



Facultad de Ciencias y Tecnologías Computacionales

Módulo para tematizar mapas en la plataforma ULTRON basado en métodos de interpolación

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

Autor:

Daymara María García González

Tutor:

Ing. Celia Torres Reyes

Msc. Grethell Castillo Reyes

Co-tutor:

Ing. Alvaro Ernesto Gutiérrez Ortiz

La Habana, noviembre de 2023

Año 65 de la Revolución

DECLARACIÓN DE AUTORÍA

La autora del trabajo de diploma con título ***“Módulo para tematizar mapas en la plataforma ULTRON basado en métodos de interpolación”*** concede a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la investigación, con carácter exclusivo. De forma similar se declara como única autora de su contenido. Para que así conste firma(n) la presente el día 14 del mes de noviembre del año 2023.

Daymara María Garcia González

Firma del Autor

Ing. Celia Torres Reyes

Firma del Tutor

Ing. Alvaro E. Gutiérrez Ortiz

Firma del Co-tutor

MsC.Grethell Castillo Reyes

Firma del Tutor

DATOS DE CONTACTO

Tutora:

MsC.Grethell Castillo Reyes

Graduado de Ingeniero en Ciencias Informáticas en La Universidad de las Ciencias Informáticas en el año 2012, MsC. En Informática Aplicada en el año 2015. Trabajador del centro GEYSED en el departamento Integración de soluciones, donde desempeñó el rol de Jefe de Proyecto de la Línea de Productos de Software Aplicativos SIG, Jefe de Departamento de Desarrollo de Componentes y Directora del Centro GEYSED. Actualmente se desempeña como Especialista A en Ciencias Informáticas en el Centro para la Representación y Análisis de Datos. Correo Electrónico: gcreyes@uci.cu.

Tutora:

Ing. Celia Torres Reyes

Especialista "A" en Ciencias Informáticas. Graduada en la Universidad de las Ciencias Informáticas en el 2014. Actualmente desempeña el cargo de Jefa de grupo administrativo vinculado a la producción, desempeñando el rol de administradora de calidad y probador del Centro de Representación y Análisis de Datos (CREAD) Email: ctreyes@uci.cu.

Co-tutor:

Ing. Alvaro E. Gutiérrez Ortiz

Estudiante RGA. Graduado en la Graduado en la Universidad de las Ciencias Informáticas en el 2022. Actualmente desempeña el rol de desarrollado en el Centro de Representación y Análisis de Datos (CREAD). Email: alvaroego@uci.cu.

Autora:

Nombre y Apellidos: Daymara María García González

Email: daymaramgg@estudiantes.uci.cu.

AGRADECIMIENTOS

A mi esposo por todos sus consejos, su infinita paciencia y su apoyo incondicional durante todos estos años.

Quiero agradecer a mi familia, en especial a mi abuela y mis tíos por toda la confianza depositada en mí y por todo el apoyo que me han dado siempre en los momentos más difíciles de mi carrera.

A mis tutoras y mi cotutor por guiarme y preocuparse por mí y a todos los profesores que han contribuido a mi formación como profesional.

A mis mejores amigas que siempre han estado ahí apoyándome en todo momento.

A todos de corazón muchas gracias por acompañarme todos estos años.

DEDICATORIA

A mi esposo Alvaro Ernesto Gutiérrez Ortiz por tenerme muchísima paciencia sobre todo en esta etapa, por siempre estar a mi lado luchando por un futuro mejor y una familia juntos, por lograr muchas metas y las que nos faltan por cumplir, por ser el mejor novio y esposo del mundo.

A mi abuela y a mis tíos por haberme guiado desde niña, darme buenos consejos y orientarme de forma correcta, por la confianza que siempre depositaron en mí, por darme tanto amor y ser la mejor familia del mundo.

A mis amigos y a todas las personas que me apoyaron y de una forma u otra forman parte de mi vida.

RESUMEN

Un Sistema de Información Geográfica es un marco de trabajo para reunir, gestionar y analizar datos. El SIG muestra el conocimiento más profundo escondido en los datos, como patrones, relaciones y situaciones, ayudando al usuario a tomar decisiones más inteligentes. La presente investigación tiene como objetivo crear un módulo para tematizar mapas en la plataforma ULTRON basado en métodos de interpolación. La investigación está regida por la metodología AUP-UCI en el escenario cuatro: historias de usuarios. Para el desarrollo de este módulo se seleccionaron varias herramientas y tecnologías como el *framework* Angular, la biblioteca OpenLayers para la parte de interacción con el mapa, y para el backend se utilizó el *framework* Loopback basado en NodeJS y el gestor de base de datos PostgreSQL. Las pruebas aplicadas al módulo arrojaron resultados satisfactorios. Se realizaron pruebas de caja negra con la técnica de partición de equivalencia y pruebas de caja blanca con las pruebas unitarias utilizando como herramienta de prueba Jasmine. La aplicación permite la mejor visualización de la concentración geográfica mediante métodos de interpolación.

PALABRAS CLAVE

Aplicación, Interpolación, Módulo, Sistema de Información Geográfica, Tematización.

ABSTRACT

A Geographic Information System is a framework for collecting, managing and analyzing data. GIS shows the deeper knowledge hidden in the data, such as patterns, relationships and situations, helping the user make smarter decisions. The objective of this research is to create a module to themize maps on the ULTRON platform based on interpolation methods. The research is governed by the AUP-UCI methodology in scenario four: user history. For development of this module, several tools and technologies were selected such as the Angular framework, the OpenLayers library for the interaction whit the map, and for the backend the Loopback framework based on NodeJS and the PostgreSQL database manager were used. The test applied to the module produced satisfactory results. Black box tests were performed with the equivalence partition technique and white box tests were performed with the unit tests using Jasmine as a testing tool. The application allows the best visualization of geographical concentration through interpolation methods.

KEYWORDS

Application, Geographic Information System, Interpolation, Module, Theming.

TABLA DE CONTENIDOS

INTRODUCCIÓN	12
CAPÍTULO I: FUNDAMENTOS TEÓRICOS-METODOLÓGICOS SOBRE LOS PROCESOS DE VISUALIZACIÓN DE DATOS EN LAS TEMATIZACIÓN DE MAPAS.....	16
1.1 Tematización basada en métodos de interpolación.....	16
1.1.1 Caracterización de los procesos de visualización de datos en la tematización de mapas.....	17
1.1.2 Descripción del proceso de interpolación.....	18
1.1.3 Caracterización de los principales métodos de interpolación.	18
1.2 Análisis de los sistemas homólogos.....	26
1.2.1 Análisis de los sistemas homólogos en Cuba.	26
1.2.1.1 Cartografía de variables climáticas basadas en gradientes, sistema de expertos y SIG utilizado en el Centro de Meteorología Provincial de Guantánamo.	26
1.2.2 Análisis de los sistemas homólogos en el mundo.....	26
1.2.2.1 OpenLayers.	26
1.2.2.2 Quantum GIS (QGIS).	27
1.3 Metodología de desarrollo de software.	27
1.4 Herramientas y tecnologías.....	29
1.4.1 OpenLayers.	29
1.4.2 Visual Paradigm.	29
1.4.3 Angular.	30
1.4.3.1 Angular Cli.	30
1.4.4 TypeScript.	30
1.4.5 PostgreSQL.	30
1.4.6 Angular Material.	31
Conclusiones del capítulo.....	31
CAPÍTULO 2: DISEÑO DE LA SOLUCIÓN PROPUESTA AL PROBLEMA CIENTÍFICO.....	32
2.1 Modelo conceptual.	32
2.2 Levantamiento de requisitos.	33
2.2.1 Requisitos Funcionales.	34
2.2.2 Requisitos No Funcionales.....	34
2.3 Historias de Usuario.	35
2.4 Estilo Arquitectónico.	42
2.5 Patrones de Diseño.	43

2.5.1 Patrones GRAPS.....	44
2.5.2 Patrones GOF.	44
2.6 Diagrama de Clase del Diseño.....	45
2.7. Modelo de datos.	45
2.8 Modelo de despliegue.	46
Conclusiones del capítulo.....	47
CAPÍTULO 3: VALIDACIÓN DE LA SOLUCIÓN PROPUESTA.....	48
3.1 Modelo Implementación.	48
3.2 Diagrama de Componentes.	48
3.3 Estándares de Codificación.....	49
3.4 Pruebas	50
3.4.1 Pruebas de Caja Negra.	51
3.4.2 Pruebas Unitarias.	55
Conclusiones del capítulo.....	57
CONCLUSIONES FINALES.....	58
RECOMENDACIONES.....	59
REFERENCIAS BIBLIOGRÁFICAS.....	60

ÍNDICE DE TABLAS

<i>Tabla 1: Tabla comparativa entre los distintos métodos de interpolación seleccionados.</i>	<i>23</i>
<i>Tabla 2: HU Realizar tematización mediante el método IDW.</i>	<i>35</i>
<i>Tabla 3: HU Realizar tematización mediante el método interpolación lineal.....</i>	<i>36</i>
<i>Tabla 4: HU Adicionar mapa temático según interpolación.</i>	<i>37</i>
<i>Tabla 5: HU Editar la tematización según interpolación.</i>	<i>38</i>
<i>Tabla 6: HU Visualizar tematización según la interpolación.</i>	<i>39</i>
<i>Tabla 7: HU Visualizar tematización en el mapa.</i>	<i>40</i>
<i>Tabla 8: HU Eliminar tematización según la interpolación.</i>	<i>41</i>
<i>Tabla 9: HU Exportar imagen de la tematización por interpolación.</i>	<i>42</i>

ÍNDICE DE FIGURAS

<i>Figura 1: Modelo conceptual.</i>	33
<i>Figura 2: Realizar tematización mediante el método IDW</i>	36
<i>Figura 3: Realizar tematización mediante el método interpolación lineal.</i>	37
<i>Figura 4: Adicionar mapa temático según interpolación.</i>	38
<i>Figura 5: Editar la tematización según interpolación.</i>	39
<i>Figura 6: Visualizar la tematización según interpolación.</i>	40
<i>Figura 7: Visualizar tematización en el mapa.</i>	41
<i>Figura 8: Eliminar tematización según la interpolación.</i>	41
<i>Figura 9: Exportar imagen de la tematización por interpolación.</i>	42
<i>Figura 10: MVC.</i>	43
<i>Figura 11: Diagrama de clases del diseño.</i>	45
<i>Figura 12: Modelo de datos.</i>	46
<i>Figura 13: Diagrama de despliegue.</i>	47
<i>Figura 14: Diagrama de componentes.</i>	49
<i>Figura 15: Estándar de codificación: Identación.</i>	¡Error! Marcador no definido.
<i>Figura 15: Prueba unitaria con jasmine.</i>	56
<i>Figura 16: Prueba unitaria con jasmine.</i>	56

INTRODUCCIÓN

Las tecnologías de la información y las comunicaciones, han transformado nuestra manera de trabajar y gestionar recursos, son un elemento clave para hacer que nuestro trabajo sea más productivo; agilizando las comunicaciones, sustentando el trabajo en equipo, gestionando las existencias, realizando análisis financieros y promocionando nuestros productos en el mercado (Suarez & Custodia, 2014).

Las nuevas tecnologías han transformado la forma en que se recopilan, gestionan, analizan y comunican los datos espaciales a través de los Sistemas de Información Geográfica (SIG). Estas herramientas y recursos digitales han enriquecido la capacidad de los SIG para incorporar y aprovechar una variedad más amplia de datos espaciales permitiendo un análisis más profundo, una toma de decisiones más informada y una mayor difusión de información geográfica para una variedad de aplicaciones en todos los campos. Estos constituyen una plataforma tecnológica de utilidad para las agencias, empresas e instituciones que trabajan con estadísticas, ofreciéndole un *software* apropiado para la gestión y administración de una base de datos geográfica.

Entre las potencialidades brindadas por los SIG se encuentran las tematizaciones de mapas basados en métodos de interpolación. Estas están hechas para reflejar un aspecto particular de la zona geográfica sobre la que se definen. Pueden centrarse en variables físicas, sociales, políticas, culturales, económicas y cualquier otra relacionada con un territorio concreto. Las tematizaciones permiten entender la información asociada a cualquier evento socioeconómico en aras de hacer comparaciones y dar un criterio propio acerca del evento cartografiado. Un papel protagónico en esta área lo desempeña la Universidad de las Ciencias Informáticas (UCI) la que cuenta con el Centro de Representación y Análisis de Datos (CREAD).

Entre los productos desarrollados por el centro se encuentra la plataforma ULTRON, realizada íntegramente con tecnologías libres. Esta plataforma fue creada para el desarrollo de aplicaciones SIG en entornos web y cuenta con las funcionalidades comunes de estos sistemas, pudiendo ser personalizada como aplicación a la medida, incluyéndosele nuevas funcionalidades en caso de ser necesario.

La plataforma ULTRON cuenta en la actualidad con tres formas de realizar tematizaciones para el análisis de la información geográfica que se representa en el mapa: coropletas, símbolos proporcionales y gráficas dinámicas. A pesar que estas son muy utilizadas, presentan problemas que influyen en los resultados en dependencia del criterio de análisis sobre el que se esté trabajando.

Por ejemplo: las tematizaciones por coropletas se restringe a colorear polígonos, lo que implica que la tematización siempre seguirá los límites definidos en el mapa según sus características. Por lo tanto, no se puede representar un área formando diferentes zonas de colores basadas en datos socioeconómicos, lo cual dificulta el análisis de indicadores como la distribución de la población. No brindan el mismo nivel de detalle en áreas grandes que en pequeñas, lo que significa que, al tematizar un elemento en el mapa, como un municipio con mucha información, se muestra mucha información sobre los barrios y concejos populares. Sin embargo, en los municipios menos poblados, es poco común encontrar mapas que representen información sobre sus barrios y concejos populares.

