

# EstacionaFlash

Integrantes: Rafael Nakata  
Erik Correa  
Álvaro Guarachi  
Angel Cabezas

Profesor: Diego Aracena  
Asignatura: Proyecto II



# Índice

- 1. Introducción**
- 2. Requerimientos**
- 3. Diagrama de Casos de Uso**
- 4. Diagrama de Clases**
- 5. Diagrama de Secuencia**
- 6. Descripción de Arquitectura**
- 7. Diseño de Interfaz**
- 8. Conclusión**

# Introducción

La gestión de estacionamientos actual es ineficiente: depende de tickets físicos y control manual.

Nuestra solución, 'EstacionaFlash', automatiza todo el proceso usando una Raspberry Pi 4B como unidad central. Se busca eliminar el ticket de papel reemplazándolo por reconocimiento de patentes y monitorear la disponibilidad con sensores ultrasónicos.

En esta presentación consolidaremos la Ingeniería del Proyecto antes de construir.

Por lo que se presentará:

## **Descripción de la problemática y la respectiva solución.**

**Requisitos:** Definimos exactamente qué hará el sistema (RF) y cómo debe rendir (RNF).

**Modelados lógicos:** Se mostrarán diagramas UML del software (Casos de Uso, Clases y Secuencia).

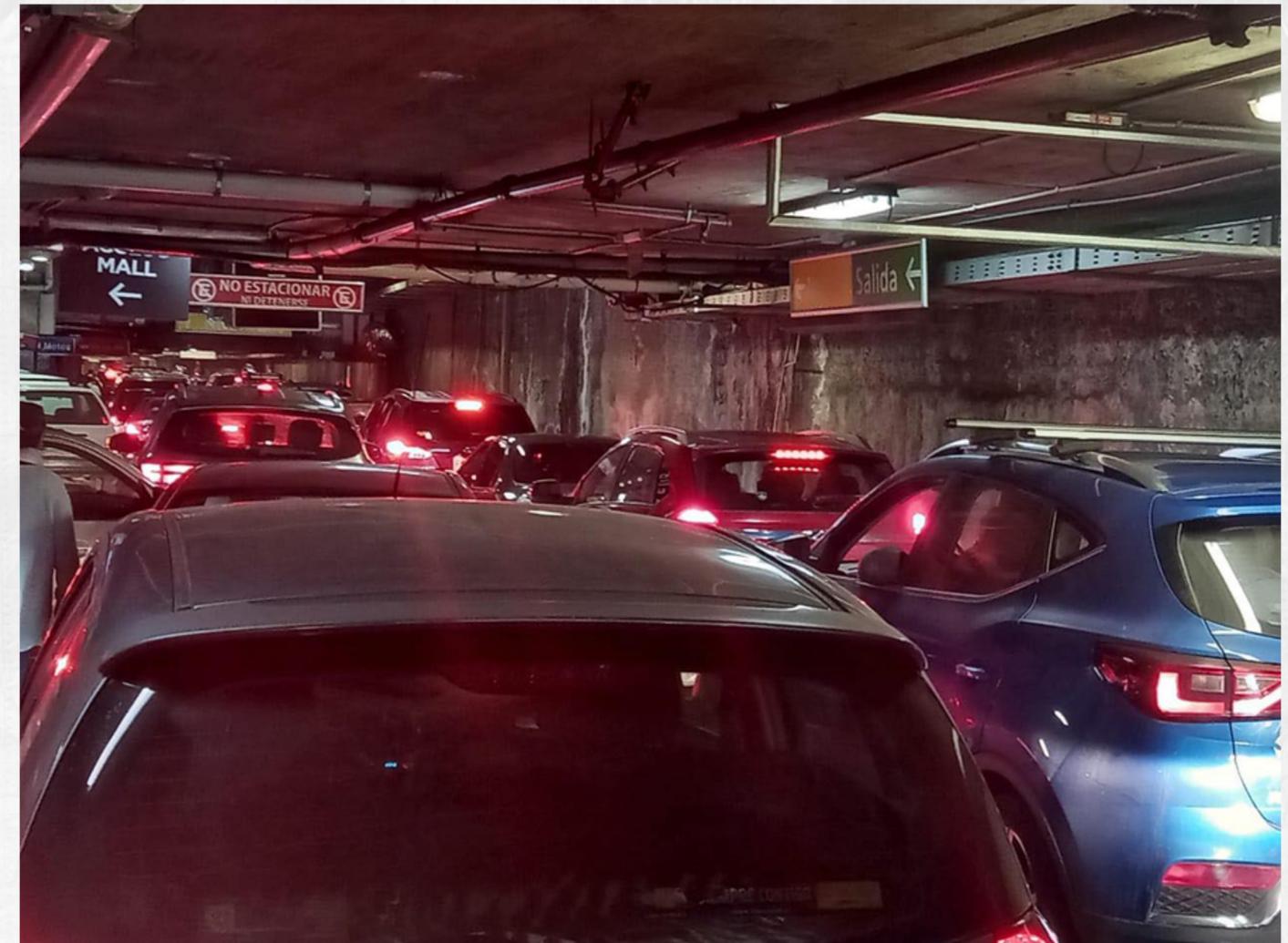
**Arquitectura:** Explicamos el esquema Cliente-Servidor donde la Raspberry Pi orquesta todos los sensores y así también la interfaz de usuario.

