



# **SISTEMA DE RIEGO AUTOMATIZADO (INFORME FASE II)**

**Integrantes:**

**Bairon Núñez**

**Christopher Romo**

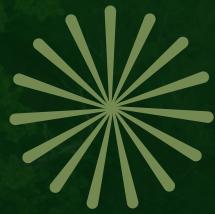
**Gustavo Morales**

**Profesor:**

**Diego Aracena**



# Introducción



**Esta presentación se centra en la fase de planificación y diseño técnico del Sistema de Riego Automatizado. Nuestro objetivo ha sido traducir los requerimientos funcionales en una arquitectura de software y hardware viable, lista para su implementación.**



**Abordaremos dos pilares fundamentales que sustentan la viabilidad del proyecto:**

- **Modelado del Sistema:** Definición precisa del comportamiento a través de diagramas UML (Casos de uso, Secuencia) y una arquitectura Cliente-Servidor.
- **Herramientas y Metodología:** Selección del stack tecnológico (Python, Raspberry Pi) y uso de prototipado iterativo.





# Modelos de proceso

## Requerimientos

Especificaciones funcionales y no funcionales del sistema de riego automatizado.

## Casos de uso

Interacciones entre el usuario y el sistema para cada funcionalidad principal.

## Diagrama de secuencia

Flujo temporal de operaciones entre componentes del sistema.

## Diagrama de clases

Estructura lógica del software y relaciones entre entidades.



