

# Informe de Trabajo Final

---

Universidad Peruana de Ciencias Aplicadas



Ingeniería de Software

Aplicaciones para Dispositivos Móviles - 14650

**Docente:** Eduardo Martin Reyes Rodriguez

**Startup:** TinkuyTech

**Producto:** Ñango

**Team members:**

| Nombre                            | Código     |
|-----------------------------------|------------|
| Gamarra Vega, Anderson Jose       | u202016154 |
| Castañeda Llanos, Kevin Alexander | U202318814 |
| Peña Riofrio, Maria Fernanda      | u202113279 |
| Grandez Mansilla, Jean Pierre     | u202212484 |
| Guevara Tejada, Jorge Enrique     | U202318814 |

**Ciclo 2025-02**

## Registro de versiones del informe

---

| Versión | Fecha      | Autor      | Descripción de Modificación   |
|---------|------------|------------|---|
| TB1     | 06/09/2025 | TinkuyTech | Redacción de los capítulos:<br>Capítulo I: Introducción<br>Capítulo II: Requirements & Analysis |

# Index

---

## Capítulo I: Introducción

- 1.1. Startup Profile
  - 1.1.1. Descripción de la Startup
  - 1.1.2. Perfiles de integrantes del equipo
- 1.2. Solution Profile
  - 1.2.1 Antecedentes y problemática
  - 1.2.2 Lean UX Process
    - 1.2.2.1. Lean UX Problem Statements
    - 1.2.2.2. Lean UX Assumptions
    - 1.2.2.3. Lean UX Hypothesis Statements
    - 1.2.2.4. Lean UX Canvas
- 1.3. Segmentos objetivo

## Capítulo II: Requirements Development and Software Solution Design

- 2.1. Competidores
  - 2.1.1. Análisis competitivo
  - 2.1.2. Estrategias y tácticas frente a competidores
- 2.2. Entrevistas
  - 2.2.1. Diseño de entrevistas
  - 2.2.2. Registro de entrevistas
  - 2.2.3. Análisis de entrevistas
- 2.3. Needfinding
  - 2.3.1. User Personas
  - 2.3.2. User Task Matrix
  - 2.3.3. User Journey Mapping
  - 2.3.4. Empathy Mapping
  - 2.3.5. Ubiquitous Language
- 2.4. Requirements specification
  - 2.4.1. User Stories
  - 2.4.2. Impact Mapping
  - 2.4.3. Product Backlog

- 2.5. Strategic-Level Domain-Driven Design
  - 2.5.1. EventStorming
    - 2.5.1.1. Candidate Context Discovery
    - 2.5.1.2. Domain Message Flows Modeling
    - 2.5.1.3. Bounded Context Canvases
  - 2.5.2. Context Mapping
  - 2.5.3. Software Architecture
    - 2.5.3.1. Software Architecture Context Level Diagrams
    - 2.5.3.2. Software Architecture Container Level Diagrams
    - 2.5.3.3. Software Architecture Deployment Diagrams
- 2.6. Tactical-Level Domain-Driven Design
  - 2.6.x. Bounded Context:
    - 2.6.x.1. Domain Layer
    - 2.6.x.2. Interface Layer
    - 2.6.x.3. Application Layer
    - 2.6.x.4. Infrastructure Layer
    - 2.6.x.5. Bounded Context Software Architecture Component Level Diagrams
    - 2.6.x.6. Bounded Context Software Architecture Code Level Diagrams
      - 2.6.x.6.1. Bounded Context Domain Layer Class Diagrams
      - 2.6.x.6.2. Bounded Context Database Design Diagram

## Capítulo III: Solution UI/UX Design

- 3.1. Product design
  - 3.1.1. Style Guidelines
    - 3.1.1.1. General Style Guidelines
  - 3.1.2. Information Architecture
    - 3.1.2.1. Organization Systems
    - 3.1.2.2. Labelling Systems
    - 3.1.2.3. SEO Tags and Meta Tags
    - 3.1.2.4. Searching Systems
    - 3.1.2.5. Navigation Systems
  - 3.1.3. Landing Page UI Design
    - 3.1.3.1. Landing Page Wireframe
    - 3.1.3.2. Landing Page Mock-up
  - 3.1.4. Mobile Applications UX/UI Design
    - 3.1.4.1. Mobile Applications Wireframes
    - 3.1.4.2. Mobile Applications Wireflow Diagrams
    - 3.1.4.3. Mobile Applications Mock-ups
    - 3.1.4.4. Mobile Applications User Flow Diagrams
    - 3.1.4.5. Mobile Applications Prototyping<sup>a</sup>

## Capítulo IV: Product Implementation & Validation

- 4. Product Implementation & Validation

- 4.1. Software Configuration Management
  - 4.1.1. Software Development Environment Configuration
  - 4.1.2. Source Code Management
  - 4.1.3. Source Code Style Guide & Conventions
  - 4.1.4. Software Deployment Configuration
- 4.2. Landing Page & Mobile Application Implementation
  - 4.2.1. Sprint n
    - 4.2.1.1. Sprint Planning n
    - 4.2.1.2. Sprint Backlog n
    - 4.2.1.3. Development Evidence for Sprint Review
    - 4.2.1.4. Testing Suite Evidence for Sprint Review
    - 4.2.1.5. Execution Evidence for Sprint Review
    - 4.2.1.6. Services Documentation Evidence for Sprint Review
    - 4.2.1.7. Software Deployment Evidence for Sprint Review
    - 4.2.1.8. Team Collaboration Insights during Sprint
- 4.3. Validation Interviews
  - 4.3.1. Diseño de Entrevistas
  - 4.3.2. Registro de Entrevistas
  - 4.3.3. Evaluaciones según heurísticas

# Capítulo I:

---

## 1.1 Startup Profile

### 1.1.1. Descripción de la Startup

TinkuyTech es una startup creada por unos estudiantes de la Facultad de Ingeniería de la Universidad Peruana de Ciencias Aplicadas (UPC) con el propósito de responder a una necesidad de la comunidad de estudiantes universitarios: un transporte seguro, accesible y confiable. Observando las dificultades que enfrentan los estudiantes para trasladarse diariamente, desde altos costos en taxis y aplicaciones de movilidad hasta la inseguridad en el transporte informal, ÑanGo propone una solución innovadora que conecta a estudiantes con movilidad propia (choferes) con compañeros que requieren transporte (pasajeros).

Por medio de nuestra plataforma digital, se busca no solo reducir los gastos individuales de transporte, sino también fomentar la confianza, la eficiencia en el desplazamiento y la creación de una comunidad universitaria solidaria. ÑanGo está comprometido con la seguridad, la innovación tecnológica y el impacto social, ofreciendo un modelo colaborativo que mejore la calidad de vida de los estudiantes.

**Misión:** Nuestra Misión es facilitar el transporte universitario mediante una aplicación accesible y colaborativa que permita a los estudiantes compartir rutas de manera segura, eficiente y económica, priorizando la confianza y la comodidad en cada viaje.

**Visión:** Nuestra visión es aspirar a ser la plataforma líder en movilidad universitaria en el Perú, reconocida por nuestro compromiso con la seguridad, la sostenibilidad y la experiencia del usuario, contribuyendo a transformar la forma en que los estudiantes se movilizan y fortaleciendo una cultura de transporte compartido responsable.

## 1.1.2. Perfiles de integrantes del equipo

- **Anderson Jose Gamarra Vega:**

### **Descripción:**

Mi nombre es Anderson Jose Gamarra Vega, tengo 25 años , estudiante de Ingeniería de Software. Desde siempre he sentido una gran pasión por la tecnología, pero me decanto especialmente por el desarrollo de software, pues me fascina aprender nuevos lenguajes de programación, diseñar soluciones digitales y afrontar retos mediante código. Esta inclinación hacia el software fue lo que me motivó a elegir esta carrera, y actualmente estoy profundizando en áreas como backend, arquitectura de software, metodologías agile.



- Kevin Alexander Castañeda Llanos:

**Descripción:**

Mi nombre es Kevin Alexander Castañeda Llanos, tengo 19 años y soy estudiante de la carrera de Ingeniería de Software. Siempre me gusto la tecnología y sentí curiosidad por aprender en como funcionaba la creación de aplicaciones o paginas web. Esta fue mi principal inspiración para estudiar algo relacionado al TI. Además, me gustó la idea de desarrollar proyectos y diseñar soluciones que ayuden a solucionar problemas de los demás con ayuda de las personas.





- **Maria Fernanda Peña Riofrio:**

**Descripción:**

Mi nombre es Maria Fernanda Peña Riofrio, tengo 21 años y soy estudiante de la carrera de Ingeniería de Software cursando el 7mo ciclo. Elegí la carrera por que siempre me ha gustado descubrir nuevas tecnologías y como se creaban y configuraban. En un futuro me gustaría dedicarme a la creación de desarrollo web y mobile que solucionen problemas de la vida real.

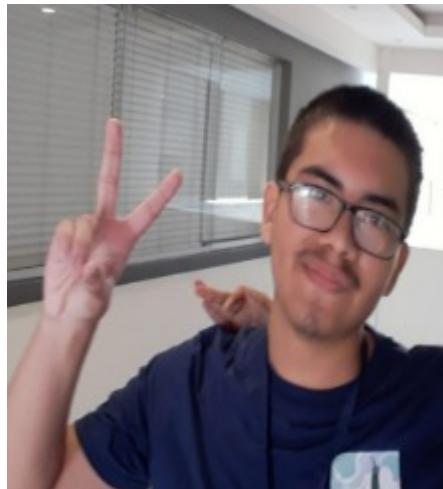


- Jorge Enrique Guevara Tejada:

**Descripción:**

Mi nombre es Jorge Enrique Guevara Tejada, soy un estudiante que se caracteriza por su alto sentido de la responsabilidad y un fuerte compromiso con su equipo. Me esfuerzo continuamente por superar los

desafíos académicos, dedicando tiempo adicional a mejorar en las áreas donde siento que puedo crecer. Mi objetivo no es sólo alcanzar un rendimiento académico satisfactorio, sino también contribuir de manera significativa al éxito del equipo, asegurando que cada proyecto en el que participó refleja nuestro esfuerzo y dedicación colectivos.



- **Jean Pierre Grandez Mansilla: Descripción:**
- Mi nombre es Jean Pierre Grandez Mansilla, tengo 21 años y soy estudiante de Ingeniería de Software, actualmente en mi 7mo ciclo. Me gusta la programación, pero sobre todo me interesa entender cómo funcionan realmente las cosas detrás de cada tecnología. Últimamente me enfoco en aprender sobre cloud, serverless y también me llama la atención el desarrollo móvil. Además, la música es una parte importante de mi día a día.

## 1.2. Solution Profile

En esta sección se abordará el perfil de la solución teniendo en cuenta el problema y las ideas que se deben resolver para fomentar la seguridad de los estudiantes.

### 1.2.1 Antecedentes y problemática

La inseguridad en los medios de transporte ha aumentado significativamente en el Perú, afectando tanto a los conductores como a los pasajeros. Este problema se acrecienta en el caso de los estudiantes universitarios, quienes muchas veces son víctimas de robos y asaltos, especialmente en zonas cercanas a sus centros de estudio. Asimismo, la extorsión a transportistas por el cobro de "cupos" limita la disponibilidad de servicios seguros, generando temor y restringiendo la libertad de movilización.

Dada las circunstancias, surge la necesidad de una solución tecnológica que permita a los estudiantes movilizarse con mayor confianza y seguridad. Es en este contexto que se propone **ÑanGo**, una aplicación móvil de transporte compartido orientada exclusivamente a la comunidad estudiantil. Esta solución facilita que los estudiantes que cuentan con movilidad propia puedan compartir sus rutas con otros compañeros, optimizando costos, ofreciendo mayor tranquilidad en los desplazamientos y reduciendo los riesgos de inseguridad mediante medidas de protección integradas en la plataforma.

### Problemática (5Ws y 2Hs)

#### WHAT

#### *¿Cuál es el problema?*

Los servicios de transporte actuales en el Perú presentan serias deficiencias en materia de seguridad. Robos, asaltos y extorsiones a conductores son frecuentes, exponiendo a los estudiantes a situaciones de riesgo. Esto ha generado desconfianza y una necesidad urgente de contar con un servicio de transporte confiable, seguro y accesible.

### **¿Cuál es la relación con los estudiantes?**

Este problema impacta directamente en los universitarios, quienes necesitan trasladarse a sus centros de estudio de manera segura y económica. La falta de opciones adecuadas genera estrés, miedo y limita su libertad de movilidad.

### **WHEN**

#### **¿Cuándo sucede el problema?**

El problema ocurre diariamente en las calles de Lima, especialmente en los horarios de entrada y salida de las universidades. Estos momentos coinciden con horas punta, cuando el tráfico es intenso y las zonas cercanas a los campus se convierten en puntos críticos para la delincuencia.

#### **¿Cuándo utilizan nuestra aplicación los estudiantes?**

Los usuarios emplearán la aplicación principalmente antes de sus clases, para programar y coordinar viajes compartidos con otros estudiantes verificados, reduciendo tiempos de espera y optimizando su movilidad diaria.

### **WHERE**

#### **¿Dónde está el cliente cuando usa el producto?**

Los estudiantes acceden a ÑanGo desde sus hogares, universidades u otros lugares con conexión a internet, para coordinar viajes y planificar sus rutas de manera anticipada.

#### **¿A dónde se dirige?**

El cliente utiliza ÑanGo principalmente para dirigirse hacia y desde su universidad, aunque la aplicación también permite coordinar rutas relacionadas con su vida estudiantil (por ejemplo, actividades extracurriculares o prácticas profesionales).

#### **¿Dónde surge el problema?**

El problema se origina en las calles y avenidas de Lima y otras ciudades universitarias, donde predominan el transporte informal, la congestión y la inseguridad, generando un entorno riesgoso para los estudiantes.

### **WHO**

#### **¿Quiénes están involucrados? ¿Quién lo utilizará?**

Los principales involucrados son los estudiantes universitarios, quienes pueden participar como pasajeros o conductores. Los conductores (con movilidad propia) buscan compartir viajes y reducir costos, mientras que los pasajeros necesitan una alternativa segura para llegar a sus universidades.

También se consideran involucrados los familiares de los estudiantes, quienes podrían ocasionalmente brindar transporte seguro a compañeros de sus hijos, fomentando la confianza dentro de la comunidad.

académica.

## WHY

### ***¿Cuál es la causa del problema?***

La principal causa es la falta de un sistema de transporte seguro, regulado y confiable para los estudiantes universitarios. La informalidad en el servicio de taxis, la insuficiente presencia policial en zonas críticas y la ausencia de tecnología que facilite la coordinación de viajes seguros agravan el problema. Además, los altos costos del transporte privado hacen que muchos jóvenes opten por alternativas más económicas, aunque menos seguras.

## HOW

### ***¿Cómo usan los estudiantes nuestra aplicación?***

Los usuarios emplearán la aplicación móvil en condiciones de alta demanda (horas punta o zonas inseguras). A través de ella podrán coordinar sus rutas, ahorrar dinero al compartir gastos y movilizarse con la confianza de hacerlo junto a otros estudiantes verificados.

### ***¿Cómo se conoce nuestra aplicación?***

ÑanGo será conocido inicialmente mediante campañas digitales en redes sociales y a través de recomendaciones entre estudiantes. La aplicación ofrecerá notificaciones instantáneas, coordinación de gastos y validación de perfiles como principales diferenciadores.

## HOW MUCH

### ***¿Cuál es la magnitud del problema?***

En Lima y Callao, el cobro de cupos por bandas criminales al transporte urbano formal representa una carga económica significativa. Se estima que más de S/ 10 millones mensuales salen del sector transporte para pagar extorsiones a estas mafias. (Infobae, 2025).

Cerca del 80 % de las empresas formales de transporte urbano están afectadas por este cobro. En algunos casos específicos, los choferes deben pagar entre S/ 5 y S/ 10 soles por cada vuelta que realizan los buses, lo cual al acumularse entre muchas unidades y rutas supone pérdidas mensuales elevadas.(Exitosa, 2025)

Además del factor económico, la inseguridad se ve agravada por amenazas, asesinatos de transportistas y extorsiones físicas o con daño a la propiedad cuando los pagos no se efectúan. (Infobae, 2025).

## 1.2.2. Lean UX Process

La metodología Lean UX se enfoca en el usuario y en la validación constante de soluciones frente a una problemática. Tras identificar la necesidad, consideramos adecuado aplicar esta metodología para diseñar una solución que sea relevante y efectiva.

### **1.2.2.1 Lean UX Problem Statements**

La inseguridad ciudadana en el Perú, especialmente en Lima, ha convertido el transporte diario en una experiencia de alto riesgo para los estudiantes universitarios, quienes enfrentan robos, asaltos y

extorsiones de manera constante. La ausencia de un sistema de movilidad seguro, confiable y accesible los obliga a optar entre alternativas costosas o arriesgadas, afectando su calidad de vida y su tranquilidad.

Ante este contexto, surge la necesidad de una solución tecnológica que permita a los estudiantes trasladarse de manera segura y económica, compartiendo rutas y gastos con compañeros verificados. El desafío es diseñar una plataforma móvil que promueva confianza, optimice los desplazamientos y brinde protección a la comunidad universitaria en su día a día.

## Aspectos

### Domain:

Movilidad urbana y seguridad ciudadana aplicada a estudiantes universitarios en contextos urbanos del Perú.

### Customer Segments

- **Estudiantes Universitarios sin vehículo propio**

Jóvenes que necesitan desplazarse diariamente a sus universidades y enfrentan desafíos como inseguridad en el transporte público, costos elevados y estrés en sus trayectos.

- **Estudiantes con vehículo privado**

Propietarios de autos que desean compartir sus viajes con compañeros de confianza para reducir gastos en combustible, aumentar la seguridad en sus trayectos y aprovechar de forma más eficiente el uso del vehículo.

### Pain Points

- **Estudiantes sin vehículo propio**

- **Inseguridad:** Exposición constante a robos y asaltos en transporte público e informal.
- **Costo elevado:** Gastos recurrentes en transporte privado (taxis, aplicativos) que no siempre garantizan seguridad.
- **Estrés y desconfianza:** Desplazamientos que generan ansiedad e incertidumbre, afectando su bienestar y rendimiento académico.

- **Estudiantes con vehículo privado**

- **Bajo aprovechamiento del vehículo:** Viajes realizados con asientos vacíos que generan gastos innecesarios en combustible.
- **Falta de confianza:** Dificultad para identificar compañeros confiables con quienes compartir el viaje.

### Gap

Actualmente no existe una solución tecnológica orientada específicamente a estudiantes universitarios que combine seguridad, accesibilidad y economía en el transporte compartido. Las aplicaciones de movilidad

disponibles no contemplan mecanismos de verificación entre pares ni se adaptan a los horarios y necesidades académicas.

## Vision / Strategy

Diseñar e implementar una aplicación móvil de carpooling enfocada en estudiantes universitarios, que permita:

- Conectar a compañeros verificados mediante la universidad.
- Compartir rutas de forma segura y transparente.
- Reducir costos de transporte, optimizando el uso de los vehículos disponibles.
- Fomentar un entorno de confianza, protección y accesibilidad en la movilidad diaria.

## Initial Segment

Estudiantes universitarios de Lima Metropolitana, especialmente en zonas con mayor incidencia de inseguridad y demanda de transporte compartido.

### 1.2.2.2. Lean UX Assumptions

#### User Assumptions

##### 1. ¿Quién es el usuario?

El usuario es cualquier estudiante universitario (como pasajero o conductor) que busca una alternativa segura, económica y confiable para movilizarse hacia su universidad. También incluye a estudiantes con vehículo propio interesados en compartir sus viajes y reducir gastos.

##### 2. ¿Dónde encaja nuestro producto en su vida?

En su rutina diaria de desplazamiento hacia y desde la universidad. La app móvil se integra como una herramienta práctica y rápida que sustituye el transporte inseguro o costoso, especialmente en horas pico y zonas de riesgo.

##### 3. ¿Qué problemas debe resolver nuestro producto?

- La inseguridad en el transporte público e informal.
- Los costos elevados de taxis y aplicativos.
- La falta de confianza al compartir viajes con desconocidos
- La ineficiencia en la coordinación de horarios y rutas.

##### 4. ¿Cuándo y cómo se usa el producto?

La aplicación se utiliza principalmente en las mañanas y tardes, al coordinar viajes hacia y desde la universidad. Los estudiantes ingresan su ruta y horario en la app, que los conecta con compañeros verificados que tengan trayectos similares.

##### 5. ¿Qué características son importantes?

- Verificación de identidad con credencial universitaria.
- Calificación y comentarios entre usuarios.
- Sistema de gastos compartidos.

- Visualización de rutas seguras y recomendadas.
- Chat interno entre pasajeros y conductores.
- Notificaciones en tiempo real sobre cambios de viaje.

## 6. ¿Cómo debe verse y comportarse el producto?

La aplicación debe ser juvenil, intuitiva y confiable, con diseño simple y botones de acción rápida.

Debe cargar rápido, enviar notificaciones claras y transmitir seguridad mediante íconos visibles de verificación, soporte de emergencia y rutas validadas.

### Business Assumptions

- Creemos que los estudiantes necesitan una aplicación móvil que les permita compartir viajes de manera segura, económica y confiable, con usuarios verificados de su propia universidad.
- Esta necesidad se puede cubrir con una app que integre: validación universitaria, rutas seguras, reparto automático de gastos y funciones de emergencia.
- El principal valor que los clientes esperan es seguridad en sus traslados, seguida de accesibilidad económica y confianza entre usuarios.
- La adquisición de usuarios se logrará a través de campañas digitales en redes sociales, difusión en campus universitarios y convenios con universidades.
- La competencia directa son taxis y aplicaciones de movilidad (Uber, InDrive, Cabify), que no ofrecen exclusividad ni seguridad universitaria.
- Superaremos a la competencia ofreciendo una comunidad cerrada de estudiantes verificados, viajes compartidos y opciones de seguridad en ruta.
- El mayor riesgo es la desconfianza inicial de los usuarios en compartir viajes.
- Este riesgo se mitigará mediante: perfiles verificados, sistema de calificaciones, protocolos de emergencia, soporte en tiempo real y retroalimentación constante.
- La confianza aumentará con el tiempo a medida que los usuarios vivan experiencias seguras y compartan reseñas positivas.
- La satisfacción se reforzará con incentivos (recompensas por uso frecuente) y mejoras continuas del sistema.

### 1.2.2.3 Lean UX Hypothesis Statements

#### 1. Optimización de tiempo y costos

**Creemos que** los estudiantes podrán optimizar su tiempo y reducir costos de transporte si implementamos una funcionalidad que les permita coordinar viajes en grupo con compañeros de su misma universidad.

**Sabremos que** hemos tenido éxito cuando al menos el **30 % de los estudiantes utilicen regularmente la plataforma** para coordinar viajes y un **25 % reporte una disminución en sus gastos de transporte mensual.**

## 2. Sostenibilidad y reducción de huella de carbono

**Creemos que** los estudiantes contribuirán a la sostenibilidad y reducirán su huella de carbono si les ofrecemos una opción sencilla para coordinar viajes compartidos, disminuyendo el número de autos individuales en rutas comunes hacia las universidades.

**Sabremos que** hemos tenido éxito cuando al menos el 20 % de los estudiantes reporten haber preferido compartir vehículo en lugar de usar transporte público o privado.

## 3. Seguridad y confianza en los traslados

**Creemos que** los estudiantes mejorarán su accesibilidad y seguridad en los traslados si incluimos un sistema de calificación y seguimiento de viajes compartidos que genere mayor confianza y control.

**Sabremos que** hemos tenido éxito cuando al menos el 15 % de los usuarios reporten mayor confianza y comodidad al usar la plataforma, y un 20 % afirme que el sistema de calificación ha mejorado su experiencia de viaje.

## 4. Verificación de identidad y afinidad

**Creemos que** los estudiantes estarán más dispuestos a utilizar la plataforma si esta ofrece perfiles verificados y filtros de coincidencia por universidad o facultad.

**Sabremos que** hemos tenido éxito cuando al menos el 60 % de los usuarios activen la verificación de identidad y un 70 % prefiera viajar con contactos verificados de su misma institución.

## 5. Notificaciones y puntualidad

**Creemos que** la adopción de la plataforma aumentará si se integra un sistema de notificaciones para viajes programados.

**Sabremos que** hemos tenido éxito cuando el 80 % de los trayectos se confirmen con más de 6 horas de anticipación y se logre una reducción del 30 % en cancelaciones de último minuto.

## 6. Pagos digitales y gestión de gastos

**Creemos que** la plataforma será más atractiva si permite compartir gastos automáticamente mediante un sistema de pagos digitales integrado.

**Sabremos que** hemos tenido éxito cuando al menos el 50 % de los viajes registrados utilicen el sistema de pago integrado y el 85 % de los usuarios lo califiquen como fácil y seguro.

### 1.2.2.4. Lean UX Canvas

| Lean Product Canvas (v3)   |   |  |
|--|---|--|
| <p>Title of initiative:<br/>NanGo – Plataforma de viajes compartidos universitarios</p> <p>Date: 05 / 09 / 2025<br/>Iteration: v1</p>  |   |  |
| <b>Business Problem</b>  | <b>Solutions</b>  | <b>Business Outcomes</b>   |
| <p>What is the most important problem your business needs to focus on right now?</p> <p>En las ciudades universitarias, los estudiantes enfrentan altos costos de transporte, congestión vehicular y preocupaciones por la sostenibilidad. Actualmente, no existe una solución eficiente y accesible para coordinar viajes compartidos, lo que genera un impacto negativo en los presupuestos de los estudiantes y en el medio ambiente.</p> <p>Nango busca resolver esta problemática ofreciendo una plataforma para que los estudiantes puedan compartir viajes de manera segura y económica. Además, integra a familiares de los estudiantes como posibles conductores, brindando un entorno de mayor confianza y apoyo para la comunidad universitaria.</p> <p>¿Cómo podemos desarrollar una plataforma que conecte de manera segura y eficiente a estudiantes y familiares que actúan como conductores, con estudiantes que necesitan transporte, permitiendo la planificación y coordinación de viajes, comunicación interna entre usuarios y un sistema de calificaciones que genere confianza en la comunidad?</p> | <p>Plataforma web y móvil para que estudiantes y familiares puedan encontrar compañeros de viaje, publicar rutas y horarios disponibles.</p> <p>Sistema de calificación y comentarios que fomente la confianza entre conductores y pasajeros, reforzando la seguridad y buena conducta.</p> <p>Notificaciones y seguimiento en tiempo real, incluyendo recordatorios de viajes, estado del conductor, y actualizaciones de ruta.</p> <p>Chat interno seguro para que estudiantes y choferes coordinen detalles del viaje sin necesidad de compartir números personales.</p> <p>Verificación de identidad y validación universitaria, vinculando la cuenta con correos institucionales o documentos que garanticen que los usuarios pertenezcan a la comunidad universitaria.</p> <p>Sistema de recompensas o puntos, donde choferes y pasajeros acumulan beneficios por viajes realizados, fomentando la participación activa.</p> <p>Filtros de búsqueda avanzada para pasajeros (ejemplo: por horario, proximidad, género del conductor, frecuencia del viaje), aumentando la personalización.</p> <p>Integración con métodos de pago y división automática de gastos, para que compartir los costos sea simple, transparente y seguro.</p> | <p>How will you know you solved the business problem? What will you measure?</p> <p>Estudiantes (choferes y pasajeros) y familiares de los estudiantes que buscan una alternativa más económica y segura para llegar a la universidad.</p> <ul style="list-style-type: none"> <li>- Los choferes tienen vehículos disponibles.</li> <li>- Los pasajeros buscan reducir sus gastos de transporte.</li> </ul>  |
| <b>Users</b>   |   | <b>User Outcomes &amp; Benefits (JTBD)</b>   |
| <p>Estudiantes (choferes y pasajeros) y familiares de los estudiantes que buscan una alternativa más económica y segura para llegar a la universidad.</p> <ul style="list-style-type: none"> <li>- Los choferes tienen vehículos disponibles.</li> <li>- Los pasajeros buscan reducir sus gastos de transporte.</li> </ul>   |   | <p>Estudiantes (choferes y pasajeros):</p> <ul style="list-style-type: none"> <li>- Ahorro económico al compartir gastos de transporte.</li> <li>- Conexión rápida y segura con compañeros de universidad para organizar viajes.</li> <li>- Optimización del tiempo de desplazamiento mediante rutas eficientes.</li> <li>- Mayor seguridad al viajar con estudiantes verificados.</li> </ul> <p>Familiares (choferes):</p> <ul style="list-style-type: none"> <li>- Posibilidad de movilizarse junto a sus hijos o con estudiantes verificados.</li> <li>- Ahorro en combustible al compartir gastos del viaje.</li> <li>- Tranquilidad al tener un historial de viajes y calificaciones.</li> <li>- Confianza en un sistema diseñado para la seguridad estudiantil.</li> </ul> |
| <b>Hypotheses</b>  | <b>What's the most important thing we need to learn first?</b>  | <b>What's the least amount of work we need to do to learn the next most important thing?</b>   |
| <p>Combine the assumptions from Boxes 2, 3, 4 &amp; 5 into the following hypothesis statement: "We believe that [business outcome] will be achieved if [user] attains [benefit] with [feature]."</p> <p>Choferes: Creemos que los estudiantes choferes usarán la plataforma si es fácil y confiable para coordinar viajes y compartir gastos. Éxito al menos 30% registrados y usuarios regulares.</p> <p>Pasajeros: Creemos que los estudiantes pasajeros usarán la plataforma si encuentran choferes de confianza y comparten gastos de forma segura. Éxito al menos 30% registrados y usuarios regulares.</p> <p>Familiares: Creemos que los familiares estarán más tranquilos si acceden a los detalles de los viajes. Éxito al menos 15% de los familiares choferes usando la plataforma.</p>   | <p>Zonas de alta demanda de transporte.</p> <p>Necesidades específicas de los estudiantes y familias.</p> <p>Patrones de uso del transporte.</p> <p>Interacción entre estudiantes y choferes.</p> <p>Medidas de seguridad necesarias.</p> <p> To help you choose which hypothesis to test, refer to our Hypothesis Prioritization Canvas (<a href="https://bit.ly/hypo-canva">https://bit.ly/hypo-canva</a>)</p>   | <p>Encuestas y entrevistas iniciales a estudiantes y familiares.</p> <p>Análisis básico de seguridad (puntualidad, estado de vehículos).</p> <p>Análisis competitivo rápido de plataformas existentes.</p>   |

 Download this canvas at: [www.senseandrespond.co/the-lean-product-canvas](http://www.senseandrespond.co/the-lean-product-canvas)

## 1.3. Segmentos objetivo

### Segmento 1: Estudiantes con vehículo

#### Características Demográficas

- **Edad:** 18 años a más
- **Nacionalidad:** Peruana
- **Ubicación:** Áreas urbanas (principalmente Lima Metropolitana y ciudades universitarias)
- **Nivel educativo:** Estudiantes universitarios
- **Intereses:** Uso eficiente del vehículo, ahorro en combustible, opciones de estacionamiento seguro, comodidad en sus desplazamientos.

#### Características Psicográficas

- Estudiantes que poseen un vehículo propio y están dispuestos a compartirlo para reducir costos de transporte.
- Valoran aplicaciones móviles que les permitan coordinar viajes en tiempo real, con recordatorios y notificaciones sobre horarios y pasajeros.
- Buscan comodidad y confianza al viajar con compañeros verificados dentro de la comunidad universitaria.
- Ven la plataforma como una forma de generar ingresos adicionales o cubrir parcialmente gastos de movilidad.

### Segmento 2: Estudiantes sin vehículo

#### Características Demográficas

- Edad: Desde los 16 años en adelante
- Nacionalidad: Peruana
- Ubicación: Áreas urbanas (zonas con alta concentración de universidades)
- Nivel educativo: Principalmente estudiantes de universidades e institutos.
- Intereses: Transporte seguro, accesible, puntual y confiable.

### Características Psicográficas

- Estudiantes que no cuentan con un vehículo propio o desean reducir sus gastos de transporte.
- Prefieren soluciones móviles que les permitan reservar viajes fácilmente, recibir alertas de salida y confirmar rutas en pocos pasos.
- Buscan seguridad a través de perfiles verificados y sistemas de calificación de conductores y pasajeros.
- Valoran la conveniencia de coordinar viajes desde el celular sin depender de alternativas riesgosas o costosas.

## Capítulo II: Requirements Development and Software Solution Design

---

### 2.1. Competidores

Se han identificado los siguientes competidores en el mercado de soluciones de movilidad colaborativa y carpooling para estudiantes universitarios y usuarios urbanos:

- Competidor 1: EnRuta (PUCP)

Descripción: Plataforma universitaria que conecta a estudiantes de la PUCP para crear "salas de viaje" según puntos de partida y destino. No gestiona pagos, su foco es la seguridad y la confianza entre miembros de la comunidad educativa.

- Competidor 2: inDrive (Peer-to-Peer)

Descripción: App internacional de ride-hailing que permite la negociación directa entre conductores y pasajeros. Aunque no está orientada exclusivamente a estudiantes, es un referente en movilidad colaborativa bajo un modelo peer-to-peer, con presencia en múltiples ciudades.

- Competidor 3: BlaBlaCar (Carpooling Global)

Descripción: Plataforma líder en carpooling a nivel global. Conecta conductores con pasajeros que comparten trayectos de media y larga distancia. Su foco es la optimización de costos y la sostenibilidad, consolidándose como el ejemplo más avanzado de carpooling digital.

#### 2.1.1. Análisis competitivo

**Competitive  
Analysis  
Landscape**

---

## Competitive Analysis Landscape

¿Por qué llevar a cabo este análisis?

Este análisis nos permite identificar competidores con distintos grados de madurez en el ecosistema de movilidad colaborativa: desde iniciativas universitarias locales hasta plataformas globales de carpooling.

De esta forma podemos contrastar nuestra propuesta con alternativas existentes, posicionándonos como una opción intermedia enfocada en **seguridad, comunidad universitaria y accesibilidad**.

|                            |                             | Nuestra Startup (ÑanGO)  | EnRuta (PUCP)  | inDrive (Peer-to-Peer)  | BlaBlaCar (Carpooling Global)  |
|----------------------------|-----------------------------|--|--|---|--|
| <b>PERFIL</b>              | Overview                    | App universitaria de carpooling que conecta <b>estudiantes conductores y pasajeros</b> de una misma comunidad académica. | App universitaria PUCP para compartir rutas entre estudiantes.   | App global peer-to-peer donde conductor y pasajero <b>negocian el precio</b> de cada viaje. | Plataforma global de carpooling que conecta viajes de media/larga distancia. |
| <b>PERFIL</b>              | Ventaja competitiva (valor) | <b>Seguridad + Comunidad:</b> usuarios verificados como estudiantes + confianza entre pares.                             | Exclusividad universitaria y foco en seguridad.                  | Flexibilidad en negociación de precios y gran disponibilidad.                               | Alcance internacional, amplia base de usuarios y trayectoria consolidada.    |
| <b>Perfil de Marketing</b> | Mercado objetivo            | Estudiantes universitarios de Lima y posteriormente de LatAm.  | Estudiantes de la PUCP.  | Usuarios urbanos globales que buscan transporte económico y flexible.                       | Viajeros interurbanos y de larga distancia en todo el mundo.                 |
| <b>Perfil de Marketing</b> | Estrategia de marketing     | Difusión universitaria, convenios institucionales, marketing digital en campus.  | Comunicación interna de la universidad, canales institucionales. | Marketing masivo con foco en ahorro y flexibilidad.   | Marketing global con casos de éxito y enfoque en sostenibilidad.             |

|                           |                       | Nuestra Startup(ÑanGO)   | EnRuta (PUCP)  | inDrive (Peer-to-Peer)   | BlaBlaCar (Carpooling Global)   |
|---------------------------|-----------------------|--|--|--|---|
| <b>Perfil de Producto</b> | Productos & servicios | App móvil con reservas de asientos, validación de identidad y pagos integrados.          | App móvil básica con salas de viaje y sin gestión de pagos.  | App móvil con negociación de tarifas, geolocalización y sistema de reputación. | App y web con planificación de viajes, reservas, pagos y sistema de calificación. |
| <b>Perfil de Producto</b> | Precios & costos      | Modelo accesible: costo compartido con tarifa base universitaria.                        | Gratis (sin monetización).                                   | Precios variables según negociación entre conductor y pasajero.                | Tarifa definida por el conductor; comisión por uso de plataforma.                 |
| <b>Perfil de Producto</b> | Canales (Web/Móvil)   | App móvil + web de soporte, con foco en comunidad universitaria.                         | App móvil interna PUCP.                                      | App móvil (iOS/Android).   | App móvil y sitio web global.   |
| <b>Análisis SWOT</b>      | Fortalezas            | Comunidad universitaria cerrada, seguridad, accesibilidad económica.                     | Exclusividad universitaria, percepción de seguridad.         | Flexibilidad en tarifas, presencia internacional.                              | Amplia base de usuarios, trayectoria global, confianza en marca.                  |
| <b>Análisis SWOT</b>      | Debilidades           | Recursos limitados, crecimiento dependiente de convenios con universidades.              | Limitado a una sola universidad, sin modelo de monetización. | Riesgo de inseguridad al negociar con desconocidos.                            | Poco enfoque en estudiantes y viajes cortos urbanos.                              |
| <b>Análisis SWOT</b>      | Oportunidades         | Escalar a otras universidades, integrar funciones de pago y reputación, crecer en LatAm. | Expandirse a otras universidades.                            | Captar estudiantes como nuevo segmento objetivo.                               | Expandirse hacia viajes urbanos cortos y convenios con universidades.             |

|               |          | Nuestra Startup(ÑanGO)   | EnRuta (PUCP)  | inDrive (Peer-to-Peer)                              | BlaBlaCar (Carpooling Global)  |
|---------------|----------|--|--|---|--|
| Análisis SWOT | Amenazas | Entrada de apps globales al nicho universitario, resistencia cultural al carpooling. | Dependencia de políticas internas de la universidad. | Competencia con taxis tradicionales y ride-hailing. | Competidores emergentes más especializados en nichos como estudiantes. |

### 2.1.2. Estrategias y tácticas frente a competidores

Para fortalecer nuestra propuesta frente a los competidores identificados, se plantean las siguientes estrategias y tácticas:

- Diferenciación por Seguridad y Comunidad
  - Implementar un sistema de verificación obligatoria de identidad universitaria (carné, correo institucional).
  - Potenciar el sentido de pertenencia y confianza al limitar la plataforma a estudiantes.
- Enfoque en Universidades y Jóvenes
  - Establecer convenios con universidades para promover la app dentro de campus.
  - Posicionar la solución como una alternativa económica, segura y confiable frente al transporte público y ride-hailing.
- Escalabilidad Regional
  - Comenzar en Lima con universidades privadas y públicas estratégicas.
  - Expandir posteriormente a otras ciudades de Latinoamérica con fuerte concentración universitaria.
- Valor agregado en la experiencia de usuario
  - Pagos integrados y transparentes dentro de la app.
  - Funciones de reputación (calificaciones, reseñas) para generar confianza.
  - Incentivos para conductores (reducción de costos de combustible, beneficios por viajes frecuentes).

## 2.2. Entrevistas.

En esta sección se detallan las entrevistas realizadas a los segmentos objetivo identificados previamente, con el fin de comprender mejor sus necesidades, comportamientos y expectativas respecto a soluciones de monitoreo de cadena de frío en el sector salud.

### 2.2.1. Diseño de entrevistas.

Se han diseñado entrevistas semiestructuradas para dos segmentos objetivo principales, con el objetivo de profundizar con sus necesidades y validar nuestras hipótesis iniciales.

- **Segmento 1: Estudiantes con vehículo**

- **Objetivo:** Entender sus motivaciones para compartir viajes, preocupaciones sobre seguridad y expectativas de una plataforma de carpooling.

#### **Preguntas para el Segmento 1: Estudiantes con vehículo**

1. ¿Con qué frecuencia utilizas tu vehículo para ir a la universidad?
2. ¿Cuánto gastas aproximadamente al mes en combustible y estacionamiento?
3. ¿Qué opinas sobre compartir tu vehículo con otros estudiantes?
4. ¿Qué incentivo te motivaría más a ofrecer viajes compartidos? (ej. reducción de costos, compañía, recompensas).
5. ¿Qué tan importante sería para ti contar con un sistema de calificaciones o validación de pasajeros?
6. ¿Qué tipo de compensación esperarías recibir por ofrecer un asiento (dinero, beneficios, descuentos, etc.)?
7. ¿Qué funcionalidades crees que debería tener una aplicación que te ayude a organizar tus viajes compartidos?
8. ¿Qué tan cómodo te sentirías coordinando viajes con pasajeros mediante una app?
9. ¿Qué preocupaciones tendrías al transportar a otros estudiantes? (ej. seguridad, puntualidad, responsabilidad legal).
10. ¿Qué condiciones deberían cumplirse para que realmente ofrezcas tu auto?

- **Segmento 2: Estudiantes sin vehículo**

- **Objetivo:** Identificar sus principales desafíos de movilidad, criterios para elegir transporte y disposición a usar una app de carpooling.

#### **Preguntas para el Segmento 2: Estudiantes sin vehículo**

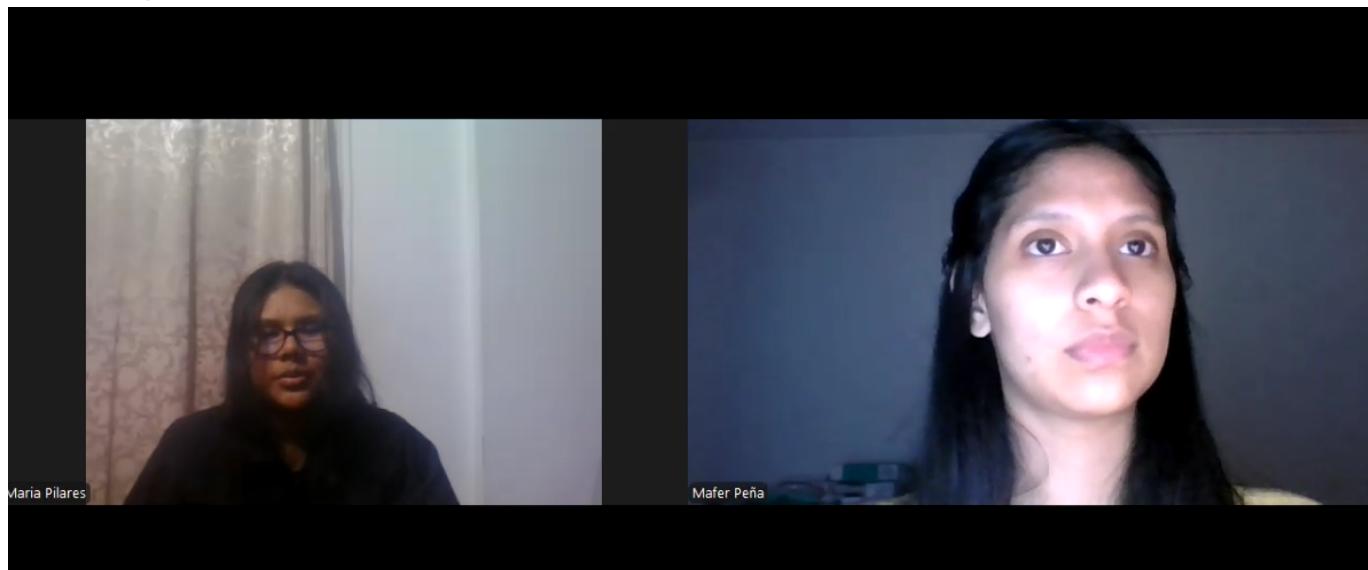
1. ¿Qué medio de transporte utilizas actualmente para ir a la universidad?
2. ¿Qué aspectos te incomodan más de tu medio de transporte actual? (ej. tiempo, seguridad, costos, comodidad).
3. ¿Cuánto tiempo promedio tardas en llegar a la universidad?
4. ¿Cuánto gastas semanalmente en transporte?
5. ¿Qué tan dispuesto estarías a compartir un viaje con otros estudiantes de tu universidad? ¿Por qué?
6. ¿Qué beneficios esperarías obtener al usar un servicio de carpooling entre estudiantes?
7. ¿Qué tan importante es para ti la seguridad al usar un servicio de transporte compartido?
8. ¿Preferirías un servicio más económico aunque tarde un poco más, o pagar más por ahorrar tiempo?
9. ¿Con qué frecuencia utilizas aplicaciones móviles para tus actividades diarias?
10. ¿Qué características valoras más en una aplicación de transporte (ej. facilidad de uso, rapidez, métodos de pago, calificaciones de usuarios)?
11. ¿Qué te generaría dudas o desconfianza al usar una plataforma de carpooling con otros estudiantes?
12. ¿Qué condiciones deberían cumplirse para que realmente decidas usar un servicio de carpooling?

#### 2.2.2. Registro de entrevistas

- **Segmento 1: Estudiantes con vehículo**

Nombre: María Pilares Edad: 23 años Universidad: Universidad Peruana de Ciencias Aplicadas (UPC)

Carrera: Ingeniería de Software



### Entrevista María Pilares

Resumen de la entrevista: María actualmente utiliza su vehículo personal para ir a la universidad, lo que le permite mayor comodidad y flexibilidad en sus horarios. Sin embargo, reconoce que los costos asociados al combustible y estacionamiento son elevados, gastando aproximadamente 300 soles mensuales en estos conceptos. Está abierta a la idea de compartir su vehículo con otros estudiantes, siempre y cuando exista un sistema de verificación y calificación que le brinde confianza. Valora mucho la seguridad y preferiría contar con una plataforma que le permita coordinar viajes de manera sencilla y segura. Además, considera que recibir algún tipo de compensación económica o beneficios adicionales sería un incentivo importante para ofrecer viajes compartidos.

