Universidad de las Ciencias Informáticas

Facultad de Ciencias y Tecnologías Computacionales

Módulo de Visualización 3D en la plataforma web ULTRON 2.0

Trabajo de diploma para optar por el título de Ingeniero en Ciencias Informáticas

Autores:

Grabiel González Lorenzo

Ricardo Díaz Vilches

Tutores:

MSc. Yunet González Mulet

Ing. Reinier Fernández Coello

La Habana, septiembre 2020

Año 62 de la Revolución

Declaración de autoría

Declaramos ser los únicos autores del presentetrabajo de diploma, y autorizamos a la Facultad de Ciencias y Tecnologías Computacionales CITEC de la Universidad de las Ciencias Informáticas hacer uso del mismo en su beneficio.

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

_____

Firma del autor
Grabiel González Lorenzo

_____

Firma del autor
Ricardo Díaz Vilches

_____

Firma del tutor
MSc. Yunet González Mulet

_____

Firma del tutor
Ing. Reinier Fernández Coello

Frase

"El futuro de nuestra Patria tiene que ser necesariamente un futuro de hombres de Ciencia."

Fidel Castro Ruz

Agradecimientos

A la Revolución y a Fidel por darme la oportunidad de estudiar, formarme como profesional y como persona en esta maravillosa universidad.

A la memoria de mi padre Luis Enrique.

A mi madre Belkis y mi padre Victor que gracias por su apoyo y abnegación he podido salir adelante y alcanzar todas mis metas. Gracias por su dedicación y comprensión.

A mishermanos, tíos y primospor estar siempre preocupados por mí. Gracias por su cariño y amor.

A “Mi Cosita” que de cariño le digo a mi esposaDailyporque siempre ha estado preocupada por mis estudios.

A mis tutoresYunet y en especial a Reinier queestuvo todo el tiempo ayudándonos, por su paciencia y deseos de que avanzáramos siempre.

Al grupo de trabajo del proyecto AplicativosSIGque sin su ayuda no hubiera sido posible lograr nuestras metas.

Al Rafa y a Ramses, y a todos mis amigos y compañeros de aula con los que siempre he podido contar.

A mis profesoresque tuvieron la paciencia de enseñarme técnicamente y formar parte de mi educación como persona.

A todos los que de una manera u otra influyeron en mi formación, muchas gracias.

Grabiel

Hoy, después de 5 años de esfuerzo y sacrificio se cumple uno de mis grandes sueños: graduarme de Ingeniero en Ciencias Informáticas. Durante esta travesía de altos estudios, estuve acompañado de varias personas que en todo este tiempo formaron parte de mi vida,por eso mis más sinceros agradecimientos.

A mis tutores Reinier Fernández Coello y Yunet González Mulet quienes con sus conocimientos y apoyo me guiaron a través de cada una de las etapas de este proyecto.

A la Universidad de Ciencias Informáticas por brindarme todos los recursos y herramientas que fueron necesarios para llevar a cabo el proceso de investigación.

A mis profesores por su contribución a mi formación académica.

A mis compañeros y a mi familia,por apoyarme aun cuando mis ánimos decaían.

Y en especial a mis padres, que siempre estuvieron ahí para darme palabras de apoyo y un abrazo reconfortante para renovar energías.

Ricardo

Dedicatoria

A toda mi familia que es una parte esencial de mí, a ellos le dedico cada paso que he logrado dar hasta el día de hoy.

A mi mamá y mis dospapás, la persona que soy hoy y todo lo que he logrado es gracias a ellos.

Grabiel

Quiero dedicar este trabajo de diploma a mi familia por estar junto a mí en estos 5 años de estudios:

A mis padres que siempre estuvieron ahí presentes.

A mis tutores que sin su ayuda hoy no estaría aquí cumpliendo este grandioso sueño.

Ricardo

Resumen

La visualización en tres dimensiones resulta útil para la interpretación de los datos proporcionando información detallada del entorno. En este sentido un Sistema de Información Geográfica en tres dimensiones mantiene en topología tridimensional la información espacial y temática. Cuba no queda exenta a este proceso global, por lo que se han dedicado recursos para alcanzar una mayor aceptación y utilización de estos sistemas por parte de la sociedad cubana. Es por ello que surge el proyecto AplicativosSIG en el centro de Representación y Análisis de Datos, perteneciente a la Universidad de las Ciencias Informáticas. Entre los proyectos productivos que se desarrollan en dicho centro se encuentra el proyecto ULTRON, el cual actualmente solo realiza representaciones en dos dimensiones, lo cual limita las posibilidades de visualizar con más detalle el terreno. Por esta razón la presente investigación tiene como objetivo general desarrollar un módulo de visualización en tres dimensiones para la plataforma web ULTRON 2.0.Para lograr este objetivo, se utilizó como metodología de desarrollo el Proceso Unificado Ágil, versión establecida por la Universidad de las Ciencias Informáticas, así como un conjunto de tecnologías definidas por el equipo de arquitectura del proyecto, permitiendo de esta forma el desarrollo de la presente investigación. Los resultados fueron validados a través de métricas y pruebas de software, donde se pudo comprobar la calidad de los artefactos, así como de las funcionalidades internas y externas del módulo propuesto.

Palabras claves:información geográfica, modelo digital, terreno, topografía, visualización en 3D.

Índice de contenidos

Índice de figuras

Índice de tablas

Introducción

El mundo se encuentra en constante cambio, la necesidad humana de vivir en mejores condiciones le ha dado lugar al surgimiento de las Tecnologías de la Información y las Comunicaciones (TICs). La informática desempeña un papel importante en la sociedad, donde se hace necesario un mejor manejo y gestión de la información, así como la informatización de diferentes sectores de la misma. Esta constituye la ciencia fundamental del desarrollo de las TICs y los avances de estas han sido muy utilizados para crear herramientas que faciliten un mejor almacenamiento y control de los datos(Muñoz Aldana, 2010).

En consecuenciacon lo anteriormente descrito, el tratamiento geográfico de la información se ha vuelto imprescindible y sus posibilidades de aplicación se hacen cada vez mayor gracias a los Sistemas de Información Geográfica (SIG). Esta tecnología permite llevar a cabo la representación cartográfica, realizar evaluaciones ambientales y de recursos naturales, estudiar y evaluar las redes de servicios de electricidad, telefonía, emergencias médicas, transportes, la ubicación geoespacial, entre otras. El impacto de los SIG ha logrado la versatilidad y desarrollo en cuanto a la toma de decisiones en las diferentes esferas de la sociedad, permitiendo que la herramienta cumpla con las funcionalidades atribuidas y ofrezca los servicios de georreferenciación y análisis de los datos según la entidad que lo utilice. Estos han podido revolucionar la forma de representar la información geográficacon numerosas facilidades que brindan: rapidez, accesibilidad, mejor nivel de detalles, fácil almacenamiento y actualización de los datos. Esto es posible, ya que permitenel análisis de patrones, relaciones y tendencias en la información, todo con el interés de contribuir a la toma de mejores decisiones(Olaya, 2019).

El uso de SIG facilita la visualización de los datos obtenidos en un mapa con el fin de reflejar y relacionar fenómenos geográficos de cualquier tipo. Estos agrupan en distintas capas la información geográfica, a medida que se realiza el acoplamiento de estas se muestra el panorama visual resultante, que permite el uso del sistema como referencia geográfica(OJEDA, 2000). Además, permiten llevar a cabo la cartografía digital, obteniendo los datos para el desarrollo del Modelo Digital del Terreno (MDT). El objetivo del modelado que se realiza, es crear una similitud entre el modelo y el objeto, llevando a dicho modelo la propiedad que en un futuro permita reconstruir esa parte de la realidad que se quiere representar del terreno.

Los MDT son una estructura numérica de datos que representa la distribución espacial de una variable cuantitativa y continua. Los modelos digitales del terreno son,por consiguiente, modelos simbólicos, ya que las relaciones de correspondencia que se establecen con el objeto real tienen la forma de algoritmos o formalismos matemáticos. A partir de la definición anterior, se pueden enunciar las propiedades básicas de los MDT(Fernández de laTorre , y otros, 2004).

- Forman una estructura de datos, lo que significa que no son sólo una acumulación o lista decifras, sino que guardan relaciones entre ellos. (son el producto de una modelización de datos).
- Representan la distribución espacial de una variable, lo que acota claramente su ámbito de actuación en la modelación.
- Sus variables son cuantitativas y continuas.

Existen varios tipos de MDT en los que se encuentra el modelo digital multivariable, las cuencas visuales, los modelos de reflectancia, las líneas de flujos, los modelos de caudales máximos y uno de los que más es utilizado es el Modelo Digital de Elevación (MDE).

Los MDT junto con los SIG son un factor importante para el conocimiento de la información geográfica a través de mapas y representaciones, hechas de distintas partes del mundo para la realización y análisis geográficos del mismo. Los mapas pueden incluir una dimensión basada en las concentraciones de ciertos químicos y minerales o qué parcelas de tierra son las más adecuadas para el desarrollo. Como también se utiliza en el estudio de geografía, hidrología, tectónica, oceanografía y otros temas estrechamente relacionados(OJEDA, 2000).

Lo que un día constituyó un gran avance a través de la visualización de los mapas en 2D, hoy se ha visto incrementado a través de la visualización 3D. Esta vista ofrece un mayor número de posibilidades, permitiendo, además de visualizar en detalle el terreno, cargar y visualizar los modelos de las edificaciones existentes, simular la luz solar, la atmósfera terrestre y otros efectos destinados a proporcionar una escena realista al usuario final.Es conveniente cuando la visualización de una tercera magnitud, resulta útil para la interpretación de los datos que se quieren mostrar. Proporciona información detallada del entorno construido a diferentes escalas, proveen una buena solución a problemas que no pueden resolverse fuera de un entorno 3D y ayuda a la comprensión de información compleja.

Cuba no queda exenta a este proceso global, por lo cual se han dedicado recursos para alcanzar una mayor aceptación y uso de los SIG por parte de la sociedad cubana. Como ejemplo de esto se encuentrael proyecto AplicativosSIG en el centro de Representación y Análisis de Datos, perteneciente a la Universidad de las Ciencias Informáticas. La misión de este proyecto está dirigida a la producción de software enmarcado en la Geografía, para almacenar información sobre los datos geográficos y la información espacial. Entre los proyectos productivos que se desarrollan en dicho centro se encuentra el proyecto ULTRON, conformado por varios módulos, que tienen como objetivo crear una plataforma basada en tecnologías libres. La documentación de la misma es llevada por un control de versiones y así los usuarios tienen un registro restringido sobre lo que se está desarrollando.

Actualmente ULTRON en su versión 2.0 solo realiza representaciones en 2D, por lo que la misma se encuentra limitada y las posibilidades de visualizar con más detalle el terreno son menores, ya que no permite el estudio de la incidencia del sol según un ángulo de vista determinado, así como

el de zonas de pocas o muchas pendientes. De igual forma no es posible el análisis del nivel del agua en zonas de embalses, aliviaderos y ríos en casos de abundantes precipitaciones.Además, no se puede realizar el análisis completo en los tres planos del espacio, pues hay datos que no son visibles en 2D.

Por esta razón al crear una visualización 3D en la plataforma sería de gran ayuda ya que permite representar productos o elementos físicos de manera precisa para tener una visión global de cómo serán una vez llevados a la práctica. Esta posibilita alcanzar una mayor libertad en cuanto a la visualización de datos globales y espaciales lo que permite ilustrar de forma más definida los rasgos topográficos o variabilidad espacial de los datos, posibilitando al usuario tener una mejor imagen de las cualidades y atributos de los objetos representados. De igual forma, se puede representar productos o elementos físicos de manera precisa, para tener una visualización global de cómo serían una vez llevados a la práctica,ya que al tener esta visualización se puede anticipar de manera integral los procesos ambientales reales y que la representación de los productos sea másatractiva.

A razón de la problemática existente surge el siguiente **problema a resolver**: ¿Cómo realizar análisis del terreno en la plataforma web ULTRON 2.0 a partir de Modelos Digitales del Terreno?

Tomando en cuenta el problema antes propuesto se define como **objeto de estudio**: proceso de representación de los MDT en 3D.

De esta forma se determina como **objetivo general**: desarrollar el módulo de visualización en 3D en la plataforma web ULTRON 2.0.

Para ello se identifica como **campo de acción:**proceso de representación de los MDT en 3D con tecnologías Web.

Para dar cumplimiento al objetivo general, se definen las siguientes **tareas de la investigación**:

- Análisis de los principales conceptos asociados al dominio del problema antes planteado.
- Análisis de los modelos digitales del terreno en 3D sobre la web.
- Análisis de losestándares para la representación en 3D sobre la web.
- Análisis del estado del arte sobre los SIG.
- Caracterización de las tecnologías y las herramientas a utilizar para el desarrollo de la solución.
- Análisis y levantamiento de requisitos para el módulo de visualización 3D.
- Caracterización y selección de los patrones de diseño más factibles para la propuesta de solución.
- Realización del diagrama de clases de diseño, modelo de datos y diseño de casos de pruebas.
- Implementación de la aplicación que demuestre los resultados de la propuesta.
- Realización de las pruebas de calidad de software.

Por lo tanto, se plantea la siguiente **idea a defender**:si se representan Modelos Digitales del Terreno en 3D sobre la plataforma web ULTRON 2.0, se logrará una mejor visualización de los datos espaciales en dicha plataforma.

Para validar metodológicamente la investigación se utilizan los **métodos científicos**, clasificados en teóricos y empíricos de los cuales se emplearon:

Métodos teóricos:

- Analítico-sintético: la utilización de este método teórico posibilitará el análisis de los elementos más importantes que se relacionan con la Visualización geográfica en 3D en la web. Dará la facilidad de obtener y procesar la información de las tendencias actuales relacionada con la utilización de estas técnicas de visualización tridimensional en plataformas web.
- Histórico-lógico: mediante la utilización de este método se puede estudiar la evolución que ha tenido hasta la actualidad todo lo referente a la visualización tridimensional sobre plataforma web.

Métodos empíricos:

- Entrevista:la utilización de este método fue con el objetivo de conocer las principales experiencias de especialistas en la representación de objetos 3D en la web y en particular de la información geográfica.Para la aplicación de este método se seleccionó como población a los especialistas informáticos del proyecto AplicativosSIG y de la Empresa de Tecnologías de la Información para la Defensa(XETID),esto permitió obtener la información necesaria para dar solución al problema planteado.

El presente documento consta de tres capítulos, conclusiones generales, referencias bibliográficas y anexos. Los capítulos se dividen según el nivel de detalle que requiere el contenido abordado en cada uno de ellos, definidos de la siguiente forma:

Capítulo 1: Fundamentación teórica. En este capítulo se establecerán los conceptos fundamentales asociados al dominio del problema, así como los principales elementos que engloban el objeto de estudio que da paso a la presente investigación.De igual forma se realizará un análisis de las tendencias actuales del desarrollo en 3Den la web a través de los estándares existentes.Por otra parte, se realizará el estado del arte sobre los sistemas homólogos al resultado que se persigue con la investigación. Por último, se describirán la metodología que guiará el proceso de desarrollo de la propuesta de solución, así como los lenguajes y herramientas a utilizar en dicho proceso.

Capítulo 2: Descripción y diseño del sistema. En este capítulo se describe el diseño de la solución propuesta, a partir de las disciplinas de la metodología definida para guiar el desarrollo de esta. Entre los contenidos analizados se encuentra el proceso de captura de los requisitos funcionales (RF) y no funcionales (RnF) con los que debe cumplir el módulo propuesto. Además, se describe la arquitectura de la solución, el diagrama de clases del diseño, el modelo de datos y

los patrones de diseño utilizados. Por otra parte, se exponen los estándares de codificación empleados en la disciplina de implementación.

Capítulo 3: Validación. En el capítulo se muestran los resultados obtenidos luego de aplicar las técnicas de validación de requisitos, así como las métricas para la validación del diseño de la solución propuesta. En cada caso se documentan los resultados obtenidos. Por otra parte, se define la estrategia de prueba aplicada a partir de las disciplinas establecidas para la etapa de pruebas por la metodología *AUP* variación UCI.

Capítulo 1: Fundamentación teórica

1.1. Introducción

En este capítulo se establecen los conceptos fundamentales asociados al dominio del problema, así como los principales elementos que engloban el objeto de estudio que da paso a la presente investigación.De igual forma se realiza un análisis de las tendencias actuales del desarrollo tridimensional en la web a través de los estándares existentes.Por otra parte, se realiza el estado del arte sobre los sistemas homólogos al resultado que se persigue con la investigación. Por último, se describe la metodología que guiará el proceso de desarrollo de la propuesta de solución, así como los lenguajes y herramientas a utilizar en dicho proceso.

1.2. Conceptos fundamentales asociados al dominio del problema

Para el desarrollo de la presente investigación se hace necesario realizar un estudio de los conceptos asociados al dominio del problema, con el objetivo de tener un mayor entendimiento de la propuesta de solución que se plantea, para ello a continuación, se establecen los conceptos fundamentales que engloban dicha solución:

Representación en 3D: es la representación de un objeto de visualización en tres dimensiones, que pueden ser el largo, ancho y profundidad de esta, proyectado en una pantalla bidimensional. Los objetos en 3D permiten una movilidad que no se puede realizar en 2D(Babylon, 2009).

SIG: Se puede definir como una tecnología de manejo de información geográfica que integra hardware, software, datos geográficos y espaciales, a un equipo científico y de trabajo que en un conjunto crean lo que puede definirse como un modelo de la parte de la realidad en lo que refiere a las coordenadas terrestres(Sommerville, 2005).

MDT:estructura numérica de datos que representa la distribución espacial de una variable cuantitativa y continua. Por lo tanto, son modelos simbólicos, ya que las relaciones de correspondencia que se establecen con el objeto real tienen la forma de algoritmos o formalismos matemáticos(Aguilar Mugica, 2015).

1.3. Modelos Digitales del Terreno en 3D sobre la Web

El uso de los MDT permite la representación digital de la topografía de un territorio facilitando la identificación de diferentes datos de interés por concepto de elevaciones, tales como cuencas visuales, curvas de nivel, entre otros, mediante la utilización de una especialización de los mismos denominados modelos digitales de elevación. Paragenerar estos modelos es necesario de una muestra del terreno a visualizar.De dicha muestra se extraen los diferentes puntos de interés para cargar en memoria una estructura de datos espaciales a través de algoritmos de triangulación, cada uno de estos identificados en la representación cartográfica.

A lo largo de la investigación se estudian diferentes modelos, métodos matemáticos, algoritmos yherramientas que permiten generar, representar y visualizar estos modelos en 3D, para ayudar a

obtener una propuesta a ser aplicada en el proyecto que permita la representación de superficies del terreno de forma eficiente en la Web.

Para la representación de los MDT es necesario tener en cuenta que todos ellos se relacionan entre sí, y es lo que da lugar a una representación de la parte del terreno que se quiere visualizar. Para crear un modelo de un lugar determinado, se requiere de datos como la iluminación, los principales puntos de altura, la representación del relieve, la vegetación e hidrología. La representación de la vegetación y la cartografía se realizan normalmente sobre papel, sin embargo, la digitalización de estos datos permite un mejor entendimiento de la información (Muñoz Aldana, 2010).

Para llevar a cabo la digitalización del terreno se requiere conocer la información referente a las coordenadas***(x, y)*** en el espacio y tener los datos de la altura de cada una de ellas. A través de estos datos se pueden construir las líneas o curvas de nivel, y de esta forma obtener información sobre el relieve y accidentes naturales (presas, ríos, valles, montañas) (Muñoz Aldana, 2010).

La generación de un Modelo Digital de Elevaciones (MDE)por medio de la cartografía, requiere de la digitalización de las curvas de nivel, además de la validación y depuración del MDE mediante la generación del relieve sombreado, para tener un mejor detalle del relieve y en su caso eliminar inconsistencias de altura. También se realiza la unión de los modelos que conforman el proyecto y se verifica la continuidad y consistencia lógica de los datos mediante una visualización de los mismos (Muñoz Aldana, 2010).

Para el proceso de representación de MDT en 3D es necesario tener presente que el MDE es un factor fundamental, así como la captura de los datos, y saber corregir los errores de captura. Entre los aspectos importantes dentro del MDE están (Muñoz Aldana, 2010):

- identificar y calcular las líneas de flujo y cuencas visuales
- definir los modelos hidrológicos, irradiancia y reflectancia
- realizar análisis de la insolación y sombra.

