ESCUELA POLITÉCNICA NACIONAL

FACULTAD DE INGENIERÍA EN SISTEMAS

DESARROLLO DE UNA APLICACIÓN WEB PARA LA VISUALIZACIÓN Y ANÁLISIS DE DATOS GEOREFERENCIADOS COMO SOPORTE EN LA TOMA DE DECISIONES PARA LA EMPRESA HANDYTEC

TRABAJO DE TITULACIÓN PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIERO EN SISTEMAS INFORMÁTICOS Y DE COMPUTACIÓN

ROBERTO CARLOS PRADO IMBACUAN
e-mail: roberto.prado@epn.edu.ec

Director: Dra. TANIA CALLE
e-mail: tania.calle@epn.edu.ec

Quito, mayo 2019

DECLARACIÓN

Yo Roberto Carlos Prado Imbacuán, declaro bajo juramento que el trabajo aquí descrito es de mi autoría; que no ha sido previamente presentada para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.

A través de la presente declaración cedo mis derechos de propiedad intelectual correspondientes a este trabajo, a la Escuela Politécnica Nacional, según lo establecido por la Ley de Propiedad Intelectual, por su Reglamento y por la normatividad institucional vigente.

Roberto Carlos Prado Imbacuán

CERTIFICACIÓN

Certifico que el presente trabajo fue desarrollado por Roberto Carlos Prado Imbacuán, bajo mi supervisión.

Dra. Tania Calle

DIRECTOR DE PROYECTO

DEDICATORIA

A mis padres, por brindarme su apoyo incondicional y consejo. Ellos me han fomentado los mejores valores en mi crecimiento personal. Por el cariño que me han dado y el sacrificio que han hecho desde el momento en que nací les estaré eternamente agradecido.

A mi hermano, por enseñarme a no rendirme nunca, a que de las malas rachas es donde más se aprende y por estar ahí siempre que lo necesito.

A mis padrinos, por su apoyo moral y económico en todas las etapas de mi educación, representan una de las principales razones para dar todo mi esfuerzo cada día.

A mis profesores y amigos que he conocido a lo largo de mi carrera universitaria, por todos los buenos y malos momentos que pase con ellos. El conocimiento y crecimiento personal que logre junto a ellos para mi es invaluable.

Roberto Prado

AGRADECIMIENTO

A toda mi familia, por todo el apoyo moral y económico que me ha permitido seguir adelante durante todas las etapas de mi vida estudiantil.

A la Dra. Tania Calle por sus consejos y guía en el desarrollo de este proyecto, sus enseñanzas, paciencia y su calidad de persona tanto como profesora y tutora ha sido un factor clave en la culminación exitosa de esta etapa de mi vida.

A la empresa Handytec, por brindarme el apoyo logístico al realizar este proyecto y por confiar siempre en las personas que demuestran pasión y entrega en el crecimiento profesional y personal.

A la Escuela Politécnica Nacional, por brindarme todas las oportunidades para obtener mi título profesional.

Roberto Prado

CONTENIDO

CONTENIDO V

RESUMEN XII

ABSTRACT XIII

1. INTRODUCCIÓN 1

1.1. Descripción de la situación actual en el desarrollo de herramientas de Geo-analítica 1

1.2. Formulación del problema 1

1.3. Alcance 2

1.4. Objetivos 2

1.4.1. Objetivo General 2

1.4.2. Objetivos Específicos 3

1.5. Marco Teórico 3

1.5.1. Geo-analítica 3

1.5.2. Analítica visual 4

1.5.3. Modelos para la información geográfica 5

1.5.4. Mapas web 6

1.5.5. UI/UX para diseñar aplicaciones de mapas 7

1.5.6. Scrum 8

1.5.7. Arquitectura Cliente-Servidor 11

1.5.8. Herramientas de desarrollo 11

1.6. Investigación preliminar 14

1.6.1. Ventajas y desventajas del software web existente para el análisis de datos geo-referenciados 14

1.6.2. Principales herramientas de sistemas de información geográfica SIG para el análisis de datos 15

1.6.3. Tipos de estructura de datos usado para el análisis de datos espaciales 15

1.6.4. Ventajas y desventajas de los modelos de información geográfica al momento de ser usados en aplicaciones de mapas 16

2. Metodologia 17

2.1. Definición del proyecto *17*
2.1.1. Definición de los requerimientos 17
2.1.2. Product Backlog 20
2.1.3. Definición de roles 22
2.1.4. Arquitectura del proyecto 23
2.2. Release Planning *25*
2.3. Ejecución de los Sprints *26*
2.3.1. Sprint 0 26
2.3.2. Sprint 1 28
2.3.3. Sprint 2 32
2.3.4. Sprint 3 38
2.3.5. Sprint 4 43
2.3.6. Sprint 5 46
2.3.7. Sprint 6 49
2.4. Pruebas funcionales *55*
Plan de pruebas 55
Pruebas de carga en el Front-End 70
Pruebas de estrés en el Back-end 74
2.5. Pruebas de usabilidad *77*
3. Resultados y discusión 78
4. Conclusiones 79
5. Referencias bibliográficas 82
6. Anexos 86

FIGURAS

Figura 1 - Visualización de tiendas y clientes por afinidad [6] 3
Figura 2 - Propagación de las enfermedades en dirección Este y Suroeste [8]..... 5
Figura 3 - Casos más comunes de la enfermedad [8]..... 5
Figura 4 - Ejemplo de modelo ráster [9] 5
Figura 5 - Ejemplos del modelo vectorial [9] 6
Figura 6 - Modelo OpenGIS de flujo de trabajo de representación [11]..... 6
Figura 7 - Arquitectura Cliente-Servidor de 3 niveles [17] 11
Figura 8 - Interfaz para visualizar mapas y gráficas con CARTO [36] 14
Figura 9 - Interfaz para visualizar mapas y gráficas en Qlik [37] 14
Figura 10 - Interfaz de visualización de datos, gráficas y mapas en Tableau [38].....15
Figura 11 - Diagrama de clases del sistema 23
Figura 12 - Diagramas de componentes del sistema 24
Figura 13 - Arquitectura cliente-servidor en 3 capas del proyecto (diagrama creado por el autor).....25
Figura 14 - Burndown Chart Sprint 0.....28
Figura 15 - Burndown Chart Sprint 1.....32
Figura 16 - Burndown Chart Sprint 2.....38
Figura 17 - Burndown Chart Sprint 3.....42
Figura 18 - Burndown Chart Sprint 4.....46
Figura 19 - Burndown Chart Sprint 5.....49
Figura 20 Burndown Chart Sprint 6.....55
Figura 21 - Interfaz de Login.....57
Figura 22 - Interfaz de Registro 58
Figura 23 - Interfaz de Inicio 58
Figura 24 - Interfaz de Gestión de cuentas, roles y usuarios.....59
Figura 25 - Gestión de accesos a datos.....60
Figura 26 - Gestión de capas de análisis 60
Figura 27 - Pantalla de inicio para una usuario 61
Figura 28 - Solicitud de acceso a cuenta 61
Figura 29 - Interfaz de mapas de usuario.....63
Figura 30 - Interfaz de mapa principal.....63
Figura 31 - Configuración de estilos de una capa 65
Figura 32 - Editor de colores.....65
Figura 33 - Capas creadas de forma dinámica en el menú izquierdo.....66

Figura 34 – Interfaz de edición de una capa 66
Figura 35 - Configuración de estilos para una capa secundaria.....66
Figura 36 - Interfaz cuando se agrega una capa secundaria.....67
Figura 37 - Capa de análisis tipo barras.....67
Figura 38 - Creación de isócronas 68
Figura 39 – Información zona de isócronas 69
Figura 40 - Ventana para ver el resumen de la zona seleccionada 69
Figura 41 - Ventana para ver el reporte de la zona seleccionada 70
Figura 42 - Archivo descargado de la zona seleccionada 70
Figura 43 - Transpilación del proyecto en modo de producción 71
Figura 44 – Gráfico de resultado para una carga normal 72
Figura 45 - Comparación de tiempos de descarga para cada archivo de la aplicación para una carga normal.....72
Figura 46 - Gráfico de resultado para una carga media 73
Figura 47 - Comparación de tiempos de descarga para cada archivo de la aplicación para una carga media 73
Figura 48 - Gráfico de resultado para una carga alta 74
Figura 49 - Comparación de tiempos de descarga para cada archivo de la aplicación para una carga alta.....74
Figura 50 - Tabla de resultados para el servicio de prueba 1.....76
Figura 51 - Gráfico de resultados para el servicio de prueba 1 76
Figura 52 - Tabla de resultados para el servicio de prueba 2.....76
Figura 53 - Gráfico de resultados para el servicio de prueba 2 76
Figura 54 - Tabla de resultados para el servicio de prueba 3.....77
Figura 55 - Gráfico de resultados para el servicio de prueba 3 77

TABLAS

Tabla 1 - Descripción de pasos del modelo OpenGIS para la representación de datos geográficos [10] 7
Tabla 2 - Funciones en un mapa web interactivo [12] 8
Tabla 3 - Reuniones que se llevan a cabo en un Sprint [15] 9
Tabla 4 - Lenguajes de programación usados en el servidor web (Back-end) y los componentes web (Front-end) 11
Tabla 5 - Lenguajes de etiquetas y estilos usados para crear las interfaces web 12
Tabla 6 - Entornos en los que se implementó el servidor y las base de datos 12
Tabla 7 - Librerías que permiten agregar nuevas funcionalidades al servidor web 12
Tabla 8 - Librerías que permiten agregar nuevas funcionalidades al front-end 13
Tabla 9 - Herramientas para ejecutar pruebas 13
Tabla 10 - Plataformas que brindan una fácil implementación de funciones y características al sistema 13
Tabla 11 - Plataformas para la gestión de versiones, repositorios de proyectos y archivos 13
Tabla 12 - Tipos de estructuras de datos para el análisis de datos espaciales [41] 16
Tabla 13 - Historia de usuario épica 1 18
Tabla 14 - Historia de usuario épica 2 18
Tabla 15 - Historia de usuario épica 3 19
Tabla 16 - Historia de usuario épica 4 19
Tabla 17 - Historia de usuario épica 5 19
Tabla 18 - Historia de usuario épica 6 20
Tabla 19 - Historias de usuario del sprint 0 20
Tabla 20 - Product Backlog con tareas priorizadas y estimadas 20
Tabla 21 - Roles del equipo Scrum 22
Tabla 22 - Release plnning 25
Tabla 23 - Historia de usuario 01 26
Tabla 24 - Historia de usuario 02 26
Tabla 25 - Sprint Backlog 01 27
Tabla 26 - Tabla de avance del proyecto 28
Tabla 27 - Historia de usuario 3 29
Tabla 28 - Historia de usuario 4 29
Tabla 29 - Historia de usuario 5 29

Tabla 30 - Historia de usuario 6 30
Tabla 31 - Historia de usuario 7 30
Tabla 32 - Historia de usuario 8 31
Tabla 33 - Sprint Backlog 01 31
Tabla 34 - Tabla de avance del proyecto 32
Tabla 35 - Historia de usuario 9 33
Tabla 36 - Historia de usuario 10 33
Tabla 37 - Historia de usuario 11 34
Tabla 38 - Historia de usuario 12 34
Tabla 39 - Historia de usuario 13 34
Tabla 40 - Historia de usuario 14 35
Tabla 41 - Historia de usuario 15 35
Tabla 42 - Historia de usuario 16 36
Tabla 43 - Sprint Backlog 02 36
Tabla 44 - Tablero de avance del proyecto 37
Tabla 45 - Historia de usuario 17 38
Tabla 46 - Historia de usuario 18 39
Tabla 47 - Historia de usuario 19 39
Tabla 48 - Historia de usuario 20 40
Tabla 49 - Historia de usuario 21 40
Tabla 50 - Historia de usuario 22 41
Tabla 51 - Sprint Backlog 3 41
Tabla 52 - Tablero de avance de proyecto 42
Tabla 53 - Historia de usuario 23 43
Tabla 54 - Historia de usuario 24 43
Tabla 55 - Historia de usuario 25 44
Tabla 56 - Historia de usuario 26 44
Tabla 57 - Sprint Backlog 4 45
Tabla 58 - Tablero de avance del proyecto 46
Tabla 59 - Historia de usuario 27 46
Tabla 60 - Historia de usuario 28 47
Tabla 61 - Historia de usuario 29 47
Tabla 62 - Historia de usuario 30 48
Tabla 63 - Sprint Backlog 5 48
Tabla 64 - Tablero de avance 49

Tabla 65 - Historia de usuario épica 6 actualizada 50
Tabla 66 - Product backlog actualizado 50
Tabla 67 - Historia de usuario 31 50
Tabla 68 - Historia de usuario 32 51
Tabla 69 - Historia de usuario 33 51
Tabla 70 - Historia de usuario 34 52
Tabla 71 - Historia de usuario 35 52
Tabla 72 - Historia de usuario 36 53
Tabla 73 - Sprint backlog 6 53
Tabla 74 - Tablero de avance del proyecto 54
Tabla 75 - Plan de pruebas de acuerdo a criterios de aceptación 55
Tabla 76 - Caso de prueba 1 57
Tabla 77 - Caso de prueba 2 58
Tabla 78 - Caso de prueba 3 60
Tabla 79 - Caso de prueba 4 62
Tabla 80 - Caso de prueba 5 63
Tabla 81 - Caso de prueba 6 67
Tabla 82 - Características del servidor Front-End 71
Tabla 83 - Configuración de las categorías de cargas usadas en Jmeter 71
Tabla 84 - Características del equipo donde se desarrolló las pruebas 71
Tabla 85 - Características del servidor Back-end 75
Tabla 86 - Configuraciones de las pruebas realizadas en SoapUI 75
Tabla 87 - Características del equipo donde se realizaron las pruebas 75
Tabla 88 - Ponderación de resultados de la encuesta por medio del sistema SUS 78

RESUMEN

En los últimos años el análisis de datos se ha vuelto una estrategia vital para las empresas, ya que de esta manera pueden ser más competitivas en el mercado en el que se desenvuelven. En el caso de las empresas del tipo retail que son aquellas especializadas en la comercialización masiva de productos o servicios a grandes cantidades de clientes abarcando grandes ciudades o países, el análisis de datos espaciales les permite descubrir información valiosa que en muchos casos solo es detectable viéndolos y analizándolos a través de mapas. Hoy en día existen una gran cantidad de herramientas para realizar este tipo de análisis. Sin embargo, muchas están enfocadas a usuarios de nivel técnico. En este proyecto se desarrolló una aplicación web para visualización y análisis de datos espaciales como soporte en la toma de decisiones para la empresa Handytec. El sistema desarrollado permite mostrar información espacial de manera sencilla por medio de mapas y gráficas, ayudando al usuario dedicado a la toma de decisiones, ejecutar tareas de análisis sin tener conocimiento técnico. Para realizar esto se investigó la teoría relacionada con geo-analítica, sistemas de información geográfica, desarrollo de mapas web, interfaces de usuario y experiencia de usuario enfocada a aplicaciones de mapas, permitiendo desarrollar el sistema por medio de Scrum, una vez terminado el desarrollo se llevaron a cabo pruebas funcionales y de usabilidad obtenido los resultados esperados y consideraciones a mejoras en este tipo de aplicaciones.

Palabras clave: datos espaciales, geo-analítica, mapas web, análisis de datos, sistemas de información geográfica, Scrum, retail

ABSTRACT

In recent years the analysis of data has become a vital strategy for companies, because in this way they can be more competitive in the market in which they operate. In the case of companies of the retail type that are those specialized in the massive sale of products or services to large quantities of clients covering large cities or countries, the analysis of spatial data allows them to discover valuable information that in many cases are only detectable by viewing them and analyzing them through maps. Today there are a lot of tools to perform this type of analysis. However, many are focused on technical-level users. This project developed a Web application for visualization and analysis of spatial data as a support in decision making for the company Handytec. The developed system allows to display spatial information in a simple way by means of maps and graphs, helping the user dedicated to the decision making, executing analysis tasks without having technical knowledge. To do this, we investigated the theory related to geo-analytics, geographic information systems, web map development, user interfaces and user experience focused on map applications, allowing to develop the system by means of the Scrum, once the development was completed, functional and usability tests were carried out, obtained the expected results and considerations to improvements in this type of application.

Keywords: spatial data, geo-analytics, web maps, data analysis, geographic information systems, Scrum, retail

1. INTRODUCCIÓN

1.1. Descripción de la situación actual en el desarrollo de herramientas de Geo-analítica

Existe un creciente interés de las organizaciones del tipo retail para proveer herramientas de visualización y análisis de datos para sus empleados. Este interés se centra en obtener una visualización más amplia del negocio. Para resolver esto, es necesario integrar los datos de la empresa con la recopilación de datos espaciales y publicarlos en un mapa interactivo en línea. Este mapa debe permitir a los sus usuarios realizar análisis en una interfaz web de forma simple [1].

En el mercado existen pocas herramientas que brinden solución a este tipo de problemas. Es decir, que se enfoquen en los problemas de análisis de datos enfocado a usuarios involucrados en tomas de decisiones. Por lo tanto, es necesario crear este tipo de aplicaciones que permitan al usuario visualizar sus datos en un mapa de manera sencilla y poder usar herramientas de fácil uso que le permitan extraer información como pronósticos o comparaciones en gráficos, que sea usable y puedan ejecutar tareas de forma rápida [2].

La implementación de herramientas interactivas basadas en mapas web debe respaldar al usuario en el análisis exploratorio de datos espaciales y la toma de decisiones, para esto el sistema debe tener los instrumentos apropiados y ayudarlos a aplicarlos de acuerdo con los objetivos particulares del usuario. Es decir, su lógica de negocio, estas herramientas deben hacer uso de interfaces para especificar criterios, pesos, herramientas de visualización como gráficos de barras, del tipo pastel, líneas, o mapas de calor [3].

En el Ecuador, cada vez son más las empresas que buscan tomar ventaja de la inteligencia de negocios, en especial en los sectores con más actividad económica como Quito, Guayaquil y Cuenca, de hecho, según las estadísticas, en los últimos años el tema de desarrollo e investigación en el área de la ciencia de datos se ha vuelto más relevante, incluso que los temas de manufactura y servicios [3].

1.2. Formulación del problema

Los sistemas más usados en datos geo-referenciados como ArcGIS contienen una gran cantidad de herramientas para visualización, manejo de datos, capas, mediciones, entre otras. Sin embargo, se requiere que los usuarios inviertan tiempo en aprender a usar estas herramientas para obtener resultados relevantes. Por ejemplo, determinar en qué sectores de la ciudad se vende más o en dónde se está incrementando las ventas en los últimos

meses, como vemos estos objetivos requieren una considerable curva de aprendizaje, mientras más necesidades se quieran satisfacer mayor será el tiempo para aprender a resolverlas [4]. En muchos casos las personas más interesadas en las empresas en obtener este conocimiento estratégico son personas involucradas en la toma de decisiones, como: gerentes, juntas directivas, personal de ventas, entre otras. Las cuales, por su cargo, generalmente alto, no tendrán el tiempo para aprender a usar este tipo de soluciones. Es decir que sus necesidades estas relacionadas a la eficacia, simplicidad y rapidez. La empresa Handytec, la cual tiene entre sus principales áreas de servicio a la inteligencia de negocios, busca mejorar sus herramientas de soporte para la toma de decisiones a su cartera de clientes del tipo retail que engloba a las empresas especializadas en la comercialización masiva de productos o servicios a grandes cantidades de clientes como por ejemplo franquicias o sucursales, a los cuales la geo-analítica representa un factor estratégico en el mercado en el que se desenvuelven, por tanto el sistema desarrollado debe permitir integrar esta información geo-referenciada, en un sistema web en la nube, considerando un diseño centrado en el usuario y brindando herramientas útiles para dar soporte a la toma de decisiones.

1.3. Alcance

En este proyecto se crea una solución para la empresa Handytec, del problema mencionado, el sistema tendrá la capacidad de satisfacer las necesidades en el ámbito de las empresas del tipo retail, presentando información estratégica de sus datos espaciales de tal forma que la información presentada sea útil al momento de toma de decisiones, para esto el sistema mostrará los datos en un mapa y brindará al usuario opciones que le permitan visualizar estos datos, filtrarlos, llevar a cabo análisis por medio de graficas comparativas, búsqueda y resumen de datos. Los datos serán obtenidos a partir de la base de datos de la empresa la cual se conecta con las bases de datos de sus diferentes clientes por medio de procesos de extracción, trasformación y carga, por ende el sistema se limitará a la transformación de estos datos del tipo resumen en vistas útiles al usuario y al almacenamiento de los estados de pantallas, el sistema ayudará a la empresa procesar toda la información recolectada y presentarla en mapas a través de internet, permitiendo a sus clientes visualizar su información de forma fácil, entendible y segura.

1.4. Objetivos

1.4.1. Objetivo General

Desarrollo de una aplicación web para la visualización y análisis de datos geo-referenciados como soporte en la toma de decisiones para la empresa Handytec.

1.4.2. Objetivos Específicos

- Levantar la información necesaria relacionada con el tópico de estudio, mediante entrevistas no estructuradas realizadas y referencias bibliográficas.
- Desarrollo de interfaces: acceso, gestión de capas de datos espaciales, configuración de usuario, administración de usuarios.
- Integración de datos espaciales del usuario en la aplicación para mostrarlos en un mapa.
- Creación de herramientas de análisis geoespacial en la aplicación web: estadísticas, búsqueda, condicionales.
- Creación de interfaces de visualización de resultados tipo graficas de las herramientas de análisis geoespacial.

1.5. Marco Teórico

1.5.1. Geo-analítica

Los sistemas de información geográfica ayudan al entendimiento de una gran cantidad de datos que serían imposibles de entender sin la ayuda de un mapa. Por ejemplo, en proyectos de salud pública, ciencias ambientales, políticas y gobierno y otros [5].

En este contexto, los sistemas de geo-analítica muestran estos datos y brindan herramientas para descubrir nueva información. En el siguiente caso de uso de la geo-analítica, se necesita entender quiénes son los clientes en un área relevante para un conjunto de tiendas, y el objetivo principal es conocer la temática a la que se debe enfocar cada tienda de acuerdo al tipo de cliente más relevante en su sector como se observa en la Figura 1 [6] .

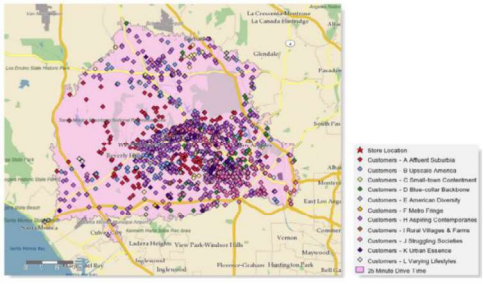

Figura 1 - Visualización de tiendas y clientes por afinidad **[6]**

Entre los principales objetivos que deben lograr los sistemas de geo-analítica tenemos [7]:

- Entendimiento: Entender dónde están las cosas, en donde están los patrones más comunes o donde varía más la información.
- Medidas: Cálculo de características geométricas como la distancia, el área.

- Relaciones - Relaciones de áreas u objetos. Por ejemplo, que está más cercano, que lugar coincide con otro.
- Búsqueda: Enfocado más a criterios de búsqueda como por ejemplo que lugar es mejor en ventas, cuales son las rutas más optimas, etc.
- Detección y cuantificación de patrones: Búsqueda de anomalías, conjunto de lugares significantes y su cambio en el tiempo.
- Predicción: Interacción entre objetos que en un cierto tiempo puedan replicar casos de éxito.

