

EstacionaFlash

Integrantes:
Rafael Nakata
Erik Correa
Álvaro Guarachi
Angel Cabezas

Profesor: Diego Aracena
Asignatura: Proyecto II



Índice

- 1. Introducción**
- 2. Problemática**
- 3. Solución propuesta**
- 4. Objetivos**
- 5. Requerimientos Funcionales**
- 6. Requerimientos No funcionales**
- 7. Descripción de Arquitectura**
- 8. Demostración**
- 9. Conclusión**

Introducción

EstacionaFlash, nace de una realidad que todos conocemos: la ineficiencia en los estacionamientos actuales. Los tickets físicos y las barreras manuales no solo generan congestión, sino que representan una tecnología obsoleta.

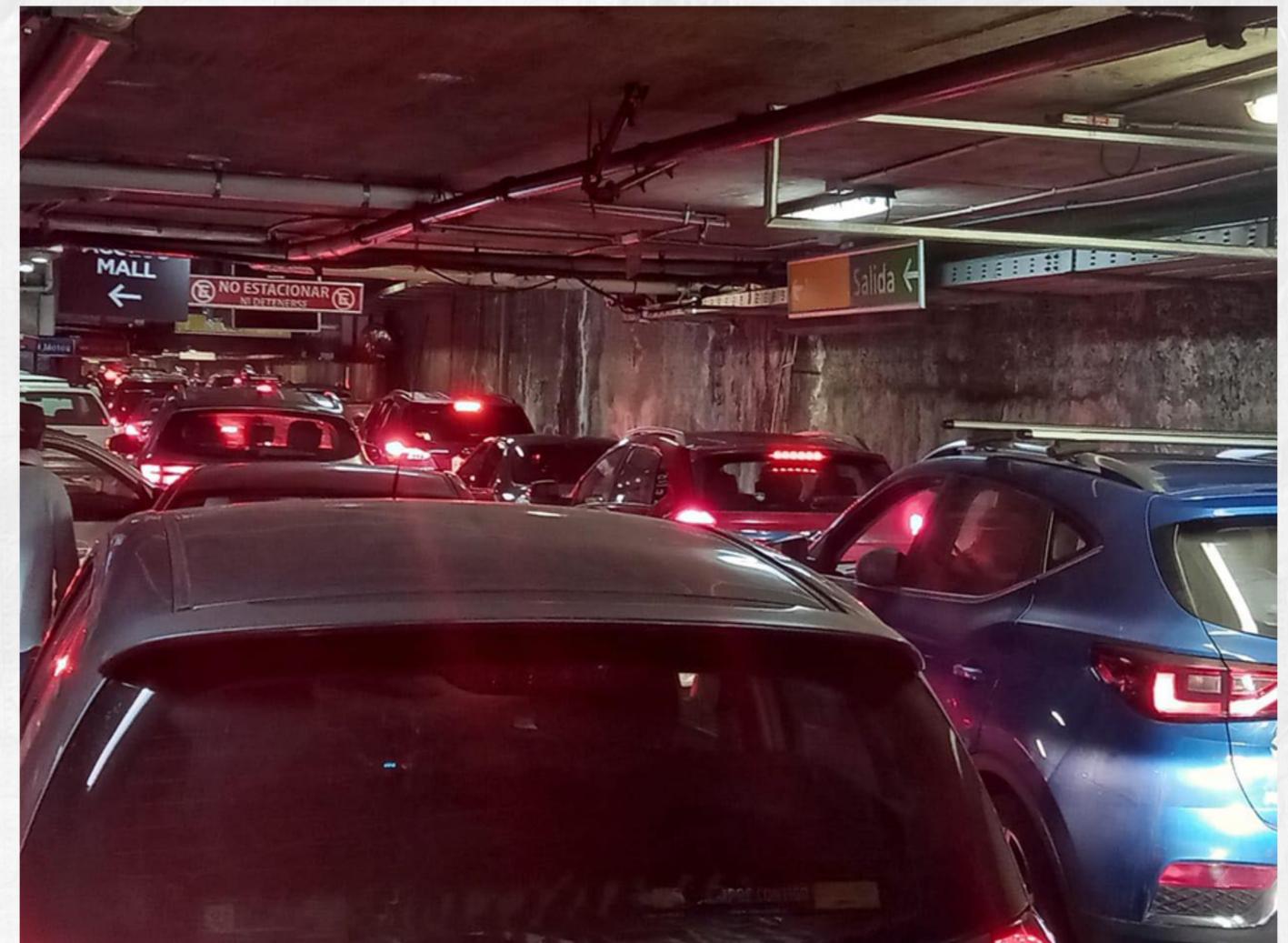
Para resolver esto, hemos desarrollado un sistema autónomo basado en Internet de las Cosas (IoT). Utilizando una Raspberry Pi 4B como nodo central, integramos Visión Artificial (LPR) y sensores ultrasónicos para gestionar el acceso y el flujo vehicular sin intervención humana.

