El lenguaje Python para la implementación de la misma, con la inclusión de la librería PyQt para el trabajo con las interfaces gráficas de usuario. Así como la biblioteca OpenGL para la graficación del modelo y como IDE se definió el Eclipse.

Proveyendo, finalmente, a la herramienta QGIS de una extensión que permite la representación de la información geográfica en tres dimensiones. Posibilitando representar dicha información con mayor claridad y precisión.

Palabras Claves: Modelado de superficies, Sistema de Información Geográfica, Gráficos 3D.



Índice de Contenido

Introducción	1
Capítulo 1: Fundamentación Teórica de los Sistemas de Información Geográfica (SIG).....	5
Capítulo 2: Tecnologías y herramientas a utilizar.	22
2.1 Programación Extrema (XP)	22
2.2 Patrones GRASP.....	25
2.3 El lenguaje de programación Python.....	27
2.4 Entorno de desarrollo Integrado (IDE) Eclipse.	27
2.5 OpenGL	28
2.6 GDAL	29
2.7 QGIS.....	30
2.8 Conclusiones Parciales.....	31
Capítulo 3: Presentación de la solución propuesta.	33
3.1 Fase de Exploración.....	33
3.1.1 Historias de Usuarios	33
3.2 Fase de planificación.....	39
3.2.1 Estimación de esfuerzos por historia de usuario.....	40
3.2.2 Plan de iteraciones	40
3.3 Plan de duración de las iteraciones.	41
3.4 Conclusiones Parciales.....	42
Capítulo 4: Construcción de la solución propuesta.	43
4.1 Diseño de la solución propuesta	43
4.1.1 Tarjetas CRC	43
4.2 Desarrollo de las iteraciones.....	47
4.2.1 Iteración 1:	48
4.2.2 Iteración 2:.....	49
4.3 Pruebas	51
4.3.1 Pruebas unitarias.....	51
4.3.2 Pruebas de aceptación.....	52
4.4 Conclusiones parciales.....	57
Conclusiones Generales	58
Recomendaciones	59
Referencias Bibliográficas.....	60
Glosario:.....	61

Índice de Figuras

fig. 1 Modelo ráster	7
fig. 2 Formato vectorial	8
fig. 3 Superficie de un terreno modelada con TIN.....	10
fig. 4 Triangulación de Delaunay	11
fig. 5 Flip.....	11
fig. 6 Algoritmo de Construcción Incremental.....	13
fig. 7 Test de pertenencia.....	13
fig. 8 Construcción incremental.....	14
fig. 9 Construcción incremental.....	15
fig. 10 Generación de sombreado	16
fig. 11 Tintas hipsométricas.....	17

Índice de Tablas

Tabla 1 Descripción de la HU	37
Tabla 2 Cargar Archivo Shape.....	38
Tabla 3 Mover vista.....	39
Tabla 4 Estimación de esfuerzo por historia de usuario.....	40
Tabla 5 Plan de duración de las iteraciones.....	41
Tabla 6 Plantilla para Tarjetas CRC.....	44
Tabla 7 Triangulación	45
Tabla 8 Extractora	45
Tabla 9 Dialogo	46
Tabla 10 Grafico.....	46
Tabla 11 Controladora	47
Tabla 12 HU abordadas en la primera iteración.....	48
Tabla 13 Tarea de la HU #1	49
Tabla 14 HU abordadas en la segunda iteración	50
Tabla 15 Tarea de la HU #2.....	50
Tabla 16 Prueba de aceptación 1 para la HU1 Cargar archivo Shape.....	53
Tabla 17 Prueba de aceptación 1 para la HU2 Mover Vista.....	53
Tabla 18 Prueba de aceptación 2 para la HU2 Mover Vista.....	54
Tabla 19 Prueba de aceptación 3 para la HU2 Mover Vista.....	55
Tabla 20 Prueba de aceptación 4 para la HU2 Mover Vista.....	56
Tabla 21 Prueba de aceptación 5 para la HU2 Mover Vista.....	56
Tabla 22 Prueba de aceptación 6 para la HU2 Mover Vista.....	57

Introducción

Introducción

El desarrollo de la informática, las telecomunicaciones y el surgimiento del internet ha dado lugar a la era de la información con repercusiones en todos los niveles de la sociedad. La aparición de los Sistemas de Información Geográfica (SIG¹) fue necesaria para dar respuestas a las numerosas necesidades de esta época, saturada de información electrónica y automatizada, donde las técnicas de adquisición de datos de la superficie terrestre y las capacidades de cómputo han mejorado notablemente.

Los Sistemas de Información Geográfica son elementos relacionados entre sí: hardware, software, personal, procesos e información geográficamente referenciada. Que tienen como objeto integrar, almacenar, editar, analizar, compartir y mostrar dicha información. En un sentido más técnico, los SIG son herramientas que permiten a los usuarios crear consultas interactivas, analizar la información espacial, editar datos, mapas y presentar los resultados de todas estas operaciones. Son consecuencia directa de la utilización de las Tecnologías de la Información y las Comunicaciones (TIC), que pueden almacenar y manipular grandes volúmenes de información. Mediante el uso de las TIC la información geográfica puede ser modificada, cuantificada y cualificada de manera eficiente, debido a la posibilidad que brindan los ordenadores para ejecutar varios procesos a la vez.

Otra área que ha recibido un notable impulso con el advenimiento de las nuevas tecnologías es la computación gráfica. Las imágenes generadas por los ordenadores se hacen más realistas e interactivas. Algunos de los SIG brindan la opción de mostrar la superficie terrestre en tres dimensiones. La visualización en 3D permite mejorar la comprensión del problema, mayor nivel de detalle y mejora el proceso de toma de decisiones de algunos negocios.

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. En ésta se desarrollan SIG vinculados a la minería, control de riesgos, agricultura y transporte. Uno de los proyectos de investigación y desarrollo que trabaja en esta línea es GeoQ el cual está destinado a la creación de aplicaciones digitales para un

¹ Esta terminología se usará de esta forma en todo el documento.

Introducción

entorno de escritorio. Como base para la creación de las mismas se utiliza la herramienta QGIS que no cuenta con la funcionalidad de mostrar la información geográfica en tres dimensiones. Siguiendo el avance de las TIC la universidad se ha propuesto agregar a las nuevas aplicaciones SIG de escritorio la funcionalidad de representar la información geográfica en 3D.

Para lo que **se defiende** que el desarrollo de una extensión, para la herramienta de escritorio QGIS, que permita la representación de información geográfica en 3D y la elaboración de su respectiva documentación técnica, permitirá representar la información geográfica con mayor claridad y precisión.

Definiéndose como **problema a resolver**, cómo representar información geográfica en 3D. Para lo cual se define como **objeto de estudio** los métodos de representación de la información geográfica en 3D, específicamente en el **campo de acción** de la herramienta QGIS.

Dando lugar al siguiente **objetivo general** de la presente investigación, desarrollar una extensión que permita la representación de la información geográfica en 3D específicamente para la herramienta QGIS.

Métodos de investigación

Los métodos de investigación se clasifican en empíricos y teóricos. En esta investigación se utilizan algunos métodos teóricos tales como el Analítico Sintético, el Histórico Lógico y el de Modelación. Estos métodos permiten estudiar las características del objeto de investigación que no son observables directamente, facilitan la construcción de modelos e hipótesis de investigación y crean las condiciones para ir más allá de las características fenomenológicas y superficiales de la realidad, contribuyendo al desarrollo de las teorías científicas y para su ejecución se apoyan en el proceso de análisis y síntesis posibilitando el conocimiento del estado del arte del fenómeno(Martinto, 2009).

- **Analítico–sintético:** Para la realización del estudio a profundidad de toda la información acerca de las tecnologías, metodologías y herramientas con posibilidad de ser utilizadas en el desarrollo de la extensión. Concluyendo con una selección apropiada para esta investigación y sintetizando las características y viabilidad de cada una.

Introducción

- **Histórico-Lógico:** Utilizado en la definición de las tendencias actuales, en el área de la visualización de la información geográfica en tres dimensiones. Los principales conceptos que giran a su alrededor, las soluciones existentes en el mundo, así como las diversas aplicaciones y funciones que pueden servir de punto de partida para la implementación de una extensión, para la herramienta QGIS, que cumpla con las exigencias de la misma.
- **Modelación:** Utilizado en la realización de los modelos correspondientes al ciclo de vida del sistema. Facilitando el cumplimiento de las tareas de análisis y diseño de los procesos que intervienen en la aplicación.

Desglose, por capítulos, de la presente investigación

La presente investigación se desglosa en cuatro capítulos. Un primer capítulo dedicado a la fundamentación teórica entorno a los SIG² y la representación de la información geográfica en tres dimensiones. Un segundo capítulo en el que se describen el análisis realizado y las decisiones tomadas en cuanto a las tecnologías y herramientas a utilizar para llevar a cabo el desarrollo de la extensión. Dos capítulos finales en los que se presenta la solución propuesta y se realiza la fase construcción y prueba de la misma.

