

## Laboratorio No. 04 - 2025-II - Robótica de Desarrollo, Intro a ROS 2 Humble - Turtlesim



### 1. Requisitos:

- Tutoriales de Linux Intro Linux.
- Tutoriales de ROS 2 Intro ROS 2 Humble.
- Tutoriales de Turtlesim Intro Turtlesim.

### 2. Resultados de aprendizaje:

- Conocer y explicar los conceptos básicos de ROS (Robot Operating System).
- Usar los comandos fundamentales de Linux.
- Conectar nodos de ROS 2 con Python.

### 3. Ejercicio en el laboratorio:

a) Familiarizarse con los comandos de mayor uso para la consola de Linux.

b) **Implementación en Python:**

#### **Procedimiento:**

- Dentro del **workspace** creado en clase (`my_turtle_controller`), se debe editar el archivo `move_turtle.py`, el objetivo es controlar el movimiento de la tortuga en el simulador `turtlesim` mediante el teclado, cumpliendo con los siguientes requerimientos:

#### 1) **Control de movimiento manual:**

- El script debe permitir mover la tortuga de forma **lineal y angular** utilizando las **flechas del teclado**.
- Acciones asignadas:
  - Flecha ↑: avanzar hacia adelante.
  - Flecha ↓: retroceder.
  - Flecha ←: girar a la izquierda.
  - Flecha →: girar a la derecha.

#### 2) **Dibujo automático de letras personalizadas:**

A partir del nombre completo cada uno de los integrantes del equipo (por ejemplo, *Manuel Felipe Carranza Montenegro*), se deben implementar funciones para que la tortuga dibuje las letras correspondientes a las iniciales de sus nombres y apellidos. En este caso específico, se deben incluir las siguientes teclas:

- Tecla **M**: dibujar la letra **M** (por *Manuel* y *Montenegro*).
- Tecla **F**: dibujar la letra **F** (por *Felipe*).
- Tecla **C**: dibujar la letra **C** (por *Carranza*).

3) **Restricción importante:**

- El movimiento de la tortuga debe ser gestionado **exclusivamente desde el script** `move_turtle.py`.
- **No está permitido utilizar el nodo `turtle_teleop_key` para el control con teclado.**

## Actividades a realizar

- Documentación del desarrollo:** Se debe incluir una descripción detallada del desarrollo del laboratorio en el archivo `README.md` del repositorio de GitHub. Este documento debe explicar claramente los objetivos, procedimientos realizados, decisiones de diseño y funcionamiento general del proyecto.
- Diagrama de flujo:** Se debe elaborar un diagrama de flujo que represente de manera clara el funcionamiento de la solución planteada. Este diagrama debe ser implementado utilizando el lenguaje `Mermaid`, y debe incluirse en el `README.md` para facilitar su visualización directa en GitHub.
- Código fuente:** Todos los scripts y archivos de código utilizados deben ser añadidos como anexos dentro del repositorio, organizados en una estructura de carpetas clara y comprensible. Es importante incluir comentarios relevantes en el código para facilitar su comprensión.
- Video explicativo:** Se debe grabar un video con una duración máxima de 10 minutos, en el que se analice, desarrolle e implemente la solución planteada. El video debe cumplir con la siguiente estructura:
  - **Introducción:** el video debe comenzar con la introducción oficial del laboratorio LabSIR Intro LabSIR.
  - **Presentación del equipo:** incluir una diapositiva donde se presenten los integrantes del grupo, sus nombres y aportes dentro del laboratorio.
  - **Análisis, desarrollo e implementación:** mostrar el proceso seguido para resolver la actividad, incluyendo explicaciones del código, decisiones técnicas tomadas y evidencia del funcionamiento correcto de la solución.
  - **Conclusiones:** incluir una sección final donde se expongan las conclusiones generales del laboratorio, reflexiones del equipo y aprendizajes obtenidos.

## Observaciones:

- Se debe crear un repositorio en GitHub donde se documente la metodología, los resultados, análisis y conclusiones.
- Forma de trabajo:** Grupos de laboratorio (2 personas).
- Los puntos que requieran implementación deben tener funciones bien comentadas, y se deben adjuntar los archivos `.py` correspondientes.

## Entrega:



- a) **Forma de trabajo:** En parejas. Cada integrante deberá incluir la URL del repositorio.
- b) **Entregables:** Repositorio en GitHub con lo solicitado en las actividades y ejercicios.
- c) **Fecha de entrega:** Según la programación en Moodle.

## Referencias

- [1] Labsir UN - Intro Linux [Repositorio en GitHub]. Disponible en: [https://github.com/labsir-un/ROB\\_Intro\\_Linux.git](https://github.com/labsir-un/ROB_Intro_Linux.git)
- [2] Labsir UN - Intro ROS 2 Humble [Repositorio en GitHub]. Disponible en: [https://github.com/labsir-un/ROB\\_Intro\\_ROS2\\_Humble.git](https://github.com/labsir-un/ROB_Intro_ROS2_Humble.git)
- [3] Labsir UN - Intro Turtlesim [Repositorio en GitHub]. Disponible en: [https://github.com/labsir-un/ROB\\_Intro\\_ROS2\\_Humble\\_Turtlesim.git](https://github.com/labsir-un/ROB_Intro_ROS2_Humble_Turtlesim.git)
- [4] ROS 2 Humble Installation Guide. Disponible en: <https://docs.ros.org/en/humble/Installation.html>
- [5] Introducing Turtlesim - ROS 2 Humble Tutorial. Disponible en: <https://docs.ros.org/en/humble/Tutorials/Beginner-CLI-Tools/Introducing-Turtlesim/Introducing-Turtlesim.html>