

**UNIVERSIDAD DE
TARAPACÁ FACULTAD DE
INGENIERÍA
INGENIERÍA CIVIL EN COMPUTACIÓN E INFORMÁTICA
ARICA – CHILE**



**Sistema de monitoreo en tiempo real para la
supervisión de flotas de vehículos en la
División Ministro Hales de Codelco:
“GeoTrackMine”**

**Equipo de Desarrollo:
Cristian Bautista Marin
Christian Caceres Marin**

**Division: Operación Mina
Ministro Hales
Curso: Proyecto IV ICCI
Profesor: Diego Aracena Pizarro**

04-06-2024

Tabla de contenido

Tabla de contenido	1
Índice de Tablas	3
Resumen	4
Objetivos	5
Objetivo general	5
Objetivos específicos	5
Definición del proyecto	6
El problema o necesidad	6
Por qué es importante abordar el proyecto	7
Descripción preliminar del producto	8
Características preliminares del producto	8
Planificación	9
Requisitos de alto nivel	9
Requisitos funcionales	9
Requisitos no funcionales	10
Carta Gantt	11
Establecimiento de Alcance	12
Diagrama de Contexto	12
Casos de Uso de Sistema	13
Modelo de Clases	25
Estructuración de Subsistemas	26
BPMN	27
Diseño	28
Arquitectura del Sistema (Cliente-Servidor)	28
Diseño de la Interfaz de Usuario	29
Diseño de la Base de Datos	34
Desarrollo	35
Configuración del Entorno de Desarrollo	35
Preparación de la Máquina Virtual	35
Configuración del Backend	37
Configuración de la Base de Datos	38
Gestión del Servidor Web y Procesos	39
Configuración del Frontend	40
Desarrollo de la Aplicación	42
Desarrollo de la base de datos	42
Desarrollo Backend	45
Desarrollo Frontend	48
Conclusión	49
Referencias	50

Índice de Figuras

Figura 1: Carta Gantt.....	11
Figura 2: Diagrama de Contexto.....	12
Figura 3: Diagrama de Casos de Usos.....	13
Figura 4: Diagrama Iniciar Sesión.....	14
Figura 5: Diagrama Iniciar Sesión.....	15
Figura 6: Diagrama Recepción de Datos de Ubicación.....	17
Figura 7: Diagrama Recepción de Datos de Ubicación.....	18
Figura 8: Diagrama Generación de Alertas.....	20
Figura 9: Diagrama Generación de Alertas.....	21
Figura 10: Diagrama Gestión de Alertas.....	23
Figura 11: Diagrama Gestión de Alertas.....	24
Figura 12: Diagrama de Modelo de Clases UML.....	25
Figura 13: Diagrama de Subsistemas.....	26
Figura 14: BPMN.....	27
Figura 15: Arquitectura de GeoTrackMine.....	28
Figura 16: Ejecución de GeoTrackMine.....	29
Figura 17: Interfaz de Inicio de Sesión.....	30
Figura 18: Interfaz de Registro de Alertas.....	31
Figura 19: Interfaz de Detalle de Alerta.....	32
Figura 20: Interfaz de Notificación de Alertas.....	33
Figura 21: Modelo Entidad Relación.....	34
Figura 22: Configuración de la Máquina Virtual.....	35
Figura 23: Máquina virtual en Virtualbox.....	36
Figura 25: Conexión SSH y Github.....	36
Figura 26: Clonación de repositorio del Backend.....	37
Figura 27: Instalación de Nodejs mediante NPM.....	37
Figura 28: Creación del Esquema GeoTrackMine.....	38
Figura 29: Creación de las tablas del esquema GeoTrackMine.....	38
Figura 30: Configuración NGINX.....	39
Figura 31: Configuración de Supervisor.....	39
Figura 32: Entorno de desarrollo de Electronjs en desarrollo.....	40
Figura 33: Interfaz inicio de sesión creada con ElectronEs.....	40
Figura 34: Interfaz de Registro de Alertas.....	41
Figura 35: Interfaz de Visualización de notificaciones.....	41
Figura 36: Primera parte del script SQL.....	42
Figura 37: Segunda parte script SQL.....	43
Figura 38: Última parte del script SQL.....	44
Figura 39: Archivo Principal.....	45
Figura 40: Funcionalidad de enrutador dinámico.....	46
Figura 41: Conexión con la base de datos.....	47
Figura 42: Estructura de carpetas de ElectronJs.....	48

Índice de Tablas

Tabla 1: Requisitos de Alto Nivel.....	9
Tabla 2: Requisitos Funcionales.....	9
Tabla 3: Requisitos No Funcionales.....	10

Resumen

El proyecto GeoTrackMine se configura como una solución tecnológica destinada a mejorar el seguimiento y control de flotas de vehículos en entornos mineros, particularmente en las operaciones llevadas a cabo por Codelco División Ministro Hales.

Este enfoque responde a la necesidad crítica de supervisar con precisión una considerable cantidad de vehículos mineros, cuyas posiciones y actividades deben ser constantemente actualizadas y evaluadas para asegurar tanto la eficiencia operacional como la seguridad dentro de la mina. El reto radica en la capacidad del administrador de mantener una vigilancia efectiva sobre estos vehículos, cada uno ejecutando tareas específicas en diferentes tiempos y ubicaciones, lo cual se complica debido a la magnitud y dinámica de la flota en movimiento.

La propuesta consiste en el desarrollo de una aplicación que opera en segundo plano, diseñada específicamente para recolectar, analizar y presentar datos geoespaciales de los vehículos en tiempo real.

Este sistema verificará que la ubicación y las actividades de los vehículos concuerden con los parámetros establecidos, alertando al administrador en caso de detectar discrepancias. Esto facilita la tarea de supervisión, posibilitando una rápida intervención ante desviaciones y contribuyendo a la optimización de procesos y seguridad.

El proyecto se alinea con el objetivo de Codelco de integrar soluciones tecnológicas avanzadas para la gestión de operaciones mineras, buscando mejorar la eficiencia, seguridad y establecer un precedente en la adopción de tecnología de punta para la gestión de flotas en la industria minera.

Objetivos

Objetivo general

Desarrollar un sistema de monitoreo en tiempo real para la supervisión de flotas de vehículos en la División Ministro Hales de Codelco.

Objetivos específicos

- Asegurar la integración del nuevo sistema para que pueda acceder y procesar datos de ubicación en tiempo real desde el sistema de GPS de la flota.
- Desarrollar un sistema que notifique desviaciones como vehículos en zonas no autorizadas o fuera de las rutas asignadas.
- Crear una interfaz efectiva que permita al despachador recibir y gestionar las alertas generadas por el sistema.
- Ejecutar pruebas en un entorno controlado para asegurar que el sistema funciona correctamente antes de su implementación completa.
- Implementar el sistema en un entorno de desarrollo, asegurando que todas las herramientas y tecnologías sean compatibles y funcionan correctamente.

Definición del proyecto

El problema o necesidad

En la División Ministro Hales de Codelco, se enfrentan desafíos significativos en la supervisión de flotas de vehículos mineros debido a las limitaciones del sistema de monitoreo GPS existente. Este sistema carece de integración con herramientas que permitan una supervisión efectiva en tiempo real y una rápida respuesta a incidencias operativas. La falta de funcionalidades como alertas en tiempo real y un registro histórico accesible de datos impide una gestión eficiente, incrementando el riesgo de ineficiencias operativas y problemas de seguridad.

Actualmente, el despachador tiene capacidades limitadas para detectar y responder a vehículos que entran en zonas no autorizadas o se desvían de rutas establecidas, lo que puede causar retrasos críticos en la toma de decisiones. Además, sin un sistema adecuado para registrar y analizar los datos históricos, es complicado llevar a cabo análisis que podrían identificar patrones de riesgos y mejorar las estrategias de operación.

Por lo tanto, existe una necesidad urgente de desarrollar una solución que integre el sistema de GPS con una plataforma de monitoreo en tiempo real capaz de generar alertas automáticas y mantener un registro detallado de todas las actividades vehiculares. Este sistema mejoraría sustancialmente la supervisión de la flota, garantizando que las operaciones dentro de la mina se realicen de acuerdo con los estándares de seguridad y eficiencia operacional, y facilitando una gestión basada en datos precisos y actuales.

Por qué es importante abordar el proyecto

La importancia de desarrollar el sistema GeoTrackMine para la División Ministro Hales de Codelco radica en varios factores críticos que impactan directamente la eficiencia operativa y la seguridad en las operación minera:

- **Mejora de la Seguridad:** La minería implica riesgos significativos, donde la seguridad de los operarios es primordial. Un sistema de monitoreo en tiempo real que pueda alertar inmediatamente sobre desviaciones de rutas o ingresos a zonas no autorizadas puede prevenir accidentes y situaciones de peligro, salvaguardando la vida del personal y protegiendo los activos de la empresa.
- **Aumento de la Eficiencia Operativa:** La capacidad para monitorear y gestionar la flota de vehículos en tiempo real permite optimizar rutas y asignaciones de tareas. Esto se traduce en una reducción de tiempos muertos y una mayor coordinación en las operaciones, lo que conduce a un incremento significativo en la productividad y reducción de costos operacionales.
- **Cumplimiento de Normativas:** Las regulaciones en la industria minera son estrictas, especialmente en términos de seguridad y medio ambiente. Un sistema eficiente y confiable ayuda a asegurar que todas las operaciones cumplen con las normativas pertinentes, evitando sanciones legales y contribuyendo a la operación sostenible de la mina.
- **Análisis y Mejora Continua:** La capacidad de almacenar y analizar datos históricos de las operaciones de los vehículos permite identificar tendencias y patrones que pueden ser usados para mejorar continuamente las estrategias de operación y seguridad. Este análisis puede llevar a innovaciones significativas en los procedimientos y técnicas utilizadas en la mina.
- **Respuesta Rápida a Incidentes:** En caso de incidentes, tener un sistema que registre y alerte en tiempo real permite una respuesta rápida y eficaz. Esto no solo ayuda a manejar el incidente de manera más efectiva sino que también reduce el impacto potencial en las operaciones.

Descripción preliminar del producto

El sistema GeoTrackMine será una aplicación de software diseñada para funcionar en segundo plano en los sistemas informáticos de la División Ministro Hales de Codelco. Esta herramienta está orientada a mejorar la supervisión y gestión de la flota de vehículos mineros a través de un monitoreo en tiempo real, utilizando datos provenientes del sistema de GPS integrado en cada vehículo.

Características preliminares del producto

- **Integración con GPS:** GeoTrackMine se conectará directamente con el sistema GPS existente, capturando datos de ubicación de todos los vehículos de la flota en tiempo real.
- **Módulo de Alertas Automatizadas:** El sistema estará equipado con un módulo de alertas que notificará al despachador sobre cualquier anomalía detectada, como desviaciones de la ruta planificada o ingreso a zonas restringidas.
- **Interfaz de Usuario Intuitiva:** Diseñada para ser clara y fácil de usar, permitirá al despachador recibir y gestionar alertas de forma eficiente.
- **Registro de Datos:** Contará con una base de datos para almacenar históricamente la información recogida, que incluirá ubicaciones, alertas y otros eventos relevantes.

Planificación

Diagramas, tablas y descripciones complementarias para la especificación del modelado y desarrollo del proyecto GeoTrackMine para la Minera Codelco Ministro Hales.

Requisitos de alto nivel

Tabla 1: Requisitos de Alto Nivel.

ID	Requisito	Descripción	Prioridad
RAN1	Integración con Sistemas de Monitoreo de GPS	El sistema deberá integrarse con el software de monitoreo de GPS existente para obtener datos de ubicación en tiempo real de los vehículos.	Alta
RAN2	Procesamiento Automático de Datos de Ubicación	Automatizar el análisis de datos de ubicación para determinar si los vehículos se encuentran dentro de sus áreas asignadas.	Alta
RAN3	Detección y Notificación de Discrepancias	El sistema identificará discrepancias entre la ubicación/actividad real de los vehículos y sus asignaciones, notificando al despachador mediante una interfaz de alerta.	Media

Requisitos funcionales

Tabla 2: Requisitos Funcionales.

ID	Requisito	Descripción	Prioridad
RF1	Seguimiento en tiempo real	El sistema debe actualizar la posición de cada vehículo cada 5 segundos.	Alta
RF2	Análisis de actividad	Debe analizar la actividad de los vehículos cada 20 segundos y compararla con su posición	Alta
RF3	Alertas de actividad	Emitir alertas si la actividad de un vehículo no coincide con su ubicación geoespacial.	Media
RF4	Registro de actividades	Mantener un registro histórico de todas las posiciones y actividades de los vehículos.	Baja
RF5	Interfaz de usuario	Proporcionar una interfaz para visualizar en tiempo real la información de los vehículos y las alertas generadas.	Alta

Requisitos no funcionales

Tabla 3: Requisitos No Funcionales.

