

Diseño de una herramienta software de gestión y control del proceso de entrada y salida de vehículos de la Universidad Industrial de Santander.

Einer Steven Jaimes Mantilla y Jhoann Sebastián Martínez Oviedo

Trabajo de Grado para Optar al Título de Ingeniero de Sistemas

Director

Jathinson Meneses Mendoza

Magister en Gestión, Aplicación y Desarrollo de Software

Universidad Industrial de Santander

Facultad de Ingenierías Fisicomecánicas

Escuela de Ingeniería de Sistemas e Informática

Bucaramanga

2023

Dedicatoria

En primera instancia a Dios, por darme la sabiduría, paciencia y resiliencia ante todas las dificultades presentadas a lo largo de este camino universitario, así mismo las bendiciones recibidas durante este camino.

A mi querida madre, Marly Mantilla, mi más grande ejemplo de perseverancia, dedicación y amor incondicional, quiero expresar mi más sincero agradecimiento por tu apoyo constante, tus palabras de aliento y tu confianza en mi capacidad para alcanzar mis metas, gracias por tus sacrificios, tus consejos y tus enseñanzas que han sido fundamentales para mi formación como persona y como profesional.

A mi padre, José Luis Jaimes, por estar presente y dispuesto a apoyarme en cualquier nuevo proyecto de vida.

A mi familia, mis abuelos, tías, primas y todos aquellos que se tomaron el tiempo de apoyarme en este camino.

A mi novia, Marly Maldonado, dedico este logro a ti, mi amor, porque tú eres mi inspiración y mi razón de ser. Espero que este sea solo el comienzo de muchos éxitos que compartiremos juntos en el futuro.

A Neytiri, por ser la mejor compañía durante esta etapa de mi vida.

A mis amigos en general, ya que sin su apoyo y palabras de aliento no hubiese logrado salir de esos momentos de frustración que puede ocasionar la universidad

Einer Steven Jaimes Mantilla.

Dedicatoria

Especialmente a mi padre Fredy quien ha sido mi apoyo incondicional durante todo el proceso y en el transcurso de este arduo ciclo de superación personal, constancia y dedicación, el cual siempre ha confiado en mí, me ha inculcado valores y criterio personal, y gracias a él puedo decir que hoy soy quien soy.

Una dedicatoria a mi madre Kelly, amigos, hermana y todos aquellos que me apoyaron y creyeron en mí, que confiaron en que podía lograrlo y hoy en día puedo decir que lo logre.

A mi prometida y compañera de vida Nicoll que hoy en día motiva mis días y genera en mi un estado de superación y mejora constante, la cual representa una gran e importante parte de mi vida y forma parte de mi futuro.

A mi compañero y mejor amigo, el cual durante el trayecto profesional compartimos juntos toda esa travesía. El cual me enseñó de constancia y dedicación, un apoyo incondicional muy grande y que hoy en día genera gratitud y buenos recuerdos.

Culminado este proceso solo me queda un grato recuerdo de mi corto paso por esta, mi alma mater, y un gran agradecimiento.

Jhoann Sebastián Martínez Oviedo.

Agradecimientos

A MSc Jathinson Meneses Mendoza, nuestro director de proyecto de grado, por su colaboración, tiempo y conocimiento brindados a lo largo del desarrollo de esta tesis.

A la Universidad Industrial de Santander, a la Escuela de Ingeniería de Sistemas y a cada uno de nuestros docentes por brindarnos el espacio para aprender, mejorar y crecer tanto intelectualmente como personalmente.

A nuestras familias y amigos por acompañarnos y apoyarnos en esta etapa de nuestra vida.

Contenido

| | |
|--|----|
| Introducción..... | 12 |
| 1 Objetivos..... | 13 |
| 1.1 Objetivo General..... | 13 |
| 1.2 Objetivos Específicos..... | 13 |
| 2 Planteamiento de problema..... | 14 |
| 3 Marco de referencia..... | 16 |
| 3.1 Marco conceptual..... | 16 |
| 3.1.1 Desarrollo de aplicaciones móviles..... | 16 |
| 3.1.2 Frontend..... | 16 |
| 3.1.3 API REST..... | 16 |
| 3.1.4 Sistema de información..... | 17 |
| 3.1.5 Tecnología de información..... | 17 |
| 3.1.6 Apis..... | 17 |
| 3.1.7 Bases de datos..... | 18 |
| 3.1.8 Modelo Relacional..... | 18 |
| 3.1.9 Flutter..... | 18 |
| 3.1.10 Android Studio..... | 19 |
| 3.1.11 Github..... | 19 |
| 3.1.12 Visual Studio Code..... | 19 |
| 3.1.13 Dart..... | 20 |
| 3.1.14 Javascript..... | 20 |
| 3.1.15 Node.js..... | 20 |
| 3.1.16 Postman..... | 20 |
| 3.1.17 MySql..... | 21 |
| 4 Antecedentes..... | 21 |
| 4.1 Análisis, diseño y desarrollo de un prototipo de software para la administración de parqueaderos..... | 21 |
| 4.2 Sistema de control vehicular utilizando reconocimiento óptico de caracteres..... | 21 |
| 4.3 Uso de las técnicas de la información y comunicación y su incidencia en el control de acceso | 22 |
| 5 Metodología..... | 22 |
| 6 Análisis..... | 25 |

DISEÑO DE SOFTWARE CONTROL Y GESTIÓN DEL PROCESO PARQUEADERO UIS

| | | |
|--------|---|----|
| 6.1 | Requerimientos del proyecto | 25 |
| 6.1.1 | Requerimientos funcionales | 26 |
| 6.2 | Análisis de requerimientos..... | 29 |
| 6.2.1 | Casos de uso..... | 29 |
| 6.2.2 | Definición del actor Usuario..... | 31 |
| 6.2.3 | Definición del actor moderador..... | 31 |
| 6.2.4 | Definición del actor administrador..... | 32 |
| 6.2.1 | Diagrama de actividades..... | 32 |
| 6.2.2 | Diagrama relacional | 36 |
| 6.2.3 | Diseño de la Arquitectura | 38 |
| 7 | Diseño..... | 39 |
| 7.1 | Diseño de la interfaz de usuario. | 39 |
| 7.1.1 | Diseño del inicio de sesión..... | 39 |
| 7.1.2 | Diseño de la pantalla de registro de usuarios..... | 42 |
| 7.1.3 | Diseño de la pantalla principal del usuario y menú..... | 43 |
| 7.1.4 | Diseño de pantalla creación, edición y eliminación de vehículos..... | 45 |
| 7.1.5 | Diseño de edición y eliminación de vehículos. | 46 |
| 7.1.6 | Diseño de pantalla QR generado por el usuario para ingreso al parqueadero. | 47 |
| 7.1.7 | Diseño de edición del perfil de usuario | 48 |
| 7.1.8 | Diseño de la pantalla principal del moderador, que a su vez es la pantalla para iniciar el proceso de registro de sesiones de parqueo y menú..... | 49 |
| 7.1.9 | Diseño de la pantalla de reporte de incidencias..... | 51 |
| 7.1.10 | Diseño de la pantalla de vehículos dentro del sistema..... | 52 |
| 7.1.11 | Diseño de la pantalla de detalles de vehículos dentro del sistema. | 53 |
| 7.1.12 | Diseño de la pantalla de registro de moderadores por parte del administrador..... | 54 |
| 7.1.13 | Diseño de la pantalla de visualización y eliminación de usuarios del sistema. | 55 |
| 7.1.14 | Diseño de la pantalla de visualización de reportes de incidencias..... | 56 |
| 8 | Desarrollo..... | 57 |
| 8.1 | Desarrollo de Software. | 57 |
| 8.1.1 | Registro de usuarios..... | 58 |
| 8.1.2 | Inicio de sesión..... | 59 |
| 8.1.3 | Visualización de vehículos..... | 60 |
| 8.1.4 | Visualización de sesiones de parqueo. | 65 |

DISEÑO DE SOFTWARE CONTROL Y GESTIÓN DEL PROCESO PARQUEADERO UIS

| | | |
|-------|------------------------------------|----|
| 8.1.5 | Registro de incidencias | 67 |
| 8.1.6 | Registro de moderador..... | 68 |
| 9 | Pruebas..... | 70 |
| 9.1 | Pruebas funcionales con carga..... | 70 |
| 9.2 | Pruebas de usabilidad..... | 71 |
| 10 | Conclusiones..... | 72 |
| 11 | Trabajo a futuro..... | 73 |
| | Referencias Bibliográficas..... | 74 |
| | Apéndices..... | 78 |
| | Encuesta..... | 79 |

Lista de figuras

Pág.

| | |
|---|----|
| Figura 1 <i>Formato de perdida de tiquete</i> | 15 |
| Figura 2 <i>Tablero de la metodología usada en el proyecto</i> | 24 |
| Figura 3 <i>Diagrama de Casos de uso</i> | 30 |
| Figura 4 <i>Diagrama de actividades para ingresar y salir del parqueadero</i> | 33 |
| Figura 5 <i>Diagrama de actividades para crear y ver una incidencia</i> | 35 |
| Figura 6 <i>Modelo relacional</i> | 37 |
| Figura 7 <i>Arquitectura del software</i> | 39 |
| Figura 8 <i>Logo de la aplicación</i> | 40 |
| Figura 9 <i>Bosquejo inicio de sesión</i> | 41 |
| Figura 10 <i>Bosquejo crear cuenta</i> | 42 |
| Figura 11 <i>Bosquejo pantalla principal</i> | 43 |
| Figura 12 <i>Bosquejo de menú</i> | 44 |
| Figura 13 <i>Bosquejo de creación de vehículos</i> | 45 |
| Figura 14 <i>Bosquejo edición y eliminación de vehículos</i> | 46 |
| Figura 15 <i>Bosquejo QR generado por el usuario para ingreso</i> | 47 |
| Figura 16 <i>Bosquejo de edición de perfil de usuario</i> | 48 |
| Figura 17 <i>Bosquejo pantalla principal Moderador</i> | 49 |
| Figura 18 <i>Bosquejo de menú del moderador</i> | 50 |
| Figura 19 <i>Bosquejo de la pantalla de reporte de incidencias</i> | 51 |
| Figura 20 <i>Bosquejo de la pantalla de vehículos dentro del sistema</i> | 52 |
| Figura 21 <i>Bosquejo de la pantalla de detalles de vehículos dentro del sistema</i> | 53 |
| Figura 22 <i>Bosquejo pantalla de registro de moderadores por parte del administrador</i> | 54 |
| Figura 23 <i>Bosquejo pantalla de visualización y eliminación de usuarios del sistema</i> | 55 |
| Figura 24 <i>Bosquejo de pantalla de visualización de reportes de incidencias</i> | 56 |
| Figura 25 <i>Vista Registro de usuarios</i> | 58 |
| Figura 26 <i>Vista inicio de sesión</i> | 59 |
| Figura 27 <i>Vista visualización de vehículos</i> | 60 |
| Figura 28 <i>Registro de vehículos</i> | 61 |
| Figura 29 <i>Edición y eliminación de vehículos</i> | 62 |
| Figura 30 <i>Vista QR a escanear</i> | 63 |
| Figura 31 <i>Vista Escaneo de QR</i> | 64 |
| Figura 32 <i>Vista de visualización de sesiones de parqueo</i> | 65 |
| Figura 33 <i>Detalles de la sesión de parqueo</i> | 66 |
| Figura 34 <i>Vista de registro de incidencias</i> | 67 |
| Figura 35 <i>Vista de registro de moderador</i> | 68 |
| Figura 36 <i>Visualización de sesiones por medio del Administrador</i> | 69 |

Lista de Tablas

| | Pág. |
|---|------|
| Tabla 1 <i>Requerimientos funcionales</i> | 26 |
| Tabla 2 <i>Requerimientos No funcionales</i> | 29 |
| Tabla 3 <i>Casos de uso Usuario</i> | 31 |
| Tabla 4 <i>Casos de uso moderador</i> | 31 |
| Tabla 5 <i>Casos de uso del administrador</i> | 32 |

Resumen

Título: Diseño de una herramienta software de gestión y control del proceso de entrada y salida de vehículos de la Universidad industrial de Santander*

Autores: Einer Steven Jaimes Mantilla y Jhoann Sebastián Martínez Oviedo**

Palabras Clave: Parqueadero, Control, Gestión, Base de datos, Método dinámico, Aplicación móvil, Vehículos.

Descripción: El parqueadero de la Universidad Industrial de Santander ofrece servicios de parqueo a toda su comunidad universitaria que cuenten con vehículos tales como: bicicletas, motocicletas y carros; este servicio es prestado de forma gratuita para todos los miembros UIS. Actualmente, el proceso de parqueadero de la universidad se lleva mediante el uso de tiquetes el cual es entregado al usuario por el guardia de seguridad una vez ingresa a la instalación y dicho guardia requiere este tiquete al momento de salida para que así el usuario se pueda retirar con su vehículo, si este tiquete se extravía se tienen que llenar formularios con información sensible lo cual vuelve este proceso bastante ineficiente e inseguro. Por lo tanto, se busca diseñar e implementar una herramienta que gestione y controle los procesos mencionados anteriormente y asegure la integridad de la información. La herramienta software será implementada utilizando la metodología Kanban y una arquitectura API rest. Para el desarrollo de este sistema utilizaremos el framework de frontend Flutter, el entorno en tiempo de ejecución multiplataforma de código abierto Node.js y la base de datos relacional MySQL Workbench.

*Proyecto de grado. Trabajo de Investigación.

**Facultad de Ingenierías Fisicomecánicas. Escuela de Ingeniería de Sistemas e Informática.

Abstract

Title: Design of a Software Tool for Managing and Controlling the In and Out Process of Vehicles at the Industrial University of Santander*

Authors: Einer Steven Jaimes Mantilla and Johann Sebastián Martínez Oviedo**

Keywords: Parking, Control, Management, Database, Dynamic Method, Mobile Application, Vehicles

Description: The parking lot of the Industrial University of Santander offers parking services to its entire university community that have vehicles such as: bicycles, motorcycles and cars; this service is provided for free to all UIS members. Currently, the university's parking process is carried out through the use of tickets, which is handed to the user by the security guard once they enter the facility and the security guard requires this ticket at the time of departure so that the user can leave with their vehicle. If this ticket is lost, forms have to be filled out with sensitive information, making this process quite inefficient and insecure.

Therefore, we seek to design and implement a tool that manages and controls the aforementioned processes and ensures the integrity of the information. The software tool will be implemented using the Kanban methodology and an Api rest architecture. For the development of this system, we will use the Flutter frontend framework, the open source cross-platform runtime environment Node.js, and the relational database MySQL Workbench.

*Degree project. Research Work.

**Faculty of Physical-Mechanical Engineering. School of Systems and Computer Engineering.

Director: MSc Jathinson Meneses Mendonza

Introducción

En la actualidad, el uso de tecnologías de información se ha convertido en una herramienta indispensable para mejorar los procesos administrativos de cualquier organización. La Universidad Industrial de Santander (UIS) no es la excepción, y en su parqueadero se presenta una problemática relacionada con la gestión y control de acceso de vehículos, debido a la ineficacia y falta de control en el manejo de los tiquetes entregados a los usuarios.

En este contexto, el objetivo principal de esta investigación es proponer una solución mediante la implementación de una herramienta software de control de acceso para el proceso de entrada y salida del parqueadero UIS, que permita la gestión y registro de los vehículos que utilizan el servicio. Además, se plantean objetivos específicos para definir los requerimientos del aplicativo, proponer una arquitectura de software y una base de datos, implementar el aplicativo y validar su funcionamiento mediante pruebas.

A través de una entrevista a los miembros de la organización, se identificaron varias problemáticas y falencias del sistema actual de parqueo. Entre ellas se encuentran la falta de registro de placas en un sistema consolidado, las horas pico que imposibilitan a los guardias seguir al pie de la letra ciertas órdenes, y el bajo control en el manejo de los tiquetes. Además, se evidenció la ineficacia en el manejo de escenarios como la pérdida de tiquetes y la falta de control en la identificación del propietario del vehículo.

Ante esta situación, se propone la implementación de un aplicativo móvil para el control de acceso al parqueadero UIS, que permita mejorar la gestión y control de los vehículos que utilizan el servicio. Esta herramienta se enfocará en reducir la intervención humana en el proceso y facilitar el registro y acceso a la información.

En resumen, esta tesis busca contribuir al mejoramiento de los procesos administrativos del parqueadero UIS, a través de la implementación de una herramienta tecnológica que permita la gestión y

control de acceso de los vehículos que utilizan el servicio, con el fin de reducir las problemáticas y falencias identificadas en el sistema actual de parqueo.

1 Objetivos

1.1 Objetivo General

Desarrollar e implementar una herramienta software de control de acceso que facilite el proceso de entrada y salida del parqueadero UIS, mediante el uso de tecnologías de información.

1.2 Objetivos Específicos

Identificar los requerimientos funcionales y no funcionales que debe cumplir el aplicativo móvil para garantizar su correcto desarrollo y construcción, basándose en los procesos administrativos de la Universidad.

Diseñar una arquitectura de software y una base de datos que se adapten a los requerimientos propuestos.

