

Smart Parking



ASIGNATURA

Internet de las Cosas

DOCENTES

Danny Alexandro Munera Ramirez

Henry Alberto Arcila Ramirez

Jose Ignacio Lopez Velez

RESPONSABLES

Juan Felipe Zora Zuluaga

Manuel Calle Garcés

Esteban Durango Vásquez

UNIVERSIDAD DE ANTIOQUIA
MEDELLÍN
2023

Contenido

1. Resumen del proyecto.....	3
2. Introducción y contextualización del proyecto.....	3
2.1 Planteamiento del problema.....	3
2.2 descripción resumida de la solución planteada.....	3
2.3 Objetivo general.....	4
2.4 Objetivos específicos.....	4
3. Descripción de la solución.....	5
4. Implementación componentes de hardware “cosas”	7
4.1 Sensores magnéticos:.....	7
4.2 Cámara para validar ingreso.....	7
4.3 Servomotor para abrir la puerta.....	8
4.4 Led RGB para dar información.....	9
4.5 Integración y control del sistema.....	9
5. Implementación componentes de software.....	10
5.1 Base de datos.....	10
5.2 SQLAlchemy ORM.....	10
5.3 Modelo de Machine Learning usando OpenCV y PyTesseract.....	10
5.4 MQTT Client.....	10
5.5 Kivy para el front-end.....	11
6. Conclusiones.....	11
7. Trabajo futuro.....	12
8. Referencias.....	12

1. Resumen del proyecto

Un piloto a escala de laboratorio del sistema de control de acceso para una unidad residencial y parqueadero inteligente compuesto por: un sensor que detectará cuando un carro se estacione en la portería de la unidad residencial, ocasionando que una cámara ubicada con visual hacia la placa de este coche tome una foto. Esta foto será procesada por un modelo de Machine Learning para hacer identificación de la placa y posteriormente se contrastará con una base de datos de placas reconocidas por la unidad residencial para permitir o denegar el pase del coche por la portería. Una vez confirmada la placa, se iluminará una o varias luces que indican el puesto de estacionamiento que corresponde a dicha placa.

2. Introducción y contextualización del proyecto

2.1 Planteamiento del problema

La entrada y salida de vehículos en parqueaderos de unidades residenciales puede ser una tarea tediosa y lenta, especialmente en lugares donde se requiere la identificación y autorización manual de los vehículos. Esto puede generar congestiones en los accesos, retrasos en los tiempos de espera y, en algunos casos, incluso aumentar el riesgo de errores humanos y seguridad.

Para solucionar este problema, se puede utilizar tecnología IoT (Internet de las cosas) para automatizar la identificación y autorización de los vehículos autorizados en tiempo real. De esta manera, se mejora la eficiencia y rapidez en el proceso de ingreso, disminuyendo la congestión y los retrasos en los tiempos de espera.

2.2 descripción resumida de la solución planteada

El proyecto tiene como objetivo mejorar la gestión y eficiencia de los estacionamientos a través de la utilización de sensores y cámaras, se busca validar el ingreso de vehículos mediante la lectura de placas.

El sistema tiene varios componentes clave. En primer lugar, se instalan sensores en el ingreso del estacionamiento para detectar si está ingresando un vehículo. Estos sensores pueden utilizar tecnología de detección de presencia o magnetismo para determinar la presencia del vehículo y activar una cámara conectada a una raspberry pi, que tomará una foto a la placa y validará el ingreso.

Esta cámara está equipada con un software de reconocimiento de placas que utiliza algoritmos de procesamiento de imágenes para identificar y extraer los caracteres de la placa del vehículo.

Una vez que se ha detectado un vehículo y se ha leído su placa, el sistema realiza una validación para determinar si el vehículo tiene permiso para ingresar al estacionamiento. Esta validación se puede realizar comparando la placa leída con una base de datos de placas autorizadas o mediante la comunicación con un sistema externo, como un sistema de gestión de acceso.

