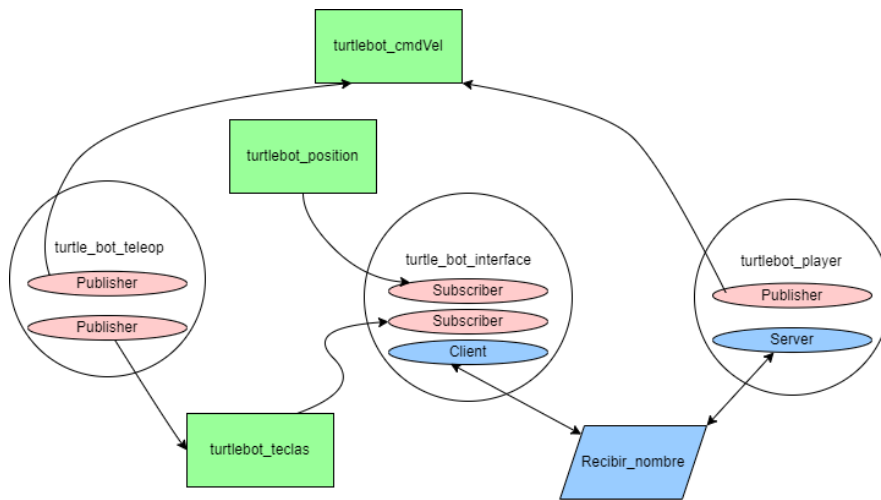
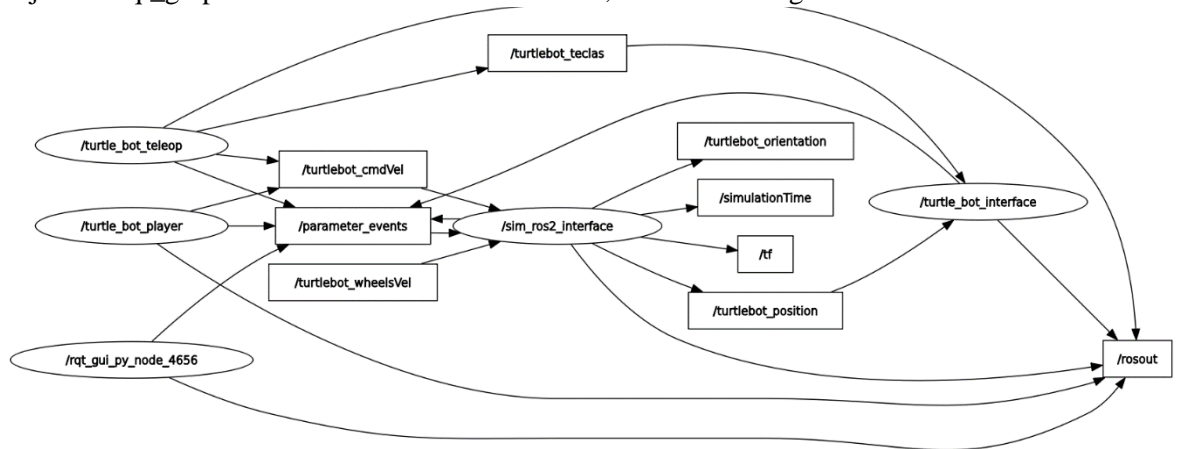


Documentación Taller 1 Robótica

Conexión general entre nodos, tópicos y servicios

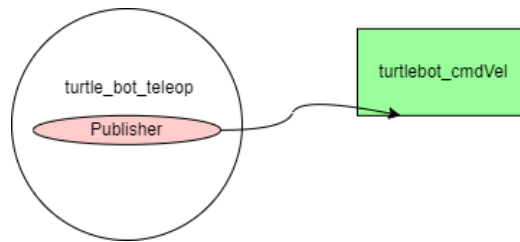


Al ejecutar `rqt_graph` sobre todo el sistema funcional, se obtuvo lo siguiente:

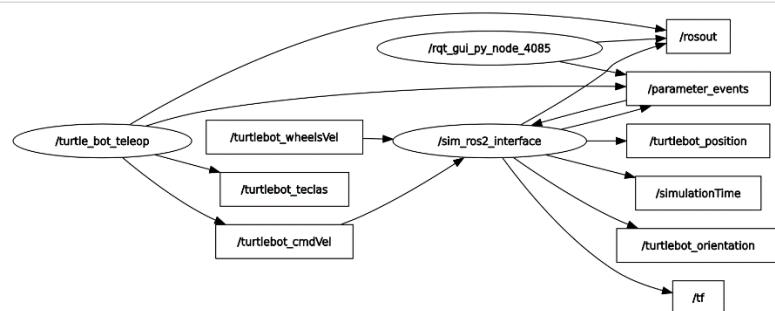


Punto 1

Grafo de conexión entre nodos y tópicos:

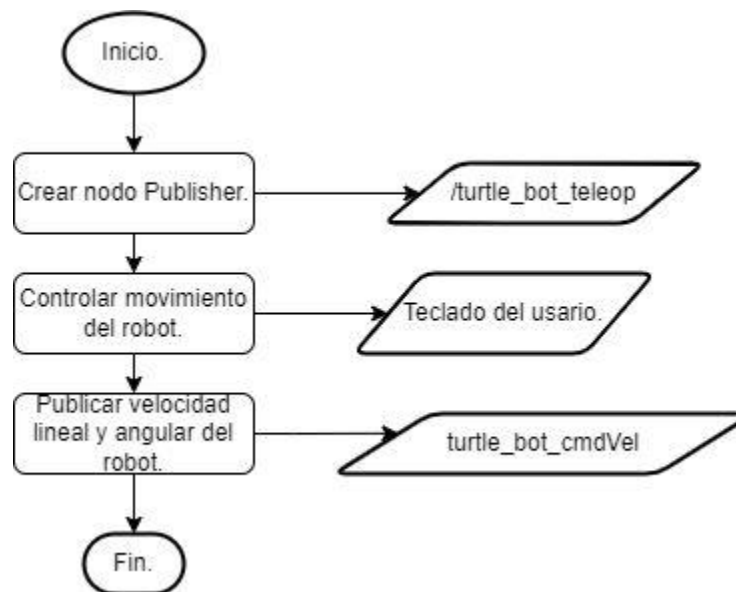


Resultado rqt_graph:



A pesar de que el tópico turtlebot_tecclas no es necesario para este primer punto, este está corriendo ya que el tercer punto necesita de este. Se observa como el nodo turtle_bot_teleop publica tanto a turtlebot_tecclas como a turtlebot_cmdVel.

Diagrama general de funcionamiento:



Para la solución de este primer punto, se creó el nodo turtle_bot_cmdVel en el cual se corre la función Listenkeyboard de python con la cual se leen las teclas pulsadas por el usuario. A partir de estas, se publica al nodo turtlebot_cmdVel, la velocidad lineal en x o angular, tomando en cuenta los inputs de velocidad lineal y angular solicitados al usuario inicialmente.

Cabe resaltar que, una vez se dejan de pulsar teclas, la velocidad tanto lineal como angular pasa a ser 0 nuevamente.

Ejecución del nodo una vez compilado el paquete `turtle_bot_2`:

1. Instalar la librería de python: *pynput*
2. Abrir coppelia en una terminal y abrir el ambiente de simulación
3. Darle play a la simulación
4. Abrir una terminal nueva y correr el comando dentro de la carpeta raíz: `. install/setup.bash`
5. `ros2 run turtle_bot_2 turtle_bot_teleop`
6. En la barra de comandos de Coppelia, escribir las teclas deseadas (esto porque si se pulsan las teclas estando dentro de la terminal, no serán reconocidas)
7. Observar los movimientos del robot

Dependencias usadas:

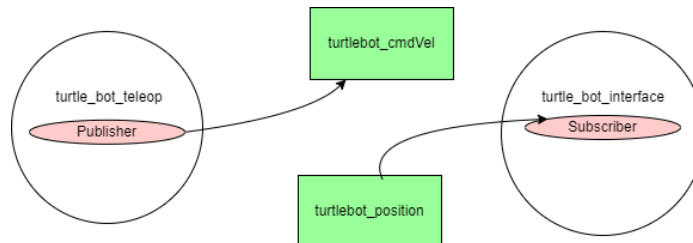
- `geometry_msgs` : tipo de mensaje que usa el tópico `turtlebot_cmdVel`
- `rclpy`

La dependencia se añadió al archivo `package.xml` como

```
<exec_depend>rclpy</exec_depend>
<exec_depend>geometry_msgs</exec_depend>
```

