



Universidad de Tarapacá



**HUSH** →

**Equipo**

Iván Collao  
Sebastian Eyraud  
Guillermo Pino  
Giorgio Rojas

**Profesor**

Diego Aracena

# Contenido →

Requerimientos funcionales

Requerimientos no funcionales

Modelo de diseño (caso de uso general)

Casos de uso y diagrama de secuencia

Descripción de la arquitectura

Herramientas y técnicas

# Requerimientos



## Requerimientos funcionales

- Reconocer sonidos en niveles altos de decibeles.
- Emitir una notificación al usuario en tiempo real (alertas).
- Permite al usuario la configuración de parámetros (establecer rangos de decibeles para la alerta de sonidos).

## Requerimientos no funcionales

- Interfaz intuitiva de fácil uso para el usuario.
- Fácil mantenibilidad para actualizar y corregir el sistema.
- Indicador del estado mediante un indicador visual (LED) que muestra el estado operativo.
- El dispositivo es capaz de funcionar con una batería portable.

# Modelo de diseño (caso de uso general)



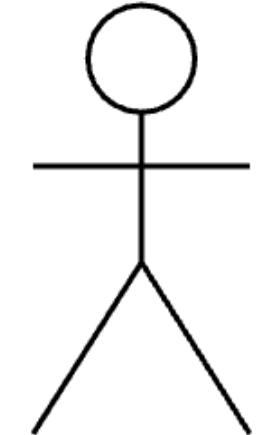
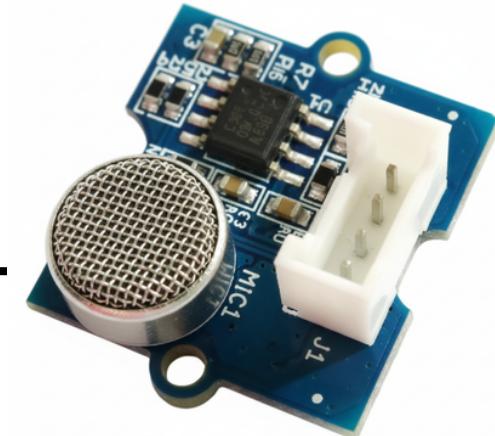
## Sistema

Detectar un ruido que sobrepasa el límite establecido

Emite un mensaje que se superó el umbral establecido

Enciende o apaga el dispositivo

Configura el límite de decibeles permitidos / Mostrará notificaciones en caso de que se haya superado el límite de ruido



