



TECNOLÓGICO DE ESTUDIOS SUPERIORES DE ECATEPEC

DIVISION DE INGENIERIA EN SISTEMAS COMPUTACIONALES

**DESARROLLO E IMPLEMENTACIÓN DE UN VISUALIZADOR
GEOGRAFICO Y CHAT TÁCTICO PARA SIGESAFAM**

T E S I S

**QUE PARA OBTENER EL TÍTULO DE
INGENIERO EN SISTEMAS COMPUTACIONALES**

P R E S E N T A N:

LUIS EDUARDO PÉREZ HERNÁNDEZ

MARLON PINDTER GOMEZ

ELICEO MIXTLI SERRANO ALTAMIRANO

DIRECTORA: GRISELDA CORTÉS BARRERA.

ASESORES: EDGAR CORONA ORGANICHE, MERCEDES FLORES FLORES.

ECATEPEC DE MORELOS, EDO. DE MÉXICO, 2022.

Agradecimientos

Agradezco a mi familia, por su comprensión y apoyo a lo largo de toda mi vida, a todos aquellos que han aportado en mi formación académica y personal de una u otra manera, a mi tutora por sus consejos y guías en mi desarrollo profesional, muchas gracias a todos.

En especial, quiero agradecer a la Dra. Griselda Cortes la confianza que depositó en mí, su constante apoyo, sus indicaciones y orientaciones indispensables en el desarrollo de este trabajo. Quisiera destacar la buena actitud que la caracteriza.

Luis Eduardo Pérez Hernández

Quiero agradecer a mi familia que han sido el principal apoyo en momentos de flaqueza y debilidad ya que con sus consejos, enseñanzas y tolerancia me han brindado ese respaldo incondicional para culminar esta etapa de mi vida.

Gracias a la Dra. Griselda Cortes, Tutora de mi Tesis, a quien hago llegar mi más sincero agradecimiento, por permitirme ser partícipe de uno de sus proyectos dentro de la Universidad y por su entrega incondicional durante el desarrollo de este trabajo de investigación y a todos mis amigos y amigas que me ayudaron en la ejecución de la tesis.

Marlon Pindter Gómez

A mi familia, porque son lo más importante que tengo en la vida, por ser siempre mis principales motivadores y los formadores de lo que ahora soy como persona, sin ustedes y sus consejos yo no habría llegado hasta donde estoy.

Gracias a mis amigos y compañeros por los buenos momentos que hemos compartido. En especial un valioso reconocimiento a Wendy García, Roberto Farfán, Fernando Ramírez, Edgar Salinas, Alejandro Omaña, Alan Cruz, Mishel Brito, Geovanni Barragán, Marlon Pindter, Evelyn Coronado y Mariana Cruz los que me han demostrado su apoyo y brindado sus ánimos y consejos durante mi formación académica.

Agradecer también de manera especial a la Dra. Griselda Cortes, por brindar el apoyo y la ayuda necesaria para realizar este proyecto, por haber sido muy paciente y haber sido esa persona que con sus directrices pudo explicarme aquellos detalles para culminar mi tesis.

Eliceo Mixtli Serrano Altamirano

Resumen

La geolocalización es una tecnología que utiliza datos obtenidos de la computadora o dispositivo móvil de un individuo para identificar o describir su ubicación física real. Actualmente una de las manifestaciones más populares del desarrollo en las tecnologías de la información y recientemente está experimentando un aumento significativo de popularidad.

La tecnología de geolocalización se ha convertido en una base para los servicios de posicionamiento por ubicación y las aplicaciones sensibles a la ubicación que se ejecutan en los dispositivos inteligentes.

Los datos de geolocalización tienen una variedad de usos, cada uno de los cuales se puede personalizar para determinadas aplicaciones, entornos o empresas. Actualmente estos usos incluyen la ubicación y/o personalización de los contenidos distribuidos, la aplicación de restricciones de acceso y distribución de acuerdo a la ubicación geográfica.

Contenido

Agradecimientos	2
Resumen.....	3
Contenido.....	4
Índice de Figuras	7
Índice de Tablas.....	9
Capítulo I Introducción.....	10
1.1 Antecedentes	11
1.2 Justificación	15
1.2 Objetivos	17
1.3.1 General	17
1.3.2 Específicos	17
1.4 Alcances y limitaciones	18
1.4.1 Alcances	18
1.4.2 Limitaciones	18
1.5 Hipótesis	19
1.6 Planteamiento del problema	19
Capítulo II Marco teórico	22
2.1 Gestor de base de datos	23
2.1.1 MariaDB	23
2.2 Servidor web	23
2.2.1 Apache HTTP Server	23
2.3 Teselas Vectoriales	24
2.3.1 Geoserver	24
2.3.2 MapTiler	24
2.4 JavaScript	25
2.4.1 NodeJs	25
2.4.2 TypeScript	25
2.4.2.1 TypeORM	25
2.4.3 Express	26
2.4.4 Angular	26
2.4.4.1 Testing	26

2.4.4.1.1 Jasmine	27
2.4.4.1.2 Karma.....	27
2.4.5 Leaflet.....	27
2.5 Sistema de Posicionamiento global (GPS).....	27
2.5.1 Segmentos del GPS.....	28
2.5.1.1 Segmento espacial	28
2.5.1.2 Segmento de control.....	29
2.5.1.3 Segmento de usuario.....	30
2.6 Señales GPS	30
2.7 Determinación de posiciones del GPS	31
2.7 Receptores GPS.....	32
2.8 Fuentes de error del GPS	32
2.9 Tecnología GSM.....	33
2.9.1 Arquitectura red GSM.....	33
2.10 Metodologías de desarrollo de software.....	35
2.10.1 SCRUM	35
2.10.1.1 Planificación del Sprint	36
2.10.1.2 Scrum Diario	37
2.10.1.3 Revisión del Sprint	37
2.10.1.4 Retrospectiva.....	37
Capítulo III	38
Metodología y arquitectura de solución.....	38
3.1 Arquitectura de la ETSA	39
3.2 Herramientas tecnológicas para desarrollo de la ETSA	40
3.3 Niveles generales ETSA.....	41
3.4 Chat táctico	43
3.4.1 Servicio de contactos.....	44
3.4.2 Servicio de conversaciones	45
3.4.3 Servicio de grupos.....	46
3.4.4 Servicio de mensaje individual nuevo.....	47
3.4.5 Servicio de mensaje grupal nuevo	48
3.4.6 Servicio para mensajes individuales nuevos	49
3.4.7 Servicio para mensajes grupales nuevos.....	50

3.4.8 Método para enviar un mensaje	51
3.4.9 Método para abrir un contacto	52
3.4.10 Método para abrir una conversación	52
3.5 Modelo entidad – relación	53
3.6 Conversión de mapas en formato ráster a teselas vectoriales	54
3.7 Servidor cartógrafo	55
3.7.1 Carga de mapas en formato ECW	55
3.7.2 Espacio de Trabajo	59
3.7.2 Previsualización de Mapas	62
Capítulo IV Análisis e interpretación de resultados	63
4.1 Instalación y Configuración del Entorno de Desarrollo	64
4.1.1 Configuración del Servidor cartográfico con Tiles	64
4.1.2 Configuración del Servidor cartográfico con ECW	65
4.2 Implementación y pruebas TRL5 del Visualizador Geográfico y Chat Táctico	69
4.3 Resultados de las pruebas unitarias	71
Capítulo V Conclusiones	73
5.2 Recomendaciones	75
Bibliografía	76

Índice de Figuras

Fig. 1 Constelación de satélites	28
Fig. 2 Ubicación de las estaciones en el mundo	29
Fig. 3 Cuatro satélites calculando las dimensiones	31
Fig. 4 Estructura de la red GSM.....	35
Fig. 5 El ciclo de SCRUM	36
Fig. 6 Diagrama de niveles	39
Fig. 7 Arquitectura ETSA	40
Fig. 8 Niveles generales ETSA	41
Fig. 9 Diagrama visualizador geográfico	43
Fig. 10 diagrama general de los servicios para el chat táctico	44
Fig. 11 Servicio de contactos	44
Fig. 12 Servicio de conversaciones.....	45
Fig. 13 Servicio de grupos	46
Fig. 14 Servicio de mensaje individual nuevo	47
Fig. 15 Servicio de mensaje grupal nuevo	48
Fig. 16 Servicio mensajes individuales nuevos	49
Fig. 17 Servicio para mensajes individuales nuevos	50
Fig. 18 Servicio para mensajes grupales nuevos	50
Fig. 19 Método para enviar un mensaje	51
Fig. 20 Método para abrir un contacto	52
Fig. 21 Método para abrir una conversación	52
Fig. 22 Modelo E.R.	53
Fig. 23 Conversión de mapas con MapTiler.....	54
Fig. 24 Id Docker	56
Fig. 25 Estatus del Docker	56
Fig. 26 Directorios del Docker.....	56
Fig. 27 Directorio data_dir.....	57
Fig. 28 Copiar mapas al docker	57
Fig. 29 Listar archivos.....	57
Fig. 30 GeoServer	58
Fig. 31 Pantalla de bienvenida GeoServer.....	58
Fig. 32 Nuevo espacio de trabajo	59
Fig. 33 Asignar nombre.....	59
Fig. 34 Nuevo almacén de datos	60
Fig. 35 Formato ECW	60
Fig. 36 Seleccionar directorio.....	61
Fig. 37 Mapas disponibles	61
Fig. 38 Guardar mapa.....	62
Fig. 39Previsualización de mapa	62
Fig. 40 Mapas alojados en el servidor.....	64
Fig. 41 Origen de datos formato ECW	65
Fig. 42 Implementación de GeoServer.....	65
Fig. 43 Consumo de memoria (método 1).....	66
Fig. 44 Consumo de memoria (método 2).....	66
Fig. 45 Representación gráfica del consumo de memoria (método 1).....	67

Fig. 46 Representación gráfica del consumo de memoria (método 2).....	67
Fig. 47 Rendimiento (método 1).....	68
Fig. 48 Rendimiento (método 2).....	68
Fig. 49 Registros de datos recibidos.....	69
Fig. 50 Mapa con marcador visible.....	70
Fig. 51 Rastro de las aeronaves.....	70
Fig. 52 Pruebas del chat táctico y varias aeronaves.....	71

Índice de Tablas

Tabla 1 Kit de herramientas	40
Tabla 2 Actividades en entorno Windows	55
Tabla 3 Actividades en entorno Linux	55
Tabla 4 Configurar servidor de pruebas Ubuntu 20.04 LTS	64
Tabla 5 Etapa uno de pruebas unitarias del software.....	71
Tabla 6 Estado de las pruebas en servicios.....	72
Tabla 7 Estado de las pruebas GUI	72

Capítulo I Introducción

Una de las necesidades más importantes para la Secretaría de Defensa Nacional (SEDENA) es mantener la comunicación constantes con las aeronaves, teniendo acceso a la ubicación actual, así como la capacidad de visualizar el rastro en caso de que se encuentren en movimiento, todo esto en tiempo real, por lo cual es indispensable mantener un sistema de comunicación estable entre la aeronave y la estación terrena, esto genera la necesidad de contar con un sistema web para la correcta visualización de los datos requeridos.

En el siguiente documento se presentará el desarrollo de un sistema de web para la visualización en tiempo real de las aeronaves conformado por dos módulos diferentes, los cuales serán incorporados para trabajar en conjunto con el módulo para la comunicación en tiempo real por chat mediante el uso de SMS.

Se propuso desarrollar e implementar un sistema de geolocalización en tiempo real para la Fuerza Aérea Mexicana (FAM) el cual facilitará la correcta visualización de las aeronaves en vuelo, así como la ubicación de las diferentes estaciones terrenas, ofreciendo la capacidad de seleccionar los mapas requeridos por el usuario.

1.1 Antecedentes

González Julia ,2021, “*Desarrollo de una aplicación web para el tratamiento de datos de navegación de una aeronave no tripulada*” resuelve la problemática de la falta de una aplicación para gestionar los datos de vuelos que realizan varios drones; donde su principal objetivo es desarrollar una aplicación web para gestionar los datos de geolocalización almacenados a tiempo real de todos los dispositivos compatibles de los que se disponga, el método de solución consiste en:

- Desarrollo de Backend, las tecnologías que se encuentran en esta parte son:
- NodeJS, Express y la base de datos que en este caso es MongoDB.
- Desarrollo del sistema que funcione como middleware; se utiliza Express.
- Desarrollo de Frontend; Angular es la tecnología que se encuentra en este apartado.
- Programación del transceptor; se utiliza IDE de Arduino.
- Pruebas.

Junto con las herramientas: Angular 5, NodeJS, Express, MongoDB, Google Maps, Arduino, chip receptor NEO 6M y Módulo SIM900. Los resultados alcanzados son [1]:

- Registrar sus propios dispositivos.
- Visualizar todos los datos almacenados por dichos dispositivos.
- Se muestran estadísticas de los datos georreferenciados y almacenados para cada dispositivo.
- Se observa en tiempo real la trayectoria realizada por uno de los dispositivos pintándola sobre un mapa.

En el año 2020, Fabricio Andréi Garibay Ornelas con su trabajo titulado “*Diseño e Implementación de un Asistente Virtual (Chat Bot) para Ofrecer Atención a los Clientes de una Aerolínea Mexicana por Medio de sus Canales Conversacionales*” la pregunta general de investigación del presente trabajo cuestiona cómo se diseña e implementa un asistente virtual (Chat Bot) para ofrecer atención a los clientes de Club Premier —programa de recompensas propiedad de Premier Loyalty & Marketing—, por medio de sus canales conversacionales en Facebook, WhatsApp y web. Por tanto, el objetivo principal será explicar el proceso de diseño e implementación de un proyecto laboral, que consiste en el desarrollo de un chatbot para ofrecer atención a los clientes de Club Premier, por medio de sus canales conversacionales en las redes mencionadas anteriormente.

Para poder brindar un estimado de los resultados esperados con la implementación de este proyecto, se retomaron algunos datos de experiencias previas y otros derivados de la aplicación inicial de pruebas de funcionamiento del chatbot. De tal modo, en este apartado se expondrán algunas estadísticas sobre los porcentajes de retroalimentación recabados, la cantidad de calificaciones positivas y negativas, la frecuencia de las interacciones con el Bot y las veces en las que éste transfirió al cliente con un asistente humano [2].

Gurumendi Ricardo, 2019, "*Diseño e implementación de dispositivo de rastreo satelital inteligente GPS mediante microcontrolador programable con activación de apagado automático mediante zonas poligonales en Google Maps*", resuelve el problema de no contar con un sistema para rastreo satelital en tiempo real de la ubicación y estado del vehículo; donde su principal objetivo es diseñar e implementar un Dispositivo GPS/GSM/GPRS mediante microcontrolador programable que permita controlar el apagado del motor vehicular por medio de una aplicación móvil desarrollada.

Los resultados obtenidos son servidor web, que permitirá la visualización del historial y recorrido en una plataforma web administrable. La aplicación muestra su marcador en movimiento, el vehículo se detiene en un semáforo, La aplicación indica que el vehículo hizo una parada, la aplicación notifica que el vehículo se encuentra en zona de riesgo mediante push y notifica a la aplicación móvil, el vehículo se encuentra dentro de la zona poligonal y se apaga, mediante un mensaje SMS al dispositivo GPS, el dispositivo es capaz de receptar hasta 21 satélites lo que permite tener mayor precisión en la ubicación geográfica [3].

En el año 2019, Ascencio Vargas Jairo Joel con título "*Diseño de un chatbot para mejorar la calidad de servicio al cliente en la empresa Fans Store S.A.C*", Lima – 2019, En el mundo, varias empresas de talla mundial están en proceso de mejora continua, saben que el mundo es cambiante; antes las ventas eran personales ahora no es necesario tener tiendas físicas para llegar a tu cliente. Los tiempos cambian y deben estar acorde a la vanguardia. El internet obliga al empresario a brindar nuevas formas de atraer a un cliente expectantes, viéndolo como un ser que siente y con necesidades; mas no como comprador que te genera ganancias. Falcón (2017) refiere que queda mucho por desarrollar a nivel de experiencia de usuario; es decir en las plataformas e-commerce, el cual puede representar una oportunidad de diferenciación con sus competidores.

La solución para el chat Bot planteado tiene como objetivo mejorar la calidad de servicio que se ofrece al cliente en base a sus funcionalidades, y las mejoras en el proceso de este; obteniendo en base a un flujo proyectado de 3 años; un VAN de 10066.82118 y un TIR de 54%; en el cual el VAN es positivo por lo cual conviene realizar la inversión, es decir en el futuro dará resultados; por otra parte, el TIR es mayor a la tasa de préstamos anual; por lo que es viable. Por lo que el proyecto brinda beneficios para la empresa [4].

Luis Felipe Garcia Reina, 2018, *“Asistente Virtual Tipo Chat Bot”*, El Colegio Angloamericano ubicado en el norte de la ciudad de Bogotá, actualmente gestiona la atención de sus clientes mediante el sistema “Comunidad Virtual”, desarrollado y en continua actualización por el departamento de sistemas, el cual administra y mantiene un desarrollo a la medida de las necesidades de la institución.

Gracias al estudio realizado a la corporación San Isidro colegio Angloamericano, fue posible caracterizar y comprender el proceso de gestión de calidad en cuanto a los mensajes recibidos diariamente, se logró establecer lo complejo y manual del proceso, razón por la cual fue necesario hacer un seguimiento directo desde que leen los mensajes por parte del coordinador de calidad, hasta su respuesta final. Con esto fue posible establecer el flujo que debe seguir el asistente virtual de tipo Chat Bot para no afectar las condiciones iniciales bajo las cuales se realiza el proceso de gestión de PQR’s en la corporación. De igual manera la creación de peticiones a diferentes prestadores de servicios permitió determinar la gestión regular que se realiza para la solución de PQR’s, y con esto fue posible establecer una estructura general para almacenar la información [5].

En el año 2017, Anchundia Junior, utiliza las herramientas: GPS del smartphone, API de Google, Android Studio, API de FireBase, MySQL, Google Cloud Messaging; para solucionar el problema de que se presentan varias dificultades a la hora de solicitar vehículos y asignar carreras, la falta del uso de un sistema informático que reemplace los procesos tradicionales y facilite al cliente el proceso de solicitud de taxis, donde el objetivo es desarrollar e implementar un sistema de rastreo vehicular, basado en el GPS de Google y la plataforma de FireBase usando como base de datos MySQL para optimizar los procesos efectuados por los clientes, taxistas y administradores. El método de solución consiste en:

- Diseño Lógico; consiste en diseñar la base de datos en MySQL.
- Asignación de roles de usuario.

- Diseño GUI cliente calificaciones, administrador pantalla de inicio.
- Administrador panel y rastreo de taxis.

Los resultados obtenidos son: mejora en la forma y el medio de obtener el servicio de taxi referente a las ubicaciones y tiempos de cada usuario, dando agilidad, facilidad y confiabilidad en el servicio, tanto para los usuarios como para los taxistas [6].

Abbud José, 2015, "*Localización y seguimiento de aeronaves mediante sistemas de multilateración de área extensa*" resuelve la problemática de las aeronaves no siguen rutas directas, debido a la existencia de zonas reservadas de espacio aéreo, principalmente para fines militares. Estas restricciones impiden la creación de trayectorias óptimas, disminuyendo la capacidad de las rutas aéreas y aumentando los costes (tiempo y combustible); donde su principal objetivo es efectuar el seguimiento de las aeronaves, corrigiendo los efectos de las principales distorsiones que afectan a la medida de TOA (la refracción troposférica) y el error de sincronismo. Junto con las herramientas multilateración de área extensa (WAM), CNS/ATM consiste en los sistemas de comunicación, navegación y vigilancia que emplean tecnología digital, incluyendo sistemas de satélites junto con varios niveles de automatización, centro de procesado que calcula la posición de la aeronave basándose en la diferencia de tiempos de llegada (TDOA) de dicha señal en cada antena, bloque demodulador ("Down converter"), Método algebraico (Bancroft), Método algebraico (Chan, Ho). Los resultados alcanzados son [7]:

Sistema que permite efectuar el seguimiento de las aeronaves, corrigiendo los efectos de las principales distorsiones que afectan a la medida de TOA:

- La refracción troposférica y el error de sincronismo.
- Triplicar la capacidad del espacio aéreo.
- Reducción a la mitad los gastos en gestión de tráfico aéreo.
- Mejorar la seguridad en un factor 10.
- Reducción en el impacto medioambiental de cada vuelo en un 10%.

1.2 Justificación

La presente investigación se enfoca en la implementación de un sistema de georreferenciación para flotillas de aeronaves incorporando un Servidor de mapas Opensource escrito en Java que permite ver, editar y compartir datos geoespaciales. Usando estándares abiertos establecidos por el Consorcio Geoespacial Abierto (OGC).

El presente proyecto propone el aprovechamiento de la tecnología de rastreo mediante la tecnología GPS/GSM con el fin de ayudar al controlador de la aeronave a ubicar en un mapa de alta resolución la trayectoria, tiempo y ubicación actual donde se encuentra operando.

Efectivamente, la principal utilidad del sistema de rastreo es la de conocer y visualizar en una plataforma web la trayectoria y posición en la que se encuentra operando cada uno de las aeronaves y acceder fácilmente a datos de interés, establecer un área geográfica de interés, que le permitirá recibir notificaciones en tiempo real cada vez que las aeronaves, entre o salga de dicho territorio. Además, permitirá el intercambio de información en tiempo real, sin importar la ubicación física de las pilotos que participan en la conversación. Creando un método directo y económico de comunicación. El sistema de rastreo también tiene múltiples aplicaciones como herramienta de seguridad personal, medidas antirrobo o su integración en alarmas, etc.

El impacto tecnológico se enfoca en la utilización del Framework Angular, ya que, en comparación a Frameworks similares los procesos de carga son más rápidos, fácil mantenimiento y escalado del sistema e incorpora un entorno para la elaboración de pruebas.

El proyecto también cuenta con un amplio impacto socioeconómico, ya que se enfoca principalmente en brindar servicio a la FAM, con la utilización de software que no requieren la compra de una licencia para su uso.

Cabe mencionar que el proyecto a desarrollarse posee bajo impacto ambiental, puesto que no emite señales que afecten al medio ambiente, por lo contrario, aprovecha las señales de acceso público que emiten los satélites geoestacionarios hacia la tierra.

El siguiente desarrollo aporta una aplicación altamente personalizada a la necesidad de mantener una comunicación constante con las aeronaves en todo momento, haciendo uso

de las tecnologías más recientes lo cual proporciona una mayor estabilidad. Todo esto sin la necesidad de que el dispositivo inteligente tenga una conexión a internet.

Los beneficios en el desarrollo de este proyecto serán los siguientes:

- Confidencialidad en datos, ya que se tendrá el control total de la información recibida y enviada, garantizando la seguridad de los datos sensibles al no existir software de terceros.
- Estabilidad en la aplicación, al ser desarrollada bajo una arquitectura estable y altamente escalable la cual permitirá agregar funciones en un futuro en caso de ser requeridas.
- Al ser una aplicación multiplataforma se determina usar las siguientes tecnologías:
 - Leaflet al ser una librería web especializada en cartografía digital.
 - TypeScript como lenguaje principal de desarrollo.
 - Angular como framework de desarrollo web.

Esto debido a la decisión de contar con una aplicación robusta pero rápida y fácil de usar para el usuario final.

