

Capítulo V: Product Implementation, Validation & Deployment.

5.1. Software Configuration Management.

Esta guía define las decisiones y acuerdos fundamentales para el desarrollo, mantenimiento y despliegue de la aplicación **WeRide**, que gestiona el alquiler de vehículos. El objetivo es asegurar la coherencia, eficiencia y calidad a lo largo del ciclo de vida del proyecto.

5.1.1. Software Development Environment Configuration.

Project Management

Github

Plataforma en línea que permite almacenar código fuente en repositorios. Gracias a la tecnología de control de versiones de Git se puede organizar el código y llevar un mejor trabajo en conjunto.

Whatsapp

Red Social destinada a la comunicación donde se realizaron acuerdos y recordatorios de las reuniones.

Trello

Software de administración y gestión de proyectos que se utilizó para establecer y designar las tareas Requirements Management

Miro

Plataforma en línea de gestión de requisitos que permite colaborar y organizar proyectos de forma visual y representativa. Product UX/UI Design

Figma

Aplicación que permite el diseño libre de interfaces a través de las múltiples herramientas que ofrece. Permitiendo la creación de prototipos interactivos que simulan la experiencia de usuario. Software Development

Git

Es un software de control de versiones para los trabajos en equipos y confiabilidad del desarrollo.

Node.js

Node.js es un entorno de ejecución de JavaScript del lado del servidor, que permite desarrollar aplicaciones web escalables y de alto rendimiento fuera del navegador.

HTML

Lenguaje de etiquetas, utilizado para la estructuración y la presentación de contenido.

CSS

CSS es un lenguaje utilizado para estilizar y dar formato a documentos HTML.

TypeScript

TypeScript es un lenguaje de programación de alto nivel, interpretado y multi-paradigma, utilizado para crear interactividad en páginas web.

VSCode

Es un editor de código fuente con extensiones que ayudan al desarrollo.

WebStorm

Es un IDE centrado en desarrollo frontend, por su gran variedad de herramientas que agilizan el proceso de desarrollo.

Angular

Framework robusto para el desarrollo de aplicaciones web modernas y escalables, basado en el patrón de Single Page Application (SPA), facilita la creación de interfaces dinámicas mediante TypeScript.

IntelliJ Idea

Entorno de desarrollo integrado (IDE) utilizado principalmente para el desarrollo en Java, que ofrece herramientas avanzadas para la codificación, depuración y pruebas.

Java

Lenguaje de programación orientado a objetos, utilizado para desarrollar aplicaciones robustas y portables.

Springboot

Framework para el desarrollo de aplicaciones Java basadas en microservicios, que facilita la creación de aplicaciones robustas y escalables.

Swagger

Herramienta para diseñar, construir, documentar y consumir APIs RESTful.

MySQLWorkbench

Herramienta visual para el diseño, modelado, generación y administración de bases de datos MySQL. Software Deployment

Github Pages

Plataforma que nos permite realizar el despliegue de nuestro landing page.

5.1.2. Source Code Management.

Hemos optado por crear un repositorio en GitHub para nuestro proyecto, tanto para el informe como para la landing page. Esto facilitó la colaboración entre los miembros del equipo, aprovechando las herramientas útiles que esta plataforma ofrece para el manejo del código fuente y sus versiones.

- URL del repositorio Report en GitHub: <https://github.com/OpenSource-Grupo-4/ReportTB1>
 - URL del repositorio Landing Page en GitHub: <https://github.com/OpenSource-Grupo-4/Landing-Page>
 - URL del repositorio de la Web Application en GitHub: <https://github.com/OpenSource-Grupo-4/Frontend-WeRide>
 - URL del repositorio de los Web Services en GitHub: <https://github.com/OpenSource-Grupo-4/Backend-WeRide>
-

5.1.3. Source Code Style Guide & Conventions

Landing Page:

Para "WeRide", hemos utilizado "**HTML y CSS**". Para estructurar el contenido usamos etiquetas de section y divisiones para contenido específico de cada una de las secciones. Además, hemos empleado atributos como **HTML Style** para personalizar el aspecto visual, definiendo propiedades como color, tamaño de fuente y tipo de letra.

Para resaltar elementos importantes, hemos aplicado **HTML Text Formatting**, incluyendo etiquetas como b para negrita, strong para resaltado y del para mostrar cambios de precios. En cuanto a la navegación, hemos implementado una barra de navegación horizontal utilizando **CSS** para mejorar la experiencia del usuario al explorar el contenido.

Los formularios, creados con **CSS**, permiten a los usuarios ingresar información relevante, como detalles de inicio de sesión, información de pago y dirección de envío. Para añadir interactividad, hemos agregado botones con efectos hover utilizando CSS y paginación CSS para facilitar la navegación entre las diferentes páginas de productos.

Finalmente, en el **footer**, hemos incluido enlaces a las redes sociales de la organización para brindar a los usuarios una forma adicional de conectarse y seguir nuestras actualizaciones.

Web Application:

Para "WeRide App", hemos utilizado "**TypeScript, HTML y CSS**". La estructura del proyecto sigue el patrón de arquitectura de **Single Page Application (SPA)**, utilizando el framework **Angular** para organizar el código en módulos, componentes y servicios.

Los componentes están estructurados según DDD (Domain-Driven Design), donde cada componente representa un bounded context específico, como **auth**, **booking**, **garage**, **plans** y **trip**. Cada componente tiene su propio archivo HTML para la estructura, CSS para el estilo y TypeScript para la lógica y enrutamiento así como la comunicación entre componentes y servicios.

5.1.4. Software Deployment Configuration.

Landing Page:

Utilizaremos GitHub Pages para alojar nuestra Landing Page. Para lograrlo, subiremos los archivos esenciales (HTML, CSS, etc.) a un repositorio público en GitHub. De esta manera, nuestra página estará disponible en línea y accesible para todos los usuarios.

The screenshot shows a GitHub repository named 'ReportTB1'. The repository has 42 commits, 6 branches, and 0 tags. The README file contains the text 'UNIVERSIDAD PERUANA DE CIENCIAS APLICADAS' and features a red flame logo. The repository has 0 stars, 0 forks, and 0 watching. It also includes sections for About, Releases, Packages, and Contributors.

Web Application:

Utilizaremos Vercel para alojar nuestra Web Application. Para lograrlo, configuraremos un proyecto en Vercel y conectaremos nuestro repositorio de GitHub. Vercel se encargará de la construcción y el despliegue de nuestra aplicación automáticamente cada vez que realicemos un push a la rama principal.

5.2. Landing Page, Services & Applications Implementation.

5.2.1. Sprint 1

5.2.1.1. Sprint Planning 1.

Elemento	Detalle
Sprint	1 - Implementación de funcionalidades básicas y diseño inicial de CultivApp
Sprint Planning Background	
Fecha	19/09/2025
Hora	22:00 horas (GMT-5)
Lugar	Reunión virtual mediante Discord
Preparado por	[WeTech]
Asistentes	[Samuel Bonifacio , Jefferson Castro, Diego Seminario, Jhemy Romero, Arnold Morales]
Sprint n-1 Review	Se creó la organización de WeTech en GitHub, se definieron ramas y se asignaron tareas iniciales como investigación de usuarios, wireframes y mockups.
Sprint n-1 Retrospective	Se implementó la estructura básica de la landing page utilizando HTML, CSS y JavaScript.
Sprint Goal & User Stories	1-20
Sprint 1 Velocity	17 puntos
Sum of Story Points	17 puntos

Objetivo del Sprint

Implementar la estructura inicial de la aplicación, el diseño de la landing page y las funcionalidades básicas de registro e inicio de sesión para los usuarios de WeRide.

User Stories seleccionadas para el Sprint 1

ID	User Story	Puntos
12	Como usuario quiero contactar con el soporte de la aplicación para resolver mis dudas.	5
10	Como usuario quiero tener acceso a la información de la aplicación para estar informado de las actualizaciones.	5
11	Como usuario quiero saber dónde se ubican presencialmente para acudir a uno de los locales.	4
14	Como usuario, quiero reportar un problema con el vehículo para alertar a soporte y obtener ayuda rápida.	3

Total de puntos: 17

5.2.1.2. Aspect Leaders and Collaborators

En esta sección se presenta la matriz de liderazgo y colaboración (Leadership-and-Collaboration Matrix, LACX) para el Sprint 1 de WeRide. Esta matriz identifica, para cada aspecto clave del Sprint, quién es el líder responsable y quiénes son los colaboradores, facilitando así la comunicación y la asignación de tareas dentro del equipo.

Los aspectos considerados en este Sprint incluyen: diseño de la landing page, desarrollo de funcionalidades de registro e inicio de sesión, implementación de la barra de navegación, y configuración del footer con enlaces de contacto y redes sociales.

Team Member (Apellido, Nombre)	GitHub Username	Landing Page Design	Registro/Inicio de Sesión	Barra de Navegación	Footer y Redes Sociales
Bonifacio, Samuel	samuelbonifacio015	L	L	L	L
Castro, Jefferson	JeffersonCastroPariona	C	L	C	C
Seminario, Diego	DiegoSeminario	C	C	L	C
Romero, Jhemy	jhemyromeromeza	L	L	L	C
Morales, Arnold	Arnold-TI	C	C	C	L

L: Leader (Líder)

C: Collaborator (Colaborador)

Esta organización permite una asignación clara de responsabilidades y fomenta la colaboración efectiva durante el desarrollo del Sprint.

5.2.1.3. Sprint Backlog 1.

ID	Title/Section	Description	Estimation (Hours)	Assigned To	Status
US01	Landing Page	Como usuario, quiero acceder a una landing page clara y atractiva para conocer los beneficios y funcionalidades de WeRide.	3	Bonifacio Jaramillo, Samuel Jesus	Done
US02	Registro de Usuario	Como usuario, quiero registrar una cuenta desde la landing page para poder acceder a los servicios de WeRide.	3	Castro Pariona, Jefferson Ernesto	Done
US03	Inicio de Sesión	Como usuario, quiero iniciar sesión desde la landing page para gestionar mis reservas y perfil.	3	Castro Pariona, Jefferson Ernesto	Done
US04	Barra de Navegación	Como usuario, quiero una barra de navegación intuitiva y responsive para moverme fácilmente entre las secciones de la landing page.	2	Seminario Castillo, Diego Vicente	Done
US05	Footer y Redes Sociales	Como usuario, quiero un footer con enlaces de contacto y redes sociales para comunicarme o seguir novedades de WeRide.	2	Morales Sosa, Arnold Gabriel	Done
US06	Página de Información	Como usuario, quiero una sección informativa sobre WeRide y sus beneficios para entender el propósito de la plataforma.	2	Bonifacio Jaramillo, Samuel Jesus	Done
US07	Validación de Formularios	Como usuario, quiero que los formularios de registro e inicio de sesión validen mis datos para evitar errores y mejorar la seguridad.	2	Castro Pariona, Jefferson Ernesto	Done
US08	Estilos Generales	Como usuario, quiero una interfaz visualmente coherente (colores, fuentes, espaciados) para una experiencia agradable y moderna.	1	Romero Meza, Jhemy Pool	Done

5.2.1.4. Development Evidence for Sprint Review

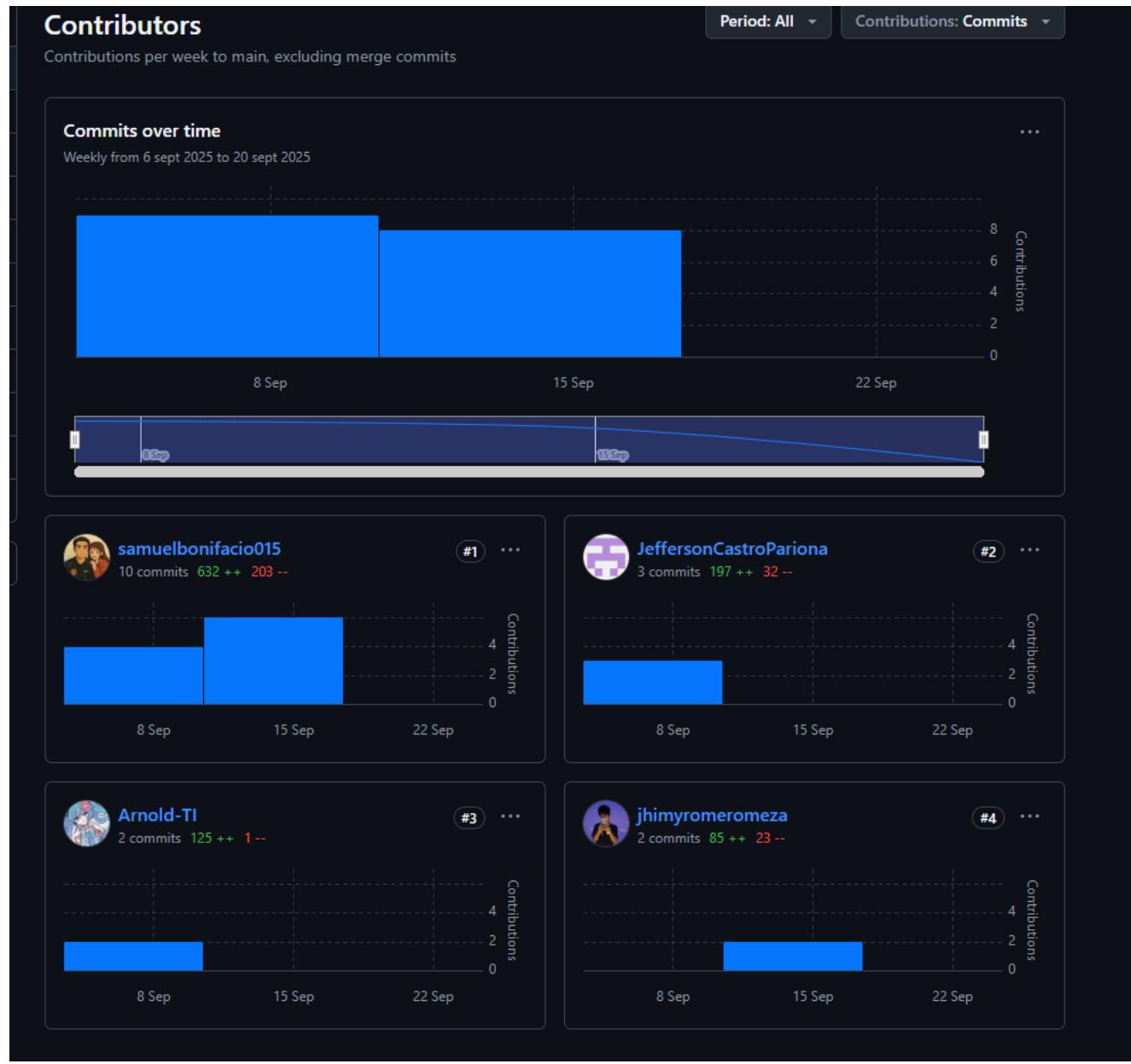
Durante el Sprint 1, el equipo de **WeTech** realizó múltiples commits en el repositorio de GitHub, evidenciando el avance y cumplimiento de las tareas planificadas. A continuación, se presenta una tabla con los principales commits realizados, siguiendo el formato requerido:

Repository	Branch	Commit Id	Commit Message	Commit Message Body	Committed on (Date)
OpenSource-Grupo-4/Landing-Page	develop	9c5de9c	Add footer styles to styles.css	Se agregaron estilos para el footer en styles.css.	13/09/2025

Repository	Branch	Commit Id	Commit Message	Commit Message Body	Committed on (Date)
OpenSource-Grupo-4/Landing-Page	develop	d0d0f0c	Add footer section with contact and social links	Se añadió la sección de footer con enlaces de contacto y redes sociales.	13/09/2025
OpenSource-Grupo-4/Landing-Page	develop	098b740	Merge pull request #2 from OpenSource-Grupo-4/develop	Se fusionaron cambios de la rama develop.	12/09/2025
OpenSource-Grupo-4/Landing-Page	feature/responsive-navbar	eddad5b	feat: added navbar & hero section	Se agregó la barra de navegación y la sección principal.	12/09/2025
OpenSource-Grupo-4/Landing-Page	develop	2a59cfe	chore: index file refactor	Refactorización del archivo index.	12/09/2025
OpenSource-Grupo-4/Landing-Page	main	69695c3	Merge pull request #1 from OpenSource-Grupo-4/develop	Se fusionaron cambios iniciales de la rama develop.	11/09/2025
OpenSource-Grupo-4/Landing-Page	hero-ui-fix	90dd977	/changes v.2 Jefferson Castro - Landing	Cambios en la landing page por Jefferson Castro.	11/09/2025
OpenSource-Grupo-4/Landing-Page	main	59a25be	feat: added steps section	Se agregó la sección de pasos.	18/09/2025
OpenSource-Grupo-4/Landing-Page	feature/responsive	4317f4c	styles: updated responsive for steps cards	Se mejoró la responsividad de las tarjetas de pasos.	18/09/2025
OpenSource-Grupo-4/Landing-Page	main	9deb5a8	feat: added steps section	Se agregó la sección de pasos.	18/09/2025
OpenSource-Grupo-4/Landing-Page	develop	6802c55	chore: added imgs & syntax updates	Se añadieron imágenes y se actualizaron detalles de sintaxis.	18/09/2025

Cada commit refleja el trabajo colaborativo y el cumplimiento de los objetivos del Sprint, asegurando la trazabilidad y transparencia en el desarrollo de **WeRide**.

