

Diseño e Implementación de la Base de Datos para la Gestión de Información Crítica en Emergencias de Bomberos

Sofía Catalina Jara Álvarez

sofia.jaraa@usm.cl

Lais San Martín Navarro
Profesora Guía

Francis Fuentes Chacana
Profesora Correferente

Resumen: Se abordará la actual problemática presentada por Bomberos de Viña del Mar: la falta de gestión de información crítica como ubicación de sustancias peligrosas, grifos, salidas de emergencia, entre otros, durante emergencias de bomberos. Para mitigar esto, se propone "FastPlan", una plataforma web y móvil diseñada para la visualización de estos datos. En este contexto, este trabajo se orienta puntualmente al diseño e implementación de una base de datos centralizada, capaz de almacenar de manera organizada la información de los planes de emergencia de distintas comunidades dentro de la ciudad. Esta estructura cumple el rol principal de columna vertebral de FastPlan, siendo el componente que permite al sistema georreferenciar los elementos críticos para hacerlos visibles en un mapa interactivo. Para validar la base de datos se realizaron diversas pruebas de simulación de emergencias donde se logró exitosamente la visualización de información importante de manera inmediata. Los resultados positivos de este proyecto indican que el sistema puede apoyar de manera efectiva las labores de bomberos en terreno, además de optimizar la toma de decisiones durante una emergencia. Asimismo, esta solución puede ser de interés para empresas que, de igual forma, manejen información crítica que debe estar disponible en todo momento.

Palabras Clave: Base de datos, planes de emergencia, georreferenciación, información crítica



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: Diseño e Implementación de la Base de Datos para la Gestión de Información Crítica en Emergencias de Bomberos

Nombre del candidato(a): Sofía Catalina Jara Álvarez _____

Carrera / Grado: Ingeniería en Informática _____

Campus: Sede Viña del Mar Departamento: Departamento de Electrotecnia e Informática

2.- VALIDACIÓN DEL PROFESOR GUÍA/DIRECTOR DE TESIS

Yo, Lais San Martín Navarro, 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: 5-marzo-2026 _____ Firma: 

Estudiante o Candidato(a):

Fecha: 5 marzo 2026 _____ Firma: 

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 definición del problema

En el contexto de este trabajo, se define una emergencia como una situación de peligro que afecta de manera colectiva a una comunidad y que debe abordarse de manera inmediata y coordinada, ya que se pone en riesgo la seguridad y la vida de múltiples personas. [1] Algunos eventos considerados como emergencias a nivel de comunidad son, por ejemplo, incendios, derrame de sustancias tóxicas, fugas de gas, terremotos, entre otros.

Actualmente en Chile, una gran cantidad de edificios, condominios, empresas e instituciones enfrentan un gran desafío, ya que cuentan con documentos, llamados planes de emergencia, principalmente en un formato físico y de gran extensión, los cuales pueden ser inaccesibles o inutilizables durante una emergencia.

Un plan de emergencia incluye datos que son clave para manejar estas situaciones y optimizar la utilización de los recursos según un procedimiento específico para cada siniestro [2]. En términos generales, dicho documento establece qué hacer en el antes, el durante y el después de una emergencia. Según la Ley N°21.442 de Copropiedad Inmobiliaria, cada condominio debe contar con un plan de emergencia actualizado y elaborado por un ingeniero en prevención de riesgos. Es obligación del administrador o del comité de administración implementar en cada condominio o edificio este documento, para posteriormente ser entregado a Carabineros de Chile y al Cuerpo de Bomberos [3]. La falta de un sistema eficiente que maneje esta información de forma accesible y actualizada provoca dificultades en el tiempo de actuación efectiva, toma de decisiones, falta de coordinación y preparación por parte de los equipos especializados de emergencias, en este caso, bomberos. El personal de bomberos se encarga de proteger a las personas que se encuentren en una situación de peligro, por lo que es esencial que tengan acceso inmediato a los datos necesarios para atender una emergencia y así no retrasar su labor en terreno.

1.2 Propuesta de solución

Como solución a esta problemática que fue presentada por Bomberos de Viña del Mar, se propone "FastPlan" una plataforma web y móvil que permite digitalizar, centralizar y facilitar el acceso a los planes de emergencia de edificios y que está dirigido para ser utilizado por parte de equipos de primera respuesta, como Cuerpos de Bomberos. FastPlan cuenta con un equipo de trabajo colaborativo donde cada integrante aborda diferentes aspectos de su desarrollo, tales como *backend* y *frontend* (web y móvil), el diseño de la base de datos, el diseño de interfaz, entre otros. El sistema busca brindar información crítica (como salidas de emergencias, grifos, ubicación de sustancias peligrosas y accesos especiales) tanto de la edificación que tenga una emergencia en curso como de los recintos que se encuentren en un radio cercano, de manera que esta información se vea de manera resumida y entendible mediante el uso de iconografía y un mapa interactivo para así apoyar la labor de bomberos durante emergencias. En sí, el proyecto busca mejorar la coordinación en emergencias, optimizar la toma de decisiones en terreno y fomentar la preparación ciudadana mediante el uso de tecnología y datos centralizados.



1.3 Objetivo general

En el caso específico de este trabajo, el objetivo es diseñar e implementar una base de datos destinada principalmente al almacenamiento y gestión de información crítica y datos geoespaciales asociados a planes de emergencia (como ubicación de salidas de emergencia, grifos, sustancias peligrosas y accesos especiales) de distintos tipos de edificaciones de comunidades, abordando la modelación, normalización, integridad y disponibilidad de la información, asegurando un acceso confiable e inmediato para enfrentar emergencias como incendios, fugas de sustancias peligrosas o desastres naturales. Esta base de datos se diseñará de manera que permita que los elementos críticos sean georreferenciados, permitiendo su visualización precisa dentro de un mapa interactivo.

1.4 Objetivos específicos

Se establecieron los siguientes objetivos específicos:

- Analizar la información de planes de emergencia con el fin de estructurar los datos sobre los elementos críticos.
- Diseñar el modelo conceptual, lógico y físico de la base de datos según los requerimientos del problema.
- Implementar la base de datos en un sistema moderno y adecuado que permita el almacenamiento organizado de los datos de los planes de emergencia.
- Implementar mecanismo de sincronización y transformación de datos que aseguren la disponibilidad de los datos críticos.
- Realizar pruebas de rendimiento para validar el correcto funcionamiento del sistema.

1.5 Metodología

En este caso se utiliza la metodología de trabajo SCRUM, basada en un enfoque ágil que fomenta la colaboración en equipo, gestión de tareas y productividad [4]. De esta forma se obtiene el mejor resultado del proyecto en el menor tiempo posible. En esta metodología se utilizan los llamados *sprints* o iteraciones, que son sesiones determinadas con un tiempo de trabajo específico, en el que al finalizar se debe entregar los avances esperados del desarrollo. A continuación, se detallan los *sprints* planificados para el desarrollo del proyecto:

- **Sprint 1:** Diseño e implementación de base de datos, desarrollo de aplicación web y móvil logrando comunicación entre ambos sistemas.
- **Sprint 2:** Integración de mapa con georreferenciación de ubicación de las emergencias y enfoque en diseño y visualización de información crítica.
- **Sprint 3:** Georreferenciación de elementos críticos dentro del mapa implementado e información pública sobre estado de planes de emergencia.



1.5.1 Plan de trabajo

Para el desarrollo de la base de datos se definen las fases a continuación:

Análisis e investigación (*Product Backlog*): Esta primera fase se centra en levantar los requerimientos del sistema en cuestión junto a Bomberos de Viña del Mar. Además, se realizará un análisis exhaustivo de diferentes planes de emergencia de condominios (los cuales serán la fuente principal de datos) de forma que se obtengan los tipos de datos y sus relaciones para que la plataforma cumpla con los requerimientos de Bomberos y los requerimientos detectados por el equipo de FastPlan.

Diseño de la base de datos: Comenzar el modelado de la base de datos relacional y no relacional según la información obtenida de la fase de análisis e investigación con la finalidad de diseñar una estructura de datos que soporte los requerimientos. Esta fase se dividirá en tres:

- **Modelado conceptual:** Identificar las entidades claves del sistema, sus relaciones y su cardinalidad, así como las restricciones relevantes para mantener la integridad de los datos. Estos son parte de los requisitos principales del proyecto, permitiendo la organización del desarrollo y ofreciendo una vista general de lo que contendrá el sistema.
- **Modelado lógico:** Considerando al modelo relacional como modelo lógico, en esta etapa se representan las entidades en base a tablas, conformadas por filas y columnas que almacenan los datos. Cada columna representa un atributo, el cual debe quedar con su nombre y tipo de dato definido. Por otro lado, cada fila tiene su clave primaria (PK; puede ser simple o compuesta) y las claves foráneas (FK) quedan definidas. Estas últimas permiten las relaciones entre las entidades. También se establecen reglas de integridad tales como valores únicos o valores que no pueden ser nulos. Se emplea el método de Normalización para organizar adecuadamente los datos, mediante el análisis de dependencias.

Adicionalmente, se define el modelado de datos lógico de la componente no relacional encargada de la sincronización móvil. En esta parte, se estructuran los datos en un formato orientado a documentos JSON, desnormalizando el modelo relacional y usando la técnica de anidamiento.

- **Modelado físico:** Se considera el SGBD a utilizar y se toma el modelo lógico y se adecúa a este sistema. Se considera el volumen de datos, distribución geográfica, ajustes de rendimiento y optimización, patrones de usabilidad, etc.

Implementación (Desarrollo incremental): Creación de las tablas según el modelado realizado, dentro de un SGBD, en este caso MySQL. Se insertarán datos simulados para posteriormente realizar las pruebas correspondientes. En esta fase también se incluye la integración del esquema no relacional diseñado para Cloud Firestore, una base de datos no relacional en la nube, que permite la sincronización y envío de datos en tiempo real sin interrupción alguna, para así lograr la comunicación eficiente entre las plataformas web y móvil. Se seleccionó por un lado MySQL para garantizar la integridad de los datos y poder realizar consultas SQL más complejas. Por su parte, la implementación de Firestore garantiza la accesibilidad a los datos en todo



momento, además de la sincronización en tiempo real sin colapsar, siendo un requisito indispensable para mantener operativo el sistema en emergencias a gran escala.

Pruebas: Con los datos simulados, realizar pruebas para verificar la comunicación entre la plataforma web y la plataforma móvil, tomando el tiempo de envío de datos (latencia) entre ambas asegurando que no haya errores en la integridad, sincronización, consistencia y disponibilidad inmediata de estos. Además, se validará la funcionalidad de la georreferenciación de la emergencia y de los elementos críticos; estos deben tener la ubicación, ya sea exacta o aproximada, y ser asignados correctamente al carro respectivo que se seleccione desde la central de bomberos.

1.6 Estructura del informe

Capítulo 1: Introducción. Visión general de la propuesta de solución, el problema, los objetivos, metodología y plan de trabajo llevado a cabo.

Capítulo 2: Marco teórico. Base teórica conceptual y tecnológica que sustenta el proyecto y soluciones similares existentes.

Capítulo 3: Desarrollo del sistema. Se detallan los requerimientos funcionales y no funcionales, la arquitectura del sistema y el diseño de la base de datos relacional y no relacional.

Capítulo 4: Implementación. Describe la estrategia de despliegue e implementación de la base de datos, el proceso de poblamiento de datos y el proceso de extracción, transformación y carga de los datos en las bases de datos seleccionadas.

Capítulo 5: Pruebas. Pruebas de rendimiento que validan la eficiencia de la solución propuesta.

Capítulo 6: Conclusiones. Conclusiones derivadas del proyecto, cumplimiento de los objetivos, impacto, proyección y mejoras a futuro.

Capítulo 7: Agradecimientos.

Capítulo 8: Referencias.

2 Marco teórico

En este capítulo se abordan los fundamentos conceptuales y tecnológicos que respaldan el trabajo realizado.

2.1 Bases de datos

Las bases de datos son sistemas sólidos diseñados para almacenar y gestionar grandes volúmenes de diferentes tipos de datos. Su arquitectura puede ser tanto centralizada como distribuida. Permiten recopilar datos relevantes permitiendo acceso a información precisa y actualizada que, a diferencia de una hoja de cálculo, que organiza los datos en filas y columnas dentro de un único archivo (y por lo general se confunde de forma errónea con una base de datos), están hechas para una gran cantidad de datos y permiten realizar consultas mucho más complejas [5].

