

PROYECTO II

# SEMÁFORO INTELIGENTE PARA PEATONES



INTEGRANTES:

TIARA CANEPA

TOMÁS CARVAJAL

FERNANDO GARRIDO



UNIVERSIDAD DE TARAPACÁ<sup>®</sup>  
*Universidad del Estado*

# RESUMEN DE CONTENIDOS

RESUMEN

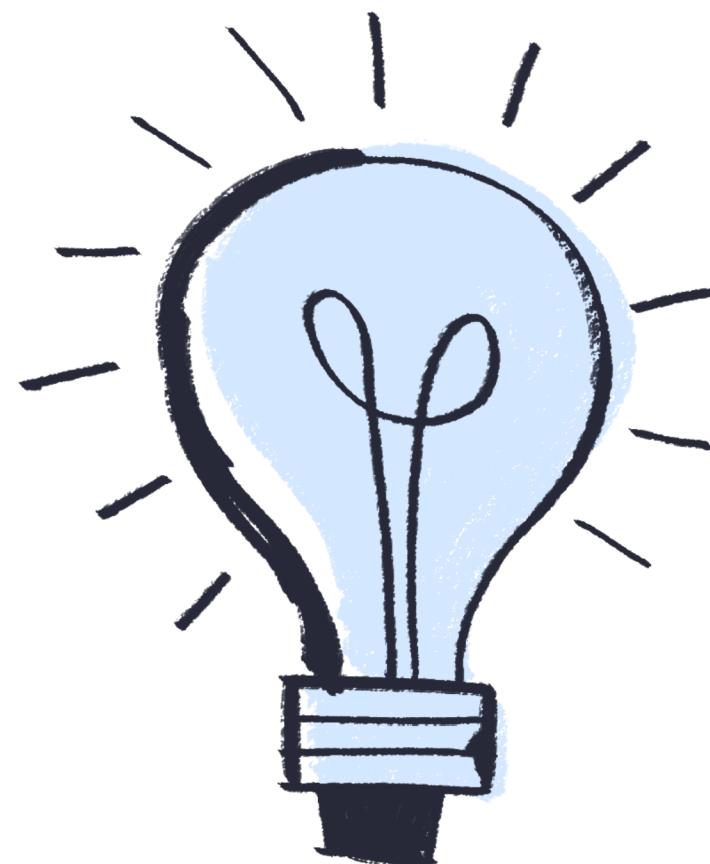
CUS

DIAGRAMAS

INTERFAZ

CONCLUSIÓN

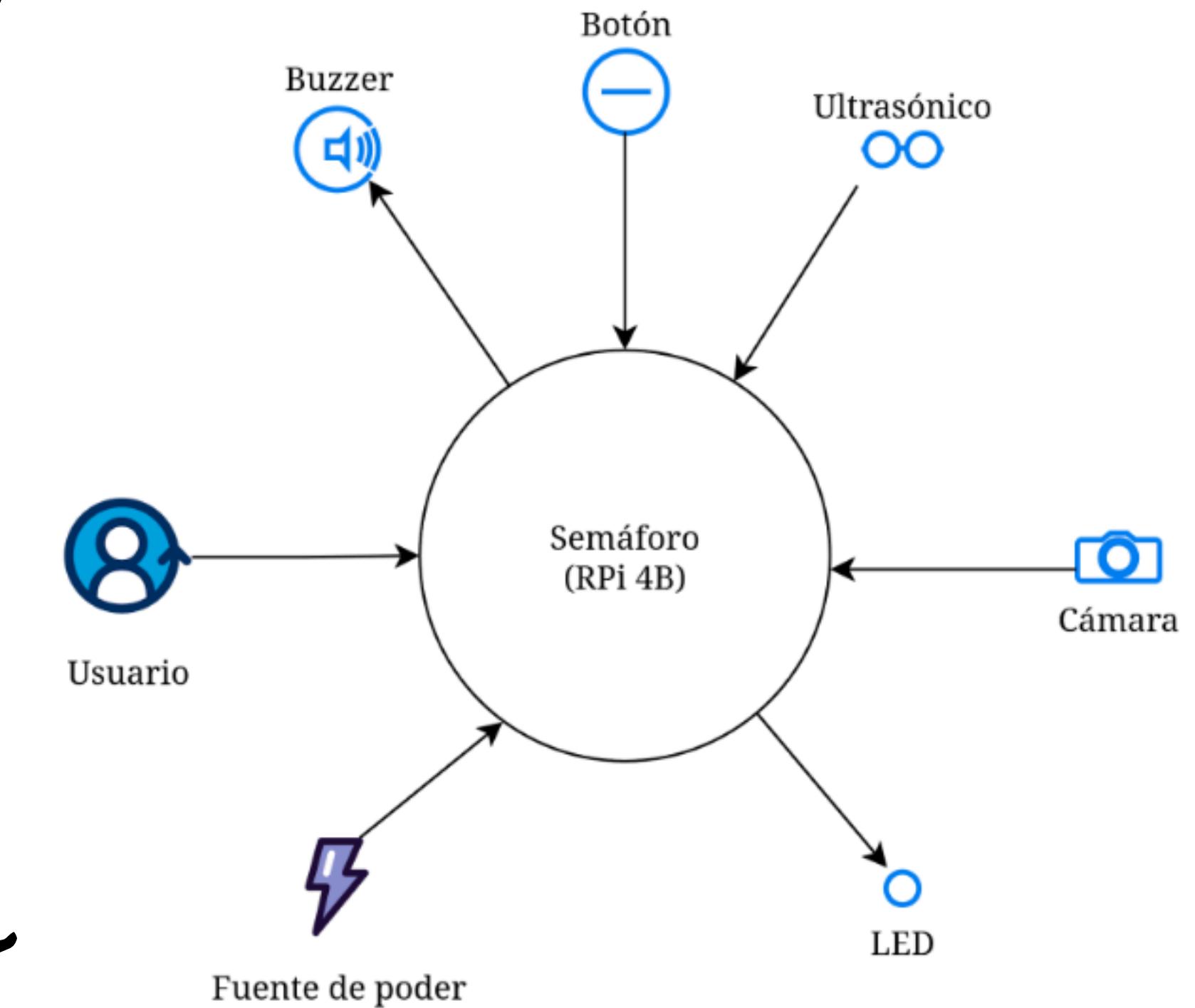
# INTRODUCCIÓN



En esta presentación se aborda la planificación técnica del Semáforo Inteligente para Peatones, comenzando con los modelos y procesos que estructuran el sistema. Se revisarán los requisitos funcionales y no funcionales, los casos de uso y la arquitectura que define el funcionamiento del proyecto. También se presentarán las herramientas, técnicas aplicadas y la interfaz de interacción del usuario. Finalmente, se mostrará una visión general del avance y conclusiones obtenidas durante el desarrollo.

# DISEÑO DE LA ARQUITECTURA

EL SISTEMA FUNCIONA MEDIANTE UN CONTROLADOR CENTRAL RASPBERRY PI 4B. LOS SENSORES BOTÓN, ULTRASÓNICO Y CÁMARA ENTREGAN DATOS SOBRE SOLICITUDES PEATONALES, DISTANCIA DE VEHÍCULOS Y PRESENCIA DE PERSONAS. A PARTIR DE ESTA INFORMACIÓN, EL RASPBERRY ACTIVA LOS LEDS Y EL BUZZER, MOSTRANDO EL ESTADO DEL SEMÁFORO Y ENTREGANDO APOYO SONORO PARA USUARIOS CON DISCAPACIDAD VISUAL.



