

**Implementación de un prototipo de una aplicación móvil para el reporte de incidentes de tránsito en términos de mejora de la movilidad de la ciudad de Medellín.**

**Daniela Vanessa Ospina Tuberquia**

**Institución Universitaria Pascual Bravo**

**Facultad de ingeniería**

**Tecnología en desarrollo de software**

**Medellín**

**2023**

**Implementación de un prototipo de una aplicación móvil para el reporte de incidentes de tránsito en términos de mejora de la movilidad de la ciudad de Medellín.**

**Daniela Vanessa Ospina Tuberquia**

**Trabajo de grado para optar al título de Tecnólogo en desarrollo de software**

**Asesores:**

**Liliana María García Aguirre**

**Magister en Ingeniería de Software**

**Juan Carlos Briñez de León**

**Doctor en Ingeniería de Sistemas**

**Institución Universitaria Pascual Bravo**

**Facultad de ingeniería**

**Tecnología en desarrollo de software**

**Medellín**

**2023**

## Tabla de contenido

Resumen.....	6
Abstract.....	7
Glosario.....	8
Introducción .....	10
1. Planteamiento del problema.....	12
1.1 Descripción .....	12
1.2 Formulación .....	12
2. Justificación .....	13
3. Objetivos .....	14
3.1 Objetivo general.....	14
3.2 Objetivos específicos .....	14
4 Marco teórico .....	15
4.1 Antecedentes .....	15
4.2 Bases teóricas.....	20
5 Metodología .....	25
5.1 Tipo de proyecto .....	25
5.2 Método .....	25
5.3.1 Fuentes primarias.....	30
6 Resultados .....	37
6.1 Análisis .....	38
6.1.1 Historias de Usuario.....	39
6.2 Diseño .....	40
6.2.1 Arquitectura .....	41
6.2.3 Diagrama de clases.....	43
6.2.4 Base de datos.....	44
6.2.5 Interfaz Gráfica .....	45
6.3 Implementación.....	48
6.4 Pruebas .....	81
7 Conclusiones .....	83
8 Recomendaciones .....	84

9 Bibliografía .....	85
10Anexos .....	88

## Lista de Figuras

Figura 1.	Principales aportaciones de la visión artificial.	21
Figura 2.	Big data.	22
Figura 3.	Semáforo con inteligencia artificial	23
Figura 4.	Plano Georreferenciado	24
Figura 5.	Tasa de crecimiento parque automotor circulante estimado	24
Figura 6.	Diseño inicial	40
Figura 7.	Diseño final	40
Figura 8.	Arquitectura del programa	41
Figura 9.	Diagrama de clases de la aplicación	42
Figura 10.	Diagrama base de datos	43
Figura 11.	Solicitud de permiso para el uso de la ubicación	44
Figura 12.	Ingresar información en el marcador	44
Figura 13.	Enviar mensaje de texto con el reporte del incidente y la ubicación en tiempo real	45
Figura 14.	Solicitud de permiso para el uso de la cámara	45
Figura 15.	Captura de fotografía	46
Figura 16.	Listado de reportes	46
Figura 17.	Marcador fijado en la ubicación exacta de un dispositivo físico	80
Figura 18.	Ubicación en tiempo real que será almacenada en la base de datos	80
Figura 19.	Envío de mensaje de prueba	81

## Lista de tablas

Tabla 1.	Tabla de presupuesto	27
----------	----------------------	----

## Lista de anexos

Anexo 1.	Manual de usuario	78
----------	-------------------	----

## Resumen

En este trabajo se encuentra la documentación del proceso de implementación de un prototipo de una aplicación móvil para el reporte de incidentes de tránsito en términos de mejora de la movilidad de la ciudad de Medellín. La aplicación permite a los usuarios activar la ubicación del usuario para agregar automáticamente la ubicación del incidente al reporte.

Para llevar a cabo este trabajo, se siguió un enfoque basado en el desarrollo ágil de software. Se realizó un análisis detallado de los requisitos del sistema, incluyendo la identificación de las funcionalidades principales y las necesidades de los usuarios. Además, se llevó a cabo una revisión de las aplicaciones similares disponibles en el mercado para identificar buenas prácticas y posibles mejoras.

El resultado final es una aplicación móvil intuitiva y fácil de usar que permite a los usuarios reportar diferentes tipos de incidentes de tráfico en tiempo real, como accidentes, congestión, obras viales y otros, solamente agregando una foto y descripción del incidente.

En cuanto al diseño y arquitectura se utilizó un enfoque orientado a objetos y se adoptó el estilo arquitectónico MVC (Modelo-Vista-Controlador) para garantizar la separación de responsabilidades y la mantenibilidad del código.

La implementación se realizó utilizando el lenguaje de programación Java, con el Framework de Android Studio y SQLite para la gestión de datos. Se realizaron pruebas de funcionalidad, usabilidad y rendimiento, y se corrigieron los errores y se optimizó el rendimiento según los resultados obtenidos.

En general, la aplicación móvil para el reporte de incidentes de tráfico es una herramienta valiosa para mejorar la movilidad en la ciudad de Medellín. Permite a los usuarios tomar decisiones informadas sobre su ruta, ayuda a las autoridades de tráfico a responder rápidamente a los incidentes y, en última instancia, mejora la eficiencia del tráfico en la ciudad.

## **Abstract**

In this work, you will find documentation on the implementation process of a prototype mobile application for reporting traffic incidents, aimed at improving mobility in the city of Medellin. The application allows users to enable their location, automatically adding it to the incident report.

To carry out this work, an approach based on agile software development was followed. A detailed analysis of the system requirements was performed, including the identification of key functionalities and user needs. Additionally, a review of similar applications available in the market was conducted to identify best practices and potential improvements.

The final result is an intuitive and user-friendly mobile application that enables users to report various types of real-time traffic incidents, such as accidents, congestion, roadworks, and others, simply by adding a photo and description of the incident.

Regarding design and architecture, an object-oriented approach was used, adopting the Model-View-Controller (MVC) architectural style to ensure separation of concerns and code maintainability.

The implementation was done using the Java programming language, with the Android Studio framework and SQLite for data management. Functionality, usability, and performance tests were conducted, and errors were corrected and performance was optimized based on the results obtained.

Overall, the mobile application for reporting traffic incidents is a valuable tool for improving mobility in the city of Medellin. It allows users to make informed decisions about their routes, helps traffic authorities respond quickly to incidents, and ultimately enhances traffic efficiency in the city.

## Glosario

**Aplicación móvil:** Una aplicación diseñada para ser utilizada en dispositivos móviles, como teléfonos inteligentes o tabletas, que permite a los usuarios reportar incidentes de tránsito y acceder a información sobre la movilidad en la ciudad.

**Condiciones viales:** Información sobre el estado de las carreteras y vías de la ciudad, incluyendo factores como la congestión, el clima, obras en la vía y cualquier otro elemento que pueda afectar la movilidad.

**Colaboración de usuarios:** La capacidad de los usuarios de interactuar entre sí a través de la aplicación, compartiendo información, alertas y consejos sobre incidentes de tránsito, lo que fomenta una comunidad de conductores y usuarios comprometidos con la mejora de la movilidad.

**Geolocalización:** Tecnología que permite determinar la ubicación geográfica exacta de un dispositivo móvil utilizando señales de GPS, Wi-Fi o torres de telefonía celular. En el contexto de la aplicación, se utiliza para marcar y localizar incidentes de tránsito en un mapa.

**Interfaz de usuario:** La presentación visual y la interacción que los usuarios tienen con la aplicación, incluyendo elementos como botones, menús, formularios y mapas, que facilitan la navegación y el uso de la aplicación.

**Mejora de la movilidad:** El objetivo principal de la aplicación, que busca generar un impacto positivo en la fluidez del tráfico, la reducción de incidentes y la optimización de los desplazamientos en la ciudad de Medellín.

**Notificaciones en tiempo real:** Mensajes o alertas que se envían a los usuarios de la aplicación de forma inmediata y actualizada sobre incidentes de tránsito, cambios en el estado de las vías o rutas alternativas.

**Privacidad de datos:** Protección de la información personal y confidencial de los usuarios, garantizando que los datos recopilados en la aplicación sean tratados de forma segura y



utilizados únicamente con fines relacionados con la mejora de la movilidad y la seguridad vial.

**Reporte de incidentes:** La acción de informar sobre cualquier evento o situación que afecte la circulación y seguridad del tráfico, como accidentes, congestiones, obras en la vía, semáforos dañados, entre otros.

**Rutas alternativas:** Trayectos o caminos alternativos propuestos a los usuarios para evitar zonas con congestión o incidentes de tránsito, proporcionando opciones más eficientes para desplazarse por la ciudad.

## Introducción

Este trabajo está enfocado en la investigación y creación de una aplicación móvil para reportar los problemas por movilidad en la ciudad de Medellín, con el fin de que las personas se enteren y eviten transitar por el sector en ese momento y así no incrementar la congestión.

En la actualidad, la movilidad urbana es un desafío constante para muchas ciudades en todo el mundo, y Medellín no es una excepción. El tráfico, los accidentes y otros incidentes de tránsito pueden tener un impacto significativo en la eficiencia del transporte público y privado, lo que a su vez afecta la calidad de vida de los ciudadanos.

Para abordar este problema, se ha desarrollado un prototipo de aplicación móvil para el reporte de incidentes de tránsito en la ciudad de Medellín. La aplicación permite a los usuarios reportar incidentes en tiempo real y compartir información valiosa sobre el estado del tráfico en la ciudad.

En la etapa de levantamiento de requisitos, se realizaron entrevistas y encuestas para comprender las necesidades y los desafíos de los usuarios en términos de movilidad y reporte de incidentes de tránsito en Medellín. Esto permitió identificar las funcionalidades y características clave que debían ser incorporadas en la aplicación.

Posteriormente, se realizó un análisis detallado de los requisitos recopilados, donde se identificaron los casos de uso y se definieron los escenarios de uso de la aplicación. Esto permitió establecer los flujos de trabajo y las interacciones entre los usuarios y el sistema, así como también identificar los posibles riesgos y desafíos técnicos.

Con base en el análisis, se procedió al diseño de la aplicación, definiendo la arquitectura general, la interfaz de usuario, la estructura de la base de datos y los servicios necesarios. Se crearon prototipos y se realizaron pruebas de usabilidad para garantizar una experiencia de usuario intuitiva y eficiente.

Luego, se pasó a la implementación, donde se desarrolló el código fuente de la aplicación siguiendo las mejores prácticas de desarrollo de software. Se utilizó un enfoque ágil, como la

metodología SCRUM, para dividir el proyecto en iteraciones cortas llamadas Sprints. Cada Sprint se enfocó en implementar un conjunto específico de funcionalidades y se llevó a cabo en colaboración con el equipo de desarrollo.

Finalmente, se realizaron pruebas exhaustivas para garantizar la calidad y estabilidad de la aplicación. Se llevaron a cabo pruebas funcionales, pruebas de rendimiento y pruebas de usabilidad para identificar y solucionar cualquier problema o error antes del lanzamiento final de la aplicación.

En resumen, el proyecto siguió una metodología que abarcó desde el levantamiento de requisitos y el análisis, pasando por el diseño, la implementación y las pruebas, con una fuerte orientación hacia la colaboración y la entrega iterativa. Esto permitió desarrollar una aplicación móvil efectiva y funcional para el reporte de incidentes de tránsito en Medellín, mejorando así la movilidad y la seguridad vial en la ciudad.

# 1. Planteamiento del problema

## 1.1 Descripción

El tráfico de la ciudad de Medellín, ha venido incrementando con el paso de los años y a su vez, ha ido dificultando la movilidad, para diferentes actores viales, causando pérdidas de tiempo, accidentes de tránsito, y otros factores, ya que según el periódico “El Colombiano”, Medellín actualmente es una de las 25 ciudades del mundo más congestionadas (El Colombiano, (2021), Medellín, entre las 25 ciudades más congestionadas. El colombiano), creando demasiados retrasos para las personas, aunque la ciudad cuenta con metro, este también se ha visto afectado en diferentes maneras, por el tráfico y también ha retrasado a muchas personas.

## 1.2 Formulación

¿Se podrá demostrar a través de una aplicación móvil como puede mejorar la movilidad de la ciudad de Medellín a partir de los reportes de accidentes?

- ¿Se ha intentado implementar antes una solución tecnológica al problema?

## 2. Justificación

Actualmente, Medellín es una de las ciudades con peor movilidad, no solo de Colombia, sino del mundo, además de esto la movilidad se ve afectada en su medio de transporte, el cual por fallas mecánicas o bien, por el gran número de personas que diariamente utilizan este, colapsan el sistema retrasando la movilidad del mismo. También, en diferentes lugares de la ciudad el tráfico no fluye con normalidad, teniendo que esperar horas para el desplazamiento por este, motivo por el cual las personas suelen usar rutas alternas u otros medios para llegar a tiempo.

Las innovaciones con las cuales contaremos en la aplicación es el uso de algoritmos de visión artificial, la cual nos permitirá hacer pronósticos de las rutas que estarán congestionadas gracias a las imágenes que procesó anteriormente, como el conocer qué rutas se suelen congestionar más a menudo, o si la tasa de accidentes de una calle principal aumenta por la lluvia, este pronóstico permitiría advertir a los usuarios sobre tener más precaución en el lugar, además de darle a conocer a la ciudad este tipo de problemas, permitiéndole a la alcaldía el investigar por qué ocurre esto, ya sea por velocidad, el estado de vías, etc.

Otro punto fundamental es que no se encontró información sobre una aplicación similar en el Valle de Aburrá, la cual permita el monitoreo y brinde información en tiempo real del estado de las vías, además de conocer los lugares que están siendo más concurridos, esto con el fin de tomar mejores decisiones como el ampliar una vía, construir puentes, viaductos, o abrir nuevas rutas de buses, el brindar alternativas que permitan no únicamente una mejor movilidad, sino también aportar a la mejora del medio ambiente, debido a que la ciudad sufre de contaminación vehicular y medidas como el pico y placa no llegan a funcionar lo suficiente, o son fácilmente reemplazadas por un pago.

### **3. Objetivos**

#### **3.1 Objetivo general**

Implementar un aplicativo móvil que contribuya a la mejora de la movilidad en la ciudad de Medellín, a través de un reporte de incidentes de tránsito.

#### **3.2 Objetivos específicos**

Realizar la elicitación de requisitos por medio de encuestas y entrevistas a los usuarios de medios de transporte y personas relacionadas como conductores, agentes de tránsito entre otros.

Identificar la metodología apropiada para desarrollar la aplicación de acuerdo a la disponibilidad de tiempo, esto incluye la arquitectura, lenguaje, base de datos entre otros.

Codificar el prototipo de la app de transporte, utilizando una arquitectura acorde a las necesidades y el Android Studio para que funcione en todos los dispositivos de los usuarios interesados.

Realizar pruebas técnicas y de usuario a la funcionalidad de la aplicación y a la interfaz, buscando que sea intuitiva y fácil de usar por el usuario del común, quienes utilizan los diferentes medios de transporte.

## 4 Marco teórico

### 4.1 Antecedentes

El artículo de la revista “Tecnologías para luchar contra la pandemia Covid-19: geolocalización, rastreo, BIG DATA, SIG, inteligencia artificial y privacidad.” Tiene como propósito brindar información sobre otros proyectos que salieron a flote durante la pandemia en diferentes lugares del mundo en los cuales se podrían realizar actividades de autodiagnóstico, geolocalización de contagiados y estudios de movilidad para la atención prioritaria y la descongestión de la línea de emergencia a través del uso de chatbots. Tiene relación con la problemática a tratar dado que, en los proyectos mencionados se habla de aspectos como el geoposicionamiento el cual hace referencia a la ubicación de una persona u objeto en la superficie terrestre, que sería una de las principales implementaciones que se debería hacer al momento de desarrollar una aplicación que ayude a evitar el tráfico vehicular, puesto dicho programa está en la obligación de ubicar y reconocer dónde se presenta un incidente de tránsito, es decir aunque estos proyectos no están directamente orientados a los servicios de tráfico y movilidad pueden ser una guía a la cual se le harían unas cuantas modificaciones con el fin de orientar el proyecto hacia los temas de transporte en el área metropolitana. (Cascón Katchadourian, 2020, #)

En el trabajo de grado “Desarrollo de una herramienta de evaluación incorporando algoritmos de clusterización de tráfico vehicular con parámetros de rendimiento, infraestructura usada, procesamiento y resultados obtenidos de la geolocalización de los algoritmos implementados usando herramientas de Big Data sobre infraestructura Cloud.” Esta arquitectura se ha creado con el fin de optimizar el procesamiento de información a gran escala en el menor tiempo posible y presentar información veraz a partir de la implementación de algoritmos de trayectoria en una aplicación tecnológica para la ayuda en la planeación urbana, ya que muchas ciudades enfrentan problemas por congestión vehicular. Se relaciona con nuestra problemática porque busca ayudar a mejorar la movilidad en diferentes ciudades implementando información de georeferencia en la base de datos bigquery de google cloud, con la finalidad de mostrar información veraz sobre los lugares en los que haya mayor cantidad de tráfico vehicular.