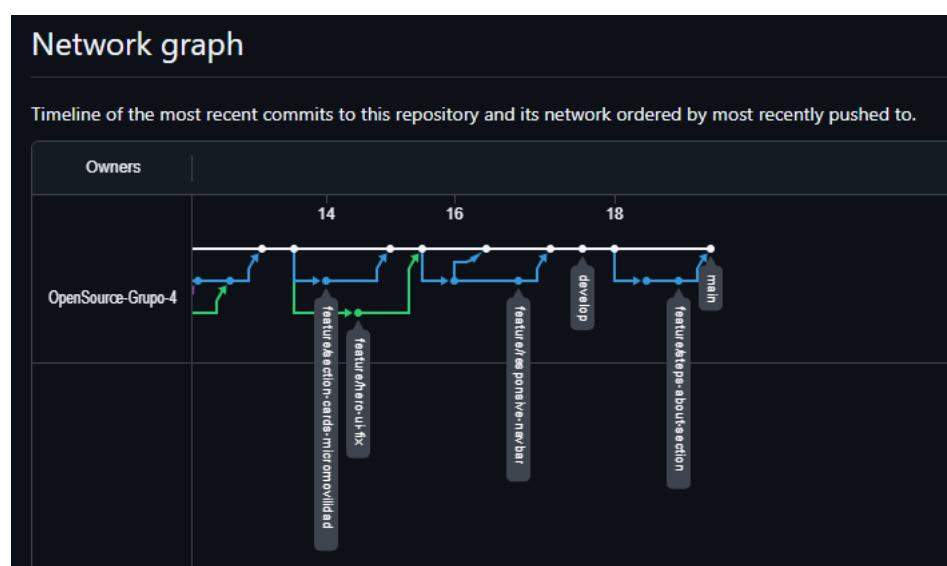
Reporte de commits:



5.2.1.5. Execution Evidence for Sprint Review.

En esta entrega, nos centramos en el desarrollo completo de la Landing Page para nuestra plataforma de educación en línea. Nuestro objetivo es desplegar una página web atractiva y efectiva que actúe como el punto de entrada para nuestros usuarios.

Es por ello que se comparte las evidencias de ambos repositorios para corroborar el trabajo hecho por cada uno de los participantes.



5.2.1.6. Services Documentation Evidence for Sprint Review.

Para esta primera entrega no hubo implementación de APIs, pues solo se realizó la Landing Page.

5.2.1.7. Software Deployment Evidence for Sprint Review.

El objetivo del primer Sprint fue el desarrollo e implementación de la Landing Page. Para eso, utilizamos GitHub y GitHub Pages. El proceso del desarrollo fue el siguiente:

1. Se creó una organización en GitHub y, dentro de ella, se crearon dos repositorios. Uno para el informe del trabajo y otro para la Landing Page

The screenshot shows the GitHub interface for the organization 'OpenSource-Grupo-4'. On the left, there's a sidebar with options like Overview, Repositories (which is selected), Projects, Packages, Teams, People, Insights, and Settings. The main area displays two repositories under the heading 'All': 'ReportTB1' (Public) and 'Landing-Page' (Public). Each repository has a card showing its name, type, last pushed date, and a link to its GitHub page.

2. Luego, a través de los commits, el equipo fue editando los archivos index.html y styles.css, además de ir añadiendo imágenes a utilizar en el directorio images.

The screenshot shows the commit history for the 'ReportTB1' repository. It lists several commits made on September 13, 2025, by different team members (samuelbonifacio015, Arnold-TI, JeffersonCastroPariona). The commits include changes to index.html, styles.css, and other files, along with the addition of images to the 'images' directory. Each commit includes a timestamp, author, commit message, SHA-1 hash, and a 'Verified' badge.

3. Finalmente, se implementó GitHub Pages en el repositorio de GitHub y se desplegó la Landing Page, la que se puede visitar en el siguiente link:

<https://opensource-grupo-4.github.io/Landing-Page>

5.2.1.8. Team Collaboration Insights during Sprint.

A continuación, se adjuntan las capturas de evidencia de los insights de los repositorios del informe y Landing Page para evidenciar la participación de todos los miembros:

- Insight Landing Page:

Pulse

Contributors

Community

Community Standards

Traffic

Commits

Code frequency

Dependency graph

Network

Forks

Actions Usage Metrics

Actions Performance Metrics

People

September 13, 2025 – September 20, 2025

Period: 1 week

Overview

5 Active pull requests 0 Active issues

Merged pull requests Open pull requests Closed issues New issues

Summary

Excluding merges, **2 authors** have pushed **8 commits** to main and **8 commits** to all branches.

On main, **12 files** have changed and there have been **344 additions** and **64 deletions**

Top Committers

Commits

5 pull requests merged by 2 people

- feat: added steps section
#10 merged 2 days ago
- feat: update button styles and structure in hero and vehicle sections
#9 merged 4 days ago
- chore: language switch to spanish
#8 merged 5 days ago
- style: fix hero design
#7 merged last week

- Insight Report

Pulse

Contributors

Community

Community Standards

Traffic

Commits

Code frequency

Dependency graph

Network

Forks

Actions Usage Metrics

Actions Performance Metrics

People

September 13, 2025 – September 20, 2025

Period: 1 week

Overview

6 Active pull requests 2 Active issues

Merged pull requests Open pull requests Closed issues New issues

Summary

Excluding merges, **4 authors** have pushed **28 commits** to main and **28 commits** to all branches.

On main, **41 files** have changed and there have been **1191 additions** and **13 deletions**

Top Committers

Commits

6 pull requests merged by 3 people

- Develop
#12 merged 43 minutes ago
- docs(diagram): add class diagram and upload documentation part
#10 merged 15 hours ago
- docs(diagram): add class diagram and upload documentation part
#9 merged yesterday
- feat: added user stories
#7 merged 2 days ago

5.2.2. Sprint 2

5.2.2.1. Sprint Planning 2

Elemento	Detalle
Sprint	2 - Implementación de base de datos local (db.json) y primera versión desplegado de WeRide.

Elemento	Detalle
Sprint Planning Background	
Fecha	8/10/2025
Hora	21:00 horas (GMT-5)
Lugar	Reunión virtual mediante Google Meet
Preparado por	[WeTech]
Asistentes	[Samuel Bonifacio , Jefferson Castro, Diego Seminario, Jhimy Romero, Arnold Morales]
Sprint n-2 Review	Se completó la primera versión desplegada del Frontend Web Application de WeRide.
Sprint n-2 Retrospective	Se destacó la buena coordinación del equipo y la claridad en las tareas asignadas; se acordó reforzar la integración del backend en el siguiente Sprint.
Sprint Goal & User Stories	1-22
Sprint 2 Velocity	20 puntos
Sum of Story Points	20 puntos

Objetivo del Sprint

Implementar el archivo `db.json` como base de datos simulada y desarrollar los bounded context clave de la aplicación web: **auth**, **booking** y **garage**, **plans** y **trip** mediante botones interactivos en la interfaz.

User Stories seleccionadas para el Sprint 2

ID	User Story	Puntos
01	Como usuario quiero poder iniciar sesión o registrarme en la app para usarla diariamente.	5
02	Como usuario, quiero introducir mi número de celular para validar mi identidad y recibir notificaciones importantes.	4
03	Como usuario, quiero introducir un código de verificación para validar mi identidad en la aplicación.	4
04	Como usuario quiero poder crear un perfil para colocar mis datos.	5
05	Como usuario quiero poder ver una pantalla principal estética que me atraiga a usar el servicio.	2

Total de puntos: 20

5.2.2.2. Aspect Leaders and Collaborators

Team Member (Apellido, Nombre)	GitHub Username	Base de Datos (db.json)	Registro de Usuario	Reserva de Vehículo	Pagos	Integración y Pruebas
Bonifacio, Samuel	samuelbonifacio015	L	L	L	L	L
Castro, Jefferson	JeffersonCastroPariona	L	L	L	L	L
Seminario, Diego	DiegoSeminario	C	C	L	C	C
Romero, Jhimy	jhimyromeromeza	L	C	C	C	L
Morales, Arnold	Arnold-TI	L	L	L	L	C

L: Leader (Líder)

C: Collaborator (Colaborador)

5.2.2.3. Sprint Backlog 2

ID	Title/Section	Description	Estimation (Hours)	Assigned To	Status
US21	Registro de Usuario	Como usuario, quiero registrar una cuenta desde la aplicación para acceder a los servicios de WeRide.	3	Castro Pariona, Jefferson Ernesto	Done
US22	Visualización de Vehículos	Como usuario, quiero visualizar los vehículos disponibles para elegir uno según mis necesidades.	4	Seminario Castillo, Diego Vicente	Done
US23	Reserva de Vehículo	Como usuario, quiero poder reservar un vehículo seleccionando fecha y hora de uso.	4	Romero Meza, Jhimy Pool	Done

ID	Title/Section	Description	Estimation (Hours)	Assigned To	Status
US24	Pago en Línea	Como usuario, quiero realizar el pago de mi reserva de manera segura mediante una interfaz simple y funcional.	5	Morales Sosa, Arnold Gabriel	Done
US25	Archivo db.json	Como desarrollador, quiero crear un archivo db.json con los datos de usuarios, vehículos y reservas para pruebas locales.	2	Bonifacio Jaramillo, Samuel Jesús	Done
US26	Pruebas e Integración	Como equipo, queremos realizar pruebas de funcionamiento e integración del flujo completo del sistema.	2	Todos	Done

5.2.2.4. Development Evidence for Sprint Review

Durante el Sprint 2 se implementaron los principales módulos de la aplicación web, incluyendo el componente de **garaje**, **mapa de viajes**, **historial de viajes**, **gestión de planes**, **integración de pagos** y **el archivo db.json**. Cada funcionalidad fue desarrollada y registrada en los siguientes commits y ramas, evidenciando la colaboración del equipo:

Repository	Branch	Commit Id	Commit Message	Autor	Committed on (Date)
OpenSource-Grupo-4/Landing-Page	samuel-1	caa30aa	feat: added garage component (pending features)	samuelbonifacio015	07/10/2025
OpenSource-Grupo-4/Landing-Page	samuel-1	33a6fef	feat: added trip-map	samuelbonifacio015	07/10/2025
OpenSource-Grupo-4/Landing-Page	samuel-1	c885f8f	feat: added db.json	samuelbonifacio015	07/10/2025
OpenSource-Grupo-4/Landing-Page	samuel-1	64615a8	feat: plan component added	samuelbonifacio015	07/10/2025
OpenSource-Grupo-4/Landing-Page	samuel-1	f078e3d	feat: added trip-history component	samuelbonifacio015	07/10/2025
OpenSource-Grupo-4/Landing-Page	jefferson	fbe7e9c	Implementation bounded-context booking	JeffersonCastroPariona	06/10/2025
OpenSource-Grupo-4/Landing-Page	diego	38ba86c	us7,8	DiegoSeminario	07/10/2025
OpenSource-Grupo-4/Landing-Page	feature/US-05-Map	3d5995a	feat(Map): add map component whit marker and location user	jhimyromeromeza	07/10/2025

Cada commit refleja el trabajo colaborativo y el cumplimiento de los objetivos del Sprint, asegurando la trazabilidad y transparencia en el desarrollo de **WeRide**.

Reporte de commits:

Frontend-WeRide Public

October 1, 2025 – October 8, 2025

Period: 1 week

Overview

- 1 Active pull requests
- 0 Active issues
- 1 Merged pull requests
- 0 Open pull requests
- 0 Closed issues
- 0 New issues

Summary

Excluding merges, 4 authors have pushed 2 commits to master and 25 commits to all branches.

On master, 28 files have changed and there have been 525 additions and 348 deletions.

Top Committers

1 pull request merged by 1 person

Samuel booking #1 merged 2 days ago

5.2.2.5. Execution Evidence for Sprint Review

En esta entrega, nos centramos en el desarrollo completo del desarrollo de la **Web Application** de WeRide.

Es por ello que se comparte las evidencias de ambos repositorios para corroborar el trabajo hecho por cada uno de los participantes.