# REQUISITOS FUNCIONALES

## **RF1 – Gestión del estado del semáforo**

El sistema debe determinar y mantener actualizado el estado del semáforo (rojo, amarillo o verde).

## **RF2 – Lectura de sensores de tráfico**

El sistema debe recibir e interpretar datos del sensor ultrasónico o cámara para determinar si hay vehículos presentes.

## **RF3 – Detección de vehículo cercano**

El sistema debe identificar cuando un vehículo se encuentra dentro del rango crítico para la toma de decisiones.

## **RF4 – Procesamiento de solicitudes peatonales**

El sistema debe recibir y registrar la solicitud de cruce cuando el peatón presiona el botón.

## **RF5 – Activación del sonido para peatones**

El sistema debe activar un buzzer o señal acústica cuando el cruce peatonal esté habilitado.

# REQUISITOS FUNCIONALES

## RF6 – Modulación del sonido durante el cruce

El sistema debe generar un patrón acústico (como sonido pulsante) durante los últimos segundos del tiempo peatonal.

## RF7 – Control de LEDs del semáforo

El sistema debe encender y apagar los LEDs correspondientes al estado del semáforo.

## RF8 – Ejecución de cambios de estado

El sistema debe ejecutar la transición entre luces (de rojo a verde, verde a amarillo, etc.) siguiendo la lógica establecida.

## RF9 – Validación de sensores

El sistema debe verificar que los sensores estén operativos antes de tomar decisiones.

## RF10 – Registro interno de eventos

El sistema debe registrar eventos como detección de vehículos, solicitudes peatonales y cambios de estado

# REQUISITOS no FUNCIONALES

## RNF1 – Tiempo de respuesta

El sistema debe procesar lecturas y actualizar el semáforo en un tiempo inferior a 1 segundo

## RNF2 – Confiabilidad en detección

Los sensores deben operar con un nivel de precisión mínimo del 90% en detección vehicular y activación del botón peatonal

## RNF3 – Disponibilidad operacional

El sistema debe funcionar continuamente, con una disponibilidad mínima del 95%.

## RNF4 – Seguridad peatonal

El sistema debe garantizar que el sonido y las luces representen correctamente el estado del semáforo para evitar confusiones.

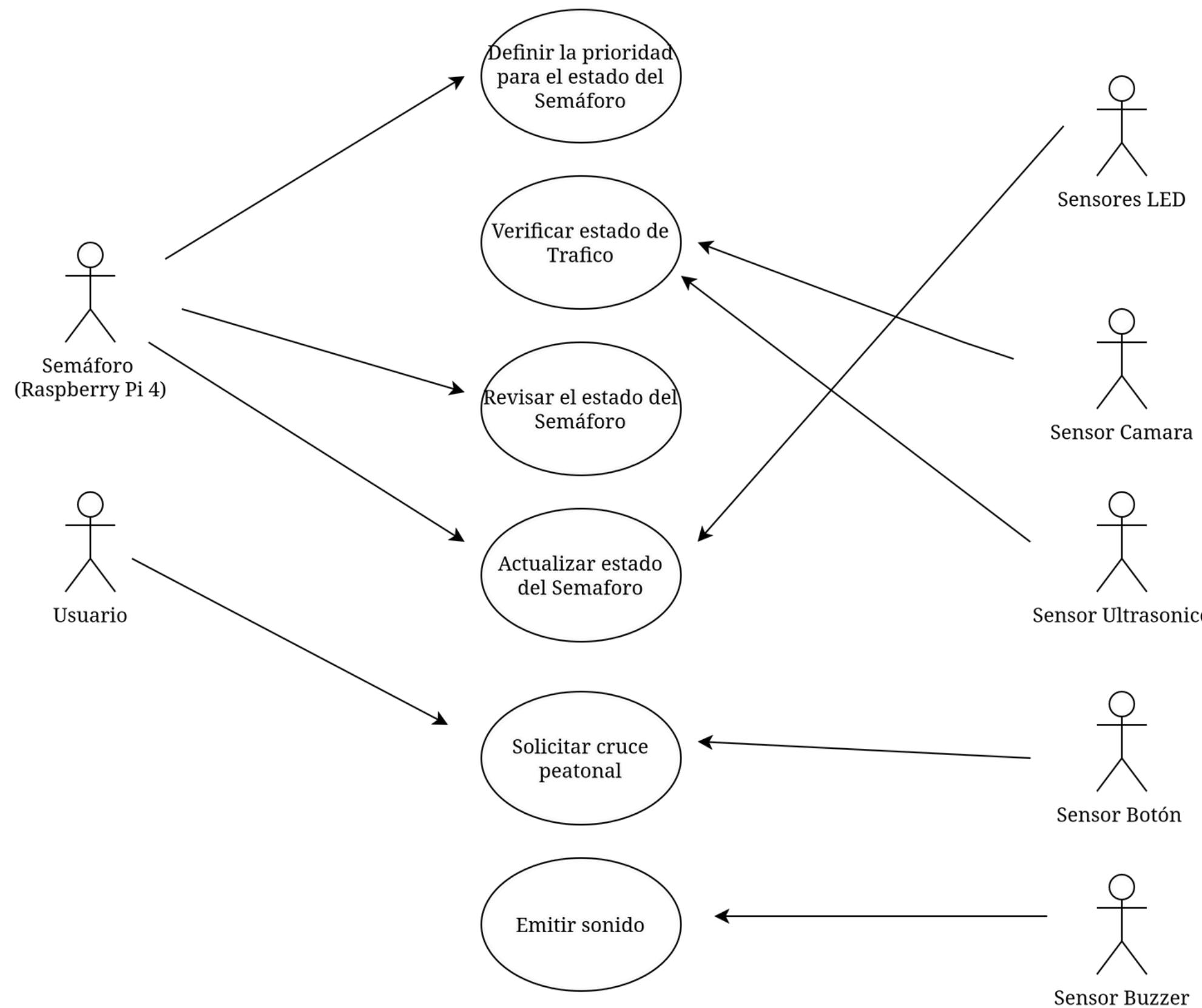
## RNF5 – Robustez del hardware

El sistema debe tolerar vibraciones, cambios de iluminación ambiental y ruido electromagnético sin fallos críticos.

## RNF6 – Mantenibilidad del software

El código debe ser modular y documentado para permitir modificaciones o sustitución de sensores sin reescritura total.

