

**Departamento de Ciencias de la  
Computación (DCCO)**

**Carrera de Ingeniería de Software**

**Análisis y Diseño de Software**

Perfil del Proyecto

Presentado por:

Verdesoto Segovia Edison

Benavides Macias Ruben

Pasquel Ruiz Juan

Cobeña Zambrano Joan

(Grupo 1)

Tutor académico: Ing. Jenny A Ruiz R

Ciudad: Rumiñahui

Fecha: 04 de mayo 2025

## PERFIL DE PROYECTO

<b>1. Introducción.....</b>	<b>5</b>
<b>2. Planteamiento del trabajo.....</b>	<b>5</b>
2.1 Formulación del problema.....	5
2.2 Justificación.....	5
<b>3. Sistema de Objetivos.....</b>	<b>5</b>
3.1. Objetivo General.....	5
3.2. Objetivos Específicos (03).....	5
<b>4. Alcance.....</b>	<b>5</b>
<b>5. Marco Teórico.....</b>	<b>6</b>
5.1 Metodología (Marco de trabajo 5W+2H).....	6
<b>8. Viabilidad(Ej.).....</b>	<b>6</b>
8.1.1 Tutor Empresarial.....	7
8.1.2 Tutor Académico.....	7
8.1.3 Estudiantes.....	7
8.2 Tecnológica.....	7
8.2.1 Hardware.....	7
8.2.2 Software.....	7
9.1 Conclusiones.....	8
9.2 Recomendaciones.....	8
<b>10. Planificación para el Cronograma:.....</b>	<b>9</b>
<b>11. Referencias.....</b>	<b>9</b>

# 1. Introducción

Actualmente, la gestión y el control de rutas de vehículos dentro de la Dirección Distrital 17D06 del Ministerio de Salud Pública (MSP) es realizada manualmente, lo cual afecta la eficiencia operativa y desperdicia el tiempo del personal implicado. Esta situación genera riesgos de desorganización, pérdida de información y dificultad en la toma de decisiones. En respuesta a esta problemática, el equipo de desarrollo ha propuesto la construcción del sistema CareRoutes, para digitalizar y optimizar la planificación, registro y control de rutas vehiculares. En el presente perfil de proyecto, se planea documentar el contexto, objetivos y metodología para el desarrollo de CareRoutes.

## 2. Planteamiento del trabajo

### 2.1 Formulación del problema

Una eficiente gestión del transporte institucional representa un desafío crítico para las entidades del sector de salud pública, donde la disponibilidad y el tiempo de llegada son factores decisivos al momento de responder a una emergencia. Actualmente, la empresa carece de un sistema digital que permita coordinar automáticamente la planificación de rutas, el mantenimiento vehicular, la asignación de custodios y la disponibilidad de vehículos en tiempo real.

Debido a esto, se han identificado deficiencias operativas, como la fragmentación de la información, duplicación de tareas, datos desactualizados y carga innecesaria de trabajo para personal clave. A ello se suma el desconocimiento en torno a sistemas complejos por parte de los usuarios, lo que conlleva a uso de herramientas obsoletas.

El problema se intensifica al tomar en cuenta las emergencias y la alta demanda operativa, donde la ausencia de control de los vehículos en tiempo real puede comprometer la disponibilidad del servicio y el uso eficiente de los recursos.

En este contexto, se identifica la necesidad de una solución tecnológica que reemplace la gestión manual y automatice las tareas. La solución propuesta optimiza la gestión del transporte institucional e implementa una vista en tiempo real del estado del mantenimiento de los vehículos, para una administración más eficiente y confiable de la flota de vehículos institucionales.

### 2.2 Justificación

El presente perfil permite observar el planteamiento de una solución software ante un caso de estudio de una empresa con requerimientos reales y específicos. Esto significa la ejecución de la fase de análisis y diseño de sistemas, que incluye la aplicación de metodologías, herramientas y estándares, impactando así directamente en el aprendizaje y conocimiento del lector como de los desarrolladores en el campo de la ingeniería de software.

## 3. Sistema de Objetivos

### 3.1. Objetivo General

Desarrollar prototipos funcionales de la solución de software propuesta, aplicando todas las fases del ciclo de vida del desarrollo de software, con énfasis en el análisis y diseño de sistemas, con el propósito de presentar un producto que satisfaga adecuadamente las necesidades del cliente, usuarios finales y partes interesadas.

