



UNIVERSIDAD DE CHILE
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS
DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN

VISUALIZACIÓN DE MALLAS DE TERRENO E IDENTIFICACIÓN DE PATRONES DE DRENAJE EN CUENCAS

MEMORIA PARA OPTAR AL TÍTULO DE INGENIERO CIVIL EN COMPUTACIÓN

PILLIPPA IGNACIA IRIS PÉREZ PONS

PROFESOR GUÍA:
NANCY HITSCHFELD KAHLER

MIEMBRO DE LA COMISIÓN:
LUISA PINTO LINCOÑIR
PATRICIO INOSTROZA FAJARDIN

Este trabajo ha sido parcialmente financiado por el Proyecto Fondecyt N° 1120495

SANTIAGO DE CHILE

2015

Resumen

El departamento de geología, continuamente está realizando investigaciones sobre los procesos de sedimentación y formación de los ríos. Con la información recolectada, la cual es extraída desde los ríos a lo largo de su recorrido, es posible tener una visión general de la evolución geomorfológica y de los procesos que han ocurrido anteriormente en aquellos lugares, y a su vez, es posible la detección de la presencia de volcanes, fracturas en las rocas, materiales presentes en la corteza del lugar, entre otros.

Por medio del estudio de las formas de los ríos, es posible clasificarlos según el tipo de red de drenaje, conjunto de ríos, vaguadas, barrancos y las huellas impresas en el terreno que deja la circulación constante e/o intermitente del flujo del agua. Con esta información, es posible la identificación de fallas en el terreno, estado de las rocas del subsuelo e identificación del tipo de sedimento que se produjo en los caudales y ahora es parte del terreno. Es por lo anterior, la necesidad de proveer a la comunidad un software gratuito que permita análisis, investigación e identificación de lugares que presenten patrones de gran interés

El objetivo de la presente memoria es desarrollar una herramienta que permita la visualización de terrenos y análisis científico por parte de los geólogos. En particular, la implementación de algoritmos para la detección de cauces e identificación de patrones de drenaje, y su posterior visualización.

Para la realización de la memoria, se procedió a la investigación e inspección en terreno de las necesidades que conlleva la investigación geomorfológica. Luego, se comenzó con la búsqueda de las tecnologías y artículos científicos relacionados al tema. Finalmente, la implementación de la solución a través de diferentes algoritmos para la detección de cauces e identificación de patrones, al mismo tiempo se enfocó en la arquitectura del software, privilegiando que fuera extensible en la extracción de datos de diferentes fuentes y algoritmos utilizados para la detección de cuencas.

Posteriormente, se procede a realizar la validación del software implementado con los resultados entregados por RiverTools y la comparación de los propios resultados entre los distintos algoritmos. Donde para este proceso de validación, se utilizan distintas imágenes satelitales, las cuales contienen datos de altura de diversos terrenos.

Como resultado de las validaciones realizadas, se obtiene de esta memoria el software Runnel, una herramienta OpenSource y gratuita, que tiene como funcionalidad la visualización de cauces y detección de patrones de drenaje para la comunidad científica.

Agradecimientos

En primer lugar agradezco Aldo Canepa quién ha estado acompañándome en este camino brindándome apoyo, conocimiento y ánimo en mis metas. También, agradezco a su familia, quiénes han estado continuamente apoyándome mientras desarrollaba esta memoria.

Agradezco a mi familia por los años de apoyo en mis estudios y a Don Conejo por acompañarme ronroneando en las noches de programación.

Agradezco a mis profesoras guías Nancy y Luisa por la disposición y el apoyo a lo largo del desarrollo de esta memoria

También, agradezco a Jorge Godoy, Sergio Villarroel, a mis amigas y amigos del colegio y de la universidad por darme ánimo en los momentos que veía que no había un camino.

Finalmente, quisiera agradecer el financiamiento del proyecto Fondecyt N°1120495: “Improving the functionality and performance of meshing tools”.

Tabla de Contenido

Capítulo 1: Introducción	1
1.2 Motivación.....	2
1.3 Objetivo General.....	3
1.4 Objetivos Específicos	3
1.5 Metodología	4
1.6 Contenido de la memoria	5
Capítulo 2: Antecedentes	6
2.1 Conceptos básicos de Geología.....	6
2.2 Definiciones relacionadas con la computación gráfica.	8
2.3 Tecnologías relacionadas.....	9
2.4 Algoritmos	12
2.5 Herramientas Similares	14
Capítulo 3: Especificación del problema	19
3.2 Tipo de Usuarios.....	19
3.3 Requisitos del Usuario.....	19
3.4 Requisitos de Software.....	21
3.5 Matriz de trazado requisitos de usuario vs requisitos de software	23
3.4 Diagrama UML para los casos de uso	24
3.5 Casos de uso de alto nivel	27
Capítulo 4: Diseño e implementación de la solución	30
4.1 Estructura del Terreno	30
4.2 Obtención de datos.....	31
4.3 Construcción de la estructura.....	31
4.4 Algoritmos para la detección de ríos	32
4.5 Algoritmo para la identificación de patrones de drenaje	34
4.6 Recorrido del agua desde un determinado lugar	38
4.7 Funcionalidades de ayuda al programador	39
4.8 Archivo de datos propietario.....	40
4.9 Arquitectura del Software	41
4.10 Diagrama de Clases	45
4.11 Funcionamiento del Software	46
4.12 Métricas del software desarrollado.....	50
4.13 Elemento de ayuda al desarrollador.....	51
Capítulo 5: Validación de la solución	52
5.1 Plataforma de pruebas	52
5.2 Metodología	52
5.3 Resultados	53

Capítulo 6: Conclusiones	69
6.1 Trabajo Futuro	69
Bibliografía.....	71

Índice de ilustraciones

Imagen 1: Patrones de drenaje.....	6
Imagen 2: Clasificación de orden de Strahler-Horton.....	12
Imagen 3: Ejemplo de búsqueda por similitud parcial de polígonos.....	14
Imagen 4: Recorrido del agua desde un determinado lugar.....	15
Imagen 5: Vista en relieve.....	16
Imagen 6: Redes de drenaje en el área.....	16
Imagen 7: Red de drenaje seleccionada.....	17
Imagen 8: Mapa Online de México con la identificación de patrones de drenaje.....	18
Imagen 9: Sector de México con la identificación de las redes.....	18
Imagen 10: Patrones de drenaje.....	34
Imagen 11: Casos de cálculo.....	35
Imagen 12: Ángulo de unión.....	36
Imagen 13: Envoltura convexa.....	37
Imagen 14: Obtención de la gradiente.....	39
Imagen 15: Arquitectura de la aplicación.....	41
Imagen 16: Mapa de Google Earth.....	46
Imagen 17: Terreno en diferentes formas de visualización.....	47
Imagen 18: Opción de archivo tipo TIFF.....	47
Imagen 19: Identificación de ríos.....	48
Imagen 20: Ejemplo de identificación de red de drenaje.....	49
Imagen 21: Ejemplo con acercamiento de la identificación de redes de drenaje.....	49
Imagen 22: Ejemplo de identificación de patrón de drenaje.....	50
Imagen 23: Terreno con vectores normales y gradiente.....	51
Imagen 24: Sector Costero en Google Earth.....	53
Imagen 25: Algoritmo de Callaghan para la identificación de ríos en sector costero.....	54
Imagen 26: Algoritmo de Peucker para la identificación de ríos en sector costero.....	54
Imagen 27: Algoritmo por gradiente para identificación de ríos en sector costero.....	55
Imagen 28: Algoritmo de Callaghan para detección de red de drenaje en sector costero.....	55
Imagen 29: Algoritmo de Peucker para detección de red de drenaje en sector costero.....	56
Imagen 30: Algoritmo de ZhangGuilbert con Callaghan para detección de patrón de drenaje en sector costero.....	56
Imagen 31: Algoritmo de ZhangGuilbert con Peucker para detección de patrón de drenaje en sector costero.....	57
Imagen 32: Recorrido del agua por medio del algoritmo de Callaghan en sector costero.....	57
Imagen 33: Recorrido del agua por medio del algoritmo por gradiente en sector costero.....	58
Imagen 34: Sector de Petorca en Google Earth.....	58
Imagen 35: Algoritmo de Callaghan para identificación de ríos en sector de Petorca.....	59
Imagen 36: Algoritmo de Peucker para identificación de ríos en sector de Petorca.....	59
Imagen 37: Algoritmo por gradiente para identificación de ríos en sector de Petorca.....	59
Imagen 38: Algoritmo de Callaghan para detección de red de drenaje en sector de Petorca.....	60
Imagen 39: Algoritmo de Peucker para detección de red de drenaje en sector de Petorca.....	60
Imagen 40: Algoritmo de ZhangGuilbert con Callaghan para detección de patrón de drenaje en sector de Petorca.....	61
Imagen 41: Algoritmo de ZhangGuilbert con Peucker para detección de patrón de drenaje en sector costero.....	61
Imagen 42: Recorrido del agua por medio del algoritmo de Callaghan en sector de Petorca.....	62
Imagen 43: Recorrido del agua por medio del algoritmo por gradiente en sector de Petorca.....	62
Imagen 44: Datos extraídos del archivo TIFF.....	63

Imagen 45: Algoritmo de Callaghan para la identificación de ríos para archivo TIFF.....	63
Imagen 46: Algoritmo de Peucker para la identificación de ríos para archivo TIFF.....	64
Imagen 47: Algoritmo de ángulo diedro para la identificación de ríos para archivo TIFF.....	64
Imagen 48: Resultado de Rivertools para la obtención de las redes de drenaje para archivo TIFF..	64
Imagen 49: Algoritmo de Callaghan para detección de red de drenaje en archivo TIFF.....	65
Imagen 50: Algoritmo de Peucker para detección de red de drenaje en archivo TIFF.....	65
Imagen 51: Resultado de Rivertools para la obtención de redes de drenaje para archivo TIFF.....	66
Imagen 52: Algoritmo de ZhangGuilbert con Callaghan para detección de patrón de drenaje para archivo TIFF.....	66
Imagen 53: Algoritmo de ZhangGuilbert con Peucker para detección de patrón de drenaje para archivo TIFF.....	67
Imagen 54: Recorrido del agua por medio del algoritmo de Callaghan en archivo TIFF.....	67
Imagen 55: Recorrido del agua por medio del algoritmo por gradiente en archivo TIFF.....	68
Imagen 56: Resultado del recorrido del agua de RiverTools para archivo TIFF.....	68

Capítulo 1: Introducción

La geomorfología fluvial es el estudio de los procesos de sedimentación y formación de los ríos. El estudio de la formación de los ríos contempla, entre otras cosas, el cómo estos se ven desde los cielos en los distintos tramos a lo largo de su recorrido. La información proporcionada por los ríos, permite tener una visión general de la evolución geomorfológica y de los procesos que han ocurrido anteriormente en aquellos lugares. Con la información obtenida se puede detectar la presencia de paleo-volcanes, fracturas en las rocas, materiales presentes en la corteza del lugar, entre otros.

Mediante el estudio de las formas de los ríos, es posible clasificarlos según el tipo de red de drenaje que los identifica. Por red de drenaje se entiende el conjunto de ríos, vaguadas, barrancos y las huellas impresas en el terreno que deja la circulación constante e/o intermitente del agua.

A través de la información entregada por los patrones de drenaje es posible identificar fallas que se han ido creando en el tiempo, estado de las rocas del subsuelo e identificar el tipo de sedimento que se produjo en los caudales y ahora es parte del terreno. La necesidad de obtener esta información hace que la investigación e identificación de los lugares que presentan patrones sean de gran interés, como así, los sectores de los caudales donde no siguen el patrón representado. Otra información importante que es entregada por los caudales, son las terrazas fluviales. Estas se caracterizan por estar a la misma altura y a ambos lados de los caudales de agua, y son plataformas sedimentarias construidas por los propios sedimentos. Estas terrazas pueden indicar si las rocas que las contienen han sufrido un desnivel sin aparente relación con fracturas preexistentes.

Para los geólogos es importante identificar los lugares que presentan algún patrón de drenaje, ya que estos, registran cambios climáticos y tectónicos. También, es interesante si una parte del caudal no sigue el patrón de drenaje por el cual se identifica. Actualmente, los investigadores buscan los lugares a través de la inspección manual del mapa satelital de Google Earth, para posteriormente, analizar el lugar en terreno. En este contexto, se hace de gran ayuda, una plataforma que les permita identificar los lugares que presenten patrones de drenaje interesantes para sus investigaciones de forma automatizada aprovechando la potencia de los computadores actuales.

Las alternativas computacionales que hay actualmente para la identificación de patrones utilizando mapas satelitales son escasas. Existen aplicaciones que permiten la creación de modelos digitales de terrenos y su procesamiento para añadir elementos en la topografía pero no así la clasificación de los patrones y el poder ver las terrazas de lugares determinados. Dentro de estas implementaciones se encuentra Topo3, TopoCal, los cuales procesan datos topológicos para ingeniería. Además, existe un software de escritorio RiverTools [1], que tiene funcionalidades relacionadas con la detección de redes de drenaje. También, hay páginas web que proveen un mapa satelital con identificación de ríos, entre ellas está [2] con la identificación de 8 tipos de drenaje del

sector de México, creada por el Instituto Nacional de Ecología y Cambio Climático de México (INECC), donde utilizó datos pertenecientes a junio del 2003.

Cabe destacar, que con una herramienta que provea a la comunidad científica la posibilidad de identificar cuencas y clasificar las redes de drenaje de acuerdo a patrones, hará que las investigaciones se puedan enfocar en determinados lugares de mayor interés, agilizando el proceso de investigación y análisis pertinentes. Además, se pretende que la herramienta proporcionada cumpla con ser gratis y OpenSource a la comunidad, alojándose en un repositorio público.

1.2 Motivación

Actualmente, los geólogos realizan estudios geomorfológicos con el fin de determinar cambios climáticos y tectónicos en una zona. Por ejemplo, en el caso del sector de Petorca, La Ligua se han identificado anomalías en los patrones de drenaje y en la morfología, lo cual puede estar representando actividad tectónica reciente (menor a 400 mil años). Para encontrar este tipo de lugar u otros lugares que desean investigar, realizan una inspección manual a través de mapas satelitales como Google Earth. De este modo, una vez identificado el lugar a investigar, proceden a realizar pruebas en terreno para determinar los sedimentos que se encuentran en el caudal existente o que ya ha desaparecido. Estas pruebas sirven para medir la edad de los sedimentos y poder obtener información sobre la formación de estos.

Las aplicaciones que han sido creadas e interactúan con elementos morfológicos como Topo3 y TopoCal están desarrollados de tal manera que es posible la visualización de las mallas de polígonos, agregación de elementos morfológicos pero no así la identificación y clasificación de los patrones de drenaje, y menos aún, detectar anomalías en secciones que no correspondan al patrón identificado. Además, de ser softwares pagados si se desea obtener las funcionalidades completas de ellos. A su vez, las aplicaciones Web encontrada de INECC, cumplen con la opción de identificación de patrones de drenaje pero solamente para el país de México, donde los datos que almacena son estáticos y su uso se basa sólo en visualización. Por otro lado, el software RiverTool cumple con la necesidad de encontrar cauces y permiten realizar estudios científicos con las herramientas que provee. Una de estas herramientas, es el cálculo del recorrido que realiza el agua desde un punto arbitrario del mapa, con estos datos entregados por el software es posible ver la trayectoria que lo identifica, haciendo que se pueda realizar un análisis sobre el movimiento y comportamiento de ésta. A su vez, es posible identificar a la red que pertenece, pero no a la identificación de dicho patrón.

Debido a lo anterior, motiva a la creación de una aplicación que permita apoyar la identificación de cauces y patrones de drenaje. También a la identificación de secciones de un caudal que difieren del patrón al cual pertenecen, con el fin de encontrar puntos de interés científico de forma automatizada y eficiente.

En el contexto de este trabajo de titulación se pretende satisfacer esta necesidad a la comunidad científica y no científica con una aplicación desarrollada bajo el marco OpenSource y gratuita. De esta forma, la aplicación quedará abierta al uso de la

comunidad y a cualquier contribución que realice otro desarrollador para extenderla en sus funcionalidades, ya que se pretende que este software abierto sea un acercamiento a este estudio.

1.3 Objetivo General

Desarrollar y proveer a la comunidad científica una herramienta interactiva, de código abierto y gratuito, que permita la obtención y procesamiento de mapas de alturas para la identificación de las causas de los patrones de drenaje, tales como dendrítico, pinado, rectangular, angular, los cuales según el tipo, pueden dar un indicio de los procesos tectónicos producidos en el lugar.

1.4 Objetivos Específicos

A continuación se presentan los objetivos específicos que se desprenden del objetivo general. Los que aparecen marcados con (✓) se cumplieron y los que aparecen con (x) no son abordados.

- ✓ Extracción y procesamiento de datos de elevación de base de datos online (Google Earth) y archivos TIFF
- ✓ Creación de la malla de polígonos con los datos extraídos de la base de datos online
- ✓ Creación de la malla de polígonos con los datos extraídos de un archivo local
- ✓ Identificación de caudales
- ✓ Identificación de patrón de drenaje según los caudales calculados
- ✓ Visualización de los patrones de drenaje en mapa de Google Earth
- ✓ Visualización de los patrones de drenaje en la malla de polígonos
- ✓ Implementación de una Interfaz interactiva, la cual permita al usuario las siguientes opciones:
 - ✓ Guardar la malla de polígonos junto con la identificación de patrones de drenaje
 - ✓ Visualizar información adicional explicativa sobre las características de los patrones
 - ✓ Ver la malla de polígonos y tener la opción de ver su vista satelital.
 - ✓ Selección de área en la vista satelital y su posterior obtención de la malla de polígonos
 - ✓ Movimiento de cámara como rotación, zoom y traslación
 - ✓ Identificación de posible patrón de drenaje
- ✓ Software OpenSource
- ✓ Software multiplataforma, en especial que sea compatible con Windows y Linux.

Esta herramienta es para ser utilizada con fines educativos para proveer a la comunidad científica, un producto mantenible y extensible, siguiendo los patrones de diseño necesarios. [3]

1.5 Metodología

El plan de trabajo que se sigue para poder alcanzar los objetivos propuestos en la memoria se divide en 3 etapas: Investigación, Diseño y Desarrollo.

1.5.1 Investigación

Esta primera etapa se enfocó en los antecedentes que se presenta en este documento, los cuales involucran una revisión bibliográfica de las tecnologías y algoritmos necesarios para solucionar el tema propuesto. Dentro de las tecnologías y algoritmos se encuentran las siguientes:

- Qt y QtCreator para la creación de las interfaces que harán posible su buen funcionamiento en distintos Sistemas Operativos
- OpenGL para renderización en aplicaciones de escritorio
- Shaders para la programación en tarjetas de video
- Distintos tipos de triangulación para las mallas de polígonos, en especial la triangulación de Delaunay
- Distintos tipos de visualizadores de malla
- Google Earth y la extracción de coordenadas de elevación
- Algoritmos de detección de flujo de caudal en mallas de superficie
- Distintos algoritmos de clasificación por similitud de atributos
- Algoritmos de clasificación y aprendizaje para identificar el tipo de red de drenaje
- Distintos tipos de presentación de la información al usuario
- Análisis de los distintos patrones de diseño que podrán ser utilizados

1.5.2 Diseño

En esta segunda etapa, se crea un sistema extensible, que permita fácilmente la agregación de nuevos tipos de patrones de drenaje y distintos tipos de clasificadores para estos. También la elección de patrones de diseño que permita el desarrollo modular de la aplicación para que satisfaga los requerimientos. Se diseñan las estructuras de datos utilizadas en la aplicación para la implementación eficiente y efectiva de los algoritmos.

1.5.3 Desarrollo

En esta tercera parte del plan de trabajo se pretende utilizar los conocimientos adquiridos de la exhaustiva investigación para desarrollar la herramienta. A continuación se presentan el orden del desarrollo:

1. Manejador de extracción de las coordenadas y elevación desde Google Earth
2. Visualizador del mapa satelital con los datos extraídos
3. Visualización de la malla de polígonos con los datos extraídos
4. Identificación de caudales por medio de polilíneas en el mapa satelital y la malla de polígonos
5. Identificación de los patrones de drenaje dentro de las polilíneas encontradas

6. Opción de guardado de malla con las identificación de los patrones elegidos por el usuario
7. Opción de cargar la malla/mapa satelital con los datos preexistentes
8. Identificación a los sectores de los caudales que no corresponden al patrón predominante
9. Refactorización para la incorporación del patrón Strategy
10. Presentación de la información adicional
11. Escribir el documento de Memoria

Para su correcto funcionamiento se evalúa las mallas de polígonos con los resultados obtenidos de la aplicación RiverTool. Cabe destacar, que el desarrollo es de forma iterativa e incremental, validando lo investigado y diseñado con lo implementado, haciendo que en cada etapa se generen prototipos, los cuales son evaluados y corregidos de ser necesario.

1.6 Contenido de la memoria

A continuación se listan los contenidos de los capítulos dentro de esta memoria:

1. **Introducción:** Se realiza la presentación del tema, la motivación la cual conlleva a realizar este tema y se da a conocer los objetivos.
2. **Antecedentes:** Se presentan los conceptos, algoritmos, y tecnologías necesarias que se fueron investigando para entender el tema. Además, se presentan algunos patrones de drenaje.
3. **Especificación del problema:** Se enuncia las alternativas de soluciones existentes, los requerimientos y casos de uso.
4. **Descripción de la solución:** En este apartado se da a conocer como fue implementada la solución, que algoritmos fueron utilizados, la arquitectura y métricas del software. Además, de algunas funcionalidades del software.
5. **Validación de la solución:** Se realiza la comparación con el software Rivertools, la base de datos de los ríos y el software implementado. Además, se da a conocer como se desempeña el software en la identificación de patrones de drenaje
6. **Conclusiones:** Se describe las conclusiones del trabajo realizado y el trabajo posterior a seguir.
7. **Referencias:** Están descritas las fuentes que se utilizaron en la investigación e implementación de este tema de memoria

Capítulo 2: Antecedentes

Para la realización de la memoria, se debió investigar tecnologías y conceptos asociados al tema.

2.1 Conceptos básicos de Geología

La información presentada en este apartado corresponde a las redes de drenaje que se ha investigado, la clasificación de Strahler-Horton, el sistema de coordenadas WGS84 y archivo tipo TIFF.