Los símbolos proporcionales, al igual que otras representaciones temáticas, visualizan las características de un mapa. Sin embargo, a diferencia de las demás, utilizan símbolos significativos como círculos, cuadrados, triángulos o su combinación, lo que permite una vista en detalle. Su uso se basa en el tamaño de la variable tematizada, siendo adecuados para variables cuantitativos y valores absolutos.

Cuando la diferencia entre el valor mínimo y máximo de una variable es muy grande, algunos símbolos pueden resultar prácticamente imperceptibles, lo que hace que las entidades con poca información sean difíciles de distinguir frente a las más pobladas.

Es importante tener en cuenta que los mapas con coropletas y símbolos proporcionales puedan influir negativamente en la toma de decisiones y el análisis erróneo de situaciones específicas.

Luego de analizar la situación problemática descrita anteriormente, se plantea el siguiente **problema a resolver**: ¿Cómo mejorar la tematización de mapas mediante los métodos de interpolación en la plataforma ULTRON?

Definiéndose como **objeto de estudio**: Procesos de visualización de datos en la tematización de mapas.

Tomando como **campo de acción**: Procesos de visualización de datos en la tematización de mapas sobre la plataforma ULTRON.

Estableciéndose como **objetivo general**: Desarrollar un módulo para la plataforma ULTRON que permita mejorar la tematización de mapas mediante los métodos de interpolación.

Para darle cumplimiento al objetivo general se proponen las siguientes tareas investigativas:

- 1- Fundamentación de los referentes teóricos-metodológicos relacionados con la tematización de mapas basados en métodos de interpolación.
- 2- Desarrollo de la solución propuesta a partir de la metodología definida y el lenguaje de modelado.
- 3- Implementación de la solución propuesta.
- 4- Validación de la propuesta de solución.

Para el desarrollo de la investigación se emplearon varios **métodos científicos** que permiten tener una idea más detallada del resultado a obtener:

- **Histórico-Lógico**: Este método se usa para la elaboración de toda la teoría de forma lógica, con coherencia y de manera cronológica y permite estudiar la evolución de los conceptos asociados a los SIG, las tematizaciones y la interpolación espacial que permite la definición de términos propios.

- **Analítico-Sintético**: Este método se utiliza para la evaluación y análisis de los sistemas homólogos, las tecnologías, herramientas y metodología a utilizar que respondan al problema. Además, permite realizar una valoración crítica y detallada de cada una de ellas.

Métodos empíricos:

Entrevista: Se emplea en encuentros con el cliente y se realizan cuestionarios para definir las funcionalidades del componente de tematización, identificando a la vez particularidades necesarias para su desarrollo.

La presente investigación se desglosa en los siguientes capítulos:

Capítulo I: Se describen los conceptos por los cuales se rige el dominio de la investigación y se realiza un análisis de los componentes de tematización más conocidos nacional e internacionalmente. Se analizan las herramientas, metodologías y tecnologías utilizadas para la solución del problema planteado en la investigación. Además, se valoran los sistemas homólogos ya existentes que de alguna manera ofrecen respuesta al problema en cuestión.

Capítulo II: En este capítulo se diseña la propuesta de solución según la metodología empleada. Se describen los requisitos funcionales y no funcionales con las historias de usuarios de los requerimientos que debe cumplir el módulo de tematización, se define el diseño de la arquitectura, los patrones del diseño, el diagrama de clase del diseño, modelo de datos y el diagrama de despliegue.

Capítulo III: En este capítulo se abordan los aspectos relacionados con la construcción de la aplicación, dentro del cual se destacan los procesos de implementación y de prueba para su validación.

CAPÍTULO I: FUNDAMENTOS TEÓRICOS-METODOLÓGICOS SOBRE LOS PROCESOS DE VISUALIZACIÓN DE DATOS EN LAS TEMATIZACIÓN DE MAPAS.

Para lograr una mayor comprensión del alcance de la investigación, en el presente capítulo se exponen y relacionan los conceptos que desde el punto de vista teórico permiten un mejor entendimiento de lo planteado en la situación problemática. También se estudiarán las tecnologías y herramientas que se emplearán para dar cumplimiento al objetivo general de la investigación, así como la metodología utilizada para el desarrollo del módulo de tematización de mapas basado en métodos de interpolación.

1.1 Tematización basada en métodos de interpolación

Análisis Espacial

El análisis espacial permite resolver problemas complejos orientados a la ubicación, exploración y entendimiento de datos desde una perspectiva geográfica. El objetivo principal es determinar si una zona cumple con los requisitos para realizar determinadas actividades y permite detectar cambios, evaluar riesgos, identificar tendencias y prevenir pérdidas (Fuenzalida et al., 2018).

Métodos de Interpolación

El método de interpolación es un método científico lógico que consiste en determinar cada una de las variables en las formas en las que se pueden reproducir y cómo afectan al resultado. Pero no sólo basándose en su relación estadística sino también en su causalidad. Esto constituye las reglas que se utilizan para llegar a una nueva conclusión, siempre de forma aproximada. Es decir, se considera todas las situaciones posibles y sus repercusiones y las interpolamos a la nueva situación por analogía o inducción (Vera, 2012).

Coropletas

Los mapas de coropletas son una forma de cartografiado cuantitativo utilizada para la representación de fenómenos discretos asociados a unidades de enumeración (provincias, municipios, países...), a las que se le aplican símbolos superficiales de acuerdo con su valor. Es simplemente una representación planimétrica de un modelo de datos tridimensional, en

donde se consideran superficies y no puntos. En estos mapas se representan normalmente datos ya clasificados mediante intervalos (López & Abascal, 2022).

Símbolos Proporcionales

Los símbolos proporcionales son muy utilizados en el campo de la cartografía. Se basan fundamentalmente en seleccionar una forma (círculo, cuadrado, triángulo) y se varía su tamaño de un punto a otro. Esto nos conduce a una representación de fácil comprensión por parte de usuarios porque la asociación de cantidades a los tamaños resulta muy intuitiva. La representación de datos referentes a la población y los datos económicos se vale muy a menudo de esta técnica (García, 2022).

Interpolación Espacial

La interpolación espacial es un proceso que utiliza mediciones realizadas sobre algún fenómeno (precipitación, temperatura o elevación) en determinados lugares, para hacer una predicción sobre un fenómeno en otros lugares donde no se han realizado mediciones (Beltrán, 2020).

La interpolación espacial es una técnica utilizada en el análisis de datos geoespaciales para estimar valores desconocidos en ubicaciones no muestreadas dentro de un área. Esta técnica se basa en el principio de que los datos en una ubicación están relacionados de alguna manera con los datos en ubicaciones cercanas (Hernández et al., 2023).

Es una herramienta importante en el análisis de datos geoespaciales que permite estimar valores desconocidos en ubicaciones no muestreadas dentro de un área, generando mapas o superficies continuas que representan variaciones espaciales de una variable de interés.

1.1.1 Caracterización de los procesos de visualización de datos en la tematización de mapas

El análisis espacial es el proceso de manipular datos espaciales para extraer información nueva y significativa a partir de los datos originales. Usualmente el análisis espacial se realiza con un Sistema de Información Geográfica y normalmente proporciona herramientas de análisis espacial para calcular estadísticas de las entidades y realizar actividades de geoprocetamiento como la interpolación de datos. La tecnología de estos sistemas ha

posibilitado la obtención de resultados cartográficos y numéricos de suma utilidad en el ámbito de las ciencias geográficas.

Cuando se refiere a interpolación espacial se puede definir como la estimación de los valores que alcanza una variable Z en un conjunto de puntos definidos por un par de coordenadas (X , Y). Los SIG emplean la interpolación espacial para obtener capas que constituyen la variable a interpolar. Existen diversos métodos de interpolación, entre los más usados están el método ADW, IDW, Vecino más cercano, TIN, SPLINE, entre otros. Todos los métodos de interpolación se basan en la semejanza que tomarán los valores de cualquier variable cuantitativa que se midan en dos puntos, mientras más cercanos estén estos. No se puede afirmar que un método es superior a otro, ya que cada uno cumple una función determinada. El empleo de un método en específico está en dependencia de la situación espacial existente (Silva et al., 2020).

1.1.2 Descripción del proceso de interpolación

El objetivo fundamental del proceso de interpolación consiste en aproximar un proceso más complejo a partir de uno más simple. La interpolación espacial se centra en el análisis y simulación de una muestra de datos, así como su comportamiento en el espacio e influencia en otros puntos para así describir de forma más fácil el comportamiento o estructura de un atributo sobre el espacio, para lo cual se determina el grado de dependencia espacial que presentan las observaciones en relación con sus vecinos y la continuidad espacial de la superficie.

Finalmente, por medio del análisis espacial es posible detectar problemas de representatividad en las muestras, es decir, sectores con menor densidad de puntos respecto a otros, valores anómalos o existencia de grupos o subpoblaciones y tendencias en los datos. Todos estos aspectos están relacionados con el control antes de la aplicación de cualquier método de interpolación (LONDOÑO, 2021).

1.1.3 Caracterización de los principales métodos de interpolación

Nearest Neighbor (Vecino más cercano)

La interpolación mediante el vecino más cercano se basa en la generación de polígonos de Voronoi. Los polígonos de Voronoi constituyen el método más básico y simple de interpolación

vectorial. El método se basa únicamente en la distancia euclidiana, obviando cualquier tipo de valor asignado a los puntos de muestreo. Este algoritmo es seleccionar un valor de K , al momento del análisis los K datos más cercanos al valor que se desea predecir será la solución. Lo importante es seleccionar un valor de K acorde con los datos para obtener una mayor precisión en la predicción.

Se fija un valor para k , habitualmente pequeño, y se hace que el algoritmo compute una instancia D del juego de datos de prueba. Fruto de este proceso, el algoritmo selecciona las k instancias del juego de datos de entrenamiento más cercanas (de acuerdo con la métrica de similitud utilizada) y se asigna la instancia D a la clase más frecuente de entre las K instancias seleccionadas como más cercanas.

Se trata de un algoritmo muy simple pero con una buena efectividad pero su mayor debilidad es la lentitud en el proceso de clasificación puesto que su objetivo no es obtener un modelo optimizado, sino que cada instancia de pruebas es comparada contra todo el juego de datos de entrenamiento y, será la bondad de los resultados lo que determinara el ajuste de aspectos del algoritmo como el propio valor k , el criterio de selección de instancias para formar parte del juego de datos D de entrenamiento o la propia métrica de medida de similitud (Roncero, 2019).

Interpolación IDW (Ponderación de Distancia Inversa)

Es un método matemático de interpolación que asume que cada punto medido tiene una influencia local que disminuye con la distancia. Les asigna una ponderación mayor a los puntos más cercanos a la posición por predecir, que a aquellos que se encuentran más alejados, de ahí el nombre de distancia inversa ponderada. La ventaja de IDW es que es intuitivo y eficiente. Esta interpolación funciona mejor con puntos distribuidos uniformemente.

Este interpolador acepta que cada punto en el set de datos tiene una influencia local que disminuye con la distancia y que por lo tanto los valores de los puntos cercanos al nodo que se procesa tienen mayor importancia o peso en el valor que será asignado al mismo. Normalmente, la búsqueda se hace considerando un número de puntos o un radio (círculo alrededor de la celda de interés).

El peso asignado a cada valor de Z' está dado por:

$$P = \frac{1}{(d)^m}$$

P = peso o ponderación que se aplicará a Z;

m: exponente seleccionado por el usuario(a).

d: distancia entre el punto que se interpola y los vecinos más cercanos utilizados en la interpolación (Murillo et al., 2012).

Entre las ventajas de este método esta que es relativamente fácil de entender y aplicar y es útil para crear superficies suaves a partir de datos dispersos. También permite ajustar el radio de influencia para obtener diferentes grados de suavidad en la superficie interpolada. Sin embargo, entre sus principales desventajas están que es sensible a valores extremos en el conjunto de datos original lo que puede afectar significativamente la precisión de la superficie interpolada y dicha superficie puede hacer que los valores se extiendan más allá del rango de los datos originales, lo que puede ser un problema en ciertas situaciones.

TIN (Redes Irregulares de Triángulos)

Este método de interpolación devuelve una superficie de triángulos formada a partir de la localización de una serie de vértices cuyos valores son conocidos. Los vértices se conectan mediante aristas para generar dicha red triangular. El resultado obtenido, la superficie TIN, es una malla o red de triángulos interconectados, donde cada uno de ellos representa una zona homogénea en lo que a la variable estudiada se refiere. El método TIN tratara, por tanto, de generar un conjunto de triángulos sobre el espacio que maximicen la relación área/perímetro.

Es muy habitual su uso sobre todo para modelos de terreno en base a mediciones de elevación conocida, aunque puede aplicarse a otras mediciones cuantitativas de distintas variables ambientales (Fallas, 2007).

Este método tiene a su favor ser un método muy preciso, especialmente para áreas con muchos cambios en la elevación. También puede generar superficies interpoladas con alta resolución espacial, pero tiene en su contra que es muy complejo y difícil de entender que otros métodos de interpolación más simples y requiere de una gran cantidad de datos de entrada para generar una superficie interpolada precisa. Además, que puede ser computacionalmente costoso.

Interpolación mediante Spline Cúbico

La herramienta Spline utiliza un método de interpolación por splines cúbicos que estima valores usando una función polinómica que minimiza la curvatura general de la superficie, lo que resulta en una superficie suave que pasa exactamente por los puntos de entrada. Es uno de los métodos de interpolación existentes que no admite aproximaciones o suavizados de los valores de entradas.

Los splines generan curvas suaves que pasan por cada punto de datos, lo que lo hace muy útil para ajustar datos experimentales o mediciones ruidosas. Entre sus ventajas se encuentran su flexibilidad y bajo error de interpolar datos dispersos, esto es, que proporciona buenos resultados en la mayoría de las circunstancias. Sin embargo, su principal desventaja puede ser que es propenso al sobreajuste (error de interpolación) en datos con muchas oscilaciones o ruido (Chica, 2018).