Problemática

- Falta de control y monitoreo en estacionamientos comerciales.
- Congestión y dificultad para encontrar espacios disponibles.
- Procesos manuales ineficientes (tickets, cobros).
- Gasto energético en iluminación sin control.

Consecuencias

- Pérdida de tiempo y mala experiencia del usuario.
- Costos operativos elevados por iluminación ineficiente.
- Errores humanos en cobros o control de accesos.
- Baja satisfacción de administradores y clientes.



Solución Propuesta

Un sistema inteligente que automatice el estacionamiento:

- Detecta vehículos y muestra el numero de plazas libres en tiempo real.
- Controla accesos automáticamente.
- Muestra disponibilidad en pantallas LED.
- Optimiza la iluminación según movimiento.
- Reconoce patentes de autos.





Objetivos



General

Desarrollar un sistema de automatización basado en sensores con Raspberry Pi para solucionar la falta de gestión, control y monitoreo en tiempo real de los estacionamientos comerciales (malls, supermercados)



Específicos:

- **Diseñar** la arquitectura integral de hardware y software del sistema "EstacionaFlash", definiendo los diagramas de conexión y la lógica de comunicación entre el Raspberry Pi, los sensores y periféricos para garantizar el flujo de información en tiempo real.
- **Implementar** los algoritmos de visión artificial y lógica de control en el entorno de la Raspberry Pi, desarrollando el reconocimiento de patentes (LPR) y la gestión de señales de los sensores ultrasónicos para la automatización del acceso.

- **Integrar** los subsistemas físicos (cámaras, sensores ultrasónicos y servomotores) con la interfaz de software, sincronizando la detección vehicular con la respuesta de los actuadores (barreras) para lograr una operación autónoma.
- **Validar** el desempeño técnico del prototipo mediante pruebas funcionales en escenarios simulados, verificando el cumplimiento de métricas críticas como la latencia de respuesta y la precisión en la lectura de patentes.
- **Garantizar** la mantenibilidad y operabilidad del sistema mediante la generación de manuales técnicos y de despliegue estandarizados, asegurando la escalabilidad futura del proyecto.

Requerimientos

Funcionales

1. Detección de Vehículo: El sistema debe detectar la presencia de un vehículo mediante sensores ultrasónicos en las entradas y salidas.

2. Captura y reconocimiento de patente: El sistema debe capturar la imagen de la patente utilizando cámaras y procesarla para obtener su texto.

3. Cálculo y Validación de Tarifa: El sistema debe calcular el tiempo de estadía basado en la hora de entrada y verificar si la patente del vehículo ha cubierto el pago correspondiente antes de autorizar la salida.



Requerimientos

Funcionales

4. Control automático de barrera: El sistema debe enviar una señal al motor para abrir o cerrar la barrera solo si la validación fue exitosa.

6. Detectar movimiento: El sistema debe detectar movimientos mediante sensores PIR para encender o apagar las luces de los sectores.

5. Actualización de disponibilidad: El sistema debe actualizar la información en el panel LED indicando disponibilidad del sector del estacionamiento o estado del acceso.

7. Gestión de Estados Lógicos: El sistema debe validar la consistencia lógica durante el proceso de egreso, impidiendo el registro de una salida (y la apertura de barrera) a cualquier vehículo que no figure con el estado 'Dentro' o que carezca de un registro de entrada activo en la base de datos.

Requerimientos No Funcionales

1. Tiempo de respuesta: El sistema debe responder en tiempo real, con una latencia máxima de 2-3 segundos para la validación de la patente y la apertura de la barrera.

2. Escalabilidad: El sistema debe permitir agregar más sensores o cámaras sin alterar su funcionamiento central.

3. Seguridad: Los datos como registros de patentes y accesos deben mantenerse protegidos y no ser accesibles por usuarios no autorizados.



