

# Trabajo Práctico Final

## Parte B

### I-402 - Principios de la Robótica Autónoma

---

Prof. Ignacio Mas, Tadeo Casiraghi y Bautista Chasco

27 de octubre de 2025

**Fecha límite de entrega:** 05/12/2025, 23.59hs.

**Modo de entrega:** Enviar por el Aula Virtual del Campus **en un solo archivo comprimido** el paquete de ROS con código comentado y el informe pdf.

En este trabajo práctico final los alumnos deberán integrar los principales conceptos abordados a lo largo de la materia de Principios de la Robótica Probabilística mediante la implementación de un sistema autónomo de localización y navegación. Utilizando un robot TurtleBot3 simulado en el entorno de Gazebo, el objetivo será que el robot explore un entorno desconocido tipo laberinto, construya un mapa del mismo y posteriormente sea capaz de navegar en él de forma autónoma.

Este proyecto representa un caso de aplicación realista donde convergen múltiples herramientas vistas durante la cursada, como la estimación de estado mediante SLAM, el uso de sensores ruidosos (como el LIDAR), la planificación de trayectorias y el seguimiento de caminos. Al finalizar, el robot deberá ser capaz de ubicarse en el mapa previamente generado y desplazarse desde cualquier punto de partida hacia una meta determinada, aprovechando exclusivamente la información obtenida durante la etapa de mapeo.

A lo largo del desarrollo se hará hincapié en la correcta integración de módulos, la toma de decisiones bajo incertidumbre y la evaluación crítica del comportamiento del sistema frente a distintas condiciones del entorno simulado.

# 1. Navegación autónoma con localización y seguimiento de trayectoria

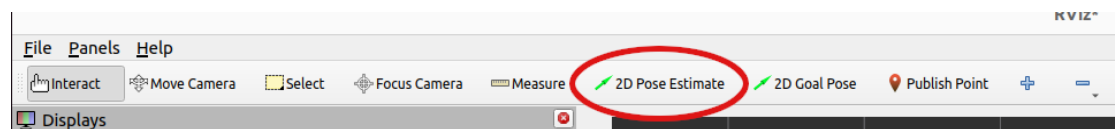
En esta sección deberán implementar la capacidad del robot para desplazarse de manera autónoma entre dos puntos arbitrarios del mapa generado en la etapa anterior. Para ello, el robot debe ser capaz de localizarse dentro del mapa estático conocido, planificar un camino factible desde su posición actual hasta el objetivo deseado, y ejecutar un controlador que permita seguir la trayectoria calculada.

Este desafío integra varios conceptos vistos durante la cursada, tales como localización probabilística, planificación de caminos, y control de seguimiento de trayectorias. La correcta combinación de estos módulos es fundamental para lograr una navegación robusta y eficiente en entornos reales o simulados.

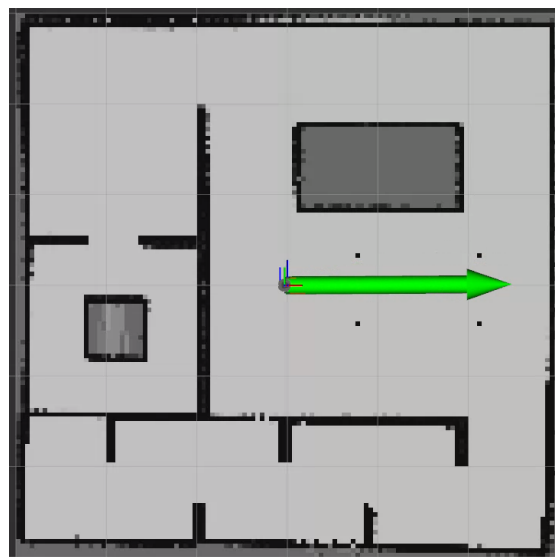
Para abordar este problema, la sección se divide en las siguientes subsecciones:

## 1.1. Localización Inicial en el mapa

Al comenzar la simulación, el robot no sabrá donde está ubicado en el mapa. En la interfaz de RVIZ2 está la herramienta de "2D Pose Estimate". Esta permite fijar una posición estimada de inicio para el robot.



Al hacer click en ella y luego en la posición esperada en el mapa podremos fijar una estimación de la posición del robot.

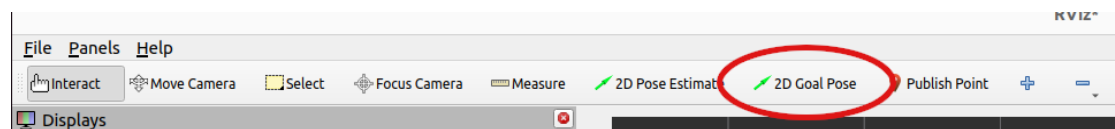


Al hacer esto se publicará automáticamente al tópico "*initialpose*" de tipo "*Pose*-

*WithCovarianceStamped*”[geometry msgs]. Deberán tomar esta primera aproximación y localizarse alrededor de esta.

