



UNIVERSIDAD PRIVADA DE TACNA

FACULTAD DE INGENIERIA

Escuela Profesional de Ingeniería de Sistemas

**Propuesta del Proyecto RideUPT – Conecta tu
camino universitario**

Curso: Patrones de Software

Docente: Mag. Ing. Patrick Cuadros Quiroga

Integrantes:

Jorge Luis BRICEÑO DIAZ	(2017059611)
Mirian CUADROS GARCIA	(2021071083)
Brayar LOPEZ CATUNTA	(2020068946)
Ricardo DE LA CRUZ CHOQUE	(2019063329)

**Tacna – Perú
2025**

Proyecto

***RideUPT – Conecta tu camino universitario,
Tacna, 2025***

Presentado por:

Briceño Diaz, Jorge Luis

Cuadros Garcia, Mirian

Lopez Catunta, Brayar

De La Cruz Choque, Ricardo

Estudiantes del Ciclo X

03/12/2025

RESUMEN EJECUTIVO

Nombre del Proyecto propuesto:

RideUPT – Conecta tu camino universitario, Tacna, 2025

Propósito del Proyecto y Resultados esperados:

El propósito del proyecto es desarrollar e implementar una plataforma móvil de carpooling orientada a estudiantes universitarios, que permita compartir viajes de forma segura, económica y organizada, reduciendo los costos de transporte y mejorando la movilidad hacia la Universidad Privada de Tacna.

Los resultados esperados son:

- *Contar con una aplicación móvil funcional que permita registrar usuarios, crear y buscar viajes, gestionar reservas y enviar notificaciones en tiempo real.*
- *Reducir entre 60% y 70% el gasto mensual en transporte de los estudiantes que usen la plataforma.*
- *Disminuir la congestión vehicular y el tiempo de búsqueda de estacionamiento en el campus universitario.*
- *Mejorar la puntualidad y la accesibilidad al campus, fortaleciendo la comunidad estudiantil.*

Población Objetivo:

Estudiantes de la Universidad Privada de Tacna, tanto conductores como pasajeros, beneficiándose directamente en su movilidad diaria.

Monto de Inversión (En Soles):

S/ 8,480.00

Duración del Proyecto (En Meses):

3 meses

CONTROL DE VERSIONES					
Versión	Hecha por	Revisada por	Aprobada por	Fecha	Motivo
1.0	RMDC	MCG	JBD	03/12/2025	Versión Original

ÍNDICE

I.	Propuesta Narrativa	5
1.1.	Planteamiento del Problema	5
1.2.	Justificación del proyecto	5
1.3.	Objetivo general	6
1.4.	Beneficios	7
1.5.	Alcance	8
1.6.	Requerimientos del sistema	9
1.7.	Restricciones.....	11
1.8.	Supuestos	12
1.9.	Resultados esperados	13
1.10.	Metodología de implementación	14
1.11.	Actores claves.....	15
1.12.	Papel y responsabilidades del personal	16
1.13.	Plan de monitoreo y evaluación	18
1.14.	Cronograma del proyecto	19
1.15.	Hitos de entregables.....	20
II.	Presupuesto	21
2.1.	Planteamiento de aplicación del presupuesto	21
2.2.	Presupuesto	22
2.3.	Análisis de Factibilidad Económica	24
2.4.	Evaluación Financiera	24
III.	Anexo 01 – Requerimientos del Sistema (RideUPT)	25
3.1.	Requerimientos Funcionales.....	25
3.2.	Requerimientos No Funcionales.....	26
3.3.	Reglas de Negocio	26
3.4.	Modelo Conceptual (Resumen)	27
3.5.	Casos de Uso del Sistema	27
3.6.	Consideraciones Técnicas	28

I. Propuesta Narrativa

1.1. Planteamiento del Problema

Los estudiantes universitarios de Tacna, y en general del Perú, enfrentan diariamente **dificultades para movilizarse** entre sus hogares y la universidad. Estas dificultades no son solo logísticas, sino también económicas y emocionales.

En primer lugar, el **costo del transporte** representa una carga importante: muchos estudiantes gastan entre **S/. 8 y S/. 15 diarios**, lo que puede equivaler al **30–40% de sus ingresos o apoyo económico mensual**. Esto obliga a priorizar entre transporte, alimentación u otros gastos esenciales, afectando su calidad de vida y su permanencia académica.

En segundo lugar, el transporte público suele ser **lento, impredecible y congestionado**, especialmente en horas punta. Las unidades van llenas, se producen retrasos y los estudiantes llegan tarde o agotados a clases. Por otro lado, los servicios de taxi o aplicaciones comerciales tienen tarifas poco sostenibles para un estudiante promedio, por lo que no pueden considerarse una solución diaria.

En tercer lugar, la presencia creciente de vehículos particulares alrededor de la universidad genera **congestión vehicular y problemas de estacionamiento**. Conductores y pasajeros pierden tiempo buscando dónde estacionar; algunas zonas se saturan, generando incomodidad tanto para la comunidad universitaria como para los vecinos.

Finalmente, se observa la **ausencia de una plataforma formal, segura y enfocada en estudiantes** que permita compartir viajes, coordinar rutas y distribuir costos. Muchos arreglos se hacen de forma informal por chats o grupos, sin control de seguridad, sin reglas claras y sin un mecanismo que optimice rutas, horarios y cupos.