Existen diferentes tipos de base de datos, los que se mencionan a continuación son las más comunes actualmente:

- **Bases de datos relacionales (RDBMS):** las bases de datos relacionales almacenan los datos en formato de tablas, estando cada tabla conformada en base a filas y columnas. Estas constan de diversas tablas que representan entidades y reducen la redundancia de los datos. La estructuración mencionada es muy adecuada para gestionar grandes volúmenes de datos que requieran un acceso eficaz y flexible. Además, entre estas tablas puede haber una interconexión mediante claves primarias y claves foráneas, lo que permite obtener datos combinados de múltiples tablas realizando consultas SQL [6].
- **Bases de datos no relacionales o bases de datos NoSQL (Not only SQL):** A comparación del modelo relacional, este modelo no utiliza el formato tabular tradicional, es mucho más flexible y escalable rápidamente ya que es ideal para manejar datos semiestructurados y no estructurados, permitiendo adaptarse a diferentes requerimientos, además de tener un mayor rendimiento al momento de realizar consultas y alta disponibilidad sin interrupción [7].

Para FastPlan se opta por una arquitectura híbrida donde se utilizará una base de datos relacional, que permite mantener la integridad de los datos y, por otro lado, una base de datos no relacional que otorga alta disponibilidad y la velocidad necesaria para un actuar rápido de parte de bomberos y así tener los datos a disposición en situaciones de emergencia, incluso antes de llegar a la zona del siniestro.

2.2 Contenedores en Docker

Docker es un software que permite crear, probar e implementar aplicaciones en unidades estandarizadas llamadas contenedores, que empaquetan todo lo que necesita el software que va a ejecutar [8]. La ventaja de la utilización de Docker es poder transferir estas aplicaciones de manera sencilla y garantizando su consistencia, desde un entorno de desarrollo a uno de producción.



2.3 Sistemas de Información Geográfica (SIG)

Los SIG son sistemas informáticos desde donde se puede analizar, manipular y visualizar datos geoespaciales, el cual funciona con un mapa que tiene integradas distintas capas de datos. Estos sistemas son comúnmente utilizados en municipios y entidades gubernamentales para poder realizar evaluaciones de impacto ambiental y realizar planificación urbana [9].

2.3.1 Datos geoespaciales

Los datos geoespaciales se refieren a cualquier información que tenga una ubicación. Estos combinan datos de atributos (características descriptivas), datos temporales (tiempo o duración de vida de la ubicación) y datos de ubicación que pueden ser en formato vectorial como coordenadas o datos de trama (cuadrículas con información espacial) [9].

2.4 Emergencia en el contexto del sistema

Una emergencia es un suceso o situación que se presenta de manera imprevista y requiere una intervención operativa de bomberos inmediata (tales como incendios, terremotos o fugas de gas) debido a que puede poner en riesgo a la vida humana y sus bienes, generar daños al medioambiente, entre otros. Para efectos de este proyecto, el término se enfoca en "emergencias comunitarias", entendidas como sucesos que afectan de manera colectiva a una comunidad, como por ejemplo a los residentes de un condominio, a trabajadores de una empresa, alumnos y profesores de un colegio o universidad, entre otros.

2.5 Planes de emergencia en Chile

Los planes de emergencia son documentos que especifican información relevante sobre procedimientos de actuación durante una posible emergencia, para que familias y comunidades puedan estar preparadas de forma ordenada e informada. Una comunidad es el conjunto de personas de un barrio, villa, pueblo, ciudad, región o país. Las comunidades que se tendrán en cuenta para este proyecto son las que se encuentren específicamente en edificios o condominios.

Según la Ley N.º 21.442 de Copropiedad Inmobiliaria, se exige que cada condominio cuente con un plan de emergencia actualizado por el comité de administración y elaborado por un ingeniero en prevención de riesgos. La ley indica lo siguiente: "*Todo condominio deberá tener un plan de emergencia ante siniestros o emergencias, tales como incendios, terremotos, tsunamis u otros eventos que puedan dañar a las personas, a las unidades y/o a los bienes de dominio común del condominio. El plan de emergencia deberá incluir las acciones a tomar antes, durante y después del siniestro o emergencia, con especial énfasis en la alerta temprana y los procedimientos de evacuación ante incendios.*" [3]. Además, se menciona que cada plan de emergencia debe ser entregado en formato material y digital a la respectiva unidad de Carabineros de Chile y del Cuerpo de Bomberos que corresponda a la comuna donde se emplaza el condominio.

2.5.1 Contenido de un plan de emergencia

Cada edificio debe ser analizado tanto de forma externa como interna; de esta forma, se podrán obtener los elementos que se deben incluir en el plan de emergencia que puedan ser posibles causantes de riesgos adicionales. En el entorno se pueden encontrar postes eléctricos, depósitos de materiales peligrosos u otros elementos de riesgo, estos podrían inflamarse o caer sobre las personas. También debe evaluarse el inventario de los recursos de seguridad que se encuentran disponibles en el edificio o condominio; estos pueden ser botiquines de primeros auxilios, extintores, red seca, red húmeda, etc. Por último, deben incluirse las zonas seguras tanto internas como externas de la edificación, junto con vías de evacuación señalizadas y un plano con toda esta información recopilada.

2.6 Estado del arte

Para el desarrollo de este proyecto, se realizó una revisión de algunos sistemas existentes que actualmente son utilizadas por Bomberos durante emergencias. Si bien estas herramientas otorgan información clave para apoyar a los equipos especializados, se detectaron algunas faltas en cuanto a la disponibilidad inmediata de la información. A continuación, se describen los sistemas analizados y sus diferencias con FastPlan:

- **Smert:** Es una aplicación diseñada para visualizar información de edificios (anteriormente conocida como LobbyControl) de manera inmediata. Su funcionamiento se basa en la instalación de placas con un código QR en cada uno de los edificios o condominios donde habitan personas. Cuando hay una emergencia, bomberos debe arribar al lugar y escanear este QR para obtener los detalles de la edificación [10].

Este sistema tiene limitaciones, ya que bomberos está condicionado a primero llegar al lugar de la emergencia para poder ingresar a visualizar la información. Con FastPlan se puede tener acceso a estos datos desde el momento que se da aviso del despacho a una emergencia; de esta forma bomberos pueden tener una noción de dónde se encuentran los elementos críticos y qué decisiones tomar desde antes de llegar al siniestro.

- **ViperGO:** Es una solución diseñada para estar en las tabletas de los carros de bomberos. Esta aplicación permite visualizar en tiempo real la ubicación de las otras unidades, el lugar de la emergencia y ubicación de grifos cercanos durante el trayecto. Su enfoque principal es el despacho y coordinación de los vehículos de bomberos, no profundiza en los elementos críticos de las edificaciones de la comunidad [11].
- **Waze:** Es una aplicación enfocada principalmente a la navegación, donde la comunidad contribuye colaborativamente a proporcionar información de tráfico y rutas. Por lo general es utilizada por bomberos para calcular la ruta más rápida y evitar congestiones vehiculares, ayudando así a ahorrar tiempo y salvar vidas [12].
- **ArcGis:** Es la plataforma líder a nivel mundial de los SIG. Permite gestionar capas de información con una gran precisión; sin embargo, esta tiene un costo de licencia elevado, requiere un hardware de alto rendimiento y no está diseñado para la respuesta inmediata ante emergencias [13].

3 Desarrollo del sistema

A continuación, se aborda el diseño de la base de datos centrándose en sus requerimientos y en sus fases del modelado: conceptual, lógico y físico. Esta etapa constituye el pilar fundamental para el funcionamiento de la solución propuesta.

3.1 Análisis de requerimientos

3.1.1 Requerimientos Funcionales

Los requerimientos funcionales describen las funciones específicas que debe ejecutar el sistema para que cumpla con el comportamiento que se espera por el usuario final. A continuación, se presentan los requerimientos funcionales que se definieron para el desarrollo de la base de datos.

Tabla 1. Requerimientos funcionales

ID de requerimiento	Requerimiento	Descripción
RF-1	Digitalización de planes de emergencia	Permite el ingreso, georreferenciación y almacenamientos de información crítica de planes de emergencia.
RF-2	Autenticación de usuarios	Bomberos debe ingresar a las plataformas utilizando sus credenciales de Microsoft.
RF-3	Georreferenciación de Planes de Emergencia y sus elementos críticos	Cada plan de emergencia y sus elementos tienen coordenadas que permiten ver su ubicación exacta.
RF-4	Envío de información desde la central	Desde la plataforma web destinada a la central, se envían los datos de los planes hacia los dispositivos en las unidades de bomberos.
RF-5	Visualización de datos de los elementos críticos en las unidades de bomberos	En un mapa interactivo se ven los elementos críticos y sus datos asociados cuando hay una emergencia en curso.
RF-6	Ver planes de emergencia cercanos	Además de visualizar el plan de emergencia de la emergencia en sí, se pueden ver los planes de emergencia cercanos al siniestro según un radio determinado en caso de que la emergencia se expanda.

RF-7	Registro de unidades de bomberos	Se pueden incorporar y eliminar unidades de bomberos desde la central.
------	----------------------------------	--

3.1.2 Requerimientos No Funcionales

Los requerimientos no funcionales definen el cómo debe comportarse el sistema, según aspectos como el rendimiento, seguridad, disponibilidad, entre otros. A continuación, se presentan los requerimientos no funcionales que se definieron para este trabajo.

Tabla 2. Requerimientos no funcionales

ID de requerimiento	Requerimiento	Descripción
RNF-1	Envío de información en tiempo real	Los datos de los planes de emergencia deben ser enviados en menos de 1 segundo (1000 ms) desde la central.
RNF-2	Seguridad de autenticación y acceso	Los usuarios tienen roles específicos que definen su nivel de autorización en las plataformas, además deben poder usar sus credenciales de Microsoft.

3.2 Arquitectura del sistema

La arquitectura híbrida seleccionada, en este caso la utilización de MySQL con Cloud Firestore, permite cumplir con los requerimientos funcionales y no funcionales mencionados, aprovechando las ventajas del uso conjunto de ambos modelos. Para los planes de emergencia y sus elementos críticos es esencial que se mantenga su integridad, por lo que es necesario un modelo de base de datos relacional que sea implementado en un gestor de bases de datos con alta trayectoria como lo es MySQL (el gestor que se utilizó en este proyecto) debido a su alto rendimiento, facilidad de uso y soporte [14].

MySQL fue desplegado en un contenedor de Docker en Ubuntu WSL para asegurar una fácil portabilidad para el momento en que se deba instalar la aplicación y sus dependencias en otros dispositivos, y evitar confusiones acerca de versiones u otros aspectos con el resto de los integrantes del equipo.

Por otro lado, también es importante que el envío de datos hacia las unidades de bomberos sea al instante; aquí actúa la base de datos no relacional Firestore, que en este caso funciona como una capa de sincronización alimentada del *backend*, para que desde la plataforma web se envíen los datos hasta la aplicación móvil y así lograr enviar instantáneamente los planes de emergencia a las tabletas de bomberos cuando ocurre una emergencia.