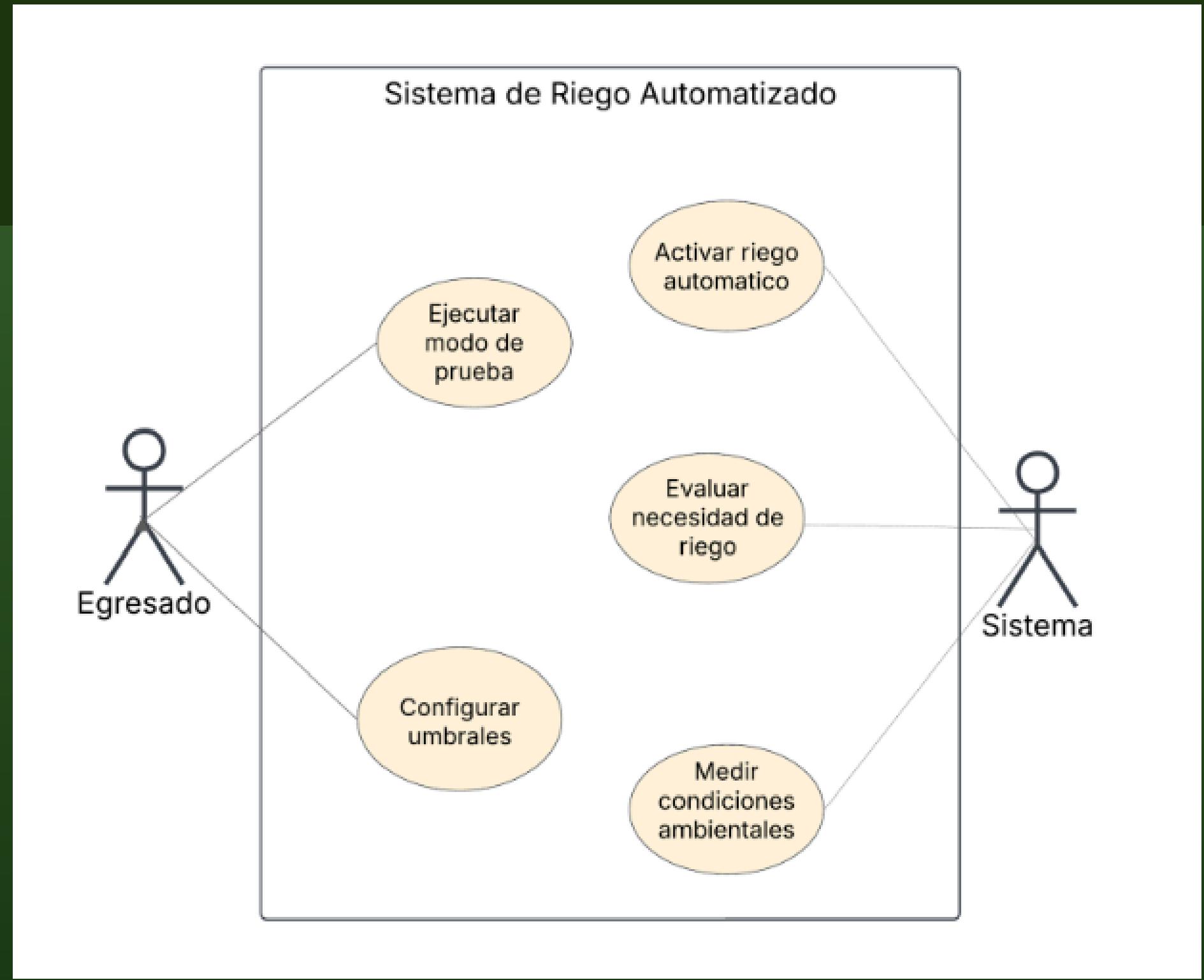
# Requerimientos



Requerimientos Funcionales		Requerimientos No Funcionales	
<b>RF1 – Medición de humedad del suelo</b>	El sistema debe leer periódicamente el valor de humedad del suelo mediante el sensor de humedad, con una frecuencia configurable (por ejemplo, cada X segundos).	<b>RNF1 – Confiabilidad de lectura</b>	Las lecturas de los sensores de humedad y temperatura deben ser filtradas o validadas (por ejemplo, promediando varias muestras) para reducir lecturas erróneas o picos de ruido.
<b>RF2 – Medición de temperatura ambiente</b>	El sistema debe leer periódicamente la temperatura ambiente mediante el sensor correspondiente , para complementar la decisión de riego y generar registros básicos de condiciones ambientales.	<b>RNF2 – Disponibilidad del sistema</b>	El sistema debe poder funcionar de manera continua durante el periodo de prueba del proyecto, suponiendo una fuente de alimentación estable, sin bloqueos frecuentes del programa.
<b>RF3 – Evaluación automática de la necesidad de riego</b>	El sistema debe comparar los valores de humedad del suelo con un umbral predefinido (humedad mínima aceptable) y determinar si es necesario activar el riego.	<b>RNF3 – Tiempo de respuesta</b>	El tiempo entre la detección de una condición de sequedad (humedad bajo umbral) y la activación de la bomba no debe superar unos pocos segundos, de manera que el riego parezca inmediato desde la perspectiva del usuario.
<b>RF4 – Activación automática de la bomba de agua</b>	Cuando la humedad del suelo sea inferior al umbral definido, el sistema debe activar automáticamente la bomba de agua hasta que la humedad alcance o supere el valor objetivo	<b>RNF4 – Seguridad básica del hardware</b>	Las conexiones eléctricas deben estar protegidas (uso de relé, fusible o componentes adecuados) para evitar daños a la Raspberry Pi, a la bomba de agua o riesgo de cortocircuitos.
<b>RF5 – Detención automática del riego</b>	El sistema debe detener la bomba de agua cuando la humedad del suelo alcance el umbral superior definido, evitando el exceso de riego.	<b>RNF5 – Mantenibilidad</b>	El código debe estar comentado y organizado en módulos (por ejemplo, módulo de sensores, módulo de lógica de riego, módulo de registros) para facilitar futuras modificaciones y ampliaciones del sistema.
<b>RF6 – Modo de prueba / configuración</b>	El sistema debe permitir ejecutar un modo de prueba en el que la bomba de agua pueda activarse brevemente para verificar el correcto funcionamiento del circuito (sin depender del valor de humedad).	<b>RNF6 – Usabilidad técnica</b>	La configuración de umbrales de humedad y tiempos de muestreo debe poder ajustarse fácilmente en el código o en un archivo de configuración, sin requerir cambios complejos.
<b>RF7 – Registro básico de eventos de riego</b>	El sistema debe registrar los eventos de riego (inicio, fin, duración aproximada y valor de humedad en esos momentos) en un archivo de log o estructura simple de datos para análisis posterior.	<b>RNF7 – Escalabilidad básica</b>	El diseño debe permitir que en el futuro pueda ampliarse el sistema para controlar más zonas de riego o integrar nuevas variables (por ejemplo, luz o pH), sin tener que rediseñar completamente la lógica principal.
<b>RF8 – Indicación de estado del sistema</b>	El sistema debe entregar alguna señal visible (por ejemplo, mensaje por terminal, LED o salida por consola) que indique al usuario el estado actual: "Midiendo humedad", "Riego activo", "Riego detenido", "Error de sensor", etc.		

# Casos de uso



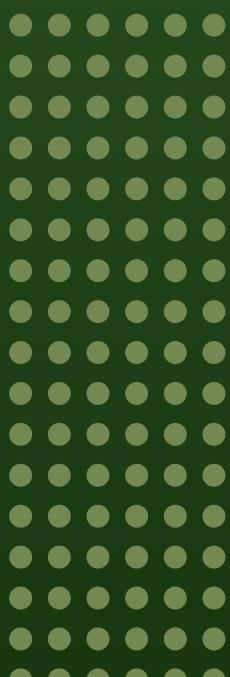


## Caso de Uso 1: Medir condiciones ambientales

<b>Nombre CU:</b> CU-01 – Medir condiciones ambientales	
Autor: Gustavo Morales, Bairon Núñez	Fecha: 11/11/2025
<b>Descripción:</b> El sistema realiza la lectura periódica del sensor de humedad del suelo y del sensor de temperatura ambiental, obteniendo los valores necesarios para evaluar la necesidad de riego.	
<b>Actor:</b> Sistema de riego automatizado (Raspberry Pi)	
<b>Precondición:</b> El sistema está encendido y ejecutando el programa de control y los sensores están correctamente conectados y alimentados.	
<b>Flujo Principal:</b> No aplica, ya que es automático	<b>Flujo Principal: Sistema</b> 1. Inicia ciclo de medición 2. Sigue al sensor de humedad el valor actual del suelo 3. Sigue al sensor de temperatura el valor ambiente. 4. Registra temporalmente ambos valores.
<b>Flujo Alternativo:</b>	<b>Flujo Alternativo: Sistema</b> 2.1 Error al registrar humedad, <u>reintenta el ciclo de medición</u> 3.1 Error al registrar temperatura, <u>reintenta el ciclo de medición</u>
<b>Postcondiciones:</b> Se dispone de lecturas actualizadas de humedad y temperatura	

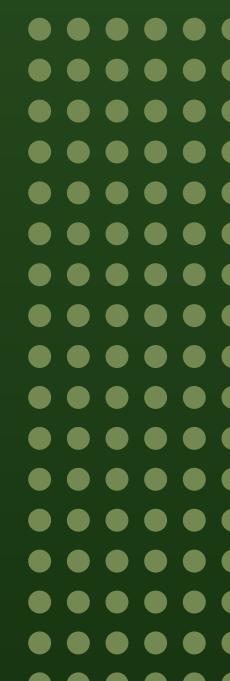
## Caso de Uso 2: Evaluar necesidad de riego

<b>Nombre CU:</b> CU-02 – Evaluar necesidad de riego	
<b>Autor:</b> Gustavo Morales, Christopher Romo	<b>Fecha:</b> 13/11/2025
<b>Descripción:</b> El sistema analiza la humedad medida para determinar si debe activar la bomba de agua.	
<b>Actor:</b> Sistema de riego automatizado (Raspberry Pi)	
<b>Precondición:</b> Valores de humedad obtenidos correctamente y Umbral de humedad configurado.	
<b>Flujo Principal:</b> No aplica, ya que es automático	<b>Flujo Principal: Sistema</b> <ol style="list-style-type: none"><li>1. Compara la humedad actual con el umbral mínimo configurado.</li><li>2. Humedad inferior al umbral, determina “riesgo necesario”</li></ol>
<b>Flujo Alternativo:</b>	<b>Flujo Alternativo: Sistema</b> <ol style="list-style-type: none"><li>2.1 humedad superior al umbral o suficiente, no activa riego</li></ol>
<b>Postcondiciones:</b> Se ha decidido si el sistema debe activar el riego	



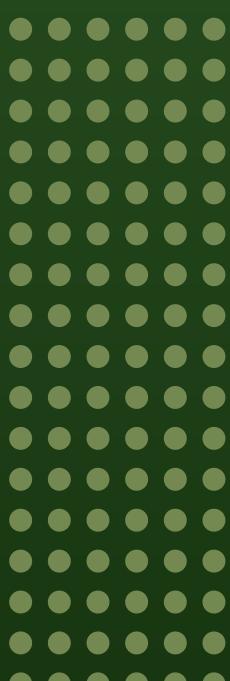
## Caso de Uso 3: Activar riego automático

