



**UNIVERSIDAD PRIVADA DE CIENCIAS ADMINISTRATIVAS Y
TECNOLÓGICAS**

R.M. 409/07 – D.S. 1081/11

PERFIL DE PROYECTO DE GRADO

**DESARROLLO DE UNA APLICACION MOVIL PARA LA
SOLICITUD DE CONDUCTORES DESIGNADOS EN LA CIUDAD
DE COCHABAMBA**

Presentado para optar el Grado Académico:

LICENCIADO EN ING. DE SISTEMAS

AUTOR: AMERICO JULIO ALVAREZ TELLEZ

TUTOR: ING. JOACYR FLORES ANDRADE

COCHABAMBA-BOLIVIA

| | |
|--|----|
| 2. DESCRIPCIÓN DEL PROYECTO..... | 4 |
| 3. IDENTIFICACIÓN Y PLANTEAMIENTO DEL PROBLEMA | 4 |
| 4. FORMULACIÓN DEL PROBLEMA..... | 6 |
| 5. OBJETO DE ESTUDIO..... | 6 |
| 6. Objetivo | 6 |
| b. Objetivos específicos..... | 6 |
| 7. JUSTIFICACIÓN..... | 7 |
| a. Justificación técnica..... | 7 |
| b. Justificación económica..... | 7 |
| b. Justificación económica | 7 |
| c. Justificación social..... | 8 |
| 8. ALCANCE | 8 |
| a. Alcance temático..... | 8 |
| b. Alcance geográfico | 9 |
| c. Alcance temporal | 9 |
| 9. INGENIERÍA DE REQUERIMIENTOS | 9 |
| a. Perspectiva del producto | 9 |
| b. Funcionalidad del producto | 10 |
| c. Características del usuario | 10 |
| d. Restricciones..... | 11 |
| e. Requerimientos funcionales (preliminares) | 11 |

| | |
|---|----|
| f. Requerimientos no funcionales | 12 |
| g. Requerimientos de interfaz | 12 |
| 10. PROPUESTA DE METODOLOGÍAS A UTILIZAR EN EL PROYECTO ... | 13 |
| a. Metodología o método de desarrollo | 13 |
| b. Diseño estructural arquitectónico | 13 |
| c. Base de datos | 14 |
| d. Lenguaje de programación | 14 |
| e. Marco de trabajo (Framework) | 14 |
| a. Diagrama de caso de uso nivel 0 | 14 |
| b. Diagrama de casos de uso explosionados | 16 |
| c. Diagramas de componentes | 16 |
| 12. MARCO TEÓRICO | 17 |
| 13. CRONOGRAMA DE ELABORACIÓN DEL PROYECTO | 19 |
| Fase 1: Planificación y Diseño (4 semanas) | 19 |
| Fase 2: Desarrollo Núcleo (8 semanas) | 19 |
| Fase 3: Pruebas y Ajustes (4 semanas) | 20 |
| Fase 4: Implementación (4 semanas) | 20 |
| 14. REFERENCIAS BIBLIOGRÁFICAS | 21 |

1. TÍTULO DEL TEMA DEL PROYECTO

Desarrollo de una aplicación móvil para la solicitud segura de conductores designados con monitoreo en tiempo real en la ciudad de Cochabamba.

2. DESCRIPCIÓN DEL PROYECTO

El proyecto por desarrollar tiene como finalidad permitirá a las personas solicitar un conductor designado en momentos en que no se encuentran aptas para conducir, especialmente después de eventos donde se haya consumido alcohol. La innovación del sistema radica en que el servicio no solo traslada al usuario a su domicilio, sino que también lleva su vehículo, evitando así el riesgo de dejarlo en la vía pública y exponerse a robos, sanciones o daños.

Para brindar seguridad, transparencia y confianza, la aplicación incorporará:

- Geolocalización en tiempo real con enlace compartible para familiares o contactos de confianza.
- Grabación continua del trayecto desde el móvil del conductor, con almacenamiento cifrado.
- Sistema de notificaciones automáticas, que informen al contacto de emergencia sobre el inicio, progreso y finalización del viaje.
- Verificación de conductores mediante filtros rigurosos y protocolos estandarizados.

El proyecto tendrá su fase inicial de implementación en la ciudad de Cochabamba, con proyección de expansión nacional. Se espera también incluir otros módulos complementarios como asistencia en carretera, alquiler de conductores por horas y reserva de estacionamiento seguro.