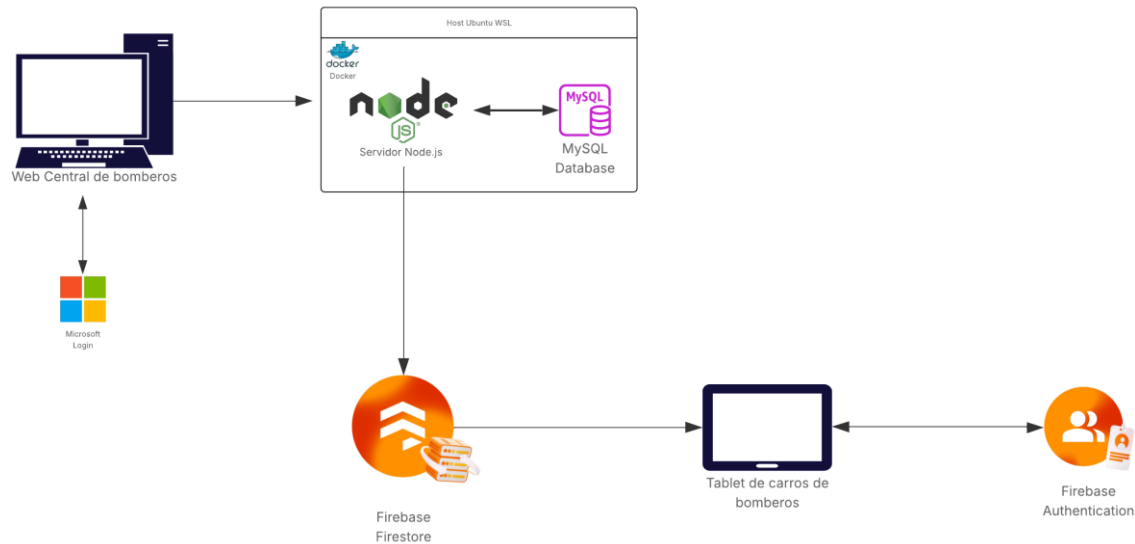


Figura 1. Diagrama de arquitectura del sistema
Fuente: Elaboración propia en Lucidchart

3.3 Diseño de la Base de Datos

3.3.1 Modelo Conceptual

En la primera etapa del modelado de la base de datos, el diagrama entidad-relación toma un rol clave; permite representar gráficamente de forma simple y resumida las entidades que estarán presentes en la base de datos junto con cómo se relacionan entre sí. Consta de tres elementos básicos u objetos semánticos:

- **Entidad:** Es un tipo de objeto que representa a un elemento de la vida real. Este puede ser concreto (perceptible por los sentidos) o abstracto (imaginable).
- **Atributos:** Características de una entidad. Según Silberschatz et al. [6], los atributos se pueden dividir en las siguientes clasificaciones:
 - **Simple o compuesto:** Los atributos simples no están divididos en subpartes (o sea, en otros atributos), en cambio los compuestos sí.
 - **Clave o descriptor:** El tipo clave individualiza a la entidad y el descriptor entrega características de esta.
 - **Monovaluado o multivaluado:** Se clasifican según el número de valores de una entidad.
 - **Base o derivado:** El atributo base (o almacenado) es el cual está explícitamente en la base de datos. El derivado, en cambio, no se almacena y se calcula. Como su nombre lo indica, su valor deriva de otros atributos.
- **Relaciones:** Son las asociaciones entre entidades y se expresan en el diagrama Entidad-Relación (en notación Chen) con un rombo que contiene anotado dentro de él, generalmente, un verbo.

Dentro de un diagrama entidad-relación es fundamental definir la cardinalidad. Esta propiedad establece si una relación es uno a uno, uno a muchos o muchos a muchos. Comprender la cardinalidad entre las tablas es fundamental para definir la estructura de una base de datos y garantizar que las relaciones entre ellas se representen con precisión.

Primeramente, se abordará una descripción de las entidades que estarán presentes en la base de datos, luego se mencionará la cardinalidad de cada una de ellas para luego terminar con el diagrama entidad-relación en notación UML. Las entidades clave para el sistema de FastPlan que se identificaron son:

1. *plan_emergencia*: Esta es la entidad principal del sistema la cual representa al documento principal que contiene toda la información sobre las edificaciones de comunidades como condominios, escuelas, etc. Tiene como propósito almacenar la dirección del edificio, nombre del lugar, estado del plan (si está vigente o desactualizado), archivo PDF del documento plan de emergencia, latitud y longitud para la georreferenciación y otros atributos.
2. *elemento_critico*: Esta entidad representa cada elemento o zona de seguridad que se encuentra dentro de un plan de emergencia, tales como grifos, vías de evacuación, sustancias químicas, extintores, red seca, entre otros. Al igual que el plan de emergencia, este también tiene latitud y longitud para poder identificar su ubicación mediante coordenadas y otro atributo que almacena datos de referencia a su ubicación. Además, es necesario incluir observaciones que sean importantes respecto al elemento en cuestión, por ejemplo, cuando el elemento crítico es una sustancia química tiene ciertos riesgos de los cuales bomberos debe tener precaución cuando haya que manipular o estar cerca de este.
3. *quimico*: Entidad que representa las sustancias químicas peligrosas que se encuentran disponibles para su posterior selección. Almacena información estandarizada para cada sustancia, tal como la hoja de seguridad, su nivel de riesgo y nombre. Su objetivo es eliminar la redundancia de datos permitiendo su reutilización en distintos elementos críticos de los planes de emergencia.
4. *ec_quimico*: Entidad intermediaria que vincula un elemento crítico de tipo sustancia química, con un químico del catálogo (*quimico*). Su función es detallar el inventario real de la sustancia que se encuentre en el punto georreferenciado.
5. *usuario*: La entidad Usuario es la persona que interactúa con el sistema, como atributo tiene "rol" el cual puede ser ya sea un operador de la central de bomberos (que se encarga de gestionar a las unidades de bomberos y de enviar la información crítica hacia la unidad que corresponda), como también un bombero o prevencionista de riesgos que se encargue de subir los planes de emergencia al sistema. Para la autenticación, el usuario debe ingresar con sus credenciales de Microsoft.
6. *cuerpo_bomberos*: Ya que el sistema puede ser escalable en un futuro a otras ciudades de Chile, se agrega esta entidad que representa a un cuerpo de bomberos. Un ejemplo de esta entidad es: Cuerpo de Bomberos de Viña del Mar o Cuerpo de Bomberos de Villa Alemana. Hay que destacar que cada cuerpo de bomberos puede tener muchas compañías asociadas.

7. *compania*: Define las distintas compañías que pueden pertenecer a un cuerpo de bomberos de una ciudad.
8. *carro*: Representa una unidad de bomberos, más conocido como carro de bomba. Este es quien recibe la información crítica y se identifica a cada unidad por su número. Además, cada carro cuenta con una patente, pertenece a una compañía y puede estar disponible, si es que no se encuentra en una emergencia, o no disponible, si está en una emergencia en curso.
9. *rol_usuario*: Roles de los usuarios del sistema según sea conveniente.
10. *estado*: Define el estado de otras entidades, como *plan_emergencia* (vigente o desactualizado) y *carro* (disponible o no disponible).
11. *tipo_elemento*: Define el tipo de elemento que puede ser un elemento crítico (sustancia química, grifo, salida de emergencia, zona segura, etc.).
12. *tipo_edificacion*: Define el tipo de edificación al cual pertenece el plan de emergencia (colegio, hospital, condominio, entre otros).

La incorporación de las cuatro últimas entidades tiene como propósito permitir que la organización pueda agregar más tipos de roles, elementos, estados, o edificaciones, según su conveniencia, evitando así dejar estos valores como un tipo fijo.

Las relaciones de cardinalidad establecidas entre ellas se mencionan a continuación:

Tabla 3. Descripción de cardinalidad entre entidades

Nombre	Descripción
<i>cuerpo_bomberos</i> → <i>compania</i>	Un cuerpo de bomberos tiene una o más compañías.
<i>compania</i> → <i>carro</i>	Una compañía tiene cero a muchos carros (unidades) asociados. Un carro pertenece a una sola compañía.
<i>usuario</i> → <i>cuerpo_bomberos</i>	Cada uno de los usuarios pertenecen respectivamente a un único cuerpo de bomberos.
<i>usuario</i> → <i>rol_usuario</i>	Cada usuario debe tener asignado un único rol de los que son definidos por la central.
<i>usuario</i> → <i>plan_emergencia</i>	En el sistema, un usuario puede crear y gestionar múltiples planes de emergencia que quedan asociados a su cuenta. Asimismo, se contempla el caso de no tener plan alguno vinculado a él (por ejemplo, si es un usuario nuevo o si no tiene los permisos de rol).
<i>plan_emergencia</i> → <i>estado</i>	Todos los planes tienen un estado asociado, como puede ser "vigente" o "desactualizado".

<i>plan_emergencia</i> → <i>tipo_edificacion</i>	Cada plan de emergencia pertenece a un único tipo específico de edificación, clasificándose en categorías como edificio, escuela, centro comercial, entre otros.
<i>carro</i> → <i>estado</i>	Todos los carros tienen un único estado asociado, que define la operatividad de un carro, como por ejemplo si es que este está disponible o no está disponible para una emergencia.
<i>plan_emergencia</i> → <i>elemento_critico</i>	Un plan de emergencia tiene de uno a muchos elementos críticos específicos, y cada uno de estos pertenecen exclusivamente a un plan.
<i>elemento_critico</i> → <i>tipo_elemento</i>	Cada elemento crítico debe ser identificado con su respectivo tipo, que determina su simbología y entrega información sobre el mismo.
<i>elemento_critico</i> → <i>ec_quimico</i>	Un elemento crítico puede tener múltiples registros de sustancias en <i>ec_quimico</i> .
<i>quimico</i> → <i>ec_quimico</i>	Una sustancia química puede estar presente en varios elementos críticos químicos (<i>ec_quimico</i>).

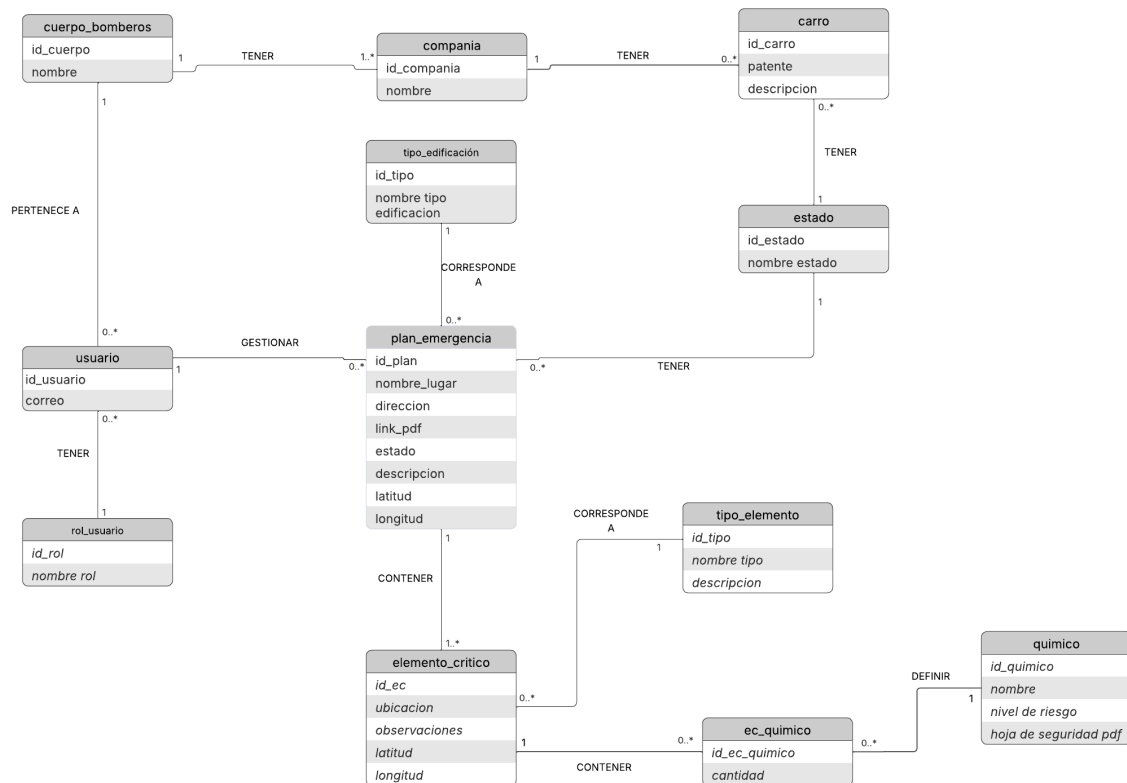


Figura 2. Modelo Entidad-Relación
Fuente: Elaboración propia en Lucidchart

3.3.2 Modelo Lógico

3.3.2.1 Componente Relacional

En el modelo lógico se realiza un mapeo del esquema conceptual al modelo relacional para lograr su transformación a dicho modelo aplicando ciertas reglas. En este proceso, las entidades se llevan a tablas y pueden surgir nuevas tablas intermediarias que no estaban en el modelo conceptual. Se definen todas las claves primarias y foráneas de las tablas de la base de datos y, además, se ha verificado que cada tabla cumpla con la Tercera Forma Normal (3NF), eliminando redundancias de datos y asegurando que cada atributo dependa únicamente de su clave primaria.