<b>Nombre CU:</b> CU-03 – Activar riego automático	
<b>Autor:</b> Gustavo Morales, Bairon Núñez	<b>Fecha:</b> 15/11/2025
<b>Descripción:</b> El sistema activa la bomba de agua para regar el huerto hasta que la humedad del suelo alcance niveles adecuados.	
<b>Actor:</b> Sistema de riego automatizado y Bomba de Agua	
<b>Precondición:</b> La bomba está conectada y operativa y se determinó que el riego es necesario	
<b>Flujo Principal:</b> No aplica	<b>Flujo Principal: Sistema</b> <ol style="list-style-type: none"><li>Envía señal al relé para encender la bomba.</li><li>Registra el inicio del riego.</li><li>Realiza lecturas periódicas de humedad.</li><li>Cuando la humedad supera el umbral superior, se apaga la bomba.</li><li>Registra el fin del riego.</li></ol>
<b>Flujo Alternativo:</b>	<b>Flujo Alternativo: Sistema</b>
<b>Postcondiciones:</b> El suelo fue regado, nivel de humedad sobre el umbral	



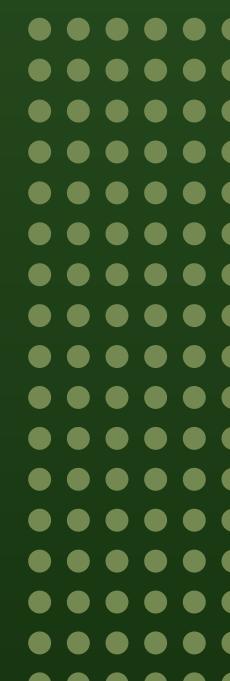
## Caso de Uso 4: Ejecutar modo de prueba

<b>Nombre CU:</b> CU-04 – Ejecutar modo de prueba de riego	
<b>Autor:</b> Gustavo Morales, Bairon Núñez	<b>Fecha:</b> 20/11/2025
<b>Descripción:</b> El usuario ejecuta un modo de prueba que activa brevemente la bomba para verificar su funcionamiento sin depender de los valores de humedad.	
<b>Actor:</b> Usuario	
<b>Precondición:</b> Sistema encendido y usuario con acceso a comandos o interfaz de prueba.	
<b>Flujo Principal:</b> Usuario 1.Iniciar modo de prueba  3. Ingresar confirmación	<b>Flujo Principal:</b> Sistema 2.Solicita confirmación  4.Activa la bomba por pocos segundos 5. Muestra resultado
<b>Flujo Alternativo:</b>	<b>Flujo Alternativo:</b> Sistema
<b>Postcondiciones:</b> El usuario conoce el estado real del hardware de riego.	



## Caso de Uso 5: Configurar umbrales y parámetros del sistema

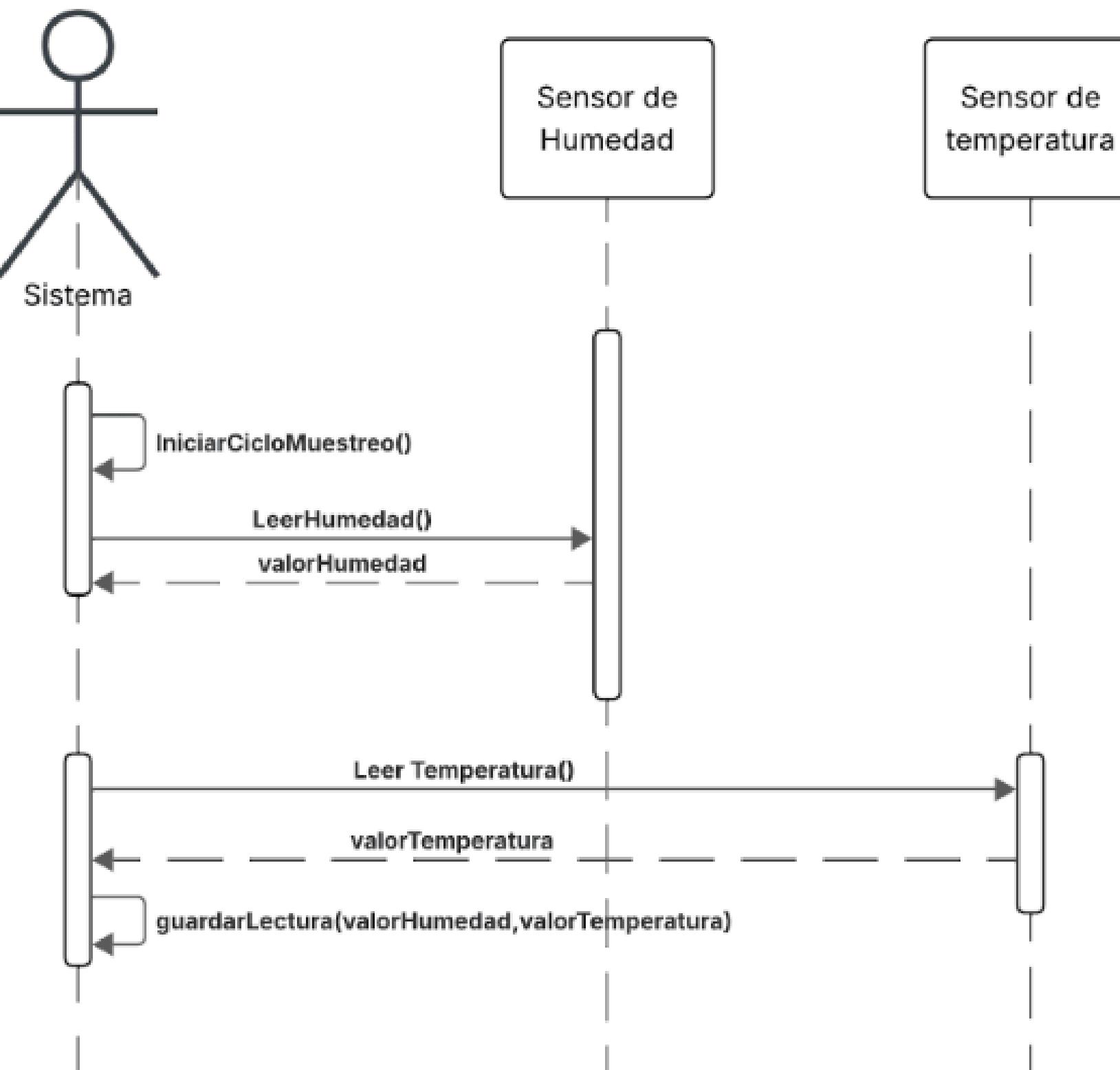
<b>Nombre CU:</b> CU-05 – Configurar umbrales de humedad	
<b>Autor:</b> Gustavo Morales	<b>Fecha:</b> 20/11/2025
<b>Descripción:</b> El usuario configura valores de humedad mínima, humedad máxima y frecuencia de muestreo del sistema	
<b>Actor:</b> Usuario	
<b>Precondición:</b> Sistema encendido y se debe poder acceder al archivo de configuración o menú de parámetros.	
<b>Flujo Principal:</b> Usuario 1.Solicita modificar parámetros del sistema. 2.Ingresa el umbral mínimo de humedad. 3.Ingresa el umbral máximo. 4.Define frecuencia de medición.  7.Ingresa guarda cambios.	<b>Flujo Principal:</b> Sistema  5. Valida los valores ingresados. 6. Actualiza configuración interna.  8. Registra cambios.
<b>Flujo Alternativo:</b>	<b>Flujo Alternativo:</b> Sistema
<b>Postcondiciones:</b> El sistema registra nuevos parámetros de funcionamiento.	