3. IDENTIFICACIÓN Y PLANTEAMIENTO DEL PROBLEMA

a. Diagrama del problema



b. Identificación del problema

En la ciudad de Cochabamba, donde las actividades sociales y festivas que involucran el consumo de bebidas alcohólicas son frecuentes, muchas personas recurren al uso de su vehículo particular para desplazarse. Sin embargo, no siempre existe un plan de acción para su regreso en condiciones de seguridad una vez que se encuentran incapacitados para conducir debido al consumo de alcohol. Esto genera una serie de problemas:

- **Riesgo de accidentes de tránsito:** El consumo de alcohol aumenta la probabilidad de accidentes al intentar conducir en estado inconveniente.

- **Exposición a robos y daños materiales:** Al no poder conducir, los usuarios dejan su vehículo en la vía pública, exponiéndolo a robos o daños.
- **Necesidad de un servicio especializado:** No existen servicios especializados que trasladen al usuario y su vehículo de manera segura, a pesar del aumento del parque automotor.

4. FORMULACIÓN DEL PROBLEMA

"¿Cómo se puede reducir el riesgo de accidentes, robos y pérdidas materiales derivados del consumo de bebidas alcohólicas y la incapacidad de conducir, mediante una solución que permita el transporte seguro de personas y vehículos?"

5. OBJETO DE ESTUDIO

El objeto de estudio es el diseño, desarrollo e implementación de una aplicación móvil funcional, que permita la gestión segura y confiable de solicitudes de conductores designados para personas no aptas para conducir. Esta solución incluirá herramientas de monitoreo en tiempo real, grabación continua, geolocalización y contacto de emergencia, orientadas a elevar el nivel de seguridad para el usuario y su entorno, con foco inicial en la ciudad de Cochabamba.

6. Objetivo

a. **Objetivo general**

Desarrollar una aplicación móvil para la solicitud segura de conductores designados, que permita trasladar al usuario junto con su vehículo en la ciudad de Cochabamba, reduciendo riesgos de accidentes, robos y daños materiales.

b. **Objetivos específicos**

- Diseñar un marco referencial que contenga el problema identificado, la metodología de recopilación de información, la justificación del proyecto y su alcance territorial y temporal.
- Elaborar un marco teórico conceptual que incluya los conceptos clave sobre seguridad vial, servicios de transporte y tecnologías móviles.
- Desarrollar los requerimientos funcionales y no funcionales para la aplicación móvil.
- Diseñar los diagramas necesarios para la arquitectura de la aplicación y el flujo de trabajo del sistema.
- Diseñar la base de datos para la gestión de usuarios, conductores, viajes y pagos dentro de la aplicación.

- Determinar los módulos principales de la aplicación, como geolocalización, verificación de conductores, grabación de trayecto y notificaciones de emergencia.

7. JUSTIFICACIÓN

a. Justificación técnica

El desarrollo de esta aplicación móvil se basa en tecnologías actuales que garantizan seguridad, escalabilidad y eficiencia en la gestión de servicios de transporte. La integración de herramientas como:

- **Geolocalización en tiempo real** (Google Maps API) para monitoreo preciso.
- **Grabación cifrada del trayecto** (almacenamiento en la nube con AWS S3 o Firebase) para garantizar transparencia.
- **Autenticación de dos factores y verificación de conductores** (OCR para documentos, bases de datos de antecedentes) para reducir riesgos.
- **Notificaciones push y enlace de seguimiento compatible** (Firebase Cloud Messaging) para contacto con familiares.

b. Justificación económica

b. Justificación económica

El desarrollo de esta aplicación móvil representa una solución económicamente viable que generará beneficios tanto para los usuarios como para los operadores del servicio. Desde el punto de vista financiero, el proyecto se justifica por:

1. **Reducción de costos para los usuarios:**
 - Elimina gastos derivados de multas por conducir bajo los efectos del alcohol.
 - Evita costos asociados a accidentes de tránsito o robos de vehículos al dejarlos abandonados.
 - Reduce el riesgo de daños materiales y pérdidas económicas por sanciones legales.
2. **Modelo de negocio sostenible y escalable:**
 - Generación de ingresos mediante tarifas competitivas por servicio.
 - Opción de suscripciones premium para clientes frecuentes (ejemplo: viajes ilimitados para empresas).

- Alianzas estratégicas con establecimientos nocturnos (bares, discotecas) que promuevan el servicio entre sus clientes.

3. Optimización de recursos en el desarrollo:

- a. Uso de tecnologías *open-source* (Flutter para el frontend, Node.js/Python para el backend) que reducen costos de licencias.
- b. Arquitectura escalable en la nube (AWS, Google Cloud) que minimiza la inversión inicial en infraestructura física.
- c. Mantenimiento y actualizaciones a bajo costo gracias al modularidad del sistema.

c. Justificación social

Este proyecto beneficiará significativamente a nuestra sociedad en varios aspectos importantes:

- **Seguridad vial:** Reduce accidentes causados por conducir en estado inconveniente.
- **Protección de bienes:** Evita pérdidas materiales al trasladar el vehículo del usuario a su domicilio.
- **Generación de empleo:** Incorpora conductores verificados, promoviendo trabajo formal en el sector transporte.
- **Concienciación:** Incluirá campañas en la app sobre riesgos del alcohol al volante.