# DIAGRAMA DE CASOS DE USO



# DISEÑO DE LA ARQUITECTURA

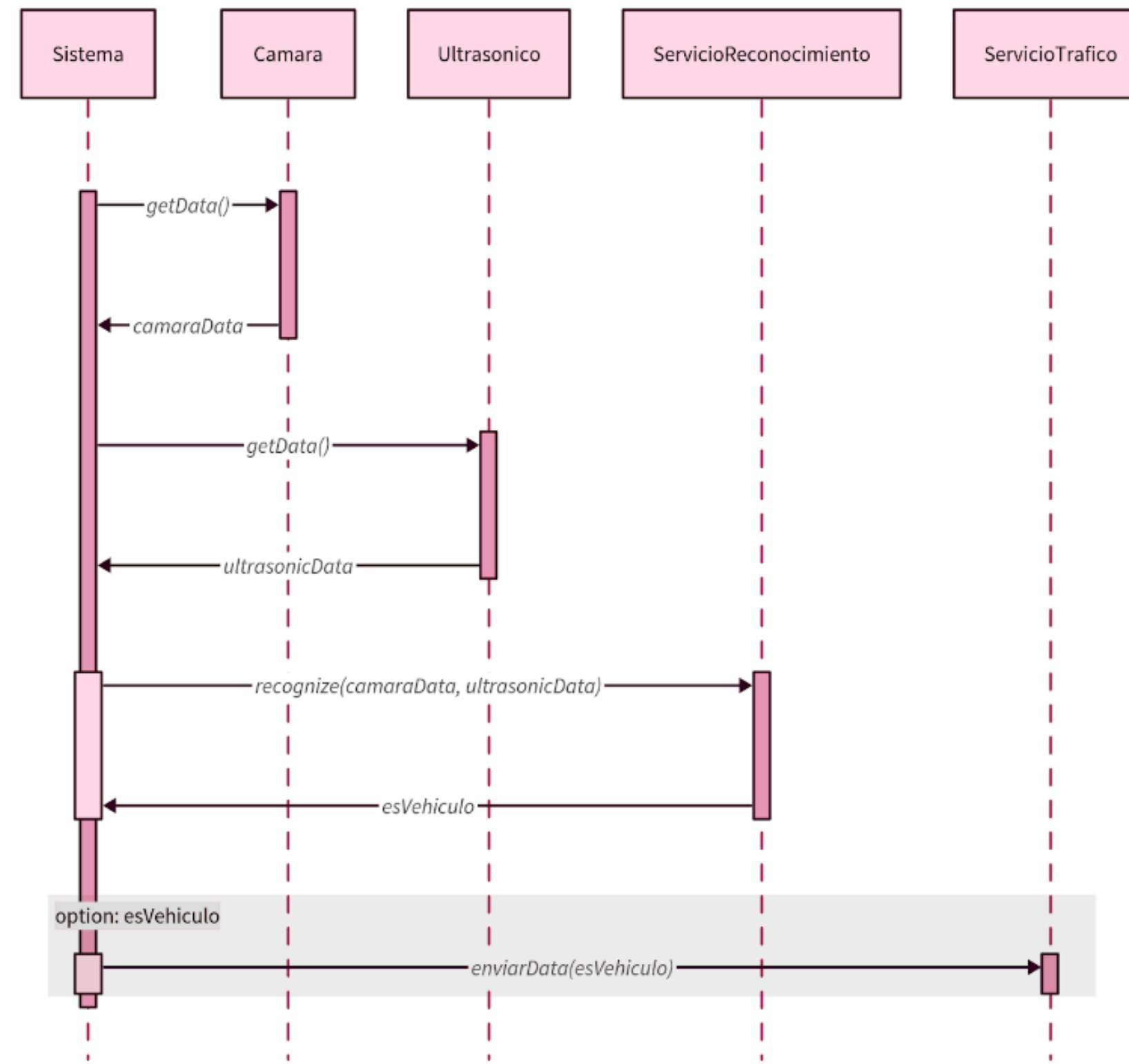


DIAGRAMA DE SECUENCIA PARA  
EL RECONOCIMIENTO DE AUTOS

# DISEÑO DE LA ARQUITECTURA

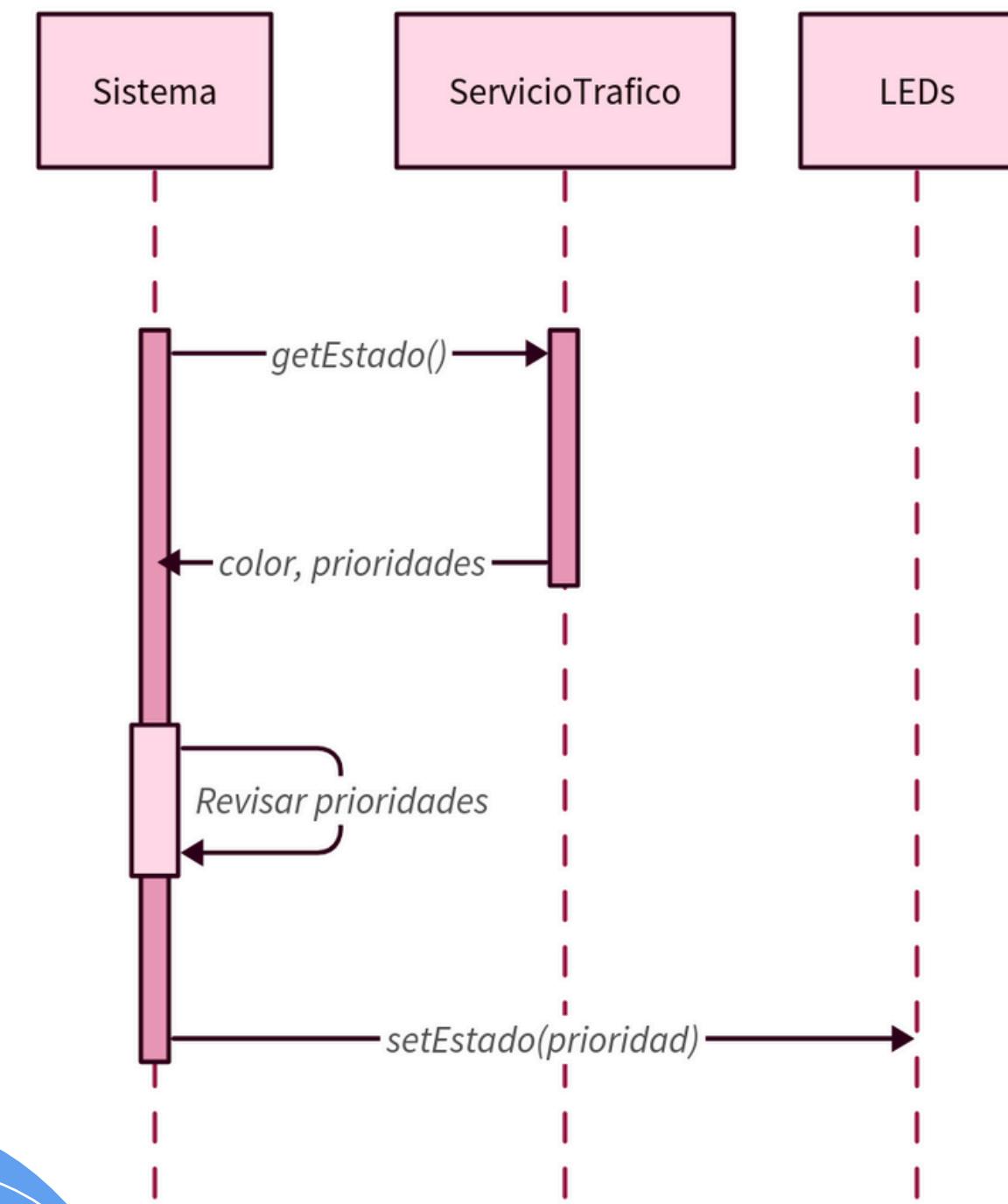


DIAGRAMA DE SECUENCIA PARA  
EL CAMBIO DE LEDS

# DISEÑO DE LA ARQUITECTURA

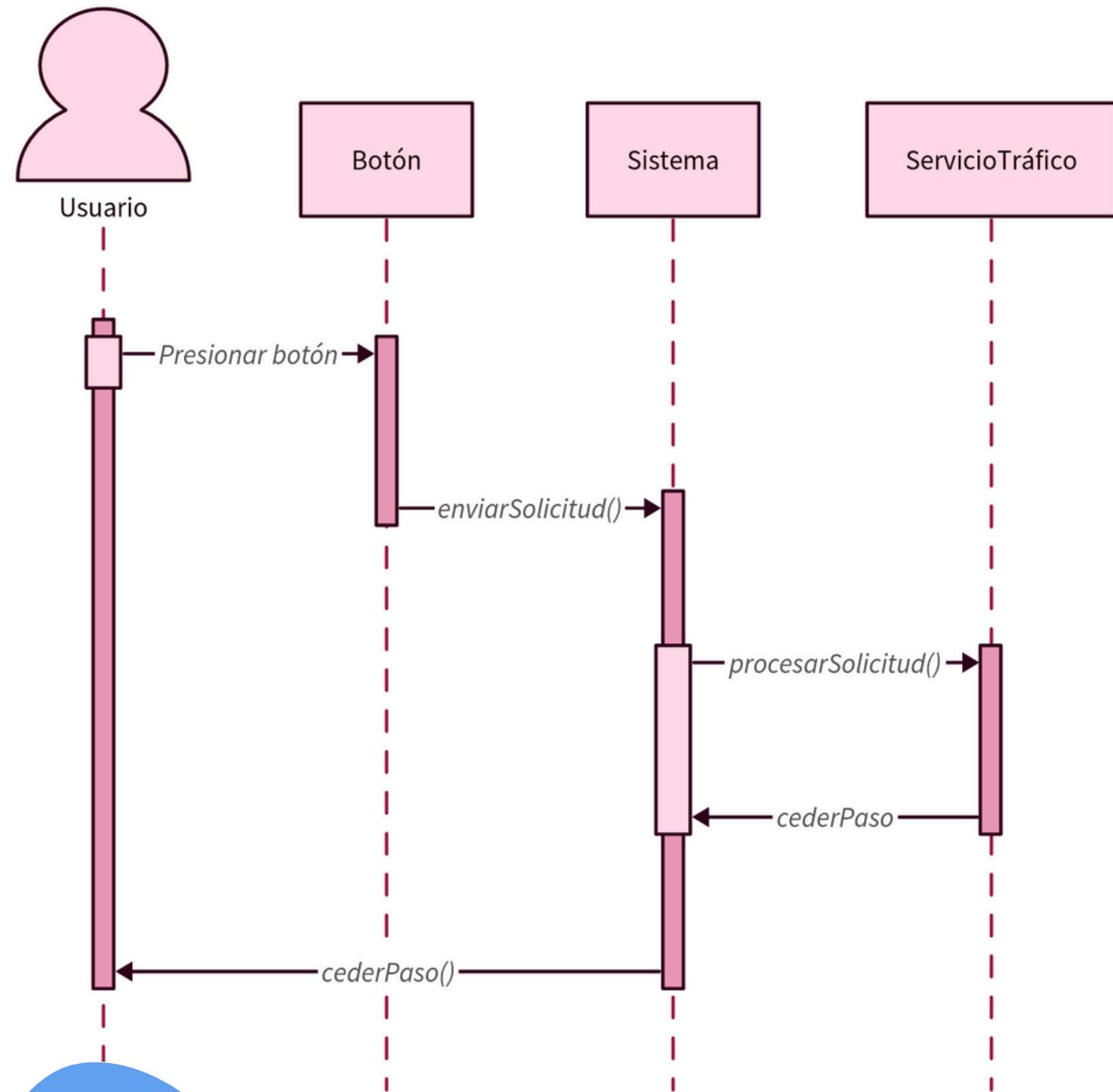