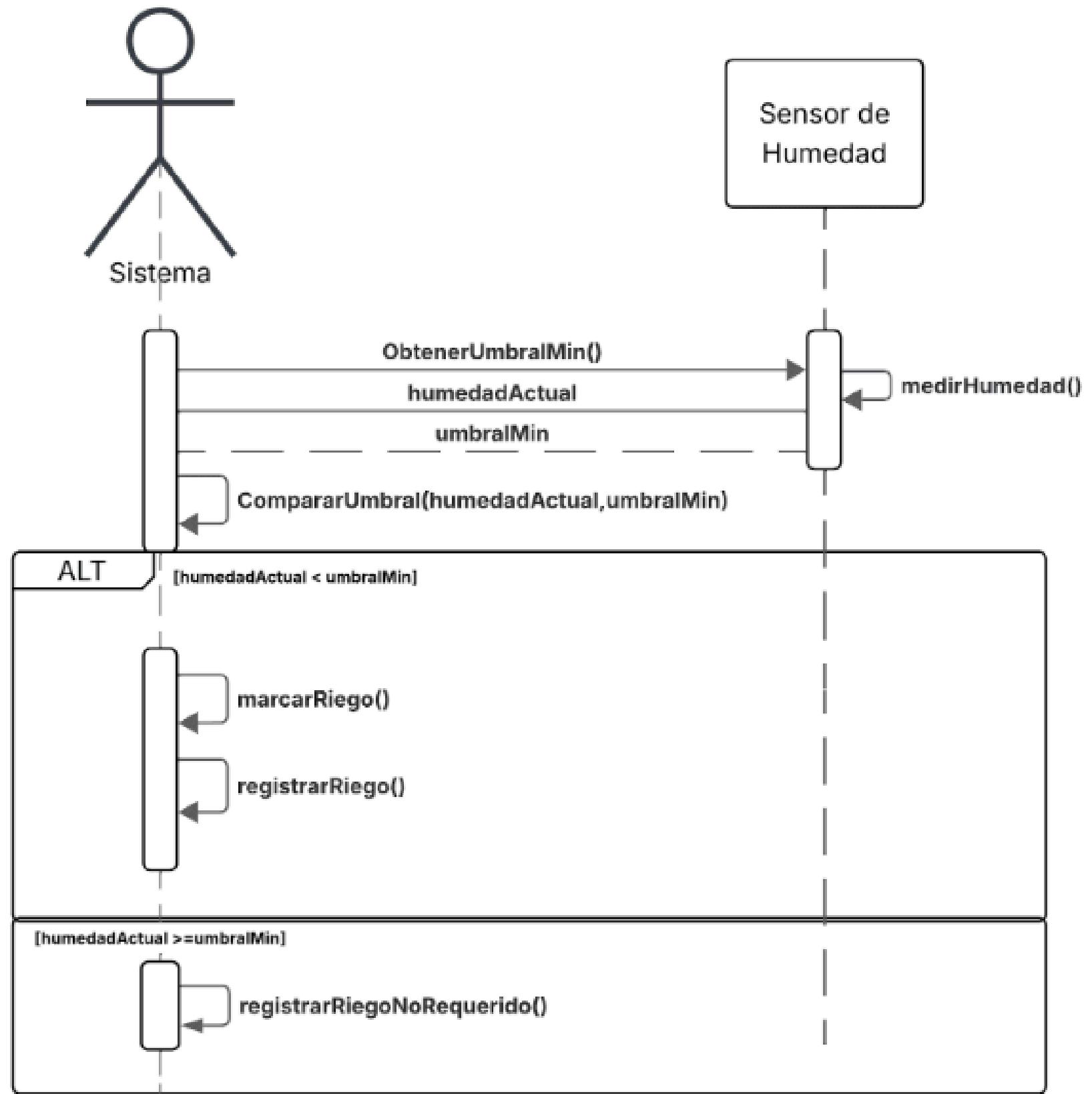
# Diagrama de secuencia



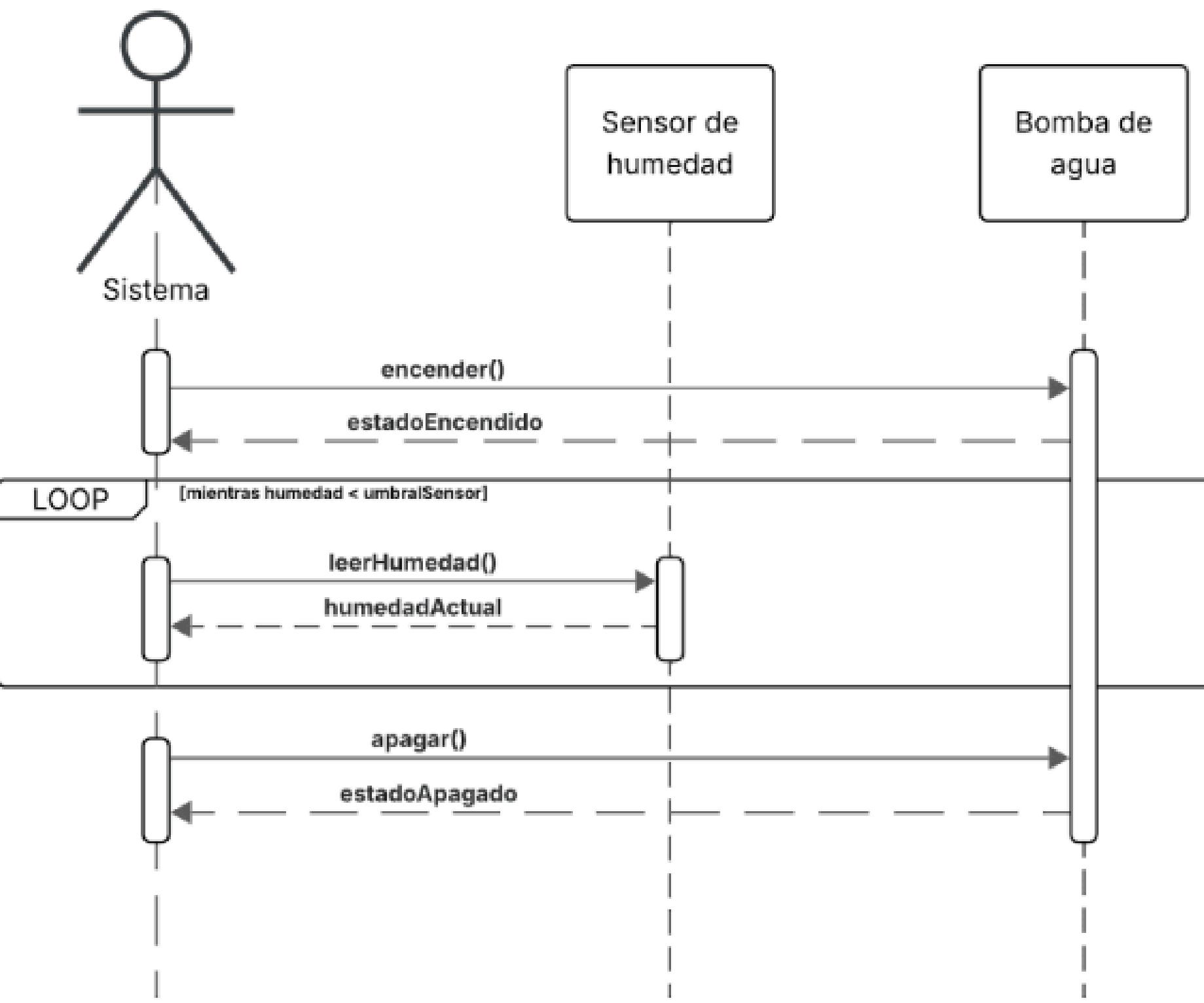
## DS Medir condiciones ambientales



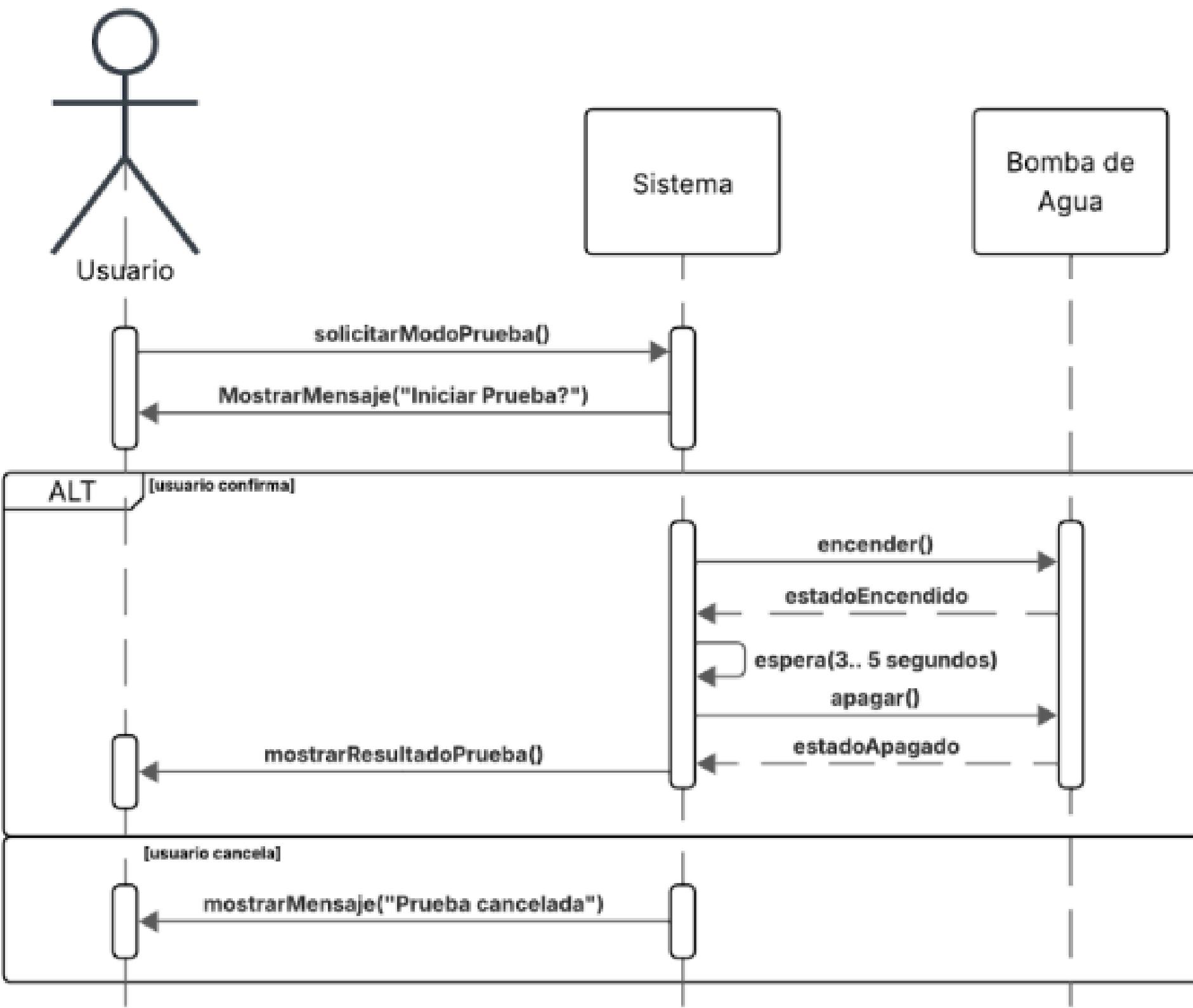
## DS Evaluar necesidad de riego



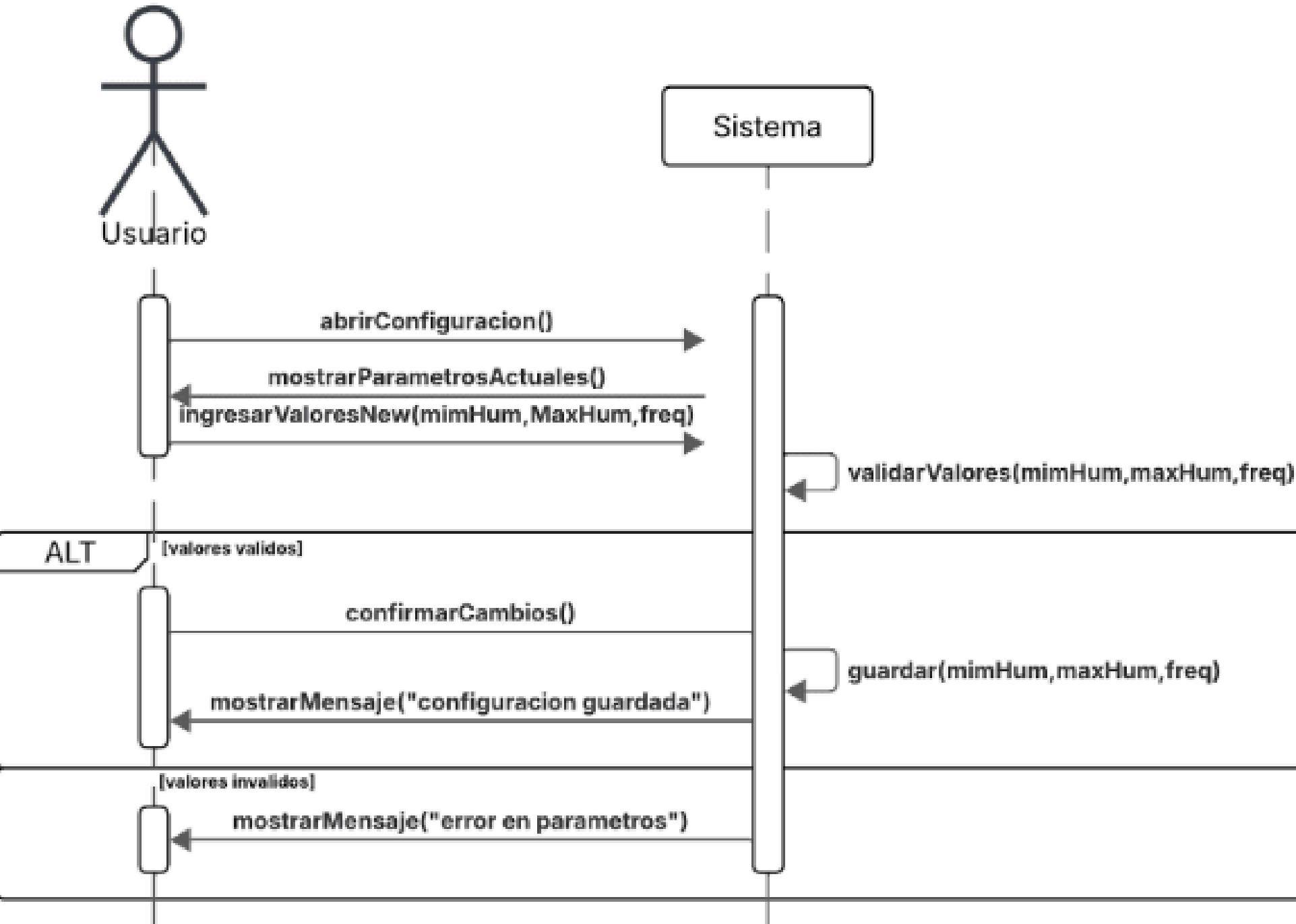
## DS Activar riego automático



## DS Ejecutar modo de prueba

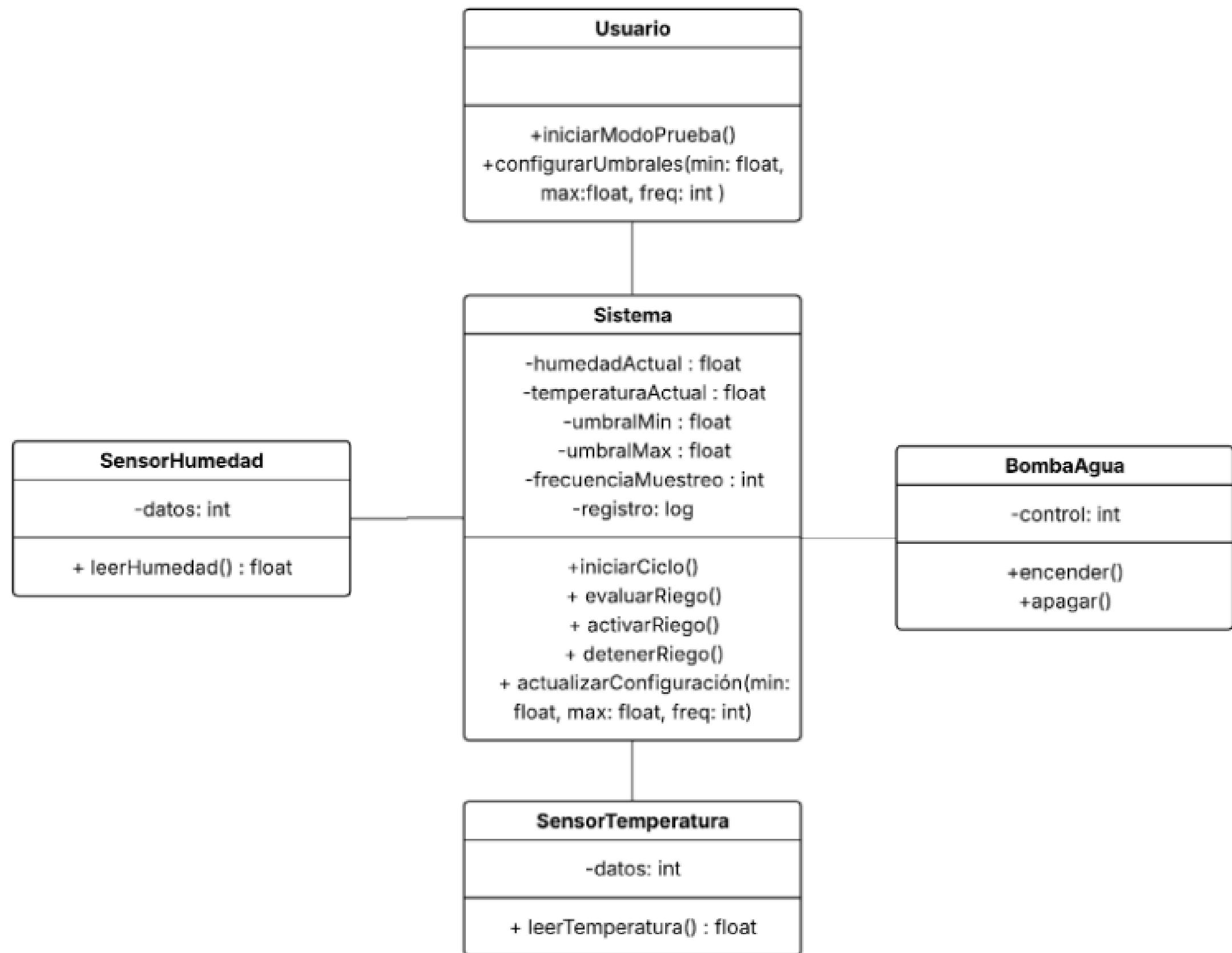


## DS Configurar umbrales y parámetros del sistema



# Diagrama de clases





# Diseño de la interfaz



# Interfaz

Sistema de riego automatizado

The interface consists of three main sections:

- Monitoreo Tiempo Real (Real-time Monitoring):** Displays soil moisture (45%) and ambient temperature (22°C).
- Configuración y Control (Configuration and Control):** Allows setting minimum (30) and maximum (60) thresholds, and includes "Guardar parámetros" (Save parameters) and "Activar Bomba" (Activate Pump) buttons.
- Historial de eventos (Event History):** Shows a log of events with columns for Date, Time, Event, and Detail.

