



UNIVERSIDAD  
DE PIURA

**FACULTAD DE INGENIERÍA**

**Desarrollo de una aplicación móvil para monitoreo en tiempo real del tráfico en la ciudad de Piura, Perú**

Tesis para optar el Título de  
Ingeniero Industrial y de Sistemas

**Juan Diego Espinoza Yovera**

**Asesor:**  
**Dra. Ing. Rosa Peregrina Gonzáles Martínez**

**Piura, julio de 2025**

### Declaración Jurada de Originalidad del Trabajo Final

Yo, Juan Diego Espinoza Yovera, egresado del Programa Académico de Ingeniería Industrial y de Sistemas de la Facultad de Ingeniería de la Universidad de Piura, identificado(a) con DNI: 71242609, declaro que:

Soy autor del trabajo final titulado:

**“Desarrollo de una aplicación móvil para monitoreo en tiempo real del tráfico en la ciudad de Piura, Perú”**

El mismo que presento bajo la modalidad de Tesis para optar el Título profesional de Ingeniero Industrial y de Sistemas.

El texto de mi trabajo final es original y no vulnera los derechos de terceros o, de ser el caso, derechos de los coautores, incluidos los derechos de propiedad intelectual, datos personales, entre otros. En tal sentido, el texto de mi trabajo final no ha sido plagiado total ni parcialmente, para lo cual, he respetado las normas internacionales de citas y referencias de las fuentes consultadas. Asimismo, el texto del trabajo final que presento no ha sido publicado ni presentado antes en cualquier medio electrónico o físico; y que la investigación, los resultados, datos, conclusiones y demás información presentada que atribuyo a mi autoría son veraces.

En caso de detectarse el incumplimiento de lo declarado asumo frente a terceros, la Universidad de Piura y/o la Administración Pública toda responsabilidad que pueda derivarse por el trabajo final presentado. Lo señalado incluye responsabilidad pecuniaria incluido el pago de multas u otros por los daños y perjuicios que se ocasionen.

La asesoría del trabajo estuvo a cargo de los siguientes docentes de la Universidad de Piura:

- Dra. Ing. Rosa Peregrina Gonzáles Martínez, identificado (a) con DNI: 18122549

Declaro (declaramos) que:

Luego de haber empleado el software de coincidencia Turnitin, revisado las fuentes de información señaladas por el autor, y en razón de mi (nuestra) experiencia como investigador(es), declaro (declaramos) que las ideas expuestas en el trabajo final alcanzan las condiciones de calidad, integridad y originalidad acorde a los objetivos institucionales y estándares en materia de investigación. Finalmente, no asumo (asumimos) responsabilidad por la posible vulneración de derechos de autor en el trabajo final referido, pues tal responsabilidad es exclusiva del autor.

Fecha: 01/07/2025.



.....  
Firma del autor<sup>1</sup>



.....  
Firma del asesor<sup>1</sup>

<sup>1</sup> Firma idéntica al DNI. No se admite digital, salvo certificado.

## **Dedicatoria**

Dedico este trabajo con profundo agradecimiento y cariño a mi familia, por ser mi motor constante y brindarme su apoyo incondicional en cada etapa de este camino. A mis padres, por sus sacrificios, enseñanzas y ejemplo de perseverancia.

A mis amigos, por su compañía y por recordarme que siempre hay espacio para una sonrisa incluso en las jornadas más exigentes.

A mi pareja, por su paciencia, amor y por estar a mi lado en cada paso de este proceso.



## **Agradecimientos**

Agradezco a mi familia por su apoyo incondicional, a mi asesora por su guía durante este proceso, y a mis amigos y pareja por su compañía y motivación constante. También extiendo mi gratitud a todas las personas que participaron en el desarrollo y validación de esta investigación.



## Resumen

El presente trabajo de investigación tiene como objetivo desarrollar una aplicación móvil para el monitoreo del tráfico vehicular en tiempo real en la ciudad de Piura, Perú, como respuesta a los crecientes problemas de congestión causados por el aumento del parque automotor, obras en la vía pública y eventos climáticos. La solución propuesta se basa en el uso de tecnologías como Flutter, Firebase, Mapbox y servicios GPS, bajo una arquitectura Clean Architecture que garantiza escalabilidad, mantenimiento y modularidad. La aplicación permite a los usuarios reportar incidencias de tráfico y visualizar el estado vial en tiempo real mediante un enfoque crowdsourcing. El sistema fue evaluado utilizando el modelo Petkins, arrojando un promedio de usabilidad de 4.57, evidenciando alta aceptación por parte de los usuarios.

Asimismo, se realizó un análisis espacial de los reportes, identificando zonas críticas de congestión. Los resultados demuestran que la herramienta propuesta es efectiva y útil para mejorar la movilidad urbana, con potencial de ampliación y mejora mediante la incorporación de inteligencia artificial y validación automática de incidentes.



## Tabla de contenido

Introducción .....	10
Capítulo 1 Introducción .....	11
1.1 Antecedentes.....	11
1.2 Justificación .....	12
1.3 Planteamiento del problema.....	12
1.4 Objetivos.....	13
1.5 Alcance.....	13
1.5.1 <i>Funcionalidades</i> .....	13
1.5.2 <i>Tareas</i> .....	14
1.5.3 <i>Exclusiones</i> .....	14
Capítulo 2 Marco teórico.....	15
2.1 Generalidades del monitoreo del tráfico vehicular.....	15
2.2 Modelo de datos.....	16
2.2.1 <i>Modelo de datos basados en Inteligencia Artificial</i> .....	16
2.2.2 <i>Modelo de datos geoespaciales</i> .....	17
2.2.3 <i>Participación ciudadana y Crowdsourcing</i> .....	18
2.3 Modelo aplicado .....	20
2.4 Tecnologías utilizadas .....	20
2.4.1 <i>Sistema de posicionamiento global (GPS)</i> .....	20
2.4.2 <i>Comunicación Inalámbrica</i> .....	21
2.4.3 <i>Integración de APIS y servicios web</i> .....	21
2.5 Enfoques y modelos tecnológicos .....	21
2.5.1 <i>Internet de las cosas (IoT)</i> .....	21
2.5.2 <i>Arquitectura Cliente – Servidor</i> .....	22
2.5.3 <i>Arquitectura de Aplicaciones móviles</i> .....	23
2.5.4 <i>Arquitectura Clean Architecture</i> .....	23
2.5.5 <i>Análisis en tiempo real (Real Time Data Analytics)</i> .....	24
2.5.6 <i>Servidor web y servicios en la nube</i> .....	25
2.6 Seguridad y privacidad.....	25
2.6.1 <i>Protección de datos del usuario</i> .....	26
2.6.2 <i>Seguridad de la comunicación</i> .....	26
2.7 Marco conceptual.....	27
Capítulo 3 Marco metodológico.....	29
3.1 Planteamiento del problema.....	29
3.2 Tipo y diseño de investigación.....	29
3.3 Población, muestra y marco muestral.....	30
3.4 Técnicas e instrumentos de recolección de datos .....	30
3.5 Herramientas informáticas .....	30
3.6 Procedimientos.....	31
3.7 Método .....	31
3.8 Evaluación de usabilidad .....	32
3.9 Método de evaluación de usabilidad .....	32
Capítulo 4 Análisis del sistema .....	35
4.1 Análisis de causas de la congestión vehicular .....	35
4.1.1 <i>Recopilación de información</i> .....	35
4.2 Requisitos del sistema .....	37

4.2.1	<i>Requisitos funcionales</i> .....	37
4.2.2	<i>Requisitos no funcionales</i> .....	38
4.3	Diagramas de casos de uso.....	38
4.4	Diagrama de procesos .....	40
4.4.1	<i>Diagrama de proceso: Registro de usuario (CU1)</i> .....	42
4.4.2	<i>Diagrama de proceso: Inicio de sesión (CU2)</i> .....	43
4.4.3	<i>Diagrama de proceso: Visualización del mapa de tráfico (CU3)</i> .....	44
4.4.4	<i>Diagrama de proceso: Reporte de incidencias (CU4)</i> .....	46
4.4.5	<i>Diagrama de procesos: Generación de reportes (CU5)</i> .....	47
4.4.6	<i>Diagrama de procesos: Recepción de notificaciones (CU6)</i> .....	48
Capítulo 5 Diseño del Sistema .....		50
5.1	Diseño de la arquitectura .....	50
5.1.1	<i>Arquitectura física</i> .....	50
5.1.2	<i>Arquitectura Lógica (Clean Architecture)</i> .....	50
5.2	Diseño de interfaces .....	51
5.2.1	<i>Pantalla de registro de usuario</i> .....	51
5.2.2	<i>Pantalla de inicio de sesión</i> .....	52
5.2.3	<i>Pantalla de monitoreo</i> .....	53
5.2.4	<i>Pantalla de registro de incidentes</i> .....	53
5.2.5	<i>Pantalla de gestión de usuarios</i> .....	55
5.2.6	<i>Pantalla de generación de reportes</i> .....	56
5.2.7	<i>Dashboard</i> .....	57
5.3	Modelo Entidad Relación.....	57
5.4	Diseño de base de datos.....	59
Capítulo 6 Desarrollo de la aplicación .....		61
6.1	Tecnologías usadas .....	61
6.2	Desarrollo del prototipo .....	62
6.2.1	<i>Estructura general de la aplicación</i> .....	63
6.3	Implementación de pruebas de usabilidad .....	67
6.4	Desarrollo del análisis geográfico en la aplicación .....	68
Capítulo 7 Resultados y Conclusiones .....		69
7.1	Resultados de las pruebas de usabilidad .....	69
7.2	Resultados del análisis geográfico de incidencias .....	70
7.3	Conclusiones .....	71
7.4	Recomendaciones.....	72
7.5	Trabajos Futuros .....	72
Referencias.....		74
Apéndices .....		78
Apéndice 1 Instrumentos de evaluación de usabilidad de la aplicación móvil.....		79
Apéndice 2 Reporte de Incidentes de tráfico período 02/06/2025 – 07/08/2025 .....		81

## Lista de tablas

Tabla 1 Criterios y definiciones según la norma ISO 9241-11.....	33
Tabla 2 Cuestionario de preguntas para medir usabilidad .....	33
Tabla 3 Descripción por escala .....	34
Tabla 4 Tabla de promedio general.....	34
Tabla 5 Código de color para nivel de criticidad .....	45



## Lista de figuras

Figura 1 Crecimiento de Parque Automotriz Piura .....	15
Figura 2 Diagrama de Modelo de Datos basado en Inteligencia Artificial.....	17
Figura 3 Diagrama Modelo de datos Geoespaciales .....	17
Figura 4 Diagrama Crowdsourcing .....	19
Figura 5 Diagrama Arquitectura Cliente - Servidor .....	22
Figura 6 Arquitectura general de desarrollo de aplicaciones móviles.....	23
Figura 7 Representación de arquitectura lógica Clean Architecture .....	24
Figura 8 Diagrama de Ishikawa para la congestión vehicular en Piura, Perú .....	36
Figura 9 Diagrama de casos de uso .....	40
Figura 10 Diagrama de procesos para el perfil Administrador .....	41
Figura 11 Diagrama de procesos para perfil Cliente .....	42
Figura 12 Diagrama de proceso: Registro de Usuario.....	43
Figura 13 Diagrama de proceso: Inicio de Sesión .....	44
Figura 14 Diagrama de proceso: Visualización del mapa de tráfico .....	45
Figura 15 Diagrama de proceso: Reporte de Incidencias.....	46
Figura 16 Diagrama de proceso: Generación de reportes .....	48
Figura 17 Diagrama de proceso: Recepción de notificaciones .....	49
Figura 18 Interfaz de registro de Usuarios .....	52
Figura 19 Interfaz de Inicio de Sesión .....	52
Figura 20 Interfaz Monitoreo de tráfico.....	53
Figura 21 Interfaz de selección de accidente de tránsito .....	54
Figura 22 Interfaz de registro de nivel de criticidad .....	55
Figura 23 Formulario para creación de usuarios.....	56
Figura 24 Módulo de reportes .....	56
Figura 25 Dashboard .....	57
Figura 26 Modelo entidad relación .....	59
Figura 27 Diseño de base de datos relacional.....	60
Figura 28 Arquitectura General de Aplicación Móvil .....	61
Figura 29 Planificación de sprints bajo metodología Scrum .....	63
Figura 30 Estructura general de la aplicación .....	64
Figura 31 Estructura del proyecto basado en Clean Architecture .....	64
Figura 32 Ejemplo de clase de dominio (capa Domain) .....	65
Figura 33 Implementación del repositorio en la capa de datos (Data) .....	65
Figura 34 Evidencia de implementación de Firebase.....	66
Figura 35 Implementación de API Mapbox.....	66
Figura 36 Código de conexión a tablas (collections) de base de datos en Firebase .....	66
Figura 37 Base de datos en Firebase.....	67
Figura 38 Promedio de usabilidad por criterio .....	70
Figura 39 Zonas de concentración de tráfico vehicular .....	71

## Introducción

El crecimiento acelerado del parque automotor en la ciudad de Piura, sumado a la limitada infraestructura vial, obras de construcción no programadas y eventos climáticos como el fenómeno de El Niño, ha generado un incremento significativo en la congestión vehicular. Esta problemática no solo afecta la movilidad urbana, sino que también impacta la calidad de vida de los ciudadanos, la eficiencia del transporte público y la sostenibilidad del entorno.

En respuesta a esta situación, la presente tesis propone el desarrollo de una aplicación móvil para el monitoreo del tráfico en tiempo real, utilizando tecnologías como Mapbox, Firebase y GPS, bajo un enfoque de Clean Architecture y desarrollo ágil. El sistema se fundamenta en el modelo de crowdsourcing, permitiendo a los propios usuarios reportar incidentes viales, facilitando así la recolección colaborativa de datos y la generación de alertas oportunas.

El trabajo se estructura en siete capítulos. En el Capítulo 1 se presenta la problemática, los objetivos y la justificación del estudio. El Capítulo 2 aborda los antecedentes y el marco teórico, explorando conceptos como movilidad urbana, aplicaciones móviles, sistemas de información geográfica y usabilidad. El Capítulo 3 detalla la metodología empleada, sustentada en el enfoque cuantitativo y el uso de encuestas para evaluar la experiencia del usuario.

El Capítulo 4 describe el análisis de requerimientos del sistema, identificando tanto aspectos funcionales como no funcionales. En el Capítulo 5 se desarrolla el diseño del sistema, abarcando desde la arquitectura física y lógica hasta el diseño de interfaces y la estructura de la base de datos. El Capítulo 6 presenta el proceso de implementación del prototipo, haciendo énfasis en las tecnologías utilizadas y la organización del código. Finalmente, en el Capítulo 7 se exponen los resultados obtenidos tras las pruebas de usabilidad, seguidos de conclusiones, recomendaciones y líneas de trabajo futuro que apuntan a la mejora continua del sistema propuesto.

Este estudio busca contribuir a una movilidad más eficiente en Piura mediante el uso estratégico de tecnologías digitales y la participación ciudadana, aportando una herramienta que favorece la toma de decisiones informadas y la gestión urbana inteligente.

## Capítulo 1

### Introducción

El presente trabajo tiene como objetivo el desarrollo de una aplicación móvil para el monitoreo en tiempo real del tráfico en la ciudad de Piura, Perú, con la finalidad de brindar información actualizada del estado del tránsito en sus calles. Debido al notable crecimiento de la congestión vehicular en la ciudad, se ha generado la necesidad de soluciones innovadoras que permitan a los conductores tomar decisiones informadas para evitar rutas congestionadas y favorecer un desplazamiento más eficiente. Ante esta situación, las aplicaciones móviles se presentan como herramientas idóneas, ya que permiten a los usuarios interactuar directamente con el entorno vial mediante reportes de incidencias escalados por la propia ciudadanía.

Este capítulo tiene como propósito presentar una visión general del estudio, abordando los antecedentes, justificación, planteamiento del problema, objetivos y alcance del estudio. La introducción delimita el contexto en el cual se desarrollará la investigación, así como los motivos que justifican la pertinencia y relevancia de la propuesta en la situación actual.

#### 1.1 Antecedentes

En diferentes ciudades del mundo, la congestión vehicular se ha convertido en un desafío para la movilidad urbana, impactando negativamente en la calidad de vida de los ciudadanos. Ciudades como Londres, Nueva York, Ciudad de México y Tokio han implementado sistemas inteligentes de transporte (ITS), compuestos por tecnologías como semáforos adaptativos, sensores de tráfico y aplicaciones móviles, que permiten una gestión más eficiente del tráfico. En otros casos, como Singapur, se ha incorporado el análisis de datos en tiempo real mediante inteligencia artificial para optimizar la movilidad urbana (Land Transport Authority [LTA], 2020). Estos ejemplos demuestran cómo la tecnología puede mejorar significativamente la gestión del tránsito, aunque su implementación requiere inversiones considerables y una infraestructura tecnológica robusta.

En el caso del Perú, los esfuerzos por implementar tecnologías para la gestión del tráfico se han concentrado principalmente en Lima. Allí se han desarrollado soluciones como cámaras inteligentes y aplicaciones como Lima Expresa, que permiten mantener informados a los usuarios sobre el estado de las vías (Municipalidad Metropolitana de Lima, 2021; Lima Expresa, 2022). Investigaciones recientes han demostrado el impacto positivo de estas tecnologías en la gestión vial urbana, así como una buena aceptación por parte de los usuarios (Ydrogo Ramírez, 2020). Un ejemplo concreto es la implementación de semáforos inteligentes, los cuales han contribuido a mejorar la fluidez del tránsito y reducir los tiempos de espera en intersecciones. Sin embargo, estas herramientas aún no se han implementado en otras

regiones del país, debido a limitaciones técnicas y financieras, lo que genera una desigualdad en el acceso a información actualizada sobre el tráfico.

La ciudad de Piura, enfrenta un crecimiento urbano y del parque automotor acelerado, lo que ha generado una creciente congestión vehicular. Aunque muchos ciudadanos recurren a aplicaciones como Google Maps o Waze, la baja cantidad de usuarios activos limita la eficacia de los datos generados (Rojas, 2022). Esta situación evidencia la necesidad de soluciones digitales adaptadas al contexto local, que fomenten la participación ciudadana y brinden información confiable en tiempo real. En este sentido, el desarrollo de una aplicación móvil representa una alternativa viable y significativa para mejorar la gestión del tránsito, siendo una herramienta accesible, participativa y con potencial para futuras mejoras.

## **1.2 Justificación**

La ciudad de Piura atraviesa actualmente un proceso de expansión urbana sostenida. Según el Instituto Nacional de Estadística e Informática (INEI), la población de Piura ha crecido en más de 300 mil habitantes entre 2010 y 2022, lo que ha generado una presión significativa sobre el sistema de transporte. Además, en los últimos 5 años, el parque automotor ha aumentado en más del 40%, incrementando aún más la carga vial (INEI, 2022; SUNARP, 2023).

Por otra parte, el Fenómeno de El Niño, afecta de forma considerable la circulación vial, al generar bloqueos, daños en la infraestructura y restricciones de rutas, lo que complica la movilidad urbana. Ante este escenario, resulta beneficioso contar con una herramienta que proporcione información confiable y en tiempo real sobre el estado del tránsito, permitiendo a los ciudadanos tomar decisiones más acertadas al momento de movilizarse.

La aplicación propuesta se presenta como una solución tecnológica de bajo costo y adaptada al entorno local. Permitirá que los usuarios estar informados mediante reportes en tiempo real sobre el tráfico vehicular, incluyendo eventos como bloqueos, accidentes o vías congestionadas. Esta información también será de utilidad para las autoridades municipales y regionales, quienes podrán emplearla en la planificación de obras viales, establecimiento de rutas alternativas en situaciones de emergencia y diseño de políticas públicas orientadas a una movilidad urbana más eficiente. Basado en los principios de eficiencia, sostenibilidad y participación ciudadana, este proyecto contribuye al desarrollo de una ciudad mejor organizada, resiliente y adaptada a los retos contemporáneos de la movilidad.

## **1.3 Planteamiento del problema**

La congestión vehicular en Piura constituye un problema recurrente que afecta a los ciudadanos en su vida cotidiana. Esta situación se ve agravada por la falta de información actualizada sobre el estado del tránsito, lo que impide a los conductores tomar decisiones acertadas para evitar zonas congestionadas o bloqueadas por obras de mantenimiento. A diferencia de otras ciudades del país que cuentan con infraestructura tecnológica para el monitoreo del tránsito eficiente, Piura carece de un sistema eficiente que permita a los

usuarios acceder a información útil y en tiempo real sobre las condiciones del tráfico en la ciudad.

#### **1.4 Objetivos**

- Objetivo general

Diseñar y validar un prototipo funcional de una aplicación móvil para el monitoreo del tráfico en la ciudad de Piura, utilizando tecnologías de geolocalización y datos colaborativos generados por los usuarios.

- Objetivos específicos

Analizar e identificar las principales causas de congestión vehicular en la ciudad de Piura a través del estudio de patrones de tráfico y reportes ciudadanos.

Diseñar e implementar la arquitectura y las funcionalidades esenciales de un prototipo móvil para el registro y visualización de incidencias viales.

Realizar pruebas de usabilidad del prototipo con usuarios reales, evaluando su funcionalidad, eficiencia y experiencia de uso.

#### **1.5 Alcance**

El proyecto contempla el desarrollo de un prototipo funcional de una aplicación móvil orientada a los ciudadanos de Piura, cuyo propósito es facilitar el monitoreo del estado del tráfico mediante reportes generados por los propios usuarios. La aplicación será desarrollada utilizando el framework Flutter y el lenguaje de programación Dart, además de integrar la API de Mapbox para la visualización de mapas.

##### **1.5.1 Funcionalidades**

- Diseño e implementación de una interfaz gráfica interactiva para dispositivo Android.
- Visualización dinámica de mapas mediante integración con la API de Mapbox.
- Sistema de registro de incidentes viales en tiempo real por parte de los usuarios.
- Almacenamiento estructurado de datos mediante una base de datos relacional.
- Generación de un archivo APK compatible con dispositivos Android.
- Generación y descarga de reportes de incidentes viales.
- Diseño visual de pantallas específicas para: visualización de mapa interactivo, registro de incidentes viales y consulta de reportes.

### 1.5.2 *Tareas*

- Elaborar diagramas de casos de uso para definir claramente interacciones usuario – aplicación.
- Diseñar esquema lógico y físico de la base de datos relacional.
- Codificar y desarrollar un prototipo de aplicación móvil.
- Realizar pruebas de usabilidad, mediante actividades específicas como: registrar incidentes viales y consultar incidentes reportados.

### 1.5.3 *Exclusiones*

- La implementación del sistema en un entorno de producción no está contemplada dentro del alcance de este proyecto debido a la limitación de tiempo según el cronograma establecido en el trabajo de tesis.



## Capítulo 2

### Marco teórico

El marco teórico conforma una parte esencial para este proyecto, porque brinda el contexto académico y científico para el desarrollo de la aplicación móvil. Este capítulo expone los fundamentos conceptuales y tecnológicos del monitoreo del tráfico, abarcando modelos de datos, técnicas de participación ciudadana y herramientas digitales utilizadas en el proyecto.

