

# Prevención de efectos de la humedad

## Presentación de informe 2

Hilda Albarracín - Mayling Álvarez - Antonella Butrón - Ana Gutiérrez

# Introducción

- La humedad en ambientes cerrados puede generar problemas como moho, corrosión y aparición de sales en las estructuras.
- Para enfrentar este problema, se plantea el diseño de un sistema de prevención y control de la humedad.

# OBJETIVOS General

Diseñar e implementar un sistema de prevención y control de la humedad en el ambiente con el fin de minimizar sus efectos negativos sobre las estructuras dentro de un área cerrada.

## Específicos

1. **Investigar** sobre la humedad y sus efectos.
2. **Analizar** los distintos tipos de sensores y actuadores disponibles.
3. **Diseñar** la arquitectura del sistema de prevención y control.
4. **Implementar** y programar el sistema utilizando Raspberry Pi.
5. **Probar** y evaluar el funcionamiento del sistema en un entorno controlado.

Situación

Objetivos

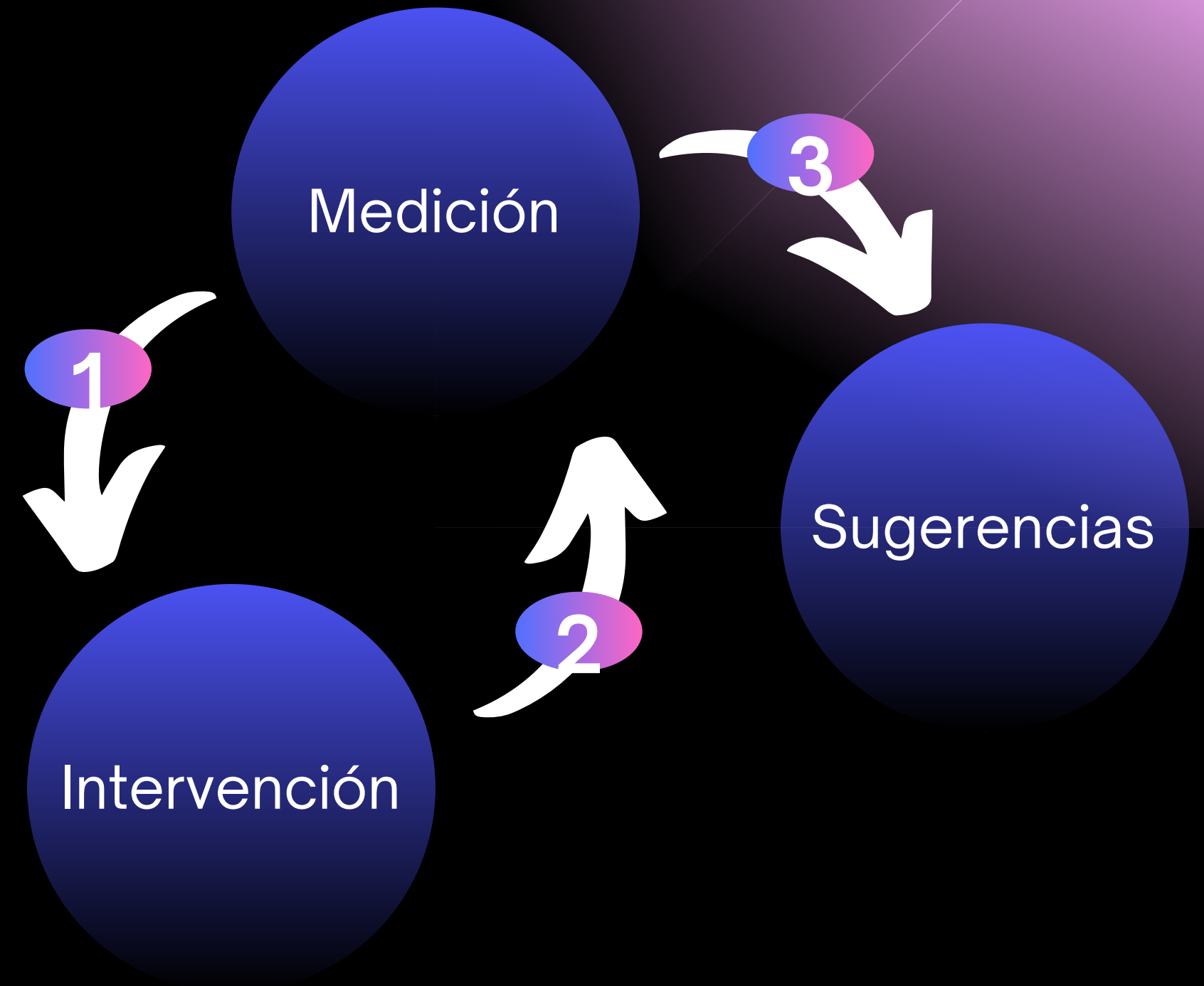
Justificación

Alcance

Solución



# Solución METODOLOGÍA





# REQUISITOS FUNCIONALES

- **Monitoreo de la humedad, el moho y la sal:**

- RF01: El sistema debe medir la humedad del ambiente interno y compararlo con el umbral configurado.
- RF02: El sistema debe mostrar los niveles de humedad y sal medidos en la habitación en la interfaz de usuario.
- RF03: Si la humedad excede el umbral configurado, el sistema debe encender el deshumidificador para la detección de sal en el ambiente.
- RF06: Si la humedad deja de exceder el umbral configurado, el sistema debe apagar el deshumidificador.}

- **Control de la humedad:**

- RF09: Si es que el usuario no desactiva o pospone las medidas de control, el sistema debe activar automáticamente los ventiladores y la luz UV.
- RF11: El sistema debe identificar si el nivel de sal sigue siendo alto después de aplicar medidas de control, comparando el valor actual con un umbral predefinido.
- RF12: Si los niveles de sal y humedad siguen siendo altos, el sistema debe avisar el nivel de gravedad al usuario a través de la interfaz.
- RF13: Si los niveles de sal y humedad siguen siendo altos, el sistema debe mostrar recomendaciones ,sobre las medidas de acción a tomar, al usuario a través de la interfaz.

