



# ParkGo

Julieth Timaure

Técnico en Programación y Análisis de Sistemas

Proyecto de Especialidad: Repositorio GitHub

# La Problemática de la Movilidad Urbana



En la actualidad, las ciudades enfrentan un crecimiento constante del parque automotor, generando una **alta demanda de estacionamientos** que supera con creces la oferta disponible y accesible.

El proceso tradicional para encontrar un estacionamiento es **ineficiente y frustrante**, basándose en métodos anticuados como carteles físicos o publicaciones dispersas en redes sociales. Esta situación genera **pérdida de tiempo, estrés** y, a menudo, congestión vehicular.

Se hace imperativa una **solución tecnológica centralizada** que conecte de manera eficiente la oferta y la demanda.

# Objetivo General: Optimizando el Estacionamiento Urbano



## Optimizar Arriendos

Mejorar el proceso de arrendamiento de estacionamientos fijos y a largo plazo en Concepción.



## Eficiencia en la Búsqueda

Facilitar la búsqueda para conductores, reduciendo tiempo y esfuerzo.



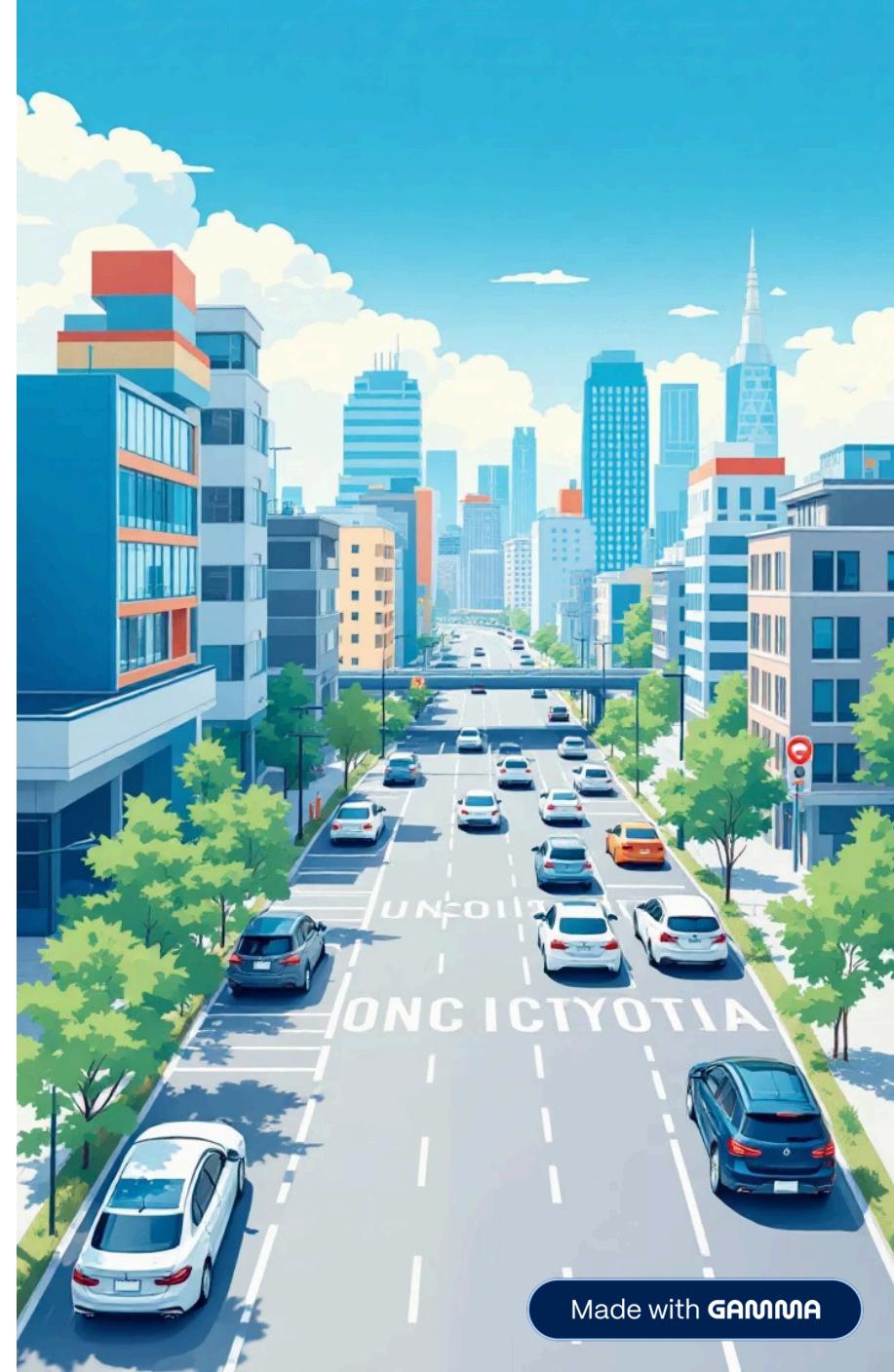
## Oportunidad Económica

Ampliar las oportunidades para propietarios de espacios sin usar.



## Seguridad y Confianza

Garantizar transacciones seguras para ambas partes a través de un sistema web centralizado.



# Objetivos Específicos

## → Implementar Arquitectura Backend RESTful

Desarrollar un sistema que gestione usuarios, vehículos, publicaciones y arriendos, asegurando respuestas eficientes y la integridad del 100% de las transacciones simuladas.



## → Desarrollar Módulo de Validación de Identidad

Verificar automáticamente el formato nacional de identificación (RUT) y garantizar la unicidad de las patentes vehiculares registradas en el sistema, elevando la seguridad y confiabilidad.



## → Integrar Servicios de Geolocalización Interactiva

Permitir la visualización de la oferta de estacionamientos en un mapa digital, facilitando la ubicación exacta de los espacios disponibles para los usuarios.



## → Desplegar Prototipo Funcional en Nube

Alojar la aplicación, incluyendo base de datos y archivos multimedia, en un entorno de nube para asegurar su disponibilidad operativa antes de la fecha de entrega final del proyecto.



# Innovación y Beneficiarios

## Innovación Funcional

### Filtros Estructurados

Ofrece filtros avanzados y campos de información obligatorios que aseguran que el 100% de las publicaciones contengan datos completos para una toma de decisiones informada.

#### Disponibilidad Horaria

##### Tipo de Acceso

Acceso 24/7 (Libre)

#### Características

##### Tipo de Cobertura



Aire Libre



Techado



Subterráneo

#### Seguridad (Selecciona múltiples)



Portón Eléct.



