

Diseño de un robot cartesiano ciberfísico con integración hardware in the loop

- Adan Rodrigo De la Cruz Osorio 192637
- Daniela Yazmin Tarango Montes 201153
- Jose Orlando Martinez Gonzalez 201471
- Bernardo David Burciaga Medina 201489

INTRODUCCIÓN

- PROYECTO: Diseño de robot cartesiano con movimiento en ejes X,Y y Z.
- Se usara Hardware-In-the-Loop (HIL) para simular el sistema en entorno virtual conectado al hardware real.
- Herramientas a utilizar:
 - SolidWorks (diseño mecánico).
 - ROS2 y Gazebo (Simulación y comunicación).
 - KiCad (diseño electrónico).
 - ESP32-S3 (controlador principal).
- META: Aprender y aplicar conceptos de diseño mecatrónico en un robot funcional.



OBJETIVOS

OBJETIVO GENERAL

Diseñar un robot cartesiano ciberfísico con integración hardware in the loop.

OBJETIVOS ESPECÍFICOS

- Diseñar el modelo del robot cartesiano.
- Simular el robot cartesiano en un entorno virtual.
- Construirlo físicamente.
- Programar la comunicación entre el hardware y la simulación.

MARCO TEÓRICO

Robots cartesianos

- Movimiento rectilíneo en 3 ejes.
- Precisión y modularidad.
- Usos: impresión 3D

Gazebo

- Simulador 3D que modela física realista.
- Se utilizó para previsualizar y probar el robot antes de fabricarlo .

HIL (Hardware-In-the-Loop)

- Pruebas de controladores reales usando simulaciones virtuales.
- Asegura seguridad y pruebas eficientes sin dañar hardware

EDA y KiCad

- Diseño de circuitos electrónicos en PCB usando KiCad.
- Se diseñó una PCB especial para el control del robot.

ROS2 y RViz2

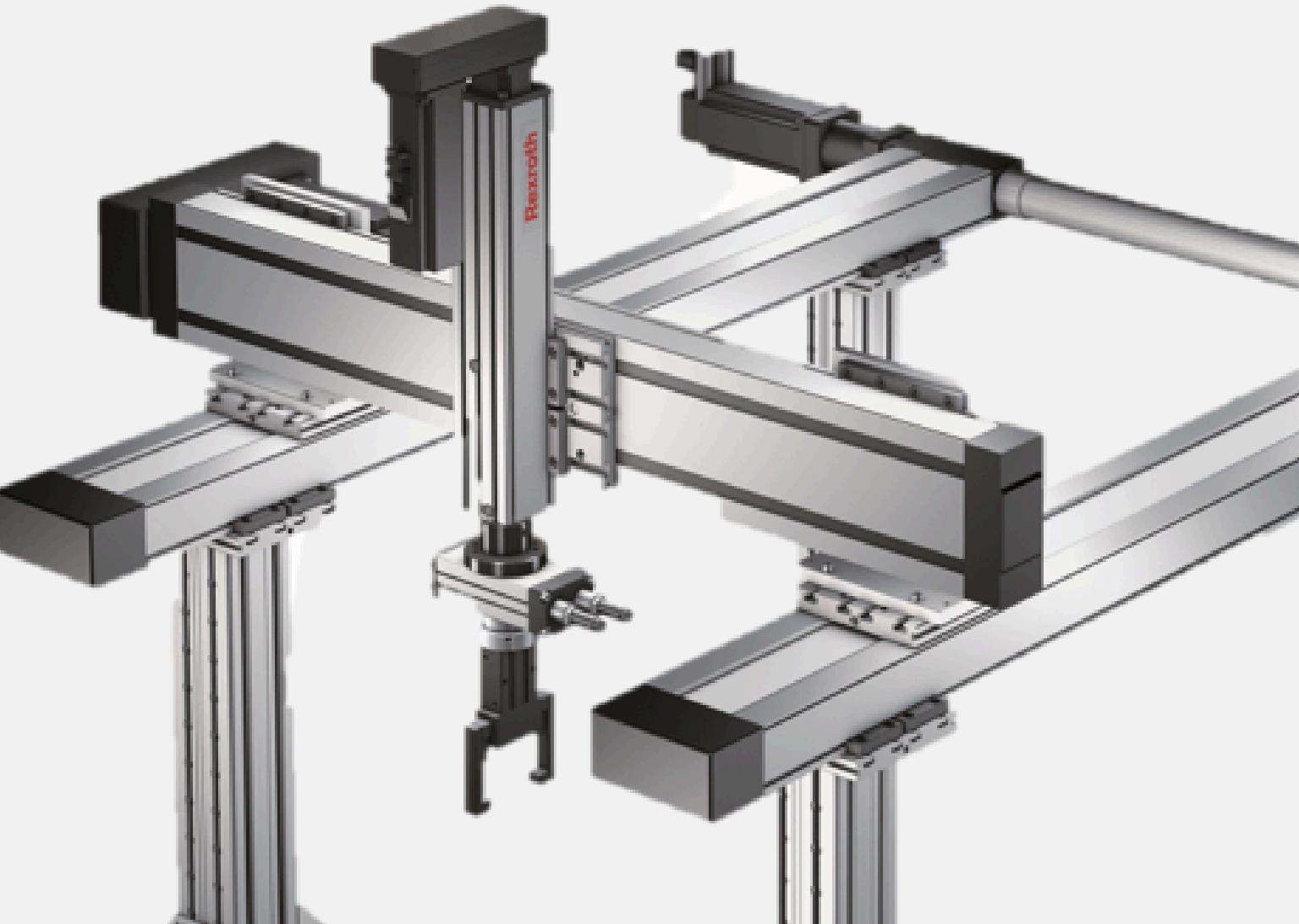
- Framework para controlar y visualizar robots en simulaciones reales.
- Visualizador 3D (RViz2) para monitorear el robot.

URDF y SDF

- URDF: Formato de descripción de robots en ROS2.
- SDF: Formato extendido para simulaciones avanzadas en Gazebo.