# REQUISITOS funcionales

- Modo de envío de datos:
  - RF14: El sistema debe enviar los datos a la aplicación móvil mediante conexión Wi-Fi.
- Almacenamiento de información en la base de datos:
  - RF14: El sistema debe almacenar los datos registrados en una base de datos (local o en la nube).
- Interfaz de usuario:
  - RF15: La interfaz debe mostrar en una sola pantalla el estado del sistema, las mediciones de humedad y sal.
  - RF16: La interfaz debe mostrar alertas cuando se detecten altos niveles de sal, permitiendo al usuario desactivar o posponer las medidas de control.
  - RF17: La interfaz debe permitir al usuario consultar el historial de lecturas registradas.

# REQUISITOS no funcionales

- **Rendimiento:**

- RNF01: El sistema debe medir la humedad del ambiente cada 5 segundos.
- RNF02: Las alertas o notificaciones deben mostrarse en la interfaz en menos de 2 segundos tras su detección.

- **Usabilidad:**

- RNF03: La aplicación debe tener una interfaz clara e intuitiva.

- **Confiabilidad:**

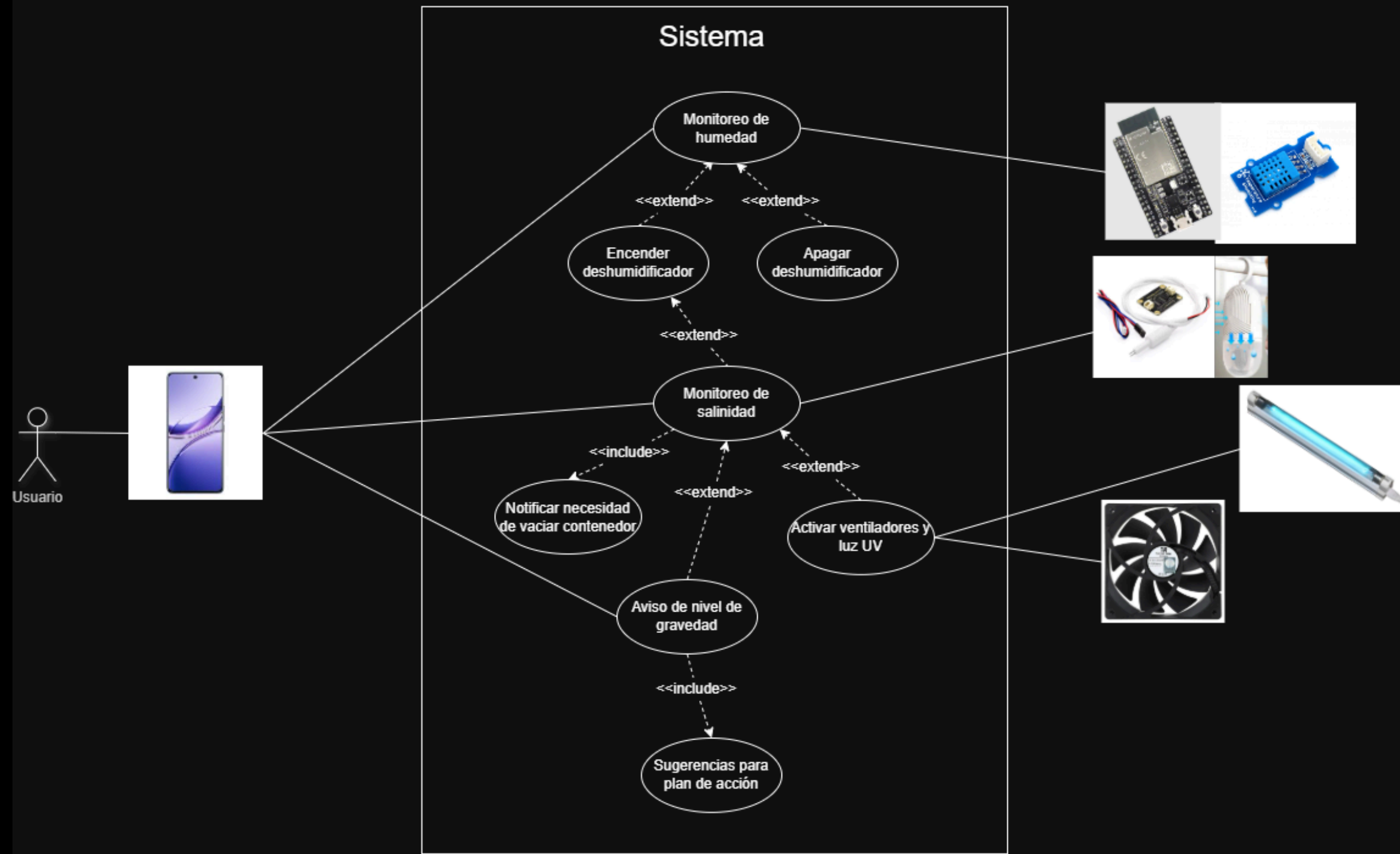
- RNF04: El sistema debe estar operativo la mayoría del tiempo para garantizar el funcionamiento continuo y estable.
- RNF05: El historial de datos debe respaldarse periódicamente para evitar pérdidas.

- **Seguridad:**

- RNF06: La comunicación entre el Raspberry Pi 4B y la aplicación debe realizarse mediante un protocolo seguro.