- **Segmento 2: Estudiantes sin vehículo**

Nombre: Huber Antonio Requejo Edad: 22 años Universidad: Universidad Pontificia Católica del Perú

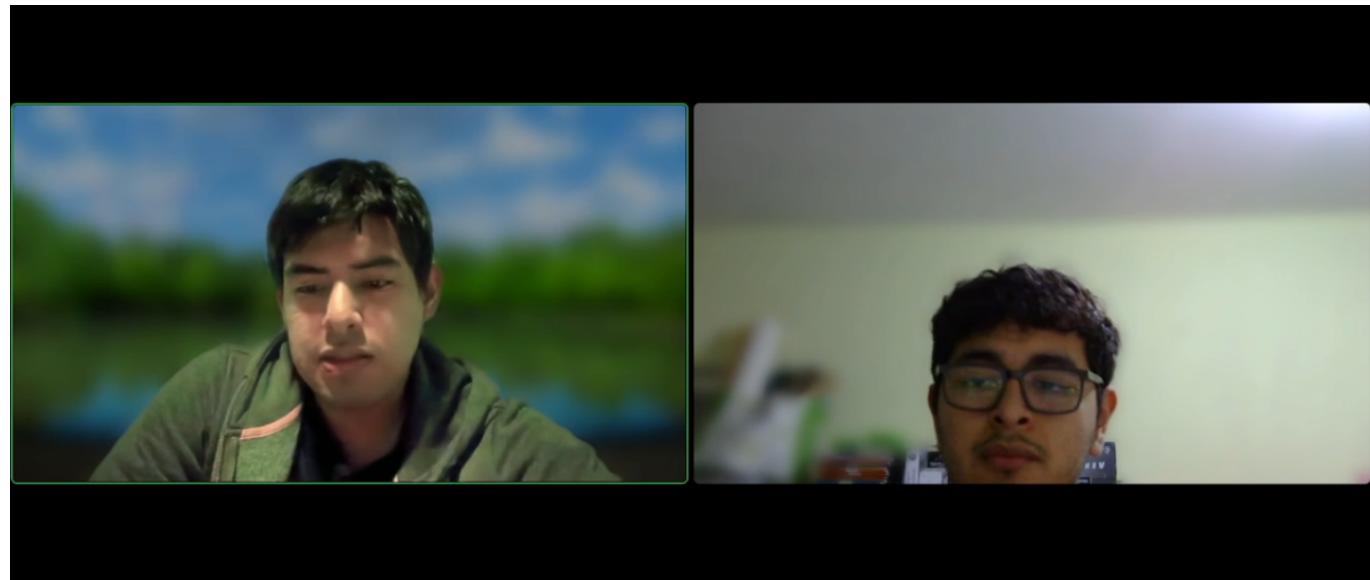
(PUCP) Carrera: Economía



### Entrevista Huber Antonio

Resumen de la entrevista: Huber nos comenta que normalmente utiliza el transporte público para ir a la universidad debido a que reside lejos del campus. Sin embargo, ha experimentado situaciones de inseguridad y robos en varias ocasiones, lo que le genera preocupación. Tarda aproximadamente 1 hora y 30 minutos en llegar a la universidad y gasta alrededor de 70 soles semanales en transporte. Además, Huber nos comenta que normalmente los taxis de aplicativos no todos suelen ser personas confiables, y que prefiere no usarlos a menos que sea una emergencia. Está bastante dispuesto a compartir un viaje con otros estudiantes, ya que considera que esto podría reducir sus costos y mejorar la seguridad al viajar con personas conocidas. Valora mucho la seguridad en un servicio de transporte compartido y preferiría pagar un poco más si eso significa llegar más rápido y seguro.

Nombre: Eduardo Ventura Edad: 22 años Universidad: Universidad Peruana de Ciencias Aplicadas (UPC)  
Carrera: Ingeniería de Software



### Entrevista Eduardo Ventura

Resumen de la entrevista: Eduardo nos comenta que actualmente utiliza el transporte público para ir a la universidad, lo cual le genera incomodidad debido a la falta de seguridad y la congestión en las horas punta. Tarda aproximadamente una hora en llegar a la universidad y gasta alrededor de 70 soles semanales en transporte. Está bastante dispuesto a compartir un viaje con otros estudiantes, ya que considera que esto podría reducir sus costos y mejorar la seguridad al viajar con personas conocidas. Valora mucho la seguridad en un servicio de transporte compartido y preferiría pagar un poco más si eso significa llegar más rápido y seguro.

Nombre: Sebastian Arevalo Edad: 20 años Universidad: Universidad San Martín de Porres (USMP) Carrera: Ingeniería de Software



## Entrevista Sebastian Arevalo

Resumen de la entrevista:

- ¿Qué medio de transporte utilizas actualmente para ir a la universidad? Generalmente uso buses de transporte público porque son lo más barato, aunque muchas veces se llenan demasiado y es incómodo. A veces recorro a aplicaciones de taxi cuando estoy apurado, pero solo en casos puntuales porque el costo es bastante más alto.
- ¿Qué aspectos te incomodan más de tu medio de transporte actual? Lo que más me incomoda es la falta de puntualidad y la cantidad de gente que viaja apretada en las horas pico, lo que hace el trayecto estresante. Además, me preocupa bastante la inseguridad, ya que he visto o escuchado de robos frecuentes dentro de los buses y combis.
- ¿Cuánto tiempo promedio tardas en llegar a la universidad? Normalmente me toma entre una hora y una hora y media llegar, aunque en hora punta puede llegar a ser casi dos horas. Esto significa que tengo que salir con mucha anticipación para no llegar tarde, lo cual me desgasta bastante todos los días.
- ¿Cuánto gastas semanalmente en transporte? Gasto en promedio entre 60 y 80 soles a la semana, considerando los pasajes diarios de ida y vuelta. A veces la cifra sube porque termino pidiendo un aplicativo de taxi cuando salgo tarde o tengo alguna entrega importante en la universidad.
- ¿Qué tan dispuesto estarías a compartir un viaje con otros estudiantes de tu universidad? ¿Por qué? Estaría bastante dispuesto, porque al final somos parte de la misma comunidad universitaria y eso me daría más confianza. Además, compartir un viaje significaría reducir costos y tener un trayecto más seguro en comparación al transporte público tradicional.
- ¿Qué beneficios esperarías obtener al usar un servicio de carpooling entre estudiantes? El primer beneficio que esperaría es el ahorro de dinero frente a lo que gasto en taxis o aplicativos. También valoro la seguridad de viajar con personas conocidas de la universidad y la comodidad de tener un trayecto más directo y menos cansado que en un bus lleno.
- ¿Qué tan importante es para ti la seguridad al usar un servicio de transporte compartido? Para mí la seguridad es un factor clave, incluso más importante que el costo o el tiempo de viaje. No usaría un

servicio si no sé con quién voy, o si no hay medidas claras para garantizar que las personas que viajan están verificadas.

- ¿Preferirías un servicio más económico aunque tarde un poco más, o pagar más por ahorrar tiempo? Depende de la situación, porque si tengo un examen o debo llegar muy puntual, preferiría pagar un poco más por un viaje rápido. Sin embargo, en la mayoría de los días me inclinaría por algo más económico para no gastar tanto a la semana.
- ¿Con qué frecuencia utilizas aplicaciones móviles para tus actividades diarias? Uso aplicaciones casi a diario, ya sea para pedir comida, comprar algo, hacer pagos o transportarme. Estoy acostumbrado a que las apps me faciliten la vida, así que no tendría problema en usar una para organizar mis viajes a la universidad.
- ¿Qué características valoras más en una aplicación de transporte? Valoro que la aplicación sea sencilla de usar, con precios claros y sin costos ocultos. También que tenga funciones de seguridad como compartir la ubicación en tiempo real y que muestre información detallada sobre los pasajeros y conductores.
- ¿Qué te generaría dudas o desconfianza al usar una plataforma de carpooling con otros estudiantes? Me generaría desconfianza que no haya un sistema de verificación de identidad o que cualquiera pueda registrarse sin control. También dudaría si no hubiera un canal claro para reportar situaciones incómodas o emergencias durante el viaje.
- ¿Qué condiciones deberían cumplirse para que realmente decidas usar un servicio como este? Que el servicio sea seguro con verificación de usuarios y monitoreo de los viajes en tiempo real. Además, que los precios sean justos y que la app sea clara con las rutas y horarios, mostrándome con quién viajo antes de aceptar el recorrido.

## 2.2.3 Análisis de entrevistas

En esta sección se presenta un análisis detallado de las entrevistas realizadas a los segmentos objetivo, identificando patrones comunes, necesidades clave y oportunidades para el diseño de la solución.

### Segmento 1: Estudiantes con vehículo

- **Patrones comunes:**
  - Alta disposición a compartir viajes con compañeros de universidad.
  - Preocupación significativa por la seguridad y confianza en los pasajeros.
  - Interés en incentivos económicos y beneficios adicionales.
  - Valoración de una plataforma fácil de usar y con funciones claras.
- **Necesidades clave:**
  - Sistema de verificación de identidad para pasajeros.
  - Funcionalidades de calificación y reseñas para generar confianza.
  - Opciones de compensación claras y justas por ofrecer viajes.
  - Notificaciones y recordatorios para coordinar viajes.

### Segmento 2: Estudiantes sin vehículo

- **Patrones comunes:**

- Uso frecuente de transporte público con preocupaciones sobre seguridad e incomodidad.
- Disposición a compartir viajes con otros estudiantes para reducir costos.
- Preferencia por aplicaciones móviles que faciliten la organización de viajes.
- Importancia de la seguridad y verificación de usuarios.
- Demoras significativas en los trayectos diarios.

- **Necesidades clave:**

- Sistema de verificación de identidad para conductores.
- Funcionalidades de calificación y reseñas para evaluar la confianza.
- Opciones de ahorro económico frente a transporte tradicional.
- Interfaz intuitiva y fácil de usar en la aplicación móvil.
- Puntualidad y claridad en rutas y horarios.

## 2.3 Needfinding

En esta sección se presentan las herramientas y técnicas utilizadas para identificar y comprender las necesidades de nuestros posibles usuarios, basándose en las entrevistas realizadas y el análisis de los segmentos objetivo.

### 2.3.1. User personas

Se desarrollaron dos user personas respecto a los segmentos objetivo identificados previamente, con el fin de representar de manera concreta y detallada a los usuarios potenciales de la aplicación ÑanGo.

\**Persona 1: Estudiante con vehículo*



### Goals

- Reducir sus gastos de combustible.
- Compartir viajes con estudiantes que tengan la misma ruta.
- Contribuir con una movilidad sostenible.

### Quote

*"Si voy en auto de todas formas, mejor compartir los gastos y de paso viajar en compañía"*

### Demographic

♂ Male      25 years

📍 Lima

Single

Estudiante

### Background

Posee un auto propio que utiliza para ir diariamente a la universidad. El tráfico y los costos de combustible son un problema constante. Le interesa aprovechar el trayecto compartiendo asientos disponibles y generar un ingreso adicional.

### Motivations

- Reducir gastos de transporte.
- Contribuir a disminuir la congestión vehicular.
- Viajar acompañada para mayor seguridad.
- Generar ingresos extras compartiendo asientos.

### Frustrations

- Gastos elevados de combustible y mantenimiento.
- Tráfico excesivo en Lima.
- Viajes solitarios que resultan monótonos e inseguros.

### Technology

### Needs

- Una app que le permita ofrecer asientos a estudiantes con rutas similares.
- Opciones de pago seguras y rápidas.
- Herramientas de reputación para generar confianza con los pasajeros.

### Expectations

- Que la plataforma asegure que los pasajeros sean estudiantes verificados.
- Que el sistema de reservas y pagos sea transparente y confiable.
- Que el uso de la app le ayude a reducir de manera significativa sus gastos de movilidad.

## Persona 2: Estudiante sin vehículo

**Demographic**

Gender  22 years  
  
 Single  
  
 Estudiante

**Technology**

**Goals**

- Llegar a la universidad de forma puntual y sin estrés.
- Ahorrar en costos de transporte diario.
- Sentirse seguro en los trayectos.

**Quote**

*“Solo quiero un transporte que sea seguro, económico y que no me haga perder tanto tiempo.”*

**Background**

Vive en un distrito alejado de su universidad. Todos los días debe usar transporte público congestionado, lo que le genera estrés y cansancio. A veces utiliza taxi por apps, pero le resulta caro a largo plazo.

**Motivations**

- Ahorrar dinero en transporte.
- Conocer a compañeros de su misma universidad.
- Optimizar tiempos de traslado.
- Sentirse parte de una comunidad segura

**Frustrations**

- Transporte público inseguro y lento.
- Altos costos en taxis o apps de ride-hailing.
- Retrasos frecuentes que afectan su puntualidad.

**Needs**

- Una app confiable que le permita coordinar viajes con estudiantes de su universidad.
- Funciones de seguridad como verificación de usuarios y calificaciones.
- Información clara sobre horarios y rutas.

**Expectations**

- Que el servicio sea accesible y económico.
- Que los conductores y pasajeros sean estudiantes verificados.
- Que la app sea fácil de usar y rápida en la búsqueda de viajes.

### 2.3.2. User Task Matrix

En esta sección se presentan las matrices de tareas para los dos segmentos objetivo, detallando las actividades clave que realizan en su rutina diaria de transporte hacia la universidad.

- Segmento 1: Estudiantes con vehículo

| Tareas                                    | Frecuencia | Importancia |
|---|------------|-------------|
| Conducir diariamente hacia la universidad | Alta       | Alta        |

| Tareas  | Frecuencia | Importancia |
|---|------------|-------------|
| Coordinar con amigos o conocidos para compartir viaje               | Media      | Media       |
| Cubrir gastos de combustible y mantenimiento                        | Alta       | Alta        |
| Adaptar horarios de salida según tráfico                            | Alta       | Alta        |
| Buscar y pagar estacionamiento en la universidad                    | Alta       | Alta        |
| Evaluar costos de transporte alternativo (apps, transporte público) | Media      | Media       |
| Considerar seguridad al viajar solo                                 | Alta       | Alta        |

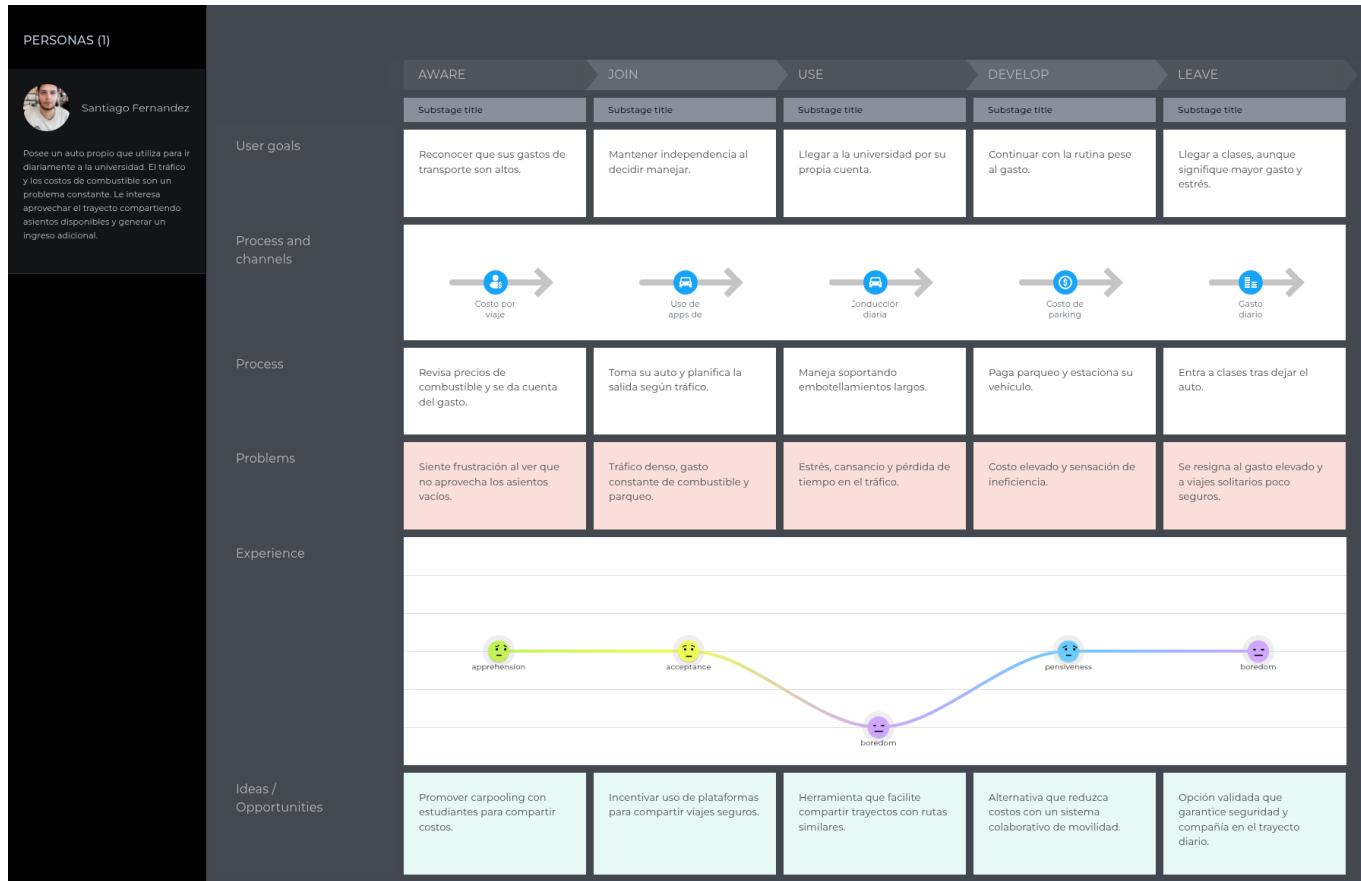
- Segmento 2: Estudiantes sin vehículo

| Tareas   | Frecuencia | Importancia |
|--|------------|-------------|
| Buscar opciones de transporte para llegar a clases     | Alta       | Alta        |
| Coordinar con amigos o conocidos para compartir viaje  | Media      | Media       |
| Pagar pasajes  | Alta       | Alta        |
| Adaptar horarios de salida según tráfico               | Alta       | Alta        |
| Viajar con incomodidad o estrés                        | Alta       | Alta        |
| Evaluar costos de transporte alternativo (taxis, apps) | Media      | Media       |
| Esperar transporte en paraderos                        | Alta       | Alta        |
| Considerar seguridad durante el trayecto               | Alta       | Alta        |

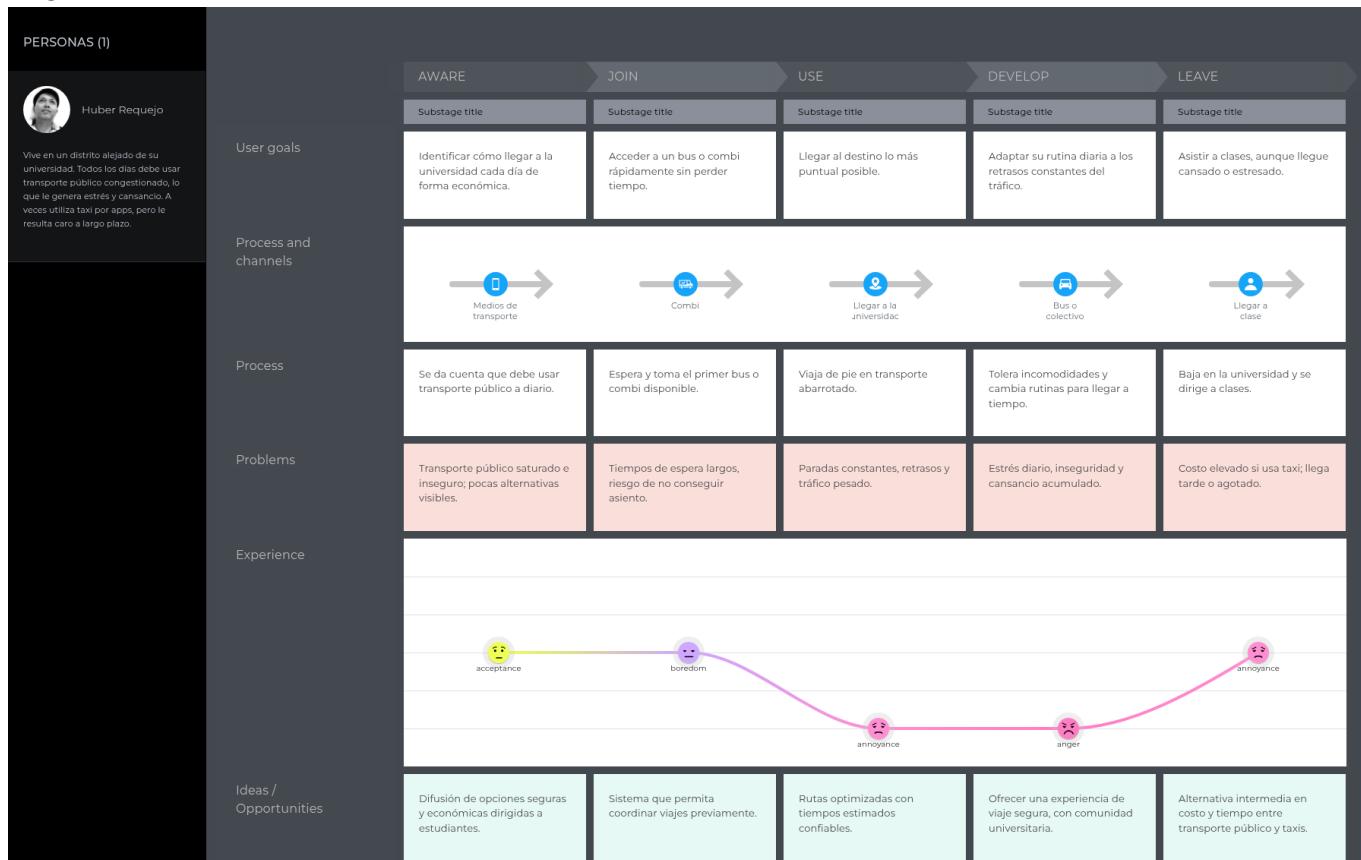
### 2.3.3. User Journey Mapping

Detallamos los recorridos de usuario para ambos segmentos, identificando puntos de contacto, emociones y oportunidades de mejora en su experiencia de transporte diario.

#### Segmento 1: Estudiante con vehículo



## Segmento 2: Estudiante sin vehículo



### 2.3.4. Empathy Map

## Segmento 1: Estudiante con vehículo

## 1.WHO are we empathizing with?

Un estudiante universitario de 24 años en Lima Metropolitana, con auto propio, que maneja diariamente a la universidad. Quiere reducir gastos de combustible y aprovechar el viaje compartiéndolo con otros estudiantes.

## 7.What do they THINK and FEEL?

- “
- Quiere optimizar sus recursos y estar acompañado en el camino.*
  - Se preocupa por la seguridad al llevar pasajeros.*
  - Le motiva la idea de aportar a una movilidad sostenible y más económica*
- ”



## 6.What do they HEAR?

- Comentarios sobre la inseguridad de compartir vehículo con desconocidos.
- Recomendaciones de amigos para usar apps de movilidad o taxis en lugar de ofrecer viajes.
- Opiniones sobre los altos precios del combustible y lo costoso que es mantener un auto.

## 2.What do they need to DO?

- Encontrar pasajeros de confianza con rutas similares.
- Reducir los costos de movilidad diaria.
- Garantizar seguridad al compartir su vehículo.

## 3.What do they SEE?

- Tráfico congestionado en Lima.
- Gastos cada vez más altos en combustible y mantenimiento.
- Opciones de carpooling generalistas que no están adaptadas a estudiantes.

## 5.What do they DO?

- Maneja solo a la universidad casi todos los días.
- Coordina ocasionalmente con amigas/os para compartir gastos, pero de manera informal.
- Usa apps de movilidad, pero como pasajero en viajes puntuales.

## PAINS

What are their fears, frustrations and anxieties?

- Gastos elevados en gasolina y mantenimiento.
- Tráfico estresante y viajes solitarios.
- Riesgo de inseguridad al compartir con desconocidos.

## GAINS

What are their needs, hopes, wants and dreams?

- Reducir costos mediante el carpooling.
- Viajar con estudiantes verificados que transmitan confianza.
- Sentirse parte de una iniciativa sostenible y colaborativa.

## 4.What do they SAY?

- “
- “Si igual voy en auto, ¿por qué no compartir los asientos vacíos?”*
  - “Me gustaría reducir mis gastos y no manejar siempre sola.”*
  - “Quiero una opción segura, sin riesgos de llevar desconocidos.”*
- ”

## Segmento 2: Estudiante sin vehículo

|  |   |   |
|--|---|---|
| <p><b>1.WHO are we empathizing with?</b></p> <p>Un estudiante universitario de 22 años en Lima Metropolitana, soltero, que depende del transporte público para llegar a la universidad. Busca opciones más seguras, económicas y puntuales.</p>  | <p><b>7.What do they THINK and FEEL?</b></p> <p>“</p> <ul style="list-style-type: none"> <li>• Se siente cansado y estresado por los viajes largos.</li> <li>• Tiene miedo a la inseguridad durante los trayectos.</li> <li>• Sueña con un transporte accesible y seguro que lo conecte con su comunidad universitaria.</li> </ul> <p>”</p> | <p><b>2.What do they need to DO?</b></p> <ul style="list-style-type: none"> <li>• Encontrar una alternativa de transporte confiable y asequible.</li> <li>• Tomar decisiones rápidas sobre cómo llegar a clases puntualmente.</li> <li>• Adaptar su rutina para optimizar tiempo y reducir estrés.</li> </ul>         |
| <p><b>6.What do they HEAR?</b></p> <ul style="list-style-type: none"> <li>• Consejos de amigos que también sufren con el tráfico.</li> <li>• Comentarios sobre inseguridad en el transporte público.</li> <li>• Opiniones de conocidos que usan taxis o apps de ride-hailing, aunque con quejas por el costo.</li> </ul> |    | <p><b>3.What do they SEE?</b></p> <ul style="list-style-type: none"> <li>• Un transporte público saturado e inseguro.</li> <li>• Compañeros que usan taxis o apps de movilidad como opción rápida, aunque costosa.</li> <li>• Publicidad de apps de transporte, pero no enfocadas en estudiantes.</li> </ul>          |
| <p><b>5.What do they DO?</b></p> <ul style="list-style-type: none"> <li>• Utiliza buses y combis para movilizarse diariamente.</li> <li>• En emergencias, usa apps como Uber/Didi, pero lo considera caro.</li> <li>• A veces coordina con amigos para ir juntos, pero sin un sistema confiable.</li> </ul>              | <p><b>PAINS</b></p> <p>What are their fears, frustrations and anxieties?</p> <ul style="list-style-type: none"> <li>• Inseguridad en transporte público.</li> <li>• Costos elevados en apps privadas.</li> <li>• Estrés y pérdida de tiempo en trayectos largos.</li> </ul>   | <p><b>GAINS</b></p> <p>What are their needs, hopes, wants and dreams?</p> <ul style="list-style-type: none"> <li>• Viajar acompañado de estudiantes de su universidad.</li> <li>• Reducir gastos y ahorrar tiempo.</li> <li>• Sentirse seguro y respaldado por una comunidad.</li> </ul>                              |
|  |   | <p><b>4.What do they SAY?</b></p> <p>“</p> <ul style="list-style-type: none"> <li>• “El transporte público me hace perder mucho tiempo.”</li> <li>• “No quiero gastar tanto en taxis o apps de transporte.”</li> <li>• “Si existiera una forma segura y más económica, la usaría todos los días.”</li> </ul> <p>”</p> |

### 2.3.5 Ubiquitous Language

| Término      | Definición  |
|--------------|---|
| Ride (Viaje) | Desplazamiento realizado desde un punto de origen hasta un destino. |

| Término                                  | Definición   |
|--|--|
| <b>Carpooling<br/>(Viaje compartido)</b> | Modalidad en la que varios usuarios comparten un mismo trayecto en un vehículo para optimizar costos y reducir tráfico.                    |
| <b>Driver<br/>(Conductor)</b>            | Usuario propietario de un vehículo que ofrece asientos disponibles para compartir.   |
| <b>Passenger<br/>(Pasajero)</b>          | Usuario que busca un asiento disponible en un vehículo para trasladarse a un destino común.  |
| <b>Route (Ruta)</b>                      | Camino definido que conecta el origen y el destino de un viaje.  |
| <b>Match<br/>(Coincidencia)</b>          | Proceso mediante el cual el sistema identifica conductores y pasajeros con rutas compatibles.  |
| <b>Fare (Tarifa)</b>                     | Costo asociado al viaje, que puede ser compartido entre conductor y pasajeros.   |
| <b>Community<br/>(Comunidad)</b>         | Conjunto de usuarios verificados (estudiantes universitarios) que participan en la plataforma.   |
| <b>Safety<br/>(Seguridad)</b>            | Conjunto de medidas y funciones que garantizan la protección del usuario durante el viaje.   |
| <b>Verification<br/>(Verificación)</b>   | Proceso por el cual se valida que un usuario pertenece a la comunidad universitaria antes de acceder a los servicios.                      |
| <b>Trust<br/>(Confianza)</b>             | Percepción positiva que los usuarios tienen sobre la fiabilidad de otros miembros de la comunidad, reforzada por calificaciones o reseñas. |
| <b>Availability<br/>(Disponibilidad)</b> | Oportunidad de encontrar un viaje en un horario y ruta determinados.   |
| <b>Schedule<br/>(Horario)</b>            | Momento específico en el que se realiza un viaje.  |

## 2.4 Requirements specification

### As-is Scenario Mapping

El As-is Scenario Mapping describe la situación actual de los estudiantes universitarios frente a la necesidad de movilizarse hacia la universidad. Este análisis permite identificar los comportamientos, pensamientos y emociones que experimentan tanto los estudiantes que no cuentan con vehículo propio como aquellos que sí lo poseen. En este escenario se evidencian los principales problemas: inseguridad, falta de coordinación, costos elevados y dependencia de medios de transporte informales o poco confiables. Este punto de partida es esencial para comprender las dificultades que ÑanGo busca resolver.

#### As-is Scenario Mapping Estudiantes universitarios que necesitan movilizarse

## Estudiantes universitarios que necesitan movilizarse\_As-Is

| Phase | Identificación de la necesidad de transporte   |  | Búsqueda de opciones disponibles       |  | Solicitud y coordinación del viaje       |   | Realización del traslado              |                           |
|-------|--|--|--|--|--|---|---------------------------------------|---------------------------|
|       | Doing  |  |  |  |  |   |                                       |                           |
|       | Thinking                                       |  |  |  |  |   |                                       |                           |
|       | Se da cuenta de que no tiene movilidad propia. | Pregunta a familiares o amigos si pueden llevarlo. | Pregunta en grupos de WhatsApp.        | Compara precios rápidamente.           | Llama o escribe por WhatsApp.            | Confirma verbalmente con el conductor.  | Se sube al bus, taxi, colectivo.      | Paga en efectivo.         |
|       | Evaluá alternativas de último minuto.          |  | Busca colectivos o taxis aplicativos.  |  | No hay comprobante de reserva.           |   | No hay soporte en caso de incidentes. |                           |
|       | ¿Cómo llegaré hoy?                             | ¿Depender del bus será confiable?                  | ¿Cuál opción es más rápida?            | ¿Qué es lo más barato?                 | ¿Realmente me recogerán?                 | ¿Qué pasa si cancela?                   | ¿Llegaré a tiempo?                    | ¿Será seguro?             |
|       | ¿Me alcanzará para taxi?                       |  | ¿Será seguro?                          |  | ¿Debo confiar en alguien que no conozco? |   | ¿Me cobrarán lo justo?                |                           |
|       | Ansiedad por falta de opciones.                | Preocupación por puntualidad.                      | Duda por información limitada.         | Frustración por falta de alternativas. | Inseguridad por falta de compromiso.     | Estrés al no tener confirmación formal. | Alivio si llega puntual.              | Tensión durante el viaje. |
|       | Estrés por depender de terceros.               |  | Inseguridad en transportes informales. | →                                      | Esperanza de que cumplan.                |   | Resignación ante el riesgo.           |                           |

## As-is Scenario Mapping Estudiantes Universitarios propietarios de un vehículo

| Phase | Planificación del viaje propio          |                             | Publicación de la ruta         |                               | Gestión de solicitudes de pasajeros |                                      | Ejecución del viaje compartido            |                               |
|-------|---|-----------------------------|--------------------------------|-------------------------------|-------------------------------------|--------------------------------------|---|-------------------------------|
|       | Doing                                   |                             |                                |                               |                                     |                                      |   |                               |
|       | Thinking                                |                             |                                |                               |                                     |                                      |   |                               |
|       | Decide llevar su auto a la universidad. | Calcula gasto en gasolina.  | Escribe en grupos de WhatsApp. | Pregunta a amigos cercanos.   | Recibe mensajes individuales.       | Confirma por WhatsApp.               | Recoge a los interesados.                 | Conduce hasta la universidad. |
|       | Piensa si ofrecerá plazas.              |                             | No formaliza horarios.         |                               | No hay control de cupos.            |                                      | Recibe pagos en efectivo (si se acuerda). |                               |
|       | ¿Vale la pena compartir?                | ¿Cuánto gastaré hoy?        | ¿Quién necesita movilidad?     | ¿Será seguro llevarlos?       | ¿Quién llegará a tiempo?            | ¿Me dejarán plantado?                | ¿Llegaremos puntuales?                    | ¿Me darán el aporte acordado? |
|       | ¿Llevar a alguien me convendrá?         |                             | ¿Me darán aporte?              |                               | ¿Habrá problemas de pago?           |                                      | ¿Será un viaje tranquilo?                 |                               |
|       | Preocupación por costos.                | Indecisión sobre pasajeros. | Duda al no tener claridad.     | Confianza solo con conocidos. | Estrés por informalidad.            | Frustración por falta de compromiso. | Inseguridad con pasajeros desconocidos.   | Alivio si llegan bien.        |
|       | Resignación si viaja solo.              |                             | Poca motivación.               | →                             | Incomodidad al hablar de dinero.    |                                      | Molestia si no recibe lo pactado.         |                               |

## To-be Scenario Mapping

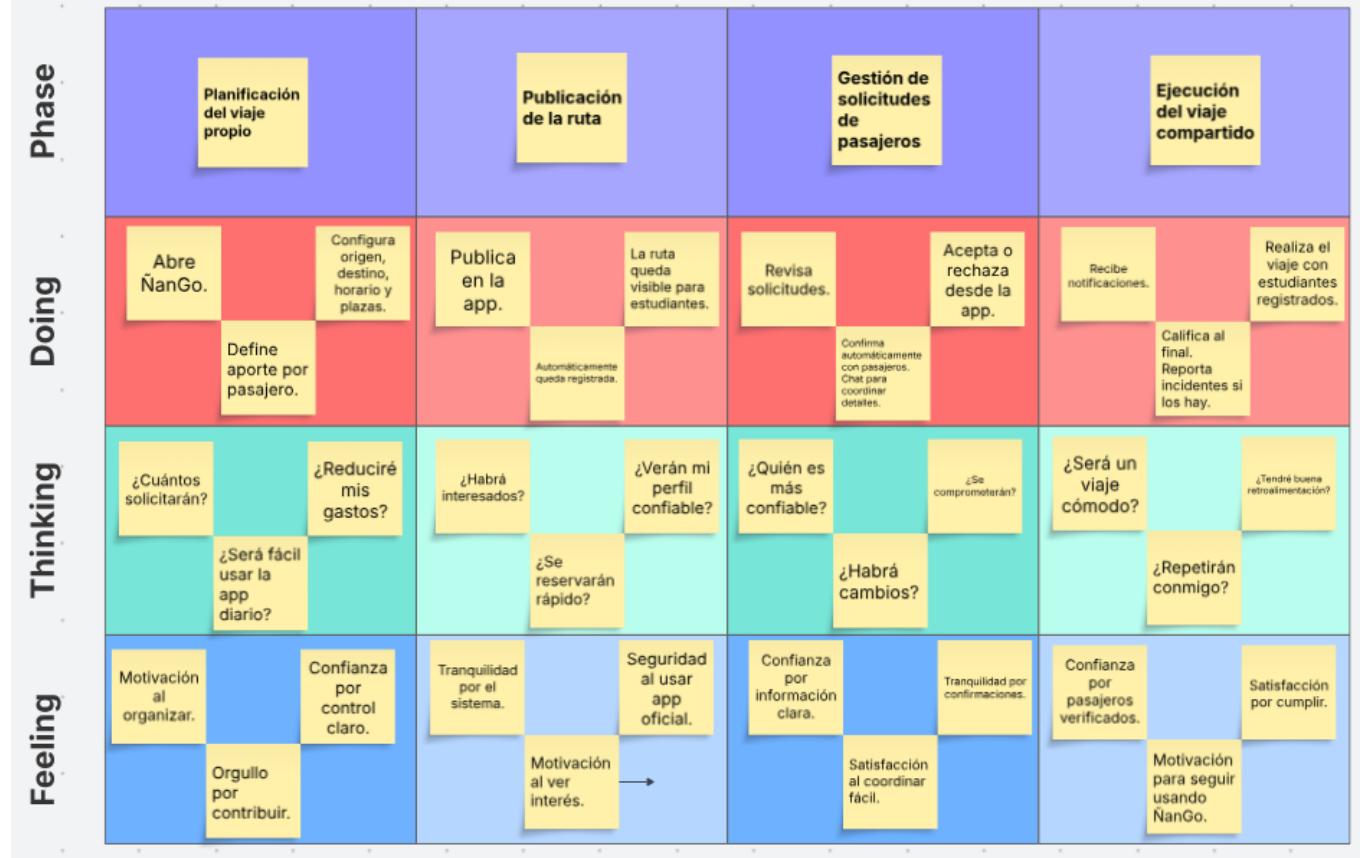
El To-be Scenario Mapping presenta la experiencia proyectada de los estudiantes universitarios al utilizar ÑanGo, una solución digital de transporte compartido. En este escenario futuro, tanto los estudiantes que requieren movilidad como los que poseen vehículo cuentan con una plataforma confiable que les permite planificar, coordinar y realizar viajes de forma segura y eficiente. Con el uso de funcionalidades como publicación de rutas, solicitudes de viaje, confirmaciones automáticas, chat interno y evaluaciones, se optimiza la experiencia de traslado, reduciendo costos y aumentando la confianza de la comunidad universitaria.

### To-be Scenario Mapping Estudiantes universitarios que necesitan movilizarse

| Estudiantes universitarios que necesitan movilizarse_To-Be |  |  |                                  |   |                               |                                    |                                      |                                   |  |
|--|--|--|----------------------------------|---|-------------------------------|------------------------------------|--------------------------------------|-----------------------------------|--|
| Phase  | Identificación de la necesidad de transporte |  | Búsqueda de opciones disponibles |   |                               | Solicitud y coordinación del viaje |                                      | Realización del traslado          |  |
| Doing  | Abre ÑanGo y selecciona horario/día.         | Confirma necesidad de movilidad en la app. | Explora rutas en la app.         | Ve perfil del conductor y calificaciones. | Envía solicitud desde la app. | Recibe confirmación automática.    | Recibe notificación de recordatorio. | Viaja con compañeros registrados. |  |
| Thinking   | ¿Habrá rutas hoy?                            | ¿Se ajusta a mi horario?                   | ¿Cuál es más conveniente?        | ¿Quién tiene mejor reputación?            | ¿Me aceptarán?                | ¿Qué pasa si hay cambios?          | ¿Será cómodo?                        | ¿Todo saldrá como planeado?       |  |
| Feeling  | Tranquilidad por tener opciones claras.      | Seguridad gracias a la plataforma.         | Alivio por varias alternativas.  | Motivación al elegir.                     | Confianza por confirmación.   | Seguridad en la coordinación.      | Relajación en el trayecto.           | Satisfacción por seguridad.       |  |

