



Universidad de las Ciencias Informáticas.

Facultad VI

Título: Prototipo no funcional del plugin de Perfiles de Visibilidad y Altura para la plataforma uDig. Rol Analista.

Trabajo de Diploma para Optar por el Título de Ingeniero en Ciencias Informáticas.

Autor: Marisleidys Vázquez Sánchez

Tutor: Ing. Yampier Medina Trancón

Ciudad Habana, Junio del 2011

“Año 52 de la Revolución”



“Si los jóvenes fallan, todo fallará. Es mi más profunda convicción que la juventud cubana luchará por impedirlo. Creo en ustedes.”

A handwritten signature in black ink, which appears to be 'Fidel Castro'.

Declaración de Autoría

Declaración de Autoría

Declaro que soy la única autora del trabajo que lleva por título: Prototipo no funcional del plugin de Perfiles de Visibilidad y Altura para la plataforma uDig, y autorizó a la Universidad de las Ciencias Informáticas los derechos patrimoniales del mismo, con carácter exclusivo.

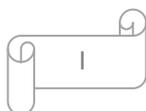
Para que así conste se firma la presente a los ____ días del mes de _____ del año 2011.

Marisleidys Vázquez Sánchez

Ing. Yampier Medina Tarancón

Firma del Autor

Firma del Tutor



Agradecimientos

Muchas han sido las personas que me han ayudado, de una u otra forma, para llegar hasta aquí, quiero que sepan que estaré eternamente agradecida, muy en especial:

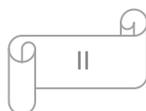
A Fidel y a la Revolución por darme la oportunidad de hacer realidad mis sueños de formarme como profesional.

A mi mamá y a Luis por ser como un padre para mí, no solo en estos años de Universidad sino durante toda mi vida, a ambos por haber estado conmigo en cada paso dándome fuerzas para continuar, por haber velado por mi bienestar, depositando su entera confianza en cada reto que se me presentaba, sin dudar ni un solo momento de mi capacidad, por haberme dado siempre su apoyo incondicional.

A mi tía Odalys por haber sido, más que mi tía, mi madre, mi amiga, mi confidente, aquella persona dispuesta ayudarme en todo momento, por fortalecer mi corazón e iluminar mi mente, con sus sabios consejos.

A Maidolis porque supo ganarse mi corazón y más que eso, mi confianza. A ti te doy una y mil veces gracias, por compartir momentos de tu tiempo conmigo, por estar incondicionalmente para mí en los buenos y malos momentos que tenemos que enfrentar en la vida, por darme tu cariño y tu amor incondicional, gracias por el cambio que provocaste en mí, aunque nos separemos siempre te recordaré porque vives y vivirás el resto de mi vida en mi mente y en mi corazón, porque a tu lado cada cosa mala se convertía en buena, la tristeza se transformaba en alegría y la soledad no existía, gracias por ser una hermanita tan bella y comprensiva. TE RE-QUIERO GRANDOTE.

A mi abuela Belkys por su preocupación, por su dedicación, amor y apoyo sin medidas.



Agradecimientos

A Israeldis por todo el amor y la ayuda que me brindó durante los años más difíciles de mi carrera.

A mis tías Cristina, Zoila, Virgen, Maritza, Catalina y Estelva por estar para mi cada vez que las necesité.

A Yosnai porque aunque se faje conmigo yo sé que muy, muy en el fondo me quiere, por prestarme la laptop y revisarme el documento cada vez que tenía corte de tesis.

Al Negro por ser mi protector y guía durante estos tres últimos años.

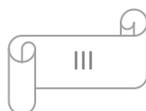
A mi tutor por todo el apoyo y la dedicación que me brindó durante el desarrollo de este trabajo.

A dos personas que aunque hoy no se encuentren formando parte de mi vida, en un momento determinado lo fueron y tengo mucho que agradecerles y en especial a una de ellas que aún sigue siendo, a pesar de todas las cosas, importante para mí: Viviana y Milayne.

Al tribunal por sus críticas constructivas para el correcto desarrollo de la investigación.

A todas mis compañeras y compañeros de aula: Sallet, Yula, Yugleinys, Saili, Ana Iris, Yanier, los Adrianes, Eyle, Yasel, en fin a todos.

Y por último y no menos importante a mi grupito de personas especiales: Yannier, Misbel, Ceci, Laimer, Liu, gracias a todos por estar para mí siempre que los necesité.



Dedicatoria

Dedicatoria

A mi mamá y a Luis por su amor incondicional.

A mi tía Odalys por ser mi otra mamá.

A lo mejor de mi vida, mi hermanita y amiga siempre, siempre Maidolis.

A Leannys, mi hermanita bella, que luche fuerte para que encuentre sus sueños, que nunca se dé por vencida y que sepa que siempre me tendrá aquí para ella.

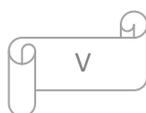
A mi papá porque a pesar de no ser la persona que yo siempre he esperado que sea, tengo la vaga esperanza que algún día cambiará.

Resumen

Un grupo de trabajo de la facultad 6 en conjunto con especialistas de GEOCUBA y el centro UCID (Unidad de compatibilización Integración de Desarrollo de Soluciones Informáticas para la Defensa) desarrollaron la plataforma GeneSIG, plataforma soberana para el desarrollo de Sistemas de Información Geográfica (SIG), la cual contiene más de 50 funcionalidades de SIG agrupadas en módulos, los cuales son la base para el desarrollo de personalizaciones de SIG para la informatización de la sociedad cubana, la sustitución de importaciones y el mercado internacional. Dentro de estos módulos se encuentra el Módulo de Análisis del Terreno donde los servicios de perfiles de visibilidad y altura que brinda son aún muy incipientes. Debido a esto, se propone realizar el análisis del prototipo no funcional del plugin de perfiles de visibilidad y altura para la plataforma uDig por ser una plataforma de código abierto que utiliza el entorno Java Eclipse para desarrollar soluciones orientadas a la visualización y edición de datos geográficos y alfanuméricos a través de Internet. Se utiliza como metodología de desarrollo de software RUP, como herramienta case Visual Paradigm 6.4 y como lenguaje de modelado UML 2.0. Además se hace un estudio de la factibilidad del sistema donde se determina la duración, el personal que lo desarrollará y los beneficios tangibles e intangibles que trae consigo el plugin. Los resultados obtenidos son avalados por las listas de chequeo, las revisiones, la matriz de trazabilidad y los prototipos.

Palabras Claves

Altura, Modelo Digital de Elevaciones, Modelo Digital del Terreno, Sistema de Información Geográfica, Visibilidad.



Índice

<i>Introducción</i>	1
<i>Capítulo 1. “Fundamentación Teórica”</i>	5
1.1. Introducción	5
1.2. Conceptos asociados al dominio del problema	5
1.2.1. Sistemas de Información Geográfica	5
1.2.1.1. Evolución de los SIG	7
1.2.1.2. Funcionamiento de los SIG	8
1.2.2. Modelos Digitales del Terreno	9
1.2.2.1. Aplicaciones de los MDT (8)	11
1.2.3. Modelo Digital de Elevaciones	12
1.2.3.1. Estructura de los MDE	13
1.2.4. uDig	17
1.3. Metodologías de Desarrollo de Software	19
1.3.1. Rational Unified Process (RUP)	19
1.3.2. Programación Extrema (Extreme Programming, XP)	21
1.4. Lenguaje Unificado de Modelado	22
1.5. Herramientas CASE (Computer Aided Software Engineering)	23
1.5.1. Visual Paradigm	24
1.5.2. Rational Rose	24
1.6. Herramienta de estimación	25
1.7. Técnica de los roles. Analista del Sistema	25
1.7.1. Analista del Sistema en RUP. (18)	26
1.8. Rol Desempeñado	27
1.9. Captura de Requerimientos	27
1.9.1. Introspección	28
1.10. Validación de los requerimientos	28
1.11. Conclusiones	28
<i>Capítulo 2. “Características del Sistema”</i>	29

2.1.	Introducción	29
2.2.	Modelo de Dominio	29
2.2.1.	Conceptos del Modelo de Dominio.....	30
2.3.	Modelo del Sistema.....	31
2.3.1.	Requisitos Funcionales.....	31
2.3.2.	Requisitos no Funcionales	33
2.3.3.	Descripción de los actores del sistema.....	34
2.3.4.	Diagrama de Casos de Uso del Sistema.....	34
2.3.5.	Descripción de los casos de uso del sistema	35
2.4	Conclusiones.....	46
	<i>Capítulo 3. “Estudio de factibilidad y Validación del sistema”</i>	<i>48</i>
3.1.	Introducción	48
3.2.	Estudio de factibilidad del sistema.....	48
3.3.	Planificación basada en Casos de Uso	49
3.3.1.	Paso 1: Cálculo de Casos de Uso Desajustados (24).....	49
3.3.2.	Paso 2: Cálculo de los puntos de casos de uso ajustados.....	51
3.3.3.	Paso 3: Estimación de esfuerzo a través de los puntos de casos de uso.	56
3.3.4.	Paso 4: Calcular el esfuerzo de todo el proyecto. (24).....	56
3.4.	Beneficios tangibles e intangibles.....	57
3.5.	Análisis Costo-Beneficio.....	58
3.6.	Métodos de Validación	58
3.6.1.	Método Delphi	58
3.6.2.	Test de Turing	59
3.6.3.	Comparación de los resultados de salida del modelo con los del sistema real. 60	
3.6.4.	Matriz de Trazabilidad.....	60
3.6.5.	Listas de Chequeo.....	62
3.6.6.	Revisiones	64
3.6.7.	Prototipos (28).....	64
3.7.	Conclusiones.....	66
	<i>Conclusiones Generales</i>	<i>67</i>

<i>Recomendaciones</i>	68
<i>Bibliografía Citada</i>	69
<i>Bibliografía</i>	71
<i>Glosario de Términos</i>	73

Índice de Tablas y Figuras

Figura 1. Esquema conceptual de un SIG(3).....	6
Figura 2. Tipos de Información Geográfica(2).....	9
Figura 3. Modelos y Estructuras de Datos.....	14
Figura 4. Modelo de Contornos (5)	16
Figura 5. Arquitectura de uDig (11).....	18
Figura 6. Metodología XP (14).....	21
Figura 7. Modelo de dominio	30
Figura 8. Diagrama de Casos de Uso del Sistema	34
Figura 9. Prototipo en el proceso de validación de requisitos.....	65
Tabla 1. Actividades y Responsabilidades del Analista del sistema en RUP.....	27
Tabla 2. Descripción de los actores del sistema	34
Tabla 3. Trazar perfiles de Altura	37
Tabla 4. Trazar Perfiles de Visibilidad.....	38
Tabla 5. Calcular Superficie	39
Tabla 6. Localizar por Coordenadas	41
Tabla 7. Obtener por Coordenadas.....	43
Tabla 8. Crear Ruta	44
Tabla 9. Visualizar Ruta.....	46
Tabla 10. Peso de Actores sin ajustar.....	50
Tabla 11. Peso de las Transacciones	51
Tabla 12. Peso de los factores de alta complejidad técnica	52
Tabla 13. Peso de los factores ambientales.....	55
Tabla 14. Distribución en porcentaje para el esfuerzo total del proyecto.....	57
Tabla 15. Matriz de Trazabilidad entre Requisitos Funcionales y Casos de Usos.....	61
Tabla 16. Lista de Chequeo “Requerimientos”	63
Tabla 17. Lista de Chequeo “Especificación de Requisitos”	64

Introducción

Desde hace algún tiempo los Sistemas de Información Geográfica (SIG) son herramientas muy efectivas en la ingeniería y las ciencias básicas, constituyendo un campo dinámico y novedoso. Se han convertido en una parte esencial de la sociedad moderna, al ser aplicados en diferentes sectores como: la edafología, la topología, la fotogrametría; en sistemas para la toma de decisiones, la gestión de recursos, la arqueología, la evaluación del impacto ambiental, la planificación urbana, la cartografía, la sociología, la geografía histórica, el marketing, la logística, la medicina, la agricultura, el transporte y la educación, así como, en estudios de impacto ambiental, planificación urbanística, viabilidad y utilización de los recursos naturales.

Los SIG operan con información geo-referenciada, manipulan, gestionan, analizan, almacenan y mapifican datos espaciales; permiten la captura, recuperación y representaciones de datos referenciados geográficamente.

Cuba se mantiene actualizada en cuanto a las aplicaciones de los SIG, a pesar de que su uso y comercialización está restringido a causa del bloqueo económico de Estados Unidos hacia la isla. Gracias a los especialistas y técnicos que trabajan en el desarrollo de estos sistemas, se ha incrementado su uso y los beneficios que traen consigo para la sociedad.

En la Universidad de las Ciencias Informáticas (UCI), el centro GEySED tiene una línea de desarrollo que se encarga de la construcción de tecnologías para el desarrollo de SIG de escritorios y para la web. Existe la necesidad de incorporar a los aplicativos SIG el módulo de Análisis de Terreno, el cual, además del cálculo de área, acimut y rutas permita el cálculo de perfiles de visibilidad y altura.

Para el desarrollo de aplicaciones de escritorio es necesario asimilar conocimientos y tecnologías que permitan recabar información sobre el funcionamiento, comportamiento y necesidades de los SIG de escritorio, en los cuales se incorpore el análisis de los perfiles de visibilidad y altura.

En el proyecto GeneSIG, perteneciente a este centro, se desarrolla la plataforma GeneSIG que cuenta con un módulo para el Análisis del Terreno donde los servicios de perfiles de visibilidad y altura que brinda son aún muy incipientes.

Durante el curso 2009-2010 se realizó una investigación que propone un conjunto de algoritmos a incorporar al módulo de Análisis del Terreno que representan una solución para los problemas de representación de los perfiles de visibilidad y altura sin que se hayan incorporado aún a ninguna tecnología.

La visibilidad es utilizada en todo lo que nos rodea, pues esta permite una visión clara de los objetos en el paisaje cotidiano. La posibilidad de ver objetos a niveles superiores depende, no solo del estado de la atmósfera, sino también de los objetos observados, su uso es muy frecuente en diferentes sectores como son: la geografía, la geología, la botánica, criterios de pendientes, criterios de distancias de carreteras, hidrología, usos forestales, distancias de núcleos urbanos, la aviación, la meteorología, etc.

La altura se utiliza de manera general en nuestro contorno, pues trata de representar las posiciones relativas de los objetos en un plano en tercera dimensión, es utilizada para representar mapas de tres dimensiones que resultan muy expresivos a la hora de representar variables continuas. La combinación de todas estas posibilidades determina que un mapa tridimensional se pueda dibujar de muchas formas distintas, cada una de ellas mostrando un aspecto diferente de la topografía. La altura se define como la distancia vertical de un cuerpo respecto al suelo u otra superficie tomada como referencia.

Se determinó que se debía asimilar la herramienta uDig como tecnología de escritorio a la cual se incorpore, de forma experimental a sus limitadas capacidades de análisis, los perfiles de visibilidad y altura. uDig (User Friendly Desktop Internet GIS¹) fue desarrollada en la primavera del 2004, tiene su origen en la empresa Refractions, es una aplicación geoespacial y una plataforma de código abierto con Licencia Publica General Reducida (LGPL) sobre la cual se pueden crear y modificar aplicaciones SIG de escritorio, desarrollada con la tecnología Eclipse Rich Client (RCP). Puede ser usada como una simple aplicación o puede ser extendida con plugins RCP. Su objetivo es proveer una

¹ GIS: *Geographic Information System*

solución completa en Java para acceso a datos, edición y visualización sobre SIG de escritorio.

Teniendo en cuenta lo expuesto anteriormente se define el siguiente **problema a resolver**: ¿Cómo contribuir al análisis del modelo digital del terreno para la plataforma uDig?

El **objeto de estudio** comprende el proceso de obtención de Perfiles de Visibilidad y Altura.

Para dar solución a la problemática descrita se ha planteado como **objetivo general**: desarrollar el prototipo no funcional del plugin de Perfiles de Visibilidad y Altura para la plataforma uDig.

Se define como **campo de acción** la modelación del prototipo no funcional del plugin de Perfiles de Altura y Visibilidad para la plataforma uDig.