# Diagrama Casos de Uso

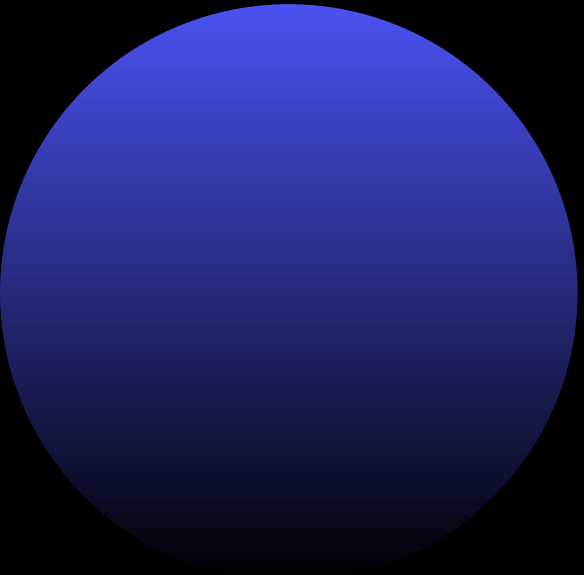




# Casos de Uso

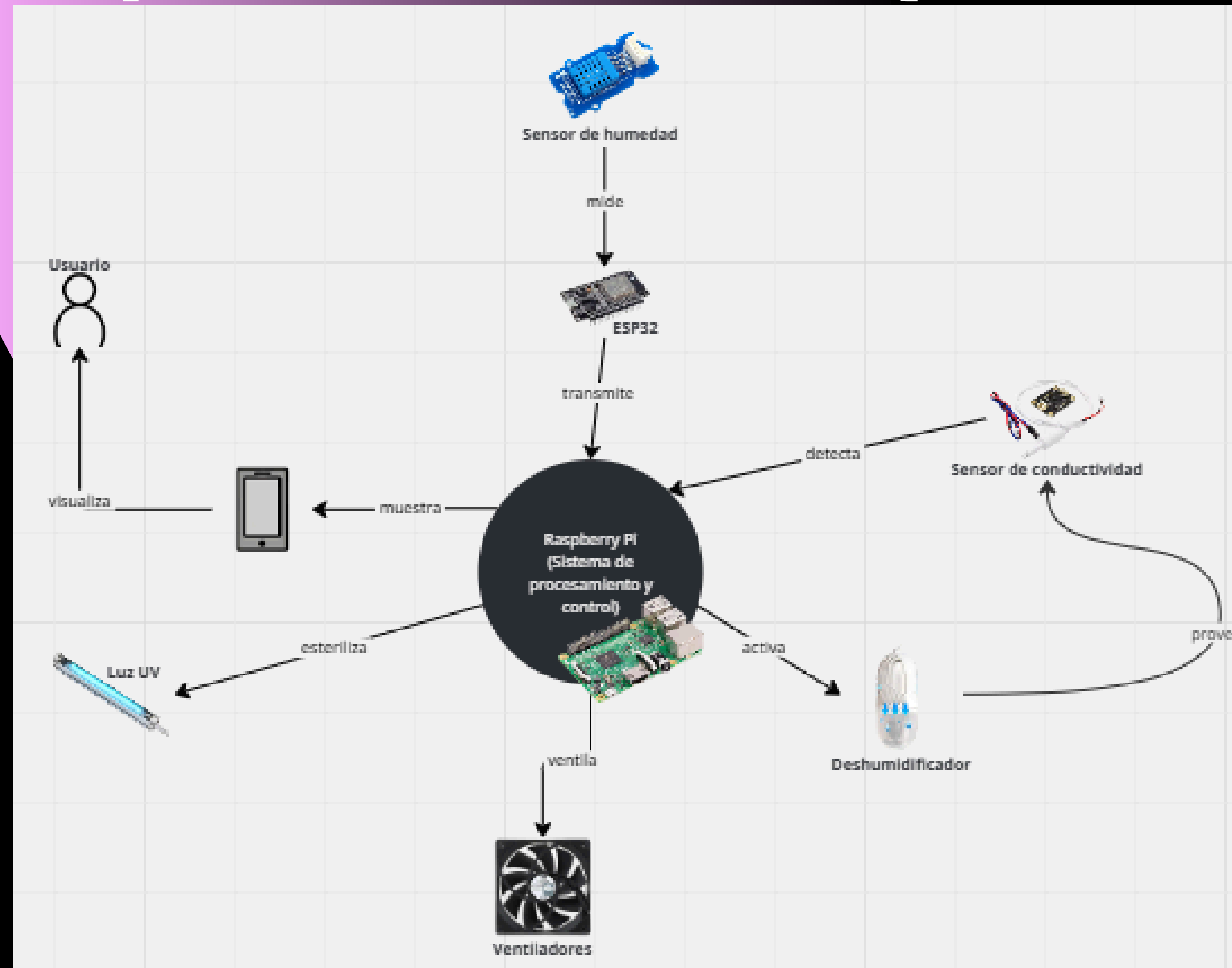
## Caso de Uso: Monitoreo de humedad

ID: CU-001	
Nombre del CU: Monitoreo de humedad	
Actor(es): Sensor de humedad, usuario	
Requisitos Relacionados: RF02	
Descripción: El sistema recibe periódicamente las mediciones de humedad desde el sensor, las procesa y las muestra al usuario en la aplicación móvil. Si la humedad supera el umbral definido, se prende el deshumidificador. De lo contrario lo apaga.	
Precondiciones: El sensor de humedad debe estar conectado al Raspberry Pi 4B	
Flujo Principal: Usuario	Flujo Principal: Sistema
1. Ingresar a la sección de monitoreo.	2. Reúne datos del sensor de humedad.
4. Visualiza el nivel de humedad.	3. Muestra los datos en la interfaz web.
	4. Evalúa si la humedad está dentro del rango normal.
	5. [Extend: CU-002] Encender deshumidificador. (Opcional)
	6. Guarda la lectura en la base de datos.
Flujo Principal: Usuario	Flujo Principal: Sistema
	5.1. [Extend: CU-003] Apagar deshumidificador. (Opcional)
	6.1. Guarda la lectura en la base de datos.
Flujo Alternativo: Usuario	Flujo Alternativo: Sistema



# DESCRIPCIÓN DE LA ARQUITECTURA

# Descripción de la Arquitectura





# ARQUITECTURA

El sistema se organiza en tres niveles principales:

## Nivel físico:

- Sensor de humedad.
- Sensor de conductividad (salinidad).
- Deshumidificador.
- Ventilación o ventilador.
- Luz UV.
- Microcontrolador (ESP32).
- Microprocesador.

