U'Parking: Desarrollo de un aplicativo PWA para el control y gestión de estacionamiento en tiempo real

Kevin Stward Cárdenas Cruz

Jhon Harold Lugo Montañez

Luis Fernando Trujillo Sánchez

Corporación Universitaria Minuto de Dios

Facultad de Ingeniería

Ingeniería de Sistemas

Soacha, Cundinamarca

2023

U'Parking: Desarrollo de un aplicativo PWA para el control y gestión de estacionamiento en tiempo real

Kevin Stward Cárdenas Cruz

Jhon Harold Lugo Montañez

Luis Fernando Trujillo Sánchez

Tutor

Jasson Díaz Zamudio

Trabajo de grado para obtener el título de Ingeniero de Sistemas

Corporación Universitaria Minuto de Dios

Facultad de Ingeniería

Ingeniería de Sistemas

Soacha, Cundinamarca

2023

Agradecimientos

En primer lugar, quisiéramos agradecer a Dios, puesto que Él ha sido nuestro sostén y nuestra fuerza durante todo este camino. Su guía ha sido fundamental y nos ha permitido enfrentar cada desafío con fuerza y determinación. A nuestros padres, quienes han estado con nosotros en cada paso, demostrándonos un apoyo incondicional. Han sido pilares esenciales para nuestro crecimiento, no solo profesional, sino también personal. Su amor y aliento han sido un combustible constante en nuestro viaje. A nuestro tutor, el docente Jasson Diaz Zamudio, que con su orientación y experiencia nos ha guiado en este proceso, y su disposición para compartir su conocimiento ha sido fundamental para superar este desafío. A nuestros amigos y familiares, porque cada uno de ellos, de una manera u otra, ha contribuido a este logro. Han sido testigos de nuestra evolución y han brindado su apoyo en diferentes momentos. A la familia de Jhon Harold Lugo, que tan generosamente nos abrió las puertas de su hogar cuando lo necesitamos y siempre nos dieron de su hospitalidad. Finalmente, agradecemos a nuestros profesores y a la universidad en conjunto. Han sido fuentes invaluables de conocimiento, inspiración y orientación. Sus recomendaciones y críticas constructivas han sido fundamentales en nuestro proceso de aprendizaje y crecimiento. Este logro es el resultado de un esfuerzo colectivo, y queremos que todos los que han contribuido sepan que estamos profundamente agradecidos.

Dedicatoria

Dedico este trabajo de investigación a mis hermanos, a mis padres y a mis amigos más cercanos, quienes han sido un apoyo emocional y una pieza fundamental en mi formación profesional y personal. Agradezco su paciencia, apoyo y compromiso, así como las enriquecedoras discusiones y conversaciones que hemos mantenido a lo largo del tiempo. También quiero reconocer el compromiso y la colaboración de mis compañeros de trabajo, quienes han contribuido de manera significativa a la culminación exitosa de este proyecto. A todos ustedes, mi más sincera gratitud por haber sido parte de este importante logro en mi vida académica.

Kevin Stward Cárdenas Cruz

Dedico este trabajo a mi madre Jeimy y a mi tía Nancy, quienes han estado apoyándome y motivándome constante e incondicionalmente durante mi desarrollo tanto profesional como personal, a mis abuelitos Alfonso y Beatriz por acompañarme y darme su apoyo permanente durante todo el proceso, a mis hermanos Mateo y Michel por su apoyo emocional durante este camino y a la música por estar presente en cada instante, en cada parte de mi ser, por ser el medio para liberar el estrés y de encaminar mis pasos.

Jhon Harold Lugo Montañez

Dedico este trabajo a Dios, cuya gloria espero reflejar en mi profesión. A mis padres, Carlos y Edith, así como a mis hermanas, Candelaria y Gabriela, y a mi sobrino Julián, por estar siempre a mi lado, por su amor y apoyo continuo e incondicional. A mi querida Wendy, por su constante aliento en momentos de alegría y preocupación, es una extensión del cielo en la tierra. También a mi tío Jimmy, cuyo apoyo a distancia ha sido muy valioso en mi camino. Agradezco a mi tía Yus y a su esposo Hugo, porque me brindan su calidez. A mis abuelas, tías, familia en general, amigos y a todos los que han contribuido en este proceso. Este logro es tanto mío como suyo.

Luis Fernando Trujillo Sánchez

Hoja de Aprobación

Jurado: Rittner Eduardo Castro Galindo

Jurado: Luis Felipe Martínez Guerrero

Jurado: Andrea Catalina Saavedra Sánchez

Tutor: Jasson Díaz Zamudio

Tabla de contenido

Lista de Tablas

Lista de Figuras

Lista de Anexos

Resumen

U'Parking es una aplicación enfocada en la gestión y agilización de accesos, vehículos y elementos tecnológicos en instituciones universitarias. La aplicación permite a los usuarios ingresar mediante las credenciales ya registradas en el establecimiento, como lo es el correo electrónico institucional, también permite almacenar información sobre su ingreso y salida, vehículos y elementos tecnológicos que llevan consigo. Genera un código QR que es escaneado por el guarda de seguridad que está en la entrada y salida de la universidad. Los datos son almacenados en un servidor para tener un control de la información sobre las personas que ingresan a la institución. Los datos recopilados se utilizarán para desarrollar modelos predictivos que permitan determinar el método más eficiente de gestión del espacio. Las posibles ventajas incluyen un mejor control de acceso y un uso más eficiente del espacio en las instituciones universitarias.

Palabras clave: Análisis de datos, aplicación web progresiva, aprendizaje automático, PWA, Arquitectura de software, Back-end, Bases de datos, código QR, elementos tecnológicos, estacionamiento, Front-end, gestión de accesos, modelos predictivos, optimización de estacionamiento, parqueadero, patrones de diseño, tecnologías WEB, test, TIC, universidades, vehículos.

Abstract

U'Parking is an application focused on managing and streamlining access, vehicles, and technological elements in university institutions. The application allows users to register using their institutional email, store information about their entry and exit, vehicles, and technological elements they carry. It generates a QR code that is scanned by the security guard at the entrance and exit of the university. The data is stored on a server to have control over the information about people entering the institution. The collected data will be used to develop predictive models that determine the most efficient method of space management. Potential advantages include better access control and more efficient use of space in university institutions.

Keywords: Data analysis, progressive web application, machine learning, PWA, Software architecture, Back-end, Databases, QR code, Technological elements, Parking, Front-end, Access management, Predictive models, Parking optimization, Parking lot, Design patterns, Web technologies, Test, ICT, Universities, Vehicles.

Introducción

En la actualidad, las instituciones de educación superior experimentan un crecimiento acelerado en términos de población estudiantil y personal administrativo, lo que ha generado una mayor demanda de espacios físicos y la necesidad de optimizar su uso, principalmente en las zonas de parqueo. Esta problemática se evidencia en la Corporación Universitaria Minuto de Dios - Centro Regional Soacha, la cual enfrenta diversos desafíos relacionados con la gestión del acceso, esto se puede abordar por medio de soluciones tecnológicas que permiten mejorar la eficiencia en la gestión del acceso y el uso del espacio físico.

Los desafíos comentados, han dado lugar a problemas como la ausencia de cupos y el proceso lento y repetitivo para ingresar a las instituciones, esto genera consecuencias como que los estudiantes lleguen tarde a clase o no tengan un cupo en el estacionamiento.

Para abordar este problema, algunos estudios previos han explorado soluciones innovadoras para la gestión del estacionamiento, como la detección automática de automóviles mal estacionados (Almawgani et al., 2021a), aplicaciones de estacionamiento inteligente con reconocimiento de matrículas (Bin Mohd Nazri et al., 2020a)y sistemas de estacionamiento basados en preferencias (Mohandes et al., 2019a). Además, algunos estudios han abordado la optimización del estacionamiento en la calle y la localización de vehículos en estacionamientos interiores.

Este proyecto se enfoca en el desarrollo de una aplicación web progresiva (PWA) que permite gestionar el acceso a la institución universitaria, así como también almacenar información sobre los elementos tecnológicos y vehículos que ingresan a la universidad. Además, se pretende

utilizar los datos recolectados por la aplicación con el objetivo de mejorar la gestión del espacio físico mediante el uso de modelos predictivos.

Planteamiento del Problema

El problema que se solucionará es la mala optimización de los espacios de estacionamiento en la Universidad Minuto de Dios CRS y la gestión de cupos de este. La falta de información en tiempo real sobre la disponibilidad de espacios de estacionamiento, el proceso manual y repetitivo para registrar el acceso y la falta de control en la asignación de espacios para el aparcamiento, han generado dificultades para encontrar un lugar de estacionamiento disponible en la institución, lo que genera retrasos en el ingreso y salida de la universidad o incluso tener que buscar acceso a parqueadero en zonas cercanas a la institución. Este problema afecta a toda la comunidad universitaria, incluyendo estudiantes, docentes, administrativos, etc., los cuales se ven afectados por la falta de espacio disponible para estacionar sus vehículos.

La metodología utilizada para la recolección de datos y la validación de la información sobre la disponibilidad de espacios de estacionamiento en tiempo real implicará el conocer en cada momento la presencia o ausencia de vehículos que se encuentran en el parqueadero. La información recolectada se procesará y almacenará en una base de datos centralizada, que alimentará el modelo de análisis de datos y permitirá el entrenamiento del modelo predictivo, buscando otorgar métodos mucho más eficientes para gestionar el espacio de la universidad.

La solución propuesta implica el diseño e implementación de una aplicación PWA, que permita a los usuarios conocer en tiempo real la disponibilidad de espacios de estacionamiento y permita a la universidad gestionar de manera eficiente la asignación de espacios a los usuarios. Para lograr esto, es necesario recolectar información sobre la cantidad de cupos para cada tipo de vehículo que existe en la universidad, así como analizar los datos para determinar una correcta gestión del espacio.

Objetivos

Objetivo General

Desarrollar un aplicativo para el control de acceso y gestión de estacionamiento que optimice el uso del espacio de aparcamiento, mejore la experiencia de los usuarios y el personal de seguridad en la Corporación Universitaria Minuto de Dios CRS.

Objetivos Específicos

- Analizar los requerimientos de los usuarios y del personal de seguridad para la implementación del aplicativo de control de acceso y gestión de estacionamiento en la Corporación Universitaria Minuto de Dios CRS.
- Integrar un sistema de gestión de acceso y asignación de espacios de estacionamiento que permita optimizar la utilización de los espacios disponibles y mejorar la eficiencia en el uso del espacio de estacionamiento en la Corporación Universitaria Minuto de Dios CRS.
- Diseñar una plataforma tecnológica que permita monitorear y registrar en tiempo real el estado de los espacios de estacionamiento en la Corporación Universitaria Minuto de Dios CRS.
- Desarrollar una simulación mediante pruebas de integración y herramientas de análisis de datos para la verificación de las diferentes respuestas del sistema ante eventos en tiempo real.

Justificación

La falta de información en tiempo real sobre el estado de los espacios de estacionamiento de la Universidad Minuto de Dios CRS y la incorrecta administración de estos han generado la necesidad de encontrar una solución a tal problemática. La implementación de un sistema de monitoreo y gestión de estacionamiento es una alternativa viable, ya que permite recopilar información en tiempo real, y generar estimaciones con modelos predictivos que proporcionen formas eficientes para distribuir los espacios, todo esto usando técnicas de Machine Learning y Data Analytics. Esto tendrá un impacto central en la calidad de vida de la comunidad universitaria al reducir el tiempo de acceso al estacionamiento y mejorar la seguridad de los vehículos de los usuarios.

La correcta gestión del espacio permitiría un acceso rápido, eficiente y priorizaría el espacio en función de las necesidades concretas del momento. Por ejemplo, en ciertas horas ingresan muchas más motos que carros, principalmente en la noche, de 6:00 PM a 10:00 PM. El análisis de los datos permitirá que el sistema determine una mejor distribución del espacio.

El sistema se basará en tecnologías como HTML, CSS y TypeScript en la parte del Front-End junto con el Framework VueJS, y en el Back-End se utiliza la herramienta PocketBase y la base de datos SQLite, los cuales permiten para diseñar un API REST. Para el análisis de datos y el aprendizaje automático se pretende utilizar Python y Power BI.

La implementación de un sistema de monitoreo y gestión de estacionamiento en la Universidad es esencial para mejorar la calidad de vida de la comunidad universitaria y reducir la congestión vehicular en la zona. Con la correcta gestión del espacio y el acceso a información en tiempo real, los usuarios podrán ahorrar tiempo y esfuerzo al encontrar un espacio de

estacionamiento disponible. Además, la implementación de tecnologías de Machine Learning y Data Analytics permitirá una distribución eficiente y efectiva del espacio de estacionamiento. En conclusión, U'Parking es un proyecto necesario para mejorar la movilidad y seguridad de la comunidad universitaria.

Sub-linea del programa

U'Parking es un proyecto relevante en el campo de investigación de bases de datos, ya que implica el uso y la gestión de información en tiempo real y la interacción con una base de datos para la autenticación y el almacenamiento de información del usuario y del acceso realizado. Además, también implica la creación y gestión de relaciones entre entidades, como lo son vehículos, dispositivos y usuarios, lo cual puede ser de gran interés para investigadores que se enfocan en el modelado de datos y las relaciones entre ellos.

U'Parking aportará al desarrollo de investigación en el tema de bases de datos al aplicar conceptos de gestión de información en tiempo real, la interacción con bases de datos y la creación y gestión de relaciones entre entidades en un contexto práctico y real como lo es la gestión de acceso y el uso de espacios físicos en zonas de parqueo.

Marco Referencial

El marco referencial de esta propuesta de grado se enfoca en los principales elementos teóricos relacionados con los sistemas de estacionamiento inteligente. El uso de tecnologías como el Internet de las Cosas (IoT), la computación en la nube, la visión por computadora y el aprendizaje automático son herramientas clave en la implementación de estos sistemas. En particular, se hace énfasis en la detección de vehículos y la identificación de los lugares de estacionamiento disponibles. En este sentido, se mencionan los trabajos de Almawgani et al. y bin Mohd Nazri et al. quienes desarrollaron sistemas de reconocimiento de matrículas y aplicaciones móviles para el estacionamiento inteligente. También se destacan las investigaciones en el área de seguimiento de ocupación de estacionamiento y preferencias de estacionamiento, respectivamente(Florea et al., 2020a). En cuanto a la optimización de la asignación de espacios de estacionamiento, se hace referencia a la propuesta de un sistema de optimización dinámica de estacionamiento en la calle (Pazos et al., 2016a). Además, se menciona el trabajo de Razak et al. (2015) sobre un localizador de vehículos basado en Android.

En términos históricos, se puede señalar que los sistemas de estacionamiento inteligente han surgido como una respuesta a la creciente necesidad de solucionar los problemas de congestión vehicular y de estacionamiento en las ciudades modernas. El avance de la tecnología ha permitido el desarrollo de soluciones más sofisticadas y precisas para estos problemas, y se espera que su implementación se siga expandiendo en el futuro.

Finalmente, en cuanto al contexto de esta propuesta de grado, se destaca la creciente necesidad de soluciones innovadoras para los problemas de estacionamiento en las universidades. Se espera que los sistemas de estacionamiento inteligente sean una herramienta importante para

mejorar la eficiencia y la seguridad en la asignación de espacios de estacionamiento, lo que a su vez puede contribuir a reducir la congestión vehicular y mejorar la calidad de vida de los estudiantes, docentes y administrativos.

Revisión de la literatura

Los autores (Razak et al., 2015)se enfocan en el desarrollo de una aplicación basada en Android, la cual proporcionará servicios de navegación para ubicar vehículos estacionados en interiores de centros comerciales, usando un sensor de movimiento, la función de escáner QR y la cámara del teléfono. La aplicación podrá mostrar la ruta desde la ubicación actual del usuario hasta su vehículo estacionado en base a un mapa interior del área de estacionamiento almacenado en una base de datos, además, también es capaz de detectar automáticamente el movimiento actual del usuario en función del cálculo de pasos. Para el respectivo análisis del documento, se tendrán en cuenta puntos como: el diseño del aplicativo, sus requerimientos, pruebas realizadas y conclusiones.

Para el diseño del aplicativo, los autores (Razak et al., 2015) se basan en la arquitectura de solicitud SQL. El teléfono móvil Android está integrado con una base de datos que permite al usuario registrar la información relacionada con el mapa interior junto con las coordenadas que se obtienen por medio del sensor de movimiento, además, el dispositivo Android actuará como un cliente para mostrar información sobre la posición actual del usuario. Una vez el usuario haya descargado la aplicación, podrá usarla directamente ya que no es necesario un registro previo. Esta aplicación consta de diez archivos java, es decir, DatabaseHelper, DBhelper, IntentIntegrator, IntentResult, Location, LocationArrayAdapter, TouchImageView, SensorService, MainActivity y MainActivity2. Estos archivos java deben compilarse con SDK de Android para el archivo APK de salida, que es el archivo de instalación de esta aplicación de Android. Además, también incluye una base de datos SQL LITE única llamada TakeMeThere.db. La aplicación fue probada en un centro comercial obteniendo los siguientes resultados:

- El aplicativo puede mostrar la ruta de navegación en el mapa interior, detectar pasos o movimientos del usuario que involucran el servicio de navegación en tiempo real y almacenar la ubicación de los estacionamientos.

- La aplicación solo puede considerar los cambios de aceleración y gravedad en cuanto a la detección de los pasos del usuario.

- Cuando un usuario solicita un cambio de ruta, la aplicación no puede redirigir y volver a calcular los pasos del usuario en tiempo real.

- El mapa de interiores se diseñó con una ruta de posición fija que no puede recomendar la ruta más corta para los usuarios.

Los autores (Razak et al., 2015) concluyen que la aplicación ha cumplido con su objetivo: proporcionar un servicio de navegación para que el usuario de Android ubique su vehículo estacionado en un espacio de estacionamiento interior. Sin embargo, la aplicación puede mejorarse si se aumenta la precisión de la navegación del usuario, además de ello, es deseable un mapa interior basado en línea para recuperar la ubicación fija del vehículo y actualizar el mapa interior, lo que proporcionaría un servicio de navegación en tiempo real.

El autor (Pazos et al., 2016) resalta un problema muy recurrente que se presenta en las grandes ciudades al momento en el que los ciudadanos tratan de encontrar un lugar de estacionamiento gratuito, lo que provoca emisiones de gases de CO2 irreversibles. Como solución, plantean diseñar un software de orientación móvil llamado SMARTPARK, que ayudará a los conductores a encontrar este tipo de estacionamientos de manera eficiente, el cual se basa en

sistemas de información de estacionamiento disponibles a través de una infraestructura basada en la nube que se actualiza continuamente cada pocos segundos.

Posteriormente, se describirá el estudio del autor (Pazos et al., 2016) teniendo en cuenta sus objetivos, novedades y resultados. Este proyecto se centra en el desarrollo de una solución inteligente y eficiente para el problema mencionado en ciudades grandes y medianas, abarcando lugares de estacionamiento en las calles, estacionamientos y las áreas de estacionamiento público, por lo tanto, se desarrolló un sensor magnético, el cual estará fijo en la calle y una cámara inteligente de procesamiento de video, no obstante, se probaron prototipos de ambos dispositivos. Las bases de datos se construirán con el tiempo, lo que permitirá que los métodos de minería de datos infieran modelos de disponibilidad de estacionamiento en un futuro, que posteriormente serán utilizados por los algoritmos que alimentan la aplicación móvil. Las principales características del software son:

- Recopilación de datos y análisis del perímetro.
- Razonamiento en tiempo real para la optimización de las respuestas a las solicitudes de los usuarios finales.
- Orientación a un lugar de estacionamiento gratuito adecuado.

Como resultado, se implementó una red de sensores magnéticos y un sistema de cámara inteligente, se construyeron sistemas prototipo durante la ejecución del proyecto y se realizaron varias pruebas en ambientes reales con los prototipos demostrando su funcionalidad y desempeño. Sin embargo, aún queda abarcar inconvenientes ocasionados por el ambiente, como las pérdidas

de conexión de la red de nodos de los sensores y las sombras que ocasionan problemas en el sistema de imágenes integradas.

El enfoque principal de los autores (Mohandes et al., 2019) es la implementación de un sistema de estacionamiento en interiores basado en preferencias con usuarios recurrentes, como empresas, entidades gubernamentales, campus universitarios, etc. Para el análisis del estudio, se tendrán en cuenta dos opciones tomadas por los autores para detectar plazas de estacionamiento: una se basa en sensores ultrasónicos y la segunda utiliza un conjunto de cámaras, se analizarán los respectivos resultados experimentales y las conclusiones generales.

Los autores (Mohandes et al., 2019) se centran en la comparación de las 2 opciones anteriormente mencionadas, con el objetivo de elegir el sistema más apropiado para el campus universitario en el que se encuentran realizando el desarrollo. En primer lugar, se explica el funcionamiento del software basado en sensores, los cuales detectan las vacantes de espacio de estacionamiento, indicando por LED rojo para 'ocupado' y LED verde para 'vacante', por lo tanto, informan de su estado a la unidad de control central de manera continua. Cabe destacar que, en experimentos iniciales con estos dispositivos se notó un problema de corrupción de datos, esto provocó que el programa mostrara una cantidad errónea de los lugares disponibles para aparcar. Como solución a este problema, se implementó una configuración de manera rotativa, es decir, cuando un sensor completa su transmisión interrumpe el siguiente circuito para transmitir sus datos. Una de las desventajas de la mencionada implementación es el retraso en la actualización del estado de estacionamiento como consecuencia de la configuración en serie. Debido al número limitado de lugares para automóviles en el proyecto, el retraso en el peor de los casos para mostrar el estado real fue de solo unos segundos, pero este retraso puede ser significativo en amplios lugares de estacionamiento. Sin embargo, esto se puede solucionar mediante el método de

implementación ‘dúplex’ en el circuito de detección del estacionamiento y en el sistema de control central.

En segundo lugar, se explica el funcionamiento del software basado en cámaras, los cuales se utilizan generalmente en áreas abiertas con espacios de estacionamiento de alta densidad. Los autores (Mohandes et al., 2019) en el proceso de investigación, notaron que la mayoría de los métodos basados en cámaras se centran en el grosor de los bordes o en los atributos de la textura de las imágenes captadas, por lo que se presentó un enfoque híbrido entre los métodos mencionados utilizando un marco bayesiano general para clasificar las características fusionadas como espacios de estacionamiento vacíos u ocupados. La primera etapa del sistema se encarga de la extracción de bordes y texturas de las imágenes captadas de cada uno de los lugares de estacionamiento, esta etapa de extracción de características está precedida por un paso de calibración en el que los diferentes puntos del automóvil cubiertos por cada una de las cámaras se identifican en la imagen mediante un conjunto de formas cuadriláteras. La primera característica extraída representa el número de objetos detectados en los espacios de estacionamiento individuales, el resultado de estas operaciones es una imagen en escala de grises con un fondo negro y diferentes tonos de blanco alrededor de los bordes de los diferentes objetos de la imagen, los cuales son un elemento primordial para la detección del estado del estacionamiento.

Como resultado, los autores (Mohandes et al., 2019) llevaron a cabo un análisis de rendimiento exhaustivo que mostró que la implementación basada en sensores supera a la basada en cámaras para aplicaciones en interiores con una precisión del 100 y el 98 %, respectivamente.