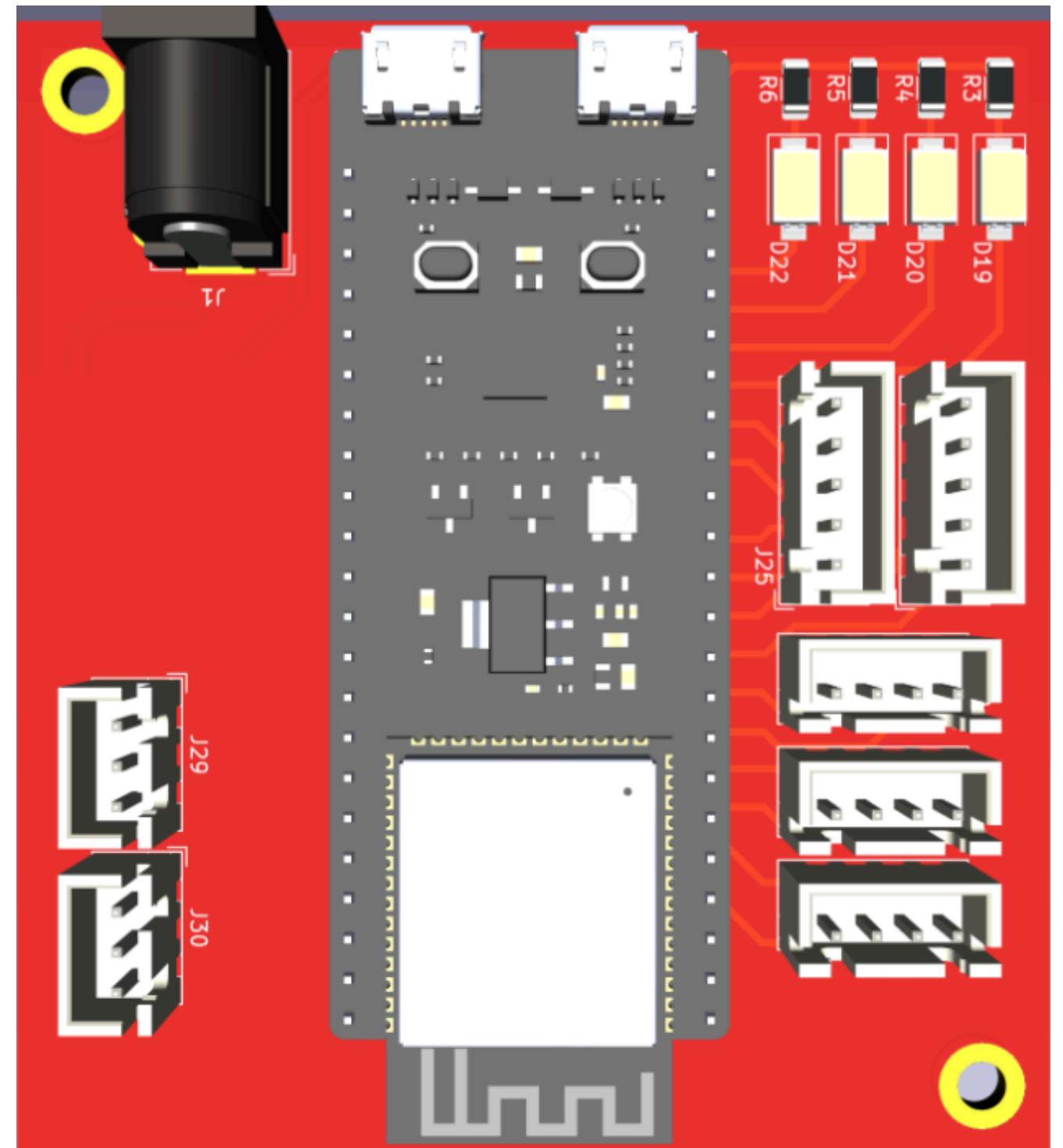
CONCEPTO Y PLANIFICACIÓN

- El concepto de este proyecto es simplicidad, bajo costo, compatibilidad HIL, modularidad.
- Fases:
 - Diseño conceptual.
 - Modelado 3D.
 - Simulación en Gazebo.
 - Construcción física.
 - Pruebas.



DISEÑO ELECTRÓNICO

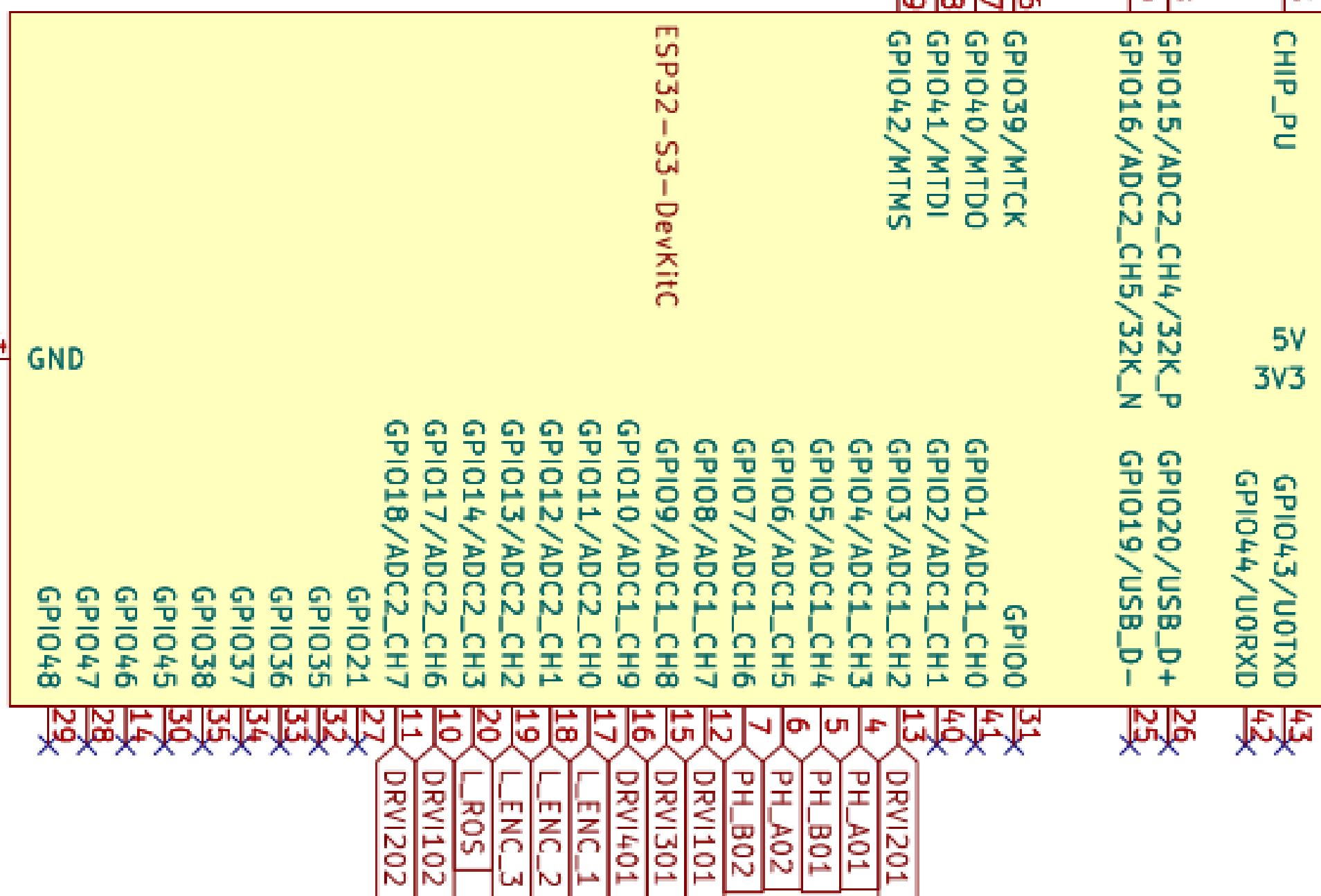
- PPCB diseñada en KiCad.
- Se diseñó una PCB que permite la conexión de:
 - La placa ESP32-S3.
 - Los drivers L298 para el control de motores.
 - Los motores con encoder.
 - Los LEDs indicadores y conectores JST para la alimentación de 12V.
- El circuito fue exportado en formato Gerber para su fabricación y ensamblaje físico.



