

UNIVERSIDAD DEL VALLE DE GUATEMALA

Colegio Universitario



Documento integrador

Grupo No.7

Catedrática: Lynette García Pérez

Adrián Ricardo González Muralles - 23152
Jose Pablo Ordoñez Barrios – 231329
Manuel Armando Ulin Pérez - 221017
Marcos Rodrigo Ambrocio Larios - 231140
José Alejandro Antón Escobar - 221041
Rene Sebastián Espinal Zamora - 228676

Ingeniería de Software 1
Sección 10
Guatemala, 2025

Resumen

El sistema de movilidad en la Universidad del Valle de Guatemala (UVGride) enfrenta múltiples desafíos que afectan a estudiantes, docentes y colaboradores, como los altos costos y tiempos prolongados de transporte, la inseguridad en medios externos y la falta de estacionamiento adecuado. La falta de opciones accesibles y seguras para trasladarse al campus genera dificultades diarias para la comunidad universitaria, especialmente para quienes viven en zonas alejadas. La dependencia del transporte público y del uso individual de vehículos no solo incrementa los costos y el tiempo de traslado, sino que también contribuye a la congestión vial y a una mayor carga ambiental. Esta problemática impacta directamente en la calidad de vida de los miembros de la universidad, dificultando su acceso oportuno y seguro a sus actividades académicas y laborales.

Introducción

Sistema de Movilidad Universitaria para la Universidad del Valle de Guatemala (UVGride)

Descripción de la entidad

La Universidad del Valle de Guatemala es una institución de educación superior reconocida por su enfoque en la innovación, la investigación y la excelencia académica. Su comunidad está conformada por estudiantes, docentes y colaboradores que diariamente se trasladan al campus para realizar actividades académicas y administrativas. Sin embargo, la falta de un sistema de transporte universitario eficiente representa un desafío significativo, especialmente para aquellos que residen en zonas alejadas dentro del área metropolitana e interdepartamental. Este proyecto se enmarca en la búsqueda de soluciones tecnológicas que mejoren la movilidad dentro de la comunidad universitaria, optimizando tiempos de traslado y ofreciendo un medio de transporte accesible y seguro.

Descripción de la idea

La movilidad dentro de la comunidad universitaria presenta diversas dificultades que impactan negativamente en el acceso a la educación y el bienestar de los estudiantes, docentes y colaboradores. La falta de opciones de transporte adecuadas genera altos costos de traslado, tiempos prolongados de viaje y preocupaciones sobre la seguridad. Muchos estudiantes que viven lejos del campus deben caminar largas distancias, pagar múltiples pasajes o depender de terceros para llegar a sus clases. Además, la ausencia de horarios fijos en los medios de transporte públicos complica la planificación del tiempo de viaje y, en algunos casos, obliga a los estudiantes a mudarse cerca de la universidad, lo que representa un gasto adicional. Quienes poseen vehículo propio también enfrentan desafíos como el alto costo del combustible y la falta de estacionamiento, lo que agrava aún más la problemática de movilidad dentro de la UVG.

Objetivos del proyecto

Objetivo general:
Identificar y analizar las dificultades de movilidad que enfrentan los miembros de la comunidad UVG con el fin de comprender su impacto y proponer posibles estrategias de mejora.

Objetivos específicos:

- Evaluar las necesidades y dificultades de transporte de los estudiantes, docentes y colaboradores de la UVG.
- Identificar los factores que influyen en los costos, tiempos y seguridad en los traslados hacia el campus.

- Analizar las limitaciones del transporte público y privado en el acceso a la universidad.
- Examinar el impacto de la falta de movilidad en la experiencia académica y laboral dentro de la comunidad UVG.
- Explorar posibles enfoques para mejorar la accesibilidad y reducir los inconvenientes relacionados con el transporte.

Este proyecto busca generar un diagnóstico detallado de la situación actual de movilidad en la UVG, permitiendo comprender las dificultades que enfrentan sus miembros y plantear estrategias que contribuyan a mejorar sus condiciones de traslado

Descubrimientos:

- El tráfico es uno de los obstáculos principales, la mayoría de los encuestados tienen problemas con el tráfico.
- Los gastos como la gasolina o el pagar por un transporte como “Uber” es un problema ya que pueden superar los límites de los usuarios.
- Al no contar con un vehículo propio se deben acudir a terceros para poder transportarse.

Necesidades:

- Encontrar alternativas seguras a los transportes públicos por un precio no tan elevado.
- Reducir el impacto que les genera el tráfico en el trayecto diaria hacia la universidad.

Oportunidades:

- Viajes seguros y con personas que sabemos que serán de la universidad.
- Puntualidad y comodidad.
- Apoyo entre colaboradores de la universidad y compañerismo al apoyarse unos a otros.
- Apoyo al estudiante en el transporte todos los días.

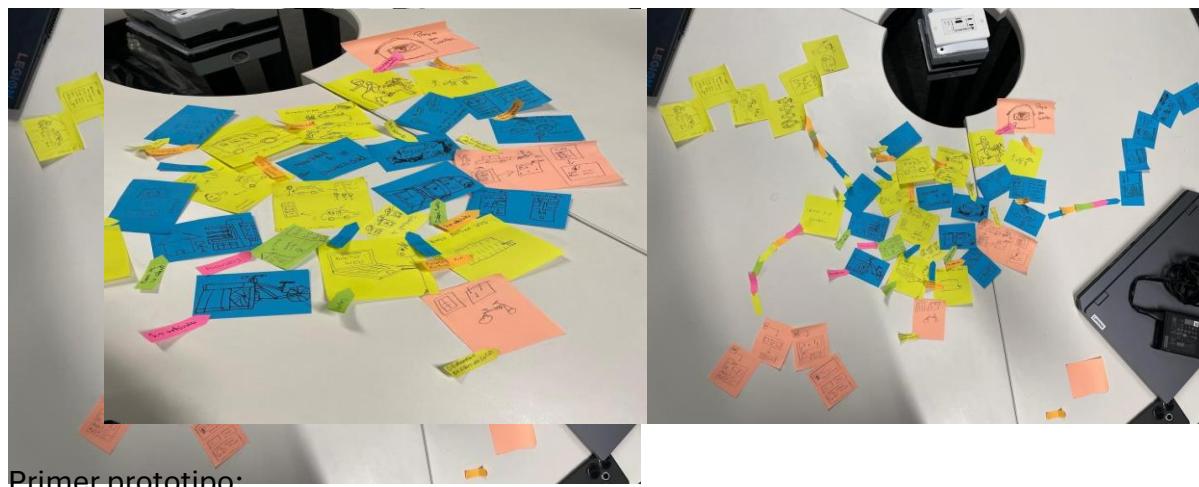
Definición del proyecto.

La comunidad universitaria de la Universidad del Valle de Guatemala, compuesta por estudiantes, docentes y colaboradores, enfrenta importantes dificultades de movilidad, especialmente aquellos que residen en áreas distantes al campus. La falta de un sistema de transporte universitario eficiente genera problemas como altos costos de transporte público, inseguridad, la ausencia de horarios fijos en los buses y escasez de estacionamiento para quienes tienen vehículo propio. Además, la dependencia de terceros para los traslados y la necesidad de mudarse cerca de la universidad debido a la falta de opciones confiables generan estrés y pérdida de tiempo. Este proyecto tiene como objetivo abordar estas dificultades y mejorar la accesibilidad, seguridad y eficiencia en los traslados hacia y desde la UVG, a través de una solución innovadora que beneficie a toda la comunidad universitaria.