1.5.2. Analítica visual

El objetivo del análisis visual es obtener información, percepciones y evaluaciones a partir de los datos. Para entender mejor esto es necesario entender al sujeto de análisis, los modelos y datos que lo representan. El objeto de análisis es una pieza de realidad, puede ser visto como un sistema de componentes y sus propiedades las cuales están relacionados, a estas relaciones se las denomina aspectos. Un modelo es cualquier representación de los aspectos de un sujeto de análisis y sus relaciones. Los datos son observaciones registradas o mediciones que describen algunos casos de relación [8]. Así, el análisis se puede definir como el proceso de derivar un modelo de comportamiento de un sujeto a partir de datos sobre este sujeto. Estos conceptos son muy útiles en la geo-analítica. Por ejemplo, se supone que un modelo del brote de enfermedad representa dos conductas de enfoque:

1) La distribución de los casos de una enfermedad en la población.
2) Las influencias del viento, el río y los contactos entre las personas.

Con estos enfoques, se puede definir un modelo descriptivo que explique la distribución observada especificando el origen de la enfermedad y el mecanismo de propagación de la enfermedad y se puede definir un modelo predictivo que debe pronosticar cómo la enfermedad evolucionará, y un modelo de apoyo a la decisión debe respaldar la elección de acciones de mitigación [8]. Ver Figura. 2 y 3.

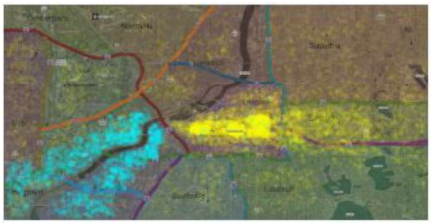

Figura 2 - Propagación de las enfermedades en dirección Este y Suroeste **[8]**

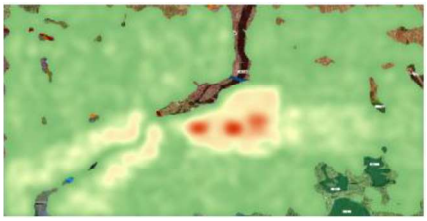

Figura 3 - Casos más comunes de la enfermedad **[8]**

1.5.3. Modelos para la información geográfica

La información geográfica debe recogerse en un formato que pueda ser entendido por el computador y que sea susceptible de emplearse dentro de un sistema. Los modelos geográficos ofrecen una concepción particular del espacio geográfico y sus atributos para lograr este entendimiento. Existen diversas alternativas en cuanto a la forma de materializar la realidad y plasmar el modelo geográfico concreto. Estas formas las podemos clasificar en dos grupos principales: modelo de representación ráster y modelo de representación vectorial [9].

1.5.3.1. Modelo ráster

En el modelo ráster, la zona de estudio se divide de forma sistemática en una serie de unidades mínimas denominadas habitualmente celdas, ver figura 4, y para cada una de estas se recoge información pertinente. Aunque la malla de celdas puede contener información sobre varias variables, lo habitual es que trate una única variable [9].

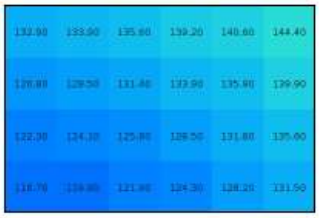

Figura 4 - Ejemplo de modelo ráster **[9]**

1.5.3.2. Modelo vectorial

En este modelo se recoge las características de un elemento geográfico mediante entidades geométricas, para cada una de las cuales se representa su espacio geográfico mediante una serie de primitivas geométricas que contienen los elementos más destacados de dicho espacio. Estas primitivas son de tres tipos: puntos, líneas y polígonos [9]. Ver Figura 5.

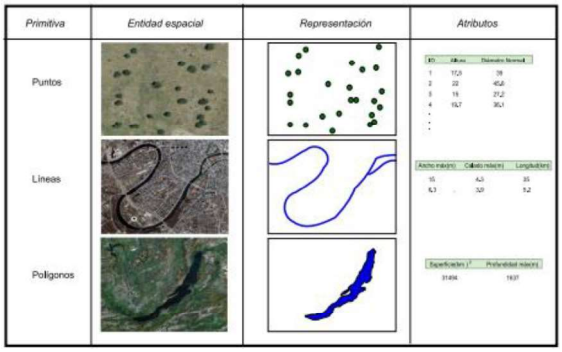

Figura 5 - Ejemplos del modelo vectorial **[9]**

1.5.4. Mapas web

La visualización de datos espaciales por medio de SIG (Sistema de información geográfica) es lo más habitual hoy en día. Sin embargo, el acceso a este tipo de software es complicado, y se requiere invertir algo de tiempo para usarlo de manera eficiente. Por estas razones los mapas web contribuyen con el acceso a datos geoespaciales [10].

Internet ha abierto camino a la implementación de este tipo de sistemas permitiendo el acceso a organizaciones, así como a numerosos hogares a una gran cantidad de datos. Además, el avance tecnológico ha permitido saber el origen de estos datos por medio de sistemas como el GPS (Global Positioning System) [10].

La implementación de mapas web debe respetar la tecnología disponible y los requisitos de la aplicación. Para esto se debe analizar los factores que determinan la implementación de este tipo de sistemas, esto es transferencia de datos, mapas web interactivos, servidores de mapas web. Con respecto a la interactividad de un mapa Web, estos se los puede categorizar desde mapas web estáticos y dinámicos. El consorcio Open GIS, cuyo fin es la definición de estándares abiertos e interoperables dentro de los Sistemas de Información Geográfica y de la World Wide Web, desarrolló un modelo esencial de representación interactiva para mapas dinámicos. Ver Figura 6.

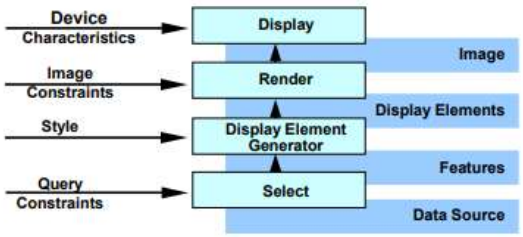

Figura 6 - Modelo OpenGIS de flujo de trabajo de representación **[11]**

En este modelo se representa de forma sencilla la transformación que los datos deben seguir en un entorno Web para desplegarse en la pantalla de un navegador web. En la siguiente tabla se detalla cada acción en el flujo de trabajo:

Tabla 1 - Descripción de pasos del modelo OpenGIS para la representación de datos geográficos **[10]**

Paso	**Descripción**
Seleccionar	Recupera datos de una fuente de datos geoespaciales de acuerdo con las restricciones de un consulta
Generador de elementos	Adjunta estilos como símbolos, estilos de línea, estilos de relleno a características espaciales, ordena los elementos de visualización en un cierto orden.
Renderizar	Toma los elementos de visualización y genera un mapa renderizado.
Mostrar	Hace que el mapa renderizado sea visible para el usuario en un dispositivo de visualización adecuado

1.5.5. UI/UX para diseñar aplicaciones de mapas

Las interfaces gráficas de usuario utilizan imágenes y gráficos para representar la entrada y salida de un programa [12]. Para desarrollar estas interfaces se debe incluir en el diseño al estudio del comportamiento del usuario final. En si el diseño de interfaces gráficas tiene que ver con un plan o dibujo que muestre la apariencia y funcionalidad de un sistema antes de ser implementado, al mismo tiempo se analiza la mejor manera de plasmar la experiencia de usuario (UX, user experience). Es decir, captar sus comportamientos, emociones y actitudes cuando usan el sistema a través de una interfaz de usuario (UI, user interface), el cual es el espacio donde las interacciones entre el humano y la maquina ocurren [12].

Para crear interfaces graficas de usuario que cumplan con las características de UI/UX se puede limitar a resolver problemas de navegación, formularios, gestión de contenido, interacciones, compatibilidad con buscadores, compatibilidad de dispositivos y responsividad, para el diseño de aplicaciones de mapas la complejidad de crear interfaces aumenta ya que se debe considerar conceptos de interacción entre usuarios y mapas, por ejemplo simbologías, leyendas, mapas base, entre otros [13].

Una forma de abordar el UI/UX en aplicaciones de mapas interactivos es agrupar las funciones inherentes a las interacciones usuario-mapa, estas son:

Tabla 2 - Funciones en un mapa web interactivo **[12]**

Grupos funcionales	**Función**
Funciones generales	Relacionado a selección de lenguaje, impresión, exportación, historia de mapas, preferencias, etc.
Funciones de navegación	Acercamientos, alejamientos, movimiento del mapa como paneos o scrolling, determinar ubicaciones, búsquedas de direcciones, etc.
Funciones didácticas	Resumen, etiquetas a lugares o áreas, textos explicativos, gráficos, imágenes, sonidos, videos, etc.
Funciones de visualización y cartografía.	Tiene que ver con leyendas, proyecciones, escalas, activar o desactivar capas, filtros, tipos de mapas, selección de datos, etc.
Funciones GIS	Creación de zonas, intersecciones, agregaciones, búsquedas espaciales, medición de áreas o distancias, acceso a base de datos, etc.

Así, de acuerdo a la necesidad del usuario se puede decidir el tipo de funciones que tendrá la aplicación. Por ejemplo, si el mapa será muy didáctico, el tipo de navegación, las características de visualización y cartografía, y que funciones GIS tendrá disponible el usuario final.

También, se debe considerar patrones de diseño de mapas, entre los más conocidos tenemos [13]:

Diseño centrado en el usuario: Implica procesos en el que las necesidades, deseos y limitaciones del usuario final de una interfaz reciben una amplia atención en cada etapa del proceso de diseño.

Estado vacío: La interfaz debe anticipar el primer paso que debe dar el usuario y proveer llamadas a la acción, evitando sobrecargar la pantalla con elementos innecesarios.

Orientado a tareas: Diseñar la experiencia de la aplicación basada en tareas. Por ejemplo, involucrando a los usuarios en flujos de trabajo claro que le permitan completar estas tareas.

Relación datos-mapa: Diseño que solo muestra secciones del mapa donde existen datos.

1.5.6. Scrum

Scrum es un marco de trabajo ágil para el desarrollo de proyectos. Por medio de este marco de trabajo los equipos de desarrollo pueden abordar problemas adaptativos al realizar un proyecto, a la vez que se entregan productos o resultados del mismo con el máximo valor. Estos resultados se entregan de manera rápida en un lapso de 2 a 4 semanas, evita la

documentación excesiva, permite gestionar proyectos con un alto riesgo de cambios, permitiendo que el desarrollo del proyecto sea flexible a la hora realizar cambios durante el proceso de desarrollo del proyecto. El marco de trabajo Scrum se compone por Artefactos, Eventos y el Equipo Scrum [14].

1.5.6.1. Los eventos Scrum

El objetivo de los eventos es por una parte reducir reuniones innecesarias y tener una oportunidad de inspección y adaptación dentro de la ejecución del proyecto. Los eventos tienen una duración máxima [14].

Sprint

Es el periodo de tiempo limitado dentro del cual se crea y entrega un incremento "Terminado" del producto final. Un Sprint comienza inmediatamente después de que finaliza otro Sprint.

Tabla 3 - Reuniones que se llevan a cabo en un Sprint **[15]**

Eventos	**Descripción**
Sprint Planning	Se planifica el trabajo a realizar en el Sprint. Participa todo el equipo Scrum, de acuerdo al tamaño del equipo el tiempo de duración de un máximo de 8 horas.
Daily Scums	Es una reunión con un tiempo máximo de 15 minutos. Permite sincronizar actividades al Scrum team, en esta reunión se identifican problemas que impidan el avance del proyecto y en caso de ser necesario se crea un plan de acción para resolver estos problemas.
Sprint Review	Se lleva a cabo al final del Sprint, permite presentar e inspeccionar el incremento del producto entregado y modificar la pila del Producto (Product Backlog), tiene una duración máxima de 4 horas. En esta reunión el Equipo Scrum y los interesados colaboran para determinar las siguientes tareas que deben hacerse para agregar valor al producto final.
Sprint Retrospective	Esta reunión permite al equipo Scrum revisar que se hizo bien y que se hizo mal en la implementación de Scrum y así crear un plan de mejoras para la adopción del mismo. Se realiza entre el Sprint Review y el siguiente Sprint Planning. Tiene un máximo de tiempo de 3 horas. Participan el Scrum Master y el Scrum team.

1.5.6.2. El equipo Scrum y sus roles (Scrum team)

El Equipo Scrum consiste en un Propietario del Producto (Product Owner), el Equipo de Desarrollo (Development Team) y un Scrum Master [14].

- El propietario del producto (Product owner): Es la única persona responsable de maximizar el valor del producto y el trabajo del equipo de desarrollo. Gestiona la pila tareas del producto, explicando claramente cada tarea y ordenarlas de acuerdo a la prioridad de cada tarea.
- El equipo de desarrollo (Scrum Team): Es el equipo responsable de realizar y entregar un incremento de producto "Terminado" que se agregará al producto final de un Sprint.
- El Scrum master: Es la persona encargada de asegurar que se entienda y se adopte Scrum. Debe asegurar que todo el equipo entiende la teoría, y sigue las prácticas y reglas de Srum. Facilita a llevar a cabo los eventos de Scrum según se requiera o necesite.

1.5.6.3. Los artefactos

Permiten representar el trabajo a realizar y el valor en el producto a medida que se desarrollan los Sprints. Su principal objetivo es proporcionar transparencia de la información clave, la inspección y adaptación. Los artefactos de Scrum son [14]:

- Pilas de tareas a desarrollar (Product Backlog): Es una lista ordenada de todo lo que tendrá el producto final, es la única fuente de requerimientos que se desarrollarán durante los Sprints. Es el Product Owner quien gestiona la lista, determinando su contenido, disponibilidad y ordenación de acuerdo a una prioridad. En la lista se incluyen funcionalidades, requisitos, mejoras y correcciones que se realizará sobre el producto. Cada elemento de lista tiene como mínimo una descripción, un orden, la estimación y el valor, siendo el Scrum Team quien realiza la estimación a cada ítem.
- Pila de tareas del Sprint (Sprint Bacaklog): Es la lista de obtenida a partir de la selección de elementos con más prioridad que se desarrollarán en el Sprint y que encuentran en la parte superior del Product Backlog. Representa una predicción por parte del Scrum Team acerca de que funciones o características tendrá el incremento del producto al final de Sprint y el trabajo que se tiene que realizar para terminar con éxito el Sprint.

- Incremento: Es la suma de todos los elementos completados durante el Sprint, así, al finalizar el Sprint se tiene un incremento terminado que está en condiciones de ser utilizado y que puede ser agregado al producto final.

1.5.7. Arquitectura Cliente-Servidor

En la arquitectura de cliente-servidor para el desarrollo de sistemas informáticos se tiene dos actores principales, el servidor web que actúa como proveedor de recursos y el cliente con el rol de solicitador y lector de estos recursos. Para el sistema desarrollado en este proyecto el cliente es la aplicación en Angular quien solicita datos a través de servicios RESTful al servidor desarrollado en NodeJS [16].

Para una arquitectura cliente-servidor en tres niveles, el servidor web de nivel 2 depende de un servidor de datos para cumplir su rol, este servidor de datos de nivel 3 responderá a las peticiones del servidor web generalmente por medio de consultas SQL (Structured Query Language). Ver Figura 7. En esta arquitectura el cliente en el nivel uno es totalmente independiente de los demás niveles [17].

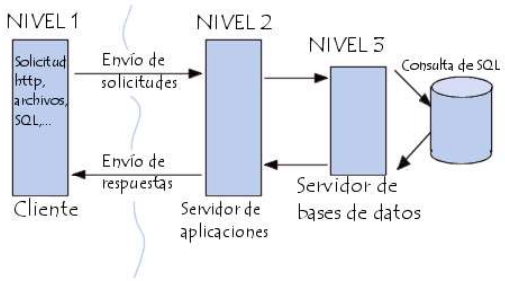

Figura 7 - Arquitectura Cliente-Servidor de 3 niveles **[17]**

1.5.8. Herramientas de desarrollo

Durante el desarrollo del proyecto se usaron herramientas que permiten desplegar aplicaciones web, para esto se necesitó de un lenguaje de programación y librerías a nivel de cliente y servidor de la aplicación. Además, se necesitó de una base de datos de la clase NewSQL para trabajar con datos del tipo JSON los cuales se adaptan de mejor manera al funcionamiento del mapa web desarrollado. A continuación, el detalle de las herramientas usadas.

Tabla 4 - Lenguajes de programación usados en el servidor web (Back-end) y los componentes web (Front-end)

Nombre	Descripción
Javascript	JavaScript (JS) es un lenguaje ligero e interpretado, orientado a objetos, más conocido como el lenguaje de script para páginas web. Es un

	lenguaje script basado en prototipos, dinámico, soporta estilos de programación funcional, orientada a objetos [18].
Typescript	Es un lenguaje que aporta comprobación de tipos a JavaScript para que pueda detectar problemas antes de ejecutar código, o incluso antes de guardar un archivo. Incluye las últimas características de JavaScript del estándar ECMAScript [19].

Tabla 5 - Lenguajes de etiquetas y estilos usados para crear las interfaces web

Nombre	**Descripción**
HTML	HTML, que significa Lenguaje de Marcado para Hipertextos (HyperText Markup Language) es el elemento de construcción más básico de una página web y se usa para crear y representar visualmente una página web [20].
SASS	Sass es una extensión de CSS. Permite usar variables, reglas anidadas, combinaciones, importaciones en línea, todo con una sintaxis totalmente compatible con CSS. Ayuda a mantener las hojas de estilo grandes bien organizadas [21].

Tabla 6 - Entornos en los que se implementó el servidor y las base de datos

Nombre	**Descripción**
Node JS	Node JS es un entorno de ejecución para JavaScript construido con el motor de JavaScript V8 de Chrome que se ejecuta en el lado del servidor. Permite por medio de su entorno crear y ejecutar un servidor web [22].
MemSQL	Entra dentro de la clase NewSQL en la cual se trata de conseguir el mismo rendimiento escalable de las bases de datos NoSQL y manteniendo las propiedades ACID de un sistema relacional tradicional, soporta operaciones geo-espaciales [23].

Tabla 7 - Librerías que permiten agregar nuevas funcionalidades al servidor web

Nombre	**Descripción**
Express	Permite configurar y levantar un servidor web en Node JS [24].
Mysql	Librería que permite realizar conexiones a una base de datos MySQL, y ejecutar consultas. [25]
Child process	Librería nativa de Node que permite crear procesos hijos en el lado del servidor y evitar que este se detenga al ejecutar cálculos exhaustivos [26].

Tabla 8 - Librerías que permiten agregar nuevas funcionalidades al front-end

Nombre	Descripción
Angular	Plataforma que facilita la creación de aplicaciones web de una sola página. Angular combina plantillas declarativas, inyección de dependencias, librerías [27].
Leaflet	Librería JavaScript de código abierto para crear mapas web interactivos. Leaflet está diseñado teniendo en cuenta la simplicidad, el rendimiento y la facilidad de uso [28].
Angularfire	Mediante programación reactiva permite interactuar con los principales servicios de Firebase, esto es: autenticación, base de datos, y almacenamiento de archivos [29].
Rxjs 6	Es una librería para programación reactiva que utiliza observables, facilita la composición de código asíncrono o basado en devolución de llamadas (callbacks) [30].

Tabla 9 - Herramientas para ejecutar pruebas

Nombre	Descripción
Jmeter	Software de código abierto, una aplicación Java diseñada realizar pruebas funcionales y de carga enfocado en aplicaciones web. [31]
SoapUI	Herramienta de prueba automatizada para las API de SOAP y REST. Permite escribir, ejecutar, integrar y automatizar pruebas API avanzadas con facilidad. [32]

Tabla 10 - Plataformas que brindan una fácil implementación de funciones y características al sistema

Nombre	Descripción
Firebase	Plataforma integral para el desarrollo de aplicaciones móviles y web, ofreciendo herramientas y servicios de autenticación, base de datos, almacenamiento de archivos, hosting, etc [33].

Tabla 11 - Plataformas para la gestión de versiones, repositorios de proyectos y archivos

Nombre	Descripción
Bitbucket	Bitbucket es una forja para alojar proyectos utilizando el sistema de control de versiones [34].

1.6. Investigación preliminar

1.6.1. Ventajas y desventajas del software web existente para el análisis de datos geo-referenciados

Para el caso de datos geo-referenciados de objetos o entidades las cuales están involucradas en un objeto de interés o análisis, por ejemplo las farmacias en la ciudad de Quito perteneciente a una distribuidora farmacéutica [35].

Entre las principales sistemas encontrados se encuentra CARTO, esta herramienta posee una gran cantidad de herramientas para crear mapas para el análisis de datos, permite crear capas de análisis y graficas con la ayuda del lenguaje Python (Ver figura 8), debido a esto, la herramienta sirve de mucha ayuda para usuarios del tipo técnico como los científicos de datos y no para personas involucradas en la toma de decisiones ya que estas no tienen el tiempo o conocimiento para manejar este lenguaje [36].

Figura 8 - Interfaz para visualizar mapas y gráficas con CARTO **[36]**

Qlik Geoanalytics cuenta con una gran cantidad de herramientas permitiendo crear una amplia variedad de mapas, su interfaz es menos técnica (Ver Figura 9). Sin embargo, se encontró poca documentación en cuanto a integración de datos, haciendo solo referencia a la carga de archivos con datos geográficos y llamadas a servicios que los proporcionen [37].

Figura 9 - Interfaz para visualizar mapas y gráficas en Qlik **[37]**

Con Tableau Online el usuario puede crear espacios de trabajo en los que puede incluir mapas de análisis, cuenta con integración de datos por medio de base de datos en la nube, servicios o por medio de archivos. Al estar los datos almacenados en la nube, el usuario puede crear tableros, publicarlos a otros usuarios y gestionar permisos (Ver Figura 10), debido a esta gran poder de análisis, visualización e integración el usuario requiere de

tiempo para aprender a usar estas herramientas, en la página oficial el usuario puede seguir una serie de tutoriales en lo que a manera de curso online se puede aprender a usar este aplicación. [38].

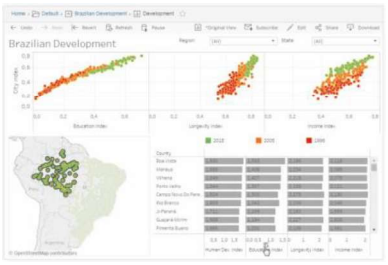

Figura 10 - Interfaz de visualización de datos, gráficas y mapas en Tableau **[38]**

1.6.2. Principales herramientas de sistemas de información geográfica SIG para el análisis de datos

Da acuerdo al enfoque que se quiera obtener al momento de crear mapas dinámicos se puede dividir a las herramientas para sistemas de mapas web en cuatro conjuntos [39]:

- Análisis de patrones: Herramientas para identificar y visualizar características comunes en determinados sectores.
- Buscador de ubicaciones: Herramientas para realizar búsquedas de acuerdo a criterios especificados, esta búsqueda puede ser por medio de atributos o elementos espaciales.
- Resumen de datos: Herramientas que permiten realizar recuentos, estadísticas descriptivas y mediciones dentro de áreas
- Proximidad: Herramientas para buscar relaciones de entidades de acuerdo a su distancia.

Un mapa web dinámico tendrá un set de herramientas necesarias de acuerdo al enfoque de análisis que se desee desarrollar. Por ejemplo, la complejidad, el tipo de usuario final, la usabilidad y el rendimiento [40].

1.6.3. Tipos de estructura de datos usado para el análisis de datos espaciales

Al momento de realizar análisis espaciales es necesario identificar qué información se adapta bien en mi modelo da análisis, para esto es necesario relacionar a las estructuras de datos y tipos de análisis que permiten llevar a cabo mi modelo de análisis, así, se identifican tres tipos de estructuras de datos:

Tabla 12 - Tipos de estructuras de datos para el análisis de datos espaciales **[41]**

Estructura de datos	Descripción
Datos de ubicación	Datos con la información espacial que permiten a los mismos ser representados en un mapa.
Datos de atributos	Hacen referencia a datos que varían a través de tiempo, por ejemplo transacciones mensuales de tiendas.
Datos de interacción	Datos que representan la relación entre datos de ubicación de acuerdo a sus atributos. Por ejemplo el flujo de personas de una lugar a otro.