DIAGRAMA DE SECUENCIA PARA  
SOLICITAR EL CRUCE PEATONAL

# DISEÑO DE LA ARQUITECTURA

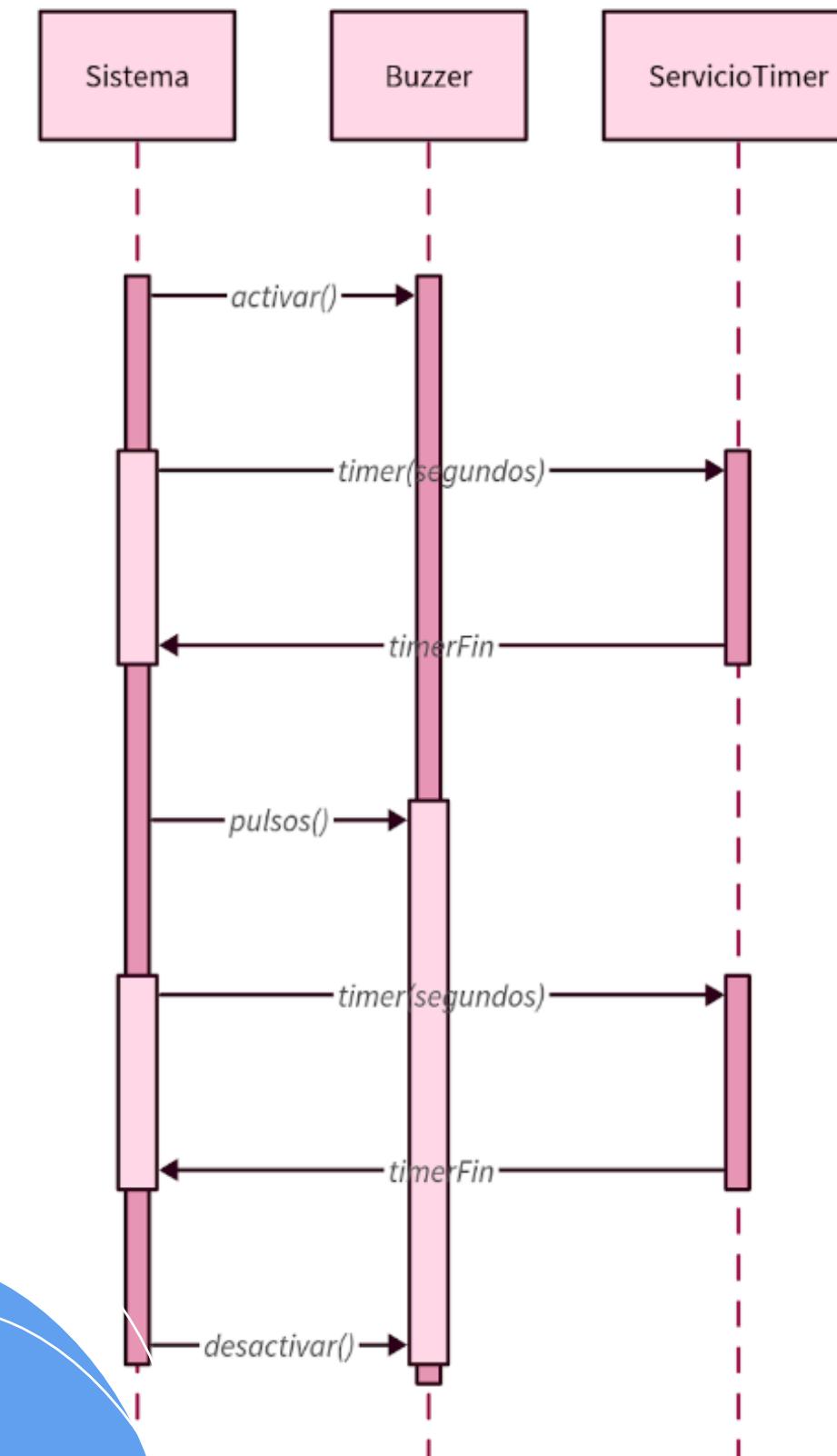
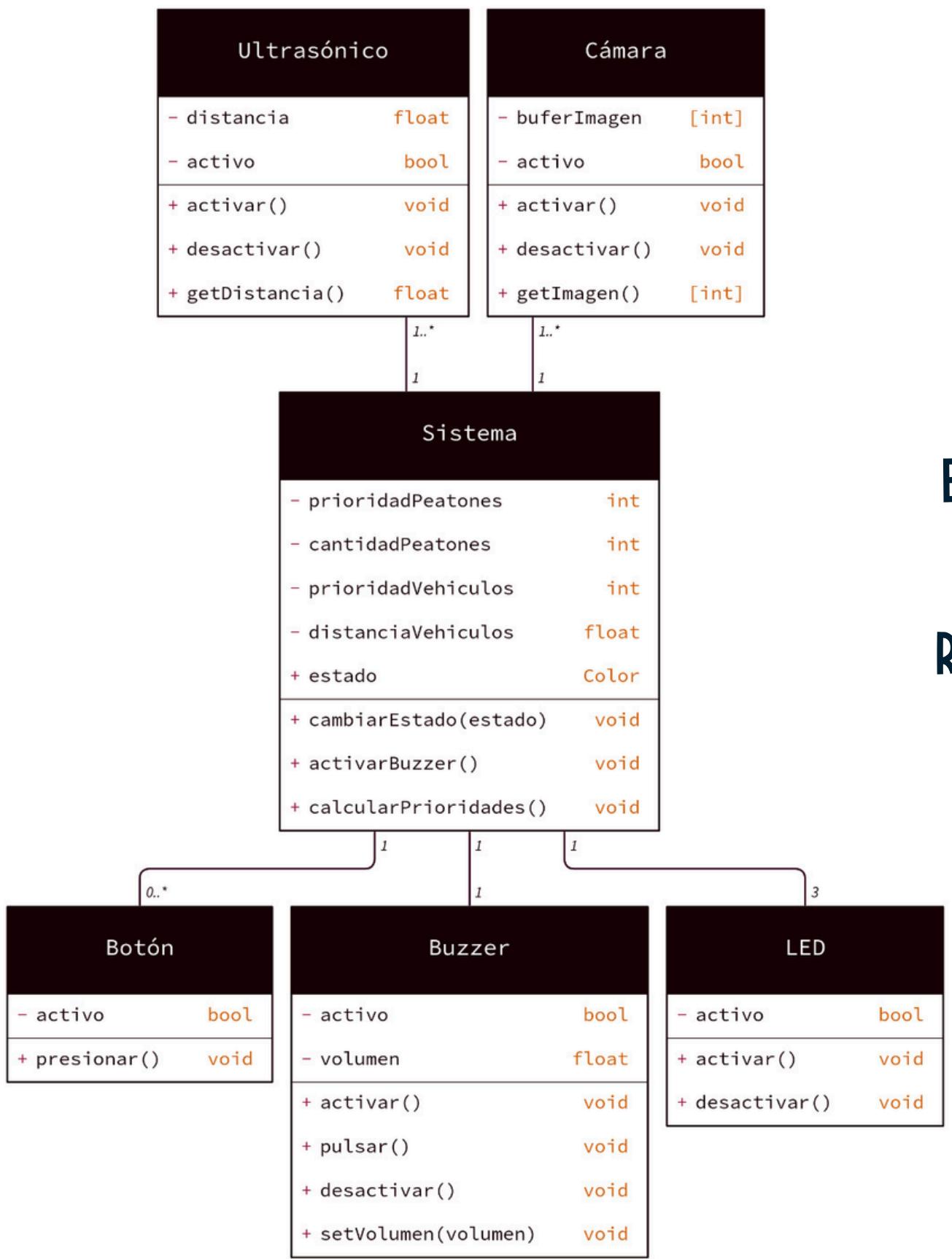


DIAGRAMA DE SECUENCIA PARA LA  
ACTIVACIÓN DEL SONIDO DEL BUZZER  
PARA PERSONAS CON DISCAPACIDADES  
VISUALES.