ADW (Distancia Angular Ponderada)

El método de interpolación ADW es una modificación del IDW (Ponderación de Distancia Inversa) en la cual se incorpora una componente angular a los pesos para tener en cuenta la distribución irregular de las observaciones. Para evitar asignar más peso a sectores con mayor densidad de observaciones se introduce una componente angular en los pesos que 'premia' las observaciones bien distribuidas en el espacio. Los pesos tienen por tanto una componente radial y otra angular (Herrera, 2007).

Este método de interpolación es útil si no se tienen datos grandes a distancias del punto a interpolar. Su principal ventaja es su simplicidad y ahorro de tiempo, especialmente en grandes sistemas de datos. Sin embargo, uno de sus aspectos desventajosos es que tiene una baja precisión para datos que no siguen un patrón de variabilidad espacial regular.

KRIGING

Kriging es un método de inferencia espacial, el cual permite estimar los valores de una variable en lugares no muestreados utilizando la información proporcionada por la muestra. El método está planteado de manera que ofrece el mejor estimador lineal no sesgado con una varianza mínima. Es un método de interpolación geoestadístico de estimación de puntos. Utiliza un

modelo de variograma para la obtención de los ponderadores que se dan a cada punto de referencia usando la estimación. Esta técnica se basa en la premisa de que la variación espacial continua con un mismo patrón homogéneo.

Tipos de KRIGING

KRIGING Simple: Asume que las medidas locales son relativamente constantes y de valor muy semejante a la medida de la población que es conocida. La medida de la población es utilizada para cada estimación local, en conjunto con los puntos vecinos establecidos como necesarios para la estimación.

KRIGING Ordinario: Las medidas locales no son necesariamente próximas a la medida de la población, usándose apenas los puntos vecinos para la estimación. Es el método más ampliamente utilizado en los problemas ambientales.

Cokriging: Es una extensión de las situaciones anteriores en las que dos o más variables tienen una dependencia espacial y esa variable se estima que no se muestra con la intensidad con la que otras variables dependientes, con estos valores y sus dependencias para estimar la variable que se requiere (Vega-Blancas et al., 2022).

Entre las ventajas de este método se encuentra su capacidad para modelar la variabilidad espacial de manera más precisa que otros métodos y la producción de estimaciones imparciales con menor varianza. Por otro lado, su principal desventaja es que requiere el conocimiento de ciertos parámetros de los datos como la varianza, lo que hace que el proceso sea más complejo y dependa considerablemente de la correcta especificación de los modelos utilizados.

Interpolación Lineal

La interpolación lineal permite determinar el conjunto de valores que toma una función en un determinado intervalo dados dos puntos o valores iniciales. Esta es la forma más sencilla de interpolación existente. Usa dos puntos con el fin de desarrollar una aproximación lineal de la función. Estos dos puntos usados en el proceso serán los más próximos al punto de interés, debiendo ser uno menor y otro mayor que este. Por tanto, entre cada pareja de valores de (x_k, y_k) y (x_{k+1}, y_{k+1}) , se calcula la recta existente entre ambos para calcular y al valor de x dado.

La interpolación lineal es un método relativamente sencillo que se basa en la suposición de que la tasa de cambio entre los valores conocidos es constante y puede calcularse a partir de estos valores utilizando la fórmula de la pendiente simple. Entonces, un valor desconocido entre los dos puntos conocidos se puede calcular usando uno de los puntos y la tasa de cambio. Proporciona resultados de forma rápida y sencilla, debido a que solo realiza el cálculo de trayectoria de dos puntos.

Este método puede ser adecuado para datos con una estructura simple como las mediciones en un tiempo uniforme. Entre sus ventajas se encuentran su simplicidad y rapidez de aplicación, pero su principal desventaja es su escasa precisión en datos con mayor variabilidad espacial (Roncero, 2019).

En general, no existe un método de interpolación que sea el mejor para todas las situaciones. Es importante evaluar las características de los datos, la estructura espacial y la precisión que se requiere para seleccionar el método más adecuado.

Tabla 1: Tabla comparativa entre los distintos métodos de interpolación seleccionados.

Método de Interpolación	Descripción	Mayor Utilidad	Ventajas	Desventajas
Nearest Neighbor	Utiliza los valores del punto de datos más cercano para estimar el valor en el punto que se desea interpolar.	Interpolación de datos discretos y espaciados irregularmente	Es rápido y fácil de implementar	Puede producir resultados artificiales, especialmente en áreas donde la densidad de los puntos de datos es baja
Inverse Distance	Este método utiliza un peso inverso a la	Interpolación de datos	El método es matemáticamente sencillo y fácil de	Puede producir resultados artificiales en áreas

Weighted (IDW)	distancia de los puntos de datos más cercano al punto que se desea interpolar. Los pasos se suman para generar una estimación de la superficie	espaciados irregularmente	entender. Además, el resultado se basa en la distancia del punto de datos más cercano, lo que puede proporcionar una buena representación de la superficie.	donde la densidad de los puntos de datos es baja
Triangulated Irregular Network (TIN)	Utiliza triángulos creados a partir de los puntos de datos para estimar la superficie interpolada.	Interpolación de datos espaciados irregularmente y para la generación de modelos 3D	Produce resultados precisos y suaves. Se adapta muy bien a la topografía irregular.	Es un método computacionalmente intensivo y puede ser costoso en términos de tiempo de procesamiento.
Spline Cúbico	Utiliza una función matemática para suavizar la superficie y producir una interpolación suave	Es útil para la interpolación de datos con grandes agujeros o valores faltantes.	Produce resultados suaves y precisos	Puede sufrir sobre suavizado y puede ser costoso en términos de tiempo de procesamiento.

Adaptative Distance Weighted (ADW)	Utiliza una estrategia de selección de vecino adaptativa para aumentar la interpolación en áreas cercanas a la discontinuidad de los datos.	Es adecuado para la interpolación de datos con un alto grado de variabilidad	Crea una interpolación suave en áreas cercanas a las discontinuidades de los datos.	Puede ser muy costoso en tiempos de procesamiento estándar.
KRIGING	Utiliza modelos estadísticos variogramas para estimar la superficie asociadas con la estimación.	Interpolación de datos de datos de la temperatura del suelo, la calidad del agua y otras aplicaciones ambientales.	Es capaz de modelar y cuantificar la incertidumbre y produce una interpolación suave y precisa	Es un método computacionalmente intensivo y requiere una gran cantidad de RAM.
Interpolación Lineal	Este método conecta los puntos de datos más cercano en línea recta y estima el valor a interpolar en función de la	Es utilizado para la interpolación de datos discretos en áreas pequeñas	Es rápido y fácil de implementar	Puede producir resultados artificiales.

	distancia ponderada.			
--	-------------------------	--	--	--

Se escogieron los métodos Interpolación lineal y el método IDW porque son 2 de los métodos de interpolación más usados debido a sus ventajas y flexibilidad en distintos escenarios. Son métodos conceptualmente simples y son más resistente a los valores atípicos Además son más rápidos y fáciles de implementar.

1.2 Análisis de los sistemas homólogos

En el desarrollo de esta investigación se analizaron varios sistemas creados en Cuba y el mundo, que utilizan la interpolación espacial para la representación de los resultados obtenidos. Los sistemas analizados son:

1.2.1 Análisis de los sistemas homólogos en Cuba

1.2.1.1 Cartografía de variables climáticas basadas en gradientes, sistema de expertos y SIG utilizado en el Centro de Meteorología Provincial de Guantánamo

Este método está basado en la aplicación de funciones SIG para modelar la distribución espacial de los factores formadores del clima a nivel local y cartografía de las variables climáticas, técnicas de interpolación espacial para la creación de mapas continuos de variables y un sistema experto basado en un motor de interferencia difusa para asignar gradientes y referencias. Se utilizó el software ArcGis v.10.2.2 para la gestión de y manipulación de datos espaciales y el software Matlab 13b para el sistema de interferencias (Delgado-Téllez & PEÑA-DE LA CRUZ, 2019).

1.2.2 Análisis de los sistemas homólogos en el mundo

1.2.2.1 OpenLayers

OpenLayers al ser una librería del lado del cliente, es un visor de mapas en JavaScript, la descarga de estos se realiza directamente desde el navegador a través de Ajax, no genera tráfico en el servidor, los mapas se descargan directamente del servidor de mapas que suelen ser una pieza diferenciada a nivel de sistemas.

OpenLayers facilita la colocación de un mapa dinámico en cualquier página web. Puede mostrar mosaicos de mapas, datos vectoriales y marcadores cargados desde cualquier fuente. OpenLayers ha sido desarrollado para promover el uso de información geográfica de todo tipo. Es JavaScript de código abierto completamente gratuito.

En el contexto de OpenLayers, la interpolación se puede lograr utilizando la extensión *ol-ext* y su capa IDW (Inverse Distance Weighting).

El estudio de los sistemas homólogos en la interpolación aporta una comprensión más profunda de cómo funcionan los sistemas y como se pueden extrapolar para obtener resultados más precisos. Cuando se analizan varios sistemas homólogos se identifican patrones y tendencias que se pueden utilizar para hacer interpolaciones con mayor precisión. Es un enfoque valioso que permite mejorar la precisión de las predicciones y la comprensión de los sistemas (Yuquin et al., 2019).

1.2.2.2 Quantum GIS (QGIS)

QGIS es un Sistema de Información Geográfica libre y de código abierto, que cuenta con una interfaz sencilla, intuitiva y amigable, con una comunidad amplia de usuarios y desarrolladores, permitiendo modificarse y adaptarse a nuestras necesidades. En QGIS, la interpolación se consigue usando el complemento interpolación Ponderación de Distancia Inversa (IDW) que ya viene incorporado.

Los resultados de la interpolación pueden variar significativamente dependiendo del método y parámetros que elige el usuario. La interpolación en QGIS permite los métodos Red Irregular Triangulada (TIN) y Peso Distancia Inversa (IDW) para la interpolación. El método TIN es comúnmente usado para datos de elevación mientras que el método IDW es usado para interpolar otros tipos de datos como concentraciones minerales, poblaciones, entre otros (Manrique López, 2017).

1.3 Metodología de desarrollo de software

Durante la creación de un *software* es necesario utilizar una metodología que se encargue de guiar el proceso de desarrollo; como todos los procesos no son iguales, no existe una única metodología, sino varias que se ajustan a las características de cada producto que se realice.

Existen metodologías ágiles y tradicionales. Las metodologías tradicionales pueden ser empleadas para guiar el proceso de desarrollo de proyectos grandes o pequeños, aunque son más apropiadas para proyectos grandes que por su importancia requieren una fuerte planificación. Las metodologías ágiles son apropiadas para guiar proyectos de poco volumen que requieran una rápida implementación (Brito et al., 2019).

La UCI desarrolló una versión de la metodología de desarrollo de *software* AUP (Proceso Ágil Unificado), con el fin de crear una metodología que se adapte al ciclo de vida definido por la actividad productiva de la universidad. Esta versión decide mantener para el ciclo de vida de los proyectos la fase de Inicio, pero modificando el objetivo de la misma y se unifican las restantes fases de la metodología de desarrollo de *software* AUP en una sola, nombrada Ejecución y agregándose también una nueva fase llamada Cierre.

La metodología de *software* AUP-UCI a partir de que el modelado de negocio propone tres variantes a utilizar en los proyectos, como son: CUN (Casos de uso del negocio), DPN (Descripción de proceso de negocio) o MC (Modelo conceptual) y existen tres formas de encapsular los requisitos, los cuales son: CUS (Casos de uso del sistema), HU (Historias de usuario), DRP (Descripción de requisitos por proceso), surgen cuatro escenarios para modelar el sistema en los proyectos, los cuales son:

- Escenario No 1: Proyectos que modelen el negocio con CUN solo pueden modelar el sistema con CUS.
- Escenario No 2: Proyectos que modelen el negocio con MC solo pueden modelar el sistema con CUS.
- Escenario No 3: Proyectos que modelen el negocio con DPN solo pueden modelar el sistema con DRP.
- Escenario No 4: Proyectos que no modelen negocio solo pueden modelar el sistema con HU.

A partir del análisis e investigación efectuada, la metodología AUP-UCI es apropiada para proyectos pequeños, permitiendo disminuir las probabilidades de fracaso, por ser un producto de fácil uso utilizando cualquier herramienta. Con la adaptación de AUP que se propone para

la actividad productiva de la UCI se logra estandarizar el proceso de desarrollo de *software*. Se logra hablar un lenguaje común en cuanto a fases, disciplinas, roles y productos de trabajos. Se seleccionó la metodología AUP-UCI en su escenario 4 porque emplea un equipo de desarrollo pequeño, en este caso un desarrollador, los requisitos del sistema están bien definidos y el cliente va a estar en todo el proceso de desarrollo de la aplicación.

1.4 Herramientas y tecnologías

1.4.1 OpenLayers

OpenLayers es una biblioteca de mapas web JavaScript de código abierto para crear aplicaciones de mapas web, es una biblioteca modular de alto rendimiento y con una amplia variedad de funciones para mostrar e interactuar con mapas y datos geoespaciales.

Permite la creación de mapas dinámicos en cualquier página web. Puede mostrar mosaicos de mapas, datos vectoriales y marcadores cargados desde cualquier fuente. OpenLayers ha sido desarrollado para promover el uso de información geográfica de todo tipo. Es completamente gratuito, de código abierto, publicado bajo la licencia Berkeley Software Distribution (por sus siglas en inglés BSD) de dos cláusulas, también conocida como FreeBSD.

Este se ejecuta en todos los navegadores modernos que admiten HTML5 y ECMAScript 5; la biblioteca está pensada para usarse tanto en computadoras de escritorio/portátiles como en dispositivos móviles, y admite interacciones táctiles y de puntero (Yuquin et al., 2019).

