

UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS

Facultad 6



*Trabajo de Diploma para optar por el título de Ingeniero en
Ciencias Informáticas.*

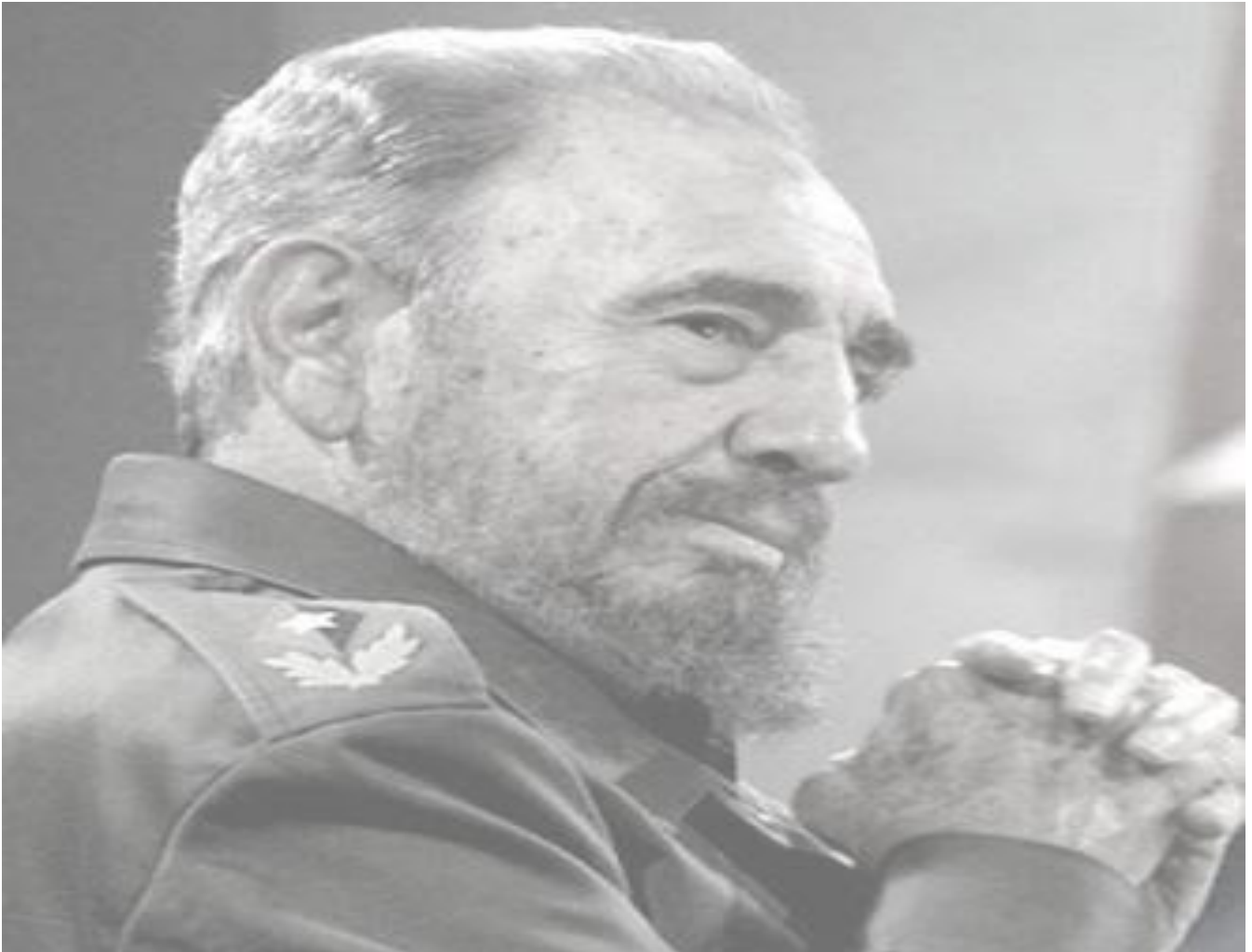
*Título: Análisis del componente para el análisis hidrológico, álgebra de
mapas y análisis geoestadístico en sistemas de información geográfica.*

Autor: Yaima Kindelán Suárez

Tutor: Ing. Lidisy Hernández Montero.

Co-Tutor: Ing. Romanuel Ramón Antunez.

“Año 53 de la Revolución.”



"... Siempre que existan tantas cabezas ardientes, llenas de fe y esperanza, tratando de hacer posible lo imposible, pensemos que todo no estará perdido..."

Fidel Castro Ruz.

Resumen

La Universidad de las Ciencias Informáticas (UCI), posee en su estructura el Centro de Desarrollo de GeoInformática y Señales Digitales (GEySED) perteneciente a la facultad 6. El departamento de GeoInformática entre sus objetivos tiene realizar la representación geoespacial de la información asociada a negocios específicos, permitiendo además realizar análisis sobre dicha información.

El departamento desarrolla la Plataforma GeneSIG que contiene un conjunto de funcionalidades específicas de este tipo de sistema, hoy carece de requisitos que permiten el análisis relacionado con el álgebra de mapas, análisis hidrológico y geoestadístico; siendo necesario realizar el análisis de un componente que contenga las mismas.

En el presente trabajo se presenta como solución al problema, el análisis del componente para álgebra de mapas, análisis hidrológico y geoestadístico, así como de algoritmos que permitirán la implementación de dicho componente.

Para lograr el objetivo general se realiza el estudio de diferentes modelos matemáticos, se trabaja en la investigación de algoritmos eficientes para resolver los problemas con el álgebra de mapas y análisis hidrológico; además se realiza un estudio de las tendencias y tecnologías actuales que fueron útiles para fundamentar y complementar el diseño de la solución propuesta.

Palabras claves

Álgebra de mapas, análisis hidrológico, análisis geoestadístico, sistemas de información geográfica (SIG).

Índice de Contenidos

INTRODUCCIÓN	6
CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA	10
1.1 INTRODUCCIÓN	10
1.2 CONCEPTOS ASOCIADOS AL DOMINIO DEL PROBLEMA	10
1.3 OBJETO DE ESTUDIO.....	14
1.3.1 ÁLGEBRA DE MAPAS	15
1.3.2 ANÁLISIS HIDROLÓGICO	18
1.3.3 ANÁLISIS GEOESTADÍSTICO	21
1.3.4 DESCRIPCIÓN DE MODELOS MATEMÁTICOS.....	22
1.4 ANÁLISIS DE SOLUCIONES EXISTENTES.....	23
1.4.1 SEXTANTE	23
1.4.2 GRASS.....	24
1.4.3 KOSMO	28
1.5 ARGUMENTAR EL USO DE LA METODOLOGÍA DE DESARROLLO.....	29
1.5.1 EL PROCESO UNIFICADO DE DESARROLLO DE SOFTWARE (RUP) COMO METODOLOGÍA PROPUESTA	31
1.6 LENGUAJE DE MODELADO	33
1.7 VISUAL PARADIGM COMO HERRAMIENTA DE MODELADO PROPUESTA	34
1.8 CONCLUSIONES PARCIALES	35
CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA	36
2.1 INTRODUCCIÓN	36
2.2 MODELO DE DOMINIO	36
2.3 DESCRIPCIÓN DE LAS FUNCIONALIDADES DEL SISTEMA	38
2.3.1 REQUISITOS FUNCIONALES	38
2.3.2 REQUISITOS NO FUNCIONALES.....	39
2.4 MODELO DE CASOS DE USO DEL SISTEMA	41
2.4.1 DETERMINACIÓN Y JUSTIFICACIÓN DE ACTORES.....	41
2.4.2 DIAGRAMA DE CASOS DE USO DEL SISTEMA	42
2.4.3 DESCRIPCIÓN TEXTUAL DE LOS CASOS DE USO DEL SISTEMA.....	42
2.5 CONCLUSIONES PARCIALES	60
CAPÍTULO 3: ESTUDIO DE FACTIBILIDAD Y VALIDACIÓN DEL SISTEMA	61
INTRODUCCIÓN.....	61
3.1 TÉCNICAS DE ESTIMACIÓN	61
3.1.1 SELECCIÓN DE LA TÉCNICA DE ESTIMACIÓN.....	62
3.2 TÉCNICAS DE VALIDACIÓN	72
3.3 CONCLUSIONES PARCIALES	78
CONCLUSIONES GENERALES	79
RECOMENDACIONES	81
BIBLIOGRAFÍA REFERENCIADA	82
BIBLIOGRAFÍA CONSULTADA	84

ANEXOS	¡ERROR! MARCADOR NO DEFINIDO.
ANEXO 1 DESCRIPCIÓN DETALLADA DE LOS REQUISITOS DEL SISTEMA.	¡ERROR! MARCADOR NO DEFINIDO.
ANEXO 2 DESCRIPCIÓN DETALLADA DE LOS CASOS DE USO DEL SISTEMA.....	¡ERROR! MARCADOR NO DEFINIDO.
GLOSARIO DE TÉRMINOS	¡ERROR! MARCADOR NO DEFINIDO.

Índice de Figuras

FIGURA 1. MODELO RASTER.	13
FIGURA 2 REPRESENTACIÓN DEL CÁLCULO POR FILTRADO DE CAPAS.	18
FIGURA 3 REPRESENTACIÓN DE UNA CUADRÍCULA QUE FORMA PARTE DE UNA REJILLA.	20
FIGURA 4: ETAPAS Y FLUJOS DE TRABAJO DE RUP	32
FIGURA 5 MODELO DE DOMINIO.	37
FIGURA 6 DIAGRAMA DE CASOS DE USO DEL SISTEMA.	42

Índice de Tablas

TABLA 1 PRINCIPALES POTENCIALIDADES DE GRASS.	24
TABLA 2 PRINCIPALES CARACTERÍSTICAS DE KOSMO.	28
TABLA 3 ACTORES DEL SISTEMA.	41
TABLA 4 REQUISITOS FUNCIONALES Y CASOS DE USOS DEL SISTEMA	59
TABLA 5 ACTORES Y CASOS DE USOS DEL SISTEMA	59
TABLA 6: FACTOR DE PESO DE LOS ACTORES SIN AJUSTAR.	63
TABLA 7: FACTOR DE PESO DE LOS CASOS DE USO SIN AJUSTAR.	64
TABLA 8: FACTOR DE COMPLEJIDAD TÉCNICA.	65
TABLA 9: FACTOR DE AMBIENTE.	67
TABLA 10: ESFUERZO DEL PROYECTO.	70
TABLA 11: LISTAS DE CHEQUEO DE ESPECIFICACIÓN DE REQUISITOS. (22)	74
TABLA 12:LISTAS DE CHEQUEO PARA CASOS DE USOS DEL SISTEMA. (22)	75

INTRODUCCIÓN

Con la presencia de las nuevas tecnologías y la necesidad del uso de estas en el manejo de la información geográfica en el mundo se fue desarrollando el ámbito perfecto para la aparición de los sistemas de información geográfica (SIG), los cuales tuvieron como factor importante para su surgimiento la necesidad del uso, gestión y manejo de la información geográfica. Uno de los pilares que favorece que estos sistemas hayan experimentado una evolución hasta llegar a la situación actual es la cartografía tradicional y los procedimientos técnicos para el análisis de mapas.

Se puede decir que un Sistema de Información Geográfica es 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 de sus atributos con el fin de satisfacer múltiples propósitos. **(1)**

Desde su surgimiento y hasta la fecha los SIG han evolucionado por varias etapas en correspondencia con el propio desarrollo de las Tecnologías de la Información y las Comunicaciones. Los primeros SIG desarrollados entre las décadas del sesenta y el ochenta estaban orientados a un proyecto, donde toda la información se almacenaba en una única computadora, ejecutándose el SIG también en ella. Después estos sistemas fueron ampliando su conectividad dentro de la empresa, en intranets corporativas, y finalmente, fue necesario que surgieran enfoques orientados a la Web para satisfacer las demandas de toda la sociedad.

Los SIG están de moda hoy en día; más allá de la espectacularidad de sus resultados, o la amplitud de su campo de aplicaciones, esta herramienta se ha popularizado básicamente porque se considera que entre el ochenta y el noventa por ciento de toda la información involucrada en la toma de decisiones de la sociedad a nivel global, tiene una componente espacial. Se trata de una disciplina joven y ciertamente compleja, un nuevo paradigma que forma parte del ámbito más extenso de los Sistemas de Información (SI). Es una herramienta multipropósito con aplicaciones en los campos más diversos.

En la actualidad estos sistemas son un instrumento fundamental en la transferencia del conocimiento del mundo real a modelos, que serán utilizados posteriormente en el análisis y toma de decisiones en sus

diversas aplicaciones dentro de actividades como: la gestión de recursos naturales y medio ambiente, la planificación urbana, el mantenimiento de redes hidráulicas, eléctricas, telefónicas y alcantarillados, por citar algunos ejemplos.

El incremento del desarrollo de estos aspectos dieron origen a diversas herramientas puntuales como: MIMO, CGIS, SyMAP, ESRI, Arcinfo, GRASS; que con el tiempo fueron perfeccionadas y enriquecidas con nuevos métodos y algoritmos que ofrecieron una nueva vía de implementación y uso a los nuevos SIG. **(2)**

Actualmente la aplicación de estas herramientas han pasado del uso investigativo y profesional para tener una actividad más social e interactiva con los usuarios, ejemplo son *GoogleMap*, *GoogleEarth*, los que han propiciado la búsqueda y desarrollo de alternativas de programas geográficos de fuente abierta para usos Web.

Cuba no está ajena a la incorporación de este tipo de herramientas, tal es así que en el año 2003 se crea la Universidad de las Ciencias Informáticas, en lo adelante UCI, estructurada en 7 facultades y estas desarrollan la actividad productiva dentro de los centros de desarrollo, como el Centro de Desarrollo de Geoinformática y Señales Digitales (GEySED), donde el Departamento de Geoinformática tiene entre sus principales objetivos ofrecer soluciones geoespaciales que puedan insertarse en el mercado nacional e internacional. GeneSIG es un proyecto en estado de progreso perteneciente a dicho departamento, no cubre los procesos de análisis del terreno relacionados con el álgebra de mapas, análisis hidrológico y geoestadístico, trayendo consigo el aumento de la carga basada en papel, la práctica del flujo es meramente manual, lo que puede acarrear errores humanos en el proceso de análisis de los datos implicados, así se describe la **problemática** actual.

El **problema a resolver** se define en: ¿Cómo contribuir al desarrollo del componente para el análisis hidrológico, álgebra de mapas y análisis geoestadístico en Sistemas de Información Geográfica?

El objeto de estudio se enmarca en los procesos de gestión pertenecientes a los Sistemas de Información Geográfica; y el campo de acción en los procesos de gestión involucrados con el álgebra de mapas, análisis hidrológico y geoestadístico.

Para darle solución a este problema se tiene como objetivo general: Elaborar el análisis correspondiente al componente para el análisis hidrológico, álgebra de mapas y análisis geoestadístico en SIG.

La idea a defender es la elaboración adecuada de la documentación técnica correspondiente al análisis

del componente para el análisis hidrológico, álgebra de mapas y análisis geoestadístico en sistemas de información geográfica, permitirá sentar la base para agilizar el desarrollo de SIG en entornos web y de escritorio que requieran de estos módulos.

Para el cumplimiento de este objetivo se han trazado las siguientes tareas:

1. Caracterizar los modelos y algoritmos para el análisis hidrológico, estadístico y álgebra de mapas.
2. Argumentar el uso de la metodología de desarrollo a utilizar, lenguaje de modelado y herramientas CASE.
3. Elaborar la documentación técnica correspondiente al análisis del componente.
4. Validar cada uno de los requisitos funcionales del sistema.

