

# Taller 3 Diseño de un End-effector

Marcela Sabogal Guerrero

Juan José Valencia Moreno

Juan Sebastian Espinosa Arciniegas

Electrical & Electronic Engineering Dept.



10 de mayo de 2022

## **Introducción**

En el presente trabajo se exponen los resultados del taller 3 en donde se buscó emplear el entorno de ROS para darle funcionalidad a un manipulador diseñado por nosotros. Para su realización fue necesario conocer las especificaciones mecánicas, eléctricas y de diseño que requería el robot. También profundizar nuestros conocimientos en raspberry para lograr controlar el robot mediante el sistema operativo de robots (ROS) en el sistema operativo Linux.

# 1. Diseño del robot

## 1.1. Lista materiales

Componente	Referencia
1. Raspberry	Raspberry pi 4
2. Arduino	Arduino mega 2560
3. Bases acrílicas	Laminas acrílicas de 3mm con corte laser
4. Rueda loca	Rueda universal de bola de acero W420
5. Separadores	Espaciadores/postes (diferentes tamaños)
6. 2 Pilas 3.7v	Bateria 18650 LiPo 3.7v 5800mAh
7. Jumpers	cables puente hembra-macho
8. Llantas	Lanta con rin para motores cuyo eje es de 3.8 mm
9. Puente H	Motor DC puente H L298 Arduino robótica
10. Motores	Motores con encoders
11. Regulador de voltaje	Modulo LM2596 regulador de voltaje DC-DC
12. Cable plug jack	Conector plug jack 2-1 DC 2.1mm
13. Adaptador tipo C	Conector USB tipo C
14. Manipulador	Brazo Robotico Arduino 4 Servos 4 Dof Grados Libertad Sg90
15. Servo motores	MicroServo SG90