Cámaras



Conserje



Luz

#### Dimensiones (Opcional)

##### Largo (m)

Ej: 5.0

##### Ancho (m)

Ej: 2.5

## Innovación Social

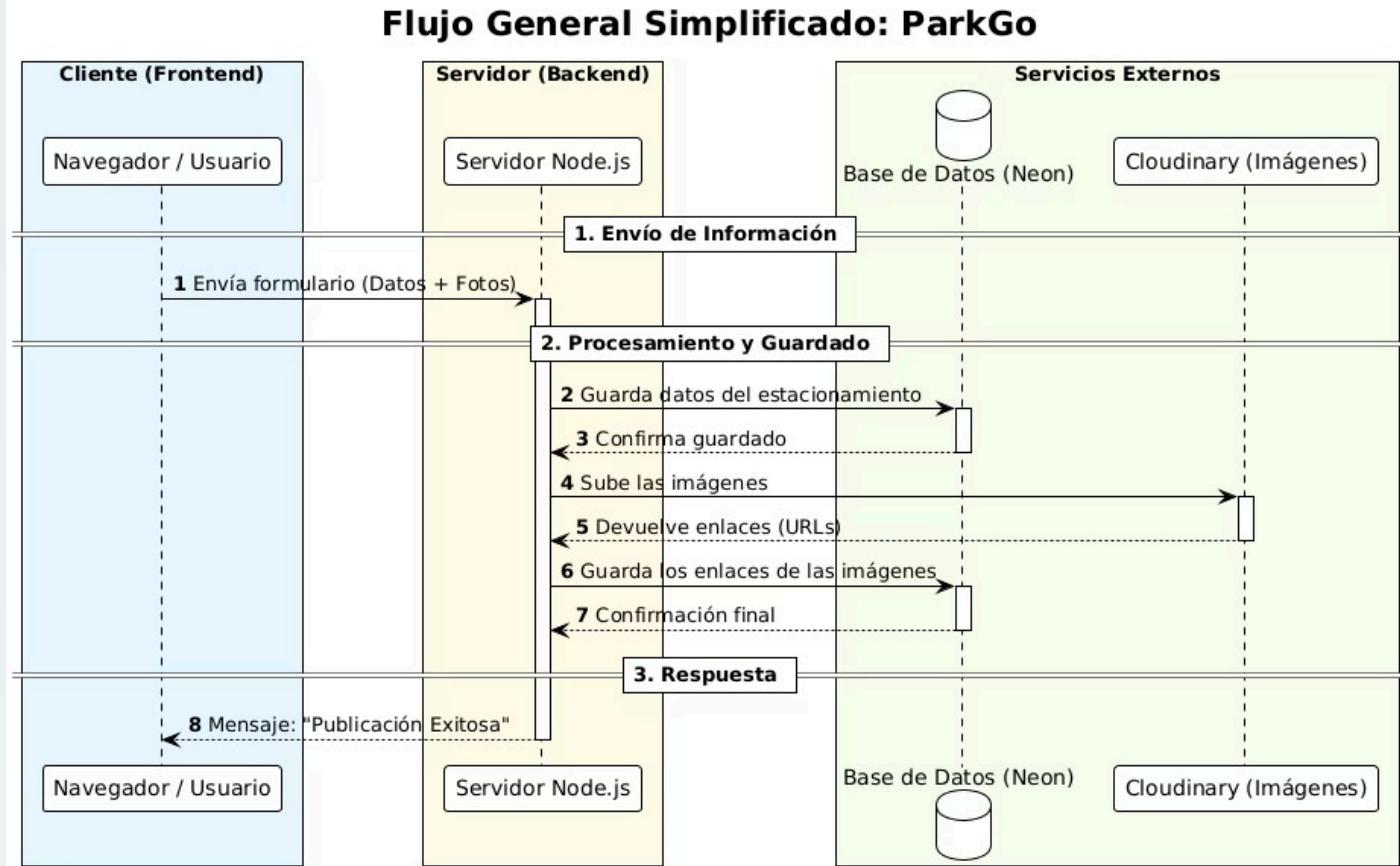
### Monetización de Activos

Permite a pequeños propietarios monetizar activos en desuso, democratizando el acceso a herramientas tecnológicas que antes solo estaban al alcance de grandes empresas. Esto fomenta una economía colaborativa y local.

The screenshot shows a user profile for "Andrea" (Dueño) managing a parking space at "Estacionamiento Laguna Redonda". The listing details include:

- Ubicación Completa:** Avenida 21 de Mayo #2313 Concepción, Biobío
- Tu Vehículo Autorizado:** Kia Morning
- Horario:** 24/7 Libre
- Nº Estacionamiento:** 5-F
- Ver Ubicación Exacta:** Button to view map
- Fin contrato:** 16-03-2026
- Suscripción Activa:** Green button indicating active subscription
- Cancelar Suscripción:** Red button to cancel subscription