Para realizar estos cálculos se utilizan métodos de interpolación, yaque permiten generar superficies continuas a partir de medidas en localizaciones puntuales (muestra o puntos muéstrales). Cuando se refiere a la interpolación espacial en los SIG,se identifican una serie de métodos que se clasifican según su naturaleza (Muñoz Aldana, 2010):

- **Deterministas:** generan superficies continuas mediante el grado de similitud o suavizado. Dentro de esta categoría se encuentran los métodos globales, locales,ponderación de distancia inversa (IDW por su nombre en inglés *InverseDistanceWeighting*) y Spline[1].
- **Geoestadísticos:** generan superficies continuas a partir de las propiedades estadísticas de los datos de partida. Dentro de esta categoría se encuentranKriging[2] y Cokriging[3].

[1]No es más que la representación de las curvas a través de polinomios de una determinada forma

[2]Técnica geoestadística que se utiliza con fines de interpolación (mapeo y contorneado).

[3]Técnica geoestadística que se utiliza con fines de interpolación (mapeo y contorneado).

La representación de MDT sobre la Web se realiza utilizando imágenes de todo lo que se quiere visualizar y no solamente cargando aquellas imágenes que se tienen guardadas en ficheros. A través de las estructuras de datos se calculan los valores en el espacio de aquellas variables de importancia para la representación(Muñoz Aldana, 2010).

1.3.1. Modelo Digital de Elevaciones

Un Modelo Digital de Elevaciones (MDE) forma parte del conjunto de MDT. Es una estructura de datos numérica que representa los datos de las elevaciones de un territorio dado, a través de una ecuación en función de *x, y, z*, donde le atribuye a *z* el valor de la altura unido a los valores de *x, y* en las coordenadas. Constituye una capa del Modelo Digital del Terreno que representan las características de la superficie terrestre. A continuación, se muestra laecuación en función de la altura (SIEBER, y otros, 2012):

$$z = f(x, y)$$

En dicha función ***(x, y)*** constituyen las coordenadas de un punto, mientras que ***z***representa la altura.El MDE está dividido en dos modelos de datos que posibilitan la representación: el modelo vectorial yelráster. Los vectoriales están definidos por sus coordenadas que representan puntos o líneas y lasrásterrepresentan los valores que ocupan las variables en la localización espacial del terreno.A continuación, se detallan estos modelos (SIEBER, y otros, 2012):

- Estructura vectorial:
  - Contornos:Utiliza una polilínea que consta de un vector con varios pares de coordenadas***(x,y)*** queconforman la trayectoria de las curvas de nivel, que son representadas mediante una serie de puntos quese sitúan sobre el modelo. Este modelo consta de varias curvas de nivel que pasan por encima dela zona a representar.
  - TIN o red de triángulos irregulares: La representación se realiza utilizando una serie de triángulos irregulares adosados[4] que son construidos usando un plano y seleccionando tres puntos subyacentes y que no sean colineales[5], donde cada triángulo puede representar una parte de la superficie y asícaracterizar el relieve.
- Estructura ráster
  - Matrices regulares: En esta estructura se superpone una retícula[6] sobre el terreno, que toma la forma de una red de mallas cuadrada de donde se extrae una serie de datos para precisar la altura de cada celda. Donde el valor de cada dato está precisado por su localización espacial en la matriz con respecto a las filas y columnas.

[4]Triángulos adyacentes, que se encuentren uno al lado del otro
[5]Cuándo al pasar una recta todos los puntos están dentro de esta, si hay alguno fuera de esta no son colineales
[6]Red de puntos sobre el terreno

1.3.2. Variables derivadas

Para realizar la representación del relieve se utilizan variables derivadas, como son: la pendiente, la orientación y la curvatura. A este proceso se le llama parametrización del relieve, que no es más que el conjunto de medidas que son definidas para la caracterización geométrica del terreno. La pendiente de un punto del terreno está definida por el ángulo del vector normal de la superficie que forma dicho punto con la vertical. En el MDE se utiliza para el cálculo de la pendiente de un plano (Muñoz Aldana, 2010).

Un Modelo Digital de Pendientes se realiza a través del plano y puede representar la información relativa de la distancia y vecindad de los valores de altitud.La orientación en un punto puede definirse como el ángulo existente entre el vector que señala el norte, y la proyección sobre el plano horizontal del vector normal a la superficie en ese punto. Como en el caso de la pendiente, el valor de orientación se estima directamente a partir del MDE (Felicísimo, 1999).

La curvatura en un punto puede definirse como la tasa de cambio en la pendiente, que depende de las derivadas de la altitud, de los cambios de pendiente en el entorno del punto. La curvatura tiene especial interés como variable influyente en fenómenos como la escorrentía superficial, canalización de aludes, erosión y flujos en general (Felicísimo, 1999).

1.3.3. Modelo Digital Multivariable

Es un conjunto de MDT con la misma referenciación geográfica que forman un modelo digital multivariable (MDM), donde cada celda queda definida por un vector de valores ***z*** o vector de características cuyos valores son los de cada MDT del MDM. Un ejemplo de ello seríaque en el primer MDT vendría el modelo de elevaciones y en el segundo las pendientes (Padrón Castillo, 2015).

1.3.4. Cuencas visuales, visualización del relieve y modelos de reflectancia

Las cuencas visuales y los modelos de reflectancia se utilizan para la visualización del relieve fácilmente interpretado. A continuación,se describen en detalle cada uno de estos conceptos.

Cuencas visuales

Las cuencas visuales son un conjunto de información que permite la visibilidad de lugares, que utilizanuna serie de puntos de donde se escogen los que permiten una mejor visualización del terreno en el áreade estudio. Para el análisis de las cuencas visuales se requiere una serie de datos entre ellos, las curvasde nivel, densidad de la vegetación, y datos del relieve. Los puntos son creados sobre curvas de nivel demediano y bajo tamaño, aquellas que son muy grandes dificultarían la visualización de áreas muy bajas enel terreno (Aguilera Fernández, y otros, 2016).

Una cuenca visual de un punto o foco determinado no es más que la conexión que existe entre este punto y el conjunto de puntos de un modelo que están conectados visualmente. Para que

esto suceda debe de existir la intervisibilidad[7] entre el punto o foco y cada uno de los puntos que forman el modelo. Esto no es más que al trazar una recta sobre estos pares de puntos se cumpla que la recta posea una altura superior a la del terreno sobre la proyección del plano sin tener en cuenta el inicio y final de la recta. Luego de obtener el modelo de puntos y de haber realizado el análisis de intervisibilidad entre cada par de puntos se realiza la cartografía, o sea, la representación de estos lugares(Aguilera Fernández, y otros, 2016).

Las cuencas visuales tienen gran importancia para las aplicaciones prácticas de visualización de lugares. Ejemplo de estoes el análisis del impacto visual para representar los efectos negativos sobre el paisaje, y puede sertomado como una base objetiva para tomar decisiones en cuanto a fiabilidad de la incidencia visual(Aguilera Fernández, y otros, 2016).

Visualización del relieve

La representación del relieve en un MDE, es muy importante ya que permite una mejor explicaciónsobre los diferentes fenómenos sobre la superficie terrestre. La representación del relieve tiene quefacilitar una mejor interpretación visual del relieve. La misma está muy relacionada con los patrones de iluminación,donde utilizan valores que de acuerdo con los valores de altitud son representados en una escala degrises, donde el menor valor corresponde al negro y el mayor alblanco. Esto permite una mejorexplicación de la apariencia visual (BARREDO CANO, y otros, 1996).

Modelo de reflectancia

El modelo de reflectancia es considerado un caso particular de la visualización del relieve debido a querealiza una simulación de forma realista de las condiciones reales de la iluminación.El proceso de iluminación es muy importante en este modelo y para esto posee dos propiedades muyimportantes (Aguilar Mugica, 2015):

- La vista cenital: el punto de vista está a una altitud infinita sobre el centro del modelo.
- El valor correspondiente a cada pixel o elemento del modelo se asigna mediante algoritmosque simulan la apariencia real de la superficie frente a condiciones específicas de iluminación (brilloaparente), de donde se deriva la denominación modelo digital batimétrico, que se adopta para estos modelos.

La iluminación depende de algunos factores que inciden sobre él. Entre ellos se encuentran que esdefinida una sola fuente de iluminación, la cual tiene que ser directa con un ángulo de incidencia solar porencima del modelo. En casos muy particulares se define otra fuente de luz hemisférica para simularlaluz difusa que depende de la absorción atmosférica, la incidencia solar y la altitud (Aguilar Mugica, 2015).

[7]Accesibilidad visual de un punto desde el resto de puntos del modelo

1.3.5. Líneas de flujo, cuencas hidrológicas y modelo de caudales máximos

La línea de flujo es el trayecto que sigue un punto inicial, toda la escorrentía[8] superficial sobre el terreno,que puede terminar en el desemboque del mar o al final de un modelo. Este se puede construir siguiendoel sentido de la máxima pendiente hasta llegar a un sumidero[9], a la costa o al borde del modelo.

A través de las líneas de flujo se puede determinar la red hidrológica, el área subsidiaria de una celda y lascuencas hidrológicas. El área subsidiaria de una celda no es más que el conjunto de celdas cuyas líneasde flujo convergen en ella, y una cuenca hidrológica está formada por el área subsidiaria de una celdasingular, que actúa como sumidero (Aguilar Mugica, 2015).

El agua que escurre en un río es captada en un área determinada, por lo general por la conformación delrelieve. A esta área se le llama cuenca hidrológica.

El área subsidiaria puede definirse para cualquier punto del territorio que se está analizando, sin embargo,no todos los puntos pueden ser considerados sumideros de una cuenca hidrológica. Un punto del modeloes considerado un sumidero de la cuenca si es el de menor altura de una concavidad, si se encuentra alborde del modelo y los demás puntos drenen hacia él, o si está situado en la costa con drenaje al mar (Alonso Sarria, 2005).

El trazado de las líneas de flujo puede realizarse de acuerdo con criterios muy simples: partiendo de unpunto del modelo, se construye siguiendo el sentido de máxima pendiente hasta llegar a un sumidero, alborde del modelo o a la costa (Alonso Sarria, 2005).

El proceso de construcción de una línea de flujo a partir de un punto inicial, es iterativo y consta de tres fases elementales (Aguilar Mugica, 2015).

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

El sentido de flujo de un punto de un determinado modelo se mantiene de forma constante, por lo tanto, esposible construir una matriz que almacena un archivo de códigos o CDF[10], que son los posibles sentidosde la escorrentía superficial, que se reduce a 9 valores, el 1 identifica los flujos en dirección norte; del 2 al8, en sentido horario, del noreste al noroeste, y el 9 identifica las

[8]Es la corriente de agua que se vierte al rebasar su depósito o embalses naturales o artificiales.

[9] Es un tipo de lago de origen cárstico circular que actúa como desagüe natural para el agua de lluvia o para corrientes superficiales como ríos o arroyos.

[10]Códigos de flujo de un punto

concavidades y zonas planas, en las queel flujo no es posible o queda indefinido (Ordóñez Galan, y otros, 2003).

Cuencas de drenaje

La identificación y delimitación de las cuencas puede considerarse una generalización de las líneas deflujo. Se crea una matriz del mismo tamaño que el MDE, donde a cada punto le corresponde un númeroindicando la cuenca a la que pertenece. El área de una cuenca se puede determinar por el número depuntos que la conforman por la distancia entre ellos (Novoa Goicochea, y otros, 2012).

Modelo de caudales máximo

La magnitud del área subsidiaria de una celda del MDE está directamente relacionada con el Caudal Máximo Potencial (CMP). El caudal que puede circular en un momento dado en un punto del terreno depende, entre otros factores, de la magnitud del área subsidiaria, de las precipitaciones sobre ella y de la pendiente de la zona, que permite la circulación con menor o mayor rapidez (Muñoz Aldana, 2010).

Cuando se realiza el trazado de las líneas de flujo, se puede calcular la extensión del área subsidiaria de cada punto del modelo. Se realiza una línea de flujo para cada punto del MDE, donde cada elemento del modelo obtiene la cantidad de líneas de flujo que pasan por él, es decir, que al final cada valor se corresponde con su área subsidiaria, que serían los puntos que drenan hacia él, desde las zonas de mayor altura. La mayor parte de los puntos tendrán valores bajos, especialmente los que se encuentran situados en zonas de cumbres y crestas, y en menor medida los constituyentes de laderas. Los flujos convergerán en el fondo de los valles y los valores van a ir aumentando hasta que alcanza el máximo valor en el punto que ejerce como sumidero de la cuenca (Muñoz Aldana, 2010).

Este modelo representa los valores de caudal máximo de cada punto, donde el área subsidiaria es proporcional a este, para alcanzar un equilibrio en el proceso. La situación de equilibrio llega cuando el número de ciclos elementales (paso de un punto a otro del modelo), ha sido suficiente para completar la línea de flujo más larga, y también cuando después de un tiempo de lluvia suficientemente largo, la escorrentía superficial, originada en el punto más alejado del sumidero de la cuenca, ha tenido tiempo de pasar por el mismo (Muñoz Aldana, 2010).

1.4. Estándares para la representación en 3D

Los estándares son acuerdos documentados que contienen especificaciones técnicas u otros criterios precisos para ser usados consistentemente como reglas, guías, o definiciones de características para asegurar que los materiales, productos, procesos y servicios se ajusten a su propósito (Huidobro, 2015).

1.4.1. VRML

El Lenguaje de Modelado de Realidad Virtual (VRML por sus siglas en inglés de *Virtual RealityModelingLanguage)*es un formato de fichero para describir objetos 3D interactivos con efectos de iluminación, texturas y detección de colisiones. Diseñado especialmente para ser usado a través de Internet o un cliente local. Se creó con el objetivo de realizar escenas y modelos vectoriales en 3D con las que el usuario pudiera tener cierta interacción. El estándar VRML fue creado y desarrollado en 1994 por el consorcio VRML y fue la primera tecnología reconocida oficialmente porlaOrganización Internacional de Estandarización14772(ISOpor sus siglas en inglés de *International OrganizationforStandarization*) como estándar para la creación, distribución y representación de elementos 3D a través de Internet. La filosofía de VRML es muy similar a la del Lenguaje de Marcado para Hipertexto (HTML por sus siglas en inglés de *HyperTextMarkupLanguage*). Al ser un estándar abierto no está limitado a ninguna aplicación o plataforma, y existen varios visores gratuitos. Al utilizar un formato de texto plano, no es necesaria ninguna herramienta para crear un mundo VRML. No se limita a la descripción de escenas estáticas, sino que permite un dinamismo en las escenas proporcionándole al observador la interacción con los objetos de la misma. Permite abrir ficheros de formato WRL, WRZ, VRML; solo que para utilizarlo se requiere de un módulo en el navegador. Fue diseñado para cumplir con los siguientes principios básicos (ISO/IEC 14772, 1997):

- Dar la posibilidad de desarrollo de programas para crear, editar y mantener archivos VRML, así como la creación de programas para la importación y exportación del formato VRML a otros formatos gráficos tridimensionales.
- Permitir la opción de utilizar y combinar objetos dinámicos tridimensionales dentro de un mismo mundo VRML.
- Incorporar la capacidad de crear nuevos tipos de objetos no definidos.
- Resaltar la importancia del funcionamiento interactivo en una amplia variedad de plataformas existentes.
- Permitir la creación de mundos tridimensionales de cualquier tamaño.

1.4.2. JAVA3D

El API Java 3D es una interfaz para escribir programas que muestran e interactúan con gráficostridimensionales. Proporciona un conjunto de clases para crear aplicaciones y applets[11] con elementos 3D. De igual forma provee funciones para la creación de imágenes, visualizaciones, animaciones y programas de aplicaciones gráficas 3D interactivas. Les permite a los desarrolladores la posibilidad de manipular geometrías complejas en tres dimensiones. La principal ventaja que presenta este API 3D frente a otros entornos de programación 3D es que permite crear aplicaciones gráficas 3D independientes del tipo de sistema. Es parte de la API

[11]Componente de una aplicación que se ejecuta en el contexto de otro programa, en estos casos un navegador Web

JavaMedia y por tanto puede hacer uso de la versatilidad del lenguaje Java, así como soportar un gran número de formatos como VRML y CAD (Jiménez, y otros, 2006).

Por otra parte, el conjunto de clases, interfaces y librerías de alto nivel que posee permiten aprovechar la aceleración gráfica por hardware que incorporan muchas tarjetas gráficas. Además, las llamadas a los métodos de Java 3D son transformadas en llamadas a funciones de OpenGL[12] o Direct3D[13]. Aunque no necesariamente soporta todas las necesidades 3D, proporciona la capacidad de implementarlo a través de código Java y en otros casos requiere de cargadores (VRML, X3D) que traducen ficheros de ese formato en objetos apropiados en Java3D. Proporciona una interfaz de programación de alto nivel basado en el modelo orientado a objetos,por lo que permite un desarrollo de aplicaciones rápido y simple (Jiménez, y otros, 2006).

1.4.3. X3D

X3D es un estándar abierto XML[14] bajo la norma ISO/IEC 19775/19776/19777, un formato de archivo 3D que permite la creación y transmisión de datos 3D entre distintas aplicaciones, especialmente en red. Sus características son(Jiménez, y otros, 2006):

- Está integrado en XML: esto representa un paso fundamental a la hora de conseguir una correcta integración en:
  - Servicios Web.
  - Redes Distribuidas.
  - Sistemas multiplataforma y transferencia de archivos y datos entre aplicaciones.
- Es Modular (tiene componentes): esto permite la creación de un núcleo 3D más ligero ajustado a las necesidades de los desarrolladores.
- Es Extensible: permite añadir componentes para ampliar las funcionalidades según lasnecesidades del mercado.
- Es Perfilado: se pueden escoger distintos grupos de extensiones apropiadas según las necesidades específicas de la aplicación.
- Es Compatible con VRML: se mantiene el desarrollo, el contenido y la base de VRM.

En vez de mantener una especificación amplia y estática como VRML, tiene una arquitectura basada en componentes que da soporte para la creación de diferentes perfiles, con esto se pretende que el desarrollo en un área no afecte ni retrase la evolución en su conjunto (Jiménez, y otros, 2006).

[12]Es una especificación estándar que define una interfaz de programación de aplicaciones multilenguaje y multiplataforma para escribir aplicaciones que produzcan gráficos 2D y 3D.

[13] Consiste en una interfaz de programación de aplicaciones para la programación de gráficos 3D.

[14] Lenguaje de Marcas Extensible (XML por sus siglas en inglés de *eXtensible Markup Language*) es un meta-lenguaje que permite definir lenguajes de marcas desarrollado por el *World Wide Web Consortium* utilizado para almacenar datos en forma legible.

Se decide utilizar el estándar X3D debido a que el mismo está integrado en XML lo cual permite una correcta integración a servicios Web además de que es una tecnología libre. Una de sus principales ventajas es la capacidad de ejecución en distintas plataformas y evitar la necesidad deinstalar un *plugin* específico para el navegador. X3D ofrece soporte para múltiples codificaciones dearchivos: VRML97, XML. También emplea una arquitectura modular para dar una mayor extensibilidad y flexibilidad. Además, permite una mayor flexibilidad del ciclo de vida del estándar, ajustándose a la evolución del mismo.

1.5. Sistemas homólogos

En el mundo se han creado programas y herramientas, que posibilitan la representación de MDT.A continuación, se abordarán variasde las soluciones existentes en algunos países y se hará un análisis de cada uno sobre la base de algunas características como:

- Sistemas operativos que soporta
- Estructuras de datos que soportan
- Visualización con más detalle el terreno
- Si es Libre, de código abierto.

1.5.1. Sistema de apoyo de análisis de recursos geográficos

GRASS GIS por su nombre en inglés (*GeographicResourcesAnalysisSupportSystem*) es un software de SIG totalmente gratuito compatible con los sistemas operativosUNIX, Linux, Solaris y Windows. Está basado en licencia libre, que soporta las estructuras de datos ráster y vectorial. Estos datos se almacenan enuna base de datos donde está toda la información de GRASS. Estabase de datos tiene los proyectos organizados por áreas de subdirectorios que son llamadas *location*. Un *location*es definido por un sistemacoordenado, una proyección cartográfica y unas fronteras geográficas. Lossubdirectorios y archivos quedefinen un *location*son creados automáticamente cuando se inicia la primera vez con un nuevo*location*. Este software permite almacenar mapas relacionados con los proyectos que son guardados en la BD[15] y puede importar capas, ya sea ráster o vectorial, o cargar mapas (Muñoz Aldana, 2010).

1.5.2. MAPINFO PRO

Es unaherramienta de Sistemas de Información Geográfica que permite realizar diversos y complejos análisis geográficos ideales para facilitar la toma de decisiones: captura, consulta, edición, análisis y reportes de información geográfica dinámicamente relacionada con bases de datos. Es compatible con Microsoft Windows 10y Office, GNU/Linux, no es compatible con Unix. Es un software propietario, que la institución misma provee las licencias(Bravo Batista, 2012).