1.6.4. Ventajas y desventajas de los modelos de información geográfica al momento de ser usados en aplicaciones de mapas

De acuerdo al tipo de uso de los datos geográficos y a los sistemas que los implementarán, la facilidad de uso de cada modelo varía, por ejemplo, el modelo ráster requiere más espacio de almacenamiento en comparación con el modelo vectorial en el que se suelen manejar archivos livianos, el modelo vectorial es más preciso ya que su naturaleza se basa en figuras geométricas a diferencia del modelo ráster que se basa en proximidad de celdas, en cuanto a la facilidad de análisis, en el modelo vectorial se vuelve más complicado debido a que los elementos están limitados a puntos, líneas y polígonos, en cambio en el modelo ráster es más simple y rápido al realizar operaciones de análisis geográficos, y es más eficiente al realizar análisis de superficies como por ejemplo el tipo de análisis que realizan los satélites. En resumen la decisión para elegir un modelo u otro se basa la precisión más exacta que puedo lograr al representar objetos en un mapa, la capacidad de almacenamiento y procesamiento de mi sistema, y la complejidad del tipo de análisis que voy a realizar [42]. En un entorno web en donde la capacidad procesamiento se limita a la del navegador, a la velocidad de conexión y por ende la capacidad de transferencia de grandes cantidades de datos es recomendable usar el modelo vectorial debido a su mejor rendimiento y adaptabilidad a los navegadores.

2. METODOLOGIA

En esta sección se describe el proceso de desarrollo del sistema web usando el marco de trabajo Scrum.

2.1. Definición del proyecto

La definición del proyecto consta de la definición de requerimientos, la creación del Prodcut Backlog, la definición de roles del proyecto y el diseño de la arquitectura del proyecto.

2.1.1. Definición de los requerimientos

A continuación se describe los requerimientos obtenidos por parte del usuario, los módulos a desarrollar y los prototipos de pantallas creadas

2.1.1.1. Descripción de los requerimientos por parte del usuario

Los requerimientos fueron obtenidos por medio de tres entrevistas no estructuradas al gerente de la empresa Handytec. Se puede ver las anotaciones relevantes obtenidas en las entrevistas en el Anexo A. Los requerimientos obtenidos se los resume de la siguiente manera:

El sistema debe ser accesible a través de una aplicación web y debe tener la posibilidad de ser implementado en la nube. Para el acceso es necesario tener un login y un registro de usuarios, existirá un administrador que gestione cuentas y roles. Además, el administrador será el responsable de configurar la información acceso de datos de cada cuenta.

El rol de usuario estará asociada a una cuenta, el usuario verá su información básica y la podrá editar, el usuario no podrá acceder a la configuración de la cuenta, pero podrá ver la pantalla de gestión de mapas, dentro de esta interfaz el usuario podrá crear accesos a la pantalla de un mapa, por lo tanto el usuario podrá crear varios mapas.

En una pantalla de mapa el usuario podrá crear capas principales con marcadores etiquetados de acuerdo al tipo de datos seleccionados, dentro de la pantalla de creación de una capa principal se debe especificar los colores de acuerdo al tipo de dato seleccionados.

El usuario podrá editar la capa principal y filtrar datos de acuerdo al tipo de dato creado y podrá activar desactivar cada capa por medio de una lista en un menú.

El usuario podrá crear por cada capa principal una capa para los datos que varían de mes a mes, estas capas deben presentar información adicional al dato localizado por medio de mapas de calor o gráficas, cuando se crea una capa de análisis temporal el sistema debe mostrar una barra de tiempo de los últimos 12 meses de la fecha actual, cuando el usuario cambie de mes el sistema debe actualizar las capas adicionales activas.

El estado del mapa se debe guardar, esto es posición, altitud, capas activadas, para que el usuario no pierda la noción del trabajo creado en ese mapa. A partir de las entrevistas se generaron prototipos de interfaces para el rol de usuario final y el rol de administrador, ver anexo D.

2.1.1.2. Identificación de módulos del sistema

De acuerdo a la descripción del usuario se identifican 6 módulos a desarrollar que son las historias de usuario épicas del proyecto, estos son:

- Desarrollar el acceso a la pantalla principal
- Desarrollar la gestión de cuentas, usuarios y acceso a datos
- Desarrollar la gestión del perfil de usuario
- Desarrollar la gestión y visualización de mapas de mapas
- Desarrollar la gestión de capas principales y secundarias
- Desarrollar las herramientas de apoyo

A continuación, la descripción de los módulos en forma de historias de usuario épicas:

Tabla 13 - Historia de usuario épica 1

	Historia de Usuario Épica
HUE01	• **Usuario:** Desarrollar el acceso a la pantalla principal por medio de la gestión de cuentas y usuarios
Nombre historia: Acceso a la pantalla principal	
Descripción: Como usuario quiero regístrame en el sistema, ingresar, poder ver la pantalla de inicio de la aplicación.	

Tabla 14 - Historia de usuario épica 2

	Historia de Usuario Épica
HUE02	**Usuario:** Administrador
Nombre historia: Desarrollar la gestión de cuentas, usuarios y acceso a datos	
Descripción: Como administrador quiero crear, leer, actualizar y eliminar cuentas, roles, usuarios y configuraciones para el acceso a datos.	

Tabla 15 - Historia de usuario épica 3

	Historia de Usuario Épica
HUE03	**Usuario:** Usuario general
Nombre historia: Desarrollar la gestión del perfil de usuario	
Descripción: Como usuario quiero ver y editar mi información personal	

Tabla 16 - Historia de usuario épica 4

	Historia de Usuario Épica
HUE04	**Usuario:** Usuario general
• **Nombre historia:** Desarrollar la gestión y visualización de mapas de mapas	
Descripción: Como usuario quiero gestionar mis mapas, esto es ver una lista de mapas, crear un mapa, editar su descripción y título, eliminar un mapa de la lista. Además, quiero que al dar clic en la opción para abrir un mapa en la interfaz de "Mis mapas" se abra otra pantalla que muestre un mapa e pantalla completa.	

Tabla 17 - Historia de usuario épica 5

	Historia de Usuario Épica
HUE05	**Usuario:** Usuario general
Nombre historia: Desarrollar la gestión de capas principales y secundarias	
Descripción: Como usuario quiero crear, listar, editar y eliminar una capa principal, activar y desactivar una capa principal en el mapa mediante un checkbox. Además, quiero crear, listar, editar y eliminar una capa secundaria dentro de una capa principal, quiero activar y desactivar un capa secundaria en el mapa mediante un checkbox	

Tabla 18 - Historia de usuario épica 6

	Historia de Usuario Épica
HUE06	**Usuario:** Usuario general
Nombre historia: Desarrollar las herramientas de apoyo	
Descripción: Como usuario quiero usar las herramientas de isócronas, búsqueda de direcciones, y conteo de objetos por área.	

2.1.2. Product Backlog

Para cada historia de usuario épica se identificó tareas la cuales se las organizará en historias de usuario que entrarán al Product Backlog, ver anexo B. Se usa el método de puntos de historia para estimar las historias de esta forma es más fácil dimensionar y relacionar la complejidad de las historias de usuario con respecto a otras, la referencia será la creación de la interfaz de login que tendrá un valor de 3 puntos de esfuerzo, por cada sprint se desarrollarán 25 puntos de esfuerzo. A las historias de usuario del sprint 0 no se las considerara dentro del Product Backlog. Mediante las gráficas Burndown Chart, ver anexo E, se inspeccionará el trabajo por hacer en cada Sprint.

Nomenclatura:

HUE: Historia de usuario épica

PH: Puntos de esfuerzo de historia de usuario

COD: Código de historia de usuario

Tabla 19 - Historias de usuario del sprint 0

COD	Historia de usuario	Prioridad	PH
HU01	Instalación del ambiente de desarrollo	Alta	10
HU02	Implementación del modelo de base de datos	Alta	10

Tabla 20 - Product Backlog con tareas priorizadas y estimadas

Product Backlog				
HUE	**COD**	**Historia de usuario**	**Prioridad**	**PH**
HUE1	HU03	Registro de un usuario	Alta	3
	HU04	Login	Alta	3
	HU05	Creación de la interfaz de la pantalla principal	Alta	3

	HU06	Implementación de seguridades de acceso a páginas por roles	Alta	7
HUE2	HU07	Creación de cuentas	Alta	3
	HU08	Lista de cuentas	Alta	3
	HU09	Actualización y eliminación de cuentas	Alta	3
	HU10	Lista de usuarios	Alta	1
	HU11	Actualización de usuario	Alta	1
	HU12	Eliminación de usuarios	Alta	2
	HU13	Creación de configuración para el accesos a datos localizados	Alta	2
	HU14	Listar, actualización y eliminación de configuraciones de capas principales	Alta	7
	HU15	Creación de opciones para la configuración de accesos a capas principales	Alta	2
	HU16	Listar, actualización y eliminación de opciones para la configuración de accesos a capas principales	Alta	5
	HU17	Creación de configuración de acceso a capa secundaria	Alta	3
	HU18	Listar, actualización y eliminación de configuraciones de accesos a capas secundarias	Alta	7
HUE3	HU19	Actualización de datos de usuario	Baja	3
HUE4	HU20	Creación de información de mapa	Alta	3
	HU21	Listar, actualización y eliminación de información de mapas creados	Alta	5
	HU22	Creación de la interfaz para visualizar un mapa en pantalla completa.	Alta	3
	HU23	Creación de controles de navegación y menú en el mapa	Media	7
HUE5	HU24	Creación de capas principales	Alta	10
	HU25	Lista de capas principales en el menú del mapa	Alta	4
	HU26	Actualización y configuración de capas principales	Alta	4
	HU27	Eliminación de la capa principales	Alta	5
	HU28	Creación de una capa secundaria	Alta	10

	HU29	Lista de capas secundarias dentro de una capa principal en el menú	Alta	5
	HU30	Actualización y configuración de una capa secundaria	Alta	5
	HU31	Eliminación de una capa secundaria	Alta	3
HUE6	HU32	Creación de la funcionalidad de búsqueda de direcciones y leyendas	Media	3
	HU33	Creación de isócronas	Alta	5
	HU34	Creación de la funcionalidad de conteo por área	Alta	5

2.1.3. Definición de roles

A continuación, se describen los roles Scrum para el equipo de trabajo:

Tabla 21 - Roles del equipo Scrum

Roles	**Responsable**
Scrum Master	Dra. Tania Calle
Product Owner	MSc. Jorge Jaramillo
Scrum Team	Roberto Prado

2.1.4. Arquitectura del proyecto

A continuación se describe mediante un diagrama la estructura del sistema de acuerdo a las clases identificadas en los módulos, en el diagrama se observa las clases para la gestión de usuarios y cuentas interactuando con las clases que representan le gestión de mapas y capas en el sistema, estas interacciones permiten modelar de manera general las vistas del diseño que permitirán a un usuario al cual se le ha asignado una cuenta, gestionar mapas, y dentro de cada mapa gestionar capas y herramientas, ver figura 11:

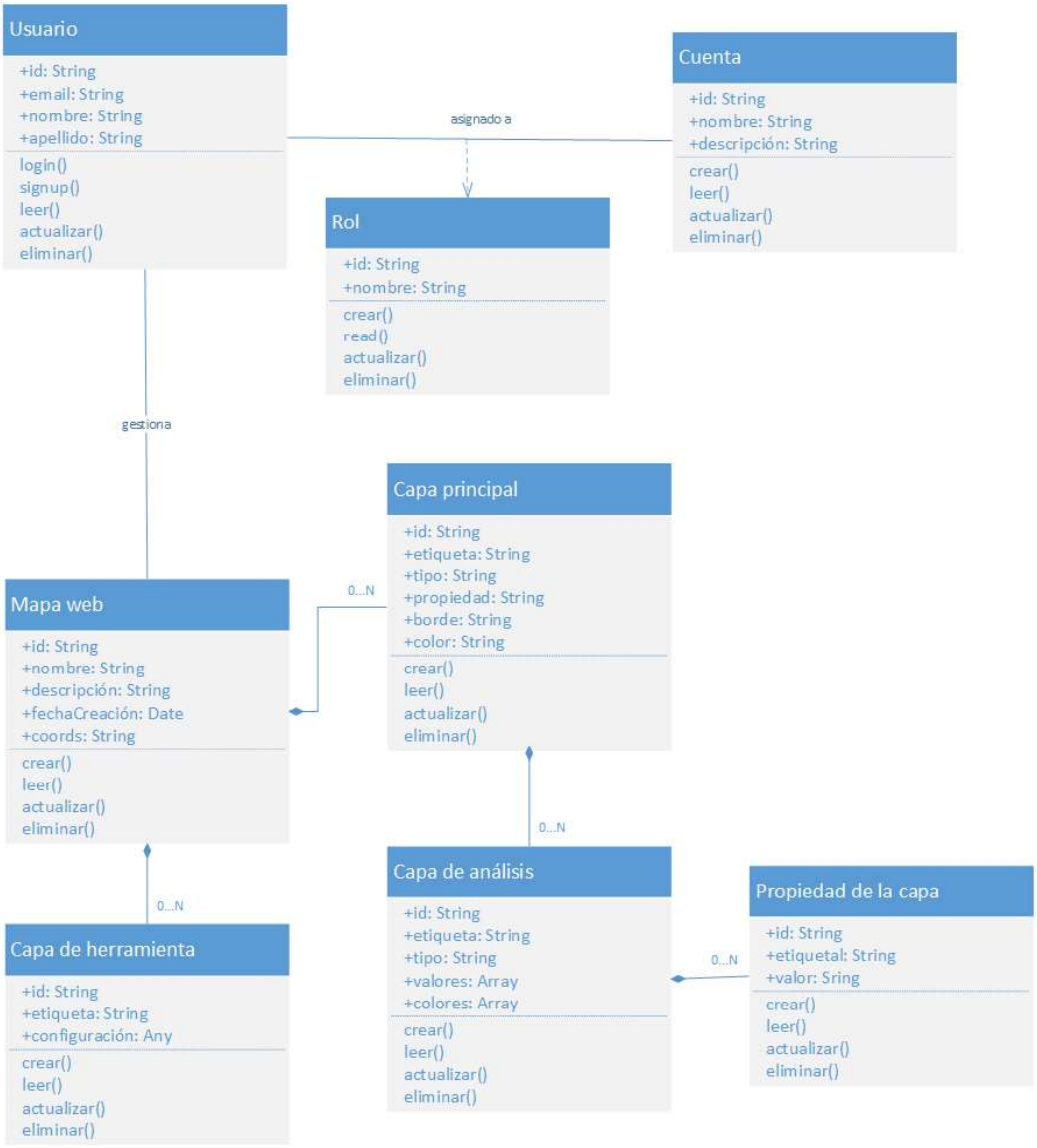

Figura 11 - Diagrama de clases del sistema

En el siguiente diagrama se muestra el diseño de la arquitectura del sistema a nivel de componentes y las dependencias de los mismos, se representan los módulos identificados y la interacción de los componentes. En la parte superior derecha se encuentra el módulo que representa al componente inicial del sistema encargado de cargar las interfaces de la aplicación. También, en la parte superior izquierda se detalla los componentes para la gestión de acceso al sistema. En la parte inferior izquierda se encuentra los módulos con los componentes para la gestión de cuentas y configuraciones de capas. El modulo para la gestión del perfil de usuario se encuentra en la parte central. En la parte lateral derecha se encuentra el módulo con los paquetes para la representación de mapas y en la parte inferior de este módulo se encuentra el módulo con los paquetes para las herramientas del mapa. Además, en la parte inferior central se encuentra el módulo con los paquetes para el acceso a datos de configuración de cuentas y a la base de datos, ver figura 12:

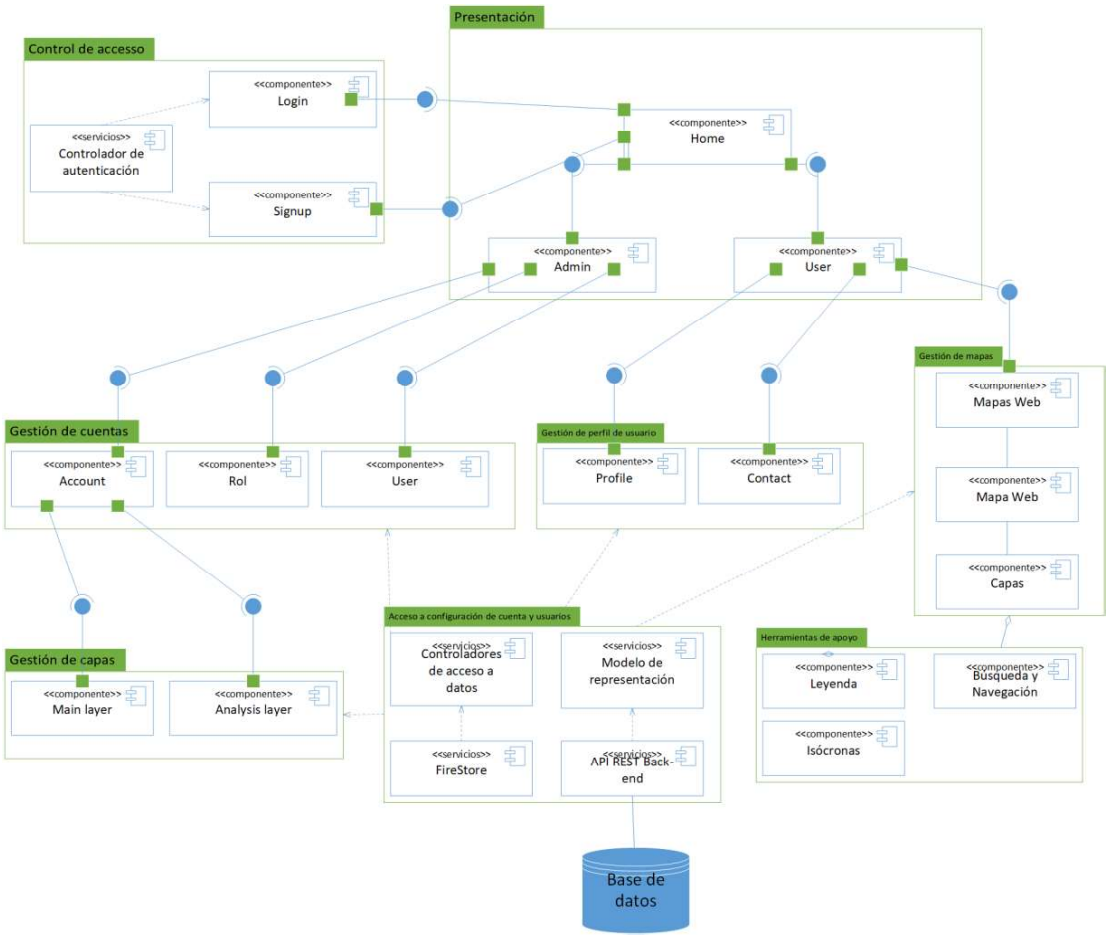

Figura 12 - Diagramas de componentes del sistema

En este proyecto se usa la arquitectura cliente-servidor en 3 capas, de esta forma se separa la funcionalidad del cliente en donde por medio del navegador se accederá a la aplicación en Angular y por medio del servidor web se provee de datos a la aplicación. Ver Figura 11.

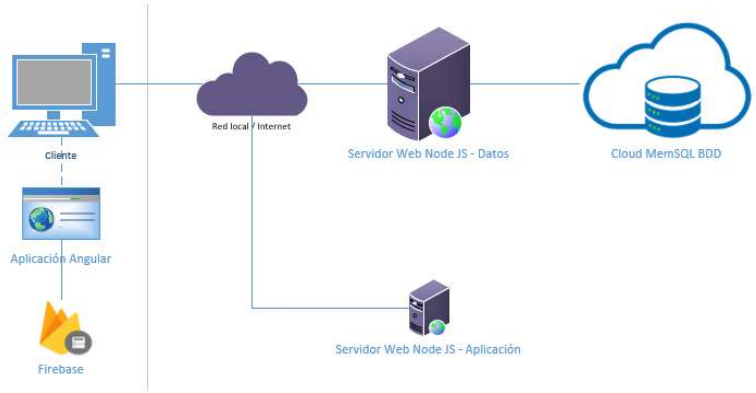

Figura 13 - Arquitectura cliente-servidor en 3 capas del proyecto (diagrama creado por el autor)

En la parte izquierda se observa la integración con Firebase que brinda el servicio de autenticación y de base de datos que es donde se guarda la configuración del usuario en tiempo real.

2.2. Release Planning

Se ejecutarán 6 Sprints, cada uno con una duración de 2 semanas, dentro de cada ejecución se planea desarrollar 25 puntos de esfuerzo.

Tabla 22 - Release plnning

Release Planning						
Sprint 0	**Sprint 1**	**Sprint 2**	**Sprint 3**	**Sprint 4**	**Sprint 5**	**Sprint 6**
HU01	HU03	HU09	HU17	HU23	HU27	HU31
HU02	HU04	HU10	HU18	HU24	HU28	HU32
	HU05	HU11	HU19	HU25	HU29	HU33
	HU06	HU12	HU20	HU26	HU30	HU34
	HU07	HU13	HU21			
	HU08	HU14	HU22			
		HU15				
		HU16				

2.3. Ejecución de los Sprints

2.3.1. Sprint 0

2.3.1.1. Sprint Planning

Objetivo del sprint:

- Preparar el entorno de desarrollo y el modelo de la base de datos.

Historias de Usuario para este Sprint

Tabla 23 - Historia de usuario 01

		Historia de Usuario
HU01	**Título:** Instalación del ambiente de desarrollo	
Prioridad: Alta		**Riesgo en desarrollo:** Baja
Puntos estimados: 10		**Sprint:** 0
Programador responsable: Roberto Prado		
Criterios de aceptación: 1.- Node JS instalado 2.- Aplicación Angular base creada 3.- Servidor web Express JS levantado 4.- Conexión a la base de datos con un cliente SQL 5.- Repositorio del proyecto		

Tabla 24 - Historia de usuario 02

		Historia de Usuario
HU02	**Título:** Implementación del modelo de base de datos	
Prioridad: Alta		**Riesgo en desarrollo:** Baja
Puntos estimados: 15		**Sprint:** 0
Programador responsable: Roberto Prado		
Criterios de aceptación: 1.- Modelo de base datos de acuerdo a la sección 1.6.4 2.- Proyecto en firebase para usar los servicios de autenticación y la base de datos.		

2.3.1.2. Sprint Backlog

Tabla 25 - Sprint Backlog 01

Cod	Prioridad	Estimación PH
HU01	Alta	10
HU02	Alta	15

2.3.1.3. Sprint Review

Las actividades realizadas en este sprint fueron:

- Se instaló la versión 8.12 de Node JS en el equipo de desarrollo.
- Se instaló de las librerías Angular-cli, Express JS y express-generator.
- Se creó la aplicación Angular versión 6.
- Se creó y configuro el servidor web para una conexión con bases de datos MySQL.
- Se accedió a la base de datos de la empresa por medio del cliente SQL DBeaver.
- Se creó el repositorio privado del proyecto en BitBucket.
- Se instaló la aplicación SourceTree para el majeo de repositorios en BitBucket.

Para la creación del modelo de base de datos, el cual se lo puede ver en el anexo C, se procedió a crear dos nuevas tablas en la base de datos debido a que las tablas que contenían los datos a ser usados no cumplían con la estructura recomendada. Para esto se usó el motor de integración de Pentaho para transformar estas tablas. Para la relación con la base de datos de firebase que se lo realiza por medio de la aplicación se usa el campo “account_name”. Con estas actividades completas que cubren todos los criterios de aceptación, el Sprint 0 se completó satisfactoriamente.