### To-be Scenario Mapping Estudiantes Universitarios propietarios de un vehículo

## Estudiantes universitarios propietarios de un vehículo\_To-Be



### 2.4.1 User Stories

| Epic / Story ID | Título                              | Descripción  | Criterios de Aceptación | Relacionado con (Epic ID) |
|-----------------|-------------------------------------|--|-------------------------|---------------------------|
| EP01            | Autenticación y gestión de usuarios | Como estudiante universitario quiero registrarme, iniciar sesión y administrar mis datos para poder usar la aplicación de manera segura. | -                       | -                         |
| EP02            | Gestión de rutas y solicitudes      | Como estudiante propietario de vehículo quiero publicar, gestionar rutas y aceptar/rechazar solicitudes para organizar mis viajes.       | -                       | -                         |
| EP03            | Búsqueda y reserva de viajes        | Como estudiante que necesita movilizarse quiero buscar rutas y unirme a viajes para llegar a la universidad de forma segura y económica. | -                       | -                         |

| Epic / Story ID | Título                         | Descripción  | Criterios de Aceptación  | Relacionado con (Epic ID) |
|-----------------|--------------------------------|--|--|---------------------------|
| EP04            | Comunicación y coordinación    | Como estudiante quiero herramientas de comunicación para coordinar detalles y confirmar viajes.                                    | -  | -                         |
| EP05            | Seguridad y confianza          | Como estudiante quiero contar con perfiles verificados, calificaciones y mecanismos de reporte para sentirme seguro en mis viajes. | -  | -                         |
| EP06            | Pagos compartidos              | Como estudiante quiero registrar y consultar los aportes compartidos de los viajes para mantener transparencia.                    | -  | -                         |
| EP07            | Historial y experiencia de uso | Como estudiante quiero consultar viajes pasados y futuros, además de opciones de personalización, para mejorar mi experiencia.     | -  | -                         |
| EP08            | Landing Page y visibilidad     | Como visitante quiero acceder a la información de ÑanGo y descargar la aplicación desde la landing page.                           | -  | -                         |
| EP09            | Integración técnica y APIs     | Como desarrollador quiero exponer y consumir APIs para asegurar el funcionamiento de la app.                                       | -  | -                         |
| EP10            | Seguridad técnica              | Como desarrollador quiero implementar autenticación, validación y cifrado para garantizar la protección de los datos.              | -  | -                         |
| EP11            | Investigación y Spikes         | Como equipo quiero realizar pruebas de integración y tecnologías clave para validar decisiones técnicas.                           | -  | -                         |
| US01            | Registro de cuenta             | Como estudiante que necesita movilizarme quiero registrarme con mi correo institucional para usar la app.                          | <b>Given</b> que ingreso mis datos válidos, <b>When</b> presiono "Registrar", <b>Then</b> se crea mi cuenta. | EP01                      |

| Epic / Story ID | Título                     | Descripción   | Criterios de Aceptación   | Relacionado con (Epic ID) |
|-----------------|----------------------------|---|---|---------------------------|
| US02            | Inicio de sesión           | Como estudiante propietario de vehículo quiero iniciar sesión para acceder a mis rutas publicadas.                | <b>Given</b> credenciales válidas, <b>When</b> presiono "Ingresar", <b>Then</b> accedo a mi perfil.             | EP01                      |
| US03            | Recuperación de contraseña | Como estudiante que necesita movilizarme quiero recuperar mi contraseña para no perder acceso.                    | <b>Given</b> correo válido, <b>When</b> solicito restablecer, <b>Then</b> recibo enlace de recuperación.        | EP01                      |
| US04            | Completar perfil personal  | Como estudiante quiero completar mi perfil con foto y datos básicos para generar confianza.                       | <b>Given</b> formulario lleno, <b>When</b> guardo cambios, <b>Then</b> mi perfil muestra la información.        | EP01                      |
| US05            | Buscar rutas disponibles   | Como estudiante que necesita movilizarme quiero buscar rutas hacia mi universidad para elegir la más conveniente. | <b>Given</b> que ingreso mi destino, <b>When</b> consulto rutas, <b>Then</b> se muestran opciones disponibles.  | EP03                      |
| US06            | Publicar ruta              | Como estudiante propietario de vehículo quiero publicar una ruta con origen, destino, horario y plazas.           | <b>Given</b> formulario completo, <b>When</b> publico la ruta, <b>Then</b> aparece disponible para estudiantes. | EP02                      |
| US07            | Editar ruta publicada      | Como estudiante propietario de vehículo quiero editar los datos de una ruta publicada para actualizar horarios.   | <b>Given</b> ruta publicada, <b>When</b> edito información, <b>Then</b> se actualiza para todos.                | EP02                      |
| US08            | Eliminar ruta publicada    | Como estudiante propietario de vehículo quiero eliminar una ruta publicada para cancelar el servicio.             | <b>Given</b> ruta activa, <b>When</b> presiono eliminar, <b>Then</b> desaparece del listado.                    | EP02                      |
| US09            | Solicitar unirse a ruta    | Como estudiante que necesita movilizarme quiero solicitar unirme a una ruta para asegurar mi traslado.            | <b>Given</b> ruta con plazas, <b>When</b> solicito unirme, <b>Then</b> el propietario recibe notificación.      | EP03                      |

| Epic / Story ID | Título                  | Descripción   | Criterios de Aceptación   | Relacionado con (Epic ID) |
|-----------------|-------------------------|---|---|---------------------------|
| US10            | Aceptar pasajero        | Como estudiante propietario de vehículo quiero aceptar solicitudes de estudiantes para confirmar cupos.           | <b>Given</b> estudiante solicita, <b>When</b> acepto, <b>Then</b> el estudiante recibe confirmación.  | EP02                      |
| US11            | Rechazar pasajero       | Como estudiante propietario de vehículo quiero rechazar solicitudes para mantener control de mi ruta.             | <b>Given</b> estudiante solicita, <b>When</b> rechazo, <b>Then</b> recibe notificación de rechazo.    | EP02                      |
| US12            | Cancelar solicitud      | Como estudiante que necesita movilizarme quiero cancelar mi solicitud de ruta en caso de cambio de planes.        | <b>Given</b> solicitud activa, <b>When</b> presiono cancelar, <b>Then</b> se notifica al propietario. | EP03                      |
| US13            | Chat básico             | Como estudiante quiero un chat básico para coordinar detalles del viaje.  | <b>Given</b> ruta confirmada, <b>When</b> envío mensaje, <b>Then</b> el otro estudiante lo recibe.    | EP04                      |
| US14            | Notificaciones push     | Como estudiante quiero recibir notificaciones de confirmaciones o cambios.  | <b>Given</b> evento relevante, <b>When</b> ocurre, <b>Then</b> recibo notificación.                   | EP04                      |
| US15            | Ver perfil de conductor | Como estudiante que necesita movilizarme quiero ver el perfil del propietario de vehículo para evaluar confianza. | <b>Given</b> ruta publicada, <b>When</b> consulto propietario, <b>Then</b> veo su perfil.             | EP05                      |
| US16            | Ver perfil de pasajero  | Como estudiante propietario de vehículo quiero ver perfil del estudiante solicitante para confirmar confianza.    | <b>Given</b> estudiante solicita, <b>When</b> consulto su perfil, <b>Then</b> veo datos básicos.      | EP05                      |
| US17            | Calificar viaje         | Como estudiante quiero calificar a mi compañero de viaje después de un recorrido.                                 | <b>Given</b> viaje completado, <b>When</b> califico, <b>Then</b> se guarda puntuación.                | EP05                      |
| US18            | Reportar incidente      | Como estudiante quiero reportar un incidente para alertar a la comunidad.   | <b>Given</b> incidente, <b>When</b> envío reporte, <b>Then</b> se registra en el sistema.             | EP05                      |
| US19            | Editar datos de perfil  | Como estudiante quiero editar mis datos personales para mantenerlos actualizados.                                 | <b>Given</b> cambios en mis datos, <b>When</b> guardo, <b>Then</b> se actualizan.                     | EP01                      |

| Epic / Story ID | Título                        | Descripción   | Criterios de Aceptación  | Relacionado con (Epic ID) |
|-----------------|-------------------------------|---|--|---------------------------|
| US20            | Registrar aporte de viaje     | Como estudiante que necesita movilizarme quiero registrar mi aporte económico para el viaje compartido. | <b>Given</b> viaje confirmado,<br><b>When</b> indico mi aporte,<br><b>Then</b> queda registrado.     | EP06                      |
| US21            | Ver listado de aportes        | Como estudiante propietario de vehículo quiero ver el aporte registrado por cada estudiante.            | <b>Given</b> viaje en curso,<br><b>When</b> consulto aportes,<br><b>Then</b> veo detalle.            | EP06                      |
| US22            | Ver historial de viajes       | Como estudiante quiero ver mis viajes pasados para consultar experiencias previas.                      | <b>Given</b> viajes completados, <b>When</b> consulto historial, <b>Then</b> veo listado.            | EP07                      |
| US23            | Ver viajes próximos           | Como estudiante quiero ver mis viajes próximos para organizar mi tiempo.                                | <b>Given</b> viajes confirmados, <b>When</b> consulto próximos,<br><b>Then</b> se muestran en orden. | EP07                      |
| US24            | Verificación básica de cuenta | Como estudiante quiero que mi cuenta se verifique con correo institucional para asegurar autenticidad.  | <b>Given</b> correo válido,<br><b>When</b> registro, <b>Then</b> se valida dominio UPC.edu.          | EP05                      |
| US25            | Cambiar contraseña            | Como estudiante quiero cambiar mi contraseña para mantener seguridad.                                   | <b>Given</b> sesión activa,<br><b>When</b> solicito cambiar,<br><b>Then</b> actualizo contraseña.    | EP01                      |
| US26            | Configurar notificaciones     | Como estudiante quiero activar o desactivar notificaciones.   | <b>Given</b> sesión activa,<br><b>When</b> cambio configuración, <b>Then</b> se guarda preferencia.  | EP07                      |
| US27            | Ver información de ÑanGo      | Como visitante quiero conocer qué es ÑanGo en la landing page.  | <b>Given</b> acceso web,<br><b>When</b> entro, <b>Then</b> veo información básica.                   | EP08                      |
| US28            | Ver beneficios de la app      | Como visitante quiero ver beneficios de usar ÑanGo para motivarme a descargarla.                        | <b>Given</b> landing, <b>When</b> consulto beneficios,<br><b>Then</b> veo listado.                   | EP08                      |
| US29            | Descarga desde la landing     | Como visitante quiero poder descargar la app desde la landing.  | <b>Given</b> landing, <b>When</b> presiono "Descargar",<br><b>Then</b> soy redirigido a la tienda.   | EP08                      |

| Epic / Story ID | Título                      | Descripción   | Criterios de Aceptación  | Relacionado con (Epic ID) |
|-----------------|-----------------------------|---|--|---------------------------|
| US30            | Endpoint de registro        | Como developer quiero exponer un endpoint de registro para crear estudiantes. | <b>Given</b> request válido,<br><b>When</b> POST /register,<br><b>Then</b> devuelve usuario creado.              | EP09                      |
| US31            | Endpoint de login           | Como developer quiero exponer un endpoint de login para iniciar sesión.       | <b>Given</b> credenciales,<br><b>When</b> POST /login,<br><b>Then</b> devuelve token válido.                     | EP09                      |
| US32            | Endpoint de rutas           | Como developer quiero exponer un endpoint de rutas para CRUD de viajes.       | <b>Given</b> request válido,<br><b>When</b> GET/POST rutas,<br><b>Then</b> devuelve listado o confirma creación. | EP09                      |
| US33            | Endpoint de solicitudes     | Como developer quiero exponer un endpoint para gestionar solicitudes.         | <b>Given</b> request válido,<br><b>When</b> POST /solicitudes, <b>Then</b> se guarda solicitud.                  | EP09                      |
| US34            | Endpoint de chat            | Como developer quiero un endpoint para mensajería básica.                     | <b>Given</b> request válido,<br><b>When</b> POST /chat,<br><b>Then</b> se almacena mensaje.                      | EP09                      |
| US35            | Autenticación con JWT       | Como developer quiero que la API use JWT para sesiones seguras.               | <b>Given</b> token válido,<br><b>When</b> consulto recurso,<br><b>Then</b> acceso autorizado.                    | EP10                      |
| US36            | Validación de inputs        | Como developer quiero validar inputs para evitar inyecciones.                 | <b>Given</b> input inválido,<br><b>When</b> envío request,<br><b>Then</b> obtengo error 400.                     | EP10                      |
| US37            | Spike: integrar Google Maps | Como equipo quiero investigar integración de Google Maps para rutas.          | <b>Given</b> documentación,<br><b>When</b> realizo prueba,<br><b>Then</b> registro conclusiones.                 | EP11                      |
| US38            | Spike: notificaciones push  | Como equipo quiero probar librerías para notificaciones push.                 | <b>Given</b> entorno de pruebas, <b>When</b> envío push, <b>Then</b> recibo confirmación.                        | EP11                      |

| Epic / Story ID | Título                        | Descripción  | Criterios de Aceptación  | Relacionado con (Epic ID) |
|-----------------|-------------------------------|--|--|---------------------------|
| US39            | Spike: sistema de pagos       | Como equipo quiero analizar pasarelas de pago para aportes compartidos.                                    | <b>Given</b> investigación, <b>When</b> comparo opciones, <b>Then</b> documento resultados.            | EP11                      |
| US40            | Spike: encriptación básica    | Como equipo quiero probar encriptación de contraseñas.   | <b>Given</b> password, <b>When</b> aplico hash, <b>Then</b> se almacena seguro.                        | EP11                      |
| US41            | Recordar sesión               | Como estudiante quiero que la app recuerde mi sesión para no loguearme siempre.                            | <b>Given</b> sesión previa, <b>When</b> abro app, <b>Then</b> entro automáticamente.                   | EP07                      |
| US42            | Modo oscuro                   | Como estudiante quiero un modo oscuro para mejor experiencia.  | <b>Given</b> sesión activa, <b>When</b> activo modo oscuro, <b>Then</b> la interfaz cambia.            | EP07                      |
| US43            | Filtrar rutas por horario     | Como estudiante que necesita movilizarme quiero filtrar rutas por horario para ahorrar tiempo.             | <b>Given</b> horarios, <b>When</b> aplico filtro, <b>Then</b> solo se muestran rutas dentro del rango. | EP03                      |
| US44            | Filtrar rutas por plazas      | Como estudiante que necesita movilizarme quiero filtrar rutas según plazas disponibles.                    | <b>Given</b> rutas con plazas, <b>When</b> aplico filtro, <b>Then</b> veo solo las que tienen espacio. | EP03                      |
| US45            | Cerrar sesión                 | Como estudiante quiero cerrar sesión cuando lo necesite.   | <b>Given</b> sesión activa, <b>When</b> presiono cerrar sesión, <b>Then</b> se termina mi sesión.      | EP01                      |
| US46            | Feedback en la app            | Como estudiante quiero enviar feedback para mejorar la app.  | <b>Given</b> formulario, <b>When</b> lo lleno, <b>Then</b> se envía al equipo.                         | EP07                      |
| US47            | Bloquear estudiante reportado | Como estudiante propietario de vehículo quiero bloquear estudiantes reportados para proteger la comunidad. | <b>Given</b> reporte válido, <b>When</b> bloquee, <b>Then</b> el usuario no accede más.                | EP05                      |

## 2.4.2 Impact Mapping



## 2.4.3 Product Backlog

| Orden | User Story Id | Título             | Descripción   | Story Points |
|-------|---------------|--------------------|---|--------------|
| 1     | US01          | Registro de cuenta | Como estudiante que necesita movilizarme quiero registrarme con mi correo institucional para usar la app. | 3            |
| 2     | US02          | Inicio de sesión   | Como estudiante propietario de vehículo quiero iniciar sesión para acceder a mis rutas publicadas.        | 3            |

| <b>Orden</b> | <b>User Story Id</b> | <b>Título</b>                 | <b>Descripción</b>  | <b>Story Points</b> |
|--------------|----------------------|-------------------------------|---|---------------------|
| 3            | US03                 | Recuperación de contraseña    | Como estudiante que necesita movilizarme quiero recuperar mi contraseña para no perder acceso.                    | 2                   |
| 4            | US06                 | Publicar ruta                 | Como estudiante propietario de vehículo quiero publicar una ruta con origen, destino, horario y plazas.           | 5                   |
| 5            | US05                 | Buscar rutas disponibles      | Como estudiante que necesita movilizarme quiero buscar rutas hacia mi universidad para elegir la más conveniente. | 5                   |
| 6            | US09                 | Solicitar unirse a ruta       | Como estudiante que necesita movilizarme quiero solicitar unirme a una ruta para asegurar mi traslado.            | 5                   |
| 7            | US10                 | Aceptar pasajero              | Como estudiante propietario de vehículo quiero aceptar solicitudes de estudiantes para confirmar cupos.           | 3                   |
| 8            | US11                 | Rechazar pasajero             | Como estudiante propietario de vehículo quiero rechazar solicitudes para mantener control de mi ruta.             | 2                   |
| 9            | US12                 | Cancelar solicitud            | Como estudiante que necesita movilizarme quiero cancelar mi solicitud de ruta en caso de cambio de planes.        | 2                   |
| 10           | US15                 | Ver perfil de conductor       | Como estudiante que necesita movilizarme quiero ver el perfil del propietario de vehículo para evaluar confianza. | 3                   |
| 11           | US16                 | Ver perfil de pasajero        | Como estudiante propietario de vehículo quiero ver perfil del estudiante solicitante para confirmar confianza.    | 3                   |
| 12           | US13                 | Chat básico                   | Como estudiante quiero un chat básico para coordinar detalles del viaje.  | 5                   |
| 13           | US14                 | Notificaciones push           | Como estudiante quiero recibir notificaciones de confirmaciones o cambios.  | 3                   |
| 14           | US17                 | Calificar viaje               | Como estudiante quiero calificar a mi compañero de viaje después de un recorrido.                                 | 3                   |
| 15           | US18                 | Reportar incidente            | Como estudiante quiero reportar un incidente para alertar a la comunidad.   | 5                   |
| 16           | US04                 | Completar perfil personal     | Como estudiante quiero completar mi perfil con foto y datos básicos para generar confianza.                       | 2                   |
| 17           | US19                 | Editar datos de perfil        | Como estudiante quiero editar mis datos personales para mantenerlos actualizados.                                 | 2                   |
| 18           | US24                 | Verificación básica de cuenta | Como estudiante quiero que mi cuenta se verifique con correo institucional para asegurar autenticidad.            | 3                   |

| <b>Orden</b> | <b>User Story Id</b> | <b>Título</b>                 | <b>Descripción</b>   | <b>Story Points</b> |
|--------------|----------------------|-------------------------------|--|---------------------|
| 19           | US20                 | Registrar aporte de viaje     | Como estudiante que necesita movilizarme quiero registrar mi aporte económico para el viaje compartido.    | 5                   |
| 20           | US21                 | Ver listado de aportes        | Como estudiante propietario de vehículo quiero ver el aporte registrado por cada estudiante.               | 3                   |
| 21           | US22                 | Ver historial de viajes       | Como estudiante quiero ver mis viajes pasados para consultar experiencias previas.                         | 2                   |
| 22           | US23                 | Ver viajes próximos           | Como estudiante quiero ver mis viajes próximos para organizar mi tiempo.                                   | 2                   |
| 23           | US25                 | Cambiar contraseña            | Como estudiante quiero cambiar mi contraseña para mantener seguridad.                                      | 2                   |
| 24           | US26                 | Configurar notificaciones     | Como estudiante quiero activar o desactivar notificaciones.  | 2                   |
| 25           | US27                 | Ver información de ÑanGo      | Como visitante quiero conocer qué es ÑanGo en la landing page.   | 3                   |
| 26           | US28                 | Ver beneficios de la app      | Como visitante quiero ver beneficios de usar ÑanGo para motivarme a descargarla.                           | 2                   |
| 27           | US29                 | Descarga desde la landing     | Como visitante quiero poder descargar la app desde la landing.   | 2                   |
| 28           | US41                 | Recordar sesión               | Como estudiante quiero que la app recuerde mi sesión para no loguearme siempre.                            | 2                   |
| 29           | US42                 | Modo oscuro                   | Como estudiante quiero un modo oscuro para mejor experiencia.  | 3                   |
| 30           | US43                 | Filtrar rutas por horario     | Como estudiante que necesita movilizarme quiero filtrar rutas por horario para ahorrar tiempo.             | 3                   |
| 31           | US44                 | Filtrar rutas por plazas      | Como estudiante que necesita movilizarme quiero filtrar rutas según plazas disponibles.                    | 2                   |
| 32           | US45                 | Cerrar sesión                 | Como estudiante quiero cerrar sesión cuando lo necesite.   | 1                   |
| 33           | US46                 | Feedback en la app            | Como estudiante quiero enviar feedback para mejorar la app.  | 2                   |
| 34           | US47                 | Bloquear estudiante reportado | Como estudiante propietario de vehículo quiero bloquear estudiantes reportados para proteger la comunidad. | 3                   |

| <b>Orden</b> | <b>User Story Id</b> | <b>Título</b>               | <b>Descripción</b>  | <b>Story Points</b> |
|--------------|----------------------|-----------------------------|---|---------------------|
| 35           | US07                 | Editar ruta publicada       | Como estudiante propietario de vehículo quiero editar los datos de una ruta publicada para actualizar horarios. | 3                   |
| 36           | US08                 | Eliminar ruta publicada     | Como estudiante propietario de vehículo quiero eliminar una ruta publicada para cancelar el servicio.           | 3                   |
| 37           | US30                 | Endpoint de registro        | Como developer quiero exponer un endpoint de registro para crear estudiantes.                                   | 5                   |
| 38           | US31                 | Endpoint de login           | Como developer quiero exponer un endpoint de login para iniciar sesión.   | 3                   |
| 39           | US32                 | Endpoint de rutas           | Como developer quiero exponer un endpoint de rutas para CRUD de viajes.   | 5                   |
| 40           | US33                 | Endpoint de solicitudes     | Como developer quiero exponer un endpoint para gestionar solicitudes.   | 5                   |
| 41           | US34                 | Endpoint de chat            | Como developer quiero un endpoint para mensajería básica.   | 5                   |
| 42           | US35                 | Autenticación con JWT       | Como developer quiero que la API use JWT para sesiones seguras.   | 5                   |
| 43           | US36                 | Validación de inputs        | Como developer quiero validar inputs para evitar inyecciones.   | 3                   |
| 44           | US37                 | Spike: integrar Google Maps | Como equipo quiero investigar integración de Google Maps para rutas.  | 3                   |
| 45           | US38                 | Spike: notificaciones push  | Como equipo quiero probar librerías para notificaciones push.   | 3                   |
| 46           | US39                 | Spike: sistema de pagos     | Como equipo quiero analizar pasarelas de pago para aportes compartidos.   | 5                   |
| 47           | US40                 | Spike: encriptación básica  | Como equipo quiero probar encriptación de contraseñas.  | 2                   |

## 2.5 Strategic-Level Domain-Driven Design

### 2.5.1 EventStorming

Una forma de poder entender a profundidad el dominio de negocio y poder definir, encontrar escenarios, reglas de negocio y posibles eventos, es la realización de un Eventstorming. Esta actividad se basa en la realización de una reunión que hubo con el equipo de aproximadamente 1-2 horas, en donde al principio, se

realiza una lluvia de ideas acerca de lo que podría pasar cuando un usuario utiliza la aplicación móvil. El enfoque se centró en capturar la mayor cantidad de conocimiento del dominio desde una perspectiva colaborativa, permitiendo así una primera aproximación al modelo general del negocio. Utilizamos Miro para realizar esta actividad

Link del Miro:

[https://miro.com/welcomeonboard/bXZGMTYzS2FOK1R5d2svRjdIdmFOT1d0ejVCZ00raFVHZHdJZWZISjRyUUJLVHNENEJ1VGppWlIMV2FITkQwK24ycWtpdjVZc2pkUFV4Qy9MYUhYY3NMVDJJZkppWIFFakFnMFdITWh1SWVjQ2xnQTNSc3pqK21xL1BWVnJyVEIQdGo1ZEV3bUdPQWRZUHQzSGI6V2NBPT0hdjE=?share\\_link\\_id=486650419475](https://miro.com/welcomeonboard/bXZGMTYzS2FOK1R5d2svRjdIdmFOT1d0ejVCZ00raFVHZHdJZWZISjRyUUJLVHNENEJ1VGppWlIMV2FITkQwK24ycWtpdjVZc2pkUFV4Qy9MYUhYY3NMVDJJZkppWIFFakFnMFdITWh1SWVjQ2xnQTNSc3pqK21xL1BWVnJyVEIQdGo1ZEV3bUdPQWRZUHQzSGI6V2NBPT0hdjE=?share_link_id=486650419475)

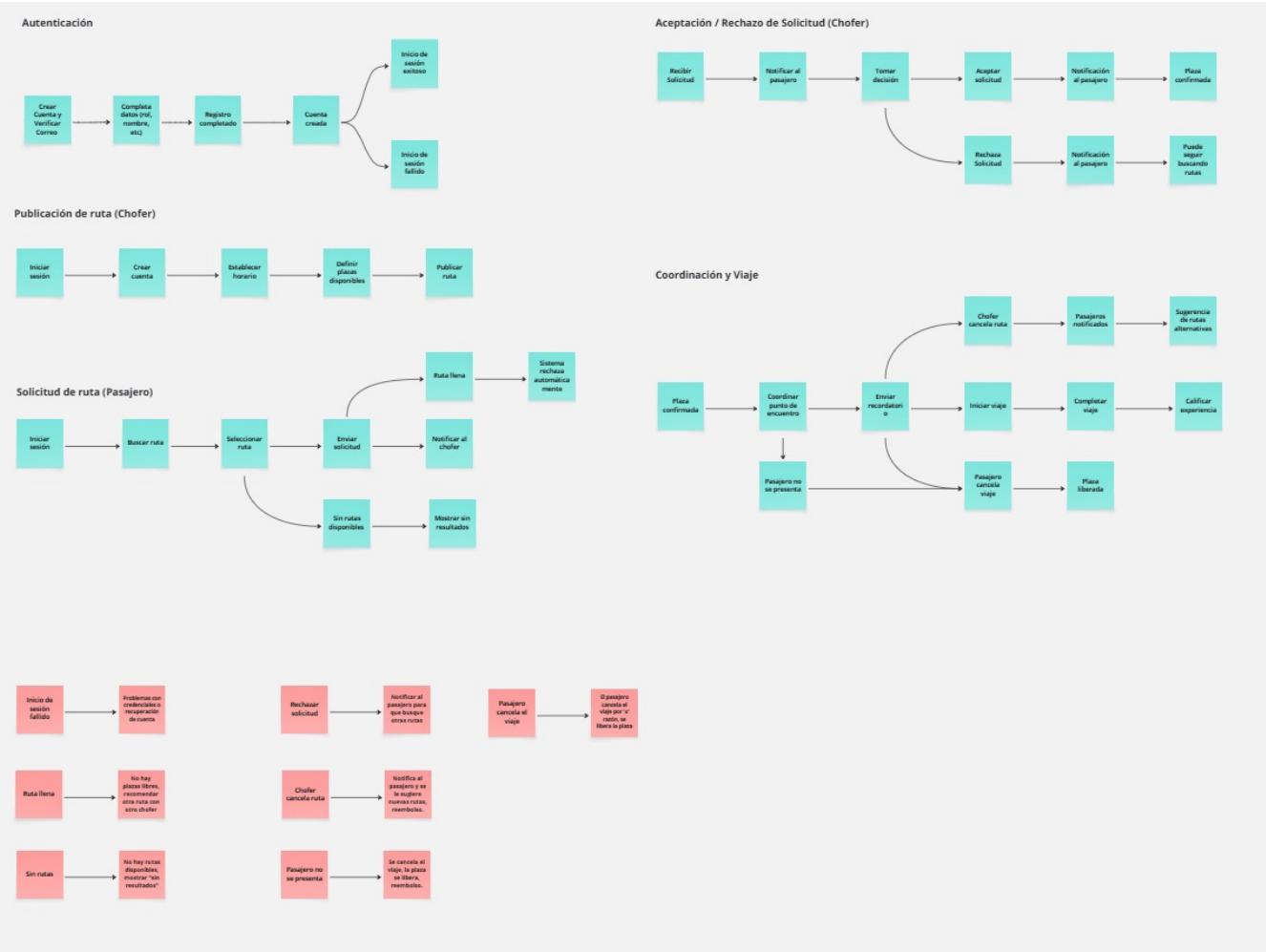
Step 1: Unstructured Exploration

Lluvia de ideas con eventos importantes dentro del dominio de Ñango:



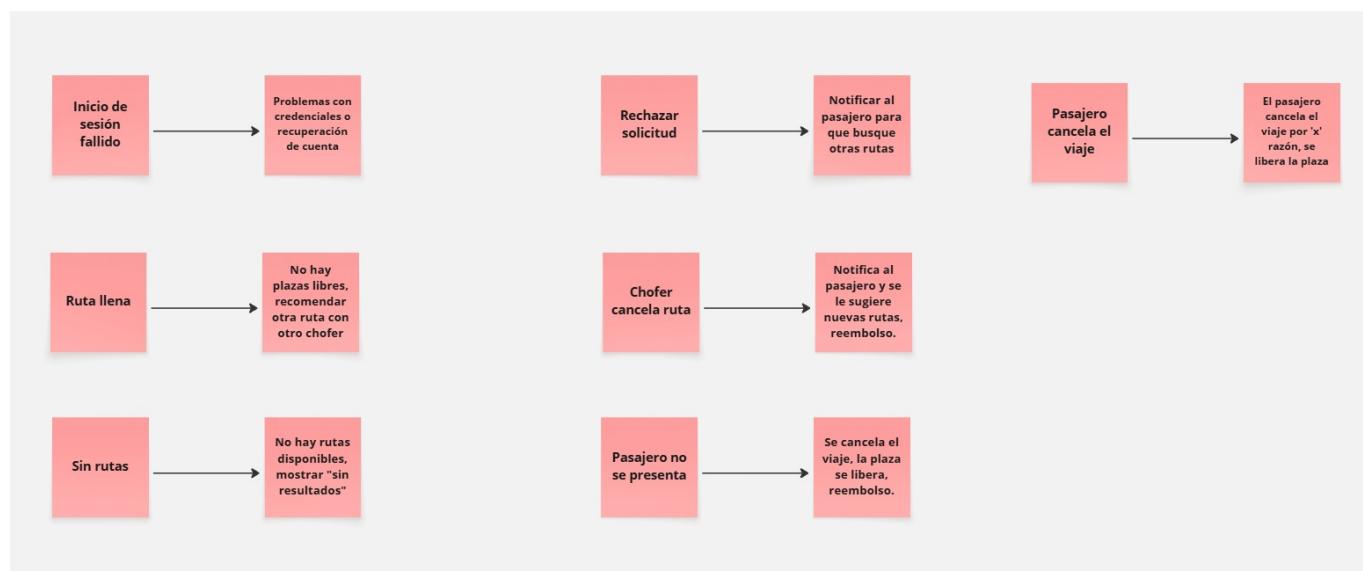
Step 2: Timelines

Organización de las ideas del paso 1, armando posibles líneas de flujo.



### Step 3: Paint Points

Los "puntos de dolor" son los eventos conflictivos que den hagan que la experiencia de usuario sea negativa.

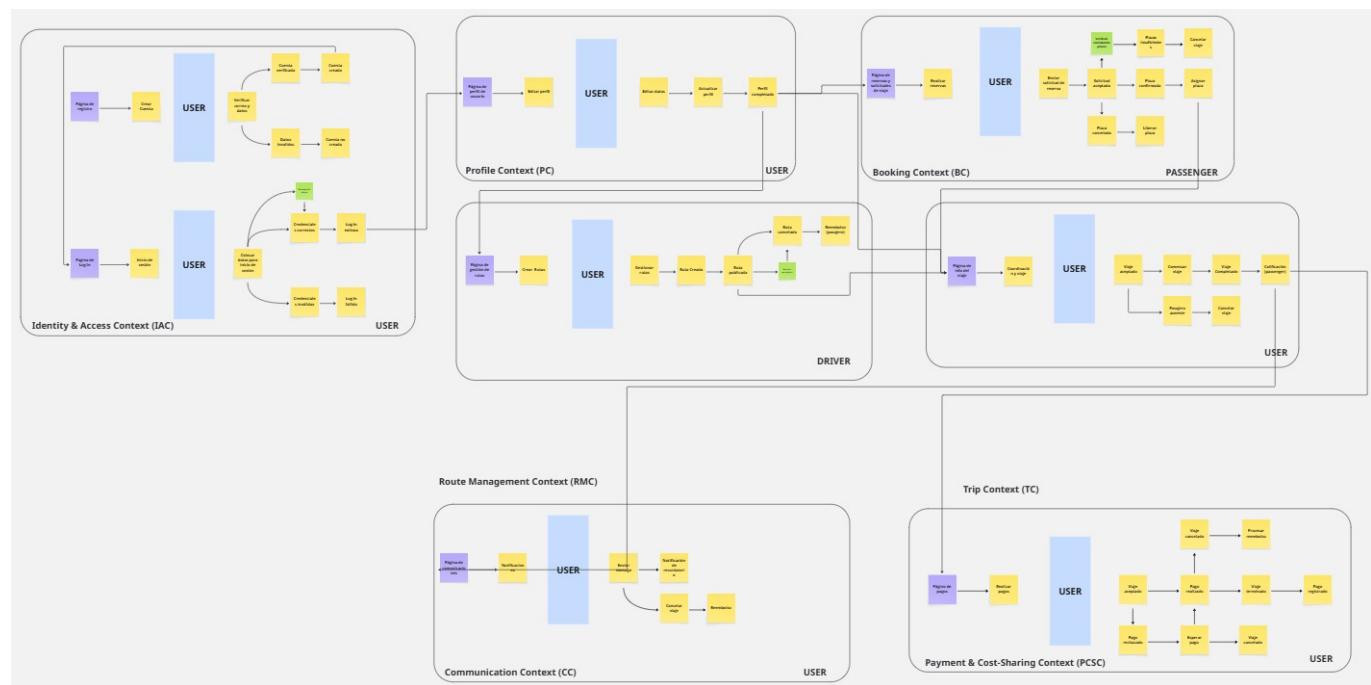


#### 2.5.1.1 Candidate Bounded Contexts

Aquí definimos los posibles bounded context que tendrá nuestra aplicación en el futuro, destacamos siete de estos:

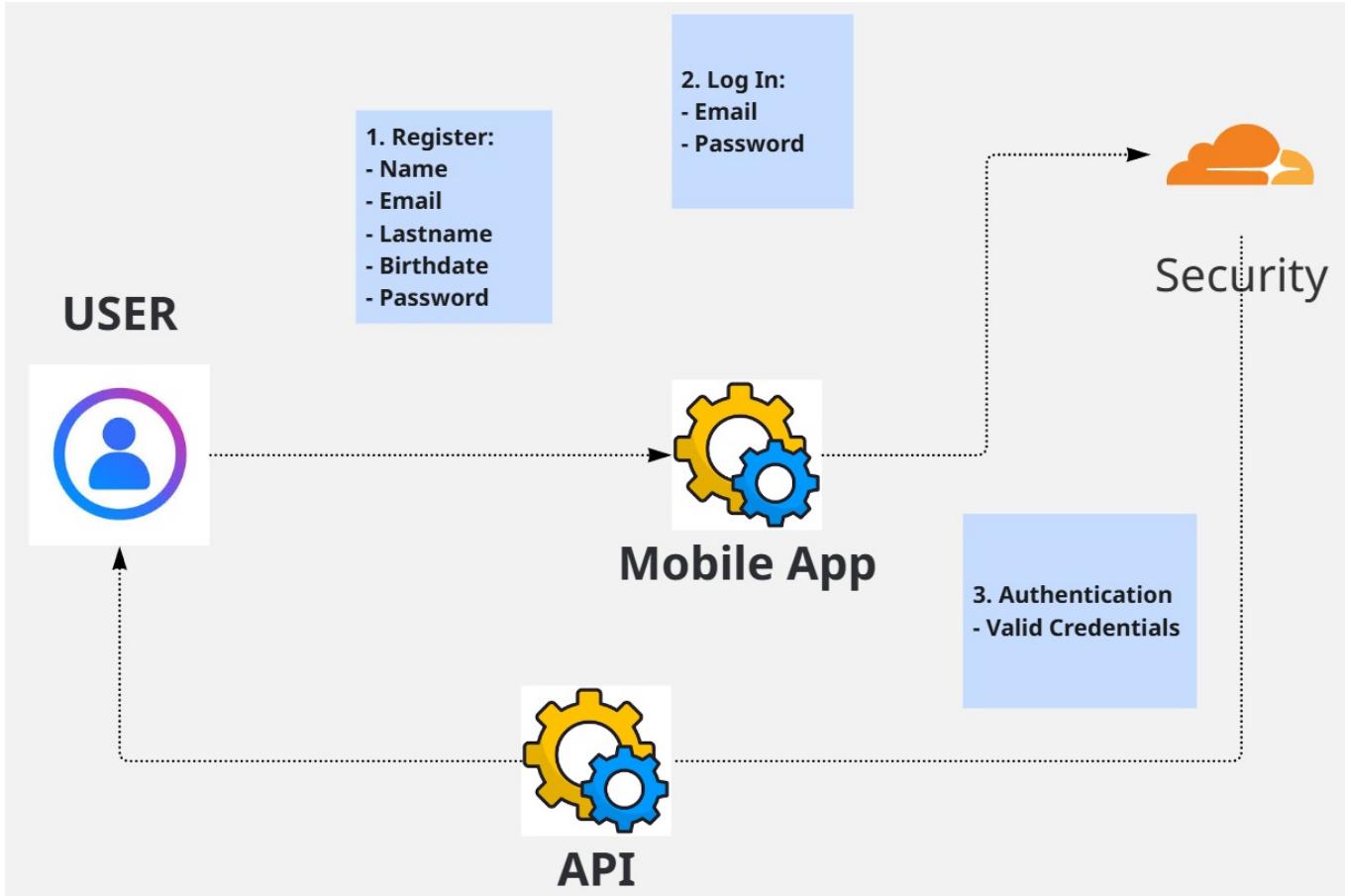
- Identify & Access Context: Encargado del registro, log in y validación de credenciales de usuarios.
- Profile Context: Se encargará de permitirle al usuario que pueda editar su perfil y cambiar cuando quiera de rol (de chofer a pasajero o viceversa).
- Route Management Context: Los choferes podrán crear rutas, publicar rutas, cancelar rutas y gestionar en general estas.
- Booking Context: Los pasajeros podrán pedir reservas para realizar viajes a sus destinos, permitirá enviar solicitudes de viaje, verificar si hay plazas para un viaje, etc.
- Trip Context: Página que se encargará de funcionar únicamente cuando el viaje este sucediendo, permitirá calificar el viaje, ver el proceso del viaje y coordinar el viaje.
- Communication Context: Página de comunicaciones durante el viaje, permitirá a los usuarios enviar mensajes entre sí, cancelar el viaje y notificar recordatorios.
- Payment & Cost-Sharing Context: Página de pagos que se encargará de verificar que el pasajero realiza el pago de parte del usuario, antes de que inicie el viaje.

Flujo general: Mostraremos la interacción y la conexión entre bounded contexts.

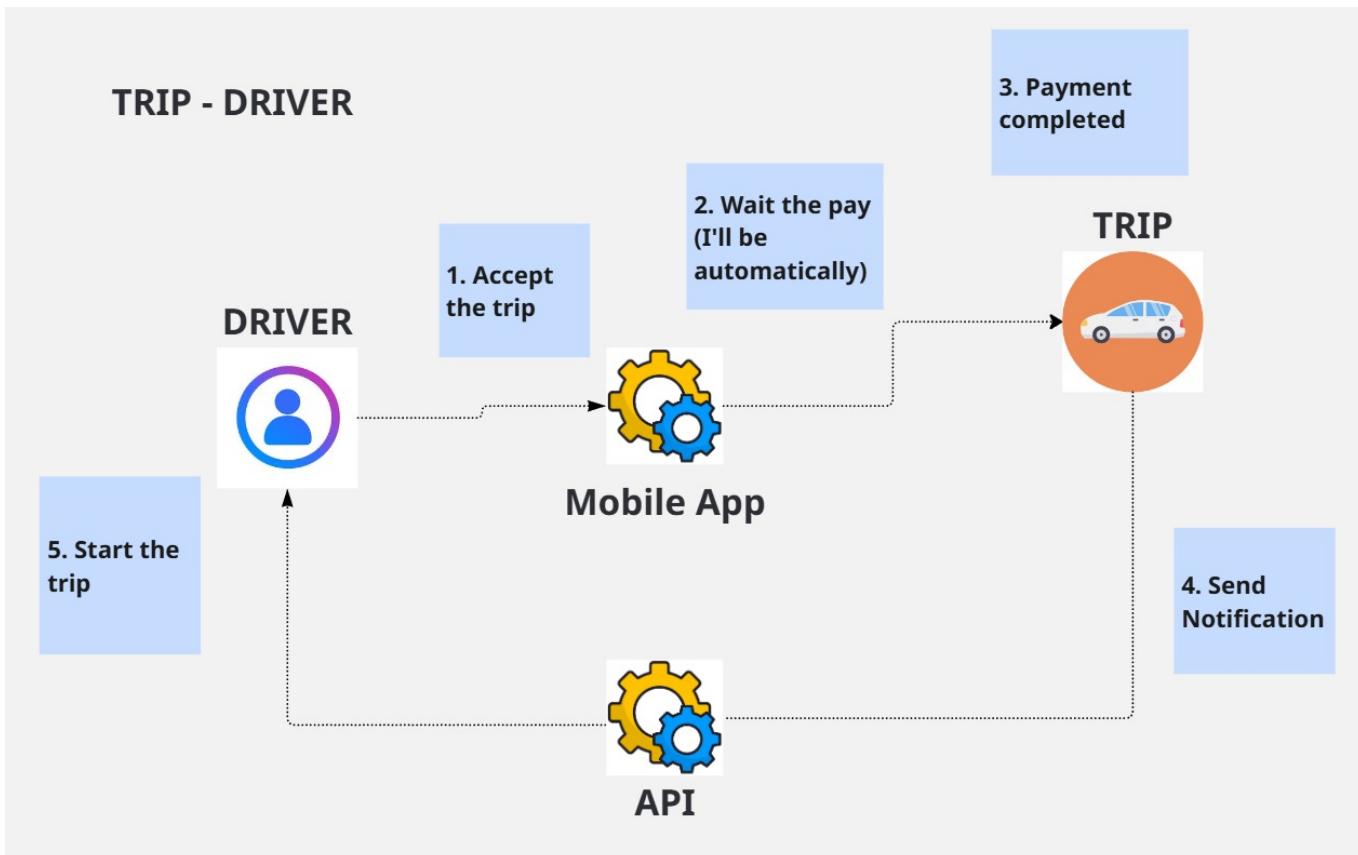


### 2.5.1.2 Domain Message Flows Modeling

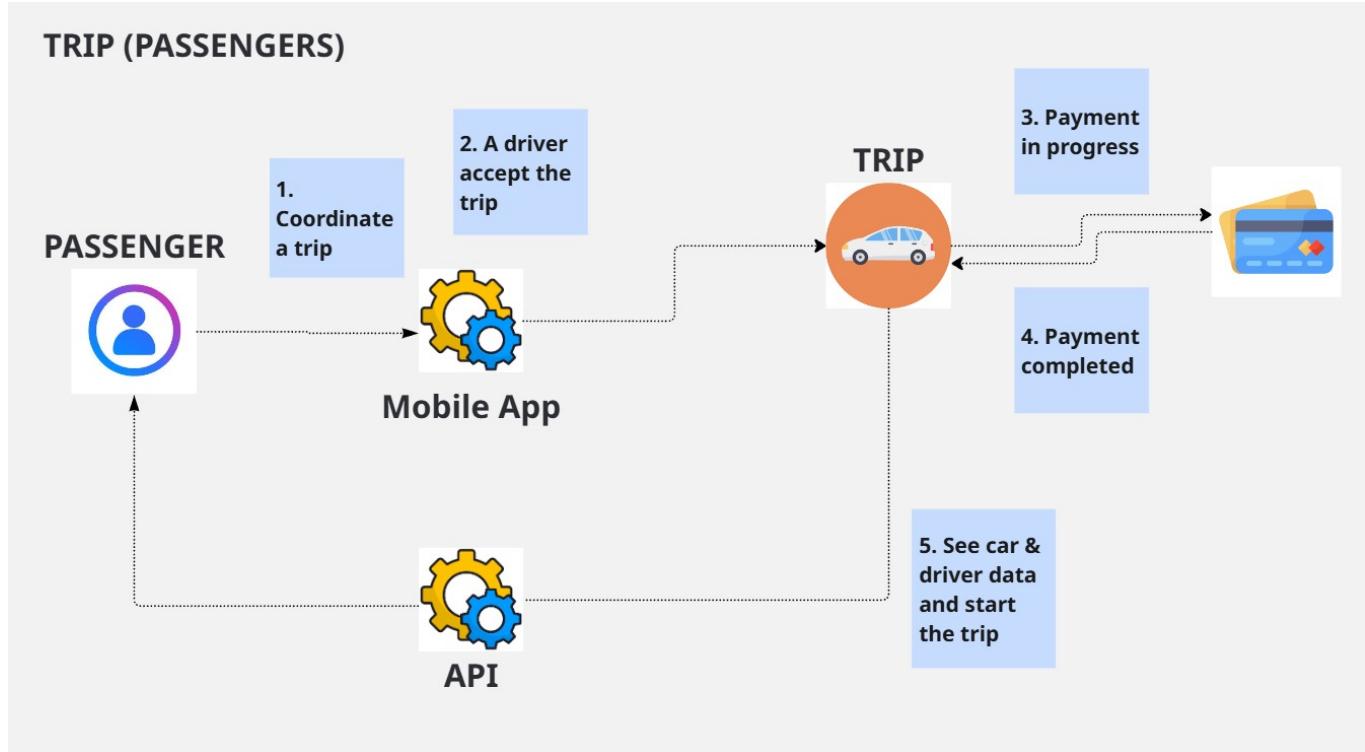
Security: Verificar que los datos del usuario estén seguros y que sus credenciales sean correctas a la hora de logearse.



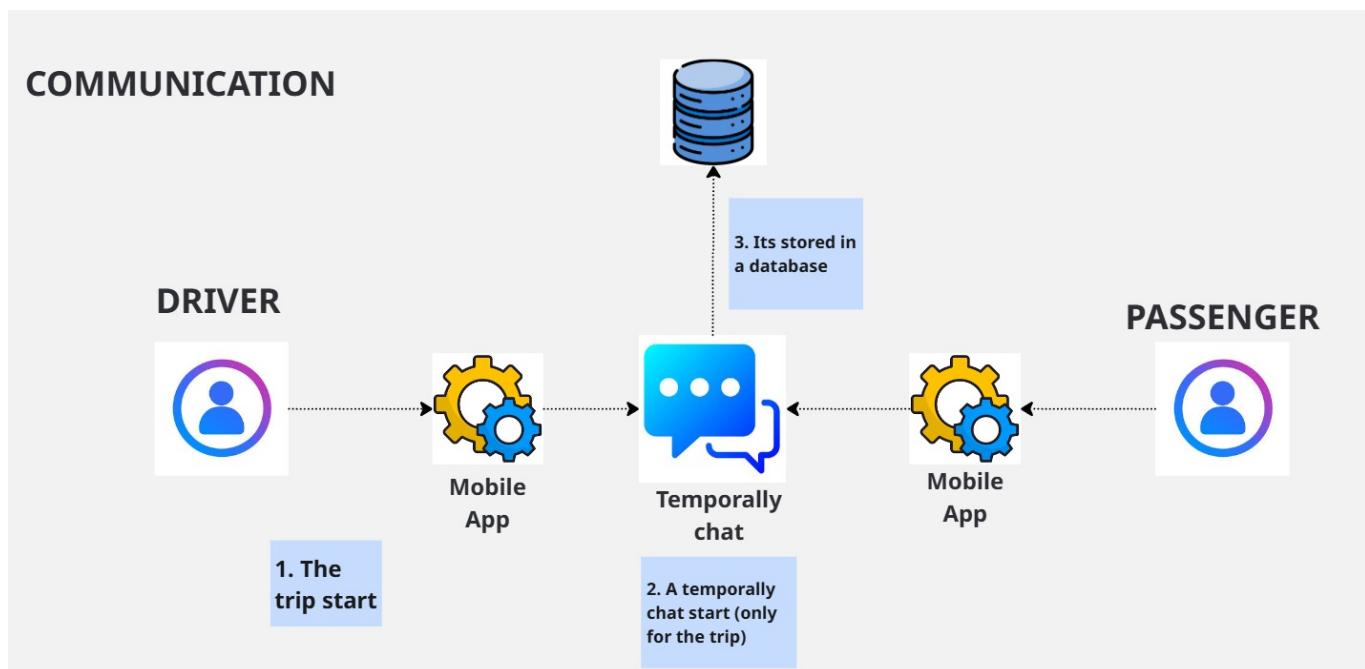
Trip - Driver: El proceso que seguirá el chofér cuando inicie una ruta.



Trip - Passenger: El proceso que seguirá un pasajero cuando necesite reservar un viaje.



Communication: Comunicación que tendrán los usuarios durante el viaje.