5.2.2.6. Services Documentation Evidence for Sprint Review

El db.json fue creado para simular una base de datos local y facilitar el desarrollo y pruebas de la aplicación web. Este archivo contiene datos estructurados en formato JSON, representando las entidades principales del sistema: **usuarios, vehículos, reservas y planes**.

json server:

JSON Server

Docs ❤ Sponsor

⚡️ 🔒 🚀

/users - 3 items
 /companies - 3 items
 /vehicles - 10 items
 /plans - 3 items
 /locations - 10 items
 /bookings - 4 items
 /trips - 3 items
 /payments - 3 items
 /unlockRequests - 3 items
 /notifications - 5 items
 /favorites - 4 items

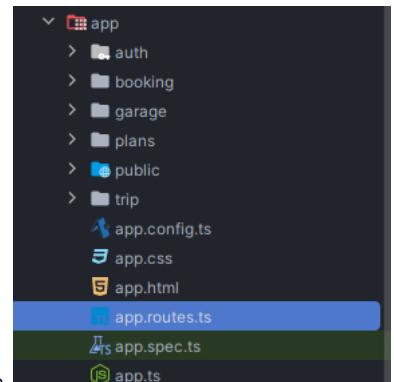
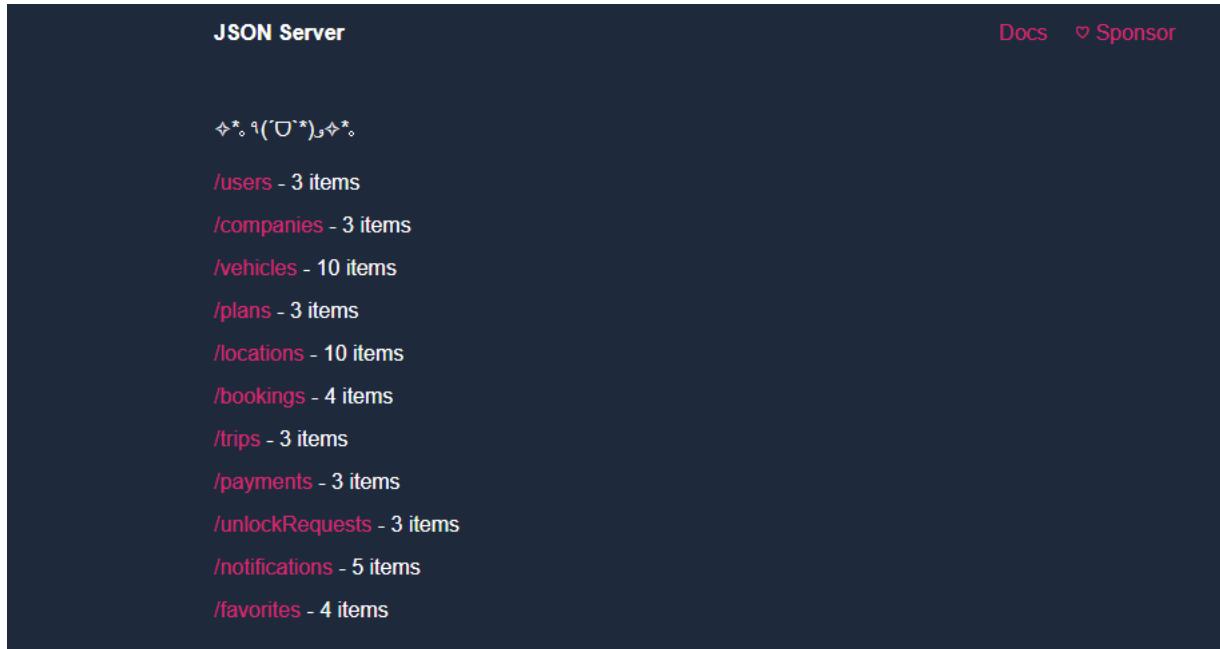
5.2.2.7. Software Deployment Evidence for Sprint Review

El objetivo del segundo Sprint fue el desarrollo e implementación de la Web Application. Para eso, utilizamos GitHub y GitHub Pages. El proceso del desarrollo fue el siguiente:

1. Se creó un repositorio para el desarrollo de la Web Application en GitHub.

2. Se creó el entorno de desarrollo con Node.js y Angular usando WebStorm como IDE.

3. Se desplegó el db.json con json-server para simular una base de datos local.



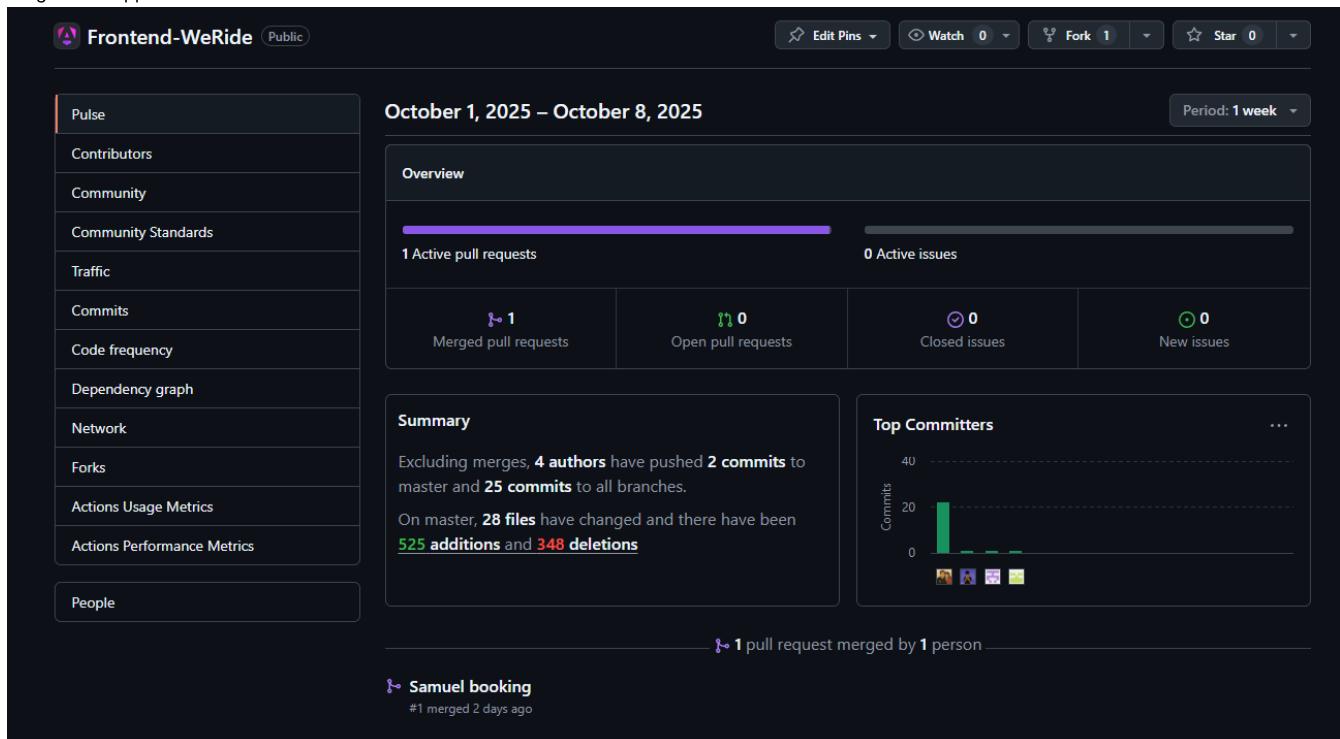
4. Se desarrollaron los componentes usando el patrón de DDD (Domain-Driven Design) para estructurar el código.

5. Se desplegó el frontend en github pages para pruebas y visualización.

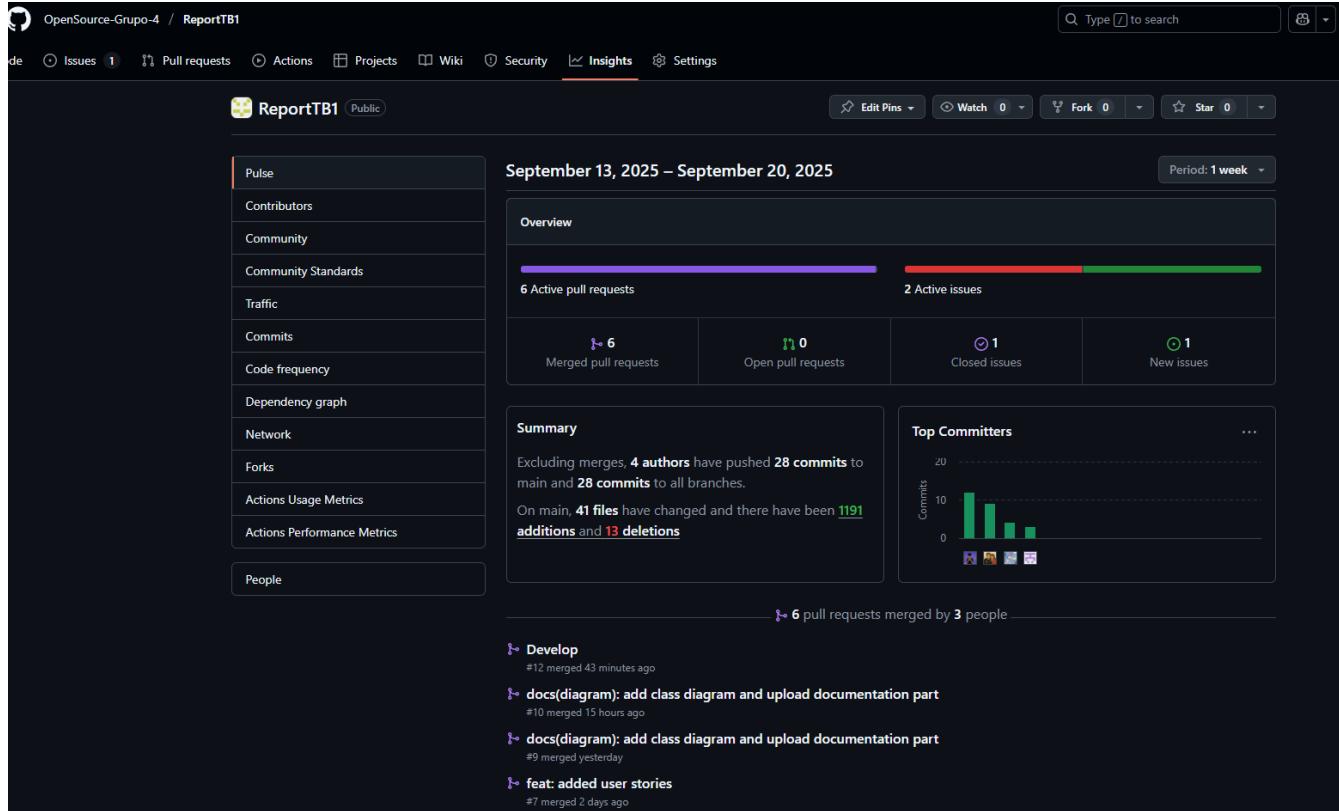
5.2.2.8. Team Collaboration Insights during Sprint

A continuación, se adjuntan las capturas de evidencia de los insights del repositorio del desarrollo de la Web Application para evidenciar la participación de todos los miembros:

- Insight Web Application:



- Insight Report



5.2.3. Sprint 3

5.2.3.1. Sprint Planning 3

Elemento	Detalle
Sprint	3 - Primera versión de backend y pruebas de conexión con MySQL Workbench
Sprint Planning Background	
Fecha	27/10/2025
Hora	20:00 horas (GMT-5)
Lugar	Reunión virtual mediante Google Meet
Preparado por	[WeTech]
Asistentes	[Samuel Bonifacio , Jefferson Castro, Diego Seminario, Arnold Morales]
Sprint n-3 Review	Se asignaron los bounded context para el backend y se inicializó con las primeras contribuciones en el repositorio.
Sprint n-3 Retrospective	Se destacó la buena coordinación del equipo y la claridad en las tareas asignadas.
Sprint Goal & User Stories	1-22
Sprint 3 Velocity	20 puntos
Sum of Story Points	20 puntos

Objetivo del Sprint

Implementar la primera versión del backend utilizando Java(Springboot) y establecer la conexión con una base de datos MySQL Workbench para gestionar usuarios, vehículos y reservas.

User Stories seleccionadas para el Sprint 3

ID	User Story	Puntos
01	Como usuario quiero poder iniciar sesión o registrarme en la app para usarla diariamente.	5
02	Como usuario, quiero introducir mi número de celular para validar mi identidad y recibir notificaciones importantes.	4
03	Como usuario, quiero introducir un código de verificación para validar mi identidad en la aplicación.	4
04	Como usuario quiero poder crear un perfil para colocar mis datos.	5
05	Como usuario quiero poder ver una pantalla principal estética que me atraiga a usar el servicio.	2

Total de puntos: 20

5.2.3.2. Aspect Leaders and Collaborators

Team Member (Apellido, Nombre)	GitHub Username	Deploy de Base de Datos	Historial de Viajes	Gestión de Perfiles	Reservas	Garage
Bonifacio, Samuel	samuelbonifacio015	L	L	L	L	L
Castro, Jefferson	JeffersonCastroPariona	L	C	L	L	C
Seminario, Diego	DiegoSeminario	L	C	L	C	L
Romero, Jhimi	Jhimiromeromeza	L	C	C	C	C
Morales, Arnold	Arnold-TI	L	L	L	L	C

L: Leader (Líder)

C: Collaborator (Colaborador)

5.2.3.3. Sprint Backlog 3

ID	Title/Section	Description	Estimation (Hours)	Assigned To	Status
US21	Registro de Usuario	Como usuario, quiero registrar una cuenta desde la aplicación para acceder a los servicios de WeRide.	3	Castro Pariona, Jefferson Ernesto	Done
US22	Visualización de Vehículos	Como usuario, quiero visualizar los vehículos disponibles para elegir uno según mis necesidades.	4	Seminario Castillo, Diego Vicente	Done
US23	Reserva de Vehículo	Como usuario, quiero poder reservar un vehículo seleccionando fecha y hora de uso.	4	Romero Meza, Jhimi Pool	Done
US24	Pago en Línea	Como usuario, quiero realizar el pago de mi reserva de manera segura mediante una interfaz simple y funcional.	5	Morales Sosa, Arnold Gabriel	Done
US25	Archivo db.json	Como desarrollador, quiero crear un archivo db.json con los datos de usuarios, vehículos y reservas para pruebas locales.	2	Bonifacio Jaramillo, Samuel Jesús	Done
US26	Pruebas e Integración	Como equipo, queremos realizar pruebas de funcionamiento e integración del flujo completo del sistema.	2	Todos	Done

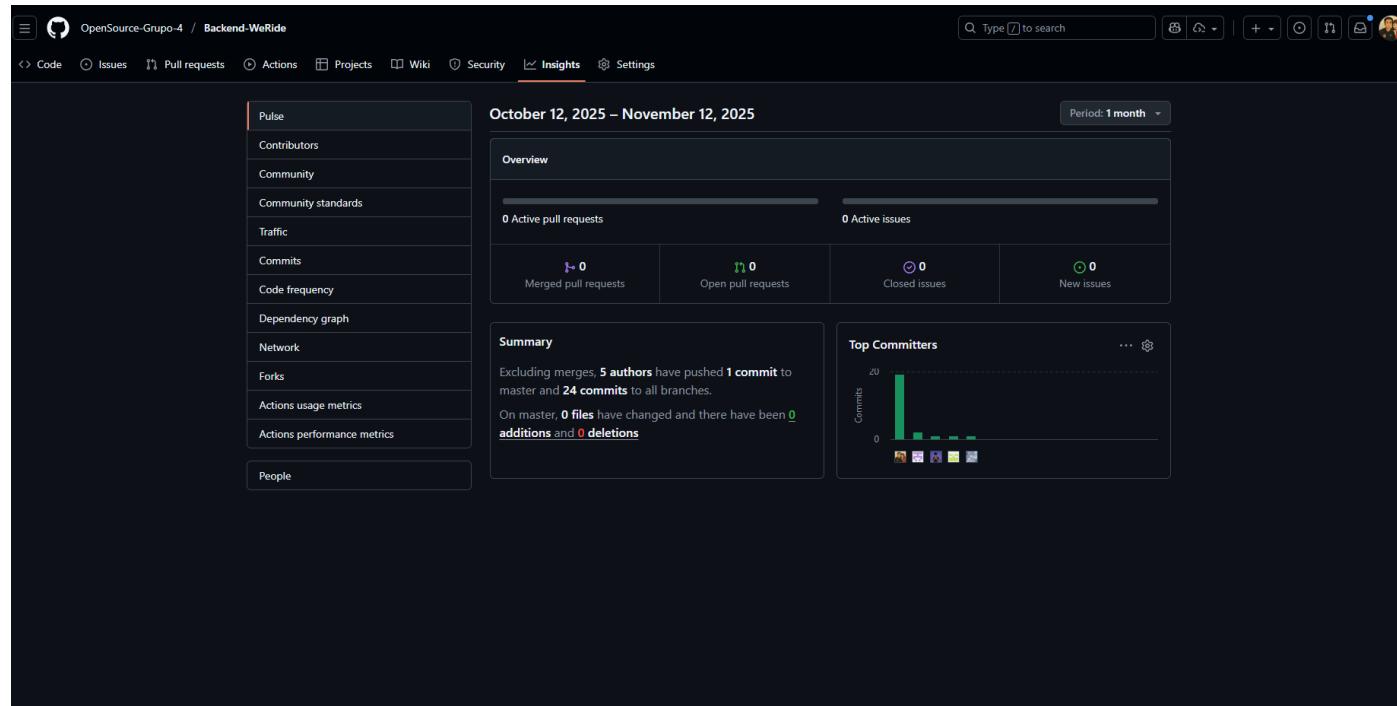
5.2.3.4. Development Evidence for Sprint Review

