



Facultad de las Ciencias y Tecnologías Computacionales

Sistema de Información Geográfica para decidimOS

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

Autor(es): Frank Alberto Plasencia Téllez

Tutor(es): Midalis Rodríguez Rosado

Yaíma Del Campo Peña

La Habana, noviembre de 2022

Año 63 de la Revolución

DECLARACIÓN DE AUTORÍA

El autor del trabajo de diploma con título “**Sistema de Información Geográfica para decidimOS**” concede a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la investigación, con carácter exclusivo. De forma similar se declara como único autor de su contenido. Para que así conste firma(n) la presente a los 23 días del mes de noviembre del año 2022.

Frank Alberto Plasencia Téllez

Firma del Autor

Midalis Rodríguez Rosado

Firma del Tutor

Yaíma Del Campo Peña

Firma del Tutor

DATOS DE CONTACTO

Tutor:

Nombre y Apellidos: Midalis Rodríguez Rosado.

Títulos académicos: Ingeniería en Ciencias Informáticas.

Lugar de trabajo: Centro CREAD, proyecto Elecciones.

Responsabilidades laborales: Administradora de base de datos.

Experiencia profesional: 5 años.

Correo electrónico: mrosado@uci.cu

Tutor:

Nombre y Apellidos: Yaíma Del Campo Peña.

Títulos académicos: Ingeniería en Ciencias Informáticas.

Lugar de trabajo: Centro CREAD, proyecto de SIG.

Responsabilidades laborales: Líder de proyecto.

Experiencia profesional: 7 años.

Correo electrónico: ydelcampo@uci.cu

Autor:

Nombre y Apellidos: Frank Alberto Plasencia Téllez.

Correo electrónico: frankapt@estudiantes.uci.cu

AGRADECIMIENTOS

A mis **Tutoras** por el esmero y la ocupación, por las horas dedicadas a una idea que hoy germina. **¡Muchas Gracias!**

A la **Universidad de Ciencias Informáticas**, a la **Facultad de las Ciencias y Tecnologías Computacionales** y al **Claustro de Profesores** por ayudarnos en nuestra formación paso a paso.

DEDICATORIA

A mis padres: Por predicar la razón y por embriagarme de amor.

A mi hermana del alma, Cosí: Por su sensatez y paciencia, por su ingenio... por compartir siempre mis yugos... por cuidar cada día de mí.

A toda mi familia, en especial mi a mi **primo Jorgito:** por estar a mi lado en todo momento de la vida.

RESUMEN

El Centro de Representación y Análisis de Datos se encuentra desarrollando el ecosistema de software decidimOS, que permite la gestión de las personas involucradas en los procesos electorales. Actualmente la información relacionada con este proceso se almacena en una base de datos y se generan reportes en formato .pdf. Representar esta información geográficamente es una de las tareas que tiene hoy dicho centro de manera que facilite la toma de decisiones. La presente investigación tiene como objetivo desarrollar un Sistema de Información Geográfica para el proceso electoral cubano. Para ello se realizó un estudio sobre los sistemas homólogos evidenciando las ventajas del análisis geoespacial en los procesos electorales para realizar análisis estadísticos y toma de decisiones. Para la implementación de la solución se utilizó Visual Paradigm for UML v5.0 para el modelado de datos, PostgreSQL v14 como gestor de base de datos y PostGIS v3.0.3 como extensión para datos espaciales, OpenLayers v4.1.1 para la visualización de mapas del lado del cliente, para la codificación Visual Studio Code v1.51.1, Express.js v4.16.3 y Node.js v16.18.0. Como servidor de mapas MapServer v6.4 y Apache JMeter v2.3.4 para realizar pruebas automatizadas de rendimiento. Para guiar el proceso de desarrollo se utilizó la metodología AUP-UCI, adaptada a los procesos productivos de la universidad. Para diseñar la solución propuesta se realizó el Modelo del Diseño, Modelo de Datos, Diagrama de Despliegue y Diagrama de Componentes. Se definió, además los estilos y patrones arquitectónicos, así como el estándar de código utilizado para las clases atributos y funcionalidades.

PALABRAS CLAVE

Sistema Electoral Cubano, Sistemas de Información Geográfica, bases de datos geoespaciales, mapas temáticos, toma de decisiones.

ABSTRACT

The Centro de Representación y Análisis de Datos is developing the decidimOS software ecosystem, which allows the management of the people involved in the electoral processes. Currently the information related to this process is stored in a database and reports are generated in .pdf format. Representing this information geographically is one of the tasks that this center has today in order to facilitate decision making. The objective of this research is to develop a Geographic Information System for the Cuban electoral process. For this purpose, a study was carried out on counterpart systems, showing the advantages of geospatial analysis in electoral processes for statistical analysis and decision making. For the implementation of the solution, Visual Paradigm for UML v5.0 was used for data modeling, PostgreSQL v14 as database manager and PostGIS v3.0.3 as extension for spatial data, OpenLayers v4.1.1 for client-side map visualization, for coding Visual Studio Code v1.51.1, Express.js v4.16.3 and Node.js v16.18.0. As map server MapServer v6.4 and Apache JMeter v2.3.4 for automated performance testing. The AUP-UCI methodology was used to guide the development process, adapted to the university's production processes. To design the proposed solution, the Design Model, Data Model, Deployment Diagram and Component Diagram were created. The architectural styles and patterns were also defined, as well as the code standard used for the classes attributes and functionalities.

KEY WORDS

Cuban Electoral System, Geographic Information Systems, geospatial databases, thematic maps, decision making.

TABLA DE CONTENIDOS

| | |
|---|----|
| INTRODUCCIÓN..... | 1 |
| CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA DEL SISTEMA DE INFORMACIÓN GEOGRÁFICA PARA DECIDIMOS | 6 |
| 1.1 Conceptos asociados a la investigación | 6 |
| 1.1.1 Sistema de Información Geográfica:..... | 6 |
| 1.1.2 Mapa temático: | 7 |
| 1.1.3 Bases de datos geoespaciales: | 8 |
| 1.2 Sistemas homólogos | 9 |
| 1.2.1 Sistemas Homólogos a nivel internacional | 9 |
| 1.2.2 Sistemas Homólogos a nivel nacional | 12 |
| 1.3 Metodología del desarrollo de software | 15 |
| 1.4 Herramientas y tecnologías a utilizar para el desarrollo del SIG | 15 |
| Conclusiones del capítulo..... | 19 |
| CAPÍTULO 2: DISEÑO DE LA SOLUCIÓN PROPUESTA AL PROBLEMA CIENTÍFICO..... | 20 |
| 2.1 Descripción del sistema propuesto..... | 20 |
| 2.2 Requisitos del sistema..... | 20 |
| 2.2.1 Requisitos Funcionales..... | 20 |
| 2.2.2 Requisitos no Funcionales..... | 24 |
| 2.3 Historias de usuarios | 28 |
| 2.4 Patrón de diseño arquitectónico | 32 |
| 2.5 Modelo de diseño | 33 |
| 2.6 Patrones para la asignación de responsabilidades (GRASP)..... | 35 |
| 2.7 Patrones GOF | 36 |
| 2.8 Modelo de datos | 37 |
| 2.9 Diagrama de Despliegue | 38 |
| Conclusiones del capítulo..... | 39 |

| | |
|---|----|
| CAPÍTULO III: IMPLEMENTACIÓN Y VALIDACIÓN DEL SISTEMA DE INFORMACIÓN GEOGRÁFICA PARA DECIDIMOS..... | 40 |
| 3.1 Diagrama de componentes..... | 40 |
| 3.2 Estándares de codificación..... | 41 |
| 3.3 Estrategia de pruebas..... | 42 |
| 3.3.1 Niveles de pruebas..... | 42 |
| 3.3.2 Tipos de pruebas..... | 43 |
| 3.3.3 Métodos de prueba..... | 43 |
| 3.3.4 Técnicas de prueba..... | 44 |
| 3.4 Aplicación de las pruebas..... | 44 |
| Conclusiones del capítulo..... | 50 |
| CONCLUSIONES FINALES..... | 51 |
| RECOMENDACIONES..... | 52 |
| REFERENCIAS BIBLIOGRÁFICAS..... | 53 |
| ANEXOS..... | 58 |
| Anexo 1: Historia de Usuario..... | 58 |
| Anexo 2: Casos de prueba..... | 60 |

ÍNDICE DE TABLAS

| | |
|---|----|
| TABLA 1: CIUDAD AUTÓNOMA DE BUENOS AIRES. RESULTADO DE LAS ELECCIONES GENERALES..... | 11 |
| TABLA 2: HISTORIAL DE USUARIO “MOSTRAR TEMATIZACIÓN DE COLORES DE LA CANTIDAD DE CIRCUNSCRIPCIONES ORDINARIAS POR PROVINCIAS”..... | 28 |
| TABLA 3: HISTORIAL DE USUARIO “MOSTRAR TEMATIZACIÓN DE COLORES DEL PORCIENTO DE ELECTORES QUE VOTARON SEGÚN PARTE BÁSICO POR PROVINCIAS”. | 29 |
| TABLA 4: HISTORIAL DE USUARIO “MOSTRAR TEMATIZACIÓN DE COLORES DEL PORCIENTO DE VOTOS NO POR PROVINCIAS”..... | 31 |
| TABLA 5: CASO DE PRUEBA MOSTRAR TEMATIZACIÓN DE COLORES DE LA CANTIDAD DE CIRCUNSCRIPCIONES ORDINARIAS POR PROVINCIAS. | 44 |
| TABLA 6: CASO DE PRUEBA MOSTRAR TEMATIZACIÓN DE COLORES DEL PORCIENTO DE ELECTORES QUE VOTARON SEGÚN PARTE BÁSICO POR PROVINCIAS..... | 47 |
| TABLA7: MOSTRAR TEMATIZACIÓN DE COLORES DE LA CANTIDAD DE ELECTORES POR PROVINCIAS..... | 58 |
| TABLA 8: MOSTRAR TEMATIZACIÓN DE COLORES DEL PORCIENTO DE ELECTORES QUE HAN VOTADO SEGÚN BOLETAS DEPOSITADAS EN URNAS POR PROVINCIAS... | 59 |
| TABLA 9: CASO DE PRUEBA MOSTRAR TEMATIZACIÓN DE COLORES DEL TOTAL DE VOTOS POR PROVINCIAS. | 60 |
| TABLA 10: CASO DE PRUEBA MOSTRAR TEMATIZACIÓN DE COLORES DEL PORCIENTO DE VOTOS NO POR PROVINCIAS..... | 62 |

ÍNDICE DE FIGURAS

| | |
|--|----|
| FIGURA 1: PANTALLA PRINCIPAL EN LA QUE SE PUEDEN VER LOS MILITANTES DEL PARTIDO DESDE UNA PERSPECTIVA GENERAL..... | 10 |
| FIGURA 2: DISTRIBUCIÓN ESPACIAL DE LOS VOTOS EN LA ELECCIÓN GENERAL A JEFE DE GOBIERNO DE LA CABA..... | 12 |
| FIGURA 3: FUNCIONALIDADES | 14 |
| FIGURA 4: REPRESENTACIÓN DEL MODELO VISTA CONTROLADOR..... | 33 |
| FIGURA 5: DIAGRAMA DE CLASES DEL DISEÑO. | 34 |
| FIGURA 6: DIAGRAMA DE ENTIDAD-RELACIÓN..... | 38 |
| FIGURA 7: DIAGRAMA DE DESPLIEGUE. | 39 |
| FIGURA 8: DIAGRAMA DE COMPONENTES. | 40 |

INTRODUCCIÓN

La importancia de las elecciones periódicas, justas, confiables y competitivas, se reconoce como uno de los principales instrumentos procedimentales que tienen las democracias contemporáneas. Si bien la democracia no queda reducida solo a los procesos electorales, estos son los mecanismos para que los ciudadanos ejerzan su poder de intervenir en las decisiones políticas mediante la selección de sus gobernantes y representantes, de manera que puedan, o renovar o revocar el mandato al reafirmar o rechazar las formas y objetivos de sus acciones para abrir paso a otro tipo de propuestas y políticas de gobierno (James et al. 2019).

El Sistema Electoral Cubano está regulado por La Ley NO. 127 Ley Electoral, aprobada por la Asamblea Nacional del Poder Popular (ANPP), en sesión ordinaria celebrada el 13 de julio del 2019. En esta se establece los tipos de elecciones, que se desarrollan de manera periódica cada 5 años, además de convocar a referendo determinada medida política de relevancia para la sociedad, siendo estas:

- 1- Elecciones municipales, en las que se eligen a los delegados a las Asambleas Municipales del Poder Popular (AMPP), se construyen estas y se eligen sus respectivos presidente y vicepresidente.
- 2- Elecciones nacionales, en las que se eligen a los diputados, se constituye la ANPP, se elige a su presidente, vicepresidente, secretario y demás miembros del Consejo de Estado, así como al presidente y vicepresidente de la Republica.
- 3- Elecciones de gobernadores y vicegobernadores y los actos de toma de posición de estos y del Consejo Provincial de Poder Popular se realizan de conformidad con lo previsto en esta ley.
- 4- Referendo, en este los ciudadanos mediante su voto libre y secreto expresan si ratifican, aprueban o modifican determinada disposición jurídica, («Ley No. 127, “Ley Electoral”, de 13 de julio de 2019» 2019)

El Consejo Electoral Nacional (CEN) es el órgano del Estado encargado de “organizar, dirigir y supervisa las elecciones, consultas populares, plebiscitos y referendos” («Ley No. 127, “Ley Electoral”, de 13 de julio de 2019» 2019). En coordinación con la Universidad de las Ciencias Informáticas (UCI) se encuentra desarrollando el ecosistema de software **decidimOS**, que

permite la gestión de las personas involucradas en los procesos electorales, los electos, las plazas vacantes, los procesos de referendo y plebiscitos y los procesos de elecciones periódicas (González et al. 2021). Este proyecto es llevado a cabo por el Centro de Representación y Análisis de Datos (CREAD).

En el último período se han llevado a cabo determinadas modificaciones jurídicas en cuanto a la aprobación de la nueva constitución (artanaisi 2019) y el código de las familias («Gaceta Oficial de la República de Cuba publicó el Código de las Familias» s. f.) mediante referendo. Teniendo en cuenta la inmediatez de estos procesos se decide priorizar en decidimOS la implementación de los procesos relacionado con Referendo.

Actualmente la información relacionada con este proceso se almacena en una base de datos y se generan reportes en formato .pdf a partir de la información solicitada, que se muestra a diferentes niveles (municipales, provinciales y nacionales) según el rol autenticado. En el proceso de Referendo existen dos etapas, la inicial que es en donde se preparan los recursos que aseguran un proceso satisfactorio y la segunda es el día en que transcurre el proceso electoral. En ambos momentos se analizan varios indicadores y se toman decisiones al respecto.

Poder representar esta información de manera sencilla y de fácil manejo, es unas de las tareas que tiene hoy dicho centro de manera que se pueda visualizar y analizar la información geográfica a diferentes niveles que facilite la toma de decisiones. Teniendo en cuenta la problemática antes expuesta destacan como solución más viable los Sistemas de Información Geográfica (SIG).

Según (Miguel 2020) un Sistemas de Información Geográfica no es más que herramientas informáticas, capaces de gestionar y analizar la información georreferenciada con vista a la resolución de problemas de base territorial y medioambiental.

Se han utilizado las Sistemas de Información Geográfica para dar soluciones en varios sectores de la sociedad como, por ejemplo:

- 1- Uso de SIG para la elaboración de planos de fincas agrícolas (Ríos, Arias, y Macías 2019).
- 2- Gestión de pavimentos vasado en SIG (Silva-Balaguera, Daza-Leguizamón, y Lopez-Valiente 2018).

- 3- Selección de emplazamiento óptimo de una base militar mediante SIG (Guerra Albarracín 2018).
- 4- Tecnologías para luchar contra la pandemia de la Covid-19: geolocalización, rastreo, big data, SIG, inteligencia artificial y privacidad (Cascón-Katchadourian 2020).
- 5- Análisis multicriterio para la localización, en Andalucía, de una central hidroeléctrica a través de un SIG (Infante Gómez 2021).

El CEN contaba con un SIG solo para el sistema SIGEL_Generales llamado SIGElectoral sobre la plataforma libre GeneSIG desarrollada por la propia UCI logrando representar la información del estado de las votaciones del país de forma centralizada. Esto contribuyó a que sea mucho más fácil y rápido acceder a dicha información (Ledea, Aguila, y Griff 2013).