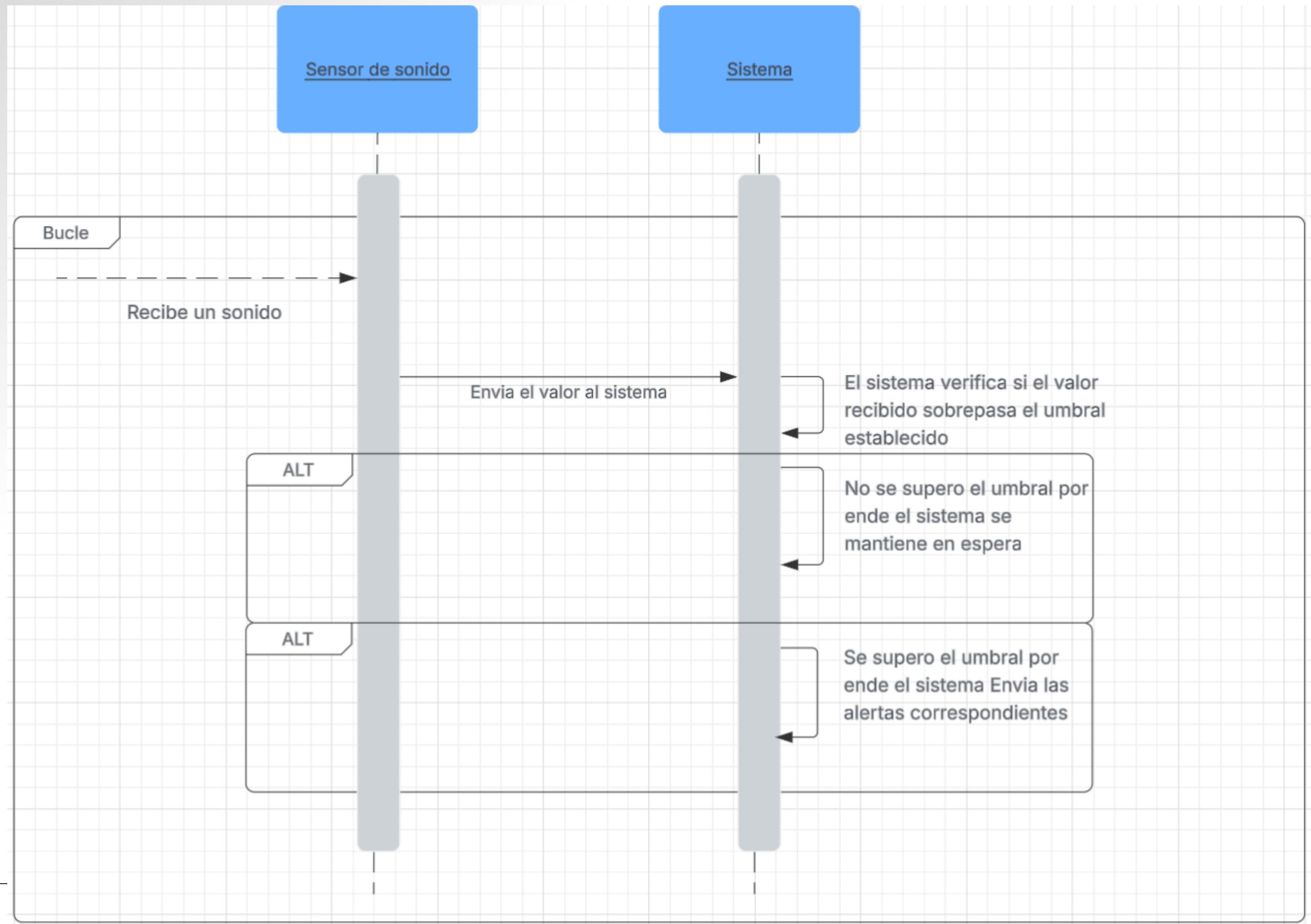
Usuario



# Casos de uso y diagramas de secuencia

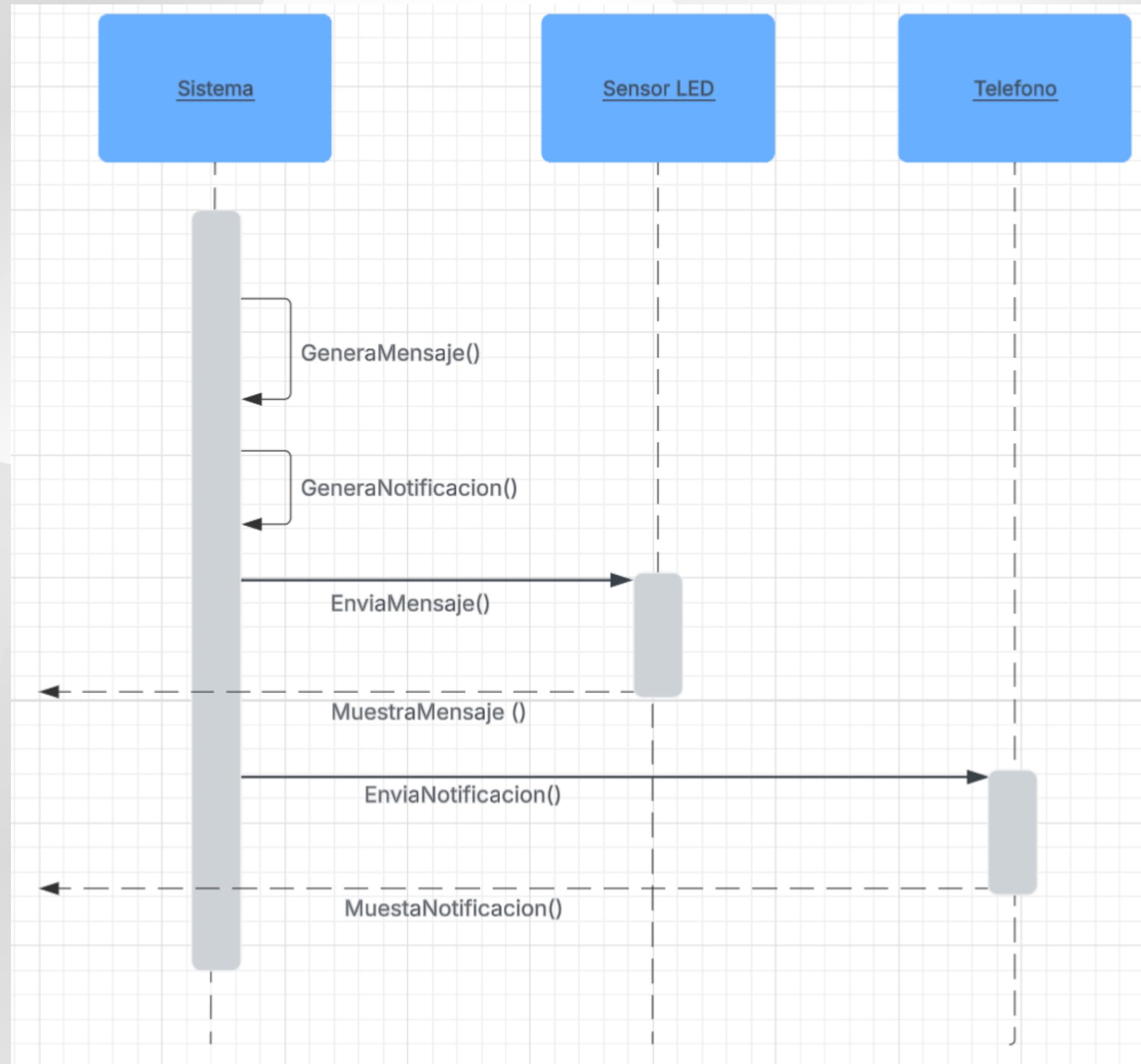
# Detección de ruido

<b>Nombre de caso de Uso:</b> Detección de ruido
<b>Resumen:</b> <p>El sistema recibirá información cada vez que el sensor de sonido registre un ruido que haya superado el límite predefinido.</p>
<b>Actor(es):</b> Sistema y sensor de sonido.
<b>Precondición:</b> <p>El sistema debe estar encendido y a la espera de que se produzca algún ruido.</p>
<b>Descripción:</b> <ol style="list-style-type: none"><li>1. El sensor de sonido detecta un valor de ruido ambiente.</li><li>2. El sistema compara el valor obtenido con el umbral definido por el usuario.</li><li>3. Si el nivel de ruido supera el umbral, el sensor entrega el valor al sistema.</li><li>4. El sistema recibe y valida la señal, confirmando que el límite de ruido fue excedido.</li><li>5. El sistema activa el LED indicador para señalar localmente la detección.</li></ol>
<b>Alternativa:</b> <p>3.1. Si el nivel de ruido detectado no supera el umbral, el sistema continúa en estado de espera sin activar el LED ni enviar notificaciones.</p>
<b>Postcondición:</b> <p>El sistema queda listo para continuar monitoreando el nivel de ruido y activar nuevas alertas cuando corresponda.</p>