Para lograr el objetivo de la investigación se aplicaron los siguientes métodos de la investigación científica:

Métodos teóricos:

- **Inductivo - Deductivo:** Facilita el análisis de los elementos generales a elementos más particulares. Teniendo como tema general el álgebra de mapas, análisis hidrológico y geoestadístico.
- **Histórico - Lógico:** Permite la recopilación de información respecto a los algoritmos y herramientas para la representación del análisis hidrológico, geoestadístico y álgebra de mapas.
- **Analítico – Sintético:** Este método hace posible la comparación entre las diferentes funciones y algoritmos para el análisis hidrológico, geoestadístico y álgebra de mapas, permitiendo seleccionar al más eficiente, así como reunir los elementos necesarios para la propuesta de un sistema potente que cumpla con los requisitos deseados.
- **Modelación:** Se utiliza como guía para la representación mediante diagramas del modelo del sistema en términos de casos de uso y actores.

Métodos empíricos:

- **Entrevista:** Necesaria para planificar entrevistas al tutor, cotutor, así como los integrantes del proyecto GeneSIG.

- **Observación:** Este método es de vital importancia ya que ayuda a percibir a partir de la situación real que se está investigando, cómo se desarrollan a groso modo los procesos que constituyen el objeto de estudio.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

1.1 Introducción

En el presente capítulo se muestra la investigación concerniente al objeto de estudio, así como los términos utilizados y el análisis de distintas documentaciones existentes. Analizando aspectos fundamentales para la comprensión del problema planteado y los conceptos más importantes.

1.2 Conceptos asociados al dominio del problema

Para la correcta comprensión del trabajo, es necesario especificar importantes conceptos, que son conducentes y esenciales para lograr el entendimiento del dominio del problema.

Definiciones de SIG

Un Sistema de Información Geográfica es una colección organizada de hardware, software y datos geográficos diseñados para la eficiente captura, almacenamiento, integración, actualización, modificación, análisis espacial, y despliegue de todo tipo de información geográficamente referenciada. **(2)**

Los Sistemas de Información Geográfica representan un campo de desarrollo, donde se interceptan muchas disciplinas, entre ellas, la cartografía, la computación, la fotogrametría, la teledetección, la estadística y otras disciplinas relacionadas con el manejo y análisis de datos territorialmente codificados. **(2)**

Según las definiciones expresadas anteriormente se puede decir que los SIG permiten la ubicación y gestión de datos georreferenciados con el fin de hacer más factible y puntual la toma de decisiones, dado el nivel de exactitud en cuanto a la ubicación geográfica de los objetos en el espacio; así como la representación geográfica de modelos de mapas utilizados para el cálculo y modelación de fenómenos ambientales. Los SIG vienen desarrollándose en escalas que van desde catastrales hasta globales por su aplicación tanto en las ciencias ambientales como en las ingeniarías de diversos tipos. Se puede decir que son el paso adelante más importante desde la invención del mapa.

Las propiedades expuestas en las definiciones anteriores no sólo definen a los SIG, sino también existen otros aspectos que le añaden ciertos matices a lo expuesto hasta ahora y que lo caracterizan:

1. La capacidad de visualizar la información geográfica a través de mapas.
2. La función como base de datos sofisticada en la que se guardan y relacionan información espacial al igual que temática.
3. La diferencia con las bases de datos convencionales, puesto que toda la información contenida en un SIG está unida con entidades geográficamente localizadas.
4. Su origen a partir de la unión e innovación tecnológica en campos especializados de la geografía y otras ciencias, propició un sistema más potente que la suma de las partes.
5. Permite agrupar la información en estructuras coherentes y aplicar a la misma una armadura variada de funciones: análisis, visualización, edición, etc.
6. El carácter integrador y abierto hace que sea uno de los más importantes entre variados tipos de aplicaciones informáticas, destinadas al manejo de información. **(4)**

Algoritmo

Un algoritmo es un conjunto de operaciones ordenadas de modo tal que puedan resolver un problema. Es un conjunto de instrucciones o reglas bien definidas, ordenadas y finitas que permiten realizar una actividad mediante pasos sucesivos que no generen dudas a quien deba realizar dicha actividad. **(5)**

Mapa

Modelo gráfico de la superficie terrestre donde se representan objetos espaciales y sus propiedades métricas, topológicas y atributivas, un mapa puede ser analógico (impreso sobre papel, por ejemplo) o digital (codificado en cifras, almacenado en un ordenador y presentado en una pantalla). Existen mapas métricos, diseñados para representar distancias, superficies o ángulos y mapas topológicos, diseñados para representar vecindad, inclusión, conectividad y orden en el contexto de los SIG. Un mapa es la presentación de cualquier estructura de datos usada para reflejar cartográficamente una variable espacial, nominal o cuantitativa, independientemente del modelo de datos utilizado que puede ser vectorial o raster. **(10)**

Modelos

Los modelos se construyen para conocer o predecir propiedades del objeto real. Un modelo es un objeto, concepto o conjunto de relaciones que se utilizan para representar y estudiar de forma simple y comprensible una porción de la realidad empírica.

Los modelos se cimientan estableciendo una relación de correspondencia con la realidad cuyas variantes pueden producir modelos de características notablemente diferentes, distingue tres tipos básicos:

- Modelos icónicos: La relación de correspondencia se establece a través de las propiedades morfológicas, una maqueta es un modelo del objeto representado donde la relación establecida es fundamentalmente una reducción de escala.
- Modelos análogos: Poseen algunas propiedades similares a los objetos representados pero sin ser una réplica morfológica de los mismos, un mapa es un modelo de la realidad establecido mediante un conjunto de convenciones relativamente complejo que conduce a un resultado final claramente distinto del objeto representado.
- Modelos simbólicos: Se llega a un nivel superior de abstracción debido a que el objeto real queda representado mediante una simbolización matemática, geométrica, estadística. **(4)**

Modelo hidrológico

Un modelo hidrológico es una representación simplificada del sistema real cuyo objetivo es estudiar la operación del sistema y predecir su salida. Sus entradas y salidas son variables hidrológicas mensurables y su estructura es un conjunto de ecuaciones que conectan las entradas con las salidas, las cuales pueden expresarse como función del tiempo. Abarcan una gran diversidad de problemas y funcionalidades tales como modelado de ríos y cuencas, calidad de aguas, predicción de crecidas, riesgos hidrológicos. **(6)**

Modelo digital de terreno (MDT)

Los MDT se incluyen en la categoría de modelos simbólicos y, consecuentemente, las relaciones de correspondencia que se establecen con el objeto tienen la forma de algoritmos o formalismos matemáticos. En este caso, los MDT presentan algunas ventajas sobre el resto de tipos de modelos,

derivadas de su naturaleza numérica: no ambigüedad, posibilidad de modelización de procesos con una deducción estricta, verificabilidad y repetitividad de los resultados. (7)

Un MDT es la representación de la topografía del terreno, aunque en realidad cualquier hecho que cumpla con la característica de la continuidad espacial de la variación (distribución espacial de una variable cuantitativa y continua), puede ser representado mediante este modelo, por ejemplo, las precipitaciones, las temperaturas o la humedad de un punto dado.

Un MDT se puede representar mediante dos modelos de datos:

1. La matriz de alturas (raster).
2. La estructura TIN (red de triángulos irregulares).

Capas raster

Se trata de un Modelo de datos en el cual las entidades geográficas son representadas usando celdas, generalmente cuadrados¹, ordenados conformando una grilla regular. Un raster es esencialmente como una matriz multidimensional cuyo origen se sitúa en el extremo superior izquierdo.



Figura 1. Modelo raster.

Dentro de las diversas clasificaciones de los datos raster se puede encontrar la clasificación según el valor del pixel:

- Continuos: Se refiere a datos que se encuentran distribuidos de forma continua, como su nombre indica, en cualquier lugar de la superficie terrestre por ejemplo: temperatura, precipitación, altitud, presión atmosférica y por tanto cualquier pixel tendrá un valor de dicha variable.

¹Aunque algunos sistemas utilizan otras figuras geométricas como triángulos o hexágonos.

- Discretos o temáticos: Los raster discretos que proceden de una discretización de acuerdo a un criterio, o se trata de representaciones de fenómenos cuyos límites quedan perfectamente establecidos como es el caso de un raster de usos del suelo, tipología de suelos, todos relacionados con una misma variable. El proceso de discretización de una variable continua pasa por la agrupación de los píxeles que componen el raster en determinadas unidades y la posterior asignación de un valor común a todos los píxeles que componen cada clase o intervalo, lo que se ha venido a denominar reclasificación de un raster continuo para la obtención de un raster discreto. **(8)**

1.3 Objeto de Estudio

La construcción e implementación de un SIG en cualquier organización es una tarea siempre progresiva, compleja, laboriosa y perenne. Los análisis y estudios anteriores a la implantación de un SIG son similares a los que se deben realizar para establecer cualquier otro sistema de información. Sin embargo, en los SIG hay que considerar las características especiales de los datos utilizados y sus correspondientes procesos de actualización.

Es indiscutible que los datos son el principal activo de cualquier sistema de información. Por ello el éxito y la eficacia de un SIG se miden por el tipo, la calidad y vigencia de los datos con los que opera. Los esfuerzos y la inversión necesaria para crear las bases de datos y tener un SIG eficiente y funcional no son pequeños, aunque tampoco significa una gran inversión. Es un esfuerzo permanente por ampliar y mejorar los datos almacenados, utilizando las herramientas más eficientes para tal propósito.

La información geográfica contiene una referencia territorial explícita como latitud y longitud o una referencia implícita como domicilio o código postal. Las referencias implícitas pueden ser derivadas de referencias explícitas mediante geocodificación. Los SIG funcionan con dos tipos diferentes de información geográfica: el modelo vector y el modelo raster.

El modelo raster funciona a través de una retícula que permite asociar datos a una imagen; es decir, se pueden relacionar paquetes de información a los píxeles de una imagen digitalizada.

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

Hoy en día el condicionante principal a la hora de afrontar cualquier proyecto basado en SIG lo constituye la disponibilidad de datos geográficos del territorio a estudiar, mientras que hace diez años lo era la disponibilidad de computadoras potentes, que permitieran realizar los procesos de cálculo involucrados en el análisis de datos territoriales.

Pero, además de ser un factor limitante, la información geográfica es a su vez el elemento diferenciador de un SIG frente a otro tipo de SI; así, la particular naturaleza de este tipo de información contiene dos vertientes diferentes: por un lado está la vertiente espacial y por otro la vertiente temática de los datos. Otros SI contienen sólo datos alfanuméricos (nombres, direcciones, números de cuenta). Las bases de datos de un SIG integran además la delimitación espacial de cada uno de los objetos geográficos.

Por tanto, el SIG tiene que trabajar a la vez con ambas partes de la información: su topografía perfectamente definida en plano y sus atributos temáticos asociados. Es decir, tiene que trabajar con cartografía y con la base de datos a la vez, uniendo ambas partes y constituyendo con todo ello una sola base de datos geográfica.

De esta manera, se define a la topología, como la capacidad de asociación de bases de datos temáticas junto con la descripción espacial precisa de objetos geográficos y las relaciones entre ellos. Es precisamente la topología lo que diferencia a un SIG de otros sistemas informáticos de gestión de información.

1.3.1 Álgebra de mapas

Los SIG presentan utilidades básicas tales como la codificación y almacenamiento de información espacial en formato raster, así como los fundamentos de la visualización y consulta de dicha información. Pero más allá de estas utilidades básicas, uno de los usos fundamentales de los SIG es la obtención de nuevas capas de información a partir de otras previamente disponibles. Para ello, se dispone de un conjunto de herramientas de cálculo con matrices de datos que reciben el nombre genérico de álgebra de mapas. Estas operaciones se hacen mejor en capas raster por su estructura regular, permitiendo mayor facilidad en la aplicación de algoritmos y formulaciones, lo que no implica que no se pueda aplicar álgebra de mapas en capas vectoriales.

Tipo de funciones

Dentro del álgebra de mapas se encuentran cuatro tipos de funciones definidas por como toman la información de las celdas de la capa inicial necesaria para su cálculo:

1. Local: Como el nombre lo indica, el resultado de la información de la nueva celda es en función de la celda inicial, es decir, la capa resultante va a tener en cada celda el valor en función del valor de la celda en la capa inicial.
2. Focal: La información de las celdas en la capa resultante va a ser en función de la celda correspondiente a la capa inicial y de las situadas en un entorno definido alrededor de la misma.
3. Zonal o Regional: El valor de cada una de las celdas de la capa resultante es función del valor de todas las celdas, que están conectadas con ellas, que tengan un mismo valor para una de las capas de entrada.²
4. Global: El valor final de la función se obtiene a partir de todas las celdas de la capa inicial.

El álgebra de mapas brinda las herramientas necesarias para analizar capas raster y hacer con ellas diversas combinaciones, utilizando las diferentes funciones, que permiten obtener información resultante derivada de la capa inicial o de varias capas en general. Esta herramienta define los procesos con los cuales desarrollar diversos análisis. Es la base de la aplicación de algoritmos a los análisis de las capas raster, por lo que se debe tener bien definido las variables de entrada (mapas) y las operaciones que se realizan en las diferentes celdas que son objeto de análisis.

Descripción de modelos matemáticos.

1. Definición de USLE

USLE³ es una función sumamente sencilla que multiplica las diferentes capas de partida, pero que también puede realizar operaciones más complejas, utilizando todos los operadores disponibles. Tiene que ser fácil de resolver e incluir solo factores cuyo valor en un lugar particular se pueda determinar a

² Las celdas cuyo valor pertenezcan a la misma clase que la celda problema.

³ **USLE** acrónimo de Ecuación de Pérdida de Suelo.

partir de los datos disponibles. Algunos detalles y perfeccionamientos posibles se sacrificarán en aras de la utilidad.

Esta ecuación viene dada por la siguiente expresión:

$$A = R * K * L * S * C * P$$

Donde las variables son:

- **A:** es la media de la pérdida anual de suelo en toneladas por hectáreas.
- **R:** es una medida de las fuerzas erosivas de las precipitaciones y la escorrentía.
- **K:** es el factor de erosionabilidad del suelo, es decir, una cifra que refleja la susceptibilidad de un tipo de suelo a la erosión o sea la recíproca de la resistencia del suelo a la erosión.
- **L:** es el factor de longitud, una relación que compara la pérdida de suelo con la de un campo de una longitud específica de 22,6 metros.
- **S:** es el factor de manejo, relación que compara la pérdida de suelo con la de un campo de pendiente específica del 9%.
- **C:** es un factor de manejo de los cultivos, relación que compara la pérdida de suelo con la de un campo sometido a un tratamiento estándar de barbecho.
- **P:** es factor de la práctica de conservación, una relación que compara la pérdida de suelo con la de un campo al que no se le aplica ninguna práctica de conservación, es decir, el tipo de arado y el sentido de la pendiente.

Tras el cálculo de todas las partes de la ecuación de USLE, esta se resuelve con una sencilla álgebra de mapas que multiplica todos los factores hasta aquí descritos. USLE es un modelo sencillo que combina cinco variables (las variables L y S se pueden definir de forma conjunta dando así la topografía), todas ellas susceptibles de ser recogidas en los correspondientes mapas. Si se extiende el cálculo puntual de la variable A a todos los puntos del mapa, se obtiene un nuevo mapa de dicha variable, para realizar esta cartografía, después de aplicar el modelo USLE se hace una reclasificación. **(11)**

2. Filtrado de capas

El filtrado de capas actúa moviendo una ventana, generalmente de 3x3 celdas, por toda la capa que recorre la capa entera. Esta ventana adjudica a la celda central la media ponderada de los valores en las 9 celdillas que abarca la ventana. A partir de diferentes coeficientes de ponderación se consiguen diferentes resultados.