Dado que el anterior sistema representaba solo uno de los procesos de elección y teniendo en cuenta la obsolescencia tecnológica se decide desarrollar un nuevo Sistemas de Información Geográfica para el ecosistema **decidimOS** sobre la plataforma que se está desarrollando actualmente en la UCI: ULTRON¹, de manera que pueda representar la información referente al proceso electoral en cuestión y visualizar lo que ocurre en tiempo real, apoyando así la toma de decisiones del ecosistema **decidimOS**.

Después de un análisis de la situación problemática se propone el siguiente **problema de investigación**: ¿Cómo contribuir a la toma de decisiones mediante el análisis geográfico del comportamiento de los procesos electorales en el país?

Se define como **objeto de estudio**: los Sistemas de Información Geográfica, enmarcado en el **campo de acción**: los Sistemas de Información Geográfica para **decidimOS**.

Se define como **objetivo general**: desarrollar un Sistema de Información Geográfica para **decidimOS** de manera que contribuya a la toma de decisiones, del cual se desglosan los siguientes **objetivos específicos**:

- 1- Elaborar los fundamentos teóricos, herramientas y metodologías a utilizar en el desarrollo del SIG para **decidimOS**.
- 2- Realizar el análisis y el diseño del SIG para **decidimOS**.
- 3- Realizar la implementación y prueba del SIG para **decidimOS**.

¹ Ultimate Run Of Nodejs

Con el propósito de dar cumplimiento a los objetivos específicos antes definidos, se plantean las siguientes **preguntas de investigación**:

- 1- ¿Cuáles son los fundamentos teóricos, herramientas y metodologías a utilizar para llevar a cabo el desarrollo del SIG para **decidimOS**?
- 2- ¿Cómo realizar el análisis y diseño para el SIG para **decidimOS**?
- 3- ¿Cómo implementar el SIG para **decidimOS** de manera que contribuya al proceso de toma de decisiones?
- 4- ¿Cómo probar el SIG para **decidimOS** de manera que se pueda garantizar el correcto funcionamiento del mismo?

Para dar respuesta a las preguntas anteriormente definidas, se proponen las siguientes **tareas de la investigación**:

- 1- Identificación de las principales soluciones existentes en el tratamiento del problema planteado mediante la realización de un estudio de los referentes teóricos – prácticos que preceden la realización de la investigación para una mejor comprensión del mismo.
- 2- Caracterización de las herramientas, tecnologías y metodología de software a utilizar en la modelación y desarrollo del componente.
- 3- Identificación de los requisitos para la implementación del SIG para **decidimOS**.
- 4- Definición de la arquitectura para definir los componentes del SIG para **decidimOS**.
- 5- Confección de la documentación técnica ingenieril asociada al desarrollo del SIG para **decidimOS**.
- 6- Implementación del SIG para **decidimOS**.
- 7- Validación del SIG para **decidimOS** implementado.

Posibles resultados:

- 1- Solución de software para **decidimOS** basada en SIG.
- 2- Artefactos de ingeniería para la documentación del SIG para **decidimOS**.

A continuación, se explican los métodos de la metodología de investigación científica empleados para la obtención de información y el futuro desarrollo del SIG para **decidimOS**

Métodos teóricos:

- 1- **Histórico-Lógico:** Análisis histórico de los Sistemas de Información Geográfica desarrollados para apoyar el proceso eleccionario de manera lógica y secuencial para determinar las características del sistema.

- 2- **Analítico-Sintético:** Será empleado el análisis de los diferentes Sistemas de Información Geográfica de apoyo a los procesos electorarios para determinar las características y limitaciones de cada uno de ellos.

Métodos empíricos:

- 1- **Análisis documental:** Se utiliza en la revisión de la literatura especializada para consultar la información necesaria en el proceso de investigación.

Estructura del trabajo de diploma:

El trabajo de diploma presenta la siguiente estructura: Introducción, tres capítulos, conclusiones generales, recomendaciones, referencias bibliográficas y anexos.

Capítulo 1: Fundamentación teórica del Sistema de Información Geográfica para decidimOS

En el capítulo se definen los conceptos relacionados con el dominio de la investigación. De igual forma, se realizó un estudio sobre los SIG de apoyo a los procesos electorarios con el fin de identificar posibles soluciones para el problema de investigación definido. También se definió la metodología y las herramientas a emplear en la construcción del SIG.

Capítulo 2: Análisis y diseño del Sistemas de Información Geográfica para decidimOS

En el capítulo se describió la propuesta de solución del problema de investigación, se especificaron los requisitos funcionales y no funcionales que debe cumplir el software. Se realizaron los diagramas de Caso de Uso, de paquete, de clase de diseño, de secuencia y entidad relación para guiar la implementación del SIG. El diagrama de despliegue se realizó para describir el entorno en el que será publicado el sistema.

Capítulo 3: Implementación y prueba del Sistemas de Información Geográfica

En el capítulo se realizó el modelo de implementación del Sistemas de Información Geográfica para la descripción de la estructura física del sistema y la relación entre sus elementos, se definieron los estándares de codificación para una mejor comprensión del código para los desarrolladores. Se describieron las pruebas realizadas para comprobar las funcionalidades del software en los diferentes escenarios, para verificar que los resultados sean los esperados.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA DEL SISTEMA DE INFORMACIÓN GEOGRÁFICA PARA DECIDIMOS

En este capítulo se definen los principales conceptos asociados al dominio del problema planteado. Se realiza un estudio de las soluciones existentes. Se definen las herramientas y tecnologías que se usaran en el desarrollo del sistema. También se define la metodología que guiará el proceso de desarrollo de software.

1.1 Conceptos asociados a la investigación

Para lograr una correcta asimilación de los temas a tratar en el presente informe, se exponen los conceptos referentes a los Sistemas de Información Geográfica, tales como: Sistema de Información Geográfica, bases de datos geoespaciales y mapas temáticos.

1.1.1 Sistema de Información Geográfica:

Según (Ávila del Campo 2014) un Sistema de Información Geográfica es un conjunto integrado de medios y métodos informáticos, capas de recoger, verificar, almacenar, gestionar, actualizar, manipular, recuperar, analizar, mostrar y transferir datos espacialmente referido a la Tierra. Los Sistema de Información Geográfica son herramientas que permiten crear consultas interactivas, analizar la información espacial, editar datos como conjuntos de mapas de un territorio específico o posición geográfica del mismo según sus coordenadas, y presentar resultados de todas estas operaciones. Además, proporciona la vinculación de diversas bases de datos.

La principal virtud de los SIG es la gran capacidad de procesamiento y almacenamiento de información, tener la información en un solo almacén de datos de tal manera que los SIG la representen tanto en reportes como en mapas geoespaciales en tiempo real, así cumplir con los principales criterios de efectividad y eficiencia a la hora de valorar los resultados de un conjunto de información base para una toma de decisiones (Herrera et al. 2018).

Es un sistema compuesto por hardware, software, procedimientos y equipo humano para capturar, manejar, manipular, transformar, analizar y modelizar datos geográficos, permitiendo representar los objetos del mundo real en términos de posición, atributos y de las interrelaciones espaciales, con el objeto de analizar estos datos y de resolver problemas de gestión y planificación pudiendo realizar preguntas a la base de datos(Martínez 2018).

Según (Martínez 2018) estos son los elementos que componen un Sistema de Información Geográfica:

1- Hardware:

- CPU, unidades de memoria
- Unidades de entrada de datos: digitalizador, escáner, imágenes por satélite, etc.
- Unidades de salida: pantalla, plotter, impresoras, etc.

2- Software. Principales módulos del software:

- Entrada y verificación de datos.
- Almacenamiento y gestión de las bases de datos.
- Salida, presentación y visualización de Datos.
- Transformación de datos.
- Interacción con el usuario = preguntas. Generación de informes. Creación de nuevos datos y nueva cartografía, transformaciones.

3- Bases de datos.

4- Equipo humano.

Según lo anterior un SIG no es más que un sistema capaz de gestionar la información espacial georreferenciada del mundo real. De esta manera se manipula esta información de manera efectiva y eficiente.

1.1.2 Mapa temático:

Mapa:

Un mapa es una representación gráfica simplificada de un territorio con propiedades métricas sobre una superficie bidimensional que puede ser plana, esférica o incluso poliédrica. Las propiedades métricas del mapa dependen de la proyección utilizada, y posibilitan la toma de medidas de distancia, ángulos o superficies sobre él y su relación con la realidad (Chiang et al. 2020).

Mapa temático:

Los mapas temáticos son mapas basados en mapas topográficos que representan cualquier fenómeno geográfico de la superficie terrestre. Persiguen objetivos bien definidos. Hacen referencia a la representación de ciertas características de distribución, relación, densidad o regionalización de objetos reales (vegetación, suelos, geología), o de conceptos abstractos

(indicadores de violencia, de desarrollo económico, de calidad de vida). Para representar variables numéricas utilizan todo tipo de recursos visuales, como superficies de diferente color, flechas que indican el movimiento de un fenómeno (a veces tienen un grosor proporcional a su magnitud), o el trazado de líneas que unen puntos de igual valor (isolineas) (Slocum et al. 2022).

1.1.3 Bases de datos geoespaciales:

Base de datos:

Una base de datos puede ser definida de la forma más sencilla como “el conjunto de datos relacionados que se encuentran agrupados o estructurados y almacenados con acceso directo a partir de un sistema de gestión de bases de datos, tipo de software muy específico para servir de interfaz entre la base de datos y el usuario” (Teja, Stuart, y Álvarez 2017).

Según (Pisco Gomez et al. 2020) una base de datos es un conjunto ordenado y estructurado de datos que representan una realidad objetiva y que están organizados independientemente de las aplicaciones, significa que pueden ser utilizadas y compartidas por usuarios y aplicaciones diferentes. O sea, que una base de datos puede considerarse una colección de datos variable en el tiempo.

Base de datos geoespacial:

Las bases de datos geoespaciales se definen como una colección de datos organizados de tal manera que sirvan efectivamente para una o varias aplicaciones de Sistemas de Información Geográfica. Estas bases de datos comprenden la asociación entre sus dos principales componentes: datos espaciales y atributos o datos no espaciales (San José et al. 2018).

Una base de datos geoespacial permite describir los objetos espaciales que la forman a través de tres características básicas: atributos, localización y topología. Los atributos representan características de los objetos que permiten saber qué son; la localización, representada por la geometría del objeto y su ubicación espacial de acuerdo a un sistema de referencia, permite saber dónde está y qué espacio ocupa el objeto, por último, la topología definida por medio de las relaciones conceptuales y espaciales entre los objetos, permite mejorar la interpretación semántica del contexto y establecer ciertas jerarquías de elementos a través de sus relaciones (Liza Contreras 2021).

Según lo anterior una base de datos geoespacial es en primer lugar, una base de datos. Dicho de otro modo, una base de datos espacial es capaz de modelar, almacenar y consultar tanto

datos estándar no espaciales (o alfanuméricos) como datos espaciales. En la práctica, los primeros siempre están conectados con los segundos, por lo que una base de datos que manejara solamente información espacial específica sería insuficiente para hacer un modelaje correcto.

1.2 Sistemas homólogos

Debido a la importancia que se le concede a SIG para la representación de datos de forma geoespacial en el mundo, en la actualidad existen varios sistemas o herramientas con este objetivo. Es por ello que se decide hacer un análisis de los sistemas ya existentes a nivel internacional y nacional con el objetivo de determinar las principales características del sistema y sus funcionalidades con las que debe contar el sistema.

1.2.1 Sistemas Homólogos a nivel internacional

Sistema de Información Geográfica para la población electoral y control de solicitudes (Bermúdez Martínez 2016).

Se presenta la implementación de un sistema de información geográfica, el cual permita tener una visualización de un comportamiento electoral de la organización del Senador Mauricio Lizcano del partido de la Unidad Nacional en la ciudad de Manizales. Teniendo como base de información la dirección de residencia de cada líder, coordinador y simpatizante, además del resto de sus datos. El sistema de información clasifica el tipo de personas que a hoy existen en la organización, de tal manera que en la cartografía de Manizales se tenga plena visualización de cada comuna con cuantos líderes cuenta, cuantos coordinadores están asignados en que barrios, y cuantos simpatizantes posee dicha organización.

Para analizar estos datos se recurre a CartoDB como visualizador porque permite interactividad, realizar acercamientos, alejarse, arrastrar, consultar atributos, generar mapas temáticos vistosos Cloropletas, heat Maps, con diferentes opciones, crear etiquetas, entre otros, las opciones pueden variar según el proceso que se requiera ejecutar.

En la siguiente figura se puede visualizar los militantes del partido:

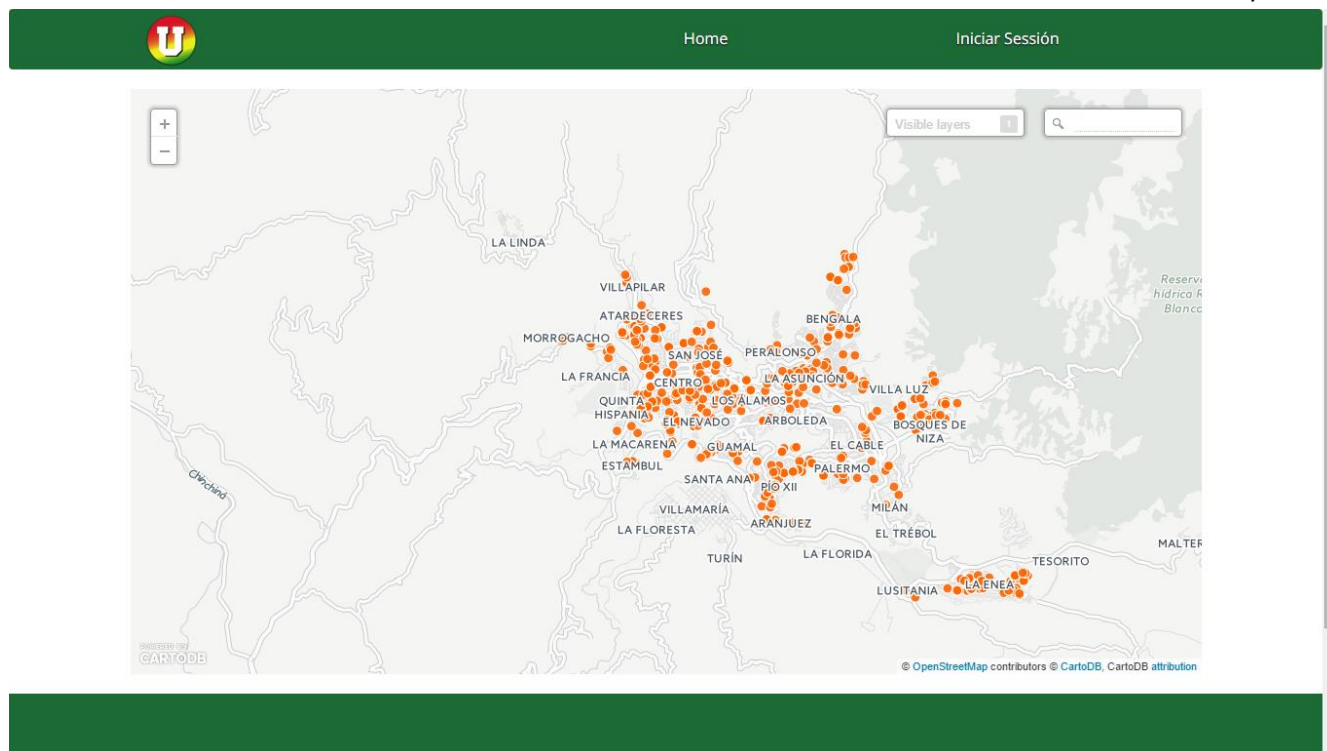


Figura 1: Pantalla principal en la que se pueden ver los militantes del partido desde una perspectiva general (Tomado de: (Bermúdez Martínez 2016)).

Uso del Índice de Moran y LISA para explicar el ausentismo electoral rural en Ecuador (Bucheli 2019).