1.2. Justificación del proyecto

La implementación de RideUPT se justifica desde distintas dimensiones:

a) **Justificación económica**

Para un estudiante, el transporte diario puede representar un gasto considerable. Al compartir viajes, se logra un **ahorro estimado del 60–70%** en comparación con el uso diario de taxis o transporte independiente, traducándose en **S/. 150–200 de ahorro mensual** por estudiante que use regularmente la plataforma. Este

ahorro contribuye a que los estudiantes tengan mayor margen económico para alimentación, materiales de estudio u otros gastos académicos.

b) **Justificación social y académica**

Mejorar las condiciones de transporte impacta directamente en la **puntualidad y asistencia a clases**, reduciendo el estrés asociado a llegar tarde o no encontrar movilidad. Además, al conectar estudiantes entre sí, se fortalece la **comunidad universitaria**, se fomenta el apoyo mutuo y se crean redes de confianza.

c) **Justificación ambiental y urbana**

Al incentivar que varios estudiantes compartan un mismo vehículo en lugar de usar varios autos o múltiples unidades de transporte público, se **disminuye la cantidad de vehículos circulando** hacia el campus, contribuyendo a la reducción de **emisiones de CO₂** y al alivio de la congestión en las vías cercanas a la universidad.

d) **Justificación tecnológica**

RideUPT utiliza una arquitectura moderna basada en **Flutter (frontend móvil), Node.js (backend), MongoDB Atlas (base de datos), Firebase y Google Maps**, lo cual garantiza **escalabilidad, rendimiento** y la posibilidad de crecer hacia nuevas funcionalidades e instituciones. Esta tecnología permite implementar **geolocalización, notificaciones en tiempo real y autenticación segura** con correos institucionales.

e) **Justificación institucional**

Para la Universidad Privada de Tacna, contar con una solución de movilidad específica para sus estudiantes representa una **ventaja competitiva y reputacional**, mostrando un compromiso real con la accesibilidad, la innovación tecnológica y el bienestar de su comunidad.

En conjunto, RideUPT no solo resuelve un problema práctico de transporte, sino que **aporta valor económico, social, ambiental y académico**, con una base tecnológica sólida y alineada a las necesidades actuales.

1.3. Objetivo general

Desarrollar e implementar la plataforma de carpooling universitario RideUPT, compuesta por una aplicación móvil y un backend en la nube, que permita a los estudiantes compartir viajes de forma segura, económica y organizada, reduciendo los costos de transporte en un 60–70%, mejorando la puntualidad y fortaleciendo la comunidad universitaria en un periodo de 3 meses.

1.4. Beneficios

La puesta en marcha de RideUPT generará beneficios para **distintos grupos de interés**:

a) **Beneficios para los estudiantes pasajeros**

- Reducción significativa del **costo de transporte mensual**.
- Mayor probabilidad de llegar puntuales a clases, al viajar en rutas directas al campus.
- Mayor **seguridad percibida** al compartir viajes con otros estudiantes verificados.
- Facilidad para encontrar viajes gracias a filtros por origen, destino y horario.

b) **Beneficios para los estudiantes conductores**

- Posibilidad de **compartir los costos de combustible y mantenimiento** de su vehículo.
- Menor carga económica al utilizar el auto para más de un pasajero.
- Herramientas para gestionar sus viajes (estado, reservas, historial) de forma clara y ordenada.
- Oportunidad de integrarse y colaborar con otros compañeros de la universidad.

c) **Beneficios para la universidad**

- Mejora en la **puntualidad y asistencia** de los estudiantes.
- Disminución del **congestionamiento vehicular** y presión sobre los estacionamientos.
- Fortalecimiento de la imagen institucional como universidad que apuesta por la innovación y la sostenibilidad.
- Disponibilidad de datos agregados (respetando la privacidad) que pueden usarse para análisis de movilidad y toma de decisiones.

d) **Beneficios para la comunidad y el entorno urbano**

- Reducción de la cantidad de vehículos que se desplazan de manera individual, lo que aporta a la disminución de **emisiones contaminantes**.
- Menor congestión en las vías de acceso a la universidad y zonas aledañas.
- Promoción de hábitos de movilidad más responsables y colaborativos.

1.5. Alcance

a) Alcance funcional

El sistema RideUPT incluirá:

- **Autenticación de usuarios** mediante cuentas de Google vinculadas a correos institucionales, permitiendo validar que los usuarios sean realmente estudiantes de la universidad.
- **Gestión de perfiles de usuario**, incluyendo información personal, datos básicos de contacto y, en el caso de conductores, datos del vehículo.
- **Registro y gestión de viajes** por parte de conductores: creación de viajes con origen mediante geolocalización, selección de destino en mapa, definición de horario, asientos y cálculo automático de precio por asiento.
- **Búsqueda y filtrado de viajes** para pasajeros, con criterios como origen, destino y horario, mostrando detalles del conductor y la ruta.
- **Sistema de reservas** de asientos en los viajes publicados, incluyendo la gestión de solicitudes, aprobación/rechazo por parte del conductor y actualización automática de asientos disponibles.
- **Manejo de estados del viaje** (esperando, en-proceso, completo, expirado, cancelado) y expiración automática de viajes que no se concreten dentro del tiempo definido.
- **Historial de viajes** tanto para pasajeros como para conductores, permitiendo revisar viajes pasados y próximos.
- **Notificaciones en tiempo real** mediante Firebase Cloud Messaging y Socket.IO, para informar sobre nuevas reservas, respuestas a solicitudes y cambios de estado de los viajes.
- **Panel administrativo básico**, orientado a supervisar el uso de la plataforma, métricas generales y, si es necesario, realizar acciones sobre usuarios o viajes.