El desarrollo de la investigación pretende corroborar la siguiente **idea a defender**: la elaboración de la documentación técnica asociada al prototipo no funcional del plugin de Perfiles de Visibilidad y Altura facilitará su incorporación a la plataforma uDig.

Con el fin de darle cumplimiento al objetivo general planteado se definen las siguientes **Tareas de la Investigación**:

- ✓ Realizar el diseño teórico y metodológico de la investigación.
- ✓ Describir los conceptos asociados a los Sistemas de Información Geográfica, Modelo Digital del Terreno y Modelo Digital de Elevaciones.
- ✓ Caracterizar los modelos de datos asociados a los MDE.
- ✓ Describir la arquitectura y los datos que soporta la plataforma uDig.
- ✓ Definir la metodología de desarrollo, el lenguaje de modelado y las herramientas CASE a emplear.
- ✓ Describir los requisitos que debe cumplir el plugin para su futura implementación.
- ✓ Desarrollar el modelo de casos de uso del sistema.
- ✓ Generar la documentación técnica relativa al desarrollo del prototipo no funcional del plugin de Perfiles de Visibilidad y Altura sobre uDig.

Posibles resultados:

- ✓ Documentación técnica de la investigación.
- ✓ Prototipo no funcional del plugin de Perfiles de Visibilidad y Altura.

Los **métodos científicos** empleados para la realización de este trabajo son:

Métodos Teóricos

Analítico-Sintético: Este método se usa para analizar la bibliografía encontrada referente al tema en cuestión y sintetizar los aspectos más importantes para la investigación.

Modelación: Servirá de apoyo a la hora de realizar los distintos diagramas que a lo largo de la investigación se realizaran.

Métodos Empíricos

Entrevista: Este método se emplea a través de la realización de entrevistas a diferentes profesores y estudiantes con conocimiento en el tema, con el objetivo de entender detalladamente cómo funcionan los perfiles de visibilidad y de altura.

El presente trabajo de diploma cuenta con 3 capítulos:

Capítulo 1 “Fundamentación Teórica”: En este capítulo se profundizan los conceptos asociados al dominio del problema.

Capítulo 2 “Características del Sistema”: En este capítulo se desarrolla el análisis del sistema. Se lleva a cabo un estudio para capturar los requisitos funcionales y no funcionales que debe cumplir el sistema, permitiendo la construcción del diagrama de casos de uso y la relación de los actores al interactuar con el sistema.

Capítulo 3 “Análisis de factibilidad y Validación del sistema”: En este capítulo se realiza un análisis de la factibilidad y se valida mediante técnicas y métodos el sistema propuesto para verificar si el plugin cumple todas las funcionalidades especificadas.

Capítulo 1. “Fundamentación Teórica”

Capítulo 1. “Fundamentación Teórica”

1.1. Introducción

En este capítulo se realiza la fundamentación teórica que sustenta la investigación, conteniendo los fundamentos necesarios para la comprensión de los temas tratados en el resto del documento. Además se hace un análisis detallado de las tecnologías y herramientas a utilizar.

1.2. Conceptos asociados al dominio del problema

1.2.1. Sistemas de Información Geográfica

Durante los últimos años los SIG se han convertido en una de las herramientas de trabajo más importantes para los investigadores, analistas y para el apoyo en la toma de decisiones, por lo que cuenta con un sinnúmero de definiciones.

Un SIG se define como un conjunto de métodos, herramientas y datos que están diseñados para actuar coordinada y lógicamente, para capturar, almacenar, analizar, transformar y presentar toda la información geográfica y la de sus atributos con el fin de satisfacer múltiples propósitos. Los SIG son una tecnología que permite gestionar y analizar la información espacial, y que surgió como resultado de la necesidad de disponer rápidamente de información para resolver problemas y contestar a preguntas de modo inmediato. (1)

Es creado en hardware, software y procedimientos elaborados para facilitar la obtención, gestión, manipulación, análisis, modelado, representación y salida de datos espacialmente referenciados para resolver problemas complejos de planificación y gestión. (2)

Son herramientas que permiten a los usuarios crear consultas interactivas, analizar la información espacial, editar datos, mapas y representar los resultados de todas estas operaciones. Es muy utilizada en muchos ámbitos, debido a que permite elaborar cartografía temática sobre cualquier aspecto ambiental, socioeconómico y políticos de la superficie terrestre. (3)

Capítulo 1. “Fundamentación Teórica”

La razón fundamental para utilizar un SIG es la gestión de información espacial. El sistema permite separar la información en diferentes capas temáticas y las almacena independientemente, permitiendo trabajar con ellas de manera rápida y sencilla, facilitando al profesional la posibilidad de relacionar la información existente a través de la topología de los objetos.

1.2.1.1. Evolución de los SIG

Los SIG tienen como origen la necesidad de la sociedad para poder manipular grandes volúmenes de datos e información. En sus inicios, las sociedades humanas hacían milagros para poder almacenar y manipular parte del conocimiento que se poseía acerca de la realidad geográfica. Así nacen los primeros elementos que guían los pasos de lo que tras años de avances es, hoy en día, un SIG moderno. (5)

En el siglo XX, donde ya existía la cartografía topográfica y temática, con la utilización de métodos cartográficos se logró que los mapas fueran separados en capas. A fines de 1960 surgieron sistemas que permitían construir las bases de datos con las figuras.

Como los SIG están fundamentados en la informática, han evolucionado por las fases típicas de este tipo de tecnología: (6)

Fase 1: Período de conceptualización. 1975-1985

El enfoque era netamente cartográfico y de naturaleza geográfica. El objetivo era determinar cómo llevar la creación de mapas al medio digital.

Fase 2: Período de implementación. 1985-1995

Sigue siendo un enfoque geocéntrico, pero surge la necesidad de integrar el aspecto sistemas de información con el aspecto geográfico (almacenar, recuperar, alterar, retroalimentar datos geográficos).

Fase 3: Período de maduración. 1995-1998

Capítulo 1. “Fundamentación Teórica”

Más aplicaciones. Mejoramiento de software, trabajos abiertos, interdisciplinarios más expansivos. Enfoque informático-céntrico. Se inicia una etapa comercial donde los SIG empezaron a difundirse al nivel del usuario doméstico debido a la generalización de los ordenadores personales.

Fase 4: Período de apertura. 1998

Estos sistemas llegan a un punto de apertura y expansión sin precedentes, gracias al mundo de Internet y su World Wide Web (WWW), que debido a las fuerzas de la tecnología informática, requieren sistemas abiertos, maniobrables y de integración. Los usuarios comenzaron a exportar el concepto de visualización de datos SIG a Internet.

Fase 5: Los SIG del siglo XXI

Se predice una integración de información geográfica mundial, acceso a datos espaciales interregionales mediante la operatividad virtual (Internet, comunicación satelital y de posicionamiento). Surge una expansión en el número de desarrollos de software SIG de código libre, los cuales, a diferencia del software comercial, suelen abarcar una gama más amplia de sistemas operativos, permitiendo ser modificados para llevar a cabo tareas específicas. Esta tecnología crece debido a la gran avalancha de productos que hicieron que se generalizara.

1.2.1.2. Funcionamiento de los SIG

Los SIG almacenan información sobre el mundo como una colección de niveles temáticos que pueden relacionarse por geografía. La información espacial contiene una referencia geográfica explícita como latitud y longitud o una referencia implícita (domicilio, código postal, nombre de una calle) que pueden ser derivadas de referencias explícitas mediante un proceso automatizado llamado geo-codificación.

Los Sistemas de Información Geográfica funcionan con dos tipos diferentes de información geográfica: el modelo ráster y modelo vectorial. (Figura 2).

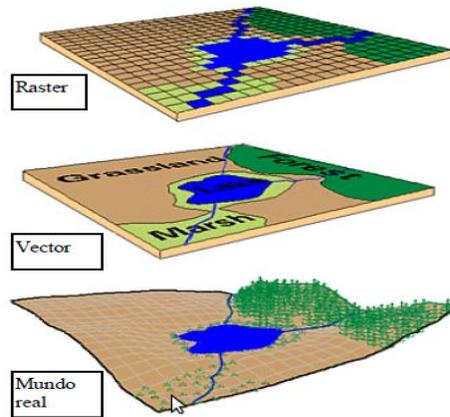


Figura 2. Tipos de Información Geográfica (2)

1.2.2. Modelos Digitales del Terreno

En la década de los años 50 tienen origen los Modelos Digitales del Terreno (MDT) y se establecen ya los primeros principios del uso de los modelos digitales para el tratamiento de problemas tecnológicos, científicos y militares. El objetivo de su creación fue agilizar mediante el tratamiento digital de datos del terreno adquiridos por fotogrametría el diseño de carreteras, planteándose una serie de algoritmos para la obtención de pendientes y áreas.

Los programas de uso más general para el tratamiento de los MDT se desarrollaron algunos años después. En 1967 surge y es presentado el SYMAP (Synagraphic Mapping Technique) que constituyó una de las primeras demostraciones de la posibilidad de manejo de la información espacial por medios informáticos. Este es un conjunto de programas de manejo de cartografía digital que incorporaba algoritmos de interpolación que permitían la generación de mapas de isolíneas a partir de puntos de altitud distribuidos irregularmente.

Los Modelos Digitales de Terreno son una parte elemental de la información de los Sistemas de Información Geográfica y uno de los elementos básicos de cualquier

Capítulo 1. “Fundamentación Teórica”

representación digital de la superficie terrestre. Constituyen la base para un gran número de aplicaciones en ciencias de la Tierra, ambientales e ingenierías de diverso tipo.

Pueden ser contruidos directamente a partir de la realidad a representar, pero es muy habitual que exista un modelo analógico intermedio a partir del cual se realiza la codificación. (7)

En su definición existen dos condiciones suplementarias: debe existir una estructura interna que represente las relaciones espaciales entre los datos y la variable representada en el modelo, debe ser cuantitativa y de distribución continúa.

Se puede definir un MDT como una estructura numérica de datos que representa la distribución espacial de una variable cuantitativa y continúa. (2)

El trabajo con MTD incluye las siguientes fases que no son necesariamente consecutivas en el tiempo:

- ✓ Generación del Modelo Digital de Elevaciones (MDE).
- ✓ Manipulación del MDE para obtener otras capas del MDT.
- ✓ Análisis del MDT.
- ✓ Visualización en dos dimensiones o mediante levantamientos 3D de todas las capas para localizar errores.
- ✓ Aplicación del MDT.

Los MDT están codificados en cifras y presentan algunas ventajas sobre otros tipos de modelos, derivadas de su naturaleza numérica:

- ✓ **No ambigüedad:** Cada elemento del modelo tiene propiedades, valores específicos y explícitos.
- ✓ **Verificabilidad:** Los resultados pueden ser analizados uno a uno y comprobados en todas las fases. Estos se construyen mediante pasos explícitos y concretos.
- ✓ **Repetitividad:** Los resultados no se someten a factores aleatorios o incontrolados a menos que estén expresamente diseñados. Así como pueden estar comprobados y replicados las veces que se desee.

Capítulo 1. “Fundamentación Teórica”

1.2.2.1. Aplicaciones de los MDT (8)

Apoyo en análisis estadísticos: Las variables incluidas en un MDT son factores de gran importancia en un gran número de procesos ambientales como: precipitación, insolación, temperatura, flujos hídricos, erosión, distribución de hábitats; por tanto van a ser un elemento clave a la hora de estimar otras variables mediante procedimientos de interpolación global por regresión.

Cálculo de magnitudes geométricas: Se pueden obtener los cálculos del volumen bajo la superficie tridimensional que constituye el MDT, cálculo del volumen entre dos superficies onduladas, una por ejemplo la del MDT real y la otra la que existiría en caso de llevar a cabo una obra, la construcción de una carretera; de este modo, es posible conocer el volumen de tierras que es preciso mover, tanto para vaciar porciones del terreno como para rellenar otras, en la realización de las diferentes alternativas de construcción existentes.

Modelos climáticos: La topografía es el principal factor local que limita la energía solar incidente sobre la superficie terrestre. La variedad de altitudes, pendientes y orientaciones crean fuertes contrastes locales que afectan directa e indirectamente a procesos biológicos y físicos.

Algunos de estos factores son modelables con los MDT. El ángulo solar resulta de utilidad en aplicaciones relacionadas con la teledetección (corrección por iluminación y cálculo de reflectividades). En cuanto a la radiación recibida se utiliza en:

- ✓ Modelos de estimación de variables climáticas como: temperatura y evapotranspiración.
- ✓ Modelos de distribución potencial de especies animales o vegetales.

Modelos hidrológicos: La superficie terrestre constituye la base sobre la que tienen lugar, y que por tanto condiciona, gran parte de los procesos de transferencia de materia y energía que tienen lugar sobre la superficie terrestre. La disponibilidad de un modelo de dicha superficie permite simular estos procesos, con lo que se consigue experimentar independientemente del sistema real.

Capítulo 1. “Fundamentación Teórica”

La simulación permite obviar los riesgos inherentes a la experimentación, alcanzar una completa independencia temporal, repetir el experimento en un número de veces arbitrario. La modelación hidrológica basada en modelos digitales de terreno pretende estimar los caudales generados en una cuenca a partir de sus características topográficas así como las áreas inundables en función de la altura esperada del nivel del agua.

Modelos de visibilidad: Los modelos de visibilidad establecen el área que se puede ver desde un punto y, por tanto, el área desde la que puede verse ese punto. Puede ser útil para el diseño de redes de control, por ejemplo: de incendios forestales. También como criterio a la hora de ubicar infraestructuras desagradables como vertederos, etc. El análisis de cuencas visuales puede utilizarse para la evaluación del impacto visual de actuaciones con efectos negativos sobre el paisaje.

Es posible construir un modelo de visibilidad, donde cada punto tiene asignado un valor proporcional a la extensión de su cuenca visual. Un modelo de este tipo puede servir de base objetiva para la toma de decisiones ya que permite conocer y comparar con fiabilidad la incidencia visual de las alternativas existentes.

1.2.3. Modelo Digital de Elevaciones

Los Modelos Digitales de Elevaciones (MDE) pueden ser tratados numéricamente de forma sencilla y derivar de ellos una gran cantidad de información tanto cuantitativa como cualitativa.

Un Modelo Digital de Elevaciones se define como una estructura numérica de datos que representa la distribución espacial de la altitud de la superficie del terreno. Consiste en una serie de puntos con coordenadas conocidas o referenciadas a un sistema de coordenadas bidimensionales a las que se les asocia un valor de elevación. (9)

Un terreno puede describirse de forma genérica como una función bivariable continua $z=Z(x, y)$ donde z representa la altitud del terreno en el punto de coordenadas (x, y) y Z es una función que relaciona la variable con su localización geográfica. La función anterior se aplica sobre un dominio espacial concreto D . En consecuencia, un MDE puede describirse genéricamente como $MDE = (D, Z)$. (8)

Capítulo 1. “Fundamentación Teórica”

Por este motivo, se han ensayado numerosas opciones en la búsqueda de una forma de representar y almacenar la altitud, donde se equilibre la pérdida de información y algunos efectos secundarios indeseables, como son el excesivo tamaño de los archivos o la dificultad de manejo. (8)

Mientras que los mapas convencionales usan casi exclusivamente una única convención (las curvas de nivel) para la representación de la superficie del terreno, los MDE disponen de alternativas más variadas, desde una transposición casi directa de las isohipsas hasta otras menos habituales en la cartografía impresa pero más adaptada al proceso digital. (2)

1.2.3.1. Estructura de los MDE

Históricamente las estructuras de datos en los Sistemas de Información Geográfica, y por extensión, en los modelos digitales del terreno se han dividido en dos grupos en función de la concepción básica de la representación de los datos.

Existen varias formas de representar un MDE de acuerdo a la estructura y organización de los datos:

- ✓ Vectorial: basado en entidades u objetos geométricos definidos por las coordenadas de sus nodos y vértices. (8)
- ✓ Ráster: basado en las localizaciones espaciales sobre una retícula regular de puntos a los cuales se les asigna el valor de elevación. (8)

Cada uno de estos modelos de datos puede expresarse mediante diferentes estructuras de datos dentro de los dos modelos básicos. (Figura 3).