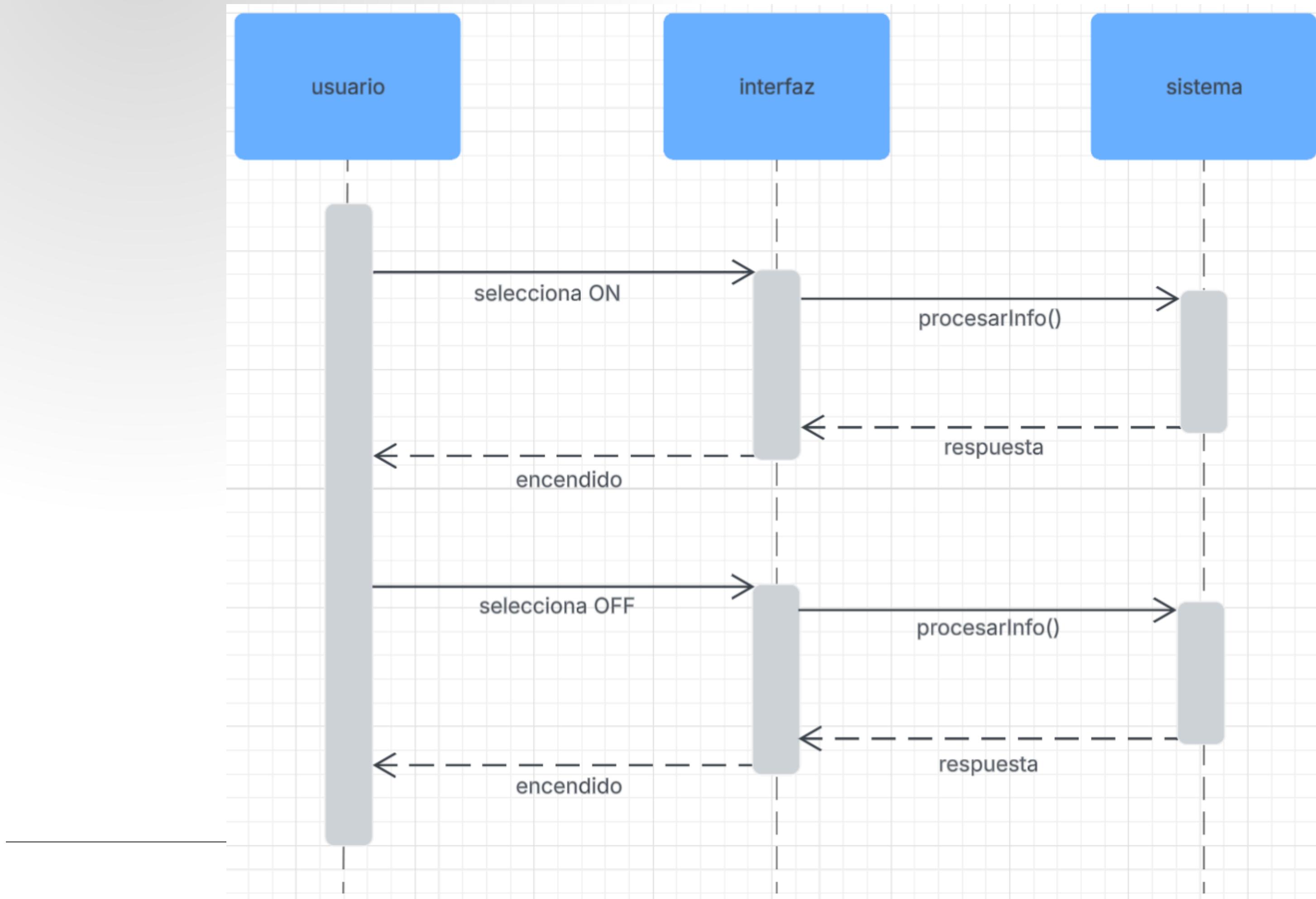
# Notificar al usuario

<b>Nombre de caso de Uso:</b> Notificar al usuario
<b>Resumen:</b> <p>El sistema, utilizando la Raspberry Pi 4B y un micrófono, detecta un nivel de sonido que excede un umbral establecido y envía una notificación de alerta al usuario por su canal preferido (e.g., email, mensaje push, Telegram).</p>
<b>Actor(es):</b> Sistema, usuario, sensor LED, Teléfono
<b>Precondición:</b> <p>El sistema debe estar encendido y ejecutando el software, el micrófono debe estar conectado y funcionando correctamente. El umbral de ruido (dB) ha sido configurado. La configuración de notificación del usuario (canal y destino) es válida y accesible</p>
<b>Descripción:</b> <ol style="list-style-type: none"><li>Al detectar un nivel de dB que supera el umbral definido y recibido el valor entregado por el sensor se inicia el evento de notificar al usuario.</li><li>El sistema genera un mensaje de alerta y envía una notificación al usuario y muestra un mensaje en el sensor led.</li></ol>
<b>Alternativa:</b> <ol style="list-style-type: none"><li>Falla al enviar la notificación. Si el servicio de notificación no responde o falla el envío.</li></ol>
<b>Postcondición:</b> El usuario ha recibido una notificación de la alerta de ruido



