



# Tecnológico de Monterrey

## **Reto Semanal 1**

Emiliano Olguin Ortega A01737561

Elias Guerra Pensado A01737354

Yestli Darinka Santos Sánchez A01736992

02 de Abril del 2025

Implementación de robótica inteligente (Gpo 501)

## Resumen

El Mini Challenge 1 tuvo como objetivo la configuración y operación de un PuzzleBot utilizando una Jetson Nano como unidad de procesamiento. El primer paso fue establecer la conexión con el sistema a través de SSH, utilizando la red Wi-Fi generada por la propia Jetson Nano. Una vez establecida la comunicación, se ejecutó el agente micro-ROS y se verificó la conexión mediante el comando ``ros2 topic list``, asegurando que los tópicos esenciales, como ``/cmd_vel``, ``/VelEncL`` y ``/VelEncR``, estuvieran disponibles.

Después de confirmar la correcta comunicación del sistema, se procedió a publicar mensajes en el tópico ``/cmd_vel`` utilizando el formato Twist Message, lo que permitió enviar comandos de movimiento al robot. De manera paralela, se monitorearon los tópicos ``/VelEncL`` y ``/VelEncR`` para registrar las respuestas del sistema y validar que los comandos se reflejaran en la movilidad del PuzzleBot.

Finalmente, se instaló el paquete ``teleop_twist_keyboard`` en un dispositivo externo para habilitar la teleoperación del robot. Con esta herramienta, se controlaron sus desplazamientos de manera remota, ajustando velocidades y direcciones. Durante esta fase, se realizaron pruebas para determinar los valores máximos y mínimos de velocidad y aceleración que el PuzzleBot podía manejar, tanto en términos de movimiento lineal como angular. Estos ensayos fueron clave para evaluar el rendimiento del sistema y su capacidad de respuesta ante distintos parámetros de control.

## Objetivo

Los objetivos del Mini Challenge 1 estuvieron enfocados en la configuración, conexión y control del PuzzleBot utilizando una Jetson Nano y el entorno de trabajo basado en ROS 2. A través de este reto, se buscó desarrollar habilidades clave en la manipulación de robots móviles, la comunicación remota y el uso de micro-ROS para la interacción con sensores y actuadores.

El primer objetivo fue configurar la Jetson Nano y establecer la comunicación con el PuzzleBot a través de una conexión SSH. Para ello, se utilizó la red Wi-Fi generada por la Jetson Nano,

permitiendo el acceso remoto al sistema. Esto garantizó que el entorno de trabajo estuviera correctamente enlazado y listo para la ejecución de comandos.

El segundo objetivo consistió en ejecutar el agente micro-ROS y verificar la comunicación entre el PuzzleBot y ROS 2. Mediante el comando `ros2 topic list`, se confirmó la detección de tópicos esenciales como `/cmd_vel`, `/VelEncL` y `/VelEncR`, los cuales son fundamentales para el control y monitoreo del robot. La correcta detección de estos tópicos aseguró que la configuración inicial fuera exitosa y que el sistema estuviera preparado para recibir y enviar información.

El tercer objetivo fue publicar mensajes en los tópicos de ROS 2 y analizar la respuesta del sistema. Utilizando mensajes tipo Twist en el tópico `/cmd_vel`, se enviaron comandos de movimiento al PuzzleBot, los cuales fueron procesados por el sistema de control del robot. Paralelamente, se monitorearon los tópicos `/VelEncL` y `/VelEncR`, donde los encoders del robot reportaban la velocidad de cada rueda, permitiendo validar si los comandos de movimiento eran correctamente ejecutados.

El cuarto objetivo se centró en habilitar la teleoperación del PuzzleBot utilizando el paquete `teleop_twist_keyboard`, lo que permitió controlar el robot de manera remota desde un dispositivo externo. Con este método, se logró un control dinámico del desplazamiento del robot mediante comandos de teclado, facilitando la exploración de sus capacidades de navegación en tiempo real.

Finalmente, el quinto objetivo fue realizar pruebas de velocidad y aceleración para analizar el rendimiento del PuzzleBot. Se verificaron los valores máximos y mínimos de velocidad lineal y angular, identificando las limitaciones operativas del sistema. Este análisis permitió comprender mejor el comportamiento del robot ante distintos parámetros de control y evaluar su estabilidad y capacidad de respuesta en diversas condiciones de operación.

En conjunto, estos objetivos permitieron una integración completa del PuzzleBot con la Jetson Nano y ROS 2, brindando un entorno de trabajo funcional para la teleoperación y el análisis de su desempeño en términos de movilidad y comunicación.

## **Introducción**

En el ámbito de la robótica móvil, el uso de plataformas accesibles y flexibles permite desarrollar y probar algoritmos de navegación y control. En este contexto, el PuzzleBot es un robot diferencial diseñado para la enseñanza y experimentación en robótica, el cual se basa en el sistema operativo para robots ROS 2 y emplea una Jetson Nano como unidad de procesamiento. Su estructura está compuesta por un chasis con dos ruedas motrices, sensores de encoders que permiten medir su velocidad y posición, y una unidad de control que procesa los comandos enviados por un operador externo.