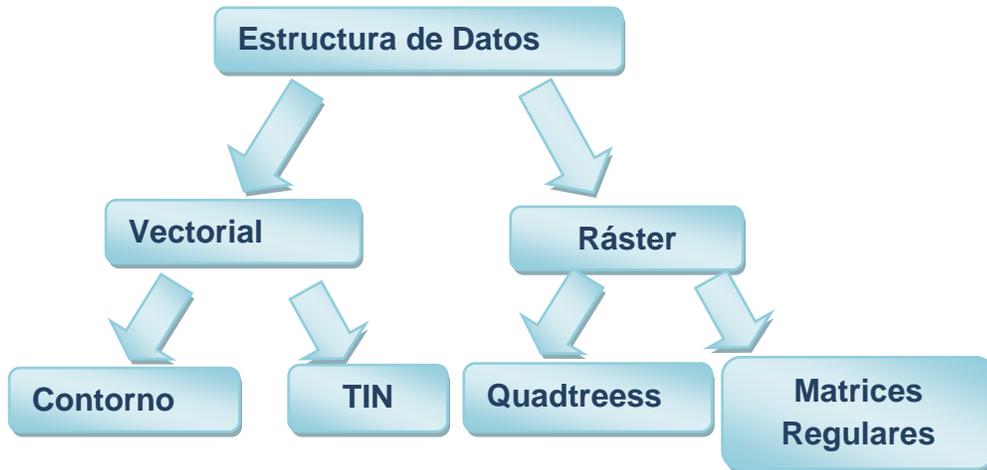


Figura 3. Modelos y Estructuras de Datos

Modelo Vectorial

Los modelos vectoriales están basados básicamente en puntos y líneas definidas por sus coordenadas.

La información gráfica en este tipo de formatos se representa internamente por medio de rectas o vectores. De este modo un mapa queda reducido a una serie de pares ordenados de coordenadas, utilizados para representar puntos, líneas y superficies. La captura de la información en el formato vectorial se hace por medio de: mesas digitalizadoras, convertidores de formato ráster a formato vectorial, GPS (Sistema de Geoposicionamiento Global), entrada de datos alfanumérica, entre otros. (5)

En este modelo se manejan dos tipos de datos:

- ✓ **Datos alfanuméricos:** Son descripciones de las características de las entidades gráficas. Generalmente son almacenados en formatos convencionales para este tipo de información.
- ✓ **Datos gráficos:** Son descripciones digitales de las entidades del plano. Suelen incluir las coordenadas, reglas y símbolos que definen los elementos cartográficos en un

Capítulo 1. “Fundamentación Teórica”

mapa. Los SIG utilizan esos datos para generar un mapa o una representación gráfica en la pantalla de un ordenador o sobre papel.

Para la representación de datos gráficos se utilizan tres tipos básicos de entidades:

- ✓ **Puntos o Nodos:** Son objetos sin dimensiones que se utilizan para las entidades geográficas que mejor pueden ser expresadas por un único punto de referencia.
- ✓ **Líneas:** Son objetos de una dimensión definidos por un nodo inicio y un nodo fin.
- ✓ **Polígonos:** Son objetos limitados y continuos de dos dimensiones.

Las dos estructuras vectoriales más empleadas son las curvas de nivel o estructuras de contorno y la red irregular de triángulos (TIN por sus siglas en ingles).

Red irregular de triángulos (TIN): esta estructura de datos es la que mejor representa las superficies continuas, en especial los terrenos. Es adaptable a formas topográficas de variada complejidad ya que la distribución y densidad de los puntos originales reflejan implícitamente las irregularidades de la superficie del terreno. (10)

Se compone de un conjunto de triángulos los cuales se construyen ajustando un plano a tres puntos cercanos no colineales, y se adosan sobre el terreno formando un mosaico que puede adaptarse a la superficie con diferente grado de detalle, en función de la complejidad del relieve. Se trata de una estructura en la que el terreno queda representado por el conjunto de superficies planas que se ajustan a un conjunto previo de puntos. (8)

La red irregular de triángulos puede considerarse como una estructura derivada de puntos o líneas. Aunque la distribución original puede ser cualquiera, incluso puntos distribuidos aleatoriamente, es frecuente partir de una base de isohipsas para generar la red de triángulos. Estos están internamente organizados en función de su vecindad mediante un conjunto de información bastante compleja que hace posible un manejo relativamente ágil y eficaz frente a alternativas menos estructuradas. (2)

De manera general, los TIN cuentan con una gran capacidad para describir superficies en diferentes niveles de resolución y gran eficiencia en el almacenamiento de los datos.

Capítulo 1. “Fundamentación Teórica”

Curvas de nivel o Contornos: es la más común, puede ser utilizada para representar el relieve de la superficie terrestre, la cual se suele representar mediante líneas que unen puntos situados a la misma altitud y que se trazan generalmente con un intervalo determinado y equidistante para todo el terreno a modelar o cartografiar. (10)

La estructura básica de un modelo de contornos es la polilínea definida como un vector de n pares de coordenadas (x, y) que describe la trayectoria de las curvas de nivel o isohipsas. (Figura 4).



Figura 4. Modelo de Contornos (5)

Modelo Ráster

El modelo ráster es el más adecuado para la integración de las elevaciones en un SIG debido a que permite la utilización de diversas herramientas para la obtención de nuevos mapas a partir del MDE.

El modelo ráster se obtiene cuando se digitaliza un mapa, una fotografía o cuando se obtienen imágenes digitales capturadas por satélites. En ambos casos se obtiene un archivo digital con esta información. La captura de la información en este formato se hace mediante: scanner, imágenes de satélite, fotografía aérea, cámaras de video, entre otros. (5)

La localización en los modelos ráster puede ser absoluta y relativa: la absoluta viene definida por el sistema de coordenadas y la relativa se define a partir de la localización del píxel en relación al número de fila y columna.

El modelo ráster es muy utilizado en estudios que requieran la generación de capas continuas, necesarias en fenómenos no discretos; también en estudios medioambientales

Capítulo 1. “Fundamentación Teórica”

donde no se requiere una excesiva precisión espacial como: contaminación atmosférica, distribución de temperaturas, localización de especies marinas y análisis geológicos.

El modelo ráster cuenta con dos estructuras: Matrices Jerárquicas (Quadtreess) y Matrices Regulares.

Matrices Regulares

La estructura de las matrices regulares es el resultado de superponer una retícula sobre el terreno y extraer la altitud media de cada celda. La retícula adopta normalmente la forma de una red regular de malla cuadrada. En esta estructura, la localización espacial de cada dato está determinada de forma implícita por su situación en la matriz, una vez definidos el origen y el valor del intervalo entre filas y columnas. (7)

Matrices Jerárquicas (Quadtreess)

El interés de las matrices de resolución variable reside en la posibilidad de solucionar el principal problema de las matrices. En este tipo de matrices los elementos pueden ser datos elementales, submatrices con un nivel de resolución diferente. La estructura final es un árbol jerárquico y dinámico de submatrices con una profundidad en principio arbitraria y cuya resolución espacial se duplica en cada nivel. Se trata de una estructura análoga a los Quadtreess, utilizados ocasionalmente en el tratamiento de variables categoriales. (8)

1.2.4. uDig

uDig (User Friendly Desktop Internet GIS) es una plataforma de desarrollo de aplicaciones SIG de escritorio que utiliza el entorno Java Eclipse para el desarrollo de soluciones orientadas a la visualización y edición de datos geográficos y alfanuméricos a través de Internet. Su desarrollo comenzó en la primavera del 2004, tiene su origen en la empresa Refrations Research.

Su principal objetivo es ofrecer un cliente de escritorio que soporte el mayor número de fuentes de datos tanto locales como remotos.

Capítulo 1. “Fundamentación Teórica”

uDig utiliza Jump como base de algoritmos para el manejo y manipulación de datos espaciales y Geotools como librería para la entrada y salida de datos con lo que se asegura un buen número de formatos soportados.

Su arquitectura hace posible su utilización con las más diversas fuentes de datos: ESRI shp, ArcSDE, Oracle Spatial, PostGIS.

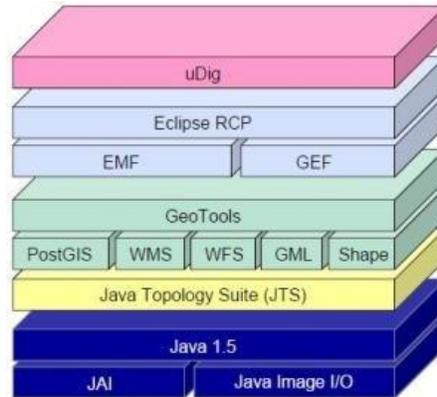


Figura 5. Arquitectura de uDig (11)

El punto más importante a destacar es que permite la conexión a servidores WFS (Web Feature Service) en modo lectura y escritura así como visor/editor de datos espaciales que hace especial énfasis en cumplir los estándares Open GIS para los servicios WMS (Web Map Service) y WFS. uDig proporciona una plataforma en Java común para desarrollar aplicaciones espaciales con componentes de código abierto.

Puntos fuertes: (11)

- ✓ Soporte del estándar WFS tanto en lectura como escritura.
- ✓ Soporte para servicios WMS.
- ✓ Soporte de acceso a todos los datos soportados por Geotools, tanto de archivos como de servidores.
- ✓ Capacidad de impresión y salidas gráficas en diversos formatos.
- ✓ Diseño modular orientado a la reutilización en otros proyecto o programas.
- ✓ En cuanto a bases de datos es robusto puede utilizar: ArcSDE, DB2, MySQL, Oracle Spatial, PostgreSQL/PostGIS y WFS.

Capítulo 1. “Fundamentación Teórica”

- ✓ La grilla, barra de escala y leyenda se integran como si fueran capas. Esto es interesante pues no son funcionalidades de la interface de despliegue sino de los datos.

Puntos débiles. (11)

- ✓ Actualmente no muy maduro en cuanto a opciones de visualización y edición.
- ✓ En cuanto a lectura de formatos vectoriales, es limitado con archivos discretos (no lee: dgn, kml, dxf, o dwg) pero sí los llamados: gml, XML. El único tradicional que lee es el shape file.
- ✓ Con imágenes ráster también tiene sus limitantes, pero se puede pegar a servicios wms y otros servicios en línea.

1.3. Metodologías de Desarrollo de Software

1.3.1. Rational Unified Process (RUP)

Cada día la producción del software busca adecuarse más a las necesidades de los usuarios, esto trae como consecuencia que aumente en tamaño y complejidad. Para lograr la productividad del software se necesita un proceso que integre las múltiples facetas del desarrollo del mismo.

RUP es una infraestructura flexible de desarrollo de software que proporciona prácticas recomendadas, probadas y una arquitectura configurable, es un proceso práctico de desarrollo de software para la Ingeniería Orientada a Objetos que proporciona un enfoque disciplinado para asignar tareas y responsabilidades dentro de una organización. Define, para cada etapa y disciplina, el flujo de trabajo, los trabajadores que intervienen, las actividades que realizan y los artefactos que se necesitan o producen. Su meta es

Capítulo 1. “Fundamentación Teórica”

asegurar la producción de software con la más alta calidad, que reúna las necesidades de los usuarios dentro del cronograma planeado y la inversión prevista. RUP deja bien definido y estructurado el proceso de diseño de un software. (12)

Según la metodología RUP, el ciclo de vida de un proyecto se divide en las siguientes fases:

- ✓ Inicio
- ✓ Elaboración
- ✓ Construcción
- ✓ Transición

En cada fase se ejecutaran una o varias iteraciones en número variable según el proyecto, y dentro de cada una de ellas se seguirá un modelo de cascada para los flujos de trabajo que requieren las nuevas actividades. (Figura 6)

RUP define nueve flujos de trabajo a realizar en cada fase del proyecto: (13)

- ✓ Modelado del negocio
- ✓ Análisis de requisitos
- ✓ Análisis y diseño
- ✓ Implementación
- ✓ Pruebas
- ✓ Distribución
- ✓ Gestión de configuración y cambios
- ✓ Gestión del proyecto
- ✓ Gestión del entorno

Mantiene una estructura bien definida del ciclo de vida de un proyecto y, de forma general, está caracterizado por los siguientes aspectos. (12)

- ✓ **Dirigido por casos de uso:** los casos de uso son el instrumento para validar la arquitectura del software y extraer los casos de prueba.
- ✓ **Centrado en la arquitectura:** los modelos son proyecciones del análisis y el diseño, constituye la arquitectura del producto que se desea desarrollar.

Capítulo 1. “Fundamentación Teórica”

- ✓ **Iterativo e incremental:** durante todo el proceso de desarrollo se producen versiones incrementales (que se acercan al producto terminado) del producto en desarrollo.

RUP se basa en casos de uso para describir lo que se espera del software y está orientado a la arquitectura del sistema, documentándose lo mejor posible, basándose un UML (Unified Modeling Language) como herramienta principal.

1.3.2. Programación Extrema (Extreme Programming, XP)

XP es una metodología ligera centrada en potenciar las relaciones interpersonales como clave para el éxito en el desarrollo de software, se basa en realimentación continua entre el cliente y el equipo de desarrollo, la comunicación fluida entre todos los participantes, la simplicidad en las soluciones implementadas y el coraje para enfrentar los cambios. Se define como metodología especialmente adecuada para proyectos con requisitos imprecisos y muy cambiantes, y donde existe un alto riesgo técnico. (Figura 7) (14)

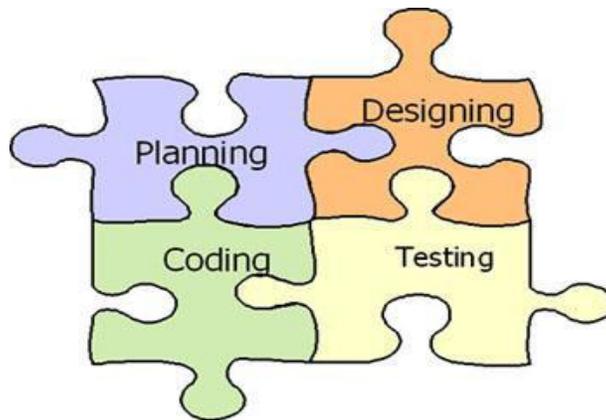


Figura 6. Metodología XP (14)

La metodología de XP se basa en:

Pruebas Unitarias: frecuentemente repetidas y automatizadas, incluyendo pruebas de regresión. Se basa en las pruebas realizadas a los principales procesos, de tal manera que adelantándonos en algo hacia el futuro, podamos hacer pruebas de las fallas que pudieran ocurrir.

Capítulo 1. “Fundamentación Teórica”

Re fabricación: consiste en la reutilización de código, para lo cual se crean patrones o modelos estándares, siendo más flexible al cambio.

Programación en pares: una particularidad de esta metodología es que propone la programación en pares, la cual consiste en que dos desarrolladores participen en un proyecto en una misma estación de trabajo. Cada miembro lleva a cabo la acción que el otro no está haciendo en ese momento. Es como el chofer y el copiloto: mientras uno conduce, el otro consulta el mapa.

Teniendo en cuenta que:

- ✓ RUP define roles fundamentales que tienen a su cargo la generación de artefactos y documentación por cada fase y flujo que tributan a un buen producto final.
- ✓ Es una metodología establecida y una de las más usadas mundialmente.
- ✓ Está respaldada por buenos resultados en proyectos en el mundo y en la UCI.
- ✓ Está diseñada de forma que existe un entendimiento continuo y gradual de todos los implicados en el proyecto.
- ✓ Centra su ciclo de vida en la arquitectura.
- ✓ Es iterativa e incremental.
- ✓ Esta dirigida por casos de uso.

Se determinó adoptar esta metodología para guiar el desarrollo del proyecto.

1.4. Lenguaje Unificado de Modelado

Lenguaje Unificado de Modelado es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad; está respaldado por el OMG (Object Management Group). Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema de software, ofrece un estándar para describir un "plano" del sistema (modelo), incluyendo aspectos conceptuales tales como procesos de negocios y funciones del sistema, aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y componentes de software reutilizables. (13)

Es el lenguaje en el que está descrito el modelo. Se puede aplicar en una gran variedad de formas para dar soporte a una metodología de desarrollo de software (tal como el

Capítulo 1. “Fundamentación Teórica”

Proceso Unificado Racional), pero no especifica en sí mismo qué metodología o proceso usar.

