

COURSE PROJECT



Universidad Peruana de Ciencias Aplicadas

Ingeniería de Software

Desarrollo de Aplicaciones Móviles - NRC: 346

Ciclo: 6to

Profesor: Jorge Luis Mayta Guillermo

INFORME

Startup: Code Mondoguito

Producto: ParkingNow

Team Members:

Member	Code
Calisaya Sánchez Juan Jesús	u202121935
Hidalgo Lopez, Mathias Adriano	U202213222
Samuel Elias Molina Asencios	U20191A456
Diego Ulises Soto Quispe	U2022144778

Abril 2025

Registro de Versiones del Informe

Versión	Fecha	Autor(es)	Descripción de la modificación
TB1	07/09/2024	Calisaya Sánchez Juan Jesús Hidalgo Lopez Mathias Adriano Samuel Elias	Se agregó el contenido del capítulo 1 (apartados 1.1, 1.1.1, 1.1.2, 1.2, 1.2.1, 1.2.2, 1.2.2.1, 1.2.2.2, 1.2.2.3, 1.2.2.4, 1.3); el contenido del capítulo 2 (apartados 2.1, 2.1.1, 2.1.2, 2.2, 2.2.1, 2.2.2, 2.2.3, 2.3, 2.3.1, 2.3.2, 2.3.3, 2.3.4, 2.3.5, 2.4); el contenido del capítulo 3 (apartados 3.1, 3.2, 3.3 y 3.4); y el contenido del capítulo 4 (apartados 4.1, 4.1.1, 4.1.1.1, 4.1.1.2, 4.1.1.3, 4.1.2, 4.1.3, 4.1.3.1, 4.1.3.2, 4.1.3.3, 4.2, 4.2.X, 4.2.X.1, 4.2.X.2, 4.2.X.3, 4.2.X.4, 4.2.X.5, 4.2.X.6, 4.2.X.6.1, 4.2.X.6.2).

Versión	Fecha	Autor(es)	Descripción de la modificación
		Molina Asencios Frezzia Eldaa Isabel Espinoza Paredes Diego Ulises Soto Quispe	
TP1	16/05/2025	Calisaya Sánchez Juan Jesús Hidalgo Lopez Mathias Adriano Samuel Elias Molina Asencios Frezzia Eldaa Isabel Espinoza Paredes Diego Ulises Soto Quispe	Se añadió el contenido del capítulo 5 Solution UI/UX Design , incluyendo los apartados: 5.1 Product design, 5.1.1 Style Guidelines (5.1.1.1), 5.1.2 Information Architecture (5.1.2.1 a 5.1.2.5), 5.1.3 Landing Page UI Design (5.1.3.1, 5.1.3.2) y 5.1.4 Mobile Applications UX/UI Design (5.1.4.1 a 5.1.4.5). También se incorporó el capítulo 6 Product Implementation, Validation & Deployment , con los apartados: 6.1 Software Configuration Management (6.1.1 a 6.1.4), 6.2 Landing Page, Services & Applications Implementation, y 6.2.1 Sprint 1 (6.2.1.1 a 6.2.1.7).
TB2	03/06/2025	Calisaya Sánchez Juan Jesús Hidalgo Lopez Mathias Adriano Samuel Elias Molina Asencios Diego Ulises Soto Quispe	Se añadió el contenido correspondiente al capítulo 6.2 Sprint 2, incluyendo los apartados: 6.2.2.1 Sprint Planning 2, 6.2.2.2 Sprint Backlog 2, 6.2.2.3 Development Evidence for Sprint Review, 6.2.2.4 Testing Suite Evidence for Sprint Review, 6.2.2.5 Execution Evidence for Sprint Review, 6.2.2.6 Services Documentation Evidence for Sprint Review, 6.2.2.7 Software Deployment Evidence for Sprint Review, y 6.2.2.8 Team Collaboration Insights during Sprint. También se incorporó el capítulo 6.3 Validation Interviews, con los apartados: 6.3.1 Diseño de Entrevistas, 6.3.2 Registro de Entrevistas y 6.3.3 Evaluaciones según heurísticas. Finalmente, se añadió el contenido del apartado 6.4 Video About-the-Product como parte del entregable final.
TF1	01/07/2025	Calisaya Sánchez Juan Jesús Hidalgo Lopez Mathias Adriano	Se añadió el contenido correspondiente al capítulo 6.2 Sprint 3, incluyendo los apartados: 6.2.3.1 Sprint Planning 3, 6.2.3.2 Sprint Backlog 3, 6.2.3.3 Development Evidence for Sprint Review, 6.2.3.4 Testing Suite Evidence for Sprint Review, 6.2.3.5 Execution Evidence for Sprint Review, 6.2.3.6 Services Documentation Evidence for Sprint Review, 6.2.3.7 Software Deployment Evidence for Sprint Review, y 6.2.3.8 Team Collaboration Insights during Sprint.

Versión	Fecha	Autor(es)	Descripción de la modificación
		Samuel Elias Molina Asencios Diego Ulises Soto Quispe	

Project Report Collaboration Insights

TB1: Las tareas asignadas para la entrega TB1 han sido completadas y están documentadas en el siguiente repositorio de GitHub perteneciente a la organización del equipo:

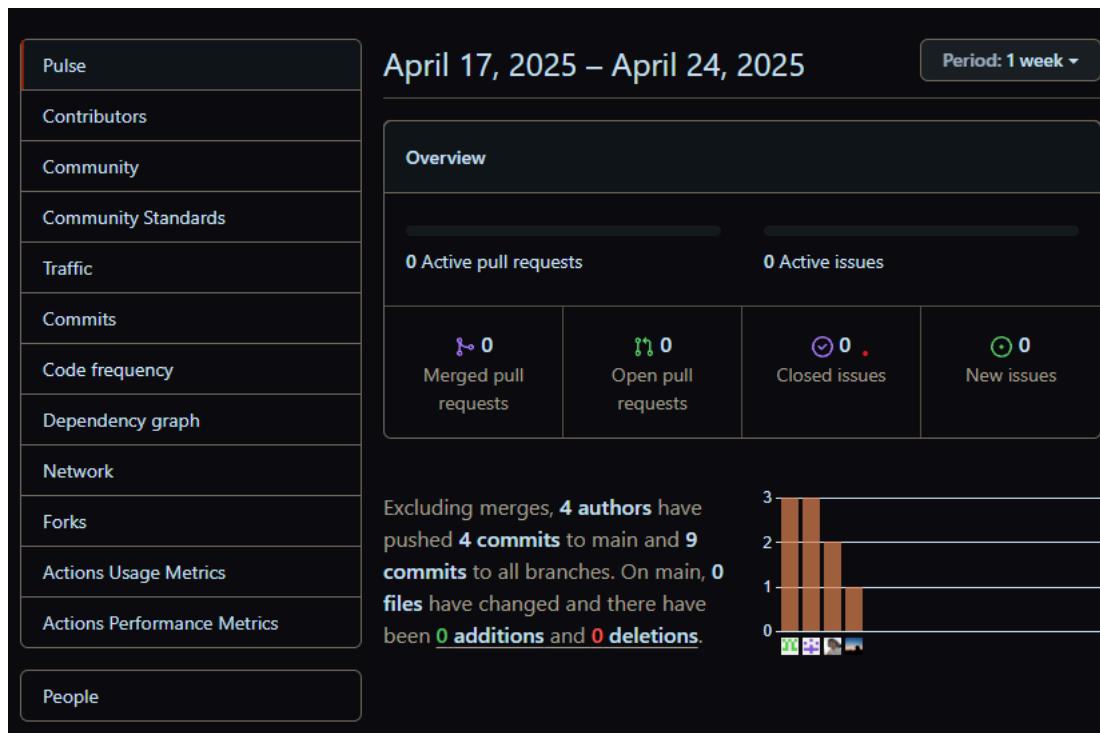
Repositorio: G3-UPC-CC238-346-ParkingNow/Report:

<https://github.com/G3-UPC-CC238-346-ParkingNow/Report>

Durante la preparación del informe, se llevaron a cabo las siguientes actividades:

- Se escribieron y diagramaron los contenidos asignados a cada miembro en formato Markdown, seguidos de commits para asegurar el progreso en el repositorio.
- Se desarrollaron los capítulos I, II, III y IV del informe, abarcando desde la presentación del proyecto hasta el diseño de software, aplicando las técnicas revisadas en clase como Lean UX, entrevistas, análisis de usuarios y Domain-Driven Design.
- Los artefactos necesarios (diagramas, mapas, matrices) fueron elaborados con herramientas recomendadas y almacenados en la carpeta `Assets` dentro de la rama `develop`, permitiendo su referencia desde el informe.
- Se organizaron reuniones virtuales para coordinar los avances y distribuir las secciones, priorizando el desarrollo del contenido teórico-práctico del informe técnico.

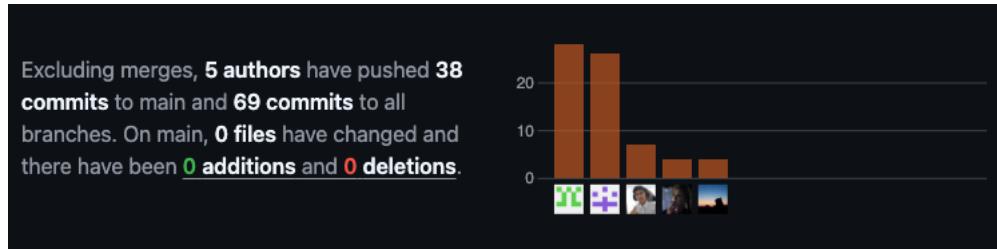
Esta entrega representa una evidencia del trabajo colaborativo del equipo, permitiendo cumplir con el Student Outcome 7 (ABET – EAC), demostrando la capacidad para adquirir conocimientos, aplicar metodologías y documentar procesos en un entorno ágil.



TP1: Project Report Collaboration Insights

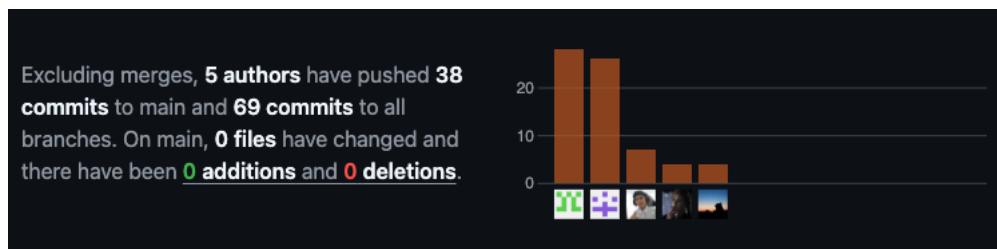
- Se desarrollaron y documentaron las **primeras 17 historias de usuario (US01 a US17)**, correspondientes al **Sprint 1**, las cuales abarcan tanto la **Landing Page web** como el inicio de la **aplicación móvil**.
- Se implementó el diseño y desarrollo de la **Landing Page**, asegurando responsividad para diferentes dispositivos (web y mobile), siguiendo la paleta de colores y lineamientos UX/UI establecidos para el proyecto.

- Se trabajaron de forma paralela dos versiones de la **aplicación móvil**:
 - Con **Flutter**, se construyó el flujo base de autenticación (login, registro, selección de rol) bajo una arquitectura modular limpia.
 - Con **Kotlin y Jetpack Compose**, se implementaron pantallas equivalentes para la versión nativa Android, cumpliendo con los lineamientos del curso.
- Se configuró e inició el desarrollo del **backend en Node.js**, implementando los primeros endpoints para login, registro y validación de usuarios. La API fue conectada al frontend mediante pruebas locales.
- Todo el avance se gestionó colaborativamente usando control de versiones en GitHub. Los contenidos fueron redactados en formato Markdown y organizados en el repositorio, haciendo commits constantes por parte de los integrantes.
- Se utilizaron herramientas recomendadas para la creación de diagramas, wireframes y mockups, los cuales fueron almacenados en la carpeta Assets dentro de la rama `develop` del repositorio.
- Finalmente, se organizaron reuniones semanales para coordinar avances, resolver bloqueos y garantizar una adecuada distribución de tareas entre el equipo de frontend, backend y diseño.



TB2: Project Report Collaboration Insights

- Se desarrollaron y documentaron nuevas historias de usuario correspondientes al Sprint 2, enfocadas en funcionalidades clave como reservas de espacios, generación de tickets, flujos de pago y opciones de configuración, mejorando la experiencia tanto para conductores como para dueños de estacionamiento.
- Se trabajaron de forma paralela las dos versiones de la aplicación móvil:
 - Con **Kotlin y Jetpack Compose**, se avanzó en el segmento del **conductor**, implementando pantallas como Dashboard, Reservas, Soporte, Seguimiento, Configuración, Notificaciones, Pago, Ticket y Confirmación de éxito.
 - Con **Flutter**, se construyó la base del segmento del **dueño de estacionamiento**, incluyendo login, dashboard, gestión de reservas y validación de espacios disponibles.
- Se fortaleció el backend en **Node.js**, añadiendo endpoints necesarios para los nuevos flujos de reserva, validación de pagos, y registro de transacciones. Se realizaron pruebas exitosas de integración con el frontend y validaciones con Postman.
- Todo el trabajo fue gestionado colaborativamente mediante control de versiones con GitHub. Los contenidos fueron redactados en formato Markdown, organizados por carpetas, y se mantuvieron commits constantes y bien distribuidos entre los integrantes.
- Se utilizaron herramientas de apoyo como diagramas de flujo, wireframes y mockups para guiar el diseño de nuevas interfaces. Estos recursos fueron almacenados en la carpeta `Assets` dentro de la rama `develop` del repositorio.
- Se organizaron reuniones periódicas para coordinar avances, resolver bloqueos y asegurar una distribución adecuada de tareas entre frontend, backend, diseño y validación. También se aplicaron entrevistas y evaluaciones heurísticas como parte del proceso iterativo de mejora del producto.



TF: Project Report Collaboration Insights

- Se desarrollaron y documentaron nuevas historias de usuario correspondientes al Sprint 3, enfocadas en funcionalidades clave como alertas, historial de transacciones, reportes, auditorías, configuración de tarifas y endpoints de seguridad. Estas funcionalidades fortalecen la capa de servicios y mejoran la trazabilidad y administración de la plataforma.
- Se trabajaron de forma paralela ambas versiones móviles de la aplicación:
 - Con **Kotlin y Jetpack Compose**, se completaron los módulos de conductor incluyendo integración con notificaciones, historial, alertas, seguimiento de reserva, favoritos y configuración de usuario.

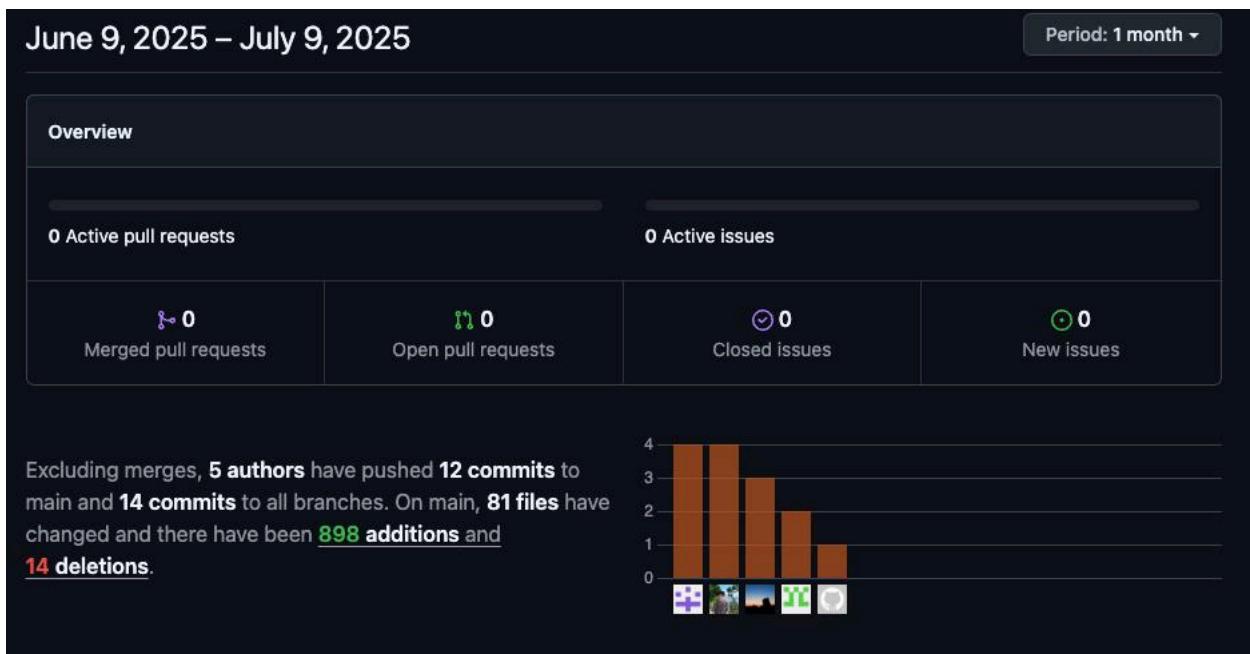
- Con Flutter, se implementaron funciones para el panel del dueño de estacionamiento, incluyendo validación de ocupación, gestión de promociones y alertas de seguridad.

Se expandió el backend en **Node.js**, desarrollando endpoints para reportes, auditorías, feedback, seguridad, cambios de tarifas y reembolsos. Estas funcionalidades fueron probadas exitosamente con Postman, y se integraron en los flujos frontend.

Todo el desarrollo fue gestionado colaborativamente mediante GitHub, con control de versiones activo y organización clara en formato Markdown. Se mantuvieron commits constantes y revisiones cruzadas entre los integrantes.

Se emplearon recursos visuales de apoyo como diagramas, flujos, wireframes y mockups ubicados en la carpeta `Assets` dentro de la rama `develop`, facilitando la alineación visual entre los equipos.

Se realizaron reuniones de cierre de ciclo Sprint, evaluaciones heurísticas finales, validación de endpoints y coordinación integral entre frontend, backend y diseño para culminar la versión estable de la aplicación.



Contenido

- Registro de Versiones del Informe
- Project Report Collaboration Insights
- Student Outcome

Capítulo I: Presentación

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

Capítulo II: Requirements Elicitation & Analysis

- 2.1. Competidores
 - 2.1.1. Análisis competitivo
 - 2.1.2. Estrategias y tácticas frente a competidores
- 2.2. Entrevistas
 - 2.2.1. Diseño de entrevistas
 - 2.2.2. Registro de entrevistas
 - 2.2.3. Análisis de entrevistas
- 2.3. Needfinding
 - 2.3.1. User Personas
 - 2.3.2. User Task Matrix
 - 2.3.3. User Journey Mapping
 - 2.3.4. Empathy Mapping
 - 2.3.5. As-is Scenario Mapping
- 2.4. Ubiquitous Language

Capítulo III: Requirements Specification

- 3.1. To-Be Scenario Mapping
- 3.2. User Stories
- 3.3. Impact Mapping
- 3.4. Product Backlog

Capítulo IV: Solution Software Design

- 4.1. Strategic-Level Domain-Driven Design
 - 4.1.1. EventStorming
 - 4.1.1.1. Candidate Context Discovery
 - 4.1.1.2. Domain Message Flows Modeling
 - 4.1.1.3. Bounded Context Canvases
 - 4.1.2. Context Mapping
 - 4.1.3. Software Architecture
 - 4.1.3.1. Software Architecture Context Level Diagrams
 - 4.1.3.2. Software Architecture Container Level Diagrams
 - 4.1.3.3. Software Architecture Deployment Diagrams
- 4.2. Tactical-Level Domain-Driven Design
 - 4.2.X. Bounded Context:
 - 4.2.X.1. Domain Layer
 - 4.2.X.2. Interface Layer
 - 4.2.X.3. Application Layer
 - 4.2.X.4. Infrastructure Layer
 - 4.2.X.5. Bounded Context Software Architecture Component Level Diagrams
 - 4.2.X.6. Bounded Context Software Architecture Code Level Diagrams
 - 4.2.X.6.1. Bounded Context Domain Layer Class Diagrams
 - 4.2.X.6.2. Bounded Context Database Design Diagram

Capítulo V: Solution UI/UX Design

- 5.1. Product Design
 - 5.1.1. Style Guidelines
 - 5.1.1.1. General Style Guidelines
 - 5.1.2. Information Architecture
 - 5.1.2.1. Organization Systems
 - 5.1.2.2. Labelling Systems
 - 5.1.2.3. SEO Tags and Meta Tags

- 5.1.2.4. Searching Systems
- 5.1.2.5. Navigation Systems
- 5.1.3. Landing Page UI Design
 - 5.1.3.1. Landing Page Wireframe
 - 5.1.3.2. Landing Page Mock-up
- 5.1.4. Mobile Applications UX/UI Design
 - 5.1.4.1. Mobile Applications Wireframes
 - 5.1.4.2. Mobile Applications Wireflow Diagrams
 - 5.1.4.3. Mobile Applications Mock-ups
 - 5.1.4.4. Mobile Applications User Flow Diagrams
 - 5.1.4.5. Mobile Applications Prototyping

Capítulo VI: Product Implementation, Validation & Deployment

- 6.1. Software Configuration Management
 - 6.1.1. Software Development Environment Configuration
 - 6.1.2. Source Code Management
 - 6.1.3. Source Code Style Guide & Conventions
 - 6.1.4. Software Deployment Configuration
- 6.2. Landing Page & Mobile Application Implementation
 - 6.2.X. Sprint n
 - 6.2.X.1. Sprint Planning n
 - 6.2.X.2. Sprint Backlog n
 - 6.2.X.3. Development Evidence for Sprint Review
 - 6.2.X.4. Testing Suite Evidence for Sprint Review
 - 6.2.X.5. Execution Evidence for Sprint Review
 - 6.2.X.6. Services Documentation Evidence for Sprint Review
 - 6.2.X.7. Software Deployment Evidence for Sprint Review
 - 6.2.X.8. Team Collaboration Insights during Sprint
- 6.3. Validation Interviews
 - 6.3.1. Diseño de Entrevistas
 - 6.3.2. Registro de Entrevistas
 - 6.3.3. Evaluaciones según heurísticas
- 6.4. Video About-the-Product

Secciones Finales

- Conclusiones
 - Conclusiones y recomendaciones
 - Video About-the-Team
- Bibliografía
- Anexos

ABET – EAC - Student Outcome 7

Criterio: La capacidad de adquirir y aplicar nuevos conocimientos según sea necesario, utilizando estrategias de aprendizaje apropiadas.

En el siguiente cuadro se describen las acciones realizadas y enunciados de conclusiones por parte del grupo, que permiten sustentar el haber alcanzado el logro del ABET – EAC - Student Outcome 7.

ABET – EAC - Resultado del Estudiante 7

Criterio específico	Acciones realizadas	Conclusiones
Actualiza conceptos y conocimientos necesarios para	TB1:	TB1: El equipo logró llevar a la práctica los conceptos aprendidos

Criterio específico	Acciones realizadas	Conclusiones
<p>su desarrollo profesional y en especial para su proyecto en soluciones de software.</p> <ul style="list-style-type: none"> - <i>Calisaya Sánchez Juan Jesús</i>: Redactó el capítulo 1 del informe aplicando conceptos de presentación de startups y problemática. - <i>Hidalgo Lopez Mathias Adriano</i>: Aplicó Lean UX Problem Statements y desarrolló el Canvas a partir de investigaciones propias. - <i>Samuel Elias Molina Asencios</i>: Implementó el Empathy Mapping y el User Journey Mapping tras investigar su aplicación en UX. - <i>Frezzia Eldaa Isabel Espinoza Paredes</i>: Aportó en entrevistas y definición de requerimientos investigando metodologías efectivas. - <i>Diego Ulises Soto Quispe</i>: Lideró la estructura del informe y aplicó documentación técnica basada en buenas prácticas. <p>TP:</p> <ul style="list-style-type: none"> - <i>Calisaya Sánchez Juan Jesús</i>: Aplicó conocimientos sobre configuración y documentación técnica de servicios backend en NestJS, participando en la escritura de endpoints y evidencias del capítulo 6. - <i>Hidalgo Lopez Mathias Adriano</i>: Redactó de forma estructurada todas las evidencias de desarrollo y ejecución del backend, demostrando dominio en documentación técnica formal del proyecto. - <i>Samuel Elias Molina Asencios</i>: Implementó toda la estructura de datos y endpoints de la API en NestJS, aplicando principios de arquitectura REST y organización por módulos. - <i>Frezzia Eldaa Isabel Espinoza Paredes</i>: Aplicó conocimientos sobre metodologías ágiles al organizar las secciones de Sprint Planning y Backlog en el informe, aunque sin profundizar técnicamente. - <i>Diego Ulises Soto Quispe</i>: Aplicó habilidades de desarrollo frontend web y mobile, trabajando desde cero en la Landing Page y en la estructura de las apps móviles (Kotlin y Flutter). <p>TB2:</p> <ul style="list-style-type: none"> - <i>Calisaya Sánchez Juan Jesús</i>: Contribuyó en la implementación del flujo de validación de entrevistas y en la gestión de reservas desde el frontend móvil, aplicando buenas prácticas en Kotlin y estructura modular. - <i>Hidalgo Lopez Mathias Adriano</i>: Documentó técnicamente la evidencia del Sprint 2, organizando las secciones de ejecución y pruebas en el backend, además de liderar la integración del backend con Postman. 	<p>en clase, plasmando conocimientos de UX, diseño centrado en el usuario y documentación formal en la planificación inicial del proyecto.</p> <p>TP:</p> <p>Cada integrante demostró haber consolidado conocimientos técnicos útiles para su perfil profesional. El trabajo se centró en programación backend, frontend y planificación ágil, usando herramientas reales como NestJS, Angular, Kotlin, y Flutter, sin necesidad de despliegue en nube para validar su progreso.</p> <p>TB2:</p> <p>Durante esta entrega, los integrantes reforzaron su dominio técnico tanto en desarrollo móvil como en backend. El equipo aplicó los conocimientos previos para expandir el alcance funcional del sistema, evidenciando crecimiento en integración de módulos, validaciones funcionales y documentación de evidencias para los distintos apartados del Sprint 2.</p> <p>TF:</p> <p>En la etapa final del proyecto, el equipo consolidó las habilidades técnicas adquiridas, enfocándose en la integración completa del sistema, la validación funcional y la documentación definitiva. Se logró una entrega sólida, evidenciando capacidad de cierre, mejora continua y dominio del ciclo de vida de desarrollo de software.</p>	

Criterio específico	Acciones realizadas	Conclusiones
	<ul style="list-style-type: none"> - <i>Samuel Elias Molina Asencios</i>: Encargado del desarrollo de endpoints adicionales en Node.js, relacionados al ticket y pagos, reforzando su dominio de controladores y rutas. - <i>Diego Ulises Soto Quispe</i>: Implementó nuevas interfaces en Jetpack Compose, asegurando la integración con el backend y manteniendo la consistencia visual y funcional del módulo del conductor. <p>TF:</p> <ul style="list-style-type: none"> - <i>Calisaya Sánchez Juan Jesús</i>: Participó en el desarrollo y validación de nuevas historias de usuario del Sprint 3, enfocadas en el módulo de reservas y mejoras UX. Redactó parte del capítulo 6.2.3 y colaboró en el despliegue de evidencias funcionales. - <i>Hidalgo Lopez Mathias Adriano</i>: Documentó las evidencias del backend y la ejecución del Sprint 3, incluyendo endpoints finales y validaciones técnicas. También revisó la consistencia del informe técnico. - <i>Samuel Elias Molina Asencios</i>: Integró y finalizó los flujos completos de pago y generación de tickets, asegurando el cumplimiento de requisitos del sistema. Participó en las pruebas con Postman y ajustes del backend. - <i>Diego Ulises Soto Quispe</i>: Completó las vistas pendientes del conductor en Compose, revisó la integración general con el backend, y lideró la edición final del documento técnico del proyecto. 	
Reconoce la necesidad del aprendizaje permanente para el desempeño profesional y el desarrollo de proyectos en soluciones de software.	<p>TB1:</p> <ul style="list-style-type: none"> - <i>Calisaya Sánchez Juan Jesús</i>: Estudió técnicas de presentación y redacción para aplicarlas al informe. - <i>Hidalgo Lopez Mathias Adriano</i>: Investigó metodologías centradas en el usuario y cómo estructurar entrevistas. - <i>Samuel Elias Molina Asencios</i>: Aprendió por cuenta propia sobre diseño de experiencias y tareas del usuario. - <i>Frezzia Eldaa Isabel Espinoza Paredes</i>: Reforzó su comprensión del levantamiento de información a través de entrevistas. - <i>Diego Ulises Soto Quispe</i>: Investigó estructuras y patrones de informes técnicos para liderar la coherencia documental. <p>TP:</p> <ul style="list-style-type: none"> - <i>Calisaya Sánchez Juan Jesús</i>: Fortaleció su 	<p>TB1:</p> <p>El equipo evidenció un compromiso claro con el aprendizaje autónomo al investigar y aplicar conceptos nuevos por cuenta propia. Cada miembro se adaptó al enfoque UX y herramientas necesarias en las primeras fases del proyecto.</p> <p>TP:</p> <p>Se reafirmó el rol del aprendizaje continuo en el desarrollo profesional. Todos los integrantes investigaron más allá del contenido visto en clase, aprendiendo nuevas tecnologías, documentando con estándares reales y adaptándose a roles técnicos clave para sacar adelante el producto sin depender de despliegues externos.</p> <p>TB2:</p> <p>El equipo continuó su camino de aprendizaje autónomo, adaptando conocimientos a requerimientos reales del producto. La autogestión y la búsqueda de soluciones técnicas viables marcaron la diferencia en el avance del proyecto, consolidando habilidades valiosas para el entorno profesional.</p>

Criterio específico	Acciones realizadas	Conclusiones
	<p>experiencia técnica aprendiendo a documentar APIs usando Swagger y a colaborar en entornos backend reales.</p> <ul style="list-style-type: none"> - <i>Hidalgo Lopez Mathias Adriano</i>: Investigó por su cuenta cómo estructurar correctamente las evidencias del backend, reflejando compromiso con la calidad del informe técnico. - <i>Samuel Elias Molina Asencios</i>: Aprendió de forma autónoma cómo implementar APIs escalables, trabajando con arquitectura modular y buenas prácticas de desarrollo. - <i>Frezzia Elida Isabel Espinoza Paredes</i>: Aprendió sobre la dinámica ágil de trabajo y los entregables formales del proyecto, con foco en la planificación y organización. - <i>Diego Ulises Soto Quispe</i>: Se capacitó de forma autónoma en frameworks frontend y mobile, construyendo interfaces desde cero con un enfoque autodidacta. <p>TB2:</p> <ul style="list-style-type: none"> - <i>Calisaya Sánchez Juan Jesús</i>: Profundizó su conocimiento en estructuración de entrevistas y validación de hipótesis de usuario mediante criterios heurísticos aplicados al diseño. - <i>Hidalgo Lopez Mathias Adriano</i>: Perfeccionó sus habilidades de documentación Markdown para backend, adaptándose a los requerimientos de Sprint 2 con formato técnico claro. - <i>Samuel Elias Molina Asencios</i>: Investigó sobre integración de pagos en Node.js y aplicó estos aprendizajes al flujo de emisión de tickets del sistema. - <i>Diego Ulises Soto Quispe</i>: Aprendió nuevos patrones de diseño en Compose para adaptar los wireframes al flujo funcional real de la aplicación nativa Android. <p>TF:</p> <ul style="list-style-type: none"> - <i>Calisaya Sánchez Juan Jesús</i>: Profundizó sus conocimientos sobre usabilidad e interacción en apps móviles, aplicando criterios UX mejorados al desarrollo del último sprint. - <i>Hidalgo Lopez Mathias Adriano</i>: Investigó mejores prácticas para refinar endpoints y documentación técnica en Markdown para el cierre del proyecto. - <i>Samuel Elias Molina Asencios</i>: Estudió e implementó medidas de validación de datos y robustez de API para garantizar la estabilidad del producto en producción. - <i>Diego Ulises Soto Quispe</i>: Se enfocó en 	<p>TF:</p> <p>En la fase final, el equipo demostró madurez en el aprendizaje continuo, asumiendo nuevos retos técnicos y documentales. La autoformación fue clave para lograr una entrega completa y profesional, aplicando mejoras finales con base en estándares reales del desarrollo de software.</p>

Criterio específico	Acciones realizadas	Conclusiones
	refinar los flujos de UI/UX mediante investigación de guías oficiales de Compose, mejorando el diseño de la app del conductor para su despliegue.	

Objetivos SMART

A continuación, cada miembro del equipo formula dos objetivos SMART que reflejan su plan de desarrollo profesional continuo una vez finalizada su carrera. Estos objetivos son **específicos, medibles, alcanzables, relevantes y con un tiempo definido**, lo que permite guiar su crecimiento profesional tras la graduación.

Integrante	Objetivo SMART 1	Objetivo SMART 2
Calisaya Sánchez Juan Jesús	Obtener una certificación profesional en desarrollo frontend (Google, Microsoft, etc.) en los primeros 6 meses tras egresar.	Ingresar como frontend developer en una startup de salud digital antes de cumplir 1 año de egresado.
Hidalgo Lopez Mathias Adriano	Publicar una aplicación móvil funcional en Google Play en los primeros 12 meses tras egresar.	Participar en al menos dos eventos tecnológicos (hackathons o ferias) antes de los 12 meses de egresado.
Samuel Elias Molina Asencios	Finalizar un diplomado en UX/UI o diseño centrado en el usuario dentro de los 8 meses posteriores a la carrera.	Diseñar un portafolio profesional digital con al menos 5 casos reales antes de cumplir 1 año.
Frezzia Eldaa Isabel Espinoza Paredes	Iniciar una maestría.	Finalizar la maestría.
Diego Ulises Soto Quispe	Iniciar una maestría en ciberseguridad o arquitectura de software dentro de los primeros 18 meses de egresado.	Liderar un proyecto tecnológico completo en una empresa o startup dentro de los primeros 2 años post-egreso.

Capítulo I: Introducción

1.1. StartUp Profile

En esta sección describiremos la descripción de nuestra Startup.

1.1.1. Descripción de la Startup

PARKINGNOW constituye una empresa emergente del sector tecnológico orientada al desarrollo de soluciones integrales para la **gestión de estacionamientos en tiempo real**. Nuestra organización ha implementado una plataforma digital que posibilita a los usuarios la **reserva y el pago de espacios de estacionamiento**, haciendo uso de tecnologías de geolocalización y **cámaras de monitoreo**, lo que permite una administración eficiente y automatizada de los espacios.

El modelo de negocio adoptado por **PARKINGNOW** se caracteriza por su **independencia y sostenibilidad**, permitiendo atender tanto a usuarios habituales como esporádicos mediante un **sistema transaccional flexible**.

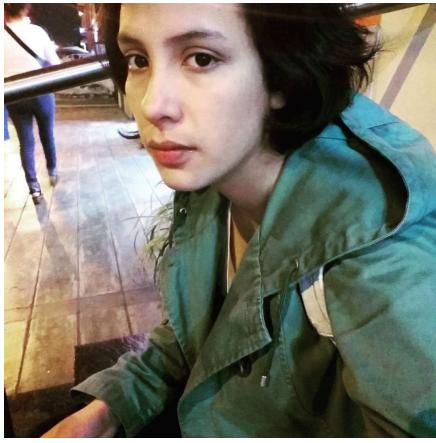
La propuesta innovadora de **PARKINGNOW** reside en la integración de **información en tiempo real**, captada a través de **cámaras instaladas en los locales**, con el objetivo de asegurar la **disponibilidad actualizada de espacios** y **optimizar las rutas** hacia los estacionamientos más cercanos. Las cámaras permiten verificar visualmente el estado de ocupación y actualizar la plataforma de manera inmediata, lo cual contribuye a **incrementar la eficiencia operativa y mejorar la experiencia del usuario**.

Misión: Nuestra misión es **fungir como nexo entre conductores y administradores de estacionamientos**, optimizando la experiencia de aparcamiento en tiempo real. **PARKINGNOW** facilita el acceso a espacios mediante una **gestión automatizada y eficiente** basada en el uso de **cámaras de monitoreo**, que incrementan la rentabilidad para los gestores y satisfacen las necesidades de ambos segmentos del mercado.

Visión: Aspiramos a consolidarnos, en un horizonte de cinco años, como la **plataforma líder en la administración de estacionamientos urbanos**, contribuyendo a la mejora de la **movilidad en las ciudades** a través de soluciones tecnológicas innovadoras y sostenibles, como el uso de **cámaras**

inteligentes, en beneficio tanto de los conductores como de los titulares de los espacios de estacionamiento.

1.1.2. Perfiles de integrantes del equipo

Integrantes	Descripción	Conocimientos
 Diego Ulises Soto Quispe	Estudiante de 7.º ciclo en Ingeniería de Software. Persona responsable, puntual y comprometida. Disfruta jugar básquet, salir a pasear, jugar videojuegos y manejar bicicleta.	HTML, CSS, JavaScript, Angular, Vue.js, Python, C++, SQL, MongoDB, MySQL, NoSQL
 Frezzia Espinoza Paredes	Estudia Ingeniería de Software como segunda carrera en la UPC.	C++, HTML, CSS, JavaScript, Python, redes
 Mathias Adriano Hidalgo Lopez	Estudiante del 7.º ciclo de Ingeniería de Software. Posee habilidades blandas que favorecen el trabajo en equipo y la organización.	HTML, CSS, JavaScript, Angular, Vue.js, Python, C++, SQL, MongoDB, MySQL

Integrantes	Descripción	Conocimientos
 Samuel Elias Molina Asencios	<p>Estudiante del 7.º ciclo de Ingeniería de Software. Habil en resolución de conflictos y negociación. Disfruta nadar y tocar piano.</p>	HTML, CSS, JavaScript, Angular, Vue.js, Python, C++, SQL, MongoDB, MySQL
 Juan Jesús Calisaya Sánchez	<p>Soy estudiante del 7.º ciclo de Ingeniería de Software (código: U202121935). Me considero una persona empática, colaboradora, responsable y con buenas habilidades de comunicación. Además de mis capacidades técnicas, estas cualidades me permitirán contribuir de manera efectiva al desarrollo exitoso del proyecto PARKINGNOW.</p>	Metodologías ágiles, Java, TypeScript, Python, C++, C#, PHP, Dart, Bash, Vue, Angular, SQL, GitFlow, etc..

1.2. Solution Profile

PARKINGNOW representa una propuesta tecnológica de vanguardia orientada a revolucionar la experiencia de estacionamiento en entornos urbanos. La solución consiste en una **plataforma digital** que permite a los usuarios **localizar, reservar y efectuar pagos** por espacios de estacionamiento en tiempo real, integrando funcionalidades de **geolocalización y monitoreo visual mediante cámaras**.

Esta tecnología permite la **detección automática de la presencia de vehículos** a través del análisis visual de las cámaras instaladas, lo que habilita la **actualización instantánea de la disponibilidad de espacios** y guía a los conductores hacia los estacionamientos más cercanos, optimizando sus trayectos y reduciendo significativamente el tiempo de búsqueda.

De este modo, **PARKINGNOW** proporciona **información precisa y en tiempo real**, mejorando sustancialmente la experiencia de los usuarios y ofreciendo una **solución integral** a los desafíos del estacionamiento urbano, en beneficio tanto de los conductores como de los administradores de estos espacios.

1.2.1. Antecedentes y problemática

En las ciudades con un elevado flujo vehicular, la ocupación indebida de espacios, la invasión de áreas peatonales y la obstrucción del tránsito son fenómenos recurrentes. Los conductores enfrentan serias dificultades para encontrar espacios de estacionamiento, especialmente en zonas urbanas congestionadas, donde la demanda supera ampliamente a la oferta y la información disponible suele ser insuficiente o desactualizada. La falta de herramientas que permitan la reserva anticipada de espacios incrementa la frustración y prolonga el tiempo invertido en la búsqueda, afectando negativamente la movilidad y la seguridad urbana.

Con el objetivo de comprender en profundidad las necesidades reales de los usuarios y en consonancia con los principios de **Lean UX**, se realizó un análisis de antecedentes y problemática empleando la metodología “**5W's & 2H's**”, la cual permite identificar oportunidades de mejora a través de un enfoque centrado en el usuario. A continuación, se presentan los hallazgos:

- **What (¿Qué?)**

El principal desafío que aborda **PARKINGNOW** es la escasez de espacios de estacionamiento adecuados. Esta situación induce a muchos conductores a estacionarse en lugares no permitidos, lo cual genera congestión, sanciones y riesgos en la seguridad vial. La falta de tecnologías avanzadas, como **sistemas de monitoreo por cámaras**, limita la gestión eficiente y en tiempo real de la disponibilidad de espacios.

- **When (¿Cuándo?)**

El problema se intensifica durante las horas pico o en momentos de alta demanda, cuando los conductores requieren espacios disponibles que se

ajusten a sus expectativas. En estos contextos, los usuarios recurren a **PARKINGNOW** para consultar disponibilidad, reservar espacios, comparar tarifas y asegurarse de elegir un estacionamiento adecuado, gracias a la información proporcionada en tiempo real.

- **Where (¿Dónde?)**

La problemática se presenta principalmente en **zonas urbanas densamente pobladas**, donde la oferta limitada de estacionamientos y la ausencia de sistemas automatizados dificultan el acceso a información precisa, lo que empeora la experiencia del conductor.

- **Who (¿Quiénes?)**

Los actores involucrados son, por un lado, los conductores que buscan estacionamientos seguros, accesibles y eficientes; y por otro lado, los administradores de locales de estacionamiento interesados en **maximizar la ocupación** de sus espacios mediante el uso de **tecnología basada en cámaras de monitoreo**. El problema afecta a todo tipo de conductores, independientemente de su perfil.

- **Why (¿Por qué?)**

La raíz del problema se encuentra en la ausencia de soluciones tecnológicas en los estacionamientos tradicionales, lo que impide una **comunicación en tiempo real sobre la disponibilidad** de espacios. Esto genera descoordinación entre la oferta y la demanda, elevando los niveles de congestión y estrés en los conductores.

- **How (¿Cómo?)**

Los usuarios pueden utilizar **PARKINGNOW** en diversos contextos: al planificar un trayecto, buscar un estacionamiento que se adapte a sus necesidades o durante eventos masivos. La plataforma proporciona información en tiempo real gracias a su integración con **cámaras inteligentes**, optimizando la experiencia del usuario. La solución se promociona a través de campañas en redes sociales, destacando los beneficios de una **gestión moderna, automatizada y eficiente**.

- **How much (¿Cuánto?)**

La magnitud del problema es considerable. Se refleja en el alto número de infracciones, incidentes y robos asociados al estacionamiento informal o indebido. La falta de sistemas de detección visual, como **cámaras**, agrava la situación y afecta a una amplia variedad de usuarios.

En síntesis, **PARKINGNOW** se posiciona como una **solución integral y tecnológica** frente a los desafíos del estacionamiento urbano. Su implementación contribuye a **facilitar la movilidad, reducir la congestión y mejorar la seguridad**, mediante un sistema de gestión de espacios que responde a las necesidades tanto de los conductores como de los administradores.

1.2.2. Lean UX Process

En esta sección se aplica la metodología **Lean UX**, cuyo alcance incluye la **conceptualización del modelo de negocio** que sustentará el desarrollo del producto digital propuesto. Este enfoque ágil permite centrar los esfuerzos del equipo en el **diseño de una solución enfocada en el usuario**, alineada con las problemáticas detectadas mediante técnicas de **Design Thinking** y validación temprana.

1.2.2.1. Lean UX Problem Statements

A continuación, se utiliza el marco de **Business Opportunity Statements** (Goethelf, 2022) con el propósito de guiar al equipo hacia la creación de un servicio **flexible, innovador y adaptable**. Esta herramienta permite identificar claramente las oportunidades de mejora, priorizar las necesidades del cliente y estructurar la implementación de manera estratégica, maximizando el valor entregado.

Plantilla de Business Opportunity Statement:

El estado actual de [domain] se ha enfocado principalmente en [segments, pain points, etc.]

Lo que los productos/servicios actuales no logran abordar es [this gap].

Nuestro producto/servicio resolverá esta brecha mediante [vision/strategy].

Nuestro enfoque inicial será [this segment].

Sabremos que hemos tenido éxito cuando observemos [these behaviors in our target audience].

Declaración de Problemas Lean UX para PARKINGNOW

- **El estado actual del dominio de estacionamientos urbanos** se ha enfocado principalmente en soluciones convencionales, sin mecanismos inteligentes que brinden información precisa y oportuna a los conductores, lo cual genera demoras, frustración y congestión vial.
- **Lo que los servicios actuales no logran abordar** es la falta de herramientas tecnológicas que permitan visualizar en tiempo real la disponibilidad de espacios y guiar eficientemente a los usuarios hacia los mismos.
- **Nuestra solución resolverá esta brecha** mediante una **plataforma digital avanzada** que utiliza **cámaras instaladas en los locales** para monitorear continuamente la ocupación de espacios y proporcionar datos confiables, integrados con un sistema inteligente de navegación.
- **Nuestro enfoque inicial** será dirigido a **conductores que transitan por zonas metropolitanas con alta densidad vehicular**, donde la demanda de soluciones automatizadas para estacionamiento es crítica.
- **Sabremos que hemos tenido éxito** cuando observemos un **aumento sostenido en el número de reservas efectuadas**, una **disminución del tiempo de búsqueda de estacionamientos** y una mejoría general en la **experiencia del usuario**, medida mediante retroalimentación directa y métricas de uso.

1.2.2.2. Lean UX Assumptions

Users

- Conductores que se desplazan en zonas urbanas con alta congestión y requieren soluciones inmediatas para encontrar estacionamiento disponible.
- Propietarios o administradores de estacionamientos que buscan maximizar el uso de sus espacios y mejorar la comunicación con los usuarios mediante el uso de cámaras inteligentes.
- Empresas logísticas y de transporte que necesitan herramientas eficientes para gestionar el aparcamiento de sus unidades en recorridos urbanos.

User Outcomes

Segmento de Conductores

- Acceder a una plataforma digital que permita buscar, visualizar y reservar espacios de estacionamiento en tiempo real gracias al monitoreo por cámaras.
- Optimizar sus trayectos mediante rutas más directas hacia espacios disponibles, disminuyendo el tiempo de búsqueda.
- Consultar tarifas y disponibilidad desde la aplicación antes de llegar al punto de destino, facilitando la planificación.

Segmento de Propietarios de Estacionamientos

- Aumentar la visibilidad de sus espacios y atraer a una mayor cantidad de conductores mediante una herramienta digital moderna.
- Administrar de forma eficiente las reservas y mantener una comunicación centralizada con los usuarios desde un solo sistema.
- Mejorar la ocupación de los espacios disponibles usando herramientas de análisis que permiten tomar decisiones basadas en datos.

Suposiciones de Negocios

- Existe una demanda creciente por plataformas tecnológicas que faciliten la localización, reserva y pago de estacionamientos en áreas urbanas con alta densidad vehicular.
- Si bien hay soluciones disponibles en el mercado, **PARKINGNOW** se diferencia por ofrecer una experiencia integral basada en información en tiempo real, respaldada por cámaras inteligentes.
- Se prevé que los usuarios valoren la comodidad, transparencia y eficiencia que ofrece la plataforma al permitir una gestión automatizada de todo el proceso de estacionamiento.

Suposiciones de Usuarios

• ¿Quién usa nuestra plataforma?

Está dirigida a conductores que buscan estacionar de forma rápida y segura, y a propietarios de estacionamientos que desean maximizar el rendimiento de sus espacios a través de tecnología moderna.

• ¿Cómo se integra nuestro producto en la rutina diaria?

PARKINGNOW se adapta de manera natural al día a día de los conductores, permitiendo encontrar y reservar estacionamientos con agilidad, especialmente en horarios y zonas de alta demanda.

• ¿Qué problemas resuelve nuestro producto?

Elimina la incertidumbre al ofrecer información confiable y actualizada sobre la disponibilidad de estacionamientos, reduciendo el tiempo perdido y mejorando la experiencia del usuario.

• ¿Qué imagen queremos proyectar con nuestro producto?

Una solución innovadora, eficiente y confiable, que destaque como referente tecnológico en la gestión urbana de estacionamientos.

• ¿Cuál es el objetivo principal de nuestra aplicación?

Simplificar el proceso de búsqueda, reserva y gestión de estacionamientos tanto para usuarios como para operadores, todo desde una única plataforma.

• ¿Qué funcionalidades son clave en nuestra aplicación?

Búsqueda avanzada con geolocalización, visualización en tiempo real con cámaras, comparación de precios, reservas en línea y herramientas analíticas para mejorar la toma de decisiones.

1.2.2.3. Lean UX Hypothesis Statements

Para formular las Hypothesis Statements, hemos seguido el formato de Lean UX: [We believe that], [This will achieve], y [We will have demonstrated this when].

Hipótesis 1

Creemos que la mejora del proceso de búsqueda y reserva de espacios mediante el uso de cámaras inteligentes motivará a los usuarios a utilizar la plataforma de forma continua.

Esto permitirá alcanzar un incremento del 30 % en la cantidad de usuarios recurrentes durante los primeros seis meses posteriores a la implementación. Lo validaremos cuando se registre un aumento del 30 % en el número de reservas efectuadas por usuarios que ya han usado previamente la aplicación.

Hipótesis 2

Creemos que el uso de notificaciones en tiempo real, basadas en la información proporcionada por cámaras, reducirá el tiempo promedio que los conductores dedican a encontrar estacionamiento.

Esto permitirá disminuir en un 20 % el tiempo invertido en la búsqueda de espacios disponibles.

Lo validaremos cuando los reportes y registros del sistema evidencien una reducción medible cercana al 20 % en los tiempos de búsqueda.

Hipótesis 3

Creemos que la incorporación de descuentos y beneficios exclusivos para usuarios frecuentes aumentará la fidelización y promoverá la conversión de usuarios ocasionales en regulares.

Esto facilitará que al menos un 25 % de los usuarios esporádicos comiencen a utilizar la plataforma de forma mensual.

Lo validaremos cuando se observe un crecimiento del 25 % en la proporción de usuarios que acceden recurrentemente a la plataforma cada mes.

Hipótesis 4

Creemos que el uso de algoritmos de optimización de rutas, alimentados por datos recogidos a través de cámaras, mejorará la experiencia general de los usuarios.

Esto se reflejará en mayores niveles de satisfacción y una reducción en la frustración asociada con el proceso de búsqueda.

Lo validaremos cuando las encuestas muestren un aumento en las puntuaciones de satisfacción y una disminución en los reclamos relacionados con tiempos de espera o falta de información.

Hipótesis 5

Creemos que un sistema de fidelización que ofrezca recompensas por el uso continuo de la plataforma mejorará la retención de clientes y fomentará las recomendaciones.

Esto generará un aumento del 30 % en la retención de usuarios activos, además de un crecimiento sostenido a través del canal de referidos.

Lo validaremos cuando las métricas indiquen un incremento del 30 % en la retención y una subida significativa en los registros provenientes de recomendaciones de otros usuarios.

1.2.2.4. Lean UX Canvas

El Lean UX Canvas desarrollado para **PARKINGNOW** permite identificar los principales desafíos y oportunidades del servicio, centrado en las necesidades tanto de los conductores como de los propietarios de estacionamientos.

Problemáticas identificadas:

- Para conductores: Dificultades constantes para localizar espacios de estacionamiento disponibles, especialmente en zonas urbanas con alta congestión.
- Para propietarios de estacionamientos: Baja ocupación de los espacios disponibles y procesos de gestión poco eficientes debido a la falta de digitalización.

Soluciones planteadas:

- Implementación de cámaras inteligentes para el monitoreo visual en tiempo real del estado de ocupación de los espacios.
- Sistema de reservas anticipadas integrado con geolocalización, permitiendo a los usuarios encontrar y asegurar un espacio cercano desde sus dispositivos.

Métricas de validación:

- Indicadores cuantitativos: Nivel de ocupación de espacios, tiempo promedio de búsqueda de estacionamiento, número de reservas generadas.
- Indicadores cualitativos: Nivel de satisfacción del usuario, percepción de mejora en la experiencia de estacionamiento, reducción de reclamos.

Figura 1. Lean UX Canvas aplicado al proyecto PARKINGNOW

Lean UX Canvas

Title: PARKINGNOW LEAN UX Canvas

Date: 25/04/25

Iteration:1

Business Problem	Solution ideas	Business Outcomes
Business Problem What business have you identified that needs help? Conductores en áreas urbanas densamente pobladas enfrentan dificultades para identificar estacionamientos disponibles en tiempo real, generando frustración, pérdida de tiempo y estrés. Paralelamente, los operadores de estacionamientos no maximizan sus ingresos debido a la subutilización de espacios y visibilidad limitada.	Solution ideas List product, feature, or enhancement ideas that help your target audience achieve the benefits they're seeking. <ul style="list-style-type: none"> Desarrollo de una plataforma digital con sensores infrarrojos para detectar automáticamente la disponibilidad de espacios. Implementación de un sistema de reservas anticipadas y notificaciones en tiempo real. Integración de una función de optimización de rutas para dirigir a los conductores al estacionamiento más cercano. 	Business Outcomes (Changes in customer behavior) What changes in customer behavior will indicate you have solved a real problem in a way that adds value to your customers? <ul style="list-style-type: none"> Incremento en la utilización de espacios mediante la integración de sensores infrarrojos para gestión eficiente en tiempo real. Reducción del tiempo promedio de búsqueda de estacionamiento para los conductores. Mejora en la satisfacción del usuario, medida a través de encuestas y Net Promoter Score (NPS). Aumento en la retención de clientes para los propietarios de estacionamientos, mejorando la repetición de uso.
Users & Customers		User Benefits
Users & Customers What types of users and customers should you focus on first? <ul style="list-style-type: none"> Conductores Urbanos: Requieren soluciones rápidas y eficientes para encontrar estacionamiento, valorando servicios que reduzcan el tiempo de búsqueda y ofrezcan seguridad y conveniencia. Propietarios de Estacionamientos: Buscan optimizar la gestión de sus espacios, mejorar la experiencia del cliente y maximizar ingresos mediante tecnología avanzada. Empresas de Transporte y Logística: Demandan soluciones eficientes para estacionar flotas en áreas congestionadas, priorizando la integración con sistemas de planificación de rutas. 		User Benefits What are the goals your users are trying to achieve? What is motivating them to seek out your solution? (e.g., do better at my job OR get a promotion) <ul style="list-style-type: none"> Conductores Urbanos: Método más fácil y rápido para reservar estacionamientos, reduciendo tiempo de búsqueda y estrés. Mayor seguridad y conveniencia mediante actualizaciones en tiempo real. Propietarios de Estacionamientos: Mayor ocupación de sus espacios y mejor experiencia del cliente. Los sensores infrarrojos optimizan la utilización de los espacios y proporcionan datos valiosos para mejorar la gestión operativa y la eficiencia del negocio. Empresas de Transporte: Reducción de tiempos y costos operativos mediante mejor planificación de rutas y gestión eficiente de flotas en áreas congestionadas, utilizando datos en tiempo real.
Hypotheses	What's the most important thing we need to learn first?	What's the least amount of work we need to do to learn the next most important thing?
Hypotheses Combine the assumptions from 2, 3, 4 & 5 into the following template hypothesis statement: <i>"We believe that [business outcome] will be achieved if [user] attains [benefit] with [feature]."</i> <ul style="list-style-type: none"> Si proporcionamos datos en tiempo real sobre la disponibilidad de estacionamientos mediante sensores infrarrojos, los conductores reducirán el tiempo de búsqueda en un 20%. Si permitimos la reserva anticipada de espacios, aumentará la satisfacción del usuario en un 25% dentro de los primeros seis meses de implementación. La optimización de rutas basada en datos en tiempo real mejorará la eficiencia de los conductores en un 15%. 	What's the most important thing we need to learn first? Validar si la provisión de datos en tiempo real sobre la disponibilidad de estacionamientos mediante sensores infrarrojos reduce el tiempo de búsqueda en un 20%. Determinar si la reserva anticipada de espacios aumenta la satisfacción del usuario en un 25% dentro de los primeros seis meses de implementación. Verificar si la optimización de rutas basada en datos en tiempo real mejora la eficiencia de los conductores en un 15%.	What's the least amount of work we need to do to learn the next most important thing? Realizar un piloto con sensores infrarrojos en un estacionamiento de alta demanda para medir precisión de detección y rapidez de actualización en tiempo real. Encuestas a conductores urbanos sobre su interés en la reserva anticipada de espacios y tecnología de optimización. A/B Testing para comparar el tiempo de búsqueda y la satisfacción del usuario entre versiones con y sin sensores.

1.3. Segmentos objetivo

Conductores Urbanos Frecuentes

Demografía:

- Género:** Masculino y femenino
- Rango etario:** Jóvenes adultos (18-34 años), adultos medios (35-54 años) y adultos mayores (55 años a más)
- Nivel socioeconómico:** Clases B, C y D (media-alta, media y media-baja)

Beneficios clave:

- Reducción del tiempo de búsqueda y del nivel de estrés gracias a la posibilidad de realizar reservas en tiempo real.
- Mayor seguridad para el vehículo mediante el monitoreo continuo a través de cámaras instaladas en los espacios de estacionamiento.
- Facilidad para comparar precios, localizar espacios y acceder a información desde una sola aplicación.

Propietarios de Estacionamientos

Demografía:

- Género:** Masculino y femenino
- Rango etario:** Jóvenes adultos (18-34 años), adultos medios (35-54 años) y adultos mayores (55 años a más)
- Nivel socioeconómico:** Clases B y C (media-alta y media)

Beneficios clave:

- Gestión de datos:** Acceso a herramientas analíticas que permiten optimizar la ocupación, ajustar precios dinámicamente y tomar decisiones basadas en evidencia.
- Automatización:** Simplificación de procesos como la asignación de espacios y los cobros digitales, reduciendo la carga operativa.
- Formalización:** Digitalización de las operaciones para incrementar el control administrativo y reducir la informalidad en la gestión de estacionamientos.

Capítulo II: Requirements Elicitation & Analysis

2.1. Competidores

Parkimeter

Parkimeter es una solución digital enfocada en la gestión y reserva de espacios de estacionamiento. Su plataforma facilita a los usuarios la búsqueda, comparación y reserva de plazas en distintas ubicaciones. También ofrece funcionalidades como visualización de tarifas, disponibilidad en tiempo real y otros servicios útiles que simplifican el proceso de estacionamiento para los conductores.

EIParking

EIParking brinda a los usuarios la posibilidad de localizar, reservar y pagar por estacionamientos desde una misma plataforma. Su sistema incorpora datos actualizados sobre espacios disponibles, además de permitir el pago de parquímetros, seguimiento de sanciones de tránsito y otros servicios complementarios que optimizan la experiencia del conductor.

Aparcados

Aparcados es una aplicación enfocada en simplificar la búsqueda de estacionamientos urbanos según las necesidades del usuario. Aunque cuenta con versión web, su versión móvil es la más utilizada debido a su interfaz amigable y diseño visual atractivo, lo que la convierte en una herramienta práctica para conductores en movimiento.

2.1.1. Análisis competitivo

A continuación, se presenta una tabla comparativa de los principales competidores a nivel internacional. Este análisis nos permitirá identificar patrones clave de funcionalidad y experiencia de usuario, con el objetivo de alinear nuestra propuesta a las preferencias reales del público objetivo.

Competitive Analysis Landscape	
¿Por qué llevar a cabo este análisis?	Escriba en el recuadro la pregunta que busca responder o el objetivo de este análisis.
	Identificar cómo PARKINGNOW se posiciona frente a sus competidores en términos de servicios ofrecidos, ventajas competitivas y estrategias de mercado para mejorar nuestra propuesta de valor y captar más clientes.

Competidor		ParkingNow	Parkimeter	EIParking	Aparcados
Logo					
Perfil	Overview	Es una aplicación web que brinda la posibilidad de reservar espacios de estacionamiento en múltiples ciudades, además de conectar a consumidores con propietarios de estacionamientos urbanos.	Es una aplicación web y móvil que ofrece servicios de reserva de plazas de aparcamiento en diversas ciudades de Europa y América Latina.	Es una aplicación web y móvil que ofrece servicios de reserva de plazas de aparcamiento en diversas ciudades de Europa y América Latina.	Es una aplicación web y móvil que ofrece servicios de reserva de plazas de aparcamiento en diversas ciudades de España principalmente, pero expandiéndose alrededor del mundo.
	Ventaja competitiva ¿Qué valor ofrece a los clientes?	Cancelación gratuita hasta con 24 horas de antelación de la reserva, inclusión de monitoreo por cámaras para los	Ofrece descuentos de hasta el 70% y no cobra gastos de gestión.	Permite más que solo pagar estacionamientos, sino gestionar multas y pagar parquímetros.	La app incluye geolocalización y ofrece notificaciones en tiempo real.

Competidor		ParkingNow	Parkimeter	EIParking	Aparcados
		vehículos y conexión directa entre consumidores y propietarios de estacionamientos urbanos.			
Perfil de Marketing	Mercado objetivo	Personas que buscan soluciones de estacionamiento en áreas urbanas congestionadas y propietarios de estacionamientos urbanos.	Personas que buscan soluciones de estacionamiento en áreas urbanas congestionadas.	Personas que buscan soluciones de estacionamiento en áreas urbanas congestionadas.	Personas que buscan soluciones de estacionamiento en áreas urbanas congestionadas.
	Estrategias de marketing	Utiliza publicidad en línea a través de plataformas y redes sociales.	Utiliza publicidad en línea a través de plataformas y redes sociales.	Utiliza publicidad en línea, redes sociales y reciben promociones de influencers.	Utiliza publicidad en línea a través de plataformas y redes sociales.
Perfil de Producto	Productos & Servicios	Ofrece una serie de servicios relacionados con la reserva de plazas de aparcamiento en áreas urbanas y otros destinos.	Realizar servicios de parking en todo el territorio español y otras grandes ciudades del mundo.	Ofrece una serie de servicios relacionados con la reserva de plazas de aparcamiento en ciudades concurridas, seguimiento de multas y pagos sencillos.	Ofrece una serie de servicios relacionados con la reserva de plazas de aparcamiento en áreas urbanas y otros destinos.
	Precios & Costos	La plataforma es gratuita, pero se debe pagar las tarifas de estacionamiento correspondientes.	La plataforma es gratuita, pero se deben pagar las tarifas de estacionamiento correspondientes y las suscripciones VIP.	La plataforma es gratuita, pero se deben pagar las tarifas de estacionamiento, además de la existencia de tarifas especiales o descuentos.	La plataforma es gratuita, pero se debe pagar las tarifas de estacionamiento correspondientes.
	Canales de distribución (Web y/o Móvil)	Web	Web/Móvil	Web/Móvil	Web/Móvil
Análisis SWOT	Fortalezas	Implementa monitoreo visual mediante cámaras en los estacionamientos asociados, ofreciendo un objetivo más amplio que otras aplicaciones y soporte en tiempo real para los consumidores.	Parkimeter opera en múltiples ciudades de Europa y América Latina.	EIParking implementa seguimiento de multas y pagos aparte de los estacionamientos.	Aparcados opera en múltiples ciudades de Europa y Asia.
	Debilidades	La industria de reserva de estacionamientos en línea es muy competitiva.	Varias empresas que proporcionan servicios similares.	La industria de reserva de estacionamientos en línea es muy competitiva.	Varias empresas que proporcionan servicios similares.
	Oportunidades	A medida que las ciudades sigan creciendo, la demanda de servicios	La congestión del tráfico se vuelve más común, provocando	En su país de origen, España, las personas en zonas urbanas	Tiene grandes posibilidades de expandir su mercado,

Competidor		ParkingNow	Parkimeter	EIParking	Aparcados
		de reserva podría aumentar.	una alta demanda en la reserva de estacionamientos.	deberán aprender más acerca de la app para vivir en comodidad.	enfocándose en la reserva de estacionamientos en América Latina.
	Amenazas	Cambios en las políticas de movilidad urbana.	Nuevas startups compitiendo por la misma base de usuarios.	Nuevas startups compitiendo por la misma base de usuarios.	Nuevas startups compitiendo por la misma base de usuarios.

2.1.2. Estrategias y tácticas frente a competidores

A continuación, encontraremos la tabla de Matriz de FODA y C.A.M.E para poder analizar nuestras estrategias a futuro para competir contra los competidores anteriormente mencionados.

MATRIZ FODA y C.A.M.E	Oportunidades: La demanda de este servicio aumenta bastante con los años	Amenazas: Cambios en la política de movilidad urbana
Fortalezas: Innovación, público objetivo más amplio y soporte en tiempo real para los consumidores	Ofrecer soporte eficiente a través de la app para fidelizar usuarios, destacando el valor diferenciador de los sensores inteligentes como una funcionalidad poco común y atractiva para el mercado.	Realizar un monitoreo constante de las políticas de movilidad urbana para adaptar nuestra propuesta a los cambios y mantenernos siempre alineados con la normativa.
Debilidades: Mercado competitivo y nuevo	Fortalecer las campañas de difusión y visibilidad digital, resaltando las características innovadoras de la app como su enfoque en tiempo real y personalización para lograr una mejor recepción en el mercado.	Desarrollar mensajes de marketing enfocados en cómo nuestra solución contribuye a una mejor gestión urbana, mostrando que no se opone a las políticas sino que las complementa.

2.2. Entrevistas

Esta sección se centra en la recopilación de información mediante entrevistas a representantes de los segmentos objetivo, con el fin de entender sus necesidades y expectativas en relación al servicio.

2.2.1. Diseño de entrevistas

Incluye la elaboración de preguntas principales y complementarias dirigidas a cada segmento, considerando aspectos demográficos, características de personalidad y preferencias tecnológicas para construir perfiles precisos de los usuarios.

Segmento objetivo #1: Conductores Urbanos Frecuentes

Preguntas demográficas:

- ¿Cuál es tu nombre?
- ¿Cuál es tu edad?
- ¿Dónde vives?
- ¿A qué te dedicas?
- ¿Cuáles consideras que son tus principales habilidades?

Preguntas principales:

- ¿Con qué frecuencia necesitas estacionamiento en la ciudad?
- ¿Cuánto tiempo dedicas normalmente a buscar estacionamiento?
- ¿Utilizas actualmente alguna aplicación o servicio para encontrar estacionamiento?
- ¿Qué tan importante es para ti conocer la disponibilidad de estacionamiento en tiempo real?
- ¿Estarías dispuesto/a a pagar una tarifa adicional por garantizar un espacio de estacionamiento reservado?
- ¿Qué consideras más importante al elegir un estacionamiento: costo, proximidad, o seguridad?
- ¿Te gustaría recibir recomendaciones de rutas hacia el estacionamiento más cercano basado en el tráfico en tiempo real?

Preguntas complementarias:

- ¿Has tenido experiencias frustrantes buscando estacionamiento en la ciudad?

- ¿Qué mejoras te gustaría ver en los servicios de estacionamiento que utilizas actualmente?

Segmento objetivo #2: Dueños de estacionamientos

Preguntas demográficas:

- ¿Cuál es tu nombre?
- ¿Cuál es tu edad?
- ¿Dónde vives?
- ¿A qué te dedicas?
- ¿Cuáles consideras que son tus principales habilidades?

Preguntas principales:

- ¿Consideras sencillo el proceso de gestionamiento de espacios en tu estacionamiento?
- ¿Qué tan importante es para ti contar con información en tiempo real sobre los espacios disponibles?
- ¿Utilizas algún sistema digital para administrar los pagos y reservas de los espacios de estacionamiento?
- ¿Qué beneficios ves en la posibilidad de automatizar el proceso de reserva y pago de los espacios de estacionamiento?
- ¿Qué tan relevante es para ti ofrecer opciones de pago seguras y rápidas a tus clientes?
- ¿De qué manera crees que la integración de una app que gestione datos y pagos podría mejorar la eficiencia de tu negocio?

Preguntas complementarias:

- ¿Cuáles son los principales desafíos que enfrentas al administrar tu estacionamiento?
- ¿Qué funcionalidades adicionales te gustaría que una app ofreciera para facilitar la gestión de tu negocio?
- ¿Te gustaría recibir reportes o análisis sobre la ocupación y rotación de vehículos en tu estacionamiento?

2.2.2. Registro de entrevistas

Segmento objetivo #1: Conductores Urbanos Frecuentes

Entrevistado N°1: Joan Talizo Balbin

Sexo: Masculino

Edad: 23

Ubicación en la que vive: Surco, Lima, Perú



Link de la entrevista: <https://youtu.be/8MtQq3fYY3s>

Resumen: Balbín utiliza su auto todos los días para ir a la universidad y a su trabajo como desarrollador web en una ONG. Esto lo obliga a buscar estacionamiento constantemente, lo cual puede tomarle entre 10 a 20 minutos, dependiendo del tráfico y la zona. Actualmente no usa ninguna aplicación para encontrar estacionamiento, pero considera que una app con información en tiempo real sería muy útil, ya que le ahorraría tiempo y evitaría dar vueltas innecesarias. También le gustaría que la app recomiende rutas según el tráfico, lo cual considera clave para optimizar sus desplazamientos. Está dispuesto a pagar una tarifa adicional por reservar un espacio, siempre que esto le garantice seguridad y cercanía. Aunque valora la seguridad, prioriza el costo y la proximidad al momento de elegir un estacionamiento. Ha vivido situaciones frustrantes, como perder tiempo en el tráfico, no encontrar espacios libres o que otro conductor le gane el lugar. Por ello, valora la posibilidad de saber en tiempo real si hay espacios disponibles e incluso recibir alertas si ocurre algo sospechoso con su vehículo.

Entrevistado N°2: Luis Ángel Anampa Lavado

Sexo: Masculino

Edad: 20

Ubicación en la que vive: San Borja, Lima, Perú



Link de la entrevista: <https://youtu.be/rVE1PmHttmg>

Resumen: Ángel utiliza su vehículo todos los días, por lo que necesita estacionamiento de forma constante. El tiempo que le toma encontrar un espacio varía según la zona y el tráfico, pudiendo tardar entre 10 a 20 minutos o más en horas punta. Actualmente no usa ninguna aplicación para encontrar estacionamiento, pero considera que sería una herramienta muy útil. Destaca que conocer la disponibilidad de espacios en tiempo real es clave, ya que le permitiría ahorrar tiempo y evitar vueltas innecesarias. Está dispuesto a pagar una tarifa adicional por reservar un espacio, siempre y cuando no sea excesiva. Al elegir un estacionamiento, valora principalmente el costo y la proximidad, dejando la seguridad en un segundo plano aunque también le parece importante. Ángel estaría muy interesado en una aplicación que le recomiende rutas hacia estacionamientos disponibles evitando el tráfico. Ha tenido experiencias frustrantes buscando estacionamiento, como perder espacios a último momento o tener que maniobrar en lugares muy estrechos. Finalmente, sugiere como mejoras la posibilidad de recibir notificaciones en tiempo real cuando se libera un espacio cercano, y también alertas de seguridad en caso de movimientos sospechosos cerca del vehículo.

Entrevistado N°3: Rodrigo Ismael Viton

Sexo: Masculino

Edad: 20

Ubicación en la que vive: Chorrillos, Lima, Perú



Link de la entrevista: <https://youtu.be/8k2rWLjlsQ>

Resumen: Rodrigo enfrenta a diario la dificultad de encontrar estacionamiento, lo que afecta su trabajo como repartidor. No usa apps especializadas, pero muestra alta disposición a utilizar una que le garantice disponibilidad en tiempo real, reserva segura y rutas optimizadas. Valora la proximidad y la rapidez. Su experiencia indica una gran oportunidad para crear una app práctica, con pagos digitales y geolocalización inteligente.

Segmento objetivo #2: Dueños de estacionamientos

Entrevistado N°1: Carlos Ponce

Sexo: Masculino

Edad: 20

Ubicación en la que vive: La Molina, Lima, Perú



Link de la entrevista: https://youtu.be/pl_iSFNsPYE

Resumen: Carlos tiene 20 años y vive en La Molina, donde administra un estacionamiento. Se considera una persona organizada, resolutiva y orientada a mantener el orden, aunque reconoce que durante momentos de alta afluencia la gestión puede volverse complicada. Respecto al control de los espacios, menciona que usualmente es sencillo, pero se dificulta cuando hay gran movimiento o los conductores no respetan los espacios asignados. Por esta razón, valora la posibilidad de contar con información en tiempo real sobre la disponibilidad, lo cual evitaría revisiones físicas continuas y mejoraría la

eficiencia operativa. Actualmente no utiliza sistemas digitales de reservas, pero acepta pagos mediante Yape y Plin, métodos que considera ágiles y populares entre sus clientes.

Carlos considera que la automatización de reservas y pagos sería altamente beneficiosa, ya que permitiría organizar mejor los tiempos de permanencia, agilizar los ingresos y salidas, y llevar un control más preciso de los cobros. Resalta la importancia de ofrecer métodos de pago seguros y rápidos para mejorar la experiencia del cliente. Además, está convencido de que una aplicación que integre la gestión de espacios, reservas y pagos facilitaría el control detallado de la ocupación, ingresos diarios y aumentaría la transparencia en su administración. Entre los principales desafíos que enfrenta señala la organización en horarios de alta demanda y el seguimiento manual de los pagos. Finalmente, le gustaría que una app permita a los clientes registrar su tiempo de permanencia, calcule automáticamente el costo, notifique la disponibilidad en tiempo real y genere reportes de ocupación y rotación para tomar decisiones más estratégicas.

Entrevistado N°2: Guillermina Asencio Sarrazar

Sexo: Femenino

Edad: 57

Ubicación en la que vive: Villa El Salvador, Lima, Perú



Link de la entrevista: <https://youtu.be/cuG7FslUnAQ>

Resumen:

Guillermina es ama de casa y también administra un estacionamiento. Considera que la gestión, en general, es sencilla, aunque se complica cuando los clientes incumplen con los pagos o solicitan descuentos injustificados. Actualmente no utiliza ningún sistema digital de gestión, pero le gustaría contar con una plataforma que permita realizar reservas y pagos virtuales anticipados, lo cual considera seguro y eficiente.

Resalta la importancia de ofrecer métodos de pago rápidos y confiables para evitar tiempos de espera innecesarios y garantizar la atención a los usuarios incluso en su ausencia. Entre los principales desafíos que enfrenta se encuentran los retrasos de los clientes al recoger sus vehículos, lo que genera inconvenientes en el flujo de otros usuarios. Guillermina manifiesta su interés en contar con una aplicación que le permita controlar mejor los tiempos de permanencia, recibir pagos anticipados y automatizar todo el proceso de administración de su estacionamiento.

Entrevistada N°3: Alexandra Yamile Teves Salas

Sexo: Femenino

Edad: 21

Ubicación en la que vive: Breña, Lima, Perú



Alexandra Yamile Téves Salas

Link de la entrevista: <https://youtu.be/Cut8KMDgN6A>

Resumen: Alexandra actualmente se encarga del estacionamiento que antes gestionaba su madre. Aunque no cuenta con experiencia previa en administración, ha demostrado habilidades organizativas y está abierta a implementar soluciones tecnológicas. Considera que gestionar el espacio manualmente puede ser complicado, especialmente en horas pico. Le interesa contar con información en tiempo real y automatizar procesos como reservas y pagos, ya que eso facilitaría su trabajo y mejoraría la experiencia del cliente. No utiliza un sistema digital actualmente, pero está dispuesta a adoptar uno que permita recibir pagos electrónicos y genere reportes sobre ocupación y rotación. Reconoce la importancia de opciones de pago seguras y funcionalidades que le ayuden a mejorar el control y eficiencia del negocio.

2.2.3. Análisis de entrevistas

De acuerdo con la información recopilada de las entrevistas, realizamos el siguiente análisis segmentado:

Segmento objetivo #1: Conductores Urbanos Frecuentes

Hallazgos:

- Todos los entrevistados utilizan su vehículo a diario, ya sea por motivos laborales, académicos o de reparto, lo que incrementa su necesidad constante de encontrar estacionamiento.
- El tiempo de búsqueda de un lugar para estacionar varía entre 10 a 20 minutos, especialmente en zonas de alta demanda o en horas pico.
- Ninguno de los entrevistados utiliza actualmente una aplicación para encontrar estacionamiento, pero todos consideran que una app con disponibilidad en tiempo real sería de gran utilidad.
- Existe una alta disposición a pagar una tarifa adicional si esta garantiza la reserva del espacio, especialmente si incluye elementos como seguridad y proximidad.
- Los criterios más valorados al momento de elegir un estacionamiento son el costo, la cercanía y, en menor medida, la seguridad.
- Las frustraciones comunes incluyen perder espacios disponibles, dar vueltas sin éxito y lidiar con zonas congestionadas.
- Se valoran funcionalidades como la sugerencia de rutas optimizadas según el tráfico y notificaciones en tiempo real sobre espacios liberados o posibles riesgos cerca del vehículo.

Segmento objetivo #2: Dueños de estacionamientos

Hallazgos:

- La gestión de espacios no es compleja cuando hay baja demanda, pero en horarios de alta rotación se vuelve difícil mantener el orden y controlar tiempos de permanencia.
- Se considera fundamental contar con información en tiempo real sobre disponibilidad para evitar supervisiones físicas y mejorar la eficiencia.
- Aunque se aceptan pagos digitales (como Yape o Plin), no se utilizan aplicaciones que permitan la reserva previa de espacios.
- La automatización es vista como una solución clave para agilizar cobros, organizar la permanencia de los clientes y reducir errores manuales.
- Ofrecer opciones de pago rápidas y seguras es una prioridad, ya que mejora la experiencia del cliente.
- Se valora el acceso a información básica del cliente (nombre, número de contacto) para reforzar la seguridad en el servicio.
- El interés en reportes de ocupación y análisis de rotación es alto, ya que ayudarían a tomar mejores decisiones operativas.

Conclusiones generales

Los hallazgos del **segmento 1** reflejan una necesidad clara de modernizar la experiencia de estacionamiento mediante tecnología. Aunque los usuarios no usan apps actualmente, muestran apertura total a una solución que les ofrezca disponibilidad en tiempo real, reserva anticipada, rutas optimizadas y alertas inteligentes. Además, están dispuestos a pagar si el servicio resuelve sus principales frustraciones: pérdida de tiempo y falta de seguridad.

En el **segmento 2**, los dueños de estacionamientos coinciden en la utilidad de digitalizar la gestión. Aunque algunos procesos se manejan de forma sencilla, hay oportunidades de mejora al automatizar cobros, reservas y notificaciones. La app ideal, para ellos, debería ofrecer herramientas que faciliten la organización diaria, aumenten la seguridad y ofrezcan estadísticas útiles sobre la ocupación.

Ambos segmentos coinciden en que una aplicación móvil que integre información en tiempo real, pagos digitales y comunicación directa generaría una experiencia más eficiente, confiable y satisfactoria.

Entrevista completa:

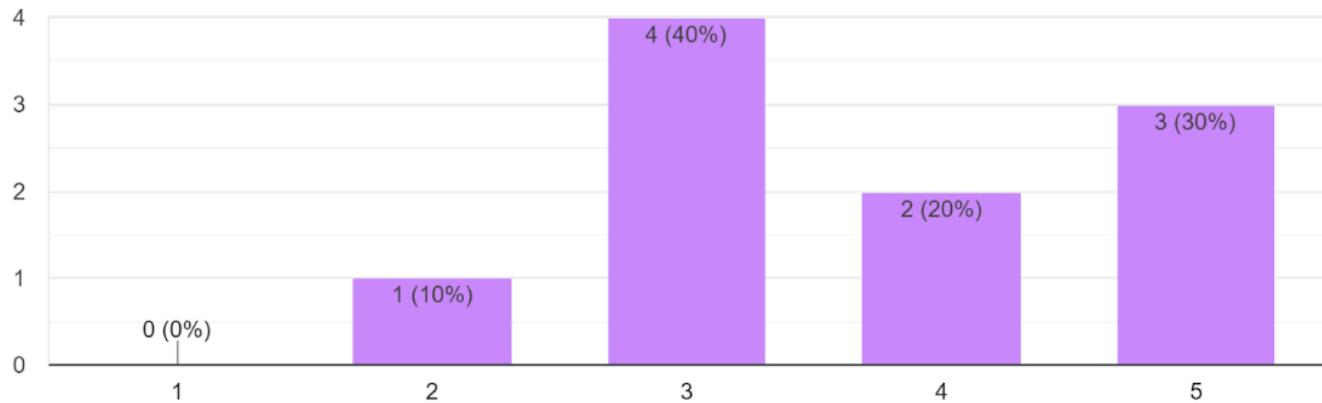
Link de las entrevistas completas: https://upcedupe-my.sharepoint.com/:v/g/personal/u202214477_upc_edu_pe/EQfuOaavJTVCi5jKvTLzMU0Bixb6cO1iHY4X7yZitnQhUw?e=UaqlvS&nav=eyJyZWZlcnJhbEluZm8iOnsicmVmZXJyYWxBcHAiOiTdHJIYW1XZWJBcHAiLCJyZWZlcnJhbFZpZXciOiJTaGFyZURpYWxvZy1MaW5rliwicVmZXJyYWxBcHBQbGF0Zm9ybSI6IldlYilsInJlZmVycmFsTW9kZSI6InZpZXcifX0%3D

Análisis por medio de herramientas estadística:

Segmento objetivo 1: <https://forms.gle/Lo3Z69GnkyYDcaAYA>

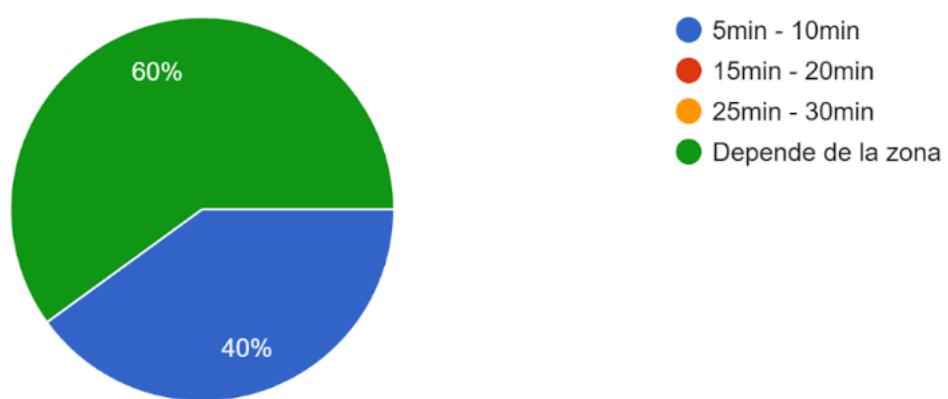
¿Con qué frecuencia necesitas estacionamiento en la ciudad?

10 responses



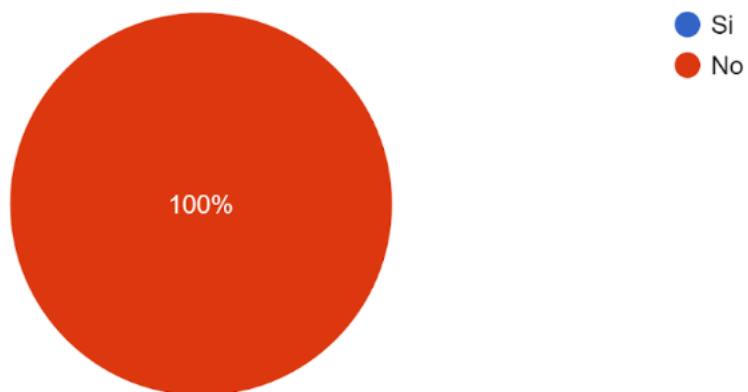
¿Cuánto tiempo dedicas normalmente a buscar estacionamiento?

10 responses



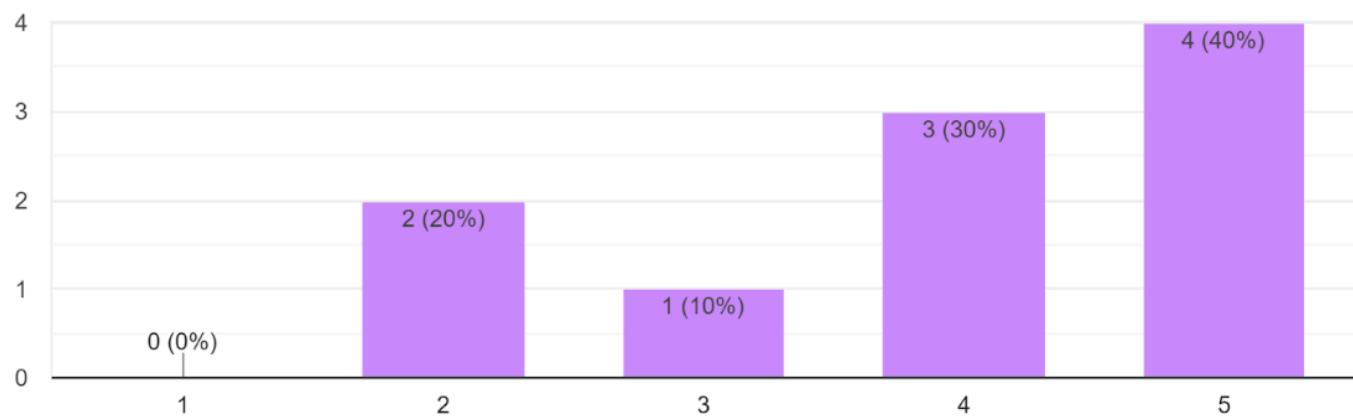
¿Utilizas actualmente alguna aplicación o servicio para encontrar estacionamiento?

10 responses



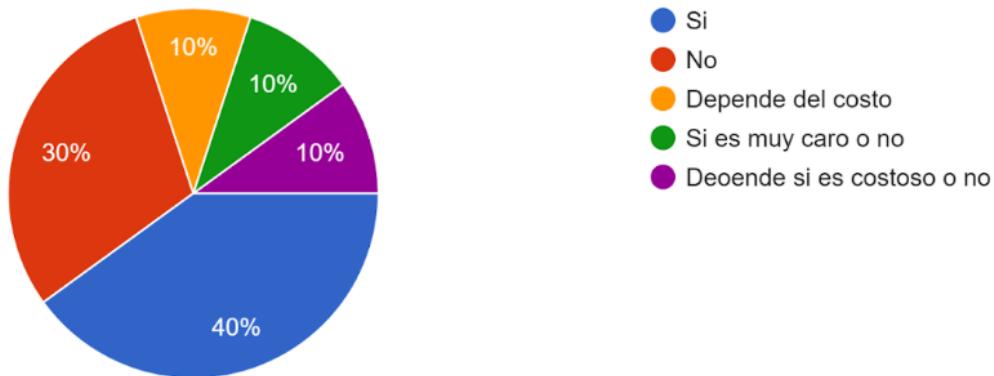
¿Qué tan importante es para ti conocer la disponibilidad de estacionamiento en tiempo real?

10 responses



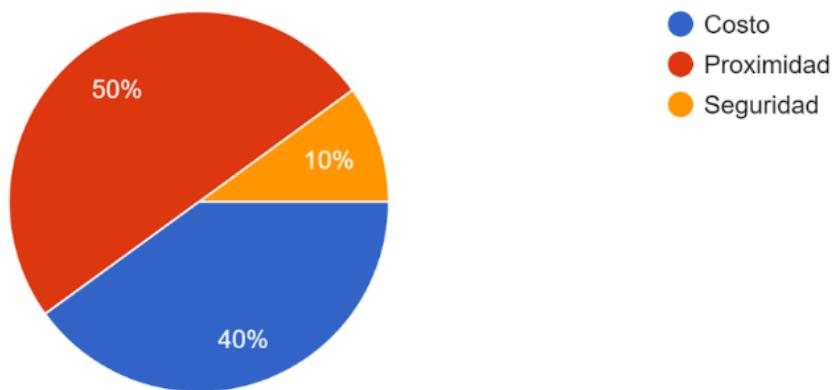
¿Estarías dispuesto/a a pagar una tarifa adicional por garantizar un espacio de estacionamiento reservado?

10 responses



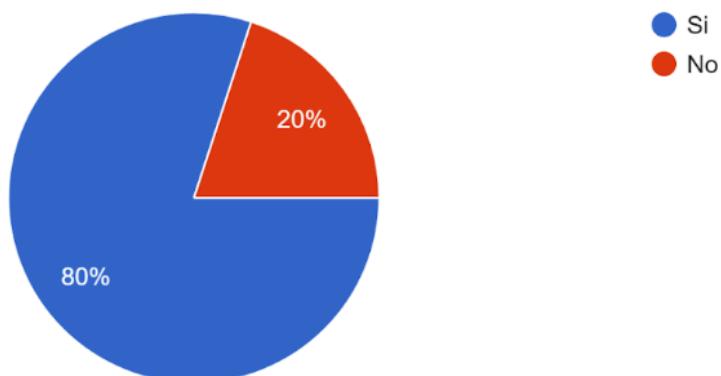
¿Qué consideras más importante al elegir un estacionamiento: costo, proximidad, o seguridad?

10 responses



¿Te gustaría recibir recomendaciones de rutas hacia el estacionamiento más cercano basado en el tráfico en tiempo real?

10 responses



¿Has tenido experiencias frustrantes buscando estacionamiento en la ciudad?

10 responses

Si, muchas veces

Si pero muy pocas veces

Muchas veces

No realmente, solo falta de estacionamiento en ciertas zonas

No muchas, siempre salgo a tiempo para encontrar estacionamiento

Muchas veces cuando las tarifas son muy caras o no indican bien el tiempo

No.

Un par de veces

No

¿Qué mejoras te gustaría ver en los servicios de estacionamiento que utilizas actualmente?

10 responses

Que me indiquen la cantidad de estacionamientos totales

La seguridad del estacionamiento en general

Las tarifas principalmente

Indicar los costos mas que todo

Trafico por la zona

Información que no cambie

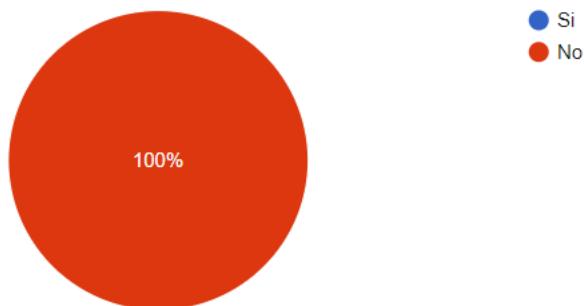
Información útil

Costo de las tarifas

Tal vez si esta cerca a algun lugar interesante

¿Consideras sencillo el proceso de gestionamiento de espacios en tu estacionamiento?

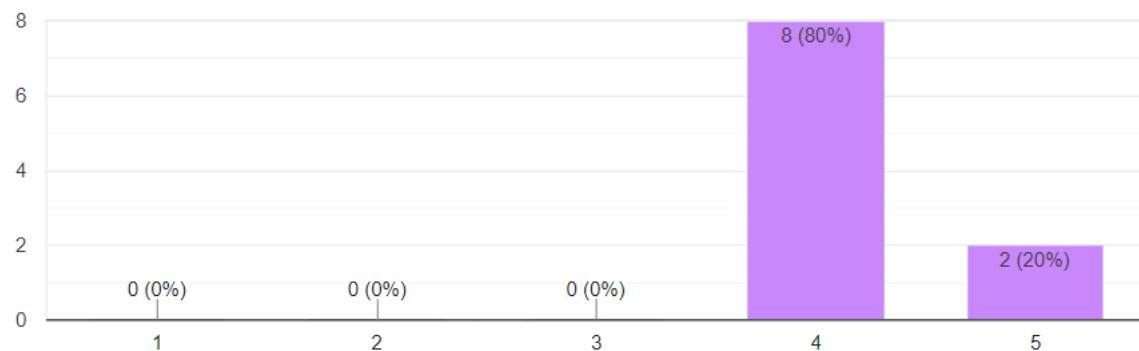
10 responses



¿Qué tan importante es para ti contar con información en tiempo real sobre los espacios disponibles?

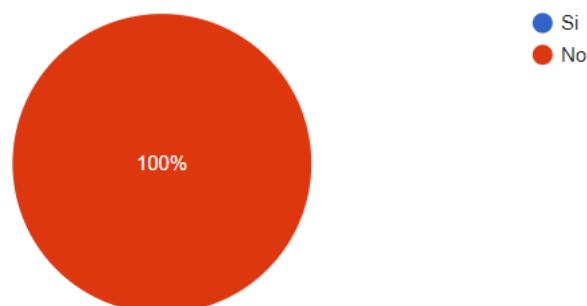
Copy

10 responses



¿Utilizas algún sistema digital para administrar los pagos y reservas de los espacios de estacionamiento?