Requerimientos No Funcionales

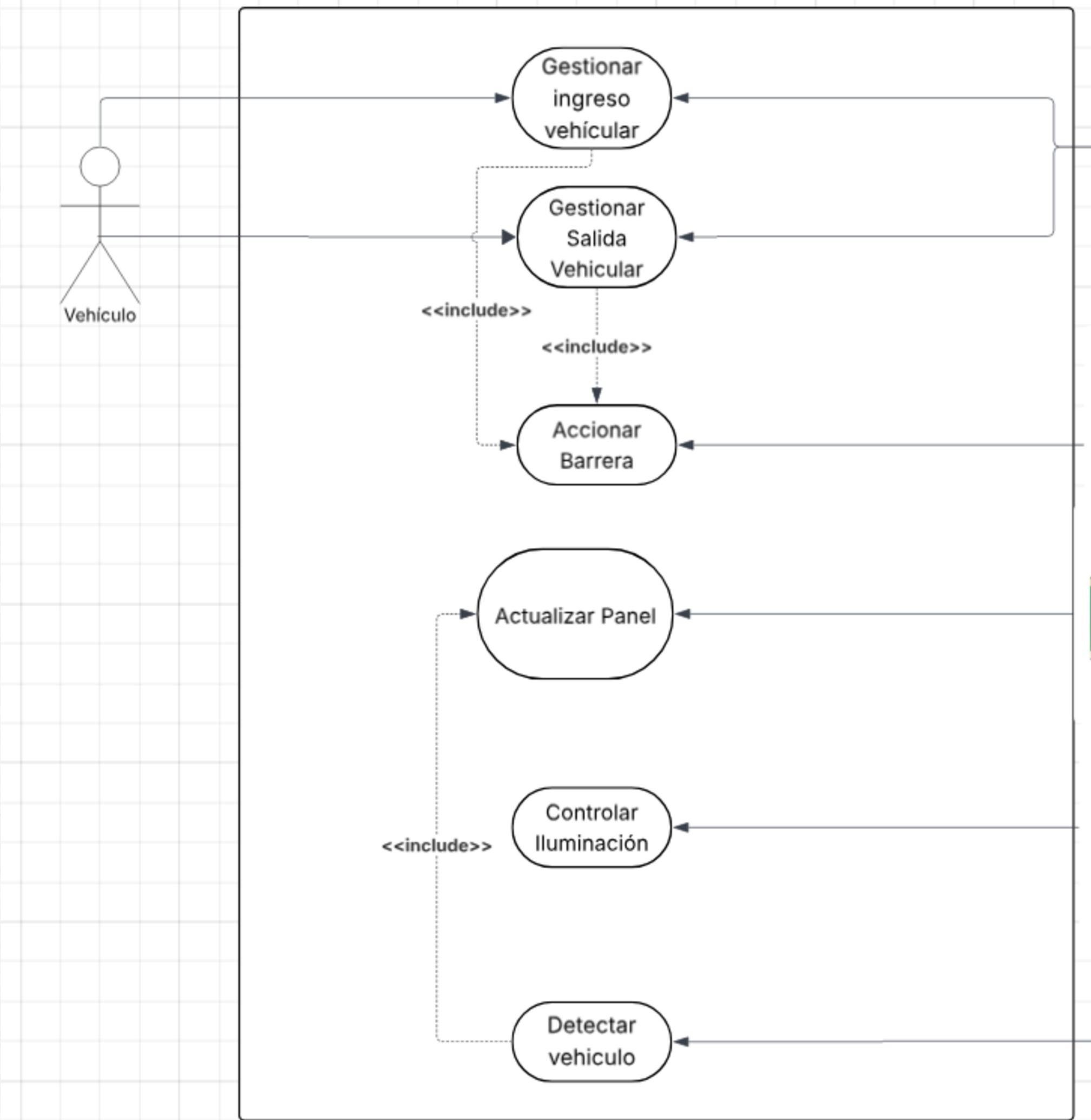
4. Eficiencia energética: El uso de iluminación automatizada debe reducir el consumo energético cuando no se detecten vehículos.

