Desarrollo de un Aplicativo Web para la Gestión y Control Vehicular de un Parqueadero en la Ciudad de Villavicencio

James Fernando Galvis Rodríguez

Corporación Universitaria Minuto de Dios

Rectoría Oriente

Sede / Centro Tutorial Villavicencio (Meta)

Programa Tecnología en Desarrollo de Software

mayo de 2025

Desarrollo de un Aplicativo Web para la Gestión y Control Vehicular de un Parqueadero en la Ciudad de Villavicencio

James Fernando Galvis Rodríguez

Trabajo de Grado presentado como requisito para optar al título de Tecnólogo en Desarrollo de Software

Asesor(a)
Adriana Yeicy Chaparro Prieto
Ingeniera de Sistemas

Corporación Universitaria Minuto de Dios

Rectoría Oriente

Sede / Centro Tutorial Villavicencio (Meta)

Programa Tecnología en Desarrollo de Software

mayo de 2025

Tabla de contenido

Lista de tablas

Lista de figuras

Lista de anexos

Resumen

El presente proyecto tiene como objetivo desarrollar un aplicativo web para la gestión y control Vehicular del parqueadero Parking NoA, ubicado en la ciudad de Villavicencio. Este aplicativo permitirá el registro de entrada y salida de vehículos, calcular tarifas según el tiempo de uso, llevar un seguimiento de los usuarios con mensualidades, visualizar graficas estadísticas, generar reportes en archivos Excel y establecer tarifas para cada tipo de cliente y vehículo.

El aplicativo contará con diferentes niveles de acceso, gestionados a través de un módulo de autenticación que solicitará correo electrónico y contraseña. Una vez autenticado, el usuario verá un menú principal con diversas opciones, tales como: Inicio, registro de vehículos por hora/fracción, registro de clientes mensuales, analíticas, personal, entre otras funcionalidades.

Con este aplicativo, busca optimizar los procesos operativos del parqueadero, mejorando los tiempos de respuestas, tanto para los administradores como para los clientes y siendo una herramienta para la toma de decisiones del parqueadero.

Palabras clave: Prototipo, aplicativo web, gestión de parqueo

Abstract

The objective of this project is to develop a web application for the management and control of vehicles in the parking lot Parking NoA, located in the city of Villavicencio. This application will allow the registration of entry and exit of vehicles, calculate rates according to time of use, keep track of users with monthly fees, view statistical graphs, generate reports in Excel files and set rates for each type of customer and vehicle.

The application will have various levels of access, managed through an authentication module that will request an e-mail address and password. Once authenticated, the user will see a main menu with diverse options, such as: Home, vehicle registration by hour/fraction, monthly customer registration, analytics, personnel, among other functionalities.

With this application, it seeks to optimize the operational processes of the parking lot, improving response times, both for administrators and customers, and being a tool for decision making in the parking lot.

Keywords: Prototype, web application, parking management

Introducción

El presente trabajo se enmarca en la necesidad de desarrollar una herramienta que permita la gestión operativa del parqueadero parking Noa. En este contexto, el proyecto tiene como objetivo principal desarrollar un aplicativo web para el proceso y control vehicular del parqueadero Parking NoA. Esta herramienta busca facilitar el control de entradas y salidas de vehículos, gestionar el registro de clientes (tanto por fracción como mensuales) y administrar de manera centralizada al personal.

Para alcanzar este objetivo, se ha llevado a cabo la identificación de las necesidades específicas de la empresa Parking NoA. Esta fase permitió definir los elementos fundamentales y técnicos del proyecto como la gestión categorizada de clientes y configuración de tarifas. Así mismo, se adoptó una metodología de desarrollo ágil basada en el marco de trabajo SCRUM. Esta metodología facilitó la planificación y ejecución del proyecto en iteraciones cortas (sprints), permitiendo la incorporación de cambios y mejoras de forma continua. Durante la fase de desarrollo se llevaron a cabo actividades clave como el modelado de la base de datos, el diseño de mockups para la interfaz de usuario y la implementación de funcionalidades importantes, tales como el control de acceso, el cálculo automático de tarifas y la generación de reportes.

CAPÍTULO I

1.1 Objetivo General.

Desarrollar un Aplicativo Web para la Gestión y Control Vehicular en el Parqueadero Parking Noa ubicado en la Ciudad de Villavicencio

1.2 Objetivos Específicos

Identificar las necesidades funcionales del aplicativo mediante el levantamiento de requerimientos con los actores del proceso.

Diseñar el modelo de la base de datos relacional que estructure la información de clientes, personal administrativo y configuración tarifaria.

Diseñar los mockups de la interfaz y los módulos funcionales del aplicativo web.

1.3 Planteamiento Del Problema

Parking NoA es una empresa dedicada a ofrecer servicios de parqueo, con aproximadamente 7 años de experiencia en el sector de Amarilo en la ciudad de Villavicencio. Actualmente, la empresa utiliza métodos manuales para gestionar el flujo de vehículos y el control financiero, lo cual genera una serie de inconvenientes que afectan la eficiencia operativa.

Para registrar a los clientes que pagan por hora, se entrega un ticket en el que se registra el número de placa y la hora de entrada. Este método depende en gran medida de la intervención humana, lo que incrementa el riesgo de errores, perdida de información y omisiones, especialmente en horas pico, cuando hay concurrencia de múltiples clientes al mismo tiempo. En el caso de los clientes mensuales, el registro se realiza de forma doble: se registra en un cuaderno físico y, además, se actualiza un archivo Excel. Posteriormente, se emite un recibo de pago. Este proceso fragmentado no solo consume tiempo, sino que también dificulta la consolidación de la información, lo que puede llevar a inconsistencias en los registros.

Adicionalmente, la verificación de las ganancias diarias se lleva a cabo de manera manual, sumando individualmente el valor de los tickets de los clientes por hora, así como el total de nuevos clientes mensuales registrados durante la jornada. Esta tarea, además de ser laboriosa y propensa a errores, retrasa el cierre financiero diario y complica la toma de decisiones basadas en datos actualizados.

1.3.1 Formulación Del Problema

¿Como desarrollar un Aplicativo Web para la Gestión y Control Vehicular del parqueadero Parking Noa en la ciudad de Villavicencio?

1.4 Justificación

Las aplicaciones web son accesibles desde cualquier dispositivo con conexión a internet, por lo cual pueden ser operadas desde cualquier lugar y en diferentes dispositivos. En términos de escalabilidad, estas aplicaciones pueden adaptarse fácilmente al crecimiento de las empresas, pues, permiten ampliar la capacidad de la aplicación, tanto en términos de usuarios como de funcionalidades, sin la necesidad de realizar cambios significativos en la infraestructura de los dispositivos. Además, Las aplicaciones web son más fáciles de integrar con otras herramientas o sistemas existentes, como sistemas de gestión empresarial, bases de datos, aplicaciones de correo electrónico o plataformas de análisis.

El desarrollo de aplicativos webs para gestionar parqueaderos ha cobrado relevancia, ya que permite ofrecer una solución accesible desde cualquier dispositivo con conexión a internet. Este tipo de herramienta facilita la administración de los espacios, el monitoreo en tiempo real, el registro de entradas y salidas de vehículos, y la automatización de procesos como el cálculo de tarifas y la generación de reportes.

Como lo destaca el RUNT, “el parque automotor en las ciudades colombianas ha incrementado destacadamente, con aproximadamente 20 millones de vehículos registrados” (RUNT, 2024), y de los cuales la mayoría son vehículos particulares; crea

la necesidad de contar con soluciones tecnológicas eficientes para la gestión de parqueaderos. La falta de espacios adecuados, el aumento de la congestión vehicular y los altos tiempos de espera han hecho evidente la necesidad de implementar sistemas automatizados que optimicen la operación de los parqueaderos, tanto para los administradores como para los usuarios.

El aplicativo web planteado para el parqueadero Parkin NoA busca permitir a los colaboradores registrar entradas y salidas de vehículos, que luego se almacenará directamente en una base de datos centralizada, eliminando el uso de papel y garantizando que la información esté siempre disponible desde computadoras o dispositivos móviles. El cálculo de tarifas se diferenciará según el tipo de vehículo y cliente, aplicando las tarifas predefinidas por los administradores, lo que reducirá errores en los cobros y posibles conflictos con los usuarios. Asimismo, el sistema generará reportes en ingresos diarios o mensuales con solo unos clics, facilitando la toma de decisiones basadas en datos reales.

Este aplicativo busca convertirse en una herramienta practica para optimizar tareas repetitivas que incrementan tiempo y recursos. Particularmente, al digitalizar los registros, los empleados podrán reducir el tiempo dedicado a llenar formularios y tickets, así como mejorar la atención al cliente. De igual forma, los administradores tendrán acceso inmediato a información relevante, como el número de vehículos atendidos en un día o los ingresos generados por tipo de clientes, sin depender de procesos manuales propensos a errores.

2 CAPITULO II

2.1 Marco Teórico

La gestión eficiente de parqueaderos es un reto cada vez más relevante en el entorno urbano, dado el crecimiento en la cantidad de vehículos y la demanda de espacios de estacionamiento organizados y seguros. El desarrollo de un sistema que permita el control automatizado y eficiente de los parqueaderos es fundamental para optimizar los tiempos de entrada y salida de vehículos, mejorar la administración de los recursos y ofrecer una mejor experiencia a los usuarios. En este sentido, el uso de tecnologías web para la creación de aplicativos ha demostrado ser una alternativa eficaz, debido a su accesibilidad, flexibilidad y capacidad para integrarse con otros sistemas.

Aplicaciones Web para la Gestión de Procesos

Las aplicaciones web se han convertido en una herramienta indispensable para gestionar procesos de forma remota y automatizada en diversas áreas. Según un informe de Statista, el acceso a internet y el uso de dispositivos móviles han crecido de manera significativa (Petrosyan, 2025), lo que ha impulsado la adopción de aplicaciones web para la gestión de servicios y procesos empresariales. En el contexto de los parqueaderos, las aplicaciones web permiten a los usuarios acceder a información en tiempo real, como la disponibilidad de espacios y el tiempo de estacionamiento, desde cualquier lugar y en cualquier momento.

Sistemas de Gestión Automatizada de Parqueaderos

Los sistemas de gestión de parqueaderos comenzaron a desarrollarse como respuesta a la necesidad de optimizar la administración de los espacios de estacionamiento y reducir la congestión. En un estudio realizado por Veloz et al. se destacó que la implementación de parqueaderos inteligentes contribuyó a la reducción de los tiempos para encontrar plazas de parqueo disponible y tarifas exactas por tiempo de uso, beneficiando tanto a la administración de la ciudad como a los ciudadanos (Veloz, Guerrero, Peñafiel, & Salinas, 2024). El análisis demostró la viabilidad del proyecto y un retorno a corto plazo en función de la demanda actual que tiene el sistema de parqueo manual.

Otro ejemplo relevante es el estudio de Aditya et al. quienes proponen un sistema de gestión de parqueaderos basado en tecnologías IoT (Internet of Things). Su propuesta permite un monitoreo en tiempo real de los espacios disponibles y el registro automático de entradas y salidas de vehículos mediante el uso de sensores y cámaras de reconocimiento de matrículas. Este sistema destaca por su capacidad para reducir los errores humanos y optimizar el uso de los espacios, lo que demuestra la importancia de integrar la tecnología en la gestión de parqueaderos. (Aditya, Anwarul, Tanwar, & Vamsi, 2023)

Desarrollo de Aplicaciones Web

El desarrollo de aplicaciones web ha cobrado relevancia en los últimos años, en parte debido a su capacidad para ofrecer servicios accesibles desde cualquier dispositivo conectado a internet. Estas aplicaciones permiten a los usuarios acceder a

funciones específicas sin necesidad de instalar software adicional en sus dispositivos, lo que facilita su uso y gestión (Beaty, 2024).

Un aspecto relevante en el desarrollo de aplicaciones web es la elección de tecnologías adecuadas. Lenguajes como HTML5, CSS3 y JavaScript son fundamentales para la creación de interfaces web interactivas y responsivas. Además, frameworks como React, Angular o Vue.js ofrecen herramientas para la creación de aplicaciones web dinámicas que pueden integrarse con sistemas backend robustos, construidos en lenguajes como Python, Java o Node.js (Osmani, 2023).

Dada la relevancia de las aplicaciones web, la experiencia del usuario (UX) juega un rol fundamental en el éxito de este tipo de aplicaciones. Según Pattiwal:

> El diseño de la experiencia del usuario (UX) es un campo bien establecido que es fundamental para el desarrollo de productos digitales. Empresas de todos los sectores reconocen la importancia de crear experiencias de usuario positivas, ya sea en el diseño de sitios web, aplicaciones o productos físicos (Pattiwal, 2024).

Por esto, para la gestión de parqueaderos, es importante que la aplicación web ofrezca una interfaz intuitiva que permita a los usuarios acceder fácilmente a las funciones principales, como la verificación de espacios disponibles, el registro de entrada y salida de vehículos, y el cálculo de tarifas.

Seguridad en Aplicaciones Web

La seguridad es un aspecto crítico en el desarrollo de cualquier aplicación web, especialmente cuando se manejan datos sensibles, como en el caso de un

parqueadero donde se recopilan y procesan datos de los vehículos y sus dueños. Hoffman destaca que:

> Las aplicaciones web actuales se construyen a menudo sobre una tecnología que no existía hace 10 años. Las herramientas disponibles para crear aplicaciones web han avanzado tanto en ese lapso que a veces parece una especialización totalmente diferente hoy en día (Hoffman, 2024).

Dado esto, múltiples métodos para mejorar la seguridad en aplicaciones web han nacido para responder a la necesidad de aplicaciones seguras. Pant et al. afirma que:

> La autenticación y la autorización son la base de la seguridad de todas las tecnologías presentes en el mundo actual. Existen diferentes tipos de métodos de autenticación que pueden utilizarse en las aplicaciones web para garantizar la seguridad del sitio web y de sus usuarios. Una de las más usada es la autenticación multifactor (MFA) la cual es una versión mejorada de la autenticación básica. A diferencia de la autenticación básica, tiene múltiples capas y fases de autenticación basadas en contraseña, OTP (contraseña de un solo uso), verificación por correo electrónico, verificación de identidad única o incluso a veces alguna pregunta especial (Pant, y otros, 2022).

Desarrollo Ágil y Buenas Prácticas en Software

El uso de metodologías ágiles, como Scrum o Kanban, en el desarrollo de software permite iterar rápidamente sobre un proyecto, obteniendo retroalimentación continua que mejora el producto final (Schwaber & Sutherland, 2020). Para el

desarrollo del prototipo de un aplicativo web para la gestión de un parqueadero, estas metodologías permiten definir claramente las funcionalidades clave desde las primeras etapas del proyecto y ajustarlas de acuerdo con las necesidades de los usuarios.

Según Hooda et al. “El desarrollo ágil de software (ASD) es importante porque hace hincapié en la entrega incremental, la colaboración en equipo y la planificación continua” (Hooda, Sood, Singh, Dalal, & Sood, 2023). Esto es especialmente relevante en proyectos como el desarrollo de aplicaciones para la gestión de parqueaderos, donde los requisitos pueden evolucionar a medida que se identifican nuevas necesidades o mejoras durante las pruebas del prototipo.

En el desarrollo de software, también es importante seguir principios de diseño centrado en el usuario (UCD), los cuales aseguran que las decisiones de diseño estén guiadas por las necesidades y expectativas de los usuarios finales. Este enfoque es clave para garantizar que la aplicación web sea fácil de usar y cumpla con los objetivos planteados en cuanto a la gestión de parqueaderos.

En conclusión, el desarrollo de un prototipo de aplicación web para la gestión de parqueaderos requiere un enfoque integral que considere aspectos como la estructura de un sistema de información eficiente, el uso de tecnologías web modernas, la implementación de medidas de seguridad adecuadas y el empleo de metodologías ágiles para asegurar un producto de calidad. La adopción de estas tecnologías y prácticas no solo optimiza el control de los parqueaderos, sino que también mejora la experiencia de los usuarios, reduce tiempos de espera y permite una mejor administración de los recursos.

2.2 Antecedentes

Diversos proyectos y estudios han abordado el desarrollo de soluciones tecnológicas para la gestión de parqueaderos. Uno de los ejemplos más relevantes es el sistema de Smart Parking implementado en el pleno centro urbano de la ciudad de Barcelona, España. Este sistema, basado en el uso de sensores y tecnología web, permite a los conductores localizar espacios de estacionamiento disponibles en tiempo real y reservarlos desde una aplicación móvil.

Figura 1. Punto de Smart Parking en la ciudad de Barcelona

Fuente: Adaptado de *Barcelona apuesta por el smart parking* [Fotografia], por Redacción TICPymes, 2020, TICPymes(https://www.ticpymes.es/tecnologia/barcelona-implanta-un-sistema-de-busqueda-de-aparcamiento/).

Otro ejemplo de éxito en el uso de aplicaciones web para la gestión de parqueaderos es el sistema implementado por la ciudad de San Francisco, en Estados

Unidos, que utiliza la plataforma SFPark. Esta aplicación web permite a los conductores encontrar espacios de estacionamiento en tiempo real y realizar pagos electrónicos a través de su teléfono móvil. Según SFMTA, el sistema SFPark ha permitido reducir el tiempo de búsqueda de estacionamiento en un 43%, mejorando la movilidad urbana y reduciendo las emisiones de gases contaminantes (SFMTA, 2022).

Figura 2. Parquímetros en la ciudad de San Francisco

Fuente. Adaptado de Demand-Responsive Parking Pricing [Fotografia]. Por SFMTA, 2022, SFMTA (https://www.sfmta.com/demand-responsive-parking-pricing). Todos los derechos reservados 2013-2024 por San Francisco Municipal Transportation Agency (SFMTA) Adaptado con permiso del autor.

Un caso destacado en Latinoamérica es el desarrollo de una aplicación web para la gestión de parqueaderos en la Universidad Nacional Autónoma de México (UNAM) (Hernández & López, 2020). Este proyecto, desarrollado por un grupo de estudiantes de ingeniería, permite a los usuarios de la universidad reservar espacios de estacionamiento y realizar pagos electrónicos de manera sencilla. Según Hernández y

López, el sistema ha mejorado considerablemente la administración de los parqueaderos dentro del campus universitario, reduciendo los tiempos de espera y mejorando la experiencia de los usuarios.

En Colombia, otras propuestas han surgido para dar respuesta a esta problemática. Tal es el caso del aplicativo prototipo para la solución a la búsqueda de parqueaderos en la ciudad de Bogotá desarrollado por Anzola el cual presenta una propuesta de solución a la problemática de parqueaderos en la ciudad a través de la incursión y uso eficiente de una aplicación móvil que permita rentar parqueaderos privados de conjuntos residenciales principalmente, por horas parciales de cada día en las proximidades de industrias, empresas, gimnasios, coworking y demás lugares de alto flujo de personas con vehículos, cuyos residentes o propietarios no hacen uso 24 horas del mismo permitiendo la alternancia de las zonas de parqueo. (Anzola, 2024)

En un contexto similar e involucrando desarrollo de hardware, Lozada y Martínez en su trabajo de grado proponen un prototipo Arduino enfocado en realizar tareas para el software en un parqueadero en específico en la ciudad de Villavicencio, con el fin de facilitar el manejo de datos de cualquier parqueadero en donde se instale el sistema. El prototipo tiene la capacidad de capturar la información necesaria para el funcionamiento del programa, como lo son el reconocimiento de placas, ocupación de puestos y el ingreso y salida de vehículos. Con estos datos se busca cubrir las diferentes actividades de control, administración y gestión que se vuelven complicadas de realizar sin un sistema, y a la vez, permitir generar un análisis de datos que permita al estacionamiento obtener información estadística (Lozada Quintero & Martinez Quevedo, 2023).

Los antecedentes presentados demuestran que el desarrollo de un aplicativo web para la gestión de parqueaderos es una solución viable y necesaria para enfrentar los desafíos actuales en la administración de estos espacios. Los avances en tecnologías web, combinados con metodologías ágiles de desarrollo de software y sistemas de seguridad robustos, permiten la creación de herramientas accesibles, eficientes y seguras que mejoran tanto la experiencia del usuario como la operación de los parqueaderos. Además, los estudios y proyectos previos confirman la efectividad de estas soluciones tecnológicas en diversos contextos, lo que respalda la pertinencia del desarrollo de un aplicativo web para la gestión de parqueaderos.

2.3 Marco Conceptual

Software de Gestión: Un software de gestión es una aplicación informática diseñada para ayudar a las empresas a administrar y controlar sus operaciones de manera eficiente (Richards, 2017).

Frontend: Es la parte de una aplicación o sitio web con la que interactúa el usuario. Incluye elementos visuales y de usuario, como botones y formularios (Richards, 2017).

Cliente: Es un programa o aplicación informática que se ejecuta en un dispositivo (como una computadora o un smartphone) y que solicita servicios o recursos a un servidor (Richards, 2017).

Servidor: Ordenador empleado por muchos usuarios para realizar una tarea específica, como ejecutar aplicaciones de red o de Internet (Stair, Perew, & Vance, 2024).

Backend: Es la parte de una aplicación o sitio web que se encuentra en el servidor y se encarga del procesamiento de datos, la lógica de negocio y la gestión de la base de datos (Richards, 2017).

Arquitectura MVC: Es un patrón de arquitectura de software que se utiliza para desarrollar interfaces de usuario dividiendo una aplicación en tres componentes interconectados: modelo, vista y controlador. Su separación permite a los desarrolladores escribir aplicaciones más limpias, mantenibles y escalables (Freeman, 2023).

Base de Datos Relacional: Es una forma sencilla pero muy útil de organizar datos en colecciones de tablas bidimensionales denominadas relaciones. Cada fila de la tabla representa una entidad y cada columna representa un atributo de esa entidad. Cada fila de una tabla se identifica unívocamente mediante una clave primaria y el tipo de datos que puede contener una columna de la tabla puede especificarse como número entero, número decimal, fecha, texto, entre otros (Stair, Perew, & Vance, 2024).

Sistema de gestión de bases de datos (DBMS): Conjunto de programas utilizados para acceder a una base de datos y gestionarla, así como para proporcionar una interfaz entre la base de datos y sus usuarios y otros programas de aplicación (Stair, Perew, & Vance, 2024).

Consulta: Desde la perspectiva de una base de datos, una consulta es una petición específica hecha al DBMS para la manipulación de datos, por ejemplo, leer o actualizar los datos (Stair, Perew, & Vance, 2024).

SQL (Structured Query Language): Es un lenguaje de programación específico para acceder a los datos almacenados en una base de datos relacional y manipularlos (Stair, Perew, & Vance, 2024).

2.4 Marco Legal

Ley Estatutaria 1581 del 17 de octubre del 2012. Ley para la protección de datos personales (Art. 1°, *Objeto*): "La presente ley tiene por objeto desarrollar el derecho constitucional que tienen todas las personas a conocer, actualizar y rectificar las informaciones que se hayan recogido sobre ellas en bases de datos o archivos, y los demás derechos, libertades y garantías constitucionales a que se refiere el artículo 15 de la Constitución Política; así como el derecho a la información consagrado en el artículo 20 de la misma." (Congreso de Colombia, 2012, Artículo 1°).

Ley de Habeas Data de Colombia. Ley 1581 de 2012. Diario Oficial de la República de Colombia, Número 48.425, 17 de octubre de 2012: "La presente ley tiene por objeto desarrollar el derecho constitucional que tienen todas las personas a conocer, actualizar y rectificar las informaciones que se hayan recogido sobre ellas en bancos de datos , y los demás derechos, libertades y garantías constitucionales relacionadas con la recolección, tratamiento y circulación de datos personales a que se refiere el artículo 15 de la Constitución Política, así como el derecho a la información

establecido en el artículo 20 de la Constitución Política, particularmente en relación con la información financiera y crediticia, comercial, de servicios y la proveniente de terceros países.” (Congreso de Colombia, 2012, Artículo 1°).

3 CAPITULO III

3.1 Tipo de Investigación

El presente proyecto se basa en las líneas de investigación definidas por Universidad Minuto de Dios, específicamente la línea de investigación “Innovaciones Sociales y Productivas”, está adscrito al semillero de investigación Data science y al grupo de investigación GITSAI.

El enfoque de investigación es **cualitativo**, ya que se centra en la recopilación y análisis de datos no numéricos, obtenidos a través de entrevistas, observaciones y revisión de documentos. El **tipo de investigación** es **tecnológica** dado que se propone una intervención mediante el desarrollo de una solución tecnológica para una empresa especifica. Además, se trata de un **estudio de caso**, puesto que el objetivo principal es recolectar información de los actores involucrados en la gestión de un parqueadero existente desde los administradores hasta los usuarios, con el fin de diseñar una aplicación web que facilite dicho proceso. Esta herramienta permitirá a los administradores y colaboradores realizar operaciones de control de manera más rápida, eficiente y precisa, mejorando así la gestión del parqueadero.

3.2 Población y Muestra

En el marco del enfoque cualitativo y la metodología de estudio de caso, la población está conformada por todos los parqueaderos ubicados en la ciudad de Villavicencio. La muestra se centra en un caso específico: el parqueadero Parking NoA.

Este parqueadero ha sido seleccionado por su relevancia y disposición para participar en el estudio, permitiendo un análisis detallado de sus procesos y sistemas de gestión.

3.3 Instrumentos y Técnicas de Recolección de Información

Para este proyecto, se utilizó una **entrevista semiestructurada** como instrumento principal de recolección de información. Esta entrevista fue dirigida al administrador y a un colaborador del parqueadero Parking NoA, con el objetivo de identificar sus necesidades, requerimientos y problemáticas en relación con la gestión de las operaciones diarias. La información obtenida permitió comprender el entorno operativo del parqueadero y establecer las bases para el desarrollo de una solución tecnológica adecuada.

3.4 Análisis De Datos

Los resultados de la entrevista, presentados en el Anexo 1, evidencian que el parqueadero Parking NoA enfrenta dificultades operativas que afectan la eficiencia y precisión en la gestión. Con base en el análisis de esta información, se identificaron los siguientes requerimientos:

- Registro automatizado de entradas y salidas de los vehículos
- Cálculo de tarifas y la gestión centralizada de clientes por hora y mensuales

- Diseñar una interfaz sencilla y adaptable que no suponga problema para aquellos que operan la aplicación.

- Implementación y generación de reportes y generación de estadísticas operativas, preferiblemente encapsuladas en un panel de control actualizado en tiempo real.

- Gestión de empleados con control de accesos basados en roles, fundamental para garantizar la seguridad y transparencia de la información tratada en la aplicación.

Con base en esto, se busca crear un aplicativo web que implemente cada uno de estos puntos como propuesta de solución a los problemas operacionales del parqueadero Parking NoA.

4 CAPITULO IV

4.1 Metodología De Desarrollo De Software

Para el desarrollo del aplicativo, se adoptó el marco de trabajo SCRUM, seleccionada por su enfoque iterativo, adaptabilidad a cambios y capacidad para entregar valor incremental en ciclos cortos (sprints). Esta elección se alinea con la línea de investigación institucional "Gestión de Proyectos de Software" del programa, la cual promueve prácticas modernas para optimizar procesos de desarrollo. A continuación, se detallan los elementos clave implementados:

Herramientas y Repositorio

- **Control de versiones:** Se utilizó **Git** como sistema de control de versiones, junto con la plataforma **GitHub** para alojar el repositorio del proyecto.
    - **Enlace al repositorio:** https://github.com/JamesGalvis/Parking-NoA.gitel
- **Gestión de tareas:** Se empleo **Trello** para organizar el *Product Backlog, historias de usuario y* los sprints.
    - **Enlace al Trello:** https://trello.com/invite/b/67de24818a6798ac0f9eb6cc/ATTI8e637943a45e81b13176c31ffb50f8c75C7AE3B6/app-web-gestion-y-control-de-parqueadero

4.2 Análisis De Requerimientos

El análisis de requisitos es una etapa fundamental para identificar las necesidades y especificaciones del aplicativo web, asegurando que cumpla con los objetivos propuestos y las expectativas de los usuarios administrativos.

A continuación, se presentan los requerimientos funcionales en la tabla 1.

Requerimientos Funcionales

Tabla 1. Requerimientos funcionales

ID Requerimiento	Descripción del Requerimiento
RF01	Implementar un sistema de login seguro para que los administradores y empleados puedan autenticarse en el aplicativo mediante sus credenciales (correo electrónico y contraseña).
RF02	Proporcionar una funcionalidad de cierre de sesión (logout) eficiente para que los usuarios puedan finalizar su sesión de manera segura y proteger los datos del sistema.
RF03	Establecer un control de acceso basado en roles (RBAC) para permitir o restringir funciones según el rol del usuario (administrador o empleado).

RF04	Gestionar los tipos de clientes (por hora y mensuales) para clasificar a los usuarios del parqueadero según su modalidad de uso y facilitar el seguimiento de sus registros.
RF05	Configurar los tipos de vehículos y clientes para definir las categorías que serán utilizadas en el sistema y personalizar las operaciones según estas clasificaciones.
RF06	Establecer tarifas personalizadas según el tipo de cliente y vehículo para aplicar costos diferenciados en función de las políticas del parqueadero.
RF07	Registrar y asignar roles a los empleados para gestionar sus permisos y responsabilidades dentro del sistema de acuerdo con su función en el parqueadero.
RF08	Registrar, editar y calcular automáticamente los costos para los clientes por hora para agilizar el proceso de facturación basado en el tiempo de estadía.

