

GeoCuadrilla: Desarrollo Front-end de panel web para la gestión y supervisión de cuadrillas en terreno en Chilquinta Transmisión

Loreto Gálvez Camilo

loreto.galvez@usm.cl

Catherine Gómez
Profesora Guía

David Larrondo
Profesor Correferente

Resumen: El presente trabajo se desarrolla en el contexto de la gestión operativa de cuadrillas en terreno dentro del sector eléctrico, donde la coordinación de actividades y el acceso oportuno a información actualizada resultan fundamentales para el correcto desarrollo de las operaciones. En este escenario, la inexistencia de una plataforma que permita visualizar de forma integrada la ubicación de las cuadrillas junto con información relevante de la red eléctrica dificulta el monitoreo y la supervisión de los equipos de trabajo. A partir de esta problemática, el proyecto se orienta al desarrollo del sistema GeoCuadrilla, cuyo propósito es proporcionar una plataforma web que concentre y organice la información operativa asociada a las cuadrillas y a la infraestructura eléctrica. La solución permite representar de manera visual la ubicación de las cuadrillas sobre un mapa, incorporando además capas de información de la red mediante archivos KMZ, lo que facilita una comprensión global del estado operativo. Como resultado, se obtuvo un panel web que centraliza la información relevante del sistema en una única plataforma, permitiendo una visualización clara y ordenada del trabajo en terreno y del entorno en el que se desarrollan las actividades operativas.

Palabras Clave: Geolocalización, gestión operativa, capas KMZ, integración de sistemas, interfaz web.



CONSTANCIA DE VALIDACIÓN Y CONFIDENCIALIDAD DE MONOGRAFÍA A REPOSITORIO ACADÉMICO

1.- IDENTIFICACIÓN DEL TRABAJO ACADÉMICO

Tipo de monografía (marcar una opción): ☒ Memoria o trabajo de título ☐ Tesis de Postgrado

Título del trabajo: GeoCuadrilla: Desarrollo Front-end de panel web para la gestión y supervisión de cuadrillas en terreno en Chilquinta Transmisión

Nombre del candidato(a): Loreto Alejandra Gálvez Camilo

Carrera / Grado: Ingeniería en Informática

Campus: Viña del Mar

Departamento: Electrotecnia e Informática

2.- VALIDACIÓN DEL PROFESOR GUÍA/DIRECTOR DE TESIS

Yo, Catherine Gómez B., en mi calidad de profesor(a) guía/director(a) del trabajo académico mencionado anteriormente **DEJO CONSTANCIA** que:

- He revisado esta versión del documento y corresponde a la versión final aprobada del trabajo.
- El trabajo cumple con los requisitos académicos y de formato establecidos por la institución.

3.- EVALUACIÓN DE CONFIDENCIALIDAD POR PROPIEDAD INDUSTRIAL (marcar una opción)

☒ El trabajo **NO contiene** información que amerite confidencialidad y puede ser publicado de inmediato en repositorio con acceso abierto.

☐ El trabajo **CONTIENE** información con potenciales implicancias de propiedad industrial o intelectual y requiere un periodo de confidencialidad (**embargo**) por (**marcar una opción**):

☐ 6 meses ☐ 12 meses ☐ 2 años ☐ 3 años ☐ 5 años ☐ 10 años

Fundamentación de la necesidad de confidencialidad (obligatorio si se solicita embargo):

4.- FIRMAS

Profesor(a) guía o director(a) de memoria o tesis:

Fecha: 24-09-2026

Firma:

Catherine BC

Estudiante o Candidato(a):

Fecha: 24 / 09 / 2026

Firma:

Loreto G

Este formulario debe ser insertado como página 2 de la memoria o tesis, completado y firmado por estudiante y profesor(a) antes de la entrega en portal PRISMA de Biblioteca USM.

1 Introducción

1.1 Contexto y relevancia del desarrollo de componentes específicos en el proyecto

En el sector de transmisión y distribución eléctrica, la gestión de cuadrillas de trabajo es un elemento clave para garantizar la continuidad y la calidad del servicio. Estas cuadrillas son las encargadas de ejecutar labores de mantenimiento, responder a emergencias y resolver fallas que puedan afectar el suministro eléctrico. En el contexto actual de operación, la empresa cuenta con cuatro vehículos activos, los cuales deben movilizarse constantemente por extensas zonas para atender incidentes en distintos puntos de la red de transmisión.

Una de las principales dificultades radica en la falta de herramientas digitales que permitan conocer en tiempo real la ubicación de las cuadrillas, su estado operativo y su disponibilidad, junto con información que facilite la coordinación y el seguimiento de sus desplazamientos. En muchos casos, estos datos se encuentran dispersos o requieren de verificación manual, lo que genera demoras en la toma de decisiones y afecta tanto los tiempos de respuesta como los costos operativos.

Frente a esta situación, Chilquinta requiere implementar un sistema que integre de manera eficiente la información operativa y geográfica, con el objetivo de entregar una visión global del estado de las cuadrillas y de la red eléctrica. Si bien el escenario actual considera un número acotado de vehículos, la solución propuesta no define un límite funcional explícito para la cantidad de cuadrillas gestionadas. No obstante, su escalabilidad dependerá de factores como la capacidad del backend, la infraestructura disponible, el volumen de actualizaciones en tiempo real y las condiciones de uso de servicios externos como Google Maps.

1.2 Definición del problema.

Actualmente, la gestión de cuadrillas presenta diversas limitaciones que afectan directamente el desempeño organizacional de la empresa. Una de las principales dificultades se refleja en la asignación de cuadrillas, la cual se ve dificultada por la falta de visibilidad sobre la ubicación y el estado real de los equipos en terreno. Esa situación genera retrasos al momento de decidir qué cuadrilla enviar a un incidente específico, lo que prolonga innecesariamente los tiempos de respuesta.

La ausencia de rutas claras y eficientes hacia los puntos de falla obliga, en algunos casos, a que las cuadrillas recorran trayectos más extensos o se enfrenten a condiciones de trabajo no previstas, disminuyendo la capacidad de atención oportuna de emergencias.

La falta de trazabilidad de jornadas laborales es otro aspecto crítico ya que en la actualidad este proceso se gestiona mediante Power Apps. Si bien esta herramienta permite digitalizar parte de la información, al no estar integrada con otros sistemas los datos deben consultarse en plataformas separadas, lo que fragmenta la información y dificulta la supervisión efectiva de las cuadrillas.

1.3 Breve descripción sobre la propuesta de solución

Para resolver las limitaciones actuales en la gestión de cuadrillas, se plantea el desarrollo de un sistema compuesto por dos componentes principales: una aplicación móvil y un panel web. La aplicación móvil está destinada al personal en terreno y permite realizar el registro del inicio y término de la jornada laboral, enviar ubicación en tiempo real, visualizar rutas asignadas y mantener contacto con la empresa. Por otra parte, el panel web constituye la herramienta central de supervisión para Chilquinta, ya que concentra la información geoespacial y operativa de las cuadrillas y toda su información,

facilitando la toma de decisiones respecto a la asignación y al desplazamiento de los equipos hacia los puntos de falla.

En este proyecto el desarrollo de front-end se plantea como una solución que permita integrar en un mismo espacio la visualización de cuadrillas en mapas interactivos y la administración del personal, con un diseño intuitivo y eficiente. De esta manera, se logra centralizar en un único panel la información que hoy se encuentra distribuida en múltiples sistemas, mejorando la experiencia de usuario y apoyando la toma de decisiones operativas en Chilquinta.

Además, el diseño del panel web se desarrolló considerando la escalabilidad del sistema. La estructura basada en componentes de Angular permite incorporar nuevas vistas, funcionalidades o conjuntos de datos sin afectar el rendimiento general. A nivel de servidor, la comunicación mediante APIs REST favorece principalmente la escalabilidad vertical, permitiendo aumentar los recursos del servidor si la cantidad de cuadrillas o solicitudes crece.

Aunque actualmente el sistema se ejecuta sobre un único equipo, su estructura modular y flexible permite proyectar una escalabilidad horizontal futura, mediante la distribución de las solicitudes en múltiples instancias si el número de usuarios o el volumen de información llegara a incrementarse.

Esta flexibilidad asegura que el sistema pueda adaptarse a una futura expansión de usuarios, zonas geográficas o nuevas fuentes de información sin requerir una modificación completa de la plataforma.

Si bien la solución desarrollada permite la integración de información en tiempo real y la visualización unificada de cuadrillas mediante la API de Google Maps, su uso actual se encuentra orientado al entorno interno de la empresa. Una eventual ampliación hacia otras zonas geográficas o un mayor volumen de usuarios no implica una limitación funcional del sistema, pero sí requeriría considerar aspectos asociados al uso eficiente de la API de Google Maps y a la capacidad de la infraestructura del servidor, con el fin de mantener un rendimiento estable y una correcta visualización de la información.

1.4 Objetivos Generales y Específicos de la Tesina

Objetivo General:

Diseñar e implementar una interfaz web interactiva y responsiva para la gestión de cuadrillas en Chilquinta, que permita visualizar en tiempo real la ubicación y estado de las brigadas, integrando información operativa en un único panel, con el fin de apoyar la asignación eficiente de equipos y facilitar la toma de decisiones relacionadas con la atención de incidencias en la red eléctrica.

En el alcance de esta tesina, el aporte a la eficiencia operativa se aborda desde una perspectiva funcional, a través de la centralización de información relevante para la toma de decisiones. Si bien no se realiza una medición cuantitativa antes y después de la implementación, se identifican indicadores que podrían utilizarse en una etapa posterior para evaluar el impacto del sistema, tales como el tiempo de asignación de cuadrilla, el tiempo de desplazamiento hacia el punto de incidencia y la disponibilidad operativa de equipos en terreno.

Los **objetivos específicos** planteados para la interfaz web son:

- **Definir** los requerimientos funcionales del front-end de GeoCuadrilla a partir del análisis de las necesidades del sistema
- **Diseñar** una interfaz intuitiva y responsiva que facilite la navegación y asegure la correcta visualización de la información en distintos dispositivos, procurando una experiencia de usuario eficiente.

- **Implementar** un mapa interactivo que muestre en tiempo real la ubicación de las cuadrillas e incorpore las líneas de transmisión eléctrica mediante archivos KMZ, con opciones de filtrado por nivel de tensión para facilitar su interpretación.
- **Integrar** el panel web con la API de Google Maps para la visualización y gestión de geolocalización en tiempo real, procurando que la información presentada sea consistente y se actualice constantemente.
- **Integrar** archivos KMZ en el panel web para complementar la visualización de cuadrillas con información geográfica relevante de la red eléctrica.
- **Desarrollar** un módulo de gestión de cuadrillas que permita consultar su estado (activas o inactivas), así como la información detallada de los integrantes, incluyendo su rol y datos de contacto, asegurando una visualización clara y accesible.
- **Analizar** cualitativamente la usabilidad del panel web mediante la revisión de sus principales flujos de interacción, con el fin de identificar mejoras en la experiencia de uso y verificar la claridad de la interfaz para el monitoreo de cuadrillas y la gestión de capas geográficas.

1.5 Justificación de proyecto

El sistema de GeoCuadrilla cumple un rol central al actuar como el punto de interacción entre los usuarios y la información operativa del sistema. A través del panel web, los usuarios pueden acceder de manera clara y ordenada a datos relevantes sobre el estado de las cuadrillas y su ubicación, lo que facilita la supervisión y el seguimiento de las operaciones en terreno.

Desde el punto de vista técnico, el proyecto aporta al desarrollo de interfaces web orientadas a la gestión y visualización de información dinámica, permitiendo que los usuarios trabajen con datos actualizados sin depender de procesos manuales o de múltiples herramientas externas. Esto contribuye a una experiencia de uso más fluida y a una mejor comprensión del estado general de las operaciones.

A nivel del área de desarrollo de software front-end, el proyecto contribuye como un ejemplo de aplicación web enfocada en resolver una necesidad concreta del ambiente empresarial, abordando desafíos habituales como la actualización continua de la información y la organización visual de datos provenientes de distintos componentes del sistema.

Uno de los principales aportes del sistema corresponde a la centralización de la información operativa, al permitir que la información relevante se encuentre disponible en un solo panel, accesible desde distintos dispositivos. Esto puede contribuir a reducir la dependencia de reportes manuales y apoyar una respuesta más oportuna ante emergencias, al facilitar una supervisión más directa de las cuadrillas en terreno.

En términos de usabilidad, la interfaz fue diseñada para ser intuitiva y fácil de utilizar, priorizando una navegación clara y el uso de elementos visuales que ayuden a identificar rápidamente el estado de las cuadrillas y la información asociada. Estos elementos buscan facilitar la interpretación de la información y mejorar la experiencia general de los usuarios.

Otro beneficio importante es la posibilidad de adaptación y crecimiento del sistema. El sistema permite incorporar ajustes y mejoras a partir de la retroalimentación recibida, lo que favorece la continuidad del proyecto y su adecuación a nuevas necesidades operativas a lo largo del tiempo.

Finalmente, el proyecto introduce una forma moderna de abordar la visualización y gestión de información dentro de un panel web, integrando distintos tipos de datos en una sola vista. El desarrollo se realizó de manera progresiva, incorporando retroalimentación a lo largo del proceso, lo que permitió adaptar la interfaz a necesidades reales y fortalecer su utilidad práctica.

1.6 Metodología

El desarrollo del sistema se llevó a cabo mediante la adaptación de la metodología ágil Scrum, marco seleccionado por su capacidad de respuesta ante requerimientos dinámicos y su enfoque en la entrega incremental de valor. El ciclo de vida del proyecto se estructuró en tres iteraciones o Sprints de trabajo, distribuidos estratégicamente a lo largo de las fases de análisis, desarrollo e integración. Esta planificación permitió articular el avance técnico en hitos verificables y facilitar la validación continua de los requerimientos junto con la empresa patrocinadora, Chilquinta Transmisión.

- **Sprint 1:** Esta iteración inicial abarcó desde la definición conceptual del sistema hasta la habilitación de su estructura base. Se comenzó con el análisis de las especificaciones funcionales y el diseño de los prototipos de interfaz, los cuales sirvieron de pauta para la construcción de los primeros módulos. A partir de estas definiciones, se implementó el panel web inicial con la carga del mapa principal, la visualización básica de cuadrillas y la estructura de navegación. En paralelo, en la plataforma móvil se desarrollaron los módulos de autenticación, el control de inicio y término de jornada, y la captura de coordenadas geográficas. Asimismo, se habilitaron los endpoints iniciales en el backend y se gestionó la configuración del servidor privado necesario para soportar la arquitectura. El ciclo concluyó con una revisión con el patrocinador, quien validó los avances y aportó observaciones para la siguiente fase.
- **Sprint 2:** Esta fase estuvo enfocada en la incorporación de funcionalidades clave y en el establecimiento de la integración entre plataformas. En el panel web se integró el despliegue y filtrado de archivos KMZ para la representación visual de las líneas de transmisión eléctrica. De forma simultánea, se construyó un canal de comunicación en tiempo real que vinculó la aplicación móvil con el panel central, permitiendo el despliegue inmediato de la ubicación de los técnicos sobre el mapa. Toda la transmisión de datos fue estabilizada en el servidor privado para asegurar la fluidez de la información. Durante la evaluación de este Sprint, se identificaron nuevas necesidades operativas junto al patrocinador, derivando en la especificación de herramientas de apoyo.
- **Sprint 3:** La última iteración estuvo dedicada a la evaluación funcional, depuración y consolidación de la solución. Se ejecutaron pruebas de integración simulando escenarios reales de uso mediante el empleo de dispositivos móviles y equipos en un ambiente controlado. A partir de los hallazgos técnicos y la retroalimentación directa del equipo operativo de la empresa, se aplicaron optimizaciones en la interfaz del panel web, ajustando componentes visuales y refinando la presentación de la información. Esta etapa final permitió asegurar la consistencia del sistema y su alineación con los requerimientos operativos dentro de los plazos establecidos.

1.7 Breve descripción de la organización del informe en capítulos.

Capítulo 1. Introducción: Describe el contexto del proyecto, la problemática abordada, los objetivos y el alcance definidos para el trabajo realizado.

Capítulo 2: Marco teórico: Aborda los conceptos necesarios para comprender el desarrollo del sistema.

Capítulo 3. Diseño e implementación: Detalla el desarrollo del front-end del sistema GeoCuadrilla, abordando la arquitectura adoptada, la integración con el sistema general, el diseño de la interfaz y los principales aspectos implementados, finalizando con las pruebas y validación.

Capítulo 4: Conclusiones y trabajo futuro

2 Marco Teórico

2.1 Fundamentos del desarrollo front-end

El Front-end constituye la capa del sistema encargada de la interacción directa con el usuario. Su propósito es mostrar la información de manera clara y permitir que las acciones del usuario se traduzcan en operaciones dentro de la aplicación. Está compuesto por la interfaz de usuario (UI) y por los mecanismos que garantizan una experiencia de uso adecuada (UX). De acuerdo con Sommerville [5], esta capa cumple el rol de presentación dentro de la arquitectura de software, siendo responsable de la comunicación entre el usuario y la lógica interna del sistema.