2.3 Objetivo general

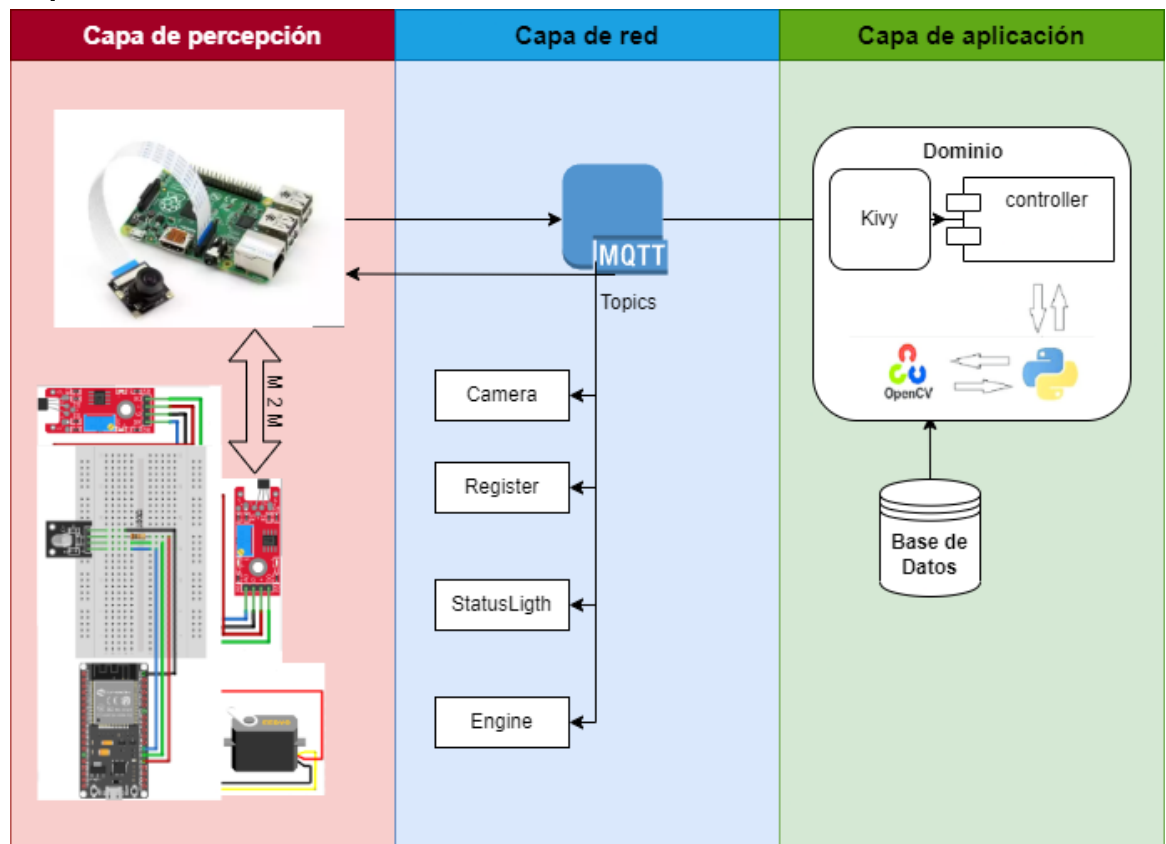
Mejorar la eficiencia y la experiencia de los usuarios en los estacionamientos mediante la implementación de un sistema de Smart Parking con validación de ingreso por reconocimiento de placa.

2.4 Objetivos específicos

- Implementar una cámara en la entrada del estacionamiento para capturar imágenes de los vehículos que ingresan.
- Desarrollar un algoritmo de reconocimiento de placas que sea capaz de extraer los caracteres de las placas de los vehículos en las imágenes capturadas.
- Establecer una base de datos de placas autorizadas que permita validar el ingreso de los vehículos.
- Integrar el algoritmo de reconocimiento de placas con el sistema de gestión del estacionamiento para realizar la validación en tiempo real.
- Generar alertas o notificaciones en caso de que se detecte un vehículo no autorizado intentando ingresar al estacionamiento.
- Realizar pruebas exhaustivas del sistema para garantizar su precisión y confiabilidad en la detección y validación de placas.

3. Descripción de la solución

Arquitectura del sistema



En este diagrama, la Raspberry Pi actúa como el dispositivo central de procesamiento y control del sistema de Smart Parking. La cámara de reconocimiento de placas se conecta a la Raspberry Pi para capturar imágenes de la placa de los vehículos que ingresan al estacionamiento. El algoritmo de reconocimiento de placas se ejecuta en la Raspberry Pi, utilizando técnicas de procesamiento de imágenes implementando la librería OpenCV para extraer los caracteres de las placas.