La siguiente ecuación calcula la media aritmética.

$$Zm_{x,y} = \frac{Z_{x-1,y-1} + Z_{x-1,y} + Z_{x-1,y+1} + Z_{x,y-1} + Z_{x,y} + Z_{x,y+1} + Z_{x+1,y-1} + Z_{x+1,y} + Z_{x+1,y+1}}{9}$$

Utilizando diferentes coeficientes de ponderación se puede obtener filtros de distintos tipos. Estos tipos de operadores se utilizan especialmente en el análisis de imágenes de satélite. **(12)**

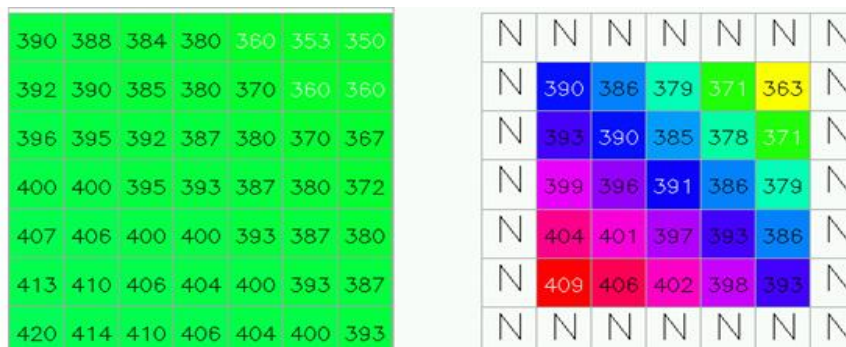


Figura 2 Representación del cálculo por filtrado de capas.⁴

El filtrado en los Modelos Digitales de Elevación (MDE) sirve a dos propósitos: suavizar o realzar los modelos, así como reducir datos. El suavizado se utiliza para eliminar detalles o hacer en general la superficie más suave. El realce, en cambio, resalta las discontinuidades, y elimina las zonas suaves.

1.3.2 Análisis hidrológico

Un modelo hidrológico es una representación simplificada del sistema real cuyo objetivo es estudiar la operación del sistema y predecir su salida. Sus entradas y salidas son variables hidrológicas y su estructura es un conjunto de ecuaciones que conectan las entradas con las salidas, las cuales pueden

⁴ A la izquierda la matriz a calcular la media aritmética y a la derecha el resultado después de aplicar la fórmula matemática.

expresarse como función del tiempo. Abarcan una gran diversidad de problemas y funcionalidades tales como modelado de ríos y cuencas, calidad de aguas, predicción de crecidas y riesgos hidrológicos. **(9)**

El mismo se ha desarrollado para suplir la falta de datos acerca de la cantidad, calidad o distribución en el tiempo del flujo de agua en cuencas o sectores de cuencas hidrográficas y para obtener un nivel de comprensión de los procesos hidrológicos inherentes, que permita pronosticar hidrogramas de salida a partir de datos climáticos (precipitación, evaporación) y de diferentes parámetros físicos de la cuenca (topografía, suelos, vegetación).

Existe una gran cantidad de programas de simulación hidrológica, que tienden a integrar diferentes procesos que anteriormente se estudiaban por separado y combinar la modelización hidrológica con los SIG.

Dentro de los modelos hidrológicos se encuentra GIS-BALAN: este es un modelo hidrológico semidistribuido ampliamente utilizado como herramienta para la evaluación de los recursos hídricos que se ha acoplado a un SIG. El mismo resuelve el balance hidrológico en el suelo edáfico, en la zona no saturada y en el acuífero evaluando secuencialmente las componentes.

Descripción de modelos matemáticos.

1. Cálculo de flujo

El proceso de construcción de una línea de flujo a partir de un punto inicial, es interactivo y consta de tres fases fundamentales.

1. Se fija el punto inicial de la línea. P (i)
2. Se calculan las pendientes hacia sus 8 vecinos más próximos, se pueden dar tres casos:
 - a) Todas las pendientes son negativas (se trata de una concavidad): fin de la línea.
 - b) Todas las pendientes son negativas y el punto está en el borde del MDE (la cuenca continua probablemente fuera de los límites del MDE): fin de la línea.
 - c) Se localiza al menos un punto con pendiente positiva: se elige el punto con pendiente máxima.
3. Punto elegido (pendiente máxima) se incorpora a la línea de flujo y se toma como base para resolver el paso 2. **(13)**

2. Modelo Spline

Bilineal

Se conoce como modelos de Spline aquellos donde, a partir de los valores (X_i, Y_i, Z_i) de un conjunto de puntos se realiza la interpolación de los valores de “Z” de aquellos puntos que se encuentran en la región interior de la poligonal determinada por los datos originales, particularmente, el Spline bilineal refiere a los puntos contenidos dentro de un cuadrilátero que puede formar parte de una rejilla.

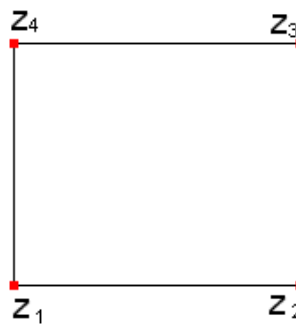


Figura 3 Representación de una cuadrícula que forma parte de una rejilla.

En tal caso la función es bilineal y toma la forma.

$$Z = A + Bx + Cy + D_{xy} \quad (\text{Ecuación 1})$$

Para “x” y “y” comprendidas en el intervalo de 0-1 y se exige que:

$$\begin{aligned} x=0, y=0: Z=Z_1 & \quad x=1, y=1: Z=Z_3 \\ x=1, y=0: Z=Z_2 & \quad x=0, y=1: Z=Z_4 \end{aligned} \quad (\text{Ecuación 2})$$

Con estos elementos pueden calcularse los valores A, B, C y D para el Spline de cada cuadrícula, siendo:

$$A=Z_1 \quad B=Z_2-Z_1 \quad C=Z_4-Z_1 \quad D=Z_3-Z_4+Z_1-Z_2 \quad (\text{Ecuación 3})$$

Permite trazar las líneas de nivel del terreno. En primer lugar, escoger el Spline bilineal garantiza que haya continuidad en el relieve para cuadrículas contiguas, además que ninguna cuadrícula pase una línea cerrada (no tiene ni máximo ni mínimo), esto se corresponde con la forma funcional de (Ecuación 1) e implica una limitación para la cuadrícula seleccionada. **(14)**

3. Cálculo de la pendiente de una cuenca según el método de Horton.

Con frecuencia basta con medir la pendiente media del cauce principal, pero en ocasiones se necesita calcular la pendiente media de toda la superficie de la cuenca. Se representa una cuenca, a la que se ha superpuesto una cuadrícula regular⁵.

Para el método de Horton según Viessman se siguen los siguientes pasos:

- Se cuentan los puntos de intersección de las líneas verticales con cualquier curva de nivel.
- Luego se mide la longitud de los tramos verticales de la rejilla dentro de los límites de la cuenca.
- Se calcula mediante la siguiente fórmula la pendiente vertical:

$$P_v = \frac{n \cdot e}{\sum L_v}$$

Donde:

n : número de intersecciones con la cuenca.

e : distancia entre las curvas de nivel.

$\sum L_v$: sumatoria de las longitudes verticales de la cuadrícula.

- Se calcula la pendiente de la misma forma pero horizontalmente.
- Luego se calcula la pendiente de la cuenca realizando la media.

$$P_m = \frac{P_v + P_h}{2} \quad (13)$$

1.3.3 Análisis Geoestadístico

La geoestadística permite el estudio de los fenómenos naturales, considerando la dependencia espacial que se presenta entre observaciones. Las técnicas de interpolación se basan en que los valores de puntos más cercanos sean más similares entre sí, que con valores de puntos más distantes. Este principio es usado para predecir valores en áreas no muestreadas. Estas estimaciones pueden ser calculadas usando

⁵ Menor espaciado de la cuadrícula dará mayor precisión, pero también más trabajo.

las siguientes formulas. (21)

1.3.4 Descripción de modelos matemáticos.

1. Kriging

El objetivo del Kriging es lograr el mínimo error de estimación. El Kriging es una combinación lineal ponderada de los datos en la vecindad a la ubicación a estimar. (21)

$$Z^*(u) = \sum_{i=1}^n \lambda_i Z(u_i)$$

Donde:

U: representa la ubicación espacial

$Z^*(u)$: es el valor estimado en esa ubicación.

Las variables $Z(u_i) \ i = 1..n$: corresponden a las leyes de las muestras en la vecindad de u .

λ_i : son los pesos atribuidos a cada muestra.

2. Cálculo del Variograma

Variograma es un estimador de la varianza poblacional, por lo tanto la población debe tener una tendencia de estacionalidad; el variograma está relacionado con una dirección y distancia (h), es la principal herramienta básica, que da soporte a las técnicas de Kriging, permite representar cuantitativamente la variación de un fenómeno regionalizado en el espacio. (21)

El variograma es definido por varios autores como:

Si $Z(x)$ es estacionaria o intrínseca

$$\gamma(h) = \frac{1}{2} \text{Var}[Z(x) - Z(x+h)] \quad x, h \in \mathcal{R}^n$$

$$\gamma(h) = \frac{1}{2} \sum [Z(x) - Z(x+h)]^2$$

- γ es independiente de la localización x
- γ depende del módulo y de la dirección del vector h

En general, las técnicas geostatísticas (Kriging y Variograma) presentan mejores resultados que las técnicas de interpolación tradicionales (Spline y Distancia Inversa). El Kriging es considerado el mejor estimador, genera los resultados más próximos a las observaciones reales.

1.4 Análisis de Soluciones Existentes

1.4.1 Sextante

Sextante es una biblioteca de algoritmos de análisis espacial de código libre⁶ disponible para varios software SIG. Su objetivo principal es crear una plataforma que facilite tanto el uso como la implementación de estos algoritmos. Actualmente Sextante contiene más de 240 herramientas de análisis geográfico.

Sextante está programado en Java. Desarrollado por la Universidad de Extremadura (UNEX) y financiado por la Junta de Extremadura que se distribuye bajo la licencia GPL cumpliendo con los cuatro principios necesarios para clasificarlo como software libre.

Algunas de sus utilidades son:

- Análisis hidrológico básico.
- Estadísticas de celda para múltiples capas raster.
- Estadísticas por vecindad para una capa raster.

⁶ El software libre suele estar disponible gratuitamente, o al precio de costo de la distribución a través de otros medios; sin embargo no es obligatorio que sea así, por lo tanto no hay que asociar software libre a "software gratuito".

- Geomorfometría y análisis del relieve.
- Herramientas básicas para capas raster.
- Herramientas de análisis para capas raster.
- Herramientas de cálculo para capas raster.
- Herramientas para capas de líneas.
- Herramientas para capas de puntos.
- Herramientas para capas raster categóricas.
- Herramientas para capas discretas e información categórica.
- Herramientas para capas vectoriales.
- Herramientas para crear nuevas capas raster.
- Rasterización e interpolación.
- Tratamiento y análisis de imágenes.
- Índices y otros parámetros hidrológicos.

Hoy existen adaptaciones de la misma para Geotools, gvSIG y OpenJump y están en desarrollo las versiones para Kosmo, OrbisGIS. **(16)**

1.4.2 GRASS

GRASS incluye módulos para el manejo de información en formato raster, vectorial, mapas de puntos (*sites*) y raster tridimensional (*voxels*). Permite la importación de un gran número de formatos de datos raster o vectorial. La siguiente tabla muestra las capacidades del sistema. **(17)**

Tabla 1 Principales potencialidades de GRASS.

Categoría	Módulos
Análisis raster	Conversión a formato vectorial de líneas o polígonos raster
	<i>Buffering</i> sobre estructuras lineales

	Consultas por celdillas y por perfiles
	Modificación de la tabla de colores
	Conversión a formato vectorial y punto de datos
	Análisis de correlación/covarianza
	Sistemas expertos
	Álgebra de mapas (r.mapcalc)
	Interpolación de datos faltantes
	Análisis de vecindad
	Superposición raster con o sin peso
	Reclasificación de las etiquetas de las celdillas
	Remuestreo (cambio de resolución)
	Reescalado de los valores de las celdillas
	Análisis estadístico de celdillas
	Generación de superficies a partir de líneas codificadas en formato vectorial
Análisis Raster 3D (<i>voxel</i>)	Importación de datos 3D en formato ASCII
	Máscaras 3D
	Álgebra de mapas en 3D (r3.mapcalc)
	Interpolación 3D (IDW, Splines regularizados con parámetro de tensión)
	Visualización 3D (isosuperficies)
	Interfaz con la herramienta de visualización Vis5D

Análisis vectorial	Generación de isolíneas a partir de superficies raster (IDW, algoritmo de splines)
	Conversión a formato raster y mapa de puntos
	Digitalización con tableta o sobre imágenes raster en pantalla con el ratón
	Reclasificación de las etiquetas en vectorial
	Superposición de capas vectoriales
Análisis de datos puntuales	Triangulación de Delaunay
	Interpolación de superficies a partir de datos puntuales
	Polígonos de Thiessen
	Análisis topográfico (curvatura, pendiente, orientación)
Teledetección	Análisis de componentes canónicos (CCA)
	Generación de composiciones de color
	Detención de bordes
	Filtrado de frecuencias (Fourier, matrices de convolución)
	Transformación de Fourier e inversa de Fourier
	Ajuste del histograma
	Transformación IHS – RGB
	Corrección geométrica (transformación polinomial y afín sobre raster o vectorial)
	Creación de Ortofotos
	Análisis de Componentes Principales (PCA)

	Corrección Radiométrica (Fourier)
	Remuestreo
	Mejora de la resolución (con RGB/IHS)
	RGB a IHS transformación
	Clasificación por textura (clasificación secuencial máxima a posteriori)
	Detección de formas
	Clasificación supervisada (aéreas de entrenamiento y clasificador de máxima verosimilitud)
	Clasificación no supervisada (agrupación de clases <i>clustering</i> por mínima distancia y clasificador de máxima verosimilitud)
Análisis de MDT	Generación de curvas de nivel
	Análisis de rutas óptimas
	Análisis de pendiente y orientación
	Generación de superficies a partir de altitudes o curvas de nivel
Visualización	Superficies 3D con consulta
	Asignación de colores
	Histograma
	Superposición de mapas
	Mapas de datos puntuales
	Mapas raster
	Mapas vectoriales
	Funciones de Zoom

Creación de mapas	Mapas en formato PPM
	Mapas en formato PostScript
	HTML mapas
SQL	Enlace a bases de datos (PostgreSQL)
Otros módulos	Análisis de estructura y ecología del paisaje
	Transporte de solutos
	Análisis hidrológico

1.4.3 KOSMO

Kosmo es un SIG de escritorio de funcionalidades avanzadas. Es el primer componente de una serie de desarrollos que están en marcha y que, a partir de ahora, serán puestos a disposición de toda la comunidad.

Kosmo se ha implementado utilizando el lenguaje de programación Java y desarrollado a partir de la plataforma JUMP, ha sido empleado en multitud de proyectos de software libre de reconocido prestigio entre las que se destacan Geotools y JTS. Presenta varias funcionalidades como se describe en la Tabla 2. (18)

Tabla 2 Principales características de KOSMO.

Nombre herramienta	Kosmo
Licencia	GNU/GPL
Lenguaje de desarrollo	Java
Arquitectura	Arquitectura basada en la gestión y análisis de la información territorial a través de Bases de Datos Espaciales, dotándolo así de carácter Corporativo.
Funcionalidades	Capacidad de acceder a múltiples formatos de datos, tanto vectoriales (en fichero, como Shapefile o DXF, o en base de datos, como PostgreSQL,

	MySQL u Oracle), como raster (TIFF, GeoTIFF, ECW, MrSID u otros formatos de imagen georreferenciados, como BMP, GIF, JPG, PNG).
--	---

Después de haber analizado algunas de las soluciones existentes y sus características se puede llegar a la conclusión que la herramienta más completa a estudiar es Sextante. Pues cuenta con algunas de las funcionalidades que deben contener el módulo a diseñar.

1.5 Argumentar el uso de la Metodología de Desarrollo

El desarrollo de software no es una tarea fácil. Como resultado a este problema ha surgido una alternativa que son las Metodologías de Desarrollo de Software. Las metodologías imponen un proceso disciplinado sobre el desarrollo de software con el fin de hacerlo más predecible y eficiente. Lo hacen desarrollando un proceso detallado con un fuerte énfasis inspirado por otras disciplinas de la ingeniería.