Capítulo 1. Los SIG y la representación de la información geográfica en 3D: En este capítulo se describen brevemente los conceptos fundamentales relacionados con el dominio del problema. Se realiza un estudio acerca de los SIG, los tipos de formatos de la información geográfica: ráster y vectorial, las redes irregulares trianguladas (TIN), los algoritmos de triangulación, en especial el de Delaunay. Además, se hace un análisis de algunas soluciones existentes que brindan la funcionalidad de mostrar la información geográfica en tres dimensiones.

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.

² Sistema de Información Geográfica

Introducción

Capítulo 3. Presentación de la solución propuesta: Aborda las dos primeras fases de la programación extrema (XP). En la fase de exploración se define las historias de usuario, 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: Aborda las restantes fases de la programación extrema, construcción y prueba. Donde se describen las tarjetas CRC³ y se detallan las 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.

³ Clase, Responsabilidad y Conexión

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

En este capítulo se lleva a cabo un estudio de los principales conceptos asociados al tema de investigación. Además, contiene los elementos que justifican y soportan teóricamente la solución propuesta. Se realiza un análisis de la forma en que se lleva a cabo el proceso de representación de la información geográfica en 3D a nivel internacional y sobre las características generales de soluciones existentes.

1.1 Conceptos asociados al dominio del problema

Para la comprensión de esta investigación es preciso contar con un conocimiento básico del tema de los SIG y la representación de la información geográfica, centrándonos en la información geográfica en 3D. Con este fin se define, a continuación, los conceptos básicos que giran en torno a este tema, entre los cuales se mencionan, Sistema de Información Geográfica (SIG), Datos espaciales (DE), Información geográfica (IG), georreferenciación, mapa, formato ráster y formato vectorial.

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.(Sistema de información geográfica.)

Datos Espaciales

Son el componente fundamental de cada proyecto o aplicación SIG. Contienen las ubicaciones y formas de características cartográficas. Son también conocidos como datos cartográficos digitales, el tipo de datos necesarios para crear mapas y estudiar relaciones espaciales. Dentro de su contexto, almacenan informaciones sobre la localización y las formas de un objeto geográfico y las relaciones entre ellos, normalmente, con coordenadas y topología. Refieren a entidades o fenómenos que cumplen los siguientes principios básicos:

- Posición absoluta: sobre un sistema de coordenadas (x, y, z).

CAPÍTULO 1: Los SIG y la representación de la información geográfica en 3D

- Posición relativa: frente a otros elementos del paisaje (topología: incluido, adyacente, cruzado).
- Figura geométrica que representan (punto, línea, polígono).
- Atributos que describen (características del elemento o fenómeno).(Sig)

Información geográfica

Es un conjunto de datos espaciales, los cuales brindan una información de algún hecho o fenómeno. (Diccionario Babylon, 2009).

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 utilizada en los Sistemas de Información Geográfica. (Diccionario Babylon, 2009).

Mapa

El concepto de mapa proviene del término latino mapa. Se trata de una representación gráfica y métrica de una porción de territorio sobre una superficie bidimensional, que por lo general suele ser plana, aunque también puede ser esférica como en el caso de los globos terráneos. Son un conjunto de datos espaciales representados mediante signos, símbolos gráficos y colores. (mapa)

1.2 Formas básicas de modelado de datos

En un SIG pueden modelarse los datos en dos formas básicas: como una colección de rasgos discretos en formato vectorial o como un conjunto de celdas con datos de atributos para cada una de ellas, formato ráster.

Formato ráster

Este formato representa datos continuos o de imágenes. Sobresale por almacenar y trabajar con datos continuos, como elevación, tabla de agua, concentración de polución y nivel de ruido ambiental. Cada celda (pixel) es una cantidad medida. La fuente más común es una imagen de satélite o fotografía aérea.

CAPÍTULO 1: Los SIG y la representación de la información geográfica en 3D

El modelo ráster divide la región en bloques rectangulares (celdas de grilla o pixels) que son completados con el valor del atributo medido. La aproximación está directamente relacionada a los marcos que controlan el espacio para poder medir los atributos. Las celdas pueden ser colocadas dentro de un sistema de referencia espacial, pero ellas deliberadamente limitan la resolución para que actúe como control. Un aspecto a considerar aquí es el tamaño de las celdas. Las celdas más pequeñas permiten al formato ráster aproximarse a la flexibilidad del formato vectorial tan cerca como sea requerido, pero surge a un alto costo por el espacio de almacenamiento. (Alzate, 2008)

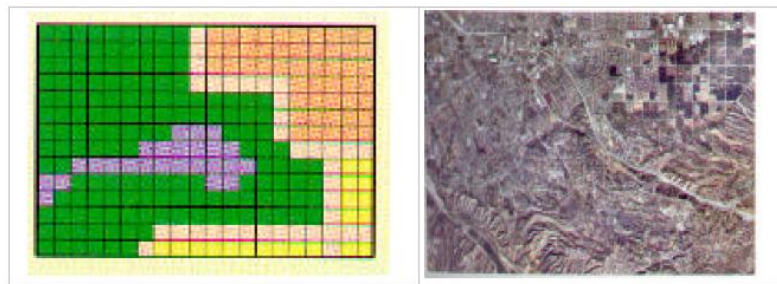


fig. 1 Modelo ráster

Formato vectorial

Los datos vectoriales representan rasgos tales como puntos, líneas y polígonos. Se aplica mejor a objetos con bordes y formas definidas. Los rasgos tienen una posición y forma precisa, atributos y metadatos. Los marcos de medición basados en el control de los atributos son implementados directamente, usando el modelo vectorial. Con base en la geometría analítica, un modelo vectorial construye una representación compleja a partir de los elementos primitivos: puntos, líneas y áreas. Los elementos primitivos tienen una dependencia entre sí: las áreas son descritas por líneas de límites y la localización de una línea puede ser aproximada por una cadena de segmentos de líneas que conectan una serie de puntos. En la base, los puntos son representados por coordenadas. (Alzate, 2008)



fig. 2 Formato vectorial

1.3 Técnicas de modelado y visualización de terrenos

Cuando el problema a resolver está dado por la visualización interactiva de superficies de terrenos en tres dimensiones, el consenso general en la comunidad científica es llevar a cabo la representación de las mismas mediante mallas poligonales. Generalmente se utilizan triángulos, debido a que tres puntos siempre determinan un único plano y por tanto su representación se hace más eficiente. Es por esta razón que para esta clase de polígonos se han desarrollado técnicas avanzadas que aceleran su representación (rendering), convirtiéndose en la primitiva de dibujo más usada en gráficos de computadoras. (Alzate, 2008)

1.4 Redes Irregulares trianguladas (TIN)

CAPÍTULO 1: Los SIG y la representación de la información geográfica en 3D

Diversas son las técnicas aplicadas en la representación de modelos digitales de terrenos. Generalmente se utilizan redes de polígonos, juntándolos para imitar la superficie terrestre. Los triángulos son polígonos simples y tienen propiedades favorables tales como que representan una superficie coplanar, de aquí que estos sean los polígonos más usados para la representación de terrenos. Los datos generalmente se presentan de dos formas, distribuidos irregularmente en el plano o formando mallas rectangulares regulares. La solución representa la superficie del terreno usando las redes irregulares trianguladas. Este es un modelo eficiente y exacto para representar superficies continuas.

Un conjunto de datos TIN es construido a partir de una serie de puntos con coordenadas X, Y y Z a través de instrumentos fotogramétricos, Sistemas de Posicionamiento Global (GPS) u otros medios. Luego se colectan las líneas de quiebre, donde la forma de la superficie cambia fuertemente, y las áreas de exclusión. A partir del conjunto de puntos el SIG crea una red opcional de triángulos, llamados triangulación Delaunay. En un TIN cada triángulo es creado para ser lo más cercano posible a un triángulo equilátero. La elevación puede ser calculada para cualquier punto con valores X y Y, primero localizando el triángulo al que pertenece y después interpolando la altura dentro de este.

Se plantea que un TIN es eficiente dado que la densidad de puntos sobre cualquier lugar de la superficie puede ser proporcional a la variación en el terreno. Para una superficie plana es suficiente con una densidad baja de puntos mientras que un terreno montañoso requiere una densidad alta de puntos, especialmente donde la superficie cambia abruptamente. Permite integrar las líneas de quiebre del terreno y las áreas de exclusión.