[15] Base de datos (BD) es un conjunto de datos pertenecientes a un mismo contexto y almacenados sistemáticamente para su posterior uso.

Esta herramienta proporciona la creación de mapas por ordenador, donde se puede visualizar los datos como puntos, regiones zonificadas temáticamente, gráficos, entre otros. Puede llevar a cabo operaciones de zonificación, combinación y división de objetos, y definición de áreas de influencia. Es de interés conocer el trabajo con los mapas en MapInfo, porque refiere al estudio de la generación de las vistas de los mapas (Bravo Batista, 2012).

MapInfo tiene entre sus característicasque soporta las estructuras de datos ráster y vectorial, que permite obtener numerosas vistas de datos en tres formatos: ventanas de mapas, listado y gráfico. Las ventanas de mapas permiten que se puedan crear mapas o editar mapas existentes, para ello se debe de ir al menú ARCHIVO, haciendo clic en ABRIR o seleccionado nueva ventana en el menú ventana. Los mapas en MapInfo están formados por capas de objetos, a partir de cinco tipos básicos de objetos(Bravo Batista, 2012):

- Regiones: objetos cerrados que cubren un área concreta. Éstos incluyen polígonos, elipses y rectángulos.
- Objetos de puntos: representan ubicaciones simples de datos.
- Objetos de línea: objetos abiertos que cubren una distancia determinada.
- Objetos de texto: texto que describe un mapa u otro objeto, como etiquetas y títulos.
- Objetos de recopilación: combinación de regiones, objetos de líneas y de multipuntos.

1.5.3. Quantum Gis (QGis)

Es unSistemadeInformaciónGeográficalibre, compatible con los sistemas operativosMac,Linux o Windows, diseñadoparadarsoluciónatodaslas necesidadesrelacionadasconelmanejodeinformación geográfica, caracterizada porseruna solucióncompleta.Es capaz de acceder a los formatos más comunes, como vectoriales, rasters, locales y remotos.Integra estándares deConsorcio Geoespacial Abierto (OGC por su sigla en inglés *Open GeospatialConsortium*), y cuenta con un amplio número de herramientas para trabajar con información de naturaleza geográfica (consulta, creación de mapas, geoprocesamiento, redes, etc.) que lo convierten en una herramienta ideal para usuarios que trabajen con la componente territorial. Su lenguajedeprogramaciónesJava.SedistribuyebajolalicenciaGNU/GPL, lo que permite su libre uso, distribución, estudio y mejora.

En QGis se puedenencontrar un amplio abanico de funcionalidades, es de interés conocer cómoestepermiterealizarlosprocesosdegeneraciónde las vistasde los mapas,apartirdeuna descripciónconcretadelagestióndelosmapasquerealizaelsistema. Primeramente,seiniciala aplicacióndondeapareceráunainterfazGestordeproyectos,quepordefectopermiterealizarlas operaciones de crear mapas, vistas y tablas.

Conelpropósitodecrearunavistadelmapa,seseleccionalaopción (Mapa), luegoseintroduceel nombre del mapa a crear, a partir de la opción que proporciona el sistema en la interfaz gestora de proyecto; seguido se genera en un listado el nuevo mapa creado y se visualiza en esta interfaz.

Con lasherramientasqueproporcionaelsistemasepuedeinsertarenelmapa,unavistaquecontiene atributos como capas, formatos, sistemas de referencias, simbología y etiquetado que refiere al color y las letras.

Además, sepuedeincorporaralmapapropiedadescomolíneas,puntos,polígonos, leyenda, escala y texto, de esta forma se genera una vista del mapa. El sistema permite, además, que se puedan concebir otras vistas a partir del procedimiento descrito; eliminar, visualizar y modificar una vista seleccionada(Mendez, 2011).

1.5.4. ArcGIS Desktop

Es un complejo Sistema de Información Geográfica para compilar, usar y administrar la información geográfica. Incluye un conjunto de aplicaciones: ArcMap, ArcCatalog, ArcGlobe, ArcScene, ArcToolbox yModelBuilder,queadmitendiversastareasSIG,incluidaslarepresentacióncartográfica,la compilacióndedatos,elanálisis,laadministracióndegeodatabasesyelusocompartidode información geográfica. Es compatible con diversos sistemas operativos como Windows, GNU/Linux y Unix.EllenguajedeprogramaciónesJava.Sepuedeusarencualquieradelostresnivelesde producto ArcView, ArcEditor y ArcInfo. ArcView se distribuye bajo una licencia de uso individual, por lo quesepuedeinstalar yusarcadacopiadeArcViewsolamenteenunamáquina,perotambiénse puede obtener una licencia flotante, al igual que ArcEditor y ArcInfo. Esta licencia flotante ofrece a los usuariosdeArcGISDesktopunagranflexibilidad,yaquepermiteinstalarelsoftwareentodaslas computadoras que sequiera. Lasactividadesclavesquerealizanlosusuariosen ArcGISDesktopsoneltrabajoconlosmapas, análisisespacialycompilacióndedatosque soporta las estructuras de datos ráster, vectorial y remotos.

Losmapassonesenciales,porquehacenquetodala informacióncobrevidaysonelmecanismoutilizadoparaeditaryproporcionaranálisisespaciala muchos usuarios. Es de interés conocer sobre el trabajo con los mapas, porque refiere al estudio de la generación de las vistas de los mapas. Cada mapa en ArcGIS Desktop es una especificación o diseño de cómo se utiliza, consulta, muestra, administra y analiza la información geográfica y las herramientas. Los mapas SIG se pueden capturar, guardarycompartircomodocumentosdemapas.Estosdocumentosypaquetessepuedenpublicar como servicio web, si se utiliza ArcGIS Desktop junto con ArcGIS Server, se puede convertir cualquier mapa, geodatabase o modelo en un servicio web de SIG para compartirlo en un grupo de trabajo, en todalaempresaoabiertamenteenInternet.

Además,sepuedencrearyguardar paquetesdemapasypaquetesdecapasquereúnentodoslosaspectosdeSIGnecesariosparala visualización y utilización de mapas, análisis, compilación de datos y administración. También,

brinda la posibilidadde compartir paquetes de mapas o paquetes de capas con otros usuarios (Bravo Batista, 2012).

1.5.5. GENESIG:

Es una plataforma creada en el Centro de Desarrollo de Geoinformática y Señales Digitales (GEySED) de la Universidad de las Ciencias Informáticas. Tiene como principal objetivo realizar la representación geoespacial de la información asociada a negocios específicos, permitiendo además realizar análisis sobre dicha información. Actualmente tiene como necesidad primaria el desarrollo de un módulo que permita el álgebra de mapas y análisis hidrológico para el cálculo entre los mapas y la modelación de los diferentes procesos hidrológicos respectivamente (Labañino, 2011).Además, la plataforma GeneSIG brinda un módulo de análisis el cual posee funcionalidades que permiten obtener detalles del procesamiento directo de la información geográfica sobre el mapa, como cálculos de distancia, superficies, azimuts, perfiles de altura y cálculos de ruta.

Surge como necesidad que tiene la UCI de contar con un producto soberano para el desarrollo de SIG, utilizando tecnologías libresque soporta las estructuras de datos ráster y vectorial. GeneSIG se crea utilizando las ventajas del frameworkcartoweb permitiendo que sea altamente modular y escalable, lo que posibilita que se puedan ir añadiendo y desarrollando nuevos *plugins* en respuesta a necesidades específicas. Posee una interfaz sencilla y de fácil manejo facilitando su uso a usuarios no especializados en sistemas de información geográficos. Tiene una agilizada interacción con la base de datos en cuanto a actualización y acceso, reduciendo el tiempo de respuesta a las peticiones por parte del usuario. Debido que es desarrollado en lapropia universidad, se puede contar con un grupo de apoyo y soporte en caso de falla en alguna funcionalidad o conexión con la base de datos (Rodríguez, 2012).

Resultado del análisis realizado

Una vez realizado el estado del arte sobre los SIG, en la siguiente tabla se muestra un análisis comparativo teniendo en cuenta las características establecidas previamente:

Tabla 1: comparación de las soluciones estudiadas. Fuente: elaboración propia

Características	GRASS GIS	MAPINFO	QGis	ArcGIS Desktop	GENESIG
Sistemas operativos que soporta	UNIX, Linux, Solaris y Windows.	Windows	Mac, Linux o Windows	Windows, Linux y Unix	Windows,Linux
Estructuras de datos que soportan	Ráster y vectorial	Ráster y vectorial	Vectoriales, rasters,	Ráster, vectorial y	Ráster y vectorial

			locales y remotos	remotos	
Visualización con más detalle el terreno	Si	Si	Si	Si	Si
Libre, de código abierto	Si	No	Si	Si	Si

Partiendo del análisis realizado de cada uno de los sistemas homólogos, y teniendo en cuenta las características principales definidas para dar solución al problema existente, se evidencia la necesidad de desarrollar un módulo de visualización 3D que mejore la plataforma ULTRON 2.0 debido a que los sistemas analizados en su mayoría son privativos y no posibilitan acceder a la representación en 3D mientras que otros proporcionan diversos elementos a tener en cuenta para el desarrollo de un módulo de visualización 3D basado en tecnologías libres y así lograr mostrar con más precisión el terreno.

1.6. Metodología de desarrollo de software

Una metodología de desarrollo en ingeniería de software, es un conjunto de herramientas, técnicas, procedimientos y soporte documental encaminados a estructurar, planificar y controlar el proceso de desarrollo de forma organizada y lógica. Ya que tienen como objetivo apoyar a los desarrolladores en la creación de un nuevo software(Kaisler, 2005).

En la presente investigación, para organizar el proceso de desarrollo de software se aplica la metodología Proceso Unificado Ágil *(*AUP por sus siglas en inglés de *Agile UnifiedProcess)* en su variante para la UCI. El uso de esta metodología se fundamenta, basándose en que la propuesta de solución forma parte de uno de los proyectos de la red de centros productivos de la universidad.

1.6.1. Proceso Unificado Ágil variación para la UCI

AUP es una versión simplificada de la metodología Proceso Racional Unificado (RUP por sus siglas en inglés de *RationalUnifiedProcess*). Este define un flujo de trabajo con disciplinas diferentes a las de RUP, aunque conserva sus fases en cada una. La disciplina de modelación incluye la modelación del negocio, requisitos, análisis y diseño. Por otra parte, se integran además la Gestión de Cambios y Gestión de Configuración en una sola disciplina.AUP-UCI se crea porque no existía una metodología de software universal, ya que toda metodología debe ser adaptada a las características de cada proyecto exigiéndose así que el proceso sea configurable, por lo que se decide hacer una variación de la metodología AUP, de forma tal que se adapte al ciclo de vida definido para la actividad productiva de la UCI(Rodríguez Sánchez, 2015).

Entre las principales diferencias presentes en esta versión se encuentran:

- AUP define 4 fases de desarrollo (Inicio, Elaboración, Construcción, Transición), y su versión para la UCI mantiene la fase de Inicio, pero se modifica el objetivo, agrupa las otras 3 fases en una llamada Ejecución e incorpora una llamada Cierre.
- AUP propone 7 disciplinas (Modelo, Implementación, Prueba, Despliegue, Gestión de configuración, Gestión de proyecto y Entorno), la metodología definida en la UCI tiene 8 disciplinas (Modelado de negocio, Requisitos, Análisis y diseño,Implementación,Pruebas Internas, Pruebasde Liberación, Pruebas Aceptación y Despliegue).
- Los flujos de trabajos: Modelado de negocio, Requisitos y Análisis y diseño en AUP están unidos en la disciplina Modelo, en la variación para la UCI se consideran cada uno de ellos disciplinas. Específicamente en la disciplina Requisitos plantean 4 posibles escenarios para la identificación y descripción de requisitos. Se mantiene la disciplina Implementación, en el caso de Prueba se desagrega en 3 disciplinas: Pruebas Internas, de Liberación y Aceptación y la disciplina Despliegue se considera opcional (Rodríguez Sánchez, 2015).

En esta variante se definen 4 escenarios para la disciplina de Requisitos:

- Proyectos que modelen el negocio conCasos de Uso del Negocio (CUN) solo pueden modelar el sistema conCasos de Uso del Sistema (CUS).
- Proyectos que modelen el negocio con Modelo Conceptual (MC) solo pueden modelar el sistema conCasos de Uso del Sistema. (CUS)
- Proyectos que modelen el negocio con Descripción de Proceso de Negocio (DPN) solo pueden modelar el sistema con Descripción de Requisitos por Proceso (DRP).
- Proyectos que no modelen negocio solo pueden modelar el sistema con Historias de usuario (HU) (Rodríguez Sánchez, 2015).

Como se tiene un negocio bien definido de lo que se va a desarrollar se escoge el escenario cuatro en el que se modela el sistema con Historias de Usuario (HU), ya que el cliente es el jefe del centroal que pertenece la presente investigación y al estar al tanto de los requisitos facilita la implementación, prueba y validación de los mismos. Además, no es un proyecto muy extenso lo que facilita que las HU no contengan tanto contenido informativo.

1.7. Ambiente de desarrollo

Contar con un ambiente de desarrollo apropiado que se adapte a sus necesidades resulta imprescindible para las empresas productoras de software, sean estos para consumo interno, o para la venta y distribución de los mismos. Además, resulta de vital importancia emplear metodologías que le garanticen a la empresa un seguimiento efectivo y preciso de los requerimientos del software a desarrollar, los procesos de desarrollo y el producto final.

El equipo de arquitecturade la plataforma ULTRON tiene establecido el siguiente ambiente de desarrollo, por ende, es el utilizado en la propuesta de solución:

- Como herramienta de modelado Visual Paradigm

- Como servidor de base de datos PostgreSQL con su extensión postGISpara el manejo de datos espaciales
- Como marcos de trabajo están Bootstrap, Angular, Node.js yLoopback
- Como servidor de mapasMapserver

A continuación,se detallan las características de cada una de las herramientas a utilizar.

1.7.1. Herramienta de modelado

Las herramientas deIngeniería de Software Asistida por Computadora (CASE por su nombre en inglés*ComputerAided Software Engineering*) son diversas aplicaciones informáticas destinadas a aumentar la productividad en el desarrollo de software reduciendo el costo de las mismas en términos de tiempo y dinero. Estas herramientas, pueden ayudar en todos los aspectos del ciclo de vida de desarrollo del software, en tareas como el proceso de realizar un diseño del proyecto (Flores, 2013).

Como herramienta CASE se utiliza Visual Paradigmen su versión 15.1 que es la establecida por el centro. Esta es una herramienta deLenguaje Unificado de Modelado (UML por sus siglas en inglés *UnifiedModelingLanguage*)que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. Ayuda a una rápida construcción de aplicaciones con calidad y a un menor costo. Permite crear todos los tipos de diagramas de clases, ingeniería inversa, generar código desde diagramas y generar documentación (Paradigm, 2017).Entre sus principales características se encuentran:

- Es una herramienta de diagrama basado en arrastrar y soltar.
- Permite la alineación automática para dibujos limpios y nítidos.
- Admite una amplia variedad de tipos de diagramas que cubren la mayoría de los dominios comerciales que van desde el desarrollo de software, la planificación estratégica, las mejoras comerciales, hasta la gestión de proyectos, la ingeniería de redes y el diseño de arquitectura de tecnología de la información (TI) basada en la nube.

1.7.2. Herramienta de base de datos

Las herramientas de gestión de bases de datos proporcionan una interfazque permite administrar las bases de datos. Estas herramientas también permiten ejecutar consultas SQL desde este interfaz de usuario(Martínez, 2010).Para la gestión de bases de datos en la propuesta de solución se utiliza la herramienta establecida por el centro PostgreSQL en su versión 11.0, con su extensión PostGis v3.0.

PostgreSQLes un sistema de gestión de bases de datos objeto-relacional, distribuido bajo licencia de Distribución de software de Berkeley (BSD por su nombre en inglés *Berkeley Software Distribution*) y con su código fuente disponible libremente. PostgreSQL utiliza un modelo

cliente/servidor y usa multiprocesos en vez de multihilos[16] para garantizar la estabilidad del sistema. Un fallo en uno de los procesos no afectará el resto y el sistema continuará funcionando. Su desarrollo comenzó hace más de 16 años, y durante este tiempo, estabilidad, potencia, robustez, facilidad de administración e implementación de estándares han sido las características que más se han tenido en cuenta durante su desarrollo. PostgreSQL funciona muy bien con grandes cantidades de datos y una alta concurrencia de usuarios accediendo a la vez al sistema (Martínez, 2010).

Entre sus principales características se encuentran:

- Alta concurrencia: mediante un sistema denominado MVCC (acceso concurrente multiversión, por sus siglas en inglés) PostgreSQL permite que mientras un proceso escribe en una tabla, otros accedan a la misma tabla sin necesidad de bloqueos.
- Arrays: los usuarios pueden crear sus propios tipos de datos los cuales pueden ser completamente indexables gracias a la infraestructura de PostgreSQL

Por su parte, PostGIS es un módulo que añade soporte de objetos geográficos a la base de datos objeto-relacional PostgreSQL, convirtiéndola en una base de datos espacial para su utilización en el desarrollo de los SIG. Convierte al sistema de administración de bases de datos PostgreSQL en una base de datos espacial mediante la adición de tres características: tipos de datos espaciales, índices espaciales y funciones que operan sobre ellos (OSGeo Live, 2012).

1.7.3. Marcos de trabajo

En el desarrollo de software, un entorno de trabajo es una estructura conceptual y tecnológica de asistencia definida, normalmente, con artefactos o módulos concretos de software, que puede servir de base para la organización y desarrollo de software. Típicamente, puede incluir soporte de programas, bibliotecas, y un lenguaje interpretado, entre otras herramientas, para así ayudar a desarrollar y unir los diferentes componentes de un proyecto (La Red Martínez, y otros, 2012). A continuación, se describen los marcos de trabajo que se utilizan para el desarrollo de la propuesta de solución.

Bootstrap: mediante la combinación dehojas de estilo en cascada (CSS por su nombre en inglés *Cascading Style Sheets*) y JavaScript, esta multiplataforma, simplifica el proceso de creación de diseños web. Es un marco de trabajoque posee numerosos componentes y permite un ahorro significativo de esfuerzo y tiempo, debido a que facilita la creación de interfaces con diseño web adaptativo o responsivo[17]. Ofrece un conjunto de plantillas CSS y ficheros JavaScript que permiten la integración del marco de trabajode forma sencilla con las aplicaciones web. Por otra parte, es capaz de integrarse con las principales librerías JavaScript, por ejemplo, jQuery (Ledesma, 2008).

[16]En programación, nos estamos refiriendo a los lenguajes de programación que permiten la ejecución de varias tareas en forma simultánea.

[17] Es una filosofía de diseño y desarrollo cuyo objetivo es adaptar la apariencia de las páginas web al dispositivo que se esté utilizando para visitarlas

Su uso en el desarrollo del módulo simplifica la creación de las interfaces de usuario, lo que permite crear con facilidad un ambiente acogedor y agradable a la vista del usuario. La versión a utilizar es la 4.3.

Angular: es de código abierto desarrollado por Google para facilitar la creación y programación de aplicaciones web de una sola página (SPA por su nombre en inglés Single Page Application). Separa completamente el frontend[18] y el backend[19] en la aplicación, evita escribir código repetitivo y mantiene todo más ordenado gracias a su patrón Modelo Vista Controlador (MVC) asegurando los desarrollos con rapidez, a la vez que posibilita modificaciones y actualizaciones. Angular es modular y escalable adaptándose a las necesidades y al estar basado en el estándar de componentes web, y con un conjunto de interfaz de programación de aplicaciones (API por su nombre en inglés *ApplicationProgrammingInterface*) permite crear nuevas etiquetas HTML personalizadas que pueden reutilizarse(QualityDevs S.L, 2020). La versión que se va a utilizar para la propuesta de solución es la 8.

Node.js: es un intérprete de JavaScript, construido sobre el motor JavaScript v8 de Chrome.Posee un entorno de ejecución multiplataforma de código abierto para el desarrollo del lado del servidor y las aplicaciones de red escalables. Contiene un entorno de programación dirigido por eventos, basado en el lenguaje de programación ECMAScript(Nodejs, 2015). La versión a utilizar esla 10.15, la cual presenta las siguientes ventajas:

- Está basado en eventos, así que toda la filosofía asíncrona utilizada con AJAX en el cliente se puede pasar al servidor.
- Permite utilizar el mismo lenguaje (JavaScript) tanto en el cliente como en el servidor.
- El desarrollo es más rápido.
- La ejecución de pruebas de unidad se puede hacer más rápido.