ID	Requisito	Descripción	Prioridad
RNF 1	Rendimiento	El sistema debe ser capaz de procesar la información de hasta 100 vehículos simultáneamente sin retrasos.	Bajo
RNF 2	Disponibilidad	El sistema debe estar operativo continuamente, con mínimas interrupciones, para apoyar las operaciones de supervisión las 24 horas del día, los 7 días de la semana.	Media
RNF 3	Seguridad	Debe asegurar la confidencialidad e integridad de los datos, implementando estándares de seguridad apropiados para proteger la información contra accesos no autorizados y ataques cibernéticos.	Alta
RNF 4	Extensibilidad	El diseño del sistema debe permitir futuras expansiones o modificaciones sin afectar las operaciones existentes, facilitando la extensibilidad del sistema..	Alta
RNF 5	Interoperabilidad	El sistema debe ser compatible con diferentes plataformas y sistemas operativos utilizados en las operaciones mineras, facilitando la integración con otras aplicaciones.	Alta
RNF 6	Usabilidad	La interfaz debe ser clara y accesible para usuarios de todos los niveles técnicos, garantizando una experiencia de usuario eficiente y reduciendo la curva de aprendizaje.	Media

Carta Gantt

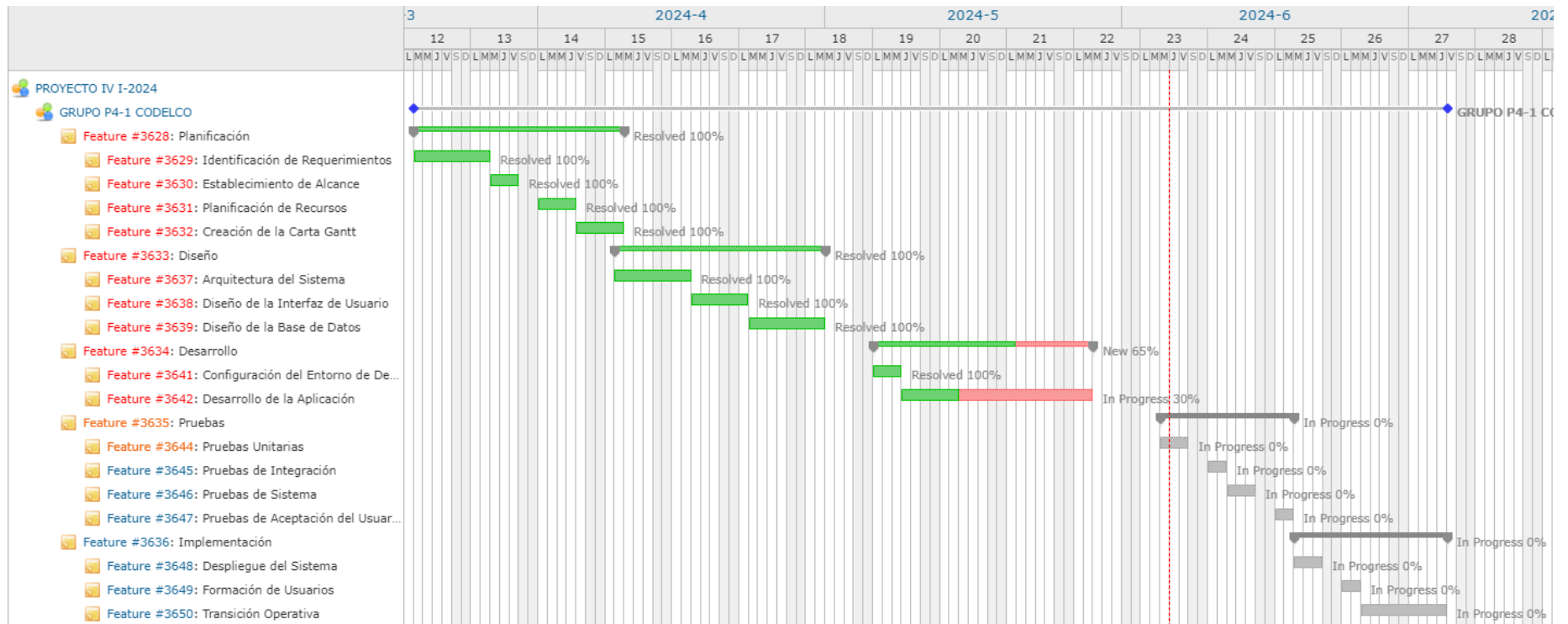


Figura 1: Carta Gantt.

Establecimiento de Alcance

El establecimiento del alcance del proyecto GeoTrackMine es una etapa crucial en la planificación, ya que define las fronteras del sistema y las funcionalidades a desarrollar, asegurando que todas las partes interesadas comprendan claramente los límites del sistema y las interacciones con actores externos. Utilizando herramientas de modelado UML como el diagrama de contexto, los casos de uso y los modelos estáticos de clases y atributos, se delimita con precisión el sistema y sus interacciones, garantizando el cumplimiento de todos los requisitos críticos. Estas herramientas UML permiten establecer una visión clara del sistema y su entorno, capturando las necesidades de los usuarios y asegurando una organización estructurada y bien definida. Además, la estructuración de subsistemas facilita la asignación de responsabilidades y mejora la mantenibilidad y escalabilidad del sistema. El uso de UML no solo ayuda a definir y comunicar el alcance del proyecto de manera precisa, sino que también proporciona una base sólida para el desarrollo y la implementación, asegurando que el proyecto se mantenga enfocado en sus objetivos primarios y cumpla con todos los requisitos críticos.

Diagrama de Contexto

El diagrama de contexto es una herramienta esencial en el modelado UML que proporciona una visión de alto nivel del sistema y sus interacciones con actores externos. En el proyecto GeoTrackMine, este diagrama se utiliza para definir claramente las fronteras del sistema y cómo interactúa con el Despachador y el Sistema GPS. Este enfoque preciso y conciso asegura que se capturen todas las interacciones críticas, estableciendo un entendimiento claro de los puntos de integración y flujos de información entre el sistema y su entorno. Al delinear estas interacciones, el diagrama de contexto facilita la comprensión de los límites del sistema, garantizando que se aborden adecuadamente todos los requisitos relevantes durante la fase de desarrollo.

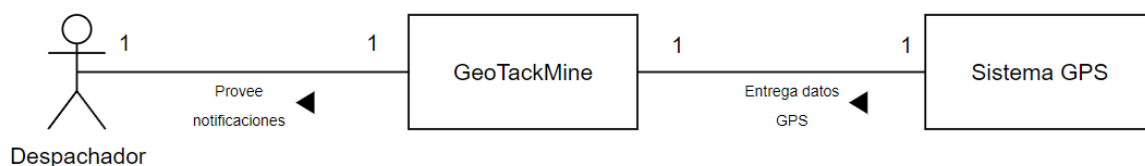


Figura 2: Diagrama de Contexto.

Casos de Uso de Sistema

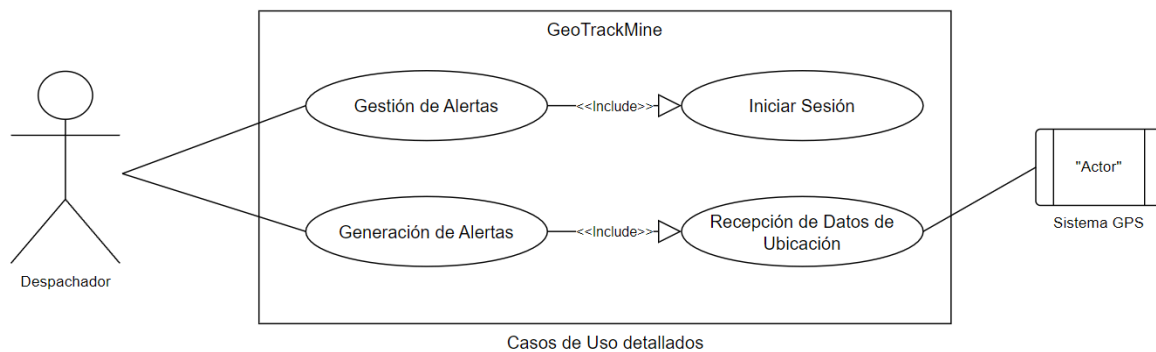


Figura 3: Diagrama de Casos de Usos.

Caso de Uso: Iniciar Sesión

Nombre CUS: Iniciar Sesión	
Descripción: El despachador se autentica en el sistema GeoTrackMine para acceder a sus funcionalidades.	
Actor: Despachador	
Precondición: El despachador debe tener una cuenta registrada en el sistema.	
Flujo Principal: Despachador	Flujo Principal: Sistema
1.- El despachador ingresa su nombre de usuario y contraseña. 2.-El despachador envía los datos para autenticación.	3.- El sistema verifica las credenciales (nombre de usuario y contraseña). 4.- Si las credenciales son correctas, el sistema otorga acceso al despachador y muestra la interfaz principal.
Flujo Alternativo: Cliente	Flujo Alternativo: Sistema
	4.1.- Si las credenciales son incorrectas, el sistema muestra un mensaje de error indicando que el nombre de usuario o la contraseña son incorrectos.
Postcondiciones: El despachador accede a la interfaz principal del sistema.	
Valor medible: Tiempo de respuesta del sistema para la autenticación y acceso.	

Diagrama de Secuencia Nivel 0: Iniciar Sesión

Iniciar Sesión

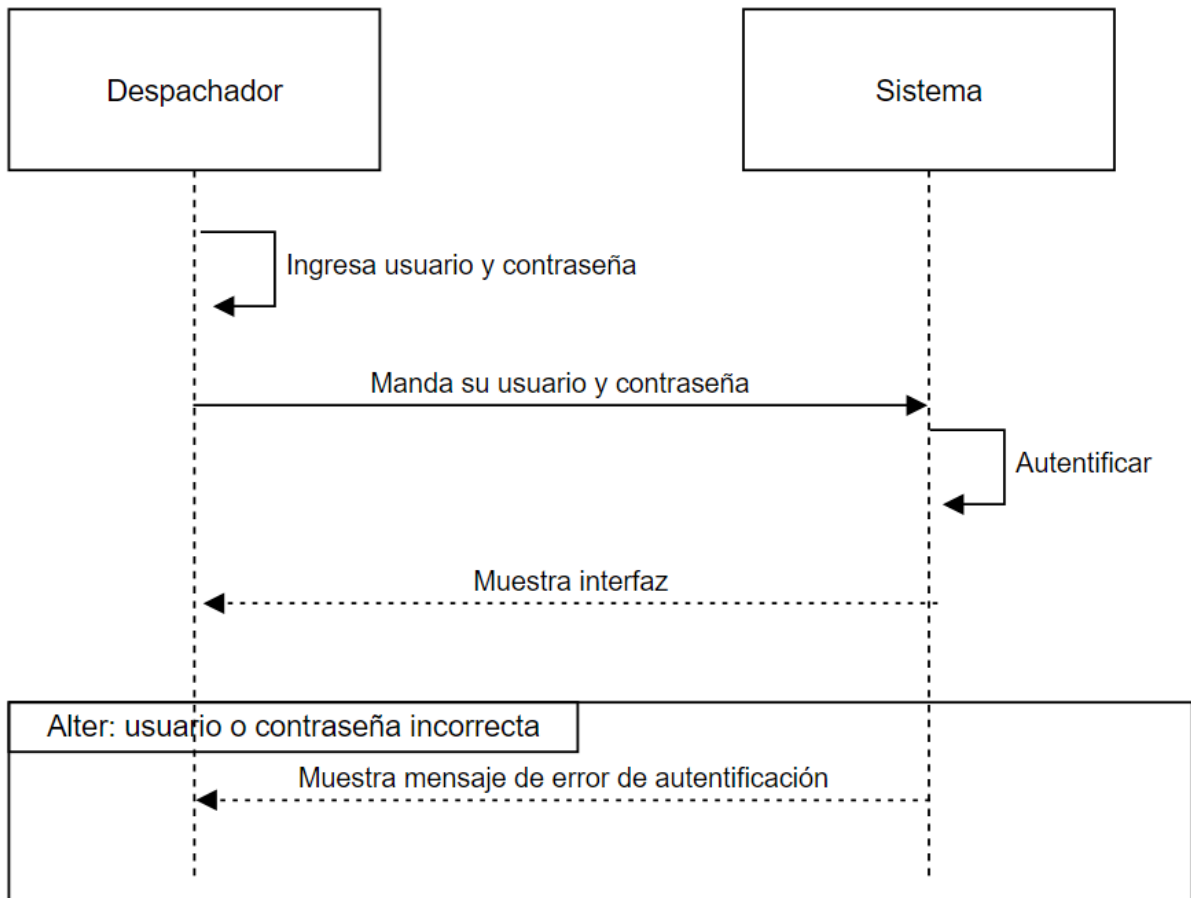


Figura 4: Diagrama Iniciar Sesión.

Diagrama de Secuencia Nivel 1: Iniciar Sesión

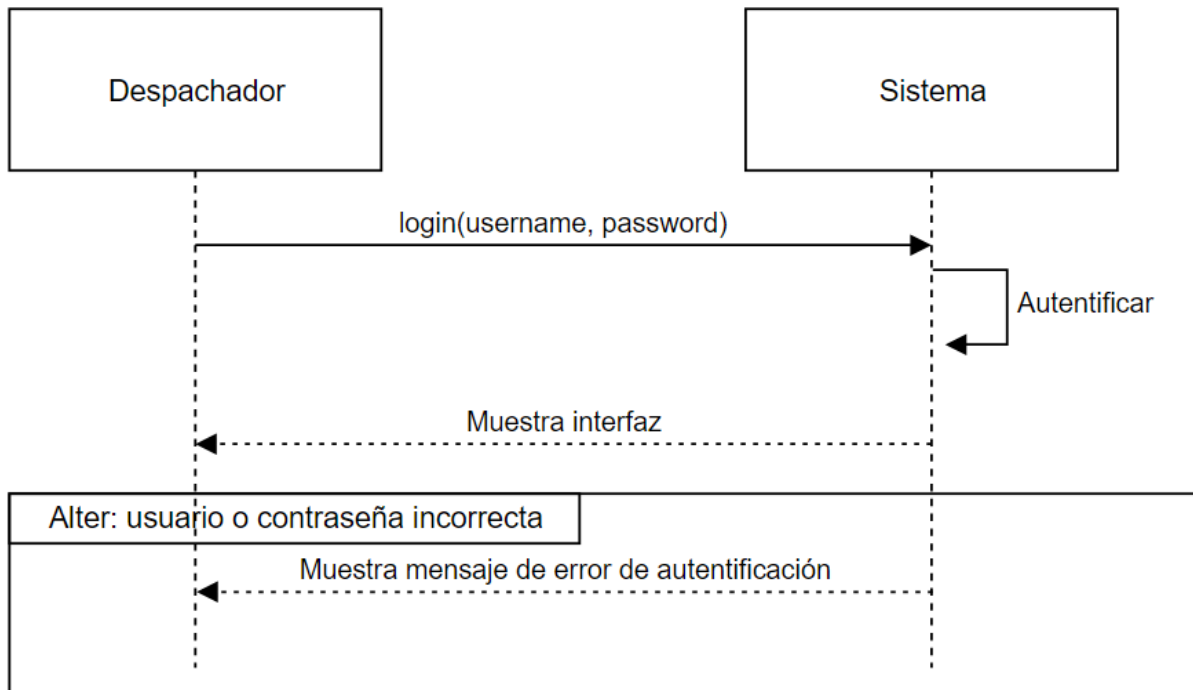


Figura 5: Diagrama Iniciar Sesión.

Caso de Uso: Recepción de Datos de Ubicación

Nombre CUS: Recepción de Datos de Ubicación	
Descripción: GeoTrackMine recibe y procesa datos de ubicación GPS en tiempo real desde el Sistema GPS de Codelco.	
Actor: Sistema GPS	
Precondición: El Sistema GPS debe estar funcionando y conectado a GeoTrackMine.	
<u>Flujo Principal: Sistema GPS</u> 1.- El Sistema GPS envía datos de ubicación a GeoTrackMine cada 5 segundos.	<u>Flujo Principal: Sistema</u> 2.- GeoTrackMine recibe los datos de ubicación. 3.- GeoTrackMine procesa los datos. 4.- GeoTrackMine actualiza la base de datos.
<u>Flujo Alternativo: Cliente</u>	<u>Flujo Alternativo: Sistema</u> 2.1.- Muestra el error en la recepción de datos. 2.2.- GeoTrackMine registra el error y reintenta la conexión con el Sistema GPS.
Postcondiciones: GeoTrackMine actualiza la ubicación de los vehículos en su base de datos.	
Valor medible: Tasa de éxito en la recepción de datos.	