+5V +3.3V

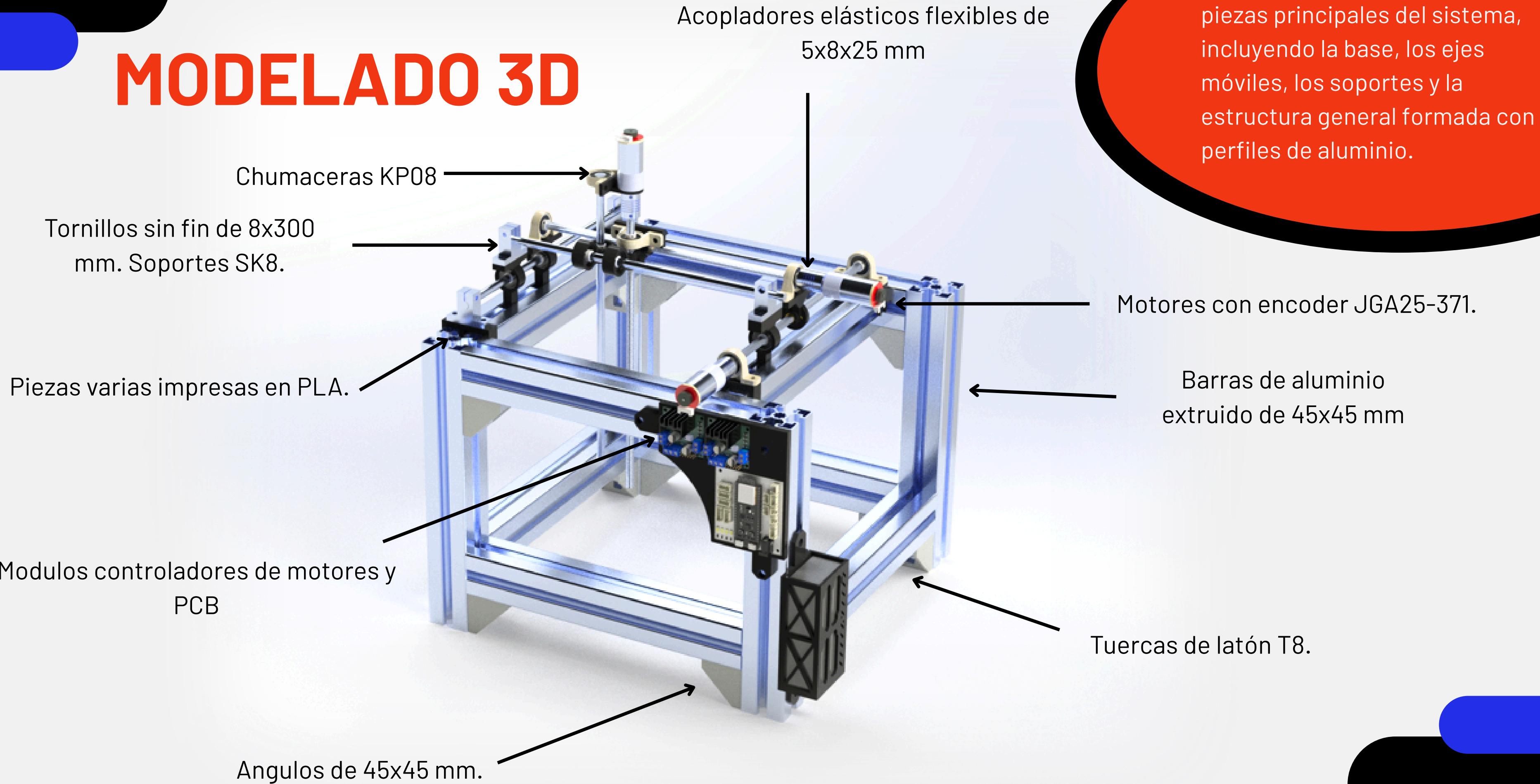
↑
21
2

X ₃	CHIP_PU	5	3	GPIO43/U0TXD	43	X
X ₂		21	2	GPIO44/U0RXD	42	X
X ₈	PH_A03	8	GPIO15/ADC2_CH4/32K_P	26	M+01	DRV0101
X ₉	PH_B03	9	GPIO16/ADC2_CH5/32K_N	25	M-01	DRV0201
X ₁₆					M+02	DRV0301
X ₁₇					M-02	DRV0401
X ₁₈	GPIO39/MTCK	36	GPIO100	31	M+03	DRV0102
X ₁₉	GPIO40/MTDO	37	GPIO1/ADC1_LCH0	41	M-03	DRV0202
X ₂₀	GPIO41/MTDI	38	GPIO2/ADC1_LCH1	40		
X ₂₁	GPIO42/MTMS	39	GPIO3/ADC1_LCH2	13		
X ₂₂			DRV1201			
X ₂₃			4			
X ₂₄			5			
X ₂₅			6			
X ₂₆			7			
X ₂₇			8			
X ₂₈			9			
X ₂₉			10			
X ₃₀			11			
X ₃₁			12			
X ₃₂			13			
X ₃₃			14			
X ₃₄			15			
X ₃₅			16			
X ₃₆			17			
X ₃₇			18			
X ₃₈			19			
X ₃₉			20			
X ₄₀			21			
X ₄₁			22			
X ₄₂			23			
X ₄₃			24			
X ₄₄			25			
X ₄₅			26			
X ₄₆			27			
X ₄₇			28			
X ₄₈			29			