Durante el Sprint 3 se implementaron los principales módulos del backend, incluyendo el componente de **gestión de perfiles, reservas, historial de viajes y garage de autos**. A continuación se listan los commits más relevantes registrados en el repositorio del backend durante este Sprint (captura en el panel de commits):

Repository	Branch	Commit Id	Commit Message	Autor	Committed on (Date)
OpenSource-Grupo-4/Backend-WeRide	feat/travelHistory	f7222d1	feat(travelHistory): Add news endpoints for create TravelHistory and find TravelHistory by userId	jhimiromeromeza	09/11/2025
OpenSource-Grupo-4/Backend-WeRide	feature/profile	6e8bd4f	feat: Implement Profile Bounded Context	Arnold-TI	08/11/2025
OpenSource-Grupo-4/Backend-WeRide	main	69c362f	feat: implement JWT auth	samuelbonifacio015	29/10/2025
OpenSource-Grupo-4/Backend-WeRide	main	e733cf1	feat: added rest interfaces for iam	samuelbonifacio015	29/10/2025
OpenSource-Grupo-4/Backend-WeRide	main	7c22d3e	feat: added iam infrastructure	samuelbonifacio015	29/10/2025
OpenSource-Grupo-4/Backend-WeRide	feat/booking	e3c8283	feat(booking): update classes logic validation	JeffersonCastroPariona	29/10/2025
OpenSource-Grupo-4/Backend-WeRide	feat/booking	a60ea90	feat(booking): implement aggregate and commands queries services	JeffersonCastroPariona	28/10/2025
OpenSource-Grupo-4/Backend-WeRide	bc/garage	5875d27	bc/garage	DiegoSeminario	11/11/2025

Cada commit refleja el trabajo colaborativo y el cumplimiento de los objetivos del Sprint, asegurando la trazabilidad y transparencia en el desarrollo de WeRide.

Reporte de commits (captura):



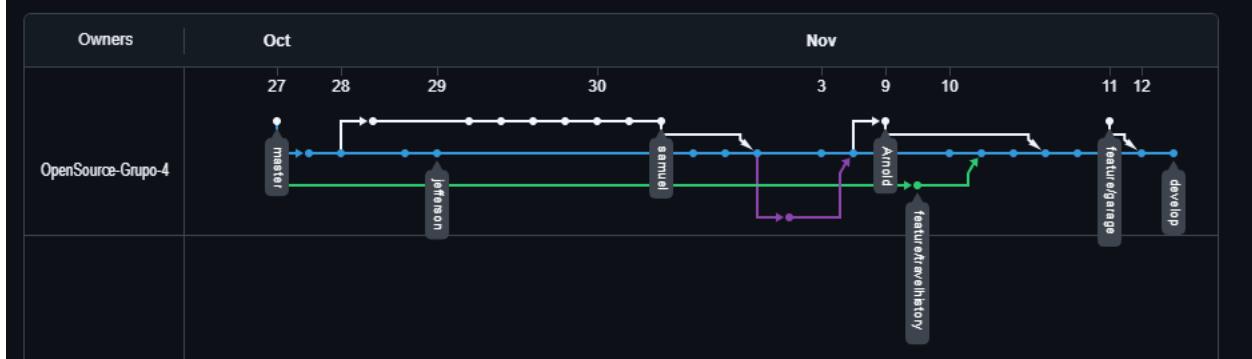
5.2.3.5. Execution Evidence for Sprint Review

En esta entrega, nos centramos en el desarrollo completo del desarrollo de los **Web Services** de WeRide.

Es por ello que se comparte las evidencias de ambos repositorios para corroborar el trabajo hecho por cada uno de los participantes.

Network graph

Timeline of the most recent commits to this repository and its network ordered by most recently pushed to.



5.2.3.6. Services Documentation Evidence for Sprint Review

Para el backend, hemos desarrollado nuestro API utilizando Java con el framework Spring Boot. Este backend se conecta a una base de datos MySQL Workbench para gestionar el inicio de sesión, gestión de perfiles, reservas, historial de viajes y garage de autos.

Para verificar que la obtención de datos se realice correctamente, utilizamos Swagger UI, una herramienta que nos permite interactuar con nuestra API de manera visual y probar los diferentes endpoints que hemos creado.

Schemas:

```

UpdateProfileResource {
    name
    phone
    profilePicture
    dateOfBirth
    address
    emergencyContact
}
  
```

```
ProfileResource ▼ {  
    id          ▼ integer($int64)  
    accountId  ▼ integer($int64)  
    name        ▼ string  
    phone       ▼ string  
    profilePicture ▼ string  
    dateOfBirth ▼ string($date)  
    address     ▼ string  
    emergencyContact ▼ string  
    language    ▼ string  
    notifications ▼ boolean  
    theme       ▼ string  
}
```

```
CreateBookingResource ▼ {  
    id          ▼ string  
    customerId ▼ string  
    vehicleType ▼ string  
    date        ▼ string($date)  
    unlockTime  LocalTime ▼ {  
        hour      ▼ integer($int32)  
        minute    ▼ integer($int32)  
        second   ▼ integer($int32)  
        nano     ▼ integer($int32)  
    }  
    durationMinutes ▼ integer($int32)  
}
```

```
LocalTime ▼ {  
    hour      ▼ integer($int32)  
    minute    ▼ integer($int32)  
    second   ▼ integer($int32)  
    nano     ▼ integer($int32)  
}
```

```
BookingResource ▼ {  
    id          ▼ string  
    customerId ▼ string  
    vehicleId  ▼ string  
    vehicleBrand ▼ string  
    vehicleModel ▼ string  
    date        ▼ string($date)  
    unlockTime  LocalTime ▼ {  
        hour      ▼ integer($int32)  
        minute    ▼ integer($int32)  
        second   ▼ integer($int32)  
        nano     ▼ integer($int32)  
    }  
    durationMinutes ▼ integer($int32)  
    pricePerMinute ▼ number  
    totalPrice   ▼ number  
    status       ▼ string  
}
```

```
SignUpResource ▼ {  
    username  ▼ string  
    password  ▼ string  
}
```

```
SignInResource ▼ {  
    username  ▼ string  
    password  ▼ string  
}
```

```
AccountResource ▼ {  
    id          ▼ integer($int64)  
    username  ▼ string  
}
```

```
AuthenticatedAccountResource ▼ {  
    id          ▼ integer($int64)  
    token      ▼ string  
}
```

```
CreateTravelHistoryResource ▼ {  
    userId     ▼ integer($int64)  
    location   ▼ string  
    vehicle    ▼ string  
    image      ▼ string  
    tripDuration ▼ string  
    travelDistance ▼ string  
}
```

```
User ▼ {  
    id          ▼ integer($int64)  
    username   ▼ string  
    email      ▼ string  
    password   ▼ string  
}
```

```
RegisterUserCommand ▼ {  
    username   ▼ string  
    email      ▼ string  
    password   ▼ string  
}
```

```

PageBookingResource √ {
    totalElements √ integer($int64)
    totalPages √ integer($int32)
    size √ integer($int32)
    content √ [BookingResource √ {
        id √ string
        customerId √ string
        vehicleId √ string
        vehicleBrand √ string
        vehicleModel √ string
        date √ string($date)
        unlockTime √ LocalTime √ {
            hour √ integer($int32)
            minute √ integer($int32)
            second √ integer($int32)
            nano √ integer($int32)
        }
        durationMinutes √ integer($int32)
        pricePerMinute √ number
        totalPrice √ number
        status √ string
    }]
    number √ integer($int32)
    first √ boolean
    last √ boolean
    numberOfElements √ integer($int32)
    sort √ [SortObject > {...}]
    pageable √ PageableObject √ {
        offset √ integer($int64)
        sort √ [SortObject √ {
            direction √ string
            nullHandling √ string
            ascending √ boolean
            property √ string
            ignoreCase √ boolean
        }]
        paged √ boolean
        pageNumber √ integer($int32)
        pageSize √ integer($int32)
        unpaged √ boolean
    }
    empty √ boolean
}

```

```

PageableObject √ {
    offset √ integer($int64)
    sort √ [SortObject √ {
        direction √ string
        nullHandling √ string
        ascending √ boolean
        property √ string
        ignoreCase √ boolean
    }]
    paged √ boolean
    pageNumber √ integer($int32)
    pageSize √ integer($int32)
    unpaged √ boolean
}

```

```

SortObject √ {
    direction √ string
    nullHandling √ string
    ascending √ boolean
    property √ string
    ignoreCase √ boolean
}

```

```

TravelHistoryResource √ {
    id √ integer($int64)
    userId √ integer($int64)
    location √ string
    vehicle √ string
    image √ string
    tripDuration √ string
    travelDistance √ string
    createdAt √ string($date-time)
}

```

```

Reservation <-
  id           <-- integer($int64)
  startDate    <-- string($date-time)
  endDate      <-- string($date-time)
  user         <-- User <-
    id           <-- integer($int64)
    username    <-- string
    email       <-- string
    password    <-- string
  vehicle      <-- Vehicle <-
    id           <-- integer($int64)
    name         <-- string
    brand        <-- string
    price        <-- number($double)
    rating       <-- number($double)
    available   <-- boolean
    favorite    <-- boolean
    imageUrl    <-- string
}

```

Swagger:

Accounts

Accounts Accounts Endpoints ^

GET /api/v1/accounts/{accountId} ▾

Parameters Try it out

Name	Description
accountid <small>* required</small> integer(\$int64) (path)	accountid

Responses

Code	Description	Links
200	OK	No links

Media type
"json" ▾
Controls Accept header.

Example Value | Schema

```
{
  "id": 0,
  "username": "string"
}
```

Authentication

Authentication Authentication Endpoints ^

POST /api/v1/authentication/sign-up ▾

POST /api/v1/authentication/sign-in ▾

SignUp:

POST /api/v1/authentication/sign-up

Parameters

No parameters

Request body required

application/json

Example Value | Schema

```
{ "username": "string", "password": "string" }
```

Responses

Code	Description	Links
200	OK	No links

Media type

application/json

Controls Accept header.

Example Value | Schema

```
{ "id": 0, "username": "string" }
```

Signin:

POST /api/v1/authentication/sign-in

Parameters

No parameters

Request body required

application/json

Example Value | Schema

```
{ "username": "string", "password": "string" }
```

Responses

Code	Description	Links
200	OK	No links

Media type

application/json

Controls Accept header.

Example Value | Schema

```
{ "id": 0, "token": "string" }
```

Auth:

POST /api/auth/login Login de usuario ^

Valida credenciales y devuelve un JWT para acceder a los endpoints protegidos

Parameters

No parameters

Request body required

application/json

Example Value | Schema

```
{ "id": 0, "username": "string", "email": "string", "password": "string" }
```

Responses

Code	Description	Links
200	Login exitoso, devuelve JWT	No links
	Media type	
	/	
	Controls Accept header.	
	Example Value Schema	
	string	
401	Credenciales inválidas	No links
	Media type	
	/	
	Example Value Schema	
	string	

Auth Autenticación de usuarios ^

POST /api/auth/register ^

Parameters

No parameters

Request body required

application/json

Example Value | Schema

```
{ "username": "string", "email": "string", "password": "string" }
```

Responses

Code	Description	Links
200	OK	No links
	Media type	
	/	
	Controls Accept header.	
	Example Value Schema	
	{ "id": 0, "username": "string", "email": "string", "password": "string" }	

Reservations:

Reservations Gestión de reservas de vehículos ^

GET /api/reservations

Parameters

No parameters

Try it out

Responses

Code	Description	Links
200	OK	No links

Media type `*/*` ▾
Controls Accept header.

Example Value | Schema

```
[ {
  "id": 0,
  "startDate": "2025-11-12T22:07:45.743Z",
  "endDate": "2025-11-12T22:07:45.743Z",
  "user": {
    "id": 0,
    "username": "string",
    "email": "string",
    "password": "string"
  },
  "vehicle": {
    "id": 0,
    "name": "string",
    "brand": "string",
    "price": 0,
    "rating": 0,
    "available": true,
    "favorite": true,
    "imageUrl": "string"
  }
}]
```

POST /api/reservations Crear reserva ^

Crea una nueva reserva para un vehículo

Parameters

No parameters

Request body required

Try it out

application/json

Example Value | Schema

```
{ "id": 0,
  "startDate": "2025-11-12T22:08:22.752Z",
  "endDate": "2025-11-12T22:08:22.752Z",
  "user": {
    "id": 0,
    "username": "string",
    "email": "string",
    "password": "string"
  },
  "vehicle": {
    "id": 0,
    "name": "string",
    "brand": "string",
    "price": 0,
    "rating": 0,
    "available": true,
    "favorite": true,
    "imageUrl": "string"
  }
}
```

Responses

Code	Description	Links
201	Reserva creada exitosamente	No links

Media type `*/*` ▾
Controls Accept header.