1.4.2 Visual Paradigm

Para el modelado se utiliza la herramienta Visual Paradigm en su versión 8.0. Es una herramienta CASE multiplataforma, que soporta el ciclo completo de desarrollo de *software*: análisis, diseño, implementación y pruebas. Ideal para ingenieros de *software*, analistas y arquitectos de sistemas, que están interesados en la construcción de sistemas a gran escala y necesitan confiabilidad y estabilidad en el desarrollo orientado a objetos. Permite la generación de bases de datos, conversión de diagramas entidad-relación a tablas de base de datos, mapeos de objetos y relaciones, ingeniería directa e inversa, la gestión de requisitos de *software* y la modelación de procesos del negocio (Mendoza & Baquero, 2016).

1.4.3 Angular

Angular es un *framework* para aplicaciones web. Desarrollado en TypeScript, de código abierto, mantenido por Google. Angular es un marco de diseño de aplicaciones y una plataforma de desarrollo para crear aplicaciones de una sola página eficientes y sofisticadas. Es una plataforma que puede escalar desde proyectos de un solo desarrollador hasta aplicaciones de nivel empresarial, está diseñada para que la actualización sea lo más sencilla posible, sirve tanto en versiones móviles como escritorio (Farinango, 2020).

1.4.3.1 Angular Cli

Herramienta para inicializar, es la forma más rápida y sencilla de desarrollar e incluso para mantener aplicaciones de Angular. También podemos ejecutar tareas de pruebas y realizar (Farinango, 2020).

1.4.4 TypeScript

TypeScript es un lenguaje de programación de código abierto creado por Microsoft en el año 2012, que implementa muchos mecanismos de programación orientada a objetos. Se trata de un superconjunto de JavaScript, lo que significa que extiende la sintaxis de JavaScript. Se puede emplear desde cualquier dispositivo, sistema operativo o navegador. Del mismo modo que JavaScript, no requiere de ninguna máquina específica. TypeScript se caracteriza por manejar diferentes tipos de datos, por lo que se trata de un lenguaje escalable y seguro debido a que se puede ir verificando la ejecución de su código (Puciarelli, 2020).

1.4.5 PostgreSQL

PostgreSQL es un sistema de GBD (Gestor de Base de Datos) objeto-relacional, de propósito general, multiusuario y de código abierto, que soporta gran parte del estándar SQL y ofrece modernas características como consultas complejas, disparadores, vistas, integridad transaccional y control de concurrencia multiversión. Puede ser extendido por el usuario añadiendo tipos de datos, operadores, funciones agregadas, funciones ventanas y funciones recursivas, métodos de indexado y lenguajes procedurales (Pilicita et al., 2021).

1.4.6 Angular Material

Angular Material es un módulo construido por y para Angular. Dicho módulo permite implementar componentes Angular con un diseño basado en Material Design. Básicamente nos ofrece una serie de componentes que enriquecerán mucho el catálogo de interfaces de usuarios disponibles al implementar una web.

Es una excelente opción para programadores, que no tienen habitualmente habilidades de diseño. Usando las directivas que nos ofrece podemos producir aplicaciones que son estéticamente brillantes, junto con una serie de componentes adicionales al propio HTML con un look muy trabajado (Guagalango, 2023).

Conclusiones del capítulo

Con el desarrollo de este capítulo se ha logrado definir de forma clara todos los aspectos teóricos y conceptuales necesarios para la realización de la investigación propuesta, se realizó el análisis y profundización de los conceptos relacionados con el objeto de estudio, lo que permitió conocer el entorno del sistema sobre el cual se trabajará. Además, el análisis y estudio de las soluciones existentes, la metodología de desarrollo, herramientas, tecnologías y lenguajes de programación permite especificar el ambiente de desarrollo para la propuesta de solución.

CAPÍTULO 2: DISEÑO DE LA SOLUCIÓN PROPUESTA AL PROBLEMA CIENTÍFICO.

Para lograr una mayor comprensión de la propuesta de solución, el presente capítulo tiene como objetivo mostrar los resultados que se obtienen una vez concluida la etapa diseño que propone la metodología AUP-UCI en el escenario 4. Entre los elementos a destacar se encuentran el diagrama del modelo de dominio, los requisitos funcionales y no funcionales y las historias de usuario. Como parte del diseño de la aplicación se definió el estilo y los patrones de arquitectura y diseño que se emplearán en el desarrollo de la propuesta de solución.

2.1 Modelo conceptual

Un modelo conceptual de una organización ofrece la posibilidad de visualizar los diferentes elementos que la conforman, la manera como se pueden organizar y la forma como se integra y articulan para atender las necesidades de los usuarios y de los servicios que ofrecen.

Su importancia radica en que permiten identificar, organizar y realizar razonamientos sobre los componentes y comportamiento del sistema, son la guía para el proceso de diseño del software y puede usarse posteriormente como una referencia para evaluar un diseño particular, razonar sobre la solución realizada y sobre el posible espacio de soluciones. Por su naturaleza deben ser expresivos, fáciles de usar y completos (Ledesma, 2018).

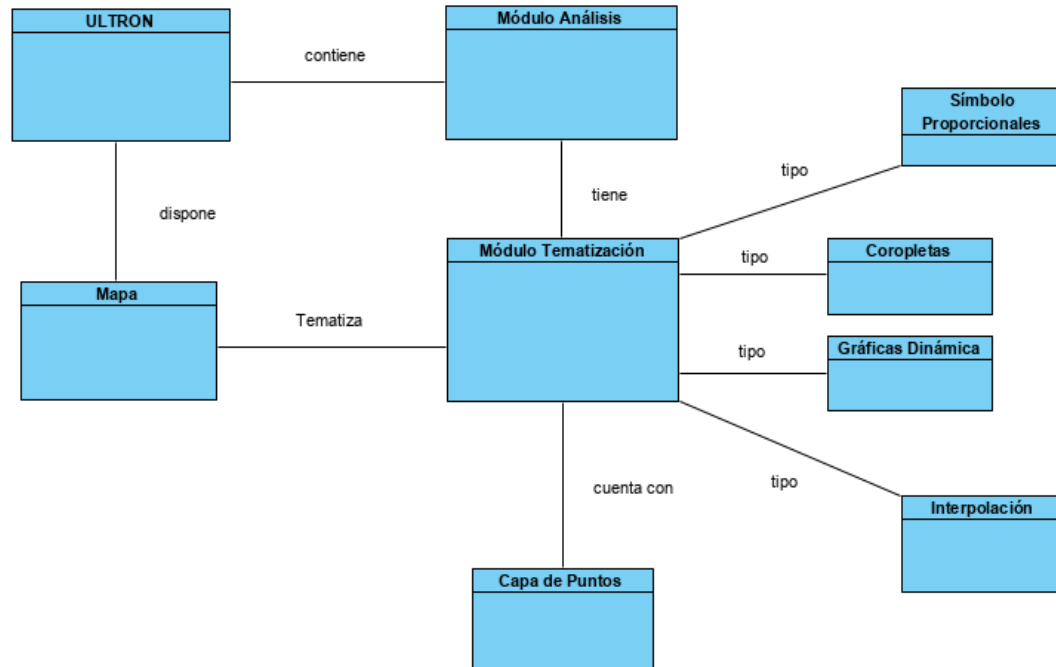


Figura 1: Modelo conceptual.

2.2 Levantamiento de requisitos

La ingeniería de requisitos se define como el uso sistemático de procedimientos, técnicas, lenguajes y herramientas para obtener con un coste reducido el análisis, la documentación, la evolución continua de las necesidades del usuario y la especificación del comportamiento externo de un sistema que satisfaga las necesidades del usuario. La ingeniería de *software* no se guía por conductas esporádicas, aleatorias o por modas pasajeras, sino que se debe basar en el uso sistemático de aproximaciones contratadas (Guevara & Torres, 2022).

Los requisitos del *software* no son más que una descripción general de cómo debe funcionar el sistema que se va a implementar. La correcta definición de los mismos es fundamental para lograr el éxito en el desarrollo de la solución propuesta. Para su mayor comprensión se clasifican en funcionales y no funcionales.

2.2.1 Requisitos Funcionales

Los requerimientos funcionales de un sistema, son aquellos que describen cualquier actividad que este deba realizar, en otras palabras, el comportamiento o función particular de un sistema o *software* cuando se cumplen ciertas condiciones (Del Águila, 2019).

A continuación, se muestran los RF identificados:

RF1. Realizar tematización mediante el método IDW.

RF2. Realizar tematización mediante el método interpolación lineal.

RF3. Adicionar mapa temático según la interpolación.

RF4. Editar la tematización según la interpolación.

RF5. Visualizar tematización según la interpolación.

RF6. Visualizar tematización en el mapa.

RF7. Eliminar tematización según la interpolación.

RF8. Exportar imagen de la tematización por interpolación.

2.2.2 Requisitos No Funcionales

Los requisitos no funcionales (RNF) son propiedades o cualidades que el sistema debe tener, restricciones y características del *software*. A diferencia de los funcionales, no determinan una funcionalidad del sistema a desarrollar. Los RNF se caracterizan por ser específicos, cuantificables y verificables (Del Águila, 2019).

A continuación, se muestran los RNF identificados:

Usabilidad

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

Interfaz de usuario y diseño

- La interfaz de usuario deberá tener **apariencia profesional y diseño gráfico sencillo**.

- Las funcionalidades a realizar deben representarse con íconos intuitivos.

Hardware

Para las PCs clientes:

- Se requiere que tengan tarjeta de red.
- Al menos 128Mb de memoria RAM.
- Procesador de 512 MHz como mínimo.

Para los servidores:

- Se requiere tarjeta de red.
- El servidor de mapas debe tener como mínimo 2 GB de RAM y 40 GB de disco duro.
- El servidor de base de datos debe tener como mínimo 2 GB de RAM y 40 GB de disco duro.
- Procesador de 3 GHz como mínimo.

Software

Un navegador web que cumpla con los estándares W3C y cuente con soporte para CCS3, HTML5 y JS.

2.3 Historias de Usuario.

Tabla 2: HU Realizar tematización mediante el método IDW.

Historia de usuario	
Número: HU_1	Nombre Historia de Usuario: Realizar tematización mediante el método IDW
Prioridad en negocio: Alta	
Descripción: Permitirá realizar la tematización mediante el método IDW.	
Observaciones:	
Prototipo:	

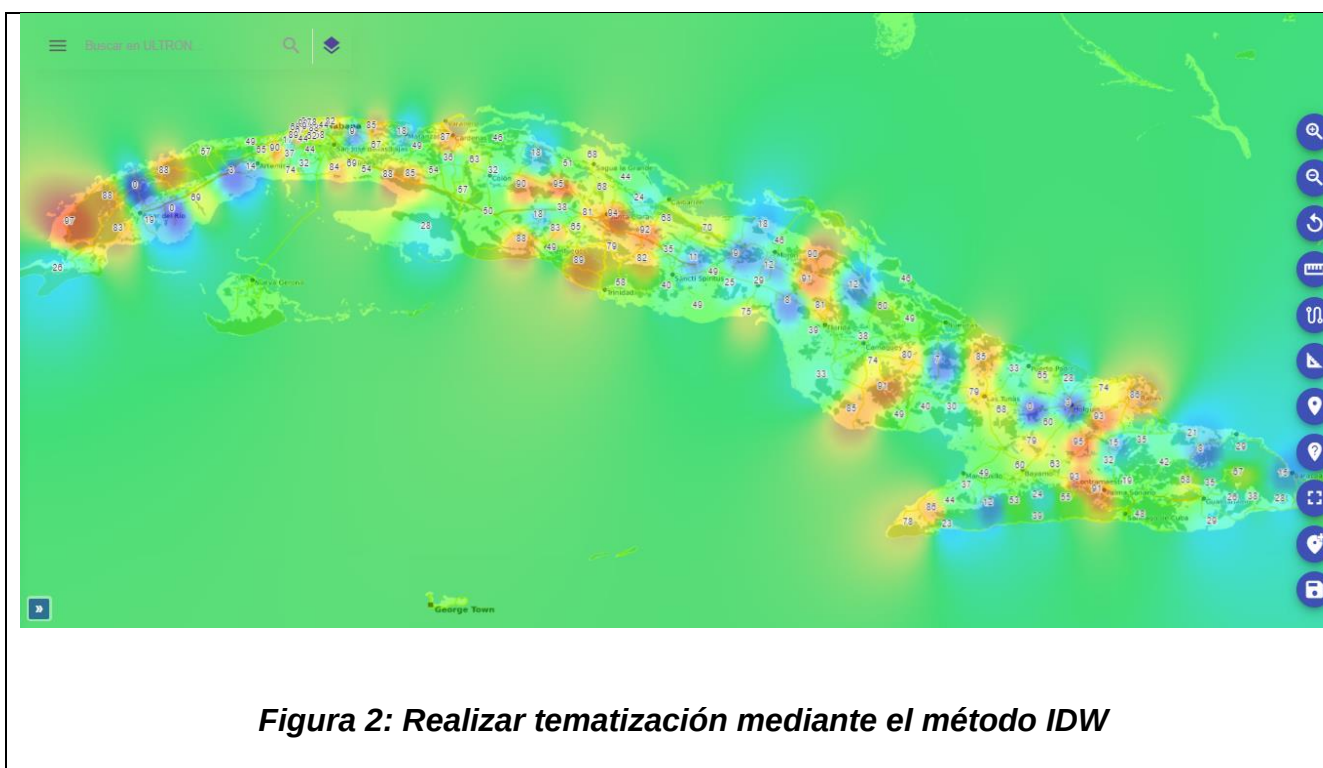


Tabla 3: HU Realizar tematización mediante el método interpolación lineal.