U1
ESP32-S3-DevKitC

MODELADO 3D



ENTORNO DE TRABAJO

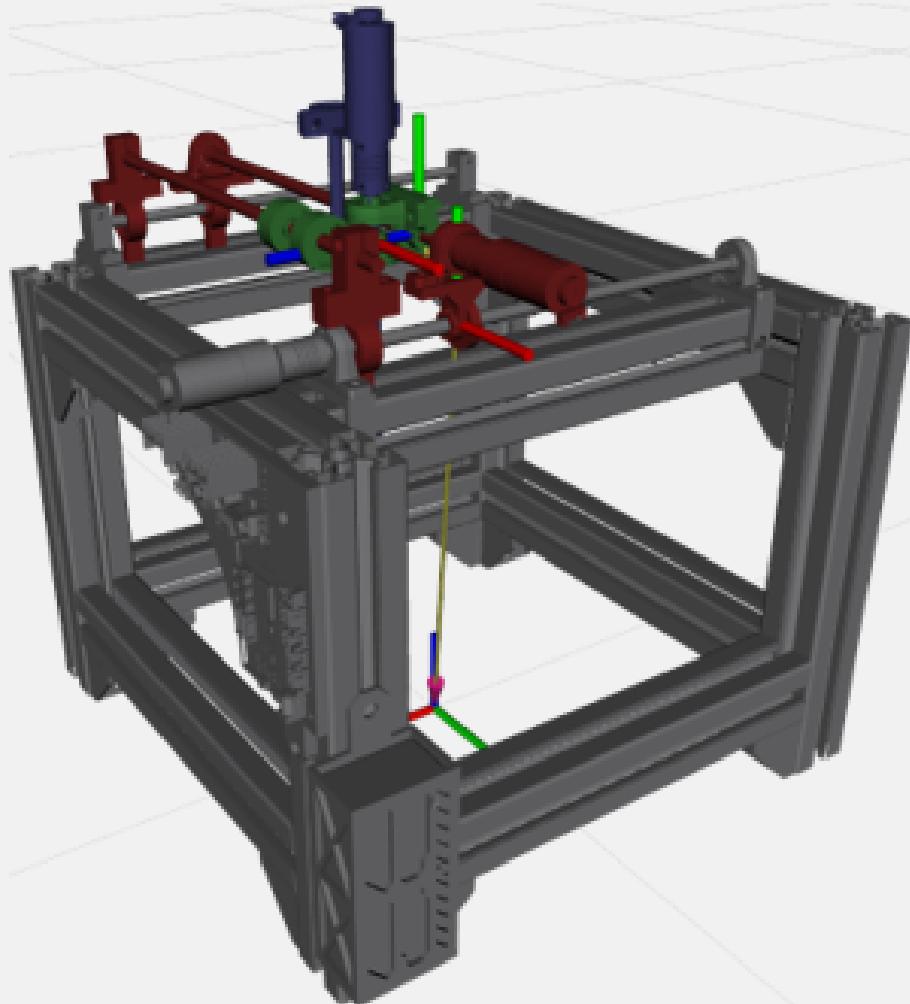
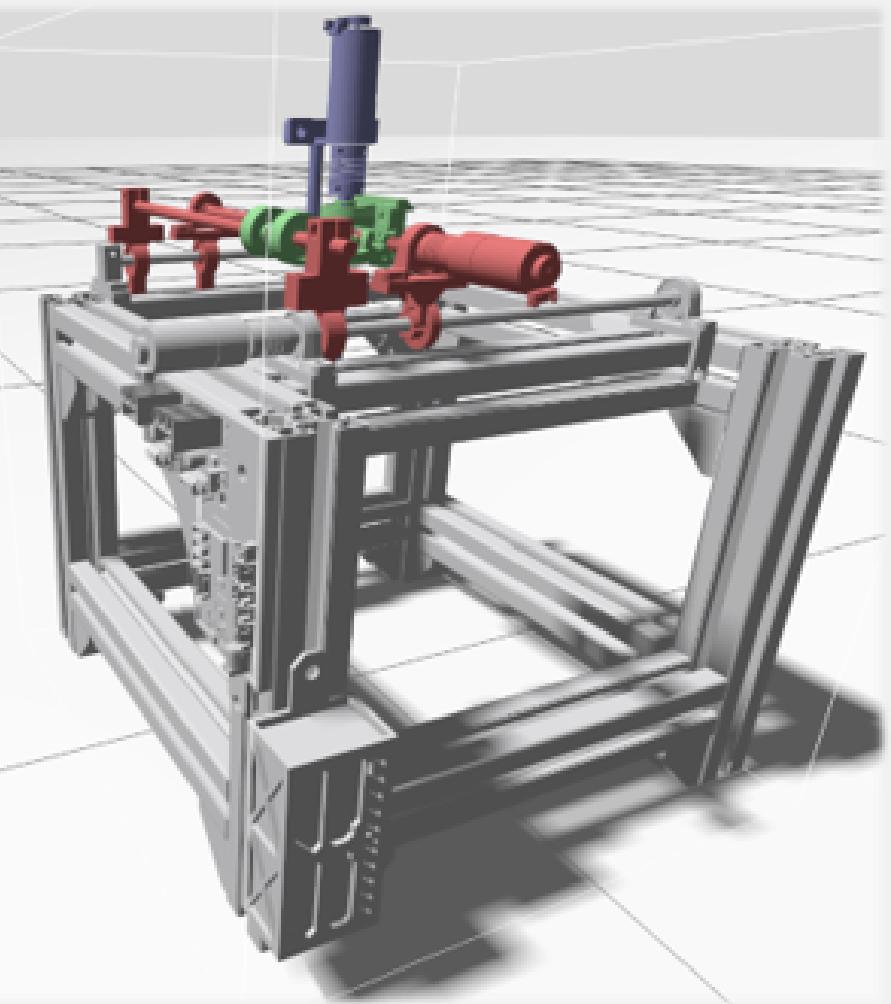
La simulación del robot se realizó en Gazebo,
utilizando ROS2 como sistema de control.

Se crearon los paquetes necesarios para:

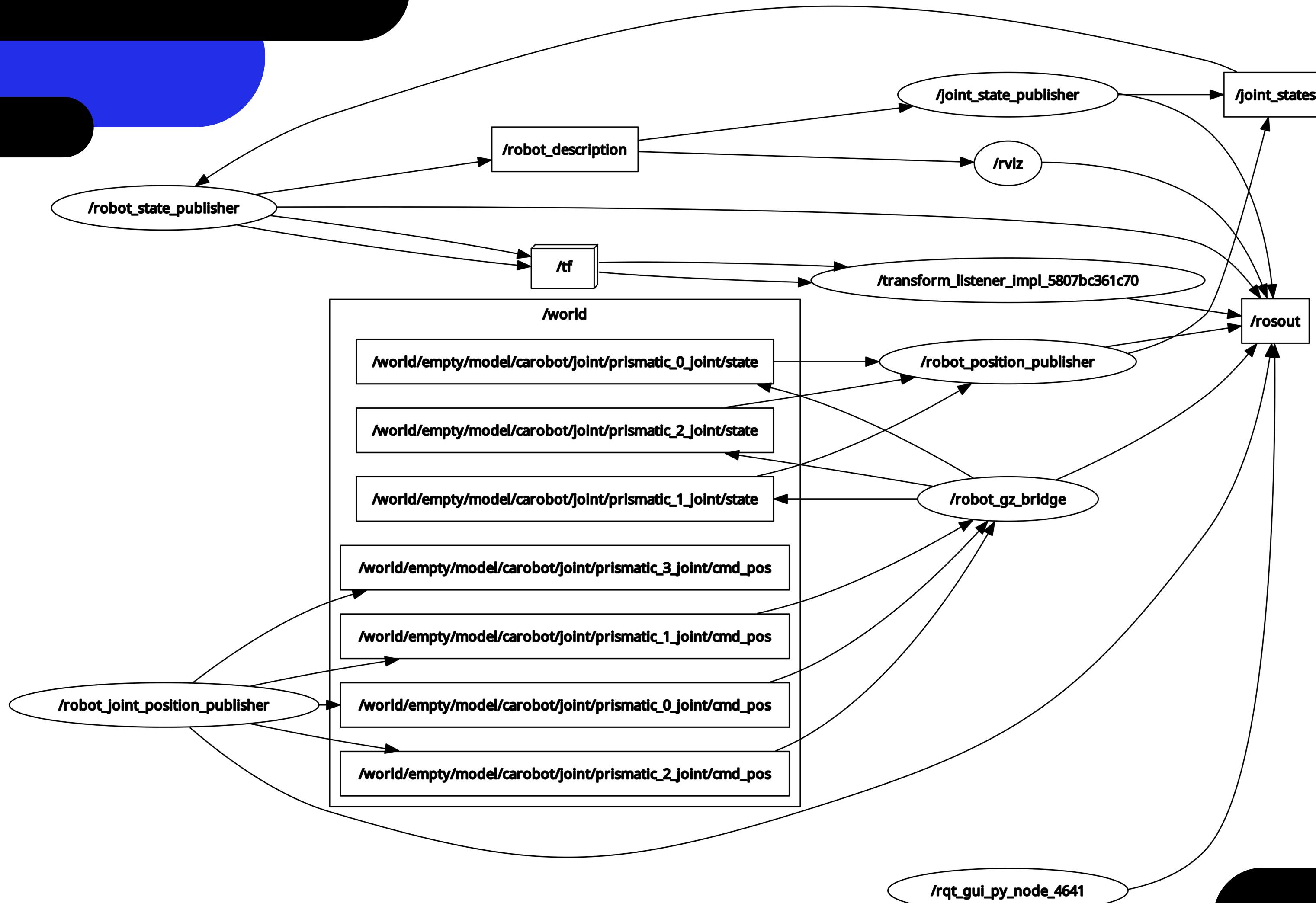
- Definir el modelo del robot (URDF/SDF).
- Publicar el estado de las juntas (\texttt{/joint_states}).
- Controlar la posición de los ejes usando plugins PID.
- Visualizar el robot en RViz2 y en el simulador Gazebo.

Comandos utilizados:

```
>> ros2 pkg create  
>> ros_gz_sim create  
>> ros_gz_bridge  
>> joint_state_publisher
```



TÓPICOS



PRUEBAS DE SIMULACIÓN Y CONTROL

Se realizaron pruebas tanto en el entorno virtual como en el sistema físico:

- Simulación de movimiento en Gazebo mediante controladores PID para cada eje.
- Pruebas de comunicación utilizando ros_gz_bridge para validar el envío de mensajes entre ROS2 y Gazebo.
- Visualización de estados en RViz2 mediante los tópicos \texttt{/joint_states} y \texttt{/tf}.
- Ensayos de ensamblaje para verificar la movilidad libre de los ejes físicos.

Durante las pruebas iniciales se detectaron ligeras oscilaciones en el movimiento, señalando la necesidad de realizar una futura sintonización de los parámetros PID.