Example Value | Schema

```
{ "id": 0,
  "startDate": "2025-11-12T22:08:22.755Z",
  "endDate": "2025-11-12T22:08:22.755Z",
  "user": {
    "id": 0,
    "username": "string",
    "email": "string",
    "password": "string"
  },
  "vehicle": {
    "id": 0,
    "name": "string",
    "brand": "string",
    "price": 0
  }
}
```

"rating": 0,
 "available": true,
 "favorite": true,
 "imageUrl": "string"
 }
 }

Responses

Code	Description	Links
201	Reserva creada exitosamente	No links
	Media type <input type="button" value="/*"/> <small>Controls Accept header.</small> Example Value Schema	
	{ "id": 0, "startDate": "2025-11-12T22:08:22.755Z", "endDate": "2025-11-12T22:08:22.755Z", "user": { "id": 0, "username": "string", "email": "string", "password": "string" }, "vehicle": { "id": 0, "name": "string", "brand": "string", "price": 0, "rating": 0, "available": true, "favorite": true, "imageUrl": "string" } }	
400	Datos inválidos	No links
	Media type <input type="button" value="/*"/> <small>Controls Accept header.</small> Example Value Schema	
	{ "id": 0, "startDate": "2025-11-12T22:08:22.755Z", "endDate": "2025-11-12T22:08:22.755Z", "user": { "id": 0, "username": "string", "email": "string", "password": "string" }, "vehicle": { "id": 0, "name": "string", "brand": "string", "price": 0, "rating": 0, "available": true, "favorite": true, "imageUrl": "string" } }	

Profiles:

Profiles Endpoints para gestionar Perfiles de Usuario

PUT /api/v1/profiles/{profileId}

GET /api/v1/profiles/by-account-id/{accountId}

Profiles Endpoints para gestionar Perfiles de Usuario

PUT /api/v1/profiles/{profileId}

Parameters

Name	Description
profileId <small>* required</small> integer(\$int64) (path)	profileId

Request body required

application/json

Example Value | Schema

```
{
  "name": "string",
  "phone": "string",
  "profilePicture": "string",
  "dateOfBirth": "2025-11-12",
  "address": "string",
  "emergencyContact": "string"
}
```

Responses

Code	Description	Links
200	OK	No links

Media type

/

Controls Accept header.

Example Value | Schema

```
{
  "id": 0,
  "accountId": 0,
  "name": "string",
  "phone": "string",
  "profilePicture": "string",
  "dateOfBirth": "2025-11-12",
  "address": "string",
  "emergencyContact": "string",
  "language": "string",
  "notifications": true,
  "theme": "string"
}
```

GET /api/v1/profiles/by-account-id/{accountId}

Parameters

Name	Description
accountId <small>* required</small> integer(\$int64) (path)	accountId

Responses

Code	Description	Links
200	OK	No links

Media type

/

Controls Accept header.

Example Value | Schema

```
{
  "id": 0,
  "accountId": 0,
  "name": "string",
  "phone": "string",
  "profilePicture": "string",
  "dateOfBirth": "2025-11-12",
  "address": "string",
  "emergencyContact": "string",
  "language": "string",
  "notifications": true,
  "theme": "string"
}
```

Bookings:

booking-controller

GET	/bookings	^
POST	/bookings	^
POST	/bookings/draft	^
GET	/bookings/{id}	^
GET	/bookings/drafts	^

GET /bookings

Parameters

Name	Description
customerId	customerId
vehicleId	vehicleId
status	status
startAtFrom	startAtFrom
startAtTo	startAtTo
page	Default value : 0
size	Default value : 20

Responses

Code	Description	Links
200	OK	No links

Media type: application/json
Content Accept header:

Example Value | Schema

```
{
  "totalElements": 0,
  "totalPages": 0,
  "size": 0,
  "content": [
    {
      "id": "string",
      "customerId": "string",
      "vehicleId": "string",
      "vehicleModel": "string",
      "date": "2025-11-12T00:00:00Z",
      "durationTime": {
        "hours": 0,
        "minutes": 0,
        "seconds": 0,
        "name": 0
      },
      "durationMinutes": 0,
      "pricePerMinute": 0,
      "totalPrice": 0,
      "status": "string"
    }
  ]
}
```

POST /bookings

Parameters

No parameters

Request body **required**

application/json

Example Value | Schema

```
{
  "id": "string",
  "customerId": "string",
  "vehicleType": "string",
  "date": "2025-11-12",
  "unlockTime": {
    "hour": 0,
    "minute": 0,
    "second": 0,
    "nano": 0
  },
  "durationMinutes": 0
}
```

Responses

Code	Description	Links
200	OK	No links

Media type

/

Controls Accept header.

Example Value | Schema

```
{
  "id": "string",
  "customerId": "string",
  "vehicleId": "string",
  "vehicleBrand": "string",
  "vehicleModel": "string",
  "date": "2025-11-12",
  "unlockTime": {
    "hour": 0,
    "minute": 0,
    "second": 0,
    "nano": 0
  },
  "durationMinutes": 0,
  "pricePerMinute": 0,
  "totalPrice": 0,
  "status": "string"
}
```

POST /bookings/draft

Parameters

No parameters

Request body **required**

application/json

Example Value | Schema

```
{
  "id": "string",
  "customerId": "string",
  "vehicleType": "string",
  "date": "2025-11-12",
  "unlockTime": {
    "hour": 0,
    "minute": 0,
    "second": 0,
    "nano": 0
  },
  "durationMinutes": 0
}
```

Responses

Code	Description	Links
200	OK	No links

Media type

/

Controls Accept header.

Example Value | Schema

```
{
  "id": "string",
  "customerId": "string",
  "vehicleId": "string",
  "vehicleBrand": "string",
  "vehicleModel": "string",
  "date": "2025-11-12",
  "unlockTime": {
    "hour": 0,
    "minute": 0,
    "second": 0,
    "nano": 0
  },
  "durationMinutes": 0,
  "pricePerMinute": 0,
  "totalPrice": 0,
  "status": "string"
}
```

GET /bookings/{id}

Parameters

Name	Description
id * required	string (path)

Responses

Code	Description	Links
200	OK	No links

Media type
/

Controls Accept header.

Example Value | Schema

```
{
  "id": "string",
  "customerId": "string",
  "vehicleId": "string",
  "vehicleBrand": "string",
  "vehicleModel": "string",
  "date": "2025-11-12",
  "unlockTime": {
    "hour": 0,
    "minute": 0,
    "second": 0,
    "nano": 0
  },
  "durationMinutes": 0,
  "pricePerMinute": 0,
  "totalPrice": 0,
  "status": "string"
}
```

GET /bookings/drafts

Parameters

Name	Description
customerId * required	string (query)
page	Default value : 0
integer(\$int32) (query)	0
size	Default value : 20
integer(\$int32) (query)	20

Responses

Code	Description	Links
200	OK	No links

Media type
/

Controls Accept header.

Example Value | Schema

```
{
  "totalElements": 0,
  "totalPages": 0,
  "size": 0,
  "content": [
    {
      "id": "string",
      "customerId": "string",
      "vehicleId": "string",
      "vehicleBrand": "string",
      "vehicleModel": "string",
      "date": "2025-11-12",
      "unlockTime": {
        "hour": 0,
        "minute": 0,
        "second": 0,
        "nano": 0
      },
      "durationMinutes": 0,
      "pricePerMinute": 0,
      "totalPrice": 0,
      "status": "string"
    }
  ],
  "number": 0,
  "first": true,
  "last": true,
  "numberOfElements": 0,
  "pageSize": 0
}
```

Vehicles:

GET /api/vehicles

Parameters

No parameters

Responses

Code	Description	Links
200	OK	No links

Media type

application/json

Controls Accept header.

Example Value | Schema

```
[{"id": 0, "name": "string", "brand": "string", "price": 0, "rating": 0, "available": true, "favorite": true, "imageUrl": "string"}]
```

POST /api/vehicles/{id}/favorite

Parameters

Name	Description
id * required	integer(\$int64) (path)

Responses

Code	Description	Links
200	OK	No links

Media type

application/json

Controls Accept header.

Example Value | Schema

```
{"id": 0, "name": "string", "brand": "string", "price": 0, "rating": 0, "available": true, "favorite": true, "imageUrl": "string"}
```

Travel History:

travel-history-controller

POST /api/travel-history

GET /api/travel-history/{userId}

GET /api/travel-history/{userId}

Parameters

Name	Description
userId <small>* required</small>	integer(\$int64) (path)

Responses

Code	Description	Links
200	OK	No links

Media type

/*
Controls Accept header.

Example Value | Schema

```
[ { "id": 0, "userId": 0, "location": "string", "vehicle": "string", "image": "string", "tripDuration": "string", "travelDistance": "string", "createdAt": "2025-11-12T22:15:31.708Z" } ]
```

travel-history-controller

POST /api/travel-history

Parameters

No parameters	
---------------	--

Request body required

application/json

Example Value | Schema

```
{ "userId": 0, "location": "string", "vehicle": "string", "image": "string", "tripDuration": "string", "travelDistance": "string" }
```

Responses

Code	Description	Links
200	OK	No links

Media type

/*
Controls Accept header.

Example Value | Schema

```
{ "id": 0, "userId": 0, "location": "string", "vehicle": "string", "image": "string", "tripDuration": "string", "travelDistance": "string", "createdAt": "2025-11-12T22:15:16.165Z" }
```

Users:

user-controller

GET /api/users

Parameters

No parameters

Responses

Code	Description	Links
200	OK	No links

Media type

/

Controls Accept header.

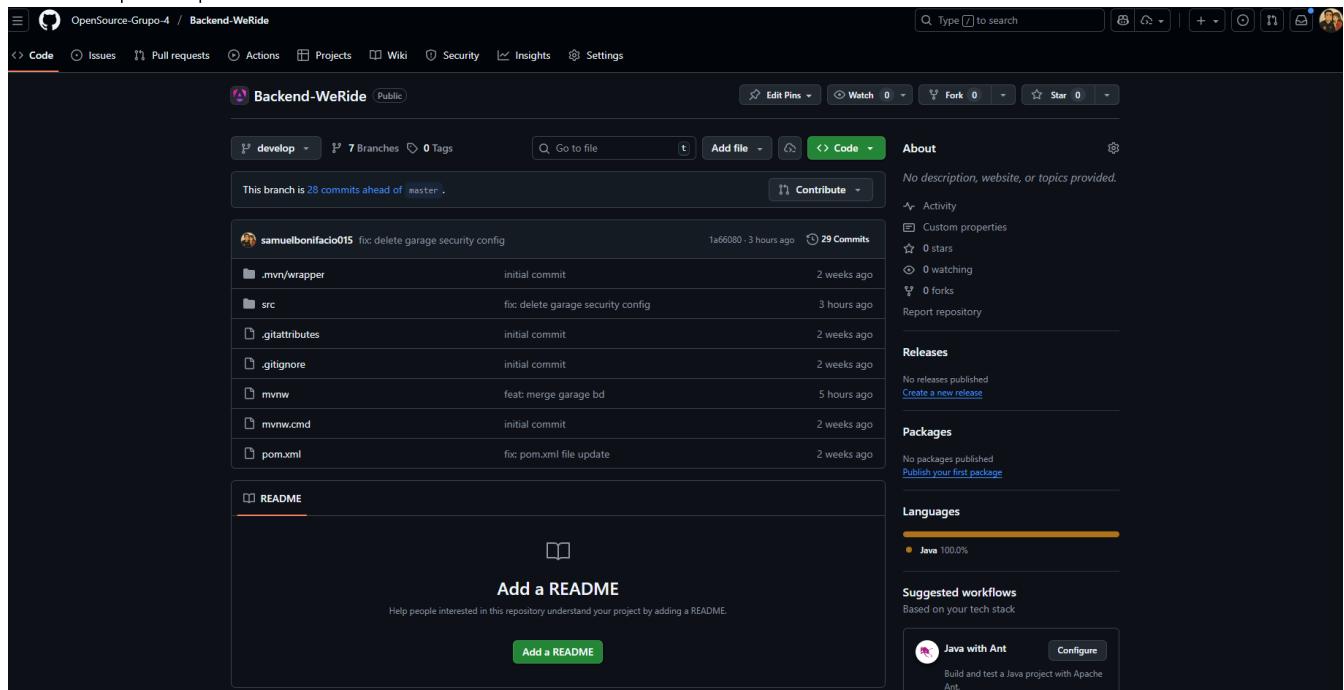
Example Value | Schema

```
[ {
  "id": 0,
  "username": "string",
  "email": "string",
  "password": "string"
}]
```

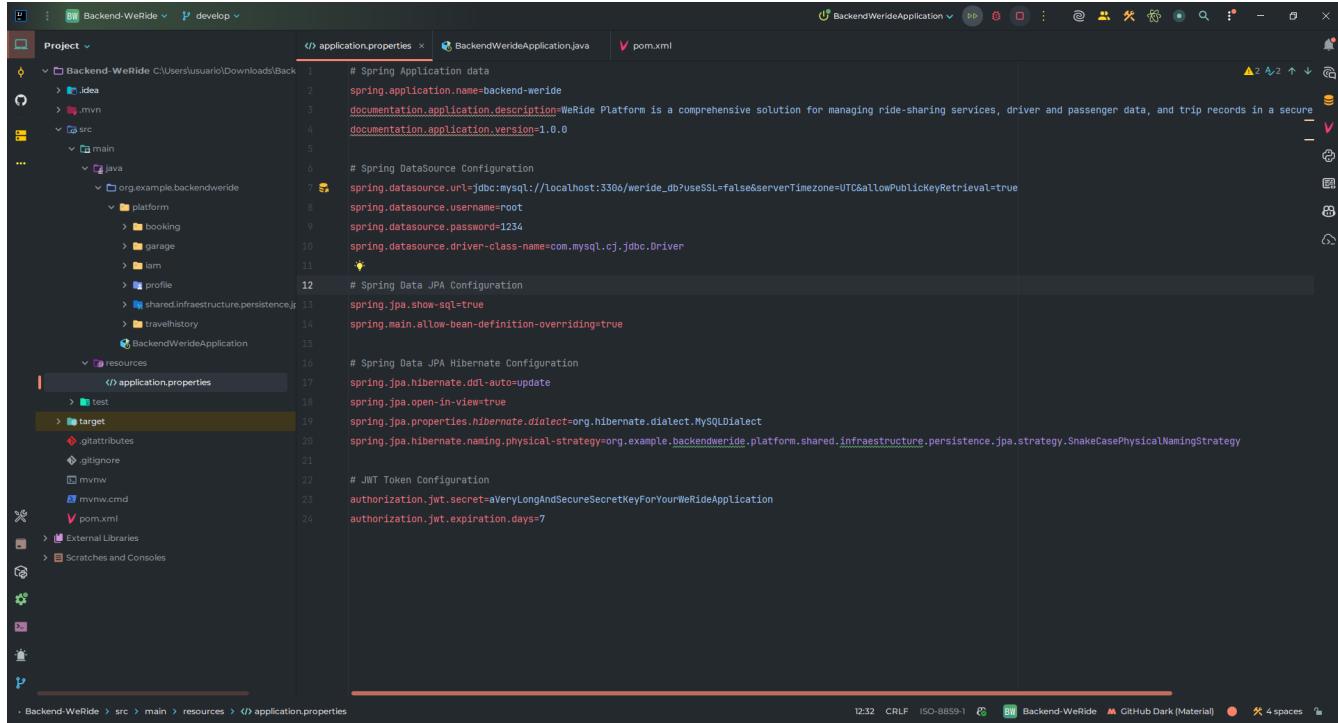
5.2.3.7. Software Deployment Evidence for Sprint Review

El objetivo del tercer Sprint fue el desarrollo e implementación del Backend de la Web Application. Para eso, utilizamos GitHub. El proceso del desarrollo fue el siguiente:

1. Se creó un repositorio para el desarrollo del Backend en GitHub.



2. Se creó el entorno de desarrollo con Java(Springboot) y MySQL Workbench usando IntelliJ IDEA como IDE.



```

# Spring Application data
spring.application.name=backend-weride
documentation.application.description=WeRide Platform is a comprehensive solution for managing ride-sharing services, driver and passenger data, and trip records in a secure
documentation.application.version=1.0.0

# Spring DataSource Configuration
spring.datasource.url=jdbc:mysql://localhost:3306/weride_db?useSSL=false&serverTimezone=UTC&allowPublicKeyRetrieval=true
spring.datasource.username=root
spring.datasource.password=1234
spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver

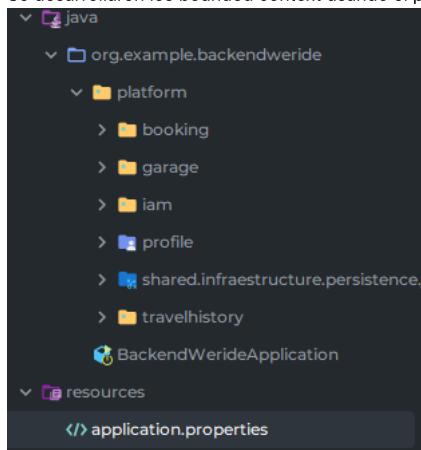
# Spring Data JPA Configuration
spring.jpa.show-sql=true
spring.main.allow-bean-definition-overriding=true

# Spring Data JPA Hibernate Configuration
spring.jpa.hibernate.ddl-auto=update
spring.jpa.open-in-view=true
spring.jpa.properties.hibernate.dialect=org.hibernate.dialect.MySQLDialect
spring.jpa.hibernate.naming.physical-strategy=org.example.backendweride.platform.shared.infraestructure.persistence.jpa.strategy.SnakeCasePhysicalNamingStrategy

# JWT Token Configuration
authorization.jwt.secret=aVeryLongAndSecureSecretKeyForYourWeRideApplication
authorization.jwt.expiration.days=7

```

3. Se desarrollaron los bounded context usando el patrón de DDD (Domain-Driven Design) para estructurar el código.



4. Se utilizó Swagger UI para documentar y probar los endpoints del API desarrollado.

5. Se desplegó el backend en un servidor local para pruebas y visualización.

5.2.3.8. Team Collaboration Insights during Sprint

A continuación, se adjuntan las capturas de evidencia de los insights del repositorio del desarrollo de los Web Services para evidenciar la participación de todos los miembros:

- Insight Web Service:

The screenshot shows the GitHub Insights interface for the repository "OpenSource-Grupa-4 / Backend-WeRide". The main header bar includes links for Code, Issues, Pull requests, Actions, Projects, Wiki, Security, Insights (which is highlighted), and Settings. A search bar at the top right says "Type ⌘ to search". The main content area displays a monthly report from October 12, 2025, to November 12, 2025. The "Overview" section shows 0 Active pull requests, 0 Active issues, 0 Merged pull requests, 0 Open pull requests, 0 Closed issues, and 0 New issues. The "Summary" section notes 5 authors pushed 1 commit to master and 24 commits to all branches, with 0 files changed and 0 additions and 0 deletions. The "Top Committers" chart shows a single author with 20 commits. The sidebar on the left lists various metrics: Pulse, Contributors, Community, Community standards, Traffic, Commits, Code frequency, Dependency graph, Network, Forks, Actions usage metrics, Actions performance metrics, and People.

- Insight Report

The screenshot shows the GitHub Insights interface for the repository "OpenSource-Grupa-4 / ReportTB1". The main header bar includes links for Code, Issues (with 1), Pull requests, Actions, Projects, Wiki, Security, Insights (highlighted), and Settings. A search bar at the top right says "Type ⌘ to search". The main content area displays a weekly report from September 13, 2025, to September 20, 2025. The "Overview" section shows 6 Active pull requests, 2 Active issues, 6 Merged pull requests, 0 Open pull requests, 1 Closed issues, and 1 New issue. The "Summary" section notes 4 authors pushed 28 commits to main and 28 commits to all branches, with 41 files changed and 1191 additions and 13 deletions. The "Top Committers" chart shows multiple authors with varying commit counts. Below the main dashboard, a list of merged pull requests is shown, each with a merge date and a brief description. The sidebar on the left lists various metrics: Pulse, Contributors, Community, Community Standards, Traffic, Commits, Code frequency, Dependency graph, Network, Forks, Actions Usage Metrics, Actions Performance Metrics, and People.

5.2.3. Sprint 4

5.2.3.1. Sprint Planning 4.

5.2.3.2. Aspect Leaders and Collaborators.

5.2.3.3. Sprint Backlog 4.

ID	Title/Section	Description	Estimation (Hours)	Assigned To	Status
US21	Registro de Usuario	Como usuario, quiero registrar una cuenta desde la aplicación para acceder a los servicios de WeRide.	3	Castro Pariona, Jefferson Ernesto	Done
US22	Visualización de Vehículos	Como usuario, quiero visualizar los vehículos disponibles para elegir uno según mis necesidades.	4	Seminario Castillo, Diego Vicente	Done

ID	Title/Section	Description	Estimation (Hours)	Assigned To	Status
US23	Reserva de Vehículo	Como usuario, quiero poder reservar un vehículo seleccionando fecha y hora de uso.	4	Romero Meza, Jhimy Pool	Done
US24	Pago en Línea	Como usuario, quiero realizar el pago de mi reserva de manera segura mediante una interfaz simple y funcional.	5	Morales Sosa, Arnold Gabriel	Done
US25	Archivo db.json	Como desarrollador, quiero crear un archivo db.json con los datos de usuarios, vehículos y reservas para pruebas locales.	2	Bonifacio Jaramillo, Samuel Jesús	Done
US26	Pruebas e Integración	Como equipo, queremos realizar pruebas de funcionamiento e integración del flujo completo del sistema.	2	Todos	Done

5.2.3.4. Development Evidence for Sprint Review.

5.2.3.5. Execution Evidence for Sprint Review.

5.2.3.6. Services Documentation Evidence for Sprint Review.

5.2.3.7. Software Deployment Evidence for Sprint Review.

5.2.3.8. Team Collaboration Insights during Sprint.

5.3 Validation Interviews

5.3.1 Diseño de Entrevistas

Segmento 1: Universitarios y Jóvenes Profesionales (B2C)

Preguntas principales (Landing Page):

- "¿Qué entiendes que ofrece WeRide al ver la landing page?"
- "¿Cuál es el elemento que más te llama la atención o te genera confianza? (ej. precios, testimonios, método de pago)"
- "¿Qué información te falta para sentirte seguro al registrarte?"
- "¿Qué cambiarías en el diseño o en el copy para que la propuesta sea más clara?"
- "¿El llamado a la acción (CTA) es claro y te invita a registrarte o saber más?"
- "¿Las imágenes y gráficos reflejan la experiencia real del servicio?"
- "¿Confías en las formas de pago mostradas en la landing? ¿Por qué sí/no?"
- "¿La landing carga rápido y se ve bien en tu dispositivo móvil?"

Preguntas principales (Frontend Web Application):

- "Si elegiste 'Continuar como invitado', ¿te quedó claro qué funcionalidades tendrías disponibles sin registrarte?"
- "Al llegar a WeRide por primera vez, ¿qué impresión te causó la página de inicio? ¿Te resultó clara la propuesta de valor?"
- "Después de acceder, ¿te resultó intuitivo entender cómo navegar por la aplicación? ¿Qué fue lo primero que intentaste hacer?"
- "Si tuvieras que buscar el historial de tus viajes anteriores, ¿dónde buscarías primero? ¿Te parece lógica esa ubicación?"
- "Al buscar un vehículo, ¿la información mostrada (imagen, batería, marca, disponibilidad, precio) fue suficiente para decidir?"
- "Si todos los vehículos cercanos estuvieran ocupados, ¿qué esperarías que la app te ofreciera? (notificaciones, reserva anticipada, alternativas)"
- "Describe el proceso de crear una reserva: ¿hubo pasos poco claros o que te generaron dudas?"
- "¿Qué te pareció el proceso de pago y los formularios asociados (tarjeta, selección de plan)?"
- "¿Hubo algún momento en que te sentiste perdido/a o no supiste qué hacer a continuación?"

Segmento 2: Empresas y Planes Corporativos (B2B)

Preguntas principales (Landing Page):

- "Al ver la landing de WeRide, ¿qué entendiste que ofrece la plataforma específicamente para empresas?"
- "¿La propuesta de valor empresarial te pareció clara o orientada más al usuario final?"
- "¿Qué elemento visual o textual te transmitió mayor confianza para un uso corporativo?"
- "¿Qué información clave te faltó para evaluar si esto es viable para tu empresa? (precios, soporte, infraestructura, contratos, SLA, mantenimiento...)"
- "¿El CTA te orientó a una solución empresarial ('solicitar demo', 'agendar llamada', etc.) o lo sentiste más de consumo público?"

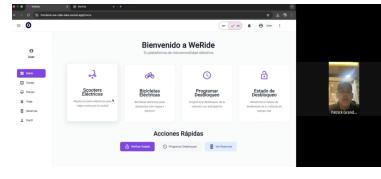
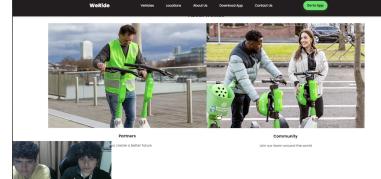
- "¿Consideras que la landing diferencia bien entre los servicios B2C y B2B?"
- "¿El diseño te transmite profesionalismo suficiente para un proveedor corporativo?"

Preguntas principales (Frontend Web Application):

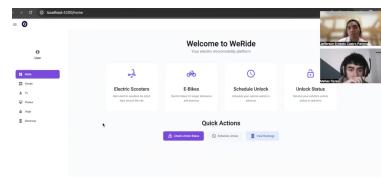
- "¿Los empleados deberían poder reservar vehículos? ¿O prefieres uso inmediato sin reserva?"
- "En una operación real, ¿cuánto antes se planifican los desplazamientos?"
- "¿Se necesitarían restricciones personalizadas (por sede, por tipo de vehículo, por horas)?"
- "Si todos los vehículos estuvieran ocupados, ¿qué solución esperas? (lista de espera, reserva prioritaria por cargo/área, flota adicional, alertas)"
- "¿Qué tipo de procesos necesitan aprobación previa?"
- "¿Qué modelo se adapta mejor a tu empresa: pago por uso, suscripción mensual, leasing de flota, tarifa fija por empleado?"
- "¿La interfaz te transmite suficiente profesionalismo para un uso corporativo?"

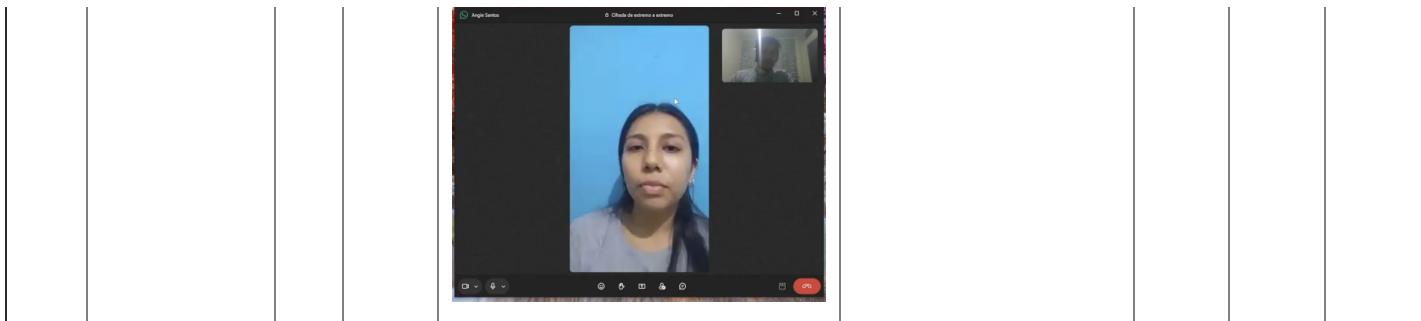
5.3.2 Registro de Entrevistas

Segmento 1: Universitarios y Jóvenes Profesionales (B2C)

#	Nombre	Edad	Distrito	Screenshot	URL Video	Timing	Duración	
1	Williams Avendaño (ESAN)	20	Surco		https://www.youtube.com/watch?v=vxBR4mxMcmQ	00:00:00	10:44	Usa Beat y ahorro y eco. Valora facilidad.
2	Patrick Cárdenas (PUCP)	19	Ate		https://www.youtube.com/watch?v=vxBR4mxMcmQ	00:10:45	8:30	Se mueve en motos por Ate. Landing clara.
3	Patrick Correa (U. Lima)	20	San Miguel		https://www.youtube.com/watch?v=vxBR4mxMcmQ	00:19:16	28:22	Transporte rápido. Landing clara. Sugiere tutoriales.

Segmento 2: Empresas y Planes Corporativos (B2B)

#	Nombre	Edad	Distrito	Screenshot	URL Video	Timing	Duración	
1	Ricardo Del Aguila (Inmobiliaria)	27	San Miguel		https://www.youtube.com/watch?v=0qsQ9N0wHMc	00:00:00	13:14	WeRide ofreciendo más. Landing más profesional. Panel corporativo de éxito.
2	Matías Flores (RRHH)	26	San Isidro		https://www.youtube.com/watch?v=vxBR4mxMcmQ	00:13:15	9:20	WeRide movilidad corporativa. Landing profesional. Requiere más información por sede, flujo.
3	Angie Santos (TI)	46	Santiago Surco		https://www.youtube.com/watch?v=vxBR4mxMcmQ	00:22:36	7:31	WeRide integración. Landing sólida. Integración con sistemas. Alertas automáticas.



5.3.3 Evaluaciones según heurísticas

Auditoria del Grupo 4:

UX Heuristics & Principles Evaluation

Usability – Inclusive Design – Information Architecture

CARRERA: Ingeniería de Software

CURSO: Desarrollo de Aplicaciones Open Source

SECCIÓN: 7380

PROFESORES: Todos

AUDITOR: Apaza Bocanegra, Elizabeth Noelia

CLIENTE(S):

- Bonifacio Jaramillo, Samuel Jesus
- Castro Pariona, Jefferson Ernesto
- Morales Sosa, Arnold Gabriel
- Romero Meza, Jhimy Pool
- Seminario Castillo, Diego Vicente

SITE o APP A EVALUAR: WeRide

TAREAS A EVALUAR:

El alcance de esta evaluación incluye la revisión de la usabilidad de las siguientes tareas:

1. Falta de Login en la Landing Page.
2. Funciones Inaccesibles en el login.
3. Interfaz demasiado limpia y sin distribucion clara, imagenes demasiado grandes.
4. Cards demasiado amplias, no funciona el boton de filtrado ni el de favoritos.
5. No hay datos para Bicicletas electricas.
6. No permite Cancelar Reservas.
7. No permite Guardar nuevas reservas desde Booking.
8. La seccion de viaje en ver detalles esta inactiva.
9. El boton Pagar de la seccion planes esta inactivo.
10. Botones de la barra principal como settings, user y demas estan inactivos.
11. No hay opciones de traducción dificultan comprension de recibos.
12. Reservar vehiculo inactivo.
13. Boton de reservar en cards tambien Inactivo.
14. No permite usar el boton edit booking.