Loopback: es una interfaz de red virtual que señala que las direcciones de rango 127.0.0.0 son direcciones de Loopback.Son redefinidas en los dispositivos incluso en las direcciones de protocolo de internet (IP por su nombre en inglés *Internet Protocol*) públicas, por ejemplo, los routers[20] realizan este tipo de actividades siempre. Estas direcciones se utilizan cuando una transmisión de datos tiene como destino el mismo host[21]. Se utiliza también en tareas de conectividad y para revisar la validez del protocolo de comunicación. La dirección de Loopback es una dirección especial que los hosts utilizan para dirigir el tráfico hacia ellos mismos. Además, crea un método de acceso directo para las aplicaciones y servicios de Protocolo de Control de Transmisión/Protocolo de Internet (TCP/IP) que se ejecutan en el mismo dispositivo para

[18]Es la parte del software que interactúa con los usuarios

[19] Es la parte que procesa la entrada desde el *frontend*

[20]Es un producto de hardware que permite interconectar computadoras que funcionan en el marco de una red.

[21] Se usa en informática para referirse a las computadoras u otros dispositivos (tabletas, móviles, portátiles) conectados a una red que proveen y utilizan servicios de ella.

comunicarse entre sí. La versión que se va a utilizar para el desarrollo de la propuesta de solución es la 4.0.

1.7.4. Entorno de desarrollo

Un entorno de desarrollo integrado o entorno de desarrollo interactivo(IDE por su nombre en inglés *IntegratedDevelopmentEnvironment*), es una aplicación informática que proporciona servicios integrales para facilitarle al desarrollador o programador el desarrollo de software (Kimmig, y otros, 2011).

MapServer: es una plataforma de código abierto para la publicación de datos espaciales y aplicaciones cartográficas interactivas para la web,también posibilita la creación de Sistemas de Información Geográfica con el fin de visualizar, consultar y analizar información geográfica a través de la red mediante la tecnología deservidor de cartografía digital (IMS por su nombre en inglés InternetMapServer).Se ejecuta bajo plataformas Linux/Apache y Windows, soporta formatos vectoriales como ESRIshapefiles,PostGIS,ESRI ArcSDE,GML y formatos ráster como JPG,PNG,GIF,TIFF/GeoTIFF,EPPL7.Otracaracterística fundamental que presentaMapScriptes la de proporcionar una API para poder acceder a las funcionalidades de MapServer mediante lenguajes de programación como PHP,Java,Perl,Python,Ruby o C#(MapServer, 2011).

1.8. Conclusiones parciales

En este capítulo se realizó la fundamentación teórica en la que se sustenta la presente investigación, donde:

- El estudio a detalle de los conceptos asociados permitió tener un mayor conocimiento sobre los principales términos asociados al dominio del problema.
- El análisis sobre los MDT realizado favoreció alcanzar un mayor entendimiento de la solución propuesta.
- El estudio de los estándares de representación en 3D permitió la selección del estándar X3D que se corresponde con las características del módulo propuesto al problema planteado.
- El análisis de los diferentes sistemas homólogos evidenció la necesidad de desarrollar un módulo de visualización 3D que mejore la plataforma ULTRON 2.0.
- La caracterización de la metodología, así como de las herramientas a utilizarfundamentó la selección de estas, debido a que están alineadas con las seleccionadas por el equipo de arquitectura del proyecto al que pertenece la propuesta de solución.

Capítulo 2: Descripción de la propuesta de solución

2.1. Introducción

En la actualidad la construcción de un software se sustenta sobre la base de la aplicación de una metodología de desarrollo de software que permita describir, agilizar y comprender este proceso.El presente capítulo sigue los principios del planteamiento anterior, en el cual se enuncian las técnicas para la obtención delos requisitos, sus descripciones, así como las Historias de usuario para el encapsulamiento de los requisitos funcionales.Además, se describe la arquitectura a utilizar en la presente solución y el diagrama de clases del diseño. De igual forma se explica en qué consiste el modelo de datos y se establecenlos patrones de diseños a utilizar. Por último, se exponen los estándares de codificación a empleardurante implementación del módulo propuesto.

2.2. Requisitos de software

Los requerimientos o requisitos de un sistema describen los servicios que ha de ofrecer este y las restricciones asociadas a su funcionamiento(Padrón Castillo, 2015). Para una mejor comprensión del sistema se establecen dos tipos de requisitos fundamentales: funcionales y no funcionales.Para la obtención de estos requisitos se utilizaron lastécnicasque semuestran a continuación.

2.2.1. Técnicas de obtención de requisitos

La determinación de los requisitos de un sistema se encuentra estrechamente vinculada a la aplicación de diversas técnicas de obtención de requisitos, las cuales permiten, a partir de la comprensión de las necesidades del usuario, definir lo que el sistema debe hacer y cómo debe funcionar(Padrón Castillo, 2015).

Enmarcados en el ámbito del problema identificado se empleó la entrevista como técnica de obtención de requisitos, la cualfue de gran utilidad para adquirir información a través del intercambio directo entre distintos usuarios de la Universidad, así como los integrantes del proyecto de la plataforma ULTRON 2.0.El uso de esta técnica permitió captar lasdistintas opiniones e ideas que permitieron comprender el proceder para establecer una solución que sea viable.

Además, serealizó un estudio de la documentación durante el análisis de los sistemas homólogos, teniendo como referencia cada uno de los documentos que describíanestos sistemas. Con esta técnica se logró identificar algunas funcionalidades que pueden ser de utilidad para el desarrollo de la presente solución. Entre estas funcionalidadesse pueden señalar la posibilidad de mostrar capas vectoriales 3D, mostrar las etiquetas del mapa y mostrar cuadros delimitadores.

Una vez aplicadas ambas técnicas para la captura de requisitos que responden a la propuesta de solución, se detallan a continuación los requisitos como resultado de las mismas.

2.2.2. Requisitos funcionales

Expresan la naturaleza del funcionamiento del sistema, es decir, cómo interacciona el sistema con su entorno y cuáles van a ser su estado y funcionamiento. Son ejemplos de requerimientos funcionales los cálculos, detalles técnicos, manipulación de datos y otras funcionalidades específicas que se supone, deban cumplirse en un sistema(Padrón Castillo, 2015).A continuación, se muestran los requisitos funcionales identificados para la solución.

RF.1:Visualizar mapa en 3D:

Esta funcionalidad permite visualizar el modelo digital del terreno de un área seleccionada del mapa en 3D.

RF.2: Calcular distancia en 3D:

Esta funcionalidad permite calcular la distancia en 3D del modelo digital del terreno de un área seleccionada del mapa en 3D.

RF.3:Configurar el terreno:

Esta funcionalidadpermite configurar los diferentes escenarios de la visualización 3D.

RF.4:Mostrar capas vectoriales 3D:

Esta funcionalidad permite mostrar una capa vectorial con valores de elevación en la vista de mapa 3D.

RF.5: Mostrar etiquetas:

Esta funcionalidad permite activa / desactiva las etiquetas del mapa.

RF.6: Mostrar información de mosaico del mapa:

Esta funcionalidad permite incluir números de borde y mosaico para los mosaicos de terreno útil para solucionar problemas de terreno.

RF.7: Mostrar cuadros delimitadores:

Esta funcionalidad permite mostrar cuadros delimitadores 3D de los mosaicos del terreno útil para solucionar problemas del terreno.

2.2.3. Requisitos no funcionales

Toma en cuenta las restricciones sobre el espacio de posibles soluciones.Estos requisitos representan aquellos atributos que debe exhibir el sistema pero que no constituyen una funcionalidad específica. Por ejemplo, requisitos de usabilidad, fiabilidad, eficiencia, portabilidad(Jacobson, y otros, 2000). A continuación, se muestra la descripción de los requisitos no funcionales del módulo de visualización en 3D de la plataforma web ULTRON 2.0:

Usabilidad:

- Debido a las acciones que se realizarán en la aplicación, los usuarios finales deberán tener conocimientos básicos de Informática.
- La solución que se propone constituye una aplicación Web que permitirá el análisis y representación sobre un mapa de los elementos que se manejan, así como otras funcionalidades básicas de todo Sistema de Información Geográfica.

- Las funcionalidades principales del sistema estarán orientadas a íconospara un mayorreconocimiento por parte del usuario.
- Los servidores de Base de Datos y de Mapas deberán tener lassiguientes características:

  Hardware:

  - Procesador: 3Ghz como mínimo.
  - Memoria RAM: 4Gb como mínimo.
  - Disco Duro: 1Tb.
  - Tarjeta de red.

  Software:

  - Sistema Operativo: GNU/Linux Debian
  - NodeJs 10.15 o superior como manejador de dependencia para JavaScript
  - PostgreSQL 9.6 como Sistema Gestor de Base de Datos.
  - PostGIS como extensión de PostgreSQL como soporte de datos espaciales.
  - MapServer 6.4.1 como servidor de mapas.
- Las estaciones clientes deberán cumplir con los siguientes requisitos:

  Hardware:

  - Procesador: 512 MHz como mínimo.
  - Memoria RAM: 512 Mb como mínimo.
  - Disco Duro: 40 Gb como mínimo.
  - Tarjeta de red.

  Software:

  - Un navegador web como Chrome v79.0, Mozilla Firefox v79.0, Safari v13.1.1 que cumpla con los estándares W3C y cuente con soporte para CCS3 y HTML5.

Confiabilidad:

- La información manejada por el sistema estará protegida de acceso no autorizado. El sistema debe solicitar usuario y contraseña para acceder a las funcionalidades de gestión. Cada usuario tendrá asignado un nivel de permisos.
- Si se interrumpe la energía en el servidor, una vez reanudada, el sistema deberá ser capaz de iniciar de forma automática los servicios.
- Si se interrumpe la energía en el cliente, una vez reanudadas las operaciones, el usuario deberá autenticarse nuevamente para acceder al sistema.
- El sistema debe hacer copias de respaldo periódicamente y hacer uso de las mismas una vez ocurrido un fallo en la base de datos.

Restricciones de diseño:

- Para la modelación se utilizará como lenguaje de modelado UML 2.1 y como herramienta el Visual Paradigmfor UML 8.0.
- El software debe diseñarse sobre una arquitectura cliente-servidor.

- Se deben emplear los estándares establecidos para diseño de interfaces, base de datos y codificación.
- Lograr un producto altamente configurable y extensible. La aplicación estará diseñada para su correcta visualización en distintos tipos de dispositivos (Móvil, Tablet, PC).

Interfaz:

- Lainterfazdeusuariodeberáteneraparienciaprofesionalydiseñográficosencillo.
- Lasfuncionalidadesarealizardebenrepresentarseconíconosintuitivos.
- Elsistemadebebrindarlaposibilidaddeocultarlospaneleslateralesparavisualizarelmapaentodalaamplituddelapantalla.

2.2.4. Historia de usuarios

Las Historias de Usuario (HU) son una forma rápida de administrar los requisitos de los usuarios sin tener que elaborar gran cantidad de documentos formales y sin requerir de mucho tiempo para administrarlos. Las historias de usuario permiten responder rápidamente a los requisitos cambiantes, por lo que una historia de usuario puede tener varios cambios a lo largo de un desarrollo sin afectarse el tiempo (Rouse, 2019).

Para el desarrollo del componente propuesto se definieron un total de 7 HU, una por cada RF. A continuación(Ver Tabla 2 y 3), se describen las HU correspondientes al RF1: Visualizar mapa en 3D y el RF2: Calcular distancia en 3D, teniendo en cuenta que estos son de los RF bases para el desarrollo del módulo y de mayor prioridad para el cliente. El resto de las HU se pueden consultar enel documento "*GEYSED Descripción de Requisitos de Software*", elaborado por los autores de la presente investigación.

Tabla 2: HU Visualizar mapa en 3D. Fuente: elaboración propia

Historia de usuario	
Número: 2	**Requisito:** Visualizar mapa en 3D
Programador: Grabiel González Lorenzo	**Iteración Asignada:** 1
Prioridad: Alta	**Tiempo Estimado:**48h
Riesgo en Desarrollo: • Rotura del host donde esté instalado el servidor de BD. • Afectaciones al personal de trabajo, debido a orientaciones de la dirección del país o de la universidad.	**Tiempo Real:**48h
Descripción:esta funcionalidad permite visualizar el modelo digital del terreno de un área seleccionada del mapa en una tercera dimensión.	
Observaciones:	

Pueden realizar esta acción todos los roles del sistema.

El usuario debe estar autenticado

Se crea la capa de superficie 3D dentro del panel control de capas.

Se selecciona en el mapa presionando el clic izquierdo y arrastrando el mouse en el área en que se desea visualizar el modelo digital del terreno en 3D; mostrándose en el área resaltada del mapa el valor de la longitud de la superficie y el punto máximo del área seleccionada.

Se muestra en la parte inferior de la ventana un control de teclas a partir de las cuales se pueden realizar visualizaciones del área mostrada:

Para arrastrar el área se presiona el clic izquierdo del mouse y se arrastra el mismo.

Para desplazarse en el área mostrada a través de las teclas **(a, w, d, s)** o con las flechas del teclado.

Para subir y bajar el área seleccionada las teclas **(r, f)** o a través de la acción del mouse utilizando el scroll.

Si se desea modificar los atributos del mapa, se da clic en la opción **Modificar atributos del mapa:**

Textura: campo de selección única donde se selecciona el tipo de textura en el que se desea mostrar la imagen, ya sea Satelital, Relieve, Mapa o Ninguna

Escala 1 en: campo de selección única donde se selecciona la escala de la textura en el que se desea mostrar la imagen.

Escala del mapa: Campo de selección únicadonde se selecciona la escala del mapa en la que se desea mostrar la vista.

Escalas de ejes (m): Campo de selección única donde se selecciona el valor de la escala en metros para los ejes de coordenadas.

Si se desea activar o desactivar la representación de visibilidad en el mapa se selecciona la opción, **Opciones para el análisis de visibilidad**, muestra la interfaz Visibilidad con las opciones siguientes:

Posicionar: se muestra en la barra de herramientas la opción **Posicionar,** se selecciona la opción, se posiciona la cámara en el origen mirando hacia la dirección escogida.

Se muestra el checkbox**Activar**, si se selecciona la opción se activan los campos Origen, Dirección y Altura del origen (m).

Origen: punto de origen de la visibilidad.

Dirección: orientación en la que se desea hacer el análisis de visibilidad. Se muestra en una línea blanca.

Altura del origen (m): altura en metros del punto de origen de la visibilidad**.**

Si se desea realizar una panorámica del mapa se selecciona la opción **Panorámica en el mapa,** se muestra la interfaz Panorámica con las acciones siguientes:

Alrededor de la cámara: Se selecciona si se desea realizar una panorámica alrededor de la cámara.

Alrededor de un punto: Se selecciona si se desea realizar una panorámica alrededor de un punto.

Si se desea realizar un análisis del nivel del agua, se selecciona la opción **Nivel del agua**, se muestra la interfaz Nivel del agua con las acciones siguientes:

Se muestra en la barra de herramientas las opciones **Subir nivel del agua** ()y **Bajar nivel del agua** ().

Nivel del agua (m): Muestra el valor del nivel del agua, que se obtiene al realizar las acciones. Permite introducir el valor del nivel del agua que se desea aumentar o disminuir.

Para incluir Complementos en el mapa, se muestra la interfaz Complementos con los datos siguientes:

Mapa de referencia: muestra el punto más alto en color rojo, el punto donde se encuentra la vista del mapa en color verde y el centro del área mostrada.

Si se desea realizar cálculos de distancia ver **RF Calcular distancia en 3D**.

Las validaciones para los campos de la interfaz son las siguientes:

Si el área seleccionada excede los 100 km/2 se muestra el mensaje de confirmación *"El área seleccionada (**longitud del área** km 2) es mayor que 100km 2, visualizar el mapa con estas dimensiones en 3d puede tardar varios minutos. **¿Desea continuar de todos modos?**"*.

Si la tarjeta gráfica de la máquina del usuario no soporta la visualización en 3D, se muestra la ventana con el siguiente mensaje *"Su tarjeta gráfica no soporta WebGL. Para saber cómo conseguirlo visite aquí (link con el sitio)".*

Si no se tiene información sobre esa área en el modelo digital del terreno (MDT), se muestra un mensaje de error: "*El modelo digital del terreno no puede ser consultado para esta área*.".

Mientras se muestra el modelo digital del terreno se muestra una máscara "*Cargando*".

Mientras se visualiza la imagen se muestra un mensaje: "*Por favor espere. Generando vista tridimensional del terreno.*".

Cuando se visualiza el modelo digital del terreno se visualiza con el punto más alto del terreno señalizado y expresado en (m), y con el mapa de referencia activo.

Se validan los campos de la interfaz Modificar mapa de la siguiente forma:

Textura: combobox que permite seleccionar el tipo de textura. Permite introducir el tipo de textura deseada, si se conoce la denominación de la misma. Si no se selecciona un tipo de textura se muestra inactivo el campo **Escala 1 en.**

Escala 1 en:combobox que solo permite introducir números enteros positivos, cuyo valor máximo requerido es 1000000. Permite realizar una selección de la escala.

Escala del mapa: combobox que solo permite introducir números enteros positivos, cuyo valor máximo requerido es 20. Permite realizar una selección del valor de la escala del mapa.

Escalas de ejes (m): combobox que solo permite introducir números enteros positivos, cuyo valor máximo requerido es 2000. Permite realizar una selección del valor de la escala de ejes en metros.

Si se introducen datos inválidos en los campos de la interfaz Modificar mapa, se muestra el mensaje de error en el tooltip del campo: *"El valor en este campo es inválido".*

Si se dejan campos obligatorios vacíos se muestra el mensaje de error en el tooltip del campo: *"Este campo es obligatorio"*.

La opción **Actualizar** solo se mostrará activa cuando se realice algún cambio en los campos de la interfaz y los mismos no contengan datos inválidos o estén vacíos. El botón tendrá un tooltip con la siguiente información: "*Actualizar el mapa con los atributos modificados*.".

Si la escala seleccionada no contiene datos georeferenciados, se muestra un mensaje de información: "*No se puede visualizar el modelo digital del terreno a la escala seleccionada.*".

Si no se selecciona el checkbox Activar, se muestran inactivas las acciones Origen y Dirección y el campo Altura del origen (m).

El campo Altura del origen (m) es un campo numérico el valor de la altura que tiene por defecto es 10. Si se introducen datos inválidos se muestra el mensaje de error en el tooltip del campo "*Solo admite números positivos".*

Si el campo Altura del origen (m), se encuentra vacío se muestra el mensaje de error en el tooltip del campo *"Este campo es obligatorio".*

La opción **Posicionar** solo se mostrará activa cuando se realice algún cambio en los campos de Visibilidad. El botón tendrá un tooltip con la siguiente información: *"Posicionar la cámara en el origen mirando hacia la dirección escogida".*

La panorámica que se muestra por defecto en la interfaz Panorámica es: Alrededor de la cámara.

Una vez que se seleccione la panorámica deseada se muestra en la barra de herramientas en la acción Realizar panorámica, el ícono que tiene relación con la panorámica seleccionada.

Una vez realizada la acción de realizar panorámica, se muestra en la barra de herramientas la acción Parar panorámica, se selecciona si se desea detener la panorámica realizada.

Una vez que se seleccione la panorámica deseada se muestra en la barra de herramientas en la acción realizar panorámica, el ícono y el mensaje en el tooltip que tiene relación con la panorámica seleccionada:

Alrededor de la cámara: se muestra la opción en la barra de herramientas Realizar panorámica alrededor de la cámara representada por la icnografía ( ).

Alrededor de un punto: se muestra la opción en la barra de herramientas Realizar panorámica alrededor de un punto representada por la icnografía ( ).