Diagrama de Secuencia Nivel 0: Recepción de Datos de Ubicación

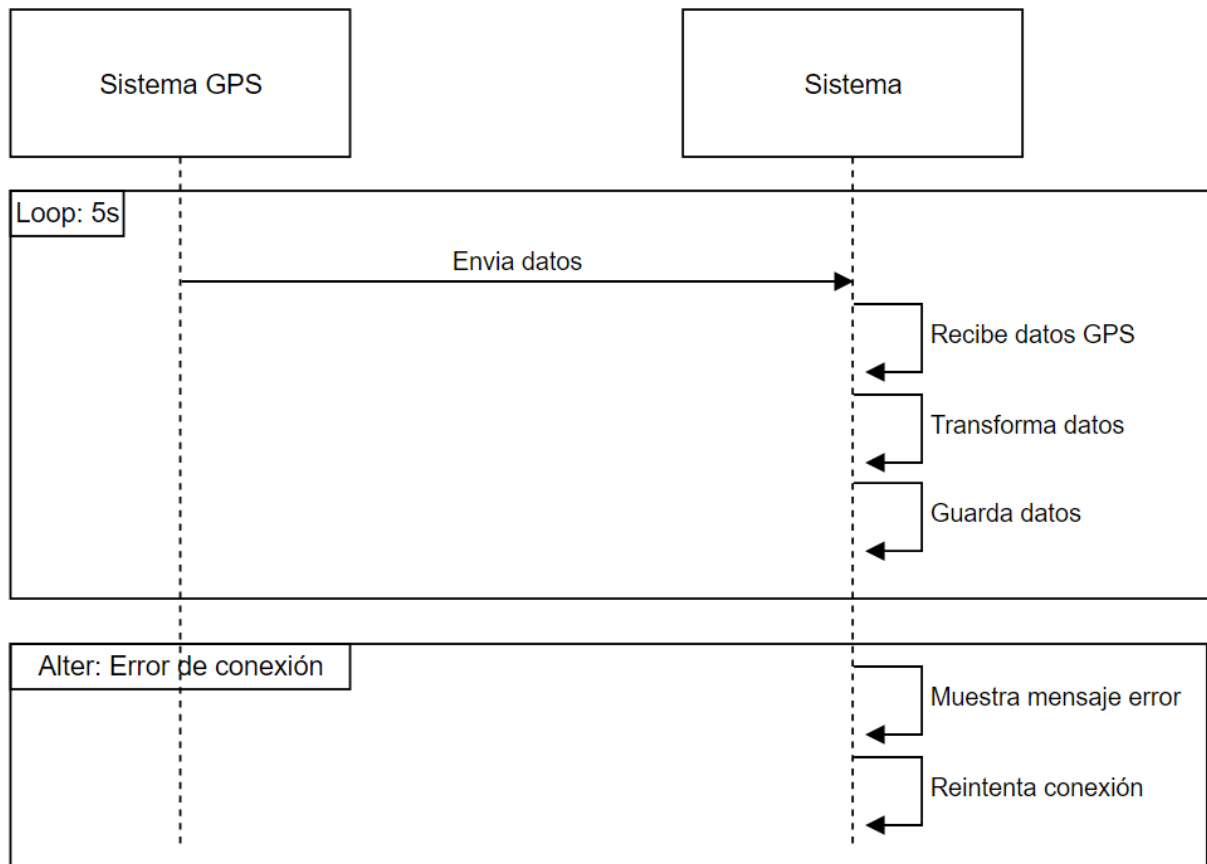


Figura 6: Diagrama Recepción de Datos de Ubicación.

Diagrama de Secuencia Nivel 1: Recepción de Datos de Ubicación

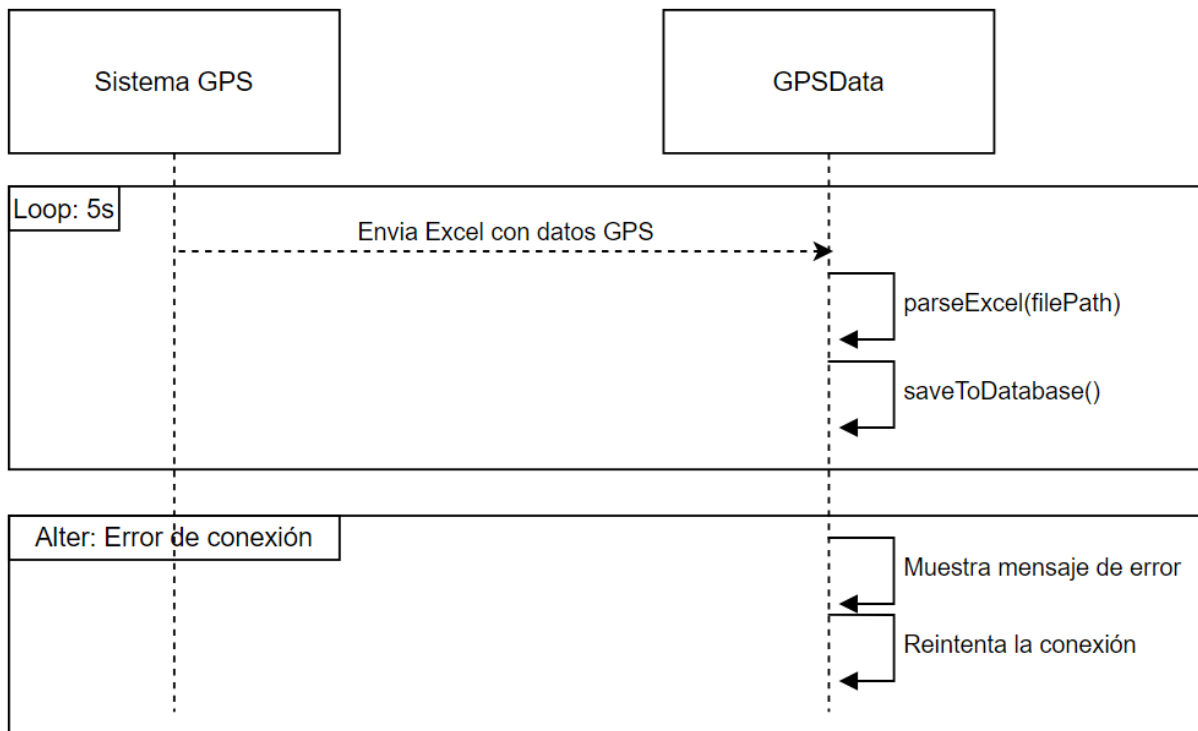


Figura 7: Diagrama Recepción de Datos de Ubicación.

Caso de Uso: Generación de Alertas

Nombre CUS: Generación de Alertas	
Descripción: GeoTrackMine genera alertas cuando detecta ineficiencias operativas basadas en los datos de ubicación recibidos.	
Actor: Despachador	
Precondición: GeoTrackMine debe estar recibiendo datos de ubicación en tiempo real.	
<p><u>Flujo Principal: Despachador</u></p> <p>5.- El despachador recibe la alerta en tiempo real.</p>	<p><u>Flujo Principal: Sistema</u></p> <p>1.- GeoTrackMine recibe y procesa datos de ubicación desde el Sistema GPS (<<include>> Recepción de Datos de Ubicación).</p> <p>2.- GeoTrackMine analiza los datos de ubicación cada 20 segundos.</p> <p>3.- GeoTrackMine detecta una ineficiencia operativa (e.g., vehículo en zona no autorizada).</p> <p>4.- GeoTrackMine genera y guarda la alerta.</p>
<p><u>Flujo Alternativo: Cliente</u></p>	<p><u>Flujo Alternativo: Sistema</u></p> <p>3.1.- No se detectan ineficiencias.</p> <p>3.2.- GeoTrackMine continúa el monitoreo sin generar alertas.</p>
Postcondiciones: El despachador recibe una alerta sobre una ineficiencia operativa.	
Valor medible: Número de alertas generadas y tiempo de detección.	

Diagrama de Secuencia Nivel 0: Generación de Alertas

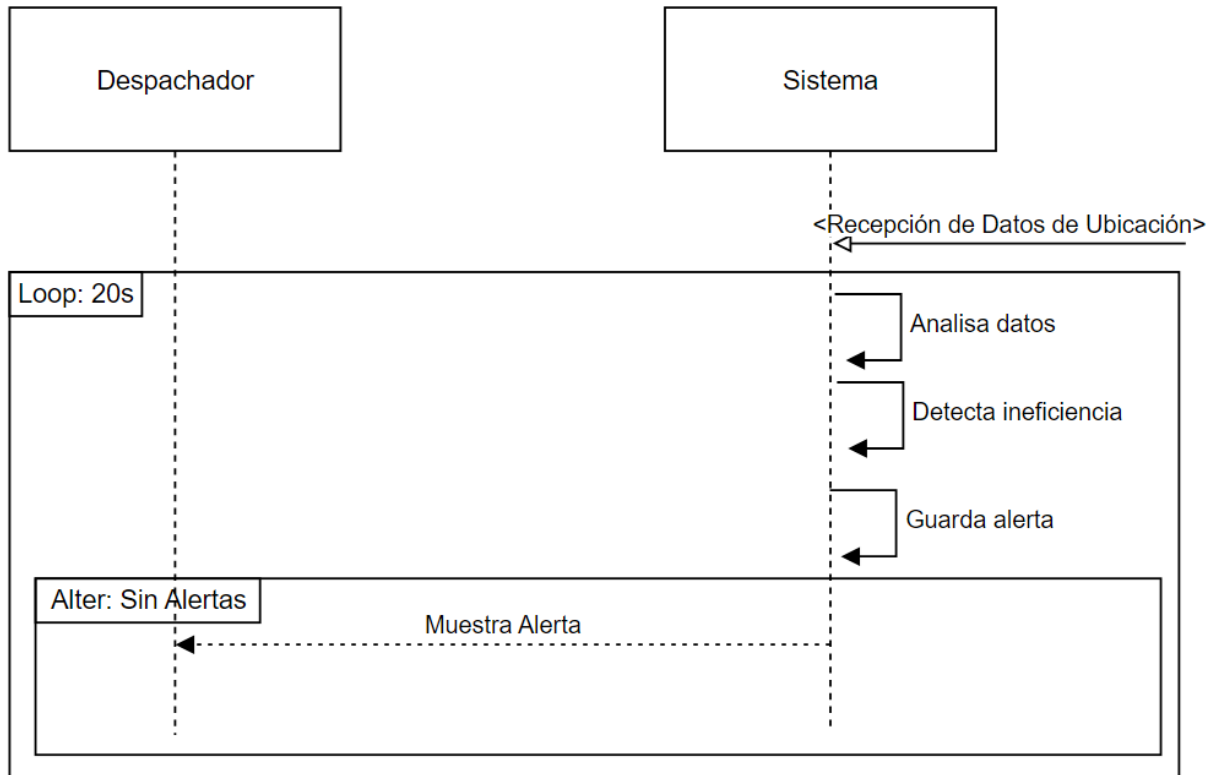


Figura 8: Diagrama Generación de Alertas.

