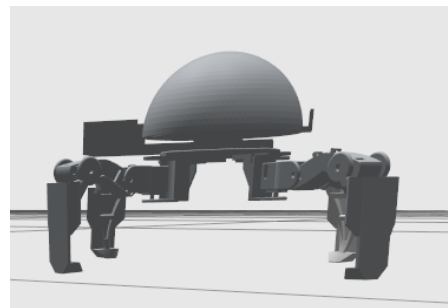
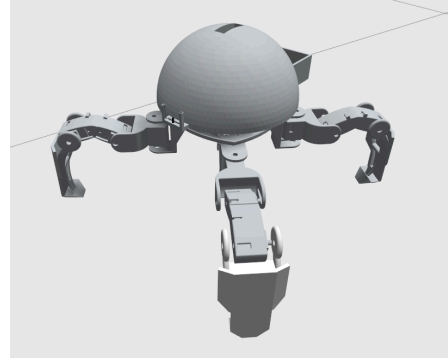
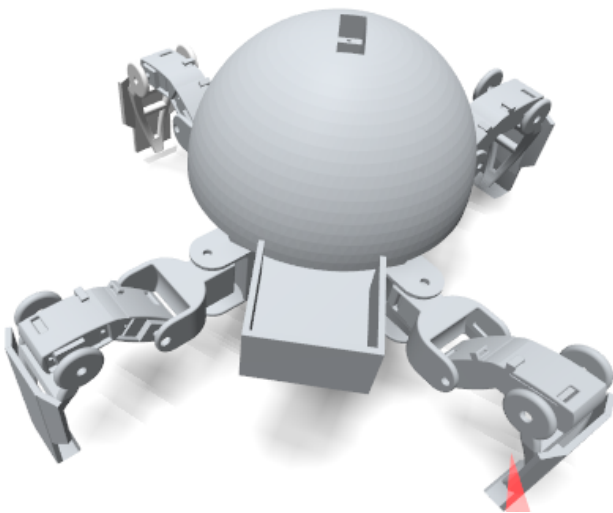


# Robot con Patas

**Grado:**

Ingeniería de Robótica Software.

**Asignatura:**

Mecatrónica.

**Integrantes del equipo:**

Alba López del Águila, Álvaro Valencia Maiquez, Daniel Guillén Badiola.

# 1. Índice:

1. Introducción
2. Diseño del sistema
  - 2.1 Arquitectura general
  - 2.2 Justificación del diseño seleccionado
3. Diseño Mecánico
  - 3.1 Configuración estructural
  - 3.2 Modelado 3D y características del chasis
  - 3.3 Fabricación del prototipo
4. Diseño electrónico
  - 4.1 Actuadores
  - 4.2 Sensores
  - 4.3 Unidad de control
  - 4.4 Fuente de alimentación
  - 4.5 Diagrama eléctrico
5. Sistema de control y software embebido
  - 5.1 Arquitectura del software
  - 5.2 Cinemática y movimiento
  - 5.3 Patrón de marcha
  - 5.4 Algoritmos de navegación
  - 5.5 Interfaz de usuario
6. Pruebas experimentales y validación
  - 6.1 Metodología de pruebas
  - 6.2 Resultados de la Prueba 1
  - 6.3 Resultados de la Prueba 2
  - 6.4 Resumen requisitos y cumplimiento
7. Discusión técnica
8. Propuestas de mejora
9. Conclusiones
10. Referencias y documentación consultada

## 1. Introducción

Se ha diseñado y construido un robot cuadrúpedo capaz de superar las dos pruebas establecidas en el enunciado: desplazamiento de 1 m paralelo a una pared transportando 250 g y navegación autónoma en un área de 2 m<sup>2</sup> con evasión de obstáculos. El sistema integra diseño mecánico propio, electrónica embarcada, control basado en ROS 2 y servomotores MG90S. Todas las restricciones de dimensiones, alimentación mediante baterías y autonomía de funcionamiento han sido verificadas experimentalmente.

## 2. Diseño del sistema

### 2.1 Arquitectura general

El robot desarrollado presenta una arquitectura cuadrúpeda con un cuerpo central rígido y cuatro patas articuladas dispuestas de forma simétrica. Cada pata dispone de tres grados de libertad, dando lugar a un total de 12 articulaciones controladas de manera independiente.