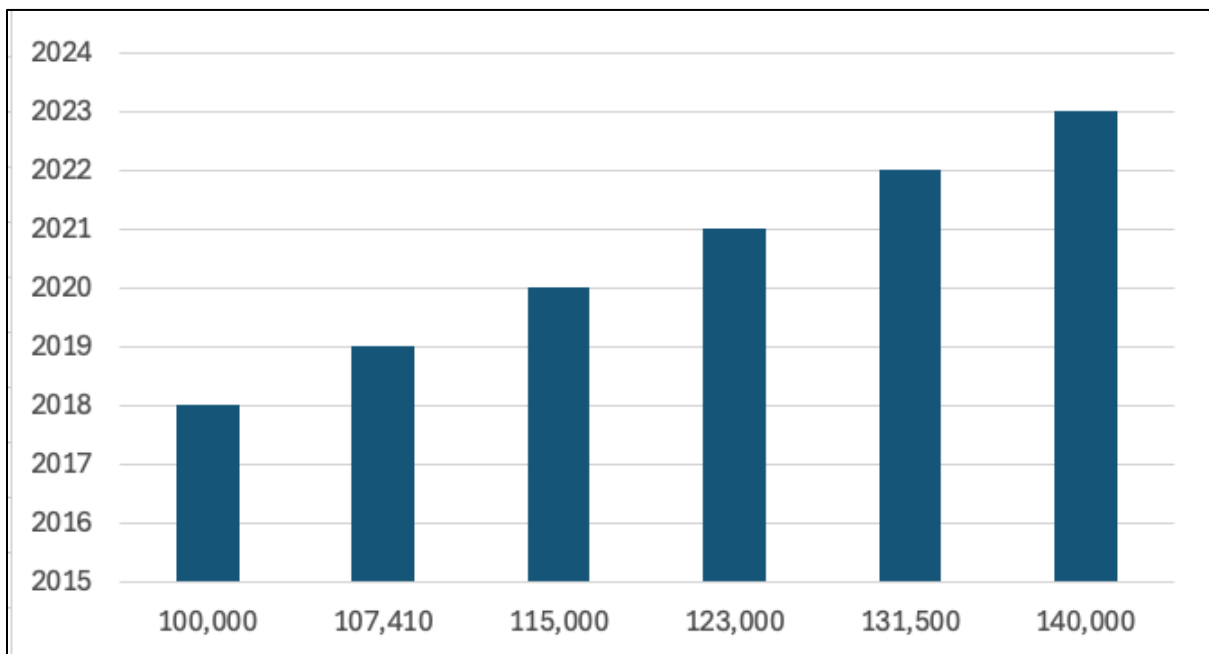
En esta sección se abordan conceptos sobre monitoreo de tráfico, tecnologías de ubicación GPS (Mapbox), framework de desarrollo (Flutter), lenguaje de programación (Dart) y Clean Architecture.

#### 2.1 Generalidades del monitoreo del tráfico vehicular

El monitoreo del tráfico vehicular es una necesidad creciente en ciudades con expansión urbana rápida. Piura enfrenta serios desafíos debido al incremento del parque automotor y una infraestructura vial insuficiente. Según el Instituto Nacional de Estadística e Informática (INEI, 2023), el número de vehículos aumentó de 94,237 en 2015 a 143,680 en 2022, lo que representa un crecimiento del 52.7%. Este aumento ha generado congestión significativa, sobre todo en zonas céntricas como el Cercado, Castilla, Centro de Piura y Veintiséis de Octubre.

#### Figura 1

*Crecimiento de Parque Automotriz Piura*



*Nota.* Tomado de Perú: INEI. (2022). Compendio estadístico del departamento de Piura 2022. Instituto Nacional de Estadística e Informática.

Además, Piura es una de las regiones más afectadas por el Fenómeno del Niño Costero. El Servicio Nacional de Meteorología e Hidrología (Senamhi, 2022), informó que las precipitaciones llegaron a medidas superiores de 250 mm<sup>3</sup> en 3 días, deteriorando más del 60% de las vías principales. Este daño genera un impacto en la congestión vehicular representando un riesgo más severo para los ciudadanos.

En este contexto, las aplicaciones móviles de monitoreo se presentan como herramientas clave para mejorar la movilidad urbana y la toma de decisiones.

Los sistemas inteligentes de transportes permiten una mejor interacción entre los usuarios y el entorno. Estos sistemas emplean tecnologías como GPS, redes móviles entre otras para recolectar y proporcionar información sobre el estado del tráfico, favoreciendo a los conductores en una toma de decisiones más acertada sobre el tránsito vehicular (Kaplan & Hegarty, 2017).

## **2.2 Modelo de datos**

Para el desarrollo de una aplicación móvil que permita el monitoreo de tráfico en tiempo real, se necesita una base teórica sobre modelos conceptuales y matemáticos que hagan posible la representación, procesamiento y prevención del comportamiento del tránsito vehicular.

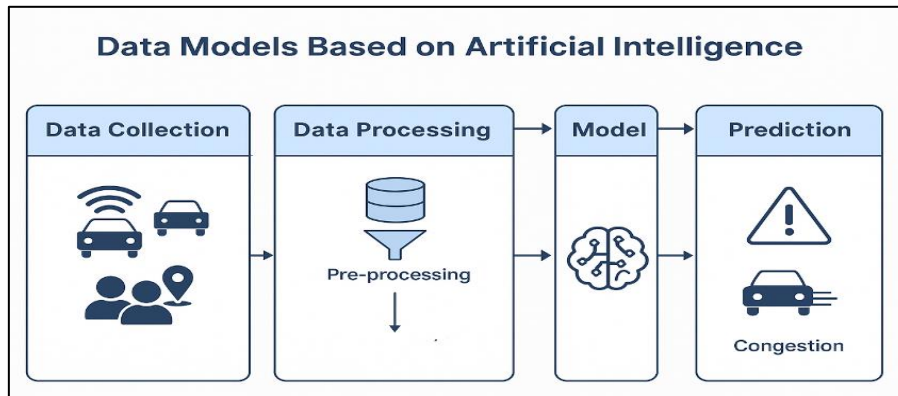
### **2.2.1 Modelo de datos basados en Inteligencia Artificial**

Estos modelos emplean técnicas avanzadas de análisis predictivo y aprendizaje automático para identificar patrones complejos en grandes cantidades de datos. Estos modelos permiten predecir situaciones de alta congestión vehicular gracias a algoritmos que se adaptan dinámicamente a cambios en las condiciones del entorno. Algunas de las técnicas más utilizadas son:

- **Redes Neuronales Recurrentes (RNN):** Su principal función es analizar secuencias de datos temporales de tráfico, permitiendo predecir flujos vehiculares en distintos momentos del día o anticipar congestiones vehiculares en función de eventos pasados.
- **Redes Neuronales Convolucionales (CNN):** Se utilizan para procesar información espacial, como imágenes satelitales, mapas, permitiendo identificar patrones visuales relevantes que indican situaciones críticas del tráfico.
- **Aprendizaje Profundo (Deep Learning):** Estos modelos combinan múltiples capas de redes neuronales para identificar relaciones no lineales altamente complejas entre diversos factores que influyen en el tráfico, como eventos climáticos, horarios pico y reportes de ciudadanos. (Wang et al., 2018).

**Figura 2**

*Diagrama de Modelo de Datos basado en Inteligencia Artificial*



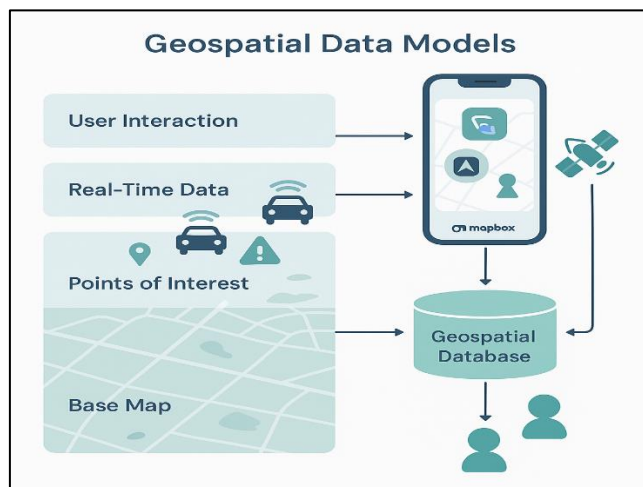
*Nota.* Elaboración propia basada en Goodfellow, Bengio y Coruville (2016).

### 2.2.2 Modelo de datos geoespaciales

Este tipo de modelos constituyen el marco estructural y técnico sobre el cual se organiza, almacena y representa información geográfica, tales como la ubicación de eventos de tráfico, rutas viales o zonas de congestión. Utilizan sistemas de referencia espacial, como coordenadas geográficas (latitud y longitud), para permitirles asociar los datos de una ubicación específica sobre un mapa digital. Este tipo de modelos emplean tecnologías que permiten integrar múltiples fuentes de información espacial, como imágenes satelitales, GPS, reportes generados por los usuarios. En la Figura 3, se muestra el diagrama de los componentes que interactúan en el modelo de datos Geoespaciales.

**Figura 3**

*Diagrama Modelo de datos Geoespaciales*



*Nota.* Elaboración propia basa en Lognley, Goodchild, Maguire y Rhind (2015).

Para el desarrollo de aplicaciones móviles de monitoreo, suelen utilizarse principalmente los modelos vectoriales y ráster.

**2.2.2.1 Modelos Vectoriales.** Este tipo de modelo representa objetos geográficos a través de entidades discretas como: puntos (para hacer referencia a pozos), líneas (representar caminos) y polígonos (para áreas de uso de suelo), lo que permite que su uso sea adecuado para representar fronteras definidas, realizar análisis topológicos y realizar medición de distancias. (Burrough & McDonnell, 1998).

Las ventajas de este modelo es que brinda precisión para tareas de ubicación y delimitación, además es ideal para análisis de redes y topología (Longley et al., 2015).

La aplicación implementa este modelo para la señalización de caminos, fijación de un punto en el mapa, esto permite que la ubicación geoespacial del usuario se gestione de manera más interactiva y visual, mejorando la interacción de los elementos geográficos con el usuario dentro de la plataforma.

**2.2.2.2 Modelos Ráster.** Este modelo divide el espacio en celdas regulares (píxeles) en las cuales, cada celda contiene un valor único (numérico, categórico, binario) que representa una característica del espacio geográfico, como elevación, uso del suelo, entre otros (Burrough, P. A., & McDonnell, R. A. (1998)).

Es adecuado para representar fenómenos continuos al dividir el espacio en una cuadrícula, donde cada celda almacena información específica sobre un atributo espacial. (Longley et al., 2015)

Este modelo es imprescindible cuando se trata de fenómenos continuos, como la temperatura o la elevación, ya que facilita la integración y visualización de datos agregados sobre incidentes.

A diferencia del modelo raster, que se enfocan en fenómenos continuos (por ejemplo, temperatura o elevación), el modelo vectorial es particularmente útil en contextos urbanos debido a que permite un análisis claro y visualmente intuitivo de la infraestructura vial.

Finalmente, en este caso se ha optado por un modelo vectorial, ya que resulta más adecuado para representar eventos discretos como incidentes de tráfico. Esta elección permite que los datos se representen de manera intuitiva en la distribución espacial de los eventos sobre un mapa digital, favoreciendo una mejor comprensión de comportamiento del tráfico urbano por parte del usuario.

### **2.2.3 Participación ciudadana y Crowdsourcing**

El concepto de crowdsourcing, o colaboración abierta, se refiere a un modelo participativo o de colaboración abierta, en el cual la información es generada directamente por una comunidad activa de usuarios. En el contexto del monitoreo vehicular, implica que los usuarios ingresan manualmente información en tiempo real sobre incidentes, congestión

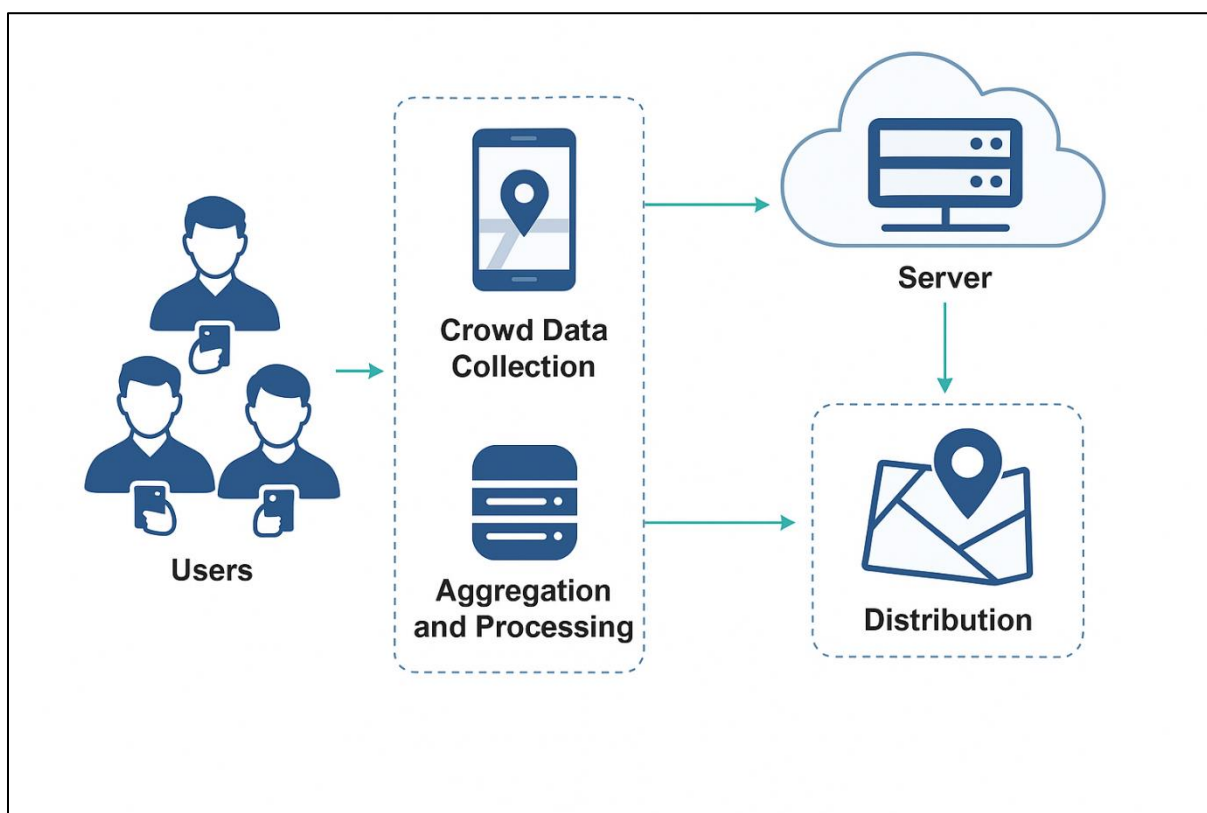
vehicular, condiciones viales o cualquier otro evento relevante, lo que genera un ciclo continuo de retroalimentación y actualización de datos entre los propios participantes (Silva & Martínez, 2020). Este modelo no solo facilita el acceso inmediato a información actualizada, sino que también fomenta la participación ciudadana activa en la gestión del tráfico urbano.

Una de las ventajas principales del crowdsourcing radica en su capacidad para proporcionar información precisa, transparente y oportuna, especialmente en situaciones donde las infraestructuras tecnológicas avanzadas, como sensores automáticos o cámaras de vigilancia, son limitadas o inexistentes. De esta forma, el crowdsourcing se convierte en una herramienta altamente eficiente y económica para la recopilación de datos geospaciales, adaptándose perfectamente a contextos urbanos con restricciones tecnológicas o presupuestarias (Goodchild, 2007).

Asimismo, diversos estudios demuestran que la participación ciudadana mediante crowdsourcing incrementa la percepción de transparencia y colaboración, fortaleciendo así el compromiso de los usuarios con la plataforma y mejorando significativamente la calidad y cantidad de los datos recopilados (Sui et al., 2013).

**Figura 4**

*Diagrama Crowdsourcing*



*Nota.* Elaboración propia basa en Howe (2008) y Estellés-Arolas & González Ladrón de Guevara (2012)

## **2.3 Modelo aplicado**

En el desarrollo de la presente investigación, se ha optado por el uso del modelo de datos geoespaciales específicamente un modelo vectorial. En este apartado, se retoma dicho modelo como base para el análisis y estructuración de la información recopilada. La elección de este modelo responde a su capacidad para utilizar sistemas de referencia espacial (latitud, longitud), para delimitar la información de ubicación geográfica en el mapa digital, lo cual resulta fundamental para los objetivos planteados.

La elección de este modelo se fundamenta principalmente en sus ventajas para la representación precisa y eficiente de la información espacial urbana, donde es fundamental identificar claramente los puntos específicos de interés (como accidentes o bloqueos viales) y rutas definidas (calles y avenidas).

En el marco del presente proyecto, el crowdsourcing resulta particularmente valioso, ya que los ciudadanos de Piura se convierten en actores fundamentales del proceso de generación y actualización constante del conocimiento geoespacial. A través de tecnologías accesibles y ligeras como Mapbox, se logra reducir la complejidad técnica y financiera, permitiendo implementar una solución tecnológica eficiente sin sacrificar funcionalidad ni precisión. Este enfoque no solo empodera a los usuarios, sino que también fortalece la confiabilidad y relevancia de la información, al ser proporcionada directamente por quienes experimentan las condiciones reales del tránsito urbano diariamente.

Finalmente, este modelo elegido no sólo permite resolver eficazmente las necesidades actuales identificadas en la ciudad de Piura, sino que también presenta ventajas significativas para que la aplicación sea adaptable y escalable para futuras mejoras tecnológicas. Además, se destacó la necesidad de fomentar una cultura de movilidad sostenible, incorporando elementos que promueven activamente la colaboración ciudadana y el uso más eficiente de las vías disponibles, enriqueciendo así el valor social y ambiental del proyecto.

## **2.4 Tecnologías utilizadas**

La tecnología orientada a la movilidad urbana ha evolucionado la manera en que las personas se trasladan dentro de las ciudades. La posibilidad de aplicaciones que se tiene es infinita y permite mejorar la toma de decisiones en cuanto a transporte se refiere, como por ejemplo evitar zonas de congestión, rutas a seguir. Estas aplicaciones integran tecnologías como sensores GPS, tecnologías de comunicación y procesamiento de datos en tiempo real para ofrecer rutas óptimas (Li et al., 2021).

### **2.4.1 Sistema de posicionamiento global (GPS)**

La implementación y uso de tecnologías para el monitoreo del tráfico necesita de dispositivos de posicionamiento satelital (GPS), sensores de tráfico, cámaras y sistemas de información geográfica (SIG). Estos dispositivos permiten recopilar datos en tiempo real sobre la velocidad, dirección y ubicación de los vehículos (Kaplan & Hegarty, 2006).

### **2.4.2 Comunicación Inalámbrica**

Las redes móviles (3G, 4G, 5G) y Wi-Fi son catalogadas como tecnologías de conexión inalámbrica y para el desarrollo de este proyecto son utilizadas para transmitir datos del usuario hacia el servidor, y viceversa. Permiten el funcionamiento en tiempo real de las aplicaciones de monitoreo de tráfico, especialmente en zonas con cobertura (Deng et al., 2020).

Esta tecnología es empleada para la implementación de sistemas de monitoreo de tráfico que utilizan sensores IoT (Internet of Things) y cámaras, facilitando la transmisión de datos a través de esta tecnología hacia la nube para ser puestos a disposición de los usuarios en un aplicativo móvil. Wang et al. (2019). Estas aplicaciones prácticas evidencian como las tecnologías de comunicación son primordiales para el funcionamiento eficiente de los sistemas de monitoreo de tráfico.

### **2.4.3 Integración de APIs y servicios web**

Las API's permiten complementar las funcionalidades de las aplicaciones, accediendo a información de diferentes fuentes. Por ejemplo, Mapbox, es una plataforma que se utiliza principalmente para el desarrollo de mapas interactivos personalizados. Permite visualizar la ubicación del usuario a través del uso del GPS de su dispositivo ubicándolo dentro de un punto de coordenadas preciso y en tiempo real, esto lo hace una herramienta útil para las diferentes necesidades de movilidad urbana. (Mapbox, 2023).

## **2.5 Enfoques y modelos tecnológicos**

El desarrollo de soluciones tecnológicas para el monitoreo y gestión del tráfico vehicular en entornos urbanos requiere la adopción de diferentes enfoques y modelos tecnológicos que permiten integrar, procesar y analizar grandes volúmenes de información en tiempo real. En esta sección se describen los principales paradigmas tecnológicos que respaldan la arquitectura y el funcionamiento del sistema propuesto.

### **2.5.1 Internet de las cosas (IoT)**

Paradigma tecnológico que establece una relación entre la interconexión de objetos físicos (sensores, dispositivos móviles, vehículos) con las redes de comunicación, permitiendo que intercambien, recolecten y procesen datos de manera autónoma y en tiempo real (Atzori et al., 2010).

En el contexto de este proyecto, el internet de las cosas (IoT) juega un rol fundamental para comprender cómo diversas fuentes de datos pueden integrarse mediante tecnologías de comunicación. Aunque este proyecto no implementará sensores físicos como cámaras o semáforos, se reconoce que los dispositivos móviles, a través de su geolocalización y conectividad, pueden actuar como fuentes de recolección de datos en tiempo real. Esta

capacidad permite recopilar información útil sobre el entorno vehicular, como la ubicación del usuario.

Alguna de las principales ventajas de IoT para el contexto de este proyecto:

Mejora la recolección de información de manera automatizada y continua.

Detección temprana de incidentes y congestión

Posibilidad de implementar sistemas inteligentes de gestión de tráfico.

### 2.5.2 *Arquitectura Cliente – Servidor*

Los sistemas informáticos pueden presentar diversas arquitecturas. La elección de la más adecuada dependerá de los requisitos, necesidades y objetivos específicos del mismo.

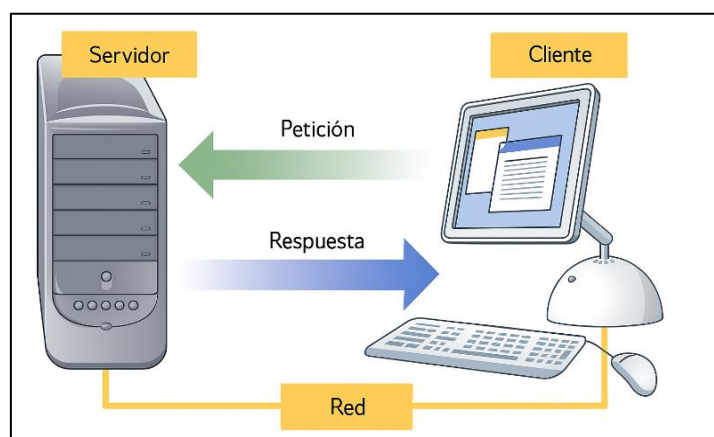
La arquitectura se define como la estructura y diseño del sistema, gracias a ella se proporciona una visión global de cómo interactúan los componentes entre sí, es fundamental para definir el rendimiento, escalabilidad y la capacidad de adaptación a los cambios.

En la arquitectura cliente/servidor, se cuenta con dos componentes principales: el cliente y el servidor. Ambos trabajan en conjunto de manera coordinada para brindar servicios. El cliente es la aplicación o sistema informático que genera peticiones de información hacia el servidor aguardando por una respuesta, el servidor es el encargado de brindar respuesta a las peticiones del cliente a través de un navegador web.

Aunque se implementa Clean Architecture como enfoque organizacional del software, la interacción entre la app y los servicios en la nube sigue el patrón cliente – servidor, permitiendo una comunicación eficiente y escalable.

**Figura 5**

*Diagrama Arquitectura Cliente - Servidor*



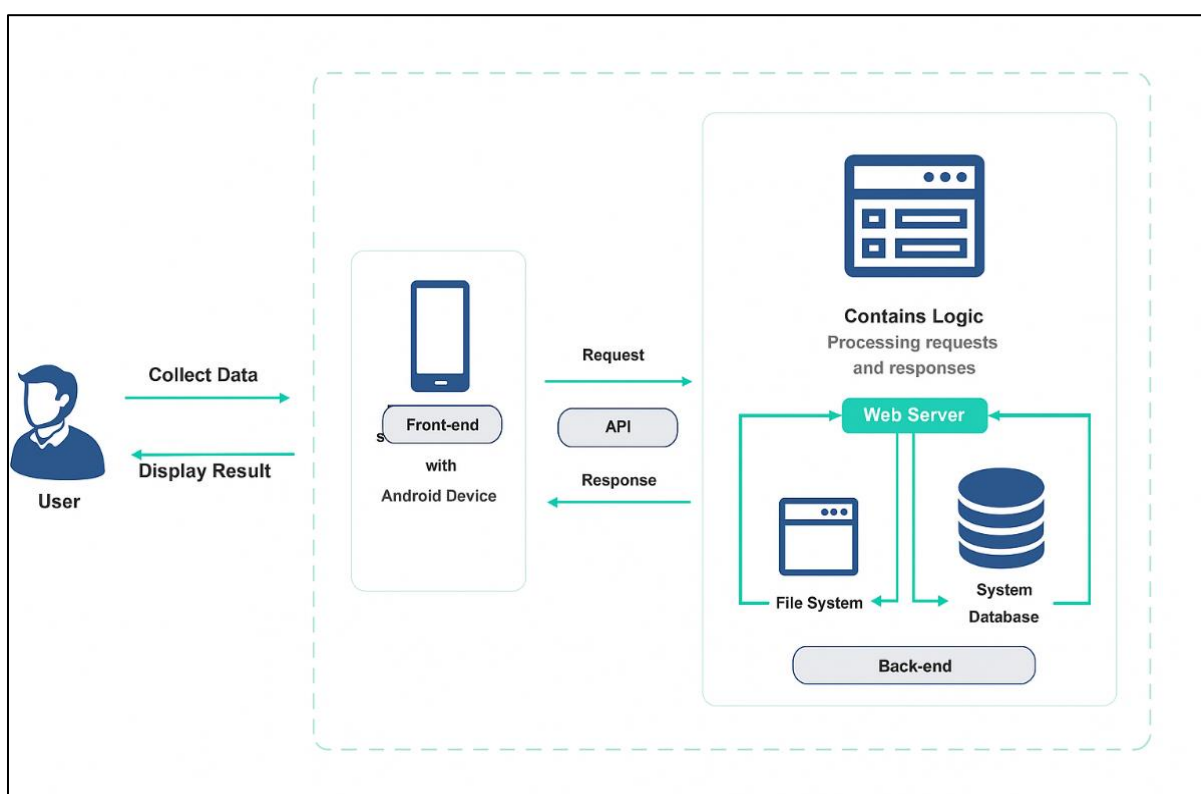
*Nota.* Reproducido de “Arquitectura cliente-servidor”, por Concepto.de, s.f.