Cuadro 1: Materiales empleados en el diseño del robot

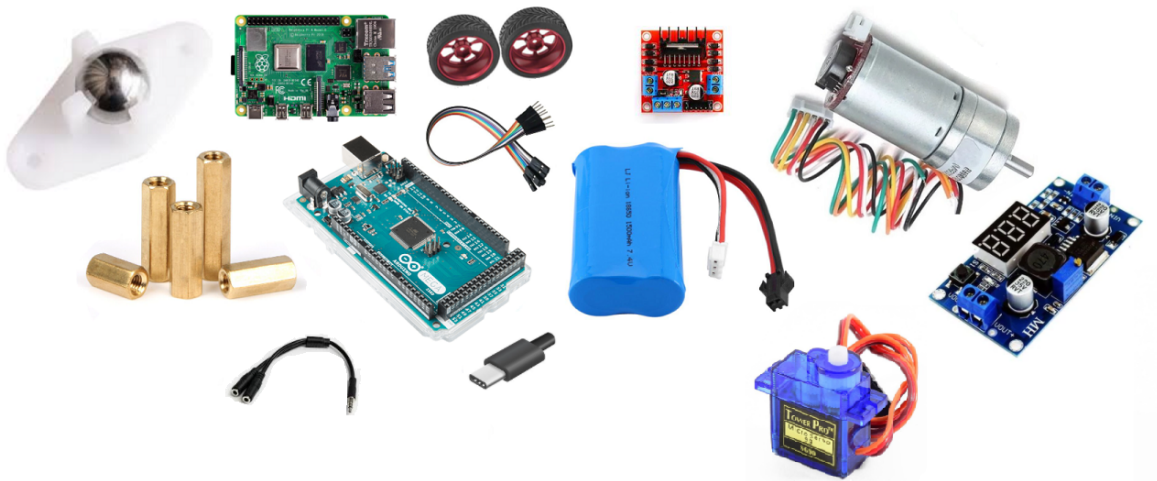


Figura 1: Materiales empleados en el diseño del robot

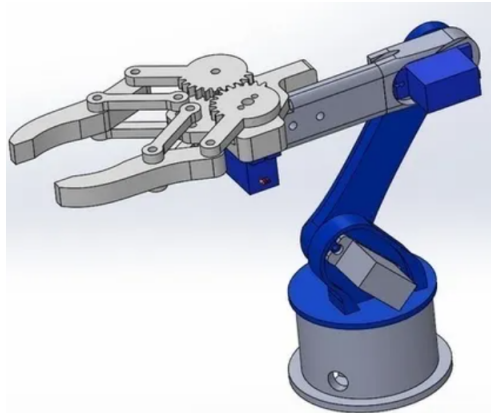


Figura 2: Manipulador robótico

## 1.2. Plano eléctrico/electrónico

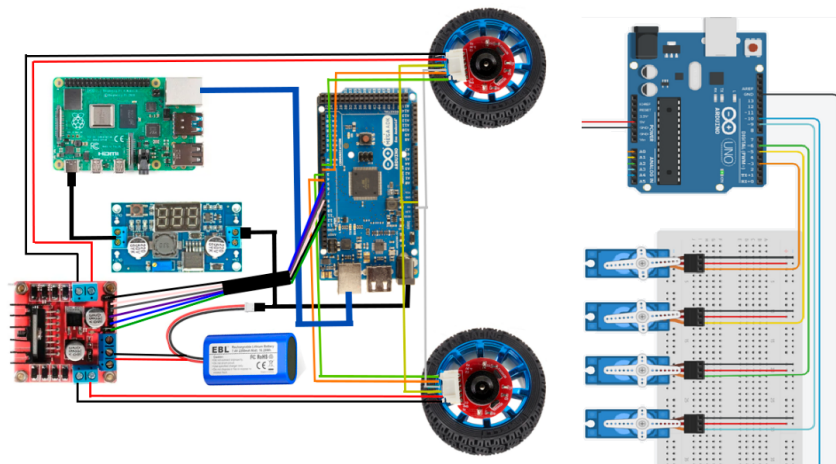


Figura 3: Plano eléctrico empleado

## 1.3. Descripción de funcionamiento del robot

El brazo está alimentado por una batería propia. Está compuesto por 4 servomotores; uno se encarga del giro de la base; otro se encarga de la altura; otro se encarga de la distancia; y el último se encarga del movimiento de la garra. El brazo tiene una extensión máxima de 18 centímetros sin contar la garra. Los servomotores implementados son SG-90, estos 4 servos están alimentados por 5V.

Para el funcionamiento del brazo, se crearon 3 nodos de ROS y dos nodos de arduino. El primer punto utilizó un nodo de arduino que recibe datos tipo String los cuales son enviados a partir de un

keylistener por medio de python. Estos String, al llegar al código de arduino, ejecutan el movimiento de un servomotor específico. Esto permite el funcionamiento del brazo. El segundo nodo de arduino, recibe datos tipo Twist que incluye la magnitud de los ángulos de los motores del brazo. Esto le permite una mayor precisión a la hora de efectuar cinemática directa e inversa. Los otros dos nodos de python del brazo se encargan de efectuar cálculos de cinemática inversa y directa para poder calcular la posición del brazo con base en los ángulos de los servos y enviar ángulos a los servos con el fin de llegar a una posición específica.

## 1.4. Integración de la parte eléctrica y mecánica

En el diseño de nuestro robot buscamos que los componentes empleados tuvieran la mejor distribución/ localización optimizando el cableado y su estética, de esta manera facilitar el entendimiento de los puertos usados en cada sección de robot. Como se mencionó previamente la principal integración de la parte eléctrica con la mecánica se da al emplear los motores con sus encoders. En la Figura 13 se presenta la distribución de la alimentación que logró alimentar y enlazar todos los componentes implicados. La parte eléctrica nos permitió tener una comunicación y alimentación asertiva para dar cumplimiento con los requerimientos que tenía el robot. La integración de esta con la parte mecánica fue de suma relevancia para poder establecer correctamente la información que recibían tanto los encoders como el puente de H que logrará la dirección y alimentación específica que recibían los motores para ejecutar la velocidad lineal, angular y las direcciones a las que debían ir. La activación y dirección de los motores es controlada por el puente de H el cual invierte la polaridad de la conexión eléctrica del motor en cuestión, logrando de alguna manera intercambiar los cables que alimentan el dispositivo.