### 3.2. Objetivos Específicos

- Proponer una solución tecnológica que automatice la gestión y el control de rutas y mantenimientos de los vehículos pertenecientes a la flota de transporte de la Dirección Distrital 17D06 del Ministerio de Salud Pública (MSP).
- Aplicar un proceso riguroso de elicitación, análisis, especificación, verificación y gestión de requisitos, utilizando técnicas de educación, herramientas CASE (Computer-Aided Software Engineering), y metodologías de validación que aseguren una comprensión universal del sistema.
- Diseñar la arquitectura del sistema utilizando técnicas y herramientas modernas de diseño de software, garantizando una solución escalable, mantenible y alineada con los objetivos del proyecto.

## 4. Alcance

- Desarrollar una plataforma que centralice y optimice la gestión de rutas y mantenimientos de los vehículos institucionales.
- Desarrollar un sistema de cálculo de kilometraje que notifique al usuario si existe un mantenimiento preventivo cercano.
- Manejar los datos de los vehículos y custodios de manera eficiente y segura.
- Implementar una herramienta que permita la visualización y exportación de reportes.

## 5. Marco Teórico

### Gestión de flotas vehiculares institucionales

La gestión de flotas abarca los sistemas, procesos y herramientas necesarios para garantizar que la flota de vehículos comerciales de una organización se mantenga durante todo su ciclo de vida. También implica la administración de los conductores de flotas, la optimización de las operaciones de la flota y la mejora de la utilización de activos. (IBM, 2023).

El mantenimiento preventivo (MP) programa intervenciones antes de la falla, basado en ciclos de kilometraje o tiempo. El dispositivo OBD(On Board Diagnostic) y su integración con el puerto OBD-II habilita la captura automática de:

- Odómetro acumulado.
- Parámetros operativos (rpm, temperatura). (OBDSOL, n.d.)

En CareRoutes se usará esta información para generar alertas cuando el kilometraje restante al próximo MP sea  $\leq 10 \%$ .

### **Arquitectura offline-first y sincronización de datos**

Una aplicación offline-first garantiza funcionalidad completa sin conexión y sincroniza los cambios cuando la red está disponible (Android Developers, 2025). Los patrones habituales incluyen:

- Persistencia local (SQLite) gestionada vía patrón Repository.
- Cola de operaciones (Outbox) para registrar transacciones pendientes.
- Sincronización por lotes empleando merge de conflictos basado en “last-write-wins” o resolución por versión (Android Developers, 2025).

SupaBase se adopta como backend-as-a-service por su compatibilidad Postgres y servicio de réplica en tiempo real, simplificando la sincronización entre la base local y la nube.

### **Herramientas y tecnologías seleccionadas**

Visual Studio Code proporciona edición ligera, terminal integrada y soporte nativo de Git, favoreciendo la colaboración (Microsoft, 2024). Flutter y su lenguaje subyacente Dart permiten mantener una única base de código para escritorio y móvil, reduciendo hasta 30 % los tiempos de desarrollo multiplataforma (Google, 2024).

SQLite es una base de datos embebida ACID, de 600 KiB de huella, apta para escenarios offline y sincronización eventual (Hipp, 2023). Combinada con SupaBase se cubren ambos extremos de persistencia local/remota.

## 5.1 Metodología (Marco de trabajo 5W+2H)

<b>What?</b> ¿Qué?	Realizar el análisis, diseño y prototipado de una aplicación multiplataforma de gestión de rutas y mantenimientos de una flota vehicular.
<b>Where?</b> ¿Dónde?	En la sección de administración de transporte de la dirección distrital 17D06 del Ministerio de Salud Pública.
<b>Why?</b> ¿Por qué?	Por la necesidad de optimizar tiempos y aumentar la fiabilidad de la planificación de rutas de la flota vehicular en la dirección distrital del MSP.
<b>Who?</b> ¿Quién?	Por parte del equipo de estudiantes y apoyo del tutor académico.
<b>When?</b> ¿Cuándo?	Desde mayo hasta agosto de 2025.
<b>How?</b> ¿Cómo?	Mediante el documento de especificación de requisitos establecido como punto de origen, la metodología ágil Scrum para gestión del desarrollo, herramientas de diseño y pruebas proporcionadas en la materia de Análisis y Diseño, y de desarrollo como VS Code, Flutter, SQLite para el prototipado.
<b>How much?</b> ¿Cuánto cuesta?	El presupuesto necesario para llevar a cabo el proyecto es de \$1651,32