## 1.2. Objetivo en el mapa

De manera similar, tendrán que luego definir la posición final deseada. En la interfaz de RVIZ2 está la herramienta de ”2D Goal Pose”. Esta permite fijar una posición final para el robot.



Al utilizarla se publicará automáticamente al tópico ”goal\_pose” de tipo ”*PoseStamped*”[geometry msgs]. Deberán utilizar esta pose final para planear el recorrido.

## 1.3. Localización en el mapa

Una vez que se fija la posición inicial, cada vez que se la fije nuevamente y mientras se mueve el robot deberá poder localizarse en el mapa utilizando los datos de odometría, el sensor lidar, y el mapa previamente generado. Se debe trabajar con filtros probabilísticos (por ejemplo, filtro de partículas o EKF) para manejar la incertidumbre.

## 1.4. Planificación de ruta

Se deberá planificar el recorrido entre el inicio y el punto objetivo dentro del mapa generado, evitando chocar con paredes u obstáculos ya mapeados. El recorrido deberá ser seguro, sin pasar demasiado cerca de objetos de ser posible.

## 1.5. Seguimiento de trayectoria (Path Following)

Se deberá implementar un algoritmo que permita que el robot siga la trayectoria planteada y ejecutarlo de manera suave y correcta.

## 1.6. Ángulo final

El robot deberá no solo llegar a la posición final, sino también al ángulo final.

## 1.7. Repeticiones

En caso de que el robot haya llegado a la posición final, se deberá poder especificar una nueva posición final. El robot deberá nuevamente planear un recorrido y ejecutarlo. Esto también aplica si durante su recorrido se especifica otra pose objetivo. En ese caso el robot deberá re-planear y ejecutar el nuevo plan.

## 1.8. Obstáculos

En caso que el robot se encuentre con obstáculos en su recorrido que no estaban mapeados previamente, deberá ejecutar una política de evasión de obstáculo para poder completar el objetivo.

## 1.9. Entornos de prueba

El sistema deberá funcionar bien para el entorno estándar y con obstáculos simple:

- `ros2 launch turtlebot3_custom_simulation custom_casa.launch.py`
- `ros2 launch turtlebot3_custom_simulation custom_casa_obs.launch.py`

De manera opcional y a modo de desafío, pueden probar si sus algoritmos pueden recorrer todo el entorno en el mundo con más obstáculos y rutas cerradas:

- `ros2 launch turtlebot3_custom_simulation custom_casa_obs2.launch.py`

## 1.10. Máquina de Estados

Para controlar todo lo que el robot hace en cada caso, deberán programar una máquina de estados. Esta deberá contemplar todos los casos y posibilidades de cambio de estado.

## 1.11. Evaluación

Se evaluará su trabajo en base a:

- Eficiencia de localización inicial del robot
- Velocidad y eficiencia de generación del recorrido del robot
- Manejo de incertidumbre durante el recorrido
- "Seguridad" del recorrido del robot. (Si el robot se choca mientras hace el recorrido nuestros jefes van a estar muy enojados)
- Capacidad de llegar al objetivo
- Capacidad de manejar pequeños obstáculos
- Opcional: Capacidad de manejar obstáculos complejos

## 2. Reglamento del trabajo

- El trabajo práctico se realiza en **grupos de hasta dos (2) personas**. No se aceptarán grupos de más de dos integrantes, salvo casos excepcionales justificados y aprobados previamente por la cátedra.
- La **entrega final** deberá incluir:
  - Un **paquete de ROS 2 autocontenido**, con todos los nodos necesarios para ejecutar SLAM, localización, planificación y control dentro del entorno de simulación provisto.
  - Un **informe técnico** que describa brevemente la arquitectura general del sistema, las decisiones de diseño tomadas, y los resultados obtenidos. También deberá tener un esquema de la máquina de estados que utilizan. El objetivo es que el informe permita a un tercero entender e interpretar el funcionamiento del sistema.
- Se realizará una **presentación oral de 20 minutos por grupo**, donde deberán explicar su solución, mostrar resultados y responder preguntas de la cátedra. Se evaluará tanto la claridad como la comprensión del trabajo realizado.
- **(Opcional - extra)**: aquellos grupos que hayan alcanzado una solución estable y funcional en simulación, tendrán la posibilidad de **probar su algoritmo en el robot real** disponible en el laboratorio.