También fue necesario emplear un brazo robótico con cuatro grados de libertad para poder manipularlo y cumplir con los requerimientos. Para el manipulador se emplearon 4 servos para cada grado de libertad, para su alimentación se requirió de un banco de baterías adicional para poder alimentarlos con 5 voltios y la corriente necesaria que requerían los servos. Cada uno de los servos contaba con su respectiva alimentación y cable PWM conectado a el arduino mega para transmitir la señal de forma digital, esto se realiza modificando el ciclo de trabajo de una señal periódica.

## 2. Solución

### 2.1. Teleoperación:

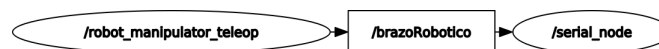


Figura 4: Nodos empleados

Para esta parte, fue necesario, crear el nodo `/robot_manipulator_teleop.py`, en ROS, el cual recibe como entradas, las teclas ( $\uparrow$ ), ( $\downarrow$ ), para mover el servo 1 ( $\leftarrow$ ), ( $\rightarrow$ ) para controlar el movimiento de la base y ( $'w'$ ), ( $'s'$ ) para el movimiento del arriba y abajo respectivamente del servo 2.

A continuación, se presenta el macro-algoritmo para este item

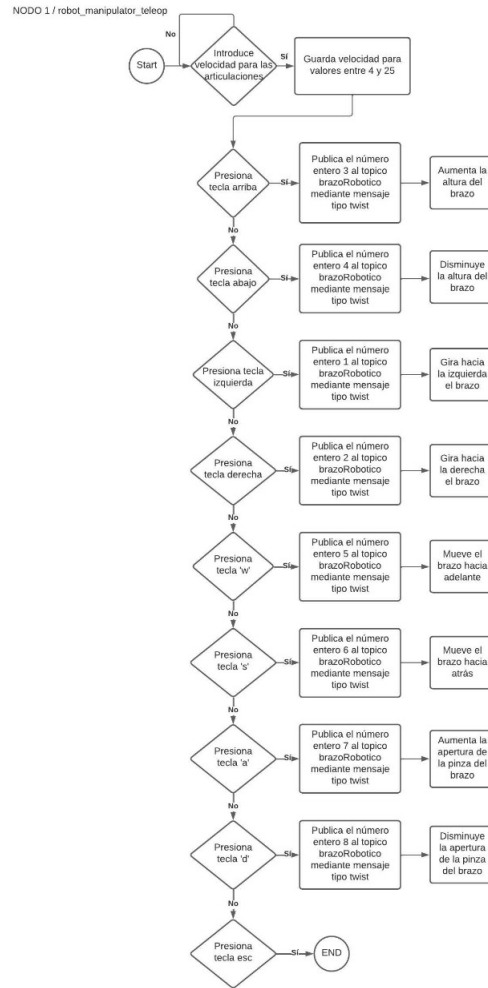
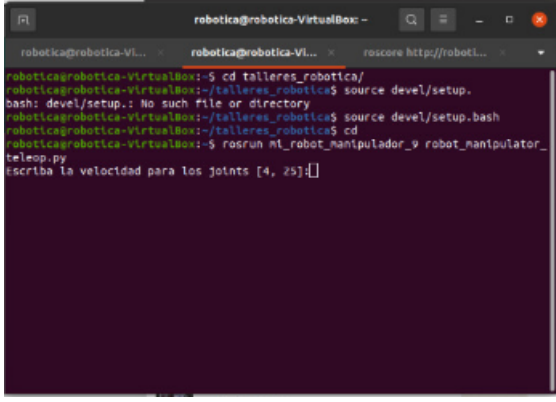


Figura 5: Macroalgoritmo punto 1.



```
robotica@robotica-VirtualBox: ~$ cd talleres_robotica/
robotica@robotica-VirtualBox:~/talleres_robotica$ source devel/setup.bash
bash: devel/setup.: No such file or directory
robotica@robotica-VirtualBox:~/talleres_robotica$ source devel/setup.bash
robotica@robotica-VirtualBox:~/talleres_robotica$ cd
robotica@robotica-VirtualBox:~$ roslaunch mt_robot_manipulador_v robot_manipulador_
teleop.py
Escriba la velocidad para los joints [4, 25]:
```

Figura 6: Definición del valor de la velocidad de cada junta

Como se puede observar en la Figura ?? se emplea el nodo `/robot_manipulador_teleop` para poder ingresar las velocidades de cada junta para controlar de forma directa el robot con las teclas del computador.