#### 2.5.1.3 Bounded Context Canvases

Identify & Access Context:

| <b>NOMBRE: IDENTIFY AND ACCESS CONTEXT</b>   |   |  |
|--|---|--|
| <p><b>Context Overview:</b><br/>Maneja la lógica de el registro y logeo de usuarios, validando las credenciales para poder ingresar a la app</p>   | <p><b>BUSINESS ROLES:</b></p> <div style="background-color: #ffffcc; padding: 5px; display: inline-block;">           El correo de usuario debe de ser único         </div> <div style="background-color: #ffffcc; padding: 5px; display: inline-block;">           Solo usuarios verificados pueden publicar rutas.         </div> |  |
| <p><b>CAPABILITIES:</b></p> <ul style="list-style-type: none"> <li>- Registrar usuarios</li> <li>- Log In de usuarios</li> <li>- Verificar cuentas</li> <li>- Validad credenciales</li> </ul>  | <p><b>UBIQUITOUS LANGUAGE:</b></p> <ul style="list-style-type: none"> <li>- Cuenta de Usuario</li> <li>- Credenciales</li> <li>- Log In</li> <li>- Registrar</li> <li>- Verificar</li> <li>- Recuperación de contraseñas</li> </ul>   |  |
| <p><b>DEPENDENCIES:</b></p> <ul style="list-style-type: none"> <li>- Ninguno, no depende de otro Bounded context</li> </ul>  |   |  |
| <p><b>DESIGN CRITIQUE:</b></p> <div style="background-color: #ffffcc; padding: 5px; display: inline-block;">           Pensar en posibles roles futuros         </div> <div style="background-color: #ffffcc; padding: 5px; display: inline-block;">           Pensar en dependencias externas críticas (Google, Facebook, etc).         </div> <div style="background-color: #ffffcc; padding: 5px; display: inline-block;">           Datos seguros y centralizados         </div> |   |  |

Profile Context:

# NOMBRE: PROFILE CONTEXT

|  |   |
|--|---|
| <p><b>Context Overview:</b><br/>Maneja la lógica de personalización del usuario (nombre, apellido, edad, etc)</p>  | <p><b>BUSINESS ROLES:</b></p> <ul style="list-style-type: none"> <li>El usuario solo puede tener un rol activo</li> <li>El usuario puede cambiar de rol cuando desee</li> <li>El usuario debe colocar datos realistas (ejem. edad &lt; 70)</li> </ul> |
| <p><b>CAPABILITIES:</b></p> <ul style="list-style-type: none"> <li>- Editar perfil de usuario</li> <li>- Cambiar de rol</li> <li>- Eliminar cuenta</li> </ul>  | <p><b>UBIQUITOUS LANGUAGE:</b></p> <ul style="list-style-type: none"> <li>- Perfil</li> <li>- Usuario</li> <li>- Rol</li> <li>- Verificación</li> <li>- Cuenta</li> <li>- Estado de perfil</li> <li>- Datos visibles</li> </ul>                       |
| <p><b>DEPENDENCIES:</b></p> <ul style="list-style-type: none"> <li>- Identity and Access Context</li> </ul>  |   |
| <p><b>DESIGN CRITIQUE:</b></p> <ul style="list-style-type: none"> <li>Pensar en que el usuario hoy puede ser pasajero, y mañana conductor</li> <li>Pensar en que los datos del usuario deben de ser coherentes</li> <li>Los datos deben de ser confidenciales y algunos deberán de ser privados</li> </ul> |   |

Route Management Context:

| <b>NOMBRE: ROUTE MANAGEMENT CONTEXT</b>   |  |  |   |
|---|--|--|---|
| <b>Context Overview:</b><br>El chofer podrá crear rutas y gestionarlas para realizar servicios a pasajeros.                                 | <b>BUSINESS ROLES:</b>   |  |   |
|   | El chofer solo puede tener una ruta activa   | Un viaje solo puede iniciarse si al menos un pasajero está confirmado. | Debe de haber al menos un pasajero para iniciar la ruta |
|   |  |  |   |
| <b>CAPABILITIES:</b>  | <b>UBIQUITOUS LANGUAGE:</b>  |  |   |
| <ul style="list-style-type: none"> <li>- Crear ruta</li> <li>- Gestionar ruta</li> <li>- Publicar ruta</li> <li>- Ruta cancelada</li> </ul> | <ul style="list-style-type: none"> <li>- Conductor</li> <li>- Ruta</li> <li>- Gestionar</li> <li>- Cancelar</li> <li>- Pasajero</li> </ul> |  |   |
| <b>DEPENDENCIES:</b>  |  |  |   |
| <ul style="list-style-type: none"> <li>- Profile Context</li> <li>- Booking Context</li> </ul>  |  |  |   |
| <b>DESIGN CRITIQUE:</b>   | Escabilidad de búsqueda  | Políticas de modificación/cancelación                                  | Dependencia de verificación del chofer                  |

Booking Context:

| <b>NOMBRE: BOOKING CONTEXT</b>   |  |                                   |  |
|--|--|-----------------------------------|--|
| <b>Context Overview:</b><br>Encargado de dar la función para el pasajero de pedir un viaje   | <b>BUSINESS ROLES:</b>   |                                   |  |
|  | Solo puede haber un booking activo   | Deben de haber plazas disponibles | Un pasajero no puede unirse dos veces a la misma ruta. |
| <b>CAPABILITIES:</b><br><ul style="list-style-type: none"><li>- Pedir viaje</li><li>- Buscar viaje</li><li>- Cancelar búsqueda</li></ul> | <b>UBIQUITOUS LANGUAGE:</b> <ul style="list-style-type: none"><li>- Pasajero</li><li>- Reserva</li><li>- Viaje</li><li>- Búsqueda</li><li>- Cancelar</li><li>- Conductor</li><li>- Plaza</li></ul> |                                   |  |
| <b>DEPENDENCIES:</b> <ul style="list-style-type: none"><li>- Profile Context</li><li>- Route Management Context</li></ul>                |  |                                   |  |
| <b>DESIGN CRITIQUE:</b>  | Consistencia de plazas disponibles   | Política de cancelaciones         | Estado de la reserva                                   |

Trip Context:

| <b>NOMBRE: TRIP CONTEXT</b>   |  |   |  |                     |
|---|--|---|--|---------------------|
| <b>Context Overview:</b><br>Operaciones que ocurrirán mientras el viaje esta en proceso, coordinación entre chofer y pasajero.  | <b>BUSINESS ROLES:</b>   |   |  |                     |
|   | Debe de haber al menos un pasajero y chofer en el viaje  | Ambos usuarios deben de estar verificados                   | Si un pasajero cancela, se libera automáticamente una plaza. |                     |
| <b>CAPABILITIES:</b><br><ul style="list-style-type: none"><li>- Aceptar viaje</li><li>- Comenzar viaje</li><li>- Completar viaje<ul style="list-style-type: none"><li>- Calificación</li><li>- Cancelar viaje</li></ul></li></ul> | <b>UBIQUITOUS LANGUAGE:</b><br><ul style="list-style-type: none"><li>- Viaje</li><li>- Coordinación</li><li>- Calificación</li><li>- Cancelar viaje</li><li>- Aceptar viaje</li><li>- Pasajero ausente</li></ul> |   |  |                     |
| <b>DEPENDENCIES:</b><br><ul style="list-style-type: none"><li>- Route Management Context</li><li>- Booking Context</li><li>- Profile Context</li></ul>  |  |   |  |                     |
| <b>DESIGN CRITIQUE:</b>   | Trip empieza solo cuando el viaje esta en EJECUCIÓN  | Inicio y finalización de viaje solo para chofer y pasajeros | Comunicación   | GPS en tiempo real. |

Communication Context:

# NOMBRE: COMMUNICATION CONTEXT

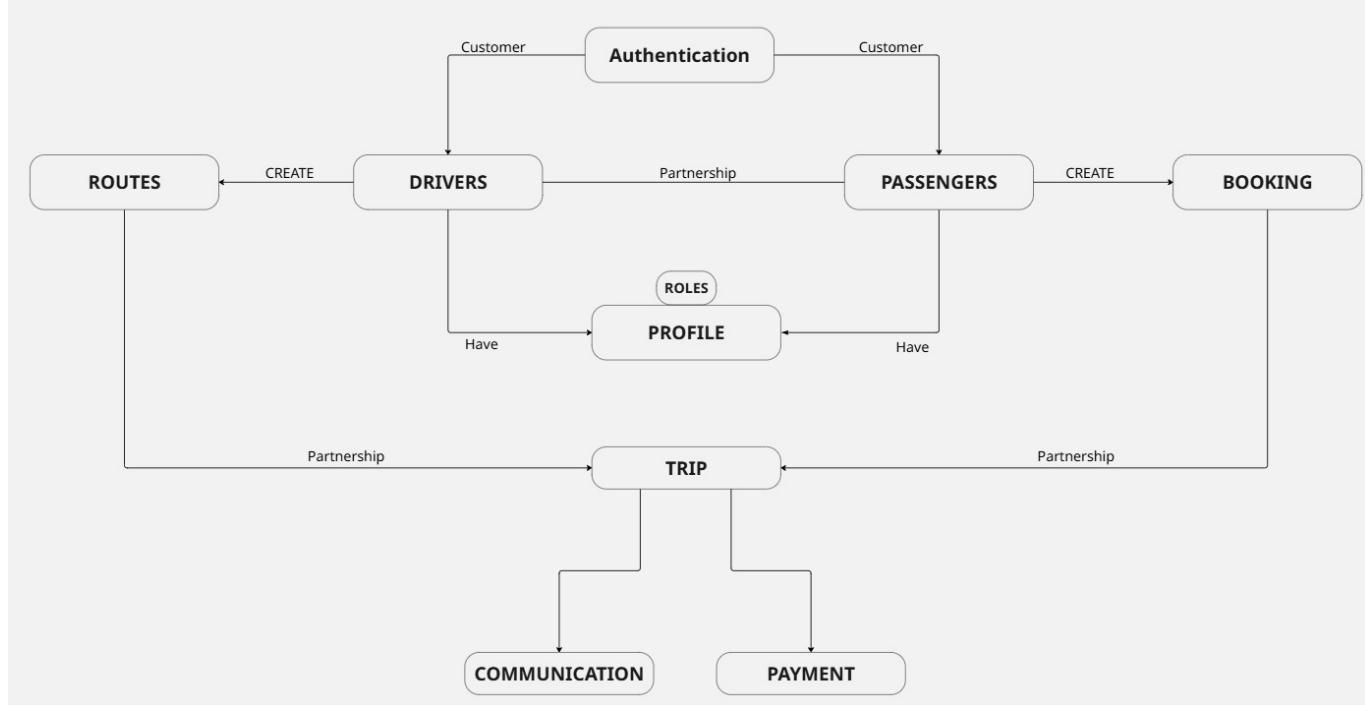
|  |   |
|--|---|
| <p><b>Context Overview:</b><br/>Se encargará del chat temporal entre el conductor y pasajero(s) mientras el viaje este activo</p>                          | <p><b>BUSINESS ROLES:</b></p> <div style="display: flex; justify-content: space-around;"> <div style="text-align: center;"> <p>Debe de haber al menos un pasajero y chofer en el viaje</p> </div> <div style="text-align: center;"> <p>Ambos usuarios deben de estar verificados</p> </div> <div style="text-align: center;"> <p>Si un pasajero cancela, se libera automáticamente una plaza.</p> </div> </div> |
| <p><b>CAPABILITIES:</b></p> <ul style="list-style-type: none"> <li>- Enviar mensajes</li> <li>- Enviar notificaciones</li> <li>- Cancelar viaje</li> </ul> | <p><b>UBIQUITOUS LANGUAGE:</b></p> <ul style="list-style-type: none"> <li>- Chat</li> <li>- Mensajes</li> <li>- Notificaciones</li> <li>- Comunicación</li> </ul>   |
| <p><b>DEPENDENCIES:</b></p> <ul style="list-style-type: none"> <li>- Booking Context</li> </ul>  |   |
| <p><b>DESIGN CRITIQUE:</b></p>   | <div style="display: flex; justify-content: space-around;"> <div style="text-align: center;"> <p>Chat cifrado de extremo a extremo</p> </div> <div style="text-align: center;"> <p>Consistencia de mensajes</p> </div> <div style="text-align: center;"> <p>Persistencia y retención del chat (¿Quedarse una hora después de que el viaje culmine?)</p> </div> </div>   |

Payment & Cost-Sharing Context:

| <b>NOMBRE:</b> Payment & Cost-Sharing Context   |   |
|---|---|
| <b>Context Overview:</b><br>Encargado de los pagos al iniciar viajes, se encargará de gestionar un pago automático cuando haya una plaza disponible para una ruta   | <b>BUSINESS ROLES:</b> <div style="background-color: #ffffcc; padding: 5px; margin-bottom: 10px;">           El usuario debe de pagar antes de iniciar el viaje         </div> <div style="background-color: #ffffcc; padding: 5px;">           Si el conductor termina el viaje, se reembolsa el dinero         </div> |
| <b>CAPABILITIES:</b> <ul style="list-style-type: none"> <li>- Realizar pago</li> <li>- Reembolso</li> <li>- Registrar pago</li> <li>- Esperar pago</li> <li>- Rechazar Pago</li> </ul>  | <b>UBIQUITOUS LANGUAGE:</b> <ul style="list-style-type: none"> <li>- Pago</li> <li>- Método de Pago</li> <li>- Viaje</li> <li>- Cancelación</li> <li>- Pasajero</li> </ul>  |
| <b>DEPENDENCIES:</b> <ul style="list-style-type: none"> <li>- Trip Context</li> </ul>   |   |
| <b>DESIGN CRITIQUE:</b> <div style="display: flex; justify-content: space-around; align-items: center;"> <div style="text-align: center;"> <div style="background-color: #ffffcc; padding: 5px; width: 150px; height: 50px;"></div> <p>Integración con pasarelas de pago (Stripe, Paypal, Yape, etc)</p> </div> <div style="text-align: center;"> <div style="background-color: #ffffcc; padding: 5px; width: 150px; height: 50px;"></div> <p>Consistencia en transacciones</p> </div> <div style="text-align: center;"> <div style="background-color: #ffffcc; padding: 5px; width: 150px; height: 50px;"></div> <p>Seguridad y Cumplimiento Legal (PCI DSS para Perú)</p> </div> </div> |   |

## 2.5.2 Context Mapping

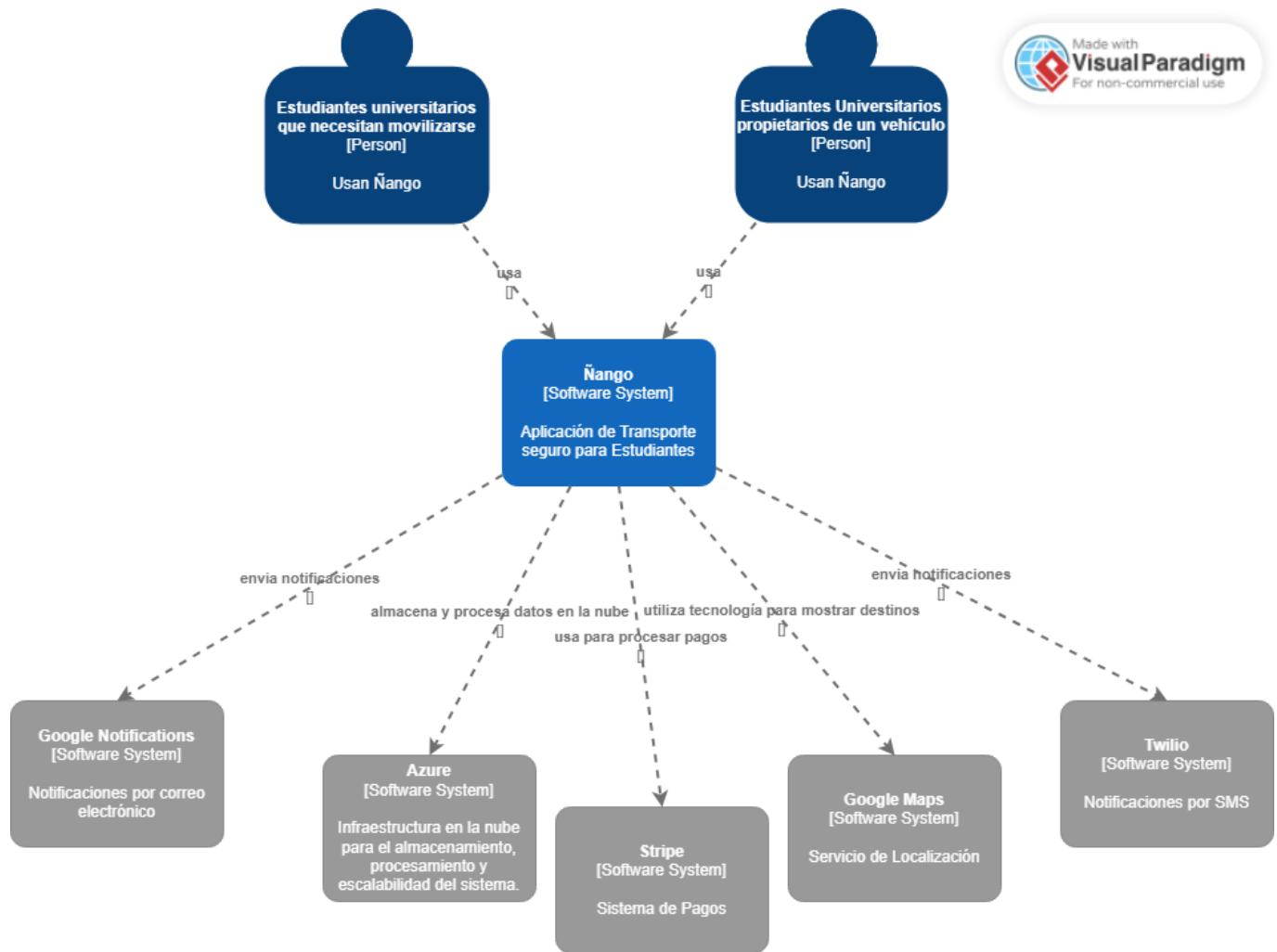
El context mapping es una técnica utilizada en el desarrollo de software, especialmente en el enfoque Domain-Driven Design (DDD), nos ayuda a visualizar y gestionar las relaciones entre los diferentes Bounded Contexts dentro de un Sistema. En este caso, se mostrará el Context Mapping desarrollado para nuestro sistema:



## 2.5.3 Software Architecture

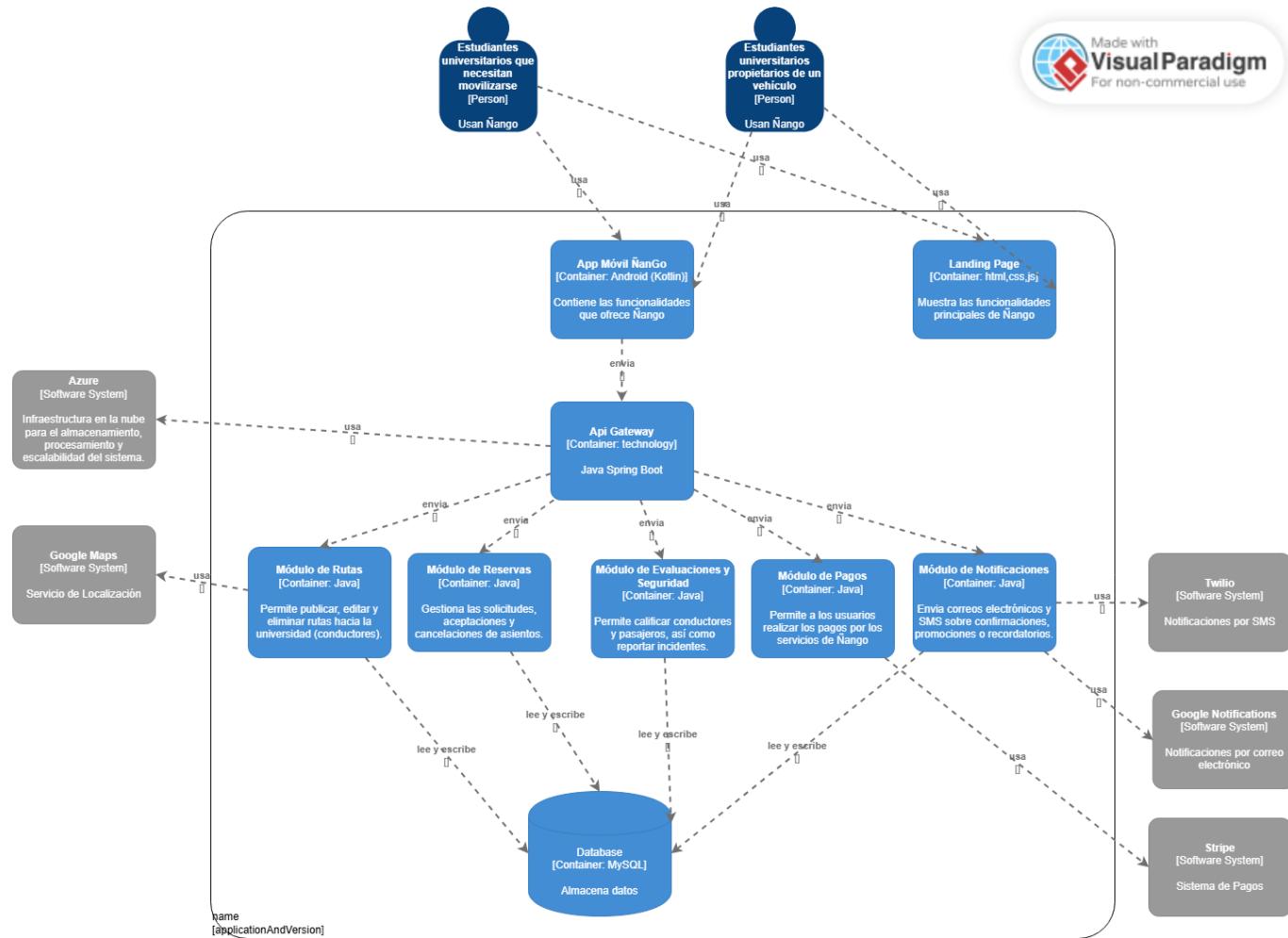
### 2.5.3.1 Software Architecture Context Level Diagrams

El Context Level Diagram muestra la visión general de la aplicación ÑanGo, identificando a los actores principales (estudiantes con y sin vehículo) y la manera en que interactúan con el sistema. Este nivel permite comprender los límites de la solución, los sistemas externos involucrados y el flujo de información esencial para el funcionamiento de la plataforma.



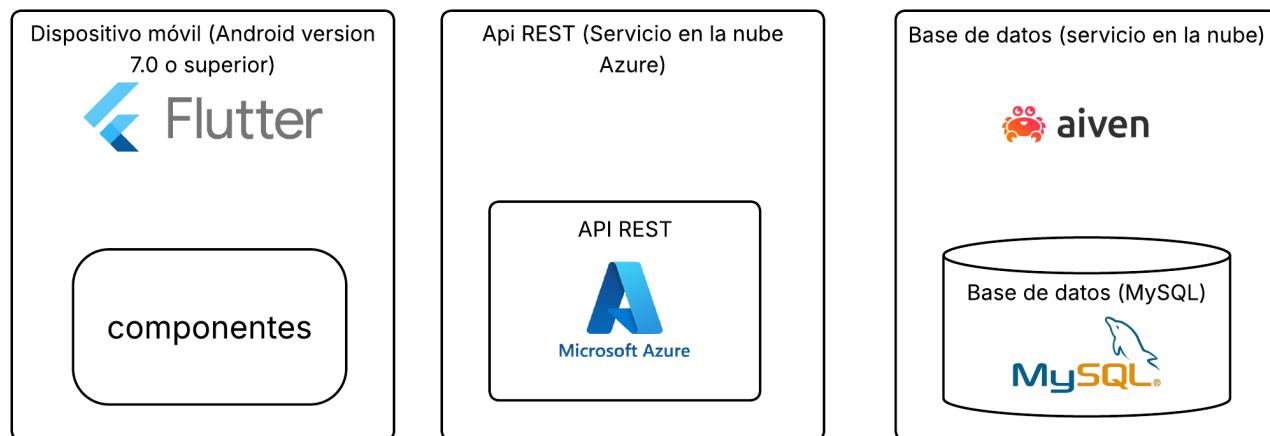
### 2.5.3.2 Software Architecture Container Level Diagrams

El Container Level Diagram detalla los principales contenedores que componen la arquitectura de ÑanGo, incluyendo la aplicación móvil, la landing page, los microservicios, la base de datos y los sistemas externos integrados. Este nivel permite visualizar cómo se distribuyen las responsabilidades, las tecnologías utilizadas y cómo se comunican entre sí los diferentes componentes.



### 2.5.3.3 Software Architecture Deployment Diagrams

El Deployment Diagram representa la disposición física de los componentes de ÑanGo en el entorno de infraestructura, mostrando los servidores, contenedores y servicios en la nube necesarios para el despliegue. Este nivel describe cómo se implementa la solución en términos de hardware, software y servicios externos, asegurando escalabilidad, seguridad y disponibilidad para los usuarios.



### 2.6.1. Bounded Context: Routes

#### 2.6.1.1. Domain Layer

**Propósito:** publicar/editar/eliminar rutas hacia la universidad.

## Entities / Aggregate

- **Route (Aggregate Root):** `routeId, driverId, origin: GeoPoint, destination: GeoPoint, departureAt, availableSeats, pricePerSeat: Money, status: RouteStatus, stops: List<Stop>, distanceKm, durationMinutes.`
- **Stop:** `stopId, order, location: GeoPoint, description?.`

## Value Objects

- `GeoPoint { lat, lon, address? }`
- `Money { amount, currency }`
- `RouteId, UserId`
- `RouteStatus (enum: DRAFT, PUBLISHED, CANCELED)`

## Repositories (interfaces)

- `RouteRepository { save, findById, findPublished(filter), delete }`

## Domain Services

- `RoutePolicies { canPublish(route), validateCapacity(route) }`

## Domain Events

- `RoutePublished, RouteEdited, RouteCanceled`

**Regla clave:** la **distancia y duración** provienen de **Google Maps Distance Matrix/Directions**; el dominio solo guarda el *resultado* (número), no datos de mapas.

### 2.6.1.2. Interface Layer

- **RoutesController (REST):**
  - `POST /routes (create & publish draft or publish later)`
  - `PUT /routes/{id}`
  - `POST /routes/{id}/publish`
  - `DELETE /routes/{id}`
  - `GET /routes (list published, filtros: origin/destination radius, time window, driverId)`

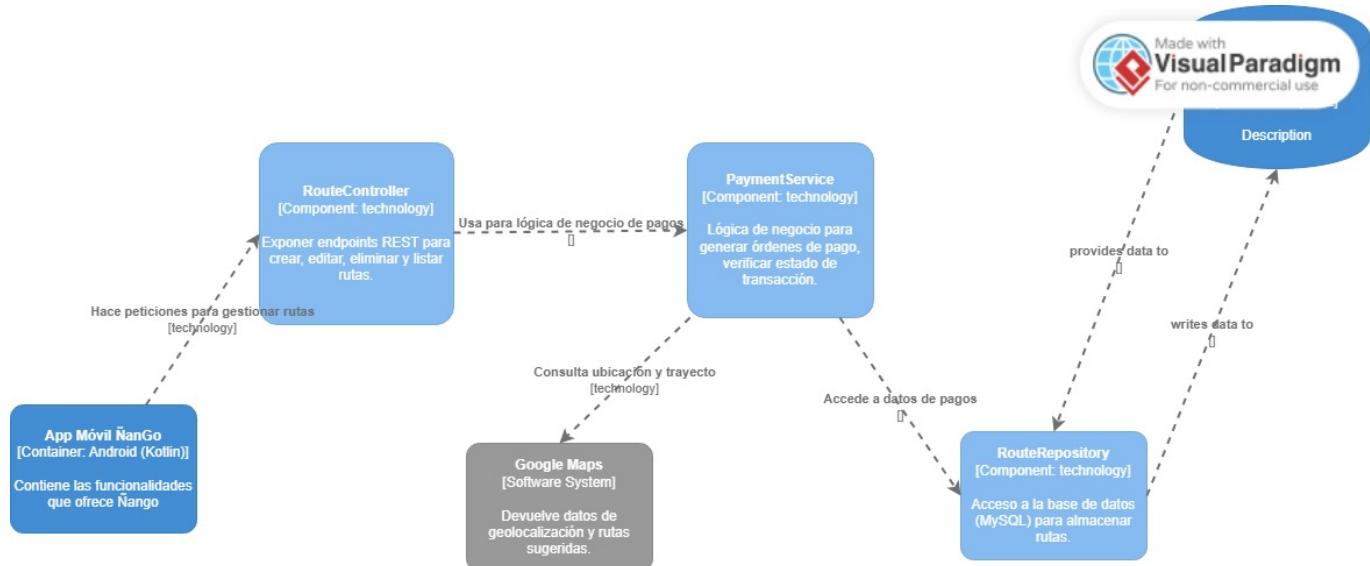
### 2.6.1.3. Application Layer

- Command Handlers:  
`CreateRouteHandler, EditRouteHandler, PublishRouteHandler, CancelRouteHandler`
- Event Handlers: publican a broker `Route*` (para que **Reservations** o **Notifications** reaccionen).
- **Integración Google Maps** (puerta de aplicación):  
`DistanceService` (puerto) ← `GoogleMapsDistanceAdapter` (adaptador de Infra).  
Se invoca al **crear/editar** para calcular `distanceKm` y `durationMinutes`.

### 2.6.1.4. Infrastructure Layer

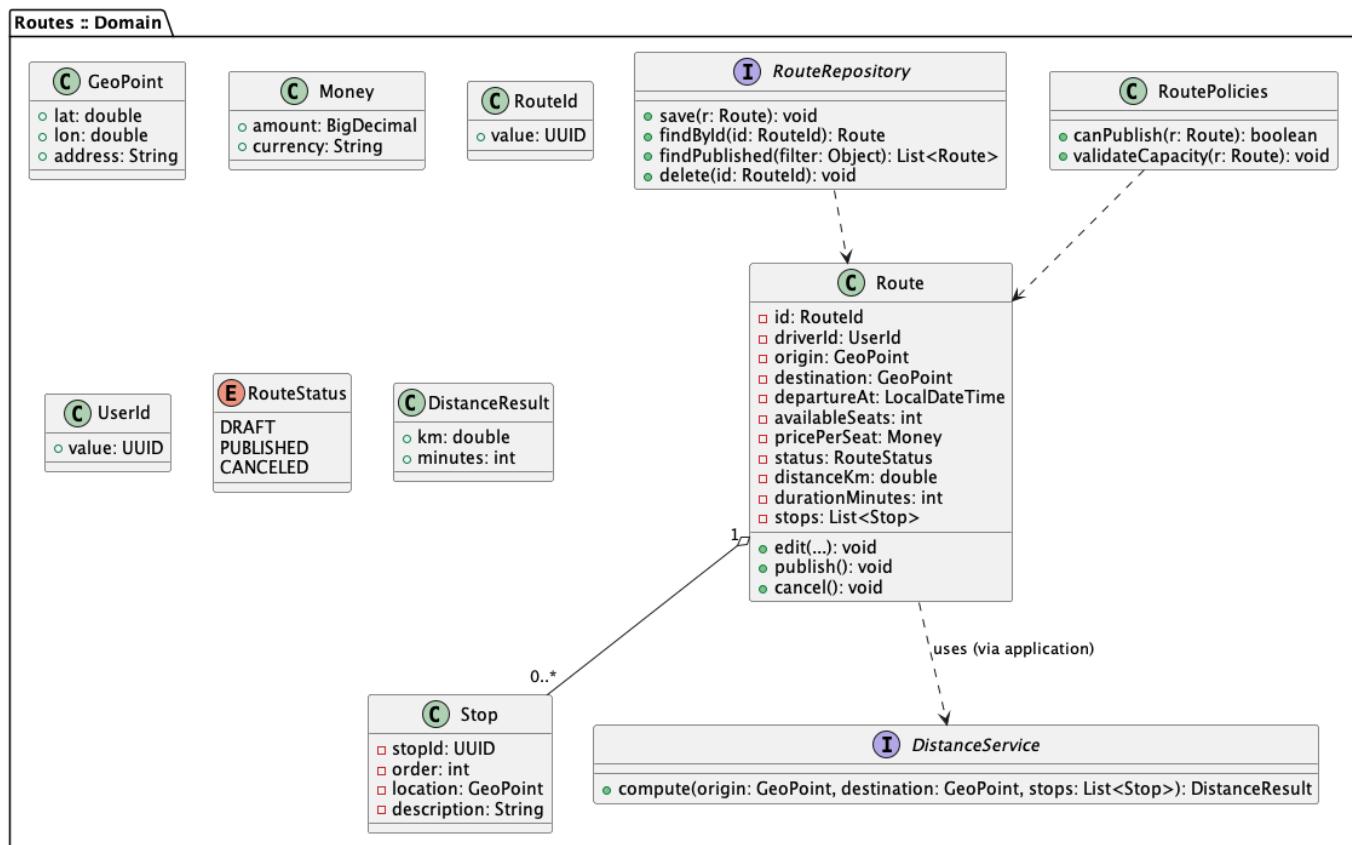
- **RouteRepositoryJpa** (PostgreSQL)
- **GoogleMapsDistanceAdapter** (HTTP → Maps Directions/Distance Matrix)
- **RoutesOutboxPublisher** (Kafka/Rabbit)

### 2.6.1.5. Software Architecture Component Level Diagram



### 2.6.1.6. Software Architecture Code Level Diagrams

#### 2.6.1.6.1. Domain Layer Class Diagrams



#### 2.6.1.6.2. Database Diagram

```

erDiagram
    ROUTES {
        uuid id PK
        uuid driver_id
        double origin_lat
        double origin_lon
        double destination_lat
        double destination_lon
        timestamp departure_at
        int available_seats
        numeric price_amount
        varchar price_currency
        varchar status
        double distance_km
        int duration_minutes
        timestamp created_at
        timestamp updated_at
    }

    ROUTE_STOPS {
        uuid id PK
        uuid route_id FK
        int stop_order
        double lat
        double lon
        varchar description
    }

    ROUTES ||--o{ ROUTE_STOPS : contains

```

## 2.6.2. Bounded Context: Payments

### 2.6.2.1. Domain Layer

**Propósito:** cobrar al pasajero por asientos reservados y liquidar al conductor.

#### Aggregate principal

- **Payment (Aggregate Root):** paymentId, reservationId, payerId, driverId, amount: Money, status: PaymentStatus, provider: PaymentProvider, providerRef (p.ej. stripePaymentIntentId), createdAt, confirmedAt?, canceledAt?.

#### Value Objects

- Money, PaymentId, UserId, ReservationId
- PaymentStatus (enum: CREATED, REQUIRES\_ACTION, CONFIRMED, CANCELED, FAILED, REFUNDED)
- PaymentProvider (enum: STRIPE)

#### Domain Services

- PaymentFactory (crea Payment desde comando de cobro)

- **SettlementPolicy** (calcula comisiones/fee)

### Repositories (interfaces)

- **PaymentRepository** { `save`, `findById`, `findByProviderRef`, `markConfirmed`, `markFailed`, `markRefunded` }

### Domain Events

- **PaymentCreated**, **PaymentConfirmed**, **PaymentFailed**, **PaymentRefunded**

## 2.6.2.2. Interface Layer

- **PaymentsController** (REST):

- `POST /payments` (crea intento de pago para una reserva)
- `POST /payments/{id}/confirm` (para flujos server-confirmation)
- `POST /payments/webhooks/stripe` (**webhook endpoint**)
- `GET /payments/{id}`

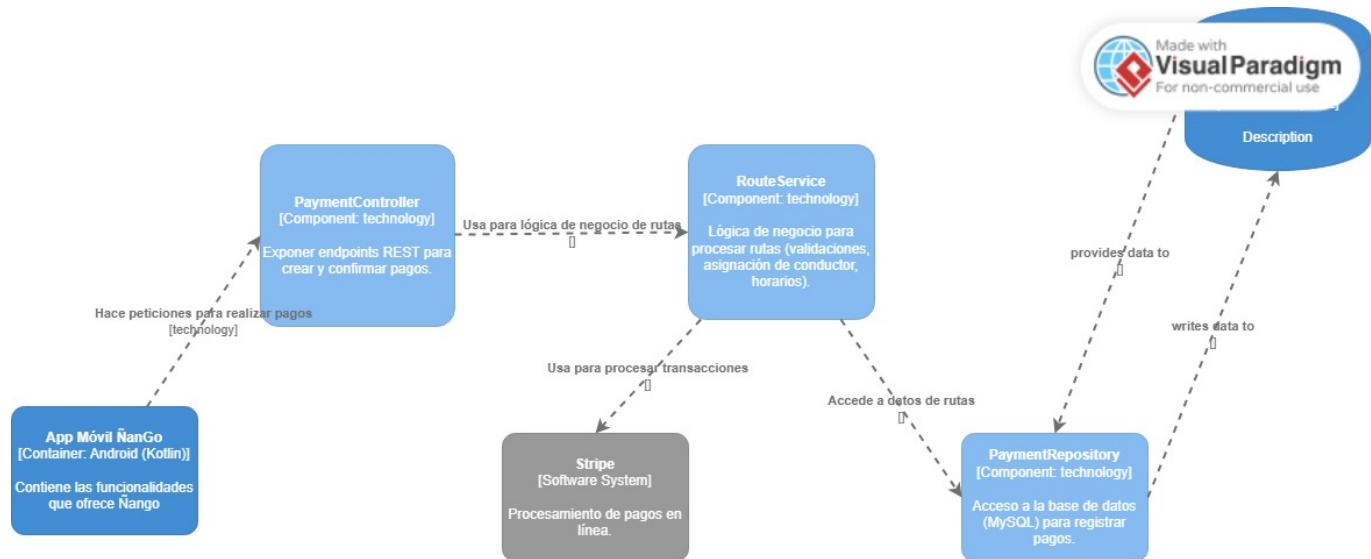
## 2.6.2.3. Application Layer

- Command Handlers: **CreatePaymentHandler**, **ConfirmPaymentHandler**, **RefundPaymentHandler**
- Event Handlers: publica **PaymentConfirmed** para que **Reservations** marque la reserva como **pagada**.
- **Integración Stripe**:
  - Puerto **PaymentGateway** con métodos: `createPaymentIntent`, `confirmPayment`, `refund`.
  - Adaptador **StripeGatewayAdapter** (SDK/HTTP).

## 2.6.2.4. Infrastructure Layer

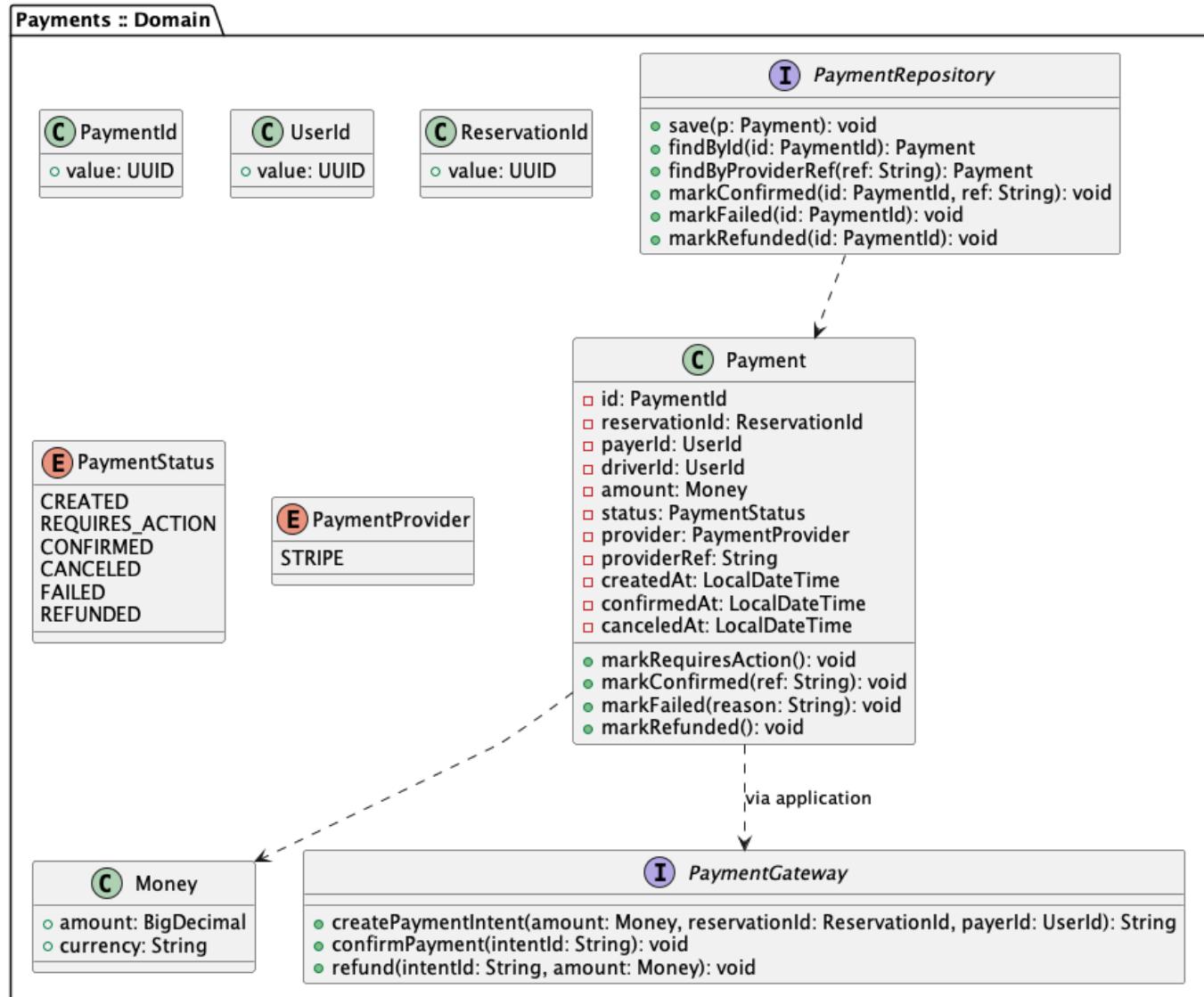
- **PaymentRepositoryJpa** (PostgreSQL)
- **StripeGatewayAdapter** (Stripe SDK/HTTP)
- **PaymentsOutboxPublisher** (Kafka/Rabbit)

## 2.6.2.5. Software Architecture Component Level Diagram



## 2.6.2.6. Software Architecture Code Level Diagrams

### 2.6.2.6.1. Domain Layer Class Diagrams



### 2.6.2.6.2. Database Diagram

```
erDiagram
    PAYMENTS {
        uuid id PK
        uuid reservation_id
        uuid payer_id
        uuid driver_id
        numeric amount
        varchar currency
        varchar status
        varchar provider           "STRIPE"
        varchar provider_ref        "stripe payment_intent id"
        timestamp created_at
        timestamp confirmed_at
        timestamp canceled_at
        timestamp updated_at
    }

    PAYMENT_LOGS {
        uuid id PK
        uuid payment_id FK
        varchar source           "api|webhook|scheduler"
        varchar event_type
        "created|requires_action|confirmed|failed|refunded"
        text message
        jsonb raw_payload
        timestamp occurred_at
    }
    PAYMENTS ||--o{ PAYMENT_LOGS : has
```

## Capítulo III: Solution UI/UX Design

---

### 3.1 Product design

#### 3.1.1. Style Guidelines

##### **Colores**

La paleta de colores se definió tomando como punto de partida el logotipo, de modo que se preserve la consistencia visual y refuerce el carácter de la marca. Se usaron gamas frías por su asociación con fiabilidad y actualidad. Esta selección permite aplicar los colores en fondos, botones y textos manteniendo suficiente contraste, cumpliendo criterios de accesibilidad y favoreciendo una experiencia clara y legible.



Fuente: Elaboración propia

## Tipografía

En ÑanGo se utilizó Inter por su alta legibilidad y tiene un estilo moderno. Armamos una jerarquía simple jugando con tamaños y pesos (de Regular a SemiBold/Bold): los títulos destacan, los subtítulos guían y el texto de cuerpo se siente cómodo. Así la lectura y la navegación son más fáciles, y el look de la app se mantiene alineado con la marca.

### Tipografía: Inter

# h1

## Header Text - Inter Extra Bold 96px

### h2

Header Text - Inter Bold 32px

### h3

Header Text - Inter Semi Bold 24px

### h4

Header Text - Inter Medium 20px

### p

Text - Inter Regular 18px

*Nota:* La jerarquía visual implementada en la aplicación móvil utiliza tamaños de fuente consistentes y proporcionales para organizar la información de manera clara y estructurada. Cabe resaltar que esta estructura está diseñada para la versión móvil de la plataforma.

Fuente: Elaboración propia

## Branding

Ñango es una plataforma orientada al carpooling universitario que prioriza una experiencia simple y ágil. La marca se presenta como soporte confiable en el día a día académico conectando a estudiantes que requieren movilidad con quienes pueden ofrecer traslado. Su propuesta promueve eficiencia operativa, seguridad y cooperación dentro del campus, posicionando a Ñango como una alternativa actual y solidaria que optimiza los desplazamientos cotidianos y fortalece el sentido de comunidad,

## Logotipo

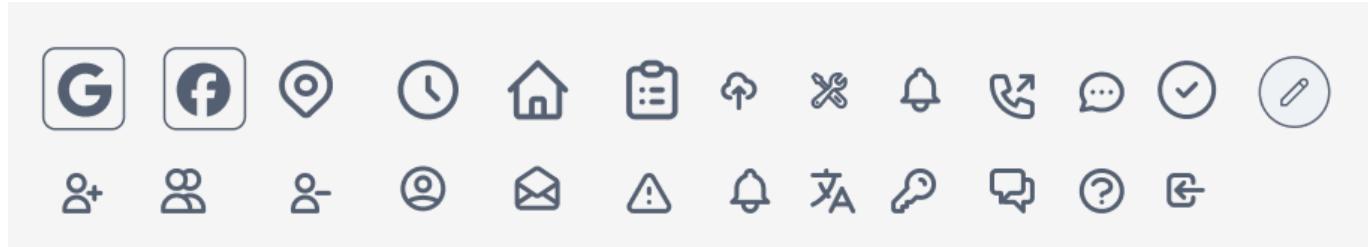
El logotipo se concibe para transmitir cercanía y confianza desde una geometría limpia y fácilmente reconocible. Las líneas suaves y contemporáneas refuerzan la idea de accesibilidad y fluidez en la experiencia. La tipografía, clara y legible, acompaña el carácter colaborativo del servicio. Inspirado en la noción de "camino compartido" y "movimiento", el símbolo representa el trayecto conjunto y el vínculo entre estudiantes que se apoyan para movilizarse, reflejando el espíritu práctico y solidario de ÑanGo.



Fuente: Elaboración propia

## Iconos

La selección de íconos prioriza la usabilidad y la orientación rápida hacia funciones críticas de la plataforma: mensajería entre pasajero y conductor, creación/gestión de reservas (incluida la reprogramación) y edición de perfil o retroalimentación del viaje. Este set contribuye a una navegación más ágil y a una interacción consistente en todas las pantallas.



*Nota:* Se emplea un logotipo alterno para mejorar la legibilidad y el contraste en fondos oscuros.

*Fuente:* Elaboración propia.

El uso sistemático de estos íconos refuerza el reconocimiento visual de las acciones, reduce la carga cognitiva y mejora la experiencia general del usuario.

### Tono de Comunicación

La voz de ÑanGo es cercana, clara y profesional. Se privilegia un estilo directo y sencillo para que pasajeros y conductores comprendan los mensajes sin ambigüedades. Este enfoque ayuda a transmitir confianza y seguridad, y humaniza cada interacción a lo largo del viaje.