Hay diversas formas de visualizar la superficie representada por un TIN. El TIN puede ser dibujado sobre un mapa planimétrico (bidimensional) con colores, representando elevación, pendiente y aspecto. Los métodos de tintas hipsométricas y generación de sombreado generan ese efecto. También se puede desplegar en tres dimensiones mostrando vistas con perspectiva de una superficie cubierta por imágenes, curvas de contornos, líneas de la malla u otros rasgos. (Alzate, 2008)

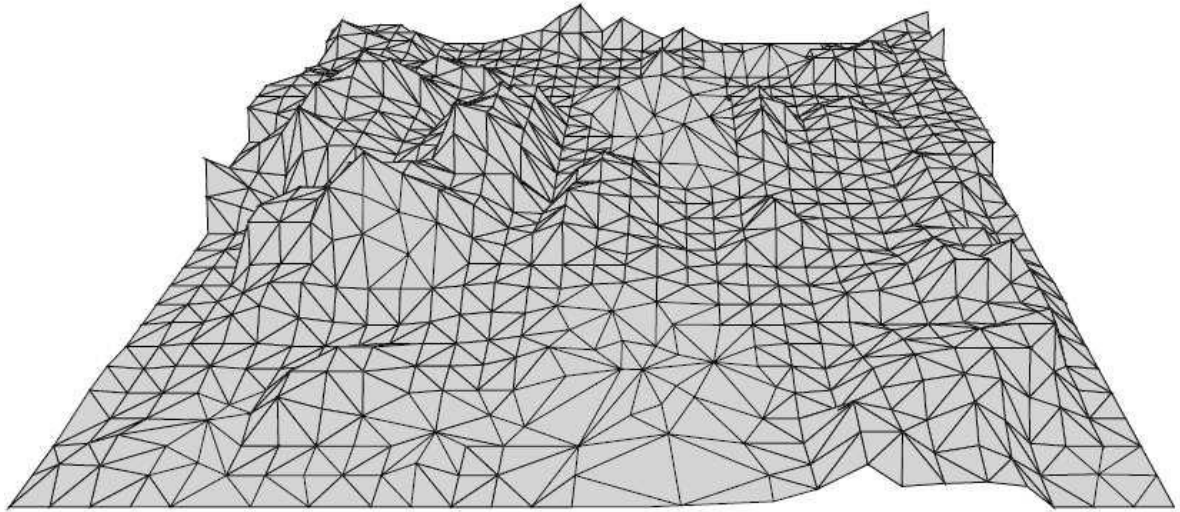


fig. 3 Superficie de un terreno modelada con TIN.

1.5 Triangulación de Delaunay

Una triangulación de Delaunay es una red de triángulos que cumple la condición de Delaunay. La cual plantea que la circunferencia circunscrita de cada triángulo de la red no debe contener ningún vértice de otro triángulo. La triangulación de Delaunay maximiza los ángulos interiores de los triángulos de la red. Al usar la triangulación como modelo tridimensional los errores de redondeo son mínimos. Esto ha contribuido a que triangulaciones de Delaunay sean generalmente utilizadas en aplicaciones gráficas y en particular en la graficación en 3D. (O'Rourke, 1998)

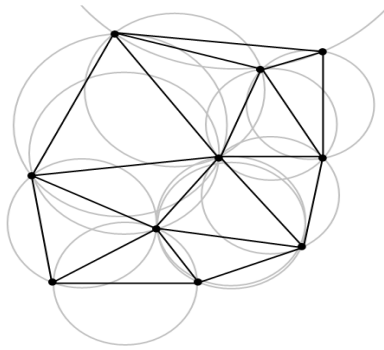


fig. 4 Triangulación de Delaunay

Flipping

Si dos triángulos tienen una arista en común y cumplen la condición de que en el área de su circunferencia circunscrita no está contenido un vértice de otro triángulo, la arista compartida se considera legal. Si dos triángulos no cumplen la condición de Delaunay al ser reemplazada su arista común por la otra arista que puede serlo, los triángulos obtenidos cumplirán dicha condición. Este proceso de sustitución se le denomina flip.

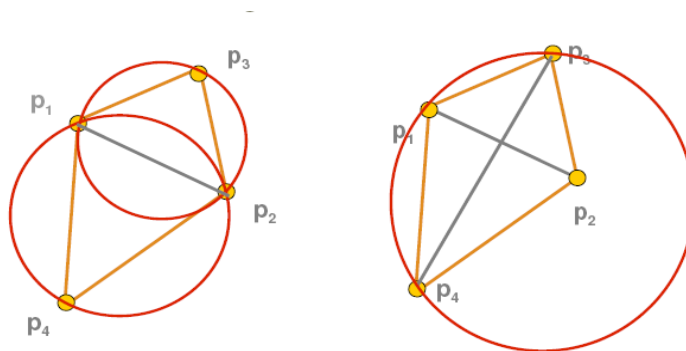


fig. 5Flip

En la fig.5 los triángulos $p_1p_2p_3$ y $p_2p_3p_4$ adyacentes a la arista p_1p_2 cumplen la condición de Delaunay en consecuencia p_1p_2 es legal. El caso contrario es la arista p_3p_4 que al no ser legal debe sustituirse por la arista p_1p_2 . (O'Rourke, 1998)

1.5.1 Algoritmos existentes

Existen varios algoritmos que se emplean en la creación de una triangulación de Delaunay a partir de un conjunto de puntos. El de construcción incremental consiste en añadir un vértice a una triangulación de Delaunay y corregir la red hasta que todos los triángulos cumplan de nuevo la condición de Delaunay. Divide y vencerás, consiste en dividir el conjunto de puntos en dos partes de igual tamaño, calcular la triangulación de Delaunay para cada parte individualmente y después reunir las dos triangulaciones corrigiendo los errores. Por último, está el algoritmo Recorre la línea el cual se basa en un principio similar al de construcción incremental, construir una pequeña parte de la triangulación final y después seguir añadiendo vértices hasta que la triangulación esté completa. La diferencia estriba en que no hay que corregir ninguno de los errores que pudieran presentarse.

Teniendo en cuenta la simplicidad del algoritmo construcción incremental (fig6) frente al resto de los algoritmos, se definió su utilización para llevar a cabo la generación de una triangulación de Delaunay a partir de un conjunto de puntos. Permitiendo definir el modelo de la información geográfica en 3D a representar.

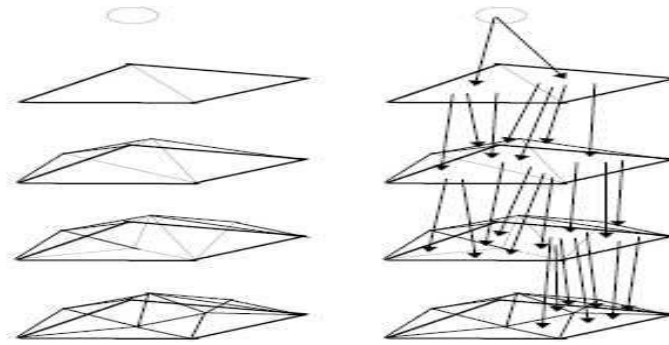


fig. 6 Algoritmo de Construcción Incremental

Algoritmo construcción incremental

Un aspecto importante de este algoritmo es que hay que inspeccionar si un vértice está dentro de una circunferencia circunscrita o no. Es posible computar el circuncentro y la circunferencia circunscrita y después examinar si el vértice está dentro del círculo, pero existe una prueba más simple y eficiente que utiliza el determinante de una matriz.

Si los tres puntos A, B y C forman un triángulo con los puntos denominados en sentido contrario al de las manecillas del reloj, el punto D está dentro de su circunferencia circunscrita si el determinante de la matriz que se forma a partir de dichos puntos es mayor que cero. Ver fig7

$$\begin{vmatrix} A_x & A_y & A_x^2 + A_y^2 & 1 \\ B_x & B_y & B_x^2 + B_y^2 & 1 \\ C_x & C_y & C_x^2 + C_y^2 & 1 \\ D_x & D_y & D_x^2 + D_y^2 & 1 \end{vmatrix} = \begin{vmatrix} A_x - D_x & A_y - D_y & (A_x - D_x)^2 + (A_y - D_y)^2 \\ B_x - D_x & B_y - D_y & (B_x - D_x)^2 + (B_y - D_y)^2 \\ C_x - D_x & C_y - D_y & (C_x - D_x)^2 + (C_y - D_y)^2 \end{vmatrix} > 0$$

fig. 7 Test de pertenencia

Es decir, si el determinante de esta matriz es mayor que cero los triángulos ABC y BCD no cumplen la condición de Delaunay. Por lo que la arista BC es ilegal y debe ser sustituida por la arista AD. Esta acción

CAPÍTULO 1: Los SIG y la representación de la información geográfica en 3D

es posible implementarla mediante el algoritmo computacional *Legalizar_Aristas*, el cual se muestra a continuación.

```
Algoritmo Legalizar_Aristas ( $p_r, p_i p_j$ )  
  Si  $p_i p_j$  no es una arista legal Entonces  
    Sea  $p_i p_j p_k$  el triángulo adyacente a  $p_i p_j$   
    Reemplazar  $p_i p_j$  con  $p_r p_k$   
    Legalizar_Aristas ( $p_r, p_j p_k$ )  
    Legalizar_Aristas ( $p_r, p_k p_i$ )  
  Fin_Si  
Fin_ Legalizar_Aristas
```