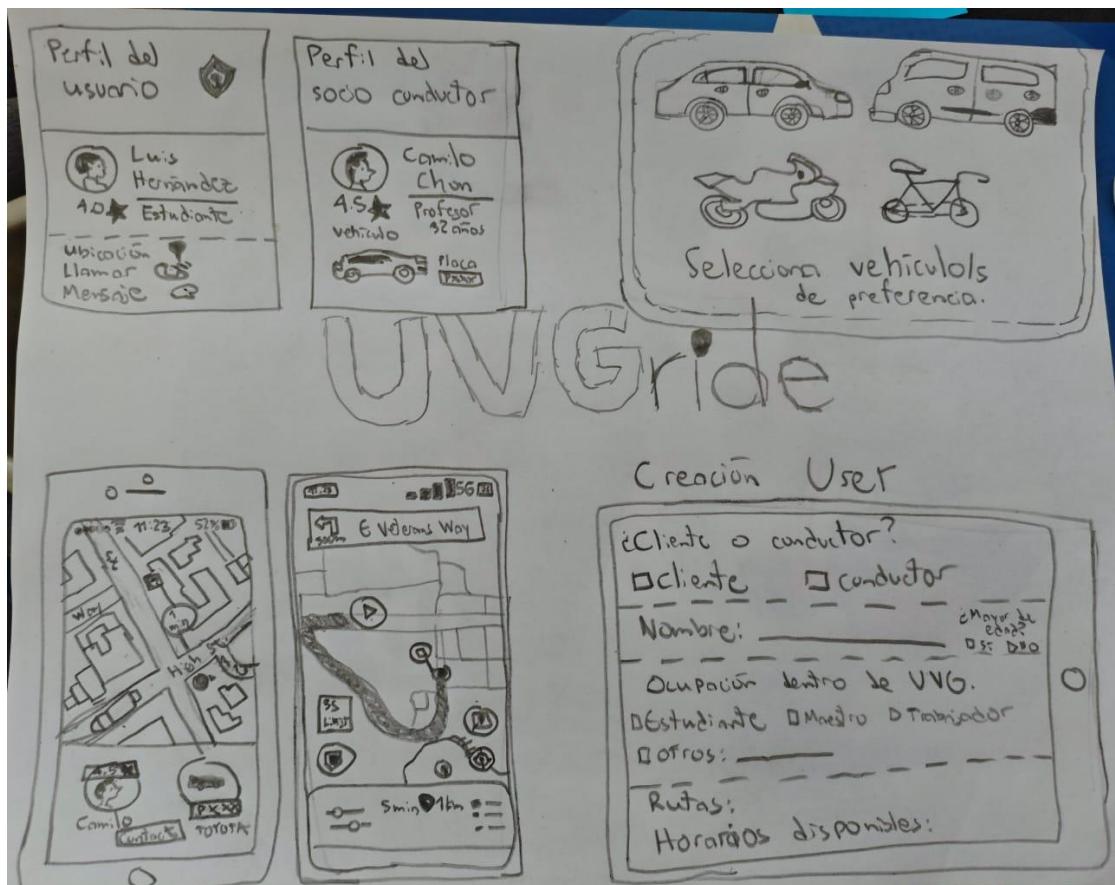
Design Thinking:

- Prototipos:

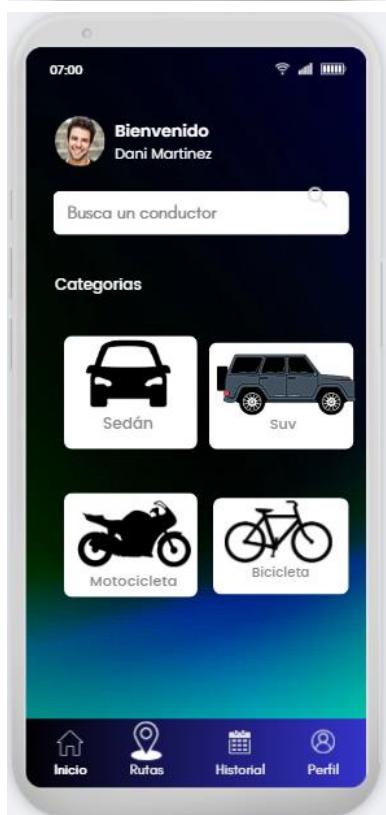
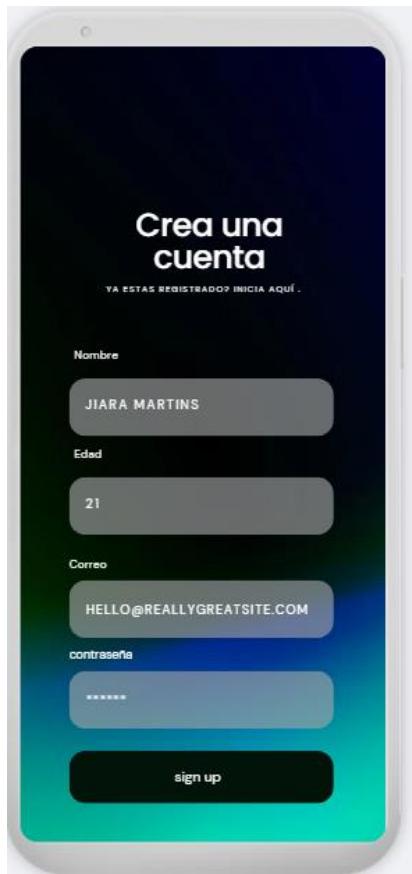
- Primera fase:

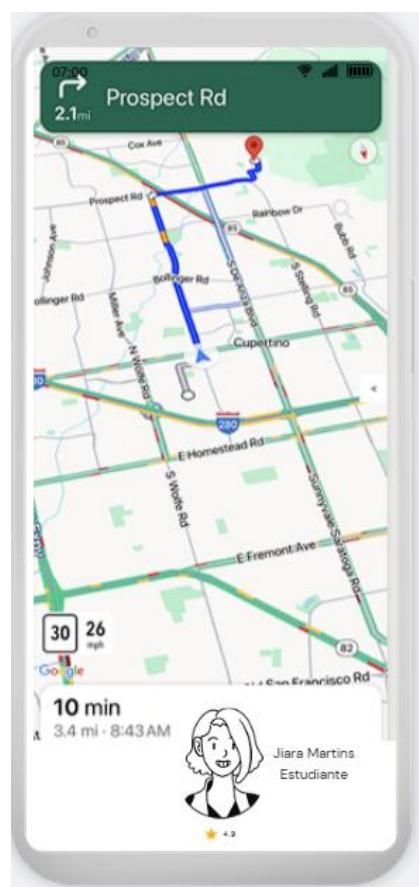
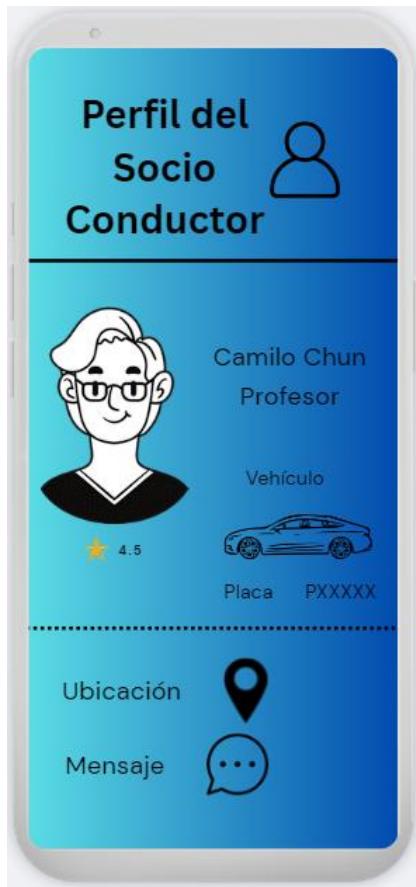


Primer prototipo:



Prototipo mejorado:





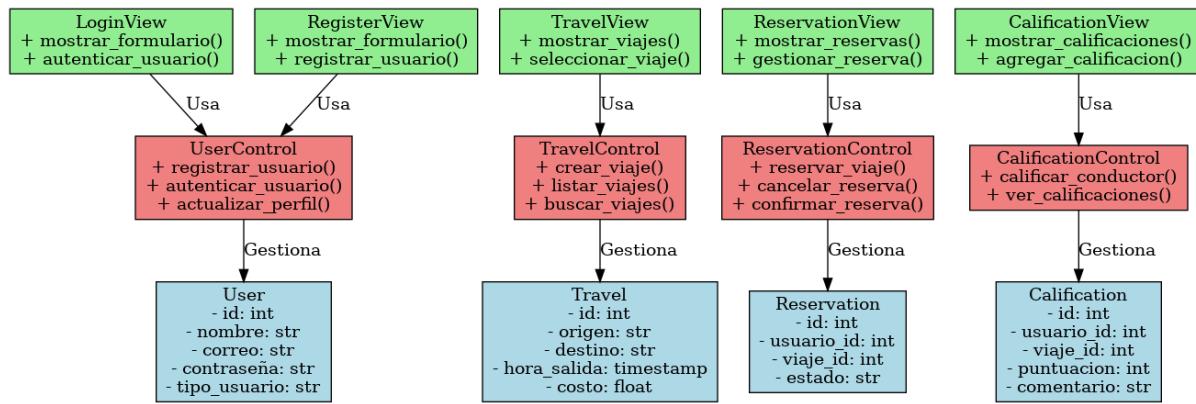
Análisis:

Lista de requisitos funcionales:

Requisito funcional	Historia de usuario
El sistema debe permitir a los usuarios registrarse con su correo institucional.	“Como estudiante sin vehículo, quiero una alternativa accesible, segura y confiable para poder llegar a la universidad sin depender de un solo medio de transporte y sin gastar demasiado.”
El sistema debe permitir a los usuarios iniciar sesión con credenciales seguras.	“Como profesor, quiero acceder a un medio de transporte seguro y cómodo que me permita transportar pertenencias personales o familiares menores, para poder llegar puntualmente a mis clases.”
El sistema debe mostrar opciones de transporte disponibles según la ubicación del usuario.	“Como estudiante que vive lejos de la universidad, quiero encontrar un medio de transporte eficiente y de bajo costo para asistir puntualmente a mis clases sin incurrir en gastos elevados.”
El sistema debe permitir filtrar opciones de transporte por precio, ubicación y tipo de viaje.	“Como usuario, quiero filtrar opciones de transporte según mi presupuesto para seleccionar la alternativa más accesible.”
El sistema debe permitir a los conductores ofrecer viajes con información de horario, precio y ruta.	“Como usuario, quiero crear un viaje compartido para que otros puedan unirse y reducir costos.”
El sistema debe enviar confirmaciones y detalles del viaje a los pasajeros y conductores.	“Como usuario, quiero recibir confirmación y detalles de la reserva.”
El sistema debe permitir a los pasajeros reservar un viaje con anticipación.	“Como usuario, quiero reservar transporte con anticipación para asegurar disponibilidad.”
El sistema debe incluir un sistema de calificación y comentarios sobre conductores y pasajeros.	“Como profesor, quiero una aplicación que me permita encontrar un conductor confiable con espacio suficiente para transportar mis pertenencias, garantizando un viaje seguro y cómodo hacia la universidad.”
El sistema debe ser accesible para usuarios con discapacidad (compatibilidad con lectores de pantalla y opciones de accesibilidad).	“Como persona con discapacidad, quiero encontrar un transporte adaptado a mis necesidades para movilizarme de manera cómoda y sin retrasos.”
El sistema debe permitir gestionar los participantes de un viaje compartido.	“Como usuario, quiero unirme a un viaje compartido existente.”

Clases preliminares:

(Revisión) Diagrama de clases MVC



Descripción de las Clases:

Modelo (Entidades)

1. User

- Descripción: Representa a los usuarios del sistema (estudiantes, docentes y conductores).
- Atributos:
 - id: int → Identificador único del usuario.
 - nombre: str → Nombre completo.
 - correo: str → Correo institucional.
 - contraseña: str → Contraseña cifrada.
 - tipo_usuario: str → Indica si es pasajero, conductor o ambos.

2. Travel

- Descripción: Representa un viaje disponible en el sistema.
- Atributos:
 - id: int → Identificador del viaje.
 - origen: str → Punto de partida.
 - destino: str → Destino final.
 - hora_salida: timestamp → Hora de inicio del viaje.
 - costo: float → Precio del viaje.

3. Reservation

- Descripción: Representa una reserva de viaje realizada por un usuario.
- Atributos:
 - id: int → Identificador de la reserva.
 - usuario_id: int → ID del usuario que reserva.
 - viaje_id: int → ID del viaje reservado.
 - estado: str → Estado de la reserva (pendiente, confirmada, cancelada).

4. Calification

- Descripción: Permite que los usuarios califiquen y comenten sobre su experiencia en un viaje.
- Atributos:
 - id: int → Identificador de la calificación.
 - usuario_id: int → Usuario que califica.
 - viaje_id: int → Viaje asociado.
 - puntuacion: int → Calificación numérica (1-5).
 - comentario: str → Opinión del usuario.

Controlador (Lógica de Negocio)

5. UserControl

- Descripción: Maneja la autenticación, registro y gestión de usuarios.
- Métodos:
 - registrar_usuario() → Crea un nuevo usuario.
 - autenticar_usuario() → Valida credenciales.
 - actualizar_perfil() → Modifica información del usuario.

6. TravelControl

- Descripción: Gestiona los viajes en la aplicación.
- Métodos:

- `crear_viaje()` → Permite a un conductor registrar un viaje.
- `listar_viajes()` → Devuelve la lista de viajes disponibles.
- `buscar_viajes()` → Filtra viajes por destino, precio o fecha.

7. ReservationControl

- Descripción: Maneja la gestión de reservas.
- Métodos:
 - `reservar_viaje()` → Permite a un usuario reservar un viaje.
 - `cancelar_reserva()` → Cancela una reserva antes del viaje.
 - `confirmar_reserva()` → Confirma la asistencia del pasajero.

8. CalificationControl

- Descripción: Gestiona las calificaciones y comentarios de los usuarios.
- Métodos:
 - `calificar_conductor()` → Permite calificar a un conductor.
 - `ver_calificaciones()` → Obtiene las calificaciones de un usuario.

Vista (Interfaz de Usuario)

9. LoginView

- Descripción: Muestra la pantalla de inicio de sesión.
- Métodos:
 - `mostrar_formulario()` → Despliega el formulario de login.
 - `autenticar_usuario()` → Valida los datos y redirige según el resultado.

10. RegisterView

- Descripción: Permite registrar un nuevo usuario.
- Métodos:
 - `mostrar_formulario()` → Despliega el formulario de registro.

- registrar_usuario() → Envía los datos al controlador.

11. TravelView

- Descripción: Muestra los viajes disponibles para el usuario.
- Métodos:
 - mostrar_viajes() → Lista los viajes disponibles.
 - seleccionar_viaje() → Permite elegir un viaje específico.

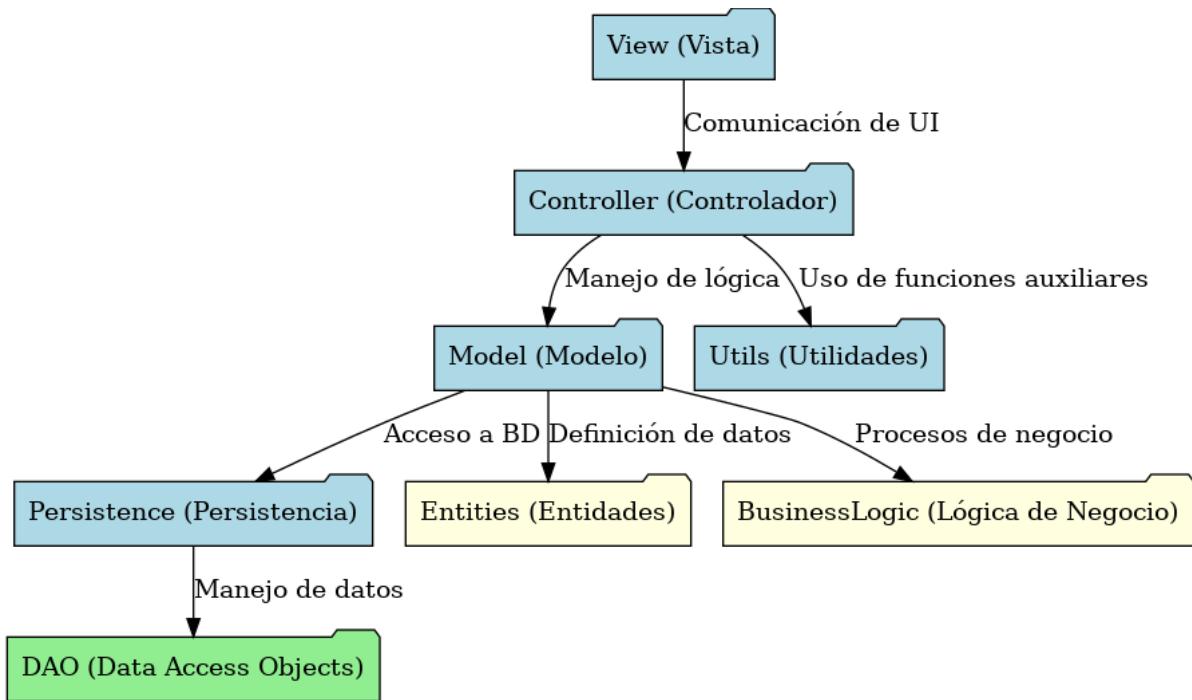
12. ReservationView

- Descripción: Administra la información de las reservas.
- Métodos:
 - mostrar_reservas() → Muestra las reservas activas.
 - gestionar_reserva() → Permite cancelar o modificar una reserva.

13. CalificationView

- Descripción: Muestra las calificaciones de los conductores y pasajeros.
- Métodos:
 - mostrar_calificaciones() → Lista calificaciones y comentarios.
 - agregar_calificacion() → Permite a un usuario dejar su evaluación.

Diagrama de paquetes:



Descripción de cada paquete y de sus componentes:

1. View

Este paquete maneja la interfaz de usuario y la interacción con el sistema. Se encarga de mostrar los datos procesados por el controlador y capturar la información ingresada por los usuarios.

Componentes:

- LoginView: Muestra la pantalla de inicio de sesión y envía los datos al controlador.
- RegisterView: Permite el registro de nuevos usuarios.
- TravelView: Muestra la lista de viajes disponibles.
- ReservationView: Permite gestionar reservas activas.
- CalificationView: Muestra y registra calificaciones de conductores y pasajeros.
- PaymentView: Muestra opciones de pago y estado de transacciones.
- NotificationView: Muestra notificaciones de reservas, pagos y cambios en el sistema.

Relación: Se comunica con el paquete Controller para enviar y recibir datos.