Este trabajo surge de la necesidad de tener comunicación por enlace de datos a través de SMS, capaz de transmitir información digital de máquina a máquina empaquetando los datos en un formato PEDAT.

La importancia de esta investigación surge en la necesidad de la creación de un canal de comunicación seguro y fiable para SIGESAFAM, que ayudara a futuras generaciones de ingenieros que necesiten una retroalimentación.

Se necesita la capacidad de transmitir de manera segura y controlada a través del medio de comunicación que aseguren la transferencia de datos entre los componentes aéreos y terrestres con un mínimo del 60% de continuidad en el servicio, durante una operación aérea.

El impacto a nivel tecnológico se verá reflejado en el método de transmisión de información, al ofrecer una alternativa en cuanto a la comunicación entre dispositivos haciendo uso de tecnologías anteriormente mencionadas, así como son los SMS aportando la capacidad de transmitir la información recuperada de un módulo GSM, GPS 808.

1.2 Objetivos

1.3.1 General

Desarrollar una aplicación multiplataforma de navegación, rastreo y comunicación aeronáutica capaz de funcionar sin internet. Usando las últimas tecnologías para el desarrollo cumpliendo con los lineamientos necesarios en cuanto a fiabilidad, seguridad y rendimiento. Además, elaborar un plan de pruebas para garantizar la calidad del sistema e instalar configurar y evaluar las funcionalidades de los servidores cartógrafos Opensource.

1.3.2 Específicos

- Analizar requerimientos y necesidades establecidas por la Fuerza Aérea Mexicana (FAM).
- Investigar herramientas de desarrollo eficientes para cumplir con los requerimientos establecidos.
- Desarrollar módulos visualizadores gráfico de mapas y marcadores en tiempo real Y un módulo de chat táctico con diferentes capacidades y funciones.
- Desarrollar una base de datos relacional para satisfacer las necesidades de la empresa y de la aplicación.
- Desarrollar un API que se encargue de la comunicación de la base de datos y la aplicación web.
- Integrar aplicación web al módulo encargado de enviar y recibir la información, realizar pruebas de mantenimiento.
- Realizar pruebas unitarias de los módulos, componentes, servicios etc. del sistema

1.4 Alcances y limitaciones

1.4.1 Alcances

- La UI de la ETSA debe ejecutarse de manera fluida y estable.
- Capacidad de visualizar objetos de emisión, posición relativa de otras aeronaves con el sistema, capacidad de visualizar el perfil del terreno en línea de vuelo, capacidad de visualizar el perfil del vuelo en tiempo real.
- La ETSA debe procesar mensajes, comandos y posiciones que se reciben o se envían a los pilotos, a través de una Interfaz de Usuario (UI) que se comunica con el servidor del SEDT.
- La UI debe poder configurarse para trabajar en dos idiomas.
- La ETSA debe contar con una interfaz gráfica ergonómica para el manejo de mensajes por parte del controlador aéreo (usuario) que recibe o envía a las aeronaves
- La ETSA debe contar con una interfaz gráfica ergonómica para la visualización de las aeronaves por parte del controlador aéreo (usuario) en tiempo real.
- La ETSA debe contar con una herramienta para consultar información de valor militar.
- La ETSA debe ser capaz de geo posicionarse a sí misma.
- La UI debe mostrar la posición y trayectoria de las aeronaves seleccionadas en todo momento
- La UI debe superponer información de valor militar sobre capas cartográficas

1.4.2 Limitaciones

- Utilizar el formato PEDAT para los archivos de lectura/escritura; implementar un temporizador para la lectura de cada línea del archivo fuente en un intervalo tiempo predeterminado (1 segundo).
- El software de la ETSA debe estar apegado a la normativa correspondiente.
- Aplicación no dependiente del uso de internet.
- Uso de software sin licenciamiento.

- Servidor de mapas no optimizado.
- Hardware incapaz de soportar toda la carga de trabajo.
- Servicio de comunicación ineficiente.

1.5 Hipótesis

La aplicación multiplataforma de navegación, rastreo y comunicación aeronáutica será capaz de operar de manera fiable y segura garantizando comunicación con el operador de la aeronave sin Internet.

1.6 Planteamiento del problema

Al paso del tiempo las tecnologías van creciendo exponencialmente por lo que se necesita actualizar en cada caso posible, uno de estos casos es la comunicación en tiempo real. Actualmente existen varios sistemas y subsistemas con la característica de comunicaciones en tiempo real, pero son altamente costosos y generalizados, es decir, son sistemas que no se pueden modificar ciertas características por lo que no son fiables al momento de la seguridad de los datos.

SIGESAFAM solo cuenta con la comunicación por medio de radiofrecuencia, tecnología que con el paso del tiempo se hará obsoleta, por lo tanto, necesitan de una actualización en su forma de comunicarse. Incluso no cuentan con una aplicación web donde se puedan mandar mensajes de texto sin necesidad de internet. En los aplicativos de geolocalización de SIGESAFAM no se cuenta con un chat en tiempo real, tampoco se tiene una aplicación para transmitir datos, posiciones, estados y comandos, no se tiene un canal seguro y confiable para la transmisión de estos datos de forma dinámica.

SEGESAFAM no tiene una base de datos relacional para toda su información, así como una API especializada en la misma que maneje todas las peticiones de los aplicativos futuros, sin la consistencia y rapidez de su manejo es imposible tener todo ordenado y accesible. Otra necesidad es la capacidad de analizar, filtrar, consultar, interpolar, proyectar y cuantificar datos de valor militar (número de operaciones realizadas, horas de vuelo, tipo de misión, tiempos estimados en ruta, tiempos estimados de arribo), derivados de las posiciones de las aeronaves en cada segundo, durante un día de operación.

Una de las necesidades más importantes para la Secretaría De Defensa Nacional (SEDENA) es mantener la comunicación constantes con las aeronaves, teniendo acceso a la ubicación actual, así como la capacidad de visualizar el rastro en caso de que se encuentren en movimiento, todo esto en tiempo real, por lo cual es indispensable mantener un sistema de comunicación estable entre la aeronave y la estación de control, esto genera la necesidad de contar con un sistema web para la correcta visualización de los datos requeridos.

En el siguiente documento se presentará el desarrollo de un sistema de web para la visualización en tiempo real de las aeronaves conformado por tres módulos diferentes, los cuales son:

- Geolocalización en tiempo real de las aeronaves.
- Visualización de posición y rastro de vehículos en el mapa seleccionado
- Módulo para la comunicación en tiempo real por chat mediante el uso de SMS.

Se propuso desarrollar e implementar un sistema de geolocalización en tiempo real para la Fuerza Aérea Mexicana (FAM) el cual permitirá la correcta visualización de las aeronaves en movimiento, así como la ubicación de las diferentes estaciones de control, ofreciendo la capacidad de seleccionar los mapas requeridos por el usuario.

El ETSA (Estación Terrena de Supervisión Aérea) requiere como función principal el facilitar una interfaz gráfica basada en mapas geográficos al controlador de misión para observar los datos de la misión y las aeronaves involucradas en la misma. Esta misma requiere comunicación con las aeronaves mediante el chat táctico.

Al no contar con seguridad en el sistema, se requiere autenticación por parte del controlador, por lo que se contará con roles, permisos y privilegios. Por lo tanto, deberá incluirse una interfaz para administrar las bases de datos del servidor del SEDT, tal como altas, bajas y cambios.

Con los datos obtenidos del servidor, las ETSA deben proporcionar la información de manera visual al controlador aéreo de seguimiento, en forma de gráficas e indicadores, numéricos o de texto (mensajes del Chat Táctico). Esta información debe estar organizada en una interfaz gráfica, con secciones, de manera ergonómica y funcional.

Mediante su interfaz gráfica, el controlador aéreo de seguimiento también debe contar con controles para procesar la información de acuerdo a las necesidades de la misión, como dar el contexto de las posiciones y trayectorias con la selección de diferentes tipos de mapas, o filtros para la visualización de determinadas aeronaves.

Finalmente, toda la información relacionada a cada misión presente se debe almacenar de manera local, para garantizar que esté disponible en caso de una falla con el servidor principal, contemplando incluso que, en caso de una falla catastrófica con el servidor, pueda realizarse una conexión directa al TDT del SEDT para que cualquiera de las ETSA pueda funcionar como servidores de respaldo, con una capacidad limitada al hardware de la misma.

La Secretaría De La Defensa Nacional (SEDENA) no cuenta con un sistema seguro de control de vuelo para la geolocalización y seguimiento de las rutas en tiempo real de las aeronaves, así como la capacidad de mantener una comunicación constante que garantice la privacidad en el manejo de información y un servicio capaz de interpretarla. Actualmente se tiene un módulo electrónico prototipo para la transferencia de información con problema de deficiencia en la precisión de los datos recibidos, pérdida de cobertura del vuelo en ruta además de un software desarrollado para pruebas el cual es poco eficiente ya que carece de una arquitectura sólida por lo cual se vuelve lento, inseguro e inestable. No se tiene el hardware necesario para cumplir con las necesidades óptimas para la transmisión de datos entre la aplicación y el sistema de comunicación por SMS (Short Message Service). Actualmente en el mercado digital ya existen aplicaciones para la gestión del tráfico aéreo, pero estas carecen de confianza por lo cual al ser una parte importante para el control e implementación de planes nacionales genera incertidumbre al no tener conocimiento de que hacen con la información recabada, siendo limitadas en la elección de mapas de vuelo y dependientes de una conexión a internet.

Capítulo II

Marco teórico

En este capítulo se abordarán los diferentes conceptos involucrados en la presente investigación para el desarrollo de una aplicación web multiplataforma en Angular, la comunicación de dicha aplicación con los servicios de NodeJs para la visualización y recepción de los datos obtenido del transceptor de la FAM. También se presentan los temas relacionados con el almacenamiento de la información geoespacial en una base de datos y el servidor de mapas.

2.1 Gestor de base de datos

Un sistema gestor de base de datos (SGBD), o también llamado DBMS por sus siglas en inglés, es un software que se utiliza para acceder, extraer y administrar datos almacenados en una fuente o base de datos, los usuarios tienen acceso a esta información usando herramientas específicas de consulta y generalmente se accede a los datos mediante lenguajes de consulta como lo es SQL (Structured Query Language) [8].

2.1.1 MariaDB

MariaDB Server es una de las bases de datos relacionales de código abierto más populares. Está hecho por los desarrolladores originales de MySQL y se garantiza que seguirá siendo de código abierto. Es parte de la mayoría de las ofertas en la nube y es el predeterminado en la mayoría de las distribuciones de Linux [9].

2.2 Servidor web

En cuanto a hardware, un servidor web es una computadora que almacena los archivos que componen un sitio web (ej. documentos HTML, imágenes, hojas de estilos CSS y archivo JavaScript) y los entrega al dispositivo del usuario final. Está conectado a internet y es accesible a través de un nombre de dominio como mozilla.org.

En cuanto a software, un servidor web tiene muchas partes encargadas del control sobre cómo tienen acceso los usuarios a los archivos, por lo menos un servidor HTTP. Un servidor HTTP es una pieza de software que comprende URLs (direcciones web) y HTTP (el protocolo que tu navegador usa para ver las páginas web) [10].

2.2.1 Apache HTTP Server

Apache HTTP Server Project es un esfuerzo de desarrollo de software colaborativo destinado a crear una implementación de código fuente robusta, de calidad comercial, con funciones y disponible de forma gratuita de un servidor HTTP (Web). El proyecto es administrado conjuntamente por un grupo de voluntarios ubicados en todo el mundo, que utilizan Internet y la Web para comunicarse, planificar y desarrollar el servidor y su documentación relacionada [11].

2.3 Teselas Vectoriales

Una tesela vectorial (vector tiles) contiene datos georreferenciados, recortados para facilitar su recuperación. Cuentan con la capacidad de crear mapas online significativamente rápidos y a su vez ofrecen una flexibilidad completa en su diseño. Son los datos vectoriales equivalentes a las teselas de imagen para el webmapping, aplicando las ventajas de estas: desarrollados para guardarse en caché, escalabilidad y servir imágenes de mapa rápidamente [12].

2.3.1 Geoserver

GeoServer le permite mostrar su información espacial al mundo. Al implementar el estándar Web Map Service (WMS), GeoServer puede crear mapas en una variedad de formatos de salida. OpenLayers, una biblioteca de mapas gratuita, está integrada en GeoServer, lo que hace que la generación de mapas sea rápida y sencilla. Se basa en GeoTools, un conjunto de herramientas Java GIS de código abierto, cumple con el estándar Web Feature Service (WFS) y el estándar Web Coverage Service (WCS) que permite compartir y editar los datos que se utilizan para generar los mapas, también utiliza el estándar Web Map Tile Service para dividir sus mapas publicados en mosaicos para facilitar el uso de mapas web y aplicaciones móviles [13].

2.3.2 MapTiler

MapTiler permite “Transformar imágenes en capas de mapas interactivas que se pueden visualizar en los sitios web, que pueden ser mostradas en teléfonos móviles, tablets, dispositivos GPS, mashups de mapas o ser abiertos en software GIS de escritorio, Google Maps o Google Earth”. Es decir, convierte las imágenes de mapas digitalizadas a un formato web y las superpone sobre un mapa actual tipo Google Maps, por citar al más famoso [14].

2.4 JavaScript

JavaScript (JS) es un lenguaje de programación ligero, interpretado, o compilado justo a tiempo (just in time) con funciones de primera clase. Si bien es más conocido como un lenguaje de scripting (secuencias de comandos) para páginas web, y es usado en muchos entornos fuera del navegador, tal como Node.js, JavaScript es un lenguaje de programación basada en prototipos, multiparadigma, de un solo hilo, dinámico, con soporte para programación orientada a objetos, imperativa y declarativa [15].

2.4.1 NodeJs

Node.js es similar en diseño y está influenciado por sistemas como Event Machine de Ruby y Twisted de Python. Pero Node.js lleva el modelo de eventos un poco más allá. Incluye un bucle de eventos como runtime de ejecución en lugar de una biblioteca. En otros sistemas siempre existe una llamada de bloqueo para iniciar el bucle de eventos. HTTP es un elemento destacado en Node.js, diseñado teniendo en cuenta la transmisión de operaciones con streaming y baja latencia. Esto hace que Node.js sea muy adecuado para la base de una librería o un framework web [16].

2.4.2 TypeScript

TypeScript es un lenguaje de código abierto que se basa en JavaScript, una de las herramientas más utilizadas del mundo, al agregar definiciones de tipos estáticos. Los tipos proporcionan una forma de describir la forma de un objeto, proporcionando una mejor documentación y permitiendo que TypeScript valide que su código está funcionando correctamente [17].

2.4.2.1 TypeORM

TypeORM es un mapeador de objetos relacionales (ORM) que se puede ejecutar en plataformas Node.js, Browser, Cordova, PhoneGap, Ionic, React Native, NativeScript, Expo y Electron y se puede utilizar con TypeScript y JavaScript (ES5, ES6, ES7, ES8). Su objetivo es siempre admitir las últimas funciones de JavaScript y proporcionar funciones adicionales que le ayuden a desarrollar cualquier tipo de aplicación que utilice bases de datos, desde

aplicaciones pequeñas con unas pocas tablas hasta aplicaciones empresariales a gran escala con varias bases de datos [18].

2.4.3 Express

Express es una infraestructura de aplicaciones web Node.js mínima y flexible que proporciona un conjunto sólido de características para las aplicaciones web y móviles. Con miles de métodos de programa de utilidad HTTP y middleware a su disposición, la creación de una API sólida es rápida y sencilla [19].

2.4.4 Angular

Angular es un framework Javascript potente, muy adecuado para el desarrollo de aplicaciones frontend modernas, de complejidad media o elevada. El tipo de aplicación Javascript que se desarrolla con Angular es del estilo SPA (Single Page Application) o también las denominadas PWA (Progressive Web App) [20].

Como plataforma, Angular incluye:

- Un marco basado en componentes para crear aplicaciones web escalables.
- Una colección de bibliotecas bien integradas que cubren una amplia variedad de características, que incluyen enrutamiento, administración de formularios, comunicación cliente-servidor y más.
- Un conjunto de herramientas para desarrolladores que le ayudarán a desarrollar, compilar, probar y actualizar su código.

2.4.4.1 Testing

Es un concepto que consiste en ejecutar una serie de scripts que testean una Aplicación como si fuera una persona, devolviendo información muy precisa de aquello que no funciona como se tiene pensado funcionar.

Al iniciar un nuevo proyecto en Angular, nos encontraremos con una carpeta en nuestra carpeta raíz llamada "e2e"; del mismo modo, cada componente, servicio o directiva tiene asociado un archivo ".spec.ts". Todo esto corresponde a archivos necesarios para testing.

Sin embargo, Angular nos provee de dos sistemas distintos de testeo, de acuerdo con nuestras necesidades [21].

2.4.4.1.1 Jasmine

Jasmine es un Behavior-Driven Development Framework para probar código JavaScript. No depende de ningún otro Framework de JavaScript. No requiere un DOM. Y tiene una sintaxis clara y obvia para que pueda escribir fácilmente pruebas [22].

2.4.4.1.2 Karma

karma.js es un test runner, desarrollado por el equipo de Angular, que nos permite automatizar algunas tareas de los frames de tests, como Jasmine. Con Karma podremos testear cada componente de nuestra aplicación, mediante la programación de un test con el archivo "*.spec.ts". El concepto detrás de Karma es poder programar los tests independientes de cada componente o servicio, en lugar de la aplicación entera [22].

2.4.5 Leaflet

Leaflet es la biblioteca JavaScript de código abierto líder para mapas interactivos compatibles con dispositivos móviles. Tiene todas las funciones de mapeo que la mayoría de los desarrolladores necesitan. El folleto está diseñado teniendo en cuenta la simplicidad, el rendimiento y la facilidad de uso. Funciona de manera eficiente en todas las principales plataformas de escritorio y móviles [23].

2.5 Sistema de Posicionamiento global (GPS)

Actualmente, el GPS está en pleno funcionamiento y cumple con los criterios establecidos en la década de 1960 para un sistema de posicionamiento óptimo. El sistema proporciona información precisa, continua, mundial y tridimensional de posición y velocidad a los usuarios con el equipo de recepción adecuado.

El GPS también difunde una forma de Universal Coordinated Time (UTC). La constelación de satélites consta nominalmente de:

- 24 satélites en 6 planos orbitales.
- 4 satélites en cada plano.
- 20200 km de altura.
- 60 grados de inclinación.

Una red mundial de control / monitoreo en tierra monitorea la salud y el estado de los satélites. Esta red también carga la navegación y otros datos a los satélites.

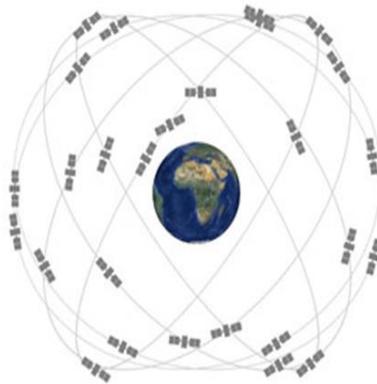
El GPS puede proporcionar servicio a un número ilimitado de usuarios ya que los receptores de usuario operan pasivamente [24].

2.5.1 Segmentos del GPS

2.5.1.1 Segmento espacial

El segmento espacial GPS como se muestra en la Fig.1 consiste en una constelación de satélites que transmiten señales de radio a los usuarios.

Estados Unidos se compromete a mantener la disponibilidad de al menos 24 satélites GPS operativos, el 95% del tiempo. Para garantizar este compromiso, la Fuerza Espacial de EE. UU. Ha estado volando 31 satélites GPS operativos durante más de una década [24].



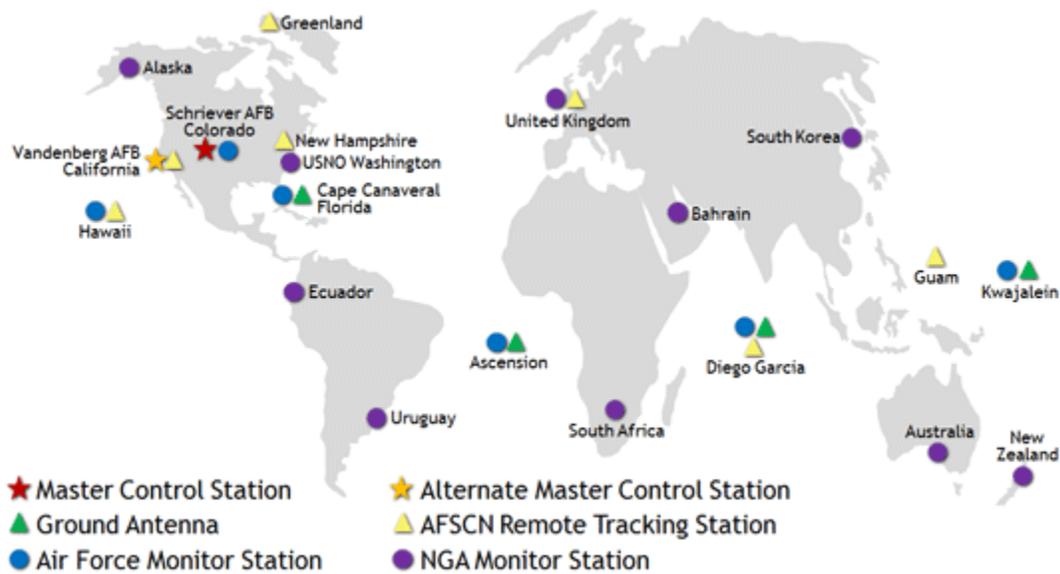
[25]

Fig. 1 Constelación de satélites

2.5.1.2 Segmento de control

El segmento de control GPS como se muestra en la Fig. 2 consiste en una red global de instalaciones terrestres que rastrean los satélites GPS, monitorean sus transmisiones, realizan análisis y envían comandos y datos a la constelación [26].

- El Segmento de Control Operacional (OCS) actual incluye:
- Estación de control maestra
- Estación de control maestra alternativa
- 11 antenas de comando y control
- 16 sitios de monitoreo.



[27]

Fig. 2 Ubicación de las estaciones en el mundo

2.5.1.3 Segmento de usuario

El segmento de usuario consiste en el equipo receptor del GPS que recibe las señales de los satélites del GPS y las procesa para calcular la posición tridimensional y la hora precisa. Es necesario de cuatro satélites para determinar las tres dimensiones de X, Y, Z (posición) y la hora. Los receptores GPS se utilizan para la navegación, posicionamiento, difusión tiempo, y otras investigaciones [24].

- Está formado por los receptores GPS. Sus funciones principales son las siguientes:
- Sintonizar las señales emitidas por los satélites.
- Decodificar el mensaje de navegación.
- Medir el retardo de la señal (desde el transmisor hasta el receptor) a partir de los cuales calculan la posición.
- Presentar la información de la posición en la que se encuentra (en 3D o en 2D).

2.6 Señales GPS

Los satélites del GPS transmiten dos señales de radio de baja potencia, llamadas "L1" y "L2". Cada señal GPS contiene tres componentes de información:

- Un código pseudoaleatorio, que identifica al satélite que transmite su señal.
- Los datos de efemérides de satélite, que proporcionan información sobre la ubicación del satélite en cualquier momento.
- Datos de almanaque, que contiene información sobre el estado del satélite y la fecha y hora actuales.