Historia de usuario	
Número: HU_2	Nombre Historia de Usuario: Realizar tematización mediante el método interpolación lineal.
Prioridad en negocio: Alta	
Descripción: Permitirá realizar la tematización mediante el método interpolación lineal.	
Observaciones:	
Prototipo:	

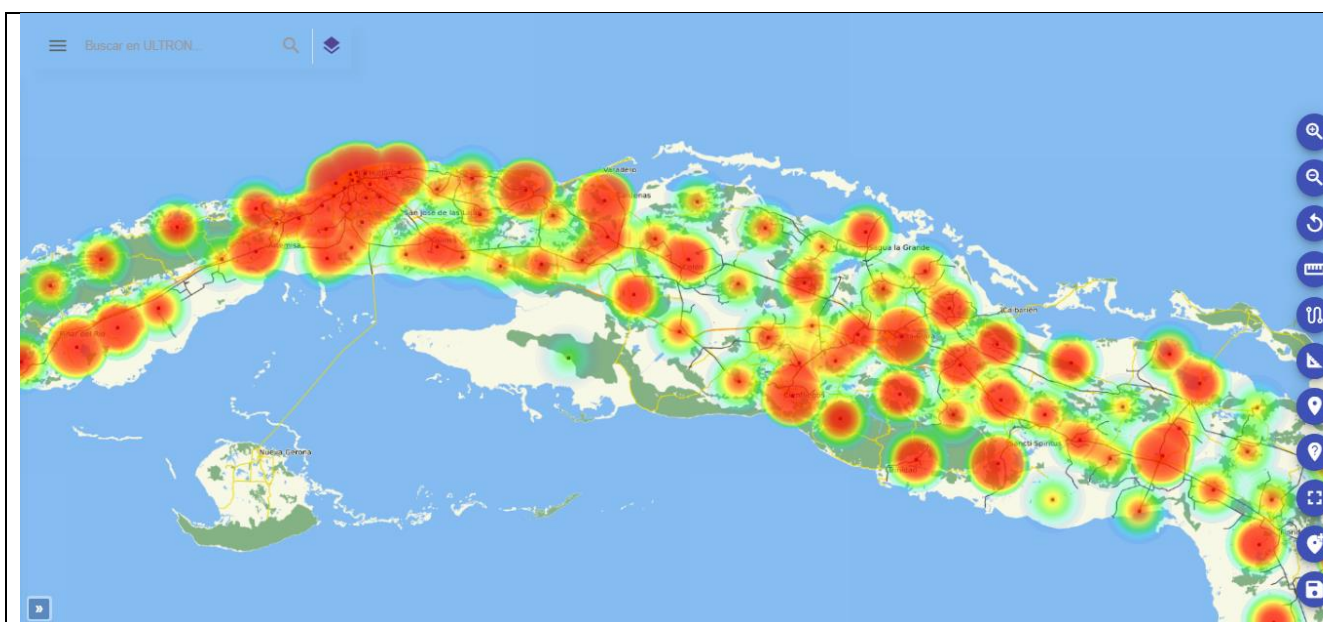


Figura 3: Realizar tematización mediante el método interpolación lineal.

Tabla 4: HU Adicionar mapa temático según interpolación.

Historia de usuario	
Número: HU_3	Nombre Historia de Usuario: Adicionar mapa temático según interpolación.
Prioridad en negocio: Alta	
Descripción: Permitirá adicionar mapa temático en el sistema, una a la vez.	
Observaciones:	
Prototipo:	

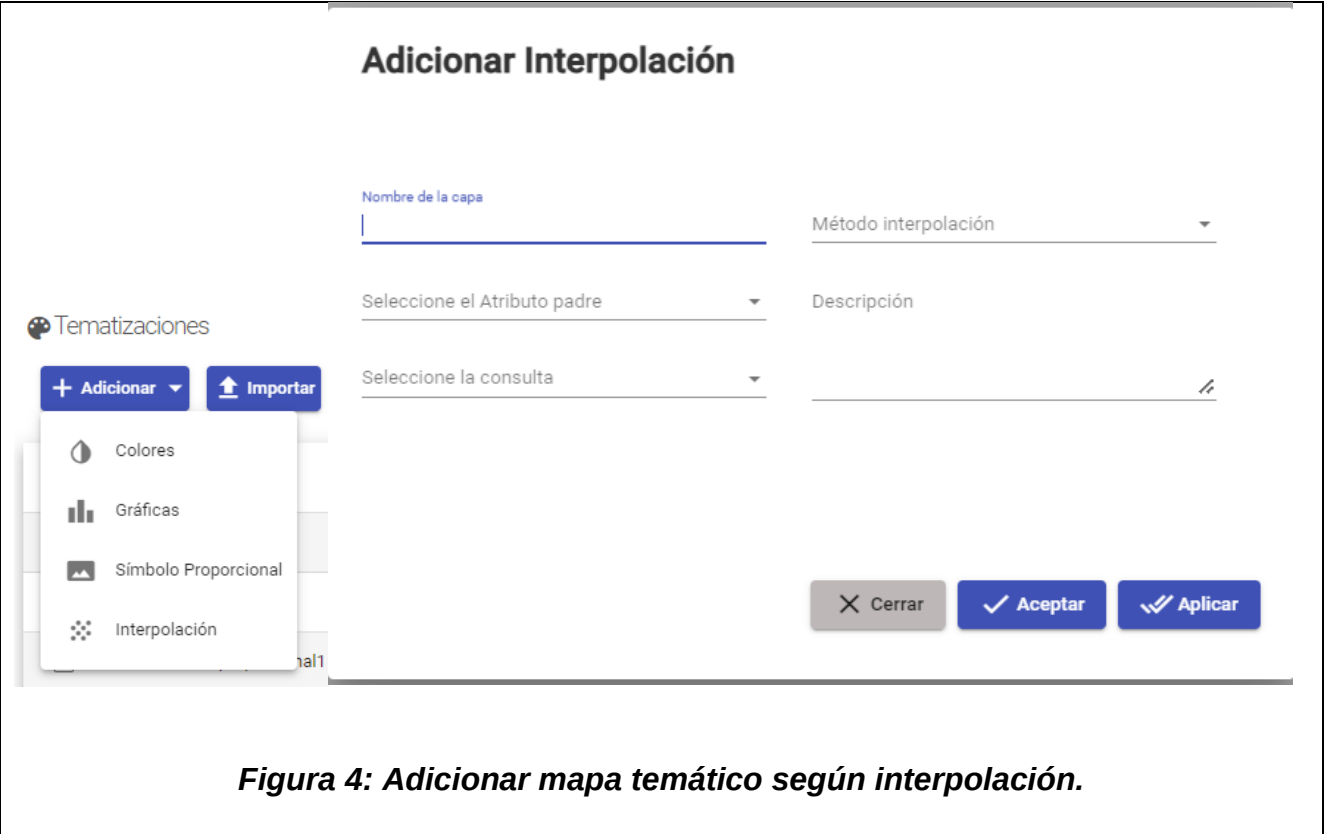


Figura 4: Adicionar mapa temático según interpolación.

Tabla 5: HU Editar la tematización según interpolación.

Historia de usuario	
Número: HU_4	Nombre Historia de Usuario: Editar la tematización según interpolación.
Prioridad en negocio: Alta	
Descripción: Permitirá modificar la información correspondiente a las tematizaciones existentes en el sistema, una a la vez.	
Observaciones:	
Prototipo:	

Editar Interpolación

Nombre de la capa Interpolate21	Método interpolación IDW
Seleccione el Atributo padre Indicador1	Descripción sadsdasd
Seleccione la consulta grafica x	

✕ Cerrar

✓ Aceptar

✓✓ Aplicar

Figura 5: Editar la tematización según interpolación.

Tabla 6: HU Visualizar tematización según la interpolación.

Historia de usuario	
Número: HU_5	Nombre Historia de Usuario: Visualizar tematización según la interpolación.
Prioridad en negocio: Alta	
Descripción: Permitirá visualizar un formulario no editable con la información correspondiente a la tematización.	
Observaciones:	
Prototipo:	

Visualizar Interpolación

Nombre de la capa

Método interpolación

Seleccione el Atributo padre

Descripción

Seleccione la consulta

X Cerrar

Figura 6: Visualizar la tematización según interpolación.

Tabla 7: HU Visualizar tematización en el mapa.

Historia de usuario	
Número: HU_6	Nombre Historia de Usuario: Visualizar tematización en el mapa.
Prioridad en negocio: Alta	
Descripción: Permitirá visualizar las tematizaciones en el mapa, una a la vez.	
Observaciones:	
Prototipo:	

Tematizaciones					
+ Adicionar Importar Eliminar Introduzca un criterio...					
☐	Id ↑	Nombre	Creado	Tipo	Acciones
☐	1	colort	2023-07-11T14:10:07.564Z	color	   
☐	2	grafica	2023-07-11T14:10:58.946Z	chart	   
☐	3	proporcional1	2023-07-11T14:13:11.195Z	symbol	   
Items per page: 10 1 - 3 of 3 < < > >					

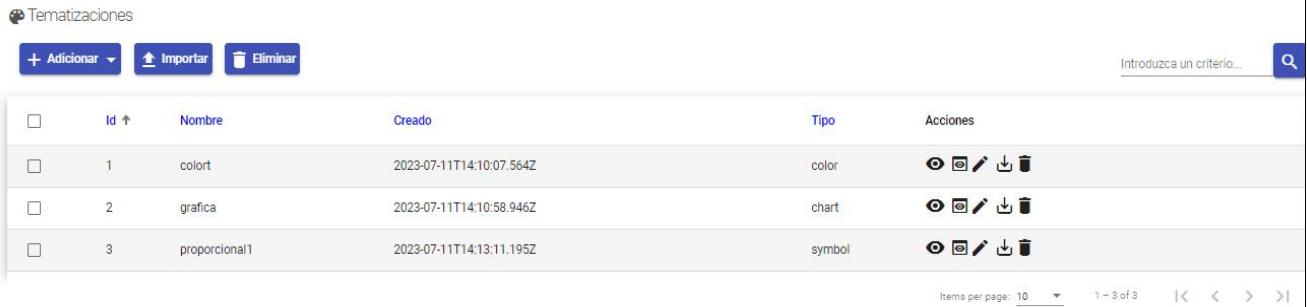
Figura 7: Visualizar tematización en el mapa.

Tabla 8: HU Eliminar tematización según la interpolación.

Historia de usuario																									
Número: HU_7	Nombre Historia de Usuario: Eliminar tematización según la interpolación.																								
Prioridad en negocio: Alta																									
Descripción: Permitirá eliminar una o varias tematizaciones previamente creadas por el usuario autenticado.																									
Observaciones:																									
Prototipo:																									
Tematizaciones + Adicionar Importar Eliminar Introduzca un criterio... <table><tr><th>☐</th><th>Id ↑</th><th>Nombre</th><th>Creado</th><th>Tipo</th><th>Acciones</th></tr><tr><td>☐</td><td>1</td><td>colort</td><td>2023-07-11T14:10:07.564Z</td><td>color</td><td> </td></tr><tr><td>☐</td><td>2</td><td>grafica</td><td>2023-07-11T14:10:58.946Z</td><td>chart</td><td> </td></tr><tr><td>☐</td><td>3</td><td>proporcional1</td><td>2023-07-11T14:13:11.195Z</td><td>symbol</td><td> </td></tr></table> Items per page: 10 1 - 3 of 3 < < > >		☐	Id ↑	Nombre	Creado	Tipo	Acciones	☐	1	colort	2023-07-11T14:10:07.564Z	color		☐	2	grafica	2023-07-11T14:10:58.946Z	chart		☐	3	proporcional1	2023-07-11T14:13:11.195Z	symbol	
☐	Id ↑	Nombre	Creado	Tipo	Acciones																				
☐	1	colort	2023-07-11T14:10:07.564Z	color																					
☐	2	grafica	2023-07-11T14:10:58.946Z	chart																					
☐	3	proporcional1	2023-07-11T14:13:11.195Z	symbol																					

Figura 8: Eliminar tematización según la interpolación.

Tabla 9: HU Exportar imagen de la tematización por interpolación.

Historia de usuario	
Número: HU_8	Nombre Historia de Usuario: Exportar imagen de la tematización por interpolación.
Prioridad en negocio: Alta	
Descripción: Permitirá exportar la tematización en un fichero, que podrá importarse con las configuraciones realizadas.	
Observaciones:	
Prototipo:  Figura 9: Exportar imagen de la tematización por interpolación.	

2.4 Estilo Arquitectónico

Modelo-Vista-Controlador (MVC) es un patrón de arquitectura de *software*, que separa los datos y principalmente lo que es la lógica del negocio de una aplicación de su representación y el módulo encargado de gestionar y las comunicaciones. Para ello MVC propone la construcción de tres componentes distintos que son el modelo, la vista y el controlador, es decir, por un lado, define los componentes para la representación de la información y por otro lado para la interacción del usuario.

Este patrón de arquitectura de *software* se basa en las ideas de reutilización de código y la separación de conceptos, características que buscan facilitar la tarea del desarrollo de aplicaciones y su posterior mantenimiento.

El **modelo** es la representación de la información con la cual el sistema opera, por lo tanto, gestiona todos los accesos a dicha información, tanto consultas como actualizaciones. Envía a la vista aquella parte de la información que en cada momento se le solicita para que sea mostrada.

La **vista** presenta el modelo (información y lógica del negocio) en un formato adecuado para interactuar (usualmente la interfaz de usuario), por tanto, requiere de dicho modelo la información que debe representar como salida.

El **controlador** responde a eventos (usualmente acciones del usuario) e invoca peticiones al modelo cuando se hace alguna solicitud sobre la información. También puede enviar comandos a su vista asociada si se solicita un cambio en la forma en que se representa el modelo, por tanto, se podría decir que el controlador hace de intermediario entre la vista y el modelo (Alonso, 2019).