2.3.1.4. Sprint Restrospective

Debido al conocimiento previo trabajando con servidores en Node JS y aplicaciones en Angular las tareas de la historia HU01 se desarrollaron de forma rápida, en al caso de la historia HU02 se necesitó de más tiempo para la investigación y diseño del modelo de base de datos. A continuación se observa el tablero de avance del proyecto:

Tabla 26 - Tabla de avance del proyecto

Pendientes	En desarrollo	Finalizadas
HU03, HU04, HU05, HU06, HU07, HU08, HU09, HU10, HU11, HU12, HU13, HU14, HU15, HU16, HU17, HU18, HU19, HU20, HU21, HU22, HU23, HU24, HU25, HU26, HU27, HU28, HU29, HU30, HU31, HU32, HU33, HU34		HU01, HU02

Burndown Chart

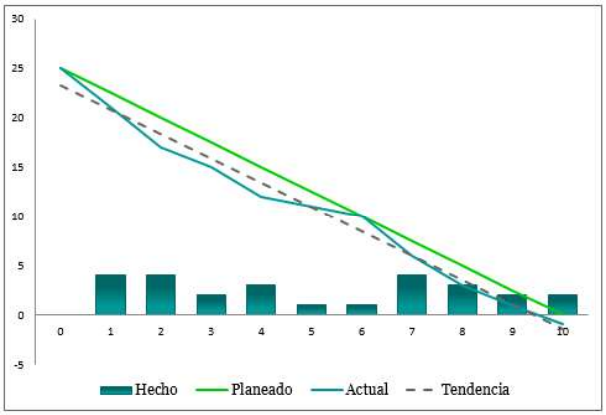

Figura 14 - Burndown Chart Sprint 0

2.3.2. Sprint 1

2.3.2.1. Sprint Planning

Objetivo del Sprint

- Desarrollar el acceso a la pantalla principal
- Desarrollar la gestión de cuentas y usuarios

Historias de Usuario para este Sprint

Tabla 27 - Historia de usuario 3

Historia de Usuario	
HU03	**Título:** Registro de un usuario
Prioridad: Alta	**Riesgo en desarrollo:** Media
Puntos estimados: 3	**Sprint:** 1
Programador responsable: Roberto Prado	
Criterios de aceptación: 1.- Al abrir la opción de crear cuenta se observa el formulario de registro. 2.- Registro por correo, nombre y apellido, contraseña.	

Tabla 28 - Historia de usuario 4

Historia de Usuario	
HU04	**Título:** Login
Prioridad: Alta	**Riesgo en desarrollo:** Baja
Puntos estimados: 3	**Sprint:** 1
Programador responsable: Roberto Prado	
Criterios de aceptación: 1.- Cuando se abre la aplicación en el navegador se observa la pantalla de login. 2.- Ingreso por mi correo y una contraseña a la aplicación.	

Tabla 29 - Historia de usuario 5

Historia de Usuario	
HU05	**Título:** Creación de la interfaz de la pantalla principal
Prioridad: Alta	**Riesgo en desarrollo:** Media
Puntos estimados: 3	**Sprint:** 1

Programador responsable: Roberto Prado
Criterios de aceptación: 1.- Pantalla que muestre las opciones de navegación 2.- Navegación a las páginas: inicio, cuentas y usuarios, mis mapas, mapa individual.

Tabla 30 - Historia de usuario 6

Historia de Usuario		
HU06	**Título:** Implementación de seguridades de acceso a páginas por roles	
Prioridad: Alta		**Riesgo en desarrollo:** Alta
Puntos estimados: 7		**Sprint:** 1
Programador responsable: Roberto Prado		
Criterios de aceptación: 1.- Gestión de roles 2.- Restricción de accesos a páginas de la aplicación de acuerdo al rol		

Tabla 31 - Historia de usuario 7

Historia de Usuario		
HU07	**Título:** Creación de cuentas	
Prioridad: Alta		**Riesgo en desarrollo:** Media
Puntos estimados: 3		**Sprint:** 1
Programador responsable: Roberto Prado		
Criterios de aceptación: 1.- Botón que permita ver el formulario de registro 2.- Formulario de registro 3.- Botones para confirmar o cancelar el registro 4.- Mensaje de confirmación el registro		

Tabla 32 - Historia de usuario 8

Historia de Usuario		
HU08	**Título:** Lista de cuentas	
Prioridad: Alta		**Riesgo en desarrollo:** Baja
Puntos estimados: 3		**Sprint:** 1
Programador responsable: Roberto Prado		
Criterios de aceptación: 1.- Tabla dinámica con la lista cuentas		

2.3.2.2. Sprint Backlog

Tabla 33 - Sprint Backlog 01

Cod	Prioridad	Estimación PH
HU03	Alta	3
HU04	Alta	3
HU05	Alta	3
HU06	Alta	7
HU07	Alta	3
HU08	Alta	3

2.3.2.3. Sprint Review

Las actividades realizadas en este sprint fueron:

- Se creó la interfaz de registro la cual presenta el formulario con los campos especificados, en esta pantalla el usuario se registra y es redirigido a la pantalla de login.
- Se creó la interfaz del login en donde el usuario ingresa su correo y su contraseña, si es válido es redirigido a la pantalla de inicio, sino se muestra un mensaje de fallo de autenticación.
- Se creó la pantalla principal en donde se muestran las opciones de navegación, el usuario podrá acceder a ellas de acuerdo al rol.
- Se creó la pantalla de gestión de cuentas, roles y usuarios en donde un administrador puede crear y listar cuentas.

Con estas actividades completas que cubren todos los criterios de aceptación el Sprint 1 se completó satisfactoriamente.

2.3.2.4. Sprint Retrospective

Al comienzo del Sprint se requirió de más tiempo debido a que se comenzaron a diseñar los prototipos de las interfaces que servirán de base para todo el proyecto. Una vez que el diseño está definido el desarrollo de proyecto se igualó al final del Sprint se puede ver que la tendencia del avance del proyecto se posicionó por encima del tiempo planeado. A continuación, se observa el tablero de avance del proyecto:

Tabla 34 - Tabla de avance del proyecto

Pendientes	En desarrollo	Finalizadas
HU09, HU10, HU11, HU12, HU13, HU14, HU15, HU16, HU17, HU18, HU19, HU20, HU21, HU22, HU23, HU24, HU25, HU26, HU27, HU28, HU29, HU30, HU31, HU32, HU33, HU34		HU01, HU02, HU03, HU04, HU05, HU06, HU07, HU08

Burndown Chart

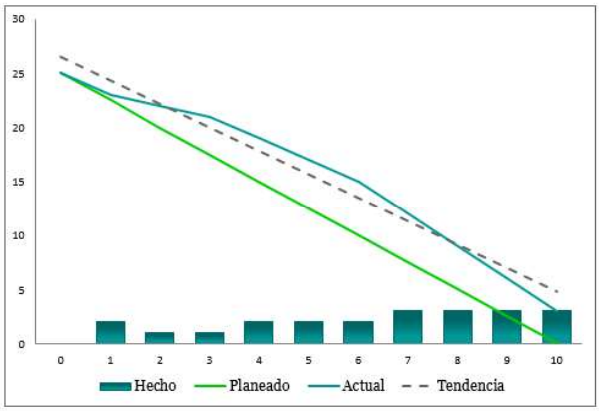

Figura 15 - Burndown Chart Sprint 1

2.3.3. Sprint 2

2.3.3.1. Sprint Planning

Objetivos del Sprint

- Desarrollar la gestión de cuentas, usuarios y acceso a datos
- Desarrollar la gestión de acceso a datos espaciales

Historias de Usuario para este Sprint

Tabla 35 - Historia de usuario 9

Historia de Usuario		
HU09	**Título:** Actualización y eliminación de cuentas	
Prioridad: Alta		**Riesgo en desarrollo:** Alta
Puntos estimados: 3		**Sprint:** 1
Programador responsable: Roberto Prado		
Criterios de aceptación: 1.- Botón que permita ver el formulario de actualización 2.- Formulario de actualización 3.- Botones para confirmar o cancelar la actualización. 4.- Mensaje de confirmación de la actualización 5.- Botón para eliminar una cuenta 6.- Ventana de confirmación para eliminar una cuenta 7.- Mensaje de confirmación a eliminar una cuenta		

Tabla 36 - Historia de usuario 10

Historia de Usuario		
HU10	**Título:** Lista de usuarios	
Prioridad: Alta		**Riesgo en desarrollo:** Alta
Puntos estimados: 1		**Sprint:** 1
Programador responsable: Roberto Prado		
Criterios de aceptación: 1.- Tabla dinámica con la lista de usuarios registrados		

Tabla 37 - Historia de usuario 11

Historia de Usuario	
HU11	**Título:** Actualización de usuario
Prioridad: Alta	**Riesgo en desarrollo:** Alta
Puntos estimados: 1	**Sprint:** 1
Programador responsable: Roberto Prado	
Criterios de aceptación: 1.- Botón que permita ver el formulario de actualización 2.- Formulario de actualización 3.- Botones para confirmar o cancelar la actualización. 4.- Mensaje de confirmación de la actualización	

Tabla 38 - Historia de usuario 12

Historia de Usuario	
HU12	**Título:** Eliminación de usuarios
Prioridad: Alta	**Riesgo en desarrollo:** Alta
Puntos estimados: 2	**Sprint:** 1
Programador responsable: Roberto Prado	
Criterios de aceptación: 1.- Botón para eliminar un usuario 2.- Ventana de confirmación para eliminar un usuario 3.- Mensaje de confirmación a eliminar un usuario	

Tabla 39 - Historia de usuario 13

Historia de Usuario	
HU13	**Título:** Creación de configuración para el accesos a datos localizados
Prioridad: Alta	**Riesgo en desarrollo:** Alta
Puntos estimados: 2	**Sprint:** 2

Programador responsable: Roberto Prado
Criterios de aceptación: 1.- Botón que permita ver el formulario de registro 2.- Formulario de registro 3.- Botones para confirmar o cancelar el registro 4.- Mensaje de confirmación el registro.

Tabla 40 - Historia de usuario 14

		Historia de Usuario
HU14	**Título:** Listar, actualización y eliminación de configuraciones de accesos a datos localizados	
Prioridad: Alta		**Riesgo en desarrollo:** Alta
Puntos estimados: 7		**Sprint:** 2
Programador responsable: Roberto Prado		
Criterios de aceptación: 1.- Tabla dinámica con la lista de configuraciones de acceso 2.- Botón que permita ver el formulario de actualización 3.- Formulario de actualización 4.- Botones para confirmar o cancelar la actualización. 5.- Mensaje de confirmación de la actualización 6.- Botón para eliminar una configuración 7.- Ventana de confirmación para eliminar una configuración 8.- Mensaje de confirmación a eliminar una configuración		

Tabla 41 - Historia de usuario 15

		Historia de Usuario
HU15	**Título:** Creación de opciones para la configuración de accesos a capas principales	
Prioridad: Alta		**Riesgo en desarrollo:** Alta
Puntos estimados: 2		**Sprint:** 2

Programador responsable: Roberto Prado
Criterios de aceptación: 1.- Botón que permita ver el formulario de registro 2.- Formulario de registro 3.- Botones para confirmar o cancelar el registro 4.- Mensaje de confirmación el registro

Tabla 42 - Historia de usuario 16

Historia de Usuario		
HU16	**Título:** Listar, actualización y eliminación de opciones para la configuración de accesos a capas principales	
Prioridad: Alta		**Riesgo en desarrollo:** Alta
Puntos estimados: 5		**Sprint:** 2
Programador responsable: Roberto Prado		
Criterios de aceptación: 1.- Tabla dinámica con la lista de configuraciones de acceso a capas principales 2.- Botón que permita ver el formulario de actualización 3.- Formulario de actualización 4.- Botones para confirmar o cancelar la actualización. 5.- Mensaje de confirmación de la actualización 6.- Botón para eliminar una configuración de acceso a una capa principal 7.- Ventana de confirmación para eliminar una configuración de acceso 8.- Mensaje de confirmación a eliminar una configuración de acceso a una capa principal		

2.3.3.2. Sprint Backlog

Tabla 43 - Sprint Backlog 02

Cod	Prioridad	Estimación PH
HU09	Alta	3
HU10	Alta	1
HU11	Alta	1
HU12	Alta	2
HU13	Alta	2

HU14	Alta	7
HU15	Alta	2
HU16	Alta	5

2.3.3.3. Sprint Review

Las actividades realizadas en este sprint fueron:

- Se creó la funcionalidad para actualizar y eliminar cuentas y roles por un administrador.
- Se creó la funcionalidad para actualizar y eliminar usuarios por un administrador.
- Se creó la interfaz de configuración de acceso por cuenta a los datos de la tabla con datos espaciales, en donde un administrador puede crear, listar, actualizar y eliminar la referencia a los datos de esta tabla.
- Se creó la interfaz de gestión de campos de búsqueda para una configuración de acceso, en esta interfaz el administrador podrá crear, listar, actualizar y eliminar campos de búsqueda.

Con estas actividades completas que cubren todos los criterios de aceptación el Sprint 2 se completó satisfactoriamente.

2.3.3.4. Sprint Retrospective

Debido a que ya se hicieron pantallas similares y servicios similares en el Sprint anterior el desarrollo del Sprint siguió una tendencia muy próxima al tiempo planeado. A continuación, se observa el tablero de avance del proyecto:

Tabla 44 - Tablero de avance del proyecto

Pendientes	**En desarrollo**	**Finalizadas**
HU17, HU18, HU19, HU20, HU21, HU22, HU23, HU24, HU25, HU26, HU27, HU28, HU29, HU30, HU31, HU32, HU33, HU34		HU01, HU02, HU03, HU04, HU05, HU06, HU07, HU08, HU09, HU10, HU11, HU12, HU13, HU14, HU15, HU16

Burndown Chart

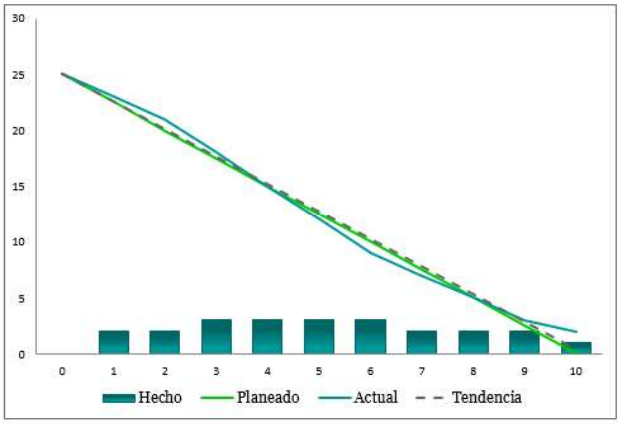

Figura 16 - Burndown Chart Sprint 2

2.3.4. Sprint 3

2.3.4.1. Sprint Planning

Objetivo del sprint

- Desarrollar la gestión de cuentas, usuarios y acceso a datos
- Desarrollar la gestión del perfil de usuario
- Desarrollar la gestión y visualización de mapas de mapas

Historias de Usuario para este Sprint

Tabla 45 - Historia de usuario 17

	Historia de Usuario
HU17	**Título:** Creación de configuración de acceso a capa secundaria
Prioridad: Alta	**Riesgo en desarrollo:** Media
Puntos estimados: 3	**Sprint:** 2
Programador responsable: Roberto Prado	
Criterios de aceptación: 1.- Botón que permita ver el formulario de registro 2.- Formulario de registro 3.- Botones para confirmar o cancelar el registro 4.- Mensaje de confirmación el registro	

Tabla 46 - Historia de usuario 18

Historia de Usuario		
HU18	**Título:** Lista, actualización y eliminación de configuraciones de accesos a capas secundarias	
Prioridad: Alta		**Riesgo en desarrollo:** Baja
Puntos estimados: 7		**Sprint:** 2
Programador responsable: Roberto Prado		
Criterios de aceptación: 1.- Tabla dinámica con la lista de configuraciones de accesos a capas secundarias 2.- Botón que permita ver el formulario de actualización 3.- Formulario de actualización 4.- Botones para confirmar o cancelar la actualización. 5.- Mensaje de confirmación de la actualización 6.- Botón para eliminar una configuración de acceso 7.- Ventana de confirmación para eliminar una configuración de acceso 8.- Mensaje de confirmación a eliminar una configuración de acceso		

Tabla 47 - Historia de usuario 19

Historia de Usuario		
HU19	**Título:** Actualización de datos de usuario	
Prioridad: Baja		**Riesgo en desarrollo:** Baja
Puntos estimados: 3		**Sprint:** 2
Programador responsable: Roberto Prado		
Criterios de aceptación: 1.- Un usuario puede ver su información personal, su rol y a la cuenta a la que está asociado. 2.- Un usuario puede abrir el formulario de actualización de datos 3- EL formulario debe tener las opciones de cancelar o guardar los datos 4.- Se muestra una notificación al momento de guardar los datos		

Tabla 48 - Historia de usuario 20

Historia de Usuario		
HU20	**Título:** Creación de información de mapa	
Prioridad: Alta		**Riesgo en desarrollo:** Media
Puntos estimados: 3		**Sprint:** 3
Programador responsable: Roberto Prado		
Criterios de aceptación: 1.- Botón que permita ver el formulario de registro 2.- Formulario de registro 3.- Botones para confirmar o cancelar el registro 4.- Mensaje de confirmación el registro		

Tabla 49 - Historia de usuario 21

Historia de Usuario		
HU21	**Título:** Lista, actualización y eliminación de mapas creados	
Prioridad: Alta		**Riesgo en desarrollo:** Baja
Puntos estimados: 5		**Sprint:** 3
Programador responsable: Roberto Prado		
Criterios de aceptación: 1.- Tabla dinámica con la lista de mapas creados 2.- Botón que permita ver el formulario de actualización 3.- Formulario de actualización 4.- Botones para confirmar o cancelar la actualización. 5.- Mensaje de confirmación de la actualización 6.- Botón para eliminar un mapa 7.- Ventana de confirmación para eliminar un mapa 8.- Mensaje de confirmación a eliminar un mapa		

Tabla 50 - Historia de usuario 22

Historia de Usuario	
HU22	**Título:** Creación de la interfaz para visualizar un mapa en pantalla completa
Prioridad: Alta	**Riesgo en desarrollo:** Media
Puntos estimados: 3	**Sprint:** 3
Programador responsable: Roberto Prado	
Criterios de aceptación: 1.- Pantalla en donde se muestra un mapa en pantalla completa 2.- El mapa debe estar ubicado en la ubicación actual del usuario	

2.3.4.2. Sprint Backlog

Tabla 51 - Sprint Backlog 3

Cod	Prioridad	Estimación PH
HU17	Alta	3
HU18	Alta	7
HU19	Baja	3
HU20	Alta	3
HU21	Alta	5
HU22	Alta	3

2.3.4.3. Sprint Review

Las actividades realizadas en este sprint fueron:

- Se creó la interfaz de configuración de acceso por cuenta a los datos de la tabla con datos de análisis, en donde un administrador puede crear, listar, actualizar y eliminar la referencia a los datos de esta tabla.
- Se creó la interfaz en donde el usuario pude ver su información personal, su rol y su cuenta asociada, el usuario tiene la opción de editar sus datos personales al abrir un formulario de actualización de datos.
- Se creó la interfaz en donde el usuario puede crear, listar, actualizar y eliminar mapas.
- Se creó el botón para abrir un mapa en cada ítem de la lista de mapas.
- Se creó la interfaz de mapa en pantalla completa.

Con estas actividades completas que cubren todos los criterios de aceptación el Sprint 3 se completó satisfactoriamente.

2.3.4.4. Sprint Retrospective

Debido a que ya se hicieron pantallas similares y servicios similares en el Sprint anterior para la creación de mapas, las historias iniciales tomaron menos tiempo de lo planificado, al final del Sprint las historias tomaron más tiempo de lo planificado debido a que el componente web para crear el menú proporcionado en Angular Material no se adaptó al funcionamiento requerido por lo que se optó por diseñar un menú completamente. A continuación, se observa el tablero de avance del proyecto:

Tabla 52 - Tablero de avance de proyecto

Pendientes	En desarrollo	Finalizadas
HU23, HU24, HU25, HU26, HU27, HU28, HU29, HU30, HU31, HU32, HU33, HU34		HU01, HU02, HU03, HU04, HU05, HU06, HU07, HU08, HU09, HU10, HU11, HU12, HU13, HU14, HU15, HU16, HU17, HU18, HU19, HU20, HU21, HU22

Burndown Chart

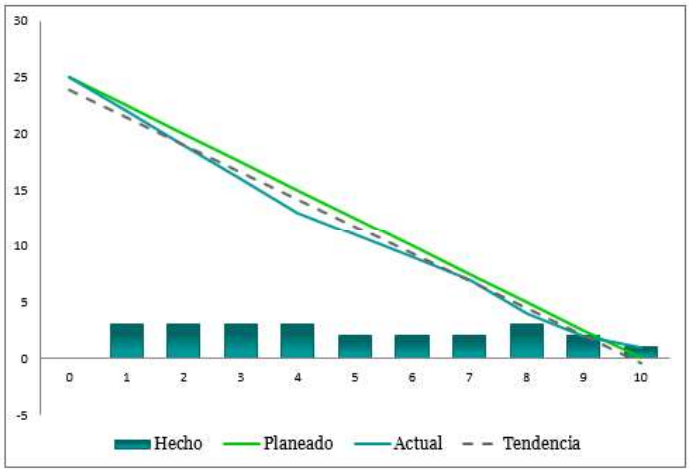

Figura 17 - Burndown Chart Sprint 3

2.3.5. Sprint 4

2.3.5.1. Sprint Planning

Objetivo del sprint

- Desarrollar la gestión y visualización de mapas de mapas
- Desarrollar la gestión de capas principales y secundarias

Historias de Usuario para este Sprint

Tabla 53 - Historia de usuario 23

		Historia de Usuario
HU23	**Título:** Creación de controles de navegación y menú en el mapa	
Prioridad: Media		**Riesgo en desarrollo:** Media
Puntos estimados: 7		**Sprint:** 3
Programador responsable: Roberto Prado		
Criterios de aceptación: 1.- Controles de acercamiento 2.- Navegación a la pantalla de mapas 3.- Abrir y cerrar menú 4.- Barra deslizadora. 5.- Menú en donde se visualizan las capas creadas 6.- Checkbox de activación de capas 7.- Botones de acción a cada capa: Eliminación, Configuración		

Tabla 54 - Historia de usuario 24

		Historia de Usuario
HU24	**Título:** Creación de capas principales	
Prioridad: Alta		**Riesgo en desarrollo:** Alta
Puntos estimados: 10		**Sprint:** 4
Programador responsable: Roberto Prado		
Criterios de aceptación: 1.- Botón en el menú para crear una nueva capa		

2.- Formulario para crear una capa 3.- Selección de colores para los marcadores de la capa

Tabla 55 - Historia de usuario 25

Historia de Usuario		
HU25	**Título:** Lista de capas principales en el menú del mapa	
Prioridad: Alta		**Riesgo en desarrollo:** Alta
Puntos estimados: 4		**Sprint:** 4
Programador responsable: Roberto Prado		
Criterios de aceptación: 1.- Lista desplegable con un checkbox para cada capa principal. 2.- Activación y desactivación de capas por medio del checkbox		

Tabla 56 - Historia de usuario 26

Historia de Usuario		
HU26	**Título:** Actualización y configuración de capas principales	
Prioridad: Alta		**Riesgo en desarrollo:** Alta
Puntos estimados: 4		**Sprint:** 4
Programador responsable: Roberto Prado		
Criterios de aceptación: 1.- Botón en una capa del menú que permita abrir el formulario de actualización 2.- Formulario de actualización 3.- Botones para confirmar o cancelar la actualización. 4.- Mensaje de confirmación de la actualización 5.- Botón en una capa del menú que permita abrir el formulario de configuración 6.- Formulario de configuración		

2.3.5.2. Sprint Backlog

Tabla 57 - Sprint Backlog 4

Cod	Prioridad	Estimación PH
HU23	Media	7
HU24	Alta	10
HU25	Alta	4
HU26	Alta	4

2.3.5.3. Sprint Review

Las actividades realizadas en este sprint fueron:

- En la interfaz del mapa se creó los botones con los cuales el usuario puede mover el mapa, realizar acercamientos y alejamientos, un botón para regresar a la lista y un botón para abrir y cerrar un panel lateral que es donde se mostrarán la lista de capas y sus controles de activación y configuración
- Se creó la opción de crear capas principales, se optó por un diseño del formulario enfocado en 4 pasos para crear el formulario para que sea más fácil al usuario crear capas, primero el usuario selecciona el nombre de la capa, selecciona el origen de datos, la configuración de colores de los marcadores y el botón de acción para crear una capa.
- Se creó la lista desplegable que muestra las capas que se van creando con el formulario, a la izquierda se encuentra el checkbox de activación, a la derecha se encuentran las opciones de eliminación, configuración de la capa y el filtro para la capa de acuerdo a las opciones de la capa creada.
- El usuario puede editar la capa, esto es, cambiar el nombre de la capa y los colores de los marcadores.
- El usuario puede filtrar la capa por medio de una interfaz en donde se muestran las opciones de filtrado.