Implementar el aplicativo de gestión y administración de parqueaderos a partir de los diseños propuestos.

Validar el prototipo mediante un conjunto de pruebas de software que permitan comprobar su correcto funcionamiento.

2 Planteamiento de problema

El parqueadero de la universidad industrial de Santander ofrece servicios de parqueo a toda su comunidad universitaria que cuenten con vehículos tales como: bicicletas, motocicletas y carros; este servicio es prestado de forma gratuita para todos los miembros UIS, el proceso de parqueadero de la universidad se lleva mediante el uso de tiquetes el cual es entregado al usuario por el guardia de seguridad una vez ingresa a la instalación y dicho guardia requiere este tiquete al momento de salida para que así el usuario se pueda retirar con su vehículo, acá podemos identificar un problema de manejo del tiempo, así como el control que los guardias pueden ejercer.

En una entrevista realizada a miembros de la organización se pudieron identificar varias problemáticas y falencias dentro de su sistema las cuales ayudaron a entender mejor sus procesos internos y como se podrían mejorar ciertos aspectos por medio de un sistema de información.

Problemáticas tales como el no registro de placas en un sistema consolidado; las horas pico que imposibilitan a los guardias seguir al pie de la letra ciertas ordenes que deben cumplir las cuales serían solicitar identificación a estudiantes UIS (carné) y solicitar el destino a los visitantes que deseen usar el servicio; el bajo control que se tiene en los tiquetes los cuales la única diferencia que tienen para llevar un control día a día es cambiar el color en sus fechas, y colores diferentes entre la UIS y el parqueadero del estadio Alfonso López; entre otras.

Adicionalmente se encuentran falencias y vulnerabilidades dentro del sistema que se viene manejando tales como:

la ineficacia al momento de manejar el escenario de pérdida del tiquete el cual conlleva cierto proceso para poder sacar su vehículo, en este caso se debe diligenciar el formato de pérdida de tiquete (Figura 1), sacar una copia a la tarjeta de propiedad y copia de la cédula de la persona haciéndose tedioso para el usuario. Una vez hecho esto los documentos solicitados son almacenados en físico teniendo así la posibilidad de pérdida de estos.

Figura 1*Formato de perdida de tiquete*

| | | | | |
|---|------------------------------|--|------|----------------|
|   | | PERDIDA TIQUETE DE VEHÍCULO | | Código: FRF-14 |
| NOMBRE Y APELLIDOS: | | PROCESO DE RECURSOS FÍSICOS SUBPROCESO DE SEGURIDAD | | Versión: 03 |
| HORA NOVEDAD: | PARQUEADERO N°: | FECHA: | CC.: | |
| TARJETA DE PROPIEDAD N°: | UAA A LA QUE ESTA VINCULADO: | | | |
| TIPO DE VEHICULO: | PLACA: | COLOR: | | |
| OBSERVACIONES (OPCIONAL) | | | | |
| FIRMA PROPIETARIO | | FIRMA DE CELADOR | | |
| NOTA: ADJUNTO A ESTE FORMATO DEBE IR LA FOTOCOPIA DE: DOCUMENTO DE IDENTIFICACIÓN Y TARJETA DE PROPIEDAD DEL VEHICULO | | | | |

Y nulo control que se tiene para poder identificar si realmente la persona que se retira con un vehículo es el propietario de este, dado que si se presenta el tiquete en el momento de salida no habrá ningún inconveniente, ya que el único filtro de control que se tiene estaría superado (el tiquete).

Es por esto que se hace necesario proponer alternativas sobre la gestión y control que se lleva con el actual sistema de parqueo tales como la implementación de un aplicativo móvil ya que hoy en día los dispositivos móviles se han vuelto casi que indispensables y de fácil acceso para tareas cotidianas; adicionalmente la reducción de intervención humana durante el proceso.

Es por lo anterior que surge la duda de ¿Cómo se puede llevar un inventario o registro al cual se pueda acceder para mantener una correcta gestión y manejo que se le da al servicio de parqueadero?

3 Marco de referencia

3.1 Marco conceptual

Para una mejor comprensión de las funcionalidades y el desarrollo del proyecto se deben tener en cuenta los siguientes conceptos:

3.1.1 Desarrollo de aplicaciones móviles

El desarrollo de aplicaciones móviles son el conjunto de procesos y procedimientos que involucran la creación de software para dispositivos informáticos pequeños e inalámbricos, tales como teléfonos inteligentes y otros dispositivos portátiles. Existen diversos tipos de tecnologías para el desarrollo de este tipo de software, entre ellos se encuentran las aplicaciones nativas, las cuales se crean utilizando entornos de desarrollo integrados (IDE) e idiomas para sistemas operativos móviles como Google Android y Apple iOS (David, 2021). Tal como ocurre en el desarrollo web, este tipo de desarrollo se divide, generalmente, en Frontend (la parte que involucra al cliente) y Backend (la parte que involucra al servidor).

3.1.2 Frontend

El Frontend de un sitio o aplicación contiene el área en la que interactúa el usuario. Su objetivo principal es trabajar con los recursos de datos del backend para ofrecer información, permitir al usuario tomar decisiones y consumir contenido, con herramientas como botones, medios y calculadoras.

Llamamos Frontend al lado del cliente del sitio web o de la aplicación, ya que el usuario suele navegar por el Frontend con un navegador u otra interfaz (Warnimont, 2022).

3.1.3 API REST

Una API REST es una manera de permitir que diferentes programas de ordenador se comuniquen entre sí a través de internet. Ya que la comunicación debe darse a través de protocolos y estándares para

enviar y recibir datos, estas APIs están diseñadas bajo los principios de REST (Representational State Transfer) y son útiles para operaciones simples (Platzi, s. f.).

3.1.4 Sistema de información

Se plantea la definición técnica de un sistema de información como un conjunto de componentes interrelacionados que reúnen, procesan, almacenan y distribuyen información para apoyar los procesos de toma de decisiones y de control en una organización. Ayuda también a la toma de decisiones, la coordinación y el control, los sistemas de información también pueden ayudar a las corporaciones a analizar problemas, visualizar temas complejos y crear nuevos productos. (kionetworks)

3.1.5 Tecnología de información

La tecnología de la información consiste en todo el hardware y software que una empresa necesita usar para poder cumplir con sus objetivos de negocios. Esto incluye tanto a los equipos de cómputo como a los dispositivos de almacenamiento y los dispositivos móviles de bolsillo, también a los componentes de software, como los sistemas operativos Windows o Linux, entre otros. (educaciongepper)

3.1.6 Apis

Las API son conjuntos de definiciones y protocolos que se utilizan para diseñar e integrar el software de las aplicaciones. Se considera como el contrato entre el proveedor de información y el usuario, donde se establece a partir de una llamada por parte del usuario y la respuesta.

Cuando el cliente envía una solicitud a través de una API de RESTful, esta transfiere una representación del estado del recurso requerido a quien lo haya solicitado. (redhat)

3.1.7 Bases de datos

En teoría la definición de bases de datos incluye los principios formales para definir y manipular datos estructurados e interrelacionados. Para determinar los datos se utiliza un modelo de datos y para su manipulación un lenguaje. Diferentes modelos de datos se han propuesto buscando un mayor nivel expresivo para representar el mundo real. La potencia y limitaciones de cada modelo se pueden evaluar desde un punto de vista teórico y se evidencian desde un punto de vista práctico cuando se trata de implementarlos en aplicaciones tradicionales y modernas. (univalle)

3.1.8 Modelo Relacional

Las bases de datos relacionales se basan en la organización de la información en partes pequeñas que se integran mediante identificadores; a diferencia de las bases de datos no relacionales que, como su nombre lo indica, no tienen un identificador que sirva para relacionar dos o más conjuntos de datos. Además, las bases de datos relacionales son más robustas, es decir, tienen mayor capacidad de almacenamiento, y son menos vulnerables ante fallas, estas son sus principales características.

Las relaciones se asocian con tablas nombradas cuyas columnas representan atributos que también pueden tener asociado un nombre. Las filas de las tablas son tuplas. Los valores que toman las tuplas se extraen de conjuntos de constantes llamados dominios. (Pragma)

3.1.9 Flutter

Flutter es un Framework de código abierto de Google para crear aplicaciones compiladas de forma nativa y multiplataforma (móvil, web, escritorio y dispositivos integrados) a partir de una única base de código. El código de Flutter se compila en código máquina ARM o Intel, así como en JavaScript, para obtener un rendimiento rápido en cualquier dispositivo (Flutter, s.f.).

3.1.10 Android Studio

Android Studio es el entorno de desarrollo integrado (IDE) oficial para el desarrollo de aplicaciones para Android (Desarrolladores de Android, 2022). Proporciona ayuda y plantillas que verifican los requisitos del sistema, como el JDK y la memoria RAM disponible, y configuran los ajustes predeterminados (Desarrolladores de Android, 2021).

3.1.11 Github

Github es una plataforma de desarrollo colaborativo que permite crear repositorios web para alojar proyectos y que otros usuarios puedan acceder a ellos. Esta plataforma permite alojar programas o sistemas operativos, para que la comunidad acceda a los códigos fuente y realice comentarios. Dependiendo del acceso que otorgue el dueño de la cuenta, los usuarios pueden realizar modificaciones que quedan registradas gracias al sistema de control de versiones Git (Next U, 2022).

3.1.12 Visual Studio Code

Visual Studio Code es un potente editor de código fuente que se ejecuta en el escritorio y está disponible para Windows, macOS y Linux. Posee un largo ecosistema de extensiones para una gran variedad de lenguajes y tiempos de ejecución (Visual Studio Code, 2021).

3.1.13 Dart

Dart es un lenguaje de cliente optimizado, gratuito y de código abierto, para aplicaciones rápidas en cualquier plataforma. Se trata de un lenguaje de programación especializado en torno a las necesidades de creación de interfaces de usuario. Dart se compila en código máquina ARM y x64 para móvil, escritorio y Backend, y en JavaScript para la web (Dart programming language, s.f.).

3.1.14 Javascript

JavaScript es un lenguaje de programación que los desarrolladores utilizan para hacer páginas web interactivas. Desde actualizar fuentes de redes sociales a mostrar animaciones y mapas interactivos, las funciones de JavaScript pueden mejorar la experiencia del usuario de un sitio web. (amazon)

3.1.15 Node.js

Ideado como un entorno de ejecución de JavaScript orientado a eventos asíncronos, Node.js está diseñado para crear aplicaciones network escalables. Con esta tecnología se hizo la creación del API REST. (pispos)

3.1.16 Postman

Postman es una plataforma con interfaz gráfica para construir y consumir API's. Postman simplifica cada paso del ciclo de vida de la API y agiliza la colaboración para que pueda crear con eficacia mejores API. (openwebinar)

3.1.17 *MySql*

MySQL es un sistema de gestión de bases de datos relacionales que cuenta con la característica de ser código abierto. MySQL es un sistema cliente/servidor que consiste en un servidor SQL de subprocesos múltiples que admite diferentes back-ends varios programas y bibliotecas de clientes diferentes, herramientas administrativas y una amplia gama de interfaces de programación de aplicaciones. (openweinar)

4 Antecedentes

Referente a este tema se encontraron 3 proyectos de grado realizados en diferentes universidades a nivel nacional, en estos se solucionan un problema similar al nuestro, pero usando inteligencia artificial e IoT.

4.1 Análisis, diseño y desarrollo de un prototipo de software para la administración de parqueaderos

En este proyecto de investigación llevado a cabo por Guillermo R Solarte, Frank Silva Castro y Luis E Muñoz de la universidad tecnológica de Pereira, ellos desarrollaron u una aplicación prototipo de escritorio que buscar facilitar el proceso del alquiler de parqueaderos.

4.2 Sistema de control vehicular utilizando reconocimiento óptico de caracteres

En este trabajo de investigación a cargo de German Benavides Calderón de la Universidad Distrital, en este se utilizan visión por computador para las placas de los vehículos que asisten a lugares como parqueaderos, peajes, avenidas con alta congestión con el fin de controlar la seguridad.

4.3 Uso de las técnicas de la información y comunicación y su incidencia en el control de acceso

En este trabajo de investigación utilizan IoT para controlar el ingreso de vehículos tales como sensores que transfieren la información a otro dispositivo que da el acceso al parqueadero, para nuestro caso no es de mucha utilidad debido a que no utilizaremos IoT.

5 Metodología

La metodología Kanban se utilizó en este proyecto de desarrollo e implementación de una herramienta software de control de acceso para el parqueadero UIS. Kanban es una metodología ágil que se enfoca en la mejora continua y el trabajo en equipo para garantizar la entrega de productos de alta calidad de manera eficiente.

El proceso de desarrollo del proyecto se dividió en cuatro fases principales: análisis, diseño, desarrollo y pruebas. Durante la fase de análisis, se identificaron los requerimientos funcionales y no funcionales que debía cumplir el aplicativo móvil para garantizar su correcto desarrollo y construcción. En esta fase se definieron las características, funcionalidades y requisitos que debía cumplir el software.

En la fase de diseño, se utilizó la información recopilada en la fase de análisis para crear una arquitectura de software y una base de datos que se adaptaran a los requerimientos propuestos. Durante esta fase, se diseñó una interfaz de usuario que permitiera a los usuarios interactuar de manera intuitiva y sencilla con la aplicación.

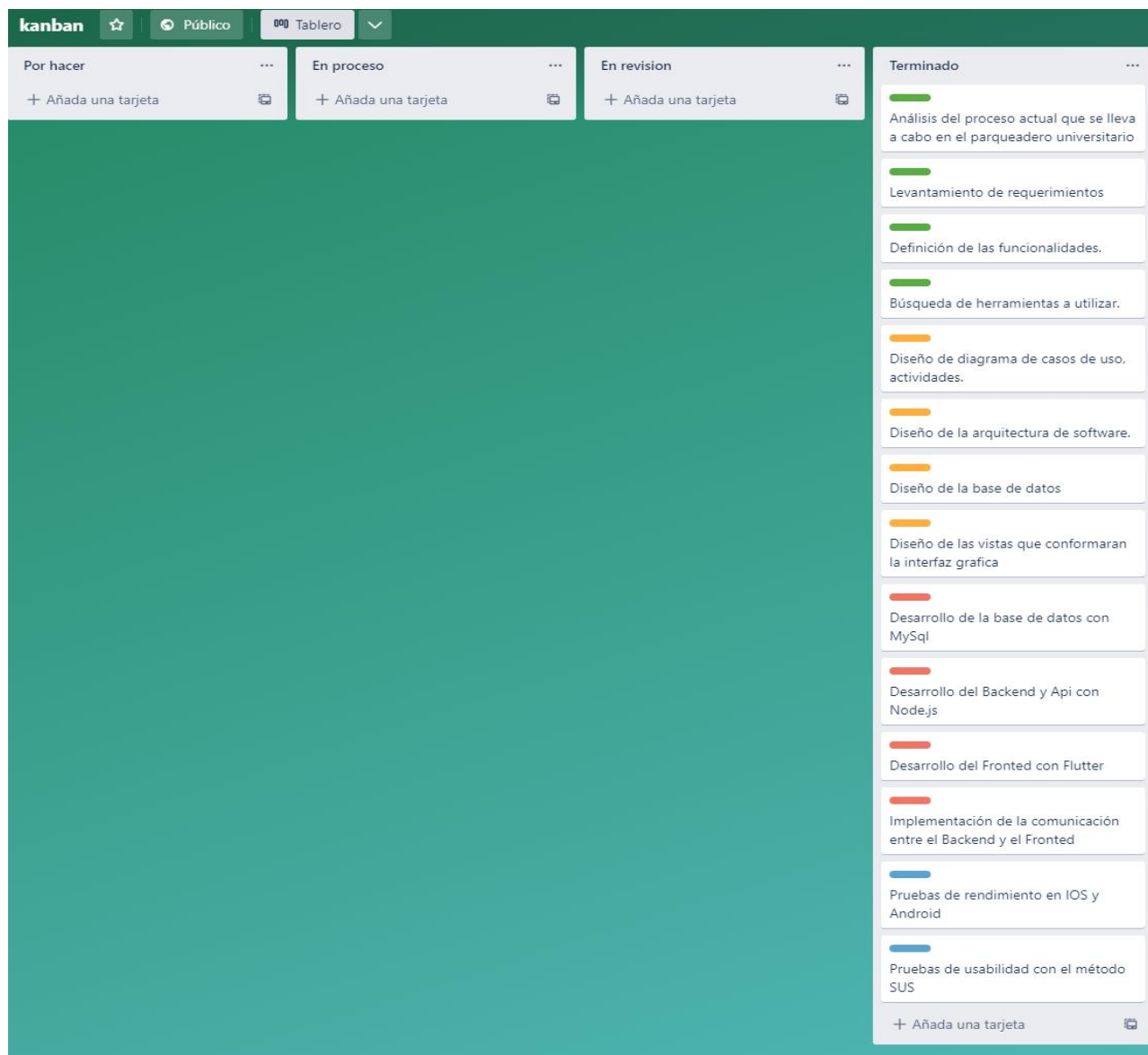
La fase de desarrollo se enfocó en la implementación del aplicativo de gestión y administración de parqueaderos a partir de los diseños propuestos. Los desarrolladores trabajaron en conjunto con el equipo de diseño para asegurar que la aplicación cumpliera con los requerimientos definidos en las fases anteriores.