Para cada satélite, el tiempo es controlado por los relojes atómicos a bordo que son cruciales para conocer su posición exacta [28].

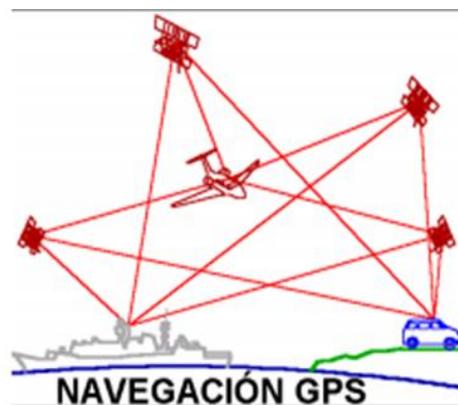
2.7 Determinación de posiciones del GPS

Las posiciones se obtienen mediante la determinación de las distancias a los satélites visibles. Este proceso se conoce como "trilateración". El momento de la transmisión de la señal en el satélite se compara con el momento de la recepción en el receptor. La diferencia de estos dos tiempos nos dice cuánto tiempo tomó para que la señal viajara desde el satélite al receptor.

Si se multiplica el tiempo de viaje por la velocidad de la luz, podemos obtener el rango de distancia, con el satélite. La repetición del proceso desde tres satélites como se muestra en la Fig.3 permite determinar una posición de dos dimensiones en la Tierra (es decir, la longitud y latitud). Un cuarto satélite es necesario para determinar la tercera dimensión, es decir la altura.

Cuanto más satélites son visibles, más precisa es la posición del punto a determinar. Las órbitas de los satélites GPS están inclinadas respecto al ecuador de la Tierra en alrededor de 55° . La distribución espacial de la constelación de satélites permite al usuario disponer de 5 a 8 satélites visibles en cualquier momento. El sistema está diseñado para asegurar que al menos cuatro satélites estarán visibles con una recepción configurada de la señal de 15° sobre el horizonte en un momento dado, en cualquier parte del mundo.

Aunque el GPS puede dar posiciones muy precisas, aún hay fuentes de error. Estos incluyen los errores del reloj, los retrasos atmosféricos, sin saber exactamente dónde están los satélites en sus órbitas, las señales que se refleja de los objetos en la superficie de la Tierra, e incluso la degradación intencionada de la señal del satélite [29].



[30]

Fig. 3 Cuatro satélites calculando las dimensiones

2.7 Receptores GPS

Existen tres tipos de receptores GPS:

- Receptores mono canal. Posee un solo canal de radio y por eso se engancha un satélite a la vez.
- Receptores multicanal. Está conformado por 4 canales, que se usan para sincronizar con las emisiones de otros satélites en forma simultánea.
- Receptores multiplexados. Posee un único canal físico sobre el cual montan varios canales lógicos mediante software, con los que se pueden sincronizar datos de todos los satélites visibles en un tiempo no máximo a los 20 milisegundos.

El receptor GPS tiene dos estados cuando un usuario entra en funcionamiento los cuales son:

- Perdido. El satélite memoriza los datos que estuvo en funcionamiento la última vez y tienen una fecha muy antigua. El receptor busca al satélite que le ofrezca una mejor relación señal a ruido y demodula su mensaje de navegación, consiguiendo de esta forma el almanaque y la referencia temporal. Esto puede llevar varios minutos.
- Memorizado. El receptor comprueba que los datos que tenía almacenados en memoria son válidos y utiliza los mismos satélites que la última vez que fue empleado [26].

2.8 Fuentes de error del GPS

Hasta este momento, hemos asumido que la posición obtenida del GPS es muy precisa y libre de errores, pero existen diferentes fuentes de error que degradan la posición GPS desde algunos metros, en teoría, hasta algunas decenas de metros. Estas fuentes de error son [28]:

- Retraso de la señal en la ionosfera y la troposfera.
- Señal multi ruta, producida por el rebote de la señal en edificios y montañas cercanos.
- Errores de orbitales, donde los datos de la órbita del satélite no son completamente precisos.

- Número de satélites visibles.
- Geometría de los satélites visibles.
- Errores locales en el reloj del GPS.

2.9 Tecnología GSM

La red GSM (Sistema global de comunicaciones móviles) es, a comienzos del siglo XXI, el estándar más usado de Europa. Se denomina estándar "de segunda generación" (2G) porque, a diferencia de la primera generación de teléfonos portátiles, las comunicaciones se producen de un modo completamente digital. En 1982, cuando fue estandarizado por primera vez, fue denominado "Groupe Spécial Mobile" y en 1991 se convirtió en un estándar internacional llamado "Sistema Global de Comunicaciones Móviles".

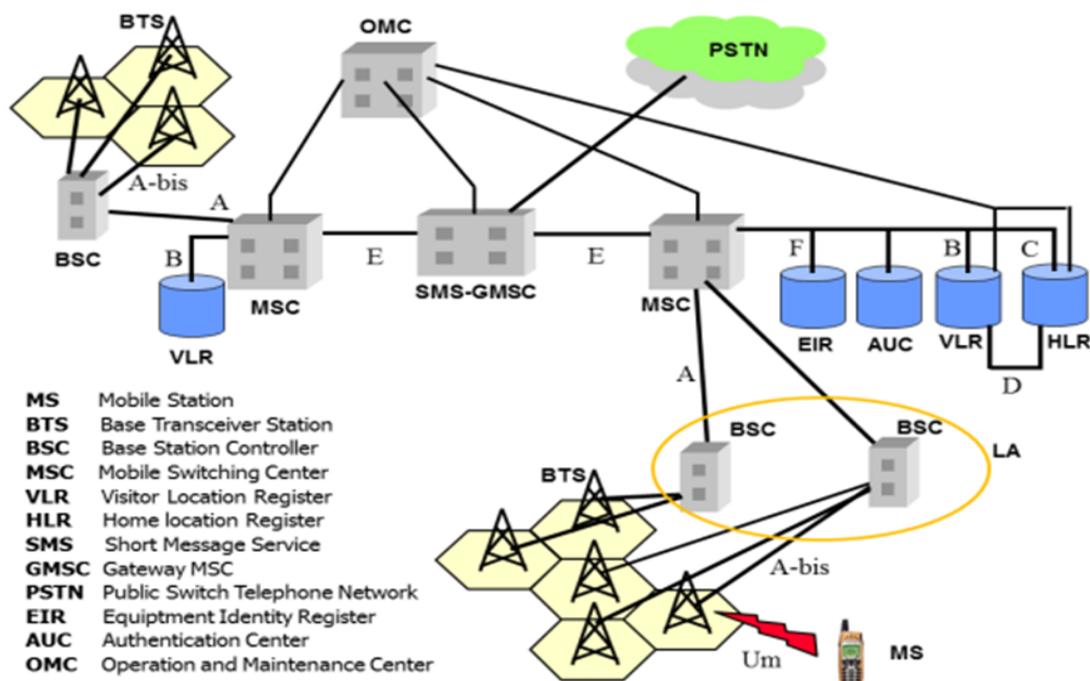
En Europa, el estándar GSM usa las bandas de frecuencia de 900MHz y 1800 MHz. Sin embargo, en los Estados Unidos se usa la banda de frecuencia de 1900 MHz. Por esa razón, los teléfonos portátiles que funcionan tanto en Europa como en los Estados Unidos se llaman tribanda y aquellos que funcionan sólo en Europa se denominan bibanda. El estándar GSM permite un rendimiento máximo de 9,6 kbps, que permite transmisiones de voz y de datos digitales de volumen bajo, por ejemplo, mensajes de texto (SMS, Servicio de mensajes cortos) o mensajes multimedia (MMS, Servicio de mensajes multimedia) [31].

2.9.1 Arquitectura red GSM

La estructura de la red del sistema GSM es una estructura de tipo jerárquico en la que los diferentes elementos están conectados mediante interfaces definidos en el estándar. En la Fig.4 se muestran los diferentes elementos y sus interfaces. A continuación, se enumeran los elementos de la estructura y se muestra el significado de las siglas:

- 1) Estación móvil (MS, "Mobile Station Equipment"): es el dispositivo móvil con el que el usuario se conecta a la red a través del interfaz Um
- 2) Estación Base (BTS, "Base Transceiver Station"): son los equipos de radiofrecuencia, antenas y los equipos electrónicos que dan servicio a una célula o a varias células (célula sectorizada).

- 3) Controlador de estaciones base (BSC, "Base Station Controller"): controla varias estaciones base.
- 4) Área de localización (LA, "Location Area").
- 5) Centro de conmutación de servicios móviles (MSC, "Mobile Service Switching Center"): es el encargado de realizar casi todas las funciones de conmutación.
- 6) Registro de localización local (HLR, "Home Location Register"): es una base de datos que contiene información de todos los usuarios abonados de la red móvil.
- 7) Registro de localización de visitantes (VLR, "Visited o Visitor Location Register"): es una base de datos que contiene información sobre los abonados que se encuentran actualmente en alguna de las LA que forman parte del MSC al que está asociado el VLR.
- 8) Puerta de Enlace o Pasarela-Centro de conmutación de servicios móviles ("SMS-GMSC" Gateway Mobile Service Switching Center): es un MSC especial que se encarga de la comunicación con la red de telefonía fija (PSTN "Public Switching Telephone Network"). También se encarga de la gestión de los SMS
- 9) Centro de operación y mantenimiento (OMC, "Operation and Maintenance Center"): se encarga de las operaciones de mantenimiento y realiza tareas de gestión. Posee conexiones con todos los elementos de la red jerárquica.
- 10) Registro de identificación de registros (EIR, "Equipment Identity Register"): es una base de datos que guarda información sobre los equipos móviles de los abonados a la red.
- 11) Centro de Autenticación (AUC, "Authentication Center"): es una base de datos que también posee equipos para realizar cálculos de autenticación de los usuarios abonados a la red [32].



[33]

Fig. 4 Estructura de la red GSM

2.10 Metodologías de desarrollo de software

Las metodologías de desarrollo de software son un conjunto de técnicas y métodos organizativos que se aplican para diseñar soluciones de software informático. El objetivo de las distintas metodologías es el de intentar organizar los equipos de trabajo para que estos desarrollen las funciones de un programa de la mejor manera posible [34].

2.10.1 SCRUM

Scrum es una de las metodologías ágiles más populares. Es una metodología de adaptación, iterativa, rápida, flexible y eficaz, diseñada para ofrecer un valor significativo de forma rápida en todo el proyecto.

Scrum garantiza transparencia en la comunicación y crea un ambiente de responsabilidad colectiva y de progreso continuo. El marco de Scrum está estructurado de tal manera que

es compatible con los Productos y el desarrollo de servicio en todo tipo de industrias y en cualquier tipo de proyecto, independientemente de su complejidad.

Una fortaleza clave de Scrum radica en el uso de equipos multi funcionales, auto organizados, y con poder que dividen su trabajo en ciclos de trabajo cortos y concentrados llamados "Sprints". proporciona una visión general de flujo de un proyecto Scrum [35].

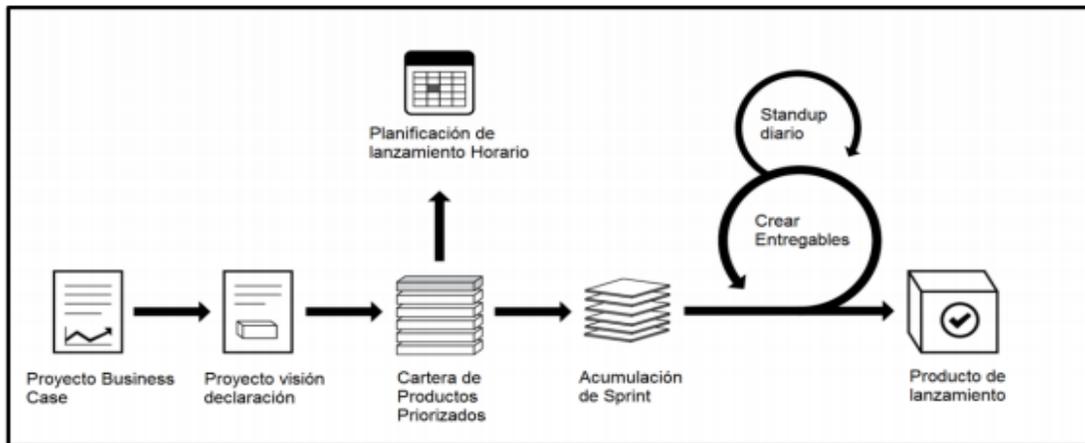


Fig. 5 El ciclo de SCRUM

En Scrum existen eventos predefinidos con el fin de crear regularidad y minimizar la necesidad de reuniones no definidas en SCRUM como se puede observar en la Fig. 5. Todos los eventos son bloques de tiempo (time boxes), de tal modo que todos tienen una duración máxima. Una vez que comienza un Sprint, su duración es fija y no puede acortarse o alargarse. Los demás eventos pueden terminar siempre que se alcance el objetivo del evento, asegurando que se emplee una cantidad apropiada de tiempo sin permitir desperdicio en el proceso [36].

- Planificación del Sprint (Sprint Planning)
- Scrum Diario (Daily Scrum)
- Revisión del Sprint (Sprint Review)
- Retrospectiva del Sprint (Sprint Retrospective)

2.10.1.1 Planificación del Sprint

La planificación de sprints es un evento en scrum que inicia el sprint. El objetivo de la planificación de sprints es definir lo que se puede entregar en el sprint y cómo se conseguirá

ese trabajo. La planificación de sprints se hace en colaboración con todo el equipo de scrum [37].

2.10.1.2 Scrum Diario

El Scrum diario es uno de los 5 eventos de Scrum con un bloque de tiempo de 15 minutos para que los desarrolladores se sincronicen.

Esta reunión diaria se realiza a la misma hora y en el mismo lugar para reducir la complejidad. Aquí se busca la transparencia y la inspección de lo realizado para tener una oportunidad de adaptación para el día siguiente [38].

2.10.1.3 Revisión del Sprint

Durante esta ceremonia se revisa el Incremento, es decir, lo que se realizó durante el Sprint, y se analizan los cambios que tuvo el producto [39].

2.10.1.4 Retrospectiva

El propósito de la retrospectiva es planificar formas de aumentar la calidad y la efectividad. El equipo Scrum inspecciona cómo fue el último Sprint con respecto a las personas, las interacciones, los procesos, las herramientas y su Definición de Terminado. El equipo Scrum identifica los cambios más útiles para mejorar su efectividad. Las mejoras más impactantes se abordan lo antes posible. Incluso se pueden agregar al Sprint Backlog para el próximo Sprint [40].

Capítulo III

Metodología y arquitectura de solución

El presente proyecto se basa en la metodología de desarrollo SCRUM, donde a través de los sprints propuestos se determinó el análisis de requerimientos, diseño de la arquitectura del software, evaluación del módulo, así como la retrospectiva para conocer a detalle las modificaciones.

3.1 Arquitectura de la ETSA

El hardware del Subsistema de la (ETSA) tiene una arquitectura multinivel de distribución física y multicapa de distribución lógica con módulos y componentes reutilizables alojados respectivamente en el backend, frontend y la API. Para organizar los elementos y darle forma a la arquitectura de sistema, se implementó un estándar que define niveles de componentes como se observa en la Fig. 7 en base a los siguientes criterios:

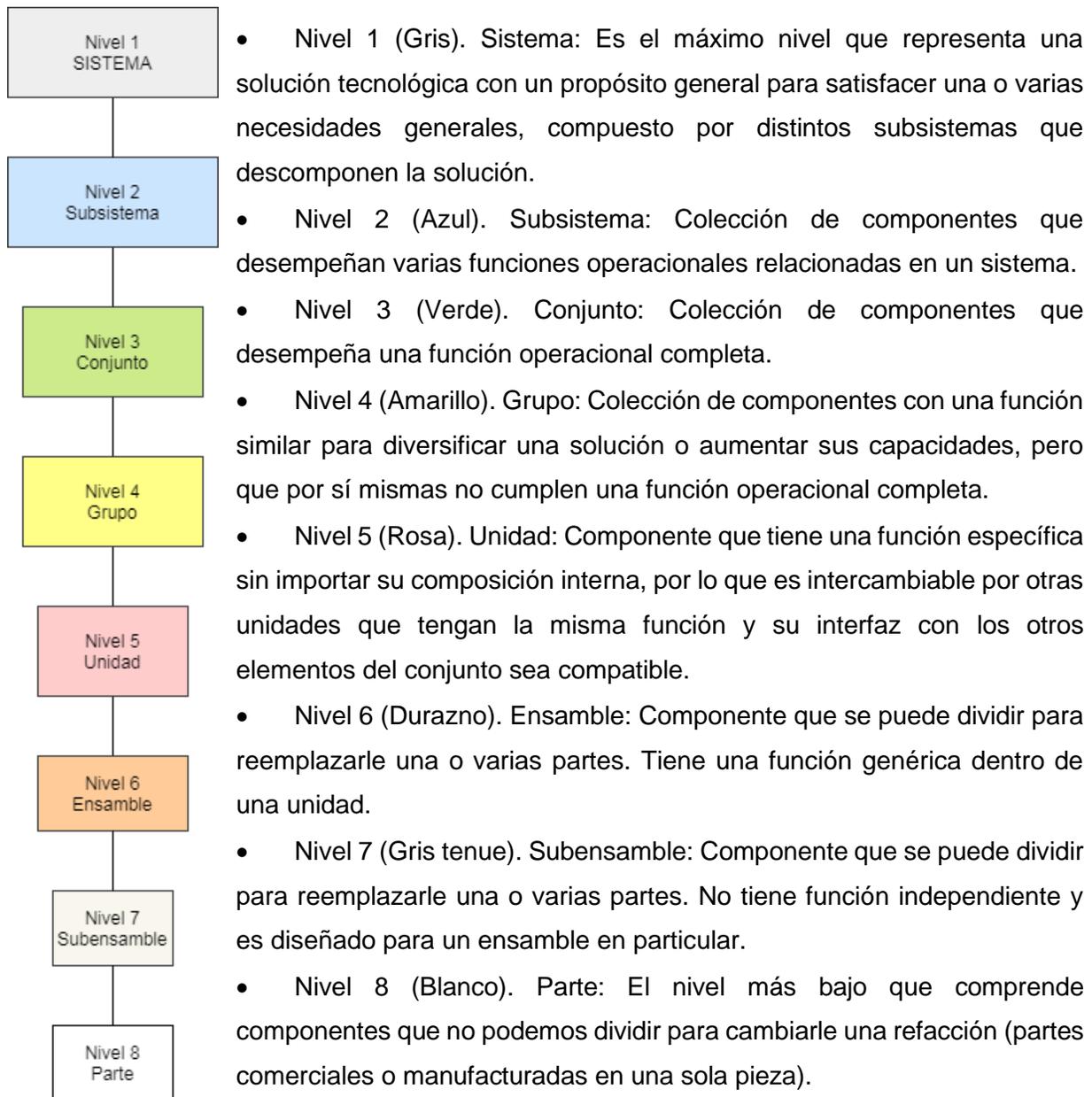


Fig. 6 Diagrama de niveles

Con base a este estándar y a partir de los componentes principales del ETSA (Servidor de mapas geodésicos, Módulo Procesador Visual, Módulo Navegador Web, Módulo del chat táctico, Módulo Gestor de Base de Datos), se puede observar en la Fig. 7 la arquitectura final de la ETSA, la cual está compuesta por:

- **El servidor** de mapas cartográficos digitales en formato de teselas y de la base de datos.
- **Backend** donde se encuentra alojada la API para permitir las peticiones y respuestas entre el servidor y el frontend.
- **Frontend** este último con sus módulos (visualizador geográfico y comunicación enlace de datos a través del chat), componentes y servicios (mapas base, mapas dinámicos, rutas, waypoints, reloj, posición aeronáutica militar, etc.).

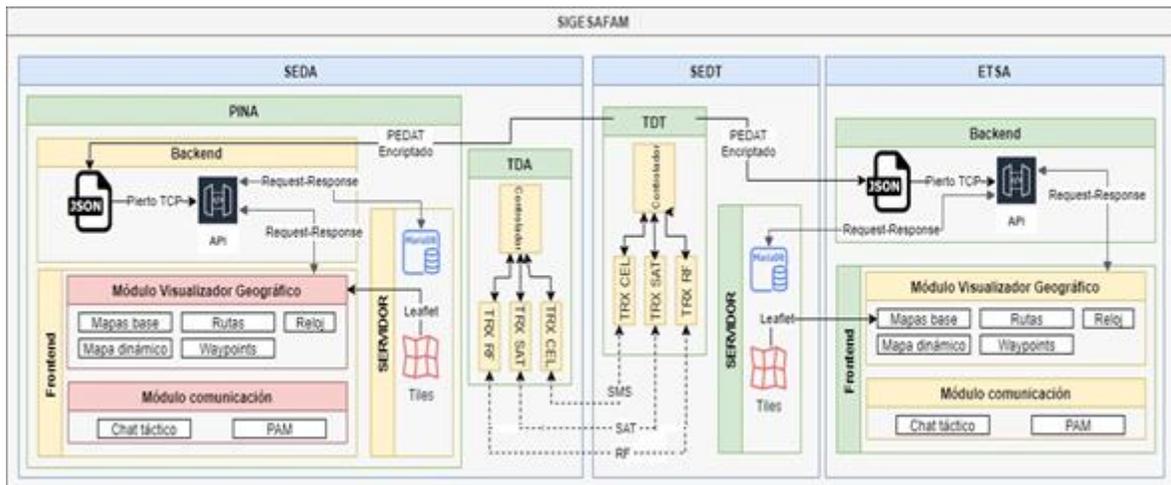


Fig. 7 Arquitectura ETSA

3.2 Herramientas tecnológicas para desarrollo de la ETSA

La Tabla 1 enlista las herramientas de infraestructura, Desarrollo y pruebas de la aplicación

Tabla 1 Kit de herramientas

Herramientas de desarrollo		
Software	Versión	Descripción
Angular	11.0.0	Se uso para el desarrollo web y ejecución de pruebas.

NodeJS	14.17.4	Se uso para desarrollar servicios en la capa del servidor, basado en el lenguaje JavaScript.
MariaDB	10.1.48	Se uso para gestionar la base de datos.
DBeaver	21.0.1	Se uso gestor grafico para la administración de la base de datos del servidor.
Google Chrome	89.0.4389.82	Se usó para ejecutar la aplicación web en el explorador y realizar pruebas.
VSCoDe	1.54.1	Se uso como editor de código fuente.
Software para realizar pruebas		
Software	Versión	Descripción
Jasmine	11.2.7	Se uso para escribir las pruebas unitarias
KARMA	5.0	Se utilizo para gestionar las pruebas y generar reportes
Ponicode	0.0.1-17	Se utilizo para tener un panorama más amplio de las pruebas que se pueden realizar
Herramientas de infraestructura		
Software	Descripción	
Servidor de aplicaciones	Se usó Apache TomCat 7.0 como servidor web.	
Servidor de Base de Datos	Se usó MariaDB como el gestor de bases de datos para almacenar la información.	