Son características deseables de una metodología las siguientes:

- Cobertura total del ciclo de desarrollo.
- Fácil formación.
- Utilización de herramientas CASE.
- Comunicación efectiva.
- Soporte al mantenimiento.
- Soporte a la reutilización de Software.
- Actividades que mejoren el proceso de desarrollo.
- Existencia de reglas predefinidas.

Si se tiene en cuenta su filosofía de desarrollo, entonces se pueden definir dos fuertes corrientes de metodologías: Metodologías Tradicionales, conocidas también como metodologías pesadas, las cuales ponen especial énfasis en la planificación y control del proyecto, así como en la especificación precisa de requisitos y el modelado; estas han demostrado ser efectivas y necesarias en un gran número de proyectos, sobre todo aquellos proyectos de gran tamaño respecto a tiempo y recursos que emplean. El otro grupo denominado por Metodologías Ágiles, dirigidas sobre todo a equipos de desarrollo pequeños,

con mayor fuerza en aspectos humanos asociados al trabajo en equipo e involucran al cliente en el proceso como parte activa del propio equipo de desarrollo y están orientadas sobre todo a la generación de código con ciclos cortos de desarrollo.

La diferencia más notable entre estos dos grupos es que, mientras los métodos pesados intentan obtener los resultados apoyándose principalmente en la documentación ordenada. Los métodos ligeros o ágiles tienen como base de sus resultados la comunicación e interacción directa con todos los usuarios involucrados en el proceso.

Las propuestas más tradicionales como son las metodologías rígidas tienen de forma general las siguientes características:

- Control del proceso, estableciendo rigurosamente las actividades involucradas, los artefactos que se deben producir, y las herramientas y notaciones que se usarán.
- Basadas en normas provenientes de estándares seguidos por el entorno de desarrollo.
- Cierta resistencia a los cambios.
- Proceso mucho más controlado, con numerosas políticas/normas.
- El cliente interactúa con el equipo de desarrollo mediante reuniones.
- Grupos grandes y posiblemente distribuidos.
- Más artefactos y roles.
- La arquitectura del software es esencial.

Por su parte las metodologías ágiles o flexibles tienen las siguientes características:

- Aplican heurísticas provenientes de prácticas de producción de código.
- Especializadas para realizar cambios durante el proyecto.
- Proceso menos controlado, con pocos principios.
- No existe contrato tradicional o al menos es bastante flexible.
- El cliente es parte del equipo de desarrollo.

- Grupos pequeños (<10 integrantes) y trabajando en el mismo sitio.
- Pocos artefactos.
- Pocos roles.
- Menos énfasis en la arquitectura del software.
- Poca documentación.

1.5.1 El Proceso Unificado de Desarrollo de Software (RUP) como metodología propuesta.

Dentro de las metodologías pesadas o tradicionales la que más se destaca es RUP. Consiste en un proceso de desarrollo de software y junto con el Lenguaje Unificado de Modelado UML, constituye la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos. No es un sistema con pasos firmemente establecidos, sino un conjunto de metodologías adaptables al contexto y necesidades de cada organización.

RUP es apropiada para proyectos grandes, requiere de un equipo de trabajo capaz de administrar un proceso complejo en varias etapas, guiando a cada trabajador durante todo el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue (figura 4). Se definen como los principales elementos de RUP: actividades, artefactos, trabajadores y flujos de trabajo; donde los trabajadores están encargados de definir el comportamiento y las responsabilidades (rol) de un individuo, grupo de individuos, sistema automatizado o máquina, que trabajan en conjunto como un equipo. Ellos realizan las actividades y son propietarios de elementos. Se escoge RUP como Metodología de Desarrollo de software para el análisis del componente para el álgebra de mapas, análisis hidrológico y geoestadístico, debido a tres características fundamentales que presenta: son dirigidos por casos de uso, iterativo e incremental y centrado en la arquitectura.

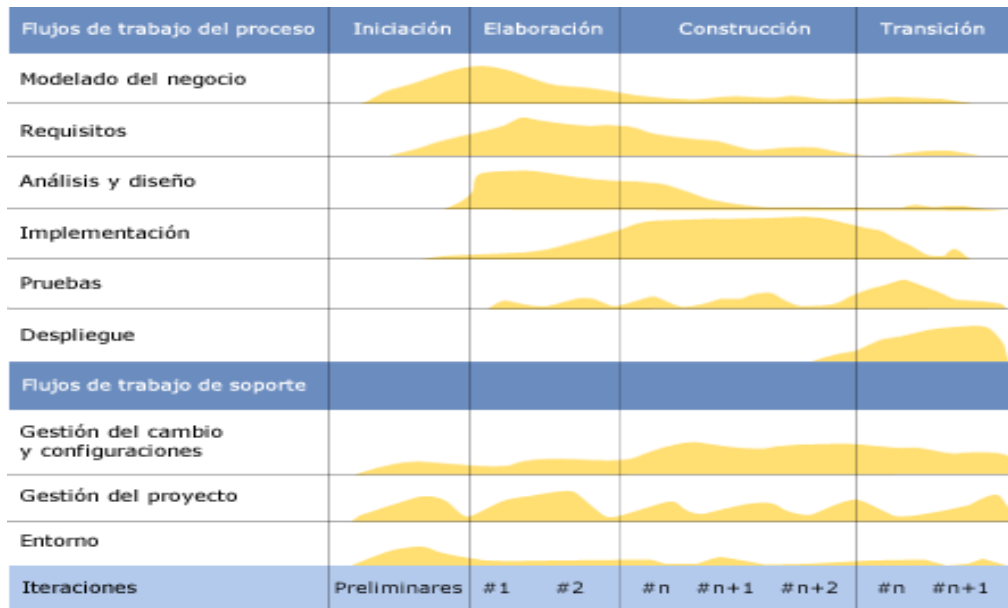


Figura 4: Etapas y flujos de trabajo de RUP

Los casos de uso además de ser una herramienta para especificar los requisitos de un sistema guían el diseño, implantación y prueba del proceso de desarrollo. Al basarse en el Modelo de Casos de Uso, los desarrolladores crean una serie de modelos de diseño e implantación que describen a cada uno de los casos de uso. Los desarrolladores revisan cada uno de los sucesivos modelos para que sean conforme a este, los ingenieros de prueba prueban la implantación para garantizar que los componentes del modelo de implantación implementen correctamente los casos de uso. De este modo, los casos de uso no solo inician el proceso de desarrollo sino que le proporcionan un hilo conductor que avanza a través de una serie de flujos de trabajo que parten de estos. Se especifican, se diseñan y son útiles en la construcción final de los casos de prueba.

Entre los casos de uso y la arquitectura debe existir interrelación, estos deben encajar cuando se lleva a cabo el proceso de desarrollo, la arquitectura debe permitir la elaboración de los casos de uso requeridos. Deben avanzar en paralelo, la arquitectura modela el sistema para darle una forma, debe diseñarse para permitir que el sistema evolucione, no solo en su desarrollo inicial, sino también a lo largo de las futuras generaciones mediante la comprensión general de las funciones claves. Con la arquitectura se obtiene la estructura que guía las iteraciones, mientras que los casos de uso definen los objetos y dirigen el trabajo de la iteración. (13)

Se escoge RUP como metodología de desarrollo porque los trabajadores involucrados en el desarrollo de GeneSIG tienen un amplio dominio en su uso, y al ser basado en casos de usos se logra una mayor comprensión y descripción de cada una de las funcionalidades del componente. Además el equipo de trabajo, conocen y comparten el proceso de desarrollo utilizando un lenguaje de modelado común: UML y que permite asegurar que la producción de un software sea de alta calidad y que cumpla con las necesidades del usuario final.

Por ser abarcadora en la guía de procesos de desarrollo de software y además por ser completa en la determinación de la documentación ingenieril ha sido RUP la más completa para la realización de este trabajo. Es significativo destacar que esta investigación solo comprende las disciplinas de Modelamiento del negocio y Requisitos, lo cual trae consigo que los flujos de trabajo de Análisis y Diseño, Implementación y Prueba tienen que seguir desarrollándose y para ello se necesita que el equipo de desarrollo que continúe trabajando en el desarrollo de esta aplicación tenga una documentación exhaustiva del negocio y sistema de la aplicación en cuestión y precisamente esta metodología provee toda la documentación completa y exacta para la realización del componente.

1.6 Lenguaje de Modelado

UML es un lenguaje de modelado visual utilizado para especificar, visualizar, construir y documentar artefactos del sistema de un software. Se usa para entender, diseñar, configurar, mantener y controlar la información sobre los sistemas a construir. El mismo capta la información sobre la estructura estática y el comportamiento dinámico de un sistema. Un sistema se modela como una colección de objetos discretos que interactúan para realizar un trabajo que finalmente beneficia a un usuario externo. El lenguaje de modelado pretende unificar la experiencia basada sobre técnicas de modelado e incorporar las mejores prácticas actuales en un acercamiento estándar. **(12)**

UML 1.0 tiene las siguientes características:

- Permite modelar sistemas utilizando técnicas orientadas a objetos (OO).
- Permite especificar todas las decisiones de análisis, diseño e implementación, construyéndose así modelos precisos, no ambiguos y completos.
- Puede conectarse con lenguajes de programación (Ingeniería directa e inversa).

- Permite documentar todos los artefactos de un proceso de desarrollo (requisitos, arquitectura, pruebas, versiones.).
- Cubre las cuestiones relacionadas con el tamaño propio de los sistemas complejos y críticos.
- Es un lenguaje muy expresivo que cubre todas las vistas necesarias para desarrollar y luego desplegar los sistemas.
- Existe un equilibrio entre expresividad y simplicidad, pues no es difícil de aprender ni de utilizar.
- Es independiente del proceso, aunque para utilizarlo óptimamente se debe usar en un proceso que fuese dirigido por los casos de uso, centrado en la arquitectura, iterativo e incremental.

1.7 Visual Paradigm como herramienta de modelado propuesta

Las herramientas para el modelado de diagramas según el lenguaje UML han evolucionado, no solo en el sentido de ir renovando las versiones a las que dan sustento, sino que a su vez constituyen un verdadero apoyo en las tareas de desarrollo de software. Se le han realizado mejoras a la interfaz, facilidades como chequeos sintácticos de los diagramas, chequeos de consistencia, facilidades de importación y exportación, la generación de código en diferentes lenguajes, uso de ingeniería inversa para obtener diagramas a partir de código, hasta la posibilidad de definir perfiles para el diseño en un dominio específico.

Visual Paradigm, es una herramienta CASE de modelado que está desarrollada por Visual Paradigm Internacional, una de las principales compañías de herramientas CASE. Su mayor éxito consiste en la capacidad de ejecutarse sobre diferentes sistemas operativos lo que le confiere la característica de ser multiplataforma. Visual Paradigm utiliza UML 1.0 como lenguaje de modelado ofreciendo soluciones de software que permiten a las organizaciones desarrollar las aplicaciones de calidad más rápido y más barato. Es muy fácil de usar y presenta un ambiente gráfico agradable para el usuario. Su notación es muy parecida a la estándar, permite configurar las líneas de redacción, el modelado de base de datos, de requisitos y del proceso de negocio, la interoperabilidad, la generación de documentación y la generación de código base para diferentes lenguajes de programación, además de permitir la integración con herramientas de desarrollo. **(14)**

Se escoge como herramienta Case a Visual Paradigm 3.4 por las siguientes razones:

- Se ajusta a las características de la universidad por ser precisamente ejecutable en entornos libre.
- Multiplataforma.
- Fácil de instalar, utilizar y actualizar.
- Es una herramienta profesional que soporta el ciclo de vida completo del desarrollo de software.
- Disponible en varios idiomas.
- Permite el control de versiones.
- Tiene características gráficas muy cómodas que facilitan la realización de los diagramas de modelado que sigue el estándar de UML.

1.8 Conclusiones parciales

Con la investigación realizada en este capítulo se arrojó a la conclusión de seleccionar RUP como metodología de desarrollo que por su adaptabilidad y documentación generada se adapta a las características y necesidades de la presente investigación, UML 1.0 como lenguaje de modelado para generar los artefactos correspondientes a cada una de las etapas, obteniéndose los siguientes artefactos: Glosario de términos, Modelo de dominio, Especificación de requisitos de software y Diagrama de Casos de Uso de Sistema. Se utilizó Visual Paradigm como herramienta CASE para modelar dichos artefactos.

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

2.1 Introducción

En este capítulo se muestra una descripción de la propuesta a defender, con vista a solucionar la situación actual en el campo de acción. Se muestran los requisitos, tanto los funcionales como los no funcionales, y los casos de uso generados a partir de dichos requisitos funcionales con sus respectivas especificaciones. Esta propuesta se elabora con el fin de facilitar el trabajo con el análisis hidrológico, álgebra de mapas y análisis geoestadístico, y agilizar el desarrollo de aplicaciones informáticas que requieran el uso de estos.

2.2 Modelo de Dominio

Al realizarse un estudio profundo del problema al que se pretende brindar una solución, no fue posible identificar con claridad los diferentes procesos del negocio en que se desarrolla el problema existente; de igual modo no se pudo concretar con seguridad cuáles eran los actores implicados en el negocio, los trabajadores y las actividades que éstos últimos realizan. Por tal motivo se le atribuyó mayor importancia a la información que puede brindar el modelo de dominio.

El modelo de dominio se obtiene partiendo de un conjunto de especificaciones iniciales brindadas por los analistas. Con esta información, de carácter preliminar, los grupos de desarrollo pueden simplificar de manera notable algunos aspectos de las fases de modelación iniciales, que permitan aprovechar el trabajo y las experiencias previas y aceleren por vías probadas el proceso total de elaboración de software.

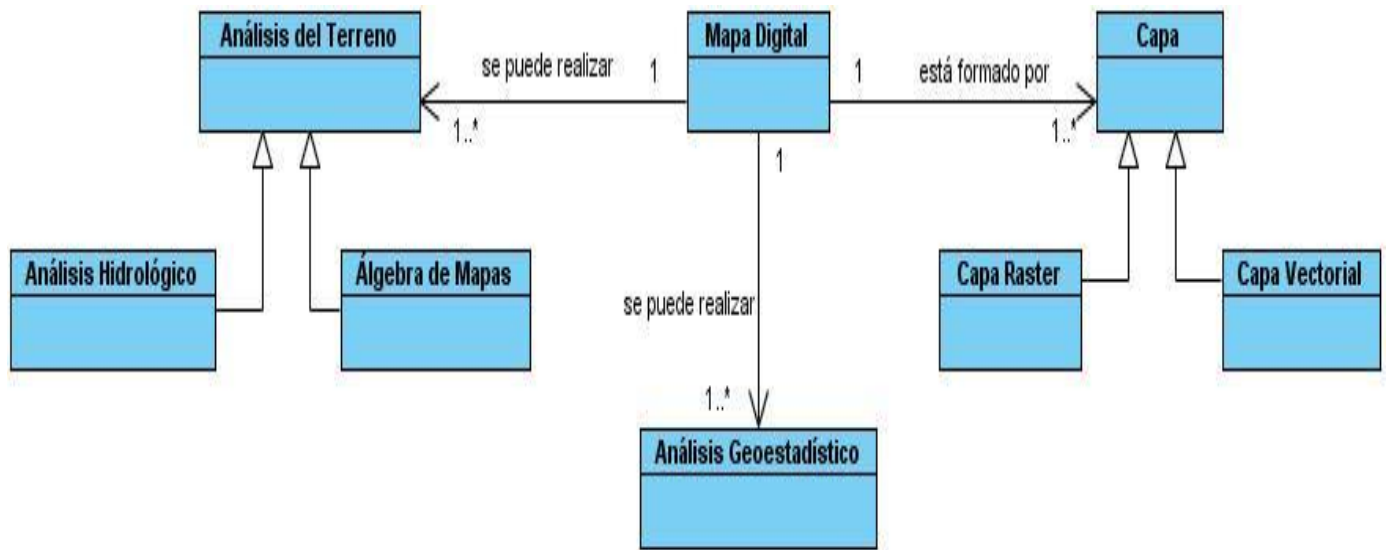


Figura 5 Modelo de Dominio.