Los autores (Florea et al., 2020) se enfocan en el desarrollo de un software para la gestión de estacionamientos que facilita la reserva de cupo, la apertura de la barrera y permite la entrada, salida y revisión del número de plazas vacantes y la sincronización con la aplicación web y la base de datos de apoyo. Primeramente, se darán a conocer las funcionalidades de la aplicación, las cuales son:

- Visualización del estado del estacionamiento en tiempo real.
- Reserva en un horario específico por número de placa.
- Módulo de administración que incluye sistema de pago y actualizaciones sobre usuarios y precios.
- Implementación del concepto de gamificación en la gestión de estacionamientos.

El concepto de “Gamificación” se introduce con el fin de que los conductores estimen una hora exacta en la que su coche estará estacionado en ese lugar, por lo tanto, este concepto tiene como objetivo la reserva de lugares de estacionamiento por un lapso lo más preciso posible, con el fin de gestionar mejor las plazas de aparcamiento libres. En segundo lugar, se hará un breve análisis de la arquitectura del proyecto teniendo en cuenta las herramientas, y el entorno de programación utilizados. El sistema integrado consta de los siguientes componentes principales:

- Componente de hardware que contiene un ordenador low post Raspberry Pi Model 3B con dos PiCam, una cámara Hikvision, motor paso a paso y algunos LED.
- Aplicación web ASP .NET desarrollada con controles DevExpress. La estructura del sitio web consta de varios módulos que se comunican entre sí para una mejor funcionalidad de

la aplicación. Message Queuing Telemetry Transport (MQTT) y Node-Red se utilizan para la comunicación entre la aplicación web y Raspberry Pi.

- Base de datos SQL, la cual registra en la base de datos todas las reservas, información de usuarios, precios, historial y el número de plazas de aparcamiento libres.

- Herramientas de software: Motor de reconocimiento óptico de caracteres Tesseract OCR, biblioteca Open CV, Leptonica para el procesamiento de las imágenes.

La solución se puso a prueba en la Universidad Lucian Blaga de Sibiu (LBUS) Rumania, como resultado, los autores (Florea et al., 2020) indican que la solución desarrollada es flexible, extensible y aplicable a ciudades universitarias abarrotadas, pero también a otras organizaciones privadas que tienen espacios de estacionamiento operados de manera ineficiente.

Los autores (Bin Mohd Nazri et al., 2020) se enfocan en un problema relacionado con el control del tráfico y la identificación de vehículos debido al creciente número de los mencionados en la carretera, por ello, se plantea un proyecto basado en un sistema de puerta/barrera inteligente de bajo costo utilizando la programación Raspberry Pi y Node-RED que utiliza un método de reconocimiento para determinar el número de placa de la parte delantera del automóvil cada vez que el automóvil se acerca a la puerta de entrada. Se utiliza una cámara para capturar la imagen de la parte delantera del coche. Para que la puerta permita el acceso al vehículo, el número de placa debe estar registrado y guardado en la base de datos. Una vez verificado, el administrador o el guardia de seguridad abrirá la puerta presionando un botón en una aplicación para permitir que el automóvil pase la puerta con facilidad sin necesidad de una tarjeta especial para abrir la puerta. Los objetivos del proyecto son desarrollar un sistema de matrículas de automóviles que detecte el número de matrícula del automóvil utilizando un método de detección de reconocimiento,

desarrollar una aplicación de estacionamiento para el pago de tarifas de estacionamiento y, por último, controlar el acceso a la puerta mediante tecnología Bluetooth.

En primer lugar, los autores (Bin Mohd Nazri et al., 2020) plantean que, para el desarrollo del sistema propuesto, la metodología que se utilizará es el modelo de Desarrollo Rápido de Aplicaciones (RAD). El aspecto principal de este modelo es asegurar el éxito de los prototipos desarrollados que sean operativos y reutilizables. Hay cuatro fases que se encuentran dentro de esta metodología, que son la planificación de requisitos, el diseño del sistema, el desarrollo y, por último, la transición.

Debido a que este no es un sistema ya implementado, los autores (Bin Mohd Nazri et al., 2020) plantean que el objetivo principal de este proyecto es detectar el número de placa del automóvil utilizando la tecnología de Reconocimiento Automático de Placa de Matrículas (ALPR) y proporcionar una aplicación para que el administrador controle la puerta desde lejos sin tener que sentarse justo al lado de la puerta. Aparte de eso, el otro objetivo principal del sistema es crear una plataforma para que los usuarios paguen las tarifas del estacionamiento a través de una aplicación en línea para teléfonos inteligentes. La tecnología de procesamiento de imágenes Raspberry Pi se puede aplicar a este proyecto. Una de las tecnologías es el reconocimiento facial que se puede utilizar para proporcionar más seguridad al sistema. Con el reconocimiento facial, el sistema puede realizar un seguimiento y registrar los rostros de los visitantes y almacenarlos en la base de datos.

Los autores (Almawgani et al., 2021) presentan un sistema de detección de estacionamiento inteligente y eficiente que consta de dos cámaras conectadas a un sistema de movimiento diseñado por Arduino, cuatro motores de corriente continua y un sensor infrarrojo pasivo, los cuales se

colocan estratégicamente para monitorear los espacios de estacionamiento, especialmente las líneas rectangulares de pintura en cada estacionamiento interior. El sistema de monitoreo de movimiento responde automáticamente a cualquier movimiento que detecta, como vehículos en un estacionamiento a lo largo de una fila de estacionamientos. Una vez detectadas, las imágenes capturadas se procesan utilizando el software MATLAB. Cualquier coche mal aparcado es detectado y las cámaras identifican su matrícula, que queda registrada en la base de datos. En el proceso de implementación se probaron prototipos de diseño del sistema propuesto en cinco supuestos casos. En cada caso, se procesan diez imágenes, lo que da como resultado 50 imágenes. De estas 50 imágenes, 48 imágenes corresponden a detecciones correctas, las otras dos imágenes corresponden a detecciones falsas. Por lo tanto, el sistema de monitoreo de estacionamiento inteligente propuesto tiene una eficiencia del 96%. El sistema proporciona una solución adecuada para ayudar al conductor a aparcar y el propietario administra adecuadamente el estacionamiento a través del sistema de monitoreo remoto.

Los autores (Levkivskyi et al., 2022) proponen la implementación de un sistema de reconocimiento de lugares de parqueo disponibles en ciudades ucranianas. Dado que el flujo vehicular crece constantemente, encontrar un espacio en áreas con mucha afluencia de gente es un problema común. La implementación de este sistema de reconocimiento de espacios de estacionamiento disponibles permitiría a los usuarios verificar si el lugar de estacionamiento especificado cuenta con espacios disponibles, además, podrán elegir la ubicación exacta dentro del estacionamiento donde se encuentran dichos espacios. El propósito principal de la implementación del sistema es minimizar los tiempos de búsqueda de un lugar de aparcamiento, por lo que se decide implementar un chatbot para mensajería con el servicio Telegram, el sitio web para la

gestión del sistema, un submódulo de búsqueda de parqueaderos disponibles el cual trabaja en tiempo real y cámaras de video tipo IP, es decir, con acceso a internet.

Como resultado, los autores (Levkivskyi et al., 2022) implementaron un chatbot en Telegram Messenger, el cual ofrece a los usuarios funciones como la tramitación de solicitudes de apartamiento, listas de estacionamientos disponibles, procesamiento de datos y búsqueda de estacionamiento en tiempo real. Además, se desarrolló un script para reconocer un lugar disponible para estacionamiento de automóviles, en el que se demostró que el uso de la red neuronal Mask R-CNN es el más adecuado para la tarea enfocada al reconocimiento de automóviles a partir del flujo de video de una cámara de seguridad en tiempo real, obteniendo como resultado experimental un equilibrio entre productividad y precisión del 0,7.

Los autores (Aditya et al., 2023) proponen el desarrollo de un sistema de estacionamiento inteligente (IPS), ya que los habitantes de ciudades con alta densidad de población tardan en encontrar un espacio de estacionamiento más cercano debido al creciente número de vehículos, lo que también afecta la seguridad de los mismos. Este sistema consiste en un marco de IoT que recopila datos en tiempo real, los envía a la nube, sugiere al usuario un lugar de aparcamiento cercano disponible, permite a los usuarios comprobar la disponibilidad de plazas de aparcamiento y finalmente, permite la reserva de aquellos espacios disponibles para aparcar. Para la implementación del sistema, los autores (Aditya et al., 2023) utilizan una computadora de bajo costo llamada Raspberry, un firmware de código abierto llamado NodeMCU, basado en Lua para la obtención de la información del chip wifi ESP8266, tecnología de Identificación por radiofrecuencia y sensores infrarrojos para la detección de los espacios disponibles.

Como resultado, tras realizar la implementación del software, los autores (Aditya et al., 2023) concluyen que el IPS propuesto es correcto y eficiente. Además, proponen una mejora agregando la función de dirección de ruta, la cual se mostrará al usuario dirigiéndose al espacio disponible y también asegurándose de que el espacio permanezca reservado hasta que el usuario llegue a la ubicación.

El enfoque principal de los autores (Harshavardhanan et al., 2022) se centra en áreas de estacionamiento completamente automatizadas, por lo que proponen la implementación de un sistema basado en IoT (Internet of Things), el cual transmite información en tiempo real sobre espacios de estacionamiento disponibles a través de una aplicación web/móvil. Este sistema utiliza sensores infrarrojo y cámaras de circuito cerrado de televisión (CCTV), las cuales indican la existencia de un auto en determinado espacio asignado. Cuando el cliente ingresa al aplicativo debe ingresar con su número de matrícula, posteriormente, el cliente reserva el espacio disponible y el sistema se encarga de crear una Contraseña de Único Uso (OTP), la cual el aplicativo asocia con el número de matrícula del vehículo.

Los autores (Harshavardhanan, P et al., 2022) mencionan que cuando llega un vehículo al establecimiento es captado por un sensor IR el cual alerta al usuario correspondiente para que ingrese la OTP a través de la aplicación, posteriormente, la cámara se encarga de tomar una foto instantánea y la envía a Matlab en el servidor para el procesamiento de esta. Luego de determinadas acciones de procesamiento de imágenes, en el momento en que Matlab reconoce la matrícula del vehículo, este se encarga de generar un archivo de texto con la información necesaria para realizar las debidas comprobaciones y almacenar en la base de datos.

Debido al creciente número de vehículos en las ciudades, la congestión del tráfico urbano es cada vez más común, los autores (Koumetio Tekouabou et al., 2022) tienen en cuenta estudios que han demostrado que los conductores en búsqueda de espacios de estacionamiento disponibles, contribuyen hasta en un 30% a la congestión del tráfico en la ciudad de Birmingham – Inglaterra. Por ello, se propone la integración de un sistema basado en IoT, el cual se compone de una arquitectura en la que se plantea un nuevo método de predicción basado en modelos de conjunto y, utilizando una combinación entre ciudades inteligentes, IoT y aprendizaje automático. Este sistema de estacionamiento inteligente utiliza sensores infrarrojos con el objetivo de definir la tasa de ocupación del estacionamiento, ayuda al conductor a estacionar con seguridad y le informa sobre la disponibilidad de plazas de estacionamiento más cercanas disponibles.

Como resultado final, los autores (Koumetio Tekouabou et al., 2022) propusieron un sistema que integra IoT y modelos de regresión de conjuntos. El IoT permite la explotación de todos los objetos conectados en relación con los estacionamientos inteligentes para la recopilación de los datos, su análisis y compartir los resultados con los conductores. El modelo basado en conjuntos para el análisis predictivo propuesto por los autores optimizó la predicción de la disponibilidad de plazas de estacionamiento en aparcamientos inteligentes.

Los autores (Veeramanickam et al., 2022) plantean un sistema de estacionamiento automatizado con el propósito de detectar los lugares de estacionamiento disponibles en un establecimiento con ayuda de sensores IR y microcontroladores encargados del procesamiento de la señal según su modelo de funcionamiento. El estacionamiento inteligente opera con recolección de datos de ingreso y salida en tiempo real para la función de reserva por orden prioritario First Come, First Served (FCFS).

Como resultado, los autores (Veeramanickam et al., 2022) diseñan un prototipo de implementación de un sistema inteligente real, enfocado en la recopilación de los datos y la reserva de espacios de estacionamiento disponibles, este se encarga de la producción de múltiples escenarios de patrones de espacios de estacionamiento dependiendo de la demanda de los usuarios. Este sistema analiza aproximadamente 180 escenarios diferentes para la asignación de estacionamientos con el tipo de reserva FCFS enfocada en los usuarios finales.

Los autores (Errousso et al., 2020) se centran en la implementación de un sistema inteligente de gestión de aparcamientos urbanos capaz de reconocer, comprender, comunicarse con su entorno, aprender y tomar decisiones Este predice, en tiempo real, la ocupación de los estacionamientos de la ciudad para cambiar su estado dinámicamente de un área de entrega a un lugar común y viceversa. El objetivo principal del aplicativo web, cuyo desarrollo está enfocado en Python, es asignar los lugares más adecuados, en términos de ubicación y distancia a pie, a los repartidores sin penalizar a otros usuarios. El aplicativo permite reservar de forma interactiva uno o varios lugares, finalizar un recorrido de entrega debidamente preestablecido y la construcción del itinerario a seguir.

Según los autores (Errousso et al., 2020) la ciudad de Melbourne – Australia ha instalado aproximadamente 4300 sensores que cubren 303 segmentos de calles en 35 áreas del distrito central de negocios, los cuales registran los tiempos de entrada y salida de los vehículos. Luego de varias acciones correctivas aplicadas a los datos históricos tomados del año 2017, los autores concluyen que el método de bosque aleatorio es el indicado debido a sus excelentes resultados y su corto tiempo de ejecución. La eficiencia y rendimiento del sistema implementado, se encuentra condicionado por la capacidad de los conductores para respetar sus reservas y llegar en el tiempo

indicado al lugar der estacionamiento, por ello, es fundamental reforzar la propuesta con servicios adicionales como: listas negras, imposición de multas y generación de alertas.

Marco teórico

Para el proyecto U'Parking, se aplicará la arquitectura de software “Clean Architecture”, la cual se basa en la separación de la aplicación en tres capas principales: la capa de dominio, la capa de aplicación y la capa de infraestructura. En la capa de dominio se definirán las reglas de negocio y se implementarán los casos de uso. En la capa de aplicación, se implementarán los adaptadores para los casos de uso definidos en la capa de dominio y se coordinarán las diferentes tareas necesarias para cumplir con los casos de uso. Por último, en la capa de infraestructura, se implementarán los detalles técnicos de la aplicación, como la persistencia de datos o la comunicación con servicios externos (Martin, 2018).

Para asegurar la calidad del código y reducir los errores, se aplicará el desarrollo dirigido por pruebas (TDD), el cual consiste en escribir pruebas automatizadas antes de escribir el código de la funcionalidad correspondiente. Esto asegura que el código esté correctamente diseñado, sea fácilmente mantenible y cumpla con las especificaciones requeridas. (Alexander Shvets, 2019)

En cuanto a la base de datos, se utilizará PocketBase, que utiliza SQLite y permite crear una base de datos local de forma sencilla. Se implementarán los modelos de bases de datos necesarios, asegurando la integridad y consistencia de los datos, y se definirán las relaciones entre ellos. (PocketBase, 2022)

Para el Front-end se utilizarán tecnologías como HTML, CSS, TypeScript, Vue y Vite. Vue es un framework progresivo de JavaScript que permite construir interfaces de usuario de manera sencilla y efectiva. Vite, por su parte, es una herramienta de construcción de proyectos que permite un desarrollo rápido y eficiente de aplicaciones web.

En cuanto al Back-end, se utilizará BaaS (Back-end as a service) con PocketBase, lo que permite la implementación de la lógica de la aplicación en un servidor externo, reduciendo la carga en los dispositivos de los usuarios y mejorando la seguridad. Además, se utilizará JSON para la comunicación entre la capa de aplicación y la capa de infraestructura. (PocketBase, 2022)

Para la gestión del proyecto es importante hacer uso de la metodología Scrum, la cual es una metodología ágil que se enfoca en la entrega iterativa e incremental de un producto de calidad. Según el autor (Martin, 2002) Scrum permite optimizar el valor del negocio y la satisfacción del cliente mediante la entrega temprana y continua de software funcional. Scrum funciona mediante el uso de sprints, que son iteraciones de tiempo fijo (generalmente de dos a cuatro semanas) en las que se planifica, se diseña, se construye, se prueba y se entrega un incremento del producto.

En conclusión, la aplicación de estos referentes teóricos permitirá un desarrollo eficiente y de calidad para el proyecto U'Parking, asegurando la separación de responsabilidades en la aplicación, la calidad del código y la eficiencia en el proceso de desarrollo, lo que resultará en una aplicación robusta y de calidad para los usuarios.

Marco Metodológico

El marco metodológico del proyecto U'Parking se basa en el enfoque exploratorio y la aplicación de las metodologías ágiles, particularmente Scrum, complementado con los principios y prácticas propuestos por Robert C. Martin en sus libros "Agile Software Development: Principles, Patterns, and Practices" (Martin, 2002) y "Clean Architecture: A Craftsman's Guide to Software Structure and Design" (Martin, 2018). Esta elección metodológica se fundamenta en la necesidad de adaptabilidad, entrega temprana y continua de funcionalidades, así como en la búsqueda de un diseño de software limpio, modular y mantenible.

Para el enfoque observatorio, se parte una entrevista realizada a uno de los vigilantes del parqueadero de la Corporación Universitaria Minuto de Dios CRS, en ella se identificaron algunas limitaciones en el proceso actual de registro de vehículos. De este proceso se destacan los siguientes aspectos:

- El registro de ingreso y salida al parqueadero se lleva a cabo de forma manual, lo que dificulta obtener datos precisos sobre el ingreso de motocicletas y automóviles al estacionamiento.
- Es importante destacar que el entrevistado manifiesta que el proceso de registro del vehículo y sus datos ha dejado de hacerse desde hace aproximadamente un año. En este momento ese proceso de registro solo se hace para los visitantes que ingresan por medio de algún vehículo a la institución.
- Duranta la mañana el flujo de vehículos que ingresan a la institución es considerablemente menor al de la noche, el vigilante manifiesta que entre las 5:30 PM y las 7:00 PM es el rango horario donde se observa un mayor ingreso de vehículos.

- Se evidenció la necesidad por pare del entrevistado de contar con un sistema de control y gestión de estacionamiento que garantice la seguridad y mejore la experiencia tanto para los usuarios como para el personal de seguridad encargado del parqueadero.

En base a la información proporcionada, se puede argumentar que actualmente no se cuenta con datos reales del ingreso de motos y carros al estacionamiento de la universidad debido a la falta de un sistema de registro adecuado. El proceso de ingreso actual se basa principalmente en la verificación manual de placas y tarjetas de propiedad, lo que dificulta llevar un registro preciso y detallado de todos los vehículos que ingresan al parqueadero.

Esta falta de datos de registro presenta varios inconvenientes y limitaciones. Por un lado, dificulta la capacidad de realizar un seguimiento preciso de la ocupación del parqueadero y determinar la disponibilidad de espacios en tiempo real. Esto puede resultar en una asignación ineficiente de espacios, congestión y dificultad para encontrar lugares de estacionamiento disponibles. Además, la falta de un sistema de registro adecuado dificulta la identificación de vehículos que ingresan sin autorización o la detección de situaciones anómalas.

Para abordar este problema y mejorar la eficiencia en la gestión del estacionamiento, el proyecto U'Parking propone la implementación de un aplicativo que permita el registro automatizado de vehículos y motos, facilitando el seguimiento y control de su ingreso y salida del parqueadero. A través de la simulación y pruebas en entornos controlados, se busca evaluar la eficiencia y viabilidad del aplicativo en términos de optimización de espacios, agilidad en el proceso de ingreso y mejora en la experiencia tanto para los usuarios como para el personal de seguridad encargado del parqueadero.

El proyecto U'Parking se basa en la observación de la problemática actual, respaldada por la encuesta realizada, donde se evidencia la necesidad de contar con un sistema de registro y control adecuado que mejore la experiencia de los usuarios y optimice el uso del espacio de estacionamiento. La falta de datos de registro y el proceso engorroso de ingreso de vehículos a la universidad son factores que afectan la eficiencia y la seguridad en el parqueadero.

Mediante la implementación del aplicativo propuesto, se espera obtener beneficios significativos, como una asignación más eficiente de espacios, reducción de la congestión y un mayor control de acceso al parqueadero. Además, el aplicativo permitirá generar informes y estadísticas sobre el uso del estacionamiento, lo que facilitará la toma de decisiones y la planificación de mejoras en el futuro.

Para desarrollar este aplicativo, se ha adoptado la metodología Scrum como enfoque principal de gestión del proyecto. Por ello es importante profundizar en los beneficios de esta:

> Scrum es una metodología ágil que es ampliamente utilizada en el desarrollo de software que se caracteriza por su enfoque iterativo e incremental. En Scrum, los proyectos se dividen en iteraciones llamadas sprints, y se realizan entregas frecuentes de funcionalidades (Martin, 2002).

Al usar esta metodología en el proyecto, es posible hacer uso de un enfoque iterativo e incremental en el desarrollo del sistema. En U’Parking se realizaron entregas frecuentes de funcionalidades, permitiendo la adaptación y retroalimentación continua.

> El desarrollo iterativo e incremental es una práctica fundamental en el concepto de las metodologías ágiles. En lugar de tratar de entregar todo el software al final del proyecto,

se enfoca en poder realizar entregas frecuentes y obtener una retroalimentación temprana del cliente (Martin, 2002).

En esta metodología es importante tener en cuenta los siguientes aspectos:

- Diseño limpio y modular: Se aplicaron los principios y patrones de diseño propuestos por Robert C. Martin para lograr un diseño de software limpio, modular y mantenible. Se siguieron los principios SOLID y se fomentó la refactorización constante para mejorar la calidad del código por medio del desarrollo de software TDD (Test Driven Development).

  Los principios SOLID promueven un diseño de software limpio y modular, lo que facilita su mantenibilidad y adaptabilidad. Estos principios incluyen la responsabilidad única, la apertura a la extensión, pero cerrada a la modificación, la sustitución de Liskov, la segregación de interfaces y la inversión de dependencias (Martin, 2018).

- Prácticas de desarrollo ágil: Se utilizaron prácticas ágiles, como el Desarrollo Dirigido por Pruebas (TDD), donde se escribieron pruebas unitarias antes de desarrollar el código, y se realizaron pruebas continuas y automatizadas para garantizar la calidad del software.

  El Desarrollo Dirigido por Pruebas (TDD) es una práctica fundamental en las metodologías ágiles que consiste en escribir pruebas unitarias antes de desarrollar el código. Esto ayuda a garantizar que el software cumpla con los requisitos y funcione correctamente (Martin, 2002).

Conforme a la metodología Scrum y el enfoque de Desarrollo Dirigido por Pruebas (TDD), se han establecido las siguientes fases de desarrollo para el proyecto U'Parking:

- Inicio: En esta fase, se realiza la planificación inicial del proyecto y se definen los objetivos generales. También se identifican las funcionalidades clave que se abordarán en cada etapa del proyecto.

- Análisis: En esta fase, se realiza un análisis detallado de los requisitos del proyecto. Se identifican las necesidades de los usuarios y se desglosan en tareas más pequeñas y manejables.