Fecha	Hora	Evento	Detalle
28/10/2025	14:00	Lectura Sensor	H: 45% - T: 22°C
28/10/2025	13:45	Riego Finalizado	Alcanzó 60%
28/10/2025	13:40	Riego Iniciado	Detectado < 30%



## 1.- Monitoreo en Tiempo Real

Su función es mostrarle al usuario qué está pasando ahora mismo en el huerto.

Mostramos la Humedad del suelo y la Temperatura ambiente.

En la parte inferior, el indicador de texto ("ESPERANDO...") le dice al usuario si la bomba está apagada (esperando) o si está regando activamente en ese momento.

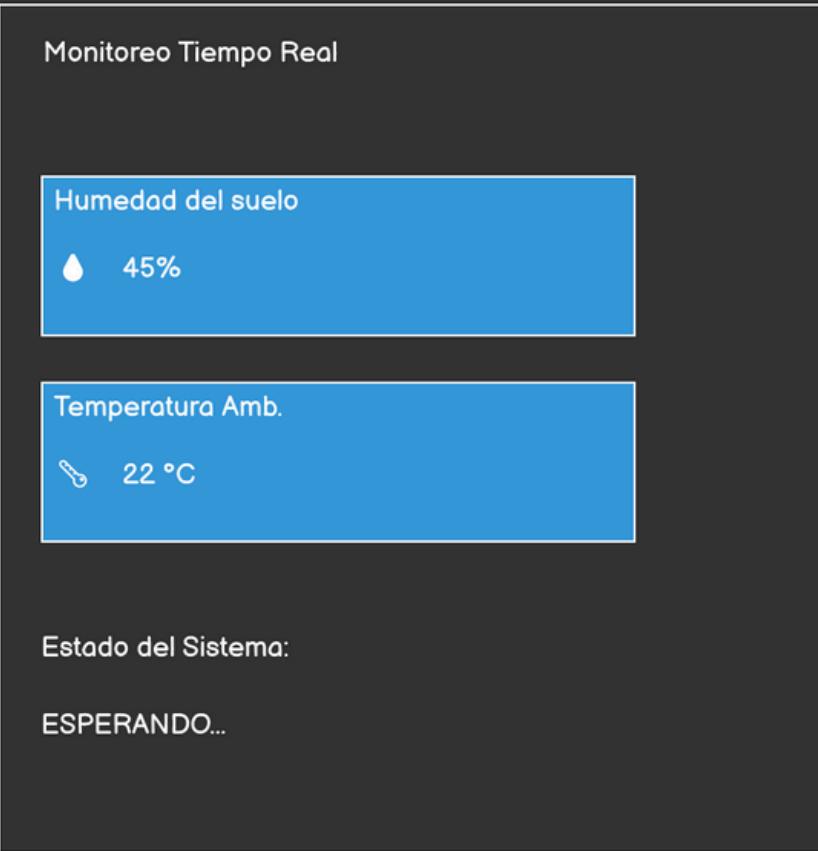
## 2.- Configuración y Control

Es donde el usuario define las reglas (automatización) y realiza pruebas manuales.

- Umbral mínimo:** Si la humedad baja de este porcentaje, el sistema activa la bomba automáticamente.
- Umbral máximo:** Cuando la humedad llega a este porcentaje, la bomba se apaga para no ahogar la planta.
- Activar bomba:** El usuario enciende la bomba manualmente por unos segundos para verificar que el hardware funciona.

# Interfaz

Sistema de riego automatizado



Configuración y Control

Umbral Mínimo (%):  
30

Umbral Máximo (%):  
60

Guardar parámetros

Activar Bomba

Historial de eventos

Fecha	Hora	Evento	Detalle
28/10/2025	14:00	Lectura Sensor	H: 45% - T: 22°C
28/10/2025	13:45	Riego Finalizado	Alcanzó 60%
28/10/2025	13:40	Riego Iniciado	Detectado < 30%