### Language

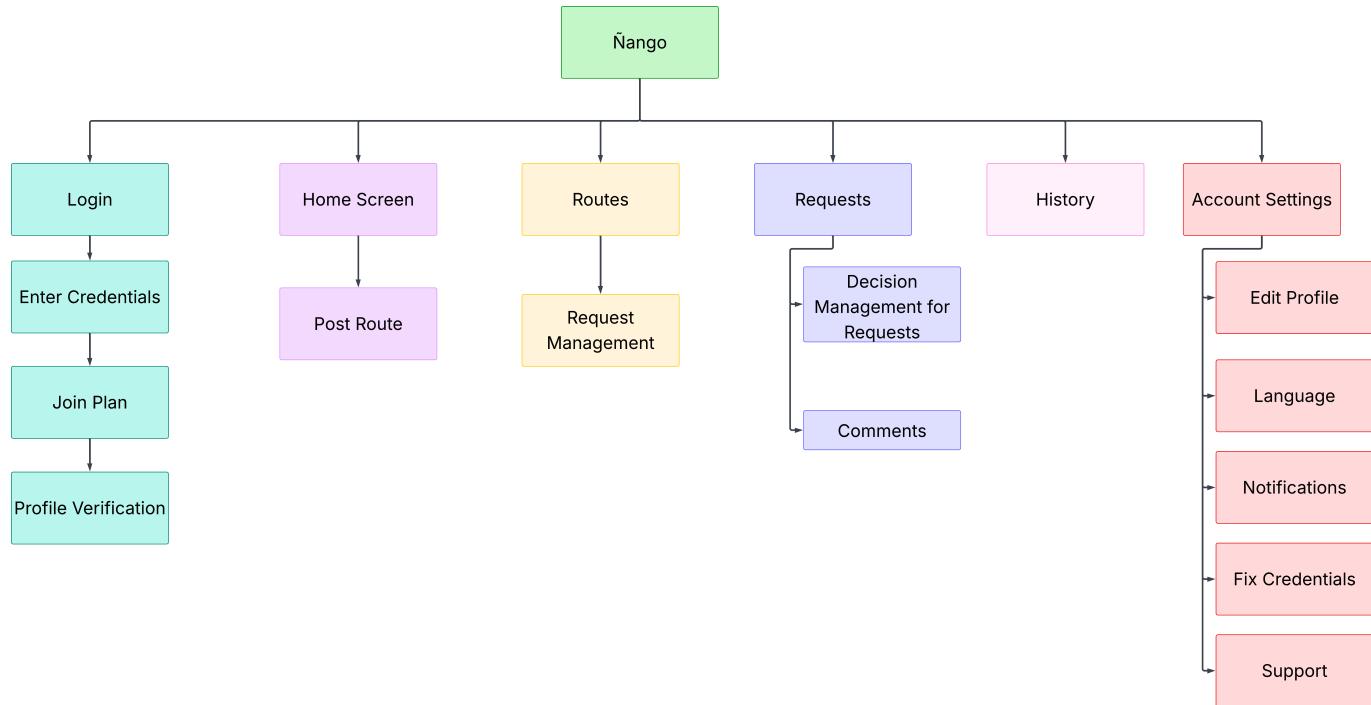
La comunicación se plantea con un lenguaje inclusivo, accesible y de fácil lectura. Se evitan tecnicismos y términos innecesariamente complejos para que cualquier persona, independientemente de su familiaridad con la tecnología, pueda interactuar con la plataforma sin barreras.

#### 3.1.2. Information Architecture

##### 3.1.2.1. Organization Systems

Se ha realizado dos vistas para la aplicación de Ñango, se basa en dos esquemas, primero es el pasajero y el segundo conductor. A continuación se mostrarán las dos estructuras para el sistemas de organización de la aplicación web.

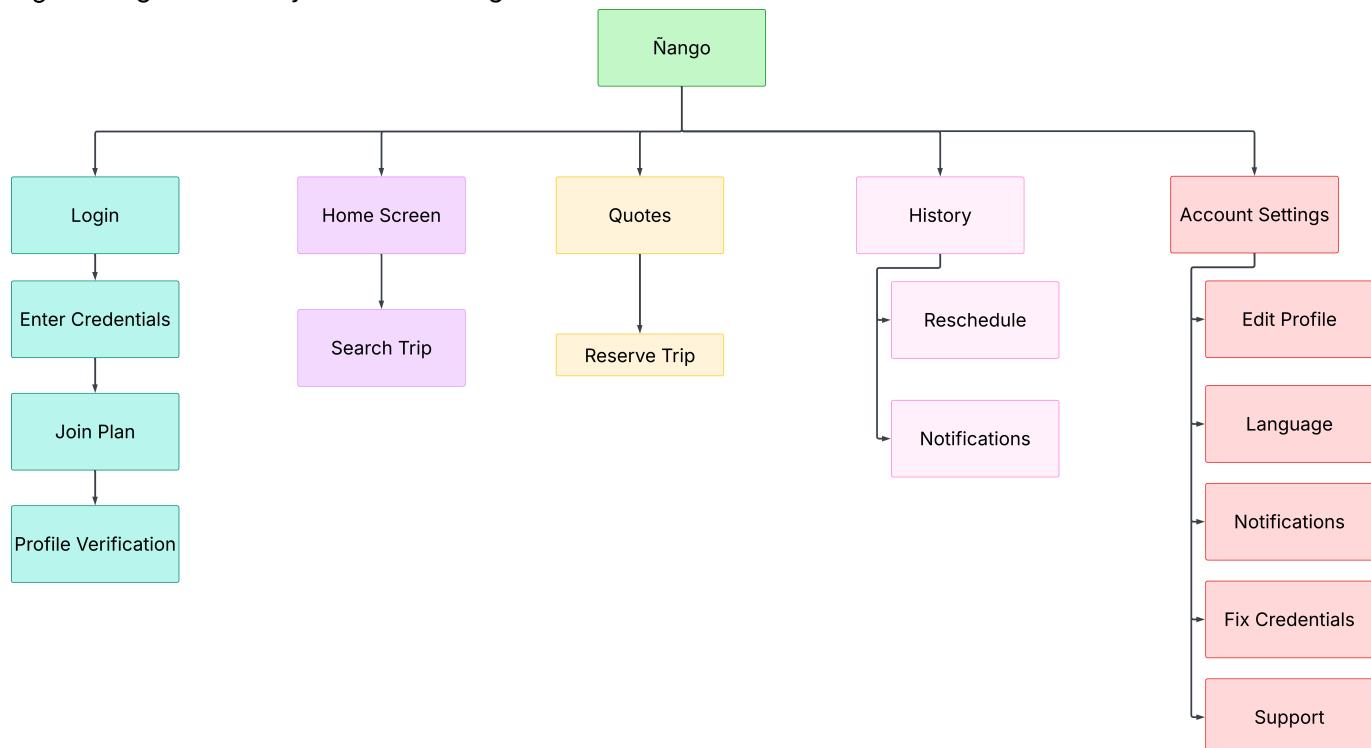
Figura: Organization System – Driver.



Fuente: Elaboración propia.

**Conductor** La figura presenta la estructura de navegación para el perfil de conductor. El acceso inicia en "Login", donde el usuario "Enter Credentials", selecciona "Join Plan" y completa la "Profile Verification". Desde "Home Screen", el conductor puede "Post Route" para publicar un viaje. En el módulo "Routes" se gestiona el estado de las reservas mediante "Request Management". A su vez, "Requests" permite revisar solicitudes recibidas y tomar acciones en "Decision Management for Requests"; en este flujo también se consultan "Comments" registrados por los pasajeros. De forma transversal, el conductor puede revisar su "History" y administrar su cuenta en "Account Settings", donde se incluyen "Edit Profile", "Language", "Notifications", "Fix Credentials" y "Support" para soporte y ajustes de perfil.

Figura: Organization System – Passenger.



*Fuente:* Elaboración propia.

**Pasajero** La figura muestra la estructura de navegación para el perfil de pasajero. El flujo comienza en "Login", continuando con "Enter Credentials", la elección del "Join Plan" y la "Profile Verification". Desde "Home Screen", el usuario accede a "Search Trip" para buscar rutas disponibles. En "Quotes", visualiza opciones y procede con "Reserve Trip" para confirmar su reserva. El historial de viajes se consulta en "History", que además habilita "Reschedule" (reprogramar con el mismo conductor cuando aplique) y "Notifications". Finalmente, en "Account Settings" se centraliza la administración de la cuenta: "Edit Profile", "Language", "Notifications", "Fix Credentials" y "Support" para asistencia.

### 3.1.2.2. Labelling Systems

El conjunto de información Ñango será representado por el sistema mediante las siguientes etiquetas.

| Labeling System | Descripción  |
|-----------------|--|
| Home            | Pantalla de bienvenida de la web/app con un mensaje destacado y un botón (CTA) que dirige al módulo "Routes".  |
| Routes          | Área donde el conductor define y gestiona los detalles de la ruta que ofrecerá a los pasajeros.  |
| Request         | Sección para revisar las solicitudes de los usuarios (pasajeros), mostrando datos como fecha de envío, nombre de la ruta, estado de la solicitud, entre otros. |
| History         | Registro de los viajes del conductor, incluyendo los estados pendiente, confirmado o realizado.  |

*Nota:* Correspondiente a la vista de Conductor

*Fuente:* Elaboración propia

| Labeling System | Descripción   |
|-----------------|---|
| Home            | Pantalla inicial que recibe al usuario con un mensaje llamativo y un botón (CTA) que lo lleva a "Quotes".     |
| Quotes          | Módulo que muestra las opciones/cotizaciones de viaje disponibles según la búsqueda realizada por el usuario. |
| History         | Historial de viajes del pasajero con sus detalles, incluido el estado actual de cada viaje.                   |

*Nota:* Correspondiente a la vista de Pasajero\*

*Fuente:* Elaboración propia

### 3.1.2.3. SEO Tags and Meta Tags

**SEO (Landing Page)** Title: "Ñango – Share the ride, save the day"

Meta Description: "Connect with classmates and share rides to campus in a safe, affordable, and eco-friendly way. Ñango offers a tech-based carpooling solution designed for students in Peru."

Keywords: "Ñango, student carpooling, shared transportation, campus rides, university mobility, safe ride app, ride sharing Peru, student transport solution"

Author: TinkuyTech Team

### **ASO (App Store Optimization)**

App Title: Ñango — Share the ride, save the day

App Keywords: carpool universitario, transporte compartido, campus, movilidad, ride sharing, estudiantes, Perú

App Subtitle: Safer campus rides with classmates

Description: "Connect with classmates and share rides to campus in a safe, affordable, and eco-friendly way. Ñango offers a tech-based carpooling solution designed for students in Peru."

#### **3.1.2.4. Searching Systems**

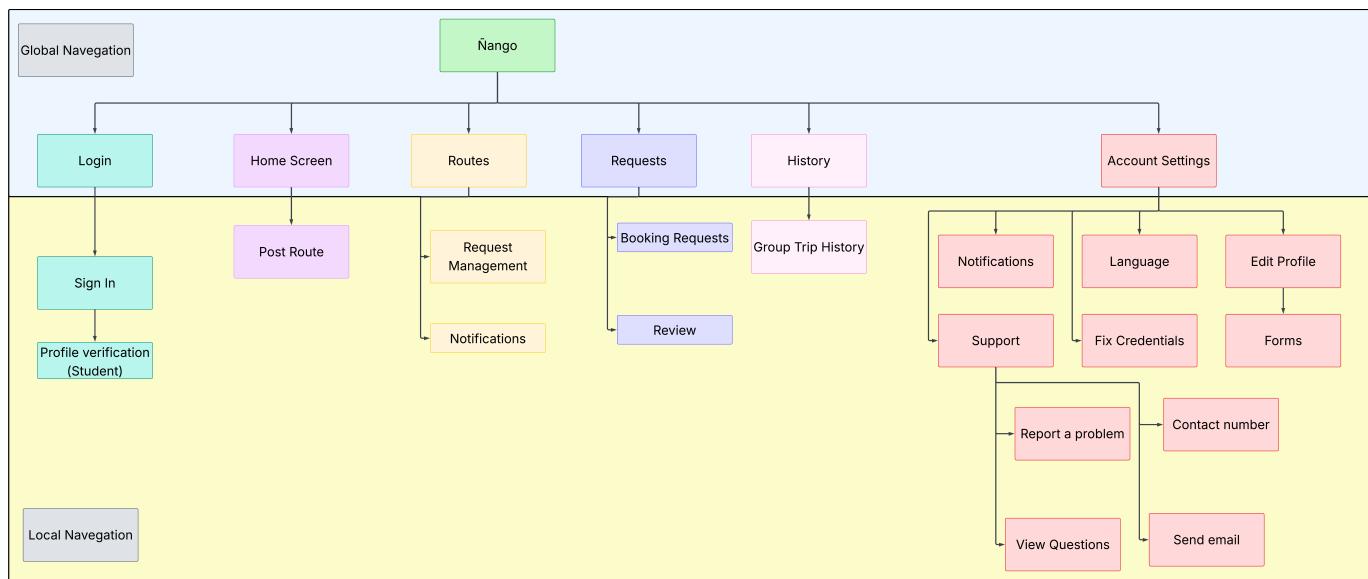
Los sistemas de búsqueda ayudan al usuario a ubicar rápidamente contenidos puntuales, ya sea mediante un campo de búsqueda o navegando por categorías. Para ello, se definió una estructura centrada en describir las funciones que permiten buscar, filtrar y encontrar. Con esto se mantiene un esquema ordenado que mejora la precisión y la velocidad durante el proceso de búsqueda.

| <b>Searching systems</b> | <b>Descripción</b>  |
|--------------------------|---|
| Search for trips         | Redirige a la sección de "Quotes", en la que este permite al usuario (pasajero) buscar viajes disponibles según "starting point", "destination", "departure date" y "departure time"  |
| Searching Join Trip      | En la sección de "Quotes", el usuario (pasajero) puede unirse a uno de los viajes disponibles, para ello presiona el botón "Join Trip".   |
| Searching Comments       | El usuario (pasajero o conductor) puede ver los comentarios hacia el conductor. El pasajero los usa para decidir unirse a un viaje; el conductor para autoevaluarse y mejorar. Ambos pueden filtrar por calificación "Rating".  |
| Searching History        | En la sección "History" el usuario (pasajero) filtra "Month", "Type of trip" and "Travel expenses" para tener un mejor control de sus reservas y del dinero gastado por viaje. Mientras que para la vista del usuario "conductor" solo se filtra "Month" y "Trip cost" pues este revisa el historial por viaje con sus pasajeros y el costo individual. |
| Searching Requests       | El usuario (conductor) busca las reservas para poder ver y aceptar a los pasajeros que deseen que se una al viaje en la sección "Request Management", a través de los filtros "type", "status" y "Date Range". De esta manera puede crear un chat personalizado solo para el grupo de viaje   |

| Searching systems | Descripción  |
|-------------------|--|
| Chat by Trip      | Filtre conversaciones grupales según el estado del viaje (Confirmado, Completado, etc.) para ambas vistas del usuario (pasajero y conductor) |

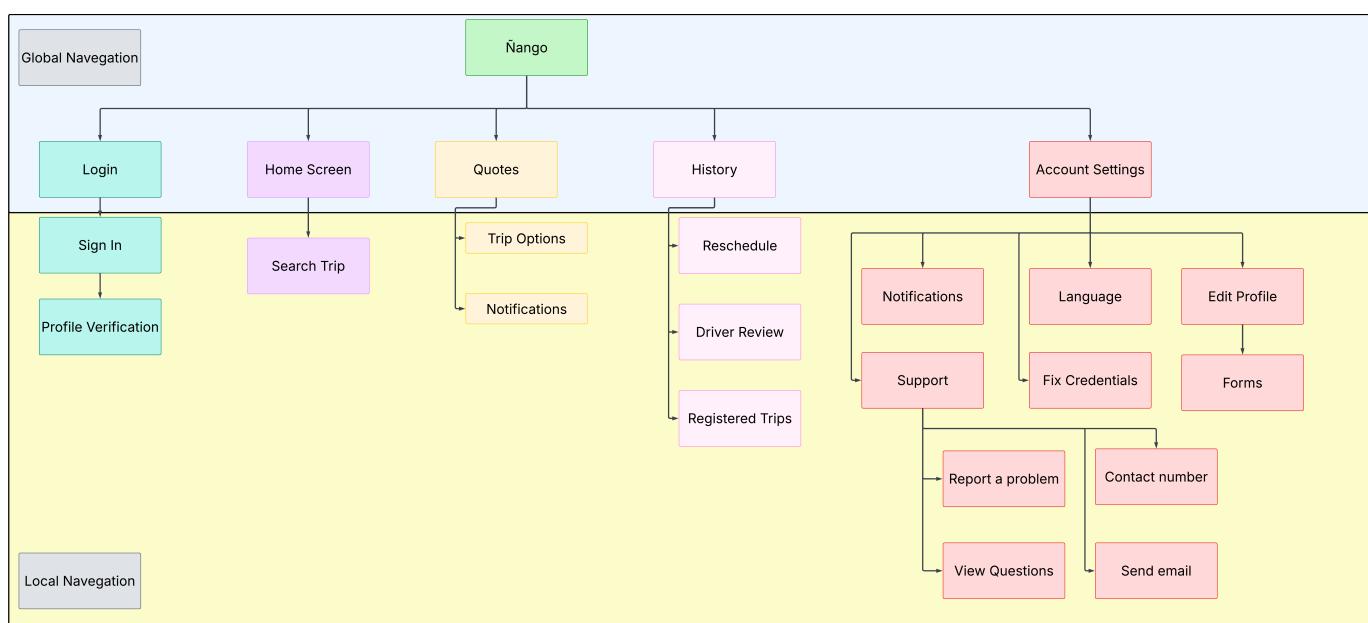
### 3.1.2.5. Navigation Systems

Para ofrecer una experiencia fluida, la aplicación organiza su navegación en dos capas: Navegación Global y Navegación Local. La global concentra los accesos a las secciones principales de la plataforma y se presenta de forma persistente en la parte inferior de la interfaz móvil. La local, en cambio, opera dentro de cada módulo principal y expone sus subsecciones (p. ej., pestañas, menús contextuales o breadcrumbs), permitiendo recorrer funciones específicas de manera ordenada y eficiente.



Nota: Estructura de navegación correspondiente al rol de conductor.

Fuente: Elaboración propia.



Nota: Estructura de navegación correspondiente al rol de pasajero.

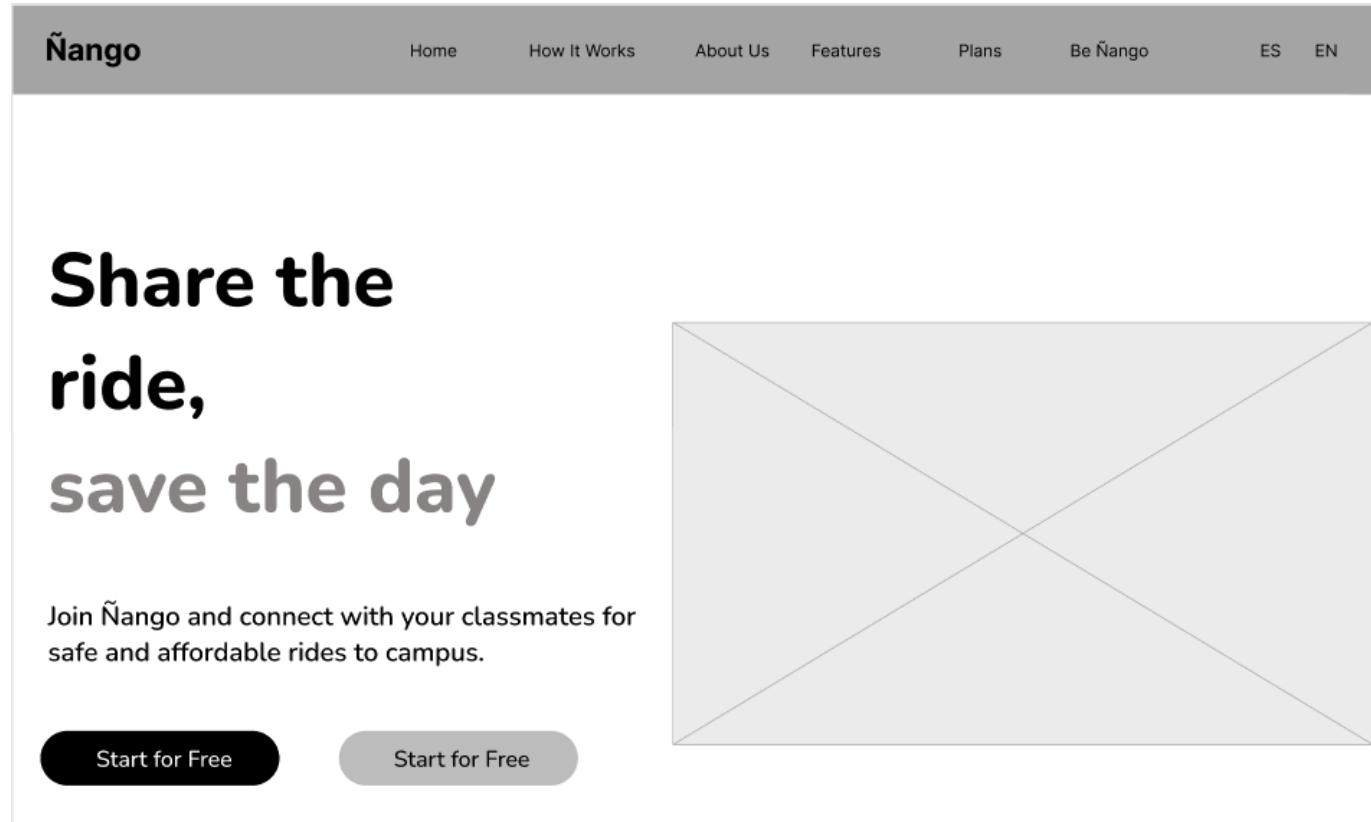
Fuente: Elaboración propia.

En conjunto, este esquema jerárquico (global + local) estructura la interfaz con claridad, mejora la descubribilidad de funciones y reduce la curva de aprendizaje. Al adaptar los recorridos según el perfil (pasajero o conductor), se ofrece una experiencia coherente y personalizada, alineada con los objetivos de accesibilidad, eficiencia y simplicidad de la plataforma.

### 3.1.3. Landing Page UI Design

#### 3.1.3.1. Landing Page Wireframe

##### Home de la landing page



Nota: El wireframe muestra la sección Home de la landing page, con un mensaje principal, botones de registro (Start for Free) y una barra de navegación con accesos a las secciones clave del sitio.

Fuente: Elaboración Propia

## Sección "How it Works"

# How It Works



1

### Find or Offer a Ride

Quickly find drivers or post your route with just one click.

2

### Confirm and Split Costs

Reserve your seat and automatically split the fuel cost.

### Ride and Rate

Enjoy a safe trip and leave a review at the end.

**Nota:** El wireframe presenta la sección How It Works, donde se explica de forma simple y visual el funcionamiento de la plataforma en tres pasos: buscar u ofrecer un viaje, confirmar y dividir costos, y calificar la experiencia.

Fuente: Elaboración Propia

## Sección "About Us"

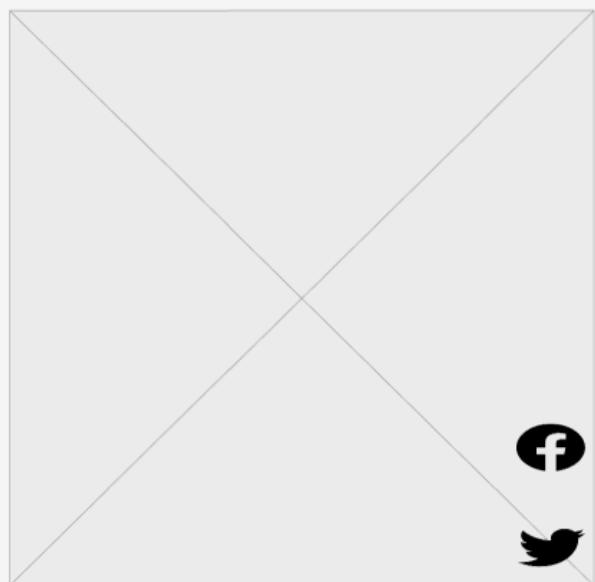
# About Us

### Our Mission

Our mission is to transform how students commute in Peru by providing a tech solution that allows easy access to a safe, affordable, and supportive shared transport system.

### Who Are We?

At TinkuTech, we work to improve student mobility in Peru. Through our ÑanGo app, we connect students who own a vehicle with those who need a ride to university. We simplify ride sharing, schedule coordination, and promote a more efficient, economical, and community-driven way of traveling.



**Nota:** Muestra la misión y visión de Ñango, junto con su enfoque en movilidad estudiantil.

Fuente: Elaboración Propia

## Sección "Features"

# Features



## Affordable Prices

By sharing rides among students, individual transportation costs are reduced, allowing users to save money on fuel or public transport fares.



## Networking

Traveling together can strengthen relationships among students from different majors, creating a network of contacts that supports academic networking and experience sharing.



## Eco-friendly

By reducing the number of individual vehicles commuting to the university, the emission of pollutants is decreased—helping the environment.

Nota: Presenta los beneficios clave de la plataforma: ahorro, networking y sostenibilidad.

Fuente: Elaboración Propia

## Sección "Plans"

# Plans

## National future

S/10 /month

- Random driver assignment
- Daily scheduling of shared rides
- Daily trip limit

Subscribe

## Pro Generation

S/50 /month

- Rescheduling with the same driver
- Frequent user discount
- Alerts for new driver routes

Subscribe

## Succes Premium

S/70 /month

- Group creation for driver priority
- Automatic scheduling based on frequent routes
- Connect with friends or meet new people during the ride

Subscribe

*Nota:* Se detallan los planes de suscripción disponibles, con sus respectivas funcionalidades.

*Fuente:* Elaboración Propia

### Sección "Be Ñango"

The image shows a registration form titled "Form". It contains three input fields: "Name", "Email", and "Password", each with a corresponding text input box. Below the input fields is a black "Login" button. The background features a large, semi-transparent "Be Ñango" logo at the top center. At the bottom of the page, there is a footer bar with links to "Política de privacidad", social media icons for Instagram, Twitter, LinkedIn, and Facebook, and "Términos y condiciones".

*Nota:* Contiene el formulario de registro para nuevos usuarios de la plataforma.

*Fuente:* Elaboración Propia

#### 3.1.3.2. Landing Page Mock-up

The image shows the homepage of the Ñango platform. At the top, there is a navigation bar with the brand name "Ñango" and links to "Inicio", "Cómo Funciona", "Nosotros", "Características", "Planes", "be Ñango", "ES", and "EN". The main visual features a large green banner with the text "Comparte el viaje, ahorra el día" and a subtext "Únete a Ñango y conecta con tus compañeros para traslados seguros y económicos al campus.". Below the banner are two buttons: "Empieza Gratis" and "Soy Conductor". To the right of the banner is an illustration of a green car with three people standing around it, one holding a smartphone displaying a map. The overall design is clean and modern, using a color palette of green, white, and grey.

**Home de la landing page**

**Nota:** El mockup muestra la sección Home de la landing page, con un mensaje principal, botones de registro (Start for Free) y una barra de navegación con accesos a las secciones clave del sitio.

Fuente: Elaboración Propia

## Sección "How it Works"

The mockup shows a screenshot of a website with a teal header. The header includes the logo 'Ñango', navigation links for 'Inicio', 'Cómo Funciona', 'Nosotros', 'Características', 'Planes', and 'be Ñango', and language options 'ES EN'. The main content area has a white background. At the top, the title 'Cómo Funciona' is displayed in a large, bold, dark green font. Below the title is a circular diagram illustrating the three-step process. On the left, there is an illustration of a woman walking towards a green car while looking at her phone. Above the car, a map shows several green icons representing users and a route line. To the right of the diagram, three numbered steps are listed: 1. 'Buscar u Ofrecer Viaje' (Search or Offer Trip), 2. 'Confirma y Comparte Gastos' (Confirm and Share Costs), and 3. 'Viaja y Opina' (Travel and Review). Each step is accompanied by a brief description in a smaller green font.

## Cómo Funciona

- Buscar u Ofrecer Viaje**  
Encuentra conductores en segundos o publica tu ruta con un clic.
- Confirma y Comparte Gastos**  
Reserva tu lugar y divide automáticamente el costo de combustible.
- Viaja y Opina**  
Disfruta tu traslado con seguridad y califica la experiencia al final.

**Nota:** El mockup presenta la sección How It Works, donde se explica de forma simple y visual el funcionamiento de la plataforma en tres pasos: buscar u ofrecer un viaje, confirmar y dividir costos, y calificar la experiencia.

Fuente: Elaboración Propia

## Sección "About Us"

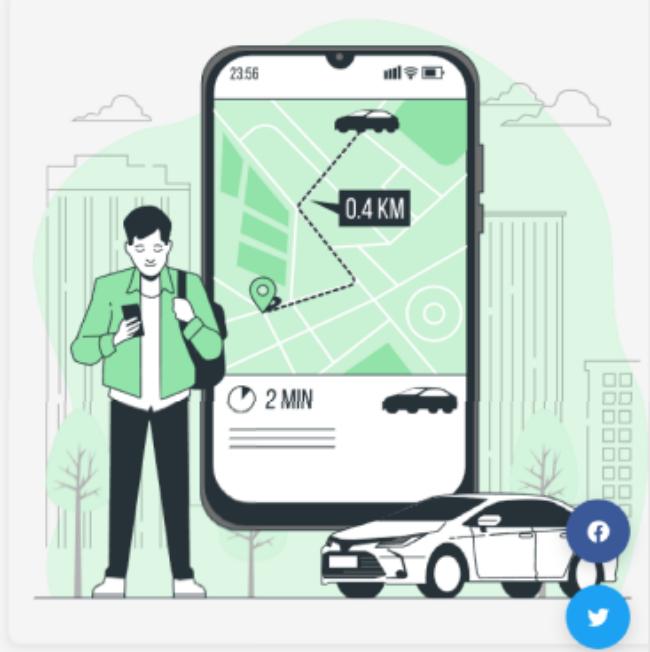
# Sobre Nosotros

## Nuestra Misión

Nuestra misión es transformar la forma en que los estudiantes se movilizan en Perú, brindándoles una solución tecnológica que les permita acceder a un sistema de transporte compartido, seguro, accesible y solidario.

## ¿Quiénes somos?

En TinkuyTech trabajamos para mejorar el transporte estudiantil en Perú. A través de nuestra app Ñango, conectamos a estudiantes con movilidad propia con quienes necesitan transporte hacia la universidad. Facilitamos rutas compartidas, coordinación de horarios y mayor seguridad, promoviendo una forma de viajar más eficiente, económica y solidaria.



*Nota:* Muestra la misión y visión de Ñango, junto con su enfoque en movilidad estudiantil.

*Fuente:* Elaboración Propia

## Sección "Features"

Nango
Inicio Cómo Funciona Nosotros Características Planos be Ñango
ES EN

## Características



### Precios cómodos

Al compartir viajes entre estudiantes, se reduce el gasto en transporte individual, permitiendo a los usuarios ahorrar dinero en combustible o tarifas de transporte público.



### Networking

Viajar juntos puede fortalecer relaciones entre estudiantes de diferentes carreras, creando una red de contactos que facilite el networking y el intercambio de experiencias académicas.



### Eco-friendly

Al reducir el número de vehículos individuales que se trasladan a la universidad, se disminuye la emisión de gases contaminantes, ayudando al medio ambiente.

[Facebook](#)
[Twitter](#)

*Nota:* Presenta los beneficios clave de la plataforma: ahorro, networking y sostenibilidad.

*Fuente:* Elaboración Propia

## Sección "Plans"

**National Future**  
s/10/mes

- Elección aleatoria de conductor
- Programación diaria de viajes compartidos
- Límite de viajes por día

**Pro generation**  
s/50/mes

- Reprogramación con un mismo conductor
- Descuento por uso frecuente
- Alertas de nuevas rutas de los conductores

**Success Premium**  
s/70/mes

- Creación de grupos para prioridad como cliente de un conductor
- Programación automática según rutas más frecuentes
- Conecta con tus amigos o los que hagas en el camino

Nota: Se detallan los planes de suscripción disponibles, con sus respectivas funcionalidades..

Fuente: Elaboración Propia

## Sección "Be Ñango"

**Form**

Name

Email

Password

Login

frame 671

Política de privacidad     Términos y condiciones

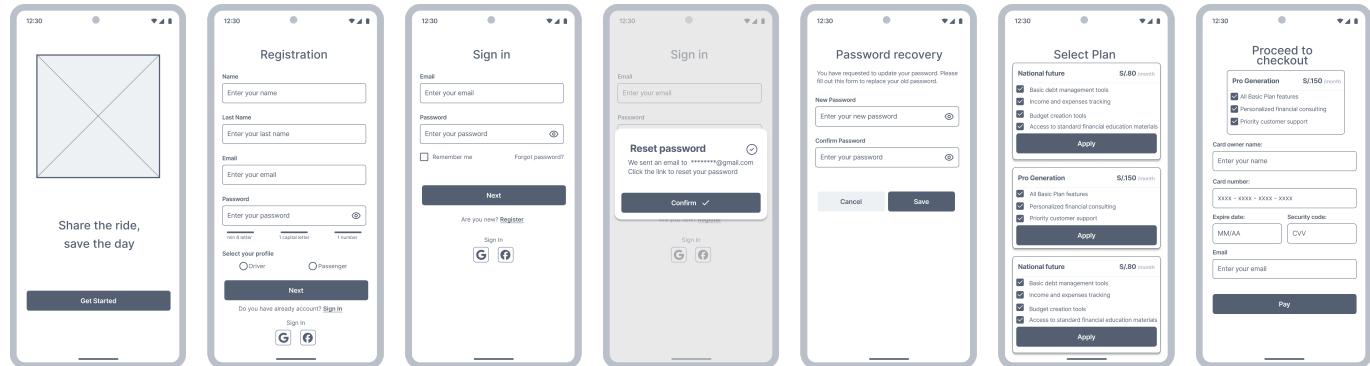
Nota: Contiene el formulario para los usuarios puedan entrar a la plataforma.

Fuente: Elaboración Propia

### 3.1.4. Mobile Applications UX/UI Design

#### 3.1.4.1. Mobile Applications Wireframes

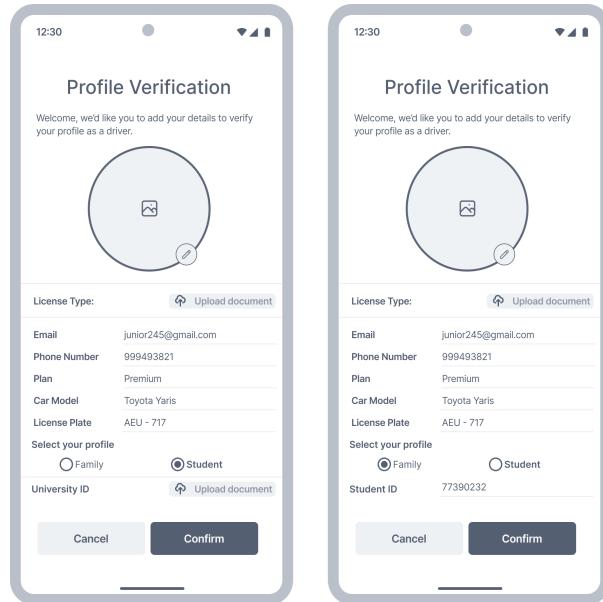
## Registro en la aplicación



Nota: Wireframes del ingreso a la aplicación, tanto el de registro e inicio de sesión, el recuperar la contraseña si ya iniciaste sesión y el seleccionar plan.

Fuente: Elaboración Propia

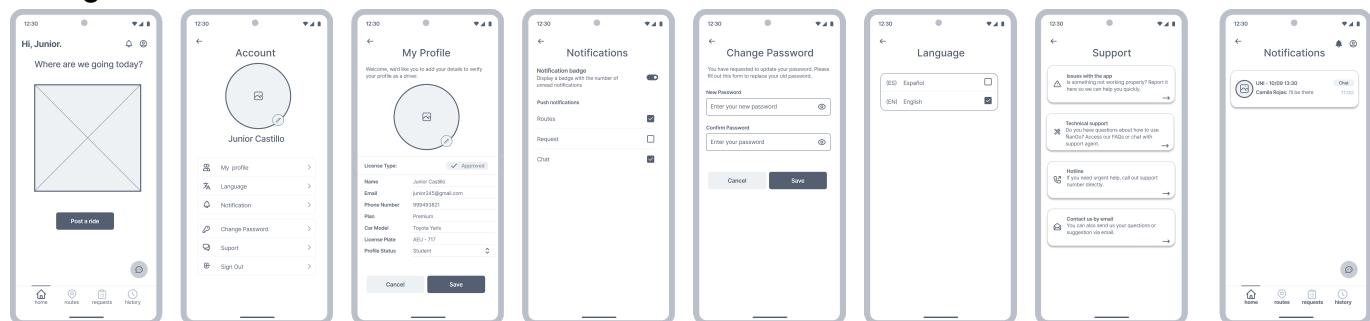
## Profile verification - Conductor



Nota: Wireframes del profile verification del los usuarios (conductores), donde se verifican sus datos personales como licencia y el ID de la universidad.

Fuente: Elaboración Propia

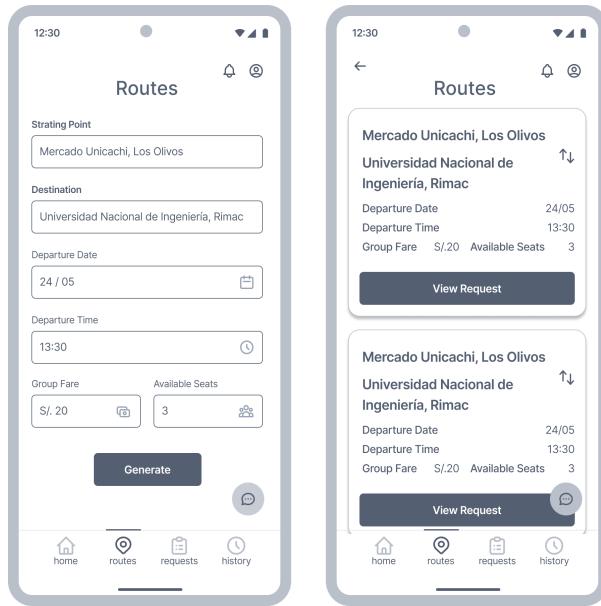
## Configuración de la cuenta - Conductor



Nota: Wireframes del inicio de la aplicación, vas a la opción de las notificaciones o a la cuenta del perfil, y te desplaza diferentes opciones como: "My Profile", "Notifications", "Change Password", "Language" y "Support".

Fuente: Elaboración Propia

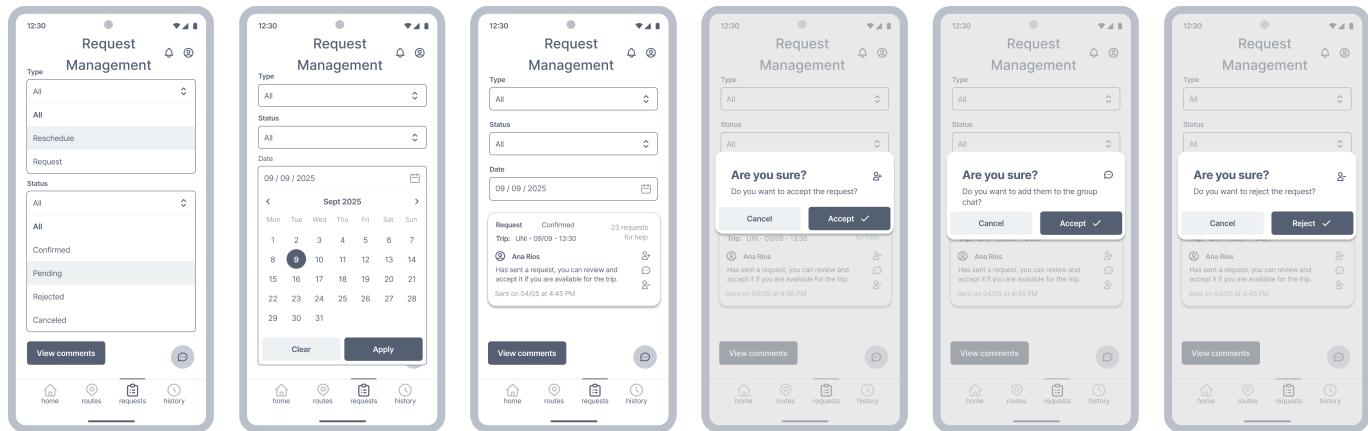
## Rutas - Conductor



Nota: Wireframes que muestra la interfaz don el conudcotr, puede generar detalles de su ruta.

Fuente: Elaboración Propria

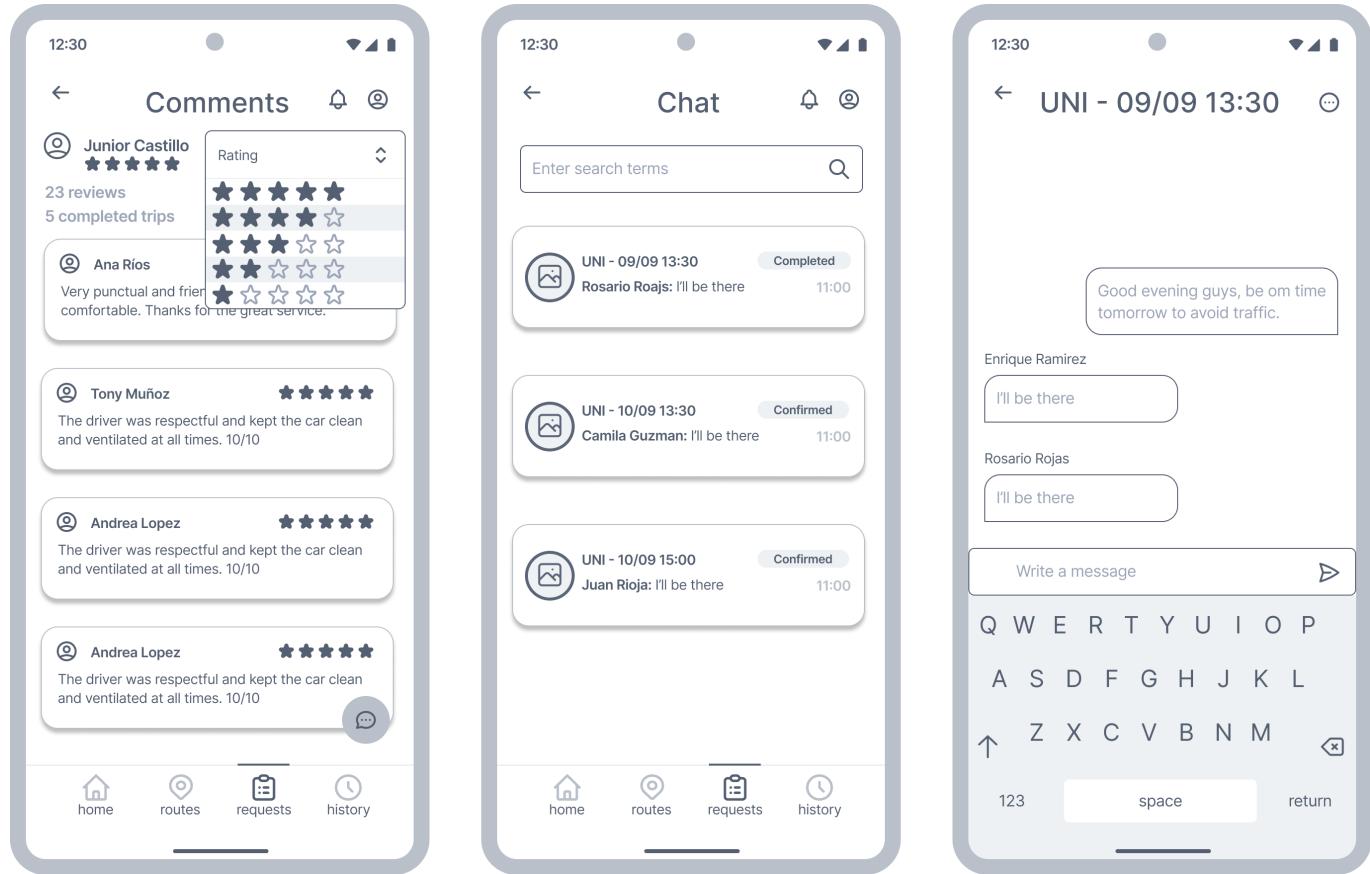
## Solicitudes - Conductor



Nota: Wireframes que muestran el proceso de las solicitudes y reprogramaciones, junto con 3 botones que permite al conductor realizar acciones como aceptar la solicitud del viaje, añadir al pasajero al grupo de chat con otros estudiantes dek viaje, y el tercero permite rechazar la solicitud del pasajero.

Fuente: Elaboración Propria

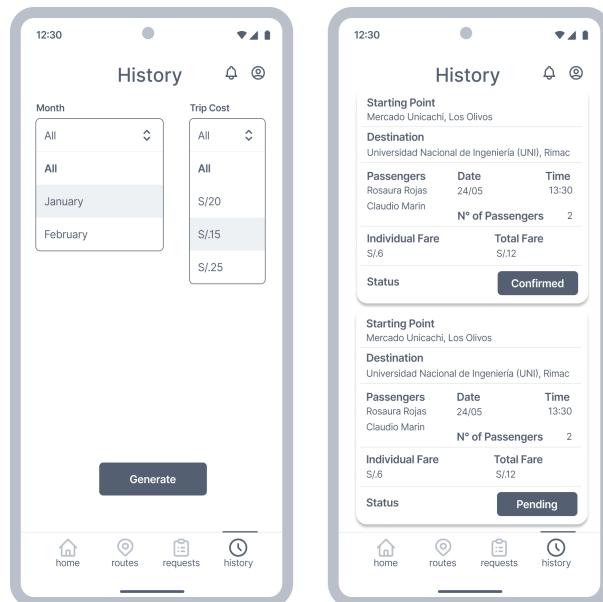
## Comentarios y chats grupales - Conductor



Nota: Wireframes donde se puede visualizar los comentarios del conductor y los wireframes de los chat grupales con los pasajeros, al costado del chat se puede observar si el viaje ya fue realizado o solo si se ha confirmado.