10 responses



¿Qué beneficios ves en la posibilidad de automatizar el proceso de reserva y pago de los espacios de estacionamiento?

10 responses

Sería más fácil manejar el flujo de autos sin intervención manual

Ahoraría tiempo al no tener que cobrar en persona

Simplificaría la contabilidad al tener todos los pagos registrados digitalmente

La automatización reduciría errores humanos en las reservas

Facilitaría la operación diaria y mejoraría la experiencia del cliente

Podría monitorear las ocupaciones en tiempo real sin estar físicamente en el lugar

Ayudaría a mantener un mejor control financiero de los ingresos

Sería conveniente para los clientes y me ahorraría esfuerzo

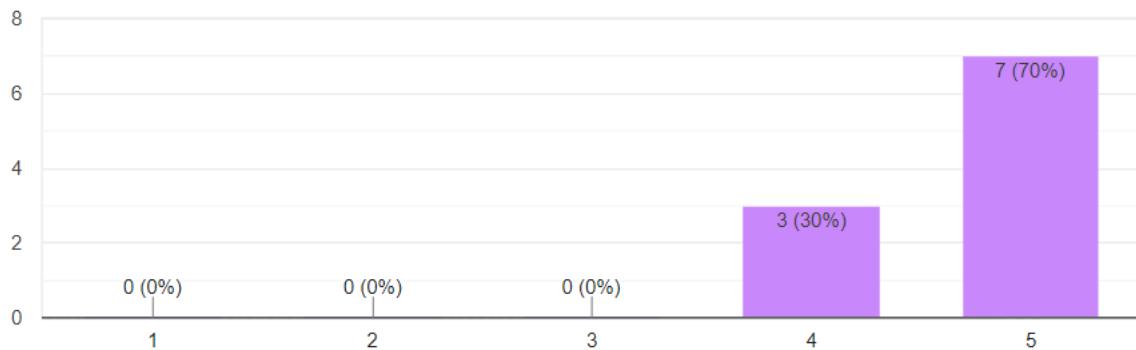
Mejoraría la seguridad al manejar menos efectivo en el sitio

Permitiría un acceso más rápido y sencillo para los clientes

 Copy

¿Qué tan relevante es para ti ofrecer opciones de pago seguras y rápidas a tus clientes?

10 responses



¿De qué manera crees que la integración de una app que gestione datos y pagos podría mejorar la eficiencia de tu negocio?

10 responses

Ayudaría a optimizar el uso de los espacios disponibles

Simplificaría la contabilidad al tener todos los pagos registrados digitalmente

Permitiría una mejor planificación diaria al conocer la demanda anticipadamente

Reduciría la necesidad de personal en la entrada y salida

Facilitaría el seguimiento de clientes frecuentes y ofrecer promociones

Mejoraría la organización y reduciría el caos en horas pico

Aceleraría los tiempos de ingreso y salida de los vehículos

Proporcionaría información valiosa para ajustar precios y ofertas

Me permitiría gestionar el estacionamiento desde cualquier lugar

¿Cuáles son los principales desafíos que enfrentas al administrar tu estacionamiento?

10 responses

Manejar la cantidad de autos en horas punta

Controlar el flujo de dinero en efectivo

Asegurar que siempre haya espacios disponibles

Evitar conflictos con los clientes por la disponibilidad

Mantener un registro exacto de las entradas y salidas

Garantizar la seguridad del área de estacionamiento

Minimizar los tiempos de espera para los clientes

Gestionar adecuadamente las reservas y cancelaciones

Optimizar el uso de los espacios en todo momento

¿Qué funcionalidades adicionales te gustaría que una app ofreciera para facilitar la gestión de tu negocio?

10 responses

Un sistema de notificaciones para avisar cuando queden pocos espacios

La posibilidad de integrar pagos con diferentes métodos

Un informe diario de ocupación y transacciones

La opción de ajustar las tarifas según la demanda en tiempo real

Una función para reservar espacios para clientes VIP

Integración con cámaras de seguridad para monitoreo en vivo

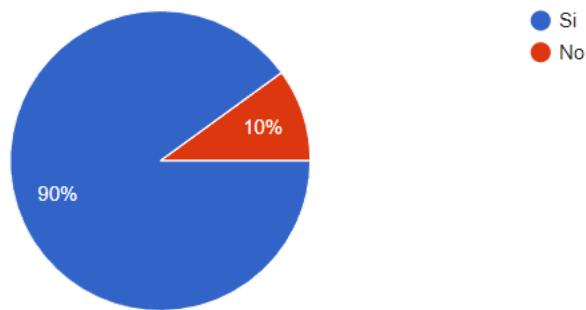
Una opción para ofrecer promociones a clientes frecuentes

Poder ajustar la capacidad máxima según eventos especiales

Acceso a estadísticas sobre los tiempos de mayor afluencia

¿Te gustaría recibir reportes o análisis sobre la ocupación y rotación de vehículos en tu estacionamiento?

10 responses



2.3. Needfinding

En esta sección se presentarán los artefactos generados a partir del proceso de recopilación llevado a cabo en los segmentos objetivos. Este proceso culmina con la creación de los User Personas, la User Task Matrix, los User Journey Maps, el Empathy Mapping y el As-Is Scenario Mapping. Los hallazgos obtenidos a través de entrevistas permiten construir representaciones precisas de los usuarios reales, facilitando un enfoque centrado en el usuario para el desarrollo de nuestra aplicación móvil.

2.3.1. User Personas

En esta sección del documento, se presentan perfiles ficticios que representan a los grupos de interés identificados y que fueron objeto de entrevistas. La información proporcionada incluye datos demográficos, rasgos de personalidad, motivaciones, preferencias, metas, desafíos y comportamientos típicos. Estos detalles están basados en las entrevistas realizadas previamente. Se han utilizado herramientas de UXPressia para facilitar la elaboración de estos perfiles y asegurar una visualización estructurada y coherente con el enfoque UX del proyecto.

User Persona para Conductores Urbanos

La "User Persona" de Anna Pacheco representa a los conductores urbanos que requieren soluciones rápidas, eficientes y seguras para estacionar cerca de sus destinos. Anna es una organizadora de eventos de 29 años, cuya rutina diaria implica múltiples desplazamientos por la ciudad. Está motivada por la comodidad y la optimización del tiempo, pero suele frustrarse por la falta de espacios disponibles y la pérdida de tiempo al buscar estacionamiento. Busca una solución móvil práctica, con información en tiempo real, reserva anticipada y funciones que mejoren su experiencia general al conducir en zonas urbanas.

User Persona: Conductora Urbana

PERSONA: Anna Pacheco

NOMBRE

Anna Pacheco

RACIONAL

Idealista



Informacion Personal

Female

29 años

Married

Organizadora de Eventos

Licenciatura de
Relaciones Públicas y
Organización de Eventos

Personalidad

Extrovertido



Emocional



Planificadora



Sensible



Objetivos

- Asegurar Estacionamientos Cercanos a las Ubicaciones de los Eventos
- Minimizar el Tiempo Perdido en la Búsqueda de Espacios Libres
- Controlar y Prever Gastos Operativos Asociados al Estacionamiento.

Quote

"La felicidad no es un destino, sino una forma de viajar."

Biografia

Anna visita nuevas ciudades con frecuencia y prefiere alquilar un coche para explorar. Necesita estacionamiento cercano a atracciones turísticas, restaurantes y lugares de interés. Busca comodidad y rapidez en sus desplazamientos.

Motivaciones

- Su familia.
- Nuevos retos.
- Pasión por la tecnología.
- Ampliar su huella digital

Frustraciones

- Limitaciones de Disponibilidad en Horas Pico
- No encontrar lugar para estacionarse cerca del lugar requerido
- Transparencia y Competitividad de Tarifas

Brands and influencers



User Person para Dueños de estacionamientos

La "User Persona" de Luis Cordova representa a propietarios de estacionamientos interesados en mejorar la gestión y aumentar ingresos. Luis, de 26 años y dueño de un estacionamiento, utiliza tecnología para optimizar la ocupación y reducir tiempos de espera. Sus objetivos incluyen la fidelización de clientes y la expansión del negocio, pero enfrenta frustraciones con problemas tecnológicos y altos costos operativos.

User Persona: Dueño de Estacionamiento

PERSONA: Luis Cordova

NOMBRE

Luis Cordova

RACIONAL

Racional



Información Personal

Male

26 años

Single

Dueño de
estacionamiento

Licenciatura en Ingeniería
Mecatrónica

Objetivos

Fidelización de Clientes
Mejora de la Experiencia del Cliente
Expansión del Negocio
Aumentar los ingresos mientras se controlan los costos operativos.

Quote

"Siempre lucha, gana a veces, nunca te rindas."

Biografía

Luis es dueño de un estacionamiento cercano a una universidad y a diversas empresas. Utiliza aplicaciones para optimizar la gestión del estacionamiento y evitar problemas de tráfico en las horas pico. Monitorea constantemente la disponibilidad de espacios para maximizar la ocupación y reducir el tiempo de espera para los conductores.

Personalidad

Extrovertido



Emocional



Planificadora



Sensible



Motivaciones

- Ser su propio jefe.
- Su familia.
- Nuevos retos.
- Pasión por la tecnología.

Frustraciones

- Problemas Tecnológicos
- Altos Costos Operativos y de Mantenimiento
- Competencia en el Mercado
- Limitaciones de Espacio

Brands and influencers



2.3.2. User Task Matrix

En el User Task Matrix hemos identificado las actividades que realizan nuestros dos segmentos y hemos evaluado la importancia de cada una de estas tareas para cada segmento.

Indicadores de Importancia:

- ALTA
- MEDIA
- BAJA

Indicadores de Frecuencia:

- ALTA
- MEDIA
- BAJA

A través del User Task Matrix, podemos identificar las frecuencias e importancias entre los diferentes segmentos que presentamos y usar esta información como guía.

Usuarios:

Para los usuarios, las tareas que presentan mayor frecuencia son:

Segmento Objetivo	Tarea	Frecuencia	Importancia
Conductores Urbanos Frecuentes	Buscar estacionamientos disponibles	Media	Alta
Conductores Urbanos Frecuentes	Comparar precios de estacionamientos	Alta	Media
Conductores Urbanos Frecuentes	Reservar estacionamientos con anticipación	Media	Alta
Conductores Urbanos Frecuentes	Pagar por estacionamiento a través de la aplicación	Alta	Alta
Conductores Urbanos Frecuentes	Recibir direcciones al estacionamiento reservado	Alta	Alta
Conductores Urbanos Frecuentes	Filtrar estacionamientos por criterios (precio, seguridad, etc.)	Alta	Alta
Conductores Urbanos Frecuentes	Recibir notificaciones sobre descuentos y ofertas	Media	Baja
Conductores Urbanos Frecuentes	Verificar disponibilidad de estacionamiento en tiempo real	Alta	Alta
Conductores Urbanos Frecuentes	Consultar historial de estacionamientos utilizados	Media	Baja
Dueños de Estacionamientos	Administrar y actualizar la disponibilidad de espacios	Alta	Alta
Dueños de Estacionamientos	Ajustar precios de los espacios de estacionamiento	Media	Alta
Dueños de Estacionamientos	Gestionar reservas de estacionamientos	Alta	Alta

Segmento Objetivo	Tarea	Frecuencia	Importancia
Dueños de Estacionamientos	Procesar pagos de estacionamiento a través de la aplicación	Alta	Media
Dueños de Estacionamientos	Revisar y ajustar las características del estacionamiento (precio, seguridad, etc.)	Media	Alta
Dueños de Estacionamientos	Revisar reportes de uso y disponibilidad	Media	Media
Dueños de Estacionamientos	Recibir y gestionar comentarios y valoraciones de los usuarios	Baja	Media
Dueños de Estacionamientos	Promocionar descuentos y ofertas especiales	Media	Media
Dueños de Estacionamientos	Consultar el historial de transacciones y reservas	Baja	Media

A través del User Task Matrix, podemos identificar las frecuencias e importancias entre los diferentes segmentos que presentamos y usar esta información como guía.

Conductores Urbanos:

Para los usuarios, las tareas que presentan mayor frecuencia son:

- Pagar por estacionamiento a través de la aplicación.
- Recibir direcciones al estacionamiento reservado.
- Filtrar estacionamientos por criterios (precio, seguridad, etc.).
- Verificar disponibilidad de estacionamiento en tiempo real.

Estas tareas son cruciales para mejorar la experiencia del usuario en la búsqueda y uso de estacionamientos.

Para los usuarios, las tareas que presentan mayor importancia son:

- Pagar por estacionamiento a través de la aplicación.
- Recibir direcciones al estacionamiento reservado.
- Filtrar estacionamientos por criterios (precio, seguridad, etc.).
- Verificar disponibilidad de estacionamiento en tiempo real.

Estas tareas son importantes porque aseguran una experiencia de estacionamiento eficiente y satisfactoria, desde la búsqueda hasta el pago y la dirección.

Dueños de Estacionamientos:

Para los dueños de estacionamientos, las tareas que presentan mayor frecuencia son:

- Administrar y actualizar la disponibilidad de espacios.
- Gestionar reservas de estacionamientos.
- Procesar pagos de estacionamiento a través de la aplicación.
- Revisar y ajustar las características del estacionamiento (precio, seguridad, etc.).

Estas tareas son esenciales para la gestión efectiva de sus espacios de estacionamiento y para asegurar que los usuarios tengan una experiencia positiva.

Para los dueños de estacionamientos, las tareas que presentan mayor importancia son:

- Administrar y actualizar la disponibilidad de espacios.
- Ajustar precios de los espacios de estacionamiento.
- Revisar reportes de uso y disponibilidad.
- Revisar y ajustar las características del estacionamiento (precio, seguridad, etc.).

Estas tareas son importantes porque permiten a los dueños optimizar la gestión de sus espacios y responder a las demandas del mercado de manera efectiva.

Diferencias:

Al analizar ambos segmentos objetivo, se observa que los usuarios están enfocados en la eficiencia y la comodidad en el proceso de estacionamiento, mientras que los dueños de estacionamientos se concentran en la gestión y optimización de sus espacios. Aunque cada segmento tiene objetivos distintos, ambos se benefician de un sistema que facilite la disponibilidad, la gestión y la experiencia de estacionamiento.

2.3.3. User Journey Mapping

En esta sección, hemos creado los mapas de recorrido del usuario para cada tipo de usuario (cliente y propietario de estacionamiento). Para el cliente, el proceso comienza con la búsqueda de un espacio de estacionamiento, continúa con la selección y reserva del espacio, el pago a través de la aplicación y la obtención de direcciones hacia el lugar reservado. Finalmente, el cliente llega al espacio y lo utiliza. Este recorrido permite identificar metas, desafíos, oportunidades y pasos clave dentro de su experiencia de uso, lo cual es fundamental para optimizar la funcionalidad de la aplicación y la satisfacción del usuario.

Desde la perspectiva del propietario de estacionamiento, el recorrido implica gestionar la disponibilidad de espacios, ajustar precios de manera estratégica, procesar reservas en tiempo real y asegurar que el proceso de pago y uso se realice de forma eficiente. El propósito es brindar una experiencia fluida y confiable que motive a los clientes a recomendar el servicio y considerar al propietario como una opción constante para futuros estacionamientos.

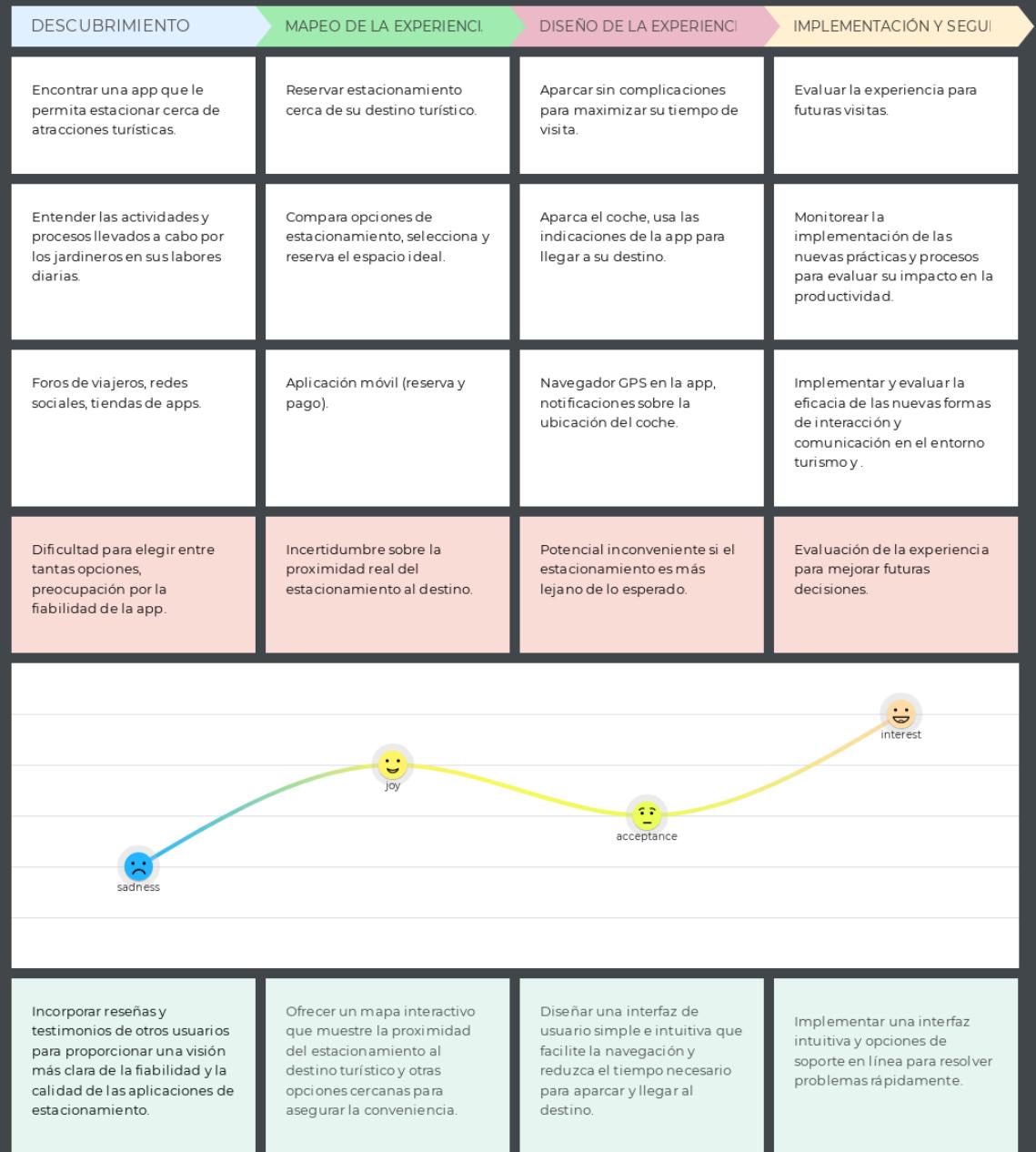
User Journey Mapping para Conductores Urbanos

Este User Journey Mapping describe las etapas que un conductor urbano atravesía al utilizar una aplicación de estacionamiento. El recorrido inicia con el descubrimiento de la app, ya sea por recomendaciones, búsqueda directa o publicidad. Luego, el usuario explora las opciones disponibles y aplica filtros según criterios como precio, ubicación y seguridad. Una vez identificado el espacio adecuado, realiza la reserva a través de la app y efectúa el pago mediante medios digitales. Posteriormente, recibe indicaciones vía GPS hacia el espacio reservado, llega al lugar y lo ocupa. Finalmente, evalúa su experiencia y puede dejar una valoración del servicio recibido.

Durante todo el recorrido, se identifican puntos de contacto clave como la interfaz de usuario, el sistema de navegación, las notificaciones automáticas y el soporte integrado. También se reconocen problemas comunes como una navegación poco intuitiva, confirmaciones confusas o lentitud en la carga de información. Como oportunidades de mejora se proponen: simplificar la experiencia de reserva, incluir mensajes de confirmación visuales y sonoros, activar alertas de disponibilidad en tiempo real, y permitir guardar espacios frecuentes o favoritos. Estos aspectos son fundamentales para reforzar la usabilidad, confianza y fidelidad del usuario hacia la aplicación.

User Journey: Conductor Urbano

MAP: Customer journey map 1



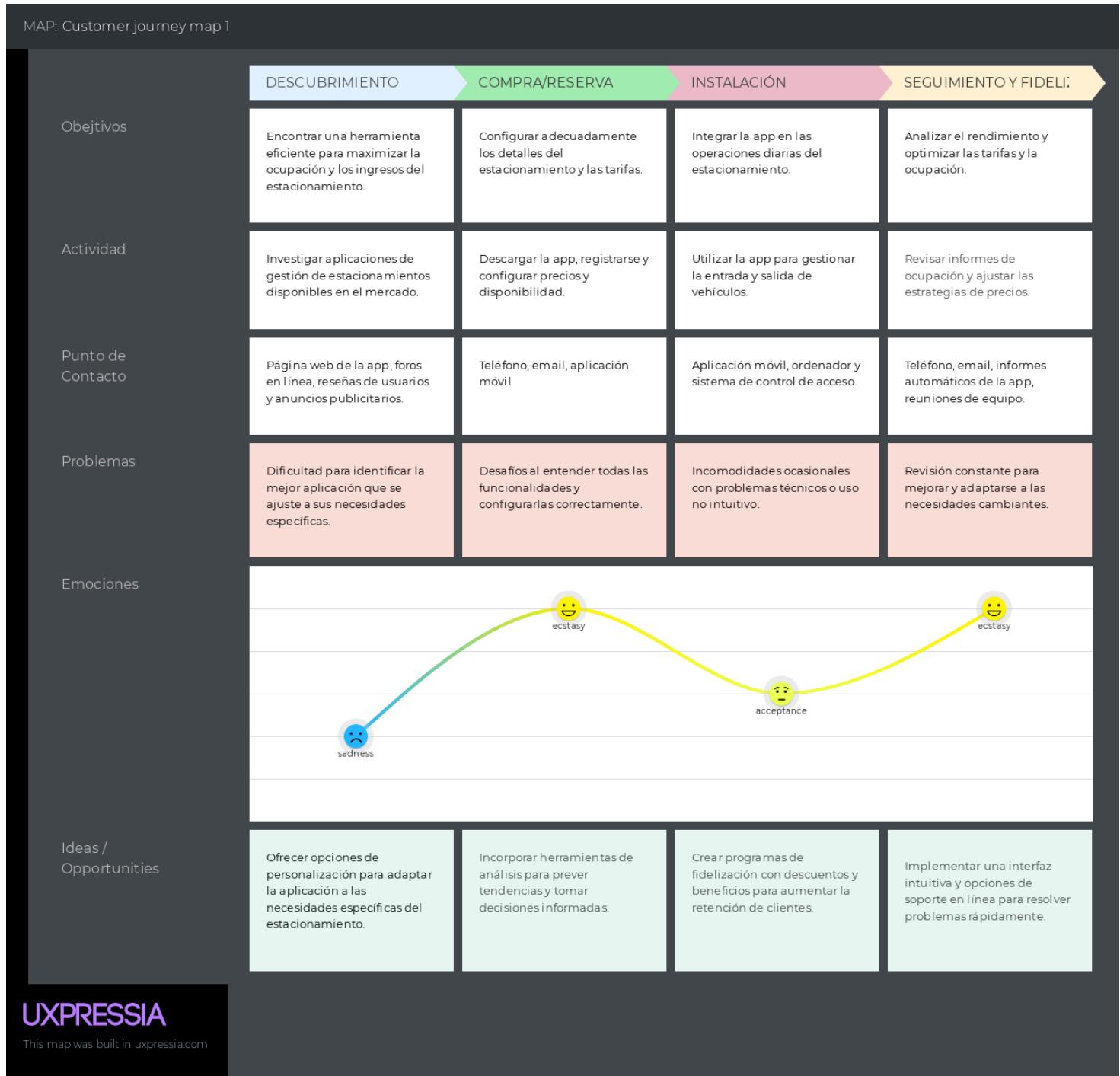
UXPRESSIA

This map was built in uxpressia.com

User Journey Mapping para Dueños de Estacionamientos

Este "User Journey Mapping" para dueños de estacionamientos describe las etapas de uso de una app de gestión, desde encontrar la herramienta adecuada y configurarla, hasta su instalación y seguimiento para optimización. Destaca desafíos en la selección de la app y la comprensión de sus funciones, con oportunidades para mejorar la personalización, la facilidad de uso y el soporte técnico.

User Journey Mapping: Conductor Urbano



2.3.4. Empathy Mapping

A continuación se pueden apreciar los Empathy Mapping de los segmentos objetivos.

Segmento objetivo: Conductores Urbanos Frecuentes

Este "Empathy Map" representa a conductores urbanos frecuentes que enfrentan problemas para encontrar estacionamiento. Expresa sus frustraciones por la falta de opciones disponibles y su deseo de contar con herramientas que les brinden información en tiempo real sobre espacios seguros y accesibles.

Empathy Map: Conductor Urbano Frecuente

Segmento Objetivo <p>Conductores que han sido afectados por la problemática de no encontrar estacionamiento cerca de sus destinos, reuniones, trabajo, etc.</p>	Que Siente? <p>“ Que es un problema el hecho de que los estacionamientos sean pequeños y que es difícil encontrar estacionamiento en ciertos distritos ”</p>	Que piensa? <p>Piensa que es importante conocer la disponibilidad de estacionamiento cuando hay actividades importantes</p>
Que oye? <p>Algunos lugares cobran caro ya que es por hora y por fraccion Algunos lugares carecen de personal que cuide los vehículos es buena idea el hecho de tener disponibilidad de estacionamiento en tiempo real</p>		Que dice? <p>En algunas zonas le es mas difícil encontrar estacionamientos necesita bastante usar estacionamientos casi diariamente Estaría dispuesto a pagar mas por una reserva de estacionamiento siempre y cuando ese costo extra no sea excesivo</p>

Que hace?

Ah tenido que dejar el auto en lugares que no eran ideales para estacionar pero no había otra opción
se demora en buscar estacionamiento, dependiendo de la zona demora entre 5 y 15 min ademas que este puede ser tomado por otro conductor

PAINS

- * No hay estacionamientos amplios
- * Los estacionamientos pueden ser tomados por otros conductores
- * Actualmente no hay ninguna app de estacionamiento que facilite su vida

GAINS

- * Necesitan poder ver la disponibilidad de estacionamientos en tiempo real.
- * Necesitan tener estacionamientos seguros cuando se dirigen a reuniones importantes

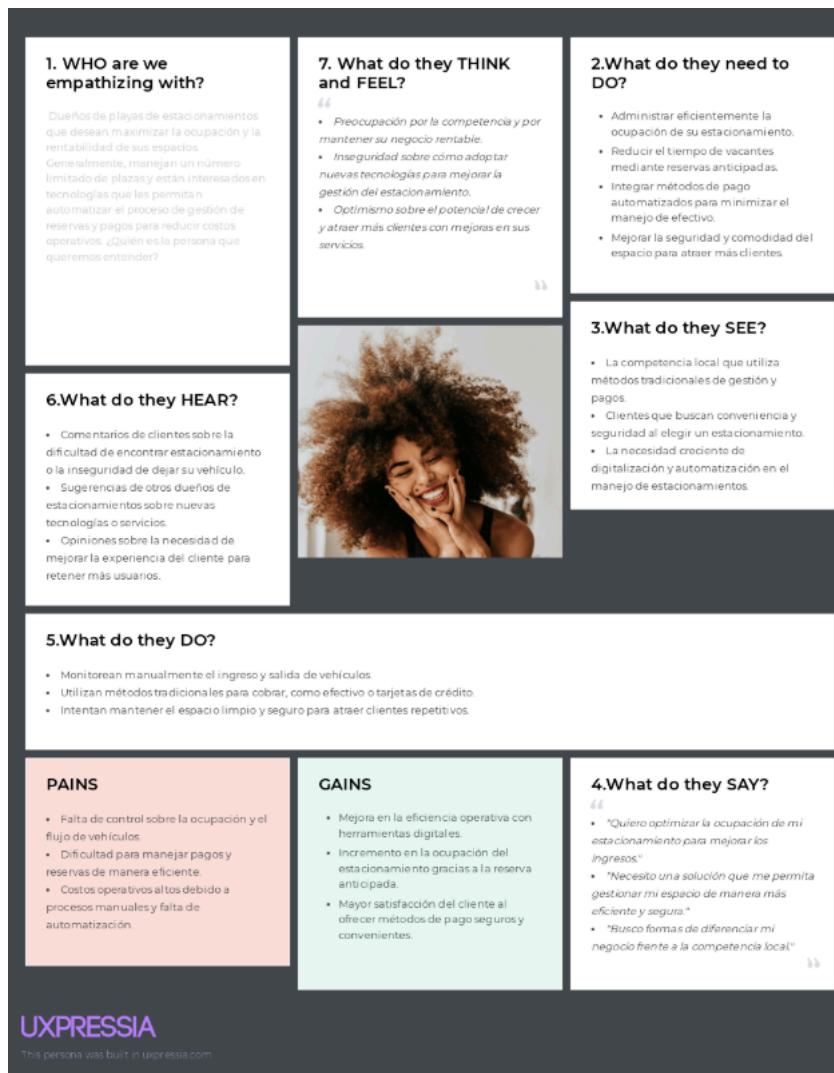
UXPRESSIA

This persona was built in upressoia.com

Segmento objetivo: Dueños de estacionamientos

El mapa de empatía para **Dueños de Estacionamientos** identifica sus necesidades y preocupaciones al gestionar espacios de estacionamiento, destacando oportunidades para optimizar operaciones y mejorar la experiencia del cliente con **PARKINGNOW**.

Empathy Map: Dueños de Estacionamientos



2.3.5. As-is Scenario Mapping

El As-Is nos proporciona un análisis visual de los escenarios actuales para identificar oportunidades de mejora en la experiencia del usuario.

Segmento Conductores Urbanos Frecuentes

El "As-is Scenario Mapping" para **Conductores Urbanos Frecuentes** resalta los desafíos y mejoras necesarias en la experiencia de estacionamiento, permitiendo optimizar las soluciones de **PARKINGNOW**.

As-is Scenario Mapping: Conductor Urbano Frecuente

Steps	Busca playas de estacionamiento cerca de su destino	Busca espacios disponibles	Encuentra un estacionamiento y se aproxima a él	Se estaciona en el lugar seleccionado previamente	Asegura sus pertenencias antes de retirarse de su vehículo	Paga el tiempo usado por el vehículo
Doing	Recuerda estacionamientos usados previamente	Recorre el lugar buscando estacionamiento disponible	Se acerca en el lugar visualizado previamente con intenciones de aparcar	Procede a estacionarse esperando no chocar y no cerrar el paso al siguiente vehículo	Lleva sus cosas de valor o las esconde en lugares no visibles del vehículo	Se acerca a caja a pagar por el tiempo que estacionó su vehículo
Thinking	"Quizá si llego rápido encontraré estacionamiento"	"No hay ni un solo estacionamiento libre!"	"Ojalá otro conductor no me gane ese estacionamiento antes que yo"	"Espero sea suficiente espacio para que el otro conductor pueda salir y no me choque"	"Quizá si lo escondo bien no me rompan la ventana"	"Espero no consideren esos 5 minutos como fracción de hora"
Feeling	Siente Frustración al no encontrar estacionamiento.	Siente ansiedad por el tiempo que comienza a estar dando vueltas en el lugar sin encontrar estacionamiento.	Siente desesperación al ver un solo lugar disponible y varios vehículos.	Siente preocupación pues al retirarse el vehículo adyacente podría colisionar con él.	Siente inseguridad pues tampoco tiene reservas del lugar de estacionamiento.	Siente estrés pues la fracción y hora cobra igual sin tiempos de gracia.

Segmento Dueños de estacionamientos

El "As-is Scenario Mapping" para **Dueños de Estacionamientos** identifica sus desafíos y necesidades, desde la preparación del espacio hasta la gestión post-servicio, ayudando a mejorar la eficiencia y satisfacción en la gestión de sus estacionamientos con PARKINGNOW.

As-is Scenario Mapping: Dueños de Estacionamientos

Steps	Abrir las puertas de su playa de estacionamiento	publicar espacios disponibles	Gestionar lugares de estacionamiento	registrar vehículos	Revisar tiempo y realizar cobro	Crear promociones y ajustar precios por temporada
Doing	Inicia el dia y abre las puertas de su playa	Escribe carteles con los precios y los saca a afuera de su playa	Verifica que sus lugares de estacionamiento esten aptos	Registra en un cuaderno la placa y la hora de entrada	Busca en su cuaderno la hora de entrada y calcula el pago	Agrega ofertas como una hora gratis por cada 24 acumuladas
Thinking	"Espero tener buena cantidad de clientes y no problemáticos"	"Ojalá los precios no se les haga caro a los clientes"	"Debería hacer estacionamientos más pequeños para albergar mas vehículos"	"Es muy tedioso tener que hacer estos registros a mano"	"Espero no equivocarme con la tarifa por horas"	"Espero que esta oferta atraiga mas clientes"
Feeling	Siente que podría tener clientes problemáticos.	Siente que los precios podrían ser altos	Siente que debería poder albergar mas vehículos en su espacio de estacionamiento	Siente que esa técnica es muy ortodoxa	Siente que podría equivocarse y sacar mal la cuenta	Siente que agregar ofertas será beneficioso para el negocio

2.4. Ubiquitous Language

Espacio de Estacionamiento: Un lugar específico dentro de un estacionamiento donde un vehículo puede ser aparcado.

Reserva: La acción de reservar un espacio de estacionamiento específico por un periodo de tiempo determinado, garantizando su disponibilidad.

Disponibilidad en Tiempo Real: Información actualizada instantáneamente sobre los espacios de estacionamiento libres y ocupados.

Geolocalización: El uso de tecnologías de ubicación para determinar la posición geográfica del usuario y mostrar opciones de estacionamiento cercanas.

Mapa Interactivo: Un mapa digital dentro de la aplicación que muestra la ubicación de estacionamientos disponibles en tiempo real.

Tarifas por Hora: El costo asociado con la reserva de un espacio de estacionamiento, calculado por horas de uso.

Pago en Línea: El proceso de completar el pago para una reserva a través de métodos de pago digitales, como tarjetas de crédito, débito o billeteras electrónicas.

Historial de Reservas: Un registro de todas las reservas pasadas realizadas por el usuario, que incluye detalles como fecha, hora, ubicación y costo.

Notificaciones en Tiempo Real: Alertas enviadas al usuario para informarlos sobre cambios en la disponibilidad, confirmaciones de reserva, o vencimientos de tiempo de estacionamiento.

Usuario: Persona que utiliza la aplicación para buscar, reservar y pagar por espacios de estacionamiento.

Propietario de Estacionamiento: Persona o entidad que posee y administra un estacionamiento y que ofrece espacios para la reserva a través de la aplicación.

Administrador del Sistema: Persona responsable de la gestión y mantenimiento de la aplicación, incluyendo la supervisión de la disponibilidad de los espacios y la gestión de usuarios y datos.

Check-in: El proceso de confirmar la llegada al espacio de estacionamiento reservado, activando el tiempo de uso.

Check-out: El proceso de marcar la salida del espacio de estacionamiento, finalizando el tiempo de uso y actualizando la disponibilidad.

Zona de Estacionamiento: Una área geográfica específica que contiene múltiples espacios de estacionamiento, gestionada dentro de la aplicación.

Tiempo de Gracia: Un periodo corto de tiempo permitido antes o después de la reserva, en el cual el usuario puede llegar sin incurrir en cargos adicionales.

Multa por Exceso de Tiempo: Un cargo adicional aplicado si un usuario excede el tiempo de estacionamiento reservado.

Sistema de Validación: Mecanismo para verificar que un vehículo tenga una reserva válida para un espacio de estacionamiento específico.

Interfaz de Usuario (UI): La parte visual de la aplicación con la que los usuarios interactúan para realizar reservas, hacer pagos y recibir notificaciones.

Backend: Infraestructura que aloja y gestiona los datos de la aplicación, incluyendo la disponibilidad de espacios y la información de los usuarios.

Capítulo III: Requirements Specification

3.1. To-Be Scenario Mapping

Segmento Conductores Frecuentes

El "To-Be Scenario Mapping" para **Conductores Urbanos Frecuentes** Muestra los resultados esperados para los conductores urbanos frecuentes que necesitan tener rápido acceso a estacionamientos

Escenario ideal para usuarios frecuentes que buscan estacionar sin complicaciones.

	Iniciar sesión en PARKINGNOW	Buscar espacios disponibles	Seleccionar y reservar espacio	Llegar al estacionamiento	Pagar desde la aplicación	Dejar una reseña
Stress	Abre la app y accede a la búsqueda de estacionamientos.	Ingresá ubicación y busca espacios disponibles.	Selecciona y reserva el espacio deseado.	Sigue las indicaciones para llegar al espacio reservado.	Completa el pago de manera digital.	Deja una reseña sobre la experiencia.
Thinking	"Necesito un espacio cerca de mi destino."	"Espero que haya opciones cercanas y económicas."	"Fácil y rápido, justo lo que necesito."	"Me guía directo, sin complicaciones."	"Me gusta no tener que usar efectivo o tarjetas."	"Quiero ayudar a otros conductores con mi opinión."
Feelings	Confiado de encontrar un lugar fácilmente.	Optimista al ver opciones disponibles.	Satisfecho con la rapidez de la reserva.	Aliviado por no tener que buscar al llegar.	Satisfecho con la facilidad del pago.	Valorado al contribuir con su retroalimentación.

Segmento Dueños de estacionamientos

El "To-Be Scenario Mapping" para **Dueños de Estacionamientos** muestra los resultados esperados para la gestión de estacionamientos brindadas por **PARKINGNOW**.

Escenario ideal para usuarios frecuentes que buscan estacionar sin complicaciones.

	Registrarse en PARKINGNOW	Publicar espacios disponibles	Gestionar reservas	Confirmar o rechazar reservas	Revisar métricas	Optimizar precios y promociones
Stress	Configura la cuenta para gestionar sus espacios.	Añade detalles y tarifas para cada espacio.	Revisa y ajusta la disponibilidad en la plataforma.	Toma decisiones sobre las reservas según disponibilidad.	Revisa las métricas de ocupación y las reseñas.	Ajusta precios y crea promociones según la demanda.
Thinking	"Quiero atraer más clientes y mejorar la ocupación."	"Detalles claros atraerán más clientes."	"Necesito maximizar mis ingresos eficientemente."	"Cada reserva debe ser rentable y conveniente."	"Me gustaría mejorar basándome en los comentarios."	"Debo ser estratégico para mantener la ocupación alta."
Feelings	Entusiasmado por nuevas oportunidades.	Confiado en la presentación de su oferta.	Satisfecho con el control de las reservas.	Seguro en sus decisiones de gestión.	Motivado por mejorar la ocupación y calidad.	Proactivo y en control del negocio.

Segmento: Conductores Urbanos Frecuentes

EPICS

Las **epics** definidas para el proyecto PARKINGNOW están diseñadas para abordar las necesidades clave de los usuarios y el equipo de desarrollo. Estas epics cubren funcionalidades críticas para la plataforma de gestión de estacionamiento en tiempo real, asegurando que tanto los visitantes como los desarrolladores puedan interactuar eficazmente con el sistema. Desde la experiencia del visitante de la aplicación móvil en la landing page hasta la gestión técnica del backend, frontend y mobile services, las epics proporcionan una guía estructurada para el desarrollo de características esenciales, garantizando una implementación exitosa y alineada con los objetivos del proyecto.

Epic / Story ID	Título	Descripción
EP01	Navegación en Landing Page	Como visitante de PARKINGNOW, deseo poder navegar de forma intuitiva por la landing page para acceder rápidamente a información importante y funciones clave.
EP02	Autenticación y Registro de Usuarios	Como visitante, deseo registrarme e iniciar sesión utilizando correo electrónico o redes sociales para acceder de manera conveniente y segura.
EP03	Recuperación de Contraseña	Como visitante, deseo recuperar mi contraseña fácilmente para poder acceder a mi cuenta en caso de olvido.
EP04	Visualización de Servicios y Beneficios	Como visitante, deseo conocer los servicios y beneficios de PARKINGNOW desde la landing page para entender cómo la plataforma puede ayudarme.
EP05	Soporte y FAQ	Como visitante, deseo acceder a una sección de soporte y preguntas frecuentes para resolver mis dudas de manera rápida y autónoma.
EP06	Conexión con Redes Sociales	Como visitante, deseo conectar mi cuenta de PARKINGNOW con mis redes sociales para compartir mi experiencia y obtener beneficios adicionales.
EP07	Búsqueda de Estacionamientos	Como visitante, deseo buscar estacionamientos y ver la disponibilidad en tiempo real para elegir el mejor lugar antes de dirigirme.
EP08	Disponibilidad en Tiempo Real	Como visitante, deseo visualizar la disponibilidad de los estacionamientos actualizada automáticamente para evitar reservas en espacios ocupados.
EP09	Gestión de Reservas	Como visitante, deseo poder realizar, modificar y cancelar mis reservas de estacionamiento para ajustar mis planes con flexibilidad.
EP10	Gestión de Perfil de Usuario	Como visitante, deseo actualizar y gestionar mi perfil de usuario para mantener mi información y preferencias personalizadas.
EP11	Notificaciones en Tiempo Real	Como visitante, deseo recibir notificaciones sobre eventos importantes y mis reservas para estar informado de manera oportuna.
EP12	Integración con Mapas	Como visitante, deseo ver rutas y ubicaciones de estacionamientos mediante integración con mapas para optimizar mi tiempo y desplazamiento.
EP13	Funcionalidad de Favoritos	Como visitante, deseo marcar estacionamientos como favoritos para acceder rápidamente a los lugares que uso frecuentemente.
EP14	Opiniones y Calificaciones	Como visitante, deseo dejar y leer opiniones de otros usuarios sobre los estacionamientos para tomar decisiones informadas.
EP15	Seguridad y Privacidad de Datos	Como visitante, deseo que mis datos personales y de pago estén protegidos con altos estándares de seguridad para sentirme seguro usando la app.
EP16	Optimización de Backend	Como desarrollador, deseo mejorar la eficiencia del backend para consultas y búsquedas para reducir los tiempos de espera y mejorar el rendimiento.
EP17	Gestión de Promociones y Beneficios	Como visitante, deseo acceder a promociones y beneficios personalizados para maximizar mis ahorros y recompensas.

Epic / Story ID	Título	Descripción
EP18	Administración de Reservas en Backend	Como developer, deseo gestionar las reservas en el backend para asegurar que los datos de reserva sean consistentes y actualizados en la plataforma.
EP19	Gestión de Pagos y Transacciones	Como visitante, deseo realizar pagos seguros y gestionar mis transacciones en la app para completar el proceso de reserva de manera confiable.

User Stories

Las **historias de usuario** para el proyecto PARKINGNOW están orientadas a mejorar la experiencia del usuario final y facilitar las tareas del equipo de desarrollo. Estas historias cubren múltiples aspectos de la plataforma, incluyendo la **landing page**, la integración y mejora del **backend**, el desarrollo de interfaces de usuario eficientes en el **frontend**, y la implementación de servicios robustos a través de **mobile application**. Cada historia está diseñada para ser fácilmente comprensible y directamente aplicable, con criterios de aceptación específicos que guían el desarrollo y aseguran que las funcionalidades implementadas satisfagan los requisitos del negocio y las expectativas del usuario final.

Epic / Story ID	Título	Descripción	Criterios de Aceptación	Relacionado con (Epic ID)
US01	Navegación Intuitiva en la Landing Page	Como visitante de PARKINGNOW, deseo que la landing page tenga una barra de navegación clara y accesible para encontrar fácilmente las secciones importantes.	Escenario 01: Given que estoy en la landing page, When hago clic en el menú de navegación, Then debería ver opciones claras como "Inicio", "Características", "Reservar Espacio" y "Contáctenos". Escenario 02: Given que navego por la página, When paso el mouse por los elementos del menú, Then deberían resaltarse para indicar que son interactivos.	EP01
US02	Visualización de Ciudades con Servicio	Como visitante, deseo ver en la landing page las ciudades donde PARKINGNOW está disponible para saber si puedo utilizar el servicio en mi ubicación.	Escenario 01: Given que estoy en la landing page, When desplazo hacia la sección de ciudades, Then debería ver información sobre las ciudades como Lima, Arequipa y Cusco. Escenario 02: Given que selecciono una ciudad, When hago clic en la información de la ciudad, Then debería ver detalles adicionales de esa ubicación.	EP04
US03	Promociones Destacadas	Como visitante, deseo ver promociones destacadas en la landing page para aprovechar descuentos y beneficios en las ciudades disponibles.	Escenario 01: Given que estoy en la landing page, When desplazo hacia la sección de promociones, Then debería ver promociones específicas como "Promoción en Lima" y "Promoción en Cusco". Escenario 02: Given que hago clic en una promoción, When selecciono una promoción específica, Then debería llevarme a una página de detalles de la promoción.	EP05
US04	Beneficios para Conductores y Dueños de estacionamiento	Como visitante, deseo ver una sección en la landing page que me explique los beneficios de usar PARKINGNOW para entender las ventajas que ofrece.	Escenario 01: Given que estoy en la landing page, When desplazo hacia la sección de beneficios, Then debería ver información sobre beneficios para conductores y dueños de estacionamiento. Escenario 02: Given que quiero más detalles, When hago clic en un beneficio específico, Then debería expandirse con más información.	EP04
US05	Visualización de Misión de la Empresa	Como visitante, deseo ver la misión de PARKINGNOW en la landing page para entender los objetivos de la empresa.	Escenario 01: Given que estoy en la landing page, When desplazo hacia la sección de "Misión", Then debería ver una breve descripción de la misión de la empresa. Escenario 02: Given que estoy en la sección de misión, When hago clic en "Leer más", Then debería abrirse una ventana con información detallada sobre la misión.	EP06

Epic / Story ID	Título	Descripción	Criterios de Aceptación	Relacionado con (Epic ID)
US06	Visualización de Visión de la Empresa	Como visitante, deseo ver la visión de PARKINGNOW en la landing page para conocer sus metas futuras y crecimiento.	Escenario 01: Given que estoy en la landing page, When desplazo hacia la sección de "Visión", Then debería ver una breve descripción de la visión de la empresa. Escenario 02: Given que quiero conocer más detalles, When hago clic en "Leer más" en la sección de visión, Then debería ver una descripción completa de los objetivos futuros.	EP06
US07	Visualización del Equipo	Como visitante, deseo ver una sección que muestre a los miembros del equipo de PARKINGNOW para conocer quiénes están detrás del proyecto.	Escenario 01: Given que estoy en la landing page, When desplazo hacia la sección "Our Team", Then debería ver fotos y descripciones de cada miembro del equipo. Escenario 02: Given que estoy en la sección del equipo, When hago clic en la imagen de un miembro, Then debería ver información detallada sobre sus roles y experiencia.	EP04
US08	Información de Contacto	Como visitante, deseo ver una sección de contacto en la landing page para saber cómo comunicarme con PARKINGNOW si tengo preguntas o comentarios.	Escenario 01: Given que estoy en la landing page, When desplazo hacia la sección "Contact Us", Then debería ver un formulario para ingresar nombre, correo electrónico y mensaje. Escenario 02: Given que estoy en la sección de contacto, When hago clic en "Enviar Mensaje" sin llenar todos los campos, Then debería ver un mensaje de error indicando que todos los campos son obligatorios.	EP07
US09	Envío de Mensaje de Contacto	Como visitante, deseo poder enviar un mensaje a PARKINGNOW desde la landing page para hacer preguntas o dar feedback.	Escenario 01: Given que estoy en la sección de contacto, When ingreso mi nombre, correo y mensaje y hago clic en "Enviar Mensaje", Then debería ver una confirmación de que el mensaje fue enviado exitosamente. Escenario 02: Given que he enviado un mensaje, When el envío es exitoso, Then debería recibir un correo de confirmación indicando que PARKINGNOW ha recibido mi mensaje.	EP07
US10	Visualización de Información de Características	Como visitante, deseo ver las características de PARKINGNOW en la landing page para entender qué ofrece el servicio.	Escenario 01: Given que estoy en la landing page, When desplazo hacia la sección de características, Then debería ver una descripción de las características principales de PARKINGNOW. Escenario 02: Given que quiero conocer más detalles, When hago clic en una característica específica, Then debería abrirse una ventana emergente con información adicional.	EP04
US11	Botón de Registro e Inicio de Sesión	Como visitante, deseo ver botones destacados de "Registrarse" e "Iniciar Sesión" en la landing page para acceder a la plataforma o crear una cuenta fácilmente.	Escenario 01: Given que estoy en la landing page, When veo la sección superior de la página, Then deberían estar visibles los botones de "Iniciar Sesión" y "Registrarse". Escenario 02: Given que hago clic en "Iniciar Sesión", When no tengo una cuenta creada, Then debería ver una opción para registrarme rápidamente.	EP02
US12	Enlace a Redes Sociales	Como visitante, deseo ver enlaces a las redes sociales de PARKINGNOW en la landing page para seguir sus novedades y actualizaciones.	Escenario 01: Given que estoy en la landing page, When desplazo hacia el pie de página, Then debería ver íconos de redes sociales que me lleven a las cuentas oficiales de PARKINGNOW. Escenario 02: Given que hago clic en un ícono de redes sociales, When selecciono Facebook, Then debería redirigirme a la página oficial de PARKINGNOW en Facebook.	EP06
US13	Visualización de Slogan	Como visitante, deseo ver un slogan atractivo en la landing page de PARKINGNOW para	Escenario 01: Given que estoy en la landing page, When veo la sección principal, Then debería encontrar un slogan o mensaje atractivo que me introduzca al servicio. 	EP04

Epic / Story ID	Título	Descripción	Criterios de Aceptación	Relacionado con (Epic ID)
		entender de inmediato la propuesta del servicio.	Escenario 02: Given que el slogan aparece en la sección principal, When paso a otra sección y vuelvo a la principal, Then el slogan debería mantenerse visible.	
US14	Acceso a Información Legal	Como visitante, deseo ver enlaces a la información legal y políticas de privacidad en la landing page para revisar los términos de uso y la protección de mis datos.	Escenario 01: Given que estoy en la landing page, When desplazo hacia el pie de página, Then debería ver enlaces a las políticas de privacidad y términos de servicio. Escenario 02: Given que hago clic en el enlace de política de privacidad, When se abre el documento, Then debería ver la información completa de privacidad en una nueva página o ventana.	EP13
US15	Inicio de Sesión para Conductores	Como developer de mobile Application de PARKINGNOW, necesito implementar la funcionalidad de inicio de sesión para conductores para que puedan acceder a sus cuentas de manera segura.	Escenario 01: Given que el conductor ingresa su correo y contraseña, When hace clic en "Iniciar sesión", Then debe acceder a su dashboard de conductor si las credenciales son correctas. Escenario 02: Given que el conductor ingresa credenciales incorrectas, When intenta iniciar sesión, Then debe mostrarse un mensaje de error indicando que las credenciales son inválidas.	EP11
US16	Registro de Conductores	Como developer de mobile Application de PARKINGNOW, necesito implementar el registro de conductores para que puedan crear una cuenta en la plataforma.	Escenario 01: Given que un nuevo usuario quiere registrarse, When ingresa los datos necesarios y hace clic en "Registrarse", Then debe recibir un mensaje de confirmación de cuenta creada. Escenario 02: Given que el usuario intenta registrarse sin llenar los campos requeridos, When intenta completar el registro, Then debe mostrarse un mensaje de error solicitando los datos faltantes.	EP02
US17	Registro de Dueños de estacionamiento	Como developer de mobile Application de PARKINGNOW, necesito implementar el registro de dueños de estacionamiento para que puedan crear una cuenta y gestionar su estacionamiento.	Escenario 01: Given que un dueño de estacionamiento quiere registrarse, When llena el formulario de registro y hace clic en "Registrarse", Then debe recibir un mensaje de confirmación de cuenta creada. Escenario 02: Given que el formulario no está completo, When intenta registrarse, Then debe mostrarse un mensaje de error indicando los campos faltantes.	EP02
US18	Búsqueda de Reservas para Conductores	Como developer de mobile Application de PARKINGNOW, necesito implementar la búsqueda de reservas para que los conductores puedan ver sus reservas activas y pasadas.	Escenario 01: Given que el conductor quiere ver sus reservas, When ingresa al menú de reservas, Then debe ver una lista de todas sus reservas activas y completadas. Escenario 02: Given que el conductor busca una reserva específica, When ingresa un criterio de búsqueda, Then debe mostrarse la reserva correspondiente si existe.	EP04
US19	Pago de Reservas por Transacción	Como developer de mobile Application de PARKINGNOW, necesito implementar el pago por transacción para que los conductores puedan completar sus reservas.	Escenario 01: Given que el conductor realiza una reserva, When selecciona "Pagar", Then debe ser dirigido a la pasarela de pago para completar la transacción. Escenario 02: Given que el pago es exitoso, When se procesa el pago, Then debe recibir una confirmación de reserva y recibo de pago.	EP12
US20	Confirmación de Reserva Registrada	Como developer de mobile Application de PARKINGNOW, necesito que se muestre una pantalla de confirmación para asegurar que el usuario sepa	Escenario 01: Given que el usuario completa una reserva, When el sistema confirma la transacción, Then debe ver una pantalla indicando que su reserva fue registrada con éxito. Escenario 02: Given que el sistema no puede confirmar la reserva, When se produce un error en el pago,	EP12

Epic / Story ID	Título	Descripción	Criterios de Aceptación	Relacionado con (Epic ID)
		que su reserva fue registrada correctamente.	Then debe mostrarse un mensaje solicitando un nuevo intento de pago.	
US21	Registro de Opiniones	Como developer de mobile Application de PARKINGNOW, necesito implementar una funcionalidad para registrar opiniones para que los usuarios puedan dar retroalimentación sobre sus experiencias.	Escenario 01: Given que el usuario quiere dejar una opinión, When completa el formulario de opinión y hace clic en "Enviar", Then debe ver un mensaje de confirmación de que su opinión fue enviada. Escenario 02: Given que el usuario intenta enviar una opinión sin completar los campos requeridos, When hace clic en "Enviar", Then debe recibir un mensaje indicando que los campos son obligatorios.	EP11
US22	Notificaciones en el Dashboard del Conductor	Como developer de mobile Application de PARKINGNOW, necesito implementar notificaciones en el dashboard del conductor para que reciban alertas relevantes como ofertas o recordatorios de reservas.	Escenario 01: Given que el conductor ingresa a su dashboard, When hay notificaciones pendientes, Then debe ver un ícono indicando nuevas notificaciones. Escenario 02: Given que el conductor visualiza una notificación, When hace clic en ella, Then debe acceder a detalles adicionales o redirigirse a la página correspondiente.	EP09
US23	Configuración de Preferencias de Usuario	Como developer de mobile Application de PARKINGNOW, necesito implementar la configuración de preferencias para que los usuarios puedan personalizar su experiencia en la plataforma.	Escenario 01: Given que el usuario quiere modificar sus preferencias, When accede a "Configuración" y guarda los cambios, Then su perfil debe actualizarse con las nuevas preferencias. Escenario 02: Given que el usuario cambia sus preferencias de notificación, When guarda la configuración, Then debe recibir notificaciones de acuerdo con sus preferencias.	EP08
US24	Historial de Reservas en el Dashboard del Conductor	Como developer de mobile Application de PARKINGNOW, necesito implementar el historial de reservas para que los conductores puedan consultar sus reservas pasadas.	Escenario 01: Given que el conductor quiere revisar su historial, When accede a la sección de historial de reservas, Then debe ver una lista de todas sus reservas previas. Escenario 02: Given que el conductor busca una reserva específica en el historial, When ingresa una fecha o nombre de estacionamiento, Then debería encontrar la reserva correspondiente.	EP04
US25	Alerta de Espacio Disponible	Como developer de mobile Application de PARKINGNOW, necesito implementar alertas de disponibilidad de espacios para notificar a los conductores cuando un espacio de interés esté disponible.	Escenario 01: Given que el conductor marca un estacionamiento como favorito, When un espacio en ese estacionamiento se desocupa, Then debería recibir una notificación en su dashboard. Escenario 02: Given que el conductor recibe una alerta de disponibilidad, When hace clic en la notificación, Then debería ser redirigido a la página de reserva para ese espacio.	EP09
US26	Soporte en Tiempo Real en el Dashboard del Conductor	Como developer de mobile Application de PARKINGNOW, necesito implementar un sistema de soporte en tiempo real para que los conductores puedan resolver dudas o problemas rápidamente.	Escenario 01: Given que el conductor necesita ayuda, When accede a la sección de soporte en el dashboard, Then debería ver opciones de contacto o un chat en tiempo real. Escenario 02: Given que el conductor inicia una consulta de soporte, When envía su mensaje, Then debería recibir una respuesta o ver un mensaje de confirmación de recibido.	EP07
US27	Favoritos para Conductores	Como developer de mobile Application de PARKINGNOW, necesito permitir que los conductores marquen estacionamientos como	Escenario 01: Given que el conductor quiere agregar un estacionamiento a sus favoritos, When hace clic en el botón "Favorito" en el detalle del estacionamiento, Then debería ver el estacionamiento en su lista de favoritos. Escenario 02: Given que el conductor quiere	EP04

Epic / Story ID	Título	Descripción	Criterios de Aceptación	Relacionado con (Epic ID)
		favoritos para facilitar futuras reservas.	reservar un estacionamiento favorito, When accede a su lista de favoritos, Then debería ver una opción para hacer la reserva directamente.	
US28	Verificación de Reserva en el Dashboard del Conductor	Como developer de mobile Application de PARKINGNOW, necesito implementar una opción para verificar el estado de la reserva para que los conductores puedan confirmar si su espacio está listo.	Escenario 01: Given que el conductor quiere verificar el estado de su reserva, When accede a la sección de reservas activas, Then debe ver un estado que indique si la reserva está confirmada o en proceso. Escenario 02: Given que el estado de la reserva cambia, When el conductor accede a la reserva, Then debe ver la información actualizada de inmediato.	EP05
US29	Cancelación de Reserva en el Dashboard	Como developer de mobile Application de PARKINGNOW, necesito implementar la opción de cancelación de reserva para que los conductores puedan cancelar en caso de cambios de planes.	Escenario 01: Given que el conductor tiene una reserva activa, When selecciona la opción de cancelación, Then debería ver una confirmación de cancelación y una notificación de reembolso si aplica. Escenario 02: Given que el conductor cancela una reserva, When intenta reservar el mismo espacio de nuevo, Then el sistema debe permitir la nueva reserva si el espacio sigue disponible.	EP05
US30	Gestión de Alertas para Conductores	Como developer de mobile Application de PARKINGNOW, necesito implementar la gestión de alertas para que los conductores puedan ver y gestionar alertas de seguridad o disponibilidad en el dashboard.	Escenario 01: Given que el conductor tiene alertas pendientes, When accede a la sección de alertas en su dashboard, Then debe ver una lista de alertas recientes. Escenario 02: Given que el conductor revisa una alerta específica, When hace clic en ella, Then debería ver detalles adicionales o sugerencias de acción.	EP09
US31	Personalización de Notificaciones para Conductores	Como developer de mobile Application de PARKINGNOW, necesito permitir que los conductores configuren sus preferencias de notificación para recibir solo las alertas que les interesan.	Escenario 01: Given que el conductor quiere personalizar sus notificaciones, When ajusta sus preferencias en la configuración, Then solo debería recibir las notificaciones seleccionadas. Escenario 02: Given que el conductor cambia sus preferencias, When guarda la configuración, Then las futuras notificaciones deberían respetar estas preferencias.	EP08
US32	Visualización de Mapa en el Dashboard del Conductor	Como developer de mobile Application de PARKINGNOW, necesito integrar un mapa interactivo en el dashboard para que los conductores vean ubicaciones de estacionamientos cercanos en tiempo real.	Escenario 01: Given que el conductor accede a su dashboard, When abre la sección de mapa, Then debería ver ubicaciones de estacionamientos cercanos con disponibilidad en tiempo real. Escenario 02: Given que el conductor selecciona una ubicación en el mapa, When hace clic en ella, Then debería ver opciones para realizar una reserva en esa ubicación.	EP10
US33	Seguimiento de Reserva en Tiempo Real	Como developer de mobile Application de PARKINGNOW, necesito implementar una funcionalidad de seguimiento en tiempo real para que los conductores puedan ver el estado de su reserva activa en todo momento.	Escenario 01: Given que el conductor tiene una reserva activa, When accede a la sección de seguimiento, Then debería ver el estado en tiempo real de su reserva. Escenario 02: Given que el estado de la reserva cambia, When el conductor revisa la reserva, Then debería ver el cambio reflejado de inmediato.	EP05
US34	Gestión de Clientes en	Como developer de mobile services, necesito crear un	Escenario 01: Given que se recibe una consulta de un cliente, When el administrador accede al endpoint	EP07

Epic / Story ID	Título	Descripción	Criterios de Aceptación	Relacionado con (Epic ID)
	Servicio al Cliente	endpoint para gestionar consultas y solicitudes de servicio al cliente para permitir soporte eficaz a los usuarios.	/api/client-service , Then debería ver todas las solicitudes pendientes con detalles. Escenario 02: Given que un cliente envía una nueva solicitud, When se crea una nueva consulta en el endpoint, Then debería notificarse al equipo de servicio al cliente.	
US35	Notificación de Cambios en la Disponibilidad de Espacios	Como developer de mobile services, necesito implementar un endpoint que envíe notificaciones automáticas cuando un espacio cambie de estado para mantener a los usuarios informados en tiempo real.	Escenario 01: Given que un espacio de estacionamiento se desocupa, When el sistema detecta el cambio en el endpoint /api/space-notifications , Then debería enviar una notificación a los usuarios interesados. Escenario 02: Given que un usuario tiene una reserva, When su espacio cambia de estado, Then debería recibir una notificación push y por correo electrónico.	EP09
US36	Historial de Transacciones para Usuarios	Como developer de mobile services, necesito desarrollar un endpoint para consultar el historial de transacciones de cada usuario para permitir la revisión de pagos y reservas anteriores.	Escenario 01: Given que un usuario desea ver su historial de transacciones, When accede al endpoint /api/user-transactions , Then debería ver una lista de todas sus transacciones ordenadas por fecha. Escenario 02: Given que el administrador necesita revisar las transacciones de un usuario específico, When ingresa el ID del usuario en el endpoint, Then debería obtener el historial detallado del usuario.	EP13
US37	Gestión de Ocupación de Espacios en Tiempo Real	Como developer de mobile services, necesito implementar un endpoint para actualizar automáticamente la disponibilidad de espacios para reflejar cambios en tiempo real según las cámaras de monitoreo visual.	Escenario 01: Given que un espacio cambia de estado, When el sistema detecta el cambio mediante las cámaras de monitoreo visual en el endpoint /api/space-occupancy , Then el sistema debería reflejar el estado actualizado en tiempo real. Escenario 02: Given que se ocupa un espacio sin reserva, When las cámaras detectan la ocupación, Then el sistema debería generar una alerta automática al administrador.	EP09
US38	Configuración de Alertas de Seguridad	Como developer de mobile services, necesito implementar un endpoint para que los administradores configuren alertas de seguridad para responder rápidamente a eventos sospechosos en los espacios de estacionamiento.	Escenario 01: Given que un administrador configura alertas, When ingresa criterios de alerta en el endpoint /api/security-alerts , Then debería poder definir parámetros específicos como hora, ubicación y tipo de alerta. Escenario 02: Given que se detecta un evento de seguridad, When se activa una alerta, Then el sistema debe enviar notificaciones automáticas a los responsables.	EP13
US39	Administración de Tarifas por Ubicación	Como developer de mobile services, necesito implementar un endpoint para configurar tarifas de estacionamiento basadas en la ubicación para que los administradores ajusten precios según la demanda.	Escenario 01: Given que un administrador necesita ajustar tarifas, When accede al endpoint /api/parking-rates , Then debería poder configurar tarifas por ciudad o zona. Escenario 02: Given que una tarifa es modificada, When se actualiza en el sistema, Then todos los precios de los espacios correspondientes deberían reflejar el cambio automáticamente.	EP12
US40	Registro de Alertas de Seguridad	Como developer de mobile services, necesito un endpoint para registrar todas las alertas de seguridad detectadas en los estacionamientos para que los	Escenario 01: Given que ocurre un incidente de seguridad, When el sistema lo registra en el endpoint /api/security-log , Then debería almacenar la fecha, hora, ubicación y tipo de incidente. Escenario 02: Given que un administrador revisa el historial de alertas,	EP13

Epic / Story ID	Título	Descripción	Criterios de Aceptación	Relacionado con (Epic ID)
		administradores revisen incidentes pasados.	When solicita los registros, Then debería ver una lista cronológica de los eventos de seguridad.	
US41	Gestión de Datos de Vehículos	Como developer de mobile services, necesito un endpoint para almacenar y gestionar información de los vehículos registrados para poder hacer seguimiento y gestionar accesos.	Escenario 01: Given que un usuario registra un vehículo, When ingresa los detalles en el endpoint /api/vehicle-management , Then debería guardarse en la base de datos. Escenario 02: Given que un administrador necesita actualizar datos de un vehículo, When accede a este endpoint, Then debería poder editar o eliminar la información del vehículo.	EP10
US42	Reportes de Promociones Utilizadas	Como developer de mobile services, necesito un endpoint para generar reportes de promociones utilizadas para analizar la efectividad de las promociones y optimizar futuras campañas.	Escenario 01: Given que un administrador solicita un reporte de promociones, When accede al endpoint /api/promotion-reports , Then debería ver un resumen con el uso de cada promoción. Escenario 02: Given que se requiere un reporte detallado, When el administrador filtra por fecha y ubicación, Then debería obtener un desglose por cada categoría seleccionada.	EP05
US43	Auditoría de Acciones en la Plataforma	Como developer de mobile services, necesito un endpoint para registrar y auditar todas las acciones críticas de los usuarios y administradores para garantizar la seguridad y trazabilidad de las operaciones.	Escenario 01: Given que un usuario realiza una acción crítica, When el sistema registra la acción en el endpoint /api/audit-log , Then debería almacenarse con detalles de la fecha, hora, usuario y tipo de acción. Escenario 02: Given que un auditor revisa el historial, When filtra por tipo de acción o usuario, Then debería obtener un listado preciso de todas las acciones que cumplen con los filtros.	EP13
US44	Historial de Opiniones de Usuarios	Como developer de mobile services, necesito implementar un endpoint para registrar y consultar el historial de opiniones de los usuarios para mejorar la experiencia de usuario a través del feedback.	Escenario 01: Given que un usuario deja una opinión, When el sistema la guarda en el endpoint /api/user-reviews , Then debería incluir detalles del usuario, calificación y comentario. Escenario 02: Given que un administrador consulta el historial de opiniones, When accede al endpoint, Then debería ver una lista cronológica de todas las opiniones con detalles.	EP11
US45	Integración con el Sistema de Pago	Como developer de mobile services, necesito un endpoint para procesar pagos y verificar transacciones para asegurar que todos los pagos de reservas se realicen de forma segura.	Escenario 01: Given que un usuario realiza un pago, When procesa la transacción en el endpoint /api/payment-processing , Then debería generar una respuesta de éxito o error según el resultado. Escenario 02: Given que ocurre un error en el pago, When el sistema detecta la falla, Then debería enviar un mensaje de error detallado y sugerencias de resolución.	EP13
US46	Generación de Reportes de Ocupación	Como developer de mobile services, necesito un endpoint para generar reportes de ocupación de los estacionamientos para analizar el uso y optimizar la disponibilidad de espacios.	Escenario 01: Given que un administrador solicita un reporte, When accede al endpoint /api/occupancy-reports , Then debería ver el porcentaje de ocupación por ubicación y fecha. Escenario 02: Given que se necesita un análisis de tendencias, When el administrador filtra por período, Then debería obtener una representación gráfica de la ocupación.	EP06
US47	Configuración de Opciones de Notificaciones	Como developer de mobile services, necesito implementar un endpoint para que los usuarios configuren sus preferencias de notificaciones	Escenario 01: Given que un usuario ajusta sus preferencias de notificaciones, When accede al endpoint /api/notification-preferences , Then debería poder activar o desactivar tipos de notificación específicos. Escenario 02: Given que el sistema envía notificaciones, When las preferencias están activadas, Then el usuario	EP09

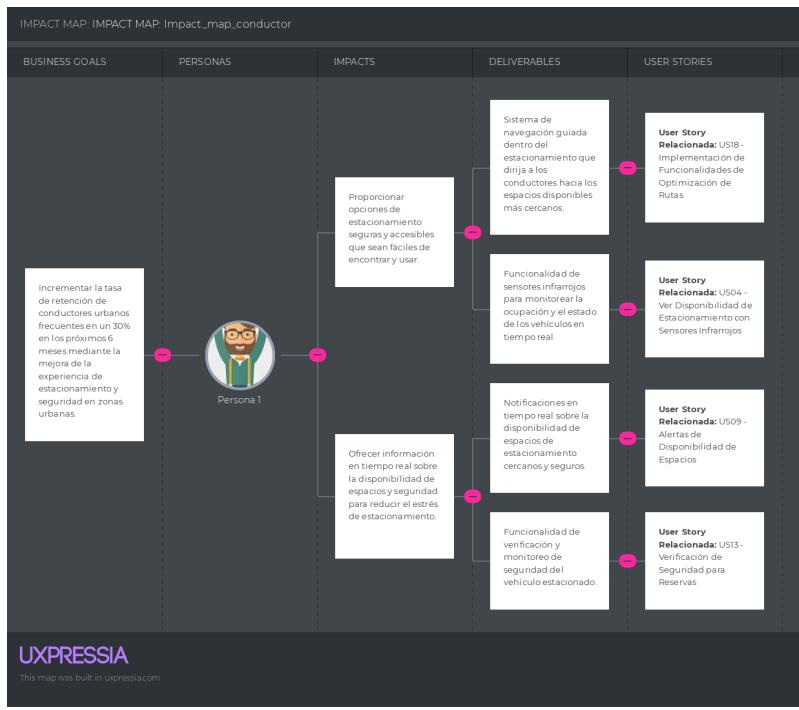
Epic / Story ID	Título	Descripción	Criterios de Aceptación	Relacionado con (Epic ID)
		para personalizar los avisos que desean recibir.	debería recibir solo las notificaciones configuradas en su perfil.	
US48	API de Feedback Automático	Como developer de mobile services, necesito implementar un endpoint para recopilar feedback de los usuarios al finalizar su reserva para obtener insights sobre su experiencia de usuario.	Escenario 01: Given que un usuario completa una reserva, When el sistema solicita feedback en el endpoint /api/automatic-feedback , Then debería almacenar los comentarios para análisis posterior. Escenario 02: Given que el administrador consulta el feedback, When revisa los comentarios en el endpoint, Then debería ver un resumen con estadísticas de satisfacción.	EP11
US49	Gestión de Reembolsos	Como developer de mobile services, necesito implementar un endpoint para gestionar solicitudes de reembolso para asegurar que los usuarios puedan solicitar reembolsos fácilmente en caso de problemas.	Escenario 01: Given que un usuario solicita un reembolso, When el sistema procesa la solicitud en el endpoint /api/refunds , Then debería generar una respuesta de éxito o error según el caso. Escenario 02: Given que se requiere seguimiento de reembolsos, When el administrador consulta el historial de reembolsos, Then debería ver un listado con detalles de cada solicitud.	EP13
US50	Auditoría de Cambios de Tarifas	Como developer de mobile services, necesito un endpoint para auditar todas las modificaciones de tarifas en la plataforma para asegurar un control y seguimiento sobre los cambios de precios.	Escenario 01: Given que un administrador modifica una tarifa, When se registra el cambio en el endpoint /api/rate-audit , Then debería almacenar la fecha, hora y detalles de la tarifa modificada. Escenario 02: Given que un auditor revisa el historial de cambios de tarifas, When solicita un reporte, Then debería obtener un listado con todas las modificaciones realizadas.	EP12

3.3. Impact Mapping

Descripción del Impact Map: Conductor Urbano Frecuente para PARKINGNOW

Este Impact Map de PARKINGNOW se enfoca en aumentar el número de conductores urbanos frecuentes que siguen utilizando el servicio, mejorando la seguridad y accesibilidad en los estacionamientos de zonas urbanas. El objetivo es aumentar el uso del servicio en un 30%, proponiendo soluciones como sistemas de navegación más eficientes, cámaras de monitoreo visual para actualización en tiempo real y notificaciones sobre espacios disponibles. Las historias de usuario se centran en mejorar las rutas, garantizar la seguridad y gestionar las alertas, ofreciendo una mejor experiencia a los conductores.

Mejorando rutas y seguridad para el conductor urbano.



Descripción del Impact Map: Dueño de Estacionamiento para PARKINGNOW

El Impact Map para los dueños de estacionamientos está diseñado para optimizar la gestión de sus espacios y aumentar la satisfacción de los clientes en un 30% durante los próximos seis meses. Las soluciones propuestas incluyen un sistema de reservas en tiempo real, herramientas visuales para monitorear la disponibilidad de espacios en tiempo real mediante cámaras, y funcionalidades para obtener opiniones de los usuarios. Estas funciones ayudan a los propietarios a maximizar el uso de los espacios, reducir tiempos de espera y mejorar la calidad del servicio basado en el feedback directo de los clientes.

Gestión optimizada para administradores de espacios.



3.4. Product Backlog

Utilizamos la escala de Fibonacci para la estimación de los Story Points.

# Orden	User Story ID	Título	Descripción	Story Points (1/2/3/5/8)
1	US01	Navegación Intuitiva en la Landing Page	Como visitante de PARKINGNOW, deseo que la landing page tenga una barra de navegación clara y accesible para encontrar fácilmente las secciones importantes.	5
2	US02	Visualización de Ciudades con Servicio	Como visitante, deseo ver en la landing page las ciudades donde PARKINGNOW está disponible para saber si puedo utilizar el servicio en mi ubicación.	3
3	US03	Promociones Destacadas	Como visitante, deseo ver promociones destacadas en la landing page para aprovechar descuentos y beneficios en las ciudades disponibles.	5
4	US04	Beneficios para Conductores y Dueños de Estacionamiento	Como visitante, deseo ver una sección en la landing page que me explique los beneficios de usar PARKINGNOW para entender las ventajas que ofrece.	3
5	US05	Visualización de Misión de la Empresa	Como visitante, deseo ver la misión de PARKINGNOW en la landing page para entender los objetivos de la empresa.	2
6	US06	Visualización de Visión de la Empresa	Como visitante, deseo ver la visión de PARKINGNOW en la landing page para conocer sus metas futuras y crecimiento.	2
7	US07	Visualización del Equipo	Como visitante, deseo ver una sección que muestre a los miembros del equipo de PARKINGNOW para conocer quiénes están detrás del proyecto.	3
8	US08	Información de Contacto	Como visitante, deseo ver una sección de contacto en la landing page para saber cómo comunicarme con PARKINGNOW si tengo preguntas o comentarios.	3
9	US09	Envío de Mensaje de Contacto	Como visitante, deseo poder enviar un mensaje a PARKINGNOW desde la landing page para hacer preguntas o dar feedback.	5
10	US10	Visualización de Información de Características	Como visitante, deseo ver las características de PARKINGNOW en la landing page para entender qué ofrece el servicio.	3
11	US11	Botón de Registro e Inicio de Sesión	Como visitante, deseo ver botones destacados de "Registrarse" e "Iniciar Sesión" en la landing page para acceder a la plataforma o crear una cuenta fácilmente.	5
12	US12	Enlace a Redes Sociales	Como visitante, deseo ver enlaces a las redes sociales de PARKINGNOW en la landing page para seguir sus novedades y actualizaciones.	2
13	US13	Visualización de Slogan	Como visitante, deseo ver un slogan atractivo en la landing page de PARKINGNOW para entender de inmediato la propuesta del servicio.	2
14	US14	Acceso a Información Legal	Como visitante, deseo ver enlaces a la información legal y políticas de privacidad en la landing page para revisar los términos de uso y la protección de mis datos.	3
15	US15	Inicio de Sesión para Conductores	Como developer de mobile Application de PARKINGNOW, necesito implementar la funcionalidad de inicio de sesión para conductores para que puedan acceder a sus cuentas de manera segura.	5
16	US16	Registro de Conductores	Como developer de mobile Application de PARKINGNOW, necesito implementar el registro de conductores para que puedan crear una cuenta en la plataforma.	5
17	US17	Registro de Dueños de Estacionamiento	Como developer de mobile Application de PARKINGNOW, necesito implementar el registro de dueños de estacionamiento para que puedan	5

# Orden	User Story ID	Título	Descripción	Story Points (1/2/3/5/8)
			crear una cuenta y gestionar su estacionamiento.	
18	US18	Búsqueda de Reservas para Conductores	Como developer de mobile Application de PARKINGNOW, necesito implementar la búsqueda de reservas para que los conductores puedan ver sus reservas activas y pasadas.	3
19	US19	Pago de Reservas por Transacción	Como developer de mobile Application de PARKINGNOW, necesito implementar el pago por transacción para que los conductores puedan completar sus reservas.	5
20	US20	Confirmación de Reserva Registrada	Como developer de mobile Application de PARKINGNOW, necesito que se muestre una pantalla de confirmación para asegurar que el usuario sepa que su reserva fue registrada correctamente.	3
21	US21	Registro de Opiniones	Como developer de mobile Application de PARKINGNOW, necesito implementar una funcionalidad para registrar opiniones para que los usuarios puedan dar retroalimentación sobre sus experiencias.	3
22	US22	Notificaciones en el Dashboard del Conductor	Como developer de mobile Application de PARKINGNOW, necesito implementar notificaciones en el dashboard del conductor para que reciban alertas relevantes como ofertas o recordatorios de reservas.	5
23	US23	Configuración de Preferencias de Usuario	Como developer de mobile Application de PARKINGNOW, necesito implementar la configuración de preferencias para que los usuarios puedan personalizar su experiencia en la plataforma.	3
24	US24	Historial de Reservas en el Dashboard del Conductor	Como developer de mobile Application de PARKINGNOW, necesito implementar el historial de reservas para que los conductores puedan consultar sus reservas pasadas.	3
25	US25	Alerta de Espacio Disponible	Como developer de mobile Application de PARKINGNOW, necesito implementar alertas de disponibilidad de espacios para notificar a los conductores cuando un espacio de interés esté disponible.	5
26	US26	Soporte en Tiempo Real en el Dashboard del Conductor	Como developer de mobile Application de PARKINGNOW, necesito implementar un sistema de soporte en tiempo real para que los conductores puedan resolver dudas o problemas rápidamente.	5
27	US27	Favoritos para Conductores	Como developer de mobile Application de PARKINGNOW, necesito permitir que los conductores marquen estacionamientos como favoritos para facilitar futuras reservas.	3
28	US28	Verificación de Reserva en el Dashboard del Conductor	Como developer de mobile Application de PARKINGNOW, necesito implementar una opción para verificar el estado de la reserva para que los conductores puedan confirmar si su espacio está listo.	3
29	US29	Cancelación de Reserva en el Dashboard	Como developer de mobile Application de PARKINGNOW, necesito implementar la opción de cancelación de reserva para que los conductores puedan cancelar en caso de cambios de planes.	5
30	US30	Gestión de Alertas para Conductores	Como developer de mobile Application de PARKINGNOW, necesito implementar la gestión de alertas para que los conductores puedan ver y gestionar alertas de seguridad o disponibilidad en el dashboard.	5
31	US31	Personalización de Notificaciones para Conductores	Como developer de mobile Application de PARKINGNOW, necesito permitir que los conductores configuren sus preferencias de notificación para recibir solo las alertas que les interesan.	3

# Orden	User Story ID	Título	Descripción	Story Points (1/2/3/5/8)
32	US32	Visualización de Mapa en el Dashboard del Conductor	Como developer de mobile Application de PARKINGNOW, necesito integrar un mapa interactivo en el dashboard para que los conductores vean ubicaciones de estacionamientos cercanos en tiempo real.	5
33	US33	Seguimiento de Reserva en Tiempo Real	Como developer de mobile Application de PARKINGNOW, necesito implementar una funcionalidad de seguimiento en tiempo real para que los conductores puedan ver el estado de su reserva activa en todo momento.	5
34	US34	Gestión de Clientes en Servicio al Cliente	Como developer de mobile services, necesito crear un endpoint para gestionar consultas y solicitudes de servicio al cliente para permitir soporte eficaz a los usuarios.	3
35	US35	Notificación de Cambios en la Disponibilidad de Espacios	Como developer de mobile services, necesito implementar un endpoint que envíe notificaciones automáticas cuando un espacio cambie de estado para mantener a los usuarios informados en tiempo real.	5
36	US36	Historial de Transacciones para Usuarios	Como developer de mobile services, necesito desarrollar un endpoint para consultar el historial de transacciones de cada usuario para permitir la revisión de pagos y reservas anteriores.	3
37	US37	Gestión de Ocupación de Espacios en Tiempo Real	Como developer de mobile services, necesito implementar un endpoint para reflejar cambios en tiempo real según las cámaras de monitoreo visual	5
38	US38	Configuración de Alertas de Seguridad	Como developer de mobile services, necesito implementar un endpoint para que los administradores configuren alertas de seguridad para responder rápidamente a eventos sospechosos en los espacios de estacionamiento.	5
39	US39	Administración de Tarifas por Ubicación	Como developer de mobile services, necesito implementar un endpoint para configurar tarifas de estacionamiento basadas en la ubicación para que los administradores ajusten precios según la demanda.	5
40	US40	Registro de Alertas de Seguridad	Como developer de mobile services, necesito un endpoint para registrar todas las alertas de seguridad detectadas en los estacionamientos para que los administradores revisen incidentes pasados.	3
41	US41	Gestión de Datos de Vehículos	Como developer de mobile services, necesito un endpoint para almacenar y gestionar información de los vehículos registrados para poder hacer seguimiento y gestionar accesos.	3
42	US42	Reportes de Promociones Utilizadas	Como developer de mobile services, necesito un endpoint para generar reportes de promociones utilizadas para analizar la efectividad de las promociones y optimizar futuras campañas.	3
43	US43	Auditoría de Acciones en la Plataforma	Como developer de mobile services, necesito un endpoint para registrar y auditar todas las acciones críticas de los usuarios y administradores para garantizar la seguridad y trazabilidad de las operaciones.	5
44	US44	Historial de Opiniones de Usuarios	Como developer de mobile services, necesito implementar un endpoint para registrar y consultar el historial de opiniones de los usuarios para mejorar la experiencia de usuario a través del feedback.	3
45	US45	Integración con el Sistema de Pago	Como developer de mobile services, necesito un endpoint para procesar pagos y verificar transacciones para asegurar que todos los pagos de reservas se realicen de forma segura.	5
46	US46	Generación de Reportes de Ocupación	Como developer de mobile services, necesito un endpoint para generar reportes de ocupación de los estacionamientos para analizar el uso y	3

# Orden	User Story ID	Título	Descripción	Story Points (1/2/3/5/8)
			optimizar la disponibilidad de espacios.	
47	US47	Configuración de Opciones de Notificaciones	Como developer de mobile services, necesito implementar un endpoint para que los usuarios configuren sus preferencias de notificaciones para personalizar los avisos que desean recibir.	3
48	US48	API de Feedback Automático	Como developer de mobile services, necesito implementar un endpoint para recopilar feedback de los usuarios al finalizar su reserva para obtener insights sobre su experiencia de usuario.	3
49	US49	Gestión de Reembolsos	Como developer de mobile services, necesito implementar un endpoint para gestionar solicitudes de reembolso para asegurar que los usuarios puedan solicitar reembolsos fácilmente en caso de problemas.	3
50	US50	Auditoría de Cambios de Tarifas	Como developer de mobile services, necesito un endpoint para auditar todas las modificaciones de tarifas en la plataforma para asegurar un control y seguimiento sobre los cambios de precios.	8

Capítulo IV: Solution Software Design

4.1. Strategic-Level Domain-Driven Design

Esta sección describe el proceso de diseño estratégico aplicado al dominio del sistema, con el fin de identificar los límites naturales de los subsistemas que forman parte de la solución. A través de herramientas como EventStorming, Bounded Context Canvas y Context Mapping, se busca descomponer el sistema en áreas coherentes y bien delimitadas, facilitando una arquitectura modular y orientada al negocio.

4.1.1. EventStorming

EventStorming es una técnica colaborativa utilizada para explorar y comprender el dominio del problema mediante la identificación de eventos clave, acciones del sistema (comandos), actores externos y entidades centrales (aggregates). En esta sección se presentan los resultados de la sesión de EventStorming, la cual permitió mapear los flujos de interacción entre usuarios y sistemas, y sentar las bases para el diseño de los Bounded Contexts en etapas posteriores del proyecto.

EventStorming - Paso 1: Domain Events

Step 1: Events

Reserva
Creada

Estacionamiento
Registrado

Sensor
Actualizado

Espacio
Liberado

Reserva
Cancelada

Pago
Procesado

Espacio
Ocupado

Notificación
Enviada

Usuario
Registrado

EventStorming - Paso 2: Commands

Step 2: Commands

Registrar
Estacionamiento

Cancelar
Reserva

Crear
Reserva

Enviar
Notificación

Procesar
Pago

Actualizar
Sensor

Confirmar
Espacio

Registrar
Usuario

EventStorming - Paso 3: Aggregates



Step 3: Aggregates

Reserva

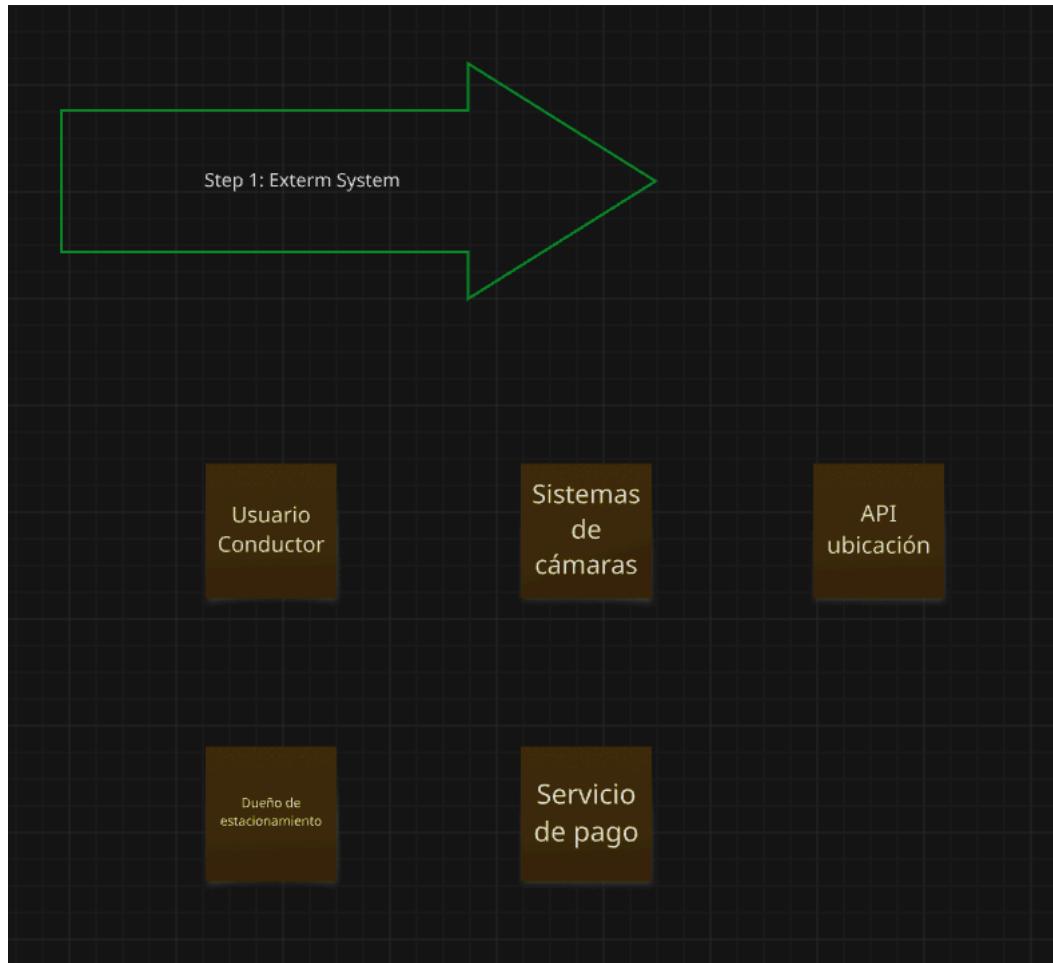
Sensor

Factura

Estacionamiento

Usuario

EventStorming - Paso 0: External Systems



4.1.1.1. Candidate Context Discovery

En esta sección se describe cómo se identificaron los principales bounded contexts del sistema ParkingNow.

Se utilizó la técnica de Event Storming para descubrir separaciones naturales basadas en las capacidades de negocio como el Monitoreo de Espacios, Gestión de Reservas y Administración de Usuarios, asegurando una estructura modular.

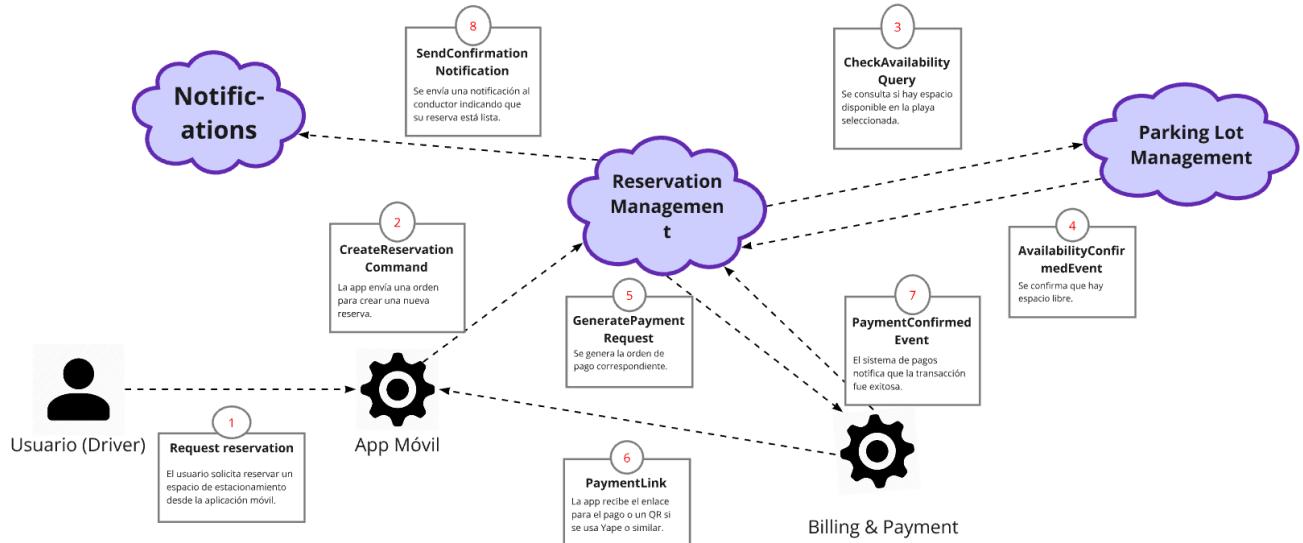
4.1.1.2. Domain Message Flows Modeling

Aquí se modelaron los flujos de mensajes entre los bounded contexts detectados.

Se aplicó Domain Storytelling para visualizar los eventos críticos como la actualización del estado de espacios, la confirmación de reservas y la notificación de disponibilidad a los usuarios.