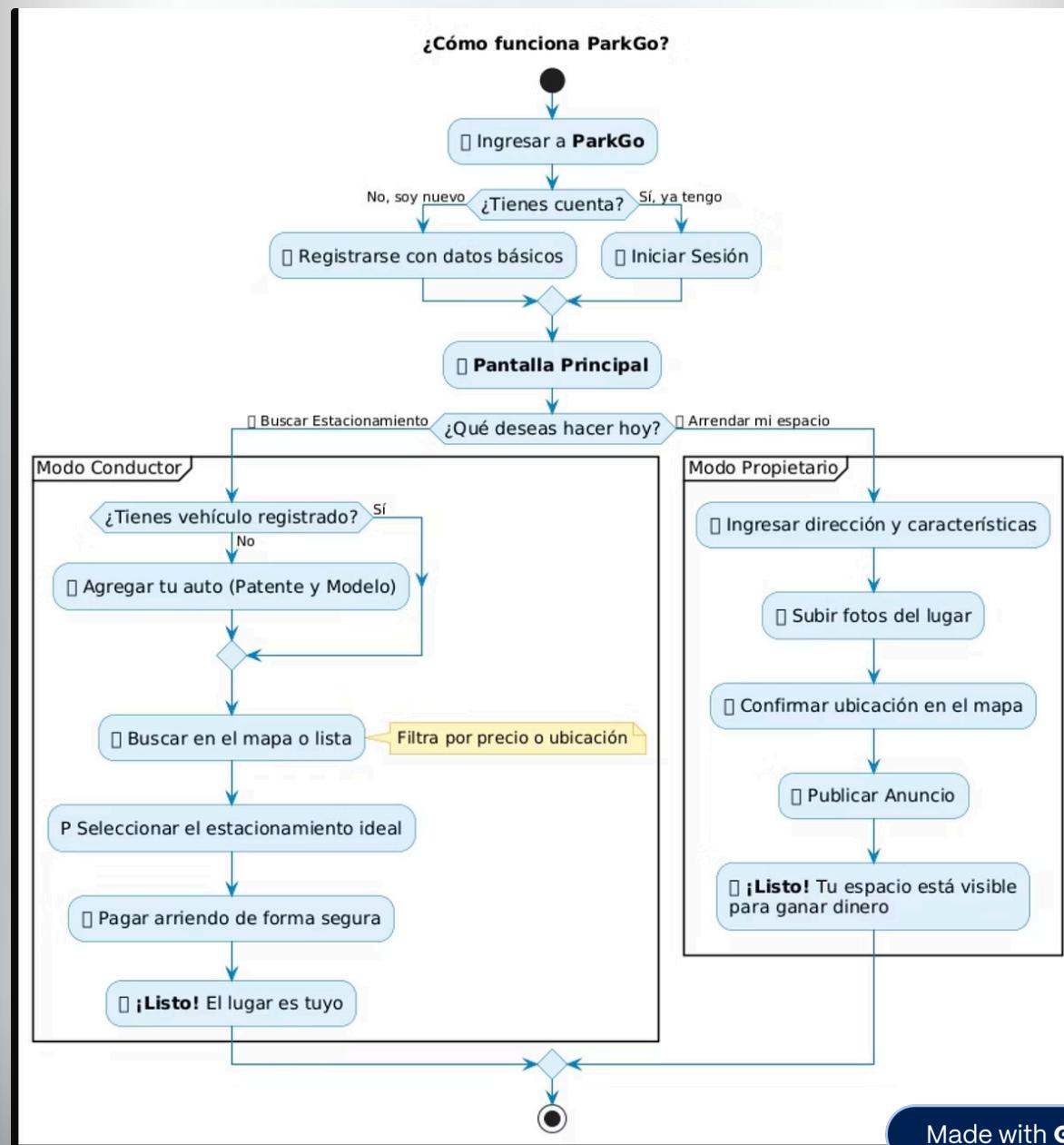
# Funcionamiento del Sistema (Nivel Interno)



Este diagrama ilustra la interacción de los componentes clave de ParkGo, desde la interfaz de usuario hasta la base de datos, mostrando cómo se gestionan las peticiones y se procesan los datos de forma interna para ofrecer una experiencia fluida y segura.

# Diagrama Simplificado de Flujo del Sistema

El diagrama muestra el ciclo de vida de una transacción en ParkGo, desde que el usuario inicia una acción hasta que se almacena y recupera la información, destacando la interacción entre los distintos módulos y servicios. Los flujos están diseñados para minimizar la latencia y maximizar la eficiencia operativa.



# Gestión de Datos Robusta y Segura



## Base de Datos PostgreSQL en Neon.tech

Uso de PostgreSQL en la plataforma Neon.tech para garantizar escalabilidad, fiabilidad y un rendimiento óptimo en el manejo de grandes volúmenes de datos.



## Manejo de Datos Normalizado (3FN)

La base de datos está diseñada siguiendo la Tercera Forma Normal (3FN) para asegurar la integridad referencial entre usuarios, vehículos, estacionamientos y contratos, evitando redundancias y anomalías.



## Transacciones con BEGIN, COMMIT, ROLLBACK

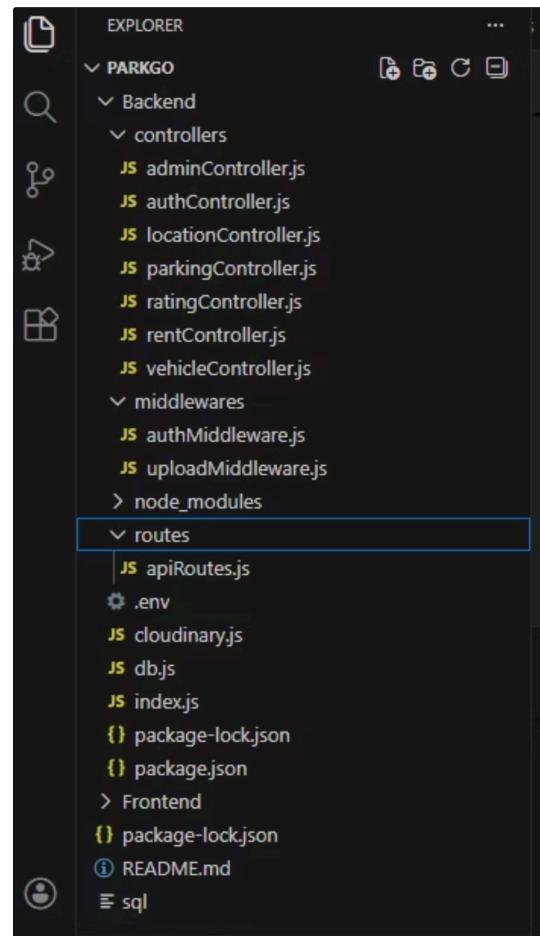
Implementación de bloques transaccionales en el código para garantizar la atomicidad de las operaciones. Si una parte de la transacción falla (ej. subida de imagen o inserción de datos), todos los cambios se revierten, previniendo registros corruptos.



## Almacenamiento de Imágenes en Cloudinary

Para optimizar el rendimiento del servidor y la base de datos, las imágenes se capturan, convierten a Base64 y se suben a Cloudinary. Solo la URL segura de la imagen se almacena en PostgreSQL, reduciendo la carga y mejorando los tiempos de respuesta.

# Arquitectura Backend



## Node.js y Express.js

El backend se construye con Node.js y el framework Express.js. En `index.js`, se configuran middlewares esenciales como `cors` para permitir peticiones desde el frontend, y `express.json` para el parseo de cuerpos de petición. La carpeta Frontend se sirve como archivos estáticos, facilitando la integración.

## Seguridad con JWT (JSON Web Tokens)

Se implementa un robusto sistema de autenticación basado en JWT. Al iniciar sesión, se genera un token que el frontend almacena localmente y envía en cada petición subsecuente. Esto permite validar la identidad y los roles del usuario, garantizando un acceso seguro y autorizado a los recursos del sistema.