Finalmente, la fase de pruebas se encargó de validar el prototipo mediante un conjunto de pruebas de software que permitieron comprobar su correcto funcionamiento. Durante esta fase, se evaluaron aspectos como la usabilidad y el rendimiento.

La metodología Kanban permitió una gestión eficiente del proceso de desarrollo, ya que se pudo visualizar y priorizar el trabajo en tiempo real, garantizando la entrega de un prototipo de alta calidad en un plazo definido. Además, se fomentó la colaboración y el trabajo en equipo, lo que permitió la toma de decisiones más informadas y una mayor implicación en el proyecto por parte de todos los miembros del equipo.

Figura 2

Tablero de la metodología usada en el proyecto



6 Análisis

Durante la fase de análisis se realizaron diversas actividades con el fin de identificar los requerimientos funcionales y no funcionales que debían cumplir el aplicativo móvil para garantizar su correcto desarrollo y construcción. Se llevaron a cabo entrevistas con diferentes usuarios del parqueadero UIS, se revisaron los procesos administrativos de la universidad y se evaluaron las tecnologías disponibles para la construcción del software.

Durante este proceso se identificaron las necesidades de los usuarios y se establecieron los requerimientos necesarios para el correcto funcionamiento del software. Se analizaron diferentes opciones de arquitectura de software y se evaluaron las posibles soluciones de bases de datos que se adaptaran a los requerimientos propuestos.

Todo este proceso de análisis fue crucial para el éxito del proyecto, ya que permitió establecer una base sólida para el diseño, desarrollo e implementación del software. Se logró tener una comprensión profunda de las necesidades de los usuarios y se definieron los requerimientos para la construcción del aplicativo móvil de control de acceso al parqueadero UIS.

6.1 Requerimientos del proyecto

En este apartado se presenta el listado de requerimientos principales obtenidos al analizar el funcionamiento actual del servicio en el parqueadero. Estos requerimientos se dividen en funcionales, los cuales enuncian los servicios que el sistema debe proveer, y no funcionales, los cuales abarcan las limitaciones sobre servicios o funciones que ofrece el sistema (Sommerville, 2011, pp. 84-85).

6.1.1 *Requerimientos funcionales*

Tabla 1

Requerimientos funcionales

| | | | |
|--|-----------------|---------------------------------|--|
| Requerimiento: RF01 | Prioridad: Alta | Nombre: Registrar usuario | Rol: Usuarios |
| Descripción: Los usuarios podrán registrarse ingresando la siguiente información: nombre, identificación, correo electrónico, contraseña y eligiendo el rol en la universidad (estudiante, docente, visitante, administrativo) | | | |
| Requerimiento: RF02 | Prioridad: Alta | Nombre: Iniciar Sesión | Rol: Usuario, Moderador y administrador. |
| Descripción: Los usuarios, moderadores y administrador registrados podrán iniciar sesión ingresando la siguiente información: correo y contraseña. | | | |
| Requerimiento: RF03 | Prioridad: Alta | Nombre: Cerrar Sesión | Rol: Usuario, Moderador y administrador. |
| Descripción: Los usuarios que hayan iniciado sesión podrán cerrarla. | | | |
| Requerimiento: RF04 | Prioridad: Alta | Nombre: Editar datos personales | Rol: Usuario |
| Descripción: Los usuarios que hayan iniciado sesión podrán visualizar y editar los datos asociados a su cuenta. | | | |
| Requerimiento: RF05 | Prioridad: Alta | Nombre: Cambiar contraseña | Rol: Usuario |

DISEÑO DE SOFTWARE CONTROL Y GESTIÓN DEL PROCESO PARQUEADERO UIS

| | | | |
|--|------------------|-----------------------------------|--|
| Descripción: Los usuarios que hayan iniciado sesión podrán cambiar su contraseña, ingresando la contraseña actual y la nueva contraseña. | | | |
| Requerimiento: RF06 | Prioridad: Alta | Nombre: Crear vehículo | Rol: Usuario |
| Descripción: Los usuarios podrán crear vehículos, ingresando la placa, el color, tipo de vehículo y una foto del mismo. Dependiendo del tipo de vehículo se deberá ingresar la placa. | | | |
| Requerimiento: RF07 | Prioridad: Alta | Nombre: Editar vehículo | Rol: Usuario |
| Descripción: Los usuarios podrán editar vehículos, ingresando la placa, el color, tipo de vehículo y una foto del mismo. | | | |
| Requerimiento: RF08 | Prioridad: Alta | Nombre: Eliminar vehículo | Rol: Usuario |
| Descripción: Los usuarios podrán eliminar los vehículos previamente creados. | | | |
| Requerimiento: RF09 | Prioridad: Alta | Nombre: Mostrar código QR | Rol: Usuario |
| Descripción: Los usuarios podrán elegir uno de sus vehículos para mostrar el respectivo código QR para ingresar o salir del parqueadero. | | | |
| Requerimiento: RF10 | Prioridad: Media | Nombre: Cargar foto de perfil | Rol: Usuario |
| Descripción: Los usuarios podrán elegir una imagen para colocar en su perfil de usuario. | | | |
| Requerimiento: RF11 | Prioridad: Baja | Nombre: Cambiar modo día y noche. | Rol: Usuario, Moderador, administrador |
| Descripción: Los usuarios, moderadores y administrador podrán cambiar la apariencia de la aplicación entre el modo día y noche. | | | |
| Requerimiento: RF12 | Prioridad: Alta | Nombre: Escanear código QR | Rol: Moderador |
| Descripción: Los moderadores podrán escanear un código QR usando la cámara del dispositivo móvil, para dar ingreso y salida a los vehículos de los usuarios. | | | |
| Requerimiento: RF13 | Prioridad: Alta | Nombre: Crear Incidencias | Rol: Moderador |

DISEÑO DE SOFTWARE CONTROL Y GESTIÓN DEL PROCESO PARQUEADERO UIS

| | | | |
|---|-----------------|---|----------------------------------|
| <p>Descripción: Los moderadores podrán crear incidencias a partir del escaneo de los códigos QR de los vehículos implicados en dicha incidencia. Para crear una incidencia se debe elegir el tipo de incidencia y una descripción.</p> | | | |
| Requerimiento: RF14 | Prioridad: Alta | Nombre: Ver sesiones de parqueo | Rol: Moderador, administrador |
| <p>Descripción: Los moderadores podrán observar los vehículos que han ingresado al parqueadero asignado, con la información de su foto, placa, fecha y hora de entrada. Adicionalmente podrán ver la capacidad restante del parqueadero.</p> <p>El administrador podrá observar todos los vehículos de los diferentes parqueaderos.</p> | | | |
| Requerimiento: RF15 | Prioridad: Alta | Nombre: Añadir moderadores | Rol: Administrador |
| <p>Descripción: El administrador podrá agregar nuevos moderadores.</p> | | | |
| Requerimiento: RF16 | Prioridad: Alta | Nombre: Eliminar usuarios y moderadores | Rol: Administrador |
| <p>Descripción: El administrador podrá eliminar usuarios y moderadores.</p> | | | |
| Requerimiento: RF17 | Prioridad: Alta | Nombre: Ver incidencias | Rol: Administrador |
| <p>Descripción: El administrador podrá ver las incidencias que han creado los moderadores.</p> | | | |
| Requerimiento: RF18 | Prioridad: Alta | Nombre: Dar ingreso o salida de vehículos | Rol: Moderadores |
| <p>Descripción: El moderador podrá dar ingreso o salida de vehículos que cumplan con la información escaneada del código QR.</p> | | | |
| Requerimiento: RF19 | Prioridad: Alta | Nombre: Eliminar sesión de parqueo | Rol: Moderadores y administrador |
| <p>Descripción: los moderadores podrán eliminar sesiones de parqueo al cual están asignados. El administrador puede eliminar cualquier sesión de parqueo.</p> | | | |

Tabla 2*Requerimientos No funcionales*

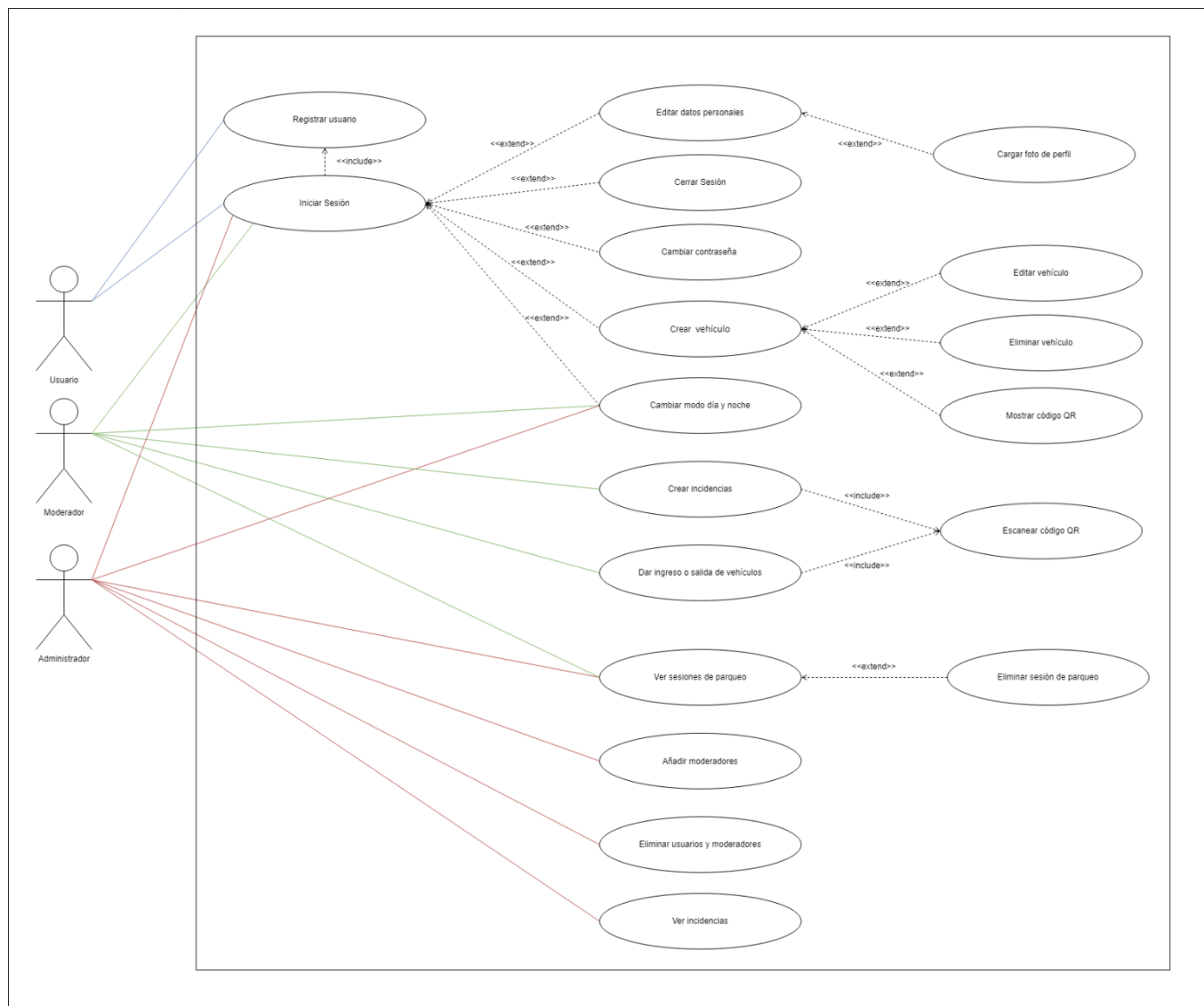
| | | |
|---|------------------|---|
| Requerimiento: RN01 | Prioridad: Alta | Nombre: Acceder desde cualquier dispositivo Android y IOS |
| Descripción: Los usuarios registrados podrán acceder al software desde cualquier dispositivo con sistema operativo Android 6.0 o superior y una versión de IOS 10 o superior. | | |
| Requerimiento: RN02 | Prioridad: Medio | Nombre: Capacidad máxima vehículos |
| Descripción: Límite de vehículos por usuario será de 5 vehículos. Límite de vehículos por parqueaderos serán 1000. | | |
| Requerimiento: RN03 | Prioridad: Medio | Nombre: Estado del vehículo |
| Descripción: El vehículo debe quedar en estado de salida hasta que se escanee nuevamente el código QR después de que se ingrese al sistema, en caso contrario la aplicación no podrá permitir el ingreso con el código QR generado por el vehículo. | | |

6.2 Análisis de requerimientos

Con el fin de analizar los requerimientos presentados, se realizaron diagramas de caso de uso, diagramas de actividades, diagrama relacional y diagrama de arquitectura que proporcionan un mejor entendimiento de las funcionalidades del sistema

6.2.1 Casos de uso

Los casos de uso modelan funcionalidades y sus relaciones, y su especificación textual especifica cómo serán implementados los requerimientos definidos. Cada funcionalidad es asignada a un actor, el cual a partir del uso del sistema genera el evento que ejecuta dicha funcionalidad (Pantaleo y Rinaudo, 2015, p. 135).

Figura 3*Diagrama de Casos de uso*

Para un mejor entendimiento del diagrama de casos de uso se muestra una lista de los actores involucrados con sus respectivos requerimientos funcionales.

6.2.2 Definición del actor Usuario

Tabla 3

Casos de uso Usuario

| | |
|----------------------------|--|
| Actor | Usuario |
| Requerimientos funcionales | RF01: Registrar usuario RF02: Iniciar Sesión RF03: Cerrar Sesión RF04: Editar datos personales RF05: Cambiar contraseña RF06: Crear vehículo RF07: Editar vehículo RF08: Eliminar vehículo RF09: Mostrar código QR RF10: Cargar foto de perfil RF11: Cambiar modo día y noche. |
| descripción | Usuario: Son aquellos miembros que usan el sistema para ingresar y salir del parqueadero. |

6.2.3 Definición del actor moderador

Tabla 4

Casos de uso moderador

| | |
|----------------------------|---|
| Actor | Moderador |
| Requerimientos funcionales | RF02: Iniciar Sesión RF03: Cerrar Sesión RF11: Cambiar modo día y noche. RF12: Escanear código QR RF13: Crear Incidencias RF14: Ver sesiones de parqueo RF18: Dar ingreso o salida de vehículos RF19: Eliminar sesión de parqueo |
| descripción | Moderador: Son aquellos miembros del cuerpo de seguridad de la universidad que controlan la entrada y salida de vehículos de los diferentes parqueaderos. |

6.2.4 Definición del actor administrador

Tabla 5

Casos de uso del administrador

| | |
|----------------------------|--|
| Actor | Administrador |
| Requerimientos funcionales | RF02: Iniciar Sesión RF03: Cerrar Sesión RF11: Cambiar modo día y noche. RF14: Ver sesiones de parqueo RF15: Añadir moderadores RF16: Eliminar usuarios y moderadores RF17: Ver incidencias RF18: Dar ingreso o salida de vehículos RF19: Eliminar sesión de parqueo |
| descripción | Administrador: Es el encargado de gestionar las sesiones de parqueo de todos los parqueaderos y la asignación de nuevos moderadores. |

6.2.1 Diagrama de actividades

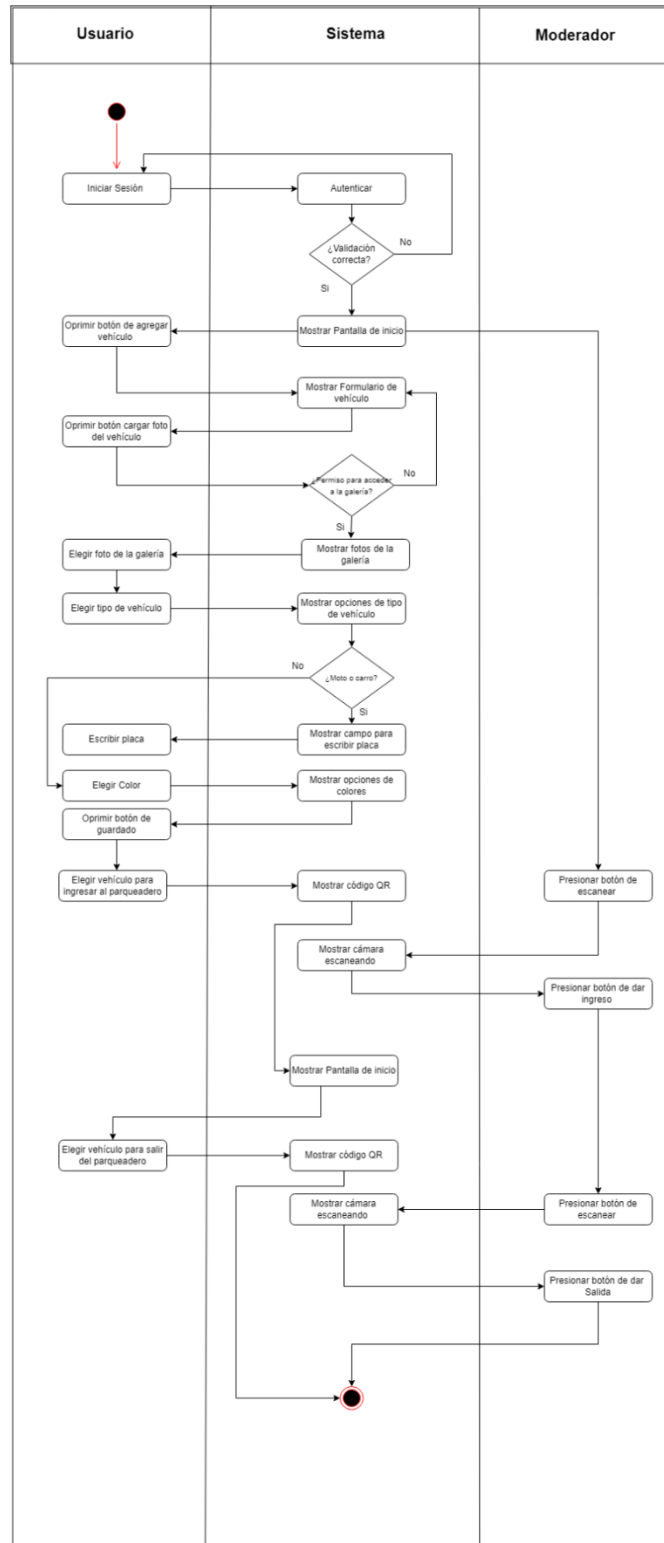
En las siguientes figuras se muestran los diagramas de actividades relacionados con las principales funcionalidades del sistema.