ESCALA DE SEVERIDAD:

Los errores serán puntuados tomando en cuenta la siguiente escala de severidad

Nivel	Descripción
1	Problema superficial: puede ser fácilmente superador por el usuario ó ocurre con muy poco frecuencia. No necesita ser arreglado a no ser que exista disponibilidad de tiempo.
2	Problema menor: puede ocurrir un poco más frecuentemente o es un poco más difícil de superar para el usuario. Se le debería asignar una prioridad baja resolverlo de cara al siguiente reléase
3	Problema mayor: ocurre frecuentemente o los usuarios no son capaces de resolverlos. Es importante que sean corregidos y se les debe asignar una prioridad alta.
4	Problema muy grave: un error de gran impacto que impide al usuario continuar con el uso de la herramienta. Es imperativo que sea corregido antes del lanzamiento.

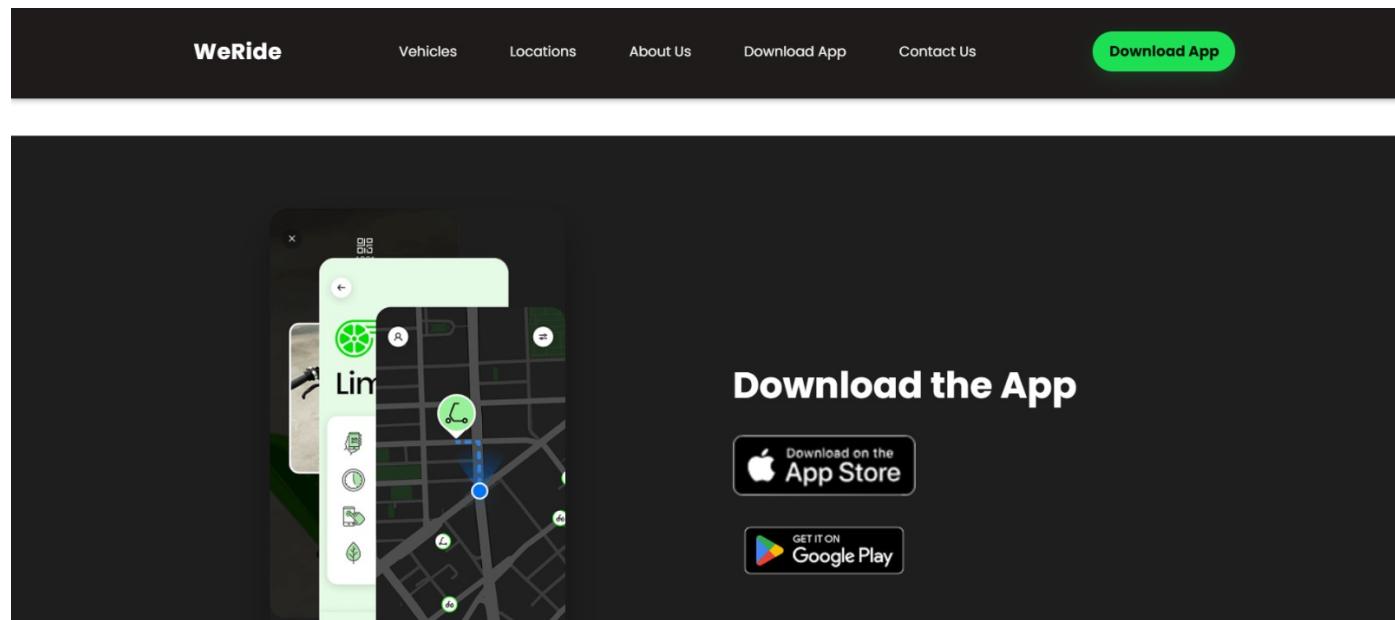
TABLA RESUMEN:

#	Problema	Escala de severidad	Heurística/Principio violada(o)
1	Falta de Login en la Landing Page	2	Usability: Libertad y control del usuario
2	Funciones inaccesibles en el Login	3	Usability: Visibilidad del estado del sistema
3	Interfaz sin distribución clara; imágenes demasiado grandes	1	Usability: Diseño estético y minimalista
4	Cards muy amplias; filtrado y favoritos inactivos	3	Usability: Libertad y control del usuario
5	No hay datos para Bicicletas Eléctricas	2	Usability: Consistencia y estándares
6	No permite cancelar reservas	4	Usability: Libertad y control del usuario
7	No se guardan nuevas reservas desde Booking	4	Usability: Prevención y recuperación de errores
8	Sección "Viaje" en Ver Detalles inactiva	2	Usability: Visibilidad del estado del sistema
9	Botón "Pagar" inactivo en la sección Planes	3	Usability: Libertad y control del usuario
10	Botones del menú superior (Settings, User, etc.) inactivos	2	Usability: Visibilidad del estado del sistema
11	No hay traducción; dificulta comprensión de recibos	1	Usability: Consistencia y estándares
12	Función "Reservar vehículo" inactiva	4	Usability: Prevención de errores
13	Botón reservar en las cards inactivo	3	Usability: Libertad y control del usuario
14	Botón "Edit Booking" no funciona	3	Usability: Flexibilidad y eficiencia de uso

DESCRIPCIÓN DE PROBLEMAS:

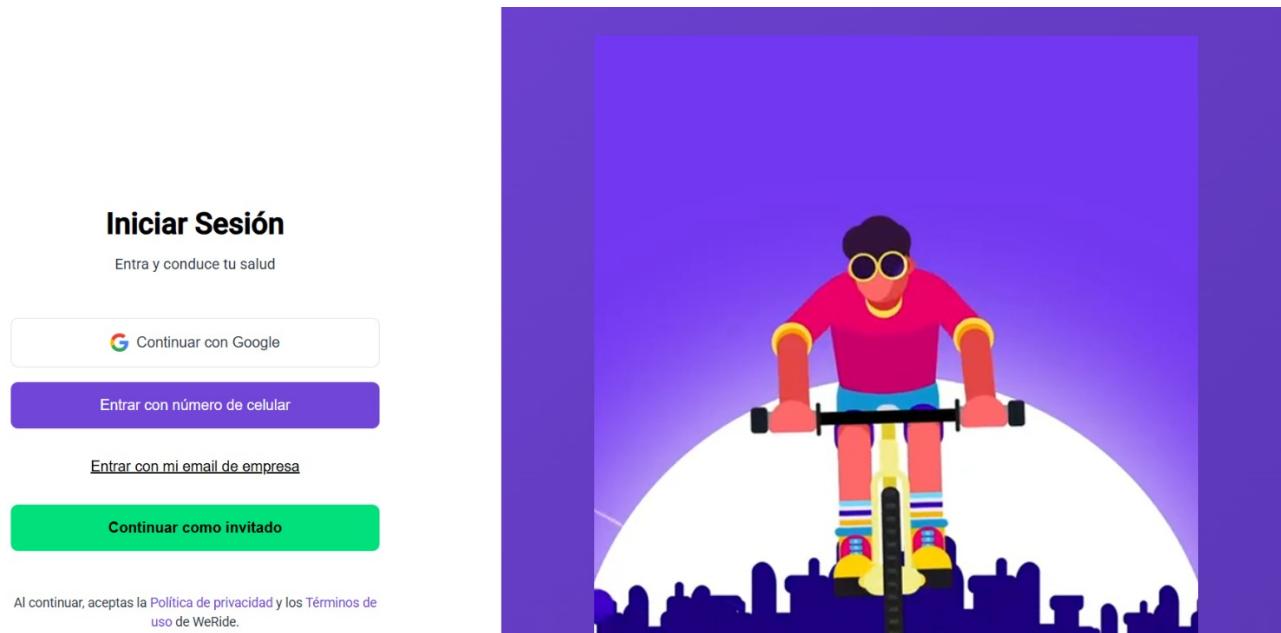
OBSERVACIÓN 1: Falta de Login en la Landing Page

- Severidad:** 2. Heurística violada: Usabilidad – Libertad y control del usuario
- Problema:** La Landing Page no ofrece un acceso visible hacia la pantalla de Login. El usuario no tiene una ruta clara para ingresar al sistema, generando confusión y aumentando la tasa de abandono.
- Recomendación:** Agregar un botón visible ("Iniciar Sesión", "Ir al Panel"), ubicado en el header o hero principal.



OBSERVACIÓN 2: Funciones inaccesibles en el Login

- Severidad:** 3. Heurística violada: Usabilidad – Visibilidad del estado del sistema
- Problema:** Algunas funciones de la pantalla de Login no responden o no muestran retroalimentación, como recuperación de contraseña o creación de cuenta. El usuario queda sin información sobre fallas.
- Recomendación:** Habilitar todos los botones y agregar mensajes de feedback claros (error, éxito, pasos a seguir).



OBSERVACIÓN 3: Interfaz demasiado limpia, sin distribución clara e imágenes muy grandes

- **Severidad:** 1. Heurística violada: Usabilidad – Diseño estético y minimalista
- **Problema:** La interfaz presenta espacios vacíos extensos y elementos demasiado grandes, lo que dificulta encontrar las funciones principales.
- **Recomendación:** Redistribuir el layout usando jerarquía visual, reducir el tamaño de imágenes y reforzar la navegación.

OBSERVACIÓN 4: Cards demasiado amplias; botones de filtrado y favoritos no funcionan

- **Severidad:** 3. Heurística violada: Usabilidad – Libertad y control del usuario
- **Problema:** Las cards ocupan demasiado espacio y los botones clave no funcionan, afectando directamente el flujo de selección de vehículos.
- **Recomendación:** Optimizar dimensiones, activar funcionalidad de filtros, favoritos y feedback visual.

Estado:	Disponible
Precio:	\$0.65/min
Rating:	★ 4.9
Batería:	95%
Velocidad Máx:	32 km/h

OBSERVACIÓN 5: No hay datos para Bicicletas eléctricas

- **Severidad:** 2. Heurística violada: Usabilidad – Consistencia y estándares
- **Problema:** La sección aparece vacía, lo que genera una experiencia inconsistente frente a otras categorías que sí muestran información.
- **Recomendación:** Cargar datos por defecto, mostrar placeholders o indicar que "Pronto habrá disponibilidad".

Garaje de Vehículos

Filtros

Tipo de Vehículo Bicicleta Eléctrica	Estado Todos	Marca	Precio Min (\$/min)	Precio Max (\$/min)
Rating Mínimo	Batería Mínima (%)			

Aplicar Filtros **Limpiar**

> encontraron vehículos con los filtros aplicados.

OBSERVACIÓN 6: No permite Cancelar Reservas

- **Severidad:** 4. Heurística violada: Usabilidad – Libertad y control del usuario
- **Problema:** Los usuarios no pueden cancelar sus reservas, bloqueando su flujo y generando frustración.
- **Recomendación:** Incorporar botón "Cancelar reserva" con confirmación de acción.

My Bookings

+ New Booking

- Xiaomi M365** COMPLETED
 - Plaza Mayor → Jockey Plaza
 - 10/7/25, 5:00 AM
 - 86 minutos
 - \$ 24.75
- Ninebot ES2** COMPLETED
 - Parque Kennedy → Larcomar
 - 10/7/25, 9:30 AM
 - 40 minutos
 - \$ 16.20
- Xiaomi Pro 2** COMPLETED
 - San Isidro Golf → Real Plaza Salaverry
 - 10/8/25, 3:15 AM
 - 42 minutos
 - \$ 16.17

Razor E300 CANCELLED

- Pueblo Libre Centro → Universidad del Pacífico
- 10/8/25, 7:00 AM
- N/A minutos
- \$ N/A

OBSERVACIÓN 7: No permite guardar nuevas reservas desde Booking

- Severidad:** 4. Heurística violada: Usabilidad – Prevención y recuperación de errores
- Problema:** El flujo de Booking falla al guardar una nueva reserva, interrumpiendo una función crítica.
- Recomendación:** Corregir el flujo técnico y añadir mensajes que indiquen causa del error y pasos sugeridos.

User

Reservas

Book Vehicle

Electric Micromobility Vehicle

Date dd / mm / aaaa

Unlock Time -- : --

Duration (minutes)

1

Book Now Save as Draft

OBSERVACIÓN 8: La sección "Viaje" en Ver Detalles está inactiva

- Severidad:** 2. Heurística violada: Usabilidad – Visibilidad del estado del sistema
- Problema:** El panel "Viaje" no despliega información ni responde a la interacción. El usuario no entiende si es temporal, un error o falta de permisos.
- Recomendación:** Activar el componente o mostrar un mensaje informativo ("Función en desarrollo", "No hay información disponible").

OBSERVACIÓN 9: El botón "Pagar" de la sección Planes está inactivo

- Severidad:** 3. Heurística violada: Usabilidad – Libertad y control del usuario
- Problema:** El usuario no puede completar el pago de un plan, bloqueando una acción central del modelo de negocio.
- Recomendación:** Activar el botón, conectar el flujo a la pasarela de pago y agregar validaciones previas.

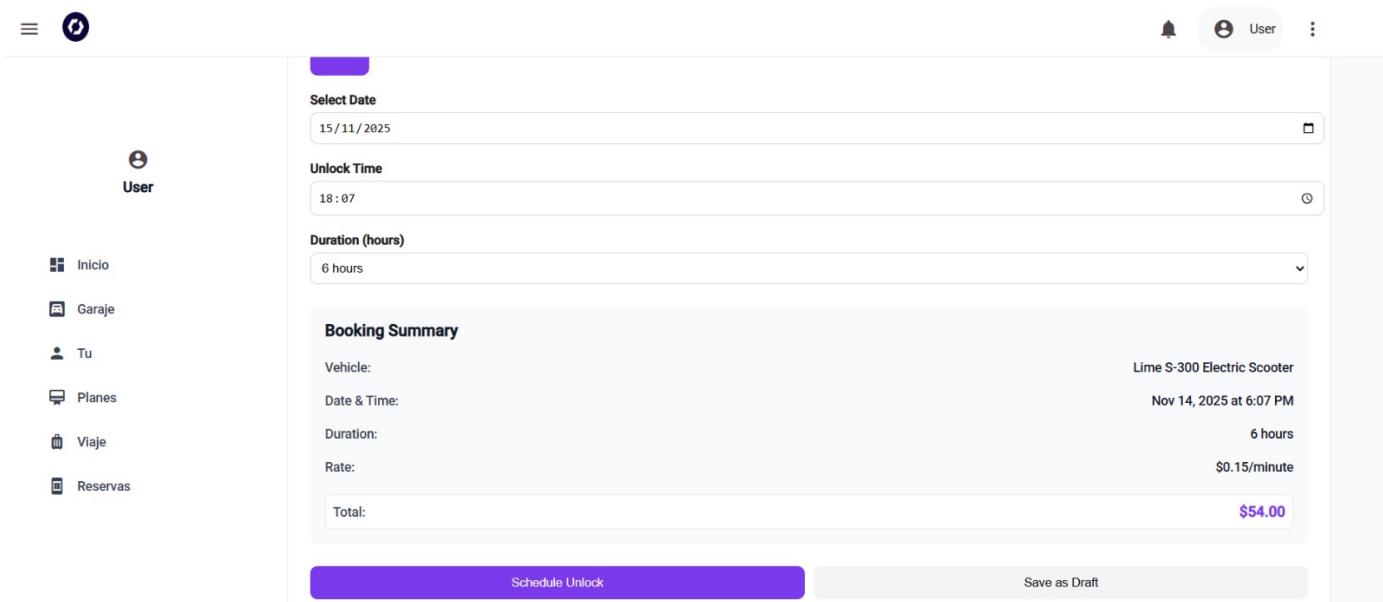
OBSERVACIÓN 10: Botones como Settings y User en la barra principal están inactivos

- Severidad:** 2. Heurística violada: Usabilidad – Visibilidad del estado del sistema
- Problema:** Los accesos principales no ejecutan acción alguna, lo que genera percepción de sistema incompleto o fallido.
- Recomendación:** Implementar navegación interna o, al menos, mensajes temporales de "módulo en construcción".