2. Controller

Este paquete gestiona la lógica de negocio. Recibe información de la Vista, la procesa y la envía al Modelo. También obtiene datos desde el Modelo y los devuelve a la Vista en un formato adecuado.

Componentes:

- UserControl: Gestiona autenticación, registro y actualización de usuarios.
- TravelControl: Maneja la creación y búsqueda de viajes.
- ReservationControl: Administra reservas y confirmaciones.
- CalificationControl: Permite registrar y consultar calificaciones.
- PaymentControl: Gestiona métodos de pago y transacciones.
- NotificationControl: Maneja la lógica de notificaciones para usuarios.

Relación: Se comunica con la Vista para recibir y mostrar datos, y con el Modelo para aplicar reglas de negocio.

3. Model

Este paquete define la estructura de los datos del sistema y la lógica de negocio sin depender de la interfaz gráfica.

Subpaquetes:

- Entities (Entidades): Clases que representan las tablas de la base de datos.
- User: Representa a los usuarios (estudiantes, docentes, conductores).
- Travel: Representa un viaje con su información (origen, destino, horario).
- Reservation: Representa una reserva asociada a un viaje y un usuario.
- Calification: Registra puntuaciones y comentarios de los usuarios.
- Payment: Gestiona métodos de pago y transacciones.
- Notification: Representa notificaciones enviadas a los usuarios.
- BusinessLogic (Lógica de Negocio): Métodos que procesan los datos antes de enviarlos a Persistencia.
- Pricing: Calcula tarifas y costos de viajes.
- MatchingService: Asigna pasajeros a viajes disponibles.
- Security: Maneja validaciones y cifrado de contraseñas.

Relación: Se comunica con Persistencia para acceder a la base de datos y con Controller para proporcionar datos procesados.

4. Persistence

Este paquete maneja la conexión con la base de datos y realiza operaciones CRUD (Crear, Leer, Actualizar, Eliminar).

Subpaquete:

- DAO (Data Access Objects): Clases encargadas de la comunicación con la base de datos.
- UserDAO: Gestiona consultas sobre los usuarios.
- TravelDAO: Maneja los viajes almacenados en la base de datos.
- ReservationDAO: Controla las reservas registradas.
- CalificationDAO: Administra las calificaciones y comentarios.
- PaymentDAO: Gestiona pagos y transacciones.
- NotificationDAO: Administra el almacenamiento y envío de notificaciones.

Relación: Se comunica con el Modelo para almacenar y recuperar datos de la base de datos.

5. Utils

Este paquete contiene herramientas auxiliares utilizadas en todo el sistema.

Componentes:

- EmailService: Maneja el envío de correos electrónicos (confirmaciones, alertas).
- Logger: Registra eventos, errores y auditorías del sistema.
- Security: Contiene métodos para cifrado de contraseñas y validaciones de seguridad.

Relación: Es utilizado por Controller y Persistencia para funcionalidades adicionales.

Diagrama de clases persistentes:

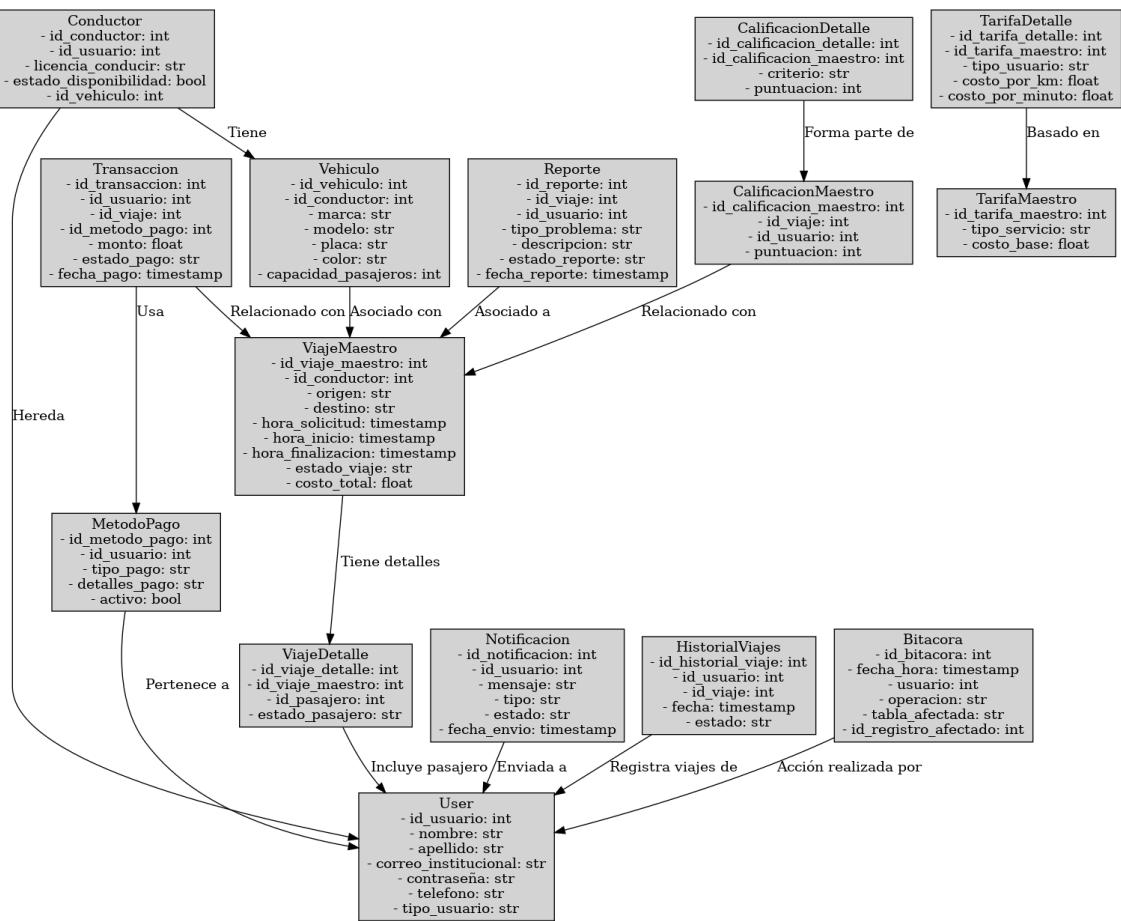
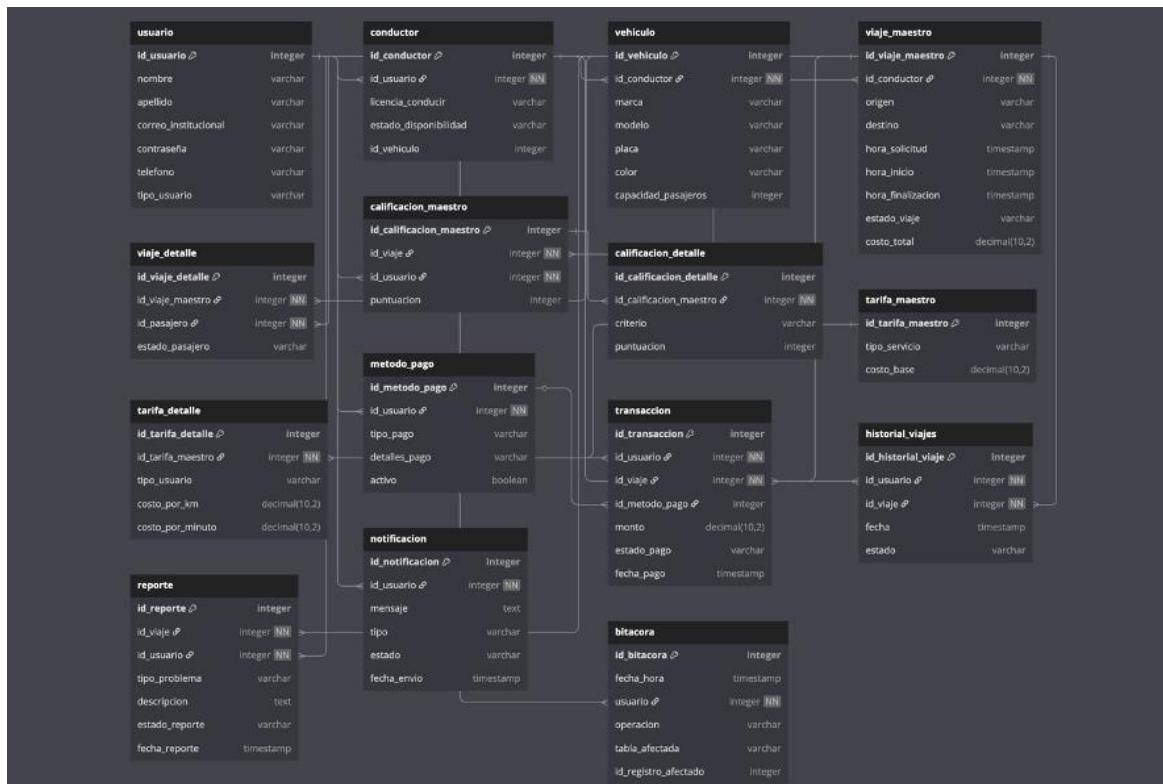


Diagrama Entidad-Relación:



VI. Diseño

1. Estimación de Dimensiones del Sistema

- Cantidad de usuarios probables

La UVG tiene aproximadamente 9,000 - 15,000 entre estudiantes y trabajadores.