## 2.2. Ruta:

Para este caso, se creó el nodo, `/robot_manipulador_interface.py`, el cual grafica la posición de la garra de nuestro end-effector para cualquier posición dentro de el **volumen de trabajo calculado** se hizo uso de las ecuaciones de cinemática de cinemática directa para un end-effector en 2D, i.e.,

$$x = a_1 \cos(\theta_1) + a_2 \cos(\theta_1 + \theta_2)$$

$$y = a_1 \sin(\theta_1) + a_2 \sin(\theta_1 + \theta_2)$$

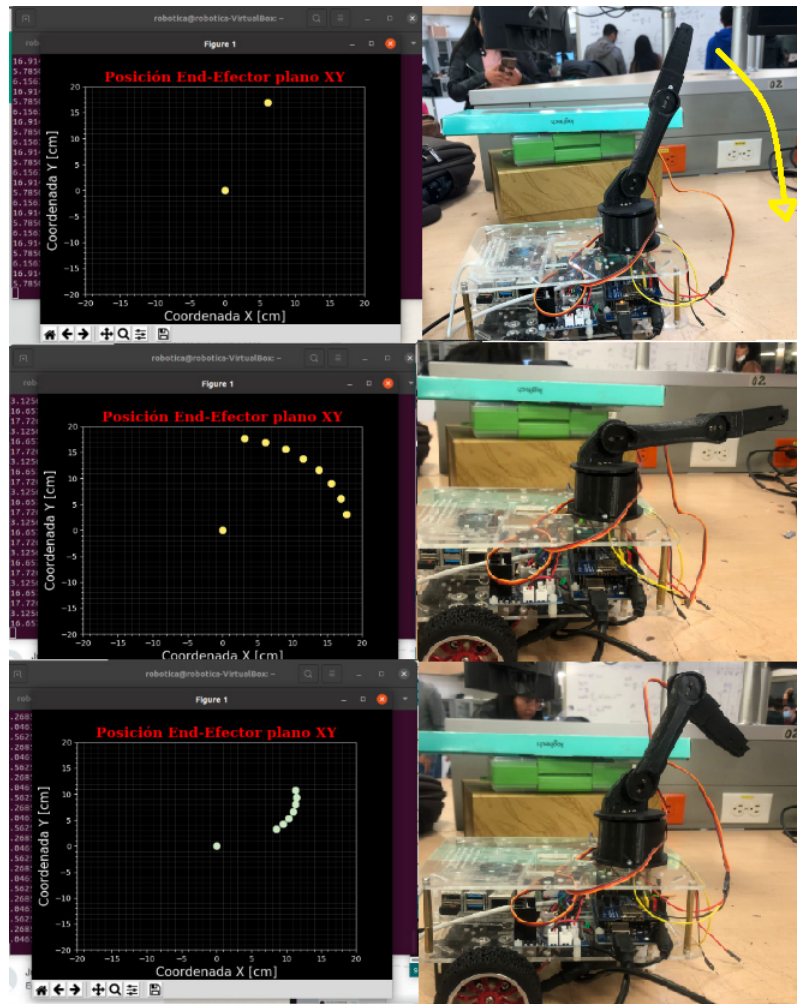


Figura 7: Interfaz gráfica para visualizar la posición en tiempo real del robot

Como se presenta en la Figura 7 en la interfaz gráfica creada se puede observar las posiciones en tiempo real del end-effector. En esta imagen se observa la posición inicial 0 y el camino que recorre. a continuación, se presenta el macroalgoritmo para esta sección:

NODO 2 / robot\_manipulator\_interface

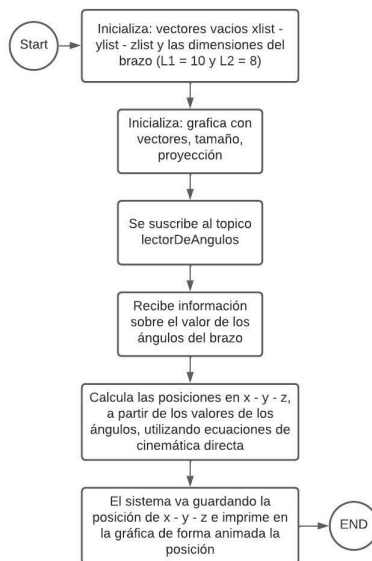


Figura 8: Macroalgoritmo punto 2.

