

DESARROLLO DE UN SISTEMA DE CONTROL DE ESTACIONAMIENTO PARA DOCENTES DE UNA UNIVERSIDAD PRIVADA EN EL AÑO 2024

Renato Mitsuhiro Hirakawa Eiwa¹

Universidad Autónoma de Encarnación - Paraguay

Recepción: 09/11/2024

Aprobación: 20/11/2025

Resumen

Una universidad privada de Encarnación se enfrenta a dificultades en la gestión de los espacios asignados para el estacionamiento de docentes. Esta situación genera problemas, como la complicación para realizar seguimientos, la falta de información actualizada sobre la disponibilidad de los espacios y una dependencia excesiva de un funcionario para su administración. En base a estos problemas se planteó como objeto de estudio desarrollar un sistema de control de estacionamiento para docentes. Este sistema está compuesto por dos módulos principales: una aplicación web y un conjunto de componentes de hardware, con el propósito de automatizar el control de entrada y salida de vehículos, mantener información actualizada sobre la disponibilidad de plazas y facilitar la administración del estacionamiento, el abordaje de la investigación es de tipología aplicada con diseño de investigación experimental, con la utilización de la metodología de desarrollo ágil Scrum, representando las actividades a desarrollar en la herramienta colaborativa Trello, mientras que las herramientas utilizadas para el desarrollo del sistema incluyen Laravel, un framework de PHP, que se complementa con la integración de Arduino Uno y diversos componentes electrónicos. Además, se emplea Python para gestionar la comunicación serial entre los dispositivos. Los resultados obtenidos han dado lugar a una aplicación web que permite gestionar usuarios y estacionamientos, verificar los registros de entrada y salida, a su vez verificar la disponibilidad de espacios en el estacionamiento. Por otro lado, el sistema hardware, que

¹ Técnico Superior en Programador de Computadoras. Universidad Autónoma de Encarnación.
mitsueiwa333@gmail.com

incluye tecnología RFID que regula el flujo de vehículos, proporcionando seguridad adicional y optimizando el uso del estacionamiento.

Palabras claves: Aplicación informática - Vehículo automotor - Tecnología electrónica - Programación informática.

Abstract

A private university in Encarnación is facing difficulties in managing the spaces assigned for faculty parking. This situation generates problems, such as complications in tracking, the lack of updated information on the availability of spaces, and an excessive dependence on a single staff member for its administration. Based on these issues, the study's objective was to develop a faculty parking control system. This system is composed of two main modules: a web application and a set of hardware components, with the purpose of automating the control of vehicle entry and exit, maintaining updated information on the availability of spaces, and facilitating parking administration. The research approach is of an applied typology with an experimental research design, utilizing the Scrum agile development methodology, representing the activities to be developed in the Trello collaborative tool, while the tools used for system development include Laravel, a PHP framework, which is complemented by the integration of Arduino Uno and various electronic components. Furthermore, Python is employed to manage serial communication between the devices. The results obtained have led to a web application that allows for user and parking management, verification of entry and exit records, and verification of parking space availability. On the other hand, the hardware system, which includes RFID technology, regulates the flow of vehicles, providing additional security and optimizing parking usage.

Keywords: Computer application - Motor vehicle - Electronic technology - Computer programming.

1. Introducción

El automóvil se ha vuelto indispensable para la sociedad actual, demostrando con el pasar del tiempo un crecimiento acelerado del parque automotor. En muchas ciudades encontrar lugar para estacionar el vehículo es un problema que afecta a la movilidad, el tránsito, la

contaminación, el consumo de combustible, espacio, pérdida de tiempo y dinero (Rodríguez et al., 2019). Al mismo tiempo, la demanda del tráfico vehicular se ha incrementado en los últimos años y esto ha ocasionado retrasos, accidentes y posibles cambios en el humor de las personas que cada día tienen que lidiar con estas situaciones cuando se desplazan de un sitio a otro (Arsenio et al., 2020).

Estos conflictos han llevado a organizaciones de todo el mundo a buscar soluciones basadas en las necesidades propias de cada ciudad para tratar de prevenir futuros problemas en estas áreas (González & Olmedo, 2019). A todo esto, la tecnología puede ayudar a sobrellevar los problemas de estacionamiento de vehículos, reduciendo el tiempo de búsqueda activa de espacios para estacionar, beneficiando a otros conductores, peatones, reduciendo la emisión de gases contaminantes ocasionados por los automóviles (Formoso et al., 2014).

Según la revista Última Hora (2022) la situación en la vía pública de grandes ciudades de Paraguay como Asunción y Encarnación se torna caótica debido al déficit existente de disponibilidad de espacios de estacionamiento, en comparación con la inmensa cantidad de vehículos que circulan diariamente en las ciudades como consecuencia vehículos en doble fila que atascan el tráfico, cocheras y rampas para personas con discapacidad o franjas peatonales obstruidas, automóviles y motocicletas estacionados sobre las aceras, lo cual atenta contra la seguridad de los peatones, son las problemáticas constantes desde hace años que se registran, por este motivo los ciudadanos del país manifiestan que es imposible encontrar un lugar para estacionar y que los costos de los estacionamientos privados son muy elevados. En este contexto para dicho problema, las universidades de Paraguay en general cuentan con un estacionamiento reservado para docentes. No obstante, como relata Chávez & Yance (2020) en una de las facultades con los que cuenta la Universidad de Guayaquil en Ecuador, los docentes mencionan que encontrar un lugar donde estacionar es un problema frecuente, lo que causa un atasco regular en horas pico, generando trabas en la circulación. Por otra parte, en algunos campus universitarios de Estados Unidos utilizan sistemas en donde indican cuántos espacios están disponibles a una determinada hora y mensajes dinámicos que advierten a los conductores los niveles de congestión dentro y en los alrededores del estacionamiento (Franco, 2014).

El servicio de atención en los estacionamientos es mínimo, ya que en la mayoría funcionan bajo un control manual, es decir, una persona controla el ingreso y egreso de vehículos, lo que crea una dependencia del personal para mantener un control efectivo de los vehículos (Rios, 2011). En ese sentido, la universidad cuenta con tres lotes de estacionamientos, uno propio en donde el control de acceso está a cargo de un funcionario y dos alquilados que no cuentan con ningún tipo de control.

La gestión actual de los espacios asignados en el estacionamiento para docentes presenta diversas dificultades, como la complejidad del seguimiento de los espacios, la falta de información actualizada sobre su disponibilidad y la excesiva dependencia de un funcionario para su administración manual. En vista a estos problemas, se planteó como objetivo desarrollar un sistema de control de estacionamiento inteligente dirigido a docentes, utilizando la metodología de desarrollo ágil Scrum como marco de trabajo para el desarrollo del software, a su vez la metodología de investigación utilizada fue la investigación aplicada, al tratarse del análisis y diseño para el desarrollo de un software y el diseño de investigación se enmarca como experimental por las características de los datos, su recopilación y requerimientos técnicos. Para comprender mejor las necesidades del sistema, se realizó una entrevista al vicerrector y director de Administración y Finanzas.

Las herramientas utilizadas para el desarrollo de la aplicación web fueron el framework Laravel basada en el lenguaje de programación PHP, por otra parte, se utilizó el lenguaje de programación Python para el sistema de control, que cumple la función de la comunicación serial con la placa Arduino, la cual está conectada vía USB por el puerto serial, como complemento cuenta con la pantalla LCD I2C para indicar los mensajes, un servo motor como barrera de paso y la tecnología RFID (Identificación por Radio Frecuencia), con el fin de eliminar los estacionamientos indebidos la cual ocasiona la obstrucción impidiendo la circulación de los vehículos en el estacionamiento, en base a eso se busca controlar la hora de entrada y salida de los vehículos. Además, la posibilidad de verificar fácilmente si hay espacios disponibles antes de llegar al estacionamiento.

2. Metodología

El presente trabajo se desarrolla bajo la tipología de investigación aplicada, al tratarse del análisis y diseño para el desarrollo de un software, sobre una problemática actual sobre la cual se da una solución mediante la incorporación de herramientas tecnológicas, como diseño de investigación se enmarca como experimental por las características de los datos, su recopilación y requerimientos técnicos.

En cuanto a la metodología de desarrollo de software, se ha seleccionado Scrum como enfoque principal para el desarrollo del proyecto. Aunque Scrum se diseñó como una metodología ágil para trabajos colaborativos, en este caso, se adaptó para su aplicación individual. La elección de Scrum se basó por su enfoque flexible en la que nos permite adaptarnos fácilmente a los cambios y necesidades del entorno. Además, la estructura iterativa y cíclica de Scrum, basada en Sprint, nos permite dividir el proyecto en tareas manejables, establecer prioridades claras y realizar un seguimiento efectivo del progreso.