No se contempla en esta primera versión:

- Integración con **pasarelas de pago en línea** (los cobros acordados se manejan fuera del sistema).
- Integración con el sistema de transporte público o con otras aplicaciones externas.
- Módulos avanzados de gamificación, ranking o recompensas (posibles extensiones futuras).

b) Alcance geográfico e institucional

- Fase inicial: **Universidad Privada de Tacna** como institución objetivo principal.
- Fase posterior (escalable): posibilidad de adaptar la plataforma para otras universidades peruanas interesadas, manteniendo la misma arquitectura tecnológica y ajustando validaciones de correos institucionales.

1.6. Requerimientos del sistema

Los requerimientos del sistema definen **qué debe hacer RideUPT y cómo debe comportarse** para cumplir con los objetivos del proyecto. Se agrupan en **requerimientos funcionales, no funcionales y reglas de negocio** que condicionan su operación.

1. Requerimientos funcionales

Son las funcionalidades mínimas que el sistema debe ofrecer a sus usuarios (pasajeros, conductores y administradores):

- **RF001 – Autenticar usuarios**
El sistema debe permitir el **registro e inicio de sesión** de estudiantes, validando que sus credenciales correspondan a usuarios autorizados.
- **RF002 – Gestionar conductor**
El sistema debe permitir la **aceptación (habilitación)** y edición de perfiles de conductor, de modo que solo usuarios que cumplan los criterios (edad, datos mínimos, vehículo registrado) puedan ofrecer viajes.
- **RF003 – Crear viajes**
Los conductores deben poder **crear viajes** usando geolocalización automática para el origen, seleccionar el destino en el mapa e indicar horario, asientos disponibles y descripción.
- **RF004 – Buscar viajes**
Los pasajeros deben poder **buscar viajes disponibles** por origen, destino y hora, y visualizar información relevante del conductor y del viaje.
- **RF005 – Notificaciones push**
El sistema debe enviar **notificaciones en tiempo real** ante eventos importantes: nuevas solicitudes de reserva, respuestas del conductor, cambios de estado del viaje, etc.
- **RF006 – Historial de viajes**
Los usuarios deben poder acceder a un **historial de viajes pasados y próximos**, tanto como pasajeros como conductores.

- **RF007 – Expirar viajes**
Los viajes deben **expirar automáticamente** después de 10 minutos si no son tomados o no cumplen las condiciones para seguir activos.
- **RF008 – Autenticar desde Google Sign-In**
El sistema debe permitir **autenticación rápida y segura** mediante cuentas de Google, vinculadas a correos institucionales.

En la propuesta formal, puedes incluirlos en tabla como anexo o dentro del cuerpo (RF, nombre, descripción, prioridad).

2. Requerimientos no funcionales

Definen el **nivel de calidad** esperado del sistema:

- **RNF001 – Usabilidad**
La interfaz debe ser lo suficientemente **intuitiva** como para que un usuario nuevo aprenda a utilizar las funciones básicas (registrarse, buscar viaje, solicitar reserva) en menos de 3 minutos.
- **RNF002 – Rendimiento**
El tiempo de respuesta para las operaciones principales (login, búsqueda de viajes, creación de viaje) debe ser menor a **2 segundos**, en condiciones normales de red.
- **RNF003 – Disponibilidad**
El sistema debe aspirar a una **disponibilidad del 99.5% mensual**, considerando ventanas controladas de mantenimiento.
- **RNF004 – Seguridad**
La información sensible (datos personales, tokens, etc.) debe ser protegida mediante mecanismos de cifrado y el uso de **autenticación basada en tokens (JWT)**. Solo usuarios autorizados pueden acceder a sus recursos.
- **RNF005 – Escalabilidad**
La arquitectura debe permitir el **crecimiento progresivo de usuarios** y volumen de viajes sin degradar significativamente el rendimiento, aprovechando servicios en la nube y bases de datos escalables.

3. Reglas de negocio

Además de los requerimientos, RideUPT se rige por **políticas internas** que condicionan cómo se usa la plataforma:

- **RN001 – Validación estudiantil**
Solo se permiten registros e inicios de sesión de usuarios con **correo institucional válido** (por ejemplo, @virtual.upt.pe).
- **RN002 – Roles de usuario**
Un usuario puede actuar como **pasajero, conductor o ambos**, según su configuración de perfil.

- **RN003 – Precios de viajes**
El **precio por asiento** en un viaje debe estar dentro de un rango predefinido (por ejemplo, entre S/. 1.00 y S/. 3.00).
- **RN004 – Expiración de viajes**
Todo viaje debe expirar automáticamente **10 minutos después de creado**, si no se concreta o no cumple condiciones para mantenerse activo.
- **RN005 – Asientos disponibles**
El número de asientos ofrecidos en un viaje no puede exceder la **capacidad real del vehículo** (por ejemplo, un máximo de 6 asientos).
- **RN006 – Viajes simultáneos**
Un conductor solo puede tener **un viaje activo** a la vez.
- **RN007 – Reservas múltiples**
Un pasajero no puede reservar **varios asientos en el mismo viaje**, ni tener reservas que se superpongan en el mismo horario.
- **RN008 – Validación de edad**
Solo usuarios **mayores de 18 años** pueden registrarse como conductores y ofrecer viajes.