Punto 2:

Grafo de conexión entre nodos y tópicos



Resultado `rqt_graph`:

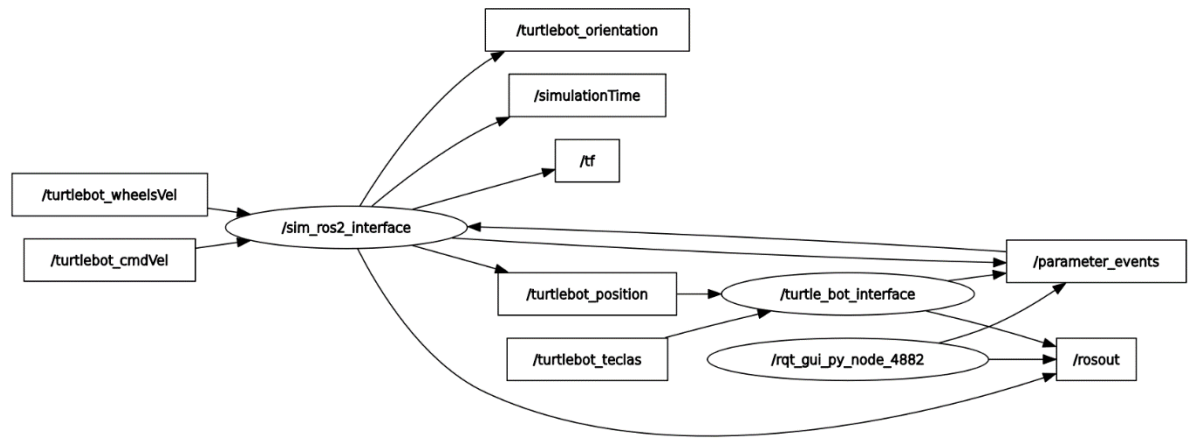
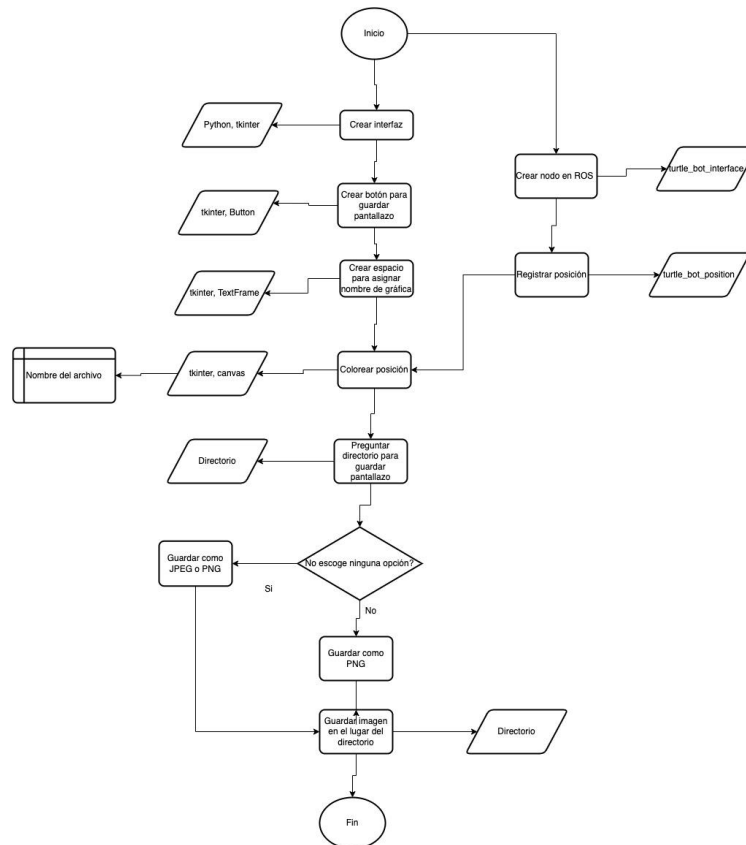


Diagrama general de funcionamiento:



Para este punto, se usaron 2 hilos de ejecución. El primero, contiene la interfaz (hecha con la librería tkinter de python) en la cual se crearon los botones y espacios para introducir el nombre de la imagen si se desea capturar pantalla. En el segundo, se ejecuta el nodo que se suscribe a turtle_bot_position para poder actualizar la posición constantemente y dibujar en la interfaz.