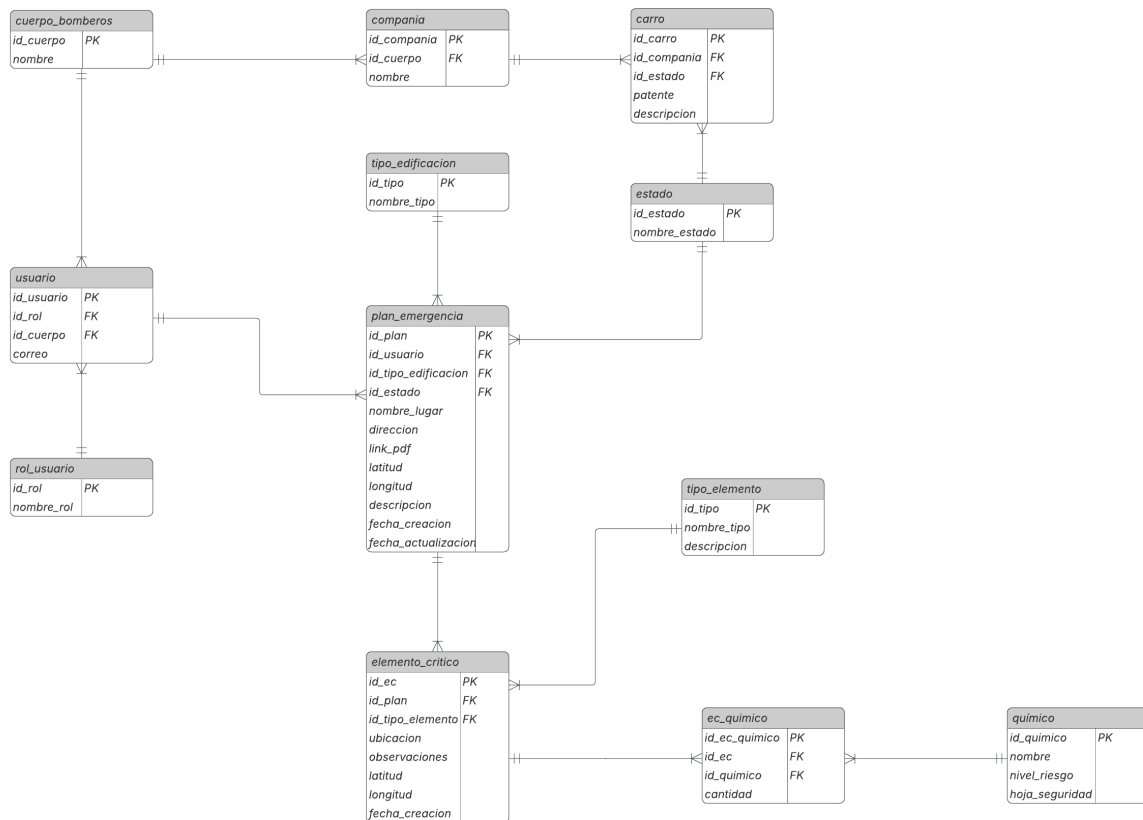


Figura 3. Modelo relacional de datos
Fuente: Elaboración propia en Lucidchart

3.3.2.2 Componente No Relacional

En esta fase del diseño, se realizó la transformación de datos desde el modelo relacional a una estructura en formato JSON (la cual fue realizada en conjunto con el equipo), para lograr la sincronización eficiente con los dispositivos móviles de los carros de bomberos.

Se aplicó una estrategia de desnormalización del modelo relacional utilizando la técnica de **anidamiento**, que consiste en agrupar todos los datos relacionados en un solo documento. Con este enfoque, se logra optimizar el tiempo de lectura y disminuye la latencia de red, facilitando rápidamente la información necesaria en momentos críticos.

Al momento de enviar la información crítica de los planes de emergencia registrados, estos se almacenan en Firestore con la siguiente estructura:

Colección *planes*: es la colección raíz que almacena los documentos para cada unidad de bomberos, garantizando así la unicidad.

Documentos <Patente>: Cada documento se sube teniendo como identificador (*Document ID*) la patente de la unidad de bomberos para que precisamente se envíen los datos hasta el dispositivo que se encuentre en el carro que se seleccione.

Esquema del documento: Cada documento encapsula todos los datos del plan de emergencia bajo una estructura desnormalizada, que incluye:

- Plan (principal): Objeto JSON con los datos de la edificación con emergencia en curso.
- Elementos críticos: arreglo que contiene todos los elementos críticos del plan (grifos, salidas de emergencia, extintores, etc.).
- Planes secundarios: Arreglo adicional destinado a almacenar los datos de las edificaciones cercanas al siniestro principal (cada uno tiene la misma estructura que el plan principal).

Tabla 4. Objeto *plan_principal* (tipo Map)

Clave	Tipo de dato	Descripción
<i>id_plan</i>	<i>number</i>	Identificador del plan dentro de la base de datos.
<i>nombre_lugar</i>	<i>string</i>	Nombre del recinto.
<i>tipo_edificio</i>	<i>string</i>	Categoría en que se clasifica la edificación.
<i>direccion</i>	<i>string</i>	Dirección del siniestro.
<i>latitud</i>	<i>number</i>	Coordenada geográfica del edificio.
<i>longitud</i>	<i>number</i>	Coordenada geográfica del edificio.
<i>fecha_creacion</i>	<i>timestamp</i>	Fecha de creación del registro.
<i>fecha_actualizacion</i>	<i>timestamp</i>	Fecha de modificación del registro.
<i>descripcion</i>	<i>string</i>	Descripción del recinto.
<i>elementos_criticos</i>	<i>array</i>	Lista de elementos críticos dentro del recinto.

Tabla 5. Array *elementos_criticos*

Clave	Tipo de dato	Descripción
<i>tipo_elemento</i>	<i>string</i>	Categoría del elemento.

<i>ubicacion</i>	<i>string</i>	Ubicación referencial del elemento.
<i>observaciones</i>	<i>string</i>	Detalles relevantes.
<i>latitud</i>	<i>number</i>	Coordenada geográfica
<i>longitud</i>	<i>number</i>	Coordenada geográfica
<i>quimicos</i>	<i>array</i>	Lista de sustancias peligrosas almacenadas en este punto.

Tabla 6. Array *quimicos*

Clave	Tipo de dato	Descripción
<i>nombre</i>	<i>string</i>	Nombre de la sustancia química.
<i>riesgo</i>	<i>string</i>	Nivel de peligrosidad.
<i>cantidad</i>	<i>string</i>	Volumen o cantidad almacenada.
<i>hoja_seguridad</i>	<i>string</i>	Ruta hacia el PDF de la hoja de seguridad del químico.

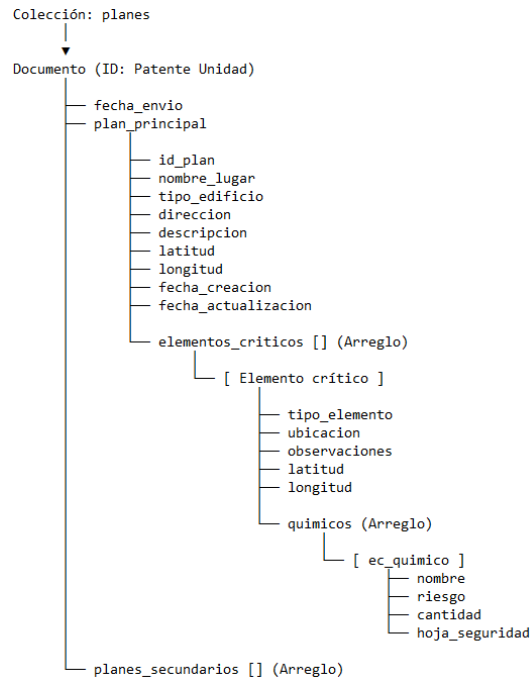


Figura 4. Esquema de documento JSON para planes de emergencia
Fuente: Elaboración propia

3.3.3 Modelo Físico

Para el diseño físico de la solución, se opta por una arquitectura "híbrida", dividiéndose en la base de datos relacional para garantizar la integridad de los datos y una base de datos no relacional que actúa como capa de sincronización en tiempo real y que está

optimizada para el envío de los datos estructurados (JSON) [15] hacia los dispositivos móviles.

Para la implementación física del modelo no relacional, se utilizó Cloud Firestore. Esta elección logró el correcto funcionamiento del sistema donde se materializó la estructura JSON diseñada. De esta forma, con esta configuración fue posible la sincronización en tiempo real con los datos estructurados, siendo administrado íntegramente por la nube de Google.

Para la componente relacional, el SGBD seleccionado fue MySQL. Para su implementación se optó por un contenedor en Docker sobre un entorno Linux (Ubuntu WSL), asegurando la portabilidad desde el entorno de pruebas al de producción y eliminando conflictos de versiones entre los distintos miembros del equipo.

Respecto al modelo relacional, se definieron tipos de datos de los atributos según el SGBD MySQL, garantizando la optimización de almacenamiento y de las consultas, y asegurando la integridad de los datos. Para las claves primarias y foráneas de las entidades principales, se seleccionó el tipo INT (4 bytes), en este caso se utiliza este tipo de dato para asegurar la escalabilidad a futuro del sistema, proyectando su despliegue a nivel nacional y su adaptación a diversas industrias. Por otro lado, las tablas de catálogo (*rol_usuario*, *estado*, *tipo_elemento* y *tipo_edificacion*), las cuales poseen un rango de valores acotados, utilizan el tipo TINYINT(3) UNSIGNED, el cual permite un máximo de 255 registros sin números negativos. En cuanto a los datos alfanuméricos, se empleó VARCHAR con longitudes variables según el contenido; para el atributo **patente** de la entidad **carro**, específicamente se usa **VARCHAR (10)**, restringiendo según el formato alfanumérico de identificación de vehículos, teniendo la posibilidad de añadir el guion separador si fuera necesario. Para la gestión de documentos extensos, como rutas para acceder a los archivos PDF de un plan de emergencia, se optó por el tipo TEXT. Debido a la importancia de la precisión geográfica en emergencias, las coordenadas (latitud y longitud) se definieron como DECIMAL (10,8), lo cual permite mayor exactitud de ubicación al no realizarse redondeos. Finalmente, para estandarizar el nivel de riesgo que puede tener un elemento crítico de tipo químico, se implementó ENUM, el cual limita los valores de entrada a: "CRÍTICO", "ALTO", "MEDIO" y "BAJO", para mantener la consistencia de los datos que se ingresen.