De manera general las principales características son: (15)

- ✓ Lenguaje unificado para la modelación de sistemas.
- ✓ Tecnología orientada a objetos.
- ✓ Corrección de errores viables en todas las etapas.
- ✓ Aplicable para tratar asuntos de escala inherentes a sistemas complejos de misión crítica, tiempo real y cliente/servidor.

UML no puede compararse con la programación estructurada, pues UML significa (Lengua de Modelación Unificada), no es programación, solo se diagrama la realidad de una utilización en un requerimiento. Mientras que, programación estructurada, es una forma de programar como lo es la orientación a objetos, sin embargo, la orientación a objetos viene siendo un complemento perfecto de UML, pero no por eso se toma UML sólo para lenguajes orientados a objetos.

1.5. Herramientas CASE (Computer Aided Software Engineering)

Existen herramientas Case de trabajos visuales que permiten realizar el modelado del desarrollo de los proyectos.

Las Herramientas CASE (Computer Aided Software Engineering, Ingeniería de Software Asistida por Ordenador) son las aplicaciones informáticas destinadas a aumentar la productividad en el desarrollo de software reduciendo el costo de las mismas.

Estas herramientas contribuyen de manera directa en todos los aspectos del ciclo de vida de desarrollo del software en tareas como el proceso de realizar un diseño del proyecto, cálculo de costes, implementación de parte del código automáticamente con el diseño dado, compilación automática, documentación o detección de errores entre otras.

Cada una de estas herramientas persigue nueve objetivos principales: (13)

- ✓ Mejorar la productividad en el desarrollo y mantenimiento del software.
- ✓ Aumentar la calidad del software.

Capítulo 1. “Fundamentación Teórica”

- ✓ Mejorar el tiempo y coste de desarrollo y mantenimiento de los sistemas informáticos.
- ✓ Mejorar la planificación de un proyecto
- ✓ Aumentar la biblioteca de conocimiento informático de una empresa ayudando a la búsqueda de soluciones para los requisitos.
- ✓ Automatizar, desarrollo del software, documentación, generación de código, pruebas de errores y gestión del proyecto.
- ✓ Ayuda a la reutilización del software, portabilidad y estandarización de la documentación
- ✓ Gestión global en todas las fases de desarrollo de software con una misma herramienta.
- ✓ Facilitar el uso de las distintas metodologías propias de la ingeniería del software.

1.5.1. Visual Paradigm

Visual Paradigm es una herramienta UML profesional que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. (16)

Esta herramienta se caracteriza por:

- ✓ Ofrece entorno de creación de diagramas para UML 2.0.
- ✓ Disponibilidad en múltiples plataformas.
- ✓ Disponibilidad de integrarse en los principales IDEs.
- ✓ Soporta una gama de lenguajes en la Generación de Código e Ingeniería Inversa en Java, C++, CORBA IDL, PHP, Esquema de XML, Ada y Python.
- ✓ Generación de código.
- ✓ Importación desde Rational Rose.

1.5.2. Rational Rose

Rational Rose es una herramienta con plataforma independiente que ayuda a la comunicación entre los miembros de equipo, a monitorear el tiempo de desarrollo y a entender el entorno de los sistemas. Una de las grandes ventajas de Rational Rose es

Capítulo 1. “Fundamentación Teórica”

que: utiliza la notación estándar en la arquitectura de software(UML), la cual permite a los arquitectos de software y desarrolladores visualizar el sistema completo utilizando un lenguaje común, además los diseñadores pueden modelar sus componentes e interfaces en forma individual y luego unirlos con otros componentes del proyecto. (15)

Rational Rose se caracteriza por:

- ✓ Ser una herramienta propietaria.
- ✓ Madura, bien establecida en el mercado.
- ✓ Incluye una Modelación Añadida de la Web que proporciona visualización, modelación y herramientas para el desarrollo de aplicaciones Web.
- ✓ Ofrece habilidades de análisis de calidad de código y generación del código, con capacidades de sincronización, así como manejo más granular y uso de modelos.

Por las características mencionadas, adicionadas a las ventajas que presenta en el diseño de sistemas y a la agilidad que proporciona en la modelación, se determinó emplear Visual Paradigm en la propuesta de solución.

1.6. Herramienta de estimación

Esta herramienta será utilizada para realizar los cálculos de factibilidad del sistema. ESTIMAC v1.0 permite automatizar los cálculos necesarios para realizar la estimación de un proyecto informático basándose en el Análisis de puntos de casos de uso. Fue realizado en la Universidad de las Ciencias Informáticas por el ingeniero Henry Raúl González Brito en el curso 2005-2006 como apoyo a la actividad docente de la asignatura Ingeniería de Software.

1.7. Técnica de los roles. Analista del Sistema.

Para llevar a cabo el perfeccionamiento de un software, debe tenerse en cuenta una metodología de desarrollo. La misma define dentro de ella una serie de roles que serán desempeñados por los miembros del equipo del proyecto.

Los roles son realizados típicamente por un individuo, o conjunto de individuos, trabajando juntos en equipo. Un miembro del equipo de proyecto cumple normalmente muchos roles. Los roles no son individuos; en lugar de ello, describe como los individuos se comportan en el negocio y que responsabilidades tienen. (17)

Capítulo 1. “Fundamentación Teórica”

Entre los principales roles a desempeñar durante el desarrollo de un proyecto, está el de Analista del Sistema. El mismo es el responsable de llevar a cabo el análisis del sistema, con el objetivo de comprender el problema a resolver, definir el alcance y brindar una visión más clara de lo que el sistema debe hacer, en cuanto a las necesidades requeridas por el cliente. Para desempeñar un buen rol, como analista del sistema se debe poseer un amplio rango de habilidades o niveles de competencias que serán mencionados a continuación: (18)

- ✓ Debe ser por encima de todo, un experto en la identificación y comprensión de problemas.
- ✓ Buen modelador y tener habilidades de comunicación.
- ✓ Tener dominio del proyecto y de las tecnologías.
- ✓ Capacidad de absorber y comprender rápidamente la información.
- ✓ Capaz de colaborar de forma efectiva con otros miembros del proyecto.

1.7.1. Analista del Sistema en RUP. (18)

La metodología de desarrollo RUP plantea que para una representación efectiva del software, debe existir un buen análisis y diseño de los requisitos funcionales y no funcionales que el sistema debe cumplir, por lo que el analista del sistema es el factor fundamental en el desarrollo del mismo.

Entre las actividades que realiza y responsabilidades que debe llevar a cabo el analista del sistema se encuentran:

Actividades	Responsabilidades
Buscar actores, trabajadores y casos de uso.	Atributos de requisitos.
Crear un vocabulario común.	Especificaciones suplementarias.
Definir el contexto del sistema.	Glosario.
Desarrollar el plan de gestión de requisitos.	Guion gráfico.
Desarrollar especificaciones suplementarias.	Modelos de Caso de uso.
Desarrollar la visión.	Plan de gestión de requisitos.

Capítulo 1. “Fundamentación Teórica”

Gestionar las dependencias.	Solicitudes de los interesados.
Obtener solicitudes del interesado.	Visión.

Tabla 1. Actividades y Responsabilidades del Analista del sistema en RUP.

1.8. Rol Desempeñado.

El rol desempeñado para el desarrollo de la investigación científica, asociado al tema de investigación del presente trabajo de diploma, es: el de Analista del Sistema. El cual obtendrá una serie de artefactos relacionados con el Objeto de Investigación.

Artefactos:

- ✓ Modelo de negocio o dominio (en este caso Dominio).
- ✓ Actores y casos de uso.
- ✓ Capturar, especificar y validar los requisitos funcionales y no funcionales del sistema.

1.9. Captura de Requerimientos

La captura de requisitos es la pieza fundamental de un proyecto marcando el punto de partida para las siguientes actividades del mismo, sirviendo de base para verificar si se alcanzaron los objetivos establecidos. Existe un factor que siempre ha acompañado a la captura de requisitos y es, a grandes rasgos, la carencia de comunicación entre los clientes y los desarrolladores. (19)

La Ingeniería de Requisitos ha trabajado arduamente para tratar de desarrollar técnicas que permitan hacer este proceso de una forma más eficiente y segura entre las que se hallan: Introspección, Entrevistas, Tormenta de Ideas, Sistemas Existentes, dentro de las entrevistas podemos encontrar:

- ✓ Entrevistas de Cuestionarios.
- ✓ Entrevistas de final abierto.
- ✓ Entrevistas en grupos de desarrollo.
- ✓ Discusiones.

De las técnicas anteriormente mencionadas se empleará la introspección.

Capítulo 1. “Fundamentación Teórica”

1.9.1. Introspección

Esta técnica recomienda que el ingeniero de requisitos se ponga en el lugar del cliente y trate de imaginar cómo desearía el sistema. En base a estas suposiciones comenzaría a recomendar al cliente sobre la funcionalidad que debería presentar el sistema. El problema radica en que un ingeniero no es un tipo normal de cliente, éste posee un conocimiento elevado, por lo que podría recomendar más de lo que el cliente necesita. (20)

1.10. Validación de los requerimientos

Los requisitos una vez definidos necesitan ser validados. La validación de requisitos tiene como misión demostrar que la definición de los requisitos define realmente el sistema que el usuario necesita o el cliente desea. Es necesario asegurar que el análisis realizado y los resultados obtenidos en la etapa de definición de requisitos son correctos. El proceso de validación tiene como finalidad comprobar que los requisitos de software poseen todos los atributos de calidad: son consistentes, completos, precisos, realistas, verificables, y definen lo que el usuario desea del producto final. (21)

Realizar la validación de los requerimientos es muy importante porque se detectarían posibles errores a tiempo, con esto se evita resultados inesperados que traerían como consecuencia gran pérdida de tiempo y costos excesivos. (20)

1.11. Conclusiones

En este capítulo se realiza un estudio detallado de definiciones de los conceptos básicos relacionados con el dominio del problema, contribuyendo al enriquecimiento del desarrollo de la investigación y logrando un mayor entendimiento de esta. De las metodologías de desarrollo existentes, se decide utilizar RUP, por las facilidades de detalles que brinda. Lo que constituye una excelente organización en el trabajo. Se hace una caracterización del rol de analista dentro de la metodología seleccionada. El lenguaje de modelado elegido es UML y la herramienta CASE, Visual Paradigm, por su peculiaridad de ser multiplataforma. Teniendo en cuenta el estudio realizado, se podría afirmar que la selección realizada es la óptima para efectuar un trabajo con calidad como se requiere.

Capítulo 2. “Características del Sistema”

2.1. Introducción

En el presente capítulo se realiza un análisis de la propuesta del sistema. Para lograr dicho objetivo se describen los conceptos más importantes del dominio, se especifican los requisitos funcionales y no funcionales, se identifican, mediante los diagramas de casos de uso del sistema, las relaciones entre los actores y casos de uso del sistema, así como las descripciones textuales de los casos de uso.

2.2. Modelo de Dominio

Haciendo un estudio profundo sobre la incorporación del prototipo no funcional del plugin de Perfiles de Visibilidad y Altura sobre la plataforma uDig, se llega a la conclusión que en la presente investigación no fue posible definir completamente todos los procesos, ni está claramente definida la frontera entre ellos, por lo que se decide dar un nuevo enfoque al mismo a través del modelo de dominio. Este modelo ayuda a identificar algunas clases que se utilizarán en el sistema, a comprender mejor la estructura y los conceptos necesarios en el sistema.

Un modelo de dominio captura los tipos más importantes de objetos que existen o los eventos que suceden en el entorno donde estará el sistema. (21)

A continuación se muestra el diagrama de clases del modelo de dominio:

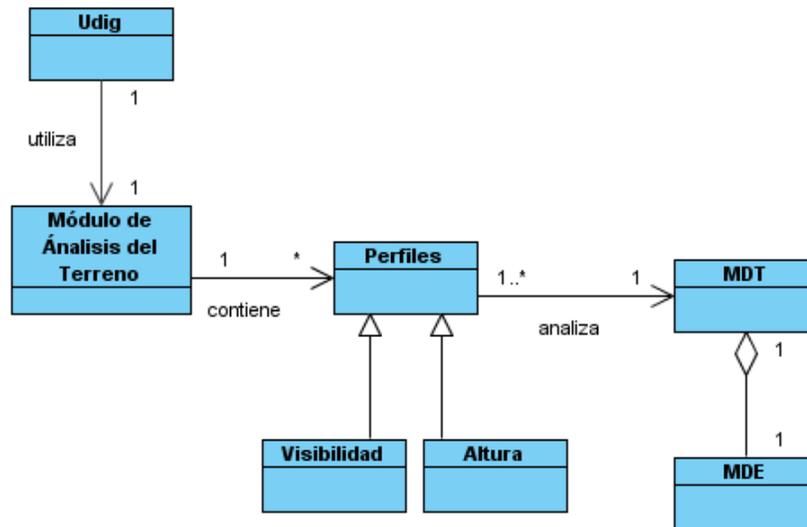


Figura 7. Modelo de dominio

2.2.1. Conceptos del Modelo de Dominio

Para un mayor entendimiento del modelo de dominio a continuación se explican todos los conceptos que intervienen en el mismo:

uDig : uDig (User Friendly Desktop Internet GIS) es una plataforma de desarrollo de aplicaciones SIG de escritorio que utiliza el entorno Java Eclipse para el desarrollo de soluciones orientadas a la visualización y edición de datos geográficos y alfanuméricos a través de Internet, es una plataforma de código abierto.

Módulo de Análisis del Terreno: En este módulo se realizan operaciones de análisis de la información espacial, como el cálculo de áreas, rutas, acimut, entre otras funciones de mayor complejidad. Los Perfiles de Visibilidad y Altura también se encuentran dentro de este módulo.

Perfiles: Los perfiles son una representación gráfica y en escala de toda o una parte de la superficie terrestre sobre un plano.

Visibilidad: La visibilidad es la cualidad perceptible que permite ver objetos a una determinada distancia.

Capítulo 2. “Características del Sistema”

Altura: Es la medida de un cuerpo o una figura según la consideración vertical desde su base hasta un punto más elevado.

Modelo Digital del Terreno: Un modelo digital del terreno no es más que una estructura numérica de datos que representa la distribución espacial de una variable cuantitativa y continúa.

Modelo Digital de Elevaciones: Se puede decir que un modelo digital de elevaciones es una estructura numérica de datos que representa la distribución espacial de la altitud de la superficie del terreno.

Breve Descripción del Modelo de Dominio

uDig es una herramienta de código abierto que utiliza un módulo de análisis del terreno para el cálculo de rutas, aéreas, acimut y perfiles. Para calcular estos perfiles es necesario analizar el Modelo Digital del Terreno, el cual genera, en una de sus fases, el Modelo Digital de Elevaciones. Los perfiles pueden ser de visibilidad y de altura.

2.3. Modelo del Sistema

Con el conocimiento de los objetivos que debe cumplir el prototipo no funcional del plugin de Perfiles de Visibilidad y Altura sobre la plataforma uDig, se pueden analizar cada una de las características que debe presentar el mismo, para ello se identifican los requisitos funcionales y no funcionales.

Los requisitos no son más que la condición o capacidad que tiene que ser alcanzada o poseída por un sistema o componente de un sistema para satisfacer un contrato, estándar u otro documento impuesto formalmente. (22)

2.3.1. Requisitos Funcionales

Los requisitos o requerimientos funcionales describen las capacidades o condiciones que el sistema debe cumplir, responden a: ¿Qué debe hacer el sistema? (22)

RF1: Trazar Perfiles de Altura.

RF 1.1: Seleccionar factor de altura.

RF 1.2: Seleccionar especulación de altura.

Capítulo 2. “Características del Sistema”

RF2: Trazar Perfiles de Visibilidad.

RF2.1: Calcular ángulo vertical que limita la exploración.

RF2.2: Determinar visibilidad a partir de un punto.