RF09	Gestionar completamente a los clientes mensuales con seguimiento de vencimientos para mantener un control preciso de sus suscripciones y fechas de renovación.
RF10	Generar gráficos de ganancias por tipo de cliente y periodo para visualizar el desempeño financiero del parqueadero de manera clara e interactiva.
RF11	Generar informes con exportación a Excel para permitir a los administradores analizar datos operativos y financieros en un formato accesible y editable.
RF12	Mostrar estadísticas en tiempo real y visualizar las tarifas en el panel principal para ofrecer a los usuarios del sistema (administradores y empleados) una visión inmediata del estado del parqueadero.
RF13	Permitir a los usuarios (administradores y empleados) editar su información personal (nombre, correo electrónico, teléfono) para mantener sus datos actualizados en el sistema.

RF14	Facilitar el cambio de contraseña de forma segura para que los usuarios puedan proteger su cuenta mediante una actualización periódica de sus credenciales.
RF15	Mostrar la información del perfil de manera clara y accesible para que los usuarios puedan consultar sus datos personales y rol asignado en el sistema.

Fuente: Propia del autor

Requerimientos no Funcionales

De igual forma, los requerimientos no funcionales se encuentran listados a continuación en la tabla 2.

Tabla 2. Requerimientos no funcionales

ID Requerimiento	Descripción del Requerimiento
RNF01	Garantizar que las operaciones críticas del sistema se ejecuten en menos de 3 segundos para proporcionar una experiencia de usuario fluida y eficiente.
RNF02	Asegurar soporte para un aumento de registros sin pérdida de rendimiento para que el sistema pueda manejar un volumen creciente de datos sin comprometer su funcionalidad.

RNF03	Implementar el cifrado de contraseñas y un control de acceso robusto para proteger la información sensible de los usuarios y garantizar la seguridad del sistema.
RNF04	Diseñar una interfaz intuitiva y adaptable que soporte modos oscuro y claro para mejorar la usabilidad y la experiencia del usuario en diferentes condiciones de visualización.
RNF05	Garantizar que el sistema funcione correctamente en navegadores modernos y que su diseño sea responsivo para adaptarse a distintos dispositivos y resoluciones.
RNF06	El sistema debe estar disponible al menos el 99% del tiempo mensual, garantizando alta disponibilidad y mínima interrupción del servicio.
RNF07	Facilitar la implementación de actualizaciones del sistema para permitir la incorporación de nuevas funcionalidades o mejoras sin interrupciones significativas.

RNF08	Permitir la implementación del sistema en servidores locales o en la nube para ofrecer flexibilidad en el despliegue según las necesidades del parqueadero.
RNF09	Incluir manuales de uso detallados para que los usuarios puedan aprender a utilizar el sistema de manera eficiente y resolver dudas operativas.

Fuente: Propia del autor

Sprint Planning

La planificación de los sprints se llevó a cabo con base en los requerimientos funcionales y no funcionales identificados a través de la entrevista realizada al administrador y colaborador del parqueadero Parking NoA, cuyos detalles se encuentran en el **Anexo 1**. Durante esta entrevista, se recopilaron las necesidades operativas clave, como la automatización del registro de entradas y salidas, el cálculo de tarifas, la gestión de clientes mensuales y por hora, la generación de reportes y la seguridad de los datos. Estos requerimientos se tradujeron en historias de usuario que reflejaron las prioridades del negocio, priorizando inicialmente las funcionalidades esenciales para la operación diaria del parqueadero, como la autenticación segura y la configuración básica del sistema, para garantizar una base sólida antes de implementar módulos más avanzados como analíticas y reportes. La priorización se realizó utilizando un enfoque basado en el valor entregado al usuario y la dependencia entre funcionalidades, asegurando que los sprints iniciales abordaran las necesidades

críticas identificadas por el personal del parqueadero, mientras que los sprints posteriores incorporaron mejoras y características adicionales.

Sprint 1: Fundamentos del Sistema

En el primer sprint, se sentaron las bases del sistema enfocándose en la autenticación y seguridad. Se implementó un sistema de login seguro para administradores y empleados utilizando Auth.js, junto con una funcionalidad eficiente de cierre de sesión. Se configuró la protección de rutas en Next.js para garantizar el control de acceso basado en roles (RBAC), y se diseñó la interfaz inicial con ShadCN y Tailwind CSS, ofreciendo un diseño intuitivo y adaptable a modos oscuro y claro. Además, se realizaron pruebas unitarias y de integración para validar el correcto funcionamiento de estas características esenciales.

Tabla 3. Sprint 1: Fundamentos del sistema

Aspecto	**Detalles**
Duración	2 semanas
Objetivo	Establecer la base de autenticación y seguridad del sistema
Requerimientos funcionales	– **RF01:** Implementar un sistema de login seguro para que los administradores y empleados puedan autenticarse en el aplicativo mediante sus credenciales (correo electrónico y contraseña). – **RF02:** Proporcionar una funcionalidad de cierre de sesión (logout) eficiente para que los

	usuarios puedan finalizar su sesión de manera segura y proteger los datos del sistema.
Requerimientos no funcionales	– **RNF03:** Implementar el cifrado de contraseñas y un control de acceso robusto para proteger la información sensible de los usuarios y garantizar la seguridad del sistema. – **RNF04:** Diseñar una interfaz intuitiva y adaptable que soporte modos oscuro y claro para mejorar la usabilidad y la experiencia del usuario en diferentes condiciones de visualización.
Historias de usuario	– **HU01:** Autenticación de usuarios**.** – **HU02:** Registro de nuevos usuarios.
Tareas Realizadas	– Implementación del sistema de autenticación con Auth.js – Configuración de la protección de rutas en Next.js. – Diseño inicial de la interfaz con ShadCN y Tailwind CSS. – Pruebas unitarias y de integración.
Resultado	Inicio de sesión (login) y cerrar sesión (logout) funcionales con control de acceso basado en roles y seguridad básica implementada.

Fuente: Propia del autor

Sprint 2: Configuración del Parqueadero

Durante el segundo sprint, se desarrolló la configuración básica del parqueadero, abarcando la gestión de tipos de clientes, vehículos y tarifas. Se crearon módulos para registrar, editar y eliminar estos elementos, permitiendo a los administradores personalizar el sistema según las necesidades del negocio. Se optimizaron las consultas a la base de datos con Prisma y CockroachDB para asegurar un rendimiento eficiente, incluso con un volumen creciente de datos. El resultado fue un módulo funcional que soporta las operaciones diarias del parqueadero.

Tabla 4. Sprint 2: Configuración del parqueadero

Aspecto	**Detalles**
Duración	2 semanas
Objetivo	Implementar la configuración básica del parqueadero, incluyendo tipos de clientes, vehículos y tarifas.
Requerimientos funcionales	– **RF05:** Configurar los tipos de vehículos y clientes para definir las categorías que serán utilizadas en el sistema y personalizar las operaciones según estas clasificaciones. – **RF06:** Establecer tarifas personalizadas según el tipo de cliente y vehículo para aplicar costos diferenciados en función de las políticas del parqueadero.

Requerimientos no funcionales	– **RNF01:** Garantizar que las operaciones críticas del sistema se ejecuten en menos de 3 segundos para proporcionar una experiencia de usuario fluida y eficiente. – **RNF02:** Asegurar soporte para un aumento de registros sin pérdida de rendimiento para que el sistema pueda manejar un volumen creciente de datos sin comprometer su funcionalidad.
Historias de usuario	– **HU05:** Configuración de tarifas**.**
Tareas Realizadas	– Desarrollo del módulo de configuración para tipos de clientes y vehículos con operaciones de creación y eliminación. – Implementación del módulo de tarifas personalizadas. – Optimización de consultas a la base de datos con Prisma y CrockroachDB.
Resultado	Módulo de configuración del parqueadero funcional, con tarifas personalizadas y rendimiento optimizado

Fuente: Propia del autor

Sprint 3: Gestión de Empleados

En el tercer sprint, se trabajó en la gestión de empleados, integrando la asignación de roles y el control de accesos. Se desarrolló un módulo CRUD (Crear, Leer, Actualizar, Eliminar) para empleados, permitiendo a los administradores gestionar

usuarios del sistema de manera efectiva. Se ajustó la interfaz para garantizar compatibilidad y responsividad en distintos dispositivos, y se realizaron pruebas para confirmar que el control de accesos basado en roles restringiera correctamente las funcionalidades según los permisos asignados.

Tabla 5. Sprint 3: Gestión de empleados

Aspecto	**Detalles**
Duración	2 semanas
Objetivo	Desarrollar la gestión completa de empleados, integrando roles y control de accesos.
Requerimientos funcionales	– **RF07:** Registrar y asignar roles a los empleados para gestionar sus permisos y responsabilidades dentro del sistema de acuerdo con su función en el parqueadero.
Requerimientos no funcionales	– **RNF05:** Garantizar que el sistema funcione correctamente en navegadores modernos y que su diseño sea responsivo para adaptarse a distintos dispositivos y resoluciones. – **RNF07:** Facilitar la implementación de actualizaciones del sistema para permitir la incorporación de nuevas funcionalidades o mejoras sin interrupciones significativas.
Historias de usuario	– **HU10:** Gestión de empleados**.**

Tareas Realizadas	– Implementación del módulo CRUD para empleados con asignación de roles. – Integración con el sistema de control de accesos. – Ajuste de la interfaz para compatibilidad y responsividad.
Resultado	Gestión de empleados operativa, con control de accesos basado en roles y diseño responsivo.

Fuente: Propia del autor

Sprint 4: Gestión de Clientes

El cuarto sprint se centró en la gestión de clientes por hora y mensuales. Se implementaron módulos para registrar y editar clientes por hora con cálculo automático de costos, así como para gestionar clientes mensuales con seguimiento de vencimientos. Se integraron notificaciones automáticas mediante cron jobs para alertar a los clientes sobre la renovación de sus servicios. Además, se realizaron pruebas de disponibilidad y se desplegó el sistema en la nube, logrando un 99% de tiempo de actividad.

Tabla 6. Sprint 4: Gestión de clientes

Aspecto	**Detalles**
Duración	2 semanas
Objetivo	Implementar la gestión de clientes por hora y mensuales, utilizando las tarifas configuradas.

Requerimientos funcionales	– **RF08:** Registrar, editar y calcular automáticamente los costos para los clientes por hora para agilizar el proceso de facturación basado en el tiempo de estadía. – **RF09:** Gestionar completamente a los clientes mensuales con seguimiento de vencimientos para mantener un control preciso de sus suscripciones y fechas de renovación.
Requerimientos no funcionales	– **RNF06:** El sistema debe estar disponible al menos el 99% del tiempo mensual, garantizando alta disponibilidad y mínima interrupción del servicio. – **RNF08:** Permitir la implementación del sistema en servidores locales o en la nube para ofrecer flexibilidad en el despliegue según las necesidades del parqueadero.
Historias de usuario	– **HU03:** Gestión de clientes por hora**.** – **HU04:** Gestión de Clientes Mensuales. – **HU07:** Notificaciones automáticas**.**
Tareas Realizadas	– Desarrollo del módulo para registrar clientes por hora con cálculo automático de costos – Implementación del módulo para clientes mensuales con seguimiento de vencimientos y notificaciones automáticas mediante **cron jobs**. – Pruebas de disponibilidad y despliegue en la nube.
Resultado	Gestión completa de clientes por hora y mensuales, con notificaciones automáticas y alta disponibilidad.

Fuente: Propia del autor

Sprint 5: Perfil de Usuario

En el quinto sprint, se desarrolló la funcionalidad de perfil de usuario, permitiendo a los usuarios gestionar su información personal y la seguridad de sus cuentas. Se crearon formularios validados para editar datos como nombre, correo y teléfono, junto con un proceso seguro para cambiar contraseñas. La interfaz del perfil se diseñó con ShadCN y Tailwind CSS, asegurando que fuera intuitiva y responsiva. Se llevaron a cabo pruebas de seguridad para verificar el cifrado de contraseñas y la protección de datos sensibles.

Tabla 7. Sprint 5: Perfil de usuario

Aspecto	**Detalles**
Duración	2 semanas
Objetivo	Implementar la funcionalidad de perfil de usuario, permitiendo a los usuarios gestionar su información personal y seguridad de la cuenta.
Requerimientos funcionales	– **RF13:** Permitir a los usuarios (administradores y empleados) editar su información personal (nombre, correo electrónico, teléfono) para mantener sus datos actualizados en el sistema. – **RF14:** Facilitar el cambio de contraseña de forma segura para que los usuarios puedan proteger su

	cuenta mediante una actualización periódica de sus credenciales. – **RF15:** Mostrar la información del perfil de manera clara y accesible para que los usuarios puedan consultar sus datos personales y rol asignado en el sistema.
Requerimientos no funcionales	– **RNF03:** Implementar el cifrado de contraseñas y un control de acceso robusto para proteger la información sensible de los usuarios y garantizar la seguridad del sistema. – **RNF04:** Diseñar una interfaz intuitiva y adaptable que soporte modos oscuro y claro para mejorar la usabilidad y la experiencia del usuario en diferentes condiciones de visualización. – **RNF05:** Garantizar que el sistema funcione correctamente en navegadores modernos y que su diseño sea responsivo para adaptarse a distintos dispositivos y resoluciones.
Historias de usuario	– Desarrollo del módulo de perfil de usuario con formularios validados para la edición de información personal (nombre, correo electrónico, teléfono).

Tareas Realizadas	– Implementación de la funcionalidad de cambio de contraseña, asegurando el cifrado de la nueva contraseña y la validación de la contraseña actual. – Integración con el sistema de autenticación para actualizar la información del usuario en la base de datos. – Diseño de la interfaz del perfil utilizando **ShadCN** y **Tailwind CSS**, garantizando que sea intuitiva, responsiva y compatible con el modo oscuro/claro.
Resultado	Módulo de perfil de usuario funcional, permitiendo a los usuarios gestionar su información personal y seguridad de la cuenta de manera eficiente y segura.

Fuente: Propia del autor

Sprint 6: Analíticas y Reportes

El sexto sprint se dedicó a implementar las funcionalidades de analíticas y reportes. Se desarrollaron gráficos interactivos con ShadCN para visualizar ganancias por tipo de cliente y periodo, y un módulo para generar informes exportables a Excel. Se integraron estadísticas en tiempo real en el panel principal, ofreciendo a los administradores una visión clara del desempeño del parqueadero.

Tabla 8. Sprint 6: Analíticas y reportes

Aspecto	**Detalles**
Duración	2 semanas

Objetivo	Desarrollar las funcionalidades de analíticas y reportes del sistema.
Requerimientos funcionales	– **RF10:** Generar gráficos de ganancias por tipo de cliente y periodo para visualizar el desempeño financiero del parqueadero de manera clara e interactiva. – **RF11:** Generar informes con exportación a Excel para permitir a los administradores analizar datos operativos y financieros en un formato accesible y editable. – **RF12**: Mostrar estadísticas en tiempo real y visualizar las tarifas en el panel principal para ofrecer a los usuarios del sistema (administradores y empleados) una visión inmediata del estado del parqueadero.
Requerimientos no funcionales	– **RNF09:** Incluir manuales de uso detallados para que los usuarios puedan aprender a utilizar el sistema de manera eficiente y resolver dudas operativas.
Historias de usuario	– **HU03:** Generación de Reportes. – **HU08:** Interfaz Responsiva**.**

Tareas Realizadas	– Implementación de gráficos interactivos para analíticas con **ShadCN.** – Desarrollo del módulo de reportes con exportación a Excel. – Integración de estadísticas en tiempo real en el panel principal.
Resultado	Sistema completo con analíticas, reportes y documentación, listo para su uso.

Fuente: Propia del autor

4.3 Historias De Usuario

Estas historias describen las necesidades de los usuarios finales y cómo interactúan con el sistema para cumplir sus objetivos.

Tabla 9. Historia de usuario autenticación de usuario

ID	Historia de Usuario	Descripción	Criterios de Aceptación
HU01	Autenticación de Usuarios	Como usuario registrado, quiero poder iniciar sesión en el sistema para acceder a las funcionalidades disponibles según mi rol.	1. Validar credenciales de acceso. 2. Mostrar mensajes de error si las credenciales sin incorrectas. 3. Administradores con acceso a opciones avanzadas.

Fuente: Propia del autor

Tabla 10. Historia de usuario registro de nuevos usuarios

ID	Historia de Usuario	Descripción	Criterios de Aceptación
HU02	Registro de Nuevos Usuarios	Como administrador, quiero poder registrar nuevos usuarios en el sistema, para que puedan acceder con los permisos correspondientes.	1. El formulario debe validar los campos obligatorios. 2. Asignar roles al momento del registro. 3. Mostrar mensaje de confirmación al crear un usuario.

Fuente: Propia del autor

Tabla 11. Historia de usuario gestión de clientes por hora

ID	Historia de Usuario	Descripción	Criterios de Aceptación
HU03	Gestión de Clientes por Hora	Como empleado o administrador, quiero registrar clientes por hora, para calcular el costo del estacionamiento al momento de su salida.	1. El sistema debe registrar automáticamente la hora de entrada al crear un nuevo cliente y la hora de salida al confirmar el pago. 2. Calcular costo automáticamente según tarifas.

			3. Permitir cancelar registros en caso de errores.

Fuente: Propia del autor

Tabla 12. Historia de usuario gestión de clientes mensuales

ID	Historia de Usuario	Descripción	Criterios de Aceptación
HU04	Gestión de Clientes Mensuales	Como empleado o administrador, quiero gestionar clientes con membresías mensuales, para asegurar el control de pagos y renovaciones.	1. Permitir registrar, editar y eliminar clientes mensuales. 2. Calcular automáticamente las fechas de vencimiento. 3. Enviar notificaciones automáticas de renovación.

Fuente: Propia del autor

Tabla 13. Historia de usuario configuración de tarifas

ID	Historia de Usuario	Descripción	Criterios de Aceptación

HU05	Configuración de Tarifas	Como administrador, quiero configurar las tarifas de estacionamiento, para ajustar los costos según las políticas de la empresa.	1. Permitir modificar tarifas en cualquier momento. 2. Aplicar tarifas a todas las transacciones nuevas. 3. Notificar al usuario en caso de conflicto al modificar tarifas.

Fuente: Propia del autor

Tabla 14. Historia de usuario gestión de reportes

ID	**Historia de Usuario**	**Descripción**	**Criterios de Aceptación**
HU06	Generación de Reportes	Como administrador, quiero generar reportes de ingresos y estadísticas de uso, para analizar el desempeño del estacionamiento.	1. Incluir ingresos diarios, mensuales y por tipo de cliente. 2. Permitir exportación en formato Excel. 3. Filtrar reportes por rango de fechas.

Fuente: Propia del autor

Tabla 15. Historia de usuario notificaciones automáticas

ID	Historia de Usuario	Descripción	Criterios de Aceptación
HU07	Notificaciones Automáticas	Como administrador, quiero que el sistema envíe notificaciones automáticas a clientes mensuales, para recordarles la renovación de su membresía.	1. Enviar notificaciones antes del vencimiento del servicio. 2. Enviar notificación adicional el día después del vencimiento si no renuevan.

Fuente: Propia del autor

Tabla 16. Historia de usuario interfaz responsiva

ID	Historia de Usuario	Descripción	Criterios de Aceptación
HU08	Interfaz Responsiva	Como usuario, quiero que la interfaz sea accesible desde cualquier dispositivo, para poder utilizar el aplicativo en PC, móvil o tableta.	1. Ajustar la interfaz automáticamente al tamaño de la pantalla. 2. Todas las funcionalidades accesibles en cualquier dispositivo. 3. Diseño consistente entre escritorio y móvil.

Fuente: Propia del autor

Tabla 17. Historia de usuario protección de rutas

ID	Historia de Usuario	Descripción	Criterios de Aceptación
HU09	Protección de Rutas	Como administrador quiero que las rutas estén protegidas según el rol del usuario, para garantizar la seguridad de los datos y funcionalidades.	1. Restringir acceso a rutas protegidas para usuarios no autenticados. 2. Mostrar solo funcionalidades autorizadas según el rol. 3. Redirigir al panel principal en caso de intento de acceso no autorizado.

Fuente: Propia del autor

Tabla 18. Historia de usuario gestión de empleados

ID	Historia de Usuario	Descripción	Criterios de Aceptación
HU010	Gestión de empleados	Como administrador, quiero gestionar a los empleados dentro de sistema, para asignarles roles, controlar accesos y mantener la información actualizada.	1. Permitir registrar, editar y eliminar empleados. 2. Asignar roles de Empleado o Administrador a cada usuario.

Fuente: Propia del autor

Tabla 19. Historia de usuario gestión de perfil de usuario

ID	Historia de Usuario	Descripción	Criterios de Aceptación
HU011	Gestión del Perfil de Usuario	Como usuario autenticado, quiero poder editar mi información personal y cambiar mi contraseña para mantener mis datos actualizados y seguros.	1. Acceso fácil a la sección de perfil desde el menú principal. 2. Edición de la imagen de perfil, nombre y teléfono de contacto, con validación y guardado de los cambios. 3. Proceso seguro para cambiar la contraseña, incluyendo verificación de la contraseña actual, confirmación de la nueva y cumplimiento de requisitos de seguridad.

Fuente: Propia del autor

4.4 Diseño De La Aplicación

La fase de diseño es fundamental en el desarrollo del software, ya que permite definir la estructura, flujo de información y la interacción entre los diferentes

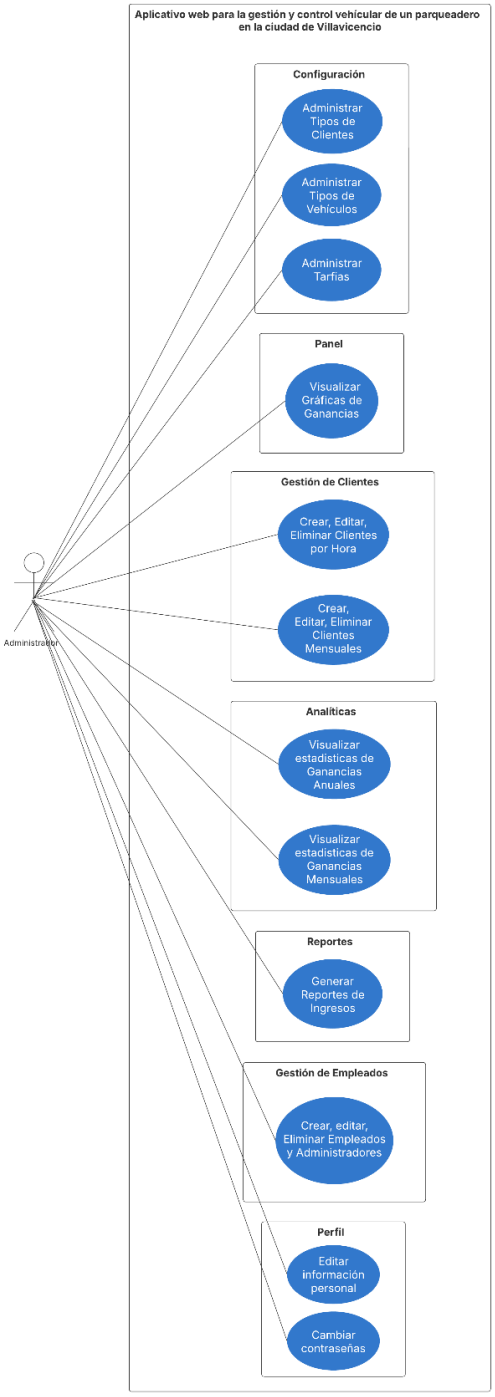
componentes del sistema. A través de los casos de uso, diagramas de secuencias, mockups y diagramas de clases, se establece una guía clara para la implementación del software. Con estos elementos se puede facilitar la comprensión de los procesos, optimizando la experiencia del usuario y garantizando un desarrollo alineado con los requerimientos.

4.4.1 Casos De Uso

Administrador

El siguiente diagrama de casos de uso representa las principales interacciones del administrador dentro del Sistema de Gestión de Parqueadero, detallando las funcionalidades y acciones que puede realizar en cada módulo. Este diagrama permite visualizar de manera clara cómo el administrador gestiona clientes, empleados, configuraciones y reportes dentro del sistema, asegurando un control eficiente de las operaciones del parqueadero.

Figura 3. Diagrama de casos de uso para el administrador

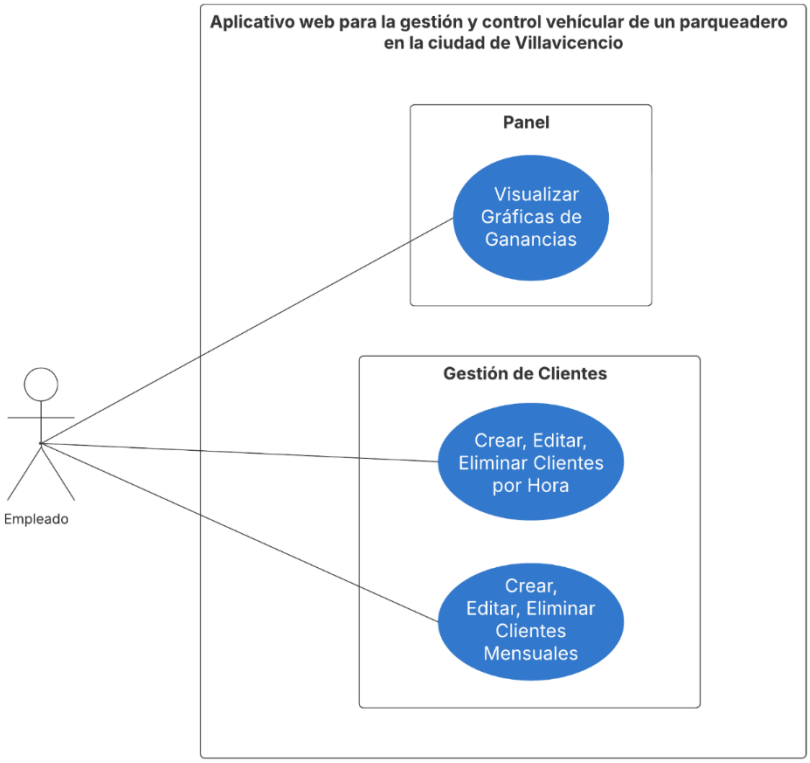


Fuente: Propia del autor

Empleado

El siguiente diagrama de caso de uso representa las funcionalidades disponibles para un Empleado dentro del Sistema de Gestión de Parqueadero. Este sistema permite a los empleados gestionar su perfil de usuario, incluyendo la edición de información personal y el cambio de contraseña. Además, brinda herramientas para la Gestión de Clientes, permitiendo la administración de clientes por hora y clientes mensuales, con opciones para crearlos, editarlos y eliminarlos. A través de este diagrama, se visualiza de manera clara la interacción del usuario con el sistema y las acciones que puede realizar en su rol.