5. Mantenibilidad: El sistema debe estar documentado para facilitar su actualización o reparación.

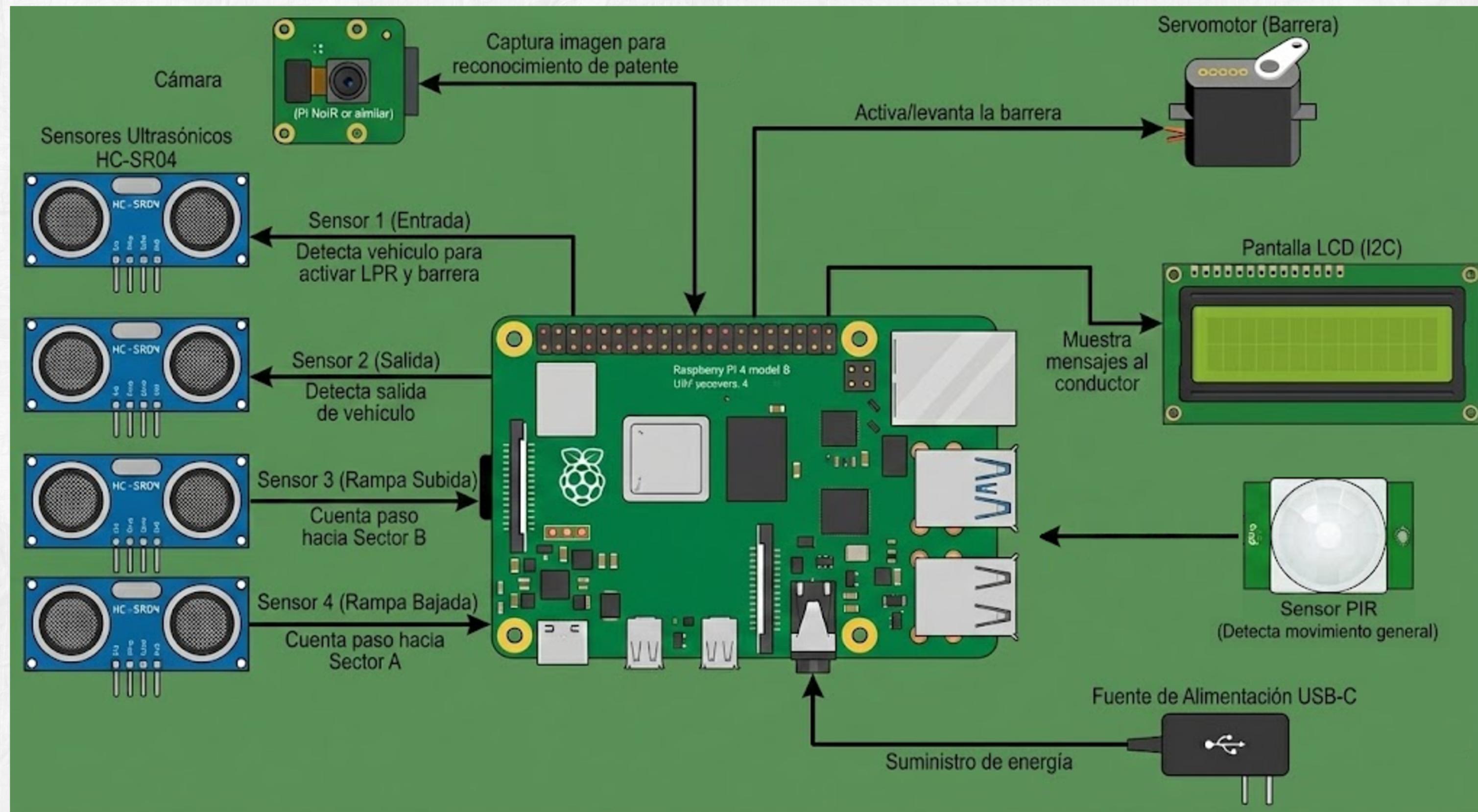
6. Fiabilidad: La detección y reconocimiento de patentes tendrá una precisión mínima del 90% en condiciones normales de iluminación, para garantizar un funcionamiento confiable.



Diagrama de casos de uso



Descripción de la arquitectura





Demostración de prototipo



Conclusión

Validación Técnica: El prototipo demostró una integración exitosa entre Hardware (Sensores/Actuadores) y Software (LPR/Python).

Rendimiento en Tiempo Real: Se cumplieron los objetivos de latencia, logrando validación de acceso y apertura de barreras en segundos.

Autonomía Total: El sistema opera independientemente, gestionando flujo vehicular e iluminación sin intervención manual.

Escalabilidad: Arquitectura IoT probada y lista para evolucionar hacia una implementación comercial industrial.



Gracias.