La base de datos de placas autorizadas se encuentra en la Raspberry Pi y almacena la información de las placas permitidas para el ingreso al estacionamiento. El MQTT Broker actúa como el intermediario de mensajería entre la Raspberry Pi y los dispositivos ESP32. Facilita la transmisión de mensajes entre ellos.

A continuación se exponen los tópicos que se usarán en MQTT

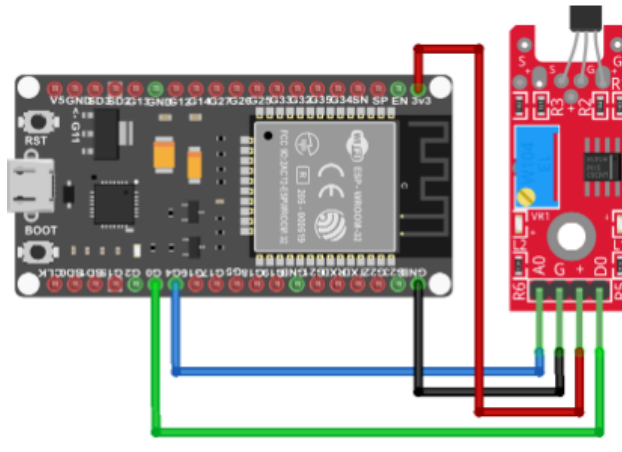
Nombre de tópico	Funcionalidad	Explicación
Engine	Tópico para la interacción con el motor de la entrada	Cuando se publica un mensaje con payload equivalente al string "1" en este tópico, el motor de la entrada se abre. Cuando se publica un "0", el motor de la entrada se cierra.
Camera	Tópico para interactuar con la cámara.	Cuando se publica un mensaje con payload equivalente al string "1" en este tópico la cámara de la raspberry pi se prende y toma una foto.
Statuslight	Tópico para interactuar con la luz led RGB	Si se publica un mensaje con payload equivalente al string: "1" se prende la luz de color rojo que simboliza que se ha denegado la entrada a la unidad. "2" se prende de color azul que significa que se está procesando la placa . "3" se prende de color verde que significa que se ha habilitado el acceso a la unidad.
Register	Tópico auxiliar en el que se registran los eventos de registro de Vehículo en la Base de Datos	Cualquier mensaje publicado a este tópico se escribirá en el frontend de la aplicación.

El sistema de gestión del parking se encuentra en la Raspberry Pi y utiliza el algoritmo de reconocimiento de placas y la base de datos de placas autorizadas para validar el ingreso de los vehículos.

La pantalla de visualización permite registrar nuevas placas y asociar estas placas a un propietario.

4. Implementación componentes de hardware “cosas”

4.1 Sensores magnéticos:



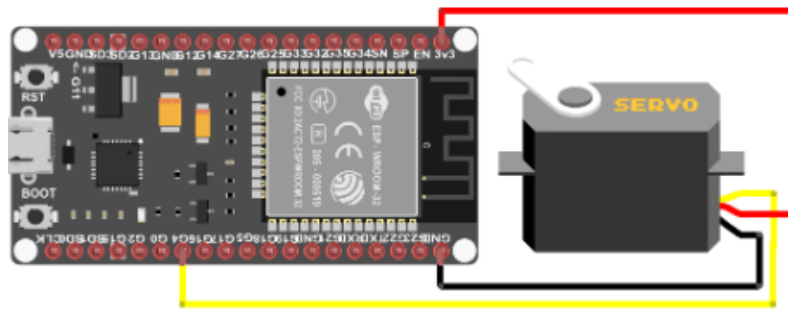
- Conectar los sensores magnéticos a la placa de desarrollo ESP32.
- Configurar y calibrar los sensores magnéticos según las necesidades del proyecto. Esto puede incluir establecer umbrales de detección para determinar si hay un vehículo en la entrada.
- Programar el microprocesador para leer los datos de los sensores magnéticos y enviar la información relevante a la Raspberry Pi.