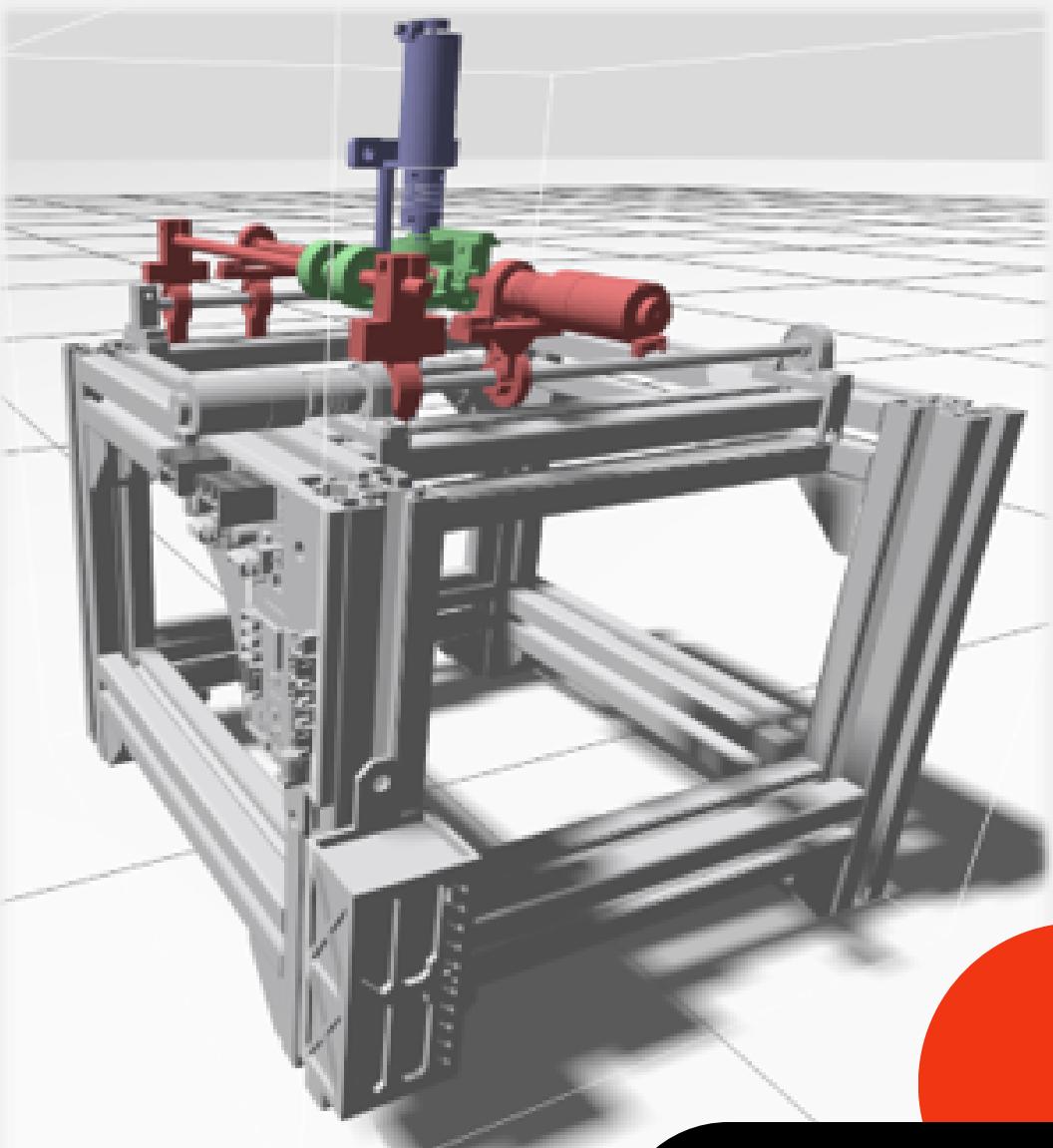
gz-sim-joint-position-controller-system

gz-sim-joint-state-publisher-system

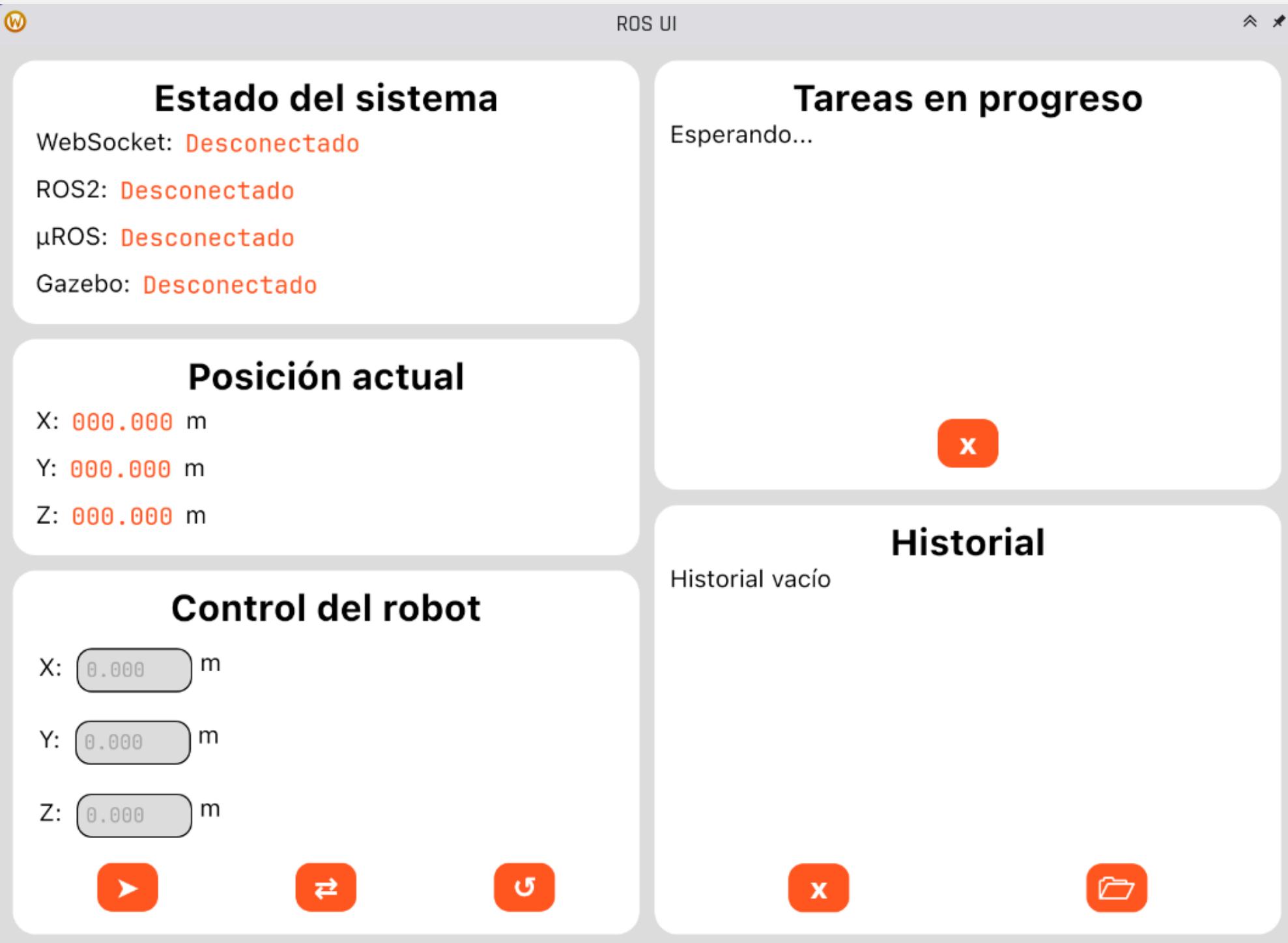
ros_gz_bridge

>>> sensor_msgs/msg/JointState [gz.msgs.Model

>>> std_msgs/msg/Float64] gz.msgs.Double



INTERFAZ GRÁFICA



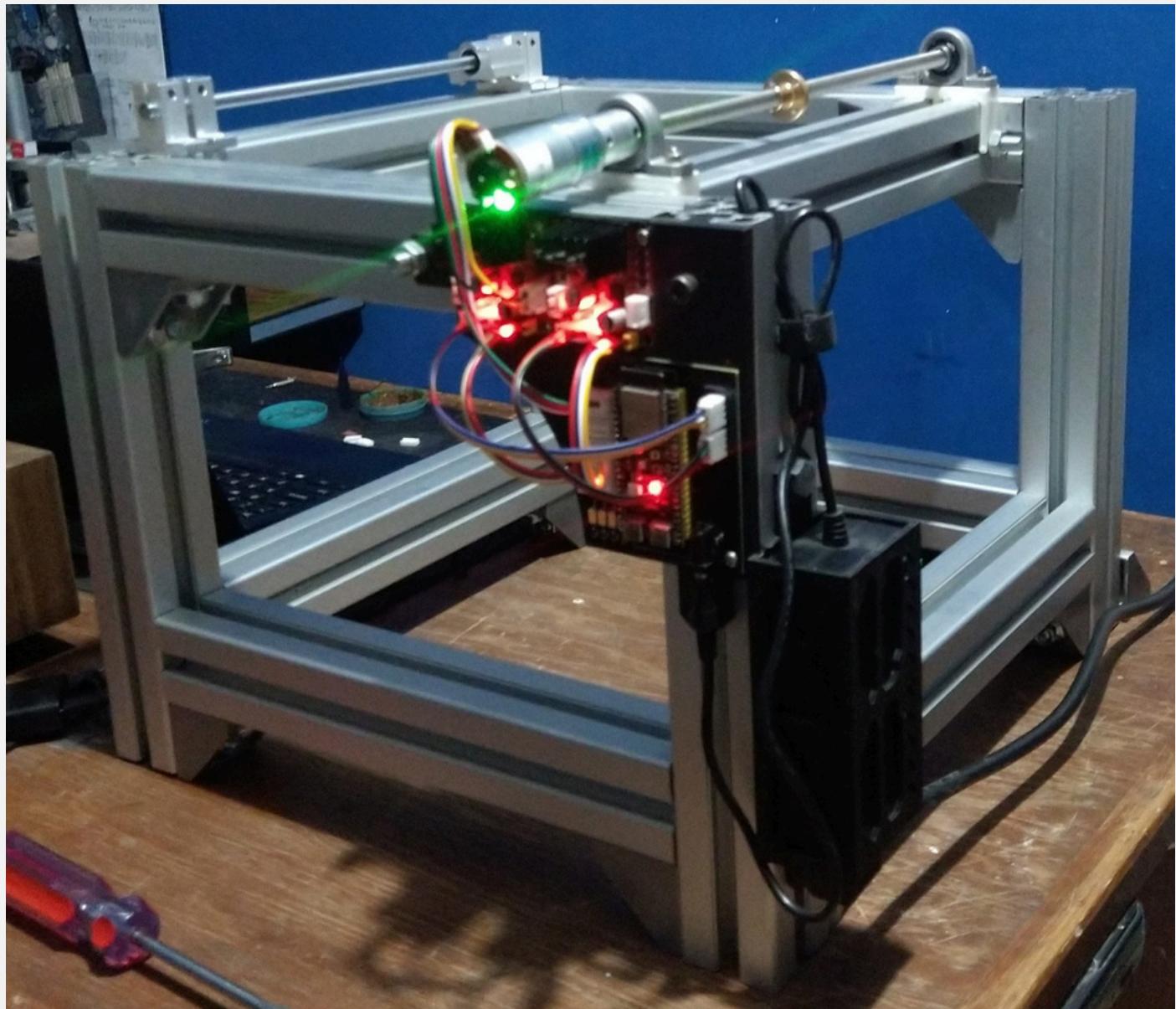
La interfaz gráfica de usuario (GUI) fue desarrollada utilizando Neutralinojs. Se creó un proyecto simple, este proyecto se basa en un archivo HTML que contiene etiquetas para:

- **Estado de comunicación:** Leer el estado de comunicación de ROS2, el socket de comunicación, μROS y Gazebo.
- **Posicion de las juntas:** Leer la posición de las juntas del robot y mostrarlas en la GUI.
- **Comandos de movimiento:** Enviar comandos de movimiento al robot.
- **Tareas:** Leer el estado de las posiciones que el robot debe alcanzar.