Finalmente, el algoritmo de construcción incremental plantea que, sea P una nube de puntos de los cuales se quiere obtener una triangulación de Delaunay. Estos serán encerrados en un gran triángulo que los contenga a todos. El punto p_r del conjunto P será elegido de forma aleatoria, con este y los tres primeros se genera una nueva triangulación de Delaunay (fig8). Luego se elige el próximo punto que pertenece a una de las aristas o triángulos ya generados, se transforma la red invocando al algoritmo *Legalizar_Aristas* y se obtiene una nueva triangulación, el proceso es repetido hasta incluir todos los puntos del conjunto (fig9). Cuando todos los puntos sean añadidos se eliminan los tres primeros. (O'Rourke, 1998)

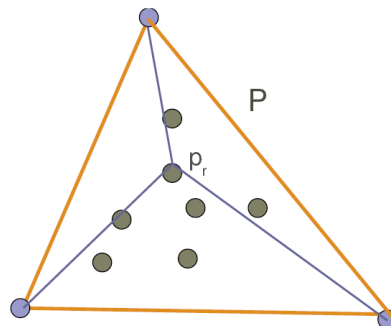


fig. 8 Construcción incremental

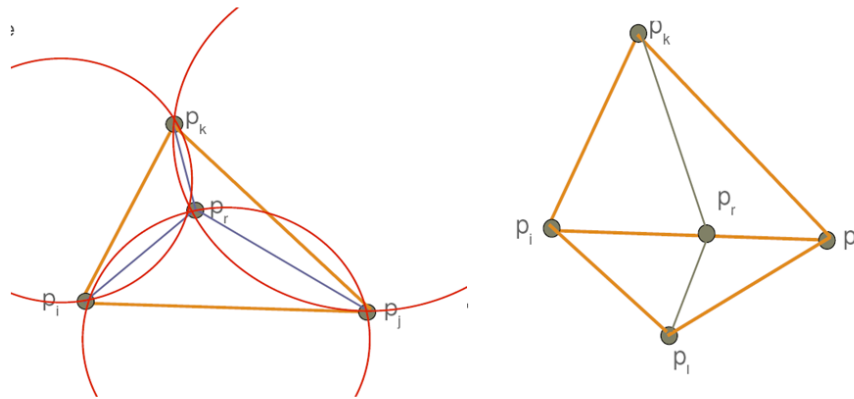


fig. 9 Construcción incremental

Este proceso de triangulación es posible implementarlo mediante el algoritmo computacional **Construcción_Incremental** el cual se describe a continuación.

Algoritmo **Construcción_Incremental** (T, P, n): T

Introducir la nube de puntos P de tamaño n en un gran triángulo conteniendo a dicha nube de puntos

Para cada uno de los puntos de P , elegir p_r de forma aleatoria y triangular:

Encontrar el triángulo $p_i p_j p_k$ conteniendo a p_r

Si p_r cae en el interior del triángulo $p_i p_j p_k$ **Entonces**

Añadir aristas desde p_r hasta los vértices p_i, p_j, p_k

Legalizar_Aristas ($p_r, p_i p_j$)

Legalizar_Aristas ($p_r, p_j p_k$)

Legalizar_Aristas ($p_r, p_k p_i$)

Fin_Si

Sino

p_r cae en la arista $p_i p_j$ que es incidente a los triángulos $p_i p_j p_k$ y $p_i p_j p_l$

Legalizar_Aristas ($p_r, p_i p_l$)

Legalizar_Aristas ($p_r, p_l p_j$)

Legalizar_Aristas ($p_r, p_j p_k$)

Legalizar_Aristas ($p_r, p_k p_i$)

Fin_Sino

Finalmente eliminar los tres puntos del gran triángulo inicial.

Fin_Construcción_Incremental

1.6 Método de generación de sombreado

El problema básico del procedimiento convencional de sombreado consiste en realizar la iluminación desde una fuente de luz única y direccional siguiendo el método básico propuesto por Horn y Sjöberg, donde el brillo de cada punto del terreno depende del coseno del ángulo de incidencia de la luz sobre él.

$$I = k \cos i.$$

El ángulo de incidencia i se estima en función de las coordenadas angulares del Sol, acimut (θ) y elevación sobre el horizonte (γ). El modelo de reflectancia se expresa como MDR (θ, γ). Este método genera sombreados “duros” donde las zonas auto ocultas ($i > 90^\circ$) aparecen en color negro. Además, se destacan las estructuras perpendiculares al vector de iluminación pero las paralelas suelen quedar poco visibles. (Cuartero, 2008)

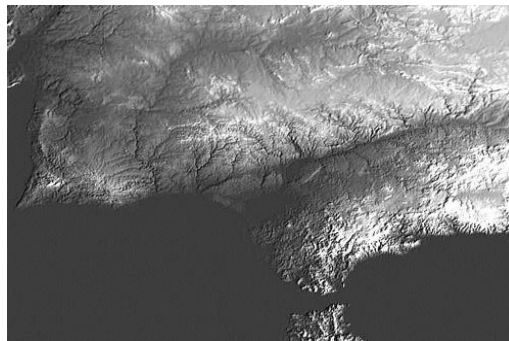


fig. 10 Generación de sombreado

1.7 Método de tintas hipsométricas

Es un método de representación cartográfica del relieve donde el color de un punto determinado depende de la altura del mismo. Cuando se utilizan tintas hipsométricas se selecciona una serie graduada de tonos de color para colorear áreas dentro de unos intervalos de altitudes determinados. Por ejemplo, los terrenos con una altitud entre 0 y 100 metros pueden colorearse con un tono verde suave, los terrenos con

CAPÍTULO 1: Los SIG y la representación de la información geográfica en 3D

una mayor altitud, comprendida entre 100 y 200 metros, con un verde más oscuro y los que están entre 200 y 300 metros con un marrón claro. Para la asignación de colores en dependencia de la altura normalmente se utilizan códigos cromáticos convencionales es decir, arbitrariamente fijados y aceptados en el ámbito. (Cuartero, 2008)

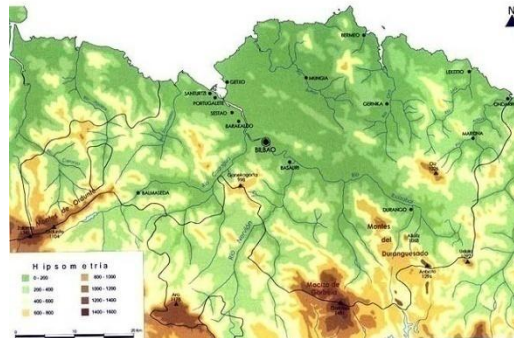


fig. 11Tintas hipsométricas

1.8 Proceso de creación de una extensión para QGIS

Una extensión es una aplicación que se relaciona con otra para aportarle una función nueva y generalmente específica. Esta aplicación adicional es ejecutada por la aplicación principal e interactúan por medio de la API. También se lo conoce como plug-in (del inglés "enchufable"), add-on (agregado), complemento, conector. Los complementos permiten:

- que los desarrolladores externos colaboren con la aplicación principal extendiendo sus funciones.
- reducir el tamaño de la aplicación.
- separar el código fuente de la aplicación a causa de la incompatibilidad de las licencias de software.

La aplicación principal proporciona servicios que el complemento puede utilizar en forma de un objeto de tipo, incluyendo un método para que los complementos se registren a sí mismos y un protocolo para el intercambio de datos. Los complementos dependen de los servicios prestados por la aplicación de acogida y no suelen funcionar por sí mismos. Por el contrario, la aplicación principal funciona

CAPÍTULO 1: Los SIG y la representación de la información geográfica en 3D

independientemente de ellos, lo que permite a los usuarios finales añadir y actualizar los complementos de forma dinámica sin necesidad de hacer cambios a la aplicación principal. (Alegsa.com.ar)

Las interfaces de programación de aplicaciones (APIs) proporcionan una interfaz estándar, lo que permite a terceros crear complementos que interactúan con la aplicación principal. Un API estable permite que complementos de terceros funcionen como la versión original y amplíen el ciclo de vida de las aplicaciones obsoletas (Alegsa.com.ar).

QGIS se ha diseñado con una arquitectura de complementos. Esto permite que se añadan nuevas funciones a la aplicación. Muchas de las funciones actuales de QGIS están en realidad implementadas como complementos. Hay dos tipos de complementos en QGIS: integrados o aportados por usuarios. Un complemento integrado es mantenido por el equipo de desarrollo de QGIS y forma parte de cada distribución de QGIS. Un complemento aportado por usuarios es un complemento externo que es mantenido por el autor individual.

Cuando QGIS arranca escanea ciertos directorios en busca de complementos tanto de C++ como de Python. Para que un archivo (biblioteca compartida, DLL o script de Python) sea reconocido como complemento tiene que tener una firma específica. Para los scripts de Python es bastante sencillo. QGIS busca en las siguientes localizaciones dentro del directorio de instalación:

- Linux y otros Unix: `./share/qgis/python/plugins`
- Mac OS X: `./Contents/MacOS/share/qgis/python/plugins`
- Windows: `.\share\QGIS\python\plugins`

Cada complemento de Python está contenido en su propio directorio. Cuando QGIS arranca busca en cada subdirectorio en `share/qgis/python/plugins` e inicializa cada complemento que encuentra. Una vez que esto está hecho, el complemento se mostrará en el administrador de complementos. Un complemento consta de la siguiente estructura básica:

`__init__.py`

recursos.py

recursos.qrc

complemento.py

La inicialización del complemento se hace en el script `__init__.py`. En varios sitios virtuales se pueden descargar script de ejemplos a los que solo es necesario modificarle las funcionalidades propias de la extensión y ubicarlos en el directorio correcto.