Para interactuar con el PuzzleBot de manera remota, es necesario establecer una conexión utilizando SSH (Secure Shell). Este protocolo permite acceder y controlar el sistema operativo del robot desde una computadora externa a través de una red. En el caso de este reto, la Jetson Nano genera una red Wi-Fi a la que se conecta el usuario para establecer la comunicación. Una vez conectado, se pueden ejecutar comandos de ROS 2, verificar la correcta detección de los tópicos y enviar instrucciones para manipular el robot.

Uno de los principales objetivos de este reto fue la implementación del paquete Teleop Twist Keyboard, una herramienta que permite controlar el movimiento del PuzzleBot mediante comandos de teclado en un dispositivo remoto. Este paquete funciona publicando mensajes tipo Twist en el tópico `/cmd_vel`, los cuales contienen valores de velocidad lineal y angular. El sistema del PuzzleBot interpreta estos mensajes y ajusta la velocidad de sus motores en consecuencia, permitiendo su desplazamiento en distintas direcciones.

A lo largo del presente reporte, se documentará el proceso de configuración, comunicación y teleoperación del PuzzleBot, detallando los pasos seguidos para garantizar una conexión exitosa, la verificación de la respuesta del sistema y el análisis de su rendimiento en términos de velocidad y control. Este desafío permitió afianzar conocimientos en ROS 2, la integración de hardware y software, y el uso de herramientas de teleoperación para robots móviles.

## **Solución del problema**

Lo primero que se realizó fue la configuración de la *HackerBoard*, en este caso una Jetson Nano, para establecer una red Wi-Fi que permitiera la comunicación con el PuzzleBot.

Luego, desde una computadora externa, se estableció conexión remota a través de SSH con el siguiente comando:

```
ssh puzzlebot@10.42.0.1
```

Con esto, se obtuvo acceso completo al entorno de la Jetson Nano para ejecutar comandos y lanzar nodos de ROS 2. Uno de los pasos cruciales fue iniciar el agente de micro-ROS para habilitar la comunicación con el sistema, lo cual se logró ejecutando el siguiente comando:

```
ros2 launch puzzlebot_ros micro_ros_agent.launch.py
```

Este *launch file* permitió iniciar la comunicación entre los microcontroladores del PuzzleBot y el sistema ROS 2 en la Jetson. Una vez iniciado, se utilizó el comando `ros2 topic list` para verificar que los tópicos como `/cmd_vel`, `/VelEncL` y `/VelEncR` estuvieran activos.

Para controlar el robot de manera remota desde el teclado, se utilizó el siguiente comando en la terminal:

```
ros2 run teleop_twist_keyboard teleop_twist_keyboard
```

Este comando permitió enviar mensajes tipo Twist al tópico `/cmd_vel`, habilitando el control del movimiento del PuzzleBot en tiempo real. Usando las teclas del teclado, se ajustaron la velocidad lineal y angular del robot para evaluar su capacidad de respuesta, navegación y estabilidad.

## Resultados

<https://youtu.be/JWsrPV7ufr4>

## Conclusión

El desarrollo del Mini Challenge 1 permitió cumplir exitosamente con todos los objetivos planteados, logrando una integración efectiva del PuzzleBot con la Jetson Nano y ROS 2. A lo largo del proceso, se configuró correctamente la comunicación SSH, lo que posibilitó el acceso remoto al sistema y la ejecución de comandos para la interacción con el robot. La verificación de los tópicos en ROS 2 confirmó la conexión adecuada del agente micro-ROS, asegurando que el PuzzleBot respondiera correctamente a las instrucciones enviadas.

Uno de los logros más importantes fue la publicación de mensajes en el tópico `/cmd_vel`, lo que permitió controlar la velocidad y dirección del robot en tiempo real. Asimismo, la

monitorización de los tópicos ``/VelEncL`` y ``/VelEncR`` permitió validar el funcionamiento de los encoders y su respuesta ante los comandos de movimiento. La implementación del paquete ``teleop_twist_keyboard`` se realizó sin inconvenientes, logrando una teleoperación eficiente del PuzzleBot desde un dispositivo externo. Además, las pruebas de velocidad y aceleración demostraron que el sistema operaba dentro de los parámetros esperados, sin fallos en la ejecución de los comandos.

El éxito del reto se debió a la correcta aplicación de la metodología propuesta, la precisión en la configuración de los dispositivos y la verificación constante de la comunicación entre los distintos elementos del sistema. No se presentaron fallas significativas que impidieran el cumplimiento de los objetivos, lo que indica que la metodología utilizada fue adecuada para la tarea.

No obstante, una posible mejora a la metodología implementada podría ser la automatización de ciertos procesos, como la verificación de la conexión de los tópicos en ROS 2 mediante scripts que faciliten la detección de errores. Asimismo, la integración de una interfaz gráfica para el control del robot podría optimizar la experiencia de teleoperación, haciéndola más intuitiva y accesible.

En conclusión, el Mini Challenge 1 proporcionó un aprendizaje valioso sobre la configuración, comunicación y control de un robot móvil mediante ROS 2. La correcta ejecución de cada etapa del proceso permitió comprobar el funcionamiento del PuzzleBot y su capacidad de respuesta ante distintos comandos de movimiento, consolidando así los conocimientos en robótica y sistemas digitales.