Con estas actividades completas que cubren todos los criterios de aceptación el Sprint 4 se completó satisfactoriamente.

2.3.5.4. Sprint Retrospective

Al inicio del Sprint se requirió un poco más de tiempo debido a que se optaron por varios diseños para la interfaz de creación de capas, se crearon interfaces adicionales para la selección de colores, otro aspecto que impacto este inicio fue el majeo de los servicios

asíncronos de firebase que es donde se guarda la configuración de las capas para cada usuario. A continuación, se observa el tablero de avance del proyecto:

Tabla 58 - Tablero de avance del proyecto

Pendientes	En desarrollo	Finalizadas
HU27, HU28, HU29, HU30, HU31, HU32, HU33, HU34		HU01, HU02, HU03, HU04, HU05, HU06, HU07, HU08, HU09, HU10, HU11, HU12, HU13, HU14, HU15, HU16, HU17, HU18, HU19, HU20, HU21, HU22, HU23, HU24, HU25, HU26

Burndown Chart

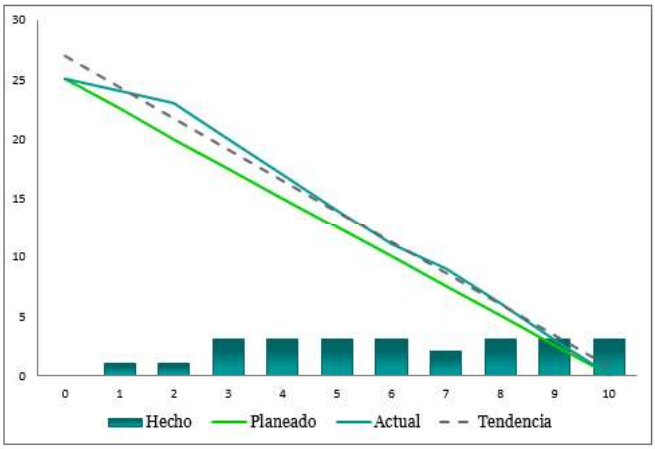

Figura 18 - Burndown Chart Sprint 4

2.3.6. Sprint 5

2.3.6.1. Sprint Planning

Objetivo del sprint

- Desarrollar la gestión de capas principales y secundarias

Historias de Usuario para este Sprint

Tabla 59 - Historia de usuario 27

Historia de Usuario		
HU27	**Título:** Eliminación de capas principales	
Prioridad: Alta		**Riesgo en desarrollo:** Media

Puntos estimados: 5	**Sprint:** 4
Programador responsable: Roberto Prado	
Criterios de aceptación: 1.- Botón para eliminar una capa 2.- Ventana de confirmación para eliminar una capa 3.- Mensaje de confirmación a eliminar una capa	

Tabla 60 - Historia de usuario 28

Historia de Usuario		
HU28	**Título:** Creación de capa secundaria	
Prioridad: Alta		**Riesgo en desarrollo:** Alta
Puntos estimados: 10		**Sprint:** 5
Programador responsable: Roberto Prado		
Criterios de aceptación: 1.- Botón en el menú para crear una nueva capa 2.- Formulario para crear una capa 3.- Configuración de visualización por cada tipo de capa		

Tabla 61 - Historia de usuario 29

Historia de Usuario		
HU29	**Título:** Lista de capas secundarias dentro de una capa principal en el menú	
Prioridad: Alta		**Riesgo en desarrollo:** Alta
Puntos estimados: 5		**Sprint:** 5
Programador responsable: Roberto Prado		
Criterios de aceptación: 1.- Lista con un checkbox de capas secundarias. 2.- Activación y desactivación de capas por medio del checkbox		

Tabla 62 - Historia de usuario 30

Historia de Usuario		
HU30	**Título:** Actualización y configuración de una capa secundaria	
Prioridad: Alta		**Riesgo en desarrollo:** Alta
Puntos estimados: 5		**Sprint:** 5
Programador responsable: Roberto Prado		
Criterios de aceptación: 1.- Botón en una capa del menú que permita abrir el formulario de actualización 2.- Formulario de actualización 3.- Botones para confirmar o cancelar la actualización. 4.- Mensaje de confirmación de la actualización		

2.3.6.2. Sprint Backlog

Tabla 63 - Sprint Backlog 5

Cod	Prioridad	Estimación PH
HU27	Alta	5
HU28	Alta	10
HU29	Alta	5
HU30	Alta	5

2.3.6.3. Sprint Review

Las actividades realizadas en este sprint fueron:

- El usuario puede eliminar una capa principal, al hacerlo previamente se presentará una ventana de confirmación.
- Se creó la opción de crear capas secundarias que se muestran dentro de un ítem de las capas principales en forma de botón, al dar clic sobre botón se abre un formulario, se optó por un diseño del formulario enfocado en 4 pasos para crear el formulario para que sea más fácil al usuario crear capas, primero el usuario selecciona el nombre de la capa, selecciona el origen de datos, la configuración de visualización de la capa que varía dependiendo del tipo de capa a crear y el botón de acción para crear una capa.
- Se creó la lista que muestra las capas que se van creando con el formulario, a la izquierda se encuentra el checkbox de activación, a la derecha se encuentran las opciones de eliminación y edición.

- El usuario puede editar la capa, esto es, cambiar el nombre de la capa y la configuración de la capa, la configuración varía de acuerdo al tipo de capa.

Con estas actividades completas que cubren todos los criterios de aceptación el Sprint 5 se completó satisfactoriamente.

2.3.6.4. Sprint Retrospective

Al inicio del Sprint las tareas se desarrollaron según lo previsto debido a que ya se tenía un diseño del formulario del Sprint anterior. Al finalizar el Sprint las historias se demoraron debido a que las librerías para crear graficas no eran compatibles con la versión de la librería que crea el mapa. A continuación, se observa el tablero de avance del proyecto:

Tabla 64 - Tablero de avance

Pendientes	En desarrollo	Finalizadas
HU31, HU32, HU33, HU34		HU01, HU02, HU03, HU04, HU05, HU06, HU07, HU08, HU09, HU10, HU11, HU12, HU13, HU14, HU15, HU16, HU17, HU18, HU19, HU20, HU21, HU22, HU23, HU24, HU25, HU26, HU27, HU28, HU29, HU30

Burndown Chart

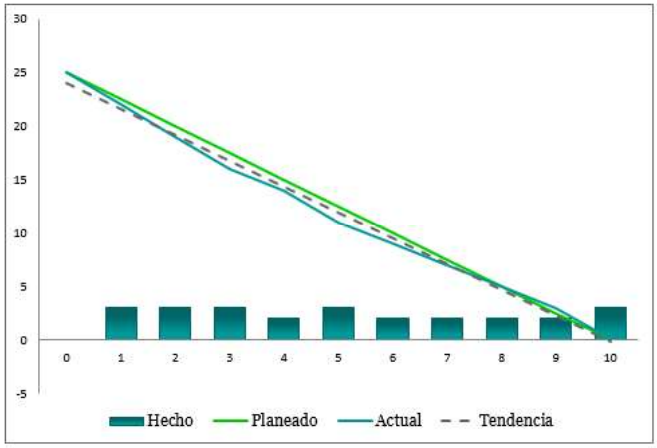

Figura 19 - Burndown Chart Sprint 5

2.3.7. Sprint 6

2.3.7.1. Modificación del Product Backlog

Se decidió aumentar en el producto el módulo de reportes por zona y la implementación del sistema en la nube de Google. Además, se definió que el módulo de conteo por zona muestre el promedio y la suma de los valores analizados. Se redefinió la historia de usuario épica correspondiente a este Sprint.

Tabla 65 - Historia de usuario épica 6 actualizada

	Historia de Usuario Épica
HUE09	**Usuario:** Usuario general
Nombre historia: Desarrollar las herramientas de apoyo	
Descripción: Como usuario quiero usar las herramientas de búsqueda de direcciones, isócronas, y conteo, suma, promedio de las capas de análisis activas por cada zona de la isócrona, el sistema debe ser capaz de implementarse en la nube.	

Una vez definidos estos cambios el Producto Backlog fue el siguiente:

Tabla 66 - Product backlog actualizado

Product Backlog				
HUE	**COD**	**Historia de usuario**	**Prioridad**	**PH**
HUE05	HU31	Eliminación de una capa secundaria	Alta	3
HUE06	HU32	Creación de la funcionalidad de búsqueda de direcciones y leyendas	Media	3
	HU33	Creación de isócronas	Alta	5
	HU34	Creación de la funcionalidad de conteo, suma y promedio por área	Alta	5
	HU35	Creación de la pantalla de reportes	Alta	5
	HU36	Implementación del sistema en la nube	Baja	3

2.3.7.2. Sprint Planning

Objetivo del sprint

- Desarrollar las herramientas de apoyo

Historias de Usuario para este Sprint

Tabla 67 - Historia de usuario 31

	Historia de Usuario
HU31	**Título:** Eliminación de una capa secundaria
Prioridad: Alta	**Riesgo en desarrollo:** Alta

Puntos estimados: 3	**Sprint:** 5
Programador responsable: Roberto Prado	
Criterios de aceptación: 1.- Botón para eliminar una capa 2.- Ventana de confirmación para eliminar una capa 3.- Mensaje de confirmación a eliminar una capa	

Tabla 68 - Historia de usuario 32

Historia de Usuario	
HU32	**Título:** Creación de la funcionalidad de búsqueda de direcciones y leyendas
Prioridad: Media	**Riesgo en desarrollo:** Baja
Puntos estimados: 3	**Sprint:** 6
Programador responsable: Roberto Prado	
Criterios de aceptación: 1.- Agregar campo para ingreso de dirección. 2.- Selección de dirección de acuerdo a sugerencias más similares 3.- Ventana que muestra la leyenda del mapa mostrando solo los elementos activos	

Tabla 69 - Historia de usuario 33

Historia de Usuario	
HU33	**Título:** Creación de isócronas
Prioridad: Alta	**Riesgo en desarrollo:** Baja
Puntos estimados: 5	**Sprint:** 6
Programador responsable: Roberto Prado	
Criterios de aceptación: 1.- Botón para abrir la ventana de creación de isócronas	

2.- Ventana para crear una isócrona 3.- Menú donde se muestren las isócronas creadas. 4.- Configuración de la isócrona.

Tabla 70 - Historia de usuario 34

Historia de Usuario		
HU34	**Título:** Creación de la funcionalidad de conteo, suma y promedio por área	
Prioridad: Alta		**Riesgo en desarrollo:** Alta
Puntos estimados: 5		**Sprint:** 6
Programador responsable: Roberto Prado		
Criterios de aceptación: 1.- Cuando se da clic en un área de una isócrona se debe abrir una pantalla que muestre los conteos, sumas y promedios de los valores de las capa activas.		

Tabla 71 - Historia de usuario 35

Historia de Usuario		
HU35	**Título:** Creación de la pantalla de reportes	
Prioridad: Alta		**Riesgo en desarrollo:** Alta
Puntos estimados: 5		**Sprint:** 6
Programador responsable: Roberto Prado		
Criterios de aceptación: 1.- Desde la pantalla de resumen por zona, al dar clic en un botón se debe abrir una pantalla con los datos usados para crear la pantalla de resumen en una tabla. 2.- Se debe tener la opción para seleccionar varias áreas. 3.- El reporte debe ser descargado en un pdf.		

Tabla 72 - Historia de usuario 36

Historia de Usuario	
HU36	**Título:** Implementación del sistema en la nube
Prioridad: Baja	**Riesgo en desarrollo:** Baja
Puntos estimados: 1	**Sprint:** 6
Programador responsable: Roberto Prado	
Criterios de aceptación: 1.- Como meta final para el sistema creado, se debe configurar el mismo a fin de que pueda ser implementado en la nube por medio del Servicio App Engine de Google	

2.3.7.3. Sprint Backlog

Tabla 73 - Sprint backlog 6

Cod	Prioridad	Estimación PH
HU31	Alta	3
HU32	Media	3
HU33	Alta	5
HU34	Alta	5
HU35	Alta	5
HU36	Baja	1

2.3.7.4. Sprint Review

Las actividades realizadas en este sprint fueron:

- El usuario puede eliminar una capa secundaria, al hacerlo previamente se presentará una ventana de confirmación.
- Se creó el botón para activar la funcionalidad de isócronas.
- Cuando el usuario activa la funcionalidad de isócronas aparece en la parte superior una instrucción para crear isócronas.
- Cuando el usuario da clic sobre el mapa se abre el formulario para crear una isócrona, cuando se crea la isócrona se muestra en la lista y se activa la capa en la sección de capas adicionales.
- El usuario puedo eliminar la capa de isócronas o editar su configuración y puede activar o desactivar la capa.

- La isócrona está dividida por tiempo y muestra la información de la población, área y tiempo.
- Al dar clic sobre una isócrona, se abre una pantalla en donde se muestra un resumen de los datos de análisis dentro de esa zona.
- Al dar clic sobre el botón “Ver más” dentro de la pantalla de resumen se abre un reporte detallado de los datos usados para crear el resumen, se tiene la opción para descargar el reporte en un archivo pdf.
- Por medio del servicio de Google “App Engine” se implementó tanto el back-end como el front-end del sistema por medio de archivos de implementación, ver anexo I, se puede acceder al sistema a través de https://tesis-frontend.appspot.com.

2.3.7.5. Sprint Retrospective

La creación de isócronas sigue el modelo de representación de capas ya implementado, los datos se obtienen del servicio de “Open route layers”. Debido a la complejidad para configurar el servicio y los estilos de la capa el proyecto tomo más tiempo en esta parte del sprint, una vez finalizada la etapa de creación de isócronas, se procedió a crear la interfaz de resumen por zona, debido a que ya se habían creado los servicios para la obtención de datos se los reutilizo, para la creación del interfaz de reportes fue necesario reestructura los métodos para la obtención de datos por medio servicios. Además, fue necesario estructuras nuevas consultas a la base de datos que permitan entregar los resultados de los reportes, para la implementación del sistema en Google debido a que ya se tenía experiencia en este tipo de tareas se lo ejecuto rápidamente permitiendo completar el Sprint de acuerdo al esfuerzo planeado.

A continuación, se observa el tablero de avance del proyecto:

Tabla 74 - Tablero de avance del proyecto

Pendientes	En desarrollo	Finalizadas
		HU01, HU02, HU03, HU04, HU05, HU06, HU07, HU08, HU09, HU10, HU11, HU12, HU13, HU14, HU15, HU16, HU17, HU18, HU19, HU20, HU21, HU22, HU23, HU24, HU25, HU26, HU27, HU28, HU29, HU30, HU31, HU32, HU33, HU34, HU35, HU36

Burndown Chart

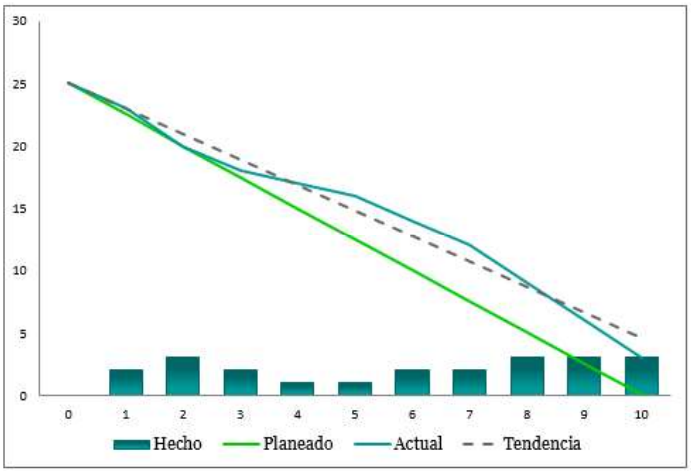

Figura 20 Burndown Chart Sprint 6

2.4. Pruebas funcionales

Las pruebas funcionales permiten determinar que un sistema cumple con los requisitos especificados, demostrar que son aptos para el propósito por el cual fueron diseñados y para detectar errores [43]. La técnica de caja negra permite realizar este tipo de pruebas ya que se basan en la especificación funcional como no funcional si tener en cuenta los procesos internos del sistema.

Plan de pruebas

Para esta sección se creará un plan de pruebas en base a los criterios de aceptación referentes a cada historia épica, creando casos de prueba a partir de estos criterios, ver anexo F:

Tabla 75 - Plan de pruebas de acuerdo a criterios de aceptación

COD	Módulos	Pruebas
CP1	Acceso a la pantalla principal	Los formularios de login y registro deben validar campos: email, campos requeridos. Mostrar mensajes de información en el proceso de login y registro.
CP2	Gestión de cuentas, roles y usuarios.	Solo el administrador debe acceder. Se debe tener las opciones para abrir los formularios de registro, actualización y eliminación de datos. Los formularios deben validarse antes de ser enviados.

	Desarrollar la gestión de acceso a datos	
CP3	Gestión del perfil de usuario	Un administrador y usuario tienen acceso. Se debe mostrar la información personal, roles y cuenta. El formulario de actualización debe estar validado.
CP4	Gestión de mapas Interfaz para la visualización y personalización de mapas	Solo un usuario tiene acceso. Se debe tener las opciones para abrir los formularios de registro, actualización y eliminación de datos. Los formularios deben validarse antes de ser enviados. Cada ítem de la lista debe tener un botón de acción para abrir un mapa en pantalla completa. Se debe tener control en el mapa, esto es: navegación, acercamientos, opción para regresar a la lista de mapas.
CP5	Gestión de capas principales Gestión de capas secundarias	Los formularios de creación, actualización y eliminación deben validarse antes de ser enviados. La lista de capas principales se muestra en el menú lateral izquierdo del mapa. La lista de capas secundarias se muestra dentro de cada ítem de las capas principales. Al activar una capa se muestran marcadores o graficas en el mapa de acuerdo a sus coordenadas y su configuración. Al desactivar o eliminar una capa se eliminan los marcadores o graficas del mapa. Los marcadores o graficas deben actualizarse automáticamente cuando se actualiza o filtra una capa. Si existe al menos una capa secundaria el mapa de mostrar una barra deslizante de los últimos 12 meses.
CP6	Herramientas de apoyo	Para la búsqueda de direcciones se debe tener la opción de autocompletado. Las isócronas deben mostrar la información de área, población, tiempo, suma y promedio de las capas de análisis activas.

		La pantalla de reporte muestra en una tabla la información de las capas activas y la opción de descarga del reporte en formato PDF.

Casos de pruebas

Tabla 76 - Caso de prueba 1

No.	CP1
Titulo	Acceso a la pantalla principal
Precondiciones	Los Servidores web deben estar activos. Acceso a internet. Navegador web con versiones más recientes.
Acciones	**Resultados**
Los formularios de login y registro deben validar campos: email, campos requeridos.	Se validan los campos por email y por campos requeridos. Se muestra mensajes cuando existen errores.
El botón de acción debe bloquearse cuando un formulario no es válido.	El botón se bloque cuando existen errores o cuando el formulario está vacío.

Capturas al desarrollar el caso de prueba 1

Al iniciar la aplicación se observa la pantalla de login. Ver Figura 19.

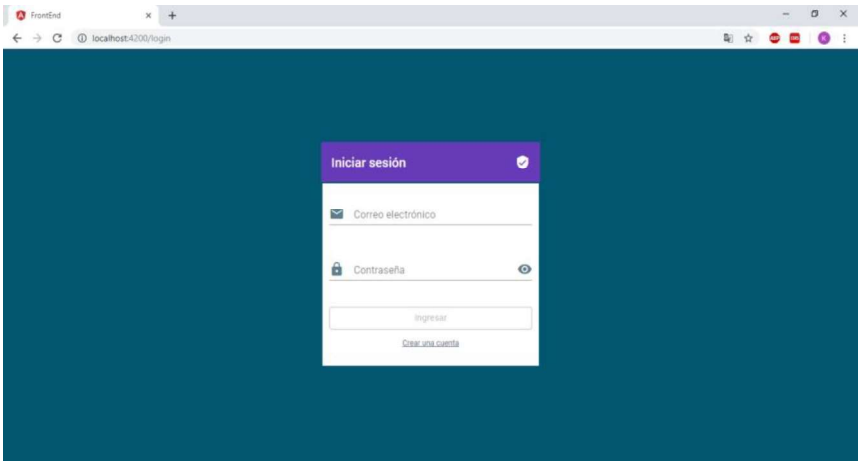

Figura 21 - Interfaz de Login

Se da clic en la opción “Crear cuenta” para ver la pantalla de registro. Ver Figura 20.

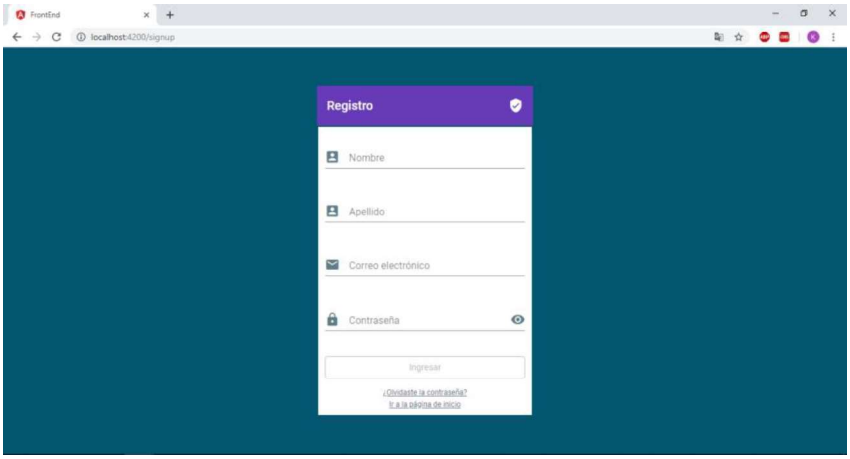

Figura 22 - Interfaz de Registro

Los campos en ambos formularios están validados y el botón permanece bloqueado mientras el formulario sea inválido. Cuando se crea un usuario se redirige a la pantalla de inicio y se muestra una notificación. La pantalla muestra la interfaz para un nuevo usuario. Ver Figura 21.

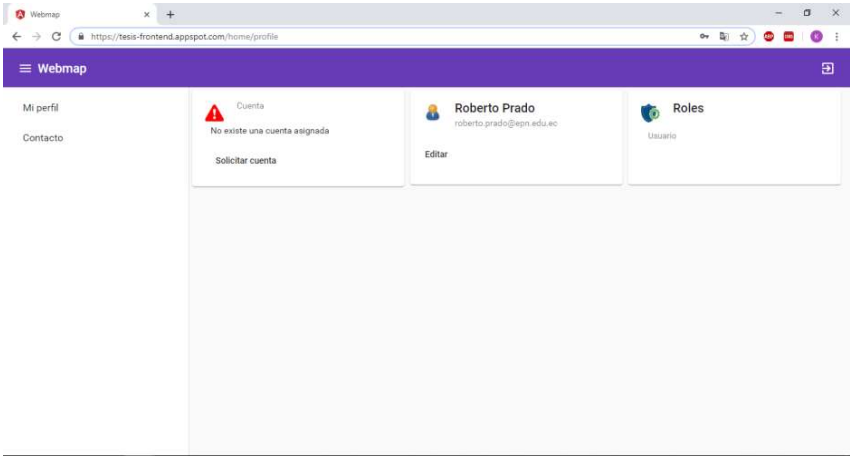

Figura 23 - Interfaz de Inicio

Tabla 77 - Caso de prueba 2

No.	CP2
Titulo	Gestión de cuentas, roles y usuarios. Desarrollar la gestión de acceso a datos.
Precondiciones	Ingresar como administrador
Acciones	**Resultados**
Solo el administrador debe acceder.	Aparecen solo las opciones de navegación de un administrador.