### 2.5.3 Arquitectura de Aplicaciones móviles

Sistema informático diseñado para su ejecución en dispositivos móviles, como teléfonos inteligentes o tabletas. Este tipo de aplicaciones permite a los usuarios acceder a diversas funcionalidades, como la de visualización y gestión de información en tiempo real. Para el presente estudio, la aplicación móvil es la plataforma principal para recibir y mostrar datos sobre el estado del tráfico vehicular (Cano et al., 2021).

**Figura 6**

*Arquitectura general de desarrollo de aplicaciones móviles*



*Nota.* Elaboración propia basada en Sommerville (2016) y Martin (2017).

### 2.5.4 Arquitectura Clean Architecture

Clean Architecture es una arquitectura de software que busca separar las responsabilidades del sistema en diferentes capas con el fin de mejorar la mantenibilidad, escalabilidad y facilitar las pruebas del código. Este enfoque promueve la independencia de frameworks, base de datos y dispositivos externos, permitiendo que el núcleo del sistema (lógica de negocio) permanezca aislado de detalles de implementación.

Para este proyecto, la arquitectura se implementa organizando el código en tres capas principales:

Capa de dominio: contiene las entidades, casos de uso y reglas de negocio.

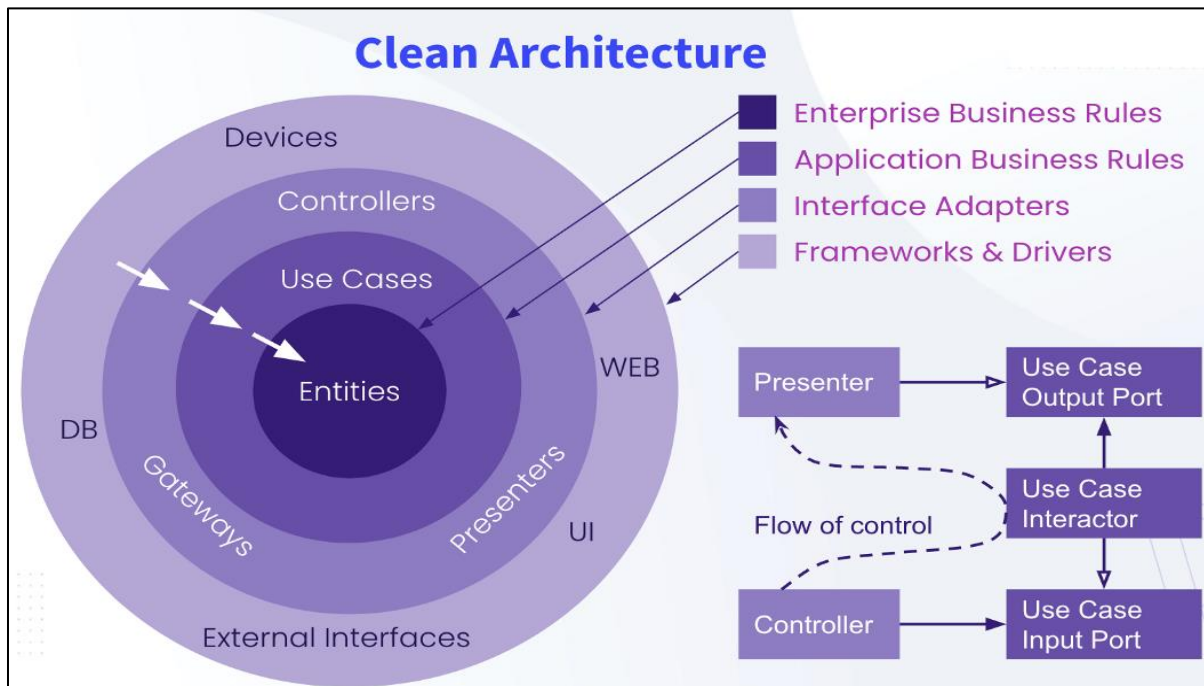
Capa de datos: maneja la comunicación con servicios externos como Firebase.

Capa de presentación: gestiona la interfaz de usuario desarrollada en Flutter.

Esta estructura modular facilita el desarrollo y las pruebas permitiendo también escalar o modificar componentes sin afectar el núcleo de la aplicación.

**Figura 7**

*Representación de arquitectura lógica Clean Architecture*



*Nota.* Adaptado de "Clean Architecture Diagram" por AltexSoft, s.f.

### 2.5.5 Análisis en tiempo real (Real Time Data Analytics)

Capacidad de procesar y examinar grandes volúmenes de datos después de ser generados, permitiendo obtener información útil para la toma de decisiones en cuestión de segundos o milisegundos (Gama et al., 2019).

En el contexto de este proyecto, el análisis en tiempo real conlleva el procesamiento instantáneo de datos proveniente de sensores, aplicaciones móviles, cámaras y otros dispositivos. Esto permite generar alertas y ofrecer recomendaciones a los usuarios para una mejor toma de decisión. Para lograrlo, se emplean plataformas y tecnologías especializadas en el procesamiento de flujos de datos, conocidas como stream processing. Entre estas, destacan soluciones como Apache Kafka, una plataforma distribuida de mensajería y transmisión de eventos que permite gestionar grandes volúmenes de datos en movimiento de manera eficiente y confiable (Kreps et al., 2011). Asimismo, servicios como Firebase de Google ofrecen herramientas de mensajería en tiempo real, bases de datos en la nube y notificaciones push, facilitando la comunicación inmediata entre el sistema y el usuario final.

Adicionalmente, el análisis en tiempo real también se integra con sistemas de inteligencia artificial y aprendizaje automático, permitiendo que el sistema reconozca patrones, anticipe eventos de congestión y optimice dinámicamente la gestión del tráfico urbano (Sakr & Gaber, 2014).

### **2.5.6 Servidor web y servicios en la nube**

En el contexto del desarrollo de aplicaciones móviles, como la propuesta en este proyecto, el servidor web cumple un rol esencial en la gestión de datos y la interacción entre el cliente (la aplicación móvil) y los servicios en la nube. Su principal función es recibir peticiones del cliente y devolver respuestas estructuradas, generalmente en formatos como JSON o XML, sin realizar interpretación directa del contenido visual.

En este proyecto, el servidor web está representado por servicios en la nube como Firebase, que permiten almacenar, autenticar y sincronizar datos en tiempo real entre múltiples usuarios. Asimismo, la integración con Mapbox requiere el consumo de servicios web para cargar mapas interactivos y consultar ubicaciones geoespaciales a través de peticiones HTTP/HTTPS.

Los principales componentes asociados al servidor web en este contexto son:

- **Infraestructura en la nube:** El backend de la aplicación no se aloja en un servidor físico tradicional, sino que utiliza plataformas como Firebase, que proveen servicios escalables en servidores distribuidos, garantizando alta disponibilidad y rendimiento.
- **Sistema operativo del servidor:** Aunque el desarrollador no gestiona directamente el sistema operativo en servicios como Firebase, estos corren sobre sistemas como Linux o distribuciones especializadas en servidores cloud.
- **Software de servidor web:** Las plataformas como Firebase utilizan tecnologías como Node.js y servidores web como Nginx o equivalentes, que permiten manejar las peticiones de los clientes móviles y distribuir la información de forma eficiente. A su vez, Mapbox opera sobre APIs REST que permiten consumir servicios de mapas en tiempo real a través del protocolo HTTPS.

En este enfoque basado en servicios web facilita el desarrollo de aplicaciones móviles modernas, al delegar el procesamiento y almacenamiento a la nube, reduciendo la carga sobre el dispositivo del usuario y asegurando sincronización.

## **2.6 Seguridad y privacidad**

Las aplicaciones de desarrollo móvil manejan información sensible como la geolocalización de los usuarios, es por ello, que se requiere la implementación de medidas de seguridad que garanticen la privacidad de la información. Es posible incorporar mecanismos que no sean de muy alta complejidad y eficientes que permitan brindar una protección a los datos sin afectar el performance de la aplicación ni la experiencia de usuario.

### **2.6.1 Protección de datos del usuario**

En este proyecto la principal prioridad es mantener la protección de los datos de la ubicación del usuario. La gestión responsable de información personal, como ubicaciones, datos GTS, cuentas bancarias o documentos de identidad es fundamental para este tipo de aplicaciones. Por ello, resulta imprescindible aplicar buenas prácticas como el cifrado de datos y el consentimiento informado, con el fin de salvaguardar la privacidad del usuario (González et al., 2018). Para lograrlo, pueden implementarse diversas técnicas y normativas orientadas a reforzar la seguridad y la confianza en el uso de la aplicación, como las que se mencionan a continuación:

**2.6.1.1 Cifrado local de datos sensibles.** Como medida de seguridad por parte de Flutter, se pueden utilizar paquetes como “encrypt” o “flutter\_secure\_storage”, que permiten encriptar la información en caso el dispositivo se vea comprometido (Flutter Secure Storage, 2023).

Un caso de éxito se observa en la aplicación móvil PassKeep, el cual es gestor de contraseñas que su principal función es almacenar datos sensibles de los usuarios de manera cifrada y segura en los dispositivos móviles (PassKeep, 2023).

Otro caso de éxito es la aplicación Extrack, una aplicación que tiene como propósito el seguimiento de gastos personales, a través del uso del paquete “flutter\_secure\_storage” busca proteger la información financiera de los usuarios, cumpliendo con los requisitos de privacidad de los datos y aumentando la confianza de los usuarios en la plataforma. (Extrack, 2023).

**2.6.1.2 Control de permisos.** Paquetes como “permission\_handler” permiten la gestión de forma explícita de cómo y cuando se accede a la ubicación, mostrándole al usuario diálogos claros que expliquen el uso de su información (Google, 2023).

**2.6.1.3 Anonimización de datos.** Para evitar guardar información sensible o identificable del usuario, se puede almacenar información anonimizada, como por ejemplo registrar un evento de congestión sin asociarlo directamente con un usuario específico (OWASP, 2021).

**2.6.1.4 Términos y condiciones claros.** Presentar al usuario un aviso de privacidad claro donde se explique qué datos se recopilan, para qué se usan y cómo se protegen. Esto refuerza el principio de consentimiento informado (González et al., 2018).

### **2.6.2 Seguridad de la comunicación**

Uno de los mejores protocolos de seguridad que se pueden implementar hoy en día para evitar las vulnerabilidades en la transmisión de la información, es el protocolo HTTPS, junto con mecanismos de autenticación, garantizan integridad y confidencialidad de la información (Dierks & Rescorla, 2008).

La importancia de HTTPS radica principalmente en tres propiedades fundamentales que garantiza:

- **Confidencialidad:** La información transmitida entre cliente y servidor se cifra, haciendo que los datos sean ilegibles para cualquier agente externo que intercepte la comunicación.
- **Integridad:** HTTPS asegura que la información transmitida no ha sido modificada o corrompida durante su tránsito.
- **Autenticación:** A través de certificados digitales, se verifica la identidad del servidor para prevenir ataques como phishing o ataques de intermediarios.

HTTPS opera principalmente utilizando certificados digitales emitidos por Autoridades Certificadoras (CA, Certificate Authorities). Estos certificados validan la identidad de un sitio web y habilitan un canal cifrado seguro mediante un intercambio inicial de claves (Handshaking), donde cliente y servidor acuerdan utilizar una clave criptográfica para cifrar la información intercambiada durante la sesión (Stallings, 2017).

## **2.7 Marco conceptual**

El presente marco conceptual tiene como finalidad establecer y precisar términos y conceptos fundamentales que sustentan el desarrollo de este estudio. En este apartado se presentan las definiciones y características de los principales elementos involucrados en el monitoreo vehicular y desarrollo de aplicaciones móviles.

- **Monitoreo Vehicular:** Proceso de seguimiento y análisis continuo de la ubicación, velocidad y trayectoria de los vehículos en circulación. Mediante la implementación de tecnologías como GPS y las tecnologías de comunicación (redes móviles) permiten la recopilación y transmisión de datos en tiempo real para su visualización y gestión (Alhassan et al., 2020).
- **Tráfico vehicular:** hace referencia al flujo continuo de vehículos que transitan por las vías urbanas. En el contexto de este proyecto, se interpreta como el comportamiento dinámico del parque automotor en la ciudad de Piura (Castillo & Huamán, 2020).
- **Geolocalización:** Tecnología que permite identificar la ubicación geográfica precisa de un dispositivo móvil a través de tecnologías de geolocalización (GPS, sistemas de posicionamiento). Es primordial para rastrear la posición de los usuarios y vehículos en tiempo real, mejoran la fiabilidad de los datos ofrecidos por la aplicación (Kaplan & Hegarty, 2017).
- **Congestión Vehicular:** término que define la acumulación de vehículos en una vía, lo que provoca una disminución significativa de la velocidad promedio del tránsito. En el contexto de esta aplicación, se considera que existe congestión cuando la velocidad del tráfico cae por debajo de un umbral específico, por ejemplo, menos de 15 km/h, lo que impacta negativamente en la movilidad urbana (Zhu et al., 2022)

- Notificación en tiempo real: mensaje emergente o alerta que la aplicación envía automáticamente al usuario cuando detecta situaciones de congestión vehicular, teniendo como objetivo redirigir la ruta del usuario.
- Sistemas de información: Se define sistemas de información como el conjunto de distintos componentes interrelacionados entre sí, que buscan recopilar, procesar, almacenar y gestionar información para un entorno en específico, con la finalidad de aportar en la operatividad, dirección y análisis según la estrategia definida.

En general, un sistema necesita como materia prima (input) los datos. Además, existen tres procedimientos fundamentales: almacenamiento, procesamiento y transformación. Finalizado el procesamiento se obtiene como resultado final, información que según el análisis y criterios correspondientes serán de utilidad para los distintos usuarios.

La información establecida en el marco teórico respecto a diversos enfoques y modelos documentados aportaron información valiosa para la definición de requisitos y la propuesta del sistema desarrollado.



## Capítulo 3

### Marco metodológico

Para el desarrollo del prototipo, se optó por un enfoque flexible y adaptativo, centrado en la evolución continua del producto a medida que surgían nuevas necesidades. A lo largo del proceso, se integraron tecnologías clave como Dart, Flutter y la API Mapbox, cuya implementación permitió construir funcionalidades en tiempo real.

Este capítulo orienta el proyecto hacia la creación de un prototipo funcional, con el objetivo de demostrar la viabilidad del monitoreo colaborativo del tráfico como una herramienta para mejorar la movilidad urbana. Se brinda una explicación detallada de cómo se abordó el proceso de investigación y desarrollo, permitiendo comprender el procedimiento seguido para lograr los objetivos planteados.

#### 3.1 Planteamiento del problema

La congestión vehicular en Piura constituye una problemática cotidiana que afecta la calidad de vida de sus ciudadanos, generando retrasos, incremento en los tiempos de desplazamiento, mayores costos económicos y un aumento de la contaminación ambiental por la emisión de gases contaminantes. Este problema se agrava por la falta de información actualizada y precisa acerca del estado del tráfico en tiempo real, lo que impide a los conductores tomar decisiones informadas sobre rutas óptimas para evitar zonas congestionadas o bloqueadas.

A diferencia de ciudades más grandes que disponen de infraestructura tecnológica avanzada y sistemas integrados para monitoreo y control vehicular, Piura aún carece de soluciones tecnológicas efectivas que permitan un seguimiento constante del flujo vehicular. Esta carencia genera un uso ineficiente de las vías existentes, especialmente en horas pico o en eventos específicos.

Por tanto, es fundamental implementar herramientas tecnológicas adaptadas a la realidad local, que proporcionen información precisa y en tiempo real, facilitando la toma de decisiones informadas, la reducción de tiempos de viaje y una mejora en la calidad de vida urbana.

#### 3.2 Tipo y diseño de investigación

La investigación es de tipo aplicada, debido a que el objetivo principal es desarrollar una aplicación móvil con tecnología específica, con la finalidad de resolver un problema existente y real, la ausencia de información oportuna sobre la congestión vehicular en la ciudad de Piura. El enfoque cualitativo es el predominante, ya que, la validación está respaldada por la retroalimentación de los usuarios mediante pruebas prácticas.

Se adopta un enfoque descriptivo porque previamente se realiza un análisis para identificar las causas principales de la congestión vehicular en la ciudad (Yin, 2017).

El desarrollo del software se basa en una metodología ágil, que permite adaptar rápidamente el producto a los cambios requeridos durante su construcción, facilitando así la entrega continua de versiones funcionales del prototipo. Esta metodología ha permitido una iteración rápida del prototipo, según los nuevos requerimientos y pruebas de usabilidad (Beck & Andres, 2004; Cohn, 2009; Pressman & Maxim, 2015).

### **3.3 Población, muestra y marco muestral**

La población objetivo está compuesta por conductores de vehículos particulares en la ciudad de Piura, quienes enfrentan a diario los problemas derivados de la congestión.

Para la evaluación de la usabilidad de la aplicación se contó con la participación de 10 usuarios. Esta cantidad fue determinada considerando fundamentos metodológicos recientes sobre pruebas de usabilidad en aplicaciones móviles. Según Sauro y Lewis (2016), así como Tullis y Albert (2013), el tamaño de muestra debe adaptarse al contexto y complejidad del sistema, priorizando la diversidad del público objetivo sobre cifras fijas. Por ello se consideró adecuada una muestra de 10 participantes con la finalidad de obtener resultados representativos.

Los participantes fueron personas adultas, residentes en la ciudad de Piura, con edades comprendidas entre 20 y 45 años, con un nivel básico – intermedio de familiaridad con el uso de dispositivos móviles. Este perfil fue seleccionado en función de su afinidad con los usuarios finales potenciales de la aplicación, es decir, ciudadanos que utilizan smartphones como herramienta cotidiana para desplazarse o informarse sobre el tráfico urbano.

### **3.4 Técnicas e instrumentos de recolección de datos**

Para el levantamiento de información de la situación actual con la finalidad de identificar problemas y oportunidades de mejora se utilizaron las siguientes técnicas e instrumentos:

**Análisis documental:** Se revisaron artículos científicos, informes técnicos y estudios previos para contextualizar el problema.

**Observación:** Se analizó el comportamiento de los usuarios frente al tráfico urbano, de manera activa (interactuando con ellos) y de manera pasiva, es decir, solo observando sus acciones sin intervención alguna.

**Entrevistas:** Se realizaron entrevistas no estructuradas para identificar necesidades, percepciones y requerimientos funcionales y no funcionales de la aplicación.

### **3.5 Herramientas informáticas**

Para el desarrollo del prototipo móvil se utilizaron las siguientes herramientas:

- Lenguaje y framework: Se utilizó Dart con el framework de desarrollo Flutter, lo que permitió desarrollar una aplicación multiplataforma (Android e iOS) (Google Developers, 2024).
- API de mapa: Se integró Mapbox para visualizar y reportar incidentes en tiempo real mediante geolocalización, garantizando una interfaz de usuario intuitiva y funcional (Mapbox, 2024).
- Arquitectura: Se empleó el patrón Clean Architecture, que favorece la separación entre lógica de negocio, interfaz y acceso a datos, garantizando modularidad, escalabilidad y eficiencia en la implementación (Martin, 2017; Fowler, 2019; Gamma et al., 1995).
- Instrumentos de validación: Se usaron formularios de Google para evaluar la experiencia de usuario y la interfaz (UX, Ui), facilitando la recopilación y análisis de datos (Nielsen Normal Group, 2023; Sommerville, 2015).

### 3.6 Procedimientos

El procedimiento desarrollado en esta investigación comprende las siguientes fases:

- Análisis del problema: Esta fase implicó la recopilación exhaustiva de información proveniente de diversas fuentes, como estudios previos, entrevistas, reportes de tráfico, con el fin de identificar claramente las principales causas que generan congestión vehicular en la ciudad.
- Análisis del sistema: En esta etapa se definieron los requerimientos y funcionalidades necesarios para la aplicación móvil.
- Diseño del sistema: Durante esta fase se definieron las interacciones usuario – aplicación mediante diagramas de casos de uso. Asimismo, se diseñaron los esquemas lógico y físico de la base de datos relacional.
- Implementación: Se realizó el desarrollo del prototipo con Flutter y Mapbox.
- Pruebas: Finalmente, se ejecutaron pruebas de usabilidad mediante tareas específicas y retroalimentación de usuarios tales como, el registro de incidentes viales dentro de la aplicación y la consulta de incidentes previamente reportados, permitiendo así evaluar la eficiencia, efectividad y satisfacción del usuario con la herramienta desarrollada.

### 3.7 Método

Se utilizó la metodología ágil SCRUM, con iteraciones breves y entregables funcionales al final de cada sprint (Schwaber & Sutherland, 2020). SCRUM es una metodología ampliamente utilizada en la industria del desarrollo de software por su capacidad para gestionar proyectos complejos de manera eficiente y adaptativa. Se caracteriza principalmente por su enfoque iterativo e incremental, permitiendo entregar partes

funcionales del producto en ciclos cortos denominados sprints, que generalmente tienen una duración entre una y cuatro semanas

Las etapas clave de SCRUM que se implementaron en el desarrollo del prototipo incluyeron:

- Planificación del sprint: En esta fase inicial, se seleccionaron las funcionalidades y objetivos por iteración.
- Ejecución del sprint: Durante esta fase, se desarrollaron las tareas asignadas, manteniendo un constante monitoreo de avances.
- Revisión del sprint: En esta etapa se evaluaron resultados, se recopiló la retroalimentación y se hicieron los ajustes necesarios.

### **3.8 Evaluación de usabilidad**

En esta sección se describe el proceso que se llevará a cabo para validar la usabilidad del prototipo. La evaluación se realizará con 10 participantes que representan al usuario final del sistema, quienes interactúan con la aplicación.

Se realizarán sesiones individuales en las que cada usuario probará funciones clave de la aplicación (registro de incidentes, navegación en el mapa, verificación de datos), mientras el investigador toma nota de observaciones relevantes. Al finalizar, los usuarios completarán un cuestionario de percepción.

Las reuniones serán de tipo presencial o remota (según disponibilidad) y tendrán una duración aproximada de 15 a 20 minutos por participante. Se utilizará Google Forms para la recolección de respuestas.

### **3.9 Método de evaluación de usabilidad**

El modelo adoptado para la evaluación se basa en la norma internacional ISO 9241-11, la cual establece que la usabilidad está compuesta por tres atributos fundamentales: eficacia, eficiencia y satisfacción, los cuales deben ser considerados en relación al contexto de uso.

**Tabla 1***Criterios y definiciones según la norma ISO 9241-11*

Criterio	Definición según ISO 9241-11
Eficacia	Precisión y grado de completitud con los que los usuarios logran los objetivos propuestos.
Eficiencia	Recursos utilizados en relación al nivel de efectividad alcanzado (por ejemplo, tiempo o esfuerzo).
Satisfacción	Grado de bienestar del usuario al utilizar el producto, así como la aceptación del sistema.