El campo Nivel del agua (m), es un campo numérico, los valores que se introducen se dan en metros. Si se introducen datos inválidos se muestra el mensaje de error en el tooltip del campo *"El valor introducido no es un número válido".*

Si se deja vacío el campo Nivel del agua (m), se muestra el mensaje de error en el tooltip del campo: *"Este campo es obligatorio".*

Prototipo de interfaz gráfica de usuario:

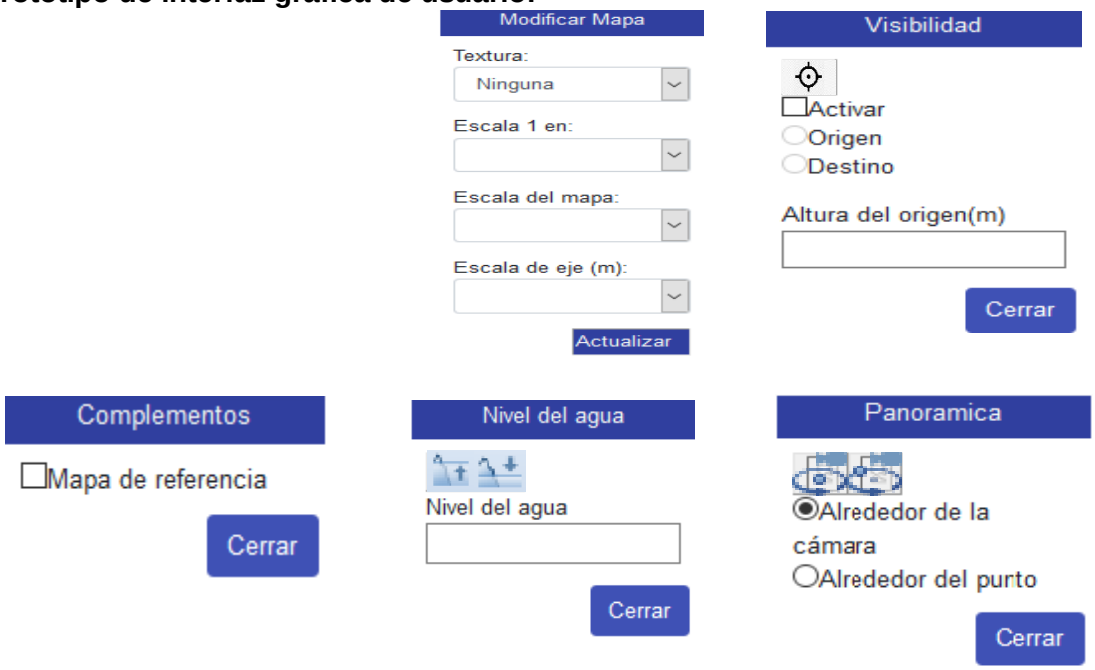


Tabla 3:HU Calcular distancia en 3D. Fuente: elaboración propia

Historia de usuario	
Número: 3	**Requisito:** Calcular distancia en 3D.
Programador: Ricardo Díaz Vilches	**Iteración Asignada:** 1
Prioridad: Alta	**Tiempo Estimado:**48h

Riesgo en Desarrollo:	**Tiempo Real:**48h
• Rotura del host donde esté instalado el servidor de BD. • Afectaciones al personal de trabajo, debido a orientaciones de la dirección del país o de la universidad.	

Descripción: Esta funcionalidad permite Calcular la distancia en 3D del modelo digital del terreno de un área seleccionada del mapa en una tercera dimensión.

Observaciones:

El usuario debe estar autenticado.

Pueden realizar esta acción todos los roles del sistema

El sistema muestra la interfaz **Distancia** con los siguientes campos:

Unidades de distancia: campo de selección única que contiene las unidades de medida en las que se puede dar la distancia.

Distancia: campo que contiene la distancia total de la línea de distancia trazada.

Dibujar nueva distancia opción que permite crear una línea de distancia en el mapa.

Seleccionar línea de distancia opción que permite seleccionar la línea de distancia, presionando el clic izquierdo sobre la línea deseada.

Eliminar último punto opción que permite eliminar el último punto trazado, de la línea que se encuentra seleccionada.

Eliminar línea de distancia, opción que permite seleccionar la línea de distancia que se desea eliminar y se muestra un mensaje de información: *¿Está seguro que quiere eliminar el trazado de distancia seleccionado?*

Eliminar todas las líneas de distancia, opción que permite eliminar todas las líneas de distancias trazadas en el mapa.

Se muestra un panel con los datos de: (**Distancia acumulada, Distancia anterior, Latitud, Longitud y la Altura del punto**) al ubicar el cursor encima del punto.

Las validaciones para los campos de la interfaz son las siguientes:

El campo Distancia es informativo, no se pueden modificar sus datos, se actualizan en dependencia de la línea de distancia que se encuentra seleccionada.

Si no existen ninguna línea de distancia trazada se muestra el mensaje en el campo Distancia: *"No existen datos para mostrar".*

Si no se ha trazado al menos una línea de distancia solo se muestra activa la opción **Dibujar nueva distancia**.

La opción **Seleccionar línea de distancia** se mostrará activa cuando se haya trazado al menos una línea de distancia.

La opción **Eliminar último punto** se mostrará activa cuando se haya trazado al menos

una línea de distancia.

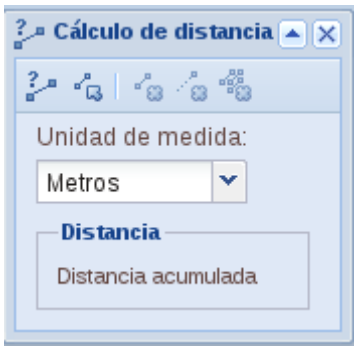
La opción **Eliminar línea de distancia** se mostrará activa cuando se haya trazado al menos una línea de distancia.

La opción **Eliminar todas las líneas de distancia** se mostrará activa cuando se haya trazado al menos una línea de distancia.

Los puntos se representan en el mapa en color rojo.

Los valores de Distancia acumulada y Distancia anterior, en el punto 1 son iguales a 0.

Prototipo de interfaz gráfica de usuario:



2.3. Disciplina de análisis y diseño

En esta disciplina los requisitos pueden ser refinados y estructurados para conseguir una comprensión más precisa, así como una descripción que sea fácil de mantener y ayude a la estructuración del sistema (incluyendo la arquitectura) (Sánchez, 2015). Además,se modela el módulo propuesto y su forma para que soporte todos los requisitos, incluyendo los requisitos no funcionales.

2.3.1. Diseño arquitectónico

El diseño arquitectónico garantiza cómo debe organizarse un sistema y cómo tiene que diseñarse la estructura global del mismo. En el modelo del proceso de desarrollo de software, el diseño arquitectónico es la primera etapa en el proceso de diseño del software. Es el enlace crucial entre el diseño y la ingeniería de requerimientos, ya que identifica los principales componentes estructurales en un sistema y la relación entre ellos. La salida del proceso de diseño arquitectónico consiste en un modelo que describe la forma en que se organiza el sistema como un conjunto de componentes en comunicación (Sommerville, 2011).

La solución propuesta tiene como base la arquitectura de la plataforma ULTRON 2.0. En la Figura 1 se representan los elementos de esta arquitectura y las partes donde incide el módulo propuesto. Es importante especificar que ULTRON 2.0 utiliza un modelo arquitectónico Cliente-Servidor. En este caso la arquitectura se divide en *frontend* (cliente) y *backend* (servidor). El *frontend* es la parte del software que interactúa con los usuarios y el *backend* es la parte que procesa la entrada desde el *frontend*. La idea general es que el *frontend* sea el responsable de recolectar los datos de entrada del usuario. Estos pueden ser de muchas y variadas formas, y los

transforma ajustándolos a las especificaciones que demanda el *backend* para poder procesarlos, devolviendo generalmente una respuesta que el *frontend* recibe y expone al usuario de una forma entendible para este. Para la solución propuesta la conexión entre ambas partes se realiza a través del protocolo *HTTP* (Protocolo de transferencia de hipertexto, por sus siglas en inglés *Hypertext Transfer Protocol*).

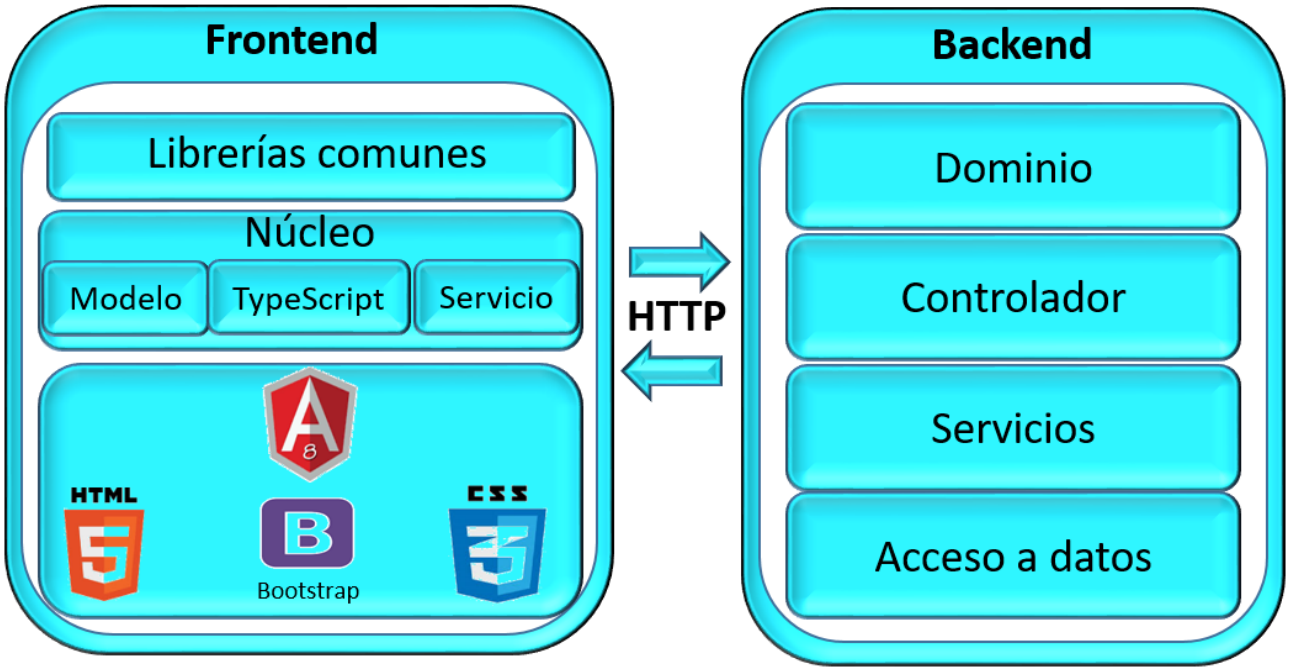


Figura 1: arquitectura ULTRON 2.0.
Fuente: elaboración propia

A partir de la figura anterior, en la parte del *frontend* el componente propuesto incide en los elementos del negocio. Este componente es implementado sobre la arquitectura *frontend* de ULTRON 2.0a través del uso del patrón arquitectónico Modelo Vista Controlador (MVC).

El patrón MVC separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos. Se trata de un modelo maduro y que ha demostrado su validez a lo largo de los años en todo tipo de aplicaciones, y sobre multitud de lenguajes y plataformas de desarrollo. El modelo contiene una representación de los datos que maneja el sistema, su lógica de negocio, y sus mecanismos de persistencia. La vista, o interfaz de usuario, compone la información que se envía al cliente y los mecanismos de interacción con éste. El controlador, actúa como intermediario entre el Modelo y la Vista, gestionando el flujo de información entre ellos y las transformaciones para adaptar los datos a las necesidades de cada uno (UA, 2019).

En la Figura 2 se muestra una descripción más detallada de cómo se representa en el *frontend*de ULTRON 2.0 la arquitectura del componente propuesto, para el caso de la HU visualizar mapa en 3D, a partir del uso del patrón arquitectónico MVC. En la figura el modelo está compuesto por el archivo *V3D.models.ts*, el cual contiene la información que se le solicita al usuario para confeccionar el reporte, ejemplo: fecha inicio y fecha fin. Por otra parte, la vista se compone por el archivo *V3D.component.html*, este representa la información visible al usuario e interactúa con

funciones específicas del controlador. El controlador está compuesto por el archivo *V3D.component.ts*, encargado de manipular el modelo y mostrar la información a la vista.

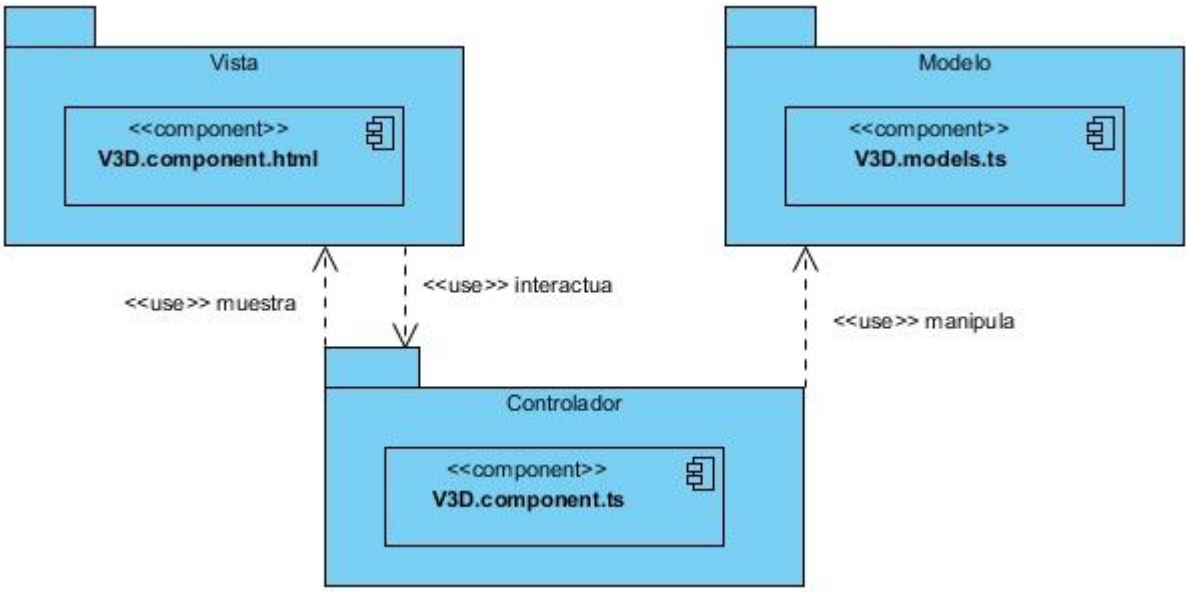


Figura 2: arquitectura *frontend* del módulo propuesto
Fuente: elaboración propia

A partir del diagrama de la Figura 1, en la parte del *backend* el componente propuesto incide en las capas controladora, servicios y acceso a datos. Este componente es implementado sobre la arquitectura *backend* de ULTRON 2.0a través del uso del patrón arquitectónico N Capas.

El patrón arquitectónico N Capas es una extensión del patrón Capas tradicional[22]. En el nivel más alto y abstracto, la vista de arquitectura lógica de un sistema puede considerarse como un conjunto de servicios relacionados agrupados en diversas capas (Safari, 2019).

En la Figura 3 se muestra una descripción más detallada de cómo se representa en el *backend* dela plataforma ULTRON 2.0la arquitectura del componente propuesto de manera genérica, a partir del uso del patrón arquitectónico N Capas. En la figura el componente *V3DController* de la capa Controlador usa al componente *V3DService* de la capa Servicios, el cual usa al componente de la capa de Acceso a Datos *V3DRepository* y los tres a la vez están orientados a la capa Dominio, la cual es el núcleo de la aplicación. La misma y su lógica de negocio definen el comportamiento y las restricciones de la aplicación, en cuanto a la forma en que se van a comunicar las demás capas con esta. Lo anterior hace que una aplicación sea diferente de otras en cuanto a la función que realiza.

[22]El patrón Capas tradicional ayuda a estructurar las aplicaciones que se pueden descomponer en grupos de subtareas en la que cada grupo de subtareas está en un nivel particular de abstracción

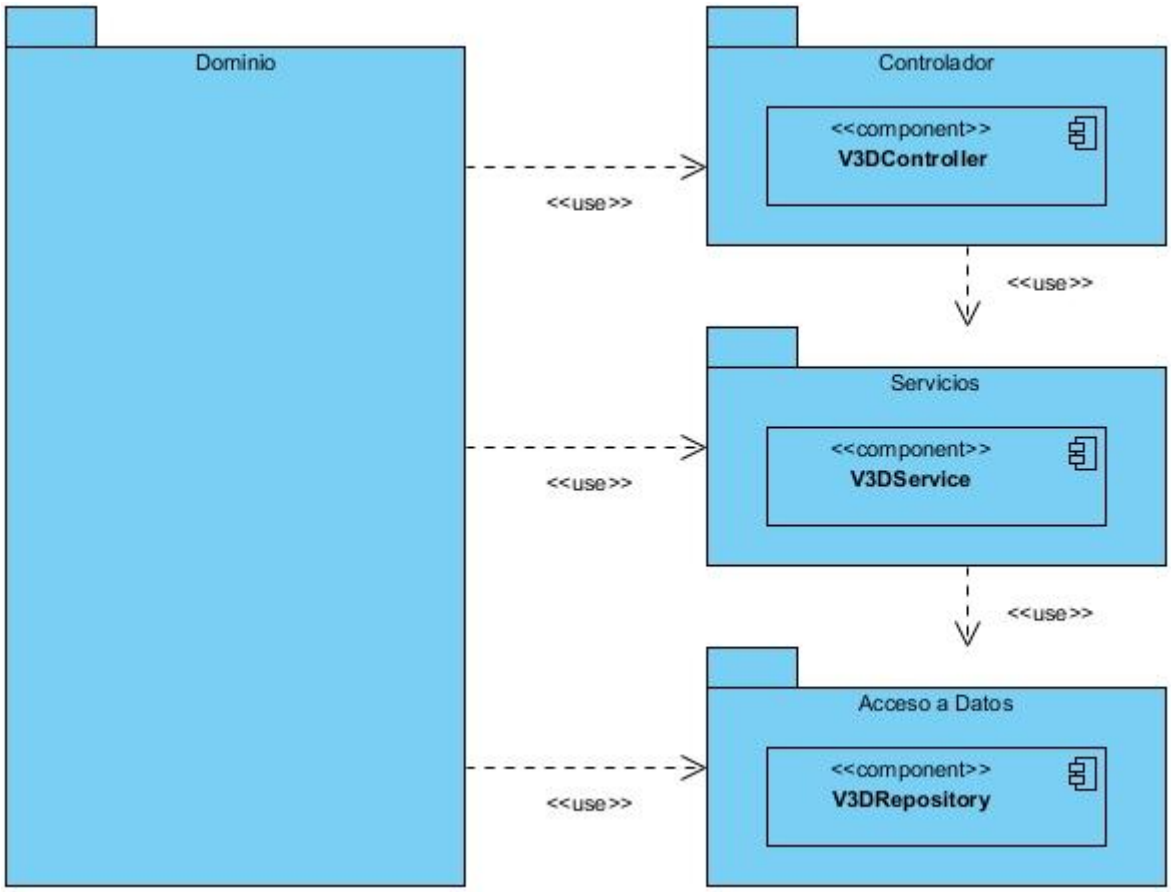


Figura 3: arquitectura *backend* del módulo propuesto
Fuente: elaboración propia

2.3.2. Patrones de diseño

Para diseñar el componente se emplearon un conjunto de patrones de diseño, los cuales son la base para la búsqueda de soluciones a problemas comunes en el desarrollo de software. A continuación, se describen los patrones empleados:

- **Patrones GRASP**[23]

Los Patrones Generales de Software para la Asignación de Responsabilidades (GRASP por sus siglas en inglés) describen los principios fundamentales de la asignación de responsabilidades a objetos, expresados en forma de patrones (Larman, 2016). En el caso de la presente investigación se aplicaron los siguientes patrones GRASP.

Creador: este patrón guía la asignación de responsabilidades relacionadas con la creación de objetos, tarea muy frecuente en los sistemas desarrollados a partir de un paradigma orientados a objetos. El propósito fundamental de este patrón es encontrar un creador que se debe conectar con el objeto producido en cualquier evento. Al escogerlo como creador, se da soporte al bajo acoplamiento lo cual supone menos dependencias respecto al mantenimiento y mejores oportunidades de reutilización (Larman, 2016).

[23] *General Responsibility Assignment Software Patterns* (Patrones Generales de Software para Asignación de Responsabilidades)

En la presente investigación el uso de este patrón se evidencia en la clase *V3D.component.ts.* El patrón se aplica en el momento de crear nuevas instancias de la clase Render3D, tal y como ilustra en el área señalada en la Figura 4.

```
var r3D = new render3D({ height:0, maxResolution:0.6, defaultHeight:3.5 });
vector.setRender3D(r3D);
 var listenerKey = this.vectorSource.on('change', function(e) {
  if (this.vectorSource.getState() == 'ready') {
    Observable.unByKey(this.listenerKey)
    $(".loading").hide();
    setTimeout (doAnime, 200);
  }
```

Figura 4:patrón Creador.Clase V3D.componet.ts
Fuente: elaboración propia

Bajo acoplamiento:su principal característica es mantener las clases más independientes entre sí y con la menor cantidad de relaciones; la cual posibilita que, en caso de producirse una modificación en alguna de ellas, se tenga la mínima repercusión posible en el resto de las clases, potenciando la reutilización y disminuyendo la dependencia entre las clases (Larman, 2016).

En la presente investigación el uso de este patrón se evidencia en el 46% de las clases del diseño, las cuales tienen una baja dependencia una de otras, tal y como se demuestra en el Capítulo 3 con los resultados de la validación del diseño a partir de la aplicación de métricas.