Históricamente, el desarrollo front-end surgió en la década de 1990 con la expansión de los navegadores web, los cuales introdujeron los primeros lenguajes de marcado y estilo como HTML y CSS. Estos permitieron estructurar y dar formato a los contenidos en la web. Posteriormente, la incorporación de JavaScript permitió generar interacción dinámica, marcando un avance significativo en la creación de interfaces más intuitivas.

Actualmente, el front-end se implementa en distintos tipos de sistemas, como aplicaciones web, móviles, aplicaciones de escritorio y dispositivos con interfaces gráficas integradas. El front-end se caracteriza por tres pilares fundamentales: la interactividad, que permite al usuario comunicarse con el sistema; la portabilidad, que facilita su ejecución en múltiples dispositivos; y la adaptabilidad, que garantiza una visualización óptima mediante diseño responsivo.

Las ventajas del front-end moderno incluyen la mejora en la experiencia de usuario, la reducción de carga en el servidor y la capacidad de actualización en tiempo real. No obstante, presenta también limitaciones, como la dependencia del rendimiento del navegador, los riesgos de seguridad del lado cliente y la complejidad del entorno de desarrollo actual.

En el contexto de la arquitectura de software, el front-end funciona como la capa de presentación, responsable de transformar la información generada en el back-end en elementos visuales para el usuario final. Esta comunicación se logra mediante APIs RESTful, que permiten el intercambio de información en formato JSON a través del protocolo HTTP [2].

En el caso del panel web de Chilquinta, el front-end cumple un rol esencial al ofrecer una interfaz que muestra en tiempo real la ubicación y el estado de las cuadrillas, consumiendo datos desde la API de Google Maps y desde servicios internos de la empresa. Gracias a su diseño visual e interactivo, la aplicación permite que las acciones del usuario, como seleccionar una cuadrilla o filtrar información, se traduzcan directamente en operaciones dentro del sistema.

2.2 Arquitectura de software

La arquitectura del sistema se compone de una interfaz web desarrollada en Angular, que actúa como cliente y consume los servicios del back-end mediante APIs REST. Esta separación entre la capa visual y la lógica del sistema permite mantener un código más ordenado y escalable.

El diseño del panel web se estructura bajo una arquitectura modular, en la cual cada parte del sistema, como el mapa y la información de las cuadrillas, se implementa como un componente independiente.

Este enfoque permite desarrollar, probar y actualizar cada sección sin afectar al resto del sistema, promoviendo la reutilización y facilitando la incorporación de nuevas funcionalidades en el futuro [1].

El sistema también aplica los principios de responsabilidad única y desacoplamiento, garantizando que cada módulo cumpla una función específica dentro de la aplicación. Además, se emplea un sistema de enrutamiento interno que gestiona las distintas vistas sin necesidad de recargar toda la página, lo que mejora el rendimiento y la experiencia de usuario [4].

En el caso del proyecto, esta estructura modular y la comunicación basada en servicios permiten que la empresa centralice toda la información operativa en un mismo entorno visual, reduciendo la dependencia de múltiples plataformas y proporcionando información centralizada como apoyo a la toma de decisiones.

2.3 Tecnologías y frameworks utilizados

El front-end se construyó con las siguientes tecnologías:

- Angular se utilizó como framework principal del proyecto, ya que su arquitectura basada en componentes y su modularidad facilitaron la construcción de un sistema escalable, organizado y con integración fluida hacia servicios externos mediante APIs.
- HTML, CSS y JavaScript conformaron la base del desarrollo, permitiendo estructurar el contenido del sistema, definir los estilos visuales e incorporar interactividad a la interfaz, garantizando así una experiencia de usuario más completa.
- TypeScript fue incorporado como lenguaje complementario, aportando tipado estático y detección temprana de errores, lo que mejoró la mantenibilidad y calidad del código en el entorno de Angular.
- La API de Google Maps se integró como soporte para la visualización geográfica, permitiendo desplegar mapas interactivos que sirvieron como base para la supervisión de cuadrillas en el panel web [3].
- Finalmente, los archivos KMZ se integraron al sistema como capas geoespaciales que representan elementos relevantes de la red eléctrica, tales como líneas de transmisión y subestaciones. Estos archivos son cargados directamente desde el panel web y enviados al backend mediante solicitudes HTTP, el cual se encarga de su almacenamiento y posterior entrega al front-end.

2.4 Consideraciones Normativas sobre Datos Personales

En el contexto de GeoCuadrilla, la aplicación utiliza información vinculada a usuarios y cuadrillas, incluyendo autenticación, correos electrónicos, estado de disponibilidad y ubicación geográfica. Si bien la ubicación en sí misma no siempre identifica directamente a una persona, puede transformarse en un dato personal cuando se asocia a un técnico, usuario o integrante de una cuadrilla determinada. Por ello, su tratamiento debe realizarse bajo criterios de finalidad, proporcionalidad y seguridad, limitando su uso a los objetivos operacionales definidos para el sistema.

La Ley N° 19.628 establece, en su artículo 4, que el tratamiento de datos personales debe contar con autorización legal o consentimiento del titular, quien además debe ser informado respecto del propósito del almacenamiento y eventual comunicación de sus datos. Asimismo, el artículo 2 letra g define los datos sensibles como aquellos relacionados con características personales o hechos de la

vida privada o intimidad. Aunque la geolocalización operacional de cuadrillas no se menciona expresamente como dato sensible en la ley vigente, sí puede involucrar aspectos de la vida privada cuando permite reconstruir desplazamientos, rutinas o permanencia de una persona en determinados lugares [7].

Por otra parte, el artículo 10 de la misma ley establece restricciones para el tratamiento de datos sensibles, permitiéndose sólo cuando exista autorización legal, consentimiento del titular o alguna causal expresamente contemplada. En consecuencia, para una implementación productiva de GeoCuadrilla, resulta recomendable establecer políticas internas claras sobre el tratamiento de la ubicación, definiendo quiénes pueden acceder a estos datos, durante cuánto tiempo se almacenan, con qué finalidad se utilizan y qué medidas de seguridad se aplican para evitar accesos no autorizados.

Desde esta perspectiva, la aplicación incorpora algunos mecanismos técnicos coherentes con la protección de datos, tales como autenticación de usuarios, uso de token de acceso, protección de rutas internas e incorporación automática de autorización en las solicitudes al backend.

Como mejora futura, se propone realizar una revisión formal de cumplimiento normativo antes de la implementación productiva del sistema. Esta revisión debería considerar la Ley N° 19.628 sobre Protección de la Vida Privada, las modificaciones incorporadas por la Ley N° 21.719 y las políticas internas de la empresa sobre tratamiento de datos personales, especialmente en relación con la geolocalización, identificación de usuarios, registros de actividad y control de acceso a la información [8].

3 Diseño e Implementación

3.1 Diseño de Componentes

3.1.1 Arquitectura y estructura de los componentes.

La Ilustración 1 muestra el diagrama de componentes del front-end de GeoCuadrilla.

Este diagrama representa exclusivamente los servicios centrales del sistema y sus relaciones de dependencia. Los componentes visibles forman la capa lógica que sustenta la autenticación, el manejo de archivos y la comunicación en tiempo real mediante SignalR.

El componente AuthService administra el ciclo de autenticación del usuario, incluyendo el envío de credenciales al backend, el almacenamiento del token JWT en localStorage y la verificación del estado de sesión. Tanto el authGuard como el authInterceptor dependen directamente de este servicio; el guard consulta isAuthenticated() para controlar el acceso a las rutas protegidas, mientras que el interceptor obtiene el token mediante getToken() para adjuntarlo a todas las solicitudes HTTP salientes.

3.1.1.1 Diagrama de componente

El front-end de GeoCuadrilla se organiza en una arquitectura modular donde los componentes de interfaz, los servicios de lógica y los sistemas externos se relacionan de manera coordinada. El Diagrama de Componentes muestra cómo la aplicación se sostiene sobre tres pilares principales:

1. Los componentes visuales (LoginComponent y MapShellComponent), representan las vistas de la interfaz.
2. Los servicios especializados (AuthService, FilesService, LocationService), que encapsulan la comunicación con el backend y el manejo de estado.

3. Los mecanismos transversales (authInterceptor y authGuard), que integran seguridad y control de acceso sin acoplarse directamente a los componentes.

La integración ocurre a través del router y del cliente HTTP configurado globalmente. Los componentes no se comunican directamente entre sí, sino que utilizan servicios independientes, manteniendo bajo acoplamiento. Cada servicio se conecta con los sistemas externos correspondientes: la API REST para autenticación y gestión de archivos, y el Hub de SignalR para actualizaciones en tiempo real.

De esta manera, el front-end funciona como una capa organizacional: MapShellComponent consume datos geoespaciales y de localización provenientes de los servicios, y AuthService determina la validez de la sesión. El interceptor agrega el token automáticamente a todas las peticiones, mientras que el guard asegura que sólo usuarios autenticados accedan al mapa.

El resultado es una estructura coherente, donde cada módulo cumple una responsabilidad única y se integra dentro de un flujo de comunicación bien definido.

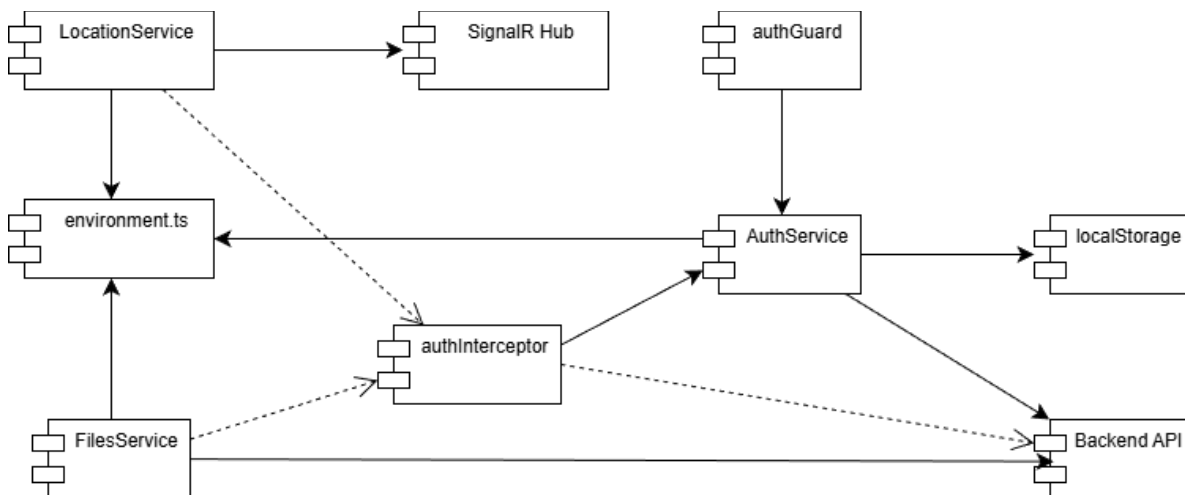


Figura 1: Diagrama de componentes (Servicios).
Fuente: Elaboración propia.

3.1.1.2 Diagrama de clases

En este modelo, MapShellComponent constituye la clase central del sistema, ya que gestiona la visualización del mapa, el renderizado de cuadrillas, la carga y activación de capas geoespaciales (KMZ/KML), y el procesamiento de actualizaciones en tiempo real. Para cumplir estas funciones, utiliza clases auxiliares como CarMarker, LayerEntry, Crew, CrewMember y RealtimeLocationUpdate, las cuales encapsulan la información operativa necesaria para la interfaz.

Por otra parte, los servicios del sistema se representan como clases independientes con responsabilidades claramente delimitadas. AuthService administra el proceso de autenticación y la persistencia de credenciales; FilesService abstrae las operaciones de carga, descarga y eliminación de archivos provenientes del backend; y LocationService establece la conexión con el hub de SignalR y expone las actualizaciones de ubicación mediante flujos observables. Estas clases no comparten estado entre sí y se relacionan únicamente a través del mecanismo de inyección de dependencias, lo que favorece un diseño modular, escalable y de fácil mantenimiento.

Asimismo, el diagrama incorpora entidades orientadas al transporte de datos, como GcFileItem, ListResp y LoginRequest, cuya función es representar estructuras enviadas o recibidas desde el backend sin incorporar lógica adicional.

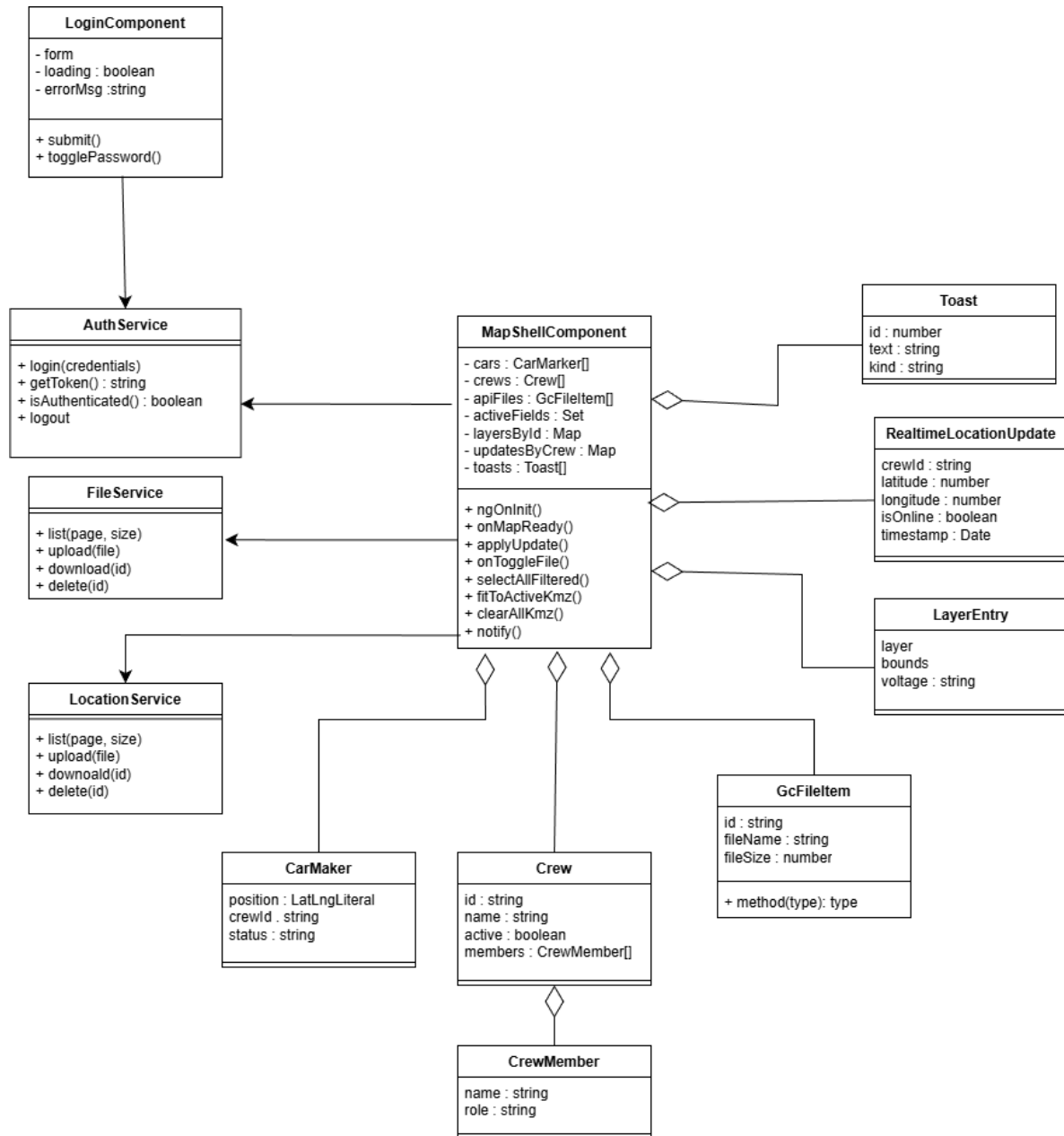


Figura 2: Diagrama de clases.
Fuente: Elaboración propia.

3.1.1.3 Diagrama de casos de uso

Este modelo permite visualizar, de manera global, las funciones que el usuario puede ejecutar y cómo estas se relacionan entre sí al momento de operar la plataforma.

En primer lugar, el sistema requiere que el usuario se autentique a través del caso de uso Iniciar sesión, lo cual habilita el acceso a las funcionalidades internas. Una vez dentro de la plataforma, la acción central corresponde al caso de uso Ver Mapa, que constituye el punto de entrada a la información operativa mostrada en tiempo real. Este caso de uso incluye dos funcionalidades complementarias:

1. **Monitorear ubicación en tiempo real**, que permite recibir y visualizar las posiciones actualizadas de las cuadrillas mediante la integración con el servicio de comunicación en tiempo real.
2. **Ver detalle de cuadrilla**, que muestra la información específica del equipo seleccionado, como personal asignado y estado operativo.