A continuación, se presenta el diccionario de datos según el diagrama relacional resultante:

Tabla 7. Diccionario de Datos del Modelo Relacional

Entidad (tabla)	Atributo	Tipo de dato	Llave	Descripción
<i>cuervo_bomberos</i>	<i>id_cuerpo</i>	INT	PK	Identificador del Cuerpo de Bomberos.
	<i>nombre</i>	VARCHAR (150)	-	Nombre de la institución (ej: Cuerpo de Bomberos de Viña del Mar)
<i>compania</i>	<i>id_compania</i>	INT	PK	Identificador de la Compañía.
	<i>id_cuerpo</i>	INT	FK	Relación con el Cuerpo de



<i>usuario</i>	<i>nombre</i>	VARCHAR (100)	-	Bomberos al que pertenece. Nombre de la Compañía.
	<i>id_usuario</i>	INT	PK	Identificador de cada usuario del sistema.
	<i>id_rol</i>	TINYINT(3) UNSIGNED	FK	Relación con el rol del usuario.
	<i>id_cuerpo</i>	INT	FK	Relación con el Cuerpo de Bomberos del usuario.
<i>rol_usuario</i>	<i>correo</i>	VARCHAR(150)	-	Correo electrónico institucional.
	<i>id_rol</i>	TINYINT(3) UNSIGNED	PK	Identificador del rol.
	<i>nombre_rol</i>	VARCHAR(45)	-	Nombre del perfil/rol (admin, usuario, prevencionista)
<i>carro</i>	<i>id_carro</i>	INT	PK	Identificador de cada unidad/carro.
	<i>id_estado</i>	TINYINT(3) UNSIGNED	FK	Estado operativo del carro.
	<i>id_compania</i>	INT	FK	Compañía de la unidad.
	<i>patente</i>	VARCHAR(10)	-	Placa patente del vehículo.
<i>estado</i>	<i>descripcion</i>	VARCHAR(100)	-	Descripción técnica del carro.
	<i>id_estado</i>	TINYINT(3) UNSIGNED	PK	Identificador del estado.
	<i>nombre_estado</i>	VARCHAR(45)	-	Nombre/descripción del estado (Disponible, Vigente, etc).
<i>tipo_edificacion</i>	<i>id_tipo</i>	INT	PK	Identificador de tipo de edificación.
	<i>nombre_tipo</i>	VARCHAR(45)	-	Nombre del tipo de edificación.
<i>plan_emergencia</i>	<i>id_plan</i>	INT	PK	Identificador del plan de emergencia.
	<i>id_usuario</i>	INT	FK	Usuario responsable



				que subió el plan.
	<i>id_tipo_edificacion</i>	TINYINT(3) UNSIGNED	FK	Categoría del edificio del plan.
	<i>id_estado</i>	TINYINT(3) UNSIGNED	FK	Estado de vigencia en el que se encuentra el plan.
	<i>nombre_lugar</i>	VARCHAR(150)	-	Nombre del recinto.
	<i>direccion</i>	VARCHAR(255)	-	Dirección del recinto.
	<i>link_pdf</i>	TEXT	-	Ruta al plan de emergencia en PDF.
	<i>latitud</i>	DECIMAL(10,8)	-	Coordenada geográfica.
	<i>longitud</i>	DECIMAL(10,8)	-	Coordenada geográfica.
	<i>descripcion</i>	VARCHAR(255)	-	Descripción del recinto (número de pisos, departamentos, etc.).
	<i>fecha_creacion</i>	TIMESTAMP	-	Fecha y hora del registro del plan.
	<i>fecha_actualizacion</i>	TIMESTAMP	-	Fecha de la última modificación del plan.
<i>tipo_elemento</i>	<i>id_tipo</i>	TINYINT(3) UNSIGNED	PK	Identificador del tipo de elemento
	<i>nombre_tipo</i>	VARCHAR(100)	-	Nombre del tipo elemento (Grifo, Extintor, Red húmeda, Químicos).
	<i>descripcion</i>	VARCHAR(255)	-	Descripción del uso del elemento.
<i>elemento_critico</i>	<i>id_ec</i>	INT	PK	Identificador del elemento crítico.
	<i>id_plan</i>	INT	FK	Plan de emergencia donde está el elemento.
	<i>id_tipo_elemento</i>	TINYINT(3) UNSIGNED	FK	Clasificación del elemento.



quimico

ec_quimico

<i>ubicación</i>	VARCHAR(150)	-	Ubicación de referencia del elemento en el recinto.
<i>observaciones</i>	VARCHAR(500)	-	Detalles relevantes para bomberos.
<i>latitud</i>	DECIMAL(10,8)	-	Coordenada geográfica.
<i>longitud</i>	DECIMAL(10,8)	-	Coordenada geográfica.
<i>fecha_creacion</i>	TIMESTAMP	-	Fecha de registro del punto.
<i>id_quimico</i>	INT	PK	Identificador de la sustancia en cuestión.
<i>nombre</i>	VARCHAR(255)	-	Nombre técnico estandarizado de la sustancia.
<i>nivel_riesgo</i>	ENUM	-	Nivel de peligrosidad (CRÍTICO, ALTO, MEDIO, BAJO).
<i>hoja_seguridad</i>	TEXT	-	Enlace a la hoja de seguridad en PDF.
<i>id_ec_quimico</i>	INT	PK	Identificador del elemento crítico químico.
<i>id_ec</i>	INT	FK	Elemento crítico al cual está asociado, correspondiente a un plan de emergencia específico.
<i>id_quimico</i>	INT	FK	Sustancia química almacenada.
<i>cantidad</i>	VARCHAR(100)	-	Cantidad específica almacenada.

Finalmente, a continuación, se observa el modelo relacional creado con ingeniería inversa en MySQL Workbench:

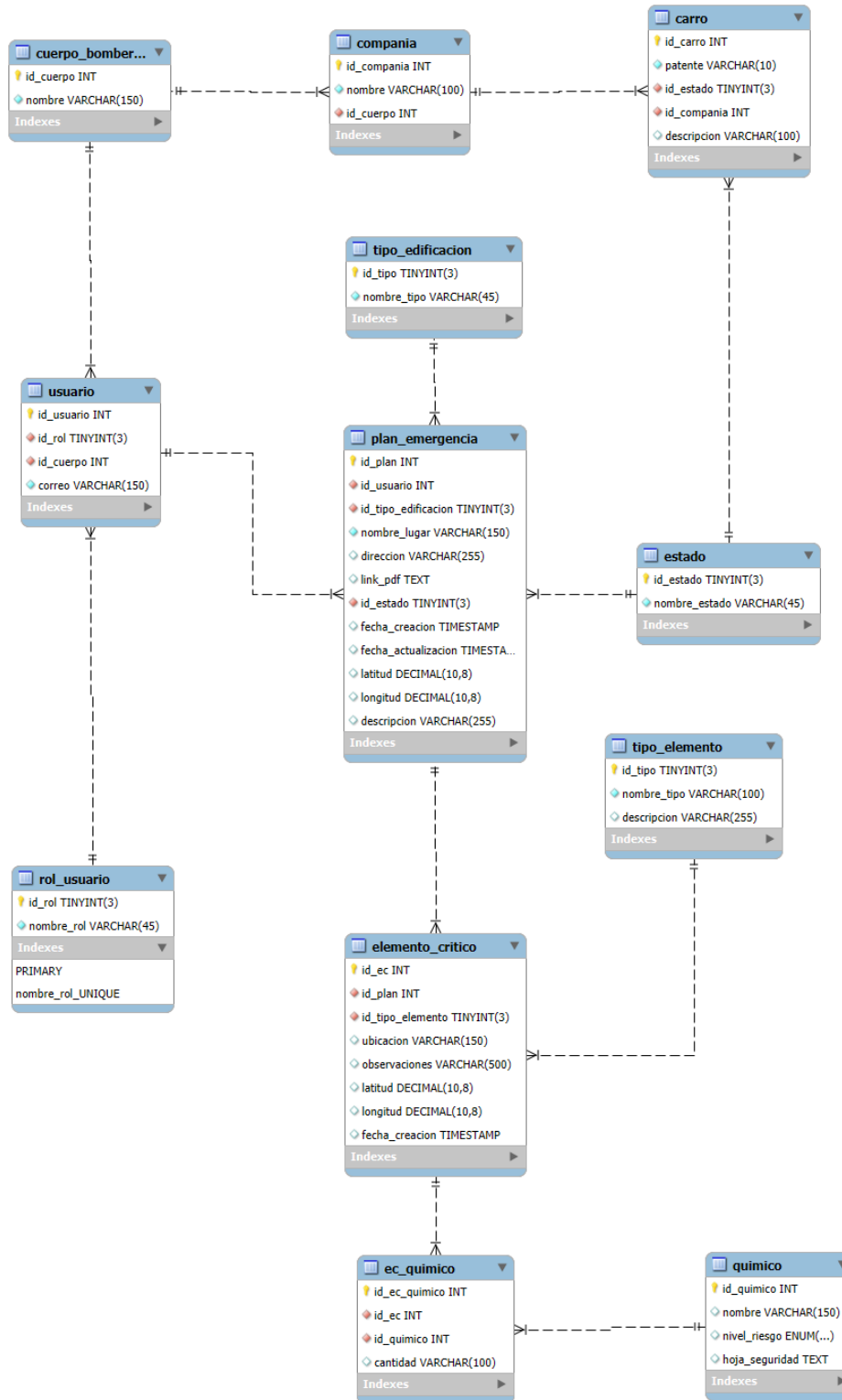


Figura 5. Modelo físico MySQL
Fuente: Elaboración en MySQL Workbench

4 Implementación

Este capítulo detalla la implementación y validación de la solución propuesta. Se aborda infraestructura, inicialización de la base de datos e integración de datos tanto simulados como reales para las pruebas posteriores.

4.1 Estrategia de despliegue

Para garantizar que el entorno de ejecución sea consistente entre los miembros del equipo de desarrollo, el SGBD MySQL será levantado en un contenedor en Docker. Esta decisión técnica se basa en la capacidad de los contenedores para aislar las dependencias, de esta forma se evitan conflictos de versiones que puedan estar instaladas localmente y asegura que todos los miembros del equipo de desarrollo utilicen exactamente la misma versión. Para ello se decidió utilizar la imagen oficial de MySQL disponible en *Docker Hub*, utilizando una versión estable (específicamente la versión 8.0) de este gestor de bases de datos asegurando que siempre se ejecute en esta.

En la etapa actual del sistema, no se definen volúmenes persistentes ya que para el equipo es necesario que la base de datos cada vez se “reinicie” a su estado original, eliminando datos de prueba acumulados anteriormente. Esto trabaja en conjunto del algoritmo de inicialización creado en el *backend* (descrito en la sección a continuación), el cual detecta el estado vacío de la base de datos y realiza el poblamiento con los datos por defecto de prueba que fueron definidos.

4.2 Implementación de la base de datos relacional

Para efectos de organización, se exportó el script del *schema* definitivo de la base de datos, mediante la utilización de la herramienta MySQL Workbench, para que quede guardada en el proyecto. Posteriormente este archivo es utilizado por el *backend* mediante un algoritmo de inicialización automática al levantar el servidor; este realiza las siguientes validaciones:

1. Verifica la conectividad asegurando que el contenedor de Docker esté activo y respondiendo.
2. El sistema toma e inyecta el script maestro *schema.sql* que se encuentra dentro del proyecto, generando automáticamente la estructura completa de las tablas y sus relaciones.
3. Gestión de datos semilla: Una vez que la estructura está creada, verifica si las tablas contienen datos; de no ser así procede a poblar insertando los datos que se encuentran en el archivo *seed.sql*.

4.3 Poblamiento de datos

Para la demostración, se pobló la base con datos de inicio en un archivo llamado *seed.sql* que también se ejecuta automáticamente tras la creación del esquema sql. Este archivo contiene datos simulados de varias edificaciones de la ciudad de Viña del Mar y se incluye a la Universidad Federico Santa María Sede Viña del Mar.

Como hito principal de validación se realizó un recorrido en terreno por la mayor parte de los emplazamientos de la universidad. Se logró recopilar la gran mayoría de elementos críticos presentes en el establecimiento junto con su ubicación, para así georreferenciarlos luego desde la central. Estos elementos críticos, en su gran mayoría,

no se encontraban en el plan de emergencia oficial de la universidad (este únicamente presentaba un plano con las zonas de seguridad y algunos desfibriladores). Una vez obtenidos los datos, fue posible insertarlos dentro de la base de datos para luego ser visualizados de manera gráfica según su categorización dentro del mapa interactivo del sistema.