Descripción del Diagrama

Mapa Digital: representación gráfica de una ciudad, mediante diferentes tipos de geometrías extraídos de imágenes fotográficas, las cuales reflejan la realidad del paisaje distrital.

Análisis Hidrológico: abarca una gran diversidad de problemas y funcionalidades tales como modelado de ríos y cuencas, calidad de aguas, predicción de crecidas y riesgos hidrológicos.

Álgebra de Mapas: es la obtención de nuevas capas de información a partir de otras previamente disponibles.

Análisis Geoestadístico: permite describir la continuidad espacial de las variables y estimar valores muy cercanos a los reales en puntos desconocidos.

Capa Raster: modelo de datos en el cual las entidades geográficas son representadas usando celdas.

Capa Vectorial: la información es representada sobre puntos, líneas y polígonos.

A través de un Mapa Digital se puede realizar análisis que puede ser del tipo Análisis Geoestadístico y Análisis del Terreno. En este último se engloban los tipos Álgebra de Mapa y Análisis Hidrológico. Dicho mapa está formado por una o varias Capas que pueden ser del tipo Raster o Vectorial.

2.3 Descripción de las funcionalidades del sistema

Se aborda una visión general del problema que se está resolviendo y las áreas claves de la funcionalidad que se deben tratar en la solución. En el mismo se capturan los requisitos funcionales y no funcionales que el sistema debe poseer lo que posibilita a desarrolladores y clientes un entendimiento común.

2.3.1 Requisitos Funcionales

Los requisitos funcionales (RF), son capacidades o condiciones que el sistema debe cumplir. Siendo estos para el componente para el análisis hidrológico, álgebra de mapas y análisis geoestadístico en sistemas de información geográfica:

RF 1 Obtener direcciones de flujo.

RF 2 Calcular la red de drenaje.

RF 3 Calcular pendiente.

RF 4 Obtener orientación.

RF 5 Calcular curvaturas.

RF 6 Realizar Análisis de Relieve sombreado.

RF 7 Realizar análisis de visibilidad.

RF 8 Realizar rasterización a una capa vectorial.

RF 9 Interpolar por distancia inversa.

RF 10 Realizar vecindad.

RF 11 Realizar clasificación supervisada.

RF 12 Estimar índices de vegetación.

RF 13 Generar MDT artificial.

En el anexo 1 se pueden localizar las descripciones de todos los RF, con el fin de obtener una mejor comprensión de lo que el sistema debe hacer.

2.3.2 Requisitos No Funcionales

Los requisitos no funcionales apoyan a que clientes y usuarios puedan valorar las características no funcionales del producto, pues si se conoce que el mismo cumple con todas las funcionalidades requeridas; las propiedades no funcionales como, cuán usable, seguro, conveniente y agradable es el mismo, pueden marcar la diferencia entre un producto bien aceptado y uno con poca aceptación.

1 Usabilidad

- El componente podrá ser usado por personas con conocimientos básicos en el manejo de computadoras. Se emplearán componentes que indiquen al usuario el estado de los procesos que por su complejidad requieran de un tiempo de procesamiento apreciable.

2 Fiabilidad

- La herramienta de implementación a utilizar debe tener soporte para recuperación ante fallos y errores. La información manejada por el sistema estará protegida de acceso no autorizado y divulgación, verificándose la inyección de código sql.
- El tiempo medio de reparación, en caso de un fallo es de 7 días como mínimo.

3 Eficiencia

- El tiempo de respuesta será de 5 segundos como máximo.
- Al igual que el tiempo de respuesta, la velocidad de procesamiento de la información, la actualización y la recuperación dependerán de la cantidad de información que tenga que procesar la aplicación.

4 Soporte

- La aplicación recibirá mantenimiento en el período de tiempo determinado por el equipo de desarrollo y los clientes, se proponen tres meses como máximo.

5 Restricciones de diseño

- Diseño sencillo, con pocas entradas, donde no sea necesario mucho entrenamiento para utilizar el sistema.

- El producto de software final debe diseñarse sobre una arquitectura cliente-servidor.
- Se deben emplear los estándares establecidos (diseño de interfaces, base de datos y codificación).
- Se debe lograr un producto altamente configurable y extensible.

6 Interfaz

Interfaces de usuario

El componente debe tener una apariencia profesional y un diseño gráfico sencillo; posibilitarle al usuario la configuración del entorno de trabajo y ser intuitivo. A continuación se proponen las interfaces de hardware y software con las mismas características del sistema que utilizará el mismo.

Interfaces de hardware

Para las PCs clientes:

- Se requiere tengan tarjeta de red.
- Al menos 256 MB de memoria RAM.
- Se requiere al menos 100 MB de disco duro.
- Procesador 512 MHz como mínimo.

Para los servidores:

- Se requiere tarjeta de red.
- El Servidor de Mapas tenga como mínimo una capacidad de 2GB de RAM y 2GB de disco duro.
- El Servidor de BD tenga como mínimo una capacidad de 2GB de RAM y 10GB de disco duro.
- Procesador 3 GHz como mínimo.

Interfaces de software

Para las PCs clientes:

- Un Navegador como Mozilla Firefox, Safari u otro navegador que cumpla con los estándares W3C.
- Sistema operativo: GNU/Linux, Windows y Mac OS.

Para los Servidores:

- Sistemas operativos GNU/Linux o Windows Server 2000 o superior.
- Servidor Web Apache 2.0 o superior, con módulo PHP 5 configurado con la extensión pgsql incluida.
- PostgreSQL como Sistema Gestor de Base de Datos.
- PostGis como extensión de PostgreSQL como soporte de datos espaciales.
- PgRouting como extensión de PostgreSQL para análisis de rutas.
- MapServer 5.2.2 o superior, con extensión PHP mapscript.

7 Requisitos para la documentación de usuarios en línea y ayuda del sistema.

El software tendrá siempre la posibilidad de ayuda disponible para cualquier tipo de usuario, lo que le permitirá un avance considerable en la explotación de la aplicación en todas sus funcionalidades.

2.4 Modelo de Casos de Uso del Sistema

2.4.1 Determinación y justificación de actores

Los actores del sistema son aquellas personas, entidades, sistemas o cualquier agente externo que interactúe con el sistema. Cada actor puede interactuar con uno o más casos de uso. En la Tabla 3 se identifican los actores del sistema y su descripción.

Tabla 3 Actores del Sistema

Actores	Descripción
Usuario	Es cualquier persona con conocimientos básicos de computación y de los Sistemas de Información Geográfica, encargada de seleccionar la opción que desea realizar.

Listado de los Casos de Usos

CUS 1: Realizar análisis hidrológico básico.

CUS 2: Realizar geomorfometría y análisis del relieve.

CUS 3: Realizar análisis de iluminación y visibilidad.

CUS 4: Realizar rasterización e interpolación.

CUS 5: Realizar análisis de imágenes.

CUS 6: Crear capa.

2.4.2 Diagrama de Casos de Uso del Sistema

Un diagrama de casos de uso del sistema representa gráficamente los procesos y su interacción con los actores. A continuación en la Figura 6 se muestra el diagrama de casos de uso perteneciente al sistema.

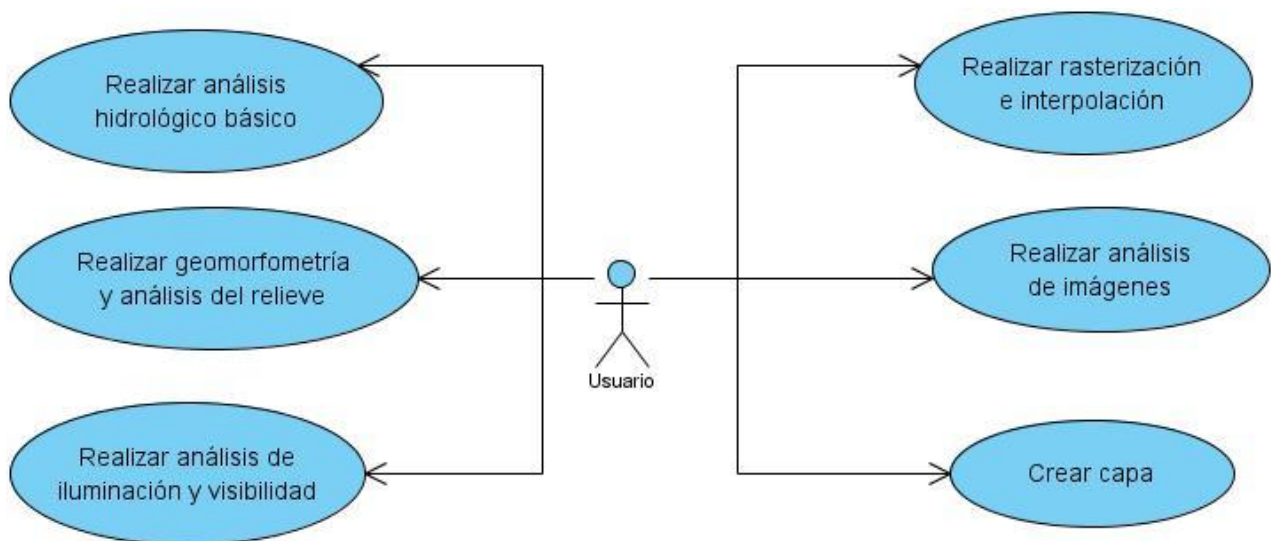


Figura 6 Diagrama de Casos de Uso del Sistema

2.4.3 Descripción textual de los Casos de Uso del Sistema

En el presente epígrafe se realizan las descripciones textuales de los casos de uso del sistema para lograr un mejor rendimiento del tiempo en la construcción del software. A continuación se describen los CUS Realizar análisis hidrológico básico y Realizar geomorfometría y análisis del relieve, en el Anexo 2 se encuentran Realizar análisis de iluminación y visibilidad, Realizar rasterización e interpolación, Realizar análisis de imágenes y Crear capa.

CUS Realizar análisis hidrológico básico

Caso de uso	Realizar análisis hidrológico básico.	
Actores	Usuario	
Propósito	Este caso de uso se lleva a cabo con el propósito de calcular el valor de la superficie situada aguas arriba de cada celda (área de todas las celdas cuyo flujo, una vez conducido aguas abajo, acabaría pasando por dicha celda). Generar nuevas capas con información de entidades, de tipo raster y vectorial.	
Resumen	El caso de uso se inicia cuando el usuario da clic en la opción realizar análisis hidrológico básico, seleccionando los parámetros necesarios para el análisis y termina cuando el sistema devuelve el resultado correspondiente a la elección del usuario.	
Precondiciones	Tener cargada al menos una capa raster.	
Referencias	RF 1, RF 2	
Prioridad	Crítico	
Flujo normal de eventos		
Acción del actor	Respuesta del sistema	
1. El caso de uso se inicia cuando el usuario selecciona la opción realizar análisis hidrológico básico.	2. El sistema visualiza una ventana para que el usuario seleccione una de las dos variantes: <ul style="list-style-type: none"> - Obtener direcciones de flujo. - Calcular la red de drenaje. Ver interfaz 1a.	
3. El usuario selecciona una de las dos opciones visualizadas.	4. El sistema realiza la operación según la opción seleccionada por el usuario: <ul style="list-style-type: none"> - Si selecciona la opción Obtener direcciones de flujo ir a la sección "Obtener direcciones de flujo". - Si selecciona la opción Calcular la red de 	

	drenaje ir a la sección “Calcular la red de drenaje”.
Sección “Obtener direcciones de flujo”	
Acción del actor	Respuesta del sistema
<p>3. El usuario selecciona la opción obtener direcciones de flujo y da clic en el botón cargar.</p>	<p>4. El sistema muestra una ventana Acumulación de Flujo con la siguiente estructura:</p> <p>Pestaña Parámetros</p> <p>Capas raster</p> <p>Con dos combobox uno para seleccionar el modelo MDE, y otro opcional para ponderar las celdas del modelo seleccionado.</p> <p>Opciones</p> <p>Con un combobox para seleccionar el método [que pueden ser D8, Rho8, DInfinity, MFD (Dirección de flujo múltiple)] y un textbox para definir el valor del factor de convergencia del método seleccionado (por defecto 1.1).</p> <p>Objetos de salida</p> <p>Con un file upload para guardar el archivo y un combobox para seleccionar la vista de salida.</p> <p>Pestaña Salida Raster para visualizar en el mapa la acumulación de flujo.</p> <p>Tres botones en la parte inferior Aceptar, Cancelar e Información.</p> <p>Ver interfaz 1b.</p>
<p>5. El usuario completa cada una de las opciones que se visualizan en la pestaña Parámetros y da clic en el botón Aceptar.</p>	<p>6. El sistema activa la pestaña Salida raster generando una nueva capa raster. Los valores de flujo acumulado vienen expresados en unidades de área. Si se emplea una capa de ponderación, las unidades de la capa resultante son las de dicha capa de ponderación y visualiza a la izquierda un</p>

árbol con las capas que componen el mapa. Ver interfaz 1c.

Prototipo de Interfaz

Interfaz 1a

Realizar análisis hidrológico básico

Obtener direcciones de flujo

Calcular la red de drenaje

Cargar

Interfaz 1b

ACUMULACIÓN DE FLUJO

Parámetros Salida raster

Capas raster

MDE

Ponderación de celdas[opcional]

Opciones

Método

Factor de convergencia para MFD

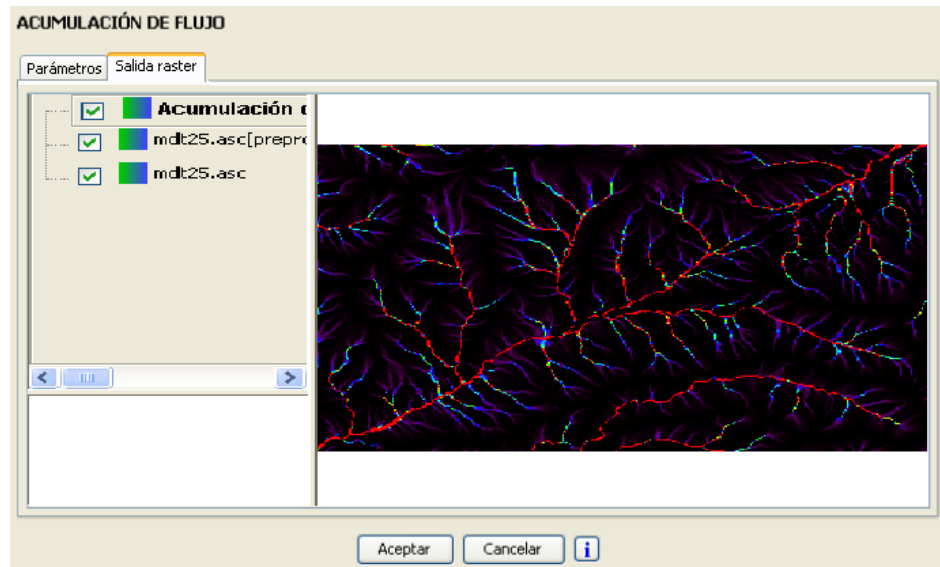
Objetos de salida

Acumulación de flujo[raster]

Vista de salida

Aceptar Cancelar

Interfaz 1c



Flujos Alternos

Acción del actor

5. El usuario da clic en el botón Cancelar.

Respuesta del sistema

6. El sistema cierra la ventana.

Flujos Alternos

Acción del actor

5. El usuario da clic en el botón Información.

Respuesta del sistema

6. El sistema visualiza una ayuda para completar los parámetros de la ventana.

Flujos Alternos

Acción del actor

Respuesta del sistema

6. El sistema lanza un mensaje de error "El factor de convergencia para MFD es incorrecto."

Sección "Calcular la red de drenaje"

Acción del actor

Respuesta del sistema

<p>3. El usuario selecciona la opción calcular la red de drenaje y da clic en el botón cargar.</p>	<p>4. El sistema muestra una ventana Red de drenaje con la siguiente estructura:</p> <p>Pestaña Parámetros</p> <p>Capas raster</p> <p>Con dos combobox uno para seleccionar el modelo MDE, y otro para seleccionar la capa umbral para localizar el inicio de cauces.</p> <p>Opciones</p> <p>Con un combobox Tipo umbral condición que tienen que cumplir las celdas de la capa umbral para el inicio de un cauce (Mayor que, Menor que) y un textbox para definir el valor del valor del umbral (numérico decimal) debe estar en las mismas unidades que la capa umbral (por ejemplo 1000000.0).</p> <p>Objetos de salida</p> <p>Con dos file upload para guardar un archivo temporal raster (los valores de las celdas indican el orden jerárquico del cauce que fluye a través de cada una de ellas. En las celdas por donde no se define un cauce, aparece el valor sin datos) y otro archivo temporal vectorial (la misma estructura que la capa raster, pero en formato vectorial de líneas), y un combobox para seleccionar la vista de salida.</p> <p>Pestaña Salida Raster para visualizar en el mapa la red de drenaje.</p> <p>Tres botones en la parte inferior Aceptar, Cancelar e Información.</p> <p>Ver interfaz 1d.</p>
<p>5. El usuario completa cada una de las opciones que se visualizan en la pestaña Parámetros y da clic en el botón Aceptar.</p>	<p>6. El sistema activa la pestaña Salida raster generando dos nuevas capas (raster y vectorial) con el trazado de los cauces a partir de un MDE y una capa con información adicional y visualiza a la</p>

izquierda un árbol con las capas que componen el mapa. Ver interfaz 1e.