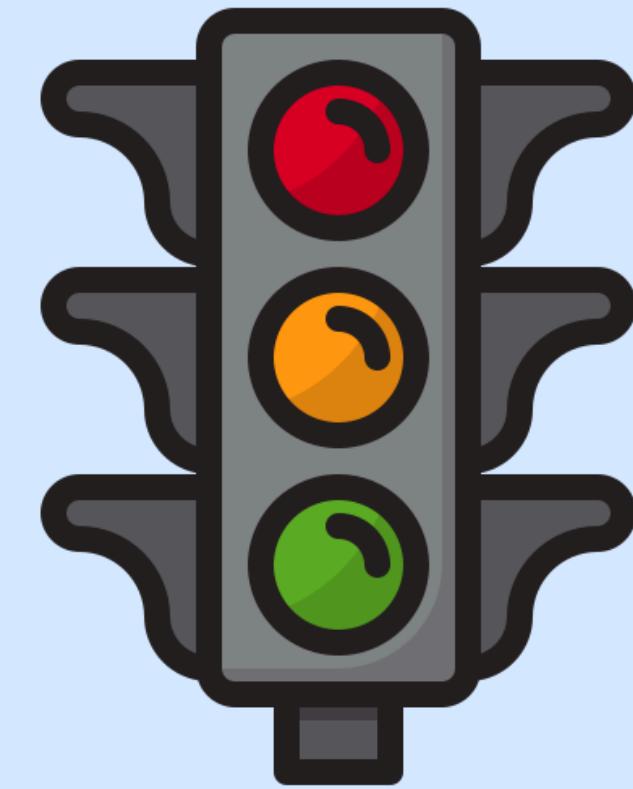
# DIAGRAMA DE CLASES



EL SIGUIENTE DIAGRAMA MUESTRA LAS FUNCIONALIDADES DE CADA UNO DE LOS COMPONENTES DEL SISTEMA Y CÓMO SE RELACIONAN ENTRE ELLOS. EN EL CENTRO ESTÁ EL SISTEMA QUE ACTÚA DE INTERMEDIARIO ENTRE LOS SENSORES Y TENIENDO CONOCIMIENTO DEL ESTADO GENERAL DEL SEMÁFORO.