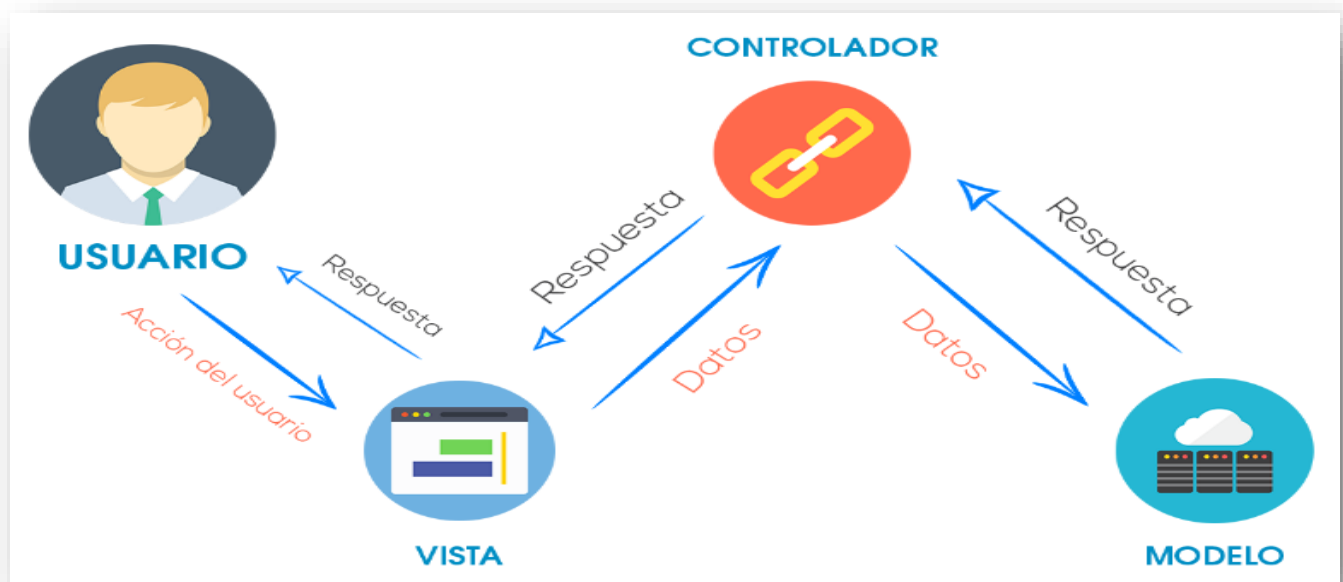


Figura 10: MVC.

2.5 Patrones de Diseño

Los patrones arquitectónicos pueden ser vistos como los bloques constructivos básicos de las arquitecturas de *software*. Estos definen y organizan un vocabulario de elementos basado en

un conjunto de componentes y conectores asociados a un grupo de restricciones que indican la forma en la cual tales elementos pueden ser combinados. El objetivo de este vocabulario es la formulación de diseños arquitectónicos que garanticen una adecuada composición de los elementos que lo conforman según el esquema de organización seleccionado. Esta selección no debe realizarse de forma arbitraria, sino que siempre se encuentra relacionada con los requerimientos no funcionales de la aplicación bajo desarrollo (Blas et al., 2019).

2.5.1 Patrones GRAPS

Patrones Grasp (*General Responsibility Assignment Software Patterns*) por sus siglas en inglés, son patrones generales de *software* para asignar responsabilidades. Son guías, buenas ideas y buenos motivos para definir a que clase asignarle ciertas responsabilidades (Fernández, 2021).

- Experto: Este patrón nos dice que la responsabilidad de hacer una acción debe ser asignada a la clase que tiene la información necesaria para realizar dicha acción.
- Alta cohesión: Asigna una responsabilidad de manera que la cohesión permanezca alta, es decir, asignar a las clases responsabilidades que trabajen sobre una misma área de la aplicación y que no tengan mucha complejidad.
- Bajo acoplamiento: Asigna una responsabilidad de manera que el acoplamiento permanezca bajo, es decir, se basa asignar responsabilidades de forma tal que cada clase se comunique con el menor número de clases.

2.5.2 Patrones GOF

Los patrones GOF se definen como combinaciones de componentes, casi siempre clases y objetos que por experiencia se sabe que resuelven ciertos problemas de diseño comunes. En términos generales, es posible decir que un patrón de diseño es una solución a un problema recurrente en el diseño de *software* (Cortez et al., 2020)

Observer (Observador): define una relación de uno a muchos entre objetos, de manera que cuando un objeto cambie de estado se notifique y actualicen automáticamente todos los objetos que dependen de él.

2.6 Diagrama de Clase del Diseño

Un diagrama de clase de diseño muestra la especificación para las clases *software* de una aplicación, incluye la siguiente información: clases, asociaciones, atributos, interfaces con sus operaciones y constantes, métodos, navegación y dependencia. Es una representación más concreta que el diagrama de clases del análisis, representa la parte estática del sistema y representa las clases y sus relaciones.

A continuación, se muestra el diagrama de clases del diseño:

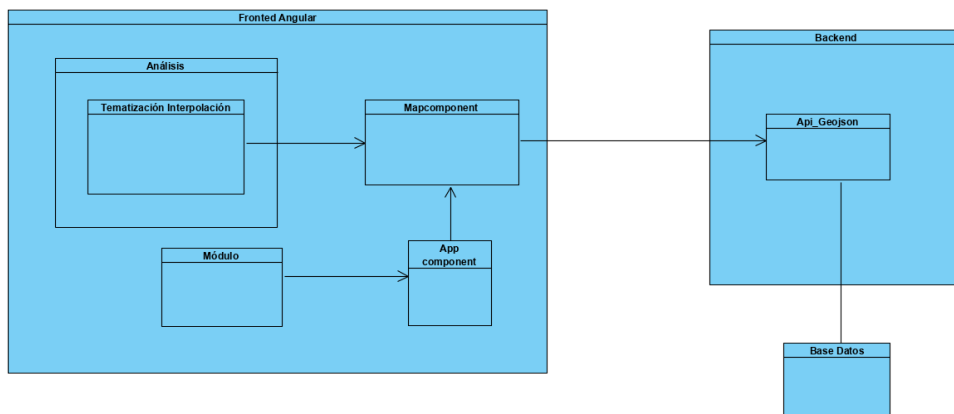


Figura 11: Diagrama de clases del diseño.

2.7. Modelo de datos

Los modelos de datos son mecanismos que permiten la abstracción y representación de un dominio, mediante un conjunto de reglas y símbolos pertenecientes a un lenguaje de modelado que es conforme al modelo, estos consideran un conjunto de elementos claves los cuales permitirán describir algo físico, abstracto o una realidad hipotética.

Para el desarrollo del componente de tematización se definió el siguiente modelo de datos apoyado en el diagrama Entidad-Relación. Este representa un ejemplo, considerando que antes deba existir una base de datos, que brinde la información necesaria a este modelo. Como la plataforma ULTRON es una plataforma personalizada en este caso se utilizaron los datos del censo de la población como indicador para la tematización

A continuación, se muestra el modelo de datos:

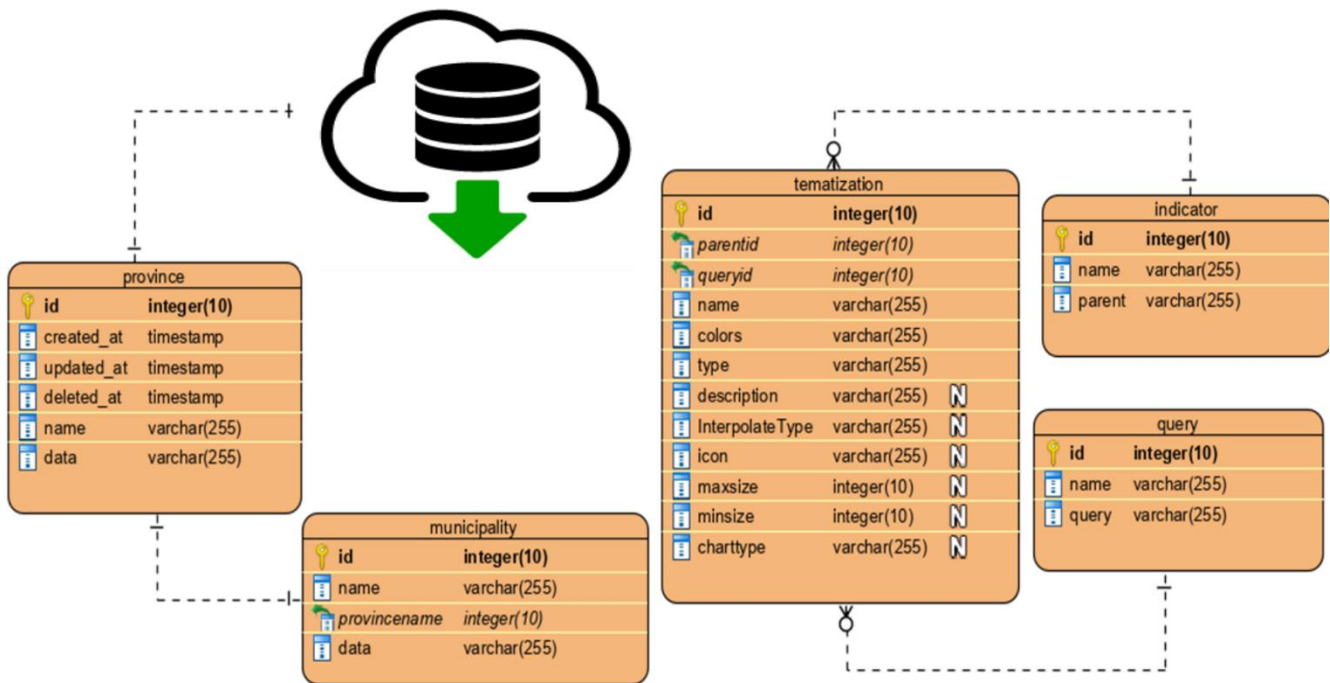


Figura 12: Modelo de datos.

2.8 Modelo de despliegue

Los Diagramas de Despliegue muestran las relaciones físicas de los distintos nodos que componen un sistema y el reparto de los componentes sobre dichos nodos. La vista de despliegue representa la disposición de las instancias de componentes de ejecución en instancias de nodos conectados por enlaces de comunicación. Un nodo es un recurso de ejecución tal como un computador, un dispositivo o memoria (Pressman, 2005).

Los diagramas de despliegue son los complementos de los diagramas de componentes que, unidos, proveen la vista de implementación del sistema. Describen la topología del sistema, la estructura de los elementos de hardware y el *software* que ejecuta cada uno de ellos. Los diagramas de despliegue representan a los nodos y sus relaciones (Vidal et al., 2012).

Los nodos son conectados por asociaciones de comunicación tales como enlaces de red y conexiones TCP/IP. Para cada componente de un diagrama es necesario que se deba documentar las características técnicas requeridas, el tráfico de la red y el tiempo de respuesta.

A continuación, se muestra el diagrama de despliegue:

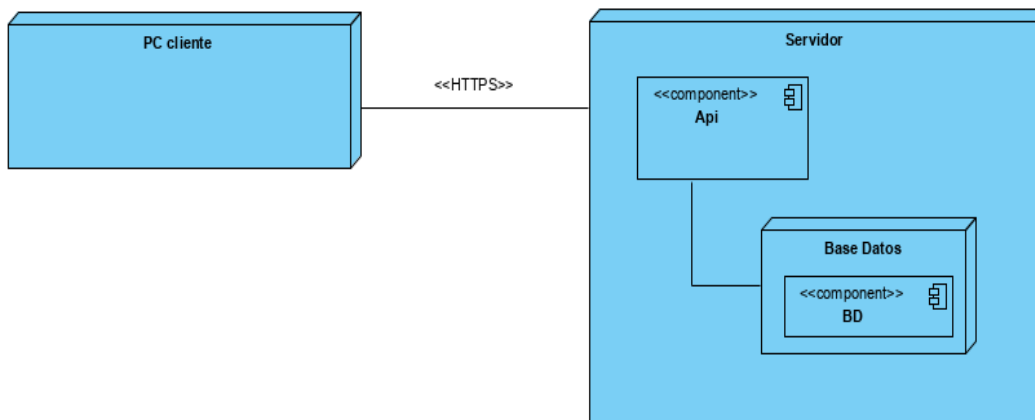


Figura 13: Diagrama de despliegue.

Conclusiones del capítulo.

Con la obtención de los requisitos funcionales y no funcionales se garantiza una propuesta de solución que satisface las necesidades del cliente, además se identificaron las funcionalidades del sistema y los atributos que debe mostrar el mismo. Los artefactos generados según la metodología AUP-UCI, la arquitectura y los patrones de diseño crearon las bases necesarias para la construcción de la propuesta de solución. También ayudó al programador a entender las funcionalidades que desea el cliente, implementando un sistema conforme a las necesidades del mismo.

CAPÍTULO 3: VALIDACIÓN DE LA SOLUCIÓN PROPUESTA.

En el presente capítulo se aborda lo referente a la implementación y prueba del sistema. Una de las disciplinas críticas en el desarrollo de *software* es la implementación, aquí es donde se materializa el análisis y el diseño. Se muestra la organización del mismo mediante el modelo de implementación utilizado, el diagrama de componentes, así como las pruebas efectuadas para verificar el buen funcionamiento del sistema.

3.1 Modelo Implementación

El modelo de implementación es comprendido por un conjunto de componentes y subsistemas que constituyen la composición física de la implementación del sistema. Entre los componentes podemos encontrar datos, archivos, ejecutables, código fuente y los directorios. Fundamentalmente, se describe la relación que existe desde los paquetes y clases del modelo de diseño a subsistemas y componentes físicos (Ramos et al., 2017)

3.2 Diagrama de Componentes

Un diagrama de componentes proporciona una visión general del sistema y documenta la organización de los componentes del sistema, sus relaciones y dependencias mutuas. Los diagramas de componentes proporcionan una visión orientada a la ejecución, es decir, dan al desarrollador información sobre el sistema, si funciona de forma coherente y cumple sus tareas y objetivos (Varón, 2018).