Es posible que al guardar el pantallazo de la interfaz se guarde una imagen en negro, ya que en algunos computadores hay un problema de seguridad con Spyware. Por ello, se recomienda correr los siguientes comandos en la terminal antes de probar por primera vez el nodo:

- `sudo nano /etc/gdm3/custom.conf`
- Modificar la linea por `WaylandEnable=false`
- Guardar los cambios
- `sudo systemctl restart gdm3`

Ejecución del nodo:

1. Los comandos del punto1 se debieron haber ejecutado
2. En una nueva terminal ejecutar: `ros2 run turtle_bot_2 turtle_bot_player` (es necesario ya que o si no el nodo se queda esperando a que el cliente del punto 4 se inicie)
3. En una nueva terminal: `ros2 run turtle_bot_2 turtle_bot_interface`

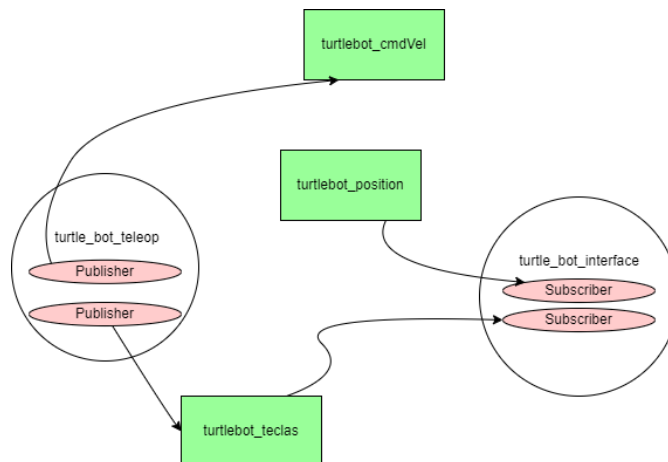
Al ejecutar el nodo se debe abrir la interfaz. Si se pulsan teclas (en un espacio diferente a la terminal), se deben ver los trazos de cada movimiento.

Dependencias usadas:

Se usaron las mismas dependencias del punto 1

Punto 3:

Grafo de conexión entre nodos y tópicos



Resultado `rqt_graph`:

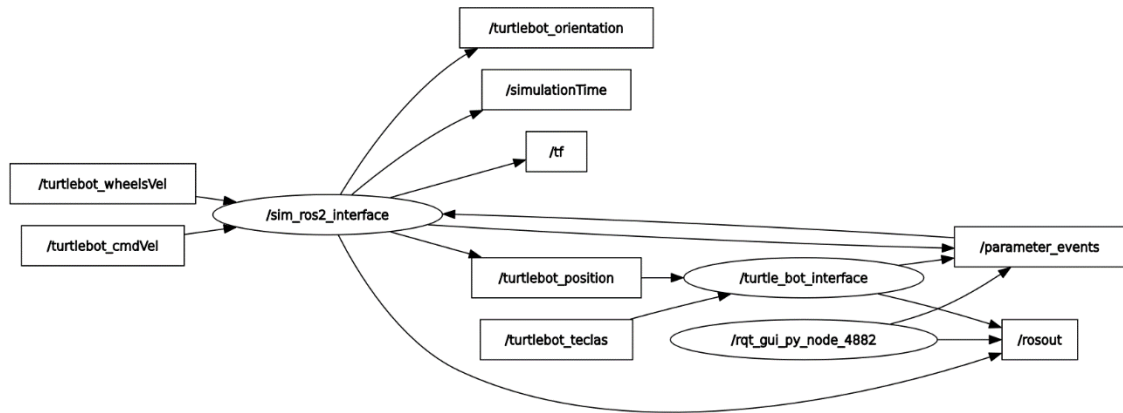
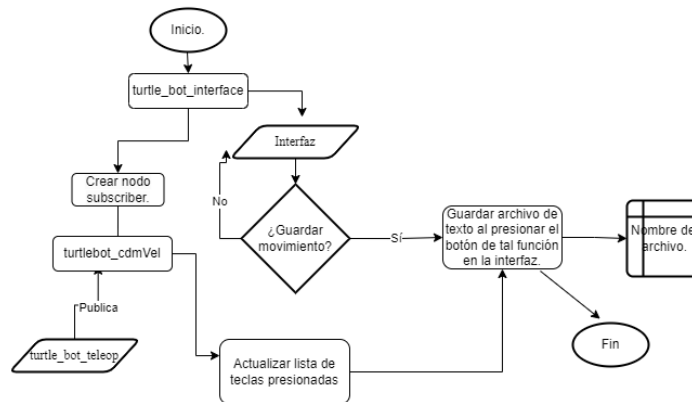


