

**UNIVERSIDAD DE TARAPACÁ**

**FACULTAD DE INGENIERÍA**



Departamento de Ingeniería en Computación e  
Informática



# **Manual de Usuario “EstacionaFlash”**

**Autores: Álvaro Guarachi  
Rafael Nakata  
Erik Correa  
Ángel Cabezas**

**Asignatura: Proyecto II  
Profesor: Diego Aracena**

ARICA, 22/12/2025

**Tabla I. Historial de Cambios**

<b>Fecha</b>	<b>Versión</b>	<b>Descripción</b>	<b>Autor(es)</b>
20/12/2025	1.0	Versión preliminar Informe	Rafael Nakata
23/12/2025	1.2	Actualizacion del Informe	Erik Correa
24/12/2025	1.3	Version Final	Rafael Nakata

# Índice

Tabla I. Historial de Cambios.....	2
<b>Índice.....</b>	<b>3</b>
<b>Introducción.....</b>	<b>4</b>
<b>Objetivo General.....</b>	<b>4</b>
<b>Diccionario.....</b>	<b>4</b>
<b>Instalación.....</b>	<b>5</b>
<b>Interfaz gráfica.....</b>	<b>5</b>
Vista Administrador.....	6
<b>Funcionalidades Principales.....</b>	<b>7</b>
Reconocimiento de Patentes (LPR).....	7
Control de Acceso y Barrera.....	7
Monitoreo de Capacidad (Sectores A y B).....	7
<b>Implementación.....</b>	<b>8</b>
<b>Soluciones a Problemas.....</b>	<b>9</b>
<b>Conclusión.....</b>	<b>9</b>
<b>Referencias.....</b>	<b>11</b>

## **Introducción**

En la actualidad, la gestión de estacionamientos en entornos urbanos presenta diversos desafíos, tales como la congestión vehicular, el uso ineficiente de los espacios disponibles y el alto consumo energético. Estos problemas afectan tanto a los usuarios finales como a los administradores de los estacionamientos.

EstacionaFlash surge como una solución tecnológica basada en Internet de las Cosas (IoT) y visión computacional, orientada a optimizar el uso de los espacios, automatizar el proceso de acceso y pago, y mejorar la experiencia del usuario mediante información en tiempo real.

Este manual de usuario tiene como objetivo guiar al usuario en la comprensión y uso correcto del sistema EstacionaFlash, describiendo sus funcionalidades, componentes y flujo de operación.

## **Objetivo General**

Desarrollar e implementar un sistema inteligente de gestión de estacionamientos que permita:

- Monitorear en tiempo real la ocupación de los sectores.
- Guiar a los conductores hacia espacios disponibles.
- Automatizar el registro de patentes, el cálculo del tiempo de uso y el cobro.
- Controlar el acceso y salida de vehículos.
- Optimizar el consumo energético mediante sensores de movimiento.

## Diccionario

**Raspberry Pi:** Microcomputador utilizado como unidad central del sistema.

**Sensor ultrasónico:** Sensor que mide distancia mediante ondas ultrasónicas para detectar presencia de vehículos.

**Sensor PIR:** Sensor infrarrojo pasivo utilizado para detectar movimiento.

**LPR:** Sistema de reconocimiento automático de patentes mediante cámara.

**Sector:** Área delimitada del estacionamiento con un número definido de espacios.

**Servomotor:** Motor utilizado para controlar el brazo de acceso y salida.

**Base de Datos:** Sistema donde se almacenan patentes, horarios y pagos.

**IoT:** Internet de las Cosas, interconexión de dispositivos inteligentes.

## Instalación

Este sistema funciona mediante scripts de Python.

**Conexión Física:** Verificar que la Cámara, Sensores Ultrasónicos (pines Echo/Trigger), Servomotor (Pin 18) y Pantalla LCD (I2C) estén conectados correctamente a la Raspberry Pi.

**Encendido:** Conectar la Raspberry Pi a la corriente eléctrica.

**Ejecución del Backend:** Abrir una terminal y ejecutar el script de control: `sudo python parking_control.py` (Esto activa los sensores, la cámara y la lógica de la barrera).

**Ejecución del Frontend:** Abrir una segunda terminal y lanzar el dashboard: `streamlit run dashboard.py`

**Acceso:** Ingresar a la URL proporcionada (ej: <http://localhost:8501>) desde el navegador.

## Interfaz gráfica

El sistema cuenta con una interfaz web:

### Vista Clientes (Pantalla Pública)

Diseñada para ser mostrada en una pantalla grande a la entrada del recinto.