RF2.3: Determinar intervisibilidad entre dos puntos.

RF3: A partir del trazado de una región determinada el sistema deberá visualizar los valores de superficie de dicha región.

RF 3.1: Calcular área.

RF3.2: Calcular Perímetro.

RF3.3: Modificar unidad de medida.

RF4: El sistema debe ser capaz de localizar cualquier objeto del mapa de acuerdo a coordenadas proporcionadas por el usuario.

RF5: El sistema debe ser capaz de obtener coordenadas mediante el establecimiento de un punto en el mapa.

RF6: Crear Rutas.

6.1: Crear la ruta como una lista.

6.2: Crear la ruta abierta.

6.3: Crear la ruta cerrada.

6.4: Crear la ruta con diferentes orígenes y destinos.

6.5: Crear la ruta con el mismo origen y el mismo destino.

RF7: Visualizar Rutas.

RF7.1: Crear rutas.

RF7.2: Eliminar rutas.

RF7.3: Representar puntos.

Capítulo 2. “Características del Sistema”

RF7.4: Eliminar puntos.

RF7.5: Eliminar todo.

2.3.2. Requisitos no Funcionales

Los requisitos no funcionales son las cualidades o propiedades que el producto debe tener. Debe pensarse en estas propiedades como las características que hacen al producto atractivo, usable, rápido y confiable. Están vinculados normalmente a requisitos funcionales, es decir, una vez se conozca lo que el sistema debe hacer, podemos determinar cómo ha de comportarse, qué cualidades debe tener o cuán rápido o grande debe ser. (22)

Para el desarrollo del prototipo no funcional del plugin de Perfiles de Visibilidad y Altura sobre la plataforma uDig se han definido los siguientes requisitos no funcionales:

Usabilidad: El sistema podrá ser usado por personas con conocimientos básicos en informática y tendrá siempre visible la opción de ayuda, posibilitando un mejor aprovechamiento por parte de los usuarios de sus funcionalidades.

Eficiencia o Rendimiento: El tiempo de respuesta estará dado por la cantidad de información a procesar, entre más información, mayor será el tiempo de respuesta, la velocidad de procesamiento de la información, la actualización y la recuperación al igual que el tiempo de respuesta, dependerán de la cantidad de información que tenga que procesar la aplicación.

Restricciones de diseño: Diseño sencillo, con pocas entradas, donde no sea necesario entrenamiento para utilizar el sistema.

Interfaz de software: Sistemas Operativos: GNU/Linux, Windows y Mac OS.

Interfaz de comunicación: La herramienta uDig garantiza mediante su interfaz la configuración del entorno de trabajo, a través de funcionalidades propias para mostrar paneles, así como elementos para cambiar las vistas, las escalas y las capas que serán visibles en la interacción.

Capítulo 2. “Características del Sistema”

2.3.3. Descripción de los actores del sistema.

A continuación se dan a conocer los actores que intervienen en el sistema, teniendo en cuenta que estos actores no forman parte del sistema que se desarrolla, sino que son personas externas que interactúan con el mismo en pos de obtener un resultado.

Actores	Descripción
Usuario	Es cualquier persona con conocimientos básicos de informática y del funcionamiento de los SIG, encargada de seleccionar la operación que desea realizar.

Tabla 2. Descripción de los actores del sistema

2.3.4. Diagrama de Casos de Uso del Sistema

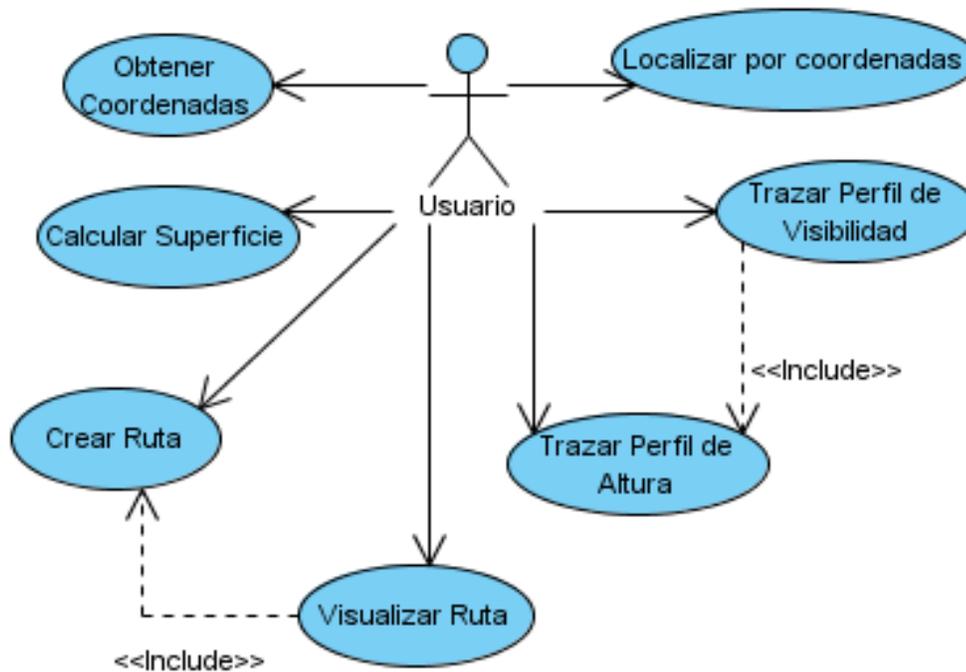


Figura 8. Diagrama de Casos de Uso del Sistema

Capítulo 2. “Características del Sistema”

2.3.5. Descripción de los casos de uso del sistema

Trazar Perfiles de Altura

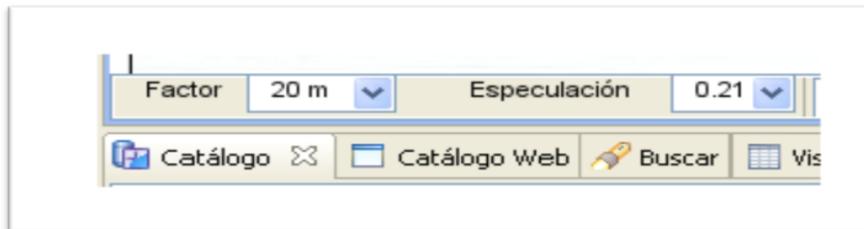
Caso de uso	Trazar Perfiles de Altura
Actores	Usuario
Propósito	Este caso de uso tiene como objetivo trazar perfiles en un mapa para obtener la altura asociada a una coordenada geográfica.
Resumen	El caso de uso es iniciado cuando el usuario desea obtener la altura asociada a una coordenada geográfica, trazando en el mapa los puntos hasta formar una polilínea, donde se determinará la altura y concluye cuando el sistema muestre el resultado.
Precondiciones	Tiene que estar seleccionada una capa.
Referencias	RF1, RF 1.1, RF 1.2
Prioridad	Crítico
Flujo normal de eventos	
Acción del actor	Respuesta del sistema
1. El caso de uso se inicia cuando el actor selecciona la opción Trazar Perfiles de Altura. Ver interfaz 1.	2. El sistema muestra la ventana de Información de Perfiles de Altura. Ver Interfaz 2.
3. El usuario hace clic izquierdo encima del mapa al que le quiere calcular el perfil trazando una polilínea y hace doble clic en el último punto de la misma. Ver Interfaz 3.	

Capítulo 2. "Características del Sistema"

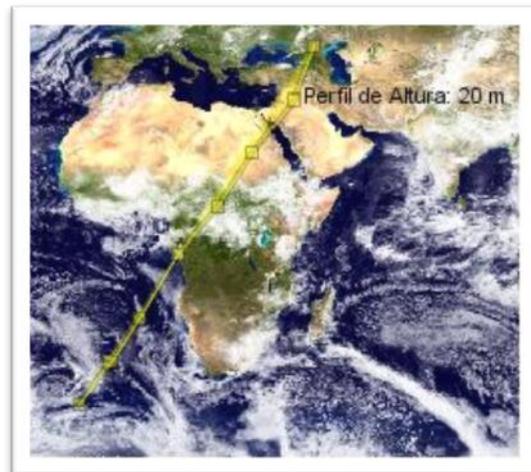
Interfaz 1



Interfaz 2



Interfaz 3



Sección Modificar Factor

Acción del Actor	Respuesta del sistema
4. El usuario modifica el valor del factor. Ver Interfaz 2.	5. El sistema actualiza el valor modificado y muestra la gráfica de altura actualizada.

Sección Modificar la Especulación de Altura

Acción del Actor	Respuesta del Sistema
------------------	-----------------------

Capítulo 2. “Características del Sistema”

6. El usuario modifica el valor de la especulación de altura. Ver Interfaz 2.	7. El sistema actualiza el valor modificado y muestra la gráfica actualizada.
Poscondiciones	Muestra los perfiles de altura correspondientes a la polilínea trazada por el usuario.

Tabla 3. Trazar perfiles de Altura

Trazar perfiles de Visibilidad

Caso de uso	Trazar Perfiles de Visibilidad
Actores	Usuario
Propósito	Este caso de uso tiene como objetivo obtener las superficies y puntos visibles desde una posición determinada en el mapa, que se registra a través de la distancia máxima visible sin interferencia de una altitud superior.
Resumen	El caso de uso es iniciado cuando el usuario desea obtener la visibilidad hacia un punto, y concluye cuando el sistema muestra la cobertura digital compuesta por polígonos en los que cada uno de ellos contiene información acerca de la superficie visible desde cada punto.
Casos de Uso Asociados	Trazar Perfiles de Altura.
Referencias	RF 2, RF 2.1, RF 2.2, RF 2.3
Prioridad	Crítico
Flujo normal de eventos	
Acción del actor	Respuesta del sistema
1. El caso de uso se inicia cuando el actor selecciona la opción Trazar Perfiles de Visibilidad. Ver interfaz 4.	2. El sistema muestra la ventana de Información de Perfiles de Visibilidad.

Capítulo 2. “Características del Sistema”

<p>3. El usuario selecciona la opción Determinar visibilidad a partir de un punto.</p> <p>4. El usuario define el punto desde el cual desea obtener las superficies y puntos visibles.</p>	<p>5. El sistema muestra una ventana con la información de todos los puntos y superficies visibles.</p>
Interfaz 4	
	
Flujos Alternos	
Acción del Actor	Respuesta del Sistema
<p>3. El usuario selecciona la opción Determinar intervisibilidad entre dos puntos.</p> <p>4. El usuario define el punto de inicio y el punto final entre los cuales desea obtener la superficie y los puntos visibles.</p>	<p>5. El sistema dibuja en el mapa los puntos y superficies visibles entre los dos puntos definidos por el usuario.</p>
Poscondiciones	<p>El sistema muestra los puntos y superficies visibles.</p>

Tabla 4. Trazar Perfiles de Visibilidad

Calcular Superficie

Caso de uso	Calcular Superficie
Actores	Usuario
Propósito	Este caso de uso tiene como objetivo formar una región para visualizar el cálculo del área y perímetro de la misma.
Resumen	El caso de uso se inicia cuando el usuario desea obtener el área y perímetro de una región trazada en el mapa, y termina cuando el sistema muestra los valores correspondientes.
Casos de Uso Asociados	

Capítulo 2. “Características del Sistema”

Referencias	RF 3, RF 3.1, RF 3.2, RF 3.3
Prioridad	Secundario
Flujo normal de eventos	
Acción del actor	Respuesta del sistema
1. El actor selecciona la opción Calcular Superficie. Ver Interfaz 5	2. El sistema muestra la ventana de Cálculo de Superficie.
3. El actor forma el área en el mapa y para cerrarla doble clic en el último vértice del área.	4. El sistema dibuja el polígono trazado por el usuario. Ver Interfaz 6
	5. El sistema muestra los valores asociados al área y el perímetro de la superficie. Ver Interfaz 7
Interfaz 5	
	
Interfaz 6	
	
Interfaz 7	
	
Poscondiciones	El sistema muestra los valores de área y perímetro de la superficie.

Tabla 5. Calcular Superficie

Localizar por Coordenadas

Caso de uso	Localizar por coordenadas.
--------------------	----------------------------

Capítulo 2. “Características del Sistema”

Actores	Usuario
Propósito	Este caso de uso se lleva a cabo con el objetivo de localizar cualquier objeto del mapa de acuerdo a las coordenadas (x, y) proporcionadas por el usuario.
Resumen	El caso de uso se inicia cuando el usuario desea localizar un objeto en el mapa por sus coordenadas, introducidas por él, y termina cuando el sistema muestra el objeto redimensionando.
Precondiciones	
Referencias	RF4
Prioridad	Secundario
Flujo normal de eventos	
Acción del actor	Respuesta del sistema
1. El caso de uso se inicia cuando el actor selecciona la opción Localizar por coordenadas. Ver Interfaz 8	2. El sistema muestra la ventana Localizar por Coordenadas.
3. El usuario introduce los datos que se le piden en la ventana y da clic en el botón Aceptar. Ver Interfaz 9	4. El sistema redimensiona el mapa moviendo hacia el centro de la pantalla las coordenadas introducidas por el usuario.

Capítulo 2. “Características del Sistema”

Flujos Alternos	
Acción del Actor	Respuesta del Sistema
3. El usuario da clic en el botón Cancelar.	4. El sistema abandona todas las operaciones realizadas y cierra la ventana.
Poscondiciones	El sistema visualiza en el centro del mapa las coordenadas introducidas por el usuario.

Tabla 6. Localizar por Coordenadas

Obtener por Coordenadas

Caso de uso	Obtener Coordenadas
Actores	Usuario
Propósito	Este caso de uso se lleva a cabo con el objetivo de localizar las coordenadas a partir de un punto definido por el usuario en el mapa.

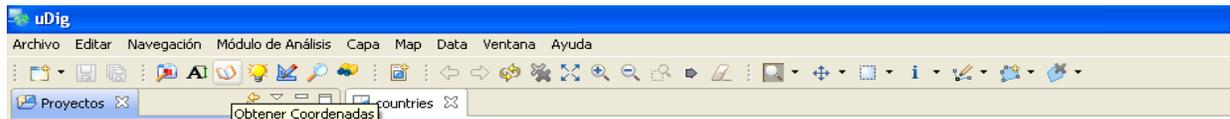
Capítulo 2. “Características del Sistema”

Resumen	El caso de uso se inicia cuando el usuario desea localizar las coordenadas de un punto definido por el mismo en el mapa y termina cuando el sistema muestra las coordenadas.
Precondiciones	-
Referencias	RF5
Prioridad	Secundario

Flujo normal de eventos

Acción del actor	Respuesta del sistema
1. El caso de uso se inicia cuando el actor selecciona la opción Obtener Coordenadas. Ver Interfaz 10.	2. El sistema muestra la ventana Obtener Coordenadas.
3. El usuario define el punto del cual desea obtener las coordenadas. Ver Interfaz 11.	4. El sistema muestra las coordenadas en las que se encuentra el punto definido por el usuario. Ver Interfaz 12.

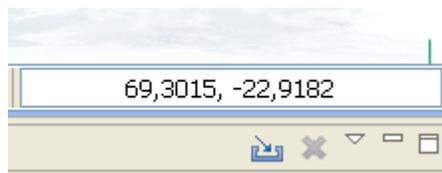
Interfaz 10



Interfaz 11



Interfaz 12



Capítulo 2. “Características del Sistema”

Poscondiciones	El sistema visualiza las coordenadas donde se encuentra el punto definido por el usuario.
-----------------------	---

Tabla 7. Obtener por Coordenadas

Crear Ruta

Caso de uso	Crear Ruta	
Actores	Usuario	
Propósito	Este caso de uso tiene como objetivo crear una ruta en dependencia de los datos proporcionados por el usuario.	
Resumen	El caso de uso se inicia cuando el usuario desea crear una ruta para posteriormente ser visualizada en el mapa y concluye cuando el sistema almacena los datos de dicha ruta.	
Precondiciones	-	
Referencias	RF6, RF6.1, RF6.2, RF6.3, RF6.4, RF6.5, RF6.6	
Prioridad	Secundario	
Flujo normal de eventos		
	Acción del actor	Respuesta del sistema
	1. El caso de uso se inicia cuando el actor selecciona la opción Crear Ruta. Ver Interfaz 13	2. El sistema muestra una ventana pidiéndole al usuario los datos asociados a la ruta como son: origen, destino y coordenadas.
	3. El usuario selecciona el botón Trazar puntos, que le permitirá colocar todos los puntos que desee en el mapa.	