*Nota.* Definiciones según la norma ISO 9241-11

Para el instrumento de evaluación, se diseñó un cuestionario compuesto por 8 preguntas cerradas tipo Likert (de 1 a 5), cada pregunta está asociada a uno de los criterios de usabilidad definidos anteriormente.

**Tabla 2***Cuestionario de preguntas para medir usabilidad*

Pregunta	Criterio
¿Pudiste completar las tareas propuestas sin dificultades?	Eficacia
¿El sistema respondió de forma correcta a tus acciones?	Eficacia
¿La información presentada te permitió entender el estado del tráfico en tu zona?	Eficacia
¿Las instrucciones y etiquetas dentro de la app fueron comprensible?	Eficacia
¿Encontraste fácilmente las opciones que necesitas usar?	Eficiencia
¿Consideras que la aplicación tiene una buena velocidad de respuesta?	Eficiencia
¿Te pareció clara la información mostrada en el mapa de monitoreo?	Eficiencia
¿Te sentiste cómodo utilizando la interfaz de la aplicación?	Satisfacción
¿Consideras que la aplicación es útil para reportar incidentes de tráfico?	Satisfacción

Pregunta	Criterio
¿Recomendarías esta aplicación a otras personas?	Satisfacción

*Nota.* Cuestionario de preguntas junto con el criterio designado.

**Tabla 3**

*Descripción por escala*

Escala	Descripción
1	Muy en desacuerdo
2	En desacuerdo
3	Neutral
4	De acuerdo
5	Muy de acuerdo

*Nota.* Descripción de la escala aplicada en cuestionario

La categorización del nivel de usabilidad se calcula de acuerdo a un promedio por criterio y un promedio general. Según los resultados, se definirá el valor de usabilidad del prototipo:

**Tabla 4**

*Tabla de promedio general*

Promedio general	Nivel de usabilidad
4.5 – 5.5	Alto
3.0 – 4.4	Medio
Menor a 3.0	Bajo

*Nota.* Descripción de promedio general.

Este análisis permitirá validar si la aplicación cumple con los estándares básicos de usabilidad y facilitará la toma de decisiones respecto a su mejora o implementación final.

## Capítulo 4

### Análisis del sistema

El presente capítulo presenta el análisis realizado como base para el desarrollo del sistema de monitoreo vehicular en la ciudad de Piura. En primer lugar, se identifican las principales causas de la congestión vehicular, lo cual permite contextualizar la necesidad de una solución tecnológica. Posteriormente, se describe el diseño general del sistema, los procesos involucrados, los requisitos funcionales y no funcionales, así como los casos de uso que guían la interacción entre los usuarios y la aplicación.

El análisis del sistema se desarrolló considerando la realidad urbana de Piura, el incremento del parque automotor, los efectos del Fenómeno del Niño sobre la infraestructura vial, y la carencia de información en tiempo real para la toma de decisiones sobre rutas óptimas. Todo ello motivó la creación de una aplicación móvil colaborativa, capaz de recopilar, procesar y mostrar información vial en tiempo real mediante la participación ciudadana.

Este análisis constituye el punto de partida para la implementación técnica del prototipo, asegurando que cada componente del sistema responda a necesidades específicas y permita una experiencia de usuario eficaz, eficiente y satisfactoria.

#### 4.1 Análisis de causas de la congestión vehicular

Con el propósito de identificar claramente las causas que generan la congestión vehicular en Piura, se llevó a cabo una revisión documental que incluyó reportes emitidos por entidades oficiales como INEI, el Servicio Nacional de Meteorología e Hidrología (Senamhi), estudios previos sobre movilidad urbana y análisis de informes de la Municipalidad Provincial de Piura.

##### 4.1.1 Recopilación de información

La recopilación de información para el presente proyecto se basó fundamentalmente en la investigación y análisis de artículos científicos, publicaciones académicas y documentos técnicos relacionados con el monitoreo vehicular, las aplicaciones móviles y las tecnologías asociadas. Se revisaron fuentes recientes, bases de datos especializados y literatura internacional para identificar las tendencias, retos y soluciones existentes en el contexto de la gestión del tráfico en entornos urbanos.

Del análisis realizado, se identificaron las siguientes causas principales:

**Crecimiento acelerado del parque automotor:** Según INEI (2023), el número de vehículos aumentó un 52.7%, entre los años 2015 y 2022, este incremento no estuvo respaldado por un desarrollo proporcional en infraestructura vial, generando saturación evidente en las principales vías de la ciudad.

**Infraestructura vial insuficiente y deteriorada:** Las vías de circulación en la ciudad, especialmente en zonas céntricas como Cercado, Castilla y Veintiséis de Octubre, presentan

problemas críticos como calles estrechas, falta de mantenimiento y señalización deficiente, generando cuellos de botella y demoras significativas durante las horas pico.

Eventos climáticos extremos: La ciudad de Piura es altamente susceptible al Fenómeno del Niño Costero, cuyas intensas precipitaciones (superiores a los 250 [mm] <sup>^3</sup> en períodos cortos según Senamhi, 2022) deterioran gravemente la infraestructura vial urbana, provocando cierres parciales o totales y disminuyendo considerablemente la capacidad de circulación.

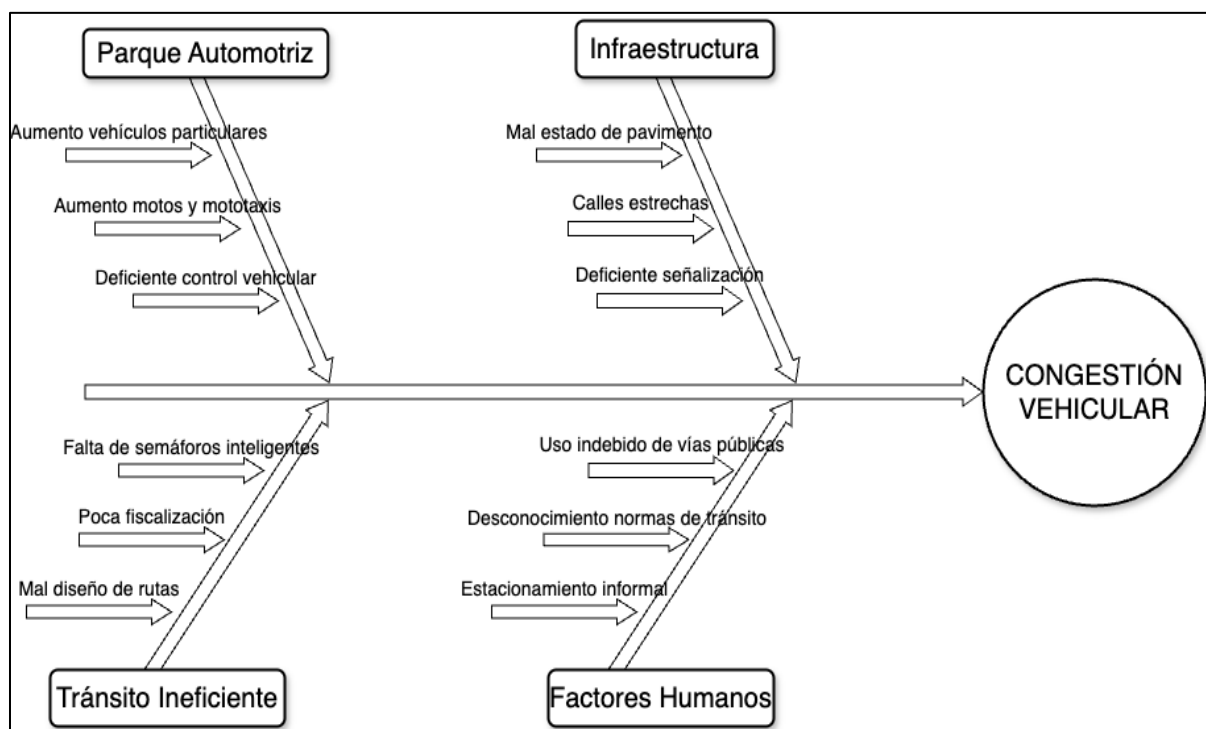
Deficiencias en el control y gestión del tránsito: La ciudad carece de una cantidad adecuada de semáforos inteligentes y sistemas de gestión de tráfico eficientes, lo que, sumado a la insuficiente fiscalización de normas de tránsito, aumenta el desorden vehicular.

Alta concentración de actividades en áreas centrales: Las zonas comerciales, administrativas y educativas concentran gran parte del tráfico diario, incrementando considerablemente la demanda vehicular y contribuyendo directamente al aumento en la congestión.

Este análisis ha permitido establecer claramente los factores clave que el desarrollo de esta investigación y la aplicación móvil propuesta buscan mitigar, mejorando así la gestión del tráfico urbano mediante información actualizada en tiempo real y participación activa de los ciudadanos.

### Figura 8

Diagrama de Ishikawa para la congestión vehicular en Piura, Perú



## 4.2 Requisitos del sistema

Aplicar ingeniería de requisitos ha permitido identificar y documentar de manera precisa las necesidades y expectativas que debe satisfacer la aplicación de monitoreo vehicular.

Para facilitar la organización y trazabilidad de los requisitos, se emplea la siguiente codificación:

RF: Requisito Funcional

RNF: Requisito No funcional

### 4.2.1 *Requisitos funcionales*

Los requisitos funcionales describen las funcionalidades específicas que el sistema debe ofrecer para cumplir con los objetivos del proyecto. A continuación, se detallan los requisitos funcionales identificados para la aplicación:

- RF1. Registro de usuarios:

El sistema debe permitir a los usuarios crear una cuenta utilizando correo electrónico y una contraseña

- RF2. Inicio de sesión:

El sistema debe permitir que los usuarios inicien sesión en la aplicación con sus credenciales

- RF3. Visualización de mapas en tiempo real:

La aplicación debe mostrar un mapa interactivo con la ubicación de los vehículos mediante sensores GPS.

- RF4. Reporte de incidencias:

El sistema debe permitir a los usuarios reportar eventos de tráfico (accidentes, bloqueos, inundaciones).

- RF5. Notificaciones

El sistema debe enviar notificaciones al usuario cuando se detecten incidencias de estado crítico.

- RF6. Consulta de historial

El sistema debe almacenar los datos de tráfico en Firebase para permitir al usuario consultar el historial de incidencias reportadas y generar reportes.

- RF7. Administración de usuarios

Los administradores tendrán acceso a la gestión de cuentas de usuario y supervisión de reportes recibidos.

- RF8. Integración con APIs externas

El sistema debe ser capaz de integrar información proveniente de fuentes externas, como datos de sensores o plataformas oficiales.

#### **4.2.2 Requisitos no funcionales**

Los requisitos no funcionales establecen los criterios de calidad, restricciones y atributos que debe cumplir el sistema para garantizar un desempeño adecuado y una experiencia satisfactoria para el usuario.

- RNF1. Usabilidad

La aplicación debe ofrecer una interfaz amigable, accesible e intuitiva para usuarios no expertos.

- RNF2. Tiempo de respuesta

El sistema debe ser capaz de procesar y mostrar la información del tráfico en tiempo real, con un retraso no mayor a 5 segundos.

- RNF3. Escalabilidad

La arquitectura del sistema debe permitir la integración futura de nuevas funcionalidades y la ampliación de la cobertura geográfica. Compatible con otros Sistemas Operativos.

- RNF4. Seguridad

La información del usuario debe estar protegida mediante mecanismos de autenticación y encriptación de datos sensibles a través de protocolo https

- RNF5. Disponibilidad

La aplicación debe estar disponible al menos el 99% del tiempo, excluyendo periodos programados de mantenimiento.

- RNF6. Mantenibilidad

El sistema debe estar diseñado bajo principios de Clean Architecture para facilitar el mantenimiento e implementación de mejoras a futuro.

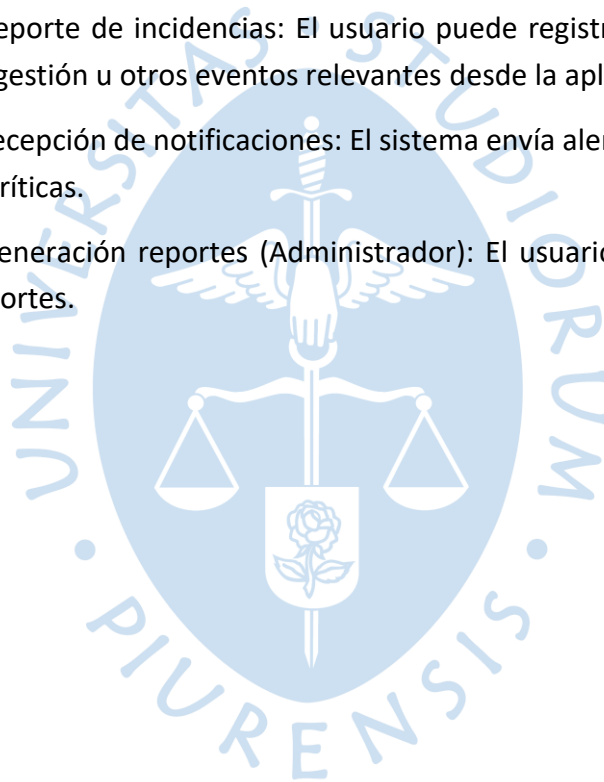
#### **4.3 Diagramas de casos de uso**

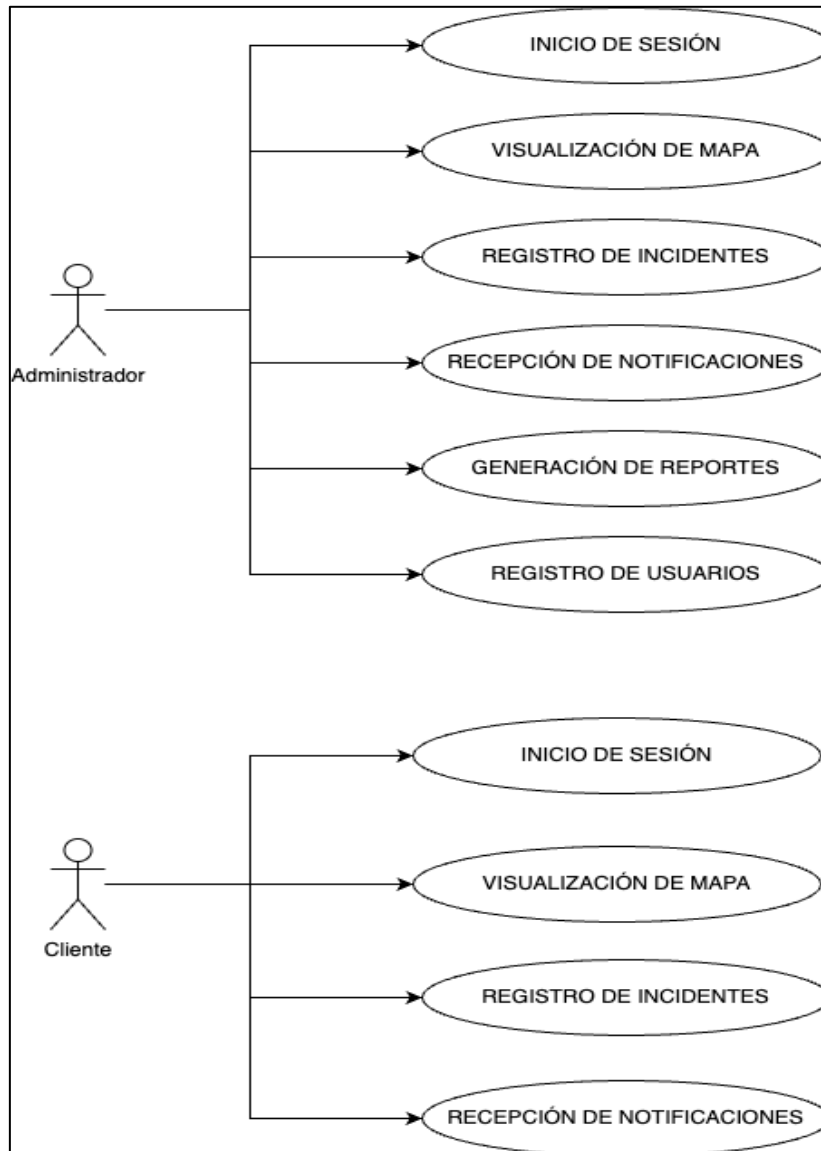
El diagrama de caso de uso constituye una herramienta fundamental para modelar las interacciones entre los diferentes perfiles de usuario y el sistema, permitiendo identificar y describir los escenarios principales de funcionamiento de la aplicación. El caso de uso

representa una secuencia de acciones que permite alcanzar un objetivo específico, tanto del administrador como del cliente.

En la presente investigación, se han definido los siguientes casos de uso para la aplicación de monitoreo vehicular en tiempo real:

- CU1. Registro de usuario: Permite que el administrador genere nuevos usuarios de clientes en la aplicación
- CU2. Inicio de Sesión: Permite a los usuarios autenticarse para acceder a las funcionalidades de la aplicación
- CU3. Visualización de mapa de tráfico: El usuario puede consultar el estado actual del tráfico y las incidencias reportadas en el mapa en tiempo real.
- CU4. Reporte de incidencias: El usuario puede registrar y enviar información sobre accidentes, congestión u otros eventos relevantes desde la aplicación.
- CU5. Recepción de notificaciones: El sistema envía alertas al usuario cuando se detectan incidencias críticas.
- CU6. Generación reportes (Administrador): El usuario administrador tiene la opción de generar reportes.



**Figura 9***Diagrama de casos de uso*

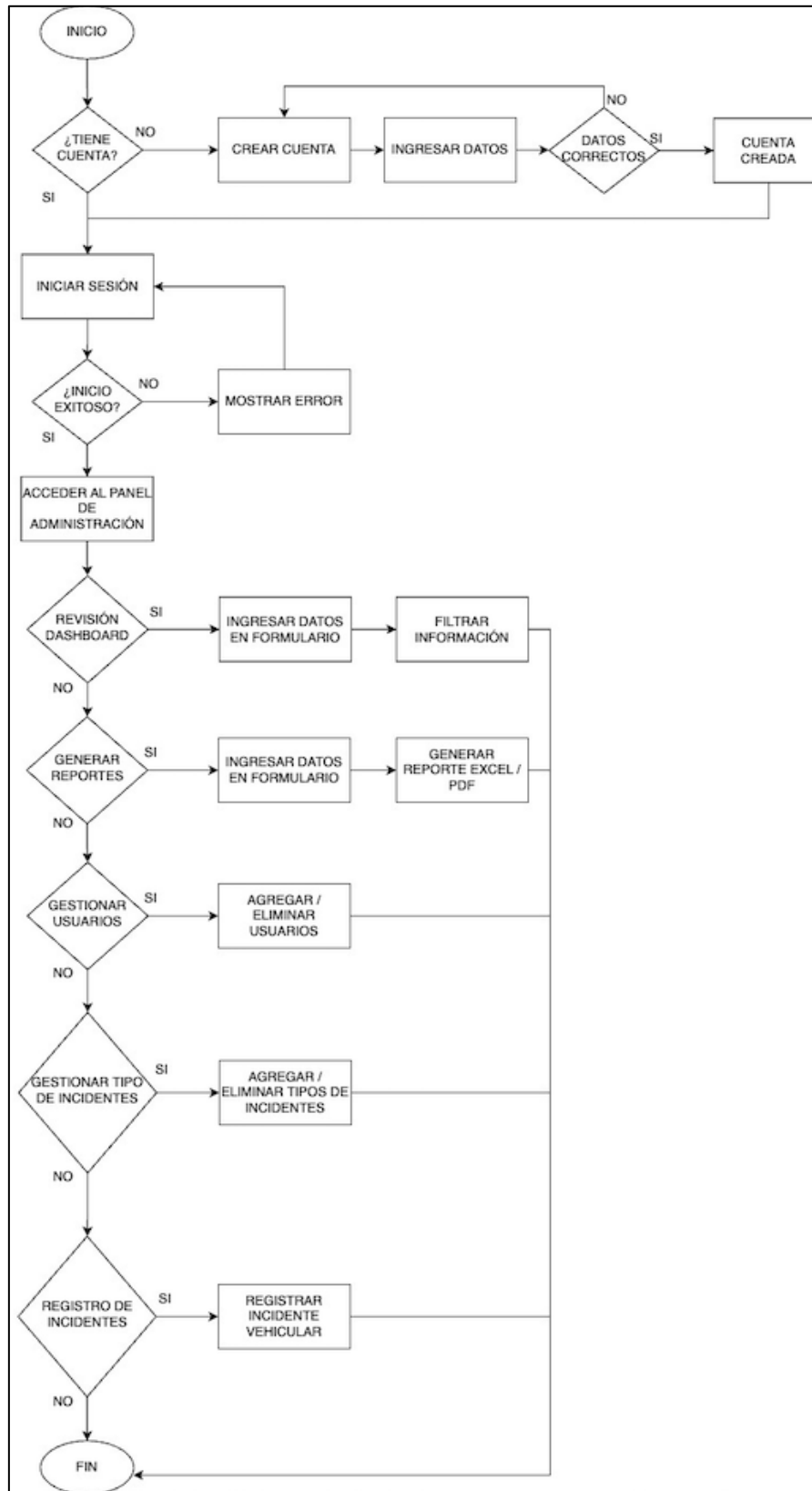
#### 4.4 Diagrama de procesos

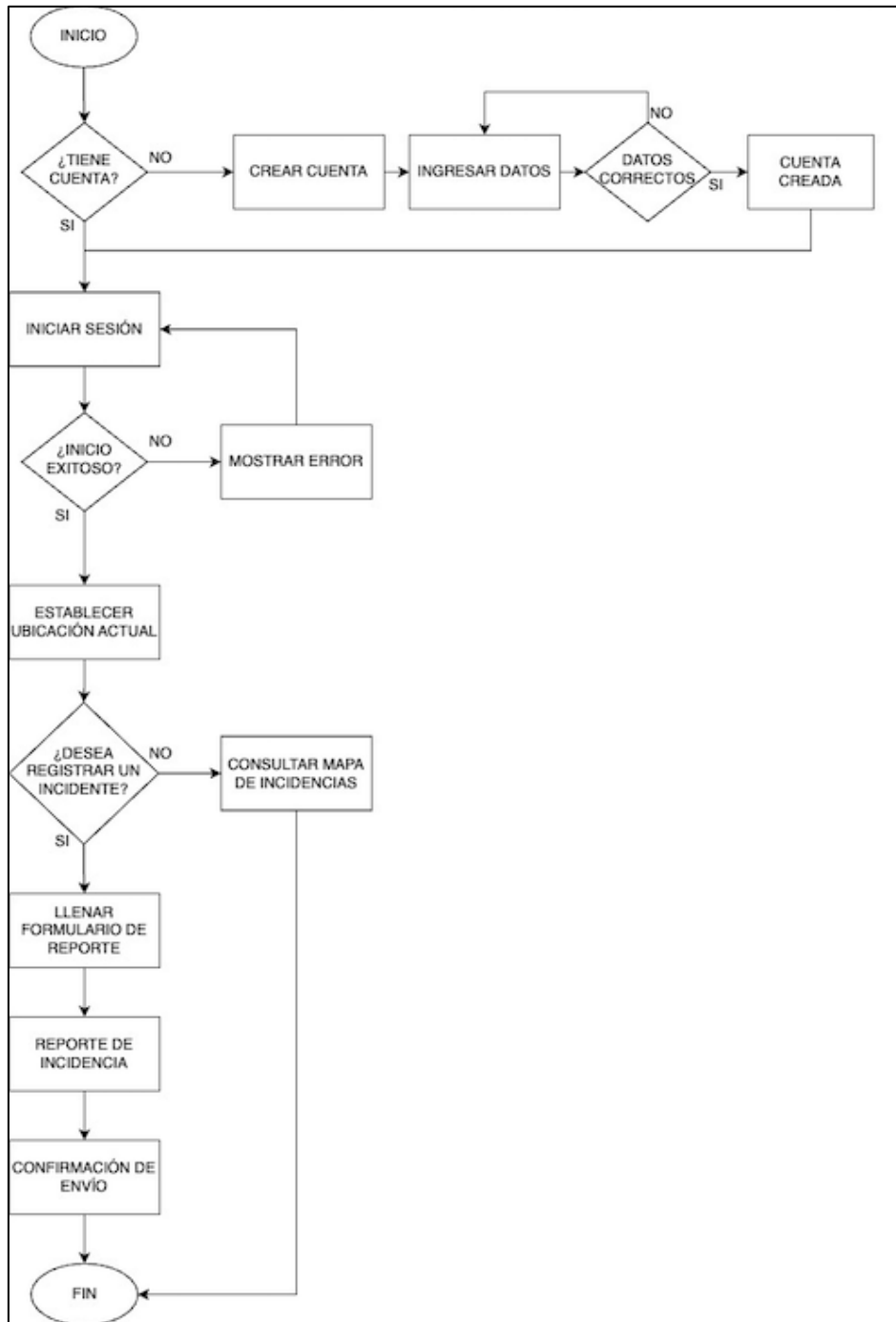
En esta sección se presentan los diagramas de procesos que describen el flujo de información y las actividades clave que ejecuta el aplicativo. Estos diagramas permiten visualizar la secuencia lógica de operaciones, la interacción entre los distintos componentes del sistema, y la manera en que los datos son gestionados desde su origen hasta su procesamiento y presentación al usuario final.