Fuente: Elaboración Propia

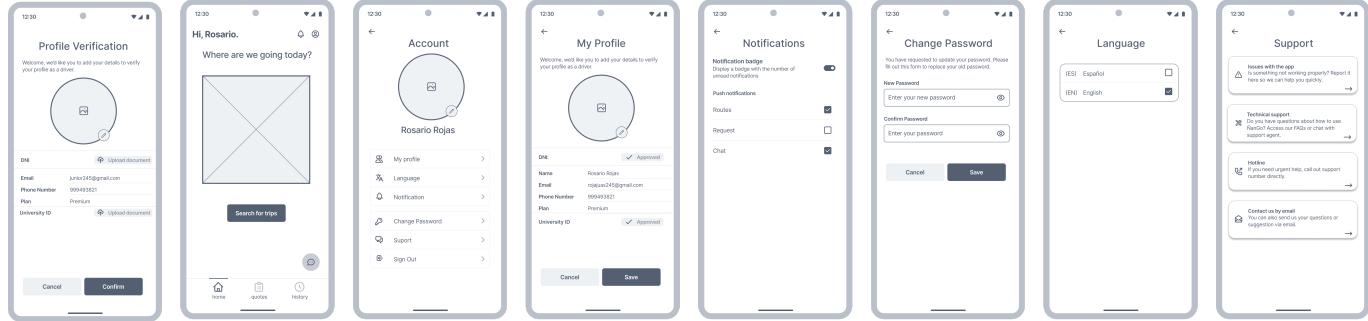
## Historial - Conductor



Nota: En estos wireframes se muestran el historial de los detalles de sus rutas como los pasajeros, en donde inicio, el destino, la fecha, hora, el numero de pasajeros, el costo total y por cada pasajero y el estado de la solicitud

Fuente: Elaboración Propia

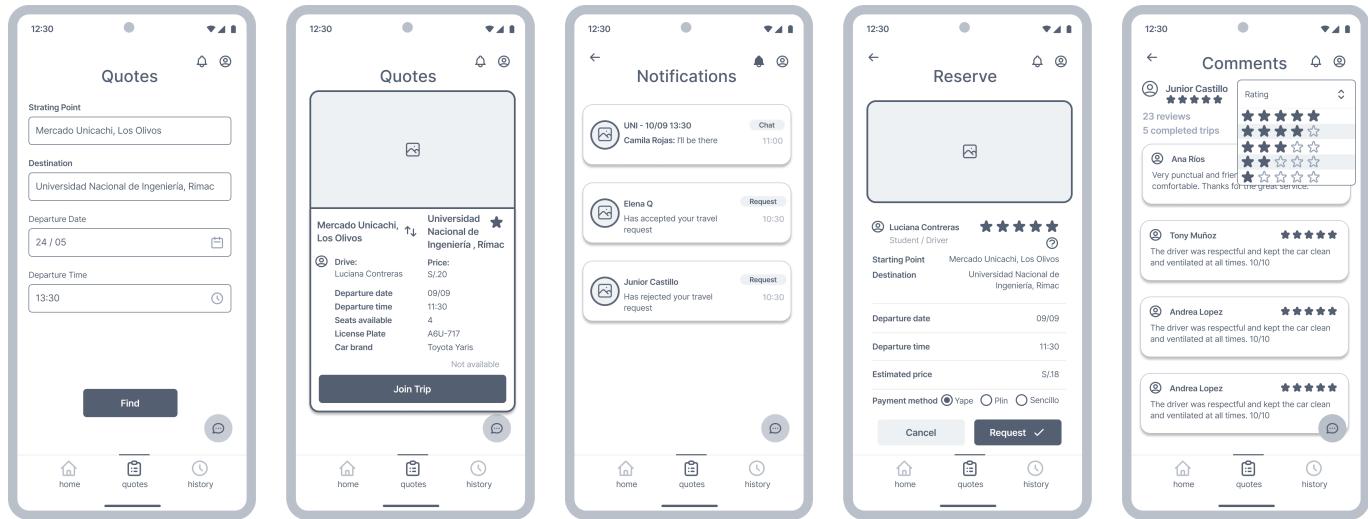
## Configuración de la cuenta - Pasajero



Nota: Wireframes del perfil del pasajero, a la ruta de configuración de la cuenta, las notificaciones, para cambiar la contraseña, cambiar de idioma y el soporte de la aplicación.

Fuente: Elaboración Propria

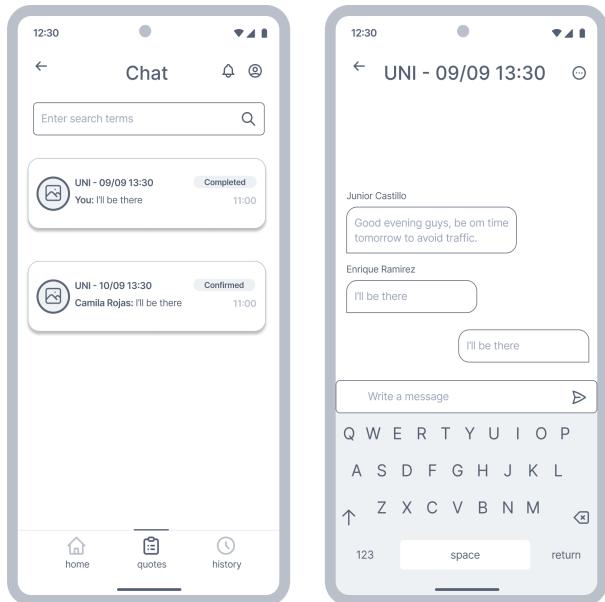
## Búsqueda de rutas - Pasajero



Nota: Wireframes sobre la búsqueda de un estudiante para las rutas disponibles, y la reserva de ella, también antes de reservar se pueden observar los wireframes de los comentarios del conductor los wireframes de notificación de cuando el conductor aceptar al viaje.

Fuente: Elaboración Propria

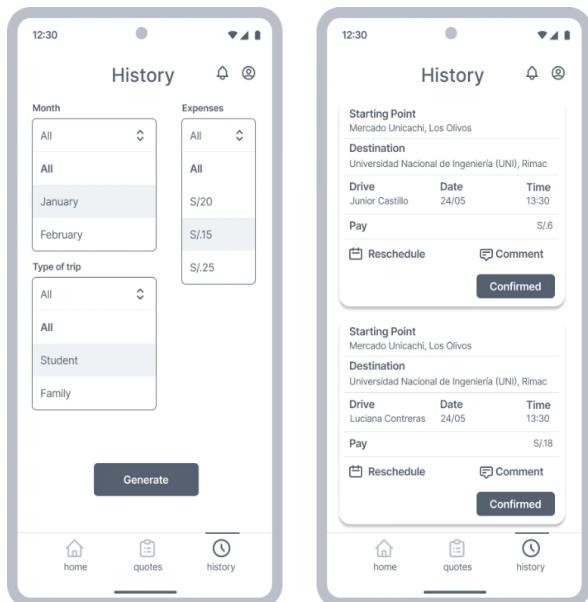
## Chat grupales - Pasajero



*Nota:* Wireframes de los chat de grupos de los pasajeros con el conductor

*Fuente:* Elaboración Propia

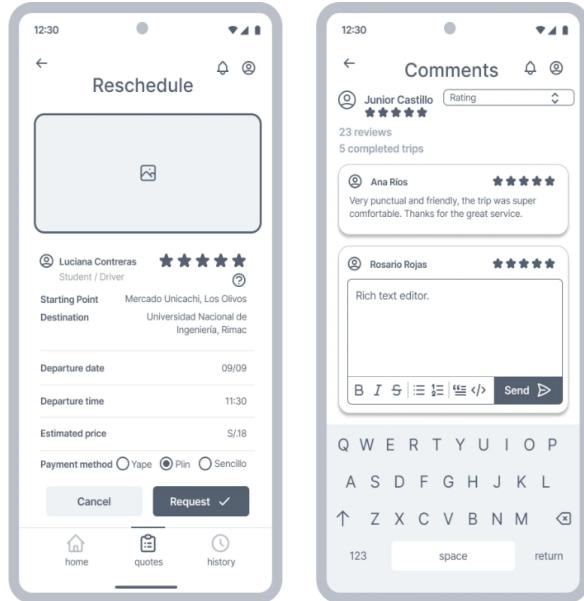
### **Historial - Pasajero**



*Nota:* Wireframes donde se muestra el historial de los viajes anteriores de los usuarios, acá se muestran las opciones que se quieren personalizar, además información detallada del viaje como en donde inicio, el destino, el día, la hora, el nombre del conductor, además de el estado de este. También se visualiza dos opciones que es sobre volver a relizar el viaje o hacer un comentario al conductor.

*Fuente:* Elaboración Propia

### **Historial 2 - Pasajero**

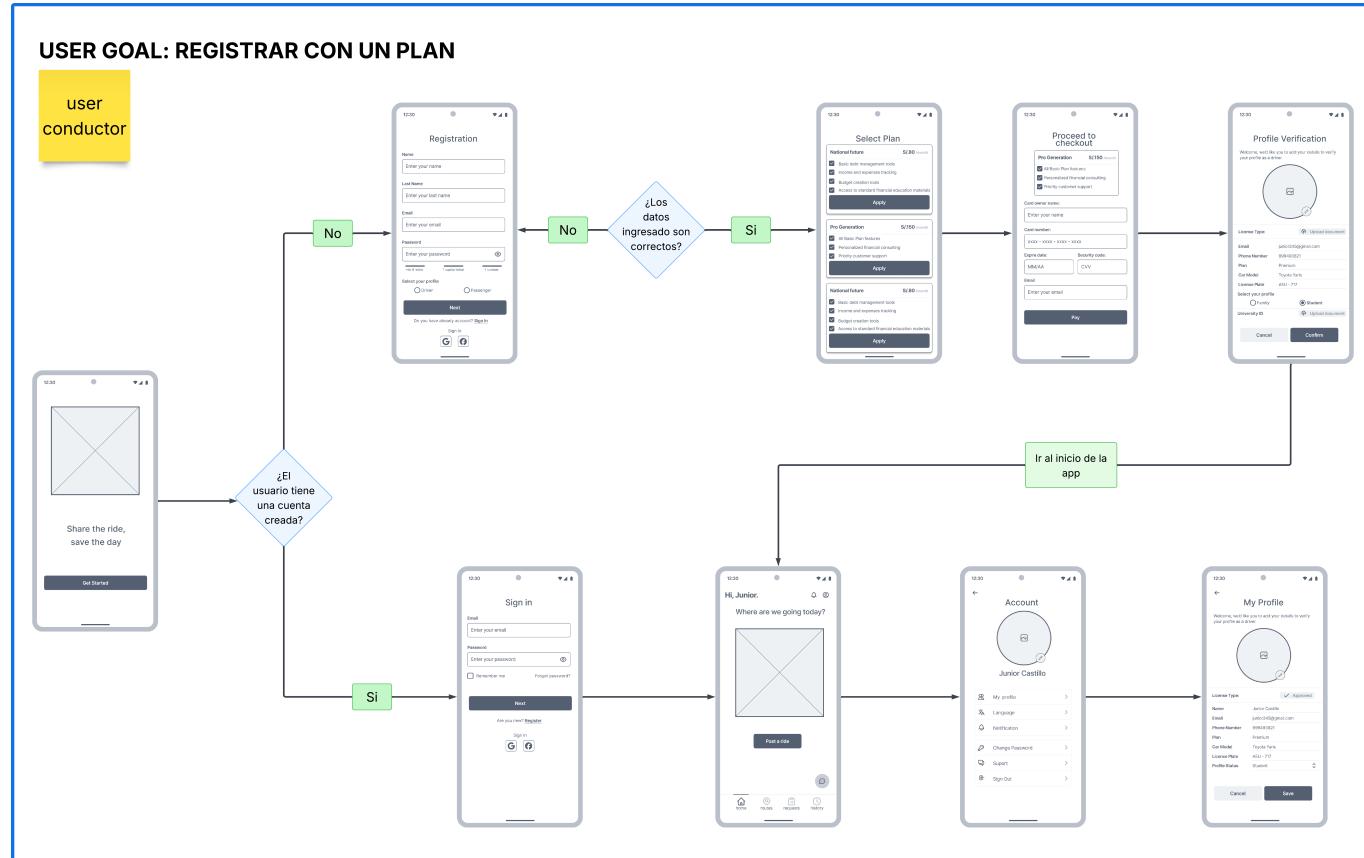


Nota: Wireframes basados en el anterior del historial, para volver realizar el viaje con la misma información y si quieres cambiar alguna opción, el siguiente es para hacer el comentario al conductor de su viaje.

Fuente: Elaboración Propia

### 3.1.4.2. Mobile Applications Wireflow Diagrams

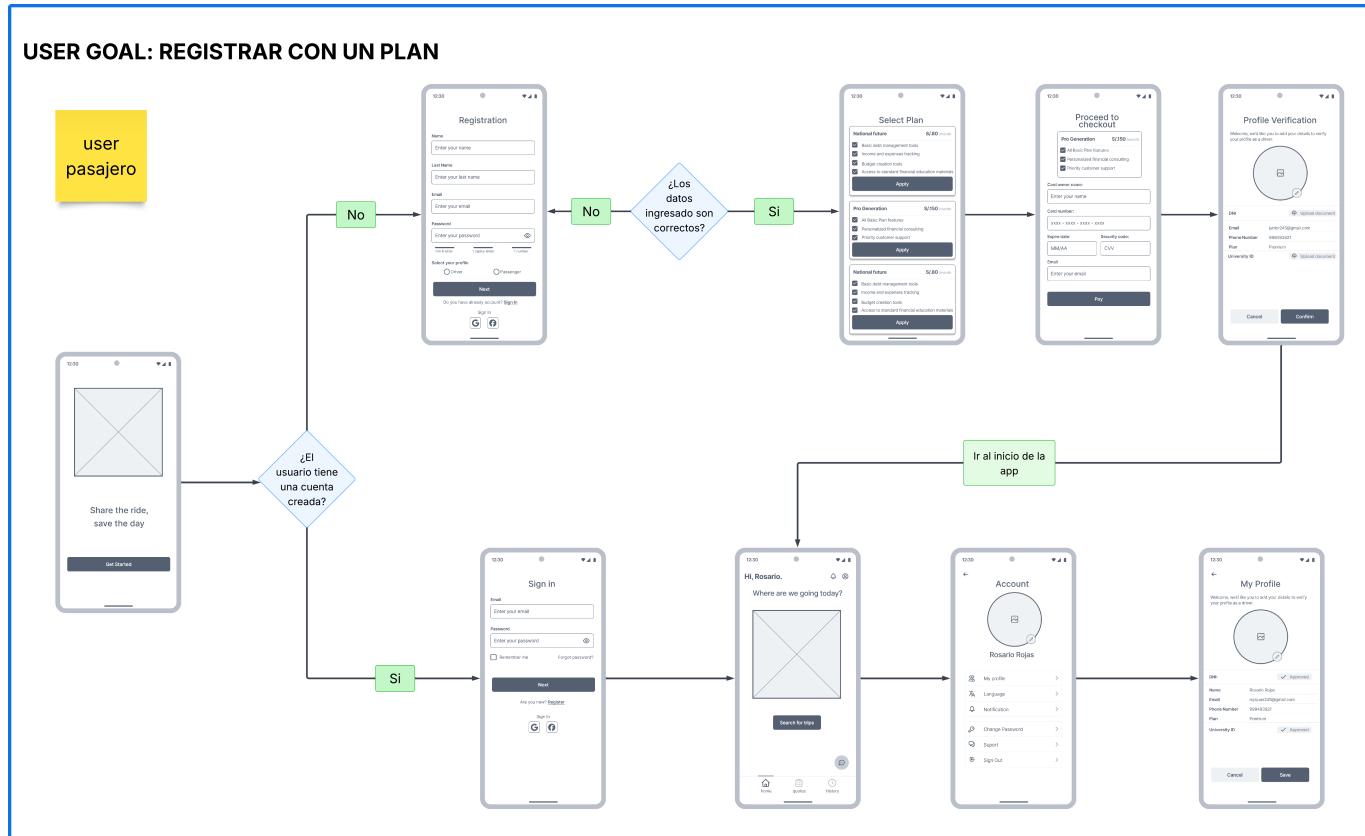
User goal: Registra con un plan



Nota: Diagrama de flujo respecto al conductor

Fuente: Elaboración Propria

## User goal: Registrar con un plan



Nota: Diagrama de flujo respecto al pasajero

Fuente: Elaboración Propia

## User goal: Recuperación de contraseña desde el inicio de sesión



Nota: Diagrama de flujo respecto al user pasajero y conductor

Fuente: Elaboración Propia

## User goal: Editar perfil desde la configuración de la cuenta



Nota: Diagrama de flujo respecto al user pasajero o conductor

Fuente: Elaboración Propia

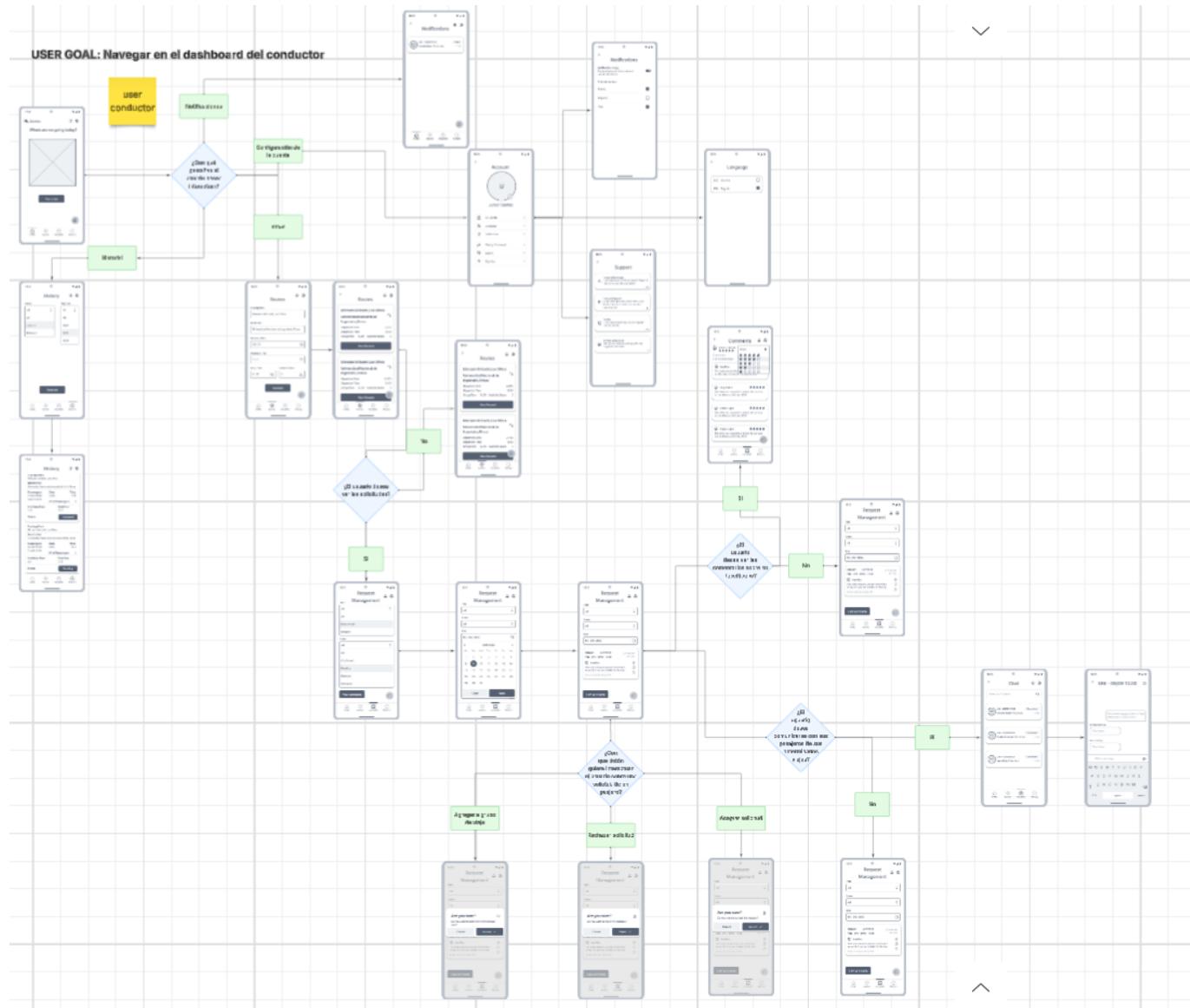
## User goal: Cambio de contraseña desde la configuración de la cuenta



Nota: Diagrama de flujo respecto al user pasajero o conductor

Fuente: Elaboración Propia

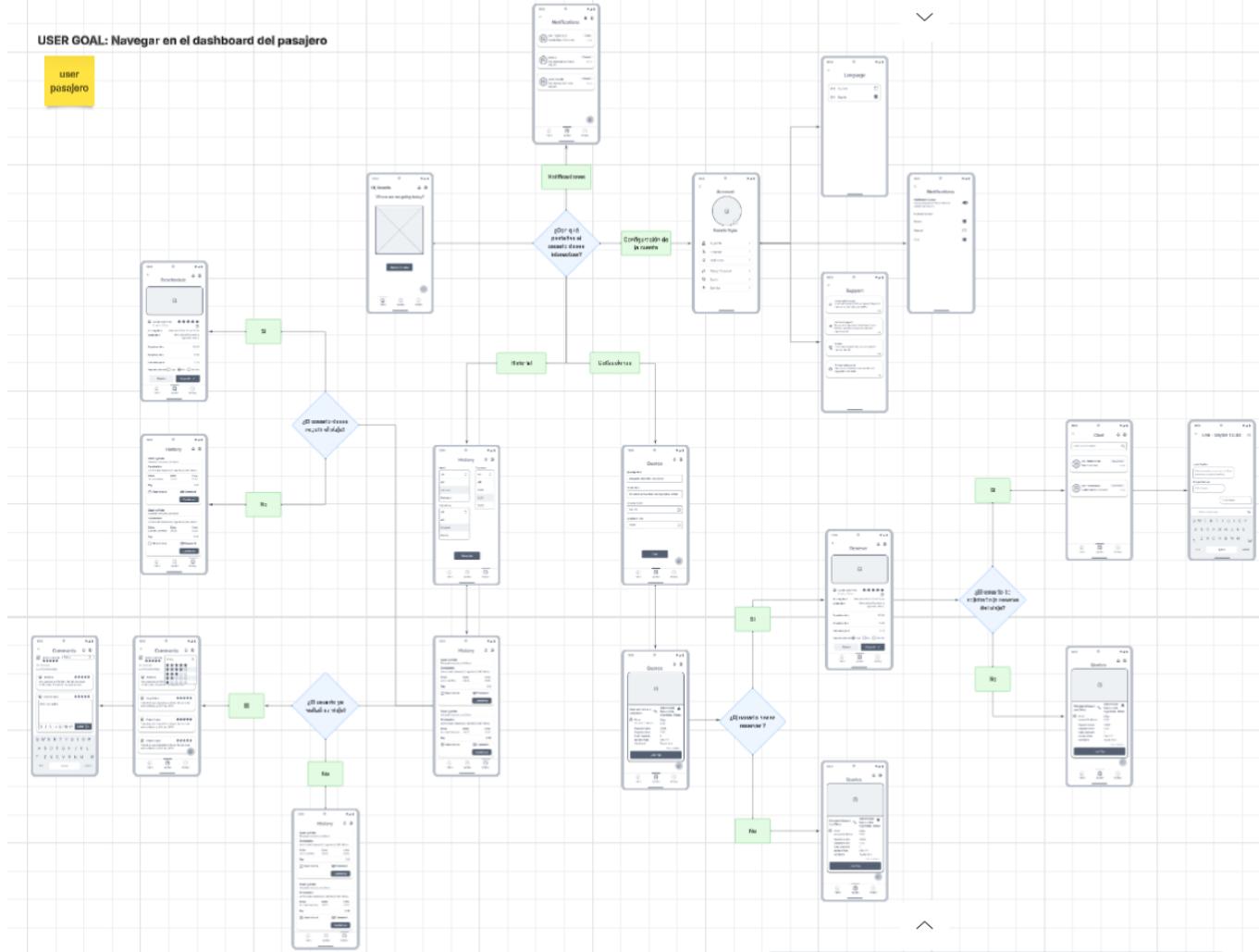
## User goal: Navegar en el dashboard del conductor



Nota: Diagrama de flujo respecto al user conductor

Fuente: Elaboración Propia

## User goal: Navegar en el dashboard de

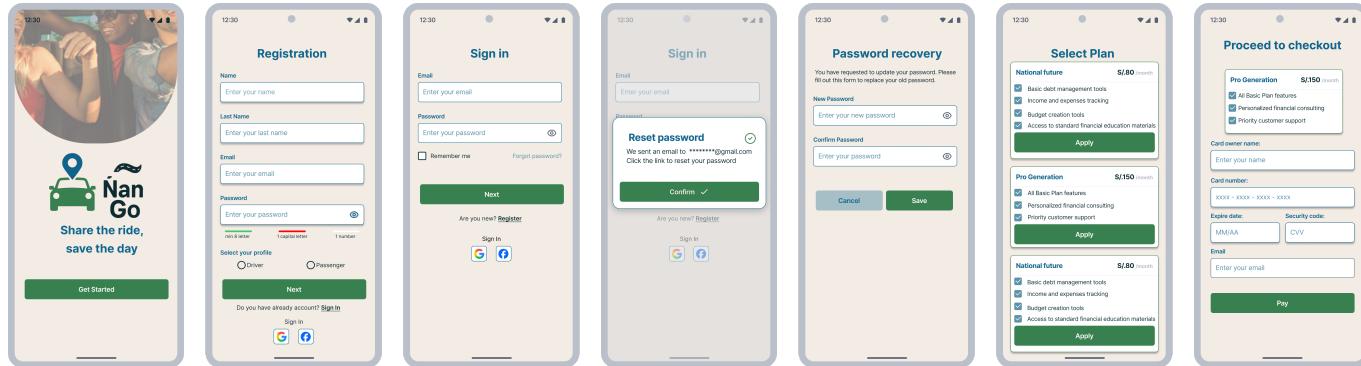


Nota: Diagrama de flujo respecto al user pasajero

Fuente: Elaboración Propia

### 3.1.4.3. Mobile Applications Mock-ups

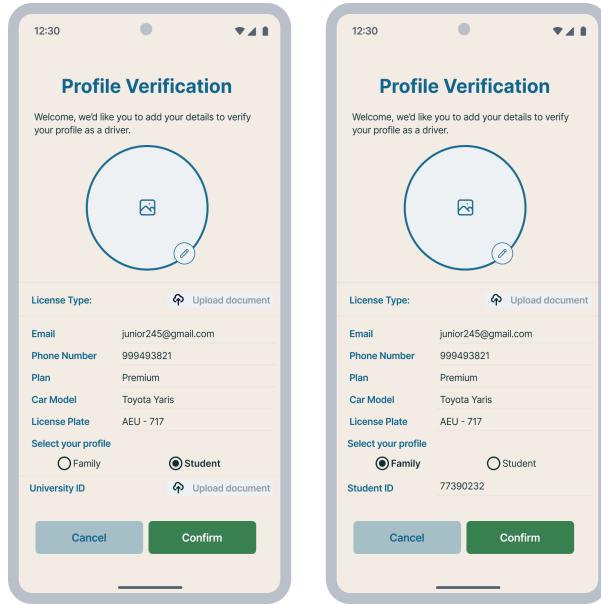
#### Registro en la aplicación móvil



Nota: Mockups del registro de la aplicación tanto para las personas que se registran por primera vez, como usuarios ya existentes, también se puede ver los mockups de la selección del plan y su pago de ella, además se ve los mockups para recuperar la contraseña desde el inicio de sesión.

Fuente: Elaboración Propia

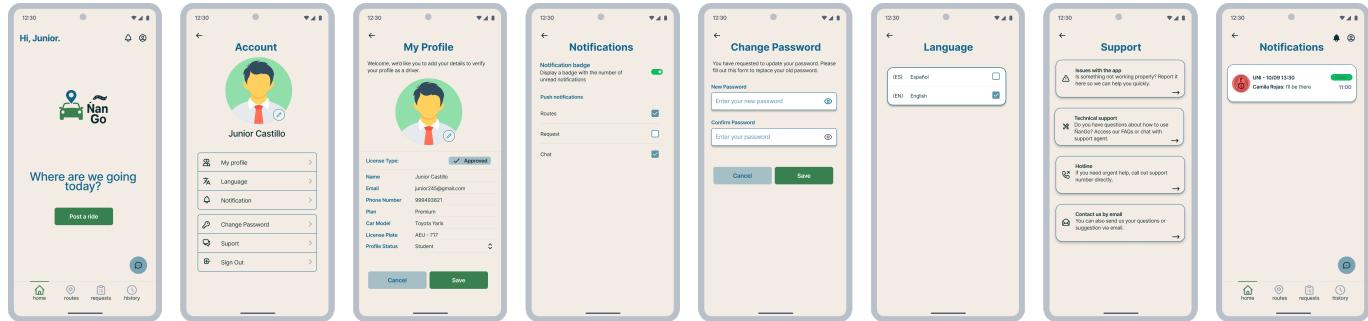
## Profile verification - Conductor



**Nota:** Acá se muestran los mockups de la verificación del perfil del conductor, para poder tener en cuenta la placa del carro, su licencia y su ID universitario.

Fuente: Elaboración Propia

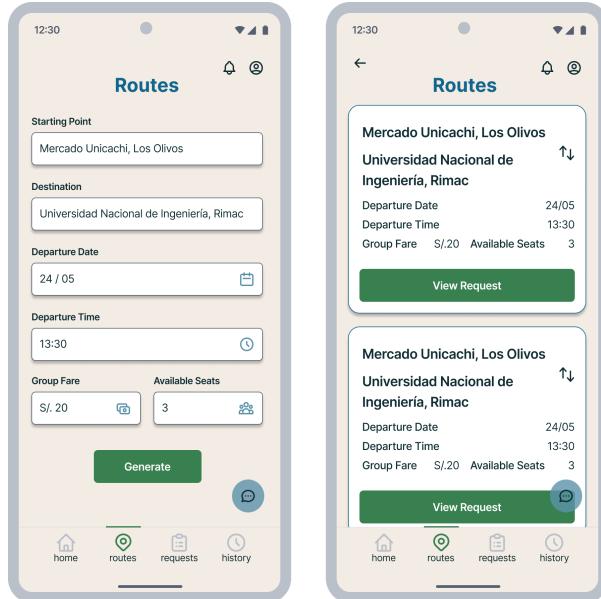
## Configuración de la cuenta - Conductor



**Nota:** Los mockups que se presentan son la configuración de la cuenta, aca se pueden ir a los otros mockups como: My profile, Notifications, Change Password, Language, Support, además tenemos el mockup de notificaciones.

Fuente: Elaboración Propia

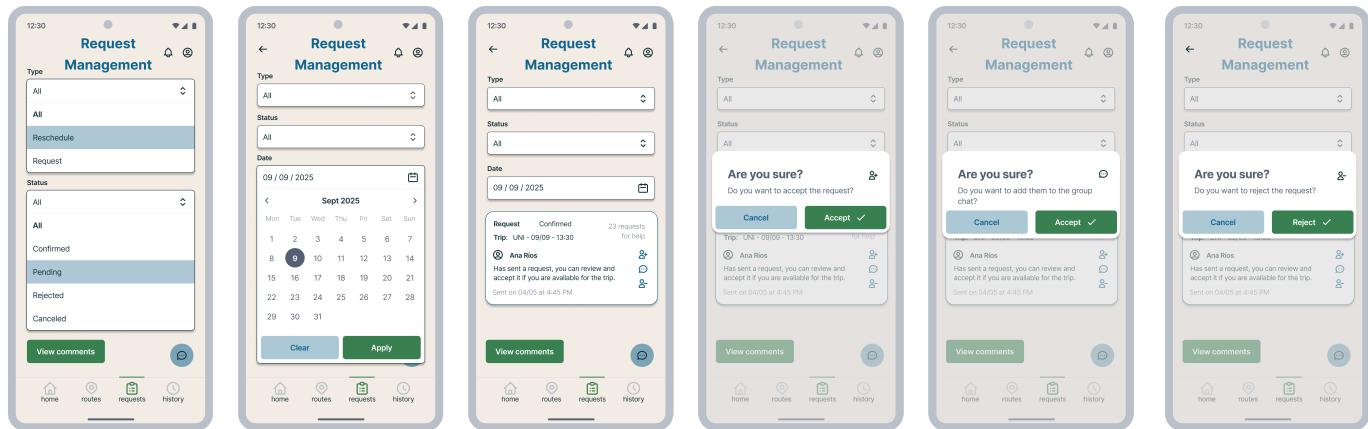
## Rutas - Conductor



**Nota:** Los mockups presentes muestra la interfaz donde el condutor puede generar el detalle de su ruta.

**Fuente:** Elaboración Propia

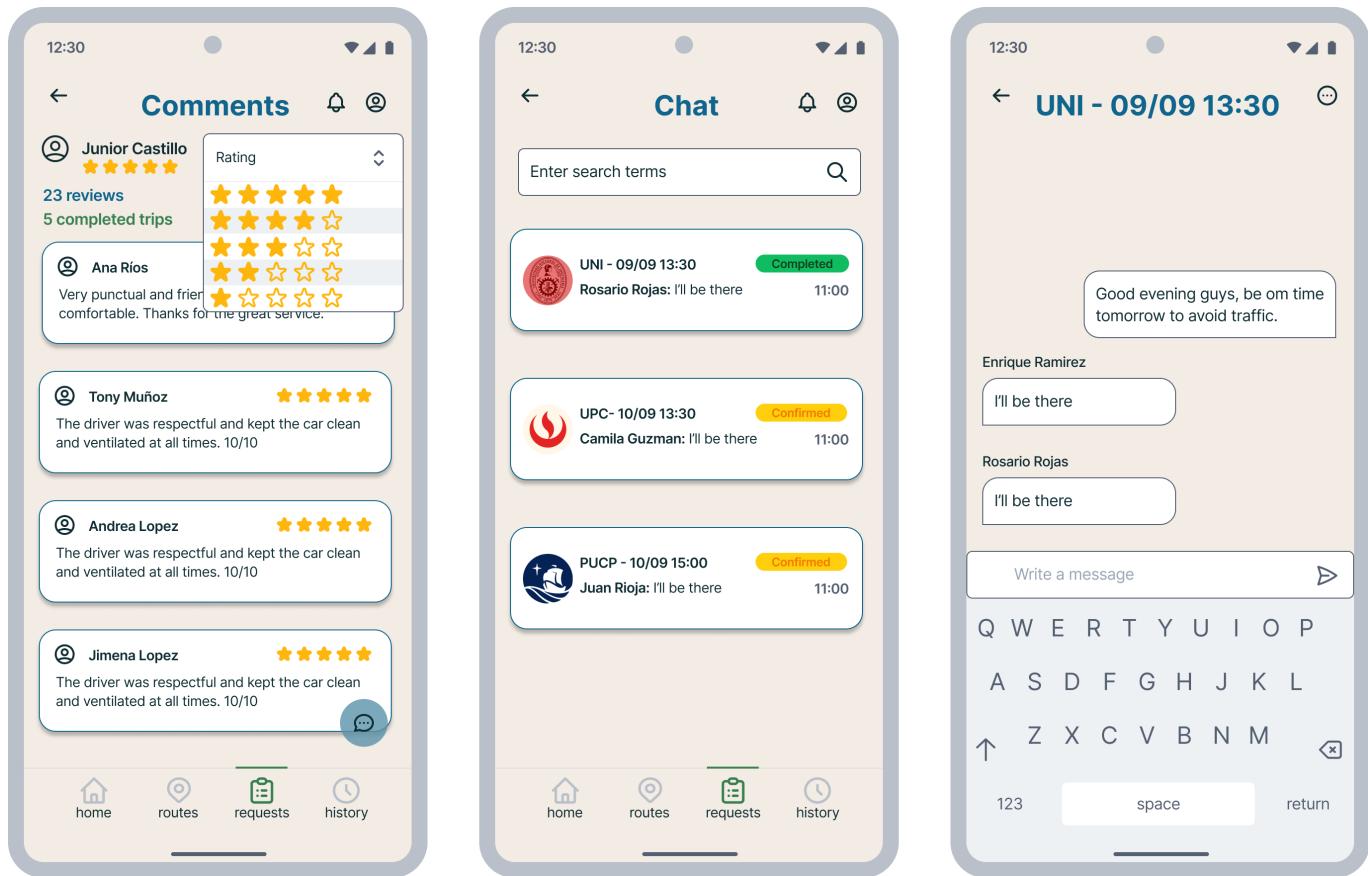
## Solicitudes - Conductor



**Nota:** Mockups que muestran el proceso de las solicitudes y reprogramaciones, junto con 3 botones que permite al conductor realizar acciones como aceptar la solicitud del viaje, añadir al pasajero al grupo de chat con otros estudiantes del viaje, y el tercero permite rechazar la solicitud del pasajero.

**Fuente:** Elaboración Propia

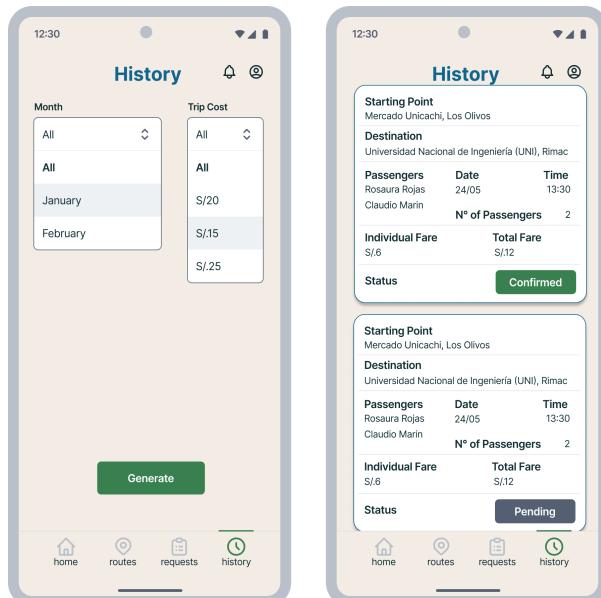
## Comentarios y chats grupales - Conductor



Nota: Mockups donde se puede visualizar los comentarios del conductor y los wireframes de los chat grupales con los pasajeros, al costado del chat se puede observar si el viaje ya fuer realizado o solo si se ha confirmado.

Fuente: Elaboración Propia

## Historial - Conductor



Nota: En estos mockups se muestran el historial de los detalles de sus rutas como los pasajeros, en donde inicio, el destino, la fecha, hora, el numero de pasajeros, el costo total y por cada pasajero y el estado de la solicitud

Fuente: Elaboración Propia

## Configuración de la cuenta - Pasajero



Nota: Mockups del perfil del pasajero, a la ruta de configuración de la cuenta, las notificaciones, para cambiar la contraseña, cambiar de idioma y el soporte de la aplicación.

Fuente: Elaboración Propia

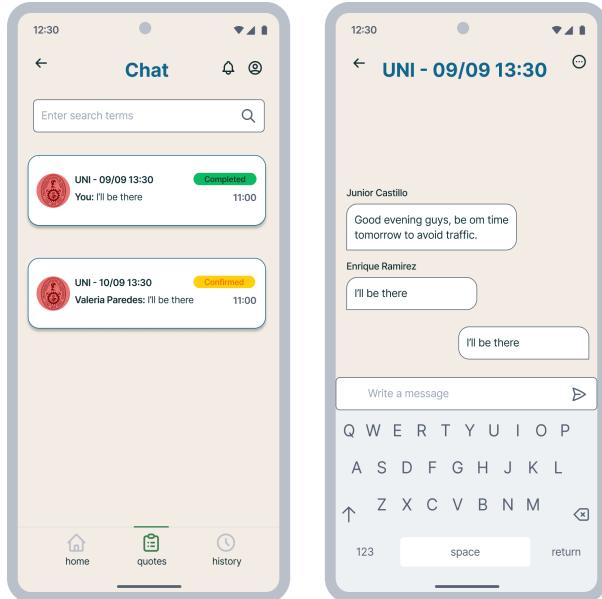
## Búsqueda de rutas - Pasajero



Nota: Mockups sobre la búsqueda de un estudiante para las rutas disponibles, y la reserva de ella, también antes de reservar se pueden observar los wireframes de los comentarios del conductor los wireframes de notificación de cuando el conductor aceptar al viaje.

Fuente: Elaboración Propia

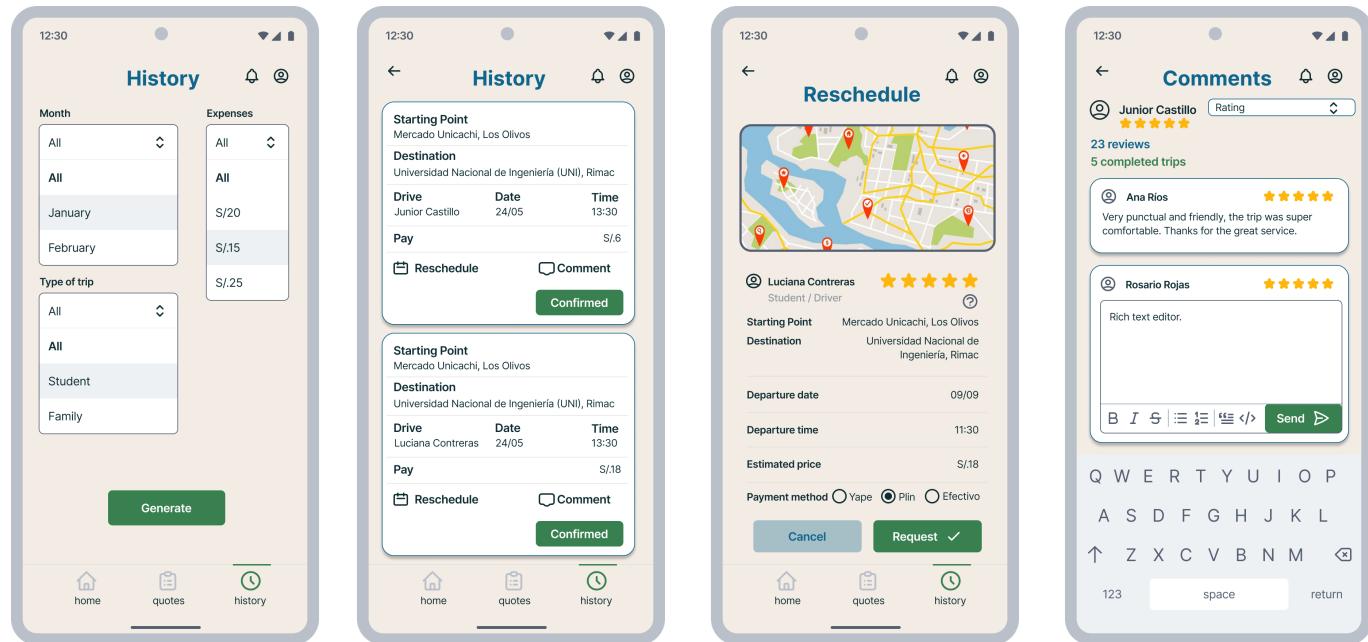
## Chat grupales - Pasajero



**Nota:** Mockups de los chat de grupos de los pasajeros con el conductor

**Fuente:** Elaboración Propia

### Histórico - Pasajero



**Nota:** Wireframes donde se muestra el histórico de los viajes anteriores de los usuarios, acá se muestran las opciones que se quieren personalizar, además información detallada del viaje como en donde inicio, el destino, el día, la hora, el nombre del conductor, además de el estado de este. También se visualiza dos mockups que es sobre volver a realizar el viaje o hacer un comentario al conductor que provienen de los botones que se muestran en el histórico.