1.7. Restricciones

Las restricciones definen **límites y condiciones externas** que el proyecto debe respetar:

1. **Restricción de población objetivo**
En su fase inicial, RideUPT está restringido a estudiantes de la **Universidad Privada de Tacna** con correo institucional válido. No se contempla, en esta versión, el uso por parte de público general ni de otras instituciones sin ajustes previos.
2. **Dependencia de conexión a Internet**
Tanto la aplicación móvil como el backend requieren **conexión a Internet** para su funcionamiento. En ausencia de conectividad, no será posible buscar viajes, crear o gestionar reservas.
3. **Dependencia de servicios externos**
El sistema depende de servicios de terceros como **Firestore Auth, Firestore Cloud Messaging, Google Maps API y MongoDB Atlas**. Fallas, cambios de políticas o interrupciones en estos servicios pueden afectar temporalmente el funcionamiento de RideUPT.
4. **Limitaciones de dispositivos**
La primera implementación estará orientada principalmente a **dispositivos móviles Android** (aunque Flutter permita otras plataformas). Teléfonos con hardware muy limitado o versiones de sistema operativo muy antiguas podrían no ofrecer una experiencia óptima.

5. **Ámbito legal de transporte**

RideUPT se plantea como un sistema de **carpooling privado entre estudiantes**, no como un servicio de transporte público ni de taxi formal. El proyecto debe respetar las normas de transporte vigentes y no puede asumir funciones propias de empresas de transporte registradas.

6. **Ausencia de pasarela de pago integrada**

En esta versión, RideUPT **no integra pagos electrónicos**. Los acuerdos económicos entre conductor y pasajero se realizan fuera del sistema. Esto restringe, por diseño, funcionalidades como cobro automático, comisiones o reembolsos.

7. **Recursos de infraestructura limitados**

La solución se desplegará considerando un **presupuesto acotado**, por lo que se deben optimizar recursos de servidor, bases de datos y servicios en la nube para no exceder los costos definidos.

1.8. Supuestos

Los siguientes **supuestos** han sido considerados para el diseño y la implementación del proyecto:

1. **Acceso a smartphone por parte de los estudiantes**

Se asume que la mayoría de estudiantes cuenta con al menos un **smartphone Android** con capacidad para instalar y usar la aplicación.

2. **Disponibilidad de conexión de datos**

Se asume que los usuarios tendrán acceso frecuente a **datos móviles o WiFi**, suficiente para ejecutar operaciones básicas (login, búsqueda, reserva, notificaciones).

3. **Colaboración de la universidad**

Se asume que la Universidad Privada de Tacna respalda el uso de RideUPT dentro de su comunidad, facilitando la **difusión**, sensibilización y posibles ajustes de políticas internas relacionadas con movilidad estudiantil.

4. **Uso responsable de la plataforma**

Se asume que los estudiantes utilizarán la aplicación de forma **honesta y respetuosa**, cumpliendo los términos de uso, respetando horarios, puntos de encuentro y acuerdos de convivencia.

5. **Disponibilidad del equipo de desarrollo**

Se asume que el equipo técnico tendrá la **disponibilidad necesaria** durante los 3 meses del proyecto para atender tareas de desarrollo, pruebas, corrección de errores y ajustes derivados de la retroalimentación de usuarios.

6. **Estabilidad básica de servicios externos**

Se asume que Firebase, Google Maps y MongoDB Atlas mantendrán sus

servicios con estabilidad y cambios graduales, sin modificaciones abruptas que invaliden la arquitectura propuesta en el corto plazo.

1.9. Resultados esperados

La implementación de RideUPT espera generar **resultados concretos y medibles** en diversos aspectos:

1. Disponibilidad de una plataforma funcional

- Aplicación móvil operativa para estudiantes de la UPT.
- Backend desplegado en la nube, con base de datos configurada y servicios externos integrados.

2. Reducción de costos de transporte para estudiantes

- Ahorro promedio estimado del **60–70%** en costos de transporte para los usuarios frecuentes de la plataforma.
- Alivio en el presupuesto mensual del estudiante, permitiendo reorientar recursos a necesidades académicas.

3. Mejora en la puntualidad y asistencia

- Incremento en la **probabilidad de llegar a tiempo** a clases, al contar con viajes más directos y coordinados.
- Disminución de ausencias relacionadas con problemas de transporte.

4. Reducción del tráfico y la presión sobre estacionamientos

- Menor número de vehículos individuales llegando al campus, gracias a la práctica del carpooling.
- Reducción de la congestión en zonas cercanas a la universidad y del tiempo dedicado a buscar estacionamiento.

5. Fortalecimiento de la comunidad universitaria

- Mayor interacción y cooperación entre estudiantes de distintos ciclos y carreras.
- Generación de una cultura de **colaboración y responsabilidad compartida** en la movilidad diaria.

6. Base tecnológica escalable para futuras mejoras

- Arquitectura preparada para incorporar nuevas funciones, conectarse con otras universidades o añadir módulos como pagos, gamificación o integración con otros servicios.

1.10. Metodología de implementación

La implementación de *RideUPT* se desarrollará en un periodo de seis meses, siguiendo un enfoque iterativo e incremental con prácticas inspiradas en metodologías ágiles, adaptadas al contexto académico.