- Muestra dos tarjetas grandes con el número de **Lugares Libres** en el Sector A y Sector B.
- La pantalla se refresca automáticamente cada 2 segundos sin intervención del usuario, mostrando siempre la capacidad real.



## Vista Administrador

Área restringida para el operador del sistema.

- Muestra la ocupación actual y la capacidad total configurada.
- Visualiza la patente y hora del último vehículo detectado por la cámara.
- Botón para forzar la actualización de datos.
- En esta vista automáticamente se despliega una tabla donde muestra ID, patente, hora de entrada, hora de salida y tiempo de estancia.
- Permite ingresar un ID y borrar registros erróneos o antiguos de la base de datos.

[illegible]

# Funcionalidades Principales

## Reconocimiento de Patentes (LPR)

Cuando el sensor ultrasónico de entrada detecta un vehículo a menos de 15cm, el sistema activa la cámara, captura una imagen y procesa la patente. Si la lectura es válida, registra la entrada en la base de datos.

## Control de Acceso y Barrera

Una vez registrada la patente (o detectada la salida), el sistema envía una señal al **Servomotor** para levantar la barrera (90 grados). Tras 5 segundos, la barrera se cierra automáticamente (0 grados).

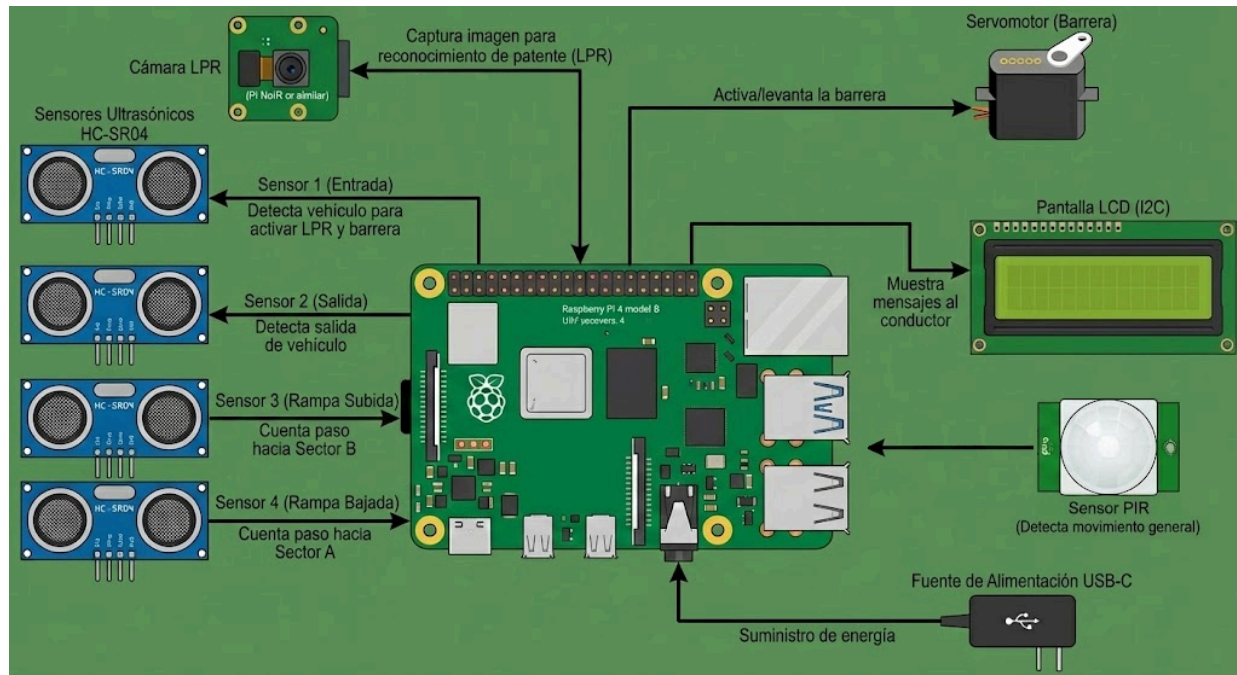
## Monitoreo de Capacidad (Sectores A y B)

El sistema gestiona lógica de transferencia:

- **Entrada Calle:** Suma al Sector A.
- **Rampa Subida:** Resta del Sector A y suma al Sector B.
- **Rampa Bajada:** Resta del Sector B y suma al Sector A.
- **Salida Calle:** Resta del Sector A.