Capítulo 2. “Características del Sistema”

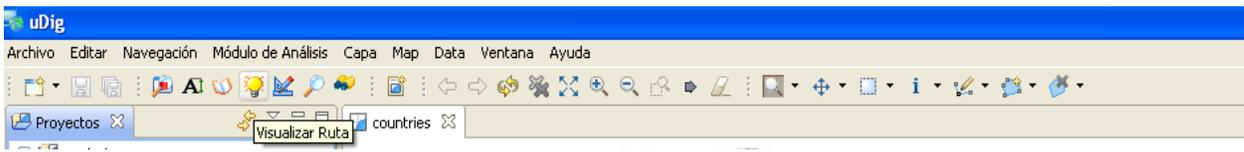
<p>4. El usuario selecciona todas las opciones que sean necesarias para crear una ruta y da clic en el botón Aceptar.</p> <p>5. El usuario da clic en el botón Crear Rutas.</p>	<p>6. El sistema almacena los datos asociados a la nueva ruta y muestra una ventana informándole al usuario que la ruta fue almacenada, si desea visualizarla, además de las opciones Sí y No.</p>
<p>7. El caso de uso concluye cuando el usuario da clic en No.</p>	
Interfaz 13	
	
Flujos Alternos	
Acción del actor	Respuesta del sistema
<p>7. El usuario de clic en Sí.</p>	<p>8. El caso de uso termina cuando el sistema activa Visualizar Ruta.</p>
Poscondiciones	<p>Se crea la ruta en dependencia de los datos introducidos por el usuario.</p>

Tabla 8. Crear Ruta

Visualizar Ruta

Caso de uso	Visualizar Ruta
Actores	Usuario
Propósito	Este caso de uso tiene como objetivo visualizar la ruta con la que el usuario desee trabajar.
Resumen	El caso de uso se inicia cuando el usuario desea visualizar una ruta que ya ha sido creada y concluye cuando el sistema dibuja la ruta en el mapa.
Casos de Uso Asociados	Crear Ruta.

Capítulo 2. “Características del Sistema”

Precondiciones	La ruta debe estar creada.
Referencias	RF7, RF7.1, RF7.2, RF7.3, RF7.4, RF7.5
Prioridad	Secundario
Flujo normal de eventos	
Acción del actor	Respuesta del sistema
1. El caso de uso se inicia cuando el usuario selecciona la opción Visualizar Rutas. Ver Interfaz 14.	2. El sistema muestra la ventana Visualizar Rutas.
3. El usuario selecciona la ruta con la que desea trabajar.	<p>4. El sistema visualiza los valores asociados a dicha ruta, como son: los puntos que contiene, el origen(es) y destinos.</p> <p>5. En dependencia de la acción que el usuario desee realizar sobre la ruta va a las siguientes secciones:</p> <ul style="list-style-type: none"> ✓ Trazar Rutas ir a la sección “Crear Rutas”. ✓ Eliminar Rutas ir a la sección “Eliminar Rutas”. ✓ Visualizar Puntos ir a la sección “Representar Puntos”. ✓ Eliminar Puntos ir a la sección “Eliminar Puntos”. ✓ Eliminar Todo ir a la sección “Eliminar Todo”.
Interfaz 14	
 <p>The screenshot shows the uDig application window with a menu bar (Archivo, Editar, Navegación, Módulo de Análisis, Capa, Map, Data, Ventana, Ayuda) and a toolbar. Below the toolbar, there are two tabs: 'Proyectos' and 'countries'. The 'Visualizar Ruta' window is active, displaying a map interface.</p>	
Sección “Crear Rutas”	
Acción del actor	Respuesta del sistema

Capítulo 2. “Características del Sistema”

1. El usuario hace clic en el botón Crear Rutas.	2. El caso de uso termina cuando el sistema visualiza en el mapa la ruta seleccionada por el usuario.
Sección “Eliminar Rutas”	
Acción del actor	Respuesta del sistema
1. El usuario hace clic en el botón Eliminar Rutas.	2. El caso de uso termina cuando el sistema borra la ruta dibujada en el mapa.
Sección “Representar Puntos”	
Acción del actor	Respuesta del sistema
1. El usuario hace clic en el botón Representar Puntos.	2. El caso de uso termina cuando el sistema representa en el mapa los puntos seleccionados por el usuario.
Sección “Eliminar Puntos”	
Acción del actor	Respuesta del sistema
1. El usuario hace clic en el botón Eliminar Puntos.	2. El caso de uso termina cuando el sistema elimina cualquiera de los puntos ya sea de origen o de destino o ambos y visualiza solo la ruta o la ruta con uno de los dos puntos.
Sección “Eliminar Todo”	
Acción del actor	Respuesta del sistema
1. El usuario hace clic en el botón Eliminar Todo.	2. El caso de uso termina cuando el sistema borra la ruta dibujada en el mapa y elimina todos los valores asociados a la misma.
Poscondiciones	El sistema muestra la ruta seleccionada.

Tabla 9. Visualizar Ruta

2.4 Conclusiones

En este capítulo se define la propuesta de solución del sistema, se presenta un listado de requisitos funcionales y no funcionales donde se recogen las principales necesidades de la herramienta uDig, las cuales fueron traducidas a un conjunto de Casos de Usos que representan las principales funcionalidades del sistema, además, se realizaron las

Capítulo 2. “Características del Sistema”

descripciones detalladas de cada uno de estos Casos de Uso donde se definen los prototipos de interfaz de usuario.

Capítulo 3. “Estudio de factibilidad y Validación del sistema”

3.1. Introducción

El análisis de factibilidad y planificación tiene como objetivo fundamental establecer planes razonables para desarrollar la Ingeniería de Software. Es de suma importancia la realización de este análisis ya que sirve de auxilio a las organizaciones para lograr sus metas y cubrir sus fines con los recursos actuales.

En este capítulo se describe la estimación de costos del plugin y los beneficios que trae consigo. Además se valida el sistema mediante las técnicas de validación: prototipo, matriz de trazabilidad y listas de chequeo.

3.2. Estudio de factibilidad del sistema

El estudio de factibilidad consiste en determinar cuáles son los objetivos del proyecto. Para ello se recopilan aquellos datos relevantes sobre el proyecto, específicamente sobre el plugin de perfiles de visibilidad y altura sobre la plataforma uDig, estimando el esfuerzo, los costos y el tiempo de duración del software.

La importancia de la estimación radica en el hecho de que en la medida en que esta sea realizada con mayor precisión y abarque detalladamente cada tarea realizada en el proyecto, más grande serán las posibilidades de que el tiempo y los costos globales del mismo se acerquen cada vez más a la realidad.

La factibilidad se apoya en 4 aspectos básicos:

- ✓ **Factibilidad Operativa:** Durante esta etapa se identifican todas aquellas actividades que son necesarias para lograr los objetivos planteados, se evalúa y determina todo lo necesario para llevarlas a cabo.
- ✓ **Factibilidad Técnica o Tecnológica:** Indica si se dispone de los conocimientos y habilidades en el manejo de métodos, procedimientos y funciones requeridas para

Capítulo 3. “Estudio de factibilidad y Validación del sistema”

el desarrollo e implantación del proyecto. Además indica si se dispone del equipo y herramientas para llevarlo a cabo, de no ser así, si existe la posibilidad de generarlos o crearlos en el tiempo requerido por el proyecto.

- ✓ **Factibilidad Económica:** Se refiere a que se dispone del capital en efectivo o de los créditos de financiamiento necesario para invertir en el desarrollo del proyecto.
- ✓ **Factibilidad de Tiempo:** En ella se verifica que se cumplan los plazos entre lo planeado y lo real, para poder llevar a cabo el software (23).

3.3. Planificación basada en Casos de Uso

La estimación mediante el análisis de Puntos de Casos de Uso es un método propuesto originalmente por Gustav Karner de Objectory AB, y posteriormente refinado por muchos otros autores. La planificación basada en casos de usos es un método de estimación del tiempo de desarrollo de un proyecto mediante la asignación de "pesos" a un cierto número de factores que lo afectan, para contabilizar el tiempo total estimado para el proyecto a partir de esos factores. (24)

Este método proporciona una estimación del esfuerzo en horas-hombre contemplando sólo el desarrollo de la funcionalidad especificada en los casos de uso. Para una estimación más completa de la duración total del proyecto, hay que agregar a la estimación del esfuerzo obtenida por los Puntos de Casos de Uso, las estimaciones de esfuerzo de las demás actividades relacionadas con el desarrollo de software.

Obviamente, los valores no son absolutos sino que pueden variar de acuerdo a las características de la organización y del proyecto. Con este criterio, y tomando como entrada la estimación de tiempo calculada a partir de los Puntos de Casos de Uso, se pueden calcular las demás estimaciones para obtener la duración total del proyecto.

A continuación, se detallan los pasos a seguir para la aplicación de éste método.

3.3.1. Paso 1: Cálculo de Casos de Uso Desajustados (24)

Este cálculo se realiza a través de la ecuación:

$$\mathbf{UUCP = UAW + UUCW}$$

Dónde:

Capítulo 3. “Estudio de factibilidad y Validación del sistema”

UUCP: Puntos de Casos de Uso sin Ajustar.

UAW: Factor de peso de los actores sin ajustar.

UUCW: Factor de peso de los casos de uso sin ajustar

Este valor del Factor de Peso de los Actores sin ajustar (UAW) se calcula mediante un análisis de la cantidad de actores presentes en el sistema y la complejidad de cada uno de ellos. La complejidad de los actores se establece teniendo en cuenta en primer lugar si se trata de una persona o de otro sistema y en segundo lugar, la forma en la que el actor interactúa con el sistema. Calculándose de la siguiente manera:

Tipo de Actor	Descripción	Factor de Peso	Actores	Total
Simple	Sistema con sistema a través de interfaz de programación.	1	0	0
Medio	Sistema con sistema mediante protocolo de interfaz basada en texto.	2	0	0
Complejo	Persona que interactúa con el sistema mediante interfaz gráfica.	3	1	3
	Total		1	3

Tabla 10. Peso de Actores sin ajustar

Por lo que:

$$UAW = \Sigma \text{ cantidad de actores} * \text{peso}$$

$$UAW = 1 * 3$$

$$UAW = 3$$

El valor Factor de Peso de los Casos de Uso sin ajustar (UUCW) se calcula mediante un análisis de la cantidad de Casos de Uso presentes en el sistema y la complejidad de cada uno de ellos. La complejidad de los Casos de Uso se establece teniendo en cuenta la cantidad de transacciones efectuadas en el mismo, donde una transacción se entiende como una secuencia de actividades atómicas, es decir, se efectúa la secuencia de

Capítulo 3. “Estudio de factibilidad y Validación del sistema”

actividades completas, o no se efectúa ninguna de las actividades de la secuencia y está representada por uno o más pasos del flujo de eventos principal del caso de uso, pudiendo existir más de una transacción dentro del mismo caso de uso. Los criterios se muestran en la siguiente tabla:

Tipo de Caso de Uso	Descripción	Factor de Peso	Cantidad de Casos Uso.	Total
Simple	El caso de uso tiene de 1 a 3 transacciones.	5	7	35
Medio	El caso de uso tiene de 4 a 7 transacciones.	10	0	0
Complejo	El caso de uso tiene más de 8 transacciones.	15	0	0
	Total		7	35

Tabla 11. Peso de las Transacciones

Por lo que:

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

$$UUCW = 35$$

Una vez calculado el valor del Factor de Peso de los Actores sin ajustar y el Peso de los Casos de Usos sin ajustar se puede calcular los Puntos de Casos de Uso sin ajustar:

$$UUCP = UAW + UUCW$$

$$UUCP = 3 + 35$$

$$UUCP = 38$$

3.3.2. Paso 2: Cálculo de los puntos de casos de uso ajustados.

Una vez que se tienen los Puntos de Casos de Uso sin ajustar, se debe ajustar éste valor mediante la siguiente ecuación:

$$UCP = UUCP * TCF * EF$$

Donde:

Capítulo 3. “Estudio de factibilidad y Validación del sistema”

UCP: Puntos de casos de uso ajustados.

UUCP: Puntos de casos de uso sin ajustar.

TCF: Factor de complejidad técnica.

EF: Factor de ambiente.

El Factor de Complejidad Técnica (TCF) se calcula mediante la cuantificación de un conjunto de factores que determinan la complejidad técnica del sistema. Cada factor se cuantifica en un valor desde 0 (aporte irrelevante) hasta 5 (aporte muy relevante).

Factor	Descripción	Peso	Valor Asignado	Total
T1	Sistema Distribuido.	2	0	0
T2	Objetivos de performance o tiempo de respuesta.	1	4	4
T3	Eficiencia del usuario final.	1	3	3
T4	Procesamiento interno complejo.	1	5	5
T5	El código debe ser reutilizable.	1	0	0
T6	Facilidad de instalación.	0.5	2	1
T7	Facilidad de uso.	0.5	5	2.5
T8	Portabilidad.	2	5	10
T9	Facilidad de cambio.	1	5	5
T10	Concurrencia.	1	2	2
T11	Incluye objetivos especiales de seguridad.	1	0	0
T12	Provee acceso directo a terceras partes.	1	5	5
T13	Se requieren facilidades especiales de entretenimiento a usuarios.	1	1	1
		Total	32	38,5

Tabla 12. Peso de los factores de alta complejidad técnica

Capítulo 3. “Estudio de factibilidad y Validación del sistema”

Significado de cada valor asignado

T1: El sistema no es distribuido.

T2: El tiempo de respuesta depende de la cantidad de información que se quiera procesar aunque debe ofrecer un buen rendimiento.

T3: El usuario final no requiere de una preparación intensa para interactuar con el sistema, solo necesita tener conocimientos básicos de informática y Sistemas de Información Geográfica.

T4: El procesamiento interno es medio.

T5: El código no es reutilizable, es decir no puede ser utilizado en el desarrollo de proyectos similares.

T6: La instalación del plugin y de la herramienta no es muy sencilla.

T7: La herramienta y el desarrollo del plugin cuentan con una interfaz gráfica, amigable, y muy sencilla de utilizar.

T8: La herramienta es multiplataforma, puede ser usada en Windows y Linux.

T9: A la herramienta se le pueden agregar los plugin que el usuario desee, ya sea para mejorarla o para facilitar su uso.

T10: Algunos de los procesos que realice el usuario pueden depender de otros.

T11: La herramienta no cuenta con servicios de seguridad, debido a que es una herramienta libre, de código abierto.

T12: La herramienta provee acceso a terceras partes ya que cuenta con conexión directa a ejemplos y a la página oficial de la misma.

T13: No es muy necesario darles preparación a los usuarios para utilizar el sistema.

Significado de los valores:

0: No presente o sin influencia.

1: Influencia o presencia incidental.

Capítulo 3. “Estudio de factibilidad y Validación del sistema”

2: Influencia o presencia moderada.

3: Influencia o presencia media.

4: Influencia o presencia significativa.

5: Influencia o presencia fuerte.

Por lo que:

$$TCF = 0.6 + 0.01 * \Sigma (\text{peso} * \text{valor asignado})$$

$$TCF = 0.6 + 0.01 * 38.5$$

$$TCF = 0.6 + 0.385$$

$$TCF = 0.985$$

El Factor de Ambiente (EF) está relacionado con las habilidades y entrenamiento del grupo de desarrollo que realiza el sistema. Cada factor se cuantifica con un valor desde 0 (aporte irrelevante) hasta 5 (aporte muy relevante). Es importante tener en cuenta:

- ✓ Para los factores E1 al E4, un valor asignado de 0 significa sin experiencia, 3 experiencia media y 5 amplia experiencia (experto).
- ✓ Para el factor E5, 0 significa sin motivación para el proyecto, 3 motivación media y 5 alta motivación.
- ✓ Para el factor E6, 0 significa requerimientos extremadamente inestables, 3 estabilidad media y 5 requerimientos estables sin posibilidad de cambios.
- ✓ Para el factor E7, 0 significa que no hay personal part-time (es decir todos son full-time), 3 significa mitad y mitad y 5 significa que todo el personal es part-time (nadie es full-time).
- ✓ Para el factor E8, 0 significa que el lenguaje de programación es fácil de usar, 3 medio y 5 que el lenguaje es extremadamente difícil.