6.2.1.1 Diagrama de actividades para ingresar y salir del parqueadero

Esta actividad consiste en que el usuario debe ingresar con su cuenta, crear un vehículo con sus respectivos datos y posteriormente el moderador le dará ingreso al parqueadero escaneando el código QR del vehículo del usuario, finalmente el usuario saldrá del parqueadero y el moderador le dará salida escaneando el código QR del vehículo del usuario.

Figura 4

Diagrama de actividades para ingresar y salir del parqueadero

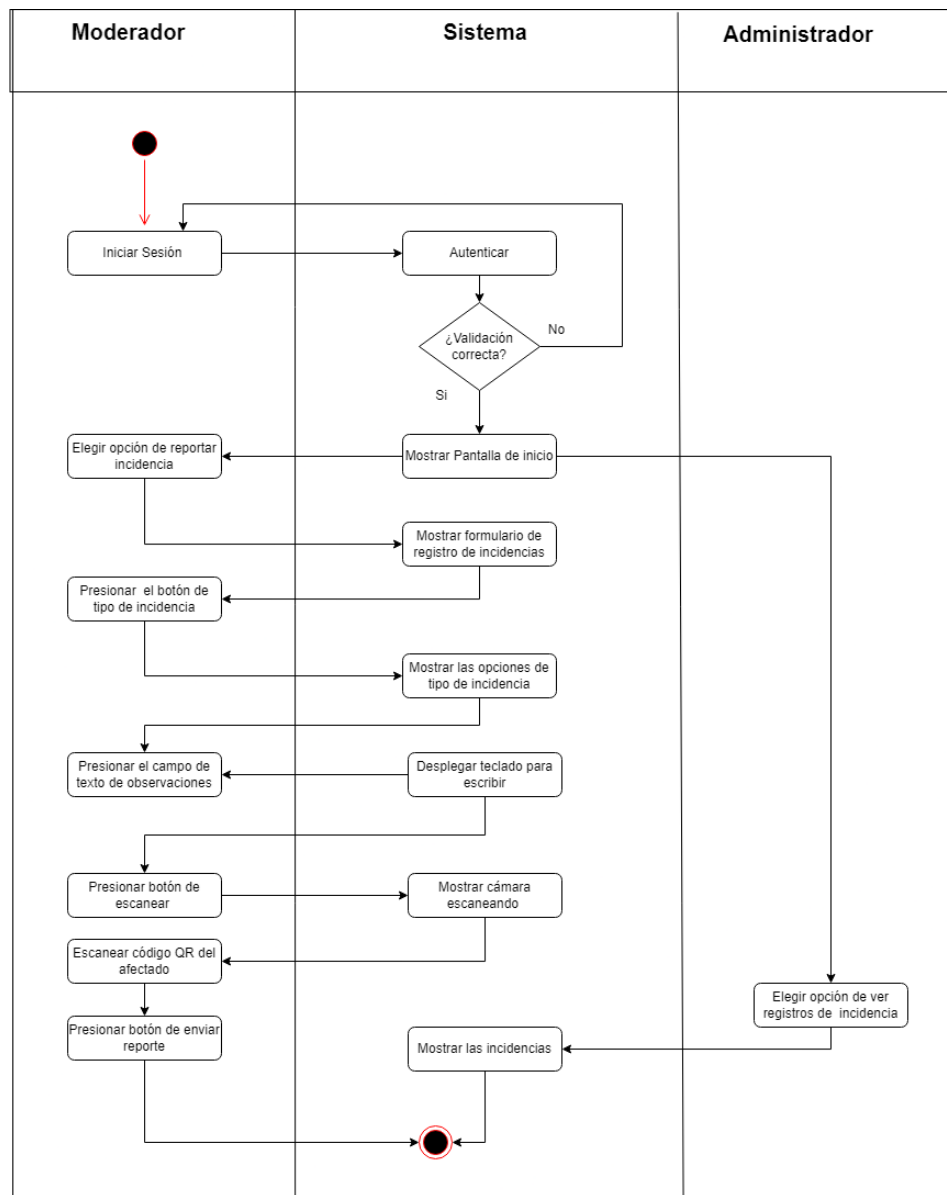


6.2.1.2 Diagrama de actividades para crear y ver una incidencia

Esta actividad consiste en que el moderador podrá crear una incidencia ocurrida en el parqueadero, en la cual se llenaran información respecto a la incidencia y para identificar al vehículo afectado, se le escaneara el código QR, por último, el Administrador es el único que puede observar todas las incidencias creadas por los moderadores en los diferentes parqueaderos.

Figura 5

Diagrama de actividades para crear y ver una incidencia

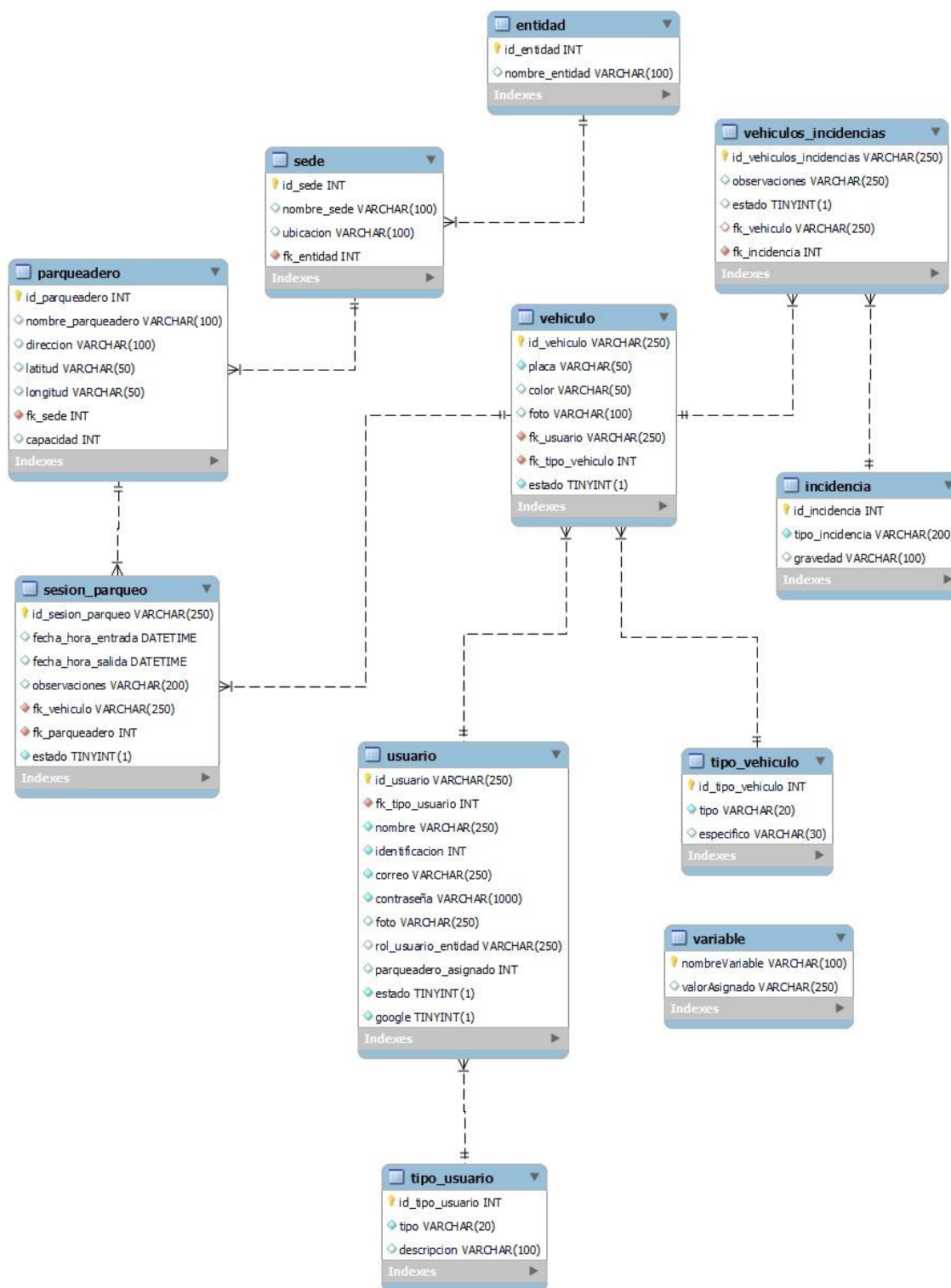


6.2.2 *Diagrama relacional*

En el desarrollo de esta aplicación, se creó un diagrama relacional que representa la estructura de la base de datos utilizada en la implementación del aplicativo. Este diagrama muestra las tablas que conforman la base de datos, así como las relaciones entre ellas. Se puede observar cómo se modelaron las entidades involucradas en el proceso de parqueadero universitario, tales como el usuario, el vehículo, la entrada y salida del parqueadero, entre otros. Además, se establecieron las llaves primarias y foráneas necesarias para asegurar la integridad de la información. El diagrama relacional fue una herramienta fundamental en la fase de desarrollo del aplicativo, ya que permitió visualizar la estructura de la base de datos y definir la forma en que se relacionan las distintas entidades.

Figura 6

Modelo relacional



6.2.3 *Diseño de la Arquitectura*

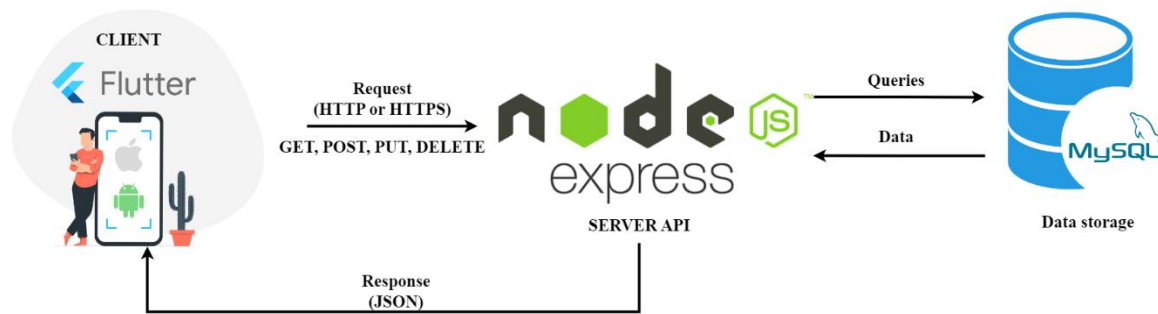
La arquitectura de software utilizada en este proyecto se basó en una API REST, la cual se compone de tres principales componentes: el servidor, el cliente y la base de datos.

El servidor fue desarrollado con Node.js, una plataforma de desarrollo en JavaScript que permite la construcción de servidores escalables y eficientes. El servidor se encarga de procesar y responder a las solicitudes del cliente, así como de interactuar con la base de datos.

El cliente fue desarrollado en Flutter, un framework de desarrollo de aplicaciones móviles multiplataforma que utiliza el lenguaje de programación Dart. El cliente se encarga de presentar la información al usuario final y de enviar solicitudes al servidor para obtener o enviar información.

La base de datos utilizada fue MySQL, un sistema de gestión de bases de datos relacional ampliamente utilizado en la industria. La base de datos se encarga de almacenar toda la información relevante del sistema, desde los usuarios hasta la información de los vehículos y las sesiones de parqueo.

En conjunto, estos tres componentes permitieron la creación de una arquitectura escalable, robusta y eficiente, capaz de manejar grandes cantidades de datos y de responder a las solicitudes de los usuarios de manera rápida y eficiente. Además, la elección de tecnologías ampliamente utilizadas en la industria permitió una fácil integración con otras herramientas y sistemas, lo que permitió una mayor flexibilidad y capacidad de adaptación en el futuro.

Figura 7*Arquitectura del software*

7 Diseño.

7.1 Diseño de la interfaz de usuario.

Para el inicio de este diseño se tuvo en cuenta la paleta de colores dispuesta por la universidad industrial de Santander. En este apartado podemos encontrar el diseño de las diferentes pantallas según los roles (Usuario, Moderador y Administrador).

7.1.1 Diseño del inicio de sesión.

Se diseñó la pantalla de inicio de sesión haciendo uso del logo prototipo:

Figura 8

Logo de la aplicación



(S/f). Com.co. Recuperado el 15 de abril de 2023, de <https://bydauto.com.co/blog/wp-content/uploads/2021/03/simbolo-de-parqueadero-exclusivo-para-carros-electricos.jpg>.

Junto con el nombre del aplicativo y los formularios necesarios para el inicio de sesión, además de la opción de dirigirse a la pantalla de registro (crear una nueva cuenta).

Figura 9*Bosquejo inicio de sesión*

P

PARKUIS

Usuario

Contraseña

Iniciar sesión

[crear una nueva cuenta](#)

7.1.2 *Diseño de la pantalla de registro de usuarios.*

Se diseña el registro de usuario (Estudiante, docente, administrativos y visitantes) con su respectivo formulario según los requerimientos del sistema.

Figura 10

Bosquejo crear cuenta

crear cuenta

7.1.3 *Diseño de la pantalla principal del usuario y menú.*

Se hizo el diseño de la pantalla principal teniendo en cuenta lo que el usuario debe tener a la mano al momento de iniciar sesión, evidenciando así principalmente los vehículos que se encuentran registrados por el usuario, seguido de botones los cuales permitirán la edición y creación de vehículos según conveniencia, junto con un menú lateral izquierdo desplegable en el cual va incluido el logo de la universidad y sus respectivas opciones.

Figura 11

Bosquejo pantalla principal

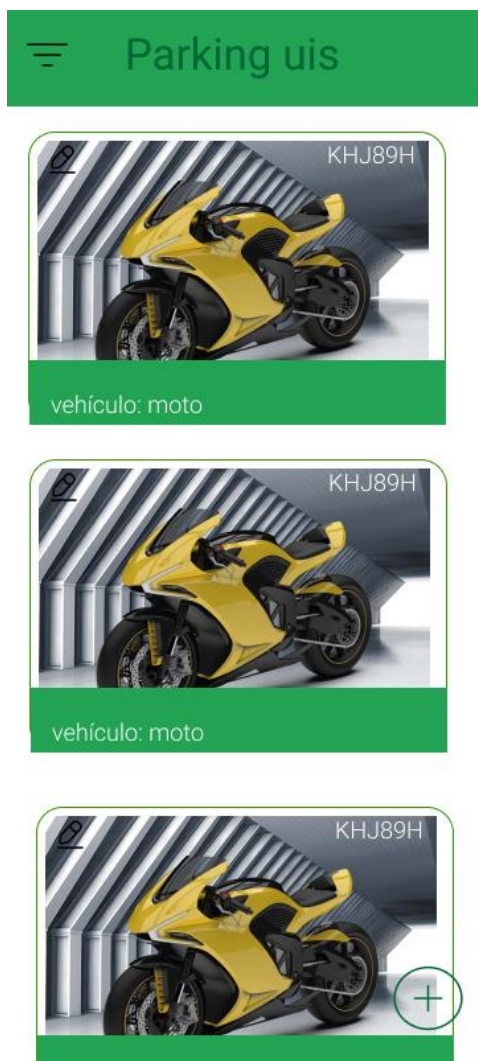


Figura 12

Bosquejo de menú



7.1.4 *Diseño de pantalla creación, edición y eliminación de vehículos.*

En este diseño se puede ver los diferentes atributos del vehículo que se quiere registrar o bien editar u eliminar, atributos tales como placa, color e imagen del vehículo.

