

PROYECTO FINAL



GRUPO 1



JUAN ALARCÓN
BRUNO SCHILLING
DIEGO VILLENA



ÍNDICE

- ACTIVIDAD 1: PREPARACIÓN DEL SIMULADOR.....4
- ACTIVIDAD 2: MAPPING.....9
- ACTIVIDAD 3: LOCALIZACIÓN (AMCL).....15
- ACTIVIDAD 4: NAVIGATION2.....19
- CONCLUSIONES.....21

ACTIVIDAD 1

Objetivo: Configurar la preparación del simulador por medio de un archivo launch.

```
1 <launch>
2   <node pkg="very_simple_robot_simulator" exec="kobuki_simulator.py" name="kobuki_simulator" output="screen">
3     <param name="initial_x" value="-1.9" />
4     <param name="initial_y" value="-1.5" />
5     <param name="initial_yaw" value="0.0" />
6
7   <node pkg="very_simple_robot_simulator" exec="lidar_simulator" name="lidar_simulator">
8     <param name="effective_hfov" value="181.0" />
9     <param name="view_depth" value="20.0" />
10    </node>
11
12  <node pkg="very_simple_robot_simulator" exec="world_state_gui.py" name="world_state_gui" output="screen">
13    <param name="map_file" value="$(find-pkg-share proyecto_grupo_1_2025)/map/mapa_bodega.yaml" />
14  </node>
15
16  <node pkg="tf2_ros" exec="static_transform_publisher" name="base_link_to_laser"
17    args="--x 0.0 --y 0.0 --z 0.0 --roll 0.0 --pitch 0.0 --yaw 0.0 --frame-id base_link --child-frame-id laser" />
18
19  <node pkg="rviz2" exec="rviz2" name="rviz" output="screen" args="-d /root/workspace/src/proyecto_grupo_1_2025/rviz/config_proyecto.rviz"/>
20 </launch>
```

Pasos generales de la solución:

- 1) Implementación del archivo launch.
- 2) Verificación de la correcta inicialización de los nodos, con especial atención al Rviz2.
- 3) Investigar brevemente los parámetros del archivo .pgm para el cálculo de las dimensiones métricas del mapa.

ACTIVIDAD 1

- ¿Cómo obtener las dimensiones métricas del mapa proporcionado?

a) Obtener la relación entre pixeles y metros a través de la resolución del archivo .yaml

```
1 image: mapa_bodega.pgm
2 resolution: 0.050000
3 origin: [-1.900000, -1.500000, 0.000000]
4 negate: 0
5 occupied_thresh: 0.65
6 free_thresh: 0.196
```

1 [pix] = 0.05 [m]

b) Obtener las dimensiones en pixeles de la imagen a través del archivo .pgm

727 [pix] x 408 [pix]

ACTIVIDAD 1

- ¿Cómo obtener las dimensiones métricas del mapa proporcionado?

c) Calcular las dimensiones en metros mediante regla de 3

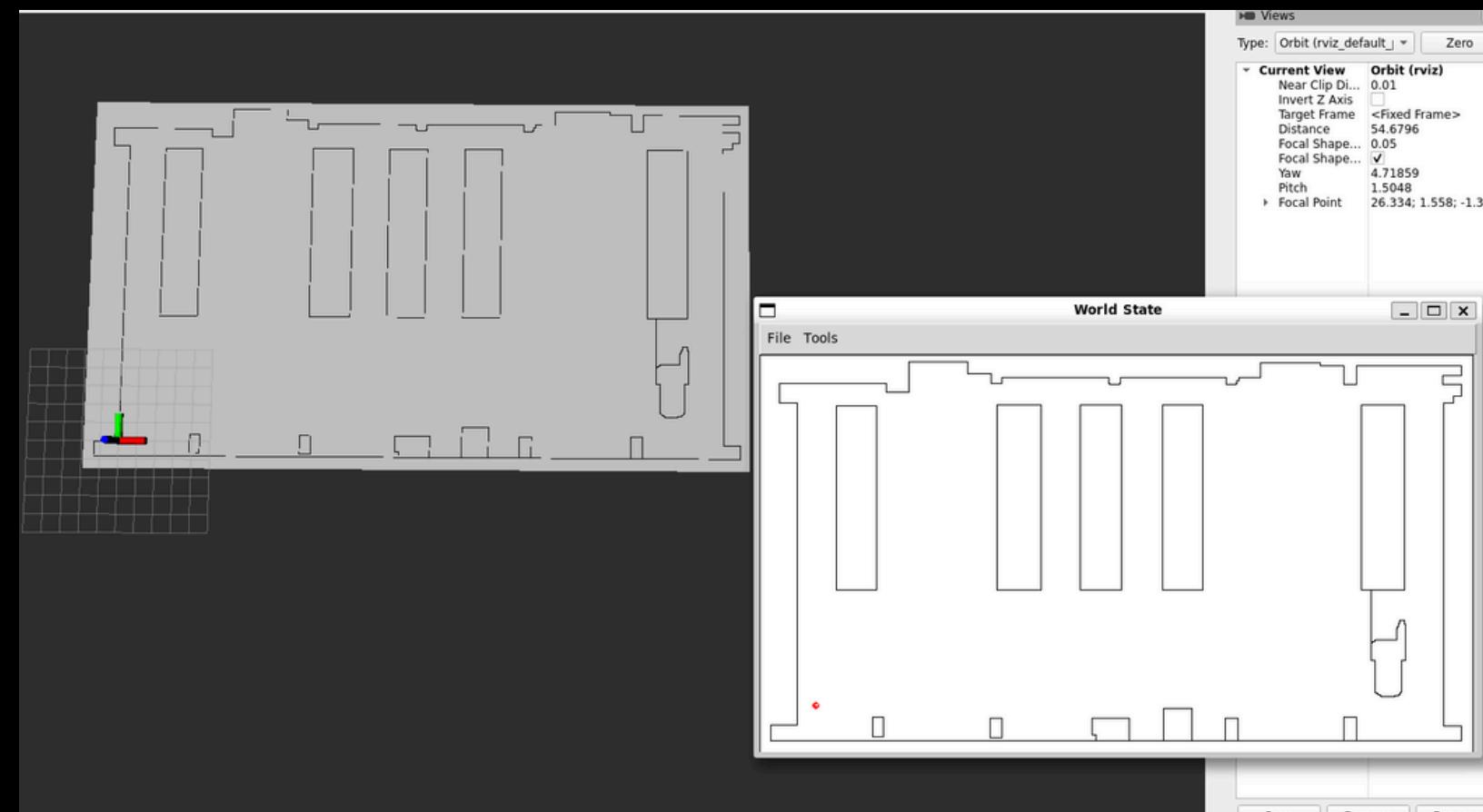
$$\text{ancho} = \frac{727 \text{ [pix]} \times 0.05 \text{ [m]}}{1 \text{ [pix]}} = 36.35 \text{ [m]}$$