El sistema se completa con un módulo superior formado por una cabeza giratoria, que integra los sensores principales del robot y puede orientarse de manera independiente al cuerpo. A nivel funcional, el robot se estructura en tres bloques principales: sistema mecánico, sistema electrónico y sistema de control y software, que interactúan de forma coordinada para permitir el movimiento y la navegación autónoma.

### 2.2 Justificación del diseño seleccionado

La elección de una configuración cuadrúpeda se debe a las ventajas que ofrece en términos de estabilidad frente a otras alternativas con menor número de patas, manteniendo al mismo tiempo una complejidad de control moderada. Esta configuración permite asegurar estabilidad estática durante la marcha a baja velocidad, adecuada para las pruebas exigidas.

El uso de tres grados de libertad por pata proporciona la flexibilidad necesaria para generar patrones de marcha estables sin incrementar excesivamente la complejidad mecánica ni computacional. Asimismo, la inclusión de una cabeza giratoria permite orientar los sensores sin modificar la postura del robot, mejorando la percepción del entorno y la toma de decisiones durante la navegación.

## 3. Diseño mecánico

### 3.1 Configuración estructural

La estructura mecánica del robot se divide en los siguientes subconjuntos:

**Cuerpo central (chasis):** estructura portante donde se integran la electrónica de control, la batería y el sistema de distribución de potencia

**Patas articuladas:** cada una formada por tres segmentos accionados mediante servomotores.

**Módulo superior (cabeza):** cúpula giratoria que aloja la cámara frontal y el sensor ultrasónico.

La disposición geométrica de las patas alrededor del chasis proporciona una base estable para el desplazamiento del robot.

### 3.2 Modelado 3D y características del chasis

El diseño mecánico del robot se ha realizado mediante herramientas CAD, generando modelos tridimensionales detallados del chasis y las patas. Estos modelos han sido posteriormente exportados a un formato compatible con ROS para su integración en el modelo URDF, permitiendo su simulación en entornos como Gazebo y RViz2.

Las principales características del chasis son:

- **Dimensiones totales:** ajustadas para cumplir la restricción máxima de 30 × 30 × 30 cm.
- **Geometría:** cuerpo central compacto con anclajes laterales para las patas y soporte superior para la cúpula.
- **Distribución de masas:** diseño orientado a mantener el centro de gravedad bajo y centrado.
- **Materiales:** piezas fabricadas mediante impresión 3D en PLA, seleccionados por su relación entre rigidez, peso y facilidad de fabricación.

### 3.3 Fabricación del prototipo

La fabricación del prototipo se ha llevado a cabo principalmente mediante impresión 3D, permitiendo una rápida iteración del diseño y la adaptación de las piezas a los componentes electrónicos y actuadores seleccionados.

El proceso incluyó la fabricación de las piezas estructurales, el ensamblaje de las patas con los servomotores, el montaje del módulo superior y ajustes mecánicos finales para mejorar la robustez.

## 4. Diseño electrónico

### 4.1 Actuadores

El sistema de locomoción del robot está basado en **servomotores de tipo MG90S**, utilizados para accionar las articulaciones de las patas y el giro de la cabeza.

Cada pata integra tres servos correspondientes a la articulación de cadera (rotación horizontal), la articulación intermedia (elevación de la pata) y la articulación distal (extensión del segmento final).

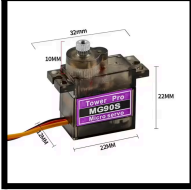
### Justificación de la selección de servos

La elección de estos servomotores se ha realizado teniendo en cuenta los siguientes criterios:

- **Par nominal suficiente** para soportar el peso del robot y la carga adicional de 250 g exigida en la prueba.
- **Compatibilidad con control PWM**, facilitando su integración con controladores externos.
- **Disponibilidad comercial y coste contenido**, adecuados para un prototipo académico.

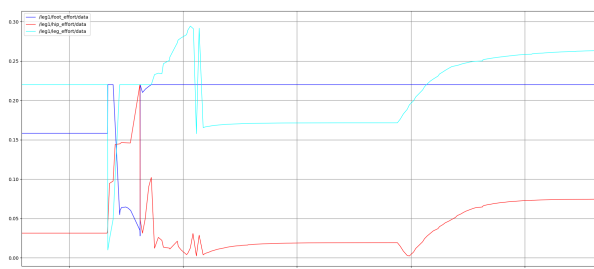
La selección de los servomotores MG90S se validó mediante simulación, verificando que el par requerido durante los movimientos se mantenía dentro de los márgenes admisibles del actuador. Esto garantiza un funcionamiento estable durante las pruebas con carga exigidas.