En Ecuador, el voto es un derecho y obligación que constituye una parte esencial en la organización democrática al momento de escoger sus representantes que gobernarán el país, es así que dentro del marco geográfico electoral es importante analizar el ausentismo electoral especialmente en áreas rurales, cuyos escenarios son más propensos a este fenómeno debido a que existen varias dificultades en el acceso a recintos electorales cercanos, lo que acarrea largas distancias de transporte y altos costos para movilizarse. Por esta razón El Consejo Nacional Electoral del Ecuador ha creado jurisdicciones denominadas “zonas electorales rurales” con el fin de facilitar el acceso al voto.

Por lo tanto, este análisis recopila datos estadísticos y espaciales de las zonas electorales rurales creadas en los años 2009 y 2014 para entender la dinámica del porqué se crean zonas electorales y cómo éstas inciden en el ausentismo electoral.

Para analizar estos datos se recurre al programa GeoDa que maneja amplios formatos de información geográfica disponible y permite trabajar con la cobertura geográfica del ausentismo electoral de los años 2009 y 2014.

El resultado tangible radica en la visualización del comportamiento mediante la generación de dos tipos de mapas. El primero se denomina mapa de significancia partiendo de un procedimiento de aleatorización se muestra para cada unidad espacial la probabilidad de que sus relaciones de contigüidad se produzcan de manera aleatoria. El segundo mapa se denomina de agrupamiento o clúster. En este se observa como cada unidad espacial se diferencia de sus unidades espaciales vecinas

Geografía Electoral de la Ciudad Autónoma de Buenos Aires, 2015. Elecciones a Jefe de Gobierno municipal y a Presidente de la República Argentina

El 2015 fue un año electoral en la República Argentina. Se llevó a cabo la elección a presidente de la Nación junto a la elección legislativa para el Congreso Nacional y parlamentarias del Parlasur (Parlamento del Mercosur). En la mayoría de las provincias (21 de 23) hubo elección a Gobernador y en la Ciudad Autónoma de Buenos Aires (CABA) se eligió Jefe de Gobierno. En este contexto, se decidió analizar el comportamiento electoral en la CABA, principal ciudad de la República Argentina.

El día 25 de octubre de 2015 se realizó la elección a la Presidencia de la República Argentina, luego de haberse definido las Elecciones Primarias y Obligatorias (PASO) a partir de las cuales fueron habilitados aquellos candidatos ganadores en sus espacios políticos y cuyos partidos hayan obtenido un 1,5% o más de los votos (Sendra y Buzai 2017). En la Tabla 1 se pueden apreciar los resultados:

Tabla 1: Ciudad Autónoma de Buenos Aires. Resultado de las elecciones generales (Tomado de:(Sendra y Buzai 2017)).

| Partidos | Fórmula | Voto | Voto% |
|------------------------------------|--|---------|-------|
| Propuesta Republicana (PRO) | Horacio Rodríguez Larreta – Diego Santilli | 818.964 | 45,50 |
| Energía Ciudadana Organizada (ECO) | Martín Lousteau – Fernando Sánchez | 458.902 | 25,50 |

| | | | |
|---|-----------------------------------|---------|-------|
| Frente para la Victoria (FPV) | Mariano Recalde – Leandro Santoro | 394.541 | 21,90 |
| Autodeterminación y Libertad (AyL) | Luis Zamora – Sergio Sallustio | 71.223 | 4,00 |
| Frente de Izquierda de los trabajadores (FIT) | Myriam Bergman – José Castillo | 56.038 | 3,10 |
| | Votos en blanco y nulos | 33.870 | 1,90 |

Se analiza con apoyo de los SIG el comportamiento de la elección general a presidente de la República Argentina en la CABA como se puede apreciar en la Figura1:

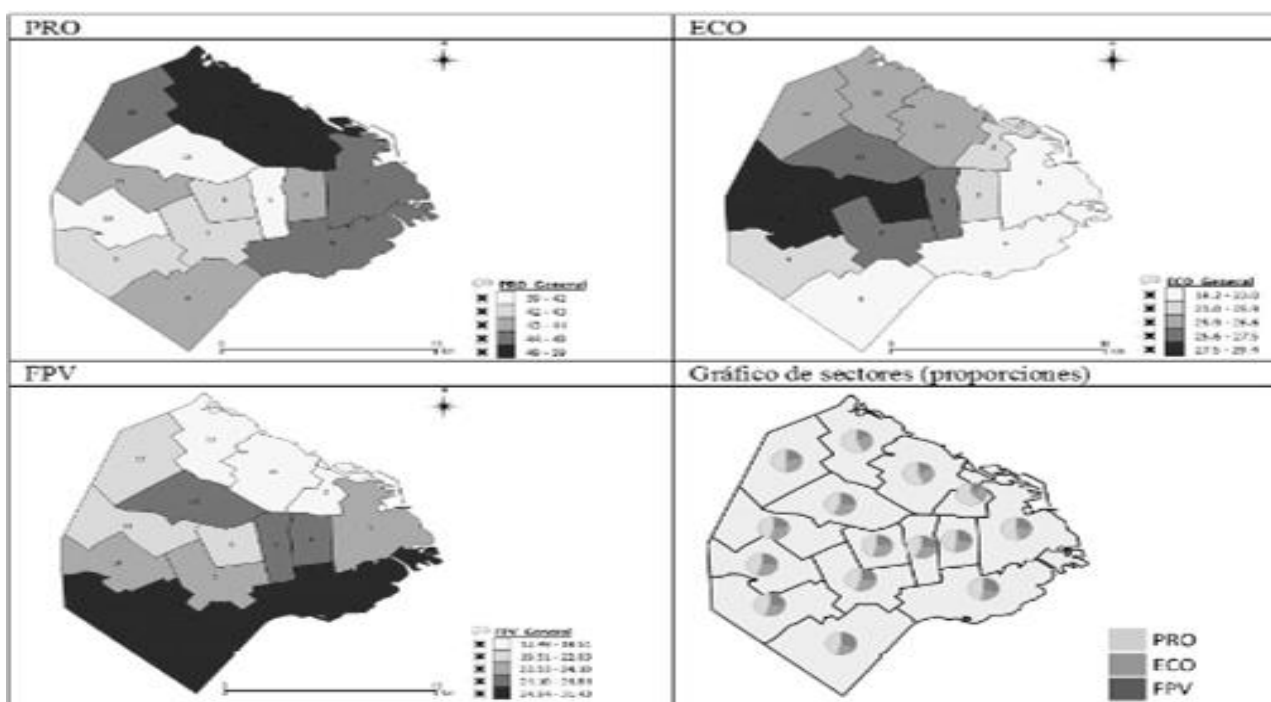


Figura 2: Distribución espacial de los votos en la elección general a Jefe de Gobierno de la CABA (Tomado de:(Sendra y Buzai 2017)).

1.2.2 Sistemas Homólogos a nivel nacional

SIGElectoral:

En nuestra Universidad ya se desarrolló un SIG para el sistema SIGEL_Generales llamado SIGElectoral sobre la plataforma GeneSIG. Este sistema se utilizó para conocer información confiable y actualizada sobre el estado de las votaciones en el país, es decir, conocer la

cantidad de electores que han ejercido el voto y los diputados que han sido electos. El sistema cuenta con una serie de funcionalidades propias de los SIG que permiten la navegación en el mapa: Acercar, Alejar, Recentrar, Mover y Ver todo el mapa. También se desarrolló una serie de funcionalidades específicas para este SIG. Algunas de estas son:

- **Actualizar información:** Permite analizar la información que se muestra en el mapa, para de esta forma poder consultar los datos actualizados de las votaciones en todo el país.
- **Información puntual:** Permite consultar la información de los diputados electos a nivel provincial y municipal.
- **Mostrar gráficos con el resumen de los electores a nivel provincial:** Permite obtener un gráfico de barras con el resumen de la cantidad de electores que han votado a nivel provincial.
- **Mostrar tematización de colores de la cantidad de electores por provincias:** Permite obtener un mapa temático donde, mediante colores, se muestre la cantidad de electores que ya han votado en cada una de las provincias del país. Se mostrará además el porcentaje que representa dicha cantidad del total de electores. Esta tematización puede ser realizada también a nivel de municipio.
- **Mostrar tematización de colores de la cantidad de diputados electos por provincia:** Permite obtener un mapa temático donde, mediante colores, se muestre la cantidad de diputados que superan o igualan el 60% de los votos en cada una de las provincias del país. Se mostrará además el porcentaje que representa dicha cantidad del total de diputados. Esta tematización puede ser realizada también a nivel de municipio.
- **Mostrar tematización de barras de la cantidad de electores por provincias:** Permite obtener un mapa temático donde, mediante barras, se muestre la relación del total de electores contra los que ya han votado en cada una de las provincias del país. Se mostrará además el porcentaje que representa dicha cantidad del total. Esta tematización también puede ser realizada a nivel de municipio.
- **Mostrar tematización de barras de la cantidad de diputados electos por provincia:** Permite obtener un mapa temático donde, mediante barras, se muestre la relación de la cantidad total de diputados contra la cantidad de diputados que superen o igualen el

60% de los votos en cada una de las provincias del país. Se mostrará además el porcentaje que representa dicha cantidad del total de diputados. Esta tematización también puede ser realizada a nivel de municipio.

En la Figura 2 se muestra como quedaron recogidas las funcionalidades en la barra de herramientas del sistema:



Figura 3: Funcionalidades (Tomado de: (Ledea, Aguila, y Griff 2013)).

Resultado del estudio y análisis de soluciones existentes:

El estudio realizado sobre sistemas homólogos evidencia las ventajas del uso de datos geoespaciales para realizar análisis estadísticos de los procesos electorales.

En la investigación Sistema de Información Geográfica de población electoral y control de solicitudes, se destaca la eficacia de los SIG a la hora de analizar la información referente a las personas relacionadas con el proceso. Por otra parte, en Geografía Electoral de la Ciudad Autónoma de Buenos Aires, 2015 se analiza con apoyo de los SIG, el comportamiento de la elección general a presidente de la República Argentina en la CABA.

En el artículo (Bucheli 2019) se realiza un análisis de la recopilación de datos estadísticos y espaciales de las zonas electorales rurales creadas en los años 2009 y 2014, permitiendo entender cómo éstas inciden en el ausentismo electoral, haciendo uso del Índice de Moran y LISA. SIGElectoral, desarrollado sobre la base de la plataforma GeneSIG, muestra información confiable y actualizada sobre el estado de las votaciones en el país; permite conocer la cantidad de electores que han ejercido el voto y los diputados que han sido electos.

El análisis bibliográfico presentado resalta los beneficios del uso de SIG para el seguimiento y control de procesos electorales. Los sistemas estudiados sirven de guía para determinar las principales funcionalidades que debe tener la solución propuesta, determinándose el uso de tematizaciones de colores analizando cantidad y porcentaje.

Tras concluir el estudio de algunas de las soluciones existentes abordadas anteriormente se evidencia las ventajas del análisis geoespacial de los procesos electorales para realizar análisis estadísticos y la toma de decisiones.

1.3 Metodología del desarrollo de software

Una metodología de desarrollo de sistemas o software se puede definir como una colección documental de políticas, procesos y procedimientos para mejorar el proceso de desarrollo de software con respecto al aumento de la productividad del personal de tecnología de la información (TI) u una mayor calidad de las soluciones de TI finales (Batra 2020).

Para guiar el proceso de desarrollo del SIG se utilizará la metodología AUP² en su variante UCI, que es utilizada actualmente en todos los proyectos que se desarrollan en los centros de producción de la universidad. Esta variante, de las cuatro fases que propone AUP (Inicio, Elaboración, Construcción, Transición), decide mantener la fase de inicio, pero modificando el objetivo de la misma, se unifican las 3 fases restantes de AUP en una sola titulada Ejecución y se agrega la fase de Cierre (Sánchez 2015).

Para modelar el sistema de la solución a desarrollar fue elegido el escenario No. 4. Para esta elección se tuvo en cuenta que el cliente (en este caso los especialistas del centro CREAD³) estará siempre acompañando al desarrollador para acordar los detalles de los requisitos y así poder implementarlos, probarlos y validarlos. No se realiza modelado del negocio, pues una vez evaluado este, se concluye que se encuentra bien definido. El proyecto no será extenso y están bien definidos los elementos a implementar, por lo que las historias de usuario ayudarán a describir los requisitos de manera concreta.

1.4 Herramientas y tecnologías a utilizar para el desarrollo del SIG

Lenguaje de Modelado Unificado v2.0.

El “lenguaje de modelado unificado” - del inglés Unified Modeling Language (UML) - es un lenguaje de modelado estandarizado que consta de un conjunto integrado de diagramas, desarrollado para ayudar a los desarrolladores de sistemas y software a especificar, visualizar, construir y documentar los artefactos de los sistemas de software, así como para el modelado de negocios y otros sistemas que no son de software. El UML representa una compilación de las prácticas exitosas en la ingeniería, específicamente en el modelado de sistemas amplios y complejos (Jacobson y Booch 2021).

Visual Paradigm for UML v5.0

² Proceso Unificado Ágil

³ Centro de Representación y Análisis de Datos

Visual Paradigm es una herramienta UML CASE compatible con UML 2, SysML y notación de modelado de procesos comerciales del grupo de gestión de objetos. Además del soporte de modelado, proporciona capacidades de ingeniería de código y generación de informes, incluida la generación de código. Puede aplicar ingeniería inversa a los diagramas del código y proporcionar ingeniería de ida y vuelta para varios lenguajes de programación (Rahwanto 2022).

PostgreSQL v14

PostgreSQL es un sistema de gestión de base de datos objeto – relación, distribuido bajo la licencia BSD y con código fuente disponible libremente. Es el sistema de gestión de base de datos de código abierto más potente del mercado. Utiliza un modelo cliente/servidor y utiliza multiprocesos en vez de multihilos para garantizar la estabilidad del sistema. Un fallo en uno de los procesos no afectará al resto y el sistema continuara funcionando (Ordóñez, Ríos, y Castillo 2017).

PostGIS v3.0.3

PostGis es un software compatible con Open Geospatial Consortium (OGC) utilizado como una extensión para PostgreSQL. PostGIS amplía las capacidades de PostgreSQL a fin de aumentar sus capacidades de gestión mediante la adición de tipos y funciones geoespaciales para mejorar los datos espaciales manejados dentro de una estructura de base de datos relacional. El lenguaje de PostGIS es similar al SQL y permite realiza análisis espaciales y consultas típicas sobre datos espaciales con relativa facilidad. Esto lo convierte en un Backend (quien se encarga de interactuar con base de datos, lenguajes, y usuarios) relativamente potente para bases de datos dentro de un software más grandes (Hsu y Obe 2021).

OpenLayers v4.1.1

OpenLayers es un paquete de biblioteca de clases de JavaScript para desarrollo de clientes WebGIS, que se utiliza para redes acceso de datos de mapas. Como proyecto de código abierto, es diseñado para proporcionar una potente función de visualización de mapas para clientes de Internet, incluida la visualización de datos de mapas y otras operaciones, con mecanismo de extensión flexible. Se basa en ActionScript y Flexb para la redacción, puede admitir varios servicios cartográficos estándar, como WMS, WFS, WMTS, OSM, etc. Una de las ventajas vitales de OpenLayers en el monitoreo de información marina es que puede hacer un uso completo de la tecnología HTML5, dar juego completo a las ventajas de HTML5, adaptarse a

la continua mejora del rendimiento del equipo cliente, y reducir la presión del procesamiento de páginas del lado del servidor (He et al. 2019).

Express.js v4.16.3

Express es un framework de aplicaciones web de código abierto, minimalista y flexible, alojado en el entorno de ejecución Node.js. Viene equipado con un conjunto versátil de funcionalidades para construir y diseñar aplicaciones web y móviles. Express es uno de los marcos web más populares para Node.js hasta la fecha, que fue escrito en JavaScript y es cómodo para los programadores para aprender y utilizar («Express - Node.js Web Application Framework» s. f.; «What Is Express.js? | Why Should Use Express.js? | Features of Express.js» 2020).

Node.js v16.18.0

En esencia, Node.js es un entorno de ejecución de JavaScript del lado del servidor de código abierto que es capaz de funcionar en varias plataformas. Fue construido sobre el motor V8 de Chrome, lo que le permite ejecutar código JavaScript fuera del navegador. El uso de Node.js es muy versátil, y es adecuado para la programación de propósito general, por lo que los desarrolladores de software pueden utilizar Node.js para crear casi cualquier tipo de programa, desde servidores web, juegos, scripts, aplicaciones CLI, etc.(Herron 2018).