$$\text{alto} = \frac{408 \text{ [pix]} \times 0.05 \text{ [m]}}{1 \text{ [pix]}} = 20.4 \text{ [m]}$$

ACTIVIDAD 1

- ¿Qué es el campo ‘origin’ en el archivo .yaml?

- El campo ‘origin’ establece las coordenadas de posición en 2D (en formato x, y, yaw) del pixel inferior izquierdo de la imagen, visto desde el marco de referencia del mapa.
- En este caso, las coordenadas de ‘origin’: [-1.900000, -1.500000, 0.000000] indican que el pixel inferior izquierdo de la imagen se ubica a -1.9 [m] del origen del mapa en el eje horizontal, -1.5 metros del origen en el eje vertical, y sin rotación respecto al marco del mapa.



ACTIVIDAD 1

-¿Qué vector publica tf_ros2 en caso de que el Lidar se posicione justo en el centro del robot?

```
1 <launch>
2   <node pkg="very_simple_robot_simulator" exec="kobuki_simulator.py" name="kobuki_simulator" output="screen">
3     <param name="initial_x" value="-1.9" />
4     <param name="initial_y" value="-1.5" />
5     <param name="initial_yaw" value="0.0" />
6
7   <node pkg="very_simple_robot_simulator" exec="lidar_simulator" name="lidar_simulator">
8     <param name="effective_hfov" value="181.0" />
9     <param name="view_depth" value="20.0" />
10    </node>
11
12   <node pkg="very_simple_robot_simulator" exec="world_state_gui.py" name="world_state_gui" output="screen">
13     <param name="map_file" value="$(find-pkg-share proyecto_grupo_1_2025)/map/mapa_bodega.yaml" />
14   </node>
15
16   <node pkg="tf2_ros" exec="static_transform_publisher" name="base_link_to_laser"
17     args="--x 0.0 --y 0.0 --z 0.0 --roll 0.0 --pitch 0.0 --yaw 0.0 --frame-id base_link --child-frame-id laser" />
18
19   <node pkg="rviz2" exec="rviz2" name="rviz" output="screen" args="-d /root/workspace/src/proyecto_grupo_1_2025/rviz/config_proyecto.rviz"/>
20 </launch>
```

- En el caso de que la posición del Lidar calce con la posición del centro centro robot, el vector de transformación que publique tf_ros2 será x=0, y=0, z=0, roll=0, pitch=0 y yaw=0, pitch=0 y yaw=0. Esto porque el base_link es precisamente el marco de referencia visto desde el punto de vista del centro del robot. Esto porque el base_link es precisamente el marco de referencia visto desde el punto de vista del centro del robot.

ACTIVIDAD 1

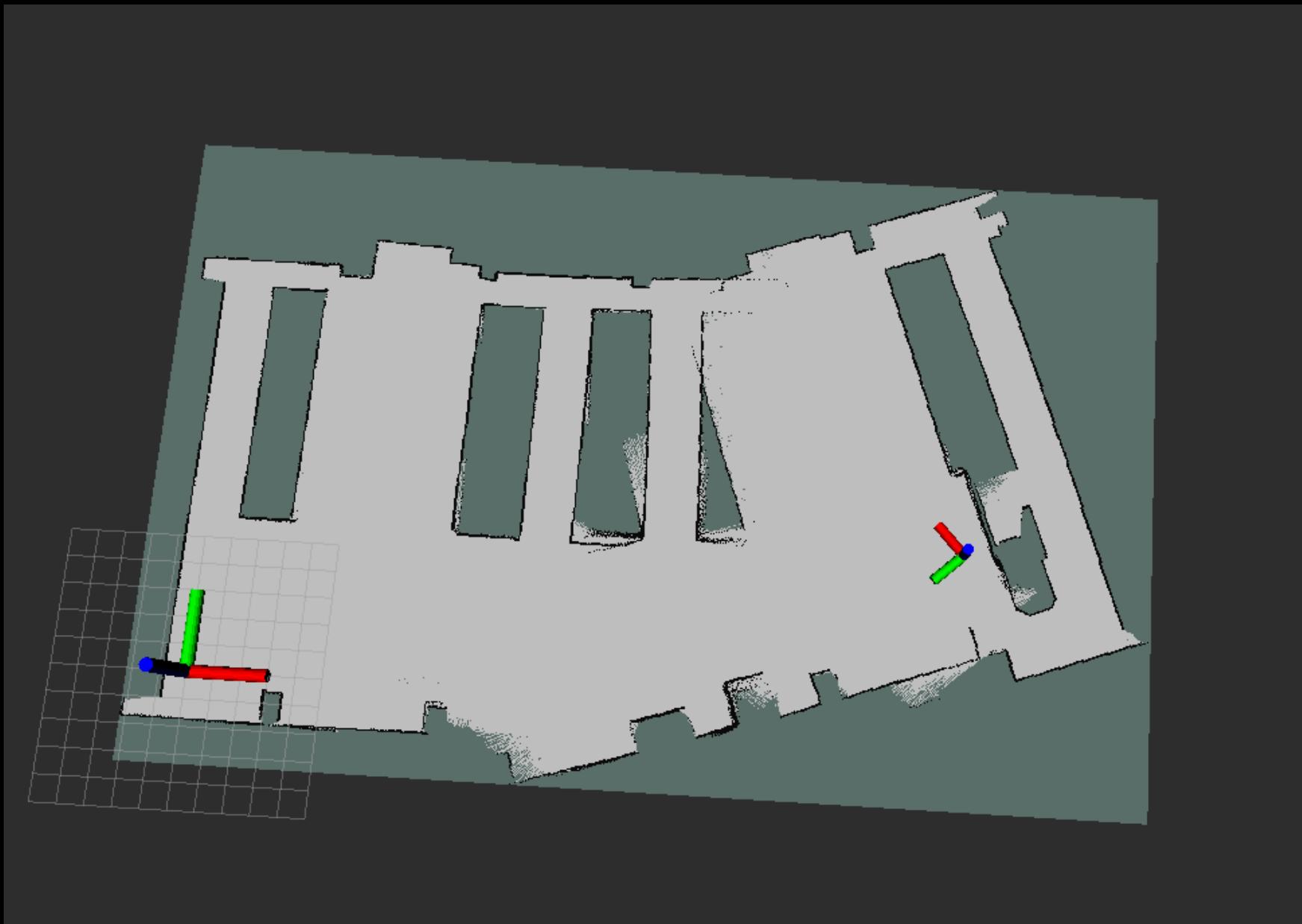
Dificultades de la actividad:

- 1) Comprender e investigar acerca de los parámetros de los archivos .yaml y .pgm para realizar el cálculo de dimensiones métricas.



ACTIVIDAD 2

Objetivo: realizar un mapeo al operar remotamente el robot en conjunto con las herramientas de slamtoolbox



Pasos generales de la solución:

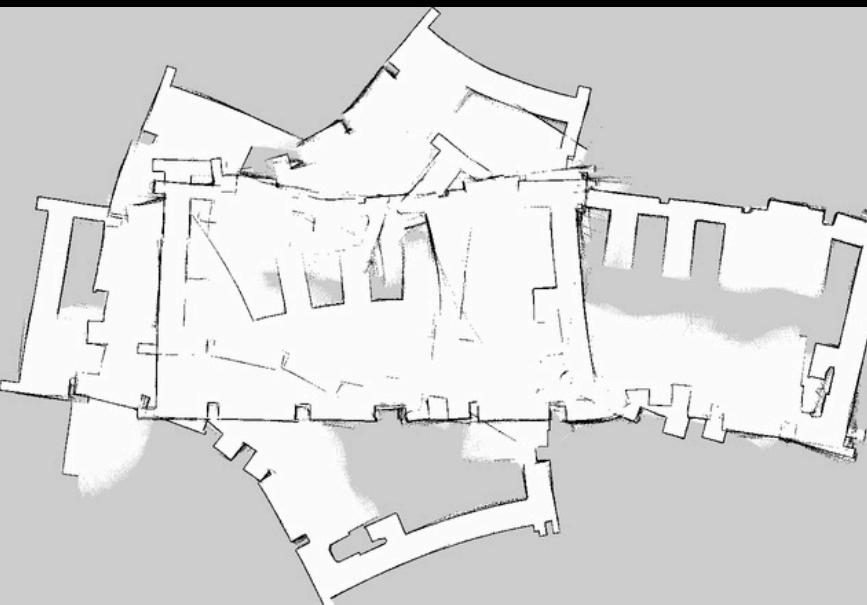
- 1) Inicialización del simulador y la plataforma de rviz.
- 2) Ejecutar slam_toolbox de ros.
- 3) Establecer los marcos de referencia para realizar slam.
- 4) Explorar el mapa mediante la teleoperación.

ACTIVIDAD 2

Resultados

```
root@LAPTOP-5RH7C0U6:~/catkin_ws/src/mt2_Juan_Alarcon# ls
dead_reckoning_act2.py
dead_reckoning_nav.py:Zone.Identifier
goals.txt
goals.txt:Zone.Identifier
log
mi_chatter.py
mi_chatter.py:Zone.Identifier
mt2_Juan_Alarcon
mt2_Juan_Alarcon.tar.gz
my_first_slam_map.pgm
my_first_slam_map.yaml
my_second_slam_map.pgm
my_second_slam_map.yaml
obstacle_detector.py
para_dibujo-angular.txt
para_dibujo-lineal.txt
pid_controller.py:Zone.Identifier
pose_loader.py
pose_loader.py:Zone.Identifier
trayectoria.txt
vel.txt
virtual_robot.py:Zone.Identifier
workspace
```

El mapeo da como resultados los archivos de my_second_slam_map.pgm y my_second_slam_map_yaml uno contiene la imagen generada a partir de la exploracion y el segundo contiene informacion



```
! my_first_slam_map.yaml ×  
! my_first_slam_map.yaml  
1   image: my_first_slam_map.pgm  
2   mode: trinary  
3   resolution: 0.05  
4   origin: [-12.3, -16, 0]  
5   negate: 0  
6   occupied_thresh: 0.65  
7   free_thresh: 0.25
```

ACTIVIDAD 2

- ¿Cómo podría generar el mismo mapa pero con otra resolución?