2.1.1 Redes de Drenaje

Por red de drenaje se entiende el conjunto de ríos, vaguadas, barrancos y las huellas impresas en el terreno que deja la circulación constante e/o intermitente del agua. A continuación se presentan algunas de las redes de drenaje [3] [4]:

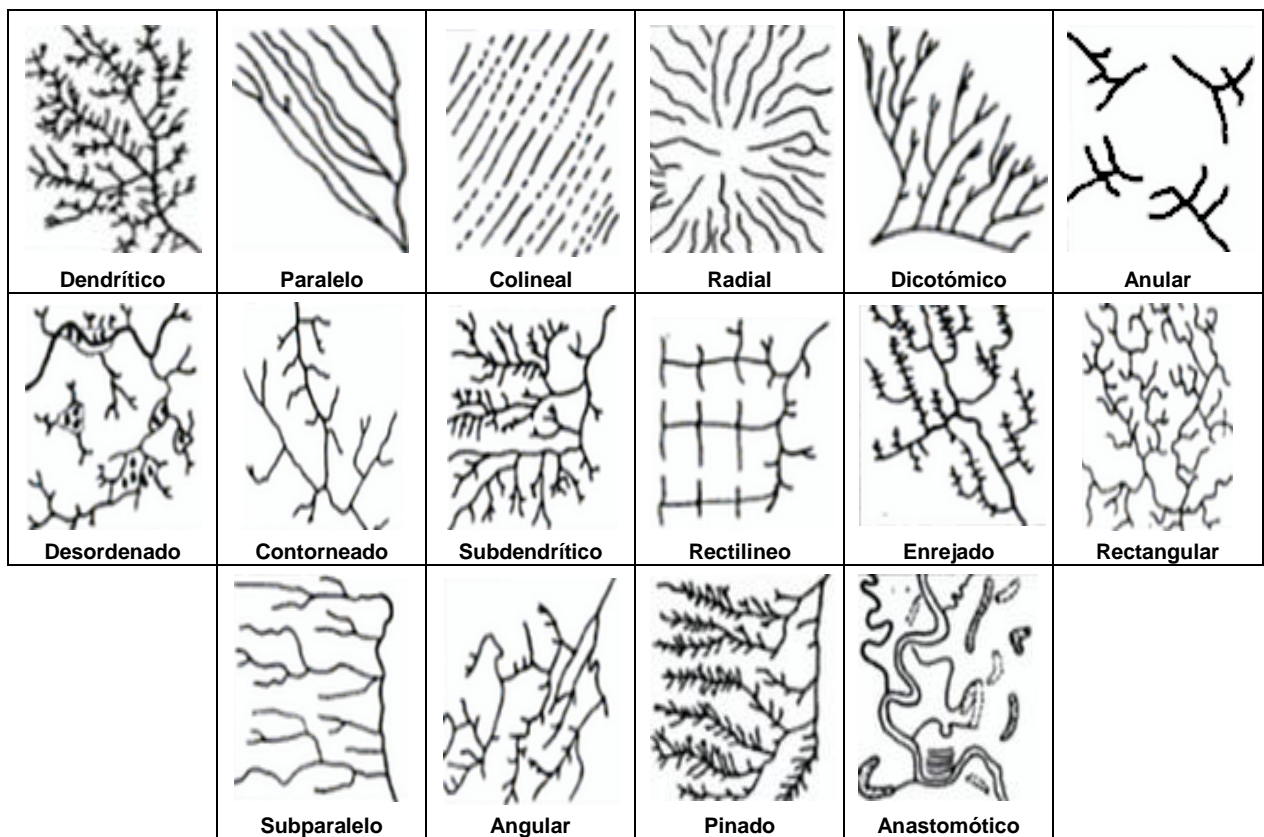


Imagen 1: Patrones de drenaje

Dendrítico: Es la forma más común de los sistemas de drenaje, posee muchas corrientes que contribuyen al río principal. Su forma principal es análoga a las ramas de un árbol

frondoso. El tipo de roca que desarrolla el patrón presenta una resistencia a la erosión uniforme, siendo la roca impermeable y no porosa.

Paralelo: Este patrón es causado por fuertes pendientes con algunos relieves, debido a esto las corrientes son rápidas y rectas, con poco afluente de agua y una misma dirección de movimiento. También este patrón está presente en fallas grandes de las rocas.

Colineal: Indica antiguos cauces que ya no existen o parcialmente desaparecidos. Comúnmente está presente en zonas de baja pendiente y con posibles avenidas temporales.

Radial: Red de drenaje donde las corrientes circulan desde un punto central alto. Suele desarrollarse en volcanes, donde este patrón aparece muy definidamente. Sus corrientes y valles se desplazan hacia fuera y abajo, desde distintos puntos alrededor del cono.

Dicotómico: Patrón que se caracteriza por la ramificación de un cauce de varias afluentes de agua, las cuales se subdividen a su vez.

Anular: Patrón en el cual los arroyos siguen un camino más o menos circular o concéntrico a lo largo de una franja de roca débil. Suelen ser arroyos donde la erosión deja en descubierto los estratos sedimentarios de diversos grados de dureza.

Desordenado: En este patrón el agua converge hacia pequeños lagos o pantanos sin un camino fijo. Ejemplo de este tipo de patrón son los depósitos glaciales denominados drift.

Contorneado: Patrón que se presenta en rocas metamórficas formadas en capas gruesas y contorneadas o plegadas. Ocasionalmente algunas de las desviaciones de las corrientes se deben a la presencia de diques y flujos de origen magmático que están en zonas resistentes a la erosión.

Subdendrítico: Es similar al patrón dendrítico, pero con la salvedad que las corrientes del río principal se unen a él en ángulos agudos. Lo último se debe al efecto empinado de las laderas sobre las cuales se desarrollan las corrientes del río.

Rectilíneo o Artificial: Patrón presentado en sectores planos y mal drenados, donde el efecto de la existencia del ser humano ha hecho que aparezcan surcos amplios y rectos.

Enrejado o Trellis: Es similar al patrón rectangular, con la diferencia de que presenta rocas que difieren en su resistencia a la erosión en el subsuelo. También se puede encontrar este tipo de redes en áreas de fracturas paralelas.

Rectangular: Patrón que se origina en rocas que son uniformes en la resistencia a la erosión con 2 tipos de dirección del afluente del agua. Además, estas rocas están cruzadas por fracturas casi perpendiculares entre sí.

Subparalelo: Patrón compuesto por rocas sedimentarias plegadas y alineadas en largos tramos por una tendencia regional de plegamiento. Estas rocas sedimentarias tales como areniscas, lutitas, etc.

Angular: Patrón formado la roca madre y las fallas en las rocas se cruzan en ángulos más agudos que en los patrones de drenaje rectangular.

Pinado: Patrón en forma de pino, el drenaje recto o semirectos puede ocurrir donde hay presencia de materiales limo-arcillosos

Anastomótico: Patrón que tiene por característica los lechos de creciente en proceso de estabilización. Los meandros¹ del cauce principal han hecho en el valle cauces abandonados, bateas, pequeñas lagunas, etc.

2.2 Definiciones relacionadas con la computación gráfica.

2.2.1 Triangulación

Se define la triangulación como la partición de un dominio de puntos en triángulos, cumpliendo que los vértices de los triángulos son los puntos originales de la nube y que toda dupla de triángulos comparte a lo más un vértice o una arista. Existen varios tipos de triangulación, dependiendo la necesidad que se tenga. A continuación se presenta algunas de las distintas triangulaciones [8]:

2.2.1.1 Triangulación de peso mínimo (MWT)

Una triangulación de peso mínimo (MWT) es la triangulación completa de una nube de puntos, donde cada vértice es conectado con el vértice de menor peso, donde el peso está definido por la distancia euclidiana de los vértices que compondrán la arista del triángulo. Si esta arista es la de menor peso dentro de las aristas que comparte el mismo vértice, se dice que es una arista ligera.

2.2.1.2 Triangulación de Greedy (Voraz)

La construcción de la triangulación se basa en ordenar las aristas de forma creciente en su tamaño y posteriormente a eso, las va agregando al grafo, tal que no haya ninguna arista que intercepte a otra, en caso de haberla se descarta.

2.2.1.3 Triangulación en abanico

A partir de un punto p , el cual se obtiene por medio del orden de los puntos de forma ascendente según su ordenada, se ordena el resto de los puntos con respecto a p por el ángulo que forman con éste. Esto conlleva a que se forme un abanico de triángulos desde

¹ Dícese de la formación provocada por el río, la cual se observa como una curva pronunciada.

p. Posteriormente, se procede a crear aristas en los otros vértices consecutivos recursivamente [9].

2.2.1.4 Triangulación de Delaunay

En la triangulación de Delaunay los triángulos que la componen poseen ángulos internos lo más grandes posible y la longitud de las aristas es mínima. Tiene las siguientes propiedades:

- La triangulación minimiza tanto el ángulo máximo como el radio de las circunferencias de sus triángulos.
- La circunferencia circunscrita de cada triángulo no contiene ningún punto dentro de ésta. [10]
- Dos puntos de la nube de puntos son vértices de un mismo triángulo de la triangulación de Delaunay si y sólo si puede trazarse un círculo, el cual contenga sólo estos 2 puntos.
- Los triángulos se asemejan en lo posible a triángulos equiláteros.

Actualmente, existe la implementación de esta triangulación en C++ [11] y en Matlab [12]. Esta triangulación tiene el beneficio que posee varias propiedades a comparación con los otros tipos de triangulación. Es por ello que esta triangulación es útil para poder modelar terrenos.

2.3 Tecnologías relacionadas

2.3.1 Archivos TIFF

El archivo de tipo “Tagged Image File Format” (TIFF) es un formato de imagen estándar en el tratamiento de imágenes. Se caracteriza por contener una serie de etiquetas o entradas que permiten almacenar información acerca de la georreferenciación² de la imagen. Algunos de los atributos que se pueden obtener de un archivo TIFF son la posición, ancho y largo de la imagen, y los colores correspondientes a los puntos entre otros. [5]

2.3.2 Mapas

Google Maps es un servidor de aplicaciones de mapas en la web perteneciente a Google, el cual provee a la comunidad de mapas digitales donde se puede obtener las coordenadas de latitud y longitud, distancia entre 2 puntos, entre otros [13]. Además, posee una API, la cual es compatible con diferentes browser como Internet Explorer, Firefox, Safari, Chrome.

² Dícese de la técnica de posicionar un objeto en una única localización geográfica.

Google Earth es un mapa satelital que está disponible para diversas plataformas, versión de escritorio, móvil y web. Cada una de estas es posible observar mapas satelitales y digitales, de la misma forma el poder ver mapas de relieve en 3D.

Google Earth, posee una Api de elevación del terreno [14], con la cual es posible obtener la elevación del terreno, pero posee la limitante de que al día se pueden realizar 2500 solicitudes. Otra de las opciones es la utilización de la función *GEGlobe.getGroundAltitude(double lat, double lon)*, en la versión web de Google Earth, la cual permite obtener la altura dada una latitud y longitud. Utilizando esta última función y dado un par de coordenadas, se podrían extraer los datos de elevación de una grilla de puntos iterativamente en un entorno web con la utilización de *javascript*

2.3.3 Bibliotecas multiplataforma.

En este apartado se presenta las bibliotecas multiplataforma que fueron consideradas:

2.3.3.1 Qt

Es una biblioteca multiplataforma gratuita desarrollada en C++ y orientada a objetos. Permite desarrollar aplicaciones de escritorio con interfaz gráfica para los usuarios (GUI), utilizando funciones propias para emular el comportamiento del sistema operativo. Estos sistemas operativos son Android, IOS, Linux/X11, Mac OS x, Windows y WinCE. Los lenguajes de programación con los cuales se puede trabajar realizando un binding en Qt son C#, PHP, Python, Ruby entre otros.

La librería tiene un ambiente de desarrollo integrado (IDE) llamado QtCreator, el cual permite a los desarrolladores acceder a bases de datos, manipular datos en formato XML, soporte de red, manejo de archivos, manejo de compiladores, entre otros. Todo esto, permite un desarrollo ágil, sin la preocupación para cual sistema operativo se esté trabajando. Además, QtCreator tiene asociada la herramienta Qt Designer, la cual provee de una interfaz gráfica para diseñar e implementar la GUI que se necesitará.

Cabe destacar, que OpenGL es compatible con esta biblioteca multiplataforma y posee una extensa documentación soportada por la comunidad [15].

2.3.3.2 WxWidgets

Es una biblioteca multiplataforma gratuita de C++ donde se puede crear aplicaciones para Windows, Mac OS X, Linux en arquitectura de 32 y 64 bits. Además, es OpenSource y puede ser utilizada desde otros lenguajes de programación como Java, JavaScript, Perl, Python, Smalltalk y Ruby. WxWidgets utiliza la API nativa del sistema operativo en vez de emularla, haciendo que la aplicación tenga un comportamiento y aspecto nativo. Para realizar aplicaciones gráfica en 3D utiliza la clase *wxGLCanvas*, la cual interactúa con OpenGL [16].

2.3.3.3 GTK+

Es un conjunto de bibliotecas multiplataforma escrita en C para el desarrollo de GUI en Windows de 32 y 64 bits, Linux y Mac OS X. GTK+ soporta otros lenguajes como C++, C#, Java, Python, JavaScript, Vala y Perl. Dentro de las bibliotecas que contiene se encuentran: [17]

- **Glib**: Biblioteca que maneja la estructura de datos para C, portabilidad, ciclos, hilos.
- **GTK**: biblioteca encargada de los objetos y la interfaz de usuario.
- **GDK**: Biblioteca intermediaria para gráficos de bajo nivel y de alto nivel.
- **ATK**: biblioteca para la creación de interfaces para la accesibilidad a personas discapacitadas. Por ejemplo, para el uso de lectores de pantalla o teclados alternativos.
- **Pango**: Biblioteca para el diseño y renderizado de texto, manejo de las fuentes.
- **Cairo**: Biblioteca de renderizado de controles de aplicación

Para la elección de la biblioteca, la cual se utiliza para que la aplicación fuera multiplataforma, se basó principalmente en la documentación encontrada y la cantidad de plataformas que soportaba. Además de lo anterior, se investigó la posibilidad de la visualización en 3D y en cada una de ella estaba la posibilidad de utilizar OpenGL. Resultó ser que Qt posee una extensa documentación, mientras que en GTK+ está programada en C, donde no tiene la orientación de objetos como C++ y WxWidgets posee una documentación limitada.

2.3.4 Software Gráfico

2.3.4.1 Qhull

Qhull es un software que permite el cálculo de la triangulación de Delaunay, diagrama de Voronoi entre otros algoritmos. El cálculo se puede realizar para objetos en 2D, 3D y 4D, donde sus superficies no sean convexas.

2.3.4.2 OpenGL

Es una especificación de una API multiplataforma para aplicaciones con gráficos 2D y 3D. Esta API provee un encapsulamiento de las funciones de distintas tarjetas gráficas y está basada en procedimientos de bajo nivel para renderizar una imagen, teniendo como misión transformar las primitivas como puntos, líneas, polígonos en píxeles [18]. Este proceso es realizado por el pipeline, el cual antiguamente era un proceso rígido, donde el usuario sólo podía escoger parámetros de configuración para buscar el resultado deseado. En la actualidad la existencia de shaders, hace que la tarjeta de video pueda ser programable, agregando un mayor grado de libertad en poder realizar cálculos y efectos personalizados sobre las primitivas que pasan por el pipeline de forma muy eficiente.

2.3.4.3 Shaders

Los shaders son programas que trabajan a nivel de vértices (Vertex Shader), primitivas (Geometry Shader) y fragmentos (Fragment Shader), con un gran grado de paralelismo, sobre la estructura geométrica, la cual se está trabajando para ser renderizada por la tarjeta gráfica (GPU). Estos programas pueden ser creados por uno mismo, haciendo que el proceso de renderizado no sea rígido, aumentando así, el rendimiento y la flexibilidad [19].

2.4 Algoritmos

2.4.1 Clasificación de Strahler-Horton

La clasificación de Strahler-Horton considera los cauces en un determinado orden. El primer orden son aquellos cauces que no reciben agua, sino por el escurrimiento de la vertiente en épocas de lluvia. En segundo orden se encuentran los que unen a los de primer orden, siendo los de tercer orden los que unen a los de segundo orden y así sucesivamente. Además, esta clasificación es importante para el desarrollo de algunos algoritmos de detección de patrones de drenaje.

A continuación se presenta una imagen que clarifica dicha clasificación [7]:

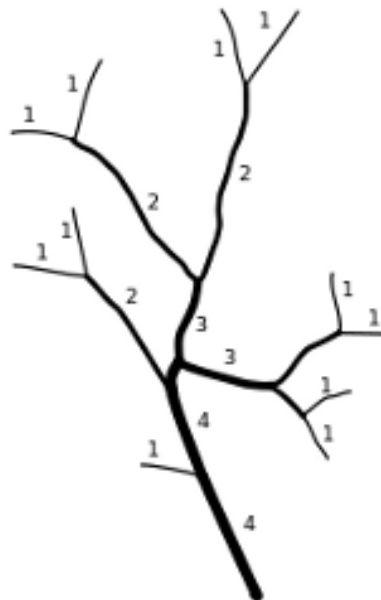


Imagen 2: Clasificación de orden de Strahler-Horton

En la imagen se observa que los cauces extremos corresponden a los de primer orden. Mientras que la suma de dos caudales del mismo orden conlleva a la creación de un caudal de un orden mayor.

2.4.2 Algoritmos para la detección de redes de drenaje y clasificación

En [20] se describe un algoritmo, el cual divide el río en sub-partes. Para ello, se define que cada red de drenaje, tiene una estructura jerárquica, la cual puede ser descrita por varias partes del río, donde cada parte puede ser representada por nodos y por líneas. Posteriormente, comienza a ordenar las sub-partes de la red de drenaje con la clasificación de Strahler-Horton, mencionada anteriormente.

Por otro lado, en [21] se describe un algoritmo que hace uso de Support Vector Machine (SVM) y logra detectar los ríos de hielo, la cual podría ser adaptada para que identifique ríos en general. El uso de SVM es porque es una técnica útil para clasificar información y minimiza el error promedio de los datos, mientras está maximizando el margen entre las clases, llamándose clases a los distintos grupos que se pueden encontrar en una clasificación.

SVM viene dado por la siguiente formula:

$$y(x, W) = \sum_{m=0}^M w_m * K(x, x_m)$$

Donde W son todos los parámetros del modelo, M la cantidad de parámetros y $K(x, x_m)$ alguna función de Kernel

Algunas de las funciones de Kernel que fueron utilizadas en el paper:

- **Polinomial Kernel:** $k(x, x_i) = (\gamma x * x_i)^d$
- **RBF Kernel:** $\exp(-\gamma ||x - x_i||^2)$

Para cada una de ellas, aplico 2 criterios para determinar la precisión de la detección en los ríos. Estas fueron:

- **Detection rate:** El número de objetivos que fueron correctamente detectados divididos por los ríos de hielo existentes en la imagen.
- **Correct rate:** El número de objetivos que fueron correctamente detectados divididos por el número de objetivos que fueron reconocidos.

Cabe destacar, que un alto valor en los criterios significa un buen funcionamiento.

También, para reconocer patrones de drenaje se puede desarrollar en base a búsqueda por similitud parcial de polígonos expuesta en el paper [22]. Comienza con la definición del polígono que se desea comparar, sus vértices y aristas. Posterior a eso, se define una función que parametriza una sección del polígono, la cual será utilizada para ser comparada con el polígono de búsqueda. Estos polígonos serán congruentes si y solo si, sus funciones de parametrización se asemejan. Se define la función de parametrización

como la métrica de un punto hacia el polígono. Entre menor sea la diferencia entre las distancias de los polígonos, estos más se parecerán.

A continuación se visualiza esta búsqueda aplicado a una polilínea:

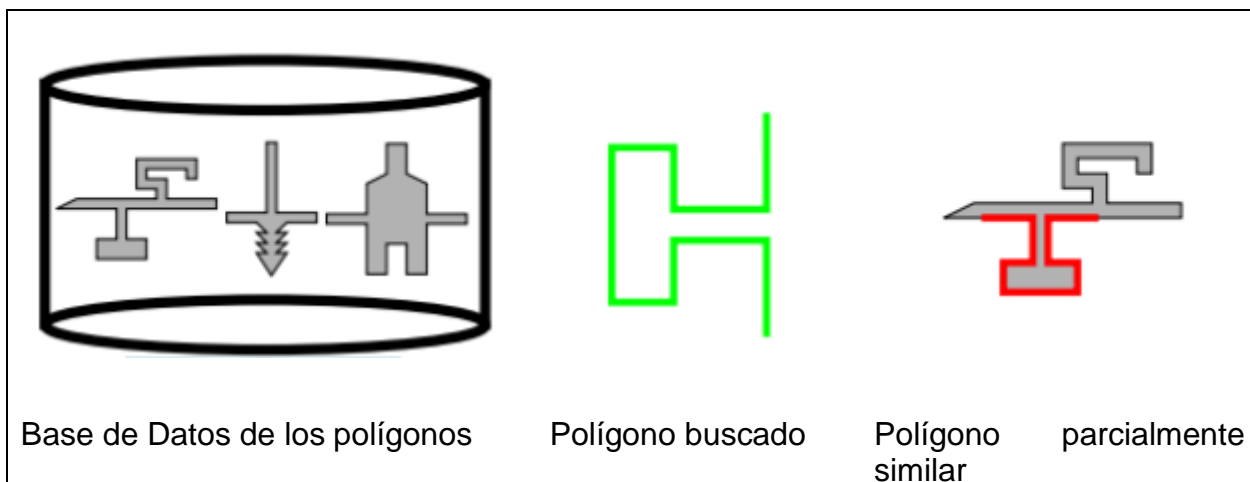


Imagen 3: Ejemplo de búsqueda por similitud parcial de polígonos.

2.5 Herramientas Similares

A continuación se presentan las herramientas que realizan algún tipo de análisis asociado a las redes de drenaje que son de interés.

2.5.1 RiverTools

RiverTools provee a la comunidad la obtención tanto de un mapa en relieve, como de la vista panorámica superior. A su vez, es posible la identificación del recorrido que realiza el agua desde un determinado lugar.

A continuación se presenta visualmente las características mencionadas anteriormente:

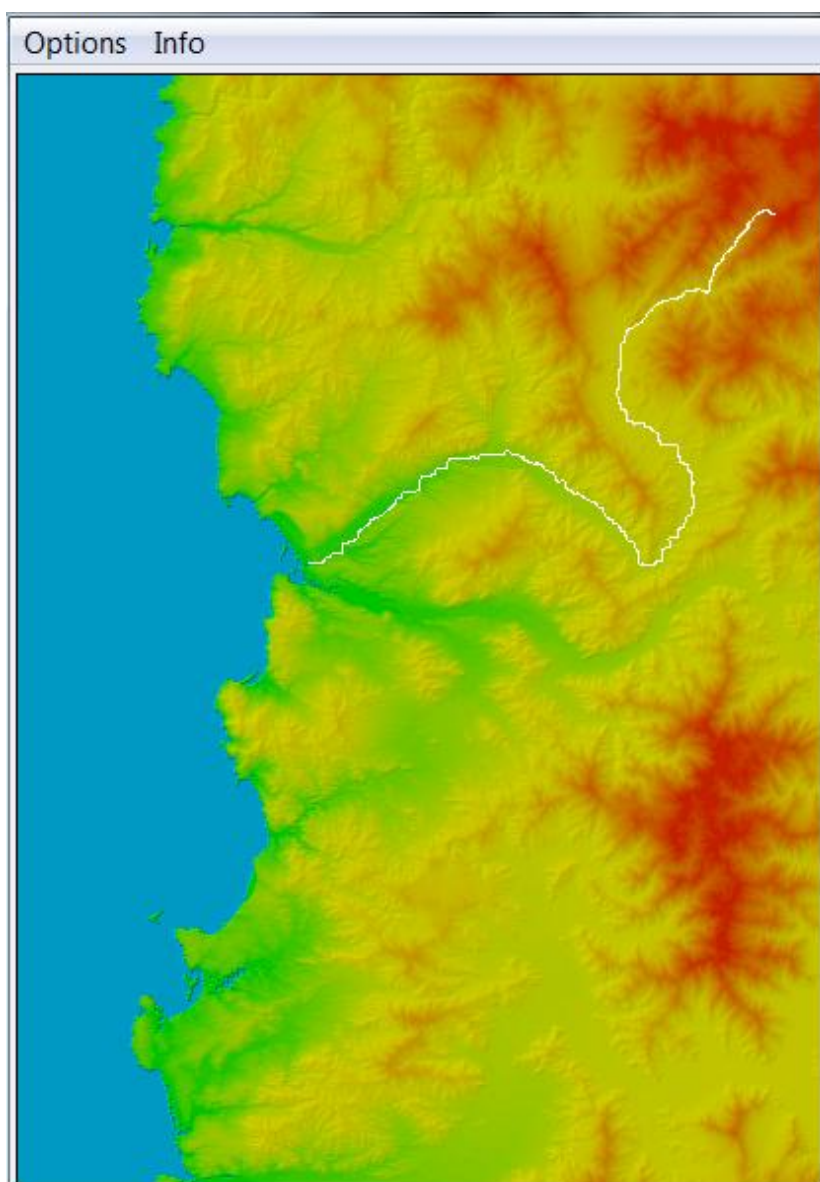


Imagen 4: Recorrido del agua desde un determinado lugar

De la imagen con vista superior del terreno se logra apreciar la línea blanca correspondiente al camino que realiza el agua desde un determinado lugar.