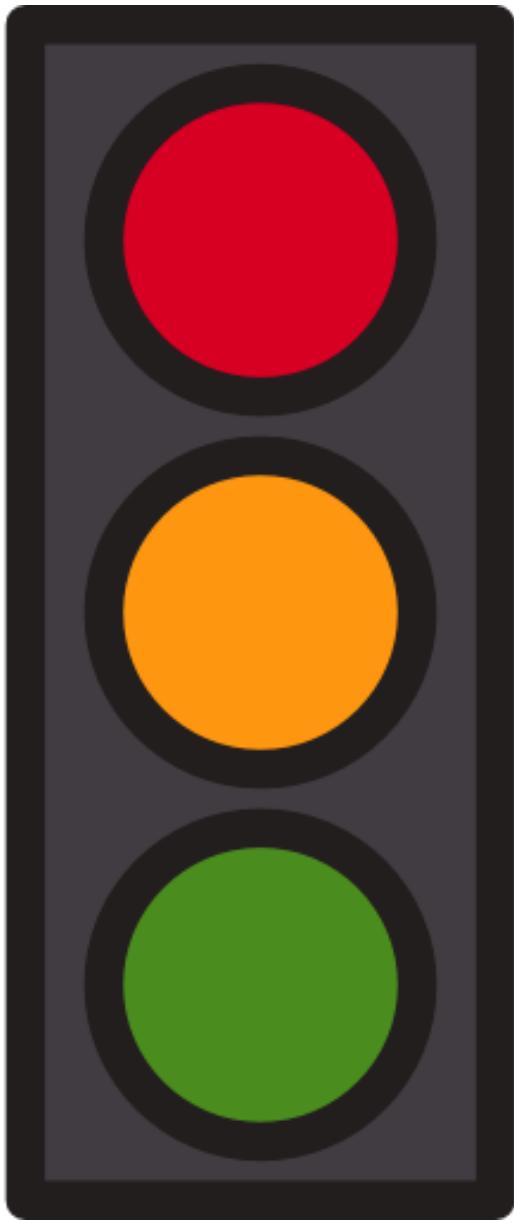
# CASOS DE USO

- Verificar estado del semáforo.
- Verificar tráfico vehicular.
- Solicitar cruce peatonal.
- Emitir sonido peatonal.
- Detectar vehículo cercano.
- Actualizar estado del semáforo
- Controlar LEDSs.



# VERIFICAR ESTADO DE SEMÁFORO

<b>Nombre del CU:</b> Verificar Estado de Semáforo	
<b>Actor(es):</b> Sistema (Raspberry Pi 4), Sensores LED	
<b>Descripción:</b> El sistema determina y almacena estado actual del semáforo (rojo, amarillo o verde) según la información recibida desde los sensores de tráfico y solicitudes peatonales. Además, actualiza las luces LED para reflejar correctamente el estado.	
<b>Precondición:</b> El sistema debe tener disponible la información actualizada del tráfico y de solicitudes de cruce para decidir el cambio de estado del semáforo	
<b>Flujo Principal: Sistema</b> <ol style="list-style-type: none"><li>1. El sistema consulta el estado del tráfico</li><li>2. El sistema verifica si existen solicitudes activas de cruce peatonal</li><li>3. El sistema evalúa las reglas de prioridad (vehículos cercanos, tiempo transcurrido, solicitudes peatonales, cantidad de peatones)</li><li>4. El sistema determina el estado que debe adoptar el semáforo</li><li>5. El sistema actualiza los LEDs mostrando el estado correspondiente a la situación</li></ol>	<b>Flujo Alternativo: Sistema</b> <ol style="list-style-type: none"><li>4.1 Si la información del tráfico es inconsistente o falta un sensor: El sistema mantiene el estado actual del semáforo.</li><li>4.2 Si no se detecta actividad vehicular durante un tiempo prolongado: El sistema prioriza el cruce peatonal o cambia a modo intermitente (según configuración).</li></ol>



**Postcondiciones:** El estado del semáforo queda actualizado y almacenado para su uso en los demás procesos del sistema.

**CU Relacionados:** Verificar estado de tráfico, Solicitar cruce peatonal

# VERIFICAR TRÁFICO

<b>Nombre del CU:</b> Verificar tráfico	
<b>Actor(es):</b> Sensor ultrasónico, cámara, sistema (Raspberry Pi 4)	
<b>Descripción:</b> El sistema detecta la presencia o ausencia de vehículos dentro del rango establecido mediante los sensores, y almacena esta información para la lógica del semáforo	
<b>Precondición:</b> El sistema debe tener los sensores activos y correctamente calibrados	
<b>Flujo Principal: Sistema</b> <ol style="list-style-type: none"><li>1. El sensor ultrasónico o la cámara monitorizan continuamente el área.</li><li>2. El sistema recibe los datos entregados por el sensor.</li><li>3. El sistema interpreta la distancia o presencia visual y determina si hay o no vehículos presentes.</li><li>4. El estado "vehículo detectado / no detectado" queda registrado internamente.</li></ol>	<b>Flujo Alternativo: Sistema</b> <ol style="list-style-type: none"><li>4.1 Si el sensor entrega datos inválidos, el sistema marca la lectura como no confiable.</li><li>4.2 Si las lecturas inválidas persisten, el sistema genera una alerta y activa un modo seguro (mantener luz roja o ciclo fijo según configuración).</li></ol>
<b>Postcondiciones:</b> El sistema actualiza la variable interna que indica el estado del tráfico y la deja disponible para la lógica del semáforo	



# SOLICITAR CRUCE PEATONAL

<b>Nombre del CU:</b> Solicitar Cruce Peatonal
<b>Actor(es):</b> Peatón, Sistema
<b>Descripción:</b> El peatón realiza una solicitud de cruce presionando un botón físico conectado al sistema, lo que activa una petición que será evaluada por el controlador del semáforo.
<b>Precondición:</b> El sistema debe estar operativo y el botón debe encontrarse correctamente conectado.
<b>Flujo Principal:</b> Peatón → Sistema <ul style="list-style-type: none"><li>1. El peatón presiona el botón de solicitud.</li><li>2. El sistema recibe la señal del botón.</li><li>3. El sistema registra la solicitud y la envía al controlador para evaluación.</li></ul>
<b>Postcondiciones:</b> La solicitud queda registrada y queda en espera de aprobación en función del estado del tráfico y la lógica del semáforo



