



UNIVERSIDAD DE TARAPACÁ  
*Universidad del Estado*

Ingeniería@  
Computación e Informática

# BIN RAIDERS FASE 2

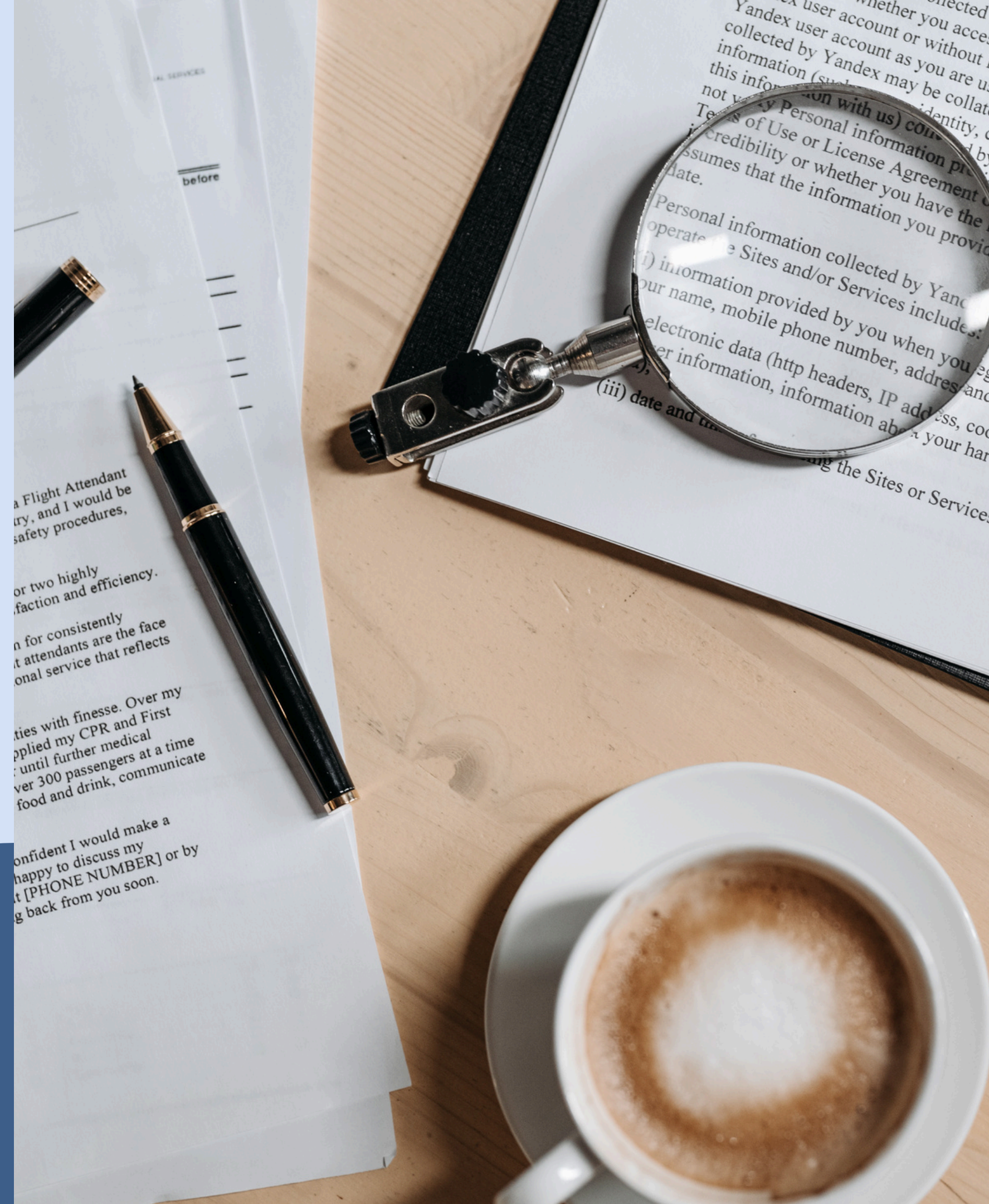


**Integrantes:** Jhon Alarcón  
André Guerra  
Benjamín Gómez  
Nelson Ramirez  
**Académico:** Diego Aracena  
**Asignatura:** Proyecto II



# Índice

- Introducción
- Objetivos
- Correcciones
- Modelos de diseño
- Secuencias de pantallas
- Diagramas
- Conclusión





# Introducción

Este segundo avance presenta el diseño técnico del proyecto Bin Raider, incluyendo:

- Objetivos
- Correcciones Fase 1
- Requerimientos
- Casos de uso
- Diagramas del sistema
- Secuencia de pantallas



# Resumen

**Problemática:** Los contenedores de basura públicos suelen desbordarse y esta situación genera acumulación de desechos, malos olores y contaminación.

**Solución:** Sistema de automatización de basureros conectado a una aplicación móvil mediante una Raspberry Pi. El sistema notificará cuando un contenedor esté lleno, utilizando sensores ultrasónicos y una cámara integrada,







# Objetivo general

Diseñar e implementar un sistema de gestión inteligente de residuos basado en IoT utilizando una Raspberry Pi 4B, que permita el monitoreo en tiempo real del nivel de llenado de los contenedores para optimizar el proceso de recolección.



A top-down view of a workspace. On the left, a silver laptop is partially visible. In the center, a white notebook with horizontal lines is open, with an orange pen resting on it. To the right of the notebook is a black stapler. In the bottom right corner, a white cup of coffee sits on a saucer. The background is a solid light blue color with a dark blue abstract shape in the top right corner.