4.2 Cámara para validar ingreso



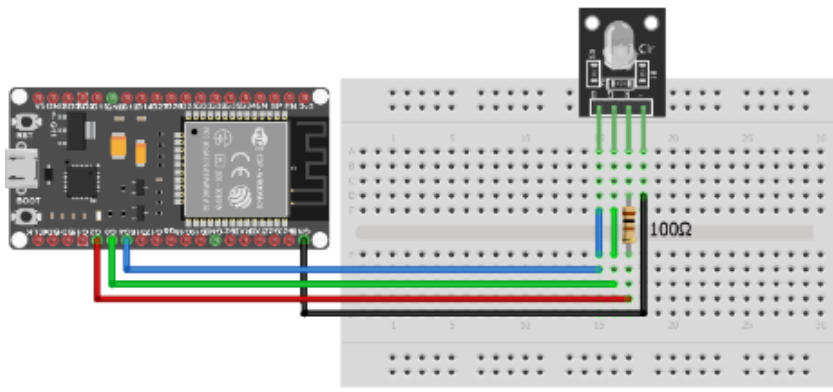
- Conectar la cámara a la Raspberry Pi o al microcontrolador utilizando los puertos adecuados y siguiendo las instrucciones del fabricante.
- Instalar y configurar el software de reconocimiento de placas en la Raspberry Pi o en el sistema embebido utilizado.
- Programar el sistema para capturar imágenes de los vehículos que ingresan al estacionamiento utilizando la cámara.
- Utilizar el software de reconocimiento de placas para procesar las imágenes capturadas y extraer los caracteres de las placas de los vehículos.

4.3 Servomotor para abrir la puerta



- Conectar el servomotor al microcontrolador utilizando los pines adecuado.
- Configurar los parámetros necesarios para controlar el servomotor, como la velocidad de rotación y el ángulo de apertura de la puerta.
- Programar el sistema para enviar las señales adecuadas al servomotor cuando se valide el ingreso de un vehículo.
- Implementar la lógica de control para abrir y cerrar la puerta en función de las condiciones de validación de ingreso.

4.4 Led RGB para dar información.



- Conectar el led a la placa ESP32 asignar un pin para cada color
- Implementar la lógica de control de los colores
 - Rojo: Denegar acceso
 - Azul: procesando información
 - Verde: Acceso habilitado
- Programar el sistema para mostrar el color correspondiente al tópico que reciba

4.5 Integración y control del sistema

- Programar la Raspberry Pi para recibir datos de los sensores magnéticos y de la cámara, y procesar la información de validación de ingreso.
- Implementar la lógica necesaria para coordinar la acción del servomotor de apertura de puerta según los resultados de la validación.
- Configurar la comunicación con otros sistemas o dispositivos, como una base de datos o una aplicación móvil, para registrar la información de ingreso y enviar notificaciones si es necesario.

5. Implementación componentes de software

El lenguaje de programación escogido para la implementación del proyecto fue Python debido a su librería PahoMQTT de fácil uso y a su dominancia como lenguaje predilecto para temas relacionados con machine learning.

5.1 Base de datos

Para el back-end se usó una base de datos relacional usando el motor SQL Lite. Esta base de datos consta de tres tablas, las cuales son

Vehículo, la cual contiene información del vehículo siendo la placa y el usuario asociado a este vehículo los campos más importantes; Usuario, la cual contiene información de un usuario de la unidad y donde su cédula de identidad es usada como primary key; y Celda la cual representa las celdas de parqueo disponibles en el parqueadero a través de un campo booleano de estatus.

5.2 SQLAlchemy ORM

Para la interacción entre la base de datos y nuestro back-end se decidió utilizar un ORM que nos permitía estructurar la base de datos y posteriormente registrar vehículos, usuarios y celdas.

5.3 Modelo de Machine Learning usando OpenCV y PyTesseract

Luego de que la cámara de la Raspberry Pi haya tomado una foto a la placa del carro debemos realizar un reconocimiento del texto o contenido de esta placa. Para realizar esta tarea, utilizamos la librería de OpenCV para realizar un pre-procesamiento de la imagen, el cual consiste en buscar un contorno con forma de rectángulo y recortar para eliminar el ruido de la imagen. Luego se pasa la imagen recortada a una escala de colores grises y blancos para hacer más fácil el reconocimiento del modelo. Una vez la foto está pre-procesada, usamos la librería de reconocimiento de texto PyTesseract para identificar el texto de la placa.

5.4 MQTT Client

Es necesario correr una instancia de un cliente MQTT junto con nuestra aplicación de backend para poder enviar mensajes a tópicos y así poder comunicarnos con las cosas de IoT. Cabe destacar que en el mismo dispositivo donde corre el back-end (Raspberry Pi), también corre el broker MQTT.