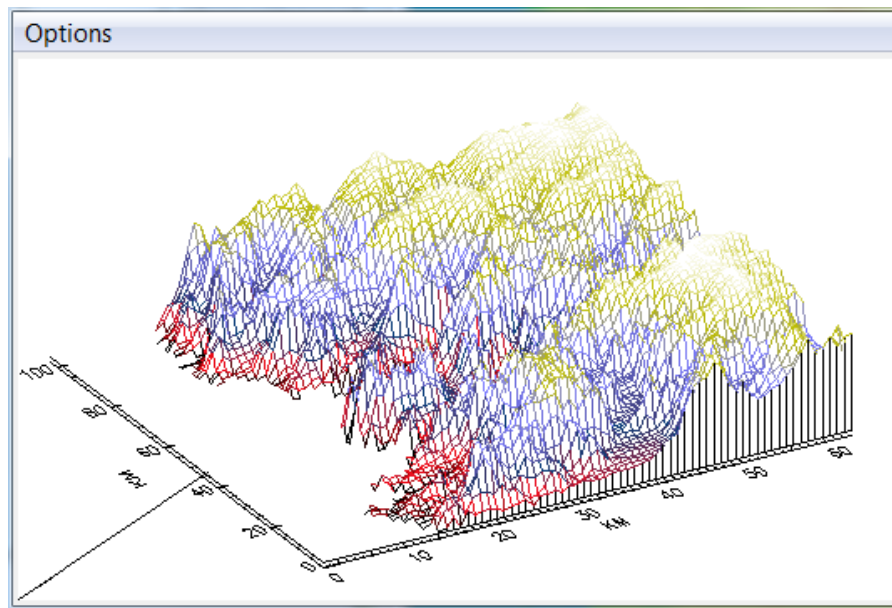


Imagen 5: Vista en relieve

La anterior imagen muestra el terreno en relieve. Donde el color más amarillo representa los sectores que están a mayor altura.

Otras de las funcionalidades que posee Rivertools es el poder obtener las redes de drenaje del terreno que se está analizando.

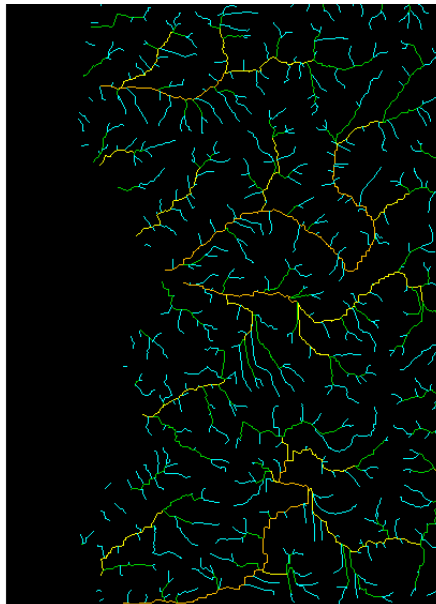


Imagen 6: Redes de drenaje en el área

Donde los colores de cada parte de la red corresponde a la clasificación de Strahler-Horton mencionada anteriormente.

También está la posibilidad de ver una red de drenaje en específico con la clasificación de Strahler-Horton. Esta red de drenaje correspondiente a la cual se realizó la operación de obtener el recorrido del agua desde un punto. A continuación se presenta la imagen:

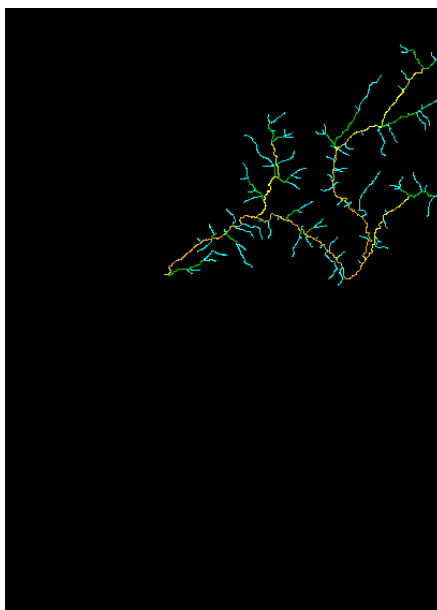


Imagen 7: Red de drenaje seleccionada.

2.5.2 Sitio Web INECC

En el capítulo anterior, se comentó la existencia de un sitio web perteneciente al Instituto Nacional de Ecología y Cambio Climático de México, el cual permite la identificación de algunos patrones de drenaje en México. En este sitio web es posible ver las redes de la localidad de México, donde en el costado inferior izquierdo aparece la clasificación correspondiente a cada una, mientras que en el mapa están marcadas según el color que las identifica.

A continuación se muestra el sitio web:

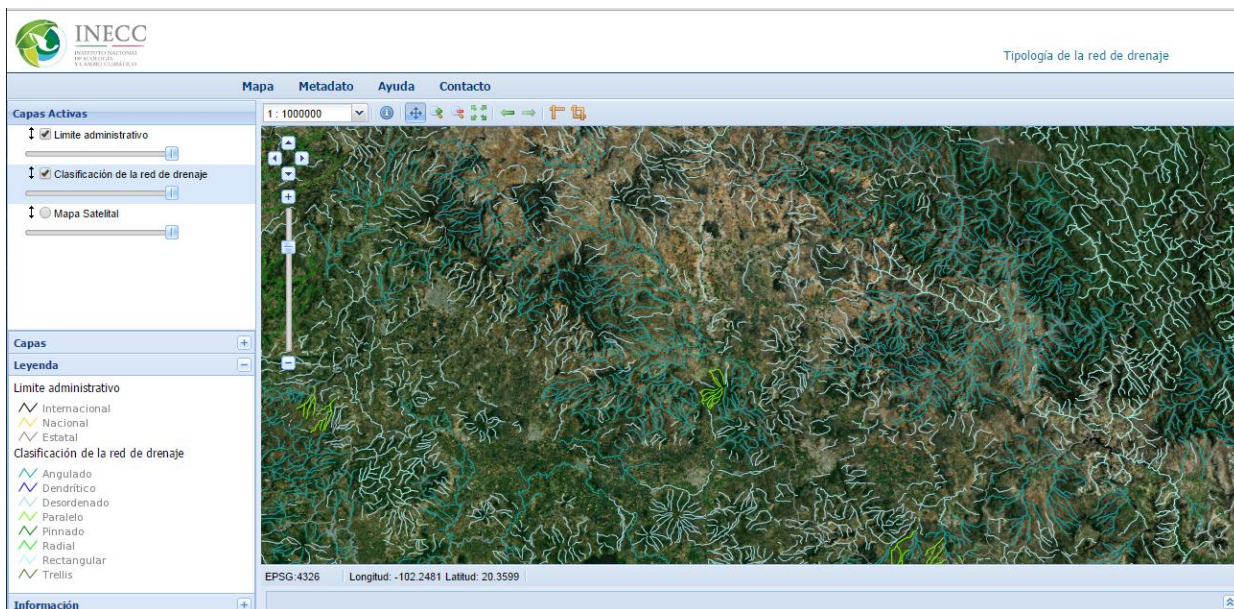


Imagen 8: Mapa Online de México con la identificación de patrones de drenaje

A continuación un sector del terreno presentado, con la identificación de las redes.



Imagen 9: Sector de México con la identificación de las redes

Capítulo 3: Especificación del problema

Actualmente en geología se hace uso de la aplicación de Google Earth para analizar los lugares de interés de forma visual. Una vez identificado el lugar de interés, se procede a georreferenciar este lugar con un archivo tipo TIFF y luego se analiza con el software RiverTools. Aunque Rivertool es la solución más aproximada a la necesidad que se tiene, el cual mantiene funcionalidades importantes las cuales se explicarán más adelante. Aun así, no es posible la selección de un terreno por Google Earth de forma automatiza y menos aún la identificación de patrones de drenaje de la misma forma. Cabe destacar, que este software es de distribución pagada y su código no es libre.

Principalmente, la necesidad que hay es disponer de un software de código abierto y gratuito que permita a la comunidad de geología analizar lugares de forma automatizada, logrando así, un enfoque más expedito en lugares de mayor interés.

El software debe permitir ver el mapa en 3D y poder seleccionar desde Google Earth áreas para dicha acción. Para cada lugar de interés que se esté investigando, obtener los cauces, el recorrido que realiza el agua desde un determinado lugar, la identificación de la red de drenaje y su patrón asociado. Con el fin, de poder saber si el lugar de interés registra algún cambio climático y/o tectónico.

3.2 Tipo de Usuarios

El sistema presenta dos tipos de usuarios:

Investigador: Usuario que maneja conocimientos en sistemas de drenaje y desea utilizar la herramienta con fines científicos.

Programador: Usuario que implementa nuevas funcionalidades y está en contacto con las características del software.

3.3 Requisitos del Usuario

3.3.1 Requisitos funcionales

RU0001: Visualizar terreno en Google Earth	
Descripción: Cada usuario podrá visualizar terrenos en el mapa de Google Earth a través de la interfaz.	
Tipo	Funcional
Tipo de usuario asociado	Investigador y programador

RU0002: Visualizar terreno en OpenGL	
Descripción: Cada usuario podrá visualizar terrenos en OpenGL, provenientes de datos extraídos por Google Earth y archivos TIFF.	
Tipo	Funcional
Tipo de usuario asociado	Investigador y programador

RU0003: Cargar de datos	
Descripción: Cada usuario podrá cargar datos provenientes de archivos TIFF, Runnel y datos extraídos de la api de Google Earth.	
Tipo	Funcional
Tipo de usuario asociado	Investigador y programador

RU0004: Guardar datos	
Descripción: Cada usuario podrá exportar los datos a un archivo tipo runnel.	
Tipo	Funcional
Tipo de usuario asociado	Investigador y programador

RU0005: Identificación de cauces.	
Descripción: Cada usuario podrá utilizar la opción de identificación de cauces en el terreno y este será visualizado en OpenGL y Google Earth.	
Tipo	Funcional
Tipo de usuario asociado	Investigador y programador

RU0006: Identificación de patrón de drenaje.	
Descripción: Cada usuario podrá utilizar la opción de identificación de patrón de drenaje en los cauces que han sido identificados previamente, y estos serán visualizados en OpenGL y Google Earth.	
Tipo	Funcional
Tipo de usuario asociado	Investigador y programador

RU0007: Visualizar información explicativa sobre los patrones.	
Descripción: Cada usuario podrá visualizar la información explicativa de los patrones que están siendo identificados.	
Tipo	Funcional
Tipo de usuario asociado	Investigador y programador

RU0008: Mover el terreno visualizado en OpenGL.	
Descripción: Cada usuario podrá mover la cámara del terreno, teniendo como opción el poder rotar, hacer zoom y trasladar.	
Tipo	Funcional
Tipo de usuario asociado	Investigador y programador

3.3.2 Requisitos de calidad.

RU0009: Interfaz interactiva.	
Descripción: El software tiene que ser fácil de utilizar e intuitivo, permitiendo la posibilidad de elegir la configuración de qué tipo de identificación de río desea.	
Tipo	Calidad
Tipo de usuario asociado	Investigador y programador

3.3.3 Requisitos de restricción.

RU0010: Software OpenSource.	
Descripción: El software tiene que ser OpenSource. Para ello, el código del software tiene que estar alojado en algún repositorio público.	
Tipo	Restricción
Tipo de usuario asociado	Programador

RU0011: Software multiplataforma.	
Descripción: El software tiene que ser compatible con Windows y Ubuntu.	
Tipo	Restricción
Tipo de usuario asociado	Programador

3.4 Requisitos de Software

RS0001: Cargar mapa de Google Earth.	
Descripción: El software tiene que tener las librerías necesarias para que pueda desplegarse y visualizar el mapa de Google Earth en la aplicación.	
Tipo	Funcional
Tipo de usuario asociado	Programador

RS0002: Opciones de visualización para el mapa de Google Earth.	
Descripción: El software tiene que tener las opciones de movimiento en el mapa de Google Earth y la creación de polilíneas para insertar polígonos.	
Tipo	Funcional
Tipo de usuario asociado	Programador

RS0003: Cargar el canvas para OpenGL.	
Descripción: Funcionalidad encargada de verificar que el software posea las librerías necesarias para que se pueda desplegar un terreno en OpenGL.	
Tipo	Funcional
Tipo de usuario asociado	Programador

RS0004: Opciones de visualización para el terreno en OpenGL.	
Descripción: El software tiene que tener las opciones de movimiento de cámara para desplazar el terreno, rotar y poder acercar o alejar.	
Tipo	Funcional
Tipo de usuario asociado	Programador

RS0005: Módulo de carga de datos y visualización.	
Descripción: El software tiene que tener la opción de cargar los desde un archivo tipo runnel, y su posterior visualización en OpenGL.	
Tipo	Funcional
Tipo de usuario asociado	Programador

RS0006: Modulo para guardar los terrenos.	
Descripción: El software tiene que tener la opción de guardar los puntos y estructuras del terreno que el usuario quiera guardar en un archivo tipo runnel.	
Tipo	Funcional
Tipo de usuario asociado	Programador

RS0007: Modulo de identificación de cauces en el terreno.	
Descripción: El software tiene que identificar los cauces que presenta el terreno.	
Tipo	Funcional
Tipo de usuario asociado	Programador

RS0008: Modulo de visualización de cauces en el terreno.	
Descripción: El software una vez que identifico los cauces en el terreno tiene que ser capaz de visualizarlo.	
Tipo	Funcional
Tipo de usuario asociado	Programador

RS0009: Modulo de identificación de patrones de drenaje.	
Descripción: El software tiene que identificar los patrones de drenaje que presente las redes de drenaje que fueron encontradas y estos tienen que ser visualizados.	
Tipo	Funcional
Tipo de usuario asociado	Programador

RS0010: Información sobre patrones de drenaje.	
Descripción: El software tiene que tener un widget donde muestre la información de los patrones de drenaje que identifica.	
Tipo	Funcional
Tipo de usuario asociado	Programador

RS0011: Interfaz amigable.	
Descripción: El software tiene que tener una interfaz amigable e interactiva, para que el usuario pueda decidir qué tipo de identificación de cauces desea aplicar al terreno. También, que tipo de identificación de patrón de drenaje quiere aplicar.	
Tipo	Calidad
Tipo de usuario asociado	Programador

RS0012: Software OpenSource	
Descripción: El software tiene que ser de código libre, donde alojara en un repositorio en Github.	
Tipo	Restricción
Tipo de usuario asociado	Programador

RS0013: Software Multiplataforma	
Descripción: El software tiene que poder ser ejecutado en sistemas operativos Windows y Ubuntu.	
Tipo	Restricción
Tipo de usuario asociado	Programador

3.5 Matriz de trazado requisitos de usuario vs requisitos de software

	RS0001	RS0002	RS0003	RS0004	RS0005	RS0006	RS0007	RS0008	RS0009	RS0010	RS0011	RS0012	RS0013
RU0001	X	X											
RU0002			X	X									
RU0003					X								
RU0004						X							
RU0005		X					X	X					
RU0006		X							X				
RU0007										X			
RU0008				X									
RU0009											X		
RU0010												X	
RU0011													X

3.4 Diagrama UML para los casos de uso

Los diagramas UML más importantes que presenta el sistema son los siguientes:

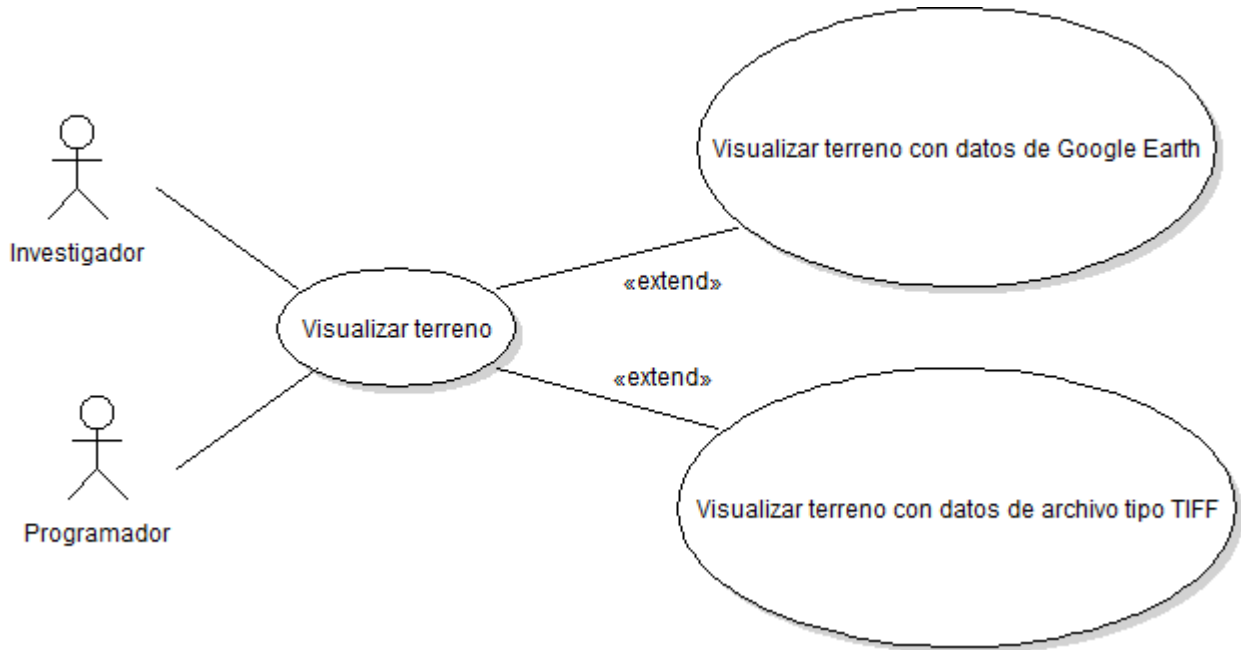


Diagrama 1: Visualización del terreno

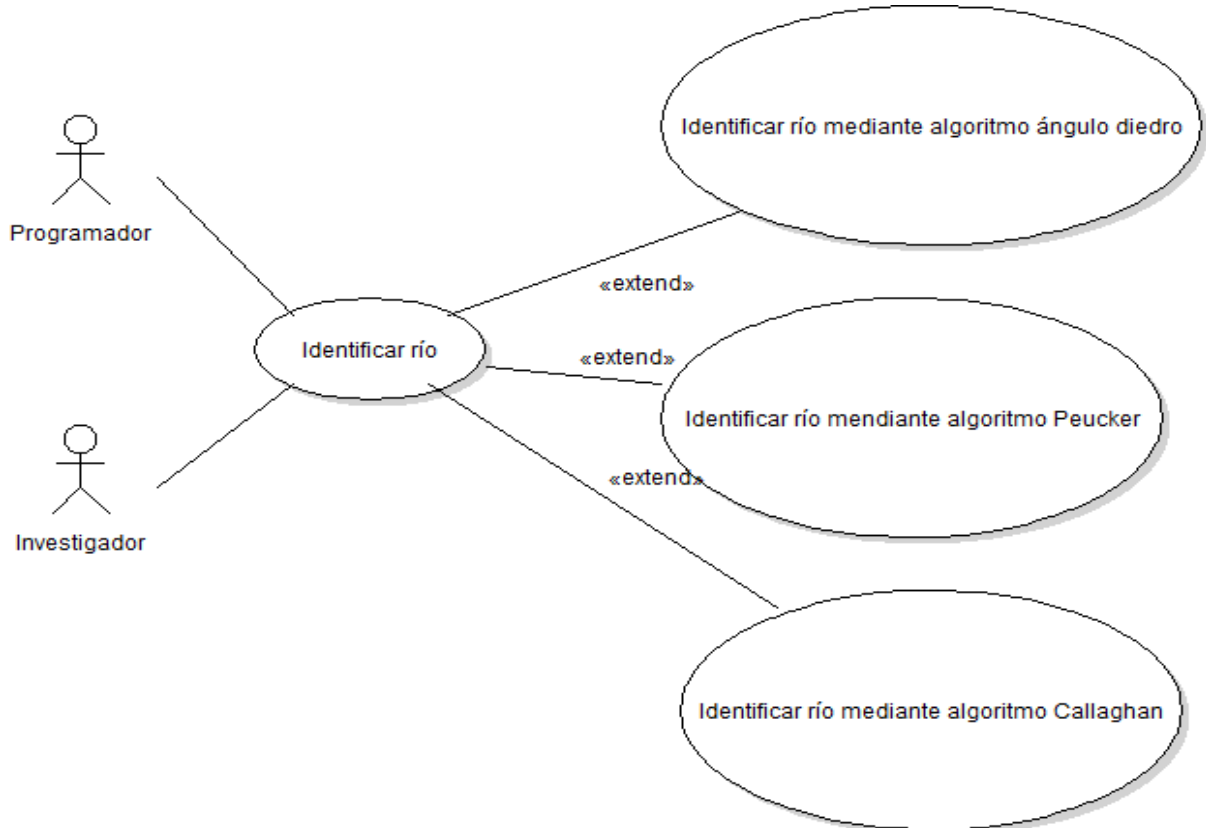


Diagrama 2: Identificación de río

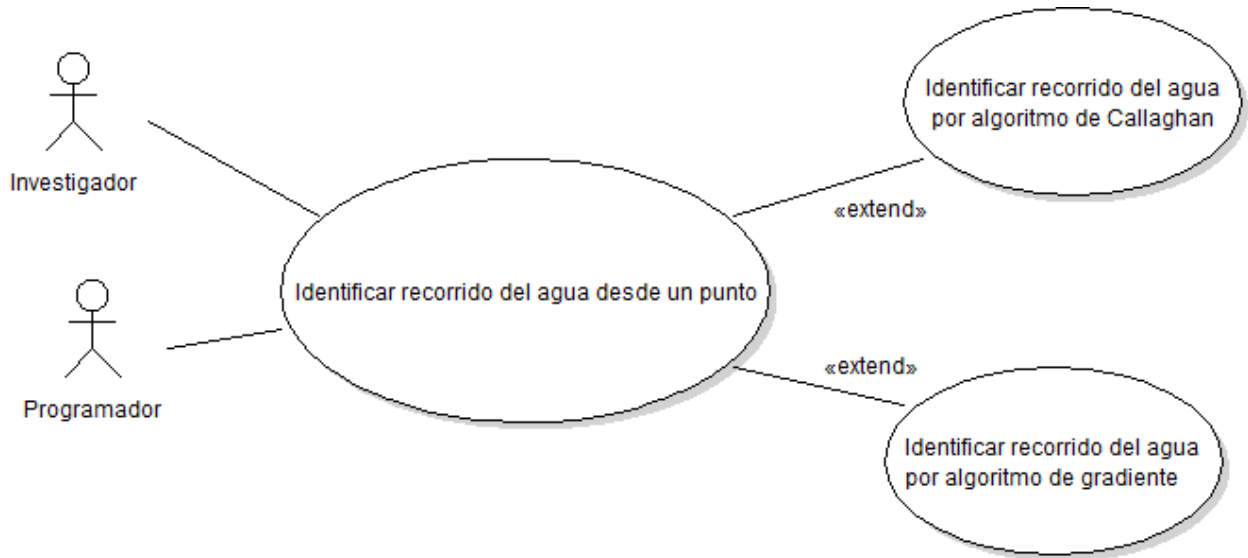


Diagrama 3: Identificación de recorrido del agua en un determinado punto

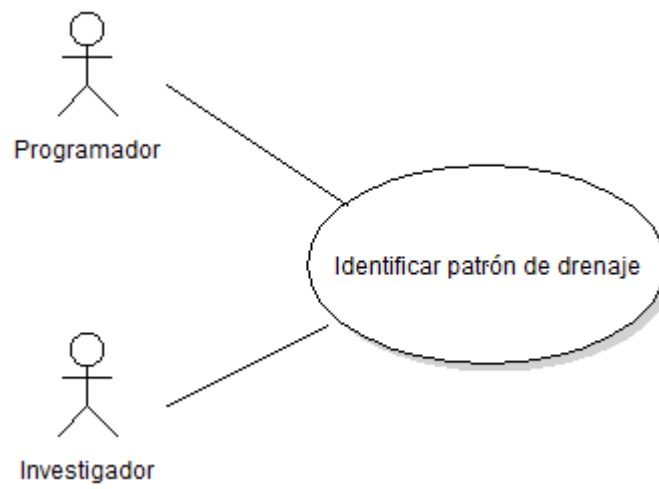


Diagrama 4: Identificación de patrón de drenaje

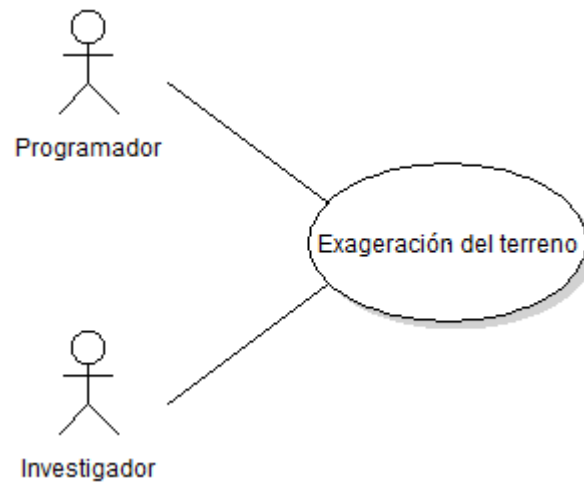


Diagrama 5: Exageración del terreno

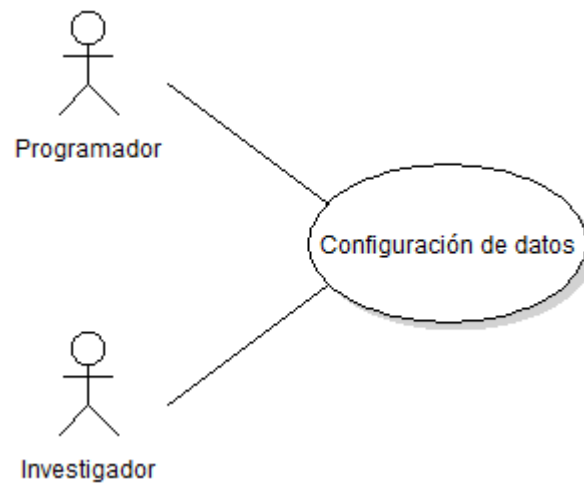


Diagrama 6: Configuración de datos

3.5 Casos de uso de alto nivel

A continuación se presentan los casos de usos de alto nivel que representan las acciones que realiza el actor dentro del sistema:

Caso de uso	Visualizar terreno con los datos Google Earth
Actores	Investigador, Programador
Tipo	Primario
Resumen	Este caso de uso comienza cuando el usuario decide visualizar el terreno con datos provenientes de Google Earth. Para ello, el usuario tiene que elegir la cantidad de puntos que desea que el terreno tenga y presionar el botón para obtener los datos de forma uniforme. Posterior, a la selección de las opciones el usuario puede visualizar el terreno en la pestaña de malla.

Caso de uso	Visualizar terreno con los datos de archivo tipo TIFF
Actores	Investigador, Programador
Tipo	Primario
Resumen	Cuando el usuario decide visualizar el terreno con datos provenientes de un archivo tipo TIFF. Tiene que optar por la opción TIFF en Open File del menú File del software. Una vez, seleccionado el archivo, se visualiza el terreno en la pestaña de malla.

Caso de uso	Identificación de río mediante ángulo diedro
Actores	Investigador, Programador
Tipo	Primario
Precondiciones	El usuario debe haber seleccionado una fuente de datos a través del caso de uso de visualización del terreno
Resumen	El usuario para la identificación de río mediante el ángulo diedro, tiene que seleccionar la opción de algoritmo por ángulo diedro ³ en la búsqueda de ríos y luego run. El usuario tiene la opción de configurar el ángulo permitido para que se coloree las aristas que tienen el ángulo permitido.
Postcondiciones	Si el usuario desea cambiar el ángulo permitido, tiene que ingresar el valor y luego presionar change.

³ Dícese del ángulo formado por dos semiplanos que comparten una arista en común.

Caso de uso	Identificación de río mediante algoritmo Peucker
Actores	Investigador, Programador
Tipo	Primario
Precondiciones	El usuario debe haber seleccionado una fuente de datos a través del caso de uso de visualización del terreno
Resumen	El usuario para la identificación de río mediante el algoritmo de Peucker, tiene que seleccionar la opción de búsqueda de ríos: algoritmo Peucker y luego run. En este caso no hay valores de configuración por el usuario.

Caso de uso	Identificación de río mediante algoritmo Callaghan
Actores	Investigador, Programador
Tipo	Primario
Precondiciones	El usuario debe haber seleccionado una fuente de datos a través del caso de uso de visualización del terreno.
Resumen	El usuario para la identificación de río mediante el algoritmo de Callaghan, tiene que seleccionar la opción de búsqueda de ríos: algoritmo Callaghan y luego run. Para este caso, el valor de configuración del algoritmo es el porcentaje de cauces que se mostrarán, siendo estos los que contienen la mayor cantidad de agua.
Postcondiciones	Si el usuario desea cambiar el porcentaje de cauces que se mostrarán, tiene que ingresar el valor y presionar change.

Caso de uso	Identificación de recorrido del agua por Callaghan.
Actores	Investigador, Programador
Tipo	Primario
Precondiciones	El usuario debe haber seleccionado una fuente de datos a través del caso de uso de visualización del terreno
Resumen	El usuario selecciona el algoritmo de Callaghan en recorrido del agua, y luego presiona run. Para este caso, la variable de configuración es la altura que puede subir el agua, donde en la altura máxima de subida, se ingresa el valor correspondiente y se presiona change.
Postcondiciones	Si el usuario desea cambiar la altura máxima que el agua puede subir, tiene que ingresar el valor y luego presionar change.

Caso de uso	Identificación de recorrido del agua por gradiente
Actores	Investigador, Programador
Tipo	Primario
Precondiciones	El usuario debe haber seleccionado una fuente de datos a través del caso de uso de visualización del terreno
Resumen	El usuario selecciona el algoritmo de gradiente en recorrido del agua, y luego presiona run. Para este caso, no hay variable de configuración.

Caso de uso	Identificación de patrón de drenaje
Actores	Investigador, Programador
Tipo	Primario
Precondiciones	El usuario debe haber seleccionado una fuente de datos a través del caso de uso de visualización del terreno
Resumen	El usuario selecciona la opción ZhangGuilbert en identificación de patrón de drenaje y presiona run. El resultado será visualizado en la malla.

Caso de uso	Exageración de terreno
Actores	Investigador, Programador
Tipo	Primario
Precondiciones	El usuario debe haber seleccionado una fuente de datos a través del caso de uso de visualización del terreno
Resumen	Si el usuario desea cambiar la exageración de la altura del terreno, tiene que ingresar el valor de exageración que desea en la opción de exageración del terreno y presionar change.
Postcondiciones	Si el usuario decide volver a cambiar la exageración, tiene que cambiar el valor y presionar change.

Caso de uso	Configuración de datos
Actores	Investigador, Programador
Tipo	Primario
Resumen	El usuario tiene que seleccionar la pestaña configuration. En esta pestaña están todos los colores correspondientes a las funcionalidades del software y estos son modificados presionando change en el color que desea cambiar.
Postcondiciones	Si el usuario decide volver a cambiar el color, tiene que presionar change y seleccionar el color que desee.

Capítulo 4: Diseño e implementación de la solución

A continuación se presenta las estructuras que fueron utilizadas para el desarrollo de la solución y los procedimientos que se llevan a cabo para la implementación

4.1 Estructura del Terreno

La estructura del terreno, se basa en la construcción de primitivas que permitan realizar los procesamientos necesarios. Las estructuras de datos creadas son:

Point: la clase Point contiene los datos correspondientes al punto, estos son:

- Coord: coordenadas en el espacio x, y, z, correspondiente al lugar en el planeta.
- Flags: corresponde a marcas que se necesite para el procesamiento de algoritmos, principalmente en la detección de ríos.
- Ident: identificador del punto.

Edge: la estructura Edge corresponde a las aristas de un polígono, en este caso, triángulos. Los datos que almacena la estructura son:

- Datos correspondiente a los valores como vector
- Punteros a los triángulos vecinos
- El ángulo diedro que forma con los triángulos vecinos
- Los ids de los puntos que lo conforman

Triangle: la estructura triangle corresponde a los triángulos que pertenecen a la malla de superficie del terreno. Los datos que almacena la estructura son:

- Puntero de los puntos que lo conforman
- Puntero de las aristas que lo conforman
- El incentro del triángulo
- Normal del triángulo
- Gradiente del triángulo
- Identificador del triángulo

Terrain: esta estructura contiene lo siguiente:

- Los puntos que lo conforman
- Las aristas que lo conforman
- Los triángulos que lo conforman
- Los mínimo y máximos valores de las coordenadas de los puntos

4.2 Obtención de datos

La obtención de datos dependerá de donde proviene la fuente. Esta fuente puede ser a través de un archivo TIFF o de Google Earth. A continuación se explica que se realiza en cada caso.

TIFF: Para la obtención de los datos contenidos en un archivo tipo TIFF, se procedió a obtener el ancho y la altura, y para cada punto la altura se ajustó con un escalar, ya que la obtenida en el archivo corresponde al rango 0 a 255, que es el producto de la obtención de un color, el cual está asociado a la altura de ese punto. Cabe destacar, que no se obtiene todo los valores que contiene un TIFF, si no, número equi-espaciados, ya que estos almacenan demasiados puntos. Posterior, a la obtención se procede a la formación de la estructura para el terreno que es similar para cada diferente tipo de obtención de datos.

Google Earth: Para la obtención de los datos a través de Google Earth, se procedió a crear una página web, la cual contenía en su interior el mapa. Por medio de la selección de un lugar a través del posicionamiento, se va consultado por medio de la API la altura dependiendo de la latitud y longitud dada, dividida en grilla, en cada punto. Cabe destacar, que la latitud y longitud no es una medida de comparación entre sí, por lo tanto se tiene que realizar una conversión a valores UTM, los cuales están representados en metros.

4.3 Construcción de la estructura

Para la construcción de la estructura se procede a realizar el mismo procedimiento independiente de la fuente de datos. Se enumera a continuación la construcción.

1. Envío de los datos a Qhull junto con los valores de configuración
2. Recepción de los datos de Qhull y creación del terreno

Para él envío de los datos a Qhull se procede a crear el formato para que reciba los datos Qhull, donde los puntos que se envían sólo corresponden al x e y, junto con la configuración, que en este caso es la construcción de una triangulación de Delaunay. Una vez, que ha salido correcto el procesamiento en Qhull, se crea la estructura agregando la altura a cada punto.

4.4 Algoritmos para la detección de ríos

Para la detección de ríos en la malla de polígonos del terreno, se procedió a implementar 3 algoritmos que reconocieran los cauces dentro del terreno que se estaba analizando. Estos algoritmos fueron:

- Algoritmo por ángulo diedro
- Algoritmo de Peucker [24]
- Algoritmo de Callaghan-Mark [24]

4.4.1 Algoritmo por ángulo diedro

Para la explicación del algoritmo por ángulo diedro, se comenzará con la definición de ángulo diedro.

Ángulo diedro: Es el ángulo determinado por la intercepción de 2 planos y se calcula por medio de las normales de cada plano. La siguiente fórmula entrega el cálculo del ángulo diedro φ_{AB} , para 2 planos A y B, donde n_A es la normal del plano A y n_B la normal del plano B:

$$\varphi_{AB} = \arccos(n_A * n_B)$$

Para la obtención de los ríos, se pensó que los ángulos agudos dentro de un terreno corresponderían a cauces. A partir de este pensamiento, se procedió a obtener los ángulos diedros formados por los triángulos. Donde, el ángulo es un valor perteneciente a la arista y es formado a partir de los triángulos que la comparten. Una vez obtenidos todos los ángulos diedros de cada arista, se procede a ajustar parametros configurables cómo un ángulo mínimo y máximo, con el fin de observar el resultado desde un ángulo límite.

4.4.2 Algoritmo de Peucker

El algoritmo de Peucker es un algoritmo que trabaja el terreno como una grilla, considerando sólo los puntos. Este algoritmo considera una ventana cuadrada de 2x2, y para cada ventana marca el punto que contiene la altura mayor, y continua con los siguientes 4 puntos correspondientes al siguiente cuadrado. Una vez, que ha terminado, los puntos que no se han marcado son los pertenecientes a una red de drenaje [23]. Cabe destacar, que este tipo de algoritmo es de orden $\theta(n)$ y puede producir puntos que están aislados como cauces interrumpidos.

A continuación se presenta un ejemplo del algoritmo:

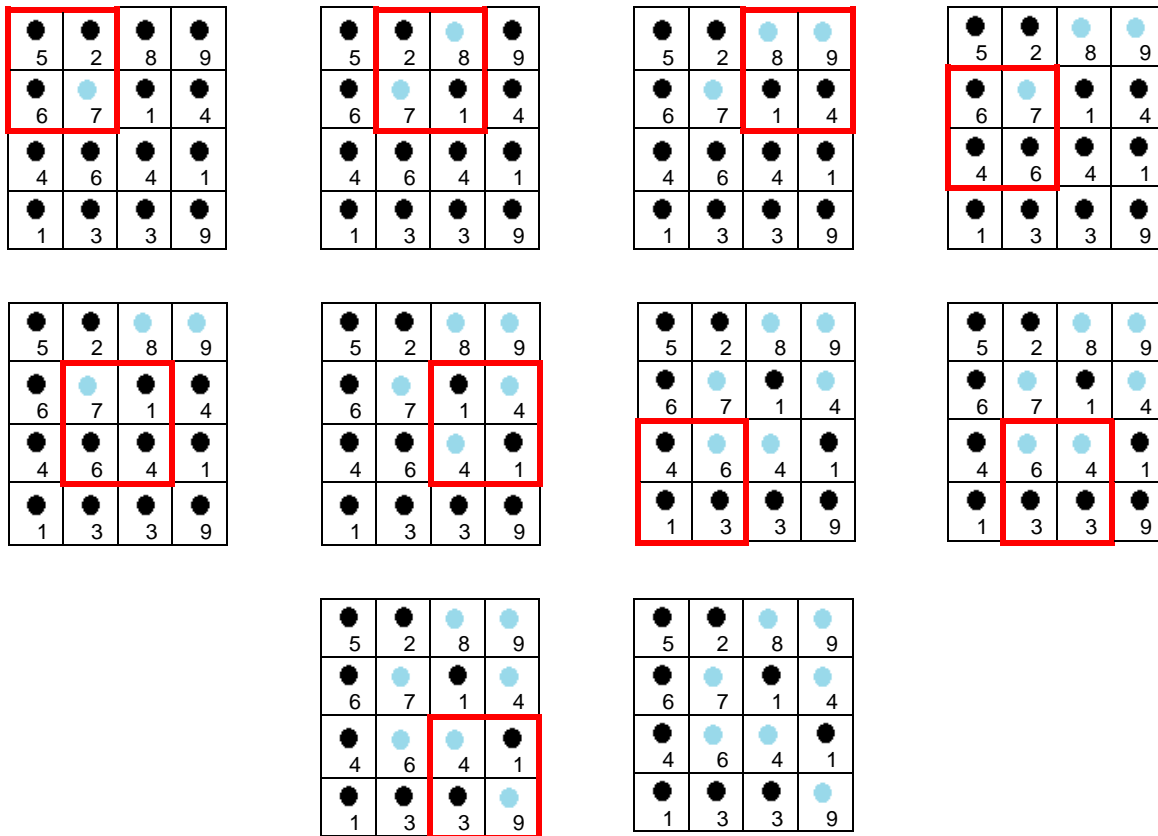


Diagrama 7: Ejemplo del algoritmo de Peucker

Se observa del diagrama que cuando en un mismo cuadrado hay 2 o más valores que tienen la mayor altura, todos estos son marcados. Finalmente, cuando se aplica el algoritmo, los puntos que no han sido marcados son los correspondientes a los cauces.

4.4.3 Algoritmo de Callaghan-Mark

El algoritmo de Callaghan-Mark, es un algoritmo de grilla, el cual considera la acumulación de agua que tiene cada punto. Para ello, ordena cada punto desde el punto de mayor altura hasta el de menor, cada punto tiene una cantidad de agua base como 1. Una vez, que estén ordenados todos los puntos, se escoge el primer punto y se define a sus vecinos como los 8 puntos que lo rodean, el punto que tenga la mayor diferencia y sea de menor altura, es el punto al cual se le sumará la cantidad de agua del punto al cual se está realizando el algoritmo. Terminado el punto, se sigue con el siguiente punto dentro de la lista ordenada. El orden de este algoritmo corresponde a $\theta(n + n \log(n))$, que es n por recorrer y $n \log(n)$ por ordenar.

Para este algoritmo, el usuario también puede definir el porcentaje de cauces que desea considerar, siendo estos los que contienen la mayor cantidad de agua. Los cauces serán dibujados o no dependiendo de si están por encima o debajo de este valor [23]

4.5 Algoritmo para la identificación de patrones de drenaje

Para la identificación de patrones de drenaje se pensaba la utilización de búsqueda por similitud de polígonos, pero esta no se realizó, ya que no había certeza de que tan precisa podría ser en la clasificación del patrón, y era una aproximación experimental que podía requerir mucho tiempo de implementación sin la seguridad de obtener los resultados deseados.

Posterior a eso, a través del paper [24] se encontró un algoritmo que provee la clasificación de 4 tipos de patrones de drenaje, los cuales son individuales, ya que estos no dependen de una organización espacial como otros patrones de drenaje. Estos tipos de patrones de drenaje son los siguientes:

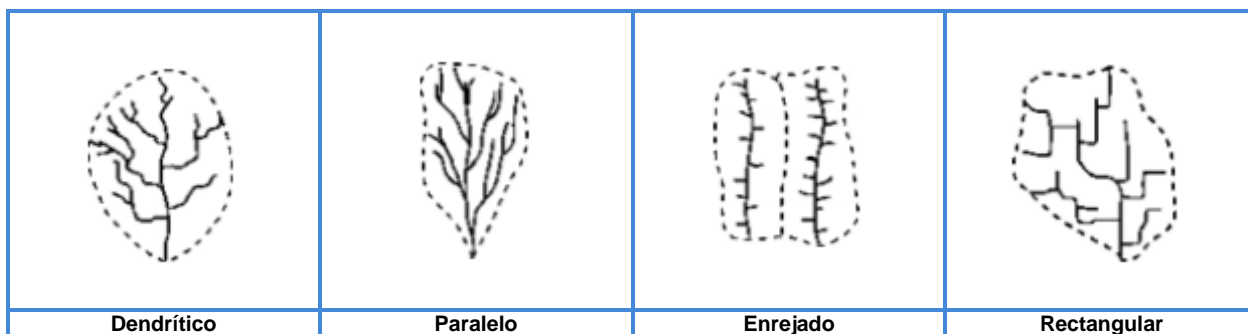


Imagen 10: Patrones de drenaje

La identificación de estos patrones se debe a sus indicadores geométricos que se explicarán más adelante. Con estos indicadores geométricos definidos es posible el reconocimiento de patrones dendríticos, paralelo, enrejado y rectangular, donde el indicador más importante será el ángulo formado por la fluencia de la corriente principal con las demás fluencias.

4.5.1 Indicadores geométricos

A continuación se presenta 4 indicadores geométricos que permiten identificar los 4 patrones de drenaje.

Ángulo de unión (Junction angle)

Es el ángulo promedio formado por la afluente principal con las afluentes de una red de drenaje. Para la obtención hay que identificar la afluencia principal, que es la de mayor

orden según la clasificación de Strahler-Horton. En caso, de haber más de una, se sigue 2 reglas para escoger el afluente principal.

- Se escoge el que tiene la misma dirección que la parte baja del río.
- Se escoge el que tiene la mayor longitud, si varios afluentes tienen una dirección similar a la parte baja del río.

A continuación se presenta un ejemplo para realizar los cálculos.

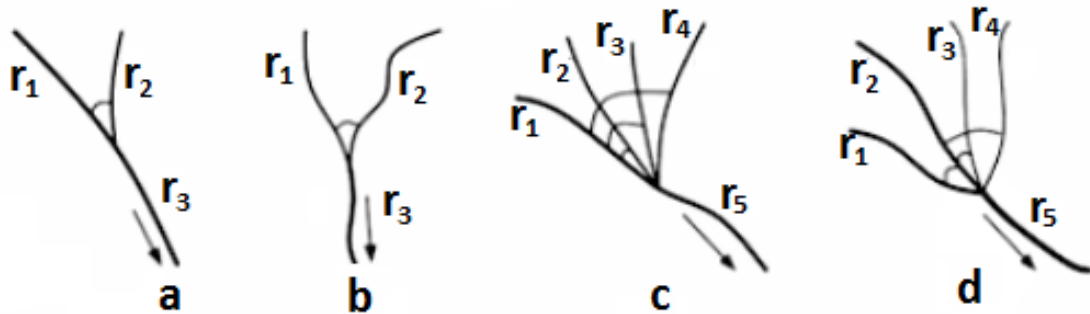


Imagen 11: Casos de cálculo

Para el primer caso (a) se tiene 2 afluentes, donde r_1 es el afluente de mayor orden con respecto a r_2 , por lo tanto el afluente principal es r_1 . Para el caso en (b), son 2 afluentes del mismo orden, pero para este caso el afluente r_2 , mantiene la misma dirección que r_3 . Para estos 2 casos anteriores el calcula entre r_1 y r_2 . En el caso (c), todos son distintos y hay más de 2 afluentes uniéndose, por lo que es necesario identificar el afluente principal, en ese caso, resulta ser r_1 que es el de mayor orden, mientras que en el caso (d), existe 2 afluentes que tienen el mismo orden r_1 y r_2 , en este caso se elige r_2 por ser el afluente con la misma dirección que la parte baja de río y ser el más largo. Para estos 2 últimos casos, se obtiene el promedio de los ángulos obtenidos.