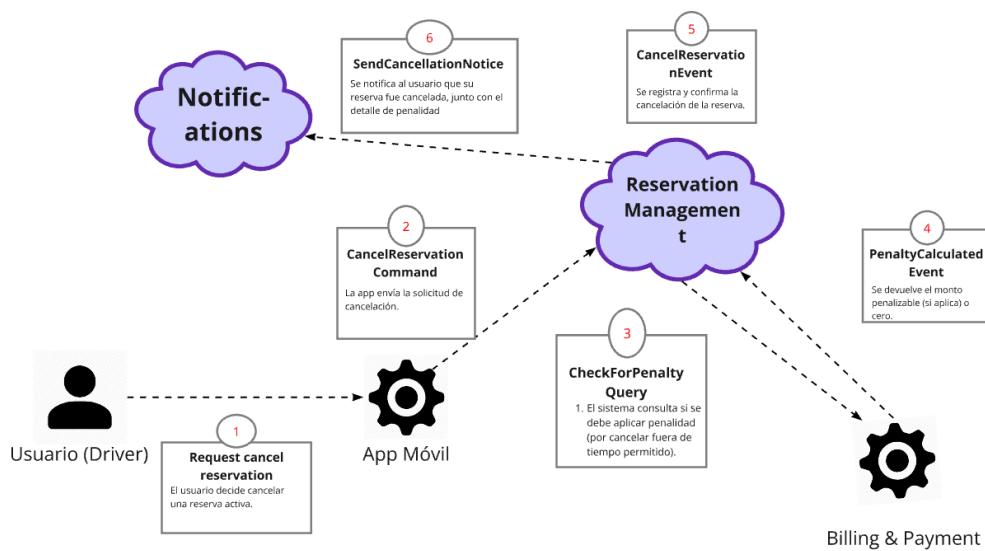
Gestión de reservas

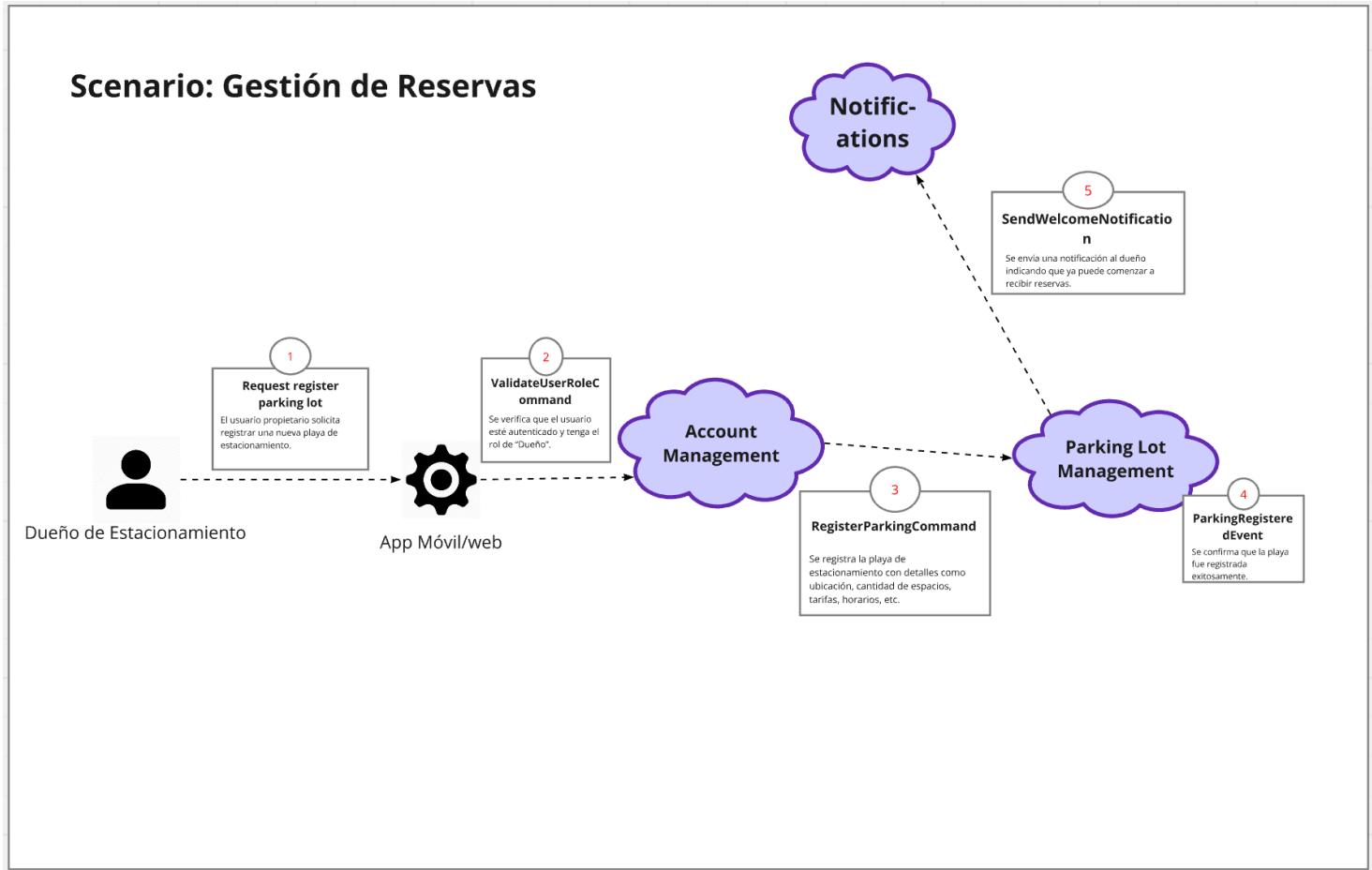
Scenario: Gestión de Reservas



Cancelación y penalidades

Scenario: Cancelación de Reserva y Gestión de Penalidad



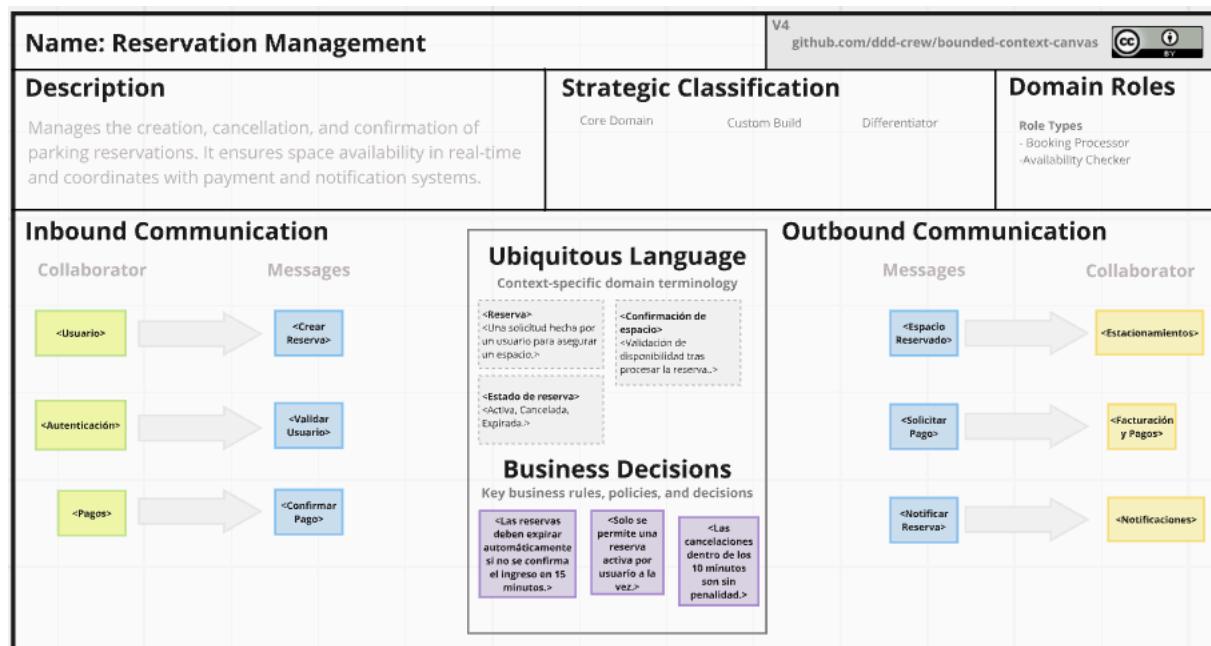


4.1.1.3. Bounded Context Canvases

En esta sección se presentan los Bounded Context Canvases desarrollados para ParkingNow.

Cada canvas detalla el propósito del contexto, sus comunicaciones inbound y outbound, lenguaje ubicuo y decisiones clave de negocio.

Reservation Management Context



Parking Space Monitoring Context

Name:

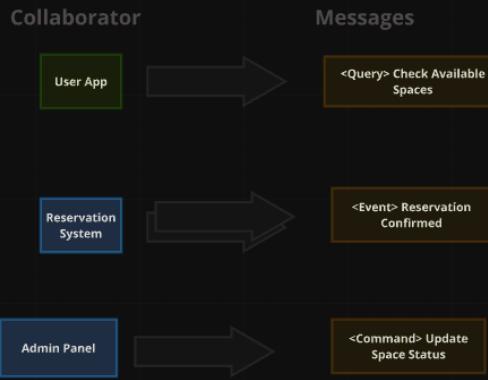
Name: Parking Space Monitoring

V4

github.com/ddd-crew/bounded-context-canvas

**Description**

Provides real-time monitoring of parking spaces using camera-based systems. It detects vehicle presence, availability, and changes in parking space status, supporting decision-making for reservations and space usage optimization.

Strategic ClassificationDomain
- coreBusiness Model
Engagement, Cost reductionEvolution
Engagement, Cost reduction**Domain Roles**Role Types
Execution context, Analysis context**Inbound Communication****Ubiquitous Language**

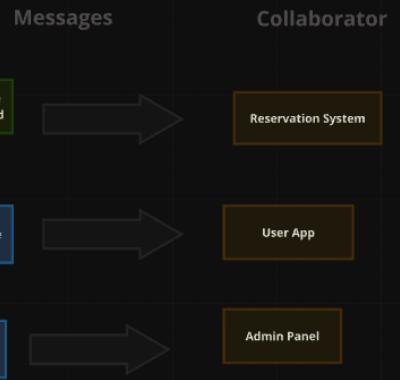
Context-specific domain terminology

<Available Space>
Free unit ready for reservation.
<Occupied Space>
Currently in use by a vehicle.
<Reservation>
Assigned parking space for a user.
<Visual Confirmation>
Camera-based image showing real-time status.
Time a space is in use.

Business Decisions

Key business rules, policies, and decisions

<Free space after no-show> <Decision>
Free space after 10 min no-show.
Block reservation for maintenance spaces.
Alert admin on status conflict.
Prioritize central spaces at peak hours.

Outbound Communication*Account Management Context*

Name: Account Management

V4

github.com/ddd-crew/bounded-context-canvas

**Description**

Handles user identity, registration, login, and profile management for both drivers and parking lot owners.

Strategic Classification

Generic

Security-Driven

Identity Provider

Domain RolesRole Types
- Usuario
- Autenticador**Inbound Communication****Ubiquitous Language**

Context-specific domain terminology

<Usuario Registrado>
<alta de cuenta>

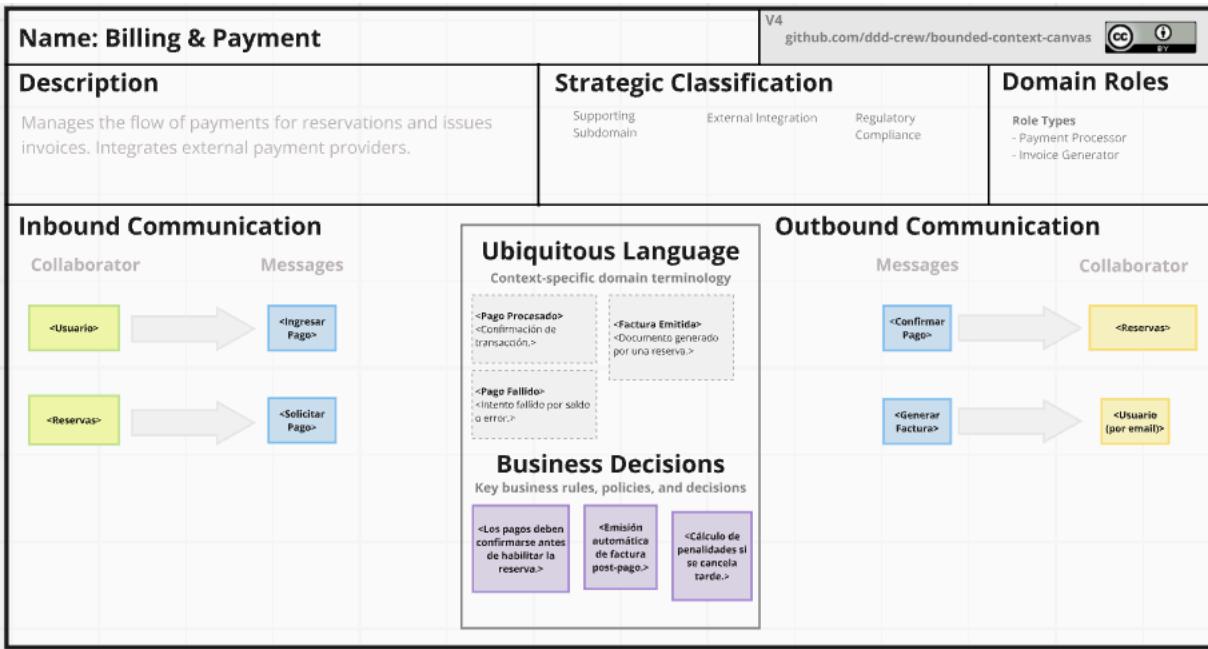
<Login Exitoso>
<Inicio de sesión válido.>

Business Decisions

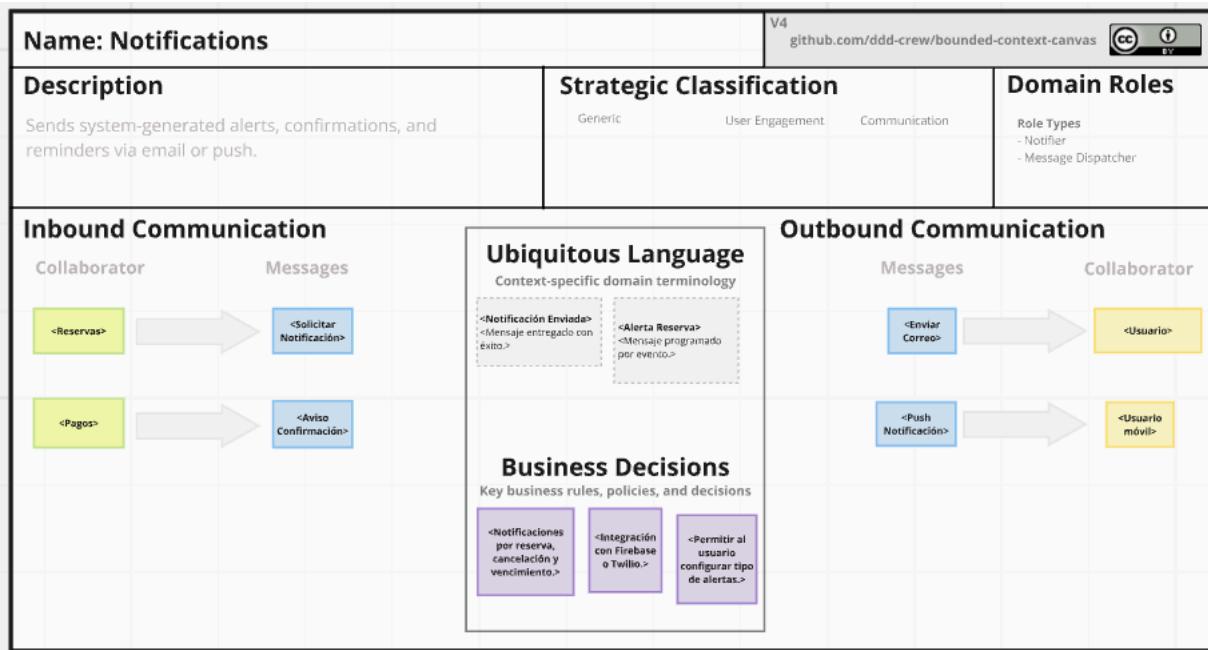
Key business rules, policies, and decisions

<Se requiere verificación de email para activar cuenta.>
<Contraseñas deben cumplir políticas de seguridad.>
<Soporte para OAuth en el futuro.>

Outbound Communication*Billing & Payment Context*



Notifications Context



Location Services Context

Name: Location Services		V4 github.com/ddd-crew/bounded-context-canvas	
Description	Strategic Classification	Domain Roles	
Provides geolocation features, maps, saved locations, and area-based triggers (geofencing).	Supporting Subdomain API-Driven UX Enhancement	Role Types - Geolocator - Mapainteractivo	

Inbound Communication

```

graph LR
    User[<User>] --> Buscar[<Buscar Ubicación>]
    Reservas[<Reservas>] --> Validar[<Validar Proximidad>]
  
```

Outbound Communication

```

graph LR
    Suggerir[<Sugerir Ubicación>] --> Usuario[<Usuario>]
    Actualizar[<Actualizar Mapas>] --> Interfaz[<Interfaz de usuario>]
  
```

Ubiquitous Language
Context-specific domain terminology

<Ubicación Actual> <Posición GPS del usuario.>	<Geocerca Activada> <Evento por entrada/salida en zona.>
<Ubicación Favorita> <Todos guardados.>	

Business Decisions
Key business rules, policies, and decisions

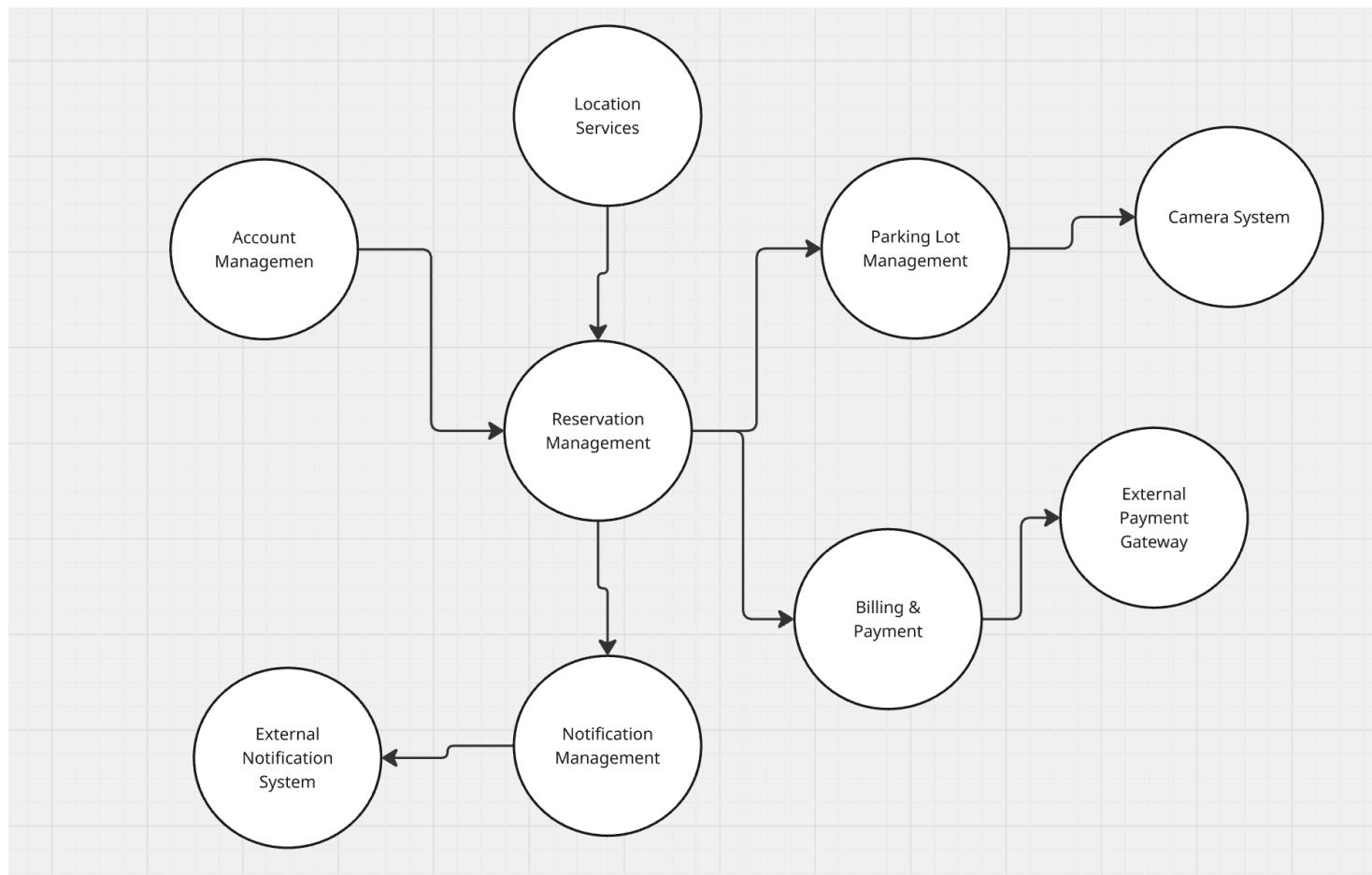
<Mostrar estaciones cercanas primero.>	<Registrar historial de ubicaciones para mejoras futuras.>	<Opcional compartir posición con contactos.>
----------------------------------------	------------------------------------------------------------	----------------------------------------------

4.1.2. Context Mapping

Esta sección muestra cómo los bounded contexts de ParkingNow se relacionan entre sí.

Se elaboraron context maps aplicando patrones como Customer/Supplier y Anti-Corruption Layer, para definir las dependencias y responsabilidades entre contextos.

Relación entre módulos funcionales de ParkingNow.



4.1.3. Software Architecture

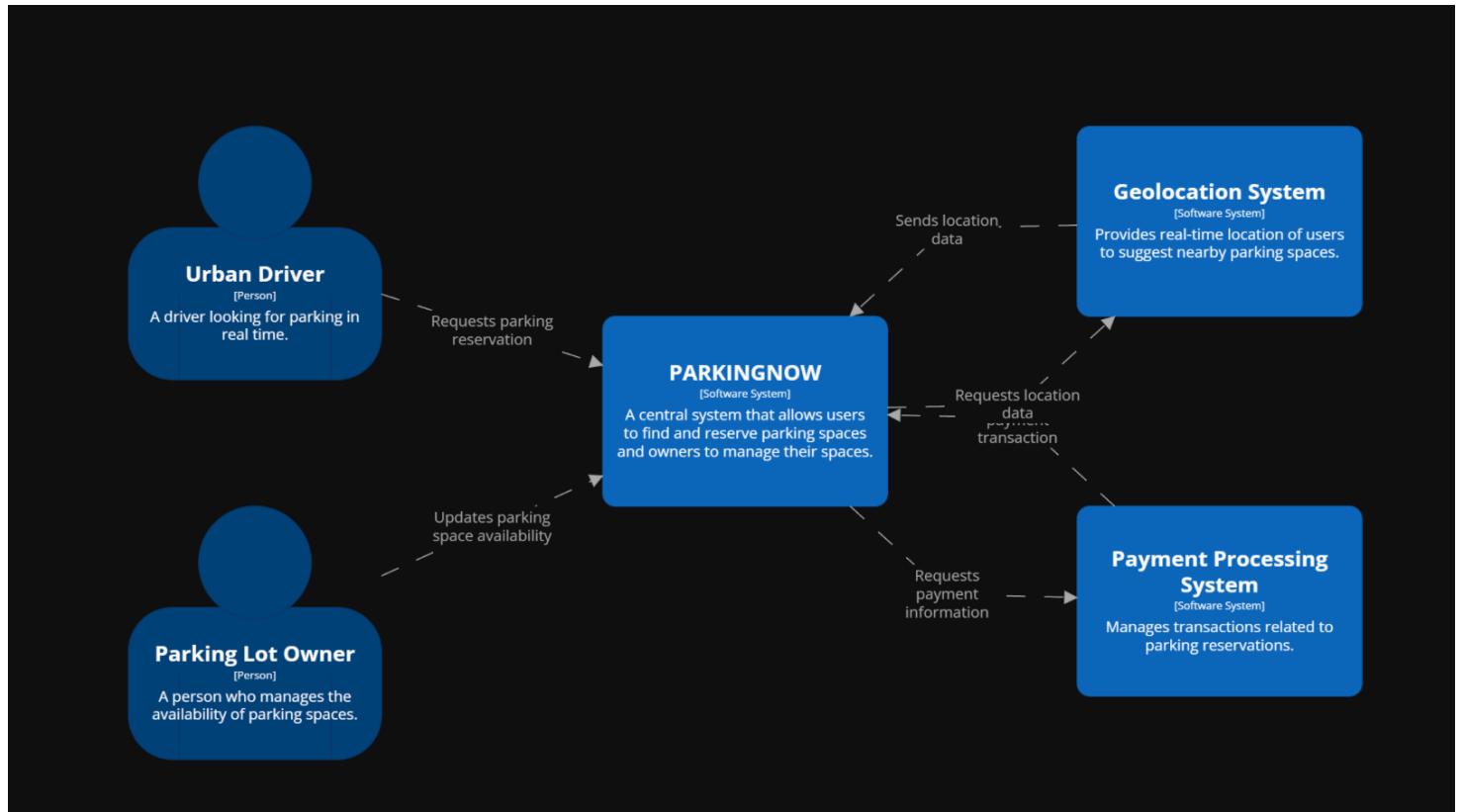
Aquí se describe la arquitectura general de software de ParkingNow siguiendo el modelo C4.

Se explican los niveles Contexto, Contenedor y Despliegue, asegurando una visión clara de la solución.

4.1.3.1. Software Architecture Context Level Diagrams

Se presenta el diagrama de nivel de contexto, mostrando ParkingNow en el centro y sus interacciones principales con usuarios, otros sistemas internos y servicios externos.

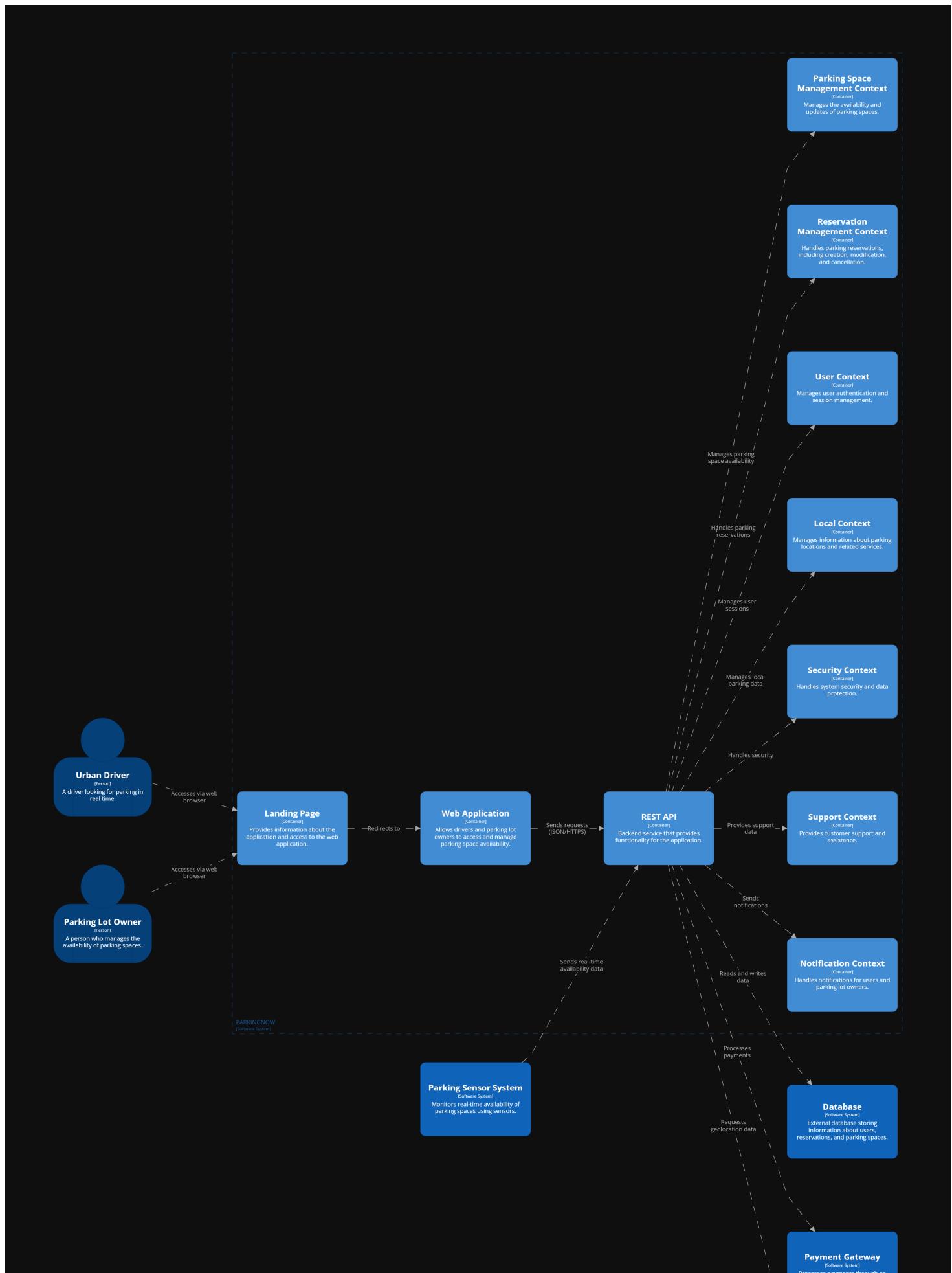
Interacción de ParkingNow con usuarios y servicios externos.



4.1.3.2. Software Architecture Container Level Diagrams

Esta sección incluye el diagrama de nivel de contenedor, detallando los principales componentes de software como la aplicación móvil, el backend API y la base de datos.

Componentes principales de la arquitectura de ParkingNow.

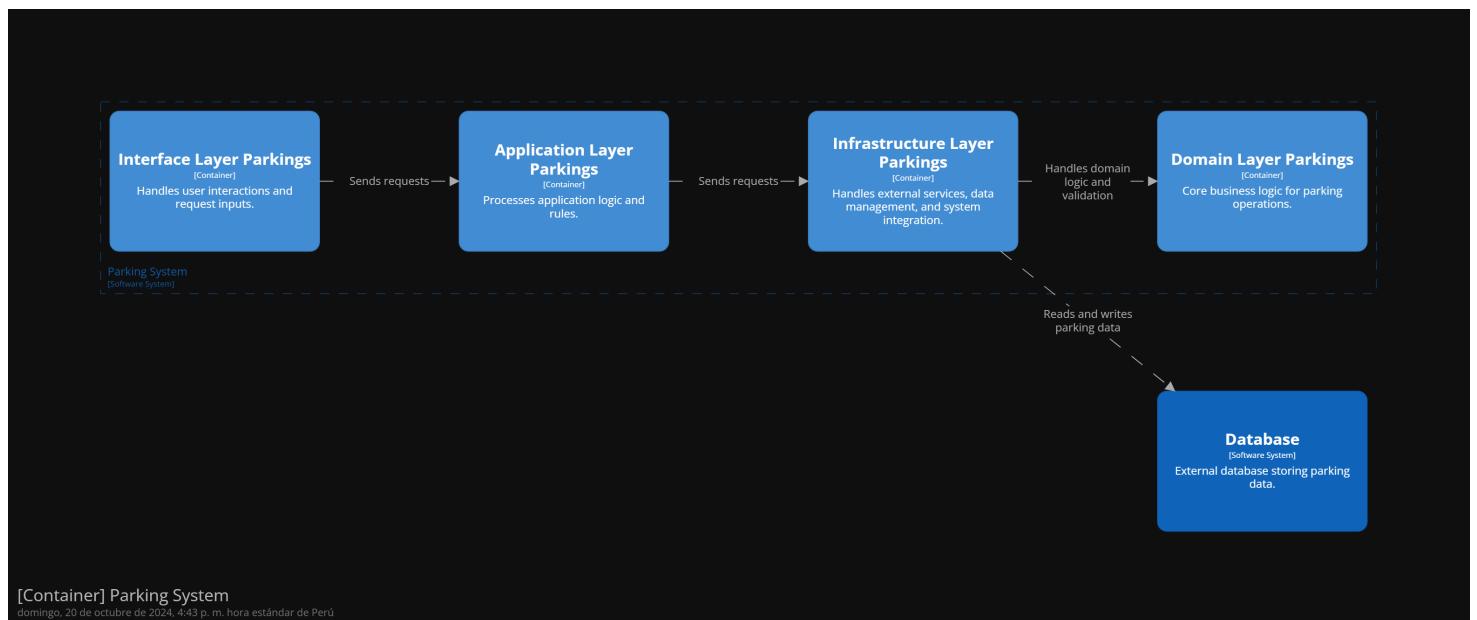




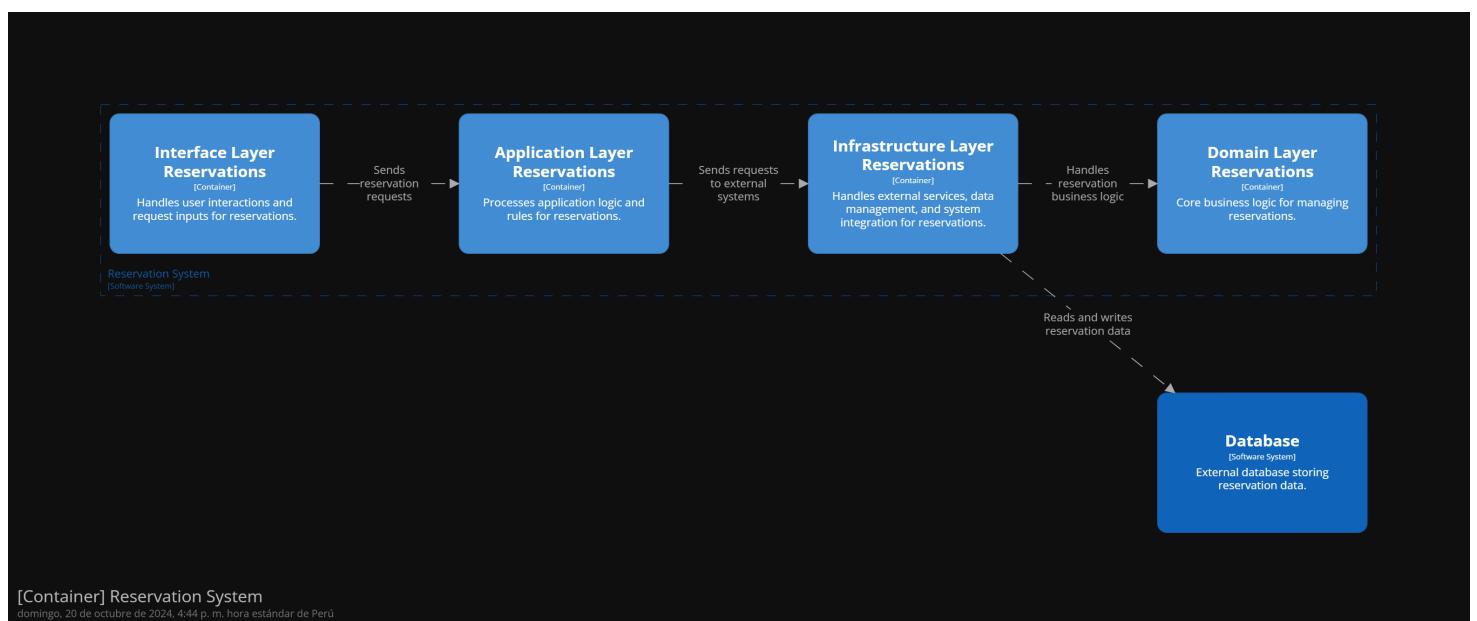
4.1.3.3. Software Architecture Deployment Diagrams

Se presenta el diagrama de despliegue, visualizando la infraestructura necesaria para soportar ParkingNow, incluyendo servidores, servicios cloud y dispositivos móviles.

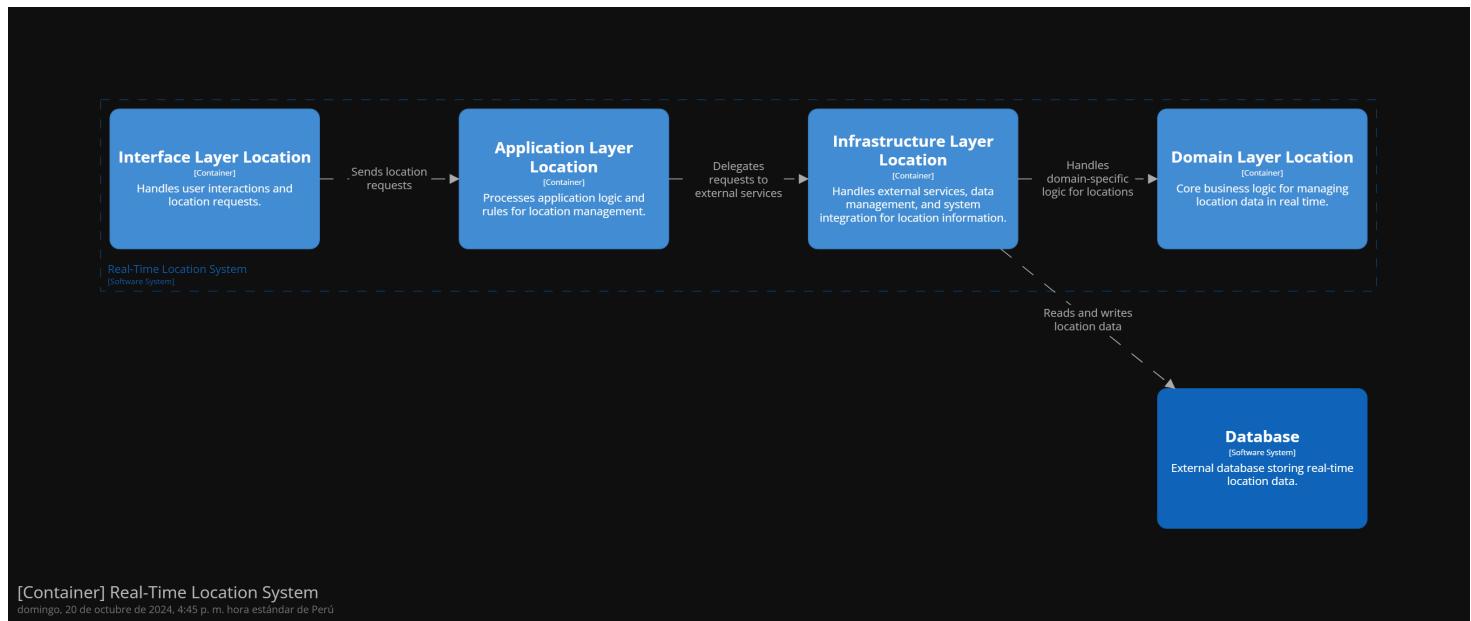
Despliegue del sistema de gestión de estacionamientos.



Despliegue del sistema de reservas.



Despliegue del sistema de ubicación en tiempo real.



4.2. Tactical-Level Domain-Driven Design

En esta sección se aborda el diseño táctico de ParkingNow, desglosando las capas internas de los bounded contexts principales según los principios de Domain-Driven Design.

4.2.1. Bounded Context: Parking Space Management Context

En esta sección, el equipo presenta las clases identificadas y las detalla a manera de diccionario, explicando para cada una su nombre, propósito y la documentación de atributos y métodos considerados, junto con las relaciones entre ellas.

4.2.1.1. Domain Layer

El núcleo de este Bounded Context (BC) gira en torno al Aggregate **Espacio**.

- **Aggregates:**
 - espacioId (identidad)
 - camaraId , localId (referencias)
 - descripcion , ubicacion (detalles)
 - estadoEspacio (estado actual)
- **Métodos:**
 - ocupar() , liberar() , marcarReservado() , marcarDisponible() , ponerEnMantenimiento() , actualizarEstadoPorCamara()
- **Value Objects:**
 - EspacioId , CamaraId , LocalId
 - EstadoEspacio : Disponible, Ocupado, Reservado, EnMantenimiento
 - UbicacionEspacio : Detalles de ubicación
- **Domain Services:**
 - VerificacionEstadoActualService contiene la lógica para consultar el estado consolidado de uno o más espacios, útil para capas superiores o read models.
- **Repositories:**
 - IEspacioRepository (interfaz) define cómo se guardan y recuperan los Aggregates **Espacio** .

4.2.1.2. Interface Layer

Esta capa permite la interacción externa con el Bounded Context.

- **Controller:**
 - EspacioController expone endpoints REST (ej. GET /espacios/{id} , PUT /espacios/{id}/ocupar)
- **Event Consumers:**
 - Escuchan eventos externos como actualizaciones desde **cámaras de monitoreo visual** o cancelaciones de reserva.
- **Event Producers:**
 - Publican eventos cuando el estado de un espacio cambia (ej. EspacioOcupadoEvent).

4.2.1.3. Application Layer

- Esta capa coordina las operaciones de negocio definidas.
- **Command Handlers:**
 - OcuparEspacioCommandHandler , MarcarReservadoCommandHandler , etc.
 - Ejecutan reglas del dominio y publican eventos.
- **Event Handlers:**

Reaccionan a eventos del dominio y de otros BCs, actualizando el estado de los espacios.
- **Query Services:**
 - ConsultaEspaciosQueryService
 - Proveen lecturas eficientes (DTOs), generalmente sin cargar el Aggregate completo.

4.2.1.4. Infrastructure Layer

- **Repositories:**

Implementaciones concretas como EspacioJpaRepository , que traduce entre entidades de dominio y la base de datos.
- **Messaging:**

Adaptadores a message brokers (ej. RabbitMQ, Kafka) para envío/recepción de eventos.
- **External Adapters:**
 - SensorIntegrationService : Comunicación con sensores.
 - Adaptadores para otros contextos como notificaciones.

4.2.1.5. Bounded Context Software Architecture Component Level Diagrams

Este diagrama representa cómo está compuesto internamente un container, en este caso el container del contexto **Parking Space Monitoring**.

Incluye:

- **Componentes principales:** Interface Layer, Application Layer, Domain Layer, Infrastructure Layer.
- **Interacciones internas** entre capas, donde cada una cumple una función especializada dentro del flujo de ejecución.
- **Responsabilidades de cada capa:**
 - **Interface Layer:** expone endpoints REST y consume/produce eventos externos.
 - **Application Layer:** coordina la ejecución de casos de uso mediante comandos, eventos y consultas.
 - **Domain Layer:** contiene las reglas de negocio, entidades, agregados, servicios de dominio y value objects.
 - **Infrastructure Layer:** conecta con sistemas externos como bases de datos, message brokers y servicios visuales de monitoreo por cámaras.

Diagrama de componentes del módulo de gestión de espacios



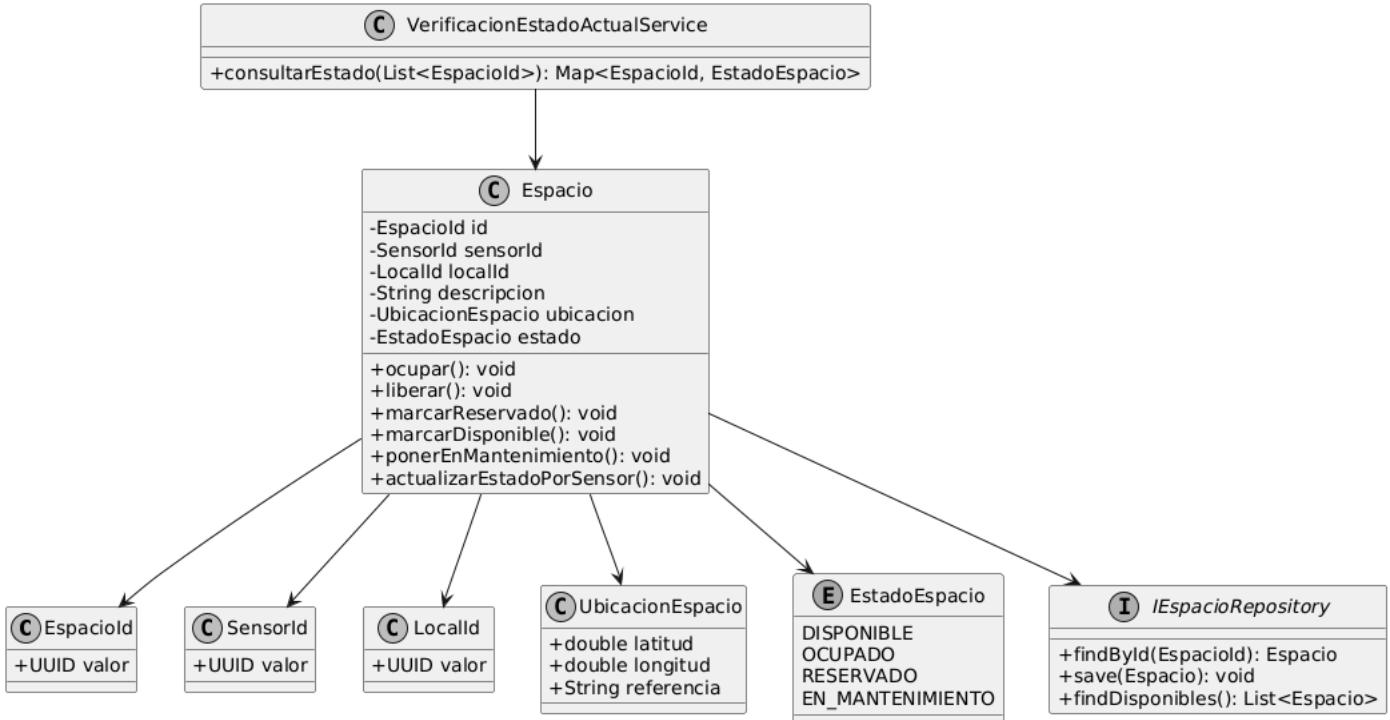
4.2.1.6. Bounded Context Software Architecture Code Level Diagrams

Esta sección se presenta las implementaciones de componentes y explica las secciones internas del Bounded Context **Parking Space Management**

4.2.1.6.1. Bounded Context Domain Layer Class Diagrams

Aquí se muestra el diseño de clases del Domain Layer, centrado en el agregado Espacio, sus Value Objects, servicios de dominio y repositorio.

Diagrama de clases del Domain Layer - Gestión de espacios



4.2.1.6.2. Bounded Context Database Design Diagram

Representación del modelo relacional para la persistencia del agregado Espacio.

Modelo de base de datos relacional del agregado Espacio

E	espacios
o	id : UUID [PK]
sensor_id :	UUID
local_id :	UUID
descripcion :	String
estado :	String
latitud :	double
longitud :	double
referencia :	String

4.2.2. Bounded Context: Reservation Management Context

En esta sección se describen las clases clave que conforman la lógica del contexto de Reservas, su propósito, atributos, métodos y relaciones.

4.2.2.1. Domain Layer

Esta capa contiene los elementos principales del dominio:

- **Aggregate Root:** `Reserva`, gestiona las reglas del negocio en torno al flujo de estados de la reserva y coordina entidades hijas.
- **Entities:** `RegistroTiempo`, se utiliza para registrar los hitos temporales relevantes.
- **Value Objects:** `ReservaId`, `PeriodoReserva`, `EstadoReserva`, `EspacioId`, `VehiculoId`, `UsuarioId`, `MultaExceso`.
- **Domain Services:** `ValidacionDisponibilidadPeriodoService` (verifica disponibilidad de espacios); `CalculoTiempoUsoService` y `DeterminacionMultasService` (manejan lógica temporal y penalidades).
- **Repository Interfaces:** `IReservaRepository`, contrato para persistencia y recuperación de `Reserva`.

4.2.2.2. Interface Layer

- **REST Controller:** `ReservaController`, expone endpoints para manejar reservas (crear, consultar, cancelar, check-in/out).
- **Event Listeners:** Responden a eventos como `EspacioDisponibleEvent`, `PagoCompletadoEvent`, etc.
- **Event Publishers:** Publican eventos del dominio como `ReservaConfirmadaEvent`, `CheckOutRegistradoEvent`.

4.2.2.3. Application Layer

- **Command Services:** `GestionReservaCommandService`, orquesta flujos a partir de comandos como `CrearReservaCommand`.

- **Query Services:** ConsultaReservasQueryService , consultas optimizadas sobre historial o estado actual.

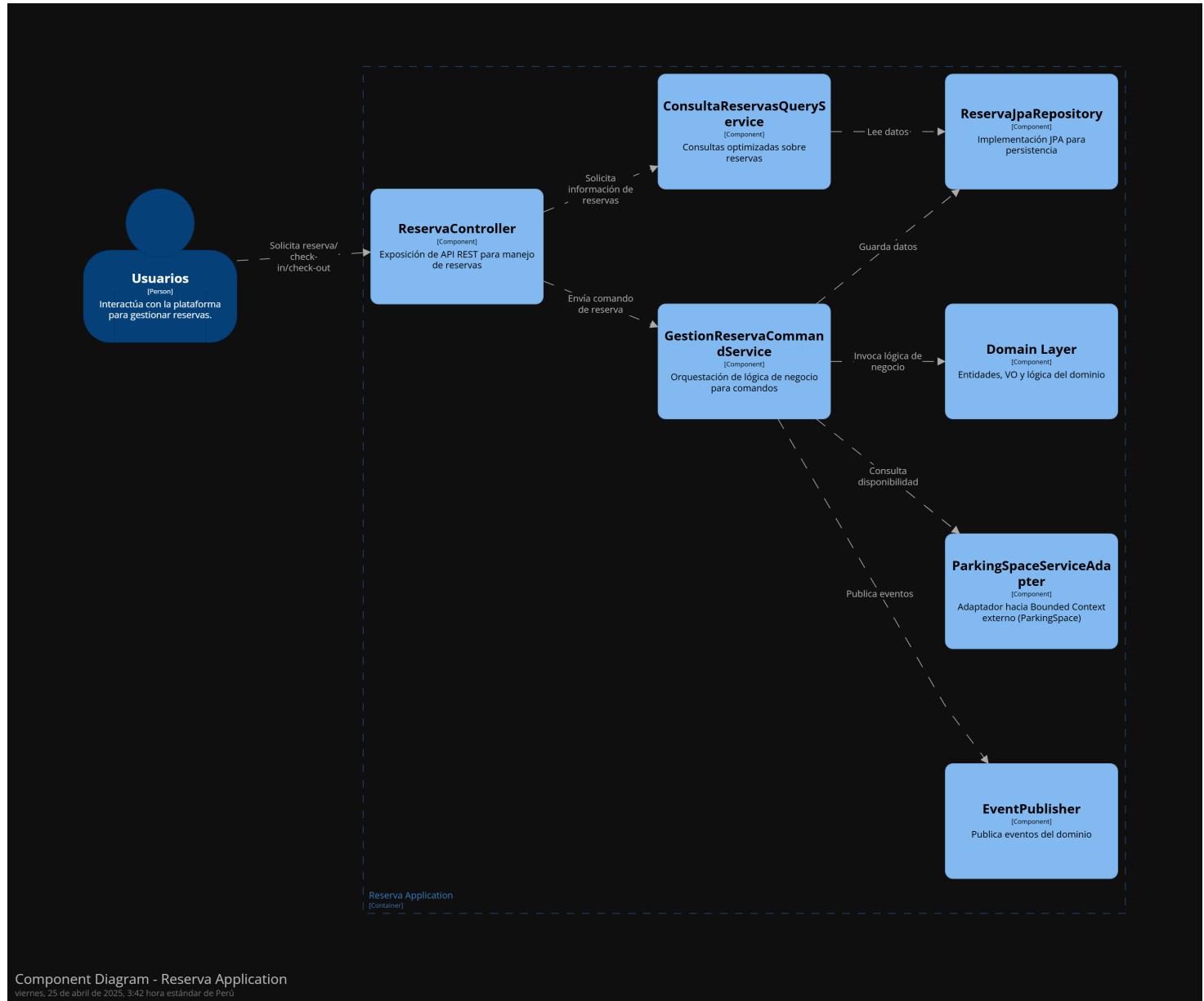
4.2.2.4. Infrastructure Layer

- **Repository Implementations:** ReservaJpaRepository , implementación concreta para IReservaRepository , mapeando tablas como Reservas y RegistroTiempo .
- **Messaging:** Brokers para emitir y consumir eventos.
- **Adapters (ACL):** Capas de adaptación hacia otros BCs como ParkingSpaceServiceAdapter , PaymentServiceAdapter , etc., garantizando acoplamiento bajo y comunicación limpia.

4.2.2.5. Bounded Context Software Architecture Component Level Diagrams

El presente diagrama representa la descomposición a nivel de componentes del Reservation Management Context, respetando los principios de Domain-Driven Design. Cada container ha sido dividido en componentes con responsabilidades claras y cohesionadas, permitiendo identificar cómo interactúan entre sí dentro de la arquitectura hexagonal. Este enfoque permite separar los flujos de comando, consulta y eventos, facilitando el mantenimiento, evolución e integración del sistema con otros bounded contexts como ParkingSpace y Payment.

Diagrama de componentes - Gestión de reservas



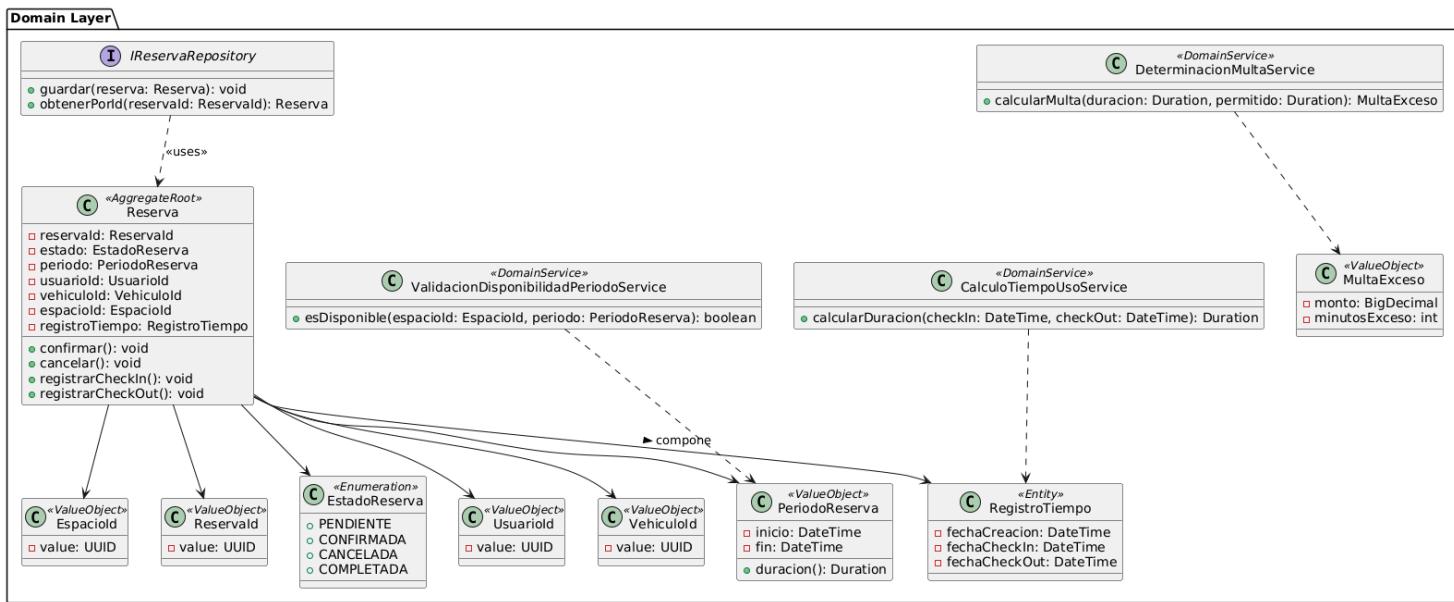
4.2.2.6. Bounded Context Software Architecture Code Level Diagrams

En esta sección se presentan los diagramas detallados a nivel de implementación del contexto de gestión de reservas. Se incluye tanto el diagrama de clases del Domain Layer como el diagrama de base de datos que soporta la persistencia del modelo.

4.2.2.6.1. Bounded Context Domain Layer Class Diagrams

A continuación, se describe el diagrama de clases UML para el Domain Layer del contexto de gestión de reservas:

Diagrama de clases - Domain Layer del contexto de reservas



4.2.2.6.2. Bounded Context Database Design Diagram

El siguiente diagrama representa el modelo relacional utilizado para la persistencia de información del contexto de reservas.

Modelo relacional - Base de datos del contexto de reservas



4.2.3. Bounded Context: User Management Context

En esta sección se describe la estructura del User Management Context del sistema ParkingNow, centrado en la gestión de los usuarios: Conductores y Dueños del Estacionamiento. Se detallan las clases principales, servicios, eventos y capas de arquitectura involucradas.

4.2.3.1. Domain Layer

Esta capa contiene los conceptos esenciales del dominio de usuarios:

- Aggregate Root:**
 - Usuario : Representa a un usuario del sistema, ya sea conductor o dueño del estacionamiento. Gestiona su identidad, credenciales, rol, perfil y relaciones con recursos como vehículos o estacionamientos.
- Entities:**

- **Perfil** : Contiene los datos personales del usuario como nombre, correo, teléfono y dirección.
- **Credenciales** : Agrupa la información de inicio de sesión (email y contraseña), incluyendo mecanismos de autenticación.
- **Vehículo** : Asociado exclusivamente a los conductores. Incluye información como marca, modelo, tipo y placa.
- **Estacionamiento** : Asociado exclusivamente a los dueños del estacionamiento. Contiene ubicación, capacidad y disponibilidad.
- **Rol** : Define el tipo de usuario (Conductor o Dueño del Estacionamiento).

- **Value Objects:**

- **UserId** : Identificador único del usuario.
- **Email , Password , NombreCompleto , Telefono , Direccion** : Objetos de valor con validaciones específicas.
- **RolUsuario** : Enum o clase de valor que representa si el usuario es **Conductor** o **Dueño del Estacionamiento**.

- **Domain Services:**

- **AutenticacionDomainService** : Encapsula la lógica de autenticación, incluyendo validación de credenciales y generación de tokens.
- **AutorizacionDomainService** : Determina si un usuario tiene permiso para realizar ciertas acciones según su rol.

- **Repository Interfaces:**

- **IUsuarioRepository** : Contrato que define las operaciones para la persistencia del aggregate **Usuario**.

4.2.3.2. Interface Layer

- **REST Controller:**

- **UsuarioController** : Expone los endpoints para registro, inicio de sesión, modificación de perfil, gestión de roles y asociación de vehículos o estacionamientos.

- **Event Listeners:**

- **EstacionamientoRegistradoEventListener** : Escucha eventos generados cuando se registra un nuevo estacionamiento, para asociarlo con el dueño correspondiente.
- **VehiculoRegistradoEventListener** : Reacciona a eventos cuando se registra un nuevo vehículo, para asociarlo con el conductor.

- **Event Publishers:**

- **UsuarioRegistradoEvent** : Emitido cuando un nuevo usuario se crea exitosamente.
- **VehiculoAsociadoEvent** : Indica que un vehículo ha sido vinculado a un conductor.
- **EstacionamientoAsociadoEvent** : Indica que un estacionamiento ha sido vinculado a un dueño del estacionamiento.

4.2.3.3. Application Layer

- **Command Services:**

- **GestionUsuarioCommandService** : Orquesta casos de uso de modificación de estado como:
 - **RegistrarUsuarioCommand**
 - **AsociarVehiculoCommand**
 - **AsociarEstacionamientoCommand**
 - **CambiarRolCommand**

- **Query Services:**

- **ConsultaUsuarioQueryService** : Provee operaciones de lectura optimizada para obtener información del perfil, vehículos o estacionamientos asociados.

- **Event Handlers:**

- **UsuarioEventHandler** : Gestiona eventos como **UsuarioRegistradoEvent** para la coordinación con otros contextos y persistencia de datos.

4.2.3.4. Infrastructure Layer

- **Repository Implementations:**

- **UsuarioJpaRepository** : Implementación de **IUsuarioRepository** usando tecnologías ORM como JPA o Entity Framework. Mapea entidades como **Usuario , Perfil , Credenciales , Vehiculo , Estacionamiento , y Rol**.

- **Messaging:**

- Uso de sistemas de mensajería como Kafka o RabbitMQ para la emisión y recepción de eventos entre bounded contexts.
 - Ejemplos: Publicar **UsuarioBloqueadoEvent** o escuchar **ReservaConfirmadaEvent**.

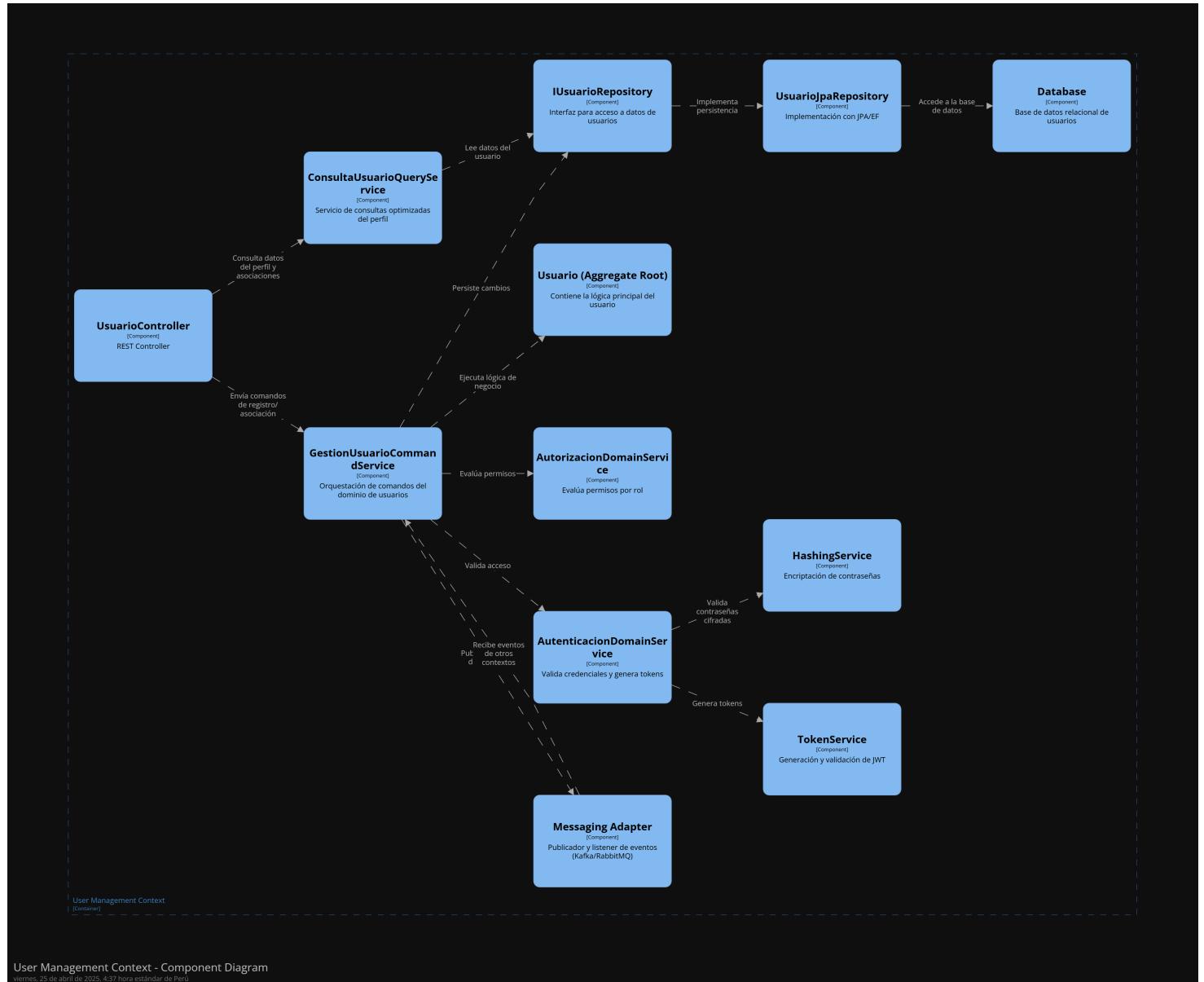
- **Adapters (ACL):**

- **TokenService** : Responsable de generar y validar tokens JWT.
- **HashingService** : Aplica algoritmos seguros para cifrado de contraseñas.
- **ParkingServiceAdapter** : Adaptador que comunica con el contexto de estacionamientos.
- **VehicleServiceAdapter** : Adaptador que comunica con el contexto de vehículos.

4.2.3.5. Bounded Context Software Architecture Component Level Diagrams

En esta sección se presentan los Component Diagrams del Modelo C4 para el Container User Management API, el cual forma parte del Bounded Context User Management Context. Este Container es responsable de manejar las operaciones relacionadas con la autenticación, registro, gestión de roles, perfiles de usuario y asociación de vehículos o estacionamientos.

Component Diagram – User Management Context



User Management Context - Component Diagram

viernes, 25 de abril de 2025, 4:37 hora estándar de Perú

4.2.3.6. Bounded Context Software Architecture Code Level Diagrams

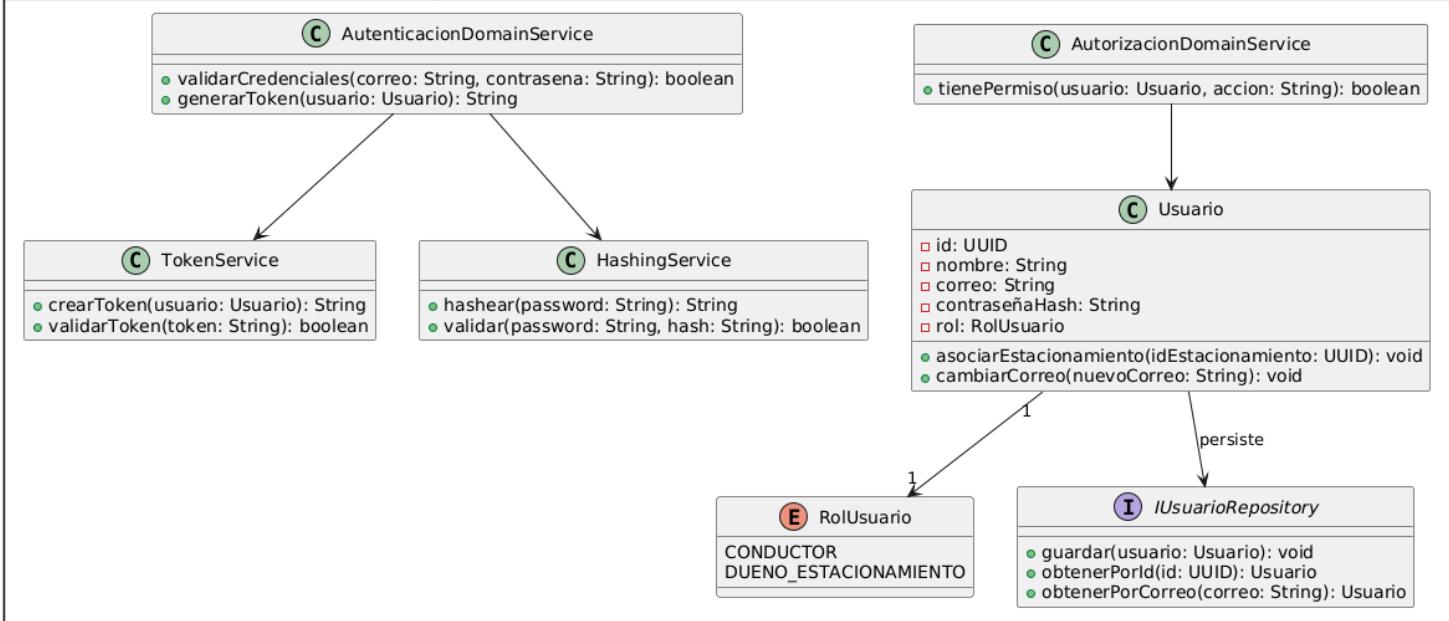
Esta sección detalla la implementación de los componentes principales del User Management Context, incluyendo el Domain Layer Class Diagram (UML) y el Database Diagram con sus entidades y relaciones de persistencia.

4.2.3.6.1. Bounded Context Domain Layer Class Diagrams

Este diagrama representa la estructura del Domain Layer, mostrando las entidades, objetos de valor, agregados, servicios de dominio, interfaces de repositorio, y otras clases relevantes con sus atributos, métodos, visibilidades y relaciones.

Domain Layer – User Management Context

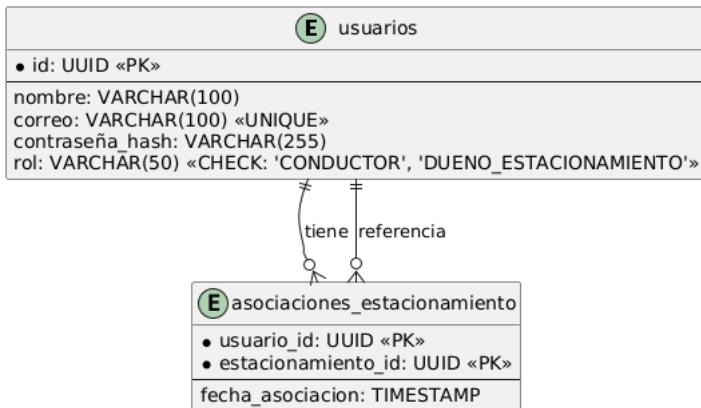
Domain Layer - User Management Context



4.2.3.6.2. Bounded Context Database Design Diagram

Este diagrama representa la estructura de la base de datos relacional para este contexto. Se definen las tablas, columnas, tipos de datos, claves primarias/foráneas y relaciones.

Database Design – User Management Context



4.2.4. Bounded Context: Local Context

En esta sección se describen las clases clave que conforman la lógica del contexto de Locales, su propósito, atributos, métodos y relaciones.

4.2.4.1. Domain Layer

- Aggregate Root:** Local , gestiona la información y reglas de negocio del estacionamiento y sus asociados (tarifas, promociones, opiniones).
- Entities:** Tarifa , Promocion , Opinion , asociadas al Local .
- Value Objects:** LocalId , UbicacionFisica , objetos de valor para encapsular detalles de tarifas, promociones y opiniones.
- Domain Services:** CalculoTarifaAplicableService , CalculoCalificacionPromedioService .
- Repository Interfaces:** ILocalRepository , contrato para persistencia y recuperación de Local .

4.2.4.2. Interface Layer

- REST Controller:** LocalController , gestiona endpoints para crear y consultar locales, tarifas, promociones y opiniones.
- Event Listeners:** Escuchan eventos como UsuarioRegistradoEvent .
- Event Publishers:** Emite eventos del dominio como LocalRegistradoEvent , PromocionActivaEvent .

4.2.4.3. Application Layer

- **Command Services:** `GestionLocalCommandService`, orquesta flujos de gestión de locales.
- **Query Services:** `ConsultaLocalQueryService`, realiza consultas de locales y sus detalles.

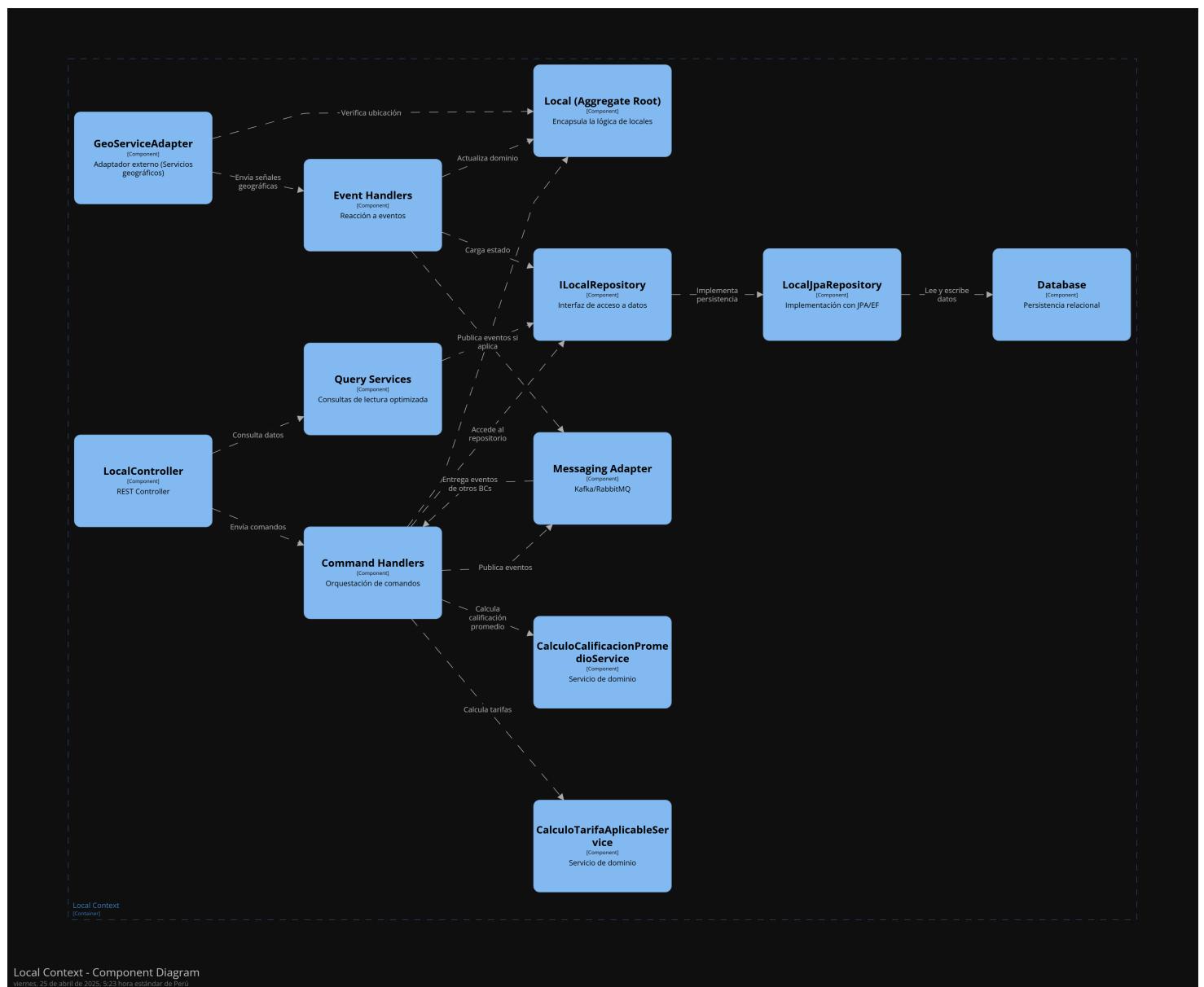
4.2.4.4. Infrastructure Layer

- **Repository Implementations:** `LocalJpaRepository`, implementa `ILocalRepository`, mapea las tablas relacionadas con locales, tarifas, promociones y opiniones.
- **Messaging:** Brokers para emisión y consumo de eventos.
- **Adapters (ACL):** Servicios como `GeoServiceAdapter`, `UserServiceAdapter` para interacción técnica con otros BCs o sistemas externos.

4.2.4.5. Bounded Context Software Architecture Component Level Diagrams

En esta sección se presenta el Component Diagram para el Bounded Context: Local Context, utilizando el C4 Model. Este diagrama muestra la descomposición del Container de gestión de locales en componentes clave, sus responsabilidades y las interacciones entre ellos. El diagrama refleja cómo los diferentes componentes trabajan juntos dentro del contexto de locales, incluyendo el manejo de tarifas, promociones, opiniones y más.

Component Diagram – Local Context



4.2.4.6. Bounded Context Software Architecture Code Level Diagrams

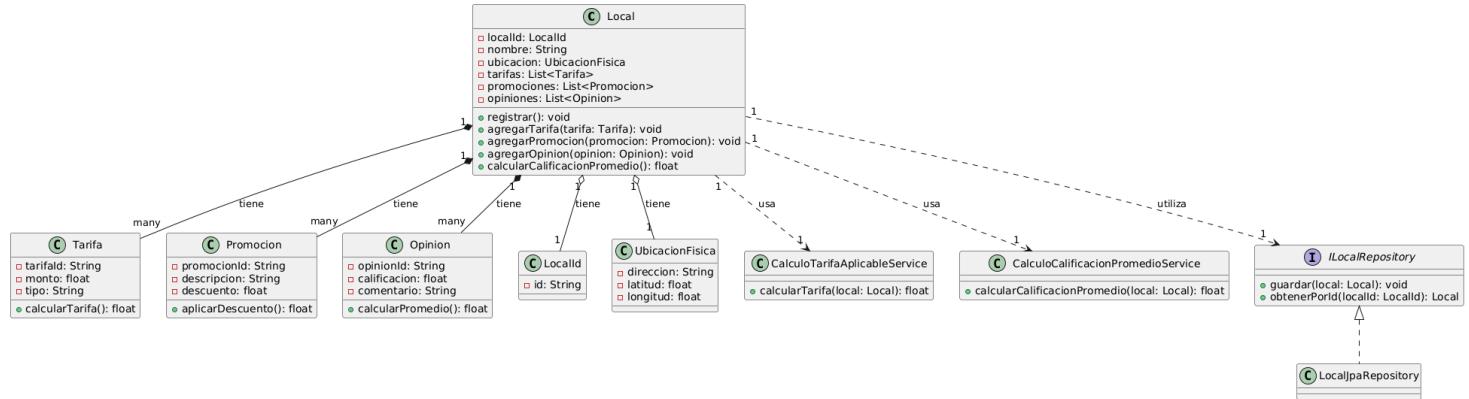
En esta sección, se presentan y explican los diagramas que muestran un mayor nivel de detalle sobre la implementación de los componentes dentro del Bounded Context: Local Context. Estos diagramas proporcionan una visión más técnica y detallada de cómo se implementan los componentes en el

código, con énfasis en las clases del Domain Layer y la estructura de la base de datos.

4.2.4.6.1. Bounded Context Domain Layer Class Diagrams

En esta sección se presenta el Class Diagram en UML para las clases del Domain Layer dentro del Bounded Context: Local Context. El diagrama incluirá las clases clave, interfaces, enumeraciones, atributos, métodos y sus relaciones, detallando también el scope de cada miembro (public, private, protected) y la multiplicidad de las relaciones. Este diagrama describe con precisión la estructura de las entidades y objetos de valor que componen el modelo de dominio, así como los servicios de dominio involucrados.

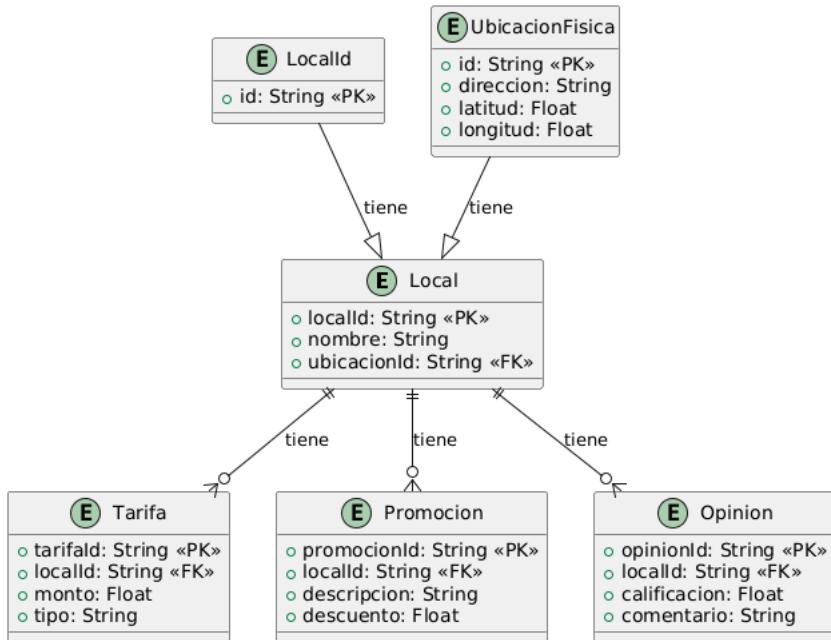
Class Diagram – Local Context Domain Layer



4.2.4.6.2. Bounded Context Database Design Diagram

En esta sección, se presenta el Database Diagram que detalla los objetos de base de datos para la persistencia de la información en el contexto de locales. El diagrama incluirá las tablas y columnas relevantes, así como las constraints (como claves primarias y foráneas), y mostrará las relaciones entre las tablas que permiten almacenar los datos de los Local, Tarifa, Promocion, Opinion, entre otros.

Database Design Diagram – Local Context



4.2.5. Bounded Context: Security Context

En esta sección se describen las clases clave que conforman la lógica del contexto de seguridad, su propósito, atributos, métodos y relaciones.

4.2.5.1. Domain Layer

- **Aggregate Root:** `Seguridad`, gestiona la información y reglas de negocio relacionadas con los tipos y niveles de seguridad.
- **Entities:** `RegistroSeguridad`, registra eventos de seguridad relevantes.
- **Value Objects:** `SecurityType`, `SecurityLevel`, `DatosAuditoria`, encapsulan detalles de seguridad, niveles y auditoría.

- **Domain Services:** AuthorizationService , verifica permisos de acceso; AuditingService , registra eventos de seguridad.
 - **Repository Interfaces:** ISecurityPolicyRepository , ISecurityLogRepository , contratos para persistencia y recuperación de políticas de seguridad y registros de auditoría.

4.2.5.2. Interface Layer

- **REST Controller:** APIs para configurar políticas de seguridad o consultar registros de seguridad y auditoría.
 - **Event Listeners:** Escuchan eventos como `AccessDeniedEvent` o `SecurityEventLoggedEvent` para aplicar auditoría.
 - **Event Publishers:** Emite eventos del dominio como `SecurityEventLoggedEvent` o `AccessDeniedEvent`.

4.2.5.3. Application Layer

- **Command Services:** `SecurityPolicyCommandService`, procesa comandos de configuración de políticas de seguridad.
 - **Query Services:** `SecurityLogQueryService`, realiza consultas sobre registros de seguridad y auditoría.

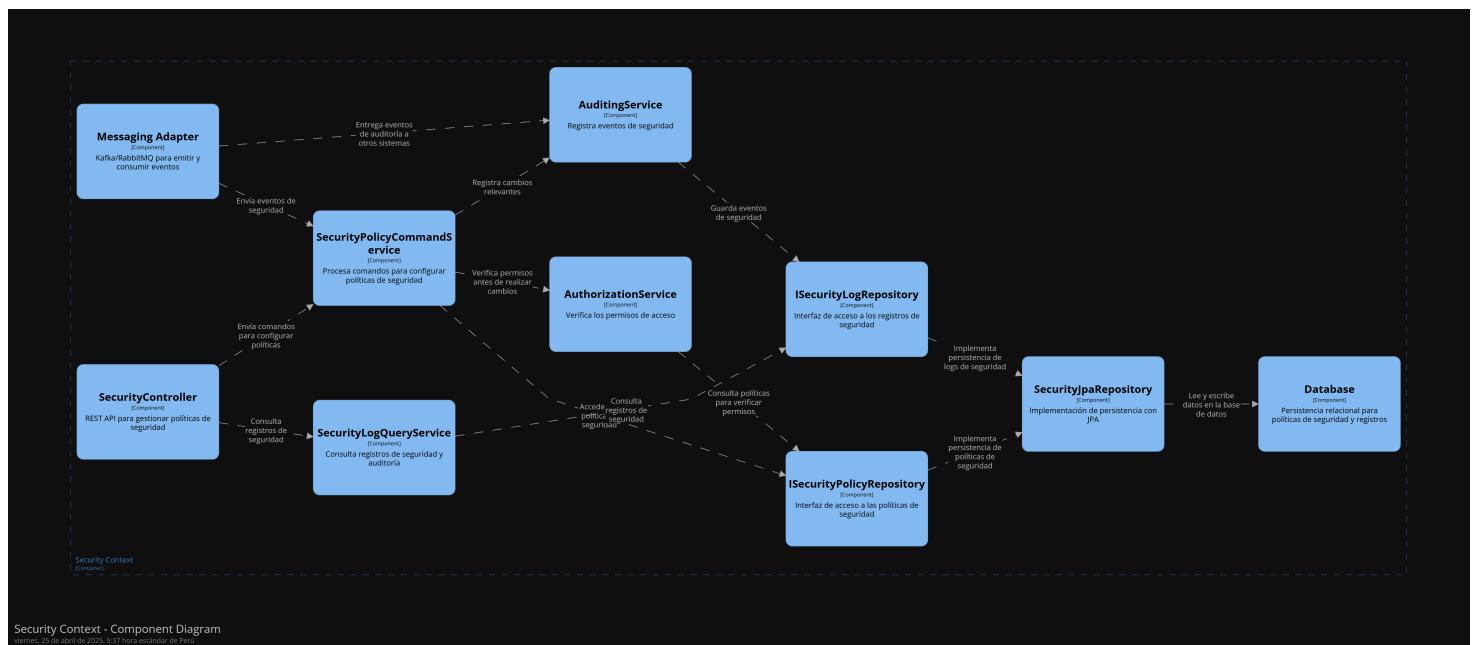
4.2.5.4. Infrastructure Layer

- **Repository Implementations:** Implementaciones JPA para Seguridad, registros de seguridad y políticas de acceso.
 - **Messaging:** Implementación de brokers para emisión y consumo de eventos relacionados con seguridad y auditoría.
 - **Adapters (ACL):** Servicios de integración con sistemas externos, como servicios de logging y monitoreo.

4.2.5.5. Bounded Context Software Architecture Component Level Diagrams

Este diagrama muestra los principales componentes del contexto de seguridad en PARKINGNOW, encargados de la autorización, auditoría y validación de políticas. Destacan servicios como `AuthorizationService`, `AuditLogService` y repositorios de políticas, integrados mediante adaptadores de mensajería y eventos.

Componentes del contexto de seguridad en ParkingNow



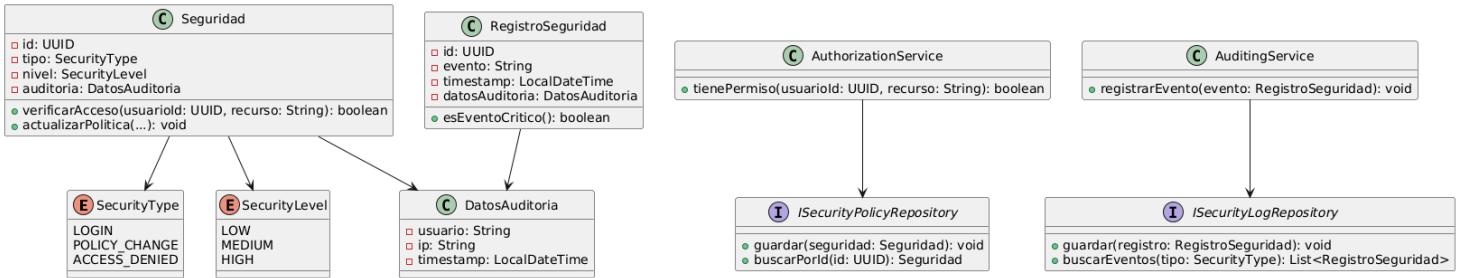
4.2.5.6. Bounded Context Software Architecture Code Level Diagrams

En esta sección, se presentan y explican los diagramas que muestran un mayor nivel de detalle sobre la implementación de los componentes dentro del Bounded Context: Security Context. Estos diagramas proporcionan una visión más técnica y detallada de cómo se implementan los componentes en el código, con énfasis en las clases del Domain Layer y la estructura de la base de datos.

4.2.5.6.1. Bounded Context Domain Layer Class Diagrams

Este diagrama representa las principales clases del Domain Layer del Security Context. Incluye entidades, objetos de valor, servicios de dominio e interfaces de repositorio, y muestra sus relaciones, métodos y atributos clave. Este nivel de modelado facilita entender cómo se organiza la lógica del negocio centrada en la gestión de políticas de seguridad, eventos de auditoría y control de acceso.

Estructura de clases del contexto de seguridad

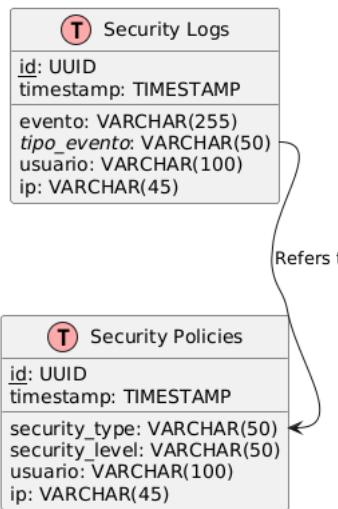


4.2.5.6.2. Bounded Context Database Design Diagram

El siguiente diagrama describe el esquema de base de datos relacional para el Security Context. Se enfoca en las tablas que almacenan políticas de seguridad y registros de eventos de auditoría, con detalles como claves primarias, tipos de datos, y relaciones entre las entidades. Este modelo garantiza una estructura persistente coherente con las necesidades del dominio.

Modelo relacional del contexto de seguridad

Security Context - Database Diagram



4.2.6. Bounded Context: Support Context

En esta sección se describen las clases clave relacionadas con el servicio de soporte al cliente y asesorías.

4.2.6.1. Domain Layer

- **Aggregate Root:** Asesoria , gestiona el ciclo de vida de una solicitud de soporte y sus reglas de negocio.
- **Entities:** Asesor , si se gestiona aquí, representa un agente de soporte.
- **Value Objects:** AsesoriaId , RazonAsesoria , PeriodoAsesoria , EstadoAsesoria , AsesorId , UsuarioId , encapsulan información relevante para las asesorías.
- **Domain Services:** AsignacionAsesorService , contiene la lógica para asignar asesores.
- **Repository Interfaces:** IAsesoriaRepository , IAsesorRepository , contratos para persistencia de Asesoria y Asesor .

4.2.6.2. Interface Layer

- **REST Controller:** AsesoriaController , gestiona APIs para solicitud, consulta y gestión de asesorías.
- **Event Listeners:** Escuchan eventos que pueden generar nuevas solicitudes de soporte.
- **Event Publishers:** Emite eventos de soporte como AsesoriaSolicitadaEvent .

4.2.6.3. Application Layer

- **Command Services:** GestionAsesoriaCommandService , procesa comandos relacionados con la gestión de asesorías.
- **Query Services:** ConsultaAsesoriaQueryService , maneja consultas sobre el estado de los casos de soporte.

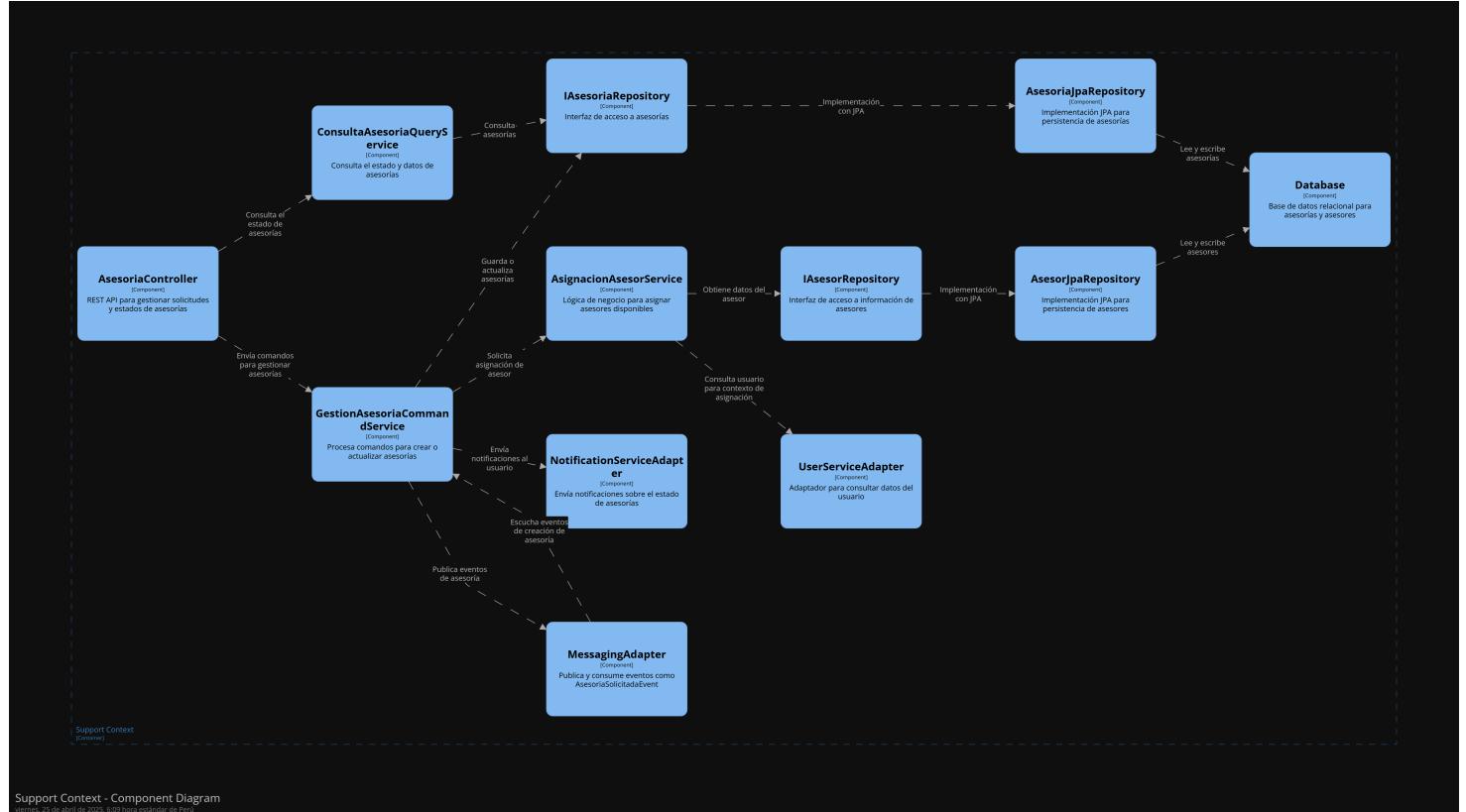
4.2.6.4. Infrastructure Layer

- Repository Implementations:** `AsesoriaJpaRepository`, `AsesorJpaRepository`, implementan las interfaces de repositorio para persistir asesorías y asesores.
- Messaging:** Brokers para emisión y consumo de eventos de soporte.
- Adapters (ACL):** Servicios de integración con otros BCs como `UserServiceAdapter` o `NotificationServiceAdapter`.