Figura 13

Bosquejo de creación de vehículos



☰ Parking uis

+

Placa

selecciona un tipo de vehiculo ▾

selecciona el color de su vehiculo ▾

💾

7.1.5 *Diseño de edición y eliminación de vehículos.*

Figura 14

Bosquejo edición y eliminación de vehículos



7.1.6 Diseño de pantalla QR generado por el usuario para ingreso al parqueadero.

En este diseño solo evidenciamos el código QR que se genera a partir del vehículo el cual debe ser seleccionado previamente por el usuario.

Figura 15

Bosquejo QR generado por el usuario para ingreso

Parking uis -QR

QR con la información del
vehículo seleccionado

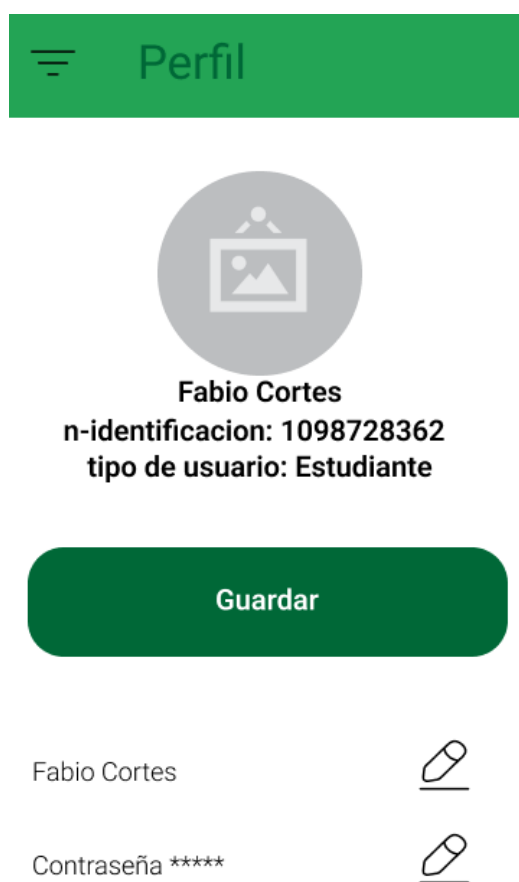


7.1.7 Diseño de edición del perfil de usuario


Este diseño nos permite evidenciar los datos del perfil del usuario, así como su rol dentro del sistema (Estudiante, Administrativo, Docente o Visitante). También la modificación del nombre de usuario y su contraseña.

Figura 16

Bosquejo de edición de perfil de usuario





Perfil



Fabio Cortes
n-identificacion: 1098728362
tipo de usuario: Estudiante

Guardar

Fabio Cortes 

Contraseña **** 

7.1.8 Diseño de la pantalla principal del moderador, que a su vez es la pantalla para iniciar el proceso de registro de sesiones de parqueo y menú.

Este diseño muestra los diferentes campos mostrados al moderador antes de permitir el ingreso a las instalaciones además del botón el cual permitirá al moderador escanear el QR generado por el usuario que desea ingresar al sistema, también un menú lateral izquierdo desplegable igualmente con el logo de la universidad, pero opciones diferente al usuario o el administrador.

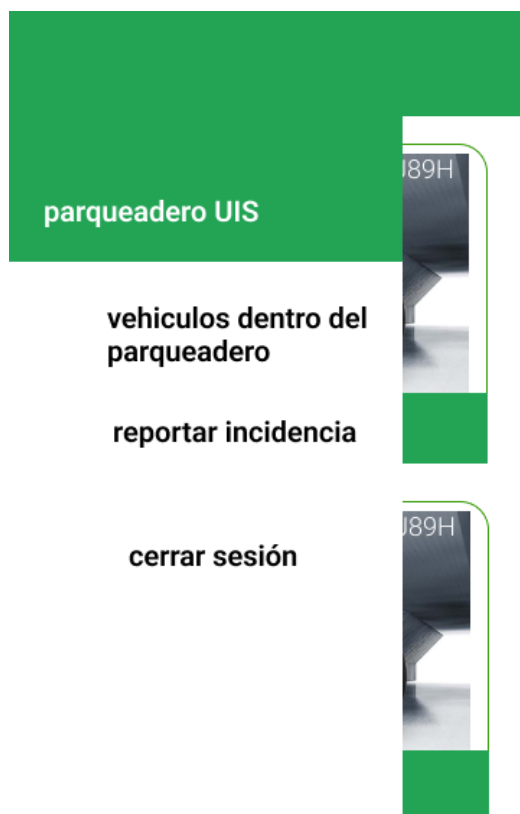
Figura 17

Bosquejo pantalla principal Moderador



Figura 18

Bosquejo de menú del moderador



7.1.9 Diseño de la pantalla de reporte de incidencias.

Este diseño deja ver los campos con la información del vehículo al cual se le iniciara un reporte de incidencias con sus respectivas observaciones, también se diseña un icono para el escaneo del código QR.

Figura 19

Bosquejo de la pantalla de reporte de incidencias

Formulario registro de incidencias

Placa

tipo de vehiculo


propietario

cedula

tipo de incidencia ∨

observaciones

enviar reporte



7.1.10 Diseño de la pantalla de vehículos dentro del sistema.

Para este apartado se tiene un diseño básico de interfaz, en el cual podemos visualizar los vehículos ingresados al sistema y su totalidad

Figura 20

Bosquejo de la pantalla de vehículos dentro del sistema



vehiculos del sistema 9



fecha de ingreso
hora de entrada
placa: abd154



fecha de ingreso
hora de entrada
placa: abd154

7.1.11 Diseño de la pantalla de detalles de vehículos dentro del sistema.

En este diseño encontramos los datos del vehículo seleccionado, nombre del propietario e identificación.

Figura 21

Bosquejo de la pantalla de detalles de vehículos dentro del sistema



7.1.12 *Diseño de la pantalla de registro de moderadores por parte del administrador.*

En este diseño encontramos el formulario necesario para la creación de un moderador

Figura 22

Bosquejo pantalla de registro de moderadores por parte del administrador

The wireframe shows a registration form for a moderator. At the top is a green header with the text "Parking uis". Below the header are five input fields, each with a green border and rounded corners. The first four fields are labeled "Nombre completo*", "Identificación*", "Correo electrónico*", and "Contraseña*", indicating they are required. The fifth field is labeled "seleccionar parqueadero asignado". At the bottom of the form is a dark green button with the text "Crear moderador".

7.1.13 Diseño de la pantalla de visualización y eliminación de usuarios del sistema.

Este diseño el cual es propio del administrador refleja el nombre, foto e identificación del usuario, así como un icono de eliminación el cual brindara la funcionalidad de eliminar dicho usuario.

Figura 23

Bosquejo pantalla de visualización y eliminación de usuarios del sistema

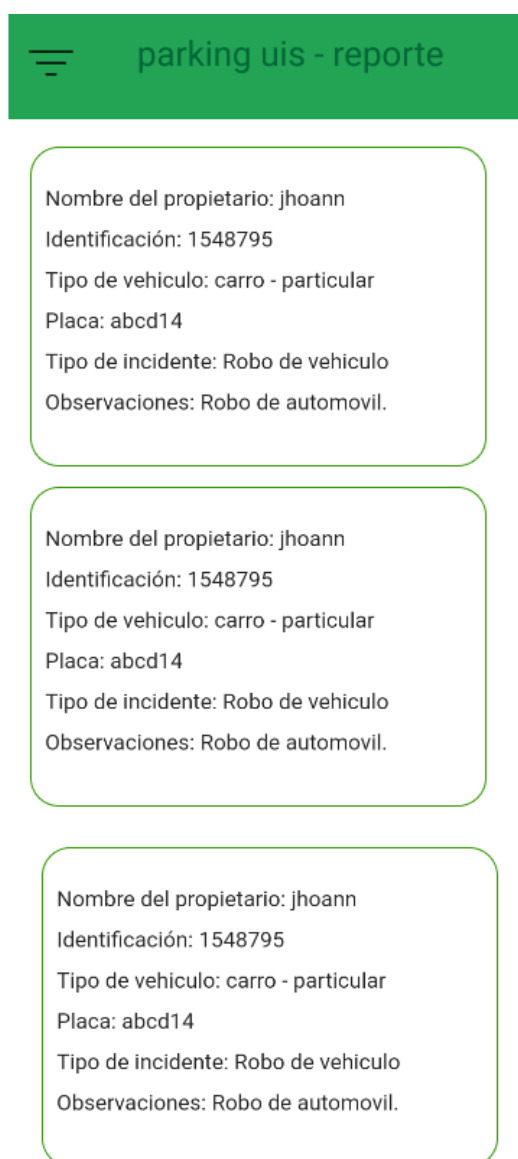


7.1.14 Diseño de la pantalla de visualización de reportes de incidencias.

En este diseño se puede visualizar todos los reportes realizados con información del usuario involucrado, el vehículo, y el motivo.

Figura 24

Bosquejo de pantalla de visualización de reportes de incidencias



8 Desarrollo

El objetivo en esta etapa es concluir la funcionalidad del aplicativo de parqueadero universitario, cumpliendo con exactitud todos los requerimientos previamente establecidos. Dicha etapa transcurre de forma iterativa, ya que se encuentra en constante proceso de evaluación y deben ser corregidos los errores con el fin de mejorar el aplicativo.

8.1 Desarrollo de Software.

Una vez definida la arquitectura y los casos de uso inicia esta fase, por tal razón fue necesario crear un flujo de trabajo por medio de un tablero virtual el cual ayudo a la gestión del proyecto, tablero conocido como TRELLO, donde se crearon los apartados de tareas pendientes, tareas asignadas, tareas en desarrollo y tareas finalizadas, del desarrollo de frontend y backend del aplicativo móvil.

En la primera parte del proyecto, se desarrollaron interfaces iniciales con el fin de generar las vistas del aplicativo móvil, esto con el fin de agilizar y optimizar el proceso de desarrollo de interfaz de usuario.

Seguido de esto se hizo la creación de un repositorio en GitHub el cual permite controlar las versiones y avances tanto en frontend como en backend, en secuencia a esto se empezó a desarrollar la base de datos relacional construida en MySQL, una vez teniendo definida la base de datos se procede con la creación del Backend (creación del REST API) el cual fue construido con node.js, encargado de proveer los servicios de los requerimientos preestablecidos.

Una vez se finaliza la etapa de creación del REST API se procede a iniciar la fase de creación de interfaz de usuario usando el framework Flutter, dicha interfaz se rige a las vistas previamente creadas, cumpliendo así con los requerimientos más importantes tales como:

8.1.1 Registro de usuarios.

El cual cumple a cabalidad con uno de los requerimientos de prioridad alta (RF01), también se evidencia un cambio y mejora de diseño en comparación a las vistas propuestas.

Figura 25

Vista Registro de usuarios

2:18

Crear Cuenta

Nombre Completo*

Identificación

Correo electrónico

Contraseña

seleccione rol dentro de la entidad ▼

crear cuenta

¿Ya tienes una cuenta?

8.1.2 Inicio de sesión.

cumpliendo en totalidad con el requerimiento (RF02) teniendo así los campos necesarios para la autenticación dentro del aplicativo.

Figura 26

Vista inicio de sesión



8.1.3 Visualización de vehículos.

También se hizo la creación del apartado principal del usuario el cual permite la visualización, registro, edición y eliminación de vehículos en el sistema por parte del usuario (requerimientos: RF06, RF07, RF08), para el caso de registro por defecto se dispuso de dar un límite a la cantidad de vehículos posibles registrados por el usuario (requerimiento: RN02), en este caso de 5 vehículos por usuario, el límite es modificable desde el backend y evidenciable al instante dentro del aplicativo.

Figura 27

Vista visualización de vehículos

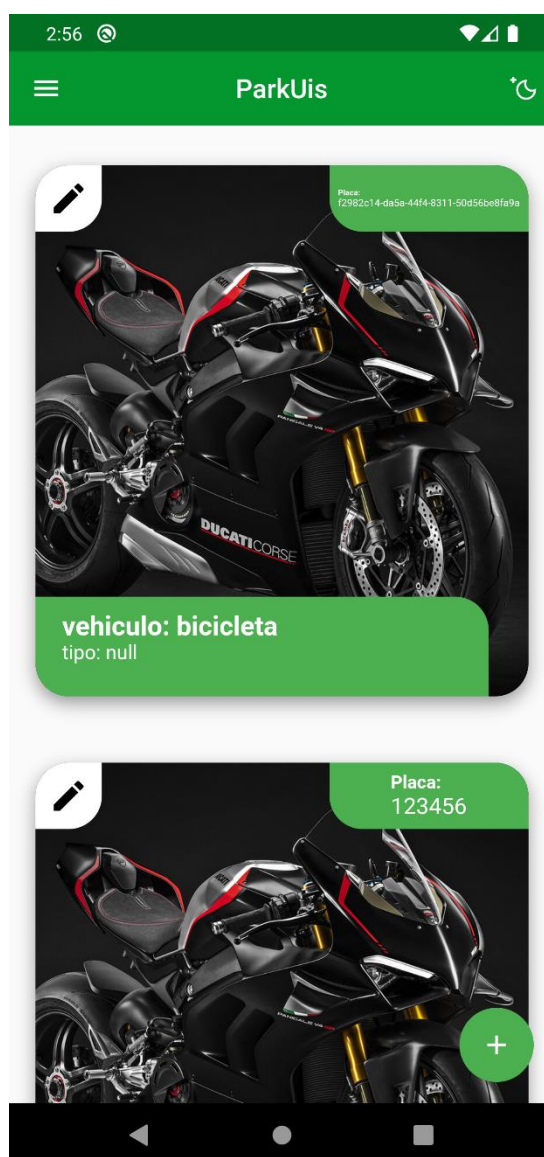


Figura 28

Registro de vehículos

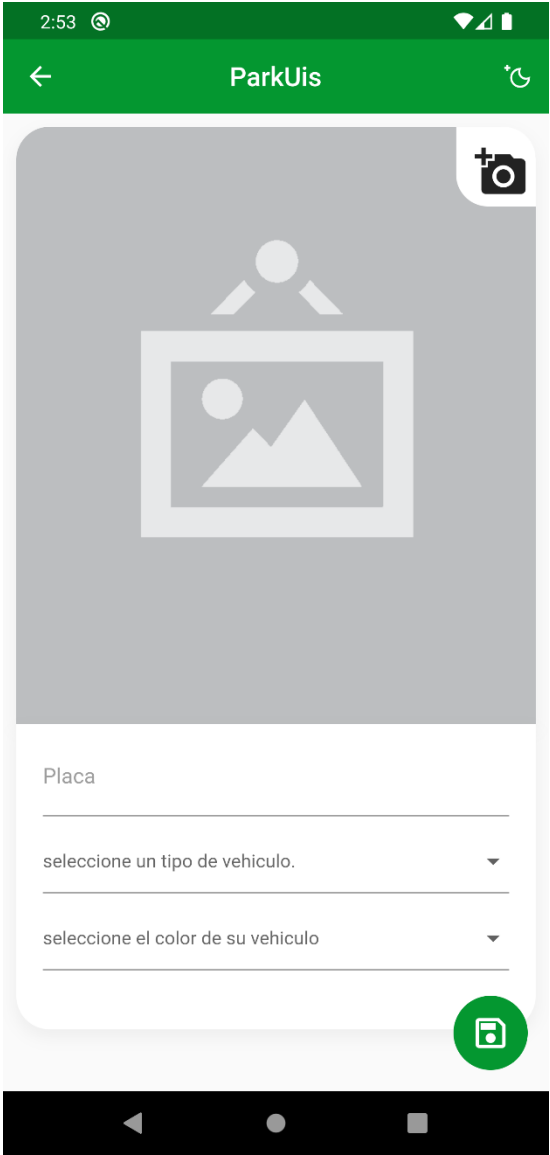
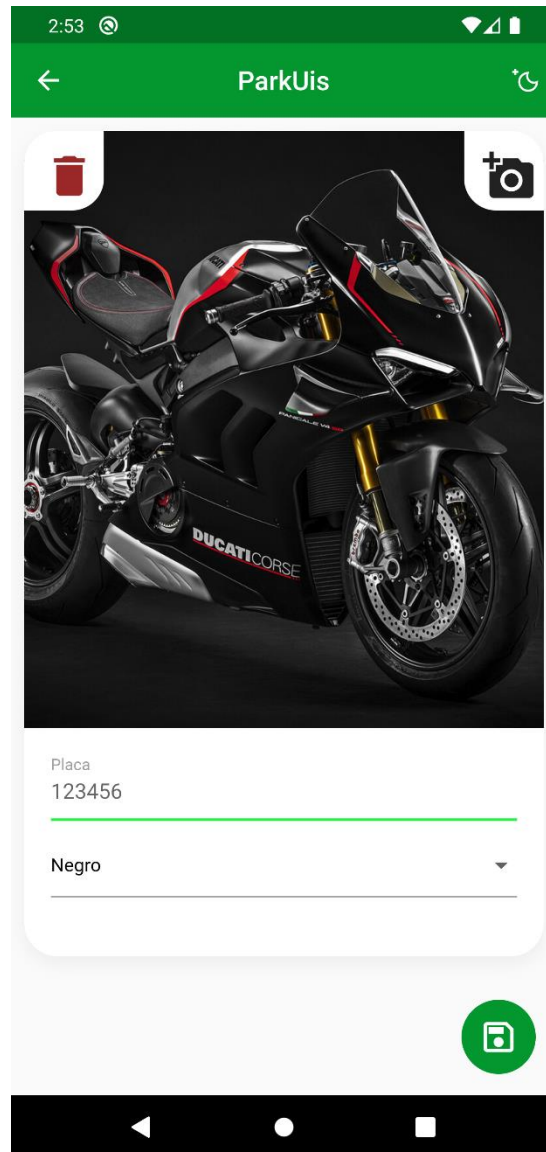


Figura 29

Edición y eliminación de vehículos



Se desarrollo también la pantalla la cual nos muestra el código QR generado por un vehículo previamente seleccionado.