- Diseño y construcción de pruebas: En esta fase, se diseñan y se crean las pruebas para asegurar que el software cumpla con los requisitos establecidos. Estas pruebas ayudan a guiar el desarrollo del software y garantizar su calidad.

- Ejecución de pruebas: Una vez que las pruebas han sido diseñadas y construidas, se ejecutan para verificar el correcto funcionamiento del software. Si alguna prueba no pasa, se realizan las correcciones necesarias en el código.

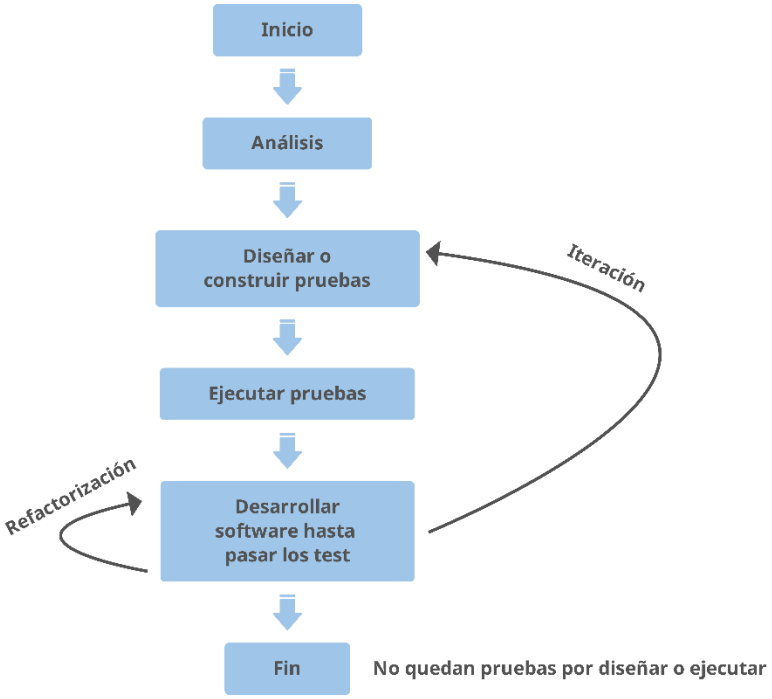
- Desarrollo del software y superación de pruebas: En esta fase, se desarrolla el software en base a los requisitos establecidos y se realiza la codificación necesaria para que las pruebas sean superadas satisfactoriamente.

- Finalización: Al finalizar el desarrollo de una funcionalidad, se realiza una revisión para asegurarse de que cumple con los requisitos establecidos. Se integra con el resto del sistema y se verifica su funcionamiento en conjunto.

Estas fases se repiten en ciclos cortos llamados "sprints", que tienen una duración fija. Al final de cada sprint, se muestra el progreso logrado y se realizan ajustes y mejoras para el

siguiente sprint. El flujo de trabajo de estas fases se representa en la siguiente imagen. Ver figura 1.

Figura 1. Fases de desarrollo



Fuente: Construcción del autor

La metodología Scrum aportó al desarrollo del proyecto U'Parking al permitir una entrega temprana y continua de funcionalidades, lo que permitirá que el equipo de desarrollo pueda recibir retroalimentación temprana. Esto significa que los requisitos pueden ser ajustados de forma rápida y eficiente a medida que se desarrolla el producto, lo que resulta en un producto final que es más cercano a las necesidades reales del usuario. La metodología es valiosa en el proyecto U'Parking, ya que permitió una entrega temprana y continua de funcionalidades, lo que concluyó en un producto final de mayor calidad y más cercano a las necesidades reales del proyecto.

Sistema de Variables

El sistema de variables del proyecto U'Parking abarca una amplia gama de aspectos cruciales relacionados con la gestión y optimización de accesos y espacios de estacionamiento en instituciones universitarias. Estas variables desempeñan un papel fundamental en el aseguramiento de un funcionamiento eficiente de la aplicación y en la mejora continua de la experiencia de los usuarios.

El sistema de variables se divide en diferentes categorías, comenzando por las variables de acceso que permiten la identificación y autenticación de los usuarios. Asimismo, se consideran variables de información del usuario, se contemplan variables específicas relacionadas con la gestión del espacio de estacionamiento y se integran variables de análisis de datos y modelos predictivos, con el objetivo de optimizar la asignación de espacios y mejorar la eficiencia en la gestión del estacionamiento. Al tener en cuenta esta amplia variedad de variables, U'Parking adopta un enfoque integral que contribuye a una gestión más eficaz y a un uso más eficiente del espacio de estacionamiento en las instituciones universitarias, beneficiando tanto a los usuarios como al personal administrativo. A continuación, se detalla la descripción de cada una de estas variables.

Variables de Acceso:

- Usuario
- Contraseña
- Código QR

Variables de Información del Usuario:

- Nombre
- Vehículos
- Elementos Tecnológicos

Variables de Gestión del Espacio de Estacionamiento:

- Espacios Disponibles
- Espacios Ocupados
- Tipos de Vehículos
- Cupos por Tipo de Vehículo
- Estacionamiento Dinámico

Variables de Análisis de Datos y Modelos Predictivos:

- Datos de Acceso
- Datos de Utilización de Espacios
- Modelos Predictivos

Operacionalización de Variables

Variable: Espacios Disponibles

- Definición: Número total de espacios de estacionamiento disponibles en la universidad.
- Operacionalización: Se obtendrá y actualizará automáticamente mediante el monitoreo en tiempo real del sistema de estacionamiento, que contará los espacios disponibles.

Variable: Espacios Ocupados

- Definición: Número de espacios de estacionamiento ocupados en un momento dado.
- Operacionalización: Se obtendrá y actualizará automáticamente mediante el monitoreo en tiempo real del sistema de estacionamiento, que contará los espacios ocupados.

Variable: Cupos por Tipo de Vehículo

- Definición: Cantidad de espacios de estacionamiento asignados para cada tipo de vehículo.
- Operacionalización: Se establecerá la cantidad de cupos asignados para cada categoría de vehículo y se actualizará automáticamente cuando los espacios sean ocupados o liberados.

Variable: Tiempo de respuesta del sistema

- Definición: Rapidez con la que el sistema de U'Parking puede procesar una solicitud y proporcionar una respuesta.
- Operacionalización: Se mide en milisegundos (ms), con mediciones menores indicando una respuesta más rápida del sistema.

Variable: Tiempos de entrada y salida

- Definición: Cantidad de tiempo que le toma a un vehículo entrar o salir del estacionamiento utilizando el sistema U'Parking.

- Operacionalización: Se mide en minutos, donde valores menores representan un tiempo de entrada o salida más rápido.

Variable: Capacidad de vehículos que se benefician con U'Parking

- Definición: Cantidad de vehículos que pueden utilizar eficientemente el sistema U'Parking en un determinado período de tiempo.

- Operacionalización: Se mide en número de vehículos por período de tiempo determinado (por ejemplo, por día).

Variable: Tipo de vehículo

- Definición: Categoría del vehículo que utiliza el sistema U'Parking (por ejemplo, carro, moto, bicicleta).

- Operacionalización: Se cuantifica para determinar la proporción de cada tipo de vehículo que utiliza el sistema, midiendo la cantidad total de cada tipo de vehículo que usa el sistema en un período de tiempo determinado.

Metodología de la Investigación

Para el desarrollo del presente trabajo de grado, y con el objetivo de generar una investigación sólida, se estructuró la ecuación de búsqueda teniendo en cuenta definiciones relacionadas al tema como: “Software”, “Data Base”, “Park” y “Camera”, acompañadas de palabras similares y/o sinónimos obtenidos mediante el diccionario online Thesaurus. A continuación, la fórmula de búsqueda:

(Software OR "computational package" OR program OR "informatic package" OR app OR application) AND (database OR db) AND (park* OR "parking lot" OR "parking spot" OR "parking place") AND (camera OR "security camera")

La consulta de investigación se realizó en las bases de datos Scopus, Science Direct y ProQuest, en las que se utilizaron filtros de búsqueda relacionados con el año de la publicación del documento, idioma, área de estudio y orden por relevancia. Para ello se tuvieron en cuenta artículos de revisión, artículos científicos, libros y documentos de conferencia publicados a partir del año 2015 en adelante, en el idioma inglés, en las áreas de ingeniería y ciencias de la computación. Posteriormente, ya establecidos estos filtros de búsqueda, se realizó la selección de los documentos teniendo en cuenta aspectos como: su relación con la gestión y optimización de parqueaderos mediante el uso de aplicaciones (Web, móviles o de escritorio), la veracidad de la información, la claridad del contenido y la pertinencia de estos, por lo que se alcanzaron 31 resultados en la base de datos Scopus, 27 resultados en ProQuest y 143 resultados en Science Direct. En el proceso de análisis de los documentos, tras aplicar los filtros correspondientes, se obtuvo un total de 17 documentos los cuales cumplen con los criterios.

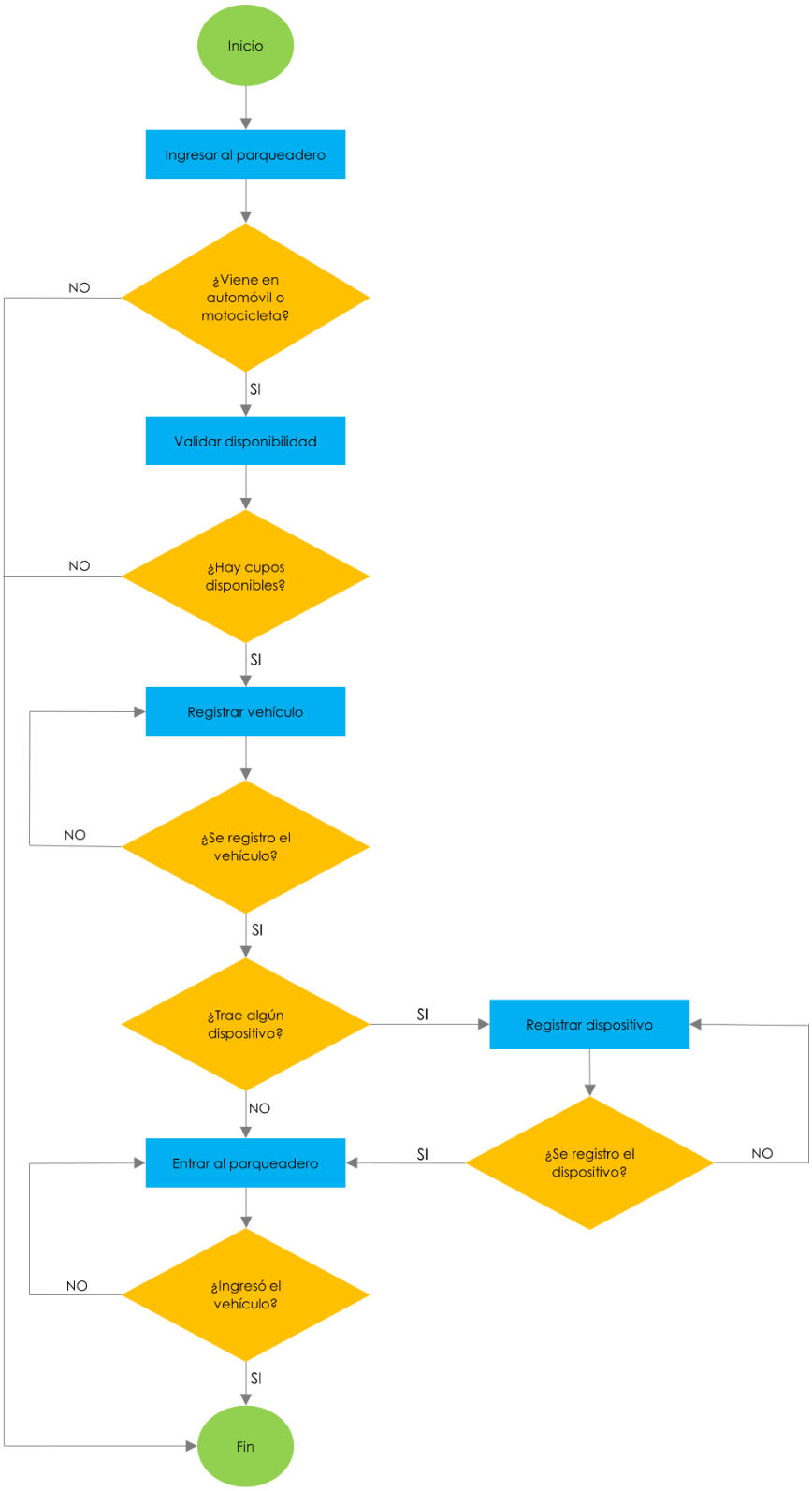
Procedimiento

Para el desarrollo del proyecto se utilizará un enfoque ágil por medio de la metodología Scrum, la cual consiste en un modelo iterativo y la colaboración entre el equipo de trabajo. El tipo de investigación será cuantitativa, ya que se realizará recolección de información por medio de encuestas, que servirán como soporte de que el proyecto es una necesidad por satisfacer. El paso a paso para desarrollar el proyecto es el siguiente:

Interacciones del sistema

Antes de iniciar el proceso de desarrollo del proyecto, se realizará un diseño detallado del sistema. Actualmente el proceso para ingresar a la institución es de la siguiente manera:

Figura 2. Flujo actual para el ingreso al estacionamiento



Fuente: Construcción del autor.

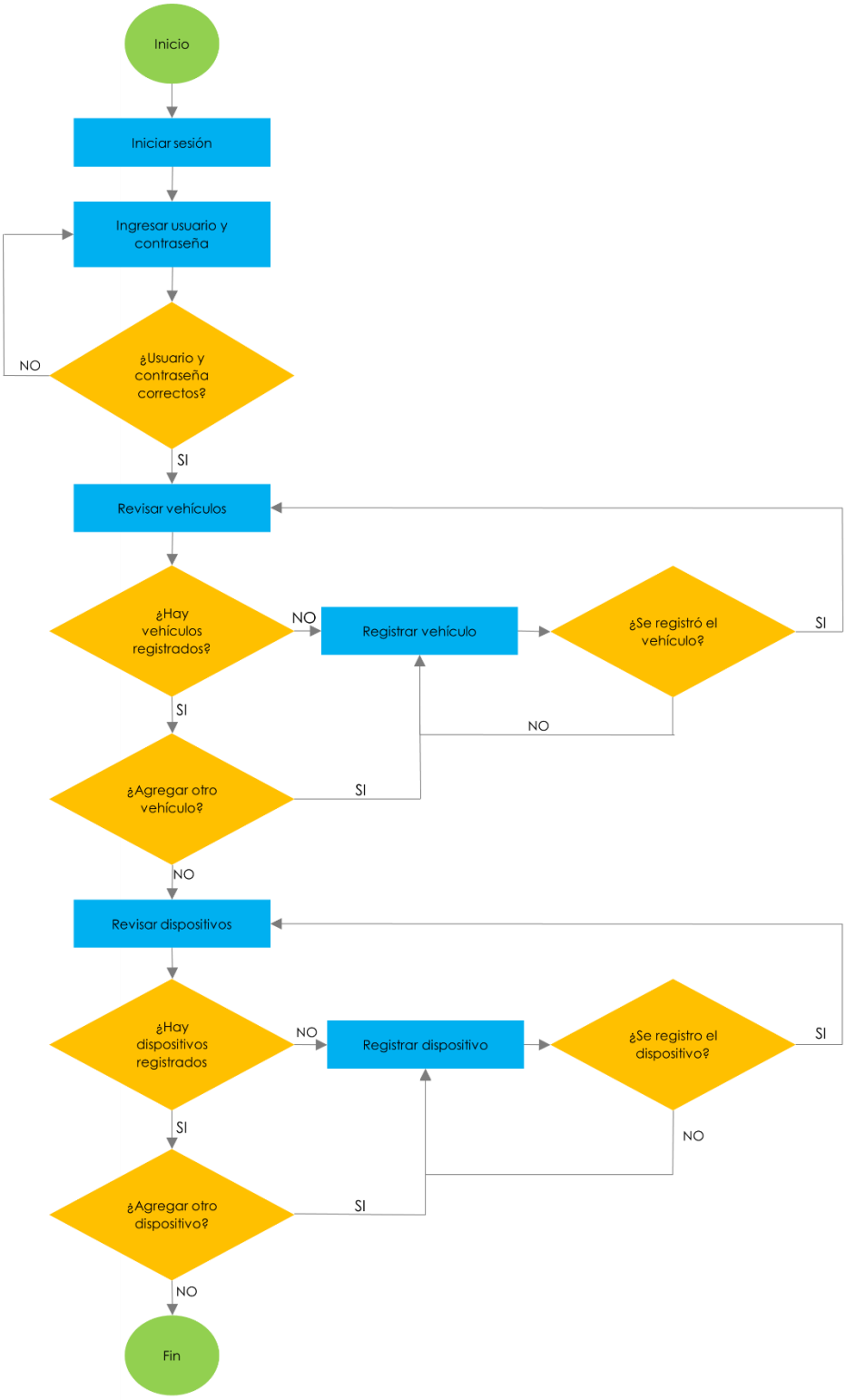
En la *figura 2* se puede evidenciar un desglose meticuloso del proceso de ingreso al estacionamiento. Todo comienza con la verificación del tipo de vehículo que se intenta ingresar, centrándose específicamente en automóviles y motocicletas. Si el usuario no llega en uno de estos dos, el proceso culmina en este punto.

Si, por el contrario, el usuario llega en moto o carro, se procede a una segunda verificación: la disponibilidad de espacios de estacionamiento. En caso de falta de disponibilidad, el proceso se interrumpe. Pero si hay espacios disponibles, se avanza a registrar la información relevante del vehículo.

Posteriormente, se le consulta al usuario si posee algún dispositivo de importancia, como una computadora, Tablet, cámara, etc. En caso de tener alguno, se registra según corresponda. Si no se presenta tal dispositivo, el usuario puede avanzar directamente hacia la zona de estacionamiento. Finalmente, el proceso se completa una vez que el vehículo está estacionado adecuadamente.

Por otro lado, el siguiente diagrama ilustra el proceso de interacción entre el usuario y la aplicación:

Figura 3. Flujo de interacción entre usuario y app



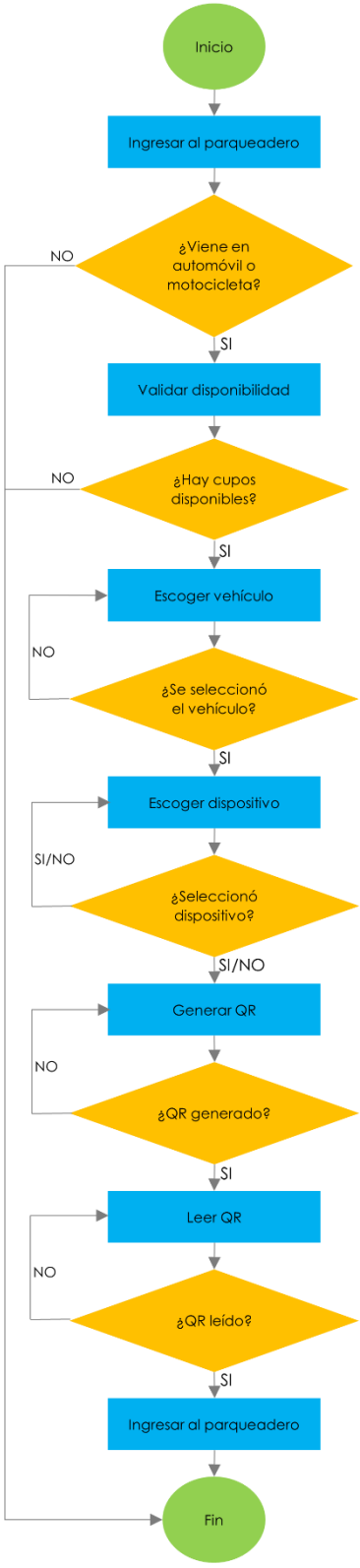
Fuente: Construcción del autor

Este proceso inicia con la autenticación del usuario, durante la cual se solicita ingresar las credenciales correspondientes: nombre de usuario y contraseña. En caso de ingresar datos incorrectos, se permitirá al usuario intentar de nuevo, hasta que los datos sean válidos o hasta que se alcance un límite predefinido de intentos fallidos. Una vez autenticado correctamente, el usuario accede a la vista principal, donde podrá verificar si tiene vehículos registrados. En caso de no tener ningún vehículo asociado a su perfil, deberá agregar al menos uno. El paso siguiente ofrece la posibilidad de agregar nuevos vehículos, un proceso que puede repetirse tantas veces como el usuario desee, dependiendo del número de vehículos que quiera registrar.

Posteriormente, el usuario tiene la opción de vincular dispositivos, aunque este es un paso opcional y puede ser adaptado según las necesidades individuales. Si el usuario decide agregar un dispositivo, podrá repetir este proceso según la cantidad de dispositivos que desee asociar. El ciclo de interacción con la aplicación concluye una vez que se han completado estos pasos. La secuencia descrita proporciona una estructura flexible que se ajusta a las preferencias individuales de cada usuario, permitiendo una experiencia de usuario personalizada y eficiente.

Adicional, entra en acción un tercer diagrama, el cual describe el procedimiento para acceder al estacionamiento a través del aplicativo. Ver Figura 4.

Figura 4. Flujo del ingreso al estacionamiento a través del app



Fuente: Construcción del autor

El primer requisito es verificar que el usuario se presente en un tipo de vehículo que está permitido en el estacionamiento. Si el vehículo no cumple con las especificaciones requeridas, el proceso se detiene aquí.

Si el vehículo es permitido, el siguiente paso es revisar si hay espacios disponibles en el estacionamiento. En caso de que este se encuentre lleno, lamentablemente, no será posible admitir más vehículos y el proceso culmina.

Una vez verificado que hay disponibilidad, el usuario tiene la opción de seleccionar el tipo de vehículo que va a ingresar al estacionamiento. Es importante aclarar que este es un paso crucial y si el usuario no completa esta acción, no podrá avanzar en el proceso.

A continuación, el sistema brinda al usuario la posibilidad de escoger el dispositivo a ingresar. Este paso es opcional y no impide avanzar en el proceso. Como un plus, el usuario tiene la flexibilidad de añadir más de un dispositivo si lo desea.

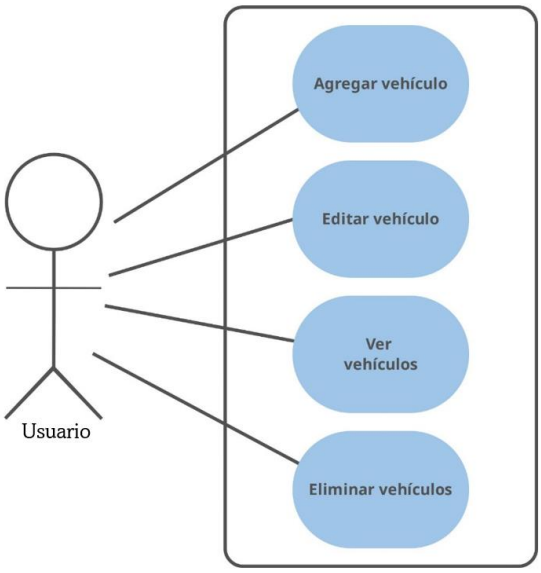
Cuando se ha determinado tanto el vehículo como el dispositivo, el sistema genera automáticamente un código QR. Aunque en este paso el usuario no tiene que realizar ninguna acción, sí es importante que verifique que el código QR ha sido generado y está disponible para su uso.