(Hidalgo Arévalo & Torres Macio, 2020)

En el artículo de investigación científica “Acceso al dato. BIG DATA en aplicaciones móviles de geolocalización y experiencia de usuarios.” Se habla sobre la recolección de información de la experiencia de usuario y hace foco en la diversidad de percepciones de movilidad (funcionamiento, discapacidad y rango etario). Este artículo se enfoca en los datos masivos que generan algunas aplicaciones como Instagram, YouTube, el buscador de google, etc. Con lo cual quieren hacer énfasis en las bases de datos y la cantidad de volúmenes de información que almacenan, con el fin de personalizar la experiencia de cada usuario. Por lo que busca asignar una categoría a la cantidad de datos de ámbito público dentro de los que cabe mencionar fotografías, medios audiovisuales y algunos escritos, los cuales generan las aplicaciones de geolocalización como Google Maps. Se conecta con nuestra temática puesto que la cantidad de información que se maneja en el sector de transporte de una ciudad es demasiado alta, por lo cual tener una orientación que se enfoque en los grandes volúmenes de datos es de vital importancia para clasificar el tipo de evento que está causando el tráfico vehicular en nuestro caso. (URROZ & ROMERO, 2020, #)

De acuerdo con el trabajo de grado “Inteligencia artificial y competencia, acceso a datos en el ejemplo de la movilidad y del vehículo autónomo”, la evolución en cuanto a temas de movilidad avanza con gran fuerza, pero el gran reto radica en el control de la información que arrojan los vehículos automáticos, los cuales van de la mano de la inteligencia artificial por lo cual el objetivo principal de este es buscar la posibilidad de implementar esta IA no solo para autos de uso particular, sino también para el transporte público. Este trabajo se relaciona directamente con nuestra problemática, dado que esta inteligencia artificial busca ser aplicada en gran parte del sector de movilidad, tanto particular como público, y hace hincapié en la administración de datos de un sistema móvil autónomo, que tendrá la capacidad de tomar decisiones de navegación, lo cual sería de gran utilidad para nuestro proyecto porque encierra la posibilidad de utilizar esa misma inteligencia en medios visuales de gran cobertura que notifiquen los incidentes de tránsito. (SUDEROW, 2021)

El artículo del libro científico “A Survey for Mobility Big Data Analytics for Geolocation



Prediction”, el cual se refiere a una encuesta para la movilidad sobre Big Data Analytics para la predicción de geolocalización, tiene como objetivo verificar las soluciones existentes para pronósticos de posicionamiento geográfico. En primer lugar, expone los conceptos, clasificación y características de las predicciones de posicionamiento geográfico, Luego, habla sobre los principios básicos y las características de la exploración en las áreas de la navegación geográfica popular y de la trayectoria personal. Este artículo nos sirve como apoyo para la elaboración de nuestro trabajo dado que explica de forma amplia diferentes aspectos que encierra la geolocalización en conjunto con las bases de datos y también analiza desafíos y oportunidades futuros en cuanto a temas de movilidad. (Yanbing Liu et al., 2016, #)

El artículo “Sistema de navegación y exploración visual articulado por técnicas de inteligencia artificial, para la movilidad de personas en condición de discapacidad visual” tiene como objetivo explicar el funcionamiento de unas gafas con inteligencia artificial, las cuales cuentan con cámaras, encargadas del procesamiento de imágenes, para las personas con movilidad reducida, estas pueden ser una base y demostración del procesamiento de imágenes en entornos de movilidad. Este artículo está relacionado con nuestro tema, debido a la implementación de una inteligencia artificial que procesa imágenes, para crear mapas, en este caso con personas y movilidad, nosotros lo usamos como base a nuestra red neuronal, la cual identificará, tráfico o accidentes en la movilidad, algo cercano a lo que hace este proyecto con sus gafas. (Aldana Porras et al., 2016, #)

El artículo “¿Cómo puede la inteligencia artificial reducir los accidentes de tráfico y evitar atascos?” de la revista Impacto social, habla de dar a conocer el proyecto “Timón”, el cual es una inteligencia artificial en forma de ecosistema, el cual permite integrar, tanto personas, como empresas e infraestructuras, para mejorar la movilidad de las ciudades, evitar accidentes todo esto gracias al manejo de Big data y la misma inteligencia artificial, este trabajo se relaciona con el nuestro, debido a que utiliza un ecosistema y crea una unión de diferentes tecnologías para lograr su objetivo, en nuestro trabajo utilizaremos aspectos relacionados, como el unir la base de datos, con la red neuronal y adicional a esto hacerlo en una aplicación móvil, en la cual le brinde información al usuario sobre lo que sucede en algún lugar de la ciudad que cuente con esta tecnología, ya que como se explica en el artículo también puede ser con infraestructuras

relacionadas a este proceso. (Serrano et al., 2019, #)

El trabajo de grado, titulado “Modelo predictivo en los accidentes de tránsito con base en data science” creado por Iván Silva Feraud y Jean Jordy Segarra Arias, menciona el uso de big data para monitorizar los accidentes de tránsito, estos dependiendo de la hora, lugar y fechas en las que ocurren más accidentes o embotellamientos. Este trabajo se relaciona en gran manera al nuestro, ya que este habla de que, mediante el uso de un software estadístico e inteligencia artificial, crear un sistema de predicción de accidentes de tráfico, congestiones, u otras incidencias viales, mediante la probabilidad y estadística, esto permite calcular si en la calle, a la misma hora habrá un trancón, se podrían tomar rutas alternas, lo cual beneficiaría a nuestro aplicativo el conocer sobre esto mismo. (Segarra Arias, 2020)

Este artículo de revista, titulado cómo “Collective Intelligence and the Mapping of Accessible Ways in the City: a Systematic Literature Review”, creado por Alexandre Reis y Valmir Luiz, habla sobre la inteligencia colectiva y el manejo de Big data para el mismo, el cual permite que varios usuarios alimenten de información la base de datos, esto mediante recompensas monetarias, rankings o bien en forma de videojuego, para incentivar la inteligencia colectiva. Se relaciona con nuestro proyecto debido a que serán los mismos usuarios quienes informaran sobre los estados de las vías, de manera que la base de datos estaría alimentada de la información de los mismos usuarios, así que podríamos tomar ideas e iniciativas de este como el manejo de la inteligencia artificial a estos mismos datos, también el identificar información, falsa o verdadera dentro de la base de datos que nos permitirá tomar mejores decisiones. (Marques & Graeml, 2017, #)

El artículo de revista científica, titulado “Overview of Road Traffic Management Solutions based on IoT and AI”, creado por Asma Ait, Ayoub, Assia y Mohamed, habla sobre el uso de la inteligencia artificial, la cual mediante el uso de esta permite el cambio de semáforos, manejo de vías e información sobre el tráfico en tiempo real de las mismas. En nuestro se relaciona en la medida que podremos conocer diferentes puntos flexibles de la movilidad de Medellín, y mediante la inteligencia artificial descubrir si es posible el detectar la congestión de la movilidad dependiendo de la zona y arrojar otras rutas, ya que en este proyecto se menciona el

posible cambio automático de semáforos, dependiendo de la cantidad de autos que pasen por allí cada x tiempo permitiendo una movilidad más dinámica ya que el flujo cambiaría dependiendo de la hora y cantidad de autos. (Ouallanea et al., 2021, #)

El trabajo de grado se titula “Aplicabilidad de herramientas de Big Data y Business Analytics al sector de la eco movilidad urbana: El caso BiciMAD.” realizado por Ignacio, Lopez el cual nos cuenta de cómo el uso del big data se utiliza en un sistema de préstamo de bicicletas eléctricas, que permiten ser monitoreadas para así obtener la información del uso de estas, la movilidad en los tramos y el manejo de estas dentro de su propio sistema, podemos encontrar que se relaciona en el uso de la recolección de datos dentro de una big data, la cual permitirá conocer dónde se podría concentra algunos puntos de movilidad ya que se tendría en cuenta el rastreo de caminos que los usuarios recorrerán en la ciudad de Medellín. Este trabajo también nos habla del aumento del uso de bicicletas durante la pandemia, lo cual cambió la movilidad y se obtuvieron nuevos registros que permitieron la información en la base de datos. (López-Linares, 2020)

El trabajo de grado “Importancia de la solución Big Data en la aplicación de movilidad Uber movement.” realizado por Montealegre, Andrés la cual se basa en el estudio de la aplicación Uber movement, el cual que por medio del uso de big data para la monitorización y el registro de patrones en la movilidad de la ciudad de Bogotá, esto con el fin de mejorar los servicios de Uber y de la movilidad, encontrando picos y patrones en ella. La relación que podemos encontrar con este trabajo es que por medio de patrones se puede realizar un registro de los lugares donde más se presenta la congestión en la movilidad de Medellín con el uso de big data e inteligencia artificial para así ofrecer rutas alternas a las personas, además de esto nos podría servir de base ya que la misma aplicación de Uber utiliza un algoritmo que cambia los precios dependiendo del tráfico, condiciones climáticas y personas que utilizan el servicio mediante una gran base de información. (Montealegre Gallo, 2017)

El trabajo de grado “El origen y la evolución de la movilidad de las personas con la utilización de las nuevas tecnologías” que fue realizado por Bandrés Balet, Cristina nos habla de que, mediante el uso de la IA, Big Data, e IoT, se puede conocer la evolución del transporte

urbano, buscar transformaciones y ver si es posible aplicarlas independientemente del tamaño de la ciudad o sus habitantes, creando un prototipo para cualquier ciudad. Podemos encontrar relación con este trabajo porque a partir de él podemos obtener información de cómo el área metropolitana ha evolucionado en su sistema de transporte además de investigar y tener en cuenta los posibles cambios que pueden llegar a ocurrir en un futuro y así coleccionar esto en una base de datos que será útil para pronosticar futuros cambios, como por ejemplo, una calle nueva o un lugar donde transiten muchos más vehículos permitiendo una mejor gestión de la movilidad en años futuros, como ejemplo que se generen rutas alternas. (Bandrés Balet, 2019).

En el trabajo de grado “Sistema experto para la gestión de estrategias de movilidad en el municipio de Itagüí” que fue realizado por Salazar, Yuliana y Calle, Daniel habla de reducir y mejorar la movilidad mediante un sistema experto, el cual estaría alimentado por cámaras de seguridad y que mediante paneles informativos, y optimización de semáforos podrá tomar mejores decisiones para evitar accidentes o incidencias en la movilidad del valle de Aburrá, esto permitirá el seguir un plan en caso de contingencia vehicular o diferentes factores que puedan afectar ello. Podemos encontrar relación en que cuenta con un sistema de inteligencia artificial que se encuentra integrado con la ciudad por lo cual nos sirve de base a una futura integración con varios elementos de la ciudad para la mejora constante de la misma, como por ejemplo un semáforo que dependiendo de la movilidad cambien cada cierto tiempo. (Salazar Montoya & Calle Vargas, 2018)

## **4.2 Bases teóricas**

### **Procesamiento digital de imágenes:**

El procesamiento digital de imágenes ha adquirido, en años recientes, un papel importante en las tecnologías de la información y el cómputo. Actualmente, es la base de una creciente variedad de aplicaciones que incluyen diagnóstico médico, percepción remota, exploración espacial, visión por computadora, etc. Como resultado directo de la reducción en el precio de las computadoras, el procesamiento digital de imágenes actualmente se puede efectuar

(aunque con ciertas limitantes) en una computadora personal.

Al conjunto de técnicas y procesos para descubrir o resaltar información contenida en una imagen usando como herramienta principal una computadora se le conoce como procesamiento digital de imágenes (PDI). Hoy en día, el PDI es un área de investigación muy específica en computación y está muy relacionada con el procesamiento digital de señales. Esta relación estriba en el hecho de que en esencia el PDI es una forma muy especial del procesamiento digital de señales en dos o tres dimensiones. (Domínguez Torres, 2022)

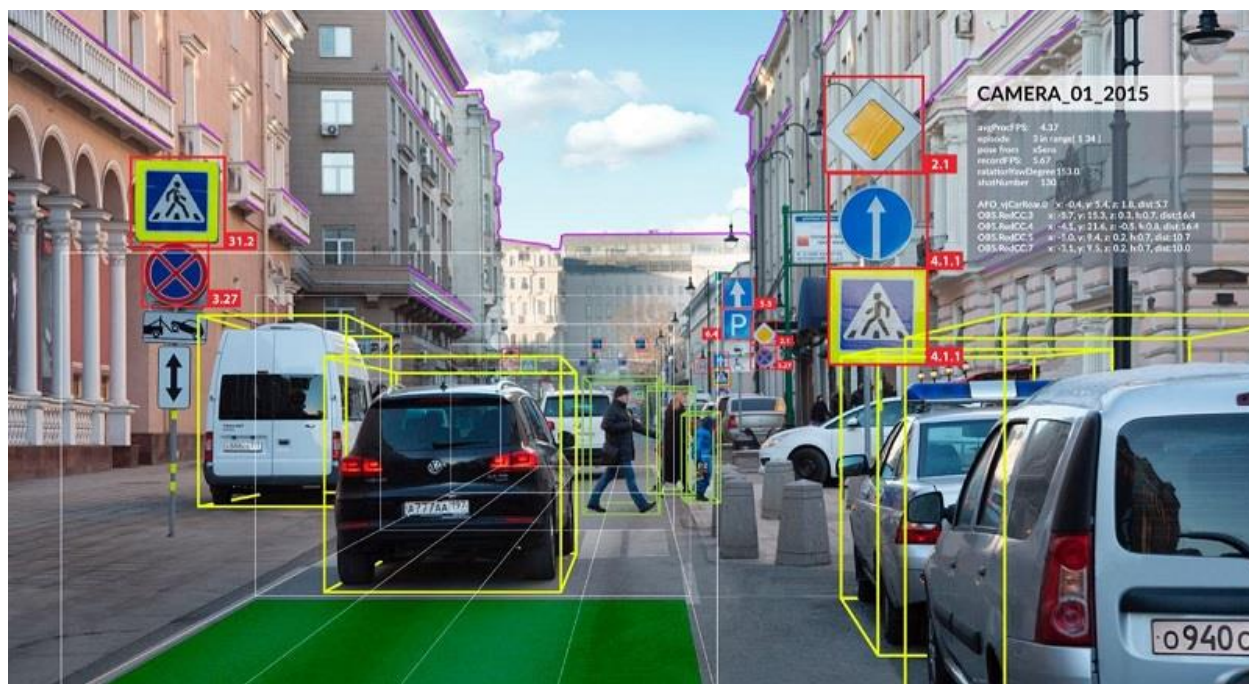


Figura 1. Principales aportaciones de la visión artificial.

Imagen tomada de: (Gavilán, 2020)

**Manejo de datos:** en los últimos años la gran expansión de la ciudad ha aumentado considerablemente y su recolección de datos se ha vuelto aún más complicada con lo que las formas de la recolección de datos han ido cambiando, es por esto que una base de datos común no tendría la suficiente capacidad para almacenar un gran volumen de datos de una manera actualizada, es por eso que la implementación de big data ofrece un mayor control al alto flujo de

datos, permitiendo obtener una información actualizada, y de una manera más fácil y rápida (Mario Saffiro C.(2018). Big Data)

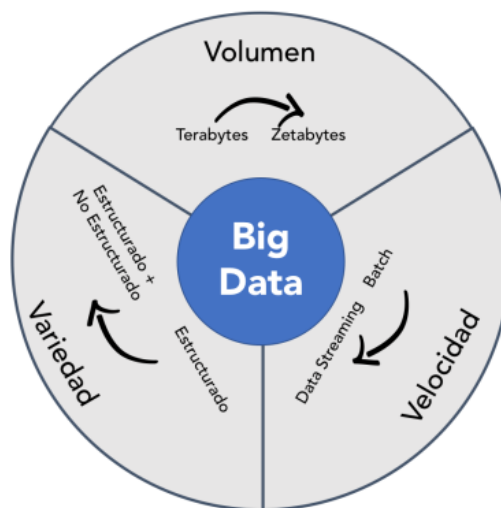


Figura 2. Big data.

Imagen tomada de: (Mario Saffiro C. (2018))

**Integración con sistemas de movilidad de la ciudad:** Es un método de implementación que se puede utilizar para la integración de distintos sistemas de movilidad, como lo es el caso de semáforos, cámaras de seguridad, sensores de velocidad, entre otros. Este método ha llegado a ser implementado en otros países, como es el caso de Argentina que por medio de la inteligencia artificial aplicada en un semáforo permite verificar que cada motociclista lleve puesto el casco, provocando que este no cambie a verde a menos de que el motociclista lleve su casco. (ACV. (2021). El semáforo detecta si un conductor no lleva casco y no se pone verde).



Figura 3. Semáforo con inteligencia artificial  
 Imagen tomada de: (ACV. (2021))

**Georreferenciación:** Esta es una técnica, la cual tiene un conjunto de parámetros cartográficos los cuales permiten hallar, localizar o ubicar, un punto o un objeto espacial, ya sea en su aspecto de vector como rastres, estos parámetros permiten tener una cercanía matemática a hallar este punto de una manera cercana a la realidad, como por ejemplo el hallar la torre Eiffel, una casa, una carretera en específico a nivel mundial, esta tecnología, permite también el usar varios medios para hallar con el punto que se quiere encontrar, ya sea de manera global o de manera local.