Figura 30

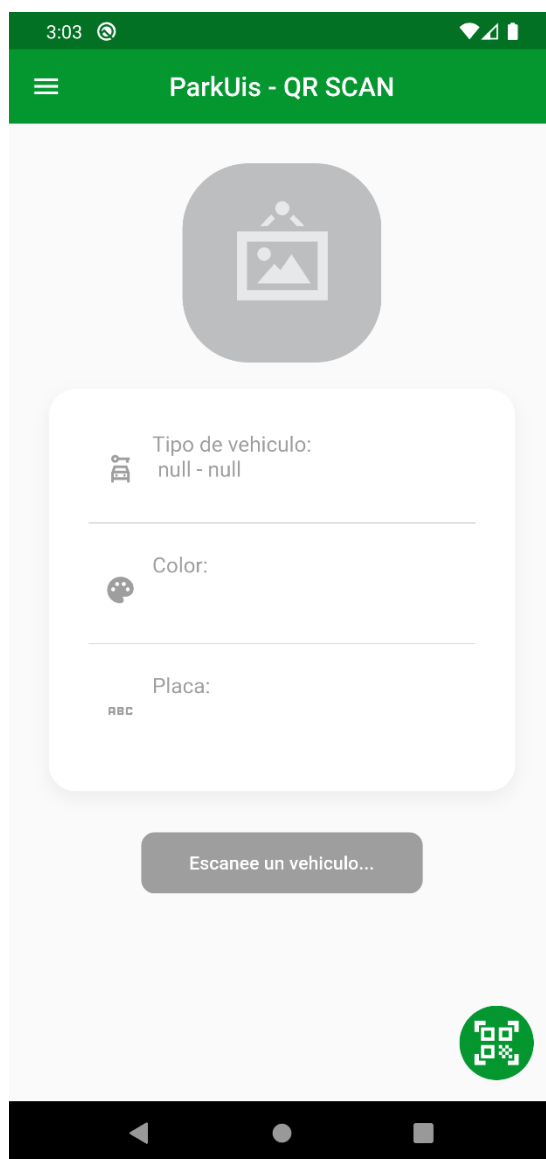
Vista QR a escanear



En el apartado de moderadores se desarrolló la interfaz de escaneo, registro, salida y visualización del vehículo al cual es escaneado su código QR cumpliendo con el requerimiento (RF12).

Figura 31

Vista Escaneo de QR



8.1.4 Visualización de sesiones de parqueo.

Se desarrollo la pantalla de visualización y eliminación de sesiones dentro del parqueadero asignado al moderador, así como la de ver detalles del vehículo dentro del sistema cumpliendo con el requerimiento (RF14).

Figura 32

Vista de visualización de sesiones de parqueo

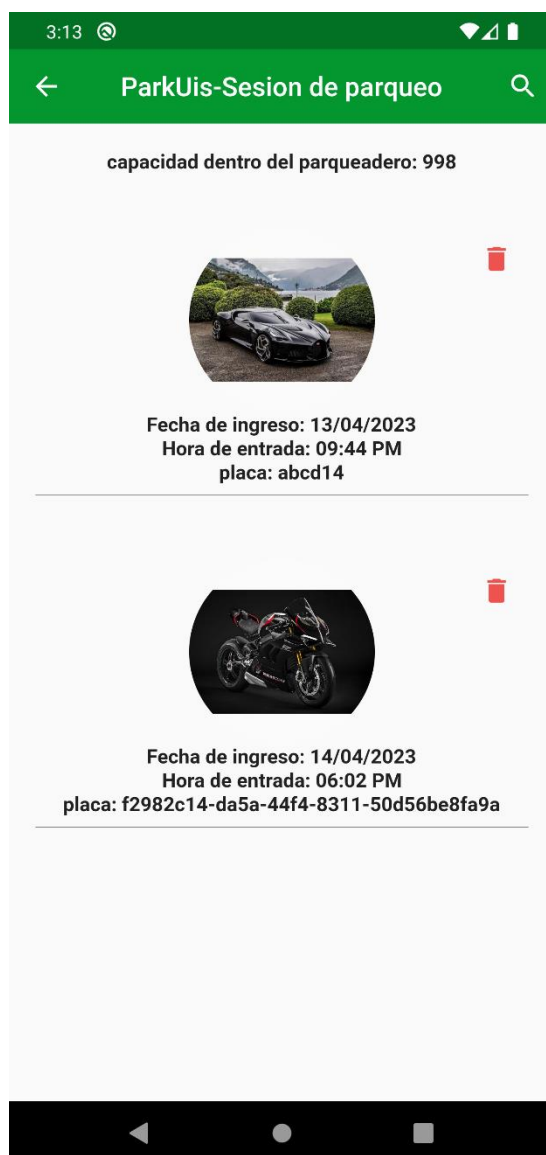
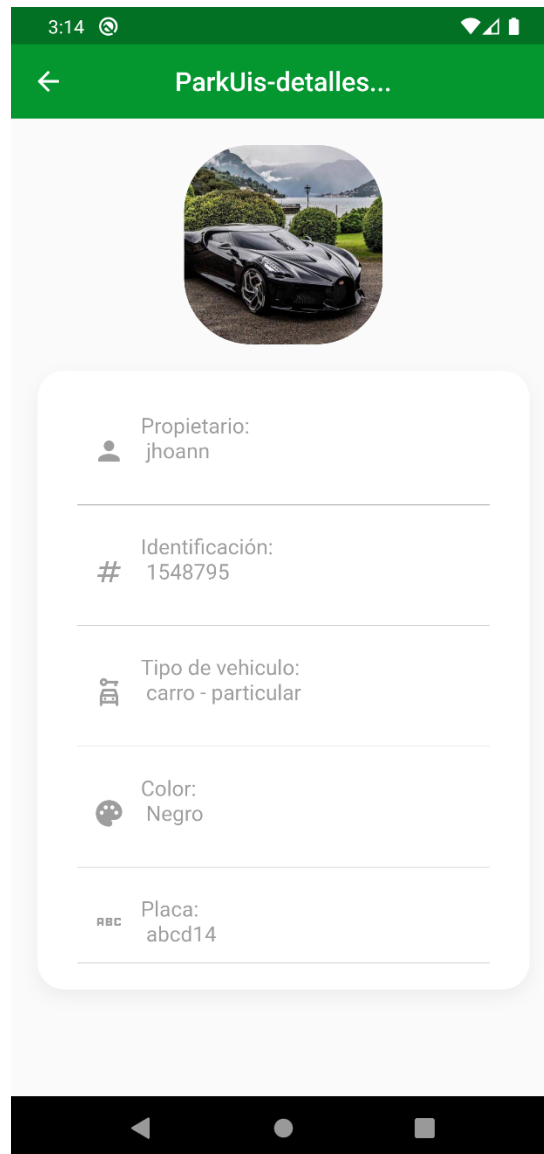


Figura 33

Detalles de la sesión de parqueo

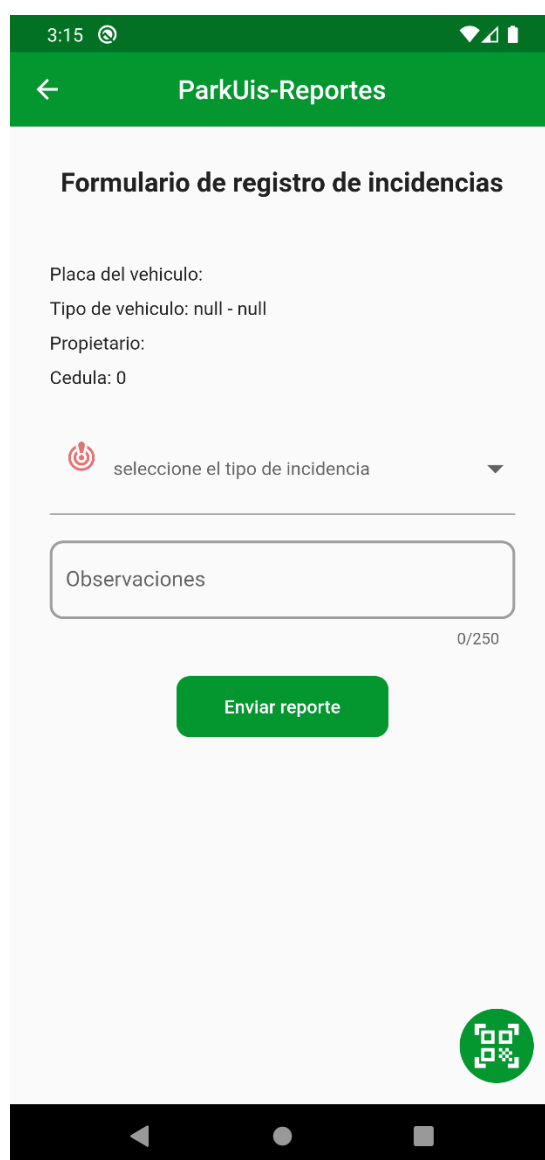


8.1.5 Registro de incidencias

Se desarrollo el apartado de registro de incidencias el cual actúa también por medio del escaneo del Código QR del vehículo al cual se le iniciara un proceso de reporte de incidencia, esta interfaz es propia del moderador.

Figura 34

Vista de registro de incidencias



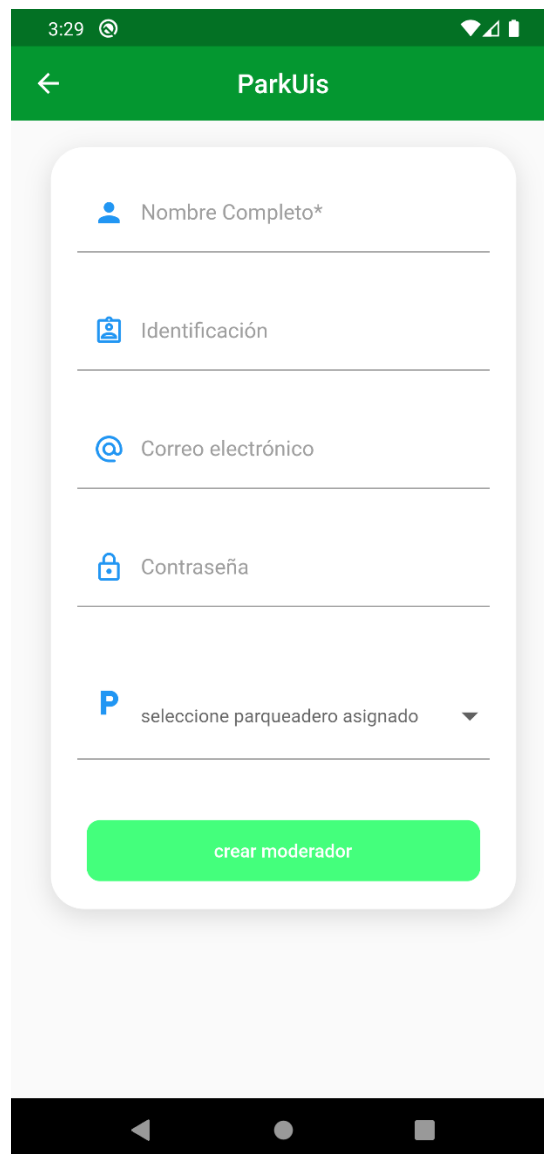
The screenshot displays the 'ParkUis-Reportes' application interface. At the top, there is a green header bar with a back arrow on the left and the text 'ParkUis-Reportes' in the center. Below the header, the title 'Formulario de registro de incidencias' is centered. The form contains several fields: 'Placa del vehiculo:', 'Tipo de vehiculo: null - null', 'Propietario:', and 'Cedula: 0'. Below these is a dropdown menu with a red power icon and the text 'seleccione el tipo de incidencia'. A text input field labeled 'Observaciones' is positioned below the dropdown, with a character count '0/250' to its right. A green button labeled 'Enviar reporte' is centered below the input field. In the bottom right corner, there is a green circular icon with a white QR code. The bottom of the screen shows the standard Android navigation bar.

8.1.6 Registro de moderador.

Finalmente, en el desarrollo correcto del aplicativo se tienen las interfaces del administrador en las cuales lo más importante a destacar sería la interfaz de creación de moderadores y visualización de la totalidad de vehículos dentro del sistema las cuales cumplen a cabalidad con los requerimientos (RF14, RF15).

Figura 35

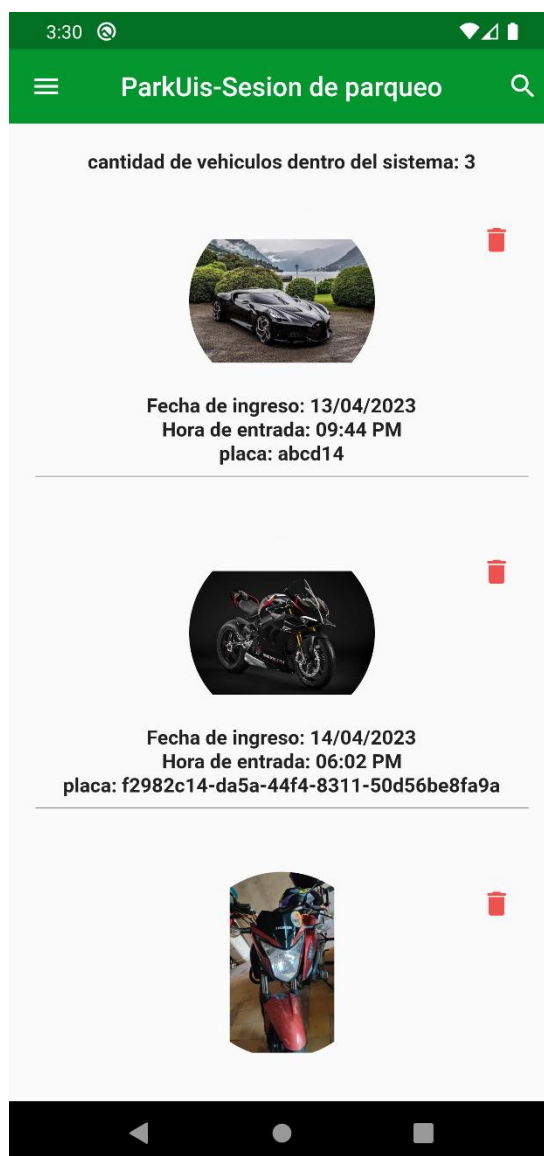
Vista de registro de moderador



The image shows a mobile application interface for creating a moderator. At the top, there is a green header bar with a back arrow on the left, the text 'ParkUis' in the center, and system icons (time 3:29, signal, Wi-Fi, battery) on the right. Below the header is a white rounded rectangle containing the registration form. The form has five input fields, each with a blue icon and a label: 'Nombre Completo*' (person icon), 'Identificación' (ID card icon), 'Correo electrónico' (email icon), 'Contraseña' (lock icon), and 'seleccione parqueadero asignado' (parking 'P' icon with a dropdown arrow). At the bottom of the form is a bright green button with the text 'crear moderador'. The bottom of the screen shows the standard Android navigation bar with back, home, and recent apps buttons.

Figura 36

Visualización de sesiones por medio del Administrador



Cabe aclarar que en cada etapa de creación de interfaz se van consumiendo a su vez los servicios del REST API según las necesidades. Finalmente, una vez se completan los diferentes módulos se evalúa la funcionalidad del prototipo y si se cumple con los requerimientos.

9 Pruebas

Para garantizar la calidad del software desarrollado en esta tesis, se llevaron a cabo pruebas de rendimiento y usabilidad. En cuanto a las pruebas de rendimiento, se evaluó la capacidad de respuesta del sistema ante un alto volumen de solicitudes simultáneas, con el fin de identificar posibles cuellos de botella y optimizar el rendimiento del software. Por otra parte, las pruebas de usabilidad se enfocaron en la facilidad de uso y comprensión del aplicativo, para lo cual se realizaron pruebas con usuarios reales con el fin de identificar posibles problemas de interfaz y experiencia de usuario y poder corregirlos. La realización de estas pruebas permitió validar el correcto funcionamiento del software y mejorar la satisfacción del usuario final.