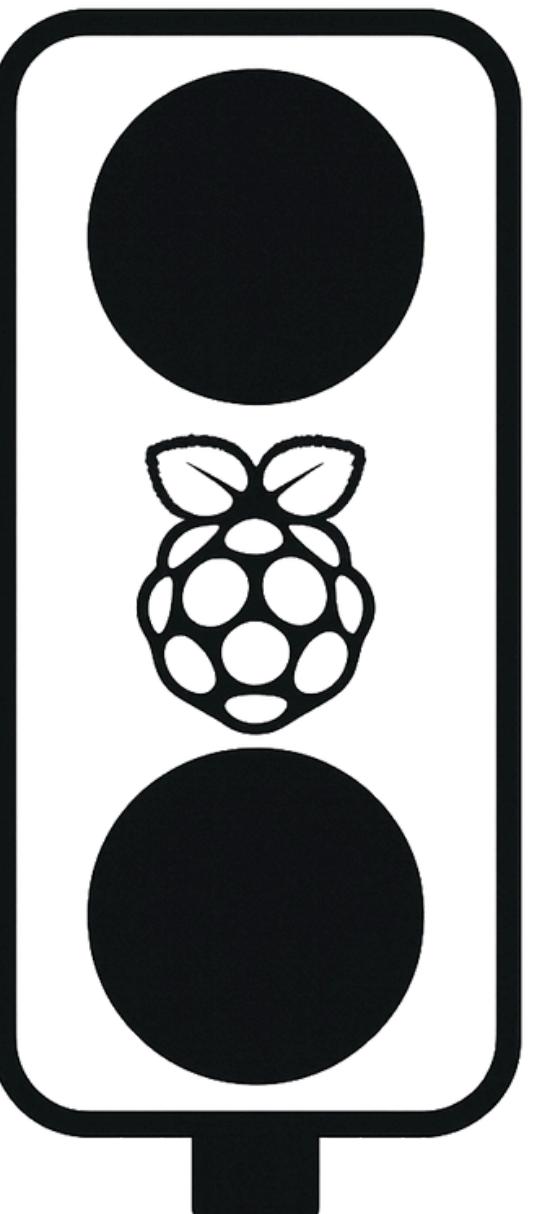
# EMITIR SONIDO DE SEMÁFORO ACTIVO

<b>Nombre del CU:</b> Emitir Sonido de Semáforo Activo
<b>Actor(es):</b> Sistema, sensor buzzer
<b>Descripción:</b> El sistema reproduce un sonido para indicar que el paso se dio a los peatones, en concordancia con una mayor accesibilidad.
<b>Precondición:</b> El sistema y el sensor buzzer deben estar operativos.
<b>Flujo Principal: Sistema</b> <ol style="list-style-type: none"><li>1. El sistema admite el paso de peatones</li><li>2. Envía una señal para activar el sensor buzzer</li><li>3. Cerca del fin de la duración del semáforo, el sistema indica que el buzzer debe reproducir un sonido pulsante</li><li>4. Cuando se acaba la duración, el sistema apaga el buzzer</li></ol>
<b>Postcondiciones:</b> El buzzer no reproduce ningún sonido



# ACTUALIZAR ESTADO DEL SEMÁFORO

<b>Nombre del CU:</b> Actualizar estado del semáforo	
<b>Actor(es):</b> Sistema, Sensores LED	
<b>Descripción:</b> El sistema ejecuta el cambio de estado del semáforo (rojo, amarillo o verde), activando los LEDs correspondientes según las reglas de prioridad y detección vehicular/peatonal	
<b>Precondición:</b> El sistema debe haber evaluado previamente el estado del tráfico y las solicitudes peatonales	
<b>Flujo Principal: Sistema</b> <ol style="list-style-type: none"><li>1. El sistema determina el estado objetivo del semáforo.</li><li>2. El sistema apaga el estado luminoso actual.</li><li>3. El sistema enciende los LEDs del estado correspondiente (rojo, amarillo o verde).</li><li>4. El sistema registra internamente el cambio realizado.</li></ol>	<b>Flujo Alternativo: Sistema</b> <ol style="list-style-type: none"><li>3.1 Si se detecta una transición inestable (encendido incorrecto), el sistema vuelve al estado previo.</li></ol>
<b>Postcondiciones:</b> El semáforo refleja el estado correcto en tiempo real.	
<b>CU Relacionados:</b> Estado de semáforo, Controlar LED's	



# DETECTOR VEHÍCULO CERCANO

<b>Nombre del CU:</b> Detectar vehículo cercano	
<b>Actor(es):</b> Sensor ultrasónico, Sensor cámara, Sistema	
<b>Descripción:</b> El sistema detecta la presencia de un vehículo dentro del rango crítico cercano al cruce para ajustar el comportamiento del semáforo.	
<b>Precondición:</b> Sensores conectados y calibrados.	
<b>Flujo Principal: Sistema</b> <ol style="list-style-type: none"><li>1. El sensor ultrasónico/cámara detecta un objeto dentro del rango configurado.</li><li>2. El sistema verifica si corresponde a un vehículo.</li><li>3. El sistema marca la variable interna "vehículo cercano = verdadero".</li><li>4. La información se envía al módulo de prioridad del semáforo.</li></ol>	<b>Flujo Alternativo: Sistema</b> <ol style="list-style-type: none"><li>1.1 Si la detección es confusa (sombra, persona, animal), el sistema solicita una segunda medición.</li><li>1.2 Si sigue inconsistente, la detección se descarta.</li></ol>
<b>Postcondiciones:</b> El sistema actualiza la variable "vehículo cercano" para la lógica del semáforo.	
<b>CU Relacionados:</b> Verificar estado del tráfico, Estado de semáforo	



# CONTROLAR LEDS DEL SEMÁFORO

<b>Nombre del CU:</b> Controlar LEDs del semáforo
<b>Actor(es):</b> Sistema, Sensores LED
<b>Descripción:</b> El sistema enciende o apaga los LEDs correspondientes al estado actual del semáforo.
<b>Precondición:</b> El sistema debe conocer el estado objetivo del semáforo.
<b>Flujo Principal: Sistema</b> <ol style="list-style-type: none"><li>1. El sistema identifica qué color debe activarse.</li><li>2. El sistema envía señal eléctrica <u>al LED</u> correspondiente.</li><li>3. El sistema verifica que el LED cambió correctamente.</li></ol>
<b>Postcondiciones:</b> Los LEDs corresponden al estado lógico del semáforo