Figura 4. Diagrama de casos de uso para el empleado



Fuente: Propia del autor

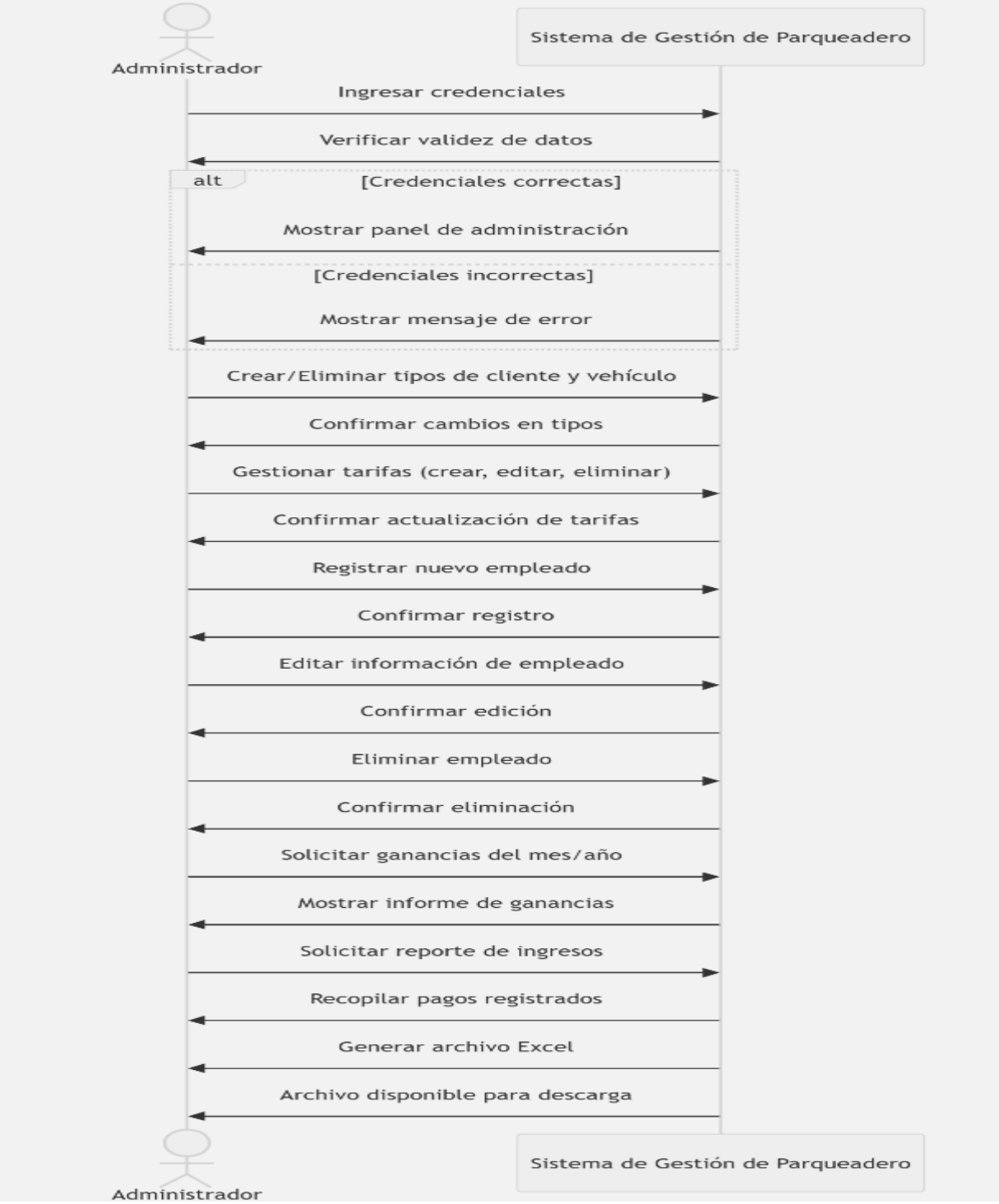
4.4.2 Diagrama De Secuencia

Configuración de parqueadero, Gestión de Empleados, Analíticas y Reportes

El siguiente diagrama representa el proceso de configuración, gestión de empleados, analíticas y reportes dentro del Sistema de Gestión de Parqueadero, mostrando la interacción entre el Administrador y el sistema. El flujo comienza con el inicio de sesión, donde el administrador ingresa sus credenciales y el sistema valida su autenticidad, permitiendo el acceso al panel de administración si son correctas o mostrando un mensaje de error en caso contrario. Una vez dentro del sistema, el administrador puede gestionar distintos aspectos, como la creación y eliminación de

tipos de cliente y vehículo, la administración de tarifas mediante su creación, edición o eliminación, y la actualización de estas configuraciones en el sistema. También puede registrar, modificar y eliminar empleados, asegurando que cada acción se refleje correctamente. Además, el sistema permite la solicitud de reportes financieros, incluyendo informes de ganancias y reportes de ingresos, los cuales pueden ser generados en formato Excel.

Figura 5. Diagrama de secuencias de configuración, gestión de empleados, analíticas y reportes

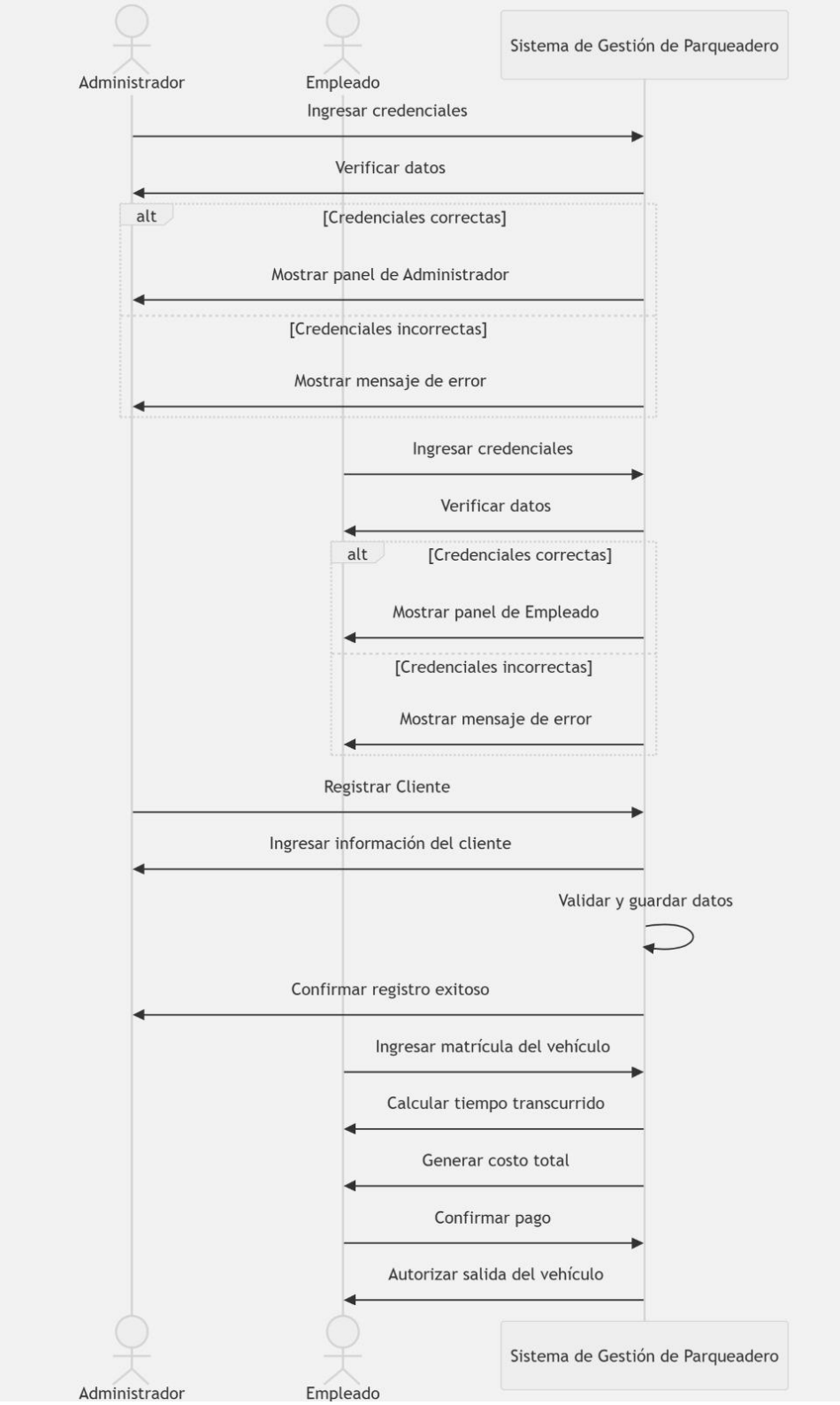


Fuente: Propia del autor

Gestión de Clientes

El siguiente diagrama representa el proceso de autenticación y gestión de clientes dentro del Sistema de Gestión de Parqueadero, detallando la interacción entre el Administrador, el Empleado y el sistema. El flujo comienza cuando el administrador o el empleado ingresan sus credenciales, lo que activa la verificación de datos por parte del sistema. Si las credenciales son correctas, el sistema muestra el panel correspondiente según el rol del usuario; de lo contrario, se genera un mensaje de error. Una vez dentro del sistema, el usuario puede registrar nuevos clientes ingresando su información, la cual es validada y almacenada por el sistema. Luego, cuando un cliente por hora necesita salir, el empleado ingresa la matrícula del vehículo, el sistema calcula el tiempo transcurrido y genera el costo total del servicio.

Figura 6. Diagrama de secuencias de gestión de clientes



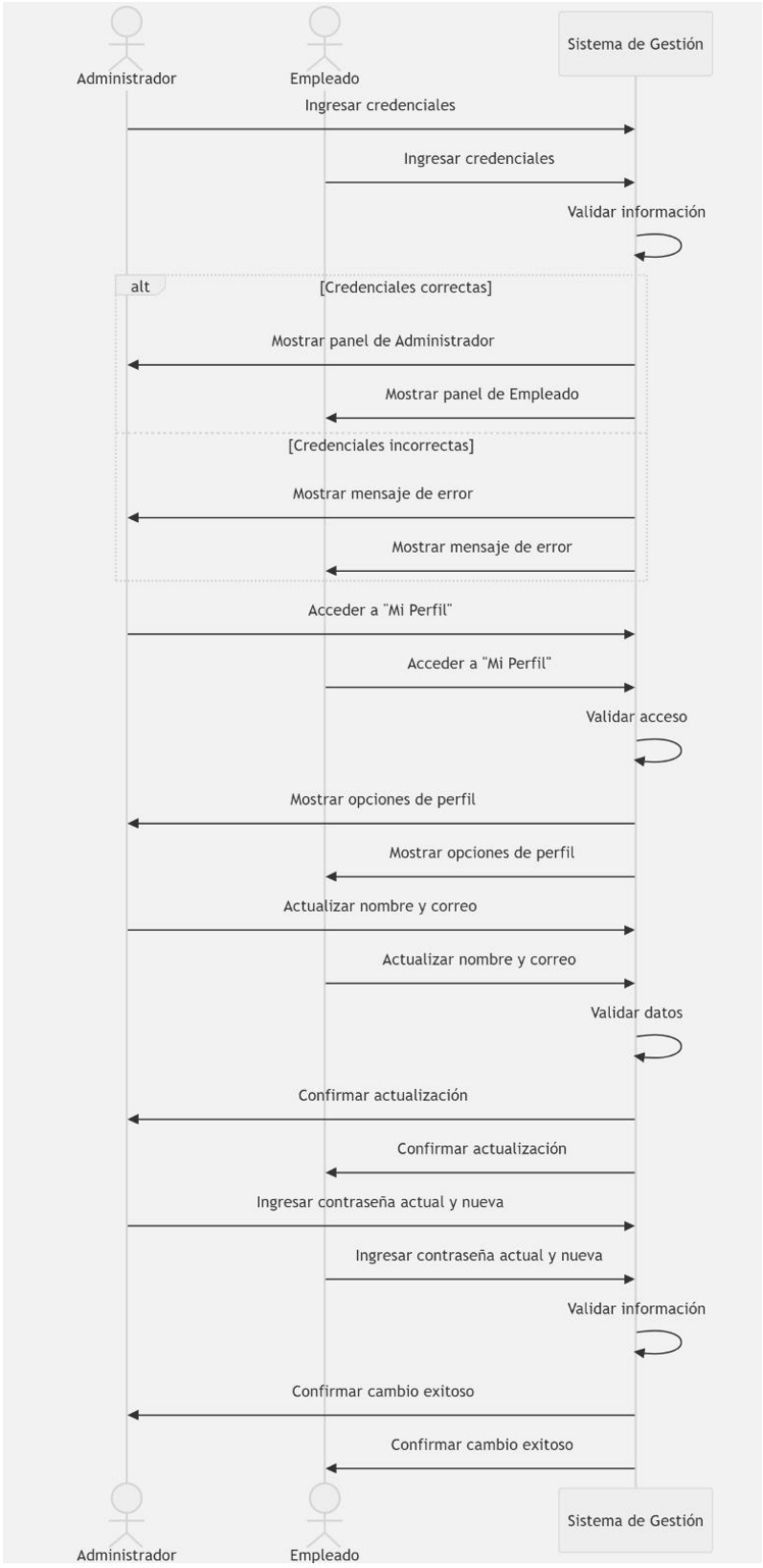
Fuente: Propia del autor

Perfil de Usuario

El siguiente diagrama incluyen interacciones relacionadas con la gestión del perfil, como acceder a la sección "Mi Perfil", para actualizar si información personal o de contacto, y cambiar la contraseña.

Cada acción pasa por una validación del sistema antes de confirmarse, asegurando que solo datos válidos sean almacenados. En caso de éxito, el sistema confirma la actualización de los datos o el cambio de contraseña.

Figura 7. Diagrama de secuencias de perfil de usuario



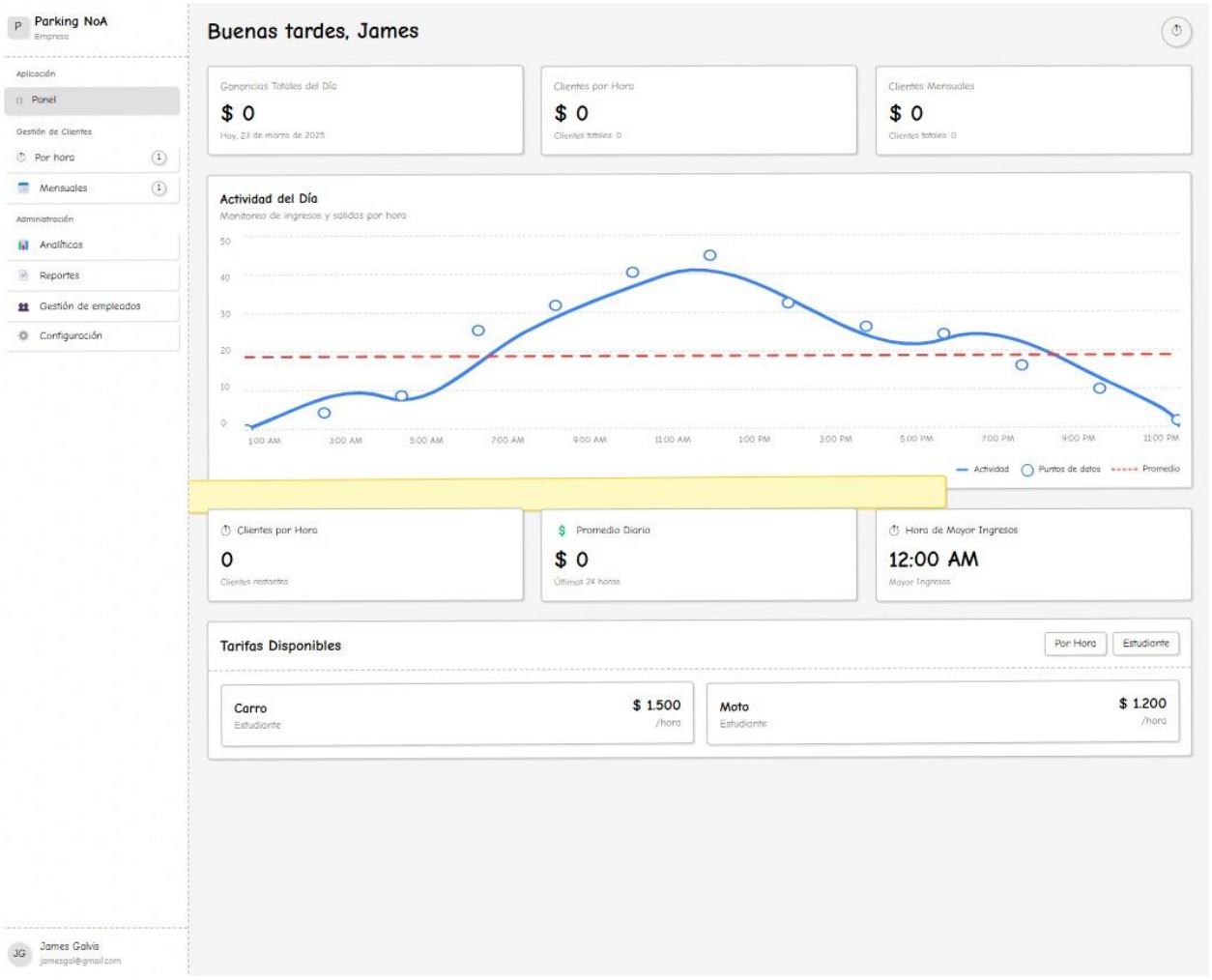
Fuente: Propia del autor

4.4.3 Mockups

El diseño de los mockups representó una etapa fundamental en el desarrollo del aplicativo, ya que permitió visualizar la interfaz de usuario y definir la disposición de los elementos funcionales antes de la implementación. Para esta tarea, se utilizó la herramienta V0, una plataforma de diseño que facilita la creación de prototipos interactivos y responsivos, permitiendo iterar rápidamente sobre las ideas iniciales y ajustarlas según las necesidades identificadas durante el levantamiento de requerimientos. V0 fue seleccionada por su capacidad para generar diseños adaptables a diferentes dispositivos, lo que aseguro que los mockups fueran representativos de la experiencia final del usuario.

El mockup de la Figura 8 muestra el panel de control principal de la aplicación. La interfaz incluye un saludo personalizado, métricas clave como ingresos totales del día y clientes mensuales, un gráfico de líneas para monitorear la actividad por hora, y una sección de tarifas disponibles con costos por tipo de vehículo. Es una herramienta clara y funcional para la administración diaria.

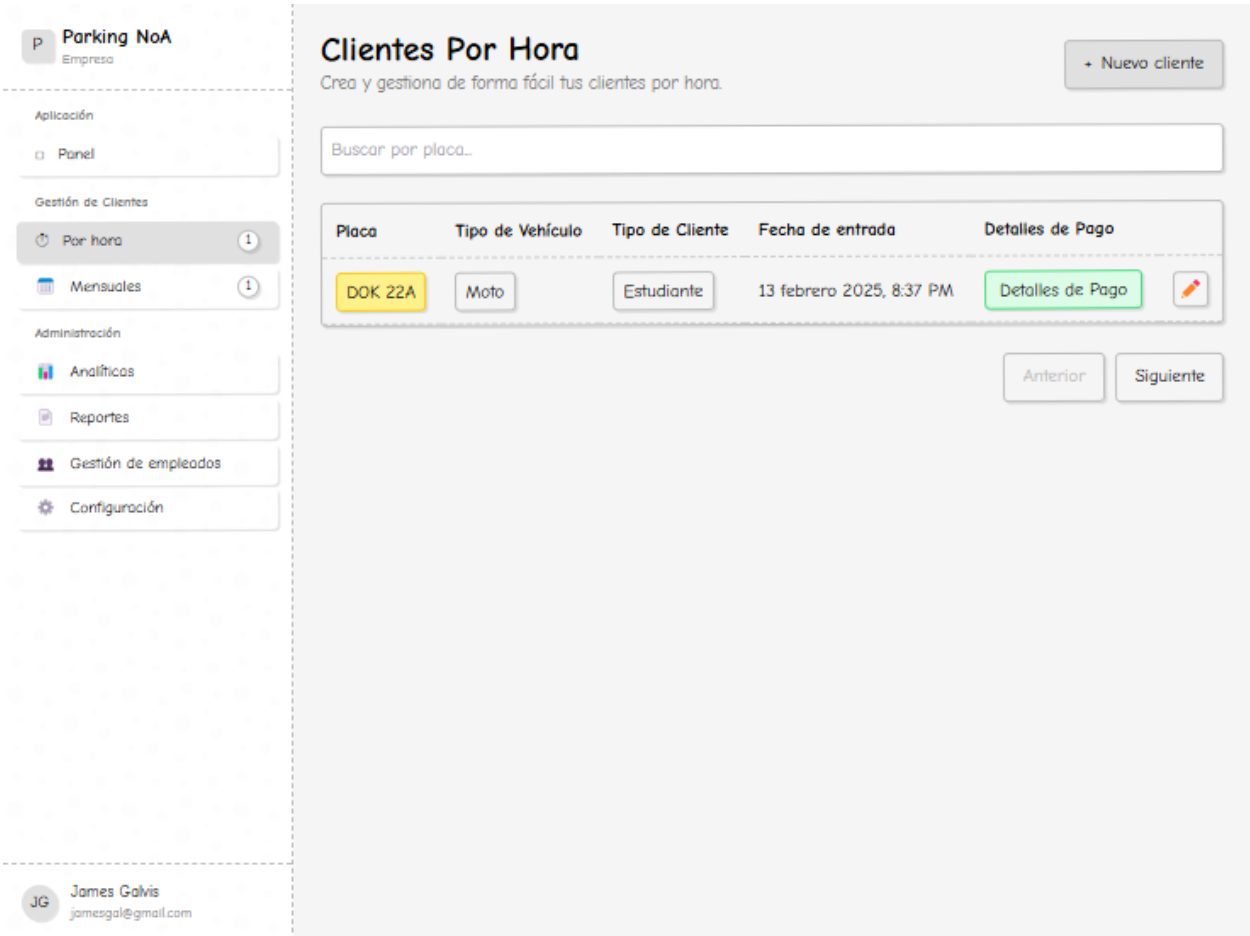
Figura 8. Panel de control



Fuente: Propia del autor

La Figura 9 presenta una pantalla para gestionar clientes por hora. Incluye un campo de búsqueda por placa, y una tabla con detalles como tipo de vehículo y fecha de entrada, así como botones para agregar nuevos clientes y ver detalles de pago.

Figura 9. Gestión de clientes por hora



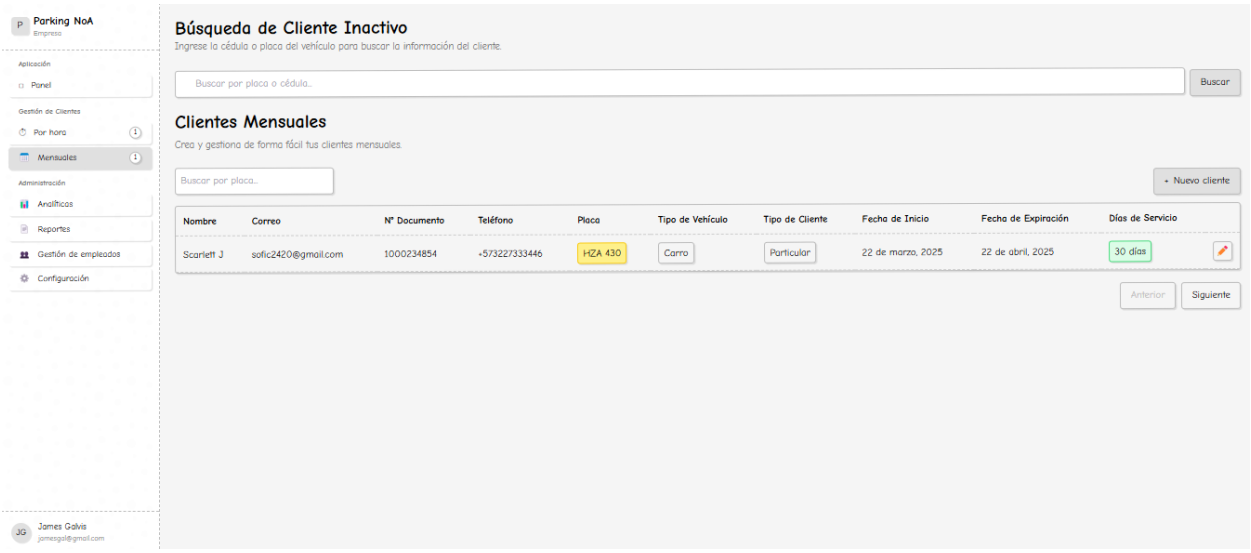
Fuente: Propia del autor

El siguiente mockup de la

Figura 10, muestra la página de gestión de clientes mensuales donde se podrán consultar clientes inactivos, así como consultar y ver a través de una tabla los clientes

que aun cuentan con la membresía activa, detallando incluso el tiempo antes de que su membresía expire.

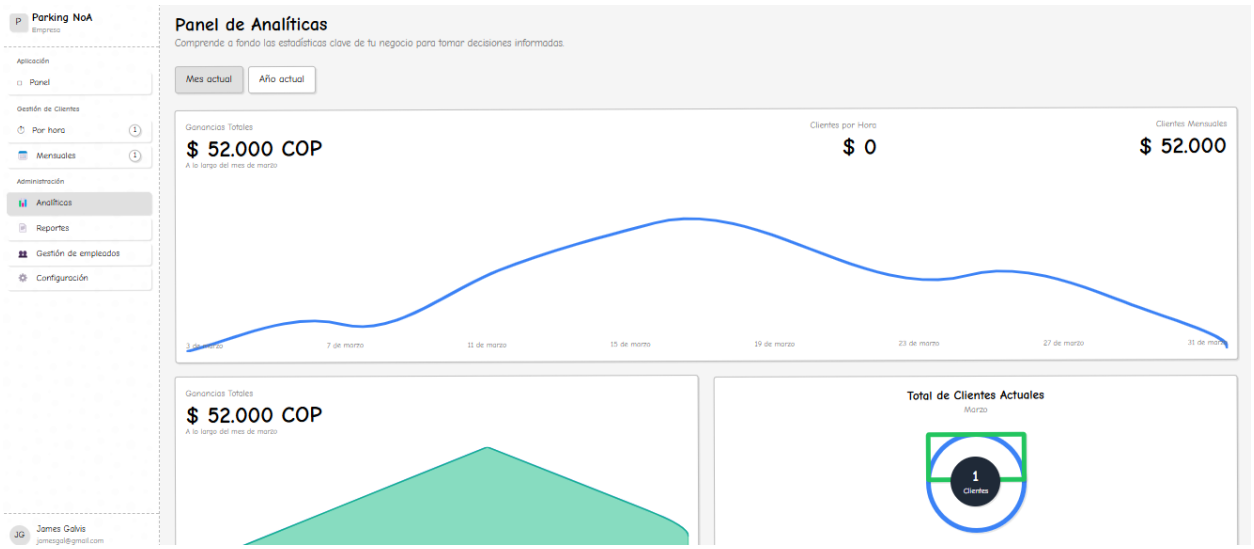
Figura 10. Gestión de clientes mensuales



Fuente: Propia del autor

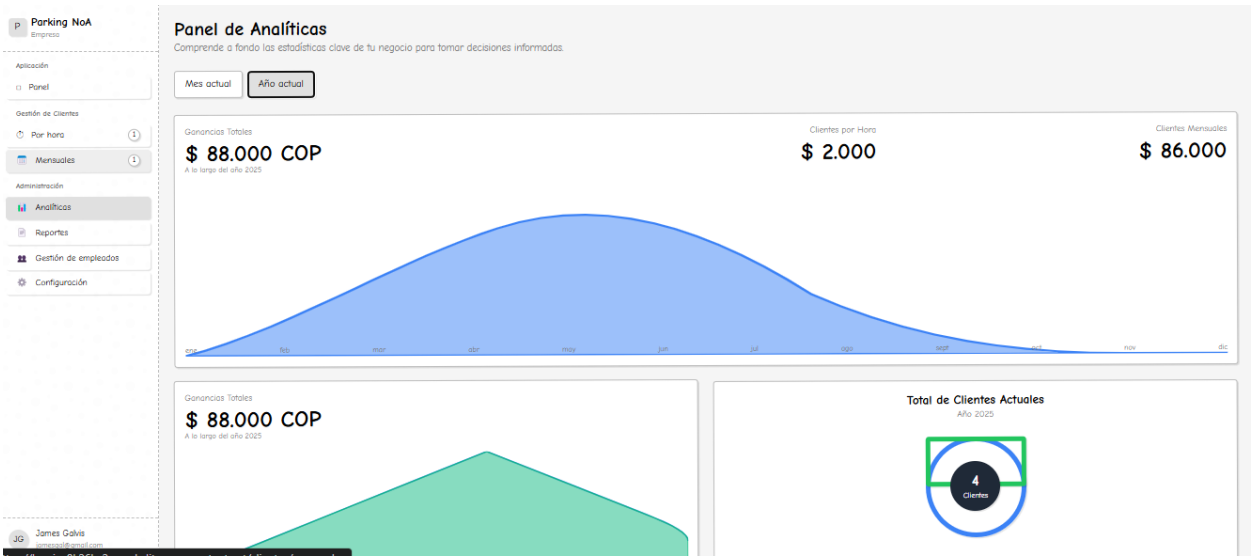
La Figura 11 y Figura 12 muestran los mockups para la sección de analíticas, donde el administrador puede visualizar las ganancias del año y mes actual, así como la cantidad de clientes que ha tenido en cada tiempo separados por tipo.

Figura 11. Página de analíticas para el mes actual



Fuente: Propia del autor

Figura 12. Página de analíticas para el año actual

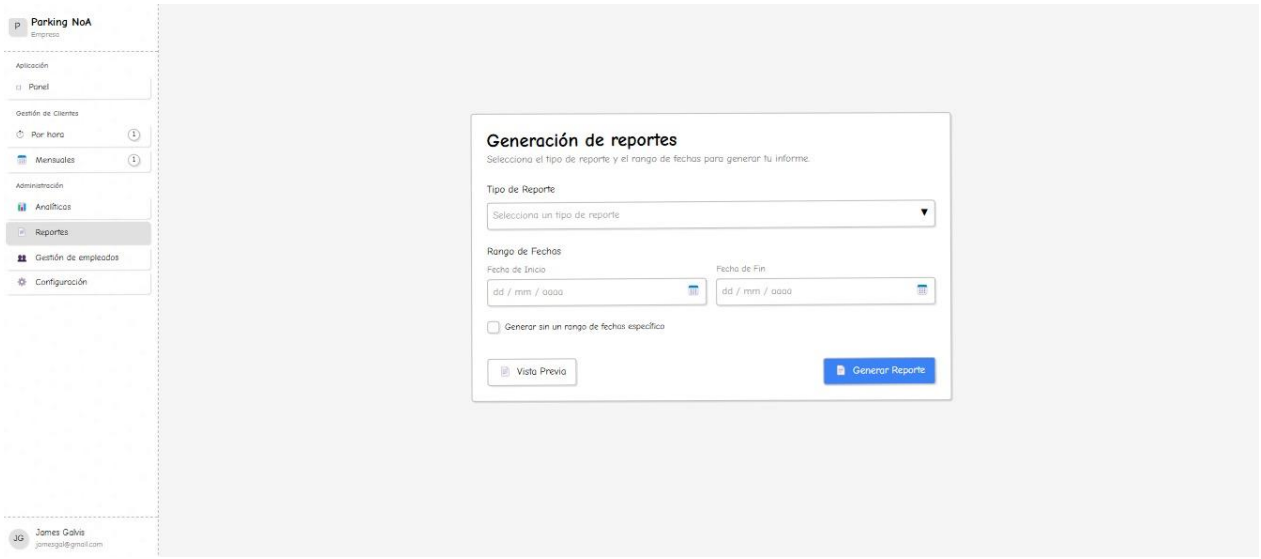


Fuente: Propia del autor

La

Figura 13 a continuación muestra la sección "Reportes", con un formulario para generar informes personalizados seleccionando tipo de reporte y rango de fechas. Con botones para previsualizar o generar el reporte.