Visual Studio Code v1.51.1

Visual Studio Code es un editor de código fuente ligero pero potente que se ejecuta en el escritorio y está disponible para Windows, macOS y Linux. Viene con soporte incorporado para JavaScript, TypeScript y Node.js y tiene un rico ecosistema de extensiones para otros lenguajes (como C++, C#, Java, Python, PHP, Go) y tiempos de ejecución (como .NET y Unity) («Documentation for Visual Studio Code» s. f.).

MapServer v6.4

MapServer es una plataforma de código abierto para publicar datos espaciales y aplicaciones cartográficas interactivas en la web. Desarrollado originalmente a mediados de los años 90 en la Universidad de Minnesota, MapServer se publica bajo una licencia del tipo MIT y funciona en las principales plataformas (Windows, Linux, Mac OS X). MapServer no es un sistema SIG completo, ni aspira a serlo. En la Galería del Sitio encontrará ejemplos de MapServer en acción (McKenna, Jeff y MapServer PSC 2021).

Apache JMeter v2.3.4

La aplicación Apache JMeter es un software de código abierto, una aplicación 100% pura de Java diseñada para cargar el comportamiento funcional y medir el rendimiento. Se diseñó originalmente para probar aplicaciones web, pero desde entonces se ha ampliado a otras funciones de prueba («Apache JMeter - Apache JMeter™» s. f.).

1.4.1 Lenguajes a utilizar

HTML5

HTML5 (HyperText Markup Language, versión 5) es la quinta versión del lenguaje de maquetado HTML. Esta versión define los nuevos estándares de desarrollo web, rediseñando el código para resolver problemas y actualizándolo así a nuevas necesidades. Es independiente de la plataforma utilizada y se basa fundamentalmente en el uso de etiquetas estructurales y semánticas, adecuadas para la creación de documentos relativamente simples que permiten simplificar su estructura. Este realiza mejoras en los formularios con nuevos tipos de datos (eMail, number, url, datetime) y facilidades para validar el contenido sin Javascript. Esta versión establece una serie de nuevos elementos y atributos que reflejan el uso típico de los sitios web modernos (Theisen 2019).

CSS3

CSS es un lenguaje simple que describe cómo se va a mostrar un documento en la pantalla. Se utiliza para dar estilo a documentos HTML y XML, separando el contenido de la presentación. Permite controlar el estilo y el formato de múltiples páginas Web al mismo tiempo. Cualquier cambio en el estilo marcado para un elemento en la CSS afectará a todas las páginas vinculadas a ella en la que aparezca ese elemento. CSS funciona a base de reglas, es decir, declaraciones sobre el estilo de uno o más elementos. Las hojas de estilo están compuestas por una o más de esas reglas aplicadas a un documento HTML o XML. La regla tiene dos partes: un selector y la declaración (Rebah, Boukthir, y Chedebois 2022).

JavaScript v1.8.5

JavaScript es un lenguaje de programación que permite a los desarrolladores interactuar con la funcionalidad proporcionada por los navegadores web. Más concretamente, JavaScript es un lenguaje de scripting, lo que significa que tradicionalmente, el código fuente de JavaScript se interpreta en tiempo de ejecución y no se precompila en código de bytes y en la práctica, su principal objetivo es modificar el comportamiento de otra aplicación escrita normalmente en

un lenguaje de programación diferente, en el que se interpreta y ejecuta en tiempo real (Theisen 2019).

Conclusiones del capítulo

El estudio de los principales conceptos relacionados con el desarrollo de un SIG permitió sentar las bases para el desarrollo de la solución y la necesidad de la realización del presente trabajo. La búsqueda y el análisis de los SIG que se utilizan para el análisis georreferenciado y la toma de decisiones, evidenciaron que el uso de los mismos brinda un conjunto de posibles funcionalidades, pero no constituyen una solución para la problemática descrita. La selección de la metodología AUP-UCI en su escenario 4 y el uso del lenguaje de modelado UML, con apoyo de las herramientas CASE Visual Paradigm para el desarrollo del módulo, permitió la creación del entorno de trabajo para el desarrollo de la solución. Además, la utilización del IDE Visual Studio Code y los lenguajes JavaScript, HTML5 y CSS3 ayudaran a obtener una solución acorde a las exigencias de los sitios web actuales.

CAPÍTULO 2: DISEÑO DE LA SOLUCIÓN PROPUESTA AL PROBLEMA CIENTÍFICO

El desarrollo de software que este trabajo propone sigue el proceso de análisis y diseño que proporciona las bases sobre las cuales se va a desarrollar la aplicación. Es por esto que en este capítulo se va a detallar los procesos de ingeniería de software, análisis y diseño que se involucran en el desarrollo de la aplicación Web. Se exponen como artefactos que permiten describir la propuesta de solución: la especificación de requisitos funcionales y no funcionales, las historias de usuario, y el modelo de diseño. Se definen otros aspectos como los patrones de diseños a utilizar, así como el modelo entidad relación de la base de datos y el diagrama de despliegue.

2.1 Descripción del sistema propuesto

El Sistema de Información Geográfica para **decidimOS** tiene como objetivo tomar la información referente a los Referendos. De esta información se utilizará la relacionada con las estructuras electorales y plebiscitos de la base de datos relacionada a las elecciones. Con esta información el sistema creará varias tematizaciones en el mapa de acuerdo a lo que el usuario necesite en ese momento.

2.2 Requisitos del sistema

En el desarrollo de un software que esté en línea con las expectativas del negocio, se necesita en primer lugar de un proceso de Ingeniería de Requisitos apropiado, donde se deben realizar las cuatro tareas comunes de la Ingeniería de Requisitos: estudio de factibilidad, obtención y análisis de requisitos, especificación de requisitos y validación de requisitos (Sommerville 2016).

2.2.1 Requisitos Funcionales

Según (Sommerville 2016), los Requisitos Funcionales son declaraciones de los servicios que debe proporcionar el sistema, de la manera en que éste debe reaccionar a entradas particulares y de cómo se debe comportar en situaciones particulares. En algunos casos, los

requerimientos funcionales de los sistemas también pueden declarar explícitamente lo que el sistema no debe hacer.

Funcionalidades específicas del SIG: Estos requisitos forman parte de la plataforma base ULTRON y ya se encuentran implementados en esta.

RF 1: Acercar determinada región del mapa: Permite aumentar el tamaño del mapa, mostrando en pantalla la región donde el usuario realice la operación de acercar.

RF 2: Alejar determinada región del mapa: Permite disminuir el tamaño del mapa, mostrando en pantalla la región donde el usuario realice la operación de alejar.

RF 3: Reiniciar vista del mapa: Permite visualizar el mapa según la escala inicial de la aplicación. Esta funcionalidad permite poner el mapa en su extensión original, si el mapa ya aparece así no se notará la acción.

RF 4: Visualizar mapa en pantalla completa: Permite visualizar el mapa en la pantalla completa del monitor.

RF 5: Visualizar diferentes regiones del mapa: Permite mover el mapa para visualizar otras regiones.

RF 6: Navegar a través del mapa de referencia: Permite visualizar otras regiones en el mapa principal, haciendo uso de un Mapa de Referencia, para posicionarse en el lugar indicado en este último. El sistema permite ocultar o mostrar el mapa de referencia.

RF 7: Visualizar control de capas: Permite visualizar u ocultar la ventana que muestra las capas del mapa, en dependencia del estado inicial en el que se encuentre la misma.

RF 8: Expandir nodos del árbol de capas: Permite expandir los nodos del árbol de capas, para que se visualicen todos sus elementos.

RF 9: Contraer nodos del árbol de capas: Permite contraer los nodos del árbol de capas, para ocultar sus elementos.

RF 10: Habilitar capas del mapa: Permite habilitar las capas del mapa que se encuentren deshabilitadas.

RF 11: Deshabilitar capas del mapa: Permite deshabilitar las capas del mapa que se encuentren habilitadas.

RF 12: Exportar mapa: Permite exportar (formato .png) el mapa que se encuentra visualizado con la información correspondiente.

En el caso de los requisitos asociados al proceso electoral se agrupan en dos grupos **Referendo (Estructuras Electorales)** y **Referendo (Plebiscitos)**.

Estructuras Electorales:

RF 13: Mostrar tematización de colores de la cantidad de circunscripciones ordinarias por provincias: Permite obtener un mapa temático donde, mediante colores, se muestre la cantidad de circunscripciones ordinarias por provincias.

RF 14: Mostrar tematización de colores de la cantidad de circunscripciones especiales por provincias: Permite obtener un mapa temático donde, mediante colores, se muestre la cantidad de circunscripciones especiales por provincias.

RF 15: Mostrar tematización de colores de la cantidad de electores por provincias: Permite obtener un mapa temático donde, mediante colores, se muestre la cantidad de electores por provincias.

RF 16: Mostrar tematización de colores de la cantidad de colegios a constituir por provincias: Permite obtener un mapa temático donde, mediante colores, se muestre la cantidad de colegios a constituir por provincias.

RF 17: Mostrar tematización de colores de la cantidad de colegios a constituir en locales particulares por provincias: Permite obtener un mapa temático donde, mediante colores, se muestre la cantidad de colegios a constituir en locales particulares por provincias.

RF 18: Mostrar tematización de colores de la cantidad de grupos auxiliares de procesamiento de la información (GAPI) por provincias: Permite obtener un mapa temático donde, mediante colores, se muestre la cantidad de grupos auxiliares de procesamiento de la información (GAPI) por provincias.

RF 19: Mostrar tematización de colores de la cantidad de autoridades electorales por provincias: Permite obtener un mapa temático donde, mediante colores, se muestre la cantidad de autoridades electorales por provincias.

Plebiscitos:

RF 20: Mostrar tematización de colores del porcentaje de electores que votaron según parte básico por provincias: Permite obtener un mapa temático donde, mediante colores, se muestre el porcentaje de electores que votaron según parte básico por provincias (- 35% mal, entre 35% y 70% regular y mayor que 70% bien).

RF 21: Mostrar tematización de colores de la cantidad de electores nuevos inscriptos en la lista por provincias: Permite obtener un mapa temático donde, mediante colores, se muestre la cantidad de electores nuevos inscriptos en la lista por provincias.

RF 22: Mostrar tematización de colores de la cantidad de electores con inscripciones excepcionales por provincias: Permite obtener un mapa temático donde, mediante colores, se muestre la cantidad de electores con inscripciones excepcionales por provincias.

RF 23: Mostrar tematización de colores de la cantidad de electores excluidos el día de la elección por fallecimiento por provincias: Permite obtener un mapa temático donde, mediante colores, se muestre la cantidad de electores excluidos el día de la elección por fallecimiento por provincias.

RF 24: Mostrar tematización de colores de la cantidad de electores en la lista actualizada por provincias: Permite obtener un mapa temático donde, mediante colores, se muestre la cantidad de electores en la lista actualizada por provincias.

RF 25: Mostrar tematización de colores del porciento de electores que han votado según boletas depositadas en urnas por provincias: Permite obtener un mapa temático donde, mediante colores, se muestre el porciento de electores que han votado según boletas depositadas en urnas por provincias (- 35% mal, entre 35% y 70% regular y mayor que 70% bien).

RF 26: Mostrar tematización de colores de la cantidad de colegios por provincias: Permite obtener un mapa temático donde, mediante colores, se muestre la cantidad de colegios por provincias.

RF 27: Mostrar tematización de colores del porciento de colegios con problemas por provincias: Permite obtener un mapa temático donde, mediante colores, se muestre el porciento de colegios con problemas por provincias (- 35% mal, entre 35% y 70% regular y mayor que 70% bien).

RF 28: Mostrar tematización de colores de la cantidad de colegios funcionando según primer parte por provincias: Permite obtener un mapa temático donde, mediante colores, se muestre la cantidad de colegios funcionando según primer parte por provincias.

RF 29: Mostrar tematización de colores del porciento de boletas depositadas en urnas según calidad del voto por provincias: Permite obtener un mapa temático donde, mediante

colores, se muestre el porcentaje de boletas depositadas en urnas según calidad del voto por provincias (- 35% mal, entre 35% y 70% regular y mayor que 70% bien).

RF 30: Mostrar tematización de colores del porcentaje de boletas en blanco por provincias: Permite obtener un mapa temático donde, mediante colores, se muestre el porcentaje de boletas en blanco por provincias (- 35% mal, entre 35% y 70% regular y mayor que 70% bien).

RF 31: Mostrar tematización de colores del porcentaje boletas anuladas por provincias: Permite obtener un mapa temático donde, mediante colores, se muestre el porcentaje de boletas anuladas por provincias (- 35% mal, entre 35% y 70% regular y mayor que 70% bien).

RF 32: Mostrar tematización de colores del porcentaje de boletas válidas por provincias: Permite obtener un mapa temático donde, mediante colores, se muestre el porcentaje de boletas válidas por provincias (- 35% mal, entre 35% y 70% regular y mayor que 70% bien).

RF 33: Mostrar tematización de colores del total de votos por provincias: Permite obtener un mapa temático donde, mediante colores, se muestre el total de votos por provincias.

RF 34: Mostrar tematización de colores del porcentaje de votos SI por provincias: Permite obtener un mapa temático donde, mediante colores, se muestre el porcentaje de votos SI por provincias (- 35% mal, entre 35% y 70% regular y mayor que 70% bien).

RF 35: Mostrar tematización de colores del porcentaje de votos NO por provincias: Permite obtener un mapa temático donde, mediante colores, se muestre el porcentaje de votos NO por provincias (- 35% mal, entre 35% y 70% regular y mayor que 70% bien).

2.2.2 Requisitos no Funcionales

Citando al mismo Sommerville, los requisitos no funcionales se refieren a las características que de una u otra forma pueda limitar el sistema. Describen una restricción sobre el sistema que limita la elección del usuario en la construcción de una solución (Sommerville 2016).

RNF 1 Usabilidad: El objetivo de este requisito es definir la usabilidad del sistema, para ello se describen las características del usuario final que lo utilizará, el tipo de aplicación informática, el objetivo que persigue la aplicación y las características de hardware y software requeridas para desplegar y utilizar la aplicación.

- **RNF 1.1:** Debido a las acciones que se realizarán en la aplicación, los usuarios finales deberán tener conocimientos básicos de Informática.

- **RNF 1.2:** 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.
- **RNF 1.3:** Las funcionalidades principales del sistema estarán orientadas a íconos para un mayor reconocimiento por parte del usuario.
- **RNF 1.4:** Los servidores de Base de Datos y de Mapas deberán tener las siguientes características:

Hardware:

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

Software:

- Sistema Operativo: GNU/Linux Debian
- NodeJs 12.13.1 como manejador de dependencia para JavaScript
- PostgreSQL 12.5 como Sistema Gestor de Base de Datos.
- PostGIS 3.0.3 como extensión de PostgreSQL como soporte de datos espaciales.
- pgRouting 2.3 como extensión de PostgreSQL para el cálculo de rutas
- **RNF 1.5:** 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, Mozilla Firefox, Safari que cumpla con los estándares W3C y cuente con soporte para CSS3 y HTML5.

RNF 2 Confiabilidad: El objetivo de este requisito es describir los aspectos que garantizarán la seguridad de la información que se maneja en el sistema propuesto, evitando así la pérdida y divulgación de datos.

- **RNF 2.1:** 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.
- **RNF 2.2:** Si se interrumpe la energía en el servidor, una vez reanudada, el sistema deberá ser capaz de iniciar de forma automática los servicios.
- **RNF 2.3:** Si se interrumpe la energía en el cliente, una vez reanudadas las operaciones, el usuario deberá autenticarse nuevamente para acceder al sistema.
- **RNF 2.4:** El sistema debe hacer copias de respaldo periódicamente y hacer uso de las mismas una vez ocurrido un fallo en la base de datos.

RNF 3 Eficiencia: Este requisito tiene como objetivo definir las características de la eficiencia del sistema, la velocidad de procesamiento y los tiempos de respuestas de la aplicación al realizar cualquier operación.

- **RNF 3.1:** La velocidad de procesamiento de la información y el tiempo de respuesta dependerán de la cantidad de información a procesar.

RNF 4 Restricciones de diseño: El objetivo de este requisito es definir las restricciones de diseño del sistema a desarrollar.

