



# PROYECTO CIRCUITRON





# CONTENIDO

01

Introducción

02

Objetivos

03

Carta Gantt

04

Caso de uso  
general

05

CUS

06

Escenario

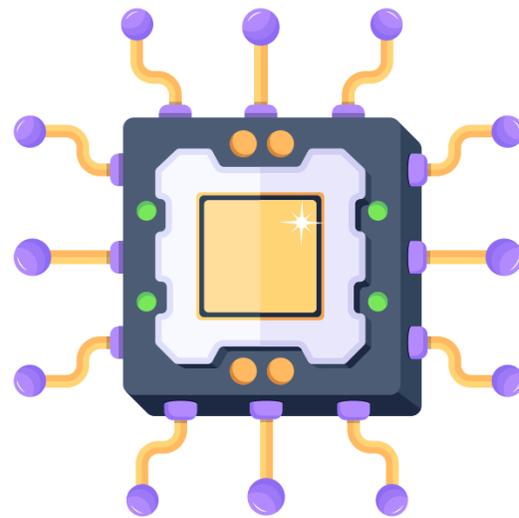
07

Conclusión





# INTRODUCCIÓN



El proyecto "Circuitron" se centra en el desarrollo de un sistema robusto y eficiente en el ámbito de la robótica y la tecnología. Aunque inicialmente se planteó participar en una competencia, el enfoque del proyecto ahora está dirigido al desarrollo de un robot. Para lograr esto, se han considerado aspectos fundamentales como el presupuesto, la organización y la planificación de actividades (Carta Gantt), la documentación continua del progreso (Informe de planificación) y un plan de gestión de riesgos. Todo esto tiene como objetivo garantizar que el proyecto se desarrolle de manera óptima, maximizando la eficiencia en el uso de recursos y cumpliendo con los plazos establecidos.

# OBJETIVOS



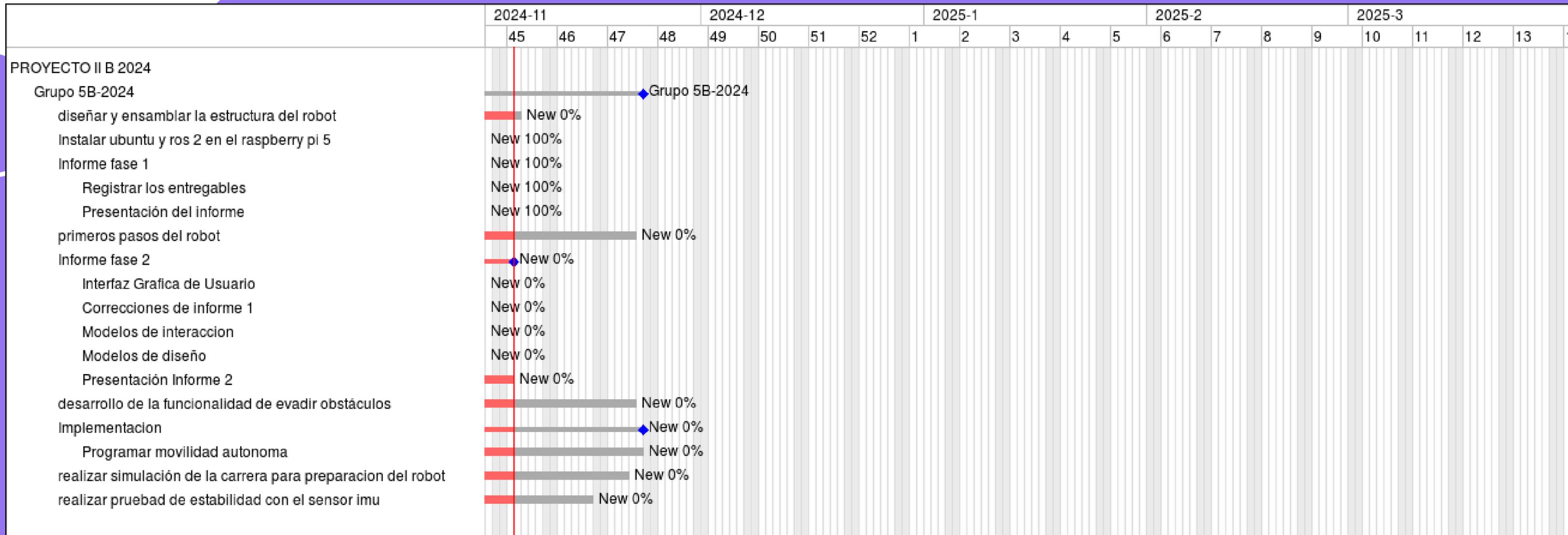
## OBJETIVO GENERAL

Desarrollar un robot bípedo humanoide capaz de desplazarse de manera autónoma y eficiente a través de circuitos con obstáculos y desniveles, integrando tecnologías avanzadas de control y detección que le permitan adaptarse dinámicamente a diferentes entornos y superar los desafíos de equilibrio y movilidad.