El último paso antes de ingresar al estacionamiento es leer el código QR generado. Este es el final del procedimiento y una vez que se ha completado con éxito, el usuario está autorizado para ingresar al estacionamiento.

Diagramas de casos de uso

Los casos de uso de la aplicación para entender el alcance y la funcionalidad del software desarrollado, los casos de uso desempeñan un papel esencial en la definición de las interacciones entre los usuarios y el sistema. Estos diagramas ayudan a visualizar el comportamiento esperado del software, al describir claramente las posibles acciones que un usuario puede realizar y cómo el sistema responde a dichas acciones. Este enfoque centrado en el usuario garantiza que el software cumpla con las expectativas y necesidades del usuario final, proporcionando una base sólida para el diseño, implementación y prueba del software. A continuación, se explorarán en detalle los casos de uso que han guiado el desarrollo del sistema, proporcionando una visión más profunda de las funciones y capacidades del software:

Figura 5. Caso de uso usuario (vehículos)



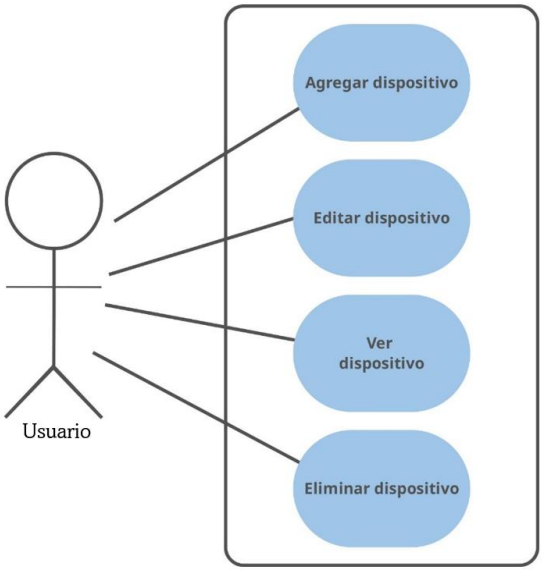
Fuente: Construcción del autor.

- Agregar vehículo: En esta interacción, el usuario tiene la capacidad de agregar un vehículo a su perfil. Esto puede implicar la introducción de información relevante sobre el vehículo,

como su tipo (carro o moto), matrícula, modelo, entre otros. Una vez que el usuario ha proporcionado toda la información necesaria, el sistema guarda estos datos.

- Editar vehículo: Esta función permite al usuario modificar la información de los vehículos que ya ha agregado. Por ejemplo, si han cambiado ciertas características del vehículo o si se cometió un error al agregar la información inicialmente. Luego de realizar los cambios, el sistema actualiza la información del vehículo en la base de datos.
- Ver vehículos: Aquí, el usuario tiene la posibilidad de ver una lista de todos los vehículos que ha registrado en el sistema. Esta función puede ser útil para llevar un registro de los vehículos que ha agregado, o si necesita información específica de alguno de ellos.
- Eliminar vehículos: En algunos casos, el usuario puede necesitar remover un vehículo de su perfil, tal vez porque ya no lo posee o porque ya no lo utilizará para estacionar. En esta interacción, el usuario selecciona el vehículo que desea eliminar y el sistema procede a eliminarlo de la base de datos.

Figura 6. Caso de uso usuario (dispositivos)

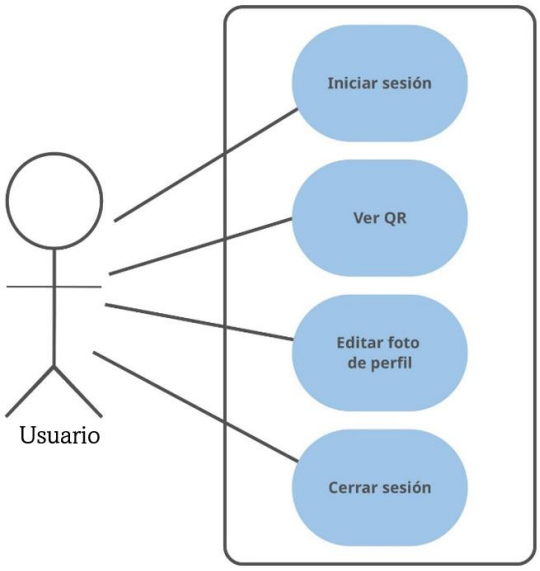


Fuente: Construcción del autor

- Agregar dispositivo: Esta interacción permite al usuario registrar un nuevo dispositivo en el sistema. Normalmente, esto implicaría proporcionar detalles sobre el dispositivo, como el tipo, la marca, el modelo, y cualquier otro dato relevante. Una vez que estos detalles son suministrados y validados, el dispositivo se agrega a la lista de dispositivos gestionados por el usuario en el sistema.
- Editar dispositivo: En esta interacción, el usuario puede modificar la información de un dispositivo que ya ha sido registrado en el sistema. Esto puede incluir cambios en los detalles suministrados durante la creación del dispositivo, como su tipo, marca, modelo, entre otros. Esta función es especialmente útil si los detalles del dispositivo cambian con el tiempo o si se cometió un error durante la etapa de registro.
- Ver dispositivo: Esta interacción permite al usuario visualizar los detalles de un dispositivo específico. Este podría ser un resumen de la información del dispositivo, como su tipo, marca, modelo, y cualquier otro dato que haya sido registrado en el sistema. Es una función útil para cuando el usuario desea confirmar detalles del dispositivo o cuando necesita recordar cierta información del mismo.
- Eliminar dispositivo: Con esta interacción, el usuario puede remover un dispositivo del sistema. Normalmente, esto implica confirmar que realmente desea eliminar el dispositivo, ya que esta acción puede ser irreversible. Esta función es útil cuando un dispositivo ya no está en uso o ya no está en posesión del usuario.

Los actores involucrados en estos los diagramas de casos de uso de las *figuras 5 y 6* serían el Usuario y el Sistema. El Usuario es quien realiza las acciones de agregar, editar, ver y eliminar dispositivos, mientras que el Sistema es la entidad que facilita estas interacciones, validando la información suministrada, guardando los cambios y proporcionando la información solicitada.

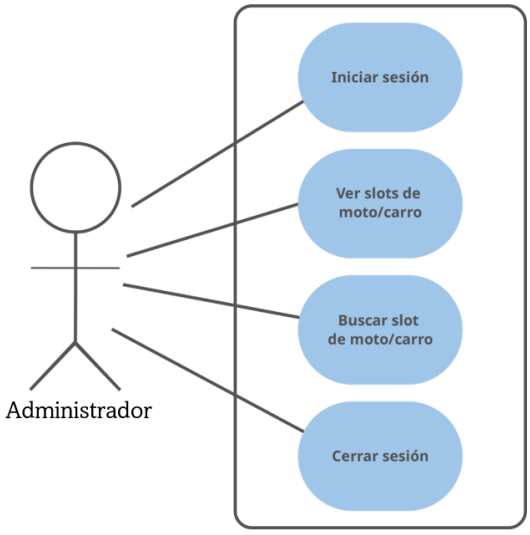
Figura 7. Caso de uso usuario y app



Fuente: Construcción del autor

- Iniciar sesión: Esta es la primera interacción que el usuario debe realizar. Para ello, debe proporcionar sus credenciales de inicio de sesión (como el nombre de usuario y contraseña). Si las credenciales son correctas, el usuario puede acceder a su cuenta.
- Ver QR: Una vez que el usuario ha iniciado sesión, puede tener la necesidad de ver el código QR asociado a su cuenta o vehículo. Este código QR puede ser útil para facilitar ciertas operaciones, como el acceso al estacionamiento.
- Editar foto de perfil: Esta función permite al usuario personalizar su perfil mediante la actualización de su foto de perfil. El usuario selecciona una nueva imagen, la carga al sistema y esta se establece como su foto de perfil actual.
- Cerrar sesión: Cuando el usuario haya terminado de usar el sistema, puede elegir cerrar la sesión. Esta acción garantiza que la cuenta del usuario no esté abierta y disponible para otros, lo que contribuye a mantener la seguridad de la cuenta.

Figura 8. Caso de uso administrador



Fuente: Construcción del autor

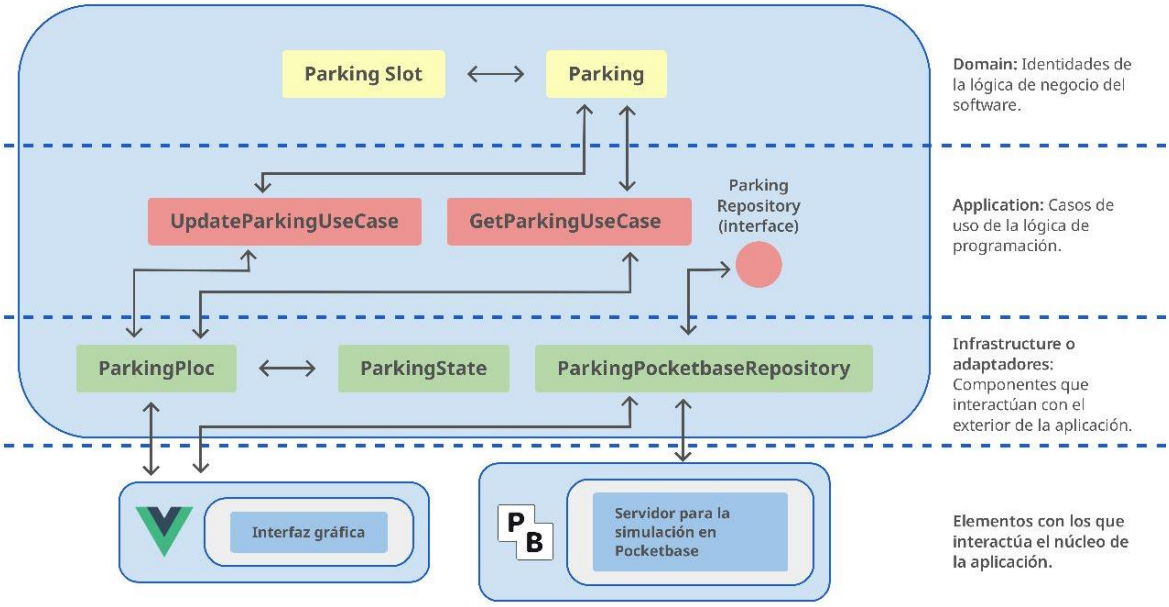
- Iniciar sesión: Esta es la primera interacción que el administrador debe realizar. Aquí, el administrador proporciona sus credenciales (usualmente un nombre de usuario y contraseña) para acceder al sistema. El sistema valida estas credenciales y, si son correctas, permite al administrador acceder a su cuenta.
- Ver Slots de moto o carro: Una vez que el administrador ha iniciado sesión, puede realizar varias funciones, una de las cuales es ver los espacios disponibles para estacionar motos o carros. Esto le permite al administrador tener una visión general de la ocupación del parqueadero y le ayuda a gestionar eficientemente los espacios disponibles.
- Buscar Slots de moto o carro: Esta interacción es similar a la anterior, pero permite al administrador buscar específicamente un espacio de estacionamiento, ya sea para una moto o un carro. Esta función es útil cuando el administrador necesita encontrar un espacio en particular, por ejemplo, para asignarlo a un usuario específico o para verificar su estado.

- Cerrar sesión: Esta es la última interacción que un administrador puede realizar. Al cerrar sesión, el administrador indica que ha terminado con sus actividades y que ya no necesita acceso al sistema. Este es un paso importante por razones de seguridad, ya que asegura que nadie más puede usar la cuenta del administrador después de que él ha terminado de usarla.

Diagrama de módulos del software

Antes de examinar los siguientes diagramas, es fundamental tener en cuenta que en una arquitectura limpia las capas operan de manera estratificada, con restricciones precisas en cuanto a las interacciones permitidas. La capa de infraestructura o adaptadores, situada en el nivel más externo, solo puede comunicarse con los adaptadores definidos dentro de la arquitectura. A su vez, esta capa tiene acceso exclusivo a la capa de aplicación, utilizando los casos de uso correspondientes. Por último, el dominio se encuentra encapsulado, sin interconexiones con las demás capas. Esta estructura garantiza la preservación y la independencia de la lógica de negocio, que constituye el núcleo fundamental y menos sujeto a cambios del software. En el contexto del proyecto U'Parking, se han establecido los siguientes diagramas de módulos de software, los cuales se alinean con las dos interfaces de la aplicación:

Figura 9. Diagrama de módulos de vista parqueadero

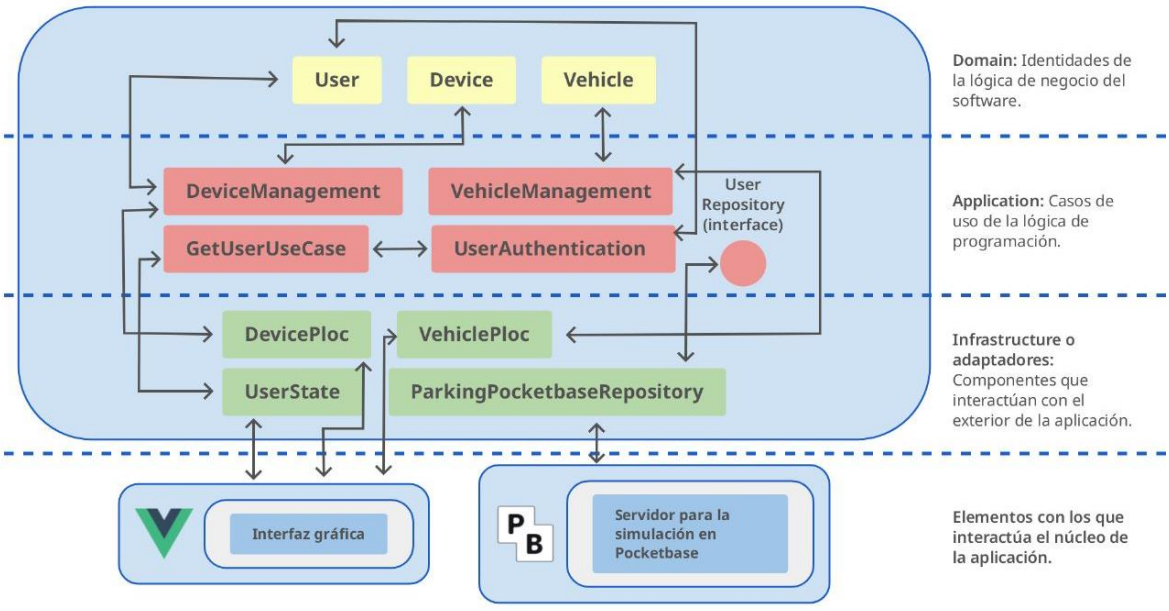


Fuente: Construcción del autor.

La *figura 9* representa los módulos de la vista del parqueadero en el contexto del proyecto U'Parking. En este se destacan las interacciones entre las diversas clases de software, así como su distribución en capas según los principios de la arquitectura limpia. En el dominio, se definen las clases "Parking" y "ParkingSlot", encargadas de establecer la estructura de datos necesaria para almacenar y gestionar cada espacio de estacionamiento en la institución. El objeto "Parking", que recopila todos los "ParkingSlots", sirve como punto de partida para los casos de uso, los cuales definen la forma en que se interactúa con las entidades del software y forman parte de la capa de aplicación. Dado que la cantidad de parqueaderos es estática, los casos de uso se limitan a obtener y actualizar los datos, considerando previamente qué atributos de cada espacio de estacionamiento son modificables, una decisión que se establece durante la creación de la estructura de "ParkingSlot".

En la capa de infraestructura, se implementan las clases concretas que cumplen la interfaz "ParkingRepository", permitiendo así la traducción de la información enviada o recibida desde el servidor de Pocketbase. También se construye en esta capa el objeto "ParkingBloc", encargado de gestionar el estado de los datos. Este enfoque no solo facilita el acceso a la información, sino que también separa la gestión del estado de la interfaz gráfica, reduciendo así la dependencia del framework Vue utilizado para la interfaz. Como se puede apreciar, la única pieza capaz de interactuar con el entorno externo es la capa de infraestructura, que no solo facilita la conexión con los componentes externos mencionados anteriormente, sino que también permite una escalabilidad más sencilla y modular de estas funcionalidades.

Figura 10. Diagrama de módulos de vista usuario



Fuente: Construcción del autor

Para el segundo componente de software, que se muestra en la figura 10, se sigue aplicando la arquitectura limpia, adaptada a las necesidades y requisitos de los clientes. En el dominio, se definen las entidades "User", "Vehicle" y "Device", las cuales permiten determinar los vehículos y dispositivos asociados a cada usuario, así como verificar su estado de activación. En la capa de

aplicación, se introduce un ligero cambio en comparación con el diagrama anterior. En este caso, en lugar de separar cada acción relacionada con una entidad en una clase independiente, se agrupan todas estas opciones en una misma clase. Esta decisión se basa en que la lógica de estos casos de uso es relativamente sencilla, por lo que no es necesario dividirla en varios archivos. Para el manejo de estado, se emplean adaptadores en la capa de infraestructura, así como la implementación concreta de cada interfaz de repositorio construida.

Esta estructura de diseño permite una organización más cohesiva y eficiente de las funcionalidades relacionadas con la gestión de usuarios, vehículos y dispositivos. Al agrupar las acciones en una sola clase en la capa de aplicación, se simplifica el desarrollo y mantenimiento del código, ya que las interacciones y dependencias se encuentran encapsuladas en un único componente. Asimismo, el uso de adaptadores en la capa de infraestructura asegura una integración adecuada con los sistemas externos y la persistencia de los datos, garantizando así un funcionamiento fluido y confiable del software.

Además, esta arquitectura limpia adaptada permite una mayor flexibilidad y escalabilidad del sistema. Al mantener una clara separación entre las diferentes capas y componentes, se facilita la incorporación de nuevas funcionalidades, así como la modificación o extensión de las existentes. Por ejemplo, en el futuro se podría agregar la opción de gestionar permisos de acceso o incorporar nuevas entidades relacionadas con el uso de los vehículos en el parqueadero. Gracias a esta estructura modular y bien definida, se pueden realizar cambios y mejoras en el software de manera más eficiente, minimizando el impacto en otras partes del sistema.

Diseño del software

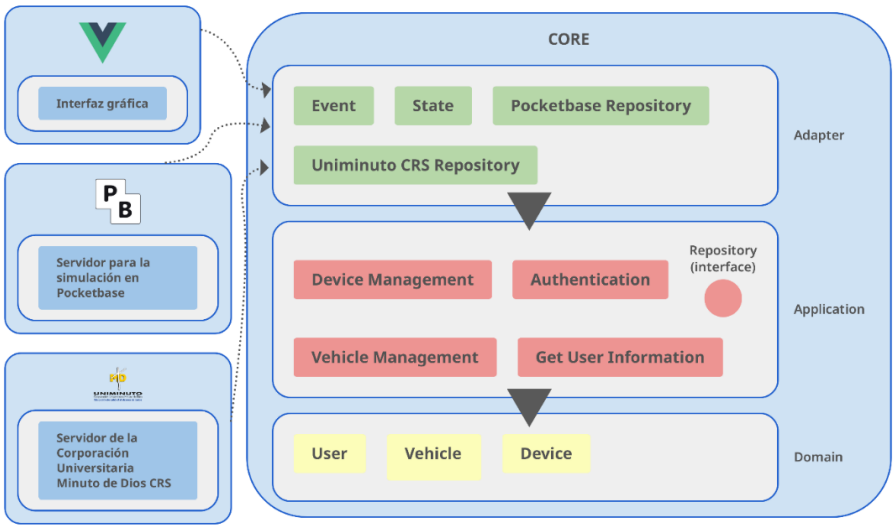
El diseño del software es un componente crítico en el desarrollo de cualquier sistema informático. Esta etapa implica trazar un plan detallado que ilustra cómo las funcionalidades y requisitos identificados se transformarán en un software funcional.

En este punto es muy importante resaltar la arquitectura de software. Esta etapa es fundamental, ya que establece la estructura y el comportamiento del sistema, así como la organización y las interrelaciones de sus componentes. La arquitectura de software influye en la escalabilidad, la mantenibilidad, la seguridad y la eficiencia del sistema. Por lo tanto, es crucial realizar una arquitectura sólida y bien planificada. Omitir este proceso puede resultar en un producto con dificultades para realizar modificaciones y mantener el software a lo largo del tiempo. Durante la fase de diseño de la arquitectura, se identificarán los componentes clave del sistema, se definirán las interfaces entre ellos y se establecerán los patrones de comunicación y flujo de datos. Además, se evaluarán las tecnologías y herramientas adecuadas para implementar la arquitectura propuesta. Este enfoque proactivo en el diseño de la arquitectura permitirá un desarrollo más eficiente y un sistema más robusto y escalable en el proyecto U'Parking. (Martin, 2018).

Para el diseño de U'Parking, se ha optado por seguir una arquitectura limpia y modular. Se enfoca en la capacidad de dividir los componentes en piezas con dependencias lineales y bien definidas. Esto es de gran importancia, ya que permite la flexibilidad y escalabilidad del sistema. Además, esta arquitectura facilita la sustitución de módulos individuales sin necesidad de modificar el resto del sistema. Al tener una estructura modular, se logra una mayor facilidad en el mantenimiento y la incorporación de nuevas funcionalidades, lo que contribuye a un sistema más robusto y adaptable a futuras necesidades. En cuanto a la simulación, se utilizará una API REST

construida en PocketBase. Sin embargo, es importante destacar que, en un entorno de producción, es probable que esta configuración varíe, ya que se requerirá conectar el sistema a los servidores de la Corporación Universitaria Minuto de Dios CRS. En el siguiente diagrama de distribución por capas y componentes del sistema se puede apreciar con mayor detalle. El núcleo de la aplicación, conocido como "Core", alberga la lógica de negocio, los casos de uso y los componentes que permiten adaptar el núcleo a distintas interfaces gráficas o a posibles cambios en la API REST a la cual se conectará. Esta flexibilidad nos brinda la capacidad de ajustar el código base según las necesidades específicas del proceso, asegurando así una mayor adaptabilidad y eficiencia en el sistema. Ver figura 11.

Figura 11. Diagrama de Arquitectura de Software



Fuente: Construcción del autor.