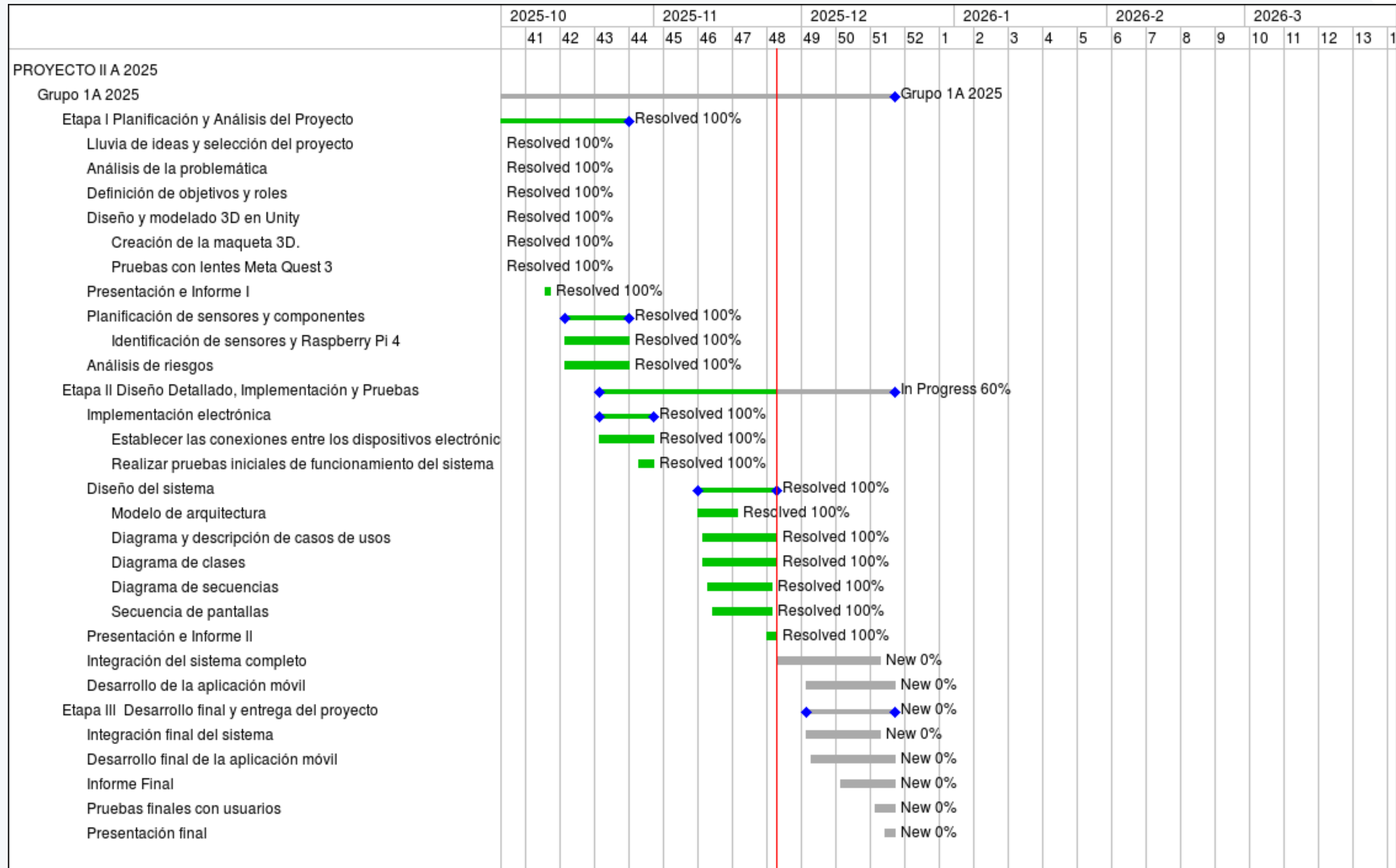
# Objetivos específicos

- **Definir** cómo se medirá el nivel de llenado de los contenedores usando sensores ultrasónicos.
- **Establecer** las funciones principales que tendrá la aplicación móvil para mostrar el estado del contenedor.
- **Incorporar** una cámara que permita revisar visualmente el interior del contenedor cuando sea necesario.
- **Definir** cómo se enviará y recibirá información entre la Raspberry Pi y la aplicación móvil.
- **Representar** el sistema completo mediante una maqueta que muestre cómo funciona el monitoreo inteligente.

“

**Correcciones**

# Correcciones





# Correcciones

## Costos de Hardware

Recursos	Cantidad	Costo
Raspberry Pi 4	1	\$142.990
Notebook (arriendo mensual)	4	\$60.000/mes x 3.75 = \$900.000
SmartPhone	1	\$450.000
Sensor Ultrasónico	1	\$2.540
Sensor de cámara	1	\$15.366
Otros suministros (cables, adaptadores, tarjeta sd, etc.)	*	\$3.000
Total	7	\$1.513.896

## 4.1.2. Planificación de recursos humanos.

Rol	Tarifa por hora	Horas de trabajo	Sueldo
Jefe de proyecto	\$18.000/hora	5 horas x semana 15 x 5 = 75 horas	\$1.350.000
Programador	\$14.000/hora	5 horas x semana 15 x 5 = 75 horas	\$1.050.000
Diseñador	\$10.000/hora	5 horas x semana 15 x 5 = 75 horas	\$750.000
Documentador	\$8.000/hora	5 horas x semana 15 x 5 = 75 horas	\$ 600.000
Total			\$3.750.000

# Requerimientos Funcionales

## **RF-01 Medir nivel del basurero:**

El sensor ultrasónico debe medir el nivel de basura dentro del contenedor y enviar el dato al sistema.

## **RF-02 Detectar contenedor lleno:**

El sistema debe identificar cuando el nivel de basura supera el límite establecido y marcar el contenedor como "Lleno".

## **RF-03 Enviar alerta de llenado:**

Cuando un contenedor está lleno, el sistema debe enviar una notificación a la aplicación móvil.

## **RF-04 Visualizar estado del basurero:**

La aplicación móvil debe mostrar el estado actual del contenedor (Vacío, Medio o Lleno) en una interfaz sencilla.

## **RF-05 Visualizar cámara:**

La app debe permitir ver la cámara asociada al contenedor para comprobar su estado en tiempo real.

## **RF-06 Generar reporte de llenado:**

El sistema debe generar y almacenar reportes con el historial de niveles de llenado y alertas emitidas.

## **RF-07 Validar el acceso a una cámara.**

El sistema debe validar el acceso a la cámara de un contenedor que será seleccionado en la aplicación móvil.

## **RF-08 Iniciar Sesión:**

El sistema debe permitir al administrador acceder, ingresando usuario y contraseña válidos.



# Requerimientos no Funcionales

## **RNF-01 Disponibilidad:**

El sistema debe mantenerse operativo y accesible en todo momento para garantizar la comunicación entre los dispositivos.

## **RNF-02 Rendimiento:**

El sistema debe procesar las lecturas de los sensores y reflejar los cambios en la aplicación en menos de 2 segundos.

## **RNF-03 Usabilidad:**

La aplicación móvil debe tener una interfaz fácil de entender, con íconos y colores que representan claramente los estados del contenedor.

## **RNF-04 Seguridad:**

La comunicación entre la Raspberry Pi y la aplicación móvil debe estar protegida mediante autenticación .