La fórmula para realizar el cálculo del ángulo viene dado por ley de los cosenos:

$$\angle P_2 P_1 P_3 = \arccos\left(\frac{a^2 + b^2 - c^2}{2ab}\right)$$

Donde a es el segmento de P_1 a P_2 , b es el segmento de P_1 a P_3 y c es el segmento de P_2 a P_3 , de la siguiente imagen.

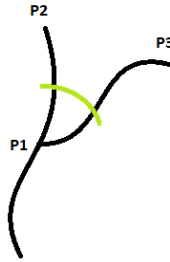


Imagen 12: Ángulo de unión

Sinuosidad (Sinuosity)

Es establecida como la relación entre la longitud de los tramos del afluente principal con respecto a la distancia entre el inicio y fin del afluente, con el fin de cuantificar la sinuosidad. Lo anterior es similar a la suma de los tramos del canal (afluente principal) divididos con respecto a la distancia del primer punto con respecto al último. Esto viene dado por la siguiente fórmula:

$$SI = \frac{\sum_{i=1}^{N-1} Dis(P_i, P_{i+1})}{Dis(P_1, P_n)}$$

Donde $Dis()$, es la distancia entre dos puntos.

Relación de longitud (Length ratio)

Este indicador geométrico es utilizado para identificar un patrón enrejado con respecto a uno paralelo, donde la medición que realiza es el cálculo de la longitud de los afluentes con respecto a la corriente principal. La corriente principal se considera como los segmentos conectados con el mismo orden. Mientras que el cálculo de los afluentes es el promedio de la suma de los afluentes.

Elongación de la cuenca (Catchment elongation)

Para calcular este indicador geométrico, se tiene que obtener el Minimum Bounding Rectangle (MBR), Axis Aligned Bounding Box (AABB) o la cerradura convexa. Donde:

- **MBR:** es el mínimo rectángulo que contiene todos los puntos de la red de drenaje.
- **AABB:** es el mínimo rectángulo que contiene todos los puntos de la red de drenaje y está alineado con los ejes.
- **Cerradura convexa:** es el conjunto de puntos que forma un polígono convexo, manteniendo a todos los puntos dentro de esta estructura.

Posterior al cálculo de alguno de estos 3 tipos de envolturas convexas. Se considera el lado de la envoltura que esté más alineado con el afluente principal y se calcula la proporción como el lado más alineado con el lado menos alineado.

A continuación se muestra las distintas cerraduras convexas:

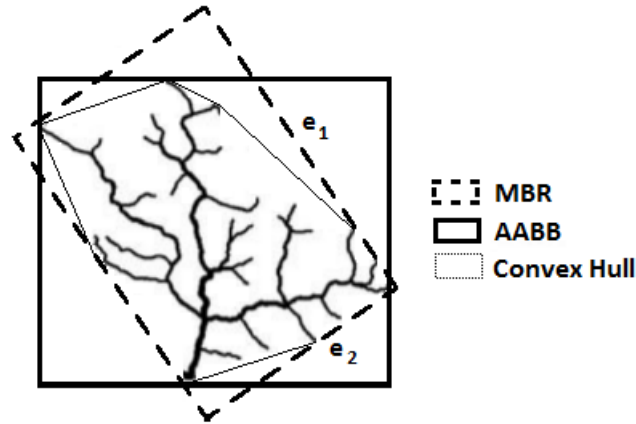


Imagen 13: Envoltura convexa

Donde la formula viene dada por:

$$ratio = \frac{e2}{e1}$$

En el caso del software se eligió el MBR por simpleza y porque posee un mejor ajuste que el AABB.

4.5.2 Identificación del patrón de drenaje

En la siguiente tabla, se muestra como se clasifican según los indicadores geométricos propuestos por [24]:

Patrón de drenaje	Promedio del ángulo de unión (α)	Sinuosidad (β)	Relación de longitud (γ)	Elongación de la cuenca (δ)
Dendrítico	Agudo, $\alpha < 90^\circ$	-	-	Ancho, $\delta < 1$ or $\delta \approx 1$
Paralelo	Muy agudo, $\alpha \ll 90^\circ$	No curvado, $\beta \rightarrow 1$	Largo, $\gamma \approx 1$ or $\gamma > 1$	Alargado, $\delta \gg 1$
Enrejado	Ángulo recto, $\alpha \approx 90^\circ$	No curvado, $\beta \rightarrow 1$	Corto, $\gamma \ll 1$	Alargado, $\delta \gg 1$
Rectangular	Ángulo recto, $\alpha \approx 90^\circ$	Curvado, $\beta > 1$	-	-

Tabla 1: Identificación de patrón de drenaje

Cabe destacar, que según estos criterios se realizó la identificación de los patrones.

4.6 Recorrido del agua desde un determinado lugar

Para el recorrido del agua desde un determinado lugar, se procedió a utilizar 2 algoritmos, el primero fue descrito en la obtención de ríos y el segundo será explicado más adelante:

- Algoritmo Callaghan
- Gradiente

Para cada uno de estos algoritmos se realizó la opción de ver el camino desde un punto que decide el usuario. El punto que es escogido por el usuario es convertido de punto de la pantalla al plano del terreno, donde sólo se compara los valores de x e y.

Cabe destacar que para ambos algoritmos se configuro un parámetro de configuración que permite tener un delta de subida para el agua, ya que en la realidad es posible que si un punto está a 1mm más alto que otro, el agua puede transcurrir por ese punto y no detenerse en el anterior.

4.6.1 Gradiente

Para obtener la gradiente fue necesario el cálculo del vector de máxima pendiente de un triángulo, vector por el cual se producirá el descenso de agua, calculamos la proyección del vector $-\hat{Z}$ sobre el plano del triángulo. Para calcular la proyección, escribimos \hat{Z} como la suma de dos vectores:

$$\vec{V}_A + \vec{V}_B = -\hat{Z}$$

Donde el vector \vec{V}_A corresponde a la componente paralela al plano definido por el triángulo y \vec{V}_B la componente perpendicular a este.

En este caso, el vector de máxima pendiente del triángulo es exactamente la componente \vec{V}_A del vector $-\hat{Z}$. Luego, la ecuación puede ser reescrita de la siguiente forma:

$$\vec{V}_A = -\hat{Z} - \vec{V}_B$$

Donde $-\hat{Z} = (0, 0, -1)$ y el vector \vec{V}_B puede ser calculado como el producto punto entre la normal \vec{N} del triángulo y el vector $-\hat{Z}$ multiplicado por la normal. De esta forma se obtiene la componente de $-\hat{Z}$ perpendicular al plano que va en la misma dirección que \vec{N} :

$$\vec{V}_B = (\vec{N} \cdot -\hat{Z}) * \vec{N}$$

Hay que tener en consideración que si el triángulo está alineado con el plano XY, el vector \vec{V}_B tendrá módulo 0, ya que no existe “máxima pendiente” para ese caso.

Finalmente, se puede normalizar \vec{V}_B o utilizar su módulo para cuantificar que tan pronunciada es la pendiente. Si el módulo es más cercano a 0, el triángulo estará más horizontal, si el módulo es mayor, el triángulo tendrá una diferencia de altura mayor entre sus vértices (estará más inclinado).

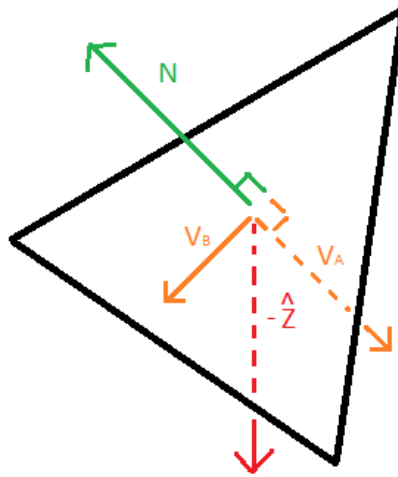


Imagen 14: Obtención de la gradiente

4.7 Funcionalidades de ayuda al programador

Las funcionalidades de ayuda al programador, fueron la visualización de las siguientes opciones:

- Vista de las normales de los triángulos
- Vista de los vectores de la gradiente del triángulo

Estas funcionalidades se realizaron con la intención de ver que lo desarrollado estaba bien encaminado y por posibles errores que se desarrollen.

4.8 Archivo de datos propietario

En la solución realizada la estructura del archivo que almacena las propiedades del terreno que el usuario desea guardar es la siguiente:

```
numero_de_puntos
id      coord_x coord_y coord_z
...
triangulation
cantidad_triangulos
id_triangle      id_pto1 id_pto2      id_pto3
....
cantidad_vecinos
id_triangle1      id_triangle2      id_triangle3
....
algorithms_river name_algorithm
cantidad_de_datos
id_pto1      id_pto2
algorithms_red name_algorithm
cantidad_de_redes
red      1
type      number
id_pto1      id_pto2      orden_strahler_horton
....
red      2
type      number
id_pto1      id_pto2      orden_strahler_horton
....
```

Diagrama 8: Formato del archivo.

Se observa que la estructura que se guarda, mantiene los índices a cada punto y por medio de estos, se guardan los valores de las redes y otros resultados de los algoritmos aplicados. Cabe destacar, que para guardar los datos de una red, se guarda por cada arista, donde se señala el tipo de red que es y el orden Strahler-Horton que posee cada arista.

4.9 Arquitectura del Software

La arquitectura del software que se realiza para la necesidad que se aborda, está representada por el siguiente diagrama:

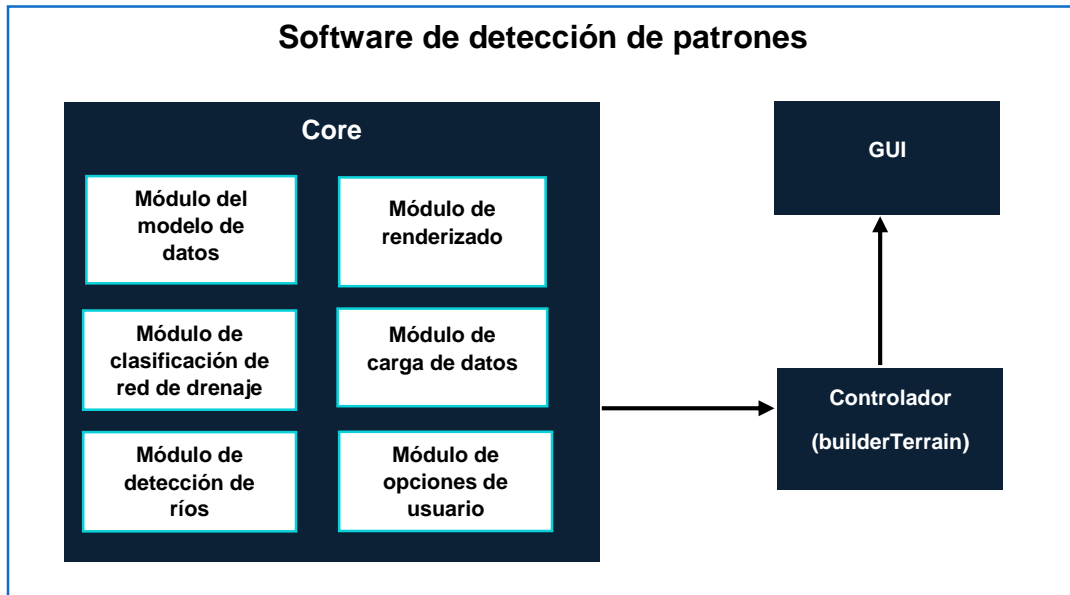


Diagrama 9: Arquitectura de la aplicación.

A continuación se explicará cada componente de la arquitectura, para ello se comenzará con la explicación del controlador y después de cada módulo.

4.9.1 Controlador

El controlador está encargado de realizar la comunicación de la GUI con el core de la aplicación. A su vez, se encarga de manejar los distintos módulos.

4.9.2 Módulo de detección de ríos

Este módulo es el encargado de contener los distintos algoritmos para el reconocimiento de ríos dentro del mapa.

A continuación se presenta el diagrama que muestra la composición del módulo.

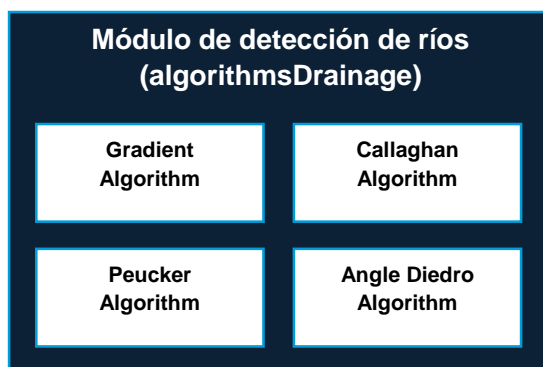


Diagrama 10: Módulo de detección de ríos.

Donde cada sub-módulo está encargado de aplicar el algoritmo correspondiente al terreno.

4.9.3 Módulo de clasificación de red de drenaje

Módulo encargado de contener los algoritmos para la identificación de patrón de drenaje en una red de drenaje. A continuación se presenta el diagrama que muestra la composición del módulo.

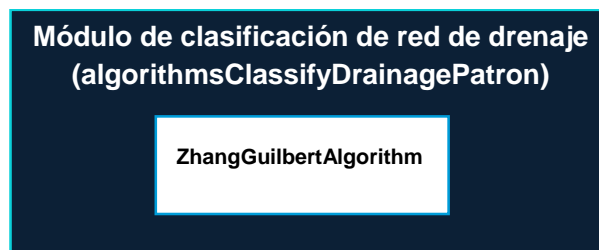


Diagrama 11: Módulo de clasificación de red de drenaje.

Donde el sub-módulo corresponde al algoritmo correspondiente a la clasificación de red de drenaje.

4.9.4 Módulo del modelo de datos

El módulo de módulo de datos es el encargado de contener las estructuras de datos necesarias para la creación del terreno.

A continuación se presenta el diagrama del módulo:

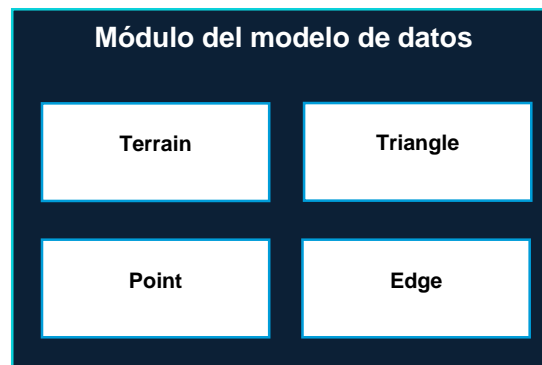


Diagrama 12: Módulo del modelo de datos.

4.9.5 Módulo de renderizado

Módulo enfocado al renderizado de la malla del terreno, ríos y geometrías necesarias. Se presenta a continuación el diagrama que contiene los sub-módulos:

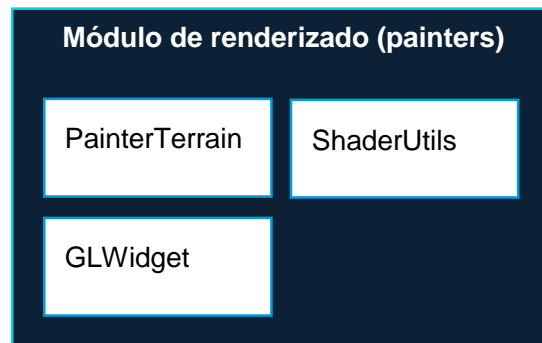


Diagrama 13: Módulo de renderizado.

Donde:

- **GLWidget**: clase encargada de la interfaz gráfica base para el dibujo del terreno
- **ShaderUtils**: clase encargada de la creación y funcionalidades de los shaders.
- **PainterTerrain**: clase que extiende de GLWidget y está encargada de manejar los datos obtenidos para dibujar el terreno, los procesa y crea los shaders necesarios para la renderización del terreno.

4.9.6 Módulo de carga de datos

Módulo que se encarga del procesamiento de los puntos, para la obtención y carga de estos. Procesa los datos de entrada para ser guardados en la estructura utilizando el

modelo de datos de la aplicación. Este módulo maneja datos de entrada desde archivos y fuente web. A continuación se presenta el diagrama que lo define:

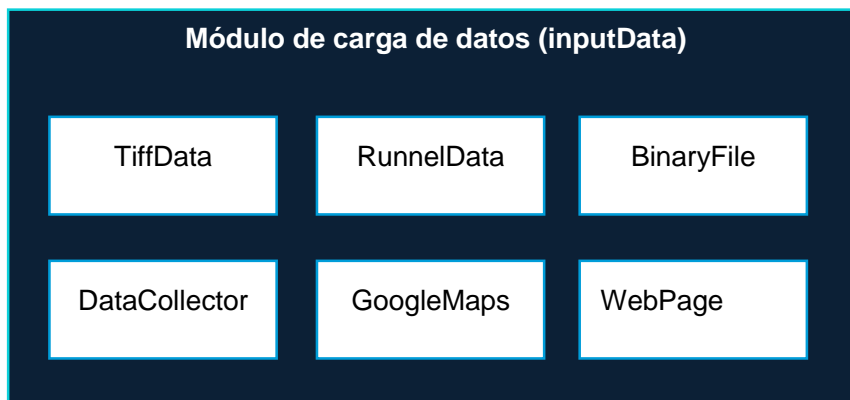


Diagrama 14: Módulo de carga de datos.

4.9.7 Módulo de opciones del usuario

Módulo encargado de los datos de configuración por parte del usuario. A continuación se presenta el esquema:

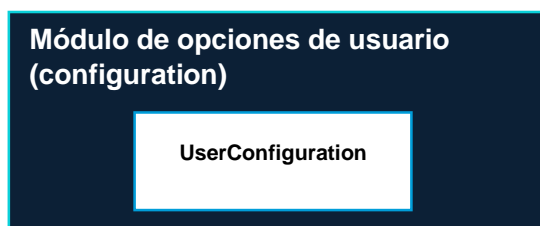


Diagrama 15: Módulo de opciones de usuario.

Donde el sub-módulo corresponde a las configuraciones que puede modificar el usuario en la interfaz.

4.10 Diagrama de Clases

A continuación se presenta el diagrama de clases de software diseñado:

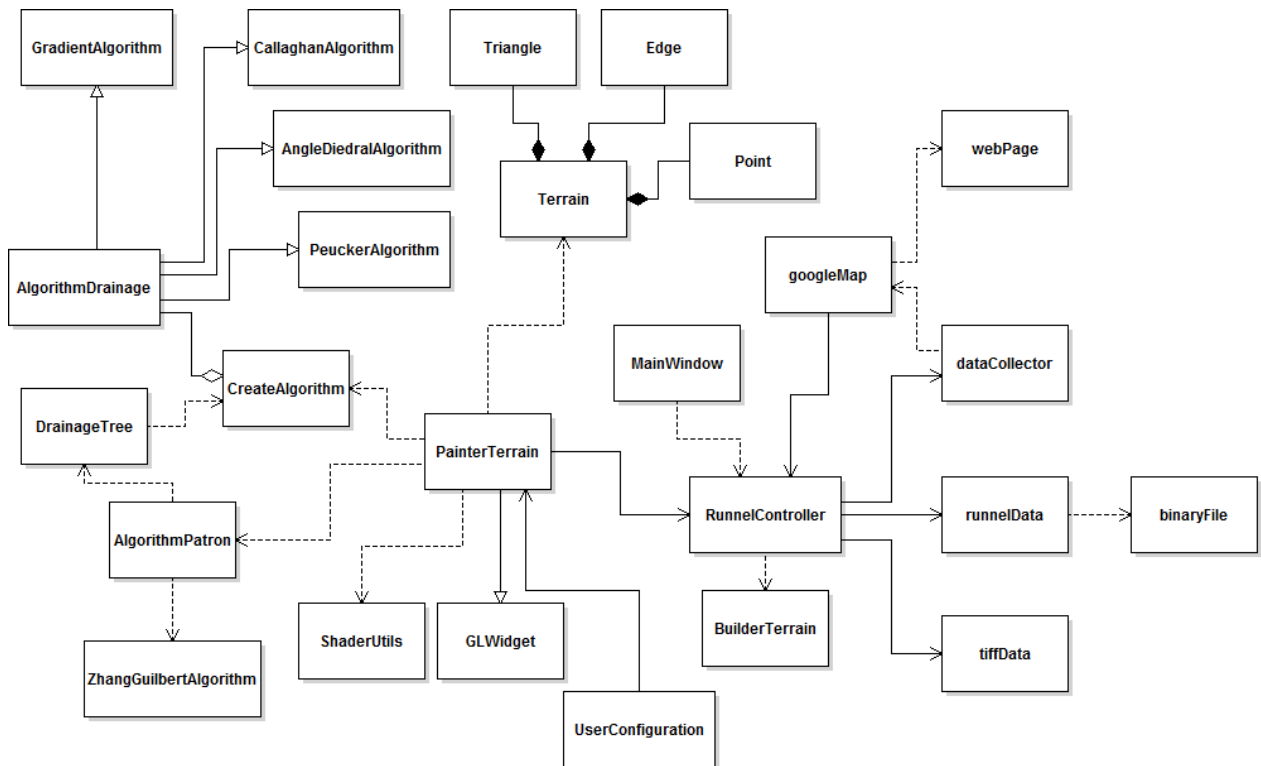


Diagrama 16: Diagrama de clases.

Del diagrama de clases se puede desprender que `runnelController`, es el encargado de manejar los input, procesar los datos y manejar la vista, pero no así del manejo de los algoritmos necesarios para el análisis. Siendo el encargado de interactuar con los algoritmos el render del terreno. Además, en el diagrama de clases se puede observar que se utilizaron algunos patrones de diseño con el fin de facilitar la extensión y modificaciones a futuro. A continuación se presenta los patrones de diseño que se utilizaron y el porqué.

4.10.1 Singleton

Para permitir la instanciación de una sola vez del mapa Google Earth, junto con la creación del mapa en sí, es que se decidió el uso de un Singleton en la clase `GoogleMap`, permitiendo así tener visibilidad desde cualquier lugar del software. También, se utilizó este patrón para la creación de los recursos necesarios para el render del terreno, siendo utilizado en la clase `PainterTerrain`.

4.10.3 Strategy

Para ser utilizado en la ejecución y configuración de diferentes algoritmos utilizados para el análisis de la red de drenaje. Con este patrón, se logra facilitar la generación, ejecución y configuración de algoritmos que se deseen agregar posteriormente. La clase que está encargada de la ejecución del patrón es *CreateAlgorithm* y es ésta la que hay que modificar cuando se desee agregar un nuevo algoritmo, mientras que *AlgorithmDrainage* es la clase que tiene que entenderse e implementar.

4.11 Funcionamiento del Software

En este apartado se presenta las principales funcionalidades desarrolladas en el software.

4.11.1 Opción de obtención de malla a través de Google Earth

A continuación, se presenta como obtener los datos desde Google Earth desde el software y qué opciones se encuentran.