## Nivel lógico:

- Módulo de sensores.
- Módulo de control.
- Módulo de procesamiento.
- Módulo de recomendaciones.
- Módulo de alertas.

## Nivel de presentación:

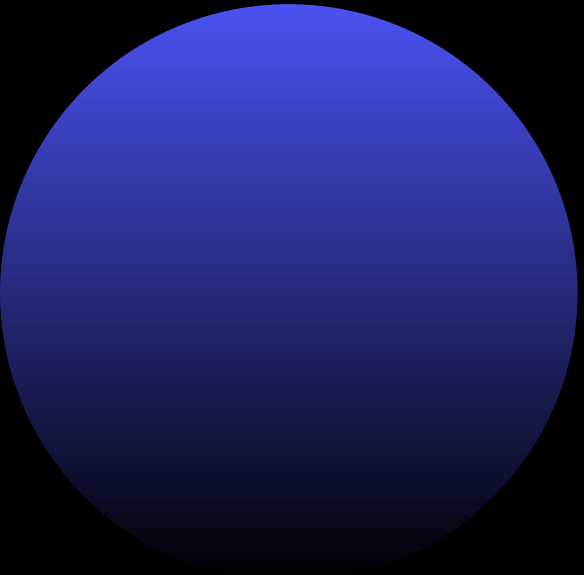
- Mostrar mediciones de humedad y salinidad.
- Indicar estados del sistema.
- Presentar alertas y notificaciones.
- Permitir al usuario recibir sugerencias para acciones correctivas.
- Diseño móvil responsive para fácil acceso.



# ARQUITECTURA

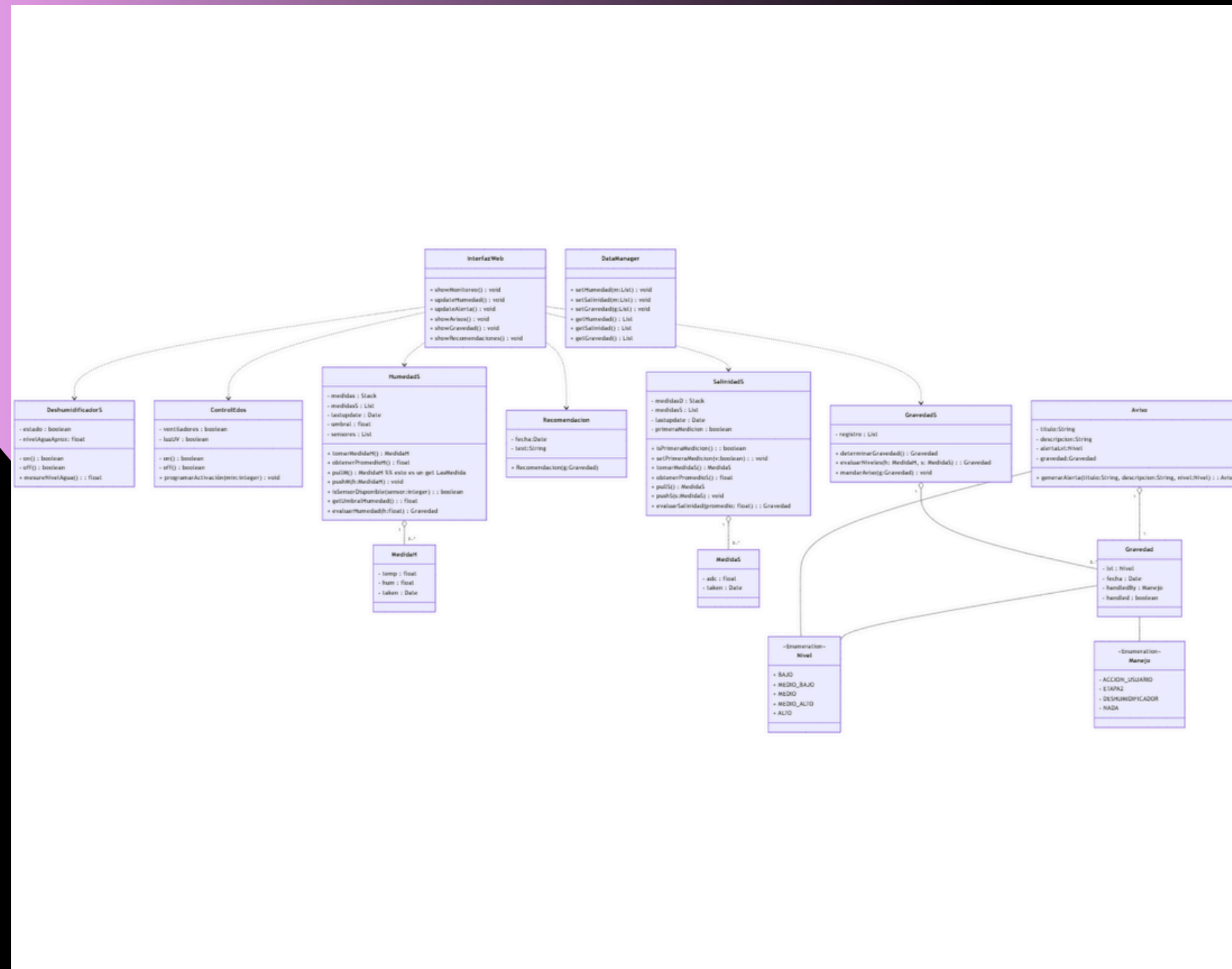
## Arquitectura de integración de los componentes:

1. Los sensores de humedad capturan datos
2. El ESP32 procesa la humedad y envía los datos al microprocesador
3. La Raspberry Pi recibe la información y procesa la salinidad desde la conductividad en el sensor
4. El backend procesa la información, compara la tabla de gravedad y determina si se necesita la activación de actuadores, mandando la señal al microprocesador.
5. La interfaz web presenta visualización, alertas y sugerencias
6. El usuario recibe notificaciones y puede tomar acciones basadas en lo mostrado.