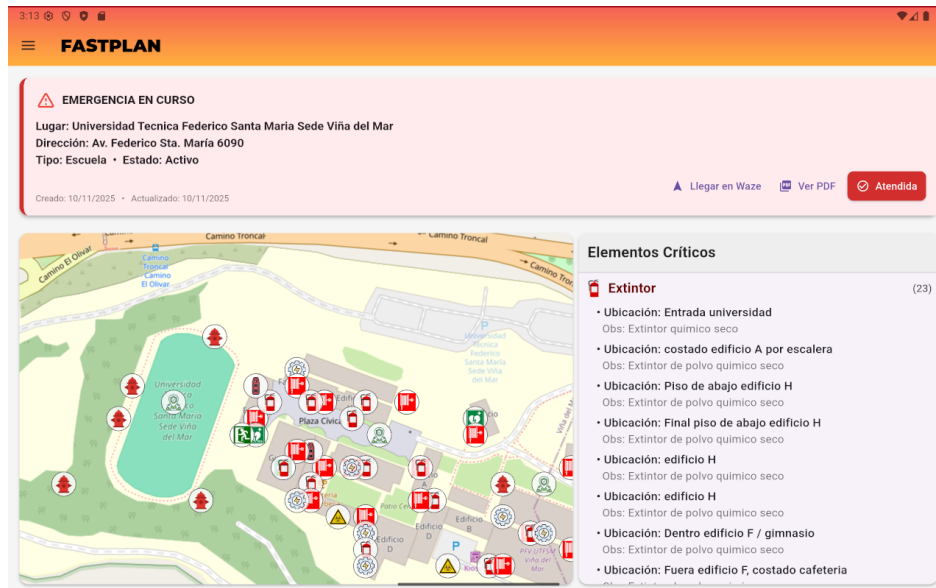


Figura 6. Vista de inicio de la aplicación con una emergencia en curso
Fuente: Captura de pantalla de FastPlan

4.4 Transformación y sincronización de los datos

Para lograr la comunicación efectiva entre el modelo relacional y el modelo NoSQL, se implementó un proceso de ETL [16] en tiempo real que es desencadenado al momento que se acciona el evento de despachar unidades desde la central. Este pasa por tres etapas:

1. Extracción: Cuando el operador confirma un despacho de información hacia una unidad de bomberos, el sistema ejecuta consultas complejas para recuperar todos los datos asociados a la emergencia seleccionada. Esto incluye:
 - Datos del plan de emergencia.
 - La totalidad de sus elementos críticos asociados.
 - Datos de planes de emergencia colindantes (los define el operador según un radio de acción).
2. Transformación: Una vez obtenidos los datos en formato tabular (filas y columnas), se aplica un algoritmo para transformarlos en una estructura jerárquica (JSON).
 - Desnormalización de elementos críticos: Las filas de los elementos críticos se agrupan en arreglos (*arrays*) dentro del objeto del plan de emergencia.
 - Planes secundarios: El sistema procesa también los planes de los edificios colindantes a la emergencia central (según lo defina el operador). Estos

se transforman y se anidan en un arreglo adicional dentro del documento de despacho a un carro de bomberos. Esta estrategia permite que bomberos tenga un acceso anticipado a la información crítica perimetral, esto es crucial si es que la emergencia llega a propagarse.

- Gestión de estado operativo del carro de bomberos: Simultáneamente, se ejecuta una transacción de actualización en MySQL, donde se modifica el *id_estado* del carro, quedando como "No disponible" para otra emergencia.

3. Carga: Finalmente el objeto JSON se carga en la nube, la cual se basa en la patente de la unidad como identificador único.

+ Iniciar colección	+ Agregar documento	+ Iniciar colección
planes >	ABC123 >	+ Agregar campo
	KLSDJ45	estado: "enviado"
	PLMNB34	fecha_envio: "2025-10-30T19:07:52.702Z"
	QWERT12	plan_principal
	ZXCVB99	direccion: "14 Nte. 821, Viña del Mar, Valparaiso"
		elementos_criticos
		0
		latitud: -33.0099
		longitud: -71.5468
		observaciones: "Extintor tipo K para cocinas"
		tipo_elemento: "Extintor"
		ubicacion: "Piso 1 - Zona restaurantes"
		1
		latitud: -33.0101
		longitud: -71.5466
		observaciones: "Ruta de evacuación señalizada"

Figura 7. Documentos JSON en Cloud Firestore
Fuente: Captura de la consola de Cloud Firestore

4.5 Autenticación y acceso de usuarios

4.5.1 Plataforma Web de la Central

Para garantizar la seguridad de la plataforma web de la Central, se optó por la utilización de Microsoft para que bomberos utilicen sus correos institucionales habituales (y evitar el tener que usar otras credenciales de acceso nuevas) y además delegar la gestión de credenciales y recuperación directamente a Microsoft. Todos los correos que estén registrados en la tabla Usuario, dentro de la base de datos, tendrán la autorización para ingresar a la plataforma de la central de bomberos. Dicha entidad también implementa un control de autorización interno del sistema, mediante roles asignados a cada usuario, complementándose así con el control de acceso de externo.

Este es esencial para la seguridad y la definición de las tareas que cada rol tiene permitido realizar. En la versión actual del sistema se definen tres únicos roles conocidos:

Admin: tiene acceso a todas las funciones de la plataforma web.

Prevencionista de riesgos: Tiene acceso solamente a gestionar los planes de emergencia, sus elementos críticos y su ubicación en el mapa.

Operador: Se encarga de la gestión de carros de bomberos y de enviar los planes de emergencia a las unidades pertinentes, no tiene acceso a gestionar los planes de emergencia.

Cabe destacar que, para complementar el control de acceso, la base de datos incorpora la vinculación de un usuario a un plan de emergencia (en este caso el que se encargó de subir la información); de esta forma, si existe algún error en los datos, se garantiza que siempre haya un registro de quien fue el responsable, lo cual es vital para mantener la veracidad y calidad de los datos.

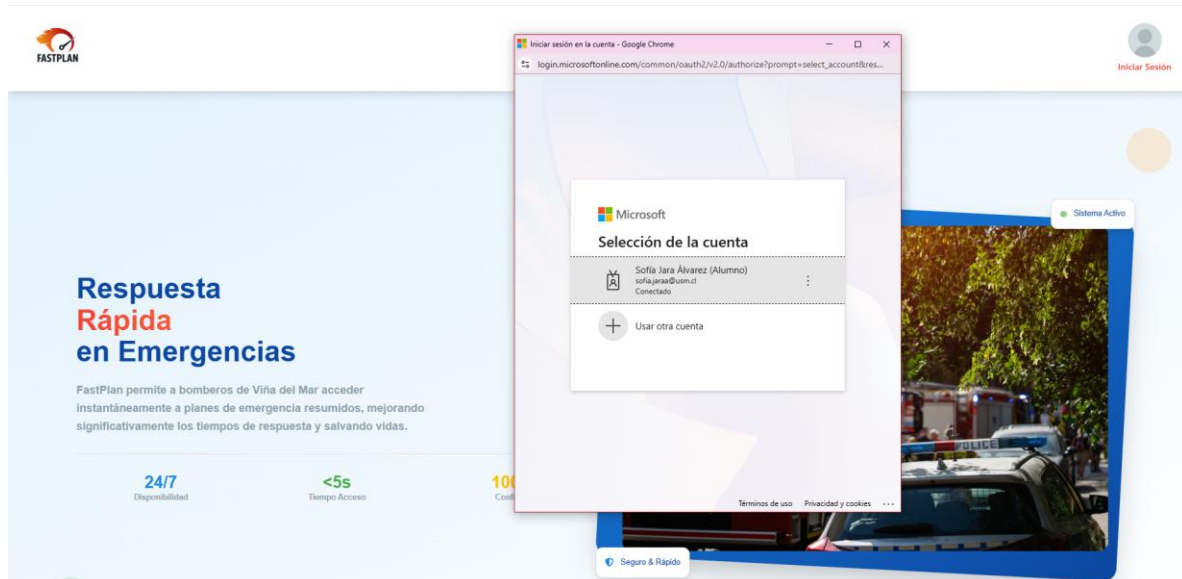



Figura 8. Inicio de sesión con Microsoft en plataforma web
Fuente: Captura de pantalla de FastPlan


4.5.2 Plataforma Móvil en tabletas

Finalmente, para la aplicación móvil desplegada en los carros de bomberos, se decidió integrar Firebase Authentication [17] para poder gestionar las cuentas con acceso desde la consola de Firebase. Se descartó el uso de un inicio de sesión usando solamente la patente del carro, ya que, cualquier tercero no autorizado que tenga conocimiento de esta identificación y que tenga la aplicación, tendría acceso a visualizar los datos sensibles de planes de emergencia o podrían llegar a interferir con los operativos en curso. Además, en este caso no se utiliza autenticación con Microsoft, ya que los carros no deben quedar atados a un usuario de bomberos.

El formato definido para las credenciales es <patente>@bomberos-vina.cl, lo que corresponde a que cada carro de bomberos tenga su propia cuenta para luego en la aplicación móvil ingresar de forma sencilla con la patente del carro y la contraseña creada por la administración desde la consola de Firebase.

Authentication

Usuarios Método de acceso Plantillas Uso Configuración  Extensions

 Las siguientes funciones de autenticación dejarán de estar disponibles cuando se cierre Firebase Dynamic Links pronto: la autenticación a través de vínculos de correo electrónico en apps para dispositivos móviles, así como la compatibilidad con OAuth de Cordova para apps web.

Buscar por dirección de correo electrónico, número de teléfono o UID de usuario

Agregar usuario

Identificador	Proveedores	Fecha de creación ↓	Fecha de acceso	UID del usuario
plmb34@bomberos-v...		25 nov 2025		DprABep3CbRAc5RgjOk0iJL...
abcd12@bomberos-vin...		25 nov 2025	26 nov 2025	PWme44biCOVLzisSm6egs7Q...
abc123@bomberos-vin...		25 nov 2025	26 nov 2025	JnOYxqHdVFM9hhFvz4LjkPe...

Filas por página: 50 1 - 3 of 3

Figura 9. Gestión de usuarios en consola de Firebase Authentication
Fuente: Captura de pantalla de consola de Firebase Authentication

4.6 Carga de datos

Para alimentar la base de datos con los planes de emergencia, se crearon interfaces web que permiten digitalizar dichos documentos y también tener un control de la flota de bomberos. Estos datos posteriormente serán procesados por el *backend* para cumplir con el propósito final: visualizar los datos críticos de los planes de emergencia desde una unidad de bomberos.

4.6.1 Digitalización de Planes de Emergencia

A través de esta interfaz donde se suben los datos, es posible ingresar nuevos planes de emergencia y sus respectivos elementos críticos. Permite ingresar lo siguiente:

- PDF del plan de emergencia
- Tipo de edificación
- Nombre del lugar
- Ubicación: esta se puede ingresar tanto manual como ser marcada desde el mapa interactivo integrado en el sistema.
- Elementos críticos: se pueden añadir tantos sean necesarios asociados a un mismo plan. Se selecciona su tipo, alguna observación importante, lugar de referencia de donde está ubicado y su ubicación mediante coordenadas, la cual no se ingresa manualmente; debe ser marcada directamente desde el mapa interactivo y la API de mapas se encarga de asignar automáticamente la latitud y longitud.

Cabe mencionar que, si el elemento crítico es de tipo "sustancia química", el sistema despliega un sub-formulario donde se selecciona la sustancia química que previamente fue definida en la base de datos.

Si es necesario, se pueden eliminar elementos críticos que se hayan añadido por equivocación.

Figura 10. Interfaz de carga de planes de emergencia
Fuente: Captura de pantalla de FastPlan

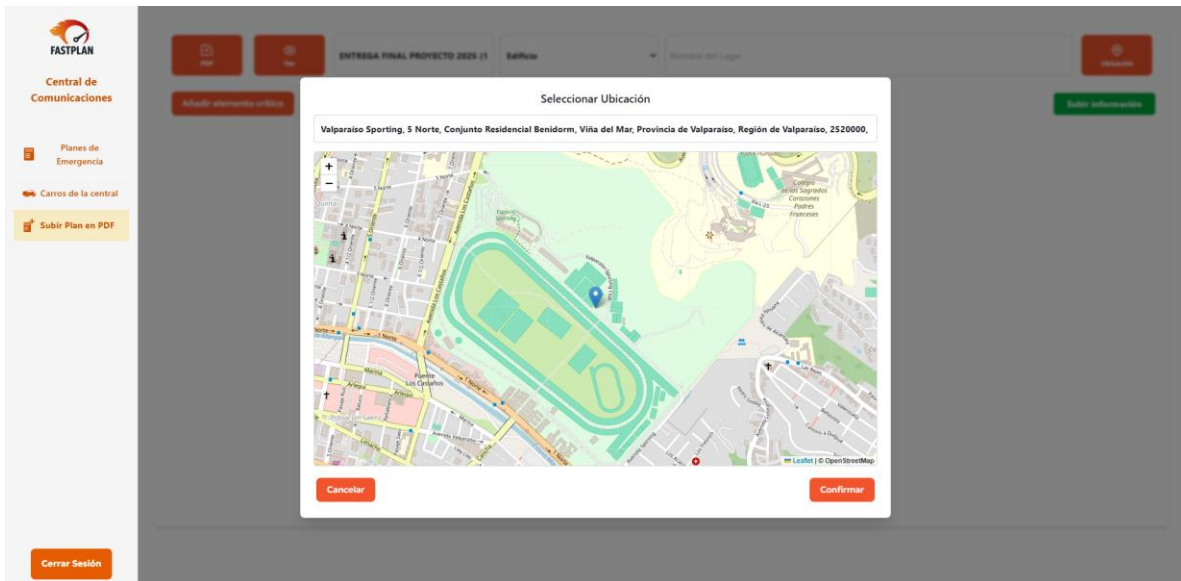


Figura 11. Selección de ubicación mediante mapa interactivo
Fuente: Captura de pantalla de FastPlan

4.6.2 Inventario de Unidades de Bomberos

Se implementó una sección de administración de unidades (carros) de bomberos, donde inmediatamente se visualiza un formulario que permite crear un nuevo carro de bomberos perteneciente a la flota, siendo el número de unidad el ID del carro. Al crearse un nuevo carro, automáticamente se le asigna el estado "Disponible".