El operador también puede filtrar cuadrillas para facilitar la búsqueda. Este proceso incorpora el caso de uso Ver detalle de cuadrilla, ya que la consulta de información individual es parte del flujo de trabajo tras aplicar filtros.

Otro grupo de funcionalidades corresponde a la administración de capas geoespaciales, representado por el caso de uso Gestionar archivos KMZ. Este caso agrupa tareas que dependen directamente de él, tales como Cargar archivo KMZ, Activar o desactivar capas, y Eliminar archivos KMZ. Estas acciones permiten al operador modificar dinámicamente la visualización de líneas de transmisión y otros elementos geográficos dentro del mapa, otorgando flexibilidad para ajustar la representación del entorno operativo según las necesidades de cada jornada.

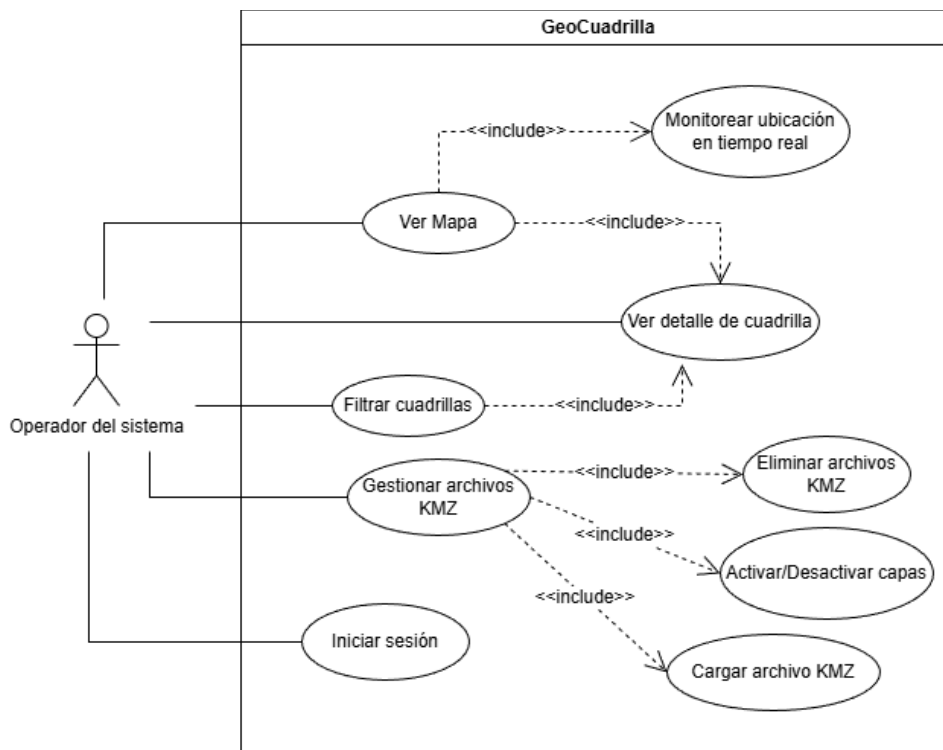


Figura 3: Diagrama de casos de uso.

Fuente: Elaboración propia.

3.1.1.4 Casos de uso

Tabla 1. Caso de uso CU-01: Iniciar sesión.

CU-01	Iniciar sesión		
Objetivo	Permitir que el operador acceda al sistema de forma segura.		
Descripción	Este caso de uso permite al operador ingresar al sistema mediante la pantalla de autenticación. A través de este proceso, el sistema verifica las credenciales ingresadas y habilita el acceso al panel principal cuando la autenticación es correcta.		
Referencias cruzadas	- CU-02 Ver mapa. - CU-03 Monitorear ubicación en tiempo real. - CU-06 Gestionar archivos KMZ.		
Precondición	- El operador cuenta con credenciales registradas. - La aplicación web se encuentra disponible desde el navegador. - El operador no ha iniciado sesión previamente o requiere ingresar nuevamente.		
Secuencia normal	Paso	Acción del operador	Respuesta del sistema
	1	Accede a la aplicación web GeoCuadrilla.	Muestra la pantalla de inicio de sesión.
	2	Ingresa su usuario o correo y contraseña.	Habilita la opción de ingreso cuando los campos requeridos están completos.
	3	Selecciona la opción “Ingresar”.	Muestra un estado de carga mientras se valida el acceso.
	4	Espera la respuesta del sistema.	Si las credenciales son válidas, redirige al operador al panel principal.
	5	Visualiza el panel principal.	Muestra el mapa y las opciones disponibles para la gestión de cuadrillas.
Postcondición	- El operador accede al panel principal del sistema. - La sesión queda activa en el navegador mientras las credenciales sean válidas.		
Secuencia alternativa	Paso	Condición	Respuesta del sistema
	A1	El operador deja campos obligatorios vacíos.	El sistema impide continuar hasta completar los datos requeridos.
	A2	El operador ingresa credenciales inválidas.	El sistema muestra un mensaje de error y mantiene la pantalla de inicio de sesión.

Fuente: Elaboración propia.

Tabla 2. Caso de uso CU-02: Ver mapa.

CU-02	Ver mapa		
Objetivo	Permitir al operador visualizar el mapa principal del sistema.		
Descripción	Este caso de uso permite al operador acceder a la vista principal del sistema, donde se presenta el mapa interactivo junto con los paneles de apoyo para monitoreo en tiempo real y gestión de capas geográficas.		
Referencias cruzadas	• CU-01 Iniciar sesión. • CU-03 Monitorear ubicación en tiempo real. • CU-04 Ver detalle de cuadrilla. • CU-08 Activar/Desactivar capas.		
Precondición	• El operador se encuentra autenticado. • La ruta del mapa está disponible. • El navegador cuenta con conexión suficiente para cargar el panel.		
Secuencia normal	Paso	Acción del operador	Acción
	1	Ingresa al panel principal.	Muestra la vista del mapa.
	2	Observa la pantalla principal.	Presenta el mapa junto con los paneles de tiempo real y capas KMZ.
	3	Desplaza o acerca el mapa.	Actualiza la vista según la interacción realizada.
	4	Revisa los elementos visibles.	Mantiene disponibles marcadores, capas y controles de navegación.
	5	Selecciona una opción del menú lateral.	Cambia a la sección seleccionada o mantiene la vista del mapa.
Postcondición	• El mapa queda visible y disponible para interacción.		
Secuencia alternativa	Paso	Condición	Acción
	A1	El operador no tiene sesión válida.	Redirige a la pantalla de inicio de sesión.
	A2	No existen datos visibles de cuadrillas.	Muestra el mapa sin marcadores actualizados.
	A3	El panel lateral se oculta.	Permite volver a mostrarlo mediante el botón correspondiente.

Fuente: Elaboración propia.

Tabla 3. Caso de uso CU-03: Monitorear ubicación en tiempo real.

CU-03	Monitorear ubicación en tiempo real		
Objetivo	Permitir que el operador observe la ubicación y estado de las cuadrillas en el mapa.		
Descripción	Este caso de uso permite visualizar la ubicación y estado de las cuadrillas sobre el mapa, mostrando la información disponible de acuerdo con las actualizaciones recibidas por el sistema.		
Referencias cruzadas	- CU-02 Ver mapa. - CU-04 Ver detalle de cuadrilla.		
Precondición	- El operador se encuentra autenticado. - El mapa está visible. - Existen datos de ubicación disponibles o una última ubicación registrada.		
Secuencia normal	Paso	Acción del operador	Acción
	1	Ingresa a la vista del mapa.	Muestra el panel de tiempo real.
	2	Observa el estado de conexión.	Indica si el sistema está conectado, reconectando o desconectado.
	3	Revisa las cuadrillas visibles.	Muestra marcadores sobre el mapa según la información disponible.
	4	Selecciona una cuadrilla o marcador.	Despliega información asociada, como estado, ubicación y última actualización.
	5	Selecciona “Ver en mapa” si está disponible.	Centra la vista en la cuadrilla seleccionada.
Postcondición	El mapa muestra la última ubicación conocida de las cuadrillas disponibles.		
Secuencia alternativa	Paso	Condición	Acción
	A1	Se pierde temporalmente la conexión.	Muestra estado “Reconectando” o “Desconectado”.
	A2	No hay actualizaciones recientes.	Mantiene la última ubicación disponible o indica ausencia de señal reciente.
	A3	No existen cuadrillas visibles.	Mantiene el mapa disponible sin marcadores actualizados.

Fuente: Elaboración propia.

Tabla 4. Caso de uso CU-04: Ver detalle de cuadrilla.

CU-04	Ver detalle de cuadrilla		
Objetivo	Permitir que el operador consulte información específica de una cuadrilla.		
Descripción	Este caso de uso permite al operador consultar información específica de una cuadrilla seleccionada, como su estado, integrantes o datos asociados disponibles en la interfaz.		
Referencias cruzadas	• CU-02 Ver mapa. • CU-03 Monitorear ubicación en tiempo real. • CU-05 Filtrar cuadrillas.		
Precondición	• El operador se encuentra autenticado. • Existe información de cuadrillas disponible. • El operador se encuentra en el mapa o en la sección de cuadrillas.		
Secuencia normal	Paso	Acción del operador	Acción
	1	Accede a la sección de cuadrillas o selecciona una cuadrilla visible.	Muestra la información disponible de la cuadrilla.
	2	Selecciona una cuadrilla del listado.	Abre la vista de detalle.
	3	Revisa la información presentada.	Muestra datos como nombre, estado e integrantes asociados.
	4	Selecciona volver.	Retorna al listado de cuadrillas.
Postcondición	• El sistema muestra información detallada de la cuadrilla seleccionada.		
Secuencia alternativa	Paso	Acción del operador	Acción
	A1	La cuadrilla no posee información completa.	Muestra únicamente los datos disponibles.
	A2	El operador cambia de sección.	Cierra o reemplaza la vista de detalle según la navegación seleccionada.
	A3	No existen cuadrillas cargadas.	Muestra la sección sin registros disponibles.

Fuente: Elaboración propia.

Tabla 5. Caso de uso CU-05: Filtrar cuadrillas.

CU-05	Filtrar Cuadrillas		
Objetivo	Facilitar la búsqueda de cuadrillas mediante criterios de texto o estado.		
Descripción	Este caso de uso permite al operador reducir la lista de cuadrillas visibles mediante un campo de búsqueda y filtros de estado. Su finalidad es agilizar la consulta de cuadrillas específicas y facilitar el acceso posterior a su detalle.		
Referencias cruzadas	CU-04 Ver detalle de cuadrilla.		
Precondición	- El operador se encuentra autenticado. - La sección de cuadrillas está disponible. - Existe un listado de cuadrillas para consultar.		
Secuencia normal	Paso	Acción del operador	Acción
	1	Ingresa a la sección de cuadrillas.	Muestra el listado disponible.
	2	Escribe un criterio de búsqueda.	Actualiza la lista según el texto ingresado.
	3	Selecciona un filtro de estado.	Muestra cuadrillas activas, inactivas o todas, según corresponda.
	4	Revisa los resultados.	Mantiene visibles solo las coincidencias.
	5	Selecciona una cuadrilla.	Abre la vista de detalle de la cuadrilla seleccionada.
Postcondición	El listado muestra las cuadrillas que cumplen con el criterio aplicado.		
Secuencia alternativa	Paso	Condición	Acción
	A1	No existen coincidencias.	Muestra la lista sin resultados visibles.
	A2	El operador limpia el filtro.	Vuelve a mostrar el listado completo.
	A3	El operador cambia de vista.	Conserva la navegación general del panel.

Fuente: Elaboración propia.

Tabla 6. Caso de uso CU-06: Gestionar archivos KMZ.

CU-06	Gestionar Archivos KMZ		
Objetivo	Permitir la administración de archivos KMZ/KML usados como capas geográficas del mapa.		
Descripción	Este caso de uso permite al operador administrar los archivos geográficos disponibles en el panel de capas, incluyendo su revisión, carga, activación, desactivación o eliminación según las necesidades de visualización.		
Referencias cruzadas	- CU-07 Cargar archivo KMZ. - CU-08 Activar/Desactivar capas. - CU-09 Eliminar archivos KMZ.		
Precondición	- El operador se encuentra autenticado. - El mapa principal está disponible. - El panel de capas KMZ está visible o puede abrirse.		
Secuencia normal	Paso	Acción del operador	Acción
	1	Abre el panel de capas KMZ.	Muestra la lista de archivos disponibles.
	2	Revisa las capas disponibles.	Presenta nombre, estado de activación y opciones de acción.
	3	Selecciona una acción sobre un archivo.	Permite cargar, activar/desactivar o eliminar archivos según la opción elegida.
	4	Espera el resultado de la operación.	Muestra una notificación de confirmación, advertencia o error.
	5	Continúa utilizando el mapa.	Mantiene el panel y el mapa disponibles.
Postcondición	La lista de archivos geográficos se mantiene disponible y actualizada según las acciones realizadas.		
Secuencia alternativa	Paso	Condiciones	Acción
	A1	No existen archivos cargados.	Muestra el mensaje “No hay archivos aún” o equivalente.
	A2	El panel está oculto.	Permite volver a abrirlo desde el botón de capas.
	A3	Una operación no puede completarse.	Muestra una notificación de error sin bloquear el uso del mapa.

Fuente: Elaboración propia.

Tabla 7. Caso de uso CU-07: Cargar archivo KMZ.

CU-07	Cargar Archivo KMZ		
Objetivo	Incorporar un archivo geográfico al panel para su posterior visualización como capa del mapa.		
Descripción	Este caso de uso permite incorporar un nuevo archivo geográfico en formato KMZ o KML al sistema, para que posteriormente pueda ser listado y utilizado como capa sobre el mapa.		
Referencias cruzadas	- CU-06 Gestionar archivos KMZ. - CU-08 Activar/Desactivar capas.		
Precondición	- El operador se encuentra autenticado. - El panel de capas KMZ está disponible. - El operador dispone de un archivo compatible en formato KMZ o KML.		
Secuencia normal	Paso	Acción del operador	Acción
	1	Selecciona la opción “Agregar KMZ”.	Abre el selector de archivos del navegador.
	2	Selecciona un archivo KMZ o KML.	Acepta la selección del archivo compatible.
	3	Espera la carga.	Muestra el resultado de la operación mediante una notificación.
	4	Revisa el panel de capas.	Actualiza la lista de archivos disponibles.
	5	Decide si desea activar la capa.	Mantiene el archivo disponible para su visualización.
Postcondición	El archivo cargado queda disponible en la lista de capas, siempre que la operación sea exitosa.		
Secuencia alternativa	Paso	Condición	Acción
	A1	El operador cancela la selección del archivo.	Cierra el selector sin modificar la lista.
	A2	El archivo ya existe.	Solicita confirmación para intentar reemplazarlo.
	A3	La carga falla.	Muestra una notificación de error.

Fuente: Elaboración propia.

Tabla 8. Caso de uso CU-08: Activar/Desactivar capas.

CU-08	Activar/Desactivar Capas		
Objetivo	Este caso de uso permite al operador controlar la visualización de capas geográficas sobre el mapa, mostrando u ocultando archivos KMZ/KML según la información que desea analizar.		
Descripción	Permitir que el operador controle qué capas geográficas se visualizan sobre el mapa.		
Referencias cruzadas	- CU-02 Ver mapa. - CU-06 Gestionar archivos KMZ. - CU-07 Cargar archivo KMZ.		
Precondición	- El operador se encuentra autenticado. - El mapa está visible. - Existen archivos KMZ/KML disponibles en el panel de capas.		
Secuencia normal	Paso	Acción del operador	Acción
	1	Abre el panel de capas KMZ.	Muestra la lista de capas disponibles.
	2	Marca una capa para activarla.	Muestra la capa geográfica sobre el mapa.
	3	Revisa la capa desplegada.	Mantiene la capa visible junto con el mapa operativo.
	4	Selecciona la opción “Centrar” si corresponde.	Ajusta la vista del mapa hacia la información geográfica visible.
	5	Desmarca una capa activa.	Retira la capa del mapa.
Postcondición	Las capas seleccionadas quedan visibles en el mapa y las desactivadas son retiradas de la visualización.		
Secuencia alternativa	Paso	Condición	Acción
	A1	La capa no puede mostrarse.	Muestra un mensaje de error y mantiene el mapa disponible.
	A2	No existen capas disponibles.	Informa que no hay archivos cargados.
	A3	El operador oculta el panel.	Mantiene las capas activas visibles en el mapa.

Fuente: Elaboración propia.

Tabla 9. Caso de uso CU-09: Eliminar archivos KMZ.