1.9 Análisis de soluciones existentes

En la actualidad son varios los SIG que permiten la visualización de la información geográfica en tres dimensiones. A continuación se destacan como principales SIG de escritorios a nivel mundial:

ArcView

ArcView es un sistema de información geográfica (SIG) que permite visualizar, gestionar, crear y analizar datos geográficos. Permite interactuar con los datos del mapa mediante la generación de informes y gráficos. Una de las funcionalidades que brinda el ArcView es la de procesamiento de datos geográficos en tres dimensiones pero el precio de su licencia es alto y tiene restricciones en su utilización. El carácter cerrado de su código no permite introducirle mejoras ni personalizaciones y afecta la conectividad del mismo. (Esri)

GRASS

GRASS (Geographical Resources Analysis Support System) 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

CAPÍTULO 1: Los SIG y la representación de la información geográfica en 3D

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. Brinda la opción de visualizar la información geográfica en 3D pero solo para formatos ráster y carece de interfaz amigable.

Google Earth

Es un software de uso gratuito para visualizar información e imágenes posicionadas geográficamente a través de la Internet. Permite visualizar imágenes en 3D del planeta, combinando imágenes de satélite, mapas y el motor de búsqueda de Google que posibilita ver imágenes a escala de un lugar específico del planeta. Todos los datos y las imágenes se encuentran almacenados en los servidores de Google, por lo que solo se descargan en una cache de la PC del visitante si su visualización es requerida. También es posible acceder a imágenes vectoriales con vista en tres dimensiones de las ciudades más importantes de Estados Unidos y Canadá. Brinda la opción obtener las coordenadas de cualquier punto, registrar anotaciones, establecer marcas y guardarlas, hacer búsquedas. Google Earth, cumple con las expectativas a la hora de visualización e interacción pero no permite trabajar con cartografías propias su código no es libre y adolece de las herramientas necesarias para un uso serio de análisis geográfico.

1.10 Conclusiones Parciales

Las soluciones existentes en la actualidad al problema de la modelación de superficies de terrenos como GRASS, Google Earth y ArcView no cuentan con una interfaz de usuario amigable, son de carácter propietario con licencias caras y restricciones de uso, o simplemente adolecen de herramientas para un uso serio de análisis geográfico. Por lo que se decide implementar una extensión para el QGIS que permita la representación de la información geográfica en 3D.

⁴ GPL: Licencia Pública General

CAPÍTULO 1: Los SIG y la representación de la información geográfica en 3D

Los SIG tienen como característica modelar los datos en dos tipos de formato ráster o vectorial. Se decidió centrar la solución en el formato vectorial que tiene como ventajas ser más preciso y ocupar menos espacio de memoria. Para la representación de la superficie se emplean las redes irregulares trianguladas. Éstas utilizan triángulos para representar una superficie lo que facilita su uso en las aplicaciones gráficas. La triangulación de Delaunay es la más eficiente en este sentido al maximizar los ángulos mínimos de los triángulos reduciendo los errores de redondeo.

Teniendo en cuenta la precisión y el bajo consumo de memoria del formato vectorial se definió implementar la extensión para representar la información geográfica en este formato.

Para llevar a cabo la implementación de dicha extensión, se define la integración de los métodos sombreado y tintas hipsométricas, Redes irregulares trianguladas y la triangulación de Delaunay.

Capítulo 2: Tecnologías y herramientas a utilizar.

En el desarrollo de este capítulo se explica mediante una descripción breve y concisa el uso de las herramientas y tecnologías a utilizar para el desarrollo de la extensión que permita la representación de la información geográfica en 3D.

2.1 Programación Extrema (XP)

La Programación Extrema es una metodología ágil que imponen un proceso disciplinado sobre el desarrollo de software con el fin de hacerlo más predecible y eficiente. Se basa en una retroalimentación continuada entre el cliente y el equipo de desarrollo, también busca la simplicidad en las soluciones implementadas y reduce al mínimo la documentación generada. Este tipo de programación es la adecuada para los proyectos con requisitos imprecisos, cambiantes y con un riesgo técnico excesivo. (Flower, 2003) 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. El cliente de esta extensión es el departamento de Geoinformática y Señales digitales representado por los profesores del proyecto GeoQ con los que existe una interacción constante.
- **Simplicidad:** La sencillez es esencial para que todos puedan entender el código y se trata de mejorar mediante recodificaciones continuas. La aplicación se irá desarrollando interactivamente con entregas constantes del resultado esperado hasta ese momento.
- **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. Los clientes al estar en el mismo laboratorio de producción exponen cualquier duda, inconformidad y aceptación sobre la extensión.
- **Coraje:** Básicamente es trabajar muy duro durante las horas dedicadas a ello.

CAPÍTULO 2: Tecnologías y herramientas a utilizar

- **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 (Candeaux, 1994).

Ciclo de vida:

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 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.

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. (Candeaux, 1994)

CAPÍTULO 2: Tecnologías y herramientas a utilizar

Características:

- Desarrollo iterativo e incremental: pequeñas mejoras, una 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.
- 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 (Carrillo, 2005).

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 existe una sistemática comunicación entre el equipo de trabajo y el cliente. También era de interés del proyecto que la entrega de la extensión fuera en el menor tiempo posible.

2.2 Patrones GRASP

Los patrones GRASP son parejas de problema solución con un nombre, que codifican buenos principios y sugerencias relacionados frecuentemente con la asignación de responsabilidades. Describen los principios fundamentales de la asignación de responsabilidades a objetos, expresados en forma de patrones. (HERNÁN, 2005)

Durante el diseño se aplicarán estos patrones porque los mismos propician la modularidad y consistencia del sistema así como la mayor independencia entre clases. Se describen a continuación los que serán utilizados en esta investigación:

Patrón creador: Guía la asignación de responsabilidades relacionadas con la creación de objetos, tarea frecuente en los sistemas orientados a objetos. El propósito fundamental de este patrón es encontrar un creador que debemos conectar con el objeto producido en cualquier evento. Al escogerlo como creador, se da soporte al bajo acoplamiento. Lo que define este patrón es que una instancia de un objeto la tiene que crear el objeto que tiene la información para ello. ¿Qué significa esto?, pues que si un objeto A utiliza específicamente otro B, o si B forma parte de A, o si A almacena o contiene B, o si simplemente A tiene la información necesaria para crear B, entonces A es el perfecto creador de B. En la aplicación la clase Controladora crea instancias de las clases Triangulación, Grafico y Extractora las utiliza y tiene la información necesaria para ello. La clase Dialogo a su vez crea un objeto de tipo Controladora porque también lo necesita y tiene la información para ello. (HERNÁN, 2005).

Patrón Alta Cohesión: Mantiene la complejidad dentro de límites manejables, es decir, asigna una responsabilidad de modo que la cohesión siga siendo alta. La cohesión es una medida de cuán relacionadas y enfocadas están las responsabilidades de una clase. Una alta cohesión caracteriza a las clases con responsabilidades estrechamente relacionadas que no realicen un trabajo enorme. (HERNÁN, 2005)

Entre los beneficios que brinda destacan los siguientes: mejoran la claridad y facilidad con que se entiende el diseño, se simplifica el mantenimiento y las mejoras de funcionalidad. A menudo, se genera un

CAPÍTULO 2: Tecnologías y herramientas a utilizar

bajo acoplamiento y soporta mayor capacidad de reutilización. Todas las clases de la aplicación cumplen este principio teniendo pocos métodos bien definidos y estrechamente relacionados entre ellos.

Bajo Acoplamiento: Es la idea de tener las clases lo menos ligadas entre sí que se pueda. De tal forma que en caso de producirse una modificación en alguna de ellas, se tenga la mínima repercusión posible en el resto de las clases, potenciando la reutilización, y disminuyendo la dependencia entre las clases. El acoplamiento es una medida de la fuerza con que una clase está conectada a otras clases, con que las conoce y con que recurre a ellas. Acoplamiento bajo significa que una clase no depende de muchas clases. (HERNÁN, 2005). Entre los beneficios que brinda destacan los siguientes: no se afectan por cambios de otros componentes, son fáciles de entender y de reutilizar. En la aplicación este principio se hace evidente en las clases Grafico, Triangulación, Extractora que no depende de otra clase para funcionar.

Experto: Su objetivo es asignar una responsabilidad al experto en información, la clase que cuenta con la información necesaria para cumplir determinada responsabilidad. Distribuye el comportamiento entre las clases que cuentan con la información requerida.(HERNÁN, 2005)En el diseño de las clases se tuvo en cuenta este principio por ejemplo la clase Triangulación es la encargada de todos los cálculos matemáticos necesarios para obtener la triangulación y la clase Gráfico solo se ocupa de la parte de graficación en 3D.