Tabla 1 Marco de trabajo 5W+2H

## 6. Ideas a Defender

- **Digitalización y automatización de la gestión de rutas vehiculares:** La implementación de CareRoutes sustituirá procesos manuales ineficientes e inefectivos, reducirá errores humanos y optimizará el tiempo invertido en esta tarea.
- **Enfoque offline-first:** Al implementar un enfoque de subida de archivos local con SQLite, se asegura la operatividad del sistema incluso en entornos offline, garantizando la disponibilidad completa del sistema para la gestión de rutas.
- **Uso de sensores OBD y GPS:** La integración de datos relevantes del vehículo como su ubicación y su kilometraje se debe hacer de forma precisa,

por lo que se incorporarán a los vehículos de la flota dispositivos OBD y GPS que provean de esta información en tiempo real.

- **Multiplataforma:** Se requiere acceso a la información útil para reportería desde fuera de la oficina, por lo que se necesita un aplicativo multiplataforma para poder acceder a la misma información desde el smartphone, lo cual nos lleva a Flutter, que permite desarrollar este tipo de aplicaciones rápidamente.
- **Sincronización continua:** Para asegurar los datos y que estos coincidan tanto en la aplicación de escritorio como la móvil, y mantener el enfoque offline first se deben subir los cambios de manera periódica a un servicio de base de datos en la nube, como puede serlo SupaBase que tiene un plan gratuito de 500mb de almacenamiento.
- **Alertas y monitoreo en tiempo real:** Al tratarse de un aplicativo con información que requiere ser notificada de forma crítica como lo son los mantenimientos, el mismo notificará de manera precisa y anticipada cuando se acerque un mantenimiento de un vehículo y permitirá la visualización de indicadores clave, como el kilometraje y estado de rutas, para tomar decisiones en base a información actualizada.

## 7. Resultados Esperados

Al terminar el desarrollo del aplicativo *CareRoutes*, se espera disponer de una solución tecnológica que asigne rutas de manera óptima, tomando en cuenta la disponibilidad operativa de los vehículos. El sistema integrará eficientemente el proceso manual existente mediante la importación estructurada de datos en una base de datos SQLite, seleccionada por su ligereza y bajo consumo de recursos. Esta integración incrementará la fiabilidad de la información gestionada, y en conjunto con los datos provenientes de sensores OBD, permitirá prever con mayor exactitud los mantenimientos preventivos de las unidades. Además, gracias a un mecanismo de sincronización continua, el sistema estará disponible desde dispositivos móviles, facilitando su acceso fuera del entorno de escritorio.

## 8. Viabilidad

Cantidad	Descripción	Valor Total / Depreciación mensual (USD)	Total
	<b>Depreciación Mensual de Equipos</b>		
4 meses	Laptop LENOVO LOQ15 16GB RAM / 1TB SSD	15,98	63,92
4 meses	Laptop HP Victus R7 8845HS / 32GB RAM / 1TB SSD	15,98	63,92
4 meses	Laptop LENOVO IdeaPad 1 / 16GB RAM / 1TB SSD	7,60	30,40
4 meses	Laptop Asus TUF A15 16GB RAM / 1TB SSD	13,30	53,20
	<b>Subtotal depreciación</b>		211,44
	<b>Costo de Equipos</b>	<b>Precio unitario</b>	
12	Dispositivos OBD II	119,99	1439,88
	<b>Subtotal equipos</b>		1439,88
	<b>Costo de Software</b>	<b>Costo de uso</b>	
1	Visual Studio Code	0	0
1	SQLite	0	0
1	SupaBase (Plan gratuito)	0	0
	<b>Subtotal Software</b>		0,00
<b>Total Proyecto</b>			1651,32

Tabla 2 Presupuesto del proyecto

### 8.1 Humana

#### 8.1.1 Tutor Empresarial

Guillermo Verdesoto

#### 8.1.2 Tutor Académico

Ing. Jenny Alexandra Ruiz Robalino



### 8.1.3 Estudiantes

Benavides Macias Ruben Dario

Cobeña Zambrano Joan Oswaldo

Pasquel Ruiz Juan David

Verdesoto Segovia Edison Damián (Líder)