Con el fin de evitar asignaciones erróneas y mantener el inventario actualizado, es posible eliminar carros de bomberos que fueron subidos por error o que estén fuera de servicio (ya no existe), retirándolos de la base de datos.

Ingresar nuevo carro de bomberos

En esta sección puedes agregar nuevos carros de bomberos a la central. Esto se verá reflejado en el dashboard.

Numero de Unidad

Patente

Descripción

Guardar carro

Carros registrados

Unidad	Patente	Estado	Descripción	Acciones
U-1	ABCD12	No Disponible	Carro de Agua / Emergencias Generales	Eliminar
U-2	KJTR34	Disponible	Carro de Químicos / Hazmat	Eliminar
U-3	ZXCV56	Disponible	Carro de Rescate / Escalera	Eliminar
U-11	QWERT12	Disponible	Carro de Agua	Eliminar
U-12	PLMNB34	Disponible	Carro de Agua	Eliminar
U-13	ZXCVB09	Disponible	Carro escala	Eliminar

Figura 12. Interfaz de gestión de unidades de bomberos
Fuente: Captura de pantalla de FastPlan

5 Pruebas

5.1 Pruebas de rendimiento

En el *backend* se realizó un monitoreo de la latencia en milisegundos del proceso de extracción, transformación y carga de datos hacia Firestore para que finalmente lleguen hasta las tabletas de bomberos. Para validar el rendimiento, se analizaron diferentes casos teniendo en consideración los posibles escenarios que se puedan dar durante una emergencia:

Caso 1 de carga básica de datos: Plan de emergencia de un único edificio con carga mínima de elementos.

Tabla 8. Tiempos de latencia - Caso 1

ID_PLAN	EXTRACCIÓN SQL (ms)	TRANSFORMACIÓN (ms)	CARGA FIRESTORE (ms)	TIEMPO TOTAL 1 (ms)
5	17,7	0,03	1.094,18	1.123,76
5	14,04	0,02	165,31	206,97
5	14,69	0,02	128,1	149,24
5	13,26	0,02	114,99	157,72
5	18,07	0,02	100,05	124,31
5	12,47	0,02	111,89	132,4
5	12,69	0,03	86,31	106,1
5	13,98	0,02	59,09	99,76

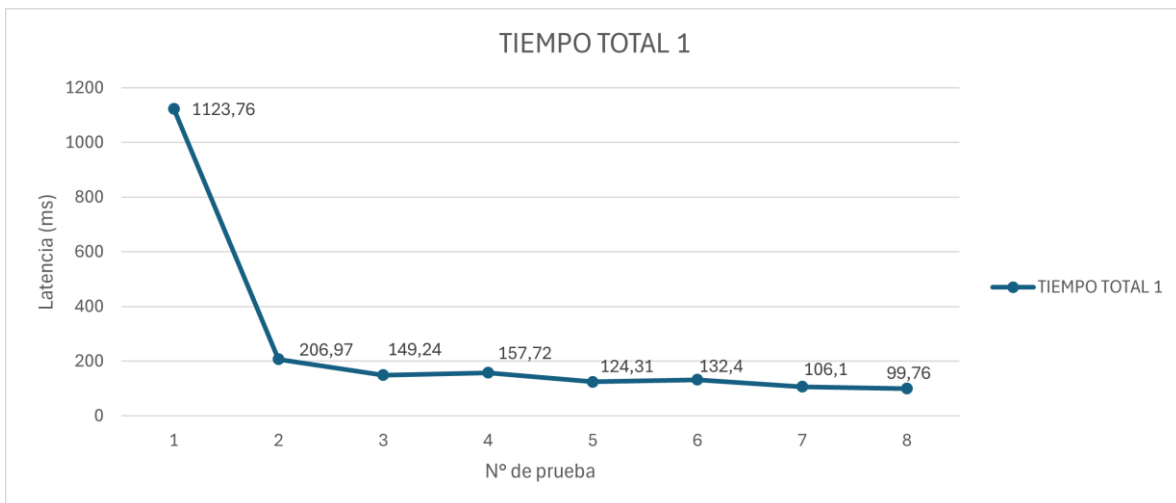


Figura 13. Gráfico de latencia carga básica de datos (Caso 1)
Fuente: Elaboración propia

Caso 2 de carga alta de datos: Plan de emergencia de la Universidad Técnica Federico Santa María Sede Viña del Mar, con más de 100 elementos críticos.

Tabla 9. Tiempos de latencia – Caso 2

ID_PLAN	EXTRACCIÓN (ms)	SQL	TRANSFORMACIÓN (ms)	CARGA FIRESTORE (ms)	TIEMPO TOTAL 2 (ms)
9	133,64		0,02	246,4	388,73
9	136,13		0,02	376,79	520,06
9	137,67		0,02	135,92	338,31
9	166,58		0,02	151,92	331,27
9	163,95		0,01	243,05	2627,03
9	129,22		0,01	113,99	269,03
9	140,93		0,01	100,39	251,61
9	142,32		0,02	102,42	254,45

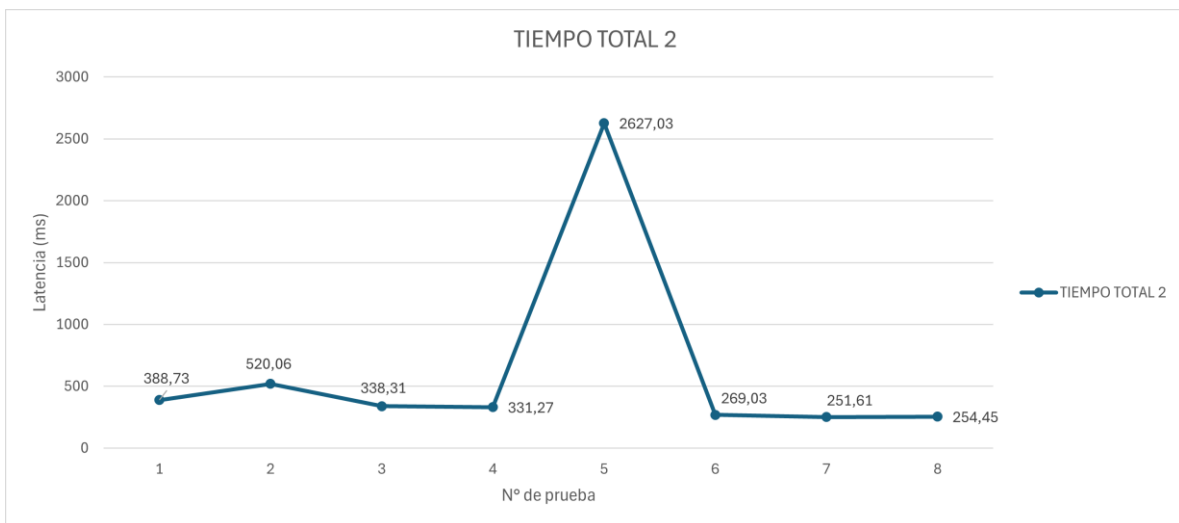


Figura 14. Gráfico de latencia en carga alta de datos (Caso 2)
Fuente: Elaboración propia

Caso 3 de carga perimetral: Plan de emergencia con pocos elementos, pero con más de 4 edificios colindantes con un radio que abarca 4 - 6 planes (caso extremo).

Tabla 10. Tiempos de latencia - Caso 3

ID_PLAN	EXTRACCIÓN (ms)	SQL	TRANSFORMACIÓN (ms)	CARGA FIRESTORE (ms)	TIEMPO TOTAL 3
5	105,03		0,03	3.169,37	3.288,71
5	37,98		0,03	155,37	224,61
5	30,69		0,02	187,67	229,27
5	26,9		0,01	164,14	209,7
5	26,96		0,01	172,67	210,84
5	28,52		0,06	2.518,69	2.574,31
5	33,48		0,01	151,7	194,31
5	27,54		0,02	151,06	205,76

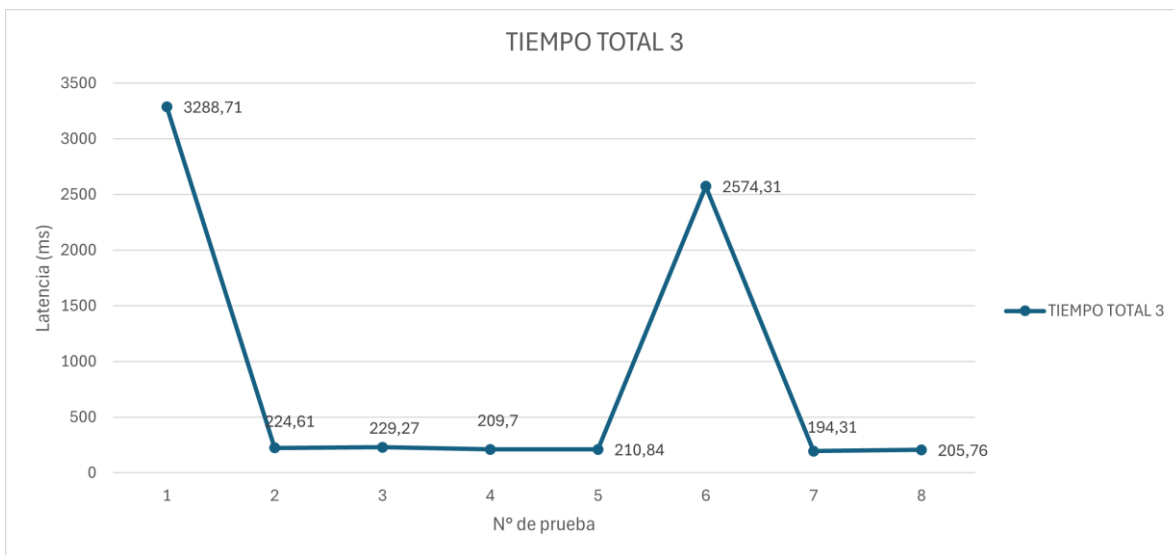


Figura 15. Gráfico de latencia en carga perimetral (Caso 3)
Fuente: Elaboración propia

Los picos de latencia observados en todos los casos corresponden a variaciones transitorias en el tiempo de escritura de Firestore. Dichos aumentos puntuales son un comportamiento habitual y se deben principalmente a fluctuaciones momentáneas en la red y, en las primeras ejecuciones, al *cold start* (establecimiento de la conexión inicial) con el cliente.

5.2 Pruebas de estrés

Para validar el límite de operabilidad de la arquitectura implementada, se sometió al sistema a pruebas de estrés con múltiples peticiones simultáneamente, con el objetivo de verificar la capacidad de respuesta en momentos donde hay una demanda crítica.

Para ello se enviaron múltiples peticiones utilizando el plan de emergencia con id = 9 que corresponde al plan con datos de la universidad, con más de 100 elementos críticos asociados. A continuación, se detallan los resultados obtenidos:

Caso 1: Se realizaron 10 pruebas consecutivas con 20 peticiones simultáneas cada una. Todas las pruebas tuvieron una respuesta exitosa, sin registrar fallos ni pérdida de datos.