La población de este estudio está constituida por docentes de una universidad privada de la ciudad de Encarnación. La elección de los docentes se debe a varios criterios: su relevancia como grupo objetivo, ya que son los principales usuarios del sistema de control de estacionamiento y el impacto directo que el sistema tiene sobre ellos. En cuanto al procedimiento de recogida de información, se efectuó mediante una entrevista con el vicerrector y director de administración y finanzas, con el fin de comprender el funcionamiento actual de los estacionamientos como también, para delinear el diseño y facilitar la implementación del sistema en desarrollo.

2.1. Entorno de desarrollo utilizado

Para el desarrollo del código, se utilizó Visual Studio Code como editor de código, por su comodidad y experiencias previas, con respecto a los requisitos técnicos y de desarrollo de la aplicación, ésta deberá ser una aplicación web. Para ello, se utilizó el framework Laravel basado en el lenguaje de programación PHP, debido a una sinergia entre mi experiencia y conocimientos previos, las características y capacidades del framework y su alineación con los objetivos y requisitos del proyecto. Laravel se destaca por ser un

framework moderno, robusto, fácil de usar y escalable. Ofrece una amplia gama de funciones y herramientas que impulsan la productividad y la seguridad, convirtiéndose en una excelente opción para el desarrollo de aplicaciones web modernas y de alto rendimiento.

El sistema se complementa con la integración de Arduino uno, cuyo código fue escrito en Arduino IDE, esta proporciona una interfaz gráfica amigable y funciones específicas para trabajar con la placa Arduino facilitando así el proceso de desarrollo. El Arduino se conecta a la computadora mediante el puerto serial USB, y controla un micro servomotor que simula la función de una barrera de entrada y salida de vehículos, Se incluye también una pantalla LCD 16x2 en la cual se muestran los mensajes al usuario y finalmente la incorporación del kit de tecnología RFID, en la que nos permite la comunicación entre un lector y una etiqueta o tag, donde nos ayudará en el proceso de control de acceso y egreso de vehículos, para llevar a cabo esta comunicación se realizó a través del puerto serial con un programa backend hecho en Python con la biblioteca PySerial. La elección de esta tecnología se basó en su simplicidad, versatilidad, robustez, amplia gama de herramientas y su facilidad de integración con el entorno Arduino. Esta combinación me permitió establecer una comunicación confiable y eficiente entre el software y el hardware, sentando las bases para el éxito del proyecto.

En lo que respecta al almacenamiento de datos, se optó por PostgreSQL como motor de base de datos, gracias a su sólida capacidad de gestión de datos, su reconocido rendimiento, su alta escalabilidad, su amplia compatibilidad con diversos lenguajes de programación y su robusto conjunto de funciones de seguridad.

2.2. Delimitación y alcance del sistema

La aplicación web está diseñada con una estructura de roles diferenciada para optimizar la gestión y el acceso. Un Super Administrador único tiene el control absoluto de la plataforma, incluyendo la administración de cuentas y el acceso exclusivo al registro de auditoría. Los Administradores comparten muchos de los permisos del Super Administrador, excepto la visibilidad del registro de auditoría. Por último, los Usuarios

comunes o docentes, disponen únicamente del acceso al apartado de inicio, donde pueden verificar la disponibilidad del estacionamiento.

- Delimitación
 - El componente de hardware del sistema de control de entrada y salida está exclusivamente diseñado para operar en el estacionamiento propio de la universidad, este no se aplica ni se extiende a estacionamientos externos asociados.
 - El Administrador ni el usuario no pueden revisar los registros de auditoría de la aplicación web.
 - La aplicación web está limitada a un borrado lógico de los usuarios, lo que significa que tanto el Super Administrador o Administrador solo puede desactivarlas, pero no eliminarlas permanentemente. Esta medida asegura la preservación de los registros y la posibilidad de reactivar las cuentas si fuera necesario.
- Alcance
 - La interfaz de usuario tendrá un diseño responsivo que permitirá su adaptación a cualquier dispositivo.
 - El usuario o el docente tienen la capacidad de consultar en tiempo real la disponibilidad de espacios en el estacionamiento.
 - El sistema de control de entrada y salida registra el usuario, la fecha y la hora en que cada docente ingresa al estacionamiento, utilizando su tarjeta RFID.
 - La aplicación web proporciona un registro de auditoría detallado para el seguimiento de las modificaciones realizadas en los campos críticos del sistema.

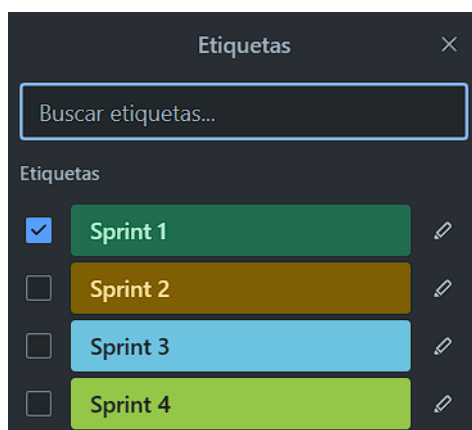
3. Análisis de los resultados

En este apartado se detalla los resultados obtenidos en el proceso del desarrollo del sistema de control de estacionamientos para docentes.

3.1 Configuración de la metodología de desarrollo de software

Se utilizó la metodología Ágil Scrum, mediante la herramienta de gestión de proyecto Trello, por medio de este se creó el product backlog donde se detallan las tareas a realizar durante el proceso del proyecto.

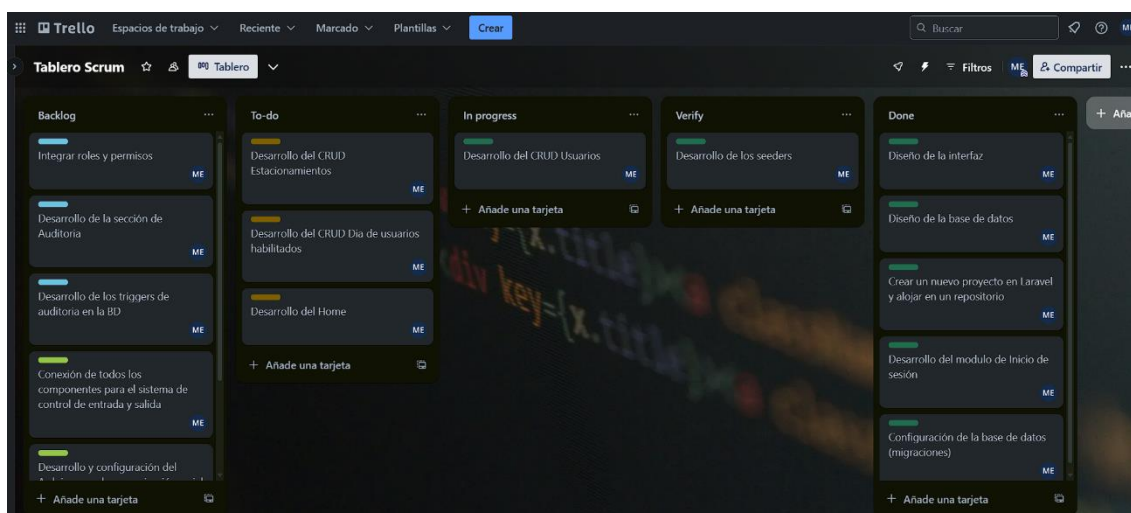
Ilustración 1. Etiquetas de identificación de Sprint.



Fuente: Elaboración propia.

Como se observa en la imagen anterior cada etiqueta corresponde a un sprint individual, los cuales fueron planificados y ejecutados siguiendo un ciclo de trabajo de dos semanas de duración para garantizar una entrega continua y eficiente de valor.

Ilustración 2. Tablero Scrum.



Fuente: Elaboración propia.