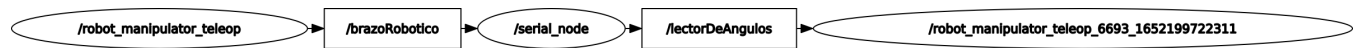


Figura 9: Nodos empleados



## 2.3. Seguidor de ruta:

NODO 3 / robot\_manipulator\_planner

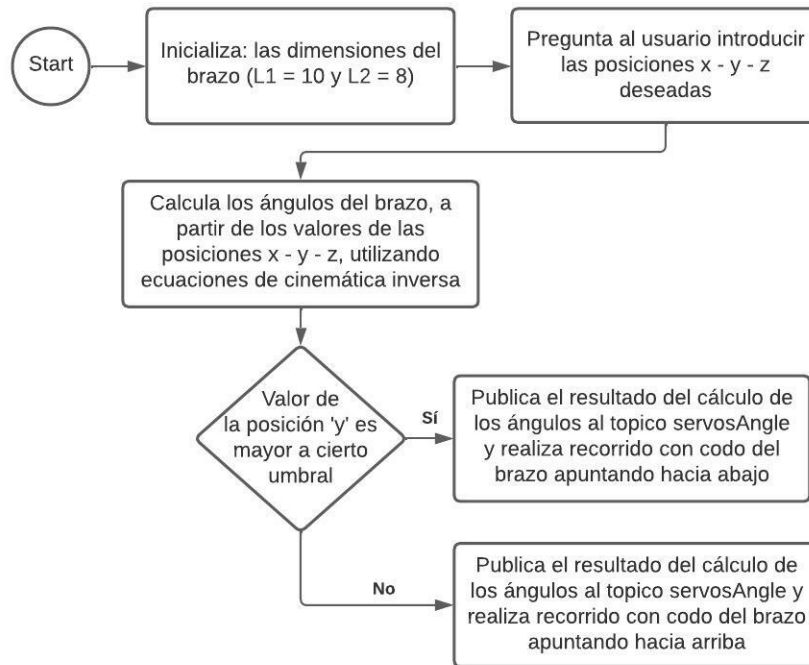


Figura 10: Macroalgoritmo



Figura 11: Nodos empleados

El seguidor de ruta ejerce cinemática inversa con el fin de que a partir de una posición en términos de  $x$ ,  $y$  &  $z$ , se puedan calcular los ángulos de los servos que corresponden a esa posición y enviarlos a los servos. Se utilizaron las siguientes fórmulas para efectuar los cálculos:

$$\cos \theta_2 = \frac{1}{2a_1a_2}((x^2 + y^2) - (a_1^2 + a_2^2))$$

$$\sin \theta_2 = \pm \sqrt{1 - \cos^2 \theta_2}$$

$$\cos \theta_1 = \frac{1}{x^2 + y^2} (x(a_1 + a_2 \cos \theta_2) \pm ya_2 \sqrt{1 - \cos^2 \theta_2})$$

$$\sin \theta_1 = \frac{1}{x^2 + y^2} (y(a_1 + a_2 \cos \theta_2) \pm xa_2 \sqrt{1 - \cos^2 \theta_2})$$

Figura 12: Fórmulas empleadas

Estas fórmulas permitieron calcular los ángulos de cada Servo a partir de una posición en  $x$ ,  $y$  &  $z$  dada

Por otro lado, el volumen de trabajo es un anillo con radio interno de  $|l_1 - l_2| = 2cm$  y un radio externo de  $l_1 + l_2 = 18cm$ , donde  $l_1 = 10cm$  y  $l_2 = 8cm$ .