### Características técnicas:

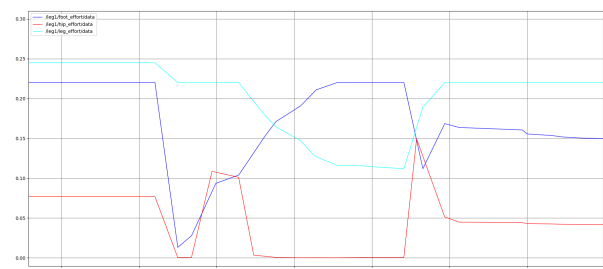
Modelo:MG90S	Tamaño:22.8x22.4x12mm	
Par:2kg/cm	Velocidad de respuesta: 0.11s para 60° (a 4.8V)	
Voltaje: 4.8V	Banda muerta: 5 ms	
Tipo de conector: JR, estándar FUTABA	Ángulo de rotación: 0-180° o 360° (continuo)	
Modo de operación: Analógico	Rango de temperatura: 0 °C a 55 °C	

### Gráficas de esfuerzo/par obtenidas en simulación

Closing legs



Opening legs



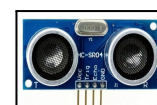
Se ha analizado el esfuerzo de cada articulación mediante el parámetro *effort* de ROS 2. Los valores se sitúan entre 0.1 y 0.25 N·m, siendo mayores en las articulaciones distales debido al efecto de palanca y al peso propio de los segmentos de la pata. Teniendo en cuenta la masa y geometría de las patas, aseguran que los actuadores seleccionados son capaces de mover la estructura del robot de forma estable y segura.

## 4.2 Sensores

El sistema de percepción del robot está compuesto por un sensor ultrasónico y una cámara frontal, integrados en la cabeza.

### - Sensor ultrasónico HC-SR04

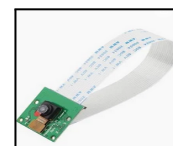
El **sensor ultrasónico HC-SR04** se utiliza para la detección de obstáculos y la navegación autónoma. Está montado sobre un servomotor que permite su rotación horizontal, funcionando como un sonar rotatorio.



**Rango de medida:** 2–400 cm

**Dimensión:** 45\*20\*15mm

### - Cámara frontal (OV5647 camera module)



La cámara frontal se conecta a la Raspberry Pi y se emplea para monitorización visual y posibles ampliaciones futuras.

### 4.3 Unidad de control

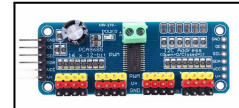
#### - Raspberry Pi 3 Model B+

Encargada de la lectura de sensores, el control de actuadores y la ejecución del software de navegación.



#### - PCA9685

Debido al elevado número de servomotores, se utiliza un **controlador PWM PCA9685**, conectado a la Raspberry Pi mediante el bus I2C, que permite controlar hasta 16 canales PWM de forma simultánea.



### 4.4 Fuente de alimentación

El robot se alimenta mediante una **batería LiPo 3S** de 11.1 V.

**Capacidad:** 1450 mAh

**Energía:** 16.1 Wh

**Descarga:** 25C / 50C



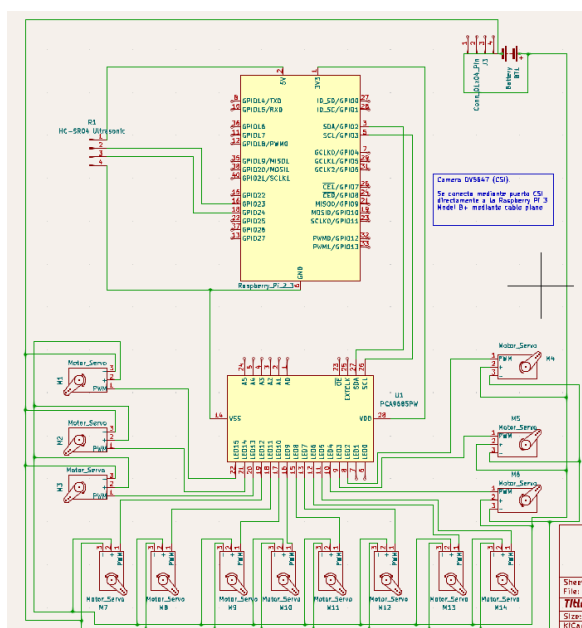
La batería alimenta el sistema a través de reguladores DC-DC que proporcionan:

5 V para la Raspberry Pi y sensores.

5–6 V para los servomotores.