En la Ilustración 2, se muestra el tablero Scrum que se utilizó durante el desarrollo del software. Este tablero fue una herramienta fundamental para la gestión y seguimiento del

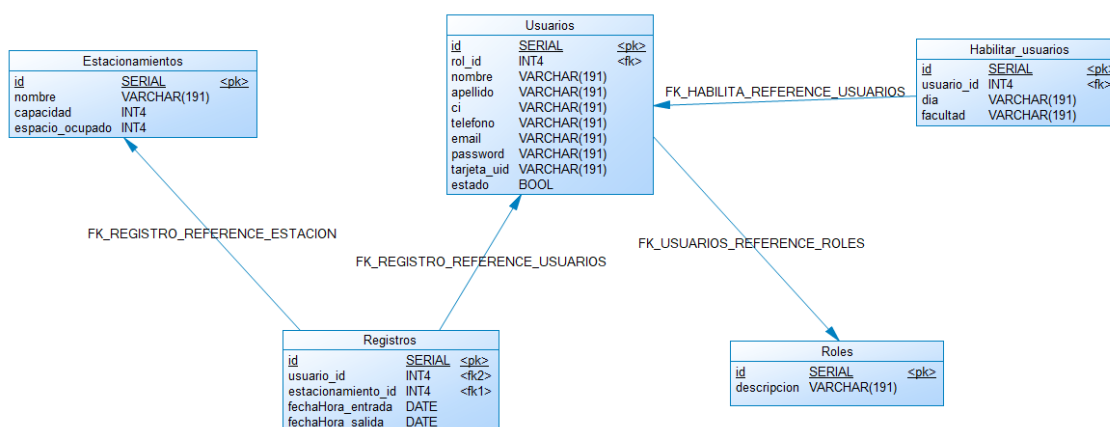
proyecto. El proceso de desarrollo comenzó el 3 de noviembre de 2023 y se completó el 30 de diciembre del mismo año. A través de este tablero, se pudo visualizar las tareas y mantener un seguimiento constante de los avances y objetivos del proyecto. Además, es importante destacar que después de estas fechas mencionadas, se realizaron ajustes adicionales para mejorar aún más el producto final.

3.2 Configuración de base de datos

Se utilizó el SGDB PostgreSQL debido a su rendimiento, estabilidad y compatibilidad con estándares SQL como se menciona en el capítulo anterior. A continuación, se muestra el diagrama de entidad-relación que se tuvo en cuenta para la configuración de la base de datos.

3.2.1. Diagrama de Entidad-Relación

Ilustración 3. Diagrama de Entidad-Relación.



Fuente: Elaboración propia.

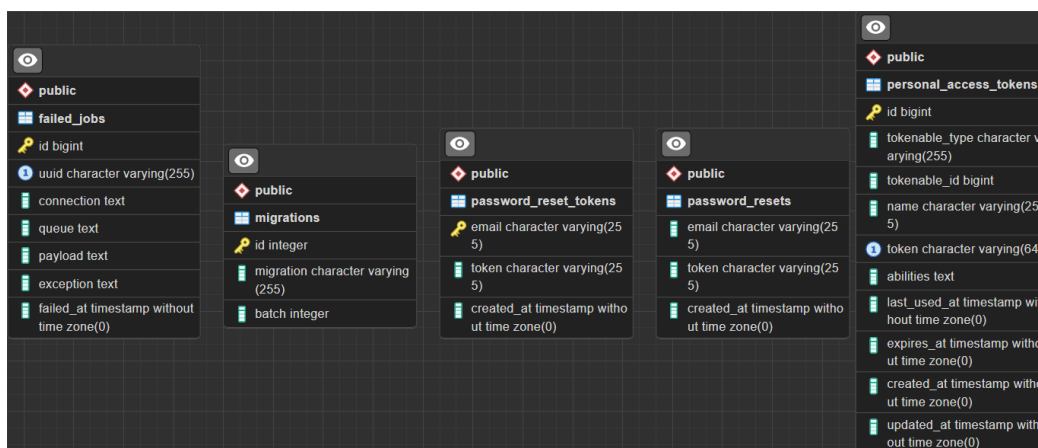
Como se aprecia en la imagen superior, este diagrama sirvió como base para lo que se deseaba realizar.

3.2.2. Diagrama físico de la base de datos

La base de datos del sistema de control de estacionamiento se ha implementado utilizando las migraciones de Laravel, una funcionalidad que permite definir de manera elegante y controlada la estructura de la base de datos. Este método proporciona una serie de

ventajas, como el seguimiento de cambios en el esquema y la posibilidad de revertir a estados anteriores de manera sencilla.

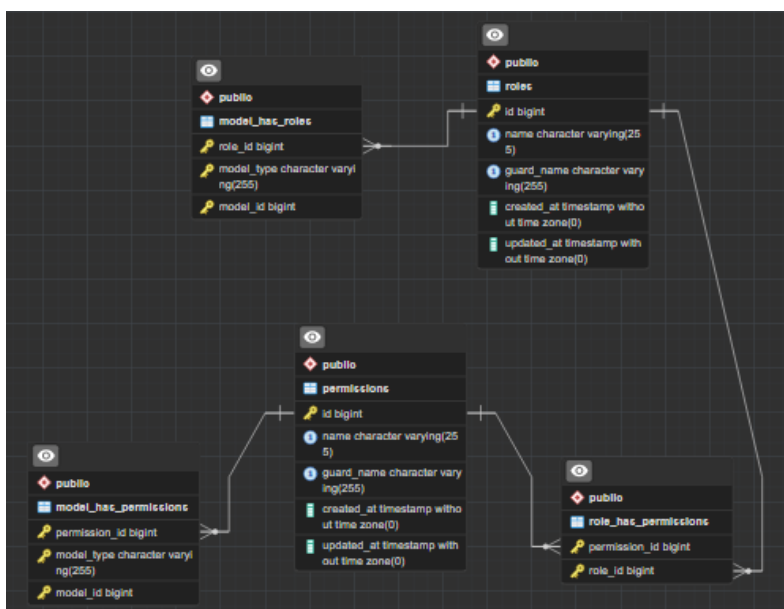
Ilustración 4. Tablas sueltas predeterminadas generadas por Laravel.



Fuente: Elaboración propia.

En la ilustración 4, se pueden observar las tablas que Laravel genera por defecto al crear un nuevo proyecto.

Ilustración 5. Tablas para roles y permisos.

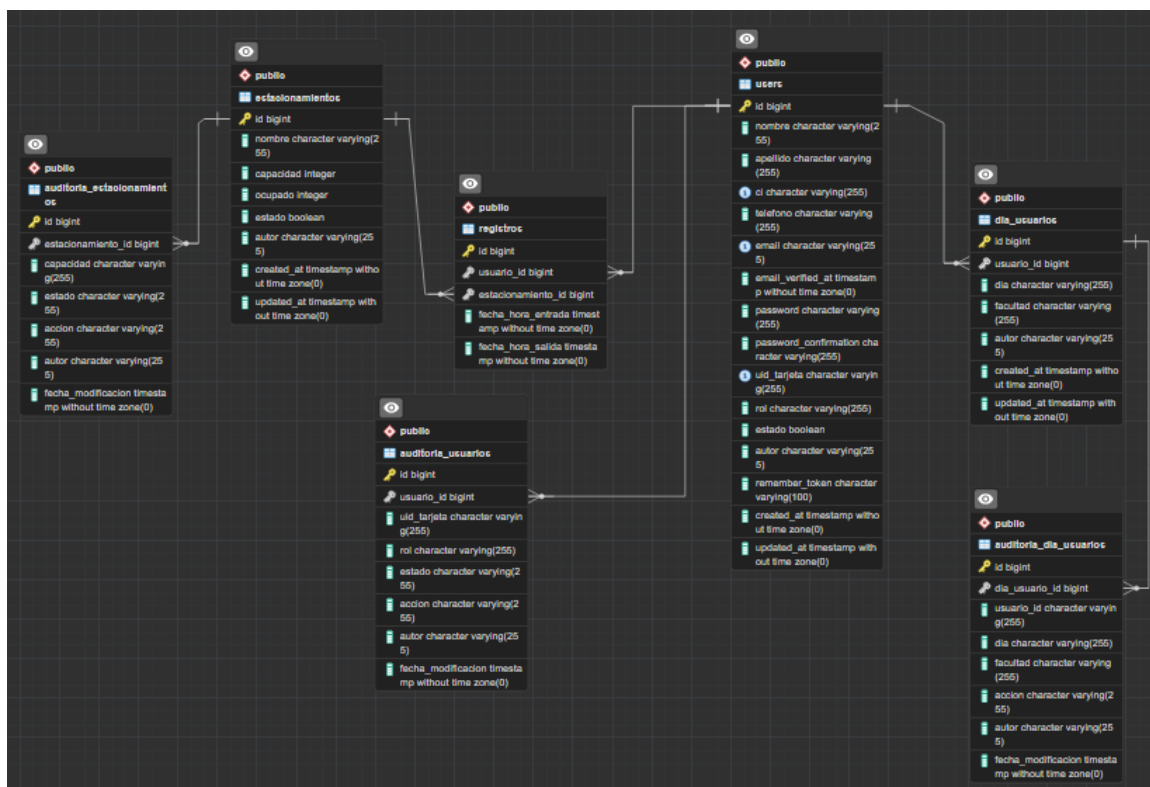


Fuente: Elaboración propia.