Para generar el mismo mapa debemos mantener las distancias métricas constantes. Entonces, para lograrlo, la cantidad de pixeles (tanto en ancho como en largo) debe ser inversamente proporcional a la resolución de la imagen. Por ejemplo, si la cantidad de pixeles en largo y en ancho se duplican, entonces la resolución debe reducirse a la mitad.

$$\text{ancho} = \frac{727 \text{ [pix]} \times 0.05 \text{ [m]}}{1 \text{ [pix]}} = 36.35 \text{ [m]}$$

$$\text{alto} = \frac{408 \text{ [pix]} \times 0.05 \text{ [m]}}{1 \text{ [pix]}} = 20.4 \text{ [m]}$$

ACTIVIDAD 2

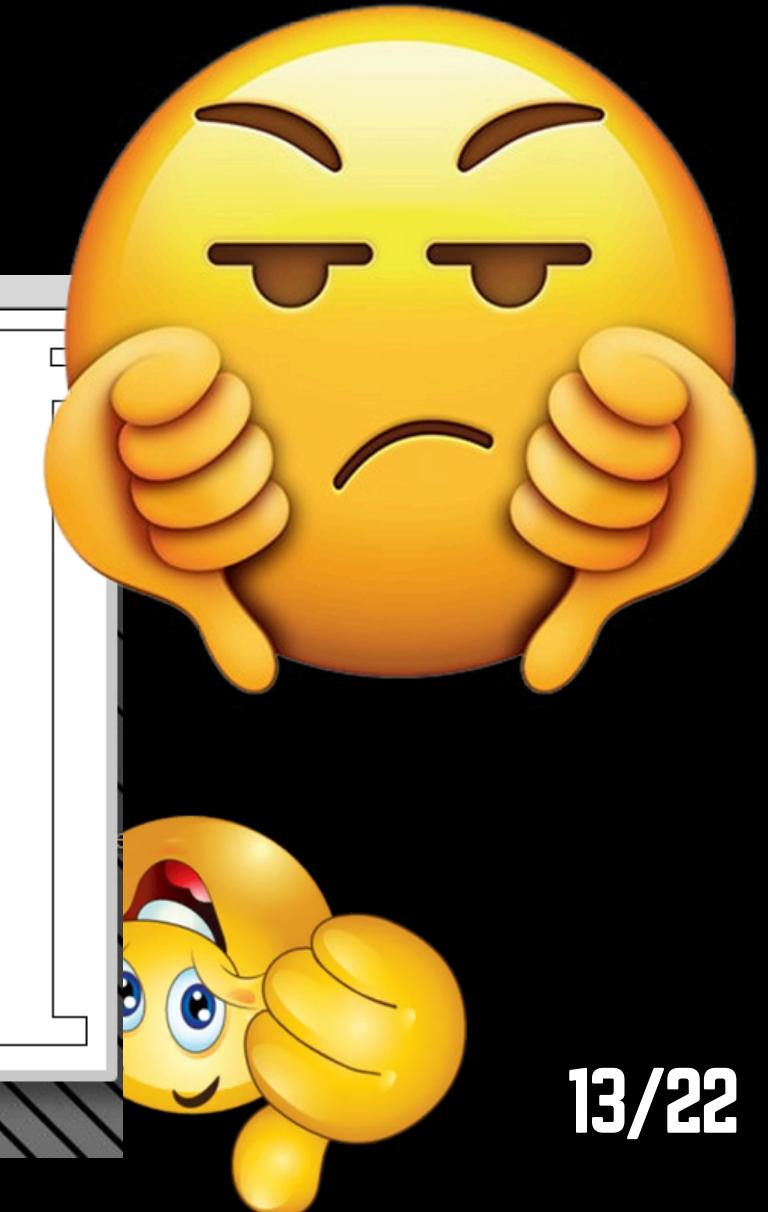
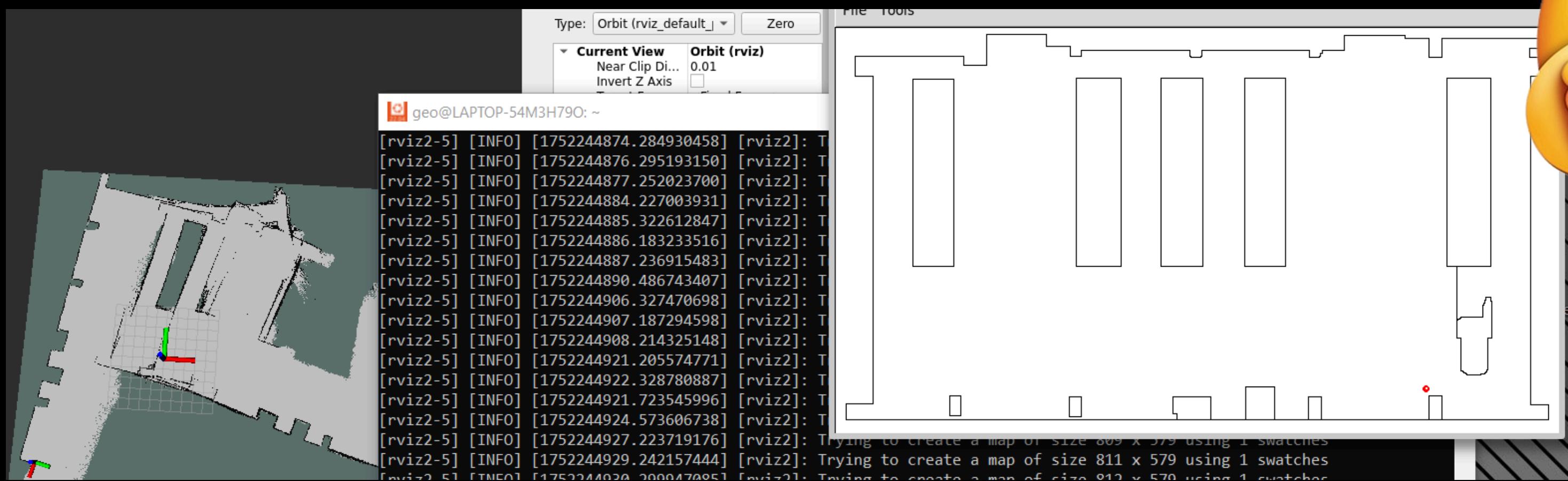
- ¿Por qué el sistema de coordenadas fijo debe establecerse como map y no como world map?

Porque el origen del mapa en el archivo .yaml se encuentra desplazado respecto al origen del simulador y del world map. Por tanto, para hacer una reconstrucción fidedigna debemos referenciarnos a su propio sistema de referencia.