Figura 13. Interfaz de generación de reportes

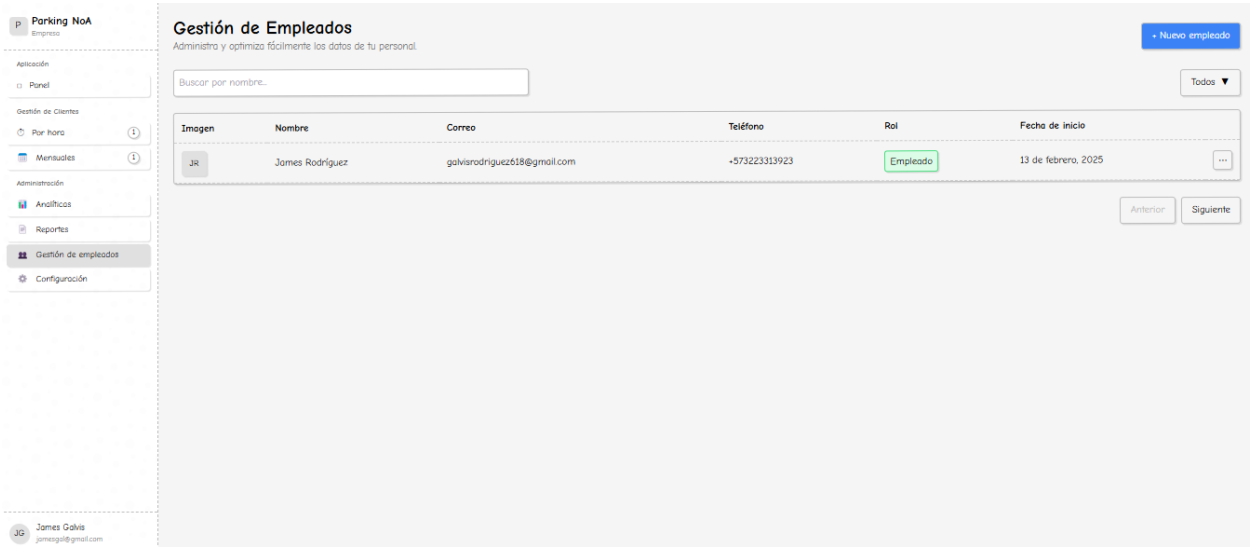


Fuente: Propia del autor

El mockup de la

Figura 14 presenta la sección “Gestión de Empleados”, desde la cual se detalla una tabla con los usuarios que actualmente tienen acceso al sistema, así como un input para buscar empleados por nombre, así como un botón para crear, editar o eliminar a un empleado en específico.

Figura 14. Interfaz gestión de empleados



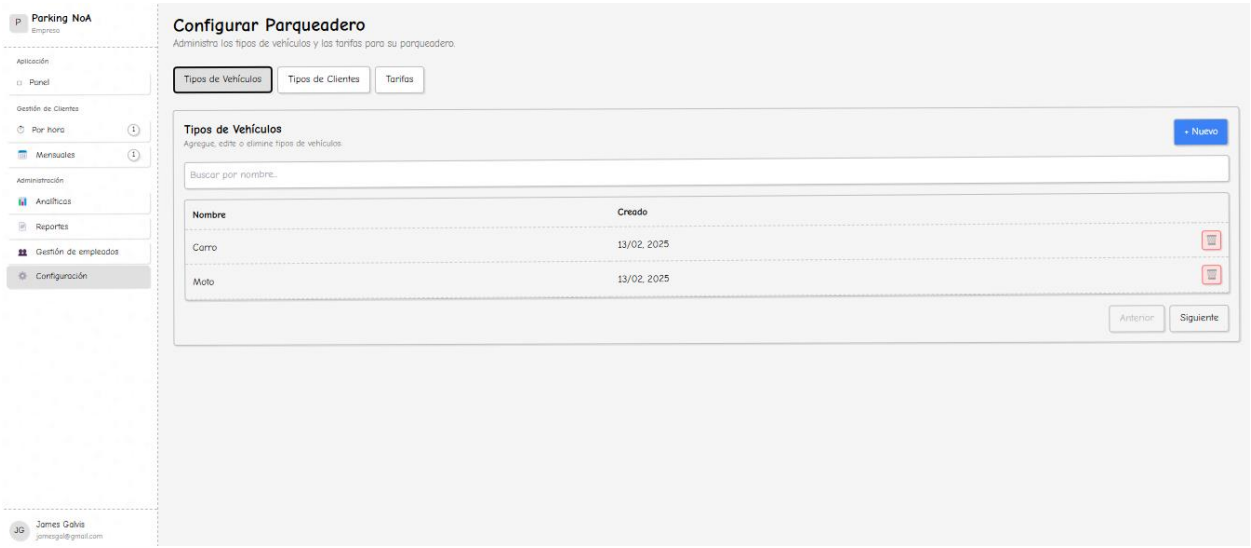
Fuente: Propia del autor

La sección de configuración, mostrada en la

Figura 15, **Figura 16** **y**

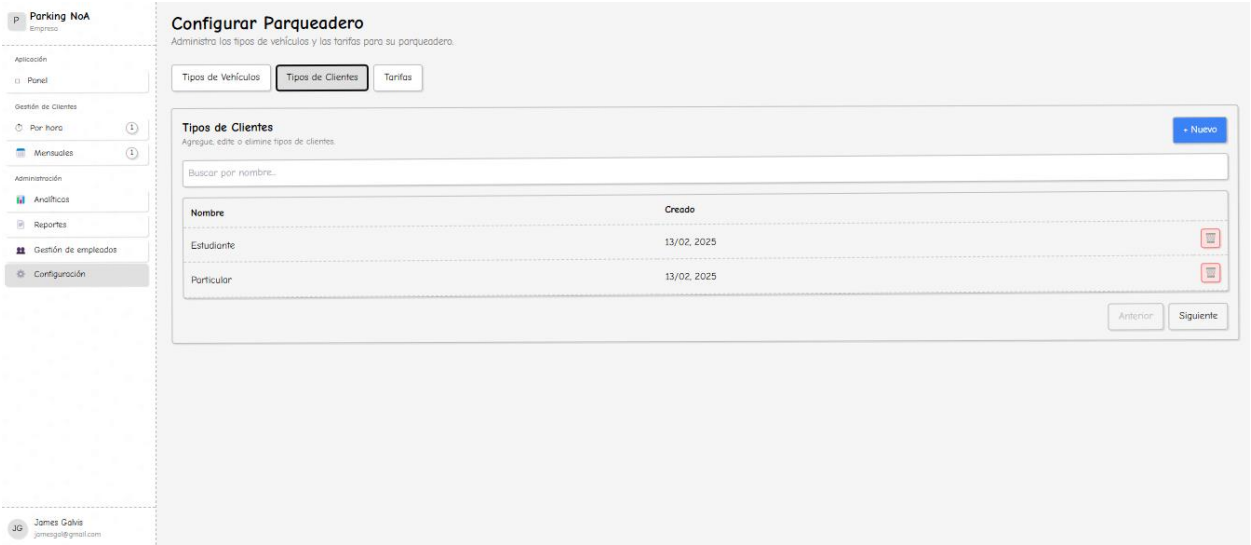
Figura 17, ofrece a los administradores herramientas esenciales para personalizar y gestionar el estacionamiento de forma eficiente. En la Figura 15, se configura los tipos de vehículos, permitiendo agregar o editar categorías como "Carro" o "Moto" y asociarles tarifas específicas. La Figura 16 aborda los tipos de clientes, como "Estudiante" o "Particular", facilitando la creación y modificación de estas categorías para adaptarse a distintos usuarios. Por su parte, la Figura 17 detalla la gestión de tarifas, donde se establecen costos por hora y mes según el tipo de vehículo y cliente, con una interfaz que incluye opciones de búsqueda y adición de nuevas tarifas.

Figura 15. Interfaz de configuración del parqueadero - sección de tipo de vehículos



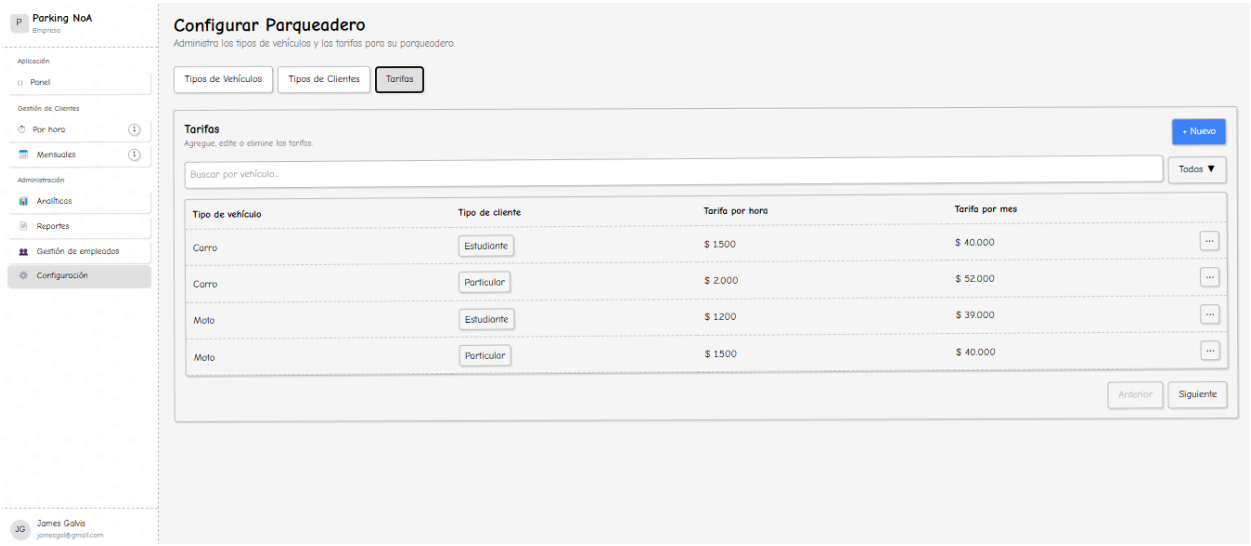
Fuente: Propia del autor

Figura 16. Interfaz de configuración del parqueadero - sección de tipo de cliente



Fuente: Propia del autor

Figura 17. Interfaz de configuración del parqueadero - sección de tarifas

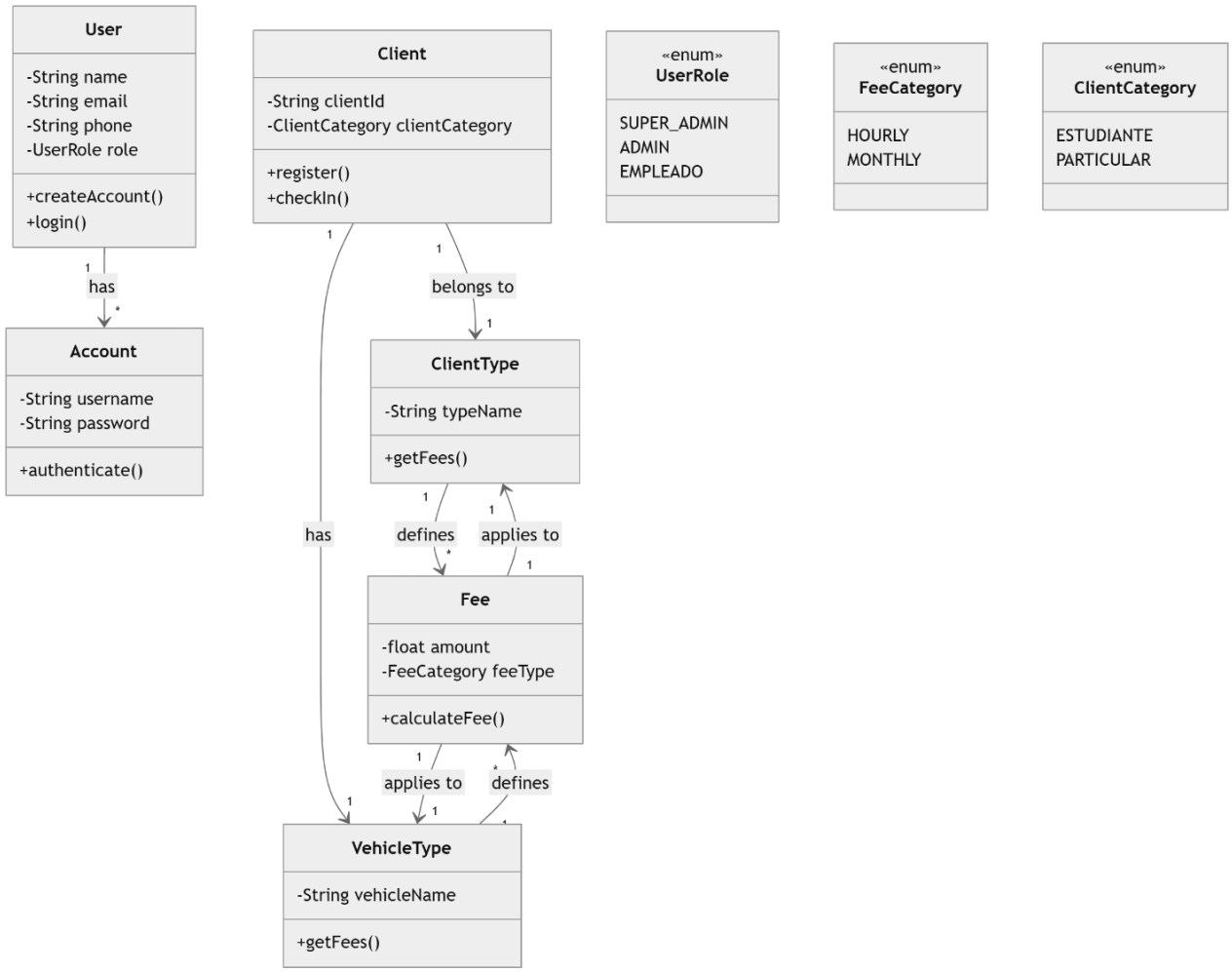


Fuente: Propia del autor

4.4.4 Diagrama De Clases

El siguiente diagrama representa la estructura del sistema y la relación entre sus componentes principales, incluyendo usuarios, clientes, tarifas y tipos de vehículos. Se evidencia cómo los distintos roles dentro del sistema, como SuperAdmin, Admin y Empleado, interactúan con las entidades y gestionan la información de los clientes y sus tarifas. A continuación, se detalla la organización y conexión entre estos elementos.

Figura 18. Diagrama de clases



Fuente: Propia del autor

4.5 Desarrollo Del Aplicativo

El aplicativo web se desarrolló utilizando un conjunto de tecnologías moderno, enfocado en maximizar la eficiencia, escalabilidad y usabilidad del sistema. Para alcanzar estos objetivos, se seleccionaron herramientas que permitieron integrar de manera fluida tanto el frontend como el backend, optimizando cada fase del desarrollo.

A continuación, se presentan las tecnologías empleadas y se describe cómo se aplicaron en cada etapa del proyecto:

Next.js con TypeScript

Se utilizó Next.js como framework principal debido a su capacidad para gestionar el lado cliente y servidor en un solo entorno, lo que optimiza el rendimiento y la organización del código. La integración de TypeScript añadió tipado estático, facilitando la detección temprana de errores y mejorando la mantenibilidad del código.

Figura 19. Configuración estricta de TypeScript para un proyecto Next.js con soporte para módulos modernos y alias de rutas.

```
{
  "compilerOptions": {
    "lib": ["dom", "dom.iterable", "esnext"],
    "allowJs": true,
    "skipLibCheck": true,
    "strict": true,
    "noEmit": true,
    "esModuleInterop": true,
    "module": "esnext",
    "moduleResolution": "bundler",
    "resolveJsonModule": true,
    "isolatedModules": true,
    "jsx": "preserve",
    "incremental": true,
    "plugins": [
      {
        "name": "next"
      }
    ],
    "paths": {
      "@/*": ["./*"]
    }
  },
  "include": ["next-env.d.ts", "**/*.ts", "**/*.tsx", ".next/types/**/*.ts"],
  "exclude": ["node_modules"]
}
```

Fuentes: Propia del autor

Server Actions de Next.js

Para simplificar la comunicación con la base de datos, se implementaron Server Actions, una funcionalidad nativa de Next.js que permite ejecutar operaciones del servidor de manera directa sin necesidad de controladores adicionales. Estas acciones se aplicaron en módulos críticos como la autenticación, la gestión de clientes, la configuración de tarifas y la generación de reportes.

Rutas API de Next.js

Se crearon rutas API específicas para manejar casos puntuales, tales como la ejecución de tareas programadas mediante cron jobs y otras operaciones que requerían interacción directa con el backend. Por ejemplo, una de estas rutas se encarga de identificar clientes mensuales próximos a expirar o ya expirados y de enviarles notificaciones por correo electrónico.

Figura 20. API para Notificación de Clientes con Servicios Mensuales Expirados

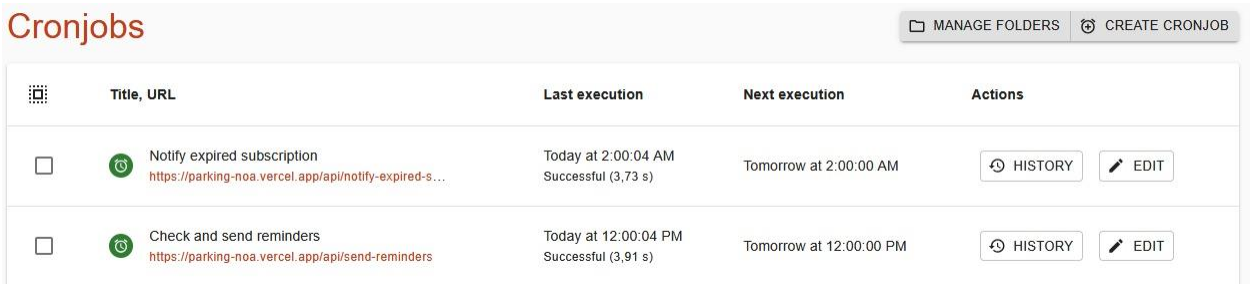
```
import { monthlyServiceExpiredEmail } from "@/lib/brevo";
import { db } from "@/lib/db";
import { DateTime } from "luxon";
import { NextRequest, NextResponse } from "next/server";

export async function GET(req: NextRequest) {
  const zone = "America/Bogota";
  const now = DateTime.now().setZone(zone);

  try {
    const clients = await db.client.findMany({
      where: {
        clientCategory: "MONTHLY",
        isActive: true,
        endDate: {
          lte: now.toJSDate(),
        },
      },
    });

    if (clients.length === 0) {
      return NextResponse.json(
        { message: "No hay clientes para notificar." },
        { status: 200 }
      );
    }

    // Procesar clientes en paralelo
    await Promise.all(
      clients.map(async (client) => {
        if (!client.email) return;

        await monthlyServiceExpiredEmail(
          client.email,
          client.name!,
          client.endDate!
        );

        await db.client.update({
          where: { id: client.id },
          data: { isActive: false },
        });
      })
    );

    return NextResponse.json(
      { message: "Notificaciones enviadas exitosamente." },
```

Fuente: Propia del autor

Cron Jobs

Para automatizar tareas esenciales, se implementaron cron jobs que realizan llamadas periódicas a las rutas API del sistema. Estas tareas automatizadas identifican clientes mensuales con servicios próximos a expirar (cinco días antes) o ya expirados, y envían notificaciones por correo electrónico para informarles sobre la necesidad de renovación.

Figura 21. Programación y ejecución de Tareas automatizadas (Cron Jobs)



Fuente: Propia del autor

CockroachDB

Como servicio de base de datos en la nube, CockroachDB fue elegido por su alta disponibilidad, escalabilidad y tolerancia a fallos. Se diseñó un esquema de datos robusto que soporta todas las funcionalidades del sistema, desde la gestión de usuarios y clientes hasta el registro de transacciones.

Auth.js

Para garantizar un proceso de autenticación seguro, se empleó Auth.js, que facilitó la implementación de funcionalidades como el registro, inicio y cierre de sesión.

Asimismo, se integraron medidas de protección de rutas basadas en roles, asegurando que cada usuario acceda únicamente a las funcionalidades autorizadas.

Figura 22. Configuración de Proveedor de Credenciales en Auth.js

```
Generate
generator client {
  provider = "prisma-client-js"
}

datasource db {
  provider = "cockroachdb"
  url      = env("DATABASE_URL")
}

enum UserRole {
  SuperAdmin
  Admin
  Empleado
}

enum FeeCategory {
  HOURLY
  MONTHLY
}

enum ClientCategory {
  HOURLY
  MONTHLY
}

model User {
  id            String    @id @default(cuid())
  name          String?
  email         String?   @unique
  emailVerified DateTime?
  password      String?
  phone         String?
  image         String?
  role          UserRole  @default(Empleado)

  accounts Account[]

  createdAt DateTime @default(now())
  updatedAt DateTime @updatedAt
}
```

Fuente: Propia del autor

Prisma como ORM

Prisma se utilizó para interactuar eficientemente con la base de datos, simplificando la gestión de modelos de datos, consultas y migraciones. Esta herramienta permitió una integración fluida con CockroachDB, optimizando las operaciones de lectura y escritura.

Figura 23. Ejemplo de definición de modelos y esquema de base de datos con Prisma

```
Generate
generator client {
  provider = "prisma-client-js"
}

datasource db {
  provider = "cockroachdb"
  url      = env("DATABASE_URL")
}

enum UserRole {
  SuperAdmin
  Admin
  Empleado
}

enum FeeCategory {
  HOURLY
  MONTHLY
}

enum ClientCategory {
  HOURLY
  MONTHLY
}

model User {
  id            String    @id @default(cuid())
  name          String?
  email         String?   @unique
  emailVerified DateTime?
  password      String?
  phone         String?
  image         String?
  role          UserRole  @default(Empleado)

  accounts Account[]

  createdAt DateTime @default(now())
  updatedAt DateTime @updatedAt
}
```

Fuente: Propia del autor

ShadCN como Biblioteca de Componentes

Se empleó ShadCN para construir componentes reutilizables (formularios, tablas, botones y gráficos), lo que contribuyó a mantener la coherencia en el diseño y a acelerar el desarrollo de la interfaz de usuario.

Tailwind CSS

Finalmente, Tailwind CSS se utilizó para estilizar la aplicación, logrando una interfaz visualmente atractiva y completamente responsive. Su sistema de clases utilitarias facilitó la creación de diseños adaptables para dispositivos móviles, tabletas y pantallas de escritorio.

Figura 24. Configuración personalizada de Tailwind CSS

```
import type { Config } from "tailwindcss";

const config: Config = {
    darkMode: ["class"],
    content: [
    "./pages/**/*.{js,ts,jsx,tsx,mdx}",
    "./components/**/*.{js,ts,jsx,tsx,mdx}",
    "./app/**/*.{js,ts,jsx,tsx,mdx}",
  ],
  theme: {
    extend: {
      screens: {
        'xs': '400px',
        'ss': '500px',
        'sm': '640px',
        'md': '768px',
        'md-plus': '960px',
        'lg': '1024px',
        'xl': '1280px',
        '2xl': '1536px',
      },
      backgroundImage: {
        'gradient-radial': 'radial-gradient(var(--tw-gradient-stops))',
        'gradient-conic': 'conic-gradient(from 180deg at 50% 50%, var(--tw-gradient-stops))'
      },
      borderRadius: {
        lg: 'var(--radius)',
        md: 'calc(var(--radius) - 2px)',
        sm: 'calc(var(--radius) - 4px)'
      },
      colors: {
        background: 'hsl(var(--background))',
        foreground: 'hsl(var(--foreground))',
        card: {
          DEFAULT: 'hsl(var(--card))',
          foreground: 'hsl(var(--card-foreground))'
        },
```

Fuente: Propia del autor

Esta integración de tecnologías permitió desarrollar un sistema robusto y escalable, capaz de responder a las necesidades específicas y ofrecer una experiencia de usuario fluida y segura.

Implementación

El robusto stack tecnológico descrito anteriormente sirvió como base para estructurar y desarrollar cada uno de los módulos del sistema de forma modular y coherente. Gracias a la integración de Next.js con TypeScript, Server Actions, rutas API, cron jobs, CockroachDB, Auth.js, Prisma, ShadCN y Tailwind CSS, se adoptó un

enfoque organizado que permitió implementar cada funcionalidad de manera independiente pero interconectada. De esta forma, se garantizó que aspectos críticos como la autenticación, la comunicación con la base de datos, la automatización de tareas y la interfaz de usuario se desarrollaran de forma eficiente y sin contratiempos. A continuación, se describe en detalle cómo se integraron estas tecnologías en los módulos principales del aplicativo:

Autenticación y Autorización:

Se implementó utilizando Auth.js con integración directa con Prisma para gestionar usuarios y roles. El flujo incluye registro, inicio y cierre de sesión, y protección de rutas según los permisos asignados.

Comunicación con la Base de Datos:

Las operaciones del sistema, como la creación de registros, consultas y actualizaciones se manejaron a través de Prisma con CockroachDB. Las Server Actions de Next.js permitieron simplificar la lógica del backend.

Tareas Automatizadas:

Las notificaciones por correo electrónico a los clientes se manejaron mediante cron jobs que llamaban a una ruta API de Next.js. Esta ruta consultaba los registros en la base de datos para determinar qué clientes debían recibir notificaciones y enviaba los correos de manera automática.

Interfaz de Usuario:

La interfaz fue desarrollada con componentes de ShadCN estilizados con Tailwind CSS. Se diseñaron vistas específicas para cada módulo (por ejemplo, clientes por hora, clientes mensuales, reportes, analíticas) y se incluyó soporte para modos oscuro y claro.

Responsividad:

Se aseguró que todo el diseño fuera completamente responsive mediante el uso de Tailwind CSS, garantizando una experiencia de usuario óptima en dispositivos de escritorio, móviles y tabletas.

Estructura del Repositorio

La estructura del proyecto fue organizada para maximizar el modularidad, facilitar la mantenibilidad y asegurar la claridad en el desarrollo. A continuación, se detalla cada uno de los directorios y archivos principales:

- **actions/**:

  Contiene las acciones del servidor utilizadas para manejar operaciones críticas, como la comunicación con la base de datos y la ejecución de lógica específica para los módulos.

- **app/**:

  Contiene las páginas y rutas principales del proyecto, siguiendo el enfoque de

rutas basado en carpetas de Next.js. Aquí se define la estructura de navegación de la aplicación.

- **components/**:

  Incluye los componentes reutilizables de la interfaz de usuario, como botones, tablas, formularios y gráficos. Fue desarrollado utilizando **ShadCN** para asegurar consistencia en el diseño.

- **constants/**:

  Almacena constantes globales utilizadas en diferentes partes del proyecto, como configuraciones de la aplicación, valores predeterminados y textos estáticos.

- **hooks/**:

  Contiene hooks personalizados de React para manejar lógica específica que puede ser reutilizada en diferentes componentes o módulos.

- **lib/**:

  Biblioteca con funciones utilitarias y herramientas comunes utilizadas en el proyecto, como helpers para validaciones y transformaciones de datos.

- **prisma/**:

  Contiene los archivos de configuración y modelos de Prisma ORM, utilizados para definir y gestionar el esquema de la base de datos en CockroachDB.

- **public/**:

  Almacena recursos estáticos como imágenes, íconos y otros archivos que son servidos directamente al cliente.

- **schemas/**:

  Define los esquemas y validaciones de datos utilizando herramientas como Zod para asegurar integridad en las operaciones y en la comunicación entre el frontend y el backend.

- **types/**:

  Contiene definiciones de tipos e interfaces para **TypeScript**, asegurando tipado consistente a lo largo de todo el proyecto.

- **utils/**:

  Incluye funciones de utilidad generales, como manejo de fechas, cálculos y formateo de datos.

Archivos Principales

- **.env:**

  Archivo de configuración que almacena variables de entorno sensibles, como claves API, credenciales y configuraciones específicas para entornos de desarrollo y producción.

- **auth.config.ts / auth.ts:**

  Archivos relacionados con la configuración y lógica de Auth.js, utilizados para manejar autenticación y autorización.

- **middleware.ts:**

Configura middlewares globales para manejar redirecciones, validaciones y protección de rutas.

- **next.config.mjs:**

  Configuración del framework Next.js, ajustando opciones como rutas personalizadas, integraciones y optimizaciones.

- **package.json:**

  Define las dependencias del proyecto, scripts disponibles y metadatos generales del repositorio.

- **prisma/schema.prisma:**

  Archivo principal de Prisma ORM que define los modelos y relaciones de la base de datos.