Prototipo de interfaz

Interfaz 1d

RED DE DRENAJE

Parámetros Salida raster

Capas raster

MDE: mdt25.asc[preprocesado][Ejercicio_cuencas]

Capa umbral: Acumulación de flujos[Ejercicio_cuencas]

Opciones

Tipo de umbral: Mayor que

Valor umbral: 1000000.0

Objetos de salida

Red de drenaje[raster]: [Guardar en archivo temporal] ...

Red de drenaje[vectorial]: [Guardar en archivo temporal] ...

Vista de salida: Auto

Aceptar Cancelar i

Interfaz 1e

RED DE DRENAJE

Parámetros Salida raster

Red de drenaje

- Default
- Red de drenaje
- Acumulación de fl
- mdt25.asc[prepro
- mdt25.asc

Aceptar Cancelar i

Flujos Alternos

Acción del actor

Respuesta del sistema

5. El usuario da clic en el botón Cancelar.

6. El sistema cierra la ventana.

Flujos Alternos	
Acción del actor	Respuesta del sistema
5. El usuario da clic en el botón Información.	6. El sistema visualiza una ayuda para completar los parámetros de la ventana.
Flujos Alternos	
Acción del actor	Respuesta del sistema
	6. El sistema lanza un mensaje de error "El Valor umbral es incorrecto."
Poscondiciones	El sistema devuelve los valores correspondientes al análisis hidrológico.

CUS Realizar geomorfometría y análisis del relieve

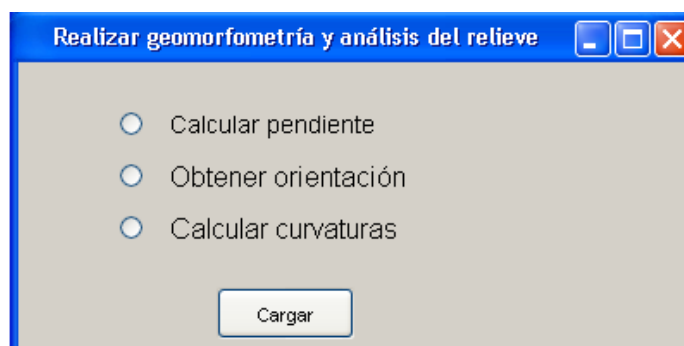
Caso de uso	Realizar geomorfometría y análisis del relieve.
Actores	Usuario
Propósito	Este caso de uso se lleva a cabo con el propósito de calcular el ángulo existente entre el vector normal a la superficie de una pendiente. Calcular el ángulo existente entre el vector que señala el Norte y la proyección sobre el plano horizontal del vector normal a la superficie de ese punto (para el caso de Calcular orientación). Aporta información sobre la concavidad o convexidad de la superficie en un punto dado; la concavidad y la convexidad se asocian a la acumulación de flujo y a la dispersión del mismo respectivamente.
Resumen	El caso de uso se inicia cuando el usuario da clic en la opción realizar el análisis del relieve y geomorfometría a un MDT y termina cuando el sistema valida los datos y visualiza la geomorfometría y análisis del relieve.
Precondiciones	Tener cargada al menos una capa raster.
Referencias	RF 3, RF 4, RF 5
Prioridad	Crítico

Flujo normal de eventos	
Acción del actor	Respuesta del sistema
<p>1. El caso de uso se inicia cuando el usuario selecciona la opción realizar geomorfometría y análisis del relieve.</p>	<p>2. El sistema visualiza una ventana para que el usuario seleccione una de las tres variantes:</p> <ul style="list-style-type: none"> - Calcular pendiente. - Obtener orientación. - Calcular curvaturas. <p>Ver Interfaz 2a.</p>
<p>3. El usuario selecciona una de las tres opciones visualizadas.</p>	<p>4. El sistema realiza la operación según la opción seleccionada por el usuario:</p> <ul style="list-style-type: none"> - Si selecciona la opción Calcular pendiente ir a la sección "Calcular pendiente". - Si selecciona la opción Obtener orientación ir a la sección "Obtener orientación". - Si selecciona la opción Calcular curvaturas ir a la sección "Calcular curvaturas".
Sección "Calcular Pendiente"	
Acción del actor	Respuesta del sistema
<p>3. El usuario selecciona la opción Calcular pendiente y da clic en el botón Cargar.</p>	<p>4. El sistema muestra una ventana Pendiente con la siguiente estructura:</p> <p>Pestaña Parámetros</p> <p>Capas raster</p> <p>Con un combobox para seleccionar el modelo MDE.</p> <p>Opciones</p> <p>Con dos combobox uno para seleccionar el método (que pueden ser Máxima pendiente (Travis et al. 1975), Máxima pendiente por triángulos (Tarboton 1997), Plano de ajuste</p>

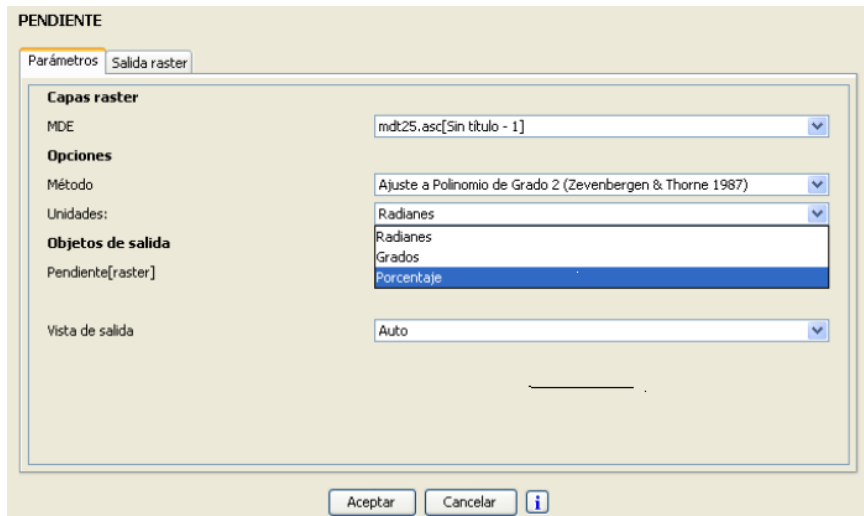
	<p>(Costa-Cabral & Burges 1996), Ajuste a Polinomio de Grado 2 (Bauer, Rohdenburg, Bork 1985), Ajuste a Polinomio de Grado 2 (Heerdegen & Beran 1982), Ajuste a Polinomio de Grado 2 (Zevenbergen & Thorne 1987), Ajuste a Polinomio de Grado 3 (Haralick 1983) y otro para seleccionar las unidades de medida (Radianes, Grados, Porcentaje).</p> <p>Objetos de salida</p> <p>Con un file upload para guardar el archivo y un combobox para seleccionar la vista de salida.</p> <p>Pestaña Salida Raster para visualizar en el mapa la pendiente.</p> <p>Tres botones en la parte inferior Aceptar, Cancelar e Información.</p> <p>Ver interfaz 2b.</p>
<p>5. El usuario completa cada una de las opciones que se visualizan en la pestaña Parámetros y da clic en el botón Aceptar.</p>	<p>6. El sistema activa la pestaña Salida raster generando una nueva capa raster de pendiente con valores expresados en las unidades elegidas y visualiza a la izquierda un árbol con las capas que componen el mapa. Ver interfaz 2c.</p>

Prototipo de interfaz

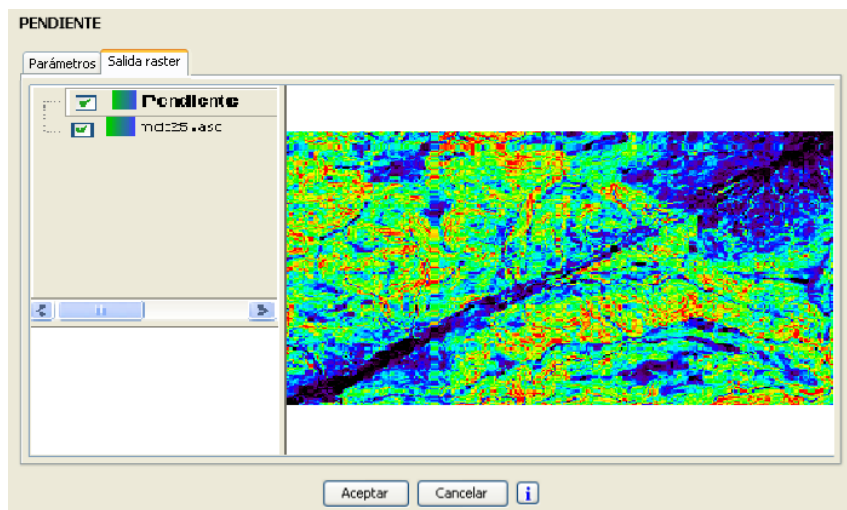
Interfaz 2a



Interfaz 2b



Interfaz 2c



Flujo Alternos

Acción del actor	Respuesta del sistema
5. El usuario da clic en el botón Cancelar.	6. El sistema cierra la ventana.

Flujo Alternos

Acción del actor	Respuesta del sistema
------------------	-----------------------

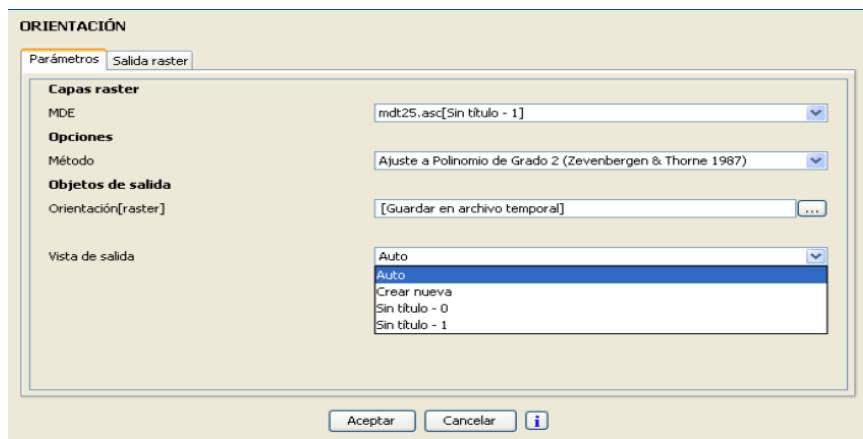
5. El usuario da clic en el botón Información.	6. El sistema visualiza una ayuda para completar los parámetros de la ventana.
Sección “Obtener Orientación”	
Acción del actor	Respuesta del sistema
3. El usuario selecciona la opción Obtener orientación y da clic en el botón Cargar.	<p>4. El sistema muestra una ventana Pendiente con la siguiente estructura:</p> <p>Pestaña Parámetros</p> <p>Capas raster</p> <p>Con un combobox para seleccionar el modelo MDE.</p> <p>Opciones</p> <p>Con un combobox para seleccionar el método (que pueden ser Máxima pendiente (Travis et al. 1975), Máxima pendiente por triángulos (Tarboton 1997), Plano de ajuste (Costa-Cabral & Burges 1996), Ajuste a Polinomio de Grado 2 (Bauer, Rohdenburg, Bork 1985), Ajuste a Polinomio de Grado 2 (Heerdegen & Beran 1982), Ajuste a Polinomio de Grado 2 (Zevenbergen & Thorne 1987), Ajuste a Polinomio de Grado 3 (Haralick 1983).</p> <p>Objetos de salida</p> <p>Con un file upload para guardar el archivo y un combobox para seleccionar la vista de salida.</p> <p>Pestaña Salida Raster para visualizar en el mapa la orientación.</p> <p>Tres botones en la parte inferior Aceptar, Cancelar e Información.</p> <p>Ver interfaz 2d.</p>
5. El usuario completa cada una de las opciones que se visualizan en la pestaña Parámetros y da	6. El sistema activa la pestaña Salida raster generando una nueva capa raster de orientación

clic en el botón Aceptar.

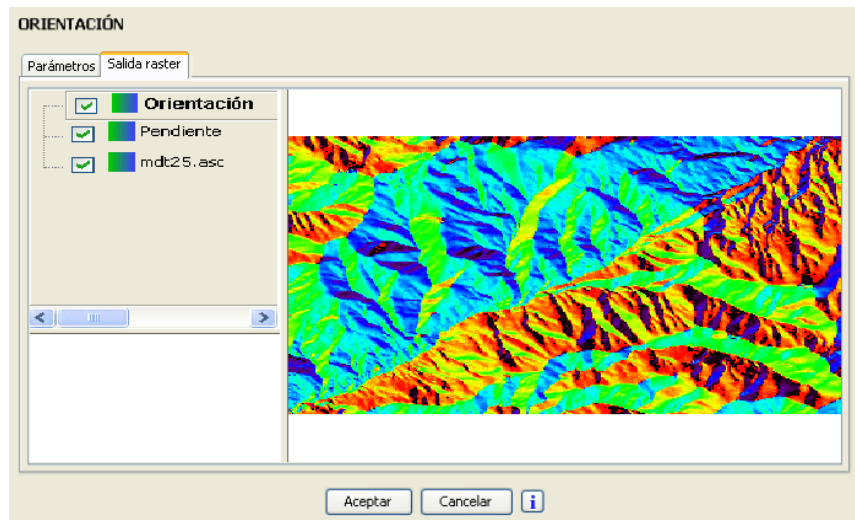
con valores expresados en las unidades elegidas. Las representaciones en relieve sombreado y este mapa de orientaciones guardan algunos puntos en común, pero no son lo mismo y visualiza a la izquierda un árbol con las capas que componen el mapa. Ver interfaz 2e.