Esta separación evita caídas de tensión durante picos de consumo y mejora la estabilidad del sistema. La autonomía obtenida es suficiente para completar las pruebas exigidas.

### 4.5 Diagrama eléctrico



*La Raspberry Pi actúa como unidad de control y se comunica por I<sup>2</sup>C con el PCA9685, que genera las señales PWM para los 14 servomotores. La alimentación procede de una batería LiPo 3S y se estabiliza mediante reguladores DC-DC independientes para electrónica y potencia, evitando caídas de tensión durante la marcha.*

*Está apartada la cámara que se conecta mediante puerto CSI directamente a la Raspberry Pi 3 B+ mediante cable plano.*

## 5. Sistema de control y software embebido

El sistema de control del robot se ha desarrollado sobre **ROS 2**, siguiendo una arquitectura modular basada en nodos. Este enfoque permite separar claramente las funciones de movimiento, cinemática y control, facilitando la depuración, la validación en simulación y la extensión futura del sistema.

A nivel general, el software se encarga de:

- Generar patrones de marcha estables.
- Calcular la cinemática inversa de las patas.
- Enviar consignas de posición a los actuadores.
- Permitir el control del robot durante las pruebas.

### 5.1 Arquitectura del software

El software se organiza en varios **nodos ROS 2**. La comunicación entre nodos se realiza mediante **topics**, siguiendo el modelo publicador–suscriptor.

**Principales módulos del sistema:**

- **Nodo de movimiento (movement\_publisher\_node)**  
Nodo principal del sistema. Genera las trayectorias de las patas, coordina el patrón de marcha y publica las posiciones articulares de cada pata.
- **Módulo de cinemática inversa (KI.py)**  
Implementa el cálculo matemático necesario para convertir posiciones cartesianas del extremo de la pata en ángulos articulares.
- **Interfaz de comunicación con actuadores**  
Los ángulos calculados se publican como mensajes ROS 2. En el sistema real, estos valores se transforman en señales PWM a través del controlador PCA9685 para accionar los servomotores.
- **Interfaz de usuario por teclado**  
Permite enviar comandos simples al sistema (inicio, parada, cambio de estado) mediante la publicación de mensajes en un topic.

### 5.2 Cinemática y movimiento

Cada pata del robot se modela como un manipulador de **tres grados de libertad**, compuesto por:

- Una articulación de cadera (rotación horizontal).
- Dos articulaciones en el plano vertical (segmento intermedio y distal).

El cálculo de la cinemática inversa se realiza en el archivo KI.py, mediante la función `inverse_kinematics(x, y, z)`. Esta función recibe la posición deseada del extremo de la pata y devuelve los ángulos necesarios para cada articulación.

**Proceso de cálculo:**

1. Cálculo del ángulo de la cadera mediante  $\text{atan2}(y, x)$ , que define la orientación horizontal de la pata.
2. Proyección de la posición objetivo sobre el plano horizontal para obtener la distancia radial.
3. Aplicación de relaciones trigonométricas (ley del coseno) para calcular los ángulos de las articulaciones intermedia y distal, teniendo en cuenta las longitudes físicas de los segmentos.
4. Comprobación de que la posición solicitada es alcanzable dentro del espacio de trabajo de la pata.

Este enfoque permite definir el movimiento en **coordenadas cartesianas**, simplificando la generación de trayectorias y aumentando la claridad del control del robot.

### 5.3 Patrón de marcha

El patrón de marcha se implementa en el nodo "movement\_publisher\_node" mediante una **máquina de estados finitos**. Cada estado representa una fase concreta del ciclo de marcha.

#### Fases principales del ciclo:

- Posición inicial.
- Elevación del extremo de la pata.
- Avance de la pata.
- Apoyo sobre el suelo.
- Cambio de pata activa.

El patrón de marcha prioriza la **estabilidad estática**, manteniendo siempre varias patas en contacto con el suelo mientras una de ellas se encuentra en fase de movimiento. Este enfoque resulta adecuado para las velocidades bajas requeridas y para garantizar la estabilidad durante las pruebas con carga.