De manera general, se proponen las siguientes fases:

1. Fase de planificación y diseño (Inicio)

- Revisión consolidada de los documentos elaborados: Factibilidad, Visión, SRS y Arquitectura.
- Definición detallada del alcance de la versión inicial (MVP).
- Revisión y ajuste de requerimientos funcionales y no funcionales según prioridades.
- Refinamiento de la arquitectura propuesta (frontend en Flutter, backend en Node.js, MongoDB, Firebase y Google Maps).
- Elaboración de un plan de trabajo con hitos claros y responsables definidos.

2. Fase de construcción del backend y servicios base

- Implementación o ajuste de la API REST con Node.js y Express.
- Configuración de la base de datos en MongoDB Atlas.
- Implementación de los modelos principales: Usuario, Viaje, Reserva (*TripPassenger*) y *Rating*.
- Integración con Firebase Auth para autenticación y Firebase Cloud Messaging para notificaciones.
- Configuración de Socket.IO para funcionalidades en tiempo real (actualización de viajes y reservas).

3. Fase de construcción de la aplicación móvil

- Implementación de las pantallas principales en Flutter: login/registro, inicio, creación de viaje, búsqueda de viajes, detalle, reservas, historial y perfil.
- Integración de Google Maps para visualizar origen/destino y permitir la selección de rutas.
- Implementación de la lógica de gestión de estado (providers) y consumo de la API backend.

- Manejo de notificaciones push según los eventos definidos (nuevas reservas, respuestas del conductor, cambios de estado del viaje).

4. Fase de pruebas y validación

- Pruebas unitarias sobre componentes críticos del backend y frontend.
- Pruebas de integración entre aplicación móvil, backend y servicios externos (Firebase, Google Maps, MongoDB).
- Pruebas de usuario con un grupo piloto de estudiantes, recopilando observaciones sobre usabilidad, tiempos de respuesta y claridad de la interfaz.
- Ajustes basados en el feedback recibido, priorizando corrección de errores y mejoras en la experiencia de usuario.

5. Fase de despliegue y capacitación

- Despliegue del backend en un entorno de producción en la nube.
- Generación y publicación del archivo APK (y, de ser aplicable, subida a la Play Store).
- Elaboración de material de apoyo (guías o videos cortos) para orientar a los estudiantes en el uso de la aplicación.
- Coordinación con la universidad para la difusión interna del sistema.

6. Fase de monitoreo inicial y ajustes

- Monitoreo de métricas técnicas: disponibilidad, tiempos de respuesta y volumen de solicitudes.
- Monitoreo de métricas de uso: usuarios registrados, viajes creados y reservas realizadas.
- Identificación de mejoras prioritarias para la siguiente iteración (por ejemplo, mejoras en la interfaz, filtros avanzados o nuevas reglas de negocio).

1.11. Actores claves

Los actores claves representan a todos los participantes directos e indirectos en el desarrollo, ejecución y futuro uso de RideUPT. Su rol es esencial para que el proyecto avance de forma ordenada, coherente y alineada a los objetivos institucionales.

1. Estudiantes universitarios (Usuarios finales)

a) Pasajeros

Estudiantes que utilizan la plataforma para buscar viajes y movilizarse hacia la Universidad Privada de Tacna.

Intereses principales: reducir costos de transporte, mejorar puntualidad y contar con un medio más cómodo y seguro.

b) Conductores

Estudiantes que ofrecen asientos en su vehículo.

Intereses principales: compartir gastos de movilidad, optimizar uso del vehículo y conectar con otros estudiantes.

2. Equipo de Proyecto RideUPT

Integrado por los estudiantes responsables del análisis, diseño, desarrollo y documentación del sistema:

– Miriam Cuadros García — Líder de Proyecto / Tester QA

Asume la coordinación general del proyecto.

– Jorge Luis Briceño Díaz — Desarrollador Fullstack

Responsable de la arquitectura técnica, frontend y backend.

– Brayar López Catunta — Diseñador UI/UX

Encargado de la experiencia de usuario, diseño visual y prototipos.

– Ricardo De La Cruz Choque — Apoyo Documental y Coordinación

Apoya en redacción, evidencia, control documental y soporte administrativo del proyecto.

3. Docente asesor / Coordinador académico

Supervisa el avance, valida decisiones técnicas y metodológicas, y garantiza que el proyecto cumpla criterios académicos y de ingeniería de software.

1.12. Papel y responsabilidades del personal

1. Líder de Proyecto – Tester/QA (Miriam Cuadros García)

Responsabilidades:

- Planificar, coordinar y supervisar el avance general del proyecto.
- Gestionar tareas, control de cronograma y cumplimiento de objetivos.
- Asegurar la comunicación continua entre los miembros del equipo.

- Dirigir el proceso de pruebas funcionales, de integración y usabilidad.
- Documentar resultados de pruebas, incidencias y propuestas de mejora.
- Apoyar en la toma de decisiones técnicas junto al desarrollador principal.

2. Desarrollador Fullstack (Jorge Luis Briceño Díaz)

Responsabilidades:

- Implementar la aplicación móvil en Flutter (pantallas, lógica, providers).
- Desarrollar la API REST en Node.js y Express.
- Modelar y gestionar la base de datos en MongoDB Atlas.
- Integrar servicios externos (Google Maps, Firebase Auth, FCM).
- Implementar sockets y notificaciones en tiempo real.
- Optimizar rendimiento y seguridad del sistema.
- Realizar despliegues, configuración de servidores y versiones finales.