# Desarrollo Frontend

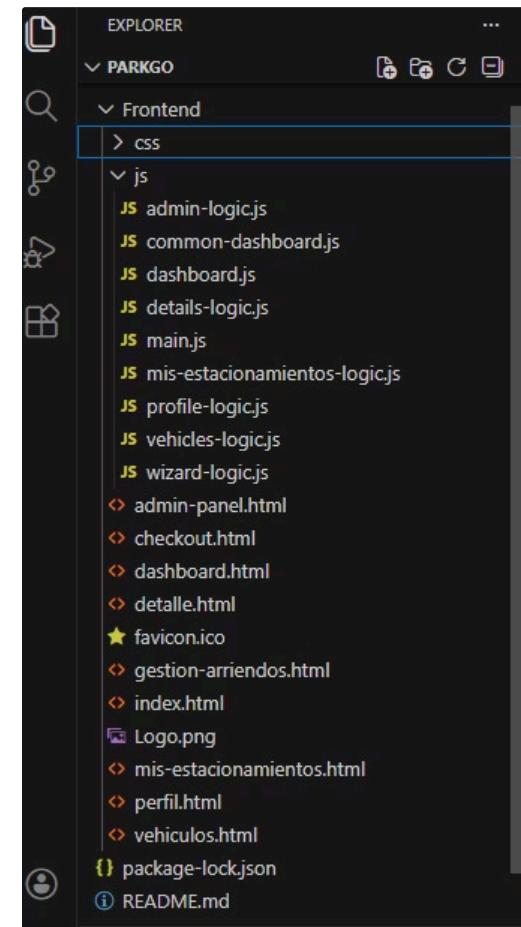
El frontend de ParkGo se desarrolla utilizando las tecnologías fundamentales de la web: **HTML5, CSS3 y JavaScript Vanilla**, lo que garantiza una aplicación ligera, rápida y compatible con todos los navegadores modernos.

## Lógica de Negocio con JavaScript

Toda la lógica de la interfaz de usuario se maneja con JavaScript. Por ejemplo, en `main.js`, se implementa la validación en tiempo real del RUT Chileno (Módulo 11) antes de enviar cualquier dato al servidor, mejorando la experiencia del usuario y la integridad de los datos.

## Comunicación Asíncrona con la API Fetch

La interacción con el backend es completamente asíncrona, utilizando `async/await` con la API `fetch`. Esto asegura que la interfaz de usuario permanezca responsive mientras se realizan operaciones de red, proporcionando una experiencia de usuario fluida y sin interrupciones.



# ParkGo: Innovando en la Movilidad Urbana

ParkGo es una solución tecnológica diseñada para transformar la gestión del aparcamiento urbano. Más allá de las operaciones básicas, ofrece una visión estratégica a administradores y usuarios, optimizando la experiencia de movilidad.

## Alcance del Proyecto: Un MVP con Visión de Futuro

La versión actual de ParkGo representa un Producto Mínimo Viable (MVP) robusto y funcional. Hemos sentado las bases para una plataforma escalable, identificando áreas clave para su evolución hacia un entorno productivo masivo.

### Integración de Pasarela de Pago Real

Facilitar transacciones seguras y eficientes es crucial. Implementaremos pasarelas de pago avanzadas para una experiencia de usuario fluida y fiable.

### Desarrollo de Aplicación Móvil

Llevaremos la experiencia ParkGo directamente a la mano de los usuarios, con una aplicación intuitiva para búsqueda, reserva y gestión de aparcamientos.

### Verificación de Identidad Avanzada

Garantizaremos la seguridad y confianza en la plataforma mediante sistemas de verificación de identidad robustos y eficientes.

### Chat en Tiempo Real

Mejoraremos la comunicación entre usuarios y administradores con un sistema de chat integrado, para resolver dudas y coordinar eficazmente.

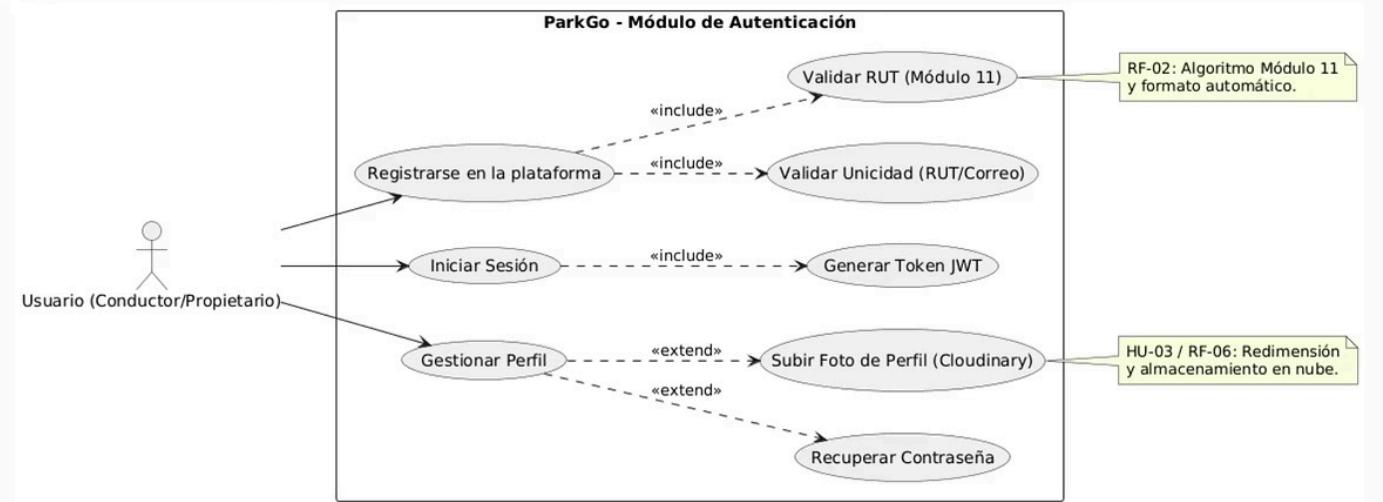
# Matriz de Trazabilidad: Progreso y Visión de Futuro

Esta matriz detalla el alcance logrado por ParkGo hasta la fecha, destacando las funcionalidades implementadas y la correspondencia con los requisitos iniciales. Sirve como una guía clara de nuestro progreso y las capacidades actuales de la plataforma.