## OBJETIVOS ESPECIFICOS

- Desarrollar un sistema de locomoción eficiente que permita al robot bípedo desplazarse de manera autónoma y mantener el equilibrio en terrenos irregulares y cambiantes.
- Diseñar una interfaz de usuario que permita monitorear en tiempo real el estado del robot y realizar ajustes durante la competencia.
- Implementar un sistema de detección y reconocimiento de obstáculos utilizando sensores como sensor ultrasonido.
- Establecer un sistema de control de movimientos articulados que permita al robot superar diferentes tipos de obstáculos.
- Realizar pruebas y ajustes iterativos del hardware y software para optimizar el rendimiento del robot.

# CARTA GANTT



# REQUERIMIENTOS FUNCIONALES



Los requerimientos funcionales descritos incluyen acciones específicas que el sistema debe cumplir para el proyecto "Circuitron", un robot bípedo humanoide:

- Mover Robot: El robot debe responder al movimiento de acuerdo con los datos recopilados e instrucciones de la Raspberry Pi.
- Estabilizar Robot: Asegura la estabilidad del robot utilizando información de un sensor giroscópico.
- Detectar Obstáculos: Permite al robot identificar obstáculos mediante un sensor ultrasónico y almacenarlos en el sistema.
- Conectar al Robot: Activa el robot y prepara el sistema para otras funciones.

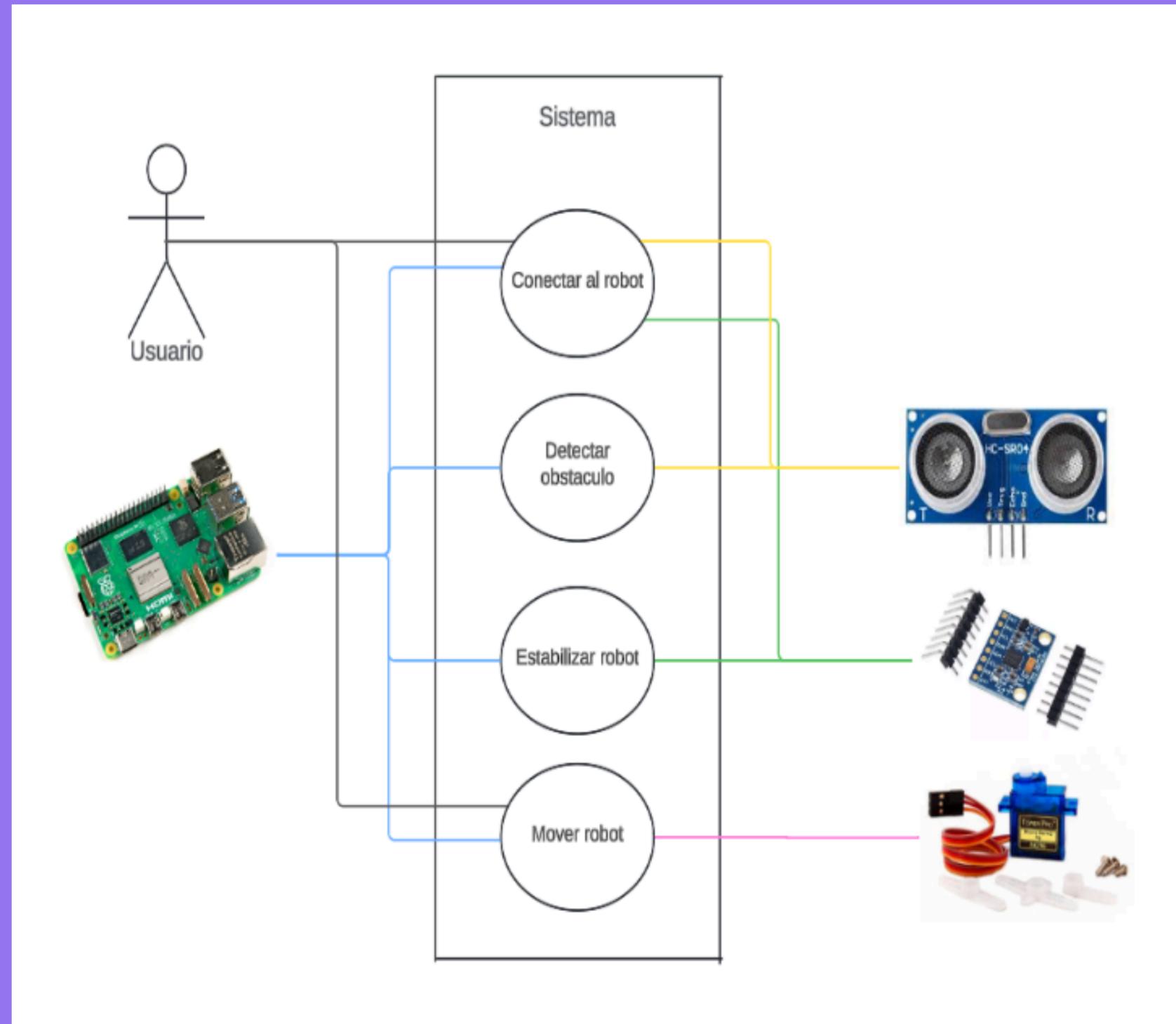
# REQUERIMIENTOS NO FUNCIONALES



Los requerimientos no funcionales se enfocan en aspectos de desempeño y limitaciones del sistema:

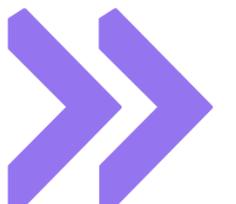
- **Estabilidad del Sistema:** Debe mantener la estabilidad del robot en tiempo real mediante la retroalimentación de sensores y algoritmos de control.
- **Optimización Continua:** Se requiere que el sistema se adapte a diferentes entornos sin modificar el hardware mediante simulaciones y ajustes continuos.
- **Restricciones de Energía:** La autonomía del robot está limitada por la capacidad de las baterías, afectando la duración de las pruebas.
- **Capacidad de Adaptación:** El sistema debe funcionar eficientemente en terrenos controlados y predefinidos.
- **Compatibilidad con sensores:** El sistema debe integrar sensores avanzados y modernos para procesamiento en tiempo real.