Controlador:

Un evento del sistema de entrada es un evento generado por un actor externo. Un Controlador Es un objeto que no pertenece a la interfaz de usuario, responsable de recibir o manejar un evento del sistema. Un Controlador define el método para las operaciones del sistema. Normalmente, un controlador delega en otros objetos el trabajo que se necesita hacer; coordina o controla la actividad. No realiza mucho trabajo por sí mismo. La clase Controladora de la aplicación cumple con todas estas características.

2.3 El lenguaje de programación Python

Teniendo en cuenta que el lenguaje de programación Python proporciona las ventajas de paradigma orientado a objetos, que es un lenguaje multiplataforma, con una sintaxis muy limpia que favorece la legibilidad del código y que la herramienta QGIS brinda facilidades para el desarrollo de extensiones que utilizan dicho lenguaje se selecciona como el apropiado para la implementación de la extensión propuesta En conjunto con el PyQt. Esta es la versión de la librerías QT para Python y permite la creación de Interfaces gráficas de usuarios

2.4 Entorno de desarrollo Integrado (IDE) Eclipse.

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 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 2: 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. Una de las características más interesantes de Eclipse es que es completamente independiente de la plataforma y del idioma. Además, de la mezcla de lenguajes que soportada. (**Eclipse**)

2.5 OpenGL

OpenGL es una interfaz de software. Consiste en un conjunto de distintos comandos, que se utilizan para especificar los objetos y operaciones necesarias para producir aplicaciones gráficas interactivas y portables tanto en dos (2D) como en tres (3D) dimensiones (**Angel, 2002**). Desde que fue introducida en 1992 por Silicon Graphics, se ha convertido en la Interfaz de programación de aplicaciones (API) gráficas en 2D y 3D más usada en la industria. Es necesaria para la graficación en tres dimensiones del modelo generado por la extensión. Esta biblioteca se escoge por ser de código abierto y multiplataforma. Otras de sus características son

- **Evolución:** Debido a su minucioso diseño, la API de OpenGL permite asimilar rápidamente las innovaciones que se producen en el hardware a través de su mecanismo de extensión. Esto permite a los desarrolladores de aplicaciones y proveedores de hardware, incorporar las nuevas características a sus productos.

CAPÍTULO 2: Tecnologías y herramientas a utilizar

- **Confiable y portable:** Todas las aplicaciones desarrolladas en OpenGL, producen resultados de visualización consistentes en cualquier plataforma que soporte su API.
- **Escalable:** Las aplicaciones que utilizan el API de OpenGL, se pueden ejecutar en una amplia gama de sistemas, que va desde las computadoras personales y estaciones de trabajo, hasta supercomputadoras.
- **Fácil de usar:** Posee una gran cantidad de primitivas eficientes, que encapsulan información sobre el hardware subyacente, liberando al desarrollador de tener que diseñar teniendo en cuenta la plataforma.
- **Amplia documentación:** El hecho de que numerosos libros que han sido publicados sobre OpenGL sean accesibles libremente, además de una gran cantidad de código de ejemplos, hace que la información sobre la misma sea fácil de obtener.

2.6 GDAL

Geospatial Data Abstraction Library o GDAL es una biblioteca de software para la lectura y escritura de formatos de datos geoespaciales, publicada bajo la licencia X/MIT style Open Source por la fundación geoespacial de código abierto (Open Source Geospatial Foundation). Como biblioteca, presenta un único modelo abstracto de datos al uso que llama para todos los formatos soportados. También viene con una variedad de utilidades en línea de comando para la traducción y el proceso de datos geoespaciales. Esta librería permite el trabajo con los ficheros Shape el formato más utilizado por el QGIS y es necesaria para la obtención de los datos espaciales que se toman como punto de partida para el funcionamiento de la aplicación (**gdal**)

2.7 QGIS

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 Source forge en junio del mismo año. Actualmente QGIS⁵ corre en la mayoría de las plataformas como Unix, Windows y OSX. Esta desarrollado utilizando el Qt y como lenguaje C++ .Esto hace que QGIS sea rápido y tenga una interfaz amigable para los usuarios.

QGIS se ha diseñado con una arquitectura de complementos. El proceso de crearle una extensión a la herramienta QGIS es bastante sencillo en especial si se emplea Python. Muchas de las funcionalidades con las que cuenta en QGIS son complementos algunos son desarrollados y mantenidos por su equipo de desarrollo, pero la mayoría son desarrollados por programadores ajenos a la empresa. Por esta razón, además de todas las funcionalidades que brinda y ser de código abierto. Se definió como aplicación principal en la implementación de la extensión propuesta. Los elementos del núcleo son la base sobre la que se construyen los diferentes plugin.

Elementos del núcleo:

- Soporte ráster y vectorial.
- Soporte para PostgreSQL con tablas espaciales utilizado PostGIS.
- Integración con GRASS incluida visualización edición y análisis.
- Diseño de mapas.

⁵ QGIS: Quantum GIS, es un sistema de información geográfica de ambiente de escritorio.

CAPÍTULO 2: Tecnologías y herramientas a utilizar

- Soporte OGC⁶.
- Panel de vista general.
- Marcadores espaciales.
- Identificar y seleccionar elementos.
- Editar, visualizar y buscar atributos.
- Etiquetado de elementos.
- Proyecciones al vuelo.
- Guardar y recuperar proyectos.
- Exportar ficheros con extensión Mapfile a MapServer.
- Cambiar simbología vectorial y ráster.
- Arquitectura extensible con complementos (Qgis).

2.8 Conclusiones Parciales.

Durante el desarrollo de este capítulo se determinaron las metodologías y herramientas utilizadas en la implementación de la extensión. Como lenguaje de programación se emplea Python en conjunto con el PyQT para el trabajo con las interfaces gráficas de usuario pues la herramienta QGIS brinda facilidades para la integración de extensiones implementadas en este lenguaje. Como IDE el Eclipse un entorno de desarrollo multiplataforma y multilenguaje que utiliza la extensión PyDev para dar soporte dicho lenguaje. La modelación en tres dimensiones utiliza OpenGL y la interacción con los archivos Shape es a través de

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

CAPÍTULO 2: Tecnologías y herramientas a utilizar

GDAL ambas librerías fueron elegidas por su carácter libre. El reducido número del equipo de desarrollo y la demanda de una primera interacción en un corto periodo de tiempo, entre otros factores condicionan el uso de la metodología de desarrollo XP.

Capítulo 3: Presentación de la solución propuesta.

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.

3.1 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 usuario 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 (Carrillo, 2005).

3.1.1 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

CAPÍTULO 3: Presentación de la solución propuesta

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.(Carrillo, 2005).

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.(Eclipse)

A continuación se hace explicación y descripción de la plantilla de HU que se ha de utilizar 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	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)

CAPÍTULO 3: Presentación de la solución propuesta

(Alta / Media / Baja)	
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

Historias de Usuarios	
Número: 1	Nombre: Cargar archivo Shape
Usuario: Cliente	
Prioridad en Negocio: Alta	Riesgo en Desarrollo: Alto
Puntos de Estimación: 4	Iteración Asignada: 1
Descripción: Permite al usuario visualizar la información geográfica en tres dimensiones.	

CAPÍTULO 3: Presentación de la solución propuesta

Observaciones: La cartografía será un archivo de extensión Shape.

Tabla 2 Cargar Archivo Shape.

CAPÍTULO 3: Presentación de la solución propuesta

Historias de Usuarios	
Número: 2	Nombre: Mover vista.
Usuario: Cliente	
Prioridad en Negocio: Alta	Riesgo en Desarrollo: Alto
Puntos de Estimación: 3	Iteración Asignada: 2
Descripción: Permite al usuario una vez que se visualice el modelo en tres dimensiones mover la vista en diferentes direcciones.	

Tabla 3 Mover vista.

3.2 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 implementar las historias se establecen por 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 3: 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. (Carrillo, 2005).

3.2.1 Estimación de esfuerzos por historia de usuario

Para el buen desarrollo del sistema propuesto, se realizó una estimación para cada una de las historias de usuario identificadas, la estimación se hizo teniendo en cuenta el grado de complejidad de las diferentes historias de usuario así como las necesidades del cliente llegando a los resultados que se muestran a continuación.

Historias de Usuarios	Puntos de estimación
Cargar archivo Shape	5 semanas
Mover vista	3 semanas

Tabla 4 Estimación de esfuerzo por historia de usuario

3.2.2 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 se implementarán en cada iteración (para

CAPÍTULO 3: Presentación de la solución propuesta

maximizar el valor de negocio). Al final de la última iteración el sistema estará listo para entrar en producción. (Carrillo, 2005).

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 detallan 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 contará con la funcionalidad descrita en la HU # 1, la cual hace alusión a todo lo referente con cargar y visualizar archivos Shape.

Iteración 2:

La presente iteración tiene como objetivo desarrollar la HU#2. Una vez concluida esta iteración se contará con la funcionalidad descrita en la HU # 2, la cual hace alusión a todo lo referente con mover la vista.