Esta representación gráfica facilita la comprensión de la arquitectura funcional del sistema y constituye una herramienta fundamental para su diseño e implementación, ya que permite identificar cuellos de botella, puntos críticos de decisión y oportunidades de optimización en los procesos.

Figura 10

Diagrama de procesos para el perfil Administrador

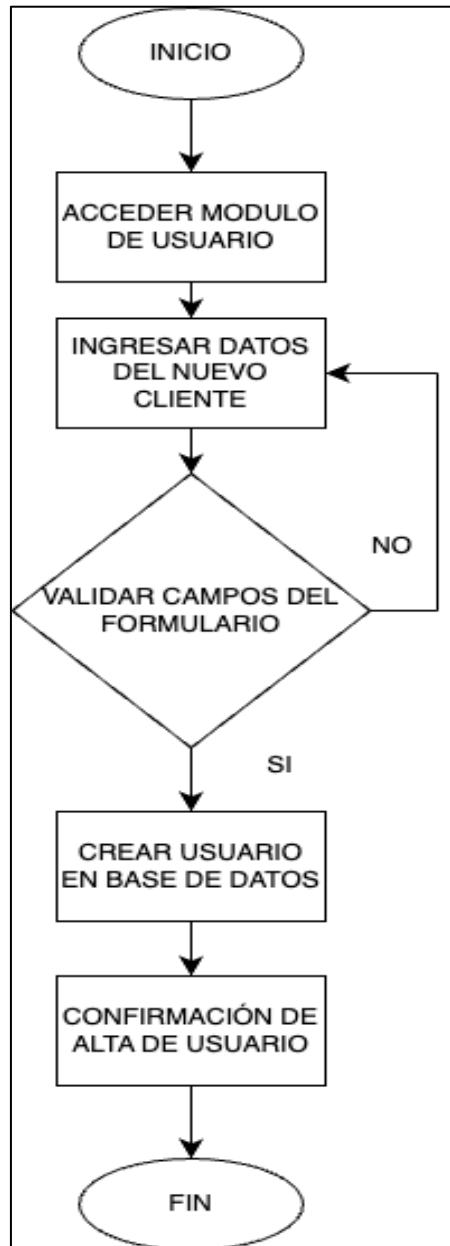


**Figura 11***Diagrama de procesos para perfil Cliente***4.4.1 Diagrama de proceso: Registro de usuario (CU1)**

Este proceso inicia cuando el administrador accede al módulo de gestión de usuarios. El sistema presenta un formulario de registro, en el cual el administrador ingresa los datos del nuevo usuario. Se valida la información ingresada, y si es correcta, el sistema crea el usuario en la base de datos y notifica al administrador del registro exitoso.

**Figura 12**

*Diagrama de proceso: Registro de Usuario*

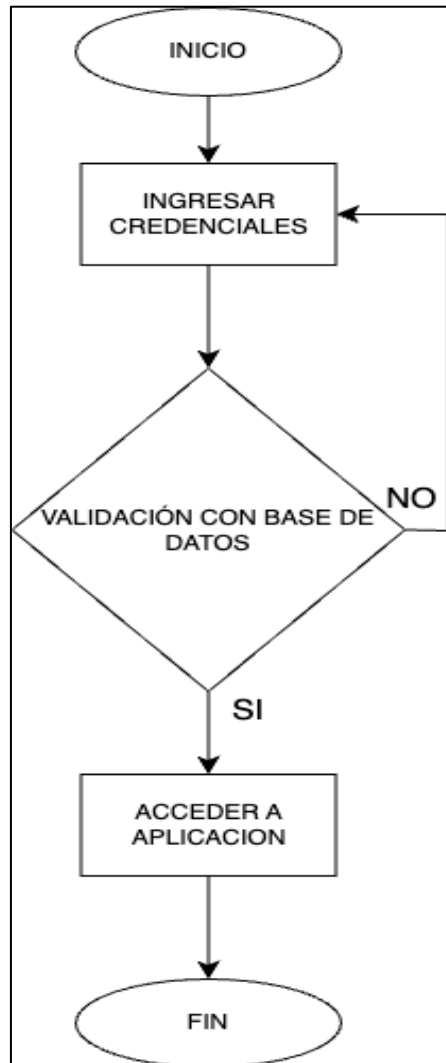


#### **4.4.2 Diagrama de proceso: Inicio de sesión (CU2)**

El proceso comienza cuando el usuario ingresa su correo y contraseña en el formulario de login. El sistema valida las credenciales contra la base de datos. Si son correctas, se concede acceso y se redirige al usuario a la pantalla principal de la aplicación. Si falla, se muestra un mensaje de error.

**Figura 13**

*Diagrama de proceso: Inicio de Sesión*



#### **4.4.3 Diagrama de proceso: Visualización del mapa de tráfico (CU3)**

Una vez autenticado, el usuario accede al mapa. El sistema obtiene en tiempo real los datos de tráfico desde la base de datos o sensores conectados (por ejemplo, reportes de usuarios o dispositivos IoT). Luego, presenta visualmente el estado del tráfico y las incidencias mediante íconos o códigos de color.

**Figura 14**

*Diagrama de proceso:  
Visualización del mapa de tráfico*



Para facilitar la interpretación del estado de tráfico por parte del usuario, se ha implementado un sistema de codificación por colores que clasifica las incidencias según su nivel de criticidad. Esta codificación se representa mediante íconos en el mapa, cuyo significado se detalla en la siguiente tabla:

**Tabla 5**

*Código de color para nivel de criticidad*

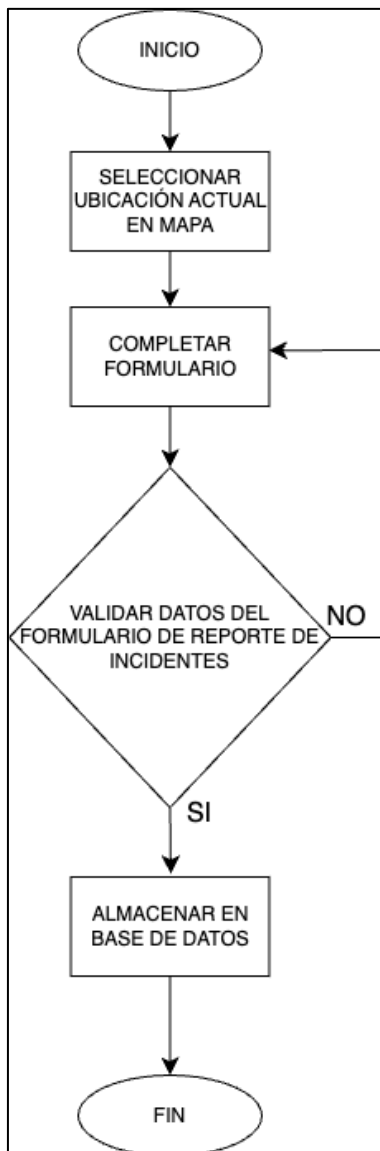
Nivel de criticidad	Descripción breve	Color representativo
Alto	Incidentes graves o bloqueos totales	Rojo
Medio	Congestión moderada	Naranja
Bajo	Advertencias leves	Amarillo

#### 4.4.4 Diagrama de proceso: Reporte de incidencias (CU4)

El usuario selecciona la opción de “Reportar incidencia” e ingresa detalles como tipo de evento, criticidad. Se considera un evento válido cuando ha seleccionado el evento dentro de los tipos (Embotellamiento, Semáforo descompuesto, Accidente de tránsito, Obras en la vía). Se considera un nivel de criticidad válido cuando seleccione Alto, Medio o Bajo. Se contempla reconocimiento de voz. Bajo estas condiciones el sistema valida el formato del reporte, lo almacena en la base de datos y actualiza el mapa de tráfico con la nueva información.

**Figura 15**

*Diagrama de proceso:  
Reporte de Incidencias*

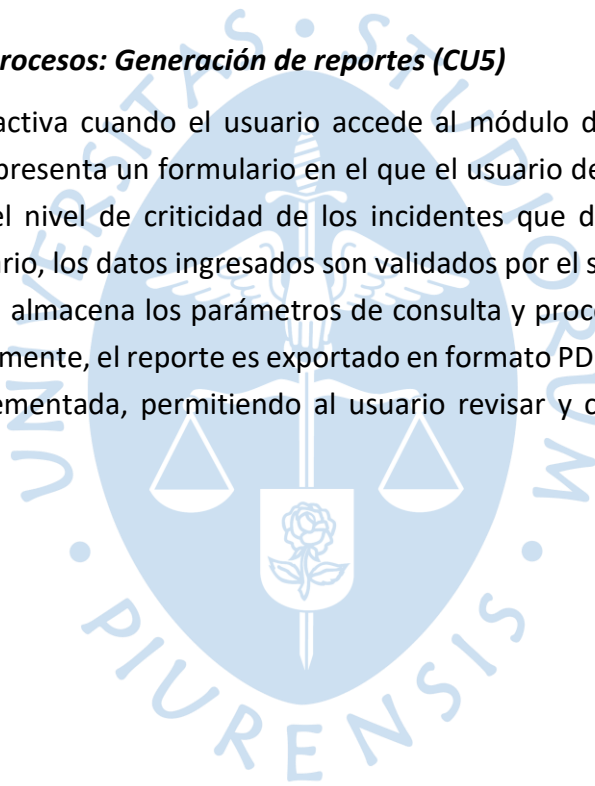


Antes de que un reporte sea almacenado en base de datos, el sistema realiza automáticamente una validación del formato para asegurar que los datos registrados sean claros y completos. Esta validación específica considera:

- Que el usuario haya seleccionado obligatoriamente uno de los íconos grandes disponibles para indicar claramente el tipo de incidente, tales como: Accidente de tránsito, Embotellamiento, Semáforo descompuesto u Obras en la vía.
- Que se haya elegido obligatoriamente un nivel de criticidad mediante íconos visibles, siendo estos niveles: Alto, Medio o Bajo. Esta selección asigna automáticamente el código de color correspondiente en el mapa.
- Que la ubicación GPS del dispositivo esté correctamente capturada y sea válida en tiempo real.

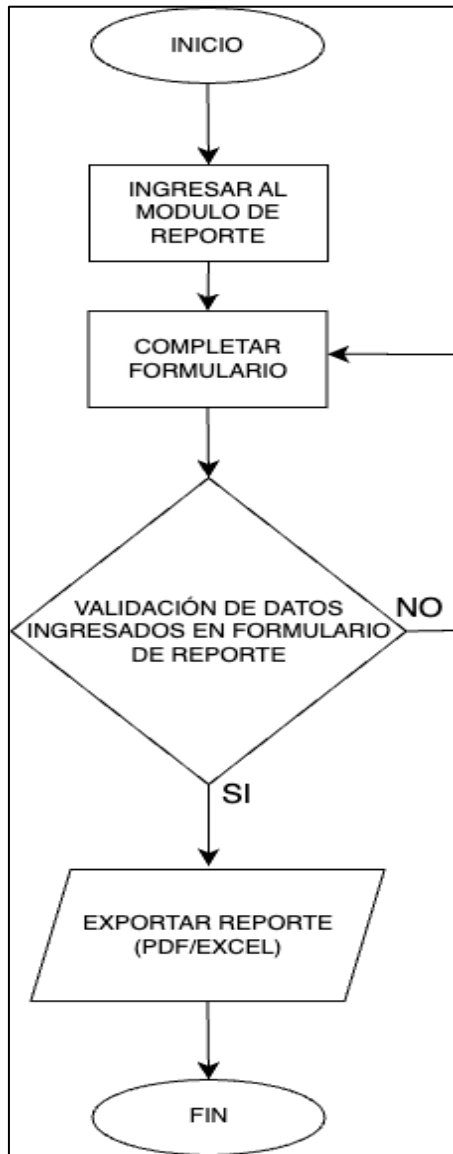
#### **4.4.5 Diagrama de procesos: Generación de reportes (CU5)**

El proceso se activa cuando el usuario accede al módulo de reportes dentro de la aplicación. El sistema presenta un formulario en el que el usuario debe ingresar un rango de fechas y seleccionar el nivel de criticidad de los incidentes que desea consultar. Una vez completado el formulario, los datos ingresados son validados por el sistema. Si la información es correcta, el sistema almacena los parámetros de consulta y procede a generar el reporte correspondiente. Finalmente, el reporte es exportado en formato PDF o Excel, de acuerdo con la funcionalidad implementada, permitiendo al usuario revisar y conservar la información generada.



**Figura 16**

*Diagrama de proceso:  
Generación de reportes*



#### **4.4.6 Diagrama de procesos: Recepción de notificaciones (CU6)**

Actualmente, la aplicación muestra una notificación push en el dispositivo del usuario cada vez que se registra una nueva incidencia. Esta funcionalidad no ha sido implementada de manera explícita en el código, sino que corresponde al comportamiento predeterminado del sistema de mensajería de Firebase (Firebase Cloud Messaging) al utilizar su consola para registrar eventos. Esta característica permite simular el envío de alertas ante nuevos reportes, funcionando como una base para futuras mejoras orientadas a una gestión dinámica de notificaciones en tiempo real.

**Figura 17**

*Diagrama de proceso:  
Recepción de notificaciones*



## Capítulo 5

### Diseño del Sistema

Este capítulo describe el diseño integral del sistema propuesto, desde la arquitectura técnica hasta la organización lógica de datos, interfaces de usuario y la estructura de la base de datos. Se detallan los elementos necesarios para garantizar la funcionabilidad, escalabilidad y usabilidad del sistema de monitoreo vehicular en tiempo real, en función de las necesidades identificadas en capítulos anteriores.

#### 5.1 Diseño de la arquitectura

El diseño arquitectónico del sistema se basa en el enfoque Clean Architecture, propuesto por Robert C. Martin, que promueve la separación de la lógica de negocio y la independencia entre niveles del sistema. Esta arquitectura favorece el mantenimiento, la escalabilidad y la reutilización del código, factores críticos en una aplicación de monitoreo de tráfico en tiempo real que gestiona información dinámica.

##### 5.1.1 *Arquitectura física*

La arquitectura física del sistema está compuesta por los siguientes componentes:

- Aplicación móvil: desarrollada en Flutter, ejecutada en dispositivos con sistema operativo Android, asume el rol de cliente en la arquitectura.
- Plataforma en la nube: se utiliza firebase como Backend-as-a-Service, proporcionando servicios de base de datos en tiempo real, autenticación, almacenamiento y notificaciones push
- Servicios de terceros (API's): se integran servicios de Mapbox, para la visualización de mapas interactivos y geolocalización en tiempo tiempo real.

La comunicación entre cliente y backend se realiza a través de internet, utilizando peticiones HTTP y suscripciones en tiempo real mediante Firebase, lo que garantiza baja latencia y sincronización eficiente.

##### 5.1.2 *Arquitectura Lógica (Clean Architecture)*

La arquitectura Clean Architecture, se organiza en capas concéntricas, donde las dependencias apuntan siempre hacia el núcleo. Las capas definidas en el sistema son:

- Entidad: contiene las reglas de negocio más generales. Aquí se definen los modelos como usuario, reporte junto con sus atributos y comportamientos esenciales.
- Casos de uso: encapsulan la lógica específica de la aplicación. Por ejemplo, generar un reporte de tráfico, consultar vías cercanas o enviar notificaciones.

- Interfaces: adaptan los datos entre los casos de uso y frameworks o controladores. Aquí se encuentran los controladores, DTOs (Data Transfer Object) y los View models.
- Frameworks y drivers: se ubican en la capa más externa, incluye la implementación concentra en Flutter, firebase, mapbox y otros servicios externos.

Esta arquitectura permite que los componentes del dominio y los casos de uso sean independientes del framework utilizado, lo que ofrece una alta capacidad de prueba, protabilidad y bajo acoplamiento entre componentes.

Gracias a la implementación de esta arquitectura junto con Firebase como backend y Flutter como frontend permite un desarrollo ágil, ordenado y orientado a buenas prácticas de ingeniería de software.

## **5.2 Diseño de interfaces**

El diseño de interfaces del sistema se ha centrado en ofrecer una experiencia intuitiva al usuario, clara y accesible. Se empleó una metodología de Diseño Centrado en el Usuario (DCU), considerando las necesidades reales del usuario final y el contexto de uso de la aplicación. La interfaz gráfica fue desarrollada en Flutter, permitiendo una experiencia visual consistente y adaptable a distintos tamaños de pantalla.

### **5.2.1 Pantalla de registro de usuario**

En la Figura 18 se muestra la interfaz de registro de usuarios, diseñada para facilitar el ingreso de nuevos usuarios mediante una presentación clara y amigable. El formulario solicita la siguiente información básica:

- Nombre: Campo destinado para capturar el nombre del usuario
- Apellido: Campo requerido para completar la identidad del usuario.
- Correo electrónico: Se solicita un correo electrónico válido para facilitar la verificación y autenticación del usuario.
- Contraseña: Campo alfanumérico de máximo 9 caracteres.

Una vez completado el formulario, el usuario debe presionar el botón “Registrarse”, claramente destacado mediante un color llamativo que facilita su identificación y uso intuitivo. Adicionalmente, para usuarios ya registrados, la aplicación ofrece un enlace directo “Iniciar sesión”, ubicado estratégicamente dejado del formulario, facilitando el acceso rápido y sencillo a cuentas existentes.

**Figura 18***Interfaz de registro de Usuarios*

Formulario de registro de usuarios con los siguientes campos:

- Nombre (Obligatorio)
- Apellido (Obligatorio)
- Correo Electrónico: Usuario @gmail.com
- Contraseña (Obligatorio)
- Confirmar contraseña (Obligatorio)

Botón Registrarse

**5.2.2 Pantalla de inicio de sesión**

En la Figura 19 fue diseñada con un enfoque minimalista y directo para facilitar la experiencia del usuario. En esta pantalla, el usuario debe ingresar únicamente su correo electrónico previamente registrado en el sistema, claramente etiquetado con la instrucción “Usuario”. Una vez completado, el usuario presiona el botón “Iniciar Sesión”, destacado en color para simplificar la identificación y acción del usuario. El Usuario inicia sesión con un código de verificación que será enviado a través de un correo electrónico a su email

**Figura 19***Interfaz de Inicio de Sesión*

Interfaz de inicio de sesión para PiuraTraffic:

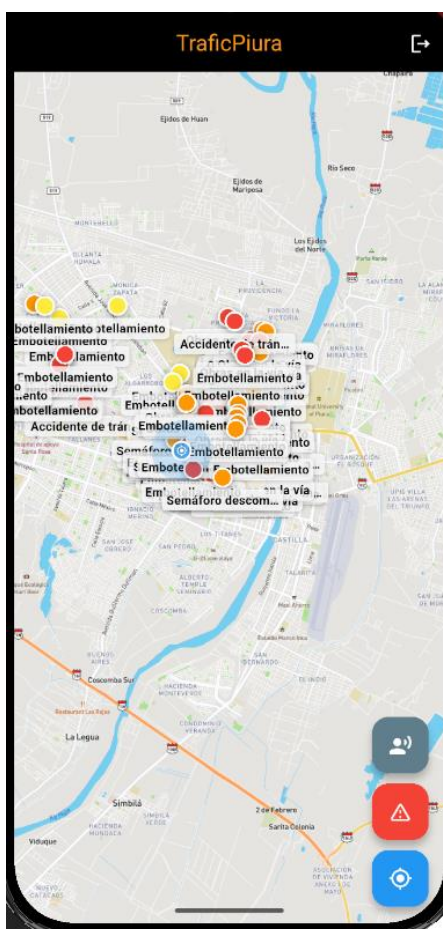
- Logo PiuraTraffic
- Correo Electrónico: Usuario @gmail.com
- Contraseña: Ingresa tu contraseña
- ¿Olvidaste tu contraseña?
- Botón Iniciar Sesión
- ¿No tienes cuenta? [Regístrate](#)

### 5.2.3 Pantalla de monitoreo

La Figura 20 muestra un mapa interactivo que permite al usuario visualizar su ubicación actual en tiempo real, destacada mediante un marcador azul claramente identificable. Esta interfaz incluye botones intuitivos que facilitan las funciones esenciales como reportar incidentes (botón rojo), actualizar la ubicación en el mapa (botón azul) y capturar comandos por voz (botón gris). Esta última funcionalidad busca ofrecer una alternativa ergonómica para el registro de incidencias.

**Figura 20**

*Interfaz Monitoreo de tráfico*



### 5.2.4 Pantalla de registro de incidentes

La aplicación permite a los usuarios registrar de forma rápida y sencilla incidentes viales que ocurren en tiempo real. Para esto, el usuario debe seleccionar dos campos clave mediante botones:

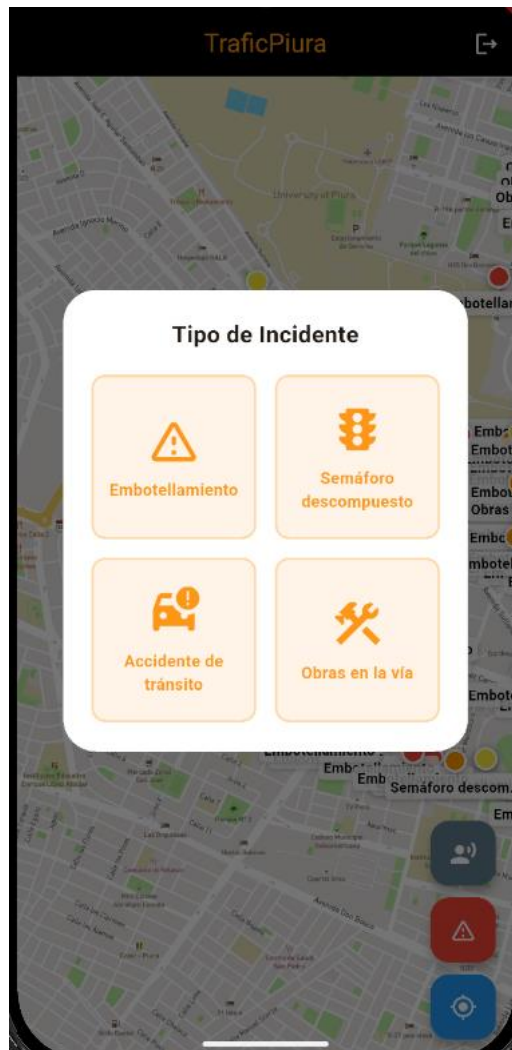
- Tipo de incidente: Como se muestra en la Figura 21, permite clasificar claramente el tipo de problema reportado. Las opciones disponibles son: Accidente de tránsito, Obras en la vía, Semáforo descompuesto, Embotellamiento.

