

HITO 2

CAPTURA Y VISUALIZACIÓN DE SEÑALES DEL ROBOT

2 0 2 4 - 2 0 2 5

Asier Bujedo

Landes Gallastegui
Marta Gorraiz Irene
Gonzalez
Iker Marcelo

ÍNDICE

1.	TÓPICOS Y NODOS CON LOS QUE INTERACTÚA	1
2.	ARQUITECTURA Y COMUNICACIÓN ENTRE NODOS	2
3.	INTEGRACIÓN	3
4.	REGISTRO DE DATOS EN INFLUXDB	4
5.	TRAYECTORIAS	6

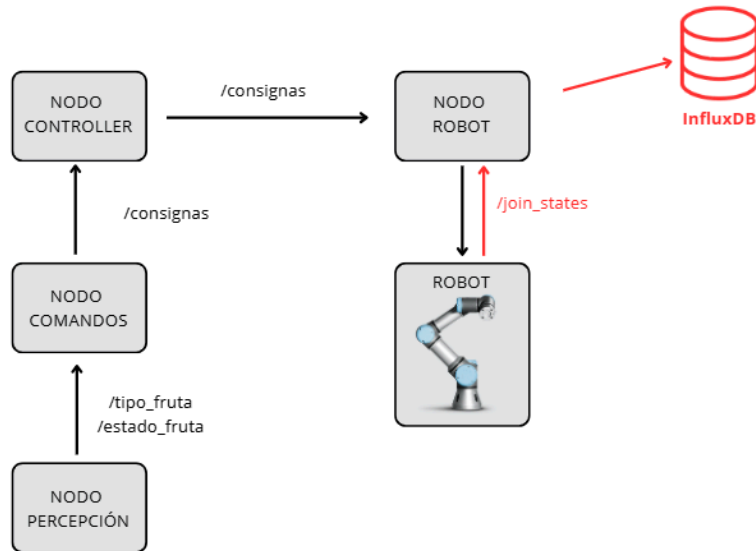
1. TÓPICOS Y NODOS CON LOS QUE INTERACTÚA

- **Nodo de percepción** (pendiente de implementación):
 - **Función:** Procesar las imágenes captadas por la cámara para identificar el tipo y estado de las frutas (buen o mal estado) usando técnicas de visión por computador y Machine Learning.
 - **Comunicación:**
 - Publica en el topic **/consignas** un mensaje que indica la acción a realizar basada en la clasificación y estado de la fruta.

- **Nodo controller** (*controller.py*):
 - **Función:** Coordinar las acciones del sistema. Escucha las consignas del nodo de percepción y traduce estas instrucciones en una secuencia de comandos para el brazo robótico
 - **Comunicación:**
 - Se suscribe en el topic **/consignas** para recibir las órdenes del nodo de percepción.
 - Implementa la lógica de secuenciación de las acciones.

- **Nodo control robot** (*control_robot.py*):
 - **Función:** Este nodo controla el brazo robótico, y se encarga de capturar el estado de las articulaciones publicadas por el robot y registrar las métricas en InfluxDB.
 - **Comunicación:**
 - Se suscribe en el topic **/consignas** para recibir las órdenes del nodo controller.
 - Ejecuta movimientos basados en posiciones predefinidas.
 - Se suscribe al topic **/joint_states** para recibir información sobre el estado de las articulaciones.

2. ARQUITECTURA Y COMUNICACIÓN ENTRE NODOS



El sistema utiliza un topic para la comunicación entre los nodos, adoptando un esquema asíncrono, es decir, este enfoque asegura que los mensajes se transmitan de manera continua entre los diferentes componentes del sistema. Los nodos pueden enviar información sin necesidad de esperar una respuesta inmediata, lo que permite que cada componente realice sus tareas de forma independiente mientras procesa los datos a su propio ritmo.

El topic **`/consignas`** de tipo `std_msgs/Int32`, actúa como el canal de comunicación donde se publican y suscriben las instrucciones que definen las acciones a realizar. El Nodo de percepción y el nodo controller actúan como emisores en el topic. Su función principal es analizar los datos disponibles y publicarlas en dicho canal de comunicación.

El nodo robot, es responsable de ejecutar las acciones del brazo robótico, como mover sus articulaciones según las consignas recibidas del nodo *controller*. Además de ejecutar estas acciones, el nodo *Robot* también publica información sobre el estado de las articulaciones en el topic **`/joint_states`**, permitiendo así que esta información sea almacenada en InfluxDB para su posterior análisis.

3. INTEGRACIÓN

El esquema de comunicación asíncrono del Hito 2 se basa en el establecido en el Hito 1, en el que se definieron los nodos y el topic **/consignas**.

El primer paso para la integración fue la ampliación del nodo control_robot para suscribirse al topic **/joint_states** y gestionar la captura y registro de datos en InfluxDB. Además, ahora el nodo es capaz de recopilar datos de las juntas en tiempo real con el objetivo de mejorar la capacidad de monitoreo y optimización del comportamiento del brazo.

Se ha empleado el nodo control_robot para planificar trayectorias mediante MoveIt. Esto permite que tras recibir una consigna, el nodo realice una secuencia de movimientos planificados y controlados. MoveIt se emplea para definir puntos de destino y generar trayectorias suaves y seguras.

Se integró la clase InfluxDBHandler para gestionar la conexión a la base de datos InfluxDB. Esto permite almacenar únicamente los datos útiles, es decir, cuando el robot está en movimiento (cuando la velocidad de las juntas es mayor a cero). Este enfoque evita la saturación de la base de datos con información redundante y mejora la eficiencia del sistema.

4. REGISTRO DE DATOS EN INFLUXDB

En cuanto al registro de datos en InfluxDB, se emplea una función que se ejecuta cada vez que se reciben datos nuevos del topic **/joint_states**. Esta función tiene la responsabilidad de capturar los datos de posiciones, velocidades y torques de las juntas, y verificar si el robot se está moviendo.

Solo se registran datos cuando el robot está en movimiento, y para saber esto, se comprueba si alguna de las velocidades de las juntas es mayor que cero. Gracias a esto, evitamos que se almacenen datos innecesarios, lo cual ocuparía mucho espacio en la base de datos.

La clase InfluxDBHandler se encarga de gestionar la conexión a InfluxDB y de escribir los puntos. Cada vez que se cumplen las condiciones de movimiento, la función write_data() registra en InfluxDB los datos de posición, velocidad y torque.

5. TRAYECTORIAS

Se ejecutaron un total de 3 trayectorias, cada una de las cuales ha sido registrada y visualizada en Grafana. Las trayectorias ejecutadas muestran movimientos no lineales de las juntas del robot, y se puede observar que las velocidades varían a lo largo de las trayectorias. Las trayectorias han sido registradas correctamente en la base de datos y posteriormente visualizadas en Grafana. Los gráficos muestran velocidades que permiten identificar cambios en los movimientos del robot sin que se hayan producido colisiones o fallos.

Las tres imágenes proporcionadas corresponden a tres trayectorias distintas, representando la velocidad de cada motor del robot mientras se mueve hacia distintos puntos. Cada una de las trayectorias registradas muestra variaciones en la velocidad, confirmando que el robot ha seguido trayectorias no rectas y que ha habido transiciones suaves entre los diferentes puntos de control. Estas trayectorias son solo ejemplos para visualizar los comportamientos de los motores únicamente en cuanto a la velocidad y no se corresponden con las trayectorias lineales solicitadas por el enunciado, aunque se proveerán en la siguiente entrega.

No ha sido posible probar trayectorias con colisión ya que es necesario utilizar el robot del aula.