3.3 Niveles generales ETSA

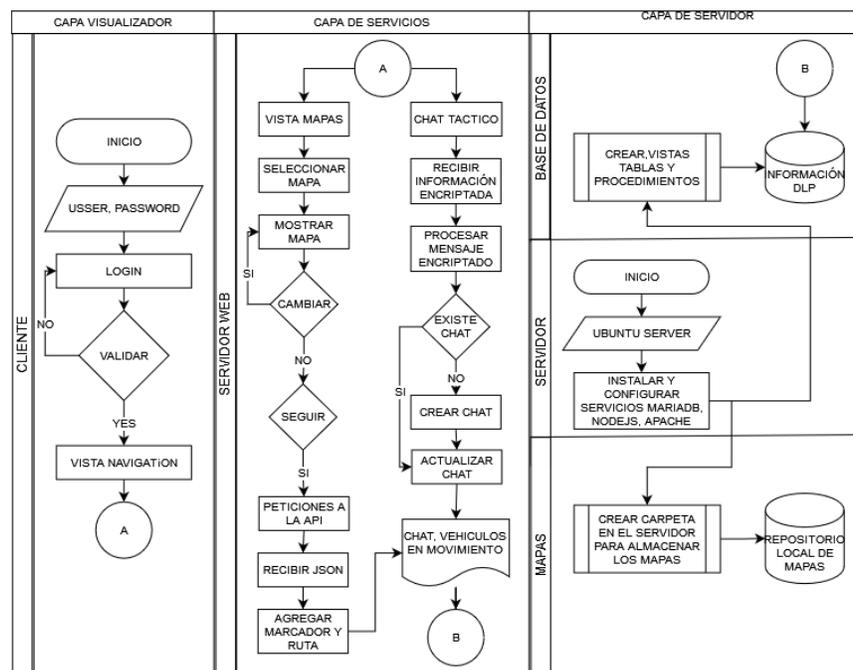


Fig. 8 Niveles generales ETSA

A continuación, se describen los diferentes niveles que requieren para una mayor funcionalidad de la aplicación como se indica en la Fig. 8

- **Nivel de cliente.** La aplicación web funcionará en GNU / Linux y Microsoft Windows; es capaz de autenticar el operador, el objeto y el controlador. Además, mostrará los geo mapas con información relevante (PEDAT), puntos de control y ruta de operación en tiempo real. Incluyendo un chat táctico el cual gestionará las operaciones y la comunicación con el enlace de datos. (Fig. 9)
- **Nivel de servidor web.** Para construir la aplicación con la arquitectura de modelo propuesto, se requieren n capas lógicas y las herramientas de desarrollo que se utilizaron son: Linux (Ubuntu Server), NodeJs (BackEnd), Angular (Frontend), Leaflet plugin para interactuar con mapas rasterizados y vectoriales con sistemas de coordenadas EPSG: 4326 y en caso de requerir un servidor interno en un dispositivo ANDROID, se hace uso de TERMUX el cual nos proporciona una terminal para el dispositivo, habilitando la opción para ejecutar comandos de Linux e instalar aplicaciones y servicios.(Fig. 9)
- **Nivel de datos.** Construye y administra la base de datos con un motor de persistencia utilizando MariaDB a través de un Objeto de Mapeo Relacional (ORM). El cual garantiza alta escalabilidad y facilidad al manipular la base de datos de una manera fácil, proporcionando un sistema de migraciones muy confiable. (Fig. 9)

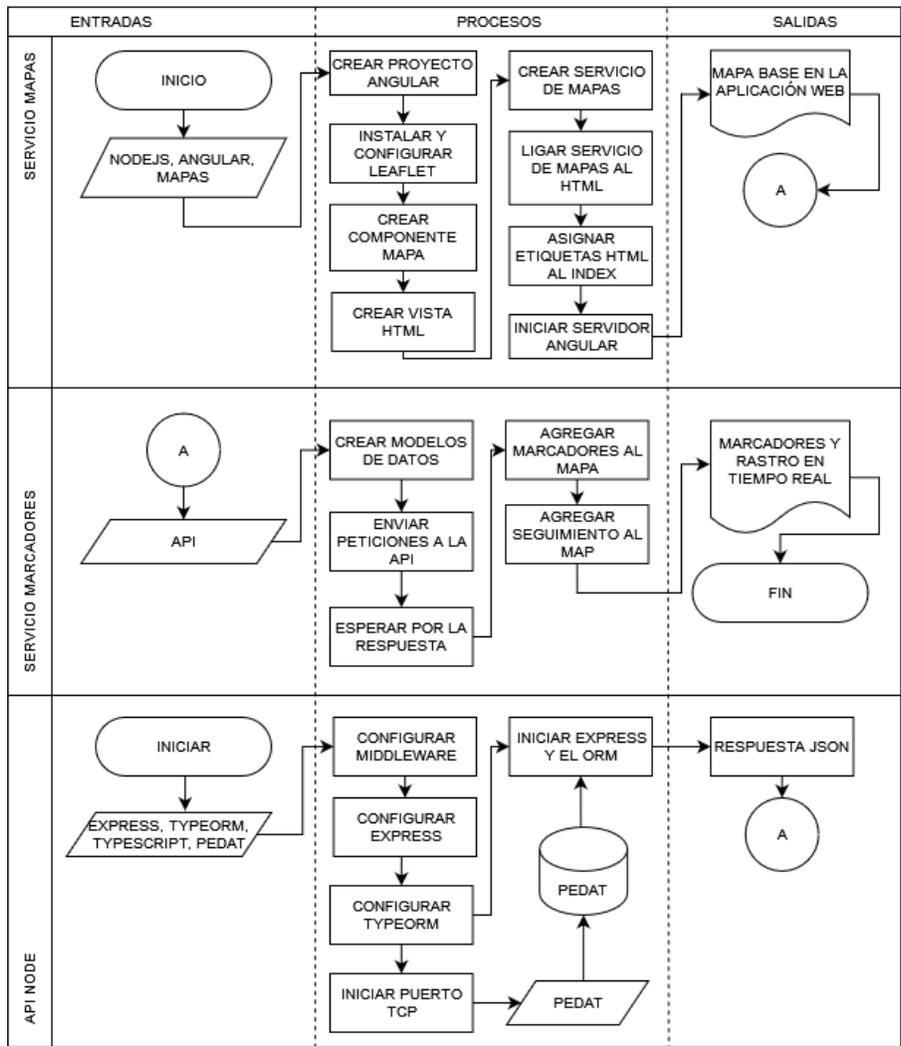


Fig. 9 Diagrama visualizador geográfico

3.4 Chat táctico

Para la funcionalidad del chat táctico se utilizan diferentes servicios para la recopilación de datos, así como datos nuevos (conversaciones, contactos, etc.), todos estos servicios se encuentran corriendo cada segundo para que la visibilidad de los datos sea casi en tiempo real y por ultimo los procesos que ayudaran al usuario al intercambio de mensajes e información, véase Fig. 10.

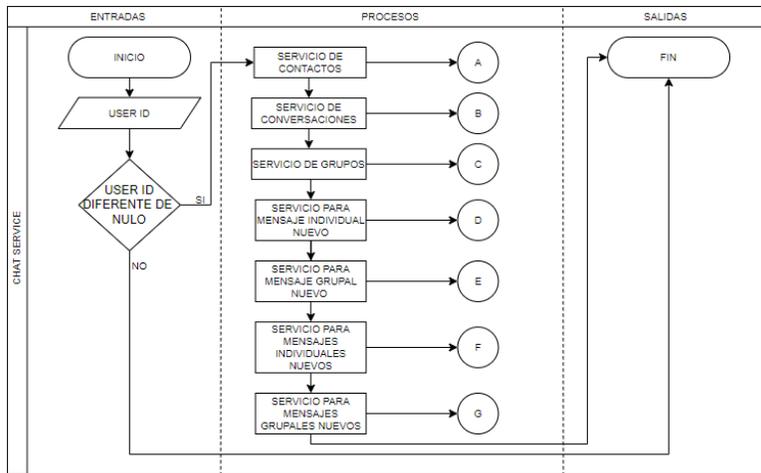


Fig. 10 diagrama general de los servicios para el chat táctico

3.4.1 Servicio de contactos

La Fig. 11 pertenece al servicio de contactos, para obtener la lista de contactos que se agreguen a la base de datos en tiempo real.

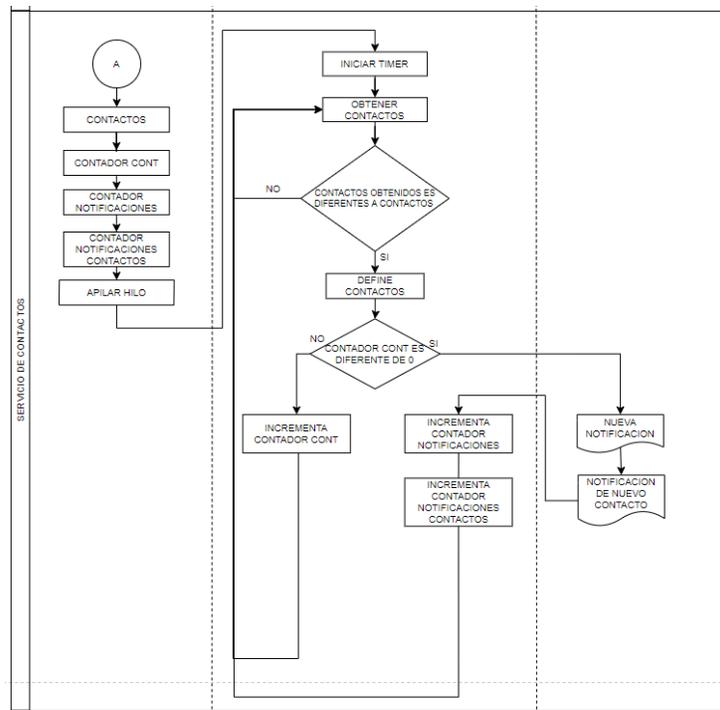


Fig. 11 Servicio de contactos

3.4.2 Servicio de conversaciones

Se utiliza para obtener el historial de conversaciones (Fig. 12), así como escuchar a la base de datos por conversaciones nuevas y notificar al usuario de estas, cabe destacar que este proceso siempre estará corriendo por lo que no tiene final.

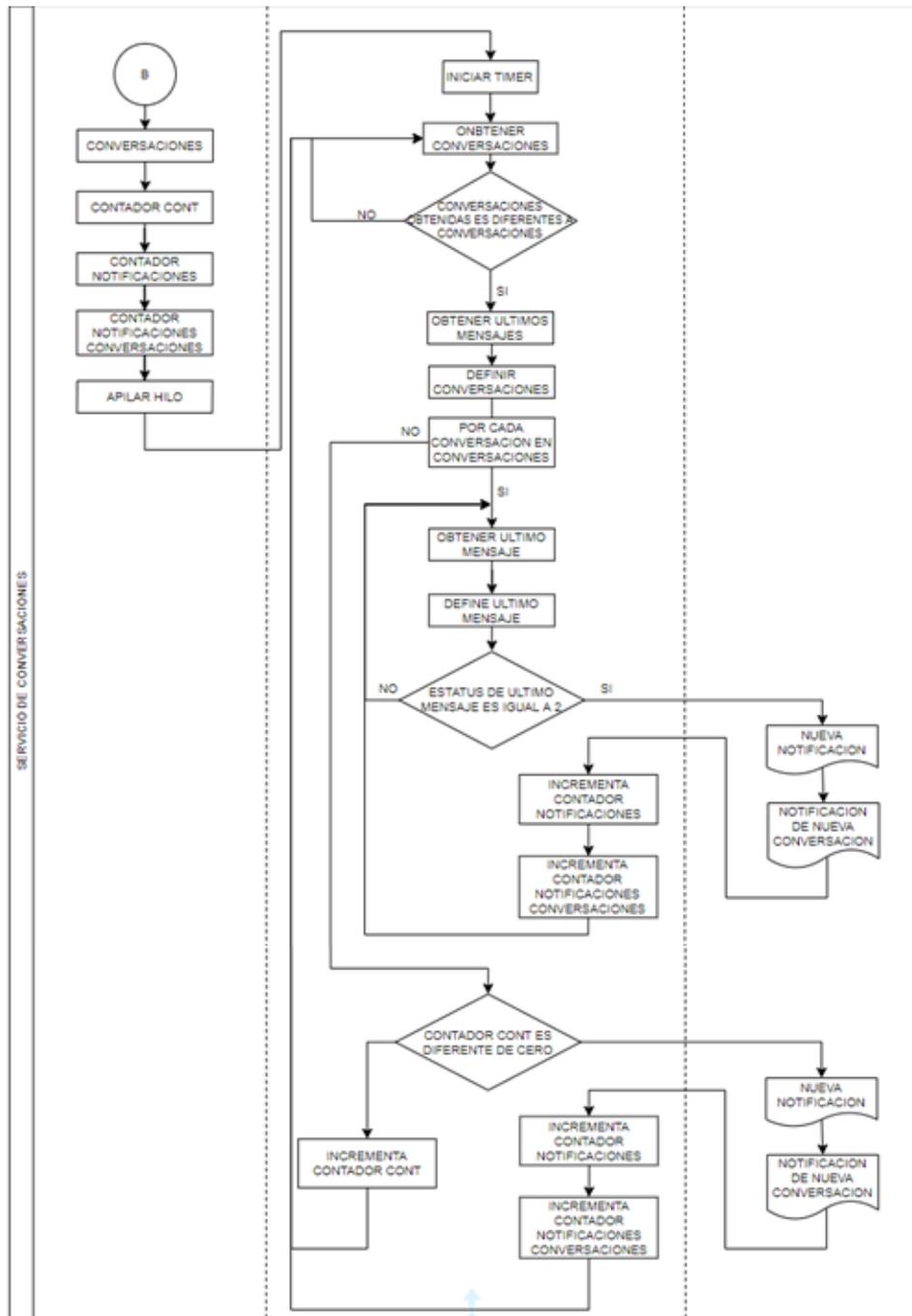


Fig. 12 Servicio de conversaciones

3.4.3 Servicio de grupos

Este servicio nos ayudará a obtener la lista de grupos y su información, también nos ayuda a obtener y notificar al usuario de grupos nuevos, el mapa de este proceso, es más complejo que los demás (Fig. 13).

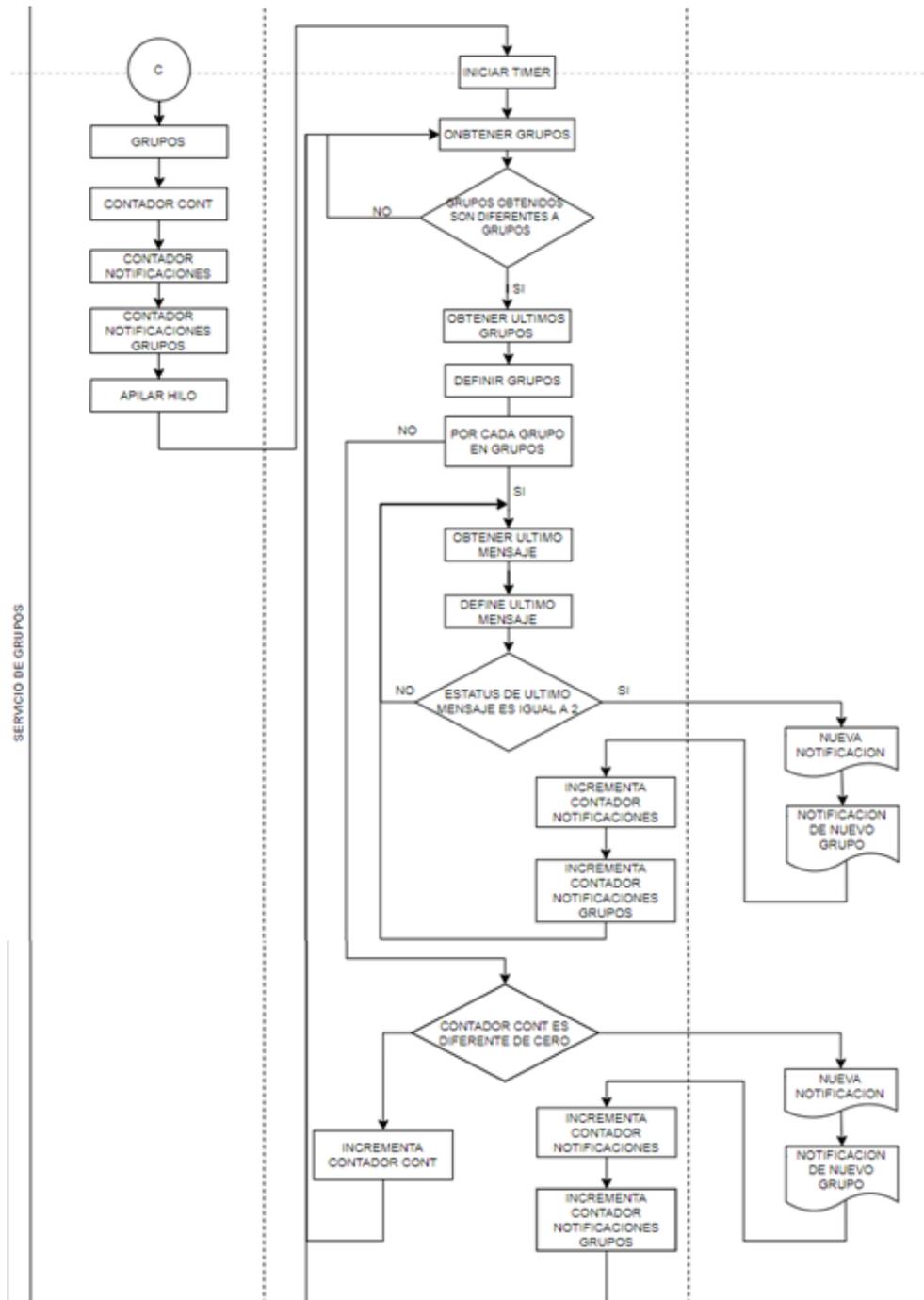


Fig. 13 Servicio de grupos

3.4.4 Servicio de mensaje individual nuevo

En el siguiente servicio se usa meramente para notificaciones de mensajes nuevos, así como para obtener el último mensaje de la conversación que tuvo un mensaje nuevo, véase Fig. 14.

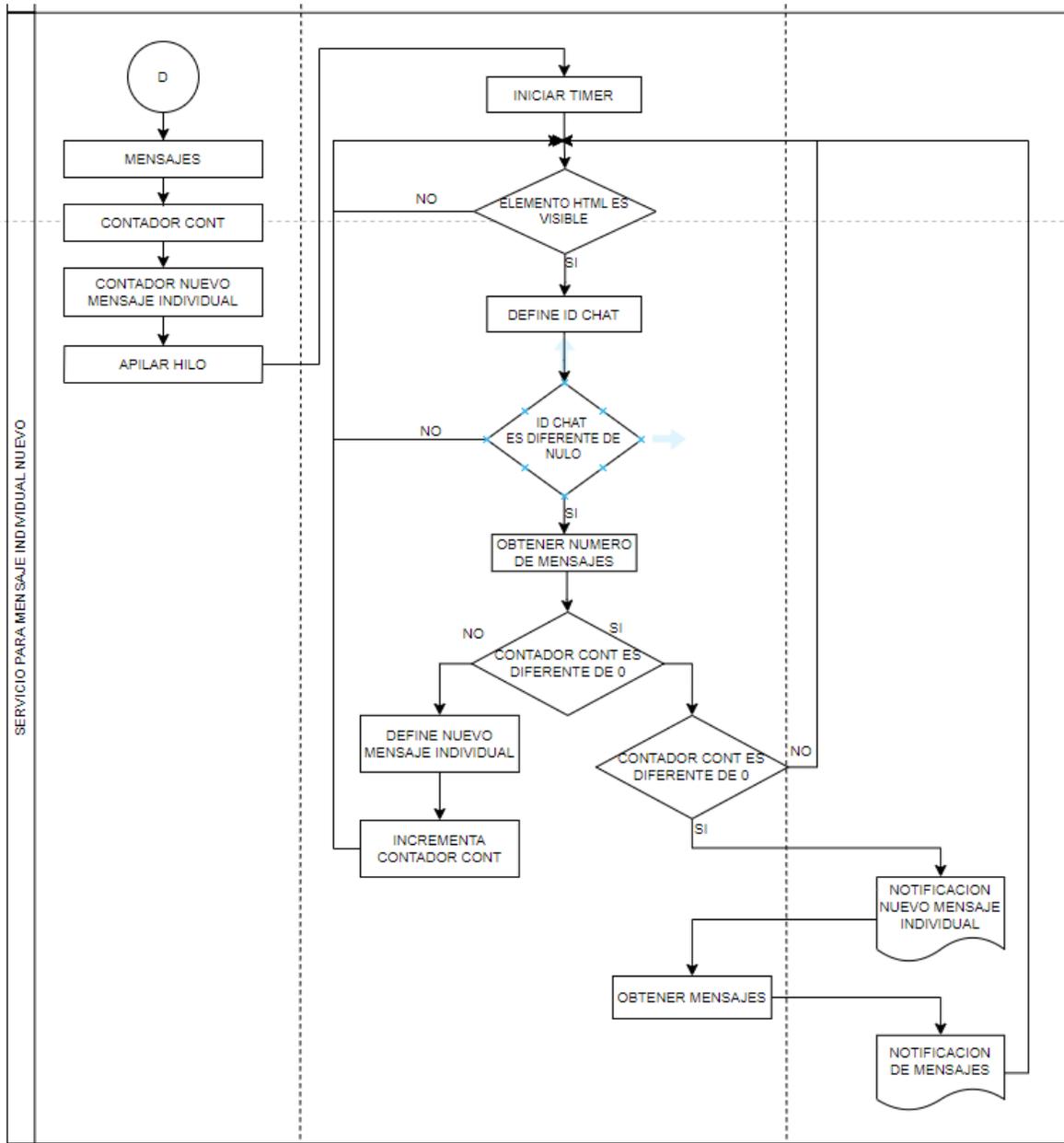


Fig. 14 Servicio de mensaje individual nuevo

3.4.5 Servicio de mensaje grupal nuevo

La Fig. 15 se usa para definir el proceso de un mensaje grupal nuevo, parecido al servicio anterior que obtiene el mensaje nuevo de la conversación grupal.