Diagrama de Secuencia Nivel 1: Generación de Alertas

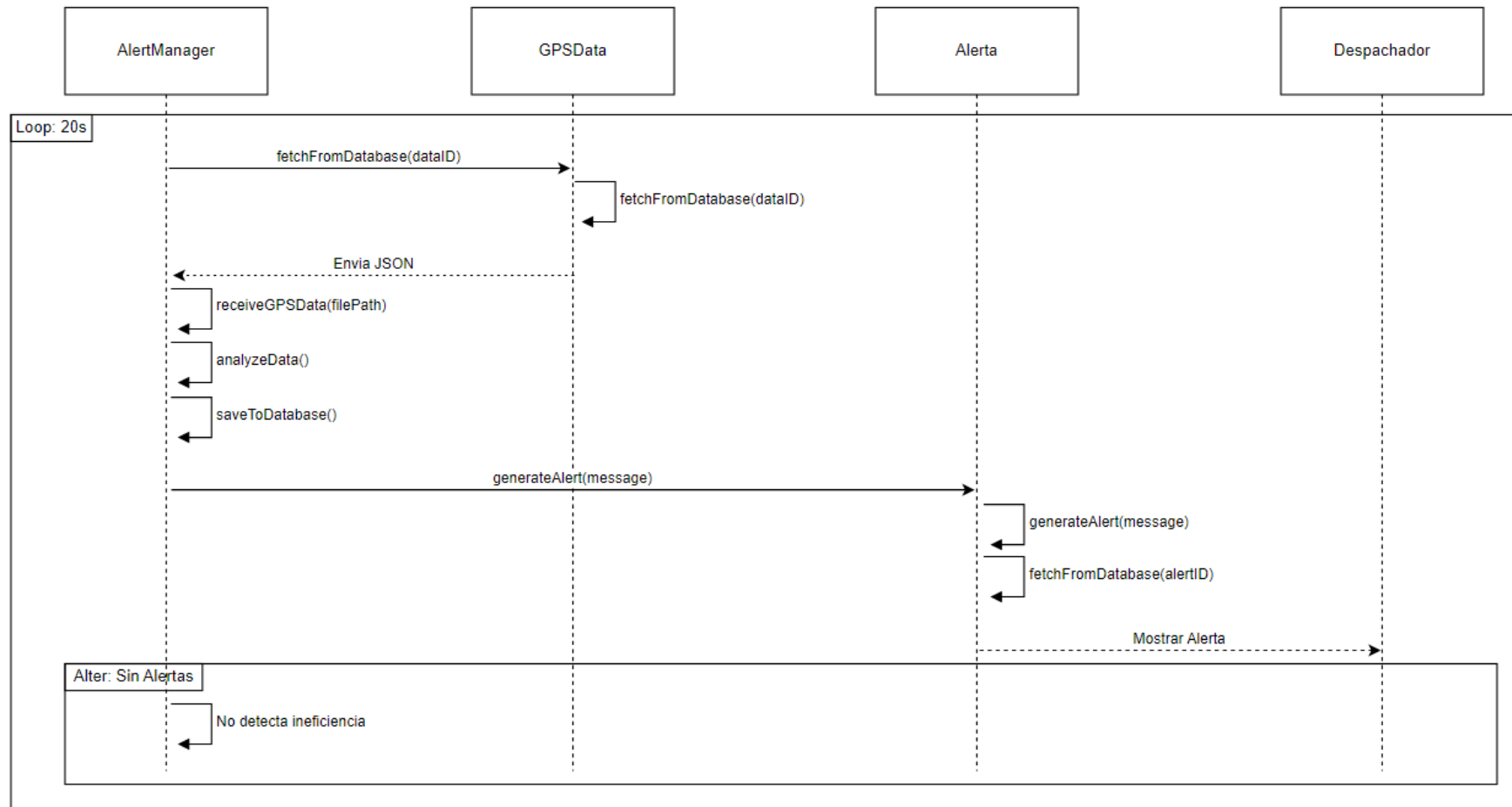


Figura 9: Diagrama Generación de Alertas

Caso de Uso: Gestión de Alertas

Nombre CUS: Gestión de Alertas	
Descripción: El despachador gestiona las alertas generadas por GeoTrackMine, visualiza el historial de alertas y puede revisar detalles específicos sobre cada una de ellas.	
Actor: Despachador	
Precondición: El despachador debe haber iniciado sesión en GeoTrackMine y recibir una alerta	
<u>Flujo Principal: Despachador</u> 1.- El despachador inicia sesión en GeoTrackMine (<<include>> Iniciar Sesión). 2.- El despachador realiza un doble clic en el icono de la aplicación para abrir la interfaz. 4.- El despachador selecciona una alerta de la lista. 6.- El despachador revisa los detalles de la alerta seleccionada. 7.- El despachador toma las acciones necesarias (e.g., reubicar el vehículo, notificar al conductor). 8.- El despachador marca la alerta como resuelta en GeoTrackMine. 11.- El despachador puede gestionar otras alertas o cerrar la interfaz.	<u>Flujo Principal: Sistema</u> 3.- La interfaz muestra las alertas ordenadas por fecha de ocurrencia. 5.- GeoTrackMine muestra los detalles de la alerta seleccionada. 9.- GeoTrackMine registra la resolución de la alerta. 10.- GeoTrackMine actualiza y muestra el historial de alertas en la interfaz del despachador.
<u>Flujo Alternativo: Despachador</u> 6.1.- El despachador no puede resolver la alerta inmediatamente. 6.2.- La alerta permanece activa hasta que se resuelva. 6.3.- El despachador puede volver a la alerta más tarde y tomar las acciones necesarias.	<u>Flujo Alternativo: Sistema</u> 8.1.- El sistema no puede actualizar la base de datos con la resolución de la alerta. 8.2.- GeoTrackMine muestra un mensaje de error al despachador y reintenta la operación.
Postcondiciones: El despachador gestiona y cierra la alerta, y puede ver el historial de alertas ordenadas por fecha de ocurrencia.	
Valor medible: Tiempo de resolución de alertas y número de alertas gestionadas.	

Diagrama de Secuencia Nivel 0: Gestión de Alertas

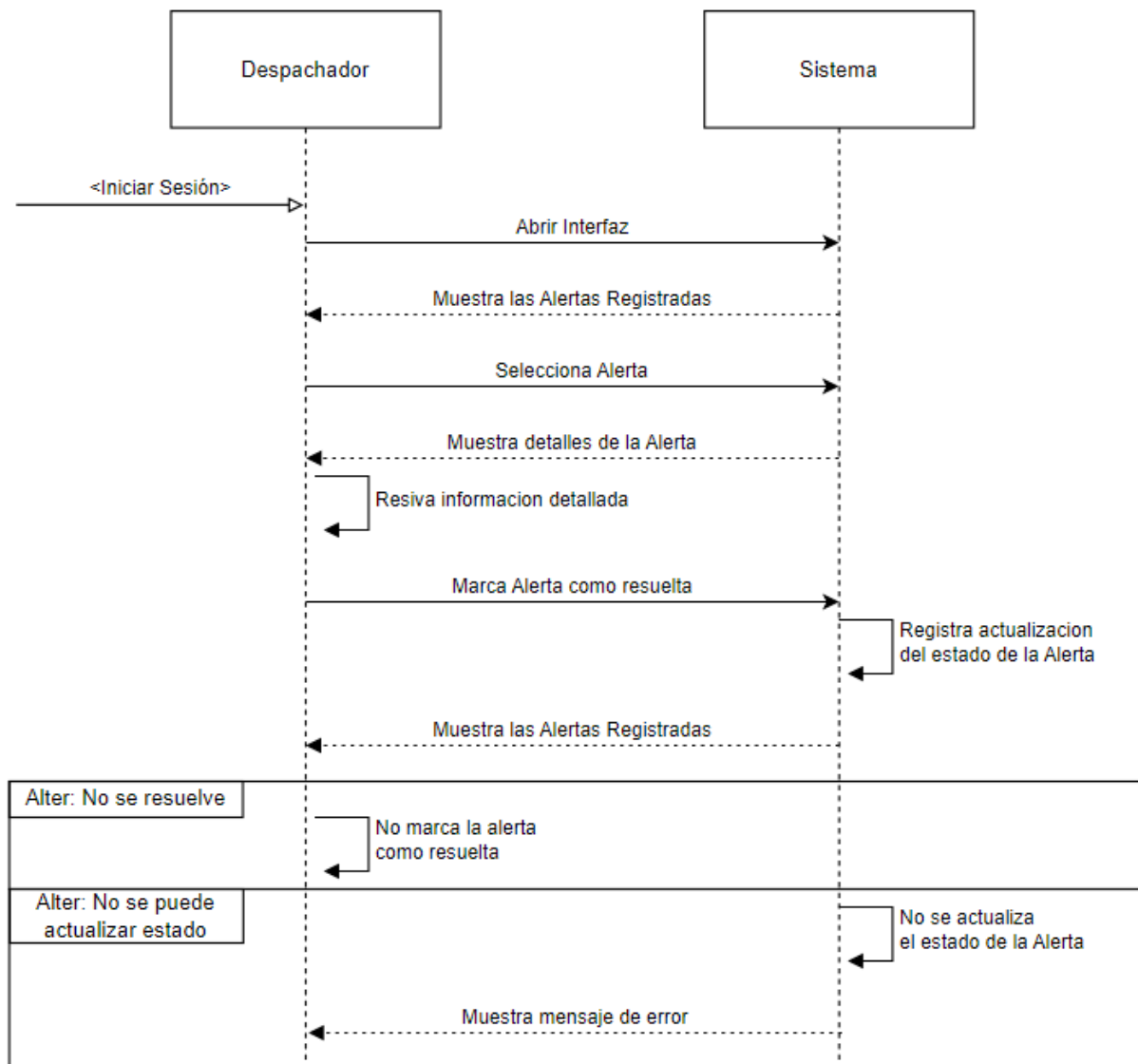


Figura 10: Diagrama Gestión de Alertas.

Diagrama de Secuencia Nivel 1: Gestión de Alertas

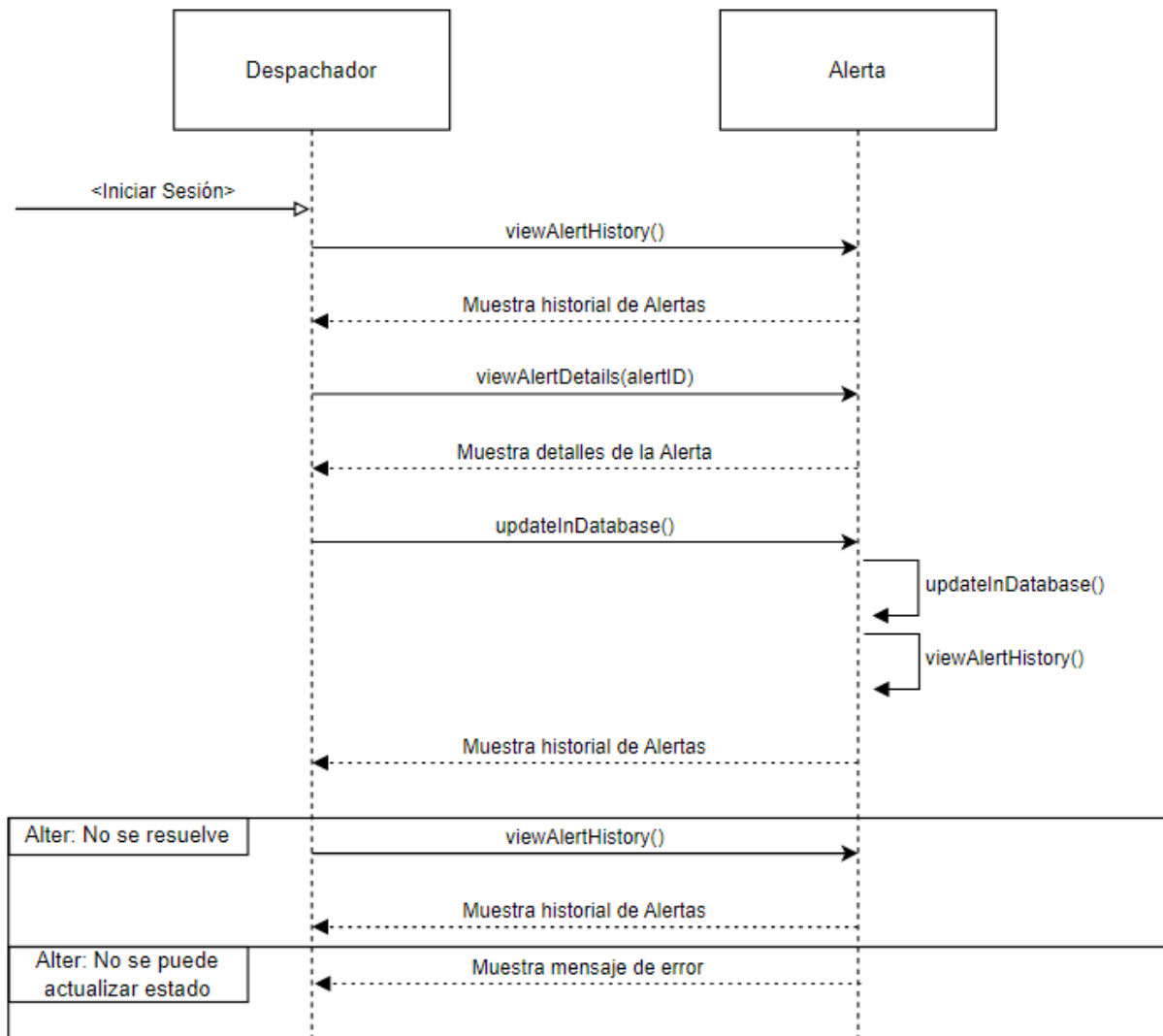


Figura 11: Diagrama Gestión de Alertas.

Modelo de Clases

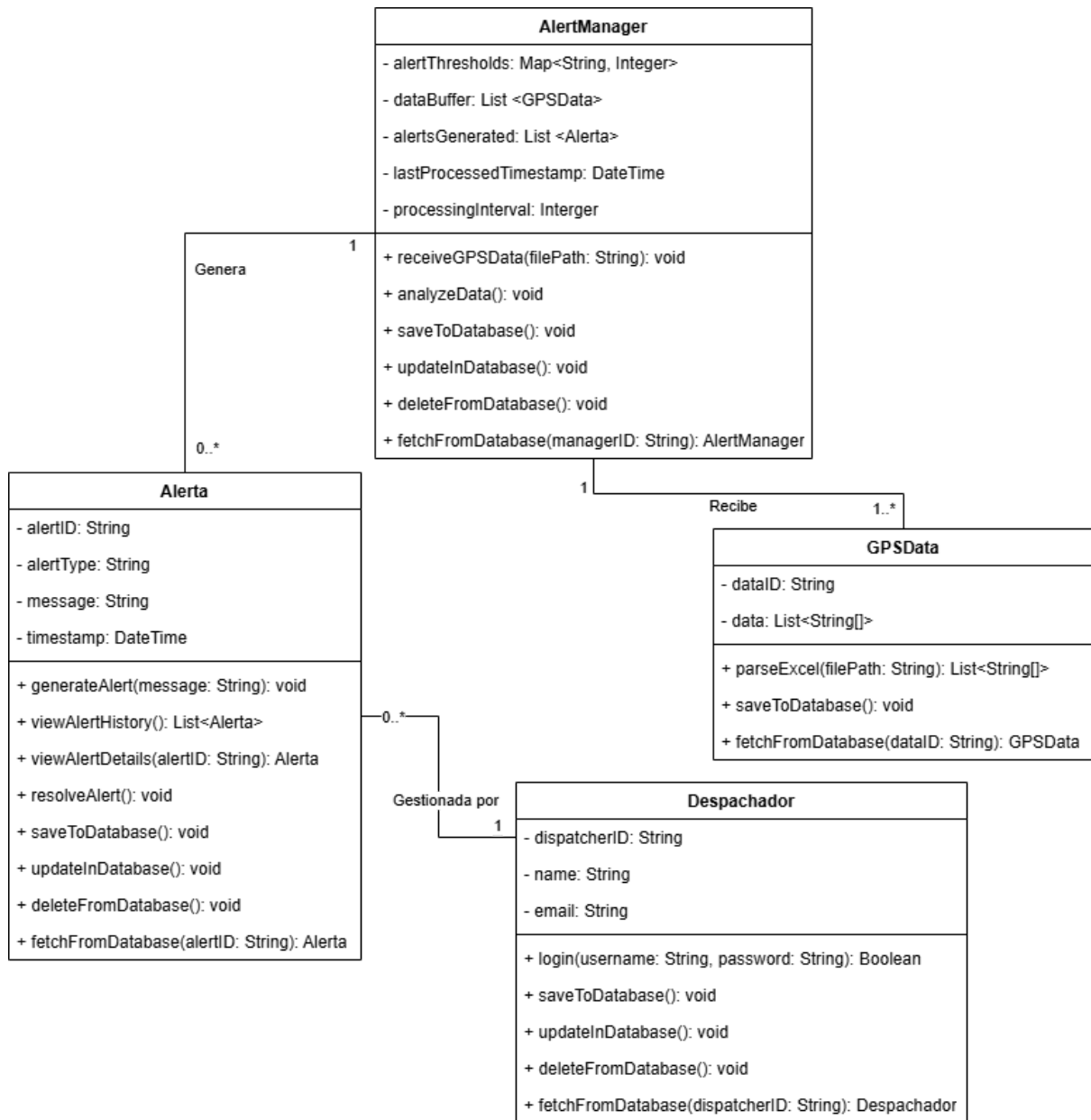


Figura 12: Diagrama de Modelo de Clases UML.