ACTIVIDAD 2

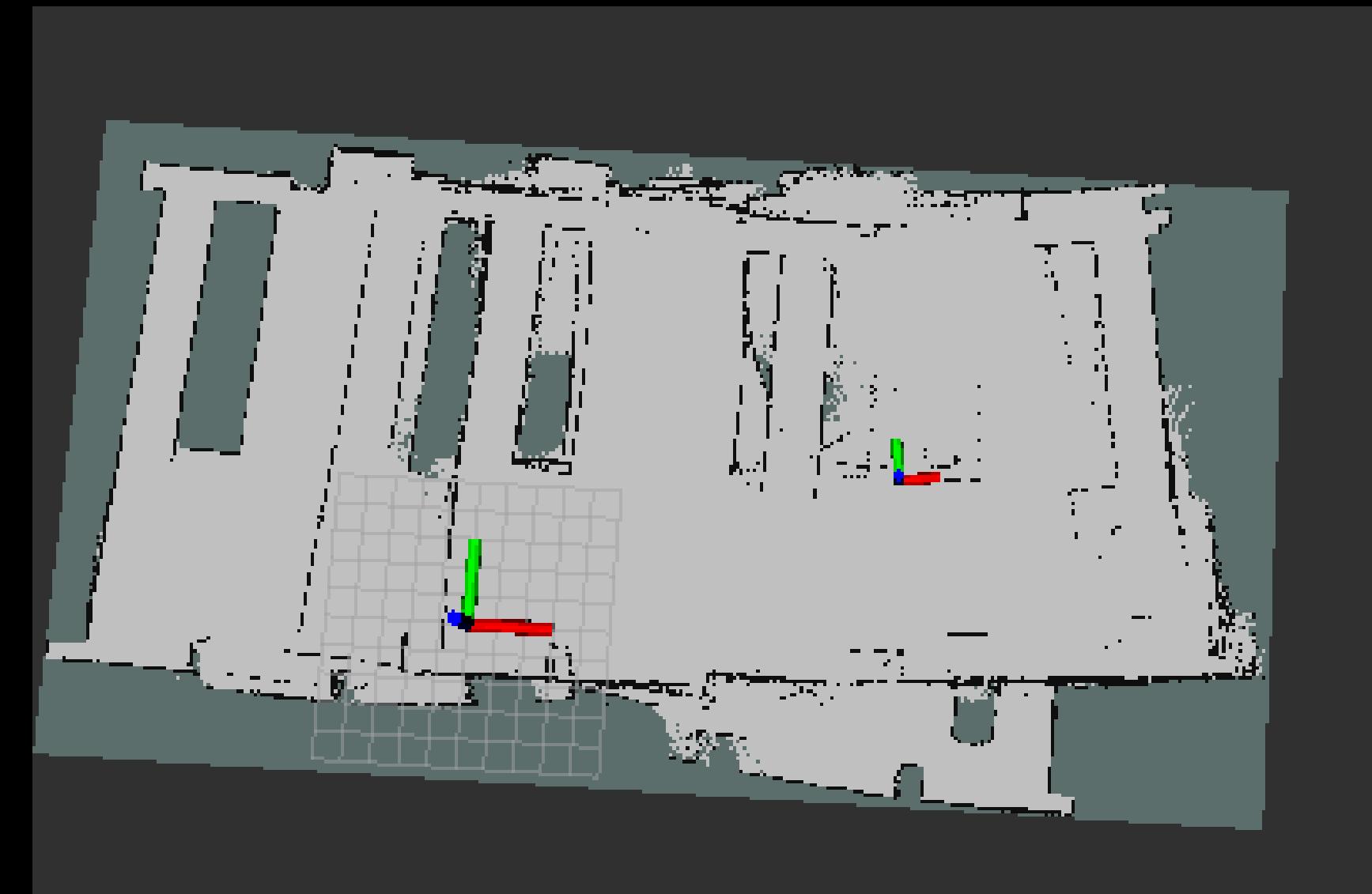
Dificultades de la actividad:

- 1) Instalación de slam-toolbox por archivos de humble que ya han quedado obsoletos.
- 2) Que el robot no se pierda (kidnapped robot)
- 3) Que el robot se ubique en los pasillos (geometrías repetitivas)