Como se aprecia en la Ilustración 5, estas tablas representan lo que se crea automáticamente al utilizar el paquete laravel-permission de Spatie en un proyecto

Laravel. Son fundamentales para gestionar roles y permisos de usuarios de manera eficiente en la aplicación web.

Ilustración 6. Diagrama físico de la base de datos.

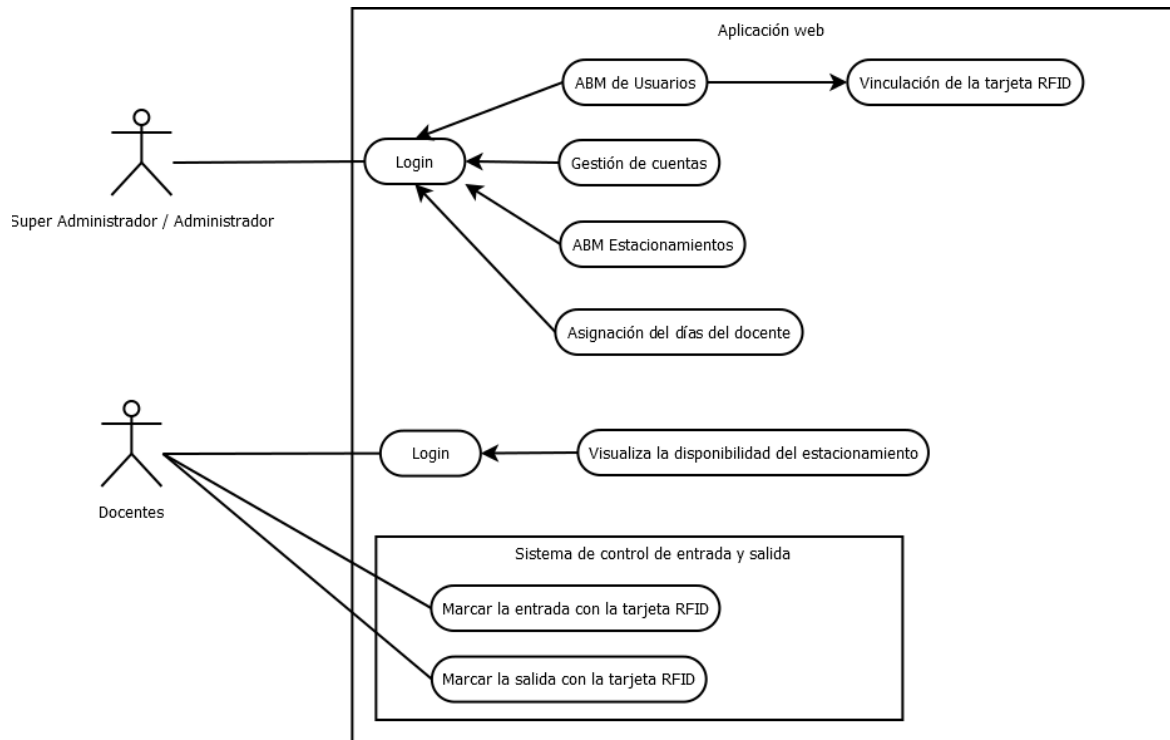


Fuente: Elaboración propia.

Como se puede observar en la Ilustración 6, estas son todas las tablas generadas para el buen funcionamiento del sistema de control de estacionamiento.

3.3 Visión general del sistema

Ilustración 7. Diagrama de visión general del sistema.



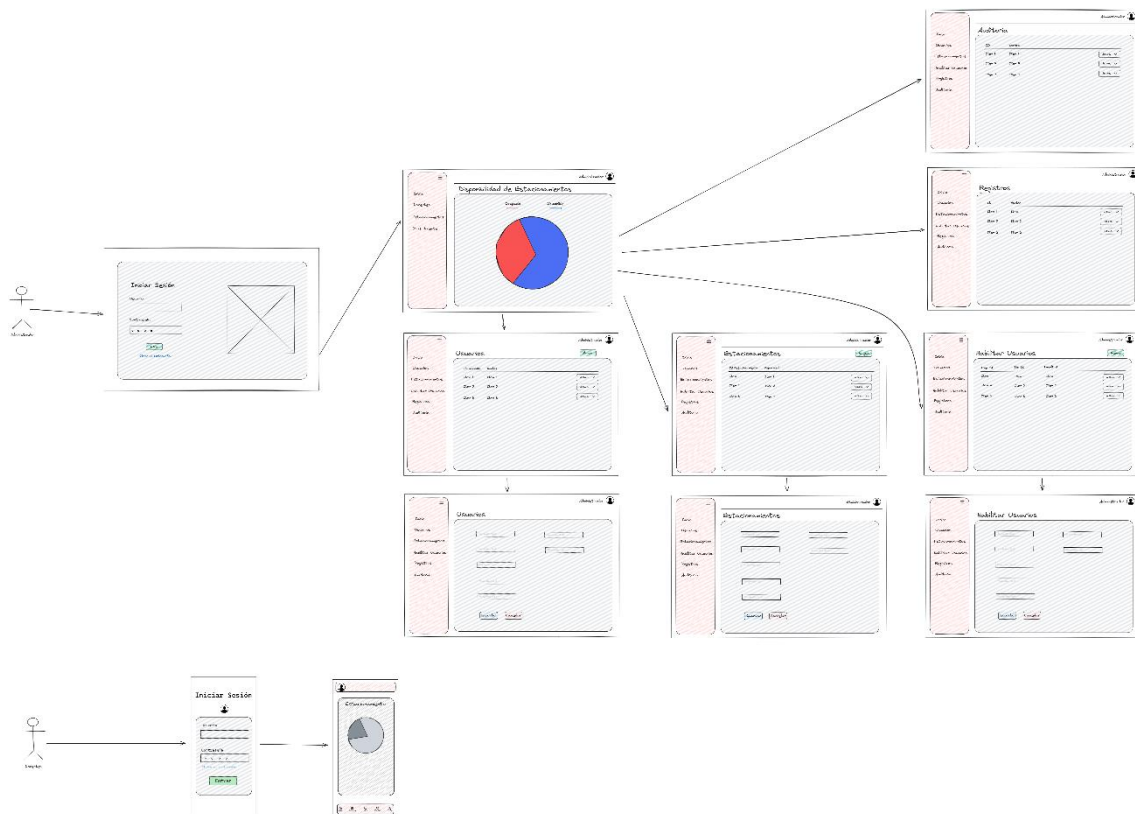
Fuente: Elaboración propia.

En la ilustración 7 se muestra el diagrama de visión general del sistema de control de estacionamiento, donde se delinean los actores específicos junto con sus casos de uso dentro del sistema. Este diagrama es esencial para entender las interacciones y permisos de cada tipo de usuario dentro del sistema.

La definición detallada de los casos de uso se encuentra en el apartado de Anexo, proporcionando una visión completa de las funcionalidades y permisos asignados a cada rol.

3.4 Interfaz navegacional

Ilustración 8. Interfaz navegacional de la aplicación web.