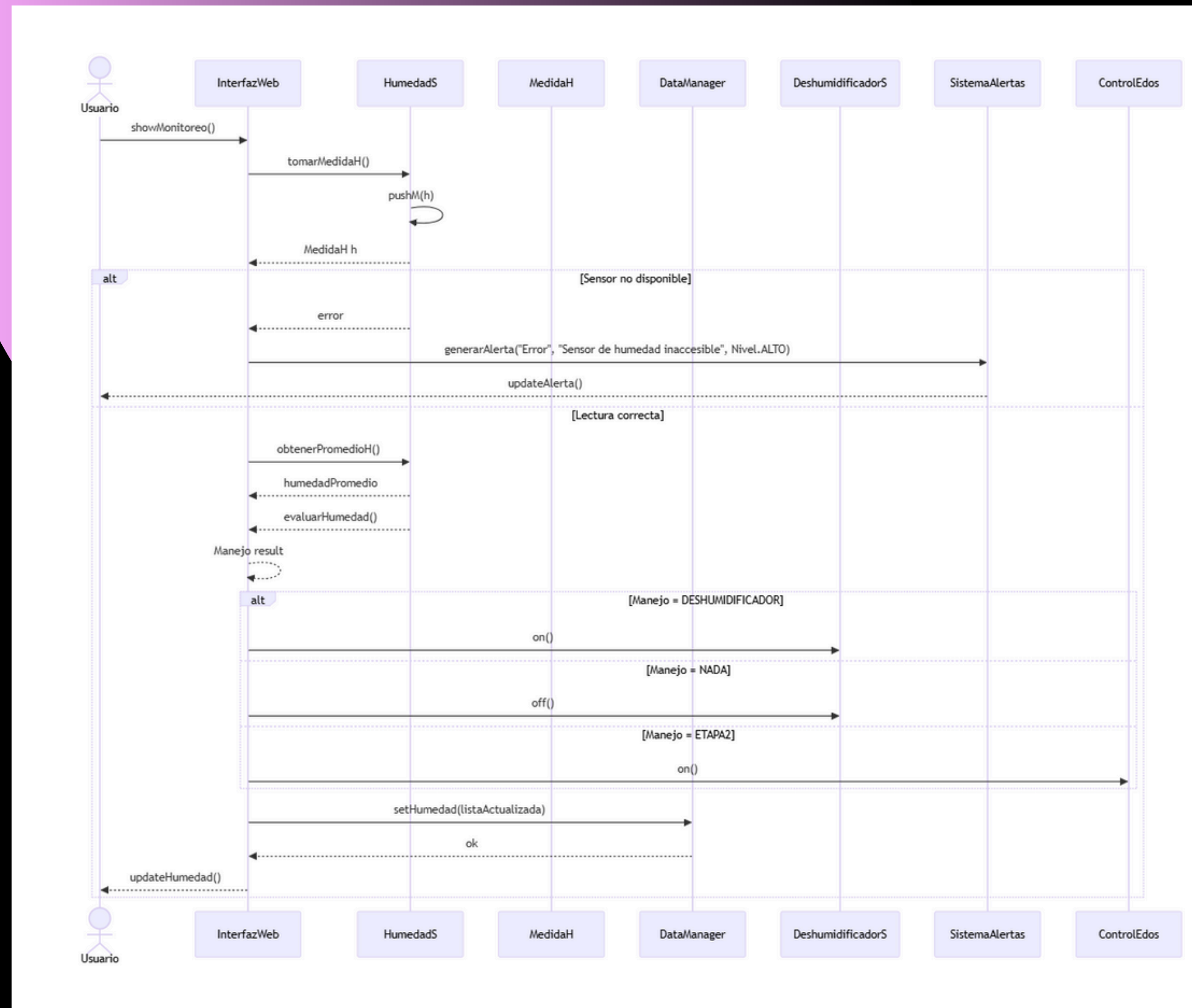


# DIAGRAMA DE CLASES Y DIAGRAMAS DE SECUENCIA

# Diagrama de