Tabla 11. Resultados prueba de estrés (Caso 1)

N.º Prueba	Carga (Peticiones Simultáneas)	Peticiones exitosas	Peticiones fallidas	Tiempo total del conjunto de peticiones (ms)	Latencia Promedio por Petición (ms)
1	20	20	0	1.826,89	91,34
2	20	20	0	1.257,22	62,86
3	20	20	0	930,98	46,55
4	20	20	0	799,41	39,97
5	20	20	0	743,35	37,17
6	20	20	0	764,24	38,21
7	20	20	0	867,62	43,38
8	20	20	0	802,82	40,14
9	20	20	0	824,04	41,2
10	20	20	0	783,58	39,18
PROMEDIO	20	100%	0	960,02	48

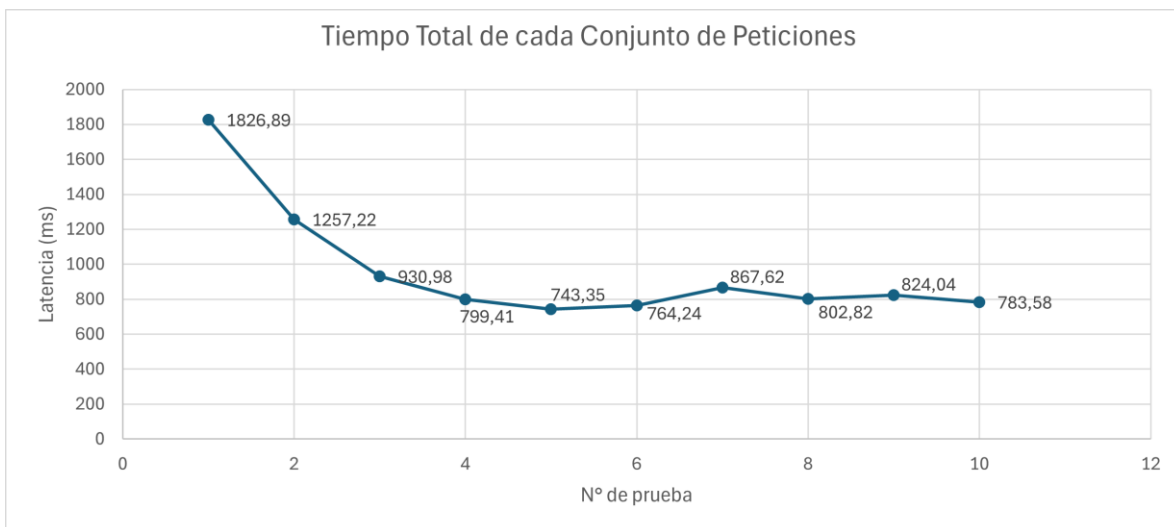


Figura 16. Tiempos de respuesta con 20 peticiones simultáneas
Fuente: Elaboración propia

Caso 2: Se realizaron 10 pruebas, las primeras 5 con 50 peticiones simultáneas y las otras 5 con 100 peticiones simultáneas, esto simulando una situación extrema. Luego de realizar estas pruebas, se obtuvo que, nuevamente, todas las respuestas fueron exitosas. Sin embargo, claramente hay un aumento en el tiempo de respuesta que es de esperarse por el nivel de estrés al que se está sometiendo; aun así, no es una latencia demasiado alta con respecto al número de peticiones.

Tabla 12. Resultados prueba de estrés (Caso 2)

Escenario	Carga (Peticiones Simultáneas)	Peticiones exitosas	Peticiones fallidas	Tiempo total del conjunto de peticiones (ms)	Latencia Promedio por Petición (ms)
Carga Alta 1	50	50	0	2.274,7	45,49
Carga Alta 2	50	50	0	2.204,21	44,08
Carga Alta 3	50	50	0	2.449,02	48,98
Carga Alta 4	50	50	0	1.902,5	38,05
Carga Alta 5	50	50	0	2.032,29	40,65
PROMEDIO (50)	50	100%	0	2.172,54	43,45
-----	-----	-----	-----	-----	-----
	-				-
Carga Crítica 1	100	100	0	3.889,18	38,89
Carga Crítica 2	100	100	0	3.985,16	39,85
Carga Crítica 3	100	100	0	3.611,88	36,12
Carga Crítica 4	100	100	0	4.261,66	42,62
Carga Crítica 5	100	100	0	3.892,91	38,93
PROMEDIO (100)	100	100%	0	3.928,16	39,28

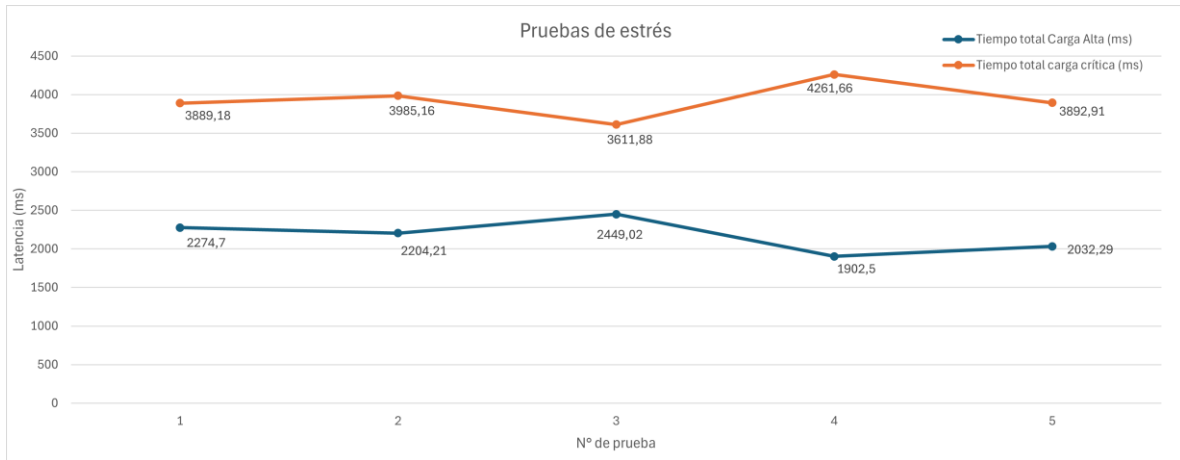


Figura 17. Comparativa de tiempos en pruebas de estrés (Carga Alta vs Crítica)
Fuente: Elaboración propia

6 Conclusiones

En la presente propuesta se cumplió con el objetivo de diseñar e implementar una base de datos capaz de centralizar y gestionar los planes de emergencia y sus elementos críticos, resolviendo la problemática y necesidad de accesibilidad detectada por Bomberos de Viña del Mar.

La decisión de implementar una arquitectura híbrida utilizando MySQL y Firebase Firestore demostró ser la estrategia adecuada para mantener la integridad de los datos y a la vez tener una alta disponibilidad de estos, requerida para el despliegue de información en las unidades de bomberos logrando una sincronización exitosa entre ambas bases de datos.

El levantamiento en terreno realizado en la Universidad Técnica Federico Santa María, un emplazamiento con una gran extensión y que tiene múltiples edificios (aulas, oficinas, casino, biblioteca y espacios abiertos), validó el correcto funcionamiento y flexibilidad del modelo de datos. Permitted digitalizar una cantidad significativa de elementos críticos dentro del recinto que no figuraban en el documento oficial de plan de emergencia de la universidad. La georreferenciación implementada ofrece otra ventaja sobre los planos en papel: logra que bomberos visualice los recursos antes de llegar a la zona de emergencia, optimizando su trabajo y la coordinación de los equipos.

A futuro, este sistema tiene el potencial de ser implementado en Cuerpos de Bomberos de otras ciudades, permitiendo la estandarización de la información crítica de planes de emergencia a nivel nacional. Asimismo, su arquitectura flexible permite su adaptación para ser utilizada en el sector privado, por ejemplo, en la industria minera donde se manejan distintas sustancias químicas peligrosas. En este contexto, se proyecta también incorporar el rombo de seguridad NFPA 704 [18] dentro de lo que son los elementos críticos de tipo sustancia química, otorgando de manera visual e inmediata los niveles de riesgo de estos elementos. De igual manera se contempla la integración de un sistema de visualización de planos en 3D, que represente de mejor manera los elementos de las edificaciones.

Adicionalmente, se contempla la interoperabilidad entre distintas instituciones, permitiendo compartir datos del sistema hacia Carabineros de Chile u otros organismos de primera respuesta.

En cuanto a seguridad de la información, se proyecta robustecer la plataforma implementando más medidas de auditoría, como registro de modificaciones de datos que puedan ser realizadas. Además, se tiene en consideración fortalecer el *backend* con medidas de seguridad avanzadas para evitar posibles inyecciones SQL y NoSQL.

Finalmente, este proyecto contribuyó a agilizar la labor de bomberos en terreno, demostrando cómo la tecnología puede modernizar la gestión de emergencias, reducir tiempos de respuesta y mejorar la toma de decisiones, incrementando así la seguridad tanto de los bomberos como de la comunidad.



7 Agradecimientos

Quisiera agradecer a toda mi familia por su constante apoyo durante estos años. También a mis amigos que me han acompañado desde el primer año de la carrera y a mi pareja por haber estado a mi lado. Finalmente quisiera agradecer a mi profesora guía, por su comprensión y orientación entregada en esta última etapa.

8 Referencias

- [1] RAE, «Emergencia,» [En línea]. Available: <https://www.rae.es/diccionario-estudiante/emergencia>.
- [2] Minvu, «Guía para elaborar un buen plan de emergencia,» [En línea]. Available: <https://www.minvu.gob.cl/wp-content/uploads/2020/09/Guia-para-elaborar-un-buen-Plan-de-Emergencia-Minvu.pdf>.
- [3] Biblioteca del Congreso Nacional de Chile, «Ley Chile – Ley N° 21.442: Ley de Copropiedad Inmobiliaria,» 13 Abril 2022. [En línea]. Available: <https://www.bcn.cl/leychile/navegar?idNorma=1174663>.
- [4] K. Schwaber y J. Sutherland, «La Guía de Scrum: Las Reglas del Juego,» Noviembre 2020. [En línea]. Available: <https://scrumguides.org/docs/scrumguide/v2020/2020-Scrum-Guide-Spanish-European.pdf>.
- [5] M. Kosinski, «¿Qué es una base de datos? | IBM,» [En línea]. Available: <https://www.ibm.com/mx-es/think/topics/database>.
- [6] A. Silberschatz, H. F. Korth y S. Sudarshan, Fundamentos de bases de datos, Quinta ed., Madrid: McGraw-Hill, 2006.
- [7] Google Developers, «Cloud Firestore,» 2025. [En línea]. Available: <https://firebase.google.com/docs/firestore>.
- [8] Docker Inc., «Docker Docs,» 2025. [En línea]. Available: <https://docs.docker.com/get-started/docker-overview/>.
- [9] IBM, «¿Qué es un sistema de información geográfica (SIG)?,» [En línea]. Available: <https://www.ibm.com/mx-es/think/topics/geographic-information-system>.
- [10] Smert Group, «¿Cómo trabajamos?,» [En línea]. Available: <https://smertgroup.com/como-trabajamos/>.
- [11] Viper, «VIPER GO,» [En línea]. Available: <https://viper.cl/viper-go>.
- [12] Waze Mobile, «Waze,» [En línea]. Available: <https://www.waze.com/>.
- [13] Esri, «ArcGIS,» [En línea]. Available: <https://www.esri.com/en-us/arcgis/geospatial-platform/overview>.
- [14] Oracle Corporation, «MySQL 8.0 Reference Manual,» 2025. [En línea]. Available: <https://dev.mysql.com/doc/refman/8.0/en/>.
- [15] IBM, «Formato JSON (JavaScript Object Notation),» 2025. [En línea]. Available: <https://www.ibm.com/docs/es/integration-designer/8.5.5?topic=formats-javascript-object-notation-json-format>.
- [16] Amazon Web Services, «¿Qué es ETL?,» 2025. [En línea]. Available: <https://aws.amazon.com/es/what-is/etl/>.
- [17] Google Developers, «Firebase Authentication,» 2025. [En línea]. Available: <https://firebase.google.com/docs/auth>.
- [18] National Fire Protection Association, «NFPA 704: Standard System for the Identification of the Hazards of Materials for Emergency Response,» 2022. [En línea]. Available: <https://www.nfpa.org/704>.