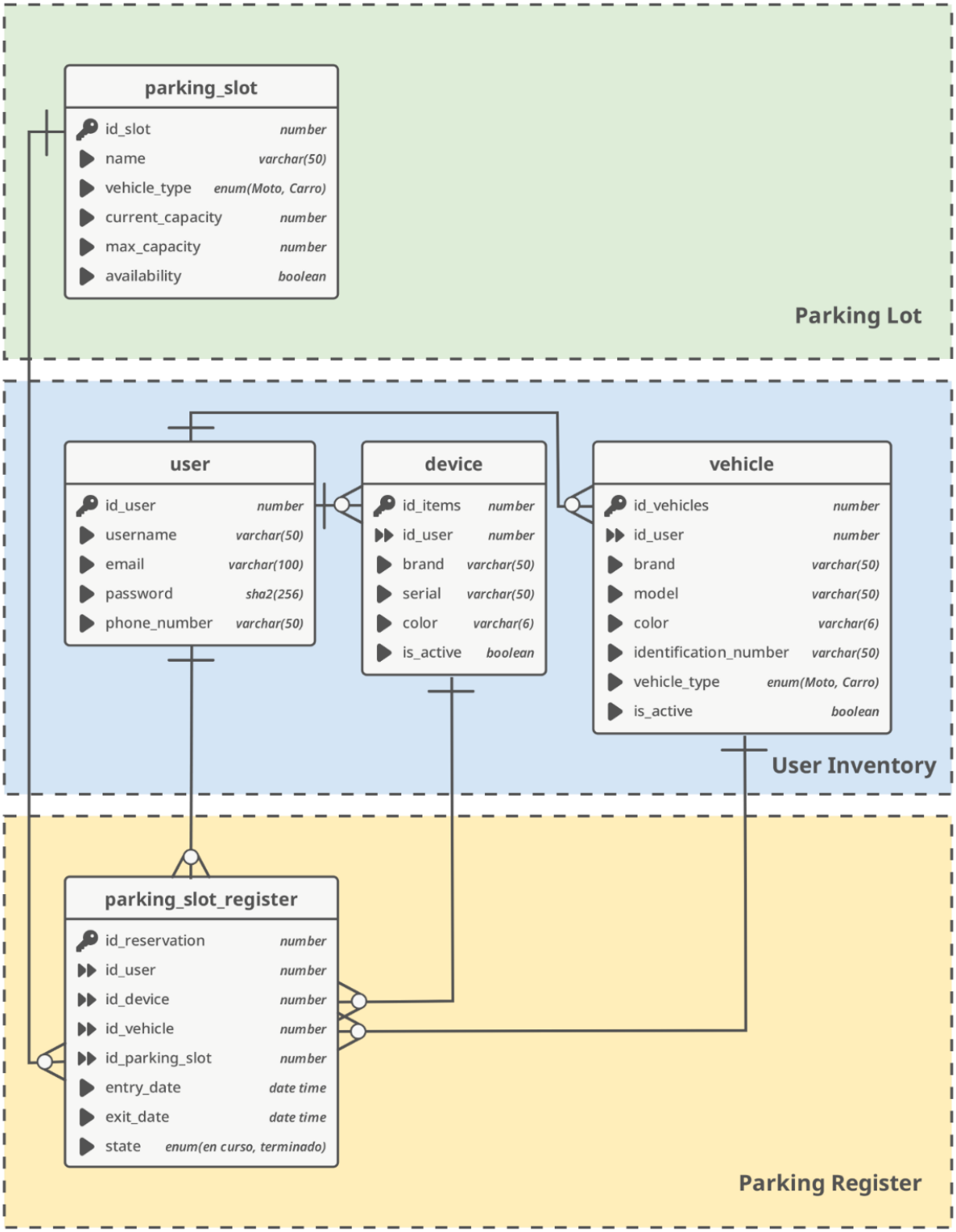
Es fundamental tener en cuenta que el mantenimiento de una base de código de software es esencial para el éxito a largo plazo de cualquier proyecto de desarrollo de software. Seguir las prácticas y principios de código limpio es crucial para mantener la calidad y la legibilidad del código a lo largo del tiempo (Robert C. Martin, 2008).

En cuanto a la base de datos, esta se ha estructurado siguiendo una arquitectura en capas, con el objetivo de lograr la máxima independencia entre ellas. Esta decisión permite ampliar las capacidades de una capa sin necesidad de realizar cambios significativos en las demás capas. Ver figura 12.

Además, se ha diseñado la estructura de la base de datos de manera que sea fácil acceder y utilizar la información almacenada para el proceso de análisis que se detallará más adelante.

También es importante tener en cuenta que únicamente los administradores de la base de datos tienen autorización para modificar y adaptar los espacios de estacionamiento. Esta restricción se implementa con el propósito de evitar conflictos y asegurar la integridad de la base de datos. La herramienta desarrollada en Python, que cuenta con acceso de administrador a la base de datos, se encargará de llevar a cabo estas modificaciones. Su función principal será analizar los datos y realizar las modificaciones necesarias en los espacios de estacionamiento, ajustándolos en función de las necesidades específicas de cada momento.

Figura 12. Diagrama de base de datos



Fuente: Construcción del autor.

Implementación del software

Durante la implementación, es crucial considerar las diversas herramientas e interfaces que interactúan entre sí en torno a la API REST y la base de datos. No solo es importante definir estas entidades, sino también comprender cómo se comunican con las demás. Las entidades principales son las siguientes:

- Usuario: El usuario interactúa con la API REST a través de la PWA construida en Vue. De esta manera, se realizan las solicitudes y se obtienen los resultados, que se presentan de forma intuitiva y organizada en la interfaz gráfica.

- Visor de Parqueadero: Este proporcionará a los administradores una vista instantánea de la ocupación y las operaciones del parqueadero. No es simplemente una representación estática del parqueadero, sino una herramienta dinámica que muestra la adaptabilidad del sistema U'Parking en respuesta a las fluctuaciones de demanda de estacionamiento. Según nuestra investigación de campo, en un espacio destinado a automóviles se pueden estacionar hasta tres motocicletas sin dificultar el movimiento de estas. Esta estrategia nos permite aumentar la disponibilidad de espacios para motocicletas durante las horas en las que el flujo de estas es considerablemente mayor que el de automóviles. Es importante tener en cuenta que los espacios designados para motocicletas no son aptos para estacionar automóviles, debido principalmente a las limitaciones físicas del espacio.

- Sistema interno: Este núcleo es la base principal del proyecto y se encarga de gestionar la base de datos, así como de devolver información a las diferentes entidades que interactúan con él. Es importante destacar que esta entidad permite la conexión con servicios externos, como APIs, con el fin de obtener información de los usuarios de otros servicios de la

Corporación Universitaria Minuto de Dios CRS. Esto amplía el sistema con U'Parking. Es relevante tener en cuenta que esta integración no se realizó en la simulación, sin embargo, la arquitectura de software del proyecto permite realizar estos ajustes sin mayores complicaciones.

- Generador de consultas aleatorias: Esta entidad es exclusiva del entorno de pruebas y no se presenta como tal en un entorno de producción. Su función principal es simular las interacciones del cliente con la API REST, lo que permite estudiar el comportamiento del sistema en diferentes escenarios. Para lograr un estado aleatorio dentro del generador, se propone consolidar una base de datos con registros falsos generados mediante la librería Faker de Python. Estos registros se utilizan para simular diversas acciones que se pueden realizar en el sistema, como la habilitación de un usuario para acceder al estacionamiento, o su salida de este. Es importante destacar que aquí se generan consultas que permiten comprobar casos excepcionales que pueden ocurrir en el sistema. Este aspecto se abordará con más detalle más adelante.

- Simulador: El proceso de simulación es una etapa crítica para demostrar la viabilidad y eficacia de U'Parking. Este componente del proyecto implica la creación de un visor de información basado en web que refleje en tiempo real la ocupación y las operaciones del sistema de gestión de estacionamiento utilizando Vue.js y TypeScript.

    - Visor de parqueadero: Será una aplicación web creada con Vue.js y TypeScript, dos tecnologías modernas que permiten una interfaz de usuario fluida y segura. Además de su utilidad para observar el comportamiento del sistema en tiempo real, el visor también proporciona una valiosa fuente de datos para la evaluación del

desempeño del sistema. Los patrones y tendencias que se pueden discernir a través de la visualización de la simulación pueden ser de gran utilidad para identificar áreas de mejora y ajustar la estrategia de gestión de estacionamiento según sea necesario.

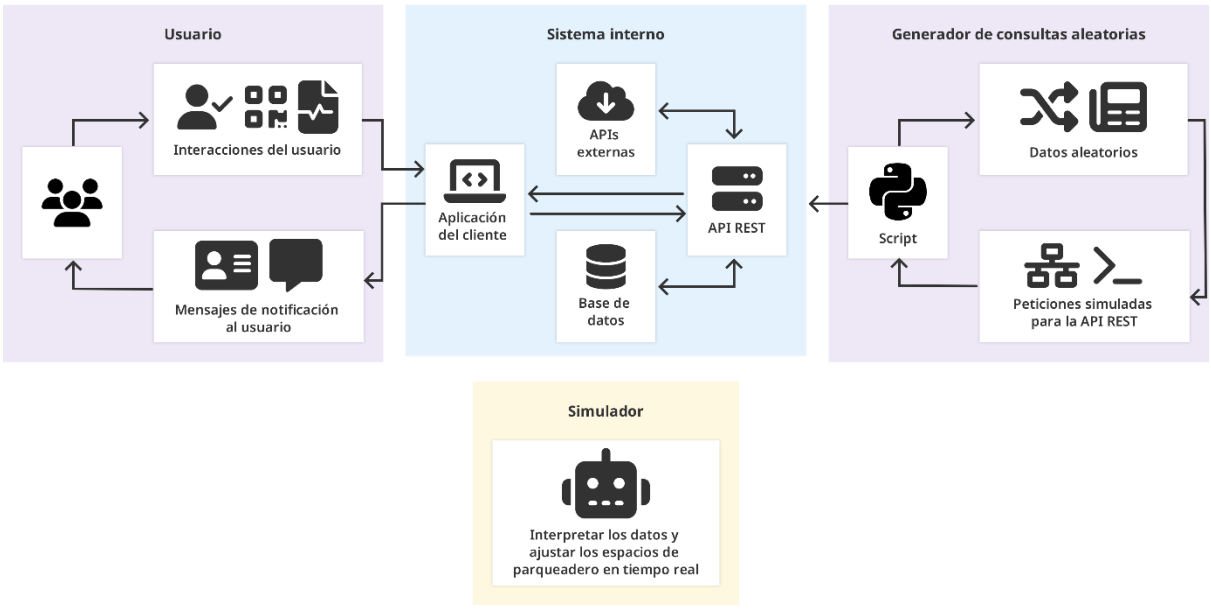
- Arquitectura Hexagonal: El desarrollo del visor de información seguirá las mejores prácticas de la arquitectura hexagonal. En esta arquitectura, la lógica de negocio está separada de la infraestructura y de la interfaz de usuario. Esto asegura que los cambios en un área no afectarán a las demás, permitiendo la adaptabilidad y evolución del sistema con el tiempo.

- Visualización en Tiempo Real: Una característica clave del visor de información es la visualización en tiempo real del estado del estacionamiento. Esta funcionalidad permitirá a los administradores tener una visión continua del estado actual del estacionamiento, incluyendo la ocupación de los espacios y el flujo de vehículos. Esta información en tiempo real mejorará la capacidad de los administradores para tomar decisiones informadas y gestionar eficientemente el parqueadero.

- Pruebas y Verificación: Además de proporcionar información en tiempo real, el visor servirá como una herramienta para las pruebas y la verificación del sistema U'Parking. Se realizarán pruebas rigurosas para verificar la eficacia del sistema, y los resultados serán cuidadosamente documentados y analizados. Esto proporcionará una evidencia valiosa de la robustez y la fiabilidad del sistema U'Parking, y permitirá identificar y solucionar cualquier problema o limitación antes de su implementación real.

El siguiente diagrama muestra la interacción entre las distintas entidades previamente mencionadas, así como una breve descripción del funcionamiento de cada una. Ver figura 13.

Figura 13. Diagrama de interacción de la API REST



Fuente: Construcción del autor.

En la implementación es importante destacar que, durante el proceso de codificación y desarrollo del software, se consideraron todos los aspectos mencionados anteriormente. A continuación, se puede apreciar cómo se estructuró la organización de archivos para adherirse fielmente al diseño mediante una arquitectura limpia y bien definida.

Figura 14. Implementación de arquitectura limpia - Aplicación general

Fuente: Construcción del autor

En la *figura 14* se puede observar la organización y división dentro de la estructura de carpetas de la aplicación, distinguiendo claramente entre el núcleo de esta y su interfaz gráfica escrita en Vue. Esta separación se logra mediante la utilización de espacios de trabajo distintos, lo que permite el uso independiente de las dos partes principales en las que se divide la aplicación: el núcleo (core) y la interfaz gráfica.

Figura 15. Implementación de arquitectura limpia - Core

Fuente: Construcción del autor

En la *figura 15* se plasma la estructura del “Core”. Aquí se puede observar primero la carpeta "Common". Esta contiene los componentes y la lógica que se comparten entre diversas partes de la infraestructura de la aplicación. Alberga funciones como el manejo de errores y el manejo de estados dentro de la aplicación. Si bien se pueden agregar más funciones según sea necesario, en este caso, se mantiene lo más simple posible. Esto se debe a que estos elementos compartidos son potencialmente más frágiles, dado que muchas partes del código los utilizan y

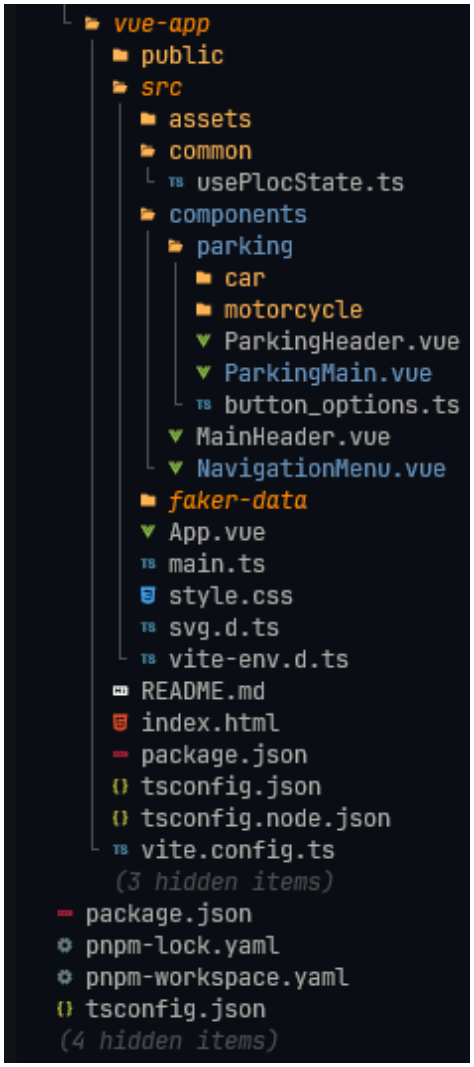
cualquier cambio profundo puede requerir modificaciones importantes en todas las implementaciones que hacen uso de la carpeta "Common".

Pasamos entonces a "Parking", que muestra la estructura y la lógica de negocio del parqueadero. Aquí se puede ver cómo en el "Domain" (dominio) se diseñan cada uno de estos componentes, las entidades del mismo y cómo se debe estructurar el objeto que se utilizará para el repositorio. En este contexto, se define una interfaz crucial, "ParkingRepository", que se emplea dentro de la infraestructura para definir las interfaces concretas.

Después encontramos la carpeta "Application", que define los casos de uso de la aplicación. En "Infrastructure" se alojan las implementaciones generales, así como las implementaciones específicas de la carpeta "Common", particularmente en lo que respecta al manejo de estado.

Para este ejercicio, el manejo de estado se incorpora en "ParkingPloc". Esta es la estructura encargada de gestionar el estado de los datos que circulan alrededor del “Core”. Se utiliza para comunicarse con la interfaz gráfica.

Figura 16. Implementación de arquitectura limpia - Vue App



Fuente: Construcción del autor

En la *figura 16* se puede observar el módulo "Vue App", que sigue una estructura bastante similar a la del “Core”. Incluye una carpeta compartida, que en este caso es aún más sencilla ya que solo gestiona el “usePlocState”, el cual permite conectar y gestionar el estado del “Core”. La interacción de este elemento con el estado predeterminado es crucial, ya que actúa como un observador, registrando los cambios de estado de cualquier objeto definido como "Ploc". En función de estos cambios, se renderiza la interfaz.

Dentro de la estructura, se encuentran diversos componentes que componen la interfaz, entre ellos una denominada “Parking”, que se encarga de la construcción de los slots para carros y para motos. Además, el “MainHeader” es el que facilita la navegación entre los diferentes tipos de espacios de estacionamiento disponibles (automóviles o motocicletas). El módulo “Main”, por su parte, permite visualizar la estructura y la implementación, integrando todos los espacios de parqueadero para formar la vista completa de la interfaz.

Adicionalmente, contamos con la función “Faker-data”, que procesa las solicitudes al servidor para el envío de datos simulados.

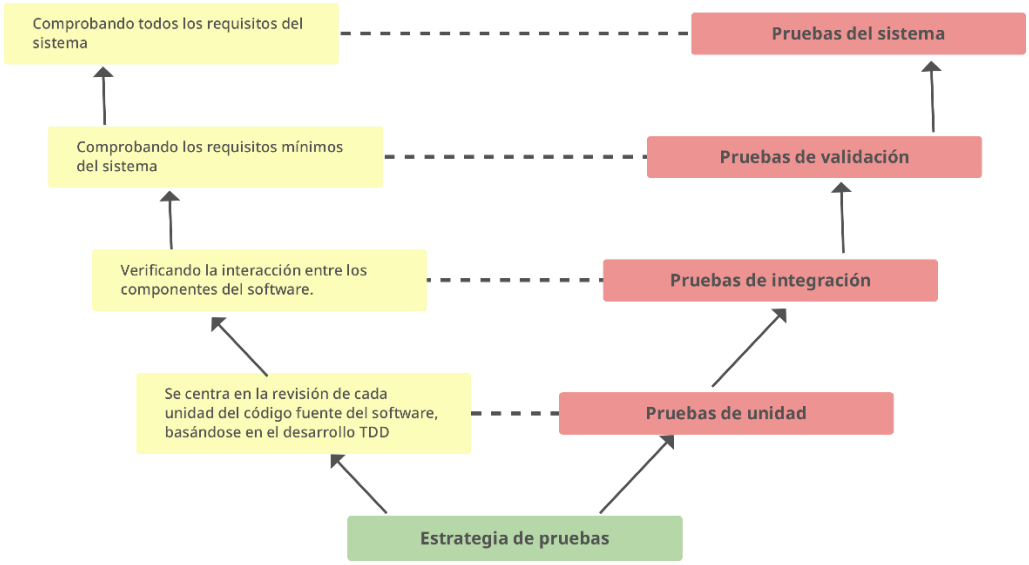
Habiendo dicho esto, se puede determinar que esta estructura permite organizar de manera eficiente las dependencias de cada elemento. Aquellas que son compartidas entre ambos se colocan en la raíz del proyecto, mientras que las específicas de cada uno se ubican en sus respectivas carpetas. Esta disposición demuestra la separación clara entre la estructura y la lógica de los dos componentes, que sólo interactúan de manera puntual. Por ejemplo, esto ocurre en las conexiones al servidor generadas por el núcleo y en el manejo del estado de la información que produce el núcleo y se comparte con la interfaz gráfica. Así es como se gestiona normalmente el flujo de trabajo en este sistema.

Pruebas del software

Las pruebas del software son una parte esencial del desarrollo de cualquier sistema de software, garantizando que el producto final cumple con las especificaciones y requisitos designados, y que proporciona una experiencia de usuario efectiva y eficiente. Para este proceso se elabora un diagrama de estrategia de pruebas, el cual abarca una serie de pruebas estructuradas y dirigidas, cada una enfocada en diferentes niveles del sistema, desde la revisión de código

individual hasta la evaluación de la funcionalidad del sistema en su conjunto. Para el proyecto se siguió un enfoque escalonado desde pruebas de unidad basadas en TDD, pruebas de integración, pruebas de validación y finalmente, pruebas del sistema. Esta secuencia asegura una cobertura de pruebas exhaustiva, identificando y corrigiendo problemas a lo largo del proceso de desarrollo para producir un software robusto y de alta calidad. Ver figura 17.

Figura 17. Diagrama de estrategia de pruebas



Fuente: Construcción del autor

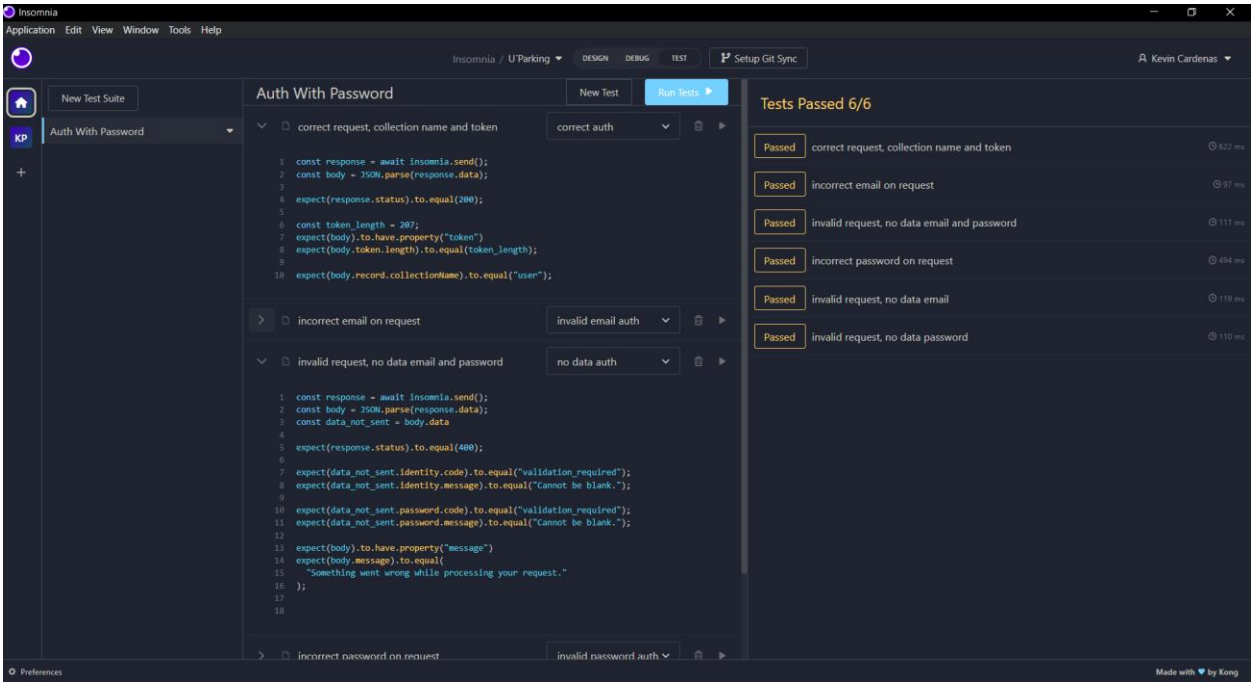
Para el desarrollo de las pruebas, PocketBase ofrece la capacidad de generar distintas bases de datos, lo cual resulta sumamente útil para crear entornos de prueba con diferentes conjuntos de datos sin afectar el entorno de producción. De esta manera, se pueden aislar las bases de datos de prueba y observar cómo reacciona el sistema en diversos escenarios. Para este proyecto en particular, se ha optado por crear una única base de datos de pruebas, debido a que se trabajará con espacios de estacionamiento estáticos. Esta elección se basa en la información disponible sobre la cantidad de espacios de estacionamiento en el Centro Regional Soacha. Considerando que la base de datos registra principalmente los ingresos y salidas de cada usuario, así como la información

sobre el espacio de estacionamiento en el que se encontraban, no es necesario generar múltiples entornos de prueba. Un solo entorno permitirá evaluar su comportamiento en diferentes escenarios.

Para las pruebas de integración, en las que se verifica el funcionamiento completo del sistema en diferentes casos, se utiliza el módulo de "generador de consultas aleatorias" mencionado anteriormente. Este módulo nos permite generar diversos escenarios y evaluar la robustez y capacidad de reacción tanto del sistema en general como del módulo de "simulación". De esta manera, se podrá determinar la eficacia y pertinencia de los ajustes realizados por dicho módulo en diversas situaciones.