Capítulo 3. “Estudio de factibilidad y Validación del sistema”

Factor	Descripción	Peso	Valor Asignado	Total
E1	Familiaridad con el modelo de proyecto utilizado.	1.5	5	7.5
E2	Experiencia en la aplicación.	0.5	0	0
E3	Experiencia en orientación a objetos.	1	1	1
E4	Capacidad del analista líder.	0.5	3	1.5
E5	Motivación.	1	5	5
E6	Estabilidad de los requerimientos.	2	5	10
E7	Personal part-time.	-1	0	0
E8	Dificultad del lenguaje de programación.	-1	3	-3
	Total		22	22

Tabla 13. Peso de los factores ambientales

Por lo que:

$$EF = 1.4 - 0.03 * \Sigma (\text{peso} * \text{valor asignado})$$

$$EF = 1.4 - 0.03 * 22$$

$$EF = 1.4 - 0.66$$

$$EF = 0.74$$

Una vez que se tiene el Factor de Ambiente y el Factor de Complejidad Técnica, conjuntamente con los Puntos de Casos de Uso sin ajustar que se habían calculado en el paso anterior, se puede calcular entonces los Puntos de casos de uso ajustados:

$$UCP = UUCP * TCF * EF$$

$$UCP = 38 * 0.985 * 0.74$$

$$UCP = 27.6982$$

Capítulo 3. “Estudio de factibilidad y Validación del sistema”

3.3.3. Paso 3: Estimación de esfuerzo a través de los puntos de casos de uso.

El esfuerzo en horas-hombres viene dado por la siguiente ecuación:

$$E = UPC * CF$$

Para obtener el factor de conversión (CF) se contabilizan cuántos valores de los que afectan el factor ambiente (E1...E6) están por debajo de la media (3) y los que están por arriba de la media para los restantes (E7, E8).

- ✓ Si el total es 2 o menos se utiliza el factor de conversión 20 Horas-Hombre / Punto de Casos de uso.
- ✓ Si el total es 3 ó 4 se utiliza el factor de conversión 28 Horas-Hombre / Punto de Casos de uso.
- ✓ Si el total es mayor o igual que 5 se recomienda efectuar cambios en el proyecto ya que se considera que el riesgo de fracaso del mismo es demasiado alto.

$$\text{Total}_{EF} = \text{Cantidad EF} < 3 \text{ (entre E1 –E6)} + \text{Cantidad EF} > 3 \text{ (entre E7, E8)}$$

$$\text{Total}_{EF} = 1 + 0$$

$$\text{Total}_{EF} = 1$$

Se puede decir que:

$$CF = 20 \text{ Horas-Hombre / Punto de Casos de uso}$$

Por tanto:

$$E = UCP * CF$$

$$E = 27.6982 * 20$$

$$E = 553.964 \text{ horas -hombres}$$

3.3.4. Paso 4: Calcular el esfuerzo de todo el proyecto. (24)

Para una estimación completa de la duración total del proyecto, hay que añadir a la estimación del esfuerzo obtenida por los Puntos de Casos de Uso, las estimaciones de esfuerzo de las demás actividades relacionadas con el desarrollo de software. Para ello

Capítulo 3. “Estudio de factibilidad y Validación del sistema”

se puede tener en cuenta el siguiente criterio, que estadísticamente se considera aceptable. El criterio plantea la distribución del esfuerzo entre las diferentes actividades de un proyecto, según la siguiente aproximación:

Actividad	Porcentaje	Horas-Hombres
Análisis	10%	55.3964
Diseño	20%	110.7928
Programación	40%	221.5856
Pruebas	15%	83.0946
Otras actividades	15%	83.0946
Total	100%	553.964

Tabla 14. Distribución en porcentaje para el esfuerzo total del proyecto

Suponiendo que 1 persona trabaje 24 días del mes y 8 horas por día tiene como promedio 192 horas de trabajo. Si $E_{Total} = 553.964$ horas-hombre y se divide por 192 horas se tiene un $ET = 2.88$ mes-hombre. Esto quiere decir que 1 persona puede realizar la implementación del prototipo no funcional del plugin de perfiles de visibilidad y altura para la plataforma uDig analizado aproximadamente en 3 meses.

3.4. Beneficios tangibles e intangibles.

La aplicación descrita brindará como beneficios tangibles un plugin eficiente para facilitarles el trabajo a todas aquellas personas que trabajen con SIG de escritorio y deseen calcular perfiles de visibilidad y altura, además de otros servicios que brinda el plugin y la herramienta.

Por tanto aparte de los beneficios tangibles que representa la realización del plugin, también se generan beneficios intangibles como son:

- ✓ Facilitar la sencillez y rapidez para localizar información acerca del plugin.
- ✓ Brindar la posibilidad de mantener actualizados los niveles de la documentación, tanto físicos como lógicos.
- ✓ Lograr de forma general dar solución a las numerosas inconformidades resumidas en el levantamiento de requisitos, satisfaciendo las necesidades del cálculo de los perfiles de visibilidad y altura.

Capítulo 3. “Estudio de factibilidad y Validación del sistema”

3.5. Análisis Costo-Beneficio

Debido a la utilización de herramientas libres, no es necesario realizar gastos referentes a las licencias de implementación e implantación.

Para el desarrollo del prototipo no funcional del plugin de perfiles de visibilidad y altura sobre la plataforma uDig la universidad cuenta con los recursos técnicos necesarios.

Si se tiene en cuenta los beneficios que se obtendrán al poner en marcha el sistema en cuanto a calidad, ahorro de recursos, tiempo y esfuerzos se puede resumir que el costo del proyecto no es elevado, por lo que se llega a la conclusión que es factible desarrollar el prototipo no funcional del plugin de perfiles de visibilidad y altura sobre la plataforma Udig.

3.6. Métodos de Validación

La validación revisa el cumplimiento de las características de la especificación de requisitos. La validación de requerimientos es importante pues de ella depende que no existan elevados costos de mantenimiento para el software desarrollado. En la validación las preguntas más frecuentes son: ¿están bien descritos los requisitos? o ¿el documento de requisitos representa claramente el sistema?, hoy en día existen diferentes técnicas y métodos de validación entre los que se encuentran:

- ✓ Reviews o Walk-throughs.
- ✓ Auditorías.
- ✓ Matrices de trazabilidad.
- ✓ Prototipos.
- ✓ Tesoros.
- ✓ Comparación de los resultados de salida del modelo con los del sistema real.
- ✓ Método Delphi.
- ✓ Test de Turing.

3.6.1. Método Delphi

El método de validación Delphi consiste en seleccionar un grupo de expertos en el tema y circular entre ellos de manera individual un cuestionario que contenga las preguntas que

Capítulo 3. “Estudio de factibilidad y Validación del sistema”

se consideren más importantes sobre el tema en cuestión. Basándose en las respuestas dadas a las cuestiones planteadas, se elaboran nuevos cuestionarios que van centrándose en temas más específicos, estos nuevos cuestionarios, son enviados a los expertos junto con las respuestas obtenidas en rondas anteriores. Estos pasos pueden ir repitiéndose hasta conseguir por parte del equipo de expertos una predicción de la respuesta del sistema.

En este método se excluyen las discusiones cara a cara de los miembros del equipo de expertos y su mayor desventaja es que consume mucho tiempo, por eso, en la mayoría de los casos no se realizan muchas iteraciones en cuanto a la elaboración de nuevos test y su análisis. (25)

3.6.2. Test de Turing

Alan Turing sugirió este método como un test de inteligencia artificial. En este test, a un experto, o grupo de expertos, se le presentan resúmenes o informes de resultados de ejecución del sistema y del modelo, a los que se les ha dado el mismo formato. Estos informes se reparten aleatoriamente a los ingenieros y administradores del sistema, para ver si son capaces de discernir cuáles son los reales del sistema y cuales la imitación resultado de la simulación. Si los expertos no son capaces de distinguir entre ambos, se puede concluir que no hay evidencias para considerar inadecuado al modelo. Si descubren diferencias las respuestas sobre lo que encuentran inconsistente se puede utilizar para realizar mejoras en el modelo.

Se puede considerar que este método es el inverso al método de Delphi. En el test de Turing se consulta a los expertos para ver si son capaces de identificar las respuestas del sistema, mientras que en el de Delphi se pregunta a los expertos para que predigan las respuestas del sistema.

Aunque este test parece muy intuitivo, hay muy pocos informes de su uso, ya que requiere un esfuerzo considerable para formatear las medidas de ejecución del sistema a la hora de crear el informe que se da a los expertos. Otra dificultad está en ajustar las medias del sistema real ya que en ellas intervienen elementos que no se han considerado en el modelo. Por último, este test requiere un análisis estadístico por parte del grupo de expertos para determinar si hay diferencias significativas entre el informe real y el simulado. (26)

Capítulo 3. “Estudio de factibilidad y Validación del sistema”

3.6.3. Comparación de los resultados de salida del modelo con los del sistema real.

Para poder hacer uso de este método, el sistema real debe existir, además se comparan las salidas del modelo con las del sistema, mediante algún método estadístico. El mayor inconveniente que tiene este método es que la mayoría de los procesos de salida no son estacionarios, provocando que estos test no sean directamente aplicables. (26)

Después de realizar un análisis sobre los diferentes métodos de validación existente se decide validar el sistema mediante matriz de trazabilidad, auditorías, revisiones y prototipos.

3.6.4. Matriz de Trazabilidad

La matriz de trazabilidad es una técnica empleada para validar los requerimientos identificados. Esta técnica consiste en verificar que los casos de usos descritos satisfacen todos los requerimientos del sistema.

Una matriz de trazabilidad es creada por la asociación de necesidades con los productos de trabajo que las satisfacen. La trazabilidad garantiza la integridad, que todos los requisitos de nivel inferior provienen de los requerimientos de nivel superior, y que todos los requisitos de nivel superior se asignan a los niveles inferiores. La trazabilidad también proporciona la base para la planificación de controles. (20)

A continuación se muestra la matriz de trazabilidad donde todos los requisitos funcionales (RF) se encuentran ubicados de forma vertical y los casos de usos del sistema (CUS) de forma horizontal, de manera que puedan hacerse corresponder entre ellos.

RF \ CUS	Trazar Perfiles de Altura.	Trazar perfiles de Visibilidad.	Calcular Superficie.	Localizar por coordenadas.
Trazar Perfiles de Altura.	X			
Trazar perfiles de Visibilidad.		X		

Capítulo 3. “Estudio de factibilidad y Validación del sistema”

A partir del trazado de una región determinada el sistema deberá visualizar los valores de superficie de dicha región.			X	
El sistema debe ser capaz de localizar cualquier objeto del mapa de acuerdo a coordenadas proporcionadas por el usuario.				X
<div style="display: flex; justify-content: space-between;"> RF CUS </div>	Obtener Coordenadas.	Crear Ruta	Visualizar Ruta	
El sistema debe ser capaz de obtener coordenadas mediante el establecimiento de un punto en el mapa.	X			
Crear Rutas.		X		
Visualizar Ruta			X	

Tabla 15. Matriz de Trazabilidad entre Requisitos Funcionales y Casos de Usos

Capítulo 3. “Estudio de factibilidad y Validación del sistema”

3.6.5. Listas de Chequeo

Estas listas de chequeo no son más que una serie de preguntas a realizar, o una serie de aspectos a considerar, los cuales pretenden apoyar al analista durante la identificación de defectos en la especificación. A través de ellas se verifica si existe necesidad de la elaboración de determinadas disposiciones, además del cumplimiento de algunas reglas establecidas. A continuación se muestran las siguientes tablas correspondientes a las listas de chequeo aplicadas a los requisitos obtenidos con el objetivo de verificar que estén correctos.

Evidencia	Nivel de importancia	Procedimiento	Respuesta
1. ¿Se refleja el seguimiento del documento?	B	Revisión del documento	Sí
2. ¿Se especifica el número de cada requisito evaluado?	A	Revisión del documento	Sí
3. ¿Se especifica el nombre de cada requisito solicitado?	A	Revisión del documento	Sí
4. ¿Se especifica si están identificados los elementos de entrada de cada requisito?	A	Revisión del documento	Sí
5. ¿Se especifica si están identificados los elementos de salida de cada requisito?	A	Revisión del documento	Sí
6. ¿Se especifica si cada requisito no es ambiguo?	A	Revisión del documento	Sí
7. ¿Se especifica si cada requisito es correcto?	A	Revisión del documento	Sí

Capítulo 3. “Estudio de factibilidad y Validación del sistema”

8. ¿El requisito puede ser verificado?	A	Revisión del documento	Sí
--	---	------------------------	----

Tabla 16. Lista de Chequeo “Requerimientos”

Evidencia	Nivel de importancia	Procedimiento	Respuesta
¿Están todos los requisitos redactados de forma simple y clara para aquellos que vayan a consultarlo en un futuro?	M	Revisión del documento	Sí
¿Se han enumerado los requisitos incluso los que se derivan de otros requisitos?	A	Revisión del documento	Sí
¿Se ha identificado si algún actor creará, almacenará, modificará o borrará información del sistema? ¿Ha especificado que casos de uso harán estas funciones?	M	Revisión del documento	Sí
¿Cada funcionalidad del sistema ha sido representada gráficamente?	M	Revisión del documento	Sí
¿Ha identificado los requisitos de software y de hardware?	A	Revisión del documento	Sí
¿Han sido identificadas las restricciones de diseño e implementación?	A	Revisión del documento	Sí

Capítulo 3. “Estudio de factibilidad y Validación del sistema”

¿La precondition es válida tanto para flujos básicos como flujos alternativos?	M	Revisión del documento	Sí
¿La poscondición plasma cambios que suceden en el sistema al terminarse de ejecutar el caso de uso?	M	Revisión del documento	Sí

Tabla 17. Lista de Chequeo “Especificación de Requisitos”

Con la utilización de las listas de chequeo de Calidad UCI, se valida la integridad de las funcionalidades que presenta la aplicación y además se garantiza la calidad del producto, el cual estará similarmente a la altura de otros a nivel internacional.

3.6.6. Revisiones

Con la revisión de la documentación del prototipo no funcional del plugin de perfiles de visibilidad y altura sobre la plataforma uDig se puede validar la correcta interpretación de la información transmitida, para que en un futuro el prototipo pueda ser implementado por personas que a pesar de no haber estado presentes en los inicios del proceso de desarrollo de software, entiendan a través de la documentación generada qué es lo que se quiere hacer. Al emplear esta técnica es recomendable proporcionar la documentación a personas ajenas al proceso para ver si comprenden lo expuesto.

3.6.7. Prototipos (28)

Los prototipos son un método de validación ampliamente utilizado en muchas disciplinas, y en todos los casos, los principios subyacentes son los mismos: el prototipado consiste en la creación de una maqueta o versión del producto final. Los objetivos de los prototipos varían en función de la disciplina. En el caso de la actividad de requisitos, los prototipos se utilizan, fundamentalmente, para comprobar la corrección y completitud de la especificación de requisitos.

Existen varios tipos de prototipos, cada uno de los cuales permite la realización de un tipo determinado de pruebas y con un determinado nivel de realismo. En ingeniería de requisitos, los prototipos más comunes son los siguientes:

Capítulo 3. “Estudio de factibilidad y Validación del sistema”

- ✓ Mock-ups. Se trata de pantallas, típicamente dibujadas a mano en papel, que representan un aspecto concreto del sistema. El soporte que proporcionan a la validación es muy limitado, con la excepción, quizás, de aclarar el interfaz gráfico deseado en casos complejos.
- ✓ Storyboards. Son una evolución de los mock-ups, ya que además del interfaz, se muestra la secuencia de acciones, o escenarios, que se deben realizar con el programa. Por ejemplo: es habitual que, antes de cerrar un programa, se pregunte si se desea cerrar el fichero de trabajo. En un storyboard, la tarea “cerrar programa”.
- ✓ Maquetas. Una maqueta es una versión simplificada del sistema software deseado. Típicamente, una maqueta representa únicamente el interfaz del sistema y, opcionalmente, las conexiones entre pantallas mediante la utilización de elementos activos como los botones. Si fuera necesaria mayor fidelidad, podrían codificarse partes del sistema, de tal modo que además, del interfaz, el software pudiera ofrecer algunos resultados reales. Ello es lo que se conoce como “prototipo funcional”.