Estructuración de Subsistemas

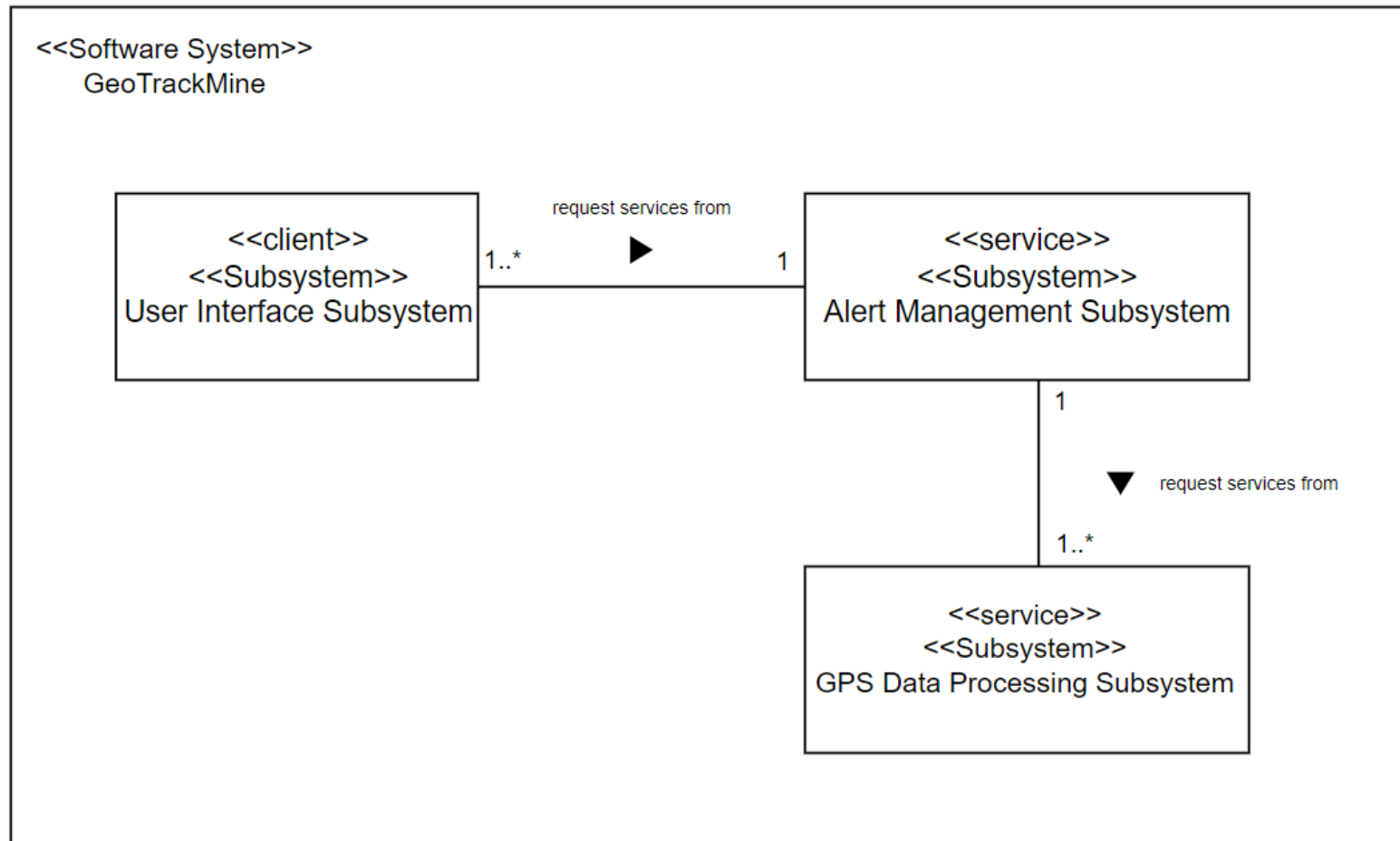


Figura 13: Diagrama de Subsistemas.

BPMN

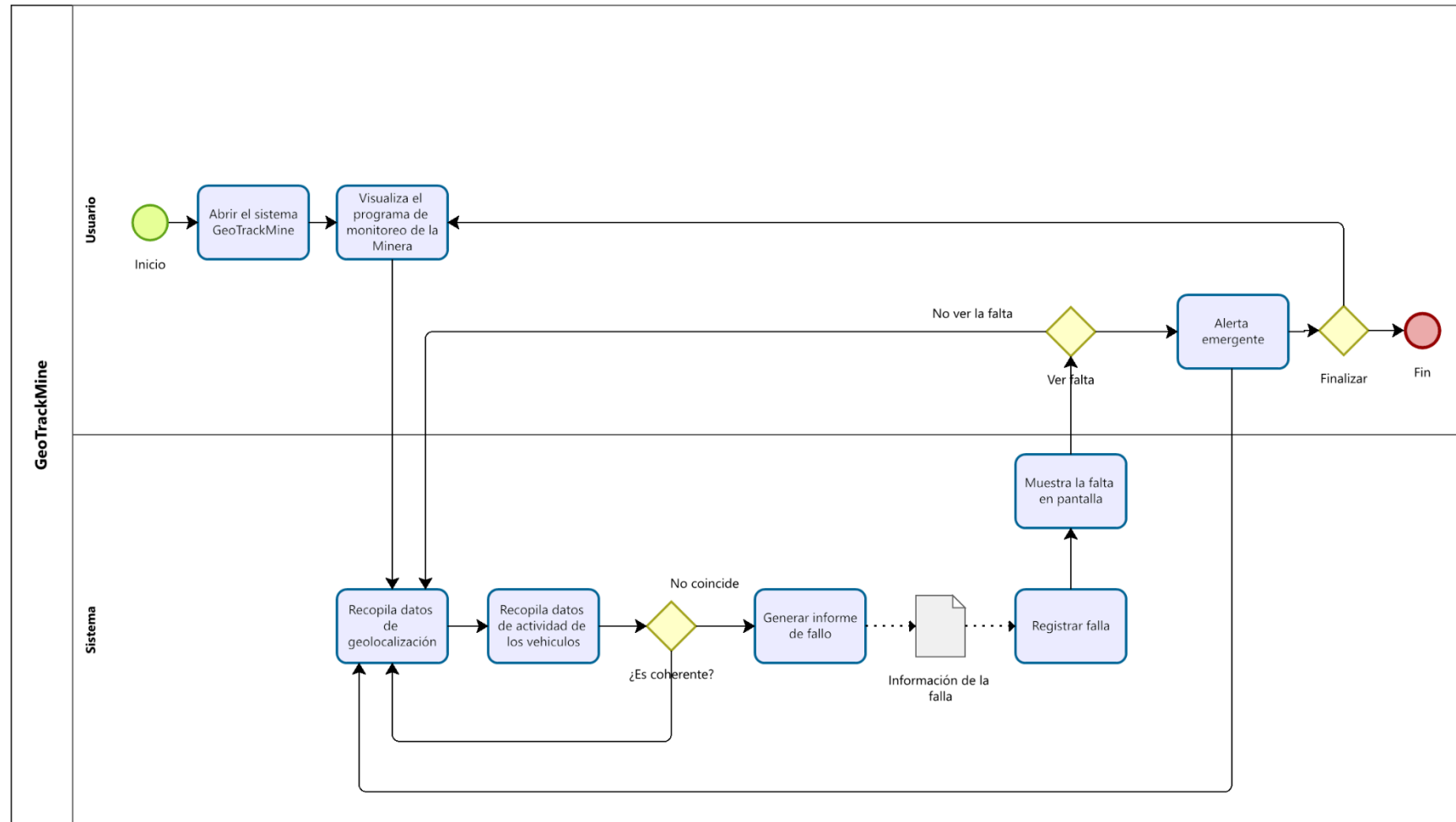


Figura 14: BPMN.

Diseño

Arquitectura del Sistema (Cliente-Servidor)

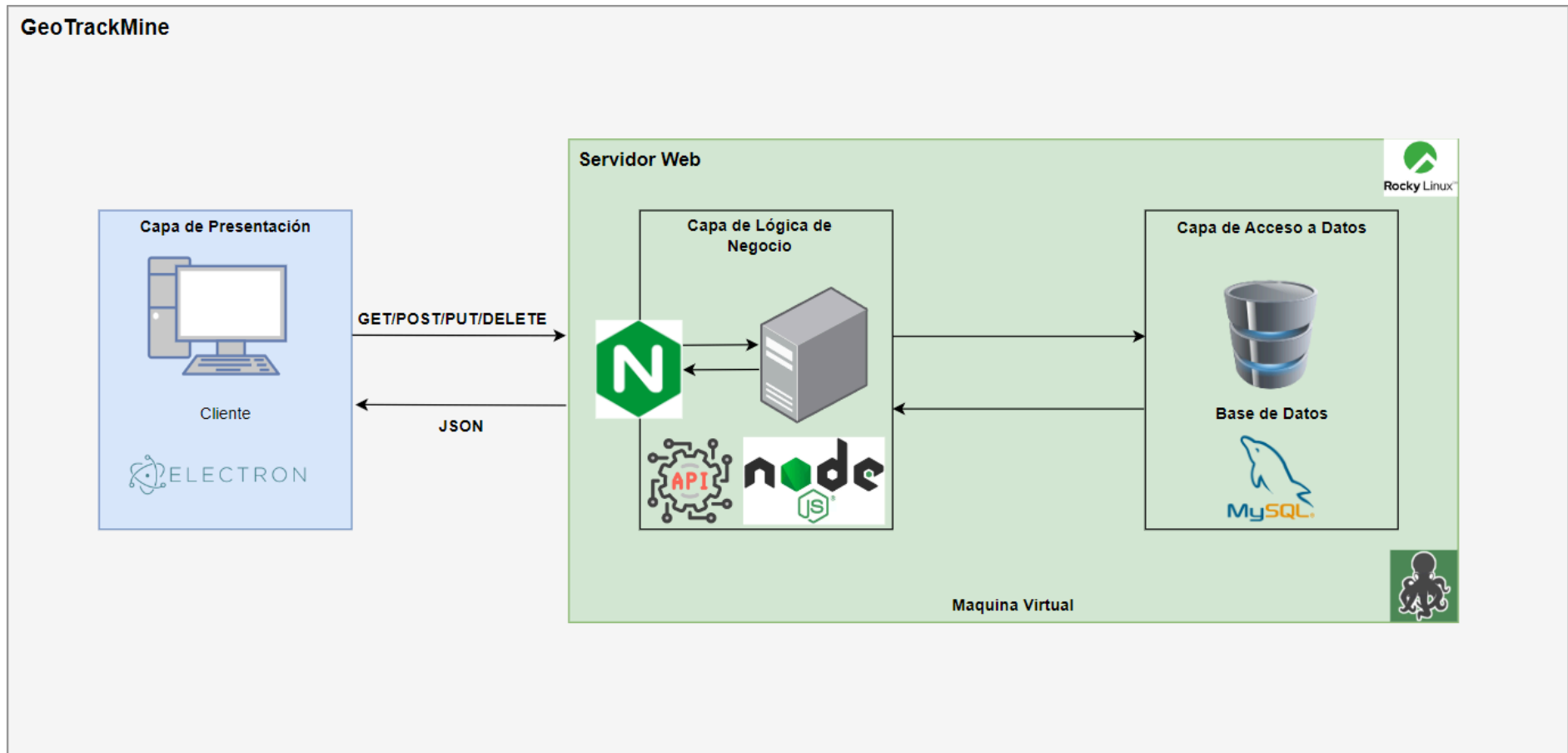


Figura 15: Arquitectura de GeoTrackMine.

Diseño de la Interfaz de Usuario

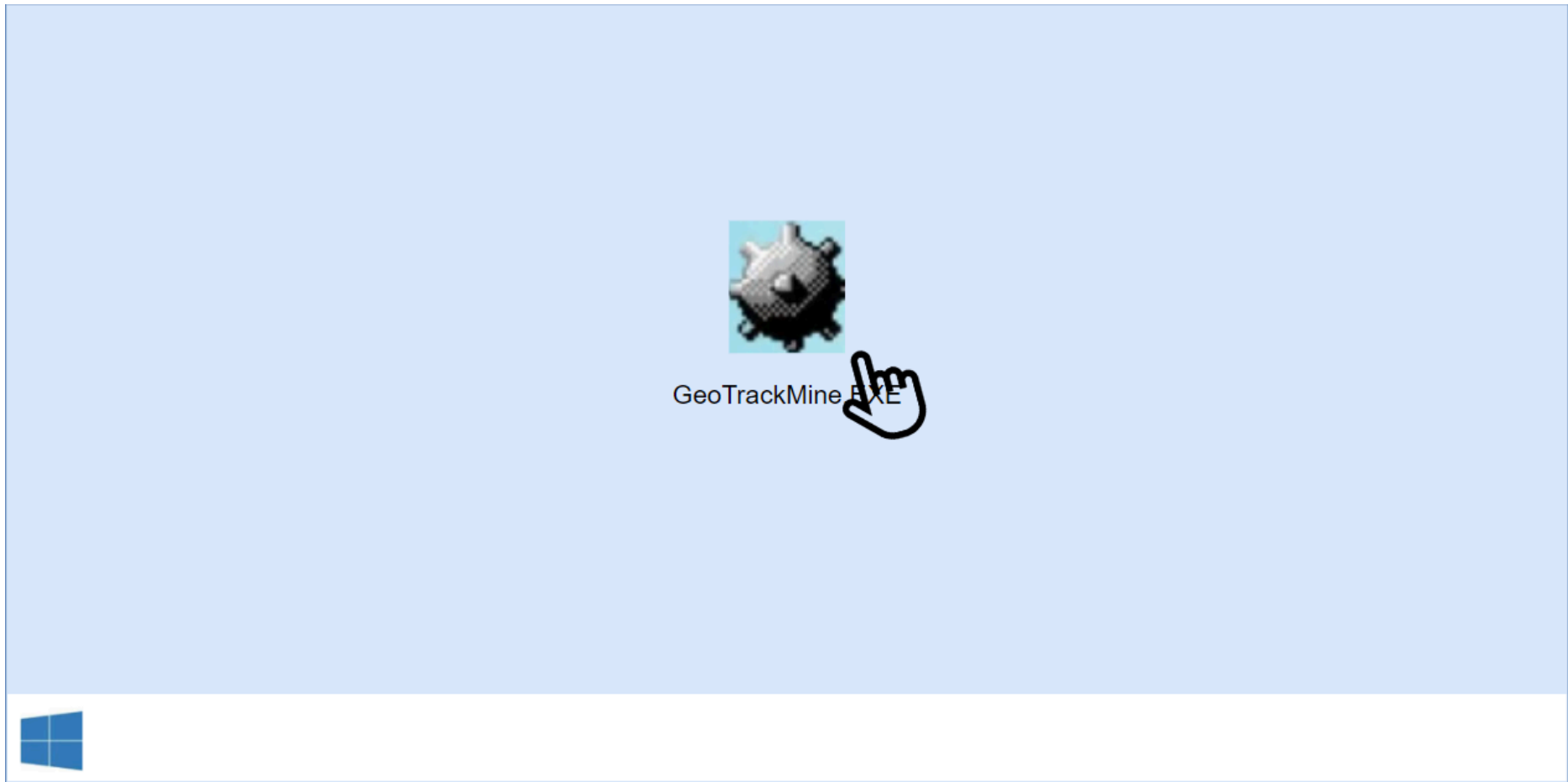


Figura 16: Ejecución de GeoTrackMine.