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 (Carrillo, 2005). Este plan tiene como finalidad mostrar la duración de cada iteración, así como el orden en que serán implementadas las HU en cada una de ellas.

Iteración	Historias de Usuario	Duración total de Iteraciones
Iteración 1	Cargar Shape.	5 semanas
Iteración 2	Mover vistas.	3 semanas

Tabla 5 Plan de duración de las iteraciones.

CAPÍTULO 3: Presentación de la solución propuesta

3.4 Conclusiones Parciales.

En este capítulo se abordaron las dos primeras fases del ciclo de vida de la metodología XP, exploración y planificación. Se define el alcance del proyecto y la descripción de las historias de usuario. Estas sirven de guía pues realizan 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. La fase de planificación, demostró que el desarrollo del software tendrá 2 iteraciones concluyendo que el proyecto demorará 8 semanas proporcionando rapidez en la entrega del producto y de esta forma ofreciendo satisfacción al cliente. Del desarrollo de estas dos primeras fases depende todo el proceso de elaboración de la extensión para QGIS que permita la representación geográfica en 3D.

Capítulo 4: Construcción de la solución propuesta.

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.

4.1 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. (Carrillo, 2005)

4.1.1 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. (Carrillo, 2005)

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

Tarjetas CRC

CAPÍTULO 4: Construcción de la solución propuesta.

Clase: Nombre de la clase que se está modelando.	
Responsabilidades: Es una descripción de alto nivel del propósito de la clase.	Colaboraciones: Indica con que otras clases se relaciona.

Tabla 6 Plantilla para Tarjetas CRC.

Tarjetas CRC	
Clase: Triangulación	
Responsabilidades: __init__ Get Iniciar Orientación Pertenece Ubicación Crear Eliminar Adyacente1	Colaboraciones:

CAPÍTULO 4: Construcción de la solución propuesta.

Adyacente2	
Test	
No_Convexo	
Legalizar	
Limpiar	
Distancia	
Pertenece_Arista	
Arista	
Construcción_Incremental	

Tabla 7 Triangulación

Tarjetas CRC	
Clase: Extractora	
Responsabilidades:	Colaboraciones:
__init__	
Extraer	

Tabla 8 Extractora

CAPÍTULO 4: Construcción de la solución propuesta.

Tarjetas CRC	
Clase: Dialogo	
Responsabilidades: __init__ GetDir AbrirDiálogo	Colaboraciones:

Tabla 9 Dialogo

Tarjetas CRC	
Clase: Grafico	
Responsabilidades : __init__ Display Onsize Keyboard Draw_triangle Principal	Colaboraciones:

Tabla 10 Grafico

CAPÍTULO 4: Construcción de la solución propuesta.

Tarjetas CRC	
Clase: Controladora	
Responsabilidades : __init__ Dirección Principal	Colaboraciones: Triangulación Grafico Dialogo Extractora Extensión extensiónDialog __init__

Tabla 11 Controladora

4.2 Desarrollo de las iteraciones

En la segunda fase de la metodología XP, se describieron las HU correspondiente 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

CAPÍTULO 4: Construcción de la solución propuesta.

pueden ser escritas en lenguaje técnico debido a que son para uso estricto de los programadores, donde el usuario no tiene que necesariamente comprenderlas. (Carrillo, 2005)

Teniendo en cuenta la planificación realizada se llevó a cabo el desarrollo del sistema en dos 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.

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

Historias de Usuario	Tiempo de Implementación (semanas)	
	Estimación	Real
Cargar archivo Shape	5	5

Tabla 12 HU abordadas en la primera iteración.

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

CAPÍTULO 4: Construcción de la solución propuesta.

Tareas de la Ingeniería	
No. de la tarea: 1	No. de la HU: 1
Nombre de la tarea: Cargar Archivo Shape.	
Tipo de tarea: Desarrollo.	Puntos estimados: 5
Fecha inicio: 7/2/2011	Fecha inicio: 10/3/2011
Programador responsable: Pavel Raúl Romero.	
Descripción: Permite al usuario visualizar la información geográfica en tres dimensiones.	

Tabla 13 Tarea de la HU #1

4.2.2 Iteración 2:

La presente iteración tiene como objetivo desarrollar la HU#2. Una vez concluida esta iteración se contará con la funcionalidad descrita en la HU # 2, la cual hace alusión a todo lo referente con mover la vista sobre los tres ejes cartesianos.

CAPÍTULO 4: Construcción de la solución propuesta.

Historias de Usuario	Tiempo de Implementación (semanas)	
	Estimación	Real
Mover vista.	3	3

Tabla 14 HU abordadas en la segunda iteración

A continuación mediante tablas se evidencian 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: 2	No. de la HU: 2
Nombre de la tarea: Mover vista.	
Tipo de tarea: Desarrollo.	Puntos estimados: 3.
Fecha inicio: 14/3/2011	Fecha inicio: 15/4/2011
Programador responsable: Pavel Raúl Romero.	
Descripción: Permite al usuario una vez que se visualice el modelo en tres dimensiones mover la vista en diferentes direcciones. Los movimientos serán de rotación y traslación.	

Tabla 15 Tarea de la HU #2

4.3 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 y 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 y satisfagan las necesidades del cliente. (Carrillo, 2005).

4.3.1 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 respaldada 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. (Carrillo, 2005)

CAPÍTULO 4: Construcción de la solución propuesta.

4.3.2 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 y 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. (Carrillo, 2005).

Caso de prueba de aceptación	
Código: HU1_P1	No. de la HU: 1
Nombre: Comprobar que el mapa se cargue.	
Descripción: Evaluar que el archivo Shape se cargue.	
Condiciones de Ejecución: El cliente debe probar que él archivo Shape se cargue correctamente.	
Entrada/ Pasos de ejecución: Se solicita la opción cargar para probar que se carga el archivo Shape seleccionado.	
Resultado Esperado: Se cargó correctamente la cartografía.	
Evaluación de la Prueba: Prueba satisfactoria.	

CAPÍTULO 4: Construcción de la solución propuesta.

Tabla 16 Prueba de aceptación 1 para la HU1 Cargar archivo Shape.

Caso de prueba de aceptación	
Código: HU2_P1	No. de la HU: 2
Nombre: Comprobar que la vista se traslade hacia arriba.	
Descripción: Evaluar que una vez visualizada la cartografía el usuario pueda trasladarla hacia arriba	
Condiciones de Ejecución: El cliente debe probar que la vista se desplace correctamente.	
Entrada/ Pasos de ejecución: Se presiona la tecla 'w' y la vista se desplaza hacia arriba.	
Resultado Esperado: La imagen se desplazó hacia arriba.	
Evaluación de la Prueba: Prueba satisfactoria.	

Tabla 17 Prueba de aceptación 1 para la HU2 Mover Vista.

CAPÍTULO 4: Construcción de la solución propuesta.

Caso de prueba de aceptación	
Código: HU2_P2	No. de la HU: 2
Nombre: Comprobar que la vista se traslade hacia abajo.	
Descripción: Evaluar que una vez visualizada la cartografía el usuario pueda trasladarla hacia abajo.	
Condiciones de Ejecución: El cliente debe probar que la vista se desplace correctamente.	
Entrada/ Pasos de ejecución: Se presiona la tecla 's' y la vista se desplaza hacia abajo.	
Resultado Esperado: La imagen se desplazó hacia abajo.	
Evaluación de la Prueba: Prueba satisfactoria.	

Tabla 18 Prueba de aceptación 2 para la HU2 Mover Vista.

Caso de prueba de aceptación	
Código: HU2_P3	No. de la HU: 2
Nombre: Comprobar que la vista se traslade hacia la derecha.	
Descripción: Evaluar que una vez visualizada la cartografía el usuario pueda trasladarla hacia derecha.	

CAPÍTULO 4: Construcción de la solución propuesta.

Condiciones de Ejecución: El cliente debe probar que la vista se desplace correctamente.
Entrada/ Pasos de ejecución: Se presiona la tecla 'd' y la vista se desplaza hacia la derecha.
Resultado Esperado: La imagen se desplazó hacia la derecha.
Evaluación de la Prueba: Prueba satisfactoria.

Tabla 19 Prueba de aceptación 3 para la HU2 Mover Vista.

Caso de prueba de aceptación	
Código: HU2_P4	No. de la HU: 2
Nombre: Comprobar que la vista se traslade hacia la izquierda.	
Descripción: Evaluar que una vez visualizada la cartografía el usuario pueda trasladarla hacia izquierda.	
Condiciones de Ejecución: El cliente debe probar que la vista se desplace correctamente.	
Entrada/ Pasos de ejecución: Se presiona la tecla 'a' y la vista se desplaza hacia la izquierda.	
Resultado Esperado: La imagen se desplazó hacia la izquierda.	

CAPÍTULO 4: Construcción de la solución propuesta.

Evaluación de la Prueba: Prueba satisfactoria.

Tabla 20 Prueba de aceptación 4 para la HU2 Mover Vista.