- **Historial de movimiento:** Visualizar el historial de movimiento del robot.

CONSTRUCCIÓN DEL ROBOT

Para la construcción del robot cartesiano se realizaron las siguientes actividades:

- Adquisición de materiales: Perfiles de aluminio, motores con encoder, tornillos sin fin, soportes y electrónica.
- Fabricación de piezas: Soportes y adaptadores diseñados e impresos en PLA.
- Ensamble mecánico: Montaje de la estructura utilizando perfiles de aluminio 45x45 mm, tornillos y ángulos de unión.
- Conexiones eléctricas: Instalación de la fuente de alimentación, drivers L298, placa ESP32-S3 y cableado de motores y sensores.



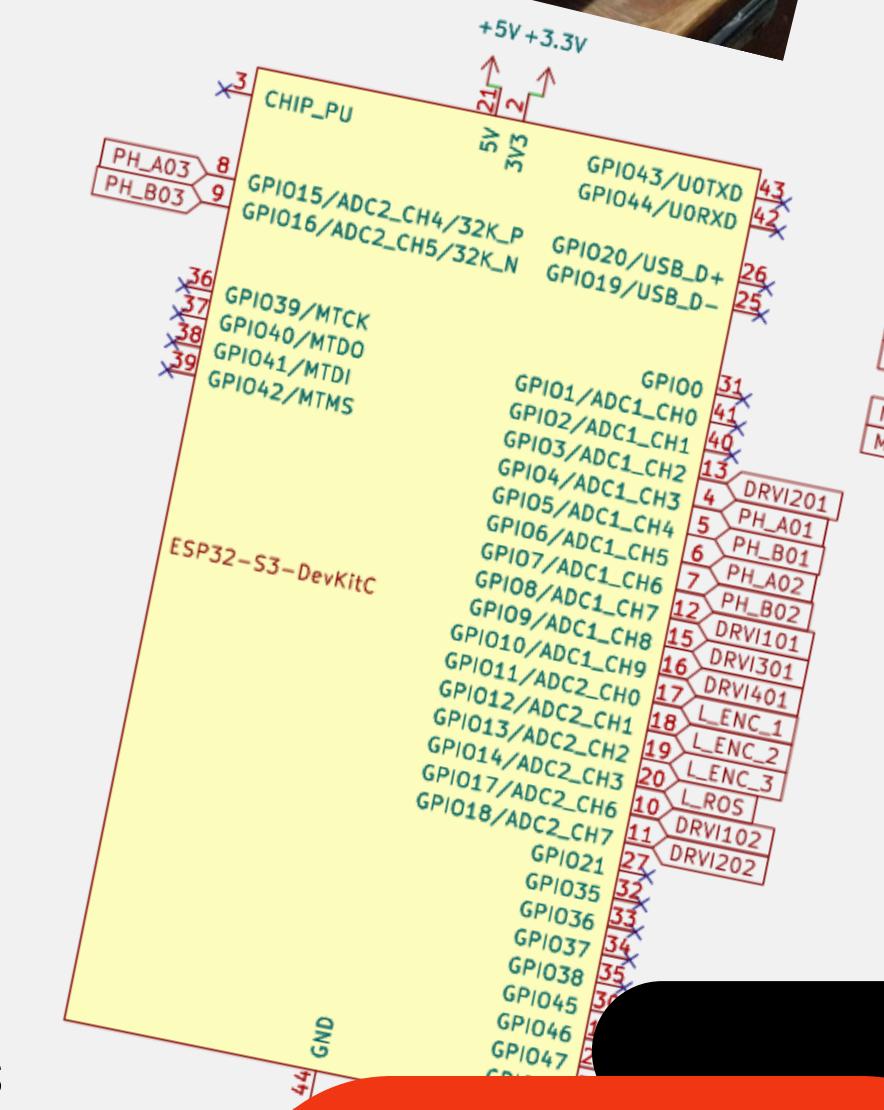
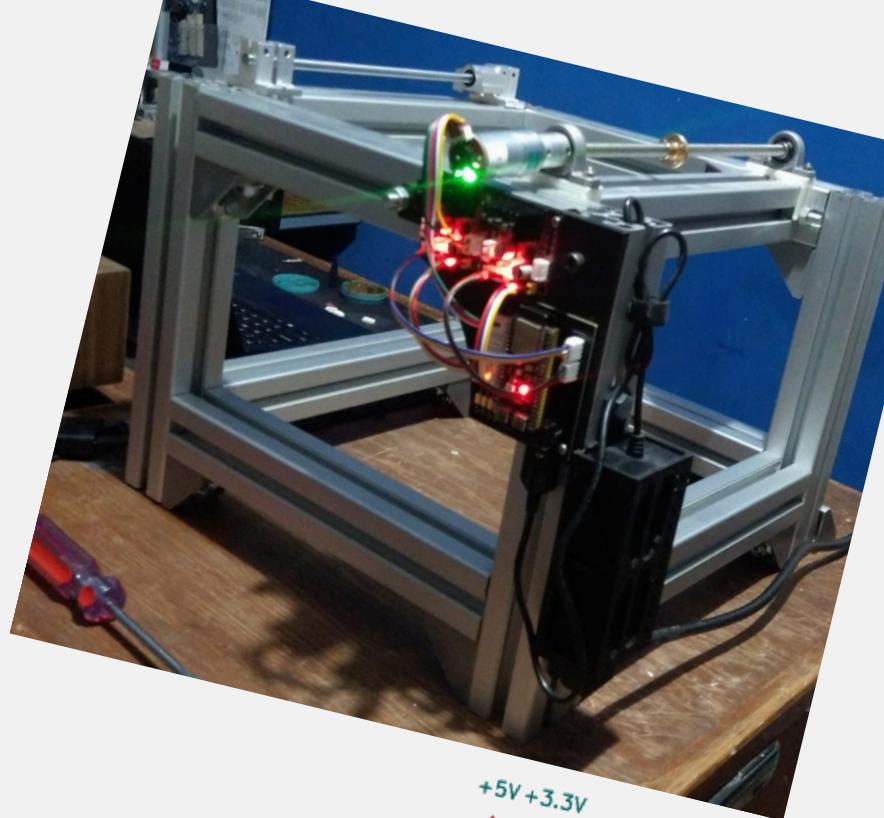
Se verificó el alineamiento de los ejes y la correcta sujeción de los componentes para asegurar la precisión del sistema.