# 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

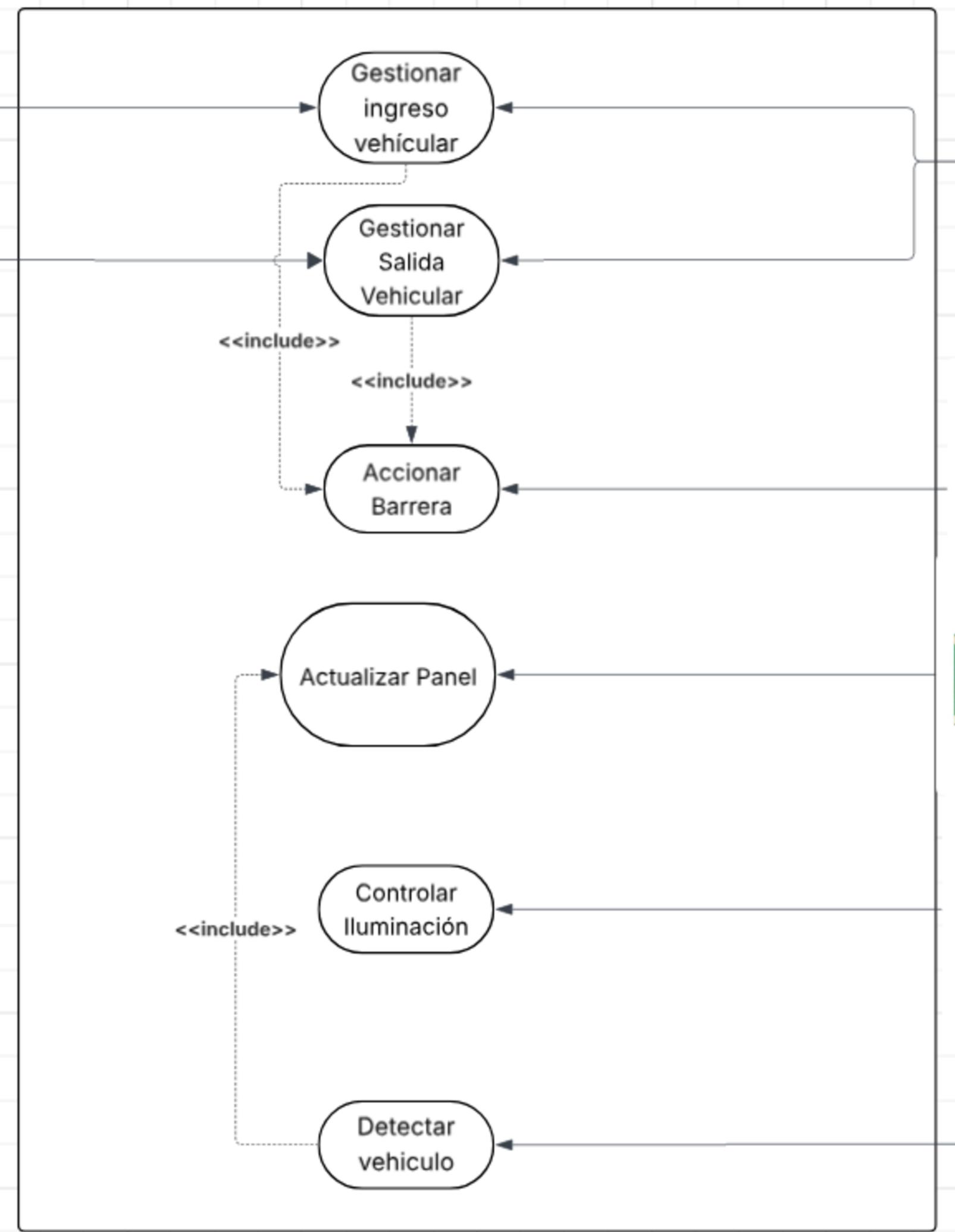
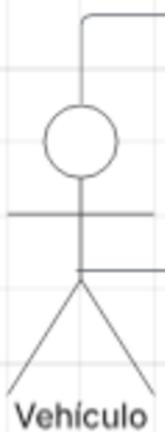
**4. Eficiencia energética:** El uso de iluminación automatizada debe reducir el consumo energético cuando no se detecten vehículos.