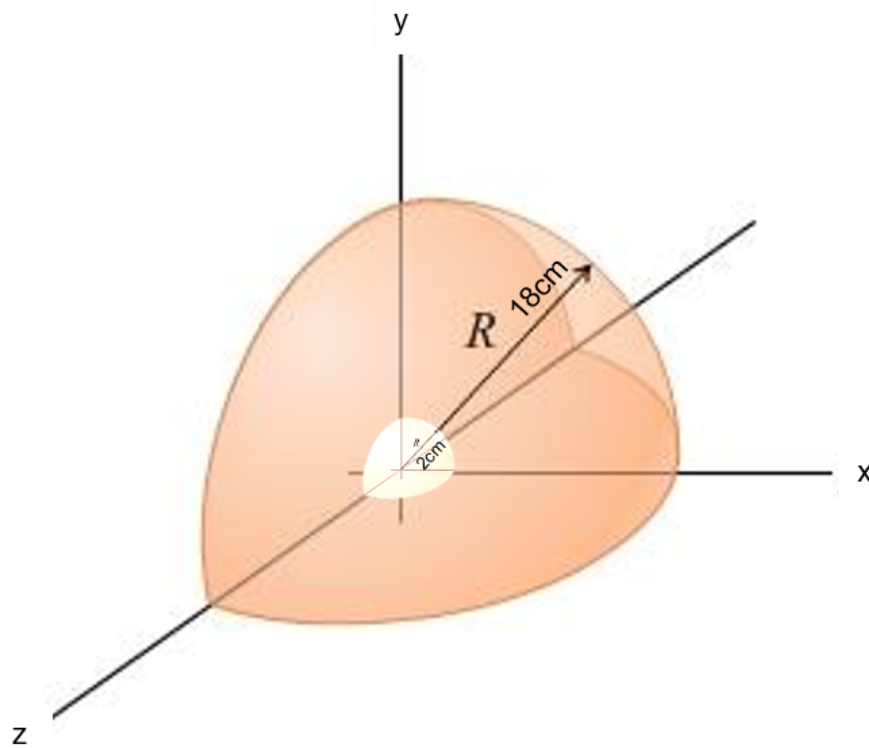


Figura 13: Volumen de trabajo del brazo

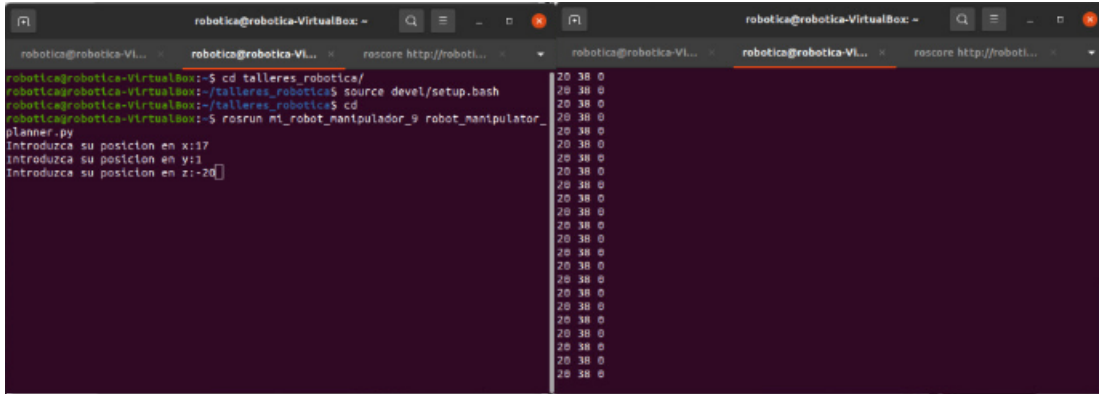


Figura 14: Llevar el end-effector del robot a una posición destino deseada

Como se observa en la Figura 14 se establece una posición del end-effector dentro de nuestro volumen de trabajo, luego el robot realiza el calculo de cinemática inversa para alcanzar la posición establecida; en la imagen de la derecha se observa los ángulos calculados dadas las coordenadas  $(x, y, z)$  deseadas con el siguiente orden (*angulo servo 1, angulo servo 2, angulo servo base*).