Alta cohesión: en la perspectiva del diseño orientado a objetos, la cohesión es una medida de cuán relacionadas y enfocadas están las responsabilidades de una clase. Una alta cohesión caracteriza a las clases con responsabilidades estrechamente relacionadas que no realicen un trabajo enorme (Larman, 2016). La principal función de la aplicación de este patrón es asignar responsabilidades de modo que la cohesión siga siendo alta. La información que almacena una clase debe de ser coherente y debe estar en la medida de lo posible relacionada con la clase.

En la presente investigación el uso de este patrón se evidencia en el 74% de las clases del diseño, las cuales tienen una baja sobrecarga de responsabilidades, tal y como se demuestra en el Capítulo 3 con los resultados de la validación del diseño a partir de la aplicación de métricas.

Patrón Controlador:un Controlador es un objeto de interfaz no destinada al usuario que se encarga de manejar un evento del sistema. Define además el método de su operación. (Larman, 2016).El empleo de este patrón se ve reflejado en la clase V3D del paquete cliente, encargada de manejar las peticiones que realiza el usuario y las respuestas que el sistema debe ofrecer.

Existe otro grupo de patrones de diseño: los patrones GOF, los cuales constan de un conjunto de 23 patrones de diseño divididos en 3 categorías: creacionales, estructurales y de comportamiento, estos fueron utilizados en la construcción del módulo, los cuales se describen a continuación.

- **Patrones de diseño GoF**

Patrones Banda de los Cuatro, por sus siglas en inglés de *GangofFour,* describen soluciones simples y eficaces a problemas específicos en el diseño de software orientado a objetos (Larman, 2016). En el caso de la presente investigación se aplicaron los siguientes patrones GoF:

Patrón Decorador: permite añadir responsabilidades a objetos concretos de manera dinámica y transparente sin afectar a otros objetos. Este patrón brinda más flexibilidad que la herencia estática y evita que las clases más altas en la jerarquía estén demasiado cargadas de funcionalidad y sean complejas(Larman, 2016).

En la presente investigación el uso de este patrón se evidencia en la clase *V3D.component.ts.* El patrón se aplica en el momento de hacer uso del decorador *@Component*, el cual es típico de las clases *Component* en Angular. En el área señalada en la Figura 5 se especifica el uso de este patrón.

```
@Component({
  selector: 'ultron-main',
  templateUrl: './V3D.component.html',
  styleUrls: ['./V3D.component.css']
})
```

Figura 5:aplicación del patrón decorador en la clase V3D.component.ts
Fuente: elaboración propia

Patrón Observador: permite observar los cambios producidos por un objeto, de esta forma, cada cambio que afecte el estado del objeto observado lanza una notificación a los observadores; a esto se le conoce como Publicador-Suscriptor. Es uno de los principales patrones de diseño utilizados en interfaces gráficas de usuario (GUI), ya que permite desacoplar al componente gráfico de la acción a realizar (Google, 2019).

En la presente investigación el uso de este patrón se evidencia en varias partes de la implementación de los componentes en el *frontend.* Porejemplo, al enviar los datos de la vista al controlador, este último hace la petición de un servicio a través de una inyección y se queda a la espera de la respuesta que este servicio debe proveer para así emitir o no un evento. En la Figura 6 se especifica a través del segundo recuadro un ejemplo de la aplicación de este patrón en la clase *V3D.component.ts.*

```
var listenerKey = this.vectorSource.on('change', function(e) {
 if (this.vectorSource.getState() == 'ready') {
   Observable.unByKey(this.listenerKey)
   $(".loading").hide();
   setTimeout (doAnime, 200);
 }
```

Figura 6: aplicación del patrón Observador en clases V3D.component.ts
Fuente: elaboración propia

Otros patrones

Patrón Inyección de dependencias: usado en la Programación Orientada a Objetos, que trata de solucionar las necesidades de creación de los objetos de una manera práctica, útil, escalable y con una alta versatilidad del código. La aplicación de este patrón se evidencia en la Figura 7 al crear los objetos, de una forma práctica, que serán utilizados en toda la clase:

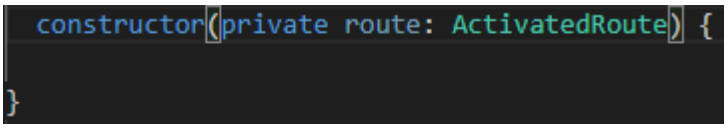


Figura 7: patrón Inyección de dependencia.Clase V3D.componet.ts
Fuente: elaboración propia

2.3.3. Diagrama de Clases del diseño

El diagrama de clases del diseño describe gráficamente las especificaciones de las clases de software y las interfaces que participan en el sistema con sus relaciones estructurales y de herencia. Los diagramas de este tipo contienen las definiciones de las entidades del software en vez de conceptos del mundo real. Los mismos son utilizados durante el proceso de análisis y diseño de los sistemas, donde se crea una vista lógica de la información que se maneja en el sistema, los componentes que se encargarán del funcionamiento y la relación entre uno y otro(Larman, 2016).

A continuación, se muestra una parte del diagrama de clases del diseño, en el cual se encuentran las clases correspondientes a la HU Modificar atributos del mapa, teniendo en cuenta que la misma responde al requisito anteriormente descrito en el documento. El diagrama de clases del diseño conla totalidad de las clases se encuentra entre los artefactos entregables de la tesis.

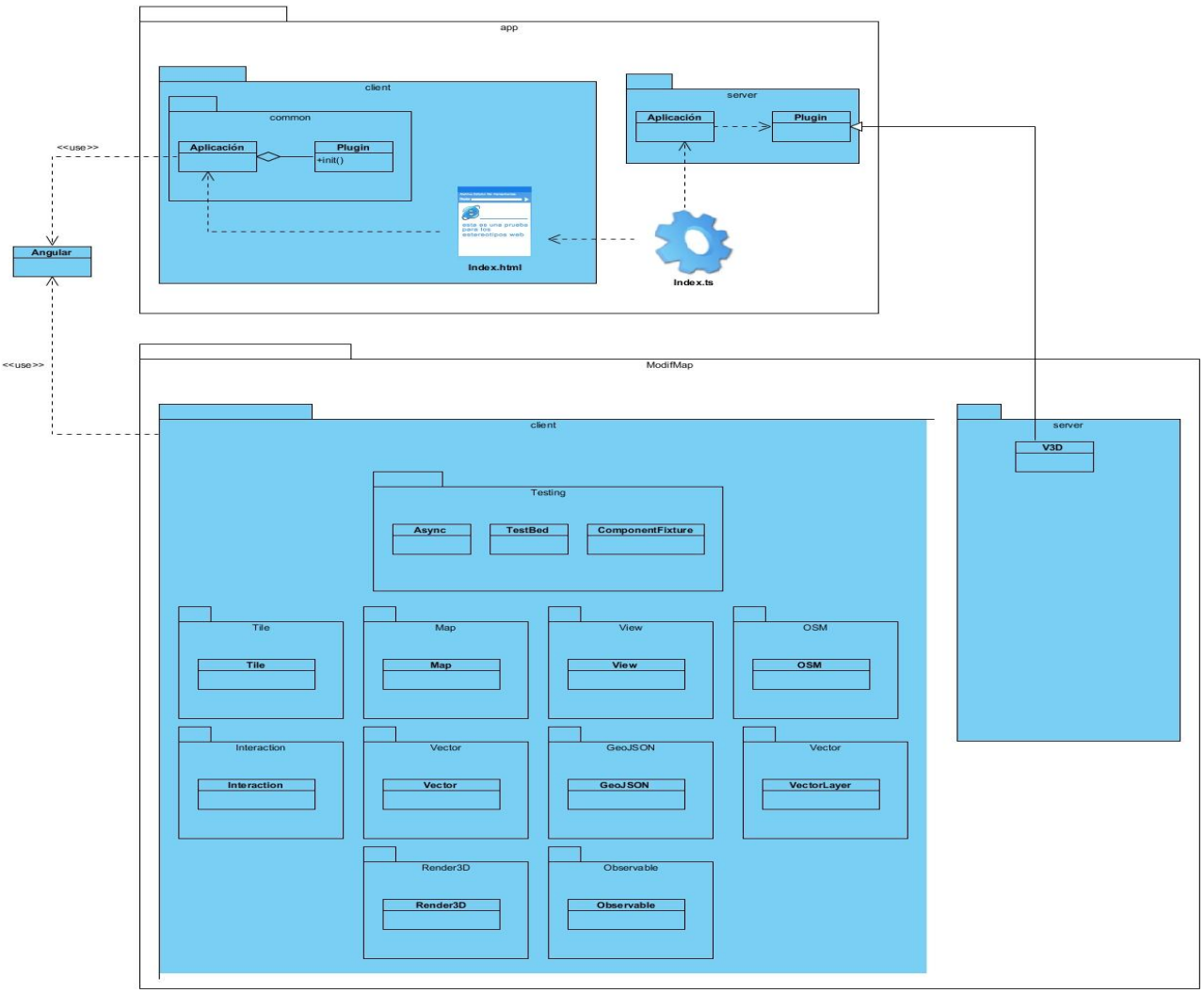


Figura 8: diagrama de clases de diseño para la HU Visualizar mapa en 3D.
Fuente: elaboración propia

2.3.4. Diseño de la Base de Datos

El diseño de la base de datos representa uno de los elementos fundamentales para la elaboración de un sistema, puesto que estructura correctamente el acceso a datos, con el objetivo de satisfacer la necesidad de obtención de reportes sobre estos datos(Padrón Castillo, 2015). Con el objetivo de describir la estructura lógica y física de la información persistente gestionada por lapropuesta de solución, a continuación, se muestra un fragmento del modelo de la base de datos existente para la plataforma web ULTRON 2.0, en el cual incide la propuesta de solución. Este fragmento del modelo de la base de datos cuenta con la información de los datos geográficos utilizados para dar respuesta al problema planteado.

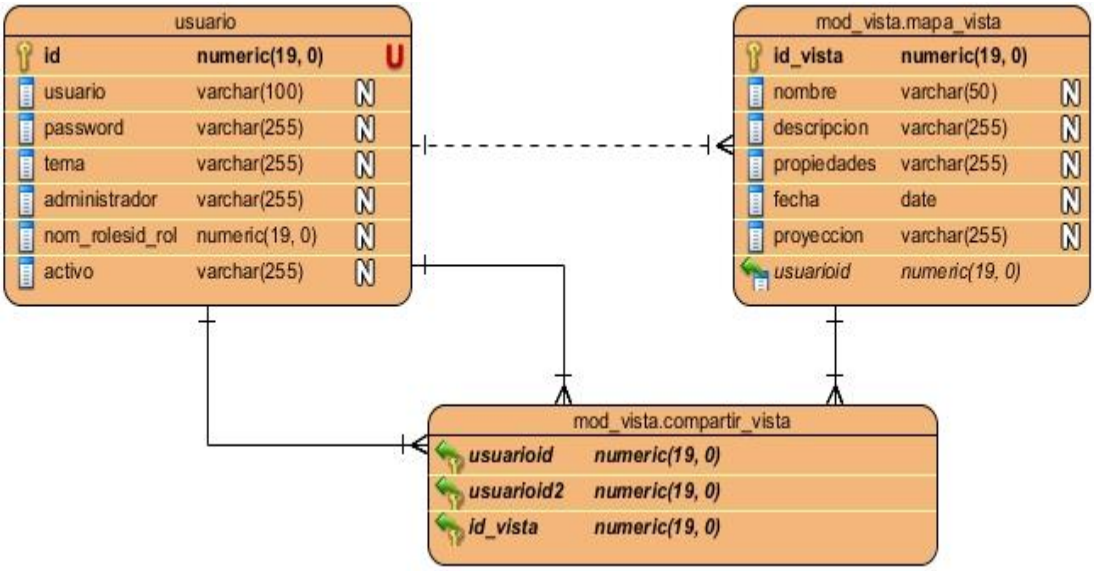


Figura 9:fragmento del modelo de datos de la plataforma web ULTRON.
Fuente: elaboración propia

Para lograr un mayor entendimiento de la Figura 9, a continuación, semuestra de forma sintetizada el objetivo que persigue larepresentación en la base de datos de cada entidad:

- **seg_perfilusuario:** es la entidad que se encarga de almacenar los datos correspondientes a los usuarios.
- **mapa_vista:** es la entidad que se encarga de almacenar los datos correspondientes a las vistas de los mapas.
- **compartir_vista:** es la entidad que se encarga de almacenar los datos generados a partir de la relación entre las entidades seg_perfilusuario y mapa_vista.

2.4. Estándares de codificación

Los estándares de codificación constituyen buenas prácticas o conjunto de reglas no formales que han ido surgiendo en las comunidades de desarrolladores de software con el paso del tiempo. Estos tienen el propósito de obtener un código fuente más legible, portable, seguro, eficiente, robusto y cohesionado, permitiendo mayor velocidad en el desarrollo y mejor coordinación entre los equipos de trabajo. Además, logran que para un desarrollador sea más fácil integrarse a un proyecto ya comenzado, se eviten *bugs*[24] y se faciliten los procesos de mantenibilidad y revisión de código (Hommel, 2019). A continuación, se describen los estándares de codificación utilizados para el desarrollo del módulo propuesto.

- Los nombres de las clases comienzan con letra mayúscula, en caso de ser un nombre compuesto las siguientes palabras se escribirán de igual forma, en la Figura 10 se evidencia el uso de este estándar en la clase *V3DComponent.*

```
class V3DComponent implements OnInit {
```

Figura 10: representación del estándar de codificación para el nombre de las clases.
Fuente: elaboración propia

[24]Error de software que desencadena un resultado indeseado

- Los nombres de los métodos se escriben con minúscula, en caso de ser un nombre compuesto las siguientes palabras inician con mayúscula.
- Los identificadores para las variables y los parámetros se escriben con letras en minúsculas y en caso de ser un nombre compuesto las siguientes palabras se escriben de igual forma.

En las sentencias*import*, en la Figura 11 se evidencia el uso de estos estándares de codificación.

```
import { Component, OnInit } from '@angular/core';
import { Tile } from 'ol/layer/Tile';
import { OSM } from 'ol/source/OSM';
import { Map } from 'ol/Map';
import { View } from 'ol/View';
import { interaction } from 'ol/interaction';
import { Vector } from 'ol/source/Vector';
import { GeoJSON } from 'ol/format/GeoJSON';
import { VectorLayer } from 'ol/layer/Vector';
import { render3D } from 'ol-ext/layer/Render3D';
import { Observable } from 'ol/Observable'
```

Figura 11: representación del estándar de para sentencia *import*.
Fuente:elaboración propia

- Los nombres de variables o funciones deben ser lo suficientemente descriptivos, sin exceder de 30 caracteres.
- Evitar líneas de más de 80 caracteres, pues no son bien interpretadas por terminales y herramientas.

2.5. Conclusiones parciales

En este capítulo se abordaron los principales elementos de la propuesta de solución con el fin de adquirir una visión de los flujos de información y las actividades que engloban el negocio, en el mismo se tiene que:

- La identificación y descripción de los requisitos del módulo propuesto permitió documentar toda la información relativa a la propuesta de solución y generar los artefactos necesarios.
- El estudio del diseño arquitectónico partiendo de los patrones de diseño, facilitó aislarse del problema en el que se enmarca la presente investigación, y a su vez identificar cómo interactúa el sistema en el que se desenvuelve la propuesta actual, obteniéndose así una solución con poca dependencia entre clases, flexible al mantenimiento ya la introducción de cambios.
- El establecimiento y utilización delos estándares de codificación contribuyó a obtener un código fuente más legible y cohesionado, permitiendo mayor velocidad en el desarrollo y mejor coordinación entre el equipo de trabajo.

Capítulo 3: Evaluación de la propuesta de solución

3.1. Introducción

En el presente capítulo se muestran los resultados obtenidos luego de aplicar las técnicas de validación de requisitos, y las métricas para la validación del diseño de la solución propuesta, documentándose los resultados obtenidosen cada caso. Por otra parte, se define la estrategia de prueba aplicada a partir de la disciplina de Pruebas internas establecida por la metodología *AUP* variación UCI para evaluar la calidad de la propuesta de solución. En la estrategia se establece que se realicen pruebas a nivel de unidad y de sistema, con la aplicación de sus respectivos métodos y técnicas.

3.2. Validación de los requisitos

Con el objetivo de asegurar que el módulo a desarrollarse corresponde con las necesidades del cliente, cada uno de los requisitos identificados fueron validados antes de llegar a la disciplina de análisis y diseño. Para la validación de estos se utilizaron las siguientes técnicas:

- **Generación de casos de prueba:**los casos de pruebas que responden a los RF del módulo propuesto fueron fáciles de diseñar, lo que significa que cada uno se interpretan de forma correcta y pueden ser comprendidos satisfactoriamente por los desarrolladores en la disciplina de implementación. En el caso de la presente investigación se generaron un total de 8 descripciones de casos de prueba (DCP), en la Tabla4 se describe la DCP correspondiente al RF: Visualizar mapa en 3D, el resto de las DCP generadas para la validación del módulo se encuentran entre los artefactos entregables de la tesis.

Tabla 4: Diseño de casos de prueba HU Visualizar mapa en 3D. Fuente: elaboración propia

Escenario(EC)	Textura	Escala 1 en	Escala del Mapa	Escalas de ejes	Respuesta del sistema	Flujo central
EC 1.1: visualizar mapa en 3D con campos correctos	V	V	V	V	El sistema permite modificar el mapa correctamente.	
	Satelital	*12459*	10	100		
EC 1.2: visualizar mapa en 3D con campos incompletos	I	V	V	V	El sistema muestra un mensaje indicando que: "Este campo es obligatorio"	
	(Vacío)	*15678*	17	50		

					y no habilita el botón Actualizar	
	V	I	V	V	El sistema muestra un mensaje indicando que: "Este campo es obligatorio" y no habilita el botón Actualizar.	Visualizar mapa en 3D
	Relieve	(Vacío)	15	150		
	V	V	I	V	El sistema muestra un mensaje indicando que: "Este campo es obligatorio" y no habilita el botón Actualizar.	
	Mapa	*2654*	(Vacío)	90		
	V	V	V	I	El sistema muestra un mensaje indicando que: "Este campo es obligatorio" y no habilita el botón Actualizar.	
	Relieve	*53421*	9	(Vacío)		
Las celdas de la tabla contienen V, I, o N/A. V indica válido, I indica inválido, y N/A que no es necesario proporcionar un valor del dato en este caso, ya que es irrelevante.						

- **Construcción de prototipos de interfaz de usuario:** esta técnica permite hacer simulacionesdel módulo a implementar y brinda la posibilidad a los especialistas de tener una idea de cómo serán lasinterfaces del módulo una vez desarrollado.En la Tabla 2 del epígrafe 2.2.4 se muestra el prototipo no funcional correspondiente al RF: Visualizar mapa en 3D. El

resto de los prototipos elaborados se encuentran relacionados en cada una de las HU generadas correspondiente a cada requisito funcional.

3.3. Validación del diseño

Un elemento clave de cualquier proceso de ingeniería es la medición. Pueden usarse medidas para entender mejor los atributos de los modelos que se crean y para valorar la calidad de los productos o sistemas sometidos a ingeniería que se construyen(Pressman, 2010). Para validar el diseño obtenido en la presente investigación, se decide aplicarlas métricas Tamaño Operacional de Clases (TOC) y Relaciones entre Clases (RC) definidas por (Lorenz, y otros, 1994), que a continuación se describen:

3.3.1. Tamaño Operacional de Clases

Esta métrica se basa en contar la cantidad de atributos y la cantidad de operaciones que tiene una clase individual y el promedio que presenta el sistema en su totalidad. A continuación, se describen los pasos que se llevaron a cabo para aplicar la métrica:

- Cálculo del umbral: el umbral se toma del tamaño general de una clase que sedetermina sumandotodas las operaciones que posee el diseño.
- Calcular el promedio de los umbrales.
- Teniendo en cuenta los valores antes obtenidos, se determina la incidencia de losatributos decalidad en cada una de las clases, según los criterios expuestos en la Tabla 5.

Tabla 5: rango de valores para la métrica TOC. Fuente: *(Lorenz, y otros, 1994)*

Atributos de calidad	Clasificación	Criterio
Responsabilidad	Baja	Umbral<= Promedio
	Media	Promedio< Umbral < = 2* Promedio
	Alta	Umbral > 2* promedio
Complejidad de implementación	Baja	Umbral <= Promedio
	Media	Promedio< Umbral < = 2* Promedio
	Alta	Umbral > 2* promedio
Reutilización	Baja	Umbral > 2* promedio
	Media	Promedio< Umbral < = 2* Promedio
	Alta	Umbral <= Promedio

En las tablas 6 y 7 se muestran los datos obtenidos una vez aplicada la métrica TOC sobre cada

una de las clases del diseño del módulo propuesto. Para el caso de la Tabla 7 se utilizan las siguientes abreviaturas:

- **Resp:** para el atributo de Responsabilidad
- **Comp. Imp:** para el atributo de Complejidad de implementación
- **Reut:** para el atributo de Reutilización

Tabla 6:resultados generales al aplicar la métrica TOC. Fuente: elaboración propia.