4.2.6.5. Bounded Context Software Architecture Component Level Diagrams

El diagrama de componentes del Support Context muestra cómo está estructurado internamente el contenedor encargado de gestionar las asesorías y solicitudes de soporte. Se identifican componentes de entrada (como controladores REST), servicios de aplicación para gestión y consulta de asesorías, lógica de dominio como la asignación de asesores, e integración con infraestructura (repositorios, adaptadores y eventos). Este desglose facilita la comprensión de responsabilidades, dependencias y puntos de integración con otros bounded contexts o sistemas externos.

Componentes del contexto de soporte y asesorías



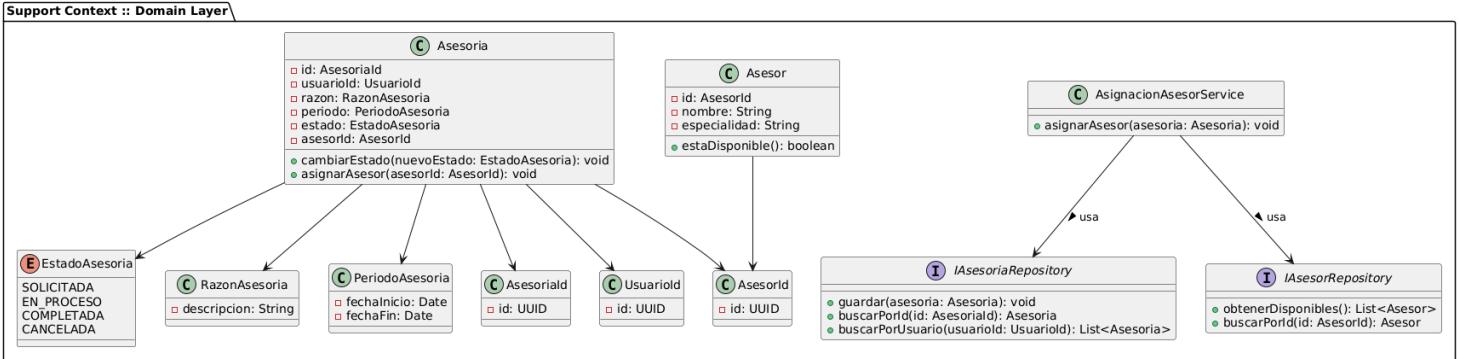
4.2.6.6. Bounded Context Software Architecture Code Level Diagrams

Esta sección presenta una descripción detallada de la implementación interna del Support Context a nivel de código. Se estructura en dos partes: el Domain Layer Class Diagram, que representa las clases, interfaces, objetos de valor y sus relaciones dentro del dominio, y el Database Diagram, que refleja cómo se modela la persistencia de los datos en una base de datos relacional. Estos diagramas permiten comprender cómo se implementan los conceptos del dominio y cómo se mantienen consistentes con la arquitectura general del sistema.

4.2.6.6.1. Bounded Context Domain Layer Class Diagrams

El siguiente diagrama de clases representa la estructura del Domain Layer del Support Context. Aquí se modelan las clases, interfaces y objetos de valor que encapsulan la lógica de negocio, así como sus relaciones, atributos y métodos. Esta capa es fundamental para mantener la lógica independiente de frameworks o tecnologías externas.

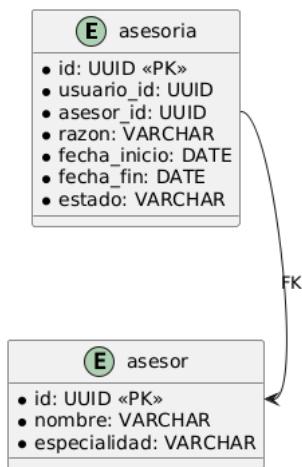
Diagrama de clases del dominio de asesorías y asignación



4.2.6.6.2. Bounded Context Database Design Diagram

Este diagrama muestra la representación de las entidades persistidas en base de datos para el Support Context. Define las tablas, columnas, claves primarias, claves foráneas y relaciones, modelando la estructura relacional necesaria para almacenar asesorías y datos de asesores.

Modelo relacional de asesorías y asesores



4.2.7. Bounded Context: Notification Context

En esta sección se describen las clases clave relacionadas con la gestión de notificaciones y alertas en tiempo real.

4.2.7.1. Domain Layer

- **Aggregates:** Notificacion (Aggregate Root) y Alerta (Aggregate Root), gestionan las reglas y estados de las notificaciones y alertas.
- **Entities:** DestinatarioNotificacion, una entidad hija dentro del Notificacion Aggregate.
- **Value Objects:** NotificacionId, ContenidoNotificacion, FechaHoraNotificacion, PrioridadNotificacion, AlertaId, EstadoAlerta, encapsulan detalles de la notificación y la alerta.
- **Domain Services:** DistribucionNotificacionesService, gestiona la lógica de envío de notificaciones; GestorAlertasService, gestiona la lógica de alertas.
- **Repository Interfaces:** INotificacionRepository, IAlertaRepository, contratos para la persistencia de Notificacion y Alerta .

4.2.7.2. Interface Layer

- **REST Controller:** NotificacionController, gestiona APIs para la solicitud y consulta de notificaciones y alertas.
- **Event Listeners (Consumers):** Suscriben a eventos de otros BCs para generar notificaciones o alertas.
- **Event Publishers:** Emite eventos de dominio como NotificacionEnviadaEvent, AlertaActivadaEvent .

4.2.7.3. Application Layer

- **Command Services:** GestionNotificacionCommandService, GestionAlertaCommandService , procesan comandos relacionados con la gestión de notificaciones y alertas.
- **Query Services:** ConsultaNotificacionQueryService, ConsultaAlertaQueryService , manejan consultas sobre notificaciones y alertas.

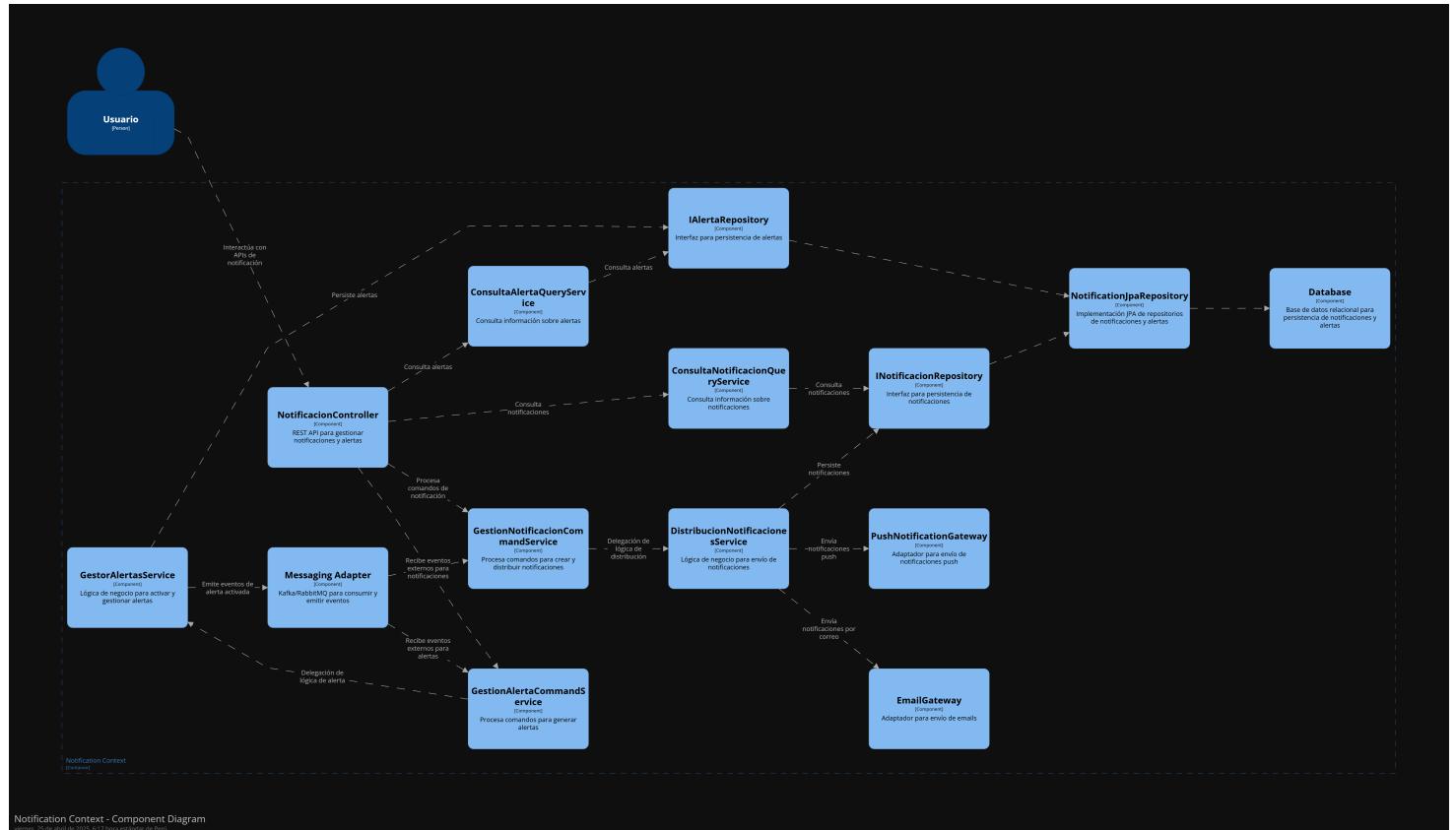
4.2.7.4. Infrastructure Layer

- **Repository Implementations:** Implementaciones JPA para Notificacion, Alertas, y sus relaciones con otros elementos.
- **Messaging:** Implementación de mensajería para escuchar múltiples eventos entrantes y emitir notificaciones.
- **Adapters (ACL):** Integración con servicios externos como EmailGateway, PushNotificationGateway, y UserServiceAdapter para los destinatarios.

4.2.7.5. Bounded Context Software Architecture Component Level Diagrams

El siguiente diagrama muestra la descomposición del container Notification Context, que es responsable de gestionar el envío de notificaciones y alertas en el sistema. Este contexto incluye componentes de control, servicios de aplicación, servicios de dominio y adaptadores de infraestructura. El diagrama refleja cómo interactúan estos componentes para procesar solicitudes, generar eventos y distribuir notificaciones en tiempo real a través de diferentes canales como email o notificaciones push.

Diagrama de componentes del sistema de notificaciones



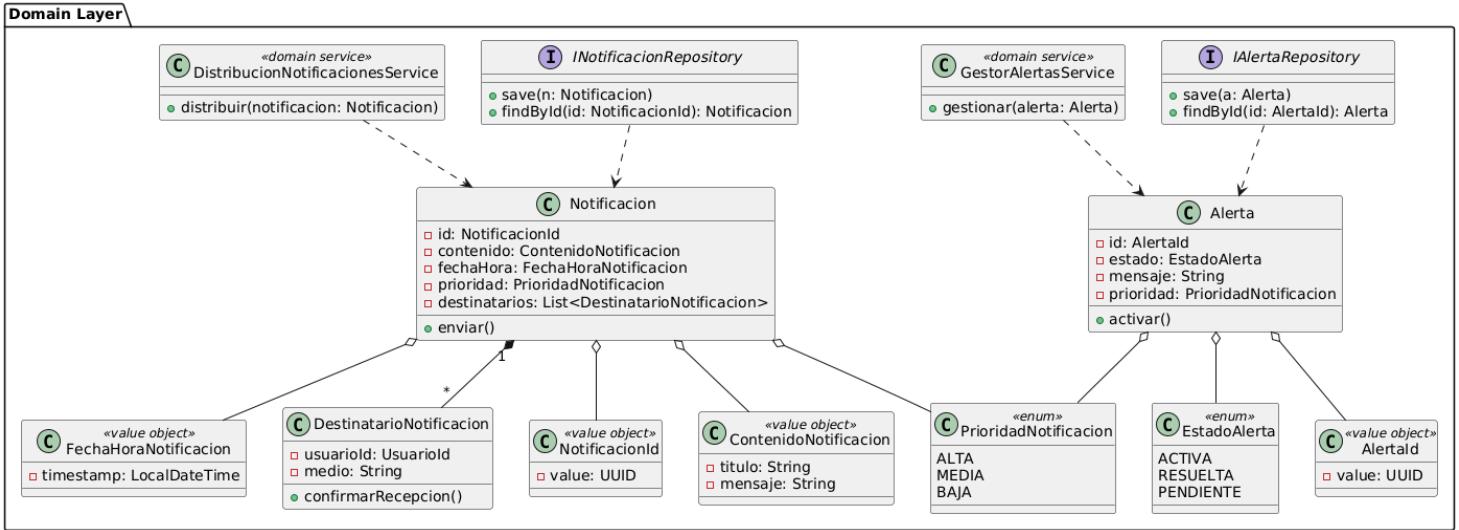
4.2.7.6. Bounded Context Software Architecture Code Level Diagrams

El siguiente apartado presenta los diagramas de nivel de código del Notification Context, los cuales permiten visualizar con mayor detalle la estructura interna del dominio. Se muestran las clases, interfaces, value objects y servicios utilizados para organizar la lógica del negocio centrada en la gestión de notificaciones y alertas. Esta representación facilita el entendimiento técnico del diseño del sistema, promoviendo su mantenibilidad y escalabilidad.

4.2.7.6.1. Bounded Context Domain Layer Class Diagrams

Este diagrama representa las clases, interfaces, value objects y servicios del Domain Layer del Notification Context. El objetivo es mostrar cómo se organiza el modelo de dominio alrededor de los aggregate roots Notificacion y Alerta, los servicios de dominio asociados, y cómo se estructuran las entidades e interfaces de repositorio. Se incluyen atributos, métodos, relaciones, visibilidad, y multiplicidades conforme al estándar de UML.

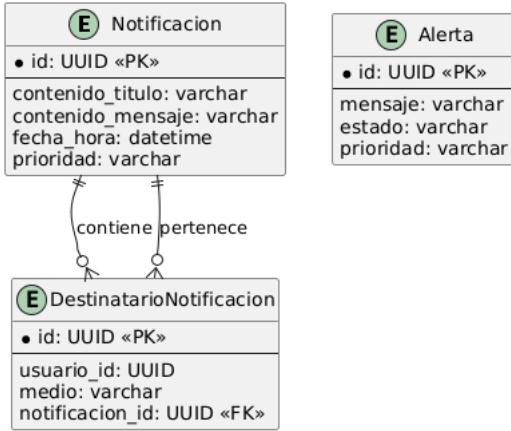
Modelo de clases del dominio de notificaciones y alertas



4.2.7.6.2. Bounded Context Database Design Diagram

El siguiente diagrama de base de datos representa las tablas que sustentan la persistencia del Notification Context. Cada entidad del modelo de dominio tiene una tabla asociada, con claves primarias y foráneas que mantienen la integridad referencial entre notificaciones, alertas y destinatarios. Se describe la estructura de columnas y constraints principales de la base de datos relacional utilizada.

Diagrama de base de datos relacional del Notification Context



Capítulo V: Solution UI/UX Desing

5.1 Product Design

En esta sección se describe el diseño general del producto digital, considerando tanto la aplicación móvil como el landing page. Se definen los lineamientos visuales, de interacción y estructura de contenido, enfocados en ofrecer una experiencia clara, accesible y coherente para el usuario. Los subapartados detallan los componentes clave del diseño UI/UX propuesto.

5.1.1 Style Guidelines

Esta sección define las guías visuales que aseguran coherencia y uniformidad en toda la solución de ParkingNow. Establece los elementos gráficos esenciales que se aplicarán tanto en la aplicación móvil como en el landing page, permitiendo una experiencia visual clara, accesible y alineada con la identidad del producto.

5.1.1.1. General Style Guidelines

Brand Overview

En diversas ciudades, encontrar un espacio de estacionamiento disponible puede ser una tarea difícil, principalmente por la falta de información en tiempo real y la ausencia de herramientas para realizar reservas anticipadas. Esta situación genera estacionamientos indebidos, mayor congestión vehicular y

una experiencia negativa para los conductores.

ParkingNow surge como una solución a estos problemas, ofreciendo una plataforma que conecta a los conductores con espacios de estacionamiento disponibles de manera inmediata. A través del uso de tecnología moderna, como cámaras de monitoreo y geolocalización, la plataforma permite no solo ubicar y reservar espacios fácilmente, sino también mejorar el flujo vehicular urbano, beneficiando tanto a conductores como a propietarios de estacionamiento.

Brand Name

El nombre de nuestra solución, **ParkingNow**, representa con claridad su propósito: brindar una herramienta ágil y efectiva para ubicar y gestionar espacios de estacionamiento. El término "Parking" hace referencia directa al servicio principal, mientras que "Now" destaca la rapidez e inmediatez de la plataforma. Se eligió un nombre en inglés por su carácter universal y su facilidad para conectar con un público amplio en un entorno digital globalizado.

A continuación, se presenta el **logo**:



Typography

La tipografía en *ParkingNow* cumple un rol esencial en la claridad visual y la usabilidad de la aplicación móvil. Se eligieron fuentes modernas, limpias y altamente legibles que permiten a los usuarios identificar rápidamente la información clave, mejorando así la experiencia general de navegación.

La estructura tipográfica se organiza en cuatro niveles jerárquicos: **headings** (títulos), **body** (texto principal), **buttons** (botones) y **links** (enlaces), garantizando uniformidad visual en toda la interfaz.

Heading 1 - Inter - 24 px - Títulos principales

Line Height: AUTO / Letter Spacing: 0 %

Heading 2 - Inter - 20 px - Subtítulos o secciones

Line Height: AUTO / Letter Spacing: 0 %

Body text - roboto - 16 px - Texto general del contenido

Line Height: AUTO / Letter Spacing: 0 %

Link text - roboto - 14 px -Enlace (Color azul, subrayado)

Line Height: AUTO / Letter Spacing: 0 %

Button Text - roboto - 14 px - Texto de botones (negrita o mayúsculas)

Line Height: AUTO / Letter Spacing: 0 %

Colors

La selección de colores desempeña un rol clave en el diseño visual de *ParkingNow*, ya que influye directamente en la percepción del usuario y en la navegación dentro de la aplicación móvil. Por ello, se eligieron cuidadosamente tonos que reflejan la identidad de la marca y mejoran la experiencia del usuario.

El color principal definido es el **azul noche**, el cual transmite tecnología y seguridad, atributos esenciales para una plataforma confiable de estacionamiento. Este tono proporciona una sensación de tranquilidad y confianza durante el uso.

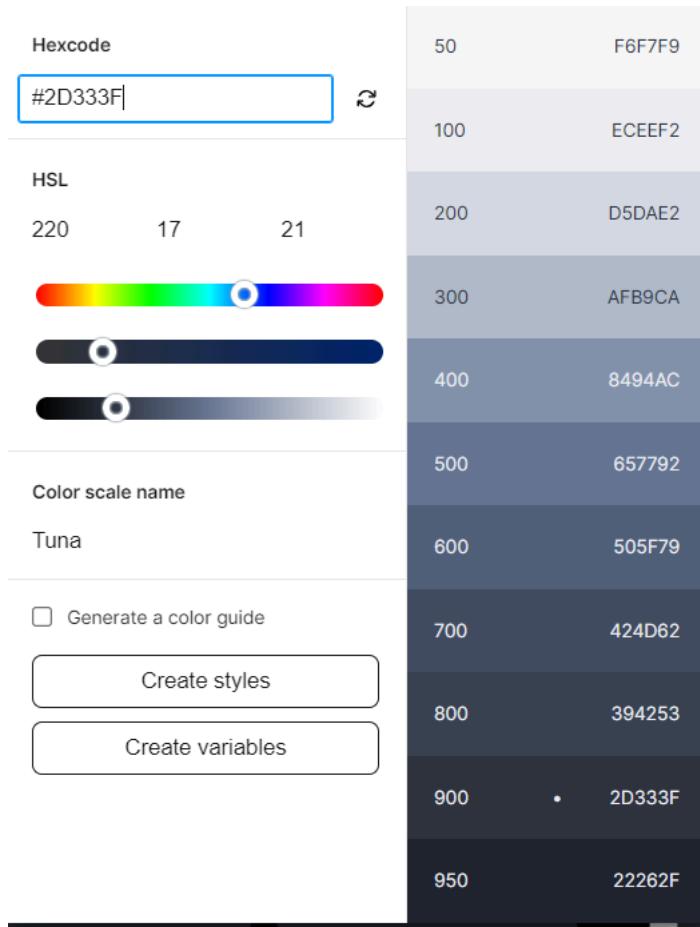
Además, se incorporaron colores secundarios como el **amarillo**, el **gris**, el **blanco** y el **negro**, los cuales aportan energía, equilibrio visual, limpieza y contraste respectivamente. Estos colores complementan la interfaz y permiten destacar elementos interactivos, zonas informativas y botones clave en la app.

A continuación, se presenta la paleta de colores utilizada en *ParkingNow* junto con sus respectivos códigos hexadecimales.

Color Primario – Azul Noche (#2D333F)

El color principal seleccionado para la aplicación móvil *ParkingNow* es el **azul noche**, una tonalidad moderna que proyecta tecnología y seguridad, pilares clave en la experiencia de estacionamiento que brinda nuestra app. Este color transmite calma y confianza, creando una interfaz amigable y profesional para los usuarios que navegan desde sus dispositivos móviles.

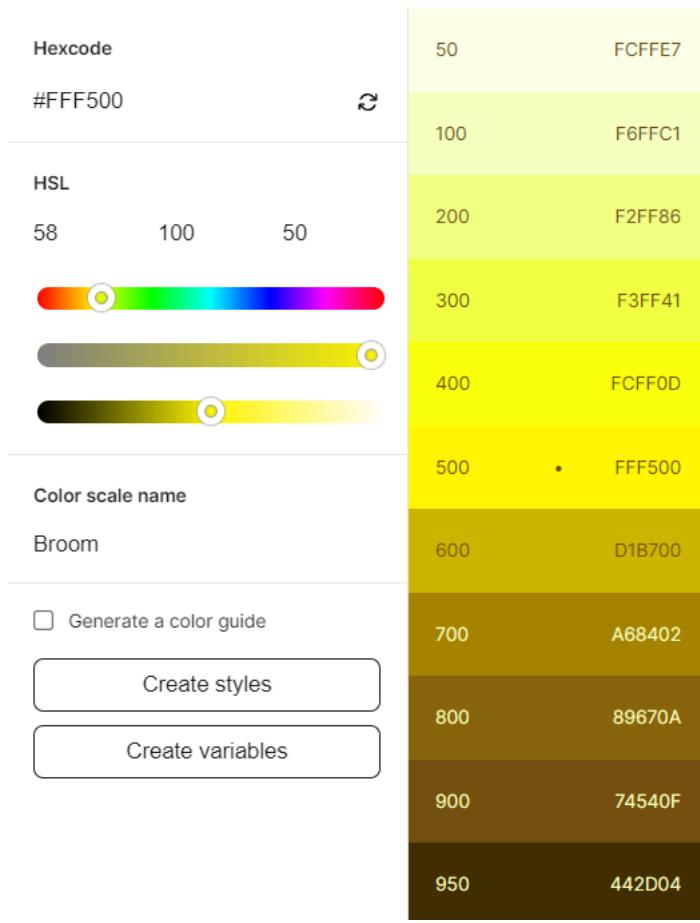
Esta tonalidad se ha aplicado especialmente en fondos, encabezados y componentes clave de navegación, manteniendo la consistencia visual en toda la aplicación. Además, se ha utilizado una escala de variaciones del azul noche para manejar estados, sombras y jerarquías de elementos.



🎨 Color Secundario – Amarillo (#FFF500)

El color **amarillo vibrante** fue elegido como tonalidad secundaria para la app móvil *ParkingNow*, debido a su capacidad para captar rápidamente la atención del usuario. Este tono resalta elementos clave de interacción dentro de la interfaz, como botones de acción, alertas destacadas o secciones prioritarias del flujo de uso.

Además de aportar un contraste dinámico con el color primario, el amarillo refuerza la percepción de eficiencia, visibilidad y rapidez del servicio de estacionamiento inteligente. Su presencia estratégica en la interfaz ayuda a guiar al usuario de forma intuitiva en su experiencia diaria.



Color Secundario – Gris (#828282)

El color **gris** ha sido incorporado como una tonalidad secundaria que aporta balance visual y profesionalismo a la interfaz de *ParkingNow*. Su neutralidad permite destacar los elementos primarios sin competir con ellos, ayudando a mantener una estética limpia y moderna en todo el diseño mobile.

Este tono se utiliza estratégicamente en componentes de menor jerarquía como bordes, divisores, placeholders o textos secundarios, brindando un contraste elegante y sutil que mejora la legibilidad y la claridad visual de la app.

UI Tailwind CSS Color Generator X

Hexcode Copy

HSL
0 0 51

Color scale name: Gray

Generate a color guide

Create styles

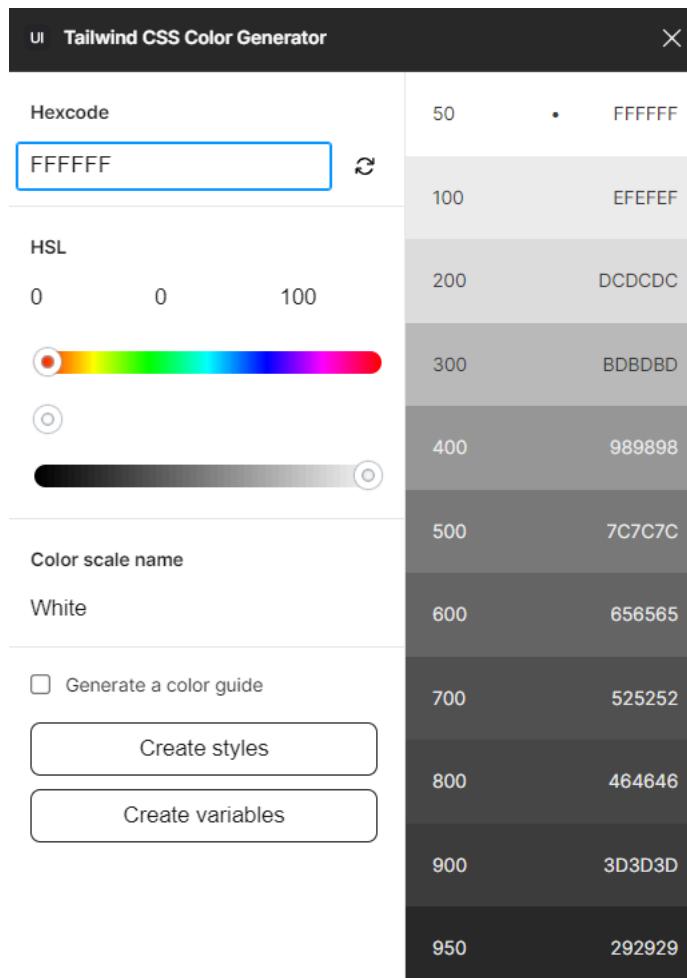
Create variables

50	F6F6F6
100	E7E7E7
200	D1D1D1
300	B0B0B0
400	828282
500	6D6D6D
600	5D5D5D
700	4F4F4F
800	454545
900	3D3D3D
950	262626

♡ Color Secundario – Blanco (#FFFFFF)

El blanco cumple un rol esencial dentro de la interfaz de *ParkingNow*, al aportar una sensación de limpieza, orden y simplicidad visual. Esta tonalidad se utiliza como fondo principal en muchas pantallas para mejorar la legibilidad de los textos y resaltar los elementos clave.

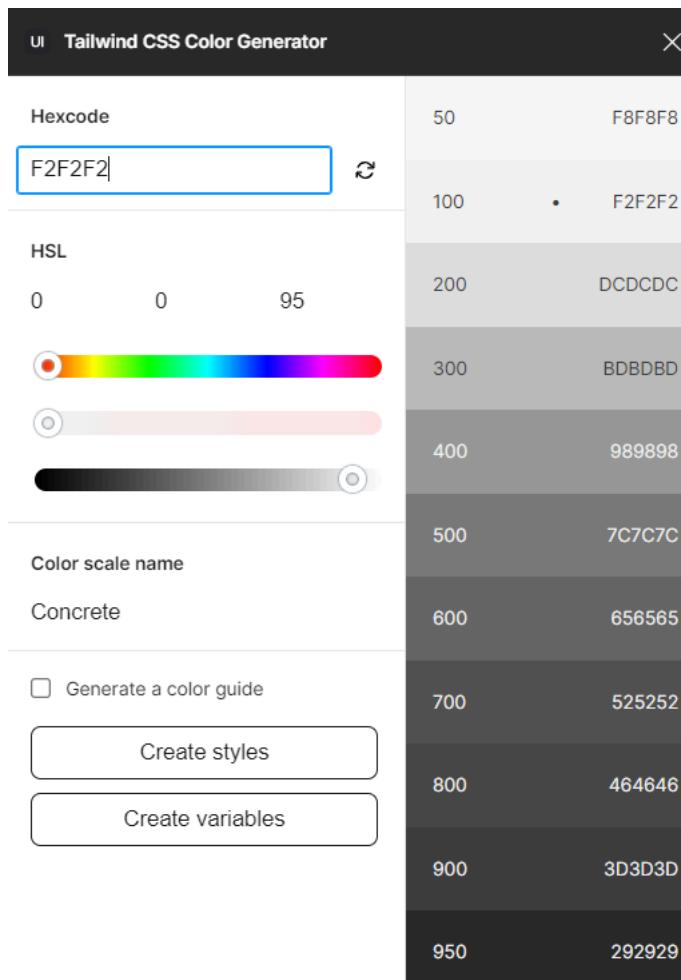
Gracias a su neutralidad, el blanco permite que los colores primarios y secundarios se destaquen adecuadamente, facilitando una experiencia de usuario clara, accesible y organizada en toda la aplicación.



○ Color Secundario – Gris Claro (#F2F2F2)

El tono `#F2F2F2` utilizado en *ParkingNow* pertenece a una escala gris clara que ofrece una sensación de **neutralidad y limpieza visual**. Este color se emplea principalmente como fondo en componentes secundarios o tarjetas, permitiendo que los elementos clave —como botones o textos— resalten con mayor claridad.

Su suavidad mejora la legibilidad y brinda un equilibrio visual elegante, siendo especialmente útil en interfaces móviles donde se busca evitar el uso excesivo de blanco puro, pero manteniendo una estética moderna y profesional.



5.1.2. Information Architecture

La arquitectura de información en *ParkingNow* ha sido diseñada siguiendo un enfoque mobile-first, con el objetivo de facilitar la navegación y el acceso rápido a las funcionalidades clave desde dispositivos móviles. La estructura de la app se organiza de forma jerárquica, priorizando flujos simples y directos como buscar estacionamiento, reservar un espacio, gestionar pagos y consultar el historial.

Se emplean etiquetas claras y concisas para los menús y botones, acompañadas de una navegación principal mediante una barra inferior que permite moverse entre las secciones más utilizadas. Además, se considera un sistema de búsqueda para ubicar rápidamente estacionamientos disponibles, mejorando así la eficiencia del recorrido del usuario dentro de la aplicación. Esta organización permite una experiencia ordenada, intuitiva y centrada en las necesidades reales del usuario móvil.

5.1.2.1. Organization Systems

Para estructurar la arquitectura de la información de *ParkingNow*, se ha adoptado un sistema de organización **jerárquico**, tanto en la aplicación móvil como en la landing page. Esta decisión permite ofrecer una navegación intuitiva y eficiente, en la que los usuarios pueden encontrar rápidamente la información o funcionalidad que necesitan según su perfil.

Aplicación Móvil

La aplicación está dividida en flujos personalizados para dos tipos de usuarios: **conductores urbanos frecuentes** y **propietarios de estacionamientos**. Cada flujo prioriza las tareas más comunes de cada perfil para reducir pasos innecesarios y mejorar la experiencia desde dispositivos móviles.

- **Para Conductores Urbanos Frecuentes:**
 - **Inicio:** Vista general de reservas activas y accesos directos a funcionalidades clave.
 - **Buscar Estacionamiento:** Búsqueda y visualización de estacionamientos disponibles en tiempo real mediante cámaras de monitoreo, con filtros por ubicación, precio y horario.
 - **Favoritos:** Listado de estacionamientos guardados para acceso rápido.
 - **Historial:** Registro de reservas pasadas que permite revisar actividad reciente.
- **Para Propietarios de Estacionamientos:**
 - **Inicio:** Panel con resumen de ocupación y notificaciones relevantes.

- **Monitoreo:** Gestión en tiempo real de la ocupación mediante cámaras de vigilancia conectadas.
- **Reservas:** Sección para aprobar o rechazar solicitudes de reserva.
- **Estadísticas:** Panel con gráficos sobre uso, disponibilidad e ingresos, ayudando a tomar decisiones operativas.

Este enfoque jerárquico permite separar de forma clara las funcionalidades principales según el tipo de usuario, minimizando la carga cognitiva y facilitando una experiencia centrada en tareas.

Landing Page

La página informativa (Landing Page) también sigue una estructura jerárquica, priorizando la información más relevante y guiando al usuario hacia la acción deseada (registro o inicio de sesión). Su diseño está optimizado para móviles y se compone de las siguientes secciones:

- **Barra de navegación superior** con enlaces directos a:
 - **Inicio:** Visión general de la app y botones principales para registrarse o iniciar sesión.
 - **Servicios:** Descripción de los beneficios que *ParkingNow* ofrece tanto a conductores como a propietarios.
 - **Planes:** Detalles de las opciones de transacción, condiciones y beneficios.
 - **Contáctanos:** Formulario y enlaces a redes sociales para contacto directo.
 - **Nosotros:** Información sobre la misión, visión y el equipo del proyecto.
- **Estructura de contenido clara:**
 - Uso de **encabezados jerárquicos** y **subsecciones** para organizar el texto.
 - **Botones de llamado a la acción (CTA)** colocados estratégicamente para guiar al usuario.
- **Pie de página (footer):**
 - Contiene enlaces complementarios como políticas de privacidad, términos de uso y redes sociales, ofreciendo navegación adicional.

Este sistema de organización refuerza la facilidad de uso en plataformas móviles, mejora la localización de contenidos y guía al usuario a través de rutas claras de interacción.

5.1.2.2. Labelling Systems

Los sistemas de etiquetado implementados en *ParkingNow* han sido diseñados para garantizar una navegación clara, accesible y rápida dentro de la aplicación móvil. El objetivo principal es que los usuarios identifiquen de inmediato el propósito de cada sección, función o acción, sin ambigüedades. Para ello, se utilizaron etiquetas consistentes, directas y visualmente reconocibles, divididas en tres tipos: textuales, de encabezado e icónicas.

Etiquetas de Encabezado (Headings)

En la landing page informativa y en la app móvil, se emplean encabezados jerárquicos que organizan visualmente el contenido y ayudan al usuario a ubicar las secciones principales. Estas etiquetas están adaptadas para pantallas pequeñas, con textos breves y descriptivos:

- **Inicio (Home):** Encabezado de bienvenida en la pantalla principal, acompañado de un banner e información general.
- **Servicios (Services):** Describe las funcionalidades disponibles tanto para conductores como para propietarios de estacionamientos.
- **Planes y Precios (Plans):** Presenta las diferentes modalidades de uso y transacción.
- **Contáctanos (Contact Us):** Sección de contacto con el equipo mediante formulario y redes sociales.
- **Sobre Nosotros (About Us):** Informa sobre la misión, visión y propósito del proyecto.

Etiquetas Textuales (Text Labels)

Las etiquetas textuales guían al usuario durante la interacción, principalmente en botones, formularios y menús. Se priorizó el uso de verbos de acción claros y sustantivos reconocibles, optimizados para la comprensión rápida en entornos móviles:

- **Buscar Estacionamiento (Find Parking):** Campo de búsqueda que permite ingresar una ubicación para encontrar espacios disponibles.
- **Reservar (Reserve):** Botón principal para confirmar la solicitud de estacionamiento.
- **Historial (History):** Acceso a la lista de reservas pasadas del usuario.
- **Favoritos (Favorites):** Sección donde se almacenan los estacionamientos preferidos.
- **Configuraciones (Settings):** Área donde el usuario puede modificar sus preferencias y datos personales.

Etiquetas Icónicas (Iconic Labels)

Complementando las etiquetas textuales, se utilizaron íconos visuales que refuerzan la comprensión y optimizan la navegación en dispositivos táctiles. Estos íconos son reconocibles a nivel universal y están alineados con las funciones clave de la app:

-  **Lupa:** Representa la acción de búsqueda de estacionamientos.
-  **Calendario:** Utilizado para seleccionar fecha y hora en las reservas.
-  **Estrella:** Indica un estacionamiento marcado como favorito.
-  **Coche:** Identifica funciones relacionadas con la ubicación o seguimiento del vehículo.

-  **Campana:** Notifica al usuario sobre alertas, recordatorios o actualizaciones relevantes.

Estos sistemas de etiquetado, combinando texto e iconografía, permiten que la navegación en *ParkingNow* sea clara, intuitiva y rápida, especialmente en el contexto de uso móvil donde la eficiencia y la comprensión inmediata son esenciales.

5.1.2.3. SEO Tags and Meta Tags

Para mejorar la visibilidad de la landing page de *ParkingNow* en los motores de búsqueda y garantizar una correcta visualización en distintos dispositivos, se han definido etiquetas SEO (Search Engine Optimization) y metaetiquetas (Meta Tags) esenciales. Estas optimizan el posicionamiento de la página y mejoran la experiencia de usuario al acceder desde navegadores móviles.

A continuación, se detallan las etiquetas implementadas:

```
<!-- Título de la página -->
<title>ParkingNow - Reserva tu estacionamiento fácilmente</title>

<!-- Descripción del sitio -->
<meta name="description" content="ParkingNow es una aplicación móvil que te permite encontrar y reservar estacionamientos de manera rápida y sencilla.">

<!-- Palabras clave -->
<meta name="keywords" content="ParkingNow, estacionamiento, app de parqueo, reserva de parqueo, movilidad urbana, aplicación móvil, fácil de usar, barato, conveniente, seguro, ubicación, mapa, horarios, precios, promociones, servicios adicionales" data-bbox="58 325 953 338">

<!-- Configuración responsive para móviles -->
<meta name="viewport" content="width=device-width, initial-scale=1.0, maximum-scale=1.0, user-scalable=no" data-bbox="58 355 953 368">

<!-- Autoría -->
<meta name="author" content="Equipo ParkingNow - © 2025. Todos los derechos reservados.">
```

Estas etiquetas SEO mejoran la visibilidad del proyecto en plataformas de búsqueda y garantizan una experiencia de usuario óptima desde cualquier dispositivo móvil.

5.1.2.4. Searching Systems

La aplicación móvil *ParkingNow* incorpora un sistema de búsqueda eficiente e intuitivo, diseñado para que los usuarios encuentren estacionamientos de manera rápida desde cualquier ubicación. Este sistema mejora la usabilidad al reducir el tiempo de búsqueda y permitir diferentes formas de explorar opciones, adaptadas al contexto urbano y a la experiencia en dispositivos móviles.

Se han implementado las siguientes modalidades de búsqueda:

-  **Búsqueda por texto:** El usuario puede ingresar directamente el nombre del estacionamiento o del propietario para filtrar resultados específicos.
-  **Búsqueda por ubicación:** Utiliza la geolocalización del dispositivo móvil para mostrar automáticamente los estacionamientos más cercanos disponibles.
-  **Búsqueda por mapa interactivo:** Mediante un mapa integrado, el usuario puede explorar visualmente los espacios disponibles en distintas zonas, tanto cercanas como alejadas. El mapa incluye marcadores, detalles en tiempo real y opciones de reserva directa desde la vista geográfica.

Estas funcionalidades están optimizadas para pantallas móviles, permitiendo búsquedas ágiles con pocos toques, mejorando la experiencia del usuario en situaciones de movilidad.

5.1.2.5. Navigation Systems

La navegación en *ParkingNow* ha sido diseñada para ser simple e intuitiva, permitiendo a los usuarios realizar acciones como iniciar sesión, registrarse, buscar estacionamientos o hacer reservas de forma rápida. La app utiliza una barra de navegación inferior que da acceso directo a las secciones principales, adaptándose a los hábitos de uso en dispositivos móviles. El diseño busca minimizar los pasos y guiar al usuario de manera fluida a través de cada funcionalidad.

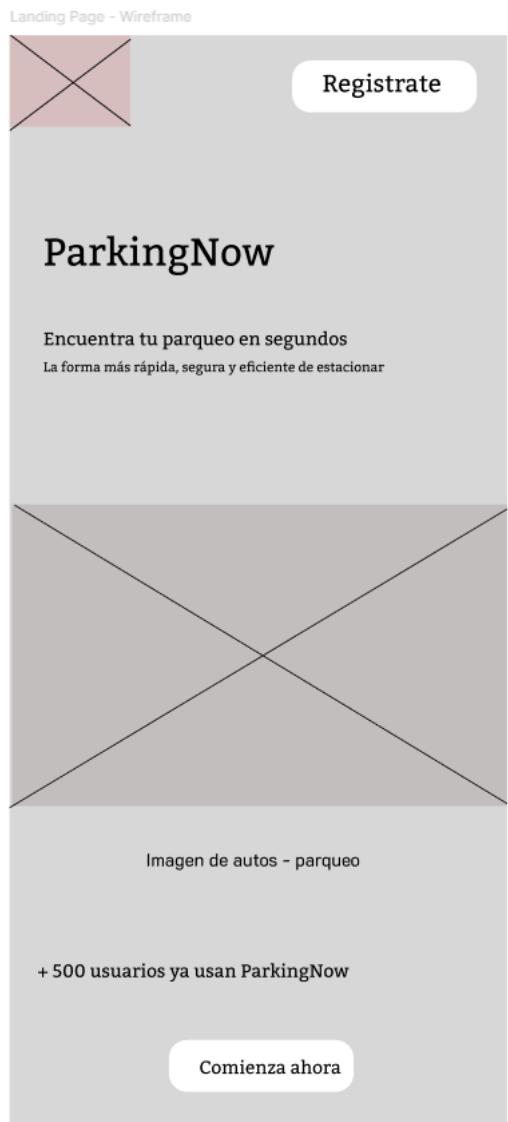
5.1.3. Landing Page UI Design

Se diseñó una landing page con enfoque mobile-first, utilizando un fondo gris claro (#F1F5F9) y acentos en amarillo para los botones principales. El diseño es simple y directo, con el logo en la parte superior, un mensaje claro y un botón de acción visible. Todo fue estructurado para facilitar la navegación desde dispositivos móviles.

5.1.3.1. Landing Page Wireframe

Se elaboró un wireframe de la landing page con enfoque mobile-first, priorizando una estructura vertical clara, botones visibles y secciones informativas. El diseño sirve como base para el desarrollo visual e implementación posterior de la página.

■ Wireframe 1: Encabezado y bienvenida



■ Wireframe 2: Acción y zonas disponibles

Ciudades donde puedes encontrar ParkingNow

Arequipa

Famosa por sus edificaciones de sillar blanco y la majestuosidad del volcán Misti que la rodea.

Frame 30

Lima

La vibrante capital de Perú, conocida por su rica historia, arquitectura colonial y vibrante vida nocturna.

Frame 28

Cusco

La antigua capital del Imperio Inca, famosa por su historia, ruinas arqueológicas y su cercanía a Machu Picchu.

■ Wireframe 3: Qué es ParkingNow y beneficios

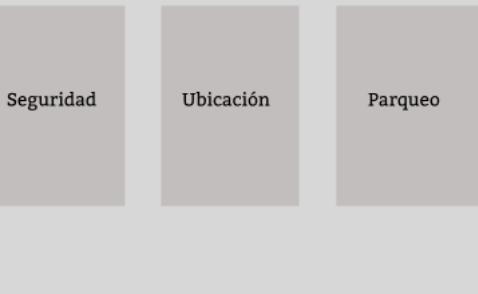
¿Qué es ParkingNow?

ParkingNow es una startup tecnológica enfocada en ofrecer una solución integral para el estacionamiento en tiempo real. Nuestra plataforma permite a los usuarios reservar y pagar por espacios de parqueo según su ubicación.

Puntos clave

- Localización geográfica precisa
- Visualización de espacios disponibles
- Optimización del tráfico en parqueos
- Integración con cámaras para supervisión
- Sistema seguro y moderno de reservas

Nuestros Pilares



■ Wireframe 4: Contacto y pie de página

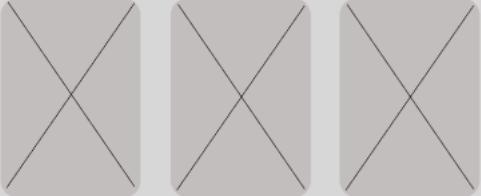
Contáctanos

✉ Correo:
ParkingNowApp@gmail.com

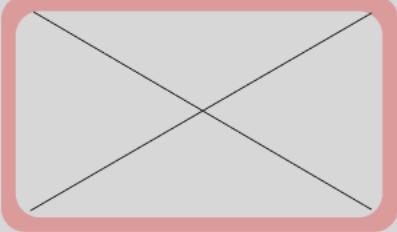
☎ Teléfono:
Atención al cliente: +51 922 000 000

📍 Dirección (opcional):
Lima, Perú

Nuestras redes



Soporte Técnico



Términos de uso | Política de privacidad

© 2025 ParkingNow. Todos los derechos reservados.

Link del figma: <https://www.figma.com/design/qb02D1snVoDtoLfT0qdkRV/Parking-now?node-id=0-1&t=vfQZ7EcT7cMorKUF-1>

5.1.3.2. Landing Page Mock-up

Se desarrolló el mock-up visual de la landing page aplicando la identidad gráfica de ParkingNow. Se incorporaron imágenes reales, íconos representativos, colores de marca como el amarillo para el botón principal y un diseño limpio enfocado en la experiencia móvil. Este mock-up representa cómo se verá la página final al implementarla.

■ Mock-up 1: Encabezado visual y CTA



Register

ParkingNow

Encuentra tu parqueo en segundos

La forma más rápida, segura y eficiente de estacionar



+ 500 usuarios ya usan ParkingNow

Comienza ahora

Mock-up 2: Ciudades con cobertura

Ciudades donde puedes encontrar ParkingNow



Arequipa: Famosa por sus edificaciones de sillar blanco y la majestuosidad del volcán Misti que la rodea.



Lima: La vibrante capital de Perú, conocida por su rica historia, arquitectura colonial y vibrante vida nocturna.



Cusco: La antigua capital del Imperio Inca, famosa por su historia, ruinas arqueológicas y su cercanía a Machu Picchu.

 Mock-up 3: Descripción y beneficios

¿Qué es ParkingNow?

ParkingNow es una startup tecnológica enfocada en ofrecer una solución integral para el estacionamiento en tiempo real. Nuestra plataforma permite a los usuarios reservar y pagar por espacios de parqueo según su ubicación.

Puntos clave

- Localización geográfica precisa
- Visualización de espacios disponibles
- Optimización del tráfico en parqueos
- Integración con cámaras para supervisión
- Sistema seguro y moderno de reservas

Nuestros Pilares



■ Mock-up 4: Contacto y redes sociales

Contáctanos

✉ Correo:
ParkingNowApp@gmail.com

📞 Teléfono:
Atención al cliente: +51 922 000 000

📍 Dirección (opcional):
Lima, Perú

Nuestras redes



Soporte Técnico

The wireframe shows a dark rectangular form with rounded corners. It contains four input fields with placeholder text: "Usuario.....", "Correo electronico.....", "Queja o consulta.....", and "Número de celular.....". Below these fields is a blue "Enviar" button.

[Términos de uso](#) | [Política de privacidad](#)

© 2025 ParkingNow. Todos los derechos reservados.

Link del figma: <https://www.figma.com/design/qb02D1snVoDtoLfT0qdkRV/Parking-now?node-id=0-1&t=vfQZ7EcT7cMorKUF-1>

5.1.4. Mobile Applications UX/UI Design

En esta sección se detalla el proceso de diseño UX/UI aplicado al aplicativo móvil de *ParkingNow*. El objetivo es garantizar una experiencia de usuario fluida e intuitiva, considerando las necesidades específicas de los usuarios al buscar, reservar y gestionar espacios de estacionamiento. Se abordarán aspectos clave como wireframes, diagramas de flujo, mock-ups visuales y prototipos interactivos que guían el desarrollo coherente de la interfaz móvil.

Link del figma: <https://www.figma.com/design/qb02D1snVoDtoLfT0qdkRV/Parking-now?node-id=0-1&t=vfQZ7EcT7cMorKUF-1>

5.1.4.1. Mobile Applications Wireframes

La imagen presenta los wireframes de alta fidelidad que representan el flujo principal de autenticación del aplicativo móvil *ParkingNow*. Se incluyen pantallas clave como la bienvenida, inicio de sesión, recuperación de contraseña, verificación mediante código y cambio de clave. Estas vistas permiten visualizar de forma clara y estructurada la secuencia de interacción que seguirá el usuario al acceder a su cuenta, garantizando una experiencia intuitiva, accesible y alineada con los principios de diseño centrado en el usuario.

Wireframes del flujo de autenticación

1.- Welcome



2.- Login

Login to your account

Phone Number
+6289876543210

Password

I don't have account

Login

11 - Forget Password

Forgot password

Please enter your phone number to change your password

Phone Number
+6289876543210

Continue

12 - Change Password

Change your password

New Password

Change Password

06 - Verification Code

Enter verification code

Please enter the verification code we sent to your mobile (+62) 898 7654 3210

Verification Code
P 9 7 5 3

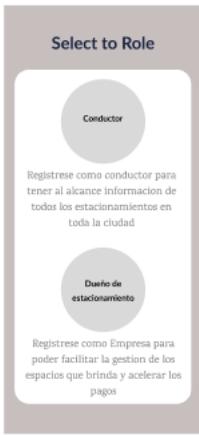
Resend Code

Verify

Estas pantallas muestran el flujo de selección de rol y registro personalizado según el tipo de usuario en la aplicación *ParkingNow*. El usuario puede elegir entre registrarse como **Conductor**, para acceder a información de estacionamientos disponibles en la ciudad, o como **Dueño de estacionamiento**, para gestionar sus espacios, agilizar pagos y tener control operativo. Cada opción conduce a un formulario adaptado, con campos específicos como RUC y nombre del establecimiento para dueños, o placa y DNI para conductores, asegurando un registro adecuado y enfocado en las necesidades de cada perfil.

Wireframes del registro por tipo de usuario

Select Role



Register dueño

← Register Dueño de estacionamiento

User Name: John Smith
Email: johnsmith@domain.abc
Password: *****
Confirmation Password: *****
RUC: Ingrese RUC de su estacionamiento
Registro del local: Ingrese el nombre del estacionamiento
Dirección del estacionamiento: Ingrese dirección

Register conductor

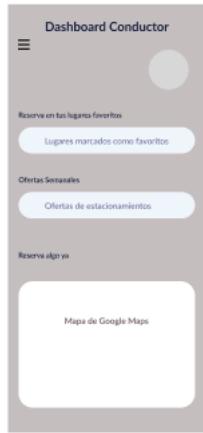
← Register Conductor

User Name: John Smith
Email: johnsmith@domain.abc
Password: *****
Confirmation Password: *****
Placa de vehículo: Ingrese la placa de su carro
DNI: Ingrese su DNI

Estas pantallas corresponden a los dashboards principales y menús laterales tanto para conductores como para dueños de estacionamientos dentro del aplicativo *ParkingNow*. El **dashboard del conductor** permite acceder rápidamente a lugares favoritos, ofertas y reservas activas, con integración visual mediante mapas. Por otro lado, el **dashboard del dueño de estacionamiento** facilita la gestión de espacios y la organización de reservas programadas. Ambos roles cuentan con un menú lateral personalizado que centraliza las funcionalidades según su tipo de usuario, mejorando la navegación y eficiencia de uso.

Wireframes de dashboards y navegación por rol

Dashboard conductor



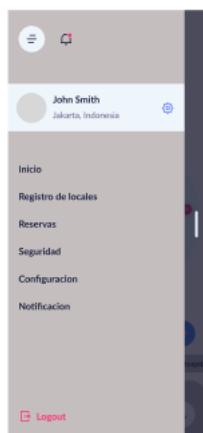
Dashboard dueño de est...



Opciones de conductor



Opciones de dueño



Estas pantallas representan funcionalidades complementarias del rol de conductor en la aplicación *ParkingNow*, enfocadas en brindar soporte, seguimiento, configuración y control de notificaciones. La sección de **Reservas** permite gestionar espacios ya agendados, mientras que el módulo de **Soporte** ofrece ayuda rápida ante incidencias. **Seguimiento** facilita visualizar el estado del estacionamiento reservado, y **Opciones** centraliza configuraciones como privacidad o seguridad. También se incluye la vista de **Notificaciones** relevantes para el usuario y una interfaz de **pago**, optimizada para procesar transacciones de manera segura y sencilla.

Wireframes de funcionalidades complementarias para conductores

Soporte conductor



Seguimiento conduc...



Opciones conductor



Notificaciones cond...



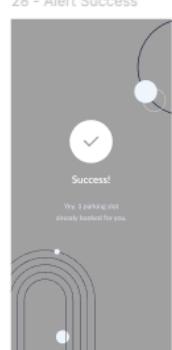
27 - Pay



29 - Ticket

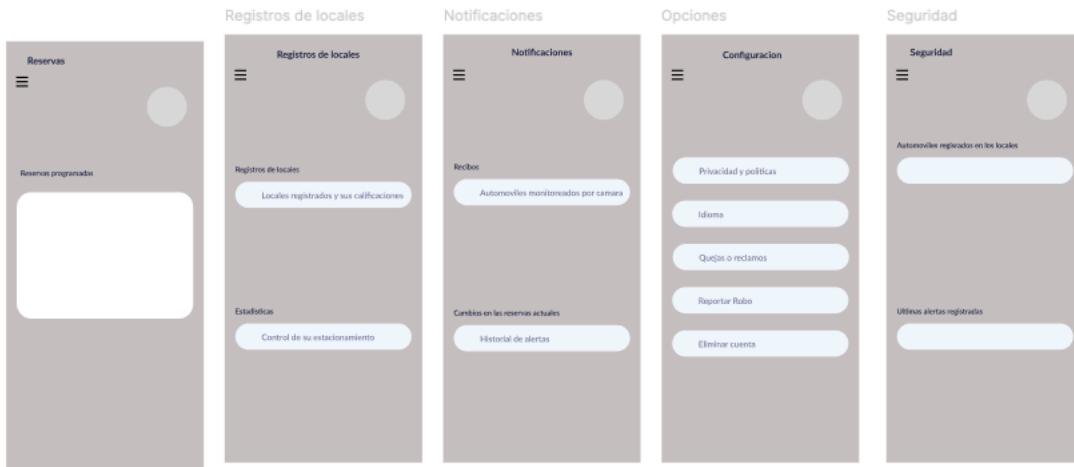


28 - Alert Success



Estas pantallas corresponden a las funcionalidades específicas para el rol de **dueño de estacionamiento** en la app *ParkingNow*. El módulo de **Registro de locales** permite dar de alta nuevos espacios con información clave como ubicación, seguridad y capacidad. Además, se incluyen vistas para **notificaciones** relacionadas con la actividad del local, así como una sección de **configuración** con opciones de perfil, soporte y seguridad. Esta última incluye el control de accesos y validación de entradas, aportando control y confianza en la gestión operativa del estacionamiento.

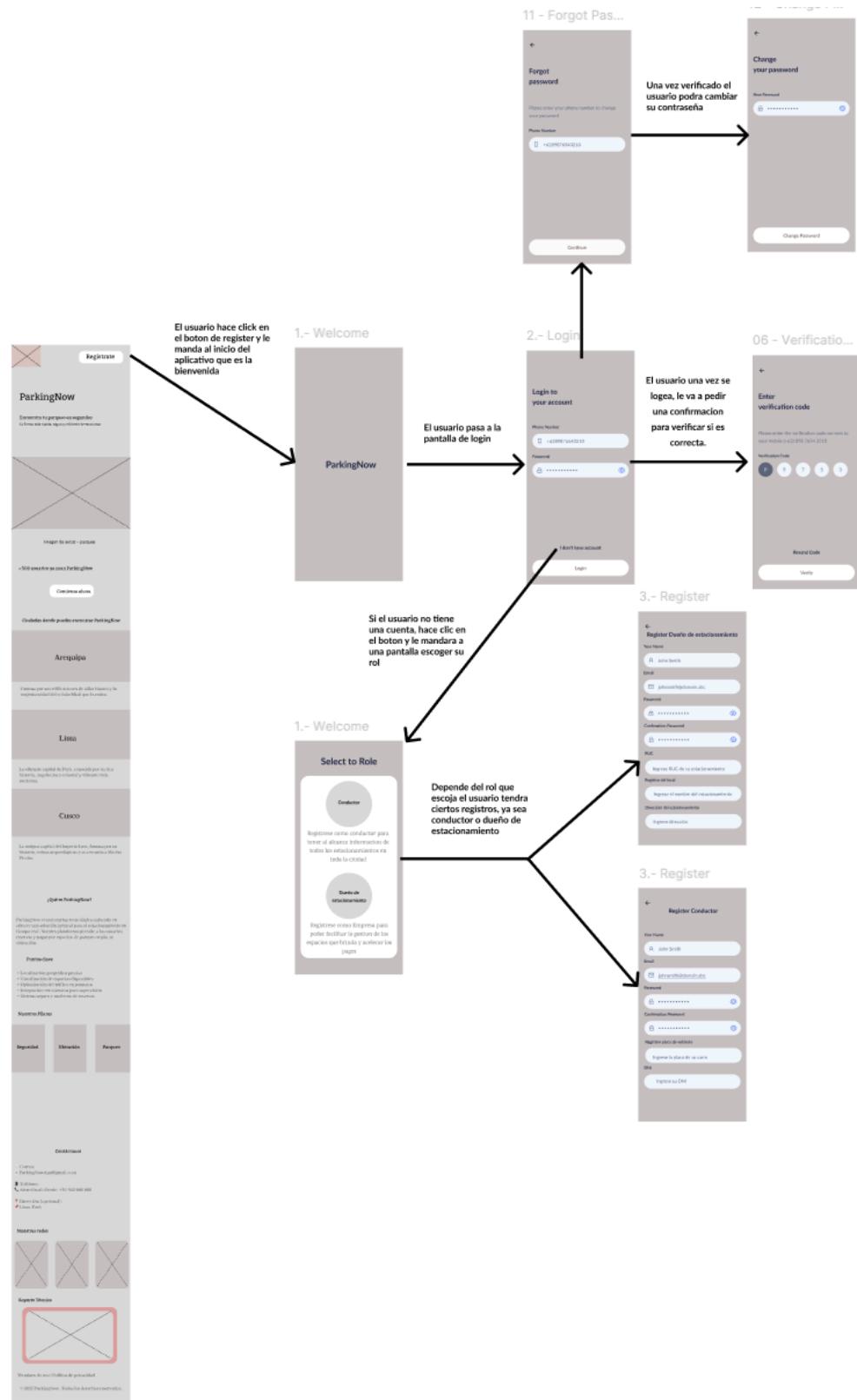
Wireframes de gestión para dueños de estacionamiento



5.1.4.2. Mobile Applications Wireflow Diagrams

User goal: Completar el registro e iniciar sesión en *ParkingNow* como conductor o dueño de estacionamiento.

Wireflow del proceso de registro e inicio de sesión



Wireflow de acceso a dashboard según tipo de usuario

05 - Confirmat...



06 - Verificati...



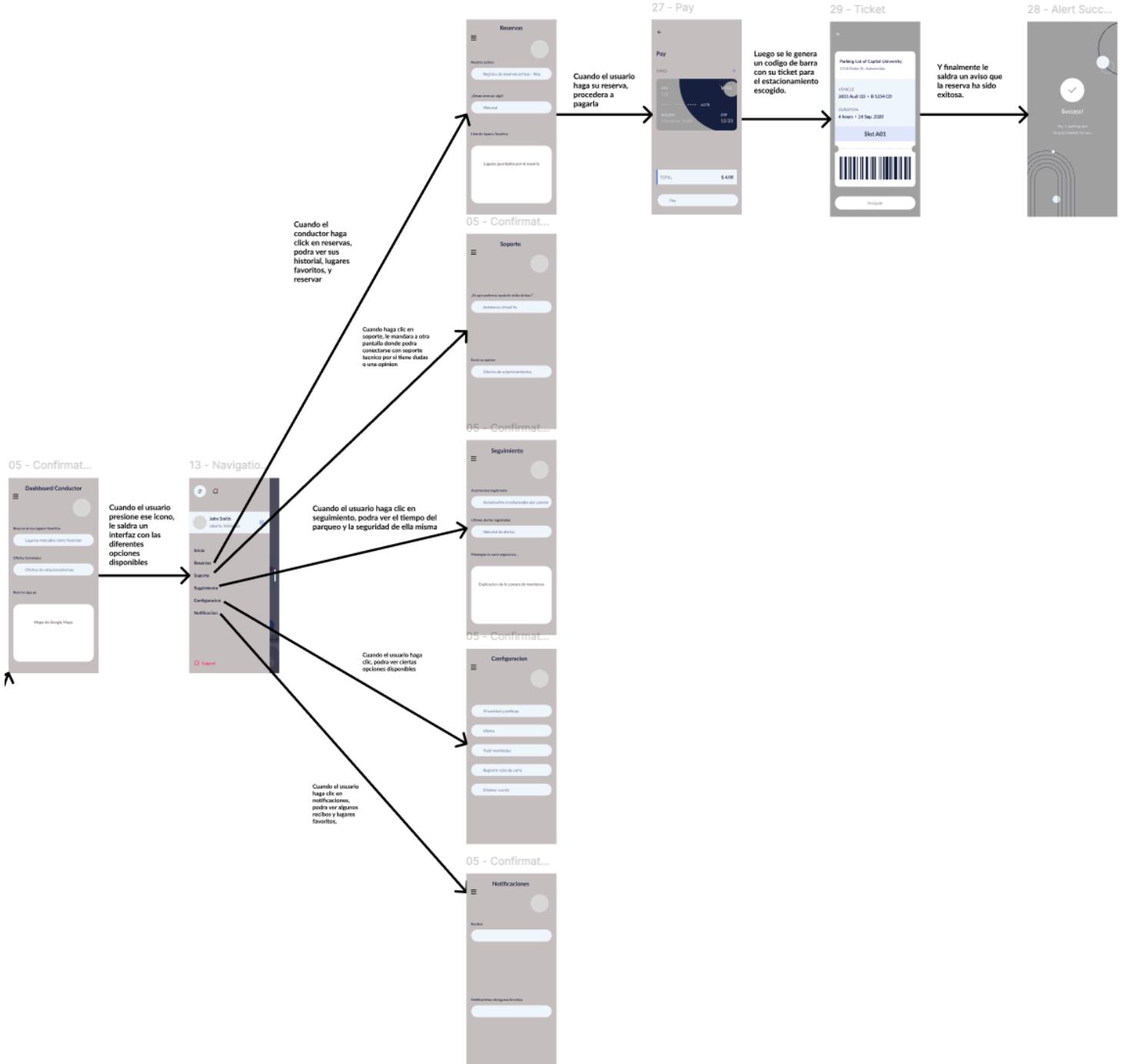
Si el logeo es exitoso,
lo mandara a su
dashboard correcto
dependiendo del rol
escogido

05 - Confirmat...



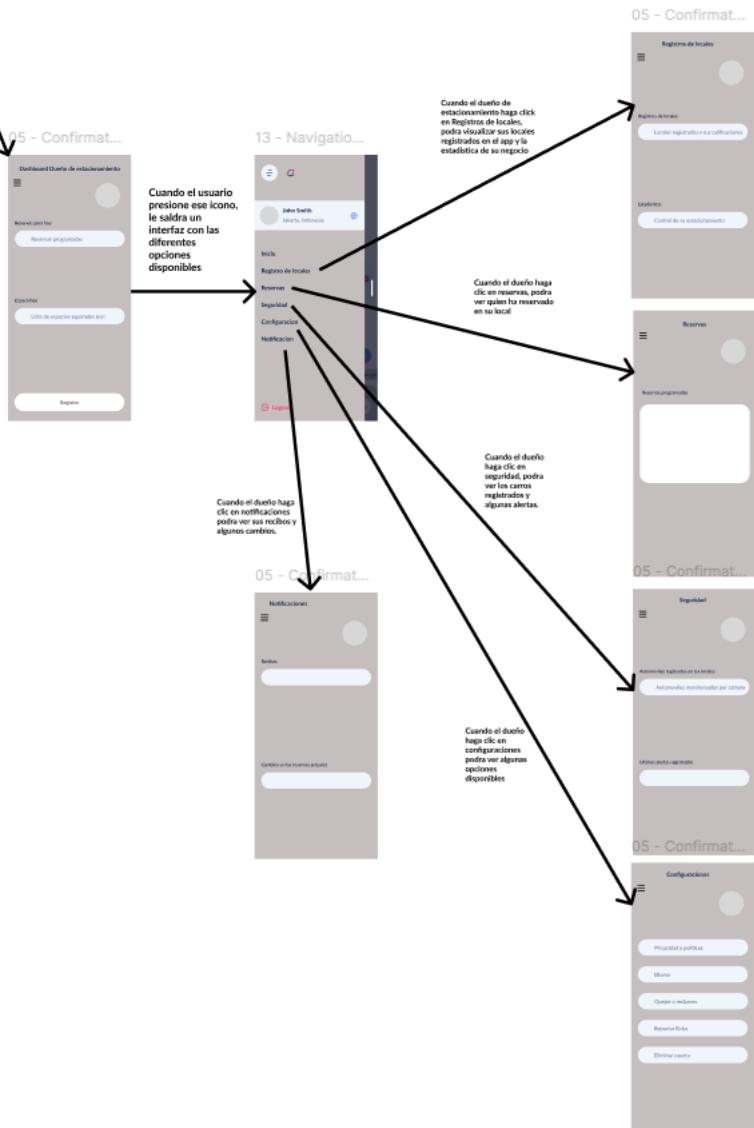
User goal: Navegar por el dashboard del conductor y completar una reserva de estacionamiento.

Wireflow de navegación del conductor y flujo completo de reserva



User goal: Navegar por las opciones administrativas del menú lateral como dueño de estacionamiento.

Wireflow de navegación y gestión para dueños de estacionamiento



Link del figma: <https://www.figma.com/design/qb02D1snVoDt0fT0qdkRV/Parking-now?node-id=0-1&t=vfQZ7EcT7cMorKUF-1>

5.1.4.3. Mobile Applications Mock-ups

Estas pantallas representan una versión visualmente refinada del flujo de autenticación del aplicativo móvil *ParkingNow*, aplicando principios modernos de diseño UI. Incluyen vistas clave como la bienvenida, inicio de sesión, recuperación de contraseña, verificación por código y cambio de clave. El estilo visual, basado en colores suaves, elementos geométricos y una disposición limpia, refuerza la identidad gráfica de la aplicación y mejora la experiencia del usuario desde el primer contacto.

Mockups estilizados del flujo de autenticación

1.- Welcome



2.- Login

 A login form titled 'Login to your account'. It contains fields for 'Phone Number' (with placeholder '+6289676543210') and 'Password' (with placeholder '*****'). Below the password field is a 'Forgot' link. At the bottom are 'I don't have account' and 'Login' buttons.

06 - Verification Code

 A verification code input screen titled 'Enter verification code'. It includes a note: 'Please enter the verification code we sent to your mobile (+62) 898 7654 3210'. A numeric keypad shows digits 0 through 9. At the bottom are 'Resend Code' and 'Verify' buttons.

11 - Forgot Password

 A forgot password screen titled 'Forgot password'. It asks for a phone number to change the password. The placeholder is '+6289676543210'. At the bottom are 'Continue' and 'Forgot' buttons.

12 - Change Password

 A change password screen titled 'Change your password'. It has a single field for 'New Password' with placeholder '*****'. At the bottom is a 'Change Password' button.

Estas pantallas muestran el flujo de selección de rol y registro personalizado para los usuarios de *ParkingNow*. El usuario puede elegir entre registrarse como **Conductor**, con acceso a todos los estacionamientos disponibles, o como **Dueño de estacionamiento**, con un formulario específico para registrar su espacio. Cada formulario incluye campos validados y adaptados a las necesidades del perfil, garantizando claridad, segmentación de funcionalidades y eficiencia desde el ingreso inicial.

Mockups del registro por tipo de usuario

Select role

Select to Role



Registrese como conductor para tener al alcance información de todos los estacionamientos en toda la ciudad



Registrese como Empresa para poder facilitar la gestión de los espacios que brinda y acelerar los pagos

registro de dueño de estacionamiento

Register Dueño de estacionamiento

Your Name
John Smith

Email
johnsmith@domain.abc

Password

Confirmation Password

RUC
Ingresé RUC de su estacionamiento

Registro del local

Ingresé el nombre del estacionamiento

Dirección del estacionamiento

Ingresé dirección

Registro de conductor

Register Conductor

Your Name
John Smith

Email
johnsmith@domain.abc

Password

Confirmation Password

Registrar placa de vehículo
Ingresé la placa de su carro

DNI
Ingresé su DNI

Estas vistas reflejan los dashboards iniciales y menús laterales disponibles para los diferentes tipos de usuario. El **Dueño de estacionamiento** accede a herramientas de gestión de locales, mientras que el **Conductor** puede visualizar y reservar espacios rápidamente. Los menús incluyen accesos directos a secciones como soporte, seguimiento, seguridad, configuración y notificaciones, facilitando la experiencia de navegación desde el primer acceso.

Mockups de dashboards y navegación lateral

Dashboard de dueño de estacionamiento



Reservas para hoy

Reservas programadas

Espacivhos

Lista de espacios separados por:

13 - Navigation Menu



John Smith
Jakarta, Indonesia

Inicio

Reservas

Soporte

Seguimiento

Configuracion

Notificacion

Logout

13 - Navigation Menu



John Smith
Jakarta, Indonesia

Inicio

Registro de locales

Reservas

Seguridad

Configuracion

Notificacion

Logout

Dashboard de conductor

Dashboard Conductor



Reserva en tus lugares favoritos

Lugares marcados como favoritos

Ofertas Semanales

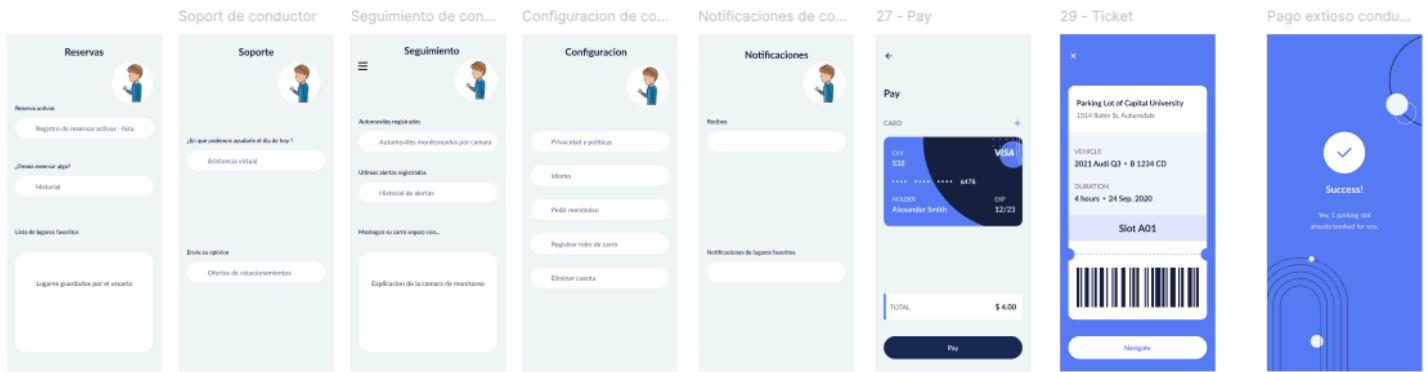
Ofertas de estacionamientos

Reserva algo ya

Mapa de Google Maps

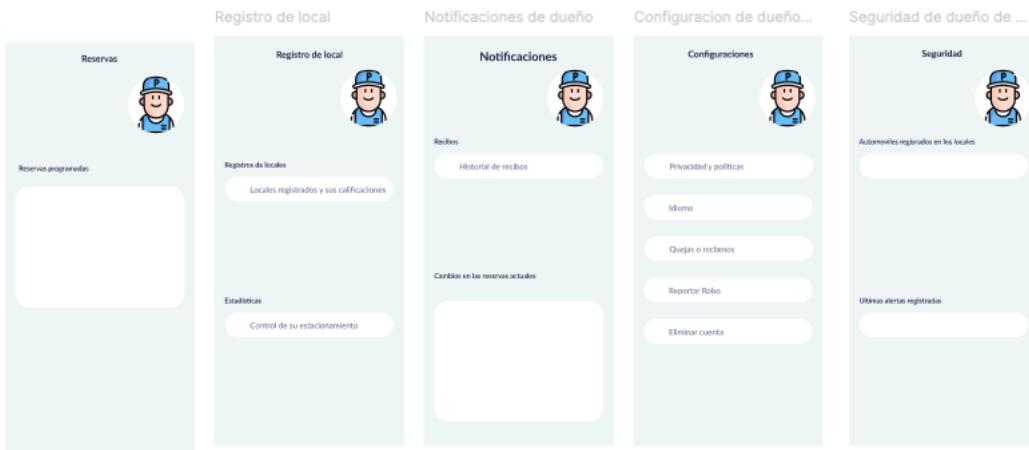
Esta sección agrupa las funcionalidades específicas para el rol de **Conductor** dentro de *ParkingNow*. Incluye soporte técnico, seguimiento de reservas activas, ajustes personales y visualización de notificaciones relevantes. También se muestra el proceso completo de **reserva y pago**, desde la selección del espacio hasta la confirmación final con emisión de ticket. Todo el flujo mantiene una estética clara, jerarquía visual coherente y accesibilidad móvil.

Mockups de funcionalidades y reservas para conductores



Aquí se visualizan las herramientas internas para los **Dueños de estacionamiento**, pensadas para facilitar la gestión eficiente de sus locales. Se incluyen vistas para registrar espacios, revisar notificaciones administrativas, configurar parámetros de operación, y validar aspectos de seguridad. Estas pantallas fortalecen el control del negocio dentro del ecosistema *ParkingNow*, asegurando trazabilidad y facilidad de uso en tareas frecuentes.

Mockups de gestión operativa para dueños de estacionamiento



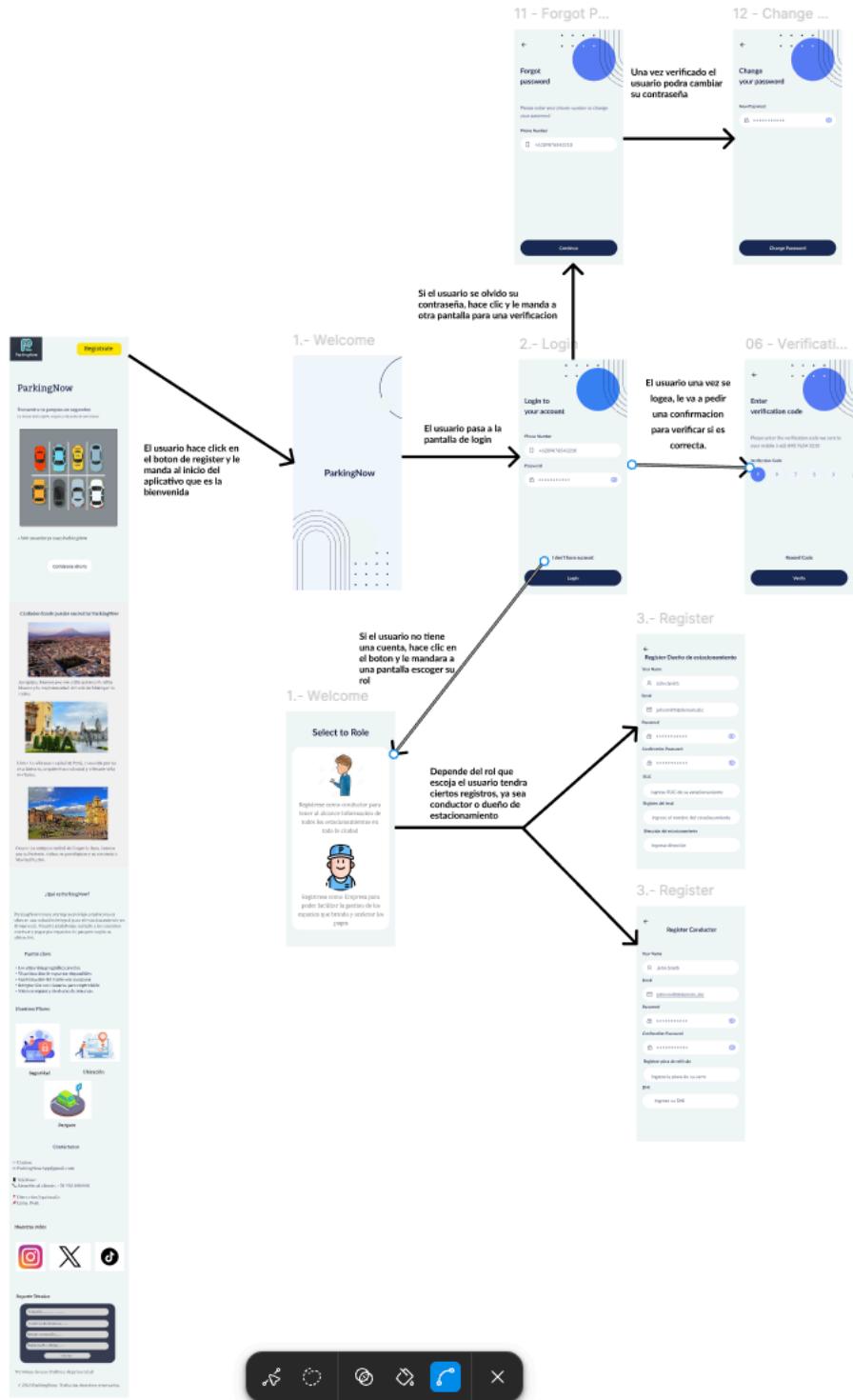
Link del figma: <https://www.figma.com/design/qb02D1snVoDtoLFT0qdkRV/Parking-now?node-id=0-1&t=vfQZ7EcT7cMorKUF-1>

5.1.4.4. Mobile Applications User Flow Diagrams

User goal: Completar el registro y autenticación para acceder a la aplicación según el tipo de usuario.

Este wireflow representa el recorrido del usuario desde el landing page hasta su ingreso efectivo a la aplicación *ParkingNow*. Luego de iniciar desde la vista principal, el usuario accede a la pantalla de bienvenida, seguido de login, selección de rol y formularios de registro específicos para conductores o dueños de estacionamiento. Se contempla además la verificación por código y el flujo de recuperación de contraseña, asegurando una experiencia guiada y segmentada según el perfil.

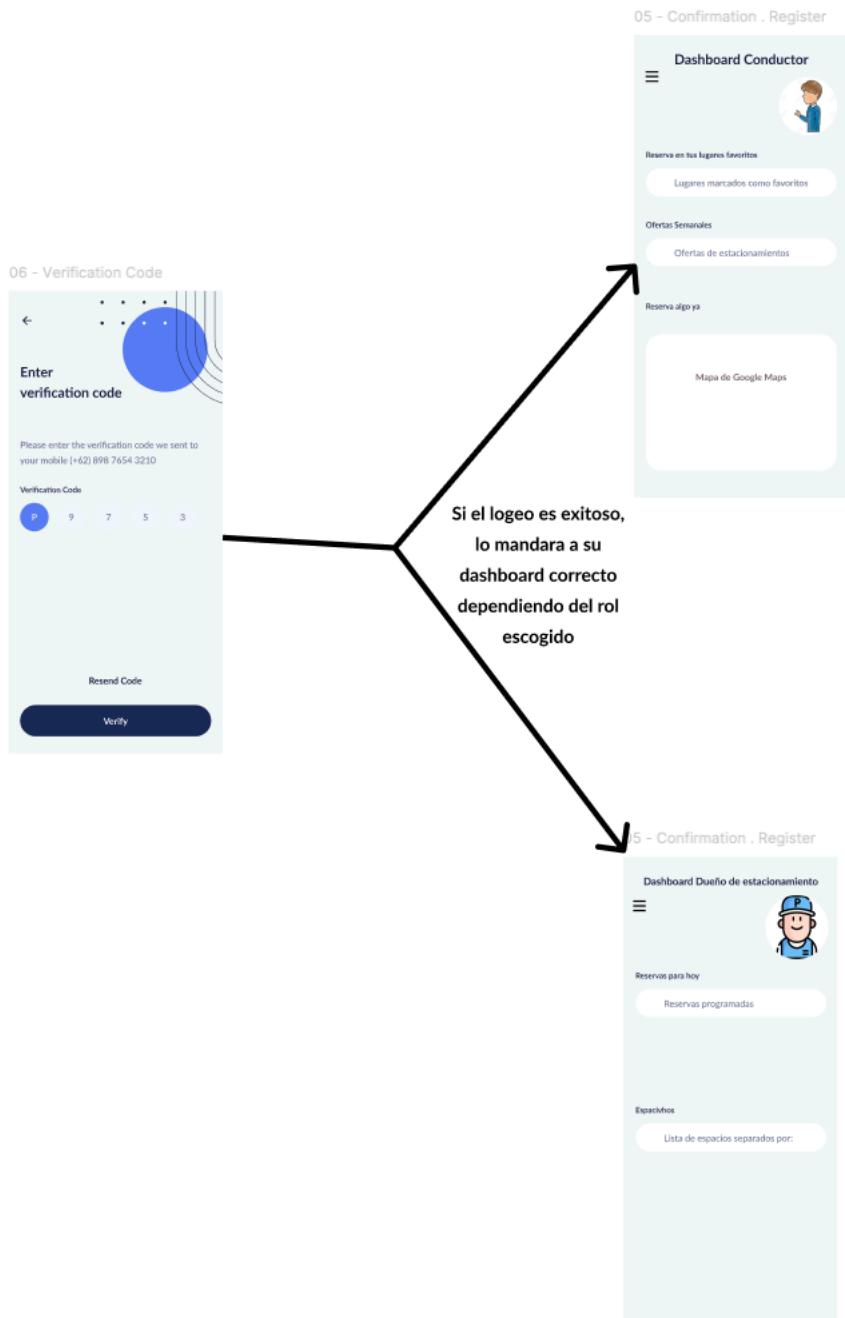
Wireflow del proceso completo de registro y autenticación



User goal: Redirigir al usuario al dashboard correspondiente luego de validar su código.

Este diagrama muestra la etapa final del login, donde tras ingresar el código de verificación correctamente, el usuario es redirigido automáticamente a su dashboard específico. Si el perfil registrado es de conductor, se carga una vista con mapas y accesos rápidos a estacionamientos. En cambio, si es dueño de estacionamiento, se presenta una interfaz orientada a la gestión de espacios. Esta lógica garantiza una experiencia diferenciada desde el primer acceso, alineada con las funcionalidades requeridas por cada tipo de usuario.

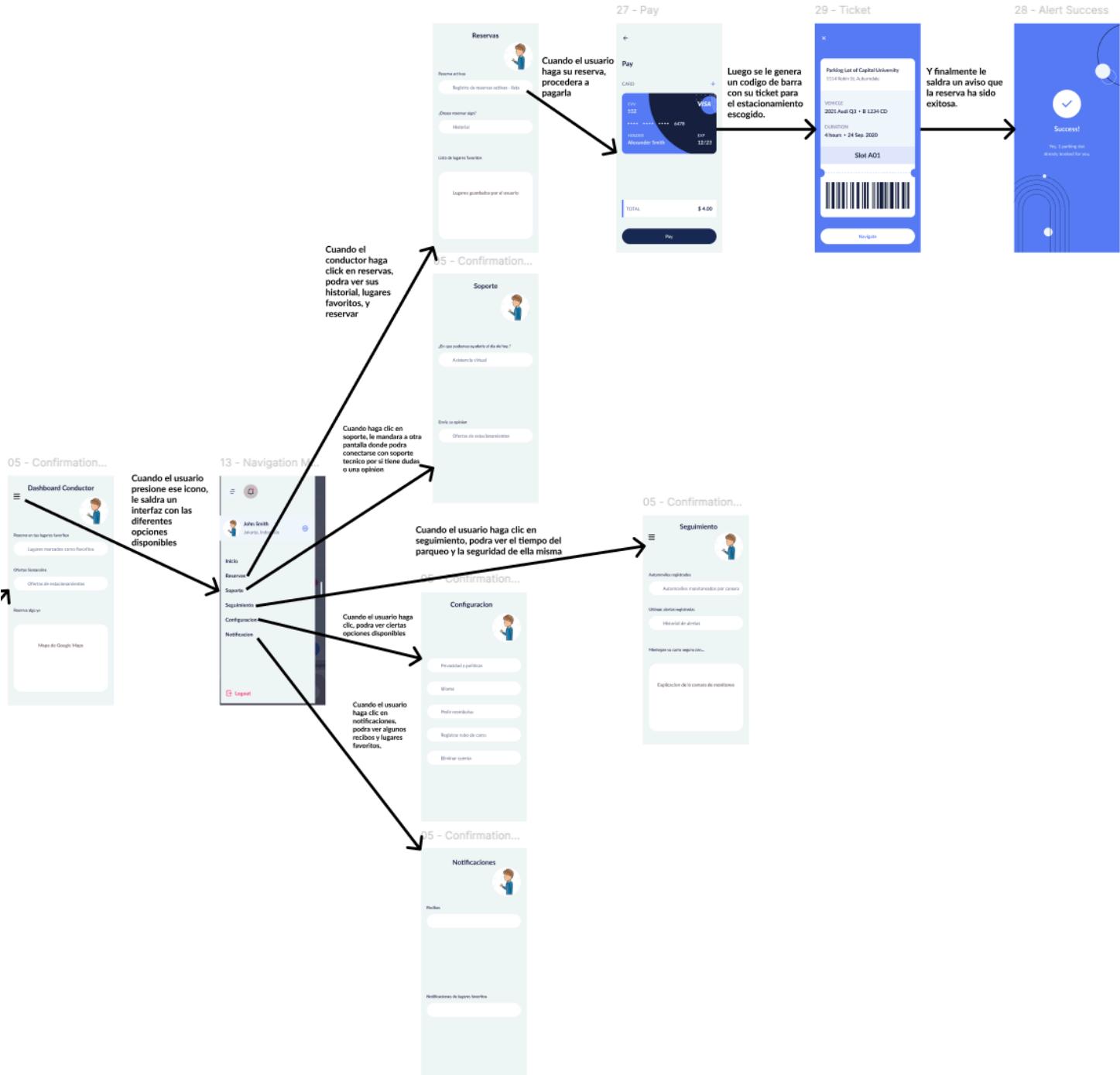
Wireflow de acceso a dashboards según perfil de usuario



User goal: Navegar por la aplicación como conductor y completar una reserva con confirmación.

Este flujo describe cómo el conductor puede desplazarse por las funcionalidades principales del menú lateral en *ParkingNow*, accediendo a soporte, seguimiento, configuración y notificaciones. El recorrido incluye la realización de una reserva de estacionamiento: el usuario selecciona un espacio, efectúa el pago correspondiente, recibe un ticket digital con código QR y visualiza una pantalla de éxito al final. Todo el proceso está diseñado para ser claro, rápido y completamente funcional desde el entorno móvil.

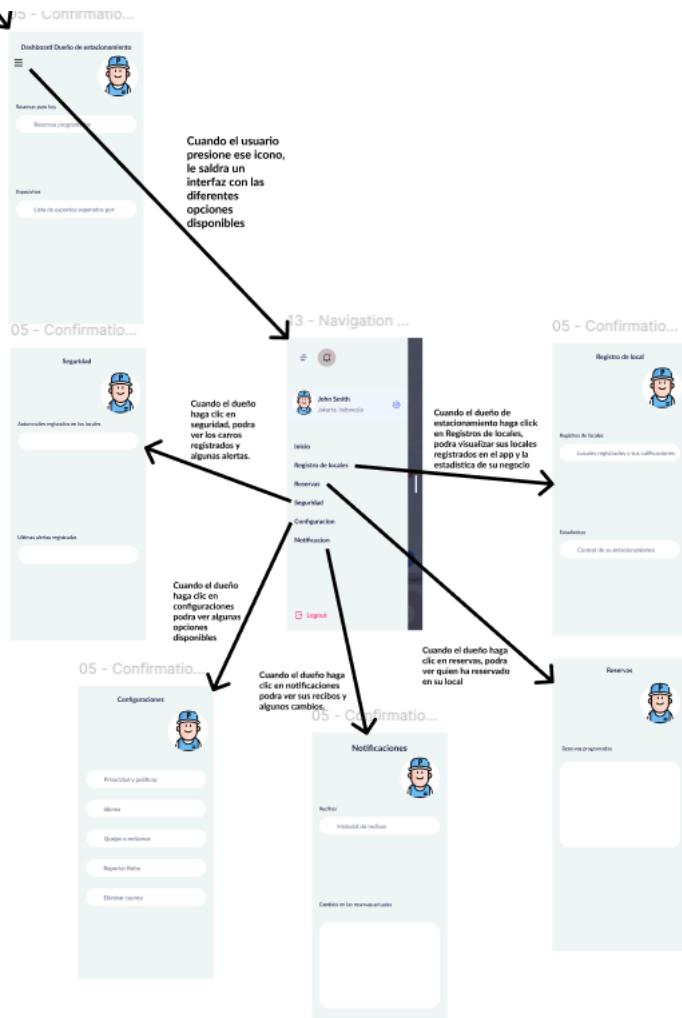
Wireflow de navegación y flujo de reserva para conductores



User goal: Gestionar espacios y configuraciones como dueño de estacionamiento desde el menú lateral.

Este wireflow muestra las acciones disponibles para el dueño de estacionamiento tras iniciar sesión. Desde el menú lateral puede acceder al registro de locales, visualizar notificaciones sobre actividades recientes, configurar parámetros de seguridad del sistema y acceder a secciones como configuración general o reservas activas. Cada vista está adaptada para optimizar el control administrativo y operativo del negocio, permitiendo gestionar recursos y validar información desde la app de forma eficiente y organizada.