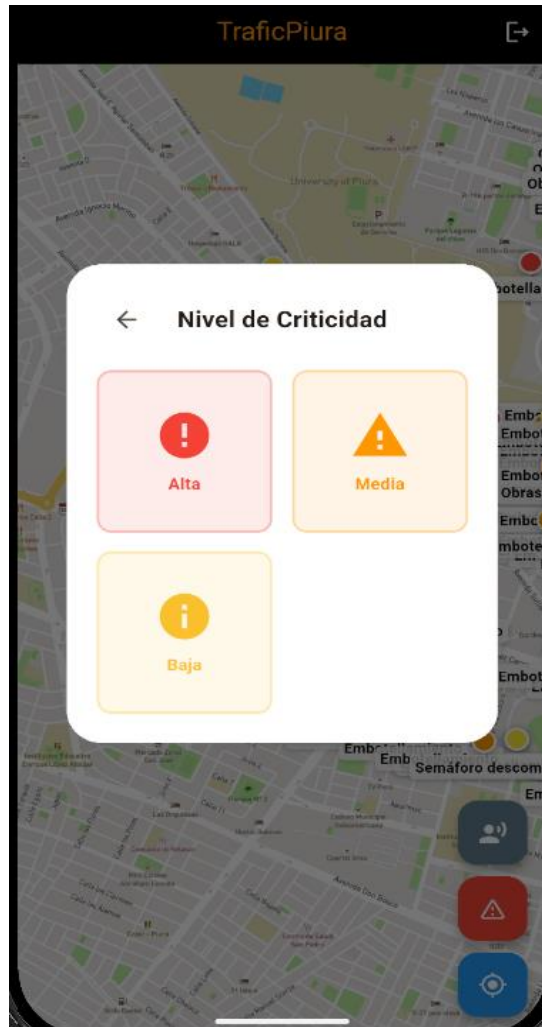
- Nivel de Criticidad: En la Figura 22, indica la criticidad del incidente, proporcionando una referencia sobre su impacto potencial en el tráfico. Las opciones para este campo son: Baja, Media, Alta.

Finalmente, una vez el usuario selecciona el nivel de criticidad, el incidente se registra automáticamente en la base de datos.

**Figura 21**

*Interfaz de selección de accidente de tránsito*



**Figura 22***Interfaz de registro de nivel de criticidad*

### 5.2.5 Pantalla de gestión de usuarios

La Figura 23, permite al administrador del sistema registrar nuevos usuarios en la aplicación. Esta funcionalidad es fundamental para el control de accesos y la personalización de la experiencia según el tipo de usuario. La interfaz presenta un diseño limpio e intuitivo, organizado en un formulario de ingreso de datos.

Los campos disponibles son:

- Nombre: Permite ingresar el nombre de usuario
- Apellido: Campo para el apellido del usuario
- Correo electrónico: Se requiere una dirección válida de correo, que servirá como identificador principal para el inicio de sesión
- Es cliente: Interruptor que permite clasificar al usuario como Cliente o Administrador según sea el caso.

**Figura 23**

*Formulario para creación de usuarios*

**Gestión de Usuarios**

**Agregar Nuevo Usuario**

Nombre  Apellido

Correo Electrónico

Contraseña

Es Cliente

Cancelar

### 5.2.6 Pantalla de generación de reportes

La Figura 24 permite al usuario generar informes detallados sobre los incidentes registrados en la aplicación, facilitando la evaluación del tráfico y la toma de decisiones basados en datos históricos.

Para el módulo de filtros permite personalizar al usuario el reporte que desea generar. Los criterios disponibles son:

- Tipo de incidente: Lista desplegable para seleccionar una categoría específica de incidente o incluir todos.
- Criticidad: Filtro para clasificar los incidentes según el nivel de gravedad.
- Fecha inicio / Fin: campos de selección de rango temporal para delimitar el periodo de análisis.

**Figura 24**

*Módulo de reportes*

**Reportes y Análisis**

**Filtros**

Tipo de Incidente  Criticidad

Todos  Todas

Fecha inicio  Fecha fin

28/04/2025  28/05/2025

Restablecer  Generar PDF

### 5.2.7 Dashboard

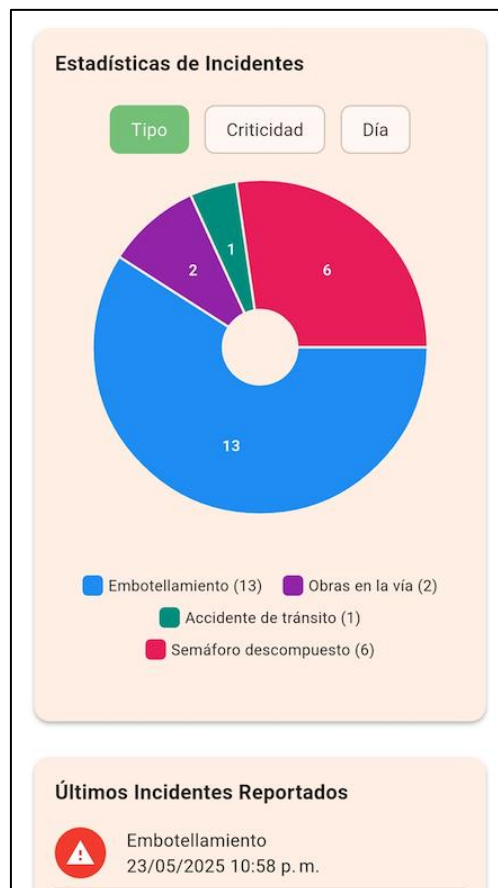
La pantalla de la figura Figura 25, ofrece una visualización rápida y comprensible del estado actual de los incidentes registrados en la aplicación. Está diseñada para facilitar el monitoreo en tiempo real, resumiendo los datos más relevantes de forma visual e interactiva.

En la parte superior se encuentra un gráfico circular que representa la cantidad absoluta de los incidentes registrados, segmentados según el tipo de incidente.

Mediante los botones ubicados en la parte superior del módulo, el usuario puede cambiar la visualización del gráfico para agrupar los datos por criticidad o por día, de acuerdo con las necesidades de su análisis.

**Figura 25**

*Dashboard*



### 5.3 Modelo Entidad Relación

El modelo entidad relación desarrollado para la aplicación, presenta claramente las entidades y sus respectivas relaciones, esenciales para organizar la información sobre los incidentes reportados en la aplicación.

La entidad principal "User", contiene los atributos UID, cellphone, name y lastname. Cada usuario puede reportar cero o más incidentes, estableciéndose así una relación uno a muchos entre User e Incidentes.

La entidad "Incidentes" almacena información clave sobre cada incidente, con atributos tales como ID Incidente, ID Usuario (para identificar quién reportó el incidente), ID Tipo incidente (tipo específico del incidente), Fecha (fecha en que se reportó el incidente) y criticidad (nivel de importancia o gravedad del incidente).

A su vez, la entidad "Incidentes" tiene una relación con otras dos entidades:

- Criticidad: Define el grado de severidad del incidente con atributos ID Criticidad y detalle, relacionados mediante una relación muchos a uno (n:1), lo que indica que múltiples incidentes pueden compartir un mismo nivel de criticidad.
- Tipo de incidente: Clasifica cada incidente reportado. Esta entidad cuenta con atributos ID Incidente y Detalle. La relación es muchos a uno, indicando que varios incidentes pueden estar asociados al mismo tipo de incidente.

Finalmente, la entidad Incidentes genera reportes, estableciendo una relación uno a muchos. La entidad Reportes cuenta con los atributos específicos id reporte e id incidente, lo que permite asociar cada reporte generado a su respectivo incidente registrado en la base de datos.

Este modelo permite gestionar eficientemente la información recopilada en la aplicación, facilitando el monitoreo, clasificación y análisis detallado de los incidentes reportados.

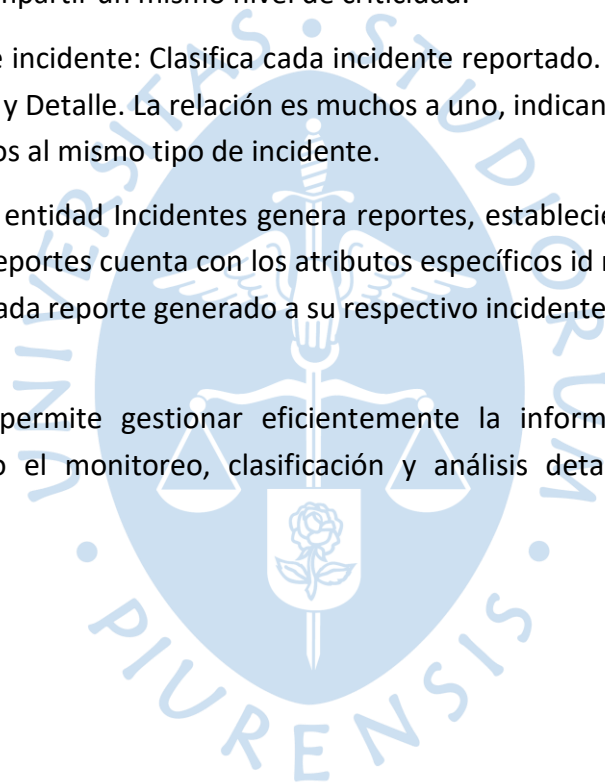
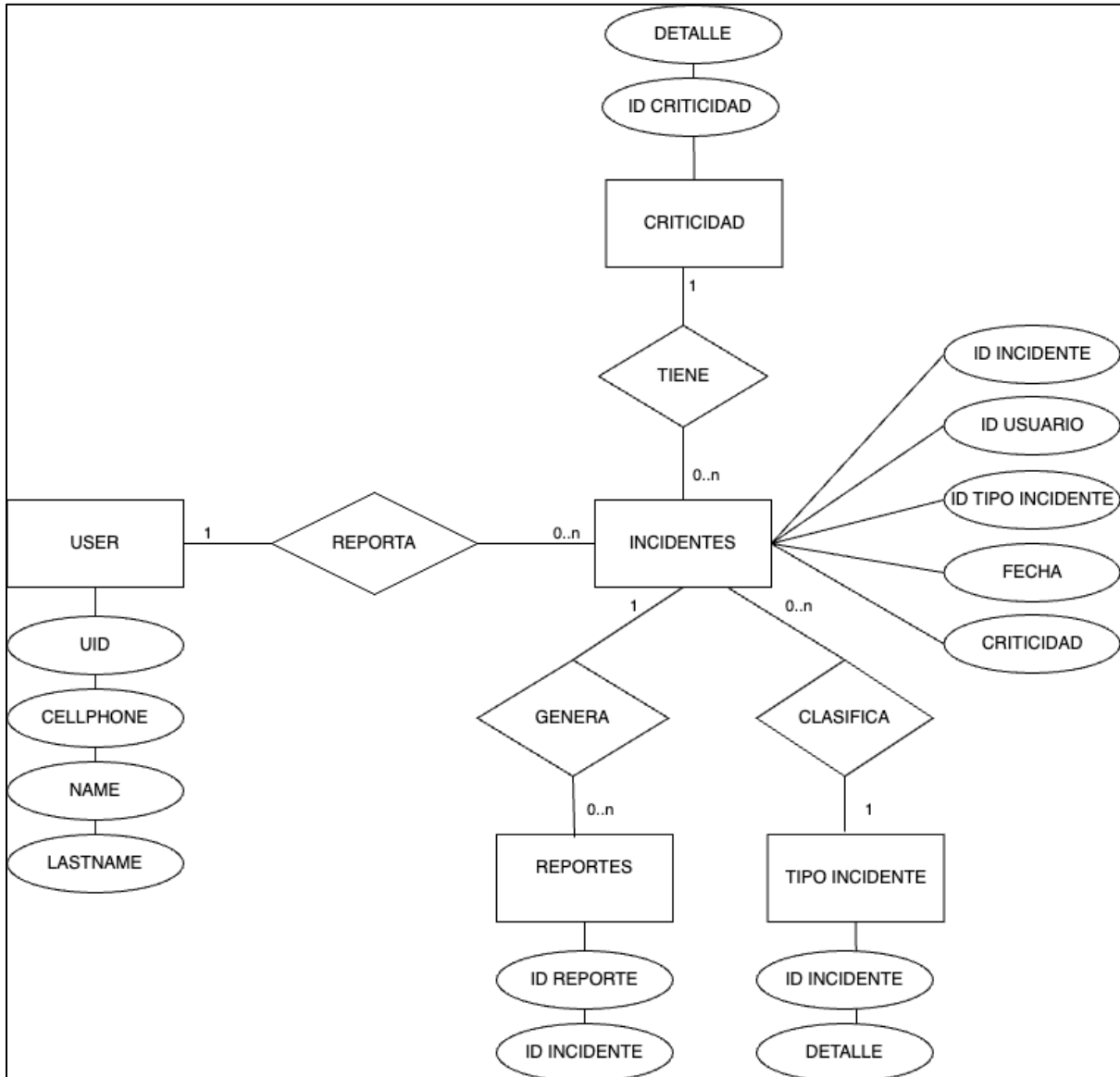


Figura 26

Modelo entidad relación



#### 5.4 Diseño de base de datos

La base de datos fue diseñada siguiendo un modelo relacional con el objetivo de garantizar la integridad de los datos, facilitar consultas eficientes y asegurar una adecuada organización de la información dentro de la aplicación monitoreo vehicular en tiempo real. La estructura se basa en tablas interrelacionadas mediante claves primarias y foráneas.

La tabla principal se denominada Incidentes y almacena los datos clave sobre los eventos reportados. Esta tabla incluye datos como la fecha del incidente y relaciones hacia tablas auxiliares que proveen información adicional necesaria para su identificación completa. Las columnas incluyen IdIncidente (llave primaria), IdUsuario (llave foránea), IdTipoIncidente (llave foránea), Criticidad (llave foránea) y Fecha.

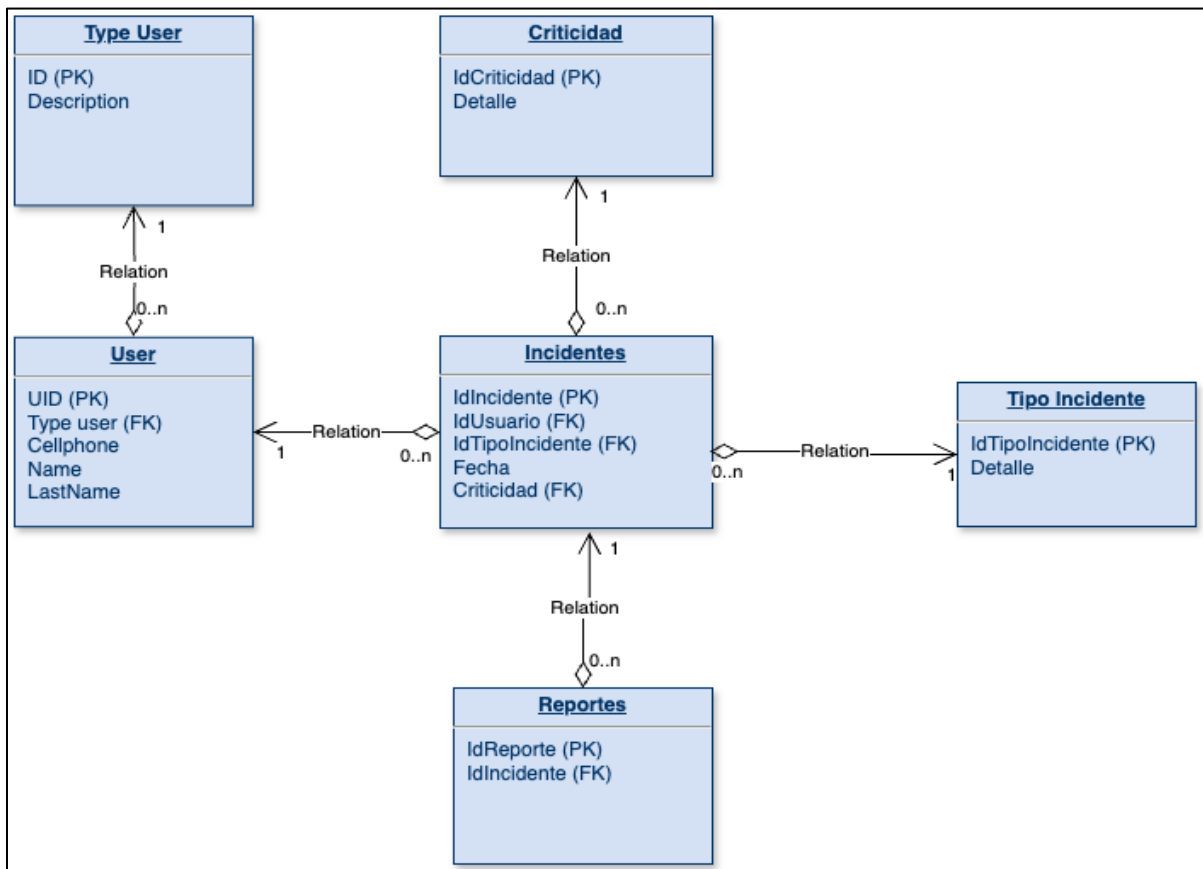
La tabla User, almacena información básica sobre cada usuario del sistema, incluyendo un identificador único (UID), número de teléfono, nombre y apellido. Esta tabla está relacionada con la tabla Type User, que categoriza a los usuarios según sus roles específicos (Administrador, Cliente).

Adicional a ello, se han implementado las tablas Criticidad y Tipo de incidente, se definirán valores predefinidos para una mayor facilidad de registro al cliente. En el caso de criticidad, se han designado los posibles valores como, Alto, Medio y bajo. Tipo de incidente tendrá valores como: Accidente de tránsito, obras en la vía, semáforo descompuesto y embotellamiento.

Finalmente, la tabla reportes se utiliza para registrar los distintos reportes generados sobre un incidente específico, estableciendo una relación de una muchos con la tabla principal incidentes.

**Figura 27**

*Diseño de base de datos relacional*



## Capítulo 6

### Desarrollo de la aplicación

Este capítulo describe de manera integral el proceso seguido para el desarrollo del prototipo funcional de la aplicación móvil. Se exponen las tecnologías seleccionadas para la implementación, los procedimientos llevados a cabo en la creación del prototipo y las pruebas realizadas para validar su correcto funcionamiento y eficacia técnica.

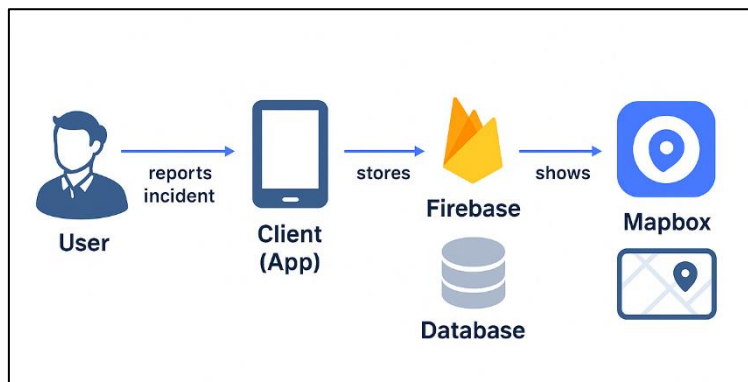
#### 6.1 Tecnologías usadas

Para llevar a cabo la implementación técnica de la aplicación, se seleccionaron tecnologías que aseguran rendimiento, flexibilidad y escalabilidad en la solución final propuesta. A continuación, se detallan las principales tecnologías usadas:

- Flutter (Dart): Framework multiplataforma utilizado para desarrollar la interfaz de usuario y la lógica del lado cliente. Flutter facilita el desarrollo rápido con interfaces fluidas y nativas en dispositivos Android e iOS (Napoli, 2019).
- Mapbox SDK: Herramienta empleada para la gestión y visualización de mapas interactivos. Su integración permite mostrar datos geospaciales y ubicaciones en tiempo real de forma dinámica y eficiente.
- Firebase: Plataforma en la nube de Google utilizada como backend para la gestión y almacenamiento de datos. Permite manejar la información en tiempo real, autenticar usuarios y realizar análisis predictivos mediante integraciones adicionales.
- Clean Architecture: Patrón arquitectónico que garantiza la separación de responsabilidades y facilita el mantenimiento, escalabilidad y evolución del código fuente del proyecto. Su adopción ha permitido implementar una estructura clara y desacoplada entre las diferentes capas del sistema (Martin, 2018).

**Figura 28**

*Arquitectura General de Aplicación Móvil*



Estas tecnologías se escogieron estratégicamente para asegurar la robustez técnica del sistema propuesto, así como para simplificar su futura escalabilidad.

## 6.2 Desarrollo del prototipo

El desarrollo del prototipo se efectuó siguiendo la metodología ágil SCRUM, mediante iteraciones breves y ciclos continuos de prueba y refinamiento (Schwaber & Sutherland, 2020). Las etapas llevadas a cabo durante el desarrollo del prototipo fueron las siguientes:

- **Análisis y definición de requerimientos:** Identificación detallada de los requerimientos funcionales y no funcionales según los objetivos del proyecto.
- **Diseño de interfaces y experiencia de usuario (UX/UI):** Creación de interfaces gráficas intuitivas, diseñadas mediante prototipos visuales utilizando herramientas como Figma, garantizando la experiencia fluida del usuario.
- **Codificación del prototipo:** Implementación de componentes mediante Dart y Flutter, integración con Mapbox para visualización geográfica y configuración de backend en Firebase.
- **Integración de servicios y datos:** Incorporación y consumo de servicios RESTful, así como implementación del acceso en tiempo real a Firebase como plataforma de almacenamiento y gestión de datos.
- **Validación incremental:** Validaciones técnicas constantes para asegurar el cumplimiento de requisitos, verificando la estabilidad, rendimiento y seguridad del prototipo.

El resultado de estas etapas ha sido un prototipo funcional, orientado a facilitar el monitoreo del tráfico en tiempo real en la ciudad de Piura.

En la Figura 29 se muestra una evidencia de la metodología SCRUM (desarrollada en Trello) aplicada para el desarrollo del prototipo, dividiendo el trabajo en 3 sprints semanales. El sprint 1, se centró en la configuración de servicios fundamentales como Firebase, GPS y la integración inicial con Mapbox. En el sprint 2, se implementó el flujo de registro de incidentes en tiempo real e integración de notificaciones push. En el sprint 3, se realizaron las pruebas de usabilidad, así como ajustes finales orientados a la optimización de la aplicación.

Figura 29

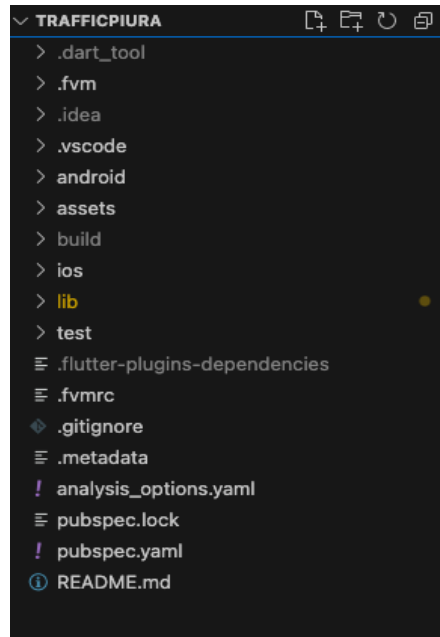
Planificación de sprints bajo metodología Scrum



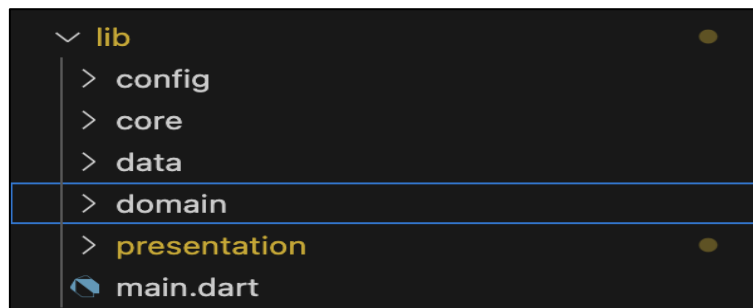
### 6.2.1 Estructura general de la aplicación

El desarrollo del prototipo se llevó a cabo siguiendo los principios de la arquitectura Clean Architecture, que promueve la separación clara entre las capas de presentación, dominio y datos. Esta organización permite escalar y mantener el sistema de forma modular y sostenible. El proyecto presenta una organización estructurada en los siguientes módulos:

- **Lib/core/:** Contiene las clases reutilizables y servicios comunes (como manejo de red, colores, utilidades).
- **Lib/data/:** Incluye modelos, fuentes de datos (datasources) y repositorios que manejan la lógica de acceso a datos remotos o locales.
- **Lib/domain/:** Define las entidades principales y las interfaces de los repositorios.
- **Lib/presentation/:** Contiene la lógica de la interfaz de usuario, páginas, widgets y controladores de estado.
- **Lib/config/:** Gestiona la inyección de dependencias.