Se estima que 1,000 - 2,500 usuarios podrían registrarse en la app dentro del primer año.

Usuarios concurrentes estimados: 50 - 300 en horas pico.

- Tiempo promedio de respuesta

La app debe procesar solicitudes en menos de 1 segundo en acciones clave como:

Buscar viajes disponibles.

Crear un nuevo viaje.

Confirmar participación en un viaje.

Para esto, el backend debe estar optimizado con caché (Redis) y consultas eficientes.

- Cantidad de usuarios promedio simultáneos

Durante picos (mañana y tarde), se espera 50 - 300 usuarios concurrentes.

Se requiere balanceo de carga para evitar cuellos de botella.

- Escalabilidad del sistema,

para esto debe considerarse la información que deberá gestionar a medida que vaya creciendo por el uso. Deberían estimar el tamaño en disco que se esperaría de la información almacenada.

- Disponibilidad de la información.
- Confiabilidad, tolerancia a fallos masivos, replicación de la información.

Solución: Uso de PostgreSQL para datos estructurados + Firebase Firestore para datos en tiempo real.

- Disponibilidad de la Información

La app debe estar disponible 24/7, con tiempo de inactividad mínimo. Los backups se harían automáticamente al menos 1 al día

- Confiabilidad y Tolerancia a Fallos

Autenticación segura con Firebase Authentication (Google, correo, contraseña).

Replicación de base de datos en AWS RDS para evitar pérdida de información.

Mensajería en tiempo real con Firebase para mantener las notificaciones activas.

Selección de Tecnologías para el Desarrollo

Frontend: React Native con TypeScript

- Ventajas:

Multiplataforma (iOS y Android).

Alto rendimiento con React Native Hermes.

Comunidad amplia y gran soporte (Facebook/META).

Reutilización de código (posible integración con web).

Soporta WebSockets y notificaciones push fácilmente.

- Desventajas:

Mayor dependencia de librerías de terceros.

Algunas funciones requieren módulos nativos adicionales.

- ¿Por qué React Native?

Elegimos React Native por el amplio soporte que tiene para desarrollar y que es una crossplataform y nos permite usar el mismo código para todos los móviles tanto android como IOs

Backend: Node.js con Express.js

- Ventajas:

Alto rendimiento con asincronía.

Compatible con WebSockets para tiempo real.

Escalabilidad en aplicaciones con múltiples conexiones.

- Desventajas:

No es eficiente para procesamiento intensivo en CPU.

Puede volverse desordenado sin una buena estructura.

- ¿Por qué Node.js?

Node.js se integra fácilmente con React Native mediante APIs REST o GraphQL, permitiendo una comunicación fluida entre el backend y la aplicación móvil. Su arquitectura basada en eventos y su modelo asíncrono lo hacen altamente escalable, soportando múltiples usuarios simultáneos sin afectar el rendimiento. Además, al combinarlo con WebSockets, facilita la comunicación en tiempo real, ideal para funcionalidades como notificaciones, chats y actualizaciones en vivo dentro de la app.

Bases de Datos

PostgreSQL

- Ventajas:

Relacional y estructurado, ideal para datos de usuarios y viajes.

Alta consistencia y transacciones seguras.

Compatible con escalabilidad vertical y horizontal.

Funciones avanzadas para geolocalización (PostGIS).

- Desventajas:

Más configuración inicial que otros sistemas NoSQL.

Menos flexible que NoSQL para datos dinámicos.

- ¿Por qué PostgreSQL?

PostgreSQL es la mejor opción para este proyecto debido a su capacidad para manejar datos estructurados, permitiendo gestionar eficientemente usuarios, historial de viajes y pagos con consultas SQL avanzadas. Además, su extensión PostGIS facilita el almacenamiento y análisis de datos geoespaciales, ideal para rastrear ubicaciones y optimizar rutas. Su escalabilidad permite mejorar el rendimiento mediante índices y particiones, asegurando que el sistema pueda crecer sin comprometer la velocidad ni la estabilidad.

Primer Sprint

Pila del Sprint

- Historias de usuario del sprint
 - Como **estudiante** sin vehículo, quiero una alternativa accesible, segura y confiable para poder llegar a la universidad sin depender de un solo medio de transporte y sin gastar demasiado.
 - Como **profesor**, quiero acceder a un medio de transporte seguro y cómodo que me permita transportar pertenencias personales, para poder llegar puntualmente a mis clases.
 - Como **equipo de desarrollo**, necesitamos tener el entorno de desarrollo listo y funcional para poder empezar a construir las funcionalidades del sistema.

Primer calendario que trabajamos

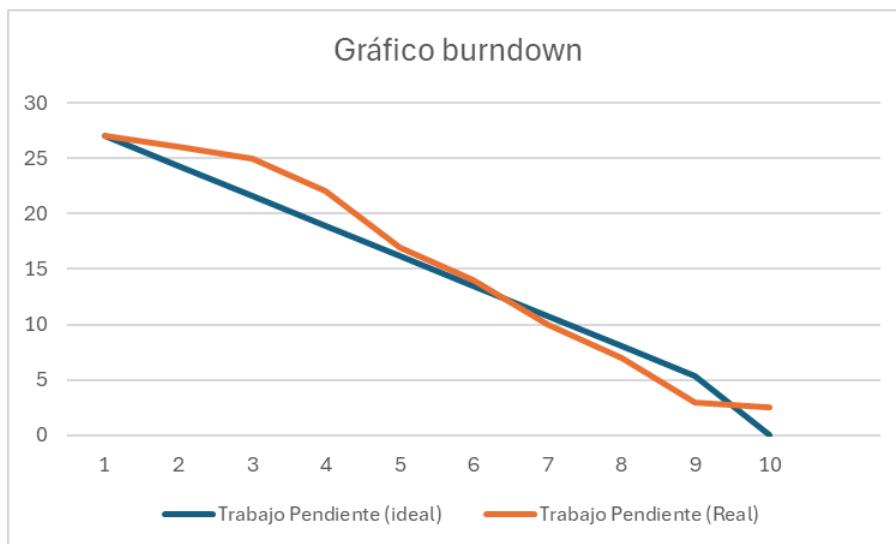
- Calendario de planificación del sprint

Nombre de la tarea	Descripción	Horas	Responsable	Fecha Estimada
Instalar herramientas base	Instalar Node.js, Git, PostgreSQL y configurar entorno local	2	Manuel Ulin	
Crear proyecto React Native	Iniciar app móvil con Expo y TypeScript	2	Jose Ordoñez	
Crear proyecto Node.js	Iniciar backend con Express y estructura MVC básica	3	Jose Ordoñez	
Instalar librerías frontend	Axios, React Navigation, componentes UI	2	Marcos Ambrocio	
Configurar navegación básica	Stack Navigator para login, registro y pantalla principal	2	Marcos Ambrocio	
Crear modelo de usuario	Clase User con id, correo, contraseña, tipo_usuario	2	Manuel Ulin	

Crear endpoint de registro	POST /register con validaciones y conexión a BD	3	Alejandro Anton	
Interfaz de registro móvil	Formulario en React Native, validaciones básicas	3	Alejandro Anton	
Conectar frontend con backend	Axios para enviar datos del registro al backend	2	Sebastián Espinal	
Validación y cifrado	Cifrar contraseña con bcrypt y validar datos en backend	2	Sebastián Espinal	
Crear base de datos y tabla	Crear BD en PostgreSQL y tabla de usuarios	2	Adrián González	
Probar flujo de registro	Registrar desde app móvil y verificar en BD que se guarde el usuario	2	Adrián González	

Resultados del Sprint

o Gráfico burndown.



A pesar de que el equipo no completó todas las tareas planificadas (quedaron pendientes 2.5 horas de trabajo), el indicador de éxito del sprint sigue siendo alto, alcanzando un 90.74%. Este valor refleja un desempeño eficiente y comprometido, ya que el equipo fue capaz de completar la mayor parte del trabajo planeado.

Las tareas pendientes están relacionadas con el cifrado de contraseñas y la integración de la base de datos, que son funcionalidades cruciales pero que no se completaron a tiempo debido a imprevistos técnicos. Sin embargo, el hecho de que se haya completado casi el total de las tareas planificadas indica que el equipo se mantuvo en el camino correcto y estuvo cerca de cumplir con los objetivos establecidos.

Segundo Sprint

Sprint Backlog

La pila del sprint actual.

- Especificación de cada tarea a implementar con sus puntos de historia y/o horas.