9.1 Pruebas funcionales con carga.

Para garantizar la eficiencia y el buen rendimiento de la aplicación de parqueadero universitario, se desarrollaron pruebas funcionales con usuarios en dispositivos móviles tanto iOS como Android. Estas pruebas se realizaron utilizando una red Wi-Fi para simular un escenario real en el que el usuario tendría acceso a internet. Las pruebas se enfocaron en medir el tiempo de respuesta de la aplicación al realizar tareas como el registro de un usuario y el escaneo de códigos QR para el ingreso y salida del parqueadero. Los resultados obtenidos fueron evaluados y se implementaron ajustes en la aplicación para optimizar su manejo y garantizar una experiencia de usuario satisfactoria.

En cuanto a la funcionalidad del aplicativo, se desarrolló un sistema de escaneo de códigos QR para permitir el ingreso y salida de vehículos del parqueadero universitario. El proceso de ingreso al parqueadero consiste en escanear un código QR que se encuentra en la entrada del parqueadero, el cual contiene información relevante sobre la hora de ingreso y la ubicación del vehículo dentro del

parqueadero. De igual forma, al momento de salir, el conductor debe escanear otro código QR ubicado en la salida del parqueadero, el cual registrará la hora de salida. Se realizaron pruebas exhaustivas de usabilidad y rendimiento en dispositivos iOS y Android, y se verificó que el sistema de escaneo de códigos QR funcionaba correctamente y de manera fluida en ambos sistemas operativos, utilizando la conexión de red WiFi para transmitir la información.

9.2 Pruebas de usabilidad

Para evaluar la usabilidad del aplicativo de parqueadero universitario, se utilizó la escala SUS (System Usability Scale). Esta escala es un método estandarizado para medir la usabilidad de un sistema, la cual se compone de 10 preguntas que miden la satisfacción del usuario, la eficiencia y la facilidad de uso del sistema. La escala SUS es una herramienta ampliamente utilizada en la evaluación de la usabilidad de aplicaciones móviles y se considera una de las herramientas más fiables y precisas para medir la usabilidad. Las pruebas de usabilidad se llevaron a cabo en un grupo de personas que utilizan el parqueadero habitualmente, y los resultados obtenidos fueron analizados para determinar los puntos fuertes y débiles de la aplicación. Con base en los resultados obtenidos, se realizaron ajustes en la interfaz de usuario para mejorar la usabilidad y la experiencia del usuario en general. (**Anexo A**)

10 Conclusiones

Se ha logrado cumplir con el objetivo general de desarrollar e implementar una herramienta software de control de acceso que gestione el proceso de entrada y salida del parqueadero UIS, mediante el uso de tecnologías de información. A su vez, los objetivos específicos planteados han sido cumplidos de manera satisfactoria, ya que se identificaron los requerimientos funcionales y no funcionales necesarios para el desarrollo de la aplicación, se diseñó una arquitectura de software que se adaptó a dichos requerimientos, se implementó la aplicación y finalmente, se validó su funcionamiento mediante un conjunto de pruebas de software.

El desarrollo de esta aplicación móvil permitirá facilitar el proceso de ingreso y salida de vehículos mediante el uso de tecnologías TI, mejorando la gestión en el parqueadero universitario.

Además, el desarrollo de esta aplicación ha permitido aplicar conocimientos teóricos y prácticos en el área de desarrollo de software, así como también en el área de gestión de proyectos, al seguir una metodología adecuada y cumplir con los plazos establecidos para la entrega del proyecto.

En cuanto a la validación del prototipo, se puede concluir que el conjunto de pruebas de software permitió comprobar el correcto funcionamiento de la aplicación, identificando oportunidades de mejora y obteniendo una retroalimentación valiosa por parte de los usuarios, lo que permitió realizar ajustes en el aplicativo para mejorar su experiencia de uso.

En resumen, se puede concluir que el desarrollo e implementación de esta herramienta software de control de acceso para el parqueadero UIS ha sido un proyecto exitoso, que permitirá mejorar la experiencia de los usuarios y llevar un control y gestión de los procesos administrativos del parqueadero de la Universidad Industrial de Santander.

11 Trabajo a futuro

Para futuros trabajos en el desarrollo de la herramienta software de control de acceso del parqueadero UIS, se sugiere realizar algunas mejoras y funcionalidades adicionales que podrían mejorar significativamente la experiencia de usuario. En primer lugar, sería beneficioso conectar con los servidores y bases de datos de la universidad para validar ciertos roles, como estudiantes, profesores, administrativos y miembros del servicio de vigilancia, lo que mejoraría la seguridad y la eficacia del sistema.

Además, se recomienda implementar patrones de diseño en el desarrollo del software para que el código fuente sea más fácil de entender y mantener. También sería interesante que la universidad tuviera el presupuesto para publicar la aplicación móvil en plataformas como Play Store y AppStore, lo que facilitaría su acceso a un público más amplio.

Otra posible funcionalidad para implementar sería el uso de inteligencia artificial para validar que el tipo de vehículo y el color correspondan a la imagen cargada por el usuario. Si la imagen no corresponde, la aplicación no permitiría agregar vehículos, lo que mejoraría la precisión y la integridad de la información.

Por último, sería útil implementar una funcionalidad de "olvidé mi contraseña" que requiera autenticación de dos factores con correo electrónico y número de teléfono, lo que mejoraría la seguridad de las cuentas de usuario. En resumen, estas mejoras podrían hacer que la herramienta software de control de acceso del parqueadero UIS sea aún más eficiente y satisfactoria para los usuarios.

Referencias Bibliográficas

(S/f-b). Mysql.com. Recuperado el 15 de abril de 2023, de

<https://dev.mysql.com/doc/refman/8.0/en/what-is-mysql.html>

(S/f-c). Amazon.com. Recuperado el 15 de abril de 2023, de [https://aws.amazon.com/es/what-](https://aws.amazon.com/es/what-is/javascript/#:~:text=JavaScript%20es%20un%20lenguaje%20de,usuario%20de%20un%20sitio%20web.)

[is/javascript/#:~:text=JavaScript%20es%20un%20lenguaje%20de,usuario%20de%20un%20sitio%20web.](https://aws.amazon.com/es/what-is/javascript/#:~:text=JavaScript%20es%20un%20lenguaje%20de,usuario%20de%20un%20sitio%20web.)

(s.f.). Obtenido de <https://www.kionetworks.com/blog/data-center/los-sistemas-de-informacion-de-una-empresa>

(s.f.). Obtenido de <https://www.kionetworks.com/blog/data-center/los-sistemas-de-informacion-de-una-empresa>

(s.f.). Obtenido de <https://sites.google.com/site/educaciongepper/unidad-i-introduccion-a-el-lenguaje-c>

(s.f.). Obtenido de <https://www.redhat.com/es/topics/api/what-are-application-programming-interfaces>

(s.f.). Obtenido de <https://bibliotecadigital.univalle.edu.co/bitstream/handle/10893/10313/Fundamentos-de-bases-de-datos.pdf?sequence=6&isAllowed=y>

(s.f.). Obtenido de <https://www.pragma.com.co/academia/lecciones/bases-de-datos-relacionales-vs.-no-relacionales>

(s.f.). Obtenido de <https://aws.amazon.com/es/what-is/javascript/>

(s.f.). Obtenido de <https://www.pispos.co/node-js-entorno-de-tiempo-de-ejecucion-de-javascript>

(s.f.). Obtenido de <https://openwebinars.net/blog/que-es-postman/>

(s.f.). Obtenido de <https://openwebinars.net/blog/que-es-mysql/>

¿Qué es una api de rest? Red Hat - We make open source technologies for the enterprise. (n.d.). Retrieved June 6, 2022, from <https://www.redhat.com/es/topics/api/what-is-a-restapi>

Acerca. (s/f). Node.js. Recuperado el 15 de abril de 2023, de <https://nodejs.org/es/about>

Cuartas Castro, K. A., Cruzado Jiménez, J. C., y Ferro Escobar, R. (2017). Diseño de un modelo de ciclo parqueadero inteligente soportado en tecnología Rfid/Nfc y aplicación móvil sobre Android para la sede de ingeniería de la Universidad Distrital Francisco José de Caldas. *Redes de Ingeniería*, 184–198. <https://doi.org/10.14483/2248762X.12491>

Díaz Díaz, Suarez Barón, M. J., Blanco Santos, J. F., & Benavides Calderón, G. (2011). Sistema de control vehicular utilizando reconocimiento óptico de caracteres. *TECCIENCIA*, 6(11), 21–29.

Figuerola, F. J., & Peralta, N. C. (2017). Factores humanos en el desarrollo de software. *SEDICI*. Recuperado de <http://sedici.unlp.edu.ar/handle/10915/61343>

Gómez, C. A. (2018). Desarrollo de un sistema de control de acceso a parqueaderos para la Universidad Autónoma de Occidente (Bachelor's thesis, Universidad Autónoma de Occidente). Retrieved from <https://red.uao.edu.co/bitstream/handle/10614/9750/T07418.pdf?sequence=1&isAllowed=y>

JAVIER SÁEZ HURTADO. (2021). Las metodologías ágiles más utilizadas y sus ventajas dentro de la empresa. 2022, febrero 20, de Thinking for Innovation Recuperado de <https://www.iebschool.com/blog/que-son-metodologias-agiles-agile-scrum/>

- Lahtela, M., & Kaplan, P. (1966). ES. Amazon. Retrieved June 6, 2022, from <https://aws.amazon.com/es/microservices/#:~:text=Los%20microservicios%20son%20un%20enfoque,servicios%20son%20equipos%20peque%C3%B1os%20independientes.>
- Laudon, K. C., Laudon, J. P. (2016). *Sistemas de información gerencial*. Pearson Educación. <http://www.ebooks7-24.com.bibliotecavirtual.uis.edu.co/?il=3300> Página 10 de 12
- Lendínez, L. C. (2019). Kanban. Metodología para aumentar la eficiencia de los procesos. *C Tecnología; Alcoy*, 8(1), 30–41. <https://www.proquest.com/scholarly-journals/kanbanmetodología-para-aumentar-la-eficiencia-de/docview/2209310110/se-2>
- Millán, M. E. (2012). *Fundamentos de bases de datos*. Programa Editorial Universidad del Valle. <https://elibro-net.bibliotecavirtual.uis.edu.co/es/ereader/uis/129060?page=9>
- Oviedo Bayas, & Torres Quijije, Ángel. (2020). Uso de las técnicas de la información y comunicación y su incidencia en el control de acceso. *Journal of business and entrepreneurial studies: JBES*, 4(1).
- Rendón, Y. A. (n.d.). *Bases de Datos Relacionales vs. no relacionales*. Prehome. Retrieved June 6, 2022, from <https://www.pragma.com.co/academia/lecciones/bases-de-datosrelacionales-vs.-norelacionales#:~:text=Las%20bases%20de%20datos%20relacionales%20se%20basan%20en%20la%20organizaci%C3%B3n,o%20m%C3%A1s%20conjuntos%20de%20datos.>
- Solarte-Martínez, Silva Castro, F., & Muñoz-Guerrero, L. E. (2020). Análisis, diseño y desarrollo de un prototipo de software para la administración de parqueaderos. *Ingeniería y Competividad*, 22(1), 1–13. <https://doi.org/10.25100/iyc.v22i1.8752>

Torres Mendoza, J. D. (2017). Aplicación Móvil para la Localización de un Vehículo en el Parqueadero.

Recuperado de: <http://hdl.handle.net/11349/6736>.

UI From Mars. (2021). Cómo medir usabilidad: ¿qué es SUS? [Blog post]. Recuperado de

<https://www.uifrommars.com/como-medir-usabilidad-que-es->

[sus/#:~:text=%C2%BFQu%C3%A9%20es%20SUS%3F,la%20usabilidad%20de%20cualquier%20sistema](https://www.uifrommars.com/como-medir-usabilidad-que-es-sus/#:~:text=%C2%BFQu%C3%A9%20es%20SUS%3F,la%20usabilidad%20de%20cualquier%20sistema).

Vargas Quintero, L, Gaona Rodríguez, J y Gutiérrez Barreto, G. (2016). Prototipo de aplicación móvil para la ubicación de parqueaderos mediante GPS. Fundación Universitaria Panamericana.

What is postman? (s/f). Postman API Platform. Recuperado el 15 de abril de 2023, de

<https://www.postman.com/product/what-is-postman/>

Apéndices

Anexo A

Análisis

El método de escala de sistema de usabilidad tiene como objetivo brindar información sobre la medición de usabilidad en la aplicación móvil del sistema de parqueadero. Este método se aplicó a 10 celadores de la universidad industrial de Santander y 40 usuarios con diferentes roles dentro de las instalaciones, siendo un total de 50 usuarios encuestados.

Esta encuesta se realizó de manera presencial, como primera medida el usuario usaba la aplicación móvil, seguido realiza el cuestionario SUS, el cual contó con 10 preguntas. Cada pregunta contada con cinco opciones de respuesta, las cuales oscilaban entre “En completo desacuerdo” a “Completamente de acuerdo”.

A continuación, se muestra el cuestionario realizado y un análisis de los resultados obtenidos:

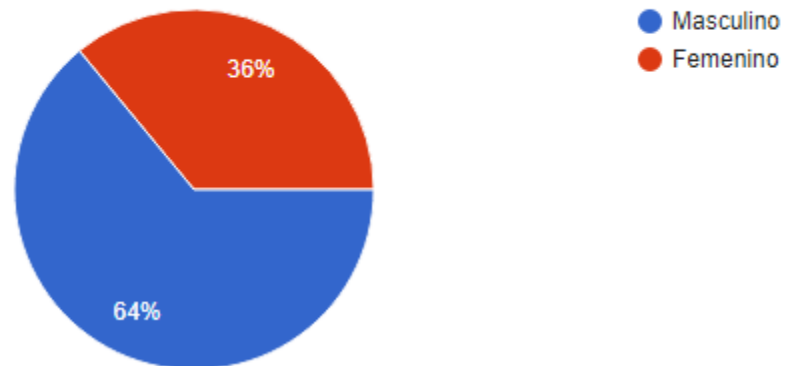
Encuesta

La presente encuesta será resuelta sólo por aquellas personas que hicieron uso de la aplicación móvil Parkuis. Cabe mencionar que los resultados de esta encuesta se almacenaron de carácter anónimo.

Preguntas de carácter demográfico

¿Cuál es su género?

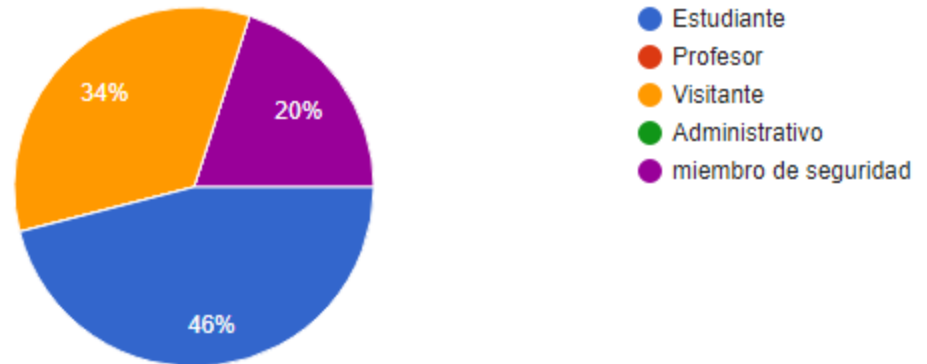
50 respuestas



Se puede observar que entre los encuestados el 36% pertenece al género femenino, y por otro lado el 64% pertenece al género masculino.

¿Cuál es su rol dentro de la universidad?

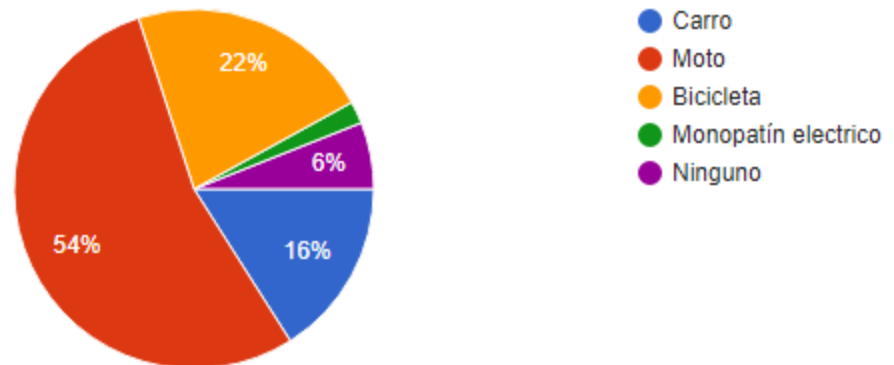
50 respuestas



Dentro de los 50 encuestados, el 46% aseguró tener un rol de Estudiante, un 20% el rol de miembro de seguridad y el 34% restante aseguró ocupar el rol de visitante.