Los diagramas de componentes UML representan las relaciones entre los componentes individuales del sistema mediante una vista de diseño estática. Pueden ilustrar aspectos de modelado lógico y físico.

Como parte del correcto desarrollo del modelo de implementación se obtiene el diagrama de componentes que a continuación se presenta.

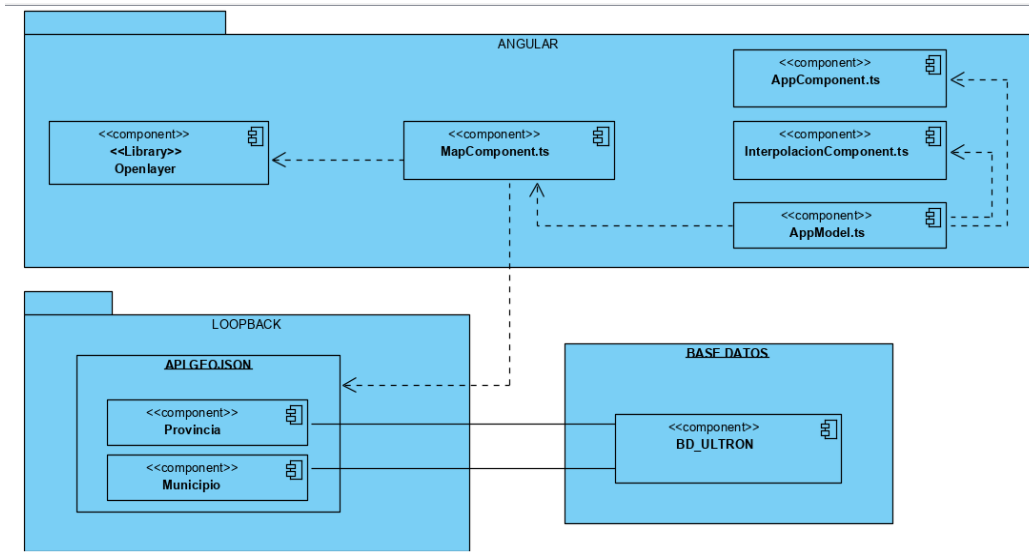


Figura 14: Diagrama de componentes.

3.3 Estándares de Codificación

Un estándar de codificación es una serie de reglas que determinan cómo debe escribirse el código. El objetivo es lograr un código fácil de leer por otros humanos (para la computadora mientras funcione todo lo demás da igual). El uso de un estándar de codificación hace más fácil la lectura del código escrito por diferentes personas, lo que hace más sencillo el mantenimiento del código a largo plazo. En definitiva, seguir un estándar de codificación permite disminuir la carga cognitiva que soportan los desarrolladores (Segura, 2022).

A continuación, se muestran los estándares de codificación que fueron utilizados en el desarrollo:

1. Márgenes, indentación, líneas y espacios

1.1 La Indentación se realizará siempre con tabuladores y no con cuatro puntos cuidando de no mezclar un estilo con el otro.

1.2 Deberá cuidarse de no dejar espacios en blanco al final de cada línea ni en las líneas en blanco.

1.3 Se dejará una línea en blanco inmediatamente después del inicio de un bloque de código, dígame bloques de funciones, bloques condicionales, bucles, etc.

1.4 Se dejará una línea en blanco inmediatamente después del bloque de declaración de variables.

1.5 Se dejará un espacio en blanco antes y después de cada operador, dígame operadores aritméticos, operadores booleanos, paréntesis, llaves, etc.

1.6 La llave de apertura de un bloque de código se coloca después de la instrucción anterior y no en una línea en blanco independiente.

1.7 La llave de clausura de un bloque de código tendrá el mismo nivel de indentación que la línea de la llave de apertura.

1.8 Cada bloque de código anidado tendrá un nivel de indentación más que el bloque padre exceptuando las llaves de apertura y cierre.

2- Variables, identificadores y parámetros

2.1 Los identificadores de variables, parámetros y nombres de métodos, en lo subsiguiente simplemente identificadores, utilizarán la convención camel case iniciando con letra minúscula y colocando en mayúsculas el resto de las palabras.

2.2 Los identificadores se escribirán en idioma inglés evitando el uso de contracciones y empleando términos comunes cercanos al inglés técnico.

2.3 Todas las variables de un bloque de código se declararán al inicio del método en cuestión.

2.4 Las variables se declaran una por línea y empleando la palabra reservada que nunca podrá omitirse, aunque el lenguaje lo permita.

2.5 Todas las variables deben ser inicializadas en el momento de declaración para indicar la naturaleza de su contenido. En caso de no estar disponible datos para su inicialización se emplea el 0 (cero) para enteros, 0.0 para números de coma flotante, "" (doble comilla) para cadenas, [] (corchetes) para arreglos y {} (llaves) para objetos.

3.4 Pruebas

Las pruebas de *software* (en inglés *software testing*) son las investigaciones empíricas y técnicas cuyo objetivo es proporcionar información objetiva e independiente sobre la calidad del producto a la parte interesada. Al planificar y ejecutar las pruebas los probadores de

software deben considerar: el *software* y su función de cálculo, las entradas y cómo se pueden combinar, y el entorno en el que el *software* eventualmente funcionará. Este difícil proceso requiere tiempo, sofisticación técnica y una adecuada planificación. Los probadores no solo deben tener buenas habilidades de desarrollo (a menudo las pruebas requieren una amplia cantidad de código), sino también conocimientos en lenguajes formales, teoría de grafos, lógica computacional y algoritmia.

De hecho, los probadores creativos aplican muchas disciplinas relacionadas con la informática al problema de las pruebas, a menudo con resultados impresionantes. Existen dos enfoques de pruebas: las pruebas de caja negra y las pruebas de caja blanca. Las pruebas de caja negra, también denominadas pruebas de comportamiento, se centran en los requisitos funcionales del *software*. Las pruebas de caja blanca, denominadas también pruebas de caja de cristal, utilizan la estructura de control del diseño procedimental para obtener los casos de prueba. (Serna, 2021).

3.4.1 Pruebas de Caja Negra

Las pruebas de caja negra, conocidas también como *black box testing*, pueden definirse como una técnica donde se busca la verificación de las funcionalidades del *software* o aplicación analizada, sin tomar como referente la estructura del código interno, las rutas de tipo internas ni la información referente a la implementación. Esto quiere decir que la prueba se lleva a cabo con desconocimiento del funcionamiento del sistema interno, debido a que se enfoca en la entrada y salida de un *software*, tomando como base sus especificaciones y requisitos. De manera que se puede asegurar que el objetivo de las pruebas de caja negra está relacionado con la validación de los recursos funcionales del *software* o aplicación que se busca examinar.

Mientras más amplio sea el espectro de elementos de entrada para realizar la prueba, las probabilidades de encontrar problemas en el *software* aumentan y, por lo tanto, será más confiable la calidad del *software*.

Este tipo de pruebas permite encontrar:

- Funciones incorrectas o ausentes
- Errores de interfaz.

- Errores en estructuras de datos o en accesos a las bases de datos externas.
- Errores de rendimiento.
- Errores de inicialización y terminación.

Para preparar los casos de prueba es necesario un número de datos que ayuden a la ejecución de estos casos y que permitan que el sistema se ejecute en todas sus variantes. Estos datos pueden ser válidos o no para el *software* y son seleccionados atendiendo a las especificidades de la funcionalidad a probar. Para realiza estas pruebas existen varias técnicas:

Partición de equivalencia: divide el campo de entrada en clases de datos que tienden a ejercitar determinadas funciones del *software*. Esta técnica es muy efectiva puesto que permite examinar los valores válidos e inválidos de las entradas existentes en el producto, descubre de forma inmediata una clase de errores que, de otro modo, requerirían la realización de muchos casos antes de detectar el error genérico. La partición de equivalente se dirige a la definición de casos de pruebas que descubran clases de errores, reduciendo de este modo el número de pruebas a realizar.

Se realizaron 3 casos de prueba a continuación se muestra un ejemplo de los mismos:

Tabla 11: Clases de equivalencia. (Elaboración Propia)

Escenario	Descripción	Nombre	Descripción de la tematización	Consulta	Indicador	Método a Interpolar	Respuesta del sistema	Resultado de la prueba
EC1 Adicionar una tematización por Interpolación con datos correctos	Permitirá al usuario crear mapa temático por interpolación	Interpolation	nueva Interpolation	consulta1	Indicador1	IDW	Operación realizada satisfactoriamente	Satisfactoria
		V	V	V	V	V		
EC2 Adicionar una tematización por Interpolación con	No permitirá al usuario crear mapa temático por interpolación						Este campo es obligatorio	No Satisfactoria
		I	V	I	I	I		

datos incorrectos		Interpolation#1	Nueva descripción	Consulta1	Indicador1	Lineal	Solo admite letras, espacios entre palabras y el signo tipográfico (-)	
		I	V	V	V	V		
		In	Nueva descripción	Consulta1	Indicador1	Lineal	El tamaño mínimo requerido es 3 caracteres	
		I	V	V	V	V		

Tabla 11: Descripción de Variables.

NO	Nombre del Campo	Clasificación	Valor Nulo	Descripción
1	Nombre	Campo de texto	No	Se introduce el nombre de la tematización que se desea crear
2	Descripción	Campo de texto	Si	Campo que permite introducir una breve descripción de la tematización que se crea.

3	Consulta	Campo de Selección	No	Seleccionar la consulta deseada.
4	Indicador	Campo de Selección	No	Seleccionar el indicador deseado
5	Método Interpolación	Campo de Selección	No	Seleccionar el método de interpolación deseado

3.4.2 Pruebas Unitarias

Las pruebas unitarias, están específicamente encaminadas al enfoque de caja blanca ya que centran el proceso de verificación en la menor unidad del diseño: el módulo, la clase, el método, entre otros. Estas utilizan la descripción del diseño detallado como guía, para probar los caminos de control importantes, con el fin de descubrir errores en la unidad.

Existen herramientas comerciales que automatizan la ejecución de las pruebas unitarias de *software*; sin embargo, el diseño de casos de prueba continúa en manos de los desarrolladores. Esto provoca que, generalmente, se obvie esta actividad o se realice un diseño empírico, que no siempre garantiza que se detecten todos los errores relacionados con la lógica interna (Rojas-Robert et al., 2019).

Jasmine, es un framework de Desarrollo Dirigido por Comportamientos (*Behavior Driven Development*) para efectuar pruebas unitarias (*Unit Testing*) de código JavaScript. Cada día es más necesario hacer pruebas a nuestro código, para asegurar su estabilidad y hacerlo escalable. Actualmente, existen muchas librerías y *frameworks* en diferentes lenguajes de programación que nos ayudan a lograrlo. Con Jasmine es posible hacer fácilmente nuestras pruebas, con una sintaxis idiomática inspirada en *ScrewUnit*, *JSSpec*, *JSPEC* y *RSpec* (Ruby).

A continuación, se muestra las pruebas efectuadas al sistema con uso de la herramienta antes mencionada:

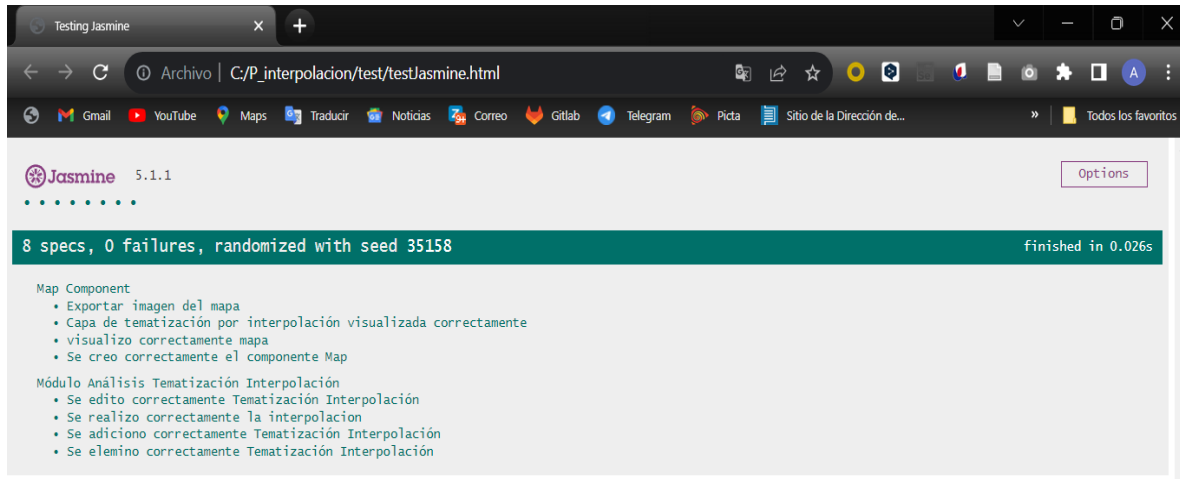


Figura 16: Prueba unitaria con jasmine.