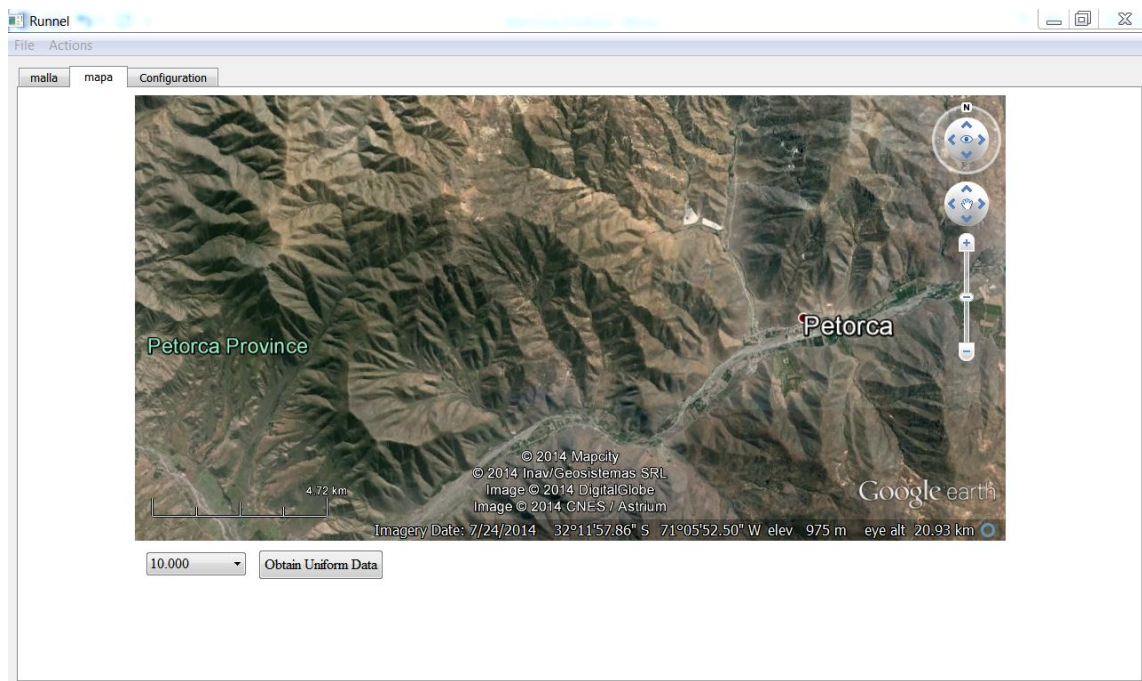


Imagen 16: Mapa de Google Earth.

Se escoge la cantidad de puntos y se hace click en la obtención de los datos de forma uniforme. Para posteriormente, observar el terreno en la pestaña de malla.

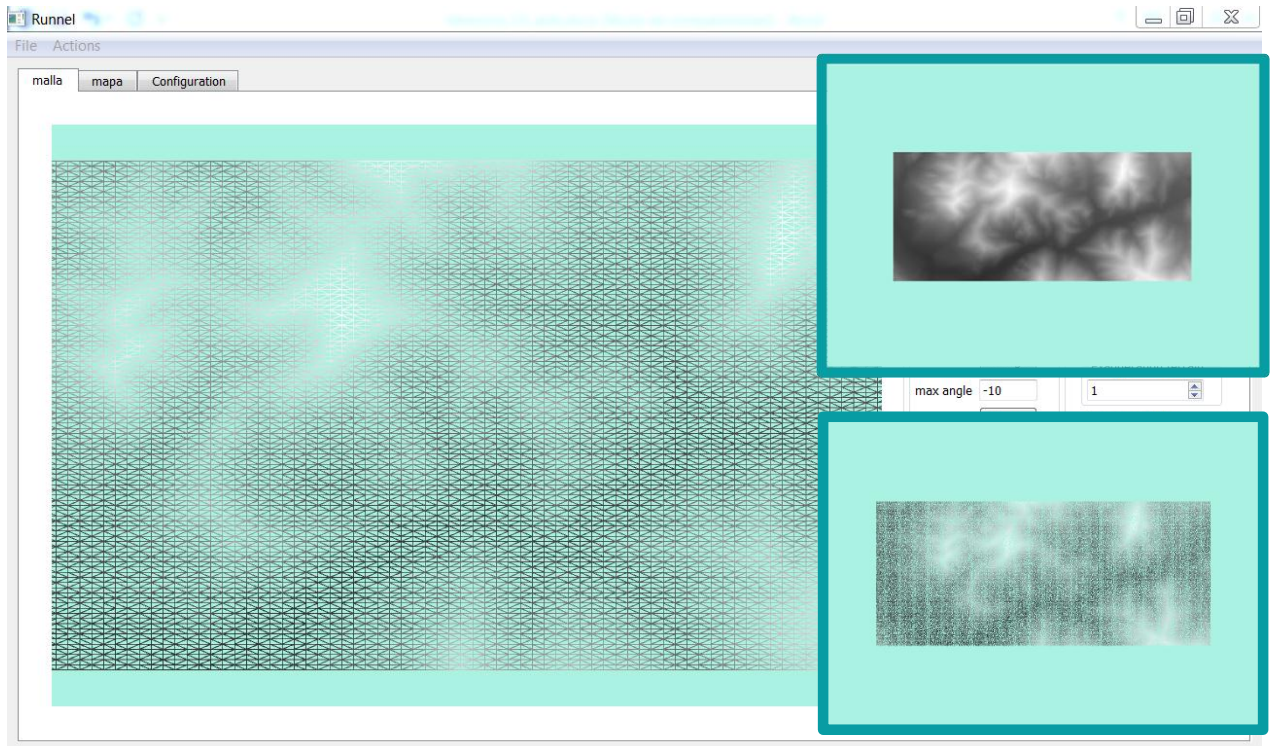


Imagen 17: Terreno en diferentes formas de visualización.

Una vez, obtenida la triangulación se puede visualizar como malla de alambre, de puntos y polígono, donde su visualización como polígonos permite identificar de forma más visible el terreno. Cabe destacar, que los sectores en blanco corresponden a las cimas, mientras que las oscuras a los valles.

4.11.2 Opción de obtención de malla a través de archivo tipo TIFF

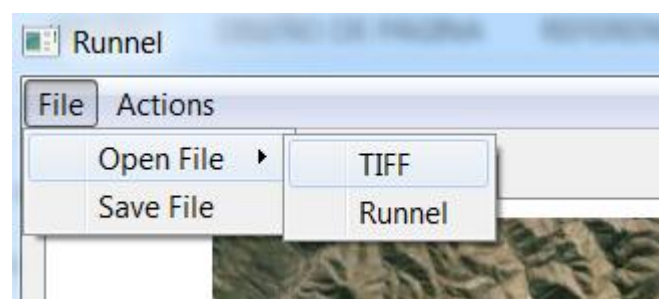


Imagen 18: Opción de archivo tipo TIFF.

Una vez abierto el archivo tipo TIFF, se tiene las mismas características que con los datos provenientes de Google Earth, sólo que con una mayor precisión

4.11.3 Opción de identificar ríos

Por medio, de los algoritmos de Callaghan, Peucker y ángulo diedro es posible visualizar las cuencas. A continuación se muestra ejemplos.

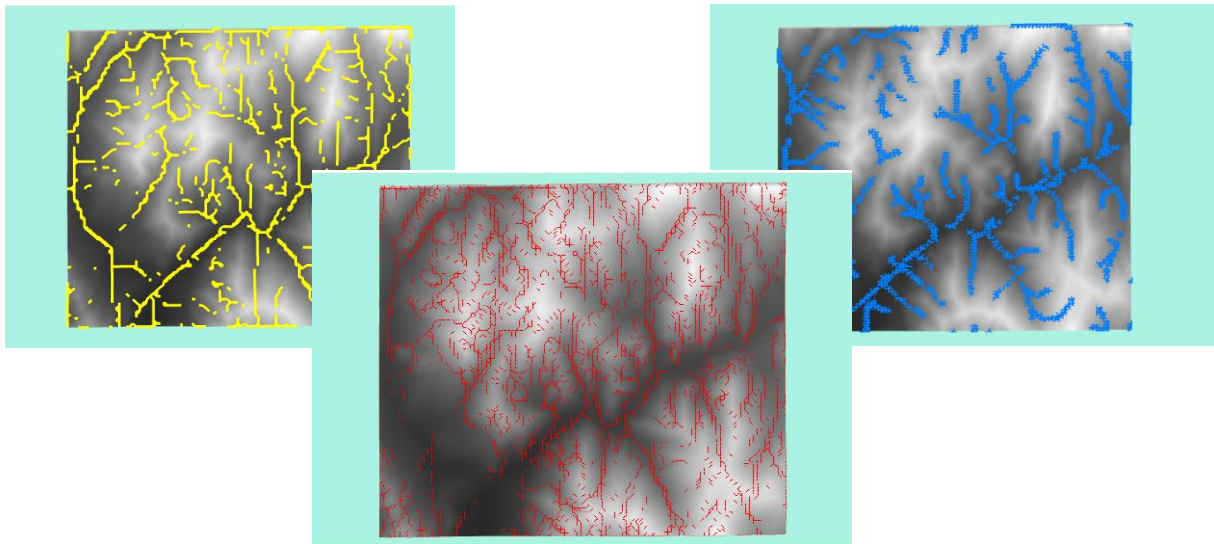


Imagen 19: Identificación de ríos.

De izquierda a derecha, se encuentra el algoritmo de Peucker, ángulo diedro y por último el algoritmo de Callaghan aplicado a un terreno. Cabe destacar, la similitud entre el algoritmo de Peucker y Callaghan, siendo el primero quién encuentra las redes de drenaje con más detalle, mientras que en Callaghan presenta las mismas redes de drenaje pero con ruido en el resultado entregado. Por otro lado, la identificación de redes de drenaje por medio del ángulo diedro presenta resultados que no permiten identificar de forma clara las redes de drenaje.

Para el caso de la identificación de redes de drenaje, el que mejor se comporta visualmente es el algoritmo de Peucker, ya que permite identificar claramente las redes de drenaje.

Cada uno de estas opciones, se puede llevar a cabo eligiendo el algoritmo y posteriormente dando la opción de ejecutar el algoritmo.

4.11.4 Opción de identificación de red de drenaje

Para la obtención de una red de drenaje, hay que proceder a la elección de qué algoritmo quiere ser aplicado para obtener la red de drenaje. A continuación, se presenta la obtención de la red de drenaje de la zona a través del algoritmo de Peucker.

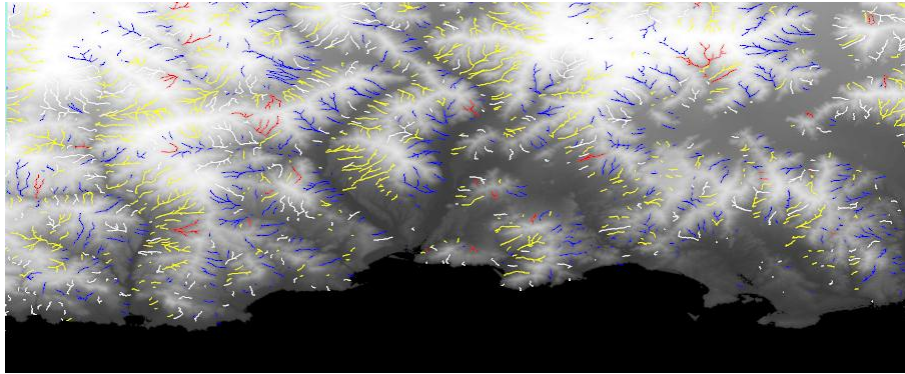


Imagen 20: Ejemplo de identificación de red de drenaje.

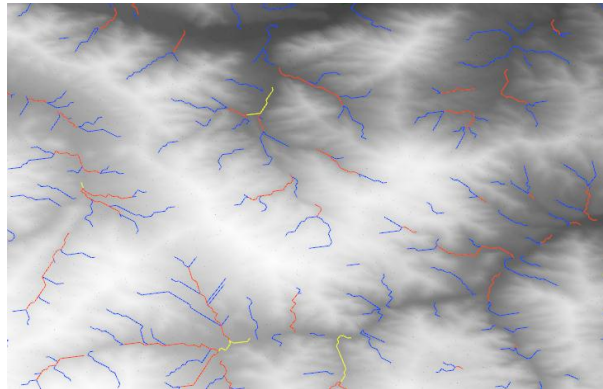


Imagen 21: Ejemplo con acercamiento de la identificación de redes de drenaje.

Se observa que cada línea perteneciente a una red, está de distinto color, esto se debe a que cada red de drenaje tiene aplicada la clasificación de Strahler-Horton, donde cada color se explica en la siguiente tabla:

Número de Strahler-Horton	Color
1	Blue
2	Red
3	Yellow
4	Magenta
5	Orange
6 >	Cyan

Tabla 2: Número de Strahler-Horton.

4.11.5 Opción de identificación de patrón

A continuación se presenta un ejemplo de la solución para la identificación de patrones de drenaje.

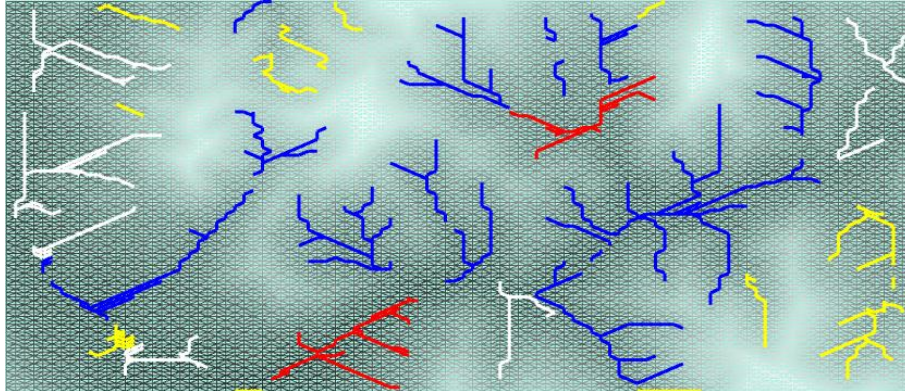


Imagen 22: Ejemplo de identificación de patrón de drenaje

De la imagen se observa que cada red está definida por un color. A continuación, se presenta la tabla con el patrón asociado a cada color:

Patrón de drenaje	Color
Dendrítico	Amarelo
Paralelo	Azul
Enrejado	Verde
Rectangular	Verde
No identificado	

Tabla 3: Color asociado a cada patrón de drenaje.

4.12 Métricas del software desarrollado

A continuación se presenta la métrica del software:

Métrica	Valor
Número de clases	24
Número de líneas	3105
Número de líneas de código	2470
Número de líneas de comentarios	635
Número de funciones	161
Ratio de líneas de comentario con líneas de código	0.26

Tabla 4: Métrica del software.

Se observa de las métricas que en promedio cada clase tienen 100 líneas de código, lo cual, en conjunto con el diseño de las clases se ha logrado tener modularidad en el código.

4.13 Elemento de ayuda al desarrollador

Los elementos de ayuda al desarrollador como se mencionó anteriormente fue la opción de ver los vectores normales en el terreno y los vectores con la gradiente de los triángulos.

A continuación se presentan lo antes mencionado:

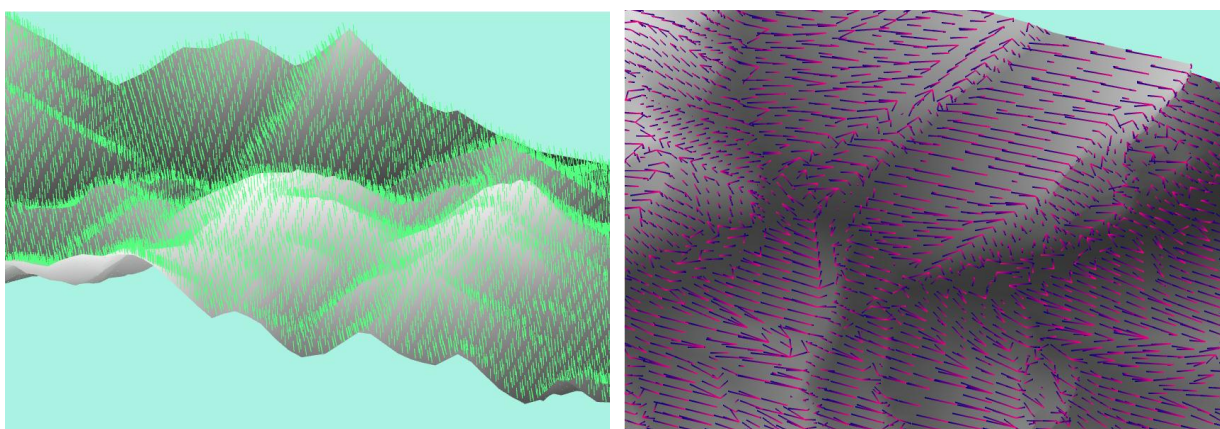


Imagen 23: Terreno con vectores normales y gradiente.

A la izquierda se encuentra el terreno con los vectores normales, mientras que a la derecha está los vectores con la gradiente de cada triángulo, donde el color más oscuro corresponde a la dirección del gradiente.

Capítulo 5: Validación de la solución

Para la validación de la solución implementada se usaron dos tipos de entrada de datos:

- Ingreso de datos a través de la selección de región con Google Earth
- Ingreso de datos por medio de un archivo tipo TIFF

Para cada una, se realizaron las siguientes acciones:

- Identificación de ríos, con cada uno de los algoritmos implementados.
- Detección de las redes producidas por las cuencas
- Detección de los patrones de drenaje
- Camino producido por el recorrido del agua desde un determinado lugar.

Cabe destacar, que para los resultados con ingreso de datos a través de archivo tipo TIFF, se procede a ser comparado con el software Rivertools.

5.1 Plataforma de pruebas

Para cada una de las pruebas que se realizan se procede a utilizar el sistema operativo Windows 7 x64, en un computador con procesador Intel® Core™ i5-3230M CPU @ 2.60GHz y memoria RAM de 8GB.

5.2 Metodología

Para la validación de los resultados se sigue la siguiente metodología:

5.2.1 Terreno creado con datos de Google Earth

Para la medición de los resultados, se estableció dos lugares con características distintas, estos fueron:

- Primer Sector: Costa de Chile entre el 32°27'S, 71°29'W y 32°33'S, 71°14'W
- Segundo Sector: Sector de Petorca entre 32°11'S, 71°01'W y 32°21'S, 70°47'W

Se realiza las siguientes mediciones:

- **Identificación de ríos:** Medición visual de similitud de la cantidad de ríos producidos por el software
- **Detección de red de drenaje:** Medición visual de la cantidad de redes de drenaje.
- **Detección de patrones:** Medición visual de similitud con el patrón identificado

- **Recorrido del agua:** Medición visual de similitud con el camino producido por el recorrido del agua desde un determinado lugar.

5.2.2 Terreno creado con datos de archivo TIFF

Para la medición de los resultados, se procederá a utilizar un archivo TIFF, al cual se realiza las acciones antes mencionadas.

- **Identificación de ríos:** Esta medición estará enfocada en la comparación de la cantidad de ríos identificados tanto para el software implementado como RiverTools.
- **Detección de red de drenaje:** Medición enfocada en la comparación de la cantidad de redes de drenaje en RiverTools y el software.
- **Detección de patrones:** Medición visual de similitud con el patrón identificado
- **Recorrido del agua:** Medición visual de similitud con el camino producido por el recorrido del agua desde un determinado lugar.

Con estas pruebas, se pretende validar los resultados entregados por el software.

5.3 Resultados

A continuación se presenta los resultados obtenidos de la aplicación de la metodología. Para ello, se presenta los resultados expuesto por Google Earth y posteriormente los obtenidos para los datos provenientes de archivos TIFF

5.3.1 Datos extraídos de Google Earth

5.3.1.1 Primer Sector



Imagen 24: Sector Costero en Google Earth.

El primer sector es el correspondiente a Costa de Chile entre el $32^{\circ}27'S$, $71^{\circ}29'W$ y $32^{\circ}33'S$, $71^{\circ}14'W$.

5.3.1.1.1 Identificación de ríos

Algoritmo Callaghan

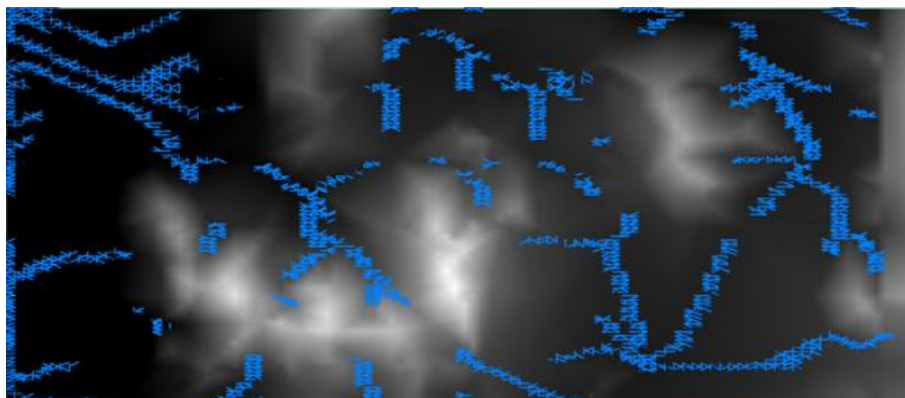


Imagen 25: Algoritmo de Callaghan para la identificación de ríos en sector costero.

Algoritmo de Peucker

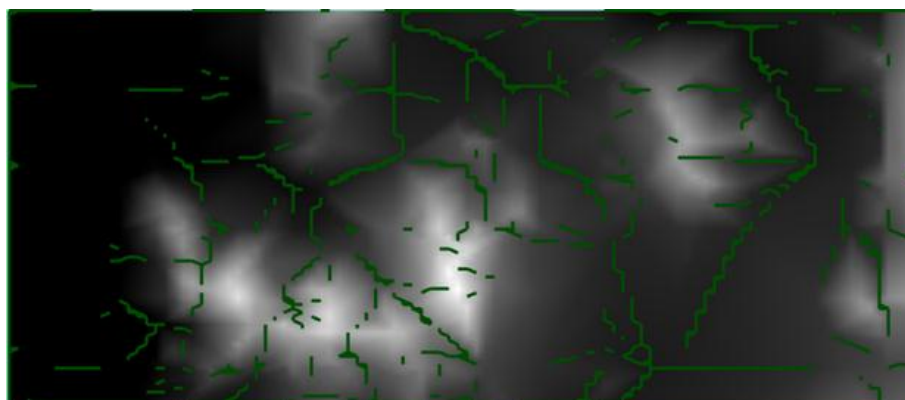


Imagen 26: Algoritmo de Peucker para la identificación de ríos en sector costero.

Algoritmo Ángulo Diedro

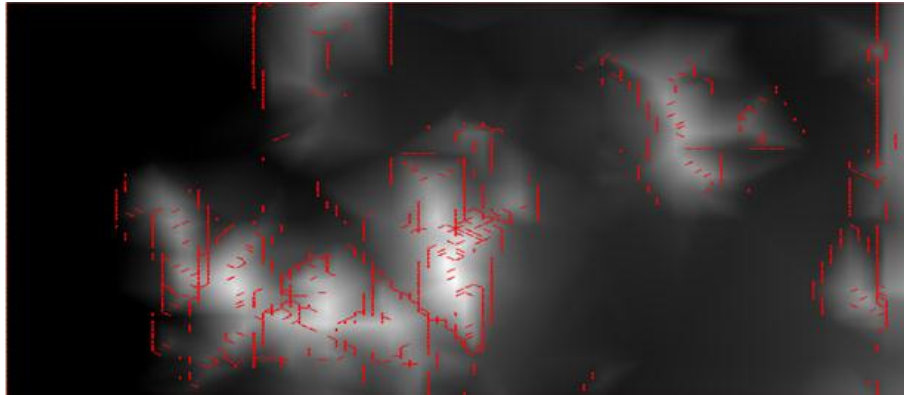


Imagen 27: Algoritmo por gradiente para identificación de ríos en sector costero.