5.5 Kivy para el front-end

Finalmente usamos la librería Kivy para disponer de un front sencillo que nos permitirá habilitar la funcionalidad de registrar placas de vehículos por medio de un formulario. En nuestro proyecto decidimos usar Kivy de la siguiente manera: se crea un componente de front en el archivo "edge.kv" en el cual ponemos los elementos que necesitamos (labels, text inputs, botones) y ligamos a estos ids y acciones (por ejemplo el método de registrar placa). Luego en nuestro archivo de app.py que es el que se ejecutará para poner a funcionar

nuestra aplicación, inicializamos este elemento de front a la vez que el cliente MQTT.

6. Conclusiones

La tecnología IoT utilizada en este proyecto permite una comunicación fluida entre los diferentes componentes del sistema, como cámaras, sensores, bases de datos y dispositivos. Esto garantiza una operación eficiente y en tiempo real, brindando a los usuarios una experiencia de estacionamiento más conveniente y sin contratiempos.

Además, la implementación de una solución de Smart Parking con validación de ingreso por reconocimiento de placa ofrece beneficios adicionales, como la capacidad de generar datos y análisis precisos sobre el uso del estacionamiento. Esto puede ayudar a los administradores a tomar decisiones informadas sobre la gestión del espacio, la planificación de la capacidad y la optimización de los recursos.

En resumen, el proyecto de Smart Parking ofrece una solución eficiente y automatizada para mejorar la gestión y la experiencia de estacionamiento. Al aprovechar la tecnología IoT, se logra una integración sin problemas de los diferentes componentes del sistema, lo que resulta en una operación más eficiente y una mayor satisfacción para los usuarios.

7. Trabajo futuro

Registrar la hora de ingreso de los vehículos autorizados y marcar los espacios de estacionamiento correspondientes como ocupados en la base de datos del sistema.

Mejoras en la precisión y velocidad del reconocimiento de placa: Se puede trabajar en el desarrollo de algoritmos más avanzados y técnicas de procesamiento de imágenes para lograr una mayor precisión en el reconocimiento de placas y reducir los tiempos de respuesta. Esto permitiría una validación más rápida y confiable de los vehículos en la entrada del estacionamiento.

Integración con sistemas de pago y reserva: Smart Parking puede evolucionar para incluir una integración completa con sistemas de pago y reserva en línea. Esto permitiría a los conductores reservar plazas de estacionamiento con anticipación y realizar pagos de manera más conveniente, ya sea a través de aplicaciones móviles, tarjetas de crédito o sistemas de pago sin contacto.

Implementación de sensores de ocupación en tiempo real: La instalación de sensores de ocupación en cada plaza de estacionamiento permitiría monitorear en tiempo real la disponibilidad de plazas. Esto ayudaría a los conductores a encontrar rápidamente lugares vacantes y optimizar la utilización del espacio.

Uso de analítica de datos para la gestión del estacionamiento: La recopilación y análisis de datos generados por el sistema de Smart Parking puede proporcionar información valiosa para la toma de decisiones. Se pueden utilizar herramientas de análisis de datos para comprender los patrones de uso del estacionamiento, identificar áreas de congestión, mejorar la planificación de la capacidad y optimizar la eficiencia operativa.

8. Referencias

1. PAIDI, Vijay, et al. Smart parking sensors, technologies and applications for open parking lots: a review. *IET Intelligent Transport Systems*, 2018, vol. 12, no 8, p. 735-741.
2. SALAZAR, Darwin Santiago Aldás, et al. Procesamiento de imágenes con visión artificial para el reconocimiento de placas vehiculares. *SATHIRI*, 2016, no 11, p. 133-142.
3. ARNEDO MENDOZA, Moises; CAIDEDO PACHECO, Arnold; FUENTES VILLERA, Maria Antonia. *Diseño y desarrollo de un prototipo para identificación de placas vehiculares y reconocimiento de caracteres en tiempo real implementado en raspberry pi 3 para la Universidad del Sinú seccional Cartagena sede santillana*. 2018. Tesis Doctoral. Universidad del Sinu, seccional Cartagena.
4. KHANNA, Abhirup; ANAND, Rishi. IoT based smart parking system. En *2016 international conference on internet of things and applications (IOTA)*. IEEE, 2016. p. 266-270.