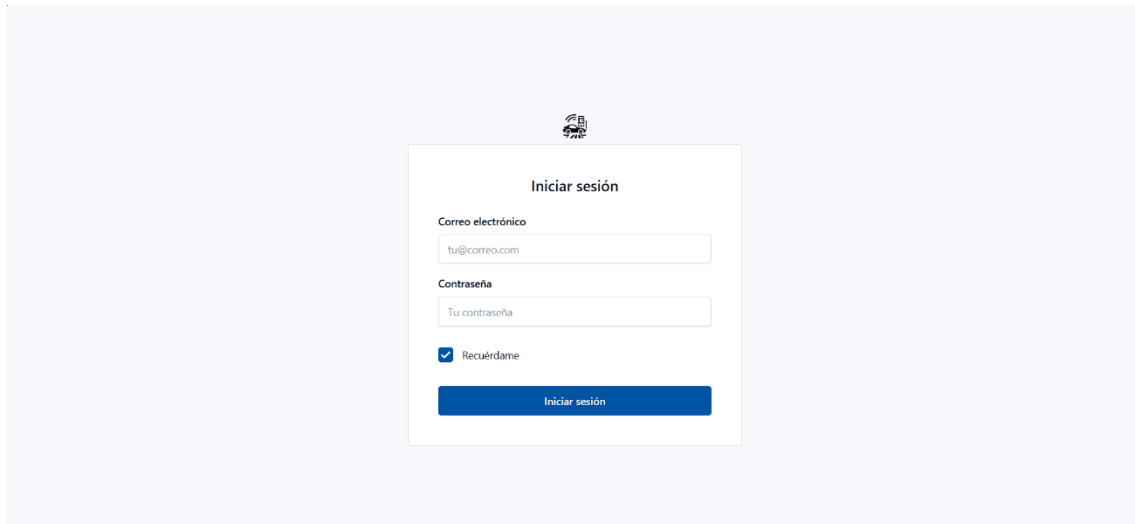
Fuente: Elaboración propia.

La interfaz navegacional se refiere al conjunto de ventanas a las que el usuario tendrá acceso durante su interacción con la aplicación web, proporcionando un camino claro y coherente para que los usuarios naveguen a través de las diversas funcionalidades y servicios ofrecidos por la aplicación.

3.5 Interfaces de la aplicación web

En este apartado se muestran algunas de las interfaces de la aplicación web que ilustran las funcionalidades y el diseño del sistema. Las interfaces adicionales se detallan en el apartado de anexo del documento.

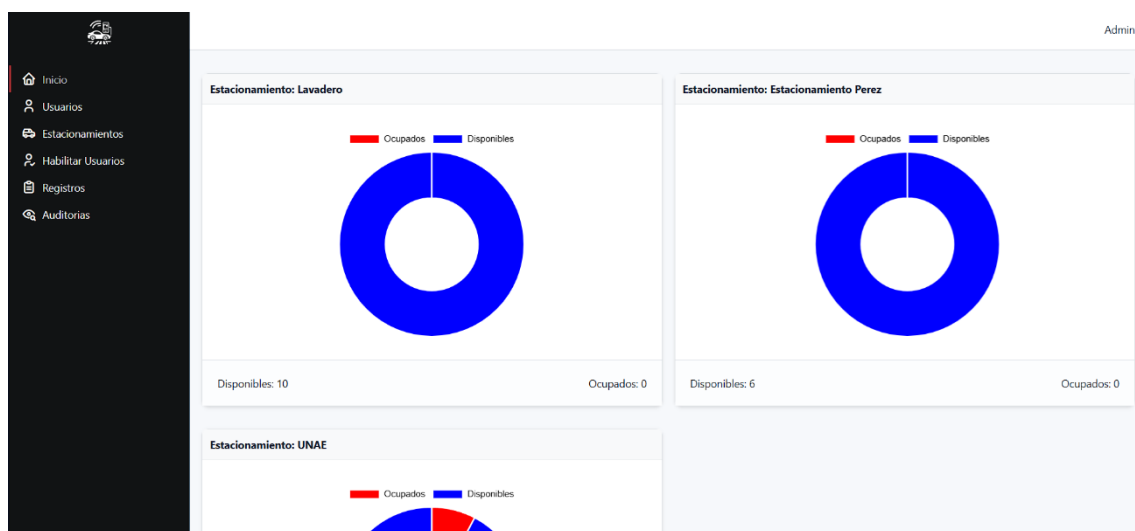
Ilustración 9. Login.



Fuente: Elaboración propia.

En la ilustración 9, se aprecia la pantalla de inicio de sesión en la que se solicitan datos de un usuario que previamente fueron creados por algún administrador. Luego de iniciar sesión éste accederá a la página de inicio.

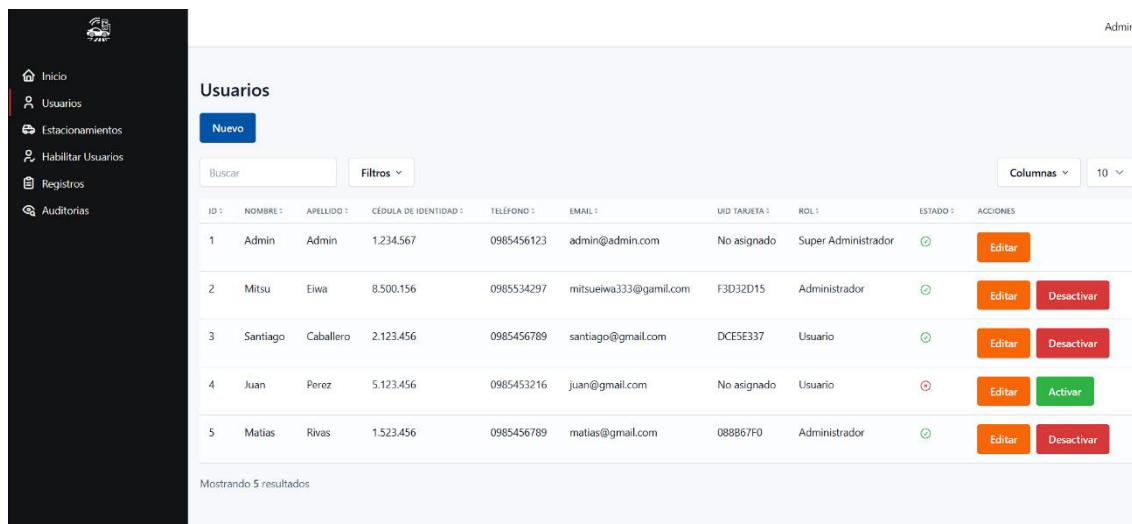
Ilustración 10. Interfaz de inicio.



Fuente: Elaboración propia.

En la imagen superior se muestra la pantalla de inicio en donde se muestran los gráficos de la disponibilidad de los estacionamientos.

Ilustración 11. Interfaz de usuarios.

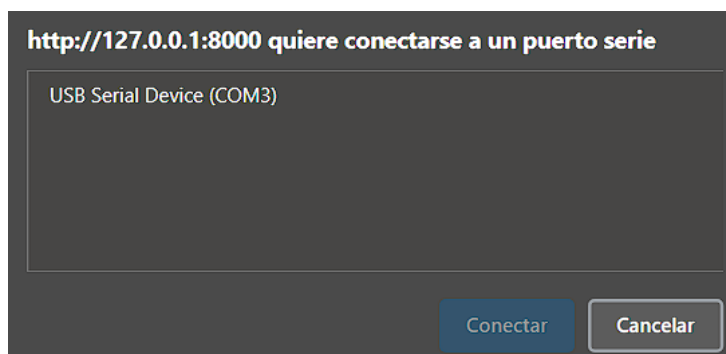


Fuente: Elaboración propia.

En la ilustración 11 se aprecia la página de gestión de usuarios, donde los administradores tienen la capacidad de añadir nuevos usuarios y asignarles una tarjeta RFID correspondiente, como también editar los datos y desactivar usuarios.

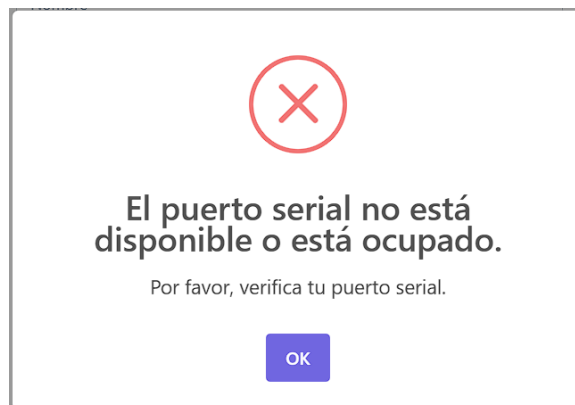
Al presionar el botón “Nuevo” para registrar un nuevo usuario se muestra un panel en donde se puede seleccionar el puerto serial en la que está conectada el Arduino como se muestra en la siguiente imagen.

Ilustración 12. Panel para seleccionar el puerto serial.