3. Diseñador UI/UX (Brayar López Catunta)

Responsabilidades:

- Diseñar prototipos de interfaz enfocados en usabilidad.
- Elaborar manual de estilos visuales y lineamientos de diseño.
- Crear flujos de navegación intuitivos para pasajeros y conductores.
- Trabajar junto al desarrollador para adaptar los diseños a Flutter.
- Participar en pruebas de usabilidad y mejoras visuales.

4. Responsable de Documentación del Proyecto (Ricardo De La Cruz Choque)

Responsabilidades:

- Redactar, ordenar y consolidar todos los documentos formales (FD01, FD02, FD03, FD04, FD06, etc.).
- Mantener control de versiones y validaciones con el docente asesor.
- Apoyar en el análisis de información, preparación de anexos y evidencias.
- Elaborar informes parciales, presentaciones y reportes finales.

1.13. Plan de monitoreo y evaluación

El monitoreo del proyecto RideUPT se realizará durante y después de su desarrollo para asegurar calidad técnica, alineamiento a objetivos y funcionamiento estable.

1. Objetivos del monitoreo

- Comprobar que las funcionalidades implementadas cumplen con los requerimientos establecidos.
- Evaluar desempeño técnico: velocidad, estabilidad y disponibilidad.
- Medir la satisfacción y experiencia de uso de los estudiantes piloto.
- Identificar oportunidades de mejora en interfaz, reglas de negocio y rendimiento.

2. Indicadores de evaluación

a) Indicadores técnicos

- Tiempo promedio de respuesta en las funciones clave (<2 segundos).
- Número de errores críticos detectados en el backend.
- Disponibilidad mensual del sistema (meta: 99.5%).

b) Indicadores de uso

- Cantidad de viajes creados por semana.
- Número de estudiantes que utilizan la app como pasajeros y conductores.
- Volumen de solicitudes de reserva procesadas.

c) Indicadores de calidad percibida

- Encuestas de satisfacción sobre facilidad de uso.
- Niveles de aceptación del diseño UI/UX.
- Retroalimentación sobre seguridad y confianza.

3. Mecanismos de monitoreo

- Revisiones semanales del avance del proyecto.
- Pruebas piloto internas con estudiantes voluntarios.
- Análisis de logs y métricas del backend.

- Reuniones con el docente asesor para evaluar progreso.

4. Evaluación final

Al término del proyecto (mes 3):

- Se elaborará un informe final de evaluación (FD05) y se procederá a la subida de todo el proyecto, junto con su documentación técnica y administrativa, a un repositorio de GitHub para su conservación, consulta y verificación.

1.14. Cronograma del proyecto

Mes 1 – Análisis, diseño y planificación

- Revisión de documentos preliminares (Factibilidad, Visión, SRS, SAD).
- Diseño de arquitectura técnica y estructura de módulos.
- Diseño UI/UX de las pantallas principales.
- Planificación de tareas en un backlog priorizado.
- Configuración de repositorios y entornos de desarrollo.

Mes 2 – Desarrollo de funcionalidades principales

Frontend (App Flutter)

- Login y autenticación con Google.
- Pantalla principal y lista de viajes disponibles.
- Módulo de creación de viajes con geolocalización.

Backend (API + Base de datos)

- Implementación de modelos: Usuario, Viaje, Reserva, Rating.
- Endpoints principales: login, crear viaje, buscar viaje, gestionar usuario.
- Configuración de notificaciones push y sockets.

Mes 3 – Integración, pruebas y despliegue

- Integración completa entre app y backend.
- Implementación de reservas, aprobaciones y rechazo.
- Manejo de estados de viaje y expiración automática.

- Ejecución de pruebas funcionales y piloto.
- Ajustes finales, optimización y documentación.
- Despliegue del backend y entrega del APK.

1.15. Hitos de entregables

Hito 1 — Fin del Mes 1:

Documento de Propuesta aprobado

- Arquitectura confirmada.
- Prototipo UI/UX listo.
- Requerimientos completos y validados.

Hito 2 — Fin del Mes 2:

Backend + App con flujos principales

- Creación de viajes funcional.
- Login operativo.
- Listado de viajes listo.
- Base de datos configurada.

Hito 3 — Fin del Mes 3:

Sistema funcional y desplegado (Versión Inicial)

- Reservas y notificaciones en tiempo real funcionando.
- Estados de viaje completos.
- Pruebas piloto realizadas.
- Documentación completa entregada.
- APK listo para distribuir a estudiantes.

II. Presupuesto

2.1. Planteamiento de aplicación del presupuesto

El presupuesto asignado al desarrollo del sistema RideUPT está orientado a cubrir los costos necesarios para la **implementación completa del proyecto**, incluyendo su análisis, diseño, desarrollo, pruebas, documentación y despliegue.

La inversión contempla:

a) Costos de Personal

Son los costos principales del proyecto, dado que la solución es 100% de desarrollo propio. Incluyen:

- Liderazgo del proyecto y control de calidad (QA).
- Desarrollo frontend y backend.
- Diseño UI/UX.
- Documentación técnica, evidencia y soporte administrativo.