La telefonía móvil es uno de los mercados que más utilizan la georreferenciación, como es el caso de usar una aplicación GPS, un videojuego con posicionamiento, entre otras, Google es el más famoso proveedor de este servicio de georreferenciación, debido a su experiencia y fama con Google Maps, o Google Earth. (Arkespai. (S.f). Plano Georreferenciado)

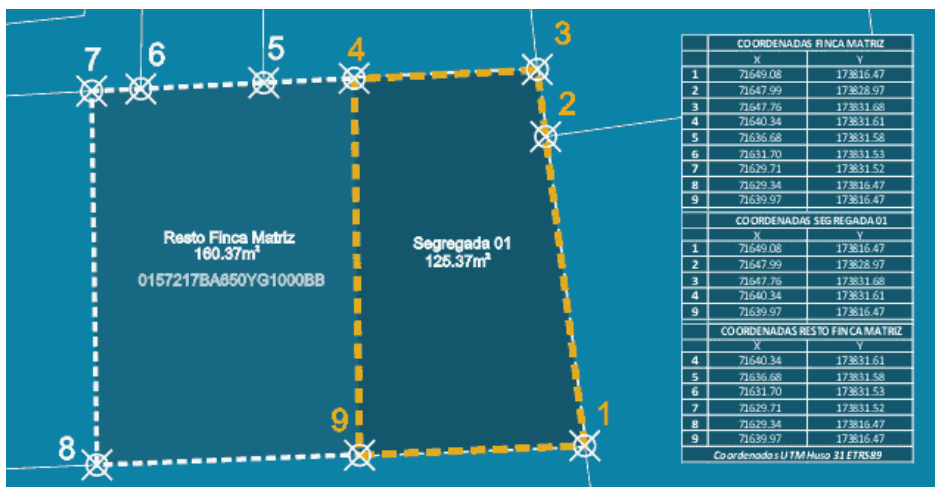


Figura 4. Plano Georreferenciado  
 Imagen tomada de: (Arkespai. (2022))

**Congestión en la ciudad:** cabe destacar que actualmente se vive una crisis en la movilidad debido al aumento de vehículos que desplazan por la ciudad, esto provoca que se pase en promedio 20 días al año en un trancón, afectando gravemente la economía haciendo que se pierda un 2% de PIB lo que equivale a 16 billones de pesos colombianos.

En el caso de la ciudad de Medellín se encuentra ubicada en el puesto 22º, entre las ciudades con más congestión vehicular. (*DNP Advierte Que Se Avecina Colapso De Movilidad En Las Principales Capitales.* / DNP, 2015)

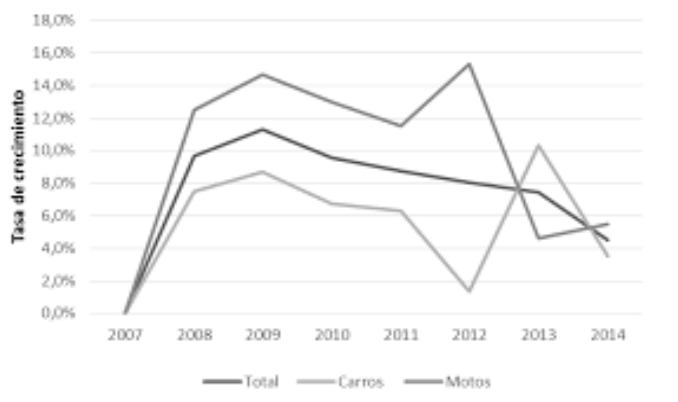


Figura 5. Tasa de crecimiento parque automotor circulante estimado  
 Fuente: (Corrales Espinosa. (2016))



## 5 Metodología

### 5.1 Tipo de proyecto

Para el desarrollo de este proyecto se cuentan con dos tipos de tipología, siendo una de ellas de tipo exploratoria, porque se investigará sobre la visión artificial, y se buscará formas de cómo podría aplicarse en el proyecto, otra tipología sería aplicado porque se busca diseñar prototipos donde se implementen bases de datos y la información que se obtiene a partir de la inteligencia artificial

### 5.2 Método

De acuerdo con la problemática presentada anteriormente y la necesidad de darle solución, a continuación, se presentan los pasos a seguir para cumplir con la meta propuesta.

Inicialmente se hará la recolección de información a través de visitas al departamento de tránsito, en las cuales se realizarán entrevistas a los agentes, donde el tema principal será el estado de la movilidad en la ciudad, adicionalmente se entrevistará a diferentes individuos sobre los medios de transporte masivo que usan y su opinión respecto a tráfico vehicular de la ciudad. Con la realización de esta actividad se obtendrán datos que nos permitirán buscar las falencias en los sistemas de transporte, para el posterior levantamiento de requerimientos.

Continuando con el proyecto y teniendo los requisitos necesarios se dará inicio al diseño y codificación del prototipo, en el cual se integrarán los algoritmos de inteligencia artificial y las bases de datos. Una vez dicho sistema esté listo, se realizarán actividades para la optimización y mejora del mismo, a partir de la corrección de fallos, manejo de datos y rectificación de la inteligencia artificial.

Para el desarrollo de la app, se utilizará la metodología SCRUM, la cual funciona de la siguiente forma:

Se necesita desarrollar una aplicación que ayude a mejorar la movilidad de la ciudad a partir del uso de un Android Studio, las API's de Google Maps y una base de datos, con el fin de poder realizar el reporte de incidencias de tránsito y que estas puedan ser vistas por los demás usuarios para que puedan tomar una vía alterna.

Con base en la información recolectada a partir de preguntas realizadas a personas que utilizan transporte público, particular, entre los cuales están las motocicletas y los automóviles, conductores de medios de transporte, entre otros, sobre que les gustaría que tuviera una aplicación para mejorar la movilidad de la ciudad, de la cual se obtuvieron los pilares o requisitos necesarios para iniciar con el desarrollo del proyecto

Se pudieron extraer las siguientes pilas, incidencias, rutas, marcadores y fotos. las cuales se encargarán de funciones específicas para el funcionamiento de la aplicación.

Las necesidades son compartidas con la persona encargada del desarrollo de dicha aplicación para ejecutar el plan de acción.

Tabla 1.

Tabla de presupuesto

<b>Recursos</b>	<b>En dinero</b>	<b>En especie</b>	<b>Total</b>
Google Cloud	1000000		1000000
Servidores	1000000		1000000
Bases de datos	840000		840000
Programador	18000000		18000000
Computadores		5250000	5250000

Internet	480000	480000
<b>Total</b>	<b>20840000</b>	<b>5730000</b>
		<b>21845000</b>

#### Pilas del SPRINT

Cargar el mapa

Mostrar el marcador con los incidentes

Mostrar la ubicación en tiempo real

Hacer la conexión a la base de datos

Replicar la información a los dispositivos cercanos

Comienza entonces el trabajo real y se convoca con la mayor frecuencia posible una reunión de equipo donde se comentan las tareas realizadas el día anterior y las que se van a realizar a continuación.

Después de la finalización de todas las pilas que componen un Sprint, el cliente ya puede ver el resultado de una pila de producto. De esta forma el cliente ya tiene un primer contacto con el producto final y puede dar su opinión. Además, puede reordenar de nuevo la pila de producto si así lo desea.

#### Sprint 1: Objetivo: Cargar el mapa Pila del Sprint:

1. Configurar la biblioteca de mapas y geolocalización.
2. Obtener los datos geográficos necesarios para mostrar el mapa.
3. Crear la interfaz de usuario para la visualización del mapa.
4. Implementar la funcionalidad de zoom y desplazamiento del mapa.
5. Realizar pruebas de integración para asegurar el correcto funcionamiento de la carga del mapa.

#### Sprint 2: Objetivo: Mostrar el marcador con los incidentes Pila del Sprint:

1. Obtener los datos de los incidentes de la base de datos.
2. Definir los iconos o símbolos a utilizar para representar los diferentes tipos de incidentes.
3. Implementar la funcionalidad para agregar los marcadores en el mapa, correspondientes a cada incidente.
4. Establecer interacciones para mostrar información detallada de los incidentes al hacer clic en los marcadores.
5. Realizar pruebas de rendimiento y corrección de errores en la visualización de los marcadores.

Sprint 3: Objetivo: Mostrar la ubicación en tiempo real Pila del Sprint:

1. Implementar la funcionalidad para obtener la ubicación en tiempo real del dispositivo.
2. Actualizar la posición en el mapa a medida que se va registrando un cambio en la ubicación.
3. Establecer un intervalo de actualización para evitar sobrecargar el sistema con actualizaciones constantes.
4. Configurar notificaciones o alertas visuales para indicar al usuario que su ubicación está siendo rastreada en tiempo real.
5. Realizar pruebas exhaustivas de la funcionalidad de seguimiento de ubicación y corregir posibles errores o problemas de precisión.

Sprint 4: Objetivo: Hacer la conexión a la base de datos Pila del Sprint:

1. Configurar los parámetros de conexión a la base de datos, como la dirección IP, el puerto y las credenciales.
2. Establecer una conexión segura utilizando los protocolos y métodos de autenticación adecuados.
3. Implementar consultas de prueba para verificar la conexión y asegurarse de que se pueda acceder correctamente a la base de datos.
4. Diseñar e implementar la estructura de la base de datos para almacenar la información de los incidentes.

5. Realizar pruebas de integración para comprobar la funcionalidad de la conexión a la base de datos y corregir cualquier error o problema identificado.

Sprint 5: Objetivo: Replicar la información a los dispositivos cercanos Pila del Sprint:

1. Diseñar e implementar un sistema de comunicación entre dispositivos cercanos, como Bluetooth o redes de área local (LAN).
2. Establecer un mecanismo de descubrimiento de dispositivos cercanos para identificar aquellos con los que se puede compartir información.
3. Implementar la funcionalidad para enviar y recibir datos relevantes de los incidentes entre dispositivos cercanos.
4. Realizar pruebas de comunicación entre dispositivos para verificar la correcta replicación de la información.
5. Optimizar el rendimiento y la seguridad del sistema de replicación de datos, y corregir cualquier problema o error identificado durante las pruebas.

Cabe destacar que los Sprints son iterativos y adaptativos, por lo que la duración de cada uno puede variar según las necesidades y la complejidad de las tareas. Además, es recomendable contar con un equipo de desarrollo y un Product Owner que priorice las tareas y brinde retroalimentación continua durante el proceso.

### **5.3 Población y muestra**

La aplicación para el reporte de incidentes de tránsito en Medellín está dirigida a la población en general que residen o transitan por la ciudad de Medellín, Colombia. La muestra objetivo de esta aplicación incluye a conductores de vehículos particulares, usuarios de transporte público, ciclistas y peatones que deseen informar sobre incidentes de tráfico y contribuir a mejorar la movilidad en la ciudad.

La población objetivo abarca a personas de diferentes edades, géneros y ocupaciones que se encuentren activamente involucradas en el tráfico y la movilidad de Medellín. Esto incluye a residentes, trabajadores, estudiantes y turistas que utilizan las vías de la ciudad.

La aplicación busca ser accesible y fácil de usar para todos los usuarios, independientemente de su nivel de experiencia tecnológica. Su objetivo es involucrar a una amplia muestra de la población para recopilar información relevante sobre los incidentes de tránsito y promover una mayor conciencia y colaboración en la mejora de la movilidad en Medellín.

### **5.3.1 Fuentes primarias.**

#### **Encuesta: Aplicación móvil para reporte de incidentes de tránsito y mejora de la movilidad en Medellín**

1. ¿Con qué frecuencia utilizas aplicaciones móviles relacionadas con el tránsito o la movilidad en la ciudad de Medellín?
  - a) Diariamente
  - b) Varias veces a la semana
  - c) Ocasionalmente
  - d) No utilizo aplicaciones móviles relacionadas con el tránsito
  
2. ¿Estarías interesado en utilizar una aplicación móvil específica para reportar incidentes de tránsito en Medellín?
  - a) Sí, definitivamente
  - b) Tal vez, dependiendo de sus características
  - c) No, no estaría interesado(a)
  - d) No estoy seguro(a)
  
3. ¿Qué características o funcionalidades consideras importantes en una aplicación

móvil para reportar incidentes de tránsito? (Selecciona todas las opciones que apliquen)

- a) Fácil y rápida interfaz de reporte de incidentes
- b) Integración con mapas y geolocalización
- c) Notificaciones en tiempo real sobre incidentes en rutas frecuentes.
- d) Información actualizada sobre el estado del tráfico y las condiciones viales
- e) Opción para adjuntar fotos o videos de los incidentes
- f) Capacidad de interactuar con otros usuarios para compartir información y alertas
- g) Acceso a información sobre rutas alternativas y transporte público
- h) Estadísticas y análisis de incidentes para mejorar la planificación vial
- i) Otra (por favor, especifica)

4. En tu opinión, ¿cómo crees que una aplicación móvil para reportar incidentes de tránsito podría contribuir a mejorar la movilidad en Medellín? (Respuesta abierta)

5. ¿Qué desafíos o dificultades crees que podrían surgir al utilizar una aplicación móvil para reportar incidentes de tránsito? (Respuesta abierta)

6. ¿Te gustaría recibir actualizaciones o noticias sobre el estado del tránsito en Medellín a través de la aplicación móvil?

- a) Sí, me gustaría recibir actualizaciones y noticias
- b) No, no estoy interesado(a) en recibir actualizaciones y noticias

¡Muchas gracias por tu participación! Tu opinión es valiosa para el desarrollo de esta

Respuesta de 6 personas

Respuesta 1:

1. ¿Con qué frecuencia utilizas aplicaciones móviles relacionadas con el tránsito o la

movilidad en la ciudad de Medellín?

- Opción seleccionada: a) Diariamente

2. ¿Estarías interesado en utilizar una aplicación móvil específica para reportar incidentes de tránsito en Medellín?

- Opción seleccionada: a) Sí, definitivamente

3. ¿Qué características o funcionalidades consideras importantes en una aplicación móvil para reportar incidentes de tránsito? (Selecciona todas las opciones que apliquen)

- Opciones seleccionadas: a) Fácil y rápida interfaz de reporte de incidentes, b) Integración con mapas y geolocalización, c) Notificaciones en tiempo real sobre incidentes en rutas frecuentes, e) Opción para adjuntar fotos o videos de los incidentes

4. En tu opinión, ¿cómo crees que una aplicación móvil para reportar incidentes de tránsito podría contribuir a mejorar la movilidad en Medellín?

- Respuesta abierta: Creo que una aplicación móvil para reportar incidentes de tránsito sería de gran ayuda para alertar a otros conductores sobre situaciones de tráfico y accidentes, lo que permitiría evitar rutas congestionadas y tomar decisiones más informadas para mejorar la fluidez del tránsito en la ciudad.

5. ¿Qué desafíos o dificultades crees que podrían surgir al utilizar una aplicación móvil para reportar incidentes de tránsito?

- Respuesta abierta: Uno de los desafíos podría ser la veracidad de la información reportada por los usuarios, ya que algunas personas podrían exagerar o falsificar incidentes. También podría haber dificultades técnicas, como la falta de conexión a Internet en algunas áreas de la ciudad.

6. ¿Te gustaría recibir actualizaciones o noticias sobre el estado del tránsito en Medellín a través de la aplicación móvil?

- Opción seleccionada: a) Sí, me gustaría recibir actualizaciones y noticias

Respuesta 2:

1. ¿Con qué frecuencia utilizas aplicaciones móviles relacionadas con el tránsito o la movilidad en la ciudad de Medellín?

- Opción seleccionada: c) Ocasionalmente



2. ¿Estarías interesado en utilizar una aplicación móvil específica para reportar incidentes de tránsito en Medellín?
  - Opción seleccionada: b) Tal vez, dependiendo de sus características
3. ¿Qué características o funcionalidades consideras importantes en una aplicación móvil para reportar incidentes de tránsito? (Selecciona todas las opciones que apliquen)
  - Opciones seleccionadas: b) Integración con mapas y geolocalización, d) Información actualizada sobre el estado del tráfico y las condiciones viales, g) Acceso a información sobre rutas alternativas y transporte público
4. En tu opinión, ¿cómo crees que una aplicación móvil para reportar incidentes de tránsito podría contribuir a mejorar la movilidad en Medellín?
  - Respuesta abierta: Creo que una aplicación móvil podría ayudar a las autoridades de tránsito a tener información en tiempo real sobre los incidentes, lo que les permitiría tomar decisiones más eficientes para desviar el tráfico y mejorar la movilidad en las vías afectadas.
5. ¿Qué desafíos o dificultades crees que podrían surgir al utilizar una aplicación móvil para reportar incidentes de tránsito?
  - Respuesta abierta: Un desafío podría ser la falta de participación de los conductores en el reporte de incidentes, lo que limitaría la cantidad de información disponible. Además, es importante garantizar la privacidad de los datos de los usuarios y asegurar que la aplicación no cause distracciones mientras se conduce.
6. ¿Te gustaría recibir actualizaciones o noticias sobre el estado del tránsito en Medellín a través de la aplicación móvil?
  - Opción seleccionada: a) Sí, me gustaría recibir actualizaciones y noticias

Respuesta 3:

1. ¿Con qué frecuencia utilizas aplicaciones móviles relacionadas con el tránsito o la movilidad en la ciudad de Medellín?
  - Opción seleccionada: d) No utilizo aplicaciones móviles relacionadas con el tránsito
2. ¿Estarías interesado en utilizar una aplicación móvil específica para reportar

incidentes de tránsito en Medellín?