**Figura 30***Estructura general de la aplicación*

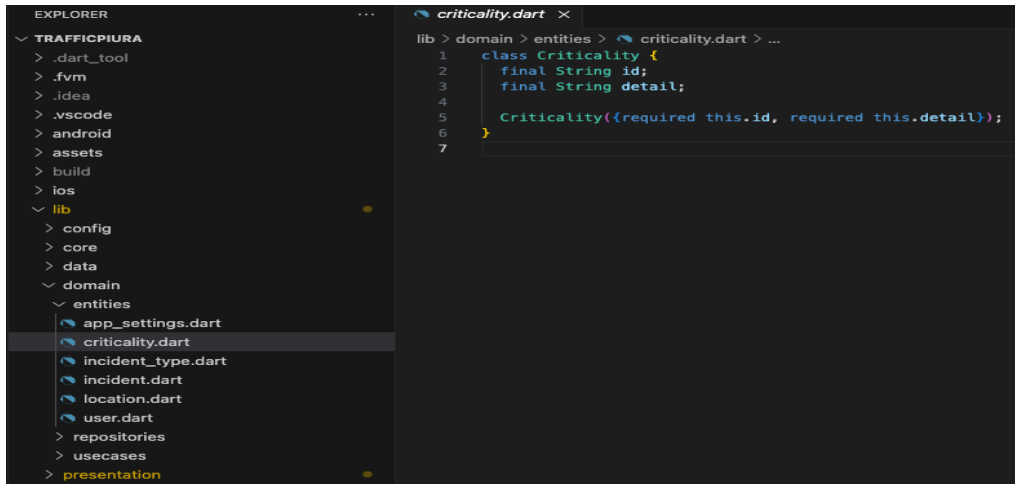
La implementación del prototipo móvil desarrollado en esta investigación se basó en la arquitectura Clean Architecture. En la Figura 31, se puede observar detalladamente la estructura de carpetas del proyecto, evidenciando la separación explícita entre las capas fundamentales: Domain, Data y Presentation.

**Figura 31***Estructura del proyecto basado en Clean Architecture*

A continuación, se muestra un ejemplo específico de implementación en la capa dominio (domain), siguiendo las directrices de Clean Architecture. En la Figura 32, se observa claramente la definición de una entidad llamada "Criticality". Esta clase tiene como objetivo representar uno de los conceptos centrales del negocio, en este caso la criticidad asociada a cada incidencia reportada, destacando así cómo la lógica fundamental del sistema está encapsulada y aislada de otras capas como la interfaz o el manejo de datos externos.

Figura 32

*Ejemplo de clase de dominio (capa Domain)*



```

EXPLORER
  TRAFFICPIURA
    .dart_tool
    .fvm
    .idea
    .vscode
    android
    assets
    build
    ios
    lib
      config
      core
      data
      domain
        entities
          app_settings.dart
          criticality.dart
          incident_type.dart
          incident.dart
          location.dart
          user.dart
      repositories
      usecases
    presentation

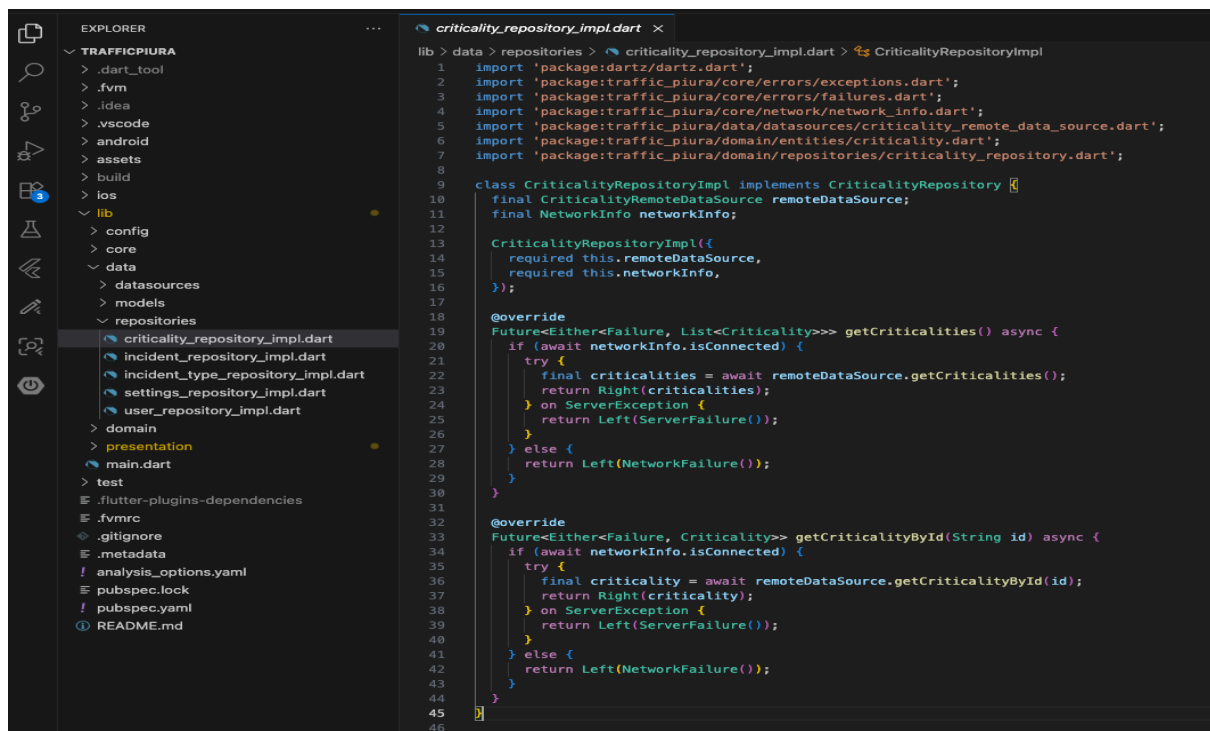
criticality.dart
lib > domain > entities > criticality.dart > ...
1 class Criticality {
2   final String id;
3   final String detail;
4
5   Criticality({required this.id, required this.detail});
6
7

```

En la Figura 33, se presenta un ejemplo de implementación concreta dentro de la capa de datos (Data) del proyecto. La clase “CriticalityRepositoryImpl”, se encarga de obtener los niveles de criticidad desde una fuente de datos remota y representa fielmente la separación de responsabilidades planteada por la arquitectura Clean Architecture.

Figura 33

*Implementación del repositorio en la capa de datos (Data)*



```

EXPLORER
  TRAFFICPIURA
    .dart_tool
    .fvm
    .idea
    .vscode
    android
    assets
    build
    ios
    lib
      config
      core
      data
        datasources
        models
        repositories
          criticality_repository_impl.dart
          incident_repository_impl.dart
          incident_type_repository_impl.dart
          settings_repository_impl.dart
          user_repository_impl.dart
      domain
      presentation
    main.dart
    test
    .flutter-plugins-dependencies
    .fvmrc
    .gitignore
    .metadata
    ! analysis_options.yaml
    pubspec.lock
    pubspec.yaml
    README.md

criticality_repository_impl.dart
lib > data > repositories > criticality_repository_impl.dart > CriticalityRepositoryImpl
1 import 'package:dartz/dartz.dart';
2 import 'package:traffic_piura/core/errors/exceptions.dart';
3 import 'package:traffic_piura/core/errors/failures.dart';
4 import 'package:traffic_piura/core/network/network_info.dart';
5 import 'package:traffic_piura/data/datasources/criticality_remote_data_source.dart';
6 import 'package:traffic_piura/domain/entities/criticality.dart';
7 import 'package:traffic_piura/domain/repositories/criticality_repository.dart';
8
9 class CriticalityRepositoryImpl implements CriticalityRepository {
10   final CriticalityRemoteDataSource remoteDataSource;
11   final NetworkInfo networkInfo;
12
13   CriticalityRepositoryImpl({
14     required this.remoteDataSource,
15     required this.networkInfo,
16   });
17
18   @override
19   Future<Either<Failure, List<Criticality>>> getCriticalities() async {
20     if (await networkInfo.isConnected) {
21       try {
22         final criticalities = await remoteDataSource.getCriticalities();
23         return Right(criticalities);
24       } on ServerException {
25         return Left(ServerFailure());
26       }
27     } else {
28       return Left(NetworkFailure());
29     }
30   }
31
32   @override
33   Future<Either<Failure, Criticality>> getCriticalityById(String id) async {
34     if (await networkInfo.isConnected) {
35       try {
36         final criticality = await remoteDataSource.getCriticalityById(id);
37         return Right(criticality);
38       } on ServerException {
39         return Left(ServerFailure());
40       }
41     } else {
42       return Left(NetworkFailure());
43     }
44   }
45 }
46

```

En el archivo main.dart se inicializa Firebase, tal como se muestra en la Figura 34.

**Figura 34***Evidencia de implementación de Firebase*

```

void main() async {
  WidgetsFlutterBinding.ensureInitialized();
  await initializeDateFormatting('es_PE', null);
  await Firebase.initializeApp();
  await FirebaseAppCheck.instance.activate(

```

La conexión con mapbox se realiza mediante la clase “map\_widget.dart” tal cómo se muestra en la Figura 35. El widget MapboxMap() es utilizado en la interfaz para mostrar el mapa y permitir visualización de incidentes.

**Figura 35***Implementación de API Mapbox*

```

// Capa de tiles de Mapbox
TileLayer(
  urlTemplate:
    "https://api.mapbox.com/styles/v1/mapbox/streets-v11/tiles/{z}/{x}/{y}?access_token={accessToken}",
  additionalOptions: {
    'accessToken':
      'pk.eyJ1IjoiaHJhZmZpY3BpdXJhIiwiaSI6ImNtOHA3OXdsdjA3eDEycXBzZTlhNnprbTAifQ.1bJQM0_DqaRD-P-kkndTgQ',
    'id': 'mapbox.streets',
  },

```

El acceso a base de datos se realiza cada vez que se gestione información de las tablas en Firebase:

**Figura 36***Código de conexión a tablas (collections) de base de datos en Firebase*

```

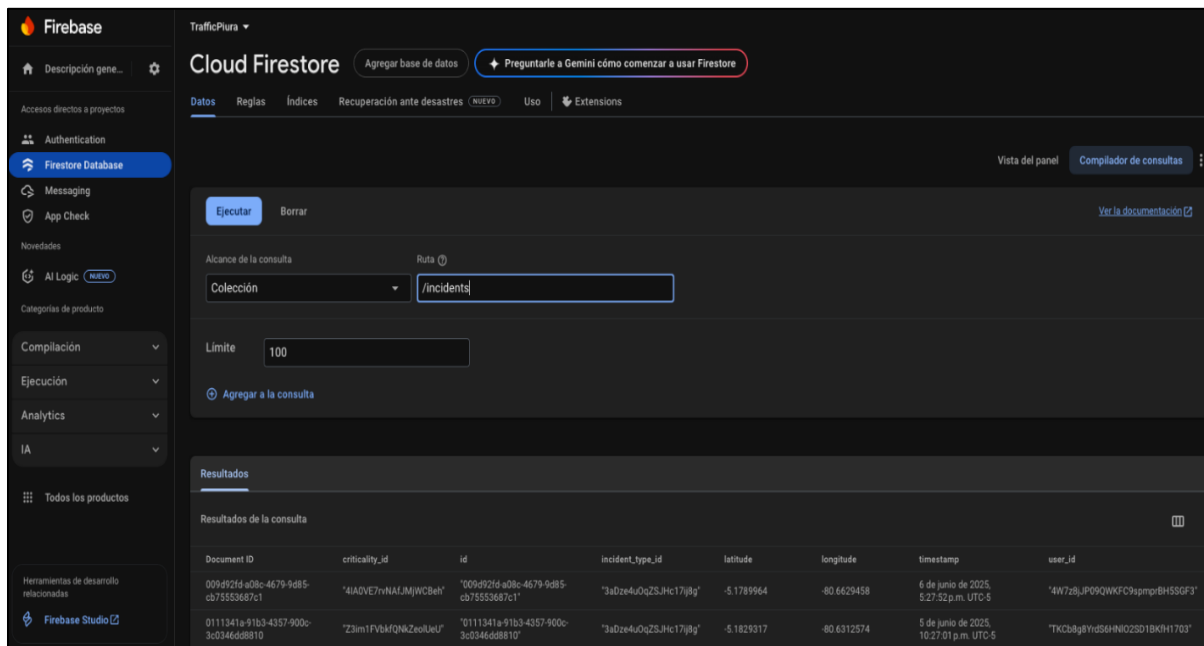
@override
Future<List<IncidentModel>> getIncidents() async {
  try {
    final snapshot = await firestore.collection('incidents').get();

```

En la Figura 37, se muestra la consola de firebase donde se almacenan los registros en base de datos, coordenadas GPS, criticidad, usuario y tipo de incidente.

Figura 37

## Base de datos en Firebase



### 6.3 Implementación de pruebas de usabilidad

Para la evaluación de la usabilidad del prototipo, se utilizó una muestra compuesta por 10 usuarios seleccionados estratégicamente de la ciudad de Piura. Este número se sustenta en estudios realizados por Nielsen Norman Group (2023), los cuales demuestran que con tan solo cinco usuarios es posible identificar hasta el 85% de los problemas de usabilidad más comunes, y que con 10 usuarios se puede alcanzar una cobertura cercana al 95%. Esto se debe al principio de rendimiento decreciente en pruebas de usabilidad, donde cada usuario adicional aporta una menor cantidad de hallazgos nuevos. En este estudio, los participantes fueron seleccionados por su familiaridad con tecnologías móviles y por representar adecuadamente al público objetivo del aplicativo.

Aspectos evaluados:

- Estabilidad general y rendimiento del aplicativo bajo condiciones de uso reales.
- Facilidad y rapidez en el proceso de registro de incidentes vehiculares.
- Claridad en el diseño de la interfaz de usuario.
- Identificación de posibles mejoras mediante retroalimentación directa de los participantes.

Las pruebas se desarrollaron en dos etapas:

Introducción al uso del aplicativo:

- Se explicó claramente el funcionamiento y los propósitos de la aplicación a los participantes.
- Se llevó a cabo proceso de instalación de APK en dispositivos móviles de los usuarios.

Registro práctico de incidentes:

- Cada participante debía registrar al menos dos incidentes vehiculares, reales o simulados, en diferentes ubicaciones de Piura durante una semana.
- Posteriormente, los usuarios completaron un formulario breve donde evaluaron aspectos específicos como facilidad, rapidez y comprensión del proceso de registro.

#### **6.4 Desarrollo del análisis geográfico en la aplicación**

Además de los indicadores de usabilidad, se llevó a cabo un análisis geográfico de los incidentes registrados a través del aplicativo.

Mediante geocodificación inversa se transformaron estas coordenadas en ubicaciones concretas dentro del área urbana de Piura. Posteriormente, se identificaron zonas de concentración o "puntos calientes" de tráfico problemático, los cuales reflejan patrones consistentes con el comportamiento vehicular observado en campo y reportado por otros estudios.



## Capítulo 7

### Resultados y Conclusiones

En este capítulo se presentan los resultados obtenidos tras la aplicación de una encuesta basada en el modelo Petkins-Kinesthetic a un grupo seleccionado de usuarios, con el objetivo de evaluar la eficiencia, eficacia y satisfacción del prototipo desarrollado. Asimismo, se analizan estos resultados en relación con los objetivos específicos definidos al inicio de la investigación, destacando conclusiones relevantes derivadas de las pruebas de usabilidad y su impacto en la comunidad. En total se recopilaron 100 reportes ciudadanos, los cuales incluyeron automáticamente sus coordenadas de latitud y longitud gracias al sistema de geolocalización integrado. Finalmente, se proponen recomendaciones y posibles líneas de trabajo futuros para mejorar y ampliar las funcionalidades del sistema.

#### 7.1 Resultados de las pruebas de usabilidad

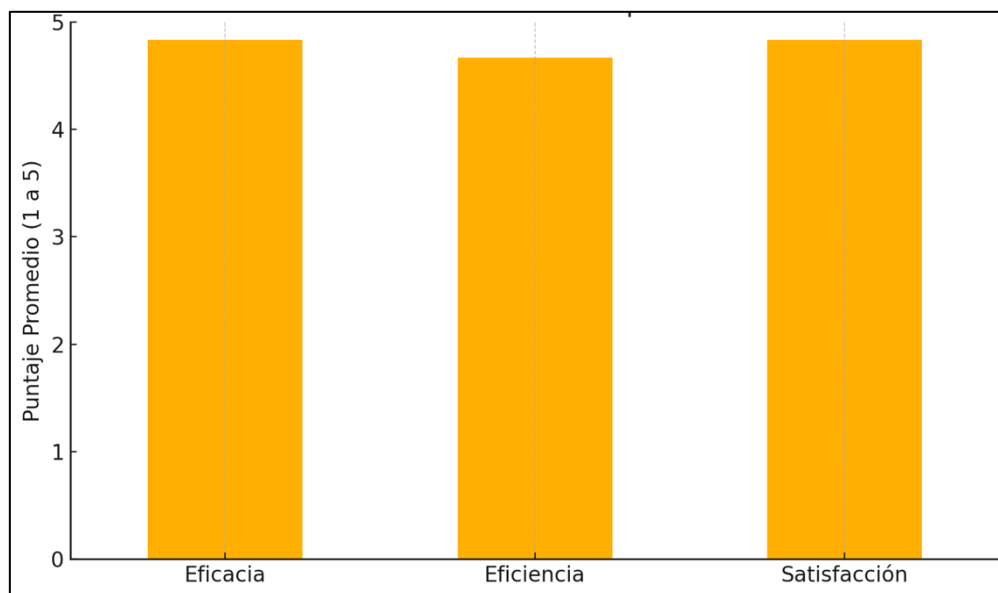
La encuesta aplicada, basada en el modelo Petkins-Kinesthetic, permitió obtener métricas clave como eficacia, eficiencia y satisfacción, logrando un promedio general de usabilidad de 4.57 sobre 5. Por tanto, se considera que la muestra utilizada fue suficiente y representativa para validar la experiencia de uso del sistema, en concordancia con estándares aceptados internacionalmente en el campo de la interacción humano-computadora. La encuesta se detalla en Apéndice A.

Los resultados obtenidos reflejan una evaluación positiva en todos los criterios evaluados:

- **Eficacia:** Promedio general de 4.6, lo que indica que los usuarios fueron capaces de completar las tareas principales (registro y visualización de incidentes) sin dificultades y comprendieron las funcionalidades esenciales del sistema.
- **Eficiencia:** Promedio de 4.4, reflejando que las tareas se realizaron con agilidad, sin necesidad de repetir acciones y con tiempos razonables para completar el registro de incidentes.
- **Satisfacción:** Promedio de 4.7, demostrando una alta aceptación por parte de los usuarios en cuanto a experiencia general con la aplicación, tanto en términos de comodidad como de utilidad percibida.

El promedio general de usabilidad fue de 4.57, lo cual posiciona a la aplicación dentro del rango de “alta usabilidad”. Este resultado respalda la solución desarrollada tanto a nivel de diseño de interfaz como de experiencia de usuario.

El gráfico siguiente resume los puntajes promedio por cada criterio evaluado:

**Figura 38***Promedio de usabilidad por criterio*

Este análisis evidencia que la aplicación cumple con los objetivos planteados, siendo percibida como útil, clara y fácil de usar por parte de los usuarios, lo que refuerza su viabilidad e impacto en el monitoreo colaborativo del tráfico en la ciudad.

## 7.2 Resultados del análisis geográfico de incidencias

Además de los indicadores de usabilidad, se llevó a cabo un análisis geográfico de los incidentes registrados a través del aplicativo. En total se recopilieron 100 reportes ciudadanos, los cuales incluyeron automáticamente sus coordenadas de latitud y longitud gracias al sistema de geolocalización integrado.

Mediante geocodificación inversa se transformaron estas coordenadas en ubicaciones concretas dentro del área urbana de Piura. Posteriormente, se identificaron zonas de concentración o "puntos calientes" de tráfico problemático, los cuales reflejan patrones consistentes con el comportamiento vehicular observado en campo y reportado por otros estudios.

Las principales áreas de concentración fueron:

- Av. Sánchez Cerro (Piura Centro): Alta incidencia de reportes relacionados con congestión y accidentes menores.
- Urb. Miraflores y Av. Progreso: Obstrucciones frecuentes debido a obras viales.
- Av. Los Cocos: Dificultades por vehículos mal estacionados.
- Santa Isabel y Los Geranios: Congestión puntual y bloqueos temporales.

Este análisis evidencia que el aplicativo no solo facilita la recolección de datos de manera colaborativa, sino que también permite identificar zonas críticas para una intervención urbana más precisa, convirtiéndose en una herramienta valiosa para la gestión del tráfico y planificación urbana en la ciudad de Piura. En la Figura 39, se puede observar los diferentes puntos de concentración de congestión vehicular generados por el aplicativo.

**Figura 39**

*Zonas de concentración de tráfico vehicular*



Finalmente, se consideró fundamental validar funcionalmente el sistema y su impacto geoespacial, por lo tanto, se eligió una combinación de encuesta más análisis geográfico para validar la experiencia de uso del prototipo. De esta forma el complemento geoespacial con 100 registros de incidencias, está equilibrando lo cualitativo y cuantitativo de forma muy razonable.

### 7.3 Conclusiones

Se logró diseñar y validar un prototipo funcional de la aplicación móvil propuesta, que permite el registro colaborativo de incidencias viales mediante geolocalización. Aunque el

sistema no fue desplegado como aplicación final, las pruebas de usabilidad y el análisis espacial evidencian su viabilidad y utilidad para el monitoreo del tráfico urbano.

De acuerdo con los objetivos específicos planteados, se concluye lo siguiente:

- **Identificación de causas de congestión vehicular:** A partir del análisis de los incidentes geolocalizados reportados durante las pruebas, se identificaron como principales causas de congestión vehicular en Piura los accidentes de tránsito, las obras públicas y los bloqueos temporales. Este diagnóstico fue posible gracias al enfoque geoespacial del prototipo, que permitió localizar zonas críticas con alta densidad de reportes, como la Av. Sánchez Cerro y la Urb. Miraflores.
- **Arquitectura y diseño funcional:** Se diseñó e implementó un prototipo funcional con una arquitectura modular basada en Clean Architecture, una base de datos estructurada y una interfaz intuitiva. El sistema permite el registro de incidencias en tiempo real y su visualización en un mapa dinámico, lo que demuestra su viabilidad técnica y potencial de escalabilidad.
- **Evaluación de la usabilidad:** Las pruebas de usabilidad aplicadas a 10 usuarios seleccionados revelaron una alta aceptación del prototipo, con un promedio de satisfacción de 4.57/5 en los criterios de eficacia, eficiencia y experiencia de uso. Estos resultados respaldan la utilidad del sistema como herramienta para el monitoreo colaborativo del tráfico urbano.
- **Participación ciudadana mediante enfoque colaborativo (crowdsourcing):** El uso del enfoque de crowdsourcing facilitó la recopilación de datos en tiempo real desde los propios usuarios, promoviendo su involucramiento activo en la solución de problemas urbanos. Esto no solo incrementó la calidad de los datos recolectados, sino que generó un impacto positivo al fomentar la corresponsabilidad ciudadana frente a los desafíos del tráfico en Piura.

#### **7.4 Recomendaciones**

Se proponen las siguientes recomendaciones para futuras iteraciones del sistema:

- **Ampliar el tiempo de visibilidad de los incidentes,** ajustándolo a la velocidad de desplazamiento registrada por el GPS, para ofrecer información más precisa y actualizada.
- **Actualizar el sistema periódicamente,** basándose en la retroalimentación constante de los usuarios, para mantener la relevancia y aceptación del aplicativo.
- **Implementar campañas de difusión y capacitación,** para fomentar el uso frecuente de la aplicación por parte de los ciudadanos, aumentando así la cantidad y calidad de los datos reportados.

#### **7.5 Trabajos Futuros**

Entre las posibles mejoras futuras, se sugiere:

- Validar el sistema en escenarios más diversos, mediante pruebas con usuarios en distintas condiciones de conectividad, edad o experiencia digital.
- Agrupar incidentes geográficamente cercanos, optimizando la visualización y evitando redundancias informativas.