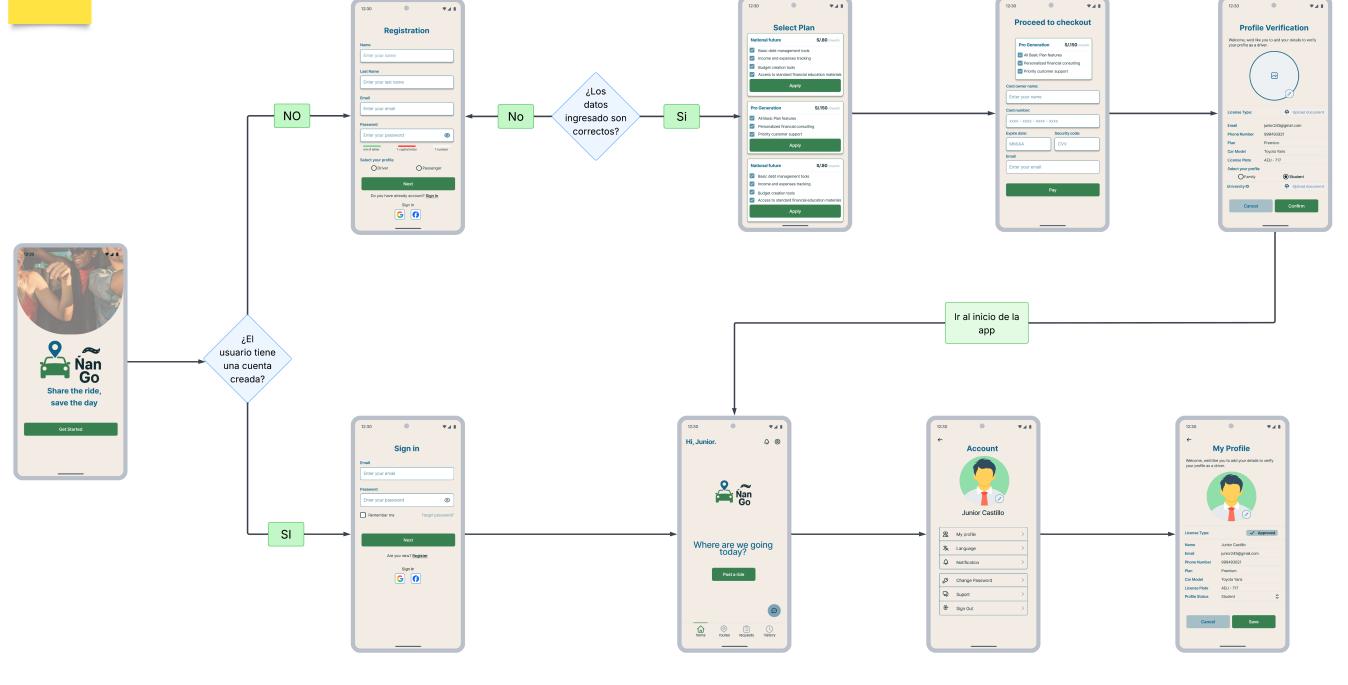
**Fuente:** Elaboración Propia

### 3.1.4.4. Mobile Applications User Flow Diagrams

## User goal: Registra con un plan

### USER GOAL: REGISTRAR CON UN PLAN

user conductor



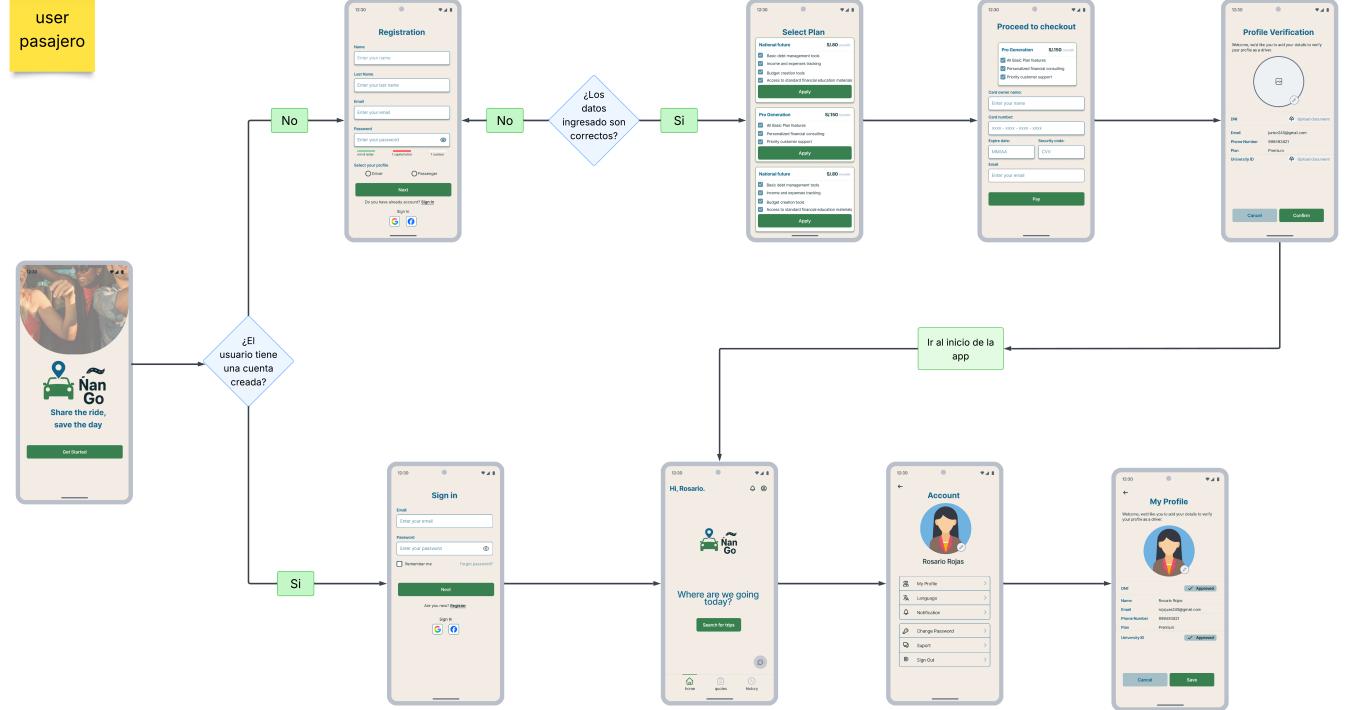
Nota: Diagrama de flujo respecto al conductor

Fuente: Elaboración Propia

## User goal: Registrar con un plan

### USER GOAL: REGISTRAR CON UN PLAN

user pasajero



Nota: Diagrama de flujo respecto al pasajero

Fuente: Elaboración Propia

**User goal: Recuperación de contraseña desde el inicio de sesión**

Nota: Diagrama de flujo respecto al user pasajero y conductor

Fuente: Elaboración Propia

## User goal: Editar perfil desde la configuración de la cuenta

### USER GOAL: EDITAR PERFIL DESDE LA CONFIGURACIÓN DE LA CUENTA

user conductor



user pasajero



Nota: Diagrama de flujo respecto al user pasajero o conductor

Fuente: Elaboración Propia

## User goal: Cambio de contraseña desde la configuración de la cuenta

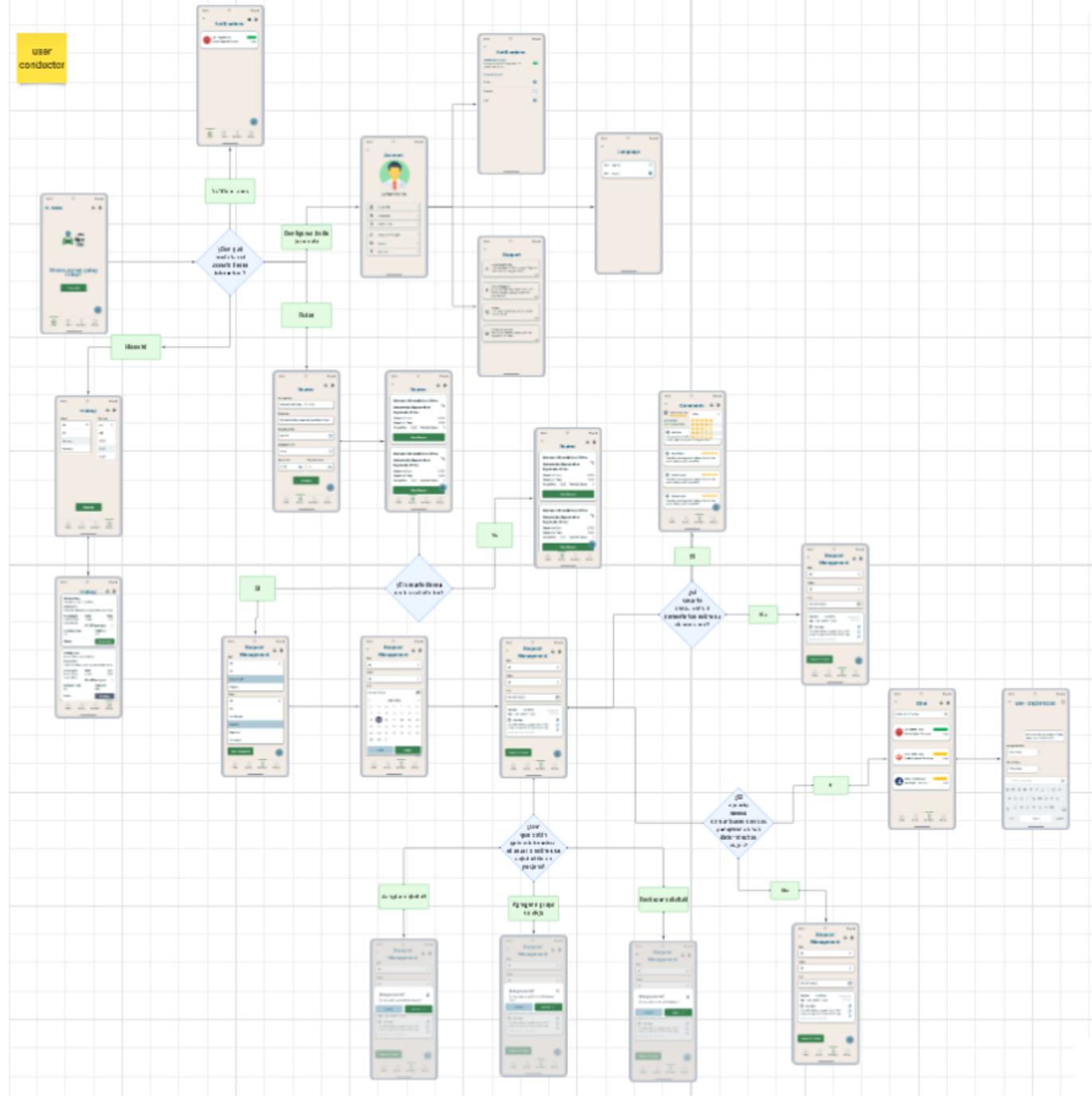


Nota: Diagrama de flujo respecto al user pasajero o conductor

Fuente: Elaboración Propia

## User goal: Navegar en el dashboard del conductor

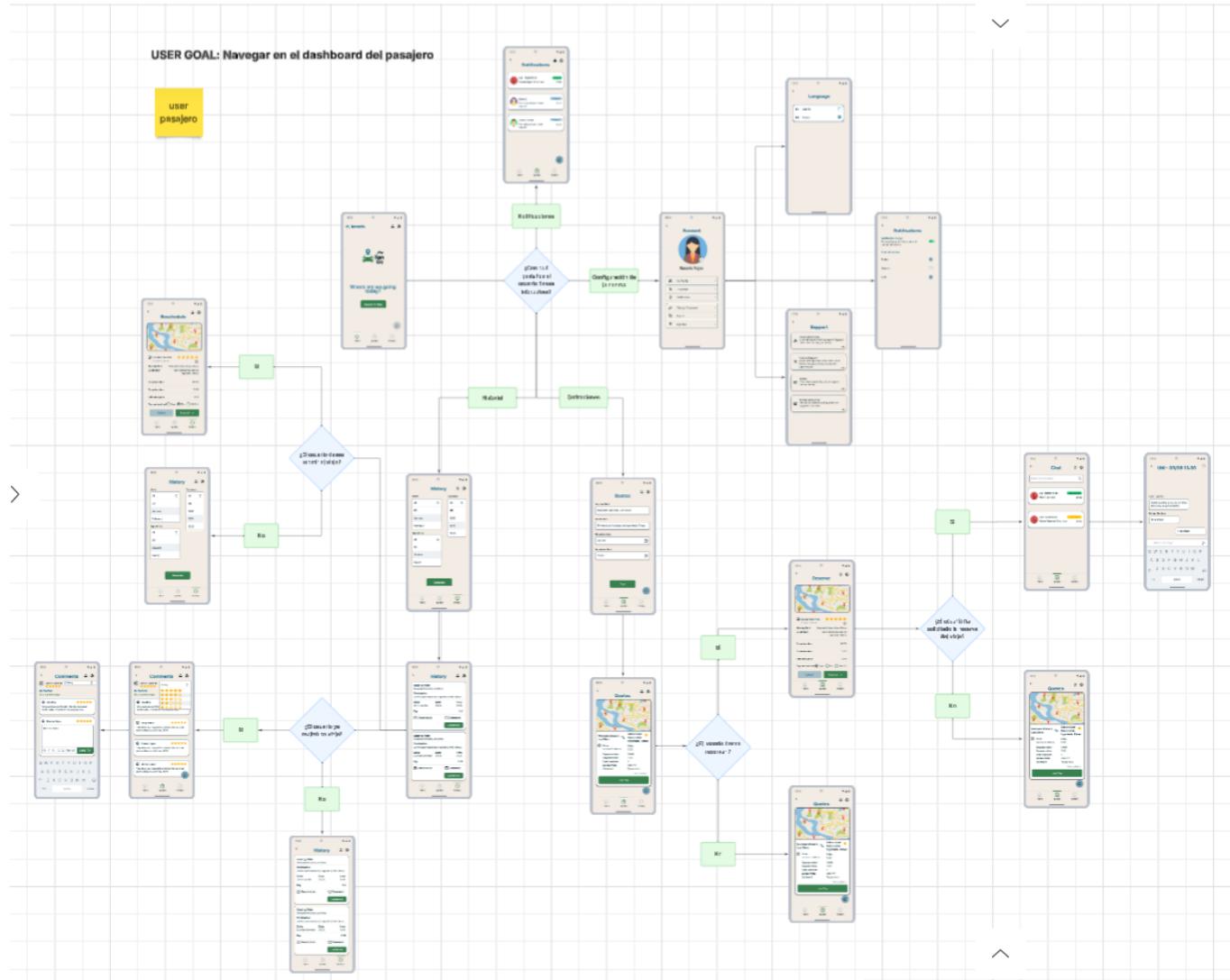
USER GOAL: Navegar en el dashboard del conductor



Nota: Diagrama de flujo respecto al user conductor

Fuente: Elaboración Propia

## User goal: Navegar en el dashboard de

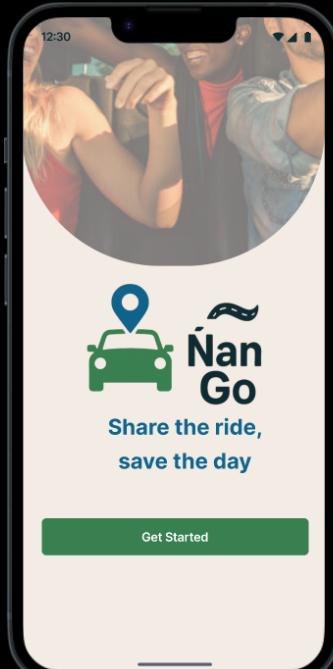


Nota: Diagrama de flujo respecto al user pasajero

Fuente: Elaboración Propia

### 3.1.4.5. Mobile Applications Prototyping

En esta sección elaboramos una muestra de lo que sería el prototipo de la aplicación móvil con interacción



Enlace a los mockups en Figma: [Mock-Up Prototype Ñango](#)

## Capítulo IV: Product Implementation & Validation

---

### 4. Product Implementation & Validation

#### 4.1. Software Configuration Management

En la sección siguiente, se describirán las herramientas, convenciones, referencias y configuraciones utilizadas durante el desarrollo del proyecto, las cuales ayudaron a mantener la coherencia en el trabajo realizado.

##### 4.1.1. Software Development Environment Configuration

En este apartado se presentan los distintos programas, plataformas y entornos de software que el equipo de desarrollo empleó para ejecutar las actividades relacionadas con la implementación del proyecto Ñango.

#### Project Management

##### Trello: <https://trello.com/>

Trello se utilizó para la gestión de tareas y la organización del flujo de trabajo del equipo. Cada tarea se representó mediante tarjetas agrupadas en listas según su estado y responsable. Además, se emplearon etiquetas de colores y checklists para visualizar el progreso, facilitando así el seguimiento y control de los avances del proyecto.

##### Google Meet: <https://meet.google.com/>

Se empleó Google Meet para realizar reuniones periódicas entre los integrantes del equipo, con el propósito

de coordinar actividades, discutir propuestas de diseño, revisar avances y definir las fechas de entrega de cada sprint.

## Requirements Management

### Trello: <https://trello.com/>

Trello también fue utilizado para la gestión de requisitos y la planificación del **product backlog**. Las historias de usuario se priorizaron utilizando la secuencia de Fibonacci, y se agruparon en épicas para mantener una estructura clara y jerárquica de las funcionalidades del sistema.

## Product UX/UI Design

### Figma: <https://www.figma.com/>

Figma fue la herramienta principal para el diseño de **wireframes, mockups y prototipos interactivos** de la aplicación móvil y la landing page de Ñango, asegurando consistencia visual y una experiencia de usuario intuitiva.

### Miro: <https://miro.com/es/>

Miro se empleó para realizar los diagramas de **As-Is y To-Be Scenario Mapping**, así como el **Strategic-Level Domain-Driven Design**, lo que permitió visualizar procesos actuales y propuestos para los distintos segmentos de usuarios objetivo.

### LucidChart: <https://www.lucidchart.com/>

LucidChart se utilizó para la elaboración de **diagramas de flujo, diagramas de arquitectura de software** y otros esquemas que describen las interacciones entre los componentes del sistema.

### Vertabelo: <https://vertabelo.com/>

Vertabelo fue empleado para el diseño de los **diagramas de bases de datos** de los principales *bounded contexts* del proyecto Ñango.

### UXPressia: <https://uxpressia.com/>

UXPressia permitió crear **User Personas, Empathy Maps, Journey Maps e Impact Maps**, empleando plantillas visuales que facilitaron la documentación y exportación del trabajo de diseño UX.

## Software Development

### Landing Page

La landing page de Ñango se desarrolló utilizando **HTML, CSS y JavaScript**, aplicando estilos personalizados para garantizar un diseño atractivo, responsive y coherente con la identidad visual del proyecto. El sitio fue optimizado para su despliegue en **GitHub Pages**.

### Frontend móvil (Android Studio)

Para el segmento de usuarios con dispositivos Android, se desarrolló un **frontend nativo en Android Studio**, utilizando **XML** para la definición de interfaces gráficas y **Kotlin** para la lógica de la aplicación. Esto permitió una experiencia fluida y una integración nativa con el ecosistema Android.

### Backend

El backend del proyecto Ñango se implementó con **Python** y las librerías **Fastapi, Pydantic, SQLAlchemy**,

exponiendo **APIs RESTful** conectadas a una base de datos **PostgreSQL**. Este componente maneja la lógica del negocio, la validación de datos y la comunicación segura entre el frontend y el servidor.

## Software Testing

Para la validación funcional de la **Landing Page**, el **Backend** y las **aplicaciones móviles**, se utilizaron herramientas de desarrollo integradas en navegadores como **Google Chrome**.

## Software Deployment

**Dokploy:** <https://dokploy.com/>

Todos los servicios fueron desplegados en **Dokploy**, integrando el repositorio del proyecto para realizar despliegues automatizados. Esta plataforma permitió configurar fácilmente el entorno de producción, gestionar variables de entorno y mantener un flujo de entrega continua (**CI/CD**) sin necesidad de configuraciones adicionales manuales.

## Software Documentation

**Structurizr:** <https://structurizr.com/>

Structurizr se empleó para generar **diagramas C4** (contexto, contenedor y componentes), permitiendo documentar la arquitectura del sistema de forma clara y actualizable.

**GitHub:** <https://github.com/>

GitHub fue la plataforma central del proyecto Ñango, albergando el código de la **Landing Page**, el **Backend**, la **aplicación móvil** y la **documentación**. Su sistema de control de versiones facilitó la colaboración y el seguimiento de cambios.

### 4.1.2. Source Code Management

La administración y versionado del código fuente se realizó a través de varios repositorios alojados en **GitHub**, bajo la organización oficial del proyecto:

**Organización:** <https://github.com/UniLink-Grupo3>

- **Repositorio de documentación:** [link](#)
- **Repositorio de la Landing Page:** [link](#)
- **Repositorio de la Aplicación Móvil:** [link](#)

Para el control de versiones, se empleó el modelo **Gitflow**, que permite estructurar las ramas y mantener un flujo de trabajo ordenado:

- **main:** Contiene las versiones estables listas para producción.
- **feature:** Se crean para el desarrollo de nuevas funcionalidades o mejoras específicas (por ejemplo: [feature/auth-module](#)).

Este enfoque permitió mantener control sobre las versiones del código, minimizar conflictos y garantizar un desarrollo colaborativo eficiente.

### 4.1.3. Source Code Style Guide & Conventions

Para mantener la coherencia del código en los distintos componentes del proyecto Ñango, se establecieron las siguientes guías y convenciones:

- **HTML**: para la estructura de la Landing Page.
- **CSS**: para los estilos visuales y el diseño responsivo.
- **TypeScript**: para la interactividad y las funcionalidades dinámicas.
- **Kotlin**: para el desarrollo nativo de la aplicación móvil Android.
- **Python**: para el backend y la creación de APIs RESTful conectadas a PostgreSQL.

#### 4.1.4. Software Deployment Configuration

Para el despliegue de la **Landing Page** del proyecto **Ñango**, se utilizó la plataforma **Dokploy**, empleando como fuente una imagen previamente construida y publicada en el **GitHub Container Registry (GHCR)**. Esta estrategia permitió automatizar el proceso de implementación y garantizar que el entorno de producción ejecutara exactamente la misma versión empaquetada durante la fase de desarrollo.

🔗 [\\_Ñango Landing Page – Dokploy](#)

## 4.2. Landing Page & Mobile Application Implementation

### 1. Publicación automática de la imagen en GitHub Container Registry (GHCR)

El proceso de construcción y publicación de la imagen Docker de la **Landing Page de Ñango** se encuentra completamente automatizado mediante un **workflow de GitHub Actions**.

Cada vez que se realizan cambios en la rama principal (**main**) y se crea una nueva etiqueta de versión (**tag**), el pipeline se ejecuta automáticamente, generando una nueva imagen y publicándola en el **GitHub Container Registry (GHCR)** bajo la organización del proyecto.

Este flujo garantiza la trazabilidad entre el código fuente y las versiones desplegadas, además de facilitar la integración continua (**CI/CD**) con Dokploy.

El proceso automatizado sigue las siguientes etapas:

#### 1. Detección del evento de despliegue:

El workflow se activa automáticamente cuando se realiza un *push* en la rama **main** o cuando se crea una etiqueta de versión (por ejemplo, **v1.0.0**).

#### 2. Construcción de la imagen Docker:

GitHub Actions ejecuta el comando de construcción dentro del pipeline:

```
docker build -t ghcr.io/unilink-grupo3/nango-landing-page:${{  
github.ref_name }} .
```

Aquí, **\${{ github.ref\_name }}** corresponde al nombre del tag o rama que generó el evento.

### 3. Autenticación en GHCR:

El workflow utiliza un **token de acceso personal (PAT)** configurado como secret dentro del repositorio ([GHCR\\_TOKEN](#)) para autenticarse de manera segura con el registro:

```
echo ${{ secrets.GHCR_TOKEN }} | docker login ghcr.io -u ${{ github.actor }} --password-stdin
```

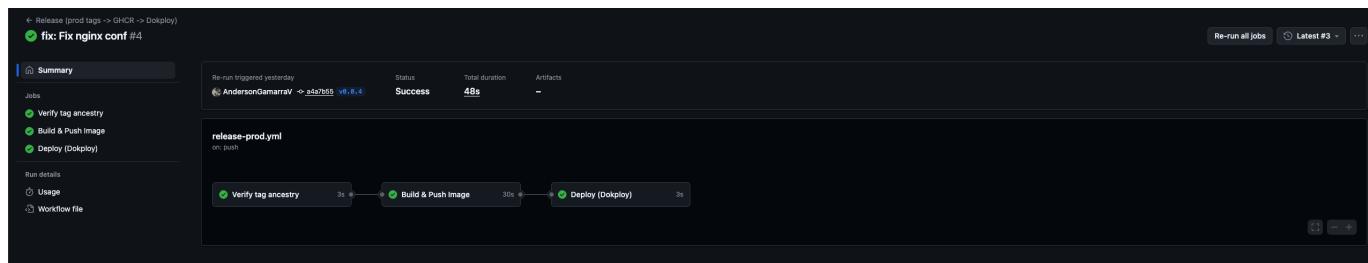
### 4. Publicación automática de la imagen:

Finalmente, la imagen se publica en el **GitHub Container Registry** bajo el nombre:

```
ghcr.io/unilink-grupo3/nango-landing-page:latest
ghcr.io/unilink-grupo3/nango-landing-page:${{ github.ref_name }}
```

Esto permite mantener una versión "latest" para el despliegue continuo y versiones específicas etiquetadas por cada release.

De esta manera, cada versión estable de la **Landing Page** queda automáticamente empaquetada y disponible en el registro de contenedores, lista para ser desplegada en **Dokploy** sin intervención manual.



## 2. Creación del servicio en Dokploy

Desde el panel principal de **Dokploy**, se seleccionó la opción “**New Project**” → “**Deploy from Container Image**”.

En el campo de configuración de la imagen, se especificó la ruta al contenedor publicado en GitHub Container Registry:

```
ghcr.io/unilink-grupo3/nango-landing-page:latest
```

Además, se configuraron los siguientes parámetros:

- **Nombre del servicio:** [nango-landing-page](#)
- **Puerto expuesto:** [80](#) o [5173](#) (según el Dockerfile del proyecto)
- **Auto-redeploy:** activado, para que cada nueva versión de la imagen publicada en GHCR se despliegue automáticamente.

The screenshot shows the Dokken interface for deploying a landing page. At the top, there's a navigation bar with tabs: General, Environment, Domains, Deployments, Preview Deployments, Schedules, Volume Backups, Logs, Monitoring, and Advanced. The Deployments tab is selected. Below the navigation is a 'Deploy Settings' section with buttons for Deploy, Reload, Rebuild, Stop, Open Terminal, Autodeploy (disabled), and Clean Cache. A 'Provider' section allows selecting code sources like GitHub, Gitlab, Bitbucket, Gitea, Docker, and Git, with Docker being selected. It also includes fields for Registry URL (https://ghcr.io) and Username (AndersonGamarraV). A 'Build Type' section notes that builders consume significant resources and recommends 4+ GB RAM and 2+ CPU cores. A 'Save' button is at the bottom right.

## 4.2.1. Sprint 1

Podemos considerar al Sprint Planning como una "ceremonia" fundamental dentro de la metodología ágil Scrum. Su propósito es definir el objetivo del sprint y planificar el trabajo que se realizará durante el mismo. En esta primera planificación, como detallaremos en adelante, el equipo se reunió para analizar los requerimientos del proyecto, priorizar las tareas del backlog y asignar responsabilidades entre los integrantes. Esta sesión permitió establecer una visión compartida sobre los entregables a corto plazo y sentar las bases del desarrollo inicial de la aplicación móvil, partiendo del desarrollo del landing page.

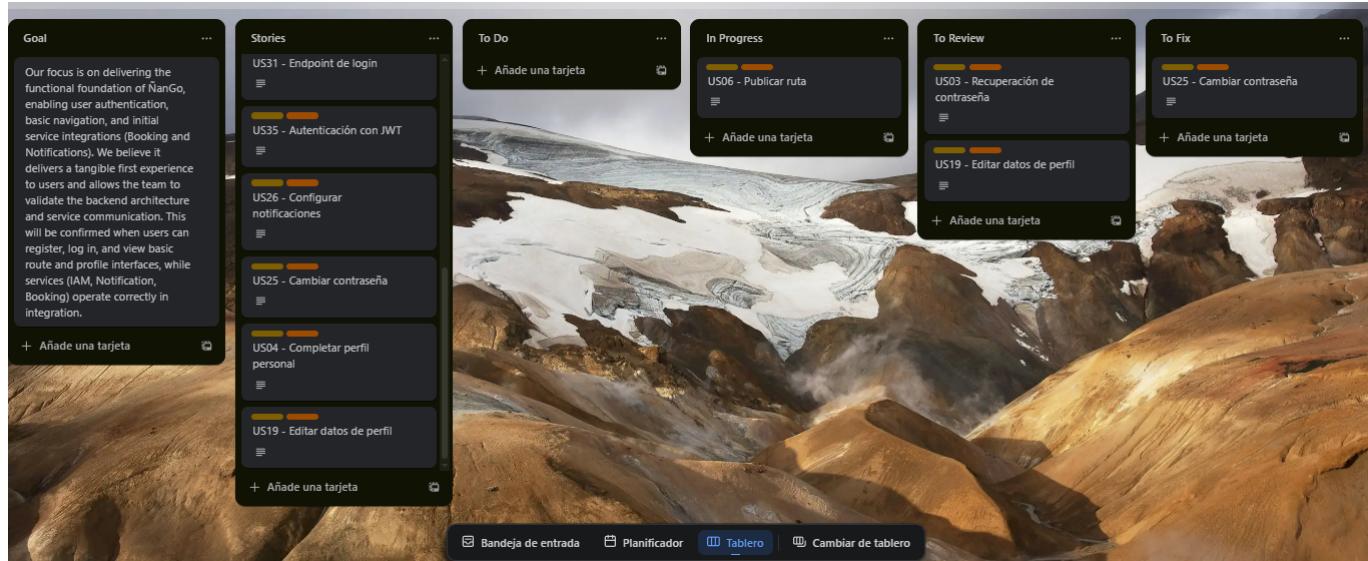
### 4.2.1.1. Sprint Planning 1

|  |   |
|--|---|
| <b>Sprint #</b>                        | 1   |
| <b>Sprint Planning Background</b>      |   |
| <b>Date</b>                            | 05/10/2025  |
| <b>Time</b>                            | 8:00 pm   |
| <b>Location</b>                        | Reunión virtual a través de Google Meet   |
| <b>Prepared by</b>                     | Gamarra Vega, Anderson Jose   |
| <b>Attendees (to planning meeting)</b> | Castañeda Llanos, Kevin Alexander, Peña Riofrio, Maria Fernanda, Grandez Mansilla, Jean Pierre, Guevara Tejada, Jorge Enrique                                   |
| <b>Sprint 1 Review Summary</b>         | Completar el desarrollo de los capítulos 3 y 4 del reporte. Codificación y despliegue del Landing Page. Creación de frontend de secciones core y backend al 70% |

| Sprint #  | 1  |
|---|--|
| <b>Sprint 1 Retrospective Summary</b>   | Se realizó la distribución de temas para completar la documentación, además de diseñar e implementar la primera versión de nuestro landing page, creación de las secciones core de la aplicación móvil, y la creación del backend. Se logró el 70% del desarrollo. |
| <b>Sprint Goal &amp; User Stories</b>   |  |
| <p><b>Nos enfocamos</b> en diseñar y desarrollar la landing page de Ñango y la creación de secciones del frontend, y la creación de algunos bounded context en el backend.</p> <p><b>Creemos que</b> esto dará una idea inicial de como servirá y los objetivos que pretendemos cumplir con nuestra aplicación móvil. <b>Esto se confirmará</b> cuando la Landing Page, frontend y backend estén listos para el despliegue y en las entrevistas de validación, sean positivas con respecto a nuestro proyecto</p> |  |
| <b>Sprint n Velocity</b>  | 1 semana   |
| <b>Sum of Story Points</b>  | 20   |

#### 4.2.1.2. Sprint Backlog 1

El objetivo principal de este Sprint fue avanzar en el desarrollo de las funcionalidades clave de nuestra solución, priorizando aquellas que nos permitieran entregar una versión funcional inicial de la Landing Page y establecer las bases para el desarrollo posterior del sistema, creando secciones core en el frontend y cumpliendo con el 70% del backend. Para organizar y dar seguimiento a las tareas, utilizamos la herramienta Trello como nuestro tablero de control, donde cada integrante podía visualizar, actualizar y colaborar en las actividades asignadas. A continuación, se presenta una captura del tablero correspondiente al sprint 1:



Enlace al trello:

<https://trello.com/invite/b/68e82abdd78299ef5e5d3523/ATTI0dc5611a4631c610dcf19bb2b603aafb35EFC687/nango-sprint>

| Sprint #   |                             | Sprint 1  |                   |             |        |  |
|------------|-----------------------------|---|-------------------|-------------|--------|--|
| User Story |                             | Work-Item / Task                                    |                   |             |        |  |
| ID         | Title                       | Description   | Estimated (Hours) | Assigned To | Status |  |
| US01       | Registro de cuenta          | Crear la opción de crear una nueva cuenta           | 1                 | Anderson    | Done   |  |
| US02       | Inicio de sesión            | Crear la opción de iniciar sesión                   | 1                 | Anderson    | Done   |  |
| US03       | Recuperación de contraseña  | Crear la opción de recuperar la contraseña          | 1                 | Anderson    | Done   |  |
| US05       | Buscar rutas disponibles    | Ver las rutas disponibles para los pasajeros        | 2                 | Kevin       | Done   |  |
| US09       | Solicitar unirse a una ruta | Pasajero que quiera realizar una reserva            | 2                 | Kevin       | Done   |  |
| US06       | Publicar Ruta               | Crear la opción para que un conductor cree una ruta | 2                 | -           | To Do  |  |
| US26       | Configurar Notificaciones   | Crear la configuración del gestor de notificaciones | 1                 | Jean        | Done   |  |
| US25       | Autenticación con JWT       | Configurar la autenticación de usuario con JWT      | 2                 | Anderson    | Done   |  |

| Sprint # | Sprint 1                 |   |   |          |      |  |
|----------|--------------------------|---|---|----------|------|--|
| US27     | Ver información de Ñango | Como usuario quiero ver la info. de Ñango en la Landing Page            | 1 | Anderson | Done |  |
| US30     | Endpoint de login        | Como developer quiero exponer un endpoint de login para iniciar sesión. | 1 | Anderson | Done |  |
| US33     | Endpoint de solicitudes  | Como developer quiero exponer un endpoint para gestionar solicitudes.   | 1 | Kevin    | Done |  |

#### 4.2.1.3. Development Evidence for Sprint Review

| Repository   | Branch | Commit Id | Commit Message                                   | Commit Message Body   | Committed on (Date) |
|--------------|--------|-----------|--|---|---------------------|
| Landing-page | prod   | 46bbe04   | first commit and add project structure           | -   | 07/10/2025          |
| Landing-page | prod   | a4a7b55   | fix: Fix nginx conf                              | Corrección de configuración Nginx para despliegue productivo.     | 07/10/2025          |
| Landing-page | prod   | 4703af8   | chore: change container name prod                | Se actualizó el nombre del contenedor para entorno de producción. | 07/10/2025          |
| Landing-page | prod   | f288caf   | chore: add deploy job on pipeline                | Se añadió job de despliegue en el pipeline CI/CD.                 | 07/10/2025          |
| Landing-page | prod   | 6b7a542   | fix: Remote sha tag for image and add latest tag | Actualización de etiquetas de imágenes Docker.                    | 07/10/2025          |
| Landing-page | prod   | 9a079e8   | add docker compose file                          | Se añadió archivo docker-compose para orquestar servicios.        | 07/10/2025          |

| Repository           | Branch       | Commit Id | Commit Message  | Commit Message Body  | Committed on (Date) |
|----------------------|--------------|-----------|---|--|---------------------|
| Landing-page         | prod         | e96572f   | add verify and build pipeline                                 | Configuración de pasos de verificación y build en el pipeline.                                   | 07/10/2025          |
| Frontend             | main         | a9cf05e   | Subiendo vistas: login, profilevalidation, home y routes      | Se agregaron las vistas principales del frontend: login, registro, perfil, home y rutas básicas. | 08/10/2025          |
| Notification Service | notification | 1bd8b7f   | feat(app): setup main application and dependencies            | Se configuró la aplicación principal en FastAPI con las dependencias necesarias.                 | 09/10/2025          |
| Notification Service | notification | 2702663   | feat(config): add configuration and dependency injection      | Se agregó configuración inicial y gestión de dependencias para el servicio.                      | 09/10/2025          |
| Notification Service | notification | 12b1e98   | feat(api): implement REST API endpoints with FastAPI          | Se implementaron endpoints principales del servicio Notification.                                | 09/10/2025          |
| Notification Service | notification | 0c0e8eb   | feat(infrastructure): implement infrastructure adapters       | Se añadieron adaptadores de infraestructura (repositorios y persistencia).                       | 09/10/2025          |
| Notification Service | notification | 4d2ccf9   | feat(application): implement application layer with use cases | Se implementó la capa de aplicación con casos de uso definidos.                                  | 09/10/2025          |

| Repository           | Branch       | Commit Id | Commit Message  | Commit Message Body  | Committed on (Date) |
|----------------------|--------------|-----------|---|--|---------------------|
| Notification Service | notification | 1a6f13c   | feat(domain): implement domain layer with DDD patterns      | Implementación de la capa de dominio con entidades bajo arquitectura DDD.  | 09/10/2025          |
| Booking Service      | bookings     | 03db29b   | add first version of booking service (only post method)     | Se implementó endpoint inicial POST para creación de reservas.             | 08/10/2025          |
| Booking Service      | bookings     | 7bccd94   | Add first version of booking-service (post, delete, getAll) | Se ampliaron endpoints para CRUD completo (POST, DELETE, GETALL).          | 09/10/2025          |
| Booking Service      | bookings     | 5754776   | Update .gitignore   | Se actualizó configuración del repositorio.                                | 09/10/2025          |
| IAM Service          | main         | ae2494f   | first commit  | Configuración inicial del servicio de autenticación.                       | 08/10/2025          |
| IAM Service          | main         | 75835b5   | add fastapi dependency                                      | Se añadió dependencia principal de FastAPI.                                | 08/10/2025          |
| IAM Service          | main         | 0e9a584   | Add structure project                                       | Se estructuró el proyecto con sus carpetas base (app, routes, core, etc.). | 08/10/2025          |
| IAM Service          | main         | 49ab2f9   | Change steps for init service mode dev                      | Ajuste de pasos para ejecución en modo desarrollo.                         | 08/10/2025          |
| IAM Service          | main         | 73c526b   | add service v0.1  | Versión inicial funcional del servicio IAM con endpoints básicos.          | 09/10/2025          |

| Repository  | Branch | Commit Id | Commit Message                              | Commit Message Body  | Committed on (Date) |
|-------------|--------|-----------|---|--|---------------------|
| IAM Service | main   | 643a047   | ci: change webhook url to deploy on dokploy | Configuración del webhook para despliegue automático del servicio IAM. | 09/10/2025          |

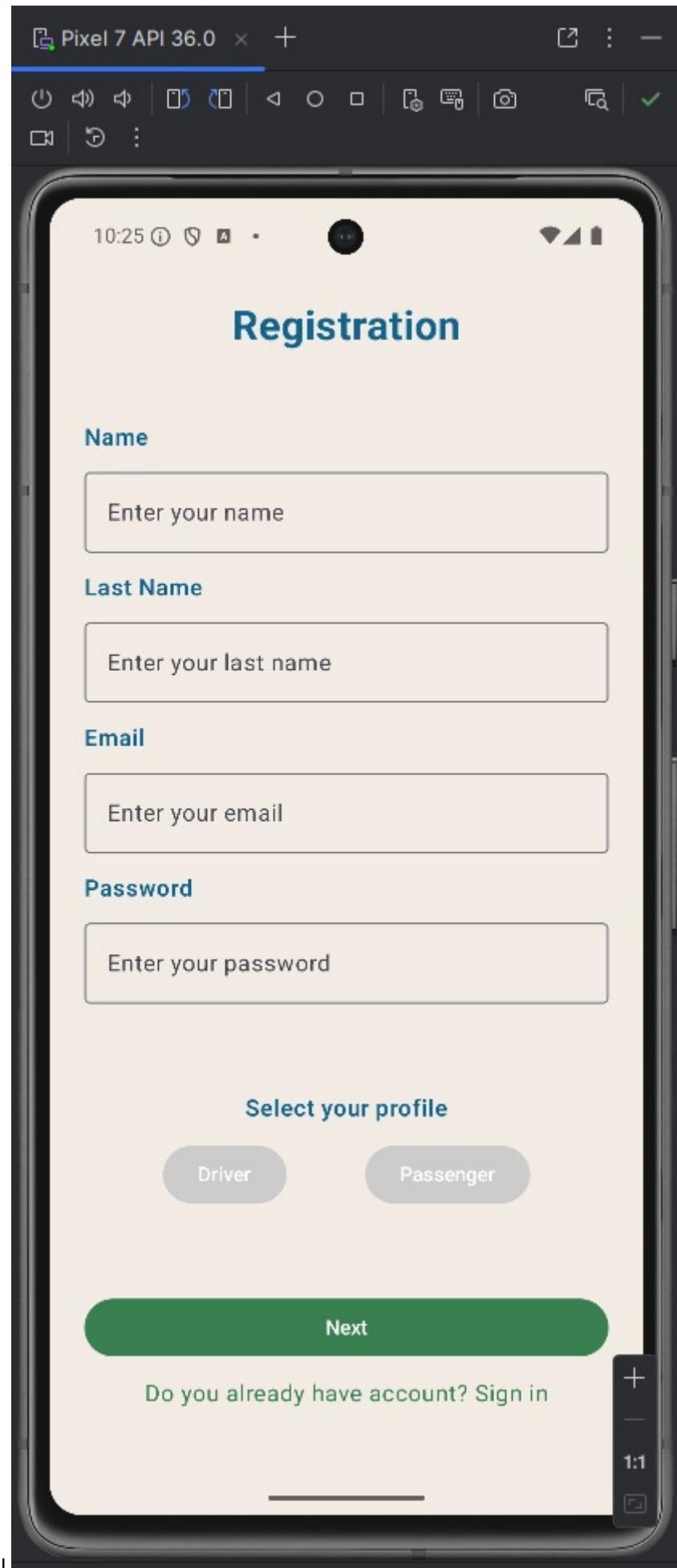
#### 4.2.1.4. Testing Suite Evidence for Sprint Review

#### 4.2.1.5. Execution Evidence for Sprint Review

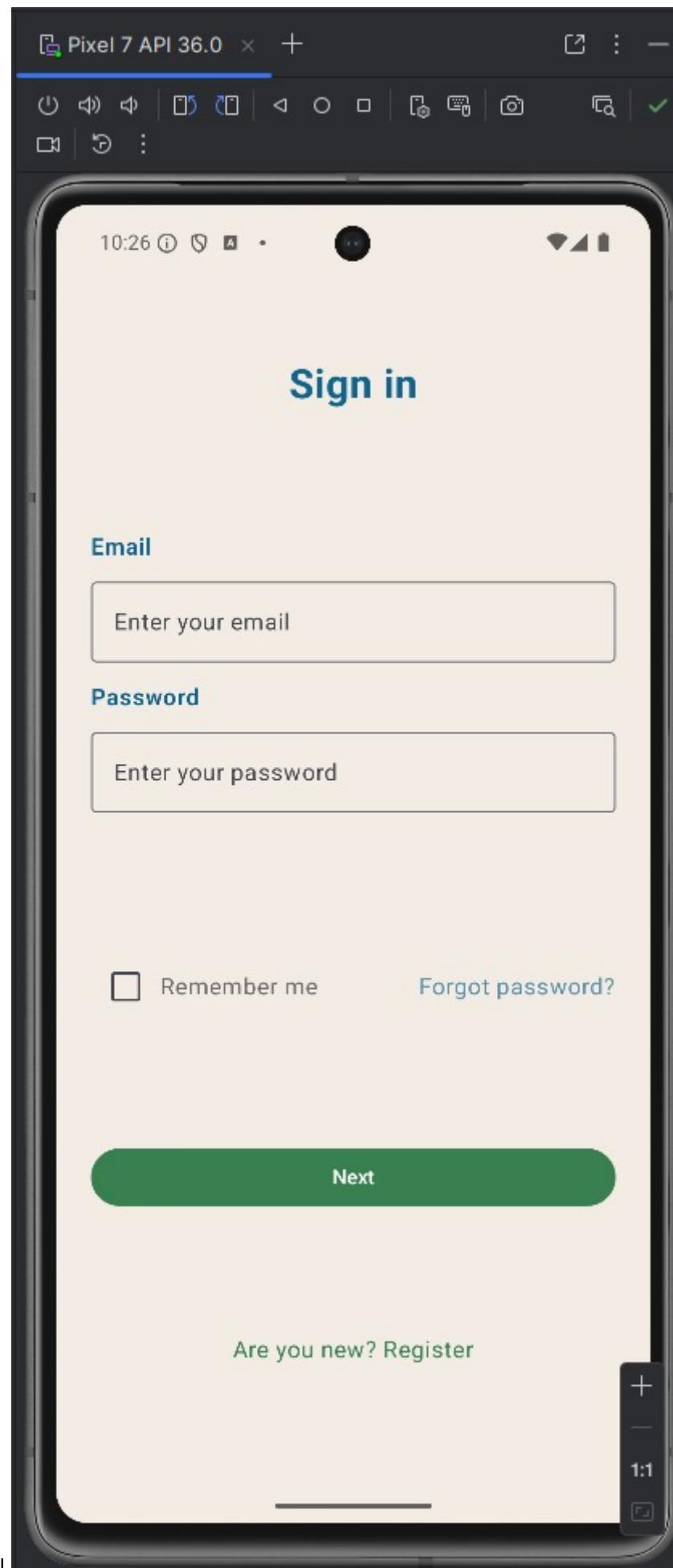
En esta sección se presentan evidencias de la ejecución de la App Movil y la Landing Page del proyecto Ñango