4.6 Diccionario De Datos

El diccionario de datos describe la estructura y organización de la base de datos relacional diseñada para el sistema, garantizando la claridad y consistencia en el manejo de la información. La base de datos, desarrollada con Prisma y alojada en CockroachDB, se compone de modelos principales como User, Client, ClientType, VehicleType y Fee, los cuales se interrelacionan para gestionar roles de usuario, tipos de clientes, categorías de vehículos y tarifas. Los modelos utilizan enumeraciones (enums) como UserRole, ClientCategory y FeeCategory para clasificar datos críticos

(p.ej., roles administrativos o modalidades de tarifas), mientras que las relaciones definidas (mediante claves foráneas y restricciones Cascade) aseguran la integridad referencial. A continuación, se detallan los campos, tipos de datos, restricciones y descripciones de cada entidad, proporcionando una guía técnica para el entendimiento y mantenimiento del esquema de datos. Este diccionario es fundamental para validar la coherencia del diseño y facilitar futuras actualizaciones del sistema.

Diccionario de Datos

Tabla 20. Enumeraciones (enums)

Nombre	Valores	Descripción
UserRole	SuperAdmin, Admin, Empleado	Define los roles de acceso dentro del sistema.
FeeCategory	HOURLY, MONTHLY	Clasifica las tarifas según su modalidad por hora o mensual.
ClientCategory	HOURLY, MONTHLY	Indica si un cliente utiliza el servicio por horas o bajo suscripción mensual.

Fuente: Propia del autor

Modelos / Tablas

Usuario

Tabla 21. Modelo de usuario

Atributo	Tipo	Restricciones/Propiedades	Descripción
id	String	PK, default: cuid()	Identificador único del usuario generado automáticamente.
name	String	Opcional	Nombre del usuario.
email	String	Opcional, único	Correo electrónico del usuario.

emailVerified	DateTime	Opcional	Fecha de verificación del correo.
password	String	Opcional	Contraseña cifrada del usuario.
phone	String	Opcional	Teléfono de contacto.
image	String	Opcional	URL de la imagen de perfil.
role	UserRole	Enum, default: Empleado	Rol del usuario: SuperAdmin, Admin o Empleado.
createdAt	DateTime	detault: now()	Fecha de creación del registro.
updatedAt	DateTime	Auto actualiza con @updatedAt	Fecha de última actualización del registro.

Fuente: Propia del autor

Relaciones:

- **accounts:** Relación uno a muchos con el modelo Account**.**

Cuenta / Account

Tabla 22. Modelo de cuenta (account)

Atributo	**Tipo**	**Restricciones/Propiedades**	**Descripción**
userId	String	FK (referencia a User)	Identificador del usuario asociado.
type	String	Requerido	Tipo de cuenta (e.g. credenciales, social).
provider	String	Requerido	Proveedor de autenticación (por ejemplo, Google o GitHub).
providerAccountId	String	Requerido	Identificador único asignado por el proveedor.

refresh_token	String	Opcional	Token para renovación de la sesión.
access_token	String	Opcional	Token de acceso.
expires_at	Int	Opcional	Tiempo de expiración del token.
token_type, scope, id_token, sesión_state	Varios	Opcionales	Información adicional para la gestión de la sesión.
createdAt	DateTime	detault: now()	Fecha de creación del registro.
updatedAt	DateTime	Auto actualiza con @updatedAt	Fecha de última actualización del registro.

Fuente: Propia del autor

Cliente / Client

Tabla 23. Modelo de cliente

Atributo	**Tipo**	**Restricciones/Propiedades**	**Descripción**
id	String	PK, default: cuid()	Identificador único del usuario generado automáticamente.
name	String	Opcional	Nombre del cliente (usado principalmente para clientes mensuales).
document	String	Opcional, único	Número del documento de identidad (clientes mensuales).
email	String	Opcional	Correo electrónico del usuario (clientes mensuales).

plate	String	Requerido	Matrícula del vehículo.
phone	String	Opcional	Teléfono de contacto (clientes mensuales).
monthsReserved	Int	Opcional	Numero de meses reservados (clientes mensuales).
reminderSent	Boolean	Default: false	Indica si se ha enviado el recordatorio de renovación (clientes mensuales).
clientTypeId	String	FK (referencia a ClientType)	Identificador del tipo de cliente.
vehicleTypeId	String	FK (referencia a VehicleType)	Identificador del tipo de vehículo.
clientCategory	ClientCategory	Enum (HOURLY, MONTHLY)	Categoría del cliente.
totalPaid	Float	Default: 0.0	Total pagado por el cliente.
isActive	Boolean	Default: true	Estado activo del cliente.
entryDate	DateTime	Opcional	Fecha de entrada (aplica para clientes por hora).
exitDate	DateTime	Opcional	Fecha de salida (aplica para clientes por hora).
startDate	DateTime	Opcional	Fecha de inicio del servicio (aplica para clientes mensuales).
endDate	DateTime	Opcional	Fecha de finalización del servicio (clientes mensuales).

createdAt	DateTime	detault: now()	Fecha de creación del registro.
updatedAt	DateTime	Auto actualiza con @updatedAt	Fecha de última actualización del registro.

Fuente: Propia del autor

Relaciones:

- **clientType:** Relación muchos a uno con **ClientType** (clientTypeId como clave foránea)
- **vehicleType:** Relación muchos a uno con **VehicleType** (vehicleTypeId como clave foránea)

Tipo de Cliente / ClientType

Tabla 24. Modelo de tipo de cliente

Atributo	**Tipo**	**Restricciones/Propiedades**	**Descripción**
id	String	PK, default: cuid()	Identificador único del tipo de cliente.
name	String	Requerido	Nombre del tipo de cliente (ej. “Estudiante”, “Particular”, etc.).
createdAt	DateTime	detault: now()	Fecha de creación del registro.
updatedAt	DateTime	Auto actualiza con @updatedAt	Fecha de última actualización del registro.

Fuente: Propia del autor

Relaciones:

- **clients:** Relación uno a muchos con Client.
- **fee:** relación uno a muchos con Fee.

Tipo de vehículo / VehicleType

Tabla 25. Modelo de tipo de vehículo

Atributo	Tipo	Restricciones/Propiedades	Descripción
id	String	PK, default: cuid()	Identificador único del tipo de vehículo.
name	String	Requerido	Nombre del tipo de vehículo (ej. “Moto”, “Carro”, etc.).
createdAt	DateTime	detault: now()	Fecha de creación del registro.
updatedAt	DateTime	Auto actualiza con @updatedAt	Fecha de última actualización del registro.

Fuente: Propia del autor

Relaciones:

- **clients:** relación uno a muchos con Client.
- **fee:** Relación uno a muchos con Fee.

Tarifa / Fee

Tabla 26. Modelo de tarifa

Atributo	Tipo	Restricciones/Propiedades	Descripción
id	String	PK, default: cuid()	Identificador único de la tarifa.
clientTypeId	String	FK (referencia a ClientType)	Referencia al tipo de cliente al que aplica la tarifa.

vehicleTypeId	String	FK (referencia a VehicleType)	Referencia al tipo de vehículo al que aplica la tarifa.
feeType	FeeCategory	Enum (HOURLY, MONTHLY)	Categoría de la tarifa (por hora o mensual).
Price	Float	Requerido	Precio de la tarifa.
createdAt	DateTime	detault: now()	Fecha de creación del registro.
updatedAt	DateTime	Auto actualiza con @updatedAt	Fecha de última actualización del registro.

Fuente: Propia del autor

Relaciones:

- **clientType:** Relación muchos a uno con ClientType (**clientTypeId** como clave foránea).
- **vehicleType:** Relación muchos a uno con VehicleType (**vehicleTypeId** como clave foránea).

4.7 Plan De Pruebas

Tabla 27. Plan de pruebas

Módulo de Prueba	Objetivo de la Prueba	Resultado Esperados	Resultados Obtenidos
Registro de Usuario	Validar que pueda registrar al dueño del parqueadero como administrador del sistema.	Los datos deben almacenarse correctamente, verificando duplicados, validaciones de campos y hacer el	El registro funcionó correctamente con validaciones aplicadas y el inicio de sesión automático.

		Login automáticamente después de que el proceso sea exitoso.	
Inicio de Sesión	Verificar que el sistema permita iniciar sesión con credenciales validas.	Los usuarios deben poder acceder al sistema solo con credenciales correctas y mostrar mensajes de error si son incorrectas.	El inicio de sesión funcionó correctamente, validando las credenciales del usuario.
Cerrar Sesión	Verificar que los usuarios puedan cerrar sesión de forma segura.	El sistema debe cerrar la sesión actual y redirigir al usuario a la pantalla de inicio de sesión.	Funcionó correctamente, redirigiendo al usuario al formulario de inicio de sesión.
Configuración	Comprobar la creación y eliminación de tipos de clientes, tipos de vehículos, así como la creación, edición y eliminación de las tarifas.	Todos los procesos deben completarse correctamente, sin errores en la base de datos.	**Intento 1**: La creación de los tipos de clientes, tipos de vehículos y tarifas funcionaron correctamente. **Intento 2**: Se identifico un caso no controlado al eliminar un tipo de cliente o tipo de vehículo asociado a tarifas, el cual se corrigió manejando el caso desde el backend y mostrando un mensaje al

			usuario sobre la situación presentada. **Intento 3**: Edición de tarifas realizada correctamente.
Gestión de Empleados	Validar la creación, edición, eliminación de empleados y asignación de roles.	Los datos de los empleados deben actualizarse correctamente, y los permisos asignados deben respetarse.	Todos los procesos se realizaron sin problemas.
Clientes por Hora	Evaluar la funcionalidad de crear, eliminar, editar, calcular pagos y visualizar clientes por hora.	Los registros deben crearse, actualizarse o eliminarse correctamente en la base de datos, y los cálculos de pago deben ser precisos.	Procesos ejecutados correctamente en todos los intentos.
Clientes Mensuales	Verificar la creación, edición y visualización de clientes mensuales, incluyendo el tiempo de expiración del servicio. Además, debe ser exitoso el envío de correo electrónico al cliente sobre su pago de mensualidad, así	Los registros deben crearse, actualizarse o eliminarse correctamente en la base de datos, y el envío de los correos electrónicos se deben realizar correctamente y en las horas establecidas.	**Intento 1**: Procesos de creación, edición y eliminación ejecutados correctamente en todos los intentos. **Intento 2**: Se comprobó el envío del correo electrónico al cliente confirmando el pago de la mensualidad. **Intento 3**: La hora establecida con los Cron

	como comprobar el envío de recordatorio cuando su tiempo de servicio este próximo a expirar (al faltar 5 días antes de que se le acabe la mensualidad) y otro cuando se le termine completamente.		Jobs no era la correcta para notificar correctamente al cliente para el correo de tiempo de expiración cercano o alcanzado. Intento 4: El envío del correo para el caso de tiempo de expiración del servicio cercano y expirado se ejecutó de la forma esperada.
Analíticas	Reflejar que los gráficos reflejen correctamente las ganancias del mes y año actual.	Los datos estadísticos deben ser precisos y los gráficos claros e interactivos.	**Intento 1**: Los gráficos en modo de desarrollo mostraban la información de forma correcta y precisa. **Intento 2**: Los gráficos en el sistema desplegado mostraban datos poco precisos. **Intento 3**: Se identifico que la causa eran las fechas, así que se hizo uso de una librería para trabajar las fechas de forma precisa y estandarizadas a la zona horaria de Colombia.

Reportes	Comprobar la generación de informes de ganancias, clientes y próximas expiraciones.	Los reportes deben incluir los datos seleccionados, exportarse correctamente y respetar el rango de fechas si se establecen.	Reportes generados y exportados en archivos Excel sin errores.
Panel Principal	Validar la visualización de las estadísticas del día actual en el gráfico y la correcta visualización de la información de las tarifas del parqueadero.	Las estadísticas y gráficos deben mostrarse correctamente y actualizarse en tiempo real.	Panel funcional y datos actualizados correctamente con cada nuevo registro, eliminación o edición de los clientes ya sean mensuales o por hora.
Interfaz Responsive	Evaluar el diseño responsive en diferentes dispositivos y resoluciones.	La interfaz debe ajustarse correctamente a dispositivos móviles, tabletas y monitores de escritorio.	Diseño ajustado correctamente en todas las pruebas.
Modo Oscuro/Claro	Comprobar la funcionalidad de cambio entre modos oscuro y claro en toda la interfaz.	El sistema debe aplicar el modo seleccionado sin errores en toda la interfaz.	Modos aplicados correctamente en todo el sistema.

Compatibilidad	Comprobar que el sistema funciona correctamente en la mayoría de los navegadores.	El sistema debe funcionar correctamente en navegadores modernos como Chrome, Firefox y Edge.	Las pruebas realizadas en Chrome, Firefox y Edge salieron de forma exitosa.

Fuente: Propia del autor

5 CAPITULO V

5.1 Conclusiones

En conclusión, se desarrolló un aplicativo web para la gestión y control vehicular en el parqueadero Parking Noa. Se identificaron las necesidades funcionales del aplicativo recopilando y analizando información clave recolectada al entrevistar administradores y colaboradores del parqueadero.

Luego, se diseñó el modelo de la base de datos relacional usando Prisma para interactuar eficientemente con la base de datos, simplificando la gestión de modelos de datos, consultas y migraciones; y en la cual se estructuro la información de clientes, personal administrativo y configuración tarifaria

De igual manera, se diseñaron los mockups de la interfaz y se definieron los módulos funcionales del aplicativo web como el control principal (figura 8), la gestión de clientes (figura 9 y 10), gestión de empleados (figura 14), sección de analíticas (figura 11 y 12), generación de reportes (figura 13) y configuración (figura 15, 16 y 17).

Parking NoA constituye un aporte relevante tanto a nivel académico como práctico, demostrando la viabilidad y eficacia de la aplicación de tecnologías modernas en la gestión de parqueaderos. Este proyecto sienta las bases para futuras mejoras y ampliaciones, abriendo la posibilidad de explorar nuevas funcionalidades que sigan optimizando la operatividad y eficiencia en el sector.

5.2 Recomendaciones

Con base en los resultados y el análisis realizado durante el desarrollo del prototipo Parking NoA, se plantean las siguientes recomendaciones para futuras mejoras y ampliaciones del sistema:

1. **Continuar con la retroalimentación de usuarios:**

   Se recomienda implementar un mecanismo continuo de recolección de opiniones de los administradores y empleados, para identificar áreas de mejora en la interfaz y la funcionalidad del sistema, y así adaptar el aplicativo a las necesidades del parqueadero.

2. **Ampliar las medidas de seguridad:**

   Aunque se han implementado medidas básicas de autenticación y control de accesos, se sugiere evaluar la incorporación de mecanismos adicionales como la autenticación multifactor (MFA) y la monitorización de actividades sospechosas, con el fin de fortalecer la seguridad del sistema.

3. **Optimizar el módulo de reportes y analíticas:**

Se recomienda profundizar en la funcionalidad del módulo de reportes, integrando opciones avanzadas de filtrado y análisis predictivo. Esto permitirá a los administradores obtener insights más precisos sobre el desempeño del parqueadero y tomar decisiones estratégicas informadas.

4. **Actualizar y ampliar la documentación técnica:**

   Es fundamental mantener actualizada la documentación del proyecto, incluyendo el diccionario de datos, la estructura del repositorio y los manuales de usuario. Esto facilitará el mantenimiento y la escalabilidad del sistema, así como la incorporación de nuevos desarrolladores en el futuro.

5. **Realizar pruebas de rendimiento periódicas:**

   Se sugiere llevar a cabo pruebas de rendimiento y escalabilidad de forma regular, especialmente tras la incorporación de nuevas funcionalidades, para garantizar que el sistema continúe respondiendo a los requerimientos de disponibilidad y velocidad establecidos.

6. **Explorar integraciones con tecnologías emergentes:**

   Considerar la posibilidad de integrar herramientas de inteligencia artificial y análisis de datos en tiempo real, que puedan optimizar la predicción de picos de uso y mejorar la gestión de recursos del parqueadero.

Con esto no solo se busca consolidar el éxito del prototipo actual, sino también sentar las bases para futuras mejoras que permitan adaptar el sistema a los cambios en las necesidades operativas y tecnológicas del entorno.

5.3 Resumen Analítico Especializado – RAE

1.	Titulo	Desarrollo de un Aplicativo Web para la Gestión y Control Vehicular de un Parqueadero en la Ciudad de Villavicencio
2.	Autores	James Fernando Galvis Rodriguez
3.	Fecha	Febrero de 2025
4.	Palabras Claves	Prototipo, aplicación web, gestión de parqueadero
5.	Descripción	Trabajo de grado para optar al título de Tecnólogo Desarrollador de Software.
6.	Problema	¿Como desarrollar un Aplicativo Web para la Gestión y Control Vehicular del parqueadero Parking Noa en la ciudad de Villavicencio?
7.	Objetivo	Desarrollar un Aplicativo Web para la Gestión y Control Vehicular en el Parqueadero Parking Noa ubicado en la Ciudad de Villavicencio
8.	Conclusiones	Se desarrolló un aplicativo web para la gestión y control vehicular en el parqueadero Parking Noa. Se identificaron las necesidades funcionales del aplicativo recopilando y analizando información clave recolectada al entrevistar administradores y colaboradores del parqueadero, luego, se diseñó el modelo de la base de datos relacional usando Prisma. De igual manera, se diseñaron los mockups de la interfaz y se definieron los módulos funcionales del aplicativo web
9.	Autor RAE	James Fernando Galvis Rodríguez

10. Fecha creación de RAE	Febrero de 2025

6 Referencias

Aditya, A., Anwarul, S., Tanwar, R., & Vamsi, S. (2023). An IoT assisted Intelligent Parking System (IPS) for Smart Cities. *ScienceDirect*. doi:https://doi.org/10.1016/j.procs.2023.01.084

Anzola, E. G. (2024). APP prototipo para la solución a la búsqueda de parqueaderos en la ciudad de Bogotá Colombia con aporte al sistema de movilidad. (U. Ean, Ed.) Obtenido de http://hdl.handle.net/10882/14408

Beaty, S. (2024). *Web App Development: A Computer Science Approach.* Cognella Academic Publishing.

Freeman, A. (2023). *Pro ASP.NET Core 7.* Manning Publications.

Hernández, F., & López, J. (2020). Desarrollo de una aplicación web para la gestión de parqueaderos en la UNAM. *Revista de Ingeniería y Tecnología*, 120-133.

Hoffman, A. (2024). *Web Application Security.* O'Reilly Media.

Hooda, S., Sood, V. M., Singh, Y., Dalal, S., & Sood, M. (2023). *Agile Software Development: Trends, Challenges and Applications.* Wiley-Scrivener.

Lozada Quintero, D. S., & Martinez Quevedo, M. H. (2023). Desarrollo prototipo Hardware para Izzi Parking. *Universidad Cooperativa de Colombia*. Obtenido de https://hdl.handle.net/20.500.12494/51479

Osmani, A. (2023). *Learning JavaScript Design Patterns: A JavaScript and React Developer's Guide.* O'Reilly Media.

Pant, P., Singh Rajawat, A., Goyal, S. B., Bedi, P., Verma, C., Raboaca, M., & Magda Enescu, F. (2022). Authentication and Authorization in Modern Web Apps for Data Security Using Nodejs and Role of Dark Web. *ScienceDirect*. doi:10.1016/j.procs.2022.12.080

Pattiwal, S. (2024). *The Rules of UX Design.* Publicado Independientemente.

Petrosyan, A. (13 de Enero de 2025). *Internet usage worldwide - statistics & facts*. Obtenido de Statista: https://www.statista.com/topics/1145/internet-usage-worldwide/#topicOverview

Richards, M. (2017). *Software Architecture Patterns.* O'Reilly Media.

RUNT. (2024). *Boletín de Prensa 005 de 2024 El parque automotor de Colombia sigue creciendo: Casi 20 millones de vehículos registrados.* Obtenido de Registro Único Nacional de Transito: https://runt.gov.co/sites/default/files/Bolet%C3%ADn%20de%20Prensa%20005%20de%202024.pdf

Schwaber, K., & Sutherland, J. (2020). *The Scrum Guide: The definitive guide to Scrum: The rules of the game.* Obtenido de scrumguides: https://scrumguides.org/docs/scrumguide/v2020/2020-Scrum-Guide-US.pdf

SFMTA. (2022). *SFpark Evaluation*. Obtenido de San Francisco Municipal Transport Agency (SFMTA): https://www.sfmta.com/getting-around/drive-park/demand-responsive-pricing/sfpark-evaluation

Stair, R., Perew, M., & Vance, L. (2024). *Principles of Information Systems (MindTap Course List).* Cengage Learning.

Veloz, D., Guerrero, P., Peñafiel, E., & Salinas, E. (2024). Análisis de factibilidad para implementar parqueaderos inteligentes en la Ciudad de Riobamba que promuevan una movilidad sostenible. *Casa Editora del Polo*. doi:10.23857/pc.v9i8.7701

7 Anexos

Anexo 1. Transcrito de la entrevista verbal realizada al administrador del parqueadero parking NoA

A continuación, se presenta la entrevista realizada al administrador del parqueadero parking NoA en la ciudad de Villavicencio:

Buenas tardes, soy James Galvis, estudiante de desarrollo de software de la Universidad Minuto de Dios y me gustaría entrevistarlo para hacerle algunas preguntas respecto al parqueadero que usted administra, podría por favor hablarnos del negocio y decirnos que labor ejerce usted aquí (en el parqueadero)

Buenas tardes, soy el administrador del parqueadero Parking NoA, ubicado en la ciudad de Villavicencio. Llevamos aproximadamente siete años ofreciendo servicios de estacionamiento en esta parte de la ciudad, nuestros clientes más habituales son estudiantes y profesores de la UNIMINUTO ya que estamos cerca.

¿Como se encarga usted y los demás empleados de operar este parqueadero, es decir como registran los vehículos o como mantienen la lista de los clientes mensuales, por ejemplo?

Actualmente, utilizamos tickets físicos que nosotros mismos firmamos para registrar la entrada y salida de vehículos, y archivamos todo en cuadernos y Excel para gestionar a los clientes mensuales.

Como cliente regular, he notado que el parqueadero siempre se mantiene a su máxima capacidad, ¿Ha encontrado usted problemas manejando y administrando tanta información de clientes y vehículos que circulan a diario?

Si, claro, de vez en cuando se encuentra uno con errores en los tickets cuando los números de placa o la hora de entrada no cuadran, o con clientes que pierden los tickets, sobre todo en horas pico cuando hay mucho movimiento. También ha pasado que las cuentas en los cuadernos no cuadran con las que hacemos en la computadora (en Excel), y pues nos hace perder clientes y dinero.

Cuando hay problemas así, la salida de los vehículos se nos vuelve lenta porque tenemos que hacer los cálculos de hora y tarifa manualmente; se vuelve todo muy laborioso.

Hay un problema que me interesa y que usted no ha mencionado y es el tema de la seguridad. ¿Cómo se encarga usted de los registros no sean alterados por otros, o que directamente desaparezcan?

Eso ya sería una cuestión de confianza con los empleados, porque no hay mucho que se pueda hacer. Como le había dicho, los registros se mantienen en un cuaderno y en el computador, y si ambos se llegaran a perder, pues no hay nada que hacer.

Si usted pudiera tener una aplicación en su computador que le ayudara a administrar el parqueadero y le ayudara a resolver todos estos problemas, ¿Qué funciones cree usted que serían esenciales?

Pues lo primero seria que fuera fácil de usar por cualquiera de nuestros empleados. Me gustaría que los empleados aquí puedan registrar la placa y el tipo de vehículo cuando ingresa, y que, al salir, el sistema calcule automáticamente el tiempo y la tarifa. También sería bueno llevar el listado de los clientes mensuales, para hacer el seguimiento de los pagos y donde uno pueda estar verificando las fechas de vencimiento.

Yo no sé mucho de computadores y programas, pero como usted había dicho, que también se pudiera encargar de la seguridad de los datos, que solo algunos empleados puedan modificarla y que no vayan a desaparecer, así como así.

También sería bueno que la aplicación se encarga de la parte financiera, ¿no?

¡Ah si!, algo así como mostrar las tablas de los ingresos diarios o mostrar reportes de ingresos a fin de mes, para poder hacer comparaciones mes a mes, o año por año.

Por último, ¿Cree usted que una aplicación que resolviera sus problemas como la que usted describió podría mejorar la eficiencia operativa del parqueadero?

Claro, si pudiera acelerar el registro de vehículos, reducir los errores de los cálculos manuales y los errores en los cobros, y que pueda ser fácil de operar, nos ahorraría gran parte del trabajo manual que hacemos aquí actualmente.

Muchas gracias por su tiempo, que tenga un buen día.

Anexo 2. Manual de usuario del aplicativo web para la gestión y control vehicular de un parqueadero en la ciudad de Villavicencio

1. Introducción

El presente manual de usuario ha sido diseñado para guiar a los administradores y empleados del parqueadero Parking NoA en el uso eficiente del aplicativo web desarrollado para la gestión y control vehicular. Este sistema permite optimizar tareas operativas como el registro de entradas y salidas de vehículos, la gestión de clientes (por hora y mensuales), la generación de reportes y analíticas, así como la configuración de tarifas y tipos de clientes/vehículos. Está dirigido a usuarios con roles de administrador o empleado, quienes podrán acceder a funcionalidades específicas según sus permisos. A lo largo de este manual, encontrarás instrucciones detalladas y capturas de pantalla que te ayudarán a navegar y utilizar todas las herramientas del sistema de manera efectiva.

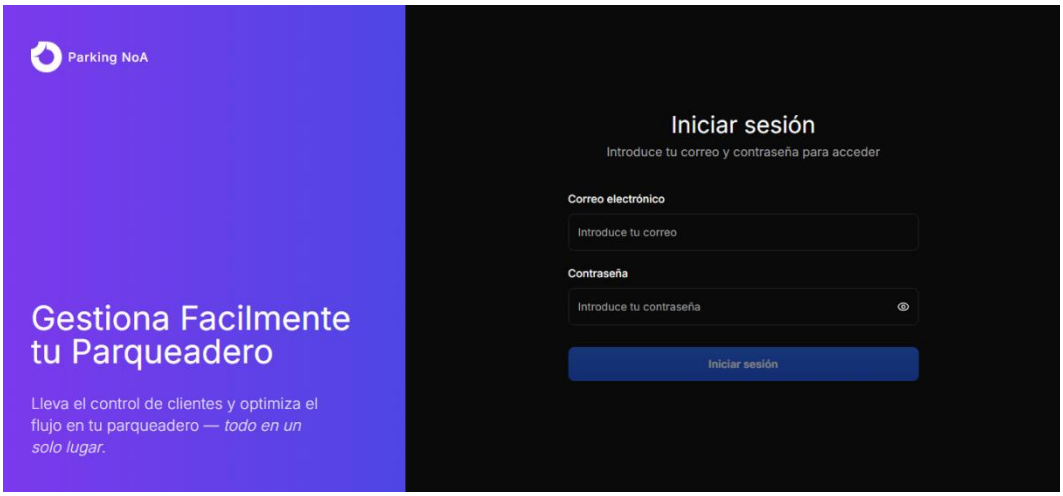
2. Acceso al Sistema

El sistema requiere autenticación para garantizar la seguridad de los datos y el acceso a las funcionalidades según el rol del usuario. A continuación, se describen los pasos para iniciar sesión y cerrar sesión en el sistema.

2.1. Iniciar Sesión

Para comenzar a utilizar el aplicativo, debes iniciar sesión con tus credenciales (correo electrónico y contraseña). Sigue estos pasos:

Figura 25. Pantalla de inicio de sesión



Fuente: Propia del autor

- **Accede a la página de inicio de sesión:** Abre tu navegador web y dirígete a la URL del aplicativo proporcionada por el administrador del sistema. Verás la pantalla de inicio de sesión como se muestra en la Figura 25.

- **Ingresa tu correo electrónico:** En el campo "Correo electrónico", introduce la dirección de correo que le proporcionaste al administrador para hacer el respectivo registro.

- **Ingresa tu contraseña:** En el campo "Contraseña", escribe la contraseña que te llego al correo cuando el administrador realizó tu registro en el aplicativo. Si deseas ver la contraseña mientras la escribes, haz clic en el ícono del ojo a la derecha del campo para alternar la visibilidad.

- **Inicia sesión:** Haz clic en el botón azul "Iniciar sesión". Si las credenciales son correctas, serás redirigido al panel principal del sistema. En caso contrario, aparecerá un mensaje de error indicando que las credenciales son incorrectas.