De los resultados expuestos para la obtención de los ríos, se observa a través de la imagen obtenida de Google Earth, que tanto como el algoritmo de Callaghan como el de Peucker, se acercan a la obtención de los ríos/cuencas que pueden ser identificados, siendo los resultados de Peucker en que presenta cuencas pronunciadas del terreno y continuas. Por otro lado, la identificación de redes de drenaje por medio del ángulo diedro presenta resultados que no se adecuan a lo que se trataba de identificar. Cabe destacar, que al ser una zona con costa, ninguno de los métodos logró identificar lo que ya era mar, haciendo que se encontraran redes de drenaje en dicho lugar, esto es debido a que la muestra de datos obtenida donde está el mar, posee distintas alturas.

5.3.1.1.2 Detección de red de drenaje

Algoritmo Callaghan

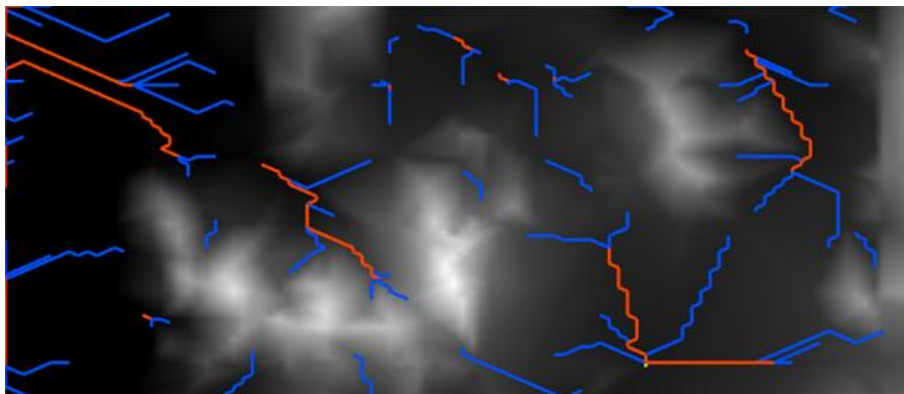


Imagen 28: Algoritmo de Callaghan para detección de red de drenaje en sector costero.

Algoritmo Peucker

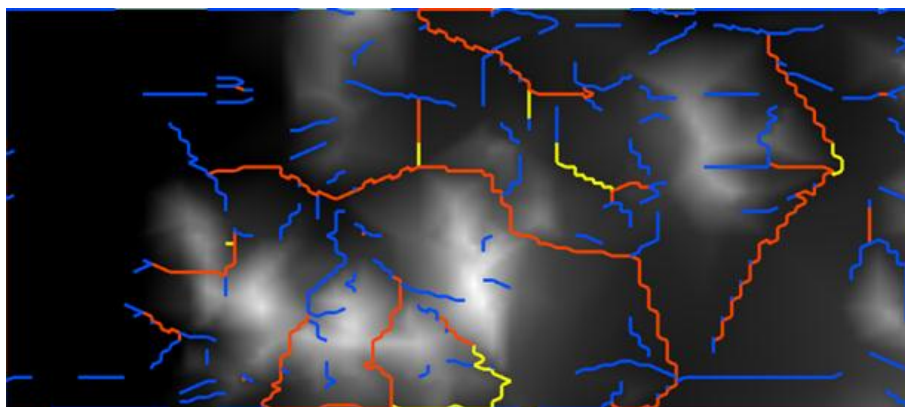


Imagen 29: Algoritmo de Peucker para detección de red de drenaje en sector costero.

Para la obtención de redes de drenaje tanto de Peucker como Callaghan. Se obtuvo que la mayor cantidad de redes encontradas fue a través del algoritmo de Peucker, ya que este presentaba una mayor conexión entre sus caudales.

5.3.1.1.3 Detección de patrones de drenaje

Algoritmo ZhangGuilbert utilizando red de drenaje por Callaghan

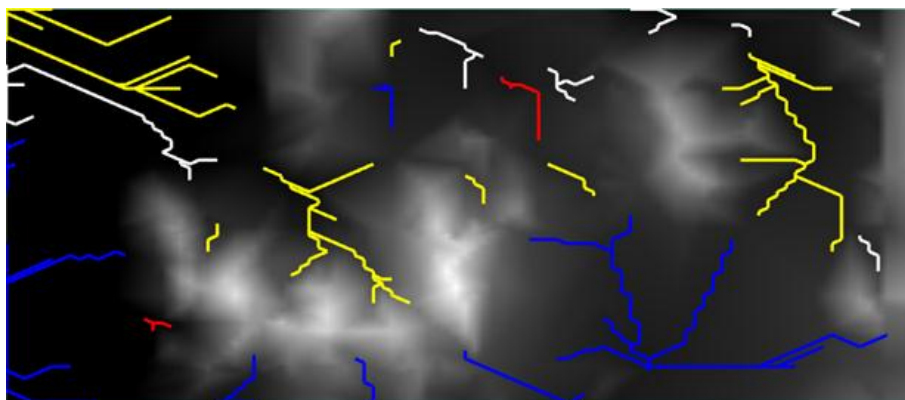


Imagen 30: Algoritmo de ZhangGuilbert con Callaghan para detección de patrón de drenaje en sector costero.

Algoritmo ZhangGuilbert utilizando red de drenaje por Peucker

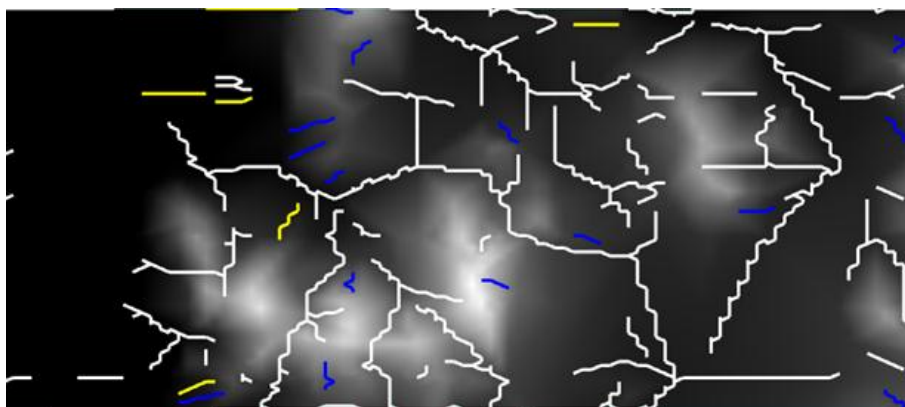


Imagen 31: Algoritmo de ZhangGuilbert con Peucker para detección de patrón de drenaje en sector costero.

En las imágenes anteriores, se observa la aplicación del algoritmo de ZhangGuilbert en el resultado obtenido a las redes de drenaje producidas por el algoritmo de Peucker y Callaghan. El resultado del algoritmo de ZhangGuilbert se debe principalmente al tipo de redes encontradas, las cuales eran demasiado extensas para el caso de las redes provenientes del algoritmo de Peucker, mientras que las redes provenientes del algoritmo de Callaghan eran continuas y pequeñas haciendo posible identificarla, esto producía que esta última combinación era la mejor para obtener el patrón de la red de drenaje.

5.3.1.1.4 Recorrido del agua

Algoritmo Callaghan

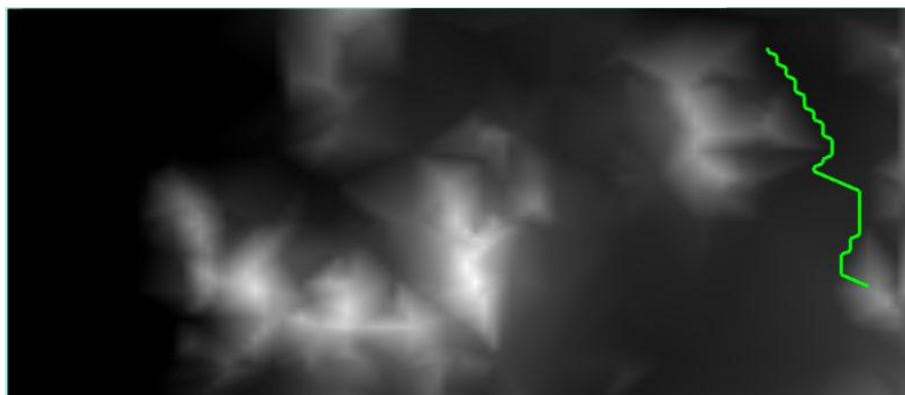


Imagen 32: Recorrido del agua por medio del algoritmo de Callaghan en sector costero.

Algoritmo por gradiente

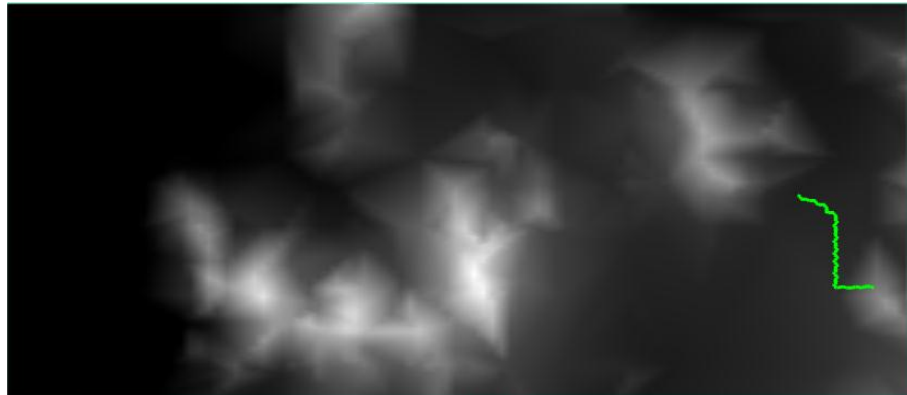


Imagen 33: Recorrido del agua por medio del algoritmo por gradiente en sector costero.

Para la identificación del recorrido del agua en el terreno de la costa, se observa que el caudal que tiene un mayor alcance es el proveniente del algoritmo de Callaghan, esto es producido a que tiene un valor configurable que permite avanzar incluso si la altura es mayor a la posición donde se encuentra, sólo si ésta está dentro del valor configurado.

5.3.1.2 Segundo Sector

El segundo sector corresponde al sector cercano a Petorca entre $32^{\circ}11'S$, $71^{\circ}01'W$ y $32^{\circ}21'S$, $70^{\circ}47'W$. A continuación se presenta los resultados correspondientes:



Imagen 34: Sector de Petorca en Google Earth.

5.3.1.2.1 Identificación de ríos

Algoritmo Callaghan

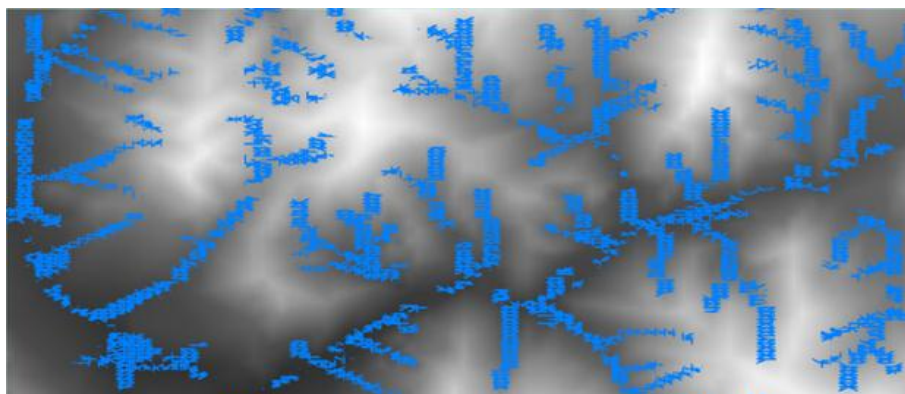


Imagen 35: Algoritmo de Callaghan para identificación de ríos en sector de Petorca.

Algoritmo de Peucker

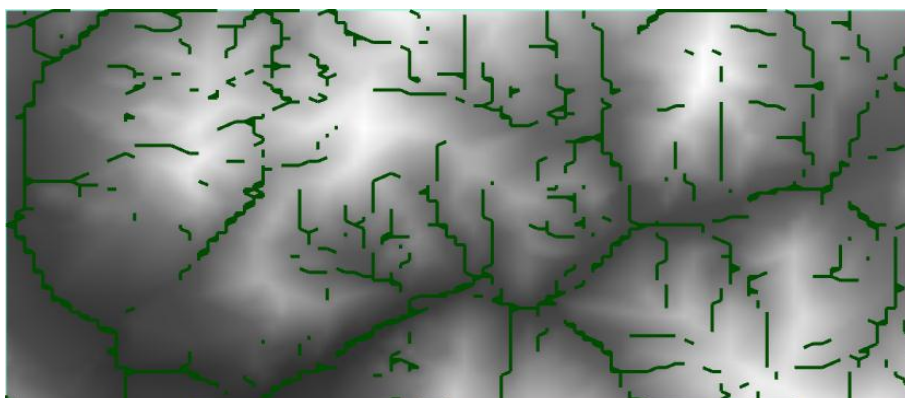


Imagen 36: Algoritmo de Peucker para identificación de ríos en sector de Petorca.

Algoritmo Ángulo Diedro

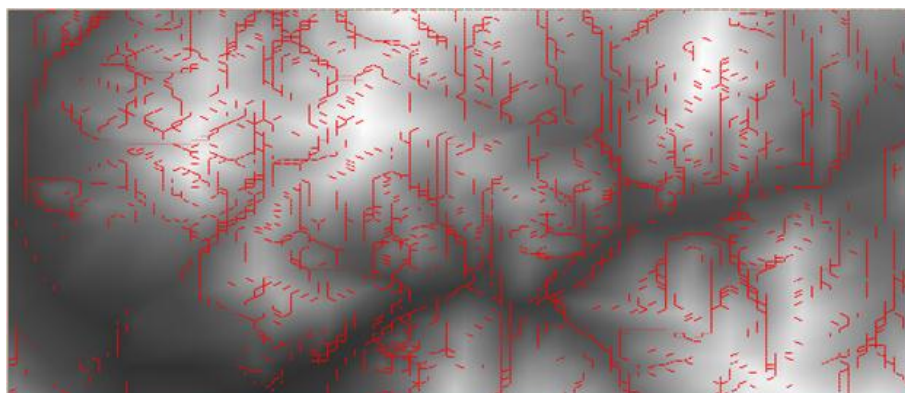


Imagen 37: Algoritmo por gradiente para identificación de ríos en sector de Petorca.

De las imágenes antes presentadas se infiere que para este sector, la identificación de ríos es mayor que para el primer sector. Donde nuevamente la identificación por el algoritmo de Peucker resulta ser más similar que las otras 2.

5.3.1.2.2 Detección de red de drenaje

Algoritmo Callaghan

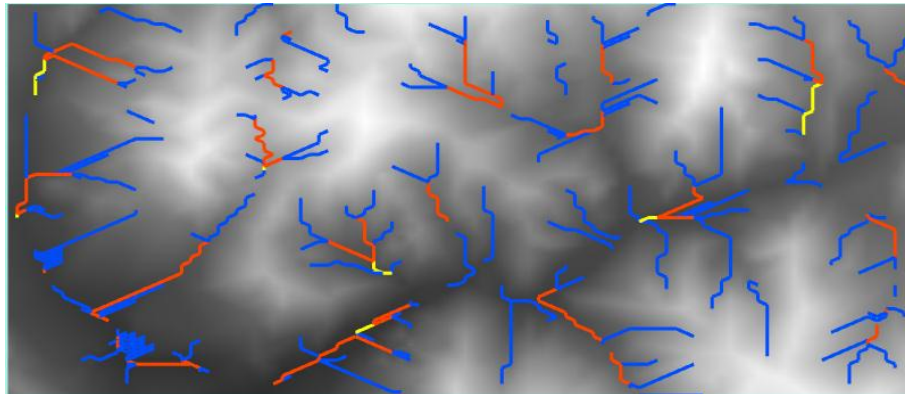


Imagen 38: Algoritmo de Callaghan para detección de red de drenaje en sector de Petorca.

Algoritmo Peucker

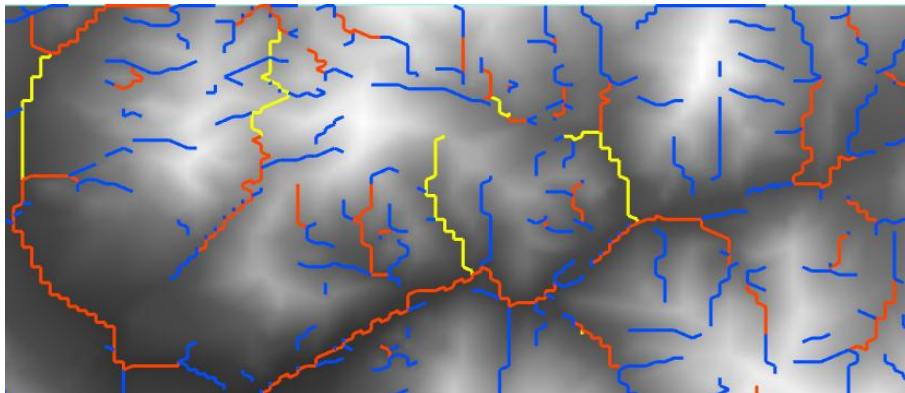


Imagen 39: Algoritmo de Peucker para detección de red de drenaje en sector de Petorca.

Se observa de las imágenes que a través del algoritmo de Peucker se logra obtener una mayor cantidad de redes de drenaje.

5.3.1.2.3 Detección de patrones de drenaje

Algoritmo ZhangGuilbert utilizando red de drenaje por Callaghan

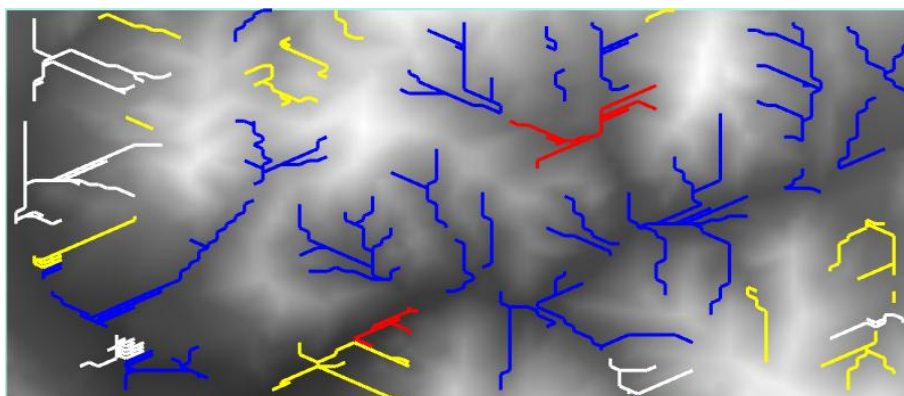


Imagen 40: Algoritmo de ZhangGuilbert con Callaghan para detección de patrón de drenaje en sector de Petorca.

Algoritmo ZhangGuilbert utilizando red de drenaje por Peucker

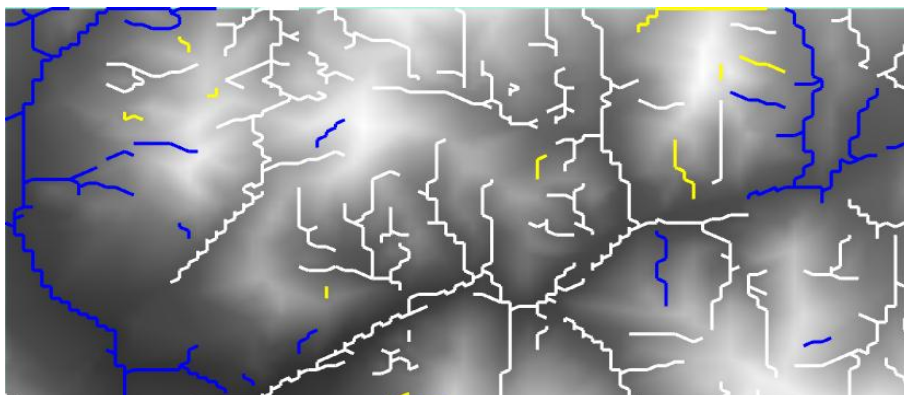


Imagen 41: Algoritmo de ZhangGuilbert con Peucker para detección de patrón de drenaje en sector costero.

De la misma forma que en el primer sector, se tiene que aunque se haya encontrado una mayor cantidad de redes de drenaje con Peucker, la identificación de patrones en estas redes de drenaje es mucho menor que en comparación con la aplicación del algoritmo de Callaghan para la obtención de redes de drenaje, y su posterior identificación de patrones, donde se observa que es mayor. Cabe destacar, que la identificación del patrón de drenaje en cada red, es parcial, ya que la comparación visual con los diagramas de cada patrón, nos demuestra que algunos pertenecen al correspondiente patrón mientras que otros se alejan de aquella clasificación.

5.3.1.2.4 Recorrido del agua

Algoritmo Callaghan

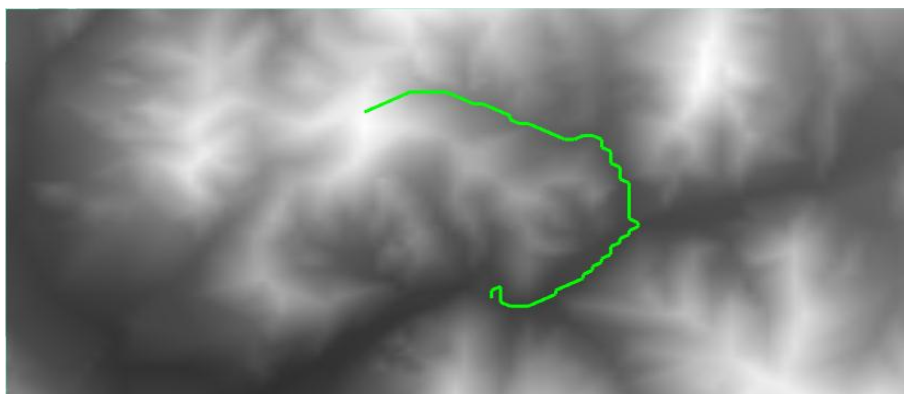


Imagen 42: Recorrido del agua por medio del algoritmo de Callaghan en sector de Petorca.

Algoritmo por gradiente

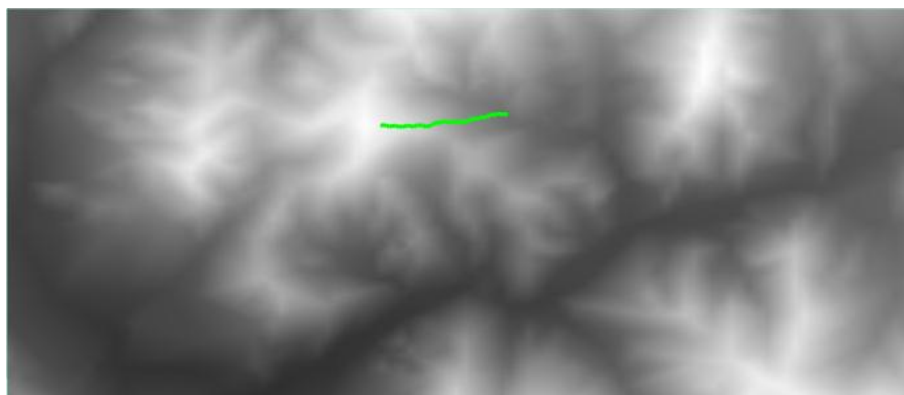


Imagen 43: Recorrido del agua por medio del algoritmo por gradiente en sector de Petorca.

La identificación del recorrido del agua a través del algoritmo de Callaghan permite obtener un recorrido más prologando que por el algoritmo por gradiente.