# HERRAMIENTAS

**Raspberry Pi 4B:** Controlador principal del sistema. Gestiona sensores, cámara y lógica del semáforo en tiempo real.

**Python:** Lenguaje utilizado para la programación del semáforo, lectura de sensores y control del buzzer y LEDs.

**Geany:** Editor usado dentro del Raspberry para ejecutar y probar código de manera directa y ágil.

# TECNICAS

**Pruebas unitarias:** Se evaluó cada sensor individualmente para verificar su funcionamiento correcto.

**Integración progresiva:** Conexión e incorporación de sensores y cámara paso a paso, verificando estabilidad.

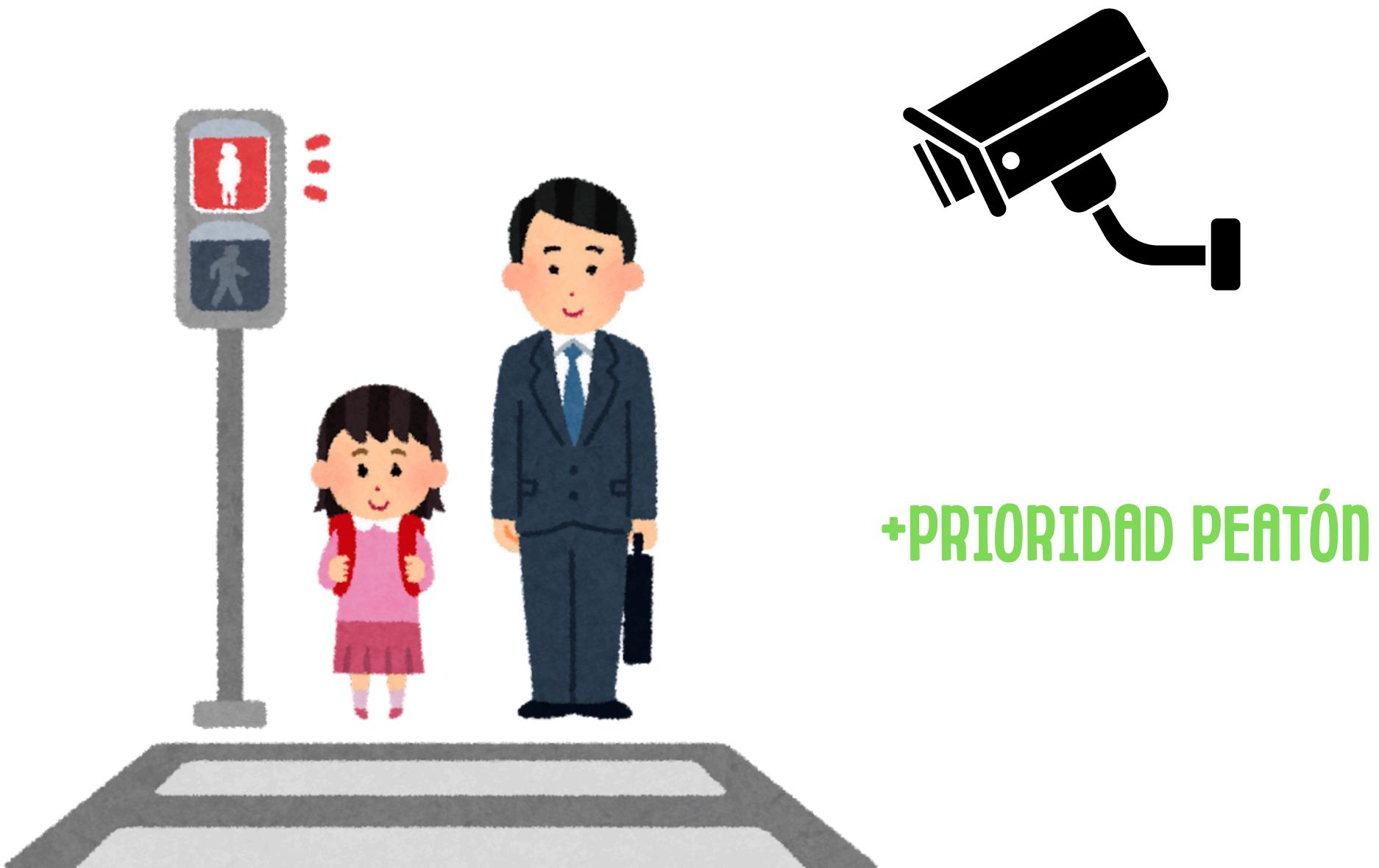
**Ciclo prueba-error:** Correcciones continuas de código y conexiones según los resultados reales de las pruebas.

# SECUENCIA DE FUNCIONAMIENTO



+PRIORIDAD PEATÓN

# SECUENCIA DE FUNCIONAMIENTO



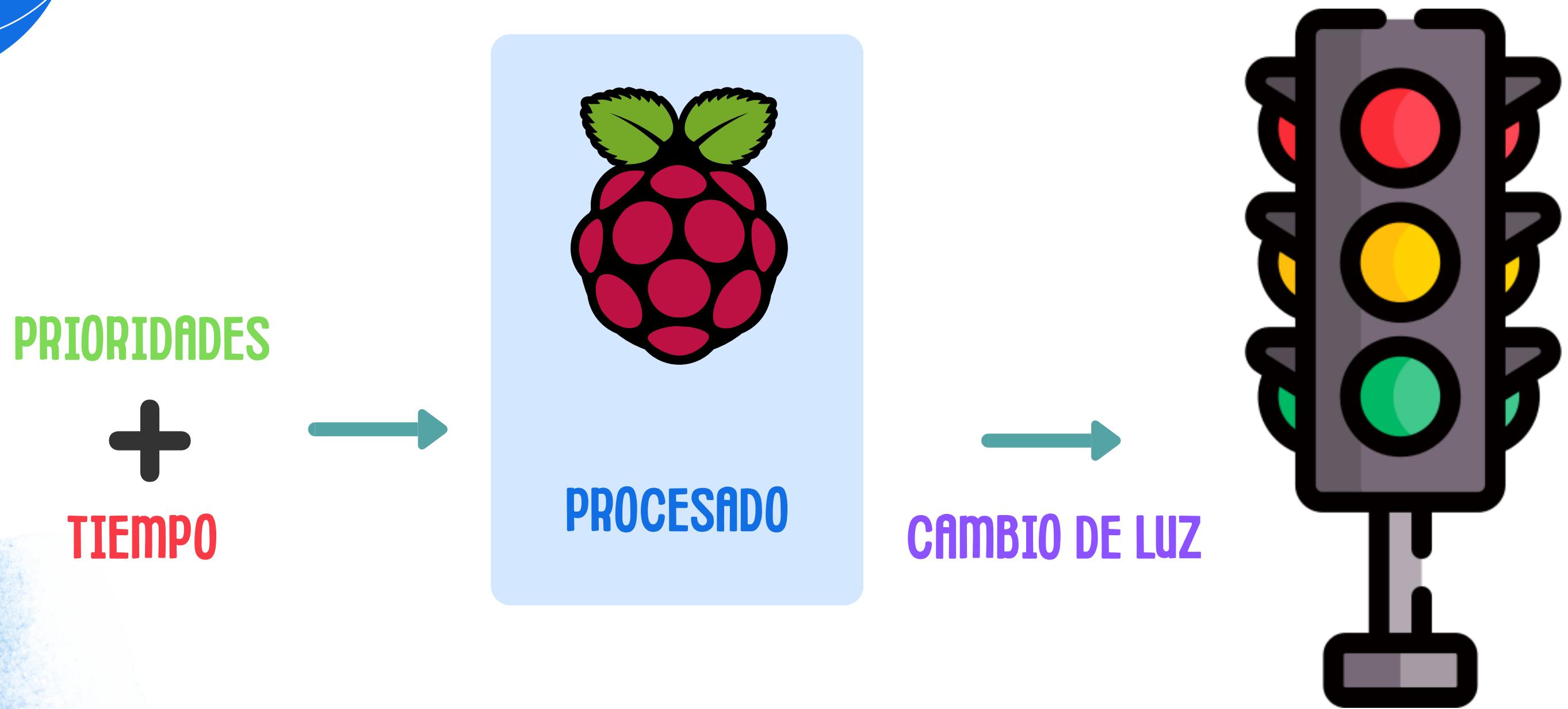
+PRIORIDAD PEATÓN

# SECUENCIA DE FUNCIONAMIENTO



+PRIORIDAD AUTOS

# SECUENCIA DE FUNCIONAMIENTO



# CONCLUSIÓN

El proyecto permitió definir la base conceptual y técnica del Semáforo Inteligente para Peatones. En esta fase se avanzó con la planificación técnica, herramientas, técnicas aplicadas y diagramas de arquitectura, casos de uso e interacción del usuario.

Con los sensores definidos y la funcionalidad del sistema avanzando bien, se espera que el proyecto llegue a una conclusión satisfactoria.

A futuro se considerarán adiciones no planeadas actualmente, como una interfaz de usuario dirigida a personas con discapacidades visuales.