CU-09	Eliminar Archivos KMZ		
Objetivo	Permitir que el operador elimine archivos geográficos que ya no requiere en el panel.		
Descripción	Este caso de uso permite al operador eliminar archivos geográficos que ya no son necesarios en el panel de capas, manteniendo actualizada la lista de información disponible para la visualización del mapa.		
Referencias cruzadas	- CU-06 Gestionar archivos KMZ. - CU-08 Activar/Desactivar capas.		
Precondición	- El operador se encuentra autenticado. - Existe al menos un archivo KMZ/KML disponible. - El panel de capas está visible.		
Secuencia normal	Paso	Acción del operador	Acción
	1	Presiona la opción eliminar sobre un archivo.	Solicita confirmación de eliminación.
	2	Confirma la operación.	Solicita confirmación de eliminación.
	3	Espera el resultado.	Muestra una notificación de eliminación exitosa.
	4	Revisa el panel de capas.	Actualiza la lista de archivos disponibles.
	5	Continúa utilizando el mapa.	Mantiene disponible la vista principal.
Postcondición	- El archivo eliminado deja de aparecer en la lista de capas. - Si estaba activo, deja de visualizarse en el mapa.		
Secuencia alternativa	Paso	Condición	Acción
	A1	El operador cancela la confirmación.	No realiza cambios en la lista ni en el mapa.
	A2	La eliminación falla.	Restaura el archivo en la lista y muestra una notificación de error.
	A3	El archivo tenía una capa activa.	Retira la capa del mapa al completar la eliminación.

Fuente: Elaboración propia.

3.1.2 Uso de buenas prácticas y patrones de diseño.

Como patrón principal se adoptó el enfoque de arquitectura basada en servicios (Service Layer), ya que permite encapsular la comunicación HTTP y la gestión de datos en clases dedicadas, manteniendo los componentes enfocados en la presentación y coordinación de la interfaz. Esto se observa en el uso de servicios para la gestión de autenticación y sesión, para la carga y administración de archivos KMZ y para la recepción de actualizaciones de ubicación en tiempo real. Este patrón fue seleccionado porque reduce el acoplamiento entre la interfaz y la lógica de integración, facilita el testing (al poder simular servicios) y permite extender funcionalidades sin alterar la estructura general del panel.

Adicionalmente, para la seguridad y control de acceso se empleó el patrón Interceptor + Guard. El interceptor centraliza la incorporación del token JWT en las solicitudes hacia endpoints protegidos, evitando implementaciones repetidas en cada llamado HTTP y disminuyendo el riesgo de errores por omisión. Por su parte, el guard controla la navegación, impidiendo el acceso a vistas internas si no existe una sesión válida, lo que refuerza un flujo consistente de autenticación.

En cuanto al manejo del estado, se optó por un estado local controlado dentro del componente principal, utilizando estructuras como mapas y colecciones para actualizar marcadores, mantener el último estado por cuadrilla y gestionar elementos activos (por ejemplo, capas KMZ seleccionadas). Esta decisión se tomó para mantener el comportamiento predecible del panel, minimizar recargas completas de la interfaz y asegurar que la actualización de datos en tiempo real se refleje de forma inmediata.

Finalmente, se aplicaron prácticas orientadas a la validación de datos entrantes antes de renderizar, control de reconexión automática en el canal de tiempo real, y manejo explícito de errores en operaciones críticas como subida/eliminación de archivos, de modo que la interfaz entregue retroalimentación y mantenga coherencia incluso ante fallos de red o respuestas inesperadas.

3.1.3 Integración con el sistema general.

La integración del front-end de GeoCuadrilla con el sistema general se realiza mediante una arquitectura cliente-servidor, en la cual el backend actúa como componente central de comunicación entre el panel web y la aplicación móvil utilizada por las cuadrillas en terreno. Esta integración se basa principalmente en solicitudes HTTP con intercambio de datos en formato JSON, complementadas por un mecanismo de comunicación en tiempo real.

El panel web, desarrollado en Angular, se integra con el backend a través de solicitudes HTTP para las operaciones de solicitud-respuesta del sistema. Entre estas operaciones se encuentra el proceso de autenticación, en el cual el front-end envía las credenciales del usuario al backend mediante una solicitud de tipo POST. Como respuesta, el backend retorna un objeto JSON que contiene un token de autenticación (JWT), el cual es almacenado por el front-end y adjuntado automáticamente en las solicitudes posteriores para acceder a los recursos protegidos del sistema.

Una vez autenticado el usuario, el panel web obtiene la información necesaria para la operación del sistema mediante solicitudes HTTP de tipo GET. Estas solicitudes permiten acceder a las tablas maestras del sistema, las cuales contienen la estructura de las cuadrillas, la información de los integrantes, sus roles y la organización asociada a los horarios de trabajo. El backend responde entregando esta información en formato JSON, la cual es procesada por el front-end para estructurar y relacionar los datos que serán utilizados en la visualización y supervisión de las cuadrillas.

La aplicación móvil se integra de manera independiente al panel web, enviando exclusivamente información de geolocalización hacia el backend. Estos datos corresponden a las coordenadas de las cuadrillas en terreno y son enviados mediante solicitudes HTTP.

Para la visualización de la ubicación de las cuadrillas en tiempo real, el front-end establece una conexión continua con el backend mediante SignalR. A través de este canal, el backend envía las actualizaciones de coordenadas recibidas desde la aplicación móvil hacia el panel web, permitiendo que la información se refleje de manera inmediata sobre el mapa interactivo.

Adicionalmente, la integración del panel web con el backend permite la gestión de archivos KMZ. Estas operaciones permiten cargar, consultar y eliminar archivos, recibiendo como respuesta mensajes en formato JSON que indican el resultado de cada operación. Los archivos KMZ son utilizados por el front-end como capas de información geográfica para la visualización de la red eléctrica.

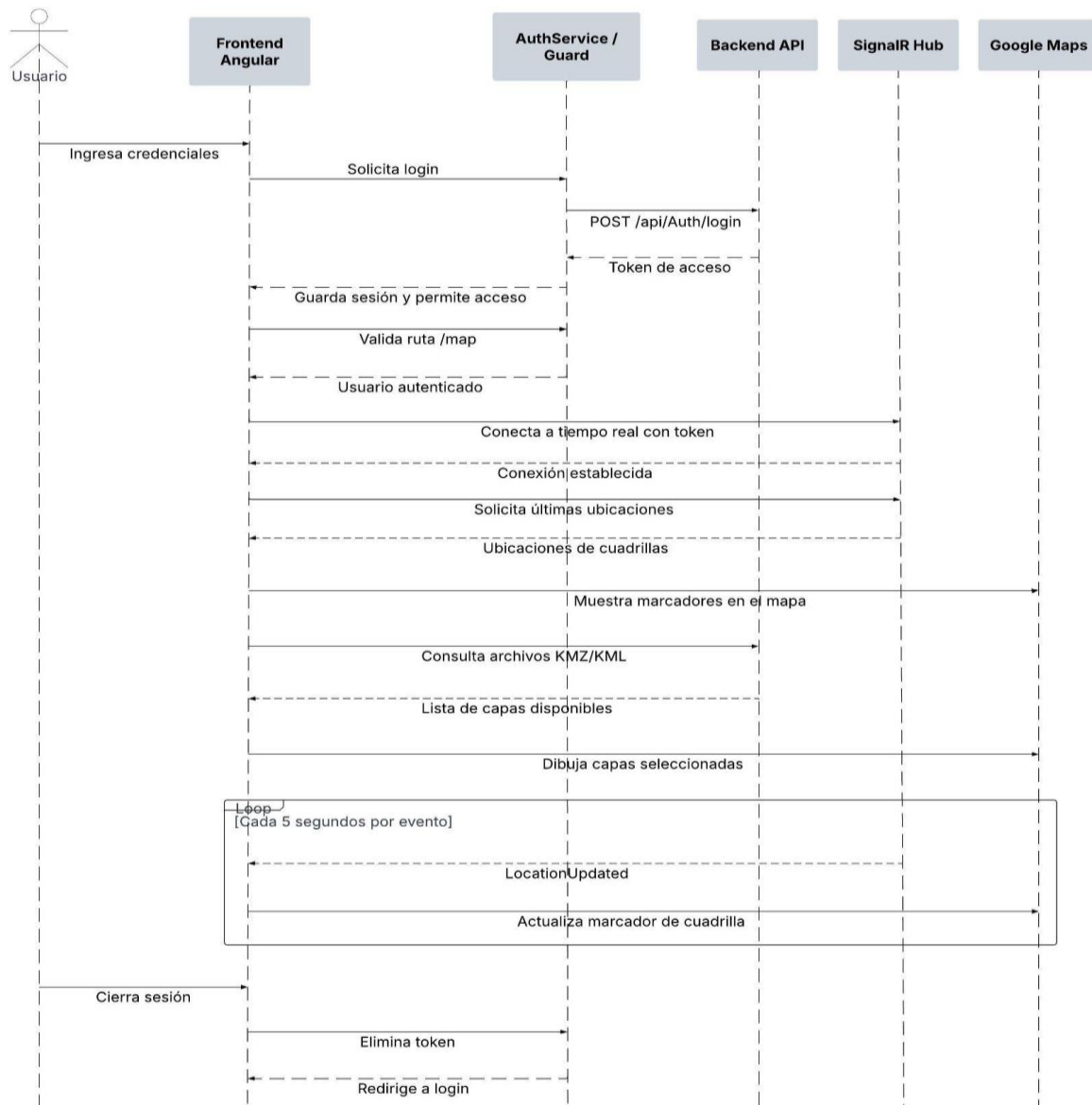


Figura 4: Diagrama de secuencia de la integración del sistema GeoCuadrilla.

Fuente: Elaboración propia.

3.2 Diseño de la Interfaz

3.2.1 Estructura y diseño de la interfaz de usuario (UI)

El acceso al sistema se realiza mediante una pantalla de autenticación simple y centrada, la cual permite el ingreso de credenciales de forma clara y directa, minimizando la presencia de elementos visuales innecesarios.

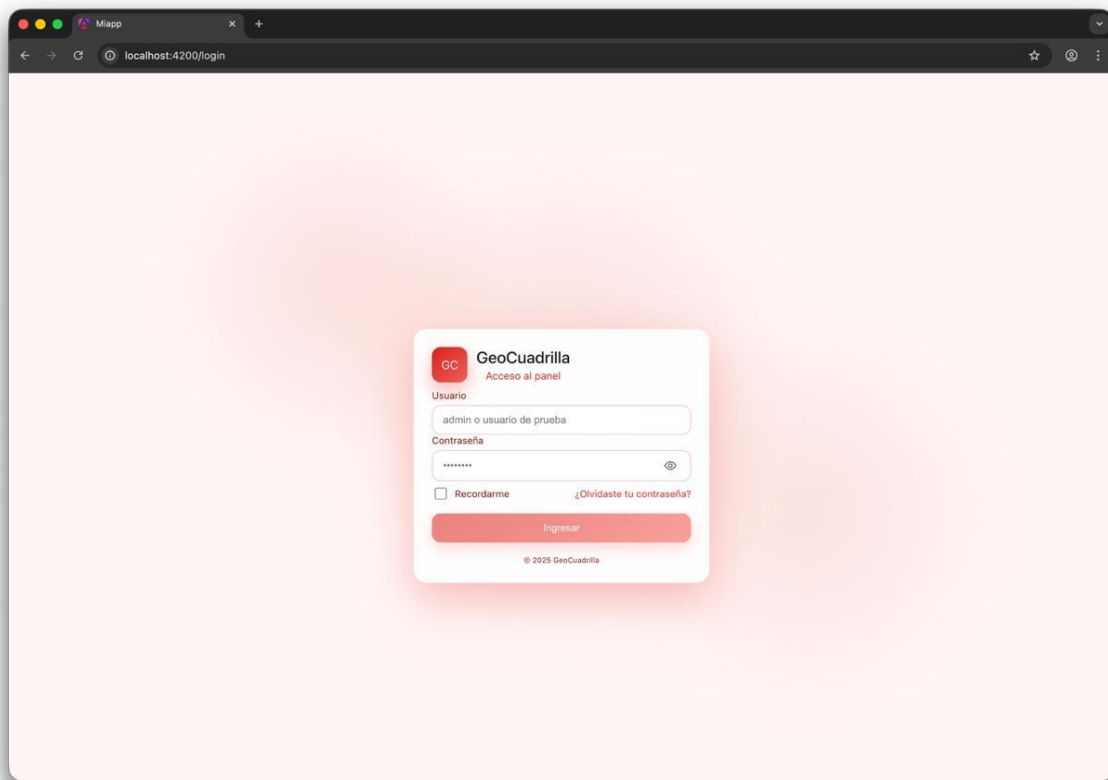


Figura 5: Pantalla de autenticación del panel web de GeoCuadrilla.

Fuente: Elaboración propia.

Una vez autenticado, el usuario accede a la vista principal del sistema, la cual se compone de un mapa interactivo como elemento central, acompañado por paneles laterales superpuestos. El panel lateral izquierdo muestra información en tiempo real sobre el estado de las cuadrillas, mientras que el panel lateral derecho permite la gestión y visualización de capas KMZ asociadas a la red eléctrica.

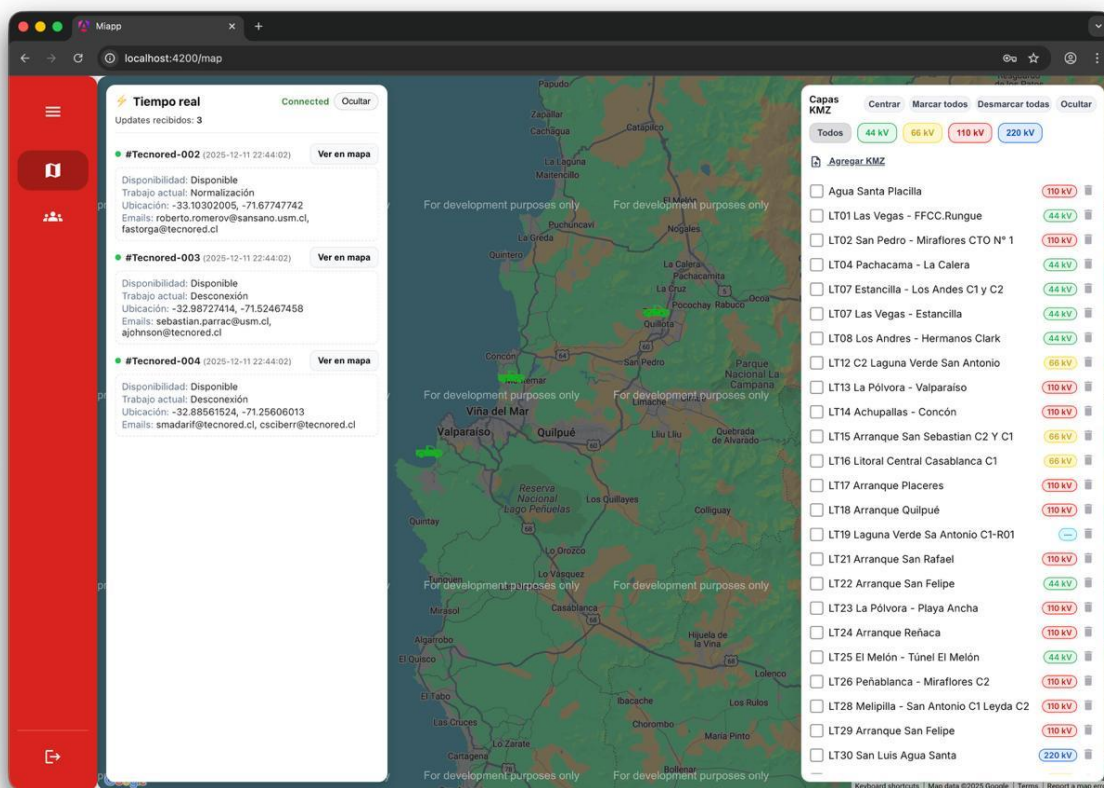


Figura 6: Vista principal del panel web con mapa interactivo y paneles laterales.

Fuente: Elaboración propia.

La figura muestra la visualización de capas KMZ activas sobre el mapa interactivo del panel web. A diferencia de la vista general presentada anteriormente, en esta captura se observa la representación de trazados geográficos asociados a líneas de transmisión eléctrica, diferenciados visualmente mediante colores según su nivel de tensión. Esta funcionalidad permite complementar la ubicación de las cuadrillas con información territorial y eléctrica relevante para el monitoreo operativo.

El panel derecho permite seleccionar, activar o desactivar capas KMZ de forma individual, además de aplicar filtros por nivel de tensión, tales como 44 kV, 66 kV, 110 kV y 220 kV. De esta manera, el operador puede ajustar la información visible en el mapa según las necesidades de análisis, evitando la sobrecarga visual y facilitando la interpretación de la red eléctrica.



Figura 7: Visualización de capas KMZ activas sobre el mapa interactivo.

Fuente: Elaboración propia.