clases

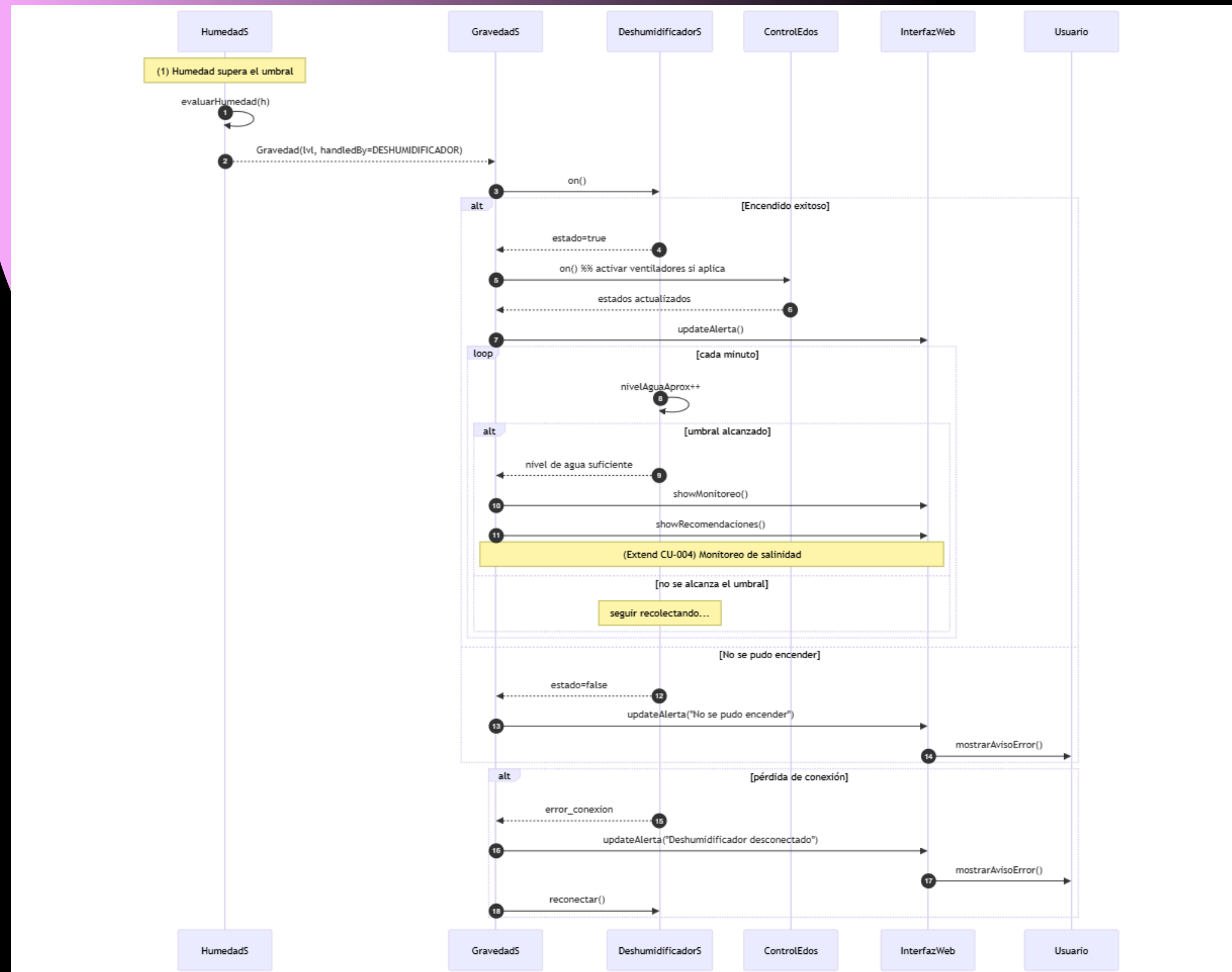


# CU-001 (Monitoreo de humedad)

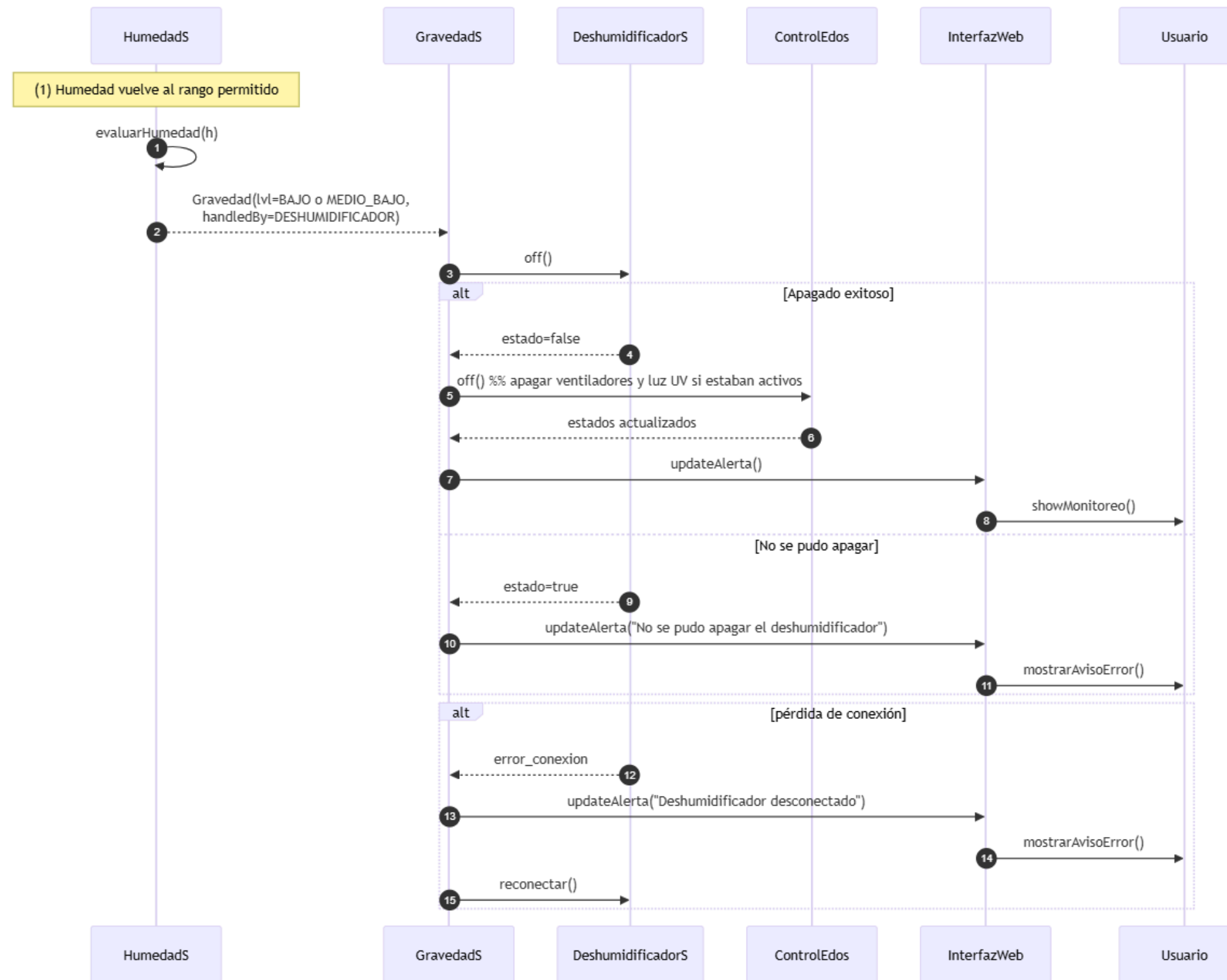