# CASO DE USO GENERAL

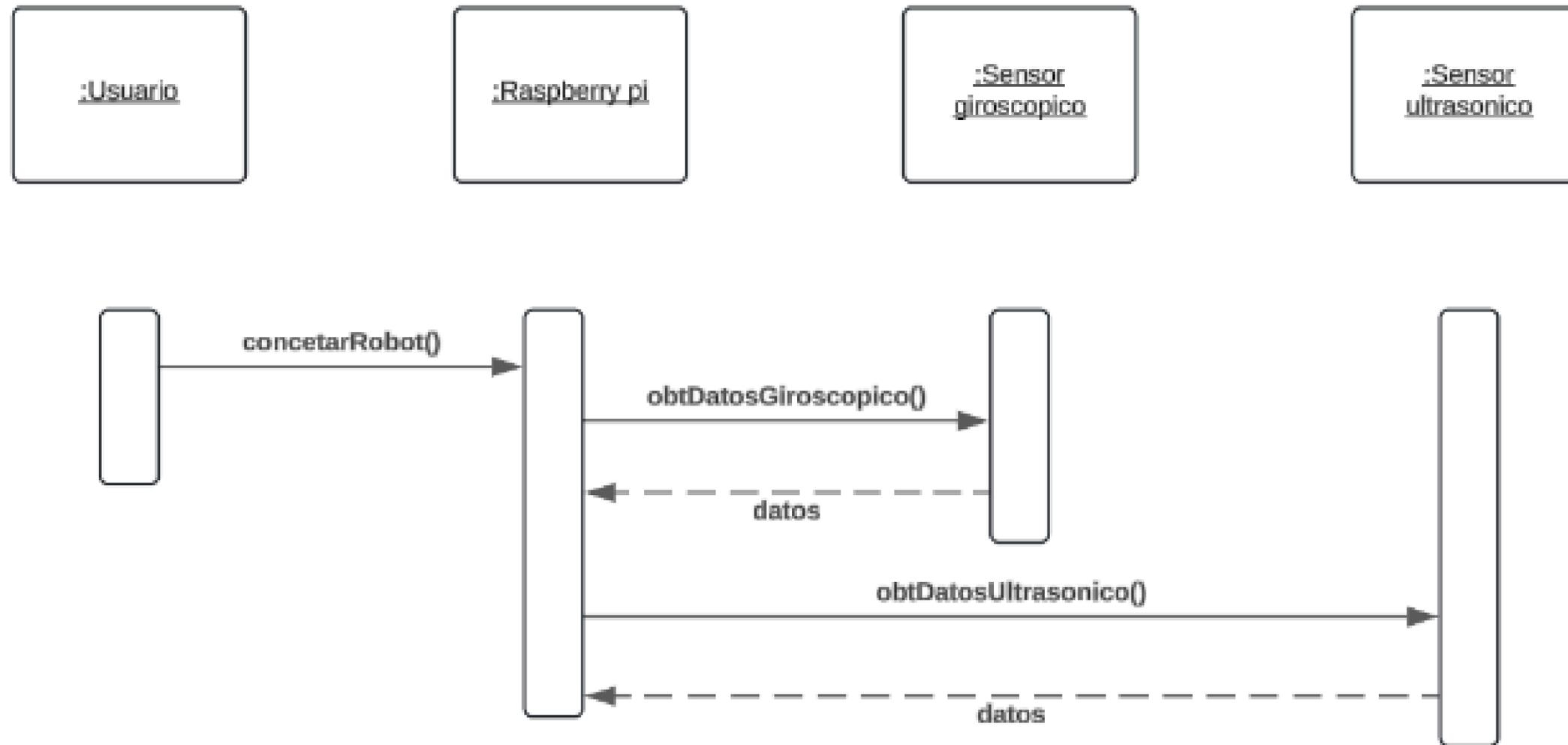


# CUS CONECTAR AL ROBOT

<b>Nombre CUS: Conectar al robot</b>	
Descripción: El robot prepara su sistema para realizar sus funciones.	
Actor: Usuario, Raspberry, sensor giroscópico, sensor ultrasónico.	
Precondición: N/A	
Flujo Principal: Usuario  2.- Selecciona "Conectar al robot".	Flujo Principal: Sistema 1.- Muestra: "Conectar al robot"  3.- Muestra: <ul style="list-style-type: none"><li>• Datos del sensor ultrasónico.</li><li>• Datos del sensor giroscópico.</li></ul>
Flujo Alternativo:	
Postcondiciones: El sistema da inicio a las funciones de los sensores y lo muestra.	