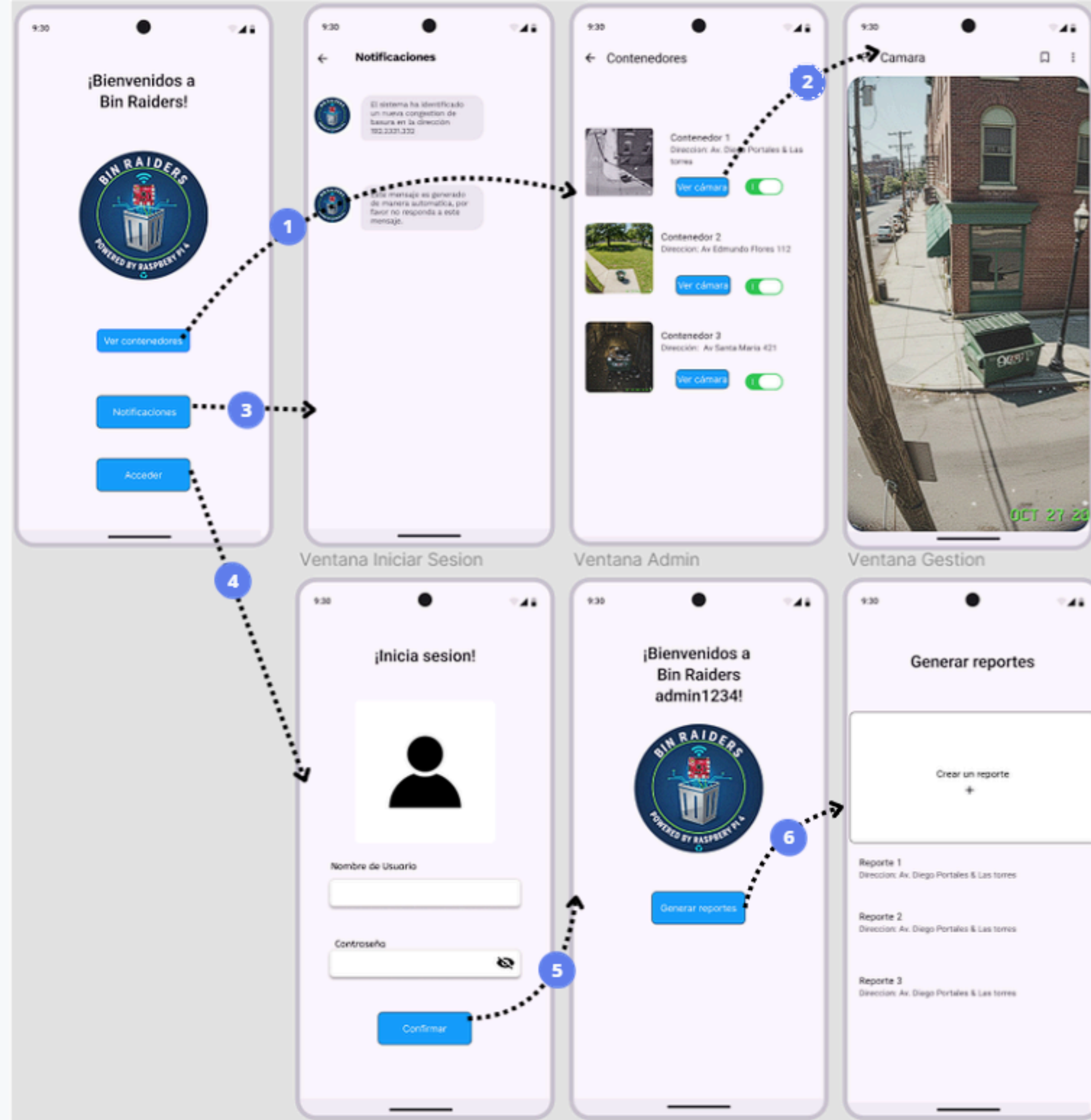
## **RNF-05 Mantenibilidad:**

El software debe estar estructurado en módulos independientes para facilitar futuras actualizaciones o reemplazo de componentes.

## **RNF-06 Compatibilidad:**

La aplicación móvil debe ser compatible con dispositivos Android y comunicarse correctamente con la Raspberry Pi 4B.