Estos costos cubren **horas de trabajo efectivamente dedicadas** a la construcción del sistema, organizadas según responsabilidades específicas.

b) Costos de Infraestructura Tecnológica

Incluyen:

- Servicios en la nube (MongoDB Atlas).
- Consumo de APIs de Google Maps (uso limitado dentro de capa gratuita y optimizada).
- Servicios de Firebase (Auth, Cloud Messaging).
- Servidores para despliegue backend (cloud hosting o VPS).

La infraestructura seleccionada se ajusta al presupuesto y está alineada con la arquitectura definida en el SAD, priorizando servicios gratuitos, escalables y de bajo costo para entornos académicos.

c) Costos Operativos y Administrativos

Comprenden:

- Elaboración de documentación técnica y del proyecto.
- Revisión, impresión o presentación de informes (si aplica).
- Materiales menores de apoyo (cuadernos, insumos).

d) Contingencias

Se asigna un pequeño margen para cubrir imprevistos derivados de pruebas, reintentos de despliegue, aumento de uso en APIs o ajustes técnicos necesarios durante la implementación.

En conjunto, el presupuesto asegura una ejecución ordenada del proyecto y la disponibilidad de recursos para alcanzar un producto funcional en 3 meses.

2.2. Presupuesto

A continuación, se detalla el presupuesto completo, tomando como referencia el cuadro oficial del Informe de Factibilidad y los costos asociados a infraestructura y operación.

A. Costos de Personal

Rol	Descripción	Cantidad	Horas Totales	Tarifa por Hora (S/.)	Costo Total (S/.)
Líder de Proyecto – tester/QA	Gestión, coordinación, seguimiento	1	120	25	S/ 3,000.00
Desarrollador Fullstack	Desarrollo frontend y backend	1	160	18	S/ 2,880.00
Diseñador UI/UX	Diseño de interfaces y experiencia	1	120	15	S/ 1,800.00
Subtotal Costos de Personal	—	—	—	—	S/ 7,680.00

Este subtotal proviene del análisis de factibilidad del FD01 y refleja el esfuerzo invertido por el equipo RideUPT.

B. Costos de Infraestructura Tecnológica

Concepto	Costo Estimado (S/.)	Observaciones
MongoDB Atlas (plan básico)	S/ 120.00	Uso mensual, 3 meses (S/ 40 cada mes)
Google Maps API	S/ 0.00	Uso dentro del crédito gratuito de \$200 al mes
Firebase Auth + FCM	S/ 0.00	Servicios gratuitos en modo estándar

Hosting backend (servidor cloud/VPS)	S/ 150.00	Plan básico económico por 3 meses
Dominio (opcional)	S/ 45.00	Dominio educativo o estándar
Subtotal Infraestructura	S/ 315.00	—

El uso de plataformas escalables y gratuitas en gran medida permite mantener los costos bajos sin sacrificar funcionalidades ni robustez.

C. Costos Operativos y Administrativos

Concepto	Costo Estimado (S/.)	Observaciones
Material de oficina / insumos	S/ 50.00	Cuadernos, copias, marcadores
Impresión y edición de documentos	S/ 35.00	Informes FD01–FD06
Herramientas digitales	S/ 0.00	Uso de herramientas gratuitas
Subtotal Operativo	S/ 85.00	—

D. Contingencias (5%)

Concepto	Costo Estimado (S/.)	Observaciones
Fondo de contingencia	S/ 400.00	Para imprevistos menores
Subtotal Contingencia	S/ 400.00	—

Resumen General del Presupuesto

Categoría	Monto (S/.)
Costos de Personal	S/ 7,680.00
Infraestructura Tecnológica	S/ 315.00
Operativos y Administrativos	S/ 85.00
Contingencias	S/ 400.00
Total del Proyecto	S/ 8,480.00

2.3. Análisis de Factibilidad Económica

Según el Informe de Factibilidad (FD01), RideUPT demuestra una viabilidad económica sólida basada en:

- Ahorros significativos para estudiantes (S/ 150–200 mensuales por usuario frecuente).
- Baja inversión inicial comparada con soluciones comerciales.
- Uso de infraestructura en la nube optimizada y de bajo costo.
- Posibilidad de escalar a otras universidades sin duplicar costos de desarrollo.

Resultados del análisis financiero:

- **VAN (Valor Actual Neto): S/ 95,114**
- **ROI (Retorno de Inversión):** Recuperación estimada en **11 meses**
- **Reducción promedio de costos estudiantiles: 60–70%**

Estos indicadores muestran que el proyecto no solo es viable, sino altamente conveniente para la institución y la comunidad estudiantil, dado que el beneficio económico mensual para los usuarios supera ampliamente el costo total del sistema.

2.4. Evaluación Financiera

La evaluación financiera confirma la sostenibilidad del proyecto, considerando:

a) Beneficios Cuantificables

- Si 100 estudiantes utilizan RideUPT regularmente:
 - Ahorro mensual total $\approx 100 \times \text{S/ } 150 = \text{S/ } 15,000$
 - En 12 meses $\rightarrow \text{S/ } 180,000$ ahorrados colectivamente.

b) Relación beneficio/costo

$$\frac{\text{Beneficio anual}}{\text{Costo del sistema}} = \frac{180,000}{8,480} \approx 21.2$$

Por cada sol invertido en RideUPT, la comunidad obtiene **S/ 21.20** en beneficios.

c) Interpretación

- Alta rentabilidad social y económica.
- Rápida recuperación de la inversión inicial.
- Impacto directo en la economía de los estudiantes.