- Opción seleccionada: a) Sí, definitivamente

3. ¿Qué características o funcionalidades consideras importantes en una aplicación móvil para reportar incidentes de tránsito? (Selecciona todas las opciones que apliquen)

- Opciones seleccionadas: a) Fácil y rápida interfaz de reporte de incidentes, b) Integración con mapas y geolocalización, d) Información actualizada sobre el estado del tráfico y las condiciones viales, e) Opción para adjuntar fotos o videos de los incidentes

4. En tu opinión, ¿cómo crees que una aplicación móvil para reportar incidentes de tránsito podría contribuir a mejorar la movilidad en Medellín?

- Respuesta abierta: Creo que una aplicación móvil permitiría a los conductores informar rápidamente sobre cualquier incidente en la vía, lo que ayudaría a las autoridades a tomar medidas inmediatas para solucionarlos y mejorar la fluidez del tránsito en la ciudad.

5. ¿Qué desafíos o dificultades crees que podrían surgir al utilizar una aplicación móvil para reportar incidentes de tránsito?

- Respuesta abierta: Uno de los desafíos podría ser garantizar que los reportes de incidentes sean verificados y no se utilicen de manera incorrecta o malintencionada. Además, es importante asegurar que la aplicación sea fácil de usar para que todos los conductores puedan reportar incidentes sin complicaciones.

6. ¿Te gustaría recibir actualizaciones o noticias sobre el estado del tránsito en Medellín a través de la aplicación móvil?

- Opción seleccionada: b) No, no estoy interesado(a) en recibir actualizaciones y noticias

Respuesta 4:

1. ¿Con qué frecuencia utilizas aplicaciones móviles relacionadas con el tránsito o la movilidad en la ciudad de Medellín?

- Opción seleccionada: b) Varias veces a la semana

2. ¿Estarías interesado en utilizar una aplicación móvil específica para reportar

incidentes de tránsito en Medellín?

- Opción seleccionada: a) Sí, definitivamente

3. ¿Qué características o funcionalidades consideras importantes en una aplicación móvil para reportar incidentes de tránsito? (Selecciona todas las opciones que apliquen)

- Opciones seleccionadas: a) Fácil y rápida interfaz de reporte de incidentes, b) Integración con mapas y geolocalización, c) Notificaciones en tiempo real sobre incidentes en rutas frecuentes, d) Información actualizada sobre el estado del tráfico y las condiciones viales, e) Opción para adjuntar fotos o videos de los incidentes

4. En tu opinión, ¿cómo crees que una aplicación móvil para reportar incidentes de tránsito podría contribuir a mejorar la movilidad en Medellín?

- Respuesta abierta: Una aplicación móvil sería una herramienta eficiente para que los conductores reporten incidentes de tránsito de manera rápida y precisa. Esto permitiría a las autoridades tomar medidas inmediatas para solucionar los problemas y agilizar el flujo de vehículos en las vías de la ciudad.

5. ¿Qué desafíos o dificultades crees que podrían surgir al utilizar una aplicación móvil para reportar incidentes de tránsito?

- Respuesta abierta: Un desafío podría ser asegurar que la aplicación sea fácil de usar para todos los usuarios, incluyendo a aquellos con poca experiencia tecnológica. También podría haber preocupaciones sobre la privacidad de los datos personales al utilizar la aplicación.

6. ¿Te gustaría recibir actualizaciones o noticias sobre el estado del tránsito en Medellín a través de la aplicación móvil?

- Opción seleccionada: a) Sí, me gustaría recibir actualizaciones y noticias

Respuesta 5:

¿Con qué frecuencia utilizas aplicaciones móviles relacionadas con el tránsito o la movilidad en la ciudad de Medellín?

- Opción seleccionada: a) Diariamente

¿Estarías interesado en utilizar una aplicación móvil específica para reportar incidentes de tránsito en Medellín?

- Opción seleccionada: a) Sí, definitivamente

¿Qué características o funcionalidades consideras importantes en una aplicación móvil para reportar incidentes de tránsito? (Selecciona todas las opciones que apliquen)

- Opciones seleccionadas: a) Fácil y rápida interfaz de reporte de incidentes, b) Integración con mapas y geolocalización, c) Notificaciones en tiempo real sobre incidentes en rutas frecuentes, e) Opción para adjuntar fotos o videos de los incidentes, f) Capacidad de interactuar con otros usuarios para compartir información y alertas

En tu opinión, ¿cómo crees que una aplicación móvil para reportar incidentes de tránsito podría contribuir a mejorar la movilidad en Medellín?

- Respuesta abierta: Una aplicación móvil sería una forma efectiva de comunicar los incidentes de tránsito en tiempo real a las autoridades y otros conductores. Esto permitiría una respuesta más rápida y coordinada para resolver los problemas de tráfico y mejorar la movilidad en la ciudad.

¿Qué desafíos o dificultades crees que podrían surgir al utilizar una aplicación móvil para reportar incidentes de tránsito?

- Respuesta abierta: Un desafío podría ser asegurar la participación activa de los conductores en el reporte de incidentes y garantizar la veracidad de la información compartida. Además, es importante evitar que la aplicación cause distracciones mientras se conduce y garantizar la seguridad de los datos de los usuarios.

¿Te gustaría recibir actualizaciones o noticias sobre el estado del tránsito en Medellín a través de la aplicación móvil?

- Opción seleccionada: a) Sí, me gustaría recibir actualizaciones y noticias

Respuesta 6:

1. ¿Con qué frecuencia utilizas aplicaciones móviles relacionadas con el tránsito o la movilidad en la ciudad de Medellín?

- Opción seleccionada: c) Ocasionalmente

2. ¿Estarías interesado en utilizar una aplicación móvil específica para reportar incidentes de tránsito en Medellín?

- Opción seleccionada: a) Sí, definitivamente
3. ¿Qué características o funcionalidades consideras importantes en una aplicación móvil para reportar incidentes de tránsito? (Selecciona todas las opciones que apliquen)
- Opciones seleccionadas: a) Fácil y rápida interfaz de reporte de incidentes, b) Integración con mapas y geolocalización, d) Información actualizada sobre el estado del tráfico y las condiciones viales, g) Acceso a información sobre rutas alternativas y transporte público
4. En tu opinión, ¿cómo crees que una aplicación móvil para reportar incidentes de tránsito podría contribuir a mejorar la movilidad en Medellín?
- Respuesta abierta: Creo que una aplicación móvil podría ayudar a los conductores a tomar decisiones más informadas sobre sus rutas, evitando zonas congestionadas y optimizando el tiempo de viaje. Además, podría fomentar una mayor colaboración entre los usuarios y las autoridades para resolver los problemas de tráfico de manera más eficiente.
5. ¿Qué desafíos o dificultades crees que podrían surgir al utilizar una aplicación móvil para reportar incidentes de tránsito?
- Respuesta abierta: Un desafío podría ser garantizar la confiabilidad de los reportes de incidentes y evitar la difusión de información falsa. También es importante garantizar que la aplicación sea accesible para todos los usuarios, incluyendo a aquellos con habilidades tecnológicas limitadas.
6. ¿Te gustaría recibir actualizaciones o noticias sobre el estado del tránsito en Medellín a través de la aplicación móvil?
- Opción seleccionada: b) No, no estoy interesado(a) en recibir actualizaciones y noticias

### 5.3.2

5.3.3 **Fuentes secundarias.** Una fuente secundaria interpreta y analiza fuentes primarias. Algunos tipos de fuentes secundarias son: enciclopedias, antologías, directorios, libros o artículos que interpretan otros trabajos o investigaciones.

## 6 Resultados

A continuación, se encuentran los pasos en los que se implementó el aplicativo de movilidad para la ciudad de Medellín, teniendo en cuenta la metodología SCRUM, que se planteó con anterioridad, en el capítulo de Metodología.

## 6.1 Análisis

Identificación de los usuarios principales:

- Los usuarios principales son aquellos que desean reportar incidentes de tráfico en la ciudad de Medellín.
- Se deben tener en cuenta las necesidades y expectativas de estos usuarios, como la capacidad de reportar diferentes tipos de incidentes y la facilidad de uso de la aplicación.
- Funcionalidades principales:
- La aplicación debe permitir a los usuarios reportar incidentes de tráfico en tiempo real.
- Debe haber una función para activar la ubicación del usuario y agregar automáticamente la ubicación al reporte del incidente.
- Los usuarios deben poder adjuntar una foto del incidente y proporcionar una breve descripción del mismo.
- Se debe implementar un mapa interactivo en tiempo real que muestre los incidentes reportados por otros usuarios.

Revisión de aplicaciones similares:

- Es importante analizar otras aplicaciones similares disponibles en el mercado para identificar buenas prácticas y posibles mejoras.
- Esto puede proporcionar ideas sobre las funcionalidades adicionales que se pueden incluir en la aplicación y cómo mejorar la experiencia del usuario.
- Diseño modular y escalable:
- Se debe adoptar una arquitectura modular que permita la escalabilidad y la incorporación de nuevas funcionalidades en el futuro.

- El diseño debe seguir principios de usabilidad y experiencia de usuario para garantizar que la aplicación sea intuitiva y fácil de usar.

Pruebas:

- Es fundamental realizar pruebas en diferentes escenarios y dispositivos para asegurar la calidad y la fiabilidad de la aplicación.
- Las pruebas deben abarcar aspectos de funcionalidad, usabilidad y rendimiento, corrigiendo cualquier error y optimizando el rendimiento según sea necesario.

### **6.1.1 Historias de Usuario.**

1. Como usuario, quiero poder registrar un accidente de tráfico en la aplicación para informar a otros conductores sobre la situación y tomar rutas alternativas si es necesario.
2. Como usuario, quiero poder reportar congestión de tráfico en tiempo real para que otros conductores puedan evitar esas áreas y encontrar rutas más eficientes.
3. Como usuario, quiero poder informar sobre obras viales en curso en la ciudad para que otros conductores puedan planificar sus rutas de antemano y evitar retrasos.
4. Como usuario, quiero recibir notificaciones en tiempo real sobre incidentes de tráfico reportados en mi ruta habitual para poder tomar decisiones informadas sobre la mejor ruta a seguir.
5. Como usuario, quiero poder adjuntar imágenes a mis informes de incidentes de tráfico para proporcionar una mayor claridad y comprensión de la situación.
6. Como usuario, quiero poder ver un mapa interactivo que muestre los incidentes de tráfico reportados por otros usuarios en mi área, para tener una visión general de la situación del tráfico.
7. Como usuario, quiero poder filtrar los incidentes de tráfico en el mapa por

tipo (accidentes, congestión, obras viales, etc.) para tener una visión más específica de los problemas que pueden afectar mi ruta.

8. Como usuario, quiero tener la opción de marcar un incidente de tráfico como resuelto una vez que haya sido atendido o haya desaparecido, para mantener actualizada la información en la aplicación.

9. Como usuario, quiero tener la posibilidad de guardar incidentes de tráfico como favoritos para referencia futura o para compartir con otros conductores.

10. Como usuario, quiero tener acceso a estadísticas y datos históricos sobre incidentes de tráfico reportados en la ciudad, para comprender las tendencias y patrones de tráfico en diferentes áreas.

Estas historias de usuario cubren diferentes funcionalidades y necesidades de los usuarios para mejorar la experiencia de reporte de incidentes de tráfico y la movilidad en la ciudad.