Figura 17: Interfaz de Inicio de Sesión.



Figura 18: Interfaz de Registro de Alertas.

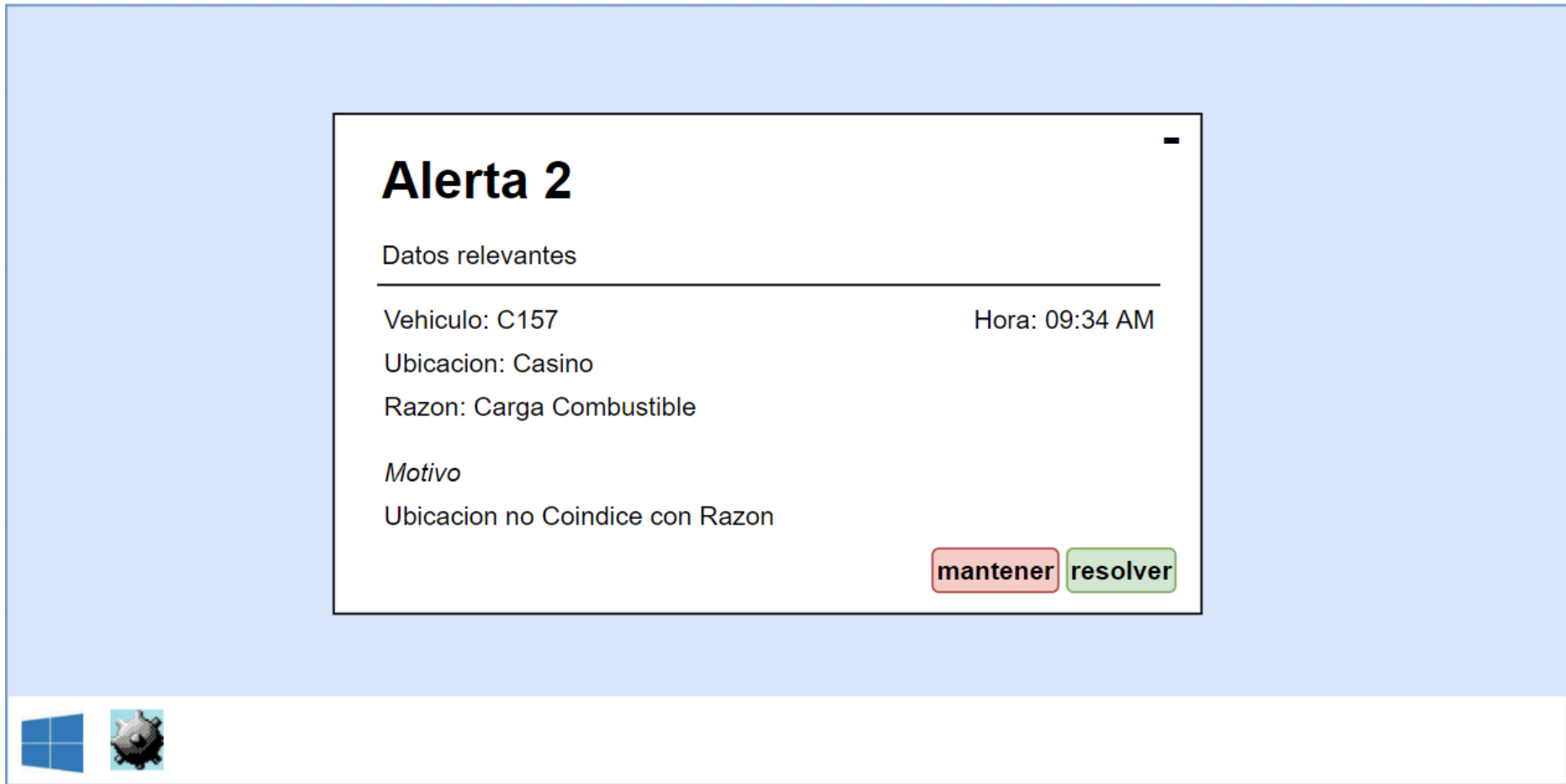


Figura 19: Interfaz de Detalle de Alerta.



Figura 20: Interfaz de Notificación de Alertas.

Diseño de la Base de Datos

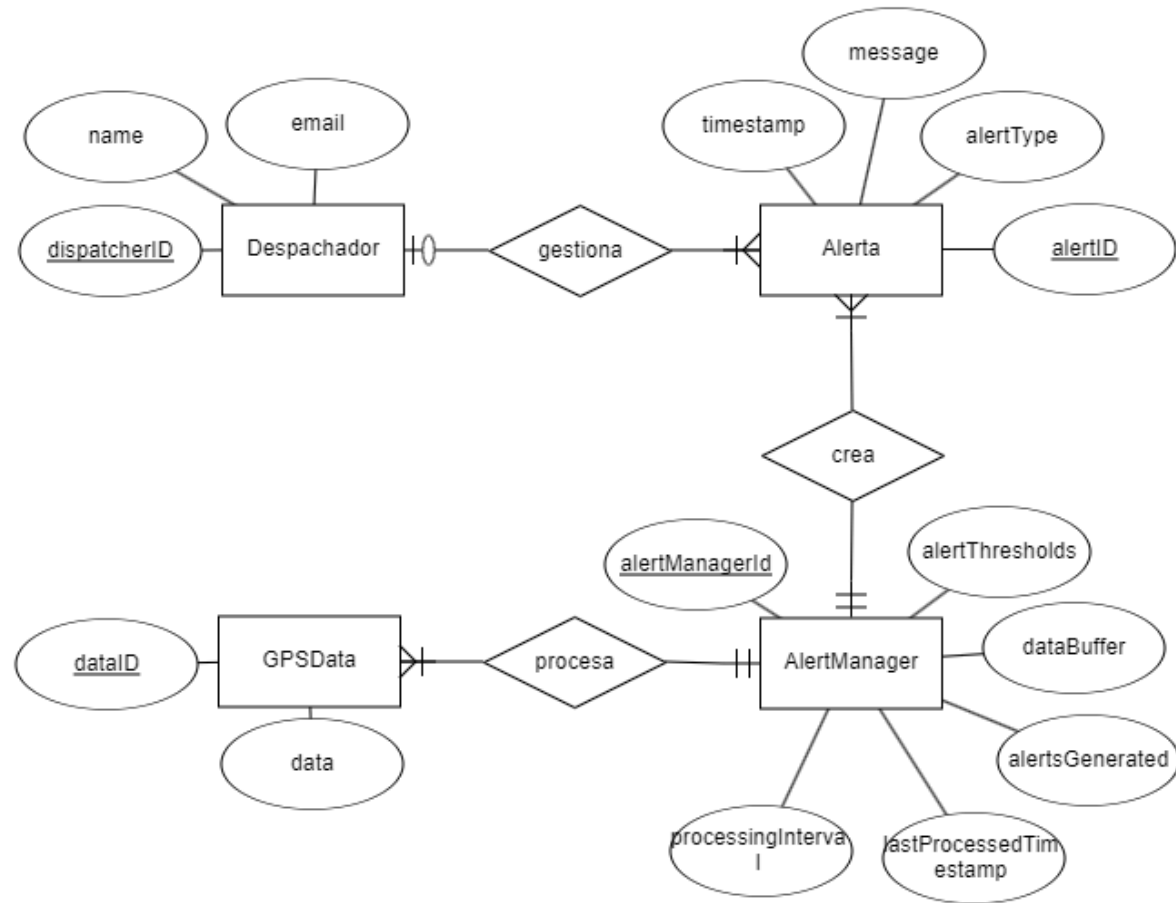


Figura 21: Modelo Entidad Relación.

Desarrollo

Configuración del Entorno de Desarrollo

Este entorno incluye una máquina virtual, herramientas de desarrollo, un servidor web y una base de datos, todo configurado para soportar la aplicación backend desarrollada con Node.js y Express.js.

Preparación de la Máquina Virtual

El primer paso fue crear una máquina virtual utilizando VirtualBox. Se optó por Rocky Linux Server 8.9 debido a su estabilidad y robustez, características esenciales para un entorno de desarrollo confiable. Se asignaron recursos adecuados a la máquina virtual, como CPU, memoria y almacenamiento, para asegurar un rendimiento óptimo durante el desarrollo y las pruebas. Una vez configurada la máquina virtual, se instaló Rocky Linux Server 8.9 y se configuró la red para permitir el acceso SSH, lo que facilita la administración remota.

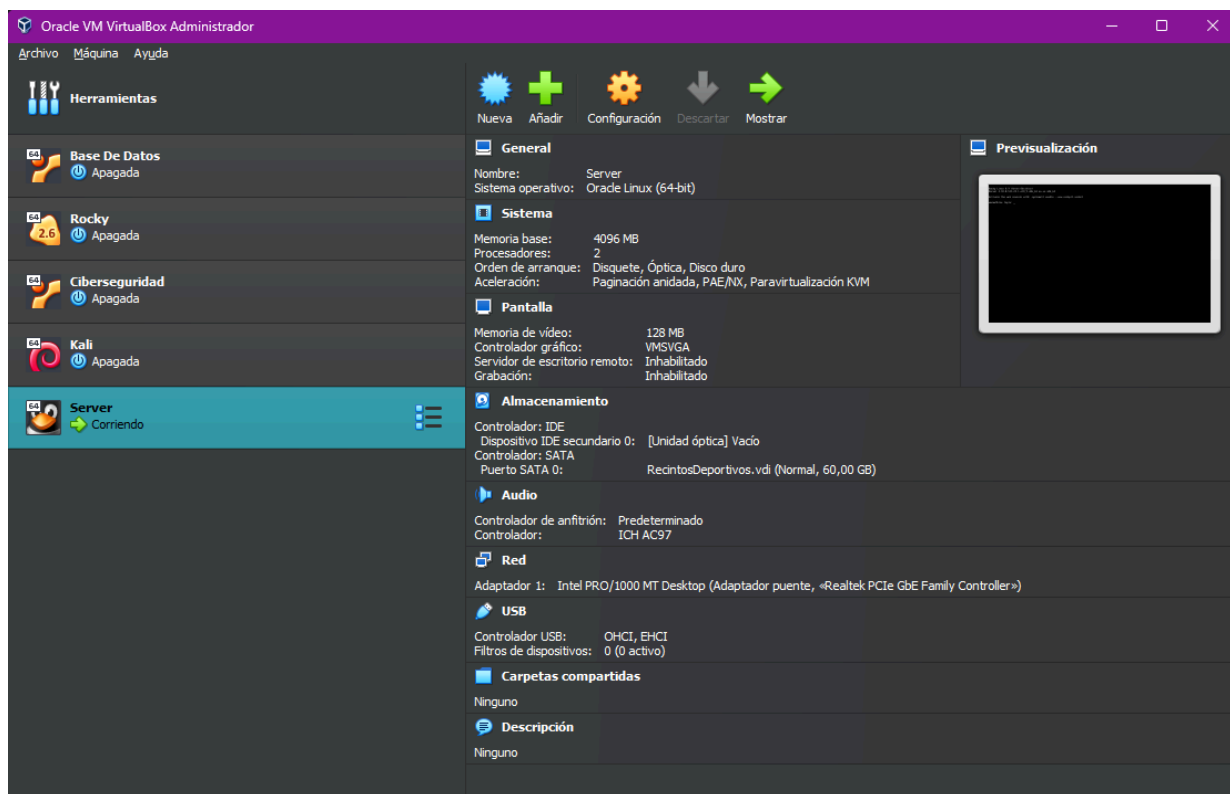


Figura 22: Configuración de la Máquina Virtual.

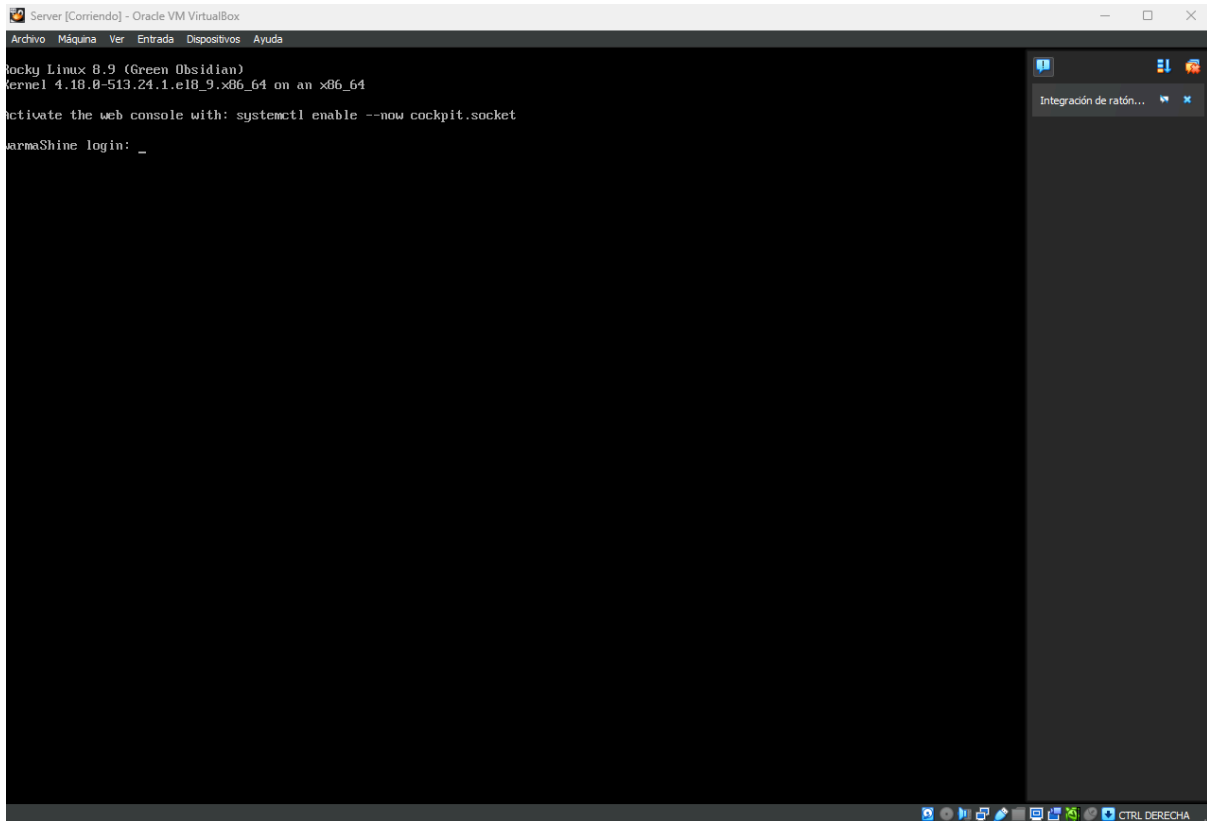


Figura 23: Máquina virtual en Virtualbox.