La pila del Sprint 2 contiene las tareas seleccionadas del Product Backlog que el equipo se comprometió a completar durante el sprint. Cada tarea tiene asignada una estimación en horas.

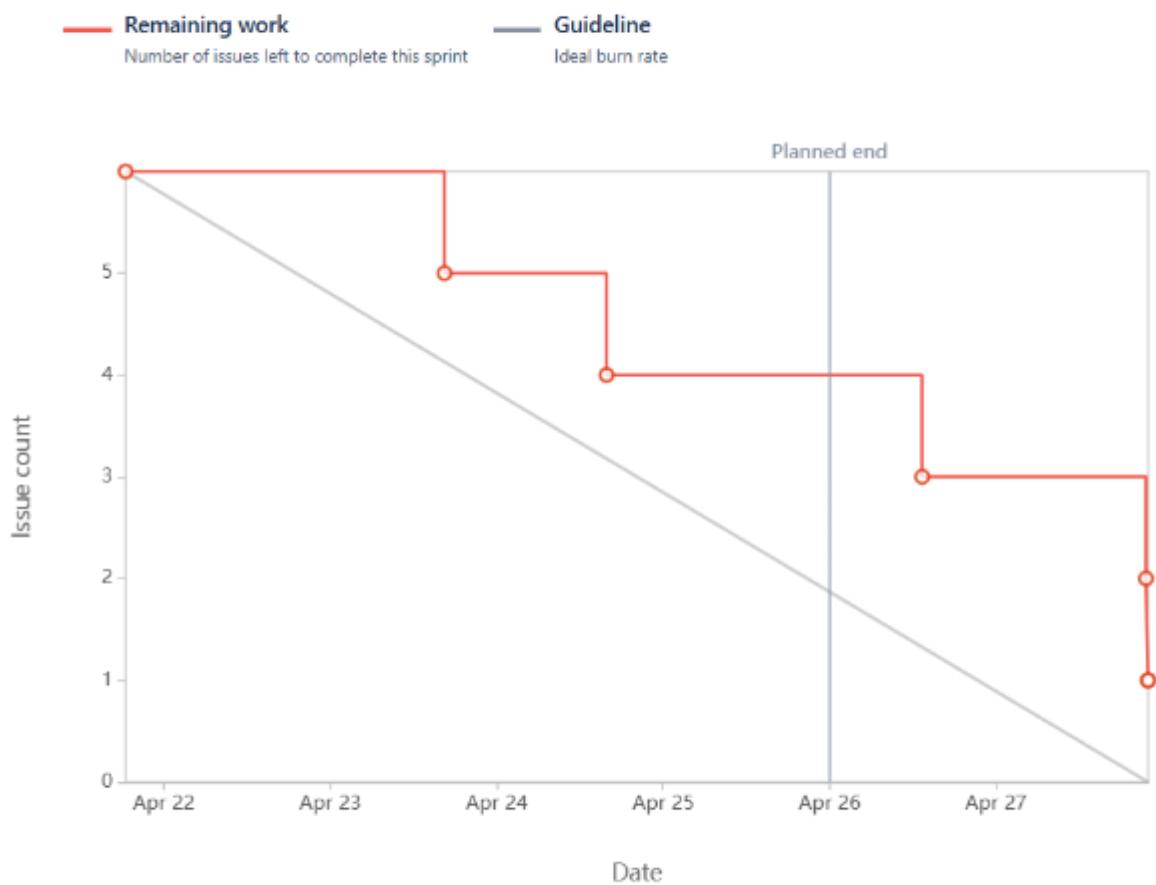
Tarea	Descripción	Estimación (horas)	Estado
SCRUM-6	Continuar Sprint 2	2 horas	Completada
SCRUM-12	Configurar emulador	3 horas	Completada
SCRUM-7	Pantalla Usuario	4 horas	Completada
SCRUM-8	Navegación entre pantallas	5 horas	Completada
SCRUM-9	Corregir Docker	3 horas	En progreso
SCRUM-11	Pantalla de viaje	4 horas	Completada
SCRUM-10	Conexión con base de datos	5 horas	Por hacer

Notas:

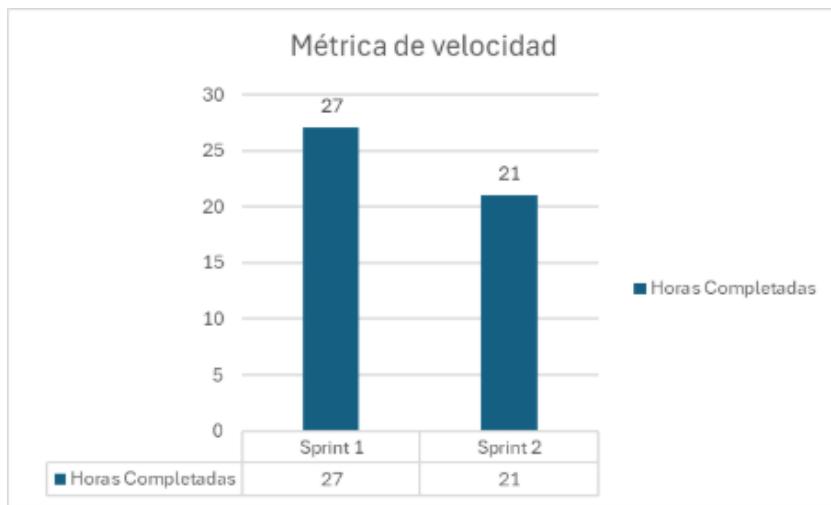
- Las estimaciones fueron asignadas considerando la complejidad y el tiempo disponible del equipo.
- La mayoría de las tareas fueron completadas o se encuentran en su fase final.

Resultados del Sprint

o Gráfico burndown.



Métrica de velocidad



- Se lograron completar 5 de 6 tareas planificadas.
- El equipo mantuvo un ritmo constante de trabajo.
- Sólo una tarea técnica (SCRUM-10) quedó en progreso, lo cual representa una desviación.
- El impacto de la tarea pendiente no afecta gravemente la funcionalidad ya entregada.

Lo que salió bien:

Mejor organización y flujo de trabajo: Se notó una mejora en la organización del equipo, apoyada por el uso de Jira para gestionar tareas y el seguimiento de las actividades.

Avances significativos: El equipo completó la mayoría de las tareas planificadas, destacando la implementación de la pantalla de usuario, la navegación entre pantallas y la pantalla de viaje, lo que contribuyó a un progreso tangible del proyecto.

Áreas de mejora:

- **Priorización de tareas técnicas:** La tarea de la conexión a la base de datos (SCRUM-10) quedó pendiente, lo que retrasó el trabajo en otras áreas. A futuro, sería útil priorizar las tareas críticas técnicas para evitar cuellos de botella.
- **Mayor visibilidad de avances diarios:** A pesar de tener reuniones diarias de Scrum, el equipo podría beneficiarse de una mayor visibilidad en cuanto al progreso de las tareas, especialmente en los aspectos más técnicos.

Tercer Sprint

Product Backlog

A continuación, se listan las funcionalidades y tareas planificadas, indicando cuáles fueron completadas y cuáles permanecen pendientes:

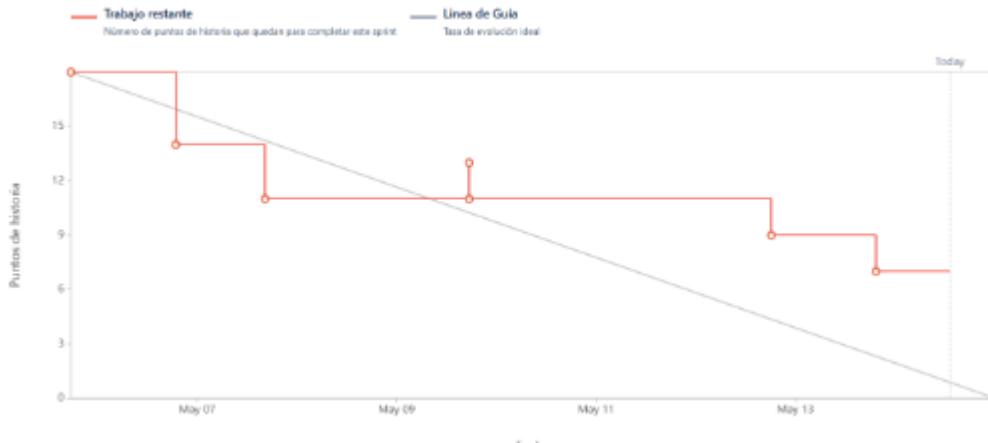
Tarea	Descripción de tarea	Estado
SCRUM-10	Conexión con base de datos	Completada
SCRUM-14	Subir aplicación al servidor	En progreso
SCRUM-15	Añadir mapa interactivo	Completada
SCRUM-16	Pantalla de usuarios	Completada
SCRUM-17	Creación de viajes	Completada
SCRUM-18	Arreglar docker	Completada
SCRUM – 19	Inicio de sesión y registro con usuarios reales	En progreso

Funcionalidades terminadas:

- Establecer conexión con base de datos.
- Añadir mapa interactivo.
- Completar la pantalla de usuarios.
- Añadir la creación de viajes.
- Correcto funcionamiento del docker.