Seleccione los vehículos que posee:

50 respuestas

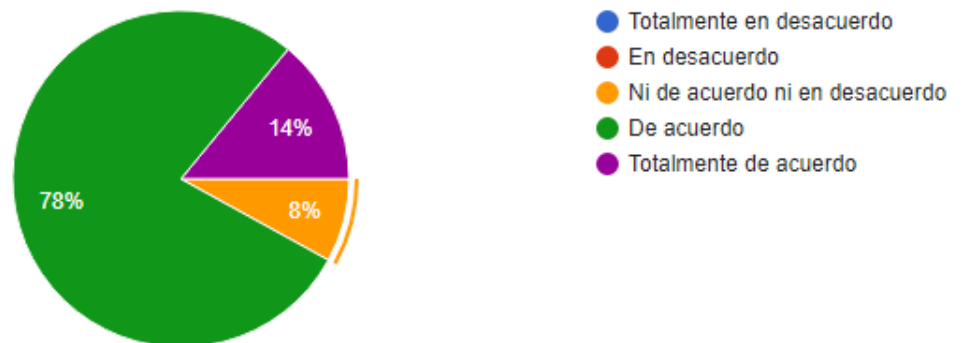


Se puede observar que un 54% posee motocicletas, un 22% tiene bicicletas y un 16% vehículos de 4 ruedas, un porcentaje muy mínimo posee monopatines eléctricos.

Evaluación de la aplicación:

1. Creo que me gustaría visitar la aplicación con frecuencia

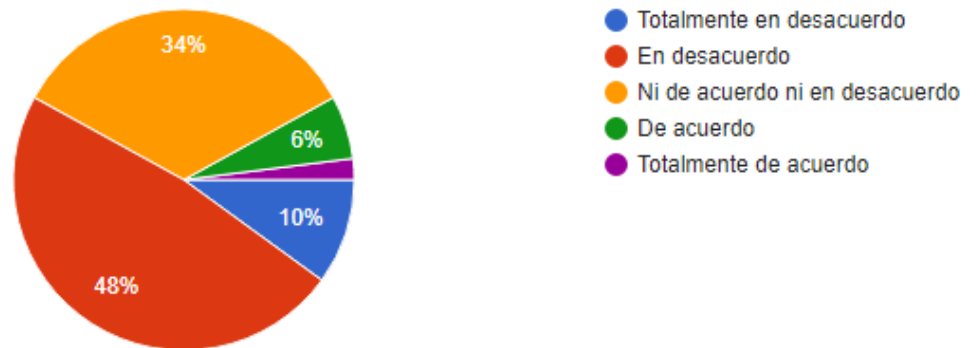
50 respuestas



La gráfica que hace referencia a la afirmación: “Creo que me gustaría visitar la aplicación con frecuencia” demuestra que la mayoría de los usuarios escogió la opción De acuerdo y totalmente de acuerdo, esta mayoría equivale al 92% de los encuestados.

2. Encontré el sistema innecesariamente complejo

50 respuestas

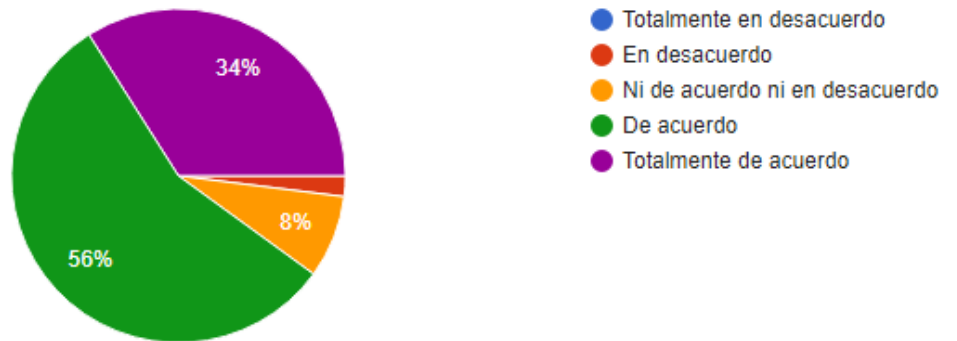


El 58% de los encuestados mostraron que están en desacuerdo con encontrar la aplicación innecesariamente compleja.

3. Pensé que el sistema era fácil de usar



50 respuestas

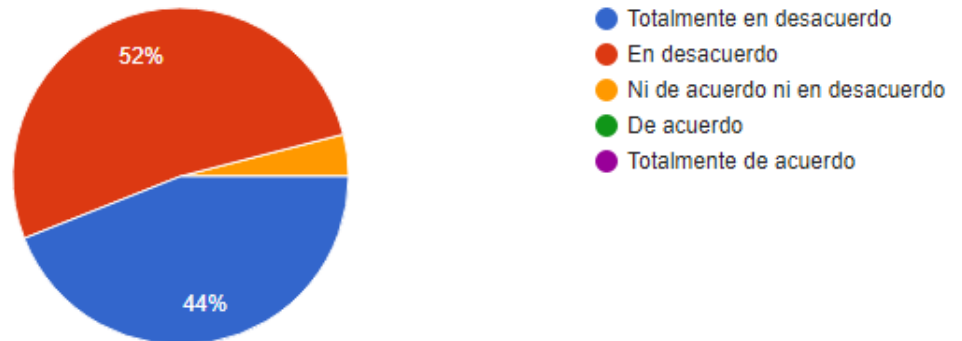


El 34% de los encuestados están en completamente de acuerdo con que la aplicación es fácil de utilizar, el 56% de los encuestados afirman que están parcialmente de acuerdo.

4. Creo que necesitaría el apoyo de un técnico para poder utilizar este sistema



50 respuestas

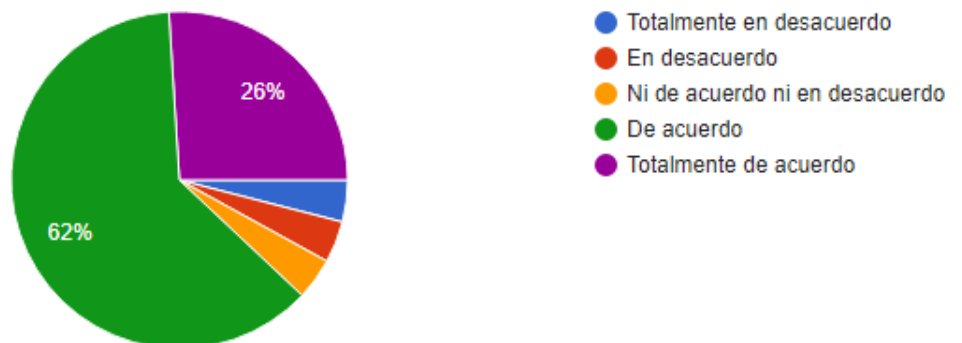


La mayoría de los encuestados afirman que están en completo desacuerdo con necesitar apoyo de un experto para realizar actividades en la aplicación, este porcentaje equivale al 96%, y el porcentaje restante está indeciso.

5. Encontré que las diversas funciones de este sistema estaban bien integradas



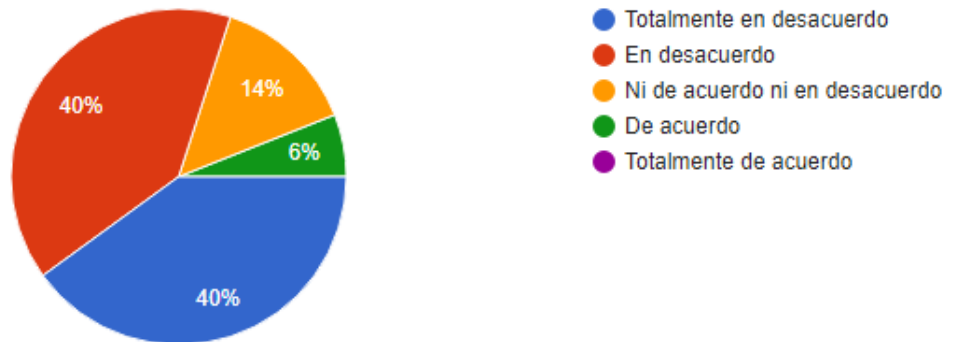
50 respuestas



Los encuestados que están completamente de acuerdo con que las diferentes funciones de la aplicación están bien integradas equivalen al 88%, y el 12% indican que están en desacuerdo e indecisos.

6. Pensé que había demasiada inconsistencia en este sistema

50 respuestas

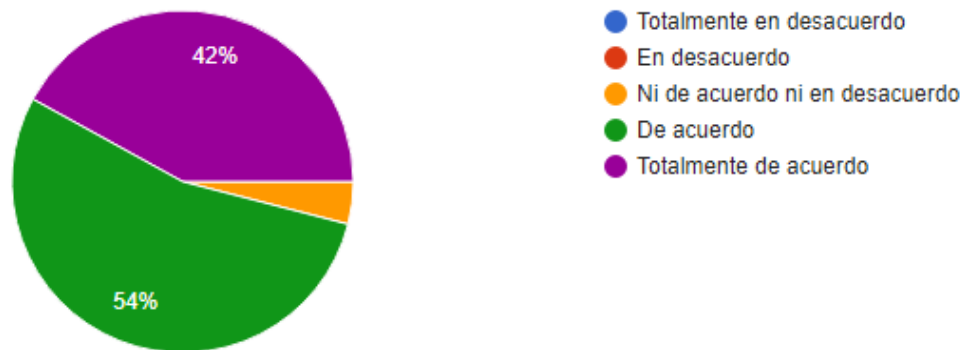


Respecto a la anterior gráfica que representa si la aplicación posee demasiada inconsistencia los encuestados respondieron que están en completo desacuerdo y parcialmente en desacuerdo. Quienes respondieron que están en completo desacuerdo equivalen al 40% y los que están parcialmente en desacuerdo a un 40%, el 20% restante está indeciso.

7. Me imagino que la mayoría de la gente aprendería a utilizar este sistema muy rápidamente



50 respuestas

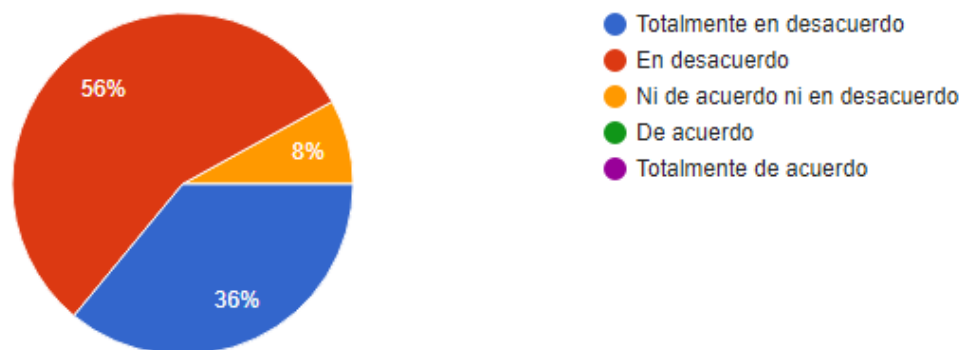


La anterior gráfica demuestra que gran número de los encuestados consideran que la mayoría de las personas aprenderían rápidamente a utilizar la aplicación, este gran número de encuestados es igual al 96%, el porcentaje restante indica que está indeciso.

8. Encontré el sistema muy complicado de usar



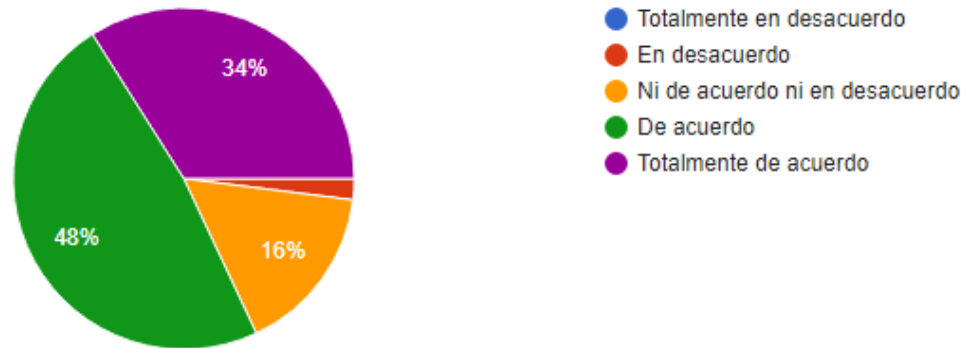
50 respuestas



Referente a la afirmación de que la aplicación es muy grande recorrerla el 36% indica que está en completo desacuerdo, mientras que el 56% está parcialmente en desacuerdo, y el 8% considera que es aceptable.

9. Me sentí muy seguro usando el sistema

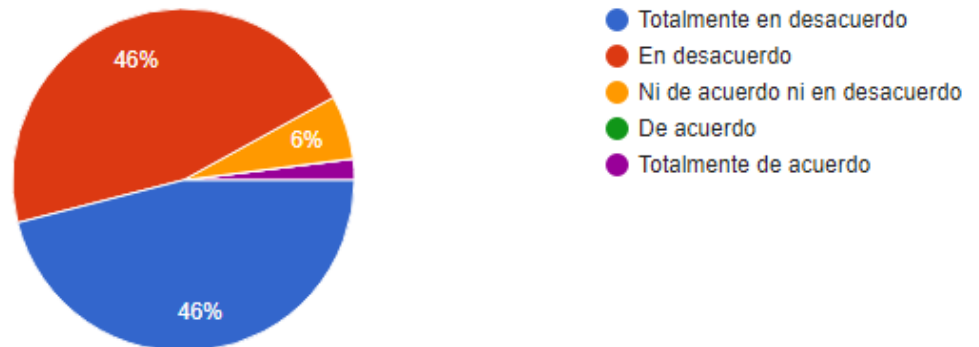
50 respuestas



El 36% de los encuestados se mostraron completamente de acuerdo con la afirmación de sentirse confiado al usar la aplicación web, mientras el otro 48% se encuentra parcialmente de acuerdo y un 16% está indeciso.

10. Necesitaba aprender muchas cosas antes de empezar con este sistema

50 respuestas



Finalmente, se encontró que la mayoría de los encuestados afirmaron que estaban en completo desacuerdo a la afirmación de necesitar aprender muchas cosas antes de manejar la aplicación. La mayoría equivale al 92% de los encuestados, el resto están parcialmente de acuerdo.

Interpretación de Resultados

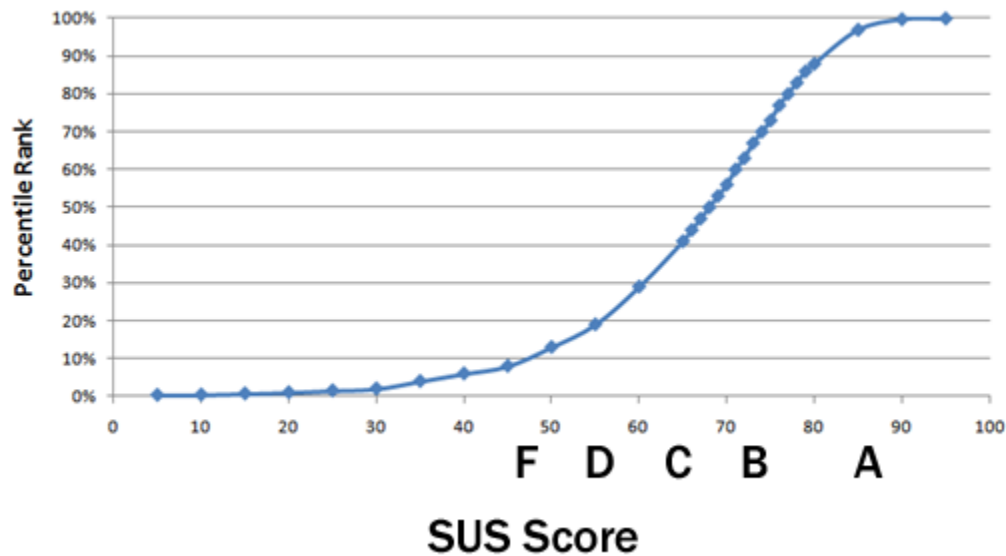
El método SUS permitió conocer de manera rápida la usabilidad que presenta la aplicación móvil Parkuis, ya que las respuestas de cada usuario se convierten en un nuevo número, puesto que en SUS se encuentran dos tipos de preguntas:

Las preguntas positivas corresponden a las preguntas 1,3,5,7 y 9; estas presentan un procedimiento para obtener el resultado en SUS, el cual es: resultado - 1.

Las preguntas negativas corresponden a las preguntas 2,4,6,8,10 e igualmente presentan un procedimiento para obtener el resultado en SUS siendo: 5 -resultado.

Seguido del procedimiento mencionado anteriormente se suma los resultados de cada pregunta, después ese resultado debe ser multiplicado por 2,5; cabe mencionar que debe ser por cada encuesta realizada. Dichos resultados pueden oscilar entre 0 y 100. Según la escala de SUS 68 es el promedio, y si un resultado se encuentra por encima del promedio se considera que la aplicación es usable, si se encuentra por debajo se considera poco usable,

A continuación, se muestra una gráfica que representa el rango percentil dependiendo del puntaje obtenido en la prueba:



Fuente: MeasuringU. (s. f.). SUS: System Usability Scale. Recuperado de <https://measuringu.com/sus/>

Los resultados obtenidos en la prueba SUS con los 50 usuarios oscilaron entre 77,5 y 100, esto permite concluir que la aplicación presenta alta usabilidad y el público objetivo puede usarla sin ningún problema.