- **RNF 4.1:** Para la modelación se utilizará como lenguaje de modelado UML 2.1 y como herramienta el Visual Paradigm for UML 5.0.
- **RNF 4.2:** El software debe diseñarse sobre una arquitectura cliente-servidor.
- **RNF 4.3:** Se deben emplear los estándares establecidos para diseño de interfaces, base de datos y codificación.
- **RNF 4.4:** Lograr un producto altamente configurable y extensible. La aplicación estará diseñada para su correcta visualización en distintos tipos de dispositivos (Móvil, Tablet, PC).

RNF 5 Documentación de usuarios en línea y ayuda del sistema: El objetivo de este requisito es definir la forma en que el usuario obtendrá documentación sobre el uso del sistema.

- **RNF 5.1:** El sistema deberá brindar a sus clientes finales un manual de usuario en el que se especificarán todas las funciones que pueden realizar y cómo hacerlo.

RNF 6 Interfaz: El objetivo de este requisito es definir las características de las interfaces que tendrá la aplicación.

RNF 6.1: La interfaz de usuario deberá tener apariencia profesional y diseño gráfico sencillo.

RNF 6.2: Tendrá pocas entradas de datos para obtener información, logrando así que no sea necesario mucho entrenamiento para utilizar el sistema.

RNF 6.3: Las funcionalidades a realizar deben representarse con íconos intuitivos.

RNF 6.4: El sistema debe brindar la posibilidad de ocultar los paneles laterales para visualizar el mapa en toda la amplitud de la pantalla.

RNF 6.5: El sistema debe tener un paginado para cada listado de elementos, de forma tal que se pueda mostrar solo una cantidad determinada de elementos por página.

RNF 6.6: Contará con un campo de búsqueda en las interfaces que lo requieran, posibilitando que el usuario pueda realizar búsquedas de información desde cualquier vista.

RNF 7 Soporte

- **RNF 7.1:** El equipo de desarrollo deberá ofrecer la capacitación en el uso y mantenimiento del sistema al personal designado en la entidad que sea desplegado

RNF 8 Licencia: El objetivo de este requisito es definir cualquier requisito de licencia o restricción de uso que serán seguidos por el software.

- **RNF 8.1:** De acuerdo a los tipos de licencias de los componentes y herramientas que se proponen para el desarrollo del producto, se puede catalogar legalmente esta arquitectura de modelo libre, permitiendo la utilización, modificación y distribución de las mismas por terceros sin necesidad de obtener la autorización de sus respectivos titulares.

RNF 9 Legales, de derecho de autor y otros: El objetivo de este requisito es describir cualquier denegación legal necesaria, garantías, notificaciones de derecho de autor, patentes, marca comercial o complacencia con logotipo para el software.

- **RNF 9.1:** El sistema debe ajustarse y regirse por la ley, decretos leyes, decretos, resoluciones y manuales (órdenes) establecidos, que norman los procesos que serán automatizados.
- **RNF 9.2:** La mayoría de las herramientas de desarrollo son libres y del resto, las licencias están avaladas.

RNF 10 Estándares aplicables: El objetivo de este requisito es describir por referencia cualquier norma o estándar aplicable y las secciones específicas que aplican al sistema.

- **RNF 10.1:** El sistema será desarrollado bajo estándares OpenGIS como aseguramiento de la parte científica y en el desarrollo se codificará y modelará siguiendo los patrones de las normativas ISO.

2.3 Historias de usuarios

Las historias de usuarios son parte del desarrollo ágil de software que ayuda a enfocarse en hablar de los requerimientos en lugar de escribir acerca de ellos. Estas son cortas descripciones de una funcionalidad desde la perspectiva de la persona que la desea, usualmente un usuario o cliente (Cohn, 2018).

La solución propuesta consta de 35 requisitos funcionales de los cuales 12 ya se encuentran implementados. En correspondencia con la selección del escenario número cuatro de la metodología empleada se procede a modelar el sistema de historias de usuario, donde se define una por cada requisito funcional. A continuación, la presente descripción:

Tabla 2: Historial de usuario “Mostrar tematización de colores de la cantidad de circunscripciones ordinarias por provincias”.

| | |
|---|--|
| Número: 13 | Requisito: Mostrar tematización de colores de la cantidad de circunscripciones ordinarias por provincias. |
| Programador: Frank Alberto Plasencia Téllez | Interacción asignada: 1 |
| Prioridad: Alta | Tiempo Estimado: 3 |
| Riesgo en Desarrollo: <ul style="list-style-type: none"> • Planificación irreal. • Interrupción del servicio eléctrico. • Afectaciones externas al personal de trabajo. | Tiempo Real: 3 |

Descripción: En la interfaz se debe mostrar una tematización en el mapa que de acuerdo a la cantidad de circunscripciones ordinarias por provincia este cambie de color.

Observaciones:

Pueden realizar la funcionalidad los roles con el permiso correspondiente.

El usuario tiene que estar autenticado.

Selección de tematización: Opción que se representa a través del ícono (T). Permite seleccionar la funcionalidad de tematizar en el mapa.

Para su uso:

1. Seleccionar la funcionalidad de tematización.
2. Seleccionar la opción Por cantidad.
3. Escoger un indicador.


Respuesta del sistema: El mapa se pinta con un color en degradado mostrando los valores por provincia.

Prototipo elemental de interfaz gráfica de usuario:



Tabla 3: Historial de usuario “Mostrar tematización de colores del porcentaje de electores que votaron según parte básico por provincias”.

| | |
|--|--|
| Número: 20 | Requisito: Mostrar tematización de colores del porcentaje de electores que votaron según parte básico por provincias. |
| Programador: Frank Alberto Plasencia Téllez | Interacción Asignada: 1 |

| | |
|--|---------------------------|
| Prioridad: Alta | Tiempo Estimado: 3 |
| Riesgo en Desarrollo: <ul style="list-style-type: none"> • Planificación irreal. • Interrupción del servicio eléctrico. • Afectaciones externas al personal de trabajo. | Tiempo Real: 3 |
| Descripción: En la interfaz se debe mostrar una tematización en el mapa que de acuerdo al porcentaje de electores que votaron según parte básico por provincia este cambie de color de acuerdo a un rango (- 35% mal, entre 35% y 70% regular y mayor que 70% bien). | |
| Observaciones: Pueden realizar la funcionalidad los roles con el permiso correspondiente. El usuario tiene que estar autenticado. Selección de tematización: Opción que se representa a través del ícono (). Permite seleccionar la funcionalidad de tematizar en el mapa. Para su uso: <ol style="list-style-type: none"> 1. Seleccionar la funcionalidad de tematización. 2. Seleccionar la opción Por porcentaje. 3. Escoger un indicador. Respuesta del sistema: El mapa se pinta con un color (Rojo, Amarillo, Verde) mostrando los valores por provincia. | |
| Prototipo elemental de interfaz gráfica de usuario: | |

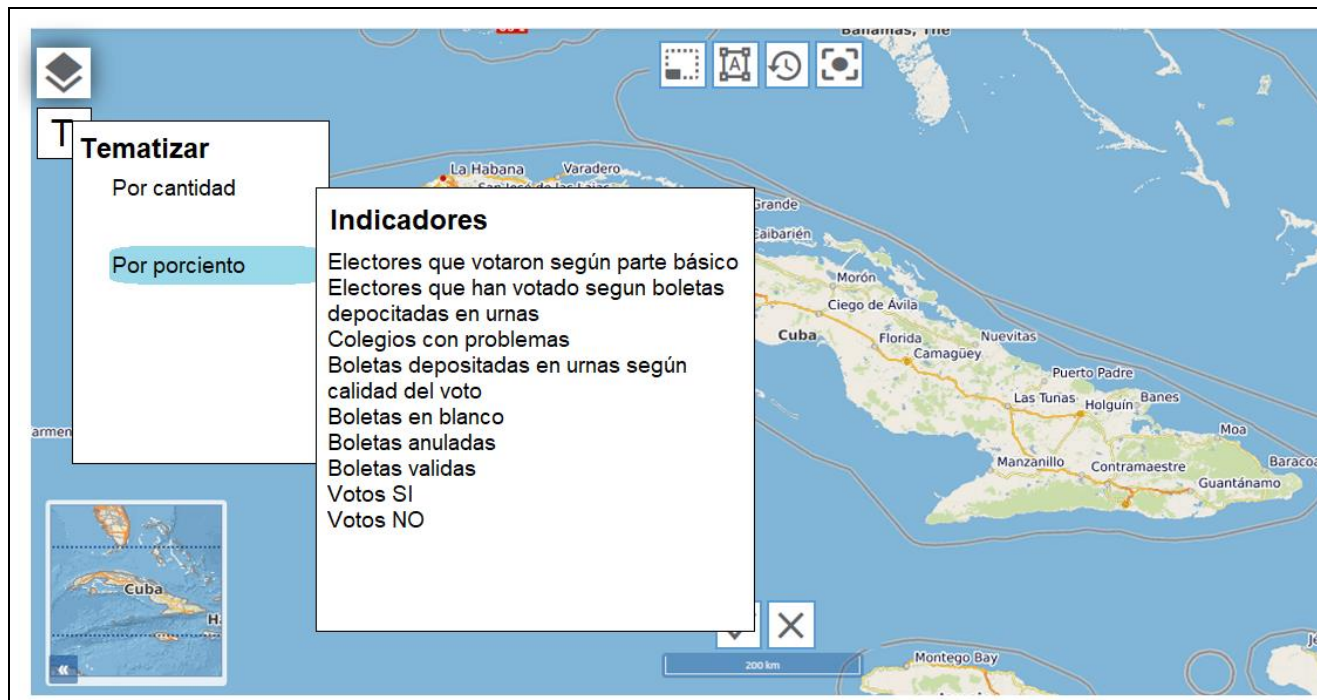


Tabla 4: Historial de usuario “Mostrar tematización de colores del porcentaje de votos NO por provincias”.

| | |
|---|--|
| Número: 35 | Requisito: Mostrar tematización de colores del porcentaje de votos NO por provincias. |
| Programador: Frank Alberto Plasencia Téllez | Interacción Asignada: 1 |
| Prioridad: Alta | Tiempo Estimado: 3 |
| Riesgo en Desarrollo: <ul style="list-style-type: none"> • Planificación irreal. • Interrupción del servicio eléctrico. • Afectaciones externas al personal de trabajo. | Tiempo Real: 3 |
| Descripción: En la interfaz se debe mostrar una tematización en el mapa que de acuerdo al porcentaje total de votos NO por provincia este cambie de color de acuerdo a un rango (-35% mal, entre 35% y 70% regular y mayor que 70% bien). | |
| Observaciones: Pueden realizar la funcionalidad los roles con el permiso correspondiente. El usuario tiene que estar autenticado. | |
| Selección de tematización: Opción que se representa a través del ícono (T). Permite seleccionar la funcionalidad de tematizar en el mapa. Para su uso: <ol style="list-style-type: none"> 1. Seleccionar la funcionalidad de tematización. 2. Seleccionar la opción Por porcentaje. | |

3. Escoger un indicador.

Respuesta del sistema: El mapa se pinta con un color (Rojo, Amarillo, Verde) mostrando los valores por provincia.

Prototipo elemental de interfaz gráfica de usuario:



2.4 Patrón de diseño arquitectónico

Los patrones de diseño arquitectónicos pueden ser vistos como los bloques constructivos básicos de las arquitecturas de software. Esto es, define y organizan un vocabulario de elementos basados en un conjunto de componentes y conectores asociados a un grupo de restricciones que indican la forma 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 (Blas, Leone, y Gonnet 2019).

El patrón arquitectónico seleccionado es Modelo Vista Controlador (MVC). El modelo MVC (Modelo, Vista, Controlador) es un esquema de arquitectura por capas muy utilizado en el desarrollo de software basado en aplicaciones web. Además, este patrón separa los datos de una aplicación, la interfaz de usuario y la lógica de control en tres componentes distintos de

forma que las modificaciones del componente de la vista puedan ser hechas con un mínimo de impacto en el componente del modelo de datos(Ávila Garzón 2019).

Modelo: Representa la información con la que trabaja la aplicación, o sea, su lógica de negocio.

Vista: Convierte el modelo en una página web que facilita al usuario interactuar con ella.

Controlador: Es el encargado de procesar las interacciones del usuario y ejecutar los cambios adecuados en el modelo o en la vista. El controlador es el encargado de aislar al modelo y a la vista de los detalles del protocolo usado para peticiones (HTTP, consola de comandos, email, etc.).

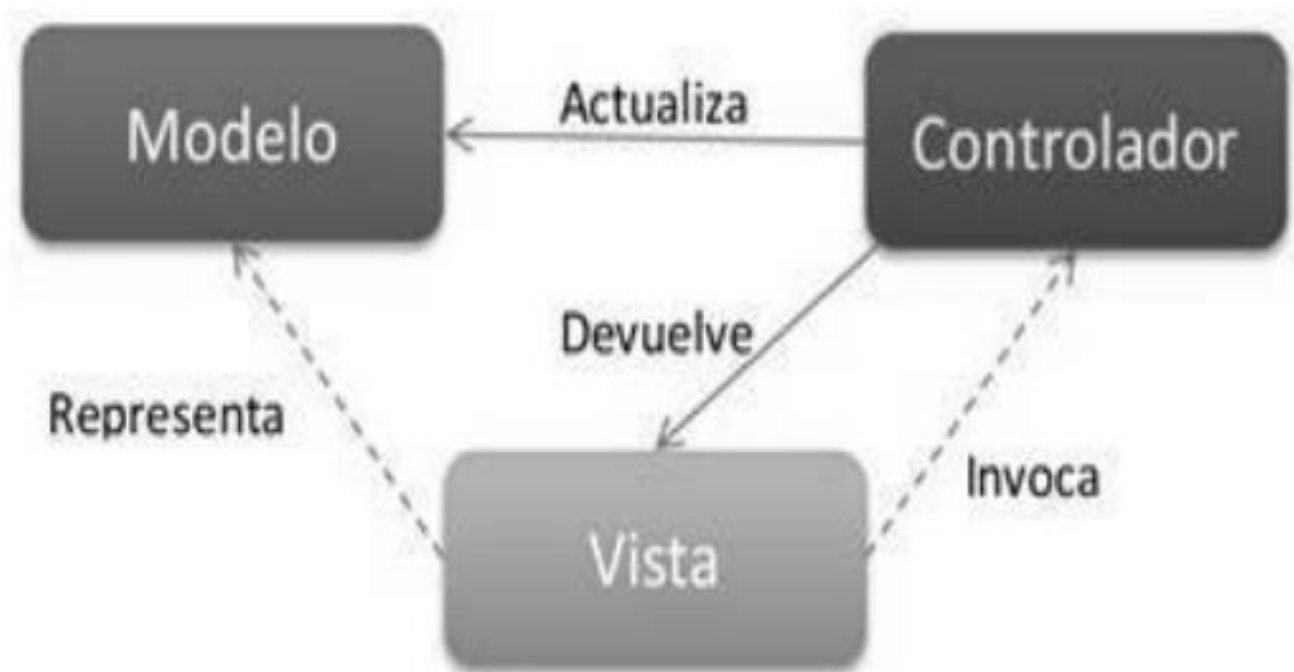


Figura 4: Representación del Modelo Vista Controlador.

2.5 Modelo de diseño

El diagrama de clases del diseño muestra los bloques de construcción de cualquier sistema orientado a objetos. Permite representar gráficamente y de manera estática la estructura general de un sistema, mostrando cada una de las clases y sus interacciones (como herencias y asociaciones), representadas en forma de bloques, los cuales son unidos mediante líneas y arcos(Hernández, Jiménez, y Godoy 2018). A continuación, se representa un diagrama de clases de diseño correspondiente al SIG para **decidimOS**.

2.6 Patrones para la asignación de responsabilidades (GRASP)

Se decide emplear para estructurar el diseño del sistema Patrones Generales de Software para Asignar Responsabilidades, más conocido como patrones GRASP (General Responsibility Assignment Software Patterns, por sus siglas en inglés), los cuales serán muy útiles para lograr un software con alto grado de usabilidad.

Describe los principios fundamentales de la asignación de responsabilidades a objetos expresados en forma de patrones (Ortega 2021). Para el desarrollo del sistema se hicieron uso de los siguientes patrones GRASP:

Experto (Expert): consiste en asignar una responsabilidad a la clase que cuenta con la información necesaria para cumplirla. Una clase contiene toda la información necesaria para realizar la labor que tiene encomendada. Se evidencia en clases como: LayerModel y Layer, debido que cada una de ellas es responsable de manejar su información.