## 6.2 Diseño

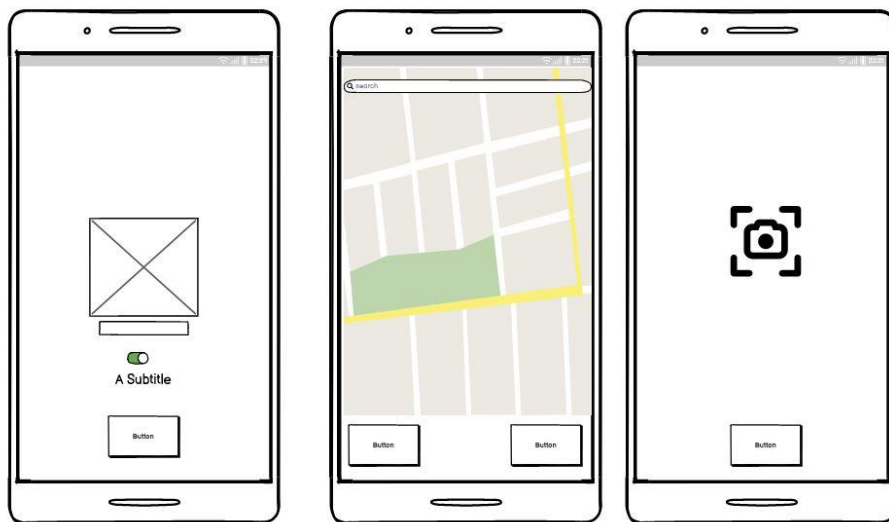


Figura 6. Diseño inicial



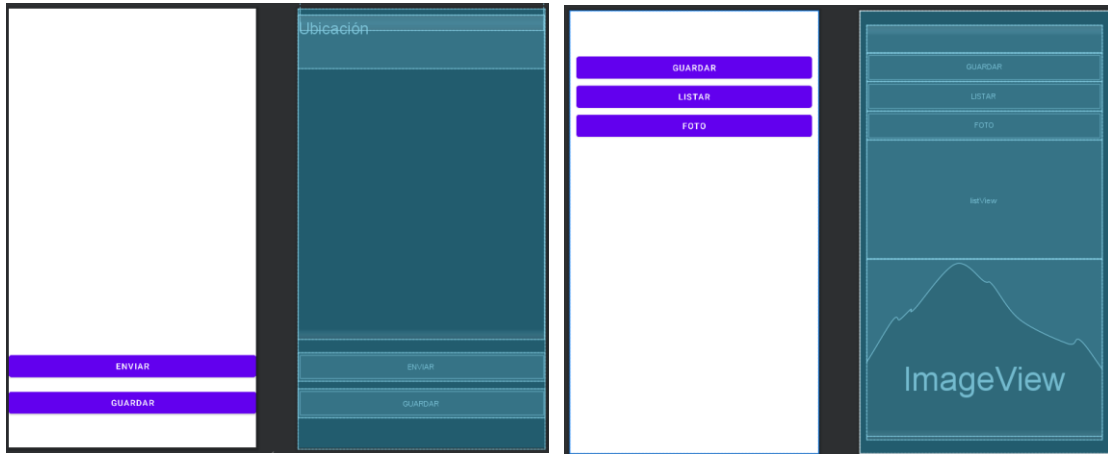


Figura 7. Diseño final

## 6.2.1 Arquitectura

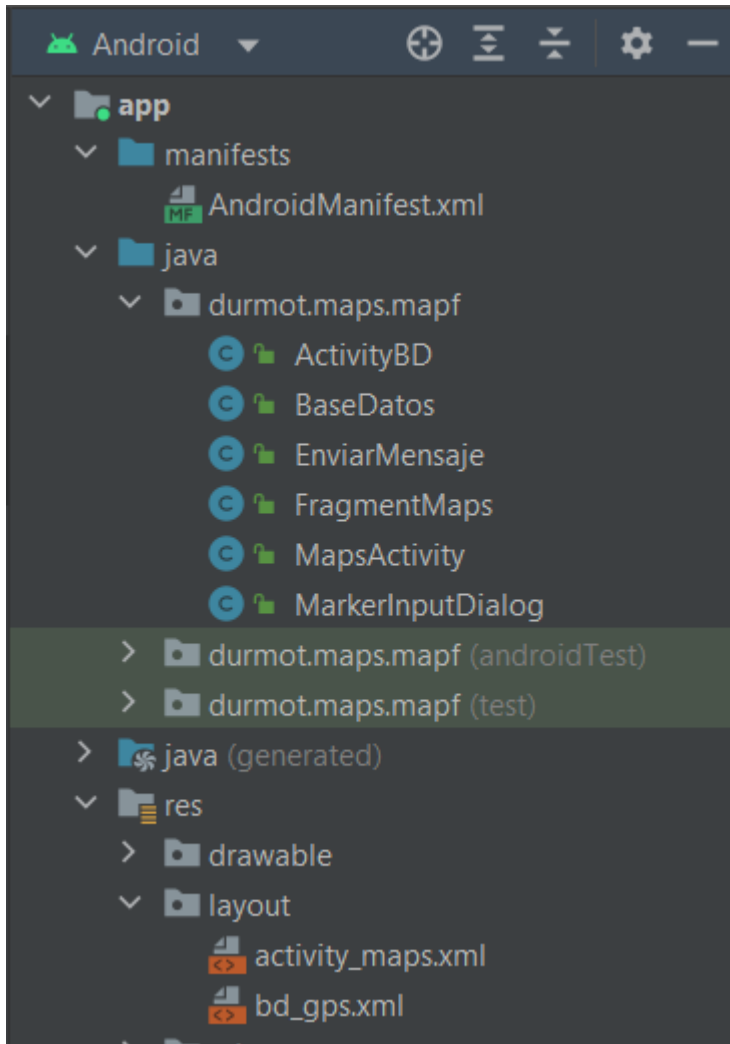


Figura 8. Arquitectura del programa

### 6.2.3 Diagrama de clases

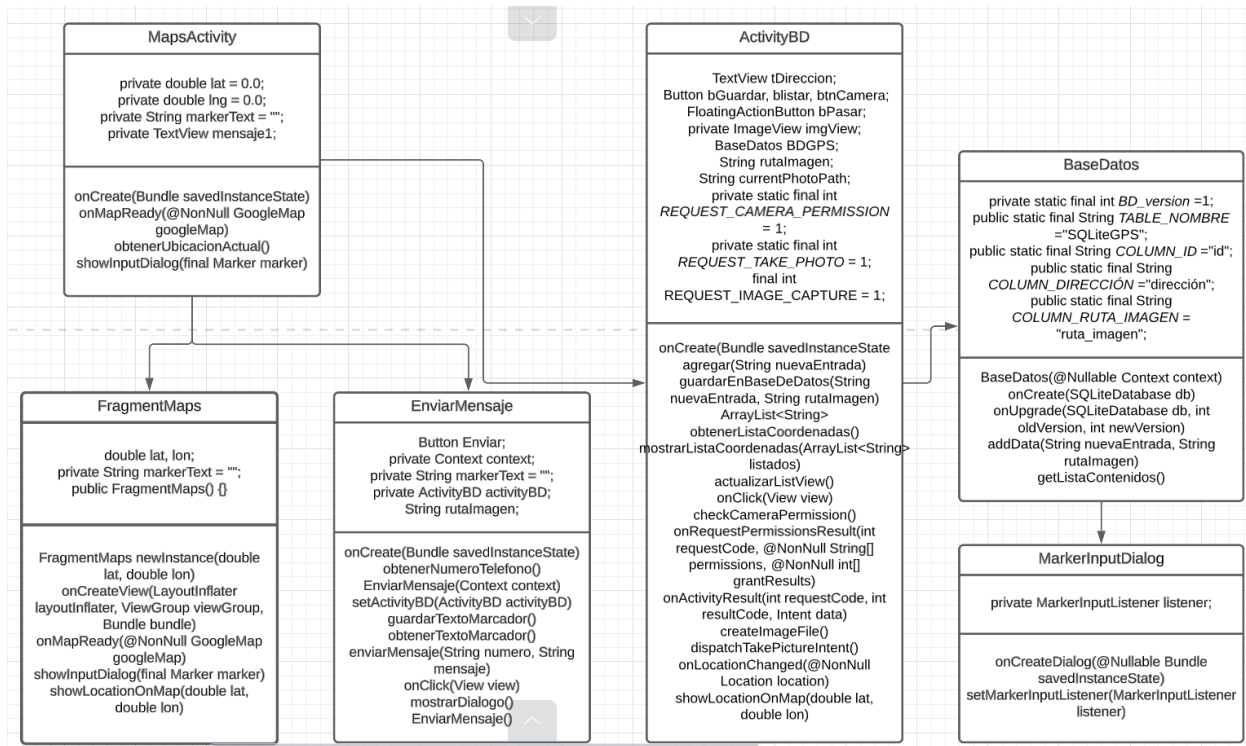


Figura 9. Diagrama de clases de la aplicación

## 6.2.4 Base de datos

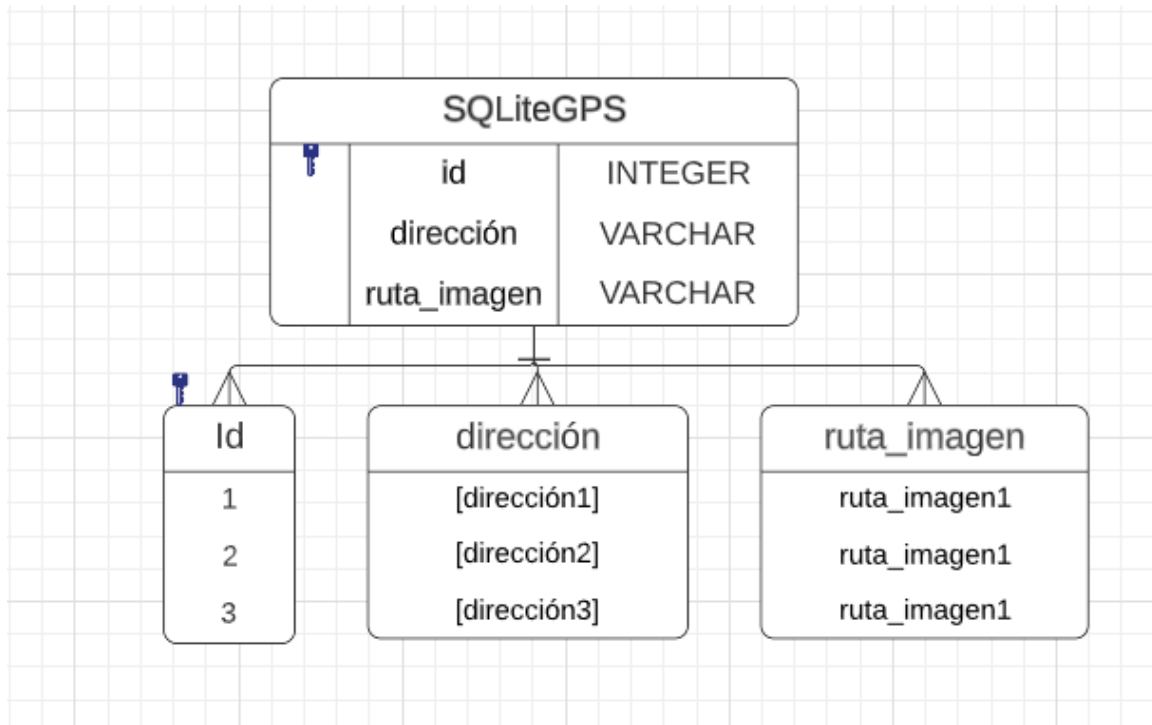


Figura 10. Diagrama base de datos

## 6.2.5 Interfaz Gráfica

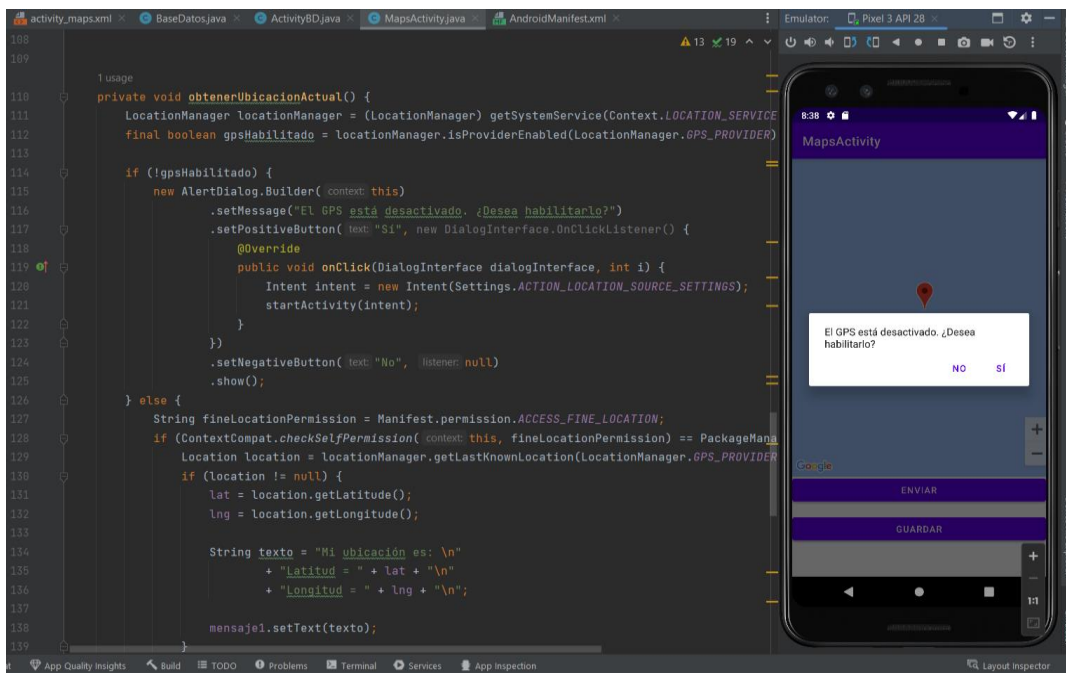


Figura 11. Solicitud de permiso para el uso de la ubicación

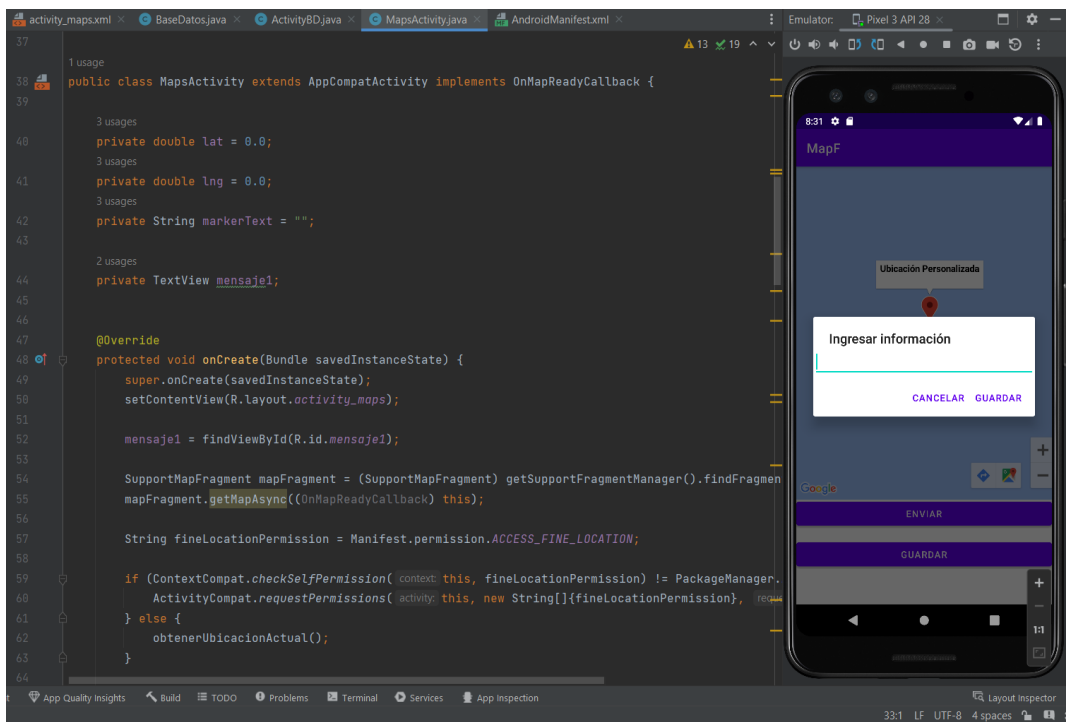


Figura 12. Ingresar información en el marcador

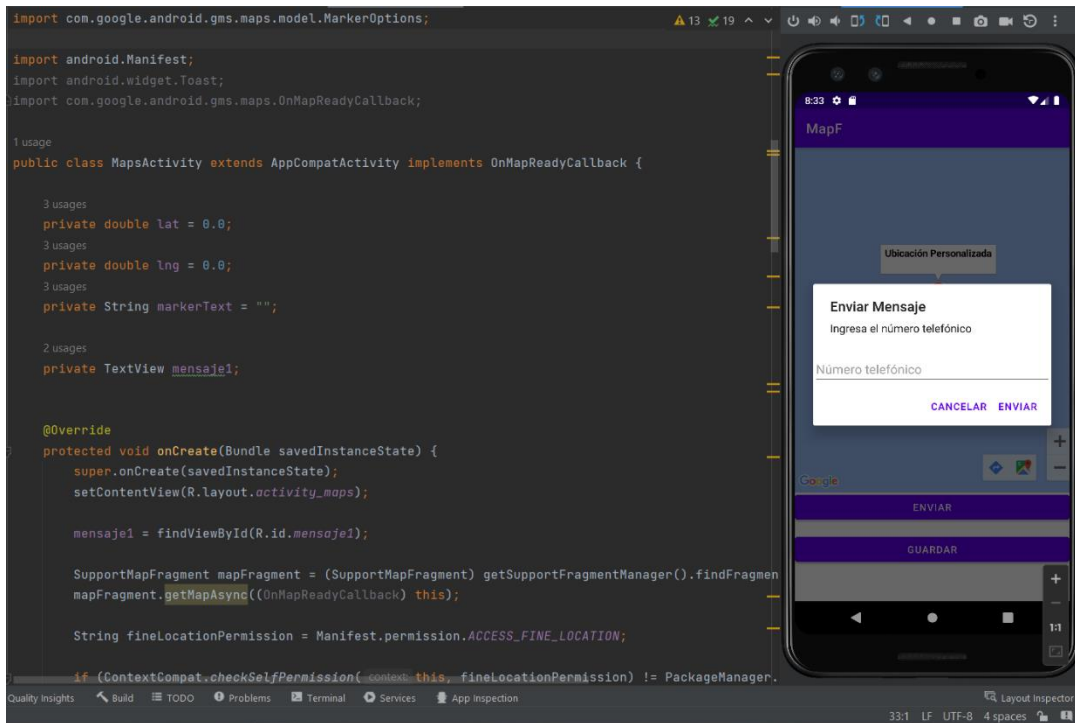


Figura 13. Enviar mensaje de texto con el reporte del incidente y la ubicación en tiempo real