Se debe tener las opciones para abrir los formularios de registro, actualización y eliminación de datos.	El botón de registro se encuentra en la parte superior izquierda de cada tabla, las opciones de actualización y eliminación se encuentran en la parte derecha de cada ítem.
Los formularios deben validarse antes de ser enviados.	Los formularios se validan por campos requeridos, longitud máxima.

Capturas al desarrollar el caso de prueba 2

Pantalla de inicio con el rol de Administrador en donde se muestra un resumen de las cuentas roles y usuarios. Ver Figura 22.

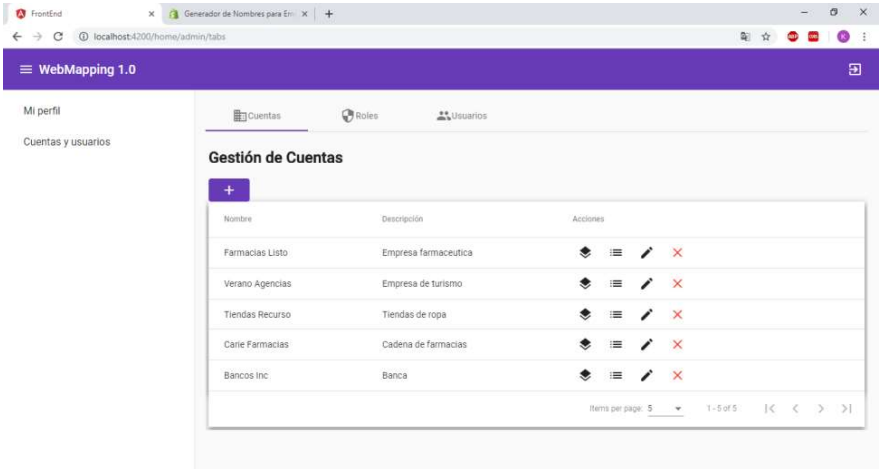

Figura 24 - Interfaz de Gestión de cuentas, roles y usuarios

En cada pantalla se muestra las opciones de registro, actualización y eliminación. Cuando se da clic sobre las opciones de registro y edición se abren en forma de pantallas modales los formularios respectivos, los formularios están validados, y el botón de registro se bloque cuando el formulario es inválido. Para la gestión de acceso a datos, en cada cuenta existen dos opciones, “Editar capas principales” y “Editar campos de análisis”, la primera opción permite definir a que datos de la tabla de datos localizados tendrá acceso la cuenta, el tipo de dato que se está consultando y el nombre del campo. Ver Figura 23.

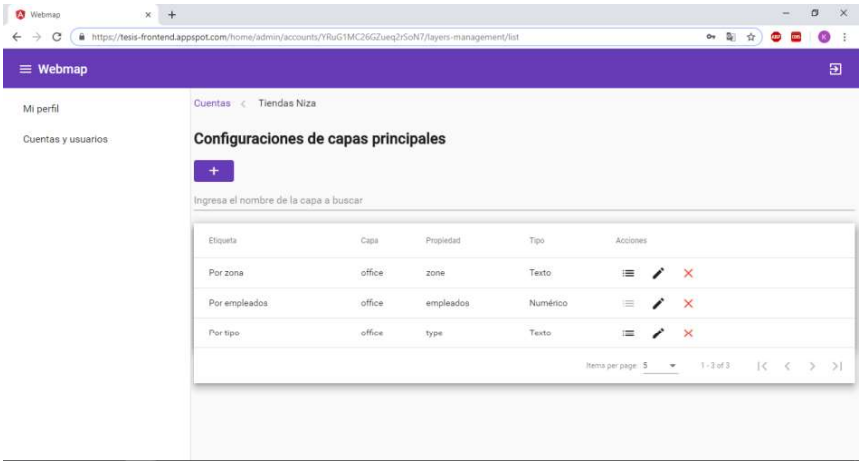

Figura 25 - Gestión de accesos a datos

Para la opción de “Editar campos de análisis” se muestra una tabla similar a la anterior, pero esta tabla permite definir los campos de análisis que aparecerán en la opción de agregar capas secundarias a un mapa. Ver Figura 24.

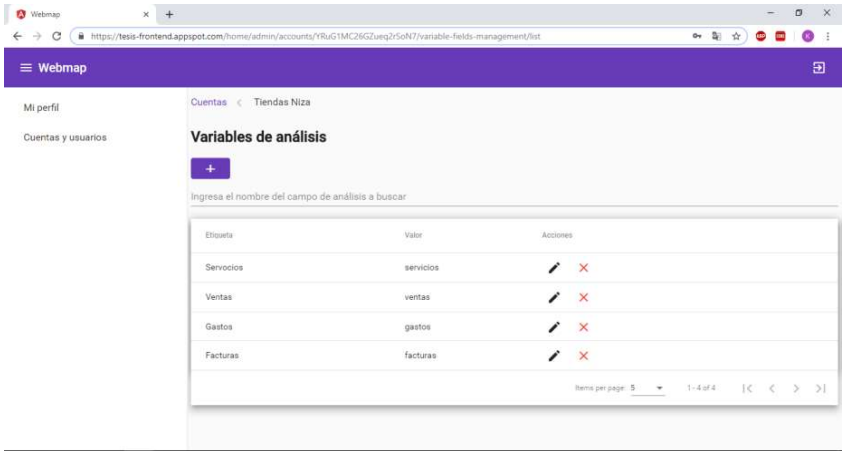

Figura 26 - Gestión de capas de análisis

Tabla 78 - Caso de prueba 3

No.	CP3
Titulo	Gestión del perfil de usuario
Precondiciones	Ingresar como usuario general
Acciones	**Resultados**
Un usuario tiene acceso.	Un usuario puede ver la pantalla de “perfil de usuario”.
Se debe mostrar la información personal, roles y cuenta.	Se observa los nombres, el email registrado, los roles y la cuenta asociada.
El formulario de actualización debe estar validado.	El formulario se valida por campos requeridos y longitud.

El formulario para solicitar cuenta debe esta validado	El formulario se valida por campos requeridos y longitud.
Se debe enviar correos al administrador cuando se solicita una cuenta y al usuario cuando se le ha activado una cuenta	Los correos deben especificar el requerimiento o notificación, y el origen.

Capturas al desarrollar el caso de prueba 3

Cuando se ingresa como un usuario normal se muestra el enlace a la página del perfil de usuario, en esta pantalla se observa los datos de usuarios, el rol y la cuenta a la que está asociado. Ver Figura 25.

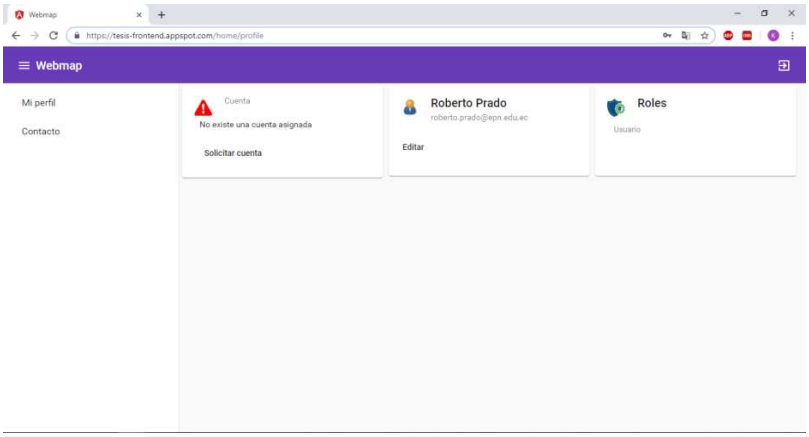

Figura 27 - Pantalla de inicio para una usuario

Para actualizar los datos personales se da clic en el botón “Editar”, se abre un formulario en donde se muestran los datos personales del usuario. Para solicitar una cuenta se abre al siguiente formulario en donde el usuario especificará la cuenta a la que desea acceder. Ver Figura 26.

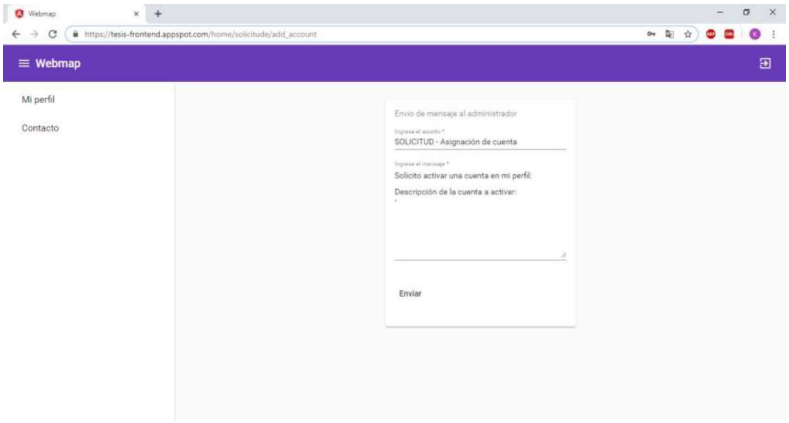

Figura 28 - Solicitud de acceso a cuenta

Tabla 79 - Caso de prueba 4

No.	CP4
Titulo	Gestión de mapas, Interfaz para la visualización y personalización de mapas
Precondiciones	Ingresar como usuario general
Acciones	**Resultados**
Solo un usuario tiene acceso.	Se accedió como un usuario normal y se accedió a la pantalla “Mis mapas”, un administrador no tiene acceso a esta pantalla.
Se debe tener las opciones para abrir los formularios de registro, actualización y eliminación de datos.	El botón de registro se encuentra en la parte superior izquierda de la tabla, las opciones de actualización y eliminación se encuentran en la parte derecha de cada ítem.
Los formularios deben validarse antes de ser enviados.	Los formularios se valida por campos requeridos y longitud.
Cada ítem de la lista debe tener un botón de acción para abrir un mapa en pantalla completa.	Se observa un icono de capas que abre las pantallas solicitada.
Se debe tener control en el mapa, esto es: navegación, acercamientos, opción para regresar a la lista de mapas.	Los controles de navegación se encuentran en la parte superior derecha, permiten desplazarse y ubicarse en el mapa de una manera sencilla. Los botones de navegación e información están en la parte inferior derecha, se presentan en forma de texto para especificar de forma más clara su acción.
Se debe mostrar botón que permita abrir un menú lateral.	El botón aparece cuando se cierra el menú lateral, al dar clic sobre el botón el menú se desliza nuevamente y se muestra en la parte izquierda de la pantalla.

Capturas al desarrollar el caso de prueba 4

Cuando se ingresa como un usuario normal se muestra el enlace a la página “Mis mapas”, se observa la lista de mapas creados por el usuario y los botones de acción para crear, actualizar, eliminar y abrir la pantalla de mapa en cada ítem de la lista. Ver Figura 27.

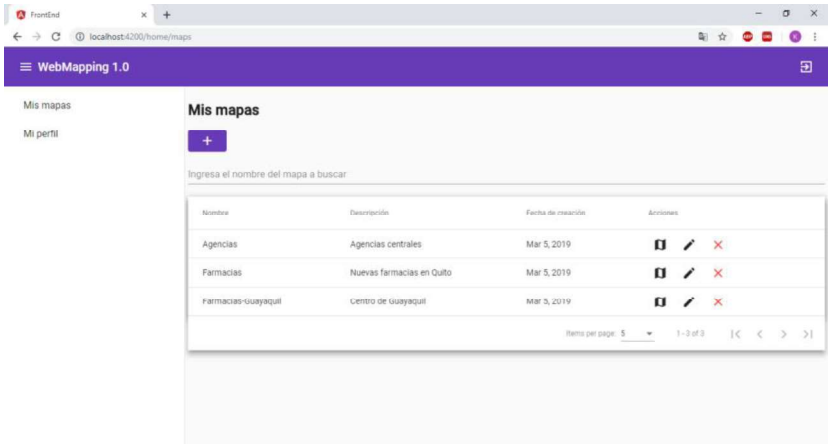

Figura 29 - Interfaz de mapas de usuario

Un mapa tiene un nombre, descripción y fecha de creación, la tabla permite buscar mapas por cualquiera de estos campos. Para crear o actualizar un mapa se da clic en el botón de acción y se abrirá el formulario de registro, los formularios se valida por campos requeridos o longitud. Al ingresar a través de la lista de mapas se observa un mapa el cual presenta los controles de navegación. Ver Figura 28.

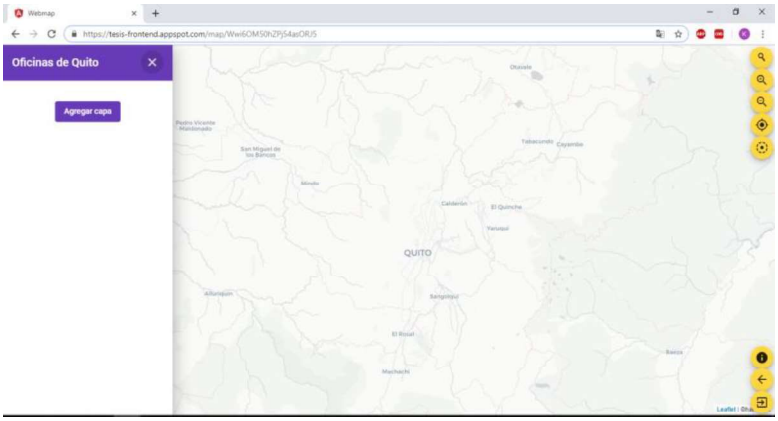

Figura 30 - Interfaz de mapa principal

Tabla 80 - Caso de prueba 5

No.	CP5
Titulo	Gestión de capas principales Gestión de capas secundarias

Precondiciones	Haber creado un mapa en la pantalla “Mis mapas”
Acciones	**Resultados**
Los formularios de creación, actualización y eliminación deben validarse antes de ser enviados.	Los formularios se valida por campos requeridos y longitud al guardar y editar capas.
La lista de capas principales se muestra en el menú lateral izquierdo del mapa.	La lista se va actualizando de forma dinámica a medida que se crea y elimina capas principales.
La lista de capas secundarias se muestra dentro de cada ítem de las capas principales.	La lista se va actualizando de forma dinámica a medida que se crea y elimina capas secundarias.
Al activar una capa se muestran marcadores o graficas en el mapa de acuerdo a sus coordenadas y su configuración.	Se crean marcadores de acuerdo a la configuración dada al momento de crear una capa.
Al desactivar o eliminar una capa se eliminan los marcadores o graficas del mapa.	Las capas se eliminan cuando se desactivan en el menú o se eliminan.
Los marcadores o graficas deben actualizarse automáticamente cuando se actualiza o filtra una capa.	Las capas se actualizan cuando se edita una capa, se la filtra, si una capa principal es filtrada este cambio se replica a las capas secundarias.
Si existe al menos una capa secundaria el mapa de mostrar una barra deslizante de los últimos 12 meses.	El control de tiempo muestra los últimos 12 meses contado desde un mes previo al actual. El control aparece automáticamente cuando se crea al menos una capa secundaria.
Las capas secundarias deben actualizarse a medida que la barra deslizante cambia su estado.	Las capas secundarias que están ligadas al cambio de tiempo cambian cuando se desliza el control de tiempo.

Capturas al desarrollar el caso de prueba 5

Al dar clic sobre una capa principal se abre una ventana modal en donde se visualiza el formulario de registro, esto formulario se lo completa en pasos, estos son: selección de capa base, nombre de la capa, etiqueta de la capa, y configuración de estilos de la capa.

En el paso de configuración de estilos se puede editar los colores de los marcadores a generar. Ver Figura 29 y 30.

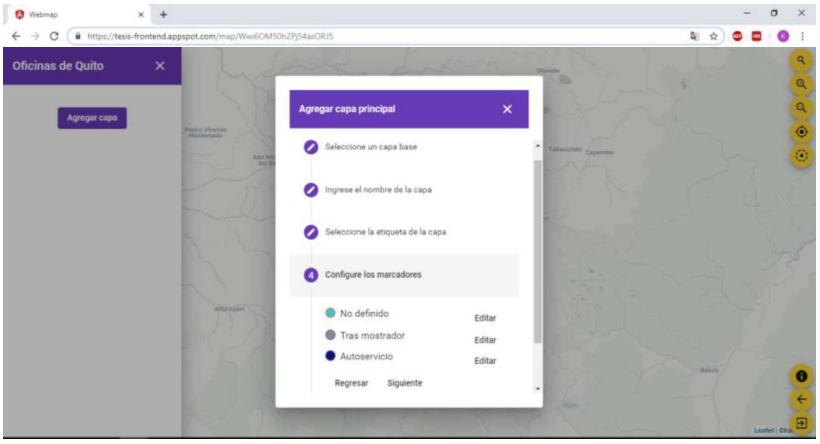

Figura 31 - Configuración de estilos de una capa

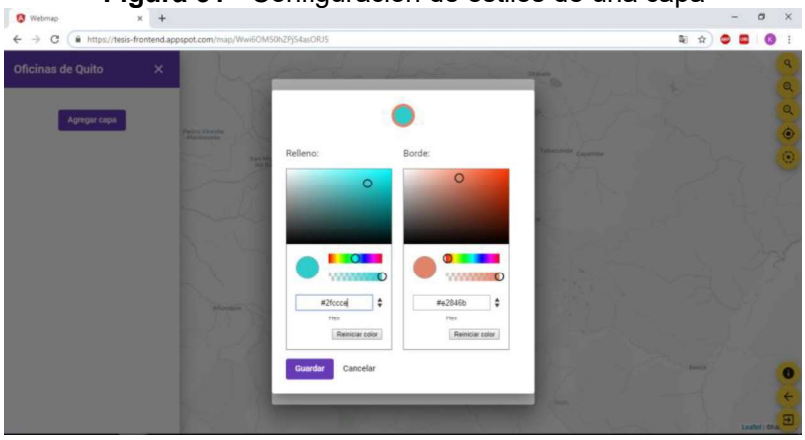

Figura 32 - Editor de colores

Una vez que todos los pasos sean válidos se procede a crear la capa, esta aparece en el menú izquierdo y se muestra un mensaje de notificación. Para activar la capa se da clic sobre el checkbox, se muestran la capa de acuerdo a la configuración especificada. Ver Figura 31.

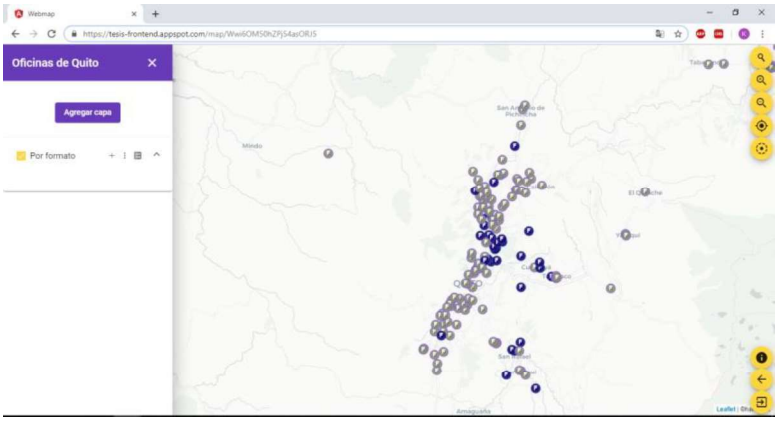

Figura 33 - Capas creadas de forma dinámica en el menú izquierdo

Para editar la configuración se abre la opción de configuración de la capa, y se realiza el cambio. Ver Figura 32.

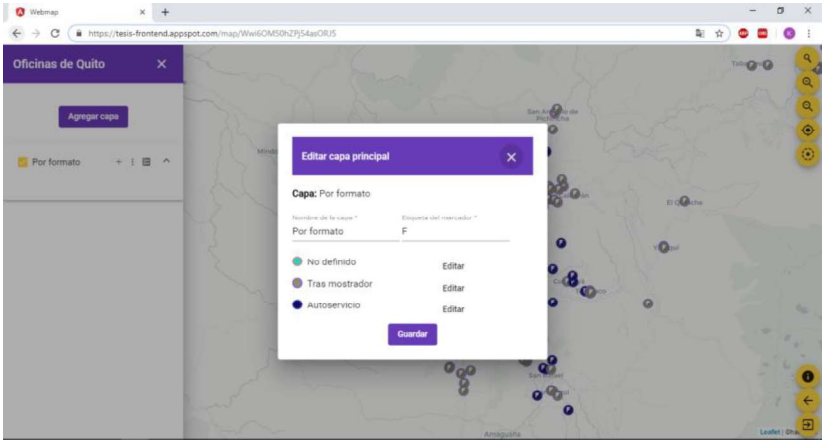

Figura 34 – Interfaz de edición de una capa

Para agregar una capa secundaria de análisis se da clic sobre el icono “+” en una capa principal. Se abre un formulario dividido en pasos en donde se selecciona el tipo de mapa a crear, se selecciona los valores a analizar y la configuración de la capa. Ver Figura 33.

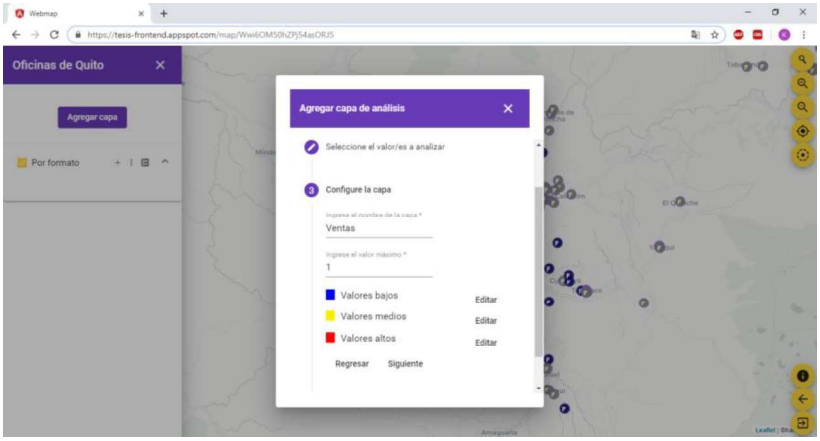

Figura 35 - Configuración de estilos para una capa secundaria

Una vez que los pasos son válidos se crea la capa secundaria. Se crea debajo de una capa principal y aparece el control de tiempo en la parte inferior. Ver Figura 34.

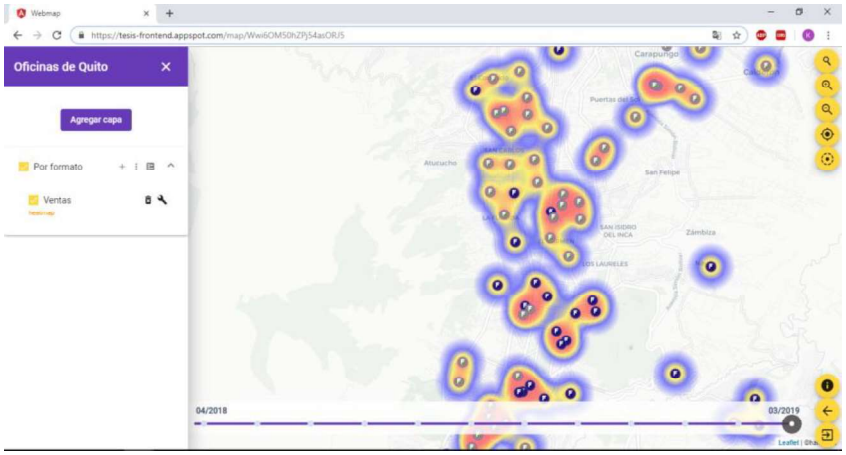

Figura 36 - Interfaz cuando se agrega una capa secundaria

Ahora se agrega otra capa secundaria, esta vez será una capa comparativa. Para cada capa secundaria existen controles para eliminar y configurar la capa, para la capa de análisis creada se procede a cambiar los colores y filtrar valores, o cambiar la fecha por medio de la barra deslizante inferior. Ver Figura 35.