# Secuencia de pantallas(IGU)

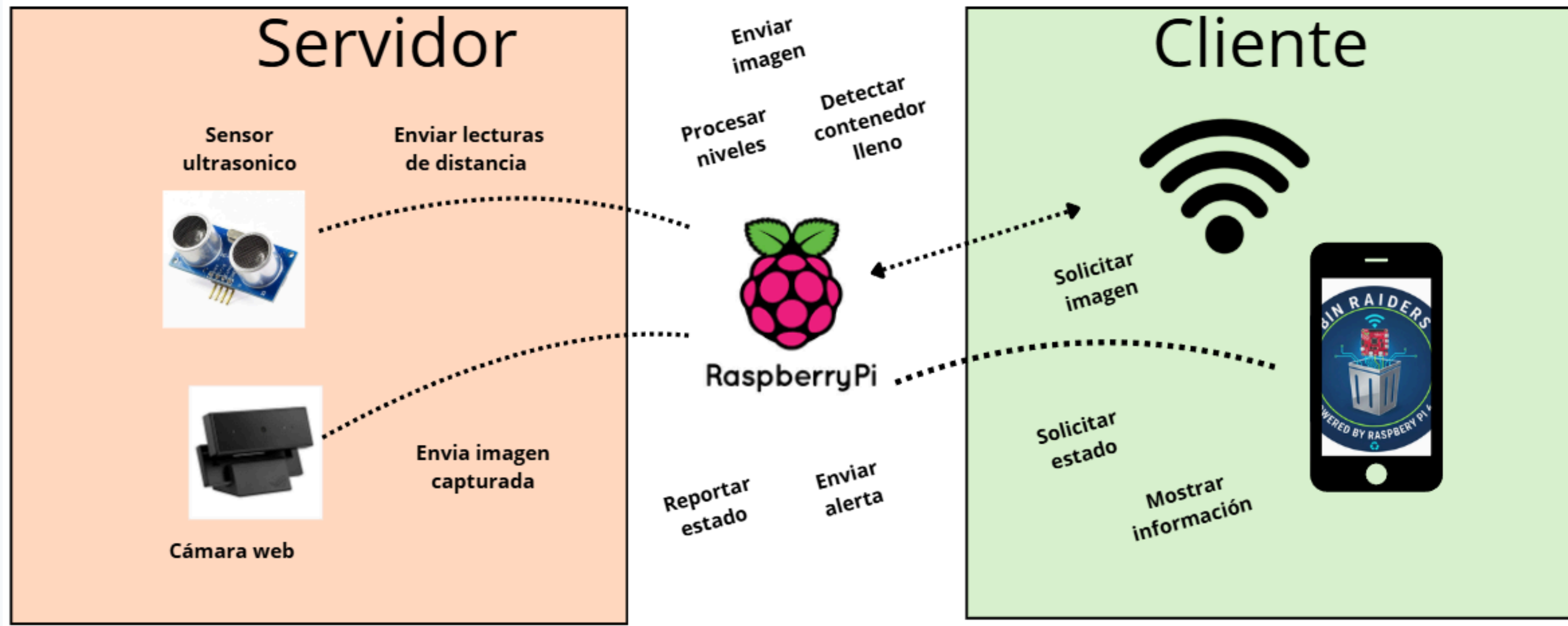




“

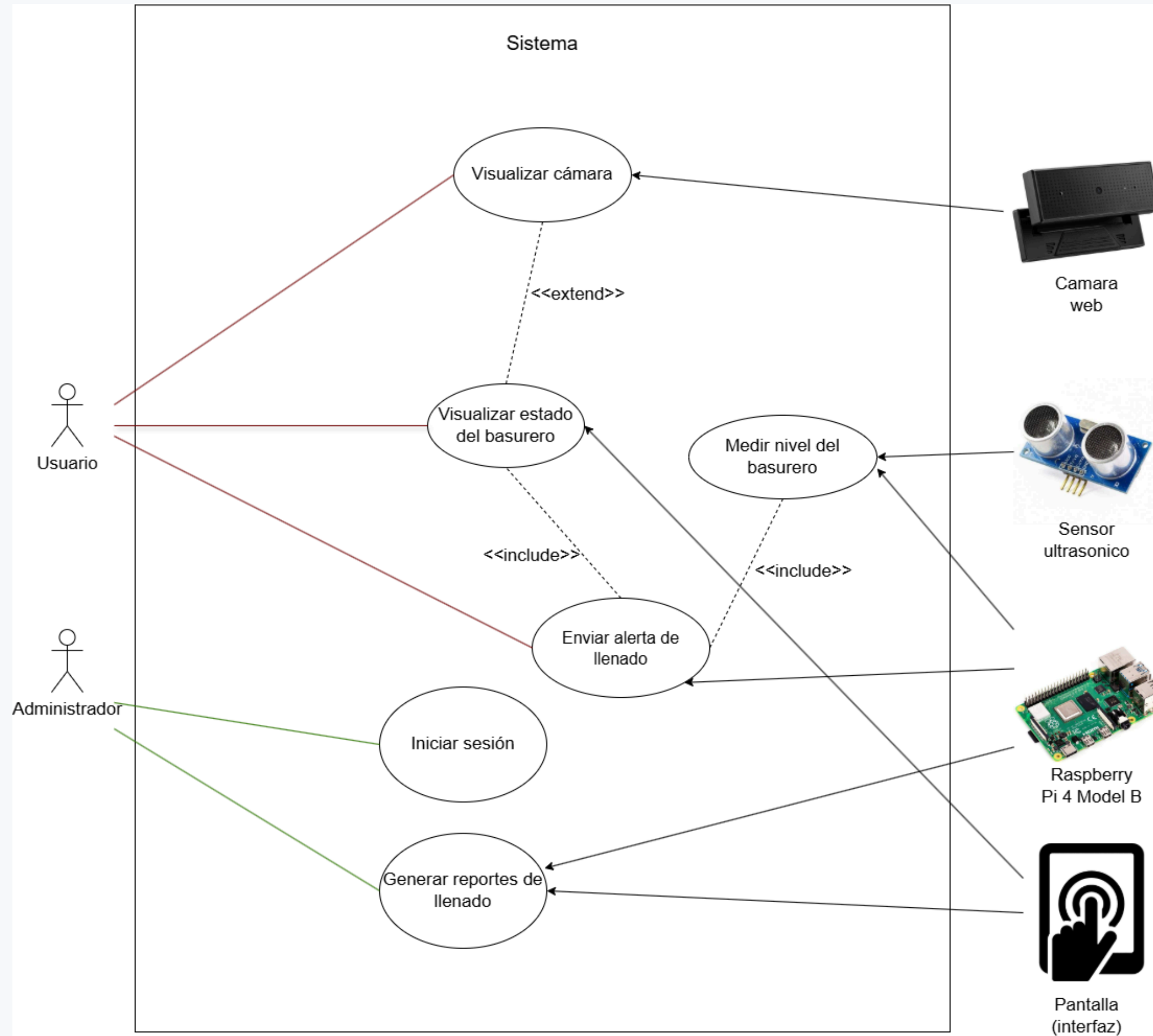
**Diagrams**

# Diagrama de Arquitectura

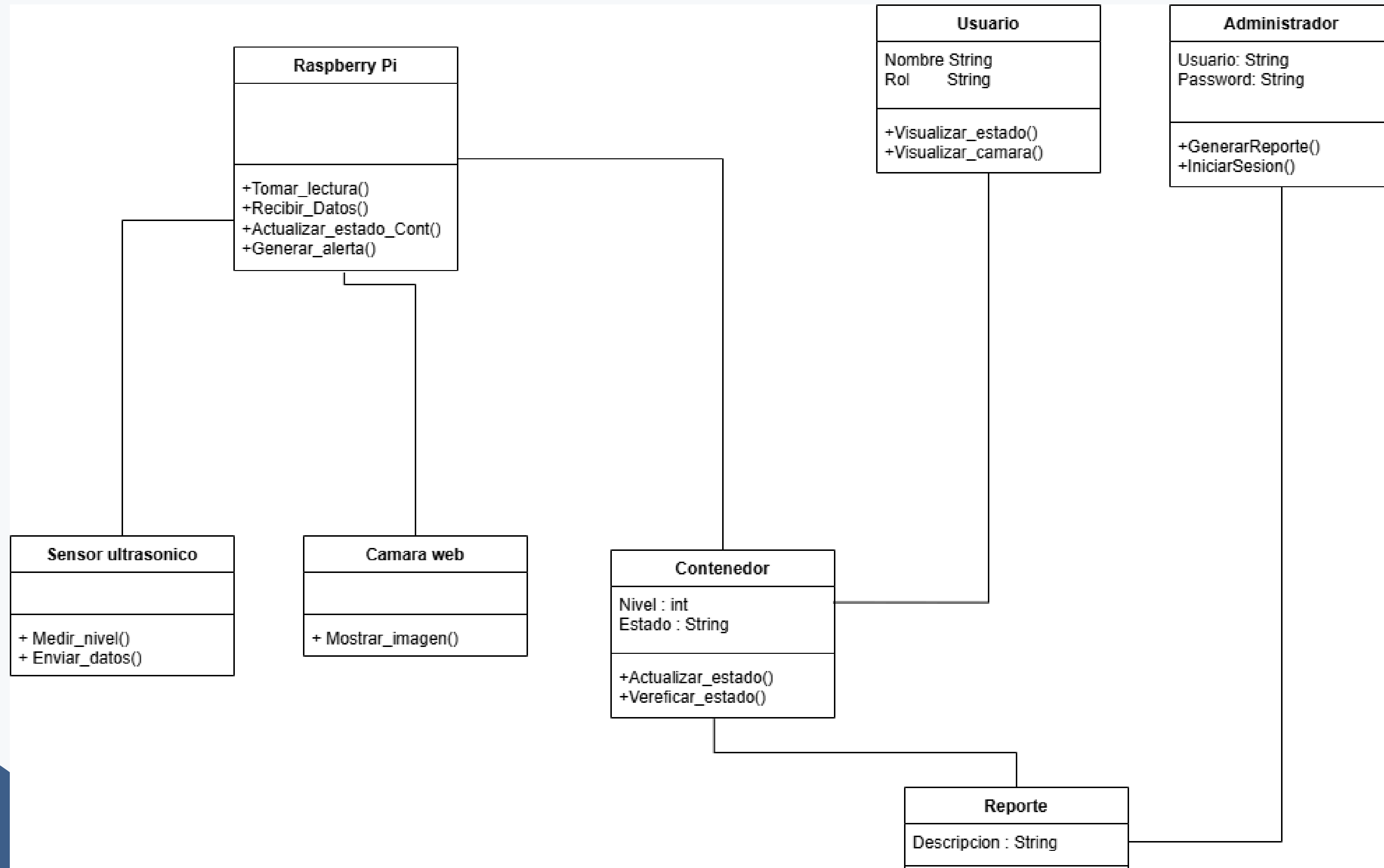




# Diagrama de casos de usos



# Diagrama de clases





“

**Casos de uso**

# Medir nivel del basurero:

Nombre CU: Medir nivel del basurero	
Descripción: La raspberry debe tomar lecturas del sensor, calcular el nivel de llenado y enviar al sistema la lectura realizada, si hay niveles superados el backend genera la alerta	
Actores: Raspberry pi, sensor ultrasonico	
Precondición: 1.- El sensor debe estar midiendo el contenedor previamente calibrado e instalado 2.- El contenedor existe en el sistema y está asociado al sensor 3.- El sistema debe estar cableado correctamente con: Sensor ultrasónico: con divisor de voltaje o driver lógico	
Flujo Principal: Sistema 1.- La raspberry pi lee el sensor  4.- Recibe y guarda los datos 5.- Actualiza el estado del contenedor 6.- si el nivel supera el límite se genera una alerta	Flujo Principal: Sensor ultrasónico  2.- Medir la distancia al nivel de basura 3.- Envía los datos al sistema
Flujo Alternativo: 1.1.- Si el sensor no responde, La Raspberry Pi reintentar la lectura  4.1.- Si no hay conexión, guarda los datos localmente y los envía cuando vuelva la red	
Postcondiciones: 1.- Estado del contenedor actualizado (Nivel, tendencia) 2.- Alertas generadas si se superan los límites	
Reglas de negocio: -	



# Visualizar cámara:

Nombre del CU: Visualizar cámara.	
Actor(es): Usuario	
Requisitos: RF-05	
Descripción: Permite al usuario visualizar el contenedor de basura mediante el sensor de cámara web.	
Precondiciones: Debe existir al menos una cámara disponible asociada a un contenedor. El usuario debe estar autenticado.	
Flujo Principal: Usuario 1. El usuario presiona el botón de “ver cámara”.	Flujo Principal: Sistema  2. El sistema validará si la cámara que quiere visualizar no está siendo ocupada por otro usuario. 3. Si la cámara está disponible, el sistema inicia la transmisión. 4. El sistema permite la visualización por un máximo de 60 segundos. 5. Transcurrido el tiempo, el sistema finaliza la transmisión automáticamente y retorna al usuario a la pantalla anterior.
Flujo Alternativo:	Flujo Alternativo: 2.1. Si la cámara ya está siendo visualizada por otro usuario, el sistema muestra el mensaje: “La cámara está siendo utilizada actualmente por otro usuario.”