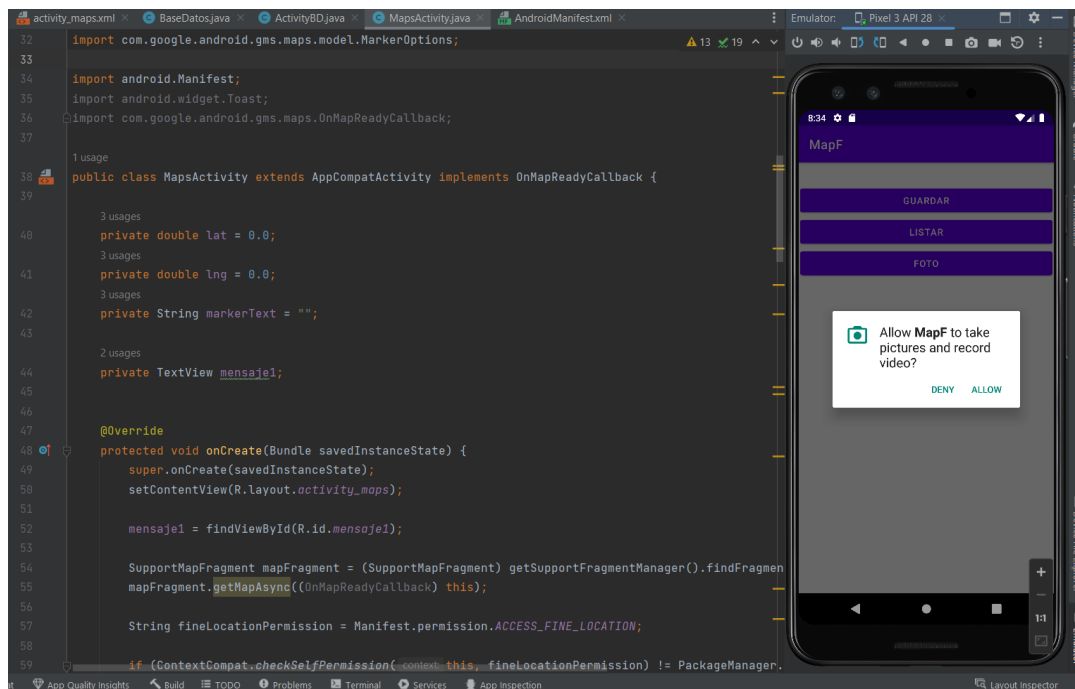


Figura 14. Solicitud de permiso para el uso de la cámara

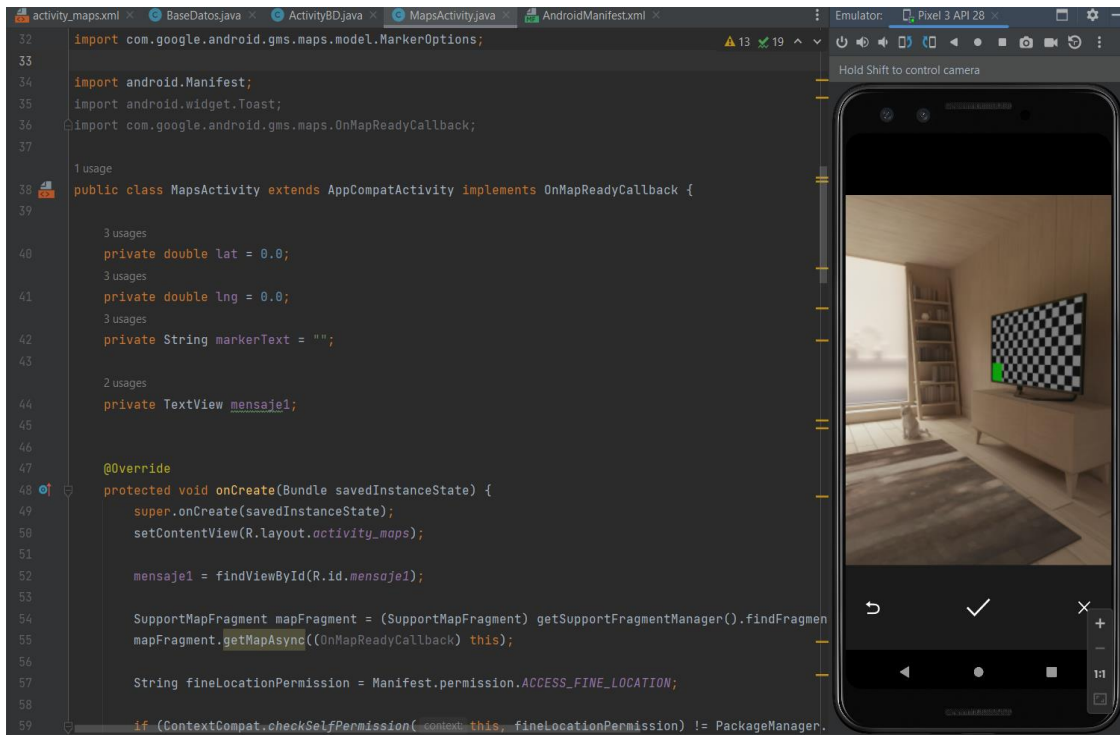


Figura 15. Captura de fotografía

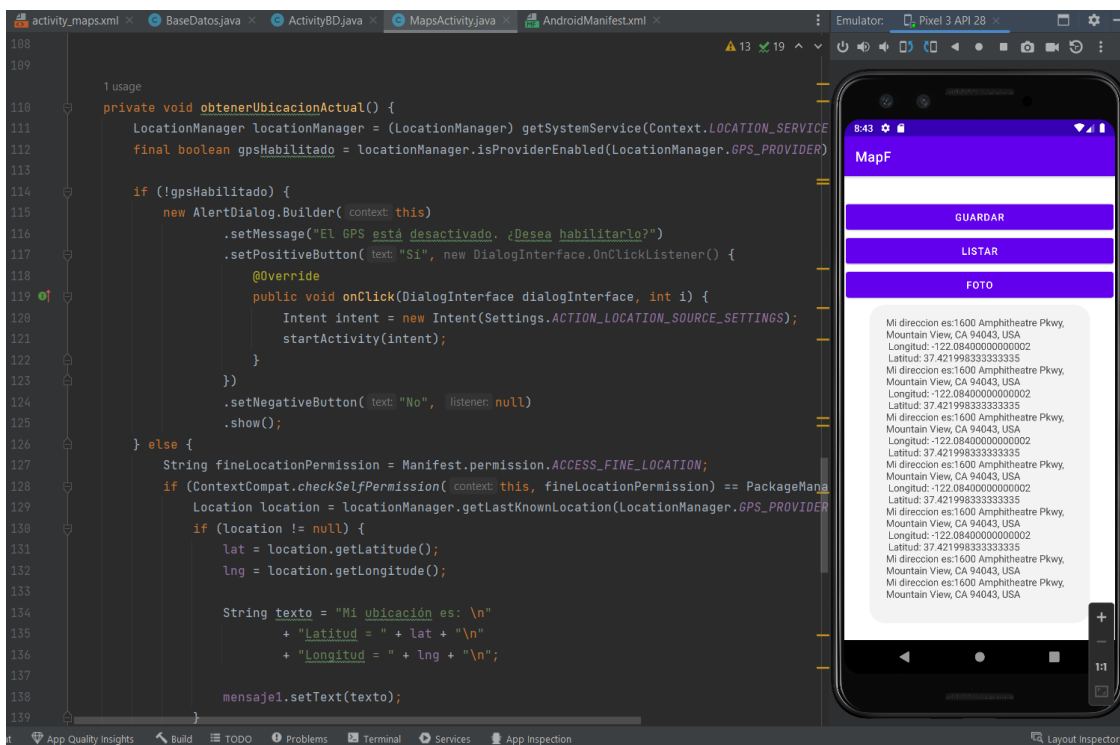


Figura 16. Listado de reportes

### 6.3 Implementación

A continuación se explica cómo se implementó la aplicación, basada en las clases, con el Framework de Android Studio y el aplicativo

1. **MapsActivity**: Esta clase representa la actividad principal que muestra el mapa y gestiona los marcadores y la interacción con el usuario.

`onCreate (Bundle savedInstanceState)`: Este método se ejecuta cuando se crea la actividad. Aquí se realiza la inicialización de la actividad, se establece el diseño de la interfaz de usuario y se configuran los listeners de los botones y del mapa.

```
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_maps);

    mensaje1 = findViewById(R.id.mensaje1);
    bPasar = findViewById(R.id.fab);

    SupportMapFragment mapFragment = (SupportMapFragment)
    getSupportFragmentManager().findFragmentById(R.id.map);

    mapFragment.getMapAsync(new OnMapReadyCallback() {
        @Override
        public void onMapReady(@NonNull GoogleMap googleMap) {
            LatLng latLng = new LatLng(lat, lng);
            float zoom = 17;

            googleMap.moveCamera(CameraUpdateFactory.newLatLngZoom(latLng, zoom));

            MarkerOptions markerOptions = new MarkerOptions()
```



```

        .position(latLng)
        .snippet(markerText);

    googleMap.addMarker(markerOptions);

    googleMap.setOnMarkerClickListener(new
    GoogleMap.OnMarkerClickListener() {

        @Override

        public boolean onMarkerClick(Marker marker) {

            showInputDialog(marker);

            return false;

        }

    });

    UiSettings settings = googleMap.getUiSettings();

    settings.setZoomControlsEnabled(true);

}

});

String fineLocationPermission =
Manifest.permission.ACCESS_FINE_LOCATION;

if (ContextCompat.checkSelfPermission(this, fineLocationPermission) !=
PackageManager.PERMISSION_GRANTED) {

    ActivityCompat.requestPermissions(this, new
String[] {fineLocationPermission}, 1000);

} else {

```

```
        obtenerUbicacionActual();
    }

    View btnGuardar = findViewById(R.id.btnguardar);
    btnGuardar.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            Intent layoutBD = new Intent(getApplicationContext(),
ActivityBD.class);
            startActivity(layoutBD);
        }
    });

    View btnEnviar = findViewById(R.id.btnEnviar);
    btnEnviar.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            Intent EnviarMen = new Intent(getApplicationContext(),
EnviarMensaje.class);
            startActivity(EnviarMen);
        }
    });

    bPasar.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
```

```

        showLocationOnMap(lat, lng);
    }
});
}

```

`onMapReady(@NonNull GoogleMap googleMap)`: Este método se llama cuando el mapa está listo para ser utilizado. Aquí se configuran las propiedades del mapa, como la posición y el zoom, se añade un marcador y se configura el evento de clic en el marcador.

```

@Override

public void onMapReady(@NonNull GoogleMap googleMap) {

    LatLng latLng = new LatLng(lat, lng);

    float zoom = 17;

    googleMap.moveCamera(CameraUpdateFactory.newLatLngZoom(latLng, zoom));

    MarkerOptions markerOptions = new MarkerOptions()
        .position(latLng)
        .snippet(markerText);

    googleMap.addMarker(markerOptions);

    googleMap.setOnMarkerClickListener(new
    GoogleMap.OnMarkerClickListener() {

        @Override

        public boolean onMarkerClick(Marker marker) {

            showInputDialog(marker);
        }
    });
}

```

```

        return false;
    }
});

    UiSettings settings = googleMap.getUiSettings();
    settings.setZoomControlsEnabled(true);
}
});

    LocationManager locationManager = (LocationManager)
    getSystemService(Context.LOCATION_SERVICE);

    final boolean gpsHabilitado =
    locationManager.isProviderEnabled(LocationManager.GPS_PROVIDER);

    if (!gpsHabilitado) {
        new AlertDialog.Builder(this)
            .setMessage("El GPS está desactivado. ¿Desea habilitarlo?")
            .setPositiveButton("Sí", new DialogInterface.OnClickListener() {
                @Override
                public void onClick(DialogInterface dialogInterface, int i) {
                    Intent intent = new
    Intent(Settings.ACTION_LOCATION_SOURCE_SETTINGS);
                    startActivity(intent);
                }
            })
            .setNegativeButton("No", null)

```

```

        .show();
    } else {
        String fineLocationPermission =
Manifest.permission.ACCESS_FINE_LOCATION;

        if (ContextCompat.checkSelfPermission(this, fineLocationPermission) ==
PackageManager.PERMISSION_GRANTED) {

            Location location =
locationManager.getLastKnownLocation(LocationManager.GPS_PROVIDER);

            if (location != null) {

                lat = location.getLatitude();

                lng = location.getLongitude();

                String texto = "Mi ubicación es: \n"
                    + "Latitud = " + lat + "\n"
                    + "Longitud = " + lng + "\n";

                mensaje1.setText(texto);
            }
        }
    }
}
}
}

```

obtenerUbicacionActual(): Este método se encarga de obtener la ubicación actual del dispositivo. Verifica si el GPS está habilitado y solicita permisos para acceder a la ubicación del usuario. Si el GPS está desactivado, muestra un diálogo para solicitar al usuario que lo active.

```

private void obtenerUbicacionActual() {

    LocationManager locationManager = (LocationManager)
    getSystemService(Context.LOCATION_SERVICE);

    final boolean gpsHabilitado =
    locationManager.isProviderEnabled(LocationManager.GPS_PROVIDER);

    if (!gpsHabilitado) {

        new AlertDialog.Builder(this)

            .setMessage("El GPS está desactivado. ¿Desea habilitarlo?")

            .setPositiveButton("Sí", new DialogInterface.OnClickListener() {

                @Override

                public void onClick(DialogInterface dialogInterface, int i) {

                    Intent intent = new
                    Intent(Settings.ACTION_LOCATION_SOURCE_SETTINGS);

                    startActivity(intent);

                }

            })

            .setNegativeButton("No", null)

            .show();

    } else {

        String fineLocationPermission =
        Manifest.permission.ACCESS_FINE_LOCATION;

        if (ContextCompat.checkSelfPermission(this, fineLocationPermission) ==
        PackageManager.PERMISSION_GRANTED) {

            Location location =
            locationManager.getLastKnownLocation(LocationManager.GPS_PROVIDER);

            if (location != null) {

                lat = location.getLatitude();
            }
        }
    }
}

```

```

        lng = location.getLongitude();

        String texto = "Mi ubicación es: \n"
            + "Latitud = " + lat + "\n"
            + "Longitud = " + lng + "\n";

        mensaje1.setText(texto);
    }
}
}
}

```

Estos son los métodos más significativos de la clase `MapsActivity`. Cada uno cumple una función específica relacionada con la visualización y manipulación del mapa.

2. **FragmentMaps**: Esta clase representa un fragmento que se utiliza para mostrar el mapa y proporcionar funcionalidades relacionadas con la ubicación y los marcadores.

`newInstance(double lat, double lon)`: Este método estático se utiliza para crear una nueva instancia del fragmento con argumentos. Recibe la latitud y longitud como parámetros, crea un nuevo fragmento, guarda los argumentos en un `Bundle` y los asigna al fragmento creado.

```

// Método estático para crear una nueva instancia del fragmento con argumentos
public static FragmentMaps newInstance(double lat, double lon) {
    FragmentMaps fragment = new FragmentMaps();
    Bundle args = new Bundle();
}

```

```

        args.putDouble("lat", lat);
        args.putDouble("lon", lon);
        fragment.setArguments(args);
        return fragment;
    }

```

`onCreateView(LayoutInflater inflater, ViewGroup viewGroup, Bundle bundle)`: Este método se ejecuta cuando el fragmento está siendo creado y necesita inflar su diseño. Aquí se obtienen los argumentos del fragmento, se inicializa el mapa de manera asíncrona y se retorna la vista inflada.

*@Override*

```

    public View onCreateView(LayoutInflater inflater, ViewGroup
viewGroup, Bundle bundle){

        // Se infla el diseño del fragmento

        View rootView = super.onCreateView(inflater, viewGroup, bundle);

        if (getArguments() != null){

            // Se obtienen las coordenadas de ubicación del fragmento

            this.lat = getArguments().getDouble("lat");
            this.lon = getArguments().getDouble("lon");
        }

        // Se inicializa el mapa asíncronamente

        getMapAsync((OnMapReadyCallback) this);

```



```

        return rootView;
    }

```

`onMapReady(@NonNull GoogleMap googleMap)`: Este método se llama cuando el mapa está listo para ser utilizado. Aquí se configuran las propiedades del mapa, como el tipo de mapa, la posición y el zoom. Además, se agrega un marcador en la ubicación especificada y se configuran eventos de clic en el marcador.

```

@Override

public void onMapReady(@NonNull GoogleMap googleMap) {

    googleMap.setMapType(GoogleMap.MAP_TYPE_NORMAL);

    LatLng latLng = new LatLng(lat, lon);

    float zoom = 17;

    // Se mueve la cámara del mapa a la ubicación especificada
    googleMap.moveCamera(CameraUpdateFactory.newLatLngZoom(latLng,
zoom));

    MarkerOptions markerOptions = new MarkerOptions()

        .position(latLng)

        .title("Ubicación Personalizada")

        .snippet(markerText); // Se utiliza el texto almacenado en la variable de
instancia

    // Se agrega un marcador al mapa

```

```

googleMap.addMarker(markerOptions);

googleMap.setOnMarkerClickListener(new
GoogleMap.OnMarkerClickListener() {

    @Override

    public boolean onMarkerClick(Marker marker) {

        // Se muestra un cuadro de diálogo para ingresar información al
marcador

        showInputDialog(marker);

        return false;

    }

});

googleMap.setOnMarkerClickListener(new
GoogleMap.OnMarkerClickListener() {

    @Override

    public boolean onMarkerClick(final Marker marker) {

        MarkerInputDialog dialog = new MarkerInputDialog();

        dialog.setMarkerInputListener(new
MarkerInputDialog.MarkerInputListener() {

            @Override

            public void onMarkerInputEntered(String text) {

                marker.setSnippet(text); // Actualiza el texto del marcador

            }

        });

        dialog.show(requireActivity().getSupportFragmentManager(),
"marker_input_dialog");

```

```

        return false;
    }
});

```

```

    UiSettings settings = googleMap.getUiSettings();
    settings.setZoomControlsEnabled(true);
}

```

`showInputDialog(final Marker marker)`: Este método muestra un cuadro de diálogo donde el usuario puede ingresar información relacionada con un marcador en el mapa. Al guardar la información, se actualiza el texto del marcador.

```

private void showInputDialog(final Marker marker) {
    AlertDialog.Builder builder = new AlertDialog.Builder(requireContext());
    builder.setTitle("Ingresar información");

    final EditText input = new EditText(requireContext());

    input.setInputType(InputType.TYPE_CLASS_TEXT);
    input.setText(markerText); // Se establece el texto del marcador en el cuadro
de texto

    builder.setView(input);

    builder.setPositiveButton("Guardar", new DialogInterface.OnClickListener()
{
    @Override

```

```

        public void onClick(DialogInterface dialog, int which) {
            String text = input.getText().toString();
            markerText = text; // Se almacena el texto en la variable de instancia
            marker.setSnippet(text); // Se establece el texto en el marcador
        }
    });

    builder.setNegativeButton("Cancelar", new
DialogInterface.OnClickListener() {
        @Override
        public void onClick(DialogInterface dialog, int which) {
            dialog.cancel();
        }
    });

    builder.show();
}

```

`showLocationOnMap(double lat, double lon)`: Este método muestra la ubicación especificada en el mapa utilizando otro fragmento `FragmentMaps`. Crea un nuevo fragmento con los argumentos de ubicación, reemplaza el fragmento actual en la vista del mapa con el nuevo fragmento y lo muestra.

```

// Método utilizado para mostrar la ubicación en el mapa
private void showLocationOnMap(double lat, double lon) {
    FragmentMaps fragment = new FragmentMaps();
    Bundle bundle = new Bundle();
}

```

```

        bundle.putDouble("lat", lat);

        bundle.putDouble("lon", lon);

        fragment.setArguments(bundle);

        fragmentManager fragmentManager = getChildFragmentManager();

        FragmentTransaction fragmentTransaction =
        fragmentManager.beginTransaction();

        fragmentTransaction.replace(R.id.map, fragment);

        fragmentTransaction.commitAllowingStateLoss();
    }
}

```

Estos son los métodos más significativos de la clase FragmentMaps. Cada uno cumple una función específica relacionada con la visualización y manipulación del mapa dentro del fragmento.

3. **EnviarMensaje:** Esta clase representa una funcionalidad para enviar mensajes, probablemente relacionada con la comunicación de incidentes de tráfico.

`onCreate(Bundle savedInstanceState):` Este método se ejecuta cuando se crea la actividad. Aquí se configuran permisos para enviar mensajes SMS y se establece un listener de clic para el botón de enviar.