## Referencias

- Alhassan, R., Li, L., & Nkenyereye, L. (2020). Real-time traffic monitoring and data analysis system using mobile applications. *IEEE Access*, 8, 148584-148597. <https://doi.org/10.1109/ACCESS.2020.3016339>
- Atzori, L., Iera, A., & Morabito, G. (2010). The Internet of Things: A survey. *Computer Networks*, 54(15), 2787-2805. <https://doi.org/10.1016/j.comnet.2010.05.010>
- Banco Interamericano de Desarrollo. (2020). Ciudades inteligentes y movilidad urbana sostenible en América Latina. <https://publications.iadb.org>
- Beck, K., & Andres, C. (2004). *Extreme programming explained: Embrace change* (2ª ed.). Addison-Wesley Professional.
- Burrough, P. A., & McDonnell, R. A. (1998). *Principles of geographical information systems*. Oxford University Press.
- Cano, A., López, G., & Mendoza, F. (2021). Aplicaciones móviles en el monitoreo del tráfico vehicular urbano. *Revista Iberoamericana de Tecnología en Educación y Educación en Tecnología*, (27), 12-21.
- Castillo, C., & Huamán, E. (2020). Movilidad urbana y congestión vehicular en ciudades intermedias del Perú. *Revista de Transporte y Territorio*, (23), 54-68.
- Cepal. (2020). *Movilidad urbana sostenible en América Latina y el Caribe*. Comisión Económica para América Latina y el Caribe. <https://www.cepal.org>
- Cohn, M. (2009). *Succeeding with agile: Software development using Scrum*. Addison-Wesley Professional.
- Creswell, J. W., & Creswell, J. D. (2017). *Research design: Qualitative, quantitative, and mixed methods approaches* (5ª ed.). SAGE Publications.
- Deng, Y., Hu, Y., & Wang, F. (2020). Mobile network data for urban mobility analysis: A review. *Cities*, 102, 102720. <https://doi.org/10.1016/j.cities.2020.102720>
- Dierks, T., & Rescorla, E. (2008). *The Transport Layer Security (TLS) Protocol Version 1.2*. RFC 5246. <https://datatracker.ietf.org/doc/html/rfc5246>
- Extrack. (2023). *Extrack – Expense Tracker*. Google Play Store. Recuperado de <https://play.google.com/store/apps/details?id=com.falconsoftware.extrack>
- Fowler, M. (2019). *Refactoring: Improving the design of existing code* (2ª ed.). Addison-Wesley Professional.
- Gama, J., Žliobaitė, I., Bifet, A., Pechenizkiy, M., & Bouchachia, A. (2019). A survey on concept drift adaptation. *ACM Computing Surveys*, 46(4), 44. <https://doi.org/10.1145/2523813>

- Gamma, E., Helm, R., Johnson, R., & Vlissides, J. (1995). *Design patterns: Elements of reusable object-oriented software*. Addison-Wesley.
- García, M., López, J., & Ruiz, D. (2020). Desarrollo de sistemas inteligentes para ciudades sostenibles. *Revista de Tecnología Urbana*, 15(2), 45–56.
- González, A., Martínez, D., & Rivas, M. (2018). Seguridad y protección de datos personales en aplicaciones móviles. *Revista Latinoamericana de Tecnología*, 17(2), 45–58.
- González, M., Rojas, C., & Vera, L. (2021). Aplicaciones móviles y gestión del tráfico urbano: estudio de caso en ciudades intermedias. *Revista Latinoamericana de Ingeniería*, 35(2), 45–58.
- Goodchild, M. (2007). Citizens as sensors: The world of volunteered geography. *GeoJournal*, 69(4), 211–221.
- Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. MIT Press. <https://www.deeplearningbook.org/>
- Google Developers. (2023). *Google Maps Platform documentation*. <https://developers.google.com/maps>
- Google Developers. (2024). *Flutter documentation*. Recuperado el 15 de abril de 2025, de <https://flutter.dev/docs>
- Google. (n.d.). *Firestore Realtime Database Documentation*. Recuperado de <https://firebase.google.com/docs/database>
- INEI. (2022). *Compendio estadístico del departamento de Piura 2022*. Instituto Nacional de Estadística e Informática.
- INEI. (2023). *Estadísticas de transporte y comunicaciones 2022*. Instituto Nacional de Estadística e Informática. <https://www.inei.gob.pe>
- Kaplan, E. D., & Hegarty, C. J. (2006). *Understanding GPS: Principles and Applications* (2nd ed.). Artech House.
- Kaplan, E. D., & Hegarty, C. J. (2017). *Understanding GPS: Principles and Applications*. Artech House.
- Kreps, J., Narkhede, N., & Rao, J. (2011). Kafka: A distributed messaging system for log processing. In *Proceedings of the NetDB* (pp. 1-7).
- Land Transport Authority (LTA). (2020). *Smart Mobility 2030: Singapore's transport master plan*.
- Li, Q., Gao, F., Wang, Q., & Chen, S. (2022). Intelligent traffic management system based on IoT and cloud computing. *IEEE Access*, 10, 34421-34435. <https://doi.org/10.1109/ACCESS.2022.3152420>

- Li, T., Zhang, Y., & Hu, W. (2021). Mobile applications for urban traffic: A review of tools and trends. *Journal of Urban Mobility*, 12(1), 35–48.
- Litman, T. (2022). *Urban Traffic Congestion: The Problem and How to Solve It*. Victoria Transport Policy Institute.
- Longley, P. A., Goodchild, M. F., Maguire, D. J., & Rhind, D. W. (2015). *Geographic Information Systems and Science* (4th ed.). Wiley.
- Mapbox. (2024). Mapbox documentation. Recuperado el 15 de abril de 2025, de <https://docs.mapbox.com>
- Martin, R. C. (2018). *Clean Architecture: A Craftsman's Guide to Software Structure and Design*. Prentice Hall.
- Ministerio de Transportes y Comunicaciones del Perú. (2019). Plan nacional de infraestructura para la competitividad 2019–2025. Gobierno del Perú.
- Moroney, L. (2022). *Firestore Essentials - Android Edition*. Packt Publishing.
- Municipalidad Metropolitana de Lima. (2021). Sistema de monitoreo de tráfico.
- Municipalidad Provincial de Piura. (2021). Plan de desarrollo urbano y vial sostenible 2021–2030.
- Napoli, M. (2019). *Beginning Flutter: A Hands On Guide to App Development*. Wiley.
- PassKeep. (2023). PassKeep Password Manager. Google Play Store. Recuperado de <https://play.google.com/store/apps/details?id=com.marcopolopass.passkeep>
- Pressman, R. S., & Maxim, B. R. (2015). *Software engineering: A practitioner's approach* (8<sup>a</sup> ed.). McGraw-Hill Education.
- Sakr, S., & Gaber, M. M. (Eds.). (2014). *Stream Processing with Apache Storm*. Springer.
- Sauro, J., & Lewis, J. R. (2016). *Quantifying the User Experience: Practical Statistics for User Research* (2nd ed.). Morgan Kaufmann.
- Schwaber, K. & Sutherland, J. (2020). *The Scrum Guide: The Definitive Guide to Scrum*. Scrum.org.
- Senamhi. (2022). Informe técnico sobre el Fenómeno El Niño en el norte del Perú. Servicio Nacional de Meteorología e Hidrología del Perú.
- Shi, Q., Liu, R., & Du, Y. (2019). Design and implementation of a real-time traffic event notification system. *Procedia Computer Science*, 147, 543-548. <https://doi.org/10.1016/j.procs.2019.01.116>
- Silva, R., & Martínez, L. (2020). Tecnologías colaborativas en ciudades emergentes: el caso de Waze. *Revista Latinoamericana de Movilidad*, 8(1), 33–44.

- Sommerville, I. (2015). *Software engineering* (10ª ed.). Pearson Education.
- Suárez, P., & Hernández, D. (2022). Tecnología y movilidad: retos en la descentralización de soluciones inteligentes. *Revista de Movilidad y Transporte*, 14(1), 22–34.
- Sui, D., Goodchild, M., & Elwood, S. (2013). *Crowdsourcing Geographic Knowledge: Volunteered Geographic Information (VGI) in Theory and Practice*. Springer.
- SUNARP. (2023). Reporte de crecimiento vehicular en el norte del Perú. Superintendencia Nacional de los Registros Públicos.
- Torres, A., & Fernández, J. (2023). Desarrollo de soluciones colaborativas para mejorar la movilidad urbana en ciudades peruanas. Tesis de maestría, Universidad Nacional de Ingeniería.
- Tullis, T., & Albert, B. (2013). *Measuring the User Experience: Collecting, Analyzing, and Presenting Usability Metrics* (2nd ed.). Morgan Kaufmann.
- Wang, D., Cao, J., Li, J., & He, Y. (2018). DeepTTE: Prediction of travel time estimation based on deep learning. *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 861–870. <https://doi.org/10.1145/3219819.3219825>
- Wang, Y., Wang, Z., Liu, Y., Zhang, J., & Liu, L. (2019). Urban Traffic Monitoring Based on Wi-Fi Signals. *Sensors*, 19(3), 613. <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6359557/>
- Ydrogo Ramírez, T. (2020). Desarrollo de un aplicativo móvil para el registro del cumplimiento y desempeño del transporte urbano en Lima [Tesis de licenciatura, Universidad Continental]
- Yin, R. K. (2017). *Case study research and applications: Design and methods* (6ª ed.). SAGE Publications.
- Zhang, L., Zhou, K., & Yang, X. (2017). Urban traffic prediction based on GPS and mobile device data: A review. *IEEE Transactions on Intelligent Transportation Systems*, 18(12), 3135–3146. <https://doi.org/10.1109/TITS.2017.2671539>
- Zheng, Y., Capra, L., Wolfson, O., & Yang, H. (2014). Urban computing: Concepts, methodologies, and applications. *ACM Transactions on Intelligent Systems and Technology*, 5(3), 1–55. <https://doi.org/10.1145/2629592>
- Zhu, Y., Zhang, S., & Wang, Y. (2022). Urban traffic congestion identification and analysis based on GPS trajectory data. *Sustainability*, 14(4), 2157. <https://doi.org/10.3390/su14042157>

## Apéndices



**Apéndice 1 Instrumentos de evaluación de usabilidad de la aplicación móvil**

1.- ¿Pudiste completar las tareas de registro de incidentes sin dificultades?

Muy de acuerdo

De acuerdo

Neutral

En desacuerdo

Muy en desacuerdo

2.- ¿El sistema respondió de forma correcta a tus acciones? (registro y visualización de incidentes)

Muy de acuerdo

De acuerdo

Neutral

En desacuerdo

Muy en desacuerdo

3.- ¿La información mostrada en el mapa de incidentes te ayudó a entender el estado del tráfico en tu zona?

Muy de acuerdo

De acuerdo

Neutral

En desacuerdo

Muy en desacuerdo

4.- ¿Las instrucciones y etiquetas al registrar o visualizar incidentes fueron clara?

Muy de acuerdo

De acuerdo

Neutral

En desacuerdo

Muy en desacuerdo

5.- ¿Encontraste fácilmente las opciones que necesitas usar?

Muy de acuerdo

De acuerdo

Neutral

En desacuerdo

Muy en desacuerdo

6.- ¿Consideras que la aplicación tiene una buena velocidad de respuesta?

Muy de acuerdo

De acuerdo

Neutral

En desacuerdo

Muy en desacuerdo

7.- ¿Te pareció clara la información mostrada en el mapa de monitoreo?

Muy de acuerdo

De acuerdo

Neutral

En desacuerdo

Muy en desacuerdo

8.- ¿Te sentiste cómodo utilizando la interfaz de la aplicación?

Muy de acuerdo

De acuerdo

Neutral

En desacuerdo

Muy en desacuerdo

9.- ¿Consideras que la aplicación es útil para reportar incidentes de tráfico?

Muy de acuerdo

De acuerdo

Neutral

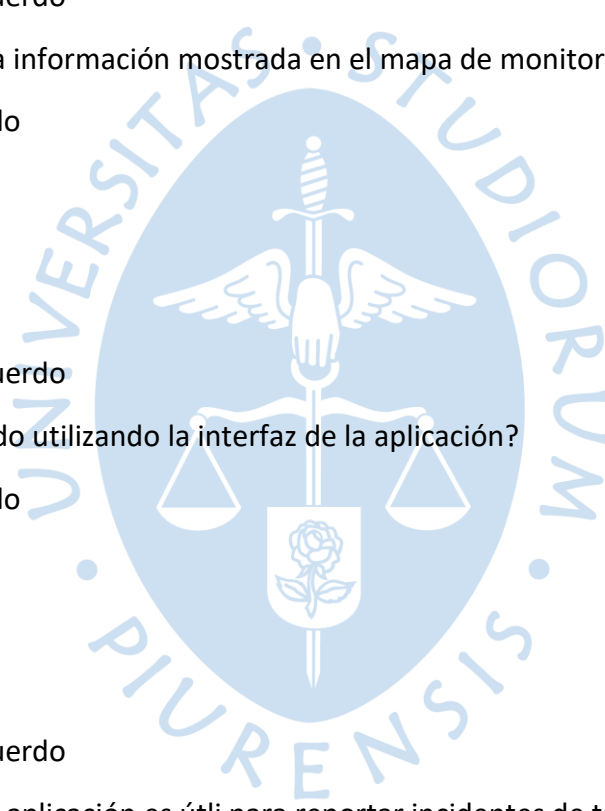
En desacuerdo

Muy en desacuerdo

10.- ¿Recomendarías esta aplicación a otras personas?

Muy de acuerdo

De acuerdo



Neutral

En desacuerdo

Muy en desacuerdo

**Apéndice 2 Reporte de Incidentes de tráfico período 02/06/2025 – 07/08/2025**

## Resumen

Métrica	Valor
Total de incidentes	100
Período (días)	7
Promedio diario	15.14
Fecha más reciente	07/06/2025

## Distribución por tipo de incidente

Tipo de incidente	Cantidad	Porcentaje
Embotellamiento	76	71.7%
Obras en la vía	23	21.7%
Semáforo descompuesto	2	1.9%
Accidente de tránsito	5	4.7%

## Registro de incidentes

Fecha	Tipo	Criticidad	Ubicación
07/06/2025 13:15	Embotellamiento	Alta	Lat: -5.182501, Long: -80.633719
07/06/2025 13:12	Embotellamiento	Media	Lat: -5.181802, Long: -80.640074
07/06/2025 13:11	Obras en la vía	Media	Lat: -5.180360, Long: -80.640013
07/06/2025 13:10	Embotellamiento	Baja	Lat: -5.176860, Long: -80.641203
07/06/2025 13:07	Embotellamiento	Baja	Lat: -5.165314, Long: -80.651790

Fecha	Tipo	Criticidad	Ubicación
06/06/2025 19:03	Obras en la vía	Baja	Lat: -5.186156, Long: -80.627171
06/06/2025 19:01	Semáforo descompuesto	Alta	Lat: -5.188367, Long: -80.626910
06/06/2025 18:59	Obras en la vía	Media	Lat: -5.192749, Long: -80.626227
06/06/2025 18:59	Accidente de tránsito	Alta	Lat: -5.193121, Long: -80.626324
06/06/2025 18:58	Embotellamiento	Media	Lat: -5.193424, Long: -80.625667
06/06/2025 18:26	Embotellamiento	Baja	Lat: -5.194287, Long: -80.634859
06/06/2025 17:27	Embotellamiento	Alta	Lat: -5.178996, Long: -80.662946
06/06/2025 17:24	Embotellamiento	Alta	Lat: -5.174314, Long: -80.661078
06/06/2025 17:02	Embotellamiento	Media	Lat: -5.183170, Long: -80.640232
06/06/2025 16:09	Accidente de tránsito	Alta	Lat: -5.183564, Long: -80.636814
06/06/2025 13:32	Embotellamiento	Media	Lat: -5.182274, Long: -80.640402
06/06/2025 11:03	Accidente de tránsito	Alta	Lat: -5.181919, Long: -80.657273
06/06/2025 10:10	Embotellamiento	Alta	Lat: -5.189413, Long: -80.635314
06/06/2025 10:10	Embotellamiento	Alta	Lat: -5.189913, Long: -80.635515
06/06/2025 10:08	Embotellamiento	Alta	Lat: -5.193769, Long: -80.636725
06/06/2025 10:07	Embotellamiento	Alta	Lat: -5.194161, Long: -80.635474
06/06/2025 10:07	Embotellamiento	Alta	Lat: -5.194346, Long: -80.634779
06/06/2025 06:54	Embotellamiento	Alta	Lat: -5.038735, Long: -80.701409
06/06/2025 06:42	Embotellamiento	Baja	Lat: -5.170894, Long: -80.688049
06/06/2025 06:41	Embotellamiento	Baja	Lat: -5.172464, Long: -80.683863
06/06/2025 06:39	Embotellamiento	Baja	Lat: -5.175119, Long: -80.676320

Fecha	Tipo	Criticidad	Ubicación
06/06/2025 06:38	Embotellamiento	Alta	Lat: -5.176700, Long: -80.671816
06/06/2025 06:29	Embotellamiento	Alta	Lat: -5.179255, Long: -80.663310
06/06/2025 06:28	Embotellamiento	Alta	Lat: -5.175327, Long: -80.661569
06/06/2025 06:28	Embotellamiento	Alta	Lat: -5.173584, Long: -80.660697
06/06/2025 06:27	Embotellamiento	Media	Lat: -5.170041, Long: -80.658651
06/06/2025 06:25	Embotellamiento	Baja	Lat: -5.167451, Long: -80.661668
06/06/2025 06:25	Embotellamiento	Baja	Lat: -5.165372, Long: -80.664136
05/06/2025 23:13	Embotellamiento	Alta	Lat: -5.183216, Long: -80.627517
05/06/2025 23:13	Embotellamiento	Alta	Lat: -5.184588, Long: -80.627249
05/06/2025 23:12	Embotellamiento	Alta	Lat: -5.186103, Long: -80.626937
05/06/2025 22:30	Embotellamiento	Alta	Lat: -5.189243, Long: -80.627395
05/06/2025 22:29	Embotellamiento	Alta	Lat: -5.187928, Long: -80.626939
05/06/2025 22:29	Embotellamiento	Alta	Lat: -5.187781, Long: -80.627371
05/06/2025 22:29	Embotellamiento	Alta	Lat: -5.187209, Long: -80.629068
05/06/2025 22:29	Embotellamiento	Alta	Lat: -5.186927, Long: -80.629939
05/06/2025 22:28	Embotellamiento	Alta	Lat: -5.186607, Long: -80.630849
05/06/2025 22:28	Embotellamiento	Media	Lat: -5.186435, Long: -80.631399
05/06/2025 22:28	Embotellamiento	Media	Lat: -5.186256, Long: -80.631692
05/06/2025 22:27	Embotellamiento	Media	Lat: -5.182932, Long: -80.631257
05/06/2025 22:26	Embotellamiento	Media	Lat: -5.182078, Long: -80.631309
05/06/2025 19:51	Embotellamiento	Alta	Lat: -5.181389, Long: -80.631186

Fecha	Tipo	Criticidad	Ubicación
05/06/2025 19:51	Embotellamiento	Media	Lat: -5.181577, Long: -80.631161
05/06/2025 19:47	Embotellamiento	Media	Lat: -5.179423, Long: -80.628661
05/06/2025 19:47	Embotellamiento	Media	Lat: -5.176598, Long: -80.628196
05/06/2025 15:05	Embotellamiento	Baja	Lat: -5.178328, Long: -80.642386
05/06/2025 15:03	Obras en la vía	Media	Lat: -5.181973, Long: -80.639926
05/06/2025 14:59	Embotellamiento	Baja	Lat: -5.182409, Long: -80.631561
05/06/2025 14:58	Embotellamiento	Alta	Lat: -5.183163, Long: -80.631389
05/06/2025 14:56	Embotellamiento	Media	Lat: -5.186067, Long: -80.631404
05/06/2025 11:39	Embotellamiento	Baja	Lat: -4.926442, Long: -80.615732
05/06/2025 07:21	Embotellamiento	Media	Lat: -5.187204, Long: -80.628619
05/06/2025 07:21	Embotellamiento	Media	Lat: -5.187204, Long: -80.628619
05/06/2025 07:21	Embotellamiento	Media	Lat: -5.187204, Long: -80.628619
05/06/2025 07:19	Embotellamiento	Media	Lat: -5.185307, Long: -80.631349
05/06/2025 07:19	Embotellamiento	Media	Lat: -5.185307, Long: -80.631349
05/06/2025 07:18	Obras en la vía	Media	Lat: -5.184269, Long: -80.631566
05/06/2025 07:18	Obras en la vía	Media	Lat: -5.184269, Long: -80.631566
05/06/2025 07:17	Embotellamiento	Media	Lat: -5.182746, Long: -80.631890
05/06/2025 07:14	Obras en la vía	Media	Lat: -5.182237, Long: -80.639779
05/06/2025 01:08	Obras en la vía	Media	Lat: -5.173285, Long: -80.627632
05/06/2025 01:07	Embotellamiento	Media	Lat: -5.169794, Long: -80.626561
05/06/2025 01:07	Embotellamiento	Media	Lat: -5.169479, Long: -80.627186

Fecha	Tipo	Criticidad	Ubicación
05/06/2025 01:07	Accidente de tránsito	Alta	Lat: -5.167986, Long: -80.631881
05/06/2025 01:06	Obras en la vía	Alta	Lat: -5.167270, Long: -80.633037
04/06/2025 19:40	Obras en la vía	Baja	Lat: -5.194573, Long: -80.628200
04/06/2025 19:38	Embotellamiento	Alta	Lat: -5.191854, Long: -80.626969
04/06/2025 19:36	Embotellamiento	Alta	Lat: -5.189237, Long: -80.627290
04/06/2025 19:19	Embotellamiento	Media	Lat: -5.185209, Long: -80.631486
04/06/2025 19:18	Embotellamiento	Media	Lat: -5.182733, Long: -80.631353
04/06/2025 19:16	Accidente de tránsito	Baja	Lat: -5.182538, Long: -80.634612
04/06/2025 19:16	Semáforo descompuesto	Baja	Lat: -5.182337, Long: -80.636275
04/06/2025 19:16	Obras en la vía	Alta	Lat: -5.182337, Long: -80.636275
04/06/2025 18:04	Embotellamiento	Alta	Lat: -5.174811, Long: -80.628070
04/06/2025 18:03	Embotellamiento	Media	Lat: -5.174457, Long: -80.628123
04/06/2025 18:02	Embotellamiento	Alta	Lat: -5.174126, Long: -80.629075
04/06/2025 18:00	Embotellamiento	Alta	Lat: -5.173734, Long: -80.630144
04/06/2025 17:58	Obras en la vía	Alta	Lat: -5.173094, Long: -80.630099
04/06/2025 17:58	Obras en la vía	Alta	Lat: -5.172805, Long: -80.630658
04/06/2025 17:57	Obras en la vía	Alta	Lat: -5.172275, Long: -80.630476
04/06/2025 17:56	Obras en la vía	Alta	Lat: -5.171708, Long: -80.630290
04/06/2025 17:56	Obras en la vía	Alta	Lat: -5.171669, Long: -80.630138
04/06/2025 17:55	Obras en la vía	Alta	Lat: -5.171579, Long: -80.629523
04/06/2025 17:54	Obras en la vía	Alta	Lat: -5.171074, Long: -80.629198

Fecha	Tipo	Criticidad	Ubicación
04/06/2025 17:53	Obras en la vía	Alta	Lat: -5.171298, Long: -80.628668
04/06/2025 17:51	Obras en la vía	Alta	Lat: -5.172027, Long: -80.627938
04/06/2025 17:50	Obras en la vía	Alta	Lat: -5.171419, Long: -80.627689
04/06/2025 17:49	Obras en la vía	Alta	Lat: -5.170655, Long: -80.627450
04/06/2025 14:00	Embotellamiento	Media	Lat: -5.165005, Long: -80.665914
04/06/2025 11:42	Embotellamiento	Media	Lat: -5.183623, Long: -80.631310
04/06/2025 11:36	Embotellamiento	Alta	Lat: -5.184119, Long: -80.631286
04/06/2025 11:23	Embotellamiento	Media	Lat: -5.193358, Long: -80.638577
04/06/2025 10:54	Embotellamiento	Media	Lat: -5.189416, Long: -80.627118
04/06/2025 10:36	Embotellamiento	Media	Lat: -5.189764, Long: -80.627584
04/06/2025 10:33	Embotellamiento	Media	Lat: -5.191475, Long: -80.630257