Creador (Creator): permite crear objetos de una clase determinada. Es utilizado en la mayoría de las clases controladoras para crear instancias de formularios y entidades, para la vista del usuario. Se evidencia en la clase LayersController, la cual utiliza la información de la clase LayerModel para crear objetos de ella.

Controlador (Controller): se basa en asignar la responsabilidad de todos los eventos realizados a una clase específica que constituye el único punto de entrada para cada evento. Se evidencia en la clase DesignerController, la cual controla el flujo de eventos del módulo.

Bajo Acoplamiento (Low Coupling): el acoplamiento mide el grado en que una clase está conectada, tiene conocimiento o de alguna manera depende de otra. Este patrón consiste en asignar la responsabilidad de manera que el acoplamiento permanezca bajo. El bajo acoplamiento permite crear clases más independientes, más reutilizables, lo que implica mayor productividad. Se evidencia en todo el diseño del sistema ya que cada clase solo se relaciona con otras que necesite, para evitar que al realizar un cambio en alguna de ellas se afecten lo menos posible las otras clases.

Alta Cohesión (High Cohesion): 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. Se evidencia en las clases Layer y LayerModel ya que cada

clase contiene la información y los métodos correspondientes específicos de ellas, logrando que no se sobrecarguen con información de otras clases.

2.7 Patrones GOF

Patrones publicados por Gamma, Helm, Johnson y Vlossodes en 1995: patrones de la banda de los cuatro (del inglés, Gang of Four). Esta serie de patrones permiten ampliar el lenguaje, aprender nuevos estilos de diseño y además introducir más notación UML. Los patrones de diseño del grupo GoF se clasifican en tres grandes categorías basadas en su propósito: creacionales, estructurales y de comportamiento (Demestre y González 2019).

- **Patrones creacionales:** Son los que facilitan la tarea de creación de nuevos objetos, de tal forma que el proceso de creación pueda ser desacoplado de la implementación del resto del sistema. En este grupo están los patrones que sirven de guía para construir objetos. Además, independizan el ¿Qué?, ¿Quién?, ¿Cómo? y ¿Cuándo? en la creación del objeto (Astorga et al. 2019).
 - **Prototipo:** El patrón de prototipo se usa cuando la creación de objetos es costosa, requiere mucho tiempo y recursos y ya poseemos un objeto similar. Este patrón proporciona un mecanismo para copiar el objeto original a un nuevo objeto y luego modificarlo de acuerdo a nuestras necesidades. Es utilizado por la clase DesignerController al disponer del Árbol de Capas como un prototipo para la elaboración de la leyenda y permitir su posterior modificación en dependencia de las capas habilitadas.
 - **Singleton:** El patrón de diseño Singleton (instancia única) está diseñado para restringir la creación de objetos pertenecientes a una clase o el valor de un tipo a un único objeto. Se evidencia en la clase DesignerController, la cual define varias variables globales de instancia única que permiten la comunicación con el modelo y demás clases.
- **Patrones estructurales:** Un patrón estructural de clases utiliza la herencia para componer interfaces o implementaciones. Un patrón estructural de objetos describe la forma en que se componen objetos para obtener nueva funcionalidad, además se añade la flexibilidad de cambiar la composición en tiempo de ejecución. En este grupo

encontramos los patrones que organizan las clases y objetos de características similares y así formar estructuras más grandes (Astorga et al. 2019).

- **Proxy:** Es una clase que funciona como interfaz hacia cualquier otra cosa: una conexión a Internet, un archivo en disco o cualquier otro recurso que sea costoso o imposible de duplicar. Este patrón es empleado por la clase DesignerController.
 - **Decorator:** Permite añadir funcionalidad extra a un objeto (de forma dinámica o estática) sin modificar el comportamiento del resto de objetos del mismo tipo. Se ve reflejado en las clases cliente de la vista al heredar el código que es común en todas las páginas del sistema, para no tener que repetirlo en cada una de estas.
- **Patrones de comportamiento:** Buscan proporcionar al usuario del código objetos que aporten una funcionalidad determinada. De este modo los objetos presentan una firma cuyos métodos resuelven un problema específico sin exponer su funcionamiento (Astorga et al. 2019).
- **Observer:** Los objetos son capaces de suscribirse a una serie de eventos que otro objetivo va a emitir, y serán avisados cuando esto ocurra. Este patrón se utiliza en la clase LayersController para controlar las llamadas sincrónicas a la base de datos; así como en las clases del paquete vista para realizar las acciones deseadas luego de concluir una interacción del mapa.

2.8 Modelo de datos

Un modelo de base de datos muestra la estructura lógica de la base, incluidas las relaciones y limitaciones que determinan cómo se almacenan los datos y cómo se accede a ellos. Los modelos de bases de datos individuales se diseñan en base a las reglas y los conceptos de cualquier modelo de datos más amplio que los diseñadores adopten. La mayoría de los modelos de datos se pueden representar por medio de un diagrama de base de datos acompañante. A continuación, se muestra una representación de la base de datos utilizada:

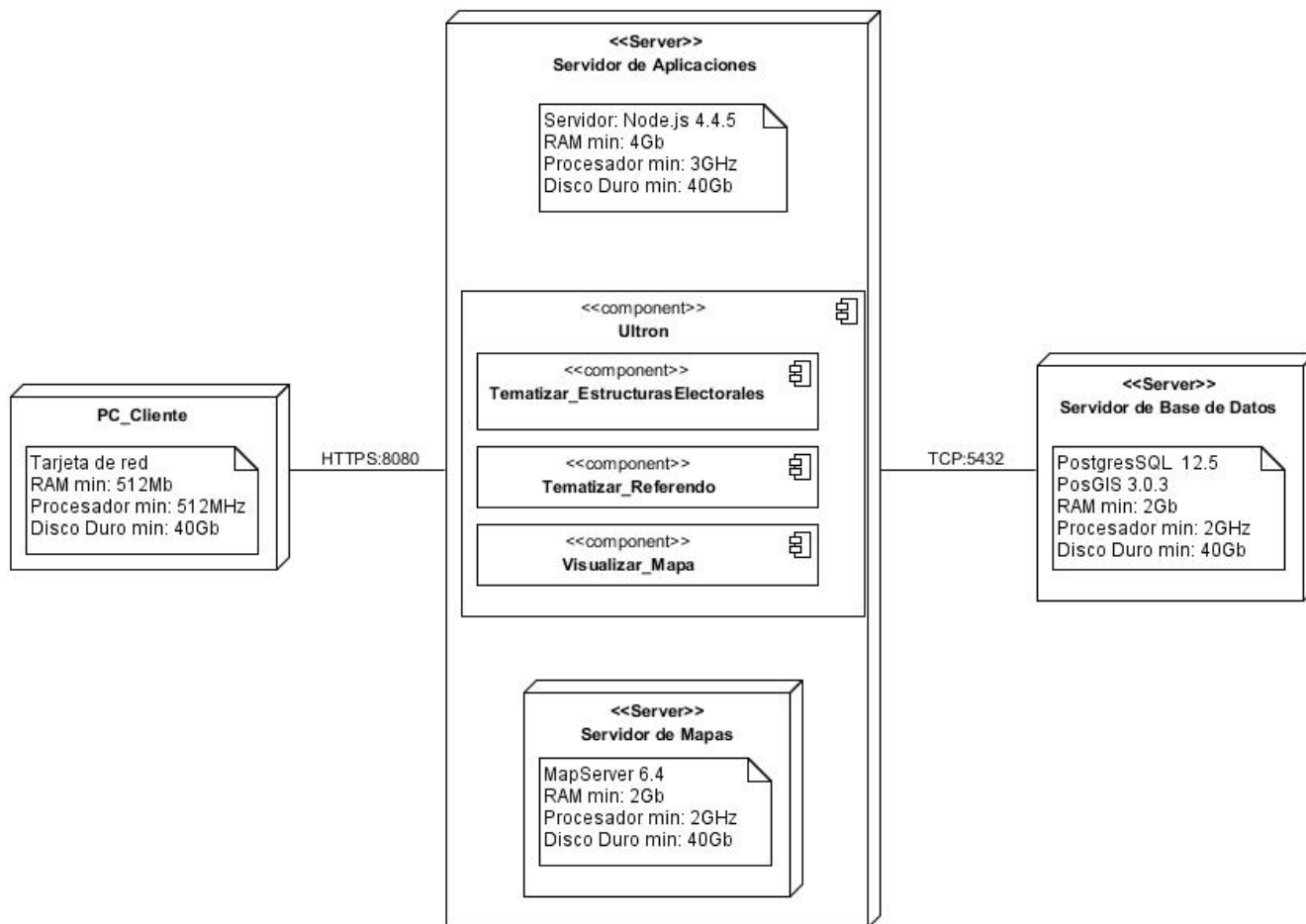


Figura 7: Diagrama de despliegue.

Conclusiones del capítulo

Con la culminación de este capítulo se logró la identificación de los requisitos funcionales y no funcionales del sistema dando paso a una mejor comprensión del mismo. Fueron definidas las historias de usuario generadas a partir de la metodología seleccionada. La definición de la arquitectura y los patrones de diseño a utilizar, permitiendo establecer la base para fomentar la reutilización y las buenas prácticas de programación, así como disminuir el impacto de los cambios futuros en el código fuente.

CAPÍTULO III: IMPLEMENTACIÓN Y VALIDACIÓN DEL SISTEMA DE INFORMACIÓN GEOGRÁFICA PARA DECIDIMOS

En el presente capítulo se presentan los componentes y estándares de codificación que sustentan la implementación del Sistema de Información Geográfica para **decidimOS**. Se describen y fundamenta el proceso de validación de la solución implementada, mediante la utilización de casos de uso de pruebas.

3.1 Diagrama de componentes

Los diagramas de componentes muestran los elementos de diseño de un sistema de información. Permiten visualizar con más facilidad la estructura general del sistema y el comportamiento del servicio que estos componentes proporcionan y utilizan a través de las interfaces. Un diagrama de componentes tiene un nivel más alto de abstracción que un diagrama de clase, usualmente un componente se implementa por una o más clases en tiempo de ejecución (Leon 2017).

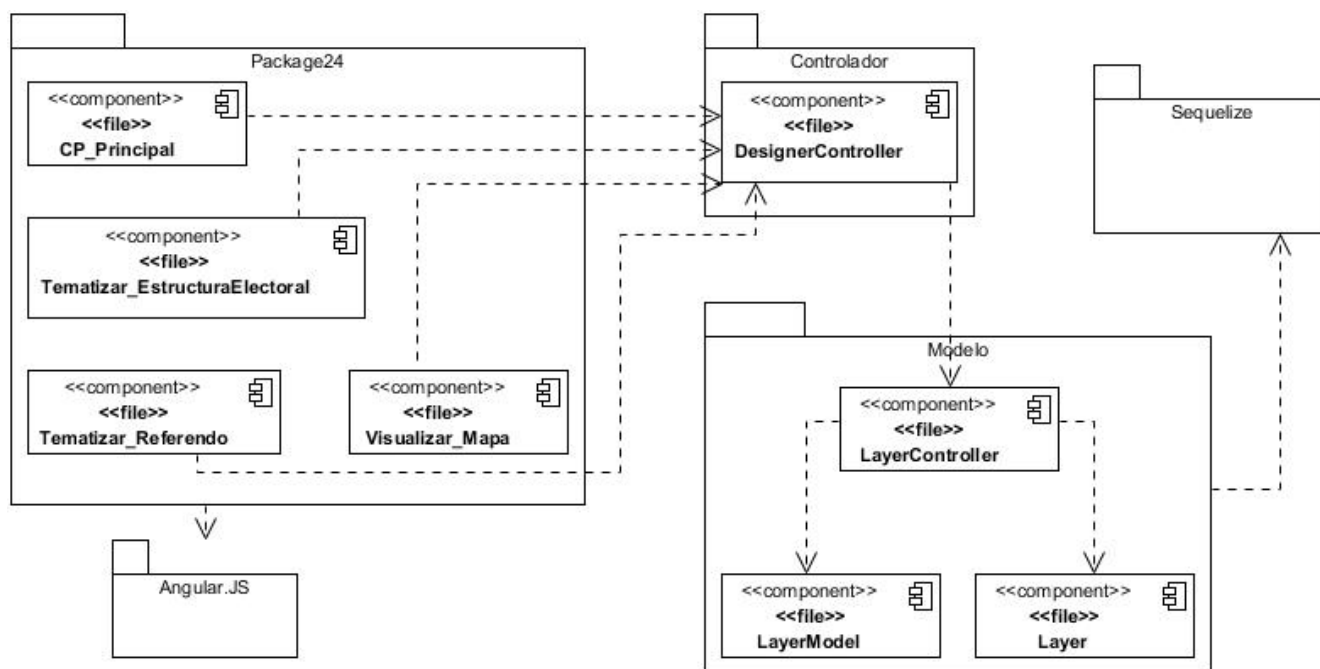


Figura 8: Diagrama de componentes.

3.2 Estándares de codificación

Entendemos como estándar de código a un conjunto de convenciones establecidas de ante mano (denominaciones, formatos, etc.) para la escritura de código. Estos estándares varían dependiendo del lenguaje de programación elegido y además varían en cobertura, algunos son más extensos que otros (Vera Pasabare 2019). Para la implementación del módulo se hacen uso de los siguientes estándares de codificación definidos por Jairo Rojas Delgado para el desarrollo de la plataforma ULTRON:

- Añadir comentarios para explicar determinadas partes de tu código.
- Al elegir nombres de las variables, se recomienda el utilizar variables descriptivas.
- Las funciones tienen que seguir el siguiente esquema de nombrado: `modulo_sub-modulo_nombre_funcion ()`.
- Las funciones se escriben en minúsculas y separados con guiones de suelo, y los argumentos que reciben son descriptivos.
- Todo el código desarrollado tendrá una indentación de 4 espacios.
- La indentación se realizará siempre con tabuladores y no con cuatro puntos cuidando de no mezclar un estilo con el otro.
- Se dejará una línea en blanco inmediatamente después del inicio de un bloque de código, dígame bloques de funciones, bloques condicionales o bucles.
- La llave de apertura de un bloque de código se coloca después de la instrucción anterior y no en una línea en blanco independiente.
- La llave de clausura de un bloque de código tendrá el mismo nivel de indentación que la línea de la llave de apertura.
- Cada bloque de código anidado tendrá un nivel de indentación más que el bloque padre exceptuando las llaves de apertura y cierre.
- Todas las variables de un bloque de código se declararán al inicio del método en cuestión.
- Las variables se declaran una por línea y empleando la palabra reservada `<var>` que nunca podrá omitirse, aunque el lenguaje lo permita.

3.3 Estrategia de pruebas

Una estrategia de prueba de software recoge las técnicas de diseño de casos de pruebas en una serie de pasos que llevan a la construcción correcta del software. Es una parte fundamental del proceso de validación y verificación del software. La verificación no es más que la actividad en la cual nos aseguramos que las distintas partes del software cumplen con las funciones para las cuales fueron diseñadas, mientras que la validación se encarga de comprobar los módulos verificados cumplen con los requisitos expresados por el cliente (Costa y Teixeira 2018).

3.3.1 Niveles de pruebas

Para la realización de las pruebas se definieron los siguientes niveles de pruebas que a continuación se describen:

- **Pruebas unitarias:** Las pruebas unitarias son pruebas automatizadas que verifican la funcionalidad en el componente, clase, método o nivel de propiedad. El objetivo principal de estas es tomar la pieza más pequeña de software comprobable en la aplicación, aislarla del resto del código y determinar si se comporta exactamente como esperamos. Cada unidad se prueba por separado antes de integrarlas en los componentes para probar las interfaces entre las unidades (Osherove, Feathers, y Martin 2020).
- **Pruebas de integración:** El objetivo de las pruebas de integración es verificar el correcto ensamblaje entre los distintos componentes una vez que han sido probados unitariamente con el fin de comprobar que interactúan correctamente a través de sus interfaces, tanto internas como externas, cubren la funcionalidad establecida y se ajustan a los requisitos no funcionales especificados en las verificaciones correspondientes (Luppen, Kletzing, y Kirchner 2018).
- **Pruebas de sistema:** Las pruebas de sistema son un nivel de prueba que valida el producto de software completo y totalmente integrado. El propósito de las pruebas del sistema es evaluar las especificaciones del sistema de un extremo a otro. El software suele ser solo una característica de un sistema informático más grande. En última instancia, el software está interconectado con otros sistemas de software / hardware.