5.3.2 Datos extraídos desde archivos TIFF

5.3.2.1 TIFF



Imagen 44: Datos extraídos del archivo TIFF.

5.3.2.1.1 Identificación de ríos

Algoritmo Callaghan

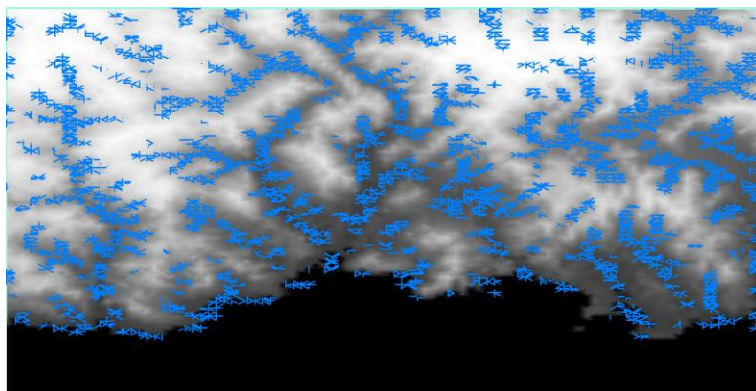


Imagen 45: Algoritmo de Callaghan para la identificación de ríos para archivo TIFF.

Algoritmo Peucker

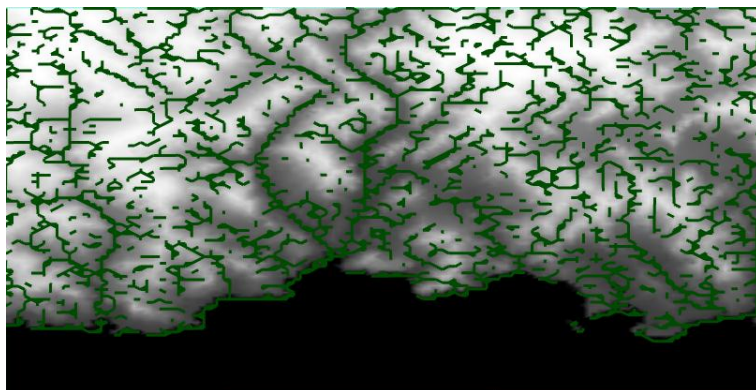


Imagen 46: Algoritmo de Peucker para la identificación de ríos para archivo TIFF.

Algoritmo de Ángulo Diedro

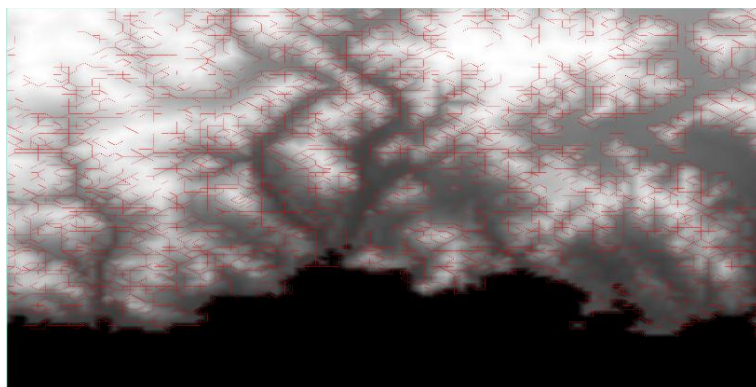


Imagen 47: Algoritmo de ángulo diedro para la identificación de ríos para archivo TIFF.

RiverTools

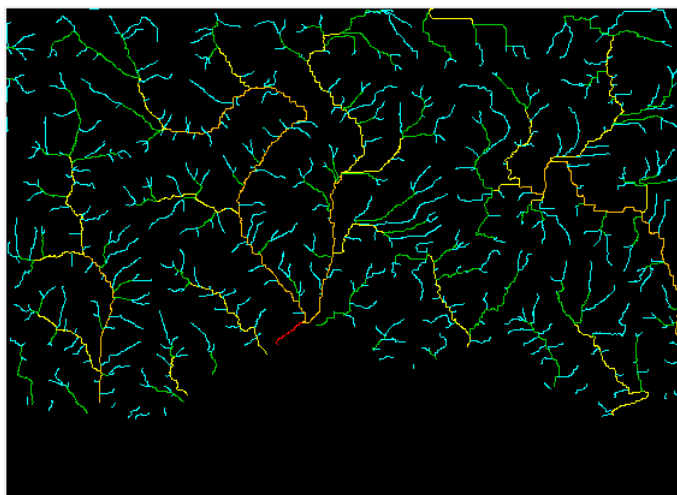


Imagen 48: Resultado de Rivertools para la obtención de las redes de drenaje para archivo TIFF.

Se observa de estos resultados que a través de la identificación de ríos, el algoritmo de Peucker consigue ser el algoritmo que obtiene las redes de drenaje similar al obtenido por RiverTools.

5.3.2.1.2 Detección de red de drenaje

Algoritmo Callaghan

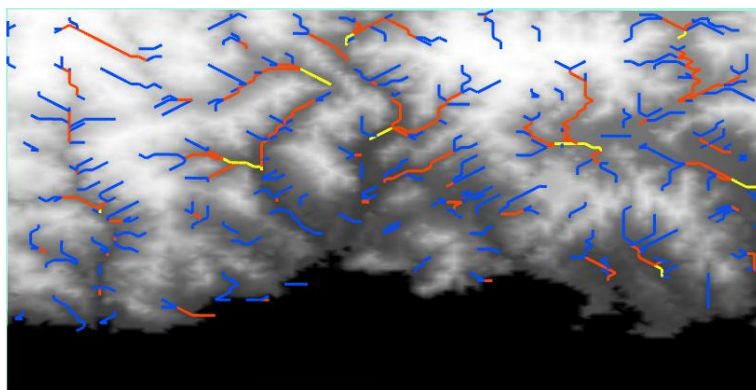


Imagen 49: Algoritmo de Callaghan para detección de red de drenaje en archivo TIFF.

Algoritmo Peucker

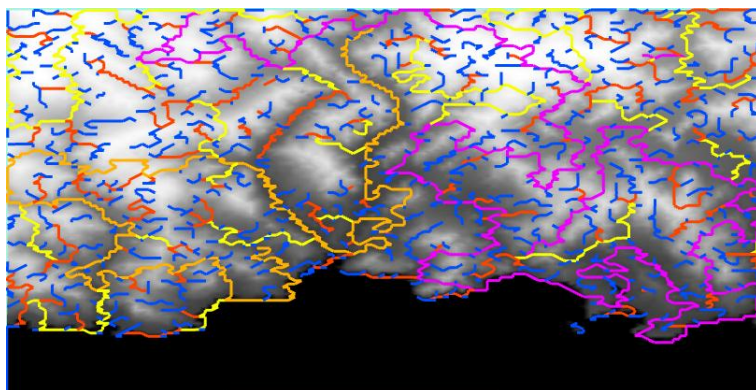


Imagen 50: Algoritmo de Peucker para detección de red de drenaje en archivo TIFF.

RiverTools

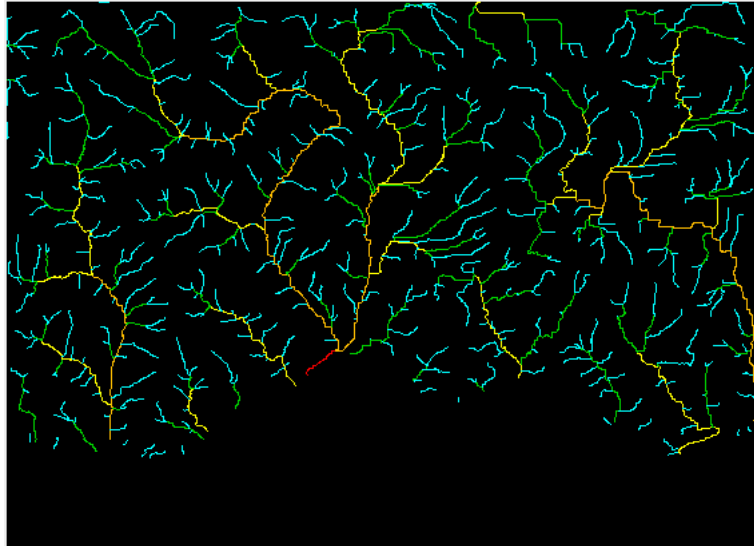


Imagen 51: Resultado de Rivertools para la obtención de redes de drenaje para archivo TIFF.

En la obtención de las redes de drenaje con la clasificación de Strahler-Horton, el algoritmo de Peucker obtiene gran cantidad de resultados clasificados, los cuales son similares a los entregados por RiverTools, pero la clasificación de estas redes difiere. Para el caso del algoritmo de Callaghan, la poca cantidad de redes de drenaje encontradas produce que se aleje de los resultados esperados.

5.3.2.1.3 Detección de patrones de drenaje

Algoritmo ZhangGuilbert utilizando red de drenaje por Callaghan

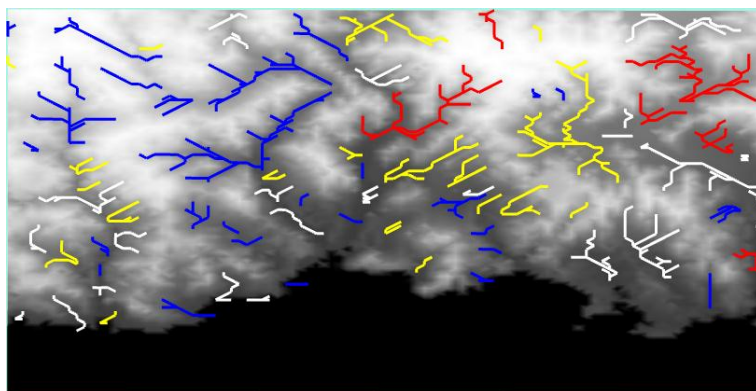


Imagen 52: Algoritmo de ZhangGuilbert con Callaghan para detección de patrón de drenaje para archivo TIFF.

Algoritmo ZhangGuilbert utilizando red de drenaje por Peucker

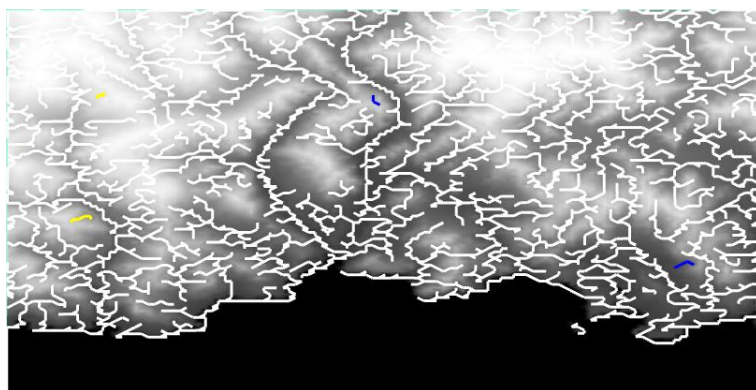


Imagen 53: Algoritmo de ZhangGuilbert con Peucker para detección de patrón de drenaje para archivo TIFF.

Para la detección de patrones de drenaje, no hubo comparación con RiverTools, pero de la misma forma que los resultados expuestos para los provenientes de Google Earth, se tiene que el algoritmo de Callaghan permite la identificación por medio del algoritmo de ZhangGuilbert. Y este algoritmo identifica los patrones de forma parcial, haciendo que la identificación de las redes de drenaje con comparación a los dibujos de los patrones de drenaje, sean visiblemente similares en algunos casos.

5.3.2.1.4 Recorrido del agua

Algoritmo Callaghan

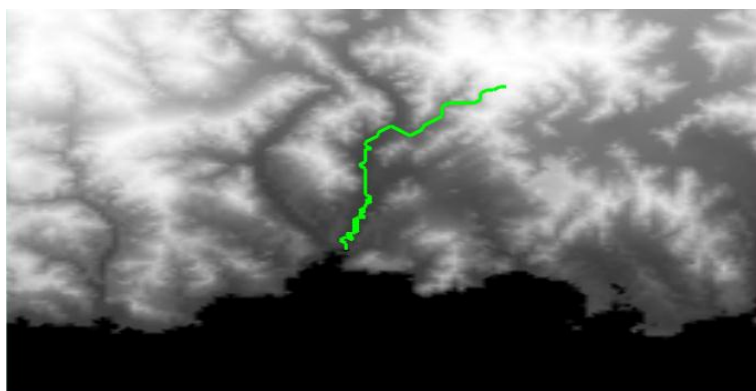


Imagen 54: Recorrido del agua por medio del algoritmo de Callaghan en archivo TIFF.

Algoritmo por gradiente

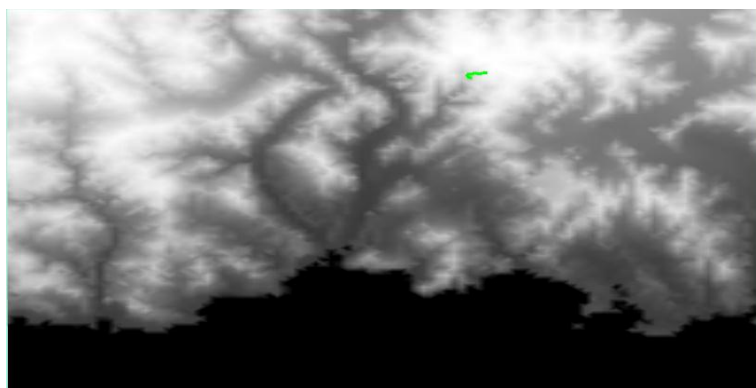


Imagen 55: Recorrido del agua por medio del algoritmo por gradiente en archivo TIFF.

RiverTools

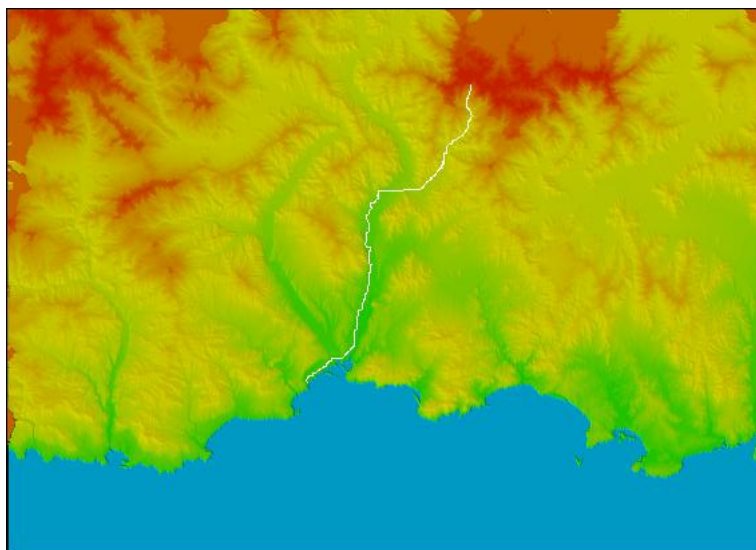


Imagen 56: Resultado del recorrido del agua de RiverTools para archivo TIFF.

De los resultados entregados por cada algoritmo y el software RiverTools, se observa que el algoritmo de Callaghan para obtener el recorrido del agua es el que tiene mayor similitud al entregado por RiverTools, donde el culmine del agua en ambas es en el mar.

Capítulo 6: Conclusiones

El presente trabajo se logró obtener una herramienta que permite a la comunidad científica, el poder visualizar terrenos mediante la obtención de datos a través de Google Earth y archivos TIFF. Dentro de las propiedades del software creado se tiene la identificación de caudales, redes de drenaje y patrones de drenaje al cual pertenece cada red de drenaje. A su vez, es posible con esta herramienta la obtención del recorrido del agua, mediante el ingreso de la posición de inicio del agua.

La herramienta cuenta con tres tipos de algoritmos para la identificación de redes de drenaje, los cuales son: algoritmo de Peucker, Callaghan y por ángulo diedro; y uno para la identificación de patrones de drenaje llamado ZhangGuilbert. Cada algoritmo implementado entrega resultados que se acercan a lo esperado dependiendo de lo que se desee identificar, siendo mejor el algoritmo de Peucker para los casos de identificación de redes de drenaje, ya que los resultados entregados por este algoritmo mostraban ríos continuos y similares a los entregados en RiverTools; y la combinación dos algoritmos para el caso de clasificar patrones de drenaje, siendo estos algoritmos el de ZhangGuilbert para clasificación de patrones y el de Callaghan para la clasificación de red de drenaje. Esta combinación de algoritmos entrega una mayor cantidad de resultados clasificados en patrones de drenaje en comparación a las demás. Cabe destacar que la identificación de patrones de drenaje se encuentra desarrollada de forma parcial, ya que está necesita ser mejorada o ser complementada con otros algoritmos que permitan la identificación y validación de otros patrones de drenaje como también, los implementados. Es por ello que es de gran utilidad la otros algoritmos que provean una clasificación de patrones más robusta en resultados.

Si bien los resultados obtenidos no se acercan del todo a los expuestos a los de RiverTools, la experimentación en la utilización de varios algoritmos permite encontrar resultados y soluciones a lo expuesto en el presente trabajo. Cabe destacar, que la investigación realizada permita el trabajo interdisciplinario en este campo de investigación para el desarrollo de nuevas soluciones.

6.1 Trabajo Futuro

Los resultados obtenidos en esta memoria fueron positivos en cuanto a los objetivos logrados. Sin embargo, este tema aún presenta muchos desafíos y problemas a ser resueltos. A continuación se presenta una lista con ideas de trabajo que haría que esta aplicación mejore y sea de más ayuda a la comunidad.

- Identificación de terrazas en un determinado lugar.
- Visualización de terrazas en caudal seleccionado
- Identificación de patrones de drenaje utilizando machine learning.

- Identificación y visualización de patrones de drenaje que compartan atributos similares
- Identificación de sectores de un caudal que difieran del patrón predominante.
- Implementación de otros algoritmos para detectar un patrón de drenaje.
- Implementación de Búsqueda por similitud en polígonos
- Identificación de patrones de drenaje no individuales, como radial y anular.
- Comparación automatizada de los resultados de las distintas herramientas.
- Arreglo de la conversión de coordenadas geográfica latitud, longitud a coordenadas geográfica mundial.
- Reconocimiento de patrones de drenaje a través de los algoritmos expuesto en [26] y [27]

Bibliografía

- [1] RIVIX, «Rivix LLC,» 17 Noviembre 2014. [En línea]. Available: <http://rivix.com/index.php>.
- [2] INECC, «Instituto Nacional de Ecología y Cambio Climático,» [En línea]. Available: <http://mapas.inecc.gob.mx/mediofisico/hidrologia/tirehi>. [Último acceso: 17 11 2014].
- [3] P. P. Pons, «Proyecto Runnel en Github,» [En línea]. Available: <https://github.com/Pillin/runnelProyect.git>. [Último acceso: 22 Enero 2015].
- [4] D. Campos Aranda, Procesos del ciclo hidrológico, México: Universitaria de Potosina, 1998.
- [5] M. López Vergara, Manual de Fotogeología, España: Servicio de publicaciones J.E.N, 1971.
- [6] F. Sánchez Menéndez, Geodesia y Cartografía, España: EOSGIS, 2004.
- [7] C. B. Leal da Silva, «TFC Triangulaciones,» [En línea]. Available: <http://www.dma.fi.upm.es/gregorio/JavaGC/MWT/teoria.html>. [Último acceso: 18 11 2014].
- [8] M. Dorzán, E. Olinda, P. Palmero y G. Hernández, Una herramienta para la generación y visualización de triangulaciones y pseudotriangulaciones, Argentina: Facultad de ciencias físico, matemáticas y naturales. Universidad de Nacional de San Luis, 2010.
- [9] M. Reparaz y N. Rodríguez, Triangulación de Delaunay de alto orden en el terreno práctico de los sistemas de información geográfica, Argentina: Facultad de ciencias exactas y naturales, Universidad de Buenos Aires, 2014.
- [10] M. Yvinec, «CGAL: Computational Geometry Algorithms Library,» [En línea]. Available: http://doc.cgal.org/latest/Triangulation_2/index.html#Chapter_2D_Triangulations. [Último acceso: 18 11 2014].
- [11] «MathWorks,» Matlab, [En línea]. Available: <http://www.mathworks.com/help/matlab/ref/delaunaytriangulation-class.html>. [Último acceso: 18 11 2014].
- [12] F. Sanchez Menendez, Georreferenciación de Cartografía: Datos Raster y Vectoriales, España: EOSGIS, 2009.
- [13] G. Developers, «Preguntas Frecuentes de Google Maps,» [En línea]. Available: <https://developers.google.com/maps/faq#whatis>. [Último acceso: 18 11 2014].
- [14] G. Developers, «Api de Elevación de Google Earth,» [En línea]. Available: <https://developers.google.com/maps/documentation/elevation>. [Último acceso: 18 11 2014].
- [15] Q. Project, «Qt page,» [En línea]. Available: <http://qt-project.org/>. [Último acceso: 18 11 2014].
- [16] «Wxwidgets page,» [En línea]. Available: <http://www.wxwidgets.org/>. [Último acceso: 18 11 2014].
- [17] «GTK page,» [En línea]. Available: <http://www.gtk.org/>. [Último acceso: 18 11 2014].
- [18] Kronos, «OpenGL,» [En línea]. Available: <https://www.khronos.org/opengles/>. [Último acceso: 18 11 2014].

- [19] «Centro de artigos,» [En línea]. Available: http://centrodeartigos.com/articulos-utiles/article_119999.html. [Último acceso: 18 11 2014].
- [20] M. M. Jarek Jasiewicz, «OSGeo SVN Repositories,» [En línea]. Available: <https://svn.osgeo.org/grass/grass-addons/grass6/raster/r.stream.order/description.html>. [Último acceso: 27 11 2014].
- [21] L. Yaolin, M. Molenaar y M.-J. Kraak, «Automated Organization of Hierarchical Catchments in River Network Based Constrained Delaunay Triangulation,» The Netherlands Department of Geo-Informatics, International Institute of Geo-Information Science and Earth Observation, 2002.
- [22] Q. Xu, L. Liu y Z. Zhou, «Detection of River Ice Using Relevance Vector Machine, School of remote sensing and information engineering,» Wuhan University, Wuhan, 2011.
- [23] S. Berchtold , D. Keim y H. Kriegel, «Using extended feature objects for partial similarity retrieval,» VLDB Journal, 1997.
- [24] M. Van Kreveld, «Digital Elevation Models: Overview and selected TIN Algorithms,» Department of Computer Science Utrecht University, the Netherland, 1996.
- [25] Z. Ling y E. Guilbert, «Automatic drainage pattern recognition in river networks,» Department of Land Surveying and Geo-Informatics, The Hong Kong Polytechnic University, Hung Hom, Kowloon, Hong Kong, 2013.
- [26] N. Haala y G. Vosselman, «Recognition of road and river patterns by relational matching,» Institute of Photogrammetry, Stuttgart University, Germany, 1992.
- [27] F. Heinzle, K. Anders y M. Sester, «Automatic detection of patterns in road networks,» Institute of Cartography and Geoinformatics, University of Hannover,, Germany, 2006.
- [28] «CGIAR,» [En línea]. Available: <http://srtm.csi.cgiar.org/SELECTION/inputCoord.asp>. [Último acceso: 28 11 2014].
- [29] GeoNames, «The GeoNames geographical database,» GeoNames, [En línea]. Available: [http://www.geonames.org/advanced-search.html?q=&country=CL&featureClass=H&continentCode=.](http://www.geonames.org/advanced-search.html?q=&country=CL&featureClass=H&continentCode=) [Último acceso: 28 11 2014].