El panel web incorpora elementos informativos superpuestos al mapa que permiten visualizar el estado operativo del sistema de forma inmediata. Cada vehículo representado en el mapa cuenta con una tarjeta informativa asociada a su marcador, la cual se despliega al seleccionar la cuadrilla correspondiente. Esta tarjeta presenta información relevante como el identificador de la cuadrilla, su disponibilidad, la última actualización registrada, el trabajo actual, la ubicación y los datos de contacto del personal técnico, facilitando la consulta rápida del estado de cada unidad en terreno.

Adicionalmente, la interfaz incluye un panel de conexión en tiempo real, visible sobre el mapa, que muestra el estado de la comunicación con el servidor y un resumen de las actualizaciones recibidas. Este componente permite verificar que el sistema se encuentra conectado y operando correctamente, entregando mayor claridad sobre la vigencia de la información visualizada.

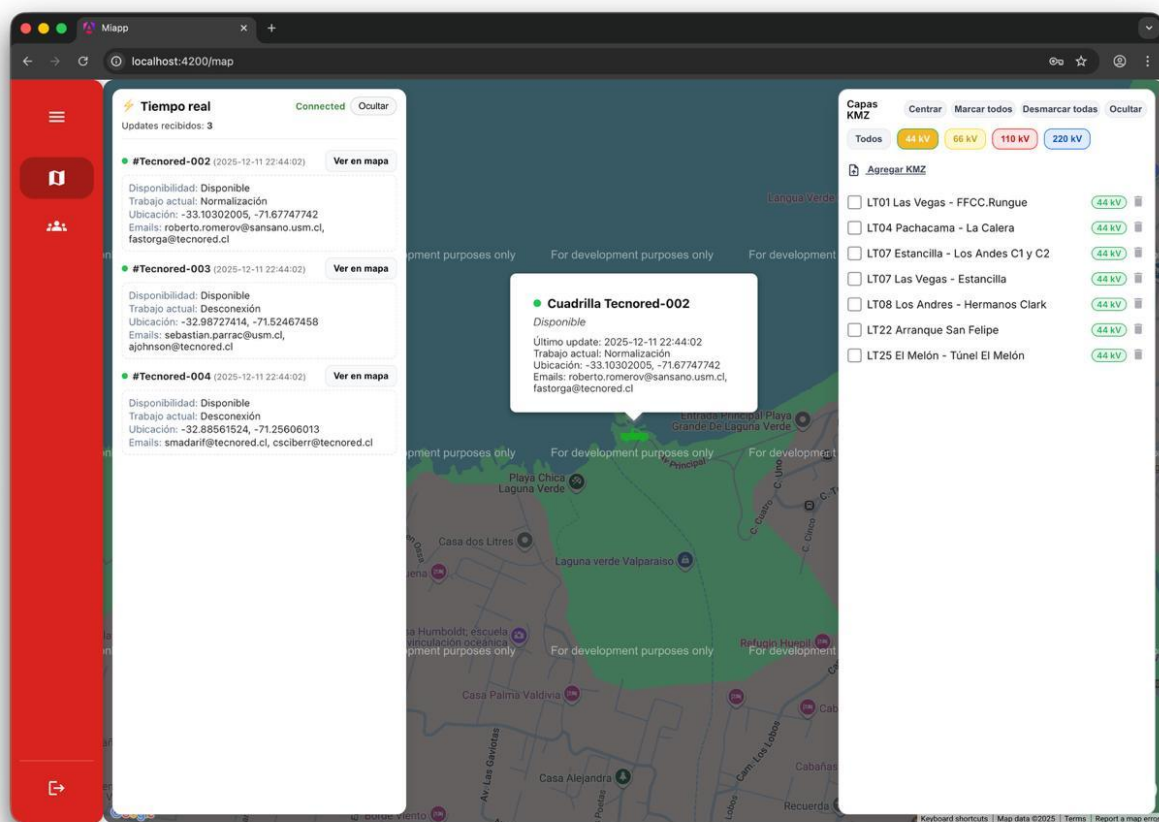


Figura 8: Panel de conexión en tiempo real y estado de cuadrillas.
Fuente: Elaboración propia.

El sistema incluye además vistas específicas para la gestión de cuadrillas, las cuales consisten en un listado general con filtros por estado y una vista de detalle que presenta la información del personal asignado a cada cuadrilla.

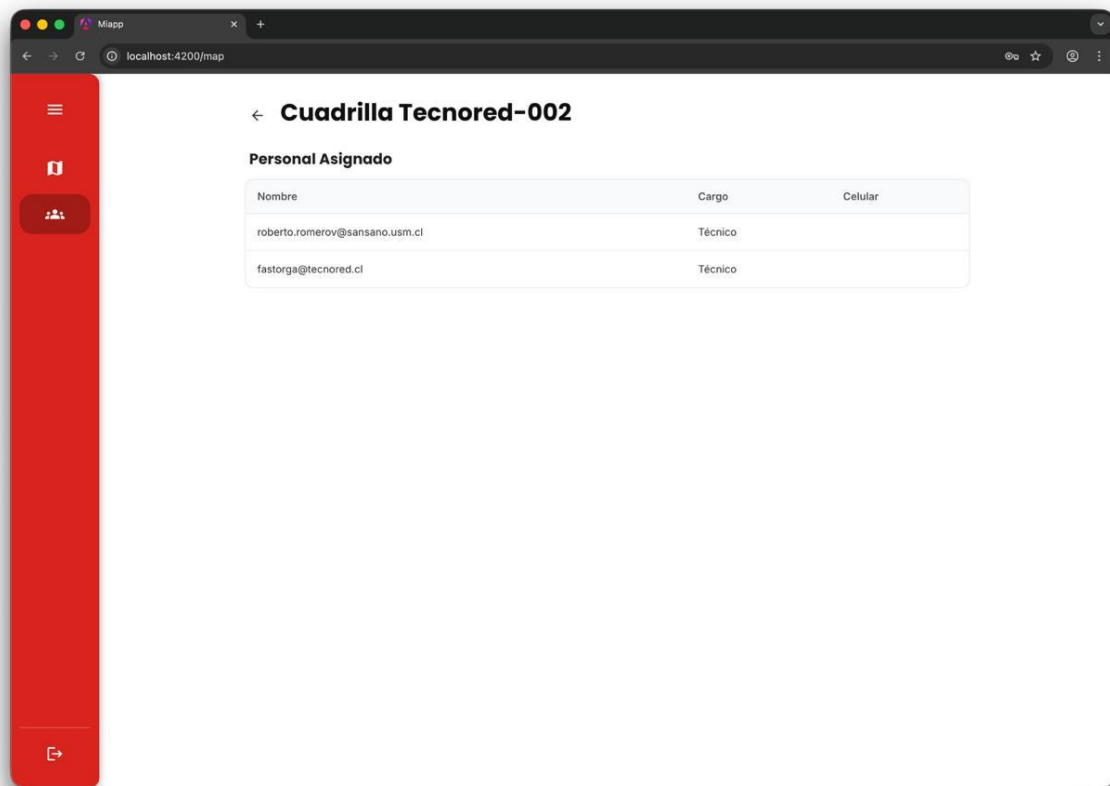


Figura 9: Vista de listado y detalle de cuadrillas.

Fuente: Elaboración propia.

3.2.2 Proceso de creación de wireframes y prototipos

El diseño de la interfaz se desarrolló a partir de prototipos elaborados durante las primeras etapas del proyecto, los cuales permitieron definir la estructura general de la interfaz, la disposición de los elementos y los flujos de navegación principales del sistema. Estos prototipos sirvieron como base para la posterior implementación del panel web, facilitando la validación temprana de las decisiones de diseño. [6]

A medida que avanzó el desarrollo del front-end, la interfaz fue evolucionando y se incorporaron elementos cruciales para el funcionamiento final de la aplicación, los cuales no se encontraban contemplados en las versiones preliminares del diseño.

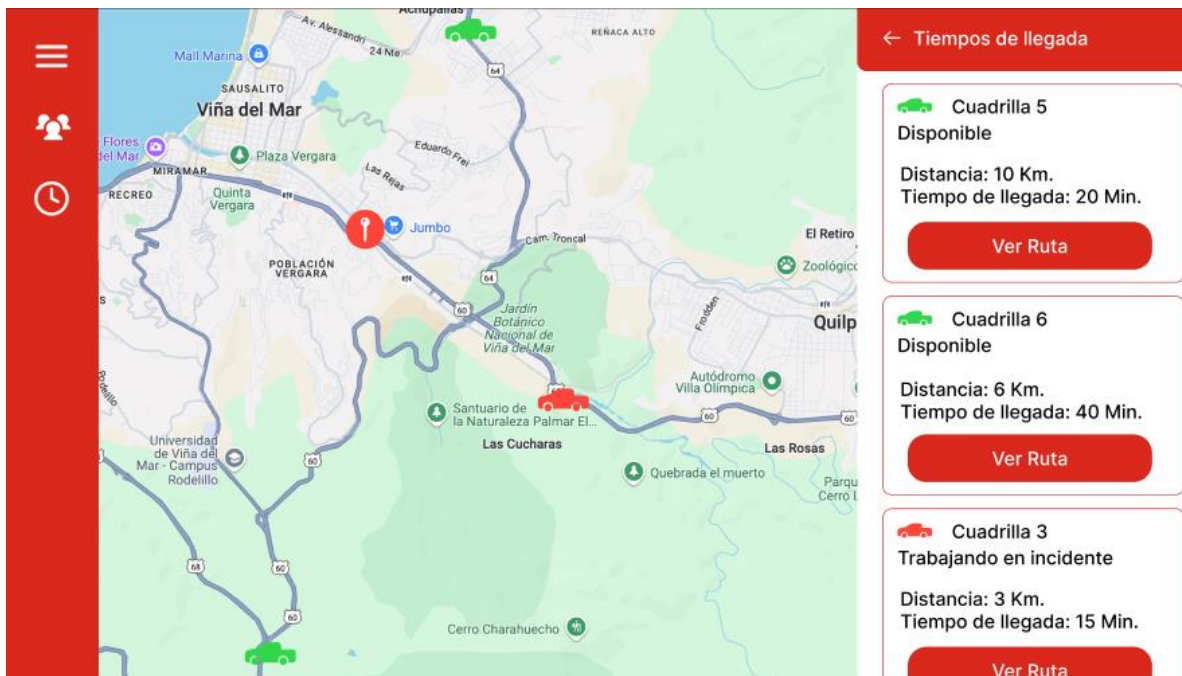


Figura 10: Prototipo en Figma para la estimación de tiempos de llegada durante la etapa de diseño.
Fuente: Elaboración propia.

3.2.3 Paleta de colores

La definición de la paleta cromática del sistema se realizó considerando criterios institucionales y operativos propios del contexto en el que se inserta la solución. La paleta cromática general de la interfaz fue seleccionada en base a los colores corporativos de la empresa, con el propósito de mantener coherencia visual con su identidad institucional y favorecer la familiaridad de los usuarios con el entorno del sistema. El uso de tonos predominantes de rojo y grises permite establecer una jerarquía visual clara entre los distintos componentes de la interfaz, tales como paneles de navegación, contenedores de información y elementos interactivos.

Adicionalmente, los colores de la interfaz cumplen un rol funcional en la representación del estado operativo de las cuadrillas. En la visualización del mapa, los íconos asociados a las camionetas utilizan codificación cromática para indicar su disponibilidad, el color rojo representa una cuadrilla ocupada, mientras que el color verde indica que la cuadrilla se encuentra disponible para la asignación de nuevos trabajos. Esta relación entre color y estado operativo permite una identificación inmediata de la situación de cada cuadrilla, reduciendo el tiempo de interpretación por parte del operador.



Figura 11: Representación de estados operativos de cuadrillas mediante codificación cromática.
Fuente: Elaboración propia a partir de material gráfico referencial.

En complemento a lo anterior, se definió una paleta de colores específica para la visualización de líneas de transmisión, utilizada en la representación de capas KMZ sobre el mapa. Los colores asociados a los distintos niveles de voltaje responden a criterios ampliamente utilizados en el sector eléctrico, lo que permite una identificación rápida y estandarizada de la infraestructura representada. Esta codificación cromática busca facilitar la interpretación de la información espacial y diferenciar visualmente los distintos niveles de tensión durante el monitoreo de la red eléctrica.



Figura 12: Paletas cromáticas utilizadas en el sistema GeoCuadrilla.

Fuente: Elaboración propia.

3.2.4 Responsividad y adaptabilidad de la interfaz

El panel web de GeoCuadrilla fue diseñado considerando escenarios de uso propios de entornos operativos de empresas eléctricas, donde el sistema puede ser visualizado en pantallas de gran formato y alta resolución destinadas al monitoreo continuo. En este sentido, la interfaz prioriza una distribución horizontal que permite aprovechar anchos de pantalla extendidos, favoreciendo la visualización simultánea del mapa y de los paneles informativos.

Durante el desarrollo y validación del sistema, la interfaz fue probada en distintos equipos portátiles, donde se verificó su correcto funcionamiento y visualización en distintas resoluciones de pantalla. La distribución flexible de los elementos permite que el panel web mantenga su funcionalidad y legibilidad en este tipo de dispositivos, asegurando una experiencia de uso consistente dentro de los entornos evaluados.

3.3 Detalles de la codificación y desarrollo.

La implementación del front-end se centró en materializar las funcionalidades definidas para el panel web, priorizando la correcta comunicación con el backend y la correcta representación de la información operativa del sistema. Para ello, se desarrollaron los mecanismos necesarios que permiten obtener, procesar y visualizar los datos requeridos para la supervisión de las cuadrillas en terreno.

El desarrollo del panel se estructuró en torno a un módulo principal que articula la visualización del mapa, los paneles informativos y los elementos de interacción del usuario, permitiendo integrar de manera coherente las distintas funcionalidades implementadas.

3.3.1 Configuración del entorno de ejecución

Para gestionar la comunicación entre el front-end y los servicios externos, la aplicación define archivos de entorno que permiten separar la configuración de desarrollo y producción. Esta estrategia evita modificar directamente la lógica de los componentes cuando cambian las rutas del backend, del servicio de autenticación o del canal de comunicación en tiempo real.

```
export const environment = {  
  production: false,  
  apiBase: '/api',  
  authEndpoint: '/api/Auth/login',  
  signalRHubUrl: '/hubs/location',  
};
```

Figura 13: Configuración del entorno de desarrollo.

Fuente: Elaboración propia a partir del código fuente del proyecto GeoCuadrilla.

Como se observa, en desarrollo se utilizan rutas relativas, lo que facilita el trabajo con un proxy local. En producción, en cambio, se definen rutas absolutas hacia el servidor correspondiente. Esta separación favorece la mantenibilidad del proyecto y permite adaptar la aplicación a distintos ambientes de ejecución sin alterar su estructura funcional.

3.3.2 Autenticación y manejo de sesión

El sistema implementa un mecanismo de autenticación basado en tokens JWT, gestionado mediante un servicio dedicado. Este servicio es responsable de enviar las credenciales al backend, procesar la respuesta y almacenar de forma persistente la información necesaria para mantener la sesión del usuario.

Cuando la autenticación es exitosa, el front-end obtiene el token de acceso, token de refresco, fecha de expiración e información del usuario. Estos datos se almacenan en el `localStorage`, lo que permite conservar la sesión activa incluso ante recargas del navegador.

```
const token = b.data.accessToken;  
const refresh = b.data.refreshToken ?? null;  
const expIso = b.data.expiresAt ?? null;  
const user = b.data.user ?? null;  
  
localStorage.setItem(this.tokenKey, token);  
if (refresh) localStorage.setItem(this.refreshKey, refresh);  
if (expIso) localStorage.setItem(this.expKey, expIso);  
if (user) localStorage.setItem(this.userKey, JSON.stringify(user));
```

Figura 14: Almacenamiento de token y datos de sesión.

Fuente: Elaboración propia a partir del código fuente del proyecto GeoCuadrilla.

Además del almacenamiento de credenciales, el servicio incorpora una validación de sesión que revisa la existencia del token y su fecha de expiración. En primer lugar, se considera la fecha entregada por el backend; si esta no se encuentra disponible, se utiliza como alternativa la información contenida en el propio JWT. Con ello, el front-end puede determinar si la sesión sigue siendo válida antes de permitir el acceso a las vistas internas de la aplicación.

```
isAuthenticated(): boolean {
  const token = this.getToken();
  if (!token) return false;

  const expIso = localStorage.getItem(this.expKey);
  if (expIso) {
    const expMs = Date.parse(expIso);
    return isFinite(expMs) ? Date.now() < expMs : true;
  }

  try {
    const payload = JSON.parse(atob(token.split('.')[1]));
    if (payload?.exp) {
      return Date.now() < payload.exp * 1000;
    }
  } catch { /* noop */ }
  return true;
}
```

Figura 15: Validación de sesión activa en el front-end.

Fuente: Elaboración propia a partir del código fuente del proyecto GeoCuadrilla.

Este enfoque permitió centralizar la lógica de autenticación en un único punto, simplificando su reutilización en otros módulos del sistema y facilitando la protección de rutas y solicitudes HTTP.