## 8.2 Tecnológica

### 8.2.1 Hardware

	Requisitos mínimos	Disponibilidad
Memoria RAM	6 GB de RAM	Alta
Almacenamiento	1 GB de espacio de almacenamiento	Alta

Tabla 3 Requisitos de Hardware

### 8.2.2 Software

	Requisitos mínimos	Disponibilidad
Sistema Operativo	Windows 10/11 Ubuntu Desktop 24.04.2 LTS	Alta
IDE	Es recomendable Visual Studio Code por la compatibilidad con Dart y Flutter.	Alta

Tabla 4 Requisitos de Software

## 9. Conclusiones y recomendaciones

.

## 10. Planificación para el Cronograma:

Nombre de la Tarea	Duración (h)	Comienzo	Fin	Responsables
Revisar la especificación de requisitos con la Matriz IREB	2.5	3/5/2025	3/5/2025	Joan Cobeña Edison Verdesoto Juan Pasquel Ruben Benavides
Realizar el FODA	1	6/5/2025	6/5/2025	Joan Cobeña Edison Verdesoto Juan Pasquel Ruben Benavides
Realizar de las preguntas de entrevista	1.5	6/5/2025	7/5/2025	Joan Cobeña Edison Verdesoto Juan Pasquel Ruben Benavides
Realizar Introducción Perfil de proyecto	1	4/5/2025	10/5/2025	Edison Verdesoto
Redactar el Planteamiento del problema. Perfil de proyecto	0.5	5/5/2025	10/5/2025	Joan Cobeña Edison Verdesoto
Planificar el cronograma para el perfil de proyecto	1	8/5/2025	9/5/2025	Joan Cobeña Juan Pasquel
Elaborar Justificación de Perfil proyecto	1	7/5/2025	8/5/2025	Edison Verdesoto
Definir los Objetivos Perfil proyecto	1	7/5/2025	8/5/2025	Ruben Benavides
Definir el Alcance del Perfil del proyecto	0.5	8/5/2025	9/5/2025	Edison Verdesoto
Desarrollar el Marco Teórico del Perfil proyecto	0.5	9/5/2025	11/5/2025	Juan Pasquel
Describir las Ideas a defender del Perfil proyecto	0.5	9/5/2025	10/5/2025	Joan Cobeña
Redactar los Resultados Esperados	0.5	10/5/2025	11/5/2025	Ruben Benavides
Detallar la viabilidad	0.5	10/5/2025	11/5/2025	Edison Verdesoto
Defender el Perfil del Proyecto	0.5	13/5/2025	13/5/2025	Joan Cobeña Edison Verdesoto Juan Pasquel Ruben Benavides


**Tabla 5. Cronograma del proyecto.**

## **11. Referencias**


- Android Developers. (2025, February 10). *Build an offline-first app | App architecture*. Android Developers. Retrieved May 11, 2025, from <https://developer.android.com/topic/architecture/data-layer/offline-first>
- Google. (2024). *Development*. Flutter. Retrieved May 11, 2025, from <https://flutter.dev/development>
- Hipp, D. R. (2023). *SQLite Documentation*. SQLite. Retrieved May 11, 2025, from <https://www.sqlite.org/docs.html>
- IBM. (2023, Dic 7). *¿Qué es la gestión de flotas?* Retrieved May 11, 2025, from <https://www.ibm.com/mx-es/topics/fleet-management>
- Liu, F., Lu, C., Gui, L., Zhang, Q., Tong, X., & Yuan, M. (2023, Mar 1). Heuristics for Vehicle Routing Problem: A Survey and Recent Advances. <https://arxiv.org/pdf/2303.04147>
- Microsoft. (2024). *Documentation for Visual Studio Code*. Visual Studio Code. Retrieved May 11, 2025, from <https://code.visualstudio.com/docs>
- OBDSOL. (n.d.). *What is OBD?* OBD Solutions. Retrieved May 12, 2025, from <https://www.obdsol.com/knowledgebase/on-board-diagnostics/what-is-obd/>

## Anexos.

### Anexo I. Crono

 G1\_Cronograma

### Anexo II. MTZ Historia de Usuario

 G1\_Matriz de Marco de Trabajo HU Plantilla