*@Override*

```

protected void onCreate(Bundle savedInstanceState) {

    super.onCreate(savedInstanceState);

    setContentView(R.layout.activity_maps);
}

```

```

        if (ActivityCompat.checkSelfPermission(
            EnviarMensaje.this, android.Manifest.permission.SEND_SMS)
            != PackageManager.PERMISSION_GRANTED &&
            ActivityCompat.checkSelfPermission(
                EnviarMensaje.this, android.Manifest
                    .permission.SEND_SMS) !=
            PackageManager.PERMISSION_GRANTED) {
            ActivityCompat.requestPermissions(EnviarMensaje.this, new String[]
                {android.Manifest.permission.SEND_SMS,}, 1000);
        } else {
        }
        ;

        View btnEnviar = findViewById(R.id.btnEnviar);

        btnEnviar.setOnClickListener((View.OnClickListener) this);
    }

```

onClick(View view): Este método se llama cuando se produce un evento de clic en la vista. Aquí se verifica si el evento proviene del botón de enviar y se muestra un diálogo para ingresar el número telefónico.

```

@Override

public void onClick(View view) {

```

```

    if (view.getId() == R.id.btnEnviar) {
        mostrarDialogo();
    }
}

```

mostrarDialogo(): Este método muestra un cuadro de diálogo que solicita al usuario ingresar un número telefónico. Se crea un cuadro de texto de entrada y se configuran los botones de "Enviar" y "Cancelar". Al hacer clic en "Enviar", se obtiene el número ingresado y se llama al método enviarMensaje() para enviar el mensaje.

```

private void mostrarDialogo() {
    AlertDialog.Builder builder = new AlertDialog.Builder(this);
    builder.setTitle("Enviar Mensaje");
    builder.setMessage("Ingresa el número telefónico");

    // Crear el elemento de entrada de texto
    final EditText etNumero = new EditText(this);
    etNumero.setInputType(InputType.TYPE_CLASS_PHONE);
    etNumero.setHint("Número telefónico");

    // Agregar el elemento de entrada de texto al cuadro de diálogo
    LinearLayout layout = new LinearLayout(this);
    layout.setOrientation(LinearLayout.VERTICAL);
    layout.addView(etNumero);
    builder.setView(layout);

    // Agregar botones de "Enviar" y "Cancelar"

```

```

builder.setPositiveButton("Enviar", new DialogInterface.OnClickListener() {
    @Override
    public void onClick(DialogInterface dialog, int which) {
        String numero = etNumero.getText().toString();
        enviarMensaje(numero, markerText); // Llamar al método
        enviarMensaje con los valores ingresados
    }
});
builder.setNegativeButton("Cancelar", null);

// Mostrar el cuadro de diálogo
builder.create().show();
}

public EnviarMensaje() {
    // Constructor sin argumentos
}
}

```

enviarMensaje(String numero, String mensaje): Este método envía un mensaje de texto a través de SMS. Utiliza la clase SmsManager para enviar el mensaje al número telefónico especificado. Si el envío es exitoso, se muestra un mensaje de confirmación. Si hay algún error, se muestra un mensaje de error y se imprime la traza de la excepción.

```

private void enviarMensaje(String numero, String mensaje) {
    try {
        SmsManager sms = SmsManager.getDefault();
        sms.sendTextMessage(numero, null, mensaje, null, null);
    }
}

```



```

        Toast.makeText(getApplicationContext(), "Mensaje Enviado.",
Toast.LENGTH_LONG).show();

    } catch (Exception e) {

        Toast.makeText(getApplicationContext(), "Mensaje no enviado, datos
incorrectos.", Toast.LENGTH_LONG).show();

        e.printStackTrace();

    }

}

```

obtenerNumeroTelefono(): Este método muestra un cuadro de diálogo para que el usuario ingrese su propio número de teléfono. Al hacer clic en "Enviar", se obtiene el número ingresado y se llama al método enviarMensaje() para enviar un mensaje a ese número.

```

private String obtenerNumeroTelefono() {

    final EditText input = new EditText(this);

    new AlertDialog.Builder(this)

        .setTitle("Ingrese su número de teléfono")

        .setView(input)

        .setPositiveButton("Enviar", new
DialogInterface.OnClickListener() {

            @Override

            public void onClick(DialogInterface dialog, int which) {

                String numero = input.getText().toString();

                // Aquí puedes realizar la validación del número de teléfono
si es necesario

                // ...

```

```

        enviarMensaje(numero, obtenerTextoMarcador());
    }
})

.setNegativeButton("Cancelar", null)

.show();

return ""; // Puedes devolver un valor predeterminado o manejarlo
de otra manera según tus necesidades
}

```

guardarTextoMarcador(): Este método muestra un cuadro de diálogo donde el usuario puede ingresar el texto del marcador. Al hacer clic en "Guardar", se obtiene el texto ingresado y se guarda en la base de datos llamando al método guardarEnBaseDeDatos() de la clase ActivityBD.

```

// EnviarMensaje.java

private void guardarTextoMarcador() {
    final EditText input = new EditText(this);

    new AlertDialog.Builder(this)
        .setTitle("Ingrese el texto del marcador")
        .setView(input)
        .setPositiveButton("Guardar", new DialogInterface.OnClickListener() {
            @Override
            public void onClick(DialogInterface dialog, int which) {
                markerText = input.getText().toString();
                // Aquí puedes llamar a la función para guardar en la base de

```

*datos*

```

        if (activityBD != null) {
            activityBD.guardarEnBaseDeDatos(markerText, rutaImagen);
        }
    }
})
.setNegativeButton("Cancelar", null)
.show();
}

```

Estos son los métodos más significativos de la clase EnviarMensaje. Cada uno cumple una función específica relacionada con el envío de mensajes SMS y la interacción con el usuario para ingresar y guardar información.

4. **ActivityBD**: Es una clase que extiende de AppCompatActivity e implementa las interfaces View.OnClickListener y LocationListener. Veamos los aspectos más importantes de esta clase:

Declaración de variables y elementos de la interfaz: La clase tiene varias variables miembro que representan diferentes elementos de la interfaz de usuario, como TextView, Button, ImageView, etc. Estos elementos se inicializan en el método onCreate() mediante la llamada a findViewById() para asignarlos a los elementos correspondientes en el diseño de la actividad (R.layout.bd\_gps).

```

public class ActivityBD extends AppCompatActivity implements
View.OnClickListener, LocationListener {

```

```

    TextView tDireccion;

```

```

    Button bGuardar, blistar, btnCamera;

```

```

    FloatingActionButton bPasar;

    private ImageView imgView;

    BaseDatos BDGPS;

    String rutaImagen;

    private static final int REQUEST_CAMERA_PERMISSION = 1;

    private static final int REQUEST_TAKE_PHOTO = 1;

```

Base de datos: La clase utiliza una instancia de la clase BaseDatos para interactuar con una base de datos. La clase utiliza métodos como: agregar(), guardarEnBaseDeDatos(), obtenerListaCoordenadas(), mostrarListaCoordenadas() para agregar datos a la base de datos, guardarlos, obtener una lista de coordenadas y mostrar la lista de coordenadas respectivamente.

```

    public void agregar(String nuevaEntrada) {
        guardarEnBaseDeDatos(nuevaEntrada, rutaImagen);
    }

    public void guardarEnBaseDeDatos(String nuevaEntrada, String
rutaImagen) {
        if (BDGPS.addData(nuevaEntrada, rutaImagen)) {
            Toast.makeText(getApplicationContext(), "Datos guardados
correctamente", Toast.LENGTH_LONG).show();

            actualizarListView();
        } else {
            Toast.makeText(getApplicationContext(), "Error al guardar los
datos", Toast.LENGTH_LONG).show();
        }
    }
}

```

```
private ArrayList<String> obtenerListaCoordenadas() {  
    ArrayList<String> listados = new ArrayList<>();  
    Cursor data = BDGPS.getListaContenidos();  
    if (data.getCount() == 0) {  
        Toast.makeText(this, "No hay lista que mostrar",  
Toast.LENGTH_LONG).show();  
    } else {  
        while (data.moveToNext()) {  
            listados.add(data.getString(1));  
        }  
    }  
    return listados;  
}  
  
private void mostrarListaCoordenadas(ArrayList<String> listados) {  
    if (listados.isEmpty()) {  
        Toast.makeText(this, "No hay lista que mostrar",  
Toast.LENGTH_LONG).show();  
    } else {  
        StringBuilder sb = new StringBuilder();  
        for (String item : listados) {  
            sb.append(item).append("\n");  
        }  
        String listaCoordenadas = sb.toString();  
        Toast.makeText(this, listaCoordenadas,  
Toast.LENGTH_LONG).show();  
    }  
}
```

```

    }
}

//Se carga y actualiza la información del ListView

private void actualizarListView() {

    ArrayList<String> listados = obtenerListaCoordenadas();

    ListAdapter listAdapter = new ArrayAdapter<>(this,
    android.R.layout.simple_list_item_1, listados);

    // Actualizar el ListView con el nuevo adaptador

    // Asegúrate de tener un ListView con el id "listView" en tu layout

    ListView listView = findViewById(R.id.listView);

    listView.setAdapter(listAdapter);

}

```

Captura de imágenes: La clase implementa la funcionalidad de captura de imágenes utilizando la cámara del dispositivo. Los métodos `checkCameraPermission()`, `onRequestPermissionsResult()`, `onActivityResult()`, `createImageFile()` y `dispatchTakePictureIntent()` se encargan de solicitar los permisos necesarios, crear un archivo de imagen temporal, iniciar la actividad de la cámara y recibir y procesar la imagen capturada.

```

private void checkCameraPermission() {

    if (ContextCompat.checkSelfPermission(this, Manifest.permission.CAMERA)
    != PackageManager.PERMISSION_GRANTED) {

        ActivityCompat.requestPermissions(this, new

```

```

String[] {Manifest.permission.CAMERA}, REQUEST_CAMERA_PERMISSION);

    } else {

        dispatchTakePictureIntent();

    }

}

@Override

public void onRequestPermissionsResult(int requestCode, @NonNull String[]
permissions, @NonNull int[] grantResults) {

    super.onRequestPermissionsResult(requestCode, permissions, grantResults);

    if (requestCode == REQUEST_CAMERA_PERMISSION) {

        if (grantResults.length > 0 && grantResults[0] ==
PackageManager.PERMISSION_GRANTED) {

            dispatchTakePictureIntent();

        } else {

            Toast.makeText(this, "Permiso de cámara denegado",
Toast.LENGTH_SHORT).show();

        }

    }

}

@Override

protected void onActivityResult(int requestCode, int resultCode, Intent data) {

    super.onActivityResult(requestCode, resultCode, data);

    if (requestCode == REQUEST_IMAGE_CAPTURE && resultCode ==
RESULT_OK) {

        if (data != null) {

            Bundle extras = data.getExtras();

```

```

        if (extras != null) {
            Bitmap imageBitmap = (Bitmap) extras.get("data");
            imageView.setImageBitmap(imageBitmap);
            if (currentPhotoPath != null) {
                rutaImagen = currentPhotoPath;
            }
        }
    }
}

String currentPhotoPath;

private File createImageFile() throws IOException {
    String timeStamp = new SimpleDateFormat("yyyyMMdd_HHmmss",
        Locale.getDefault()).format(new Date());
    String imageFileName = "JPEG_" + timeStamp + "_";
    File storageDir =
        getExternalFilesDir(Environment.DIRECTORY_PICTURES);
    File image = File.createTempFile(imageFileName, ".jpg", storageDir);
    currentPhotoPath = image.getAbsolutePath();
    return image;
}

private void dispatchTakePictureIntent() {
    Intent takePictureIntent = new

```



```

Intent(MediaStore.ACTION_IMAGE_CAPTURE);

    if (takePictureIntent.resolveActivity(getPackageManager()) != null) {

        File photoFile = null;

        try {

            photoFile = createImageFile();

        } catch (IOException ex) {

            ex.printStackTrace();

        }

        if (photoFile != null) {

            // Obtener la ruta de la imagen usando FileProvider

            Uri photoURI = FileProvider.getUriForFile(this,
"durmot.maps.mapf.fileprovider", photoFile);

            rutaImagen = photoURI.toString(); // Asignar la ruta de la imagen a la
variable rutaImagen

            takePictureIntent.putExtra(MediaStore.EXTRA_OUTPUT, photoURI);

            startActivityForResult(takePictureIntent,
REQUEST_IMAGE_CAPTURE);

        }

    }

}

```

Localización: La clase implementa el método `onLocationChanged()` de la interfaz `LocationListener`, que se invoca cuando hay cambios en la ubicación del dispositivo. En este método, se obtiene la latitud y longitud de la ubicación y se utiliza un objeto `Geocoder` para obtener la dirección correspondiente a esas coordenadas. Luego, se actualiza un `TextView` con la información de ubicación y se muestra la ubicación en un mapa mediante el uso de la clase `FragmentMaps`.

*@Override*

```
public void onLocationChanged(@NonNull Location location) {  
  
    double lat = location.getLatitude();  
  
    double lon = location.getLongitude();  
  
    Geocoder geocoder = new Geocoder(this, Locale.getDefault());  
  
    try {  
  
        List<Address> addresses = geocoder.getFromLocation(lat, lon, 1);  
  
        if (!addresses.isEmpty()) {  
  
            Address address = addresses.get(0);  
  
            String direccionExacta = address.getAddressLine(0);  
  
            String texto = "Mi ubicación es: \n"  
                + "Latitud = " + lat + "\n"  
                + "Longitud = " + lon + "\n"  
                + "Dirección: " + direccionExacta;  
  
            tDireccion.setText(texto);  
  
        }  
  
    } catch (IOException e) {  
  
        e.printStackTrace();  
  
    }  
}
```

```
        showLocationOnMap(lat, lon);  
  
    }  
  
}
```

```
private void showLocationOnMap(double lat, double lon) {  
  
    FragmentMaps fragment = new FragmentMaps();  
  
    Bundle bundle = new Bundle();  
  
    bundle.putDouble("lat", lat);  
  
    bundle.putDouble("lon", lon);  
  
    fragment.setArguments(bundle);  
  
  
    FragmentManager fragmentManager = getSupportFragmentManager();  
  
    FragmentTransaction fragmentTransaction =  
fragmentManager.beginTransaction();  
  
    fragmentTransaction.replace(R.id.map, fragment);  
  
    fragmentTransaction.commitAllowingStateLoss();  
  
    }  
  
}
```

Estos son los aspectos más relevantes de la clase ActivityBD. La clase combina la funcionalidad de almacenamiento de datos en una base de datos, captura de imágenes y seguimiento de la ubicación del dispositivo.

5. **BaseDatos**: es una subclase de SQLiteOpenHelper que se utiliza para crear y manejar una base de datos SQLite en la aplicación. Aquí está la explicación de la clase:

Variables:

BD\_version: Es una variable que representa la versión de la base de datos.

TABLE\_NOMBRE: Es una constante que define el nombre de la tabla en la base de datos.

COLUMN\_ID, COLUMN\_DIRECCIÓN, COLUMN\_RUTA\_IMAGEN: Son constantes que representan los nombres de las columnas en la tabla.

Métodos:

BaseDatos: Es el constructor de la clase. Recibe un contexto como parámetro y llama al constructor de la clase padre SQLiteOpenHelper, pasando el nombre de la base de datos, la versión y el contexto.

onCreate: Es llamado cuando la base de datos se crea por primera vez. Aquí se ejecuta una sentencia SQL para crear la tabla en la base de datos utilizando el nombre de la tabla y los nombres de las columnas definidos en las variables.

*@Override*

```
public void onCreate(SQLiteDatabase db) {

    String sql = "CREATE TABLE " + TABLE_NOMBRE
                + " (" + COLUMN_ID + " INTEGER PRIMARY KEY
    AUTOINCREMENT, "
                + COLUMN_DIRECCIÓN + " VARCHAR (100), "
                + COLUMN_RUTA_IMAGEN + " VARCHAR (100)); ";

    db.execSQL(sql);
}
```

onUpgrade: Es llamado cuando se actualiza la versión de la base de datos. En este caso, se elimina la tabla existente y se llama a onCreate para crear una nueva tabla.

*@Override*

```
public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {
    String sql = "DROP TABLE IF EXISTS SQLiteGPS";
    db.execSQL(sql);
    onCreate(db);
}
```

addData: Permite agregar una nueva entrada a la base de datos. Recibe una nueva dirección y una ruta de imagen como parámetros. Obtiene una instancia de la base de datos en modo escritura, crea un objeto ContentValues y agrega los valores de dirección y ruta de imagen. Luego, utiliza el método insert para insertar los valores en la tabla y devuelve un resultado indicando si la operación fue exitosa.

```
public boolean addData(String nuevaEntrada, String rutaImagen) {
    SQLiteDatabase db = getWritableDatabase();
    ContentValues contentValues = new ContentValues();
    contentValues.put(COLUMN_DIRECCIÓN, nuevaEntrada);
    contentValues.put(COLUMN_RUTA_IMAGEN, rutaImagen);

    long result = db.insert(TABLE_NOMBRE, null, contentValues);
    return result != -1;
}
```

getListaContenidos: Obtiene un Cursor que contiene todos los datos de la tabla. Se obtiene una instancia de la base de datos en modo escritura y se ejecuta una consulta SQL para seleccionar todos los registros de la tabla. El Cursor resultante se devuelve como

resultado.

```
public Cursor getListaContenidos(){
    SQLiteDatabase db = this.getWritableDatabase();
    Cursor data = db.rawQuery("SELECT * FROM "+ TABLE_NOMBRE,null);
    return data;
}
```

En cuanto al modelo de base de datos, se trata de una base de datos relacional utilizando SQLite. La clase BaseDatos extiende SQLiteOpenHelper, que es una clase proporcionada por Android para interactuar con bases de datos SQLite en la aplicación.