Diagrama general de funcionamiento:



En este punto se adicionó el publisher turtlebot_tecclas dentro del nodo del turtle_bot_teleop de tipo String, que publica las teclas presionadas por el usuario y el nodo turtle_bot_interface se suscribe a este. En interfaz se añaden los botones de guardar movimientos y reproducir movimientos, así como un espacio para ingresar en texto el nombre del archivo que tendrá registradas las pulsaciones de las teclas en base a la información recibida desde el publisher. El botón de guardar movimientos genera el archivo mencionado en formato .txt.

Para guardar los movimientos, se acude a la lista de movimientos que se ha ido actualizando con el topico turtlebot_tecclas al cual el nodo turtlebot_interface se suscribe y es actualizado en el nodo turtlebot_teleop.

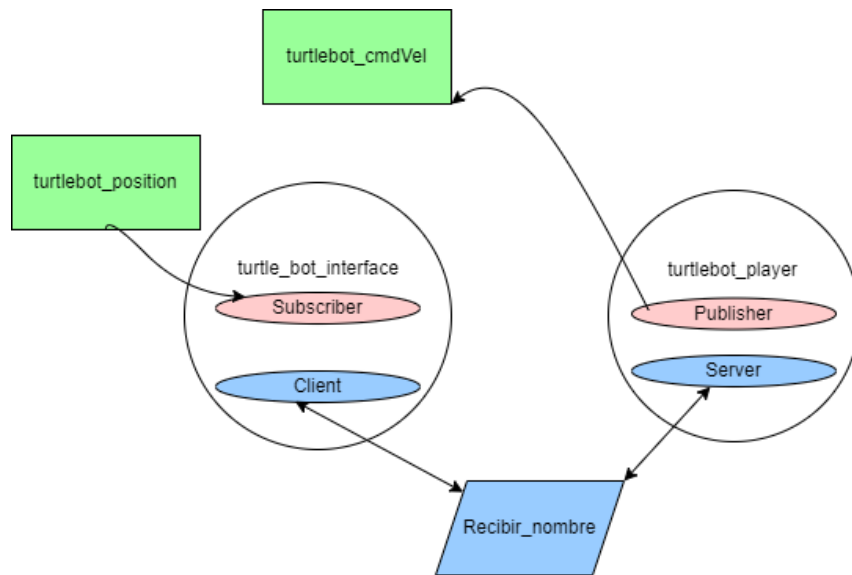
Ejecución del nodo:

Para la ejecución de este punto se deben ejecutar los mismos comandos del punto 2. Al introducir un nombre y seleccionar el botón de guardar movimientos, se debe generar el archivo txt con la lista de teclas pulsadas. Es importante resaltar que el archivo guarda las teclas que han sido pulsadas antes de presionar el botón de guardar.

Punto 4:

Grafo de conexión entre nodos y tópicos:

La conexión de nodos de este punto se puede ver tanto como en la conexión general del sistema mostrada al iniciar el documento o como los componentes esenciales para reproducir el movimiento, es decir:



En este caso, el nodo teleop no es necesario ya que el nodo turtlebot_player también publica al tópico turtlebot_cmdVel.

Resultado rqt_graph (corriendo solo estos 2 nodos):