Wireflow de navegación y gestión interna para dueños de estacionamiento

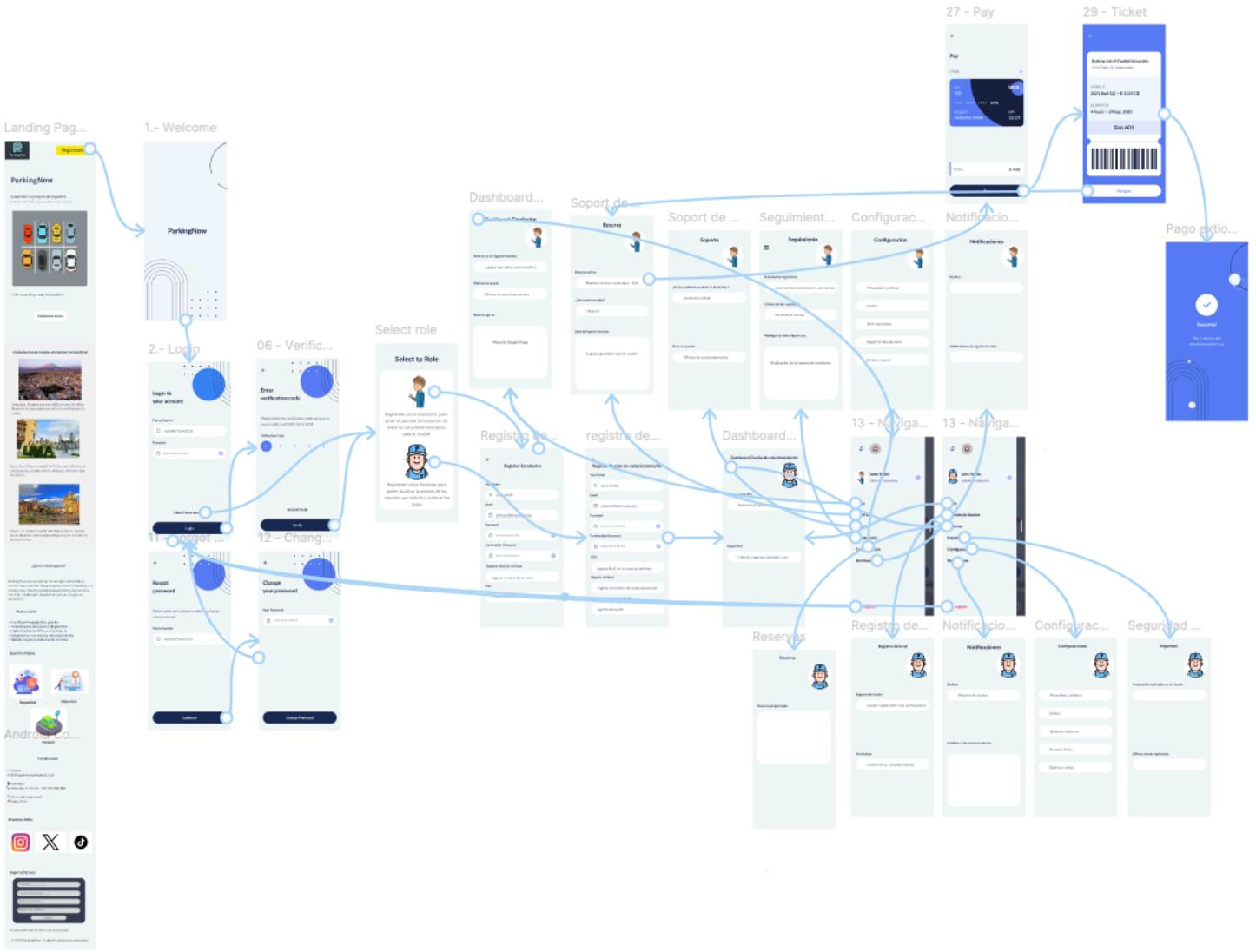


Link del figma: <https://www.figma.com/design/qb02D1snVoDtoLFT0qdkRV/Parking-now?node-id=0-1&t=vfQZ7EcT7cMorKUF-1>

5.1.4.5. Mobile Applications Prototyping

Este prototipo visual realizado en Figma muestra el recorrido completo del usuario dentro del aplicativo móvil *ParkingNow*, integrando tanto la experiencia del conductor como del dueño de estacionamiento. Incluye todas las rutas posibles desde el landing page, login, verificación, selección de rol y formularios de registro, hasta la navegación interna y funcionalidades clave como reservas, pagos, emisión de tickets, soporte, configuración y seguridad. Las flechas azules indican las conexiones interactivas establecidas entre pantallas, lo que permite validar la experiencia de usuario antes de su desarrollo.

Prototipo interactivo de alta fidelidad en Figma



Link del figma: <https://www.figma.com/design/qb02D1snVoDt0LfT0qdkRV/Parking-now?node-id=0-1&t=vfQZ7EcT7cMorKUF-1>

Capítulo VI: Product Implementation, Validation & Deployment

6.1. Software Configuration Management

Esta sección presenta las decisiones y convenciones adoptadas por el equipo para asegurar la consistencia técnica en la implementación del sistema, desde el desarrollo hasta el despliegue.

6.1.1. Software Development Environment Configuration

Esta sección detalla las herramientas y tecnologías que el equipo utilizará para colaborar eficazmente en el desarrollo, pruebas y despliegue de los productos digitales que componen la solución. Se especifica su propósito y la ruta de acceso correspondiente.

Requirements Management

Trello: Herramienta para gestionar proyectos bajo metodologías ágiles. Permite la planificación, seguimiento y priorización de tareas, así como la asignación de responsables durante cada sprint.



Ruta de referencia: <https://trello.com/es>

User Experience Design (UX/UI)

Figma: Plataforma colaborativa de diseño digital. En este proyecto se utilizará para crear prototipos de alta fidelidad, mockups y wireflows tanto para la aplicación móvil como para la web.



Ruta de referencia: <https://www.figma.com/login>

Software Testing

Gherkin + Cucumber: Lenguaje estructurado y herramienta que permiten definir criterios de aceptación, pruebas de integración y pruebas de aceptación automatizadas en formato legible. Su aplicación será en el backend desarrollado con Spring Boot.



Ruta de referencia: <https://cucumber.io/docs/gherkin/>

Software Development

Android Studio: IDE oficial para el desarrollo de aplicaciones móviles Android. Se utilizará para la primera versión de la aplicación nativa desarrollada en Kotlin.



Ruta de referencia: <https://developer.android.com/studio>

Kotlin: Lenguaje de programación moderno, interoperable con Java y orientado al desarrollo Android.



Ruta de referencia: <https://kotlinlang.org/>

Flutter (*planificado en siguientes entregas*): Framework UI de código abierto para desarrollar aplicaciones nativas multiplataforma desde una sola base de código.



Ruta de referencia: <https://flutter.dev/>

HTML5: Lenguaje de marcado para la estructura de contenido en la Landing Page.



Ruta de referencia: https://www.w3schools.com/html/html5_syntax.asp

CSS: Lenguaje de estilos usado para definir la presentación visual del contenido web en la Landing Page.



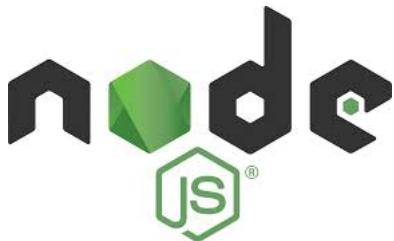
Ruta de referencia: <https://developer.mozilla.org/es/docs/Web/CSS>

JavaScript: Lenguaje de programación para la interactividad de la Landing Page.



Ruta de referencia: <https://developer.mozilla.org/es/docs/Web/JavaScript>

Node.js + NestJS: Lenguaje y framework utilizados para desarrollar los servicios RESTful del backend de manera eficiente, modular y escalable. NestJS aprovecha el motor de ejecución de Node.js junto con las ventajas de TypeScript, permitiendo una arquitectura limpia y mantenible.



Ruta de referencia Node.js: <https://nodejs.org>

Ruta de referencia NestJS: <https://nestjs.com>

IntelliJ IDEA Ultimate: Entorno de desarrollo integrado utilizado para construir y mantener el backend en Java con Spring Boot.



Ruta de referencia: <https://www.jetbrains.com/idea/>

Software Deployment

Git: Sistema de control de versiones que permite gestionar los cambios del código fuente y facilitar el trabajo colaborativo.



Ruta de referencia: <https://git-scm.com/>

Software Documentation and Project Management

GitHub: Plataforma de gestión de repositorios donde se almacenará el código fuente y documentación del proyecto. También se usará para la revisión de cambios y gestión de issues.



Ruta de referencia: <https://github.com/>

6.1.2. Source Code Management

Esta sección describe la organización del código fuente, el control de versiones y las convenciones de trabajo colaborativo que el equipo utilizará mediante GitHub. Se detallan las prácticas asociadas al uso de GitFlow y estándares para los nombres de ramas, versiones y mensajes de commit.

Repositorio principal

El equipo utilizará GitHub como sistema de gestión de versiones. La organización del proyecto se encuentra bajo el siguiente espacio:

- **Organización GitHub:** <https://github.com/G3-UPC-CC238-346-ParkingNow>
- **Landing Page:** <https://github.com/G3-UPC-CC238-346-ParkingNow/New-Landing-Page>
- **Web Services:** <https://github.com/G3-UPC-CC238-346-ParkingNow/Backend>
- **Mobile Application Flutter:** <https://github.com/G3-UPC-CC238-346-ParkingNow/Flutter-Mobile>
- **Mobile Application Kotlin:** <https://github.com/G3-UPC-CC238-346-ParkingNow/Kotlin-Mobile>

Modelo de Ramas: GitFlow

El equipo implementará el modelo de desarrollo basado en **GitFlow**, propuesto por Vincent Driessen. Este flujo de trabajo permite mantener un control ordenado y colaborativo del proyecto mediante ramas específicas para cada tipo de tarea.

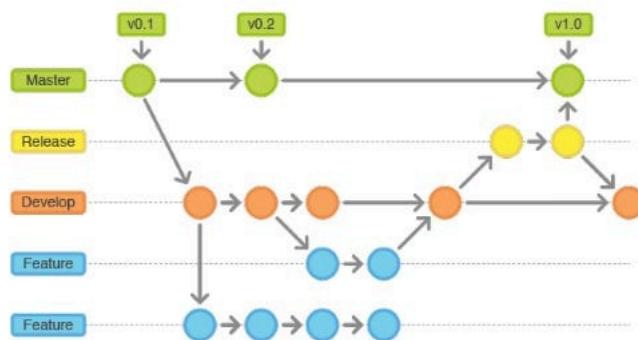
Ramas principales:

- `main` : Contendrá el código de producción estable.
- `develop` : Contendrá las últimas funcionalidades en desarrollo integradas, previo a su paso a producción.

Ramas auxiliares:

- `feature/*` : Ramas para el desarrollo de nuevas funcionalidades. Se deben nombrar con prefijo claro y descriptivo.
Ejemplo: `feature/login-screen` , `feature/rent-request-api`
- `release/*` : Ramas para preparar versiones de producción. Se crearán a partir de `develop` y se nombrarán con la versión.
Ejemplo: `release/1.0.0` , `release/2.1.0`
- `hotfix/*` : Ramas para corregir errores críticos directamente desde `main` .
Ejemplo: `hotfix/fix-payment-validation` , `hotfix/1.0.1`

Release Branches



Semantic Versioning

El equipo aplicará **Semantic Versioning 2.0.0** para el nombrado de versiones, usando el siguiente esquema:

MAJOR.MINOR.PATCH

- **MAJOR** : Cambios incompatibles con versiones anteriores.
- **MINOR** : Nuevas funcionalidades compatibles con versiones previas.
- **PATCH** : Corrección de errores sin afectar la compatibilidad.

Ejemplo de versión: 1.2.0 , 2.0.1

Referencias: <https://semver.org/>

Conventional Commits

Para los mensajes de commit, se seguirá el estándar **Conventional Commits**, con el fin de mantener un historial claro y automatizable.

Ejemplos de prefijos válidos:

- **feat** : Para nuevas funcionalidades.
- **fix** : Para corrección de errores.
- **docs** : Para cambios en documentación.
- **style** : Cambios de formato o estilo (sin cambios de lógica).
- **refactor** : Reestructuración del código sin cambiar funcionalidad.
- **test** : Agregado/modificación de pruebas.
- **chore** : Tareas de mantenimiento (builds, dependencias, etc.).

Ejemplo de commit:

```
feat: add rental request flow for university students
```

Referencias: <https://www.conventionalcommits.org/>

Tabla de Evidencia de Commits – Sprint 1

Commit ID	Repositorio	Rama	Autor	Fecha	Mensaje de Commit
c8b6e6d	Kotlin-Mobile	main	Diego Soto	16/05/2025	Complete registration and dashboard features
bbeed2a	Flutter-Mobile	main	Diego Soto	15/05/2025	Initial Flutter app structure with backend integration models and API ready
78db499	Landing Page	main	Diego Soto	13/05/2025	Fix responsive header on About page for mobile devices
eee2609	Landing Page	main	Diego Soto	13/05/2025	Update 404 page layout, fix footer position and apply background improvements
8c1f8ca	Landing Page	main	Diego Soto	13/05/2025	Update landing page: enhanced footer, social icons, and support section
f597680	Backend (NestJS)	main	Juan Jesús Calisaya	14/05/2025	feat: initialize NestJS project with default structure
33919ff	Backend (NestJS)	main	Samuel Molina	16/05/2025	feat(documentation) Documentation for endpoints
b0324fe	Backend (NestJS)	main	Samuel Molina	16/05/2025	feat(usuario) Adding API for Usuario Management
a1f7d92	Backend (NestJS)	main	Samuel Molina	16/05/2025	feat(reserva) Adding API for reserva management
a2c8c38	Backend (NestJS)	main	Samuel Molina	16/05/2025	feat(local) Adding API for Local management
d4c260b	Backend (NestJS)	main	Samuel Molina	16/05/2025	fix(app) Adding Services and Controllers
f6b7b5f	Backend (NestJS)	main	Samuel Molina	16/05/2025	fix(core) Adding dependencies

Web Services – Backend

El backend del proyecto está desarrollado con **NestJS + TypeScript** y estructurado por módulos independientes como `local` , `reserva` y `usuario` , cada uno con sus respectivos controladores, servicios, entidades y pruebas unitarias.

El repositorio está alojado en:

🔗 <https://github.com/G3-UPC-CC238-346-ParkingNow/Backend>

📁 Estructura real del backend:

```
src/
├── local/
│   ├── local.controller.ts
│   ├── local.service.ts
│   ├── local.entity.ts
│   └── local.controller.spec.ts ✅
├── reserva/
│   ├── reserva.controller.ts
│   ├── reserva.service.ts
│   ├── reserva.entity.ts
│   └── reserva.controller.spec.ts ✅
└── usuario/
    ├── usuario.controller.ts
    ├── usuario.service.ts
    ├── usuario.entity.ts
    └── usuario.controller.spec.ts ✅
├── app.controller.spec.ts ✅
├── app.module.ts
└── app.service.ts
└── main.ts
```

```
test/
├── app.e2e-spec.ts ✅
└── jest-e2e.json
```

📝 Pruebas y configuración

El proyecto ya incluye archivos de prueba unitaria (`*.spec.ts`) y pruebas end-to-end, los cuales están listos para ser ejecutados usando el comando: `npm run test`

Esto permitirá verificar individualmente la lógica de cada módulo. A pesar de que no se ha ejecutado localmente, el repositorio cuenta con la preparación técnica para validar controladores, servicios y flujos de integración conforme avancen los sprints.

🧠 Estándares aplicados

- ✅ **GitFlow** (`main` , `develop` , `feature/*` , `release/*` , `hotfix/*`)
- ✅ **Semantic Versioning** (`v1.0.0` , `v1.1.0`)
- ✅ **Conventional Commits** (`feat:` , `fix:` , `test:`)

Con esta estructura, el módulo de Web Services cumple con los estándares de calidad, pruebas y control de versiones exigidos por el curso.

6.1.3. Source Code Style Guide & Conventions

Esta sección detalla las guías de estilo y convenciones de codificación que seguirá el equipo para asegurar la legibilidad, coherencia y mantenibilidad del código fuente en todos los lenguajes usados. Se adoptan estándares reconocidos y nomenclatura en inglés para todos los elementos del código.

HTML & CSS

- Se aplicarán las guías del **Google HTML/CSS Style Guide**.
- Nombres de clases en CSS en **kebab-case**.
- Uso semántico de etiquetas HTML5.
- Separación clara entre estructura (HTML) y estilo (CSS).
- Ruta de referencia: <https://google.github.io/styleguide/htmlcssguide.html>

JavaScript

- Se sigue el estándar **Airbnb JavaScript Style Guide**.

- Uso de **camelCase** para variables y funciones.
- Constantes en **UPPER_SNAKE_CASE**.
- Ruta de referencia: <https://github.com/airbnb/javascript>

TypeScript (Backend con NestJS)

- Se aplica el **Google TypeScript Style Guide**.
- Uso de **camelCase** para variables y funciones.
- Clases nombradas en **PascalCase**.
- Separación modular entre `controllers`, `services`, `dto`, y `entities`.
- Se respeta el principio de responsabilidad única por archivo.
- Se aplican prácticas recomendadas de NestJS para mantener la inyección de dependencias, uso de decoradores y estructura clara.
- Ruta de referencia: <https://google.github.io/styleguide/tsguide.html>

Kotlin (Android)

- Se utiliza el **Kotlin Coding Conventions** oficial.
- Uso de **camelCase** para nombres de variables y funciones.
- Propiedades inmutables (`val`) por defecto.
- Nombres claros, concisos y en inglés.
- Ruta de referencia: <https://kotlinlang.org/docs/coding-conventions.html>

Gherkin

- Se sigue el estándar **Gherkin Conventions for Readable Specifications**.
- Escenarios escritos en inglés con formato: `Given`, `When`, `Then`.
- Archivos `.feature` organizados por funcionalidad principal.
- Ruta de referencia: <https://cucumber.io/docs/gherkin/>

Flutter (Dart)

- Se sigue la convención oficial **Dart Style Guide**.
- Uso de **camelCase** para nombres de variables, métodos y funciones.
- Nombres de clases y widgets en **UpperCamelCase** (PascalCase).
- Archivos organizados en carpetas como `screens/`, `widgets/`, `models/`, `services/`, según su responsabilidad.
- Cada archivo contiene una sola clase principal, manteniendo la modularidad y legibilidad.
- Se utilizan comentarios en inglés para describir funciones clave y lógica compleja.
- Se prioriza el uso de `const` y widgets inmutables para mejorar el rendimiento.
- Se evita la lógica de negocio en la capa de UI (`StatelessWidget`, `StatefulWidget`).
- Ruta de referencia: <https://dart.dev/guides/language/effective-dart/style>

6.1.4. Software Deployment Configuration

Se utilizó GitHub Pages para desplegar la landing page de manera estática, aprovechando la integración nativa de GitHub con Jekyll (motor de generación de sitios estáticos) y la automatización del despliegue mediante commits en la rama principal.

GitHub Pages:

es un servicio de alojamiento web gratuito proporcionado por GitHub que permite publicar sitios web estáticos directamente desde un repositorio. Está diseñado para proyectos personales, organizacionales o documentación, y es ampliamente utilizado para portafolios, blogs, landing pages y documentación técnica.

Link del despliegue del landing page: <https://g3-upc-cc238-346-parkingnow.github.io/New-Landing-Page/>

Backend (NestJS + Node.js)

Actualmente, la documentación Swagger **aún no se encuentra desplegada públicamente**, pero ya ha sido **implementada y completada en el entorno local**. La configuración de los endpoints y sus descripciones están listas, permitiendo probar y visualizar la API correctamente desde la máquina del desarrollador.

Se está trabajando en su despliegue en un entorno, lo cual permitirá el acceso remoto a la documentación desde cualquier navegador.

Aplicaciones móviles

Flutter

La aplicación desarrollada en Flutter será distribuida como un archivo instalable (.apk), el cual puede ser compartido para pruebas internas en dispositivos Android. Este archivo no genera un enlace web público, ya que se instala manualmente en celulares físicos o emuladores desde el entorno de desarrollo. Esta forma de despliegue permite validar la app antes de cualquier publicación en tiendas oficiales.

Kotlin (Jetpack Compose)

La aplicación móvil nativa en Kotlin también será compilada como archivo .apk , generado desde Android Studio. Al igual que en Flutter, este archivo no se despliega en una URL pública, sino que se instala directamente en dispositivos Android para fines de validación y pruebas. El despliegue se realiza de forma local en los equipos de desarrollo, sin requerir una publicación inmediata en Google Play.

6.2. Landing Page & Mobile Application Implementation

En esta sección se explica y documenta el proceso de implementación, pruebas, documentación técnica y despliegue de los tres componentes principales del proyecto:

- **Landing Page:** desarrollo del sitio informativo con diseño responsive.
- **Web Services:** construcción del backend con Node.js + NestJS, incluyendo controladores, servicios, pruebas unitarias y despliegue.
- **Mobile Applications:** implementación de las aplicaciones móviles utilizando Flutter (cross-platform) y Kotlin (Android nativo), con su respectiva estructura y empaquetado en archivos .apk .

Cada uno de estos bloques será evidenciado en los siguientes subapartados, organizados por Sprint.

6.2.1. Sprint 1

En este sprint se desarrollaron las HU US01 a US17, centradas en la Landing Page (navegación, beneficios, promociones, contacto, etc.) y el flujo de login y registro móvil para conductores y dueños. Estas funcionalidades forman la base inicial del proyecto ParkingNow.

6.2.1.1. Sprint Planning 1

Sprint #	Sprint 1
Sprint Planning Background	
Date	2025-05-02
Time	15:00 horas (GMT-5)
Location	Modalidad remota a través de Discord
Prepared By	Soto Quispe, Diego Ulises
Attendees (to planning meeting)	Calisaya Sánchez, Juan Jesús Espinoza Paredes, Frezzia Eldaa Isabel Hidalgo Lopez, Mathias Adriano Molina Asencios, Samuel Elias Soto Quispe, Diego Ulises
Sprint 1 – 1 Review Summary	Al tratarse del primer sprint del proyecto, no existe un sprint anterior que revisar. Este será el punto de partida para medir el progreso futuro.
Sprint 1 – 1 Retrospective Summary	Dado que este es el primer sprint del proyecto, no se identifican mejoras aún. Las retrospectivas comenzarán a partir del próximo sprint.
Sprint Goal & User Stories	
Sprint 1 Goal	Desarrollar la landing page con sus secciones, el backend funcional, y un primer avance de la app móvil. Al finalizar, tanto la landing como el backend deben estar desplegados y operativos.
Sprint 1 Velocity	116
Sum of Story Points	120

6.2.1.1. Sprint Backlog 1

Sprint #	Sprint 1						
User Story		Work-Item / Task					
Id	Title	Id	Title	Description	Estimation (Hours)	Assigned To	Status (To-do / In-Process / To-Review / Done)
US01	Navegación Intuitiva en la Landing Page	WT01	Implementación del header, estilos y barra de navegación	Realizar la barra de navegación utilizando JS/CSS	6	Diego Soto	Done
US02	Visualización de Ciudades con Servicio	WT02	Implementación de la sección de Ciudades	Realizar la sección que muestre las ciudades donde PARKINGNOW está disponible	4	Juan Calisaya	Done
US03	Promociones Destacadas	WT03	Implementación de la sección de Promociones	Mostrar promociones destacadas en la landing page	5	Frezzia Espinoza	Done
US04	Beneficios para Conductores y Dueños de estacionamiento	WT04	Implementación de la sección de Beneficios	Describir los beneficios de usar PARKINGNOW	4	Samuel Molina	Done
US05	Visualización de Misión de la Empresa	WT05	Implementación de la sección de Misión	Mostrar la misión de PARKINGNOW en la landing page	4	Mathias Hidalgo	Done
US06	Visualización de Visión de la Empresa	WT06	Implementación de la sección de Visión	Mostrar la visión de PARKINGNOW en la landing page	4	Diego Soto	Done
US07	Visualización del Equipo	WT07	Implementación de la sección del Equipo	Mostrar a los miembros del equipo de PARKINGNOW	4	Juan Calisaya	Done
US08	Información de Contacto	WT08	Implementación de la sección de Contacto	Proveer información de contacto de PARKINGNOW	4	Frezzia Espinoza	Done
US09	Envío de Mensaje de Contacto	WT09	Implementación del formulario de Contacto	Permitir a los visitantes enviar mensajes a PARKINGNOW	5	Samuel Molina	Done
US10	Visualización de Información de Características	WT10	Implementación de la sección de Características	Mostrar características del servicio PARKINGNOW	4	Mathias Hidalgo	Done
US11	Botón de Registro e Inicio de Sesión	WT11	Implementación de botones de Registro e Inicio de Sesión	Agregar botones para registro y acceso a cuentas	5	Diego Soto	Done
US12	Enlace a Redes Sociales	WT12	Implementación de enlaces a Redes Sociales	Proveer enlaces a las redes sociales de PARKINGNOW	4	Juan Calisaya	Done
US13	Visualización de Slogan	WT13	Implementación del Slogan	Mostrar un slogan atractivo en la landing page	4	Frezzia Espinoza	Done

Sprint #	Sprint 1						
User Story		Work-Item / Task					
Id	Title	Id	Title	Description	Estimation (Hours)	Assigned To	Status (To-do / In-Process / To-Review / Done)
		WT14	Implementación del botón Call To Action	Mostrar un botón atractivo en la landing page que redirige a la aplicación móvil	4	Juan Calisaya	Done
US14	Acceso a Información Legal	WT15	Implementación de enlaces a Información Legal	Proveer enlaces a políticas de privacidad y términos de uso	5	Samuel Molina	Done
US15	Inicio de Sesión para Conductores	WT16	Implementación de Inicio de Sesión	Desarrollar la funcionalidad de inicio de sesión para conductores en la aplicación.	6	Diego Soto	To-do
US16	Registro de Conductores	WT17	Implementación de Registro	Crear la funcionalidad de registro de conductores en la plataforma.	6	Juan Calisaya	To-do
US17	Registro de Dueños de estacionamiento	WT18	Implementación de Registro de Dueños	Desarrollar el registro para dueños de estacionamiento en la aplicación.	6	Frezzia Espinoza	To-do

Trello: <https://trello.com/invite/b/682808c86697c9a394dc8b12/ATTlida1c9216f53ce3913ed85ccb9445734eA3B9B093/parkingnow>

6.2.1.3. Development Evidence for Sprint Review.

Durante este sprint se completó el desarrollo de la landing page con todas sus secciones, así como la implementación básica del backend y el primer avance de la app móvil. Se entregaron funcionalidades clave como navegación, ciudades disponibles, formulario de contacto, y el inicio de sesión/registro. Todo fue validado en base a las HU del 1 al 17, y se presenta evidencia visual de cada uno para su revisión.

📋 Tabla de Evidencia de Commits – Sprint 1

Commit ID	Repositorio	Rama	Autor	Fecha	Mensaje de Commit
c8b6e6d	Kotlin-Mobile	main	Diego Soto	16/05/2025	Complete registration and dashboard features
bbeed2a	Flutter-Mobile	main	Diego Soto	15/05/2025	Initial Flutter app structure with backend integration models and API ready
78db499	Landing Page	main	Diego Soto	13/05/2025	Fix responsive header on About page for mobile devices
eee2609	Landing Page	main	Diego Soto	13/05/2025	Update 404 page layout, fix footer position and apply background improvements
8c1f8ca	Landing Page	main	Diego Soto	13/05/2025	Update landing page: enhanced footer, social icons, and support section
f597680	Backend (NestJS)	main	Juan Jesús Calisaya	14/05/2025	feat: initialize NestJS project with default structure
33919ff	Backend (NestJS)	main	Samuel Molina	16/05/2025	feat(documentation) Documentation for endpoints
b0324fe	Backend (NestJS)	main	Samuel Molina	16/05/2025	feat(usuario) Adding API for Usuario Management
a1f7d92	Backend (NestJS)	main	Samuel Molina	16/05/2025	feat(reserva) Adding API for reserva management
a2c8c38	Backend (NestJS)	main	Samuel Molina	16/05/2025	feat(local) Adding API for Local management
d4c260b	Backend (NestJS)	main	Samuel Molina	16/05/2025	fix(app) Adding Services and Controllers
f6b7b5f	Backend (NestJS)	main	Samuel Molina	16/05/2025	fix(core) Adding dependencies

6.2.1.4. Execution Evidence for Sprint Review.

A continuación, se presentan screenshots de las vistas desarrolladas durante el Sprint 1, incluyendo la landing page, el backend desplegado y el avance inicial de la app móvil. Además, se incluye un video que muestra la navegación y funcionalidades implementadas hasta el momento.

Landing

Pantalla principal de la Landing Page con secciones dinámicas

Si necesitas estacionamiento,
PARKINGNOW es la solución a tus problemas

Deja de preocuparte por el estacionamiento; PARKINGNOW se encarga de todo.

[Descubre PARKINGNOW](#)

Miles de conductores ya confían en PARKINGNOW para encontrar y reservar sus espacios de estacionamiento.



Ciudades donde puedes encontrar PARKINGNOW



Lima
La vibrante capital del Perú, conocida por su rica historia, arquitectura colonial y vida nocturna.



Arequipa
Famosa por sus edificios blancos de sillería blanca y la majestuosidad del volcán Misti que la rodea.



Cusco
La antigua capital del Imperio Inca, famosa por su historia y proximidad a Machu Picchu.

Promociones Destacadas



Sección de Promociones por Ciudad

Promociones Destacadas



Promoción en Lima

¡20% de descuento en tu primera reserva!



Promoción en Arequipa

¡Reserva ahora y obtén un 15% de descuento!



Promoción en Cusco

¡10% de descuento en todos los espacios!

Beneficios del Servicio para Usuarios y Propietarios

Beneficios del Servicio

Conductores Urbanos Frecuentes

PARKINGNOW ofrece una experiencia de estacionamiento sin problemas para conductores urbanos, permitiéndoles encontrar y reservar espacios en tiempo real. Ahorra tiempo, reduce el estrés y accede a información útil sobre la disponibilidad de espacios.

Propietarios de Estacionamientos

Con PARKINGNOW, los propietarios pueden gestionar sus espacios fácilmente. Nuestra plataforma ayuda a monitorear disponibilidad, establecer precios y atraer nuevos clientes, mejorando la ocupación y tus ganancias.

Integración Comercial

Ofrecemos integración con plataformas de terceros, permitiendo a negocios incorporar nuestro servicio como valor añadido. Mejora la fidelidad de tus clientes y optimiza operaciones comerciales.

Características de Seguridad y Confianza de la App

Seguridad y Confianza



Una App que te Cuida

Pensamos en ti y en tus necesidades; tu seguridad es nuestra prioridad. Nos comprometemos a brindarte una experiencia segura.



Monitoreo en Tiempo Real
Captura de pantalla 2025-03-25 a la(s) 18:49:20

Con nuestro monitoreo basado en nuestro sistema de monitoreo de cámaras, puedes verificar que tu vehículo permanezca donde lo estacionaste.



Tu Lugar, Tu Elección

Consulta opiniones de clientes y lugares de estacionamiento para ver dónde te quedarás y en quién confías tu vehículo.

Banner de Llamado a la Acción para Reserva Rápida

**¿BUSCAS UN ESTACIONAMIENTO?
¡HAZLO AQUÍ EN SEGUNDOS!**

Con PARKINGNOW puedes encontrar y reservar tu espacio en segundos.

[Comienza Ahora](#)



Formulario de Contacto para Soporte Inmediato

Soporte Rápido

Nombre

Correo electrónico

Escribe tu mensaje aquí...

Enviar

Sección de Misión de la Plataforma

Parkinghow

Inicio Nosotros Características Contacto

Misión

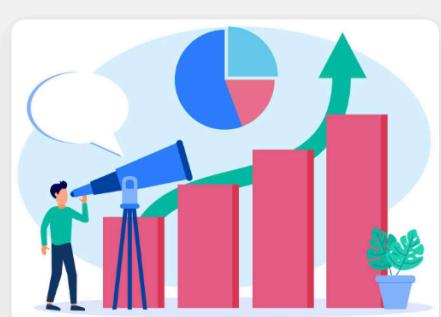
Nuestra misión es revolucionar la forma en que las personas encuentran y gestionan espacios de estacionamiento, ofreciendo una solución eficiente, confiable y accesible desde cualquier dispositivo.



Sección de Visión de la Plataforma

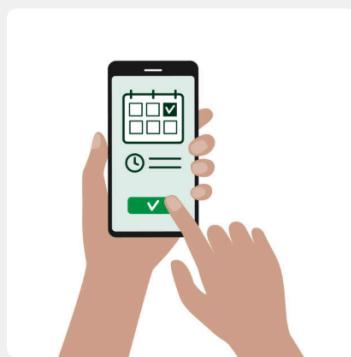
Visión

Ser la plataforma líder de estacionamiento inteligente en Latinoamérica, brindando una experiencia segura, rápida y tecnológica tanto para conductores como propietarios.



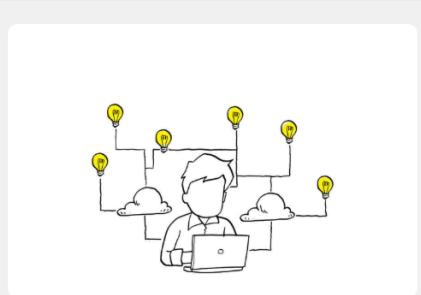
Características Clave del Servicio

Características Clave



Reserva en Tiempo Real

Encuentra y asegura tu espacio de estacionamiento en cuestión de segundos desde cualquier



Gestión Inteligente

Los propietarios pueden monitorear ocupación, establecer precios y recibir pagos con facilidad.



Integración Empresarial

Incorpora nuestro sistema en tu negocio para ofrecer valor agregado a tus clientes.



Pantalla de Inicio de la Aplicación Móvil

4:05

94%

Tu estacionamiento ideal,
más cerca que nunca.



ParkingNow

Encuentra tu espacio con un solo
toque

Empezar

Pantalla de Inicio de Sesión

4:05

剩余电量 94%

Login to your account



Correo electronico



Password



[Forgot password?](#)

Login

[I don't have account](#)

Pantalla de Recuperación de Contraseña

4:05

剩余电量 94%



Forgot password

Please enter your phone number to change
your password

Correo electronico



ejemplo@gmail.com

Continue

Cambiar Contraseña

4:05

剩余电量 94%



Change your password

New Password





Change Password

Selecciona tu Rol

Select to Role



Regístrate como conductor para tener al alcance información de todos los estacionamientos en toda la ciudad



Regístrate como Empresa para poder facilitar la gestión de los espacios que brinda y acelerar los pagos

Registro de Conductor



Register Conductor

Your Name



John Smith

Email



johnsmith@domain.abc

Password



Confirmación Password



Registrar placa de vehículo

Ingrese la placa de su carro

DNI

Ingrese su DNI

Registrarse

Panel Principal del Conductor

4:05

94%

≡ Dashboard Conductor



Reserva en tus lugares favoritos

Lugares marcados como favoritos

Ofertas Semanales

Ofertas de estacionamientos

Reserva algo ya

Mapa de Google Maps

Menú de Navegación del Conductor



John Smith
Lima, Perú

4%

ctor



s

favoritos

entos

Inicio

Reservas

Soporte

Seguimiento

Configuración

Notificación

ips

Logout



Registro de Dueño de Estacionamiento

卷之三



Register Dueño de estacionamiento

Your Name

John Smith

Email



johnsmith@domain.abc

Password



Confirmación Password



RUC

Ingrese RUC de su estacionamiento

Registro del local

Ingrese el nombre del

Dirección del estacionamiento

Inrese dirección

Registrarse

Dashboard de Dueño de Estacionamiento

4:05

94%



Dashboard Dueño de estacionamiento

Reservas para hoy

Reservas programadas

Espacios:

Lista de espacios separados por:

Menú de Navegación - Dueño de Estacionamiento



John Smith

Lima, Perú



namiento

Inicio

Registro de locales

Reservas

Seguridad

Configuración

Notificación

or:

Logout

Backend

Se evidencia el demo local del backend en la siguiente imagen

The screenshot shows two windows side-by-side. On the left is a REST client interface with a collection named 'parkingnow'. It lists several endpoints: 'GET get' (selected), 'POST Crear nueva reserva', 'POST Crear un nuevo usuario', 'PUT Actualizar por ID', and 'DEL Eliminar por ID'. A 'Body' tab is open for the 'POST Crear nueva reserva' endpoint, showing a JSON payload:

```
1. {  
2.   "id": 1,  
3.   "usuario": { "id": 2 },  
4.   "local": { "id": 6 },  
5.   "nro_plaza": 5,  
6.   "fh_inicio": "2025-05-16T08:00:00Z",  
7.   "fh_fin": "2025-05-16T10:00:00Z",  
8. }  
9.  
10. }  
11.  
12. }
```

On the right is a code editor window showing the file 'local.service.ts' from the 'src/local' directory. The code defines a service class 'LocalService' with methods for finding all local entries, finding one by ID, and creating a new local entry. The 'create' method includes logic to validate the user ID.

Video explicativo: https://upcedupe-my.sharepoint.com/:v/g/personal/u202214477_upc_edu_pe/EYOajxvFR85Nu8jiZWjN-

LMBsXskewjVOlyAhocolZQdGQ?

e=9BCcb5&nav=eyJzZWlcnJhbEluZm8iOnsicmVmZXJyYWxBcHAIoJTdHJIYW1XZWJBcHAIoJCJyZWlcnJhbFZpZXciOjTaGFyZURpYWxvZy1MaW5rlwicVmZXJyYWxBcHBQbGF0Zm9ybSI6IldlYilsInJlZmVycmFsTW9kZSI6InZpZXcifX0%3D

6.2.1.5. Services Documentation Evidence for Sprint Review

Durante este Sprint se documentaron los endpoints correspondientes a los siguientes módulos del backend de ParkingNow:

- local
- reserva
- usuario

Cada módulo cuenta con sus respectivos endpoints documentados mediante Swagger. A continuación, se presentarán capturas de la documentación generada que ilustran la estructura, rutas y organización de los servicios expuestos.

Usuario

<http://localhost:3000/usuario>

POST Crear un nuevo usuario



Collection Edit View Window Help

bruno

Collections

Search requests ...

- > contigoNovoy
- > assistent
- < parkingnow
 - GET get
 - POST nuevo local
 - POST Crear un nuevo usuario

parkingnow

Safe Mode No Environment

Collection GET get POST nuevo lo... POST Crear un...

POST http://localhost:3000/usuario

Params Body * Headers Auth * Vars Script

Assert Tests Docs JSON Prettify

Response Headers 7 Timeline Tests

201 Created 57ms 153B

```

1 {
2   "id": 2,
3   "name": "Jesús Lopez",
4   "email": "jesuslz@example.com",
5   "password": "pass123",
6   "placa": "ABC-321",
7   "dni": "72787997"
8 }
```

<http://localhost:3000/usuario>

GET Obtener todos los usuarios

Collection Edit View Window Help

bruno

Collections

Search requests ...

- > contigoNovoy
- > assistent
- < parkingnow
 - GET get
 - POST nuevo local
 - POST Crear un nuevo usuario

parkingnow

Safe Mode No Environment

Collection GET get POST nuevo lo... POST Crear un...

GET http://localhost:3000/usuario

Params Body Headers Auth * Vars Script

Assert Tests Docs

Query

Name	Path
+ Add Param	

Path ?

Name	Value

Response Headers 7 Timeline Tests

200 OK 29ms 304B

```

1 [
2   {
3     "id": 1,
4     "name": "Juan Pérez",
5     "email": "juan@example.com",
6     "password": "123456",
7     "placa": "ABC-123",
8     "dni": "12345678",
9     "ruc": null,
10    "tipoUsuario": "conductor"
11  },
12  {
13    "id": 2,
14    "name": "Jesús Lopez",
15    "email": "jesuslz@example.com",
16    "password": "pass123",
17    "placa": "ABC-321",
18    "dni": "72787997",
19    "ruc": null,
20    "tipoUsuario": "conductor"
21  }
22 ]
```

Activar Wideo
Ve a Configuración

<http://localhost:3000/usuario/1>

get Obtener un usuario por ID

Collection Edit View Window Help

bruno ...

Collections ↑

Q Search requests ...

- > contigoNovoy
- > assistent
- ✓ parkingnow
 - GET get
 - POST nuevo local
 - POST Crear un nuevo usuario

parkingnow

GET http://localhost:3000/usuario/1

Params Body Headers Auth * Vars Script

Assert Tests Docs

Query

Name	Path
+ Add Param	
Path ?	
Name	Value

Response Headers 7 Timeline Tests

200 OK 124ms 148B

```

1  {
2   "id": 1,
3   "name": "Juan Pérez",
4   "email": "juan@example.com",
5   "password": "123456",
6   "placa": "ABC-123",
7   "dni": "12345678",
8   "ruc": null,
9   "tipoUsuario": "conductor"
10 }

```

<http://localhost:3000/usuario/1/locale>

GET Obtiene todos los locales asociados a un usuario específico.

bruno ...

Collections ↑

Q Search requests ...

- > contigoNovoy
- > assistent
- ✓ parkingnow
 - GET get
 - POST nuevo local
 - POST Crear un nuevo usuario

parkingnow

GET http://localhost:3000/usuario/1/locale

Params Body Headers Auth * Vars Script

Assert Tests Docs

Query

Name	Path
+ Add Param	
Path ?	
Name	Value

Response Headers 7 Timeline Tests

200 OK 33ms 101B

```

1  [
2   {
3     "id": 4,
4     "nombre": "Playa Central",
5     "direccion": "Av. Principal 123",
6     "telefono": "987654321",
7     "plazas": 1
8   }
9 ]

```

<http://localhost:3000/usuario/1>

PUT Actualizar un usuario por ID

bruno ...

Collections ↑

Q Search requests ...

- > contigoNovoy
- > assistent
- ✓ parkingnow
 - GET get
 - POST nuevo local
 - POST Crear un nuevo usuario
 - PUT Actualizar un usuario po...

parkingnow

PUT http://localhost:3000/usuario/1

Params Body * Headers Auth * Vars Script

Assert Tests Docs JSON ▾ Pretty

```

1  {
2   "name": "Juan Actualizado",
3   "email": "juan_updated@example.com",
4   "placa": "XYZ-789"
5 }

```

Response Headers 7 Timeline Tests

200 OK 148ms 161B

```

1  {
2   "id": 1,
3   "name": "Juan Actualizado",
4   "email": "juan_updated@example.com",
5   "password": "123456",
6   "placa": "XYZ-789",
7   "dni": "12345678",
8   "ruc": null,
9   "tipoUsuario": "conductor"
10 }

```

<http://localhost:3000/usuario/1>

DELETE Eliminar un usuario por ID

The screenshot shows the Postman interface. On the left, the sidebar lists collections: contigoNovoy, assistent, and parkingnow. Under parkingnow, there are five requests: GET get, POST nuevo local, POST Crear un nuevo usuario, PUT Actualizar un usuario po..., and DEL Eliminar un usuario por ID. The main area shows a DELETE request to <http://localhost:3000/usuario/1>. The Params tab is selected, showing an 'Assert' section with a dropdown menu. The Response tab shows a status of 200 OK with a response body containing the number '1'.

Local

<http://localhost:3000/local>

POST Crear un nuevo local

The screenshot shows the Postman interface. On the left, the sidebar lists collections: contigoNovoy, assistent, and parkingnow. Under parkingnow, there are five requests: GET get, POST Crear nuevo local, POST Crear un nuevo usuario, PUT Actualizar un usuario po..., and DEL Eliminar un usuario por ID. The main area shows a POST request to <http://localhost:3000/local>. The Body tab is selected, showing a JSON payload:

```
1 {  
2   "nombre": "Playa Central"  
3   ,  
4   "direccion": "Av. Principal 123",  
5   "telefono": "987654321",  
6   "usuario": {  
7     "id": 2  
8   }  
9 }  
10 ]
```

The Response tab shows a status of 201 Created with a response body containing a JSON object with fields: id: 5, nombre: "Playa Central", direccion: "Av. Principal 123", telefono: "987654321", plazas: 1, and usuario: { id: 2 }.

<http://localhost:3000/local/5>

GET Obtener un local por ID

Collections

Collection

GET get • POST Crear nu... • POST Crear un... • > +

Search requests ...

> contigoNovoy
> assistent
▼ parkingnow
 GET get
 POST Crear nuevo local
 POST Crear un nuevo usuario
 PUT Actualizar un usuario po...
 DEL Eliminar un usuario por ID

Params Body Headers Auth *

Vars Script Assert Tests Docs

Query

Name	Path
------	------

+ Add Param

Path ?

Name	Value
------	-------

Response Headers 7 Timeline Tests

200 OK 34ms 263B

```
1 {  
2   "id": 5,  
3   "nombre": "Playa Central",  
4   "direccion": "Av. Principal 123"  
5   ,  
6   "telefono": "987654321",  
7   "plazas": 1,  
8   "usuario": {  
9     "id": 2,  
10    "name": "Jesús Lopez",  
11    "email": "jesuslz@example.com"  
12    ,  
13    "password": "pass123",  
14    "placa": "ABC-321",  
15    "dni": "72787997",  
16    "ruc": null,  
17    "tipoUsuario": "conductor"  
18  }  
}
```

http://localhost:3000/local

GET Obtener todos los locales

Collections

Collection

GET get • POST Crear nu... • POST Crear un... • > +

Search requests ...

> contigoNovoy
> assistent
▼ parkingnow
 GET get
 POST Crear nuevo local
 POST Crear un nuevo usuario
 PUT Actualizar un usuario po...
 DEL Eliminar un usuario por ID

Params Body Headers Auth *

Vars Script Assert Tests Docs

Query

Name	Path
------	------

+ Add Param

Path ?

Name	Value
------	-------

Response Headers 7 Timeline Tests

200 OK 36ms 265B

```
1 [  
2   {  
3     "id": 5,  
4     "nombre": "Playa Central",  
5     "direccion": "Av. Principal 1  
23",  
6     "telefono": "987654321",  
7     "plazas": 1,  
8     "usuario": {  
9       "id": 2,  
10      "name": "Jesús Lopez",  
11      "email": "jesuslz@example.c  
om",  
12      "password": "pass123",  
13      "placa": "ABC-321",  
14      "dni": "72787997",  
15      "ruc": null,  
16      "tipoUsuario": "conductor"  
17    }  
18  }
```

http://localhost:3000/local/5

PUT Actualizar un local por ID

bruno

parkingnow

GET get • POST Crear nu... • POST Crear un... • PUT Actualizar... • > +

PUT http://localhost:3000/local/5

Params Body * Headers Auth *

Vars Script Assert Tests Docs

JSON Prettify

```

1 {
2   "nombre": "Playa Actualizada",
3   "direccion": "Av. Nueva 789",
4   "telefono": "987654323"
5 }
```

Response Headers 7 Timeline Tests

200 OK 36ms 263B

```

1 {
2   "id": 5,
3   "nombre": "Playa Actualizada",
4   "direccion": "Av. Nueva 789",
5   "telefono": "987654323",
6   "plazas": 1,
7   "usuario": {
8     "id": 2,
9     "name": "Jesús Lopez",
10    "email": "jesuslz@example.com"
11   },
12   "password": "pass123",
13   "placa": "ABC-321",
14   "dni": "72787997",
15   "ruc": null,
16   "tipoUsuario": "conductor"
17 }
```

<http://localhost:3000/local/5>

DELETE Eliminar un local por ID

Collections

Search requests ...

> contigoNovoy

> assistent

< parkingnow

GET get • POST Crear nuevo local • POST Crear un nuevo usuario • PUT Actualizar por ID • DELETE Eliminar por ID

DELETE http://localhost:3000/local/5

Params Body Headers Auth *

Vars Script Assert Tests Docs

Query

Name	Path
+ Add Param	

Path ?

Name	Value

Response Headers 5 Timeline Tests

200 OK 30ms 0B

1

<http://localhost:3000/local/6/usuario>

GET Obtener el usuario dueño de un local

bruno ...

Collections ↑

Search requests ...

- > contigoNovoy
- > assistent
- ✓ parkingnow
 - GET get
 - POST Crear nuevo local
 - POST Crear un nuevo usuario
 - PUT Actualizar por ID
 - DEL Eliminar por ID

parkingnow

Safe Mode ⚡ ⏱ ⚡ ⏱ ⏵ ⏵ ⏵ ⏵ ⏵ ⏵ No Environment ▾

GET http://localhost:3000/local/6/usuario

Params Headers Auth *
Vars Script Assert Tests Docs

Query

Name	Path
+ Add Param	

Path ⓘ

Name	Value
+ Add Path	

Response Headers 7 Timeline Tests

200 OK 4ms 153B

```

1 {
2   "id": 2,
3   "name": "Jesús Lopez",
4   "email": "jesuslz@example.com",
5   "password": "pass123",
6   "placa": "ABC-321",
7   "dni": "72787997",
8   "ruc": null,
9   "tipoUsuario": "conductor"
10 }

```

Reserva

<http://localhost:3000/reserva>

POST Crear una nueva reserva

bruno ...

Collections ↑

Search requests ...

- > contigoNovoy
- > assistent
- ✓ parkingnow
 - GET get
 - POST Crear nueva reserva
 - POST Crear un nuevo usuario
 - PUT Actualizar por ID
 - DEL Eliminar por ID

parkingnow

Safe Mode ⚡ ⏱ ⚡ ⏱ ⏵ ⏵ ⏵ ⏵ ⏵ ⏵ No Environment ▾

POST http://localhost:3000/reserva

Params Body * Headers Auth * Vars
Script Assert Tests Docs

JSON Prettify

```

1 {
2   "usuario": { "id": 2 },
3   "local": { "id": 6 },
4   "nro_plaza": 5,
5   "fh_inicio": "2025-05-16T08:00:00Z",
6   "fh_fin": "2025-05-16T10:00:00Z"
7 }

```

Response Headers 7 Timeline Tests

201 Created 42ms 125B

```

1 {
2   "id": 1,
3   "usuario": {
4     "id": 2
5   },
6   "local": {
7     "id": 6
8   },
9   "nro_plaza": 5,
10  "fh_inicio": "2025-05-16T08:00:00Z",
11  "fh_fin": "2025-05-16T10:00:00Z"
12 }

```

<http://localhost:3000/reserva>

GET Obtener todas las reservas

The screenshot shows the Postman interface with a search bar at the top left. Below it, a sidebar lists collections: 'contigoNovoy', 'assistent', and 'parkingnow'. Under 'parkingnow', there are five methods: 'GET get' (selected), 'POST Crear nueva reserva', 'POST Crear un nuevo usuario', 'PUT Actualizar por ID', and 'DEL Eliminar por ID'. The main area shows a GET request to 'http://localhost:3000/reserva'. The 'Params' tab is selected, showing a table with columns 'Name' and 'Path'. A row for 'Name' is present but empty. The 'Path' section also has a table with columns 'Name' and 'Value', which is currently empty. The 'Response' tab is selected, displaying the JSON response from the server. The response is a JSON array with two elements, each representing a parking reservation:

```
[  
  {  
    "id": 1,  
    "usuario": {  
      "id": 2,  
      "name": "Jesús Lopez",  
      "email": "jesuslz@example.com",  
      "password": "pass123",  
      "placa": "ABC-321",  
      "dni": "72787997",  
      "ruc": null,  
      "tipoUsuario": "conductor"  
    },  
    "local": {  
      "id": 6,  
      "nombre": "Playa Central",  
      "direccion": "Av. Principal 123",  
      "telefono": "987654321",  
      "plazas": 1  
    },  
    "nro_plaza": 5,  
    "fh_inicio": "2025-05-16T08:00:00.000Z",  
    "fh_fin": "2025-05-16T10:00:00.000Z"  
  }]  
]
```

<http://localhost:3000/reserva/1>

GET Obtener una reserva por ID

The screenshot shows the Postman application interface. At the top, there's a header with the user icon "bruno", a "Safe Mode" button, and a "No Environment" dropdown. Below the header, the left sidebar lists collections: "contigoNovoy", "assistent", and "parkingnow". Under "parkingnow", several requests are listed: "GET get", "POST Crear nueva reserva", "POST Crear un nuevo usuario", "PUT Actualizar por ID", and "DEL Eliminar por ID". The main area shows a "GET" request to "http://localhost:3000/reserva/1". The "Params" tab is selected, showing a table with columns "Name" and "Path". There is one row with the value "1" in the Path column. The "Response" tab is selected, showing a JSON response with line numbers:

```
1 {  
2   "id": 1,  
3   "usuario": {  
4     "id": 2,  
5     "name": "Jesús Lopez",  
6     "email": "jesuslz@example.com",  
7     "password": "pass123",  
8     "placa": "ABC-321",  
9     "dni": "72787997",  
10    "ruc": null,  
11    "tipoUsuario": "conductor"  
12  },  
13  "local": {  
14    "id": 6,  
15    "nombre": "Playa Central",  
16    "direccion": "Av. Principal 123",  
17    "telefono": "987654321",  
18    "plazas": 1  
19  },  
20  "nro_plaza": 5,  
21  "fh_inicio": "2025-05-16T08:00:00.000Z",  
22  "fh_fin": "2025-05-16T10:00:00.000Z"  
23 }
```

<http://localhost:3000/reserva/1>

PUT Actualizar una reserva por ID

PUT <http://localhost:3000/reserva/1>

Params Body * Headers Auth * Vars Script

Assert Tests Docs **JSON** Prettify

```
1 {  
2   "nro_plaza": 10,  
3   "fh_fin": "2025-05-16T12:00:00Z"  
4 }
```

Response Headers 7 Timeline Tests

200 OK 31ms 366E

```
1 {  
2   "id": 1,  
3   "usuario": {  
4     "id": 2,  
5     "name": "Jesús Lopez",  
6     "email": "jesuslz@example.com",  
7     "password": "pass123",  
8     "placa": "ABC-321",  
9     "dni": "72787997",  
10    "ruc": null,  
11    "tipoUsuario": "conductor"  
12  },  
13  "local": {  
14    "id": 6,  
15    "nombre": "Playa Central",  
16    "direccion": "Av. Principal 123",  
17    "telefono": "987654321",  
18    "plazas": 1  
19  },  
20  "nro_plaza": 10,  
21  "fh_inicio": "2025-05-16T08:00:00.000Z",  
22  "fh_fin": "2025-05-16T12:00:00Z"  
23 }
```

<http://localhost:3000/reserva/1>

DELETE Eliminar una reserva por ID

bruno ... parkingnow Safe Mode No Environment

Collections GET get POST Crear nu... POST Crear un... PUT Actualizar... DELETE Elimin...

Search requests ...

> contigoNovoy
> assistent
< parkingnow
 GET get
 POST Crear nueva reserva
 POST Crear un nuevo usuario
 PUT Actualizar por ID
 DEL Eliminar por ID

DELETE <http://localhost:3000/reserva/1>

Params Body Headers Auth * Vars Script

Assert Tests Docs

Query

Name	Path
------	------

+ Add Param

Path ?

Name	Value
------	-------

Response Headers 5 Timeline Tests

200 OK 133ms 0B

```
1
```

6.2.1.6. Software Deployment Evidence for Sprint Review

Durante este Sprint se realizaron los despliegues correspondientes de los tres productos principales del proyecto: la Landing Page, los Web Services y la Aplicación Móvil. A continuación, se presentarán capturas de cada uno de ellos como evidencia del correcto funcionamiento y despliegue:

- **Landing Page:** desplegada mediante GitHub Pages.

Link del landing page desplegada : <https://g3-upc-cc238-346-parkingnow.github.io/New-Landing-Page/>

The screenshot shows the GitHub repository interface for 'G3-UPC-CC238-346-ParkingNow'. The left sidebar has sections for 'Code', 'Issues', 'Pull requests', 'Actions', 'Projects', 'Wiki', 'Security', 'Insights', and 'Settings'. Under 'Deployments', it shows 'All deployments' and 'Environments'. The 'github-pages' environment is selected. Below that is a 'Manage environments' button. The main area is titled 'github-pages deployments' and shows 'Latest deployments' for 'github-pages'. It lists a deployment from 3 days ago and a recent one. Below this is a table for '10 deployments' with columns for status, commit, date, branch, and more.

Ciudades donde puedes encontrar PARKINGNOW



Lima

La vibrante capital del Perú, conocida por su rica historia, arquitectura colonial y vida nocturna.



Arequipa

Famosa por sus edificios blancos de sillería blanca y la majestuosidad del volcán Misti que la rodea.



Cusco

La antigua capital del Imperio Inca, famosa por su historia y proximidad a Machu Picchu.

- **Web Services (Backend):** ejecutado localmente para documentar los endpoints.

The screenshot shows the Postman application and a side-by-side view of the VS Code code editor. The Postman interface shows a collection named 'parkingnow' with various API endpoints like 'GET contigoNovoy', 'POST Crear nuevo usuario', etc. A specific POST request to 'http://localhost:3000/reserva' is selected, showing a JSON response with a created local object. The VS Code editor shows the 'app.module.ts' file with Nest.js code for a 'LocalService' constructor and methods like 'findAll', 'findOne', and 'create'. The 'src' folder structure is visible on the right.

- **Aplicación Móvil:** ejecutada en emulador o dispositivo real, mostrando su pantalla de inicio.

The screenshot shows the Android Studio interface with the NavGraph.kt file open in the code editor. The code defines a navigation graph for the ParkingNow app, mapping routes to specific screens like HomeScreen, LoginScreen, and ForgotPasswordScreen. To the right, a preview of the app's landing page is displayed on a Pixel 7 Pro API 36 device. The landing page features a car icon with a parking sign, the text "Tu estacionamiento ideal, más cerca que nunca.", and a large yellow "Empezar" button.

```

1 package pe.upc.parkingnow.ui
2
3 import androidx.compose.runtime.Composable
4 import androidx.navigation.NavHostController
5 import androidx.navigation.compose.NavHost
6 import androidx.navigation.compose.composable
7 import pe.upc.parkingnow.presentation.home.HomeScreen
8 import pe.upc.parkingnow.presentation.login.LoginScreen
9 import pe.upc.parkingnow.presentation.forgotpassword.ForgotPasswordScreen
10 import pe.upc.parkingnow.presentation.resetpassword.ResetPasswordScreen
11 import pe.upc.parkingnow.presentation.SelectRole.SelectRoleScreen
12 import pe.upc.parkingnow.presentation.SelectRole.Role
13 import pe.upc.parkingnow.presentation.login.RegisterDriver.RegisterDriverScreen
14 import pe.upc.parkingnow.presentation.login.RegisterOwner.RegisterOwnerScreen
15 import pe.upc.parkingnow.presentation.dashboarddriver.DashboardDriverScreen
16 import pe.upc.parkingnow.presentation.dashboardowner.DashboardOwnerScreen
17
18 @Composable
19 fun AppNavGraph(navController: NavHostController) {
20     NavHost(
21         navController = navController,
22         startDestination = "home"
23     ) {
24         // Home
25         composable(route: "home") {
26             HomeScreen(
27                 onStartClick = { navController.navigate(route: "Login") },
28                 onRegisterClick = { navController.navigate(route: "selectRole") }
29             )
30         }
31         // Login
32         composable(route: "login") {
33             LoginScreen(
34                 onLoginSuccess = { /* Javega donde corresponde después de login */ },
35                 onForgotPassword = { navController.navigate(route: "Forgot") },
36                 onRegisterClick = { navController.navigate(route: "selectRole") }
37             )
38         }
39         // Forgot Password
40         composable(route: "forgot") {
41             ForgotPasswordScreen()
        }
    }
}

```

6.2.1.7. Team Collaboration Insights during Sprint

Durante este Sprint, el equipo trabajó de manera colaborativa para lograr el desarrollo completo de los tres productos digitales: la Landing Page, el aplicativo móvil y los Web Services. A continuación, se detallan los aportes por cada sección, incluyendo espacio para evidencias de commits y capturas relevantes.

Landing Page

El equipo frontend trabajó en la implementación de todas las secciones de la página, utilizando HTML, CSS, JavaScript y Angular. Las tareas se asignaron equitativamente y se registraron los avances mediante commits individuales en GitHub.

Capturas y Commits (Landing Page):

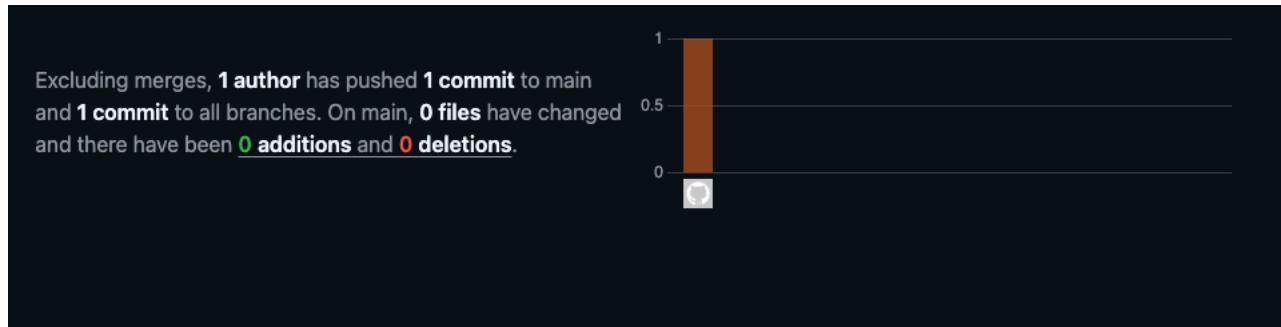


Aplicación Móvil

La app se desarrolló en dos partes: una versión nativa con Kotlin + Jetpack Compose y una estructura multiplataforma con Flutter. Se implementaron pantallas funcionales como login, registro, selección de rol y dashboard.

Capturas y Commits (Mobile App):

Kotlin



Flutter



🔧 Web Services (Backend)

El equipo backend estructuró módulos como `usuario`, `reserva` y `local`, desarrollando los endpoints principales. La documentación se realizó con OpenAPI y se versionó correctamente en GitHub.

Capturas y Commits (Backend):



📝 Análisis de colaboración - Reporte

Cada integrante participó activamente en su área asignada. Se revisaron los analíticos de GitHub para verificar contribuciones equitativas y se adjuntan capturas que evidencian la distribución del trabajo en el repositorio grupal.

Capturas de actividad en GitHub:



6.2.2. Sprint 2

En este sprint se desarrollaron las **US18 a US33**, enfocadas principalmente en funcionalidades clave del **segmento conductor**, como la visualización de reservas, historial, pagos, notificaciones, mapa, seguimiento en tiempo real y soporte desde el dashboard.

Paralelamente, se avanzó en la construcción del segmento **dueño de estacionamiento** con la estructura base del dashboard, validación de espacios y consultas iniciales sobre la disponibilidad, con enfoque en el diseño de vistas y navegación entre funcionalidades.

Estas implementaciones permitieron establecer una estructura sólida para los siguientes módulos operativos de la plataforma ParkingNow.

6.2.2.1. Sprint Planning 2

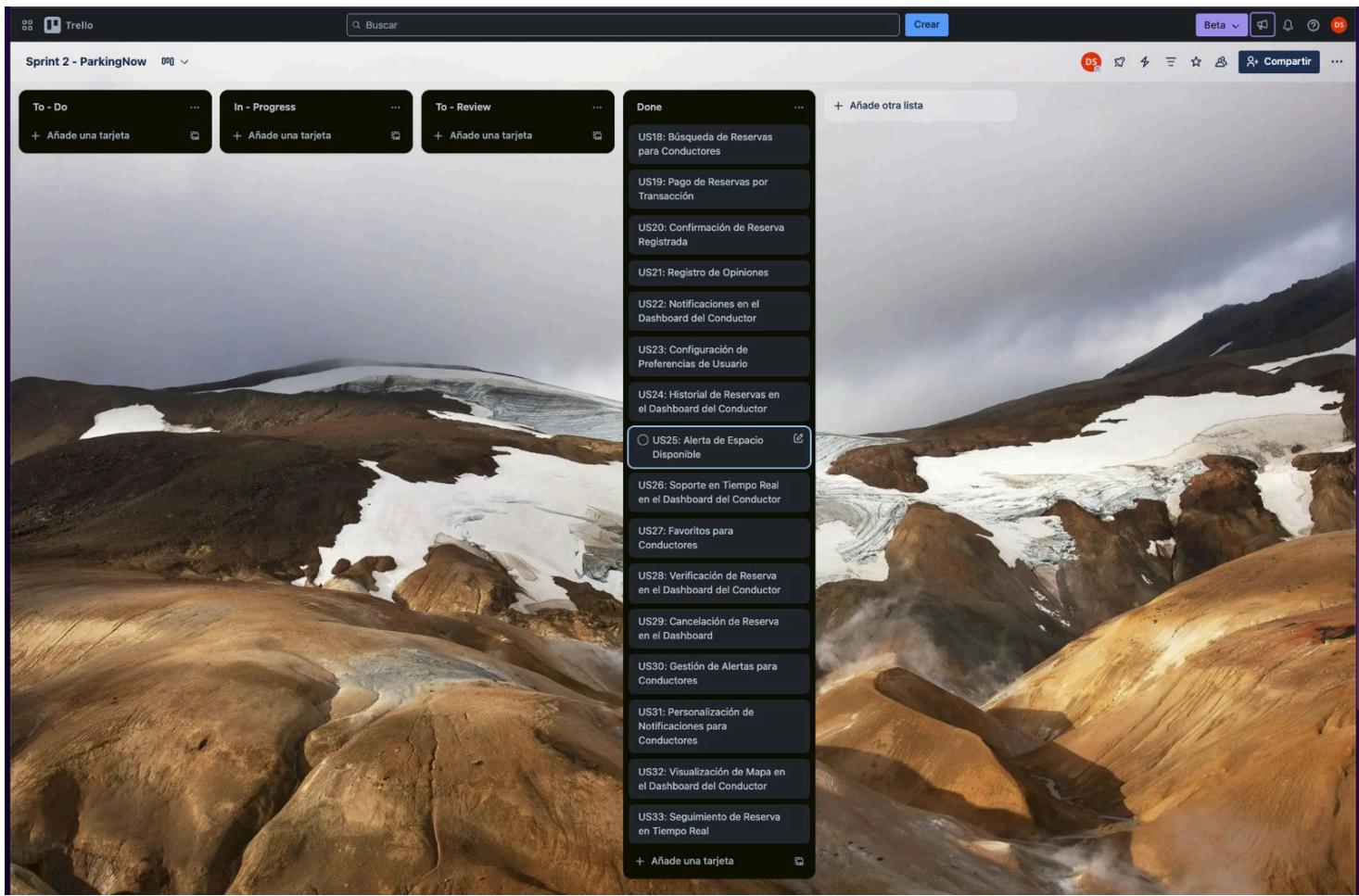
Sprint #	Sprint 2
Date	2025-06-03
Time	17:00 horas (GMT-5)
Location	Modalidad remota a través de Discord
Prepared By	Soto Quispe, Diego Ulises
Attendees	Calisaya Sánchez, Juan Jesús Hidalgo Lopez, Mathias Adriano Molina Asencios, Samuel Elias Soto Quispe, Diego Ulises
Sprint 1 Review Summary	En el Sprint 1 se completaron las funcionalidades de la landing page y el flujo base de autenticación para conductores y dueños. Se cumplieron los objetivos planificados en cuanto a diseño visual, experiencia de usuario y conectividad básica con backend.
Sprint 1 Retrospective	Se identificó la necesidad de mejorar la distribución de tareas y realizar validaciones más frecuentes en backend. Se acordó usar pruebas unitarias más temprano en el desarrollo.

Sprint #	Sprint 2
Summary	
Sprint 2 Goal	Desarrollar funcionalidades clave del módulo conductor y del módulo dueño: reservas, pagos, tickets, seguimiento, configuración, notificaciones y mejoras de perfil.
Sprint 2 Velocity	120
Sum of Story Points	112

6.2.2.2. Sprint Backlog 2

User Story	Title	Work-Item ID	Work-Item Title	Description	Estimation (Hours)	Assigned To	Status
US18	Búsqueda de Reservas	WT18	Ver reservas activas y pasadas	Mostrar historial de reservas del conductor	6	Samuel Molina	Done
US19	Pago de Reservas	WT19	Integrar pasarela de pago	Flujo completo de pago móvil	6	Juan Calisaya	Done
US20	Confirmación de Reserva	WT20	Pantalla de éxito	Indicar que la reserva se registró correctamente	4	Diego Soto	Done
US21	Registro de Opiniones	WT21	Enviar opiniones	Permitir enviar feedback luego de una reserva	4	Mathias Hidalgo	Done
US22	Notificaciones en Dashboard	WT22	Alertas y notificaciones	Mostrar notificaciones al usuario	5	Samuel Molina	Done
US23	Configuración de Preferencias	WT23	Ajustes de usuario	Cambiar idioma, tema y notificaciones	5	Diego Soto	Done
US24	Historial de Reservas	WT24	Visualizar historial completo	Mostrar reservas anteriores por fechas	6	Juan Calisaya	Done
US25	Alerta de Espacio Disponible	WT25	Activar alertas	Notificar disponibilidad de espacios	6	Mathias Hidalgo	Done
US26	Soporte en Tiempo Real	WT26	Sistema de soporte	Chat o formulario de contacto en dashboard	5	Samuel Molina	Done
US27	Favoritos para Conductores	WT27	Marcar como favorito	Guardar y listar estacionamientos favoritos	5	Diego Soto	Done
US28	Verificación de Reserva	WT28	Estado de la reserva	Mostrar si la reserva está confirmada o en proceso	4	Juan Calisaya	Done
US29	Cancelación de Reserva	WT29	Cancelar reservas	Permitir anular una reserva activa	5	Mathias Hidalgo	Done
US30	Gestión de Alertas	WT30	Ver y administrar alertas	Mostrar y gestionar alertas en el dashboard	4	Samuel Molina	Done
US31	Personalización de Notificaciones	WT31	Ajustar notificaciones	Activar/desactivar tipos de alertas	4	Diego Soto	Done
US32	Visualización de Mapa	WT32	Mapa interactivo	Mostrar mapa con espacios disponibles	6	Juan Calisaya	Done
US33	Seguimiento en Tiempo Real	WT33	Tracking de reserva	Ver estado actualizado de la reserva	5	Mathias Hidalgo	Done

Trello: <https://trello.com/invite/b/6851bac9c06fde8f5e2e88f5/ATTI4f480fdeeccada7d967d82c1e5eb20578F4C1A5D/sprint-2-parkingnow>



6.2.2.3. Development Evidence for Sprint Review

Durante el segundo sprint se implementaron funcionalidades clave para el segmento conductor, como el flujo completo de reservas, pagos, notificaciones, visualización de mapa, seguimiento en tiempo real y ajustes personalizados. Asimismo, se continuó con la implementación del segmento dueño en Flutter, construyendo vistas base e integración visual.

A continuación, se presenta la evidencia técnica mediante commits asociados a las historias US18 a US33:

◆ Kotlin – Segmento Objetivo: Conductor

ID Commit ID	Repository	Rama	Autor	Fecha	Mensaje de Commit
7a88dbc	Kotlin-Mobile	main	mathiasadriano	20/06/2025	Update colors.xml
4152f23	Kotlin-Mobile	main	mathiasadriano	20/06/2025	Update AndroidManifest.xml
cc2e16f	Kotlin-Mobile	main	mathiasadriano	20/06/2025	edit gradle.properties
055bb79	Kotlin-Mobile	main	mathiasadriano	20/06/2025	Update README.md
e398f27	Kotlin-Mobile	main	mathiasadriano	20/06/2025	feat:edit kotlin
ccbe57a	Kotlin-Mobile	main	JuanCali999	18/06/2025	feat: implement user registration flow with Retrofit and update API structure
db38a28	Kotlin-Mobile	main	JuanCali999	18/06/2025	feat: configure network security and update API base URL for emulator
f362356	Kotlin-Mobile	main	JuanCali999	18/06/2025	feat: implement user registration API integration
49de69c	Kotlin-Mobile	main	JuanCali999	18/06/2025	feat: add Retrofit and Gson converter dependencies

ID Commit ID	Folder Repository	Plantilla Rama	User Autor	Date Fecha	Message Mensaje de Commit
9e9c8ad	Kotlin-Mobile	main	JuanCali999	18/06/2025	Merge branch 'main' of https://github.com/G3-UPC-CC238-346-ParkingNow/Kotlin-Mobile into feature/consume-user-api
0844608	Kotlin-Mobile	main	Diego Soto	11/06/2025	Add GPS activation prompt to DashboardScreen
6b984a	Kotlin-Mobile	main	Diego Soto	11/06/2025	Add debug APK for testing
2d5deb4	Kotlin-Mobile	main	Diego Soto	09/06/2025	update app icon alignment
00dc823	Kotlin-Mobile	main	Diego Soto	09/06/2025	Adjust FAB position on DashboardScreen for better alignment with top bar
7a26d28	Kotlin-Mobile	main	Diego Soto	04/06/2025	Update LoginScreen: remove social login and enable guest mode navigation
ac81138	Kotlin-Mobile	main	Diego Soto	04/06/2025	polish UI for password update and validation flow
d61546e	Kotlin-Mobile	main	Diego Soto	30/05/2025	Add custom app icon for ParkingNow
c1b1114	Kotlin-Mobile	main	Diego Soto	30/05/2025	Fix: Apply global dark mode using AppViewModel
da42c27	Kotlin-Mobile	main	Diego Soto	30/05/2025	Fix bottom navigation layout on Tracking screen to match Support screen style
4db18aa	Kotlin-Mobile	main	Diego Soto	29/05/2025	Fix TicketScreen background and remove unused imports
2e2a736	Kotlin-Mobile	main	Diego Soto	29/05/2025	Update: improved booking screen layout and visibility
4c88466	Kotlin-Mobile	main	Diego Soto	29/05/2025	Refactor UI: adjusted layout spacing for favorites and reservation button visibility
025323e	Kotlin-Mobile	main	Diego Soto	29/05/2025	Fix navigation: ensure safe routes and prevent crashes on back navigation
e84797c	Kotlin-Mobile	main	Diego Soto	29/05/2025	Show real-time user location with car icon in Dashboard map
d3231a4	Kotlin-Mobile	main	Diego Soto	28/05/2025	Enhance NotificationsScreen UI with back navigation and layout adjustments
3266b1c	Kotlin-Mobile	main	Diego Soto	28/05/2025	Feature: Improved ticket UI and payment method handling
abfdc25	Kotlin-Mobile	main	Diego Soto	27/05/2025	Smooth welcome transition with fade and scale animation
b403f37	Kotlin-Mobile	main	Diego Soto	26/05/2025	Adjust map position at bottom of dashboard
2ffcc0b	Kotlin-Mobile	main	Diego Soto	26/05/2025	Show registered username in Dashboard
90d538d	Kotlin-Mobile	main	Diego Soto	26/05/2025	Add NotificationsScreen and complete all routing
fa34cc2	Kotlin-Mobile	main	Diego Soto	26/05/2025	Add full payment flow and ticket generation screen
e683b14	Kotlin-Mobile	main	Diego Soto	25/05/2025	Add real-time user location to map in Dashboard

Flutter – Segmento Objetivo: Dueño de Estacionamiento

ID Commit ID	Folder Repository	Plantilla Rama	User Autor	Date Fecha	Message Mensaje de Commit
7c950c5	Flutter	feature/register-parking	mathiasadriano	20/06/2025	feat: edit api_constants
9fde524	Flutter	feature/register-parking	JuanCali999	20/06/2025	feat: implement registration functionality for parking owners and locals
2efa9fe	Flutter	main	Diego Soto	11/06/2025	Add APK
b694f82	Flutter	main	Diego Soto	09/06/2025	update launcher icon configuration and generate icons

ID	Commit ID	Repository	Rama	Autor	Fecha	Mensaje de Commit
c42bece	Flutter	main		Diego Soto	09/06/2025	Fix: default theme mode now respects system setting
b4aa3de	Flutter	main		Diego Soto	09/06/2025	Fix: full dark mode applied in Security UI
7920a4b	Flutter	main		Diego Soto	08/06/2025	Clean up alert cards and remove subscription dialog
22524bb	Flutter	main		Diego Soto	08/06/2025	Fix: remove duplicate ChangePasswordPage from settings
19ee43a	Flutter	main		Diego Soto	08/06/2025	Fix: initialize locale for date formatting in history
74a9c3e	Flutter	main		Diego Soto	08/06/2025	remove settings from quick actions
1493af1	Flutter	main		Diego Soto	08/06/2025	Improve reservation popup UI
d3e5d2f	Flutter	main		Diego Soto	07/06/2025	Improve step 4 summary card and success pop-up design
ec090f0	Flutter	main		Diego Soto	07/06/2025	Improve drawer header spacing and fix overflow
eaddcd1	Flutter	main		Diego Soto	07/06/2025	Fix: remove tab transition animation
001d355	Flutter	main		Diego Soto	07/06/2025	update color styles
92aabc2	Flutter	main		Diego Soto	07/06/2025	remove fade animation from login
f496ec8	Flutter	main		Diego Soto	07/06/2025	update
83599e0	Flutter	main		Diego Soto	07/06/2025	Fix UI bugs: remove missing image and update section titles
5c9f139	Flutter	main		Diego Soto	06/06/2025	Fix: restrict RUC input to numbers only
09b465b	Flutter	main		Diego Soto	05/06/2025	Feature: first commit from external drive

Backend – Web Services

ID	Commit ID	Repository	Rama	Autor	Fecha	Mensaje de Commit
f83243a	Backend	main		samuelmolina201016	17/06/2025	fix adding new endpoint for manage parking available
5392d32	Backend	main		samuelmolina201016	17/06/2025	fix adding user manage guide
88b445a	Backend	main		samuelmolina201016	17/06/2025	fix adding API for get all reserves of a Local - manage parking plance
e483b6e	Backend	main		samuelmolina201016	17/06/2025	fix Adding Api's guide
33919ff	Backend	main		samuelmolina201016	16/05/2025	feat(documentation) Documentation for endpoints
38224ffe	Backend	main		samuelmolina201016	16/05/2025	feat(usuario) Adding API for Usuario Managment
a171b92	Backend	main		samuelmolina201016	16/05/2025	feat(reserva) Adding API for reserva managment
a2b3c38	Backend	main		samuelmolina201016	16/05/2025	feat(local) Adding API for Local managment
d4233bd	Backend	main		samuelmolina201016	16/05/2025	feat(app) Adding Services and Controllers
f8b795f	Backend	main		samuelmolina201016	16/05/2025	fix(core) Adding dependencies
f5976b8	Backend	main		JuanCali999	14/05/2025	feat: Initialize NestJS project with default structure
a218916	Backend	main		Diego2005z	10/05/2025	Initial commit

Landing Page

ID	Commit ID	Repository	Rama	Autor	Fecha	Mensaje de Commit
	1594898	Landing-ParkingNow	main	Diego2005z	20/06/2025	Fix: embed YouTube videos
	8d4a6be	Landing-ParkingNow	main	Diego2005z	19/06/2025	Fix i18n translation issues and update HTML entities
	82d13ed	Landing-ParkingNow	main	Diego2005z	18/06/2025	Add landing page with updated README and styles

6.2.2.4. Testing Suite Evidence for Sprint Review

Repository	Branch	Author	Message	Date	Link
Backend	main	Samuel Molina	feat(documentation) Documentation for endpoints	16/05/2025	 https://github.com/G3-UPC-CC238-346-ParkingNow/Backend

6.2.2.5. Execution Evidence for Sprint Review

➡ **Segmento Objetivo: Conductor – Kotlin, Jetpack Compose y Android Studio**

➡  **Pantalla de Inicio de Sesión – Login del Conductor**



ParkingNow

Encuentra el mejor estacionamiento



¡Bienvenido de vuelta!

Inicia sesión en tu cuenta



Correo electrónico



Contraseña



?

¿Olvidaste tu contraseña?

→ Iniciar Sesión

0



Pantalla de Recuperación – ¿Olvidaste tu Contraseña?



¿Olvidaste tu contraseña?

Ingresa tu correo y te enviaremos un enlace para restablecerla



Correo electrónico

Enviar enlace de recuperación

[Volver al inicio de sesión](#)



 **Pantalla de Cambio de Contraseña – Validación y Seguridad**



Cambiar Contraseña

Crea una nueva contraseña segura
para proteger tu cuenta



Nueva
contraseña



Confirmar
contraseña



Requisitos de contraseña:

- ✖ Al menos 8 caracteres
- ✖ Al menos una letra mayúscula

 **Registro de Nuevo Usuario – Creación de Cuenta para Conductores**

← Crear Cuenta



¡Únete a ParkingNow!

Crea tu cuenta y encuentra estacionamiento fácilmente



Información Personal



Nombre completo



Correo electrónico



Seguridad



Contraseña



Confirmar
contraseña



 **Dashboard del Conductor – Vista General de Actividades**

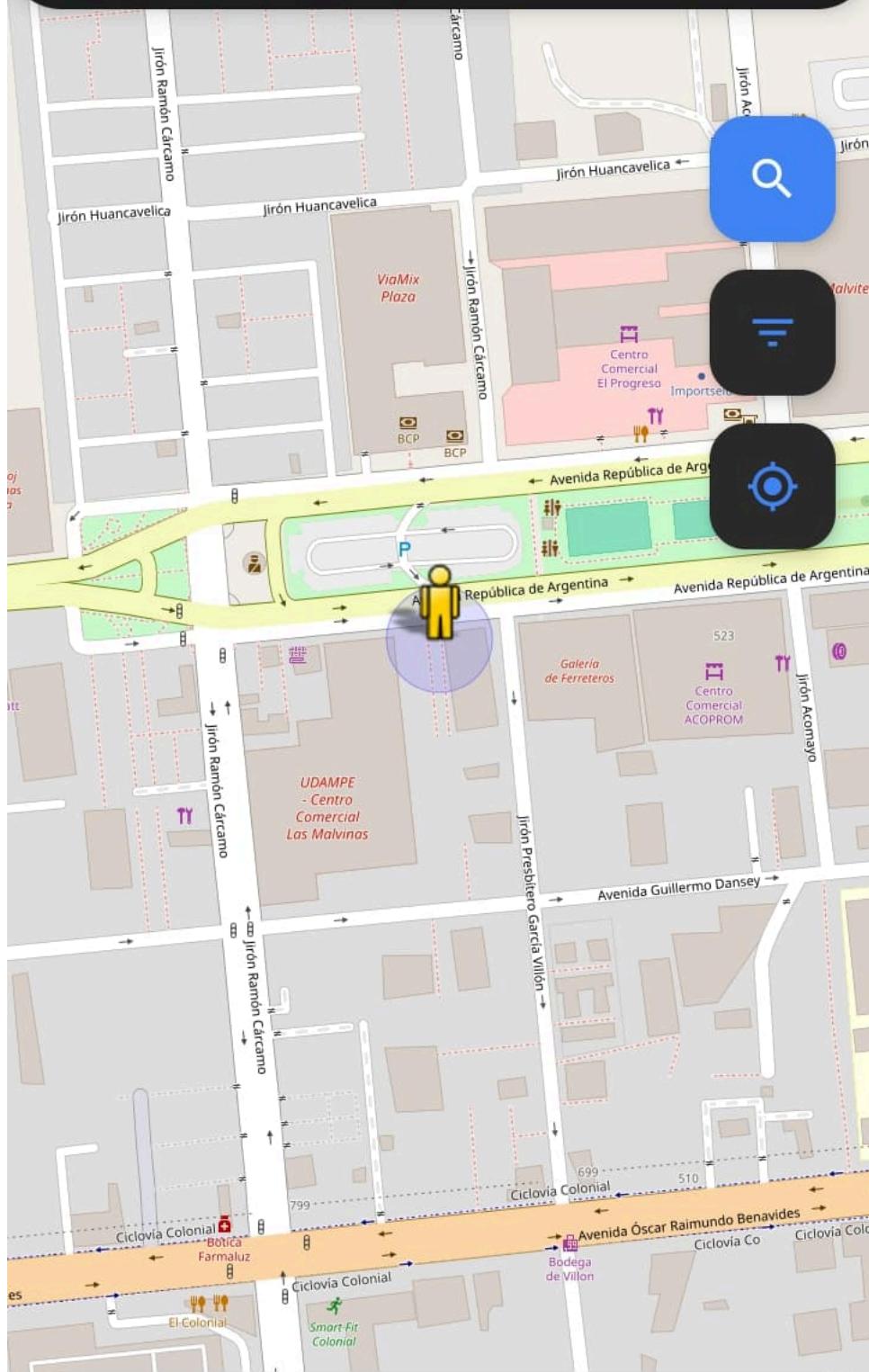
1:09 Ct • VoLTE 56%

Dashboard Conductor



D

Lima



Fotocionamientos Cerecinos

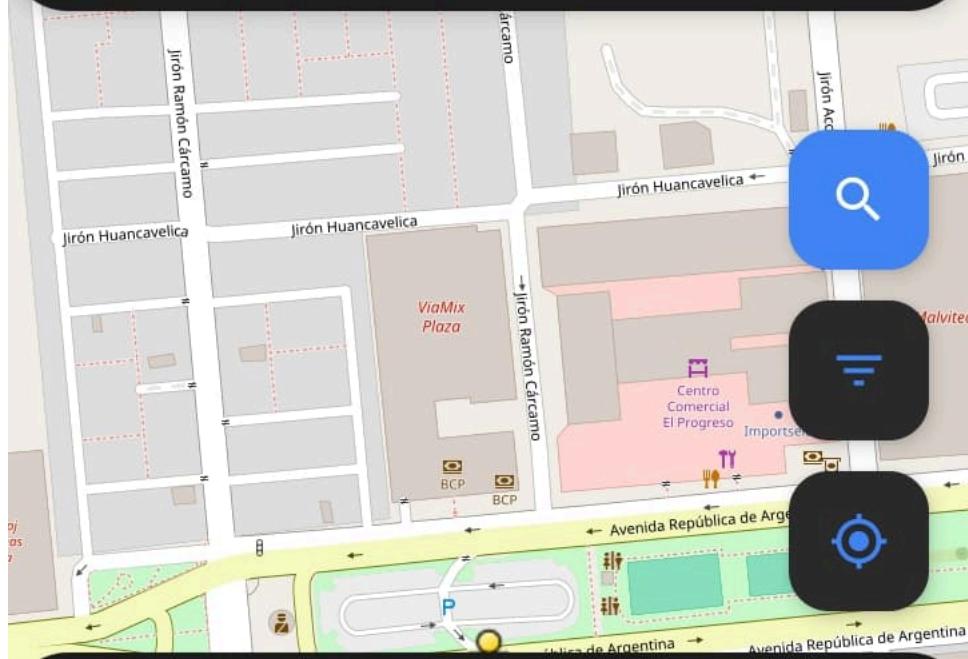
1:10 Ct • VoLTE 56%

Dashboard Conductor



D

Lima



Estacionamientos Cercanos

3 lugares disponibles

Todos

Más cercano

Más barato

Mejor valorado

Centro Comercial Jockey S/ 5.00/hora

Plaza

★ 4.5

Av. Javier Prado Este 4200,

Santiago de Surco

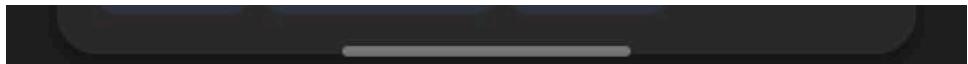
45/200 disponibles

0.8 km

Techado

Seguridad 24h

Cámaras



 Barra Lateral – Navegación con Opciones del Módulo Conductor

1:10 Ct •

VoLTE 56%

D

Hola,

**Diego Ulises Soto
Quispe**

Lima, Provincia de Lima

 **Inicio**

 **Reservas**

 **Soporte**

 **Seguimiento**

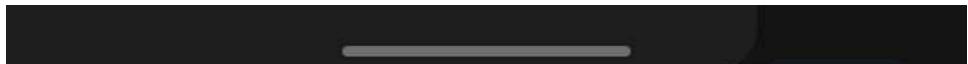
 **Configuración**

 **Notificación**

 **Cerrar Sesión**

D





 **Historial de Reservas – Visualización de Reservas Activas y Pasadas**

← Mis Reservas



Reservas Activas



Gestiona tus reservas actuales

[Ver reservas activas](#)

Nueva Reserva



Encuentra y reserva un
estacionamiento

[Realizar reserva](#)

Historial



Revisa tus reservas anteriores

[Ver historial](#)

Lugares Favoritos



Inicio



Reservas



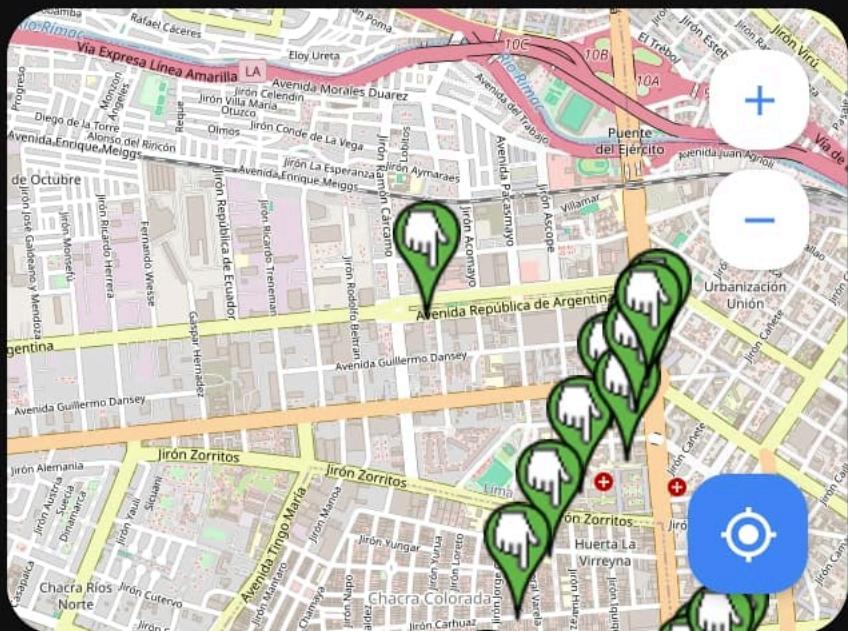
Ajustes



 Reservar Estacionamiento – Selección de Espacio Cercano Según Ubicación

← Seleccionar Estacionamiento

Ubicación



Buscar estacionamiento



Estacionamientos Cercanos Ver todos

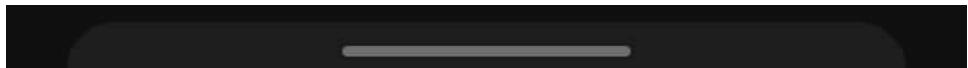
Real Plaza Salaverry

★ 4.5

📍 0.8 km

S/ 5.00 / hora P 12 lugares disponibles

Seleccionado ✓



 Método de Pago – Selección de Tarjeta y Confirmación de Transacción

← Métodos de Pago Guardados

No hay métodos guardados.

Pagar con Nuevo Método

Visa

Yape

Card Number



Card Holder Name



Expiry Date (MM/YY)



CVV

Guardar

Pagar con este método

Seleccione un método para pagar.

Código QR – Generación de Código para Validar la Reserva

1:10 Ct •

VoLTE 56%

Parking Larcomar

Malecón de la Reserva, Miraflores

VEHICLE

2021 Audi Q3 • B 1234 CD

DURATION

1 hora • 26 May. 2025

Slot A01



ID: PN-843043

Navegar

 Confirmación de Reserva – Mensaje de Éxito tras la Transacción

1:10 Ct •

57%



Success!

Yey, 1 parking slot
already booked for you.



 Soporte Técnico – Envío de Consultas y Comentarios desde el Dashboard

← Centro de Soporte



¡Estamos aquí para ayudarte!



Describe tu problema o envíanos tus sugerencias

Ayuda Rápida



FAQ



Chat

Reportar un Problema



Describe detalladamente el problema que estás experimentando

Describe tu problema aquí...



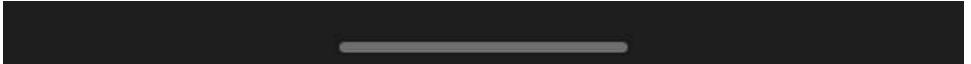
Inicio



Soporte



Ajustes



 **Seguimiento del Vehículo – Tracking en Tiempo Real del Espacio Reservado**

← Seguimiento



Estado del Sistema

Monitoreo activo



2

Vehículos



4

Cámaras



0

Alertas

Vehículos Registrados

Toyota Corolla

Placa: ABC-123



Real Plaza Salaverry -

Estacionado

Nivel 2

Hace 5 minutos

Honda Civic

Placa: XYZ-789



Av. Javier Prado Este

En tránsito

Hace 2 minutos



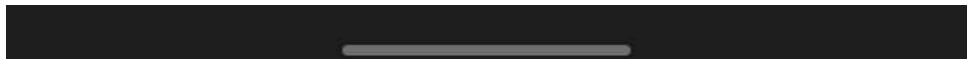
Inicio



Seguimiento



Ajustes



Configuración del Sistema – Preferencias de Usuario y Notificaciones

← Configuración



ParkingNow

Versión 1.0.0

Cuenta y Seguridad



Privacidad y políticas



Pedir reembolso



Registrar robo de carro



Eliminar cuenta



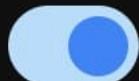
Preferencias



Notificaciones



Modo oscuro



Idioma

Español →

 Centro de Notificaciones – Alertas de Reserva, Promociones y Seguridad

← Notificaciones



Hoy

Nuevo recibo disponible

Tu recibo por el
estacionamiento en Real Plaza •
Salaverry está disponible



Hace 2 horas

Estacionamiento disponible

Hay espacios disponibles en tu
estacionamiento favorito: •
Larcomar



Hace 5 horas

Anteriores

Oferta especial

50% de descuento en tu próxima
reserva en Centro Cívico



Inicio



Notificaciones



Ajustes



Segmento Objetivo: Dueño de Estacionamiento – Flutter, Dart y Android Studio

📸 Pantalla de Inicio de Sesión – Login del Dueño de Estacionamiento

1:01

54%



ParkingNow

Panel de Propietarios

Administra tu estacionamiento de manera
inteligente



¡Bienvenido de
vuelta!