Postcondiciones: Si el acceso fue exitoso, el usuario visualizó el contenedor durante un máximo de 60 segundos. El sistema regresa automáticamente a la pantalla anterior al finalizar el tiempo o antes si el usuario cierra la visualización.	
Reglas de Negocio: Solo se puede visualizar una cámara por usuario. El tiempo máximo de visualización continua es de 60 segundos, después de lo cual el sistema cierra la transmisión automáticamente.	
CU Relacionados : -	

# Enviar alerta de llenado:

Nombre del CU: Enviar alerta de llenado	
Actor(es): Sistema , Raspberry pi	
Requisitos: RF-03	
Descripción: Permite notificar al usuario si algún contenedor de basura se encuentra en su máxima capacidad.	
Precondiciones: El contenedor de basura debe estar a su máxima capacidad.	
Flujo Principal: Sistema          2. El Sistema recibirá la señal del sensor ultrasónico.   3.El Sistema generará una notificación.	Flujo Principal: Raspberry pi 1. El sensor debe mandar una señal a la aplicación móvil.
Flujo Alternativo:	Flujo Alternativo:
Postcondiciones: La aplicación móvil avisa al usuario mediante una notificación.	
Reglas de Negocio: -	
CU Relacionados : -	

# Visualizar estado del basurero:

Nombre del CU: Visualizar estado del basurero	
Actor(es): Usuario y administrador	
Requisitos Relacionados: RF-01, RF-02, RF-04	
Descripción: El usuario visualiza el estado actual del contenedor de basura(vacío, medio, lleno)	
Precondiciones: Deben existir información ya registrada	
Flujo Principal: 1. El usuario accede a la opción de "Consultar el estado de los contenedores".          5. Selecciona un contenedor o vista general.       6. El usuario revisa la información mostrada.	Flujo Principal:   2. Obtiene las últimas actualizaciones registradas para los contenedores.  3. Determina el estado de cada contenedor (Vacío, Medio, Lleno)}  4. Muestra en pantalla el estado de los contenedores.
Flujo Alternativo:	Flujo Alternativo: 2.1. Si no existen actualizaciones recientes, el sistema muestra "Sin datos disponibles para el contenedor seleccionado".
Postcondiciones: El usuario queda informado del estado actual de los contenedores seleccionados.	
Reglas de Negocio: -	
CU Relacionados : CU-01,CU-03	



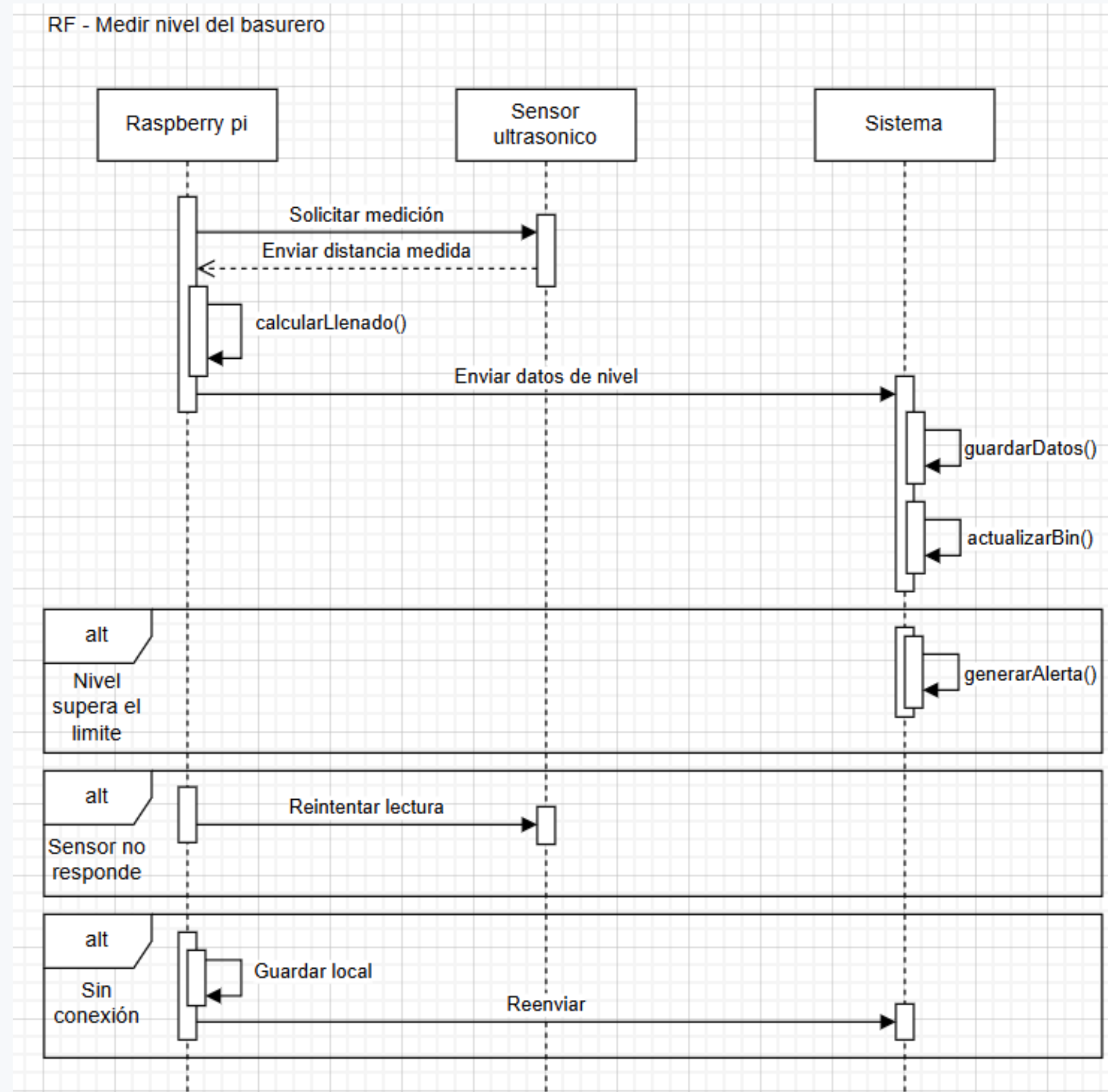
# Generar reporte de llenado:

Nombre CU: Generar reporte de llenado	
Descripción: El administrador genera reportes sobre los niveles de llenado y alertas de los contenedores para análisis y toma de decisiones.	
Actor: Administrador del sistema	
Precondición: Debe existir historial de lecturas y alertas almacenado.	
Flujo Principal: Administrador del sistema	Flujo Principal: Sistema
1. Accede al botón "reportes"	2. Muestra el formulario para seleccionar contenedores.
3. Selecciona el rango de fechas y los contenedores a analizar.	4.muestra el historial de lecturas y alertas según los filtros.
5. solicita reporte de actividad	6. Genera el reporte y lo muestra en la interfaz.
7. Revisa el reporte generado y si corresponde elige guardar el reporte.	8. Guarda el reporte
Flujo Alternativo:	Flujo Alternativo:
	4.1. Si no hay reportes en el rango seleccionado, el sistema muestra "No existen reportes para la fecha ingresada".
Postcondiciones: Se dispone de un reporte con la información de llenado de los contenedores para analizar correspondiente.	
Reglas de Negocio: -	
CU Relacionados: CU-01, CU-03	

## Iniciar sesión:

Nombre CU: Iniciar sesión	
Descripción: Permite que el administrador acceda al sistema ingresando sus datos.	
Actor: Administrador	
Requisitos relacionados: RF- 08	
Precondición:	
Flujo Principal: Administrador del sistema  1. Ingresa sus datos: <ul style="list-style-type: none"><li>• Usuario</li><li>• Contraseña</li></ul> 2. Confirma el inicio de sesión	Flujo Principal: Sistema    3. Recibe los datos ingresados 4. Valida los datos(usuario,contraseña) 5. Permite el acceso al administrador 6. Muestra el menú de sus funciones correspondientes.
Flujo Alternativo:  1.1. Datos incorrectos, el sistema rechaza el acceso.	Flujo Alternativo:  1.2. Muestra mensaje : " Usuario o contraseña incorrectos".
Postcondiciones: Accede a las funciones del sistema exitosamente.	
Reglas de negocio: Solo los administradores tienen acceso a las funciones de gestión.	
CU Relacionados: --	