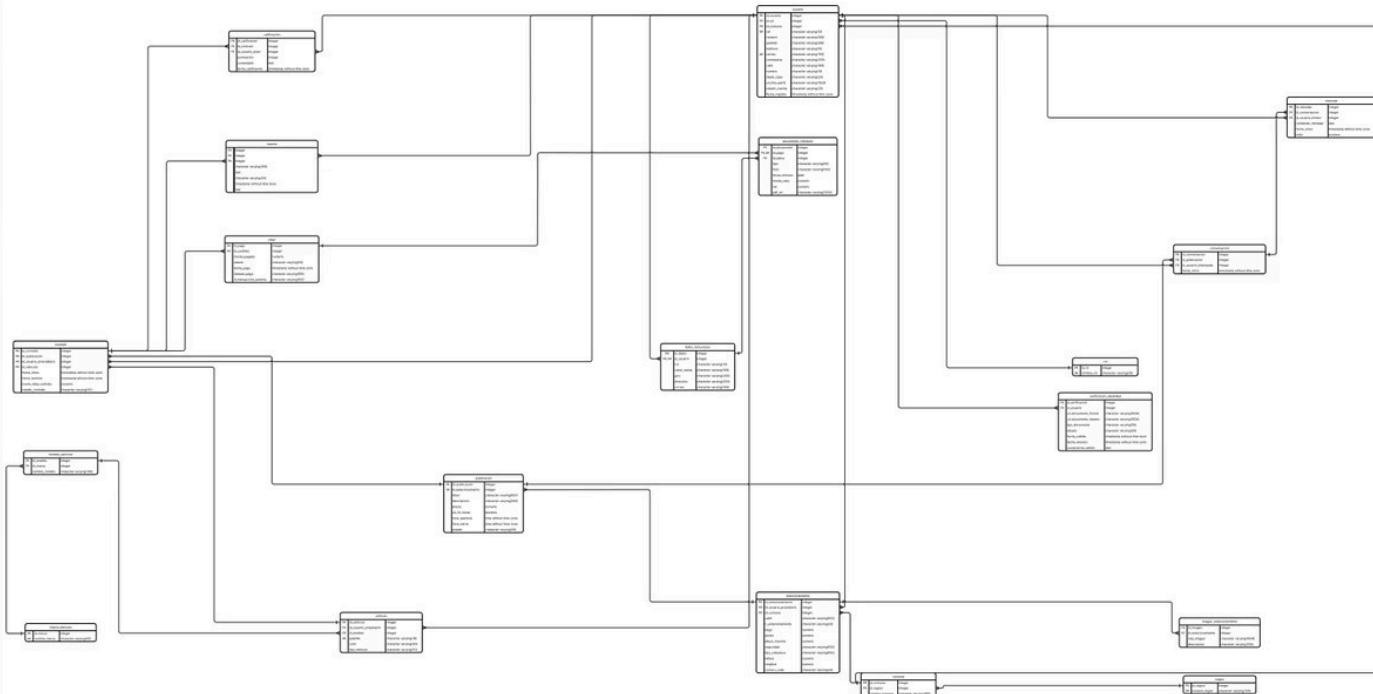
ID RF	DESCRIPCIÓN DEL REQUERIMIENTO	MÓDULO DEL SISTEMA	CASO DE USO (CU)
RF-01	Registro de usuarios con validación de RUT y datos personales.	Gestión de Usuarios (Auth)	CU-01: Registrar Nuevo Usuario
RF-03	Control de unicidad de cuentas (RUT y Correo no duplicados).	Gestión de Usuarios (Auth)	CU-01: Registrar Nuevo Usuario
RF-04	Inicio de sesión seguro con encriptación y Token JWT.	Gestión de Usuarios (Auth)	CU-02: Iniciar Sesión
RF-05	Edición de perfil de usuario y actualización de contraseña.	Gestión de Usuarios (Perfil)	CU-03: Gestionar Perfil
RF-07	Registro de vehículos con validación de patente única.	Gestión de Vehículos	CU-04: Registrar Vehículo
RF-09	Visualización y eliminación de vehículos propios.	Gestión de Vehículos	CU-05: Gestionar Flota
RF-10	Publicación de estacionamiento mediante asistente (Wizard).	Gestión de Parkings	CU-06: Publicar Estacionamiento
RF-13	Carga y gestión de galería de imágenes (Cloudinary).	Gestión de Parkings	CU-08: Subir Fotos
RF-14	Geolocalización exacta mediante mapa interactivo.	Gestión de Parkings	CU-07: Geolocalizar Espacio
RF-15	Búsqueda de estacionamientos con filtros (precio, ubicación).	Búsqueda y Arriendo	CU-09: Buscar Estacionamiento
RF-18	Simulación de costos y aplicación de descuentos por tiempo.	Búsqueda y Arriendo	CU-10: Arrendar Espacio
RF-20	Generación y aceptación de contrato digital.	Búsqueda y Arriendo	CU-10: Arrendar Espacio
RF-21	Transacción de arriendo y bloqueo de disponibilidad.	Búsqueda y Arriendo	CU-10: Arrendar Espacio
RF-23	Moderación de usuarios (Banear/Activar) por Administrador.	Administración	CU-11: Administrar Usuarios
RF-24	Calificación del servicio (Estrellas y comentarios).	Feedback	CU-12: Calificar Servicio

# Caso de Uso: La Experiencia ParkGo

El diagrama de caso de uso ilustra la interacción clave entre los usuarios y el sistema ParkGo. Visualiza cómo los usuarios pueden registrarse, iniciar sesión o gestionar perfil.

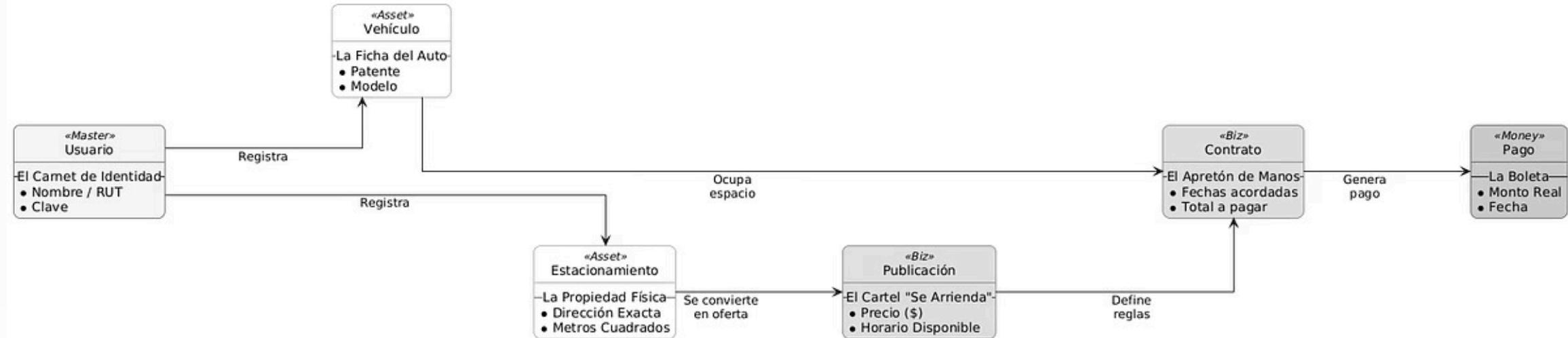


# Modelo Entidad- Relación: La Estructura de Datos de ParkGo



Este diagrama ER representa la organización fundamental de la base de datos de ParkGo, mostrando las entidades (como Usuarios, Plazas, Arriendos) y las relaciones entre ellas. Es el esqueleto lógico que soporta todas las funcionalidades de la plataforma.