Prototipo de interfaz

Interfaz 2d



Interfaz 2e



Flujo Alternos	
Acción del actor	Respuesta del sistema
5. El usuario da clic en el botón Cancelar.	6. El sistema cierra la ventana.
Flujo Alternos	
Acción del actor	Respuesta del sistema
5. El usuario da clic en el botón Información.	6. El sistema visualiza una ayuda para completar los parámetros de la ventana.
Sección “Calcular Curvaturas”	
Acción del actor	Respuesta del sistema
3. El usuario selecciona la opción Calcular curvaturas y da clic en el botón Cargar.	<p>4. El sistema muestra una ventana Pendiente con la siguiente estructura:</p> <p>Pestaña Parámetros</p> <p>Capas raster</p> <p>Con un combobox para seleccionar el modelo MDE.</p> <p>Opciones</p> <p>Con un combobox para seleccionar el método (que pueden ser Ajuste a Polinomio de Grado 2 (Bauer, Rohdenburg, Bork 1985), Ajuste a Polinomio de Grado 2 (Heerdegen & Beran 1982), Ajuste a Polinomio de Grado 2 (Zevenbergen & Thorne 1987), Ajuste a Polinomio de Grado 3 (Haralick 1983).</p> <p>Objetos de salida</p> <p>Con cuatro file upload para guardar los archivos raster (Clasificación, Curvatura Horizontal, Curvatura, Curvatura vertical) y un combobox para seleccionar la vista de salida.</p>

	<p>Pestaña Salida Raster para visualizar en el mapa las curvaturas.</p> <p>Tres botones en la parte inferior Aceptar, Cancelar e Información.</p> <p>Ver interfaz 2f.</p>
<p>5. El usuario completa cada una de las opciones que se visualizan en la pestaña Parámetros y da clic en el botón Aceptar.</p>	<p>6. El sistema activa la pestaña Salida raster visualizando el valor correspondiente a la curvatura calculada y visualiza a la izquierda un árbol con las capas que componen el mapa. Ver interfaz 2g.</p>

Prototipo de interfaz

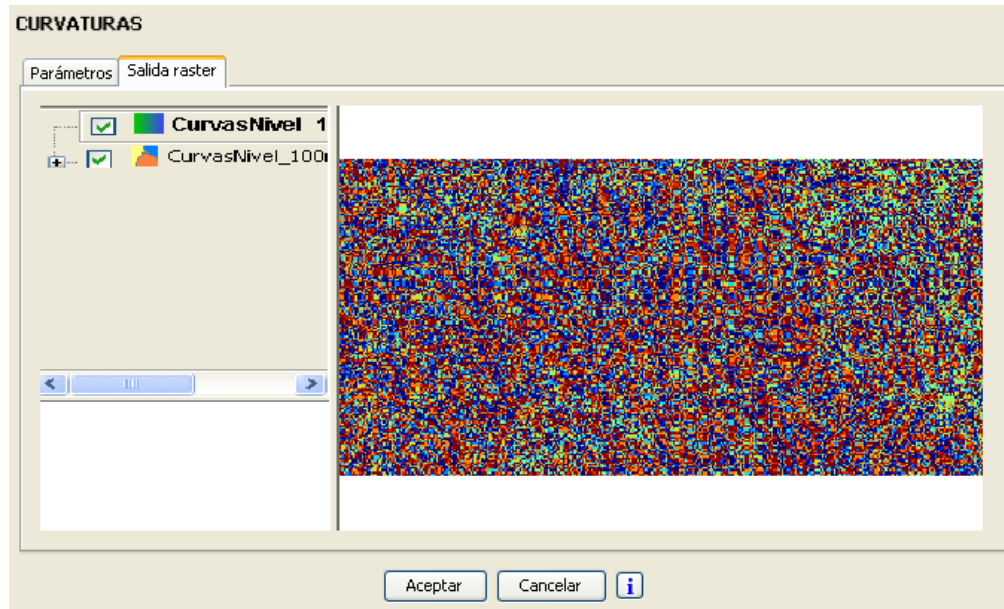
Interfaz 2f

The screenshot shows a software window titled "CURVATURAS" with two tabs: "Parámetros" and "Salida raster". The "Salida raster" tab is active. The interface is organized into several sections:

- Capas raster:** A dropdown menu showing "mdt25.asc[Sin título - 1]".
- Opciones:** A dropdown menu showing "Ajuste a Polinomio de Grado 2 (Zevenbergen & Thorne 1987)".
- Objetos de salida:** Three text input fields, each with a browse button (...):
 - Clasificación[raster]: "[Guardar en archivo temporal]"
 - Curvatura Horizontal[raster]: "C:\Curvatura Horizontal"
 - Curvatura[raster]: "[Guardar en archivo temporal]"
 - Curvatura Vertical[raster]: "[Guardar en archivo temporal]"
- Vista de salida:** A dropdown menu showing "Auto".

At the bottom of the window, there are three buttons: "Aceptar", "Cancelar", and an information icon (i).

Interfaz 2g



Flujo Alternos

Acción del actor

5. El usuario da clic en el botón Cancelar.

Respuesta del sistema

6. El sistema cierra la ventana.

Flujo Alternos

Acción del actor

5. El usuario da clic en el botón Información.

Respuesta del sistema

6. El sistema visualiza una ayuda para completar los parámetros de la ventana.

Poscondiciones

Muestra los resultados correspondientes a la opción seleccionada por el usuario.

¿Qué son los patrones de Casos de Uso?

Los patrones de Casos de Uso facilitan, a partir de su correcta aplicación, la identificación de los casos de uso que componen el sistema, así como las relaciones entre casos de uso. De este modo se consigue: (1)

facilitar la detección de requisitos, (2) mejorar la alineación de procesos de negocio, (3) mejorar la trazabilidad entre procesos de negocio y sistemas (minimizando el riesgo de las evoluciones). Algunos de estos patrones son:

- CRUD.
- Reglas de negocio.
- Sistemas en capas.
- Múltiples actores.
- Casos de Uso en secuencia.
- Servicios opcionales.
- Vistas ortogonales.
- Jerarquía de componentes.
- Casos de Uso largos.

Luego de analizar la complejidad del Diagrama de Casos de Uso del Sistema y analizar las características de los patrones antes mencionados se concluye que por su simplicidad no es necesario aplicar ninguno de estos patrones.

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. (24)

Entre Requisitos Funcionales y Casos de Usos del Sistema

Tabla 4 Requisitos Funcionales y Casos de Usos del Sistema

Requisitos	CU 1	CU 2	CU 3	CU 4	CU 5	CU 6
RF 1	X					
RF 2	X					
RF 3		X				
RF 4		X				
RF 5		X				
RF 6			X			
RF 7			X			
RF 8				X		
RF 9				X		
RF 10				X		
RF 11					X	
RF 12					X	
RF 13						X

Entre Actores y Casos de Usos del Sistemas

Tabla 5 Actores y Casos de Usos del Sistema

Actor	CU 1	CU 2	CU 3	CU 4	CU 5	CU 6
Usuario	X	X	X	X	X	X

2.5 Conclusiones parciales

Como resultado de la realización de este capítulo se obtuvieron los artefactos correspondientes al modelado del dominio y del sistema, esta documentación posibilitará un mejor entendimiento entre analistas y desarrolladores, se describió el dominio del problema a través de su diagrama de clases conceptual, permitiendo un mayor entendimiento de los elementos que giran alrededor de la situación problemática, se realizó el diagrama de Casos de Usos del Sistema correspondiente, determinándose el actor que interviene en el sistema, así como los casos de uso que son iniciados por el mismo, además de las descripciones textuales de estos con el objetivo de lograr un mejor entendimiento de lo que se quiere realizar.

CAPÍTULO 3: ESTUDIO DE FACTIBILIDAD Y VALIDACIÓN DEL SISTEMA

Introducción

Para el desarrollo de un proyecto de software y lograr el éxito del mismo, es crucial cumplir con el paso de definición de requisitos. Para ello es necesaria la asesoría de ingenieros de gran experiencia en el análisis, planeación, diseño y desarrollo. Todo esto hace que se hayan creado una serie de pautas y lineamientos básicos. Los análisis de factibilidad son de suma importancia ya que se realizan con el objetivo de auxiliar a una organización a lograr sus metas y a su vez de cubrir sus fines con los recursos actuales. El objetivo fundamental de la planificación y el análisis de la factibilidad es establecer planes razonables para desarrollar la Ingeniería de Software y manejar los cambios de los proyectos de software.

3.1 Técnicas de estimación

Realizar la estimación de un proyecto de software no es tarea fácil, especialmente cuando no existe un método único viable para cada tipo de proyecto. Actualmente la planificación se ha convertido en uno de los principales retos para la gestión de proyectos. Se considera imprescindible para planificar, la aplicación de buenos métodos de estimación, por la creciente influencia que ejercen en el control preciso, predecible y repetido sobre los procesos de producción y los productos de software. Para la elección de la técnica de estimación se investigaron dos métodos. A continuación se realiza un resumen de las características de cada uno de ellos.

COCOMO: fue propuesto y desarrollado por Barry Boehm en 1981, es uno de los modelos de estimación de costo mejor documentado, estudiado y utilizado en la industria de software. El modelo permite, basándose en un grupo de ecuaciones no lineales obtenidas mediante técnicas de regresión a través de un histórico de proyectos ya realizados; estimar el esfuerzo, costo y tiempo que se requiere en un proyecto de software a partir de una medida del tamaño del mismo, expresada en el número de líneas de código que se estimen generar para la creación del producto software. El modelo original ha evolucionado a un modelo más completo llamado COCOMO III. **(19)**

La **planificación basada en Casos de Uso** es una técnica de estimación mediante el análisis de Puntos de Casos de Uso sin ajustar, que es un método propuesto originalmente por Gustav Karner de Objectory AB, y posteriormente refinado por muchos otros autores. Se trata de 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 finalmente, contabilizar el tiempo total estimado para el proyecto a partir de esos factores. A continuación, se detallan los pasos a seguir para la aplicación de éste método. **(20)**

3.1.1 Selección de la técnica de estimación

El método de estimación seleccionado para la siguiente investigación es Análisis de Punto de Caso de Uso por ser un método de estimación y cálculo de tamaño del software basado en cuentas hechas sobre los casos de uso para un sistema de software, es uno de los métodos más efectivos para capturar la funcionalidad de un sistema. Este método posee ventajas con respecto a los demás, como por ejemplo, el método COCOMO que no fue seleccionado por tener el inconveniente de que hay que contar con una cantidad de líneas de código fuente a implementar por casos de uso y para lograr este indicador se debe tener experiencia en el desarrollo de aplicaciones.

Puntos de caso de uso sin ajustar (UUCP)

Este cálculo se lleva a cabo a través de la ecuación:

$$UUCP = UAW + UUCW$$

Donde:

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.

Factor de peso de los actores sin ajustar (UAW)

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:

Tabla 6: Factor de Peso de los Actores sin Ajustar.

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

Por tanto:

$$UAW = \sum \text{cant actores} * \text{peso}$$

$$UAW = 1*3$$

$$UAW = 3$$

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 atómica, es decir, se efectúa la secuencia de actividades completa, o no se efectúa ninguna de las actividades de la secuencia. Otro de los métodos a utilizar para el cálculo del valor **Factor de Peso de los Casos de Uso sin ajustar (UUCW)** es el basado en clases de análisis.

El método basado en clases de análisis donde se toma en cuenta el número de clases de análisis que tiene un caso de uso y se evalúa según la siguiente tabla:

Tabla 7: Factor de peso de los Casos de Uso sin Ajustar.

Tipo de CU	Descripción	Peso	Cantidad de CU	Total
Simple	El Caso de Uso contiene de 1 a 3 transacciones	5	0	0
Medio	El Caso de Uso contiene de 4 a 7 transacciones	10	6	60
Complejo	El Caso de Uso contiene más de 8 transacciones	15	0	0
Total			6	60

Por tanto:

$$UUCW = 60$$

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**:

$$UUCP = UAW + UUCW$$

$$UUCP = 3 + 60$$

$$UUCP = 63$$

Puntos de caso de uso ajustados (UCP)

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

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

Donde:

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).

Tabla 8: Factor de Complejidad Técnica.

Factor	Descripción	Peso	Valor asignado	Total
T1	Sistema distribuido.	2	1	2
T2	Objetivo de performance o tiempo de respuesta.	1	4	4
T3	Eficiencia del usuario final.	1	1	1
T4	Procesamiento interno complejo.	1	1	1
T5	El código debe ser reutilizable.	1	5	5
T6	Facilidad de instalación.	0.5	3	1.5
T7	Facilidad de uso.	0.5	4	2
T8	Portabilidad.	2	5	10
T9	Facilidad de cambio.	1	5	5
T10	Concurrencia.	1	3	3
T11	Incluye objetivos especiales de seguridad.	1	3	3
T12	Provee acceso directo a terceras partes.	1	0	0
T13	Se requieren facilidades especiales de entrenamiento a usuarios.	1	1	1
Total			0	40.5

Significado de los valores:

0: No presente o sin influencia.

1: Influencia o presencia incidental.

2: Influencia o presencia moderada.

3: Influencia o presencia media.

4: Influencia o presencia significativa.

5: Influencia o presencia fuerte.

Comentarios:

T1: El sistema no es distribuido.

T2: Los tiempos de respuesta deben ser muy altos, debe ofrecer un buen rendimiento.

T3: El usuario final requiere la adecuada preparación para usar el sistema.

T4: El procesamiento interno es medio, puesto que para la gestión de documentos se utiliza un framework como subsistema base.

T5: El código es reutilizable, ya que puede ser utilizado en desarrollos similares.

T6: Aunque son varios los módulos a instalar, el trabajo de instalación es muy sencillo.

T7: El sistema cuenta con una interfaz gráfica y amigable, muy sencilla de utilizar.

T8: El sistema debe ser multiplataforma, siendo usado bajo los Sistemas Operativos Windows y Linux.

T9: Se le podrá realizar algún cambio específico, no todos los cambios de forma general.

T10: Algunos procesos a los que accede el usuario dependen de que esté funcionando otro.

T11: Se crean diferentes cuentas de usuario y se asignan a cada uno los permisos pertinentes, mostrándose a cada usuario sólo las funcionalidades del sistema sobre las cuales tiene permiso de acceso.

T12: No provee acceso a terceras partes.

T13: No es tan necesario darles preparación a los usuarios para poder usar el sistema.

Por lo que:

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

$$TCF = 0.6 + 0.01 * 40.5$$

$$TCF = 1.04$$

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 requisitos extremadamente inestables, 3 estabilidad media y 5 requisitos 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.

Tabla 9: Factor de Ambiente.

Factor	Descripción	Peso	Valor asignado	Total
E1	Familiaridad con el modelo de proyecto utilizado.	1.5	1	1.5
E2	Experiencia en la aplicación.	0.5	0	0
E3	Experiencia en orientación a objetos.	1	3	3
E4	Capacidad del analista líder.	0.5	5	2.5
E5	Motivación.	1	4	4

E6	Estabilidad de los requisitos.	2	2	4
E7	Personal part-time.	-1	4	-4
E8	Dificultad del lenguaje de programación.	-1	4	-4
Total				14

Comentarios:

E1: El equipo se encuentra familiarizado con el modelo utilizado.

E2: No se tiene experiencia en el trabajo con aplicaciones similares.

E3: Hay una experiencia moderada con la programación orientada a objetos.

E4: El analista líder es una persona capacitada.

E5: Hay una alta motivación en el equipo de trabajo para la construcción del producto final.

E6: Los requisitos no cambian, aún cuando se implementa el sistema.

E7: Los miembros no trabajan a tiempo completo debido a que parte del grupo de trabajo son estudiantes.

E8: Se programa con PHP, un lenguaje de poca complejidad.

Por lo que:

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

$$EF = 1.4 - 0.03 * 14$$

$$EF = 0.8$$

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**:

$$UCP = UUCP * TCF * EF$$

$$UCP = 63 * 1.04 * 0.8$$

$$UCP = 54.42$$

Esfuerzo horas-hombre (E)

El esfuerzo en horas-hombre viene dado por:

$$E = UCP * CF$$

Para obtener el **factor de conversión (CF)** se cuentan 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 o 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.

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

$$\text{Total}_{EF} = 0+2$$

$$\text{Total}_{EF} = 2$$

En este caso se puede decir que:

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

Por lo que:

$$E = UCP * CF$$

$$E = 54.42 * 20$$

$$E = 1048.3 \text{ Horas-Hombre}$$

Calcular el esfuerzo de todo el proyecto

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.

Para ello 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:

Tabla 10: Esfuerzo del Proyecto.