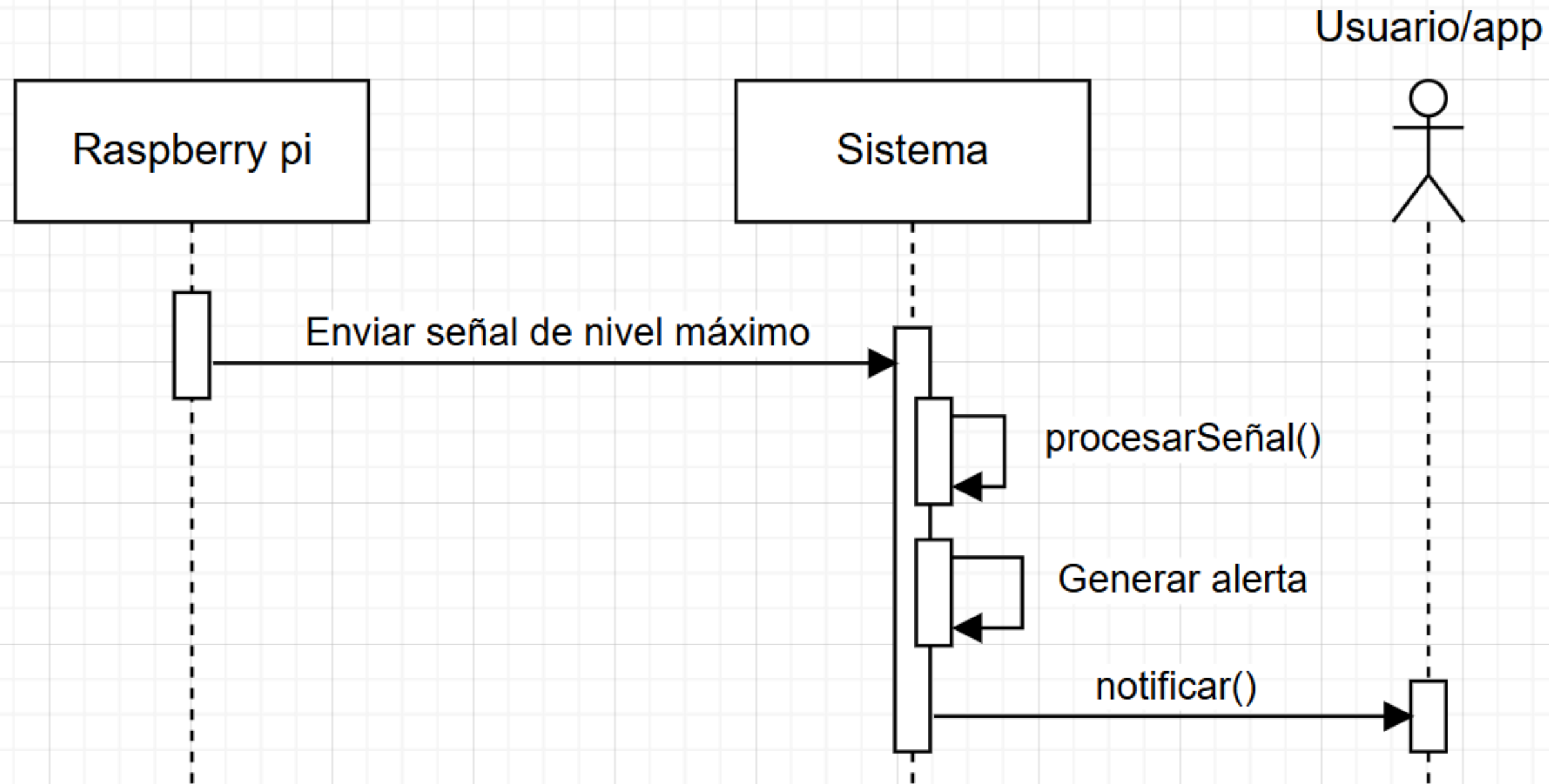
# Diagrama de secuencias



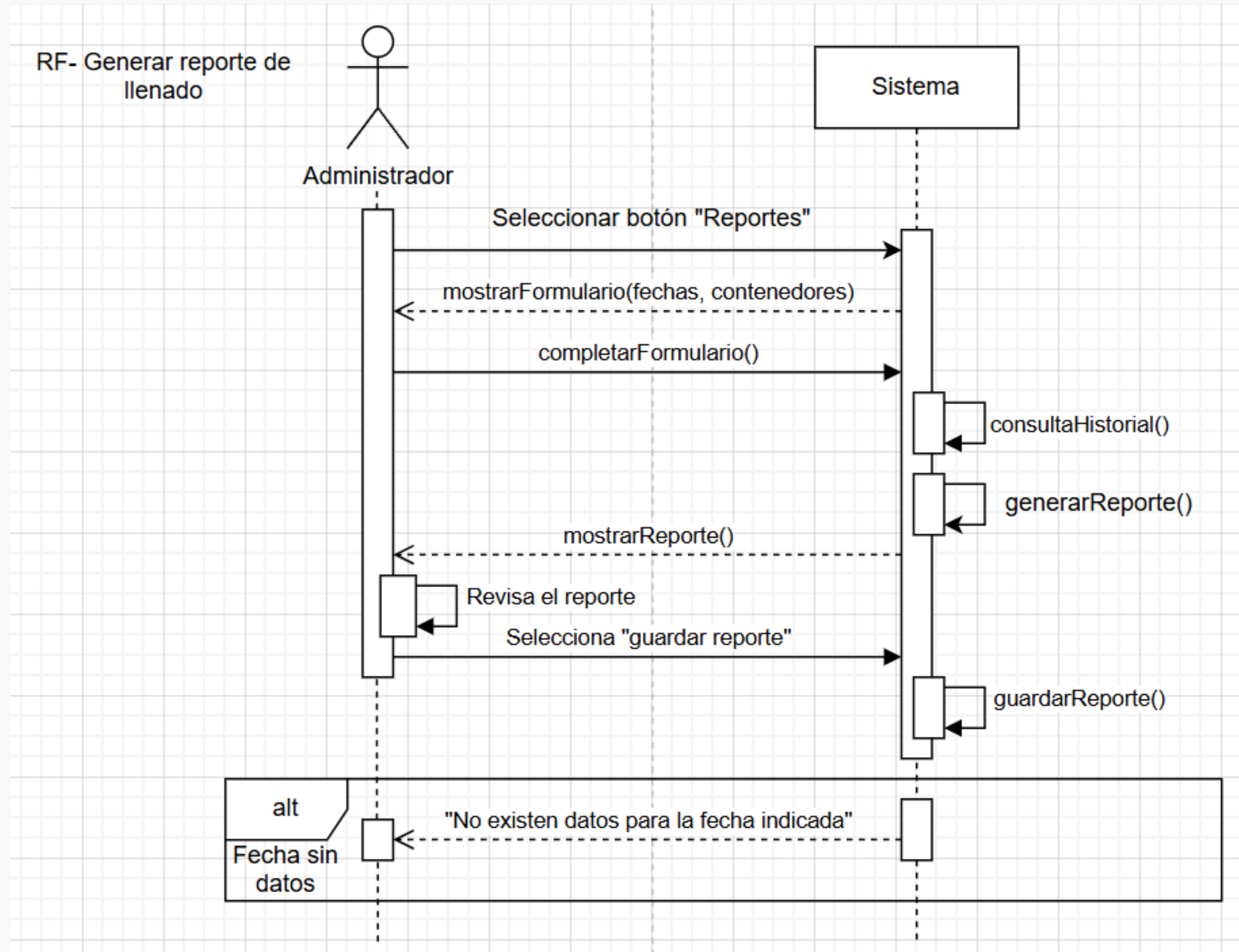


# Diagrama de secuencias

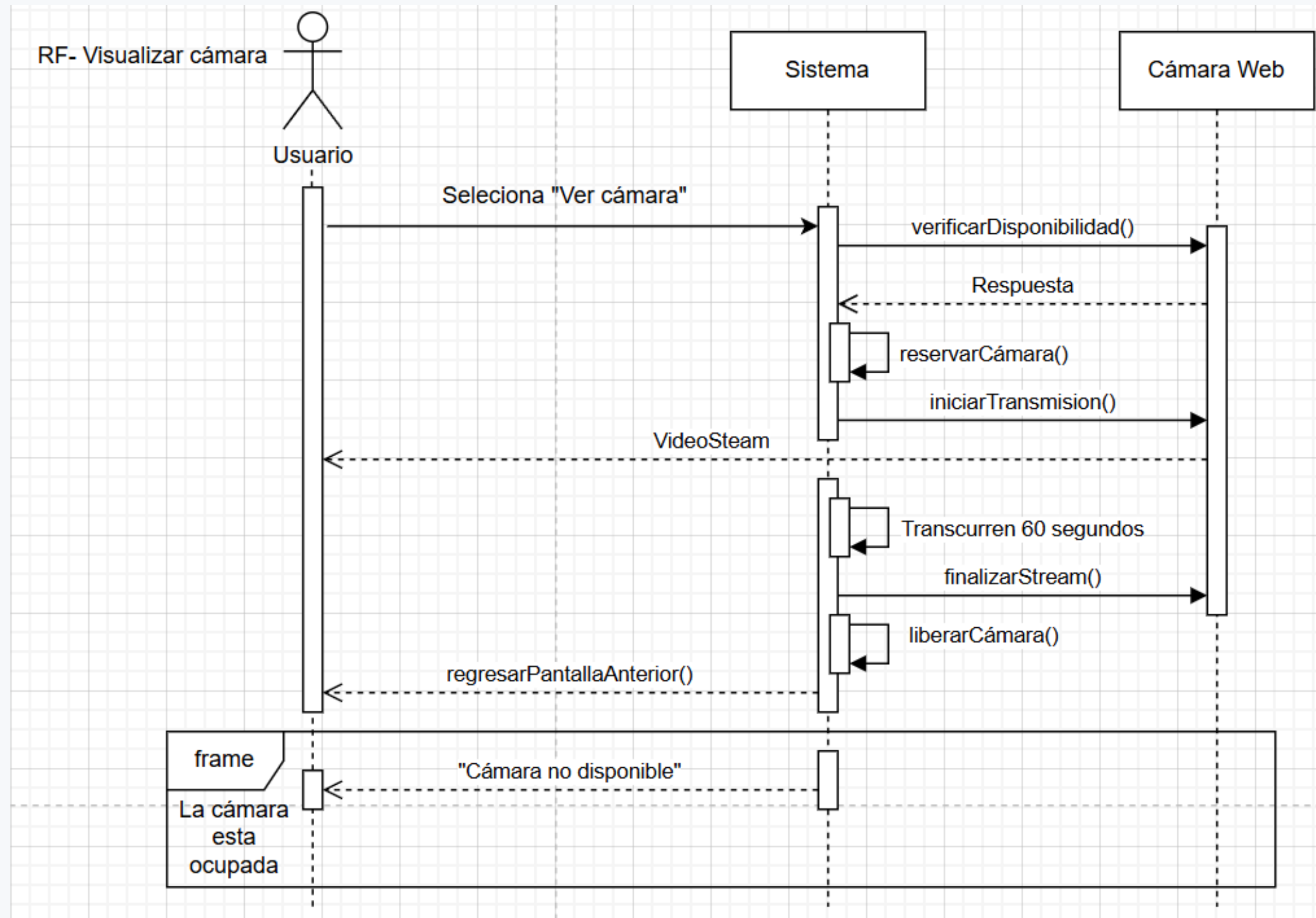
RF - Enviar alerta de llenado



# Diagrama de secuencias

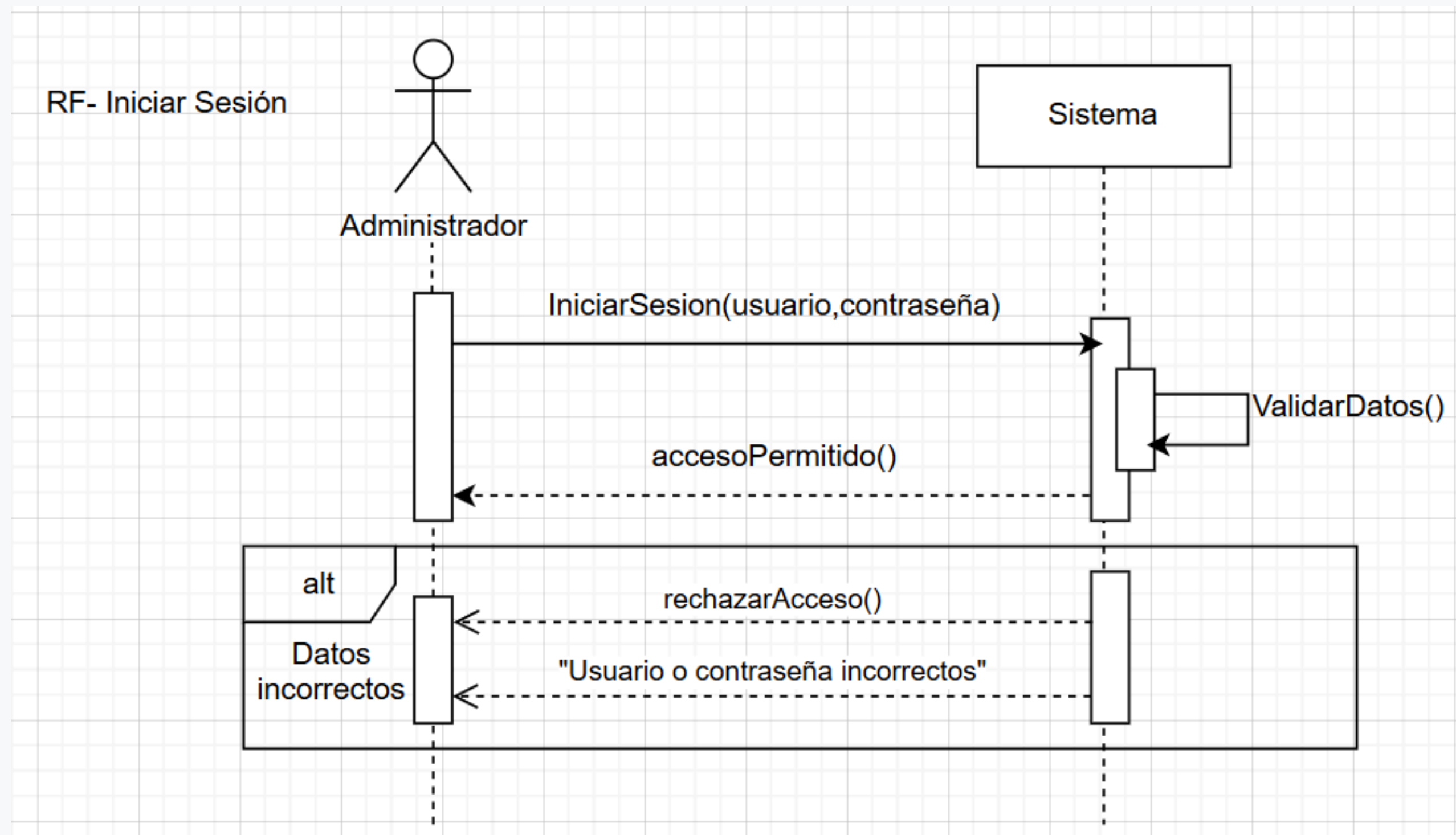


# Diagrama de secuencias





# Diagrama de secuencias



# Conclusión

En este segundo avance se consolidó la estructura del proyecto Sistema de detección y Alerta de basura : “Bin Raider” mediante la definición de los requerimientos, la elaboración de los casos de uso y la construcción del diseño completo del sistema (clases, secuencias y arquitectura). También se desarrolló la secuencia de pantallas de la aplicación móvil y se incorporaron las correcciones del Informe I, fortaleciendo la claridad del proyecto. Con todo esto, quedaron definidos los flujos y responsabilidades de cada módulo, permitiendo avanzar hacia la etapa final de implementación y pruebas del prototipo.

**¡Gracias!**



# Referencias bibliográficas

[1] Raspberry Pi Foundation, Foros de Raspberry Pi, 2025. [En línea]. Disponible en: <https://forums.raspberrypi.com>