- Primera View de la App Móvil

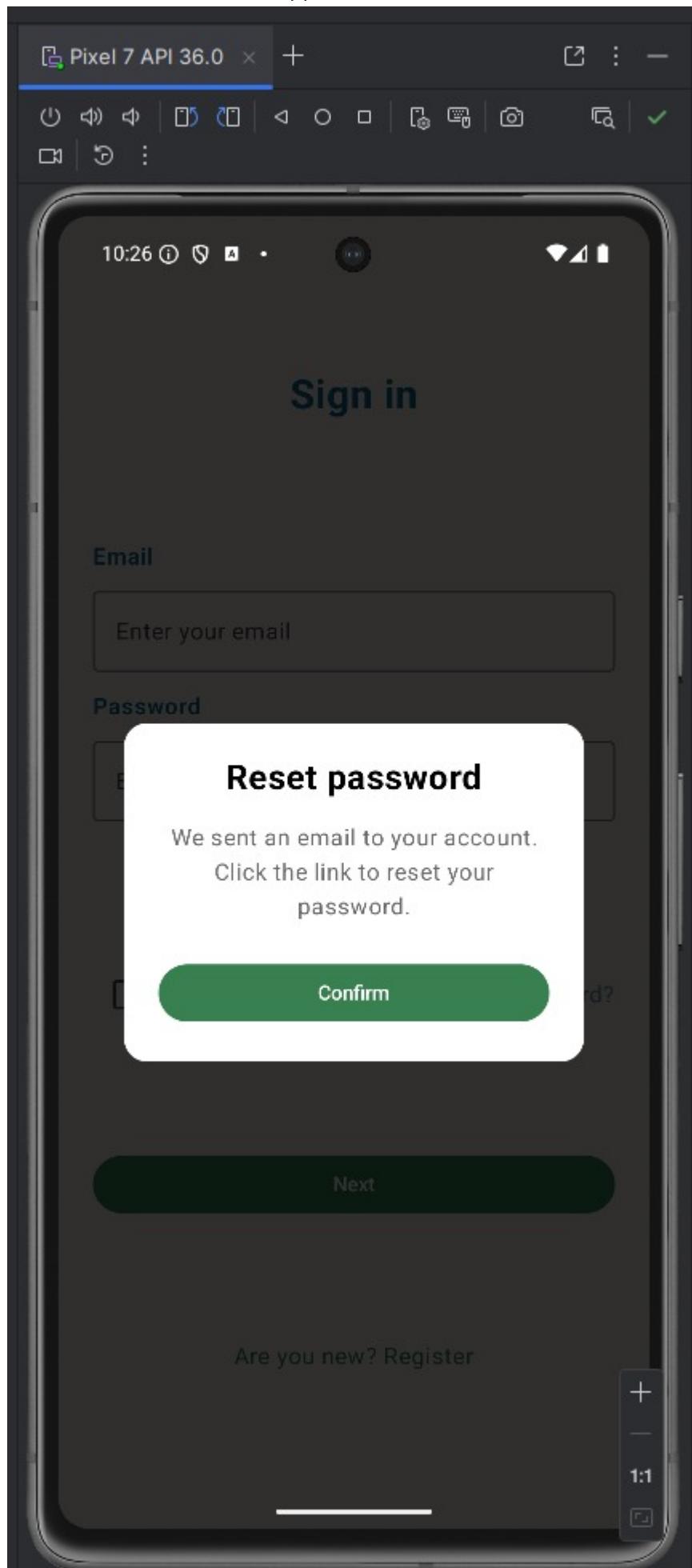


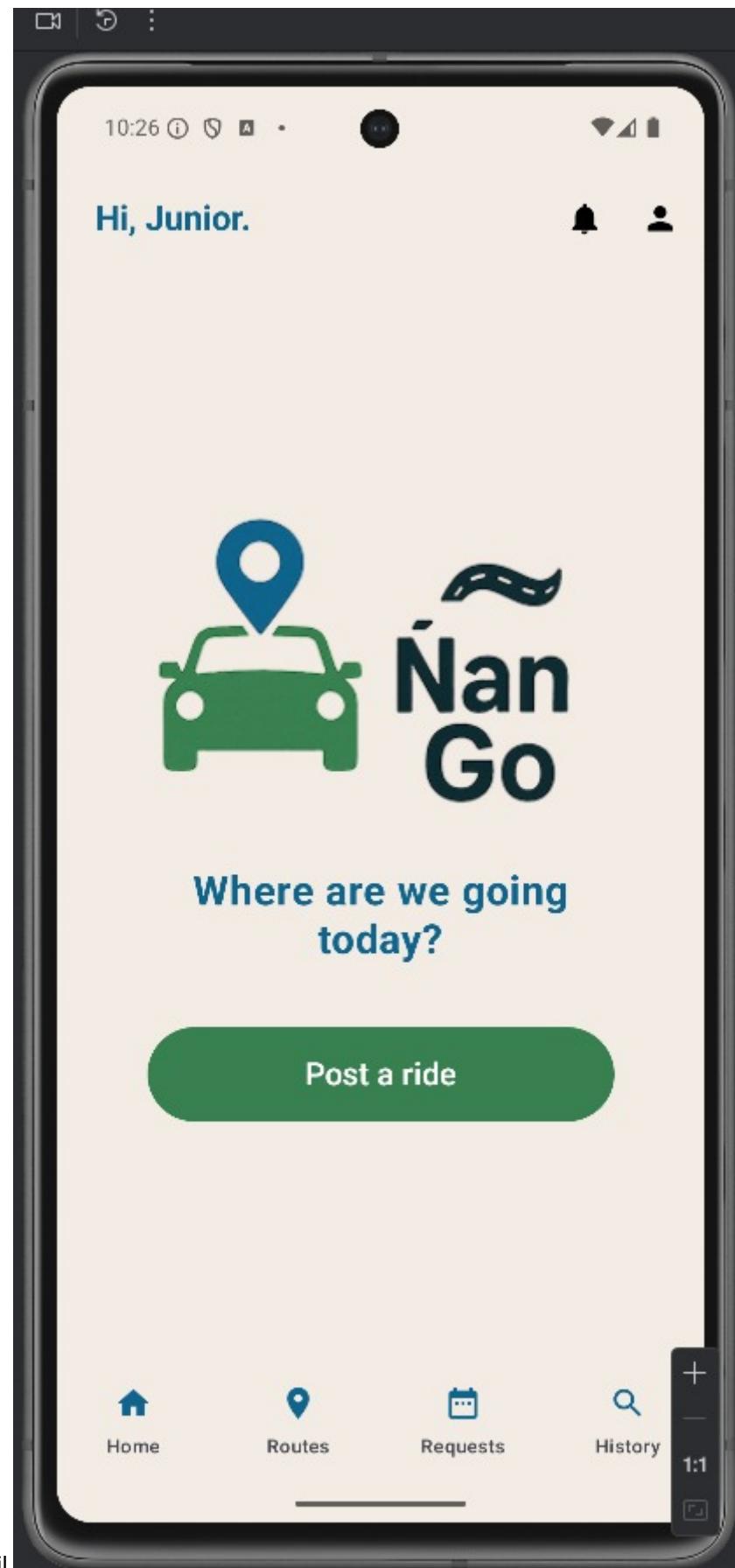
- Registro de usuario en la App Móvil



- Inicio de sesión en la App Móvil

- Resetear contraseña en la App Móvil

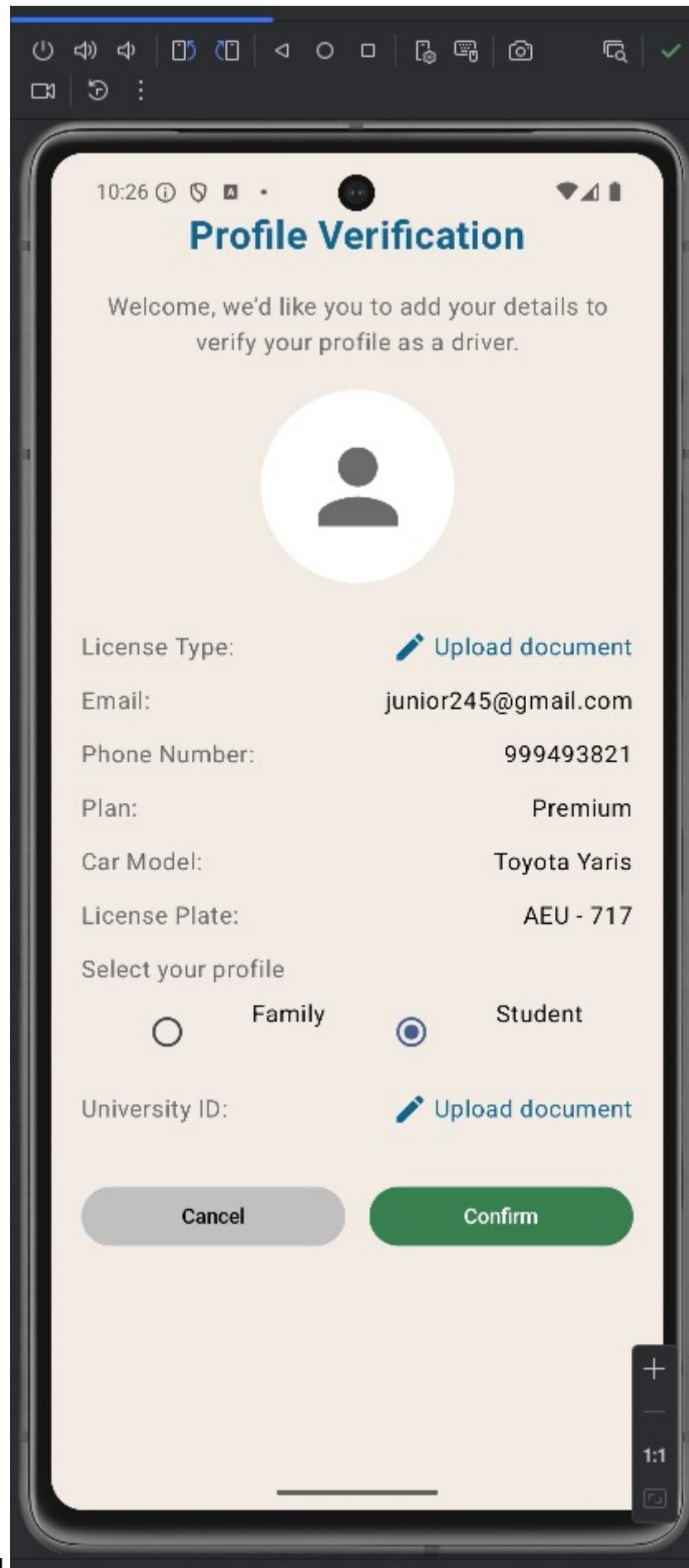




- Home de la App Móvil

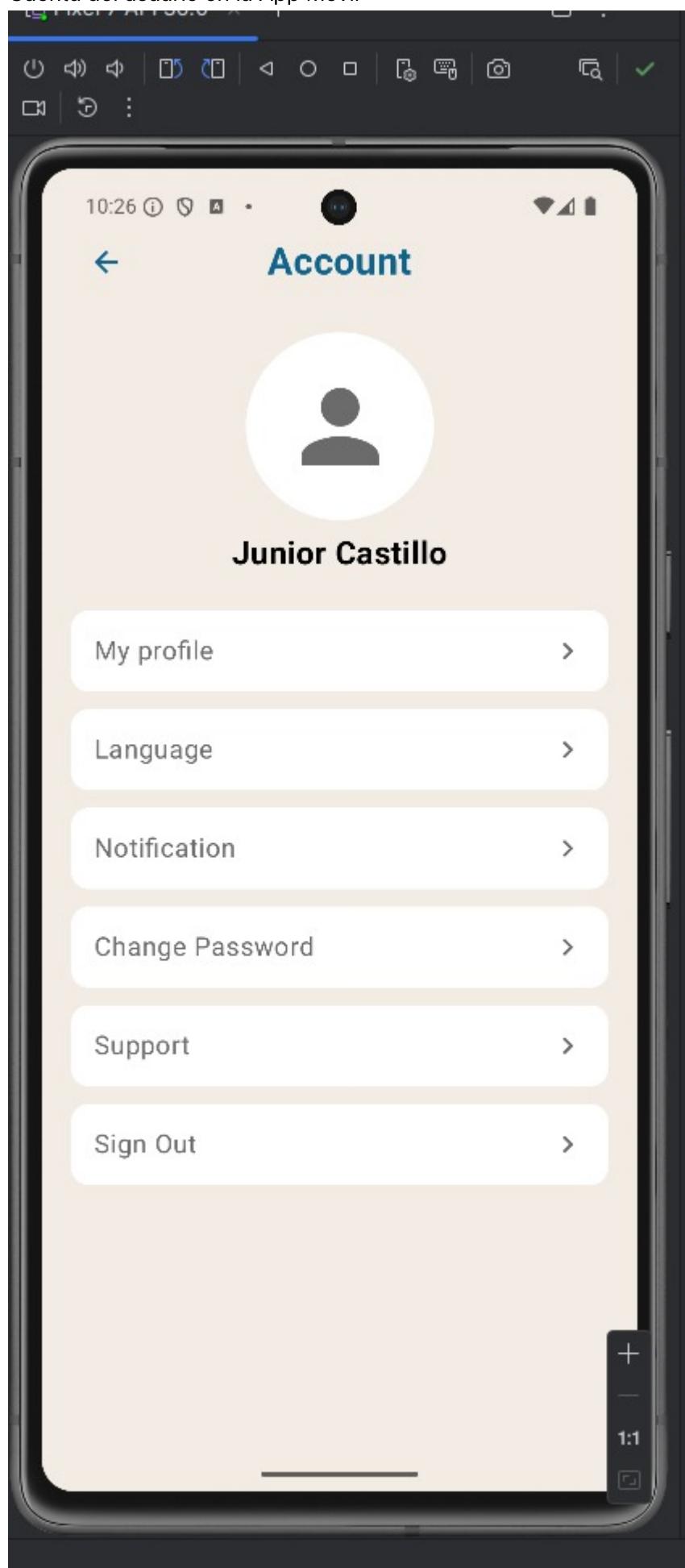
- Landing Page de Ñango

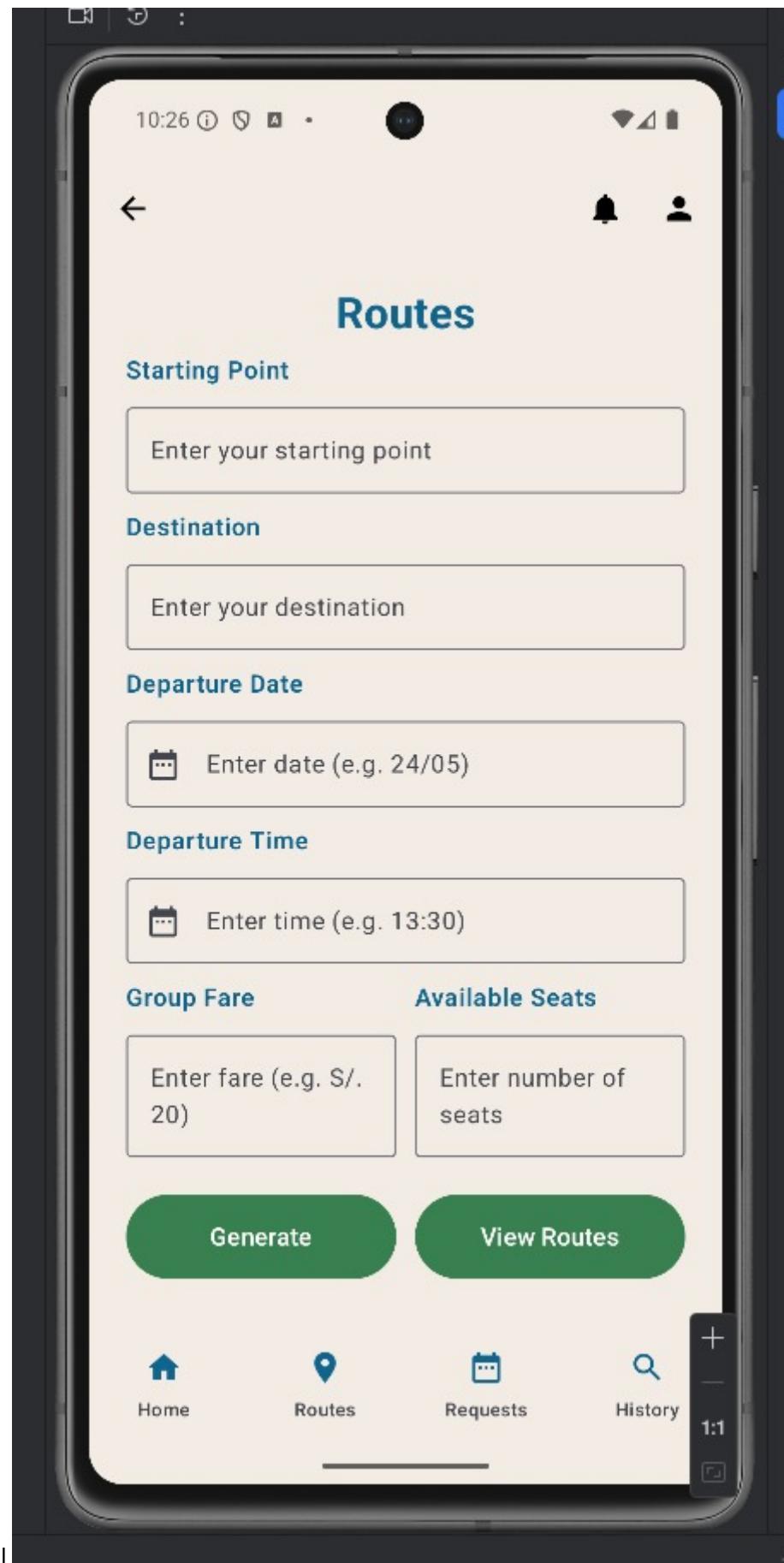
The screenshot shows the Ñango application's landing page configuration. At the top, there are tabs for General, Environment, Domains, Deployments, Preview Deployments, Schedules, Volume Backups, Logs, Monitoring, and Advanced. The General tab is selected. Below the tabs are buttons for Deploy, Reload, Stop, Rebuild, Open Terminal, Autodeploy (disabled), and Clean Cache (disabled). A 'Deploy Settings' section contains a 'Provider' dropdown with options: Github, Gitlab, Bitbucket, Git, Docker (selected), and Drop. Under Docker, the 'Docker Image' field contains 'ghcr.io/acc0238-2520-dam-14650-group3/landing-page:latest'. The 'Registry URL' field is set to 'https://ghcr.io'. The 'Username' field contains 'AndersonGamarraV' and the 'Password' field contains a masked password. A 'Save' button is located at the bottom right of this section. Below this is a 'Build Type' section with a 'Select the way of building your code' dropdown. A note states: 'Builders can consume significant memory and CPU resources (recommended: 4+ GB RAM and 2+ CPU cores). For production environments, please review our Production Guide for best practices and optimization recommendations. Builders are suitable for development and prototyping purposes when you have sufficient resources available.' The 'Build Type' dropdown is currently set to 'Builder'.



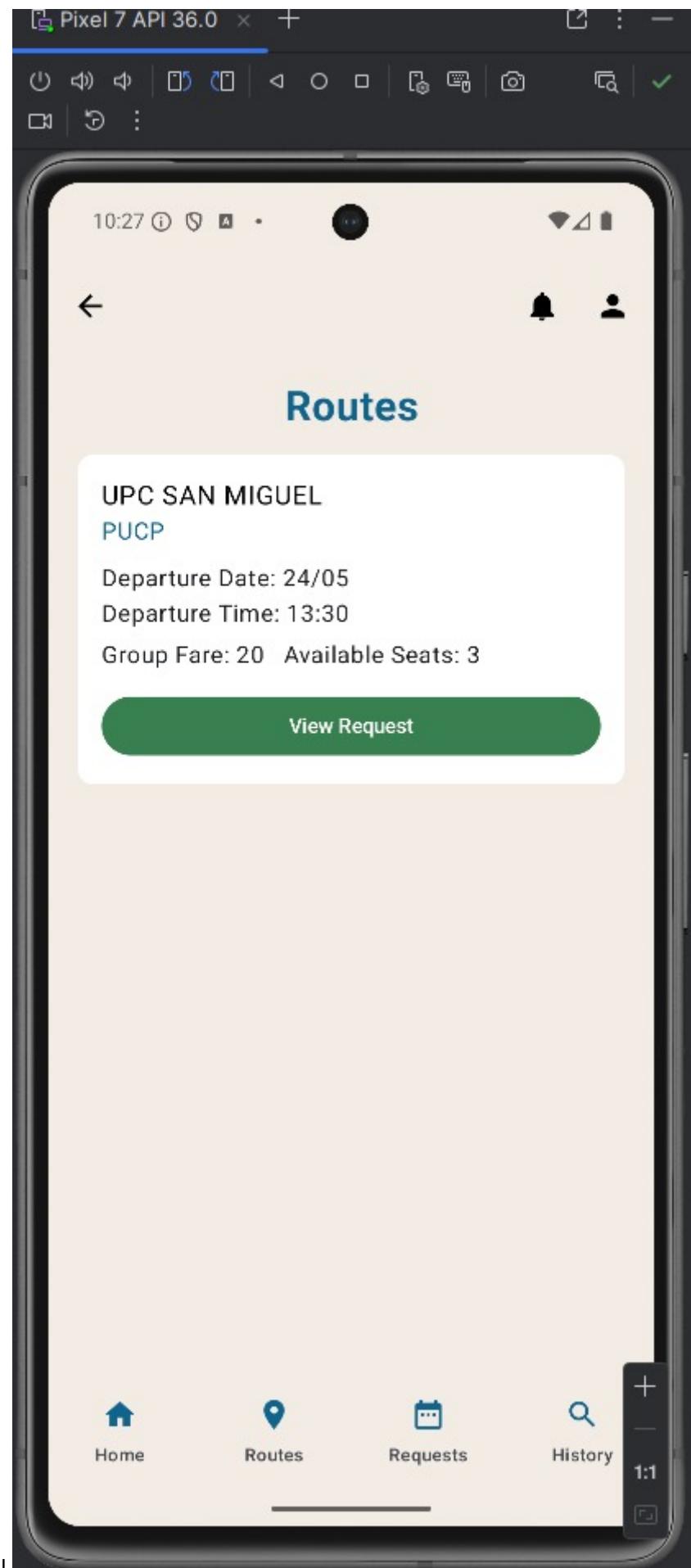
- Validación de perfil en la App Móvil

- Cuenta del usuario en la App Móvil





- Rutas en la App Movil



- Rutas detalladas en la App Móvil

#### 4.2.1.6. Services Documentation Evidence for Sprint Review

Aquí se documentan los servicios del backend del proyecto Ñango

## IAM Service

**Base URL:** `/api/v1` **Auth requerida:** Bearer Token (para endpoints protegidos como `/users/`)

| Método      | Endpoint                    | Descripción                               |
|-------------|-----------------------------|---|
| <b>POST</b> | <code>/auth/register</code> | Registrar nuevo usuario.                  |
| <b>POST</b> | <code>/auth/login</code>    | Iniciar sesión y obtener token JWT.       |
| <b>GET</b>  | <code>/users/</code>        | Listar usuarios (requiere autenticación). |

**Schemas visibles en Swagger:** `HTTPValidationError`, `LoginRequest`, `RegisterRequest`, `TokenResponse`, `UserOut`, `ValidationError`.

The screenshot shows the FastAPI OpenAPI UI interface. At the top, it displays "FastAPI 0.1.0 OAS 3.1" and a link to "/api/v1/openapi.json". Below this, there's a "Servers" dropdown set to "/api/v1" and an "Authorize" button with a lock icon. The main content area is organized into sections: "auth" (containing POST /auth/register and POST /auth/login), "users" (containing GET /users/), and "Schemas" (listing HTTPValidationError, LoginRequest, RegisterRequest, TokenResponse, UserOut, and ValidationError). Each schema entry includes a "Expand all" link.

## Booking Service

| Método        | Endpoint                                   | Descripción                                |
|---------------|--|--|
| <b>GET</b>    | <code>/api/v1/bookings</code>              | Listar todas las reservas registradas.     |
| <b>POST</b>   | <code>/api/v1/bookings</code>              | Crear una nueva reserva.                   |
| <b>DELETE</b> | <code>/api/v1/bookings/{booking_id}</code> | Eliminar una reserva por su identificador. |

**Booking Service** 0.1.0 OAS 3.1

[/openapi.json](#)

**default**

- GET** /api/v1/bookings List Bookings
- POST** /api/v1/bookings Create Booking
- DELETE** /api/v1/bookings/{booking\_id} Delete Booking

**Schemas**

- BookingRequest > Expand all object
- HTTPValidationError > Expand all object
- ValidationError > Expand all object

## Notification Service

Base URL: [/api/v1](#)

| Método       | Endpoint   | Descripción  |
|--------------|--|--|
| <b>POST</b>  | <a href="#">/api/v1/notifications</a>                        | Crear una nueva notificación.                        |
| <b>GET</b>   | <a href="#">/api/v1/notifications/users/{user_id}</a>        | Obtener las notificaciones de un usuario específico. |
| <b>PATCH</b> | <a href="#">/api/v1/notifications/{notification_id}/read</a> | Marcar una notificación como leída.                  |
| Método       | Endpoint   | Descripción  |
| <b>POST</b>  | <a href="#">/api/v1/preferences</a>                          | Crear preferencias de notificación.                  |
| <b>GET</b>   | <a href="#">/api/v1/preferences/users/{user_id}</a>          | Obtener preferencias de un usuario.                  |
| <b>PATCH</b> | <a href="#">/api/v1/preferences/{preference_id}</a>          | Actualizar preferencias de notificación.             |

The screenshot shows a detailed API documentation interface. At the top, there are sections for 'notifications', 'preferences', 'health', and 'default'. Each section contains a list of API endpoints with their methods, URLs, and descriptions. Below these sections is a 'Schemas' section containing definitions for various objects like CreateNotificationCommand, CreateNotificationPreferenceCommand, HTTPValidationError, NotificationDTO, and NotificationPreferenceDTO.

| notifications |  |
|---------------|--|
| POST          | /api/v1/notifications Create Notification                              |
| GET           | /api/v1/notifications/users/{user_id} Get User Notifications           |
| PATCH         | /api/v1/notifications/{notification_id}/read Mark Notification As Read |

| preferences |  |
|-------------|--|
| POST        | /api/v1/preferences Create Notification Preference                 |
| GET         | /api/v1/preferences/users/{user_id} Get User Preferences           |
| PATCH       | /api/v1/preferences/{preference_id} Update Notification Preference |

| health |                             |
|--------|-----------------------------|
| GET    | /api/v1/health Health Check |

| default |        |
|---------|--------|
| GET     | / Root |

| Schemas                             |                     |
|-------------------------------------|---------------------|
| CreateNotificationCommand           | > Expand all object |
| CreateNotificationPreferenceCommand | > Expand all object |
| HTTPValidationError                 | > Expand all object |
| NotificationDTO                     | > Expand all object |
| NotificationPreferenceDTO           | > Expand all object |

¿Deseas que integre este bloque actualizado dentro del documento completo con **IAM**, **Notification** y **Booking**, en formato Markdown uniforme para tu reporte final del sprint? (Puedo reemplazar la versión anterior del IAM por esta y entregarte todo el bloque completo final listo para copiar y pegar.)

#### 4.2.1.7. Software Deployment Evidence for Sprint Review

Durante este Sprint, se llevó a cabo el despliegue de la landing page de Ñango, intentamos crear esta página web con la mayor cantidad de información posible.

- Landing Page de Ñango

The landing page for the Ñango app features a top navigation bar with the brand logo, functional links (Funcionalidades, Cómo funciona, Planes, FAQ), and buttons for 'Ver demo' and 'Descargar'. A central headline reads 'Comparte el viaje, ahorra y llega seguro con Ñango.' Below it, a subtext encourages users to connect with university students to share routes. A large graphic shows two students using their phones to access the app's interface, which includes a map and a car icon. A call-to-action button labeled 'Empezar' is visible. At the bottom, a statistic '+2,500 estudiantes' is displayed. The page also lists several key features: Verificación de identidad, Notificaciones en tiempo real, and Pagos seguros. A section titled 'Funcionalidades clave' highlights six core features: Perfiles verificados, Rutas y horarios, Notificaciones, Pagos integrados, Idiomas, and Soporte.

Carpool universitario seguro

## Comparte el viaje, ahorra y llega seguro con Ñango.

Conecta con estudiantes de tu universidad para compartir rutas confiables, validar perfiles y pagar sin fricción.

Descargar app Cómo funciona

✓ Verificación de identidad ✓ Notificaciones en tiempo real ✓ Pagos seguros

+2,500 estudiantes

### Funcionalidades clave

Inspirado en el diseño de la app: simple, claro y pensado para estudiantes.

- Perfiles verificados**  
Validación con credencial universitaria y estado de aprobación visible.
- Rutas y horarios**  
Publica y reserva asientos con filtros por campus, horarios y costos.
- Notificaciones**  
Actualizaciones sobre solicitudes, chat y estado del viaje.
- Pagos integrados**  
Procesamiento seguro para dividir costos de manera transparente.
- Idiomas**  
Interfaz bilingüe (ES/EN) desde ajustes.
- Soporte**  
Centro de ayuda y reporte de incidentes en la app.

The screenshot shows the Ñango landing page. At the top left is the Ñango logo with a green icon. To its right are two buttons: "Ver demo" (in white) and "Descargar" (in white on a green background). Below these is a section titled "Carpool universitario seguro". The main headline reads: "Comparte el viaje, ahorra y llega seguro con Ñango." Below the headline is a description: "Conecta con estudiantes de tu universidad para compartir rutas confiables, validar perfiles y pagar sin fricción." At the bottom are two buttons: "Descargar app" (green) and "Cómo funciona" (white). Below the buttons are three icons with descriptions: a checkmark next to "Verificación de identidad", another checkmark next to "Notificaciones en tiempo real", and a third checkmark next to "Pagos seguros". A large graphic at the bottom shows two students, a boy and a girl, looking at their phones. One phone screen displays the Ñango app interface with a car icon, the text "Comparte el viaje, salva el día", and a green "Empezar" button. The other phone screen shows a map with a location pin.

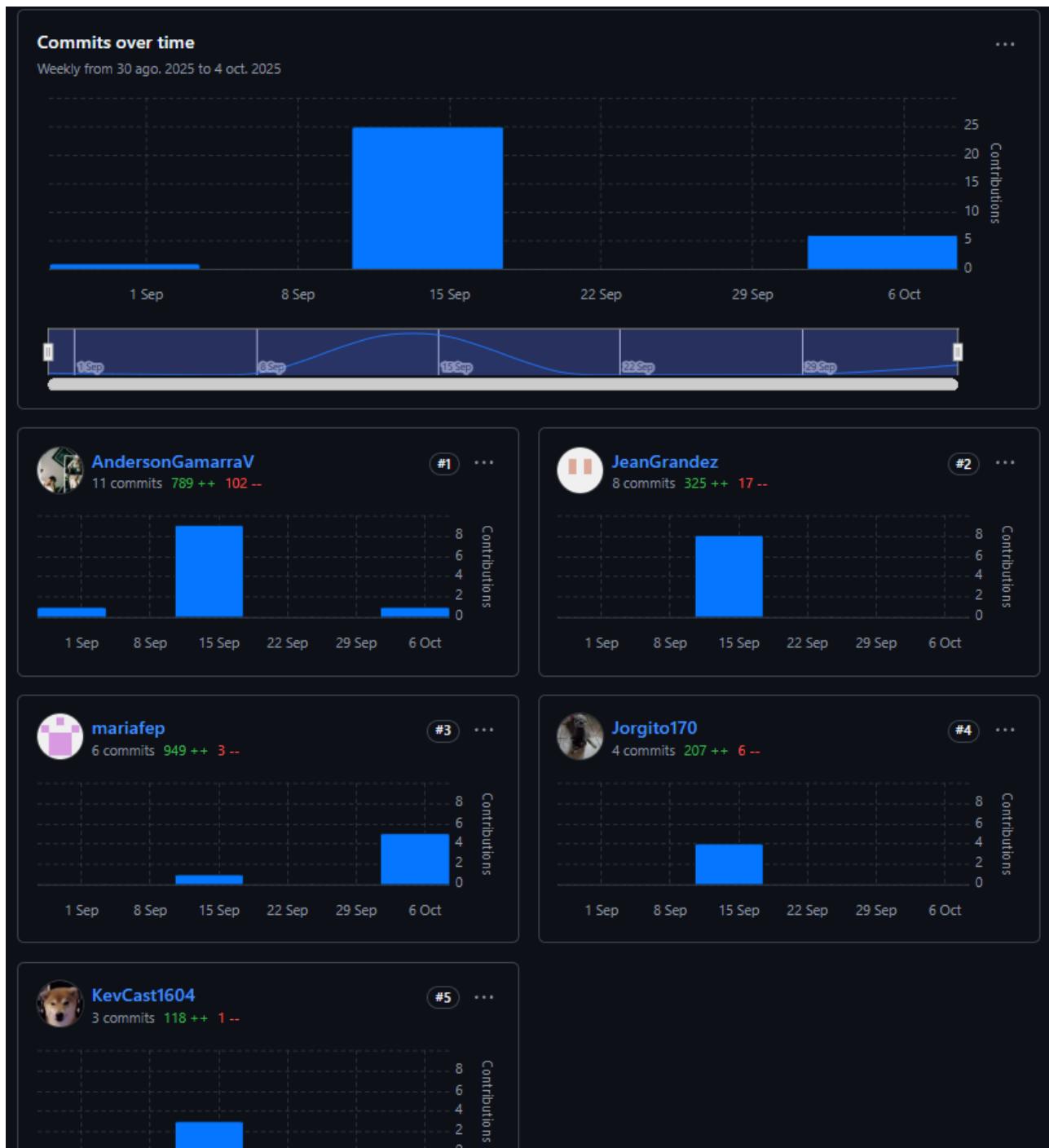
- Landing Page Responsive

#### 4.2.1.8. Team Collaboration Insights during Sprint

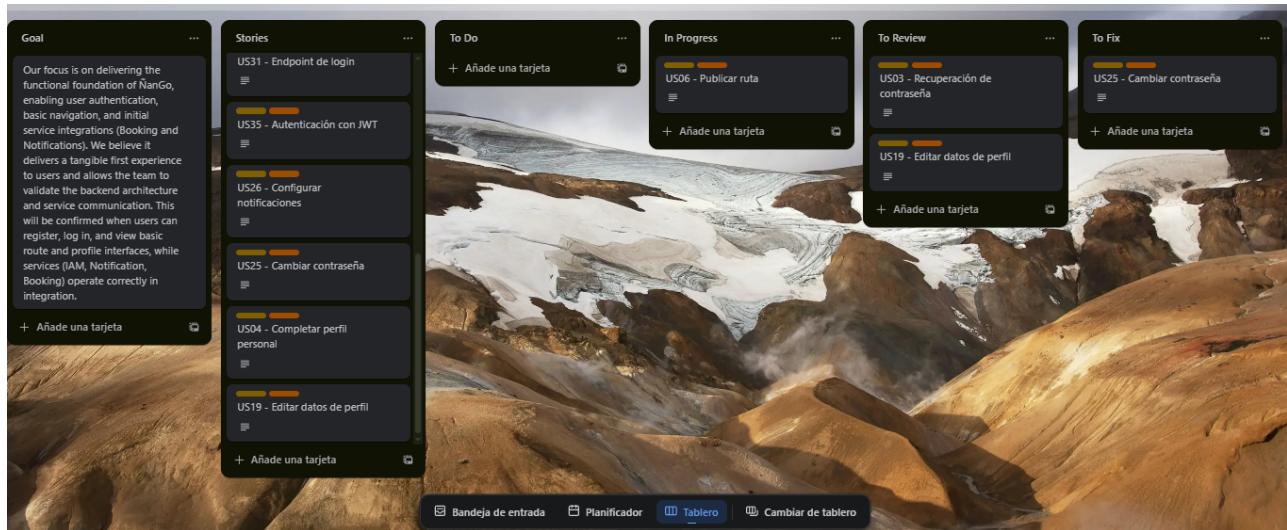
Aquí se documentan las reuniones de equipo y las decisiones tomadas durante el sprint.

La comunicación fue constante a través de plataformas usadas: WhatsApp; y utilizamos herramientas como Trello, GitHub Pull Requests para gestionar tareas y hacer seguimiento de los avances. La colaboración fue eficiente y fluida, pero se identificó la necesidad de mejorar la planificación temporal para optimizar la entrega de futuros sprints.

- Commits en GitHub:



- Trello:



## 4.3. Validation Interviews

### 4.3.1. Diseño de Entrevistas

## Segmento 1: Estudiantes con vehículo (Conductores)

**Objetivo:** Validar la claridad del proceso de registro, la facilidad de uso y la percepción de seguridad en la aplicación ÑanGo desde la perspectiva del conductor.

### Preguntas para el Segmento 1: Estudiantes con vehículo

1. ¿El proceso de registro y verificación del perfil como conductor donde se sube la licencia, se colocan los datos del vehículo y se elige el tipo de usuario te resulta claro y confiable?
2. ¿Consideras que la verificación de identidad y datos del vehículo aumenta tu confianza para usar la aplicación con otros estudiantes?
3. ¿La pantalla de inicio y el mensaje "Share the ride, save the day" te motivan a usar la app y entender rápidamente su propósito?
4. ¿Qué tan sencillo te resulta generar una ruta indicando punto de partida, destino, horario y precio grupal en la pantalla "Routes"?
5. ¿Te parece útil la función de "View Request" para revisar y aceptar pasajeros interesados en tus rutas?
6. ¿Crees que la posibilidad de compartir tus viajes con otros estudiantes te ayudaría a reducir costos de gasolina y estacionamiento?
7. ¿La distribución de iconos y botones en el menú inferior (Home, Routes, Requests, History) te resulta intuitiva y fácil de usar mientras estás en movimiento?
8. ¿Te sentirías cómodo compartiendo tu ruta y horario con otros usuarios verificados dentro de tu misma universidad?
9. ¿Qué tan importante consideras recibir notificaciones sobre nuevas solicitudes de pasajeros o recordatorios de tus viajes programados?
10. ¿Qué tipo de funciones adicionales te motivarían a usar ÑanGo con más frecuencia? (Por ejemplo, un sistema de calificaciones, chat interno o recompensas por viajes compartidos).

---

## Segmento 2: Estudiantes sin vehículo (Pasajeros)

**Objetivo:** Validar la percepción de confianza, accesibilidad y usabilidad de la aplicación ÑanGo desde la perspectiva de los estudiantes que buscan unirse a rutas.

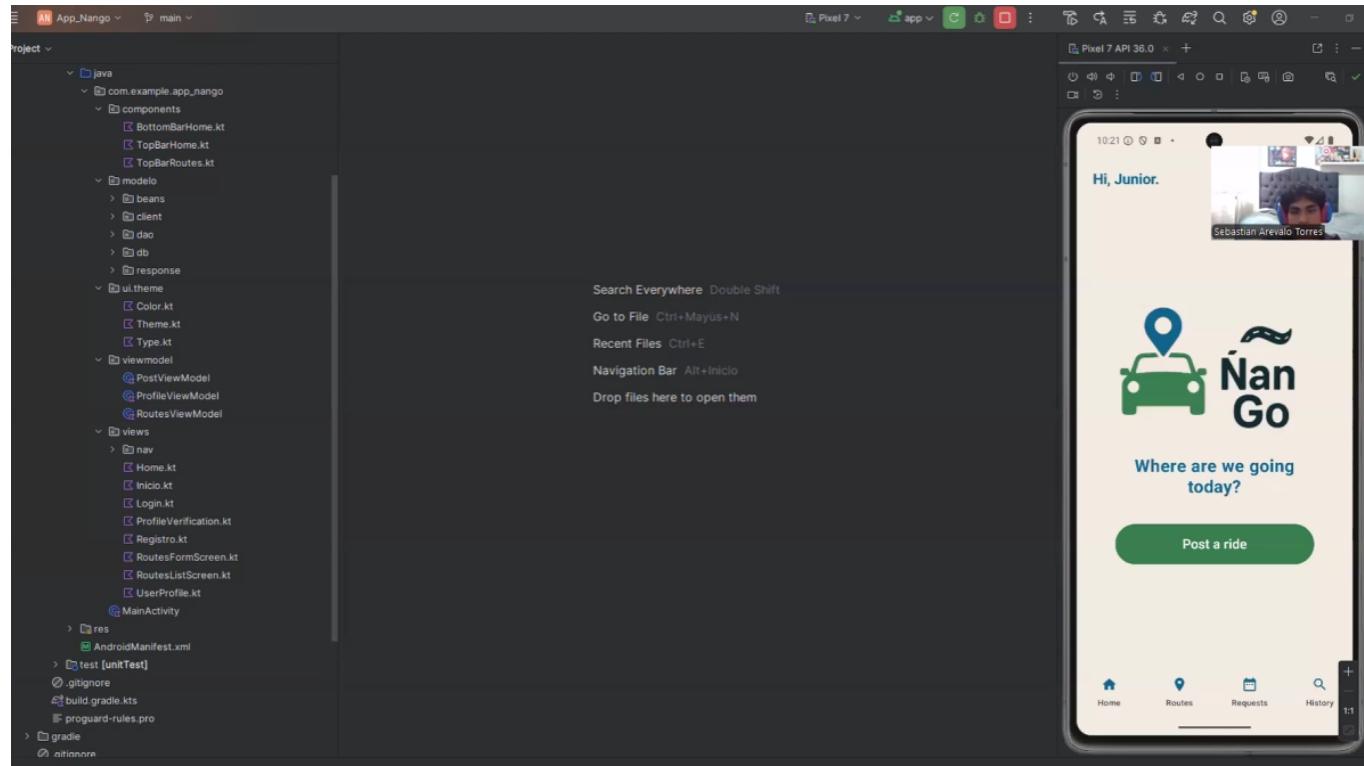
### Preguntas para el Segmento 2: Estudiantes sin vehículo

1. ¿El proceso de registro y verificación de identidad como pasajero te parece confiable y necesario para sentirte seguro al usar la aplicación?
2. ¿La pantalla de inicio y el mensaje "Share the ride, save the day" te ayudan a comprender rápidamente el propósito de la app?
3. ¿Qué tan sencillo te resulta buscar y unirte a rutas disponibles creadas por otros estudiantes?
4. ¿Consideras útil la opción de visualizar detalles del conductor (como carrera, calificación o número de viajes realizados) antes de unirte a una ruta?
5. ¿Te gustaría poder enviar solicitudes de viaje personalizadas indicando tu punto de recogida o tus horarios disponibles?
6. ¿Qué tan importante consideras recibir notificaciones sobre la aceptación de tus solicitudes o recordatorios de tus viajes próximos?
7. ¿La distribución de los iconos y menús (Home, Routes, Requests, History) te resulta intuitiva para encontrar y gestionar tus viajes?
8. ¿Qué tan cómodo te sentirías compartiendo tus datos básicos (como nombre, carrera o foto) con otros usuarios verificados dentro de la misma universidad?
9. ¿Qué elementos te harían confiar más en los conductores dentro de la aplicación (verificación, reseñas, historial de viajes, etc.)?
10. ¿Qué funcionalidades adicionales te motivarían a usar ÑanGo con más frecuencia? (Por ejemplo, recompensas, historial de viajes, chat interno, o sistema de calificaciones).

#### 4.3.2. Registro de Entrevistas

##### Segmento 1: Estudiantes con vehículo (Conductores)

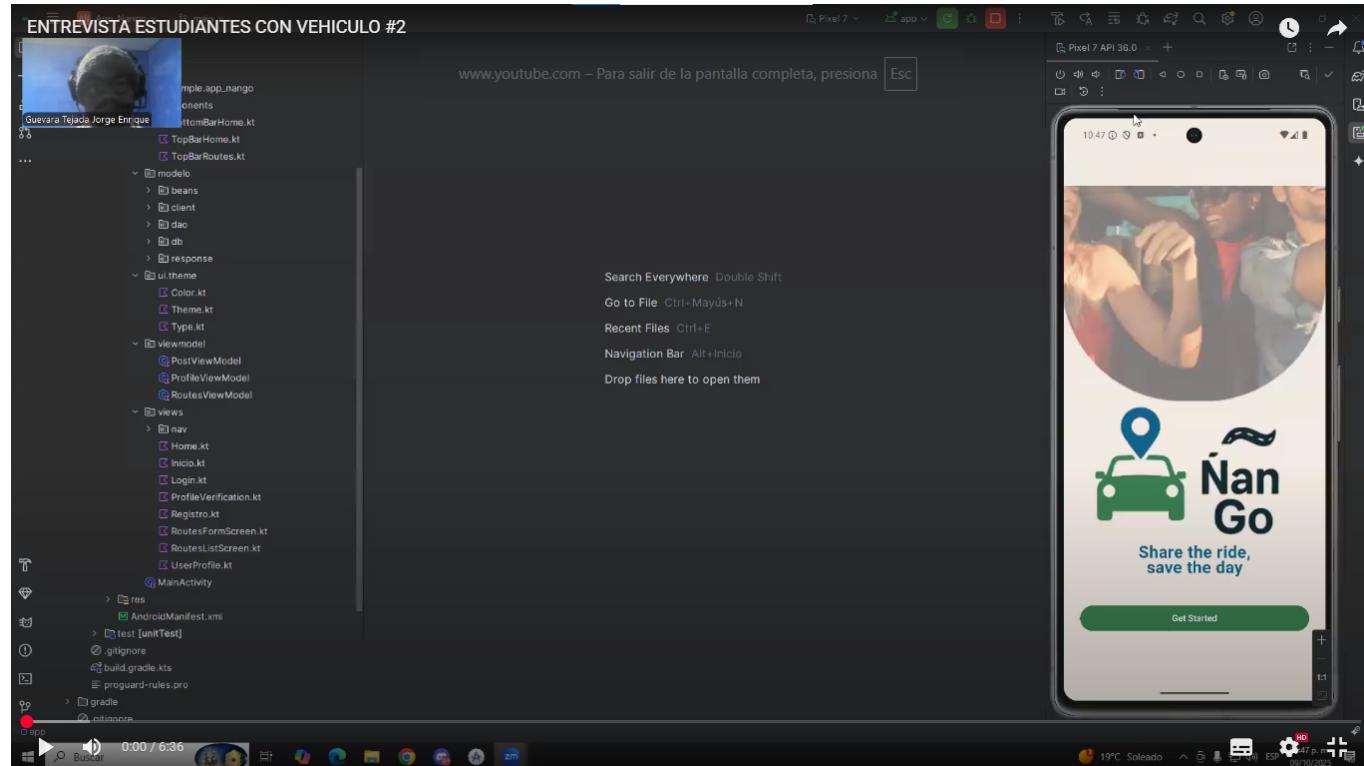
- Entrevista a Sebastian Arevalo: Edad: 23 años Universidad: Universidad Peruana de Ciencias Aplicadas (UPC) Carrera: Ingeniería de Software



## Entrevista a Sebastian Arevalo

- Entrevista a Jose Sosa Edad: 20 Universidad: Universidad Peruana de Ciencias Aplicadas (UPC) Carrera: Ingeniería de Sistemas

### 4.3.3. Evaluaciones según heurísticas



## Entrevista a Jose Sosa