Actividad	Porcentaje	Horas-Hombre
Análisis	10.00 %	262.1 horas -hombre
Diseño	20.00 %	524.2 horas -hombre
Programación	40.00 %	1048.3 horas-hombre
Pruebas	15.00 %	393.1 horas-hombre
Sobrecarga(otras actividades)	15.00 %	393.1 horas-hombre
Total	100 %	2620.8 horas-hombre

Suponiendo que una persona trabaje 8 horas por día, y un mes tiene como promedio 20 días de trabajo; la cantidad de horas que puede trabajar una persona en 1 mes es 160 horas. Esto quiere decir, que una persona puede realizar el problema analizado en más o menos 1 año y 4 meses.

Si en el equipo hay 5 personas y todas realizan el mismo esfuerzo entonces el problema analizado puede terminarse en aproximadamente 3 meses y medio (3.2 meses).

Costo del proyecto

Una vez desarrollada la técnica Planificación de Casos de Uso se procede a llevar a cabo el cálculo de los costos.

CH: Cantidad de hombres.

TTP: Tiempo total del proyecto.

SM: Salario mensual.

TEF: Total de esfuerzo.

CP: Costo del proyecto.

En caso de ser estudiante:

Asumiendo que el salario básico de un estudiante es de \$ 100.00 mensual (estipendio).

$$CH = 1$$

$$SM = \$100$$

$$TEF = 4$$

$$CP = SM * CH * TEF$$

$$CP = \$400$$

$$TTP = TEF/CH$$

$$TTP = 4$$

Se estima que el proyecto tenga un costo total de \$400, trabajando 1 estudiante, durante 4 meses.

En caso de ser profesor:

Asumiendo que el salario básico de un profesor de la UCI es de \$ 495.00 mensual.

$$CH = 4$$

$$SM = \$495$$

$$TEF = 4$$

$$CP = SM * CH * TEF$$

$$CP = \$ 7920$$

$$TTP = TEF/CH$$

$$TTP = 4$$

Se estima que el proyecto tenga un costo total de \$ 7920, trabajando 4 profesores de la UCI, durante 4 meses.

Por lo que:

Al utilizar herramientas de software libre, no se hace necesario los gastos referentes a las licencias para los permisos de implementación e implantación. Para el desarrollo del sistema la universidad cuenta con los recursos técnicos necesarios de tipo tecnológico y con el personal, sustentado lo anterior no se incurrirán en gastos de hardware o software. Por lo expuesto hasta el momento, desarrollado por pasos todos los cálculos y análisis pertinentes de los costos del proyecto, se llegó a la conclusión de que el diseño del componente para el análisis hidrológico, álgebra de mapas y análisis geoestadístico en sistemas de información geográfica es factible.

3.2 Técnicas de validación

La validación de requisitos es un proceso que tiene como misión demostrar que la definición de los requisitos precisa realmente lo que el cliente necesita, hoy en día existen diferentes técnicas y métodos de validación entre los que se encuentran: **(22)**

- Método Delphi.
- Test de Turing.
- Comparación de los resultados de salida del modelo con los del sistema real.
- Listas de chequeo.
- Matrices de trazabilidad.

A continuación se realiza una breve descripción de cada una de las técnicas de estimación.

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 se consideren más importantes sobre el mismo. 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.

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 resultante 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 entre las respuestas sobre lo que encuentran inconsistente, se puede utilizar para realizar mejoras en el modelo.

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.

Listas de chequeo.

Son frecuentemente usadas en inspecciones o revisiones de artefactos generados en el proceso de producción de software; son listas de aspectos que deben ser completados o verificados. Las Listas de Chequeo están basadas en la identificación de las técnicas de Prueba para evaluar cada Subcaracterística de las Características de Calidad. Una lista de chequeo es un formulario de preguntas, las cuales dependen del objetivo para el cual son usadas. Estas listas están clasificadas según las etapas del Proceso de Prueba. Para dar respuesta a cada pregunta se considera una escala del 1 al 5, en donde el uno (1) siempre es la respuesta menos significativa y cinco (5) la más significativa.

Aplicación y resultado de la validación de la propuesta

Después de haber realizado un estudio de las diferentes técnicas se llegó a la conclusión que la que aplicará son las listas de chequeo.

Tabla 11: Listas de Chequeo de Especificación de Requisitos. (22)

Nivel	Criterio de evaluación	Evaluación	N.P.	Observaciones
!!!	¿Están todos los requisitos redactados de forma simple y clara para aquellos que vayan a consultarlo en un futuro?	4		
	¿Debería especificarse algún requisito con más detalle?	4		
	¿Debería especificarse algún requisito con menos detalles?		NP	
	¿Todos los requisitos identificados se centran en lo que el sistema debe hacer y no cómo el sistema debe hacerlo?	5		Se centran en lo que el sistema debe hacer.
	¿Se han identificado los requerimientos de software y de hardware?	5		
	¿Los requerimientos de soporte y usabilidad se han identificados?	5		
	¿Se han identificado los requerimientos de seguridad (confidencialidad, integridad, disponibilidad)?	5		
	¿Se puede verificar cada requisito?	5		
	¿Se han enumerado los requisitos incluso los que se derivan de otros requisitos?	5		
	¿No aparece un mismo requisito en más de un lugar del documento de	5		No aparecen repetidos los requisitos.

	especificación?			
!!!	¿No existe contradicción entre lo especificado por un requisito y lo especificado por otro?	5		No
!!!	¿Ha identificado errores ortográficos?	5		Está correctamente redactado.

Tabla 12: Listas de Chequeo para Casos de Usos del Sistema. (22)

Nivel	Criterio de evaluación	Evaluación	N.P.	Observaciones
	¿Cada caso de uso registra claramente lo que el sistema debe hacer?	4		
!!!	¿Se ha descrito con precisión todas las alternativas o excepciones?	0		
!!!	¿Está en infinitivo y refleja de manera clara el objetivo del usuario sobre el sistema?	5		
!!!	¿El nombre del caso de uso es único?	5		
	¿El nombre del caso de uso es intuitivo?	5		
	¿El resumen dice como se inicia, como termina y las operaciones principales que realiza el caso de uso?	4		
	¿Se escribe una precondición si y solo si a partir de la ocurrencia de un suceso determinado comienza el	5		

	caso de uso?			
	¿La precondición es válida tanto para flujos básicos como flujos alternativos?	5		
	¿La pos condición plasma cambios que suceden en el sistema al terminarse de ejecutar el caso de uso?	5		
	¿Se especifica la complejidad del caso de uso?	5		
!!!	¿Está descrito el caso de uso en presente?	5		
!!!	¿Se describe de manera comprensible y detallada las acciones del actor frente al sistema? ¿Está lo más parecido a un manual de ayuda?	5		
!!!	¿El Caso de Uso está relacionado con al menos un actor?	5		
!!!	¿Si hay dos actores interactuando con el caso de uso está generalizado en uno solo?	5		
	¿Comienza diciendo “El caso de uso se inicia cuando el usuario...”?	4		
	¿Termina diciendo en un evento independiente “El caso de uso termina”?	4		
	¿No existen abreviaturas?	3		MDT, MDE.
!!!	¿Las alternativas o excepciones se	5		

	reflejan como flujos alternos?			
	¿En todos los CU que se introducen datos tienen un flujo alternativo donde el sistema valida la integridad de los datos que se introducen y muestra un mensaje en caso de que los datos estén incompletos?	5		
	¿Los flujos alternativos se nombran con el número del paso que lo generó en el flujo básico, una letra, ordenados alfabéticamente que lo produjo?	5		
!!!	¿En la sección flujos alternativos se describen todas las excepciones que existan por muy evidentes que parezcan?	3		No se describen las excepciones si son muy evidentes.
!!!	¿El diagrama de casos de uso expresa en detalles y claramente lo que debe hacer el sistema?	4		
	Si la modelación de las interacciones con el sistema es muy extensa ¿ha empleado los paquetes de caso de uso?		NP	
!!!	¿Ha identificado errores ortográficos?	5		Todo el documento está bien redactado.

Con la utilización de las listas de chequeo no solo se valida la integridad de las funcionalidades que presenta la aplicación, sino que se garantiza la calidad del producto, el cual similarmente está a la altura de cualquier otro a nivel internacional.

3.3 Conclusiones parciales

En este capítulo se realiza un análisis de la factibilidad del sistema donde se estima el esfuerzo que requiere el mismo. Teniendo en cuenta que en el desarrollo del producto laboran 5 personas, se debe desarrollar en aproximadamente 4 meses, luego del análisis del costo total se concluye que es factible la puesta en marcha de la aplicación. Además se lleva a cabo la validación de la solución propuesta la cual se realiza a través de listas de chequeo, verificándose que no existan inconsistencias en los requisitos y casos de uso del sistema. Es por ello, que se concluye que los requisitos definidos están descritos con claridad, son verificables, consistentes, completos, alcanzables y cumplen con los estándares de calidad requeridos además de ser los deseados por el cliente.

CONCLUSIONES GENERALES

Una vez concluida la investigación correspondiente al análisis de un componente para el álgebra de mapas, análisis hidrológico y geoestadístico, se llega a la conclusión que los objetivos propuestos para el presente trabajo han sido cumplidos satisfactoriamente

- Se realizó un estudio acerca de la Metodología de Desarrollo quedando seleccionada la metodología RUP, permitiendo guiar todo el proceso del análisis del componente, así como utilizar las herramientas más adecuadas para el trabajo, que a menudo son herramientas simples o incluso herramientas de código abierto. El Lenguaje de Modelado UML y la Herramienta CASE Visual Paradigm que es una herramienta multiplataforma permitió generar todos los artefactos correspondientes a las diferentes etapas de desarrollo de RUP.
- Luego de reunirse con los analistas y obtener las especificaciones brindadas por ellos, se obtuvo el diagrama de clases del dominio con el fin de simplificar de manera notable algunos aspectos de las fases de modelación iniciales.
- Una vez realizado el diagrama de clases del dominio, se procedió al levantamiento de requisitos, permitiendo obtener las funcionalidades requeridas para el componente, se realizó la documentación referente al análisis de sistema, logrando la descripción textual de todos los casos de usos que permitirá mejor entendimiento de lo que se quiere realizar permitiendo construir el componente con todas las funcionalidades que se describieron.
- Por último se realizó una evaluación de la calidad y factibilidad del componente mediante técnicas de validación y de factibilidad, una vez aplicada la técnica de validación se puede decir que los requisitos definidos están descritos con claridad, son verificables, consistentes, completos, alcanzables y cumplen con los estándares de calidad requeridos, al realizar los cálculos correspondientes a la técnica de factibilidad se concluye que la puesta en marcha de la aplicación es factible.

De modo general, permitió la elaboración de todos los artefactos correspondientes al rol de analista de sistemas; obteniendo un idioma común entre el cliente y el grupo de desarrollo. Obteniendo como

resultado una amplia y organizada documentación sobre los procesos relacionados con el álgebra de mapas, análisis hidrológico y geoestadístico.

RECOMENDACIONES

A pesar que se le dio cumplimiento a todos los objetivos planteados para el desarrollo de este trabajo, en el transcurso del mismo han surgido nuevas ideas que podría implicar mejoras en las funcionalidades del sistema propuesto, para ello se recomienda:

- El estudio de esta tesis para poner en práctica la solución propuesta, para la implementación de estas herramientas de análisis hidrológico, geoestadístico y álgebra de mapas.
- Se propone darle seguimiento para lograr un producto de mayor calidad.
- Se exhorta continuar el estudio de otros algoritmos para el análisis hidrológico o modificar los propuestos según las necesidades del cliente.

BIBLIOGRAFÍA REFERENCIADA

1. Sarría., Francisco Alonso. Introducción a los Sistemas de Información Geográfica.
2. ORDENAMIENTO TERRITORIAL RURAL. MANUAL CURSO. Santiago, Chile : s.n., Mayo 2003.
3. Calzadilla., Mileidis Nieves. Algoritmos para el módulo de análisis del terreno de GeneSIG: Análisis hidrológico y álgebra de mapa. La Habana,Cuba : s.n., 2009.
4. Urciuolo, Adriana. Modelado de simulación hidrológica utilizando un enfoque de desarrollo basado en componentes.
5. Felicísimo, Ángel Manuel. Modelos Digitales del terreno: Introducción y aplicaciones en las ciencias ambientales.
6. Peralta, F. J. Análisis Espacial con Datos Raster en ArcGIS Desktop 9.2.
7. Adriana, I. R. Urcolo. Modelado de simulación hidrológica utilizando un enfoque de desarrollo basado en componentes. Universidad Nacional de la Patagonia San Juan Bosco – Sede Ushuaia : s.n.
8. Sánchez, Henry Garzón. Evaluación de la erosión hídrica y la escorrentía superficial bajo sistemas agroforestales, en tierras de ladera Centro Agrónomo tropical de investigación y enseñanza. Turrialba, Costa Rica. : s.n., 1991.
9. <http://www.programarvba.com/componentes-sistema-gis.htm>
10. Pesce., Aportes de la Hidrología Morfométrica a la Ecología Fluvial. Chile. : s.n., 2006.
11. Belmonte., Silvina. Desarrollo de modelos hidrológicos con herramientas SIG. Instituto de Recursos Naturales y Ecodesarrollo. Universidad Nacional de Salta
12. Craig Larman., UML y Patrones.
13. Canós, José H., Letelier, Patricio y Penadés, María del Carmen. Metodología Ágiles en el desarrollo de Software. : s.n., 2004.

BIBLIOGRAFÍA CONSULTADA

1. I.Jacobson, G. Booch, J. Rumbaugh. *L PROCESO UNIFICADO DE DESARROLLO DE SOFTWARE*.
2. *Herramientas Case, COLECCION CULTURA INFORMATICA*. Noviembre de 1999.
3. Calzadilla Nieves, Mileidis. Algoritmos para el módulo de análisis del terreno de GeneSIG: Análisis hidrológico y álgebra de mapa. La Habana,Cuba : s.n., 2009.
4. Pesce., Aportes de la Hidrología Morfométrica a la Ecología Fluvial. Chile. : s.n., 2006.
5. Belmonte., Silvina. Desarrollo de modelos hidrológicos con herramientas SIG. Instituto de Recursos Naturales y Ecodesarrollo. Universidad Nacional de Salta
6. Galván, L. Olías, M.; Fernández de Villarán, R.; Domingo, J.M; Nieto, J.M. (Huelva, España) Modelización de la cuenca del río Meca.
7. Hidro-SIG: Una herramienta para la estimación de balances hidrológicos de Colombia. Mesa, Oscar. Proveda, Germán. Universidad Nacional de Colombia, Sede Medellín, Pregrado de Ingeniería Civil.
8. Urcolo Adriana, I. R. Modelado de simulación hidrológica utilizando un enfoque de desarrollo basado en componentes. Universidad Nacional de la Patagonia San Juan Bosco Sede Ushuaia, Darwin y Canga.WordReference.com. [En línea] 2010, diccionario de la lengua española.
9. Pérez., Munir Morad y Alejandro Triviño. Modelo digitales de elevación para el continuo de elevaciones mexicanas.
10. Mapping Interactive. http://www.mappinginteractivo.com/plantilla-ante.asp?id_articulo=1318.
11. Llorente Isidro, Miguel, Andrés Díez-Herrero y Luis Laín Huerta: Aplicaciones de los SIG al análisis y gestión del riesgo de inundaciones: Avances recientes. , Instituto Geológico y Minero de España. Madrid.
12. Pressman, R. S. (2005). Ingeniería Del Software. Mcgraw-hill (2005, 6ª edición).
13. Alan, M. D. (1993). Requisitos de software: objetos, funciones y los Estados. Prentice Hall.

14. Jacobson, I. B. (2004). El Proceso Unificado de desarrollo de Software. La Habana: Félix Varela.
15. Bosque, Sendra. 1992. Sistemas de Información Geográfica. Madrid: Rialp, 1992.
16. Isaías Carrillo Pérez, Rodrigo Pérez González, Aureliano David Rodríguez Martín. Metodología de desarrollo del software. [documento] 15 de enero de 2008.