Resultados del Sprint

Gráfico burndown:



Métrica de velocidad (Gráfico con los datos de todos los sprints hasta el momento):



Durante el Sprint 3, el equipo tuvo un compromiso inicial de 18 story points y logró completar 13 story points, lo que representa un **72% de cumplimiento** del objetivo del sprint.

El aumento momentáneo en la carga de trabajo (visible como un leve incremento en story points en el burndown chart) sugiere una incorporación tardía de tareas o una reestimación, lo cual podría ajustarse en futuros sprints con una planificación más estable. Aun así, el equipo supo adaptarse y logró cumplir la mayoría de los objetivos, destacando la mejora del flujo de desarrollo y el funcionamiento del entorno con Docker.

En el Sprint 3, el rendimiento del equipo es principalmente positivo. Se han realizado la mayoría de las tareas planificadas, incluidas las funciones principales, como la

conexión de la base de datos, la creación de turismo, las pantallas de usuario y las tarjetas interactivas. Los aspectos técnicos importantes también se han resuelto, como la actividad correcta del entorno Docker. Esto muestra una mejora extraordinaria en el potencial técnico y la coordinación grupal. Sin embargo, se han determinado los aspectos a mejorar. Algunas tareas, como iniciar sesión con usuarios reales y aumentar las aplicaciones del servidor, no se realizan al 100%. Esto muestra que todavía existe la oportunidad de mejorar la evaluación de los esfuerzos necesarios, así como la gestión del tiempo y la distribución de la misión. Además, después de eso, se aceleran las tareas que afectan el cumplimiento común de las necesidades, lo que muestra que es necesario hacer un plan más estable y preciso desde el principio.

Lean UX Canvas

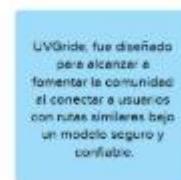
Capturas del proceso completo.

Cuadro 1. ¿Qué problema del negocio estás tratando de resolver?

1 Redacte su problema del negocio actual usando la siguiente plantilla:

Nuestro servicio/producto fue diseñado para alcanzar **[estos metas]**, sin embargo [de esta forma] que el producto/servicio no está alcanzando estos objetivos por lo que está causando **[estos efectos adversos/problemas de negocio]** al negocio.

Cómo podríamos mejorar el servicio/producto para que nuestros clientes tengan más entusiasmo determinado por **[estos cambios positivos en el comportamiento del cliente]**.



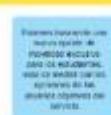
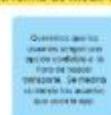
2 Cuando terminé, Copie su nueva declaración del problema del negocio en su Lean UX Canvas.

Cuadro 2. ¿Cómo se resolvió la declaración del negocio o el business statement?

1) Trabajando individualmente, tome 5 minutos para responder esta pregunta:

• ¿Qué resultados está buscando su producto? Cómo se medirán?
Escriba un resultado y una forma de medir en cada postit.

Sugerencia: ¿Qué harán las personas/usuarios de manera diferente si sus soluciones funcionan? Considere las métricas que indican el éxito del cliente, como el valor promedio del pedido, el tiempo en el sitio y la tasa de retención.



2) Ahora trabajando en equipo, tome 5 minutos:

• Coloca tus notas debajo. Léelos en voz alta al equipo.

• Ordenelos en temas que tengan sentido y asigne un nombre a cada tema.



3) En 5 minutos, vote por los 3 mejores resultados. Píselos en los que cree que su equipo debería centrarse primero. Estos van en la casilla #2 de su Lean UX Canvas.

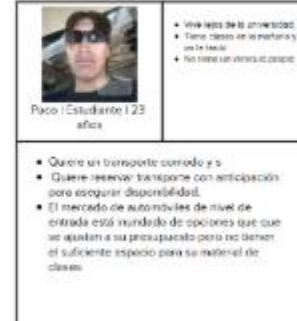
Cuadro 3. Usuarios

¿En qué tipos (es decir, personas) de usuarios y clientes debe centrarse primero? (Sugerencia: ¿Quién compra su producto o servicio? ¿Quién lo usa? ¿Quién lo configura?)

1) Trabaja con tu equipo para crear 2 Prototipos Personas que creas que representan parte del público objetivo para tu producto / servicio. Estos deberían ser los segmentos o audiencia a los que se desea apuntar primero.

A la derecha hay un ejemplo de Prototipo persona y debajo hay dos plantillas de Proto-Persona para que completes.

2) Cuando haya terminado esta información irá al cuadro 3 del Lean UX Canvas.



Cuadro 4. Resultados (Outcomes) y Beneficios a los usuarios

¿Por qué sus usuarios buscan su producto o servicio? ¿Qué beneficio obtendrán el usuario? ¿Qué cambio de comportamiento podemos observar que nos dice que han logrado su objetivo?

- Comenzando con la sesión anterior de tus Personas, pregunta de tener tanto los objetivos como los beneficios. Estos te ayudarán con esta actividad.
- Descubrir al menos 3 resultados de usuario, cambios mentales y cambios en el comportamiento del usuario.
- Pide que estén tratando de lograr por sí mismos?
 - ¿Por qué buscan tu producto o servicio?
- Esta información irá en el Cuadro 4 del Lean UX Canvas.

Sugerencia: Ahorre dinero, obtenga una promoción, pase más tiempo con la familia.

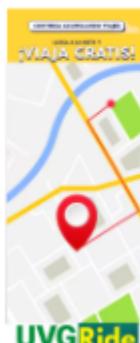


4

Cuadro 5. Soluciones

¿Qué podemos hacer para resolver nuestro problema comercial y satisfacer las necesidades de nuestros clientes al mismo tiempo? Encuentra aquí las ideas de producto, características o mejoras.

- Recuerda nuestro problema comercial, los usuarios que tienen objetivos, sus motivaciones y el comportamiento que nos gustaría ver en ellos ...
- Trabaja con tu equipo para responder la siguiente pregunta:
- ¿Qué solución servirá a nuestras posibles y creará los resultados deseados?
- Crea un dibujo simple, boceto o pictograma para aumentar sus ideas
- Estas ideas irán al cuadro 5 del Lean UX Canvas



Cuadro 6. Hipótesis

Combine los supuestos de 2.3.4 y 5 en la siguiente declaración de hipótesis:

"Creemos que este [resultado comercial] se logrará si [el usuario] alcanza [beneficio] con [función]."

(Sugerencia: cada hipótesis debe centrarse solo en una característica).

Dile a tu pareja: Cada pareja escribe una declaración de hipótesis que cree que puede resolver su problema comercial para su persona objetivo.

Cosas clave para recordar:

- Ya has indicado los supuestos que vamos a utilizar para formar el enunciado de hipótesis en los cuadros 2.3.4.5.
- Recombinándolos en este formato nos ayuda a ver si incluso lo creemos.
- Cada hipótesis debe centrarse solo en una característica / funcionalidad

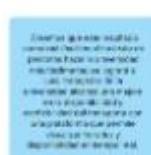
Luego irás al Cuadro de priorización de hipótesis que está más arriba.

We believe that [business outcome] will be achieved if [user] attains [outcomes & benefits] with [feature].

Example:

We believe that sales of Ford's existing cars will increase if features, like 10% of family sellers at Ford successfully drives more consumers to Ford dealerships with the ability to Buy Ford + Directions installed at the dealership.

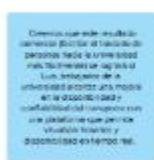
We believe an increase in percentage of active riders will be achieved if Carlos the elderly driver attains driving confidence and the ability to see his grandkids more with 1 tap "Navigate to Sweet Location".



6

Cuadro 7. ¿Qué es lo más importante que debemos aprender?

Como equipo, elija qué hipótesis le gustaría probar primero.
Luego pregunte: ¿Qué es lo más importante que debemos aprender? ¿Qué suposición causaría un fracaso total si nos equivocamos?



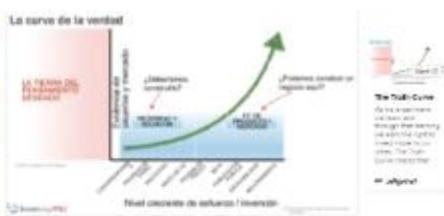
- Lo más importante que debemos aprender:
Si Luis usaría una plataforma digital para consultar horarios y disponibilidad de transporte y confiaría en ella para planificar su traslado.
- Suposición crítica que causaría fracaso: Que Luis no use la plataforma porque no confie en la información o no tiene el hábito de planificar su transporte con tecnología.

Cuadro 8. ¿Qué es lo más importante que debemos aprender?