Si bien esto proporciona un entorno sólido y estructurado, también es necesario realizar pruebas unitarias en cada módulo de la API REST. Esto permite no solo identificar las acciones disponibles en cada uno, sino también comprender claramente las capacidades del sistema. Para llevar a cabo estas pruebas, se ha decidido utilizar la aplicación Insomnia, la cual es un cliente de APIs REST que no solo permite probar cada solicitud HTTP posible, sino también crear código para pruebas unitarias. De esta manera, se logra verificar la estructura de respuesta de cada solicitud y crear un conjunto de pruebas capaz de identificar cambios significativos en el sistema que puedan afectar su funcionamiento. Esto nos brinda la oportunidad de abordar dichas situaciones en un entorno de pruebas y corregirlas antes de lanzar una nueva versión o parche de la aplicación. A continuación, se muestra una gráfica que ilustra una parte de este conjunto de pruebas. Ver figura 18.

Figura 18. Pruebas unitarias en Insomnia



Fuente: Construcción del autor

Documentación del software

Para la elaboración del documento del proyecto U'Parking, se adoptó la metodología Scrum, que permitió tener un enfoque iterativo e incremental en la creación y revisión del contenido. La documentación fue considerada una parte integral del desarrollo del proyecto y se llevó a cabo de manera continua a lo largo de todo el proceso.

La herramienta Todoist fue fundamental en la planificación, seguimiento y ejecución de las tareas asociadas con la documentación. Cada aspecto del documento de grado fue asignado como una tarea en Todoist con fechas de inicio y fin definidas, lo que ayudó a garantizar la puntualidad y la eficiencia en el trabajo. La documentación se dividió en los siguientes segmentos:

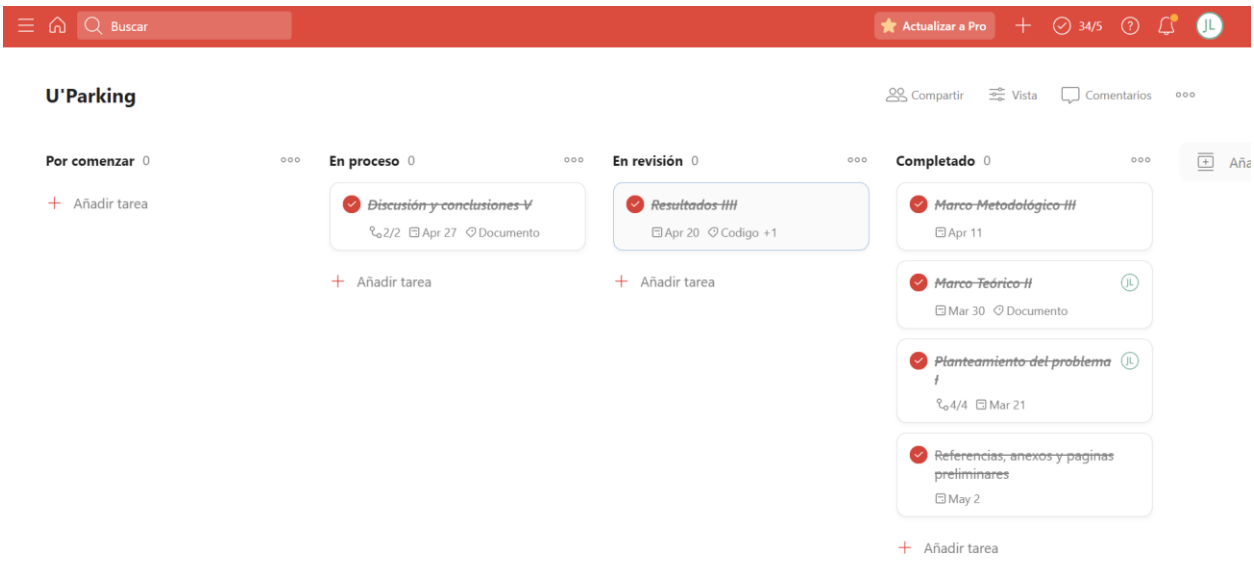
- Documentación de Contenido: Este segmento abarcó la creación y revisión de todo el contenido textual y gráfico del documento. Cada capítulo, sección y subsección del documento se trató como una tarea individual, permitiendo un manejo detallado y meticuloso.

- Documentación de Formato: La atención al formato y estilo del documento de grado también fue esencial. Se establecieron tareas para revisar y ajustar la estructura, citas, referencias, tablas y figuras en todo el documento.

- Documentación de Revisión: Finalmente, se llevó a cabo un proceso de revisión exhaustiva. Cada revisión fue registrada como una tarea, y cualquier cambio o mejora identificada se convirtió en una nueva tarea para asegurar que no se pasaran por alto.

Cabe destacar que la documentación no fue un proceso lineal, sino un esfuerzo iterativo que permitió la mejora continua del documento de grado. Ver figura 19.

Figura 19. Estructura de actividades en Todoist



Fuente: Construcción del autor

Dentro de la documentación del software también se ha puesto un especial énfasis en la creación de dos manuales de usuario integrales. Estos manuales están diseñados para proporcionar un acceso fácil y claro a la información necesaria para la correcta utilización de la aplicación. Estos documentos son de suma importancia para asegurar la satisfacción del usuario, al brindar la información precisa que permite explotar al máximo las funcionalidades del software.

El primer manual de usuario se centra meticulosamente en las diferentes vistas e interacciones que posee la aplicación. En este se detallan y explican las diversas pantallas, menús, ventanas, y otros elementos visuales que conforman la interfaz de usuario. Cada vista es descrita con la suficiente profundidad, especificando las funciones y opciones disponibles, además de proporcionar instrucciones claras sobre cómo navegar entre ellas. Este manual sirve como una guía visual detallada que permite al usuario familiarizarse con la interfaz de la aplicación y entender cómo los diversos componentes y elementos interactúan entre sí.

El segundo manual está enfocado en la perspectiva del administrador. Este aborda de manera exhaustiva las capacidades que el aplicativo ofrece desde la vista de administración, permitiendo la supervisión y el control efectivo del estado del parqueadero.

Este manual detalla el método a través del cual se pueden monitorear en tiempo real las condiciones y el estado del parqueadero. Proporciona una visión precisa de las plazas de estacionamiento ocupadas y disponibles, facilitando la gestión y la toma de decisiones.

Además, el manual explica cómo se pueden ejecutar diferentes funcionalidades dentro de la aplicación. Una de estas es la capacidad de buscar un slot en específico. A través de este

procedimiento, el administrador puede localizar de manera eficiente cualquier espacio de estacionamiento, proporcionando una visión clara y rápida de su estado.

Otra función destacada es la posibilidad de alternar la vista entre espacios destinados a automóviles y motocicletas. Esta funcionalidad permite al administrador manejar y supervisar de forma efectiva los diferentes tipos de espacios de estacionamiento disponibles, mejorando la capacidad de gestión y optimizando el uso del parqueadero.

Ambos manuales han sido diseñados con un lenguaje sencillo y accesible, acompañado de gráficos explicativos y ejemplos prácticos, para que cualquier usuario, independientemente de su nivel de experiencia técnica, pueda comprender y utilizar la aplicación con facilidad. Los manuales se mantienen actualizados para reflejar cualquier cambio o mejora realizada en el software, asegurando que los usuarios siempre tengan acceso a la información más reciente.

Población y muestra

Población

La población del proyecto U'Parking está compuesta por la comunidad universitaria de la Corporación Universitaria Minuto de Dios CRS y las personas que utilizan el estacionamiento de la institución. Esta población incluye:

- Estudiantes: Este grupo abarca a todos los estudiantes matriculados en la Corporación Universitaria Minuto de Dios CRS, tanto en programas de pregrado como de posgrado. Incluye estudiantes de diferentes carreras, semestres y modalidades de estudio.

- Profesores: Este grupo está conformado por el cuerpo docente de la institución, que incluye profesores de tiempo completo y medio tiempo, así como profesores invitados o visitantes que imparten clases en la universidad.

- Personal Administrativo: Comprende a los empleados y funcionarios que desempeñan labores administrativas en la Corporación Universitaria Minuto de Dios CRS. Esto incluye personal de diferentes áreas como recursos humanos, finanzas, servicios generales, seguridad, entre otros.

Muestra

La muestra de este proyecto está compuesta por 25 personas de la comunidad universitaria de la Corporación Universitaria Minuto de Dios CRS. La selección de los participantes se realizó de manera no aleatoria, considerando diferentes categorías de la población, como estudiantes, profesores, personal administrativo y visitantes. Los participantes fueron encuestados con el

objetivo de obtener información sobre su pertenencia a la jornada diurna o nocturna, el medio de transporte utilizado para llegar a la universidad, la frecuencia de uso del parqueadero, el tiempo de ingreso al parqueadero, la percepción sobre la optimización del proceso de registro, la disponibilidad de espacios de estacionamiento, la necesidad de parquear por fuera de la universidad y las recomendaciones para mejorar el servicio de parqueadero. Esta muestra proporciona datos relevantes para comprender las necesidades y opiniones de los usuarios y orientar las mejoras en la gestión y uso del espacio de estacionamiento en la institución.

Instrumentos para la recolección de datos

El proceso de recolección de datos para el proyecto U'Parking se llevó a cabo utilizando la plataforma en línea Google Forms. Se diseñó un cuestionario que constaba de varias preguntas relacionadas con el acceso, uso y percepción del parqueadero en la institución universitaria.

Se generó un enlace único a la encuesta y se compartió con los miembros de la comunidad universitaria de la Corporación Universitaria Minuto de Dios CRS a través de medios electrónicos, como correos electrónicos institucionales y grupos de mensajería.

Se solicitó la participación voluntaria de los miembros de la comunidad universitaria y se les brindó la opción de responder la encuesta en su tiempo libre y desde cualquier dispositivo con acceso a internet.

Durante un período de tiempo determinado, se recopilaron las respuestas de los participantes a través del formulario en línea. Se garantizó la confidencialidad y privacidad de las respuestas, y se aseguró a los participantes que los datos recopilados se utilizarían únicamente con fines de carácter académico. A continuación, se presenta la encuesta realizada:

1. ¿A qué jornada pertenece?

- Diurna
- Nocturna

2. ¿Qué medio de transporte usa para llegar a la universidad?

- Moto

- Carro
- Bicicleta
- Transporte Público
- Otros: _____

3. ¿Con qué frecuencia usa el parqueadero de la universidad?

- 1 o 2 días a la semana
- 3 o más días a la semana

4. ¿Por lo general cuánto tiempo le toma ingresar al parqueadero de la universidad?

- Menos de 2 minutos
- Entre 2 y 5 minutos
- Más de 5 minutos

5. ¿Cree usted que el proceso manual de registro al ingresar se puede optimizar?

- Sí
- No

6. ¿Considera usted que hay suficientes espacios de estacionamiento disponibles en el parqueadero?

- Sí

- No

7. ¿Alguna vez ha tenido que parquear su vehículo por fuera de la universidad?

- Sí
- No

8. En caso de que haya respondido con sí, ¿Observó espacios disponibles para un vehículo diferente al suyo? (Por ejemplo, si no pudo ingresar en moto, observó que había espacios disponibles para carro)

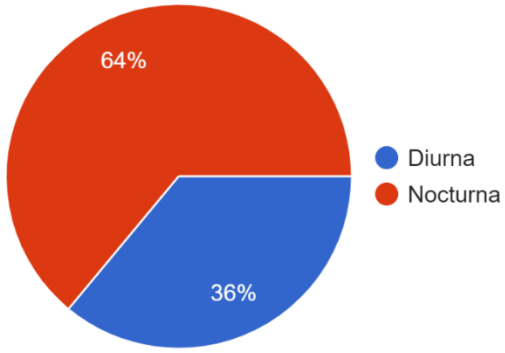
- Sí
- No
- No Aplica

Cabe mencionar que, en el instrumento anterior, la respuesta de la pregunta 2 determina la continuación de la encuesta, ya que la población objetivo del estudio son las personas que hacen uso del parqueadero, y estas solo serían las que ingresen en carro, moto o bicicleta. Si la opción elegida no es ninguna de las mencionadas previamente, la encuesta se dará por finalizada.

Análisis e interpretación de datos

De acuerdo con las respuestas de la pregunta: ¿A qué jornada pertenece?, se puede determinar que 16 de los encuestados pertenecen a la jornada nocturna, y 9 pertenecen a la jornada diurna. Ver figura 20.

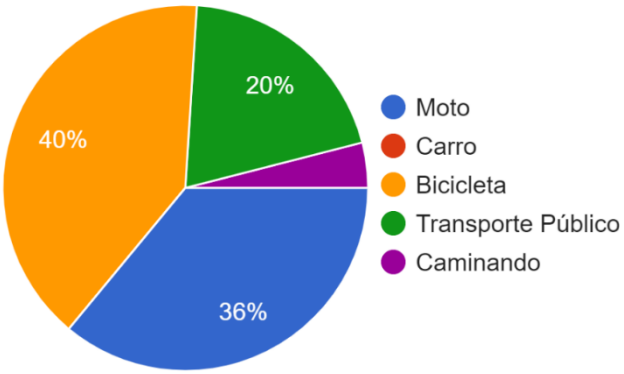
Figura 20. Resultados pregunta 1



Fuente: Construcción del autor.

A partir de las respuestas a la pregunta: ¿Qué medio de transporte usa para llegar a la universidad?, se puede afirmar que 10 de los encuestados ingresar a la institución en bicicleta, 9 en moto, 5 en transporte público, y 1 en otro medio (caminando). Ver figura 21.

Figura 21. Resultados pregunta 2



Fuente: Construcción del autor.

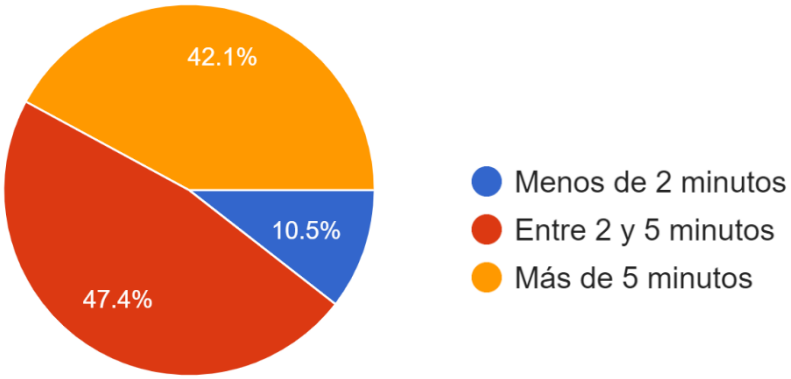
En este punto solo continúan la encuesta las personas que en la pregunta anterior respondieron “Moto”, “Carro”, o “Bicicleta” (19 de las 25 cumplen con este requisito). Habiendo dicho esto, se puede verificar que para la pregunta: ¿Con qué frecuencia usa el parqueadero de la universidad?, un total de 17 personas hacen uso del parqueadero 3 o más días a la semana, y únicamente 2 hacen uso 1 o 2 días. Ver figura 22.

Figura 22. Resultados pregunta 3

Fuente: Construcción del autor.

Según lo que se ha respondido a la pregunta: ¿Por lo general cuánto tiempo le toma ingresar al parqueadero de la universidad?, se puede señalar que 9 personas tardan entre 2 y 5 minutos, 8 se demoran más de 5 minutos, y solamente 2 ingresan en menos de 2 minutos. Ver figura 23.

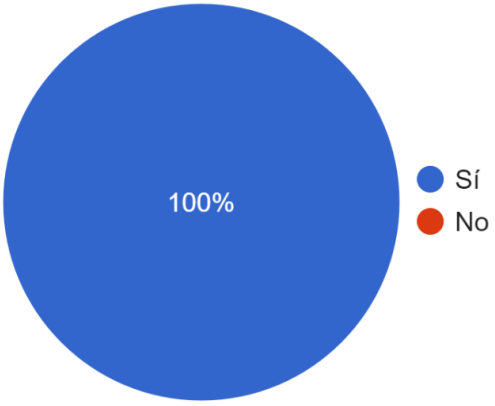
Figura 23. Resultados pregunta 4



Fuente: Construcción del autor.

De conformidad con la respuesta de la pregunta ¿Cree usted que el proceso manual de registro al ingresar se puede optimizar?, se puede asegurar que los 19 encuestados creen que sí. Ver figura 24.

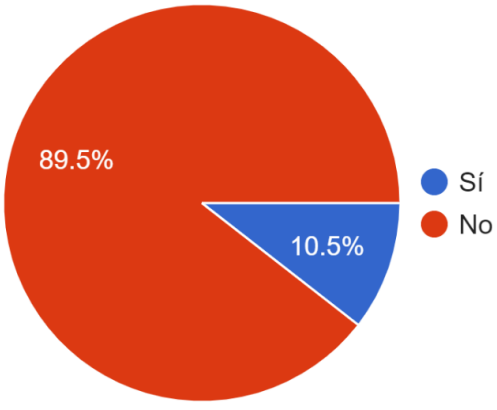
Figura 24. Resultados pregunta 5



Fuente: Construcción del autor.

A partir de las respuestas de la pregunta: ¿Considera usted que hay suficientes espacios de estacionamiento disponibles en el parqueadero? Se puede deducir que 17 afirman que no, y 2 dicen lo contrario. Ver figura 25.

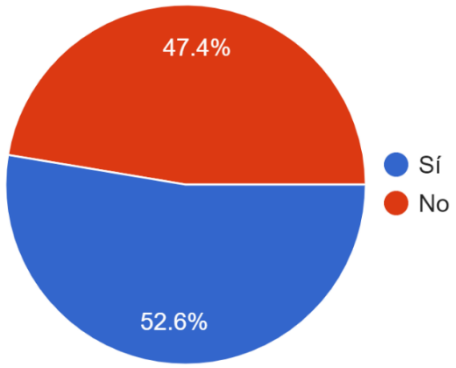
Figura 25. Resultados pregunta 6



Fuente: Construcción del autor.

Según el contenido de las respuestas de la pregunta: ¿Alguna vez ha tenido que parquear su vehículo por fuera de la universidad? Se puede diagnosticar que 10 de los encuestados sí lo han tenido que hacer, en comparación con 9 que no. Ver figura 26.

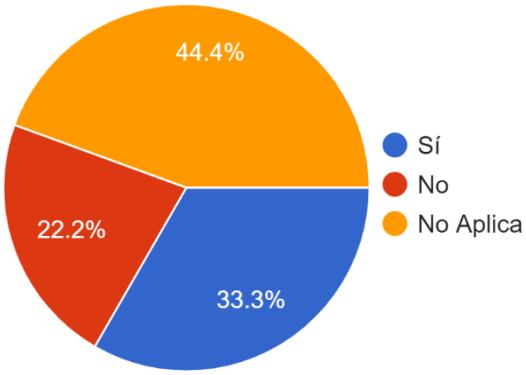
Figura 26. Resultados pregunta 7



Fuente: Construcción del autor.

Por último, para la pregunta: En caso de que haya respondido con sí, ¿Observó espacios disponibles para un vehículo diferente al suyo? (Por ejemplo, si no pudo ingresar en moto, observó que había espacios disponibles para carro), Se puede inferir que de los 10 participantes que respondieron sí en la anterior respuesta, 6 de ellos alguna vez han tenido que parquear su vehículo por fuera de las instalaciones, mientras que los 4 restantes no. Ver figura 27.

Figura 27. Resultados pregunta 8



Fuente: Construcción del autor.

Resultados

A partir de los datos recogidos y su análisis, se pueden destacar varios puntos importantes que pueden influir en el desarrollo del proyecto:

1. La mayoría de los participantes pertenecen a la jornada nocturna. Esto puede influir en las consideraciones de seguridad y accesibilidad del estacionamiento durante las horas de la noche.

2. La mayoría utilizan bicicletas y motos como medio de transporte para llegar al campus. Esta tendencia implica la necesidad de una infraestructura adecuada que garantice un estacionamiento seguro y eficiente para estos vehículos. Además, es esencial optimizar el proceso de registro e ingreso a la institución, de manera que sea eficiente y conveniente para los usuarios de estos medios de transporte.

3. Uso frecuente del parqueadero: La mayoría de los encuestados utilizan el parqueadero de la universidad tres o más días a la semana, lo que indica una alta demanda de este servicio.

4. Tiempo de ingreso al parqueadero: La mayoría de los encuestados reportó un lapso superior a los dos minutos. Esta situación sugiere posibles deficiencias de eficiencia en el proceso de acceso, especialmente durante las horas pico. Se observa que este problema es más acentuado durante la jornada nocturna, sugiriendo la necesidad de optimizar el proceso de ingreso durante estas horas.

5. Optimización del proceso de registro: Todos los participantes creen que el proceso manual de registro al ingresar al parqueadero se puede optimizar, lo que indica un área clara de

mejora, importante tener en cuenta que esta es la principal problemática que se busca solventar con el proyecto.

6. Insuficiencia de espacios de estacionamiento: La mayoría de los participantes no creen que haya suficientes espacios de estacionamiento disponibles en el parqueadero.

7. Necesidad de parquear fuera de la universidad: Un número significativo de participantes ha tenido que parquear su vehículo fuera de la universidad en algún momento, lo que demuestra la falta de capacidad del parqueadero.

8. Observación de espacios disponibles para vehículos diferentes: Algunos de los participantes que han tenido que parquear fuera de la universidad han observado espacios disponibles para un tipo de vehículo diferente al suyo. Esto puede sugerir un desequilibrio en la distribución de espacios para diferentes tipos de vehículos.

Estos hallazgos proporcionan valiosas medidas para el desarrollo del proyecto y subrayan la necesidad de mejorar la eficiencia, capacidad y flexibilidad del parqueadero en la Corporación Universitaria Minuto de Dios CRS.