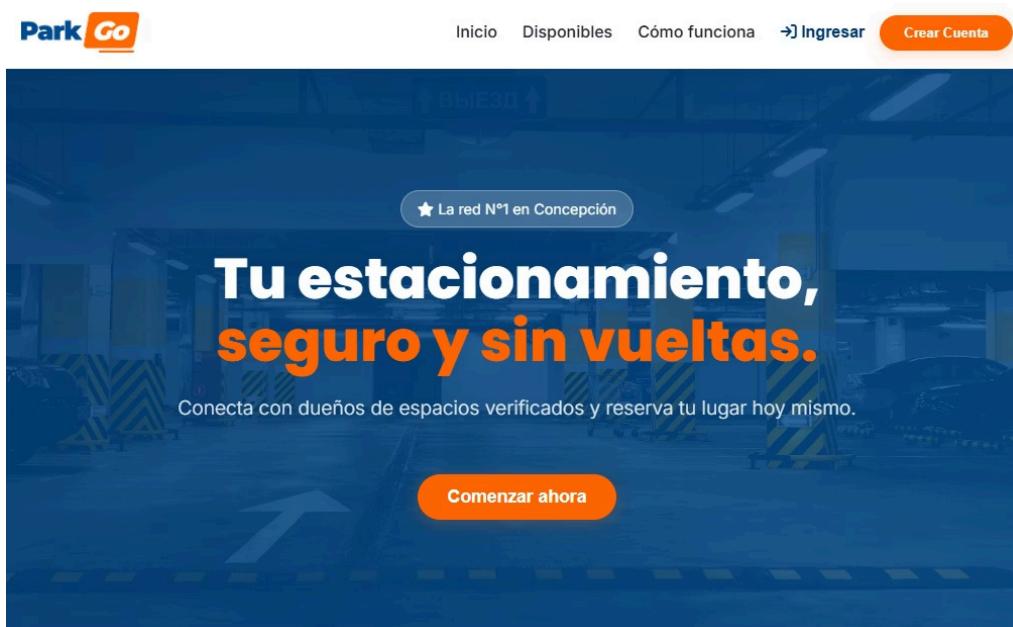
### Modelo Simplificado: Park Go



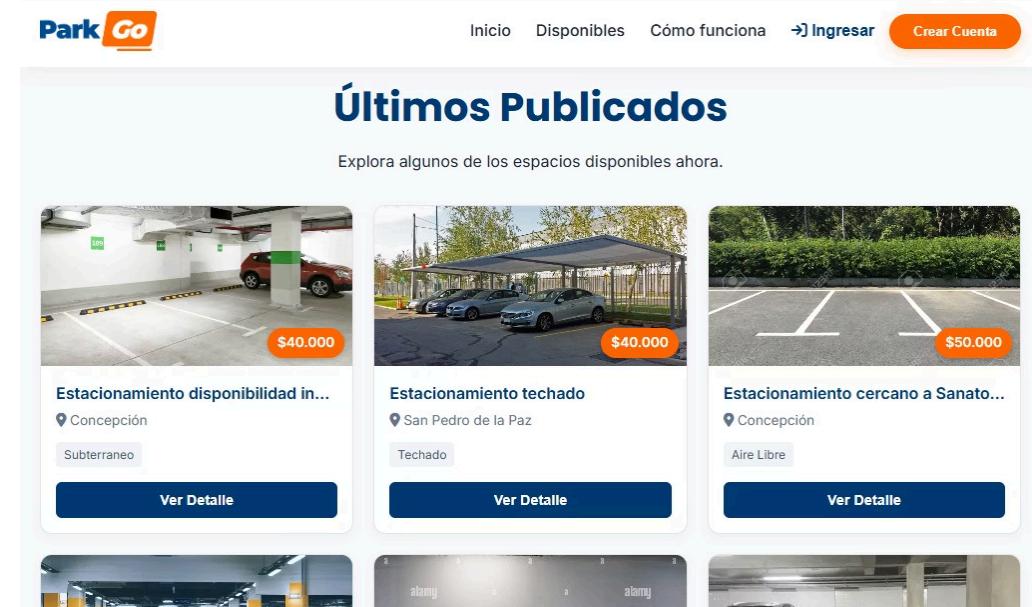
## Flujo de Operaciones: De la Consulta a la Gestión

Este diagrama ilustra el flujo y las tablas principales del sistema, así como su funcionamiento.

# Pantallas Principales: Interfaz Intuitiva de ParkGo



The homepage features a dark blue header with the ParkGo logo and navigation links: Inicio, Disponibles, Cómo funciona, Ingresar, and Crear Cuenta. Below the header is a banner with the text "★ La red N°1 en Concepción". The main headline reads "Tu estacionamiento, seguro y sin vueltas." with a subtitle "Conecta con dueños de espacios verificados y reserva tu lugar hoy mismo." A large orange button labeled "Comenzar ahora" is at the bottom.

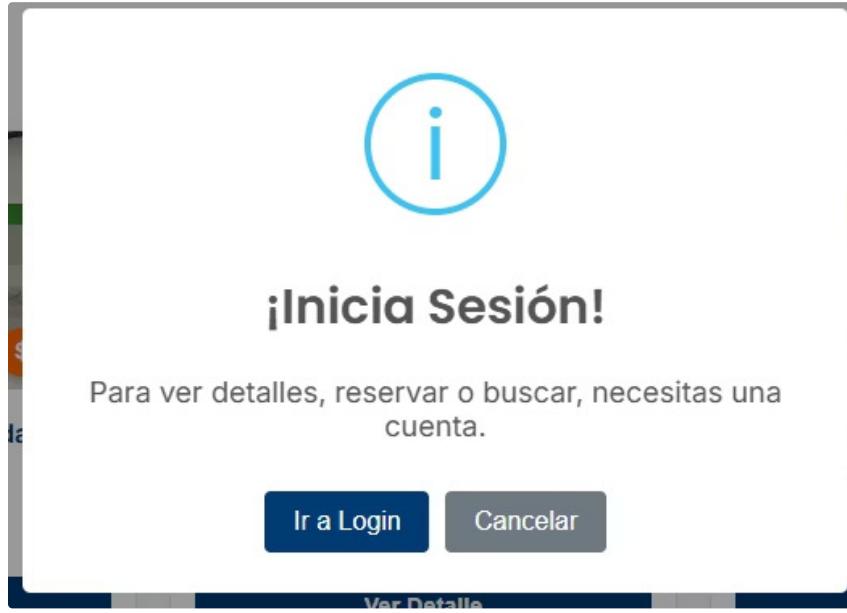


The section title is "Últimos Publicados" with a subtitle "Explora algunos de los espacios disponibles ahora." It displays three parking options:

- Estacionamiento disponibilidad in...** \$40.000 | Subterráneo | Ver Detalle
- Estacionamiento techado** \$40.000 | Techado | Ver Detalle
- Estacionamiento cercano a Sanato...** \$50.000 | Aire Libre | Ver Detalle

Las pantallas principales de ParkGo han sido diseñadas pensando en la usabilidad, ofreciendo una experiencia clara y directa tanto para usuarios que buscan aparcamiento como para administradores que gestionan la plataforma.

# Inicio de Sesión: Acceso Seguro y Simplificado

A screenshot of a login form titled 'Bienvenido'. It asks for 'Ingresa tus credenciales'. There are fields for 'Correo Electrónico' (with placeholder 'ejemplo@correo.com') and 'Contraseña' (with placeholder '\*\*\*\*\*'). To the right of the password field is an eye icon for password visibility. A large blue 'Iniciar Sesión' button is centered below the fields. At the bottom, there is a link '¿Nuevo aquí? Regístrate'.

El proceso de inicio de sesión de ParkGo ha sido diseñado para ser rápido, seguro y fácil de usar, garantizando que los usuarios puedan acceder a sus cuentas sin fricciones, mientras se mantiene la integridad de sus datos. La seguridad es nuestra prioridad, combinada con una experiencia de usuario sin complicaciones.

# Formulario de Registro de Usuario: ¡Bienvenido a ParkGo!

El formulario de registro de ParkGo es sencillo e intuitivo, diseñado para facilitar la incorporación de nuevos usuarios a la plataforma. Con pocos pasos, los usuarios pueden crear una cuenta y empezar a disfrutar de los beneficios de una gestión de aparcamiento inteligente.

## Crear Cuenta

Únete a Park Go

RUT  Teléfono

Nombre  Apellido

**UBICACIÓN**

Región  Comuna

Calle / Avenida  N°  Depto (Op)

Correo Electrónico  Contraseña  

**Registrarse**

¿Ya tienes cuenta? [Inicia Sesión](#)

# Panel de Usuario Común: Tu Centro de Control de Aparcamiento

El panel de usuario es el corazón de la experiencia ParkGo para cada individuo. Desde aquí, los usuarios pueden ver sus reservas activas, historial de aparcamiento, gestionar su perfil y acceder a todas las funcionalidades personalizadas de la plataforma. Una interfaz clara y organizada para un control total.

The screenshot displays the ParkGo user interface. On the left, a sidebar includes the ParkGo logo, a search bar labeled 'Buscar Parking', and links for 'Mi Perfil', 'Mis Vehículos', 'Mis Publicaciones', and 'Gestión Arriendos'. At the bottom of the sidebar is a red 'Cerrar Sesión' button. The main area shows a search interface with fields for location ('¿DÓNDE?'), price ('\$ PRECIO MENSUAL'), type ('TIPO'), and sorting ('ORDENAR'). It also features a 'Publicar Espacio' button. Below this, a message indicates '8 resultado(s) encontrado(s)'. Two parking options are listed: 'Estacionamiento disponibilidad inmediata' at Avenida Chacabuco, Concepción, and 'Estacionamiento techado' at Avenida Pedro Aguirre Cerda, San Pedro de la Paz. Both listings include details like 'Porton Electrico, Camaras, Conserje, Iluminacion', 'Subterraneo', and 'Techado', along with a '\$40.000 / mes' price tag and a 'Ver Detalles' button.

# Panel de Administración: Visión General

Explora la interfaz de administración diseñada para una gestión eficiente y centralizada de todos los recursos del sistema de movilidad urbana.

Desde este panel, los administradores pueden supervisar vehículos, gestionar usuarios y ver publicaciones con facilidad, garantizando un flujo de trabajo optimizado y una experiencia de usuario superior.

The screenshot displays the Park Go Admin Dashboard. At the top right, there is a user profile for 'A Julieth'. The dashboard features several key metrics: 4 Usuarios Registrados, 12 Total Publicaciones (with 8 Activas and 1 Arrendadas), and a Total Generado of \$90.000. Below these, a 'Reporte Gerencial' button is visible. The main area is titled 'Administración' and contains a 'Gestión de Usuarios' table. This table lists two users: Andrea Chirinos (andrea@parkgo.cl) and Carlos Perez (test@parkgo.cl), both marked as 'ACTIVO'. Each user row includes an 'Acciones' column with a 'Banear' button. On the left side, a sidebar menu for 'Park Go' includes options like 'Buscar Parking', 'Mi Perfil', 'Mis Vehículos', 'Mis Publicaciones', 'Gestión Arriendos', and 'Administración' (which is currently selected). A 'Cerrar Sesión' button is also present at the bottom of the sidebar.

Usuario	Correo	Estado	Acciones
Andrea Chirinos andrea@parkgo.cl	andrea@parkgo.cl	ACTIVO	
Carlos Perez test@parkgo.cl	test@parkgo.cl	ACTIVO	

# Gestión de Vehículos: Visualización y Registro

La plataforma ofrece una vista detallada y un proceso de registro intuitivo para todos los vehículos en el sistema, asegurando la trazabilidad y el control.

## Vista Detallada de Vehículos

Accede a información completa de cada vehículo, incluyendo su estado, historial y detalles técnicos.

## Mis Vehículos

The screenshot shows a card for a vehicle named "Chevrolet Silverado". It includes a small car icon, a trash can icon for deletion, and a license plate placeholder "ABCDEF CHILE". Below the name, it says "Camioneta - azul".

## Registro Simplificado

Un formulario ágil permite añadir nuevos vehículos al sistema de manera eficiente, minimizando errores y optimizando el tiempo.

### Registrar Vehículo

Patente

Sin guiones ni puntos.

Marca

Modelo

Tipo

Color

Cancelar Guardar

La eficiencia en la gestión de vehículos es clave para la operatividad de nuestra solución de movilidad, y estas herramientas garantizan una administración sin fisuras.