RESULTADOS OBTENIDOS

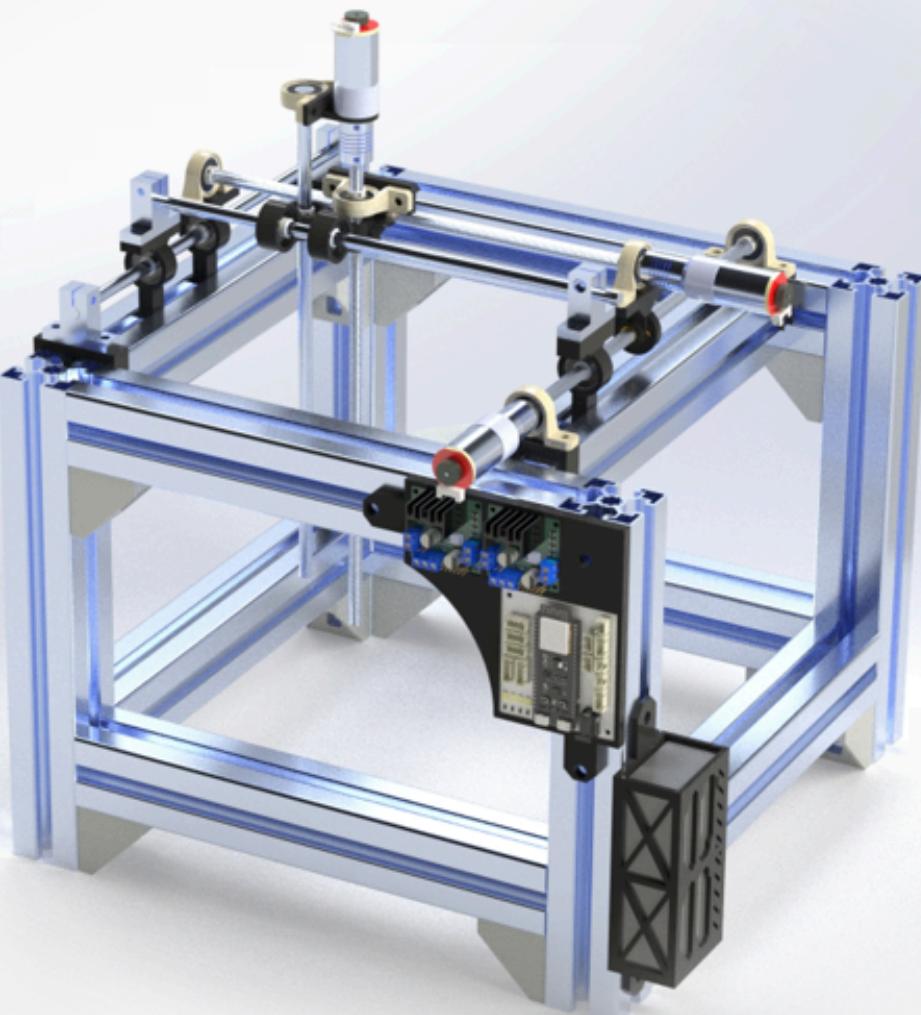
Hasta el momento se han logrado los siguientes avances:

- Modelado 3D completo del robot cartesiano en SolidWorks.
- Diseño y fabricación de la PCB para el sistema de control basado en ESP32-S3.
- Simulación exitosa del comportamiento del robot en Gazebo utilizando ROS2.
- Construcción parcial de la estructura física del robot.
- Validación inicial de la comunicación Hardware-in-the-Loop (HIL) entre el hardware real y el entorno virtual.

Estos resultados proporcionan una base sólida para finalizar el ensamblaje, ajustar el control de movimiento y realizar pruebas funcionales completas.



CONCLUSIÓN



Se ha logrado progresar en el modelado 3D, adquisición de materiales, diseño electrónico y simulación inicial del robot cartesiano. Aún queda trabajo por hacer, particularmente en la programación de controladores, integración HIL y pruebas funcionales. Sin embargo, los resultados hasta ahora son prometedores y sientan una base sólida para continuar con el proyecto.

Gracias