La prueba del sistema es en realidad una serie de pruebas diferentes cuyo único propósito es operar todo el sistema basado en computadora (Palomo y Gil 2020).

3.3.2 Tipos de pruebas

- **Pruebas de funcionalidad:** El servicio de pruebas funcionales se centra en comprobar que los sistemas desarrollados funcionan acorde a las especificaciones funcionales y requisitos del cliente. Este servicio ayuda a su organización a detectar los posibles defectos derivados de errores en la fase de programación. Las características no funcionales del software se pueden medir de diversas maneras, por ejemplo, por medio de tiempos de respuesta en el caso de pruebas de rendimiento o por número máximo de sesiones en pruebas de estrés (Excelencia 2021)
- **Pruebas no funcionales:** Se enfocan en validar un sistema o aplicación por medio de sus requerimientos no funcionales, la forma en que el sistema funciona y no por medio de comportamientos específicos. Estas permiten conocer qué riesgos corre el producto y nos dicen si tiene un mal desempeño o un bajo rendimiento en los entornos de producción. Permiten explicar lo que soporta el producto y si cumple con las expectativas de los clientes (Morales 2018).
 - **Pruebas de carga:** Consiste en someter al sistema a altas cargas de trabajo, simulando la actividad real de los futuros usuarios del sistema. Estas pruebas nos ayudarán a predecir el comportamiento de nuestro sistema y conocer el grado en que realiza las funciones para las que ha sido diseñado, sin pérdidas de servicio y con un tiempo de respuesta óptimo y estable (Excelencia 2021).

3.3.3 Métodos de prueba

Método de Caja Negra: Estas pruebas son realizadas desde la interfaz gráfica, es un método de pruebas de software en el cual la funcionalidad se verifica sin tomar en cuenta la estructura interna de código, detalles de implementación o escenarios de ejecución internos en el software. Se enfoca solamente en las entradas y salidas del sistema, apoyándose en la especificación de requisitos y documentación funcional (Aliaga Meléndez 2021).

3.3.4 Técnicas de prueba

- Partición de equivalencia:** Esta técnica consiste en clasificar las entradas de datos del sistema en grupos que presentan un comportamiento similar. Las particiones también pueden definirse en función de las salidas de datos, valores internos, valores relacionados antes o después de ciertos eventos, y también para los valores que reciben las interfaces. Además, se definen pruebas para cubrir todos o parte de las particiones de datos válidos y datos inválidos. Es aplicable a entradas de datos realizadas por personas o vía interfaces con otros sistemas (Excelencia 2021).

3.4 Aplicación de las pruebas

Pruebas funcionales: Para realizar las pruebas se diseñaron casos de pruebas como el que se muestra a continuación:

Caso de prueba: Mostrar tematización de colores de la cantidad de circunscripciones ordinarias por provincias.

Descripción general: Permitirá al usuario realizar un análisis sobre el mapa a nivel de provincia de la cantidad de circunscripciones ordinarias.

Condiciones de ejecución: El usuario debe estar autenticado.

SC 1: Gráficas dinámicas.

Tabla 5: Caso de prueba mostrar tematización de colores de la cantidad de circunscripciones ordinarias por provincias.

| Escenario | Descripción | Categoría | Criterio de análisis | Respuesta del sistema | Flujo central |
|---|---|------------|----------------------|--|---|
| EC 1.1 Tematiza teniendo en cuenta el análisis de la cantidad de circunscripciones ordinarias | Se muestra un mapa tematizado por colores según la cantidad de circunscripciones ordinarias | V | V | Se muestra un mapa tematizado con un degradado de colores con los valores de la cantidad de circunscripciones ordinarias | 1. El usuario selecciona la opción: "Mapa temático", presente en la barra de herramientas o en la barra |
| | | Plebiscito | Criterio | | |

| | | | | | |
|---|---|---|---|---|--|
| | | | | | <p>de menú en el módulo Análisis/Mapa temático.</p> <p>2. Selecciona la categoría "X" presente en el panel lateral izquierdo.</p> <p>3. Selecciona el criterio de análisis "Z".</p> <p>4. Presiona el botón "Aceptar"(Solo se habilita cuando se han seleccionado los valores en los campos Categoría y Criterio de análisis)</p> |
| EC 1.2 Dejar campos obligatorios en blanco. | El usuario deja campos obligatorios en blanco en el | V | I | El sistema señala de color rojo el campo obligatorio en | |

| | | | | | |
|---|---|------------|-----|---|--|
| | formulario, se muestra inactivo el botón "Aceptar". | Plebiscito | | blanco con un logo que al poner el mouse por encima de él muestra un mensaje informando del error y mantiene el botón "Aceptar" inactivo. Para campos obligatorios vacíos el sistema muestra el mensaje de error: "Este campo es obligatorio.". | |
| EC 1.3 Cierra la ventana "Análisis temático". | El usuario cierra la ventana "Análisis temático". | N/A | N/A | Se cierra la ventana "Análisis Temático". | |
| EC 1.4 Cancela la tematización. | Presiona la opción "Cancelar". Cierra la ventana "Análisis temático". | N/A | N/A | Se cierra la ventana "Análisis Temático". | |

Caso de prueba: Mostrar tematización de colores del por ciento de electores que votaron según parte básico por provincias.

Descripción general: Permitirá al usuario realizar un análisis sobre el mapa a nivel de provincia del por ciento de electores que votaron según parte básico por provincias.

Condiciones de ejecución: El usuario debe estar autenticado.

SC 1: Gráficas dinámicas.

Tabla 6: Caso de prueba mostrar tematización de colores del por ciento de electores que votaron según parte básico por provincias.

| Escenario | Descripción | Categoría | Criterio de análisis | Respuesta del sistema | Flujo central |
|---|--|-----------------|----------------------|---|--|
| EC 1.1 Tematiza teniendo en cuenta el análisis del por ciento de electores que votaron según parte básico por provincias | Se muestra un mapa tematizado por colores según el por ciento de electores que votaron según parte básico por provincias | V Plebiscito | V Criterio | En la parte inferior del mapa muestra una leyenda donde los colores representan los criterios representados en el mapa que se corresponden con los rangos de por ciento definidos para el análisis de electores que votaron según parte básico por provincias | 1. El usuario selecciona la opción: "Mapa temático", presente en la barra de herramientas o en la barra de menú en el módulo Análisis/Mapa temático. 2. Selecciona la categoría "X" presente en el panel lateral izquierdo. 3. Selecciona el |

| | | | | | |
|--|--|------------|---|--|---|
| | | | | | <p>criterio de análisis "Z".</p> <p>4. Presiona el botón "Aceptar"(Solo se habilita cuando se han seleccionado los valores en los campos Categoría y Criterio de análisis)</p> |
| <p>EC 1.2 Dejar campos obligatorios en blanco.</p> | <p>El usuario deja campos obligatorios en blanco en el formulario, se muestra inactivo el botón "Aceptar".</p> | V | I | <p>El sistema señala de color rojo el campo obligatorio en blanco con un logo que al poner el mouse por encima de él muestra un mensaje informando del error y mantiene el botón</p> | |
| | | Plebiscito | | | |

| | | | | | |
|---|--|-----|-----|--|--|
| | | | | <p>“Aceptar” inactivo. Para campos obligatorios vacíos el sistema muestra el mensaje de error: "Este campo es obligatorio."</p> | |
| EC 1.3 Cierra la ventana “Análisis temático”. | El usuario cierra la ventana “Análisis temático”. | N/A | N/A | Se cierra la ventana “Análisis Temático”. | |
| EC 1.4 Cancela la tematización. | Presiona la opción "Cancelar". Cierra la ventana “Análisis temático”. | N/A | N/A | Se cierra la ventana “Análisis Temático”. | |

Pruebas de carga: Para la realización de las pruebas de carga se utilizará la herramienta Apache JMeter en su versión 2.3.4 con el fin de comprobar el rendimiento del sistema. Para esta prueba se hará una simulación de 100 usuarios concurrente haciendo peticiones al servidor.

Entorno en que se realizara la prueba:

Hardware de prueba (PC Servidor):

- Sistema Operativo: Windows 10 Pro
- Microprocesador: Intel(R) Core(TM) i3-5010U
- Memoria RAM: 6.00 Gb

Pruebas unitarias: Para llevar a cabo las pruebas unitarias se utilizó la herramienta QUnit la cual basa su funcionamiento en una clase test que posibilita probar el código de aplicaciones web desarrolladas en JavaScript.

Conclusiones del capítulo

En el presente capítulo fueron abordados los temas referentes a la implementación de la propuesta de solución y la estrategia de pruebas a seguir durante la elaboración del sistema. La confección del diagrama de componentes, permitió comprender la interacción entre los componentes que conforman el sistema y mostrar las dependencias que existen entre los mismos. Fueron definidos los estándares de codificación a emplear para la implementación de la aplicación.

CONCLUSIONES FINALES

Con la realización de la presente investigación se le dio cumplimiento al objetivo general, así como a las tareas de la investigación, arribando a las siguientes conclusiones:

- El estudio de los conceptos asociados a los SIG, permitió la comprensión de los principales términos que se manejaron durante la investigación.
- El análisis de los sistemas homólogos permitió identificar las principales funcionalidades a tener en cuenta en el desarrollo de la solución propuesta.
- La identificación de los requisitos funcionales y no funcionales, descritos a través de las historias de usuario, facilitaron su comprensión y la posterior codificación de la solución.
- Se logró la planificación de las pruebas funcionales y no funcionales para la validación de la solución.

RECOMENDACIONES

Independientemente de que se hallan alcanzado los objetivos alcanzados al inicio de la investigación se recomienda:

- Adicionar nuevos indicadores que pueda necesitar el cliente a fin de satisfacer sus necesidades.
- Desarrollar nuevo componente para los demás procesos electorales del país, con el objetivo de apoyar el proceso de toma de decisiones.

REFERENCIAS BIBLIOGRÁFICAS

1. Aliaga Meléndez, Javier Nicolás. 2021. «Plataforma web para elaborar y aplicar rúbricas de evaluación altamente configurables». PhD Thesis, Universidad de Talca (Chile). Escuela de Ingeniería Civil en Computación.
2. «Apache JMeter - Apache JMeter™». s. f. Accedido 9 de junio de 2022. <https://jmeter.apache.org/>.
3. artanaisi. 2019. «Gaceta Oficial No. 60 Ordinaria de 2019». Text. Gaceta Oficial. 19 de septiembre de 2019. <https://www.gacetaoficial.gob.cu/es/gaceta-oficial-no-60-ordinaria-de-2019>.
4. Astorga, José Alejandro Ibarra, Jorge Luis Osuna Tirado, Ricardo Ulises Reyes Ramírez, José Carlos Lepe Mendoza, y Álvaro Peraza Garzón. 2019. «CLASIFICACIÓN DE LOS PATRONES DE DISEÑO IDÓNEOS EN PROGRAMACIÓN ANDROID». *Revista Digital de Tecnologías Informáticas y Sistemas* 3 (3).
5. Ávila del Campo, Arturo. 2014. «Diseño de un SIG para la ubicación óptima de una instalación minera».
6. Ávila Garzón, Cecilia. 2019. «Modelo Vista Controlador». *Fundación Universitaria Konrad Lorenz*, diciembre. <https://repositorio.konradlorenz.edu.co/handle/001/1528>.
7. Batra, Dinesh. 2020. «Job-Work Fit as a Determinant of the Acceptance of Large-Scale Agile Methodology». *Journal of Systems and Software* 168 (octubre): 110577. <https://doi.org/10.1016/j.jss.2020.110577>.
8. Bermúdez Martínez, Jhonnyer Fernando. 2016. «Sistema de información geográfica de población electoral y control de solicitudes». <https://ridum.umanizales.edu.co/xmlui/handle/20.500.12746/2653>.
9. Blas, María Julia, Horacio Pascual Leone, y Silvio Miguel Gonnet. 2019. «Modelado y verificación de patrones de diseño de arquitectura de software para entornos de computación en la nube», diciembre. <https://doi.org/10.17013/risti.35.1-17>.
10. Bucheli, Grace Estefanía Hidalgo. 2019. «Uso del Índice de Moran y LISA para explicar el ausentismo electoral rural en Ecuador». *Revista Geográfica*, n.º 160: 91-108. <https://doi.org/10.35424/regeo.160.2019.746>.

11. Carrillo Peña, Andrés Leonardo. 2020. «Prácticas académicas en Electrosoftware S.A.S». <https://repository.unab.edu.co/handle/20.500.12749/14272>.
12. Cascón-Katchadourian, Jesús-Daniel. 2020. «Tecnologías para luchar contra la pandemia Covid-19: geolocalización, rastreo, big data, SIG, inteligencia artificial y privacidad//Technologies to fight the Covid-19 pandemic: geolocation, tracking, big data, GIS, artificial intelligence, and privacy». *Profesional de la información* 29 (4).
13. Chiang, Yao-Yi, Weiwei Duan, Stefan Leyk, Johannes H. Uhl, y Craig A. Knoblock. 2020. *Using historical maps in scientific studies: Applications, challenges, and best practices*. Springer.
14. Costa, Alex, y Leopoldo Teixeira. 2018. «Testing Strategies for Smart Cities applications: A Systematic Mapping Study». En *Proceedings of the III Brazilian Symposium on Systematic and Automated Software Testing*, 20-28. SAST '18. New York, NY, USA: Association for Computing Machinery. <https://doi.org/10.1145/3266003.3266005>.
15. Demestre, Dayana Hardy, y Yulina Hernández González. 2019. «Design of a Human Resources Information Management System».
16. «Documentation for Visual Studio Code». s. f. Accedido 9 de junio de 2022. <https://code.visualstudio.com/docs>.
17. Excelencia, Panel Testing-Centro de. 2021. «Software QA - ¿Cuáles son los tipos de pruebas software?» *Panel Sistemas* (blog). 10 de enero de 2021. <https://www.panel.es/software-qa-cuales-son-los-tipos-de-pruebas-software/>.
18. «Express - Node.js Web Application Framework». s. f. Accedido 9 de junio de 2022. <https://expressjs.com/>.
19. «Gaceta Oficial de la República de Cuba publicó el Código de las Familias». s. f. Asamblea Nacional del Poder Popular. Accedido 6 de noviembre de 2022. <https://www.parlamentocubano.gob.cu/noticias/gaceta-oficial-de-la-republica-de-cuba-publico-el-codigo-de-las-familias>.
20. González, Laura Mercedes Camacho, Reynaldo Alvarez Luna, Lester Collado Rolo, Lisaidi Lucia Coca, Lisdan Rodríguez Pérez, Oswald Aguilar Dowins, y Yaidel Castro Díaz. 2021. «Ecosistema de software para la gestión de procesos electorales», 10.
21. Guerra Albarracín, Rafael. 2018. «Selección de emplazamiento óptimo de una base militar mediante sistemas de información geográfica».

22. He, Yuqin, Zhenbo Bi, Haobin Tian, Kai Duan, Jiashuai Wu, y Hongwei Wang. 2019. «Application of OpenLayers in Marine Information Monitoring». *E3S Web of Conferences* 118: 03006. <https://doi.org/10.1051/e3sconf/201911803006>.
23. Hernández, Nancy Blanco, Luis G. Montané Jiménez, y Carmen Mezura Godoy. 2018. «Sistemas groupware para el diseño de diagrama de clases uml en ambientes táctiles». *Pistas Educativas* 39 (127).
24. Herrera, Roger Amílcar González, Beth sua Iztaccihuatl Albornoz Euán, Ismael Abelardo Sánchez y Pinto, y José Humberto Osorio Rodríguez. 2018. «EL ACUÍFERO YUCA-TECO. ANÁLISIS DEL RIESGO DE CONTAMINACIÓN CON APOYO DE UN SISTEMA DE INFORMACIÓN GEOGRÁFICA». *Revista Internacional de Contaminación Ambiental* 34 (4): 667-83. <https://doi.org/10.20937/RICA.2018.34.04.09>.
25. Herron, David. 2018. «Node.js Web Development - Fourth Edition [Book]». mayo de 2018. <https://www.oreilly.com/library/view/nodejs-web-development/9781788626859/>.
26. Hsu, Leo S., y Regina Obe. 2021. *PostGIS in action*. Simon and Schuster.
27. Infante Gómez, Pablo. 2021. «Análisis multicriterio para la localización, en Andalucía, de una central hidroeléctrica a través de un sistema de información geográfica».
28. Jacobson, Lvar, y James Rumbaugh Grady Booch. 2021. «The unified modeling language reference manual».
29. James, Toby S, Holly Ann Garnett, Leontine Loeber, y Carolien van Ham. 2019. «Electoral Management and the Organisational Determinants of Electoral Integrity: Introduction». *International Political Science Review* 40 (3): 295-312. <https://doi.org/10.1177/0192512119828206>.
30. Ledea, Lilianne Martínez, Adrián Gracia Aguila, y Lisandra Escalona Griff. 2013. «Sistema de Información Geográfica para el proceso electoral de la República de Cuba», agosto.
31. Leon, Johnny. 2017. «Diagramas de Componentes UML: Diagrama de Componentes». *Diagramas de Componentes UML* (blog). 29 de junio de 2017. <http://diagramasdecomponentes.blogspot.com/2013/06/diagrama-de-componentes.html>.
32. «Ley No. 127, "Ley Electoral", de 13 de julio de 2019». 2019. Text. Gaceta Oficial. 4 de febrero de 2019. <https://www.gacetaoficial.gob.cu/es/ley-no-127-ley-electoral-de-13-de-julio-de-2019>.