**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.

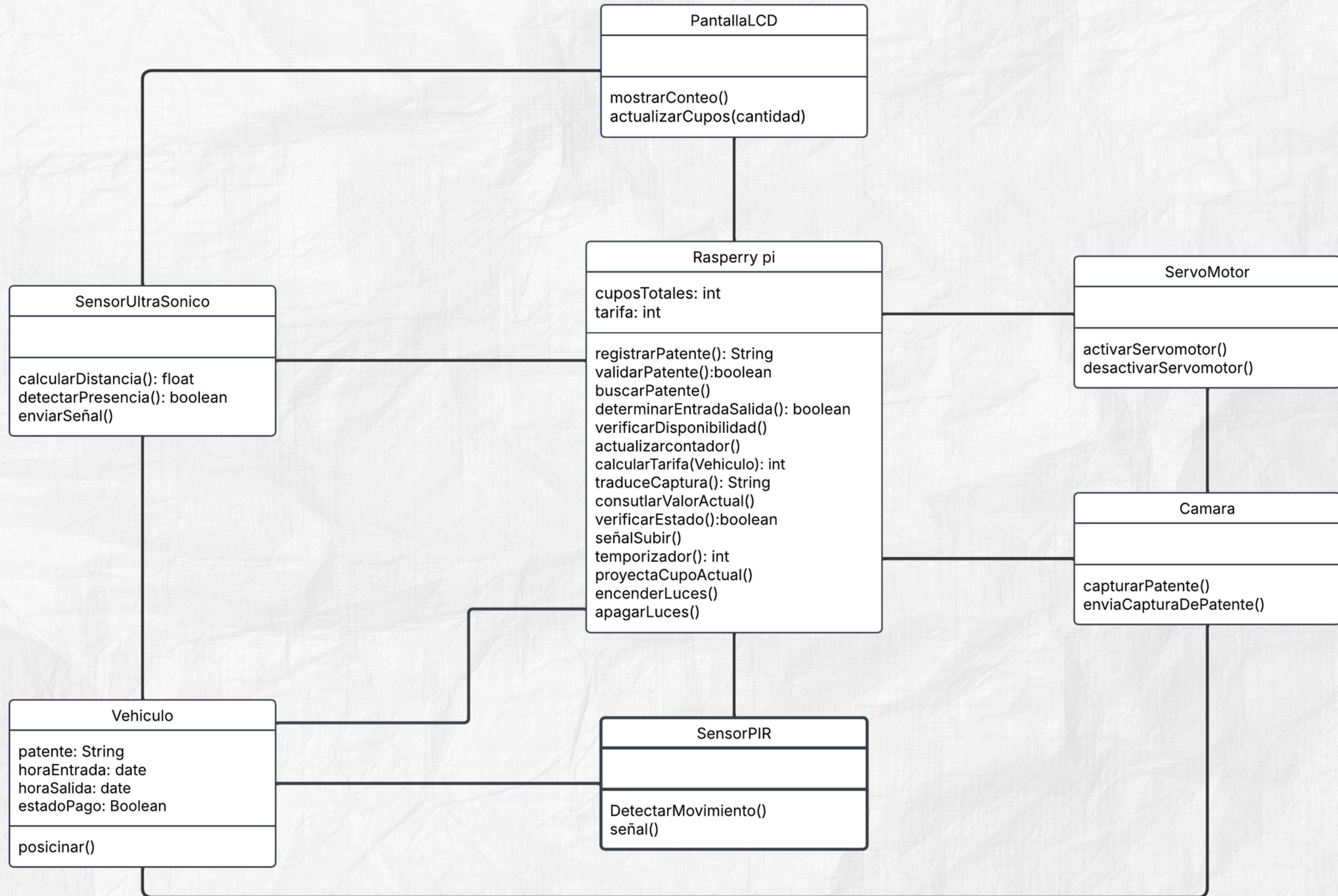


# Diagrama de casos de uso





# Diagrama de clases





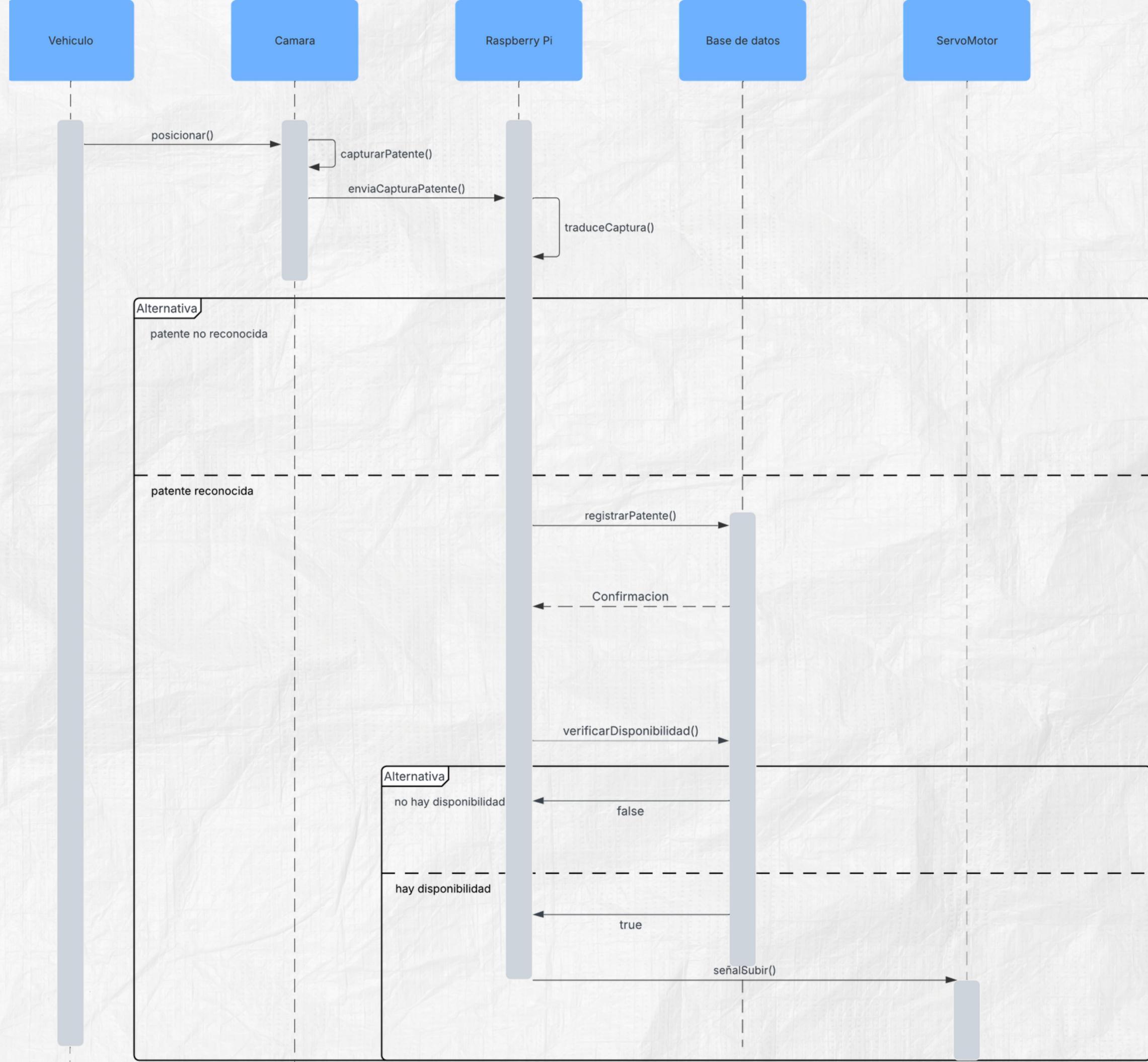
# Diagrama de Secuencia y Casos de Uso



<b>ID: CU-002</b>	
<b>Nombre del CU: Gestionar Ingreso Vehicular</b>	
<b>Actor(es):</b> Vehículo, Cámara, Servomotor	
<b>Requisitos Relacionados:</b> RF2, RF4	
<b>Descripción:</b> Gestiona el proceso completo de entrada: detecta el vehículo, lee la patente, valida disponibilidad, registra la entrada y abre la barrera.	
<b>Precondiciones:</b> El estacionamiento tiene plazas disponibles y la cámara está operativa.	
<b>Actor(Vehículo/Cámara)</b>	<b>Sistema</b>
<b>Flujo Principal:</b> 1. El conductor posiciona el vehículo frente a la barrera. 2. La cámara detecta presencia y captura imagen.	<b>Flujo Principal:</b>  3. Procesa la imagen (LPR) para obtener la cadena de texto de la patente. 4. Verifica disponibilidad de estacionamiento $> 0$ . 5. Registra la patente y hora de entrada en BD. 6. <<include>> <b>CU-004</b> (Accionar Barrera - Abrir).
<b>Flujo Alternativo:</b>	<b>Flujo Alternativo:</b> 3.1. No se reconoce la patente. El sistema no acciona la barrera y alerta del error. 4.1. No se permite el ingreso hasta que haya disponibilidad en el estacionamiento.
<b>Postcondiciones:</b> Se crea un registro de sesión abierto para el vehículo.	
<b>CU Relacionados :</b> CU-004 (Accionar barrera).	