8. ALCANCE

a. Alcance temático

Frontend:

- Desarrollo en **Flutter** (multiplataforma: iOS/Android).
- Interfaz intuitiva con módulos: solicitud de viaje, seguimiento en tiempo real, historial de viajes, perfil de usuario/conductor.

Backend:

- **Lenguaje:** Node.js (Express) o Python (Django).
- **Base de datos:** PostgreSQL (estructura relacional para usuarios, viajes, pagos).

- **APIs externas:** Google Maps (geolocalización), Firebase (notificaciones), Stripe/PayPal (pagos).

Funcionalidades clave:

- Registro y autenticación de usuarios/conductores.
- Solicitud de viaje con geolocalización y selección de conductor.
- Grabación y almacenamiento cifrado del trayecto.
- Sistema de valoración y retroalimentación.

b. Alcance geográfico

- **Fase inicial:** Ciudad de Cochabamba (zonas con alta demanda nocturna: centro, Zona Sur, Recoleta).
- **Fase de expansión:** Otras ciudades de Bolivia (La Paz, Santa Cruz, Chuquisaca, Oruro tarija) tras validación del modelo.

c. Alcance temporal

El proyecto de se desarrollará a partir del 12 de mayo hasta 12 de octubre

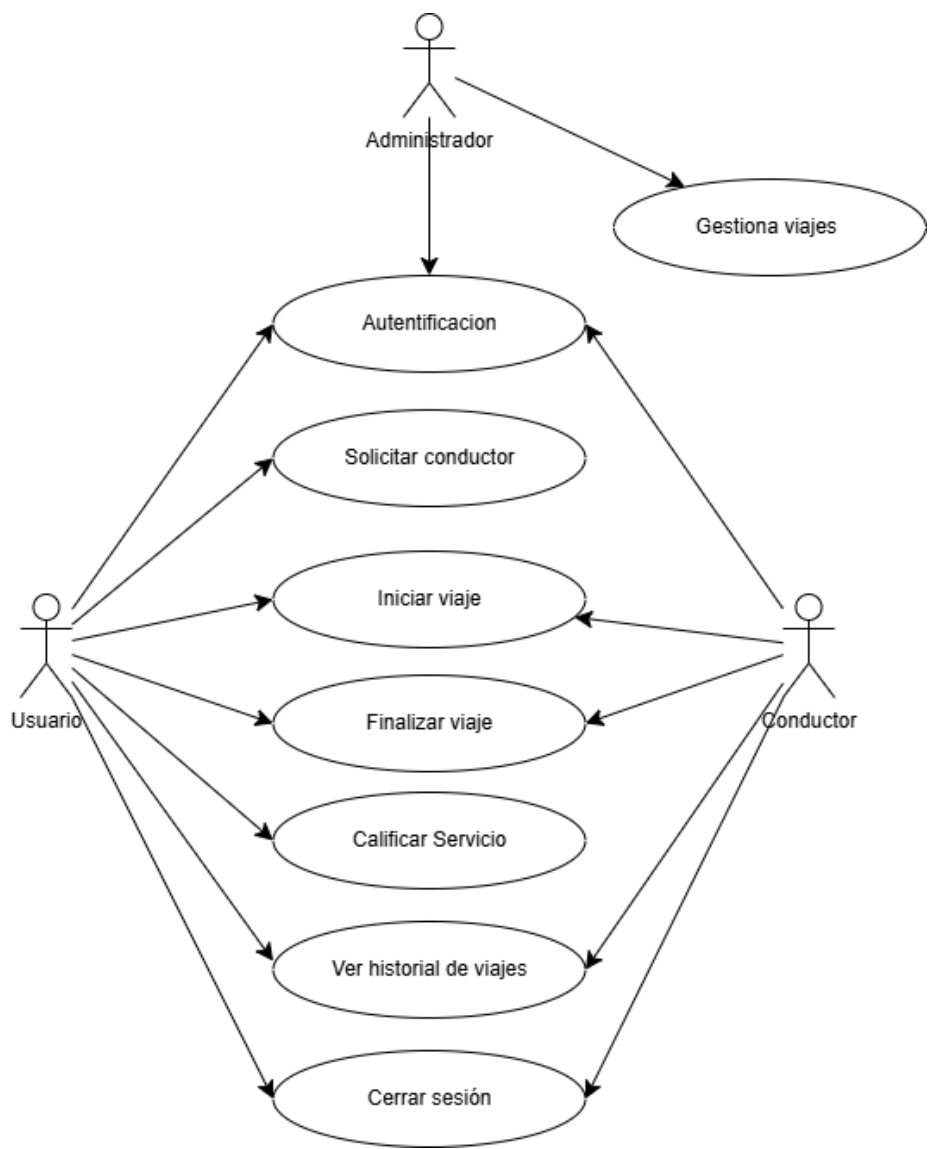
9. INGENIERÍA DE REQUERIMIENTOS

a. Perspectiva del producto

El sistema móvil permitirá:

- A **usuarios:** Solicitar conductores designados, monitorear el viaje en tiempo real, compartir ubicación con contactos de emergencia y gestionar pagos electrónicos.
- A **conductores:** Aceptar solicitudes, navegar con GPS, grabar trayectos y recibir pagos.
- Al **administrador:** Gestionar registros, verificar conductores, resolver incidencias y generar reportes de uso.

b. Funcionalidad del producto



c. Características del usuario

| Rol | Formación | Actividades |
|-----|-----------|-------------|
|-----|-----------|-------------|

| | | |
|----------------------|-------------------------------|---|
| Usuario | Sin requisitos técnicos. | Registrarse, solicitar conductor, monitorear viaje, calificar servicio. |
| Conductor | Licencia de conducir vigente. | Aceptar viajes, seguir ruta GPS, subir comprobante de entrega del vehículo. |
| Administrador | Conocimientos en TI. | Validar documentos de conductores, resolver conflictos, analizar métricas. |

d. Restricciones

- **Técnicas:**
 - La app funcionará en Android 10+ e iOS 14+ (por compatibilidad con SDKs de geolocalización).
 - Uso obligatorio de MySQL para la base de datos (por políticas de la universidad).
- **Seguridad:**
 - Grabación de viajes cifrada con AES-128 (mínimo requerido por la Ley 164 de Protección de Datos en Bolivia).

e. Requerimientos funcionales (preliminares)

| ID | Requerimiento | Descripción |
|----|---------------|-------------|
|----|---------------|-------------|

| | | |
|-------|--------------------------------|--|
| RF-01 | Registro de usuarios | El sistema debe permitir registro con email, teléfono y verificación de identidad. |
| RF-02 | Solicitud de conductor | El usuario seleccionará origen, destino y tipo de vehículo. |
| RF-03 | Geolocalización en tiempo real | Mostrará ubicación del conductor y ETA al usuario y contacto de emergencia. |
| RF-04 | Grabación de trayecto | Almacenará video cifrado durante el viaje, accesible para disputas. |

f. Requerimientos no funcionales

| Tipo | Descripción |
|----------------------|--|
| Rendimiento | La app debe responder en < segundos ante solicitudes de viaje (estándar UX). |
| Escalabilidad | Soporte para 1,000 usuarios concurrentes (usando AWS EC2). |
| Usabilidad | Interfaz intuitiva con texto legible incluso en condiciones de baja visibilidad. |

g. Requerimientos de interfaz

- **Interfaz de usuario:**
 - Diseño responsive (Figma) con botones grandes para uso en estado de emergencia.
 - Modo nocturno automático (por alta demanda en horarios nocturnos).
- **Interfaz de software:**

- o APIS externas: Google Maps (geolocalización), Firebase (notificaciones), Stripe (pagos).

10. PROPUESTA DE METODOLOGÍAS A UTILIZAR EN EL PROYECTO

a. Metodología o método de desarrollo

Se adoptará la **metodología ágil SCRUM** Esta elección se justifica por:

- **Flexibilidad:** Permite adaptar requisitos durante el desarrollo (ej: ajustar módulos de pago según feedback de usuarios piloto).
- **Entregas incrementales:** Cada *sprint* (2 semanas) generará un entregable funcional:
 - o *Sprint 1:* Prototipo de UI/UX (Figma).
 - o *Sprint 2:* Módulo de registro y autenticación.
 - o *Sprint 3:* Integración de Google Maps API.
- **Transparencia:** Reuniones diarias (*daily stand-ups*) y retrospectivas aseguran calidad.

b. Diseño estructural arquitectónico

Se usará **arquitectura MVC (Modelo-Vista-Controlador)** para:

- **Modelo:**
 - o Base de datos (MySQL) para almacenar: usuarios, conductores, viajes, pagos.
 - o Lógica de negocio (ej: cálculo de tarifas dinámicas por distancia).
- **Vista:**
 - o Interfaz móvil (Flutter) con componentes reutilizables.
- **Controlador:**
 - o Backend (Node.js) manejará solicitudes HTTP y conexiones con APIs.