- Costos de mantenimiento extremadamente bajos.

III. Anexo 01 – Requerimientos del Sistema (RideUPT)

3.1. Requerimientos Funcionales

ID	Nombre	Descripción	Prioridad
RF001	Autenticar Usuarios	El sistema debe permitir el registro e inicio de sesión de usuarios con credenciales válidas de estudiantes.	Alta
RF002	Gestionar Conductor	El sistema debe permitir habilitar y editar perfiles de conductor, asegurando que cumplan con los criterios establecidos.	Alta
RF003	Crear Viajes	Los conductores deben poder crear viajes usando geolocalización automática para el origen y seleccionando destino, horario y asientos.	Alta
RF004	Buscar Viajes	Los pasajeros deben poder buscar viajes filtrando por origen, destino y hora.	Alta
RF005	Notificaciones Push	El sistema debe enviar notificaciones en tiempo real por eventos como solicitudes, aprobaciones y cambios de estado.	Media
RF006	Historial de Viajes	Los usuarios deben tener acceso a un historial de viajes pasados y próximos.	Media
RF007	Expirar Viajes	Los viajes deben expirar automáticamente 10 minutos después de creados si no se concretan.	Media
RF008	Autenticar vía Google Sign-In	El sistema debe permitir autenticación rápida y segura mediante cuentas de Google.	Alta

3.2. Requerimientos No Funcionales

ID	Nombre	Descripción	Prioridad
RNF001	Usabilidad	La interfaz debe ser intuitiva y permitir que un usuario nuevo aprenda a usarla en menos de 3 minutos.	Alta
RNF002	Rendimiento	El tiempo de respuesta para operaciones principales debe ser menor a 2 segundos.	Alta
RNF003	Disponibilidad	El sistema debe aspirar a un uptime del 99.5% mensual.	Alta
RNF004	Seguridad	La información sensible debe ser protegida mediante mecanismos de cifrado y JWT.	Alta
RNF005	Escalabilidad	La arquitectura debe permitir crecimiento sin degradación relevante del rendimiento.	Media

3.3. Reglas de Negocio

ID	Regla	Descripción
RN001	Validación Estudiantil	Solo se permiten registros e inicios de sesión con correo institucional (ej. @virtual.upt.pe).
RN002	Roles de Usuario	Un usuario puede actuar como pasajero, conductor o ambos.
RN003	Precios de Viajes	El precio por asiento debe estar en el rango S/. 1.00 – S/. 3.00.
RN004	Expiración de Viajes	Todo viaje debe expirar automáticamente 10 minutos después de creado.
RN005	Asientos Disponibles	Un conductor no puede publicar más asientos que la capacidad de su vehículo (máximo 6).
RN006	Viajes Simultáneos	Un conductor solo puede tener un viaje activo a la vez.
RN007	Reservas Múltiples	Un pasajero no puede reservar múltiples asientos en un mismo viaje.
RN008	Validación de Edad	Solo mayores de 18 años pueden ofrecer viajes como conductores.

3.4. Modelo Conceptual (Resumen)

Las principales entidades del sistema RideUPT son:

Usuario

Contiene datos personales y académicos del estudiante, incluyendo rol, correo institucional, edad, foto de perfil y vehículo (solo para conductores).

Vehículo

Asociado opcionalmente a un usuario conductor, incluye marca, modelo, año, placas y capacidad de asientos.

Viaje

Es creado por un conductor e incluye origen, destino, disponibilidad de asientos, precio por asiento, horario, estado y pasajeros.

Reserva (TripPassenger)

Representa la solicitud de un pasajero para un viaje, con estado: *pending*, *confirmed* o *rejected*.

Notificación

Registro de eventos enviados a conductores o pasajeros.

Rating

Permite dejar calificaciones entre usuarios luego de un viaje.

3.5. Casos de Uso del Sistema

UC001 – Registrar Usuario

El estudiante se registra usando su correo institucional y Google Sign-In. El sistema valida dominio, crea cuenta y habilita acceso.

UC002 – Iniciar Sesión

El usuario ingresa con Google Sign-In o credenciales institucionales. El sistema valida y crea sesión JWT.

UC003 – Crear Viaje

El conductor define origen (GPS), destino (mapas), horario y asientos. El sistema calcula precio, crea el viaje y envía notificaciones.

UC004 – Buscar Viajes

El pasajero explora viajes disponibles y puede filtrarlos. El sistema devuelve resultados con información del conductor y la ruta.

UC005 – Solicitar Reserva

El pasajero selecciona viaje y solicita un asiento. El sistema envía solicitud al conductor y cambia el estado de la reserva.

UC006 – Gestionar Reservas

El conductor revisa solicitudes, aprueba o rechaza, y el sistema notifica al pasajero.

3.6. Consideraciones Técnicas

Arquitectura General

- **Frontend:** Flutter (Android principalmente)
- **Backend:** Node.js con Express
- **BD:** MongoDB Atlas
- **Servicios externos:** Firebase Auth, FCM, Google Maps API
- **Comunicación en tiempo real:** Socket.IO
- **Autenticación:** JWT + Firebase

Disponibilidad

Se requiere infraestructura en la nube con disponibilidad mínima del 99.5%.

Seguridad

- Cifrado de contraseñas (hash)
- Tokens JWT
- Control de acceso por roles
- Validación estricta de dominios institucionales