# DIAGRAMA DE SECUENCIA

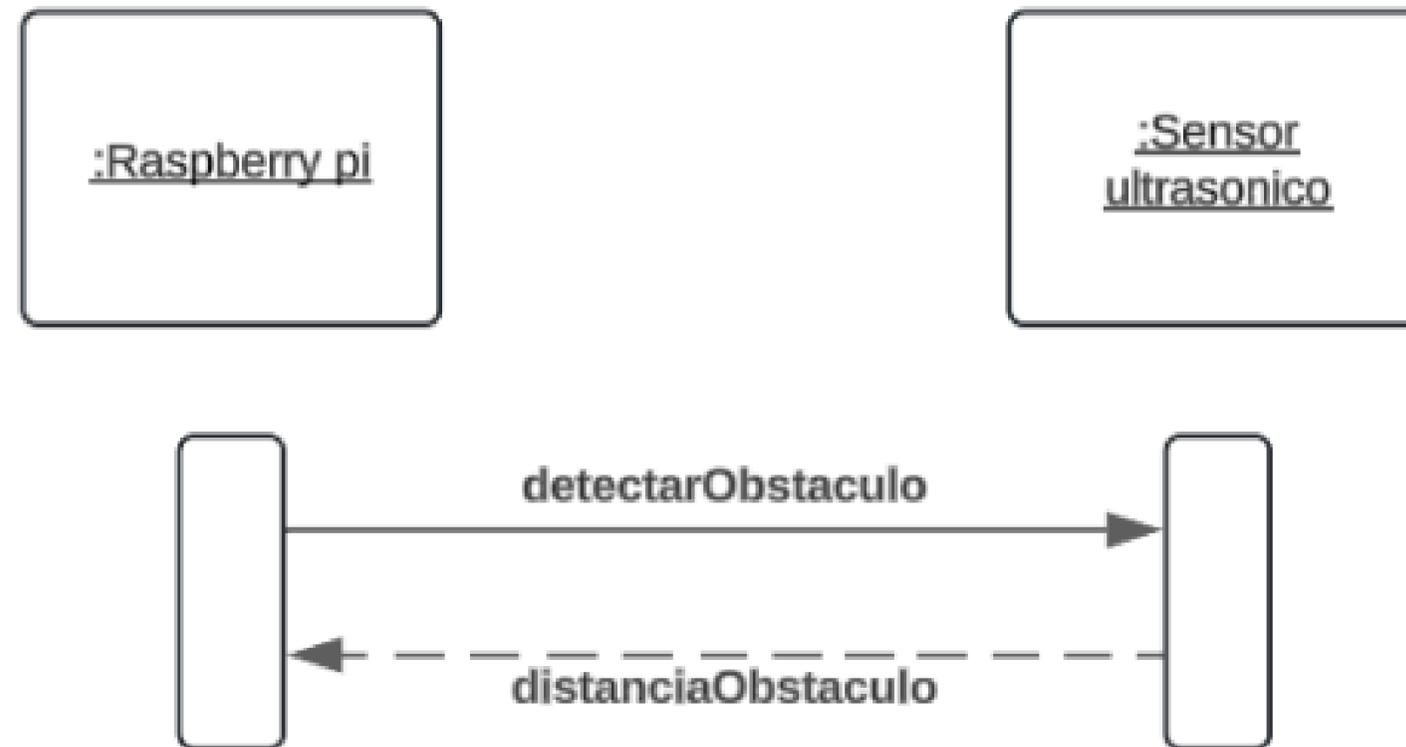


# CUS DETECTAR OBSTACULO

<b>Nombre CUS: Detectar obstáculo</b>	
Descripción: Este CUS permite detectar los obstáculos al robot.	
Actor: Raspberry pi, sensor ultrasónico.	
Precondición: El robot debe haber sido encendido y sus sensores deben poder obtener los datos correspondientes.	
Flujo Principal: Robot	Flujo Principal: Sistema 1.- Include <<Conectar al robot>>  2.- El sensor ultrasónico obtiene los datos de distancia.  3.- La raspberry pi recepciona y almacena el dato.  4.- Muestra la distancia a la que se encuentra el obstáculo.
Flujo Alternativo:	
Postcondiciones: El sistema tiene los datos para evitar el obstáculo.	