Por otra parte, en lo que se refiera a la simulación, se puede observar lo siguiente:

Se realizaron pruebas con diferentes cantidades de archivos generados y se registraron los siguientes tiempos de ejecución:

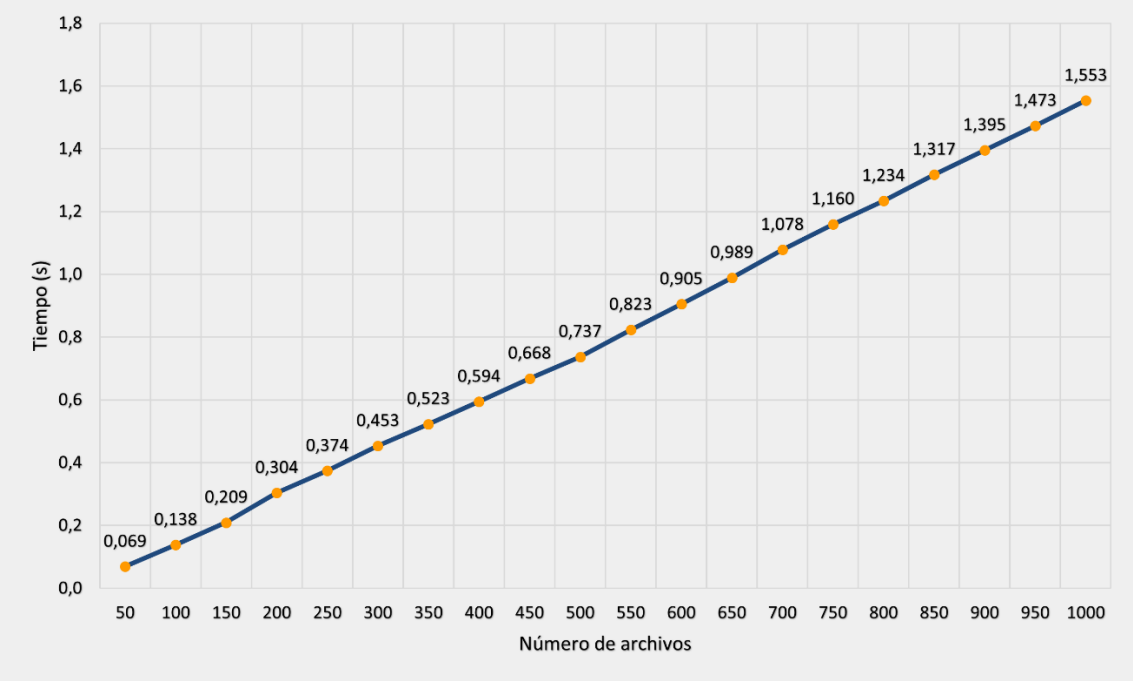
Tabla 1. Tiempo de ejecución de archivos generados

Ejecución	Tiempo	Archivos
Primera	1.553149 segundos	1.000
Segunda	14.904239 segundos	10.000
Tercera	186.865091 segundos	100.000

Fuente: Construcción del autor

A partir del ejercicio realizado, se puede evaluar el comportamiento del algoritmo, el cual se puede ver reflejado más al detalle en las siguientes gráficas.

Figura 28. Archivos generados vs Tiempo - Primer caso

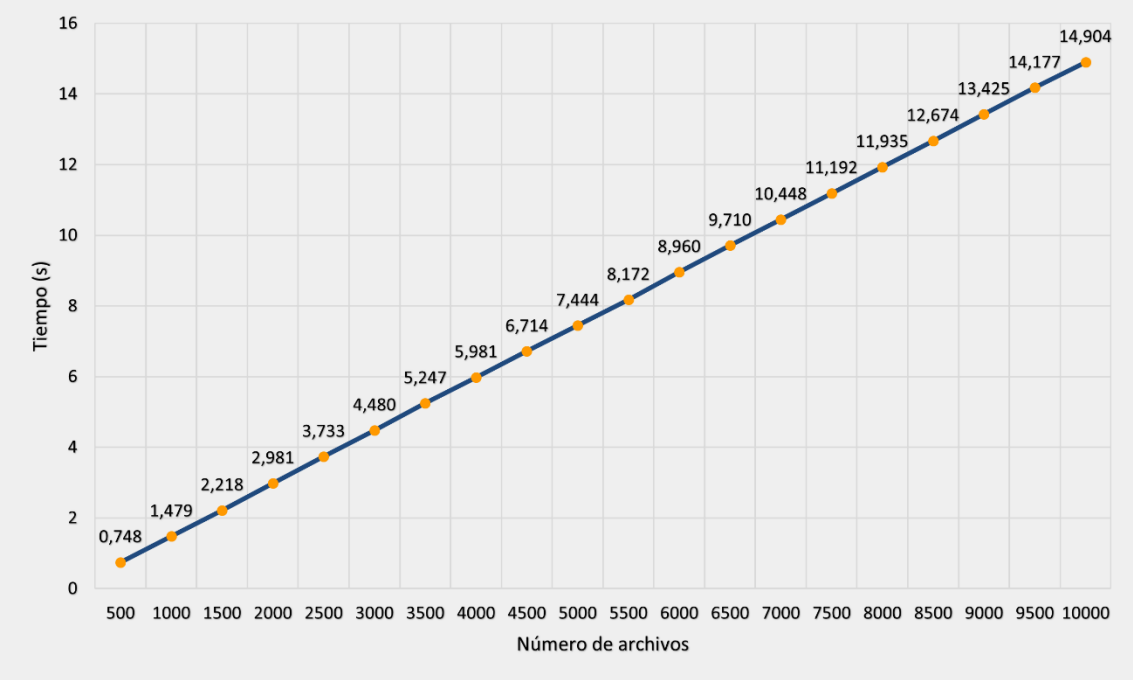


Fuente: Construcción del autor

Para un total de 1.000 archivos, el tiempo de procesamiento es de 1,553149 segundos. Como se evidencia en la *figura 28*, se observó una tendencia lineal en el comportamiento, indicando que a medida que aumenta la cantidad de archivos, el tiempo de procesamiento también

incrementa proporcionalmente. Sin embargo, a pesar de este aumento, el tiempo promedio de generación se mantiene, aproximadamente en torno a 0,001553 segundos por archivo.

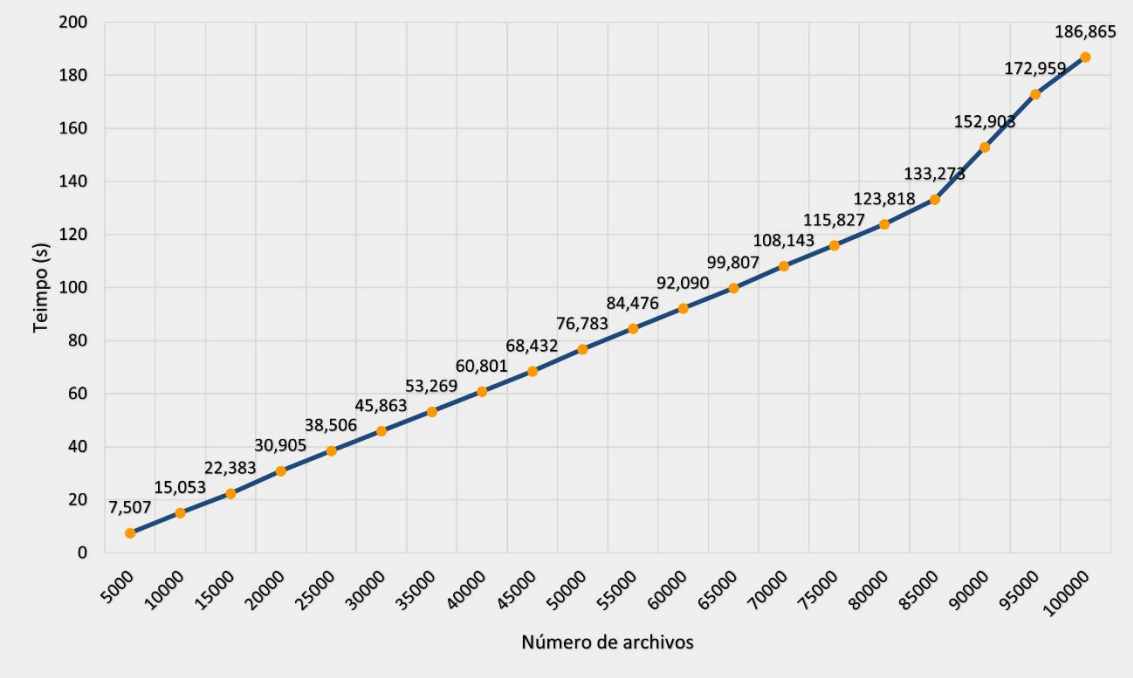
Figura 29. Archivos generados vs Tiempo - Segundo caso



Fuente: Construcción del autor

En una instancia donde se generaron 10.000 archivos, el tiempo total de procesamiento ascendió a los 14,904239 segundos. El escenario de la *figura 29* también muestra una tendencia lineal. Sin embargo, a pesar de la significativa diferencia en la cantidad de archivos en comparación con la primera prueba, el tiempo promedio de procesamiento por archivo se mantuvo casi constante, registrando aproximadamente 0,00149 segundos por archivo.

Figura 30. Archivos generados vs Tiempo - Tercer caso



Fuente: Construcción del autor

En el tercer escenario, en el que se generaron 100.000 archivos aleatorios, el algoritmo tardó un total de 186,865091 segundos en ejecutarse. Esto implica un tiempo de procesamiento promedio de 0,001869 segundos por archivo. Como se puede observar en la *figura 30*, la generación de entre 85.000 y 100.000 archivos mostró un incremento en el tiempo de procesamiento, lo cual sugiere que el algoritmo requiere de mayor capacidad de procesamiento para manejar grandes volúmenes de registros. Sin embargo, a pesar de este incremento, la tasa de crecimiento no es exponencial, lo que indica que el algoritmo, a pesar de la mayor carga, sigue respondiendo de manera eficiente.

Discusión

El proyecto de investigación U'Parking ha permitido abordar una problemática común en muchas instituciones universitarias: la gestión eficiente y eficaz de los espacios de estacionamiento. Esta cuestión, tiene un impacto directo en la vida diaria de los estudiantes, docentes y personal administrativo que requieren ingresar con frecuencia a la institución y se ven limitados debido al lento y manual proceso de registro al momento de ingresar o salir de la sede.

Es importante mencionar que la elección de una solución tecnológica para resolver este problema no es casual, sino que se basa en la comprensión de que la tecnología puede mejorar notablemente los procesos logísticos y administrativos de la institución. Además, la elección de la metodología Scrum y el desarrollo de software basado en la arquitectura limpia aportan robustez, flexibilidad y adaptabilidad al proyecto, permitiendo abordar los cambios y ajustes necesarios no solo durante el proceso de desarrollo y simulación, si no al momento de implementar este sistema en un entorno de producción.

De esta manera, es importante destacar que este proyecto no pretende ser una implementación completa en un entorno de producción, sino un diseño riguroso que pueda justificarse mediante simulaciones y entornos de prueba. La capacidad de comprobar la robustez del proyecto y cómo se adapta a diferentes entornos es crucial para garantizar su viabilidad y eficacia.

Conclusiones

Se han analizado a fondo los requerimientos de los usuarios y del personal de seguridad para la implementación del aplicativo de control de acceso y gestión de estacionamiento. Esto ha permitido comprender las necesidades y expectativas de los usuarios y personal de seguridad, lo que ha sido clave para diseñar un sistema que se adapte a sus necesidades y mejore su experiencia.

El sistema U'Parking ha demostrado ser efectivo en la integración de un sistema de gestión de acceso y asignación de espacios de estacionamiento que optimiza la utilización de los espacios disponibles y mejora la eficiencia en el uso del espacio de estacionamiento. Esto favorece una mayor dinámica en el movimiento y gestión de los vehículos y una utilización más eficiente del espacio de estacionamiento.

Se ha diseñado y puesto en marcha una plataforma tecnológica que permite monitorear y registrar en tiempo real el estado de los espacios de estacionamiento. La plataforma ha probado ser una herramienta efectiva para la gestión y control del estacionamiento, permitiendo una mejor visibilidad y control sobre los espacios disponibles y ocupados.

Finalmente, el desarrollo de una simulación mediante pruebas de integración y herramientas de análisis de datos ha permitido verificar las diferentes respuestas del sistema ante eventos en tiempo real. Estas pruebas han demostrado la robustez y la capacidad de adaptación del sistema, permitiendo verificar que U'Parking es capaz de responder eficientemente a diferentes situaciones.

Recursos de apoyo

La tabla de costos del proyecto U'Parking es una herramienta fundamental para evaluar y planificar los recursos financieros necesarios para su implementación. Esta tabla presenta de manera organizada y detallada los diferentes componentes y rubros que implican costos, permitiendo una visión clara y transparente de los gastos asociados al proyecto. El objetivo de esta tabla es proporcionar una visión global y precisa de los costos asociados al proyecto, lo que permitirá una correcta planificación y asignación de los recursos financieros necesarios para su ejecución exitosa. Ver Tabla 1.

Tabla 2. Costos

Categoría	Descripción	Tiempo	Precio
Talento Humano	1 desarrollador Back-end y Database Manager	2 meses	$5.000.000 x mes
Talento Humano	1 desarrollador Front-end y Diseñador UX/UI	2 meses	$4.000.000 x mes
Infraestructura	Servidor de aplicaciones	12 meses	$9.900 x mes
Otros costos	Equipos de cómputo	2 meses	$5.000.000 c/u

Fuente: Construcción del autor.

Cronograma de actividades

El cronograma de actividades del proyecto U'Parking, basado en la metodología Scrum, es una herramienta esencial para la planificación y seguimiento de las iteraciones y tareas que componen el desarrollo del proyecto. Se organiza en períodos de tiempo definidos en los que se llevan a cabo las actividades planificadas. Este se compone de elementos como la planificación, ejecución de tareas, reuniones semanales de seguimiento y revisión de resultados. Este enfoque iterativo e incremental permite una adaptación ágil a medida que se avanza en el proyecto y se obtienen nuevos aprendizajes. El cronograma proporciona una guía dinámica para el equipo de desarrollo, facilitando la toma de decisiones y la entrega continua de valor en cada sprint. Ver Tabla 2.

Tabla 3. Cronograma de actividades

Actividades	**Semanas**											
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**
Inicio del proyecto	**X**											
Organización y planeación	**X**											
Recolección de Información		**X**										
Definición de requisitos y método de desarrollo		**X**	**X**									
Diseño del aplicativo				**X**	**X**							
Codificación					**X**	**X**	**X**	**X**	**X**			
Pruebas y test							**X**	**X**	**X**	**X**	**X**	
Documentación					**X**	**X**	**X**	**X**	**X**	**X**	**X**	
Elaboración del documento que soporta el proyecto	**X**	**X**	**X**	**X**	**X**	**X**	**X**	**X**	**X**	**X**	**X**	**X**
Reuniones de seguimiento	**X**	**X**	**X**	**X**	**X**	**X**	**X**	**X**	**X**	**X**	**X**	**X**
Entrega y finalización del proyecto												**X**

Fuente: Construcción del autor.

Referencias

Aditya, A., Anwarul, S., Tanwar, R., & Koneru, S. K. V. (2023). An IoT assisted Intelligent Parking System (IPS) for Smart Cities. *Procedia Computer Science*, *218*, 1045–1054. https://doi.org/10.1016/j.procs.2023.01.084

Alexander Shvets. (2019). *Dive into Design Patterns*.

Almawgani, A. H. M., Alsuwian, T., Alhawari, A. R. H., Alhuthari, A. N., Alhezabr, M. A., Alharethi, M. S., & Alqahtani, F. H. (2021a). Smart and efficient system for the detection of wrong cars parking. *Bulletin of Electrical Engineering and Informatics*, *10*(4), 1968–1978. https://doi.org/10.11591/EEI.V10I4.2634

Bin Mohd Nazri, M. S., Long Alif Faiqal Bin Tengku Long Gaafar, T., Sofian, H., & Bakar Sajak, A. A. (2020a). IoT Parking Apps with Car Plate Recognition for Smart City using Node Red. *2020 11th International Conference on Information and Communication Systems, ICICS 2020*, 324–330. https://doi.org/10.1109/ICICS49469.2020.239511

Errousso, H., Malhene, N., Benhadou, S., & Medromi, H. (2020). Predicting car park availability for a better delivery bay management. *Procedia Computer Science*, *170*, 203–210. https://doi.org/10.1016/j.procs.2020.03.026

Florea, A., Fleaca, V., & Marcu, S. D. (2020a). Innovative solution for parking-sharing of private institutions using various occupancy tracking methods. *Advances in Science, Technology and Engineering Systems*, *5*(5), 808–819. https://doi.org/10.25046/AJ050598

Koumetio Tekouabou, S. C., Abdellaoui Alaoui, E. A., Cherif, W., & Silkan, H. (2022). Improving parking availability prediction in smart cities with IoT and ensemble-based model. *Journal*

of King Saud University - Computer and Information Sciences, *34*(3), 687–697. https://doi.org/10.1016/j.jksuci.2020.01.008

Levkivskyi, V. L., Marchuk, D. K., Lobanchykova, N. M., Pilkevych, I. A., & Salamatov, D. I. (2022). *Available parking places recognition system*. https://cs.ztu.edu.ua/our-team

Martin, R. C. (2002). *Agile software development, principles, patterns, and practices*.

Martin, R. C. (2018). *Clean Architecture: A Craftsman's Guide to Software Structure and Design*.

Mohandes, M., Deriche, M., Abuelma'Atti, M. T., & Tasadduq, N. (2019a). Preference-based smart parking system in a university campus. *IET Intelligent Transport Systems*, *13*(2), 376–384. https://doi.org/10.1049/iet-its.2018.5207

Pazos, N., Müller, M., Favre-Bulle, M., Brandt-Dit-Grieurin, K., Hüsser, O., Aeberli, M., & Ouerhani, N. (2016a). Dynamic street-parking optimisation. *Proceedings - International Conference on Advanced Information Networking and Applications, AINA*, *2016-May*, 1020–1026. https://doi.org/10.1109/AINA.2016.171

Razak, S. F. A., Liew, C. L., Lee, C. P., & Lim, K. M. (2015). Interactive android-based indoor parking lot vehicle locator using QR-code. *2015 IEEE Student Conference on Research and Development, SCOReD 2015*, 261–265. https://doi.org/10.1109/SCORED.2015.7449337

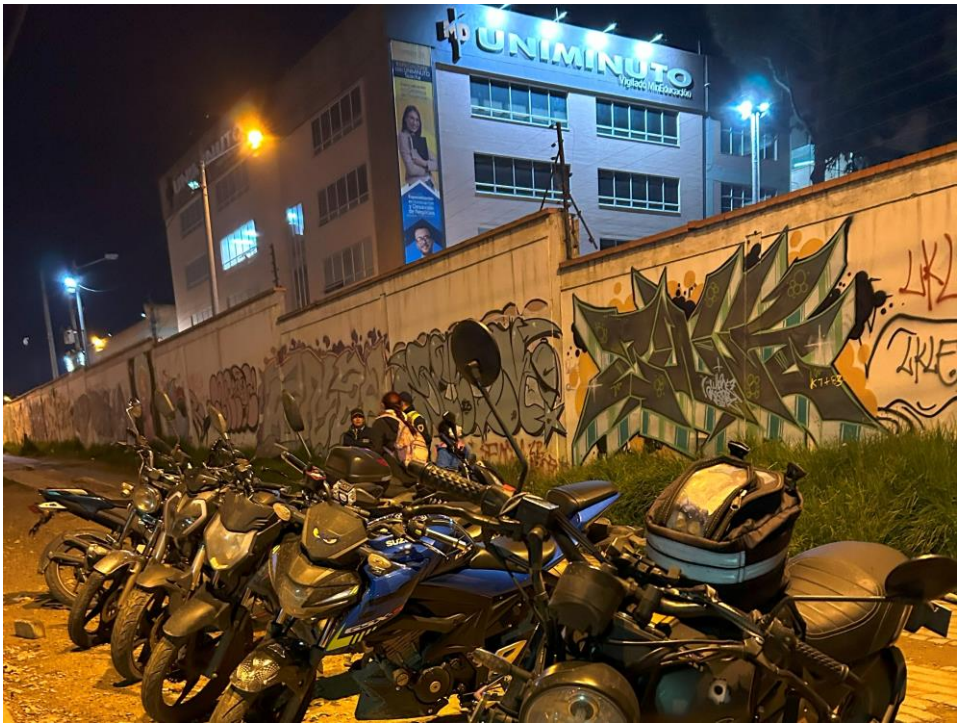
Robert C. Martin. (2008). *Clean Code: A Handbook of Agile Software Craftsmanship*.

S, S., Harshavardhanan, P., Subramani, K., Senthil, P., Veena, T., Faith S, J., & V, N. (2022). Conceptual approach on smart car parking system for industry 4.0 internet of things assisted networks. *Measurement: Sensors*, *24*. https://doi.org/10.1016/j.measen.2022.100474

Veeramanickam, M. R. M., Venkatesh, B., Bewoor, L. A., Bhowte, Y. W., Moholkar, K., & Bangare, J. L. (2022). IoT based smart parking model using Arduino UNO with FCFS priority scheduling. *Measurement: Sensors*, *24*. https://doi.org/10.1016/j.measen.2022.100524

Anexos

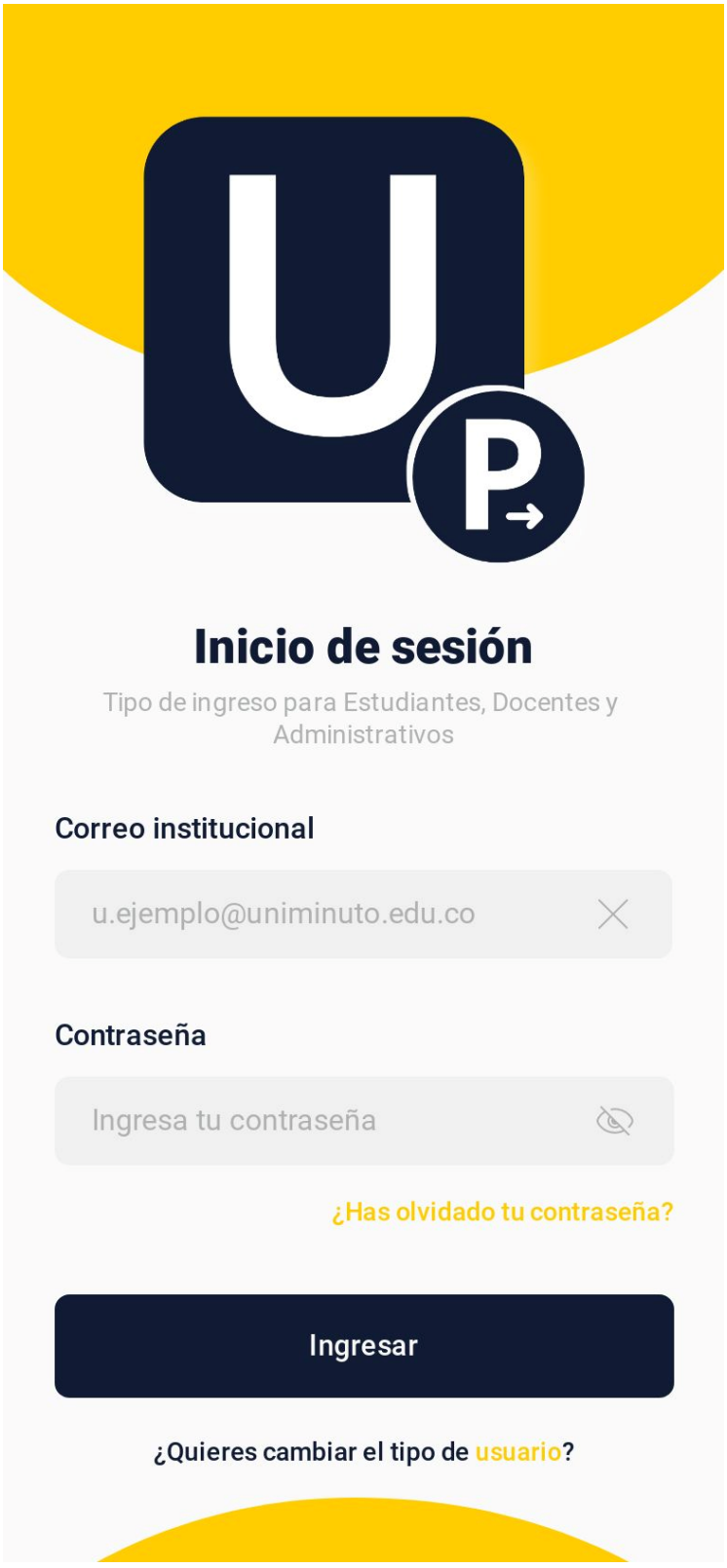
Anexo A. Problemática de distribución de espacios en el parqueadero de la Corporación Universitaria Minuto De Dios - Centro Regional Soacha