Innovación vs. proyecto JC:

- Se incluirá **Firebase RealTime Database** para notificaciones push y seguimiento en tiempo real.

c. Base de datos

| Característica | Detalle |
|------------------------|---|
| Motor | PostgreSQL (compatibilidad con proyectos previos). |
| Tablas críticas | usuarios, conductores, viajes, pagos, registro_gps. |
| Seguridad | Cifrado AES-256 para datos sensibles (ej: tarjetas de crédito). |

d. Lenguaje de programación

- **Frontend:** Dart (con Flutter) para compatibilidad iOS/Android.
- **Backend:** Node.js (Express) o Python (Django Rest Framework), por:
 - Alta escalabilidad (Node.js) y rapidez en prototipado (Python).
 - Soporte para JWT (autenticación segura).

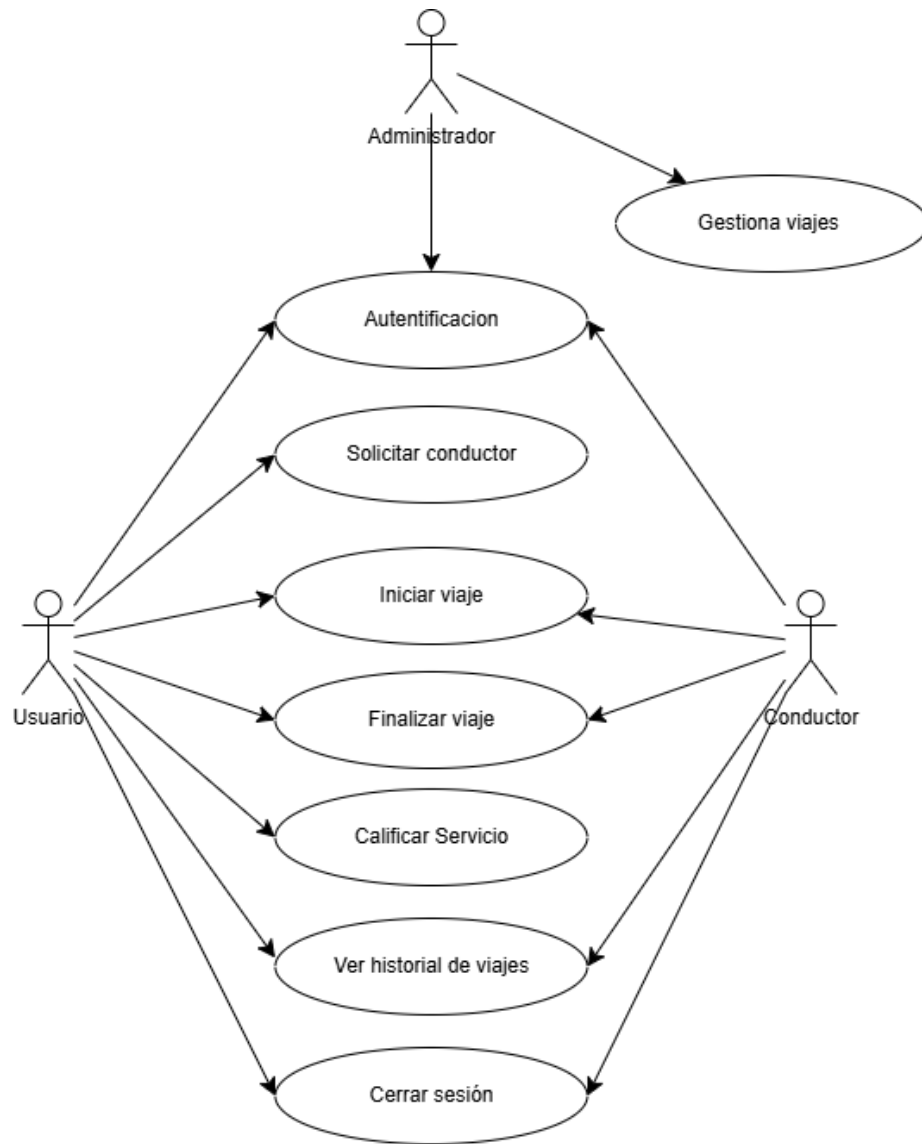
e. Marco de trabajo (Framework)

- **Frontend: Flutter** (Google).
 - Ventaja: Código único para ambas plataformas (iOS/Android), reduciendo tiempos de desarrollo.
- **Backend: Laravel** (PHP) o **Express.js** (Node.js).