## 3.- Historial de eventos

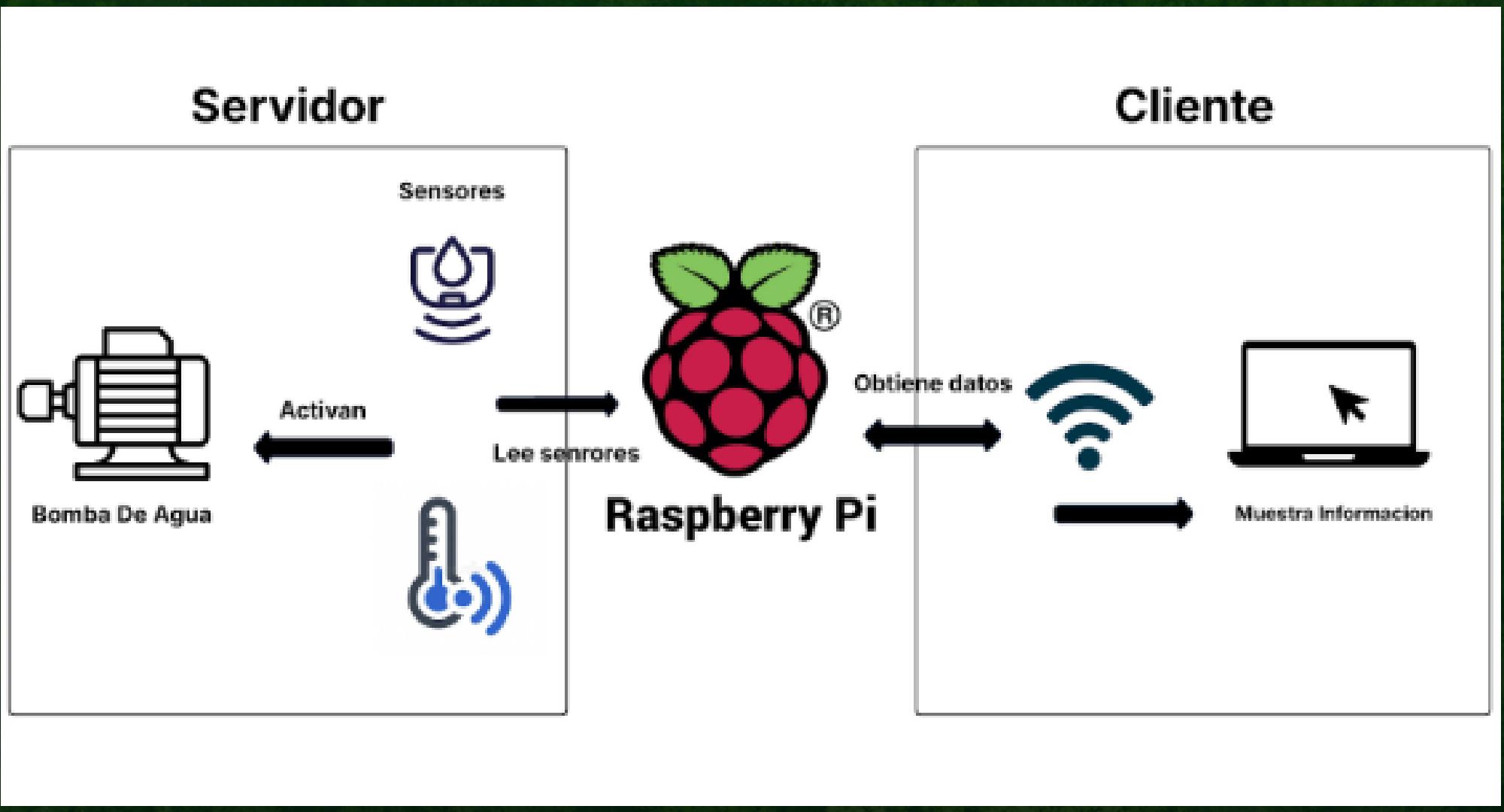
Permite visualizar las acciones pasadas.

- **Registro de Datos:** Como el sistema es automático, el usuario no siempre estará mirando la pantalla. Esta tabla registra cada vez que el sistema tomó una decisión
- **Detalle de las filas:** Podemos ver, por ejemplo, que a las 13:40 se inició un riego porque se detectó menos del 30% de humedad, y a las 13:45 finalizó al alcanzar la meta del 60%.

# **Descripción de la arquitectura**



# Arquitectura



## 1.- Lado servidor (izquierda)

Este bloque representa el **Hardware y el Control Físico** que está instalado en el huerto.

- **Sensores (Entrada):** La Raspberry Pi lee sensores constantemente. Recibe datos de humedad del suelo y temperatura ambiente.
- **Bomba de Agua (Salida/Actuador):** Basándose en las lecturas de los sensores, la Raspberry Pi envía una señal ("Activan") para encender la bomba cuando hace falta regar.

## 2.- El canal de Comunicación (centro)

- **Wi-Fi / Red:** La flecha con el símbolo de Wi-Fi indica que la Raspberry Pi y el Computador no necesitan estar conectados por cables. Se comunican de forma inalámbrica. La Raspberry Pi envía los datos recolectados hacia el cliente.

## 3.- El lado cliente (derecha)

- **Notebook / Dispositivo:** Es el computador desde donde el usuario supervisa el huerto.
- **Obtiene datos:** El cliente solicita la información actual al servidor (Raspberry Pi).
- **Muestra Información:** El software del cliente procesa esos datos y los presenta en la pantalla.

# Herramientas



- **Visual Studio Code:** Entorno principal para escribir código
- **Python:** Elegido por su compatibilidad con Raspberry Pi para controlar sensores.
- **Raspberry Pi OS:** Sistema operativo encargado de gestionar los pines GPIO y la ejecución de scripts.
- **RPi.GPIO / Adafruit\_DHT:** Librerías específicas para la lectura precisa de sensores (DHT22).
- **Git y GitHub:** Para el control de cambios, trabajo colaborativo y respaldo del código fuente y documentación.
- **Lucidchart:** Utilizado para crear todos los diagramas UML (Casos de uso, Secuencia, Clases).



# Técnicas

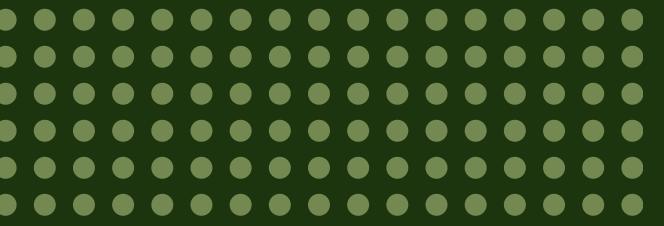
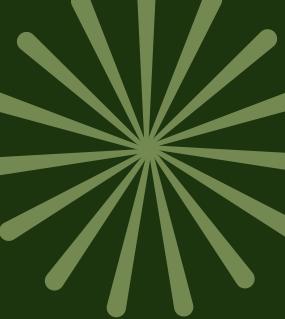


**Se aplicó la Técnica de Prototipado Iterativo, esta metodología se basa en ciclos de mejora continua en lugar de intentar hacer todo perfecto en un solo intento.**

**Se divide en tres pilares:**

- **Desarrollo Incremental de Módulos:** No construimos todo el sistema a la vez. Diseñamos y probamos cada parte por separado, primero el módulo de sensores, luego la lógica de decisión y finalmente el control de la bomba.
- **Pruebas y Retroalimentación Continua:** En cada iteración, evaluamos la precisión de las lecturas y la estabilidad del programa en el Raspberry Pi. Los resultados nos permiten ajustar los umbrales o el código antes de avanzar.
- **Documentación Constante:** Cada cambio en los diagramas o solución de problemas queda registrado.





# Conclusión

Podemos afirmar que el equipo ha consolidado los conocimientos necesarios para llevar a cabo esta fase del proyecto.

A través de la elaboración de los diagramas UML y la arquitectura del sistema, logramos estructurar claramente cómo interactuarán los sensores, la lógica de control y la interfaz de usuario, evitando improvisaciones futuras, este enfoque nos permitió fortalecer nuestras habilidades.

La siguiente fase será crítica, ya que llevaremos estos diseños a la realidad mediante la integración física de los componentes, la programación del código en Python y, lo más importante, la calibración y pruebas reales en la maqueta para asegurar que el sistema responda eficientemente a las necesidades del riego.



**iGRACIAS!**