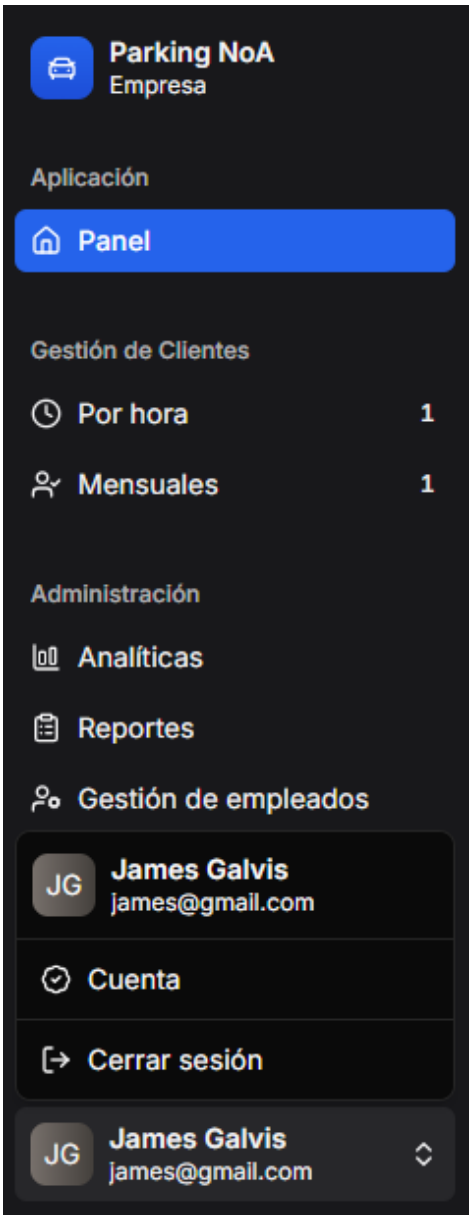
Nota: Si olvidaste tu contraseña o no tienes una cuenta, contacta al administrador del sistema para obtener asistencia, ya que el sistema no cuenta con una funcionalidad de recuperación de contraseña visible en esta pantalla.

2.2. Cerrar Sesión

Cuando hayas terminado de usar el sistema, es importante cerrar sesión para proteger tu cuenta y los datos del parqueadero. Sigue estos pasos:

- **Accede al menú de usuario:** Una vez dentro del sistema, encontrarás un menú lateral a la izquierda de la pantalla. En la parte inferior del menú, se muestra tu perfil de usuario con tu nombre (por ejemplo, "James Galvis") y correo electrónico (por ejemplo, james@gmail.com). Haz clic en esta sección para desplegar las opciones del perfil, como se muestra en la Figura 26.

Figura 26. Menú de usuario con opción de cerrar sesión



Fuente: Propia del autor

- **Selecciona “Cerrar sesión”:** En el menú desplegable, haz clic en la opción "Cerrar sesión". Esto finalizará tu sesión actual de manera segura.

- **Redirección a la página de inicio de sesión:** Tras cerrar sesión, serás redirigido automáticamente a la pantalla de inicio de sesión (Figura 25), donde podrás volver a ingresar con tus credenciales si es necesario.

***Nota*:** Asegúrate de cerrar sesión siempre que termines de usar el sistema, especialmente si estás utilizando un dispositivo compartido.

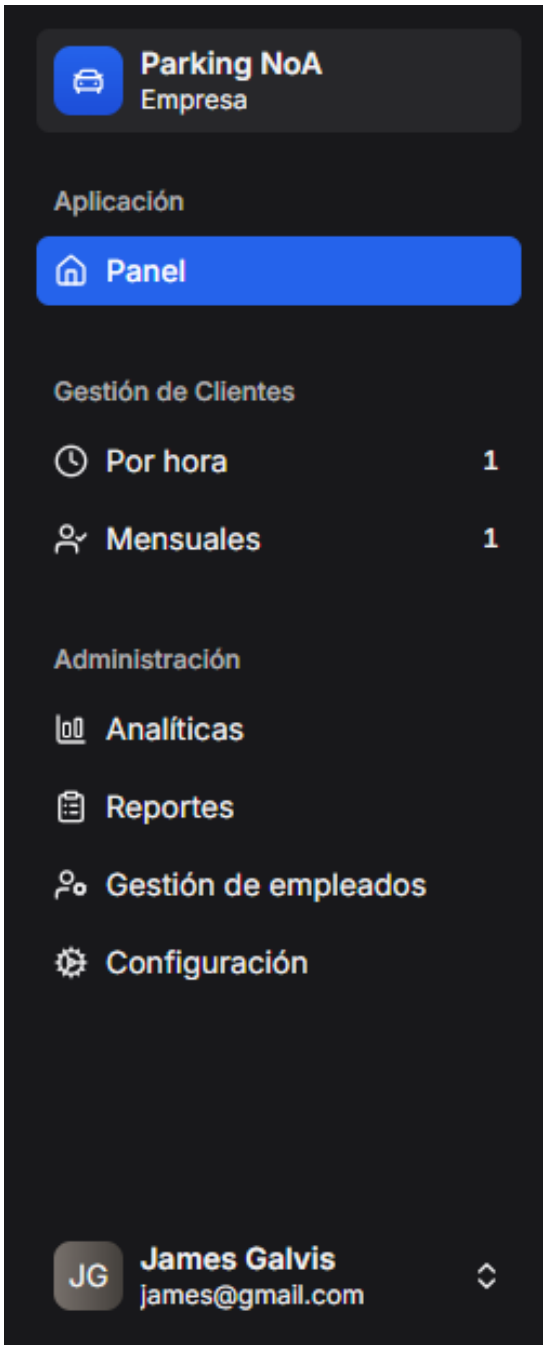
3. Navegación General

El sistema cuenta con un menú lateral que facilita la navegación entre las diferentes secciones del aplicativo. Este menú está diseñado para que los usuarios puedan acceder rápidamente a las funcionalidades según su rol (administrador o empleado). A continuación, se describe cómo utilizar el menú y las opciones disponibles.

3.1. Uso del Menú Lateral

- **Accede al menú lateral**: Una vez que hayas iniciado sesión, verás el menú lateral en el lado izquierdo de la pantalla, como se muestra en la Figura 27.

Figura 27. Menú lateral de navegación



Fuente: Propia del autor

- **Explora las secciones disponibles:** El menú está dividido en tres categorías principales:

- **Aplicación:** Incluye las opciones principales del sistema:
  - **Panel:** Lleva al panel de control principal.
  - **Gestión de Clientes:**
    - **Por hora**: Permite gestionar clientes que pagan por hora.
    - **Mensuales**: Permite gestionar clientes con suscripciones mensuales.
- **Administración:** Incluye herramientas para administradores:
  - **Analíticas:** Muestra gráficos y estadísticas del desempeño del parqueadero.
  - **Reportes:** Permite generar y exportar informes.
  - **Gestión de empleados:** Administra el personal del parqueadero.
  - **Configuración:** Permite configurar tipos de clientes, vehículos y tarifas.
- **Perfil de usuario:** En la parte inferior, se muestra tu nombre y correo electrónico. Al hacer clic aquí, puedes acceder a las opciones de "Cuenta" (para gestionar tu perfil) y "Cerrar sesión".

- **Navega entre secciones:** Haz clic en cualquier opción del menú para acceder a la sección correspondiente. La sección activa se resaltará en azul (por ejemplo, "Panel" en la Figura 27).

***Nota*:** Las opciones del grupo de “Administración” solo estarán disponibles para usuarios con rol de administrador. Si eres un empleado, solo tendrás acceso a las opciones del grupo de “Gestión de clientes”, “Panel” y “Cuenta”.

4. Módulos Principales

Esta sección describe las funcionalidades clave del sistema, organizadas por módulos. Comenzaremos con el **Panel de Control**, que es la pantalla principal que los usuarios ven al iniciar sesión.

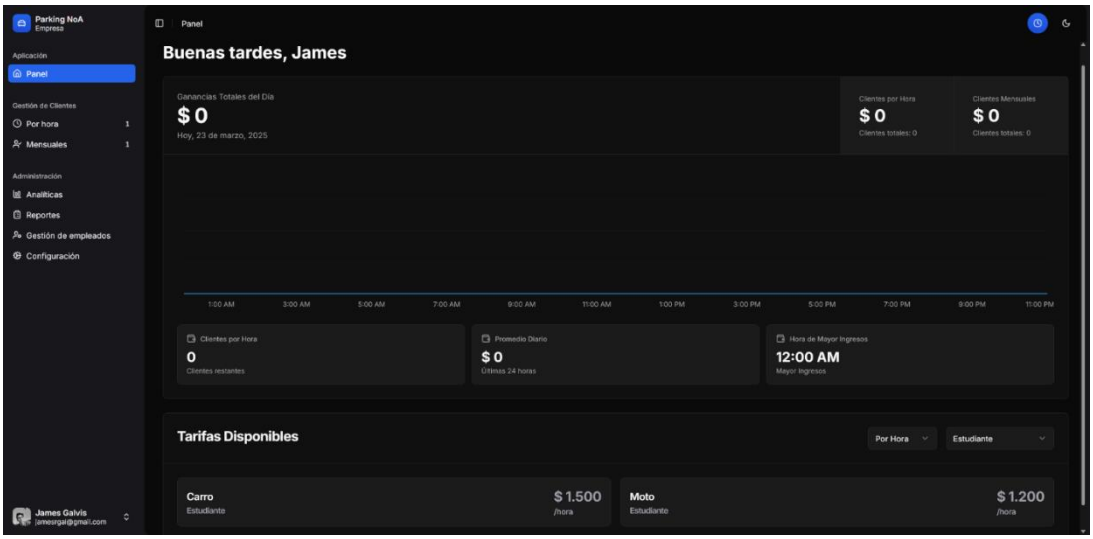
4.1. Panel de Control

El panel de control es la pantalla principal del sistema y proporciona una visión general del estado del parqueadero. Aquí puedes consultar métricas clave, como los ingresos del día, el número de clientes atendidos y las tarifas disponibles. A continuación, se detalla cómo utilizar esta sección.

4.1.1. Visualización de Métricas y Estadísticas

- **Accede al panel de control:** Desde el menú lateral, haz clic en la opción "Panel" bajo la categoría "Aplicación". Esto te llevará a la pantalla del panel de control, como se muestra en la Figura 28.

Figura 28.Panel de control



Fuente: Propia del autor

- **Revisa las métricas principales:** En la parte superior de la pantalla, encontrarás un saludo personalizado y las siguientes métricas:
    - **Ganancias Totales del Día:** Muestra los ingresos acumulados del día actual.
    - **Clientes por Hora:** Indica el total de clientes por hora atendidos en las últimas 24 horas y el valor recaudado.
    - **Clientes Mensuales:** Muestra el total de clientes con suscripciones mensuales que se registraron en el día y el valor recaudado.
- **Consulta el gráfico de actividad:** Debajo de las métricas, hay un gráfico de líneas que muestra la actividad del parqueadero a lo largo del día, dividido por horas. Este gráfico te permite identificar los momentos de mayor ingreso.

- **Revisa las tarifas disponibles:** En la parte inferior, se encuentra la sección "Tarifas Disponibles", que muestra los costos por hora para diferentes tipos de clientes y vehículos. Por ejemplo:
    - **Carro (Estudiante):** $1,500 por hora.
    - **Moto (Estudiante):** $1,200 por hora.

      Puedes cambiar el tipo de cliente (por ejemplo, de "Estudiante" a "Por Hora") usando el menú desplegable para ver las tarifas correspondientes.

***Nota*:** Las métricas y el gráfico se actualizan en tiempo real a medida que se registran nuevos clientes.

4.2. Gestión de Clientes por Hora

El módulo de "Gestión de Clientes por Hora" permite registrar y gestionar los clientes que utilizan el parqueadero por períodos cortos, calculando automáticamente el costo de su estadía según las tarifas configuradas. Este módulo incluye la creación de nuevos clientes, la visualización de clientes activos y la gestión de pagos al momento de su salida. A continuación, se describen los pasos para utilizar esta funcionalidad.

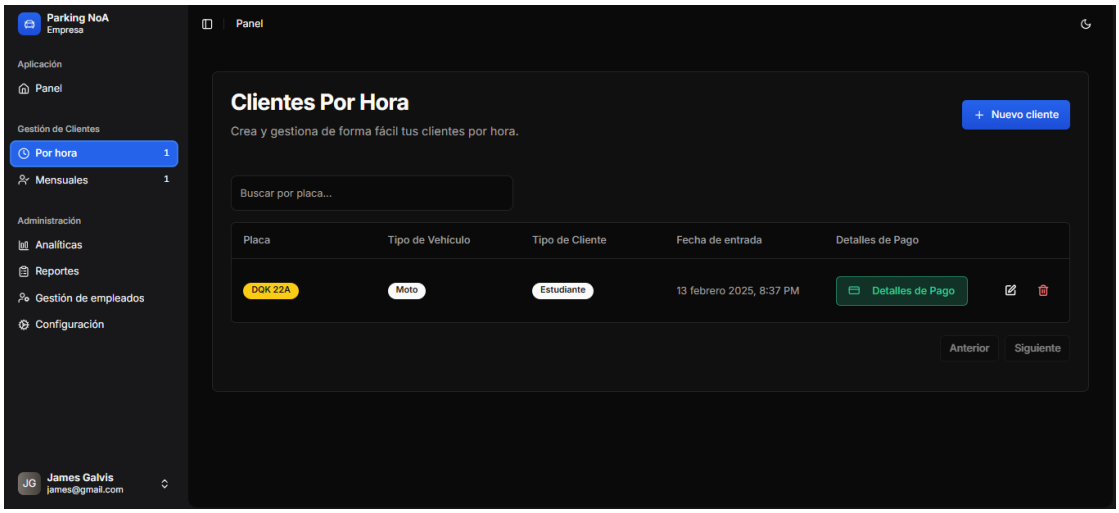
4.2.1. Registrar un Cliente por Hora

Puedes registrar un nuevo cliente por hora desde el formulario de creación, al que puedes acceder de dos formas: desde el módulo "Clientes por Hora" o usando un atajo en la barra de navegación.

- **Accede al formulario de creación:**

    - **Opción 1 - Desde el módulo "Clientes por Hora":** En el menú lateral, bajo "Gestión de Clientes", selecciona "Por hora". Esto te llevará a la pantalla de "Clientes por Hora", como se muestra en la Figura 29. Haz clic en el botón azul "Nuevo cliente" ubicado en la parte superior derecha de la pantalla.

Figura 29. Pantalla de clientes por hora



Fuente: Propia del autor

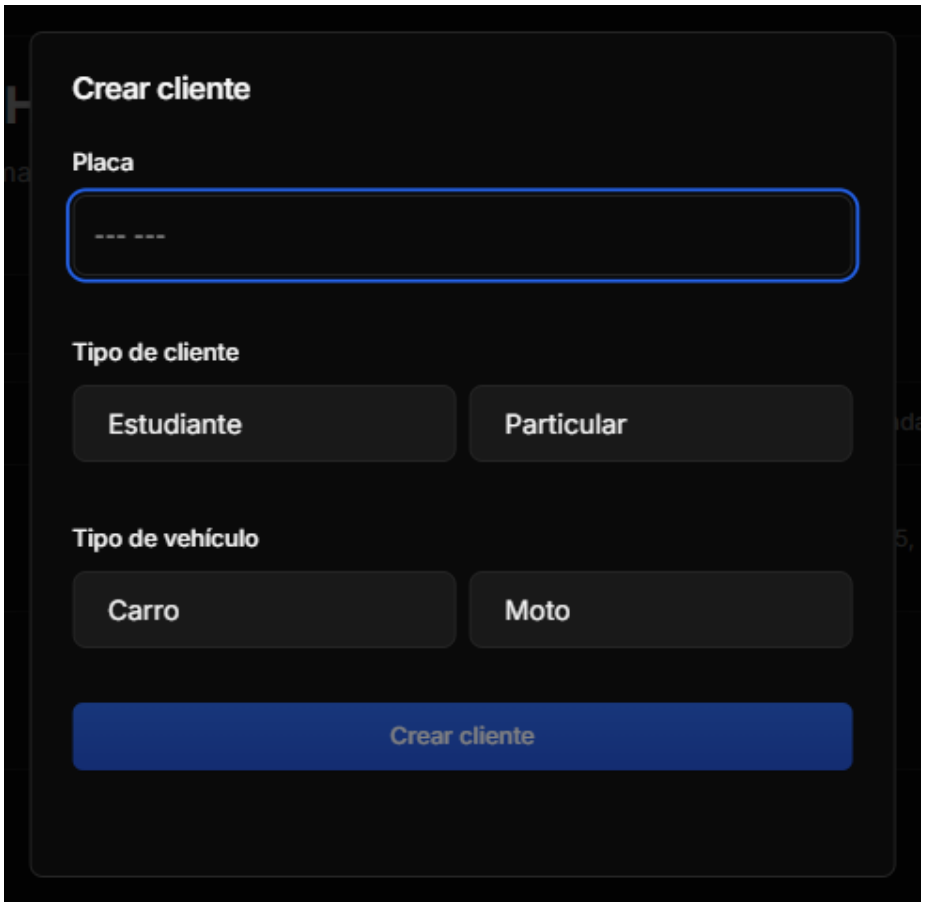
    - **Opción 2 - Atajo desde el Panel de Control:** Desde el panel de control, haz clic en el botón azul con el ícono de reloj ubicado en la barra de navegación superior. Esto abrirá directamente el formulario de creación.

- **Completa el formulario:** Se abrirá un formulario como el que se muestra en la Figura 30. Rellena los siguientes campos:

- **Placa:** Ingresa la placa del vehículo.
- **Tipo de cliente:** Selecciona el tipo de cliente entre las opciones disponibles.
- **Tipo de vehículo:** Selecciona el tipo de vehículo entre las opciones disponibles.

Una vez completados los campos, haz clic en el botón azul "Crear cliente" para registrar el cliente.

Figura 30. Formulario de creación de clientes por hora



Fuente: Propia del autor

- **Verifica el registro:** Una vez creado, el cliente aparecerá en la lista de "Clientes por Hora" (Figura 29), mostrando detalles como la placa, tipo de vehículo, tipo de cliente y fecha de entrada.

Nota: Asegúrate de ingresar correctamente la placa del vehículo, ya que este dato se usará para identificar al cliente al momento de calcular el pago.

4.2.2. Consultar y Gestionar Clientes Activos

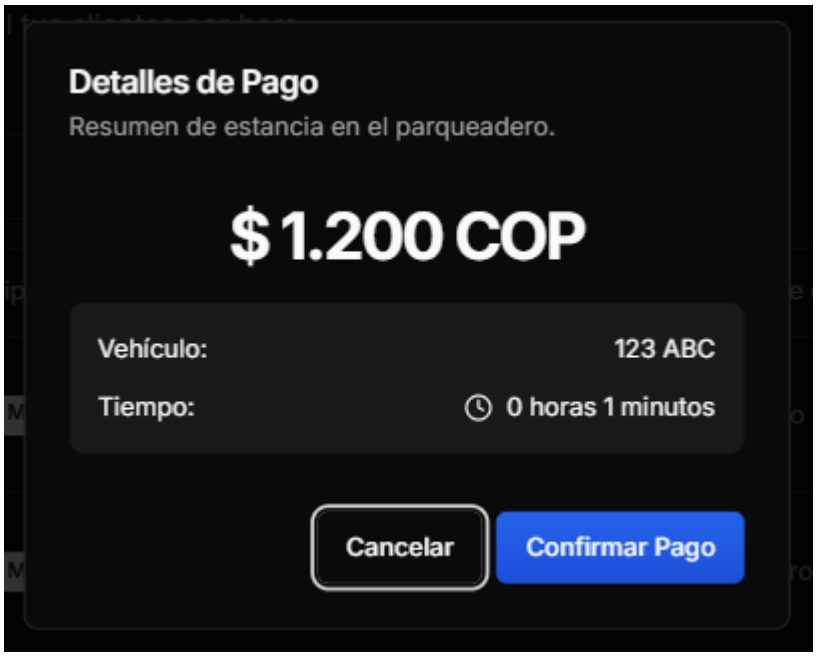
- **Accede a la lista de clientes por hora:** Desde el menú lateral, selecciona "Por hora" bajo "Gestión de Clientes". Esto mostrará la lista de clientes activos, como se observa en la Figura 29.

- **Revisa los detalles:** La lista incluye las siguientes columnas:
  - **Placa.**
  - **Tipo de Vehículo.**
  - **Tipo de Cliente.**
  - **Fecha de entrada.**
  - **Detalles de Pago.**
  - **Acciones.**
- **Navega por la lista:** Si hay varios clientes, usa los botones "Anterior" y "Siguiente" en la parte inferior para moverte entre las páginas de la lista.

4.2.3. Calcular y Confirmar el Pago al Salir

Cuando un cliente por hora desea salir del parqueadero, el empleado debe calcular el costo de la estadía y confirmar el pago. Sigue estos pasos:

Figura 31. Modal de detalles de pago



Fuente: Propia del autor

- **Accede a los detalles de pago:** En la lista de "Clientes por Hora" (Figura 29), identifica el cliente que va a salir y haz clic en el botón verde "Detalles de Pago" en la columna correspondiente. Esto abrirá un modal con el resumen del pago, como se muestra en la Figura 31.

- **Revisa el resumen del pago:** El modal mostrará:

    - **Monto que pagar:** El costo total calculado según la tarifa y el tiempo de estadía.

    - **Vehículo:** La placa del vehículo.

- **Tiempo:** La duración de la estadía.

  Asegúrate de que el monto sea correcto y que el cliente esté de acuerdo con el valor.

- **Confirma o cancela el pago:**
  - **Confirmar Pago:** Si todo está correcto, haz clic en el botón azul "Confirmar Pago". Esto registrará el pago y eliminará al cliente de la lista de clientes activos, indicando que ha salido del parqueadero.
  - **Cancelar:** Si necesitas revisar algo o el cliente no está listo para pagar, haz clic en el botón "Cancelar". Esto cerrará el modal sin realizar cambios, y el cliente permanecerá en la lista de clientes activos.

4.2.4. Editar o Eliminar un Cliente

- **Editar un cliente:** En la lista de "Clientes por Hora", haz clic en el ícono de lápiz en la columna de "Acciones" para modificar los datos del cliente (como la placa, tipo de cliente o tipo de vehículo).
- **Eliminar un cliente:** Haz clic en el ícono de basurero para eliminar el registro del cliente. Esto es útil si el cliente fue registrado por error.

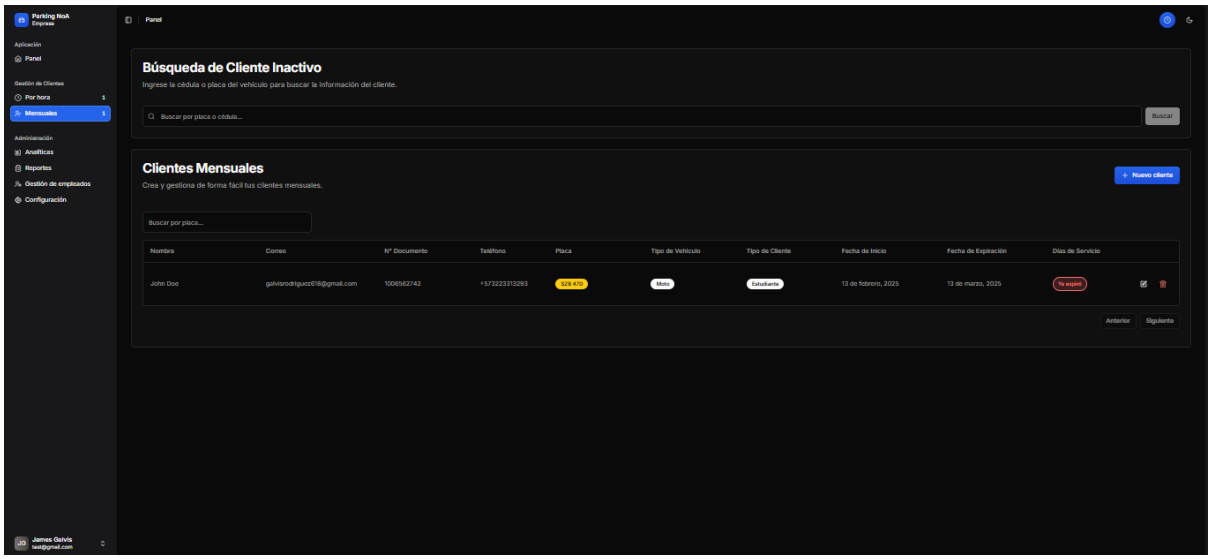
***Nota*:** Ten cuidado al eliminar un cliente, ya que esta acción no se puede deshacer.

4.3. Gestión de Clientes Mensuales

El módulo de "Gestión de Clientes Mensuales" permite registrar y administrar los clientes que tienen suscripciones mensuales en el parqueadero. Este módulo incluye la creación de nuevos clientes, la visualización de clientes activos, la búsqueda de clientes inactivos (para verificar información de suscripciones pasadas) y la edición o eliminación de registros. A continuación, se describen los pasos para utilizar esta funcionalidad.

4.3.1. Registrar un Cliente Mensual

Figura 32. Pantalla de clientes mensuales

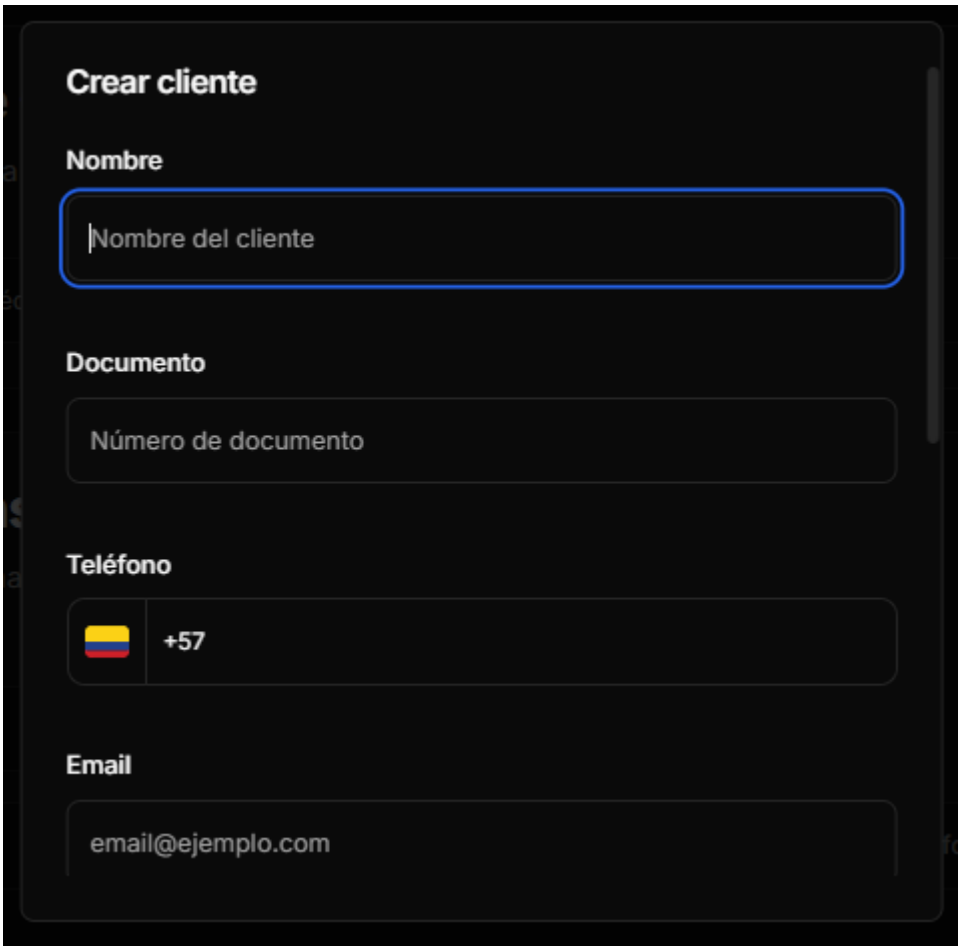


Fuente: Propia del autor

- **Accede al módulo de Clientes Mensuales:** En el menú lateral, bajo "Gestión de Clientes", selecciona "Mensuales". Esto te llevará a la pantalla de "Clientes Mensuales", como se muestra en la Figura 32.

- **Inicia el formulario de creación:** Haz clic en el botón azul "Nuevo cliente" ubicado en la parte superior derecha de la pantalla. Esto abrirá un formulario como el que se muestra en la Figura 33.

Figura 33. Formulario de creación de cliente mensual



Fuente: Propia del autor

- **Completa el formulario:** Rellena los siguientes campos:
  - **Nombre:** Ingresa el nombre completo del cliente.
  - **Número de documento:** Ingresa el número de identificación del cliente.
  - **Teléfono:** Ingresa el número de teléfono del cliente.

- **Email:** Ingresa el correo electrónico del cliente.
- **Matrícula (Placa):** Ingresa la placa del vehículo.
- **Meses para reservar:** Especifica la duración de la suscripción en meses.
- **Tipo de cliente:** Selecciona el tipo de cliente entre las opciones disponibles.
- **Tipo de vehículo:** Selecciona el tipo de vehículo entre las opciones disponibles.
  Una vez completados los campos, haz clic en el botón azul "Crear cliente" para registrar el cliente.

- **Verifica el registro:** Una vez creado, el cliente aparecerá en la lista de "Clientes Mensuales" (Figura 32), mostrando sus datos y el estado de su suscripción.

***Nota*:** Asegúrate de ingresar correctamente los datos, especialmente la placa, correo y el número de documento, ya que estos se usarán para buscar al cliente en el futuro y para hacer el respectivo envío de correos al cliente para avisarle sobre el estado de su servicio.

4.3.2. Consultar Clientes Mensuales Activos

- **Accede a la lista de clientes mensuales:** Desde el menú lateral, selecciona "Mensuales" bajo "Gestión de Clientes". Esto mostrará la lista de clientes activos, como se observa en la Figura 32.