# DIAGRAMA DE SECUENCIA

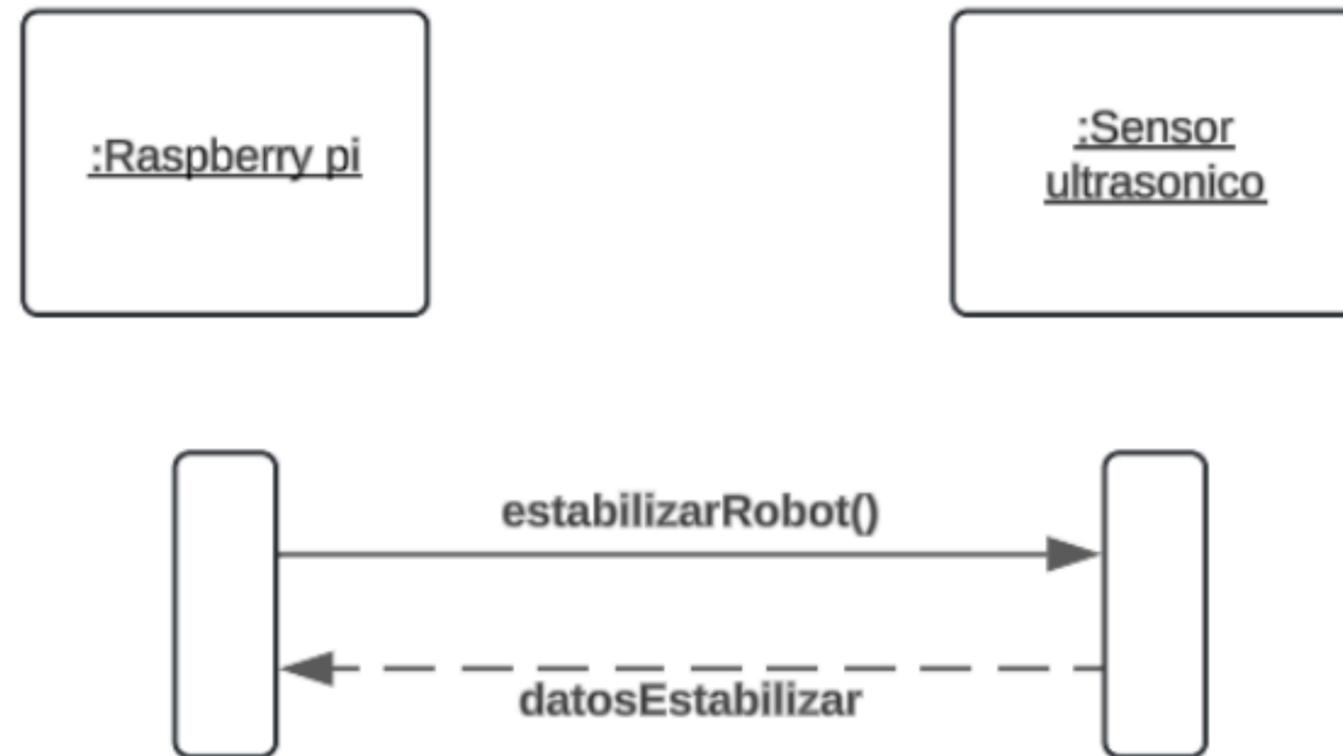


# CUS ESTABILIZAR ROBOT

<b>Nombre CUS: Estabilizar Robot</b>	
Descripción: Este CUS permite que el robot permanezca estable a través de la información recibida del sensor giroscópico.	
Actor: Raspberry pi, sensor giroscópico.	
Precondición: El robot debe haber sido encendido y sus sensores deben poder obtener los datos correspondientes.	
Flujo Principal: Robot	Flujo Principal: Sistema 1.- Incluye <<Conectar al Robot>>  2.- El sensor giroscópico obtiene los datos para estabilizar el robot.  3.- La raspberry pi recibe los datos y los almacena en el sistema.  4.- Muestra los datos del sensor giroscópico
Flujo Alternativo:	
Postcondiciones: El sistema tiene los datos para mantener estable el robot.	