3.3.3. Protección de rutas y control de acceso

Para garantizar que únicamente usuarios autenticados puedan acceder al panel web, se implementó un guard de rutas. Este mecanismo valida el estado de autenticación antes de permitir la navegación hacia las vistas protegidas, redirigiendo automáticamente al usuario a la pantalla de inicio de sesión en caso contrario.

```
export const authGuard: CanActivateFn = () => {
  const auth = inject(AuthService);
  const router = inject(Router);
  if (auth.isAuthenticated()) return true;
  router.navigateByUrl('/login');
  return false;
};
```

Figura 16: Guard de autenticación para protección de rutas.

Fuente: Elaboración propia a partir del código fuente del proyecto GeoCuadrilla.

Este control asegura que el acceso al sistema esté restringido desde la navegación del front-end. Si bien la validación definitiva debe mantenerse en el backend, el guard contribuye a evitar el acceso directo a vistas internas mediante la URL y permite mantener un flujo de navegación coherente para el usuario.

3.3.4 Interceptor HTTP para autorización

Con el objetivo de evitar la repetición de lógica en cada solicitud HTTP, se desarrolló un interceptor que adjunta automáticamente el encabezado de autorización a todas las peticiones enviadas al backend. De esta forma, el token JWT es incluido de manera transparente, sin intervención directa de los componentes.

```
export const authInterceptor: HttpInterceptorFn = (req, next) => {  
  const token = inject(AuthService).getToken();  
  
  if (token) {  
    req = req.clone({ setHeaders: { Authorization: `Bearer ${token}` } });  
  }  
  return next(req);  
};
```

Figura 17: Interceptor HTTP para incorporación del token JWT.

Fuente: Elaboración propia a partir del código fuente del proyecto GeoCuadrilla.

Esta estrategia mejora la consistencia de las solicitudes y reduce la posibilidad de errores por omisión del token.

3.3.5 Comunicación en tiempo real mediante SignalR

Uno de los elementos centrales del sistema es la visualización en tiempo real de la ubicación de las cuadrillas. Para ello, se integró SignalR como mecanismo de comunicación entre el backend y el front-end, utilizando WebSockets como transporte principal. El servicio de ubicación establece una conexión persistente con el servidor y escucha el evento `LocationUpdated`, mediante el cual recibe actualizaciones de posición, estado de disponibilidad, identificador de cuadrilla, correo del técnico y fecha de actualización.

La conexión se configura con una política de reconexión automática, permitiendo que el sistema intente restablecer el canal de comunicación ante interrupciones temporales. Esta característica es relevante para escenarios operacionales donde la conectividad puede variar durante el monitoreo.

```
this.connection = new signalR.HubConnectionBuilder()  
    .withUrl(hubUrl, {  
        accessTokenFactory: tokenFactory,  
        withCredentials: false,  
        skipNegotiation: true,  
        transport: signalR.HttpTransportType.WebSockets  
    } as signalR.IHttpConnectionOptions)  
    .withAutomaticReconnect([0, 2000, 10000, 30000])  
    .configureLogging(signalR.LogLevel.Information)  
    .build();
```

Figura 18: Configuración de conexión en tiempo real mediante SignalR.

Fuente: Elaboración propia a partir del código fuente del proyecto GeoCuadrilla.

Cada actualización recibida es transformada a una estructura interna antes de ser enviada al componente principal. Durante este proceso se validan datos mínimos, como la existencia del identificador de cuadrilla y la presencia de coordenadas válidas. Esta validación evita que registros incompletos o malformados sean representados en el mapa, contribuyendo a mantener la coherencia visual de la información mostrada al usuario.

Además, la conexión en tiempo real incorpora reconexión automática mediante una política de reintentos progresivos. En caso de una interrupción temporal, el sistema cambia su estado a reconectando e intenta restablecer la comunicación en intervalos definidos de 0, 2, 10 y 30 segundos. También se configuran parámetros de tiempo de espera y mantenimiento de conexión para disminuir desconexiones inesperadas o detecciones erróneas de pérdida de comunicación. La interfaz consume este estado y lo muestra al usuario, permitiendo distinguir si el sistema se encuentra conectado, conectando, reconectando o desconectado.

```
this.connection.onreconnecting(err => {  
    this.connectionState$.next(ConnectionState.Reconnecting);  
    console.warn('[SignalR] Reconnecting...', err?.message || err);  
});  
this.connection.onreconnected(id => {  
    this.connectionState$.next(ConnectionState.Connected);  
    console.log('[SignalR] Reconnected. ConnectionId:', id);  
    this.logTransport();  
});  
this.connection.onclose(err => {  
    this.connectionState$.next(ConnectionState.Disconnected);  
    console.error('[SignalR] Closed', err?.message || err);  
});
```

Figura 19: Manejo de estados de conexión en tiempo real.

Fuente: Elaboración propia a partir del código fuente del proyecto GeoCuadrilla.

Por lo tanto, ante una pérdida temporal de conectividad, la aplicación no requiere una recarga manual inmediata, ya que intenta recuperar automáticamente el canal de comunicación. Sin embargo, si la desconexión se mantiene por un periodo prolongado, la información visualizada puede corresponder al último dato recibido, por lo que la vigencia de la ubicación depende del restablecimiento de la conexión con el servidor.

Si bien el sistema incorpora reconexión automática ante interrupciones temporales de red, la versión desarrollada no contempla un modo offline completo ni sincronización diferida de información. La aplicación depende de la conectividad con el backend para autenticar usuarios, recibir actualizaciones de ubicación en tiempo real y gestionar archivos KMZ. En consecuencia, ante una pérdida prolongada de conexión, el panel puede conservar temporalmente la última información visualizada, pero no garantiza la actualización de ubicaciones ni el envío posterior de datos pendientes.

El almacenamiento local utilizado por la aplicación se limita principalmente a la persistencia de credenciales de sesión, como token de acceso, token de refresco, fecha de expiración e información básica del usuario. Por lo tanto, no se emplea como mecanismo de almacenamiento offline de coordenadas y archivos. La incorporación de un modo offline requeriría diseñar una estrategia adicional de almacenamiento local, control de conflictos y sincronización diferida con el backend una vez recuperada la conexión.

3.3.6 Gestión de archivos KMZ

La gestión de archivos KMZ se implementa mediante un servicio especializado que interactúa con el backend a través de solicitudes HTTP. Este servicio permite listar, subir, descargar y eliminar archivos geográficos, los cuales son utilizados por el front-end como capas de información sobre el mapa. La carga se realiza mediante FormData, lo que permite transmitir archivos binarios junto con información adicional asociada.

```
upload(file: File, description?: string): Observable<HttpResponse<{ isSuccess: boolean; data: GcFileItem }>> {
  const fd = new FormData();
  fd.append('file', file, file.name);
  if (description) fd.append('description', description);

  console.log('[UPLOAD] file.name =', file.name);
  console.log('[UPLOAD] file.type =', file.type);
  console.log('[UPLOAD] file.size =', file.size);
  fd.forEach((v, k) => console.log('[UPLOAD] form field:', k, v));

  return this.http.post<{ isSuccess: boolean; data: GcFileItem }>(
    `${this.base}/files/upload`,
    fd,
    { observe: 'response' }
  );
}
```

Figura 20: Carga de archivos KMZ/KML mediante FormData.

Fuente: Elaboración propia a partir del código fuente del proyecto GeoCuadrilla.

Desde la interfaz, la selección de archivos se restringe a formatos .kmz y .kml, de acuerdo con el tipo de información geográfica requerida por el sistema.

```
<input #fileChooser type="file" accept=".kmz,.kml" style="display:none"
  (change)="uploadFromForm(fileChooser.files?.[0] || null); fileChooser.value='';" />
```

Figura 21: Restricción de selección de archivos KMZ/KML desde la interfaz.

Fuente: Elaboración propia a partir del código fuente del proyecto GeoCuadrilla.

Una vez obtenido el archivo, el componente principal identifica si corresponde a un KMZ mediante la revisión de su cabecera binaria. En caso afirmativo, se descomprime el contenido utilizando una librería especializada y se busca el archivo KML interno. Posteriormente, el contenido KML es transformado a GeoJSON, formato que permite renderizar las geometrías sobre Google Maps.

```
private async kmlOrKمزBlobToGeoJson(blob: Blob, filenameHint = ''): Promise<{ gj: FeatureCollection }> {  
  let kmlText: string;  
  const isKمز = await this.blobLooksLikeKمز(blob) ||  
    (blob.type || '').toLowerCase().includes('kمز') ||  
    filenameHint.toLowerCase().endsWith('.kمز');  
  
  if (isKمز) {  
    const zip = await JSZip.loadAsync(blob);  
    const kmlEntry = Object.values(zip.files).find(f => f.name.toLowerCase().endsWith('.kml'));  
    if (!kmlEntry) throw new Error('KMZ sin KML interno');  
    // @ts-ignore  
    kmlText = await kmlEntry.async('text');  
  } else {  
    kmlText = await blob.text();  
  }  
  
  const dom = new DOMParser().parseFromString(kmlText, 'text/xml');  
  const gj = toGeoJSON.kml(dom) as FeatureCollection;  
  return { gj };  
}
```

Figura 22: Procesamiento de archivos KMZ/KML para visualización geográfica.

Fuente: Elaboración propia a partir del código fuente del proyecto GeoCuadrilla.

Esta implementación permite representar líneas de transmisión y puntos geográficos como capas activables dentro del mapa. Además, el sistema incorpora filtros por nivel de tensión, permitiendo diferenciar visualmente capas asociadas a 44 kV, 66 kV, 110 kV y 220 kV. La aplicación también adapta el grosor de líneas y el tamaño de puntos según el nivel de zoom, con el propósito de mejorar la legibilidad y evitar sobrecarga visual.

En cuanto al manejo de errores, las operaciones de carga, reemplazo y eliminación entregan retroalimentación al usuario mediante notificaciones. En el caso de eliminación, la interfaz actualiza temporalmente la lista de archivos y conserva el estado previo; si la operación falla, se restaura la información anterior para evitar inconsistencias entre la interfaz y el backend.

```
const before = this.apiFiles.slice();  
  
this.apiFiles = this.apiFiles.filter(f => {  
  const sameId = String(f.id) === id;  
  const sameNm = this.sameName(f.fileName, name);  
  return !(sameId || sameNm);  
});  
  
try {  
  await firstValueFrom(this.filesService.delete(id));  
  this.notify(`${name} fue eliminado.`, 'success');  
} catch (e) {  
  this.apiFiles = before;  
  this.notify(`No se pudo eliminar "${name}".`, 'error');  
}
```

Figura 23: Manejo de errores y restauración de estado en eliminación de archivos KMZ.

Fuente: Elaboración propia a partir del código fuente del proyecto GeoCuadrilla.

No obstante, la validación actual se concentra principalmente en el formato permitido y en la capacidad de procesamiento del archivo. Validaciones más estrictas, como tamaño máximo permitido, análisis de contenido malicioso o detección de archivos corruptos deben reforzarse desde el backend antes de una implementación productiva. Por esta razón, estas medidas se consideran una mejora futura del sistema.

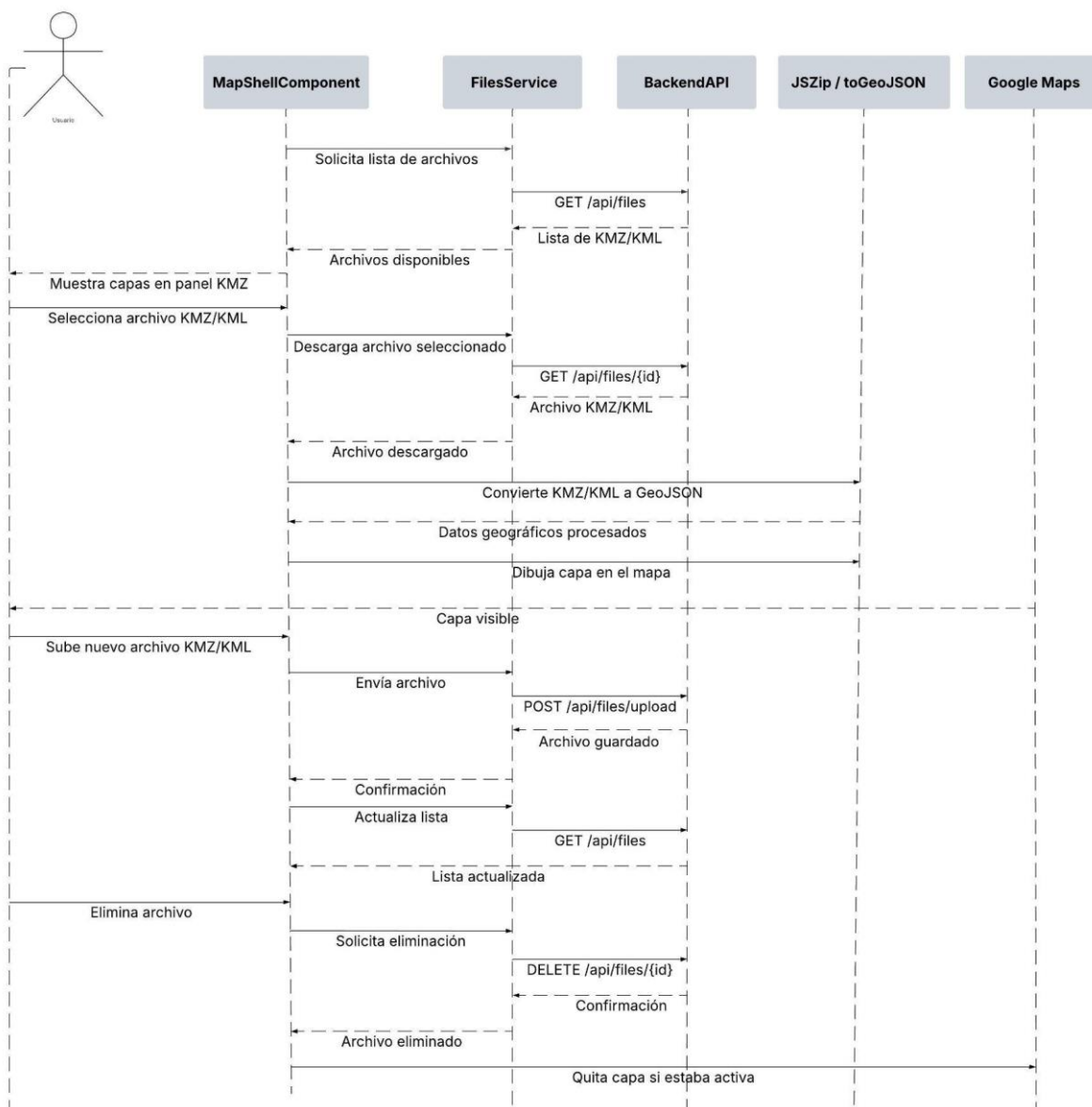


Figura 24: Diagrama de secuencia para la gestión de archivos KMZ en GeoCuadrilla.

Fuente: Elaboración propia.

3.3.7 Componente principal de visualización

El componente principal de visualización concentra la interacción entre los servicios del sistema y la interfaz del usuario. Este componente integra Google Maps, los marcadores de cuadrillas, las capas KMZ/KML y los paneles informativos asociados al estado operacional. Su función es coordinar la información proveniente del servicio de autenticación, del servicio de archivos y del servicio de ubicación en tiempo real, transformándola en elementos visuales comprensibles para el usuario.

```
constructor(  
  private authService: AuthService,  
  private locationService: LocationService,  
  private router: Router,  
  private filesService: FilesService,  
) {}
```

Figura 25: Inyección de servicios en el componente principal.

Fuente: Elaboración propia a partir del código fuente del proyecto GeoCuadrilla.

Al inicializarse, el componente verifica la existencia de una sesión activa y se suscribe a los cambios provenientes del servicio de ubicación. De esta manera, el panel puede reflejar el estado de conexión y las actualizaciones recibidas sin requerir recargas manuales de la aplicación.

```
async ngOnInit(): Promise<void> {  
  const token = this.authService.getToken();  
  if (!token) {  
    this.router.navigateByUrl('/login');  
    return;  
  }  
  
  const stSub = this.locationService.connectionState?.subscribe?.((s: any) => {  
    this.connState = String(s) as any;  
  });  
  if (stSub) this.subs.add(stSub);  
  
  const upSub = this.locationService.locationUpdates?.subscribe?.((u: RealtimeLocationUpdate) => {  
    this.updateCount++;  
    this.lastUpdates.unshift(u);  
    if (this.lastUpdates.length > 10) this.lastUpdates.pop();  
    this.applyUpdate(u);  
  });  
  if (upSub) this.subs.add(upSub);  
  
  await this.locationService.connect(token);  
}
```

Figura 26: Inicialización y suscripción a actualizaciones en tiempo real.

Fuente: Elaboración propia a partir del código fuente del proyecto GeoCuadrilla.