### 5.4 Algoritmos de navegación

El robot implementa un algoritmo de navegación autónoma de tipo reactivo basado en el sensor ultrasónico frontal. El sistema mide continuamente la distancia a obstáculos y, cuando esta desciende por debajo de un umbral predefinido, modifica la dirección del movimiento para evitar colisiones.

Tras el ajuste de los parámetros de detección, el robot fue capaz de desplazarse de forma autónoma dentro del área de prueba, alcanzando las paredes colindantes sin colisionar, cumpliendo los requisitos del proyecto.

### 5.5 Interfaz de usuario

La interfaz de usuario se basa en el envío de comandos mediante el mando de ps4 a través de topics ROS 2, permitiendo iniciar y detener el movimiento del robot de forma sencilla y segura.



## 6. Pruebas experimentales y validación

### 6.1 Metodología de pruebas

Las pruebas experimentales se realizaron con el prototipo físico final del robot, completamente ensamblado y alimentado exclusivamente mediante batería, cumpliendo las restricciones del enunciado.

Para la Prueba 1, el robot se colocó paralelo a una pared plana y se le añadió una carga de 250 g sobre el chasis. Se activó el patrón de marcha y se midieron la distancia recorrida, el tiempo empleado y la estabilidad durante el desplazamiento.

Para la Prueba 2, el robot se situó en el centro de un área cuadrada de 2 m<sup>2</sup>. Utilizando el sensor ultrasónico frontal, el robot detectó obstáculos y paredes, modificando su trayectoria para evitar colisiones y alcanzar las paredes colindantes.

Ambas pruebas se repitieron varias veces para comprobar la consistencia del comportamiento observado.

### 6.2 Resultados de la Prueba 1

Ensayo	Distancia(m)	Tiempo (s)	Carga (g)	Resultado
1	1.02	48	250	Superado
2	1.01	52	250	Superado
3	1.03	50	250	Superado

Se cumple la prueba 1: 1 m paralelo a pared en < 1 min transportando 250 g

### 6.3 Resultados de la Prueba 2

Ensayo	Tiempo (s)	Obstáculos evitados	Paredes alcanzadas
1	97	3	4
2	95	2	4
3	102	4	4

Se cumple la prueba 2: navegación autónoma en < 2 min recorriendo todas las paredes.

### 6.4 Resumen requisitos y cumplimiento

Requisito	Resultado
Dimensiones $\leq 30 \times 30 \times 30$ cm	Cumplido
Alimentación por batería	Cumplido
Carga 250 g	Cumplido

Prueba 1 < 1 min	Cumplido
Prueba 2 < 2 min	Cumplido
Interfaz usuario	Cumplido

## 7. Discusión Técnica

Se identificaron problemas de tracción, estabilidad con carga y calibración de la cinemática inversa. Se solucionaron añadiendo material antideslizante en los pies, ajustando la postura para bajar el centro de gravedad y recalibrando los servomotores, logrando un funcionamiento estable en las pruebas.

## 8. Propuestas de mejora

Como líneas de mejora para una versión futura del robot se proponen:

- Integración de una unidad inercial (IMU) para mejorar la estimación de la orientación y estabilidad del robot.
- Sustitución del sensor ultrasónico por un sensor LiDAR, que permitiría una percepción más precisa del entorno.
- Implementación de algoritmos de navegación más avanzados basados en mapas o control reactivo mejorado.

## 9. Conclusiones

En este proyecto se ha desarrollado un robot cuadrúpedo funcional que integra diseño mecánico, electrónica y software, cumpliendo los requisitos establecidos.

El robot ha demostrado ser capaz de desplazarse de forma estable, transportar carga y realizar navegación autónoma básica con evasión de obstáculos. El proyecto ha permitido aplicar de manera práctica los conocimientos adquiridos, destacando la importancia de la correcta integración entre los distintos subsistemas.

## 10. Referencias y documentación consultada

### ROS 2 Documentation

<https://docs.ros.org/en/rolling/>

### MG90S Servo Motor Datasheet

<https://components101.com/motors/mg90s-metal-gear-servo-motor>

### HC-SR04 Ultrasonic Sensor Datasheet

<https://cdn.sparkfun.com/datasheets/Sensors/Proximity/HCSR04.pdf>

### PCA9685 16-Channel PWM Driver Datasheet

<https://www.nxp.com/docs/en/data-sheet/PCA9685.pdf>

### Raspberry Pi 3 Model B+ Documentation

<https://www.raspberrypi.com/documentation/>