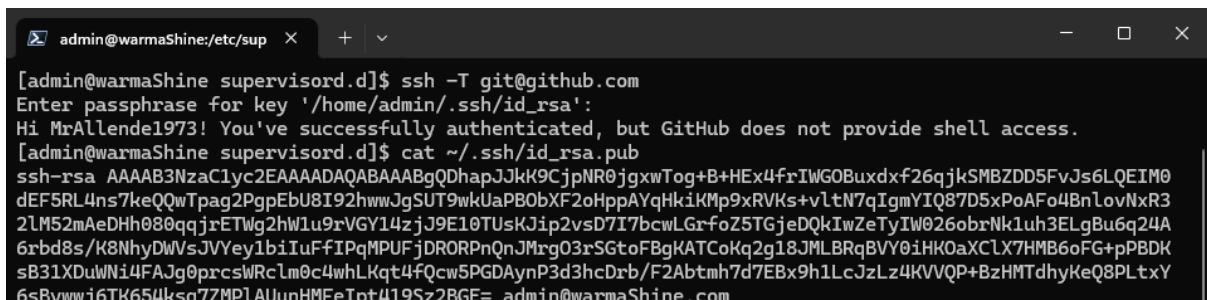


Figura 25: Conexión SSH y Github.

Configuración del Backend

El backend del proyecto GeoTrackMine se desarrolló utilizando Node.js y Express.js. La elección de estas tecnologías se basó en su capacidad para manejar aplicaciones de alta concurrencia y su eficiencia en el procesamiento de I/O. La primera tarea fue clonar el repositorio del proyecto desde GitHub utilizando SSH para asegurar la transferencia segura de datos. Una vez clonado el repositorio, se procedió a instalar Node.js y NPM (Node Package Manager) en la máquina virtual. Esto estableció la base para el desarrollo del backend.

```
[admin@warmaShine Proyectos]$ git clone git@github.com:MrAllende1973/Backend-GeoTrackMine.git
Clonando en 'Backend-GeoTrackMine'...
Enter passphrase for key '/home/admin/.ssh/id_rsa':
remote: Enumerating objects: 17, done.
remote: Counting objects: 100% (17/17), done.
remote: Compressing objects: 100% (14/14), done.
remote: Total 17 (delta 0), reused 17 (delta 0), pack-reused 0
Recibiendo objetos: 100% (17/17), 23.92 KiB | 11.96 MiB/s, listo.
```

Figura 26: Clonación de repositorio del Backend.

```
[admin@warmaShine Backend-GeoTrackMine]$ npm i

added 177 packages, and audited 178 packages in 3s

19 packages are looking for funding
  run `npm fund` for details

1 high severity vulnerability

To address all issues, run:
  npm audit fix

Run `npm audit` for details.
npm notice
npm notice New minor version of npm available! 10.5.0 -> 10.8.1
npm notice Changelog: https://github.com/npm/cli/releases/tag/v10.8.1
npm notice Run npm install -g npm@10.8.1 to update!
npm notice
[admin@warmaShine Backend-GeoTrackMine]$ npm audit fix

changed 1 package, and audited 178 packages in 1s

19 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
```

Figura 27: Instalación de Nodejs mediante NPM.

Configuración de la Base de Datos

La gestión de datos para GeoTrackMine se realiza mediante MySQL. Se instaló y configuró MySQL en la máquina virtual, y se creó una nueva base de datos para la aplicación. Posteriormente, se ejecutó un script SQL para crear todas las tablas necesarias según el modelo de datos definido. Esta base de datos soporta todas las operaciones CRUD necesarias para la gestión de usuarios, alertas y datos de GPS.

```
[admin@warmaShine db]$ mysql -u admin -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 10
Server version: 8.0.36 Source distribution

Copyright (c) 2000, 2024, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> CREATE DATABASE GeoTrackMine;
Query OK, 1 row affected (0,02 sec)

mysql> USE GeoTrackMine;
Database changed
mysql> exit
Bye
[admin@warmaShine db]$ |
```

Figura 28: Creación del Esquema GeoTrackMine.

```
[admin@warmaShine db]$ mysql -u admin -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 12
Server version: 8.0.36 Source distribution

Copyright (c) 2000, 2024, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> USE GeoTrackMine;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> SHOW TABLES;
+-----+
| Tables_in_GeoTrackMine |
+-----+
| AlertManager            |
| Alertas                 |
| Despachadores          |
| GPSData                 |
| HTTPLogs                |
| Logs                    |
+-----+
6 rows in set (0,00 sec)
```

Figura 29: Creación de las tablas del esquema GeoTrackMine.

Gestión del Servidor Web y Procesos

Para servir la aplicación y manejar los procesos del backend, se configuraron NGINX y Supervisor. NGINX se utilizó como servidor proxy inverso, gestionando las solicitudes entrantes y redirigiéndolas al backend de Node.js. La configuración de NGINX incluyó la creación de archivos específicos para el sitio de GeoTrackMine, asegurando una correcta redirección del tráfico.

```
[admin@warmaShine supervisor.d]$ sudo systemctl status nginx
● nginx.service - The nginx HTTP and reverse proxy server
   Loaded: loaded (/usr/lib/systemd/system/nginx.service; enabled; vendor preset: disabled)
   Active: active (running) since Thu 2024-06-06 02:01:27 EDT; 12min ago
     Process: 2338 ExecStart=/usr/sbin/nginx (code=exited, status=0/SUCCESS)
     Process: 2336 ExecStartPre=/usr/sbin/nginx -t (code=exited, status=0/SUCCESS)
     Process: 2334 ExecStartPre=/usr/bin/rm -f /run/nginx.pid (code=exited, status=0/SUCCESS)
   Main PID: 2339 (nginx)
     Tasks: 3 (Limit: 23130)
   Memory: 5.0M
   CGroup: /system.slice/nginx.service
           └─2339 nginx: master process /usr/sbin/nginx
             └─2340 nginx: worker process
               └─2341 nginx: worker process

jun 06 02:01:27 warmaShine.com systemd[1]: nginx.service: Succeeded.
jun 06 02:01:27 warmaShine.com systemd[1]: Stopped The nginx HTTP and reverse proxy server.
jun 06 02:01:27 warmaShine.com systemd[1]: Starting The nginx HTTP and reverse proxy server...
jun 06 02:01:27 warmaShine.com nginx[2336]: nginx: the configuration file /etc/nginx/nginx.conf syntax is correct
jun 06 02:01:27 warmaShine.com nginx[2336]: nginx: configuration file /etc/nginx/nginx.conf test is successful
jun 06 02:01:27 warmaShine.com systemd[1]: Started The nginx HTTP and reverse proxy server.
lines 1-20/20 (END)
```

Figura 30: Configuración NGINX.

Supervisor se configuró para gestionar el proceso de Node.js, asegurando que la aplicación se inicie automáticamente y se mantenga en ejecución. Esto es crucial para la estabilidad y disponibilidad de la aplicación, ya que Supervisor reinicia el proceso en caso de fallos, minimizando el tiempo de inactividad.

```
[admin@warmaShine Backend-GeoTrackMine]$ sudo supervisorctl status geotrackmine
geotrackmine RUNNING pid 2373, uptime 0:03:08
[admin@warmaShine Backend-GeoTrackMine]$ sudo supervisorctl tail -f geotrackmine
==> Press Ctrl-C to exit <==
Conexion exitosa a la base de datos
Servidor corriendo en puerto 3007
Conexion exitosa a la base de datos
Servidor corriendo en puerto 3007
Conexion exitosa a la base de datos
Servidor corriendo en puerto 3007
:::1 - - [06/Jun/2024:06:06:27 +0000] "GET / HTTP/1.1" 404 139 "-" "curl/7.61.1"
^C
[admin@warmaShine Backend-GeoTrackMine]$ sudo netstat -tuln | grep 3007
tcp6      0      0 :::3007          :::*              LISTEN
[admin@warmaShine Backend-GeoTrackMine]$ sudo ss -tuln | grep 3007
tcp       LISTEN 0      511              *:*3007          *:*
[admin@warmaShine Backend-GeoTrackMine]$ curl http://localhost:3007
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<title>Error</title>
</head>
<body>
<pre>Cannot GET /</pre>
</body>
</html>
[admin@warmaShine Backend-GeoTrackMine]$ sudo supervisorctl tail -f geotrackmine
==> Press Ctrl-C to exit <==
Conexion exitosa a la base de datos
Servidor corriendo en puerto 3007
Conexion exitosa a la base de datos
Servidor corriendo en puerto 3007
Conexion exitosa a la base de datos
Servidor corriendo en puerto 3007
:::1 - - [06/Jun/2024:06:06:27 +0000] "GET / HTTP/1.1" 404 139 "-" "curl/7.61.1"
:::1 - - [06/Jun/2024:06:09:52 +0000] "GET / HTTP/1.1" 404 139 "-" "curl/7.61.1"
```

Figura 31: Configuración de Supervisor.

Configuración del Frontend

El frontend del proyecto GeoTrackMine se desarrolló utilizando Electron.js, un framework que permite crear aplicaciones de escritorio utilizando tecnologías web como HTML, CSS y JavaScript. La elección de Electron.js se basó en su capacidad para proporcionar una experiencia de usuario nativa en múltiples plataformas y su facilidad para integrarse con el backend desarrollado en Node.js.

Además, se configuró un entorno de desarrollo en el PC del desarrollador para poder generar el ejecutable de la aplicación. Esto incluye la configuración de scripts de construcción en el archivo package.json y el uso de herramientas como electron-packager o electron-builder para empaquetar la aplicación en un ejecutable que pueda ser fácilmente instalado en el computador del despachador.

```

PS C:\Users\cbaut\OneDrive\SSD\Universidad\Semestres\2024_1_Semestre\Proyecto IV\GeoTrackMine\Frontend\geo-track-mine> ls
Directory: C:\Users\cbaut\OneDrive\SSD\Universidad\Semestres\2024_1_Semestre\Proyecto IV\GeoTrackMine\Frontend\geo-track-mine

Mode                LastWriteTime         Length Name
----                -
la----             04-06-2024 23:39             .vscode
la----             04-06-2024 23:39             build
la----             04-06-2024 23:42             node_modules
la----             04-06-2024 23:42             out
la----             04-06-2024 23:39             resources
la----             04-06-2024 23:39             src
la----             04-06-2024 23:39             154 .editorconfig
la----             04-06-2024 23:39             37 .eslintignore
la----             04-06-2024 23:39             319 .eslinttrc.cjs
la----             04-06-2024 23:39             44 .gitignore
la----             04-06-2024 23:39             171 .npmrc
la----             04-06-2024 23:39             71 .prettiignore
la----             04-06-2024 23:39             70 .prettierrc.yaml
la----             04-06-2024 23:39             101 dev-app-update.yml
la----             04-06-2024 23:39             1475 electron-builder.yml
la----             04-06-2024 23:39             440 electron.vite.config.ts
la----             04-06-2024 23:42             353021 package-lock.json
la----             04-06-2024 23:39             1935 package.json
la----             04-06-2024 23:39             179035 pnpm-lock.yaml
la----             04-06-2024 23:39             88 postcss.config.js
la----             04-06-2024 23:39             563 README.md
la----             04-06-2024 23:39             177 tailwind.config.js
la----             04-06-2024 23:39             111 tsconfig.json
la----             04-06-2024 23:39             238 tsconfig.node.json
la----             04-06-2024 23:39             399 tsconfig.web.json
    
```

Figura 32: Entorno de desarrollo de Electronjs en desarrollo.



Figura 33: Interfaz inicio de sesión creada con ElectronEs.



Figura 34: Interfaz de Registro de Alertas.

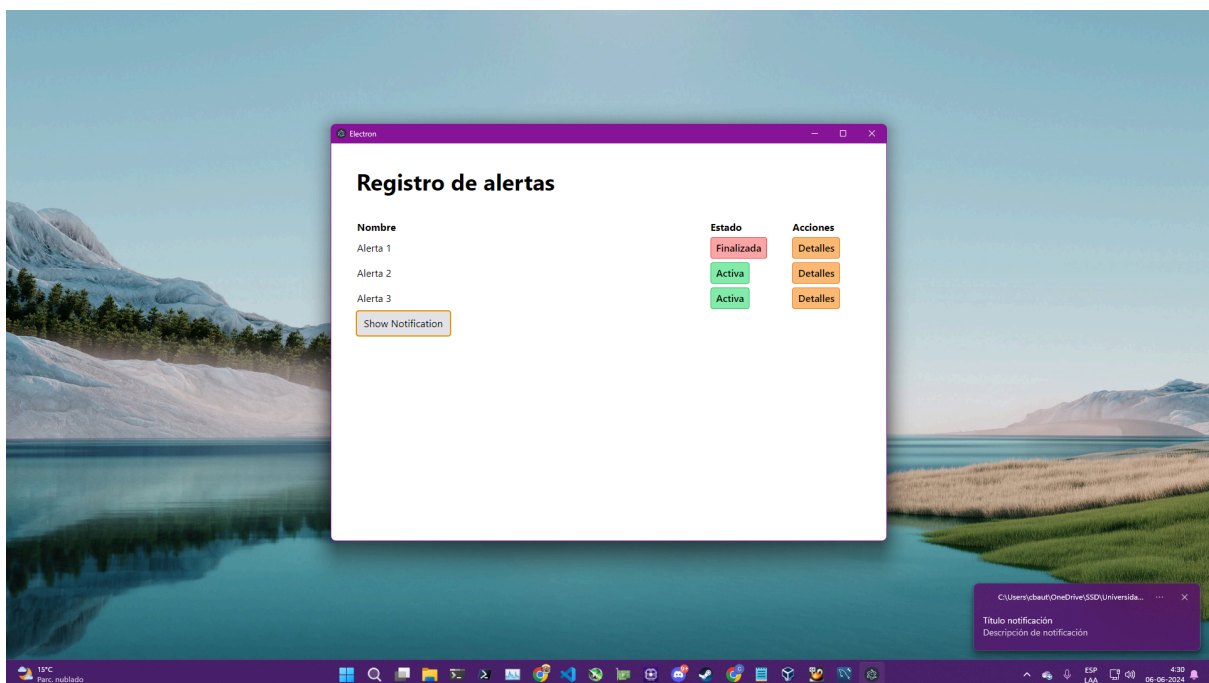


Figura 35: Interfaz de Visualización de notificaciones.