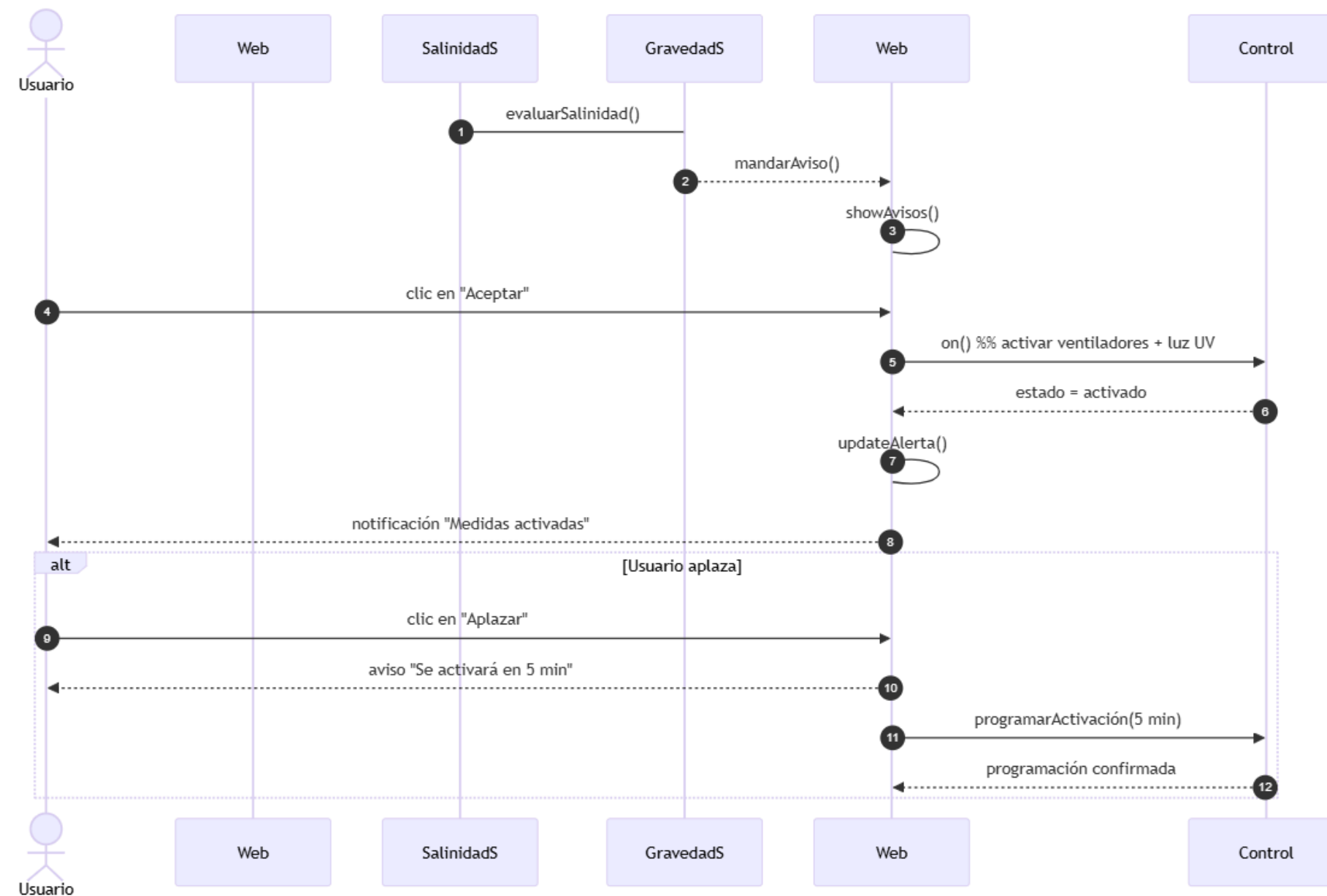
# CU-002 (Encender deshumidificador)