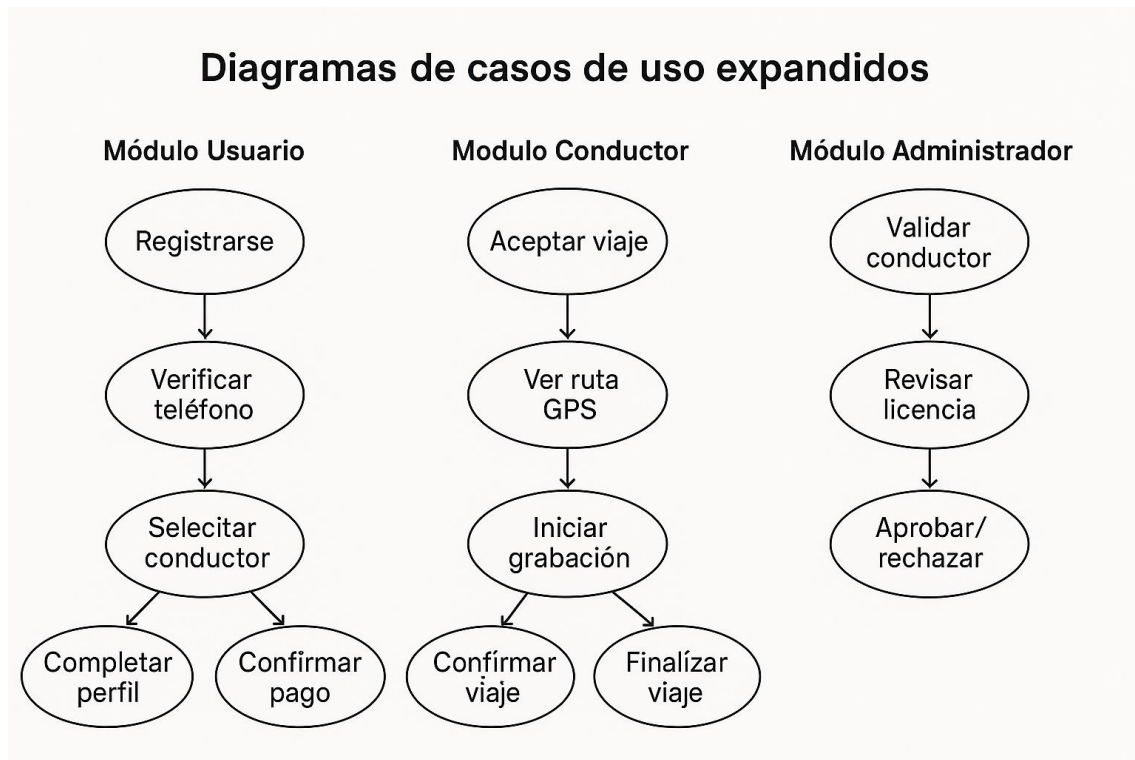
11. DISEÑO DE LA PROPUESTA

a. Diagrama de caso de uso nivel 0

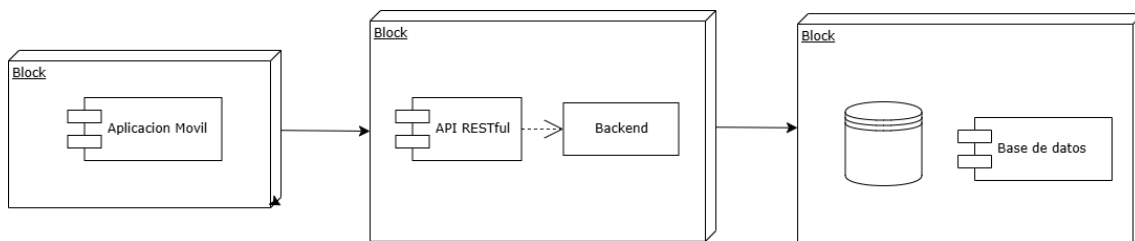
El sistema central interactúa con tres actores principales:



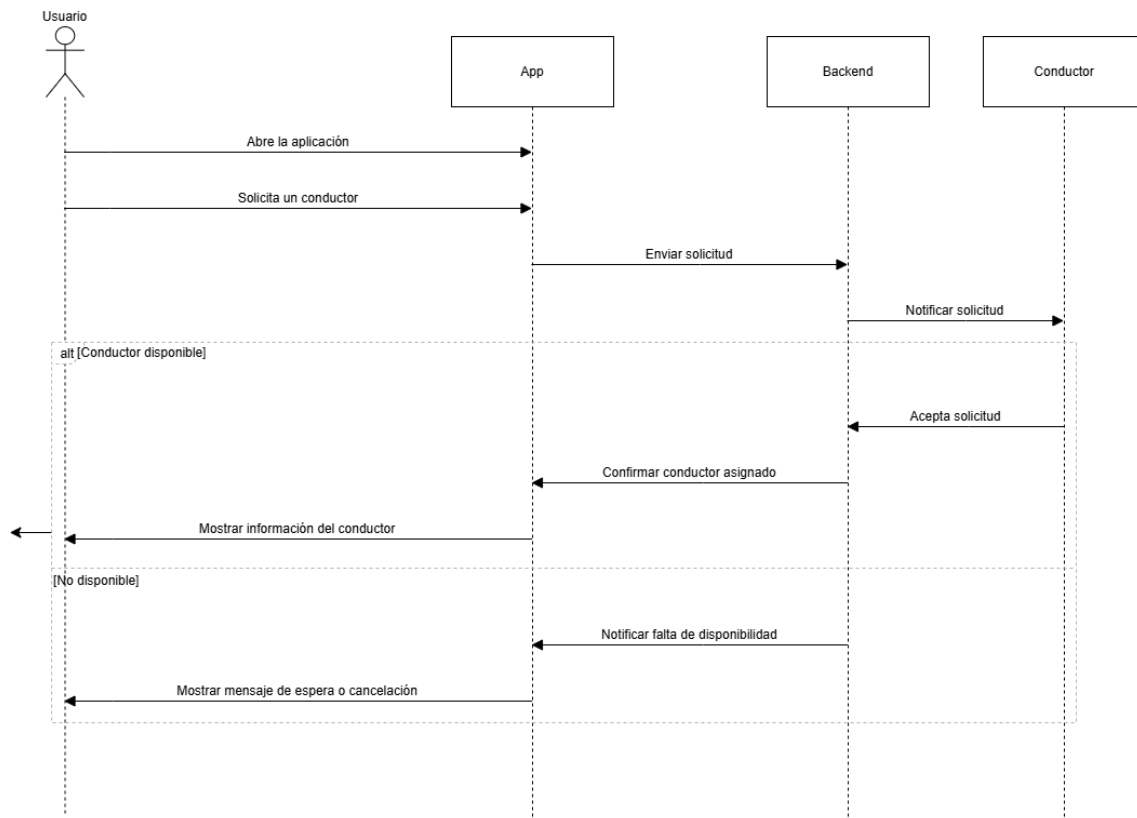
b. Diagrama de casos de uso explosionados



c. Diagramas de secuencia



c. Diagramas de componentes



12. MARCO TEÓRICO

Flutter: Flutter es un framework de código abierto desarrollado por Google para la creación de aplicaciones móviles nativas compiladas para múltiples plataformas desde una única base de código. Utiliza el lenguaje de programación Dart y permite el desarrollo de interfaces de usuario altamente personalizables mediante widgets. Flutter se ha posicionado como una de las herramientas más populares para el desarrollo móvil debido a su rendimiento similar al código nativo y su capacidad de hot reload para agilizar el proceso de desarrollo. (<https://flutter.dev/docs>)

Node.js: Node.js es un entorno de ejecución para JavaScript construido sobre el motor V8 de Chrome que permite ejecutar código JavaScript del lado del servidor. Su arquitectura basada en eventos y no bloqueante lo hace ideal para construir aplicaciones escalables que manejen múltiples conexiones simultáneas, como es el caso de sistemas de transporte en tiempo real. Node.js incluye npm, el gestor de paquetes más grande del ecosistema JavaScript. (<https://nodejs.org/docs>)

Google Maps API: La API de Google Maps es un servicio que proporciona herramientas para integrar mapas interactivos en aplicaciones web y móviles. Ofrece funcionalidades como geocodificación, cálculo de rutas y seguimiento en tiempo real con una precisión de hasta 2 metros en áreas urbanas. Esta API es

ampliamente utilizada en aplicaciones de transporte y logística debido a su confiabilidad y precisión. (<https://developers.google.com/maps/documentation>)

SCRUM: SCRUM es un marco de trabajo ágil para la gestión de proyectos de desarrollo de software. Se basa en iteraciones llamadas Sprint que generalmente duran entre 2 y 4 semanas, con roles definidos como Product Owner, Scrum Máster y Equipo de Desarrollo. SCRUM permite adaptarse rápidamente a cambios en los requisitos mediante reuniones diarias y revisiones periódicas. (<https://www.scrum.org/>)

Trello: Trello es una herramienta de gestión de proyectos basada en el método Kanban que utiliza tableros visuales, listas y tarjetas para organizar tareas. Permite la asignación de responsabilidades, establecimiento de plazos y colaboración en equipo, siendo especialmente útil para proyectos que siguen metodologías ágiles. (<https://help.trello.com/>)

Postman: Postman es una plataforma para el desarrollo de APIs que permite probar, documentar y compartir endpoints de manera eficiente. Facilita el proceso de integración entre frontend y backend mediante colecciones de peticiones HTTP y entornos configurables. (<https://learning.postman.com/>)

Firebase: Firebase es una plataforma de desarrollo de aplicaciones que ofrece diversos servicios como autenticación de usuarios, base de datos en tiempo real, hosting y notificaciones push. Es particularmente útil para aplicaciones que requieren sincronización de datos en tiempo real y funciones de mensajería. (<https://firebase.google.com/docs>)