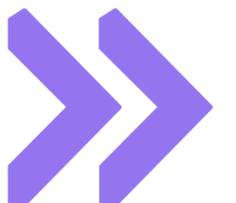


# DIAGRAMA DE SECUENCIA

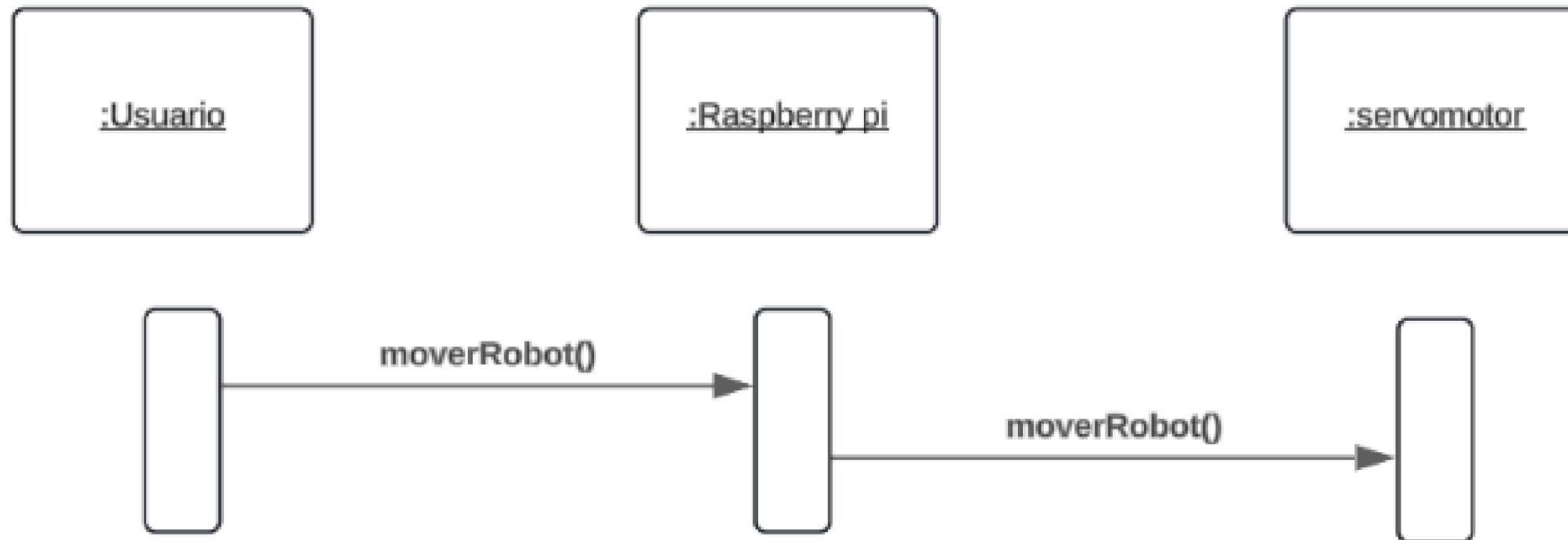


# CUS MOVER ROBOT

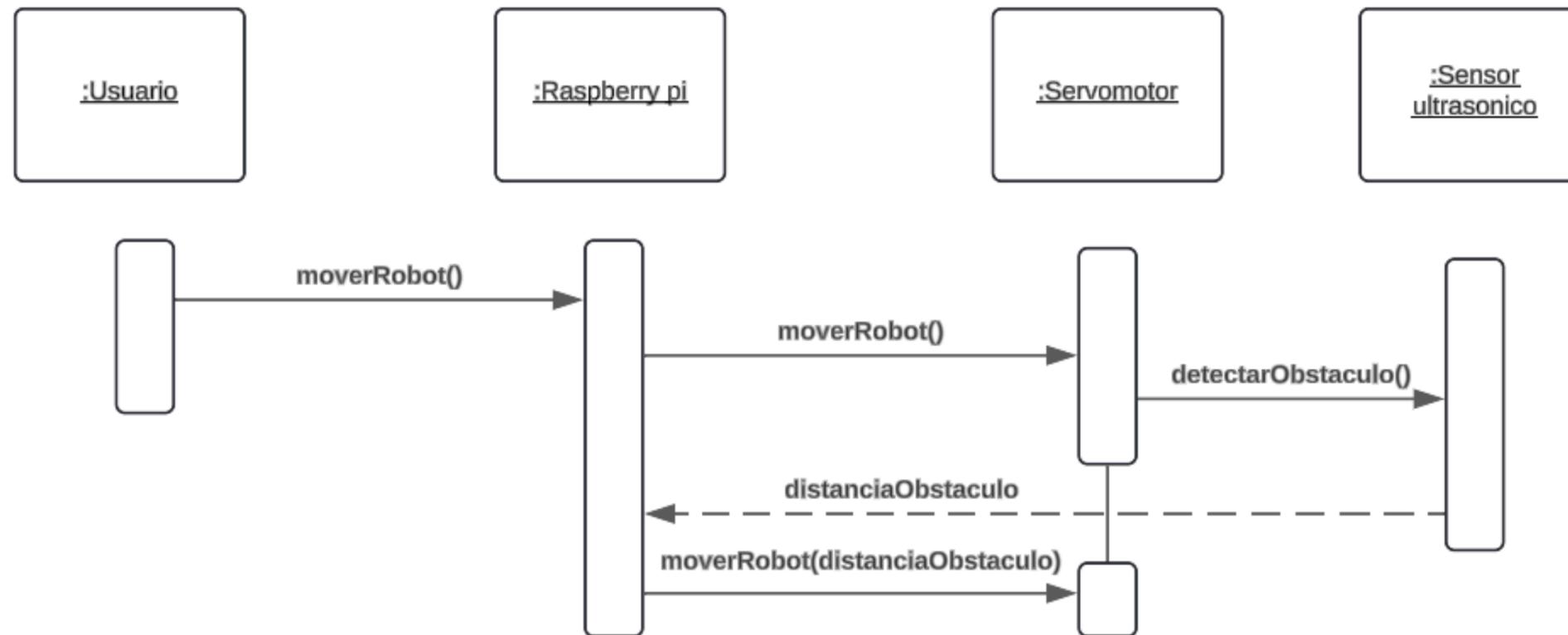
<b>Nombre CUS: Mover Robot</b>	
Descripción: El robot responde con movimiento de acuerdo a la recopilación de datos y a las instrucciones de la raspberry	
Actor: Servo Motor,Raspberry	
Precondición: Los datos para manejar la estabilidad del robot deben estar en el sistema, la información de los obstáculos debe estar en el sistema.	
Flujo Principal: Usuario  3.- Pulsa el botón mover robot.	Flujo Principal: Sistema 1.- Include<<Estabilizar Robot>>  2.- Muestra "Mover robot".  4.- El sistema debe dar la orden de movilizarse según los datos obtenidos.
Flujo Alternativo: robot encuentra un obstáculo.	4.-Include<<Detectar Obstáculo>>  5.-El sistema debe dar la orden de esquivar obstáculos.
Postcondiciones: Robot en movimiento con sus servomotores funcionando	