33. Liza Contreras, Romina Alejandra. 2021. «Creación de bases de datos geográficos en la identificación de la intervención de los bosques en Perú».
34. Luppen, Zachary, Craig Kletzing, y Donald Kirchner. 2018. «Integration and Automation Software for Testing of RIME and REASON Instruments». *Integration*.
35. Martínez, F. J. 2018. «Introducción a los Sistemas de Información Geográfica». *Obtenido de <https://sge.org/ibercarto/wp-content/uploads/sites/4/2016/01/sig2.pdf>*.
36. McKenna, Jeff, y MapServer PSC. 2021. «MapServer». Zenodo. <https://doi.org/10.5281/ZENODO.5842012>.
37. Miguel, SANTOS PRECIADO José. 2020. *SISTEMAS DE INFORMACIÓN GEOGRÁFICA*. Editorial UNED.
38. Morales, Víctor Manuel Soto. 2018. «Conoce qué son las pruebas no funcionales de software». 2018. <https://www.pragma.com.co/blog/conoce-que-son-las-pruebas-no-funcionales-de-software>.
39. Muñoz Muñoz, Cristian Antonio. 2020. «Aplicación de la metodología mobile-d en el desarrollo de una app móvil para gestionar citas médicas del centro Jel Riobamba». B.S. thesis, Riobamba: Universidad Nacional de Chimborazo.
40. Ordóñez, Mariuxi Paola Zea, Jimmy Rolando Molina Ríos, y Fausto Fabían Redrován Castillo. 2017. *Administración de Bases de datos con PostgreSQL*. Vol. 19. 3Ciencias.
41. Ortega, Gilberto Andrés Vargas. 2021. «Lineamientos para el diseño de aplicaciones web soportados en patrones GRASP». *Ciencia e Ingeniería* 8 (2): e5716304-e5716304.
42. Osherove, Roy, Michael Feathers, y Robert Martin. 2020. *The Art of Unit Testing: Deutsche Ausgabe*. BoD—Books on Demand.
43. Palomo, Sebastián Rubén Gómez, y Eduardo Moraleda Gil. 2020. *Aproximación a la ingeniería del software*. Editorial Centro de Estudios Ramon Areces SA.
44. Pisco Gomez, Angel Pisco, Julio Johnny Regalado Jalca, Jimmy Gutierrez Garcia, Omar Quimis Sanchez, Kleber Marcillo Parrales, y Javier Marcillo Merino. 2020. «Fundamentos sobre la Gestión de Bases de Datos».
45. Rahwanto, Henderi Dr Untung Rahardja Efana. 2022. *UML POWERED DESIGN SYSTEM USING VISUAL PARADIGM*. CV Literasi Nusantara Abadi.
46. Rebah, Hassen Ben, Hafedh Boukthir, y Antoine Chedebois. 2022. *Website Design and Development with HTML5 and CSS3*. John Wiley & Sons.

47. Ríos, Roberto Carlos Pauta, David Mayorga Arias, y Elvis Rafael Castro Macías. 2019. «Uso de sistemas de información geográfica SIG para la elaboración de planos de fincas agrícolas». *Opuntia Brava* 11 (1): 217-23.
48. San José, Jorge del Río, Junta de Castilla, León Delegación Territorial de Valladolid Servicio, y Territorial de Medio Ambiente. 2018. *La vía ecléctica de producción y consumo de datos espaciales*. Universidad de León.
49. Sánchez, Tamara Rodríguez. 2015. «Metodología de desarrollo para la Actividad productiva de la UCI». *La Habana*.
50. Sendra, Joaquín Bosque, y Gustavo D. Buzai. 2017. «Geografía Electoral de la Ciudad Autónoma de Buenos Aires, 2015. Elecciones a Jefe de Gobierno municipal y a Presidente de la República Argentina». *Persona y Sociedad* 31 (1): 48-73. <https://doi.org/10.53689/pys.v31i1.114>.
51. Silva-Balaguera, Andrés, Omar Daza-Leguizamón, y Lesly Lopez-Valiente. 2018. «Gestión de pavimentos basado en sistemas de información geográfica (SIG): una revisión». *Revista Ingeniería Solidaria* 14 (26).
52. Slocum, Terry A., Robert B. McMaster, Fritz C. Kessler, y Hugh H. Howard. 2022. *Thematic cartography and geovisualization*. CRC Press.
53. Sommerville, Ian. 2016. «„Software Engineering 10 “». *Harlow: Pearson Education Limited*.
54. Teja, MSc Luis Jacinto López de la, Dra C. Ángela Sarría Stuart, y Dr C. Denis Fernández Álvarez. 2017. «La formación de conceptos en el proceso de enseñanza-aprendizaje de los sistemas de gestión de bases de datos». *Revista Conrado* 13 (57): 139-45.
55. Theisen, Kevin J. 2019. «Programming Languages in Chemistry: A Review of HTML5/JavaScript». *Journal of Cheminformatics* 11 (1): 11. <https://doi.org/10.1186/s13321-019-0331-1>.
56. Vera Pasabare, Johan. 2019. «LAS MEJORES PRACTICAS EN CODIFICACION DE SOFTWARE». PhD Thesis.
57. «What Is Express.js? | Why Should Use Express.js? | Features of Express.js». 2020. *Besant Technologies* (blog). 19 de junio de 2020. <https://www.besanttechnologies.com/what-is-expressjs>.

ANEXOS

Anexo 1: Historia de Usuario.

Tabla7: Mostrar tematización de colores de la cantidad de electores por provincias.


| | |
|---|---|
| | |
| Número: 15 | Requisito: Mostrar tematización de colores de la cantidad de electores por provincias. |
| Programador: Frank Alberto Plasencia Téllez | Iteración Asignada: 1 |
| Prioridad: Alta | Tiempo Estimado: 3 |
| Riesgo en Desarrollo: <ul style="list-style-type: none"> • Planificación irreal. • Interrupción del servicio eléctrico. • Afectaciones externas al personal de trabajo. | Tiempo Real: 3 |
| Descripción: En la interfaz se debe mostrar una tematización en el mapa que de acuerdo a la cantidad de electores por provincia este cambie de color. | |
| <p>Observaciones: Pueden realizar la funcionalidad los roles con el permiso correspondiente. El usuario tiene que estar autenticado.</p> <p>Selección de tematización: Opción que se representa a través del ícono (). Permite seleccionar la funcionalidad de tematizar en el mapa. Para su uso:</p> <ol style="list-style-type: none"> 1. Seleccionar la funcionalidad de tematización. 2. Seleccionar la opción Por cantidad. 3. Escoger un indicador. <p>Respuesta del sistema: El mapa se pinta con un color en degradado mostrando los valores por provincia.</p> | |
| Prototipo elemental de interfaz gráfica de usuario: | |



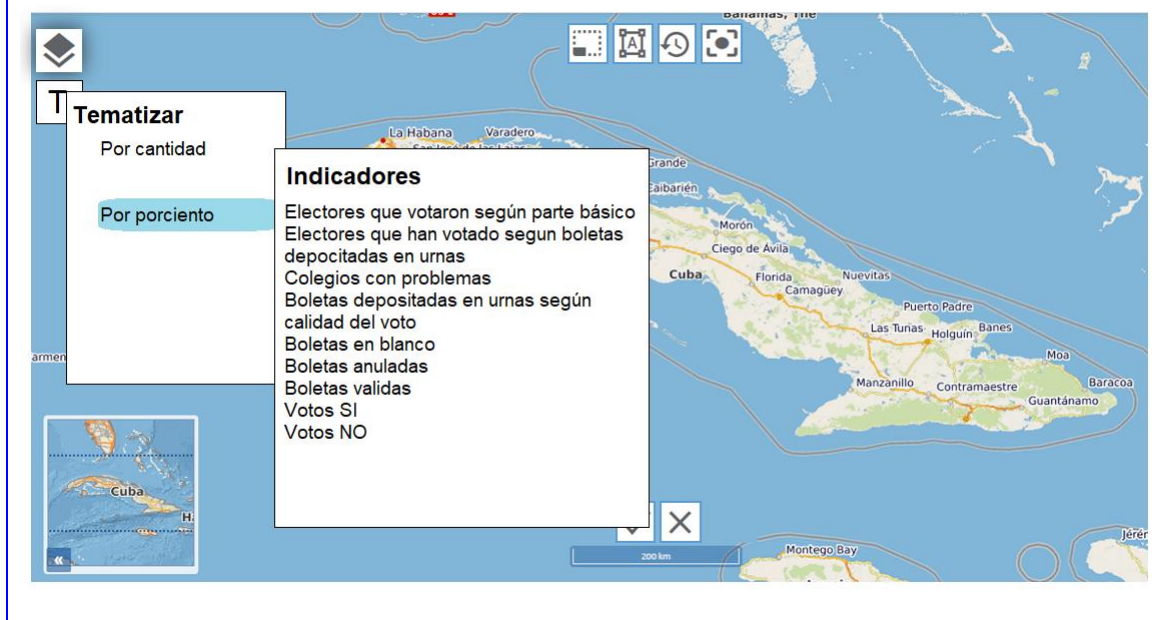
Tabla 8: Mostrar tematización de colores del porciento de electores que han votado según boletas depositadas en urnas por provincias.

| | |
|--|--|
| | |
| Número: 25 | Requisito: Mostrar tematización de colores del porciento de electores que han votado según boletas depositadas en urnas por provincias. |
| Programador: Frank Alberto Plasencia Téllez | Iteración Asignada: 1 |
| Prioridad: Alta | Tiempo Estimado: 3 |
| Riesgo en Desarrollo: <ul style="list-style-type: none"> • Planificación irreal. • Interrupción del servicio eléctrico. • Afectaciones externas al personal de trabajo. | Tiempo Real: 3 |
| Descripción: En la interfaz se debe mostrar una tematización en el mapa que de acuerdo al porciento de electores que han votado según boletas depositadas en urnas por provincia este cambie de color de acuerdo a un rango (- 35% mal, entre 35% y 70% regular y mayor que 70% bien). | |
| Observaciones: Pueden realizar la funcionalidad los roles con el permiso correspondiente. El usuario tiene que estar autenticado. Selección de tematización: Opción que se representa a través del ícono (T). Permite seleccionar la funcionalidad de tematizar en el mapa. Para su uso: <ol style="list-style-type: none"> 1. Seleccionar la funcionalidad de tematización. 2. Seleccionar la opción Por porciento. | |

3. Escoger un indicador.

Respuesta del sistema: El mapa se pinta con un color (Rojo, Amarillo, Verde) mostrando los valores por provincia.

Prototipo elemental de interfaz gráfica de usuario:



Anexo 2: Casos de prueba.

Caso de prueba: Mostrar tematización de colores del total de votos por provincias.

Descripción general: Permitirá al usuario realizar un análisis sobre el mapa a nivel de provincia del total de votos.

Condiciones de ejecución: El usuario debe estar autenticado.

SC 1: Gráficas dinámicas.

Tabla 9: Caso de prueba mostrar tematización de colores del total de votos por provincias.

| Escenario | Descripción | Categoría | Criterio de análisis | Respuesta del sistema | Flujo central |
|--|---|-----------------|----------------------|--|--|
| EC 1.1 Tematiza teniendo en cuenta el análisis del total de votos | Se muestra un mapa tematizado por colores según el total de votos | V Plebiscito | V Criterio | Se muestra un mapa tematizado con un degradado de colores con los valores del total de votos | 1. El usuario selecciona la opción: "Mapa temático", presente en la barra de herramientas o en la barra de menú en el módulo Análisis/Mapa temático. 2. Selecciona la |

| | | | | | |
|---|---|------------|-----|---|---|
| | | | | | <p>categoría "X" presente en el panel lateral izquierdo.</p> <p>3. Selecciona el criterio de análisis "Z".</p> <p>4. Presiona el botón "Aceptar" (Solo se habilita cuando se han seleccionado los valores en los campos Categoría4 y Criterio de análisis)</p> |
| EC 1.2 Dejar campos obligatorios en blanco. | El usuario deja campos obligatorios en blanco en el formulario, se muestra inactivo el botón "Aceptar". | V | I | El sistema señala de color rojo el campo obligatorio en blanco con un logo que al poner el mouse por encima de él muestra un mensaje informando del error y mantiene el botón "Aceptar" inactivo. Para campos obligatorios vacíos el sistema muestra el mensaje de error: "Este campo es obligatorio.". | |
| | | Plebiscito | | | |
| EC 1.3 Cierra la ventana "Análisis temático". | El usuario cierra la ventana | N/A | N/A | Se cierra la ventana "Análisis Temático". | |

| | | | | | | |
|--------|--------------------------|--|-----|-----|---|--|
| | | “Análisis temático”. | | | | |
| EC 1.4 | Cancela la tematización. | Presiona la opción "Cancelar". Cierra la ventana “Análisis temático”. | N/A | N/A | Se cierra la ventana “Análisis Temático”. | |

Caso de prueba: Mostrar tematización de colores del por ciento de votos NO por provincias.

Descripción general: Permitirá al usuario realizar un análisis sobre el mapa a nivel de provincia del por ciento de votos NO.

Condiciones de ejecución: El usuario debe estar autenticado.

SC 1: Gráficas dinámicas.

Tabla 10: Caso de prueba mostrar tematización de colores del por ciento de votos NO por provincias.

| Escenario | Descripción | Categoría | Criterio de análisis | Respuesta del sistema | Flujo central |
|-----------|---|-----------------|----------------------|--|--|
| EC 1.1 | Se muestra un mapa tematizado por colores según el por ciento de votos NO | V Plebiscito | V Criterio | En la parte inferior del mapa muestra una leyenda donde los colores representan los criterios representados en el mapa que se corresponden con los rangos de por ciento definidos para el análisis de votos NO | <ol style="list-style-type: none"> 1. El usuario selecciona la opción: “Mapa temático”, presente en la barra de herramientas o en la barra de menú en el módulo Análisis/Mapa temático. 2. Selecciona la categoría "X" presente en el panel lateral izquierdo. 3. Selecciona el criterio de análisis “Z”. 4. Presiona el botón "Aceptar"(Solo se habilita cuando se han seleccionado los valores en los campos Categoría y |

| | | | | | Criterio de análisis) |
|---|--|---------------------|-----|---|------------------------|
| EC 1.2 Dejar campos obligatorios en blanco. | El usuario deja campos obligatorios en blanco en el formulario, se muestra el botón "Aceptar". | V Plebiscito | I | El sistema señala de color rojo el campo obligatorio en blanco con un logo que al poner el mouse por encima de él muestra un mensaje informando del error y mantiene el botón "Aceptar" inactivo. Para campos obligatorios vacíos el sistema muestra el mensaje de error: "Este campo es obligatorio.". | |
| EC 1.3 Cierra la ventana "Análisis temático". | El usuario cierra la ventana "Análisis temático". | N/A | N/A | Se cierra la ventana "Análisis Temático". | |
| EC 1.4 Cancela la tematización. | Presiona la opción "Cancelar". Cierra la ventana "Análisis temático". | N/A | N/A | Se cierra la ventana "Análisis Temático". | |