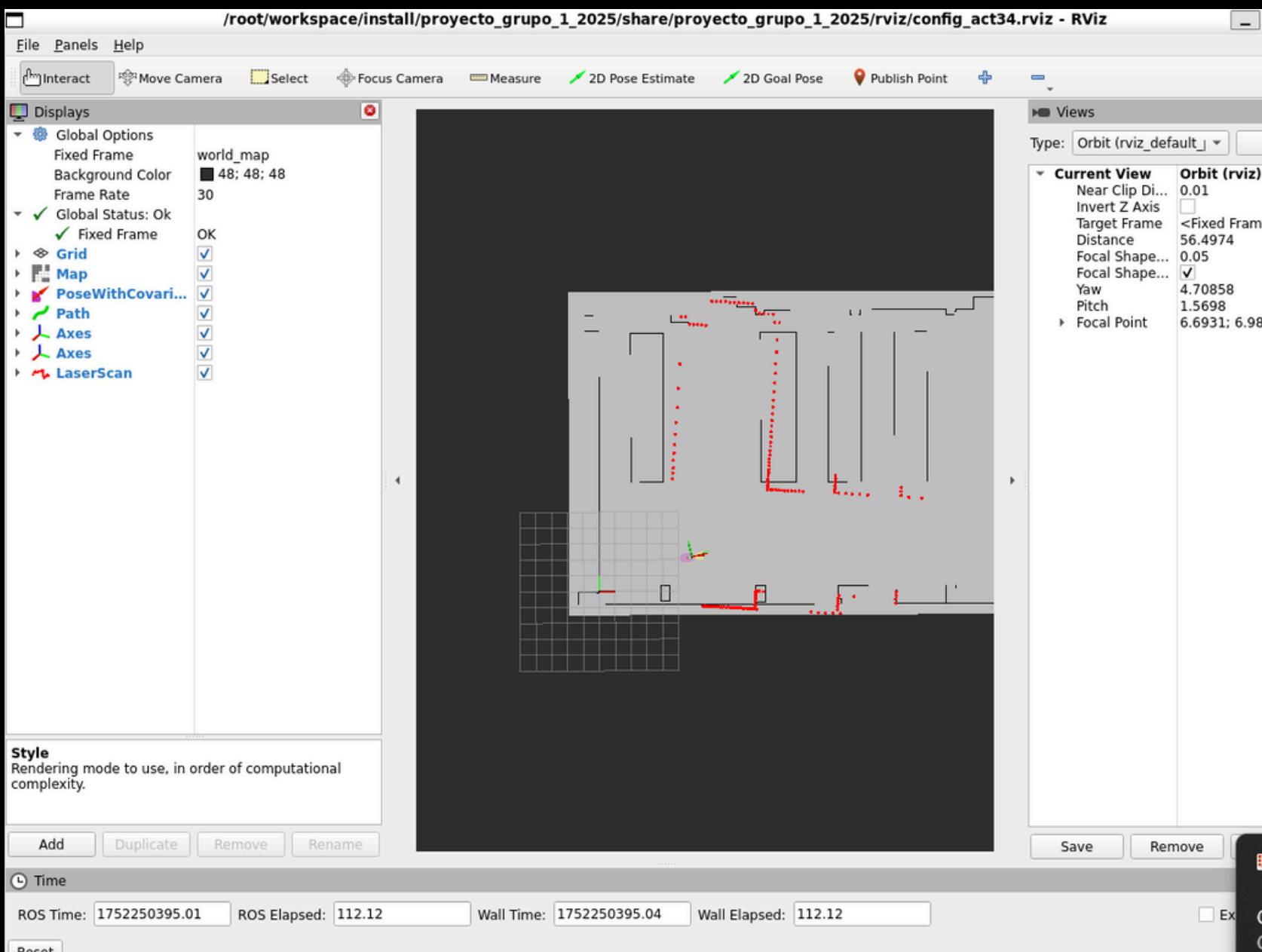
ACTIVIDAD 2

Mejores intentos logrados



ACTIVIDAD 3

Objetivo: Hacer uso del algoritmo Adaptive Monte Carlo Localization.



Pasos generales de la solución:

- 1) Incluir la inicialización de un nuevo nodo de algoritmo AMCL en el launch.
- 2) Activar este nodo a través del nodo lifecycle Bringup.
- 3) Publicar la pose estimada del robot a la hora de iniciar el algoritmo a través de Rviz.

ACTIVIDAD 3

- ¿Cuáles son las principales diferencias entre el AMCL y el MCL?

En relación con el MCL tradicional, la versión adaptativa del algoritmo presenta las siguientes ventajas: es capaz de ajustar de manera adaptativa el tamaño de la muestra durante el proceso de localización, permite una localización estable en entornos con alta simetría, y logra una localización precisa utilizando un tamaño de muestra reducido (*L. Ronghua, H. Bingrong & L. Maohai; Based Adaptive Monte Carlo Localization*).

ACTIVIDAD 3

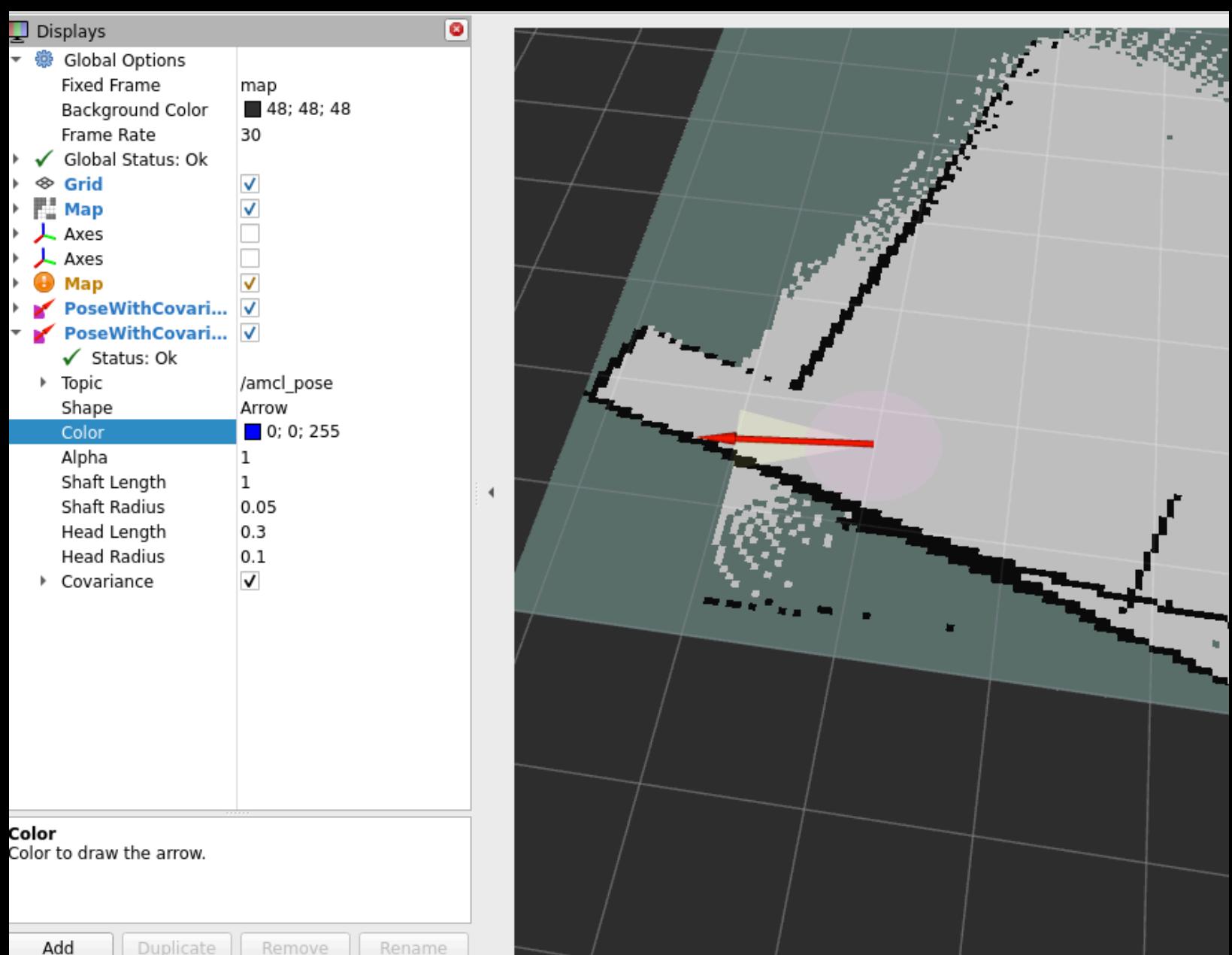
- Parámetros del nodo AMCL:

- Global_frame_id: define el marco de referencia globalizado que utiliza el robot para hacer el mapeo.
- Base_frame_id: especifica el sistema de coordenadas del robot, en este caso desde su centro.
- Odom_frame_id: define el marco de referencia desde donde comienza el sensado del robot con odometría.
- robot_model_type: define el modelo del robot usado por el algoritmo de AMCL, que en este caso es de ruedas diferenciales.
- map_topic: puede activar o desactivar el uso del tópico del mapa. Al estar en true se usa el mapa global disponible para que el robot haga el mapeo.
- laser_max_beams: sirve para determinar el número de haces que puede disparar el robot con el Lidar. En este caso al ser 181, el robot puede sensar hacia el frente de él, con visión de 90° hacia los lados además del ángulo 0.
- min_particles: se define el número mínimo de partículas que utiliza el algoritmo de localización. Es importante considerar que con más partículas se logra una mejor localización pero el computo es mayor.

ACTIVIDAD 3

Consideraciones de la actividad:

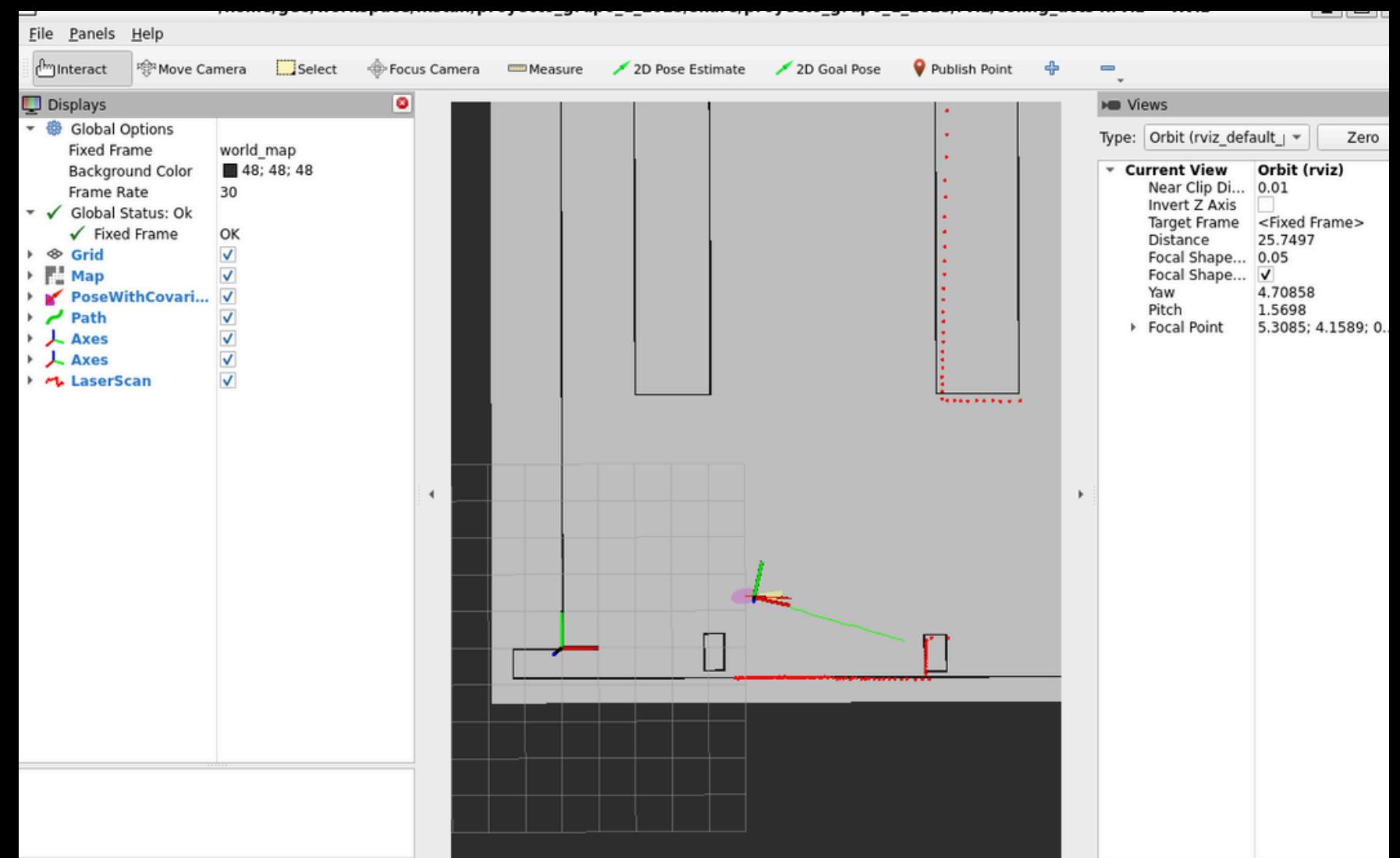
- El círculo burdeos representa la posición estimada del robot en base a las partículas con mayor peso calculadas.
- La pose del robot visualizada en RVIZ no necesariamente debe ser igual a la mostrada en la interfaz del simulador, pues RVIZ puede utilizar otros marcos de referencia distintos de world_map. Además Rviz esta suscrito a los tópicos del simulador, y esto involucra un pequeño delay e inclusive pérdida de información en algunos casos.
- A través del parámetro set_initial_pose, es posible declarar la posición inicial 2D del robot en el nodo de AMCL, indicando x, y, z y yaw.