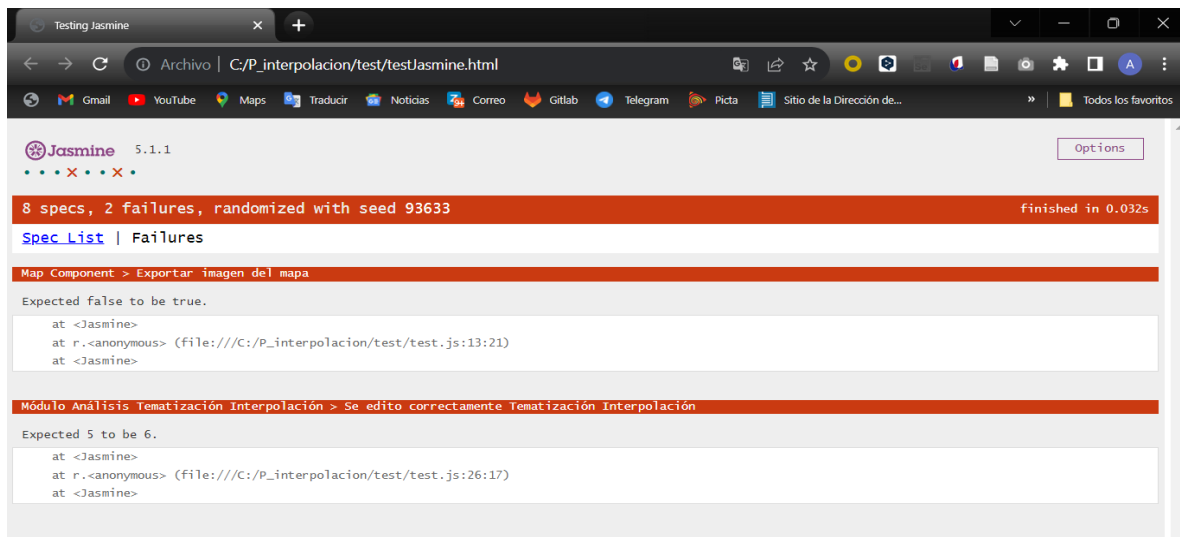
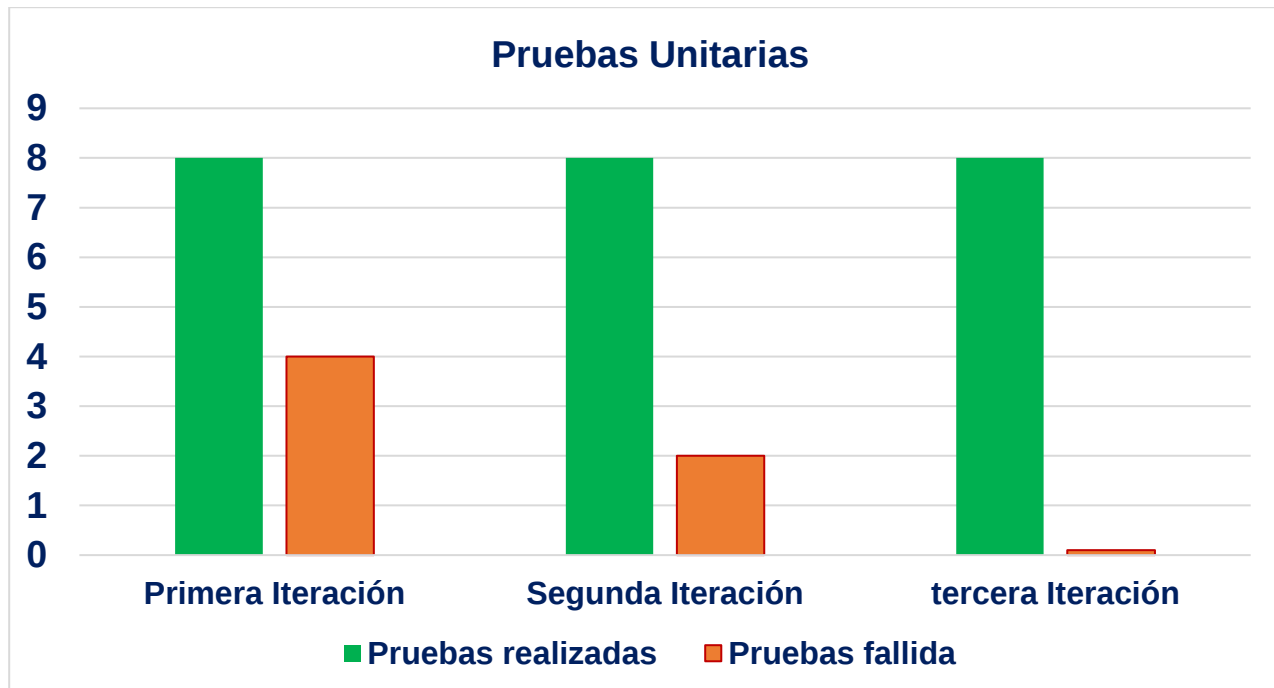


Figura 17: Prueba unitaria con jasmine.

A continuación, se muestra un gráfico con las pruebas unitarias realizadas por cada una de las iteraciones, las fallidas y los errores arreglados en cada una.



Conclusiones del capítulo

Durante el presente capítulo se diseñó el Diagrama de Componente como sustento de la implementación de la solución al problema de investigación. La definición y utilización de los estándares de codificación en la implementación del sistema propuesto permitió el desarrollo de los componentes con un alto grado de legibilidad; lo que provee una guía para el mantenimiento y actualización del sistema, con código claro y bien documentado. Las pruebas realizadas ayudaron a identificar los errores existentes en el módulo para darle solución, se logró de esta forma que el producto cumpla sus expectativas deseadas.

CONCLUSIONES FINALES

Una vez culminada la investigación es posible afirmar que se cumplió el objetivo general trazado al inicio, por lo que se concluye que:

1. Se diseñó la propuesta de solución del módulo y se definieron los requisitos con los que debe cumplir, a partir de los artefactos correspondientes a la metodología de desarrollo AUP-UCI en su escenario 4.
2. La identificación de los patrones de diseño y el estilo arquitectónico evidencia que la solución propuesta tiene un alto grado de adaptabilidad ante posibles modificaciones.
3. Las pruebas permitieron detectar y corregir los errores durante la implementación, posibilitando cumplir con las especificaciones requeridas y la validación de la aplicación implementada.

RECOMENDACIONES.

Se recomienda:

- Ampliar los servicios implementados para el análisis de datos mediante otros métodos de interpolación.

REFERENCIAS BIBLIOGRÁFICAS

- Alonso, C. (2019). *Modelo-Vista-Controlador. Lenguaje UML*.
<https://hdl.handle.net/10953.1/11437>
- Beltrán, P. A. (2020). *Comparación de técnicas de interpolación espacial (IDW y Kriging) utilizando medidas de campo eléctrico en la Comuna Dos de la ciudad de san José de Cúcuta*. 1-2.
- Blas, M. J., Horacio, L., & Gonnet, S. M. (2019). *Modelado y verificación de patrones de diseño de arquitectura de software para entornos de computación en la nube*. <https://ri.conicet.gov.ar/handle/11336/125130>
- Brito, D., Bravo, J., & Jiménez, L. (2019). *Aplicación móvil para el análisis de la información captada en SIGE v3.0*. <http://publicaciones.uci>
- Chica, J. A. (2018). *Interpolación Spline y aplicación a las curvas de nivel*.
<https://hdl.handle.net/2445/122512>
- Cortez, A. A., Naveda, C. A., Garis, A. G., & Riesgo, D. (2020). *Especificación del patrón de diseño memento a través de un perfil UML*.
<https://www.researchgate.net/publication/342409507>
- Del Águila, I. M. (2019). *Ingeniería de requisitos: Material didáctico*. (Vol. 35). Cuaderno de teoría.
- Delgado-Téllez, R., & PEÑA-DELA CRUZ, A. (2019). *Cartografía de variables climáticas basada en gradiente, sistema de expertos y Sistema de Información Geográfica*.
<https://rcm.insmet.cu/index.php/rcm/article/view/464>
- Fallas, J. (2007). *Modelos digitales de elevación: Teoría, métodos de interpolación y aplicaciones*. <https://www.researchgate.net/profile/Jorge->

Fallas/publication/229021279_Modelos_digitales_de_elevacion_Teoria_metodos_de_interpolacion_y_aplicaciones

Farinango, E. P. (2020). *Estudio del framework angular para desarrollar aplicaciones single_page. Desarrollo del sistema de información, seguimiento y control para la hacienda ganadera La Vega.* <http://repositorio.utn.edu.ec/handle/123456789/10364>

Fernández, C. (2021). *Portal web de la Dirección Provincial de Justicia de Artemisa.* <https://repositorio.uci.cu>

Fuenzalida, M., Buzai, G. D., Moreno, A., & García de León, A. (2018). *Geografía, Geotecnología y Análisis Espacial: Tendencias, métodos y aplicaciones.* Universidad Alberto Hurtado.

García, J. A. (2022). *De la topología a la geometría: Implementación de mapas mentales a los Sistemas de Información Geográfica.* 61(2), 88-107. <https://doi.org/10.30827/cuadgeo.v61i2.22859>

Guagalango, J. E. (2023). *Sistema web para la gestión de proyectos de la empresa H&H Publicidad y diseño 3D utilizando herramientas de software libre.* [Universidad Estatal Península de Santa Elena]. <https://repositorio.upse.edu.ec/handle/46000/9276>

Guevara, J. del C., & Torres, D. (2022). *Metodología para especificar requisitos de gestión documental desde la ingeniería de requisitos.* 36(91). <https://doi.org/10.22201/iibi.24488321.xe.2022.91.58555>

Hernández, J. L., Tun-Dzul, F., Mauricio, J. A., & Hernández Martínez, L. Á. (2023). *Métodos de interpolación espacial para el mapeo de la riqueza de especies usando R.* <https://www.researchgate.net/publication/373139552>

- Herrera, S. (2007). *Homogeneización e Interpolación de Observaciones de Alta resolución*. https://www.researchgate.net/profile/Sixto-Herrera-Garcia/publication/266497362_Homogeneizacion_e_Interpolacion_de_Observaciones_de_Alta_Resolucion
- Ledesma, L. D. (2018). *Modelo conceptual para el aseguramiento de la calidad de los requerimientos en proyectos de software*. <http://hdl.handle.net/10785/5017>
- LONDOÑO, A. (2021). *Diseño de mapas de información de humedad relativa en el departamento de Caldas a través de métodos de interpolación y herramientas SIG*. <https://docplayer.es/217065584-Modalidad-de-grado-tabajo-de-grado.html>
- López, E., & Abascal, R. (2022). *Procesamiento y visualización de la información geográfica: Una guía para la extracción de conocimiento*. 43(141). <http://pistaseducativas.celaya.tecnm.mx/index.php/pistas/article/view/2695>
- Manrique López, D. A. (2017). *Desarrollo de un complemento de QGIS, para la estimación de curvas, intensidad, duración y frecuencia a partir de datos del IDEAM*. https://ciaf.igac.gov.co/sites/files/files_ciaf/Manrique-Lopez-Dany-Alexander.pdf
- Mendoza, D., & Baquero, L. (2016). *Extensión de la herramienta Visual Paradigm for UML para la evaluación y corrección de Diagramas de Casos de Uso*. https://www.researchgate.net/profile/Lionel-Baquero-Hernandez/publication/305160646_Extension_de_la_herramienta_Visual_Paradigm_for_UML_para_la_evaluacion_y_correccion_de_Diagramas_de_Casos_de_Uso

- Murillo, D., Ortega, I., Carrillo, J. D., Pardo, A., & Rendón, J. (2012). *Comparación de métodos de interpolación para la generación de mapas de ruido en entornos urbanos*. 3(1), 62-68. <https://doi.org/DOI:10.21500/20275846.265>
- Pilicita, A., Borja, Y., & Gutiérrez, G. (2021). *Rendimiento de MariaDB y PostgreSQL*. Editorial La Libertad. 7(2). <https://repositorio.upse.edu.ec/handle/46000/7315>
- Pressman, R. (2005). *Software Engineering: A Practitioners Approach* (6ta ed.).
- Puciarrelli, L. (2020). *Angular: TypeScript–Arquitectura–Instalación–Directivas y Bindings–Forms–Ruteo y más*.
- Ramos, D., Noriega, R., Laínez, J. R., & Durango, A. (2017). *Curso de Ingeniería de Software 2* (2da ed.). Campus academy.
- Rojas-Robert, D. M., Pérez-Morales, Z., & Delgado-Dapena, M. D. (2019). *Generador de valores interesantes para casos de pruebas unitarias*. 40(2), 183-193.
- Roncero, A. (2019). *La interpolación aplicada al procesamiento de imágenes digitales*. <http://hdl.handle.net/10234/185159>
- Segura, E. A. (2022). *Componentes de calidad software y su utilización en aplicaciones web*. 6(3), 3193-3204.
- Serna, E. (2021). *Métodos Formales, Ingeniería de Requisitos y Pruebas del Software*/. <https://doi.org/10.5281/zenodo.4534359>
- Silva, J. A., Rivero, M. A., Scialfa, E. A., & Linares, S. (2020). *Análisis espacial de la seropositividad de leptospirosis humana en un espacio urbano del partido de Tandil mediante Sistemas de Información Geográfica*. <https://revistageosig.wixsite.com/geosig/geosig-17-2020>

- Suarez, N., & Custodia, J. (2014). *Evolución de las tecnologías de información y comunicacione en el proceso de enseñanza-aprendizaje*. 11(1).
<https://doi.org/10.14483/23229X.8>
- Varón, Á. (2018). *Ingeniería de Software I*.
<https://digitk.areandina.edu.co/repositorio/handle/123456789/1235>
- Vega-Blancas, V. de J., Fernández Reynoso, D. S., Macedo-Cruz, A., Ríos-Berber, J. D., & Ruiz-Bello, A. (2022). *Análisis de la fertilidad del suelo mediante la validación e interpolación Kriging de sus variables*.
- Vera, J. M. (2012). *Incertidumbre y confiabilidad de métodos de interpolación espacial en el complemento de registros hidrológicos*. <https://ri-ng.uaq.mx/handle/123456789/493>
- Vidal, C. L., Schmal, R. F., Rivero, S., & Villarroel, R. H. (2012). *Extensión del Diagrama de Secuencias UML (Lenguaje de Modelado Unificado) para el Modelado Orientado a Aspectos*. *Información tecnológica*. 23(6).
<https://doi.org/10.4067/S0718-07642012000600007>
- Yuquin, H., Zhenbo, B., Duan, K., Wu, J., & Wang, H. (2019). *Application of OpenLayers in marine information monitoring*. 118.
<https://doi.org/10.1051/e3sconf/201911803006>