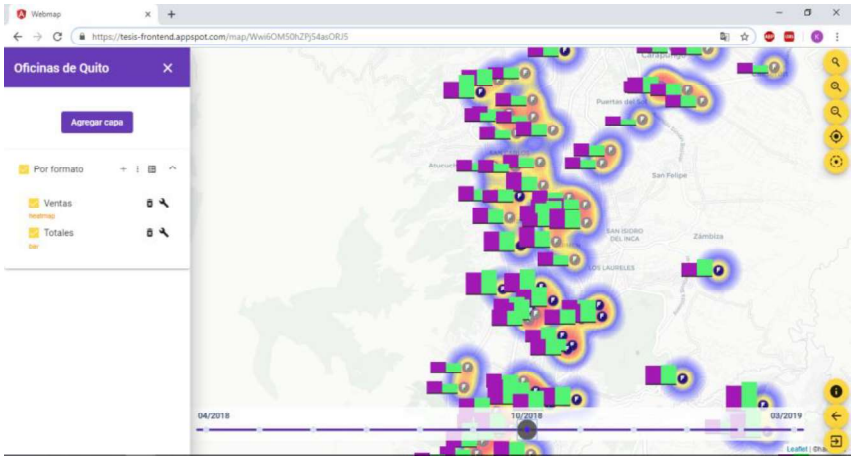

Figura 37 - Capa de análisis tipo barras

Tabla 81 - Caso de prueba 6

No.	CP6
Titulo	Herramientas de apoyo
Precondiciones	Haber creado un mapa en la pantalla “Mis mapas”
Acciones	**Resultados**

Para la búsqueda de direcciones se debe tener la opción de autocompletado.	Al dar clic sobre el botón de búsqueda, se abre el campo para ingresar la dirección, al dar "Enter", aparecen opciones para autocompletar la búsqueda.
Al buscar una dirección se debe re direccionar al sitio indicado.	Al seleccionar una opción de búsqueda el mapa se sitúa sobre las coordenadas correspondientes.
Las isócronas deben mostrar la información de área, población, tiempo, suma y promedio de las capas de análisis activas.	La información se muestra en forma de popup y en la pantalla de resumen de la isócrona. Los valores de conteo, suma y promedio se muestra en la pantalla de resumen de la zona.
La pantalla de reporte muestra en una tabla la información de las capas activas y la opción de descarga del reporte en formato PDF.	En la pantalla de reporte se muestra en un mapa la zona analizada, la información de la zona y la tabla con los datos de análisis. Se tiene la opción para seleccionar varias capas y la opción para exportar el reporte en formato PDF.

Capturas al desarrollar el caso de prueba 6

Para crear una capa de isócronas se da clic sobre el botón de activación de la herramienta en la parte superior derecha, aparece la instrucción en la parte superior derecha, para crear una capa se da clic sobre una posición en el mapa. Ver Figura 36.

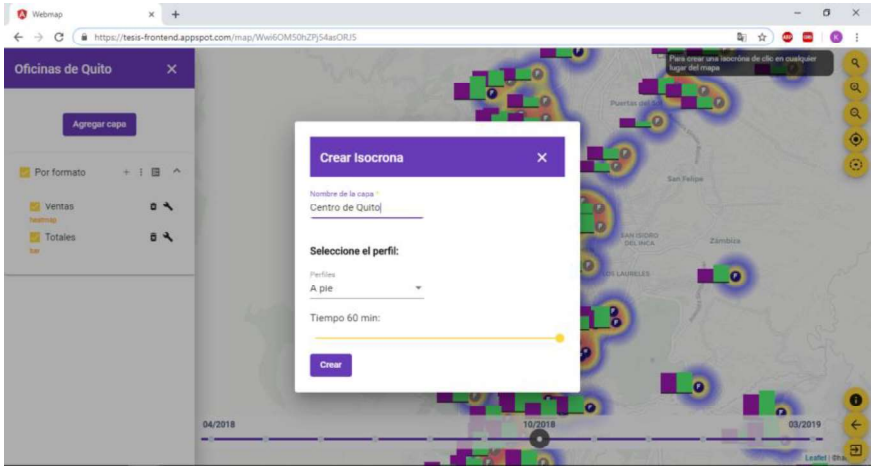

Figura 38 - Creación de isócronas

Cada isócrona muestra la información de la zona resaltada. Ver Figura 37.

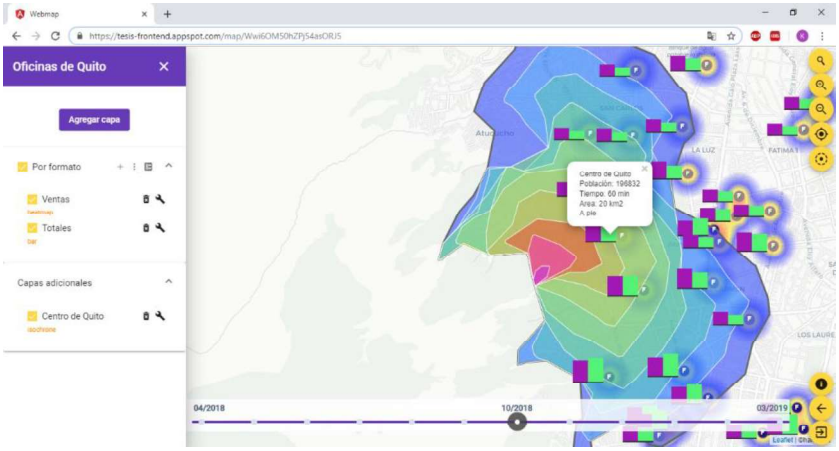

Figura 39 – Información zona de isócronas

Al dar clic sobre una zona de la isócrona se abre una ventana que muestra la información de la zona y el resumen de las capas de análisis activadas. Ver Figura 38.

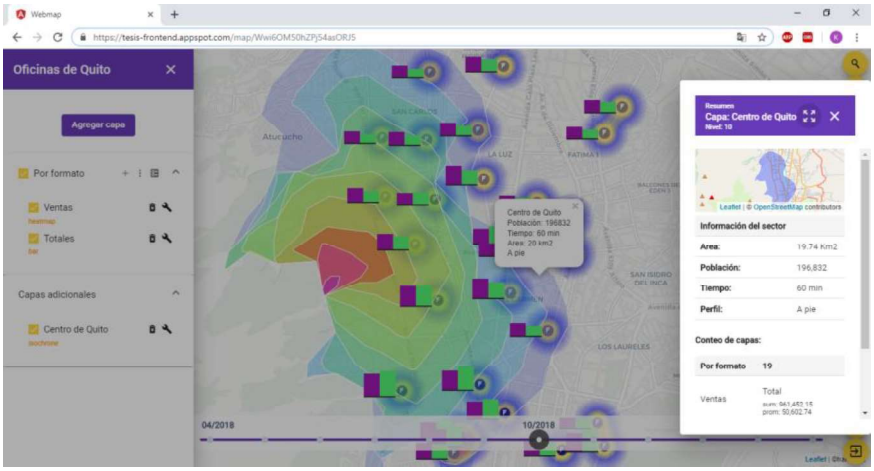

Figura 40 - Ventana para ver el resumen de la zona seleccionada

Al dar clic sobre la opción para ver más información se abre una ventana que muestra de forma más detallada la información de la zona seleccionada. Ver Figura 39.

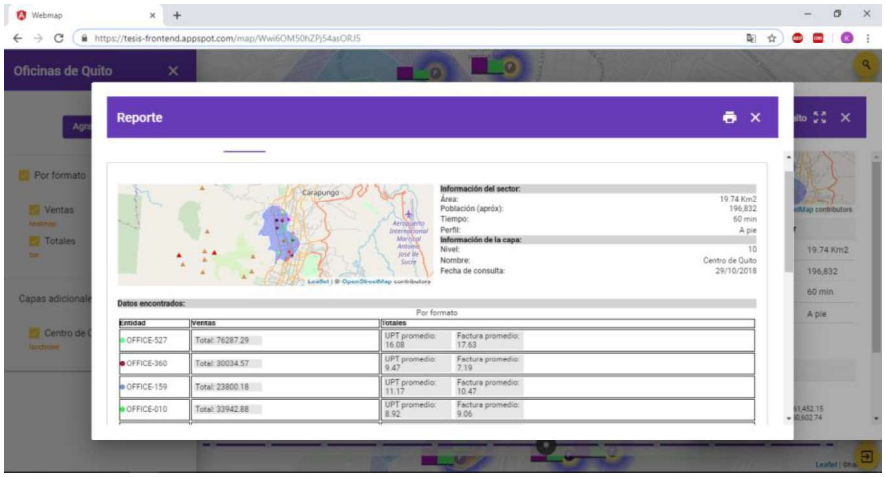

Figura 41 - Ventana para ver el reporte de la zona seleccionada

Se tiene la opción para agregar más zonas de la isócrona al reporte y la opción para exportarlo a un archivo PDF. Ver Figura 40.

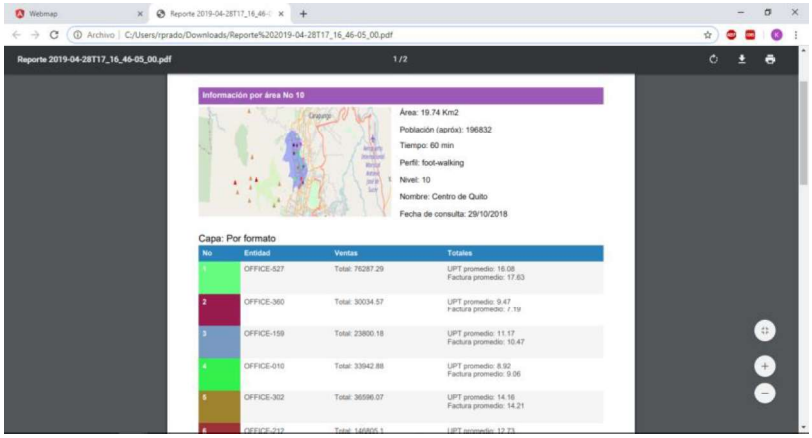

Figura 42 - Archivo descargado de la zona seleccionada

Pruebas de carga en el Front-End

Debido a que angular crea aplicaciones de una sola página, lo que significa que el contenido de la aplicación se descarga una sola vez al navegador y es desde donde se llama a toda la funcionalidad las pruebas de carga se limitan a determinar cuánto tiempo se demora el servidor en descargar estos scripts de la aplicación, los archivos son los siguientes (Ver Figura 41):

Figura 43 - Transpilación del proyecto en modo de producción

Para realizar las pruebas se usó Jmeter que es la herramienta más popular para realizar pruebas de carga de aplicaciones web y de servicios [31]. Las pruebas se realizaron contra el servidor implementado en Google Cloud con las siguientes configuraciones:

Tabla 82 - Características del servidor Front-End

Características del servidor	
CPU	2 Núcleos
Memoria	2 GB

Para las pruebas se consideró grupos de usuarios en tres categorías de carga y se consideró un tiempo de espera aceptado de 3000 milisegundos a 4000 milisegundos:

Tabla 83 - Configuración de las categorías de cargas usadas en Jmeter

Categoría	**No de usuarios**	**Periodo de peticiones (milisegundos)**	**No Bucles**
Normal	100	10	10
Media	250	5	10
Alta	500	1	10

Debido a que todos los archivos de la aplicación se descargan en el navegador para su posterior uso, los resultados de las pruebas, ver anexo H, fueron dependientes de la velocidad de conexión desde donde se realizado las pruebas. A continuación, la descripción de los recursos desde donde se realizó las pruebas:

Tabla 84 - Características del equipo donde se desarrolló las pruebas

Características del equipo	
CPU	2 núcleos
Memoria	8 GB
Velocidad de descarga	3MB

Pruebas para una carga normal

El tiempo medio de descarga de la aplicación fue de 2712ms, en el gráfico se puede observar como el rendimiento de descarga disminuye a medida que se realizan las peticiones. Sin embargo, el valor de la mediana para el tiempo de respuesta se mantiene en un rango aceptable. También, se puede notar una estabilización en la respuesta del servidor tanto en el rendimiento como en el valor de la mediana al final de la prueba. Ver Figura 42.

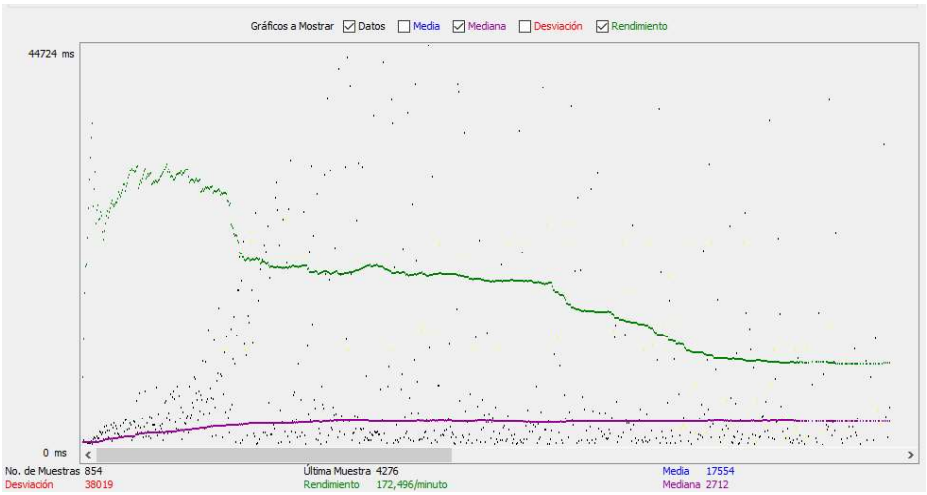

Figura 44 – Gráfico de resultado para una carga normal

En los resultados de comparación de la media y la mediana para cada archivo se pude observar que casi todos los archivos se descargan dentro del rango aceptable de tiempo, a diferencia del archivo principal de la aplicación que aumento de manera exponencial a medida que aumentaban las peticiones. Ver Figura 43.

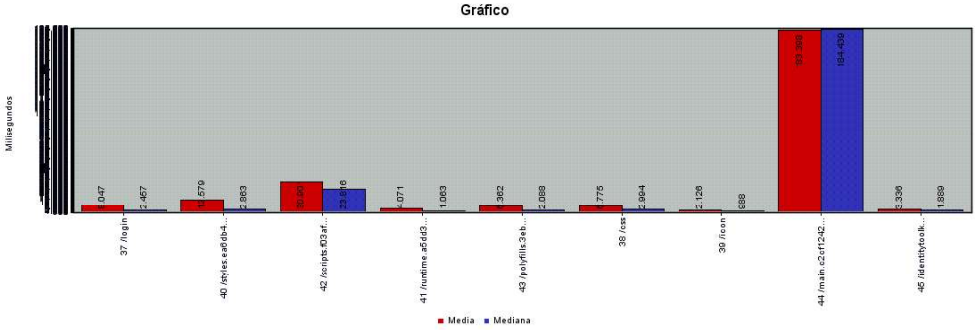

Figura 45 - Comparación de tiempos de descarga para cada archivo de la aplicación para una carga normal

Pruebas para una carga media

El tiempo medio de descarga de la aplicación fue de 10241ms, permaneciendo dentro del rango permitido en tiempos iniciales de la prueba, en el gráfico se puede observar como el

rendimiento de descarga disminuye a medida que se realizan las peticiones, aquí el valor de la mediana para el tiempo de respuesta aumenta a medida que se realizan las peticiones y se mantiene en un rango aceptable en tiempos iniciales de la prueba. También, se puede notar una estabilización en la respuesta del servidor tanto en el rendimiento como en el valor de la mediana al final de la prueba. Ver Figura 44.

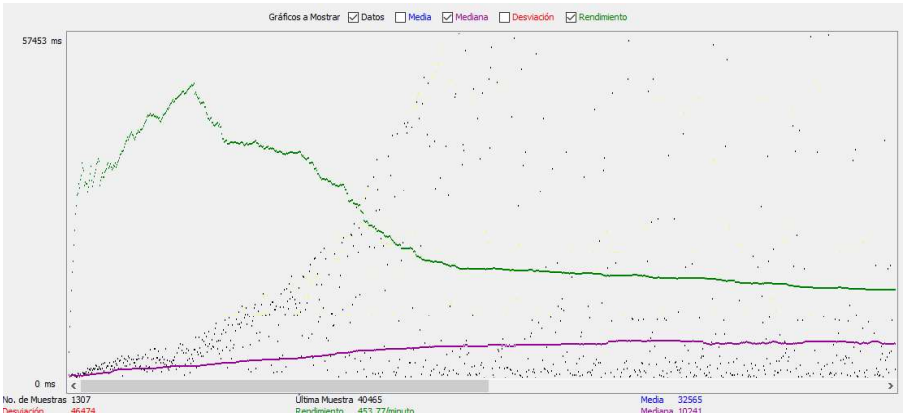

Figura 46 - Gráfico de resultado para una carga media

En los resultados de comparación de la media y la mediana para cada archivo se pude observar que casi todos los archivos se descargan dentro del rango aceptable de tiempo, a diferencia del archivo principal de la aplicación y de los scripts o librerías usadas las mismas que aumentaron de manera exponencial a medida que aumentaban las peticiones. Ver Figura 45.

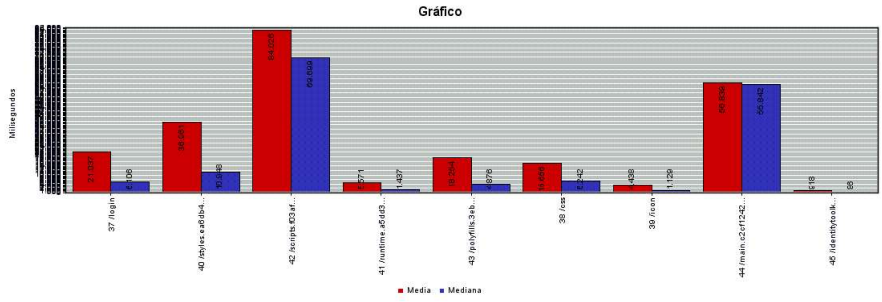

Figura 47 - Comparación de tiempos de descarga para cada archivo de la aplicación para una carga media

Pruebas para una carga alta

El tiempo medio de descarga de la aplicación fue de 29720ms, en el gráfico se puede observar como el rendimiento de descarga varia a medida que se realizan las peticiones, el valor de la mediana para el tiempo de respuesta no entra dentro del rango aceptable y su tendencia de crecimiento es positiva a medida que transcurre la prueba. También, no se puede notar una estabilización en la respuesta del servidor tanto en el rendimiento como en el valor de la mediana al final de la prueba. Ver Figura 46.

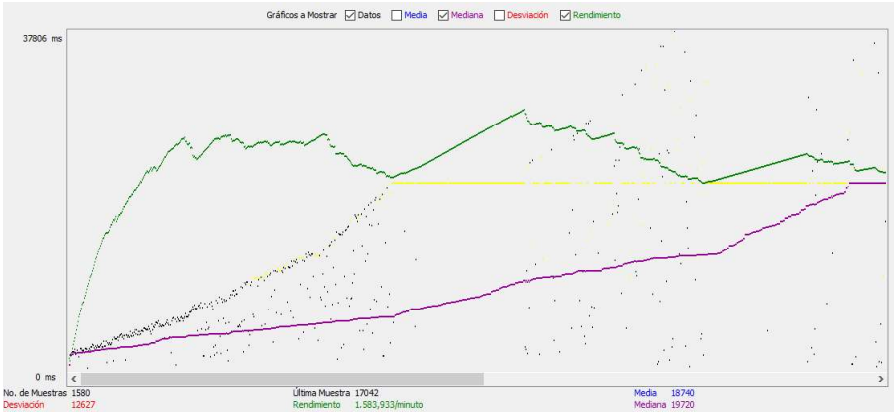

Figura 48 - Gráfico de resultado para una carga alta

En los resultados de comparación de la media y la mediana para cada archivo se pude observar que casi todos los archivos superan el tiempo aceptable de descarga, a diferencia de los archivos de iconos, se puede evidenciar que la carga de scripts y estilos es más lenta en este tipo de carga de usuarios. Ver Figura 47.

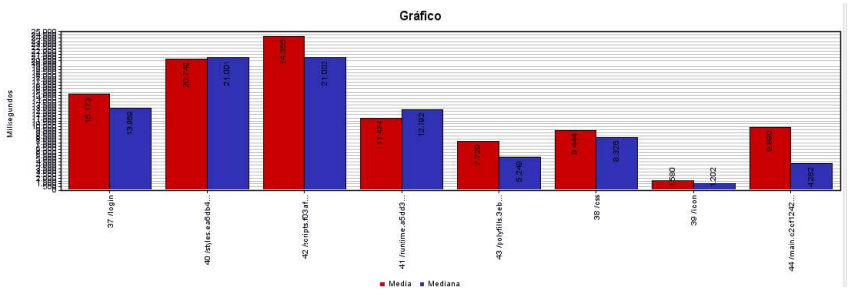

Figura 49 - Comparación de tiempos de descarga para cada archivo de la aplicación para una carga alta

Pruebas de estrés en el Back-end

Se decidió realizar pruebas de estrés para el servidor back-end debido a que es el responsable de leer y transformar grandes cantidades de datos durante periodos cortos y de forma variable en el tiempo. También, debido a que no se sabe cuál exactamente es la carga normal ya que varía de usuario a usuario las pruebas de carga no darían resultados confiables para sacar conclusiones.

Para realizar las pruebas se usó SoapUI que es la herramienta de prueba automatizada más utilizada del mundo para las API de SOAP y REST con arquitectura orientada a servicios y transferencia de estado representacional [32]. Las pruebas se realizaron contra el servidor implementado en Google Cloud con las siguientes configuraciones:

Tabla 85 - Características del servidor Back-end

Características del servidor	
CPU	4 Núcleos
Memoria	8 GB

Para las pruebas se creó tres servicios de prueba con las características de petición de datos más usadas y más costosas en cuanto a tiempo dentro de la aplicación, se consideró un tiempo de espera aceptado de 3000 milisegundos a 4000 milisegundos:

Tabla 86 - Configuraciones de las pruebas realizadas en SoapUI

Categoría	**No de usuarios**	**Periodo de peticiones (milisegundos)**	**Limite (segundos)**
Servicio de carga de entidades	9999	1ms	300
Servicio de carga de datos de análisis	9999	1ms	300
Servicio de conteo y suma de datos de análisis	9999	1ms	300

A continuación, la descripción de los recursos desde donde se realizó las pruebas:

Tabla 87 - Características del equipo donde se realizaron las pruebas

Características del equipo de pruebas	
CPU	2 Núcleos
Memoria	8 GB
Velocidad de descarga	3MB

Los resultados de las pruebas para los tres servicios se los puede observar en el anexo H.

Prueba al servicio para obtener entidades, sus características y coordenadas

En esta prueba, el servidor mantuvo un tiempo de respuesta óptima durante toda la prueba, entro los 148ms a 429 ms, lanzo algunos errores al inicio de las mismas. Sin embargo, estos no aumentaron durante casi los 5 minutos de prueba, asumiendo que el servidor respondió bien ante esta carga. En esta prueba la tasa de transacciones por segundo fue de 23286 por segundo, un valor óptimo para el numero de datos que se transmiten para este servicio. Ver Figura 48 y 49.