A medida que se reciben actualizaciones de ubicación, el componente mantiene el último estado conocido de cada cuadrilla y actualiza su marcador correspondiente en el mapa. Además, conserva información complementaria, como disponibilidad, correos asociados y tipo de actividad cuando estos datos se encuentran disponibles. Esto permite que la interfaz entregue una representación actualizada del estado operativo sin necesidad de recargar la aplicación.

```
if (!marker) {  
  const icon = this.iconForStatus(status);  
  marker = new google.maps.Marker({  
    position: pos,  
    map: this.map,  
    icon,  
    zIndex: 99999  
  });  
  
  this.markers.push(marker);  
  this.markersById.set(crewId, marker);  
} else {  
  marker.setPosition(pos);  
  marker.setIcon(this.iconForStatus(status));  
}
```

Figura 27: Creación y actualización de marcadores de cuadrillas.

Fuente: Elaboración propia a partir del código fuente del proyecto GeoCuadrilla.

Respecto de las capas geográficas, el componente permite activar, desactivar, centrar y limpiar archivos KMZ/KML cargados desde el backend. Cada capa se administra de forma independiente, lo que facilita su visualización selectiva según el nivel de tensión o la necesidad del usuario. Esta organización contribuye a que el mapa no solo muestre la ubicación de las cuadrillas, sino también el contexto territorial y eléctrico en el que se desarrollan las actividades.

```
async onToggleFile(item: GcFileItem, checked: boolean): Promise<void> {  
  const myGen = this.kmzGeneration;  
  const id = String(item.id);  
  
  if (checked) {  
    await this.ensureLayerForFile(item);  
    if (myGen !== this.kmzGeneration) return;  
  
    const entry = this.layersById.get(id)!;  
    entry.layer.setMap(this.map);  
    entry.pointMarkers.forEach(m => m.setMap(this.map));  
    this.activeFileIds.add(id);  
  
    this.applyZoomStyles();  
  
    if (entry.bounds && !entry.bounds.isEmpty()) {  
      try { this.map.fitBounds(entry.bounds, 60); } catch {}  
    }  
  } else {  
    this.detachLayerById(id);  
    this.activeFileIds.delete(id);  
  }  
}
```

Figura 28: Activación y desactivación de capas KMZ/KML.

Fuente: Elaboración propia a partir del código fuente del proyecto GeoCuadrilla.

En conjunto, el componente principal actúa como articulador de la experiencia del panel web, integrando datos operacionales, información geoespacial y estados de conexión en una única vista. Esta centralización permite que el usuario acceda de forma rápida a la información relevante para el monitoreo de cuadrillas y la interpretación del entorno eléctrico.

3.4 Pruebas y Validación

3.4.1 Estrategia de testing aplicada a los componentes

La revisión funcional del front-end de GeoCuadrilla se orientó a observar el comportamiento de los principales requerimientos definidos para el panel web, considerando las restricciones propias del contexto del proyecto. Dado que no se contó con acceso directo al entorno operativo completo de la empresa, las pruebas fueron realizadas en un ambiente controlado, utilizando equipos personales, cuentas de prueba y datos proporcionados durante el desarrollo.

La estrategia de validación contempló principalmente pruebas funcionales y de integración, enfocadas en revisar el comportamiento de los flujos principales del sistema, así como la interacción entre el front-end, el backend y la aplicación móvil.

Tabla 10. Resumen de pruebas funcionales y de integración aplicadas al panel web.

Funcionalidad evaluada	Tipo de prueba	Escenario de prueba	Resultado observado
Inicio de sesión	Funcional / Integración	Acceso al panel mediante credenciales válidas y respuesta ante credenciales inválidas	Se observó el acceso correcto al panel y la visualización de mensajes ante errores de autenticación
Protección de rutas	Funcional	Redirección del usuario hacia el login cuando no existe una sesión activa.	Se observó la redirección hacia la pantalla de inicio de sesión.
Visualización del mapa	Funcional	Carga del mapa como vista principal del panel web.	Se observó la carga del mapa junto con los paneles laterales.
Integración con aplicación móvil	Integración	Revisión del flujo de ubicación desde la aplicación móvil hacia el backend y su visualización posterior en el panel web.	Se observó que las actualizaciones disponibles podían reflejarse en el panel web dentro del ambiente de prueba.

Visualización de cuadrillas en el mapa	Integración	Representación de ubicaciones disponibles mediante marcadores en el mapa.	Se observó la actualización de marcadores según la información recibida.
Panel de conexión en tiempo real	Funcional	Visualización del estado de conexión con el servicio en tiempo real.	Se observó la visualización de estados como conectado, reconectando o desconectado.
Carga de archivos KMZ/KML	Funcional / Integración	Selección y carga de archivos geográficos compatibles desde la interfaz.	Se observó la incorporación de archivos a la lista de capas.
Activación y desactivación de capas	Funcional	Activación y desactivación de capas geográficas sobre el mapa.	Se observó la representación de capas activas y su retiro al desactivarlas.
Filtro por nivel de tensión	Funcional	Filtrado de capas según el nivel de tensión seleccionado	Se observó la actualización de la lista de capas según el filtro aplicado.
Eliminación de archivos KMZ/KML	Funcional / integración	Confirmación, eliminación del archivo y retroalimentación al usuario.	Se observó la eliminación desde la lista y la visualización de notificaciones.
Responsividad de la interfaz	Visual / cualitativa	Visualización y legibilidad de la interfaz en distintos tamaños de pantalla.	Se observó una visualización adecuada en equipos portátiles y resoluciones evaluadas.

Fuente: Elaboración propia.

La tabla anterior sintetiza los escenarios considerados durante la revisión funcional e integrada del panel web, organizando las funcionalidades evaluadas, el tipo de prueba aplicado y los resultados observados en el ambiente de prueba disponible.

En primer lugar, la integración con el backend se evaluó mediante solicitudes HTTP. Este procedimiento permitió validar procesos clave como la autenticación de usuarios, la obtención de la información estructural del sistema, incluyendo cuadrillas, integrantes y estados, y la correcta recepción de respuestas en formato JSON. La revisión de dichos flujos se efectuó analizando el comportamiento de la interfaz tanto ante respuestas válidas e inválidas como frente a la expiración de sesiones.

Adicionalmente, se verificó el mecanismo de comunicación en tiempo real, confirmando que las coordenadas enviadas desde la aplicación móvil fuesen recibidas por el backend y distribuidas correctamente hacia el panel web. Con ello se constató que las actualizaciones de ubicación y estado de las cuadrillas se reflejan de forma inmediata en el mapa y en los paneles informativos, sin requerir la recarga de la aplicación.

Por último, en relación con la gestión de archivos KMZ, se desarrollaron pruebas funcionales orientadas a la carga, eliminación y visualización de capas sobre el mapa. Asimismo, se comprobó el funcionamiento de los filtros asociados a los distintos niveles de voltaje y la aplicación de estilos visuales coherentes con cada tipo de línea de transmisión, verificando que el panel responde adecuadamente a las interacciones del usuario y mantiene la consistencia visual esperada.

3.4.2 Pruebas Funcionales

Las pruebas funcionales se enfocaron en verificar que cada funcionalidad implementada en el panel web cumpliera con el comportamiento esperado desde la perspectiva del usuario. En esta categoría se evaluaron acciones como el inicio de sesión, la protección de rutas internas, la visualización del mapa, la carga de archivos KMZ/KML, la activación y desactivación de capas geográficas, el filtrado por nivel de tensión, la visualización de cuadrillas y el cierre de sesión.

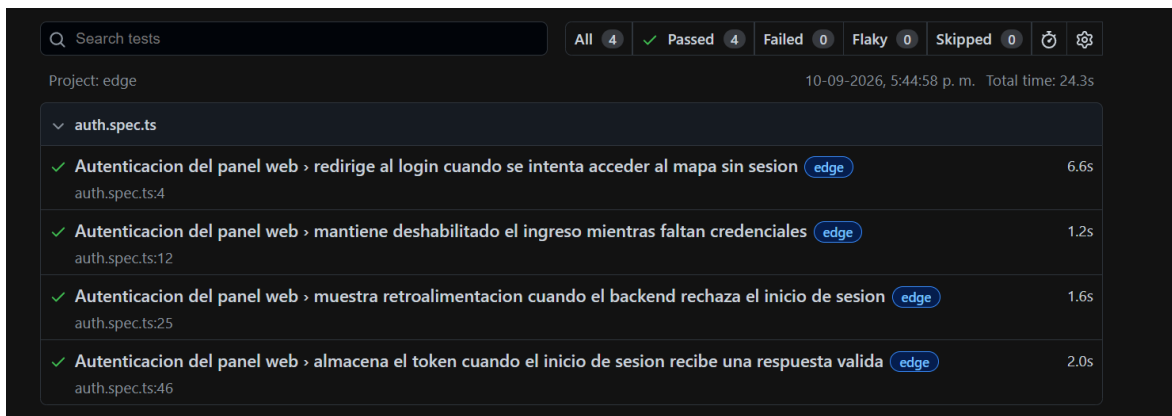
Los criterios de aceptación considerados para estas pruebas fueron los siguientes: el usuario debe poder iniciar sesión con credenciales válidas; el sistema debe impedir el acceso al panel sin autenticación; el mapa debe visualizarse correctamente; los archivos KMZ/KML deben cargarse y representarse como capas geográficas; los filtros de voltaje deben modificar la visualización según la selección del usuario; y las operaciones de carga o eliminación deben entregar retroalimentación visual mediante mensajes.

A partir de las pruebas realizadas, se verificó que los flujos principales del front-end se comportaron de acuerdo con los requerimientos definidos dentro del ambiente de prueba disponible. Las incidencias detectadas durante el proceso correspondieron principalmente a ajustes visuales, validaciones de interacción y manejo de respuestas del backend, las cuales fueron corregidas durante las iteraciones de desarrollo.

3.4.3 Pruebas funcionales automatizadas

Como complemento a la validación funcional del panel web, se incorporaron pruebas automatizadas mediante Playwright sobre flujos críticos de acceso al sistema. Estas pruebas permitieron ejecutar escenarios desde la perspectiva del usuario, simulando interacciones reales con la interfaz y verificando la respuesta del sistema ante distintas condiciones de autenticación.

Los flujos automatizados consideraron el intento de acceso a una ruta protegida sin una sesión activa, la validación del formulario cuando las credenciales se encuentran incompletas, la visualización de retroalimentación ante una respuesta de autenticación rechazada y el almacenamiento del token cuando el backend simula una autenticación exitosa. De esta forma, se complementó la revisión funcional manual con evidencia automatizada sobre el comportamiento del sistema en escenarios de acceso al panel web.



Search tests		All 4	Passed 4	Failed 0	Flaky 0	Skipped 0	⌂	⚙
Project: edge		10-09-2026, 5:44:58 p. m. Total time: 24.3s						
▼ auth.spec.ts								
✓	Autenticación del panel web › redirige al login cuando se intenta acceder al mapa sin sesion edge						6.6s	
auth.spec.ts:4								
✓	Autenticación del panel web › mantiene deshabilitado el ingreso mientras faltan credenciales edge						1.2s	
auth.spec.ts:12								
✓	Autenticación del panel web › muestra retroalimentacion cuando el backend rechaza el inicio de sesion edge						1.6s	
auth.spec.ts:25								
✓	Autenticación del panel web › almacena el token cuando el inicio de sesion recibe una respuesta valida edge						2.0s	
auth.spec.ts:46								

Figura 29: Resultado de pruebas funcionales automatizadas mediante Playwright.

Fuente: Elaboración propia.

La ejecución finalizó con cuatro pruebas completadas correctamente, sin fallos reportados. Estos resultados respaldan el comportamiento esperado de los flujos iniciales de autenticación y navegación protegida, manteniendo como alcance de la validación manual otros procesos del sistema, tales como la visualización de cuadrillas, la gestión de capas KMZ/KML y el monitoreo de información en tiempo real.

3.4.4 Pruebas Unitarias

Con el propósito de fortalecer la validación técnica del front-end, se incorporó un conjunto de pruebas unitarias desarrollada con Jasmine y Karma, herramientas integradas al ecosistema de Angular. Estas pruebas se orientaron a verificar el comportamiento aislado de componentes y servicios relevantes del panel web.

Para ello, se utilizaron dependencias simuladas o mocks en aquellos casos donde el componente evaluado requería servicios externos, como solicitudes HTTP o la visualización geográfica mediante Google Maps. De esta forma, fue posible controlar las respuestas esperadas durante la ejecución de las pruebas y comprobar el funcionamiento de unidades específicas del código en un entorno controlado.

Entre los aspectos evaluados se consideran la creación de componentes base, el manejo de autenticación, la vigencia de sesión, el cierre de sesión y las operaciones principales asociadas a archivos geográficos KMZ/KML, tales como listado, carga, descarga y eliminación.

Tabla 11. Resumen de pruebas unitarias aplicadas al front-end.

Módulo evaluado	Escenario de prueba	Resultado esperado
Componente base del front-end	Creación de los componentes principales de la aplicación.	Los componentes se instancian correctamente en el entorno de pruebas.
Componente de visualización geográfica	Creación del componente principal del mapa utilizando mocks para Google Maps y servicios asociados.	El componente se crea sin depender directamente de servicios externos reales.
Servicio de autenticación	Inicio de sesión exitoso y almacenamiento de datos de sesión.	El token, refresh token y datos del usuario se almacenan correctamente.
	Manejo de inicio de sesión inválido o sin token.	El servicio retorna un error y no registra credenciales locales.
	Validación y cierre de sesión.	El sistema reconoce sesiones vigentes, detecta sesiones expiradas y elimina credenciales al cerrar sesión.
Servicio de archivos geográficos	Listado, carga, descarga y eliminación de archivos KMZ/KML.	Las solicitudes HTTP se generan correctamente según la operación requerida.

Fuente: Elaboración propia.

La Tabla 11 resume los principales escenarios abordados mediante las pruebas unitarias. Aunque la ejecución contempló trece pruebas individuales, estas se agrupan según el módulo evaluado y el propósito técnico de cada validación.

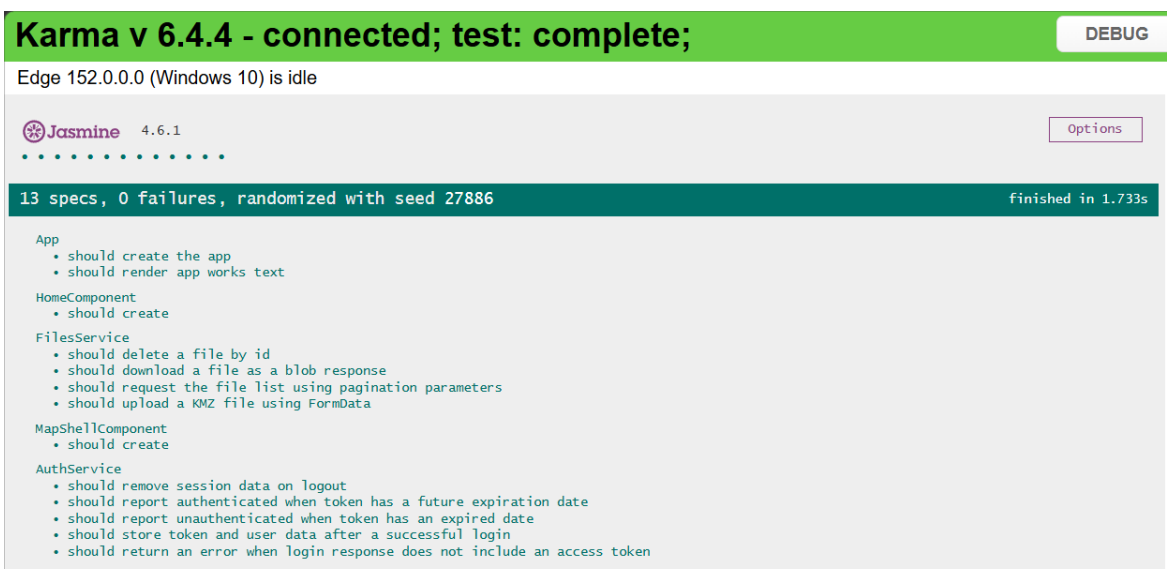


Figura 30: Ejecución de pruebas unitarias mediante Jasmine y Karma.

Fuente: Elaboración propia.

Como resultado, en todos los casos, el resultado obtenido fue satisfactorio, registrándose trece pruebas exitosas sobre trece ejecutadas. Esto permite respaldar una cobertura unitaria inicial sobre funcionalidades relevantes del front-end, especialmente en los servicios de autenticación y gestión de archivos geográficos. No obstante, esta validación no representa una cobertura exhaustiva de toda la aplicación, por lo que se reconoce como una base inicial que puede ampliarse en futuras iteraciones del sistema.

3.4.5 Evaluación de experiencia de usuario y retroalimentación

La experiencia de uso del panel web fue revisada a partir de los principales flujos de interacción definidos para el sistema. Esta revisión consideró la claridad de la interfaz, la facilidad para acceder a las funcionalidades principales y la retroalimentación entregada al usuario ante acciones exitosas o errores operacionales.