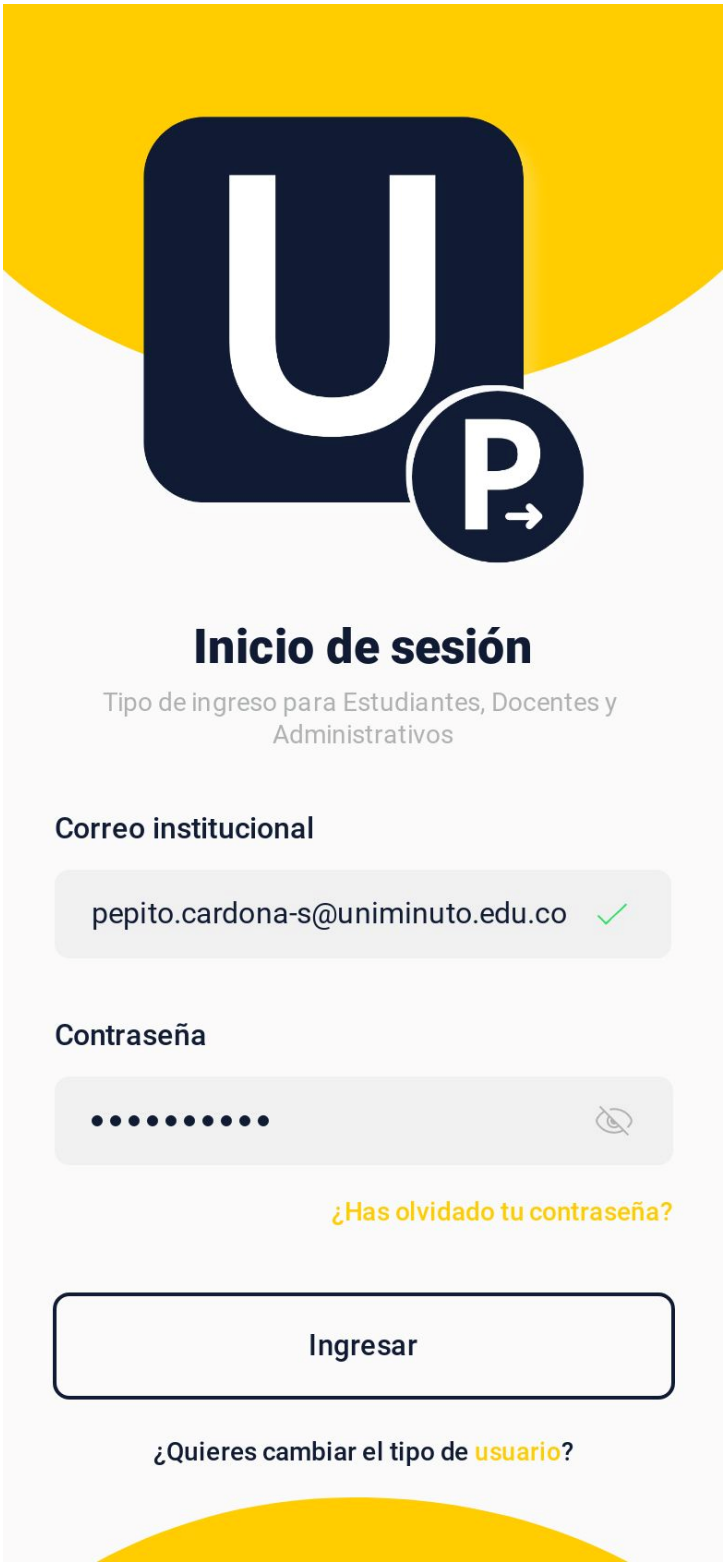

Derechos
Humanos
Diferencia
PAZ

Anexo B. Logo para la Aplicación Web Progresiva

Anexo C. Vista de Inicio de Sesión

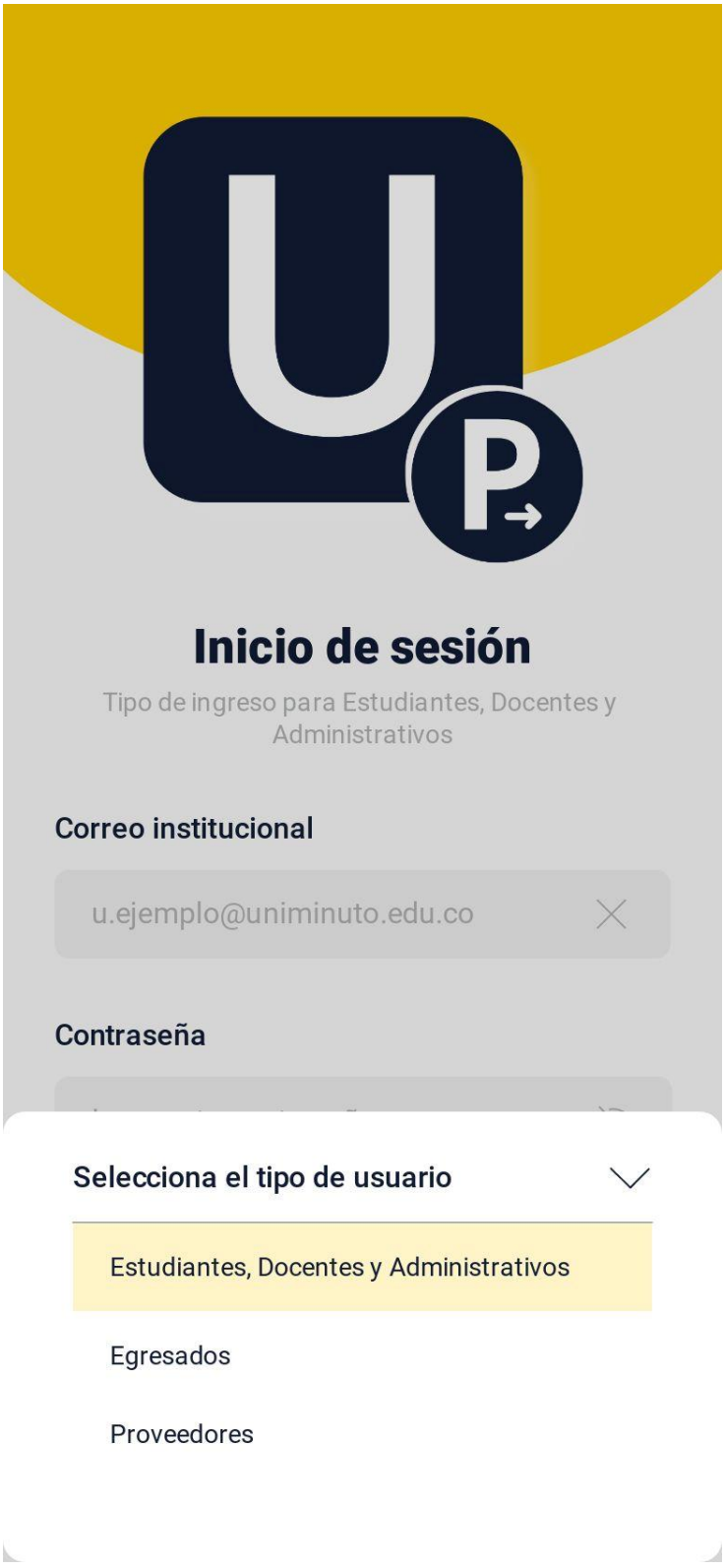


Anexo D. Vista de validación de campos de ingreso

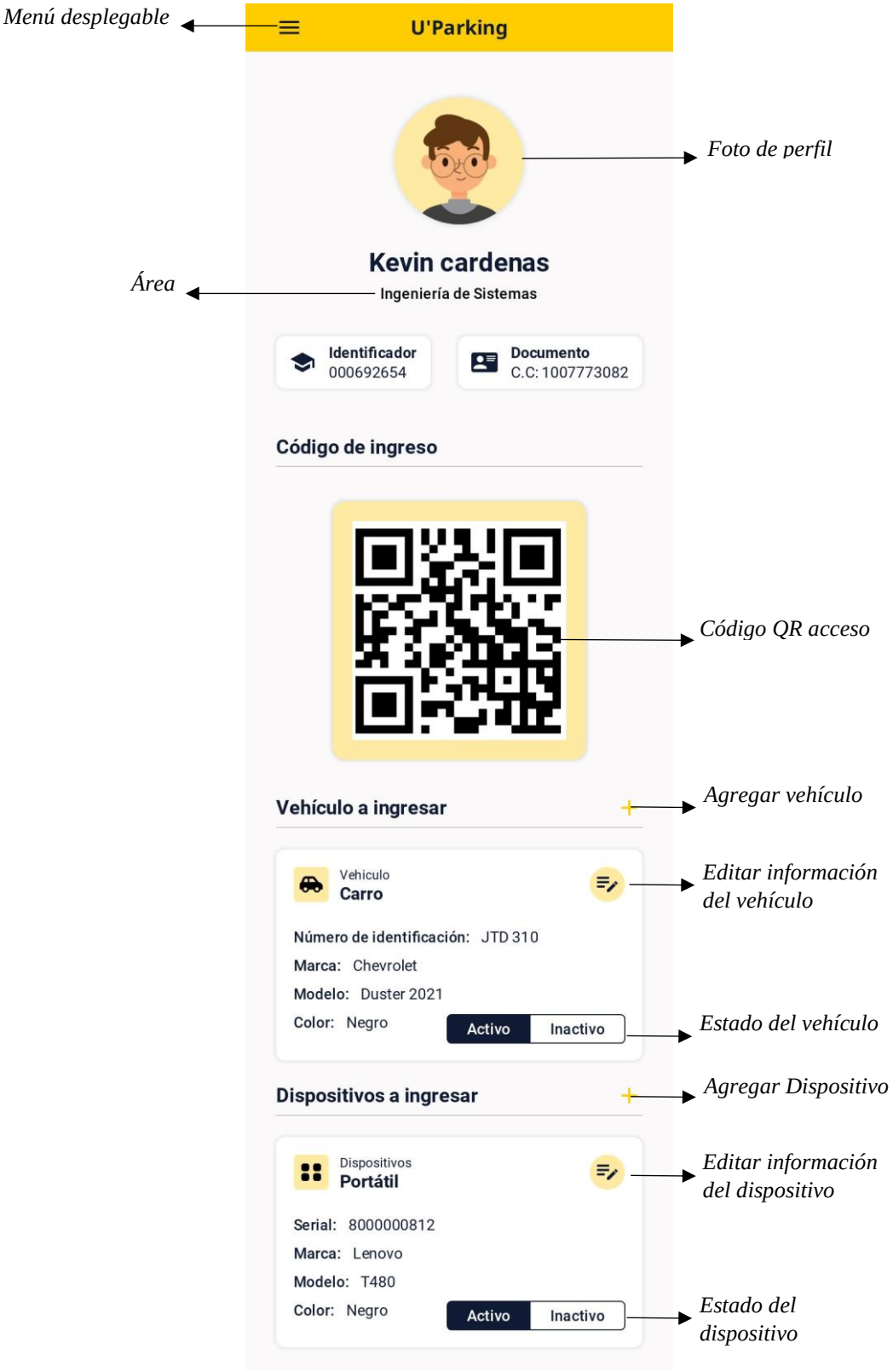


Anexo E. Vista de Inicio de Sesión incorrecto

Anexo F. Vista de Selección de tipo de usuario

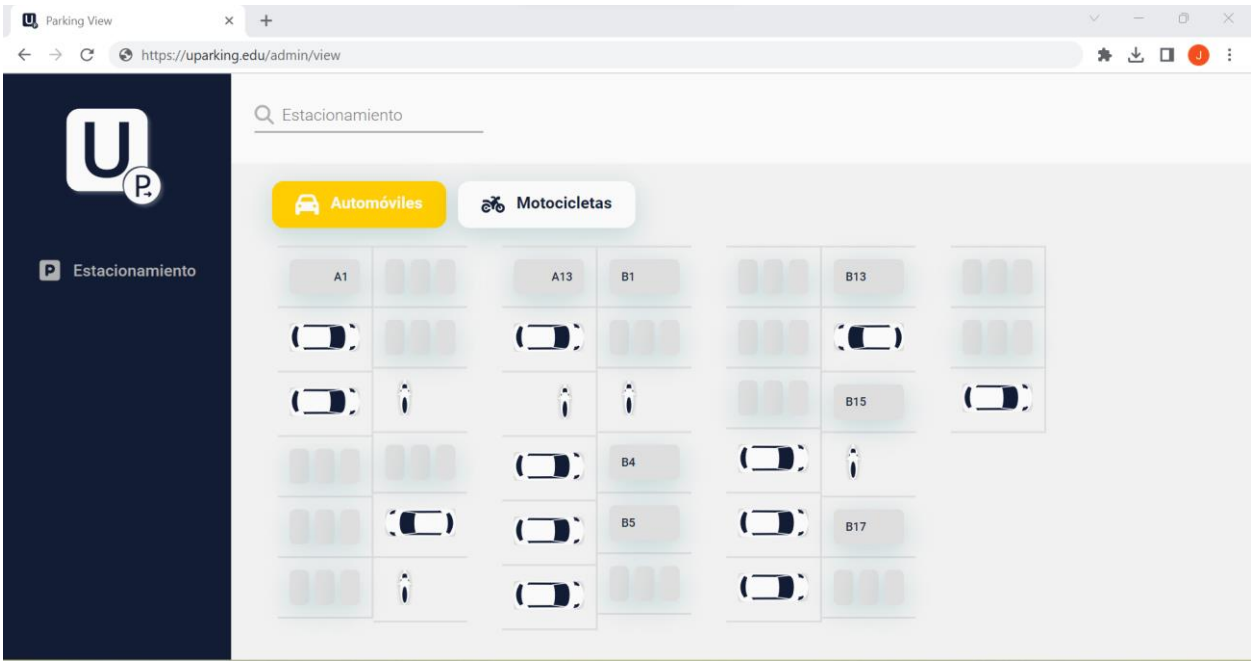


Anexo G. Mockup de pestaña principal

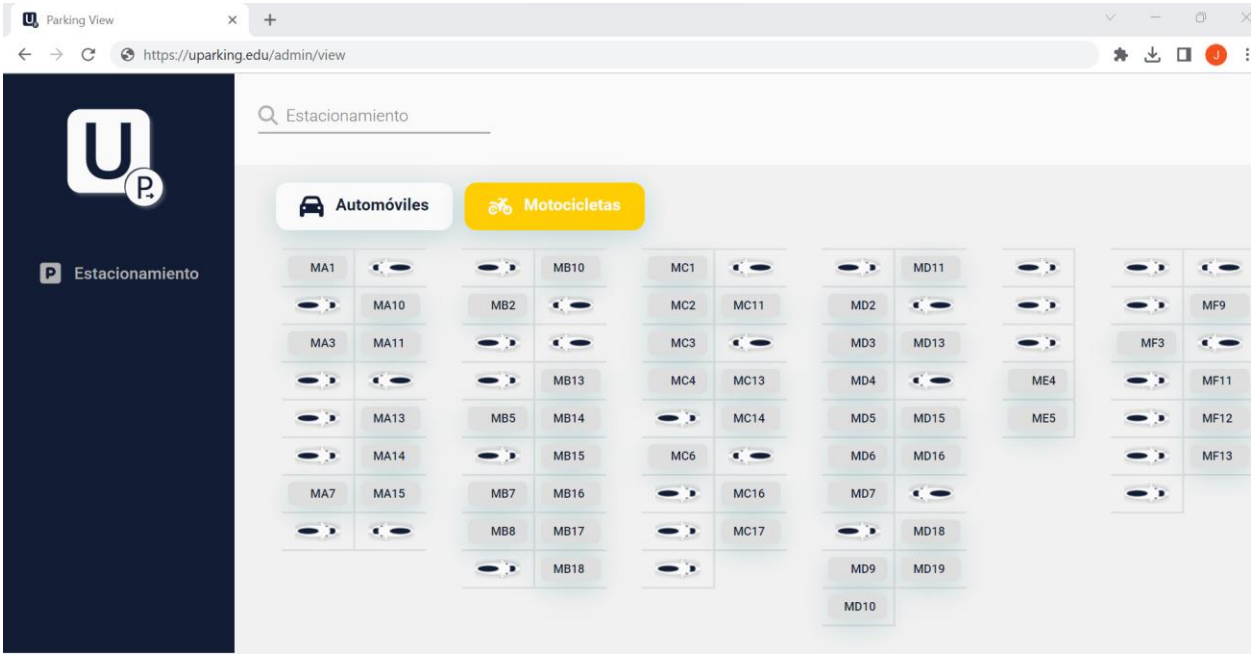


Anexo H. Vista final pestaña principal.

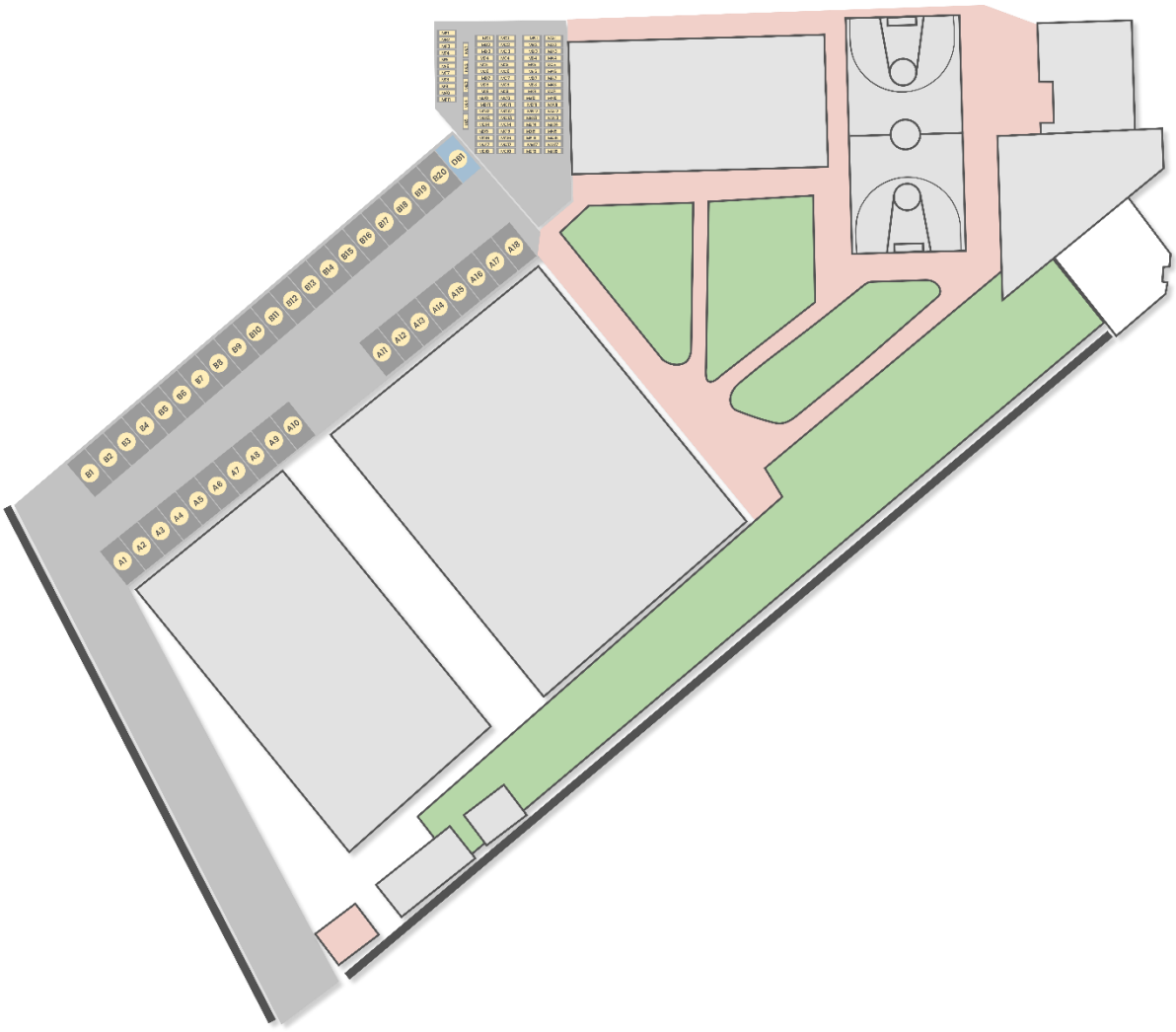
Anexo I. Vista de Administrador – Parqueadero automóviles



Anexo J. Vista de Administrador – Parqueadero motocicletas



Anexo K. Plano general Corporación universitaria Minuto de Dios - CRS



Anexo L. Consentimiento informado para entrevista.

Consentimiento Informado para Participar en la Recolección de Datos

Título del estudio: Estado Actual del Parqueadero en la Corporación Universitaria Minuto de Dios CRS

Estimado participante:

Se le invita a participar en la recolección de datos para el estudio sobre el estado actual del parqueadero en la Corporación Universitaria Minuto de Dios CRS. Antes de tomar la decisión de participar, lea atentamente la siguiente información sobre el estudio y sus derechos como participante. Si tiene alguna pregunta o inquietud, no dude en hacerla antes de tomar una decisión.

1. Objetivos del estudio: El objetivo principal de este estudio es obtener información sobre el estado actual del parqueadero de la Corporación Universitaria Minuto de Dios CRS, específicamente en relación con la existencia de un registro formal o minuta para el control de acceso y gestión de estacionamiento.

2. Procedimiento de recolección de datos: Participar en el estudio implica responder un cuestionario con preguntas relacionadas con el registro de vehículos que ingresan al parqueadero. Sus respuestas serán tratadas de manera confidencial y anónima.

3. Beneficios potenciales: Su participación en este estudio contribuirá a recopilar información relevante sobre el estado del parqueadero, lo cual puede ayudar a mejorar la eficiencia en la gestión de estacionamientos y la seguridad de la institución.

4. Riesgos potenciales: No se anticipan riesgos significativos asociados con su participación en este estudio. Sin embargo, si experimenta alguna incomodidad o malestar al responder las preguntas, puede optar por no participar o interrumpir su participación en cualquier momento.

5. Confidencialidad: Todas las respuestas proporcionadas serán tratadas de manera confidencial y se mantendrán anónimas en los informes del estudio. Los datos recopilados se utilizarán exclusivamente para los fines de la investigación.

6. Voluntariedad y derecho a retirarse: Su participación en este estudio es voluntaria, y tiene el derecho de retirarse en cualquier momento sin ninguna consecuencia o perjuicio. No se le solicitará ninguna justificación si decide no participar o si decide retirarse antes de finalizar el cuestionario.

7. Contacto: Si tiene alguna pregunta o inquietud sobre el estudio, puede comunicarse con el investigador responsable en kevin.cardenas-c@uniminuto.edu.co, luis.trujillo@uniminuto.edu.co o jhon.lugo@uniminuto.edu.co.

Al firmar a continuación, confirmo que he leído y comprendido la información proporcionada en este Consentimiento Informado y estoy de acuerdo en participar voluntariamente en la recolección de datos para el estudio sobre el estado actual del parqueadero en la Corporación Universitaria Minuto de Dios CRS.

Nombre del participante: Meltsin Jose Castro Sosa

Firma del participante: Meltsin Castro

Fecha: 29-05-2023