Total de clases	24
Promedio de procedimientos	2.375

Tabla 7: resultados obtenidos al aplicar la métrica TOC.Fuente: elaboración propia.

No.	Clase	Cantidad de procedimientos	Resp	Comp. Imp	Reut
1	*V3D.html*	2	Baja	Baja	Alta
2	*ModificarMapa.html*	1	Baja	Baja	Alta
3	*Visibilidad.html*	2	Baja	Baja	Alta
4	*Complementos.html*	1	Baja	Baja	Alta
5	*NivelAgua.html*	2	Baja	Baja	Alta
6	*Panoramica.html*	3	Media	Media	Media
7	*Distancia3D.html*	2	Baja	Baja	Alta
8	*ConfigurarTerreno.html*	4	Media	Media	Media
9	*MostrarCapasVectoriales3D.html*	5	Alta	Alta	Baja
10	*Etiquetas.html*	6	Alta	Alta	Baja
11	*MosaicoMapa.html*	2	Baja	Baja	Alta
12	*CuadrosDelimitadores.html*	8	Alta	Alta	Baja
13	*V3D.ts*	2	Baja	Baja	Alta
14	*ModificarMapa.ts*	1	Baja	Baja	Alta
15	*Visibilidad.ts*	1	Baja	Baja	Alta
16	*Complementos.ts*	3	Media	Media	Media
17	*NivelAgua.ts*	1	Baja	Baja	Alta
18	*Panoramica.ts*	1	Baja	Baja	Alta

19	*Distancia3D.ts*	1	Baja	Baja	Alta
20	*ConfigurarTerreno.ts*	2	Baja	Baja	Alta
21	*MostrarCapasVectoriales3D.ts*	1	Baja	Baja	Alta
22	*Etiquetas.ts*	1	Baja	Baja	Alta
23	*MosaicoMapa.ts*	3	Media	Media	Media
24	*CuadrosDelimitadores.ts*	2	Baja	Baja	Alta

En la Figura 12 se muestran los resultados en porciento (%) obtenidos a partir del análisis de los datos mostrados en las tablas 5 y 6, los cuales permiten evaluar los atributos anteriormente descritos.

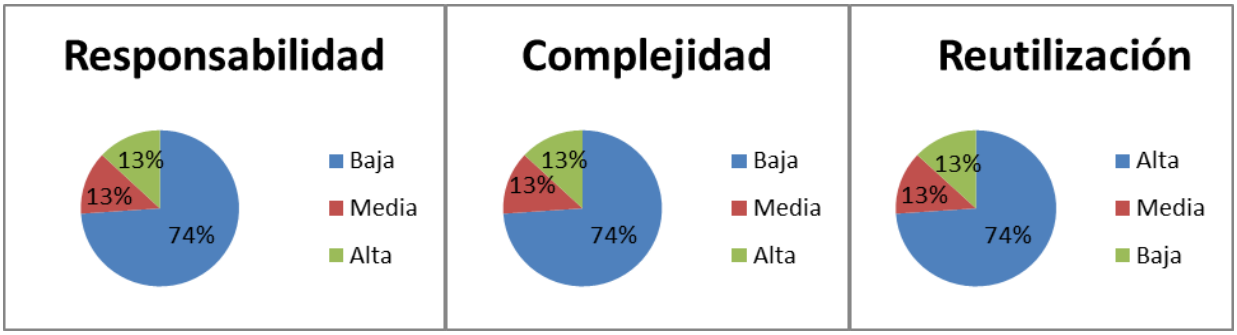


Figura 12: representación en (%) de los resultados de la aplicación de la métrica TOC.
Fuente: elaboración propia

Con la aplicación de la métrica TOC se obtienen los siguientes resultados:

- Responsabilidad: los resultados fueron satisfactorios, teniendo en cuenta que el 74% de las clases tienen una responsabilidad baja.
- Complejidad de implementación: los resultados fueron positivos, pues se demostró que el 74% de las clases tienen una complejidad baja.
- Reutilización: los resultados obtenidos fueron satisfactorios, demostrándose que el 74 % de las clases poseen una reutilización alta.

Haciendo un análisis de los resultados obtenidos con la aplicación de la métrica TOC se concluye que el diseño del módulo propuesto tiene una baja responsabilidad y complejidad de implementación de las clases que lo componen, así como un alto grado de reutilización de estas. Todo lo anterior facilita la obtención de una solución informática con poca dependencia entre clases, flexible al mantenimiento y a la introducción de cambios.

3.3.2. Relaciones entre clases

La métrica RC está dada por el número de relaciones de uso de una clase con otra. Elprimer paso en la aplicación de esta es evaluar los siguientes atributos de calidad: acoplamiento,complejidad

de mantenimiento, reutilización de cada clase y la cantidad de pruebas que cada clase requiere (Pressman, 2010).

A continuación, se exponen los pasos que se llevaron a cabo para aplicar la métrica atodas las clasesdel módulo propuesto en la presente investigación:

- Determinar la Cantidad de Relaciones de Uso (CRU) que poseen las clases a medir.
- Calcular el promedio de las CRU.
- Teniendo en cuenta los valores antes obtenidos se determina la incidencia de los atributos decalidad en cada una de las clases, según los criterios expuestos en la Tabla 8.

Tabla 8: rango de valores para medir la métrica RC. Fuente: *(Lorenz, y otros, 1994)*

Atributos de calidad	Clasificación	Criterio
Acoplamiento	Ninguna	CRU=0
	Baja	CRU=1
	Media	CRU=2
	Alta	CRU>2
Complejidad de mantenimiento	Baja	CRU<= Promedio
	Media	Promedio <CRU< = 2* promedio
	Alta	CRU> 2* promedio
Reutilización	Baja	CRU> 2* promedio
	Media	Promedio <CRU< = 2* promedio
	Alta	CRU<= Promedio
Cantidad de pruebas	Baja	CRU<= Promedio
	Media	Promedio <CRU< = 2* promedio
	Alta	CRU> 2* promedio

En las tablas9y 10se muestran los datos obtenidos una vez aplicada la métrica RC sobre cada una de las clases del diseño del módulo propuesto. Para el caso de la Tabla 10se utilizan las siguientes abreviaturas:

- **Acop:**para el atributo de acoplamiento.
- **Comp. Mant:** para el atributo de complejidad de mantenimiento.
- **Reut:**para el atributo de reutilización.
- **Cant. Prueb:** para el atributocantidad de pruebas.

Tabla 9: resultados generales al aplicar la métrica TOC. Fuente: elaboración propia.

Total de clases	24
Promedio de asociaciones de uso	1.7083333333

Tabla 10: resultados obtenidos al aplicar la métrica RC. Fuente: elaboración propia

No.	Clase	Cantidad de Relaciones de Uso	Acop.	Comp. Mant.	Reut.	Cant. Prueb.
1	*V3D.html*	1	Bajo	Baja	Alta	Baja
2	*ModificarMapa.html*	1	Bajo	Baja	Alta	Baja
3	*Visibilidad.html*	3	Alto	Media	Media	Media
4	*Complementos.html*	2	Medio	Media	Media	Media
5	*NivelAgua.html*	1	Bajo	Baja	Alta	Baja
6	*Panoramica.html*	1	Bajo	Baja	Alta	Baja
7	*Distancia3D.html*	2	Medio	Media	Media	Media
8	*ConfigurarTerreno.html*	1	Bajo	Baja	Alta	Baja
9	*MostrarCapasVectoriales3D.html*	1	Bajo	Baja	Alta	Baja
10	*Etiquetas.html*	1	Bajo	Baja	Alta	Baja
11	*MosaicoMapa.html*	2	Medio	Media	Media	Media
12	*CuadrosDelimitadores.html*	2	Medio	Media	Media	Media
13	*V3D.ts*	2	Medio	Media	Media	Media
14	*ModificarMapa.ts*	1	Bajo	Baja	Alta	Baja
15	*Visibilidad.ts*	2	Medio	Media	Media	Media
16	*Complementos.ts*	3	Alto	Media	Media	Media
17	*NivelAgua.ts*	1	Bajo	Baja	Alta	Baja
18	*Panoramica.ts*	2	Medio	Media	Media	Media
19	*Distancia3D.ts*	1	Bajo	Baja	Alta	Baja
20	*ConfigurarTerreno.ts*	3	Alto	Media	Media	Media
21	*MostrarCapasVectoriales3D.ts*	2	Medio	Madia	Media	Media
22	*Etiquetas.ts*	1	Bajo	Baja	Alta	Baja

23	*MosaicoMapa.ts*	2	Medio	Media	Media	Media
24	*CuadrosDelimitadores.ts*	3	Alto	Media	Media	Media

En la Figura 13 se muestran los resultados en porciento (%) obtenidos a partir del análisis de los datos mostrados en las tablas 9 y 10, los cuales permiten evaluar los atributos: acoplamiento, complejidad de mantenimiento, reutilización y cantidad de pruebas, con la aplicación de la métrica RC.

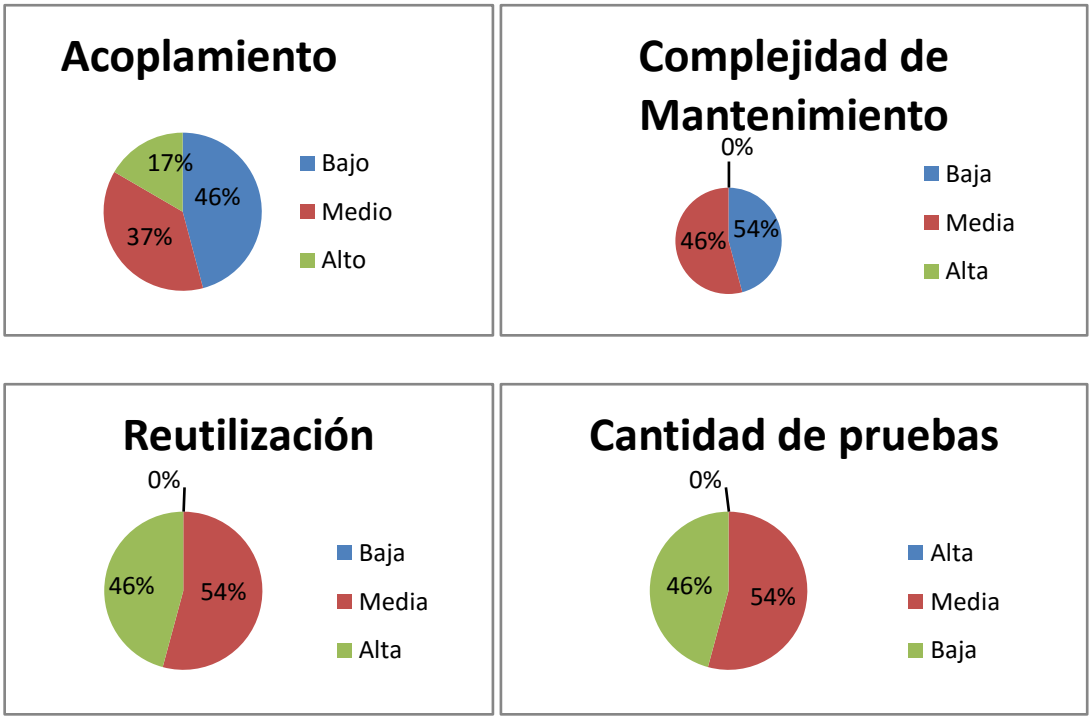


Figura 13: representación en (%) de los resultados de la aplicación de la técnica RC.
Fuente:elaboración propia

Con la aplicación de dicha métrica se obtienen los siguientes resultados:

- Acoplamiento: los resultados obtenidos son positivos teniendo en cuenta que el 46% de las clases poseen un bajo acoplamiento.
- Complejidad de mantenimiento: los resultados mostrados en la Figura 13, demuestran que el 46% de las clases del módulo propuesto presentan una complejidad de mantenimiento baja, facilitando así las futuras actividades de soporte sobre el mismo.
- Cantidad de pruebas: los resultados demuestran que el 46% de las clases poseen un bajo grado de esfuerzo a la hora de realizar cambios, rectificaciones o pruebas de software sobre ellas.
- Reutilización: los resultados obtenidos fueron satisfactorios, demostrándose que el 46% de las clases tienen una reutilización alta.

Con los resultados obtenidos una vez aplicada la métrica RC se concluye que el diseño del módulo propuesto tiene una baja complejidad de mantenimiento y acoplamiento entre sus clases.

Además, se requiere de un bajo grado de esfuerzos para realizar cambios sobre la mayoría de las clases y estás a su vez presentan un elevado porciento de reutilización. Todo lo anterior facilita la obtención de una solución informática escalable, con poca dependencia entre clases, flexible al mantenimiento y a la introducción de cambios.

3.4. Pruebas de software

Las pruebas de software son un conjunto de actividades que pueden ser planificadas con antelación y ejecutarse sistemáticamente durante la implementación o al finalizar el desarrollo del software. Estas se realizan para identificar posibles fallos de funcionamiento, configuración o usabilidad de un programa o aplicación. Las pruebas de software se ejecutan a partir de la aplicación de métodos y técnicas. La organización del proceso de pruebas se realiza a través de cuatro niveles: unidad, integración, sistema y aceptación (Pressman, 2010).

Para la validación del módulo propuesto en la presente investigación se define una estrategia de prueba de software aplicada a partir de las disciplinas establecidas para el desarrollo de las pruebas por la metodología *AUP* variación UCI. En la estrategia se define aplicar la disciplina de Pruebas internas a nivel de unidad y de sistema, con la aplicación de sus respectivos métodos y técnicas. Se decide no aplicar las disciplinas de Pruebas de Liberación y Aceptación que plantea dicha metodología, teniendo en cuenta que el alcance de la tesis solo comprende el desarrollo del módulo para ser integrado a la plataforma web ULTRON 2.0, software que aún no se ha terminado la implementación en su totalidad y por ende no se ha dado paso a la aceptación con el cliente.

3.4.1. Pruebas internas

En la disciplina de Pruebas internas se verifica a nivel de equipo de desarrollo el resultado de la implementación. Para la ejecución de estas pruebas se deben desarrollar artefactos de apoyo, tales como: diseños de casos de prueba, listas de chequeo y de ser posible componentes de prueba ejecutables para automatizar el proceso(Sánchez, 2015).

Para la validación del módulo propuesto, las pruebas internas se ejecutan a nivel de unidad y sistema. A continuación, se detalla cómo se llevaron a cabo dichas pruebas.

Pruebas a nivel de unidad

Las pruebas a nivel de unidad se concentran en el esfuerzo de verificación de la unidad más pequeña del diseño: el componente o módulo de software. Tomando como guía la descripción del diseño a nivel de componente, se prueban importantes caminos de control para describir errores dentro de los límites del módulo. El alcance restringido que se ha determinado para las pruebas de unidad limita la relativa complejidad de las pruebas y los errores que éstas descubren (Pressman, 2010).En el caso de la presente investigación en este nivel de prueba se aplicó el método de caja blanca, que a continuación se describe.

Método de caja blanca:

El método de caja blanca posibilita el desarrollo de casos de prueba que garanticen la ejecución, al menos una vez, de las rutas independientes (Pressman, 2010). En el caso de la presente tesis el método fue ejecutado aplicando la técnica de ruta básica.

La técnica de ruta básicatiene como objetivo comprobar que cada ruta se ejecute independiente de un componente o programa, obteniéndose una medida de la complejidad lógica del diseño. Esta técnica debe ser utilizada para evaluar la efectividad de los métodos asociados a una clase, con el objetivo de asegurar que cada ruta independiente sea ejecutada por lo menos una vez en el sistema (Pressman, 2010).

En la validación del módulo propuesto la técnica de ruta básica se aplicó a todos los métodos de las clases controladoras, teniendo en cuenta que estos agrupan las principales funcionalidades de la solución. A continuación, se describe la aplicación del método de caja blanca sobre la funcionalidad *animación* (ver Figura 14) de la clase *V3D.html.* Se toma como muestra esta funcionalidad teniendo en cuenta que es una de las de mayor complejidad, en la cual se definen un grupo de condicionales que posibilitan la obtención de varias rutas como resultado de la ejecución de la técnica aplicada.

```
render3D.prototype.getFeatureHeight = function (f) {   } 1
if (this.animate_) {   } 2
var h1 = this.height_(f);
var h2 = this.toHeight_(f);                                  } 3
Return (h1*(1-this.elapsedRatio_)+this.elapsedRatio_*h2);
} else   } 4
return this.height_(f);  } 5
}   } 6
}   } 7
```

Figura 14: método animación
Fuente*:* elaboración propia

Una vez definido el código sobre el cual se aplica el método, los pasos a seguir para desarrollar la técnica de ruta básica son los siguientes:

1) Confeccionar el grafo de flujo: este muestra el flujo de control lógico (Pressman, 2010). Está compuesto por los siguientes elementos:
   - Nodos: son círculos que representan una o más sentencias procedimentales.
   - Aristas: son flechas que representan el flujo de control y son análogas a las flechas deldiagrama de flujo.
   - Regiones: son las áreas delimitadas por aristas y nodos.

En la Figura 15 se muestra el grafo de flujo obtenido:

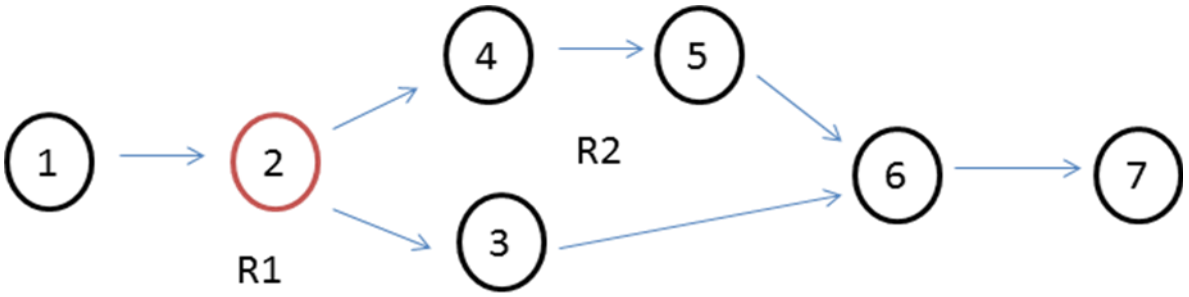


Figura 15: grafo de flujo del método animación
Fuente: elaboración propia

2) Calcular la complejidad ciclomática:

El valor calculadopor la complejidad ciclomática define el número de rutas independientes del conjuntobásico de un programa y brinda una cota superior para el número de pruebas quese deben realizar a fin de asegurar que todos los enunciados se ejecutaron al menos unavez (Pressman, 2010).

La complejidad ciclomática se calcula de tres formas diferentes, lascuales deben llegar al mismo resultado para comprobar que el cálculo es el correcto (Pressman, 2010).

- El número de regiones del grafo de flujo corresponde a la complejidadciclomática.
- La complejidad ciclomática V(G) para un grafo de flujo G se define como:

  V(G)=E-N+2

  Donde E es el número de aristas del gráfico de flujo y N el número de nodos delgráfico de flujo.
- La complejidad ciclomática V(G) para un grafo de flujo G también se definecomo

  V(G)=P+1

  Donde P es el número de nodos predicado (nodos de donde parten al menos dos aristas) contenidos en el grafo de flujo G.

En el grafo de flujo de la Figura 14, la complejidad ciclomática puede calcularse utilizándosecada una de las vías anteriormente descritas:

1. El grafo de flujo tiene 2 regiones, por tanto, V(G) = 2
2. V(G) = (7 aristas - 7 nodos) +2 = 2
3. V(G) =1 nodos predicado + 1= 2

Por tanto, la complejidad ciclomática del grafo de flujo de la Figura 15 es 2.

3) Determinar un conjunto básico de rutas linealmente independientes:

El valor de V(G) proporciona la cota superior sobre el número de rutas linealmenteindependientes a través de la estructura de control del programa(Pressman, 2010). En el caso de la funcionalidad*animación*, se definen 2 rutas básicas:

Ruta básica#1: 1, 2, 3, 6, 7

Ruta básica#2: 1, 2, 4, 5, 6, 7

4) Obtención de casos de prueba (CP):

Una vez definidas las rutas, se procede a diseñar los casos de prueba para cada una de las rutas básicas obtenidas. A continuación, en lastablas 11 y 12 se presentan el caso de prueba definido para cada una de las rutas básicas.