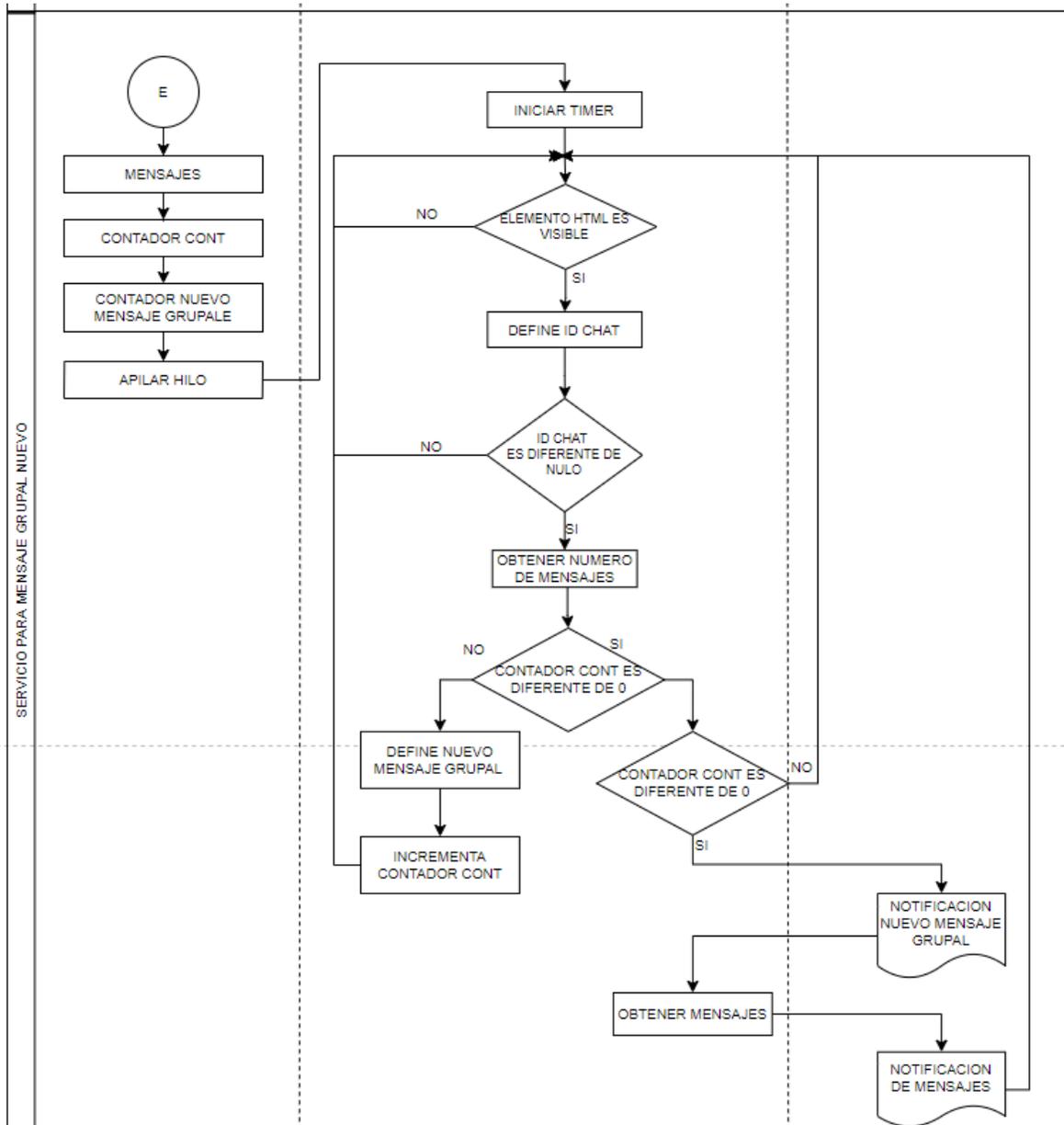


Fig. 15 Servicio de mensaje grupal nuevo

3.4.6 Servicio para mensajes individuales nuevos

Para este servicio se obtienen los mensajes nuevos de todo el usuario, este servicio solo es para temas de notificaciones y también actualiza la conversación con los mensajes nuevos, véase Fig. 16.

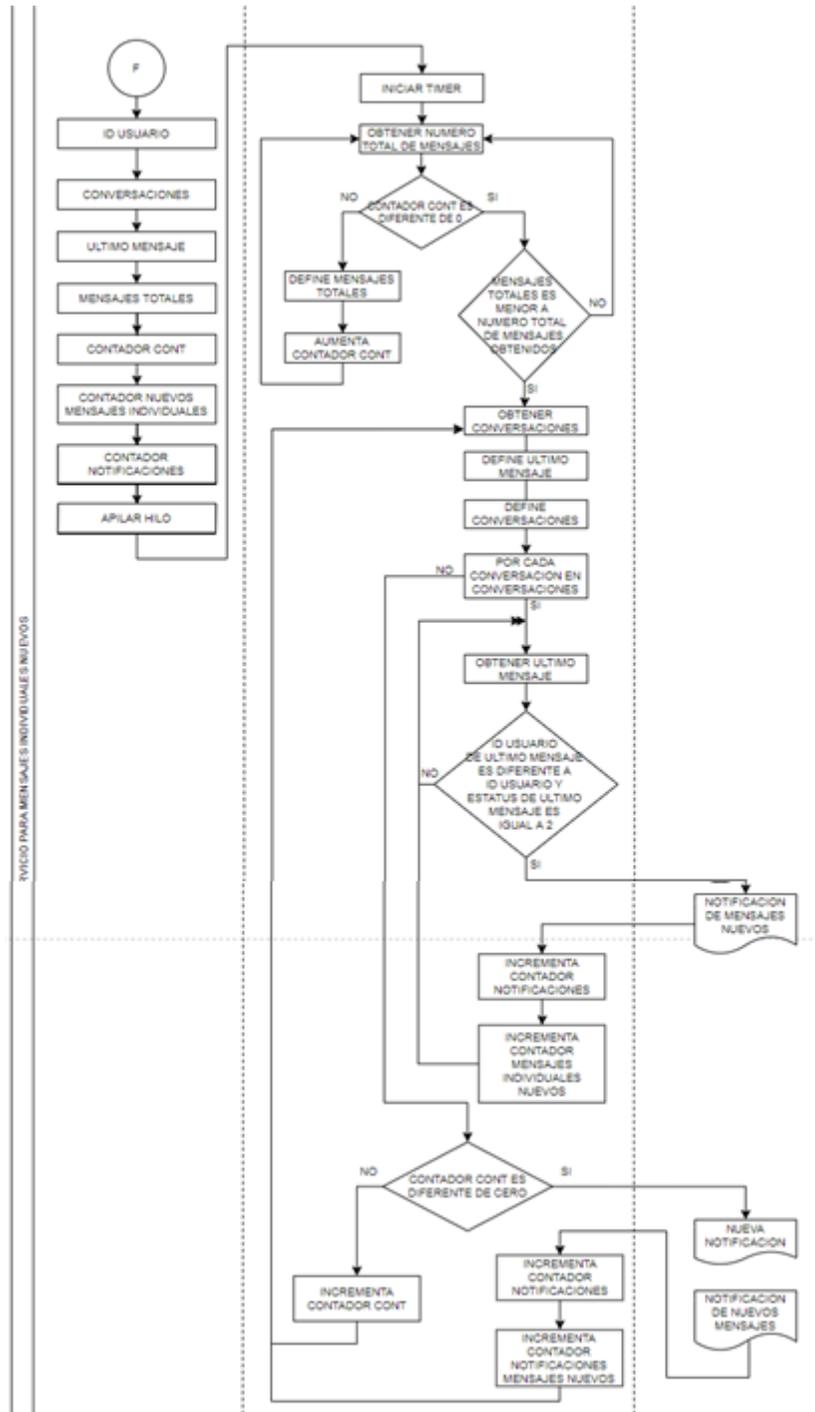


Fig. 16 Servicio mensajes individuales nuevos

3.4.7 Servicio para mensajes grupales nuevos

Al igual que el anterior servicio se ocupará para obtener los mensajes nuevos para los grupos, meramente para notificaciones nuevas para el usuario, véase Fig. 18.

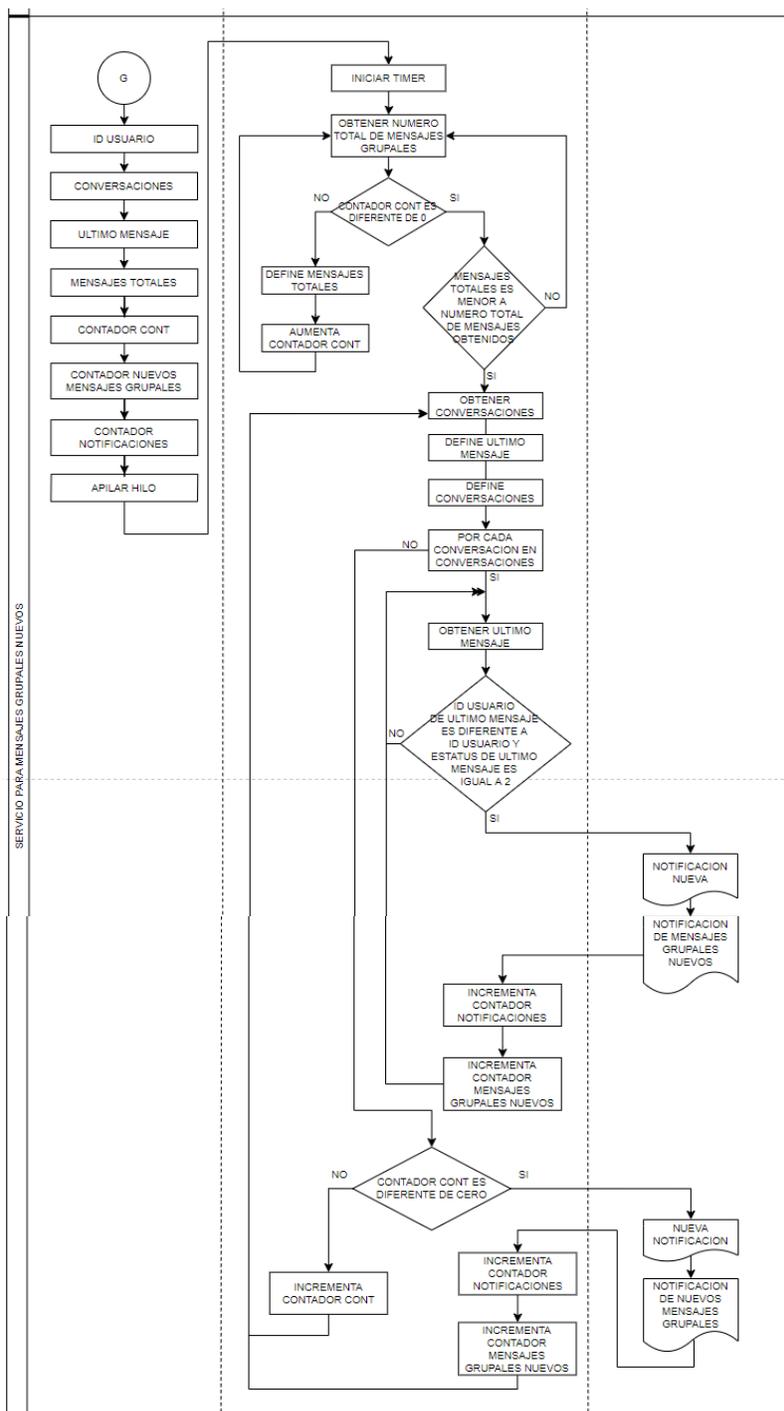


Fig. 18 Servicio para mensajes grupales nuevos

3.4.8 Método para enviar un mensaje

Como bien dice el nombre, se muestra el proceso para mandar un mensaje ya sea a una conversación nueva o a una conversación existente, este proceso actualiza también la lista de mensajes que se enviaron además de notificar al usuario que el mensaje ha sido enviado exitosamente, véase el proceso de la Fig. 19.

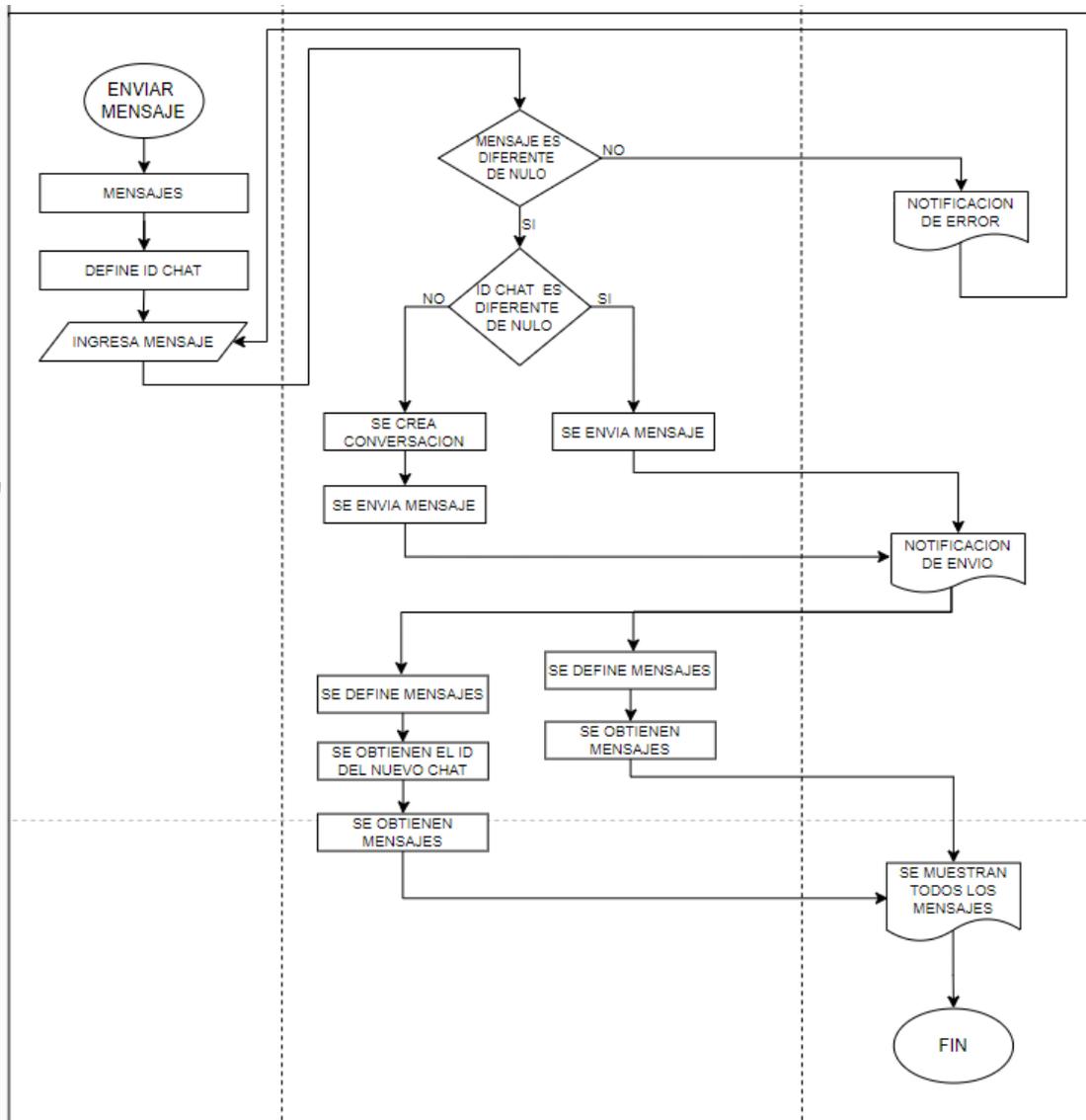


Fig. 19 Método para enviar un mensaje

3.4.9 Método para abrir un contacto

En este método (Fig. 20) se abrirá un contacto de la lista de contactos que se llenó en el previo servicio de contactos.

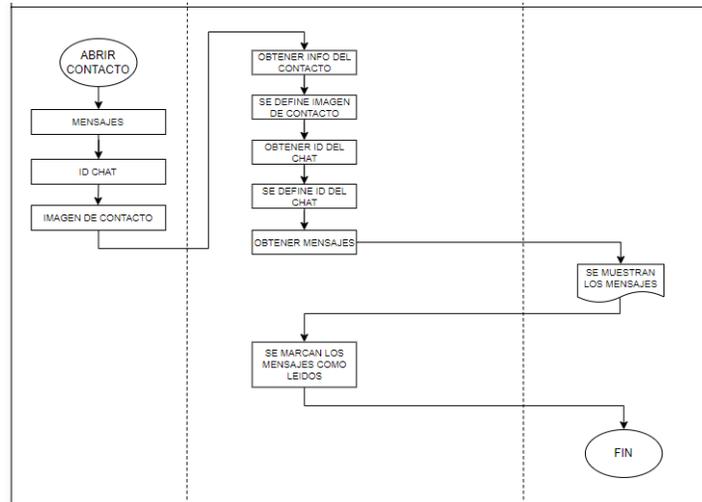


Fig. 20 Método para abrir un contacto

3.4.10 Método para abrir una conversación

Este método (Fig. 21) se usa cuando se abre una conversación que ya este creada, después de conseguir toda la información del usuario se marcaran como leídos los mensajes que aún no estén leídos.

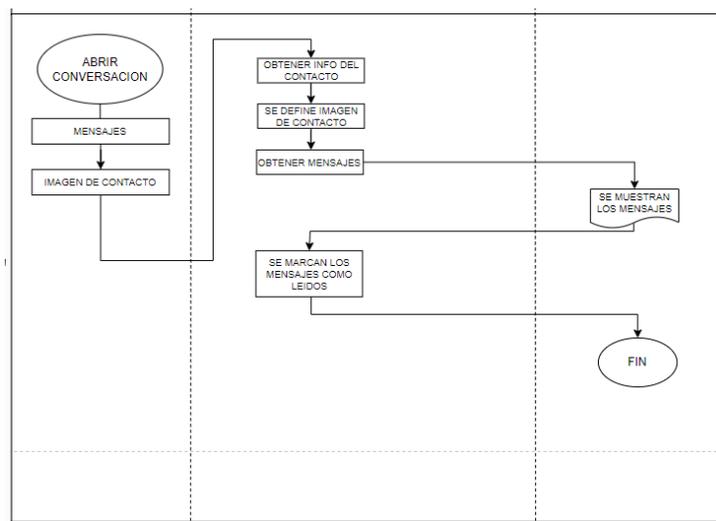


Fig. 21 Método para abrir una conversación

3.5 Modelo entidad – relación

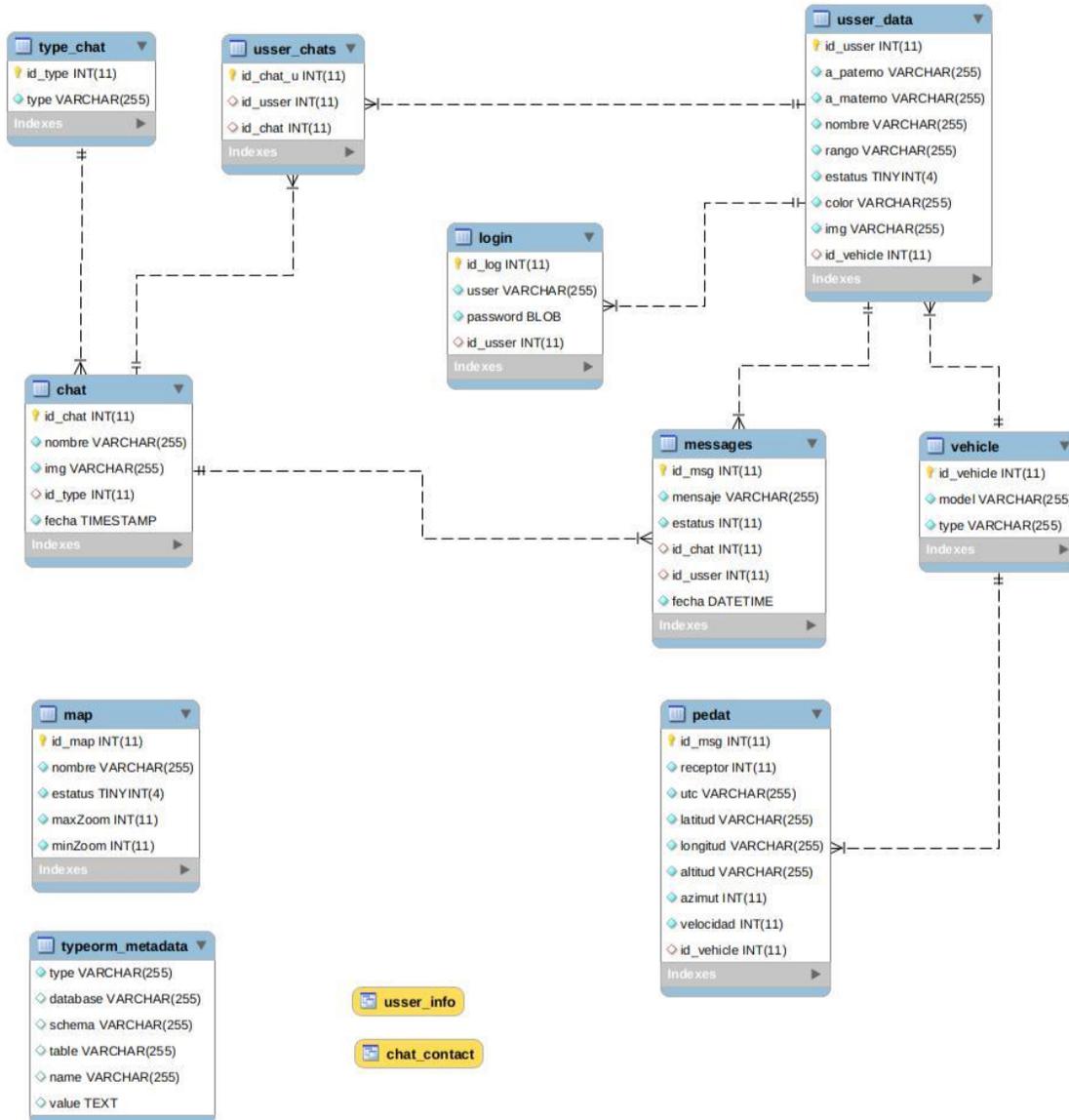


Fig. 22 Modelo E.R.

Como se puede observar en la Fig. 22 se muestra el modelo propuesto para la primera etapa del proyecto, cabe destacar que las entidades sin relación como es el caso de la tabla “map” sirve para control y manejo de los mapas que actualmente se encuentran en uso, así como la tabla “typeorm_metadata” nos proporciona un control de migraciones con ayuda del ORM usado en la app de NodeJs. Por consiguiente se implementó el uso de dos vistas llamadas “usser_info” y “chat_contact” para optimizar las consultas requeridas.

3.6 Conversión de mapas en formato ráster a teselas vectoriales.

Para realizar la conversión de un mapa en formato TIFF o ECW, se hizo uso del software de licencia de paga MAPTILER, a continuación, como se observa en la Fig. 23 se muestra de manera grafica el proceso de obtención de las teselas vectoriales las cuales posteriormente serán almacenadas en nuestro servidor.