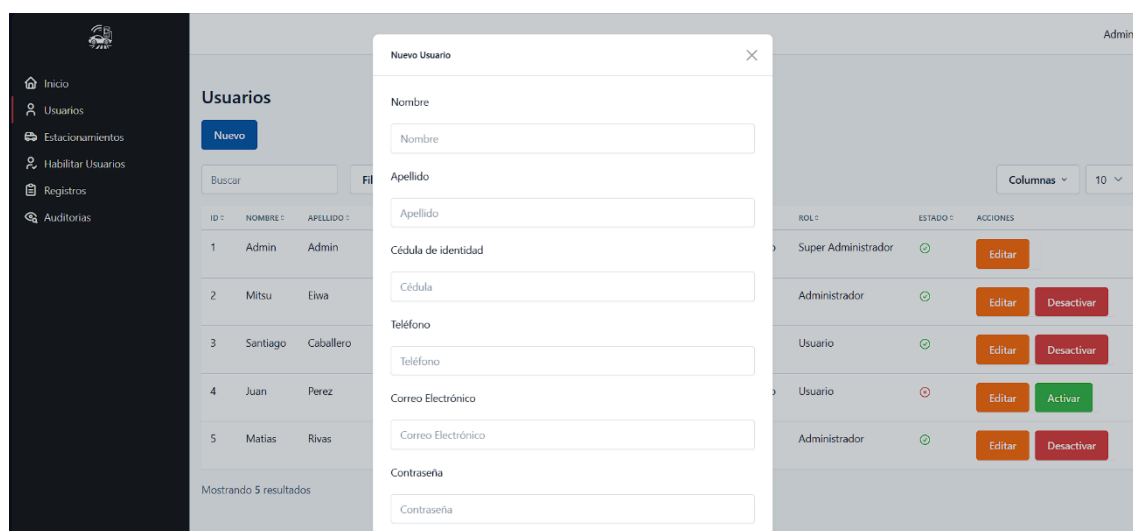


Fuente: Elaboración propia.

En caso de no conectar con ningún puerto se muestra un mensaje como se aprecia en la siguiente ilustración.

Ilustración 13. Mensaje de advertencia.


Fuente: Elaboración propia.

Ilustración 14. Crear nuevo usuario.


Fuente: Elaboración propia.

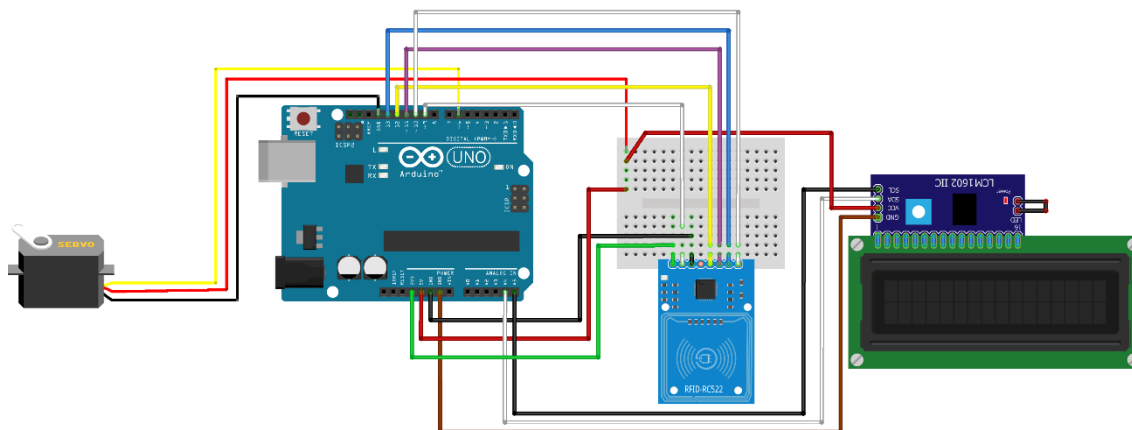
En la ilustración superior se aprecia el formulario para agregar un nuevo usuario en la cual el administrador debe cargar los datos de un nuevo usuario.

3.6 Sistema de control de entrada y salida

Para el sistema de control de entrada y salida se utilizaron Arduino UNO con el lector RFID, LCD I2C y un servo motor.

3.6.1. Conexiones de los componentes

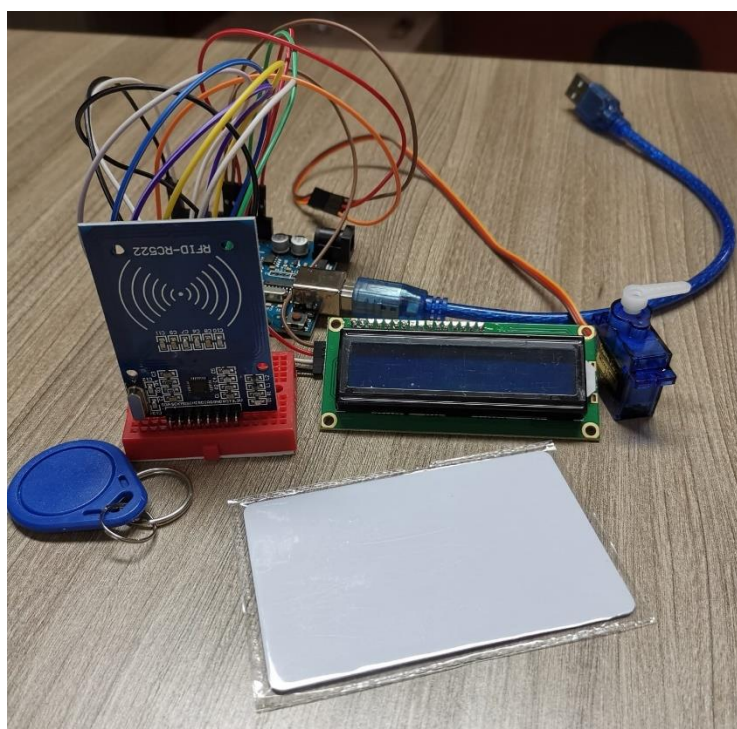
Ilustración 15. Diseño de conexión de los componentes.



Fuente: Elaboración propia.

En la imagen superior se muestra el diseño de las conexiones de los componentes con el Arduino UNO.

Ilustración 16. Conexión de los componentes.

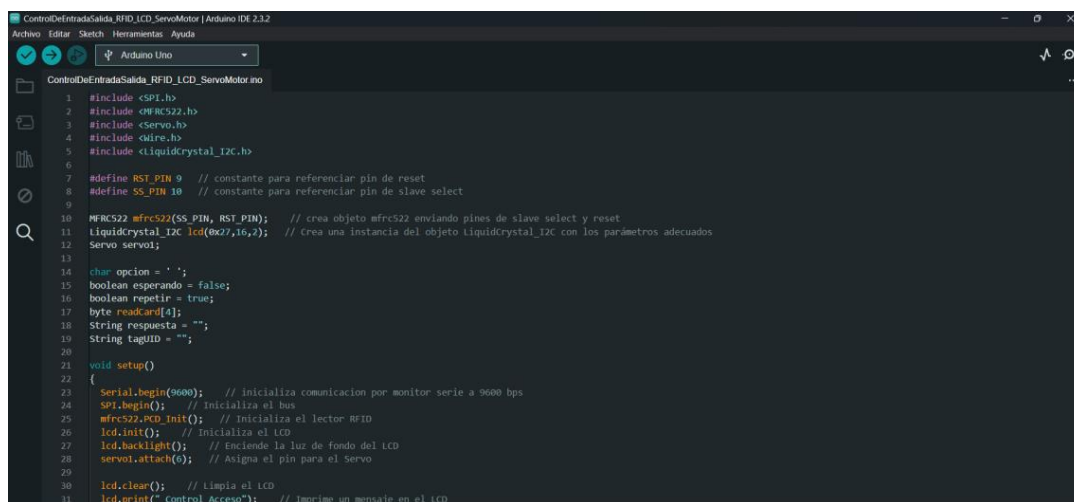


Fuente: Elaboración propia.

Como se observa en la ilustración 16, se han realizado las conexiones de los componentes con el Arduino UNO de manera precisa y ordenada. Esta configuración es esencial para el óptimo funcionamiento del sistema.