# CU 002:

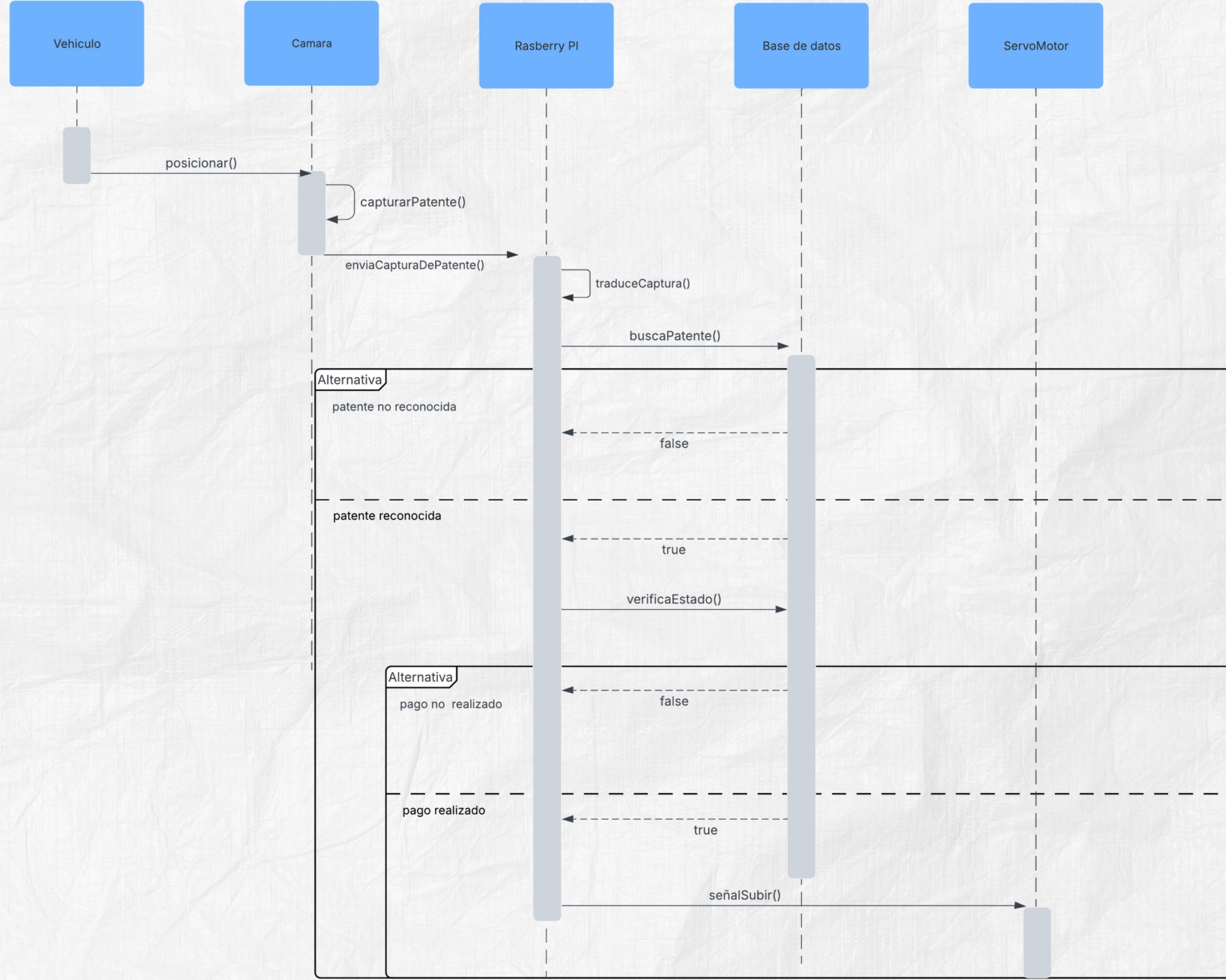
## Gestionar ingreso Vehicular





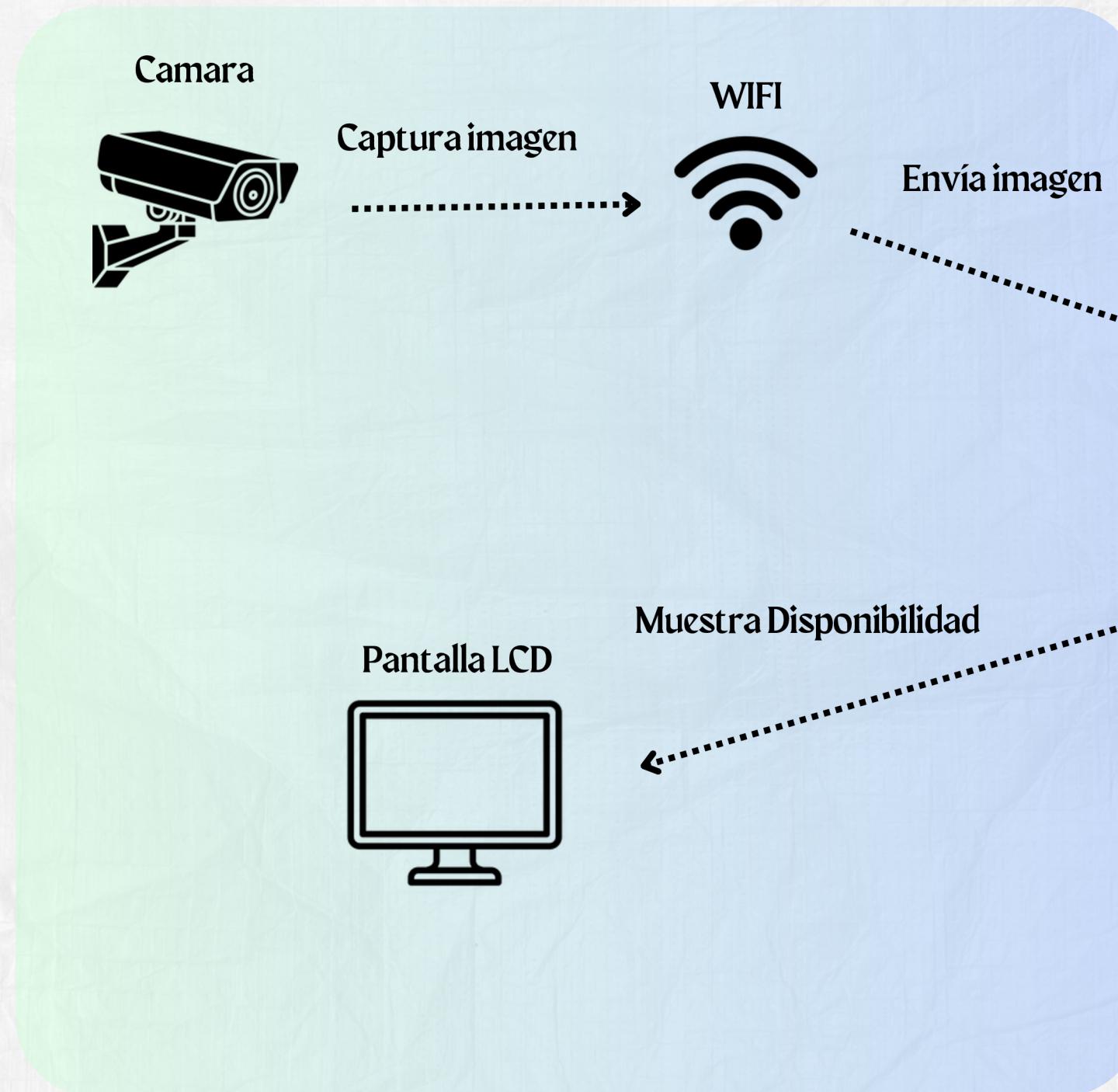
ID: CU-003	
<b>Nombre del CU:</b> Gestionar Salida Vehicular	
<b>Actor(es):</b> Vehiculo, Cámara, Servomotor	
<b>Requisitos Relacionados:</b> RF2, RF3, RF4, RF7	
<b>Descripción:</b> El sistema verifica que la patente capturada tenga su deuda saldada antes de liberar el paso.	
<b>Precondiciones:</b> La patente del vehículo debe estar en la base de datos del sistema.	
Actor(Cámara)	Sistema
<b>Flujo Principal:</b> 1. El conductor posiciona el vehículo frente a la barrera. 2. Captura imagen del vehículo en zona de salida.	<b>Flujo Principal:</b>  3. Procesa imagen (LPR) y obtiene patente. 4. Busca el registro de entrada activo para esa patente. 5. Calcula el tiempo total y verifica el estado del pago en BD. 6. Si estado == "PAGADO", cierra el ticket en la BD. 7. <<include>> CU-004 (Accionar Barrera - Abrir).
<b>Flujo Alternativo:</b>	<b>Flujo Alternativo:</b>  4.1 Vehículo no tiene entrada registrada ( posible intrusión). Alerta seguridad.  5.1 El sistema detecta estado "IMPAGO". Muestra monto a pagar en pantalla y mantiene barrera cerrada.
<b>Postcondiciones:</b> El vehículo es autorizado a salir y se cierra su ciclo de facturación.	
<b>CU Relacionados :</b> CU-004 (Accionar barrera).	