Mirando su respuesta en el Cuadro 7 del lienzo Lean UX, pregunte:
¿Cuál es la menor cantidad de trabajo que debemos hacer para aprender eso? ¿Qué técnicas podríamos usar?

Discuta las opciones en equipo y diseñe diferentes experiencias en cada uno de los siguientes plazos. Cada uno debe "montar la Curva de la Verdad".

Pese luego los postit al cuadro 8 del Lean UX Canvas

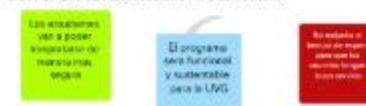


Las técnicas para validar la hipótesis con mínimo esfuerzo que utilizamos:

- 1 Entrevistas cualitativas con trabajadores universitarios para entender sus hábitos actuales
- 2 Prototipo de baja fidelidad (papel o digital simple) para probar la interacción básica
- 3 Landing page de prueba con formulario de interés
- 4 Simulación con datos falsos (Wizard of Oz) para mostrar disponibilidad en tiempo real
- 5 A/B testing con diferentes versiones de información de transporte

- Diseño de experiencias por plazos (Curva de la Verdad)
- Corto plazo (1-2 semanas)
 - Crear prototipo en papel o herramienta como Figma
 - Realizar 5-10 entrevistas con trabajadores universitarios mostrando el prototipo
 - Medir interés y comprensión del concepto
 - Medio plazo (3-4 semanas)
 - Desarrollar MVP mínimo con funcionalidad básica de consulta
 - Implementar en un área pequeña del campus
 - Medir采纳 y confiabilidad percibida
 - Largo plazo (6-8 semanas)
 - Integración con sistemas reales de transporte
 - Prueba piloto con grupo más amplio
 - Medir impacto real en disponibilidad y confiabilidad del transporte

Capturas del lienzo de estrategia del producto.

<p>Problema del negocio</p> <p>¿Qué idea de negocio has identificado que necesita ayuda?</p> <p>1</p> 	<p>Ideas de solución</p> <p>Entra más ideas de productos/características mejoras que ayuden a tu público objetivo a lograr las necesidades que buscan.</p> 	<p>Outcomes del negocio</p> <p>¿Qué cambios en el comportamiento del cliente indicarían que ha resuelto un problema real de una manera que agrega valor?</p>   
<p>Usuarios y Clientes</p> <p>¿En qué tipos de usuarios y clientes deben centrarse primero?</p>  <ul style="list-style-type: none"> • Los estudiantes y los profesores tienen diferentes necesidades y expectativas de la aplicación. • Los profesores tienen más responsabilidades y necesitan más información. • Los estudiantes tienen más tiempo libre y necesitan una aplicación que sea fácil de usar y no requiera mucha concentración. 		<p>Beneficio a los usuarios / clientes</p> <p>¿Cuáles son los objetivos que intentan alcanzar sus usuarios? ¿Qué los motiva a tratar su solución?</p> 
<p>Hipótesis</p> <p>Combiné los supuestos de 2,3 y 5 en una plantilla de descripción de hipótesis.</p> <p>Cada hipótesis deberá basarse en una característica</p> 	<p>¿Qué es lo más importante que tenemos que aprender primero?</p> <p>Para cada hipótesis, identifica el supuesto más arriesgado. Esta es la suposición que haría lo ideal fracasar si esté mal.</p> 	<p>¿Cuál es la menor cantidad de trabajo que tienes que hacer para aprender la siguiente suposición más importante?</p> <p>Pensa en los tipos de experimentos que puedes realizar para saber si tu suposición más arriesgada es verdadera o falsa.</p> 

Download this canvas at: www.vfpoائد.com/toolkit.html

Adapted from Jeff Paton's Opportunity Canvas. Download at: <http://jeffpaton.com/opportunity-canvas>

opportunity-canvas

Cuarto Sprint

Sprint Backlog

La pila del Sprint 4 contiene las tareas seleccionadas del Product Backlog que el equipo se comprometió a completar durante el sprint. Cada tarea tiene asignada sus story points.

ID Tarea	Descripción	Story Points	Estado
SCRUM-14	Subir aplicación al servidor	1	 En progreso

SCRUM-19	Inicio de sesión y registro	2	<input checked="" type="checkbox"/>	Completado
SCRUM-20	Backend HomeScreen	4	<input checked="" type="checkbox"/>	Completado
SCRUM-21	Backend LoginScreen	4	<input checked="" type="checkbox"/>	Completado
SCRUM-22	Backend ProfileScreen	4	<input checked="" type="checkbox"/>	Completado
SCRUM-23	Backend RegisterScreen	4	<input checked="" type="checkbox"/>	Completado
SCRUM-24	Backend SettingScreen	4	<input checked="" type="checkbox"/>	Completado
SCRUM-25	Backend TravelScreen	4	<input checked="" type="checkbox"/>	Completado
SCRUM-26	Backend TripFormScreen	4	<input checked="" type="checkbox"/>	Completado
SCRUM-27	Backend UserScreen	4	<input checked="" type="checkbox"/>	Completado
SCRUM-28	Consolidar documento final del sprint	10	<input checked="" type="checkbox"/>	Completado

Total de Story Points comprometidos: 45

Story Points completados: 44

Notas:

- Las estimaciones fueron asignadas considerando la complejidad y el tiempo disponible del equipo.
- La mayoría de las tareas fueron completadas o se encuentran en su fase final.

Durante el Sprint 4 se completaron las siguientes funcionalidades:

- **Inicio de sesión y registro con usuarios reales**
 - Implementación completa del flujo de autenticación con JWT y bcrypt.
- **Backend HomeScreen**
 - Endpoints para cargar datos dinámicos de inicio.
- **Backend LoginScreen**
 - Manejo de autenticación y validación de credenciales.
- **Backend ProfileScreen**

- o Obtención y actualización de perfil de usuario.
- **Backend RegisterScreen**
 - o Registro seguro de usuarios en la base de datos.
- **Backend TravelScreen**
 - o Listado de viajes y lógica para filtrado por categorías.
- **Backend TripFormScreen**
 - o Envío y recepción de datos para crear o modificar viajes.
- **Backend SettingScreen**
 - o Configuración de preferencias del usuario.
- **Backend UserScreen**
 - o Consulta y modificación de datos personales.
- **Consolidar documento final del sprint**
 - o Documento integrador con datos de todos los sprints.

Resultados del Sprint

Gráfico burndown:



Métrica de velocidad (Gráfico con los datos de todos los sprints hasta el momento):



Durante el Sprint 4, el equipo mostró una mejora significativa en comparación con los sprints anteriores, tanto en términos de compromiso como de trabajo completado.

Según el **Velocity Report**, el equipo se comprometió a completar 50 puntos de historia y logró terminar 44, alcanzando un **90% de cumplimiento del compromiso**. Esta cifra representa un avance notable en relación con los sprints anteriores (Sprint 3 tuvo un cumplimiento del 72% y Sprint 2 no tuvo compromiso registrado por no trabajar en Jira). Además, la **velocidad promedio del equipo** (19 puntos) fue ampliamente superada en este sprint, lo que indica un aumento en la productividad y madurez del equipo.

¿Qué hicimos bien?

- **Alta productividad:** El equipo logró completar 44 de los 50 puntos comprometidos, alcanzando un 88% de cumplimiento. Esta es una mejora considerable respecto a los sprints anteriores.

¿Qué podemos mejorar?

- **Inicio lento del sprint:** El Burndown Chart muestra que no se avanzó significativamente durante la primera mitad del sprint. Esto sugiere una necesidad de mejorar la planificación o el enfoque inicial del equipo.

- **Carga de trabajo ligeramente sobredimensionada:** Aunque se completó la mayoría de los puntos, el equipo no alcanzó el 100%. Para futuros sprints, podría ajustarse mejor el compromiso según la capacidad real.



Reflexión Final del Proyecto (Fin del Semestre)

Este último sprint marcó el cierre de una etapa clave en el desarrollo del proyecto. A lo largo del semestre, el equipo enfrentó distintos retos relacionados con la planificación, coordinación y ejecución técnica, pero logró mantener una curva de aprendizaje constante que se reflejó en la mejora de la velocidad, la calidad de los entregables y la organización interna.

El Sprint 4 evidenció el punto más alto de rendimiento del equipo, tanto en términos de compromiso como de cumplimiento, lo cual demuestra la madurez alcanzada en esta fase del proyecto. Si bien no se completó el 100% del compromiso, se logró una cobertura del 88%, una cifra positiva que muestra compromiso, responsabilidad y capacidad de respuesta frente a los desafíos.

Aún quedan funcionalidades por implementar y aspectos por refinar, lo cual será abordado en el siguiente semestre. No obstante, el trabajo realizado hasta ahora ha sentado las bases técnicas y metodológicas para continuar de manera sólida en futuras iteraciones.

Doc en el que se trabajó : [Documento integrador.docx](#)