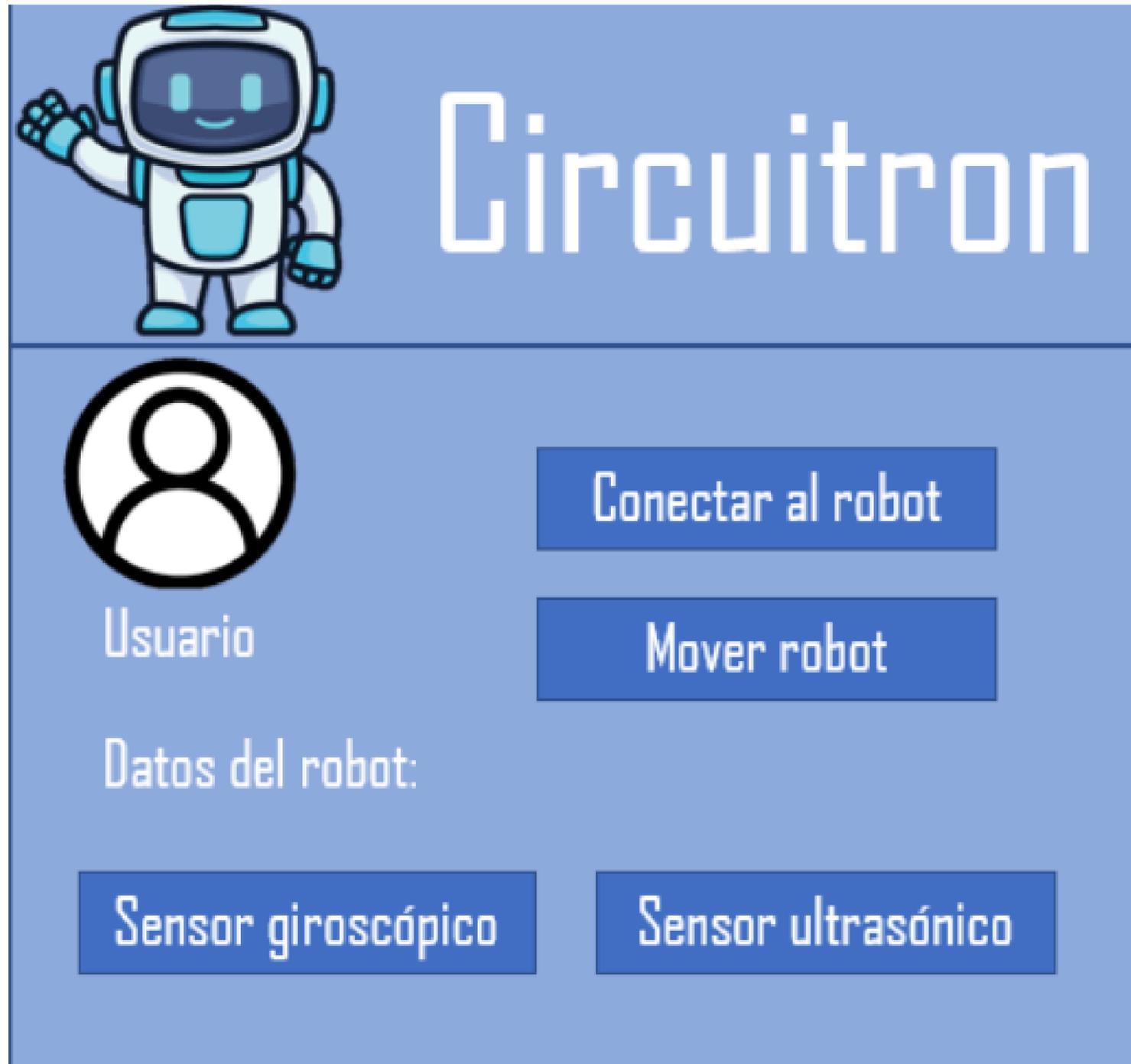
# DIAGRAMA DE SECUENCIA



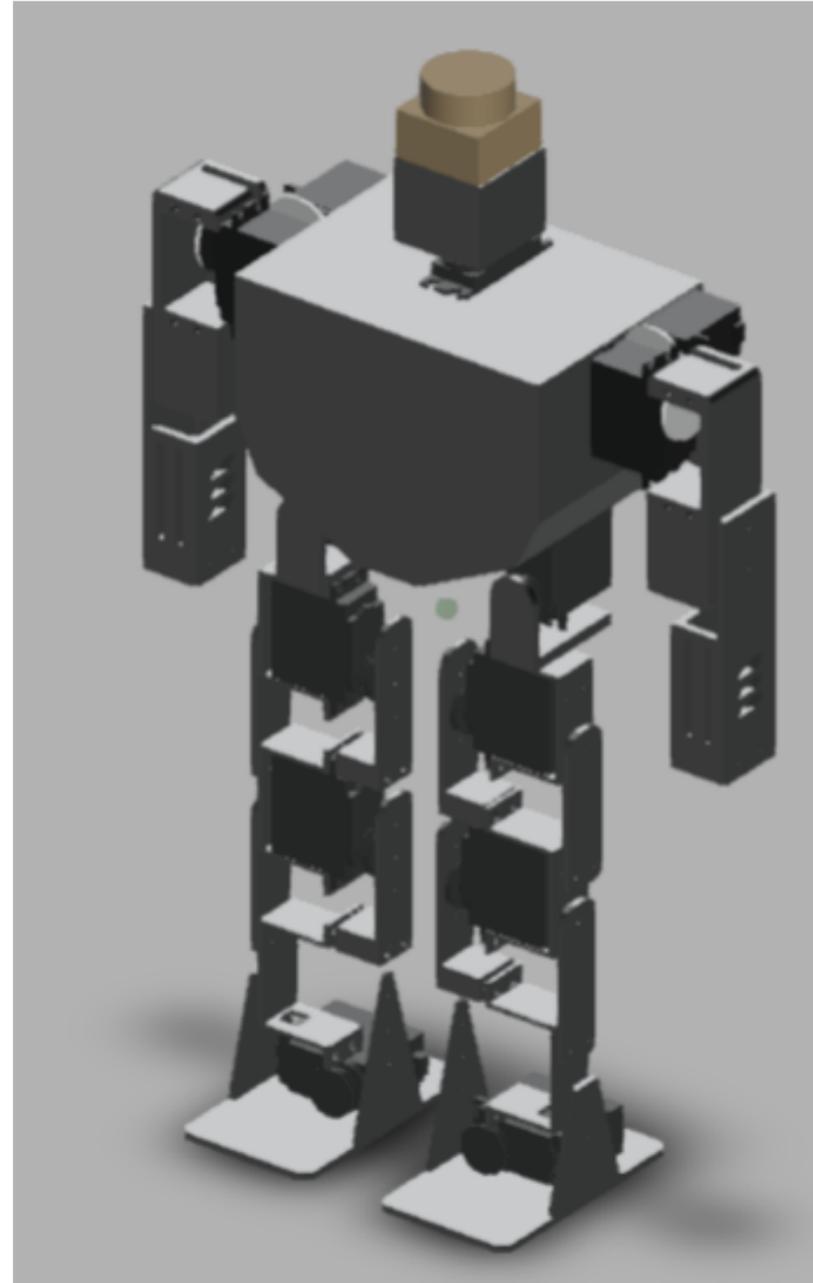
# DIAGRAMA DE SECUENCIA ESCENARIO(OBSTACULO DETECTADO)



# INTERFAZ GRAFICA (IGU)



# DISEÑO DEL ROBOT



# HERRAMIENTAS Y TÉCNICAS

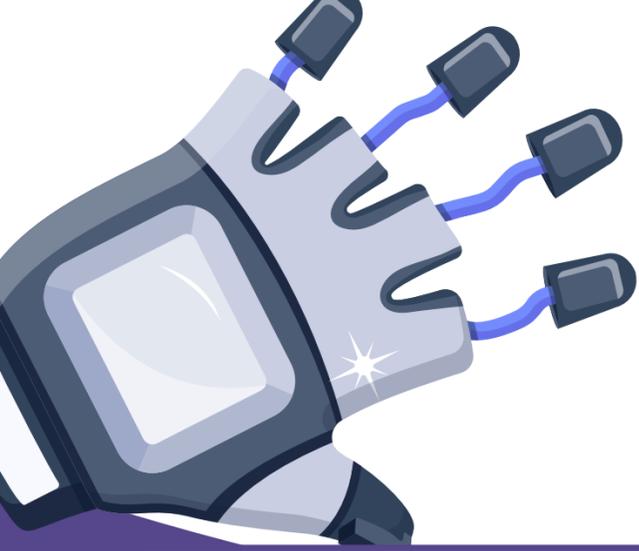


# HERRAMIENTAS Y TÉCNICAS

## TÉCNICAS:

- **DIVIDIR PARA CONQUISTAR: DESCOMPOSICIÓN DE TAREAS COMPLEJAS EN SUBTAREAS MANEJABLES PARA OPTIMIZAR EL DESARROLLO.**
- **PROTOTIPADO ITERATIVO: DESARROLLO DE VERSIONES PRELIMINARES DEL ROBOT PARA AJUSTES Y MEJORAS.**
- **ANÁLISIS DE FALLOS: EVALUACIÓN DE POSIBLES PUNTOS DE FALLO PARA INCREMENTAR LA CONFIABILIDAD DEL ROBOT.**





# CONCLUSIÓN



**Desafíos de Ingeniería y Adaptabilidad:** El proyecto demostró que desarrollar robots bípedos humanoides requiere un control preciso y adaptabilidad en tiempo real. Mediante el uso de sensores avanzados y algoritmos predictivos, el robot fue capaz de adaptarse a entornos cambiantes, lo que valida la efectividad de esta combinación para futuros desarrollos robóticos.

**Eficiencia en la Gestión de Recursos:** A pesar de contar con un presupuesto y un tiempo de desarrollo limitados, el equipo logró implementar una solución funcional. Esto subraya la importancia de una planificación eficaz y una asignación de recursos estratégicamente pensada, especialmente en proyectos con restricciones significativas.

**Limitaciones y Oportunidades de Mejora:** Los desafíos de estabilidad, energía y limitaciones de tiempo resaltan áreas donde futuras iteraciones pueden enfocarse en optimizar el hardware y software del robot, especialmente en relación con la autonomía y resistencia en entornos más exigentes.



**MUCHAS GRACIAS**  
**POR VER ESTA PRESENTACIÓN**