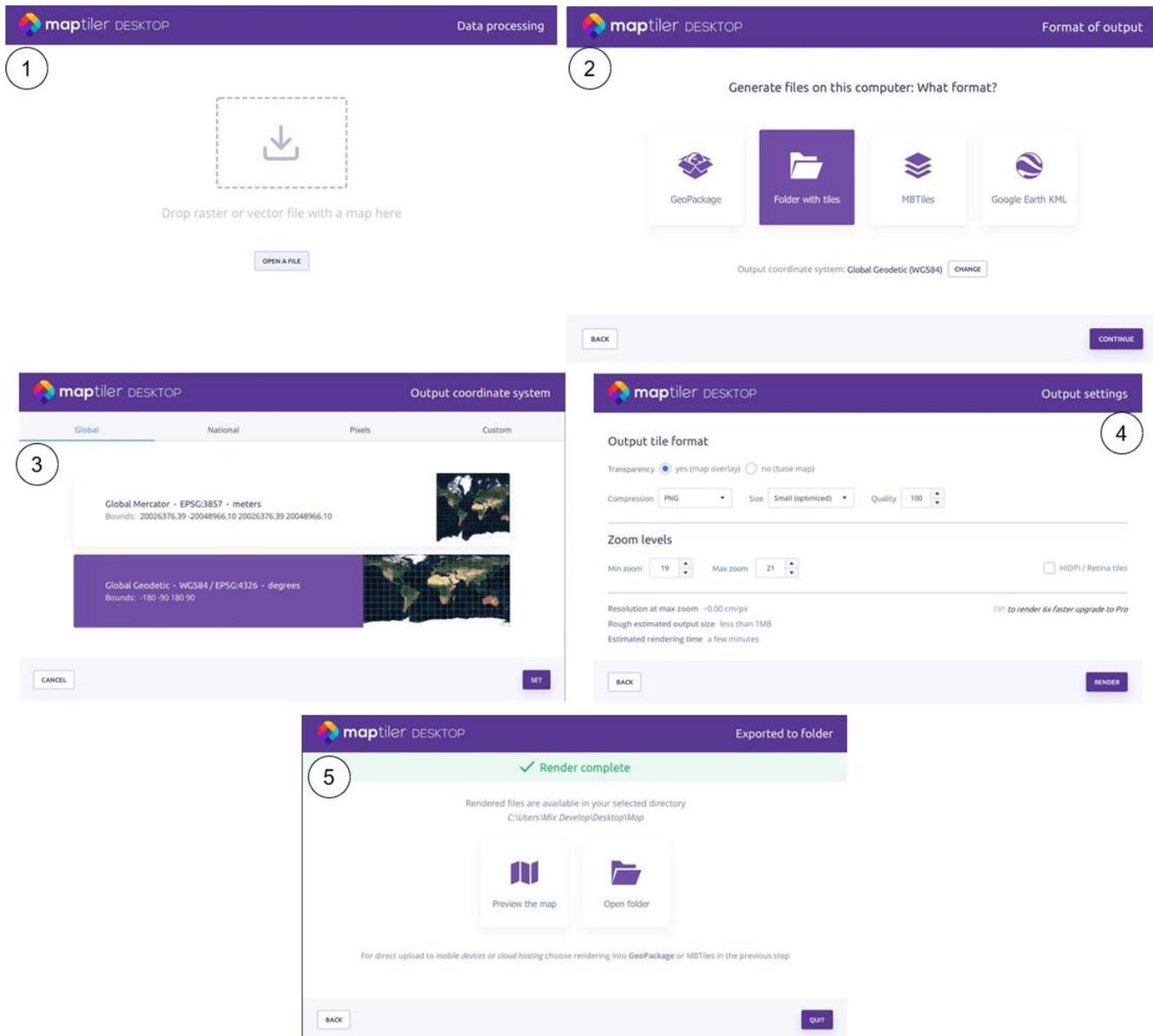


Fig. 23 Conversión de mapas con MapTiler

3.7 Servidor cartógrafo

En esta etapa se llevó a cabo el despliegue del servidor cartógrafo en plataformas Windows (Tabla 2) y Linux (Tabla 3) utilizando el software Geoserver y la librería GDAL para cargar mapas en formato ECW, específicamente la versión 2.2.1 y por último se Dockerizo la instalación para evitar volver a configurar y cargar mapas en un futuro.

Tabla 2 Actividades en entorno Windows

Actividad	Descripción
Instalación de JDK y JRE	Se instalaron las librerías necesarias para ejecutar código compilado en Java.
Instalación de Geoserver	Se instalo la versión 2.11.3 del servidor cartógrafo GeoServer.
Instalación de librerías GDAL	Se instalo la versión 2.2.1 de las librerías GDAL, las cuales se utilizaron para tener soporte de mapas en formato ECW.
Cargar mapas en formato ECW	Se cargaron los mapas en el servido cartógrafo.

Tabla 3 Actividades en entorno Linux

Actividad	Descripción
Instalación de Docker	Se instalo la versión 20.10.8 sistema operativo para contenedores (Docker), el cual se utilizó para virtualizar
Instalación de GeoServer	Se instalo una imagen Docker de GeoServer versión 2.11.3
Instalación de librerías GDAL	Se instalo en la imagen Docker la versión 2.2.1 de las librerías GDAL, las cuales se utilizaron para tener soporte de mapas en formato ECW.
Cargar mapas en formato ECW	Se copiaron los mapas en el docker y se cargaron en el servidor cartógrafo.
Dockerizar instalación	Se configuro y respaldo la imagen Docker de Geoserver para posteriormente subirla a Docker Hub.

3.7.1 Carga de mapas en formato ECW

Una vez instalado el docker de GeoServer 2.11. Con el siguiente comando “sudo docker images” listamos las imágenes que hemos descargado.

```
Terminal - geoserver@debian: ~
Archivo Editar Ver Terminal Pestañas Ayuda
geoserver@debian:~$ sudo docker images
[sudo] password for geoserver:
REPOSITORY          TAG             IMAGE ID        CREATED         SIZE
winsent/geoserver   2.11-ecw       2c890c470e57   3 years ago    1.82GB
geoserver@debian:~$
```

Fig. 24 Id Docker

Con el siguiente comando “sudo Docker run -d [ID]” reemplazando ID por el IMAGE ID que nos ha proporcionado el comando anterior como se muestra en la Fig. 24 corremos la imagen de Geoserver el resultado se observa en la Fig. 25.

```
geoserver@debian:~$ sudo docker run -d 2c890c470e57
83d8a5ee3293b034c06b100ff7db79fdfa0808da09e8b982f8665e522c22373e
geoserver@debian:~$ sudo docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS
PORTS         NAMES
83d8a5ee3293   2c890c470e57  "/opt/geoserver/bin/..." 32 seconds ago Up 28
seconds      8080/tcp     eager_wilson
geoserver@debian:~$
```

Fig. 25 Estatus del Docker

Con el comando “sudo Docker exec -i -t [ID] sh” reemplazando ID por el IMAGE ID que nos ha proporcionado la Fig. 24 accedemos al contenedor como se muestra en la Fig. 26:

```
geoserver@debian:~/Descargas$ sudo docker exec -i -t ed4c0ee2cc14 sh
# ks
sh: 1: ks: not found
# ls
bin dev home lib64 mnt proc run srv tmp var
boot etc lib media opt root sbin sys usr
#
```

Fig. 26 Directorios del Docker

Dirigirse al directorio data_dir como se muestra en la Fig. 27, dentro de la carpeta de geoserver y creamos una carpeta llamada PINA_MAPS.

```
# cd geoserver
# ls
GPL.txt      RELEASE_NOTES.txt  bin      lib      resources  webapps
LICENSE.txt  RUNNING.txt        data_dir logs     start.ini
README.txt   VERSION.txt        etc      modules  start.jar
# cd data_dir
# ls
README.rst  demo      gwc-layers  palettes  styles      wfs.xml
coverages   global.xml layergroups  plugIns   user_projections  wms.xml
csw.xml     gwc       logging.xml printing  validation      workspaces
data        gwc-gs.xml logs       security  wcs.xml         www
# pwd
/opt/geoserver/data_dir
#
```

Fig. 27 Directorio data_dir

Con el comando “sudo docker cp [nombre del archivo][Id]:[Directorio de destino]” copiaremos los mapas en formato ECW en la carpeta anteriormente creada como se visualiza en la Fig. 28, confirmar mediante el comando “ls” que las carpetas se hayan copiado correctamente como se observa en la Fig. 29.

```
geoserver@debian:/media/geoserver/ADATA HD720/INSTALACION_PINA/software_para_p
ina/PINA_MAPS$ sudo docker cp 100KSAT.ecw ed4c0ee2cc14:/opt/geoserver/data_dir
/PINA_MAPS
geoserver@debian:/media/geoserver/ADATA HD720/INSTALACION_PINA/software_para_p
ina/PINA_MAPS$
geoserver@debian:/media/geoserver/ADATA HD720/INSTALACION_PINA/software_para_p
ina/PINA_MAPS$
```

Fig. 28 Copiar mapas al docker

```
# ls
PINA_MAPS  demo      layergroups  printing      wcs.xml
README.rst global.xml logging.xml  security      wfs.xml
coverages  gwc       logs         styles        wms.xml
csw.xml    gwc-gs.xml palettes     user_projections  workspaces
data       gwc-layers plugIns      validation     www
# cd PINA_MAPS
# ls
# pwd
/opt/geoserver/data_dir/PINA_MAPS
# ls
100KSAT.ecw
#
```

Fig. 29 Listar archivos

Abrir GeoServer entrando al navegador y colocando la dirección IP que se muestra en la Fig. 30

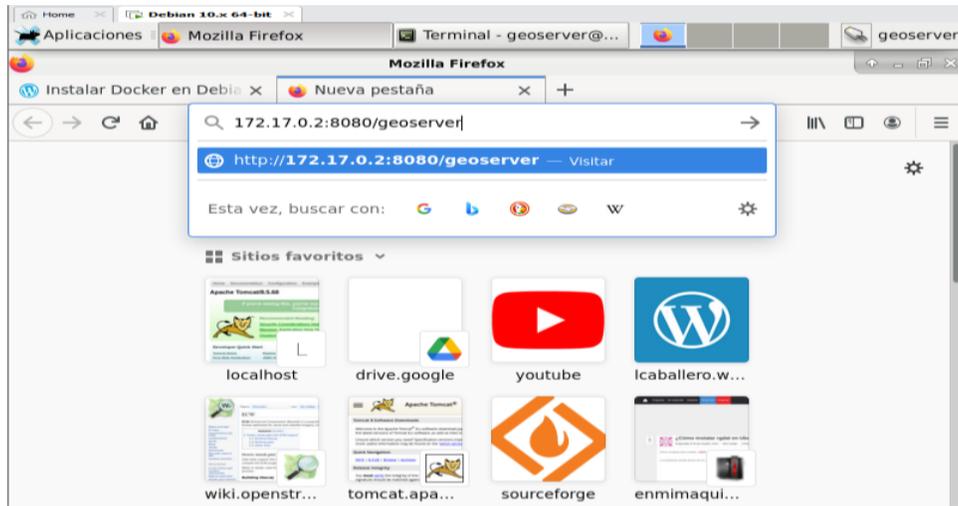


Fig. 30 GeoServer

Ingresar con las siguientes credenciales:

- User: admin
- Password: geoserver

Se visualizará la pantalla de bienvenida que se muestra en la Fig. 31

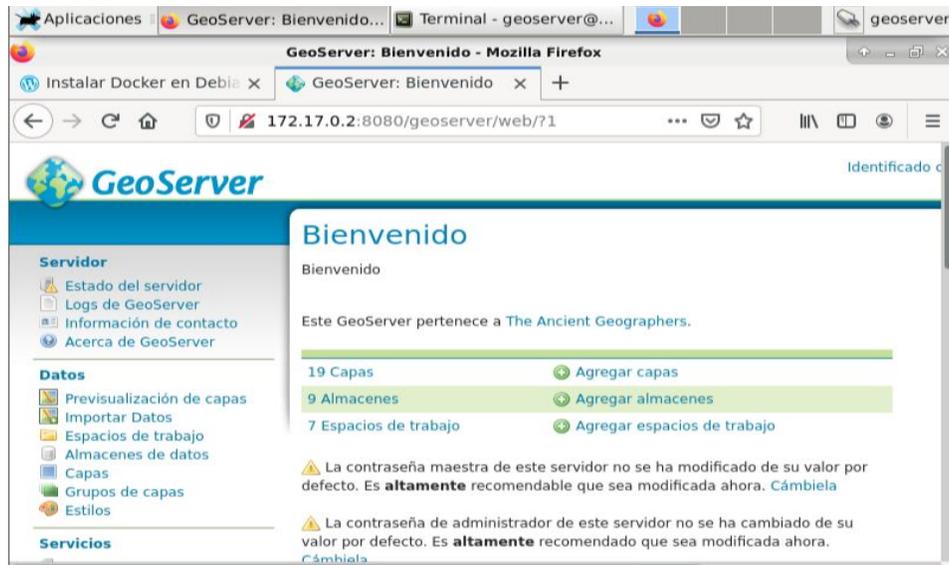


Fig. 31 Pantalla de bienvenida GeoServer

3.7.2 Espacio de Trabajo

Dirigirse al espacio de trabajo y crear un espacio de trabajo nuevo, como se muestra en la Fig. 32 posteriormente colocar un nombre como indica la Fig. 33



Fig. 32 Nuevo espacio de trabajo

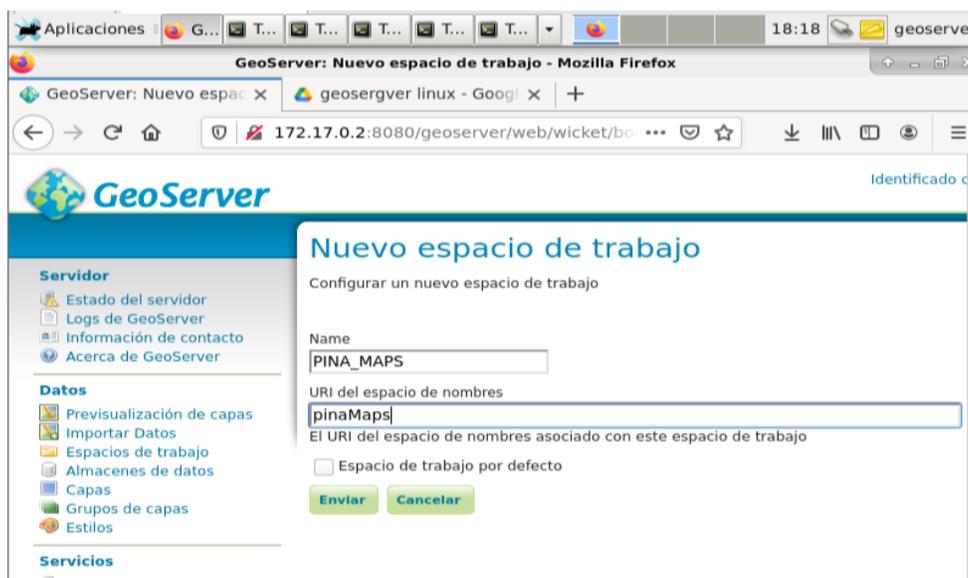


Fig. 33 Asignar nombre

Dar clic en el botón enviar y después dirigirse al apartado de almacén de datos para posteriormente agregar uno nuevo como se observa en la Fig. 34, seleccionar formato ECW como indica la Fig. 35

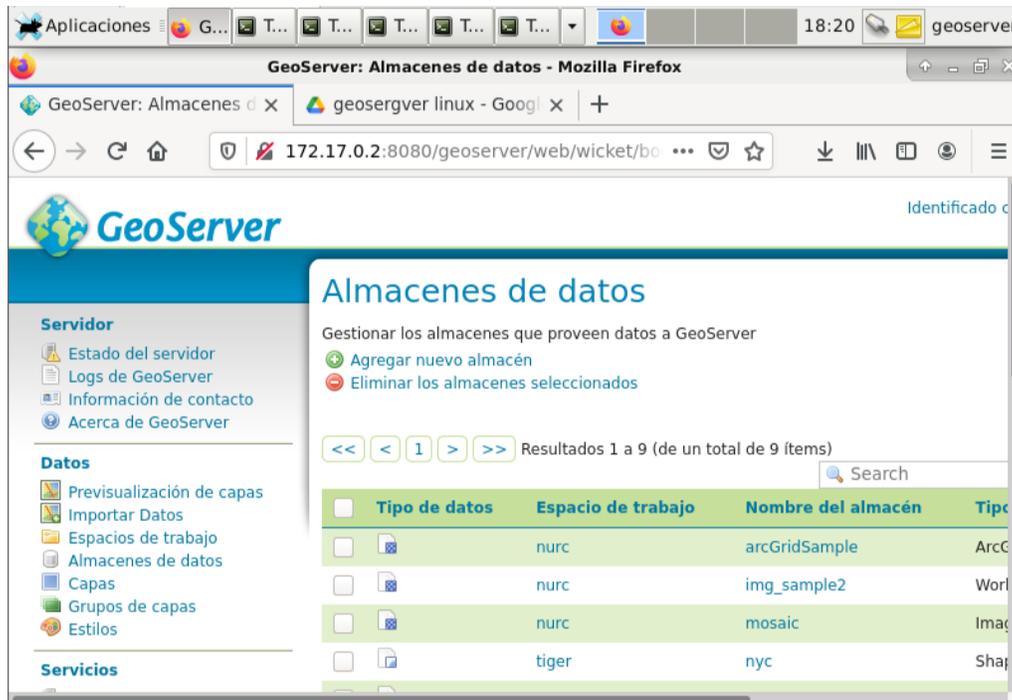


Fig. 34 Nuevo almacén de datos

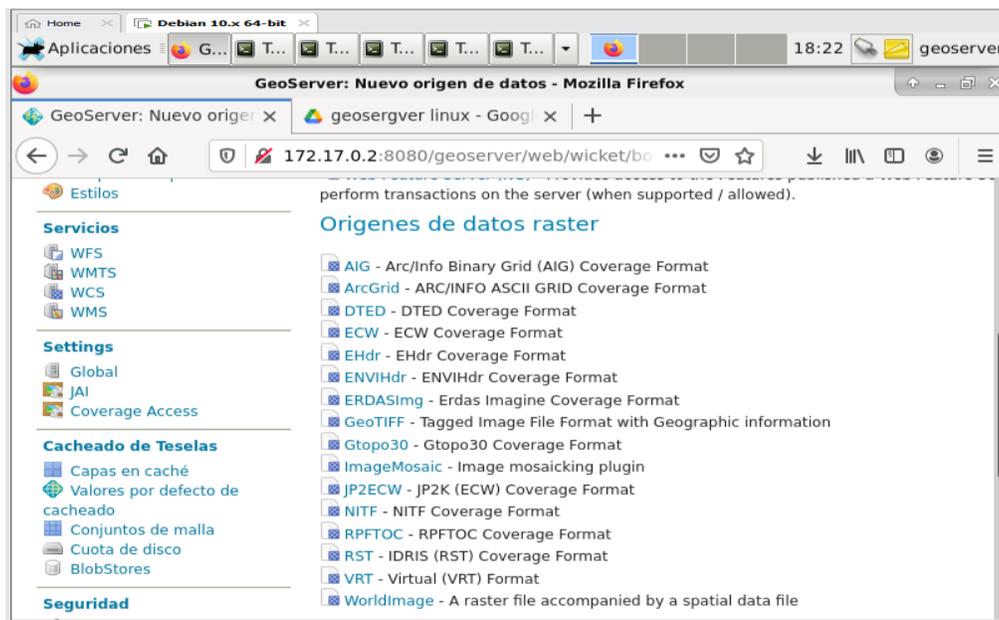


Fig. 35 Formato ECW

Seleccionar la carpeta PINA_MAPS como se observa en la Fig. 36, elegir el mapa a cargar como indica en la Fig. 37 por ultimo dar clic en guardar como se visualiza en la Fig. 38

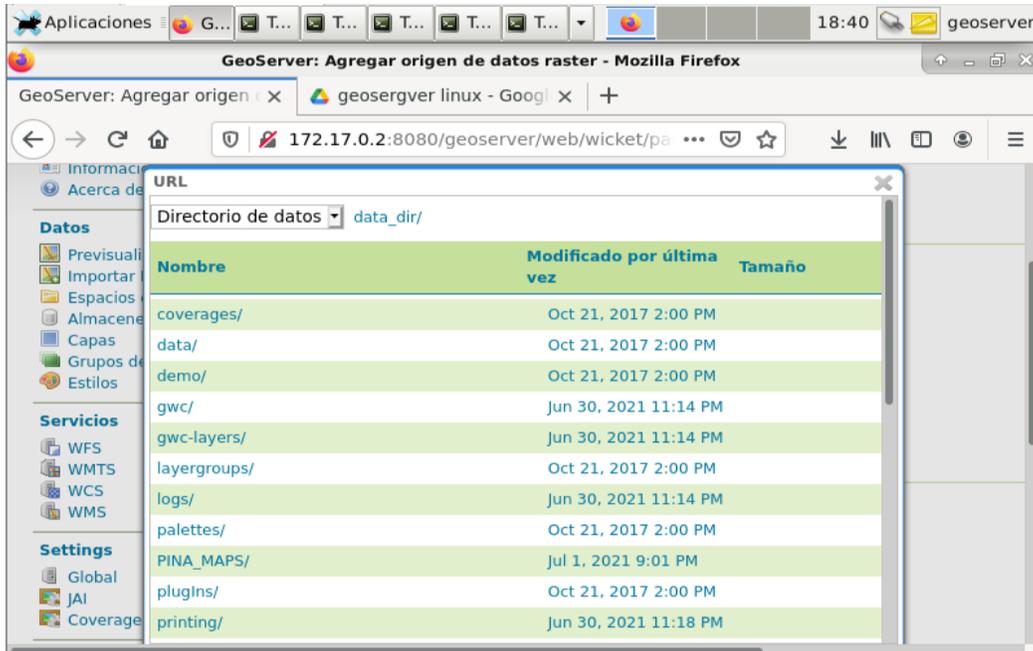


Fig. 36 Seleccionar directorio

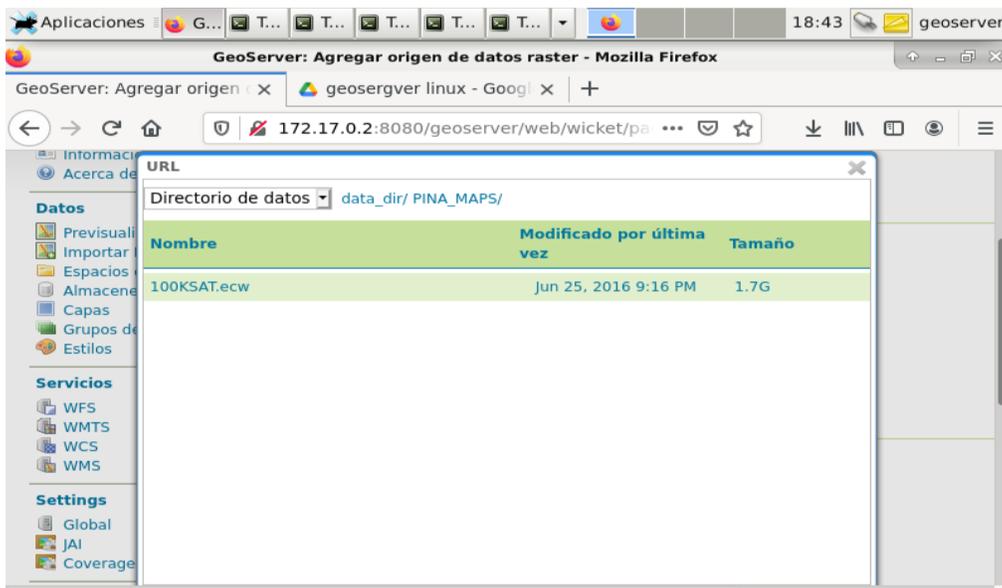


Fig. 37 Mapas disponibles

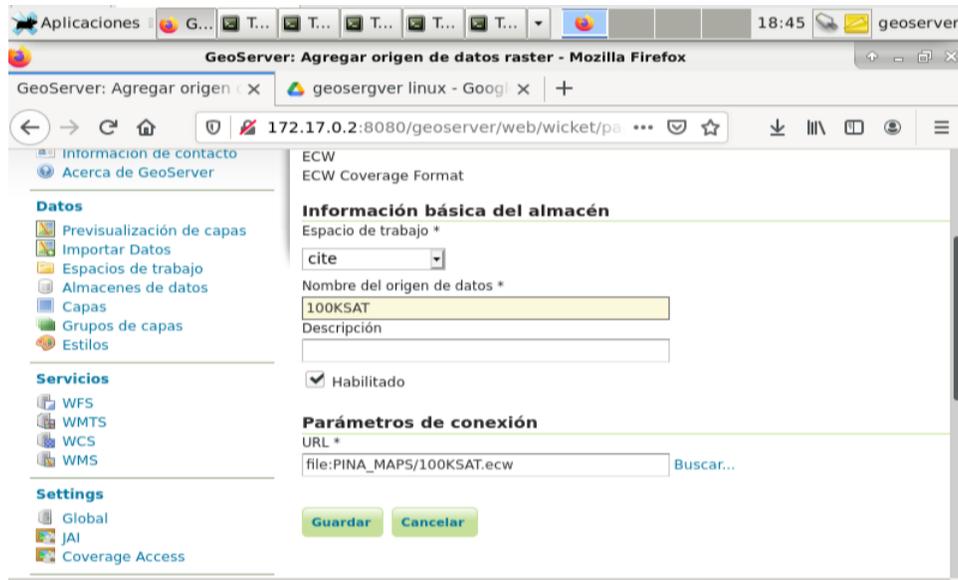


Fig. 38 Guardar mapa

3.7.2 Previsualización de Mapas

Dar clic en previsualización de capas, posteriormente dar clic en OpenLayers KML y visualizar que el mapa se muestre correctamente como se observa en la Fig. 39