# CU-004 (Monitoreo de salinidad)



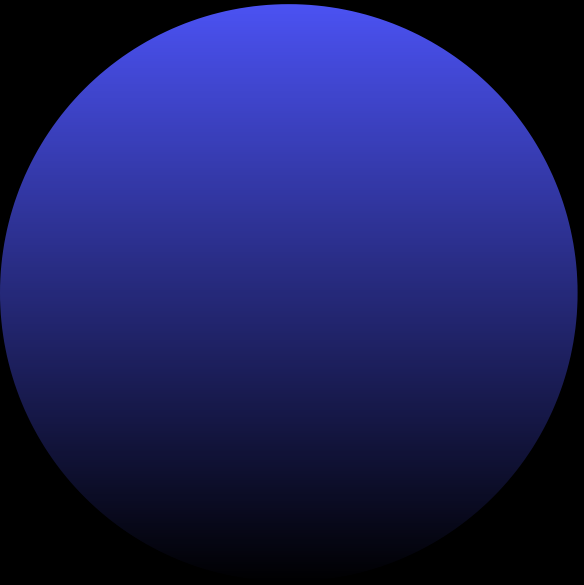
# CU-006 Activar ventiladores y luz UV



# INTERFAZ







# CORRECCIONES

# Costos Hardware

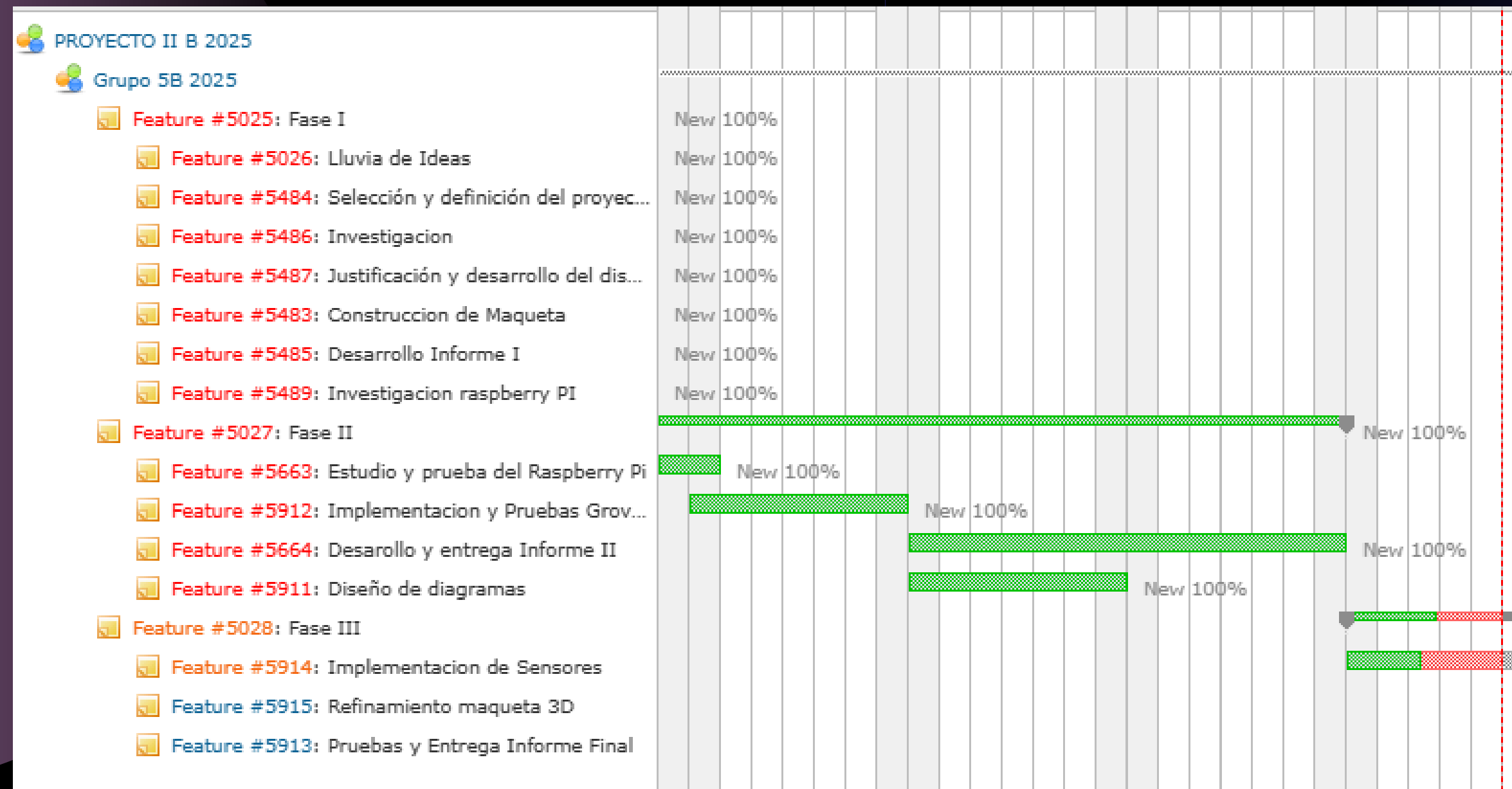
Hardware	Cantidad	Costo
GrovePi+ Starter Kit for Raspberry Pi	1	\$243.939
ESP32	1	\$10.000
ProtoBoard	2	\$6.000
Deshumidificador	1	\$6.590
Sensor de conductividad	1	\$3.000
Raspberry PI 4 Model B/ 8GB RAM	1	\$164.990
Ventiladores de extracción	1	\$9.000
Luz UV	1	\$7.000
HP <u>Pavillion</u> Plus (Costo de Uso)	1	\$120.000
MSI Katana <u>GF66</u> (Costo de Uso)	1	\$120.000
Acer (Costo de Uso) ▼	1	\$60.000
Lenovo Thinkpad <u>x390</u> yoga (Costo de Uso)	1	\$ 120.000
SD 32 GB con adaptador	1	\$8.396
Total:	14	\$878.915

(Activos preexistentes del equipo)

# Planificación de recursos humanos

Rol	Cantidad	Sueldo/Hora	Horas Totales	Costo
Jefe	2	\$25.000	4.5	\$225.000
Ensamblador	2	\$12.000	10	\$240.000
Programador	2	\$18.000	12	\$432.000
Documentador	3	\$10.000	6	\$180.000
Diseñador	2	\$15.000	8	\$240.000
Total:	11	-	-	\$1.317.000

# Carta Gantt





# Conclusión

Se pudo evidenciar que el uso de una planificación efectiva nos permitió desarrollar con facilidad la segunda etapa. Y nos da mas facilidad para terminar el proyecto. Entregando una visual más profunda de cada procedimiento.

# Muchas gracias

Hilda Albarracín - Mayling Álvarez - Antonella Butrón - Ana Gutierrez