Accede a tu panel de control

Correo electrónico



ejemplo@correo.com

Contraseña



Tu contraseña es:



Tu Contraseña Seg...

❓ Pantalla de Recuperación – ¿Olvidaste tu Contraseña?

1:01

54%



¿Olvidaste tu contraseña?

Ingresá tu correo y te enviaremos un enlace para restablecerla



Correo electrónico

Enviar enlace de recuperación

[Volver al inicio de sesión](#)

¿Necesitas ayuda? Contacta a nuestro soporte



 Pantalla de Cambio de Contraseña – Validación Segura para el Dueño



Cambiar Contraseña

Crea una nueva contraseña segura para proteger tu cuenta



Nueva contraseña



Confirmar contraseña



Requisitos de contraseña:

- ✗ Al menos 8 caracteres
- ✗ Al menos una letra mayúscula (A-Z)
- ✗ Al menos un número (0-9)
- ✗ Al menos un símbolo (!@#\$%^&*)
- ✗ Las contraseñas coinciden

¿Necesitas ayuda? Contacta a nuestro soporte



Registro de Nuevo Usuario – Crear Cuenta como Dueño de Estacionamiento

1:02

54%



Paso 1 de 3



ParkingNow Owner

Únete a nuestra red de
propietarios



Información Personal

Cuéntanos sobre ti

Nombre completo



Ingresá tu nombre y apellidos

Correo electrónico



ejemplo@correo.com

Número de teléfono



+51 999 999 999

Siguiente



1:02

54%



Paso 2 de 3



ParkingNow Owner

Únete a nuestra red de
propietarios



Seguridad

Protege tu cuenta

Contraseña



.....



Confirmar contraseña



.....



Fortaleza de la contraseña: Débil

Anterior

Siguiente



1:02

54%



Paso 3 de 3



ParkingNow Owner

Únete a nuestra red de
propietarios



Información del Negocio

Detalles de tu estacionamiento

RUC



20123456789

Nombre del negocio



Estacionamiento Central

Dirección del estacionamiento



Av. Principal 123, Lima

Capacidad total



Dashboard Principal – Vista General de la Actividad del Estacionamiento

1:03

54%

Dashboard



¡Bienvenido de vuelta!

John Smith

📍 Lima, Perú

P

12

Espacios



8

Reservas



75%

Ocupación

Resumen de Ingresos

Junio 2025

Hoy

S/ 150.00

+12% vs semana pasada

Esta semana

S/ 1,200.00

+8% vs semana pasada

[Ver reporte completo](#)



Dashboard



Estaciona...



Reservas



Notificacio...



 **Términos y Condiciones – Revisión Legal por Parte del Dueño**

1:06 Ct

55%

Dashboard



¡Bienvenido de vuelta!

John Smith

Lima, Perú

P

12

Q

8

C

75%

Términos y Condiciones

Para continuar usando la app,
debes aceptar los términos y
condiciones.

Aceptar

Rechazar

Esta semana

S/ 1,200.00

+8% vs semana pasada

Ver reporte completo



Dashboard



Estaciona...



Reservas



Notificacio...



 **Barra Lateral – Navegación entre Funciones del Dueño de Estacionamiento**

1:06 Ct

55%



Juan Pérez

Propietario



Inicio



Registros de Locales

3



Reservas

5

%
ción

2025



Seguridad



Notificaciones

3

[→ Cerrar Sesión

P ParkingNow v1.0.0

© 2025 Todos los derechos reservados

Notificacio..

 Registro Avanzado – Añadir Múltiples Estacionamientos y Configurar Extras

1:03

54%



Registro de Estacionamiento

Paso 1 de 4

25%

Información Básica



Ingresá los datos principales de tu estacionamiento

Nombre del estacionamiento



Ej: Parking San Isidro

Descripción



Describe brevemente tu estacionamiento

Tipo de estacionamiento



Público



Número de espacios

...

Ej: 20

Siguiente

Ubicación del Estacionamiento – Marcado y Visualización Geográfica



Registro de Estacionamiento

Paso 2 de 4

50%

Ubicación



Indica dónde se encuentra tu
estacionamiento

Dirección o lugar



Ej: Av. Javier Prado 1234, ...



Información adicional

Referencias



Ej: Frente al parque, esquina con Av....

Anterior

Siguiente



 **Servicios del Estacionamiento – Horarios de Atención y Beneficios para Conductores**

1:04 Ct ☰ •

• VoLTE 54%



Registro de Estacionamiento

Paso 3 de 4

75%

Carga eléctrica

Acceso para discapacitados

Horario de atención



Horario de apertura



7:00 AM



Horario de cierre



10:00 PM

Días de atención

L

M

X

J

V

S

D

Anterior

Siguiente



 Configuración General – Tarifas, Métodos de Pago y Preferencias



Registro de Estacionamiento

Paso 4 de 4

100%



Precios y Términos

Establece las tarifas y condiciones

Tarifas

Precio por hora (S/)



Ej: 5.00



Tarifa por día

Estacionamiento
por día completo

S/ 25.00



Tarifa mensual

Abono mensual

S/ 300.00



Métodos de pago aceptados



Efectivo



Tarjeta

Anterior

Registrar



Gestión de Reservas – Visualización y Administración de Espacios Reservados

1:05 Ct

LTE 55%



Reservas



4

Total



1

Confirmadas



1

Pendientes



Carlos Rodríg...

NUEVO

Confirmada

ID: RES001

Toyota Corolla

ABC-123

P A-15

09:00 - 11:00

S/ 10.00

Contactar

Gestionar



María González

Pendiente

ID: RES002

Honda Civic

XYZ-789

P B-08

10:30 - 12:30

S/ 10.00

Contactar

Gestionar



Luis Pérez

Completada

ID: RES003

Nissan Sentra

DEF-456

P C-22

13:00 - 15:00

S/ 10.00



Contactar

Gestionar

 Módulo de Seguridad – Protección del Espacio y Alertas del Estacionamiento

1:05 Ct

55%



Centro de Seguridad



3

Vehículos
Registrados



1

Alertas
Importantes



24/7

Monitoreo
Activo



99.9%

Tiempo
Operativo

Acciones Rápidas



Registrar

Nuevo vehículo



Historial

Ver actividad

Actividad Reciente

 Centro de Notificaciones y Configuración – Alertas y Preferencias del Dueño

1:05 Ct

55%



Notificac...



¡Hola, John!

Tienes 3 notificaciones sin leer



3

Nuevas



2

Urgentes

✓ Todas

Reservas

Pagos

Recientes

Historial



Nueva reserva confirmada



Carlos Rodríguez ha confirmado su reserva para mañana a las 09:...

5m



Pago recibido



Has recibido un pago de S/ 25.00 por la reserva #12345. El pago ha...

15m



URGENTE

✓ Marcar como leídas



Configuración del Sistema – Ajustes Generales del Dueño y Preferencias de la Plataforma

1:05 Ct

55%



Configuraciones

Personaliza tu experiencia



Juan Carlos Pérez

Propietario Premium



Cuenta Verificada

General



Modo Oscuro

Cambia la apariencia de la aplicación



Idioma

Español



Moneda

PEN (S/)



Zona Horaria

GMT-5 (Lima)



Notificaciones

6.2.2.6. Services Documentation Evidence for Sprint Review

Durante este Sprint se documentaron los endpoints correspondientes a los siguientes módulos del backend de ParkingNow:

Link base de la aplicación: <https://parkingnow-app-963963f523b4.herokuapp.com/>

Endpoint obtencion de usuarios: <https://parkingnow-app-963963f523b4.herokuapp.com/usuario>

- local
- reserva
- usuario

Cada módulo cuenta con sus respectivos endpoints documentados mediante Swagger. A continuación, se presentarán capturas de la documentación generada que ilustran la estructura, rutas y organización de los servicios expuestos.

Usuario

<http://localhost:3000/usuario>

POST Crear un nuevo usuario

The screenshot shows the Postman application interface. On the left, there's a sidebar with collections: 'contigoNovoy', 'assistent', and 'parkingnow'. Under 'parkingnow', there are three requests: 'GET get', 'POST nuevo local', and 'POST Crear un nuevo usuario'. The 'POST Crear un nuevo usuario' request is selected. The main panel shows a POST request to 'http://localhost:3000/usuario'. The 'Body' tab is selected, showing a JSON payload:

```
1 {  
2   "name": "Jesús Lopez",  
3   "email": "jesuslz@example.com",  
4   "password": "pass123",  
5   "placa": "ABC-321",  
6   "dni": "72787997"  
7 }
```

The response tab shows the result of the POST request:

```
1 {  
2   "id": 2,  
3   "name": "Jesús Lopez",  
4   "email": "jesuslz@example.com",  
5   "password": "pass123",  
6   "placa": "ABC-321",  
7   "dni": "72787997",  
8   "ruc": null,  
9   "tipoUsuario": "conductor"  
10 }
```

The status bar at the bottom right indicates a 201 Created response with 57ms latency and 153B size.

<http://localhost:3000/usuario>

GET Obtener todos los usuarios

Collection Edit View Window Help

bruno ... parkingnow Safe Mode No Environment

Collections

Search requests ...

> contigoNovoy

> assistent

✓ parkingnow

 GET get

 POST nuevo local

 POST Crear un nuevo usuario

Params Body Headers Auth * Vars Script

Assert Tests Docs

Query

Name Path

+ Add Param

Path ?

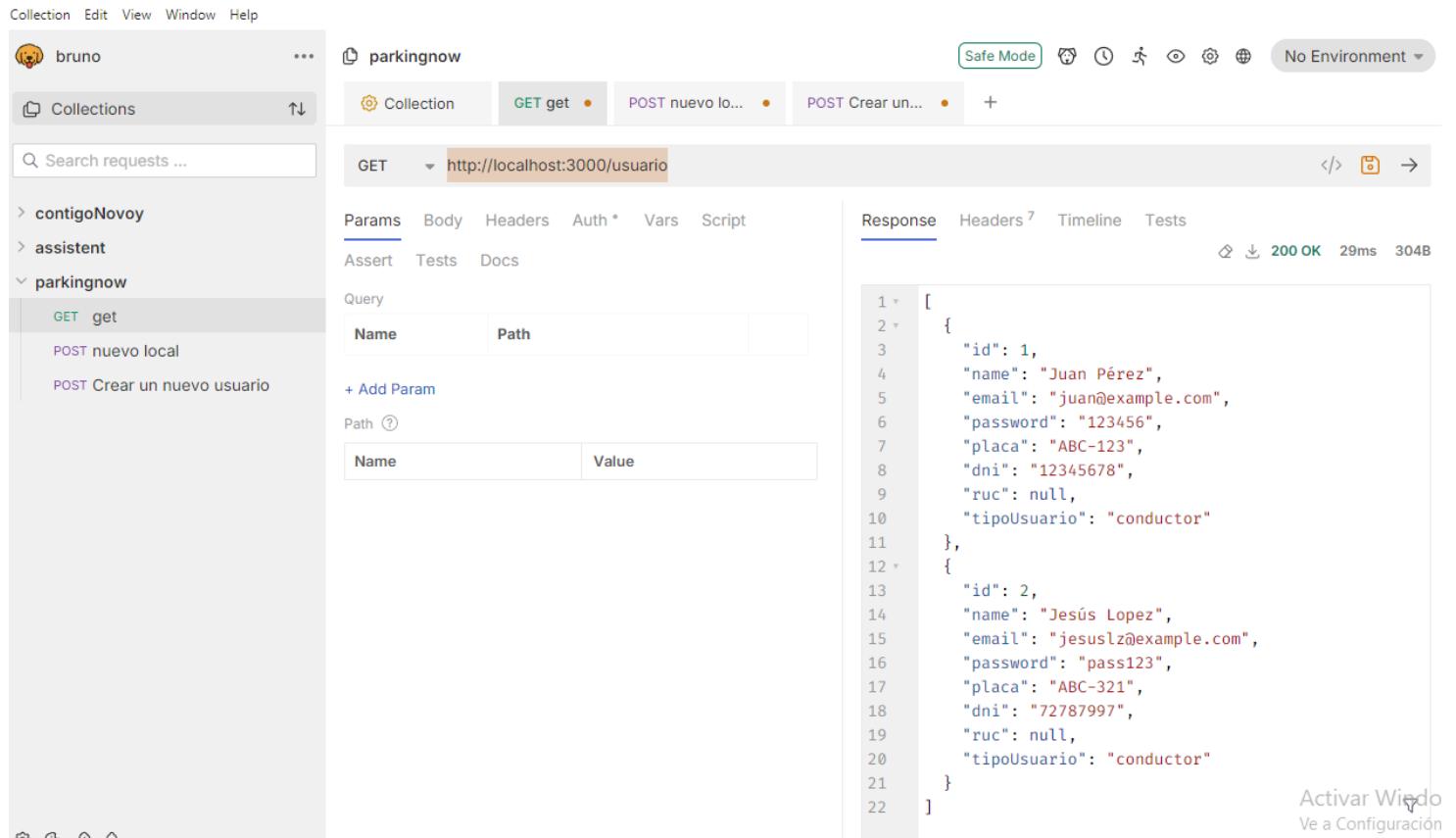
Name Value

Response Headers 7 Timeline Tests

200 OK 29ms 304B

```
[  
  {  
    "id": 1,  
    "name": "Juan Pérez",  
    "email": "juan@example.com",  
    "password": "123456",  
    "placa": "ABC-123",  
    "dni": "12345678",  
    "ruc": null,  
    "tipoUsuario": "conductor"  
  },  
  {  
    "id": 2,  
    "name": "Jesús Lopez",  
    "email": "jesuslz@example.com",  
    "password": "pass123",  
    "placa": "ABC-321",  
    "dni": "72787997",  
    "ruc": null,  
    "tipoUsuario": "conductor"  
  }]
```

Activar Winfo
Ve a Configuración



<http://localhost:3000/usuario/1>

get Obtener un usuario por ID

Collection Edit View Window Help

bruno ... parkingnow Safe Mode No Environment

Collections

Search requests ...

> contigoNovoy

> assistent

✓ parkingnow

 GET get

 POST nuevo local

 POST Crear un nuevo usuario

Params Body Headers Auth * Vars Script

Assert Tests Docs

Query

Name Path

+ Add Param

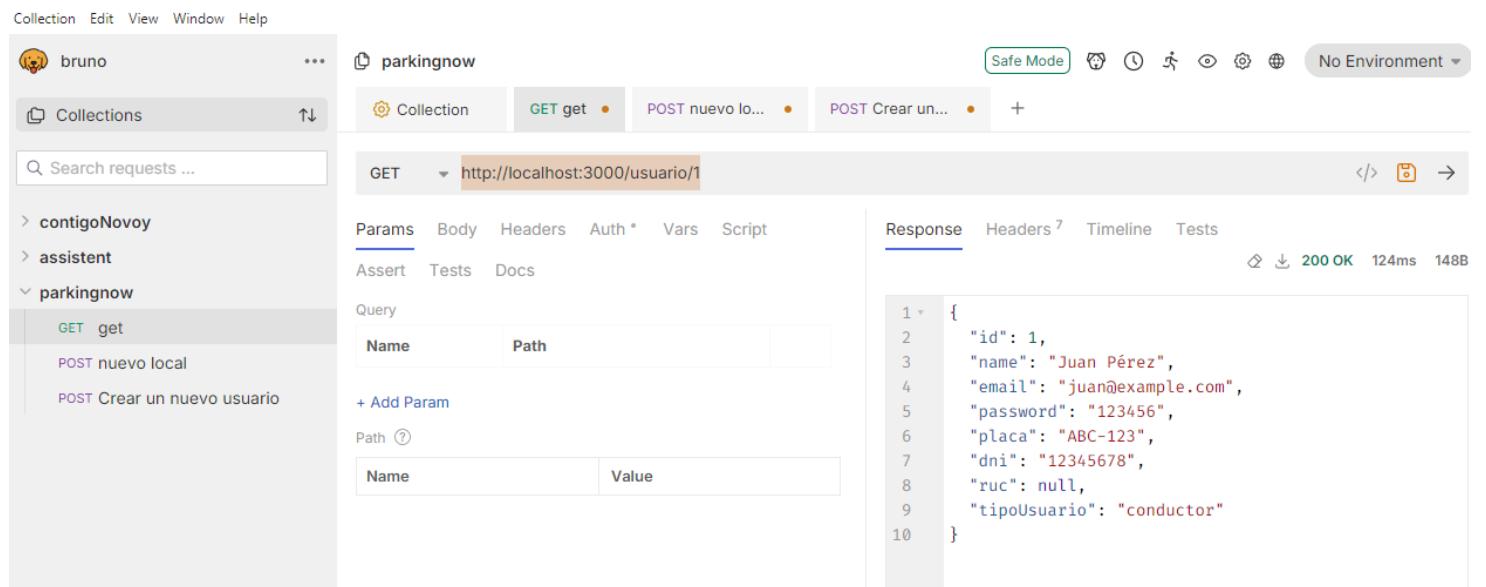
Path ?

Name Value

Response Headers 7 Timeline Tests

200 OK 124ms 148B

```
{  
  "id": 1,  
  "name": "Juan Pérez",  
  "email": "juan@example.com",  
  "password": "123456",  
  "placa": "ABC-123",  
  "dni": "12345678",  
  "ruc": null,  
  "tipoUsuario": "conductor"  
}
```



<http://localhost:3000/usuario/1/locales>

GET Obtiene todos los locales asociados a un usuario específico.

parkingnow

GET http://localhost:3000/usuario/1/locales

Params Body Headers Auth * Vars Script

Assert Tests Docs

Query

Name	Path

+ Add Param

Path ?

Name	Value

Response Headers 7 Timeline Tests ⏲️ 200 OK 33ms 101B

```

1 [
2   {
3     "id": 4,
4     "nombre": "Playa Central",
5     "direccion": "Av. Principal 123",
6     "telefono": "987654321",
7     "plazas": 1
8   }
9 ]

```

http://localhost:3000/usuario/1

PUT Actualizar un usuario por ID

parkingnow

PUT http://localhost:3000/usuario/1

Params Body * Headers Auth * Vars Script

Assert Tests Docs JSON Prettify

Name	Value
"name"	Juan Actualizado
"email"	juan_updated@example.com
"placa"	XYZ-789

Response Headers 7 Timeline Tests ⏲️ 200 OK 148ms 161B

```

1 {
2   "id": 1,
3   "name": "Juan Actualizado",
4   "email": "juan_updated@example.com",
5   "password": "123456",
6   "placa": "XYZ-789",
7   "dni": "12345678",
8   "ruc": null,
9   "tipoUsuario": "conductor"
10 }

```

http://localhost:3000/usuario/1

DELETE Eliminar un usuario por ID

parkingnow

DELETE http://localhost:3000/usuario/1

Params Body Headers Auth * Vars Script

Assert Tests Docs

Query

Name	Path

+ Add Param

Path ?

Name	Value

Response Headers 5 Timeline Tests ⏲️ 200 OK 39ms 0B

```

1

```

Local

<http://localhost:3000/local>

POST Crear un nuevo local

POST http://localhost:3000/local

Body (JSON)

```
{
  "nombre": "Playa Central",
  "direccion": "Av. Principal 123",
  "telefono": "987654321",
  "usuario": {
    "id": 2
  }
}
```

Response (201 Created)

```
{
  "id": 5,
  "nombre": "Playa Central",
  "direccion": "Av. Principal 123",
  "telefono": "987654321",
  "plazas": 1,
  "usuario": {
    "id": 2
  }
}
```

<http://localhost:3000/local/5>

GET Obtener un local por ID

GET http://localhost:3000/local/5

Query

Name	Path

+ Add Param

Path

Name	Value

Response (200 OK)

```
{
  "id": 5,
  "nombre": "Playa Central",
  "direccion": "Av. Principal 123",
  "telefono": "987654321",
  "plazas": 1,
  "usuario": {
    "id": 2,
    "name": "Jesus Lopez",
    "email": "jesuslz@example.com",
    "password": "pass123",
    "placa": "ABC-321",
    "dni": "72787997",
    "ruc": null,
    "tipoUsuario": "conductor"
  }
}
```

<http://localhost:3000/local>

GET Obtener todos los locales

Collections

Collection GET get POST Crear nu... POST Crear un... > +

Search requests ...

- > contigoNovoy
- > assistent
- < parkingnow
 - GET get**
 - POST Crear nuevo local
 - POST Crear un nuevo usuario
 - PUT Actualizar un usuario po...
 - DEL Eliminar un usuario por ID

Params Body Headers Auth *

Vars Script Assert Tests Docs

Query

Name	Path
+ Add Param	

Path ?

Name	Value

Response Headers 7 Timeline Tests

200 OK 36ms 265B

```

1 [
2 {
3   "id": 5,
4   "nombre": "Playa Central",
5   "direccion": "Av. Principal 1
23",
6   "telefono": "987654321",
7   "plazas": 1,
8   "usuario": {
9     "id": 2,
10    "name": "Jesús Lopez",
11    "email": "jesuslz@example.c
om",
12    "password": "pass123",
13    "placa": "ABC-321",
14    "dni": "72787997",
15    "ruc": null,
16    "tipoUsuario": "conductor"
17  }
18 }

```

http://localhost:3000/local/5
PUT Actualizar un local por ID

bruno ***

Collections

parkingnow

Save Mode No Environment

Search requests ...

- > contigoNovoy
- > assistent
- < parkingnow
 - PUT Actualizar...**
 - POST Crear nu... POST Crear un... PUT Actualizar... > +

PUT http://localhost:3000/local/5 </> →

Params Body * Headers Auth *

Vars Script Assert Tests Docs

JSON Prettify

```

1 {
2   "nombre": "Playa Actualiz
ada",
3   "direccion": "Av. Nueva 7
89",
4   "telefono": "987654323"
5 }

```

Response Headers 7 Timeline Tests

200 OK 36ms 263B

```

1 {
2   "id": 5,
3   "nombre": "Playa Actualizada",
4   "direccion": "Av. Nueva 789",
5   "telefono": "987654323",
6   "plazas": 1,
7   "usuario": {
8     "id": 2,
9     "name": "Jesús Lopez",
10    "email": "jesuslz@example.com"
11 ,
12     "password": "pass123",
13     "placa": "ABC-321",
14     "dni": "72787997",
15     "ruc": null,
16     "tipoUsuario": "conductor"
17   }

```

http://localhost:3000/local/5
DELETE Eliminar un local por ID

The screenshot shows the Postman interface with a collection named 'parkingnow'. A DELETE request is selected with the URL `http://localhost:3000/local/5`. The 'Params' tab is active, showing a single parameter 'name' with value 'Path'. The 'Response' tab shows a status of 200 OK with a response body containing the number '1'.

<http://localhost:3000/local/6/usuario>

GET Obtener el usuario dueño de un local

The screenshot shows the Postman interface with a collection named 'parkingnow'. A GET request is selected with the URL `http://localhost:3000/local/6/usuario`. The 'Params' tab is active, showing a single parameter 'name' with value 'Path'. The 'Response' tab shows a status of 200 OK with a response body containing a JSON object:

```
{  
  "id": 2,  
  "name": "Jesús Lopez",  
  "email": "jesuslz@example.com",  
  "password": "pass123",  
  "placa": "ABC-321",  
  "dni": "72787997",  
  "ruc": null,  
  "tipoUsuario": "conductor"  
}
```

Reserva

<http://localhost:3000/reserva>

POST Crear una nueva reserva

bruno ... parkingnow Safe Mode No Environment

Collections ↕ Search requests ...

- > contigoNovoy
- > assistent
- ✓ parkingnow
 - GET get
 - POST Crear nueva reserva**
 - POST Crear un nuevo usuario
 - PUT Actualizar por ID
 - DEL Eliminar por ID

POST http://localhost:3000/reserva

Params Body * Headers Auth * Vars
Script Assert Tests Docs JSON Prettify

```

1 {
2   "usuario": { "id": 2 },
3   "local": { "id": 6 },
4   "nro_plaza": 5,
5   "fh_inicio": "2025-05-16T08:00:00Z",
6   "fh_fin": "2025-05-16T10:00:00Z"
7 }
```

Response Headers 7 Timeline Tests
🔗 ⏺ 201 Created 42ms 125B

```

1 {
2   "id": 1,
3   "usuario": {
4     "id": 2
5   },
6   "local": {
7     "id": 6
8   },
9   "nro_plaza": 5,
10  "fh_inicio": "2025-05-16T08:00:00Z",
11  "fh_fin": "2025-05-16T10:00:00Z"
12 }
```

http://localhost:3000/reserva
GET Obtener todas las reservas

Q Search requests ...

GET http://localhost:3000/reserva

Params Body Headers Auth * Vars Script
Assert Tests Docs

Query

Name	Path
+ Add Param	

Path ⓘ

Name	Value

Response Headers 7 Timeline Tests
🔗 ⏺ 200 OK 29ms 371B

```

1 [
2   {
3     "id": 1,
4     "usuario": {
5       "id": 2,
6       "name": "Jesús Lopez",
7       "email": "jesuslz@example.com",
8       "password": "pass123",
9       "placa": "ABC-321",
10      "dni": "72787997",
11      "ruc": null,
12      "tipoUsuario": "conductor"
13    },
14    "local": {
15      "id": 6,
16      "nombre": "Playa Central",
17      "direccion": "Av. Principal 123",
18      "telefono": "987654321",
19      "plazas": 1
20    },
21    "nro_plaza": 5,
22    "fh_inicio": "2025-05-16T08:00:00.000Z",
23    "fh_fin": "2025-05-16T10:00:00.000Z"
24  }
25 ]
```

http://localhost:3000/reserva/1
GET Obtener una reserva por ID

bruno ... parkingnow Safe Mode No Environment

Collections < on GET get • POST Crear nu... • POST Crear un... • PUT Actualizar... • DELETE Elimina... • > +

Q Search requests ...

> contigoNovoy
> assistent
✓ parkingnow
 GET get
 POST Crear nueva reserva
 POST Crear un nuevo usuario
 PUT Actualizar por ID
 DEL Eliminar por ID

GET http://localhost:3000/reserva/1

Params Body Headers Auth * Vars Script
Assert Tests Docs

Query

Name	Path
+ Add Param	

Path ?

Name	Value

Response Headers 7 Timeline Tests
200 OK 139ms 369E

```

1 {  
2   "id": 1,  
3   "usuario": {  
4     "id": 2,  
5     "name": "Jesús Lopez",  
6     "email": "jesuslz@example.com",  
7     "password": "pass123",  
8     "placa": "ABC-321",  
9     "dni": "72787997",  
10    "ruc": null,  
11    "tipoUsuario": "conductor"  
12  },  
13  "local": {  
14    "id": 6,  
15    "nombre": "Playa Central",  
16    "direccion": "Av. Principal 123",  
17    "telefono": "987654321",  
18    "plazas": 1  
19  },  
20  "nro_plaza": 5,  
21  "fh_inicio": "2025-05-16T08:00:00.000Z",  
22  "fh_fin": "2025-05-16T10:00:00.000Z"  
23 }

```

<http://localhost:3000/reserva/1>

PUT Actualizar una reserva por ID

PUT http://localhost:3000/reserva/1

Params Body * Headers Auth * Vars Script
Assert Tests Docs JSON Prettify

1 {
2 "nro_plaza": 10,
3 "fh_fin": "2025-05-16T12:00:00Z"
4 }

Response Headers 7 Timeline Tests
200 OK 31ms 366E

```

1 {  
2   "id": 1,  
3   "usuario": {  
4     "id": 2,  
5     "name": "Jesús Lopez",  
6     "email": "jesuslz@example.com",  
7     "password": "pass123",  
8     "placa": "ABC-321",  
9     "dni": "72787997",  
10    "ruc": null,  
11    "tipoUsuario": "conductor"  
12  },  
13  "local": {  
14    "id": 6,  
15    "nombre": "Playa Central",  
16    "direccion": "Av. Principal 123",  
17    "telefono": "987654321",  
18    "plazas": 1  
19  },  
20  "nro_plaza": 10,  
21  "fh_inicio": "2025-05-16T08:00:00.000Z",  
22  "fh_fin": "2025-05-16T12:00:00Z"  
23 }

```

<http://localhost:3000/reserva/1>

DELETE Eliminar una reserva por ID

The screenshot shows the Postman interface. On the left, there's a sidebar with collections like 'contigoNovoy', 'assistant', and 'parkingnow'. Under 'parkingnow', there are several methods: 'GET get', 'POST Crear nueva reserva', 'POST Crear un nuevo usuario', 'PUT Actualizar por ID', and 'DEL Eliminar por ID'. The 'DEL Eliminar por ID' method is selected. In the main area, a 'DELETE' request is made to 'http://localhost:3000/reserva/1'. The 'Params' tab is active, showing a single parameter 'Name' with value 'Path'. The 'Response' tab shows a status of '200 OK' with a response body containing the number '1'.

Durante este Sprint desplegamos el backend e hicimos las pruebas requeridas

Inicio del despliegue

Empezamos logueandonos en la plataforma

```
samuel@DESKTOP-JPEIUCT MINGW64 /d/upc/Aplicaciones Moviles/Backend (main)
$ npm install --save @nestjs/cli
up to date, audited 895 packages in 17s

163 packages are looking for funding
  run `npm fund` for details

1 low severity vulnerability

To address all issues, run:
  npm audit fix

Run `npm audit` for details.

samuel@DESKTOP-JPEIUCT MINGW64 /d/upc/Aplicaciones Moviles/Backend (main)
$ heroku login
heroku: Press any key to open up the browser to login or q to exit:
Opening browser to https://cli-auth.herokuapp.com/auth/cli/browser/c2ebda91-4842-47c8-b8ca-4e1b729c9e4?requestor=SFMyNTY.g2g0bQAAAAsxNzkuNi4zLjIyNg4GACjK6Y-XAIIAAVGA.wQuA0jvUEmrhi3t3vGvn7GY0xPq26Z90YRykHNqp8
Logging in... done
Logged in as samuelmolina281816@gmail.com
```

Configuracion de la base de datos

Configuramos una base de datos con NEON

```
samuel@DESKTOP-JPEIUCT MINGW64 /d/upc/Aplicaciones Moviles/Backend (main)
$ heroku config:set DATABASE_URL="postgresql://neondb_owner:npq_Ytv3go6FPQM@ep-floral-shadow-a5w34cdf-pooler.us-east-2.aws.neon.tech/neondb?sslmode=require"
Setting DATABASE_URL and restarting ⚡ parkingnow-app... done, v3
DATABASE_URL: postgresql://neondb_owner:npq_Ytv3go6FPQM@ep-floral-shadow-a5w34cdf-pooler.us-east-2.aws.neon.tech/neondb?sslmode=require

samuel@DESKTOP-JPEIUCT MINGW64 /d/upc/Aplicaciones Moviles/Backend (main)
$

samuel@DESKTOP-JPEIUCT MINGW64 /d/upc/Aplicaciones Moviles/Backend (main)
$ git status
On branch main
Your branch is up to date with 'origin/main'.

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
    (use "git restore <file>..." to discard changes in working directory)
      modified: package-lock.json
      modified: package.json
      modified: src/app.module.ts

no changes added to commit (use "git add" and/or "git commit -a")
```

Cambios finales para producción

Se agregan cambios a producción para la ejecución correcta de Nest en la plataforma

The screenshot shows a terminal window with the following content:

```
1 package.json > {} scripts
8 "scripts": {
9   "build": "nest build",
10  "format": "prettier --write \"src/**/*.ts\" \"test/**/*.ts\"",
11  "start": "node dist/main.js",
12  "start:dev": "nest start --watch",
13  "start:debug": "nest start --debug --watch",
14  "start:prod": "node dist/main",
15  "lint": "eslint \"{src,apps,libs,test}/**/*.{ts,js}\" --fix",
16  "test": "jest",
17  "test:watch": "jest --watch",
18  "test:cov": "jest --coverage",
19  "test:debug": "node --inspect-brk -r tsconfig-paths/register -r ts-node/register node_modules/.bin/jest --runInBand",
20  "test:e2e": "jest --config ./test/jest-e2e.json"
21 },
22 "dependencies": {
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
samue@DESKTOP-JPEIUCT MINGW64 /d/upc/Aplicaciones Moviles/Backend (main)
● $ git status
On branch main
Your branch is up to date with 'origin/main'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    modified:   package-lock.json
    modified:   package.json
    modified:   src/app.module.ts

samue@DESKTOP-JPEIUCT MINGW64 /d/upc/Aplicaciones Moviles/Backend (main)
● $ git commit -m "Configuraciones para la conexión a producción a Neon PostgreSQL"
[main 99663ca] Configuraciones para la conexión a producción a Neon PostgreSQL
 3 files changed, 53 insertions(+), 10 deletions(-)

samue@DESKTOP-JPEIUCT MINGW64 /d/upc/Aplicaciones Moviles/Backend (main)
● $ git push heroku main
Enumerating objects: 84, done.
Counting objects: 100% (84/84), done.
Delta compression using up to 8 threads
Compressing objects: 100% (82/82), done.
Writing objects: 100% (84/84), 117.11 KiB | 3.90 MiB/s, done.
Total 84 (delta 37), reused 18 (delta 0), pack-reused 0 (from 0)
remote: Updated 29 paths from eb4b912
remote: Compressing source files... done.
remote: Building source:
remote:
remote: -----> Building on the Heroku-24 stack
remote: -----> Determining which buildpack to use for this app
remote: -----> Node.js app detected
remote:
```

Obtenemos el log del despliegue y comprobamos la conexión a la base de datos

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
remote:     npm notice Changelog: https://github.com/npm/cli/releases/tag/v11.4.2
remote:     npm notice To update run: npm install -g npm@11.4.2
remote:     npm notice
remote:
remote: ----> Build
remote:     Running build
remote:
remote: > backend-parkingnow@0.0.1 build
remote: > nest build
remote:
remote:
remote: ----> Caching build
remote:     - npm cache
remote:
remote: ----> Pruning devDependencies
remote:
remote:     up to date, audited 215 packages in 1s
remote:
remote:     47 packages are looking for funding
remote:       run `npm fund` for details
remote:
remote:     1 low severity vulnerability
remote:
remote:     To address all issues, run:
remote:       npm audit fix
remote:
remote:     Run `npm audit` for details.
remote:     npm notice
remote:     npm notice New major version of npm available! 10.9.2 -> 11.4.2
remote:     npm notice Changelog: https://github.com/npm/cli/releases/tag/v11.4.2
remote:     npm notice To update run: npm install -g npm@11.4.2
remote:     npm notice
remote:
remote: ----> Build succeeded!
remote: ----> Discovering process types
remote:     Procfile declares types    -> (none)
remote:     Default types for buildpack -> web
remote:
remote: ----> Compressing...
remote:     Done: 65M
remote: ----> Launching...
remote:     Released v4
remote:     https://parkingnow-app-963963f523b4.herokuapp.com/ deployed to Heroku
remote:
remote: Verifying deploy... done.
To https://git.heroku.com/parkingnow-app.git
 * [new branch]      main -> main

samue@DESKTOP-JPEIUCT MINGW64 /d/upc/Aplicaciones Moviles/Backend (main)
$ heroku config:get DATABASE_URL
postgresql://neondb_owner:npg_Ytv3go6FPQOM@ep-floral-shadow-a5w34cdf-pooler.us-east-2.aws.neon.tech/neondb?sslmode=require

samue@DESKTOP-JPEIUCT MINGW64 /d/upc/Aplicaciones Moviles/Backend (main)
$ heroku open
```

Comprobacion del despliegue

Revision de los logs de despliegue

```
samue@DESKTOP-JPEIUCT MINGW64 /d/upc/Aplicaciones Moviles/Backend (main)
$ heroku logs --tail
2025-06-21T00:40:20.199533+00:00 app[api]: Release v1 created by user samuelmolina281816@gmail.com
2025-06-21T00:40:20.199533+00:00 app[api]: Initial release by user samuelmolina281816@gmail.com
2025-06-21T00:40:20.402899+00:00 app[api]: Release v2 created by user samuelmolina281816@gmail.com
2025-06-21T00:40:20.402899+00:00 app[api]: Enable Logplex by user samuelmolina281816@gmail.com
2025-06-21T01:16:08.600619+00:00 app[api]: Release v3 created by user samuelmolina281816@gmail.com
2025-06-21T01:16:08.600619+00:00 app[api]: Set DATABASE_URL config vars by user samuelmolina281816@gmail.com
2025-06-21T01:18:25.000000+00:00 app[api]: Build started by user samuelmolina281816@gmail.com
2025-06-21T01:18:52.897281+00:00 app[api]: Deploy 99663ca6 by user samuelmolina281816@gmail.com
2025-06-21T01:18:52.897281+00:00 app[api]: Release v4 created by user samuelmolina281816@gmail.com
2025-06-21T01:18:52.909856+00:00 app[api]: Scaled to web@1:Basic by user samuelmolina281816@gmail.com
2025-06-21T01:18:55.000000+00:00 app[api]: Build succeeded
```

Sin conflicto, despliegue exitoso

Detalles de la aplicación en la plataforma

dashboard.heroku.com/apps/parkingnow-app

HEROKU from Salesforce

Overview Resources Deploy Metrics Activity Access Settings

Get a complete visualization of your app in a team-based continuous delivery environment with [Heroku Pipelines](#). [Hide](#) [Create a Heroku Pipeline](#)

Metrics (last 24hrs) [All Metrics](#)

Response Time
191ms

Throughput
< 1 rps

Memory
12 %

Latest Activity [All Activity](#)

samuelmolina281816@gmail.com: Deployed dd1a4b0d Today at 8:24 PM

samuelmolina281816@gmail.com: Build succeeded Today at 8:24 PM [View build log](#)

samuelmolina281816@gmail.com: Deployed 99663c6b Today at 8:18 PM v4

samuelmolina281816@gmail.com: Build succeeded Today at 8:18 PM [View build log](#)

samuelmolina281816@gmail.com: Set DATABASE_URL config var Today at 8:16 PM v3

samuelmolina281816@gmail.com: Enable Logplex Today at 7:40 PM v2

samuelmolina281816@gmail.com: Initial release Today at 7:40 PM v1

Installed Add-ons [Configure Add-ons](#)

There are no add-ons for this app
You can add add-ons to this app and they will show here. [Learn more](#)

Dyno Formations [Configure Dynos](#)

This app is using basic dynos

web npm start **ON**

Collaborator Activity [Manage Access](#)

samuelmolina281816@gmail.com [2 deploys](#)

Testeo de la aplicación

Testeo de la aplicacion en producción

HTTP <http://localhost:3000/usuario> [Save](#)

POST <https://parkingnow-app-963963f523b4.herokuapp.com/usuario> [Send](#)

Params Authorization Headers (8) **Body** [Pre-request Script](#) [Tests](#) [Settings](#)

None form-data x-www-form-urlencoded raw binary JSON

```

1
2   "name": "Samuel Molina",
3   "email": "samuel@example.com",
4   "password": "Pa$$word963",
5   "placa": "AKL-963",
6   "dni": "12345678",
7   "ruc": "20123456789"
8

```

Body [Cookies](#) [Headers \(11\)](#) [Test Results](#) Status: 201 Created Time: 306 ms Size: 933 B [Save Response](#)

Pretty Raw Preview Visualize JSON

```

1
2   "id": 2,
3   "name": "Samuel Molina",
4   "email": "samuel@example.com",
5   "password": "Pa$$word963",
6   "placa": "AKL-963",
7   "dni": "12345678",
8   "ruc": "20123456789",
9   "tipoUsuario": "dueno_estacionamiento"
10

```

Verificamos la creacion de registros en la base de datos de producción

The screenshot shows the Neon SQL Editor interface. On the left, there's a sidebar with 'PROJECT' and 'BRANCH' sections. Under 'PROJECT', 'Dashboard', 'Branches', 'Integrations', 'Auth', and 'Settings' are listed. Under 'BRANCH', 'production' is selected. In the main area, there's a 'SQL Editor' tab with a query window. The query is:

```

1 /*
2 In Neon, databases are stored on branches. By default, a project has one branch and one database.
3 You can select the branch and database to use from the drop-down menus above.
4
5 Try generating sample data and querying it by running the example statements below, or click
6 New Query to clear the editor.
7 */
8 select * from usuario;

```

Below the query window, there's a results table titled 'Connected (1 query)'. It shows two rows of data from the 'usuario' table:

#	id	name	email	password	placa	dni	ruc	tipo_usuario
1	1	Diego Ulises	diego@example.com	123456	ABC-123	12345678	20123456789	dueno_estacionamiento
2	2	Samuel Molina	samuel@example.com	Pa\$\$word963	AKL-963	12345678	20123456789	dueno_estacionamiento

At the bottom right of the results table, it says '152ms 2 rows'.

Comprobacion de las apis abieretas

6.2.2.7. Software Deployment Evidence for Sprint Review

Durante este sprint, ambas aplicaciones móviles fueron desarrolladas utilizando **Android Studio** y desplegadas correctamente para pruebas funcionales.

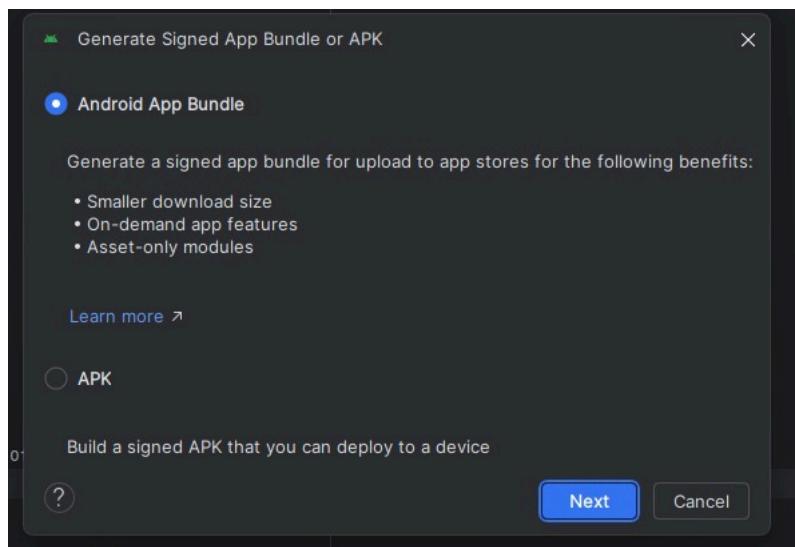
Cada segmento fue ejecutado mostrando correctamente su pantalla de inicio, navegación y funcionalidades clave, validando la estabilidad visual y lógica en entorno Android.

Además, se ha generado el archivo **APK instalable**, disponible públicamente para su descarga desde cualquier dispositivo Android.

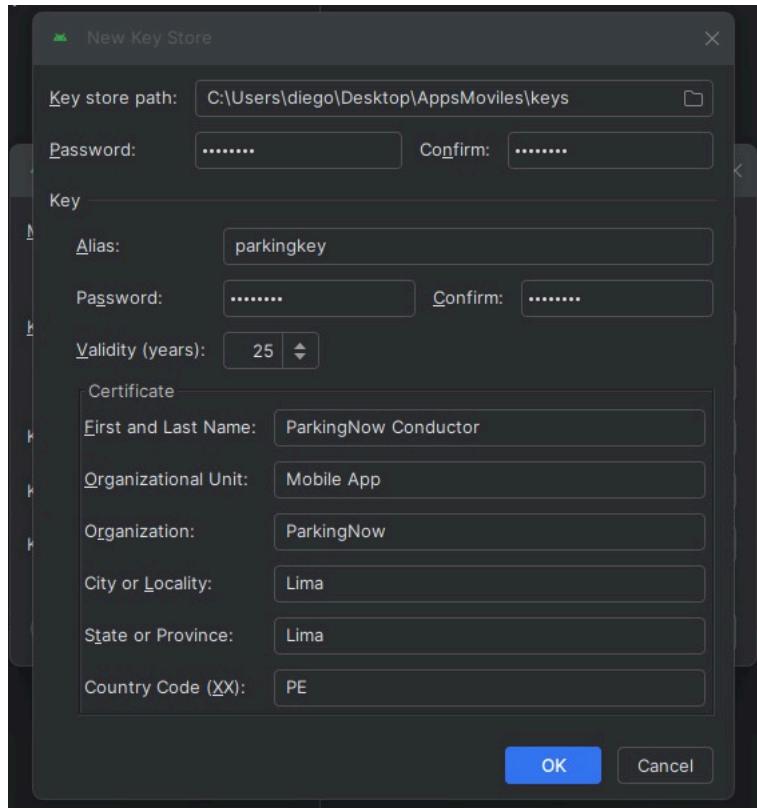
APK disponible aquí: <https://drive.google.com/drive/folders/1z7wYWMZDh1zbb6wGD3-tZEP9n5YQBU84?usp=sharing>

Deployment de Flutter y Kotlin

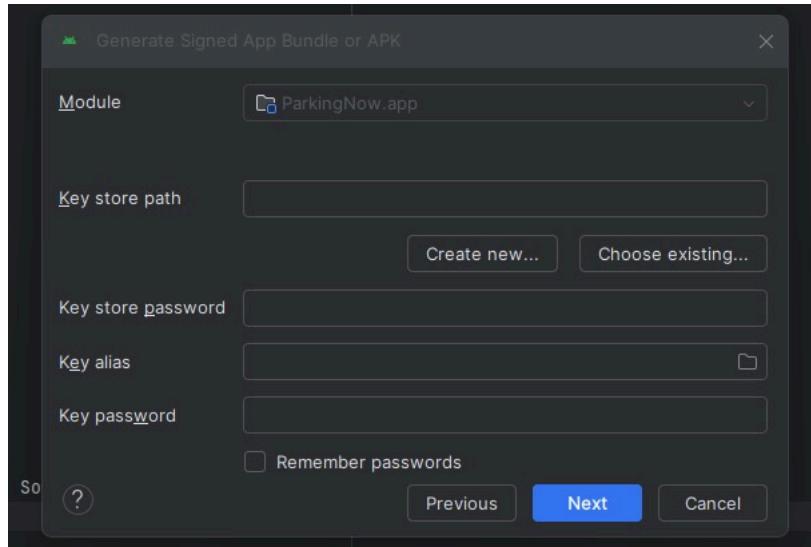
Primero vamos a build y lo que haremos es firmar la aplicación, los requisitos es igual tanto para flutter y kotlin así que se cumplen los mismos pasos y le damos clic en next



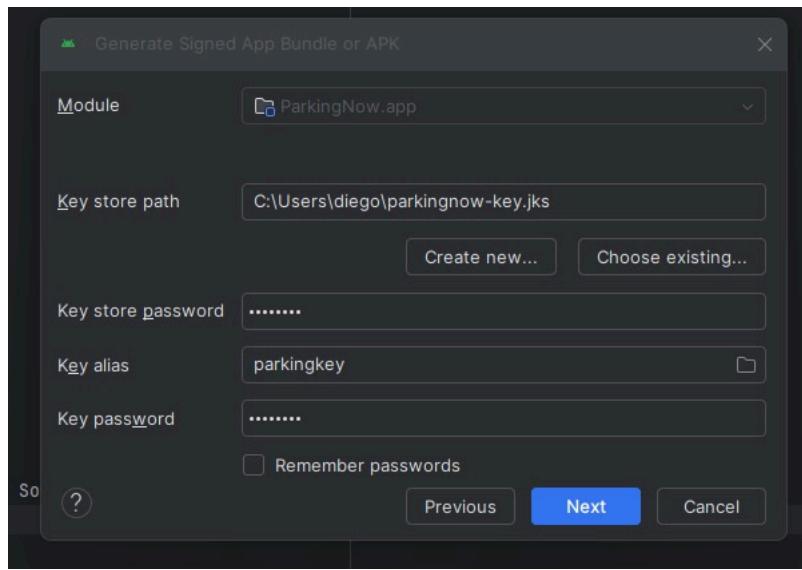
Una vez llenado los datos y completado el formulario le damos clic en okay, todo esto se repite igual en flutter



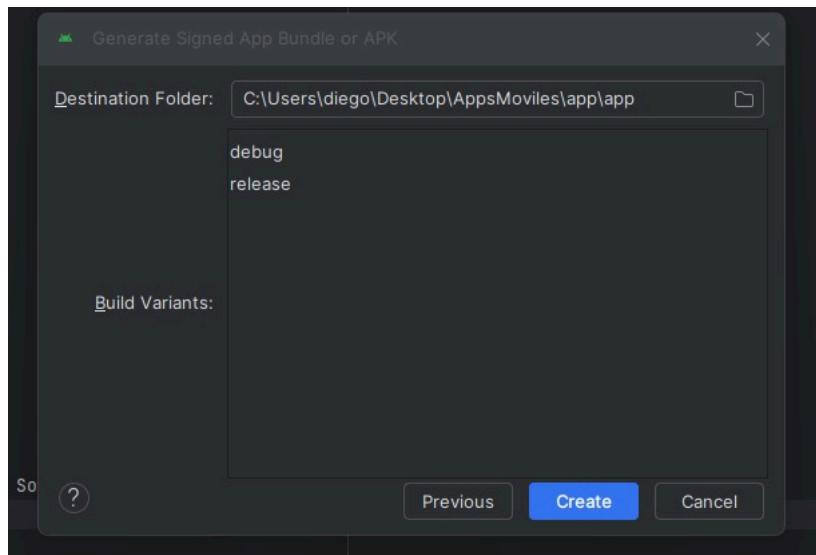
Aquí nos pide en este caso algunos requisitos para poder firmar el apk, le vamos a dar clic en create new



Aquí validamos en este caso que todo este correctamente y coincidan con lo que hemos llenado



Y aquí finalizamos con el proceso y le damos en create, lo mismo con flutter



Aquí nos vamos a app distribution en firebase go to console

The screenshot shows the Firebase console interface. On the left, there's a sidebar with navigation links like 'Aplicaciones', 'Gmail', 'YouTube', 'Maps', 'Plantillas para Power...', 'Convertir tus PDF a...', 'Descargar archivo', 'Home - Replit', 'Universe | The Largest...', 'Stitch - Design with...', 'v0 by Vercel', and 'Postgres'. The main area is titled 'ParkingNow-App > Descripción general del proyecto > Productos y funciones de Firebase'. Under 'Ejecución', there are five cards: 'Realtime Database' (almacena y sincroniza datos en tiempo real), 'Storage' (almacena y recupera contenido generado por el usuario), 'A/B Testing' (mejora los flujos y las notificaciones clave), 'Analytics' (mide y analiza la participación de los usuarios), and 'App Distribution' (distribuye versiones de app previas al lanzamiento).

Luego aca añadimos y registramos el apk, llenamos todo los campos como kotlin y flutter

Agregar Firebase a tu app para Android

-  Registrar app
Nombre del paquete de Android: pe.edu.upc.parkingnow
-  Descargar y, luego, agregar el archivo de configuración
-  Agregar el SDK de Firebase
-  Próximos pasos

JListo!

Asegúrate de revisar la [documentación](#) para conocer los primeros pasos con cada producto de Firebase que quieras usar en tu app.

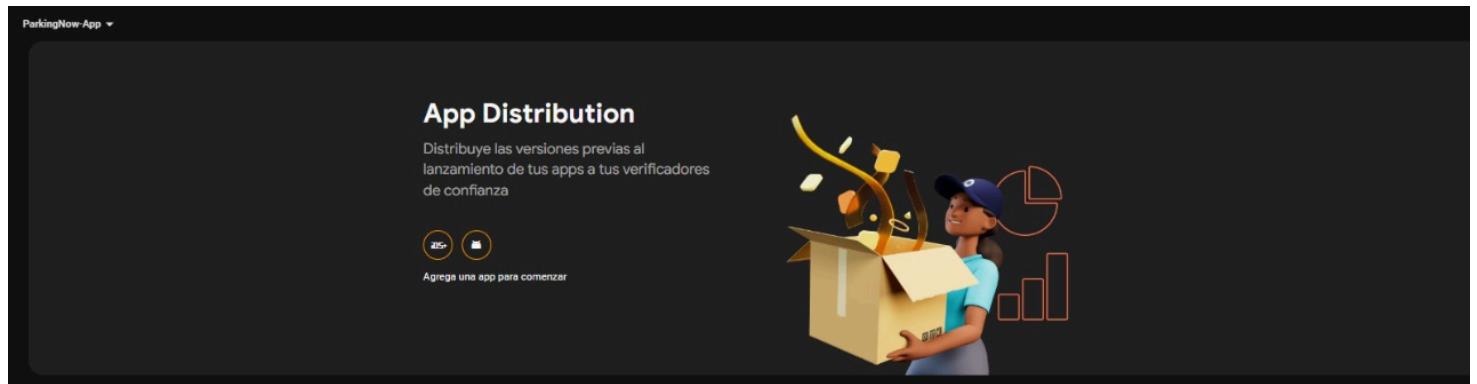
También puedes explorar las [apps de muestra de Firebase](#).

También puedes ir a consola para explorar Firebase

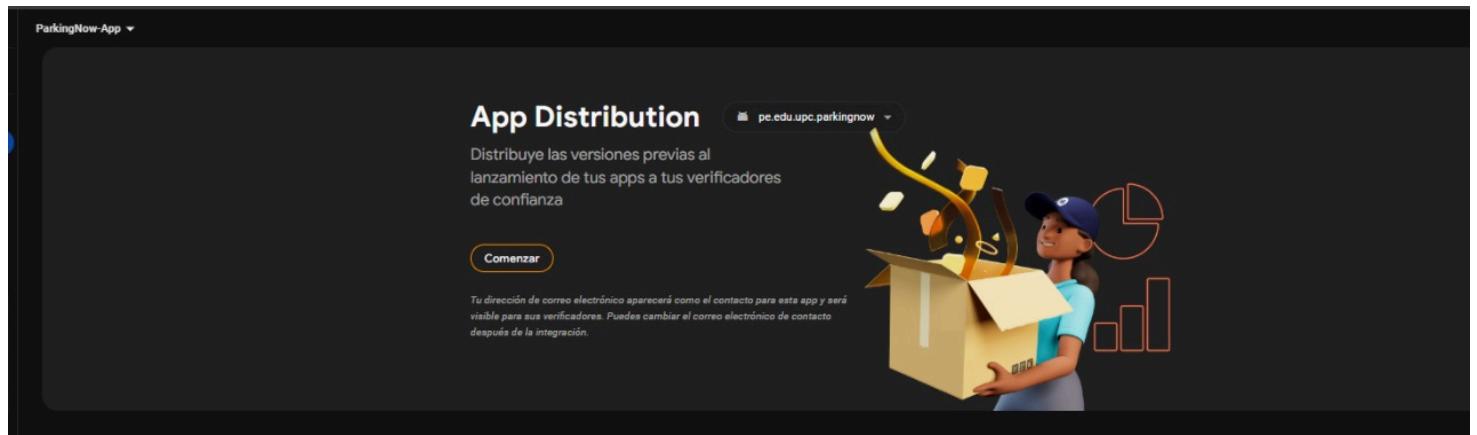
Anterior

Ir a la consola

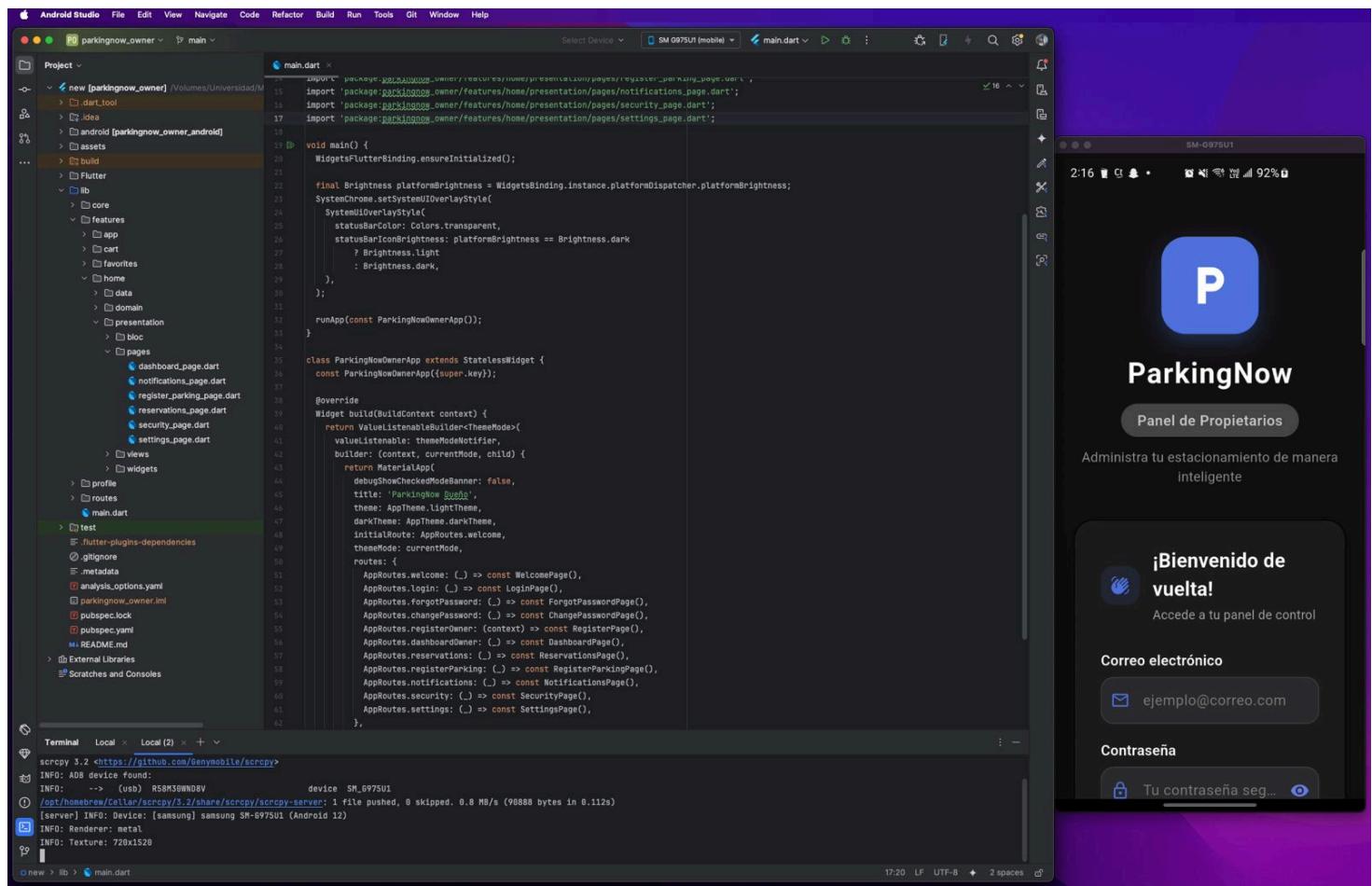
Seleccionamos el logo de android



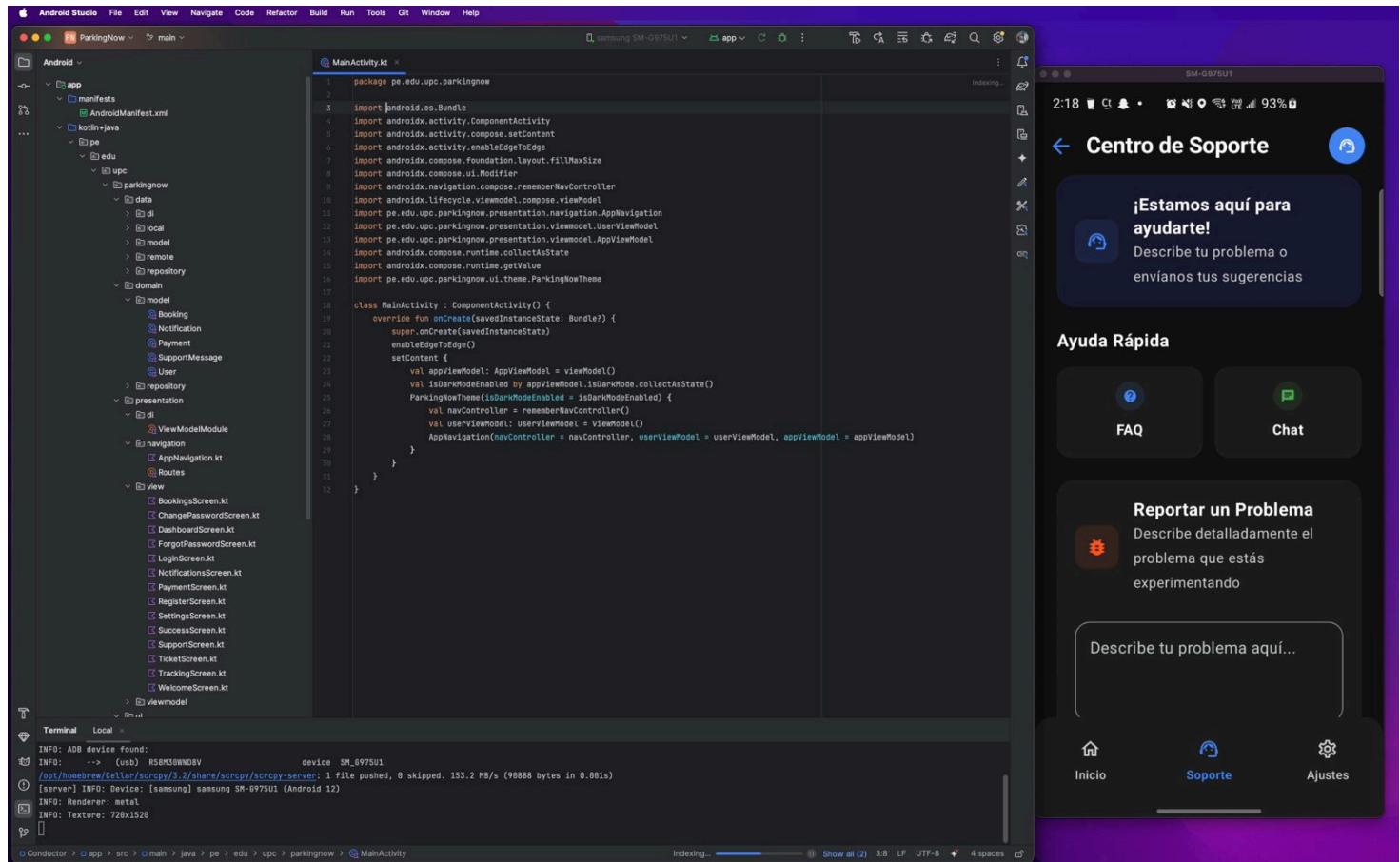
Le damos clic en comenzar



 Flutter – Segmento Objetivo: Dueño de Estacionamiento



◆ Kotlin – Segmento Objetivo: Conductor



Backend

Durante este Sprint desplegamos el backend e hicimos las pruebas requeridas

Inicio del despliegue

Empezamos logueandonos en la plataforma

```
 samue@DESKTOP-JPEIUCT MINGW64 /d/upc/Aplicaciones_Moviles/Backend (main)
● $ npm install --save @nestjs/cli
  up to date, audited 895 packages in 17s
  163 packages are looking for funding
    run `npm fund` for details
  1 low severity vulnerability
  To address all issues, run:
    npm audit fix
  Run `npm audit` for details.

 samue@DESKTOP-JPEIUCT MINGW64 /d/upc/Aplicaciones_Moviles/Backend (main)
● $ heroku login
heroku: Press any key to open up the browser to login or q to exit:
opening browser to https://cli-auth.herokuapp.com/auth/cli/browser/c2ebda91-4842-47c8-b8ca-4e1b729c9e44?requestor=SFMyNTY.g2g0bQAAAAsxNzkuNi4zLjIyNg4GACjK6Y-XAIIAVGA.wQuA0jvUEmrhi3t3vGVn7GY0xPq26Z90YRykHNqo8
Logging in... done
Logged in as samuelmolina281816@gmail.com
```

Configuracion de la base de datos

Configuramos una base de datos con NEON

```
 samue@DESKTOP-JPEIUCT MINGW64 /d/upc/Aplicaciones_Moviles/Backend (main)
● $ heroku config:set DATABASE_URL="postgresql://neondb_owner:npg_Ytv3go6FPQCM@ep-floral-shadow-a5w34cdf-pooler.us-east-2.aws.neon.tech/neondb?sslmode=require"
Setting DATABASE_URL and restarting ⚡ parkingnow-app... done, v3
DATABASE_URL: postgresql://neondb_owner:npg_Ytv3go6FPQCM@ep-floral-shadow-a5w34cdf-pooler.us-east-2.aws.neon.tech/neondb?sslmode=require

 samue@DESKTOP-JPEIUCT MINGW64 /d/upc/Aplicaciones_Moviles/Backend (main)
$ 
○
 samue@DESKTOP-JPEIUCT MINGW64 /d/upc/Aplicaciones_Moviles/Backend (main)
● $ git status
On branch main
Your branch is up to date with 'origin/main'.

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
    (use "git restore <file>..." to discard changes in working directory)
      modified:  package-lock.json
      modified:  package.json
      modified:  src/app.module.ts

 no changes added to commit (use "git add" and/or "git commit -a")
```

Cambios finales para producción

Se agregan cambios a producción para la ejecucion correcta de Nest en la plataforma

```
{} package.json > {} scripts
8   "scripts": {
9     "build": "nest build",
10    "format": "prettier --write \"src/**/*.ts\" \"test/**/*.ts\"",
11    "start": "node dist/main.js",
12    "start:dev": "nest start --watch",
13    "start:debug": "nest start --debug --watch",
14    "start:prod": "node dist/main",
15    "lint": "eslint \"{src,apps,libs,test}/**/*.{ts,json}",
16    "test": "jest",
17    "test:watch": "jest --watch",
18    "test:cov": "jest --coverage",
19    "test:debug": "node --inspect-brk -r tsconfig-paths/register -r ts-node/register node_modules/.bin/jest --runInBand",
20    "test:e2e": "jest --config ./test/jest-e2e.json"
21  },
22  "dependencies": {
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
samue@DESKTOP-JPEIUCT MINGW64 /d/upc/Aplicaciones Moviles/Backend (main)
● $ git status
On branch main
Your branch is up to date with 'origin/main'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    modified:   package-lock.json
    modified:   package.json
    modified:   src/app.module.ts

samue@DESKTOP-JPEIUCT MINGW64 /d/upc/Aplicaciones Moviles/Backend (main)
● $ git commit -m "Configuraciones para la conexion a produccion a Neon PostgreSQL"
[main 99663ca] Configuraciones para la conexion a produccion a Neon PostgreSQL
  3 files changed, 53 insertions(+), 10 deletions(-)

samue@DESKTOP-JPEIUCT MINGW64 /d/upc/Aplicaciones Moviles/Backend (main)
● $ git push heroku main
Enumerating objects: 84, done.
Counting objects: 100% (84/84), done.
Delta compression using up to 8 threads
Compressing objects: 100% (82/82), done.
Writing objects: 100% (84/84), 117.11 KiB | 3.90 MiB/s, done.
Total 84 (delta 37), reused 18 (delta 0), pack-reused 0 (from 0)
remote: Updated 29 paths from eb4b912
remote: Compressing source files... done.
remote: Building source:
remote:
remote: -----> Building on the Heroku-24 stack
remote: -----> Determining which buildpack to use for this app
remote: -----> Node.js app detected
remote:
```

Obtenemos el log del despliegue y comprobamos la conexión a la base de datos

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
remote:     npm notice Changelog: https://github.com/npm/cli/releases/tag/v11.4.2
remote:     npm notice To update run: npm install -g npm@11.4.2
remote:     npm notice
remote:
remote: ----> Build
remote:     Running build
remote:
remote: > backend-parkingnow@0.0.1 build
remote: > nest build
remote:
remote:
remote: ----> Caching build
remote:     - npm cache
remote:
remote: ----> Pruning devDependencies
remote:
remote:     up to date, audited 215 packages in 1s
remote:
remote:     47 packages are looking for funding
remote:       run `npm fund` for details
remote:
remote:     1 low severity vulnerability
remote:
remote:     To address all issues, run:
remote:       npm audit fix
remote:
remote:     Run `npm audit` for details.
remote:     npm notice
remote:     npm notice New major version of npm available! 10.9.2 -> 11.4.2
remote:     npm notice Changelog: https://github.com/npm/cli/releases/tag/v11.4.2
remote:     npm notice To update run: npm install -g npm@11.4.2
remote:     npm notice
remote:
remote: ----> Build succeeded!
remote: ----> Discovering process types
remote:     Procfile declares types    -> (none)
remote:     Default types for buildpack -> web
remote:
remote: ----> Compressing...
remote:     Done: 65M
remote: ----> Launching...
remote:     Released v4
remote:     https://parkingnow-app-963963f523b4.herokuapp.com/ deployed to Heroku
remote:
remote: Verifying deploy... done.
To https://git.heroku.com/parkingnow-app.git
 * [new branch]      main -> main

samue@DESKTOP-JPEIUCT MINGW64 /d/upc/Aplicaciones Moviles/Backend (main)
$ heroku config:get DATABASE_URL
postgresql://neondb_owner:npg_Ytv3go6FPQOM@ep-floral-shadow-a5w34cdf-pooler.us-east-2.aws.neon.tech/neondb?sslmode=require

samue@DESKTOP-JPEIUCT MINGW64 /d/upc/Aplicaciones Moviles/Backend (main)
$ heroku open
```

Comprobacion del despliegue

Revision de los logs de despliegue

```
samue@DESKTOP-JPEIUCT MINGW64 /d/upc/Aplicaciones Moviles/Backend (main)
$ heroku logs --tail
2025-06-21T00:40:20.199533+00:00 app[api]: Release v1 created by user samuelmolina281816@gmail.com
2025-06-21T00:40:20.199533+00:00 app[api]: Initial release by user samuelmolina281816@gmail.com
2025-06-21T00:40:20.402899+00:00 app[api]: Release v2 created by user samuelmolina281816@gmail.com
2025-06-21T00:40:20.402899+00:00 app[api]: Enable Logplex by user samuelmolina281816@gmail.com
2025-06-21T01:16:08.600619+00:00 app[api]: Release v3 created by user samuelmolina281816@gmail.com
2025-06-21T01:16:08.600619+00:00 app[api]: Set DATABASE_URL config vars by user samuelmolina281816@gmail.com
2025-06-21T01:18:25.000000+00:00 app[api]: Build started by user samuelmolina281816@gmail.com
2025-06-21T01:18:52.897281+00:00 app[api]: Deploy 99663ca6 by user samuelmolina281816@gmail.com
2025-06-21T01:18:52.897281+00:00 app[api]: Release v4 created by user samuelmolina281816@gmail.com
2025-06-21T01:18:52.909856+00:00 app[api]: Scaled to web@1:Basic by user samuelmolina281816@gmail.com
2025-06-21T01:18:55.000000+00:00 app[api]: Build succeeded
```

Sin conflicto, despliegue exitoso

Detalles de la aplicación en la plataforma

dashboard.heroku.com/apps/parkingnow-app

HEROKU from Salesforce

Overview Resources Deploy Metrics Activity Access Settings

Get a complete visualization of your app in a team-based continuous delivery environment with [Heroku Pipelines](#). [Hide](#) [Create a Heroku Pipeline](#)

Metrics (last 24hrs) [All Metrics](#)

Response Time
191ms

Throughput
< 1 rps

Memory
12%

Latest Activity [All Activity](#)

samuelmolina281816@gmail.com: Deployed `dd1a4b0d` Today at 8:24 PM - v5

samuelmolina281816@gmail.com: Build succeeded Today at 8:24 PM - [View build log](#)

samuelmolina281816@gmail.com: Deployed `99663ca6` Today at 8:18 PM - v4

samuelmolina281816@gmail.com: Build succeeded Today at 8:18 PM - [View build log](#)

samuelmolina281816@gmail.com: Set `DATABASE_URL` config var Today at 8:16 PM - v3

samuelmolina281816@gmail.com: Enable Logplex Today at 7:40 PM - v2

samuelmolina281816@gmail.com: Initial release Today at 7:40 PM - v1

Installed Add-ons [Configure Add-ons](#)

There are no add-ons for this app
You can add add-ons to this app and they will show here. [Learn more](#)

Dyno Formations [Configure Dynos](#)

This app is using basic dynos

web npm start **ON**

Collaborator Activity [Manage Access](#)

samuelmolina281816@gmail.com [2 deploys](#)

Testeo de la aplicación

Testeo de la aplicacion en producción

HTTP <http://localhost:3000/usuario> [Save](#)

POST <https://parkingnow-app-963963f523b4.herokuapp.com/usuario> [Send](#)

Params Authorization Headers (8) **Body** Pre-request Script Tests Settings

none form-data x-www-form-urlencoded raw binary JSON

```

1
2   "name": "Samuel Molina",
3   "email": "samuel@example.com",
4   "password": "Pa$$word963",
5   "placa": "AKL-963",
6   "dni": "12345678",
7   "ruc": "20123456789"
8

```

Body Cookies Headers (11) Test Results

Pretty Raw Preview Visualize JSON

```

1
2   "id": 2,
3   "name": "Samuel Molina",
4   "email": "samuel@example.com",
5   "password": "Pa$$word963",
6   "placa": "AKL-963",
7   "dni": "12345678",
8   "ruc": "20123456789",
9   "tipoUsuario": "dueno_estacionamiento"
10

```

Status: 201 Created Time: 306 ms Size: 933 B Save Response

Verificamos la creacion de registros en la base de datos de producción

The screenshot shows the Neon SQL Editor interface. On the left, there's a sidebar with sections for PROJECT (Dashboard, Branches, Integrations, Auth, Settings), and BRANCH (production). The main area is titled "SQL Editor" with tabs for Untitled, Save, Primary (ACTIVE), neondb, and a preview icon. Below the tabs, there are two saved queries: "retrieve all user data" (Jun 20, 2025 - 8:36pm) and "list all users" (Jun 20, 2025 - 8:12pm). The current query in the editor is:

```
1 /*
2 In Neon, databases are stored on branches. By default, a project has one branch and one database.
3 You can select the branch and database to use from the drop-down menus above.
4
5 Try generating sample data and querying it by running the example statements below, or click
6 New Query to clear the editor.
7 */
8 select * from usuario;
```

Below the editor, a results table is displayed with the following data:

#	id	name	email	password	placa	dni	ruc	tipo_usuario
1	1	Diego Ulises	diego@example.com	123456	ABC-123	12345678	20123456789	dueno_estacionamiento
2	2	Samuel Molina	samuel@example.com	Pa\$\$word963	AKL-963	12345678	20123456789	dueno_estacionamiento

At the bottom right of the results table, it says "152ms 2 rows".

Comprobacion de las apis abieretas

The screenshot shows a browser developer tools Network tab. The URL in the address bar is "parkingnow-app-963963f523b4.herokuapp.com/usuario". The response body contains JSON data representing two user records:

```
{
  "id": 1,
  "name": "Diego Ulises",
  "email": "diego@example.com",
  "password": "123456",
  "placa": "ABC-123",
  "dni": "12345678",
  "ruc": "20123456789",
  "tipoUsuario": "dueno_estacionamiento"
},
{
  "id": 2,
  "name": "Samuel Molina",
  "email": "samuel@example.com",
  "password": "Pa$$word963",
  "placa": "AKL-963",
  "dni": "12345678",
  "ruc": "20123456789",
  "tipoUsuario": "dueno_estacionamiento"
}
```

Link base de la aplicación: <https://parkingnow-app-963963f523b4.herokuapp.com/>

Endpoint obtencion de usuarios: <https://parkingnow-app-963963f523b4.herokuapp.com/usuario>

Testeo de la aplicacion en producción

The screenshot shows the Postman interface with a successful POST request to `https://parkingnow-app-963963f523b4.herokuapp.com/usuario`. The response status is `201 Created`, time is `306 ms`, and size is `933 B`. The request body is a JSON object representing a user:

```

1  [
2    "name": "Samuel Molina",
3    "email": "samuel@example.com",
4    "password": "Pa$$word963",
5    "placa": "AKL-963",
6    "dni": "12345678",
7    "ruc": "20123456789"
8  ]

```

The response body is also a JSON object with the same user data and an additional field `"tipoUsuario": "dueno_estacionamiento"`:

```

1  [
2    "id": 2,
3    "name": "Samuel Molina",
4    "email": "samuel@example.com",
5    "password": "Pa$$word963",
6    "placa": "AKL-963",
7    "dni": "12345678",
8    "ruc": "20123456789",
9    "tipoUsuario": "dueno_estacionamiento"
10   ]

```

Verificamos la creacion de registros en la base de datos de producción

The screenshot shows the Neon SQL Editor interface. It displays two queries:

- `retrieve all user data` (Jun 20, 2025 - 8:36pm)
- `list all users` (Jun 20, 2025 - 8:12pm)

The results of the second query are shown in a table:

#	id	name	email	password	placa	dni	ruc	tipo_usuario
1	1	Diego Ulises	diego@example.com	123456	ABC-123	12345678	20123456789	dueno_estacionamiento
2	2	Samuel Molina	samuel@example.com	Pa\$\$word963	AKL-963	12345678	20123456789	dueno_estacionamiento

Comprobacion de las apis abieretas

6.2.2.8. Team Collaboration Insights during Sprint

Durante este Sprint, el equipo trabajó de manera colaborativa para lograr el desarrollo completo de los tres productos digitales: la Landing Page, el aplicativo móvil y los Web Services. A continuación, se detallan los aportes por cada sección, incluyendo espacio para evidencias de commits y capturas relevantes.

Reporte



Landing Page

Showing most recent first ▾

- Fix embed YouTube videos**
Diego2005z pushed 1 commit to `main` · 0d46d4e...15a4d98 · 2 hours ago
- Fix i18n translation issues and update HTML entities**
Diego2005z pushed 1 commit to `main` · 82a13ed...0d46d4e · yesterday
- Add landing page with updated README and styles** (Force push)
Diego2005z force pushed to `main` · 41d1e52...82a13ed · 2 days ago
- Initial commit**
Diego2005z created `main` · 41d1e52 · 2 days ago

Kotlin



Flutter

Activity

All branches	All activity	All users	All time	Showing most recent first
Add APK	...			
JuanCali999 created feature/register-parking · 2efa9fe · 23 hours ago	...			
Add APK	...			
JuanCali999 created develop · 2efa9fe · 23 hours ago	...			
Add APK	...			
Diego2005z pushed 1 commit to main · b694f82...2efa9fe · 9 days ago	...			
fix: update launcher icon configuration and generate icons	...			
Diego2005z pushed 1 commit to main · c42bce...b694f82 · 11 days ago	...			
Fix: default theme mode now respects system setting	...			
Diego2005z pushed 1 commit to main · b4aa3de...c42bce · 11 days ago	...			
Fix: full dark mode applied in Security UI	...			
Diego2005z pushed 1 commit to main · 7920a4b...b4aa3de · 11 days ago	...			
Clean up alert cards and remove subscription dialog	...			
Diego2005z pushed 1 commit to main · 22524bb...7920a4b · 12 days ago	...			

Backend



6.2.3. Sprint 3

En este sprint se desarrollaron las **US34** a **US50**, orientadas al cierre funcional de la aplicación, reforzando la experiencia de usuario y completando flujos esenciales como generación de tickets, pago en línea, confirmación de reservas, visualización de tickets activos y finalizados, así como validación de entrevistas.

En el segmento del **conductor**, se implementaron mejoras visuales y técnicas en las pantallas de pagos, tickets y confirmación, asegurando la integración con el backend y la experiencia fluida dentro de la app.

Simultáneamente, en el segmento del **dueño de estacionamiento**, se integraron nuevos módulos como gestión de reservas, validación de espacios en tiempo real y ajustes visuales en la interfaz del dashboard, cumpliendo con los requisitos finales definidos por el equipo de diseño y funcionalidad.

Estas implementaciones consolidaron la versión funcional de ParkingNow para su validación completa, permitiendo una experiencia fluida entre los distintos perfiles de usuario de la plataforma.

6.2.3.1 Sprint Planning 3

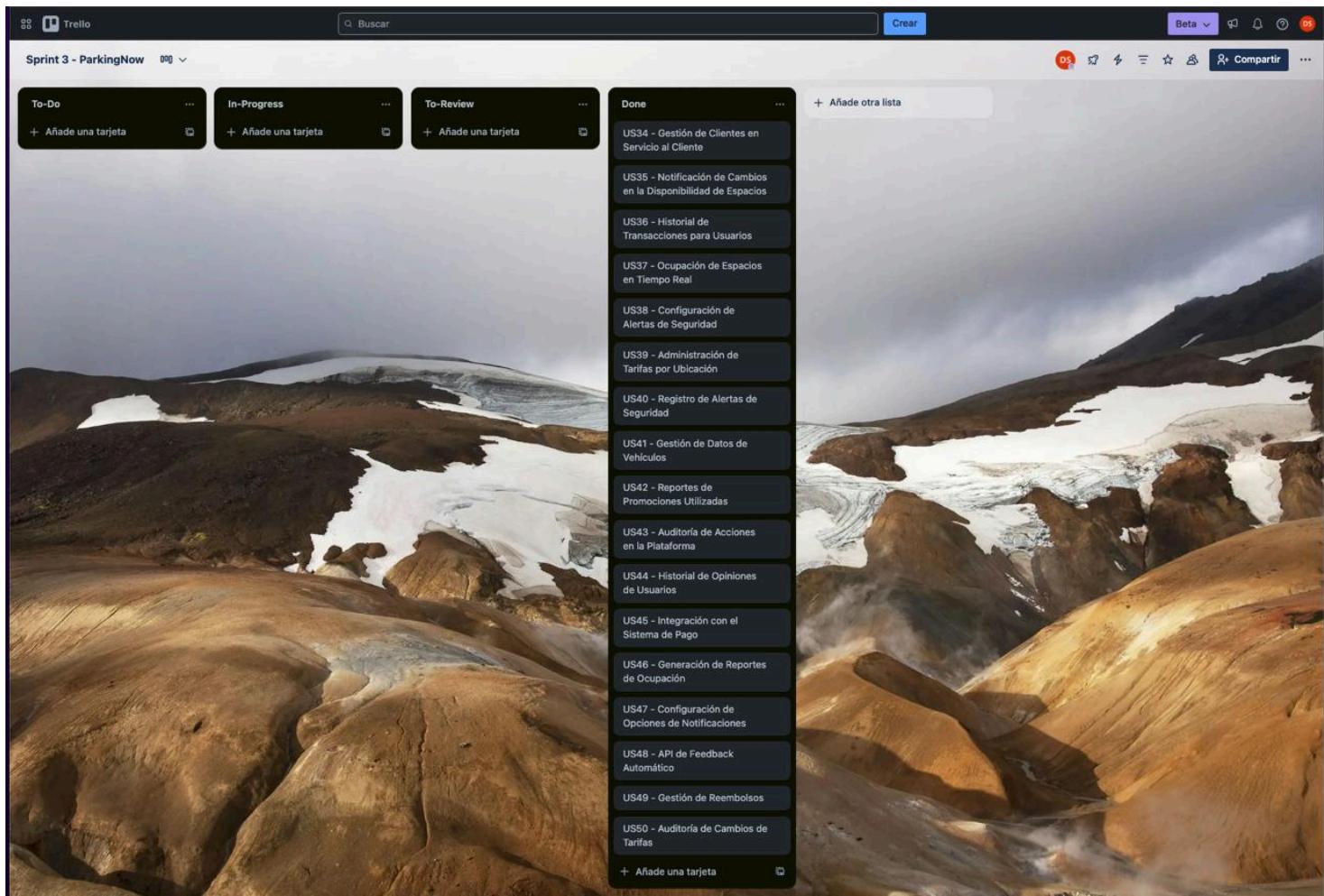
Sprint #	Sprint 3
Sprint Planning Background	
Date	2025-06-30

Sprint #	Sprint 3
Time	17:00 horas (GMT-5)
Location	Modalidad remota a través de Discord
Prepared By	Soto Quispe, Diego Ulises
Attendees (to planning meeting)	Calisaya Sánchez, Juan Jesús / Hidalgo Lopez, Mathias Adriano / Molina Asencios, Samuel Elias / Soto Quispe, Diego Ulises
Sprint n – 2 Review Summary	En el Sprint 2 se desarrollaron las funcionalidades principales del segmento conductor, incluyendo reservas, historial, pagos, notificaciones, mapa, seguimiento en tiempo real y soporte. Paralelamente, se avanzó en el segmento dueño con vistas iniciales, dashboard y consultas sobre disponibilidad.
Sprint n – 2 Retrospective Summary	Se resaltó la mejora en la distribución de tareas y documentación. Se acordó mantener validaciones funcionales constantes, usar mejores prácticas de integración y mantener sincronía en los módulos frontend-backend.
Sprint Goal & User Stories	
Sprint 3 Goal	Finalizar y consolidar todos los módulos funcionales de conductor y dueño de estacionamiento: generación y visualización de tickets, sistema de pagos, validación de reservas, despliegue local, pruebas integradas, mejoras visuales y colaborativas.
Sprint 3 Velocity	130
Sum of Story Points	122

6.2.3.2. Sprint Backlog 3

User Story	Title	Work-Item ID	Work-Item Title	Description	Estimation (Hours)	Assigned To	Status
US34	Gestión de Clientes en Servicio al Cliente	WI-34	Endpoint para consultas y solicitudes	Crear endpoint /api/client-service para gestionar y notificar solicitudes al equipo de servicio.	3	Samuel Molina	Done
US35	Notificación de Cambios en la Disponibilidad de Espacios	WI-35	Endpoint de notificaciones	Endpoint /api/space-notifications que envíe notificaciones push y email cuando cambie el estado del espacio.	5	Juan Calisaya	Done
US36	Historial de Transacciones para Usuarios	WI-36	Endpoint historial de transacciones	Endpoint /api/user-transactions para listar y consultar pagos y reservas pasadas por usuario o ID.	3	Diego Soto	Done
US37	Ocupación de Espacios en Tiempo Real	WI-37	Actualización con cámaras	Endpoint /api/space-occupancy que actualice estado según cámaras de monitoreo visual y genere alertas por ocupación sin reserva.	5	Mathias Hidalgo	Done
US38	Configuración de Alertas de Seguridad	WI-38	Endpoint configuración alertas	Endpoint /api/security-alerts para definir parámetros (hora, ubicación, tipo) y enviar notificaciones ante eventos sospechosos.	5	Samuel Molina	Done
US39	Administración de Tarifas por Ubicación	WI-39	Endpoint tarifas por ciudad	Endpoint /api/parking-rates para configurar tarifas de estacionamiento por zona y actualizar automáticamente precios.	5	Juan Calisaya	Done
US40	Registro de Alertas de Seguridad	WI-40	Registro de eventos de	Endpoint /api/security-log que almacene fecha, hora y detalles de incidentes	3	Diego Soto	Done

User Story	Title	Work-Item ID	Work-Item Title	Description	Estimation (Hours)	Assigned To	Status
			seguridad	registrados para ser revisados por el administrador.			
US41	Gestión de Datos de Vehículos	WI-41	Endpoint gestión de vehículos	Endpoint /api/vehicle-management para registrar, editar o eliminar información de vehículos vinculados a usuarios.	3	Mathias Hidalgo	Done
US42	Reportes de Promociones Utilizadas	WI-42	Reportes de promociones	Endpoint /api/promotion-reports para generar reportes de promociones usadas y filtrar por fecha y ubicación.	3	Samuel Molina	Done
US43	Auditoría de Acciones en la Plataforma	WI-43	Auditoría de acciones	Endpoint /api/audit-log para registrar acciones críticas y permitir filtros por tipo de acción o usuario.	5	Juan Calisaya	Done
US44	Historial de Opiniones de Usuarios	WI-44	Opiniones de usuarios	Endpoint /api/user-reviews para registrar opiniones con calificación y comentario, y listar cronológicamente para revisión.	3	Diego Soto	Done
US45	Integración con el Sistema de Pago	WI-45	Endpoint pagos y verificación	Endpoint /api/payment-processing para procesar pagos con respuesta de éxito/error y manejo de fallos.	5	Mathias Hidalgo	Done
US46	Generación de Reportes de Ocupación	WI-46	Reportes de ocupación por ubicación	Endpoint /api/occupancy-reports para ver porcentaje de ocupación y análisis de tendencias por períodos.	3	Samuel Molina	Done
US47	Configuración de Opciones de Notificaciones	WI-47	Preferencias de notificación	Endpoint /api/notification-preferences para que el usuario active/desactive tipos de notificación desde su perfil.	3	Juan Calisaya	Done
US48	API de Feedback Automático	WI-48	Endpoint feedback post-reserva	Endpoint /api/automatic-feedback para almacenar comentarios post-reserva y generar estadísticas para administrador.	3	Diego Soto	Done
US49	Gestión de Reembolsos	WI-49	Endpoint reembolsos	Endpoint /api/refunds para procesar solicitudes de reembolso y consultar historial por parte del administrador.	3	Mathias Hidalgo	Done
US50	Auditoría de Cambios de Tarifas	WI-50	Auditoría de tarifas	Endpoint /api/rate-audit para auditar modificaciones de tarifas con fecha, hora y cambios realizados.	8	Samuel Molina	Done



Trello: <https://trello.com/invite/b/6862f6cec946d682de2ee8fb/ATTI93acc70cafbc864a63510ce995133ee5F54112B0/sprint-3-parkingnow>

6.2.3.3 Development Evidence for Sprint Review

Durante el tercer sprint se implementaron las historias US34 a US50, abarcando funcionalidades clave como gestión de usuarios, reservas, tarifas, alertas y auditorías. Se desarrollaron e integraron endpoints backend, y se completaron las pantallas restantes en las apps de Flutter (dueño) y Kotlin (conductor).

Como entrega final, se verificó el funcionamiento de todo el sistema: backend, aplicaciones móviles y la landing page, asegurando una experiencia de usuario fluida y coherente.