Tabla 11: caso de prueba de la ruta básica # 1. Fuente: elaboración propia

Descripción	Permite configurar una animación con una actualización de la altura del sistema
Condición de ejecución	La animación no se encuentre entre los datos del sistema
Entradas	Se introduce una nueva altura dada
Resultados esperados	Se debe mostrar la animación con la nueva altura introducida en el sistema

Tabla 12: caso de prueba de la ruta básica # 2.Fuente: elaboración propia

Descripción	Permite configurar la animación con la misma altura que tenía anteriormente el sistema
Condición de ejecución	La animación se encuentre entre los datos del sistema
Entradas	Se introduce la misma altura que se encuentra en el sistema
Resultados esperados	Se debe mostrar la animación con la altura que tenía anteriormente el sistema

Una vez ejecutados todos los casos de pruebas obtenidos con la técnica empleada, se concluye que los mismos fueron probados satisfactoriamente, corrigiéndose los errores surgidos en una primera iteración y comprobándose su corrección en una segunda. Al concluir la prueba se demuestra que todas las funcionalidades del módulo se ejecutan satisfactoriamente, quedando libres de código repetido o innecesario.

Pruebas a nivel de sistema

Las pruebas a nivel de sistema tienen como objetivo verificar que se han integrado adecuadamente todos los elementos del sistema y que realizan las operaciones apropiadas funcionando como un todo. Es similar a la prueba de integración, pero con un alcance más amplio(Pressman, 2010).En el caso de la presente investigación en este nivel de prueba se aplicó el método de caja negra, que a continuación se describe.

Método de caja negra:

El método de caja negra se centra en los requisitos funcionales del software. Es decir, permite al ingeniero de software derivar conjuntos de condiciones de entrada que ejercitarán por completo todos los requisitos funcionales de un programa. El método de caja negra no es una opción frente a caja blanca. Es, en cambio, un enfoque complementario que tiene probabilidades de describir

una clase diferente de errores de los que se identifican con los métodos de caja blanca (Pressman, 2010).

El método de caja negra se ejecuta a partir del desarrollo de pruebas funcionales, con la intención de identificar errores en las siguientes categorías (Pressman, 2010):

- Funciones incorrectas o faltantes.
- Errores de interfaz.
- Errores en estructuras de datos o en acceso a bases de datos externas.
- Errores de comportamiento o desempeño.
- Errores de inicialización y término.

Para desarrollar el método de caja negra se encuentra el uso de las técnicas (Pressman, 2010):

- Partición de equivalencia: divide el campo de entrada en clases de datos que tienden a ejercitar determinadas funciones del software.
- Análisis de valores límites: prueba la habilidad del programa para manejar datos que se encuentran en los límites aceptables.

Para aplicar este método, según las técnicas descritas anteriormente, se tuvo en cuenta los Diseños de casos de pruebas generados en la presente investigación (ver epígrafe 3.2 del presente documento). Para ellos se efectuaron un total de 3 iteraciones para poder alcanzar resultados satisfactorios, atendiendo al correcto comportamiento del módulo ante diferentes situaciones. En la Figura 16 se realiza un resumen mediante una gráfica con el número total de No conformidades (NC) por iteración.

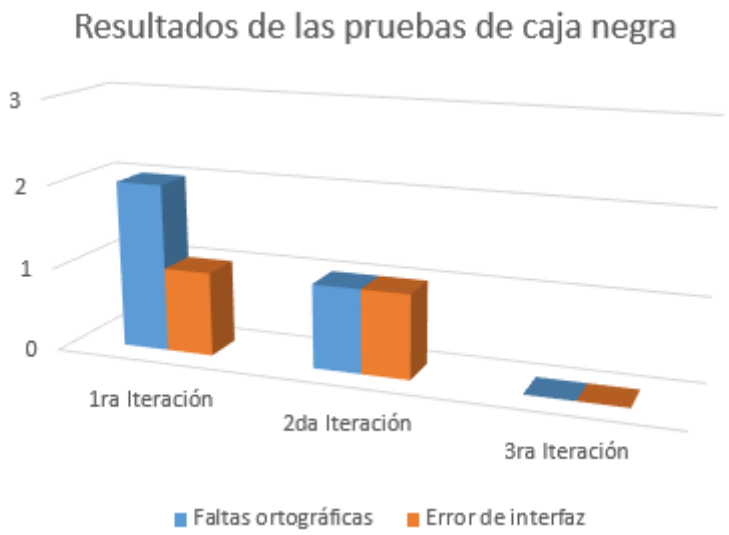


Figura 16: resultados de las pruebas de caja negra
Fuente: elaboración propia

En cada una de las iteraciones se corrigieron las NC dando paso a que en la tercera no se detectaran NC, lo que trajo consigo que la propuesta de solución se declarara libre de posibles errores durante su utilización.

3.5. Conclusiones parciales

En este capítulo se abordaron los principales aspectos con el fin de evaluar la calidad de los artefactos generados en la presente investigación, en el mismo se tiene que:

- La utilización de las técnicas de validación de requisitos aseguró que el módulo a desarrollarse corresponde con las necesidades del cliente.
- La aplicación de las métricas de validación del diseño permitió definir de forma cuantitativa la calidad del mismo en el desarrollo del módulo de visualización 3D en la plataforma web ULTRON 2.0.
- La ejecución de las pruebas de software, a nivel de unidad y de sistema evidenció que las funciones externas son operativas, que las entradas se aceptan de forma adecuada y que se producen salidas correctas, obteniendo finalmente una aplicación que satisface los requisitos definidos por el cliente.

Conclusiones generales

Con los resultados obtenidos a raíz de la presente investigación se concluye que:

- El establecimiento de los referentes teóricos sobre el proceso de representación de los MDT, enmarcándose en la visualización 3D en plataformas web, permitió lograr un mayor entendimiento sobre la propuesta de solución al problema planteado en esta investigación.
- La identificación de los requisitos, así como el análisis y diseño del módulo de visualización 3D en la plataforma web ULTRON 2.0, permitió obtener una aproximación y sentar las bases para la implementación de dicha propuesta de solución.
- La implementación del módulo de visualización 3D en la plataforma web ULTRON 2.0 permitió lograr una mejor visualización de los datos espaciales en dicha plataforma.
- La validación de la solución a través de las técnicas de validación de los requisitos, así como las métricas de validación del diseño y las pruebas de software favoreció comprobar de forma cuantitativa la calidad de los artefactos obtenidos durante el desarrollo de la solución propuesta.

Referencias bibliográficas

- **Aguilar Mugica, Karen . 2015.***LOS MODELOS DIGITALES DE ELEVACIÓN Y SUS PRODUCTOS DERIVADOS APLICADOS AL ESTUDIO DEL DETERIORO DE LA INFRAESTRUCTURA VIAL DEL TRANSPORTE, EN UN SECTOR DE LA HABANA.* La Habana : s.n., 2015.
- **Aguilera Fernández, Idania, y otros. 2016.***Impacto visual generado por la explotación minera en el yacimiento Punta Gorda, Moa.* Holguin : s.n., 2016. ISSN 1993 8012.
- **Alegsa. 2019.** [En línea] 2019. http://www.alegsa.com.ar/Dic/framework.php.
- **Alonso Sarria, Francisco . 2005.***SIG aplicados al análisis y cartografía de riesgos climáticos.* España : Universidad de Murcia, 2005.
- **Association of Modern Technologies Professionals. 2019.** It Knowledge portal. [En línea] 2019. http://www.itinfo.am/eng/software-development-methodologies/.
- **Babylon. 2009.** [En línea] 2009. http://diccionario.babylon.com/3D.
- **BARREDO CANO, JOSÉ IGNACIO y BOSQUE SENDRA, JOAQUÍN. 1996.***DELIMITACIÓN DE UNIDADE S HOMOGÉNEAS DEL RELIEVE A PARTIR DE UN MODELO DIGITAL DE ELEVACIONES.* España : s.n., 1996.
- **Bass, Len, Clements, Paul y Kazman, Rick. 2007.***Software architecture in practice.* s.l. : Pearson Education India, 2007.
- **Bravo Batista, Claudia. 2012.***Desarrollo del módulo para la generación de las vistas de los mapas para la plataforma soberana genesig.* La Habana : s.n., 2012.
- **Buschmann, Frank, y otros. 2000.***Pattern-Oriented Software Architecture, Patterns for Concurrent and Networked Objects, Volume 2.* s.l. : John Wiley & Sons, 2000.
- **de la Torre, César y Zorrilla, Unai. 2010.***Guía de Arquitectura N-Capas orientada al dominio con .NET 4.0.* España : s.n., 2010.
- **deperu. 2019.** deperu. [En línea] 2019. www.deperu.com.
- **eclac. 2019.** Economic Commission for Latin America and the Caribbean. [En línea] 2019. www.cepal.org/.
- **Felicísimo, Angel Manuel. 1999.***Simulación de procesos:modernización climática.* 1999.
- **Fernández de laTorre , Reynier y Geler Roffe, Tatiana. 2004.***MODELO DIGITAL DE ELEVACIÓN DE LA ZONA EMERGIDA DEL ECOSISTEMA SABANA CAMAGÜEY, CUBA.* La Habana : Editora Geotech, 2004.
- **Flores, Araceli Soledad Domínguez. 2013.***Unidad 1: UML y el proceso unificado. Desarrollo e implementación de Sistemas de Información.* 2013.
- **Gamma, Erich, y otros. 1994.***Patrones de diseño: Elementos del software orientado a objetos.* s.l. : Addison-Wesley, 1994.

- **Google. 2019.** Angular. [En línea] 2019. https://angular.io/guide/observables-in-angular.
- **—. 2019.** Angular. [En línea] 2019. angular.io/guide/dependency-injection.
- **—. 2019.** Angular. [En línea] 2019. https://angular.io/docs.
- **Hommel, Scott. 2019.***Estandares_de_codificacion_para_Java.* 2019.
- **Huidobro, José Manuel. 2015.***Telecomunicaciones. Tecnologías, Redes y Servicios.* Segunda. 2015. ISBN 9788499642741.
- **inegi. 2019.** [En línea] 2019. en.www.inegi.org.mx/Default.html.
- **ISO/IEC 14772. 1997.** Information technology — Computer graphics and image processing — The Virtual Reality Modeling Language — Part 1: Functional specification and UTF-8 encoding. [En línea] 1997. https://www.iso.org/standard/25508.html.
- **Jacobson, Ivar , Booch, Grady y Rumbaugh, James . 2000.***El proceso unificado de desarrollo de software.* Madrid : Addison Wesley, 2000.
- **Janssen, Thorben. 2019.** Thoughts on JAVA. *Implementing the Repository pattern with JPA and Hibernate.* [En línea] 2019. https://thoughts-on-java.org/implementing-the-repository-pattern-with-jpa-and-hibernate/.
- **JasperSoft. 2019.** JasperSoft Community. [En línea] 2019. community.jaspersoft.com/project/jasperreports-library.
- **JetBrains. 2019.** JetBrains. [En línea] 2019. https://www.jetbrains.com/idea/.
- **—. 2019.** JetBrains. [En línea] 2019. https://www.jetbrains.com/webstorm/.
- **Jiménez, Emilio Macías y Pérez, Mercedes de la Parte. 2006.***Simulación con VRML, JAVA3D y X3D.* España : s.n., 2006.
- **Kaisler. 2005.***S.H.Software Paradigms.* New Jersey : John Wiley & Sons, 2005. ISBN 0-471-48347-8.
- **Kimmig, Markus, Monperrus, Martin y Mezini, Mira. 2011.** [En línea] 2011. https://hal.archives-ouvertes.fr/hal-00640496/document.
- **Klipfolio Inc. 2019.** Klipfolio. [En línea] 2019. https://www.klipfolio.com/resources/articles/what-is-data-dashboard.
- **La Red Martínez, David L., y otros. 2012.***Aprendizaje combinado, aprendizaje electrónico centrado en el alumno y nuevas tecnologías.* Argentina : Red de Universidades con Carreras en Informática (RedUNCI), 2012.
- **Labañino, Linet Columbiet. 2011.***Informática Habana.* 2011.
- **Larman, Craig. 2016.***UML y Patrones. Una introducción al análisis y diseño orientado a objetos y al proceso unificado.* 3ra . s.l. : Prentice-Hall, 2016.
- **Ledesma, Rubén. 2008.***Introduccción al Bootstrap.* Argentina : s.n., 2008.
- **Lorenz, M. y Kidd, J. 1994.***Object-Oriented Software Metrics.* s.l. : Prentice-Hall, 1994.

- **Lorenz, M., & Kidd, J. 1994.***Object-oriented software metrics: a practical guide.* New Jersey, Prentice-Hall : s.n., 1994.
- **MapServer. 2011.** MapServer. *MapServer.* [En línea] 2011. [Citado el: 1 de noviembre de 2019.] http://mapserver.org/.
- **Martínez, Rafael. 2010.** Sobre postgreSQL. *PostgreSQL-es.* [En línea] octubre de 2010. [Citado el: 11 de diciembre de 2019.] http://www.postgresql.org.es/sobre_postgresql.
- **Mendez, Juan. 2011.** Quantum Gis (QGIS): Un Sistema De Información Geográfica Basado en Software Libre. [En línea] Kudos Ltda, 15 de Febrero de 2011. [Citado el: 20 de Diciembre de 2019.] https://gkudos.com/Quantum Gis (QGIS): Un Sistema De Información Geográfica Basado en Software Libre.html.
- **Mozilla. 2019.** MDN web docs mozilla. [En línea] 2019. developer.mozilla.org/en-US/docs/Web/JavaScript.
- **—. 2019.** MDN web docs mozilla. [En línea] 2019. developer.mozilla.org/en-US/docs/Web/CSS/CSS3.
- **—. 2019.** MDN web docs mozilla. [En línea] 2019. developer.mozilla.org/en-US/docs/Web/HTML.
- **Muñoz Aldana, Yennis. 2010.***Representación de Modelos Digitales del Terreno en 3D sobre la Web.* La Habana : Universidad de las Ciencias Informatica, 2010.
- **Nodejs. 2015.** Nodejs. [En línea] 6 de febrero de 2015. [Citado el: 11 de diciembre de 2019.] https://nodejs.org/en/blog/release/v0.12.0/..
- **Novoa Goicochea, Zaniel I y Ordoñez Gálvez, Juan Julio. 2012.***¿QUÉ ES CUENCA HIDROLÓGICA?* Perú : Sociedad Geográfia de Lima, 2012. ISBN: 978-9972-602-76-4.
- **OJEDA, JOSÉ ZÚJARLOS. 2000.***SISTEMAS DE INFORMACIÓN GEOGRÁFICA Y LA MODELIZACIÓN DEL PAISAJE.* España : s.n., 2000.
- **Olaya, Víctor. 2019.***Sistemas de Información Geográfica.* 2019.
- **ONEI. 2019.** Portal Web de la ONEI. *Portal Web de la ONEI.* [En línea] 17 de 05 de 2019. http://www.onei.cu/queeslaone.htm.
- **ORACLE. 2019.** ORACLE. [En línea] 2019. https://docs.oracle.com/en/java/.
- **Ordóñez Galan, Celestino y Martínez-Alegría, Roberto. 2003.***Sistemas de Información Geográfica. Aplicaciones prácticas con Idrisi 3.2 al análisis de riesgos naturales y problemáticas medioambientales.* España : RA-MA Editorial, 2003. ISBN: 978-8478975433.
- **OSGeo Live. 2012.** OSGeo Live. *OSGeo Live.* [En línea] 2012. [Citado el: 25 de 2 de 2020.] http://live.osgeo.org/es/overview/postgis_overview.html.
- **Padrón Castillo, Gabriel. 2015.***Componente web de visualización en 3D para objetos geo-referenciados en el SIG UCI.* La Habana : Universidad de las Ciencias Informática , 2015.

- **Paradigm, Visual. 2017.** [En línea] 2017. [Citado el: 20 de diciembre de 2019.] http://www.visualparadigm.com/aboutus/newsreleases/vpuml80.jsp..
- **Pivotal. 2019.** Spring. [En línea] 2019. spring.io/projects/spring-boot.
- **Pressman, Roger S. 2010.***Ingeniería de Software. Un enfoque práctico.* Septima . México DF : McGraw-Hill INTERAMERICA EDITORES, 2010.
- **QualityDevs S.L. 2020.** Quality Devs. [En línea] 2020. https://www.qualitydevs.com/2019/09/16/que-es-angular-y-para-que-sirve/.
- **Rodríguez Sánchez, Tamara. 2015.***Metodología de desarrollo para la Actividad productiva UCI.* La Habana : s.n., 2015.
- **Rodríguez, Nelson Santiesteban. 2012.***Sistema para la Gestión de la Información.* 2012.
- **Rojas, M. J. 2010.***Patrones de Diseño.* 2010.
- **Rouse, Margaret. 2019.** TeachTarget. *Database management system (DBMS).* [En línea] 2019. https://searchsqlserver.techtarget.com/definition/database-management-system.
- **—. 2019.** TechTarget. *User Story.* [En línea] 2019. https://searchsoftwarequality.techtarget.com/definition/user-story.
- **—. 2019.** TechTarget. *Integrated Development Environment (IDE).* [En línea] 2019. searchsoftwarequality.techtarget.com/definition/integrated-development-environment.
- **—. 2019.** TechTarget. *pattern (design pattern).* [En línea] 2019. https://searchsoftwarequality.techtarget.com/definition/pattern.
- **Safari. 2019.** Safari Books Online. *Oreilly.* [En línea] 2019. www.oreilly.com/library/view/software-architecture-patterns/9781491971437/ch01.html.
- **Sanchez, Ruby. 2018.** Especificación de Requisitos Software según el estándar de IEEE 830. [En línea] 2 de octubre de 2018. https://www.academia.edu/6647065/Especificaci%C3%B3n_de_Requisitos_Software_seg%C3%BAn_el_est%C3%A1ndar_de_IEEE_830.
- **Sánchez, Tamara Rodríguez. 2015.***Metodología de desarrollo para la Actividad productiva de la UCI. .* La Habana. Cuba : s.n., 2015.
- **Shaw, Mary y Garlan, David. 1996.***Software Architecture: Perspectives on an Emerging Discipline.* 1996.
- **SIEBER, R. y SCHNÜRER, R. 2012.***The Power of 3D Real-Time Visualization in Atlases–Concepts,.* 2012.
- **SNI. 2019.** [En línea] 2019. http://www.sni.org.pe/?page_id=872.
- **Sommerville, Ian. 2005.***Ingeniería del software. Séptima Edición.* Madrid. España : Pearson Educación. S. A., 2005. 84-7829-074-5.
- **The PostgreSQL Global Development Group. 2019.** PostgreSQL. *The World's Most Advanced Open Source Relational Database.* [En línea] 2019. https://www.postgresql.org/.

- **Typescript. 2019.** Typescript. [En línea] 2019. www.typescriptlang.org.
- **UA. 2019.** Universidad de Alicante. [En línea] 2019. https://si.ua.es/es/documentacion/asp-net-mvc-3/.
- **Vignaga, Andrés y Perovich, Daniel. 2003.***Enfoque metodológico para el desarrollo basado en componentes.* 2003.
- **Visual-Paradigm. 2019.** Visual-Paradigm. [En línea] 2019. www.visual-paradigm.com/solution/freeumltool/.
- **W3C. 2019.** W3C. [En línea] 2019. https://www.w3.org/html/.

Anexos

Anexo 1:Entrevistarealizada a los profesores del proyecto AplicativosSIG y especialistas del centro XETID

Acontinuación,serelacionanlaspreguntasutilizadasenlaentrevista.

1. ¿Cuáles son las funciones o aspectos elementales que deben tenerse en cuenta al realizar una representación de la información geográfica?
2. ¿Cuáles son las funciones o aspectos elementales que deben tenerse en cuenta al realizar una visualización en 3D?
3. ¿Cuáles son las ventajas de una visualización en 3D con respecto a la 2D?
4. ¿Qué beneficios traería esta visualización a la plataforma web ULTRON 2.0?
5. ¿Existen estándares o algoritmos para la visualización en 3D? ¿Cuáles serían estos?

Anexo 2:Entrevistarealizada a los profesores del proyecto AplicativosSIG para la obtención de los requisitos

Acontinuación,serelacionanlaspreguntasutilizadasenlaentrevista.

1. ¿Cuáles son los requisitos que ustedes desean que tengamos en cuenta para la realización de la visualización en 3D?
2. ¿Cuáles son los estándares establecidos que deben tenerse en cuenta para el diseño de las interfaces?
3. La plataforma web ULTRON 2.0 cuenta con una base de datos donde almacenan los datos para la representación de los mapas. ¿Cuál es la estructura de estos datos?