

# Sistema de Control de Acceso con Raspberry Pi y Reconocimiento de Placas

September 14, 2024

## Contents

<b>1</b>	<b>Descripción General</b>	<b>2</b>
<b>2</b>	<b>Especificaciones del Sistema</b>	<b>2</b>
2.1	Hardware Utilizado . . . . .	2
2.2	Software Utilizado . . . . .	2
<b>3</b>	<b>Instalación y Configuración del Sistema</b>	<b>2</b>
3.1	Preparación de la Tarjeta microSD . . . . .	2
3.2	Instalación de Dependencias . . . . .	3
<b>4</b>	<b>Conexiones del Hardware</b>	<b>3</b>
4.1	Sensor PIR . . . . .	3
4.2	Relé y Láser . . . . .	3
4.3	Cámara . . . . .	3
<b>5</b>	<b>Código Fuente</b>	<b>3</b>
5.1	Consulta a Firebase . . . . .	4
<b>6</b>	<b>Pruebas del Sistema</b>	<b>4</b>
6.1	Prueba del Sensor PIR . . . . .	4
6.2	Prueba de Captura de Matrícula . . . . .	4
6.3	Prueba del Relé y Láser . . . . .	5
<b>7</b>	<b>Conclusiones y Mejoras Futuras</b>	<b>5</b>
7.1	Resultados . . . . .	5
<b>8</b>	<b>Recursos y Herramientas</b>	<b>5</b>

# 1 Descripción General

Este proyecto tiene como objetivo controlar el acceso de vehículos a un parqueadero mediante reconocimiento de matrículas, empleando una Raspberry Pi, cámaras, sensores y una base de datos en Firebase.

## Componentes Principales:

- **Hardware:** Raspberry Pi 4, Cámara, Relé, Láser, Sensor PIR.
- **Software:** Python, Firebase, OpenCV, RPi.GPIO.

# 2 Especificaciones del Sistema

## 2.1 Hardware Utilizado

- **Raspberry Pi 4** (2GB RAM)
- **Cámara:** Módulo de 5MP con lente de foco fijo
- **Relé:** Control de activación/desactivación del láser
- **Láser:** Indicador de barrera
- **Sensor PIR:** Detección de vehículos
- **Cables, conectores, y fuente de alimentación** (5V/3A)

## 2.2 Software Utilizado

- **Raspberry Pi OS**
- **Python 3**
- **Librerías:** RPi.GPIO, opencv-python, firebase-admin

# 3 Instalación y Configuración del Sistema

## 3.1 Preparación de la Tarjeta microSD

Para instalar el sistema operativo, se usó Raspberry Pi Imager para grabar Raspberry Pi OS en la tarjeta microSD. A continuación se habilitó SSH y se configuró la red Wi-Fi.

## 3.2 Instalación de Dependencias

Una vez iniciada la Raspberry Pi, se instalaron las dependencias necesarias para el proyecto.

```
sudo apt update
sudo apt install python3-rpi.gpio python3-opencv
pip install firebase-admin
```

## 4 Conexiones del Hardware

### 4.1 Sensor PIR

El sensor PIR se conectó de la siguiente manera:

- **VCC:** Pin 5V de la Raspberry Pi
- **GND:** Pin GND de la Raspberry Pi
- **OUT:** Pin GPIO 17 de la Raspberry Pi

### 4.2 Relé y Láser

El relé está conectado a GPIO 27 para controlar el encendido del láser.

### 4.3 Cámara

La cámara se conectó al puerto CSI de la Raspberry Pi.

## 5 Código Fuente

A continuación se presenta el código fuente utilizado para el control del relé y el láser.

```
import RPi.GPIO as GPIO
import time

# Configuración del pin para el relé
RELAY_PIN = 27
GPIO.setmode(GPIO.BCM)
GPIO.setup(RELAY_PIN, GPIO.OUT)
```

```
def activate_laser():
    GPIO.output(RELAY_PIN, GPIO.HIGH)
    time.sleep(5)
    GPIO.output(RELAY_PIN, GPIO.LOW)
```

## 5.1 Consulta a Firebase

Para verificar la matrícula del vehículo, se utilizó el siguiente código para conectarse a Firebase.

```
import firebase_admin
from firebase_admin import credentials, firestore

cred = credentials.Certificate("/ruta/a/credenciales.json")
firebase_admin.initialize_app(cred)
db = firestore.client()

def verificar_placa(placa):
    doc_ref = db.collection('reservas').document(placa)
    doc = doc_ref.get()
    if doc.exists:
        return True
    else:
        return False
```

## 6 Pruebas del Sistema

### 6.1 Prueba del Sensor PIR

Se verificó que el sensor PIR detecte correctamente cuando un vehículo se aproxima al área de detección.

### 6.2 Prueba de Captura de Matrícula

Se realizaron pruebas para verificar que la cámara capturara correctamente la matrícula del vehículo y procesara la imagen para obtener el número de la placa.

### **6.3 Prueba del Relé y Láser**

Se probó el funcionamiento del relé, asegurando que el láser se encendiera y apagara correctamente según la lógica del código.

## **7 Conclusiones y Mejoras Futuras**

### **7.1 Resultados**

El sistema de control de acceso funcionó correctamente, capturando imágenes, verificando matrículas en Firebase y controlando el acceso con el láser.

## **8 Recursos y Herramientas**

- Raspberry Pi Imager: <https://www.raspberrypi.com/software/>
- Firebase Admin SDK: <https://firebase.google.com/docs/admin/setup>
- OpenCV Python: <https://opencv.org/>