6. **MarkerInputDialog:** Es un diálogo emergente en Android que permite al usuario ingresar información para un marcador en un mapa. Aquí te explico de manera sencilla y clara los métodos más significativos de esta clase:

onCreateDialog(Bundle savedInstanceState): Este método se llama cuando se crea el diálogo. Aquí se configura el diseño y el contenido del diálogo, que incluye un campo de texto de entrada donde el usuario puede ingresar información para el marcador. Se establecen los botones "Guardar" y "Cancelar" y se maneja la lógica de clics en estos botones.

```
private MarkerInputListener listener;

@NonNull
@Override
public Dialog onCreateDialog(@Nullable Bundle savedInstanceState) {
    AlertDialog.Builder builder = new AlertDialog.Builder(requireActivity());
    builder.setTitle("Ingresar información");

    final EditText input = new EditText(requireActivity());
```

```
input.setInputType(InputType.TYPE_CLASS_TEXT);
builder.setView(input);

builder.setPositiveButton("Guardar", new DialogInterface.OnClickListener()
{
    @Override
    public void onClick(DialogInterface dialog, int which) {
        String text = input.getText().toString();
        if (listener != null) {
            listener.onMarkerInputEntered(text);
        }
    }
});

builder.setNegativeButton("Cancelar", new
DialogInterface.OnClickListener() {
    @Override
    public void onClick(DialogInterface dialog, int which) {
        dialog.cancel();
    }
});

return builder.create();
}
```

`setMarkerInputListener(MarkerInputListener listener)`: Este método permite establecer un objeto `MarkerInputListener` que actuará como un listener para capturar la entrada del usuario en el diálogo. Al establecer un listener, se puede recibir la información ingresada por el usuario mediante el método `onMarkerInputEntered()` del listener.

```
public void setMarkerInputListener(MarkerInputListener listener) {  
    this.listener = listener;  
}  
}
```

`MarkerInputListener`: Esta es una interfaz definida en la clase `MarkerInputDialog`. Permite definir un método `onMarkerInputEntered(String text)` que debe ser implementado por cualquier clase que implemente la interfaz. Este método se llama cuando el usuario hace clic en el botón "Guardar" del diálogo y se pasa la información ingresada como argumento.

```
public interface MarkerInputListener {  
    void onMarkerInputEntered(String text);  
}
```

En resumen, la clase `MarkerInputDialog` proporciona un diálogo de entrada de texto donde el usuario puede ingresar información para un marcador en un mapa. Permite establecer un listener para capturar la información ingresada por el usuario y realizar las acciones necesarias con ella, como guardarla en una base de datos o actualizar el marcador en el mapa.



## 6.4 Pruebas

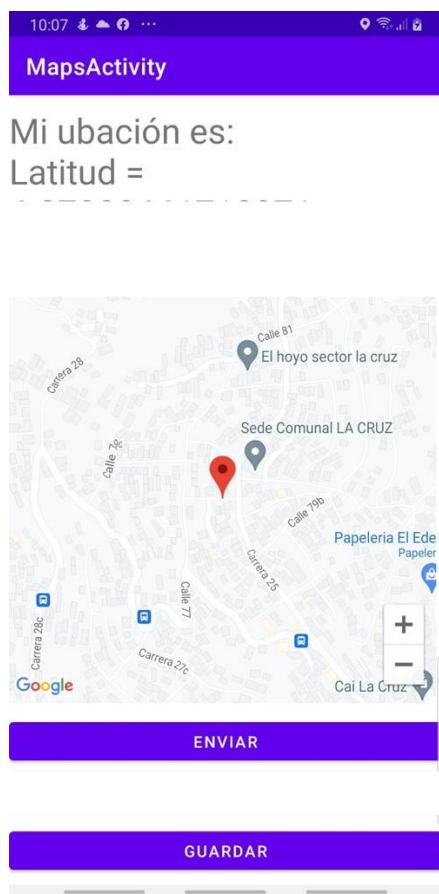


Figura 17. Marcador fijado en la ubicación exacta de un dispositivo físico



Figura 18. Ubicación en tiempo real que será almacenada en la base de datos

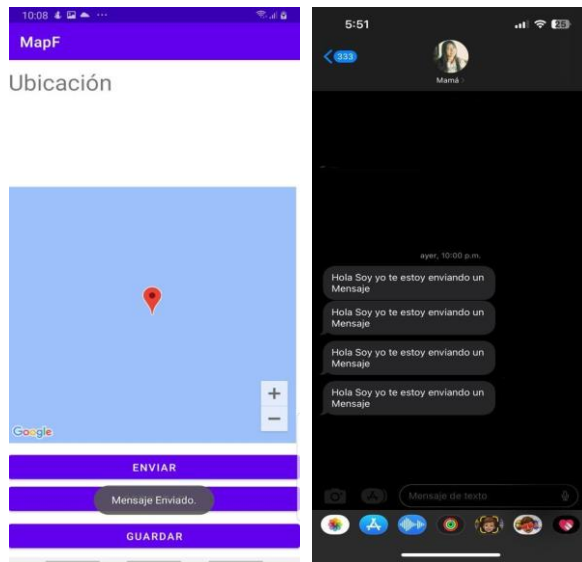


Figura 19. Envío de mensaje de prueba

## 7 Conclusiones

1. La aplicación móvil tiene el potencial de ser una herramienta efectiva para mejorar la movilidad en la ciudad de Medellín al permitir a los usuarios reportar incidentes de tránsito de manera rápida y precisa.

2. Las características y funcionalidades de la aplicación, como una interfaz fácil de usar, integración con mapas y geolocalización, notificaciones en tiempo real y la opción de adjuntar fotos o videos de los incidentes, son fundamentales para garantizar una experiencia óptima para los usuarios.

3. La conexión a una base de datos confiable y actualizada es esencial para almacenar y procesar la información de los incidentes reportados, lo que permitirá a las autoridades tomar medidas adecuadas y mejorar la gestión del tráfico.

4. Es importante considerar los desafíos y dificultades que podrían surgir al utilizar la aplicación, como la privacidad de los datos personales de los usuarios y garantizar que la aplicación sea accesible para todos, incluyendo a aquellos con poca experiencia tecnológica.

5. La colaboración entre los usuarios y las autoridades es crucial para lograr una mejora significativa en la movilidad de la ciudad. La opción de interactuar y compartir información con otros usuarios en la aplicación fomenta la creación de una comunidad comprometida con la seguridad vial y la eficiencia en el tráfico.

En resumen, el desarrollo de la aplicación móvil para el reporte de incidentes de tránsito en Medellín se presenta como una solución innovadora que puede contribuir a mejorar la movilidad en la ciudad al proporcionar una plataforma para la comunicación rápida y precisa de los incidentes de tránsito. Con la implementación adecuada y la participación activa de los usuarios, la aplicación tiene el potencial de hacer una diferencia significativa en la seguridad y eficiencia del tráfico en Medellín.

## 8 Recomendaciones

1. Asegúrate de tener una cuenta de facturación activa en Google Cloud. Si no la tienes, debes crear una antes de empezar. Ten en cuenta que algunos servicios de Google Cloud tienen un costo asociado, por lo que es importante que conozcas los precios y limitaciones de cada uno antes de utilizarlos.

2. Utiliza la documentación de Google Cloud para entender cómo funcionan los servicios que necesitas utilizar, como por ejemplo Google Maps Platform, Cloud Functions y Firebase. Si la documentación es limitada, busca tutoriales en línea, foros y grupos de discusión para obtener más información y resolver posibles dudas.

3. Asegúrate de que tu proyecto tenga una estructura bien definida y separada por capas, para facilitar el mantenimiento y la escalabilidad en el futuro. Es recomendable utilizar patrones de diseño como *cliente-servidor* para la arquitectura de tu aplicación móvil.

4. Aprovecha las funcionalidades de Google Maps Platform para mejorar la precisión y la calidad de los reportes de incidentes de tránsito. Puedes utilizar la API de Geolocalización para obtener la ubicación del usuario y la API de Direcciones para obtener la ruta más adecuada para llegar al destino.

5. Utiliza Cloud Functions para crear servicios que se ejecuten en la nube y que interactúen con Firebase, por ejemplo, para enviar notificaciones push a los usuarios de la aplicación.

6. Realiza pruebas de rendimiento y seguridad para garantizar que tu aplicación móvil funcione correctamente y que no haya posibles vulnerabilidades que puedan afectar a los usuarios.

Recuerda que la implementación de una aplicación móvil es un proceso complejo que requiere tiempo, dedicación y conocimientos técnicos avanzados. Si no tienes experiencia en desarrollo de software, es recomendable que busques ayuda de un equipo de profesionales o que te formes adecuadamente antes de comenzar.

## 9 Bibliografía

*Plano Georeferenciado* ▷REGISTRO-NOTARIA-CATASTRO. (n.d.). Arkespai. Retrieved May 26, 2022, from <https://www.arkespai.com/plano-georreferenciado/>

Banco Interamericano de Desarrollo. (2019, Febrero). *CÓMO APLICAR BIG DATA EN LA PLANIFICACIÓN DEL TRANSPORTE URBANO EL USO DE DATOS DE TELEFONÍA MÓVIL EN EL ANÁLISIS DE LA MOV.* Publications. Retrieved April 20, 2022, from [https://publications.iadb.org/publications/spanish/document/Como\\_aplicar\\_big\\_data\\_en\\_la\\_planificaci%C3%B3n\\_del\\_transporte\\_urbano\\_El\\_uso\\_de\\_datos\\_de\\_telefon%C3%ADa\\_m%C3%B3vil\\_en\\_el\\_an%C3%A1lisis\\_de\\_la\\_movilidad\\_es.pdf](https://publications.iadb.org/publications/spanish/document/Como_aplicar_big_data_en_la_planificaci%C3%B3n_del_transporte_urbano_El_uso_de_datos_de_telefon%C3%ADa_m%C3%B3vil_en_el_an%C3%A1lisis_de_la_movilidad_es.pdf)

Katchadourian, C. Daniel, J.(2020). Tecnologías para luchar contra la pandemia Covid-19: geolocalización, rastreo, big data, SIG, inteligencia artificial y privacidad. Recuperado el 2 de marzo de 2022, de: <http://eprints.rclis.org/40350/>

Torres, D. Hidalgo, C.(2020). Desarrollo de una herramienta de evaluación incorporando algoritmos de clusterización de tráfico vehicular con parámetros de rendimiento, infraestructura usada, procesamiento y resultados obtenidos de la geolocalización de los algoritmos implementados usando herramientas de Big Data sobre infraestructura Cloud. Tesis de Ingeniería en Sistemas Computacionales. Universidad de Guayaquil.

Romero, F. Urroz, G.(2020). Acceso al dato. BIG DATA en aplicaciones móviles de geolocalización y experiencia de usuarios. SI+ Herramientas y Procedimientos. Pág 2485. Recuperado el 2 dde marzo de 2022, de: <https://publicacionescientificas.fadu.uba.ar/index.php/actas/article/download/1618/2230>.

Suderow, J.(2021).Inteligencia artificial y competencia, acceso a datos en el ejemplo de la movilidad y del vehículo autónomo. Tesis de ingeniería . Universidad de Deuston, Bilbao.

Guanxia, X., Shiyi, G., Daneshmad, M., Chonggang, W., & Yanbing, L. (2016, 11 3). A Survey for Mobility Big Data Analytics for Geolocation Prediction. *IEEE Wireless Communications*, 24(1), 111-119. <https://uvadoc.uva.es/handle/10324/48573>

Aldana Porras, J. M., Ferreira Manzanares, N. A., & Montes Mora, J. F. (2016, Diciembre). SISTEMA DE NAVEGACIÓN Y EXPLORACIÓN VISUAL ARTICULADO POR TÉCNICAS DE INTELIGENCIA ARTIFICIAL, PARA LA MOVILIDAD DE PERSONAS EN CONDICIÓN DE DISCAPACIDAD VISUAL. *ITFIPVIRTUAL*, 1. <https://revistasitfip.comunisoft.com/index.php/reite/article/view/2/4>

Onieva, E., Serrano, L., Landaluce, H., D. Masegosa, A., & Moreno, A. (2019). *Impacto social*. Rosa Maria Santibañez Gruber. [https://web.archive.org/web/20210217064013id\\_/https://www.deusto.es/cs/Satellite?blobcol=urldata&blobheader=application%2Fpdf&blobheadername1=Expires&blobheadername2=content-type&blobheadername3=MDT-Type&blobheadername4=Content-Disposition&blobheadervalue1](https://web.archive.org/web/20210217064013id_/https://www.deusto.es/cs/Satellite?blobcol=urldata&blobheader=application%2Fpdf&blobheadername1=Expires&blobheadername2=content-type&blobheadername3=MDT-Type&blobheadername4=Content-Disposition&blobheadervalue1)

Silva, I., & Segarra, J. (2020, 2 13). *Modelo predictivo en los accidentes de transito con basa en data science*. Repositorio Digital de la Universidad de Especialidades del Espíritu Santo. Retrieved 3 3, 2022, from <http://repositorio.uees.edu.ec/123456789/3192>

Ouallane, A. A., Bahnasse, A., & Bakali, A. (2022). Overview of Road Traffic Management Solutions based on IoT and AI. *Procedia Computer Science*, 198. <https://doi.org/10.1016/j.procs.2021.12.279>

Marques, V. L., & Graeml, A. R. (2017). Systematic Literature Review. *Collective Intelligence and Accessible Ways in the City*. [https://pessoal.dainf.ct.utfpr.edu.br/graeml/ParticipacaoEventos/AMCIS/AMCIS2017\\_Boston/AMCIS2017\\_MappingOfAccessibleWaysInTheCity.pdf](https://pessoal.dainf.ct.utfpr.edu.br/graeml/ParticipacaoEventos/AMCIS/AMCIS2017_Boston/AMCIS2017_MappingOfAccessibleWaysInTheCity.pdf)

Montealegre, A. (2017). *Importancia de la solución Big Data en la aplicación de movilidad Uber movement*. Recuperado de: <https://hdl.handle.net/10901/11203>.

Bandrés Cristina, (2019). *El origen y la evolución de la movilidad de las personas con la utilización de las nuevas tecnologías*. Recuperado de:

<http://uvadoc.uva.es/handle/10324/37435>

Salazar Yuliana & Calle Daniel,( 2018 ).*Sistema experto para la gestión de estrategias de movilidad en el municipio de Itagüí*. Recuperado de:

<https://dspace.tdea.edu.co/bitstream/handle/tda/428/Sistema%20experto%20para%20la%20gestion%20de%20estrategias%20para%20la%20movilidad.pdf?sequence=1&isAllowed=y>

Linares, A. (2020). *Aplicabilidad de herramientas de Big Data y Business Analytics al sector de la ecomovilidad urbana: El caso BiciMAD*. Repositorio comillas. Retrieved 03 03, 2022, from <https://repositorio.comillas.edu/xmlui/handle/11531/38079>

*Visión artificial: qué es, aplicaciones*. (2022). Iberdrola. Retrieved May 26, 2022, from <https://www.iberdrola.com/innovacion/vision-artificial>

*Procesamiento digital de imágenes*. (2020). Gavilán. Retrieved October 16, 2020, from <https://ignaciogavilan.com/principales-aportaciones-de-la-vision-artificial/>

*Big Data – Tecnologías de la Información y Procesos de Negocios*. (2018, March 2). Tecnologías de la Información y Procesos de Negocios. Retrieved May 26, 2022, from <https://msaffirio.com/2018/03/02/big-data/>

*El semáforo de la conciencia: no se pone verde si no llevas el casco en la moto*. (2021, April 20). El Confidencial. Retrieved May 26, 2022, from [https://www.elconfidencial.com/alma-corazon-vida/2021-04-20/semaforo-conciencia-moto-argentina-casco\\_3041899/](https://www.elconfidencial.com/alma-corazon-vida/2021-04-20/semaforo-conciencia-moto-argentina-casco_3041899/)

## 10 Anexos



### Aplicación de Reporte de Incidentes de Tráfico

La Aplicación de Reporte de Incidentes de Tráfico es una herramienta móvil diseñada para mejorar la movilidad en la ciudad de Medellín. Permite a los usuarios reportar diferentes tipos de incidentes de tráfico en tiempo real y recibir información actualizada sobre incidentes en su ruta. Este manual proporciona instrucciones sobre cómo utilizar la aplicación de manera efectiva.

#### Requisitos del Sistema

- Dispositivo móvil con sistema operativo Android.
- Acceso a Internet para la descarga y actualización de la aplicación.
- Permiso de ubicación activado para agregar automáticamente la ubicación del incidente al reporte.

- Si lo desea, puede adjuntar una foto del incidente haciendo clic en el botón "Tomar Foto".
- Haga clic en el botón "Guardar" para enviar el reporte del incidente.

#### Instalación

Siga los pasos a continuación para instalar la aplicación en su dispositivo móvil:

- Abra la tienda de aplicaciones en su dispositivo móvil.
- Busque "Aplicación de Reporte de Incidentes de Tráfico".
- Seleccione la aplicación y haga clic en "Instalar".
- Espere a que se complete la instalación.

#### Lista de Incidentes Reportados

- Para ver la lista de incidentes reportados, haga clic en el botón "Listar Incidentes".
- Se mostrará una lista de los incidentes reportados por los usuarios.

#### Visualización de Incidentes en el Mapa

- El mapa mostrará marcadores que representan los incidentes reportados.
- Puede acercar o alejar el mapa para ver más detalles.
- Haga clic en un marcador para ver información específica sobre un incidente.

#### Reporte de Incidentes de Tráfico

- Una vez que haya iniciado sesión, se le presentará la pantalla principal de la aplicación.
- Para reportar un incidente de tráfico, haga clic en el botón "Reportar Incidente".
- Se abrirá una nueva pantalla donde podrá agregar los detalles del incidente.
- Proporcione una descripción breve del incidente en el campo correspondiente.
- Haga clic en el botón "Activar Ubicación" para agregar automáticamente la ubicación actual al reporte.

#### Notificaciones de Incidentes en Ruta

- La aplicación enviará notificaciones cuando haya incidentes reportados en su ruta, a través de mensajes de texto.

### Anexo 1. Manual de usuario