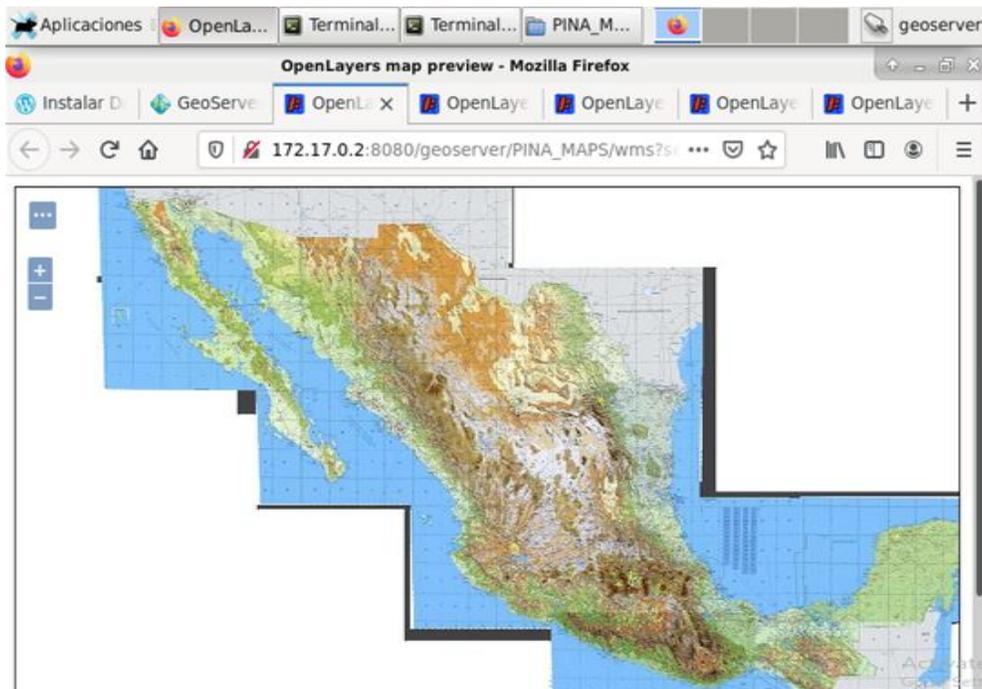


Fig. 39 Previsualización de mapa

Capítulo IV Análisis e interpretación de resultados

En este capítulo se aborda la evaluación y el funcionamiento de la versión final de la aplicación web en conjunto con una aeronave de la FAM para proporcionar datos en tiempo real, detallando cada una de las pruebas con los resultados obtenidos del transceptor dentro del vehículo, visualizando en tiempo real las coordenadas y mensajes transmitidos por el dispositivo.

4.1 Instalación y Configuración del Entorno de Desarrollo

En esta primera etapa se llevó a cabo la instalación de software que se requiere para ejecutar correctamente el sistema, así como el desarrollo de los manuales de configuración de este para realizar la prueba de instalación, esta consistió en evaluar el comportamiento del software en su entorno de pruebas Linux usando Ubuntu 20.04 LTS como se puede observar en la Tabla 4.

Tabla 4 Configurar servidor de pruebas Ubuntu 20.04 LTS

Actividad	Descripción
Instalar Ubuntu 20.04 LTS	Se realizó la instalación de Ubuntu en una computadora de escritorio con 16 GB de RAM DDR4, grafica GTX 1650 y 2TB de disco duro
Instalar Git	Se realizó la instalación de Git 2.31
Instalar NodeJs	Se realizó la instalación de NodeJs 14.17.4
Instalar Angular	Se realizó la instalación de Angular 11.0
Instalar Servidor De Base De Datos	Se realizó la instalación de MariaDB 10.1.48
Instalar Servidor Web	Se realizó la instalación de Apache Server

4.1.1 Configuración del Servidor cartográfico con Tiles

Se editó la dirección predeterminada por Apache Server asignándola a una partición con 300 giga bytes (gb). Dentro del directorio "/www/html" se creó una carpeta con el nombre Maps en la cual se colocaron todas las carpetas generadas al usar MapTiler como se puede observar en la Fig. 40.

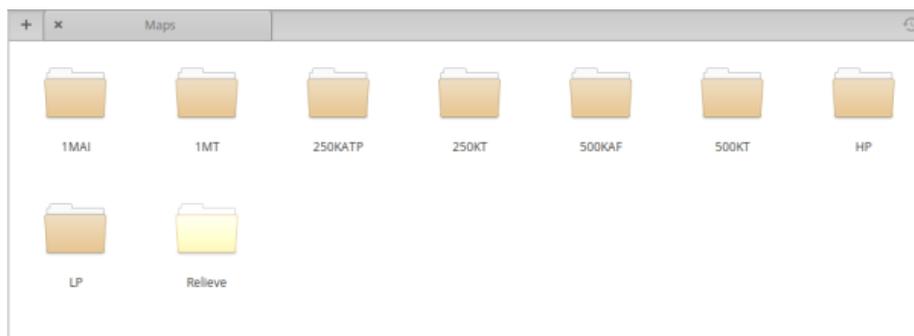


Fig. 40 Mapas alojados en el servidor

4.1.2 Configuración del Servidor cartográfico con ECW

El servidor cartógrafo se encuentra configurado para almacenar imágenes diseñadas para su uso con datos geospaciales en el formato Enhanced Compression Wavelet (ECW) utilizando una biblioteca traductora para formatos de datos geospaciales vectoriales y ráster (GDAL).

Posteriormente se utilizó una plataforma de virtualización a nivel de sistema operativo (DOCKER), de esta manera, dicha aplicación puede ejecutarse sin problema alguno en cualquier entorno con un despliegue rápido.

En la Fig. 41 se muestra, que se encuentran instaladas las librerías GDAL para soporte de mapas en formato ECW.

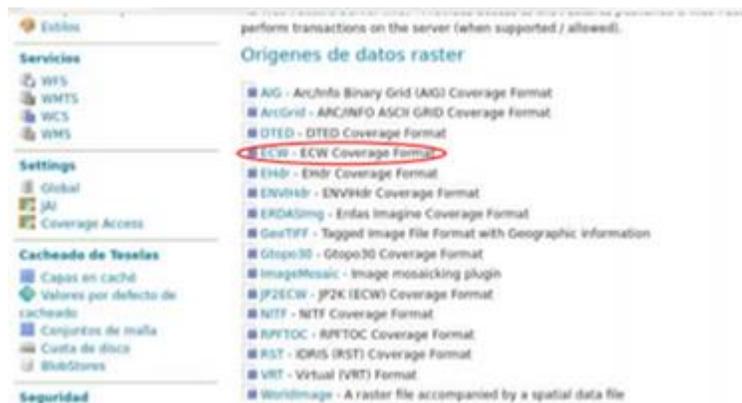


Fig. 41 Origen de datos formato ECW

Como alternativa En la Fig. 42 se puede observar el Servidor Geografico implementado en la aplicación.

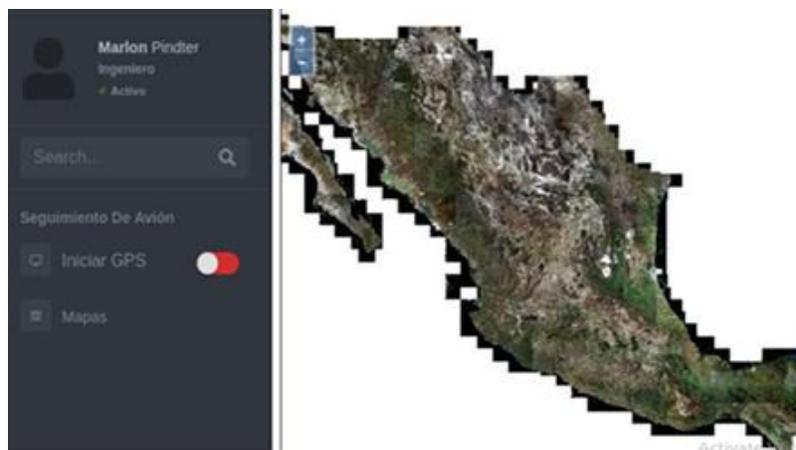


Fig. 42 Implementación de GeoServer

Se realizó una comparación de rendimiento entre los dos métodos para visualizar los mapas en formato ECW.

Como se muestra en la Fig. 43 (método 1) y Fig. 44 (método 2), se realiza la comparativa en cuanto a la cantidad de recursos que se utilizan durante la carga de la aplicación, donde el método 1 consume menos recursos que el método 2, el cual consume 10.98 Mb más.

Bytes	Contar	Total de bytes	Total de la cuenta	Nombre
3 863 096 19%	9 569 5%	2 788 048 13%	7 613 4%	js:BaseScript
3 593 408 18%	1 0%	3 593 408 18%	1 0%	Idocument
2 229 536 11%	31 073 17%	2 229 536 11%	31 073 17%	Function
2 194 928 11%	38 404 21%	2 194 928 11%	38 404 21%	js:Shape
2 180 912 11%	17 222 9%	2 180 912 11%	17 222 9%	Object
1 998 176 10%	43 475 24%	1 998 176 10%	43 475 24%	strings
994 856 5%	11 059 6%	994 856 5%	11 059 6%	Array
843 992 4%	784 0%	843 992 4%	784 0%	js:it:ItCode
554 256 3%	10 639 6%	554 256 3%	10 639 6%	js:ObjectGr...
425 184 2%	5 185 3%	425 184 2%	5 185 3%	Call
422 096 2%	5 293 3%	422 096 2%	5 293 3%	js:Scope
159 808 1%	908 0%	159 808 1%	908 0%	XMLHttpReq...
151 872 1%	2 373 1%	151 872 1%	2 373 1%	Map
0 0%	0 0%	6 439 304 32%	71 064 39%	objects
106 312 1%	196 0%	106 312 1%	196 0%	js:RegExpSh...
94 688 0%	807 0%	94 688 0%	807 0%	LexicalEnvi...
89 760 0%	935 1%	89 760 0%	935 1%	Arguments
67 648 0%	2 114 1%	67 648 0%	2 114 1%	js:BaseShape
62 976 0%	74 0%	62 976 0%	74 0%	text

Fig. 43 Consumo de memoria (método 1)

Bytes	Contar	Total de bytes	Total de la cuenta	Nombre
6 799 512 22%	55 283 18%	6 799 512 22%	55 283 18%	Object
4 425 544 14%	68 481 23%	4 425 544 14%	68 481 23%	Function
3 886 344 12%	3 0%	3 886 344 12%	3 0%	Idocument
3 270 512 10%	12 087 4%	2 097 168 7%	7 570 3%	js:BaseScript
3 125 784 10%	40 449 14%	3 125 784 10%	40 449 14%	Array
2 604 392 8%	9 281 3%	2 604 392 8%	9 281 3%	Map
2 037 968 7%	61 150 20%	2 037 968 7%	61 150 20%	strings
1 204 624 4%	17 714 6%	1 204 624 4%	17 714 6%	Call
713 448 2%	4 895 2%	713 448 2%	4 895 2%	js:PropMap
687 288 2%	4 091 1%	687 288 2%	4 091 1%	XMLHttpReq...
519 296 2%	629 0%	519 296 2%	629 0%	js:it:ItCode
404 192 1%	4 814 2%	404 192 1%	4 814 2%	js:Scope
361 152 1%	4 104 1%	361 152 1%	4 104 1%	Arguments
308 736 1%	2 642 1%	308 736 1%	2 642 1%	LexicalEnvi...
102 944 0%	3 212 1%	102 944 0%	3 212 1%	js:Shape
91 112 0%	215 0%	91 112 0%	215 0%	js:RegExpSh...
88 560 0%	738 0%	88 560 0%	738 0%	SavedFrame

Fig. 44 Consumo de memoria (método 2)

En la Fig. 45 (método 1) y Fig. 46 (método 2), podemos observar de una manera grafica el total de memoria utilizada para cada función, la función que nos interesa es la de “map”, la cual se puede notar claramente que el consumo de memoria aumenta con el método 2.

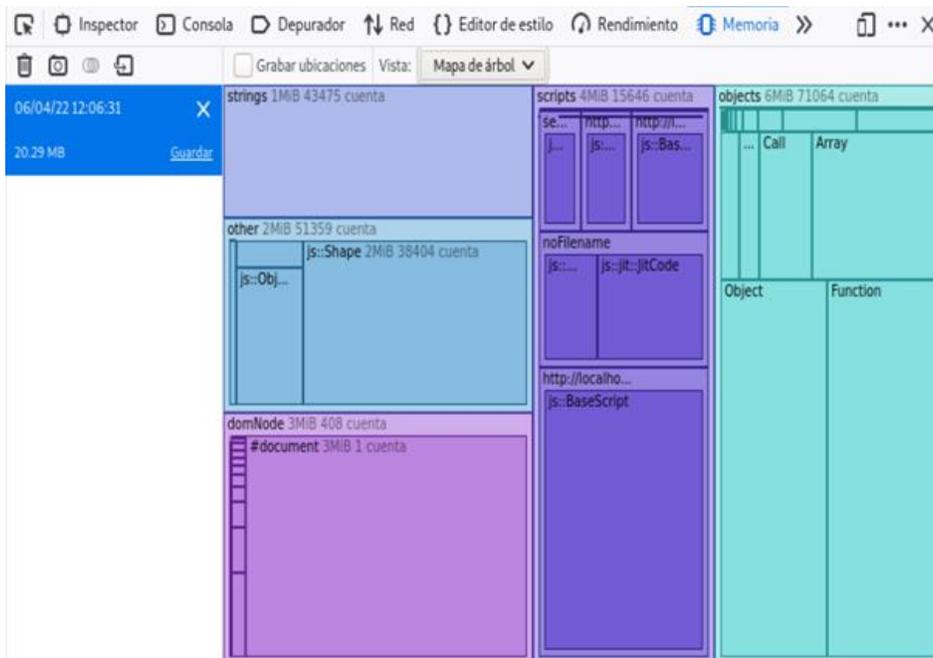


Fig. 45 Representación gráfica del consumo de memoria (método 1).

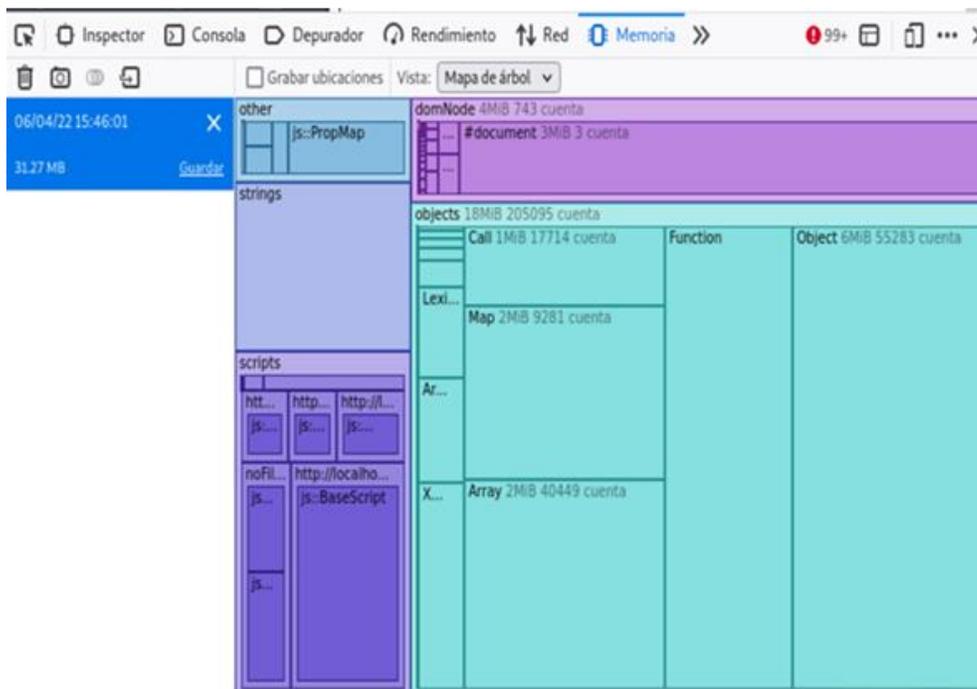


Fig. 46 Representación gráfica del consumo de memoria (método 2)

Por último, en la Fig. 47 (método 1) y Fig. 48 (método 2), se puede ver el tiempo que demora en cargar cada método, donde el método 1 es más rápido en comparación al método 2 por una diferencia de 6,280 ms

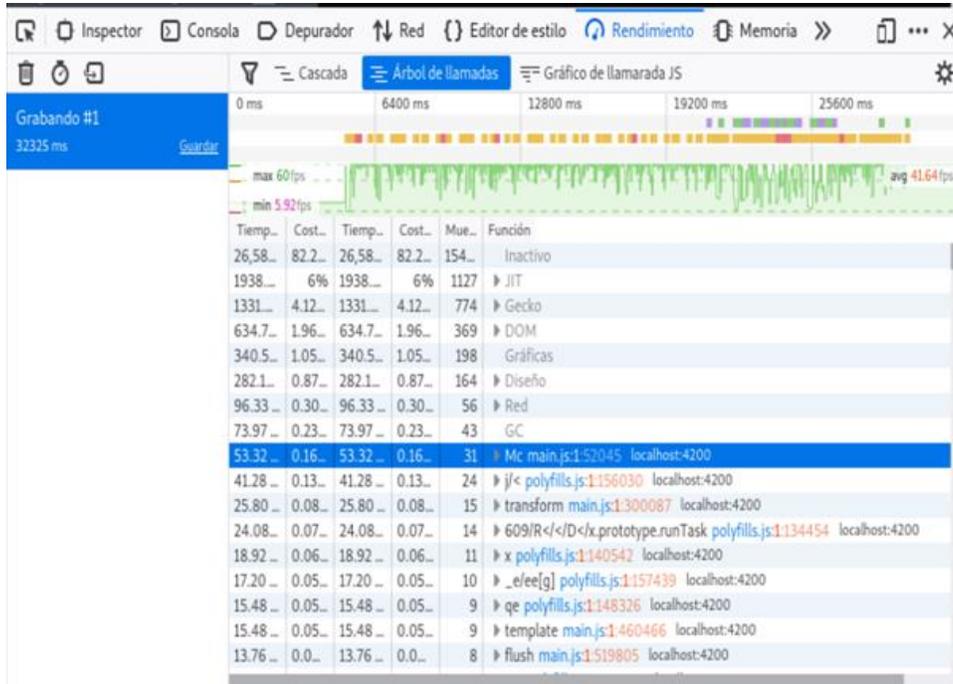


Fig. 47 Rendimiento (método 1)

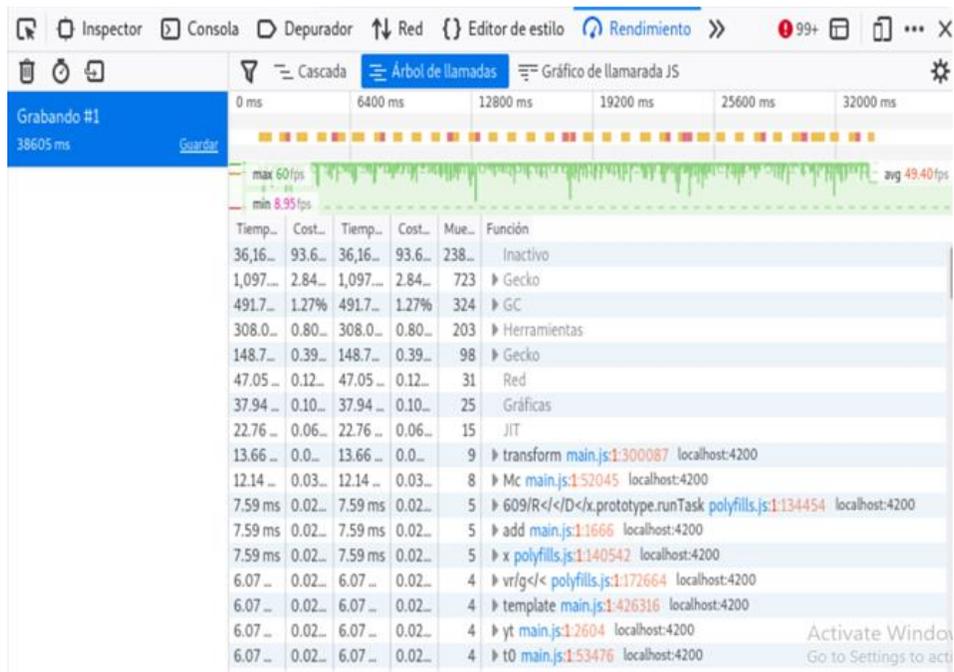


Fig. 48 Rendimiento (método 2)

4.2 Implementación y pruebas TRL5 del Visualizador Geográfico y Chat Táctico

Se realizaron pruebas con una aeronave la cual mediante el uso de un tranceptor , envía la información en formato PEDAT para posteriormente ser recuperada por los servicios de NodeJs usando el servidor físico en la estación terrena que cuenta con el sistema operativo con Ubuntu 20.04 LTS, actividad realizada en las oficinas del CIDTFA, donde se muestran los geo datos encriptados y recuperados del tranceptor en formato del PEDAT, mismos que son enviados al visualizador geográfico y al chat táctico con su respectivo mensaje como se observa en la Fig. 49.

```
Mensaje recibido de Tranceptor 001 - 13/07/2021 06:10:51 p. m. - (SMS)
AT+CMGD=1
{"Pedat_etsa":"001002210713181044+1946160-0990406022360000349"}

Mensaje recibido de Tranceptor 001 - 13/07/2021 06:10:57 p. m. - (SMS)
AT+CMGD=1
{"Pedat_etsa":"001002210713181053+1946150-0990406022900000349"}

Mensaje recibido de Tranceptor 001 - 13/07/2021 06:11:07 p. m. - (SMS)
AT+CMGD=1
{"Pedat_etsa":"001002210713181100+1946150-0990406022900000349"}

Mensaje recibido de Tranceptor 001 - 13/07/2021 06:11:08 p. m. - (SMS)
AT+CMGD=1
{"Pedat_etsa":"001002210713181105+1946150-0990406022900000349"}

Mensaje recibido de Tranceptor 001 - 13/07/2021 06:11:13 p. m. - (SMS)
AT+CMGD=1
{"Pedat_etsa":"001002210713181110+1946150-0990406022900000349"}

Mensaje recibido de Tranceptor 001 - 13/07/2021 06:11:19 p. m. - (SMS)
AT+CMGD=1
{"Pedat_etsa":"001002210713181116+1946130-0990405024690000349"}

Mensaje recibido de Tranceptor 001 - 13/07/2021 06:11:24 p. m. - (SMS)
AT+CMGD=1
{"Pedat_etsa":"001002210713181121+1946140-0990405024690000349"}
```

Fig. 49 Registros de datos recibidos

Iniciando la aplicación de angular, mediante peticiones a la API externa se obtienen todos los mapas disponibles como se observa en la Fig. 50 estos son desplegados en forma de lista, al dar clic en alguna opción se podrá visualizar el mapa, de igual manera activando el servicio de GPS iniciará la visualización en tiempo real de las aeronaves.