Caso de prueba de aceptación	
Código: HU2_P5	No. de la HU: 2
Nombre: Comprobar que la imagen rote en el sentido de las manecillas del reloj sobre el eje z.	
Descripción: Evaluar que una vez visualizada la cartografía el usuario pueda rotarlas en sentido de las manecillas del reloj sobre el eje z.	
Condiciones de Ejecución: El cliente debe probar que la vista rota correctamente.	
Entrada/ Pasos de ejecución: Se presiona la tecla 'b' y la vista rota en sentido de las manecillas del reloj sobre el eje z	
Resultado Esperado: La imagen roto en el sentido de las manecillas del reloj.	
Evaluación de la Prueba: Prueba satisfactoria.	

Tabla 21 Prueba de aceptación 5 para la HU2 Mover Vista.

CAPÍTULO 4: Construcción de la solución propuesta.

Caso de prueba de aceptación	
Código: HU2_P6	No. de la HU: 2
Nombre: Comprobar que la rote en el sentido contrario de las manecillas del reloj sobre el eje z.	
Descripción: Evaluar que una vez visualizada la cartografía el usuario pueda rotarlas en sentido contrario de las manecillas del reloj sobre el eje z.	
Condiciones de Ejecución: El cliente debe probar que la vista rota correctamente.	
Entrada/ Pasos de ejecución: Se presiona la tecla 'n' y la vista rota en sentido contrario de las manecillas del reloj sobre el eje z	
Resultado Esperado: La imagen roto en el sentido contrario de las manecillas del reloj.	
Evaluación de la Prueba: Prueba satisfactoria.	

Tabla 22 Prueba de aceptación 6 para la HU2 Mover Vista.

4.4 Conclusiones parciales

Una vez finalizado el presente capítulo se obtuvo una extensión para la representación de la información geográfica con las características definidas en el capítulo anterior, al mismo se le realizaron pruebas de aceptación para comprobar su buen funcionamiento. Una vez realizadas las pruebas se arribó a la conclusión de que la extensión permite una representación más clara, precisa e interactiva de la información geográfica.

Conclusiones.

Conclusiones Generales

Como resultado de este trabajo de diploma se incorpora a la herramienta QGIS una extensión para la representación de la información geográfica en tres dimensiones. Permitiendo visualizar la información geográfica con mayor claridad y precisión. Tras la realización de las tareas de investigación científica, se ha arribado a las siguientes conclusiones.

- El lenguaje de programación Python, con Eclipse como IDE, PyQt en la creación de interfaces y OpenGL para graficar demostró ser la opción viable para la implementación de la extensión.
- La metodología desarrollo XP guía el proceso de elaboración de software desde los requerimientos del cliente capturados en las Historias de usuario hasta el producto terminado sin obstaculizarlo y en un breve espacio de tiempo.
- Las redes irregulares trianguladas en específico la triangulación de Delaunay son un modelo sencillo y cómodo que permite la visualización de superficies de manera rápida y eficiente.
- Se puede aprovechar la precisión y sencillez del modelo vectorial como fuente para la visualización de la información geográfica.
- Es posible la elaboración de una extensión para la representación de la información geográfica en 3D empleando tecnologías y herramientas distribuidas bajo licencias de software libre, logrando la libertad tecnológica del producto y cumpliendo con los lineamientos de producción de software de la Universidad de Las Ciencias Informáticas y el país, garantizando así la obtención de un producto software multiplataforma.
- Es posible incluir la extensión a todos los SIG desarrollados en el proyecto GeoQ que tienen como base el QGIS.

Recomendaciones

Recomendaciones

Luego de haber concluido la aplicación y cumplido los objetivos trazados, se plantean las siguientes recomendaciones:

- Continuar desarrollando funcionalidades para adicionárselas al sistema.
- Trabajar en mejoras en cuanto al diseño del sistema, teniendo en cuenta la opinión de los usuarios.

Referencias Bibliográficas

Referencias Bibliográficas

- *Eclipse*. (s.f.). Recuperado el 2 de 2 de 2011, de blog.elartedeprogramar.cl/page.
- *Alegsa.com.ar*. (s.f.). Obtenido de <http://www.alegsa.com.ar/Dic/extensi3n>
- Alzate, B. (2008). *Introduccion a los SIG*.
- Angel, E. (2002). *Interactive Computer Graphics: A Top-Down Approach*.
- Carrillo, I. P. (2005). *Metodologia de Desarrollo de software*.
- Cuartero, A. (2008). *Elaboraci3n de un mapa del relieve de Cantabria*.
- *Diccionario Babylon*. (2009). Recuperado el 30 de 10 de 2010
- *Esri*. (s.f.). Recuperado el 12 de 1 de 2011, de <http://mastergis.esri.es/productos.html#>.
- *gdal*. (s.f.). Recuperado el 4 de 2 de 2011, de <http://www.gdal.org/>
- Gonzalez, R. D. (2010). *Python para todos*.
- *Grass*. (s.f.). Recuperado el 29 de 11 de 2010, de www.scribd.com/doc/2096954/SIG-Libres.
- HERNÁN, V. a. (2005). *Fundamentos de Ingeniería de Software, Patrones de Diseño*.
- *mapa*. (s.f.). Recuperado el 15 de 10 de 2010, de <http://definicion.de/mapa/>.
- Martinto. (2009). *El diseño metodologico en la investigacion cientifica*. Ciudad Habana.
- O'Rourke, J. (1998). *Computational Geometry in C*.
- *PyQt*. (s.f.). Recuperado el 12 de 1 de 2011, de <http://www.slideshare.net/juancabicho/py-qt-desarrollando-aplicaciones-de-escritorio-el-co-di-go-k-presentation>.
- *Qgis*. (s.f.). Recuperado el 29 de 11 de 2010, de <http://www.qgis.org>.
- *Sig*. (s.f.). Recuperado el 16 de 10 de 2010, de http://app.jalisco.gob.mx/enlaceiit.nsf/Ed6_Posg?OpenPage
- *Sistema de informaci3n geogr3fica*. (s.f.). Recuperado el 5 de 11 de 2010, de http://www.mappinginteractivo.com/plantilla-ante.asp?id_articulo=1184

Glosario:

API: Una interfaz de programación de aplicaciones o API (del inglés application programming interface) es el conjunto de funciones y procedimientos (o métodos, en la programación orientada a objetos) que ofrece cierta biblioteca para ser utilizado por otro software como una capa de abstracción, usados generalmente en las bibliotecas. Uno de los principales propósitos de una API consiste en proporcionar un conjunto de funciones de uso general, de esta forma, los programadores se benefician de las ventajas de la API haciendo uso de su funcionalidad, evitándose el trabajo de programar todo desde el principio.

GPL: La Licencia Pública General de GNU o GNU GPL (del inglés GNU General Public License), es una licencia creada por la Free Software Foundation en 1989 y está orientada principalmente a proteger la libre distribución, modificación y uso de software. Su propósito es declarar que el software cubierto por esta licencia es software libre y protegerlo de intentos de apropiación que restrinjan esas libertades a los usuarios.

GUI: La interfaz gráfica de usuario o GUI (del inglés graphical user interface) actúa de interfaz de usuario, utilizando un conjunto de imágenes y objetos gráficos para representar la información y acciones disponibles en la misma. Su principal uso, consiste en proporcionar un entorno visual sencillo para permitir la comunicación con el sistema operativo de una máquina o computador.

IDE: Un Ambiente o Entorno de Desarrollo Integrado (“Integrated Development Enviroment” por sus siglas en inglés) es una herramienta que integra un conjunto de aplicaciones para crear software u otro tipo de contenidos digitales, ya sea imágenes, audios, videos, etc. En el desarrollo de software su principal objetivo es automatizar tareas, empaquetar, compilar, ejecutar y generalmente depurar el código, por mencionar algunos ejemplos. Por otro lado, en los IDEs para la creación de contenidos visuales se ofrece una serie de herramientas de dibujo, trazado, edición de propiedades, edición de color y otros.

Plugin: Es una aplicación informática que interactúa con otra aplicación para aportarle una función o utilidad específica. También conocido como addin, add-in, addon o add-on.

Glosario

SIG: Un Sistema de Información Geográfica es una integración organizada de hardware, software y datos geográficos diseñada para la gestión de datos geográficos.

TIC: Tecnologías de la informática y las comunicaciones, las redes mundiales de información están transformando al mundo y acercando más a la gente a través de la innovación de las comunicaciones mundiales, lo cual posibilita cambios en todos los ámbitos de la actividad humana. Con las nuevas tecnologías, el tiempo y la distancia dejan de ser obstáculos, los contenidos pueden dirigirse a una audiencia masiva o a un pequeño grupo y buscar un alcance mundial o meramente local. Las redes mundiales de información, como Internet no conoce fronteras.

TIN: Red Irregular Triangulada (Triangulated Irregular Network) se refiere a la estructura de datos usada en los Sistemas de Información Geográfica para representar una superficie, que se basa en nodos y líneas distribuidos irregularmente.