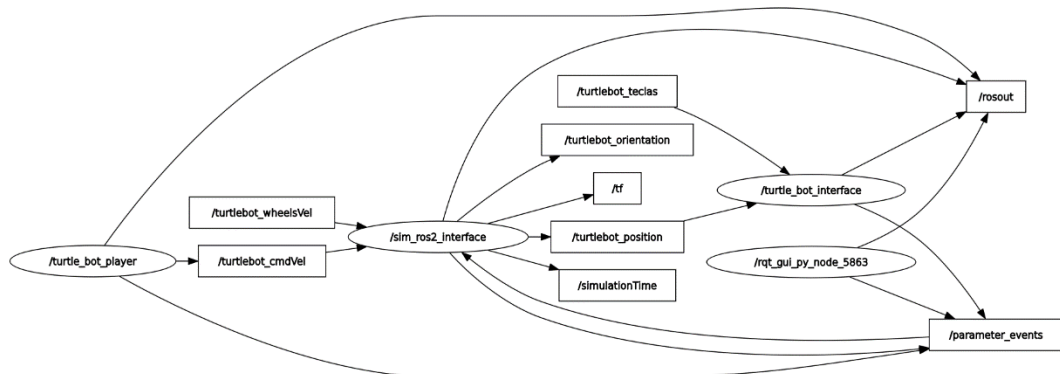
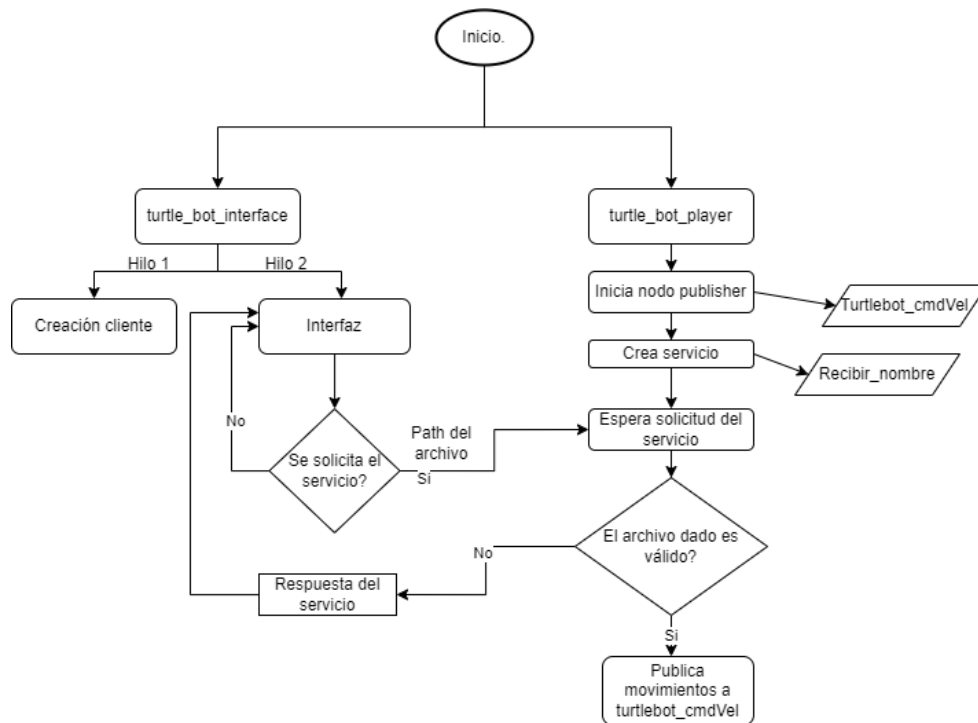


Diagrama general de funcionamiento:



Finalmente, para el punto 4 se creó un servicio llamado `Recibir_nombre`, en el nodo `turtle_bot_player`. El servicio recibe un `String` con la ubicación del archivo que se desea ejecutar y devuelve un booleano indicando si se ejecutó o no la solicitud. Si se solicita el servicio, se lee el archivo en mención y se publica al tópic `turtlebot_cmdVel` para mover el robot.

Así mismo, se añadió a la interfaz un botón para poder escoger el archivo a reproducir. Al hacer esto, en el nodo `turtle_bot_interface` se creó un cliente para solicitar el servicio `Recibir_nombre`, previamente iniciado en el nodo `turtle_bot_player`.

Para probar el punto, con Copelia corriendo:

1. Ejecutar los nodos `turtle_bot_interface` y `turtle_bot_player`
2. Seleccionar el boton de reproducir movimientos y seleccionar el archivo deseado
3. Observar en coppelia el movimiento del robot

Dependencias usadas:

- `std_msgs` : ya que el servicio recibe un `string` con la ubicación del archivo
- `turtlebot_interfaces`: paquete donde está almacenado el servicio `Recibir_nombre`

Dentro del archivo `package.xml` se incluyó:

```

<exec_depend>std_msgs</exec_depend>
<exec_depend>turtlebot_interfaces</exec_depend>

```