ACTIVIDAD 4

Objetivos:

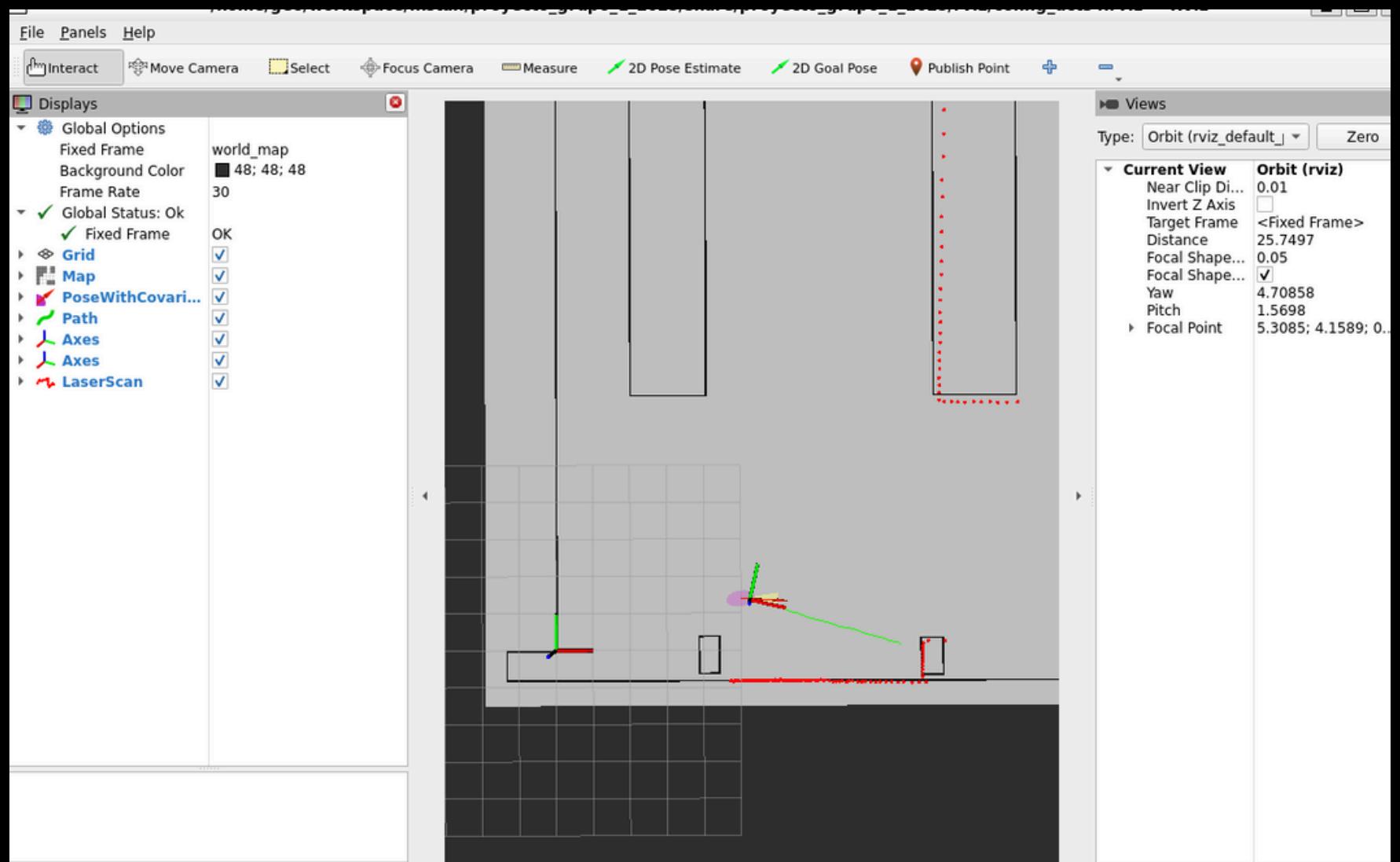
- Que el robot se mueva autónomamente proporcionando únicamente la posición objetivo y una posición inicial.



ACTIVIDAD 4

Consideración de la actividad:

- Se puede visualizar claramente que los haces del Lidar no se ajustan perfectamente al entorno del mapa. Esto puede ocurrir por delay que ocurre al llegar los datos a Rviz al hacer la suscripción del mapa y por pequeño error acumulado de la odometría.



CONCLUSIONES

- Las herramientas utilizadas en este proyecto ahorran una gran cantidad de tiempo tanto en programación del código como en ejecución del programa. Respecto a la experiencia del lab 3 ahorramos más del 50% de tiempo y obtuvimos mejores resultados.
- Con respecto a la actividad 2, hubo demasiadas complicaciones para lograr un resultado medianamente decente del mapeo. A medida que realizábamos los intentos, nos dimos cuenta que las zonas amplias, pasillos indistinguibles con respecto a otros, y entornos muy similares afectaban completamente el proceso de mapeo. También, que a menor velocidad e ir pausando el robot (con avances muy mínimos), se lograba una mayor precisión.
- Realizar un mapeo exitoso requiere de una odometría muy precisa y de un Lidar bueno, en espacial para mapas de gran tamaño.

PROYECTO FINAL

JUAN ALARCÓN
BRUNO SCHILLING
DIEGO VILLENA

GRUPO 1