3.6.2. Desarrollo del sistema de control

Ilustración 17. Desarrollo de código en Arduino IDE.



```

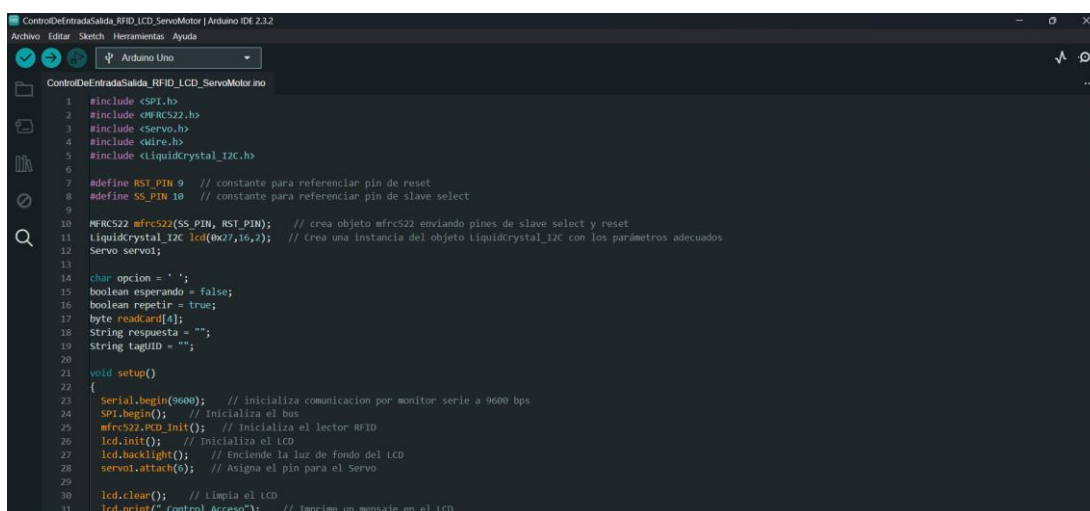
1 #include <SPI.h>
2 #include <MFRC522.h>
3 #include <Servo.h>
4 #include <Wire.h>
5 #include <LiquidCrystal_I2C.h>
6
7 #define RST_PIN 9 // constante para referenciar pin de reset
8 #define SS_PIN 10 // constante para referenciar pin de slave select
9
10 MFRC522 mfrc522(SS_PIN, RST_PIN); // crea objeto mfrc522 enviando pines de slave select y reset
11 LiquidCrystal_I2C lcd(0x27,16,2); // Crea una instancia del objeto LiquidCrystal_I2C con los parámetros adecuados
12 Servo servo;
13
14 char opcion = ' ';
15 boolean esperando = false;
16 boolean repetir = true;
17 byte readCard[4];
18 String respuesta = "";
19 String tagUID = "";
20
21 void setup()
22 {
23   Serial.begin(9600); // inicializa comunicacion por monitor serie a 9600 bps
24   SPI.begin(); // Inicializa el bus
25   mfrc522.PCD_Init(); // Inicializa el lector RFID
26   lcd.init(); // Inicializa el LCD
27   lcd.backlight(); // Enciende la luz de fondo del LCD
28   servo.attach(0); // Asigna el pin para el Servo
29
30   lcd.clear(); // Limpia el LCD
31   lcd.print(" Control Acceso"); // Imprime un mensaje en el LCD

```

Fuente: Elaboración propia.

Tras la conexión de los componentes, se realizó el desarrollo del sistema en Arduino IDE como se detalla en la ilustración 17.

Ilustración 18. Desarrollo del script en Python.



```

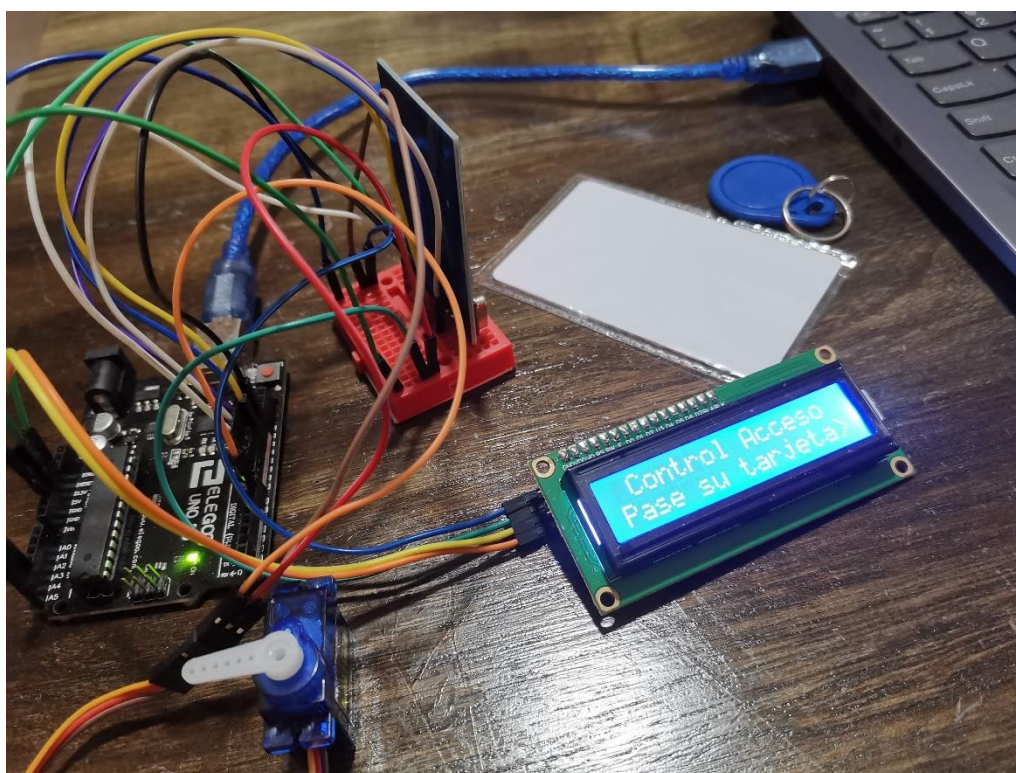
1 #include <SPI.h>
2 #include <MFRC522.h>
3 #include <Servo.h>
4 #include <Wire.h>
5 #include <LiquidCrystal_I2C.h>
6
7 #define RST_PIN 9 // constante para referenciar pin de reset
8 #define SS_PIN 10 // constante para referenciar pin de slave select
9
10 MFRC522 mfrc522(SS_PIN, RST_PIN); // crea objeto mfrc522 enviando pines de slave select y reset
11 LiquidCrystal_I2C lcd(0x27,16,2); // Crea una instancia del objeto LiquidCrystal_I2C con los parámetros adecuados
12 Servo servo;
13
14 char opcion = ' ';
15 boolean esperando = false;
16 boolean repetir = true;
17 byte readCard[4];
18 String respuesta = "";
19 String tagUID = "";
20
21 void setup()
22 {
23   Serial.begin(9600); // inicializa comunicacion por monitor serie a 9600 bps
24   SPI.begin(); // Inicializa el bus
25   mfrc522.PCD_Init(); // Inicializa el lector RFID
26   lcd.init(); // Inicializa el LCD
27   lcd.backlight(); // Enciende la luz de fondo del LCD
28   servo.attach(0); // Asigna el pin para el Servo
29
30   lcd.clear(); // Limpia el LCD
31   lcd.print(" Control Acceso"); // Imprime un mensaje en el LCD

```

Fuente: Elaboración propia.

En la ilustración 18 se muestra una parte del script desarrollado en Python que realiza la comunicación serial con el Arduino, encargado de realizar las validaciones pertinentes para el control de entrada y salida de los docentes.

Ilustración 19. Sistema de control de estacionamiento.



Fuente: Elaboración propia.

Como se puede ver en la ilustración 19 luego de realizar todas las configuraciones se procedió a conectar el sistema hardware con la computadora por el puerto serial para verificar su funcionamiento.

3.7 Casos de prueba

Es esencial implementar casos de prueba para comprobar la reacción de los campos, botones y funciones del sistema frente a la introducción de datos incorrectos, así como para asegurar que desempeñan sus roles conforme a lo previsto.

3.7.1 Casos para crear y editar usuarios por parte de los administradores

Tabla 1. Casos de prueba de usuarios.

Campo de ID requerido.	SI
Campo de ID es establecido automáticamente por el sistema.	SI
Campo de Nombre requerido.	SI
Campo de Apellido requerido.	SI
Campo de CI requerido.	SI
Campo de CI permite el ingreso de datos si es unico en la base de datos.	SI
Campo de Teléfono requerido.	SI
Campo de Correo Electrónico requerido.	SI
Campo de Correo Electrónico permite el ingreso de datos solamente en formatos válidos para correos.	SI
Campo de Contraseña requerido.	SI
Campo de Contraseña necesita mínimamente 8 caracteres.	SI
Campo de Contraseña debe tener al menos una letra mayúscula, una letra minúscula y un número.	SI
Campo de Confirmar Contraseña requerido.	SI
Campo de Contraseña y Confirmar Contraseña deben ser iguales.	SI
Campo UID Tarjeta requerido.	NO
Campo UID Tarjeta se autocompleta al pasar la tarjeta RFID por el lector luego de presionar el botón asignar.	SI
Campo UID Tarjeta se puede asignar solamente cuando está conectado al arduino con el puerto serial.	SI
Campo de Rol requerido.	SI

Fuente: Elaboración propia.