# Implementación



## Sensores y Actuadores

- **Raspberry Pi 4:** Cerebro del sistema. Procesa imágenes y coordina los hilos de ejecución.
- **Cámara USB/PiCam:** Captura la imagen del vehículo para el proceso de LPR.
- **Sensores Ultrasónicos (4 unidades):**
  - *Sensor 1 & 2 (Entrada/Salida):* Activan la cámara y la barrera.
  - *Sensor 3 & 4 (Rampas):* Solo cuentan el paso de vehículos entre pisos para actualizar la disponibilidad.
- **Servomotor:** Ayuda a abrir y cerrar la barrera de acceso al estacionamiento.
- **Pantalla LCD (I2C):** Muestra los espacios disponibles.

## Soluciones a Problemas

### 1. ¿Por qué el servomotor vibra mucho?

- **Solución:** El código incluye una función "suave" que corta la señal eléctrica después de mover el motor. Si persiste, verifique que la fuente de poder de la Raspberry sea suficiente (mínimo 3A).

### 3. La cámara no detecta la patente.

- **Solución:** Asegúrese de que haya buena iluminación. El sistema LPR requiere contraste. Verifique que la patente esté dentro del ángulo de visión (aprox. 15-50 cm del sensor ultrasónico).

### 4. Los números en la pantalla LCD no coinciden con la TV.

- **Solución:** Verifique que las variables CAPACIDAD\_A y CAPACIDAD\_B sean iguales en los archivos parking\_control.py y dashboard.py. Reinicie ambos scripts.

### 5. El sistema dice "Error I2C".

- **Solución:** Revise los cables SDA y SCL de la pantalla LCD. Ejecute `i2cdetect -y 1` en la terminal para ver si la Raspberry detecta la dirección (usualmente 0x27 o 0x3e).

## Conclusión

EstacionaFlash proporciona una solución integral para la gestión inteligente de estacionamientos, combinando sensores, visión computacional y automatización. Su diseño modular permite escalar el sistema a grandes instalaciones, mejorando la experiencia del usuario, reduciendo tiempos de espera y optimizando recursos energéticos.

## Referencias

- [1] Arduino.cl, "Micro Servo Motor SG90 9G," *Arduino.cl*, [En línea]. Disponible en: <https://arduino.cl/producto/micro-servo-motor-sg90-9g/>.
- [2] MCI Electronics, "Cámara V2 para Raspberry Pi," *MCI Electronics*, [En línea]. Disponible: <https://mcielectronics.cl/shop/product/camara-v2-para-raspberry-pi-raspberry-pi-21481/>.
- [3] MCI Electronics, "Sensor de Proximidad de Ultrasonido HC-SR04," *MCI Electronics*, [En línea]. Disponible: <https://mcielectronics.cl/shop/product/sensor-de-proximidad-de-ultrasonido-hc-sr04-23582/>.
- [4] Mediatech, "Tienda de Electrónica y Robótica en Chile," *Mediatech*, [En línea]. Disponible: <https://mediatech.cl/>.
- [5] D. Aracena, "Material compartido por el profesor - Asignatura Proyecto II," *Google Drive*, Universidad de Tarapacá, 2025. [En línea]. Disponible: <https://drive.google.com/drive/u/1/folders/13I54lW9OofTCxDqKDl1iKrSNhdGpzlt0>.
- [6] Equipo EstacionaFlash, "Diagramas de casos de uso y clases," *Lucidchart*, 2025. [En línea]. Disponible: [https://lucid.app/lucidchart/edb1af02-4c03-4e51-a8c2-74a1f786ef9b/edit?beaconFlowId=C72FA5319B624A87&invitationId=inv\\_965c987c-aef3-48e8-a5ad-8e0d74eb2035&page=0\\_0#](https://lucid.app/lucidchart/edb1af02-4c03-4e51-a8c2-74a1f786ef9b/edit?beaconFlowId=C72FA5319B624A87&invitationId=inv_965c987c-aef3-48e8-a5ad-8e0d74eb2035&page=0_0#).
- [7] Equipo EstacionaFlash, "Diagramas de secuencias," *Lucidchart*, 2025. [En línea]. Disponible: [https://lucid.app/lucidchart/e0672d13-0039-4a33-90ca-a4c60d857f69/edit?invitationId=inv\\_acb7248a-15d4-4c13-a968-2797d384e685&page=0\\_0#](https://lucid.app/lucidchart/e0672d13-0039-4a33-90ca-a4c60d857f69/edit?invitationId=inv_acb7248a-15d4-4c13-a968-2797d384e685&page=0_0#).
- [8] Equipo EstacionaFlash, "Diseño de Interfaz de Usuario (Digital Signage)," *Figma*, 2025. [En línea]. Disponible: <https://www.figma.com/make/OzkXidTqwnWDRfjRvoqCyk/Digital-Signage-Interface-Design?node-id=0-1&p=f&t=6X7ESNJ22uxl3TRQ-0&fullscreen=>