Desarrollo de la Aplicación

Desarrollo de la base de datos

```
1  -- MySQL Script for GeoTrackMine
2  -- Model: GeoTrackMine
3
4  • SET @OLD_UNIQUE_CHECKS=@UNIQUE_CHECKS, UNIQUE_CHECKS=0;
5  • SET @OLD_FOREIGN_KEY_CHECKS=@FOREIGN_KEY_CHECKS, FOREIGN_KEY_CHECKS=0;
6  • SET @OLD_SQL_MODE=@SQL_MODE, SQL_MODE='STRICT_TRANS_TABLES,NO_ZERO_IN_DATE,NO_ZERO_DATE,ERROR_FOR_DIVISION_BY_ZERO,NO_ENGINE_SUBSTITUTION';
7
8  -----
9  -- Schema GeoTrackMine
10 -----
11 • CREATE SCHEMA IF NOT EXISTS `GeoTrackMine` DEFAULT CHARACTER SET utf8;
12 • USE `GeoTrackMine`;
13
14 -----
15 -- Table `GeoTrackMine`.`Despachadores`
16 -----
17 • CREATE TABLE IF NOT EXISTS `GeoTrackMine`.`Despachadores` (
18   `dispatcherID` VARCHAR(255) PRIMARY KEY,
19   `name` VARCHAR(255) NOT NULL,
20   `email` VARCHAR(255) UNIQUE NOT NULL
21 ) ENGINE = InnoDB;
22
23 -----
24 -- Table `GeoTrackMine`.`Alertas`
25 -----
26 • CREATE TABLE IF NOT EXISTS `GeoTrackMine`.`Alertas` (
27   `alertID` VARCHAR(255) PRIMARY KEY,
28   `alertType` VARCHAR(255) NOT NULL,
29   `message` TEXT NOT NULL,
30   `timestamp` DATETIME NOT NULL,
31   `dispatcherID` VARCHAR(255),
32   FOREIGN KEY (`dispatcherID`) REFERENCES `GeoTrackMine`.`Despachadores` (`dispatcherID`) ON DELETE SET NULL
33 ) ENGINE = InnoDB;
```

Figura 36: Primera parte del script SQL.

```
35  -----
36  -- Table `GeoTrackMine`.`GPSData`
37  -----
38  ● ○ CREATE TABLE IF NOT EXISTS `GeoTrackMine`.`GPSData` (
39      `dataID` VARCHAR(255) PRIMARY KEY,
40      `data` JSON NOT NULL
41  ) ENGINE = InnoDB;
42
43  -----
44  -- Table `GeoTrackMine`.`AlertManager`
45  -----
46  ● ○ CREATE TABLE IF NOT EXISTS `GeoTrackMine`.`AlertManager` (
47      `managerID` INT AUTO_INCREMENT PRIMARY KEY,
48      `alertThresholds` JSON NOT NULL,
49      `dataBuffer` JSON NOT NULL,
50      `alertsGenerated` JSON NOT NULL,
51      `lastProcessedTimestamp` DATETIME NOT NULL,
52      `processingInterval` INT NOT NULL
53  ) ENGINE = InnoDB;
```

Figura 37: Segunda parte script SQL.

```
55 -----
56 -- Table `GeoTrackMine`.`HTTPLogs`
57 -----
58 ● CREATE TABLE IF NOT EXISTS `GeoTrackMine`.`HTTPLogs` (
59     `logId` INT AUTO_INCREMENT PRIMARY KEY,
60     `timestamp` TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
61     `requestMethod` VARCHAR(10),
62     `requestURL` VARCHAR(255),
63     `userAgent` VARCHAR(255),
64     `clientIP` VARCHAR(45) NULL,
65     `referer` VARCHAR(255) NULL,
66     `logLevel` VARCHAR(10) DEFAULT 'INFO',
67     `httpStatus` INT,
68     `responseTime` FLOAT,
69     `errorMessage` TEXT
70 ) ENGINE = InnoDB;
71
72 -----
73 -- Table `GeoTrackMine`.`Logs`
74 -----
75 ● CREATE TABLE IF NOT EXISTS `GeoTrackMine`.`Logs` (
76     `logId` INT AUTO_INCREMENT PRIMARY KEY,
77     `timestamp` TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
78     `level` VARCHAR(10), -- INFO, WARNING, ERROR, DEBUG
79     `message` TEXT,
80     `component` VARCHAR(255), -- Part of the system where the log occurs
81     `details` JSON -- JSON to store additional structured details
82 ) ENGINE = InnoDB;
83
84 ● SET SQL_MODE=@OLD_SQL_MODE;
85 ● SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS;
86 ● SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS;
```

Figura 38: Última parte del script SQL.

Desarrollo Backend

El desarrollo del backend de la aplicación GeoTrackMine se estructuró en torno a la creación de una API REST robusta y escalable utilizando Node.js y Express.js.

El archivo principal app.js es la columna vertebral de la aplicación backend de GeoTrackMine. Este archivo inicializa y configura el servidor Express.js, maneja la configuración de middleware, y asegura la conexión a la base de datos, además de gestionar el manejo de errores de manera efectiva.

```
1  import dotenv from 'dotenv';
2  dotenv.config();
3
4  // Importaciones de módulos necesarios
5  import express from 'express';
6  import { pool } from './config/db.js';
7  import morgan from 'morgan';
8  import routes from './routes/index.routes.js';
9  import cors from 'cors';
10 import { errorHandler } from './middleware/error.middleware.js';
11
12 const app = express();
13 const PORT = process.env.PORT || 3000;
14
15 /**
16  * Middleware Configuration
17  */
18 // Logging de solicitudes HTTP
19 app.use(morgan('combined'));
20
21 /**
22  * Configuración de CORS para controlar los dominios permitidos
23  */
24 app.use(cors());
25
26 // Middleware para analizar solicitudes con payloads JSON
27 app.use(express.json());
28
29 // Montaje de las rutas API
30 app.use('/api', routes);
31
32 // Middleware para manejo de errores
33 app.use(errorHandler);
```

Figura 39: Archivo Principal.

El archivo `index.routes.js` tiene como objetivo principal la carga dinámica y la gestión centralizada de todas las rutas de la API en la aplicación `GeoTrackMine`. Utiliza el módulo `express` para definir el enrutador, y módulos del sistema de archivos (`fs`) y del sistema de módulos de ES (`url`, `path`) para leer y registrar las rutas automáticamente.

```
index.routes.js M x
src > routes > index.routes.js > default
You, anteaer | 1 author (You)
1 // Importaciones necesarias
2 import { Router } from 'express';
3 import { readdirSync } from 'fs';
4 import { fileURLToPath } from 'url';
5 import { dirname } from 'path';
6 import { AppError } from '../utils/error.handle.js';
7 // Recreando la funcionalidad de __dirname para módulos ES
8 const __filename = fileURLToPath(import.meta.url);
9 const __dirname = dirname(__filename);
10 // Definición de constantes
11 const PATH_ROUTER = `.${__dirname}`;
12 const router = Router();
13 /**
14  * Limpia el nombre del archivo, eliminando la extensión ".routes.js"
15  * @param {string} fileName - Nombre del archivo a limpiar
16  * @returns {string} Nombre del archivo sin extensión
17  */
18 const cleanFileName = (fileName) => fileName.replace('.routes.js', '');
19 /**
20  * Intenta importar y montar las rutas de un archivo específico.
21  * @param {string} file - Nombre del archivo a importar
22  */
23 Codiumate: Options | Test this function
24 const loadRoutesFromFile = async (file) => {
25   const cleanName = cleanFileName(file);
26   if (cleanName !== 'index') {
27     try {
28       const module = await import(`./${cleanName}.routes.js`);
29       console.log(`Se estan cargando las rutas... /${cleanName}`);
30       router.use(`/${cleanName}`, module.default);
31     } catch (error) {
32       throw new AppError(`Error al cargar rutas de ${cleanName}: ${error.message}`, 500);
33     }
34   }
35 };
36 /**
37  * Lee dinámicamente los archivos de rutas en el directorio "src/routes"
38  * y registra las rutas en el router.
39  */
40 Codiumate: Options | Test this function
41 const loadRoutes = () => {
42   const files = readdirSync(PATH_ROUTER);
43   files.forEach(loadRoutesFromFile);
44 };
45 // Carga las rutas al iniciar el módulo
46 loadRoutes();
47 export default router; You, anteaer • Configuración del Entorno de Desarrollo Inicial
```

Figura 40: Funcionalidad de enrutador dinámico.

El archivo db.js se encarga de configurar la conexión a la base de datos MySQL utilizando el módulo mysql2/promise. Esta configuración es fundamental para permitir que la aplicación GeoTrackMine interactúe con la base de datos de manera eficiente y segura.

```
db.js M X
src > config > db.js > ...
You, anteayer | 1 author (You)
1 import dotenv from 'dotenv';
2 import { createPool } from 'mysql2/promise';
3 import { AppError } from '../utils/error.handle.js';
4
5 dotenv.config();
6
7 const dbConfig = {
8   host: process.env.DB_HOST,
9   user: process.env.DB_USER,
10  password: process.env.DB_PASSWORD,
11  port: parseInt(process.env.DB_PORT),
12  database: process.env.DB_NAME,
13  waitForConnections: true,
14  connectionLimit: 50, // Aumentado desde 10 a 50
15  queueLimit: 500, // Establecido en 500 para controlar la cola de espera
16 };
17
18 const pool = createPool(dbConfig);
19
20 pool.getConnection((err, connection) => {
21   if (err) {
22     let errorMsg = 'Error en la conexión a la base de datos';
23
24     if (process.env.NODE_ENV !== 'production') {
25       switch (err.code) {
26         case 'PROTOCOL_CONNECTION_LOST':
27           errorMsg = 'La conexión a la base de datos fue cerrada.';
28           break;
29         case 'ER_CON_COUNT_ERROR':
30           errorMsg = 'La base de datos tiene demasiadas conexiones.';
31           break;
32         case 'ECONNREFUSED':
33           errorMsg = 'La conexión a la base de datos fue rechazada.';
34           break;
35         default:
36           errorMsg = err.message;
37           break;
38       }
39     }
40     console.error(errorMsg);
41     throw new AppError(errorMsg, 500);
42   }
43   if (connection) connection.release();
44   return;
45 });
46
47 export { pool }; You, anteayer • Configuración del Entorno de Desarrollo Inicial
```

Figura 41: Conexión con la base de datos.

Desarrollo Frontend

La estructura del proyecto se organizó en varias carpetas para facilitar el mantenimiento y la escalabilidad, incluyendo configuraciones específicas para herramientas como ESLint, Prettier y TypeScript. La aplicación se diseñó para integrar servicios que permiten la comunicación con el backend desarrollado en Node.js, asegurando una interacción fluida con la API REST. Se utilizó Electron Builder para empaquetar la aplicación, permitiendo generar instaladores para diferentes sistemas operativos y facilitar su distribución e instalación en los computadores de los despachadores. Esta elección tecnológica y la estructura del proyecto garantizan una aplicación robusta, eficiente y mantenible.

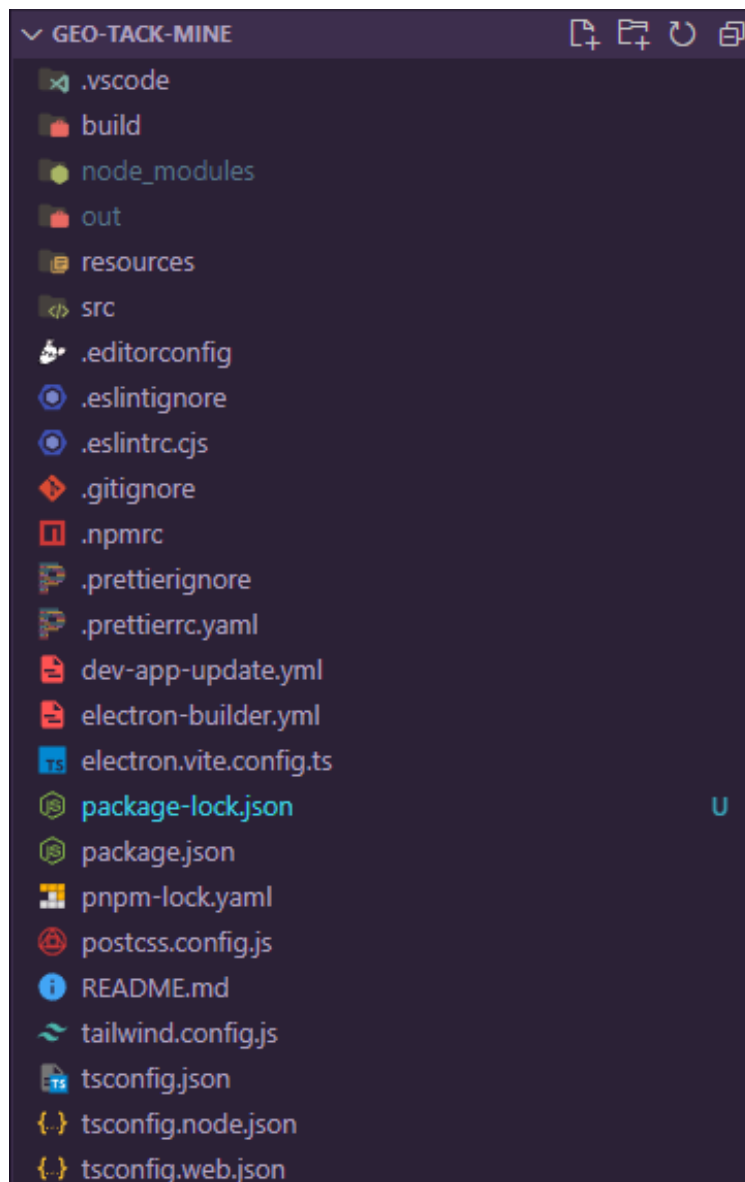


Figura 42: Estructura de carpetas de ElectronJs.

Conclusión

Dado el correcto establecimiento de las bases para albergar todas las funcionalidades que requiere el sistema, podemos decir que el proyecto GeoTrackMine va por buen camino. Hemos concretado una planificación bien estructurada y minuciosamente revisada, simulado pruebas de la lógica en los casos de uso y establecido Clases y Métodos para su correcto funcionamiento. Además, logramos diseñar una base de datos que recopilará todas las alertas encontradas y ayudará en el análisis de los datos necesarios para identificar dichas alertas. Incluso se documentó la arquitectura, diagramas y modelos para proporcionar un mejor entendimiento de este sistema. Como proyecto, tenemos plena confianza en que desarrollar las funcionalidades faltantes será un camino expedito, dado que hemos montado una base sólida y escalable con la cual finalizar y expandir este sistema llamado GeoTrackMine.

Referencias

- [1]. Cockburn, A. (2000). *Writing Effective Use Cases*. Addison-Wesley.
- [2]. Gooma, H. (2011). *Software Modeling and Design: UML, Use Cases, Patterns, and Software Architectures*. Cambridge University Press.

Algunas referencias de IoT, que den cuenta del uso de gps, y otros recursos nuevos que mencionan