Construir el prototipo es únicamente una parte, y quizás la más sencilla, de la tarea de validación. Para validar los requisitos, es necesario realizar adicionalmente las tareas indicadas a continuación:

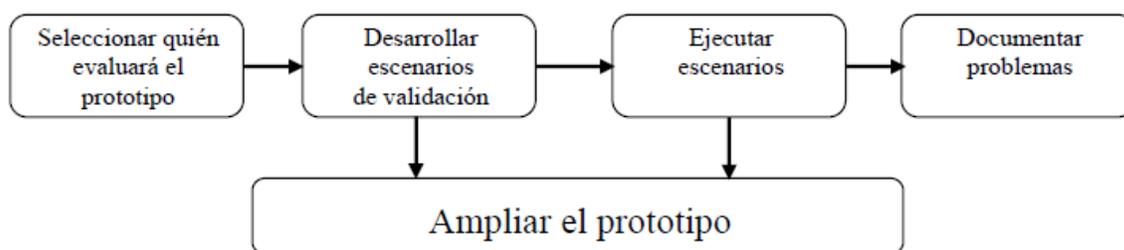


Figura 9. Prototipo en el proceso de validación de requisitos

Seleccionar quién evaluará el prototipo. Para que la evaluación del prototipo sea lo más efectiva posible, deben seleccionarse adecuadamente los usuarios que participarán en la evaluación. Siempre que sea posible, debe seleccionarse un conjunto de usuarios representativos de los distintos perfiles, de tal modo que sea posible validar el software en todos sus modos de utilización. Por ejemplo: Si un software X es utilizado por

Capítulo 3. “Estudio de factibilidad y Validación del sistema”

administrativos, ingenieros y contables, al menos un usuario de cada uno de estos tipos de usuarios debería evaluar el prototipo de X. En la mayoría de los casos, los usuarios pueden identificarse directamente a partir del detalle de la especificación.

Desarrollar escenarios de validación. En muchos textos, se dice que el prototipo se presenta a los usuarios para que “jueguen con él”. Ello, no obstante, dista de ser cierto. Para evaluar correctamente el prototipo, deben identificarse una serie de escenarios o tareas, los cuales deben ser llevados a cabo por los usuarios utilizando el prototipo. Ejemplos de estos escenarios podrían ser: crear un pedido para el que hay existencias, crear un pedido con algún producto sin existencias suficientes, crear un pedido para un producto inexistente.

Ejecutar los escenarios. Esta tarea consiste en que el usuario realice, o ejecute, cada uno de los escenarios previstos. Dependiendo del tipo de prototipo utilizado (mock-up, storyboard, maqueta, etc.), el analista deberá orientar al usuario acerca de la diferencia entre el producto final y el prototipo (de tal modo que el usuario no se muestre descontento). Asimismo, el analista deberá paliar las carencias del prototipo, proporcionando al usuario la información que el prototipo no ofrezca y que el usuario necesite para ejecutar los escenarios; esto es: el analista debe suplir la funcionalidad no presente en el prototipo.

Documentar los problemas. Esta tarea consiste, simplemente, en confeccionar la lista de problemas encontrados. Dado que el analista que confecciona los requisitos es, habitualmente, el mismo que valida el prototipo con los usuarios, no es usual confeccionar una lista de problemas recomendados.

3.7. Conclusiones

En este capítulo se realiza un análisis de la factibilidad del prototipo no funcional del plugin de perfiles de visibilidad y altura sobre la plataforma uDig, el cual sirve para determinar la duración del mismo estimándose el esfuerzo y que es viable llevarlo a cabo, además, se efectúa la validación de la solución propuesta, la cual permite confirmar que los requisitos funcionales definidos están descritos con claridad, son medibles y cumplen con los estándares de calidad requeridos.

Conclusiones Generales

- ✓ El desarrollo de este trabajo permitió elaborar el prototipo no funcional del plugin de perfiles de visibilidad y altura sobre la plataforma uDig, dándole cumplimiento a los objetivos trazados.
- ✓ Para el avance del prototipo se utilizaron herramientas libres y multiplataforma lo que permitirá que su código sea eficiente.
- ✓ Los diferentes métodos teóricos aplicados como la observación y la entrevista permitieron obtener la información necesaria para conocer las particularidades y funcionamiento de la herramienta lo que posibilitó realizar el proceso de modelación del mismo.
- ✓ El levantamiento de requisitos permitió obtener las funcionalidades requeridas, se realizó la documentación referente al análisis del sistema posibilitando el cumplimiento de los objetivos trazados.
- ✓ El desarrollo del presente trabajo de diploma permitió la elaboración de todos los artefactos correspondientes al análisis, obteniendo como resultado una amplia y organizada documentación sobre el plugin de los perfiles de visibilidad y altura sobre la plataforma uDig.

Recomendaciones

Al concluir el presente trabajo de diploma se realiza una serie de recomendaciones que podrán tenerse en cuenta para el desarrollo futuro del sistema:

- ✓ Se propone darle seguimiento para lograr un producto de mayor calidad.
- ✓ Proseguir con el estudio realizado con el propósito de añadir nuevas funcionalidades a la herramienta.

Bibliografía Citada

1. Instituto de Investigación de Recursos Biológicos Alexander Von Humboldt. [En línea] <http://www.humboldt.org.co/humboldt/mostrarpagina.php?codpage=70001.>].
2. **Felicísimo, Miguel Angel.** *Modelo Digital del Terreno. Introducción y aplicaciones en las ciencias ambientales.*
3. **Leyva Ruiz, Mariem.** Algoritmos para el Módulo de análisis del terreno de la plataforma GENESIG: Modelos Perfiles de Visibilidad y Altura. Ciudad Habana : s.n., 2010.
4. **Gutierrez Puebla, Javier y Gould, Michael.** Los Sistemas de Información Geográfica.
5. **Gómez, Heriberto y Linares, Rosalba.** *SIG:Un arma para la frontera.*
6. **Rodríguez Vázquez, Dwam.** *Algoritmos para la Modelación Digital del Terreno.* Ciudad Habana : s.n., 2005.
7. **Felicísimo, Miguel Angel.** *Introducción a los modelos digitales del terreno.*
8. —. *El modelo digital de elevaciones.* 1994.
9. —. *La utilización de los MDT en los estudios del medio físico.*
10. **Durán Díaz, Ing. Juan Javier.** *MODELOS DIGITALES DE ELEVACIÓN PARA EL CONTINUO DE ELEVACIONES MEXICANO.*
11. **Jimenez Berni, José Antonio, Aguilera, Jesús y Morroño, José Emilio.** *ALTERNATIVAS DE SOFTWARE LIBRE A LOS SISTEMAS.*
12. EcuRed. [En línea] <http://www.ecured.cu/index.php/RUP>.
13. **González Abad, Israeldis.** *Diseño e Implementación del Sistema de Gestión de Información de los Recursos de la Facultad 3.* Ciudad Habana : s.n., 2010.
14. **Fernández, Escribano Gerardo.** *Introducción a Extreme Programing.* 9/12/2002.
15. **García Hernández, Yanisleidys.** *PORTAL DIGITAL PARA EL GRUPO EMPRESARIAL ASTRO.* 2006.
16. free dowload manager. [En línea] [Citado el: 4 de diciembre de 2010.] http://www.freedownloadmanager.org/es/downloads/Paradigma_Visual_para_UML_%28M%C3%8D%29_14720_p/.
17. **Case, Instituto Nacional de Estadísticas e Informática. Herramientas.** *Talleres de la Oficina de Impresiones de la Oficina Técnica de Difusión.* Noviembre, 2009.
18. **Gerrero, Hanz Cocchi.** El Geek. [En línea] hanzcocchi.net.

19. **Escalona Cuaresma, Dra. María José y González Romano, Dr. José Mariano.** Lenguajes y Sistemas de Información. [En línea] 2006-2007.
<http://www.lsi.us.es/docencia/get.php?id=2118>.
20. **Lazo, Yugleiny Pellicier.** *Módulo para la Gestión de Metadatos Geográficos según la norma ISO19115 del sistema GeoMetaWeb. Rol analista.* Habana : s.n., 2011.
21. **Jacobson, y otros.** 2000.
22. **Estrugo Lahera, Evelyn y Cuellar Rodríguez, Dany.** *Diseño e implementación del portal web para la dirección general de prevención del delito.* Ciudad Habana : s.n., 2010.
23. **LAGRAN., Profesor: Yussef Farrán.** *Estudio de Factibilidad.* Chile : s.n., 2000.
24. **Ibarra, Marcia y Fernández, Perez.** *Análisis y Diseño de los servicios de Información para la Intranet Corporativa PDVSA.* Universidad de las Ciencias Informáticas. : s.n., 2007.
25. **Betancourt Quintanal, Danay y Abreus Ruiz, Dairon.** *Desarrollo de Plug-ins de soporte para la herramienta AuditBD.* Habana : Universidad de las Ciencias Infomáticas., 2010.
26. **Miles, Russ y Pitone, Dan.** *Head First Software Development.*
27. **Soto Santana, Saili.** *Análisis del Geoportal de la Universidad de las Ciencias Informáticas.* . Habana : s.n., 2011.
28. Validación de Requisitos. [aut. libro] Master de Ingeniería de Software. *Requisitos de Sistemas de Software.*
29. **Hernández Placencia, Yelina.** *Análisis del Sistema de Información Geográfica para la representación de los servicios e informaciones médicas del Ministerio de Salud Pública.*

Bibliografía

1. Instituto de Investigación de Recursos Biológicos Alexander Von Humbolt. [En línea] <http://www.humboldt.org.co/humboldt/mostrarpagina.php?codpage=70001.>].
2. **Felicísimo, Miguel Angel.** *Modelo Digital del Terreno. Introducción y aplicaciones en las ciencias ambientales.*
3. **Leyva Ruiz, Mariem.** Algoritmos para el Módulo de análisis del terreno de la plataforma GENESIG: Modelos Perfiles de Visibilidad y Altura. Ciudad Habana : s.n., 2010.
4. **Gutierrez Puebla, Javier y Gould, Michael.** Los Sistemas de Información Geográfica.
5. **Gómez, Heriberto y Linares, Rosalba.** *SIG:Un arma para la frontera.*
6. **Rodríguez Vázquez, Dwam.** *Algoritmos para la Modelación Digital del Terreno.* Ciudad Habana : s.n., 2005.
7. **Felicísimo, Miguel Angel.** *Introducción a los modelos digitales del terreno.*
8. —. *El modelo digital de elevaciones.* 1994.
9. —. *La utilización de los MDT en los estudios del medio físico.*
10. **Durán Díaz, Ing. Juan Javier.** *MODELOS DIGITALES DE ELEVACIÓN PARA EL CONTINUO DE ELEVACIONES MEXICANO.*
11. **Jimenez Berni, José Antonio, Aguilera, Jesús y Morroño, José Emilio.** *ALTERNATIVAS DE SOFTWARE LIBRE A LOS SISTEMAS.*
12. EcuRed. [En línea] <http://www.ecured.cu/index.php/RUP>.
13. **González Abad, Israeldis.** *Diseño e Implementación del Sistema de Gestión de Información de los Recursos de la Facultad 3.* Ciudad Habana : s.n., 2010.
14. **Fernández, Escribano Gerardo.** *Introducción a Extreme Programing.* 9/12/2002.
15. **García Hernández, Yanisleidys.** *PORTAL DIGITAL PARA EL GRUPO EMPRESARIAL ASTRO.* 2006.
16. free dowload manager. [En línea] [Citado el: 4 de diciembre de 2010.] http://www.freedownloadmanager.org/es/downloads/Paradigma_Visual_para_UML_%28M%C3%8D%29_14720_p/.
17. **Case, Instituto Nacional de Estadísticas e Informática. Herramientas.** *Talleres de la Oficina de Impresiones de la Oficina Técnica de Difusión.* Noviembre, 2009.
18. **Gerrero, Hanz Cocchi.** El Geek. [En línea] hanzcocchi.net.

19. **Escalona Cuaresma, Dra. María José y González Romano, Dr. José Mariano.** Lenguajes y Sistemas de Información. [En línea] 2006-2007. <http://www.lsi.us.es/docencia/get.php?id=2118>.
20. **Lazo, Yugleiny Pellicier.** *Módulo para la Gestión de Metadatos Geográficos según la norma ISO19115 del sistema GeoMetaWeb. Rol analista.* Habana : s.n., 2011.
21. **Jacobson, y otros.** 2000.
22. **Estrugo Lahera, Evelyn y Cuellar Rodríguez, Dany.** *Diseño e implementación del portal web para la dirección general de prevención del delito.* Ciudad Habana : s.n., 2010.
23. **LAGRAN., Profesor: Yussef Farrán.** *Estudio de Factibilidad.* Chile : s.n., 2000.
24. **Ibarra, Marcia y Fernández, Perez.** *Análisis y Diseño de los servicios de Información para la Intranet Corporativa PDVSA.* Universidad de las Ciencias Informáticas. : s.n., 2007.
25. **Betancourt Quintanal, Danay y Abreus Ruiz, Dairon.** *Desarrollo de Plug-ins de soporte para la herramienta AuditBD.* Habana : Universidad de las Ciencias Infomáticas., 2010.
26. **Miles, Russ y Pilone, Dan.** *Head First Software Development.*
27. **Soto Santana, Saili.** *Análisis del Geoportal de la Universidad de las Ciencias Informáticas.* . Habana : s.n., 2011.
28. Validación de Requisitos. [aut. libro] Master de Ingeniería de Software. *Requisitos de Sistemas de Software.*
29. **Hernández Placencia, Yelina.** *Análisis del Sistema de Información Geográfica para la representación de los servicios e informaciones médicas del Ministerio de Salud Pública.*

Glosario de Términos

Cartografiar: Levantar y trazar la carta geográfica de una porción de superficie terrestre. Es la ciencia que se encarga del estudio y de la elaboración de los mapas geográficos, territoriales y de diferentes dimensiones lineales y demás.

Edafología: Ciencia que estudia las características de los suelos, su formación, su evolución, sus propiedades físicas, morfológicas, químicas, mineralógicas y su distribución. Comprende el estudio de las aptitudes de los suelos para la explotación agraria o forestal.

Evapotranspiración: Es la pérdida de humedad de una superficie por evaporación directa junto con la pérdida de agua por transpiración de la vegetación. Se expresa en mm por unidad de tiempo.

Fotogrametría: Ciencia desarrollada para obtener medidas reales a partir de fotografías, tanto terrestres como aéreas, para la realización mapas topográficos, mediciones y otras aplicaciones geográficas.

Geomorfología: Es la rama de la geografía que estudia el relieve de la Tierra, el cual es el resultado de un balance dinámico entre procesos constructivos y destructivos, dinámica que se conoce de manera genérica como ciclo geográfico.

Isohipsas: Líneas continuas utilizadas en la representación del relieve en los mapas topográficos que unen puntos situados a la misma altitud.

Isolíneas: Son curvas que se obtienen de unir o enlazar puntos con los que un determinado fenómeno alcanza el mismo valor, las isolíneas que se representan en un mapa son líneas, rectas o curvas, que describen la intersección de una superficie real o hipotética con uno o más planos horizontales.

Modelo Digital de Elevaciones (MDE): Es una estructura numérica de datos que representa la distribución espacial de la altitud de la superficie del terreno.

Modelo Digital del Terreno (MTD): No es más que una estructura numérica de datos que representa la distribución espacial de una variable cuantitativa y continua.

Topografía: Representación de los elementos naturales y humanos de la superficie terrestre. Esta ciencia determina los procedimientos que se siguen para poder representar esos elementos en los mapas y cartas geográficas.