GitHub: GitHub es una plataforma de alojamiento de código basada en Git que ofrece control de versiones distribuido y funcionalidades de colaboración. Permite gestionar repositorios, revisar código mediante pull requests y automatizar flujos de trabajo con GitHub Actions. (<https://docs.github.com/>)

Figma: Figma es una herramienta de diseño de interfaces de usuario basada en la web que permite crear prototipos interactivos y colaborar en tiempo real con equipos de diseño y desarrollo. Soporta sistemas de diseño, componentes reutilizables y plugins para extender su funcionalidad. (<https://help.figma.com/>)

AES-256: AES-256 (Advanced Encryption Standard) es un algoritmo de cifrado simétrico que utiliza una clave de 256 bits, considerado como uno de los métodos más seguros para proteger datos sensibles. Es utilizado por gobiernos y organizaciones financieras para proteger información clasificada. ([www-progress-com/](http://www.progress-com/))

13. CRONOGRAMA DE ELABORACIÓN DEL PROYECTO

Duración total estimada: 5 meses (20 semanas)

Fase 1: Planificación y Diseño (4 semanas)

| Semana | Actividades Clave | Entregables |
|--------|---|---|
| 1 | <div>- Reunión con stakeholders</div> <div>- Análisis de requisitos</div> <div>- Estudio de mercado</div> | Documento de requisitos |
| 2 | <div>- Diseño de arquitectura técnica</div> <div>- Creación de wireframes</div> | Diagramas de arquitectura Prototipos básicos |
| 3 | <div>- Modelado de base de datos</div> <div>- Diseño de API</div> | Esquema de BD Documentación API |
| 4 | <div>- Configuración de entornos de desarrollo</div> <div>- Plan de pruebas</div> | Entornos configurados Plan de QA |

Fase 2: Desarrollo Núcleo (8 semanas)

| Semana | Módulos Principales | Tecnologías Clave |
|--------|---|---------------------------|
| 5-6 | <div>- Autenticación de usuarios</div> <div>- Perfiles de conductores</div> | Firestore Auth Node.js |

| | | |
|--------------|--|--|
| 7-8 | <ul style="list-style-type: none"> - Geolocalización en tiempo real - Mapa interactivo | Google Maps API WebSockets |
| 9-10 | <ul style="list-style-type: none"> - Sistema de reservas - Gestión de viajes | Flutter MySQL |
| 11-12 | <ul style="list-style-type: none"> - Pasarela de pagos - Notificaciones push | Stripe API Firebase Cloud Messaging |

Fase 3: Pruebas y Ajustes (4 semanas)

| Semana | Tipo de Pruebas | Cobertura |
|---------------|------------------------|-------------------------|
| 13 | Pruebas unitarias | 70% del código |
| 14 | Pruebas de integración | Módulos principales |
| 15 | Pruebas de rendimiento | 1000 usuarios simulados |
| 16 | Pruebas de seguridad | OWASP Top 10 |

Fase 4: Implementación (4 semanas)

| Semana | Actividad | Responsable |
|---------------|---|--------------------|
| 17 | Despliegue en staging Pruebas UAT | Equipo DevOps |
| 18 | Corrección de bugs Optimización final | Desarrollo |
| 19 | Despliegue en producción Monitoreo inicial | DevOps |

| | | |
|-----------|--|-----------------|
| 20 | Capacitación a usuarios Lanzamiento oficial | Equipo completo |
|-----------|--|-----------------|

Hitos importantes:

1. Semana 4: Finalización de diseño aprobado
2. Semana 12: Versión beta funcional
3. Semana 16: Certificación de seguridad
4. Semana 20: Lanzamiento oficial

Notas:

- El cronograma incluye 2 semanas de buffer para imprevistos
- Reuniones semanales de seguimiento los días viernes
- Revisiones técnicas cada 2 semanas con el tutor

14. REFERENCIAS BIBLIOGRÁFICAS

Google. (2023). *Flutter Documentation*. <https://flutter.dev/docs>

Node.js Foundation. (2022). *Node.js Documentation*. <https://nodejs.org/docs>

Google Developers. (2023). *Google Maps API Guide*. <https://developers.google.com/maps/documentation>

Firebase. (2023). *Firebase Documentation*. <https://firebase.google.com/docs>

GitHub. (2023). *GitHub Guides*. <https://docs.github.com/>

Figma. (2023). *Figma Help*. <https://help.figma.com/>

Schwaber, K. (2017). *La guía de SCRUM*. Scrum.org.

Trello. (2023). *Trello Help Center*. <https://help.trello.com/>

Postman. (2023). *Postman Learning Center*. <https://learning.postman.com/>