[2] MCIElectronics, Listado de precios 2025. [En línea]. Disponible en: <https://mcielectronics.cl>

[3] MIDDA, Inspiración del proyecto, 2025. [En línea]. Disponible en: <https://www.midda.cl>

[4] Instructables, Automatic Trash Bin – Información de componentes, 2025. [En línea]. Disponible en: <https://www.instructables.com/Automatic-Trash-Bin/>

[5] Hubot, Sensores compatibles con Raspberry Pi, 2025. [En línea]. Disponible en: [https://hubot.cl/?s=sensores+raspberrypi&post\\_type=product](https://hubot.cl/?s=sensores+raspberrypi&post_type=product)

[6] Raspberry Pi Foundation, Documentación oficial Raspberry Pi 4 Model B, 2025. [En línea]. Disponible en: <https://www.raspberrypi.com/documentation/>

[7] HC-SR04, Hoja técnica del sensor ultrasónico HC-SR04, 2024. [En línea]. Disponible en: <https://cdn.sparkfun.com/datasheets/Sensors/Proximity/HCSR04.pdf>

[8] Python Software Foundation, Documentación oficial Python 3.10, 2025. [En línea]. Disponible en: <https://docs.python.org/3/>

[9] Instructables, Smart Garbage Monitoring System – Proyecto de referencia, 2024. [En línea]. Disponible en: <https://www.instructables.com/>