3.7.2 Casos para crear y editar estacionamientos por parte de los administradores

Tabla 2. Casos de prueba de estacionamientos.

Campo de ID requerido.	SI
Campo de ID es establecido automáticamente por el sistema.	SI
Campo de Nombre requerido.	SI
Campo de Capacidad requerido.	SI
Campo de Capacidad solamente acepta valores numéricos.	SI

Fuente: Elaboración propia.

3.7.3 Casos para crear y editar días hábiles de usuarios por parte de los administradores

Tabla 3. Casos de prueba de día hábiles de usuarios.

Campo de ID requerido.	SI
Campo de ID es establecido automáticamente por el sistema.	SI
Campo de Usuario requerido.	SI
Campo de Día requerido.	SI
Campo de Facultad requerido.	SI

Fuente: Elaboración propia.

3.7.4 Casos para el eliminado los registros de la base de datos por parte de los administradores

Tabla 4. Casos de prueba de eliminado de los registros.

Registros de Usuarios permite solamente eliminado lógico.	SI
Registros de estacionamientos permite solamente eliminado lógico.	SI
Registros de Día Hábiles de Usuarios permite solamente eliminado físico.	SI

Fuente: Elaboración propia.

4. Discusiones

Tras el análisis de los resultados obtenidos en el desarrollo del sistema de control de estacionamiento para docentes, podemos observar que los principales módulos y funcionalidades implementados fueron la aplicación web y el sistema hardware.

En cuanto a la aplicación web, esta fue desarrollada en Laravel, tal como destaca Morales (2023) que Laravel es uno de los frameworks de PHP más utilizados y populares en la actualidad, gracias a sus características como robustez, escalabilidad y facilidad de uso. Estas características han sido fundamentales para el desarrollo de una aplicación web amigable, eficiente y fácil de mantener, permitiendo a los usuarios docentes verificar la disponibilidad de espacios en el estacionamiento de manera sencilla.

Por otro lado, el sistema de hardware se basa en la utilización de Arduino Uno y tecnología RFID, para la comunicación serial del sistema de hardware se utilizó en el backend un programa hecho en Python con la biblioteca PySerial, facilitando así el intercambio de datos entre ambos. Esta integración ha sido fundamental para regular el flujo de vehículos y mantener actualizada la información de los registros de entrada y salida de los docentes.

Estos hallazgos se enmarcan dentro de las ventajas de los sistemas de control de estacionamiento inteligentes, tal como es mencionado por Hernandez & Ruiz (2019), adicionalmente se alinea con lo expuesto por Rios (2011) en donde destaca que los sistemas de control de estacionamiento inteligentes tienen un impacto positivo en la calidad de vida de las personas, al optimizar el uso de los espacios de estacionamiento y brindar información oportuna a los usuarios, lo cual se ve reflejado en el sistema desarrollado en este proyecto.

5. Conclusiones

En la universidad privada donde se realizó la investigación, la gestión de los espacios de estacionamiento para docentes presenta varias dificultades, como la falta de seguimiento y de información actualizada sobre la disponibilidad de plazas, así como la dependencia de un funcionario para su administración. Con el objetivo de resolver esta problemática, se ha desarrollado un sistema de control de estacionamiento para docentes, utilizando la

metodología de desarrollo ágil, Scrum. La aplicación de esta metodología ha permitido una mayor flexibilidad y adaptación a los cambios durante el proceso, facilitando la gestión de las actividades y el seguimiento del progreso del proyecto.

Este sistema está compuesto por dos módulos principales: una aplicación web y un sistema hardware. La aplicación web permite gestionar los usuarios y los estacionamientos, ver la disponibilidad de espacios en el estacionamiento como también, ver los registros de entrada y salida de los docentes. Por otro lado, el sistema hardware, compuesto por Arduino uno, la tecnología RFID y otros componentes, permite marcar la entrada y salida de los usuarios del estacionamiento a través de la tarjeta RFID.

El proceso de desarrollo incluyó diversas etapas, como el análisis de los requerimientos, la configuración de la metodología de desarrollo, el diseño de las interfaces y la base de datos, desarrollo de la aplicación, como también la integración de los componentes hardware con el sistema, durante dicho proceso una de las principales limitaciones encontradas fue la complejidad de realizar la comunicación serial de los dispositivos hardware con la aplicación web, lo que demandó un mayor esfuerzo y tiempo de desarrollo, que requirió soluciones creativas para asegurar una conexión estable y eficiente.

Estas limitaciones abren nuevas oportunidades para futuras investigaciones. Una línea de mejora podría centrarse en la optimización de la integración de hardware y software, explorando tecnologías inalámbricas más avanzadas que proporcionen una comunicación más confiable y sin interrupciones. Otra línea de mejora es la posibilidad de integrar nuevas funcionalidades al sistema. Por ejemplo, la incorporación de otro método para el control del flujo vehicular en el estacionamiento, como el reconocimiento de matrículas, la implementación de otras tecnologías, como sensores para detectar la ocupación de espacios específicos, así como también un sistema de señalización para guiar a los conductores en los espacios disponibles.

Este proyecto aporta un prototipo que elimina la necesidad de un control manual del estacionamiento, reduciendo la dependencia de un funcionario para la gestión de los

espacios, facilitando así el flujo de vehículos en el estacionamiento. Con estos resultados obtenidos se demuestra que el sistema de control de estacionamiento cumple con los objetivos planteados. Además, las futuras mejoras, como la optimización de la integración tecnológica y la incorporación de nuevas funcionalidades, permitirán aumentar aún más la eficiencia, la seguridad y la satisfacción de los usuarios del sistema.

6. Referencias bibliográficas

- Arsenio, G. H., Delgado, S. C., & Souberbielle, A. N. (2020). Adaptación cultural, validez discriminante y análisis de confiabilidad del test de ira en la conducción en una muestra de conductores particulares residentes en Asunción. *ACADEMO*, 82-88.
- Chávez, C. Y., & Yance, R. M. (2020, Octubre). Implementación de un prototipo para el control del uso del parqueadero en los horarios de los docentes de las Carreras de Ingeniería en Sistemas e Ingeniería en Networking. Guayaquil, Guayas, Ecuador: Universidad de Guayaquil. Facultad de Ciencias Matemáticas y Físicas. Carrera de Ingeniería en Networking y Telecomunicaciones.
- Formoso, A., Mazzilli, A., & Sotelo, R. (2014). ParkIt - Plataforma inteligente de estacionamiento público. *Memoria Investigaciones en Ingeniería*, 85-94.
- Franco, C. L. (2014). La movilidad sostenible en campus universitarios: una comparación de las mejores prácticas en Estados Unidos y Europa. Aplicabilidad en universidades venezolanas. *Revista de la Facultad de Ingeniería Universidad Central de Venezuela*, 23-40.
- González, G. S., & Olmedo, F. Á. (2019). Viabilidad de aplicación móvil para la búsqueda de plazas de estacionamientos libres en el Centro de Negocios de Asunción. *SCIENIAMERICANA, Revista Multidisciplinaria*, 17-21.
- Hernandez, J. J., & Ruiz, Y. P. (2019). Prototipo de un Si para la Gestión y Control del Estacionamiento de la Universidad del Pacifico. Universidad del Pacífico. <https://is.gd/bTasI9>

- Morales, J. (2023). *¿Qué es Laravel y cuáles son sus beneficios para desarrollo web?*
Wwwhat's New: <https://is.gd/PBWZ8z>
- Rios, V. J. (2019). *Diseño de un sistema de control vehicular basado en el acceso de espacios libres y ubicación en estacionamientos usando RFID*. PUCP:
<https://tesis.pucp.edu.pe/repositorio/handle/20.500.12404/910>
- Rodríguez, M. G., Santos, O. R., Ordaz, B. C., & Lopez, R. J. (2019). Estacionamiento inteligente. *Revista de Ingenieria Innovativa*, 34-39.
- Última Hora. (2022, Agosto 12). *Caos reina en Asunción por la falta de espacios para estacionamiento*. UltimaHora: <https://is.gd/1Pp1KG>