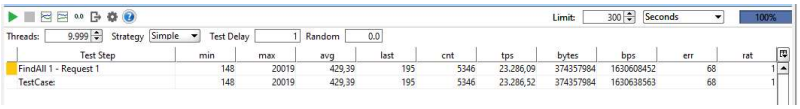

Figura 50 - Tabla de resultados para el servicio de prueba 1

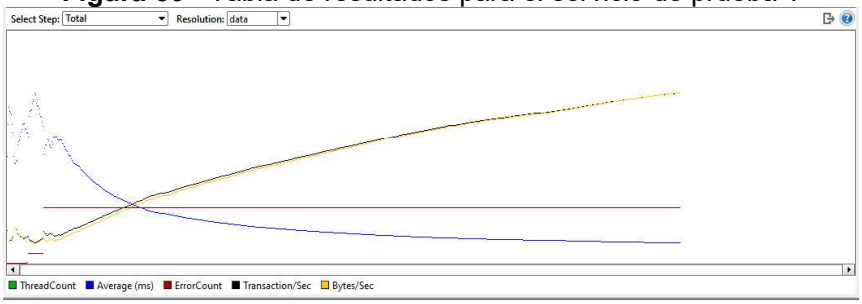

Figura 51 - Gráfico de resultados para el servicio de prueba 1

Prueba al servicio para obtener datos de las capas de análisis.

En esta prueba el servidor mantuvo un tiempo de respuesta optima durante toda la prueba, entro los 140ms a 426 ms, lanzo algunos errores al inicio de la mismas, estos no aumentaron durante casi los 5 minutos de prueba, asumiendo que el servidor respondió bien ante esta carga. En esta prueba, la tasa de transacciones por segundo fue de 37290 por segundo, un valor óptimo para el numero de datos que se transmiten para este servicio y superior al servicio para obtener datos de entidades. Ver Figura 50 y 51.

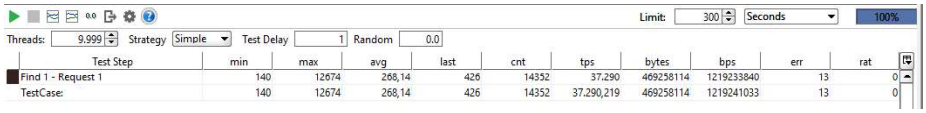

Figura 52 - Tabla de resultados para el servicio de prueba 2

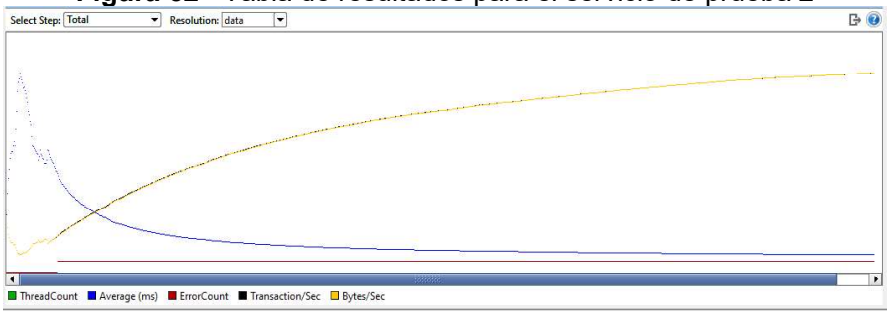

Figura 53 - Gráfico de resultados para el servicio de prueba 2

Prueba al servicio para realizar conteos y resúmenes en las capas de análisis.

En esta prueba el servidor mantuvo un tiempo de respuesta muy alta durante toda la prueba, entro los 1086ms a 66601 ms, lanzo errores durante toda la prueba, estos aumentaron durante casi los 5 minutos de prueba, la cantidad de datos entregados por el servicio es relativamente menor que los anteriores servicios. Sin embargo, el tiempo de espera es más elevado en este tipo de peticiones, en esta prueba la tasa de transacciones por segundo fue de 93301 por segundo, un valor óptimo para el numero de datos que se

transmiten para este servicio pero no se nota una recuperación o estabilización en cuanto al manejo de errores. Ver Figura 52 y 53.

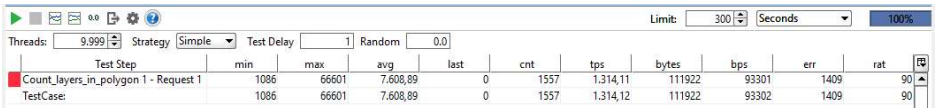

Figura 54 - Tabla de resultados para el servicio de prueba 3

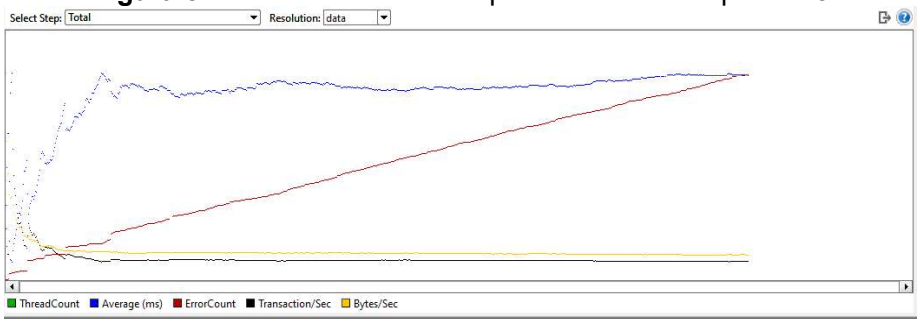

Figura 55 - Gráfico de resultados para el servicio de prueba 3

2.5. Pruebas de usabilidad

Para las pruebas de usabilidad se usa el método de encuesta implementando el sistema de escalabilidad (SUS, System usability scale). Esta escala está basada en preguntas de elección forzada, donde se hace una declaración y el encuestado indica el grado de acuerdo o desacuerdo con la declaración en una escala de 5 puntos. Si bien las escalas de Likert se presentan en esta forma, las declaraciones con las que el encuestado indica acuerdo y desacuerdo deben seleccionarse cuidadosamente. La técnica utilizada para seleccionar elementos para una escala Likert es identificar ejemplos de cosas que conducen a expresiones extremas de la actitud que se desea capturar del encuestado [44]. En los resultados de la encuesta, ver anexo G, de acuerdo a SUS se usó las siguientes ponderaciones:

- Si es una pregunta par se debe restar un punto.
- Si es una pregunta impar se debe restar 5 menos el valor marcado.
- Se multiplica el resultado de cada pregunta por 2.5.
- Se suma los 10 resultados.

Para la evaluación del test se usa las siguientes ponderaciones:

- Alrededor de 50 o más bajo es un sistema fallido.
- Alrededor de 68 se considera correcto, pero se puede mejorar.
- Más de 80 es sobresaliente.

Calculando las ponderaciones de cada pregunta de acuerdo al sistema SUS se obtuvieron los siguientes resultados:

Tabla 88 - Ponderación de resultados de la encuesta por medio del sistema SUS

E	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	SUM
1	10	7,5	10	5	10	7,5	7,5	10	10	5	82,5
2	7,5	10	10	10	10	10	7,5	10	10	2,5	87,5
3	7,5	2,5	10	5	5	10	5	5	5	7,5	62,5
4	10	2,5	10	2,5	10	10	5	7,5	10	2,5	70
5	7,5	5	5	5	7,5	5	5	5	7,5	5	57,5
6	10	7,5	7,5	7,5	10	10	10	10	10	10	92,5
7	7,5	2,5	7,5	10	7,5	10	10	10	10	10	85
8	7,5	10	7,5	10	10	0	10	10	7,5	10	82,5
9	10	7,5	10	7,5	10	10	7,5	10	5	7,5	85
10	7,5	2,5	7,5	7,5	7,5	5	5	7,5	7,5	2,5	60
										PROM	76.5

3. RESULTADOS Y DISCUSIÓN

En cuanto a funcionalidad, la integración con tecnologías que trabajan muy bien en el entorno de tiempo real permitió crear interfaces de configuración rápidas. Esto debido a, tareas como el guardar y actualizar datos se resuelven automáticamente y permiten un desempeño más rápido al momento de actualizar las interfaces.

La velocidad de carga de capas también se encuentra en un rango aceptable, de 1 a 3 segundos [45]. El promedio de capas que un usuario usa en un mismo análisis, de tres a cinco capas, la velocidad desde que inicia la petición, la transformación de datos a través del modelo de representación y la visualización se encuentra en este rango aceptable, lo que es muy importante para mejorar la experiencia de usuario, las herramientas de filtro y de visualización ayudan en gran medida ya que previenen una excesiva carga de elementos a la pantalla. En las pruebas de carga el sistema respondió bien para cargas normales del sistema, para cargas extremadamente altas el sistema aumenta su tiempo de respuesta en proporciones manejables sin que este dejara de funcionar, lo que representa un éxito a poder soportar este tipo de cargas.

Para crear este tipo de sistemas es necesario la integración de los procesos extracción, transformación y carga de la organización, en este sentido una tabla estandarizada ayuda gestionar datos espaciales que provengan de distintas fuentes.

La selección de procesos que se correrán en el servidor o en el navegador permite mejorar el rendimiento de la aplicación, se evidenció cómo en procesos en donde se requería cálculos en listas y que se las llevó a cabo en el lado del servidor permitió liberar al navegador evitando que este se bloqueara durante estos cálculos para esto se configuro un servidor de Back-end con una alta capacidad de procesamiento para que soporte una gran cantidad de peticiones como se pudo observar el la pruebas de estrés ejecutadas en contra de este servidor.

Mientras más capas se agregan al mapa, el rendimiento de la aplicación va disminuyendo, esto debido al modelo vectorial usado y a la capacidad del equipo usado. Por lo tanto, es necesario usar para este tipo de aplicaciones equipos con una alta capacidad, se recomienda una RAM de 8GB y al menos 2 núcleos en el procesador.

De acuerdo al resultado promedio el sistema se encuentra en un valor superior a la valoración para un sistema usable de manera correcta, se acerca mucho al valor de 80 que es el valor que da un sistema perfecto en cuanto a usabilidad.

4. CONCLUSIONES

Al desarrollar el sistema mediante el framework Scrum se logró mantener un ritmo estable en el desarrollo permitiendo distribuir el esfuerzo de trabajo y controlar la entrega de resultados según lo planificado en cada Sprint. Las metodologías usadas en Scrum, como la estimación de historias de usuario por puntos de esfuerzo, roles dentro del proyecto o los artefactos de Scrum como los burndown charts permitieron tener un control e inspección del proyecto al final de cada iteración obteniendo resultados funcionales de acuerdo a los requerimientos especificados en el Product Backlog por parte del dueño del producto y cumpliendo con los criterios de aceptación en cada historia de usuario. El proyecto no presento grandes cambios a largo del proyecto, excepto en el último Sprint en donde se añadió o actualizo historias de usuario con un alto riesgo de desarrollo, estos cambios fueron resueltos rápidamente y agregados al producto final satisfactoriamente.

El uso de una base de datos como MemSQL que es considera de la clase “New SQL”, permitió manipular datos estructurados y no estructurados, esto es importante ya que los datos para crear elementos espaciales deben tener el formato JSON mientras que los datos de los cuales se obtienen la información de las organizaciones se origina en bases de datos relacionales como SQL Server y MySQL. Así, mediante procesos de carga se creó tablas que combinan estas características para tener un modelo estandarizado y llevar cabo la transformación de datos en elementos espaciales de forma más eficiente. La base de datos usada permite hacer consultas geográficas permitiendo emular en conjunto con un servidor

Node JS, un servidor GIS con una alta capacidad de respuesta y procesamiento en consultas.

Otra parte fundamental en el desarrollo del sistema fue definir las operaciones que se ejecutaron en el servidor y las que se ejecutaron en el navegador como recomienda el modelo de representación mencionado en la sección 1.5.4, esto permitió aumentar el rendimiento de la aplicación. Por ejemplo, las funciones que llevaban a cabo operaciones en listas se las ejecuto en el servidor evitando que la aplicación que se ejecuta en el navegador se bloqueara, esto se evidenció en las pruebas de rendimiento, para las pruebas de carga en la aplicación web se logró mantener un tiempo aproximado de 3 a 4 segundos, aunque este valor depende mucho de la velocidad de conexión a internet y los recursos del equipo. Para las pruebas de estrés en los servicios para consulta de datos se lograron tiempos de menos de 1 segundo, esto se refleja en la rapidez en la que se representan esto datos en la aplicación, para el tercer servicio que se usa en las consultas de análisis y que por ende se realizan consultas más complejas, el tiempo de respuesta para cargas muy altas es más alto, alrededor de 10 segundos. Sin embargo, tanto el servidor como la aplicación estuvieron disponibles a lo largo de todas las pruebas. También, al final de proyecto se realizó pruebas funcionales, para esto se creó casos de prueba de acuerdo a los criterios de aceptación de cada historia de usuario épica, se evidencio que todos estos criterios de aceptación fueron resueltos a lo largo de todo el proyecto a medida que se completaban los Sprints.

En cuanto a la usabilidad del proyecto, las interfaces de creación de gráficas funcionan según los objetivos planteados. Lo que significa que, sean fáciles de entender, configurar y que permitan al usuario llevar a cabo completar tareas. Esto se reflejó en la prueba de usabilidad realizada por medio de la encuesta, la evaluación de la misma estuvo dentro de la categoría “Correcta” y aproximándose a la calificación de “Sobresaliente”, el sistema puede llegar a tener esta calificación mejorando la parte visual de las interfaces generales requiriendo en este caso conocimientos más avanzados de diseño gráfico y en “User experience/User Interface”. Sin embargo, el trabajo realizado en cuanto al diseño de mapas web dinámicos y de su funcionalidad que fue el objetivo principal de este proyecto se logró con éxito.

La percepción de usabilidad varía de usuario en usuario esto hace que el desarrollo de interfaces intuitivas sea extremadamente complejo, para solventar eso es necesario fijar los objetivos y usar cualquier elemento de apoyo para brindar información extra al usuario como botones con avisos, mensajes de notificación, colores, leyendas, etc.

Al momento de crear aplicaciones de mapas web dinámicos es de vital importancia analizar y decidir cuál será el objetivo de análisis y que tareas de análisis se llevara a cabo, esto permitirá crear aplicaciones que sean útiles y que permitan al usuario llevar a cabo tareas para las que se supone fue creado el mapa. Implementar demasiados objetivos de análisis conlleva a crear más interfaces y herramientas provocando confusión en el usuario y aumentado la posibilidad de que no puede completar tareas.

Debido a la transferencia de datos estratégicos en las peticiones que se ejecuta es necesario que la petición de datos se ejecute siempre a través de dominios seguros, esto es usando el protocolo “https”, así mismo se debe evitar el envío de credenciales y en el caso de que se requiera enviar estas credenciales estas deben estar encriptados e ir dentro de la cabecera de la petición.

5. REFERENCIAS BIBLIOGRÁFICAS

[1] A. Karabegovic y M. Ponjavic, «Geoportal as Decision Support System with Spatial Data Warehouse,» *Computer Science and Information Systems, Sarajevo,* pp. 915-918, 2012.

[2] C. Rinner, «Web-based Spatial Decision Support: Status and Research Directions,» *Geography Publications and Research, Ryerson University, Toronto,* pp. 14-31, 2003.

[3] INEC, «Ecuador en cifras,» Agosto 2014. [En línea]. Available: http://www.ecuadorencifras.gob.ec/informacion-censal-por-provincias/.

[4] V. Estévez, L. Garmendia y E. Giménez, «Visión general de los Sistemas de Información Geográfica,» *eprints ucm es,* pp. 7-72, 2013.

[5] J. Heard, J. McGee y B. Blanton, «Geoanalytics - A Solution Driven CI Platform for Geospatially Engaged Science,» p. 6, 2011.

[6] F. Halper, «Seven Use Cases for Geospatial Analytic,» *TDWI Checklist Report,* p. 8, 2013.

[7] A. Sandeep y L. Hui Yee, «BIG DATA GEOANALYTICS,» *ESRI,* p. 32, 2017.

[8] N. Andrienko, T. Lammarsch, G. Andrienko, G. Fuchs, D. Keim, S. Miksch y A. Rind, «Viewing Visual Analytics as Model Building,» *COMPUTER GRAPHICS Forum,* 2018.

[9] V. Olaya, Sistemas de Información Geográfica, Girona, 2014, pp. 79-116.

[10] A. Alesheikh y H. Helali, «Web GIS: Technologies and its applications,» *Research Gate,* 2015.

[11] A. Doyle, «Web Map Server Interface Specification,» *OpenGIS Project.*

[12] R. Sieber y S. Wiesmann, «Graphical User Interface - Layout and Design,» 2015.

[13] M. Gaigg, «UI/UX Best Practices for designing map apps,» 2015.

[14] K. Schwber y J. Sutherland, «La Guia Scrum TM,» *Scrum Guides,* 2016.

[15] J. Palacio, «Scrum Manager I: Las reglas de Scrum,» 2015. [En línea]. Available: https://www.scrummanager.net/files/scrum_I.pdf.

[16] L. Oscar y G. Kindley, «Redes de computadores: Arquitectura Cliente - Servidor,» pp. 3-4.

[17] J. Pillou, «Redes - Arquitectura Cliente/Servidor en 3 niveles,» [En línea]. Available: https://es.ccm.net/contents/147-redes-arquitectura-cliente-servidor-en-3-niveles.

[18] Mozilla.org, «JavaScript (JS),» MDN Web Docs, 2019. [En línea]. Available: https://developer.mozilla.org/es/docs/Web/JavaScript.

[19] D. Rosenwasser, «Microsoft: Dev Blogs,» 2019. [En línea]. Available: https://devblogs.microsoft.com/typescript/announcing-typescript-3-3/.

[20] Mozilla.org, «HTML: HyperText Markup Language,» MDN Web Docs, 2018. [En línea]. Available: https://developer.mozilla.org/es/docs/Web/HTML.

[21] C. Hampton, «Sass,» 2018. [En línea]. Available: https://sass-lang.com/documentation/file.SASS_REFERENCE.html.

[22] Node Foundation , «Node JS Guides,» 2018. [En línea]. Available: https://nodejs.org/es/docs/guides/.

[23] J. Oliveira y J. Bernardino, «NewSQL Databases: MemSQL and VoltDB Experimental Evaluation,» 2015.

[24] Node Foundation, «Express JS: Fast, unopinionated, minimalist web framework for Node.js,» StrongLoop, 2019. [En línea]. Available: https://expressjs.com/es/resources/glossary.html.

[25] D. Wilson, N. Lillich y A. Sidorov, «mysql: node.js driver for mysql,» npm, 2019. [En línea]. Available: https://www.npmjs.com/package/mysql.

[26] Node Foundation, «Node JS: Child Process,» Joyent, Inc, 2018. [En línea]. Available: https://nodejs.org/api/child_process.html.

[27] Angular.io, «Introduction to the Angular Docs,» Google, 2019. [En línea]. Available: https://angular.io/docs.

[28] V. Agafonkin, «Leaflet,» [En línea]. Available: https://leafletjs.com/index.html.

[29] J. Delaney, «Angular Firebase,» [En línea]. Available: https://angularfirebase.com/.

[30] Angular.io, «Reactive Extensions for JavaScript,» Google, 2019. [En línea]. Available: https://angular.io/guide/rx-library.

[31] Apache Software Foundation, «Apache JMeter,» apache.org, 1999. [En línea]. Available: https://jmeter.apache.org/. [Último acceso: 2019].

[32] SmartBear, «SoapUI: REST & SOAP Testing Tool,» SmartBear Software, 2019. [En línea]. Available: https://www.soapui.org/.

[33] Firebase, «Firebase: A comprehensive mobile development platform,» Google, 2019. [En línea]. Available: https://firebase.google.com/?hl=es-419.

[34] Atlassian, «Bitbucket: Solución de Git para equipos profesionales,» Atlassian, 2019. [En línea]. Available: https://es.atlassian.com/software/bitbucket.

[35] SIGTE, «¿QUÉ SOFTWARE Y HERRAMIENTAS DE SIG UTILIZAMOS?,» UNIGIS, 2017. [En línea]. Available: https://www.unigis.es/software-sig/.

[36] Carto, «The world's leading Location Intelligence platform,» Carto, 2019. [En línea]. Available: https://carto.com/platform/spatial-data-science/.

[37] Qlik, «Qlik GeoAnalytics,» QlikTech International , 2019. [En línea]. Available: https://www.qlik.com/us/products/qlik-geoanalytics.

[38] Tableau, «Tableau Online,» TABLEAU SOFTWARE, 2019. [En línea]. Available: https://www.tableau.com/es-es/products.

[39] ArcGis, «Descripción general de la caja de herramientas GeoAnalytics,» Esri, 2019. [En línea]. Available: https://pro.arcgis.com/es/pro-app/tool-reference/big-data-analytics/an-overview-of-the-big-data-analytics-toolbox.htm.

[40] Gis geography, «Seven Geoprocessing Tools Every GIS Analyst Should Know,» GISGeography, 2019. [En línea]. Available: https://gisgeography.com/geoprocessing-tools/.

[41] P. Rogerson y A. Fotheringham, «GIS and spatial analysis: introduction and overview,» [En línea]. Available: https://www.researchgate.net/publication/246324467_GIS_and_spatial_analysis_introduction_and_overview.

[42] L. Pesquer, J. Masó y X. Pons, «Herramientas de Análisis Combinado Ráster/Vector en un Entorno SIG,» *Research Gate,* 2019.

[43] J. Peña, «Pruebas de software, fundamntos y tecnicas.,» 2015. [En línea]. Available: http://oa.upm.es/40012/1/PFC_JOSE_MANUEL_SANCHEZ_PENO_3.pdf.

[44] T. Nathan, «How to use the system usability scale (SUS) to evaluate your Website,» UsabilityGeek, 2018. [En línea]. Available: https://usabilitygeek.com/how-to-use-the-system-usability-scale-sus-to-evaluate-the-usability-of-your-website/.

[45] SmartBear, «The cost of poor web performance,» 2012. [En línea]. Available: https://smartbear.com/blog/test-and-monitor/the-cost-of-poor-web-performance-infographic/.

6. ANEXOS

I. Anexo A: Entrevistas – puntos relevantes

Documento adjunto:

- Anotaciones relevantes obtenidas en las entrevistas – Anotaciones de entrevistas (formato docx).

II. Anexo B: Pila del producto

Documento adjunto:

- Pila del producto – Lista de Historias de Usuario (formato xlsx).

III. Anexo C: Base de datos – Modelo Físico

Documentos adjuntos:

- Base de datos - Modelo Físico (formato png).
- Script para implementación de estructura Base de Datos (formato sql).

IV. Anexo D: Prototipos de interfaces (Perfil Usuario - Administrador)

Documentos adjuntos:

- Carpeta: Prototipos interfaces usuario final (formato PNG).
- Carpeta Prototipos interfaces usuario administrador (formato PNG).

V. Anexo E: Tablas y gráficos Burndown Charts

Documento adjunto:

- Tablas y gráficos Burndown Sprint 1 - 6 (formato xlsx).

VI. Anexo F: Casos de prueba

Documento adjunto:

- Tablas Casos de prueba Sprint 1 - 6 (formato xlsx).

VII. Anexo G: Encuesta de Usabilidad y Satisfacción del Usuario

Documentos adjuntos:

- Encuesta de Usabilidad y Satisfacción del Usuario (formato pdf).
- Enlace Web Google Forms (formato txt).
- Respuestas de la encuesta (formato xslx)

VIII. Anexo H: Pruebas de rendimiento

Documentos adjuntos:

- Resultados de pruebas con Jmeter (formato png).
- Resultados de pruebas con SoapUI (formato csv).

IX. Anexo I: Archivos de configuración para implementar el sistema en la nube

Documentos adjuntos:

- Archivo de configuración para el Back-end (formato yaml).
- Archivo de configuración para el Front-end (formato yaml).