◆ Kotlin – Segmento Objetivo: Conductor

ID Commit ID	Repository	Rama	Autor	Fecha	Mensaje de Commit
5bf4755	Kotlin-Mobile	main	JuanCali999	11/07/2025	add active reservations modal and integrate with reservation view model
284e4d4	Kotlin-Mobile	main	JuanCali999	11/07/2025	add reservation functionality with API integration and update user data handling
7cbb92a	Kotlin-Mobile	main	JuanCali999	09/07/2025	implement user login and registration functionality with Firebase integration
7a88dbc	Kotlin-Mobile	main	mathiasadriano	20/06/2025	Update colors.xml
4152f23	Kotlin-Mobile	main	mathiasadriano	20/06/2025	Update AndroidManifest.xml
cc2e1ef	Kotlin-Mobile	main	mathiasadriano	20/06/2025	edit gradle.properties

ID Commit ID	Repository	Rama	Autor	Fecha	Mensaje de Commit
055bb79	Kotlin-Mobile	main	mathiasadriano	20/06/2025	Update README.md
e398f27	Kotlin-Mobile	main	mathiasadriano	20/06/2025	feat:edit kotlin
ccbe57a	Kotlin-Mobile	main	JuanCali999	18/06/2025	feat: implement user registration flow with Retrofit and update API structure
db38a28	Kotlin-Mobile	main	JuanCali999	18/06/2025	feat: configure network security and update API base URL for emulator
f362356	Kotlin-Mobile	main	JuanCali999	18/06/2025	feat: implement user registration API integration
49de69c	Kotlin-Mobile	main	JuanCali999	18/06/2025	feat: add Retrofit and Gson converter dependencies
9e9c8ad	Kotlin-Mobile	main	JuanCali999	18/06/2025	Merge branch 'main' of https://github.com/G3-UPC-CC238-346-ParkingNow/Kotlin-Mobile into feature/consume-user-api
0844608	Kotlin-Mobile	main	Diego Soto	11/06/2025	Add GPS activation prompt to DashboardScreen
6b984a	Kotlin-Mobile	main	Diego Soto	11/06/2025	Add debug APK for testing
2d5deb4	Kotlin-Mobile	main	Diego Soto	09/06/2025	update app icon alignment
00dc823	Kotlin-Mobile	main	Diego Soto	09/06/2025	Adjust FAB position on DashboardScreen for better alignment with top bar
7a26d28	Kotlin-Mobile	main	Diego Soto	04/06/2025	Update LoginScreen: remove social login and enable guest mode navigation
ac81138	Kotlin-Mobile	main	Diego Soto	04/06/2025	polish UI for password update and validation flow
d61546e	Kotlin-Mobile	main	Diego Soto	30/05/2025	Add custom app icon for ParkingNow
c1b1114	Kotlin-Mobile	main	Diego Soto	30/05/2025	Fix: Apply global dark mode using AppViewModel
da42c27	Kotlin-Mobile	main	Diego Soto	30/05/2025	Fix bottom navigation layout on Tracking screen to match Support screen style
4db18aa	Kotlin-Mobile	main	Diego Soto	29/05/2025	Fix TicketScreen background and remove unused imports
2e2a736	Kotlin-Mobile	main	Diego Soto	29/05/2025	Update: improved booking screen layout and visibility
4c88466	Kotlin-Mobile	main	Diego Soto	29/05/2025	Refactor UI: adjusted layout spacing for favorites and reservation button visibility
025323e	Kotlin-Mobile	main	Diego Soto	29/05/2025	Fix navigation: ensure safe routes and prevent crashes on back navigation
e84797c	Kotlin-Mobile	main	Diego Soto	29/05/2025	Show real-time user location with car icon in Dashboard map
d3231a4	Kotlin-Mobile	main	Diego Soto	28/05/2025	Enhance NotificationsScreen UI with back navigation and layout adjustments
3266b1c	Kotlin-Mobile	main	Diego Soto	28/05/2025	Feature: Improved ticket UI and payment method handling
abfdc25	Kotlin-Mobile	main	Diego Soto	27/05/2025	Smooth welcome transition with fade and scale animation
b403f37	Kotlin-Mobile	main	Diego Soto	26/05/2025	Adjust map position at bottom of dashboard
2ffcc0b	Kotlin-Mobile	main	Diego Soto	26/05/2025	Show registered username in Dashboard
90d538d	Kotlin-Mobile	main	Diego Soto	26/05/2025	Add NotificationsScreen and complete all routing
fa34cc2	Kotlin-Mobile	main	Diego Soto	26/05/2025	Add full payment flow and ticket generation screen
e683b14	Kotlin-Mobile	main	Diego Soto	25/05/2025	Add real-time user location to map in Dashboard

● Flutter – Segmento Objetivo: Dueño de Estacionamiento

ID Commit ID	getRepository	Rama	Autor	Fecha	Mensaje de Commit
aef7579	Flutter	main	JuanCali999	11/07/2025	implement registration functionality for parking owners and locals
361190a	Flutter	main	samuelmolina281816	01/07/2025	CAmbios para usar el backend
7c950c5	Flutter	feature/register-parking	mathiasadriano	20/06/2025	feat: edit api_constants
9fde524	Flutter	feature/register-parking	JuanCali999	20/06/2025	feat: implement registration functionality for parking owners and locals
2efa9fe	Flutter	main	Diego Soto	11/06/2025	Add APK
b694f82	Flutter	main	Diego Soto	09/06/2025	update launcher icon configuration and generate icons
c42bece	Flutter	main	Diego Soto	09/06/2025	Fix: default theme mode now respects system setting
b4aa3de	Flutter	main	Diego Soto	09/06/2025	Fix: full dark mode applied in Security UI
7920a4b	Flutter	main	Diego Soto	08/06/2025	Clean up alert cards and remove subscription dialog
22524bb	Flutter	main	Diego Soto	08/06/2025	Fix: remove duplicate ChangePasswordPage from settings
19ee43a	Flutter	main	Diego Soto	08/06/2025	Fix: initialize locale for date formatting in history
74a9c3e	Flutter	main	Diego Soto	08/06/2025	remove settings from quick actions
1493af1	Flutter	main	Diego Soto	08/06/2025	Improve reservation popup UI
d3e5d2f	Flutter	main	Diego Soto	07/06/2025	Improve step 4 summary card and success pop-up design
ec090f0	Flutter	main	Diego Soto	07/06/2025	Improve drawer header spacing and fix overflow
eaddcd1	Flutter	main	Diego Soto	07/06/2025	Fix: remove tab transition animation
001d355	Flutter	main	Diego Soto	07/06/2025	update color styles
92aabcc2	Flutter	main	Diego Soto	07/06/2025	remove fade animation from login
f496ec8	Flutter	main	Diego Soto	07/06/2025	update
83599e0	Flutter	main	Diego Soto	07/06/2025	Fix UI bugs: remove missing image and update section titles
5c9f139	Flutter	main	Diego Soto	06/06/2025	Fix: restrict RUC input to numbers only
09b465b	Flutter	main	Diego Soto	05/06/2025	Feature: first commit from external drive

● Backend – Web Services

ID Commit ID	getRepository	Rama	Autor	Fecha	Mensaje de Commit
743221b	Backend	main	JuanCali999	11/07/2025	reserva: enhance active reservations retrieval with detailed logging and state filtering
b5571c8	Backend	main	JuanCali999	10/07/2025	add reserva

ID	Commit ID	Repository	Rama	Autor	Fecha	Mensaje de Commit
d4672ca	Backend	main	JuanCali999		10/07/2025	api: add comprehensive testing guide for expanded locales API
57fce54	Backend	main	JuanCali999		09/07/2025	fix: correct secretOrKey retrieval in JwtStrategy
874bb2c	Backend	main	JuanCali999		09/07/2025	add auth-jwt
1375c52	Backend	main	JuanCali999		09/07/2025	chore: update pg
1335fc3	Backend	main	samuelmolina281816		01/07/2025	Endpoint para logueo
dd1a4b0	Backend	main	samuelmolina281816		20/06/2025	Correcciones para la ejecucion de nest en produccion
99663ca	Backend	main	samuelmolina281816		20/06/2025	Configuraciones para la conexion a produccion a Neon PostgreSQL
f83243a	Backend	main	samuelmolina201016		17/06/2025	fix adding new endpoint for manage parking available
5392d32	Backend	main	samuelmolina201016		17/06/2025	fix adding user manage guide
88b445a	Backend	main	samuelmolina201016		17/06/2025	fix adding API for get all reserves of a Local - manage parking plance
e483b6e	Backend	main	samuelmolina201016		17/06/2025	fix Adding Api's guide
33919ff	Backend	main	samuelmolina201016		16/05/2025	feat(documentation): Documentation for endpoints
38224ffe	Backend	main	samuelmolina201016		16/05/2025	feat(usuario): Adding API for Usuario Managment
a171b92	Backend	main	samuelmolina201016		16/05/2025	feat(reserva): Adding API for reserva managment
a2b3c38	Backend	main	samuelmolina201016		16/05/2025	feat(local): Adding API for Local managment
d4233bd	Backend	main	samuelmolina201016		16/05/2025	feat(app): Adding Services and Controllers
f8b795f	Backend	main	samuelmolina201016		16/05/2025	fix(core): Adding dependencies
f5976b8	Backend	main	JuanCali999		14/05/2025	feat: Initialize NestJS project with default structure
a218916	Backend	main	Diego2005z		10/05/2025	Initial commit

Landing Page

ID	Commit ID	Repository	Rama	Autor	Fecha	Mensaje de Commit
1594898		Landing-ParkingNow	main	Diego2005z	20/06/2025	Fix: embed YouTube videos
8d4a6be		Landing-ParkingNow	main	Diego2005z	19/06/2025	Fix i18n translation issues and update HTML entities
82d13ed		Landing-ParkingNow	main	Diego2005z	18/06/2025	Add landing page with updated README and styles

6.2.3.4 Testing Suite Evidence for Sprint Review

Durante este sprint, se desarrollaron pruebas unitarias e integradas para los endpoints construidos con **NestJS**, el framework de Node.js utilizado en el backend del proyecto **ParkingNow**. Se emplearon herramientas como **Jest** (para lógica de negocio, servicios y controladores) y **Postman/Newman** (para pruebas de APIs REST).

Las pruebas incluyeron:

- Verificación de rutas y códigos de respuesta esperados (200 OK , 404 Not Found , 500 Internal Server Error).
- Validación de estructuras de datos de entrada y salida (DTOs).
- Comprobación del manejo de errores y excepciones en controladores y servicios.
- Evaluación de casos límite: parámetros inválidos, datos inexistentes o duplicados.

Todos los casos críticos fueron superados exitosamente, lo cual garantiza la estabilidad y correcto funcionamiento del backend en los entornos de desarrollo y pruebas.

Repository	Branch	Author	Message	Date	Link
Backend	main	Samuel Molina	feat(documentation) Documentation for endpoints	16/05/2025	 https://github.com/G3-UPC-CC238-346-ParkingNow/Backend

6.2.3.5 Execution Evidence for Sprint Review

Cada integrante realizó commits asociados a las historias correspondientes del Sprint 3.

Se usaron convenciones claras en los mensajes (`feat` , `fix` , `test` , etc.) y ramas específicas por funcionalidad (`feature/USxx-nombre`).

Los commits contienen:

- Implementación completa del endpoint.
- Documentación embebida en código.
- Pruebas locales realizadas.
- Refactorizaciones cuando fue necesario.

Se consolidó la rama principal (`develop`) al finalizar el sprint con todos los cambios revisados y funcionales.

💡 Segmento Objetivo: Conductor – Kotlin, Jetpack Compose y Android Studio

➡️ Pantalla de Inicio de Sesión – Login del Conductor



ParkingNow

Encuentra el mejor estacionamiento



¡Bienvenido de vuelta!

Inicia sesión en tu cuenta



Correo electrónico



Contraseña



?

¿Olvidaste tu contraseña?

→ Iniciar Sesión

0



Pantalla de Recuperación – ¿Olvidaste tu Contraseña?



¿Olvidaste tu contraseña?

Ingresa tu correo y te enviaremos un enlace para restablecerla



Correo electrónico

Enviar enlace de recuperación

[Volver al inicio de sesión](#)



Pantalla de Cambio de Contraseña – Validación y Seguridad



Cambiar Contraseña

Crea una nueva contraseña segura
para proteger tu cuenta



Nueva
contraseña



Confirmar
contraseña



Requisitos de contraseña:

- ✖ Al menos 8 caracteres
- ✖ Al menos una letra mayúscula

 **Registro de Nuevo Usuario – Creación de Cuenta para Conductores**

← Crear Cuenta



¡Únete a ParkingNow!

Crea tu cuenta y encuentra estacionamiento fácilmente



Información Personal



Nombre completo



Correo electrónico



Seguridad



Contraseña



Confirmar
contraseña



 **Dashboard del Conductor – Vista General de Actividades**

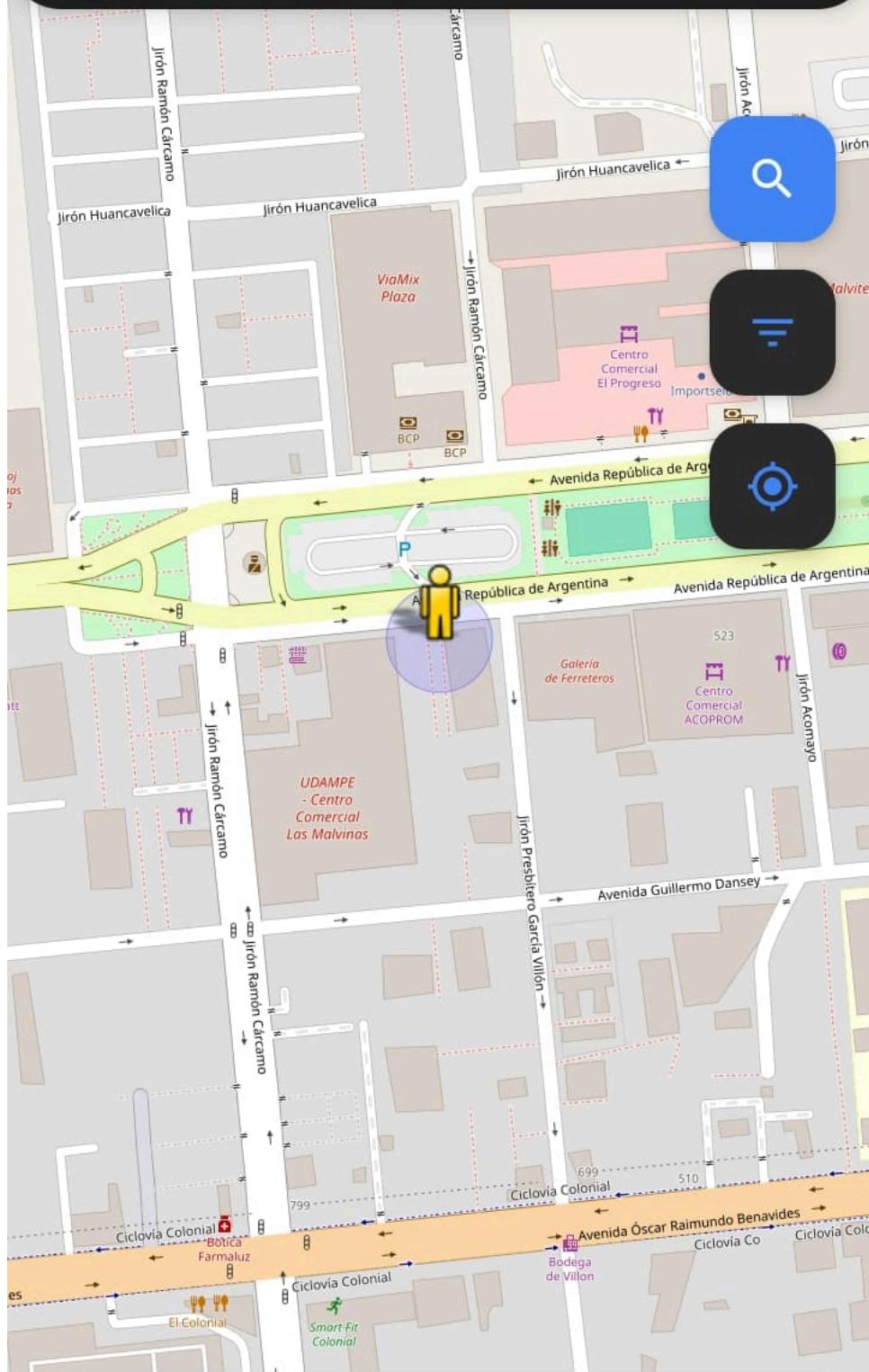
1:09 Ct • VoLTE 56%

Dashboard Conductor



D

Lima



Efectos e impactos Cercanos

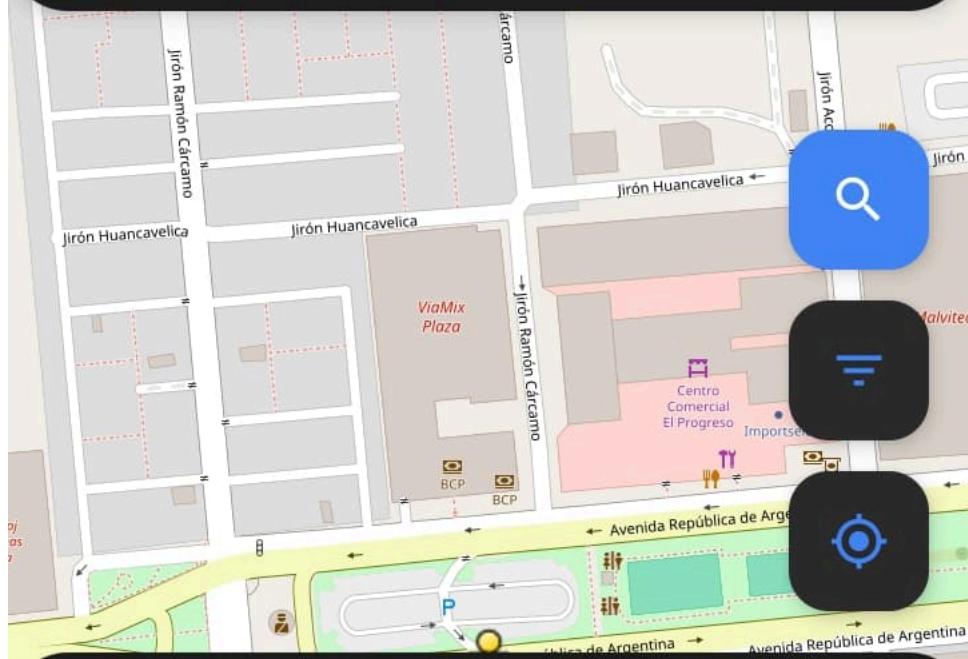
1:10 Ct • VoLTE 56%

Dashboard Conductor



D

Lima



Estacionamientos Cercanos

3 lugares disponibles

Todos

Más cercano

Más barato

Mejor valorado

Centro Comercial Jockey S/ 5.00/hora

Plaza

★ 4.5

Av. Javier Prado Este 4200,

Santiago de Surco

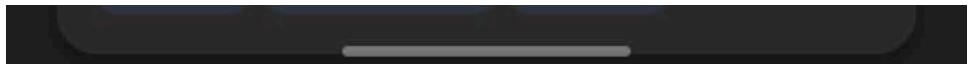
45/200 disponibles

0.8 km

Techado

Seguridad 24h

Cámaras



 Barra Lateral – Navegación con Opciones del Módulo Conductor

1:10 Ct • VoLTE 56%

Hola,

D

**Diego Ulises Soto
Quispe**

Lima, Provincia de Lima

 **Inicio**

 **Reservas**

 **Soporte**

 **Seguimiento**

 **Configuración**

 **Notificación**

 **Cerrar Sesión**

D





 **Historial de Reservas – Visualización de Reservas Activas y Pasadas**

← Mis Reservas



Reservas Activas

Gestiona tus reservas actuales

[Ver reservas activas](#)



Nueva Reserva

Encuentra y reserva un estacionamiento

[Realizar reserva](#)



Historial

Revisa tus reservas anteriores

[Ver historial](#)



Inicio



Reservas



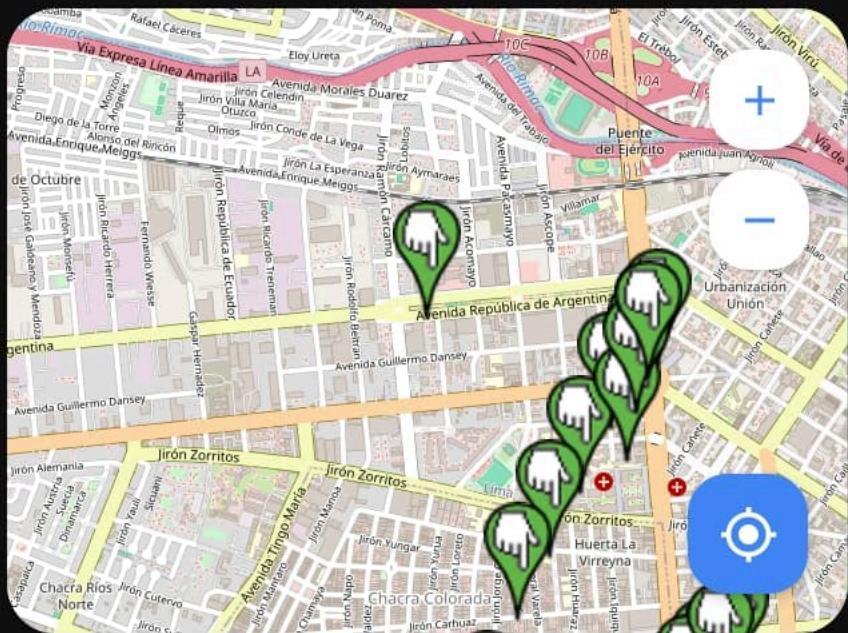
Ajustes



Reservar Estacionamiento – Selección de Espacio Cercano Según Ubicación

← Seleccionar Estacionamiento

Ubicación



Buscar estacionamiento



Estacionamientos Cercanos Ver todos

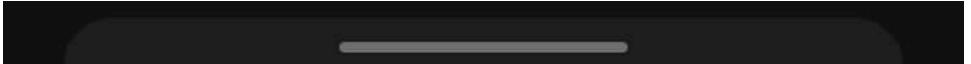
Real Plaza Salaverry

★ 4.5

📍 0.8 km

S/ 5.00 / hora P 12 lugares disponibles

Seleccionado ✓



 Método de Pago – Selección de Tarjeta y Confirmación de Transacción

← Métodos de Pago Guardados

No hay métodos guardados.

Pagar con Nuevo Método

Visa

Yape

Card Number



Card Holder Name



Expiry Date (MM/YY)



CVV

Guardar

Pagar con este método

Seleccione un método para pagar.

Código QR – Generación de Código para Validar la Reserva

1:10 Ct •

VoLTE 56%

Parking Larcomar

Malecón de la Reserva, Miraflores

VEHICLE

2021 Audi Q3 • B 1234 CD

DURATION

1 hora • 26 May. 2025

Slot A01



ID: PN-843043

Navegar

 Confirmación de Reserva – Mensaje de Éxito tras la Transacción

1:10 Ct •

57%



Success!

Yey, 1 parking slot
already booked for you.



 Soporte Técnico – Envío de Consultas y Comentarios desde el Dashboard

← Centro de Soporte



¡Estamos aquí para
ayudarte!



Describe tu problema o
envíanos tus sugerencias

Ayuda Rápida



FAQ



Chat

Reportar un Problema



Describe detalladamente el
problema que estás
experimentando

Describe tu problema aquí...



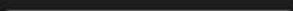
Inicio



Soporte



Ajustes



Seguimiento del Vehículo – Tracking en Tiempo Real del Espacio Reservado

← Seguimiento



Estado del Sistema

Monitoreo activo



2

Vehículos



4

Cámaras



0

Alertas

Vehículos Registrados

Toyota Corolla

Placa: ABC-123



Real Plaza Salaverry -

Estacionado

Nivel 2

Hace 5 minutos

Honda Civic

Placa: XYZ-789



Av. Javier Prado Este

En tránsito

Hace 2 minutos



Inicio



Seguimiento



Ajustes



Configuración del Sistema – Preferencias de Usuario y Notificaciones

← Configuración



ParkingNow

Versión 1.0.0

Cuenta y Seguridad



Privacidad y políticas



Pedir reembolso



Registrar robo de carro



Eliminar cuenta



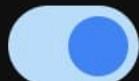
Preferencias



Notificaciones



Modo oscuro



Idioma

Español →

 Centro de Notificaciones – Alertas de Reserva, Promociones y Seguridad

← Notificaciones



Hoy

Nuevo recibo disponible

Tu recibo por el
estacionamiento en Real Plaza •
Salaverry está disponible



Hace 2 horas

Estacionamiento disponible

Hay espacios disponibles en tu
estacionamiento favorito: •
Larcomar



Hace 5 horas

Anteriores

Oferta especial

50% de descuento en tu próxima
reserva en Centro Cívico



Inicio



Notificaciones



Ajustes

 Segmento Objetivo: Dueño de Estacionamiento – Flutter, Dart y Android Studio

 Pantalla de Inicio de Sesión – Login del Dueño de Estacionamiento

1:01

54%



ParkingNow

Panel de Propietarios

Administra tu estacionamiento de manera
inteligente



¡Bienvenido de
vuelta!

Accede a tu panel de control

Correo electrónico



ejemplo@correo.com

Contraseña



Tu contraseña es:



Tu contraseña seg...

?

Pantalla de Recuperación – ¿Olvidaste tu Contraseña?

1:01

54%



¿Olvidaste tu contraseña?

Ingresá tu correo y te enviaremos un enlace para restablecerla



Correo electrónico

Enviar enlace de recuperación

[Volver al inicio de sesión](#)

¿Necesitas ayuda? Contacta a nuestro soporte



 Pantalla de Cambio de Contraseña – Validación Segura para el Dueño



Cambiar Contraseña

Crea una nueva contraseña segura para proteger tu cuenta



Nueva contraseña



Confirmar contraseña



Requisitos de contraseña:

- ✗ Al menos 8 caracteres
- ✗ Al menos una letra mayúscula (A-Z)
- ✗ Al menos un número (0-9)
- ✗ Al menos un símbolo (!@#\$%^&*)
- ✗ Las contraseñas coinciden

¿Necesitas ayuda? Contacta a nuestro soporte



Registro de Nuevo Usuario – Crear Cuenta como Dueño de Estacionamiento

1:02

54%



Paso 1 de 3



ParkingNow Owner

Únete a nuestra red de
propietarios



Información Personal

Cuéntanos sobre ti

Nombre completo



Ingresá tu nombre y apellidos

Correo electrónico



ejemplo@correo.com

Número de teléfono



+51 999 999 999

Siguiente



1:02

54%



Paso 2 de 3



ParkingNow Owner

Únete a nuestra red de
propietarios



Seguridad

Protege tu cuenta

Contraseña



.....



Confirmar contraseña



.....



Fortaleza de la contraseña: Débil

Anterior

Siguiente



1:02

54%



Paso 3 de 3



ParkingNow Owner

Únete a nuestra red de
propietarios



Información del Negocio

Detalles de tu estacionamiento

RUC



20123456789

Nombre del negocio



Estacionamiento Central

Dirección del estacionamiento



Av. Principal 123, Lima

Capacidad total



Dashboard Principal – Vista General de la Actividad del Estacionamiento

1:03

54%

Dashboard



¡Bienvenido de vuelta!

John Smith

Lima, Perú

P

12

Espacios



8

Reservas



75%

Ocupación

Resumen de Ingresos

Junio 2025

Hoy

S/ 150.00

+12% vs semana pasada

Esta semana

S/ 1,200.00

+8% vs semana pasada

[Ver reporte completo](#)



Dashboard



Estaciona...



Reservas



Notificacio...



 **Términos y Condiciones – Revisión Legal por Parte del Dueño**

1:06 Ct

55%

Dashboard



¡Bienvenido de vuelta!

John Smith

Lima, Perú

P

12

Q

8

C

75%

Términos y Condiciones

Para continuar usando la app,
debes aceptar los términos y
condiciones.

Aceptar

Rechazar

Esta semana

S/ 1,200.00

+8% vs semana pasada

Ver reporte completo



Dashboard



Estaciona...



Reservas



Notificacio...



 **Barra Lateral – Navegación entre Funciones del Dueño de Estacionamiento**

1:06 Ct

55%



Juan Pérez

Propietario



Inicio



Registros de Locales

3



Reservas

5

2025



Seguridad



Notificaciones

3

[→] Cerrar Sesión

P ParkingNow v1.0.0

© 2025 Todos los derechos reservados

Notificacio..

 Registro Avanzado – Añadir Múltiples Estacionamientos y Configurar Extras

1:03

54%



Registro de Estacionamiento

Paso 1 de 4

25%

Información Básica



Ingresá los datos principales de tu estacionamiento

Nombre del estacionamiento



Ej: Parking San Isidro

Descripción



Describe brevemente tu estacionamiento

Tipo de estacionamiento



Público



Número de espacios

...

Ej: 20

Siguiente

Ubicación del Estacionamiento – Marcado y Visualización Geográfica



Registro de Estacionamiento

Paso 2 de 4

50%

Ubicación



Indica dónde se encuentra tu
estacionamiento

Dirección o lugar



Ej: Av. Javier Prado 1234, ...



Información adicional

Referencias



Ej: Frente al parque, esquina con Av....

Anterior

Siguiente

 **Servicios del Estacionamiento – Horarios de Atención y Beneficios para Conductores**

1:04 Ct ☰ •

🔊 🔇 VoLTE 54%



Registro de Estacionamiento

Paso 3 de 4

75%

Carga eléctrica

Acceso para discapacitados

Horario de atención



Horario de apertura



7:00 AM



Horario de cierre



10:00 PM

Días de atención

L

M

X

J

V

S

D

Anterior

Siguiente



 Configuración General – Tarifas, Métodos de Pago y Preferencias



Registro de Estacionamiento

Paso 4 de 4

100%



Precios y Términos

Establece las tarifas y condiciones

Tarifas

Precio por hora (S/)



Ej: 5.00



Tarifa por día

Estacionamiento
por día completo

S/ 25.00



Tarifa mensual

Abono mensual

S/ 300.00



Métodos de pago aceptados

Efectivo Tarjeta [Anterior](#)[Registrar](#)



Gestión de Reservas – Visualización y Administración de Espacios Reservados

1:05 Ct

LTE 55%



Reservas



4

Total



1

Confirmadas



1

Pendientes



Carlos Rodríg...

NUEVO

Confirmada

ID: RES001

Toyota Corolla

ABC-123

P A-15

09:00 - 11:00

S/ 10.00

Contactar

Gestionar



María González

Pendiente

ID: RES002

Honda Civic

XYZ-789

P B-08

10:30 - 12:30

S/ 10.00

Contactar

Gestionar



Luis Pérez

Completada

ID: RES003

Nissan Sentra

DEF-456

P C-22

13:00 - 15:00

S/ 10.00



Contactar

Gestionar

 Módulo de Seguridad – Protección del Espacio y Alertas del Estacionamiento

1:05 Ct

55%



Centro de Seguridad



3

Vehículos
Registrados



1

Alertas
Importantes



24/7

Monitoreo
Activo



99.9%

Tiempo
Operativo

Acciones Rápidas



Registrar

Nuevo vehículo



Historial

Ver actividad

Actividad Reciente

 Centro de Notificaciones y Configuración – Alertas y Preferencias del Dueño

1:05 Ct

55%



Notificac...



¡Hola, John!

Tienes 3 notificaciones sin leer



3

Nuevas



2

Urgentes

✓ Todas

Reservas

Pagos

Recientes

Historial



Nueva reserva confirmada



Carlos Rodríguez ha confirmado su reserva para mañana a las 09:...

5m



Pago recibido



Has recibido un pago de S/ 25.00 por la reserva #12345. El pago ha...

15m



URGENTE

✓ Marcar como leídas



Configuración del Sistema – Ajustes Generales del Dueño y Preferencias de la Plataforma

1:05 Ct

55%



Configuraciones

Personaliza tu experiencia



Juan Carlos Pérez

Propietario Premium



Cuenta Verificada

General



Modo Oscuro

Cambia la apariencia de la aplicación



Idioma

Español



Moneda

PEN (S/)



Zona Horaria

GMT-5 (Lima)



Notificaciones

6.2.3.6 Services Documentation Evidence for Sprint Review

Se documentaron todos los endpoints desarrollados en este sprint en el archivo `API_DOC.md`, incluyendo:

- Método (GET, POST, PUT, DELETE).
- Ruta base.
- Parámetros de entrada.
- Respuesta esperada.
- Ejemplo de uso.
- Código de respuesta (200, 400, 401, 404, 500).

Esta documentación es parte del repositorio y está sujeta a actualizaciones en tiempo real conforme se despliegue en producción.

Usuario

<http://localhost:3000/usuario>

POST Crear un nuevo usuario

The screenshot shows the Postman application interface. On the left, there's a sidebar with collections: 'contigoNovoy', 'assistent', and 'parkingnow'. Under 'parkingnow', there are three requests: 'GET get', 'POST nuevo local', and 'POST Crear un nuevo usuario', with the last one selected. The main area shows a POST request to 'http://localhost:3000/usuario'. The 'Body' tab is selected, showing a JSON payload:

```
1 {  
2   "name": "Jesús Lopez",  
3   "email": "jesuslz@example.com",  
4   "password": "pass123",  
5   "placa": "ABC-321",  
6   "dni": "72787997"  
7 }
```

The response pane shows the JSON response received from the server:

```
1 {  
2   "id": 2,  
3   "name": "Jesús Lopez",  
4   "email": "jesuslz@example.com",  
5   "password": "pass123",  
6   "placa": "ABC-321",  
7   "dni": "72787997",  
8   "ruc": null,  
9   "tipoUsuario": "conductor"  
10 }
```

The status bar at the bottom indicates a 201 Created response with a 57ms duration and 153B size.

<http://localhost:3000/usuario>

GET Obtener todos los usuarios

Collection Edit View Window Help

bruno ... parkingnow Safe Mode ⚡ ⏱ ⚡ ⏱ ⏵ ⏵ ⏵ ⏵ ⏵ ⏵ No Environment

Collections Search requests ...

> contigoNovoy
> assistent
v parkingnow
 GET get
 POST nuevo local
 POST Crear un nuevo usuario

... GET http://localhost:3000/usuario

Params Body Headers Auth * Vars Script
Assert Tests Docs

Query
Name Path
+ Add Param

Path ?
Name Value

Response Headers 7 Timeline Tests
200 OK 29ms 304B
[
1 {
2 "id": 1,
3 "name": "Juan Pérez",
4 "email": "juan@example.com",
5 "password": "123456",
6 "placa": "ABC-123",
7 "dni": "12345678",
8 "ruc": null,
9 "tipoUsuario": "conductor"
10 },
11 {
12 "id": 2,
13 "name": "Jesús Lopez",
14 "email": "jesuslz@example.com",
15 "password": "pass123",
16 "placa": "ABC-321",
17 "dni": "72787997",
18 "ruc": null,
19 "tipoUsuario": "conductor"
20 }
21]

Activar Wdio
Ve a Configuración

http://localhost:3000/usuario/1

get Obtener un usuario por ID

Collection Edit View Window Help

bruno ... parkingnow Safe Mode ⚡ ⏱ ⚡ ⏱ ⏵ ⏵ ⏵ ⏵ ⏵ ⏵ No Environment

Collections Search requests ...

> contigoNovoy
> assistent
v parkingnow
 GET get
 POST nuevo local
 POST Crear un nuevo usuario

... GET http://localhost:3000/usuario/1

Params Body Headers Auth * Vars Script
Assert Tests Docs

Query
Name Path
+ Add Param

Path ?
Name Value

Response Headers 7 Timeline Tests
200 OK 124ms 148B
{
1 "id": 1,
2 "name": "Juan Pérez",
3 "email": "juan@example.com",
4 "password": "123456",
5 "placa": "ABC-123",
6 "dni": "12345678",
7 "ruc": null,
8 "tipoUsuario": "conductor"
9 }
10 }

http://localhost:3000/usuario/1/locales

GET Obtiene todos los locales asociados a un usuario específico.

The screenshot shows the Postman application interface. On the left sidebar, there's a user icon for 'bruno' and a 'Collections' section containing 'contigoNovoy', 'assistent', and 'parkingnow'. The 'parkingnow' collection is expanded, showing three requests: 'GET get' (selected), 'POST nuevo local', and 'POST Crear un nuevo usuario'. The main workspace shows a 'parkingnow' collection with a 'GET get' request. The request URL is 'http://localhost:3000/usuario/1/locales'. The 'Params' tab is selected, showing an 'Assert' section. The 'Response' tab shows a JSON response with one item:

```
[  
  {  
    "id": 4,  
    "nombre": "Playa Central",  
    "direccion": "Av. Principal 123",  
    "telefono": "987654321",  
    "plazas": 1  
  }  
]
```

<http://localhost:3000/usuario/1>

PUT Actualizar un usuario por ID

The screenshot shows the Postman application interface. In the top left, there's a user icon for 'bruno'. The top right has a 'Safe Mode' button and a 'No Environment' dropdown. On the far right, there are several global settings icons.

The main workspace is titled 'parkingnow'. It features a navigation bar with tabs: 'Collection' (highlighted), 'GET get', 'POST nuevo lo...', 'POST Crear un...', 'PUT Actualizar...', and a '+' button. Below this, a search bar says 'Search requests ...'.

On the left, a sidebar lists collections: 'contigoNovoy', 'assistent', and 'parkingnow' (which is expanded). Under 'parkingnow', there are four methods: 'GET get', 'POST nuevo local', 'POST Crear un nuevo usuario', and 'PUT Actualizar un usuario po...'. The 'PUT' method is selected.

The central area shows the request details for the selected PUT method:

- Method: PUT
- URL: http://localhost:3000/usuario/1
- Params tab (selected)
- Body tab (selected): Contains a JSON payload:

```
1 {  
2   "name": "Juan Actualizado",  
3   "email": "juan_updated@example.com",  
4   "placa": "XYZ-789"  
5 }
```
- Headers tab
- Auth tab
- Vars tab
- Script tab
- Assert tab
- Tests tab
- Docs tab
- JSON dropdown
- Prettify button

The right side displays the response:

- Response tab (selected)
- Headers tab (with a count of 7)
- Timeline tab
- Tests tab
- Response body:

```
1 {  
2   "id": 1,  
3   "name": "Juan Actualizado",  
4   "email": "juan_updated@example.com",  
5   "password": "123456",  
6   "placa": "XYZ-789",  
7   "dni": "12345678",  
8   "ruc": null,  
9   "tipoUsuario": "conductor"  
10 }
```
- Metrics at the bottom: 200 OK, 148ms, 161B

<http://localhost:3000/usuario/1>

DELETE Eliminar un usuario por ID

The screenshot shows the Postman application interface. On the left, there's a sidebar with a profile icon for 'bruno' and a 'Collections' section containing several items: 'contigoNovoy', 'assistant', 'parkingnow' (which is expanded), and three items under it: 'GET get', 'POST nuevo local', 'POST Crear un nuevo usuario', 'PUT Actualizar un usuario po...', and 'DEL Eliminar un usuario por ID'. The main area has a header 'parkingnow' with tabs for 'Collection', 'GET get', 'POST nuevo lo...', 'POST Crear un...', 'PUT Actualizar...', and 'DELETE Elimin...'. Below this, a search bar says 'Search requests ...'. The main content area shows a 'DELETE' request to 'http://localhost:3000/usuario/1'. The 'Params' tab is selected, showing fields for 'Name' and 'Path'. Other tabs include 'Body', 'Headers', 'Auth *', 'Vars', 'Script', 'Assert', 'Tests', 'Docs', 'Query', '+ Add Param', 'Path', and 'Name' and 'Value' input fields. To the right, there's a 'Response' tab showing a status of '200 OK' with a response body containing the number '1'. Other tabs in this section include 'Headers 5', 'Timeline', and 'Tests'.

Local

<http://localhost:3000/local>

POST Crear un nuevo local

POST http://localhost:3000/local

Body (JSON)

```
{
  "nombre": "Playa Central",
  "direccion": "Av. Principal 123",
  "telefono": "987654321",
  "usuario": {
    "id": 2
  }
}
```

Response (201 Created)

```
{
  "id": 5,
  "nombre": "Playa Central",
  "direccion": "Av. Principal 123",
  "telefono": "987654321",
  "plazas": 1,
  "usuario": {
    "id": 2
  }
}
```

<http://localhost:3000/local/5>

GET Obtener un local por ID

GET http://localhost:3000/local/5

Query

Name	Path

+ Add Param

Path

Name	Value

Response (200 OK)

```
{
  "id": 5,
  "nombre": "Playa Central",
  "direccion": "Av. Principal 123",
  "telefono": "987654321",
  "plazas": 1,
  "usuario": {
    "id": 2,
    "name": "Jesus Lopez",
    "email": "jesuslz@example.com",
    "password": "pass123",
    "placa": "ABC-321",
    "dni": "72787997",
    "ruc": null,
    "tipoUsuario": "conductor"
  }
}
```

<http://localhost:3000/local>

GET Obtener todos los locales

Collections

Collection

GET get • POST Crear nu... • POST Crear un... • > +

Search requests ...

> contigoNovoy
> assistent
parkingnow

GET get
POST Crear nuevo local
POST Crear un nuevo usuario
PUT Actualizar un usuario po...
DEL Eliminar un usuario por ID

Params Body Headers Auth *

Vars Script Assert Tests Docs

Query

Name	Path
+ Add Param	

Path ?

Name	Value

Response Headers 7 Timeline Tests

200 OK 36ms 265B

```

1 [
2 {
3   "id": 5,
4   "nombre": "Playa Central",
5   "direccion": "Av. Principal 1
23",
6   "telefono": "987654321",
7   "plazas": 1,
8   "usuario": {
9     "id": 2,
10    "name": "Jesús Lopez",
11    "email": "jesuslz@example.c
om",
12    "password": "pass123",
13    "placa": "ABC-321",
14    "dni": "72787997",
15    "ruc": null,
16    "tipoUsuario": "conductor"
17  }
18 }

```

<http://localhost:3000/local/5>

PUT Actualizar un local por ID

bruno ***

Collections

parkingnow

Save Mode No Environment

GET get • POST Crear nu... • POST Crear un... • PUT Actualizar... • > +

Search requests ...

> contigoNovoy
> assistent
parkingnow

GET get
POST Crear nuevo local
POST Crear un nuevo usuario
PUT Actualizar por ID
DEL Eliminar un usuario por ID

Params Body * Headers Auth *

Vars Script Assert Tests Docs

PUT http://localhost:3000/local/5

JSON Prettify

```

1 {
2   "nombre": "Playa Actualiz
ada",
3   "direccion": "Av. Nueva 7
89",
4   "telefono": "987654323"
5 }

```

Response Headers 7 Timeline Tests

200 OK 36ms 263B

```

1 {
2   "id": 5,
3   "nombre": "Playa Actualizada",
4   "direccion": "Av. Nueva 789",
5   "telefono": "987654323",
6   "plazas": 1,
7   "usuario": {
8     "id": 2,
9     "name": "Jesús Lopez",
10    "email": "jesuslz@example.com"
11   },
12   "password": "pass123",
13   "placa": "ABC-321",
14   "dni": "72787997",
15   "ruc": null,
16   "tipoUsuario": "conductor"
17 }

```

<http://localhost:3000/local/5>

DELETE Eliminar un local por ID

Collections

Search requests ...

- > contigoNovoy
- > assistent
- < parkingnow
 - GET get**
 - POST Crear nuevo local**
 - POST Crear un nuevo usuario**
 - PUT Actualizar por ID**
 - DEL Eliminar por ID**

DELETE http://localhost:3000/local/5

Params Body Headers Auth *
Vars Script Assert Tests Docs

Query

Name	Path
+ Add Param	

Path ?
Name Value

Response Headers 5 Timeline Tests
200 OK 30ms 0B

```
1
```

<http://localhost:3000/local/6/usuario>

GET Obtener el usuario dueño de un local

bruno ***

Collections

Search requests ...

- > contigoNovoy
- > assistent
- < parkingnow
 - GET get**
 - POST Crear nuevo local**
 - POST Crear un nuevo usuario**
 - PUT Actualizar por ID**
 - DEL Eliminar por ID**

parkingnow Safe Mode No Environment

GET http://localhost:3000/local/6/usuario

Params Body Headers Auth *
Vars Script Assert Tests Docs

Query

Name	Path
+ Add Param	

Path ?
Name Value

Response Headers 7 Timeline Tests
200 OK 4ms 153B

```

1 {
2   "id": 2,
3   "name": "Jesús Lopez",
4   "email": "jesuslz@example.com",
5   "password": "pass123",
6   "placa": "ABC-321",
7   "dni": "72787997",
8   "ruc": null,
9   "tipoUsuario": "conductor"
10 }
```

Reserva

<http://localhost:3000/reserva>

POST Crear una nueva reserva

bruno ... parkingnow Safe Mode No Environment

Collections ↕ Search requests ...

- > contigoNovoy
- > assistent
- ✓ parkingnow
 - GET get
 - POST Crear nueva reserva**
 - POST Crear un nuevo usuario
 - PUT Actualizar por ID
 - DEL Eliminar por ID

POST http://localhost:3000/reserva

Params Body * Headers Auth * Vars
Script Assert Tests Docs JSON Prettify

```

1 {
2   "usuario": { "id": 2 },
3   "local": { "id": 6 },
4   "nro_plaza": 5,
5   "fh_inicio": "2025-05-16T08:00:00Z",
6   "fh_fin": "2025-05-16T10:00:00Z"
7 }
```

Response Headers 7 Timeline Tests
🔗 ⏲ 201 Created 42ms 125B

```

1 {
2   "id": 1,
3   "usuario": {
4     "id": 2
5   },
6   "local": {
7     "id": 6
8   },
9   "nro_plaza": 5,
10  "fh_inicio": "2025-05-16T08:00:00Z",
11  "fh_fin": "2025-05-16T10:00:00Z"
12 }
```

http://localhost:3000/reserva
GET Obtener todas las reservas

Q Search requests ...

GET http://localhost:3000/reserva

Params Body Headers Auth * Vars Script
Assert Tests Docs

Query

Name	Path
+ Add Param	

Path ⓘ

Name	Value

Response Headers 7 Timeline Tests
🔗 ⏲ 200 OK 29ms 371B

```

1 [
2   {
3     "id": 1,
4     "usuario": {
5       "id": 2,
6       "name": "Jesús Lopez",
7       "email": "jesuslz@example.com",
8       "password": "pass123",
9       "placa": "ABC-321",
10      "dni": "72787997",
11      "ruc": null,
12      "tipoUsuario": "conductor"
13    },
14    "local": {
15      "id": 6,
16      "nombre": "Playa Central",
17      "direccion": "Av. Principal 123",
18      "telefono": "987654321",
19      "plazas": 1
20    },
21    "nro_plaza": 5,
22    "fh_inicio": "2025-05-16T08:00:00.000Z",
23    "fh_fin": "2025-05-16T10:00:00.000Z"
24  }
25 ]
```

http://localhost:3000/reserva/1
GET Obtener una reserva por ID

The screenshot shows the Postman application interface. On the left, there's a sidebar with user information (bruno) and a list of collections: 'contigoNovoy', 'assistent', and 'parkingnow'. Under 'parkingnow', several requests are listed: 'GET get', 'POST Crear nueva reserva', 'POST Crear un nuevo usuario', 'PUT Actualizar por ID', and 'DEL Eliminar por ID'. The main area shows a 'parkingnow' collection with a single GET request to 'http://localhost:3000/reserva/1'. The 'Params' tab is selected, showing a table with columns 'Name' and 'Path'. Below it is a '+ Add Param' button and a 'Path' section with a table for 'Name' and 'Value'. The 'Response' tab is selected, displaying a JSON response with line numbers from 1 to 23. The response object contains details about a parking reservation, including user information, local details, and timing.

```
{
  "id": 1,
  "usuario": {
    "id": 2,
    "name": "Jesús Lopez",
    "email": "jesuslz@example.com",
    "password": "pass123",
    "placa": "ABC-321",
    "dni": "72787997",
    "ruc": null,
    "tipoUsuario": "conductor"
  },
  "local": {
    "id": 6,
    "nombre": "Playa Central",
    "direccion": "Av. Principal 123",
    "telefono": "987654321",
    "plazas": 1
  },
  "nro_plaza": 5,
  "fh_inicio": "2025-05-16T08:00:00.000Z",
  "fh_fin": "2025-05-16T10:00:00.000Z"
}
```

<http://localhost:3000/reserva/1>

PUT Actualizar una reserva por ID

The screenshot shows the Postman application interface with the following details:

- Method:** PUT
- URL:** http://localhost:3000/reserva/1
- Body (JSON):**

```
1 {  
2   "nro_plaza": 10,  
3   "fh_fin": "2025-05-16T12:00:00Z"  
4 }
```
- Response Headers:** 200 OK, 31ms, 366B
- Response Body (Pretty):**

```
1 {  
2   "id": 1,  
3   "usuario": {  
4     "id": 2,  
5     "name": "Jesús Lopez",  
6     "email": "jesuslz@example.com",  
7     "password": "pass123",  
8     "placa": "ABC-321",  
9     "dni": "72787997",  
10    "ruc": null,  
11    "tipoUsuario": "conductor"  
12  },  
13  "local": {  
14    "id": 6,  
15    "nombre": "Playa Central",  
16    "direccion": "Av. Principal 123",  
17    "telefono": "987654321",  
18    "plazas": 1  
19  },  
20  "nro_plaza": 10,  
21  "fh_inicio": "2025-05-16T08:00:00.000Z",  
22  "fh_fin": "2025-05-16T12:00:00Z"  
23}
```

<http://localhost:3000/reserva/1>

DELETE Eliminar una reserva por ID

The screenshot shows the Postman interface. On the left, there's a sidebar with collections like 'contigoNovoy', 'assistant', and 'parkingnow'. Under 'parkingnow', there are several methods: 'GET get', 'POST Crear nueva reserva', 'POST Crear un nuevo usuario', 'PUT Actualizar por ID', and 'DEL Eliminar por ID'. The 'DEL Eliminar por ID' method is selected. In the main area, a 'DELETE' request is made to 'http://localhost:3000/reserva/1'. The 'Params' tab is active, showing a single parameter 'Name' with value 'Path'. The 'Response' tab shows a status of '200 OK' with a response body containing the number '1'.

Durante este Sprint desplegamos el backend e hicimos las pruebas requeridas

Inicio del despliegue

Empezamos logueandonos en la plataforma

```
samuel@DESKTOP-JPEIUCT MINGW64 /d/upc/Aplicaciones Moviles/Backend (main)
$ npm install --save @nestjs/cli
up to date, audited 895 packages in 17s

163 packages are looking for funding
  run `npm fund` for details

1 low severity vulnerability

To address all issues, run:
  npm audit fix

Run `npm audit` for details.

samuel@DESKTOP-JPEIUCT MINGW64 /d/upc/Aplicaciones Moviles/Backend (main)
$ heroku login
heroku: Press any key to open up the browser to login or q to exit:
Opening browser to https://cli-auth.herokuapp.com/auth/cli/browser/c2ebda91-4842-47c8-b8ca-4e1b729c9e4?requestor=SFMyNTY.g2g0bQAAAAsxNzkuNi4zLjIyNg4GACjK6Y-XAIIAAVGA.wQuA0jvUEmrhi3t3vGvn7GY0xPq26Z90YRykHNqp8
Logging in... done
Logged in as samuelmolina281816@gmail.com
```

Configuracion de la base de datos

Configuramos una base de datos con NEON

```
samuel@DESKTOP-JPEIUCT MINGW64 /d/upc/Aplicaciones Moviles/Backend (main)
$ heroku config:set DATABASE_URL="postgresql://neondb_owner:npq_Ytv3go6FPQM@ep-floral-shadow-a5w34cdf-pooler.us-east-2.aws.neon.tech/neondb?sslmode=require"
Setting DATABASE_URL and restarting ⚡ parkingnow-app... done, v3
DATABASE_URL: postgresql://neondb_owner:npq_Ytv3go6FPQM@ep-floral-shadow-a5w34cdf-pooler.us-east-2.aws.neon.tech/neondb?sslmode=require

samuel@DESKTOP-JPEIUCT MINGW64 /d/upc/Aplicaciones Moviles/Backend (main)
$

samuel@DESKTOP-JPEIUCT MINGW64 /d/upc/Aplicaciones Moviles/Backend (main)
$ git status
On branch main
Your branch is up to date with 'origin/main'.

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
    (use "git restore <file>..." to discard changes in working directory)
      modified: package-lock.json
      modified: package.json
      modified: src/app.module.ts

no changes added to commit (use "git add" and/or "git commit -a")
```

Cambios finales para producción

Se agregan cambios a producción para la ejecución correcta de Nest en la plataforma

The screenshot shows a terminal window with the following content:

```
1 package.json > {} scripts
8 "scripts": {
9   "build": "nest build",
10  "format": "prettier --write \"src/**/*.ts\" \"test/**/*.ts\"",
11  "start": "node dist/main.js",
12  "start:dev": "nest start --watch",
13  "start:debug": "nest start --debug --watch",
14  "start:prod": "node dist/main",
15  "lint": "eslint \"{src,apps,libs,test}/**/*.{ts,js}\" --fix",
16  "test": "jest",
17  "test:watch": "jest --watch",
18  "test:cov": "jest --coverage",
19  "test:debug": "node --inspect-brk -r tsconfig-paths/register -r ts-node/register node_modules/.bin/jest --runInBand",
20  "test:e2e": "jest --config ./test/jest-e2e.json"
21 },
22 "dependencies": {
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
samue@DESKTOP-JPEIUCT MINGW64 /d/upc/Aplicaciones Moviles/Backend (main)
● $ git status
On branch main
Your branch is up to date with 'origin/main'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    modified:  package-lock.json
    modified:  package.json
    modified:  src/app.module.ts

samue@DESKTOP-JPEIUCT MINGW64 /d/upc/Aplicaciones Moviles/Backend (main)
● $ git commit -m "Configuraciones para la conexión a producción a Neon PostgreSQL"
[main 99663ca] Configuraciones para la conexión a producción a Neon PostgreSQL
 3 files changed, 53 insertions(+), 10 deletions(-)

samue@DESKTOP-JPEIUCT MINGW64 /d/upc/Aplicaciones Moviles/Backend (main)
● $ git push heroku main
Enumerating objects: 84, done.
Counting objects: 100% (84/84), done.
Delta compression using up to 8 threads
Compressing objects: 100% (82/82), done.
Writing objects: 100% (84/84), 117.11 KiB | 3.90 MiB/s, done.
Total 84 (delta 37), reused 18 (delta 0), pack-reused 0 (from 0)
remote: Updated 29 paths from eb4b912
remote: Compressing source files... done.
remote: Building source:
remote:
remote: -----> Building on the Heroku-24 stack
remote: -----> Determining which buildpack to use for this app
remote: -----> Node.js app detected
remote:
```

Obtenemos el log del despliegue y comprobamos la conexión a la base de datos

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
remote:     npm notice Changelog: https://github.com/npm/cli/releases/tag/v11.4.2
remote:     npm notice To update run: npm install -g npm@11.4.2
remote:     npm notice
remote:
remote: ----> Build
remote:     Running build
remote:
remote: > backend-parkingnow@0.0.1 build
remote: > nest build
remote:
remote:
remote: ----> Caching build
remote:     - npm cache
remote:
remote: ----> Pruning devDependencies
remote:
remote:     up to date, audited 215 packages in 1s
remote:
remote:     47 packages are looking for funding
remote:       run `npm fund` for details
remote:
remote:     1 low severity vulnerability
remote:
remote:     To address all issues, run:
remote:       npm audit fix
remote:
remote:     Run `npm audit` for details.
remote:     npm notice
remote:     npm notice New major version of npm available! 10.9.2 -> 11.4.2
remote:     npm notice Changelog: https://github.com/npm/cli/releases/tag/v11.4.2
remote:     npm notice To update run: npm install -g npm@11.4.2
remote:     npm notice
remote:
remote: ----> Build succeeded!
remote: ----> Discovering process types
remote:     Procfile declares types    -> (none)
remote:     Default types for buildpack -> web
remote:
remote: ----> Compressing...
remote:     Done: 65M
remote: ----> Launching...
remote:     Released v4
remote:     https://parkingnow-app-963963f523b4.herokuapp.com/ deployed to Heroku
remote:
remote: Verifying deploy... done.
To https://git.heroku.com/parkingnow-app.git
 * [new branch]      main -> main

samue@DESKTOP-JPEIUCT MINGW64 /d/upc/Aplicaciones Moviles/Backend (main)
$ heroku config:get DATABASE_URL
postgresql://neondb_owner:npg_Ytv3go6FPQOM@ep-floral-shadow-a5w34cdf-pooler.us-east-2.aws.neon.tech/neondb?sslmode=require

samue@DESKTOP-JPEIUCT MINGW64 /d/upc/Aplicaciones Moviles/Backend (main)
$ heroku open
```

Comprobacion del despliegue

Revision de los logs de despliegue

```
samue@DESKTOP-JPEIUCT MINGW64 /d/upc/Aplicaciones Moviles/Backend (main)
$ heroku logs --tail
2025-06-21T00:40:20.199533+00:00 app[api]: Release v1 created by user samuelmolina281816@gmail.com
2025-06-21T00:40:20.199533+00:00 app[api]: Initial release by user samuelmolina281816@gmail.com
2025-06-21T00:40:20.402899+00:00 app[api]: Release v2 created by user samuelmolina281816@gmail.com
2025-06-21T00:40:20.402899+00:00 app[api]: Enable Logplex by user samuelmolina281816@gmail.com
2025-06-21T01:16:08.600619+00:00 app[api]: Release v3 created by user samuelmolina281816@gmail.com
2025-06-21T01:16:08.600619+00:00 app[api]: Set DATABASE_URL config vars by user samuelmolina281816@gmail.com
2025-06-21T01:18:25.000000+00:00 app[api]: Build started by user samuelmolina281816@gmail.com
2025-06-21T01:18:52.897281+00:00 app[api]: Deploy 99663ca6 by user samuelmolina281816@gmail.com
2025-06-21T01:18:52.897281+00:00 app[api]: Release v4 created by user samuelmolina281816@gmail.com
2025-06-21T01:18:52.909856+00:00 app[api]: Scaled to web@1:Basic by user samuelmolina281816@gmail.com
2025-06-21T01:18:55.000000+00:00 app[api]: Build succeeded
```

Sin conflicto, despliegue exitoso

Detalles de la aplicación en la plataforma

dashboard.heroku.com/apps/parkingnow-app

HEROKU from Salesforce

Overview Resources Deploy Metrics Activity Access Settings

Get a complete visualization of your app in a team-based continuous delivery environment with [Heroku Pipelines](#). [Hide](#) [Create a Heroku Pipeline](#)

Metrics (last 24hrs) [All Metrics](#)

Response Time
191ms

Throughput
< 1 rps

Memory
12%

Latest Activity [All Activity](#)

samuelmolina281816@gmail.com: Deployed `dd1a4b0d` Today at 8:24 PM - v5

samuelmolina281816@gmail.com: Build succeeded Today at 8:24 PM - [View build log](#)

samuelmolina281816@gmail.com: Deployed `99663ca6` Today at 8:18 PM - v4

samuelmolina281816@gmail.com: Build succeeded Today at 8:18 PM - [View build log](#)

samuelmolina281816@gmail.com: Set `DATABASE_URL` config var Today at 8:16 PM - v3

samuelmolina281816@gmail.com: Enable Logplex Today at 7:40 PM - v2

samuelmolina281816@gmail.com: Initial release Today at 7:40 PM - v1

Installed Add-ons [Configure Add-ons](#)

There are no add-ons for this app
You can add add-ons to this app and they will show here. [Learn more](#)

Dyno Formations [Configure Dynos](#)

This app is using basic dynos

web npm start **ON**

Collaborator Activity [Manage Access](#)

samuelmolina281816@gmail.com [2 deploys](#)

Testeo de la aplicación

Testeo de la aplicacion en producción

HTTP <http://localhost:3000/usuario> [Save](#)

POST <https://parkingnow-app-963963f523b4.herokuapp.com/usuario> [Send](#)

Params Authorization Headers (8) **Body** Pre-request Script Tests Settings

none form-data x-www-form-urlencoded raw binary JSON

```

1
2   "name": "Samuel Molina",
3   "email": "samuel@example.com",
4   "password": "Pa$$word963",
5   "placa": "AKL-963",
6   "dni": "12345678",
7   "ruc": "20123456789"
8

```

Body Cookies Headers (11) Test Results

Pretty Raw Preview Visualize JSON

1
2 "id": 2,
3 "name": "Samuel Molina",
4 "email": "samuel@example.com",
5 "password": "Pa\$\$word963",
6 "placa": "AKL-963",
7 "dni": "12345678",
8 "ruc": "20123456789",
9 "tipoUsuario": "dueno_estacionamiento"
10

Status: 201 Created Time: 306 ms Size: 933 B Save Response

Verificamos la creacion de registros en la base de datos de producción

The screenshot shows the Neon SQL Editor interface. On the left, there's a sidebar with 'PROJECT' and 'BRANCH' sections. Under 'PROJECT', 'Dashboard', 'Branches', 'Integrations', 'Auth', and 'Settings' are listed. Under 'BRANCH', 'production' is selected. In the main area, the 'SQL Editor' tab is active. A query is being typed into the editor:

```

1 /*
2 In Neon, databases are stored on branches. By default, a project has one branch and one database.
3 You can select the branch and database to use from the drop-down menus above.
4
5 Try generating sample data and querying it by running the example statements below, or click
6 New Query to clear the editor.
7 */
8 select * from usuario;

```

Below the editor, a results table is displayed:

#	id	name	email	password	placa	dni	ruc	tipo_usuario
1	1	Diego Ulises	diego@example.com	123456	ABC-123	12345678	20123456789	dueno_estacionamiento
2	2	Samuel Molina	samuel@example.com	Pa\$\$word963	AKL-963	12345678	20123456789	dueno_estacionamiento

At the bottom right of the results table, it says '152ms 2 rows'.

Comprobacion de las apis abiertas.

6.2.3.7 Software Deployment Evidence for Sprint Review

Durante este sprint, ambas aplicaciones móviles fueron desarrolladas utilizando **Android Studio** y desplegadas correctamente para pruebas funcionales.

Cada segmento fue ejecutado mostrando correctamente su pantalla de inicio, navegación y funcionalidades clave, validando la estabilidad visual y lógica en entorno Android.

Además, se ha generado el archivo **APK instalable**, disponible públicamente para su descarga desde cualquier dispositivo Android.

El despliegue se logró con éxito en los entornos de prueba y está listo para producción tras la revisión final.

APK disponible aquí: <https://drive.google.com/drive/folders/1z7wYWMZDh1zbb6wGD3-tZEP9n5YQBU84?usp=sharing>

Capturas de Pantalla Oficiales - ParkingNow

Estas imágenes reflejan el funcionamiento en tiempo real de la app ParkingNow, para conductores y dueños de estacionamiento. Están listas para ser utilizadas en la Play Store o App Store.

Pantalla de Bienvenida e Inicio de Sesión



¡Bienvenido a ParkingNow!

Encuentra tu espacio en segundos



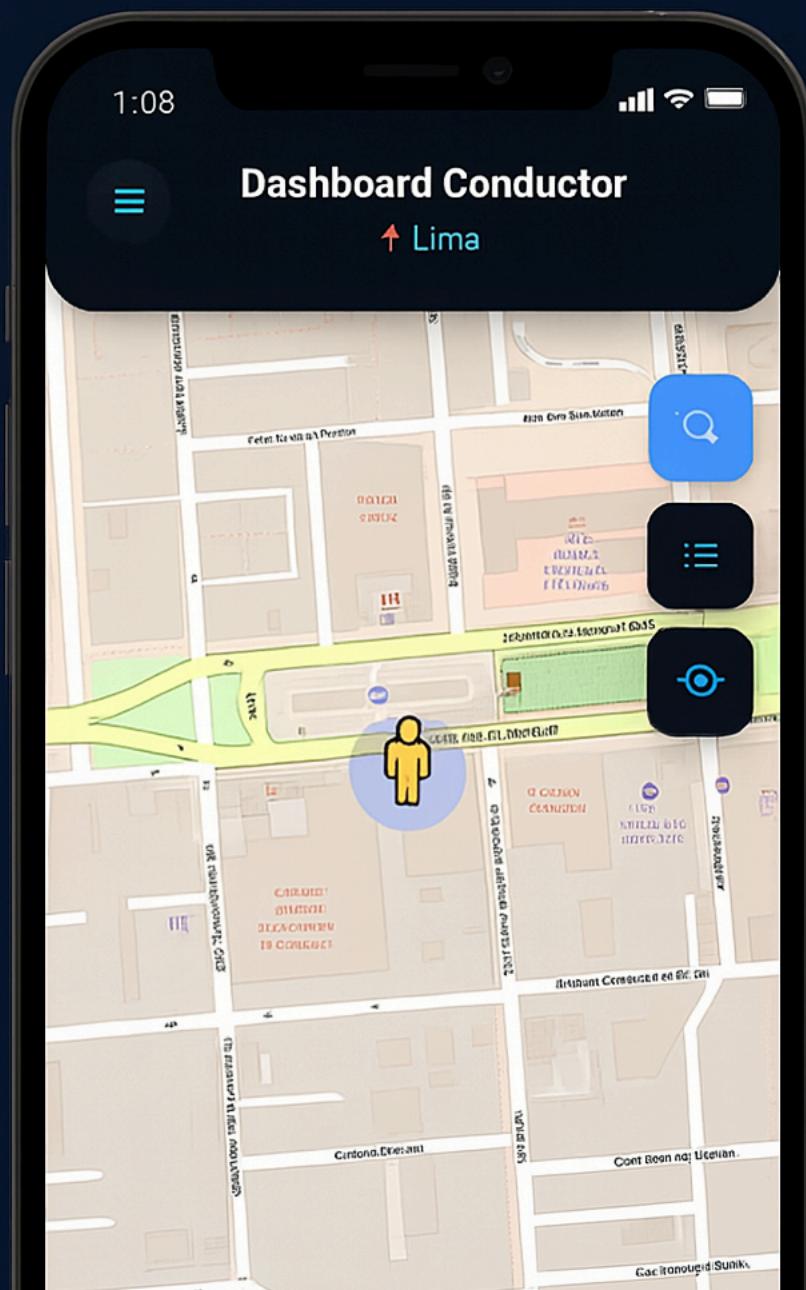


Conecta con tu cuenta en segundos. Interfaz limpia, moderna y lista para ayudarte a encontrar estacionamiento fácilmente desde el primer toque.

Reserva desde el Mapa en Tiempo Real

Reserva fácil y rápida

Olvídate de dar vueltas



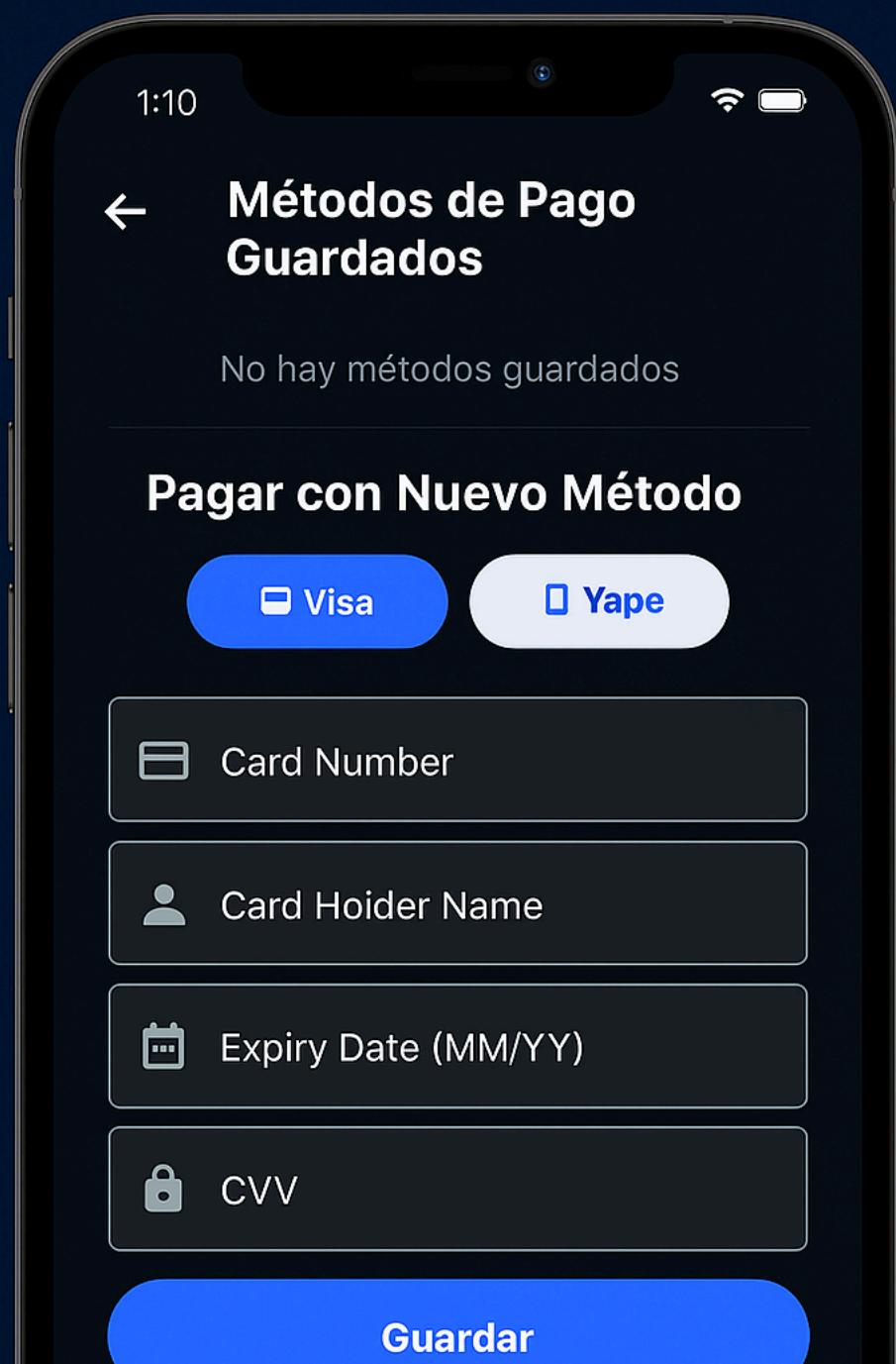


Olvídate de dar vueltas. Encuentra un estacionamiento cercano con geolocalización precisa y resérvalo desde el mapa en segundos.

Gestión de Métodos de Pago

Gestiona tus métodos de pago

Añade nuevas tarjetas o billeteras





Agrega tarjetas o billeteras digitales como Yape o Visa. Guarda métodos de pago de forma segura y paga con un solo toque.

Dashboard para Dueños de Estacionamientos

Para dueños de estacionamientos

Controla
tus ingresos
y reservas



Controla tus ingresos y el estado de tus espacios en tiempo real. Visualiza ocupación, reservas, ganancias y más desde un panel intuitivo.

Centro de Seguridad 24/7

Monitoreo las 24 horas

Mayor tranquilidad

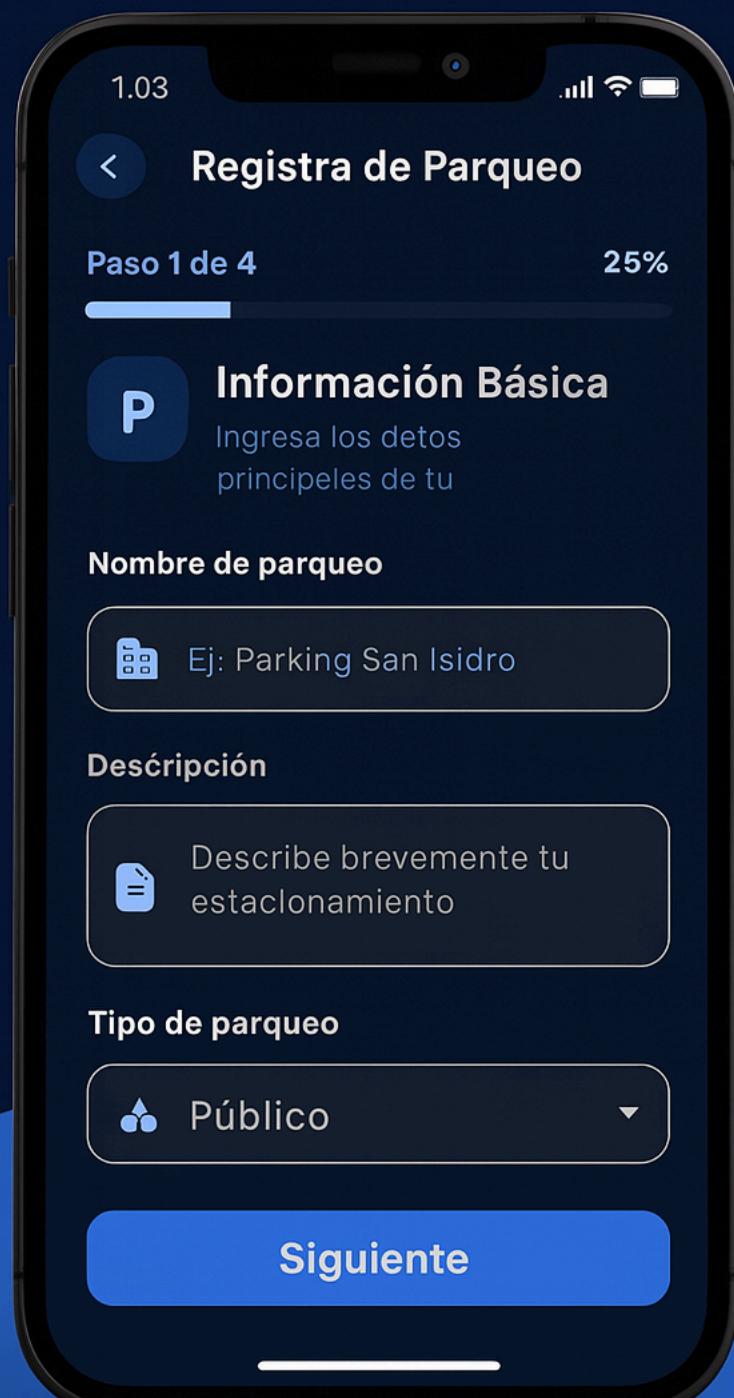


Supervisa alertas, vehículos registrados y tiempo operativo de tu parqueo. Brinda mayor tranquilidad a los usuarios con monitoreo activo.

Registro de Parqueo Paso a Paso

Registra tu parqueo

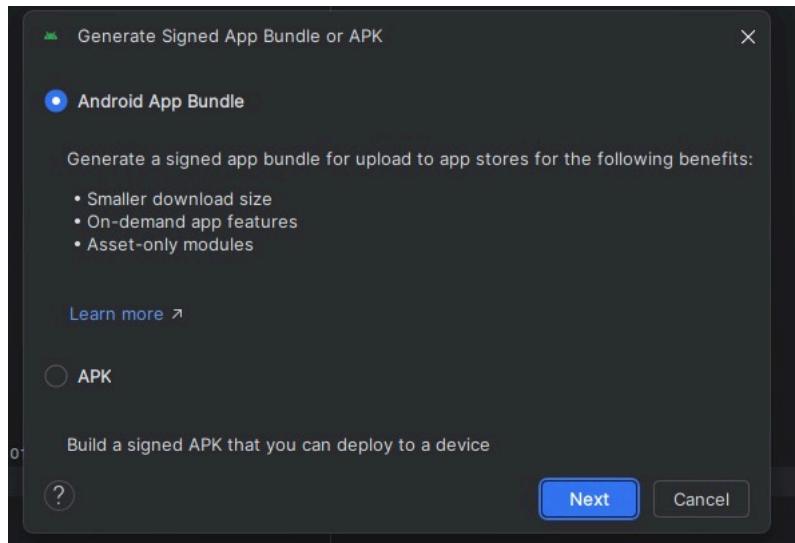
En unos minutos



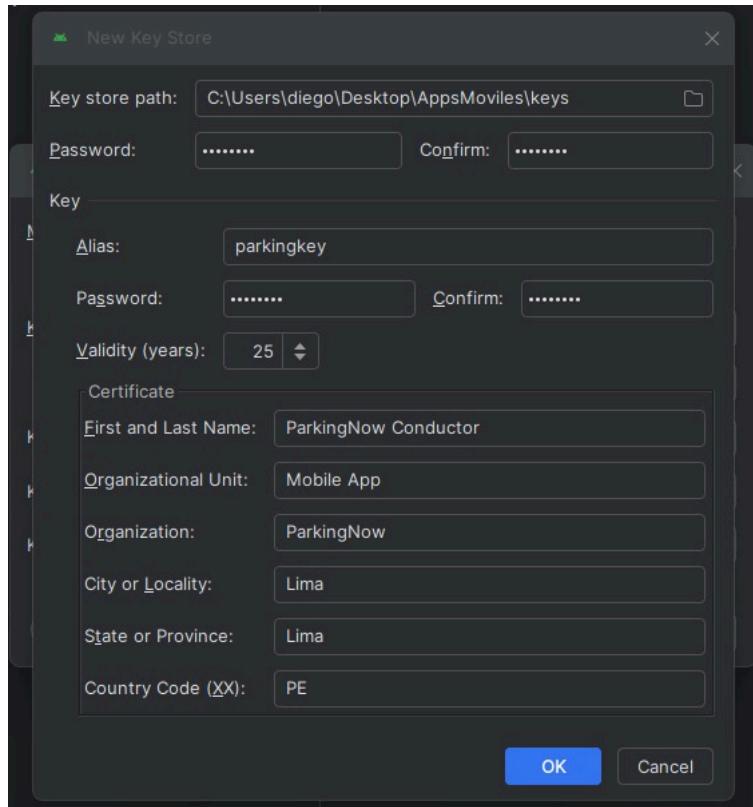
Publica tu parqueo en minutos. Ingresa la información básica y comienza a recibir reservas directamente desde tu celular.

Deployment de Flutter y Kotlin

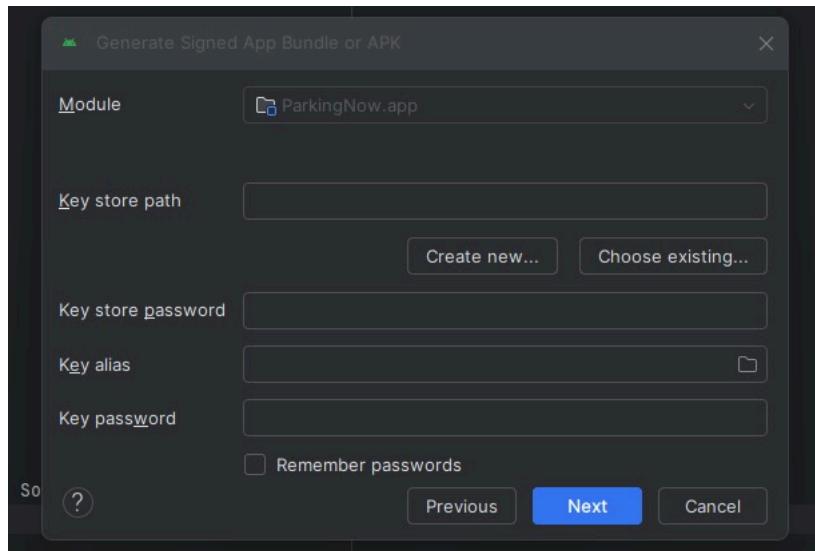
Primero vamos a build y lo que haremos es firmar la aplicación, los requisitos es igual tanto para flutter y kotlin así que se cumplen los mismos pasos y le damos clic en next



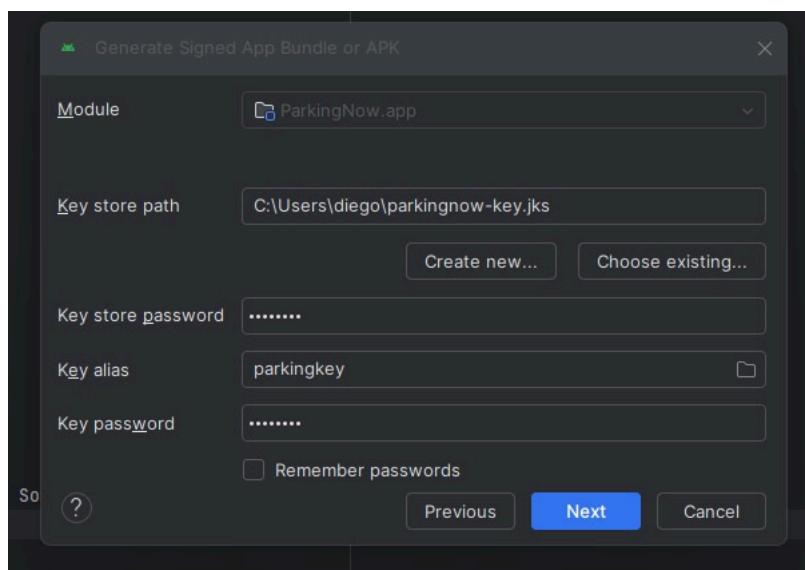
Una vez llenado los datos y completado el formulario le damos clic en okay, todo esto se repite igual en flutter



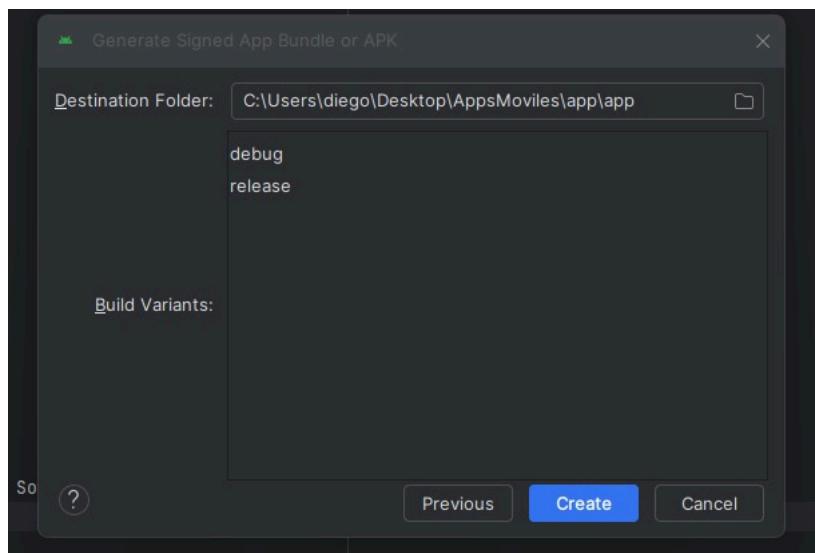
Aquí nos pide en este caso algunos requisitos para poder firmar el apk, le vamos a dar clic en create new



Aquí validamos en este caso que todo este correctamente y coincidan con lo que hemos llenado



Y aquí finalizamos con el proceso y le damos en create, lo mismo con flutter



Aquí nos vamos a app distribution en firebase go to console

Luego aca añadimos y registramos el apk, llenamos todo los campos como kotlin y flutter

× Agregar Firebase a tu app para Android

1 **Registrar app**
Nombre del paquete de Android: pe.edu.upc.parkingnow

2 Descargar y, luego, agregar el archivo de configuración

3 Agregar el SDK de Firebase

4 Próximos pasos

¡Listo!

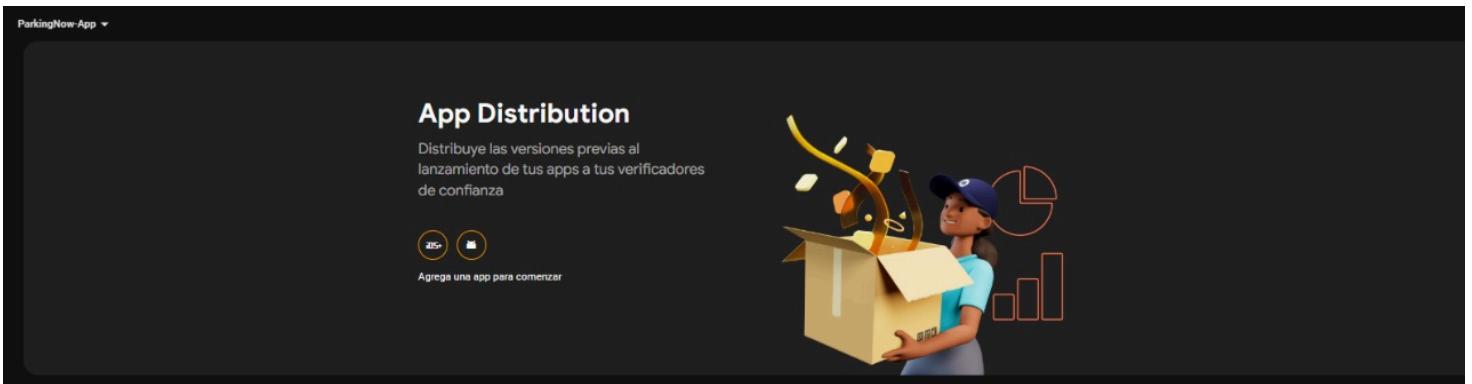
Asegúrate de revisar la [documentación](#) para conocer los primeros pasos con cada producto de Firebase que quieras usar en tu app.

También puedes explorar las [apps de muestra de Firebase](#).

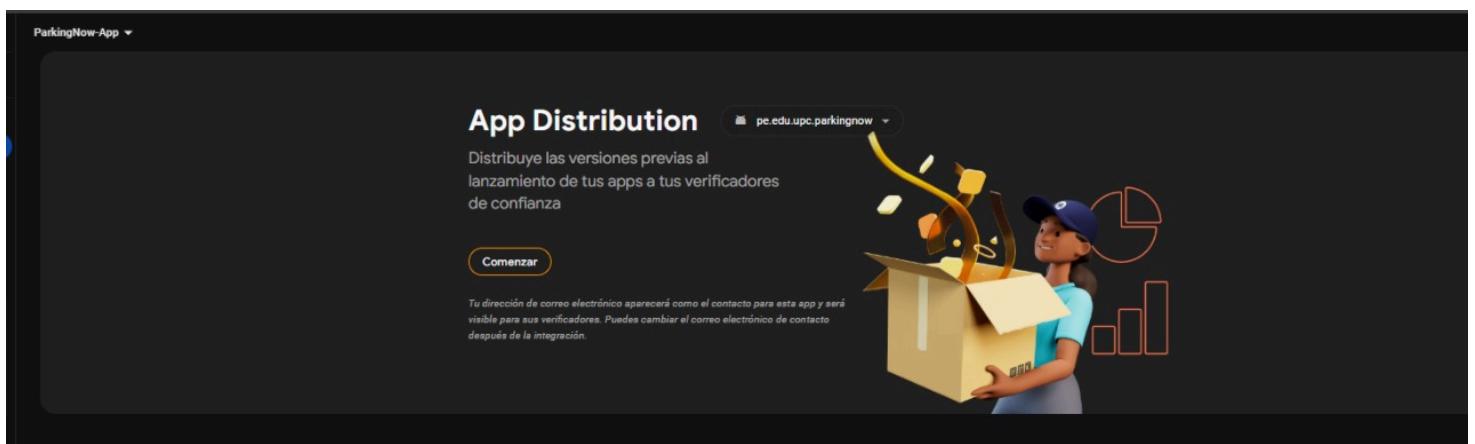
También puedes ir a [console](#) para explorar Firebase

Anterior [Ir a la consola](#)

Seleccionamos el logo de android



Le damos clic en comenzar



Y pues una vez tengamos todo configurado, podremos subir el aplicativo que generamos en el Android Studio e invitar a nuestros compañeros por correo que les llegara para testearlo.

 **Firebase**

ParkingNow Conductor ▾

App Distribution

pe.edu.upc.parkingnow ▾

Versión Vínculos de invitación Verificadores y grupos

Explora el futuro de las pruebas
Obtén acceso anticipado a la función de agente de pruebas de apps y lleva tu app al siguiente nivel.

Aceptar Más información La página se actualizará cuando hagas clic en Habilitar

Arrastra y suelta un archivo APK o AAB para crear una versión nueva. Los archivos AAB requieren vinculación. ⓘ

Más información Explorar

Versión (1)

Correo electrónico de contacto ⓘ d.ulises.sotoquispe.fcalma@gmail.com ⓘ

Buscar versiones y notas

1.0 (1)	Invitado	Aceptada	Descargado	Comentarios
11 de julio de 2025, 10:45:36 p.m. UTC-5	2	1	0	0

Copiar Descargar Borrar

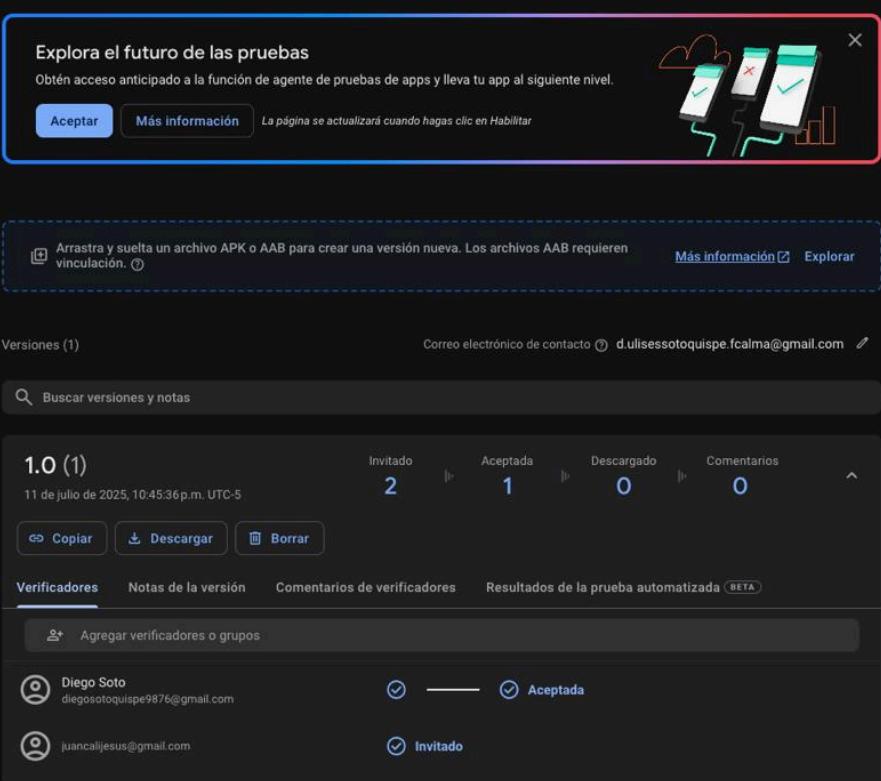
Verificadores Notas de la versión Comentarios de verificadores Resultados de la prueba automatizada (BETA)

Agregar verificadores o grupos

 Diego Soto diegosotoquispe9876@gmail.com   Aceptada

 juancalijesus@gmail.com  Invitado

Herramientas de desarrollo



Una vez le enviamos por correo a nuestros compañeros, solo tendrían que aceptar la invitación



Firebase App Distribution

You've been invited to test:

P

ParkingNow Conductor

 pe.edu.upc.parkingnow

Developer contact:

d.ulises.sotoquispe.fcalma@gmail.com

How to start testing:

- 1 Open this email on your mobile device
 - 2 Using your Google account, accept the **ParkingNow Conductor** invitation
- [Get started](#)
- 3 Enable installations from unknown sources
 - 4 Use Firebase App Distribution to download **ParkingNow Conductor**
 - 5 Start testing!

This invitation is valid for 30 days. If your invitation is expired, [contact the developer](#) to request a new invitation.



Firebase App Distribution helps app developers distribute pre-release apps to testers like you. Need help testing an app? See [tester instructions](#) and [troubleshooting](#).

Google

Google Inc.
1600 Amphitheatre Pkwy
Mountain View, CA, 94043 USA

Una vez aceptado, podran instalar el aplicativo

[← Todas las apps](#)

J



ParkingNow Conductor

pe.edu.upc.parkingnow

Contacto: d.ulises.sotoquispe.fcalma@gmail.com

Versiones

Buscar versiones y sus notas

25 ▾

1.0 (1) Más reciente

11 jul 2025 22:45:36



Comenzó la descarga...

Haz clic en "Abrir" más abajo o revisa tus notificaciones para buscar e instalar el archivo APK.



No hay notas de la versión



Obtén la app de Firebase App Tester para acceder de forma fácil a todas tus apps y versiones de prueba.

[Descargar App Tester](#)

|||

□

<

Prueba que se esta instalando en el dispositivo Mobile

< Instalar aplic. desco... 🔎 :

Instalar aplicaciones de esta fuente puede poner en riesgo su teléfono y sus datos.



Agente de Quick Share

14.77 MB



Bluetooth

824 KB



ParkingNow Conductor

Instalando...

Cancelar



Drive

68.71 MB



Suspensión profunda



Galaxy Store

150 MB



Gmail

159 MB



Internet de Samsung

320 MB





Y finalizamos probando ambas aplicaciones, ya que estan correctamente instaladas.

11:10 📺 📞 📸 • 🔋 ⏱ 🔇 ⌂ ⌂ LTE 32% ⚡



ParkingNow
Dueño

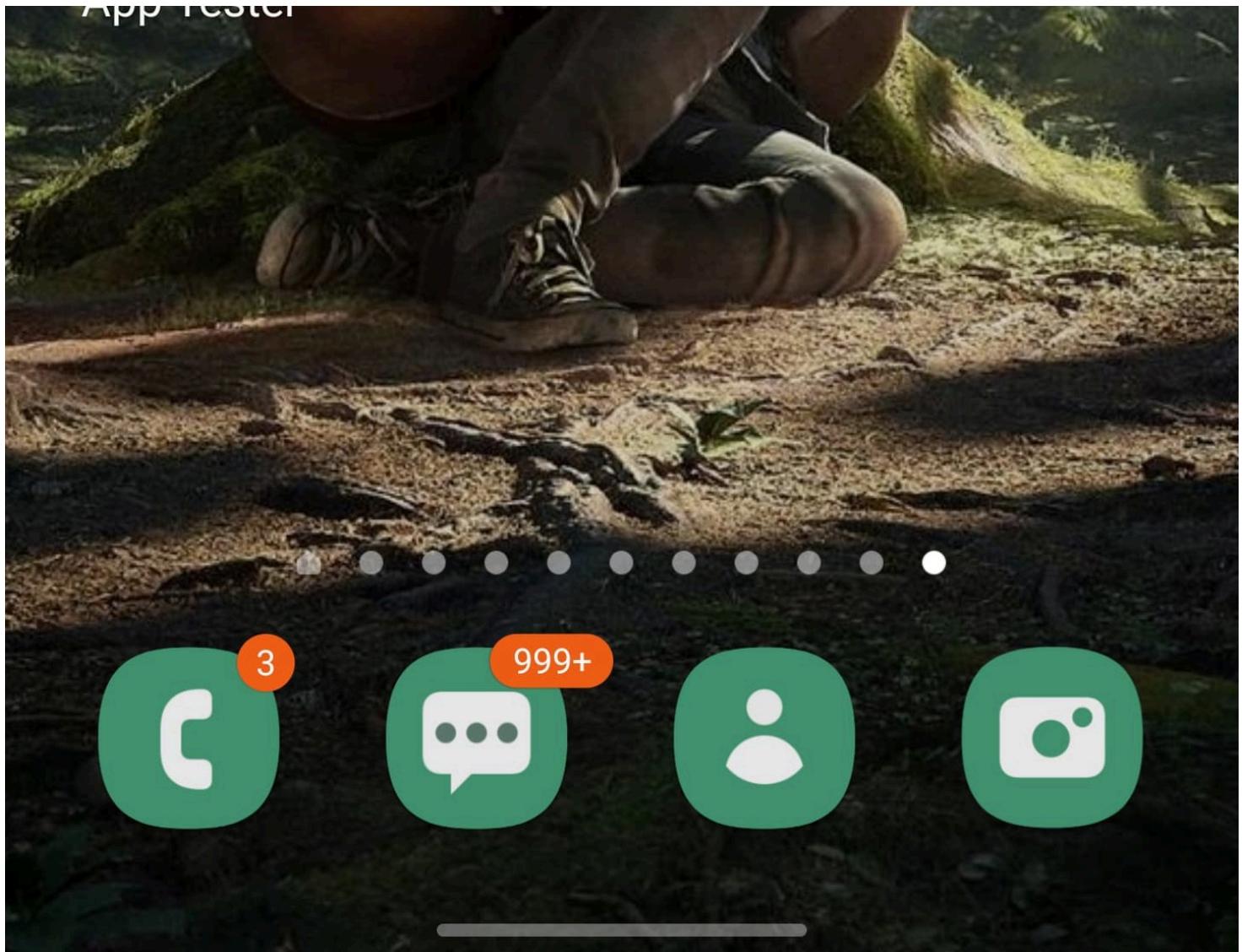


ParkingNow
Conductor



App Tester

App Tester



La app **ParkingNow Conductor** ya está desplegada para pruebas internas.

Enlace: https://appdistribution.firebaseio.google.com/testerapps/1:428034346426:android:7536cb1fe8c9d484bc6183/releases/1g055g04j6f78?utm_source=firebase-console

Este proyecto ha sido desarrollado tanto en **Kotlin** (nativo) como en **Flutter** (multiplataforma).

● Flutter – Segmento Objetivo: Dueño de Estacionamiento

Android Studio

File Edit View Navigate Code Refactor Build Run Tools Git Window Help

Project

new [parkingnow_owner] /Volumes/Universidad/M

build

assets

Flutter

core

features

app

cart

favorites

home

data

domain

presentation

bloc

pages

- dashboard_page.dart
- notifications_page.dart
- register_parking_page.dart
- reservations_page.dart
- security_page.dart
- settings_page.dart

views

wIDGETS

profile

routes

main.dart

test

.flutter-plugins-dependencies

.gitignore

.metadata

analysis_options.yaml

parkingnow_owner.iml

pubspec.lock

pubspec.yaml

README.md

External Libraries

Scratches and Consoles

Select Device

SM G975U1 (mobile)

main.dart

```
void main() {  
    WidgetsFlutterBinding.ensureInitialized();  
  
    final Brightness platformBrightness = WidgetsBinding.instance.platformDispatcher.platformBrightness;  
    SystemChrome.setSystemUIOverlayStyle(  
        SystemUiOverlayStyle(  
            statusBarColor: Colors.transparent,  
            statusBarIconBrightness: platformBrightness == Brightness.dark  
                ? Brightness.light  
                : Brightness.dark,  
        ),  
    );  
  
    runApp(const ParkingNowOwnerApp());  
}  
  
class ParkingNowOwnerApp extends StatelessWidget {  
    const ParkingNowOwnerApp({super.key});  
  
    @override  
    Widget build(BuildContext context) {  
        return ValueListenableBuilder<ThemeMode>(  
            valueListenable: themeModeNotifier,  
            builder: (context, currentMode, child) {  
                return MaterialApp(  
                    debugShowCheckedModeBanner: false,  
                    title: 'ParkingNow',  
                    theme: AppTheme.lightTheme,  
                    darkTheme: AppTheme.darkTheme,  
                    initialRoute: AppRoutes.welcome,  
                    themeMode: currentMode,  
                    routes: {  
                        AppRoutes.welcome: () => const WelcomePage(),  
                        AppRoutes.login: () => const LoginPage(),  
                        AppRoutes.forgotPassword: () => const ForgotPasswordPage(),  
                        AppRoutes.changePassword: () => const ChangePasswordPage(),  
                        AppRoutes.registerOwner: (context) => const RegisterPage(),  
                        AppRoutes.dashboardOwner: () => const DashboardPage(),  
                        AppRoutes.reservations: () => const ReservationsPage(),  
                        AppRoutes.registerParking: () => const RegisterParkingPage(),  
                        AppRoutes.notifications: () => const NotificationsPage(),  
                        AppRoutes.security: () => const SecurityPage(),  
                        AppRoutes.settings: () => const SettingsPage(),  
                    },  
                );  
            },  
        );  
    }  
}
```

2:16 92%

ParkingNow

Panel de Propietarios

Administra tu estacionamiento de manera inteligente

¡Bienvenido de vuelta!

Accede a tu panel de control

Correo electrónico

ejemplo@correo.com

Contraseña

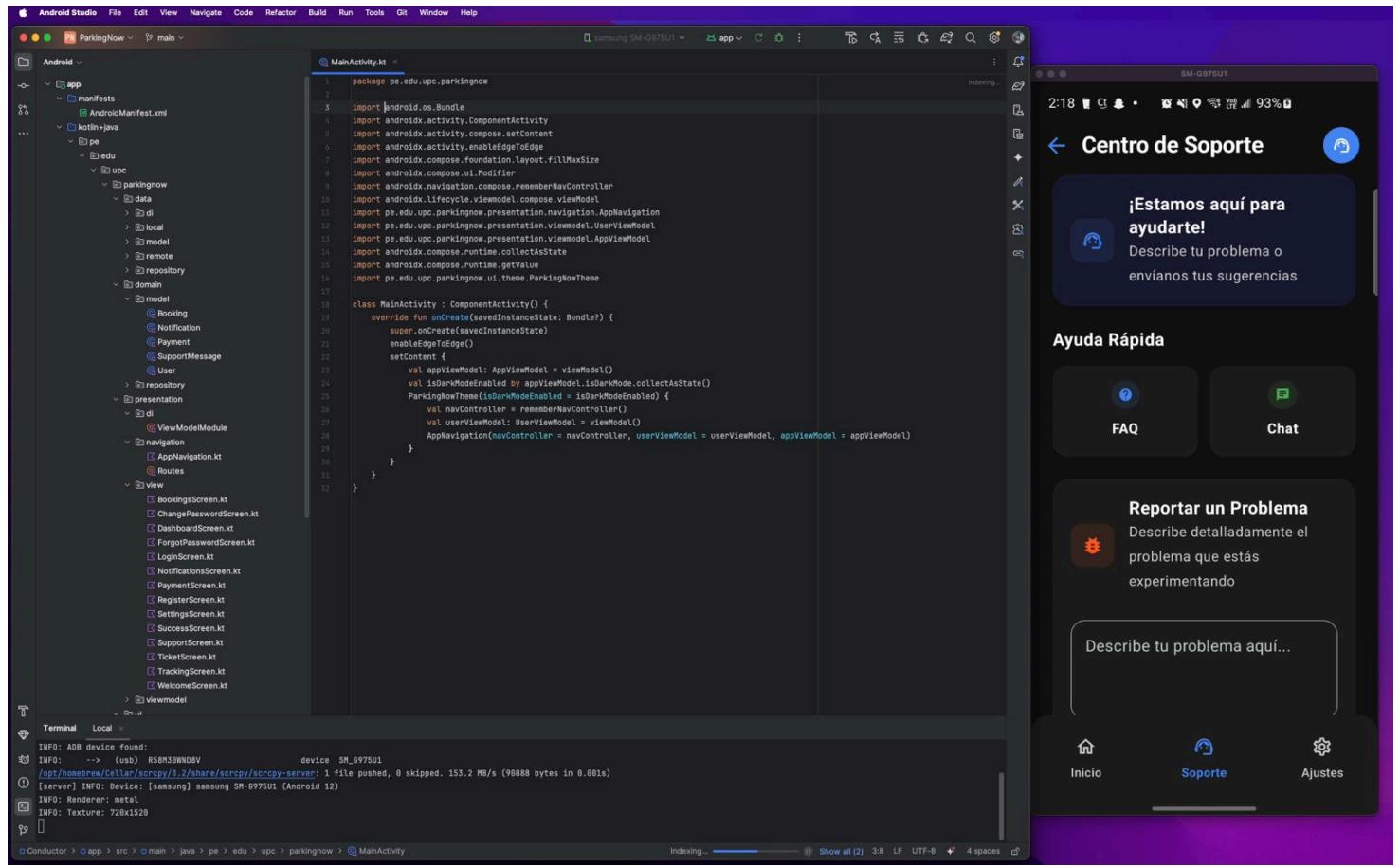
Tu contraseña seg...

Terminal Local Local (2)

```
scrcpy 3.2 <https://github.com/Genymobile/scrcpy>  
INFO: ADB device found:  
INFO: --> (usb) RSMG975U1 device SM_G975U1  
INFO: /opt/homebrew/cellarm/scrcpy/3.2/share/scrcpy/scrcpy-server: 1 file pushed, 0 skipped. 0.8 MB/s (98888 bytes in 0.112s)  
[server] INFO: Device: [samsung] samsung SM-G975U1 (Android 12)  
INFO: Renderer: metal  
INFO: Texture: 720x1528
```

new > lib > main.dart

◆ Kotlin – Segmento Objetivo: Conductor



Backend

Durante este Sprint desplegamos el backend e hicimos las pruebas requeridas

Inicio del despliegue

Empezamos logueandonos en la plataforma

```

Samuel@DESKTOP-JPEIUCT MINGW64 /d/upc/Applicaciones Moviles/Backend (main)
$ npm install --save @nestjs/cli

up to date, audited 895 packages in 17s
163 packages are looking for funding
  run `npm fund` for details

1 low severity vulnerability

To address all issues, run:
  npm audit fix

Run `npm audit` for details.