OBSERVACIÓN 11: No hay opciones de traducción y dificulta la comprensión de recibos

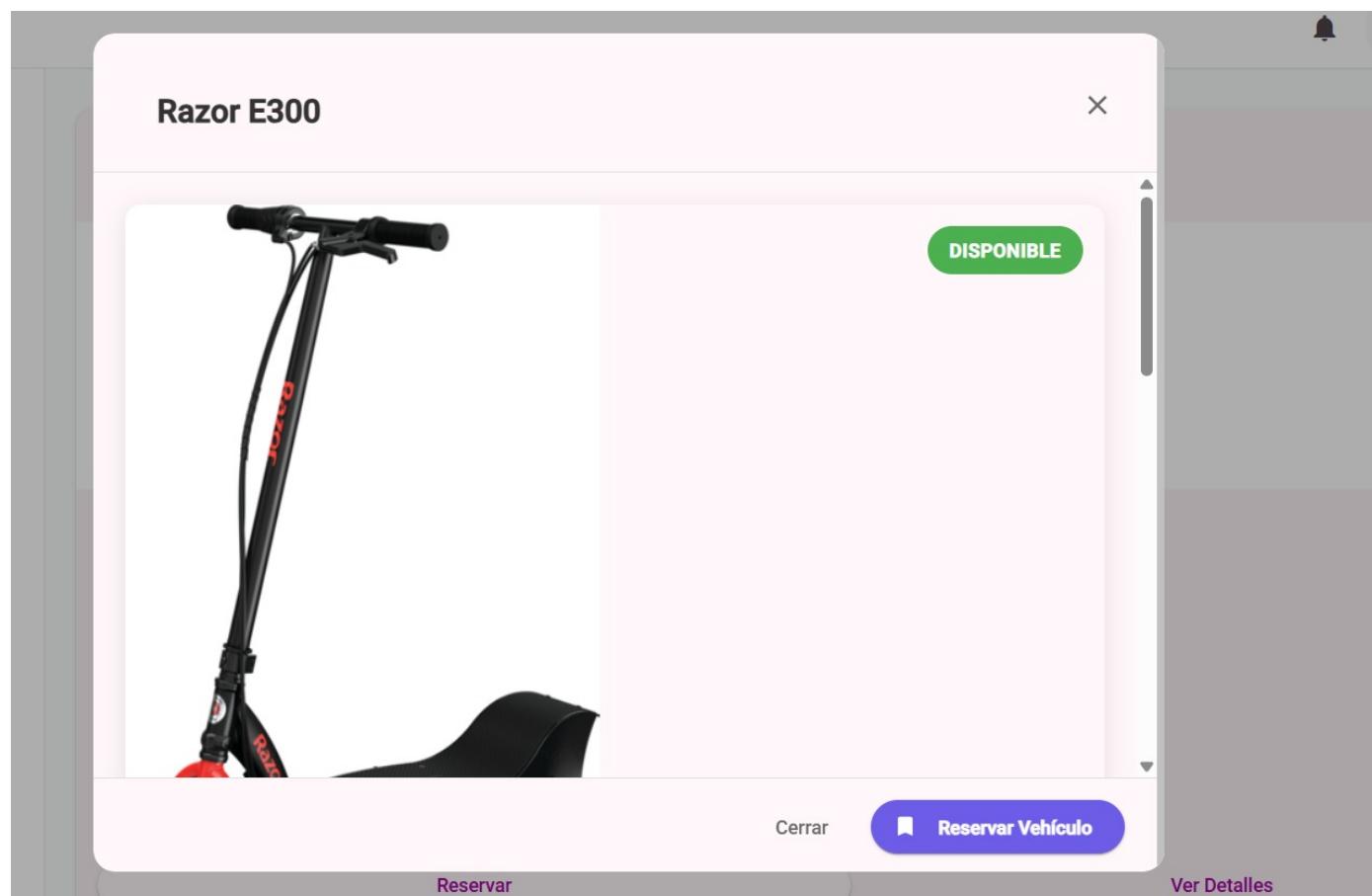
- Severidad:** 1. Heurística violada: Usabilidad – Consistencia y estándares

- **Problema:** El contenido de recibos y pantallas clave está solo en un idioma, dificultando la comprensión para usuarios que no lo manejan.
- **Recomendación:** Integrar sistema de internacionalización (i18n) y traducciones completas.



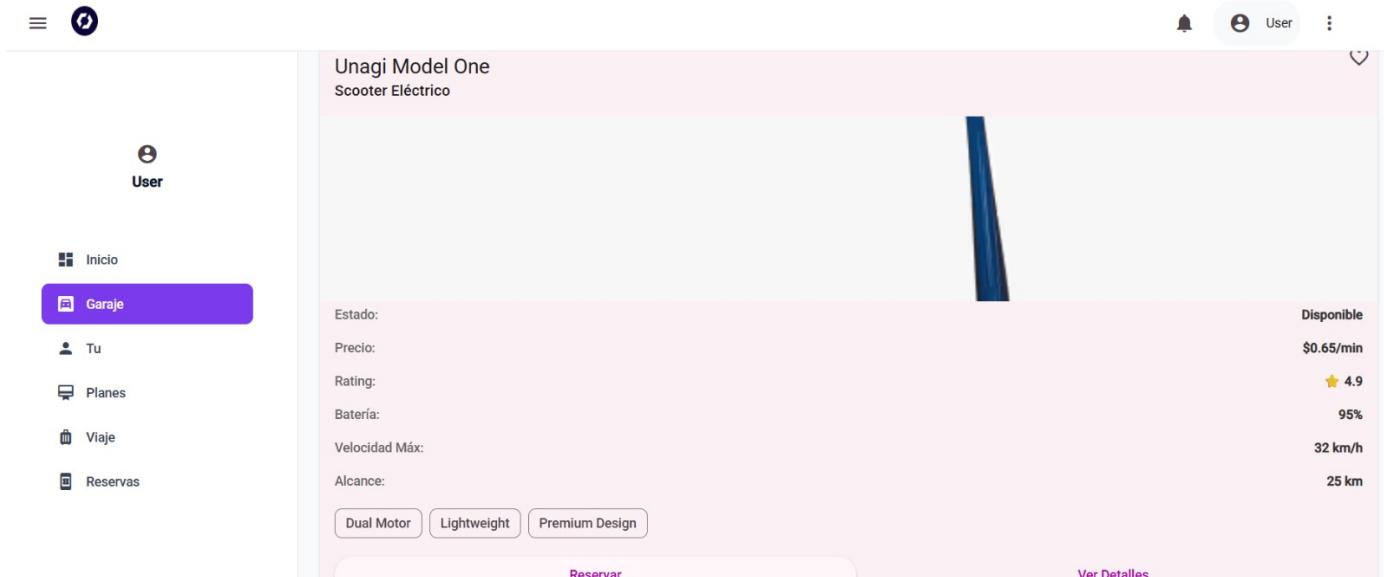
OBSERVACIÓN 12: Reservar vehículo está inactivo

- **Severidad:** 4. Heurística violada: Usabilidad – Prevención de errores
- **Problema:** Al intentar reservar un vehículo, el botón o acción no funciona. Esto afecta directamente la funcionalidad principal del sistema.
- **Recomendación:** Habilitar el botón, validar datos y mostrar confirmación de reserva exitosa.



OBSERVACIÓN 13: Botón de reservar dentro de las cards también está inactivo

- **Severidad:** 3. Heurística violada: Usabilidad – Libertad y control del usuario
- **Problema:** El usuario no puede reservar desde la vista de cards, obligándolo a pasos adicionales o impidiendo continuar.
- **Recomendación:** Activar interacción, añadir feedback visual y redirigir al flujo de Booking.

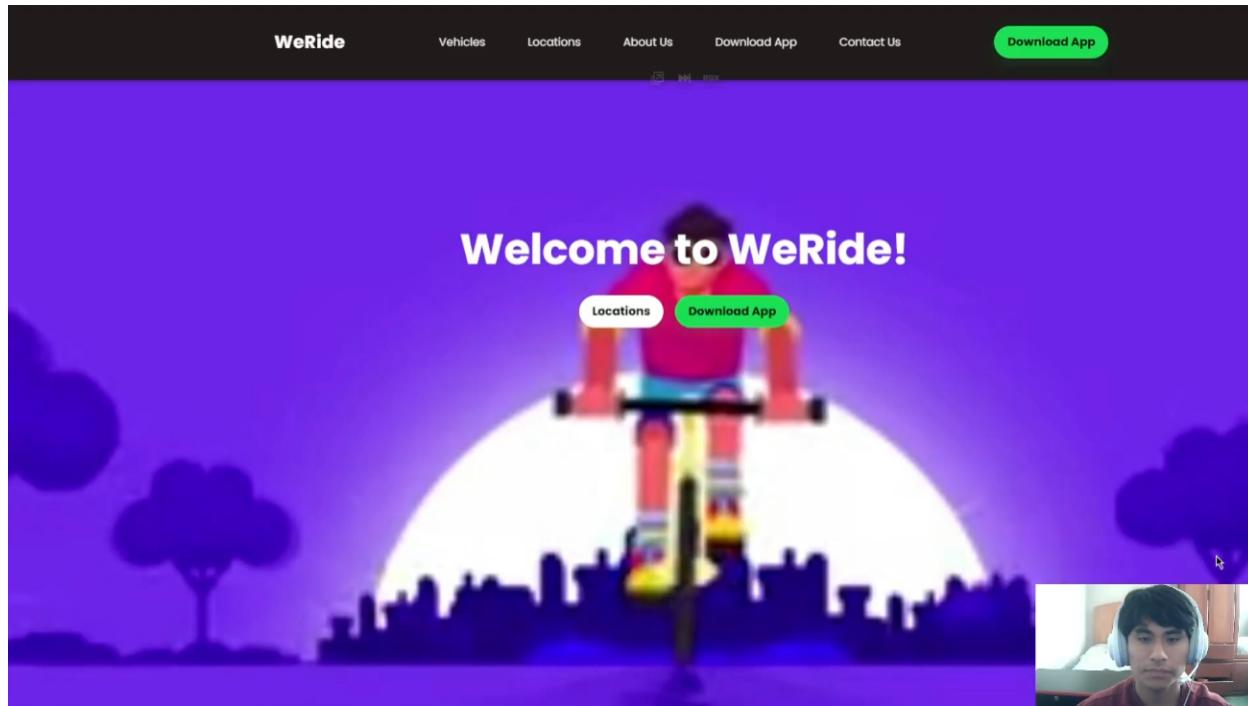


OBSERVACIÓN 14: No permite usar el botón "Edit Booking"

- Severidad:** 3. Heurística violada: Usabilidad – Flexibilidad y eficiencia de uso
- Problema:** El botón para editar una reserva no responde, y el usuario no puede modificar fechas, horarios o vehículo seleccionado.
- Recomendación:** Implementar formulario editable, mensajes de validación y confirmación de cambios.

The screenshot shows a mobile application interface titled 'My Bookings'. It lists two completed bookings: 'Xiaomi M365' and 'Ninebot ES2'. Each booking card shows the vehicle name, destination, date, duration, and cost. Below each card are 'Edit' and 'Cancel' buttons. To the right of the main content is a developer tools console window showing log messages related to API service initialization and data loading. The console also displays error and warning messages, such as 'No warnings' and 'No verbose'.

5.4. Video About The Product



[Ver Video del Producto](#)

Conclusiones

- El proyecto de micromovilidad eléctrica compartida responde a necesidades reales de movilidad en áreas urbanas peruanas: congestión, tiempos de traslado elevados, y demanda por alternativas más económicas y sostenibles. La combinación de scooters, bicicletas y motos eléctricas gestionadas desde una única plataforma web representa una propuesta con alto potencial de adopción, especialmente en segmentos juveniles y corporativos.
- La identificación de dos segmentos prioritarios —jóvenes universitarios y empresas con planes de suscripción— es coherente con los hallazgos cualitativos. Los primeros buscan conveniencia, bajo costo y métodos de pago locales (Yape/Plin); las empresas buscan soluciones de movilidad para colaboradores y valoran la previsibilidad de costos y la imagen de sostenibilidad. Esta segmentación favorece estrategias de adquisición y retención diferenciadas.
- La propuesta de combinar pago por uso (desbloqueo + tarifa por minuto) con pasos diarios, mensuales y corporativos ofrece una mezcla adecuada para equilibrar ingresos inmediatos y recurrentes. Los precios sugeridos y el descuento universitario crean incentivos para adopción temprana y fidelización. No obstante, la rentabilidad dependerá de la tasa de utilización por vehículo (objetivo mínimo: 4–5 viajes/día) y de una gestión eficiente de costos operativos (mantenimiento, recarga, logística).
- Las funciones críticas —búsqueda en tiempo real (GPS/IoT), mapa interactivo, reservas con temporizador y desbloqueo digital (QR/PIN/IoT)— están correctamente priorizadas. Estas funcionalidades garantizan una experiencia simple y confiable desde la web, mientras que el desbloqueo digital y la gestión remota son componentes esenciales para control y seguridad de la flota.
- Alexander, I., & Beus-Dukic, L. (2009). *Discovering Requirements: How to Specify Products and Services*. Wiley.
- Cohn, M. (2004). *User Stories Applied: For Agile Software Development*. Addison-Wesley.
- Gamma, E., Helm, R., Johnson, R., & Vlissides, J. (1994). *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley.
- Kruchten, P. (2003). *The Rational Unified Process: An Introduction*. Addison-Wesley.
- Newman, S. (2015). *Building Microservices: Designing Fine-Grained Systems*. O'Reilly Media.
- Sommerville, I. (2016). *Software Engineering* (10th ed.). Pearson.
- W3C. (2024). *HTML & CSS Standards*. Recuperado de: <https://www.w3.org/>
- Mozilla Developer Network (MDN). (2024). *JavaScript Guide*. Recuperado de: <https://developer.mozilla.org/es/docs/Web/JavaScript/Guide>
- GitHub Docs. (2024). *GitHub Pages Documentation*. Recuperado de: <https://docs.github.com/en/pages>
- Stripe Docs. (2024). *Stripe Payments Integration Guide*. Recuperado de: <https://stripe.com/docs>
- Figma. (2024). *Figma Design Documentation*. Recuperado de: <https://help.figma.com/>
- Google Maps Platform. (2024). *Maps JavaScript API Documentation*. Recuperado de: <https://developers.google.com/maps/documentation/javascript>
- Node.js Foundation. (2024). *Node.js Documentation*. Recuperado de: <https://nodejs.org/en/docs/>
- Vue.js. (2024). *Vue.js Guide*. Recuperado de: <https://vuejs.org/guide/introduction.html>

Anexos.

[About The Product](#)

[Organización](#)

[Reporte del Proyecto](#)

[Repositorio de Frontend Web Application](#)

[Deploy de Frontend Web Application](#)

[Repositorio de Backend Web Services](#)

[Deploy de Backend Web Services](#)

[Repositorio de Landing Page](#)

[Deploy de Landing Page](#)

[Presentación de Canvas](#)

[Figma](#)