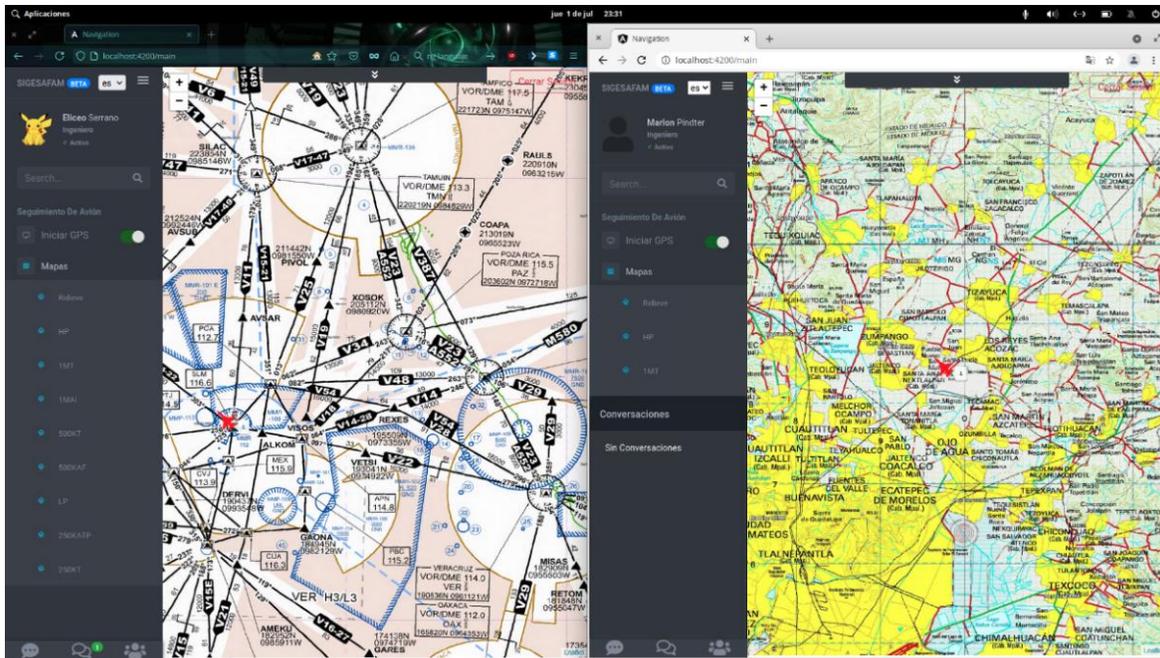


Fig. 50 Mapa con marcador visible

La Fig. 51 visualiza geo mapas vectoriales (50KT, 100KT, 250KT, 500KT, 1MKT, etc.) con datos geodésicos fuera de línea, una vez activado el servicio de GPS, se podrá observar la ruta seguida por la aeronave, así como su ubicación actual y una etiqueta con el id.

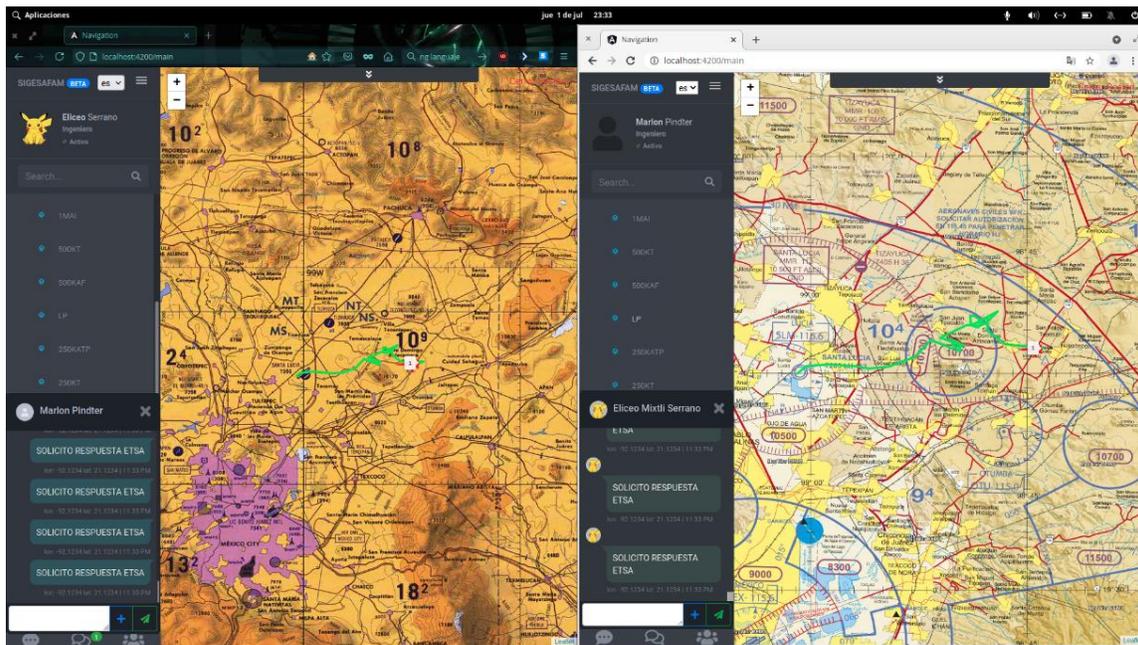


Fig. 51 Rastro de las aeronaves.

La Fig. 52 presenta los resultados obtenidos de las pruebas realizadas con el sistema de comunicación (chat táctico) ya que permite enviar mensajes predefinidos o personalizados cumpliendo el formato PEDAT.

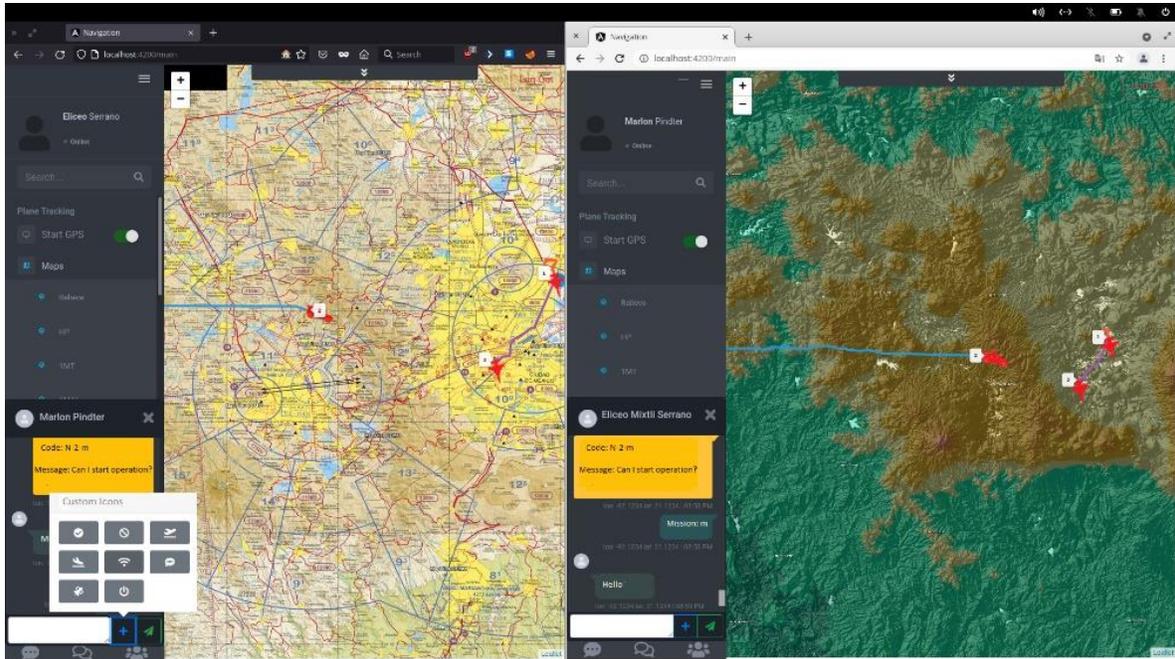


Fig. 52 Pruebas del chat táctico y varias aeronaves

4.3 Resultados de las pruebas unitarias.

En la etapa dos se implementaron las distintas técnicas de Testeo a cada uno de los componentes, servicios y se generó el reporte de calidad del software, véase la Tabla 5.

Tabla 5 Etapa uno de pruebas unitarias del software

Actividad	Descripción
Analizar los diagramas de flujo	Se llevo a cabo el análisis de los diagramas que componen el sistema, con la finalidad de conocer los componentes que dependen de la inyección de datos de otros componentes.
Implementar pruebas de Unidad en los componentes	Se realizaron las pruebas de correcto despliegue de cada una de las funciones contenidas en cada componente.
Implementar pruebas de Unidad en los servicios	Se implementaron las pruebas a los servicios que componen el sistema, se usaron las siguientes técnicas de Testeo usando los servicios: real, virtual, spy y testeo de llamadas asíncronas.

Implementar pruebas de validación	La ejecución de esta prueba consistió en validar cada uno de los campos del sistema para intentar encontrar irregularidades en el mismo, así también como validar que los campos solo acepten el tipo de dato establecido.
Implementar pruebas de seguridad	Se llevó a cabo la comprobación de cada uno de los hipervínculos para validar que el usuario no puede entrar a un módulo sin antes autenticarse, que el usuario no pueda acceder al sitio colocando únicamente o cambiando el hipervínculo y validar que los hipervínculos no estén “rotos”.
Implementar pruebas de despliegue	Esta prueba consistió en la visualización del sistema en diferentes navegadores. Así como también diversos Sistemas Operativos.
Implementar pruebas de Interfaz Gráfica de Usuario	Esta prueba consistió en evaluar los componentes de la interfaz y la interacción del usuario final con el sistema.
Implementar pruebas de llamadas HTTP	Esta prueba consistió en comprobar que la petición se ha realizado correctamente.

Se enlistan las distintas pruebas realizadas al sistema en la parte de comunicación con los servicios de NodeJs como indica la Tabla 6 y en la interfaz gráfica de usuario (GUI) como se observa en la Tabla 7.

Tabla 6 Estado de las pruebas en servicios

Pruebas de Servicios		
Aspecto	Descripción	Estado
Pruebas de Validación	Validar cada campo, de cada módulo del sistema y comparar si cumple los requisitos con respecto al diccionario de datos del sistema.	ok
Pruebas de Seguridad	Validar que el usuario no pueda acceder al sitio colocando únicamente o cambiando el hipervínculo y validar que los hipervínculos no estén “rotos”.	ok
Pruebas de despliegue	validar el desempeño del sistema en diferentes navegadores Web y diferentes sistemas operativos	ok
Pruebas de llamadas HTTP	Comprobar que la petición se ha realizado correctamente y verificar que la información que llega de la API es correcta.	ok

Tabla 7 Estado de las pruebas GUI

Pruebas de GUI		
Navegación sencilla	Validar que el esquema de navegación sea sencillo y fácil de entender para evitar confusiones por los usuarios	ok
Resoluciones de pantalla	Validar que la aplicación se adapte a las diferentes resoluciones de pantalla existentes	ok

Capítulo V

Conclusiones

En este capítulo se presentan las conclusiones después de haber realizado la prueba de recepción de datos y visualización de la aeronave en movimiento, así como el rendimiento de la aplicación y almacenamiento de datos geoespaciales.

Una de las necesidades más importantes para la Secretaría de Defensa Nacional (SEDENA) es mantener la comunicación constantes con las aeronaves, teniendo acceso a la ubicación actual, así como la capacidad de visualizar el rastro en caso de que se encuentren en movimiento, todo esto en tiempo real, por lo cual es indispensable mantener un sistema de comunicación estable entre la aeronave y la estación terrena, esto genera la necesidad de contar con un sistema web para la correcta visualización de los datos requeridos.

Se presentará el desarrollo de un sistema de web para la visualización en tiempo real de las aeronaves conformado por dos módulos diferentes, los cuales serán incorporados para trabajar en conjunto con el módulo para la comunicación en tiempo real por chat mediante el uso de SMS.

Se propuso desarrollar e implementar un sistema de geolocalización en tiempo real para la Fuerza Aérea Mexicana (FAM) el cual facilitará la correcta visualización de las aeronaves en vuelo, así como la ubicación de las diferentes estaciones terrenas, ofreciendo la capacidad de seleccionar los mapas requeridos por el usuario.

El presente proyecto se basó en la metodología de desarrollo SCRUM, donde a través de los sprints propuestos se determinó el análisis de requerimientos, diseño de la arquitectura del software, evaluación del módulo, así como la retrospectiva para conocer a detalle las modificaciones.

También, se evaluó el funcionamiento de la versión final de la aplicación web en conjunto con una aeronave de la FAM para proporcionar datos en tiempo real, detallando cada una de las pruebas con los resultados obtenidos del transceptor dentro del vehículo, visualizando en tiempo real las coordenadas y mensajes transmitidos por el dispositivo.

Con la implementación de las técnicas de Testeo realizadas en el proyecto se comprobó que es un sistema factible para los interesados en llevar a cabo un producto comercial viable, debido a que puede adaptarse fácilmente a extensiones y modificaciones de hardware y software al contar con una arquitectura definida de integración con los servicios utilizados y los protocolos de transmisión GPS, GSM, HTTP.

Por otra parte se demostró que cualquiera de los métodos propuestos es viable ya que aunque el método 2 presentó un tiempo de carga mayor por 6,2 segundos y un consumo de recursos de 10.98 Mb mayor con respecto al método 1 la diferencia es mínima.

De igual manera, el metodo 2 resulta ser mas eficiente, si lo recursos de hardware son limitados y no se cuenta con la cantidad de memoria suficiente para almacenar los mapas de manera local.

Por conclusión se determinó que la aplicación desarrollada cuenta con mejoras notables respecto al prototipo PINA actualmente ocupado por la FAM. Las cuales son las siguientes:

- Mejora de rendimiento al separar el cliente del servidor
- Seguridad en peticiones HTTP
- Estabilidad en la aplicación

5.2 Recomendaciones

Actualmente se traza la ruta de seguimiento de una forma muy recta, se recomienda implementar un algoritmo para predecir los puntos de curva donde la aeronave hizo un giro.

Existe la posibilidad de perder comunicación por temas de cobertura o clima, por lo que se ofrece la posibilidad de implementar una técnica de inteligencia artificial la cual recupere los últimos datos obtenidos y en base a esto generar una posible ruta lo más exacta al movimiento de la aeronave.

Bibliografía

- [1 J. González Valiente, «Desarrollo de una aplicación web para el tratamiento de datos de navegación de una aeronave no tripulada,» 2021.
- [2 F. A. G. Ornelas, «<https://infotec.repositorioinstitucional.mx/>,» 2020. [En línea]. Available: https://infotec.repositorioinstitucional.mx/jspui/bitstream/1027/402/1/INFOTEC_MGITIC_FAGO_27082020.pdf. [Último acceso: 4 Abril 2022].
- [3 R. Gurumendi Alvarado , «Diseño e implementación de dispositivo de rastreo satelital inteligente GPS mediante microcontrolador programable con activación de apagado automático mediante zonas poligonales en Google Maps,» 2019.
- [4 J. J. Ascencio Vargas, «Diseño de un chatot para mejorar la calidad de servicio al cliente en la empresa Fans Store S.A.C,» Lima, 2019.
- [5 L. F. Garcia Reina, «Asistente Virtual Tipo Chat Bot,» Bogotá, 2018.
- [6 J. Anchundia Soza , 2017.
- [7 J. Abbud Momma, «Localización y seguimiento de aeronaves mediante sistemas de multilateración de área extensa,» 2015.
- [8 IBM, «<https://www.ibm.com/>,» 20 Diciembre 2021. [En línea]. Available: <https://www.ibm.com/mx-es/analytics/database-management>.
- [9 MariaDB Foundation, «<https://mariadb.org/>,» 20 Diciembre 2021. [En línea]. Available: <https://mariadb.org/>.
- [1 MDN Web Docks, «<https://developer.mozilla.org/>,» 10 Febrero 2022. [En línea]. Available: https://developer.mozilla.org/es/docs/Learn/Common_questions/What_is_a_web_server. [Último acceso: 11 Febrero 2022].
- [1 The Apache Software Foundation, «<https://httpd.apache.org/>,» 20 Diciembre 2021. [En línea]. Available: <https://httpd.apache.org/>.
- [1 A. Morales, «<https://mappinggis.com/>,» Septiembre 2017. [En línea]. Available: <https://mappinggis.com/2017/09/que-son-los-vector-tiles-y-como-generarlos-con-geoserver/>.
- [1 Open Source Geospatial Foundation, «<http://geoserver.org/>,» Enero 2022. [En línea]. Available: <http://geoserver.org/about/>. [Último acceso: 11 Febrero 2022].

- [1 J. D. Cascón Katchadourian y A. Á. Ruiz Rodríguez, «Descripción y valoración del software
4] MapTiler:del mapa escaneado a la capa interactiva publicada en la Web.,» MapTiler, 2016. [En línea]. Available:
https://digibug.ugr.es/bitstream/handle/10481/43542/Cascon_Maptiler.pdf?sequence=1&isAllowed=y. [Último acceso: 11 Febrero 2022].
- [1 MDN Web Docs, «<https://developer.mozilla.org>,» 2022 Febrero 10. [En línea]. Available:
5] <https://developer.mozilla.org/es/docs/Web/JavaScript>. [Último acceso: 11 Febrero 2022].
- [1 OpenJS Foundation, «<https://nodejs.org/es/>,» 20 Diciembre 2021. [En línea]. Available:
6] <https://nodejs.org/es/>.
- [1 Microsoft, «<https://www.typescriptlang.org>,» 20 Diciembre 2021. [En línea]. Available:
7] <https://www.typescriptlang.org/es/>.
- [1 TypeORM, «<https://typeorm.io/#/>,» 20 Diciembre 2021. [En línea]. Available:
8] <https://typeorm.io/#/>.
- [1 OpenJS Foundation, «<https://expressjs.com/es/>,» 20 Diciembre 2021. [En línea]. Available:
9] <https://expressjs.com/es/>.
- [2 A. angular, «Angular guide,» 08 Marzo 2021. [En línea]. Available:
0] <https://angular.io/guide/what-is-angular>.
- [2 W. A. S. Inc, «Introduction to Testing Angular,» s.f..
1]
- [2 E. Hahn, JavaScript Testing with Jasmine, USA: O'REILLY, 2013.
2]
- [2 V. Agafonkin, «<https://leafletjs.com/>,» 4 Septiembre 2020. [En línea]. Available:
3] <https://leafletjs.com/>.
- [2 GPS.GOV., «www.gps.gov,» 09 08 2021. [En línea]. Available:
4] <https://www.gps.gov/systems/gps/control/>. [Último acceso: 01 02 2022].
- [2 NASA, «Constelación de satélites,» 2015. [En línea]. Available:
5] <https://www.gps.gov/system/gps/control/>. [Último acceso: 02 04 2021].
- [2 J. Capdevila, «ansenuza.unc.edu.ar,» [En línea]. Available:
6] <https://ansenuza.unc.edu.ar/comunidades/bitstream/handle/11086.1/1258/EI%20Sistema%20Ode%20Posicionamiento%20Global.pdf?sequence=1&isAllowed=yz>. [Último acceso: 01 02 2022].
- [2 NASA, «GPS.GOV,» 2014. [En línea]. Available: <https://www.gps.gov/systems/gps/control/>.
7] [Último acceso: 02 04 2021].

- [2 L. G. AG, «Coello.ujaen.es,» 2004. [En línea]. Available:
8] http://coello.ujaen.es/Asignaturas/topometria/descargas/GPSBasics_es.pdf. [Último acceso: 01 02 2022].
- [2 INEGI, «/www3.inegi.org.mx,» s.f.. [En línea]. Available:
9] http://www3.inegi.org.mx/contenidos/temas/mapas/geodesia/metadatos/sistema_de_posicionamiento_global.pdf. [Último acceso: 01 02 2022].
- [3 P. Dana, «ecoursesonline,» 17 08 1994. [En línea]. Available:
0] <http://ecoursesonline.iasri.res.in/mod/page/view.php?id=2080>. [Último acceso: 03 04 2021].
- [3 C. Jurado, «CCM,» Febrero 2021. [En línea]. Available: <https://es.ccm.net/contents/681-1>
1] estandar-gsm-sistema-global-de-comunicaciones-moviles.
- [3 j. Pascual, J. Molina y L. Llácer, SISTEMAS DE COMUNICACIONES MÓVILES, Cartagena:
2] Universidad Politécnica de Cartagena, 2014.
- [3 T. Satpathy, Guía SBOK™, Phoenix: SCRUMstudy™, 2013.
3]
- [3 Banco Santander, «<https://www.becas-santander.com>,» Enero 2022. [En línea]. Available:
4] <https://www.becas-santander.com/es/blog/metodologias-desarrollo-software.html>. [Último acceso: 2022 Febrero 11].
- [3 T. Satpathy, Guía SBOK™, Phoenix: SCRUMstudy™, 2013.
5]
- [3 K. Schwaber y J. Sutherland, «<https://scrumguides.org>,» Julio 2016. [En línea]. Available:
6] <https://scrumguides.org/docs/scrumguide/v2016/2016-Scrum-Guide-Spanish.pdf#zoom=100>. [Último acceso: 2022 Febrero 11].
- [3 D. West, «<https://www.atlassian.com>,» 2022. [En línea]. Available:
7] <https://www.atlassian.com/es/agile/scrum/sprint-planning>. [Último acceso: 11 Febrero 2022].
- [3 M. Garcia, «<https://ittude.com.ar>,» 13 Abril 2020. [En línea]. Available:
8] <https://ittude.com.ar/b/scrum/que-es-la-daily-scrum/>. [Último acceso: 2022 Febrero 11].
- [3 M. Garcia, «<https://ittude.com.ar>,» 25 Mayo 2020. [En línea]. Available:
9] <https://ittude.com.ar/b/scrum/que-es-el-sprint-review/>. [Último acceso: 2022 Febrero 11].
- [4 O. Levy, «<https://ittude.com.ar>,» 23 Abril 2020. [En línea]. Available:
0] <https://ittude.com.ar/b/scrum/retrospectiva/>. [Último acceso: 11 Febrero 2022].