# Plan de Pruebas: Validando la Integridad del Sistema

Nuestro plan de pruebas se enfoca en asegurar la robustez de la entrada de datos críticos y la integridad referencial de la base de datos, pilares fundamentales de la fiabilidad del sistema.

## Caso de Uso: Eliminación de Publicación con Dependencias

Validar el manejo del sistema al intentar eliminar una publicación que tiene dependencias activas.

## Resultado Esperado

El sistema debería:

- Validar previamente si existen relaciones (hijos).
- Bloquear la acción de forma amigable o realizar un "Soft Delete" (marcar como inactivo).

## Fallo Crítico: Eliminación Directa

En la prueba CP-03, la eliminación de una publicación resultó en un error crítico debido a dependencias no controladas.

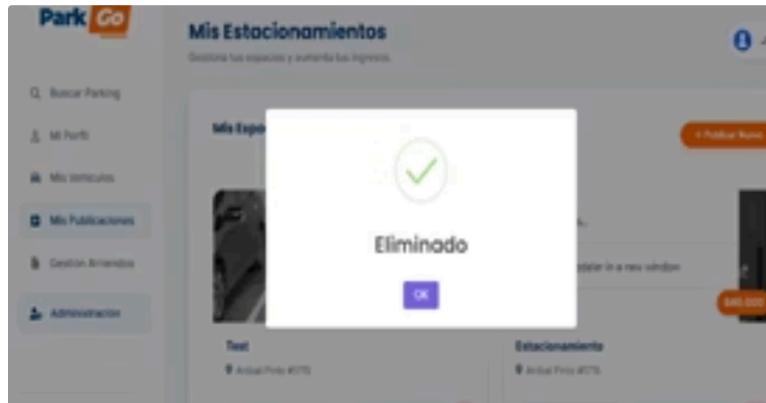
## Plan de Mitigación

Implementar una lógica de "Eliminacion logica(Soft Delete) y modificar los archivos clave: `parkingController.js`, `mis-estacionamientos-logic.js` y `wizard-logic.js`.

Este enfoque proactivo nos permite identificar y corregir vulnerabilidades, reforzando la estabilidad y la consistencia de los datos en toda la aplicación.

# Análisis Detallado de Errores Críticos

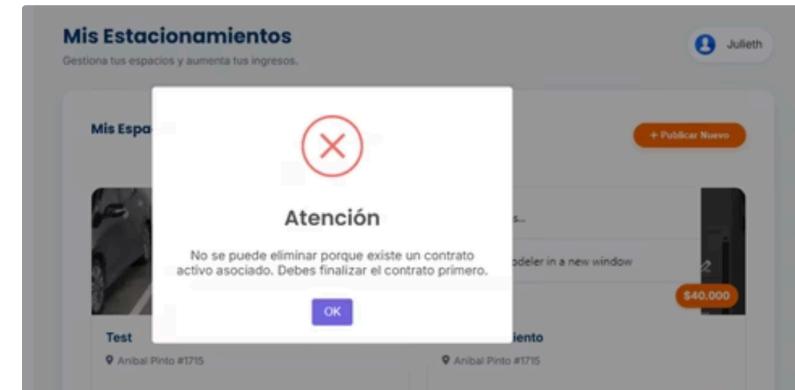
Para entender la raíz del problema, realizamos pruebas específicas que simulan escenarios de eliminación de datos con dependencias, revelando la necesidad de una solución robusta.



## Fallo en la Integridad de Datos

La imagen muestra el error crítico que ocurre al intentar eliminar una publicación vinculada a otros registros, comprometiendo la integridad de la base de datos.

La implementación de una "Eliminación Lógica" se presenta como la estrategia más efectiva para mantener la consistencia y evitar la pérdida accidental de información relacionada.



## Detalle del Mensaje de Error

Este mensaje de error específico indica una violación de la restricción de clave foránea, confirmando que la eliminación directa no es una opción segura.

The screenshot shows a GitHub repository named 'Parkgo'. At the top, there's a profile picture of a woman, the repository name 'Parkgo', and a 'Public' button. To the right are icons for sharing and a pull request. Below this, navigation links show 'main' (with a dropdown arrow), '1 Branch' (with a dropdown arrow), and '1 Tag'. There's also a search bar with 'Go to file' and a 't' icon, and a 'Add file' button with a dropdown arrow.

A commit by 'JuliethTimaure' is listed, adding documentation and diagrams. The commit message is 'Agrega documentación y diagramas del proyecto' and it was made at 'ce1c404 · now'. The commit details show changes to 'Backend' (fixing errors in testing), 'Frontend' (fixing errors in testing), 'docs' (adding documentation and diagrams), 'README.md' (initial upload of the Park Go project v1.0), 'package-lock.json' (fixing errors in testing), and 'sql' (initial upload of the Park Go project v1.0).

# Estructura de carpetas en GitHub

# Tablero Kanban del proyecto

Done 10 Estimate: 0

This has been completed

Parkgo #1

Iniciarizar el proyecto.

Parkgo #2

Instalar dependencias.

Parkgo #3

Diseño de base de datos.

Parkgo #4

Configurar conexion con base de datos.

Parkgo #5

Autenticacion y perfil de usuario.

Parkgo #6

Gestion de vehiculos.

Parkgo #7

Publicacion de estacionameintos.

Parkgo #8

Busqueda y arriendo.

Parkgo #9

Gestion post-venta y Admin

Parkgo #10

Pruebas de flujo completo.

# Reflexiones Finales: Desafíos Superados y Aprendizajes

El camino del desarrollo de software está lleno de retos técnicos y personales que, al ser superados, fortalecen al equipo y mejoran el producto final.

## Dificultades Técnicas

### → Manejo de Asincronía en Cargas Multimedia (Cloudinary)

La sincronización de la subida de archivos a Cloudinary con la actualización de la base de datos fue un desafío clave, exigiendo transacciones atómicas para evitar inconsistencias.

### → Configuración de Base de Datos en la Nube (SSL y Conectividad)

La migración a [Neon.tech](#) presentó obstáculos relacionados con certificados SSL y conectividad, requiriendo ajustes de seguridad y configuración en el entorno de desarrollo.

## Dificultades Personales

### → Gestión del Tiempo y Colaboración

Coordinar el trabajo en equipo y gestionar el tiempo de manera efectiva fue crucial, aprendiendo a equilibrar las responsabilidades individuales con los objetivos del proyecto.

### → Resolución de Problemas y Adaptabilidad

Cada dificultad nos impulsó a investigar, aprender y adaptarnos rápidamente a nuevas tecnologías y metodologías, fomentando un crecimiento constante.

Estas experiencias nos han permitido crecer como profesionales y construir una solución más robusta y fiable para la movilidad urbana.