Samuel@DESKTOP-JPEIUCT MINGW64 /d/upc/Applicaciones Moviles/Backend (main)
$ heroku login
heroku: Press any key to open up the browser to login or q to exit.
Opening browser to https://cli-auth.herokuapp.com/auth/cli/browser/c2ebda91-4842-47c8-b8ca-4e1b729c9e4?requestor=SFMyNTY.g2g0bQAAAAsxNzkuNi4zLjIyNG4GACjK6Y-XAIIAAVG.wQuA0jvUEmrhi3t3vGVn7GY0xXpq2Z90YRykHNqo8
Logged in... done
Logged in as samuelmolina28181@gmail.com

```

Configuracion de la base de datos

Configuramos una base de datos con NEON

```

samue@DESKTOP-JPEIUCT MINGW64 /d/upc/Aplicaciones Moviles/Backend (main)
$ heroku config:set DATABASE_URL="postgresql://neondb_owner:npg_Ytv3go6FPQCM@ep-floral-shadow-a5w34cdf-pooler.us-east-2.aws.neon.tech/neondb?sslmode=require"
Setting DATABASE_URL and restarting ⚡ parkingnow-app... done, v3
DATABASE_URL: postgresql://neondb_owner:npg_Ytv3go6FPQCM@ep-floral-shadow-a5w34cdf-pooler.us-east-2.aws.neon.tech/neondb?sslmode=require

samue@DESKTOP-JPEIUCT MINGW64 /d/upc/Aplicaciones Moviles/Backend (main)
$ 
samue@DESKTOP-JPEIUCT MINGW64 /d/upc/Aplicaciones Moviles/Backend (main)
● $ git status
On branch main
Your branch is up to date with 'origin/main'.

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
    modified:  package-lock.json
    modified:  package.json
    modified:  src/app.module.ts

no changes added to commit (use "git add" and/or "git commit -a")

```

Cambios finales para producción

Se agregan cambios a producción para la ejecución correcta de Nest en la plataforma

```

{} packagejson > {} scripts
8   "scripts": {
9     "build": "nest build",
10    "format": "prettier --write \"src/**/*.{ts,js}\" \"test/**/*.{ts,js}\"",
11    + "start": "node dist/main.js",
12    "start:dev": "nest start --watch",
13    "start:debug": "nest start --debug --watch",
14    "start:prod": "node dist/main",
15    "lint": "eslint \"{src,apps,libs,test}/**/*.{ts,js}\" --fix",
16    "test": "jest",
17    "test:watch": "jest --watch",
18    "test:cov": "jest --coverage",
19    "test:debug": "node --inspect-brk -r tsconfig-paths/register -r ts-node/register node_modules/.bin/jest --runInBand",
20    "test:e2e": "jest --config ./test/jest-e2e.json"
21  },
22  "dependencies": {

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

samue@DESKTOP-JPEIUCT MINGW64 /d/upc/Aplicaciones Moviles/Backend (main)
● $ git status
On branch main
Your branch is up to date with 'origin/main'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    modified:  package-lock.json
    modified:  package.json
    modified:  src/app.module.ts

samue@DESKTOP-JPEIUCT MINGW64 /d/upc/Aplicaciones Moviles/Backend (main)
$ git commit -m "Configuraciones para la conexión a producción a Neon PostgreSQL"
[main 99663ca] Configuraciones para la conexión a producción a Neon PostgreSQL
 3 files changed, 53 insertions(+), 10 deletions(-)

samue@DESKTOP-JPEIUCT MINGW64 /d/upc/Aplicaciones Moviles/Backend (main)
● $ git push heroku main
Enumerating objects: 84, done.
Counting objects: 100% (84/84), done.
Delta compression using up to 8 threads
Compressing objects: 100% (82/82), done.
Writing objects: 100% (84/84), 117.11 KiB | 3.90 MiB/s, done.
Total 84 (delta 37), reused 18 (delta 0), pack-reused 0 (from 0)
remote: Updated 29 paths from eb4b912
remote: Compressing source files... done.
remote: Building source:
remote:
remote: -----> Building on the Heroku-24 stack
remote: -----> Determining which buildpack to use for this app
remote: -----> Node.js app detected
remote:

```

Obtenemos el log del despliegue y comprobamos la conexión a la base de datos

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
remote:     npm notice Changelog: https://github.com/npm/cli/releases/tag/v11.4.2
remote:     npm notice To update run: npm install -g npm@11.4.2
remote:     npm notice
remote:
remote: ----> Build
remote:     Running build
remote:
remote: > backend-parkingnow@0.0.1 build
remote: > nest build
remote:
remote:
remote: ----> Caching build
remote:     - npm cache
remote:
remote: ----> Pruning devDependencies
remote:
remote:     up to date, audited 215 packages in 1s
remote:
remote:     47 packages are looking for funding
remote:       run `npm fund` for details
remote:
remote:     1 low severity vulnerability
remote:
remote:     To address all issues, run:
remote:       npm audit fix
remote:
remote:     Run `npm audit` for details.
remote:     npm notice
remote:     npm notice New major version of npm available! 10.9.2 -> 11.4.2
remote:     npm notice Changelog: https://github.com/npm/cli/releases/tag/v11.4.2
remote:     npm notice To update run: npm install -g npm@11.4.2
remote:     npm notice
remote:
remote: ----> Build succeeded!
remote: ----> Discovering process types
remote:     Procfile declares types    -> (none)
remote:     Default types for buildpack -> web
remote:
remote: ----> Compressing...
remote:     Done: 65M
remote: ----> Launching...
remote:     Released v4
remote:     https://parkingnow-app-963963f523b4.herokuapp.com/ deployed to Heroku
remote:
remote: Verifying deploy... done.
To https://git.heroku.com/parkingnow-app.git
 * [new branch]      main -> main

samue@DESKTOP-JPEIUCT MINGW64 /d/upc/Aplicaciones Moviles/Backend (main)
$ heroku config:get DATABASE_URL
postgresql://neondb_owner:npg_Ytv3go6FPQOM@ep-floral-shadow-a5w34cdf-pooler.us-east-2.aws.neon.tech/neondb?sslmode=require

samue@DESKTOP-JPEIUCT MINGW64 /d/upc/Aplicaciones Moviles/Backend (main)
$ heroku open
```

Comprobacion del despliegue

Revision de los logs de despliegue

```
samue@DESKTOP-JPEIUCT MINGW64 /d/upc/Aplicaciones Moviles/Backend (main)
$ heroku logs --tail
2025-06-21T00:40:20.199533+00:00 app[api]: Release v1 created by user samuelmolina281816@gmail.com
2025-06-21T00:40:20.199533+00:00 app[api]: Initial release by user samuelmolina281816@gmail.com
2025-06-21T00:40:20.402899+00:00 app[api]: Release v2 created by user samuelmolina281816@gmail.com
2025-06-21T00:40:20.402899+00:00 app[api]: Enable Logplex by user samuelmolina281816@gmail.com
2025-06-21T01:16:08.600619+00:00 app[api]: Release v3 created by user samuelmolina281816@gmail.com
2025-06-21T01:16:08.600619+00:00 app[api]: Set DATABASE_URL config vars by user samuelmolina281816@gmail.com
2025-06-21T01:18:25.000000+00:00 app[api]: Build started by user samuelmolina281816@gmail.com
2025-06-21T01:18:52.897281+00:00 app[api]: Deploy 99663ca6 by user samuelmolina281816@gmail.com
2025-06-21T01:18:52.897281+00:00 app[api]: Release v4 created by user samuelmolina281816@gmail.com
2025-06-21T01:18:52.909856+00:00 app[api]: Scaled to web@1:Basic by user samuelmolina281816@gmail.com
2025-06-21T01:18:55.000000+00:00 app[api]: Build succeeded
```

Sin conflicto, despliegue exitoso

Detalles de la aplicación en la plataforma

dashboard.heroku.com/apps/parkingnow-app

HEROKU from Salesforce

Overview Resources Deploy Metrics Activity Access Settings

Get a complete visualization of your app in a team-based continuous delivery environment with [Heroku Pipelines](#). [Hide](#) [Create a Heroku Pipeline](#)

Metrics (last 24hrs) [All Metrics](#)

Response Time
191ms

Throughput
< 1 rps

Memory
12 %

Latest Activity [All Activity](#)

samuelmolina281816@gmail.com: Deployed dd1a4b0d Today at 8:24 PM

samuelmolina281816@gmail.com: Build succeeded Today at 8:24 PM [View build log](#)

samuelmolina281816@gmail.com: Deployed 99663c6b Today at 8:18 PM v4

samuelmolina281816@gmail.com: Build succeeded Today at 8:18 PM [View build log](#)

samuelmolina281816@gmail.com: Set DATABASE_URL config var Today at 8:16 PM v3

samuelmolina281816@gmail.com: Enable Logplex Today at 7:40 PM v2

samuelmolina281816@gmail.com: Initial release Today at 7:40 PM v1

Installed Add-ons [Configure Add-ons](#)

There are no add-ons for this app
You can add add-ons to this app and they will show here. [Learn more](#)

Dyno Formations [Configure Dynos](#)

This app is using basic dynos

web npm start **ON**

Collaborator Activity [Manage Access](#)

samuelmolina281816@gmail.com [2 deploys](#)

Testeo de la aplicación

Testeo de la aplicacion en producción

HTTP <http://localhost:3000/usuario> [Save](#)

POST <https://parkingnow-app-963963f523b4.herokuapp.com/usuario> [Send](#)

Params Authorization Headers (8) **Body** [Pre-request Script](#) [Tests](#) [Settings](#)

None form-data x-www-form-urlencoded raw binary JSON

```

1
2   "name": "Samuel Molina",
3   "email": "samuel@example.com",
4   "password": "Pa$$word963",
5   "placa": "AKL-963",
6   "dni": "12345678",
7   "ruc": "20123456789"
8

```

Body [Cookies](#) [Headers \(11\)](#) [Test Results](#) Status: 201 Created Time: 306 ms Size: 933 B [Save Response](#)

Pretty Raw Preview Visualize JSON

```

1
2   "id": 2,
3   "name": "Samuel Molina",
4   "email": "samuel@example.com",
5   "password": "Pa$$word963",
6   "placa": "AKL-963",
7   "dni": "12345678",
8   "ruc": "20123456789",
9   "tipoUsuario": "dueno_estacionamiento"
10

```

Verificamos la creacion de registros en la base de datos de producción

The screenshot shows the Neon SQL Editor interface. On the left, there's a sidebar with sections for PROJECT (Dashboard, Branches, Integrations, Auth, Settings), BRANCH (production selected), Overview, Monitoring, and SQL Editor (which is currently active). In the center, the SQL Editor window has tabs for Untitled, Save, Primary (ACTIVE), neondb, and a preview icon. Below the tabs, there are two saved queries: "retrieve all user data" (Jun 20, 2025 - 8:36pm) and "list all users" (Jun 20, 2025 - 8:12pm). The main area shows the query "select * from usuario;" and its results. The results table has columns: #, id, name, email, password, placa, dni, ruc, and tipo_usuario. It contains two rows of data.

#	id	name	email	password	placa	dni	ruc	tipo_usuario
1	1	Diego Ulises	diego@example.com	123456	ABC-123	12345678	20123456789	dueno_estacionamiento
2	2	Samuel Molina	samuel@example.com	Pa\$\$word963	AKL-963	12345678	20123456789	dueno_estacionamiento

Comprobacion de las apis abieretas

The screenshot shows a browser developer tools Network tab. A request to "parkingnow-app-963963f523b4.herokuapp.com/usuario" is listed. The status is "OK", the response time is 152ms, and it returned 2 rows of data. The response body is a JSON array containing two user objects:

```
{
  "id": 1,
  "name": "Diego Ulises",
  "email": "diego@example.com",
  "password": "123456",
  "placa": "ABC-123",
  "dni": "12345678",
  "ruc": "20123456789",
  "tipoUsuario": "dueno_estacionamiento"
},
{
  "id": 2,
  "name": "Samuel Molina",
  "email": "samuel@example.com",
  "password": "Pa$$word963",
  "placa": "AKL-963",
  "dni": "12345678",
  "ruc": "20123456789",
  "tipoUsuario": "dueno_estacionamiento"
}
```

Link base de la aplicación: <https://parkingnow-app-963963f523b4.herokuapp.com/>

Endpoint obtencion de usuarios: <https://parkingnow-app-963963f523b4.herokuapp.com/usuario>

Testeo de la aplicacion en producción

The screenshot shows the Postman interface with a successful POST request to the endpoint `https://parkingnow-app-963963f523b4.herokuapp.com/usuario`. The response status is `201 Created`, and the response body contains the newly created user data:

```

1  "id": 2,
2   "name": "Samuel Molina",
3   "email": "samuel@example.com",
4   "password": "Pa$$word963",
5   "placa": "AKL-963",
6   "dni": "12345678",
7   "ruc": "20123456789",
8   "tipoUsuario": "dueno_estacionamiento"
9
10

```

Verificamos la creacion de registros en la base de datos de producción

The screenshot shows the Neon SQL Editor interface. A query has been run to retrieve all user data from the `usuario` table:

```

1 /*
2 In Neon, databases are stored on branches. By default, a project has one branch and one database.
3 You can select the branch and database to use from the drop-down menus above.
4
5 Try generating sample data and querying it by running the example statements below, or click
6 New Query to clear the editor.
7 */
8 select * from usuario

```

The results show two users:

#	id	name	email	password	placa	dni	ruc	tipo_usuario
1	1	Diego Ulises	diego@example.com	123456	ABC-123	12345678	20123456789	dueno_estacionamiento
2	2	Samuel Molina	samuel@example.com	Pa\$\$word963	AKL-963	12345678	20123456789	dueno_estacionamiento

Comprobacion de las apis abiertas

6.2.3.8 Team Collaboration Insights during Sprint

El equipo mantuvo comunicación constante mediante **Discord** y actualizaciones semanales en **Trello**.

Principales logros colaborativos:

- Asignación equitativa de historias entre miembros.
- Sincronización en el diseño de endpoints para evitar duplicidades.
- Apoyo cruzado en pruebas y documentación.
- Compromiso general para cerrar las 17 historias del sprint dentro del tiempo previsto.

Durante este Sprint, el equipo trabajó de manera colaborativa para lograr el desarrollo completo de los tres productos digitales: la Landing Page, el aplicativo móvil y los Web Services. A continuación, se detallan los aportes por cada sección, incluyendo espacio para evidencias de commits y capturas relevantes.

Reporte

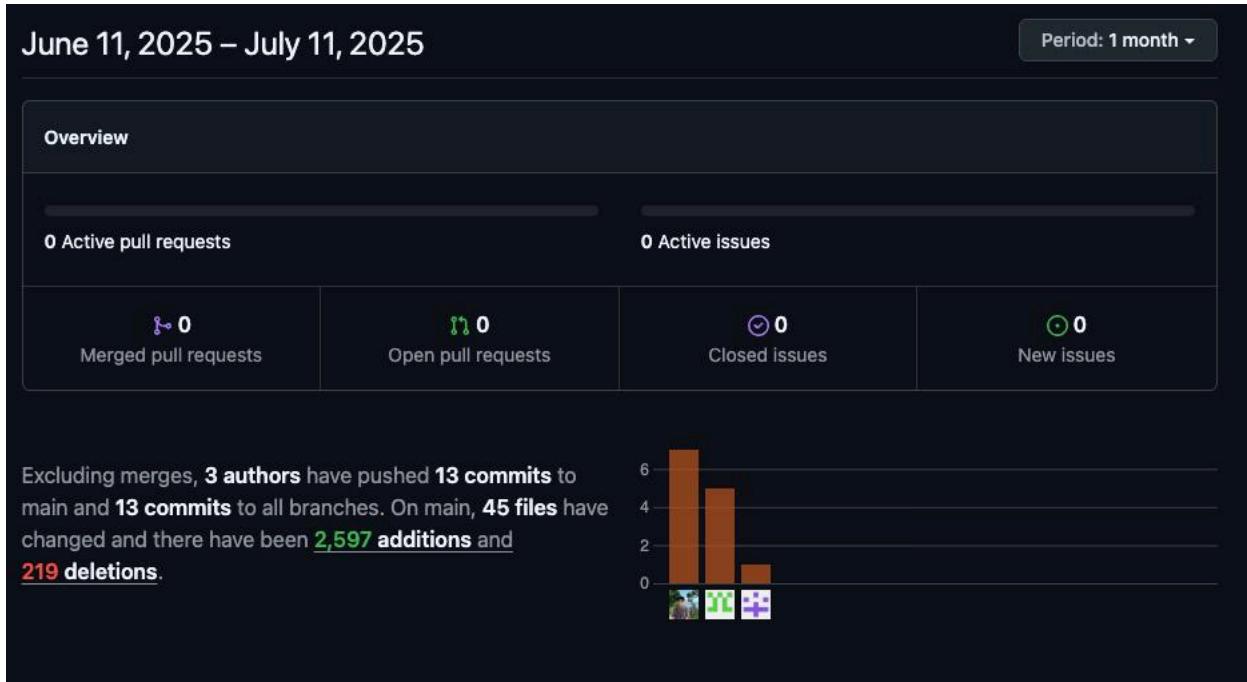


Landing Page

A screenshot of the GitHub landing page showing a list of recent commits. The commits are listed in chronological order from most recent to oldest. The commits are as follows:

- Fix embed YouTube videos
- Diego2005z pushed 1 commit to main · 0d46d4e...15a4d98 · 2 hours ago
- Fix i18n translation issues and update HTML entities
- Diego2005z pushed 1 commit to main · 82a13ed...0d46d4e · yesterday
- Add landing page with updated README and styles (Force push)
- Diego2005z force pushed to main · 41d1e52...82a13ed · 2 days ago
- Initial commit
- Diego2005z created main · 41d1e52 · 2 days ago

Kotlin



Flutter

Activity

All branches All activity All users All time Showing most recent first

Add APK JuanCali999 created feature/register-parking · 2efa9fe · 23 hours ago

Add APK JuanCali999 created develop · 2efa9fe · 23 hours ago

Add APK Diego2005z pushed 1 commit to main · b694f82...2efa9fe · 9 days ago

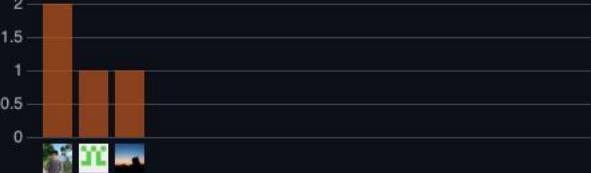
fix: update launcher icon configuration and generate icons Diego2005z pushed 1 commit to main · c42bece...b694f82 · 11 days ago

Fix: default theme mode now respects system setting Diego2005z pushed 1 commit to main · b4aa3de...c42bece · 11 days ago

Fix: full dark mode applied in Security UI Diego2005z pushed 1 commit to main · 7920a4b...b4aa3de · 11 days ago

Clean up alert cards and remove subscription dialog Diego2005z pushed 1 commit to main · 22524bb...7920a4b · 12 days ago

Excluding merges, **3 authors** have pushed **2 commits** to main and **4 commits** to all branches. On main, **9 files** have changed and there have been **494 additions** and **28 deletions**.



The chart shows three bars representing file changes for three authors: JuanCali999, Diego2005z, and another author whose name is partially visible. The y-axis represents the number of files changed, ranging from 0 to 2.

Author	Files Changed
JuanCali999	2
Diego2005z	1
Others	1

Backend



El trabajo conjunto permitió cumplir con el **Sprint Goal** y sentar las bases para las funcionalidades finales del proyecto.

6.3. Validations Interviews - Sprint 2 - TB2

Como parte del proceso de validación del producto, se realizaron entrevistas con usuarios potenciales de ambos segmentos: conductores y dueños de estacionamientos. El objetivo fue recoger retroalimentación temprana sobre la experiencia de uso, funcionalidades clave y percepciones generales de la propuesta de valor de ParkingNow.

Estas entrevistas permitieron identificar necesidades reales, validar hipótesis de diseño, y orientar ajustes tanto en el flujo de navegación como en la priorización de futuras funcionalidades.

6.3.1. Diseño de Entrevistas

Segmento Objetivo #1: Conductores Urbanos Frecuentes

Preguntas personales:

- ¿Cuál es tu nombre?
- ¿Cuántos años tienes?

Preguntas principales:

- ¿Sentiste que lograste navegar por la app sin complicaciones?
- ¿Qué opinas acerca del diseño visual y la presentación de la app?
- ¿Tuviste algún problema al buscar estacionamientos cercanos en el mapa?
- ¿El proceso de reserva te pareció claro y eficiente?
- ¿Tuviste dificultades para completar el pago desde la app?
- ¿Consideras útil la funcionalidad del código QR generado al finalizar una reserva?
- ¿Las notificaciones dentro de la app te resultaron claras y oportunas?
- ¿Tienes alguna sugerencia o comentario que deseas compartir para mejorar la experiencia?

Segmento Objetivo #2: Dueños de Estacionamientos

Preguntas personales:

- ¿Cuál es tu nombre?
- ¿Cuántos años tienes?

Preguntas principales:

- ¿Pudiste navegar correctamente por la app como dueño de estacionamiento?
- ¿Qué opinas del diseño de la interfaz desde tu rol de administrador?
- ¿Tuviste algún inconveniente al registrar o editar información de tus estacionamientos?
- ¿Fue sencillo colocar la ubicación geográfica de tu estacionamiento en el mapa?
- ¿La funcionalidad para establecer horarios y tarifas por hora fue fácil de utilizar?
- ¿Sientes que la sección de reservas te permite tener control sobre tu disponibilidad?
- ¿Te pareció clara la sección de configuración de servicios y condiciones del local?
- ¿Tienes alguna sugerencia o comentario sobre cómo mejorar la administración desde la app?

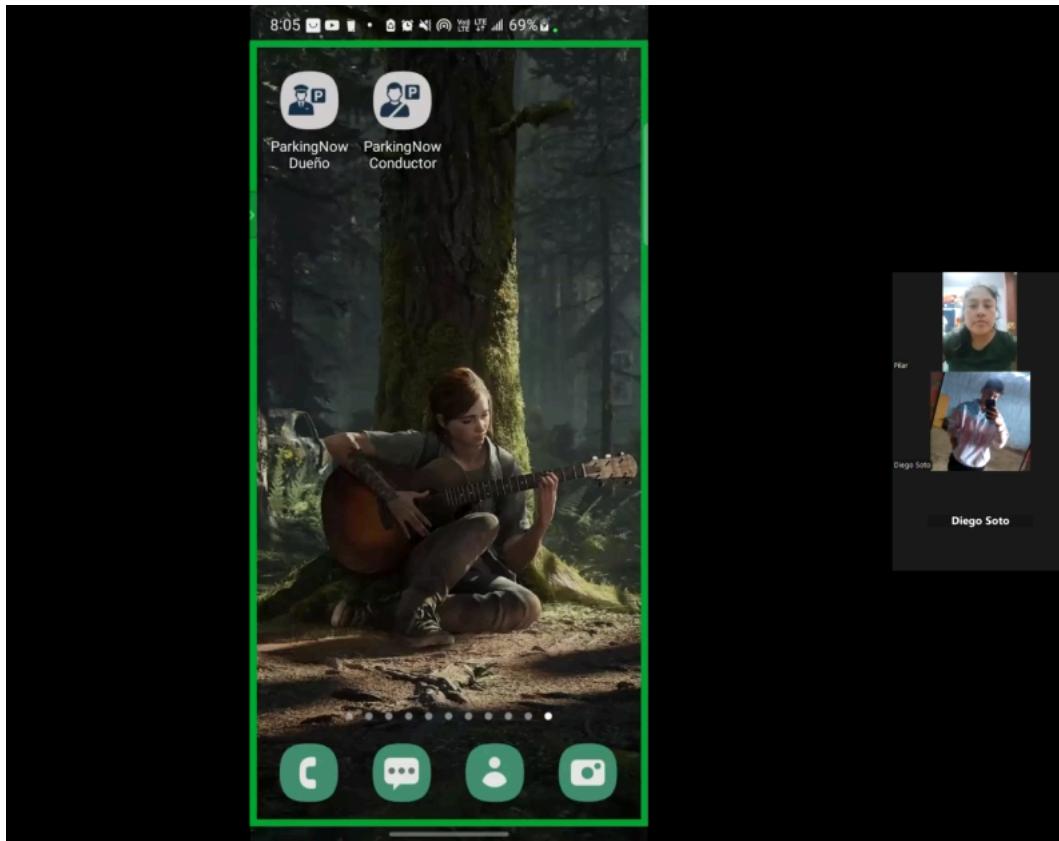
Estas entrevistas permitieron identificar mejoras clave en la experiencia de usuario, detectar puntos de fricción en la navegación y recopilar comentarios para futuras iteraciones de la aplicación.

6.3.2. Registro de Entrevistas

Entrevistada – Segmento Conductor

Entrevistadas: Pilar – Segmento Conductor

Pilar de 30 años destacó la **facilidad de navegación** dentro de la app, especialmente al buscar estacionamientos en el mapa. Elogió el **diseño limpio y moderno**, y señaló que el proceso de **reserva y pago fue rápido y sin errores**. Consideró muy útil el uso del **código QR para validar la reserva**. Como sugerencias, propuso incluir un **tutorial inicial para nuevos usuarios** y la posibilidad de **compartir reservas activas con otros usuarios**, aunque resaltó que la app es muy intuitiva en general.



👤 Entrevistada: Marycielo Sánchez Soto – Segmento Conductor

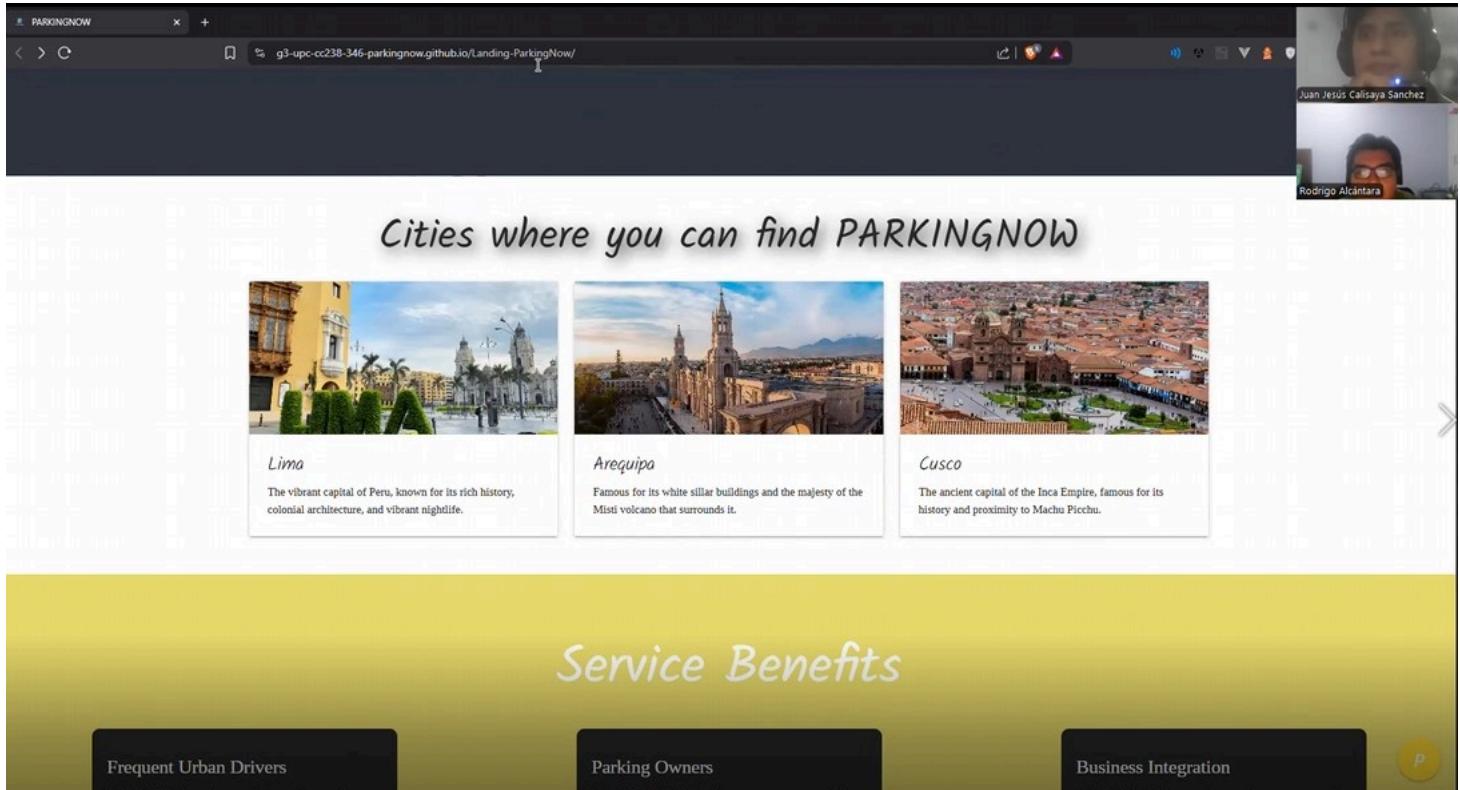
Marycielo Sánchez Soto, de 21 años, comentó que la aplicación le pareció completa y destacó positivamente la opción de poder contactar a un oficial policial en caso de robo o daño. En cuanto a la navegación, mencionó que pudo usarla sin dificultades, ya que es intuitiva y fácil de utilizar.

Respecto al diseño, señaló que es amigable y valoró especialmente la inclusión del modo oscuro. También le agració el proceso de reserva de espacios de estacionamiento por su rapidez y eficiencia. En relación al pago, no presentó inconvenientes y resaltó la opción de pagar con Yape mediante código QR, por su rapidez y practicidad.

Como sugerencia, indicó que sería ideal mejorar la visualización del mapa para que se refleje correctamente.

A screenshot of the ParkingNow mobile application showing the 'Seguimiento' (Tracking) screen. The top section displays the 'Estado del Sistema' (System Status) as 'Monitoreo activo' (Active Monitoring) with a green shield icon. Below this are three summary statistics: 2 Vehículos (Vehicles), 4 Cámaras (Cameras), and 0 Alertas (Alerts). The main area is titled 'Vehículos Registrados' (Registered Vehicles) and lists two vehicles: a 'Toyota Corolla' with Placa ABC-123, currently 'Estacionado' (parked) at 'Real Plaza Salverry - Nivel 2' (Real Plaza Salverry - Level 2) 'Hace 5 minutos' (5 minutes ago); and a 'Honda Civic' with Placa XYZ-789, currently 'En tránsito' (in transit) on 'Av. Javier Prado Este' (Av. Javier Prado Este) 'Hace 2 minutos' (2 minutes ago). At the bottom is a section titled 'Monitoreo por Cámara' (Camera Monitoring) which says 'Vigilancia 24/7 de tus vehículos con tecnología de reconocimiento' (24/7 vehicle surveillance with recognition technology). A blue button labeled 'Ver cámaras' (View cameras) is located here. At the very bottom of the screen are three navigation icons: 'Inicio' (Home), 'Seguimiento' (Monitoring), and 'Ajustes' (Settings).

Entrevistada: Alcántara Cruz, Rodrigo Alonso – Segmento Conductor - Landing Page



Cities where you can find PARKINGNOW

Lima
The vibrant capital of Peru, known for its rich history, colonial architecture, and vibrant nightlife.

Arequipa
Famous for its white sillar buildings and the majesty of the Misti volcano that surrounds it.

Cusco
The ancient capital of the Inca Empire, famous for its history and proximity to Machu Picchu.

Service Benefits

Frequent Urban Drivers Parking Owners Business Integration

Alcántara Cruz, Rodrigo Alonso, de 20 años, comentó que la landing page le pareció muy intuitiva y que logró comprender claramente el propósito de la aplicación gracias al mensaje principal, el cual transmite de forma directa la idea central del servicio.

Destacó que el diseño le parece moderno y bien distribuido, y que las imágenes utilizadas ayudan a contextualizar la propuesta. Además, las ciudades destacadas en la página le transmitieron una sensación de realismo y confianza, especialmente al mencionar aspectos relacionados con la seguridad.

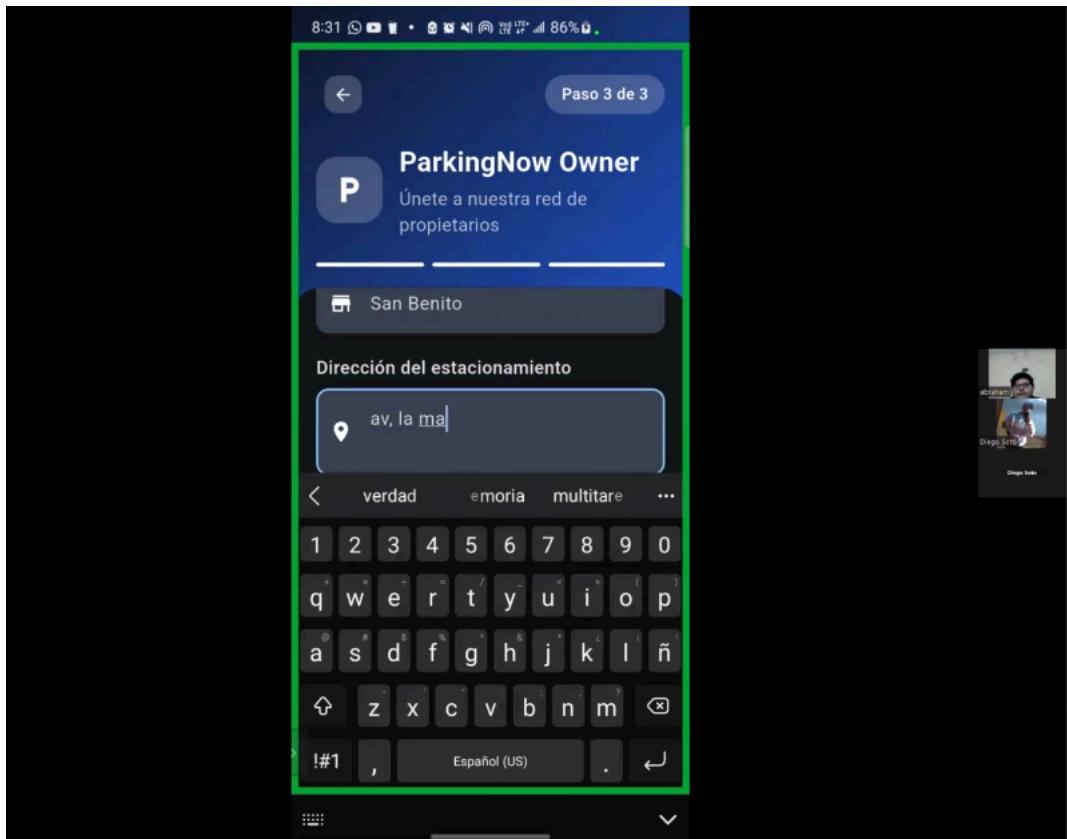
También valoró positivamente la diferenciación entre los botones dirigidos a conductores y a dueños de estacionamientos, lo cual facilita la navegación según el rol del usuario.

Finalmente, expresó que se animaría a unirse a la aplicación, ya que considera que en Lima es muy estresante encontrar un espacio disponible para estacionar, y esta solución le parece útil frente al tráfico y la falta de disponibilidad.

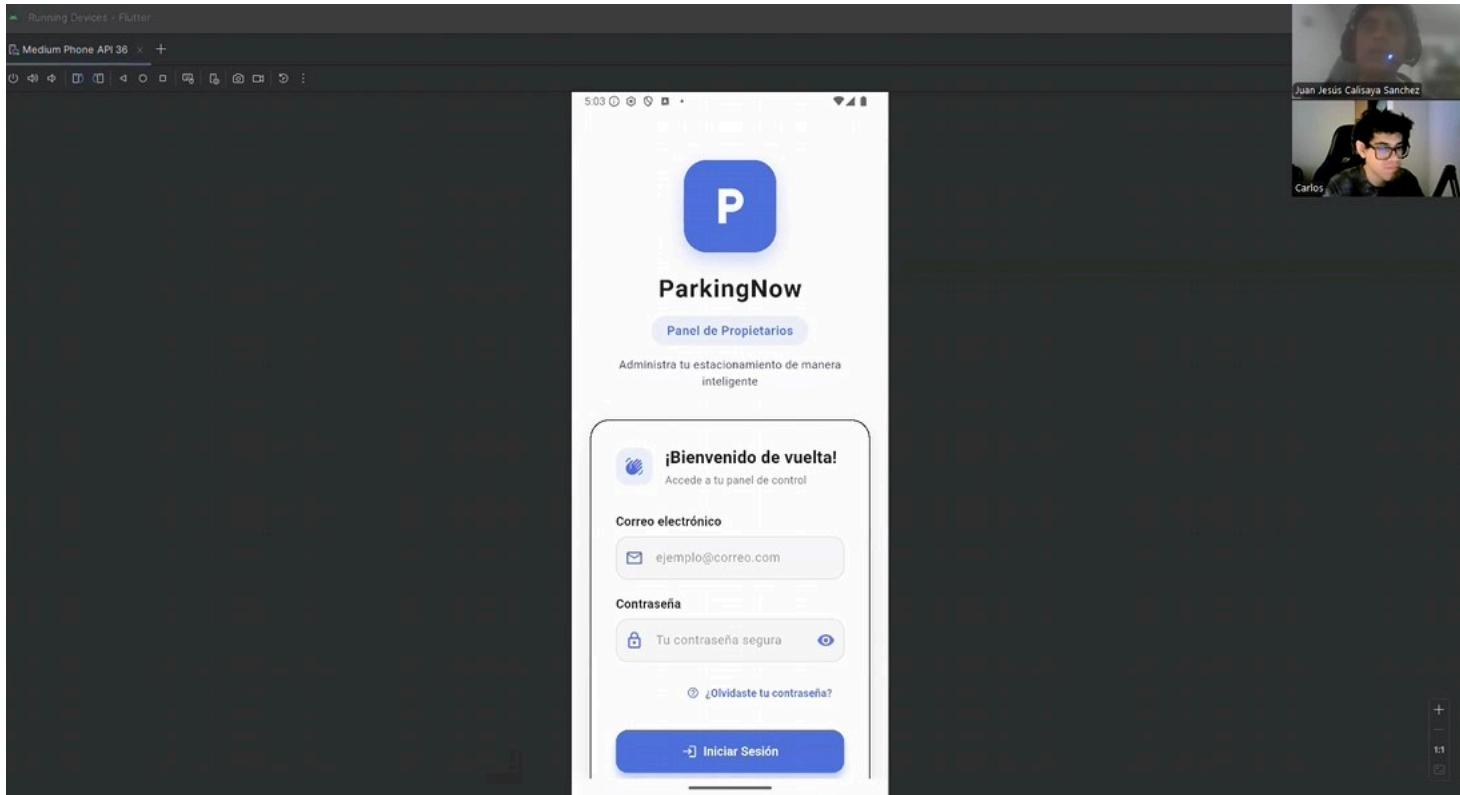
Entrevistada – Segmento Dueño de Estacionamiento

Entrevistado: Abraham Estrada – Segmento Dueño de Estacionamiento

Abraham, de 21 años, comentó que la app le pareció **muy clara y funcional** desde el primer uso. Destacó que el proceso para **registrar su estacionamiento, ubicarlo en el mapa y establecer tarifas** fue sencillo y directo. Valoró positivamente el **control de reservas** desde el dashboard y la opción para **gestionar los servicios ofrecidos**. Como mejora, sugirió incorporar un **sistema de respuestas automáticas para consultas frecuentes**, de modo que los conductores puedan recibir atención inmediata sobre horarios, tarifas o normas del local sin necesidad de intervención directa.



Entrevistado: Carlos Ochoa – Segmento Dueño de estacionamiento



Carlos Ochoa, de 25 años, comentó que la navegación dentro de la aplicación fue sencilla gracias a que la interfaz es intuitiva y está bien estructurada. Destacó que los botones grandes y las fuentes legibles hacen que el uso sea cómodo y visualmente agradable.

Respecto al registro y edición de su estacionamiento, indicó que no tuvo problemas y que fue fácil ubicar su dirección en el mapa. También mencionó que ingresar datos como tarifas y horarios fue un proceso sencillo.

Valoró positivamente el control que la aplicación le brinda sobre sus reservas de espacios, y señaló que la sección de notificaciones es clara, con botones grandes que facilitan no pasar por alto ningún detalle.

Como sugerencia, mencionó que el logo de la aplicación le parece demasiado simple, ya que consiste únicamente en una "P".

Link de las entrevistas: https://upcedupe-my.sharepoint.com/:v/g/personal/u202213222_upc_edu_pe/EQ4vKYPrxpNOrT7tJGKZidwBA2jaipNS-bhcYoqQpgUVFg?e=NkeEAJ&nav=eyJyZWlcnJhbEluZm8iOnsicmVmZXJyYWxBcHAIoJTdHJIYW1XZWJBcHAIoLCJyZWlcnJhbFZpZXciOjTaGFyZURpYWxvZy1MaW5rlwiVmZXJyYWxBcHBQbGF0Zm9ybSI6IldlYiIsInJZmVycmFsTW9kZSI6InZpZXciF0%3D

6.3.3. Evaluaciones según heurísticas

SITE O APP A EVALUAR:

ParkingNow – Aplicación móvil para conductores y dueños de estacionamiento.

TAREAS A EVALUAR:

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

Segmento Conductor:

1. Registro de un nuevo conductor
2. Inicio de sesión
3. Recuperación y cambio de contraseña
4. Búsqueda de estacionamientos cercanos en el mapa
5. Reserva de un espacio de estacionamiento
6. Confirmación de reserva con generación de código QR
7. Pago mediante método en línea
8. Visualización del historial de reservas
9. Gestión de notificaciones
10. Acceso a soporte técnico desde el dashboard
11. Modificación de preferencias del sistema

Segmento Dueño de Estacionamiento:

1. Registro como nuevo usuario dueño
2. Inicio de sesión
3. Recuperación y cambio de contraseña
4. Registro de un estacionamiento nuevo
5. Ubicación del estacionamiento en el mapa
6. Configuración de horarios de atención y servicios ofrecidos
7. Establecimiento de tarifas por hora
8. Gestión y visualización de reservas recibidas
9. Módulo de seguridad (alertas, control de acceso)
10. Centro de notificaciones
11. Barra lateral para navegación entre módulos
12. Revisión de términos y condiciones legales

No están incluidas en esta versión de la evaluación las siguientes tareas:

- Mostrar un tutorial inicial para usuarios nuevos
- Compartir reservas activas entre usuarios
- Implementar respuestas automáticas para preguntas frecuentes del conductor

ESCALA DE SEVERIDAD:

Nivel	Descripción
1	Problema superficial: fácilmente superado por el usuario

Nivel	Descripción
2	Problema menor: ocurre ocasionalmente, requiere revisión
3	Problema mayor: impide el uso correcto de funciones clave
4	Problema muy grave: bloquea el uso o continuidad de la app

TABLA RESUMEN:

#	Problema detectado	Escala de Severidad	Heurística/Principio violado
1	En la sección de soporte no hay un buscador de preguntas frecuentes	2	Ayuda y documentación
2	No hay opción clara para editar una reserva confirmada	3	Flexibilidad y eficiencia del uso
3	No se muestra confirmación visual tras actualizar configuraciones importantes	1	Visibilidad del estado del sistema

DESCRIPCIÓN DE PROBLEMAS:

PROBLEMA #1: En la sección de soporte no hay un buscador de preguntas frecuentes

Severidad: 2

Heurística violada: Ayuda y documentación

Descripción:

Aunque existe una sección para soporte, no hay un campo de búsqueda ni sugerencias automáticas para preguntas frecuentes. Esto puede dificultar encontrar ayuda rápida en problemas comunes.

← Centro de Soporte



¡Estamos aquí para ayudarte!



Describe tu problema o envíanos tus sugerencias

Ayuda Rápida



FAQ



Chat

Reportar un Problema



Describe detalladamente el problema que estás experimentando

Describe tu problema aquí...



Inicio



Soporte



Ajustes

Recomendación:

Incluir una barra de búsqueda con autocompletado que filtre las preguntas frecuentes o cree una base de conocimientos básica.

PROBLEMA #2: No hay opción clara para editar una reserva confirmada

Severidad: 3

Heurística violada: Flexibilidad y eficiencia del uso

Descripción:

Actualmente, una vez realizada la reserva, no se ofrece una opción directa para modificar la hora, el estacionamiento o los datos de la reserva. El único camino es cancelarla y volver a crearla desde cero, lo que afecta la eficiencia.

← Mis Reservas



Reservas Activas

Gestiona tus reservas actuales

[Ver reservas activas](#)



Nueva Reserva

Encuentra y reserva un estacionamiento

[Realizar reserva](#)



Historial

Revisa tus reservas anteriores

[Ver historial](#)



Inicio



Reservas



Ajustes

Recomendación:

Agregar un botón “Editar reserva” dentro del detalle de cada reserva activa, permitiendo modificar datos como hora o estacionamiento sin cancelar.

PROBLEMA #3: No se muestra confirmación visual tras actualizar configuraciones importantes

Severidad: 1

Heurística violada: Visibilidad del estado del sistema

Descripción:

Al modificar preferencias del sistema como idioma o tema, la app no muestra un mensaje de confirmación visual, lo que puede generar dudas al usuario sobre si se guardaron correctamente los cambios.

1:05 Ct

55%



Configuraciones

Personaliza tu experiencia



Juan Carlos Pérez

Propietario Premium



Cuenta Verificada

General



Modo Oscuro

Cambia la apariencia de la aplicación



Idioma

Español



Moneda

PEN (S/)



Zona Horaria

GMT-5 (Lima)



Notificaciones

Recomendación:

Añadir un snack bar o mensaje emergente breve que indique que los cambios fueron aplicados exitosamente.

6.3.1 Validations Interviews - Sprint 3 - Trabajo Final

Como parte del proceso de validación del producto, se realizaron entrevistas con usuarios potenciales de ambos segmentos: conductores y dueños de estacionamientos. El objetivo fue recoger retroalimentación temprana sobre la experiencia de uso, funcionalidades clave y percepciones generales de la propuesta de valor de ParkingNow.

Estas entrevistas permitieron identificar necesidades reales, validar hipótesis de diseño, y orientar ajustes tanto en el flujo de navegación como en la priorización de futuras funcionalidades.

6.3.1.1 Diseño de Entrevistas

Segmento Objetivo #1: Conductores Urbanos Frecuentes

Preguntas personales:

- ¿Cuál es tu nombre?
- ¿Cuántos años tienes?

Preguntas principales:

- ¿Te sentiste guiado en cada paso del proceso dentro de la app?
- ¿Te hubiera sido útil ver recomendaciones personalizadas de parqueos?
- ¿Qué opinas del tiempo de respuesta al momento de hacer una reserva?
- ¿Sientes que la app ayuda a ahorrar tiempo en tu rutina diaria?
- ¿Percibiste alguna funcionalidad innecesaria o que te confundiera?
- ¿Qué mejorarías específicamente en la parte de pagos o QR?

Segmento Objetivo #2: Dueños de Estacionamientos

Preguntas personales:

- ¿Cuál es tu nombre?
- ¿Cuántos años tienes?

Preguntas principales:

- ¿Te resultó útil el resumen de ingresos diarios y semanales?
- ¿Pudiste entender claramente los porcentajes de ocupación de tus espacios?
- ¿Has usado la función de monitoreo o alertas? ¿Te pareció útil?
- ¿Consideras que la app te ayuda a reducir el tiempo de gestión?
- ¿Te gustaría ver reportes descargables o estadísticas más detalladas?
- ¿La app refleja fielmente el estado real de tu parqueo?
- ¿Qué funcionalidad agregarías para mejorar la administración desde tu celular?

Evaluación de la Landing Page – ParkingNow

Preguntas personales:

- ¿Cuál es tu nombre?
- ¿Cuántos años tienes?

Preguntas principales:

- ¿El diseño de la página te pareció atractivo y profesional?
- ¿Pudiste entender rápidamente de qué trata la app?
- ¿La información está bien organizada o sentiste que faltaba algo?
- ¿Te pareció útil ver capturas o demostraciones de la app dentro del sitio?
- ¿Te motivaría registrarte o descargar la app después de visitar la página?

- ¿Qué mejorarías visual o funcionalmente en esta landing?
- ¿Qué contenido agregarías para convencer a más usuarios como tú?

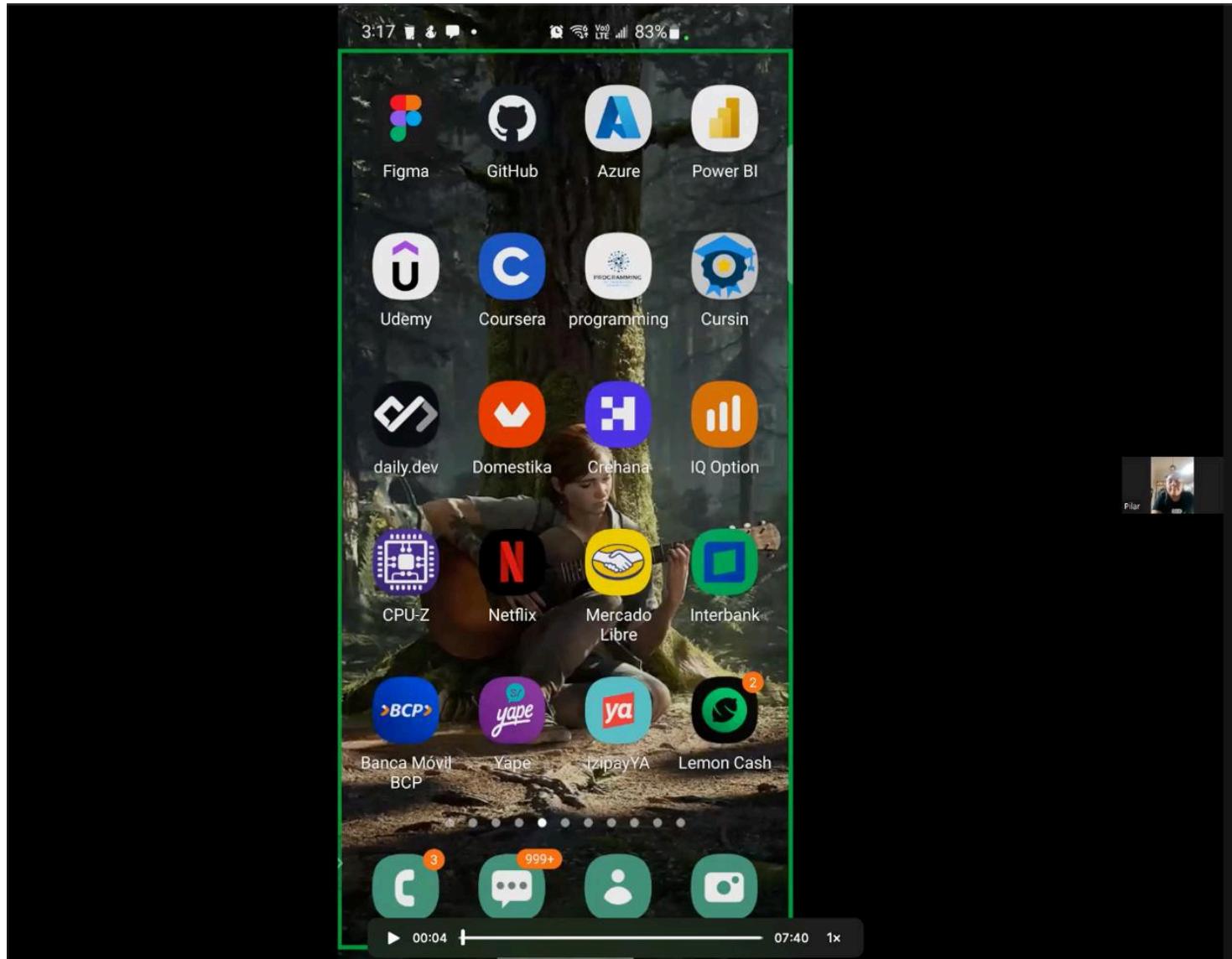
Estas entrevistas permitieron identificar mejoras clave en la experiencia de usuario, detectar puntos de fricción en la navegación y recopilar comentarios para futuras iteraciones de la aplicación.

6.3.2.1 Registro de Entrevistas

Entrevistada – Segmento Conductor

Entrevistadas: Pilar – Segmento Conductor

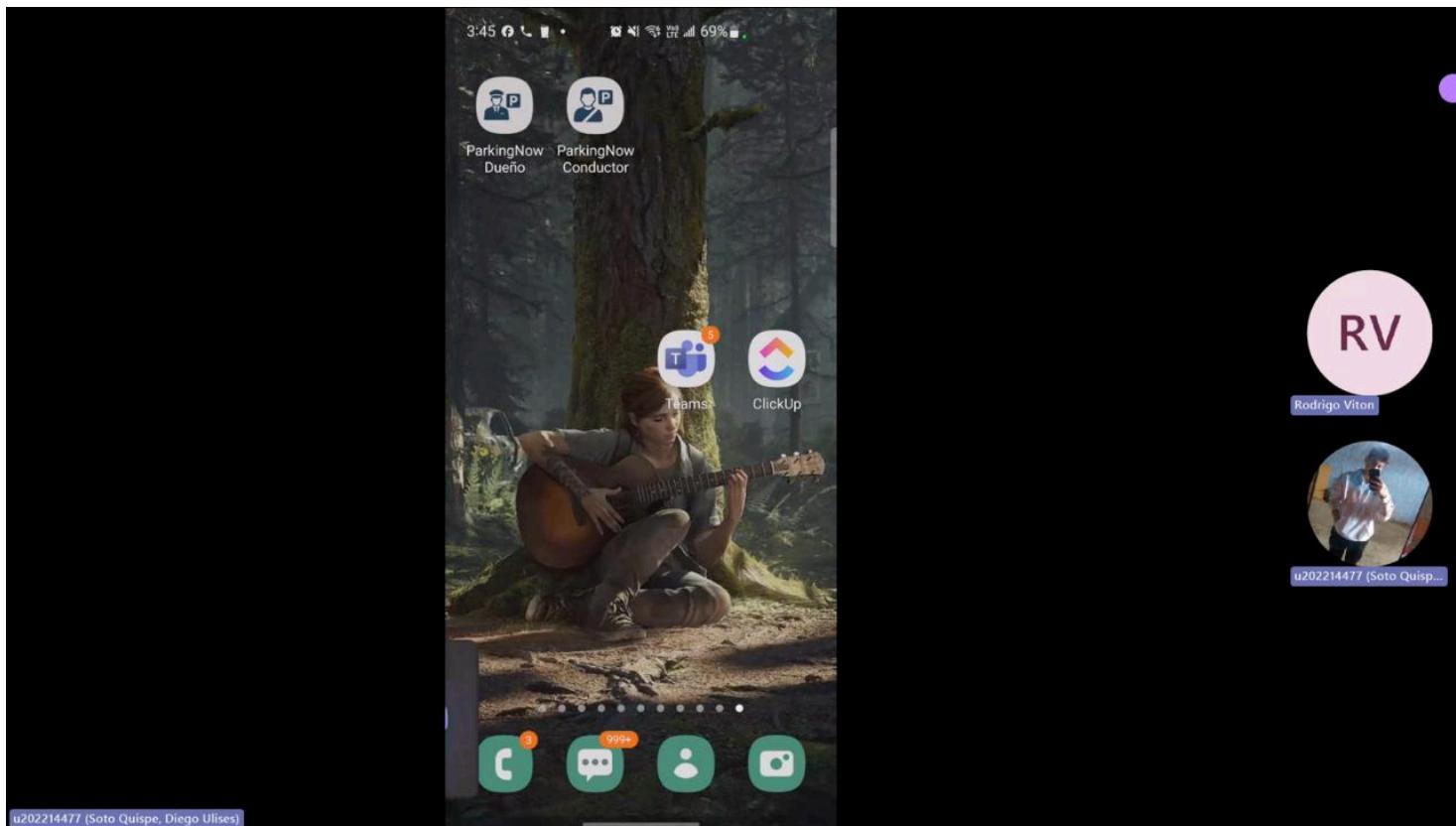
Pilar consideró que la app es clara y fácil de usar, con un flujo bien guiado. Valoró la rapidez del proceso de reserva y cómo le ayuda a ahorrar tiempo. Sugirió mejorar la visualización de funciones en el mapa, ofrecer más métodos de pago como Plin y dar instrucciones más claras para el uso del código QR.



Entrevistada: Rodrigo Vitor – Segmento Conductor

Rodrigo tuvo una experiencia positiva tras entender el flujo inicial. Resaltó la rapidez en la reserva y la utilidad de la app para moverse por zonas complicadas.

Recomendó permitir guardar o compartir el código QR y simplificar la interfaz ocultando funciones innecesarias para usuarios ocasionales.



👤 Entrevistada: Alcántara Cruz, Rodrigo Alonso - Landing Page

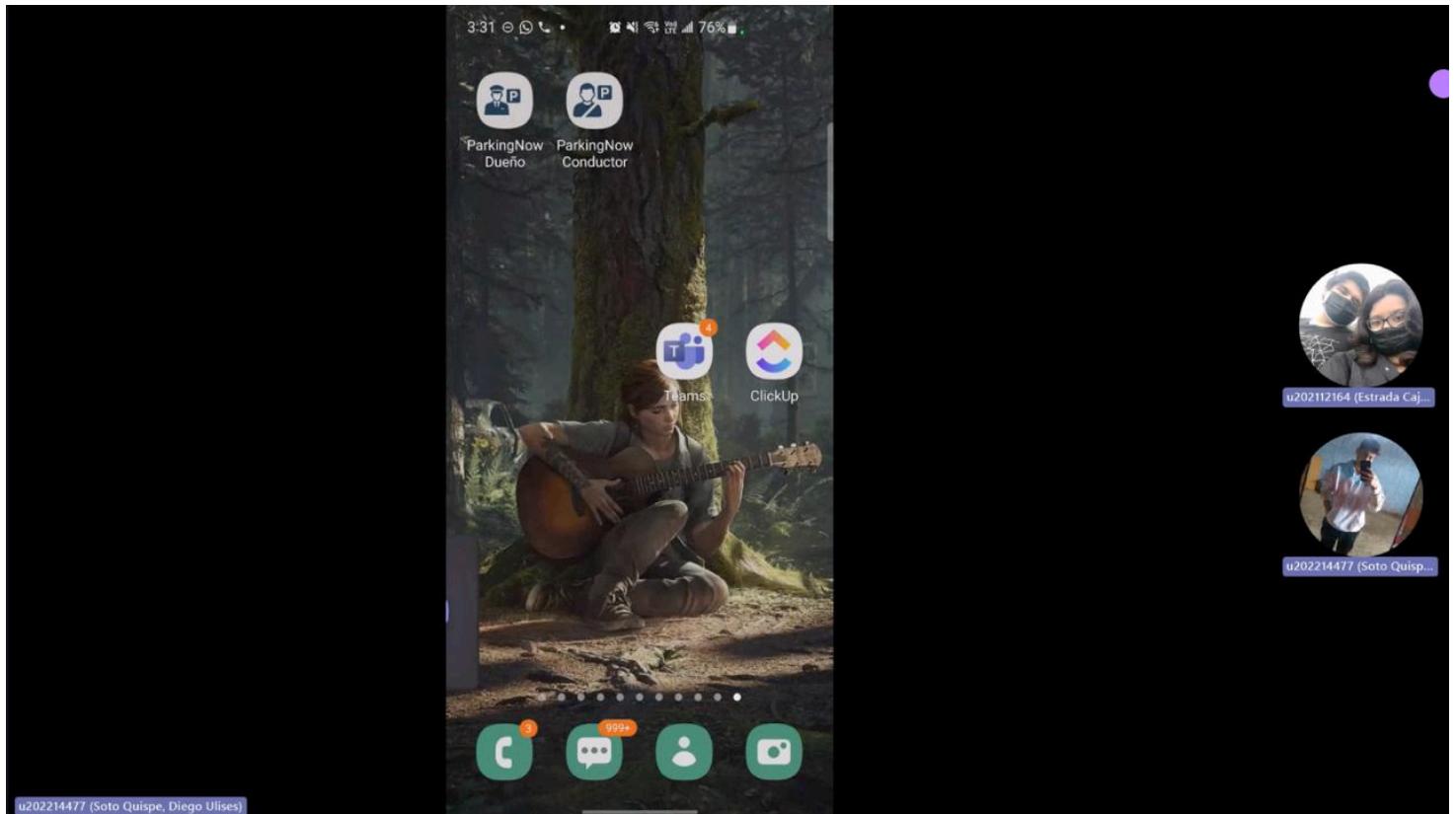
A screenshot of a web browser displaying the "PARKINGNOW" landing page. The page has a dark header with the title "PARKINGNOW" and a sub-header "g3-upc-cc238-346-parkingnow.github.io/Landing-ParkingNow/". To the right of the header, there are video call windows for "Juan Jesús Calisaya Sanchez" and "Rodrigo Alcantara". The main content features a section titled "Cities where you can find PARKINGNOW" with three cards: "Lima" (vibrant capital of Peru), "Arequipa" (white sillar buildings and Misti volcano), and "Cusco" (ancient capital of the Inca Empire). Below this is a yellow banner with the text "Service Benefits" and three corresponding cards: "Frequent Urban Drivers", "Parking Owners", and "Business Integration".

Destacó el diseño moderno y bien distribuido de la página, y que las imágenes y ciudades mostradas transmiten confianza. Valoró la separación clara entre conductores y dueños, y señaló que se animaría a usar la app por su utilidad frente al tráfico y la dificultad de encontrar estacionamiento en Lima.

👤 Entrevistada – Segmento Dueño de Estacionamiento

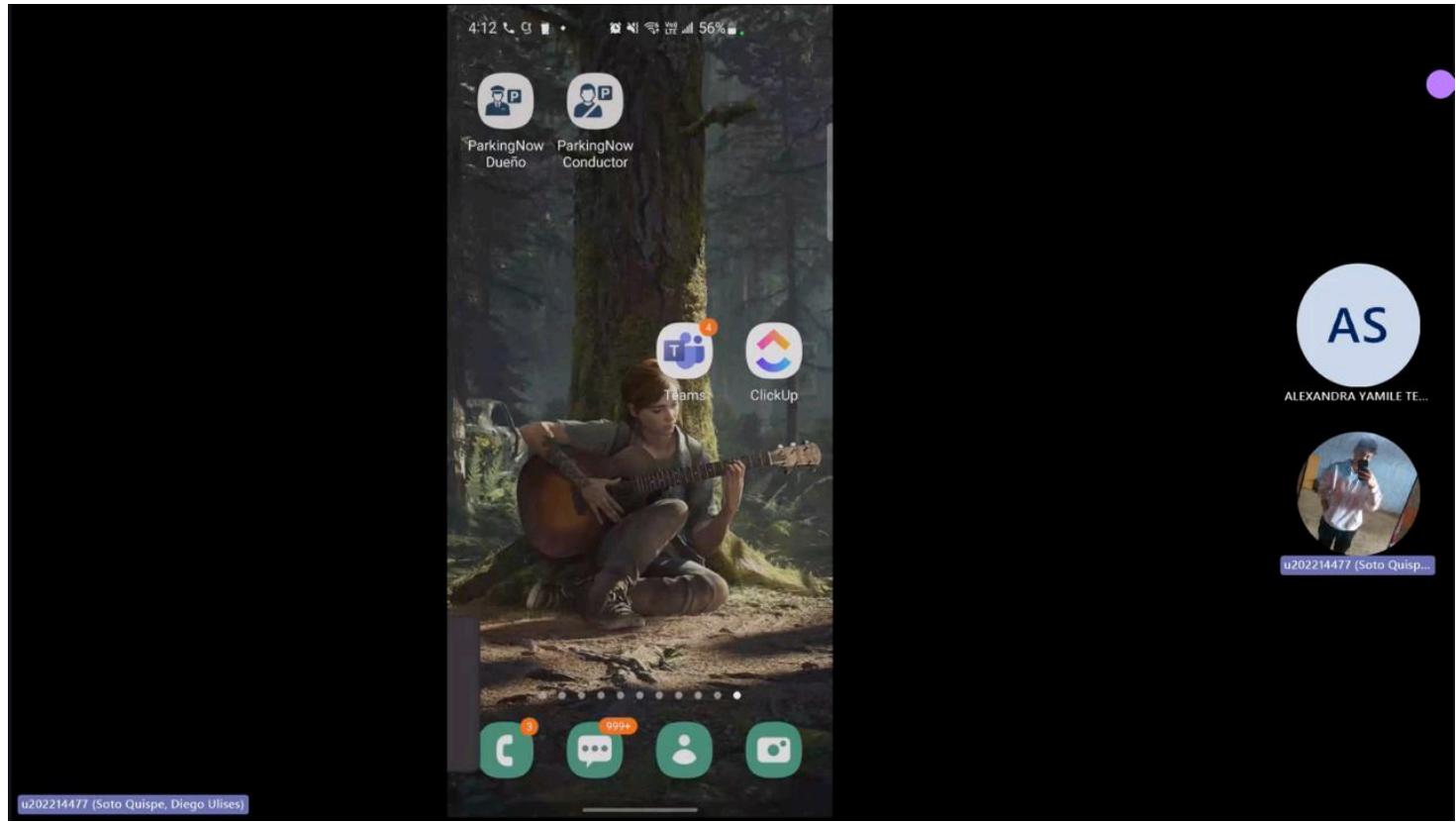
Entrevistado: Abraham Estrada – Segmento Dueño de Estacionamiento

Abraham tuvo una experiencia positiva pese a ser nuevo en la app. Apreció el resumen semanal y cómo le facilita la gestión diaria. Mencionó que algunos indicadores podrían explicarse mejor y propuso integrar una vista tipo calendario y conectarse con sensores o cámaras en el futuro. También valoró la reducción de tareas manuales gracias a la app.



Entrevistado: Alexandra Teves – Segmento Dueño de estacionamiento

Alexandra valoró mucho el resumen de ingresos y la visualización general del estado del parqueo. Le pareció útil la función de alertas, y reconoció que la app le ahorra tiempo en la gestión diaria. Sugirió mejorar la explicación visual de los porcentajes de ocupación y le gustaría exportar reportes y enviar promociones a conductores frecuentes.



Link de las entrevistas: https://upcedupe-my.sharepoint.com/:v/g/personal/u202214477_upc_edu_pe/EXEtNF96czNDldQBMxxR1oBzamOgCmxSzM9kjuRcv-ixw?e=TO1CCZ&nav=eyJyZWZlcnJhbEluZm8iOnsicmVmZXJyYWxBeHAiOjTdHJIYW1XZWJBcHAiLCJyZWZlcnJhbFZpZXciOiJTaGFyZURpYWxvZy1MaW5rIiicmVmZXJyYWxBeHBQbGF0Zm9ybSI6IldlYlslslnJlZmVycmFsTW9kZSI6InZpZXcifX0%3D

6.3.3.1 Evaluaciones según heurísticas

📌 SITE O APP A EVALUAR:

ParkingNow – Aplicación móvil para conductores y dueños de estacionamiento.

📝 TAREAS A EVALUAR:

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

🚗 Segmento Conductor:

1. Registro de un nuevo conductor
2. Inicio de sesión
3. Recuperación y cambio de contraseña
4. Búsqueda de estacionamientos cercanos en el mapa
5. Reserva de un espacio de estacionamiento
6. Confirmación de reserva con generación de código QR
7. Pago mediante método en línea
8. Visualización del historial de reservas
9. Gestión de notificaciones
10. Acceso a soporte técnico desde el dashboard
11. Modificación de preferencias del sistema

🏢 Segmento Dueño de Estacionamiento:

1. Registro como nuevo usuario dueño
2. Inicio de sesión
3. Recuperación y cambio de contraseña

4. Registro de un estacionamiento nuevo
5. Ubicación del estacionamiento en el mapa
6. Configuración de horarios de atención y servicios ofrecidos
7. Establecimiento de tarifas por hora
8. Gestión y visualización de reservas recibidas
9. Módulo de seguridad (alertas, control de acceso)
10. Centro de notificaciones
11. Barra lateral para navegación entre módulos
12. Revisión de términos y condiciones legales

No están incluidas en esta versión de la evaluación las siguientes tareas:

- Mostrar un tutorial inicial para usuarios nuevos
- Compartir reservas activas entre usuarios
- Implementar respuestas automáticas para preguntas frecuentes del conductor
- Guardar o compartir el código QR generado después de la reserva
- Exportar reportes o ingresos en formato Excel para los dueños
- Mostrar leyendas o ayudas visuales para interpretar los porcentajes de ocupación

ESCALA DE SEVERIDAD:

Nivel	Descripción
1	Problema superficial: fácilmente superado por el usuario
2	Problema menor: ocurre ocasionalmente, requiere revisión
3	Problema mayor: impide el uso correcto de funciones clave
4	Problema muy grave: bloquea el uso o continuidad de la app

TABLA RESUMEN:

#	Problema detectado	Escala de Severidad	Heurística/Principio violado
4	No se puede guardar o compartir el código QR generado tras la reserva	2	Control y libertad del usuario
5	No existe una leyenda que explique el porcentaje de ocupación en el dashboard del dueño	1	Visibilidad del estado del sistema
6	No hay opción para exportar reportes o ingresos en formatos como Excel	3	Flexibilidad y eficiencia del uso

DESCRIPCIÓN DE PROBLEMAS:

PROBLEMA #4: No se puede guardar o compartir el código QR generado tras la reserva

Severidad: 2

Heurística violada: Control y libertad del usuario

Descripción:

Actualmente, al finalizar una reserva, se genera un código QR que se muestra en pantalla, pero no se ofrece la opción de guardarlo como imagen ni de compartirlo fácilmente con terceros (por ejemplo, vigilantes o encargados del estacionamiento). Esto puede generar molestias si el usuario necesita acceder al QR fuera de la app o si su conexión se pierde.

← Métodos de Pago Guardados

No hay métodos guardados.

Pagar con Nuevo Método

Visa

Yape

Card Number



Card Holder Name



Expiry Date (MM/YY)



CVV

Guardar

Pagar con este método

Seleccione un método para pagar.

Recomendación:

Añadir un botón visible para **guardar el QR en la galería del dispositivo y/o compartirlo directamente por apps externas** como WhatsApp, correo o Google Drive. Esto mejoraría la portabilidad del comprobante de reserva y aumentaría la confianza del usuario.

PROBLEMA #5: No se puede visualizar un historial detallado de reservas anteriores

Severidad: 2

Heurística violada: Recuperación y seguimiento de información

Descripción:

Actualmente, la app muestra las reservas activas, pero **no permite acceder a un historial completo de reservas pasadas**, con detalles como fecha, hora, monto pagado o parqueo utilizado. Esto dificulta el seguimiento del uso de la app, tanto para conductores como para dueños.

1:03

54%

Dashboard



¡Bienvenido de vuelta!

John Smith

Lima, Perú

P

12

Espacios



8

Reservas



75%

Ocupación

Resumen de Ingresos

Junio 2025

Hoy

S/ 150.00

+12% vs semana pasada

Esta semana

S/ 1,200.00

+8% vs semana pasada

[Ver reporte completo](#)



Dashboard



Estaciona...



Reservas



Notificacio...

Recomendación:

Agregar una sección de "**Historial de reservas**" que muestre un listado con información básica y filtros por fecha. Esto permitiría al usuario revisar sus movimientos previos, resolver dudas o incluso repetir reservas frecuentes con más facilidad.

PROBLEMA #6: No hay opción para exportar reportes o ingresos en formatos como Excel

Severidad: 3

Heurística violada: Flexibilidad y eficiencia del uso

Descripción:

Aunque el dashboard muestra ingresos diarios y semanales, **no existe una función para exportar esa información** en formatos útiles como Excel o PDF. Esto limita el análisis externo y la posibilidad de llevar un control financiero fuera de la app, especialmente para dueños que necesitan presentar reportes o cruzar datos.

1:05 Ct

55%



Reservas



4
Total



1
Confirmadas



1
Pendientes



Carlos Rodríg...

NUEVO

Confirmada

ID: RES001

Toyota Corolla

ABC-123

P A-15

09:00 - 11:00

S/ 10.00

Contactar

Gestionar



María González

Pendiente

ID: RES002

Honda Civic

XYZ-789

P B-08

10:30 - 12:30

S/ 10.00

Contactar

Gestionar



Luis Pérez

Completada

ID: RES003

Nissan Sentra

DEF-456

P C-22



13:00 - 15:00

S/ 10.00

Contactar

Gestionar



Recomendación:

Incorporar una funcionalidad que permita **exportar reportes financieros o de actividad en formatos estándar** (CSV, Excel, PDF). También se podría ofrecer un filtro por fechas o por parqueo, en caso de tener múltiples espacios.

6.4. Video About the Product

Este video explica las funcionalidades principales y ventajas de la aplicación ParkingNow, mostrando cómo facilita la gestión de estacionamientos para los usuarios.



Link de Youtube: https://youtu.be/XsC8lhDq_5g

Link de OneDrive: https://drive.google.com/drive/folders/1hj5VXcLA54MAAI0q2zQwFMNUgJC4RYWE?usp=drive_link

6.4.1. Video About the Team

Este video presenta al equipo de desarrollo detrás de ParkingNow, destacando el rol y las contribuciones de cada miembro al éxito del proyecto.



Link de Youtube: <https://youtu.be/9jXLs6pOUi0>

Link de OneDrive: https://drive.google.com/drive/folders/1hj5VXcLA54MAAl0q2zQwFMNUgJC4RYWE?usp=drive_link

Conclusiones y recomendaciones

El proyecto PARKINGNOW ha logrado avances significativos en el desarrollo de una solución tecnológica integral para resolver los problemas de estacionamiento en entornos urbanos. Al integrar cámaras de monitoreo visual en tiempo real con una plataforma intuitiva, PARKINGNOW optimiza la ocupación de los espacios disponibles y facilita el acceso a estacionamiento para los conductores urbanos. Esta solución ha demostrado ser eficaz en la reducción de tiempos de búsqueda y en la mejora de la rentabilidad para los administradores de espacios de estacionamiento, mostrando un balance adecuado entre tecnología y experiencia del usuario.

Durante su desarrollo, se priorizó un enfoque de automatización e integración fluida con sistemas existentes de gestión de estacionamientos. Las entrevistas de validación con usuarios y las pruebas de campo confirmaron que la solución no solo cumple con las expectativas iniciales, sino que también supera las necesidades de los principales actores involucrados. La aplicación no solo resuelve la dificultad de encontrar estacionamiento, sino que también contribuye a la reducción del tráfico urbano y a una gestión más eficiente de los recursos, alineándose con los objetivos de sostenibilidad a largo plazo.

PARKINGNOW tiene el potencial de consolidarse como un referente en la gestión de estacionamientos urbanos, especialmente en ciudades donde el tráfico y la falta de espacios representan un desafío diario. La plataforma ha demostrado capacidad de escalabilidad y adaptación tanto en diferentes contextos geográficos como tecnológicos.

Recomendaciones:

- **Expansión de funcionalidades:**

Se recomienda continuar con el desarrollo de nuevas funcionalidades que agreguen valor a los usuarios y administradores, como la integración de sistemas de pago automatizados, alertas de disponibilidad en tiempo real y mejoras en la personalización mediante análisis predictivo. La incorporación de inteligencia artificial para pronosticar ocupación sería una ventaja competitiva.

- **Expansión geográfica y sectorial:**

Tras el éxito inicial, PARKINGNOW debería expandirse a nuevas ciudades con alta demanda de estacionamiento y explorar oportunidades en sectores como aeropuertos, centros comerciales y eventos masivos.

- **Estrategia de marketing y educación:**

Es esencial diseñar una campaña de marketing digital enfocada en conductores y administradores de estacionamientos. El uso de testimonios reales y estudios de caso sobre mejoras en tiempos de búsqueda puede aumentar significativamente la adopción. Además, se debe educar al público sobre los beneficios en sostenibilidad urbana.

- **Monitoreo y análisis continuo:**

Se debe implementar el análisis constante de datos de uso y ocupación de espacios para anticipar necesidades del mercado y corregir posibles deficiencias, asegurando una mejora continua de la experiencia del usuario.

- **Sostenibilidad y responsabilidad social:**

PARKINGNOW debe fortalecer su compromiso con la movilidad sostenible, promoviendo iniciativas que reduzcan la huella de carbono y colaborando con proyectos de ciudades inteligentes, para posicionarse no solo como una plataforma tecnológica, sino también como un actor clave en la sostenibilidad urbana.

Bibliografía - 4 papers

Ali, G., Ali, T., Irfan, M., Draz, U., Sohail, M., Glowacz, A., ... Martis, C. (2020). IoT-based smart parking system using deep long short-memory network. *Electronics*, 9(10), 1696. <https://doi.org/10.3390/electronics9101696>

Saharan, S., Kumar, N., & Bawa, S. (2020). An efficient smart parking pricing system for smart city environment: A machine-learning based approach. *Future Generation Computer Systems*, 106, 622–640. <https://doi.org/10.1016/j.future.2020.01.031>

Shen, L., Wu, X., Li, R., Chen, J., & Zhao, Y. (2020). Logistics sustainability practices: An IoT-enabled smart indoor parking system for industrial hazardous-chemical vehicles. *International Journal of Production Research*, 58(11), 3350–3367. <https://doi.org/10.1080/00207543.2020.1720928>

Simsek, M., Polat, K., Damas, M., & Celebi, M. (2020). Smart parking system with dynamic pricing, edge-cloud computing and LoRa. *Sensors*, 20(17), 4669. <https://doi.org/10.3390/s20174669>

Bibliografía

1. **Conducimos contigo.** (s.f.). [Elparking.com](https://elparking.com/). Recuperado el 1 de septiembre de 2024, de <https://elparking.com/>
2. **ICCSI.** (2023, 12 de marzo). *Estacionamiento inteligente con sensores: Revolución en estacionamientos*. ICCSI. Recuperado el 6 de septiembre de 2024, de <https://iccsi.com.ar/estacionamiento-inteligente-con-sensores>
3. **InnovaTech.** (2023, 15 de junio). *Smart Parking IoT: Descubre cómo funciona*. InnovaTech. Recuperado el 6 de septiembre de 2024, de <https://www.innova-tech.mx/smart-parking-iot>
4. **Intertraffic.** (2023, 28 de junio). *Revolucionando las experiencias de estacionamiento: Las últimas soluciones inteligentes para un estacionamiento eficiente*. Intertraffic. Recuperado el 6 de septiembre de 2024, de <https://www.intertraffic.com/news/espanol/revolucionando-las-experiencias-de-estacionamiento>
5. **People, S.** (s.f.). App Aparcados. [Apk2gestion.com](https://apk2gestion.com/es/servicios/conoce-nuestra-app). Recuperado el 2 de septiembre de 2024, de <https://apk2gestion.com/es/servicios/conoce-nuestra-app>
6. **Propuesta País.** (2023, 22 de julio). *La necesidad de una política de estacionamiento vehicular en Lima Metropolitana*. Propuesta País. Recuperado el 6 de septiembre de 2024, de <https://propuestapais.pe/noticia/la-necesidad-de-una-politica-de-estacionamiento-vehicular-en-lima-metropolitana/>
7. **Reserva de parking - Parkimeter.** (s.f.). [Parkimeter.com](https://parkimeter.com/). Recuperado el 3 de septiembre de 2024, de https://parkimeter.com/es?srsltid=AfmBOoqJCDIJN_hdtP3dMuF6fXWF8B1PytqQ2hWnVsgR7rEAH4a00y3P
8. **Sernauto.** (s.f.). *Tipos de sensores de los vehículos: Qué función cumplen*. Sernauto. Recuperado el 6 de septiembre de 2024, de <https://www.sernauto.es/blog/tipos-de-sensores-de-losvehiculos-que-funcion-cumplen/>

Anexos

Sección	
⌚ Lean UX Canvas	https://www.canva.com/design/DAGOyaW6z7E/ub8lVBXsKH3Em5YVQb223g/edit
⌚ To-be Scenario Mapping - Dueños de Estacionamiento	https://miro.com/app/board/uXjVKmNkzqc=/?share_link_id=307784420338

Sección	
To-be Scenario Mapping - Conductores Urbanos Frecuentes	https://miro.com/app/board/uXjVKmOWBhk=/?share_link_id=425124811159
Entrevistas en general - Segmento 1 y 2	https://upcedupe-my.sharepoint.com/:g/personal/u202214477_upc_edu_pe/EQfuOaavJTCi5jKvTLzMU0Bixb6cO1iHY4X7yZitnQhl
Event Storming	https://miro.com/welcomeonboard/VTBYK0pNaFZXMHdaZkJJVkrQTJlbkVVK1hua3l3SGxZS0tJWDlrdkdSKzBCMStzODVJdnRCYUV
Bounded Context Canvas	https://miro.com/welcomeonboard/cGZuTUVpS2dKQmNEMytdnVVNmRwMHJjSHNZWnFXYW0wanRFSzJwbXJPZCtJZXVudE9mNI
Video de Sprint Review (6.2.1.4)	https://upcedupe-my.sharepoint.com/:g/personal/u202214477_upc_edu_pe/EYOajxvFR85Nu8jiZWJN-LMBsXskewjVOlyAhocoIZQde=EYdvDO&nav=eyJyZWZlcnJhbEluZm8iOnsicmVmZXJyYWxBcHAIoJTDHJIYW1XZWJBcHAiLCJyZWZlcnJhbFZpZXciOjTaGFyZL
Figma – UI del Proyecto	https://www.figma.com/design/qb02D1snVoDtoLfT0qdkRV/Parking-now?node-id=0-1&t=vfQZ7EcT7cMorKUF-1
Tablero Trello – Gestión de Tareas	https://trello.com/invite/b/682808c86697c9a394dc8b12/ATTlida1c9216f53ce3913ed85ccb9445734eA3B9B093/parkingnow
Repositorio GitHub – Kotlin (Android Nativo)	https://github.com/G3-UPC-CC238-346-ParkingNow/Kotlin-Mobile
Repositorio GitHub – Backend	https://github.com/G3-UPC-CC238-346-ParkingNow/Backend
Repositorio GitHub – Flutter (Cross-platform)	https://github.com/G3-UPC-CC238-346-ParkingNow/Flutter-Mobile
Repositorio GitHub – Landing Page Web	https://github.com/G3-UPC-CC238-346-ParkingNow/New-Landing-Page
Repositorio GitHub – Reporte Final	https://github.com/G3-UPC-CC238-346-ParkingNow/Report