3.4.5.1 Evaluación Cualitativa de Usabilidad

La evaluación de usabilidad se realizó de manera cualitativa, considerando la interacción con usuarios de prueba durante la revisión del panel web. Esta evaluación permitió observar la claridad de la interfaz, la facilidad para identificar las funciones principales, la comprensión de los estados de conexión y la interpretación visual de las capas geográficas sobre el mapa.

Dado que no se aplicó un instrumento formal con una muestra amplia de usuarios, los resultados deben entenderse como una validación exploratoria de la experiencia de uso. Entre los aspectos revisados se consideró si el usuario podía iniciar sesión, interpretar la ubicación de las cuadrillas, activar o desactivar capas KMZ, utilizar los filtros de voltaje y reconocer mensajes de retroalimentación del sistema.

No obstante, esta evaluación presenta limitaciones, ya que no incorpora métricas cuantitativas como tiempo promedio de ejecución de tareas, tasa de error, porcentaje de éxito por flujo o nivel de satisfacción medido mediante encuesta. Por ello, se propone que futuras evaluaciones incorporen

usuarios representativos del contexto operacional y métricas específicas que permitan medir la usabilidad de forma objetiva.

3.4.6 Resultados y Limitaciones de la Validación

Los resultados obtenidos permitieron verificar el comportamiento de los principales flujos definidos para el front-end de GeoCuadrilla dentro del ambiente controlado utilizado. La validación contempló pruebas funcionales y de integración, pruebas funcionales automatizadas y un conjunto de pruebas unitarias sobre componentes y servicios relevantes del sistema, complementadas por una evaluación cualitativa de usabilidad.

No obstante, debido a las condiciones y al alcance del proyecto, no se dispone de mediciones cuantitativas formales que permitan comparar el impacto operacional antes y después de la implementación, por ejemplo, en términos de tiempos de respuesta, tiempos de asignación de cuadrillas o reducción de consultas manuales. Asimismo, las pruebas unitarias realizadas abarcan funcionalidades específicas del front-end y no constituyen una cobertura completa de la aplicación.

En consecuencia, los resultados permiten respaldar el funcionamiento de las principales funcionalidades implementadas dentro del ambiente de prueba, pero no permiten establecer conclusiones cuantitativas sobre el impacto del sistema en la eficiencia operativa ni generalizar los resultados de usabilidad a un entorno productivo real.

3.4.7 Manejo de Errores y Retroalimentación al Usuario

Durante la validación funcional también se revisó el comportamiento del sistema ante errores operacionales. La aplicación incorpora mecanismos de retroalimentación visual para informar al usuario cuando una acción no puede completarse correctamente, por ejemplo, ante fallos en la carga de archivos KMZ, errores al eliminar capas, problemas al obtener la lista de archivos o fallas de autenticación. Estos mensajes permiten que el usuario comprenda el estado de la operación sin necesidad de interpretar errores técnicos del navegador o del backend.

```
notify(text: string, kind: ToastKind = 'info', ms = 3800) {  
  const id = this.toastSeq++;  
  this.toasts.set([...this.toasts()], { id, text, kind });  
  setTimeout(() => this.dismissToast(id), ms);  
}  
dismissToast(id: number) {  
  this.toasts.set(this.toasts().filter(t => t.id !== id));  
}
```

Figura 31: Implementación de notificaciones para retroalimentación al usuario.
Fuente: Elaboración propia a partir del código fuente del proyecto GeoCuadrilla.

En operaciones críticas, como la eliminación de archivos geográficos, la interfaz aplica una actualización temporal del estado visual y conserva una copia de la información previa. Si la operación se completa correctamente, el cambio se mantiene; si ocurre un error, el sistema restaura el estado anterior, evitando inconsistencias entre la interfaz y la información real disponible en el backend. Este comportamiento funciona como una estrategia de reversión o rollback a nivel de interfaz.

```
const before = this.apiFiles.slice();

this.apiFiles = this.apiFiles.filter(f => {
  const sameId = String(f.id) === id;
  const sameNm = this.sameName(f.fileName, name);
  return !(sameId || sameNm);
});

try {
  await firstValueFrom(this.filesService.delete(id));
  this.notify(`${name}" fue eliminado.`, 'success');
} catch (e) {
  console.error('Eliminar KMZ error', e);
  this.apiFiles = before;
  this.tombstoneIds.delete(id);
  this.tombstoneNames.delete(this.normalizeName(name));
  this.notify(`No se pudo eliminar "${name}".`, 'error');
}
```

Figura 32: Restauración de estado ante error en eliminación de archivos KMZ.

Fuente: Elaboración propia a partir del código fuente del proyecto GeoCuadrilla.

En el caso de la comunicación en tiempo real, los errores de conexión se reflejan mediante cambios en el estado del servicio, diferenciando entre conectado, reconectando y desconectado. Esto permite advertir al usuario que la información de ubicación puede no encontrarse actualizada, especialmente si la desconexión se mantiene por un periodo prolongado.

```
this.connection.onreconnecting(err => {
  console.warn('[SignalR] Reconnecting..', err);
  this._connectionState$.next(ConnectionState.Reconnecting);
});

this.connection.onreconnected(id => {
  console.log('[SignalR] Reconnected. ConnId:', id);
  this._connectionState$.next(ConnectionState.Connected);
});

this.connection.onclose(err => {
  console.warn('[SignalR] Closed', err);
  this._connectionState$.next(ConnectionState.Disconnected);
});
```

Figura 33: Actualización de estado ante errores de conexión en tiempo real.

Fuente: Elaboración propia a partir del código fuente del proyecto GeoCuadrilla.

Esta lógica se observa en el uso de notificaciones `notify`, en la restauración de la lista previa cuando falla una operación de eliminación y en la actualización del estado de conexión frente a eventos del servicio en tiempo real.

De manera general, durante esta etapa se realizó una revisión funcional y exploratoria de las principales funcionalidades implementadas en el sistema dentro del ambiente de prueba disponible. Esta revisión permitió observar el comportamiento de los flujos esenciales del panel web, aunque no constituye una validación formal en un entorno operacional real. Por ello, la evaluación con métricas cuantitativas y usuarios representativos se plantea como una línea de trabajo futuro.

4 Conclusiones y trabajo futuro

4.1 Conclusiones

Este trabajo de título se orientó al desarrollo de GeoCuadrilla, una solución orientada a apoyar la gestión operativa de cuadrillas en terreno para Chilquinta Transmisión mediante la integración de información de ubicación, estado operacional y contexto geográfico en una plataforma centralizada. Con base en el desarrollo efectuado, es posible señalar que el sistema permitió abordar la necesidad de contar con una visualización más clara y oportuna de las cuadrillas, facilitando el acceso a información relevante para la toma de decisiones relacionadas con la atención de incidencias en la red eléctrica.

El sistema desarrollado integra componentes móviles, servicios backend y un panel web de supervisión, permitiendo que la información generada en terreno pueda ser procesada y representada de forma visual para el usuario operador. Esta integración posibilita visualizar la ubicación de las cuadrillas sobre un mapa interactivo, consultar su estado, revisar información asociada y complementar la supervisión mediante capas geográficas KMZ correspondientes a elementos de la red eléctrica.

Desde una perspectiva funcional, GeoCuadrilla logró implementar flujos relevantes para la operación, tales como autenticación de usuarios, comunicación en tiempo real, visualización de cuadrillas, gestión de archivos geográficos, activación de capas sobre el mapa y retroalimentación ante errores operacionales. Estas funcionalidades permiten centralizar información que anteriormente podía encontrarse dispersa o requerir consultas manuales, contribuyendo a una supervisión más ordenada y comprensible.

Desde el punto de vista técnico, el sistema evidencia la importancia de una arquitectura modular y desacoplada, donde la comunicación entre componentes se realiza mediante servicios especializados, solicitudes HTTP y mecanismos de actualización en tiempo real. El uso de tecnologías como Angular, SignalR, Google Maps y archivos KMZ/KML permitió construir una solución flexible, capaz de integrar información operacional y geoespacial en una misma interfaz.

La validación realizada permitió observar el comportamiento de los principales flujos del sistema en un ambiente controlado, incluyendo inicio de sesión, actualización de ubicaciones, visualización del mapa, carga y eliminación de archivos KMZ, activación de capas y manejo de errores. Además, se incorporaron pruebas unitarias y pruebas funcionales automatizadas para fortalecer la verificación técnica de componentes, servicios y flujos críticos de autenticación. Sin embargo, dicha revisión no incorporó mediciones cuantitativas formales que permitan afirmar una reducción objetiva de tiempos de respuesta o una mejora estadística en la eficiencia operacional.

En términos generales, GeoCuadrilla constituye una base funcional para apoyar la gestión y supervisión de cuadrillas en terreno mediante una plataforma integrada. Si bien el sistema requiere futuras validaciones en un entorno productivo real, el trabajo desarrollado permitió demostrar la viabilidad técnica de la solución y sentar las bases para su evolución, integración con sistemas internos de la empresa y mejora continua.

4.2 Limitaciones del Trabajo

Esta tesina delimitó su alcance al diseño e implementación del front-end del sistema GeoCuadrilla, por lo que no se abordó en profundidad el desarrollo del backend, la aplicación móvil ni la infraestructura productiva completa requerida para una operación real. En consecuencia, la validación del sistema se realizó principalmente desde la interfaz web, en un ambiente controlado y utilizando los servicios disponibles durante el desarrollo.

Una de las principales limitaciones corresponde a la ausencia de mediciones cuantitativas formales sobre eficiencia operativa. Si bien la aplicación centraliza información relevante para apoyar la asignación de cuadrillas y la toma de decisiones, no se realizó una comparación antes y después que permitiera medir indicadores como tiempo de asignación de cuadrilla, tiempo de desplazamiento hacia el punto de incidencia, reducción de consultas manuales o disminución efectiva de tiempos de respuesta.

Asimismo, la evaluación de usabilidad tuvo un carácter cualitativo y exploratorio. Durante el desarrollo se revisaron aspectos como claridad visual, facilidad de navegación, comprensión de los estados de conexión y uso de capas; sin embargo, no se aplicó un instrumento formal con una muestra representativa de usuarios. Por ello, no se dispone de métricas como tasa de éxito de tareas, tiempo promedio de ejecución, cantidad de errores por usuario o nivel de satisfacción medido mediante encuesta. Por otra parte, aunque se incorporaron pruebas unitarias y pruebas funcionales automatizadas, estas se concentraron en componentes, servicios y flujos críticos específicos, por lo que no constituyen una cobertura completa de toda la aplicación.

Otra limitación relevante se relaciona con el funcionamiento offline. La versión desarrollada depende de la conectividad con el backend para autenticar usuarios, recibir actualizaciones de ubicación en tiempo real y gestionar archivos. Aunque la comunicación mediante SignalR incorpora reconexión automática ante interrupciones temporales, el sistema no cuenta con un modo offline completo ni con mecanismos de sincronización diferida de datos cuando la conexión se restablece.

Finalmente, no se realizaron pruebas exhaustivas de escalabilidad en escenarios de alta carga. Por lo tanto, el comportamiento del sistema frente a un número elevado de cuadrillas, múltiples usuarios concurrentes, gran volumen de actualizaciones en tiempo real o gran cantidad de capas KMZ activas debe ser evaluado en etapas posteriores. Además, el uso de servicios externos como Google Maps puede estar sujeto a cuotas, costos, límites de uso y condiciones técnicas que deben revisarse antes de una implementación productiva.

4.3 Trabajo Futuro

Como trabajo futuro, se propone incorporar métricas cuantitativas que permitan evaluar con mayor precisión el impacto del sistema en la eficiencia operativa. Entre los indicadores sugeridos se consideran el tiempo promedio para localizar una cuadrilla, el tiempo de asignación ante una incidencia, el tiempo de desplazamiento hacia el punto de atención, la disponibilidad operativa de los equipos y la reducción de consultas manuales durante la coordinación de cuadrillas.

Como complemento, se propone ampliar progresivamente la cobertura de pruebas unitarias y automatizadas sobre los componentes y servicios principales del panel web, con el fin de fortalecer la validación técnica del sistema y detectar errores de forma temprana en futuras iteraciones.

También se recomienda realizar una evaluación formal de usabilidad con usuarios representativos del contexto operacional. Esta evaluación podría considerar pruebas guiadas sobre los principales flujos del sistema, tales como inicio de sesión, búsqueda de cuadrillas, activación de capas KMZ,

interpretación del estado de conexión y consulta de información sobre el mapa. Además, podría incorporar indicadores como tasa de éxito de tareas, cantidad de errores cometidos por usuario, tiempo promedio de ejecución y nivel de satisfacción percibida.

Otra línea de mejora corresponde a la incorporación de un modo offline con sincronización diferida, especialmente relevante para escenarios de trabajo en terreno con conectividad limitada. Esta funcionalidad permitiría conservar temporalmente información o acciones relevantes y sincronizarlas con el backend cuando la conexión vuelva a estar disponible. Para ello, sería necesario definir mecanismos de almacenamiento local, control de conflictos, validación de datos y actualización segura de la información pendiente.

Asimismo, se propone evaluar la integración del panel web con sistemas internos de la empresa, tales como plataformas de agenda operacional, autenticación institucional, sistemas de gestión de incidencias o fuentes corporativas de información geográfica. Esta integración permitiría reducir la carga manual de información, fortalecer la continuidad entre el sistema desarrollado y los procesos reales de operación, y ampliar el valor del panel como herramienta de apoyo a la toma de decisiones.

Desde el punto de vista técnico, se recomienda analizar la escalabilidad del sistema en condiciones de mayor carga, considerando cantidad de cuadrillas monitoreadas, usuarios conectados simultáneamente, frecuencia de actualizaciones en tiempo real, tamaño de archivos KMZ/KML y número de capas geográficas activas. También sería necesario revisar las cuotas, costos y límites asociados a la API de Google Maps, con el fin de asegurar que el sistema pueda operar de manera estable y sostenible en un entorno productivo.

Finalmente, se considera necesario profundizar el cumplimiento normativo asociado al tratamiento de datos personales y geolocalización. Una futura implementación debería considerar políticas de consentimiento, retención de datos, auditoría de accesos y resguardo de información conforme a la normativa chilena aplicable y a las políticas internas de la empresa. Esto permitiría fortalecer la seguridad, trazabilidad y responsabilidad en el uso de información asociada a trabajadores y cuadrillas en terreno.

5 Agradecimientos

Quiero expresar mi más profundo agradecimiento a mis padres por su amor, apoyo y confianza a lo largo de este camino. Su compañía, esfuerzo y constante fe en mí fueron fundamentales para alcanzar esta importante etapa.

A mis hermanas y a Alexis, por su cariño y por estar siempre presentes. A Danae, Mailen y Aneley, por llenar este proceso de alegría y hacer más llevaderos sus momentos más exigentes.

Asimismo, a mis amigos de la universidad, a quienes guardo un profundo cariño, por el compañerismo, la ayuda, las risas y las experiencias compartidas durante estos años. Su presencia hizo de esta etapa una experiencia enriquecedora y llena de recuerdos significativos.

Finalmente, a mi amigo Luis, por su apoyo constante, su cercanía y por acompañarme a lo largo de los años.

6 Referencias

- [1] Angular, Angular Documentation, Google LLC, 2024. [Online]. Available: <https://angular.dev/>. [Accessed: Oct. 04, 2025]
- [2] R. Fielding, *Architectural Styles and the Design of Network-Based Software Architectures*, Doctoral Dissertation, Univ. of California, Irvine, 2000.
- [3] Google Developers, *Google Maps JavaScript API Documentation*, Google LLC, 2024. [Online]. Available: <https://developers.google.com/maps/documentation/javascript?hl=es-419>. [Accessed: Oct. 04, 2025]
- [4] O'Reilly Media, *Learning Angular: A Hands-On Guide to Building Web Applications Using Angular and TypeScript*, O'Reilly Media, 2018.
- [5] I. Sommerville, *Software Engineering*, 10th ed., Pearson Education, 2016.
- [6] Figma, "Prototipo" [Online]. Available: <https://www.figma.com/design/WHkzK2kQSf1LDpyfSS7kaa/Prototipo?t=SJUw1705UVXyGWiI-0> [Accessed: Dec. 11, 2025].
- [7] Biblioteca del Congreso Nacional de Chile, Ley N.º 19.628 sobre Protección de la Vida Privada, Ministerio Secretaría General de la Presidencia, 1999. [Online]. Available: <https://www.bcn.cl/leychile/navegar?idNorma=141599> [Accessed: Jun. 10, 2026].
- [8] Biblioteca del Congreso Nacional de Chile, Ley N.º 21.719: Regula la protección y el tratamiento de los datos personales y crea la Agencia de Protección de Datos Personales, Ministerio Secretaría General de la Presidencia, 2024. [Online]. Available: <https://www.bcn.cl/leychile/navegar?idNorma=1209272> [Accessed: Jun. 10, 2026].