# Encender y apagar el sistema

<b>Nombre de caso de Uso:</b> Encender y apagar el sistema
<b>Resumen:</b> El usuario puede encender o apagar el sistema mediante un botón físico o interruptor conectado a la Raspberry Pi. El encendido inicializa los componentes necesarios para el monitoreo del sensor de sonido, mientras que el apagado detiene todos los procesos y coloca al sistema en un estado seguro
<b>Actor(es):</b> Sistema, usuario.
<b>Precondición:</b> El sistema debe contar con una batería o pila instalada y con suficiente carga. El usuario debe interactuar con los botones de la interfaz gráfica.
<b>Descripción:</b> <ol style="list-style-type: none"><li>1. El usuario presiona el botón físico de encendido/apagado del dispositivo.</li><li>2. El sistema detecta la señal manual enviada por el usuario.</li><li>3. Si la señal corresponde a encender, el sistema inicializa los módulos principales, incluyendo el sensor de sonido, la interfaz de notificaciones y el LED indicador.</li><li>4. Si la señal corresponde a apagar, el sistema detiene de forma segura todas las funciones activas, apaga el LED e ingresa en modo de bajo consumo para preservar la batería.</li><li>5. El sistema verifica que cada componente haya terminado su proceso de inicialización o apagado correctamente.</li></ol>
<b>Alternativa:</b> <b>2.1</b> Si el usuario presiona el botón cuando la batería se encuentra con carga insuficiente, el sistema no inicia e informa la condición mediante un parpadeo breve del LED.
<b>Postcondición:</b> El usuario ha recibido una notificación de la alerta de ruido



# Arquitectura →

## **Capa de adquisición de datos**

Compuesta por el sensor de sonido FC-04. (Sound Sensor Groove v1.6), captura niveles de ruido en tiempo real y envía los datos a la Raspberry Pi mediante señales procesadas por librerías de Python.

## **Capa de procesamiento y lógica**

Ejecutada en la Raspberry Pi 4 usando Python. Se encarga de interpretar valores del sensor, compararlos con umbrales definidos, encendido y apagado del sistema, controlar el LED indicador y preparar y enviar notificaciones al usuario.

## **Capa de notificación e interacción**

Informa al usuario mediante el LED indicador conectado a GPIO y notificaciones enviadas por la red (alertas push), esto permite que el usuario configure los parámetros del sistema (umbrales de ruido y ajustes de sensibilidad).

---

# Herramientas y técnicas



## Software

Raspberry Pi OS.  
Python 3.  
Thonny IDE.  
Visual Studio.  
Redmine UTA.  
Google Docs.

## Hardware

Raspberry Pi 4.  
Sensor de sonido FC-04.  
LED indicador.  
Grove button ON/OFF.  
Tarjeta MicroSD.

## Técnicas

Programación modular.  
Metodología incremental.

# Conclusión

A pesar de que las principales dificultades iniciales se centraron en la instalación del sistema operativo en la Raspberry Pi 4B y los conflictos con la librería Grove Pi, estos desafíos fueron superados con éxito, permitiendo que el proyecto avanzara hacia su fase de término. El objetivo actual es implementar la lógica de programación para el correcto funcionamiento de los sensores Grove Pi en la detección de ruido, mientras que de forma paralela se finalizarán tareas críticas como el diseño de la carcasa, la elaboración del manual de usuario y la documentación final del proyecto.



Muchas gracias  
por su atención :)

**HUSH** A red bell icon with three concentric arcs above it, suggesting sound or notification.