- **Revisa los detalles:** La lista incluye las siguientes columnas:
  - **Nombre:** Nombre del cliente.
  - **Correo:** Correo electrónico del cliente.
  - **N° Documento:** Número de identificación.

- **Teléfono:** Número de contacto.
- **Placa:** Placa del vehículo.
- **Tipo de Vehículo.**
- **Tipo de Cliente.**
- **Fecha de Inicio:** Fecha en que comenzó la suscripción.
- **Fecha de Expiración:** Fecha en que expira la suscripción.
- **Días de Servicio:** Días restantes hasta que expire el servicio (por ejemplo, "20 días").
- **Acciones:** Incluye íconos para editar (lápiz) o eliminar (basurero) el registro del cliente.

- **Navega por la lista:** Si hay varios clientes, usa los botones "Anterior" y "Siguiente" en la parte inferior para moverte entre las páginas de la lista.

4.3.3. Buscar un Cliente Inactivo

El componente "Búsqueda de Cliente Inactivo" permite verificar la información de clientes cuyas suscripciones han expirado, lo que es útil si un cliente reclama que debiese tener más días de servicio o no está seguro de la fecha de expiración de su suscripción anterior.

- **Accede al componente de búsqueda:** En la pantalla de "Clientes Mensuales" (Figura 32), en la parte superior, encontrarás el componente "Búsqueda de Cliente Inactivo".

- **Ingresa la placa o cédula:** En el campo "Buscar por placa o cédula", escribe la placa del vehículo o el número de documento del cliente. Haz clic en el botón "Buscar".

- **Revisa los resultados:** Si el cliente tiene un registro previo, el sistema mostrará los datos de su última suscripción, incluyendo fechas de inicio y expiración, días de servicio y otros detalles relevantes. Si no se encuentra el cliente, aparecerá un mensaje indicando que no hay registros.

***Nota*:** Esta funcionalidad es útil para resolver dudas de los clientes sobre sus suscripciones pasadas. Asegúrate de ingresar correctamente la placa o el número de documento para obtener resultados precisos.

4.3.4. Editar o Eliminar un Cliente Mensual

- **Editar un cliente:** En la lista de "Clientes Mensuales", haz clic en el ícono de lápiz en la columna de "Acciones" para modificar los datos del cliente (como el nombre, correo, placa, etc.).

- **Eliminar un cliente:** Haz clic en el ícono de basurero para eliminar el registro del cliente. Esto es útil si el cliente fue registrado por error o ya no utiliza el servicio.

Nota: Ten cuidado al eliminar un cliente, ya que esta acción no se puede deshacer. Si el cliente aún está activo, considera editar su información en lugar de eliminarla.

4.4. Analíticas

El módulo de "Analíticas" proporciona una visión detallada del desempeño del parqueadero a través de gráficos interactivos y estadísticas en tiempo real. Este

módulo es útil para administradores que desean analizar métricas como ingresos, número de clientes atendidos y patrones de uso del parqueadero, permitiendo tomar decisiones informadas para optimizar las operaciones. A continuación, se describen los pasos para utilizar esta funcionalidad.

4.4.1. Acceder al Módulo de Analíticas

- **Navega al módulo de Analíticas:** En el menú lateral, bajo la categoría "Administración", selecciona la opción "Analíticas". Esto te llevará a la pantalla de analíticas, donde podrás visualizar gráficos y estadísticas detalladas.

Nota: Si no ves la opción "Analíticas" en el menú, es posible que tu rol de usuario (por ejemplo, empleado) no tenga permisos para acceder a esta sección. Contacta al administrador del sistema para obtener más información.

4.4.2. Visualizar Gráficos y Estadísticas

- **Explora las métricas principales:** Al ingresar al módulo de Analíticas, encontrarás gráficos interactivos y estadísticas similares a los del Panel de Control, pero con mayor detalle. Estas métricas pueden incluir:
    - **Ingresos por periodo:** Gráficos que muestra los ingresos acumulados por mes y año, desglosados por tipo de cliente (por hora y mensuales).

- **Número de clientes atendidos:** Estadísticas sobre la cantidad de clientes por hora y mensuales atendidos en un periodo específico.

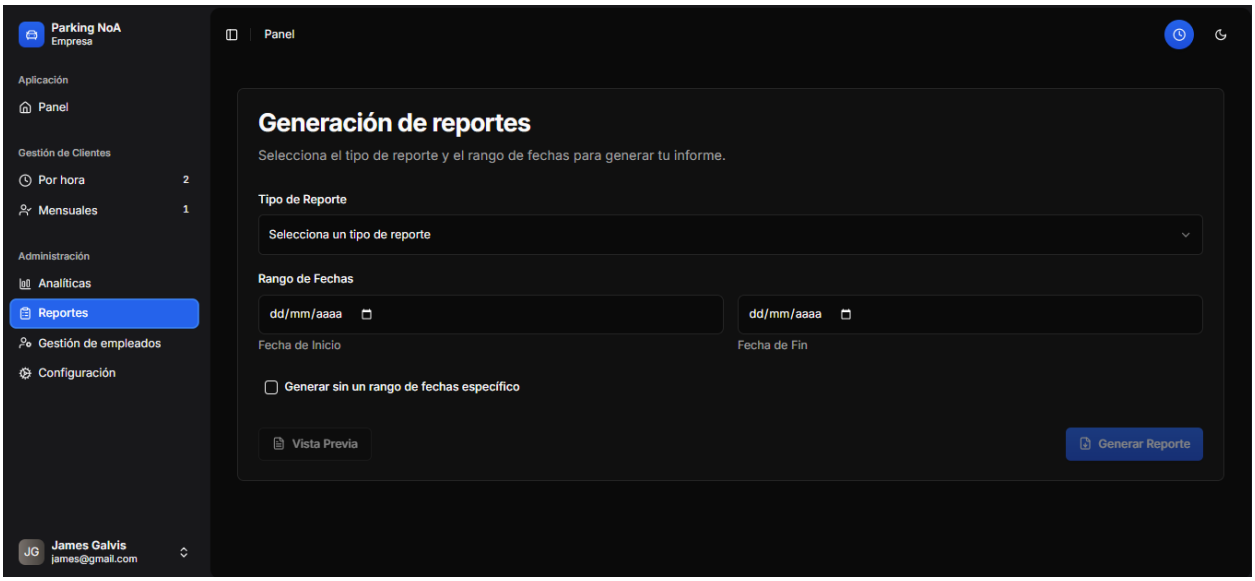
4.5. Reportes

El módulo de "Reportes" permite a los administradores generar informes detallados sobre las operaciones del parqueadero, como las ganancias, los clientes atendidos y los servicios próximos a expirar. Este módulo ofrece opciones para personalizar los reportes seleccionando el tipo de informe y un rango de fechas, además de una vista previa para verificar los datos antes de generar el informe final. A continuación, se describen los pasos para utilizar esta funcionalidad.

4.5.1. Acceder al Módulo de Reportes

- **Navega al módulo de Reportes:** En el menú lateral, bajo la categoría "Administración", selecciona la opción "Reportes". Esto te llevará a la pantalla de "Generación de reportes", como se muestra en la Figura 34.

Figura 34. Pantalla de generación de reportes



Fuente: Propia del autor

***Nota**:* Si no ves la opción "Reportes" en el menú, es posible que tu rol de usuario (por ejemplo, empleado) no tenga permisos para acceder a esta sección. Contacta al administrador del sistema para obtener más información.

4.5.2. Configurar y Generar un Reporte

- **Selecciona el tipo de reporte:** En la pantalla de "Generación de reportes", encontrarás un menú desplegable etiquetado como "Tipo de Reporte". Haz clic en el menú para ver las opciones disponibles:
    - **Ganancias:** Muestra los ingresos generados por el parqueadero.
    - **Clientes por Hora:** Detalla los clientes atendidos por hora.

- **Clientes Mensuales:** Lista los clientes con suscripciones mensuales.

- **Próximos a Expirar:** Muestra los clientes mensuales cuyas suscripciones están próximas a vencer.

  Selecciona el tipo de reporte que deseas generar (por ejemplo, "Ganancias").

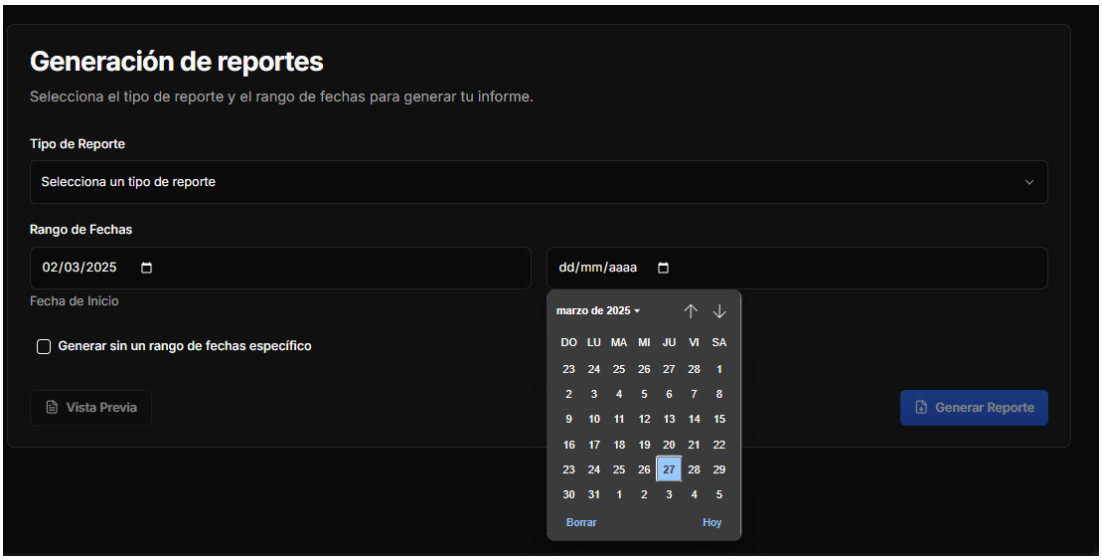
- **Define el rango de fechas:**

  - Si deseas un reporte con un rango de fechas específico, desmarca la casilla "Generar reporte de ganancias anual" (si está marcada).

  - En los campos "Fecha de Inicio" y "Fecha de Fin", ingresa las fechas deseadas en el formato "dd/mm/aaaa" (por ejemplo, "01/01/2025" y "31/03/2025"). Puedes usar el calendario que aparece al hacer clic en el ícono de calendario junto a cada campo, como se muestra en la Figura 35.

  - Si prefieres un reporte anual sin un rango específico, marca la casilla "Generar reporte de ganancias anual".

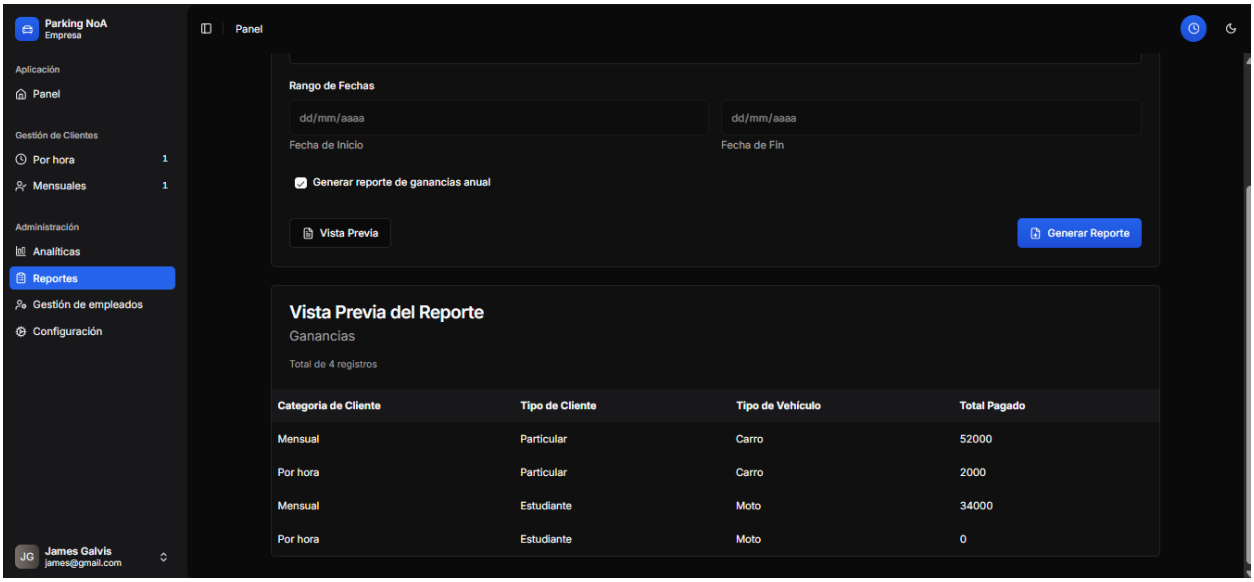
Figura 35. Selección de fechas con el calendario



Fuente: Propia del autor

- **Visualiza la vista previa (opcional):** Haz clic en el botón "Vista Previa" para ver una previsualización del reporte antes de generarlo. Esto abrirá una tabla con los datos correspondientes al tipo de reporte seleccionado, como se muestra en la Figura 36.

Figura 36. Vista previa del reporte de ganancias



Fuente: Propia del autor

- **Genera el reporte:** Una vez que hayas configurado el tipo de reporte y el rango de fechas, y estés satisfecho con la vista previa (si la usaste), haz clic en el botón azul "Generar Reporte". Esto descargará el informe en formato Excel.

***Nota*:** Si no hay datos para generar el reporte el sistema te lo indicara con un mensaje en pantalla.

4.6. Gestión de Empleados

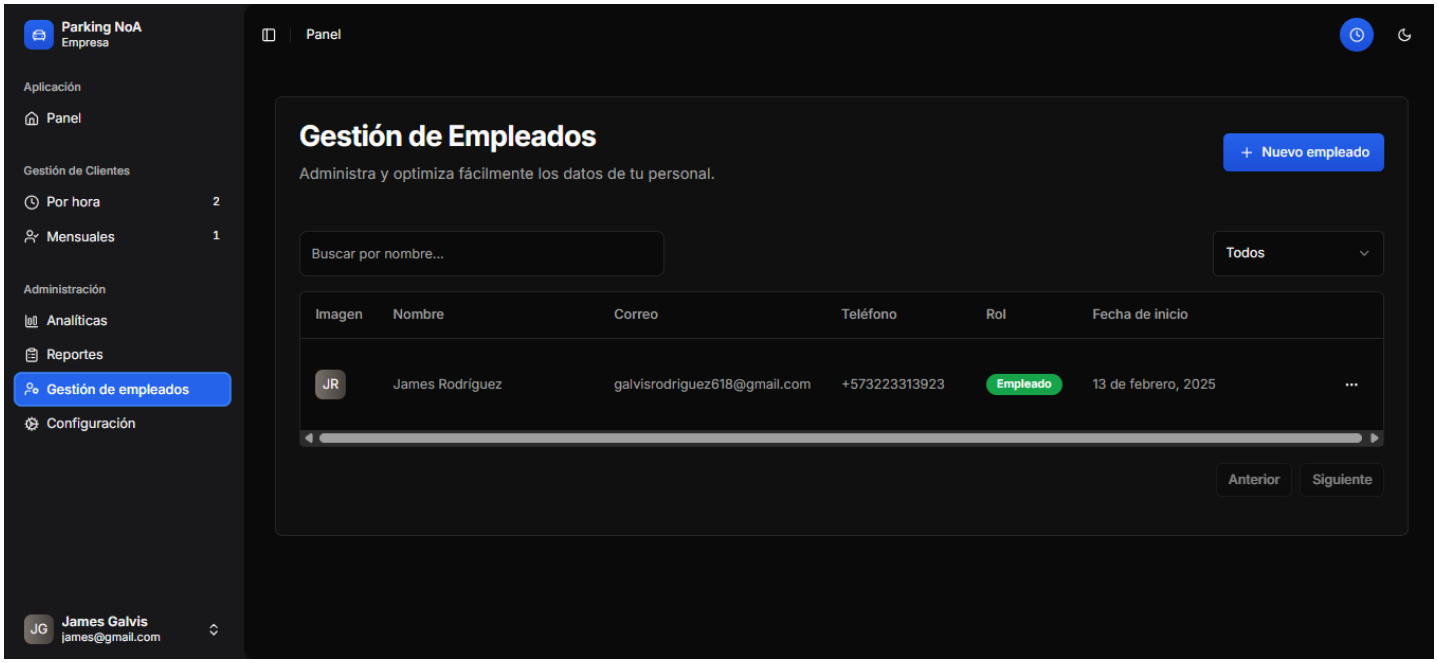
El módulo de "Gestión de Empleados" permite a los administradores gestionar el personal del parqueadero, incluyendo la creación de nuevos empleados, la visualización de empleados activos, la edición de sus datos y la eliminación de registros. Este módulo es esencial para mantener un control eficiente del equipo de

trabajo y sus roles dentro del sistema. A continuación, se describen los pasos para utilizar esta funcionalidad.

4.6.1. Acceder al Módulo de Gestión de Empleados

- **Navega al módulo de Gestión de Empleados:** En el menú lateral, bajo la categoría "Administración", selecciona la opción "Gestión de empleados". Esto te llevará a la pantalla de "Gestión de Empleados", como se muestra en la Figura 37.

Figura 37. Pantalla de Gestión de Empleados



Fuente: Propia del autor

***Nota**:* Si no ves la opción "Gestión de empleados" en el menú, es posible que tu rol de usuario (por ejemplo, empleado) no tenga permisos para acceder a esta sección. Contacta al administrador del sistema para obtener más información.

4.6.2. Consultar Empleados Activos

- **Accede a la lista de empleados:** En la pantalla de "Gestión de Empleados", encontrarás una tabla con la lista de empleados activos, como se observa en la Figura 37.

- **Revisa los detalles:** La tabla incluye las siguientes columnas:
  - **Imagen:** Muestra las iniciales del empleado.
  - **Nombre:** Nombre completo del empleado.
  - **Correo:** Correo electrónico del empleado.
  - **Teléfono:** Número de contacto del empleado.
  - **Rol:** Rol asignado al empleado (por ejemplo, "Admin" o "Empleado").
  - **Fecha de inicio:** Fecha en que el empleado comenzó a trabajar (por ejemplo, "13 de febr., 2025").
  - **Acciones:** Incluye opciones para editar (ícono de lápiz) o eliminar (ícono de basurero) el registro del empleado.

- **Filtra la lista:** Usa el menú desplegable en la parte superior derecha (por ejemplo, "Todos", "Admin", "Empleado") para filtrar la lista según el rol del empleado. Esto es útil si tienes muchos empleados y deseas ver solo los de un rol específico.

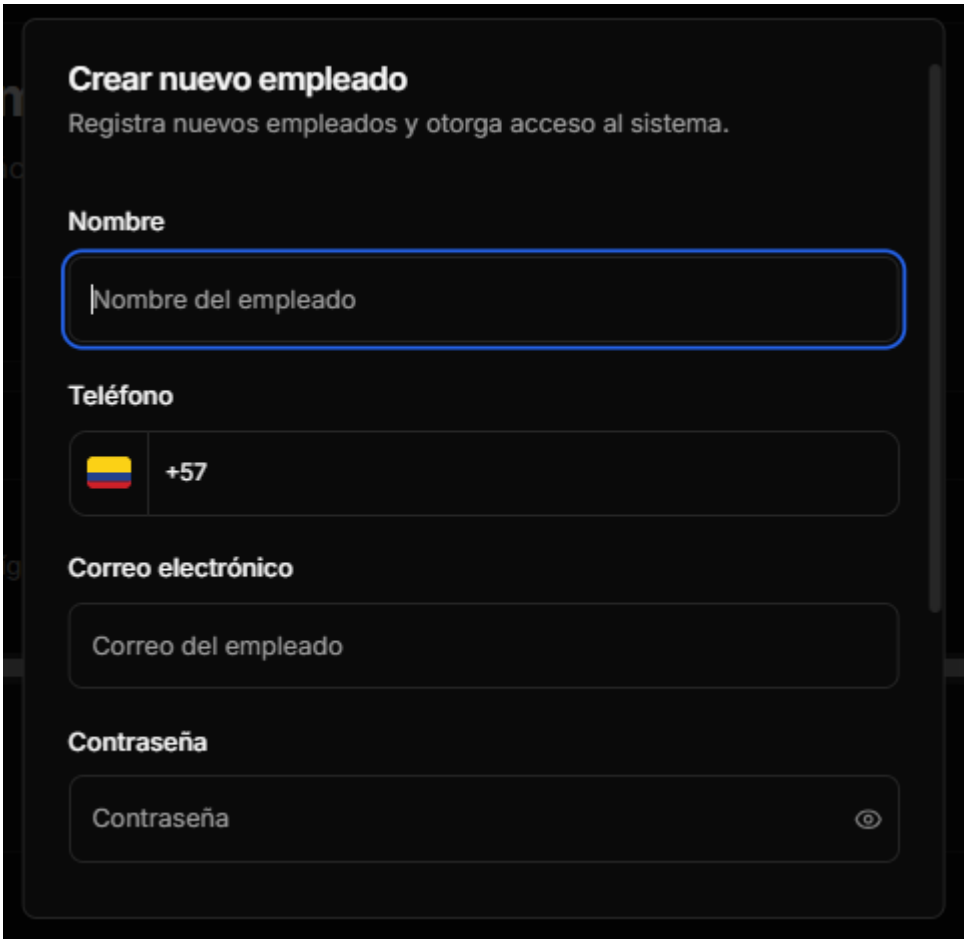
- **Busca un empleado:** En el campo "Buscar por nombre...", ingresa el nombre del empleado que deseas encontrar. La tabla se actualizará automáticamente para mostrar los resultados coincidentes.

- **Navega por la lista:** Si hay varios empleados, usa los botones "Anterior" y "Siguiente" en la parte inferior para moverte entre las páginas de la lista.

4.6.3. Crear un Nuevo Empleado

- **Inicia el formulario de creación:** Haz clic en el botón azul "Nuevo empleado" ubicado en la parte superior derecha de la pantalla de "Gestión de Empleados".

Figura 38. Formulario de creación de empleado



Fuente: Propia del autor

- **Completa el formulario:** Rellena los siguientes campos:
    - **Nombre:** Ingresa el nombre completo del empleado.
    - **Correo electrónico:** Ingresa el correo del empleado.
    - **Teléfono:** Ingresa el número de teléfono.
    - **Rol:** Selecciona el rol del empleado entre las opciones disponibles: "Admin" o "Empleado".
    - **Contraseña:** Ingresa una contraseña para el empleado. La contraseña debe cumplir con los siguientes requisitos:
        - Mínimo 8 caracteres.
        - Al menos 1 número.
        - Al menos 1 carácter especial (por ejemplo, !, @, #).
        - Al menos 1 carácter en mayúscula
- **Crea el empleado:** Una vez completados los campos, haz clic en el botón azul "Crear usuario". El empleado será registrado en el sistema y aparecerá en la lista de "Gestión de Empleados".

***Nota*:** Asegúrate de que el correo electrónico sea único, ya que el sistema no permite registrar dos empleados con el mismo correo.

4.6.4. Editar o Eliminar un Empleado

- **Editar un empleado:** En la lista de empleados, haz clic en el ícono de lápiz en la columna de "Acciones" para modificar los datos del empleado (como nombre, correo, teléfono o rol). Esto abrirá un formulario similar al de creación, pero con los datos actuales del empleado. Realiza los cambios necesarios y guarda la información.

- **Eliminar un empleado:** Haz clic en el ícono de basurero en la columna de "Acciones" para eliminar el registro del empleado. Se mostrará un mensaje de confirmación para evitar eliminaciones accidentales.

***Nota*:** Ten cuidado al eliminar un empleado, ya que esta acción no se puede deshacer. Asegúrate de que el empleado ya no forme parte del equipo antes de eliminar su registro.

4.6.5. Gestionar Permisos y Roles

- **Rol de Admin:** Los administradores tienen acceso a todas las funcionalidades del sistema, incluyendo la gestión de empleados, analíticas, reportes y configuración.

- **Rol de Empleado:** Los empleados tienen acceso limitado, generalmente a la gestión de clientes por hora y mensuales, y no pueden acceder a módulos como "Analíticas", "Reportes" o "Gestión de empleados".

- Si necesitas cambiar el rol de un empleado (por ejemplo, de "Empleado" a "Admin"), edita su perfil y selecciona el nuevo rol en el formulario de edición.

***Nota*:** Cambiar el rol de un empleado puede afectar su acceso al sistema. Asegúrate de que el cambio sea necesario y de informar al empleado sobre las nuevas responsabilidades.

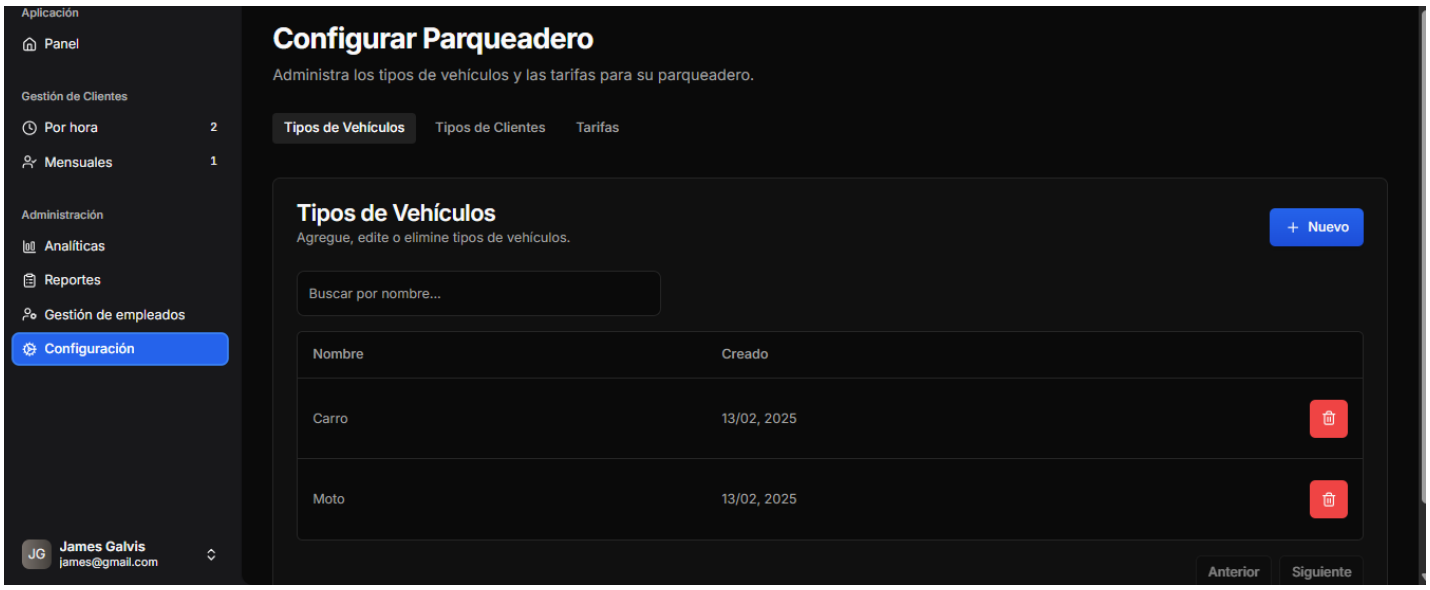
4.7. Configuración

El módulo de "Configuración" permite a los administradores personalizar los parámetros operativos del parqueadero, incluyendo los tipos de vehículos, tipos de clientes y tarifas. Este módulo es esencial para definir las categorías de vehículos y clientes que el sistema manejará, así como establecer las tarifas aplicables para los servicios por hora y mensuales. A continuación, se describen los pasos para utilizar esta funcionalidad.

4.7.1. Acceder al Módulo de Configuración

- **Navega al módulo de Configuración:** En el menú lateral, bajo la categoría "Administración", selecciona la opción "Configuración". Esto te llevará a la pantalla de "Configurar Parqueadero", como se muestra en la Figura 39.

Figura 39. Pantalla de configuración – Tipos de vehículos



Fuente: Propia del autor

***Nota**:* Si no ves la opción "Configuración" en el menú, es posible que tu rol de usuario (por ejemplo, empleado) no tenga permisos para acceder a esta sección. Contacta al administrador del sistema para obtener más información.

4.7.2. Gestionar Tipos de Vehículos

- **Accede a la pestaña de Tipos de Vehículos:** En la pantalla de "Configurar Parqueadero", selecciona la pestaña "Tipos de Vehículos". Esto mostrará una lista de los tipos de vehículos registrados, como se observa en la Figura 39.