# CU 003: Gestionar Salida Vehicular

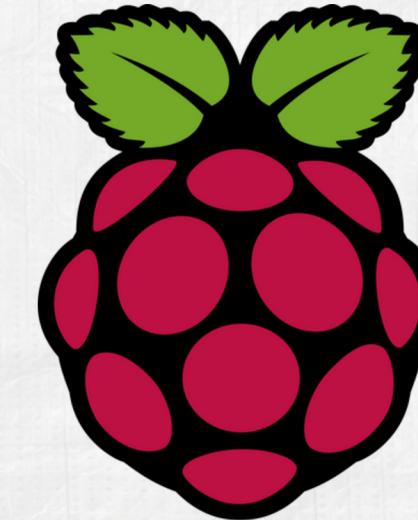


# Descripción de la arquitectura

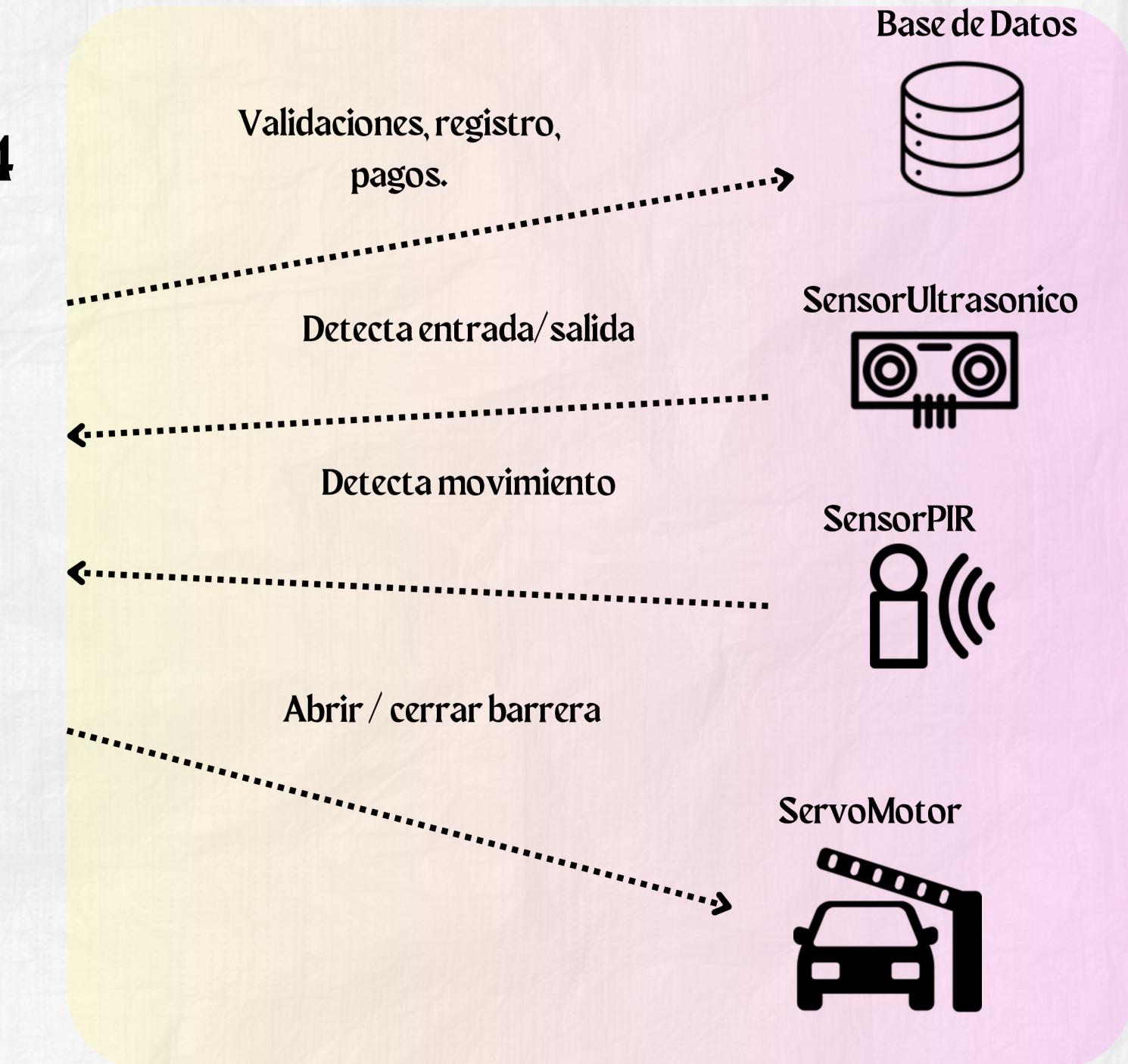
## Cliente



## Raspberry pi 4



## Servidor





# Diseño interfaz de usuario

EstacionaFlash

Total: 38/95

00:08:15

Martes, 25 De Noviembre De 2025

Mall Plaza Arica  
Av. Diego Portales

Sector A

24



24 ESPACIOS  
DISPONIBLES

Sector B

14



14 ESPACIOS  
DISPONIBLES

Gerenciar cookies ou desativar



# Conclusión

**Arquitectura Robusta:** Confirmamos que la Raspberry Pi funciona como un controlador independiente. El sistema sigue controlando barreras y pagos aunque falle la pantalla, lo que garantiza seguridad.

**Hoja de Ruta Clara:** Los diagramas UML generados eliminan las dudas. Tenemos una guía exacta para programar y además de la base de datos.

Nuestros diseños es prescriptivo ya funciona como un plano de construcción real.

**Resultado:** Entramos a la fase de implementación sin improvisar. Esto reduce drásticamente los errores de código y el tiempo perdido arreglando fallos.



# Referencias

- 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-aeef3-48e8-a5ad-8e0d74eb2035&page=0\\_0#](https://lucid.app/lucidchart/edb1af02-4c03-4e51-a8c2-74a1f786ef9b/edit?beaconFlowId=C72FA5319B624A87&invitationId=inv_965c987c-aeef3-48e8-a5ad-8e0d74eb2035&page=0_0#).
- 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#).
- 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=1>



Gracias.