- **Consulta los tipos de vehículos:** La lista incluye las siguientes columnas:

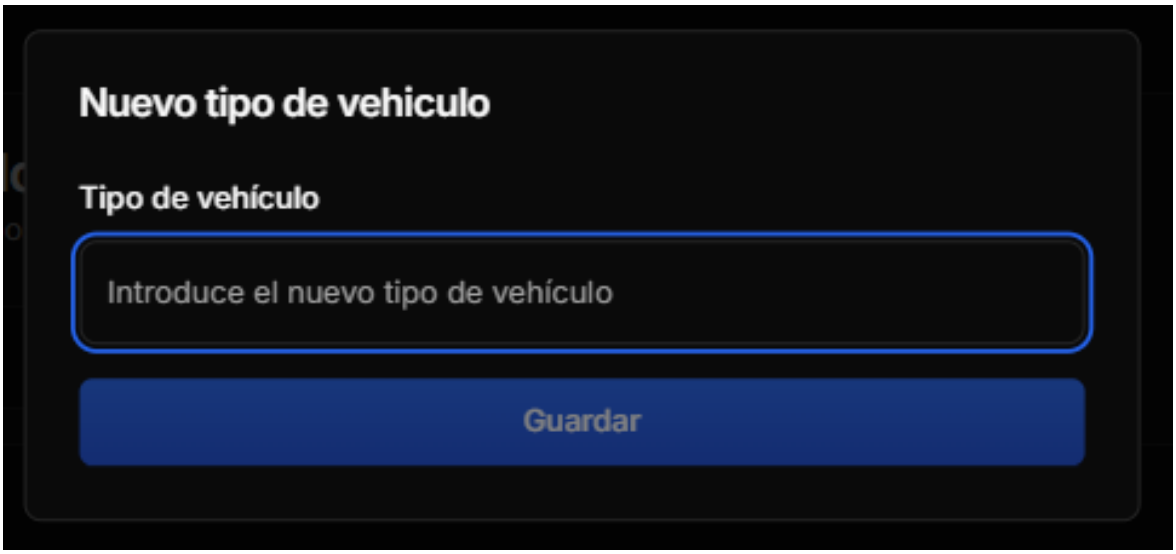
    - **Nombre:** Nombre del tipo de vehículo (por ejemplo, "Carro", "Moto").

- **Creado:** Fecha de creación del tipo de vehículo (por ejemplo, "13/02/2025").
- **Acciones:** Incluye un ícono de basurero para eliminar el tipo de vehículo.

- **Busca un tipo de vehículo:** En el campo "Buscar por nombre...", ingresa el nombre del tipo de vehículo que deseas encontrar. La lista se actualizará automáticamente para mostrar los resultados coincidentes.

- **Agrega un nuevo tipo de vehículo:**
  - Haz clic en el botón azul "Nuevo" en la parte superior derecha de la sección "Tipos de Vehículos". Esto abrirá un formulario como el que se muestra en la Figura 40.
  - En el campo "Tipo de vehículo", ingresa el nombre del nuevo tipo de vehículo (por ejemplo, "Bicicleta").
  - Haz clic en el botón azul "Guardar" para registrar el nuevo tipo de vehículo. El tipo aparecerá en la lista.

Figura 40. Formulario para agregar un nuevo tipo de vehículo



Fuente: Propia del autor

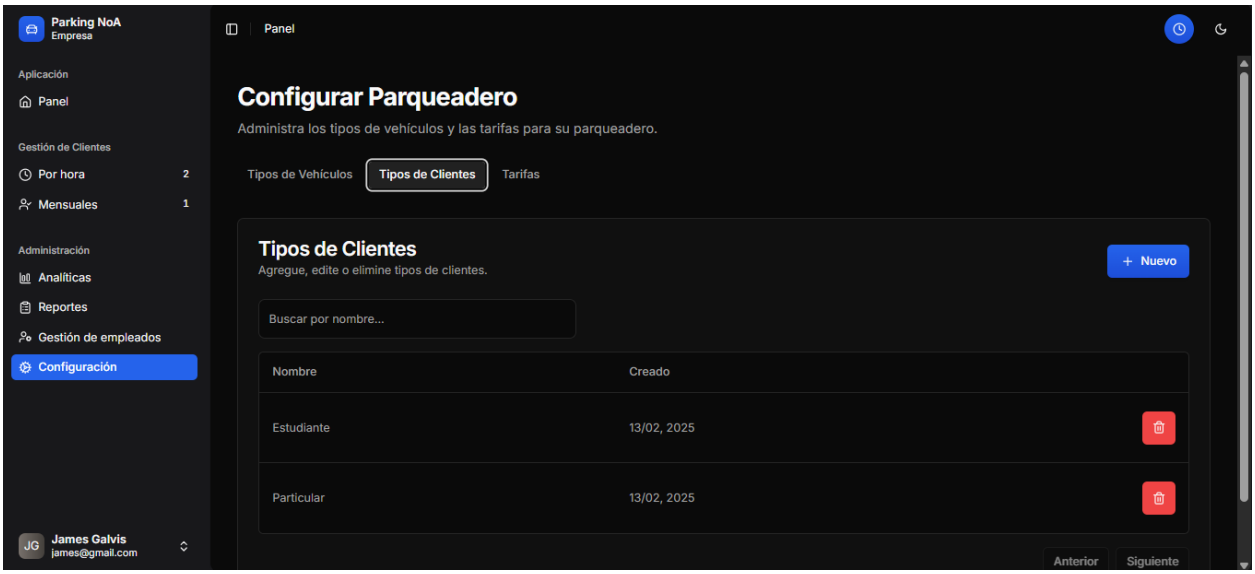
- **Elimina un tipo de vehículo:** Haz clic en el ícono de basurero en la columna de "Acciones" para eliminar un tipo de vehículo. Se mostrará un mensaje de confirmación para evitar eliminaciones accidentales.

***Nota*:** Ten cuidado al eliminar un tipo de vehículo, ya que esta acción no se puede deshacer y puede afectar los registros de clientes asociados a ese tipo de vehículo.

4.7.3. Gestionar Tipos de Clientes

- **Accede a la pestaña de Tipos de Clientes:** En la pantalla de "Configurar Parqueadero", selecciona la pestaña "Tipos de Clientes". Esto mostrará una lista de los tipos de clientes registrados, como se observa en la Figura 41.

Figura 41. Pantalla de configuración – tipos de clientes

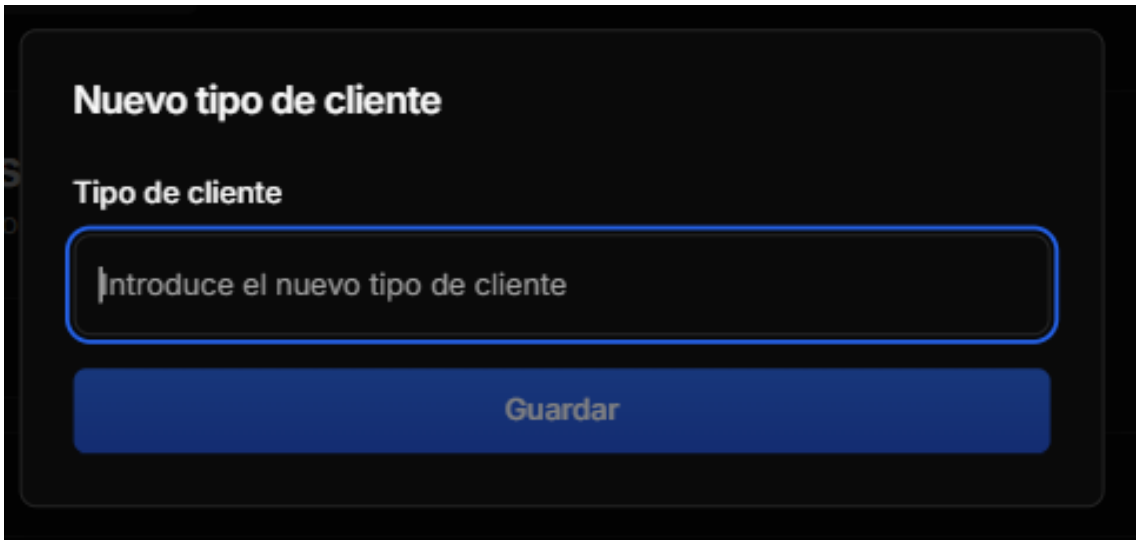


Fuente: Propia del autor

- **Consulta los tipos de clientes:** La lista incluye las siguientes columnas:
    - **Nombre:** Nombre del tipo de cliente (por ejemplo, "Estudiante", "Particular").
    - **Creado:** Fecha de creación del tipo de cliente (por ejemplo, "13/02/2025").
    - **Acciones:** Incluye un ícono de basurero para eliminar el tipo de cliente.
- **Busca un tipo de cliente:** En el campo "Buscar por nombre...", ingresa el nombre del tipo de cliente que deseas encontrar. La lista se actualizará automáticamente para mostrar los resultados coincidentes.
- **Agrega un nuevo tipo de cliente:**

- Haz clic en el botón azul "Nuevo" en la parte superior derecha de la sección "Tipos de Clientes". Esto abrirá un formulario como el que se muestra en la Figura 42.
- En el campo "Tipo de cliente", ingresa el nombre del nuevo tipo de cliente (por ejemplo, "Profesor").
- Haz clic en el botón azul "Guardar" para registrar el nuevo tipo de cliente. El tipo aparecerá en la lista.

Figura 42. Formulario para agregar un nuevo tipo de cliente



Fuente: Propia del autor

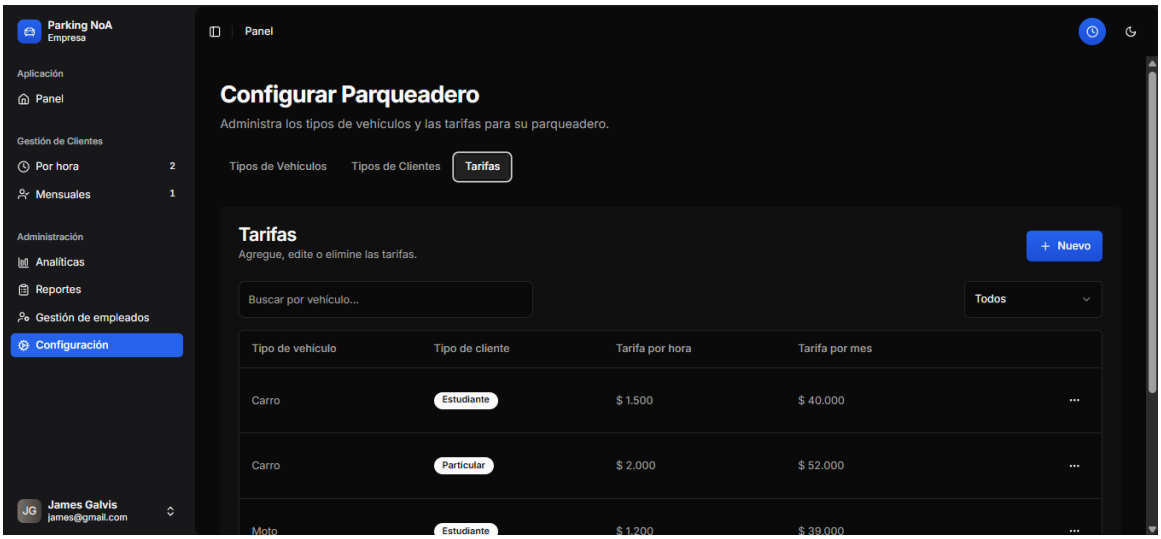
- **Elimina un tipo de cliente:** Haz clic en el ícono de basurero en la columna de "Acciones" para eliminar un tipo de cliente. Se mostrará un mensaje de confirmación para evitar eliminaciones accidentales,

***Nota*:** Ten cuidado al eliminar un tipo de cliente, ya que esta acción no se puede deshacer y puede afectar los registros de clientes asociados a ese tipo.

4.7.4. Gestionar Tarifas

- **Accede a la pestaña de Tarifas:** En la pantalla de "Configurar Parqueadero", selecciona la pestaña "Tarifas". Esto mostrará una lista de las tarifas registradas, como se observa en la Figura 43.

Figura 43. Pantalla de configuración - Tarifas



Fuente: Propia del autor

- **Consulta las tarifas:** La lista incluye las siguientes columnas:
  - **Tipo de vehículo:** Tipo de vehículo al que aplica la tarifa (por ejemplo, "Carro", "Moto").
  - **Tipo de cliente:** Tipo de cliente al que aplica la tarifa (por ejemplo, "Estudiante", "Particular").
  - **Tarifa por hora:** Costo por hora para el servicio por hora (por ejemplo, "$1,500").
  - **Tarifa por mes:** Costo mensual para el servicio de suscripción (por ejemplo, "$40,000").

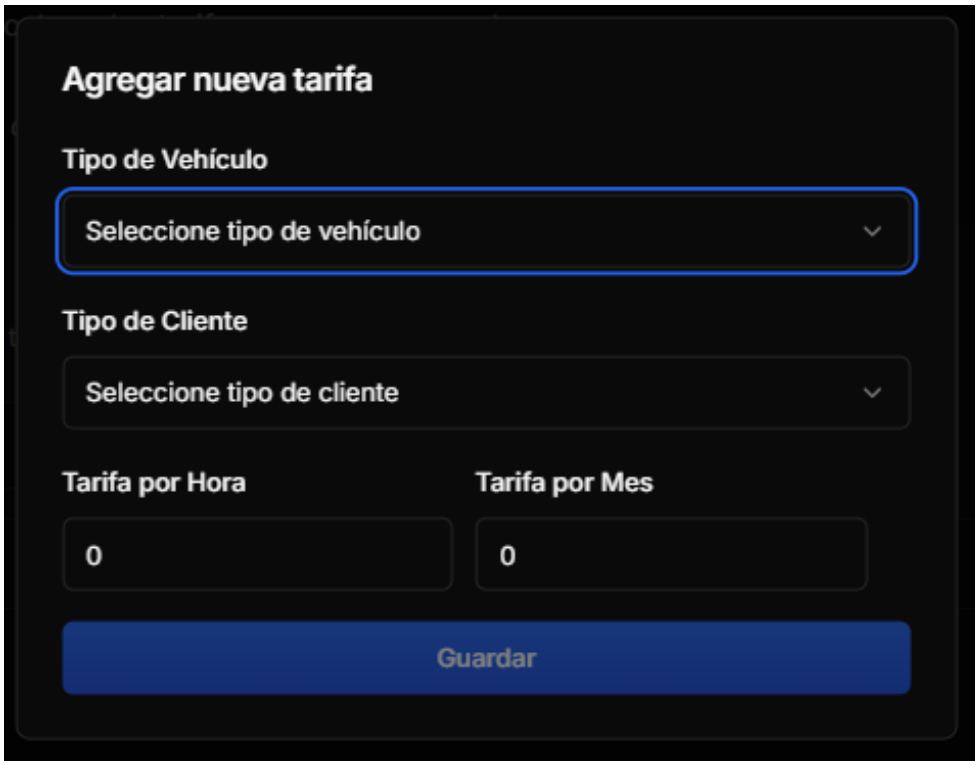
- **Acciones:** Incluye un ícono de tres puntos para opciones adicionales (como editar o eliminar)

- **Busca una tarifa:** En el campo "Buscar por vehículo...", ingresa el tipo de vehículo o cliente que deseas encontrar. La lista se actualizará automáticamente para mostrar los resultados coincidentes.

- **Agrega una nueva tarifa:**

  - Haz clic en el botón azul "Nueva" en la parte superior derecha de la sección "Tarifas". Esto abrirá un formulario como el que se muestra en la Figura 44.
  - En el campo "Tipo de vehículo", selecciona un tipo de vehículo de la lista desplegable (por ejemplo, "Carro").
  - En el campo "Tipo de cliente", selecciona un tipo de cliente de la lista desplegable (por ejemplo, "Estudiante").
  - En el campo "Tarifa por hora", ingresa el costo por hora (por ejemplo, "1500").
  - En el campo "Tarifa por mes", ingresa el costo mensual (por ejemplo, "40000").
  - Haz clic en el botón azul "Guardar" para registrar la nueva tarifa. La tarifa aparecerá en la lista.

Figura 44. Formulario para agregar una nueva tarifa



Fuente: Propia del autor

- **Edita o elimina una tarifa:** Haz clic en el ícono de tres puntos en la columna de "Acciones" para ver las opciones de edición o eliminación.
    - **Editar:** Selecciona "Editar" para modificar los valores de la tarifa (por ejemplo, cambiar la tarifa por hora o por mes).
    - **Eliminar:** Selecciona "Eliminar" para remover la tarifa. Se mostrará un mensaje de confirmación para evitar eliminaciones accidentales.

***Nota*:** Asegúrate de que las tarifas sean coherentes con las políticas del parqueadero. Cambiar o eliminar una tarifa puede afectar los cálculos de ingresos y los servicios ofrecidos a los clientes.

4.7.5. Consideraciones Generales

- **Impacto de los cambios:** Los cambios realizados en los tipos de vehículos, tipos de clientes o tarifas se reflejarán inmediatamente en el sistema. Por ejemplo, si eliminas un tipo de vehículo, no podrás registrar nuevos clientes con ese tipo de vehículo.

- **Permisos:** Solo los usuarios con rol de "Admin" pueden acceder al módulo de "Configuración". Los empleados con rol de "Empleado" no tienen acceso a esta sección.

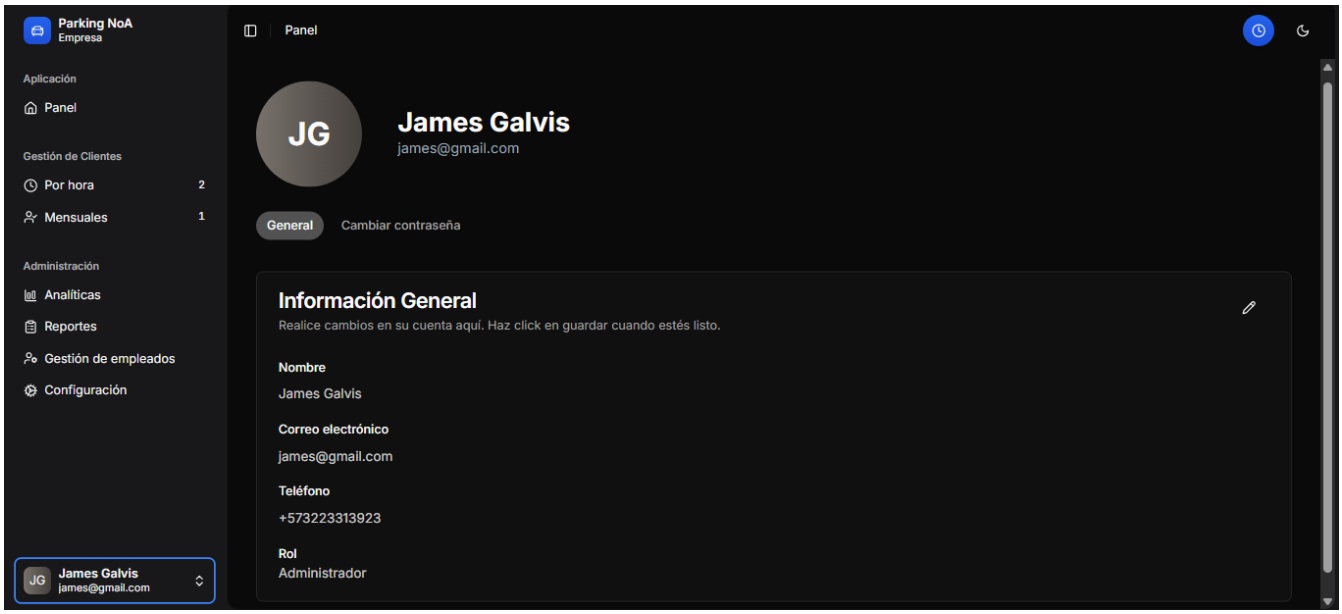
4.8. Perfil de Usuario

El módulo de "Perfil de Usuario" permite a los usuarios visualizar y gestionar su información personal registrada en el sistema, como su nombre, correo electrónico, número de teléfono y rol. Además, ofrece la opción de cambiar la contraseña para mantener la seguridad de la cuenta. Este módulo es accesible para todos los usuarios, independientemente de su rol (Admin o Empleado). A continuación, se describen los pasos para utilizar esta funcionalidad.

4.8.1. Acceder al Módulo de Perfil de Usuario

- **Navega al módulo de Perfil de Usuario:** En la parte inferior del menú lateral, haz clic en el nombre de usuario. Esto te llevará a la pantalla de "Perfil de Usuario", como se muestra en la Figura 45.

Figura 45. Pantalla de perfil de usuario - Información general



Fuente: Propia del autor

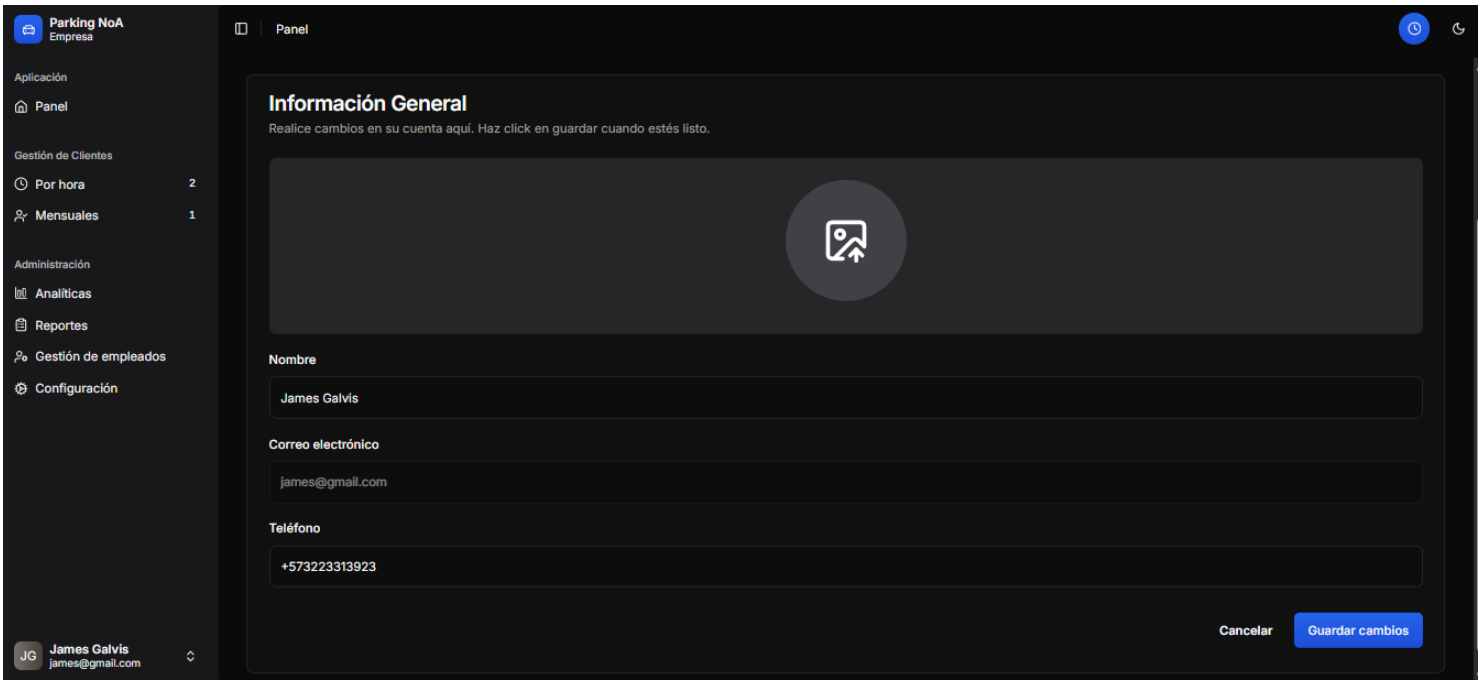
4.8.2. Consultar Información del Perfil

- **Accede a la pestaña de Información General:** En la pantalla de "Perfil de Usuario", asegúrate de estar en la pestaña "General". Aquí podrás ver los siguientes detalles:
    - **Nombre:** Nombre completo del usuario.
    - **Correo electrónico:** Correo registrado en el sistema.
    - **Teléfono:** Número de contacto, incluyendo el código de país.
    - **Rol:** Rol asignado al usuario

4.8.3. Editar Información del Perfil

- **Inicia la edición:** En la pestaña "General", haz clic en el ícono de lápiz ubicado en la parte superior derecha de la sección "Información General". Esto habilitará los campos para edición, como se muestra en la Figura 46.

Figura 46. Edición de Información General.



Fuente: Propia del autor

- **Modifica los datos:** Actualiza los campos según sea necesario:
  - **Nombre:** Cambia tu nombre completo si es necesario.
  - **Correo electrónico:** Actualiza tu correo electrónico.
  - **Teléfono:** Modifica tu número de teléfono.

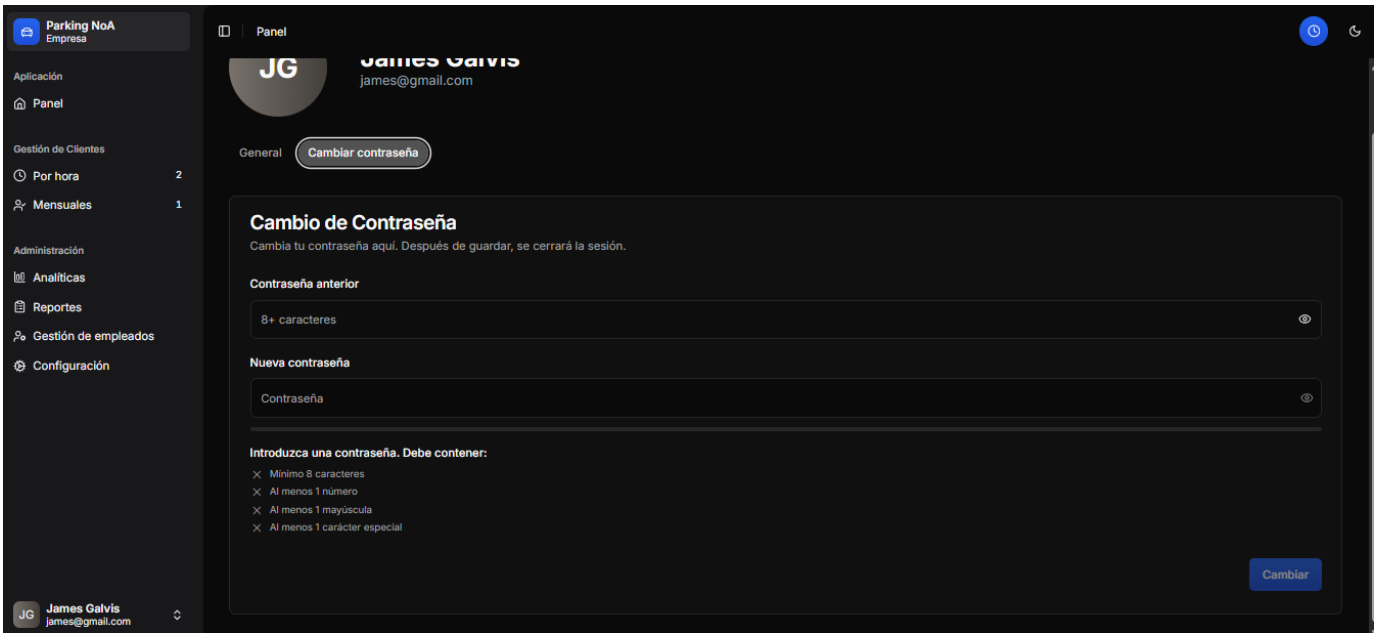
Nota: El campo "correo electrónico" no es editable, ya que solo un administrador puede cambiar el correo electrónico de un usuario desde el módulo de "Gestión de Empleados".

- **Guarda los cambios:** Una vez realizados los cambios, haz clic en el botón azul "Guardar cambios" en la parte inferior derecha. Si no deseas guardar los cambios, haz clic en "Cancelar" para descartarlos.

4.8.4. Cambiar la Contraseña

- **Accede a la pestaña de Cambio de Contraseña:** En la pantalla de "Perfil de Usuario", selecciona la pestaña "Cambiar contraseña". Esto mostrará un formulario para actualizar tu contraseña, como se observa en la Figura 47.

Figura 47. Pantalla de cambio de contraseña



Fuente: Propia del autor

- **Completa el formulario:** Rellena los siguientes campos:
    - **Contraseña anterior:** Ingresa tu contraseña actual.
    - **Nueva contraseña:** Ingresa la nueva contraseña que deseas establecer. La contraseña debe cumplir con los siguientes requisitos:

- Mínimo 8 caracteres.
- Al menos 1 número.
- Al menos 1 carácter en mayúscula.
- Al menos 1 carácter especial (por ejemplo, !, @, #).

- **Confirma el cambio:** Haz clic en el botón azul "Cambiar" para actualizar tu contraseña. Si los datos son correctos y la nueva contraseña cumple con los requisitos, el sistema actualizará tu contraseña y cerrará la sesión automáticamente para que inicies sesión con la nueva contraseña.

Nota: Si olvidas tu contraseña actual, contacta a un administrador para que la restablezca desde el módulo de "Gestión de Empleados". Además, guarda tu nueva contraseña en un lugar seguro para evitar problemas de acceso.

4.8.5. Consideraciones Generales

- **Seguridad:** Cambiar tu contraseña regularmente es una buena práctica para mantener la seguridad de tu cuenta. Asegúrate de usar una contraseña única que no utilices en otros servicios.

- **Cierre de sesión:** Después de cambiar la contraseña, el sistema cerrará tu sesión automáticamente. Deberás iniciar sesión nuevamente con tu nueva contraseña.

- **Permisos:** Los cambios en el perfil no afectan los permisos del usuario. Por ejemplo, un usuario con rol de "Empleado" no puede cambiar su rol a "Admin" desde este módulo.