



## **Guía Laboratorio ErCoT – L2T**

### **Introducción:**

Bienvenido a esta guía de laboratorio, donde nos embarcamos en un apasionante recorrido por el fascinante mundo de la robótica. Aquí, nuestros objetivos se centran en aprender a operar y optimizar el entorno de enseñanza-aprendizaje ErCoT, respaldados por la sólida metodología L2T (Learn-To-Teach). Esta plataforma ha sido minuciosamente diseñada para abordar con eficacia las maniobras de recogida, colocación y transporte.

En el transcurso de esta guía, tanto estudiantes como docentes tendrán la oportunidad de sumergirse en una experiencia completa y detallada que abarca diversos aspectos relacionados con la robótica. Desde la instalación del software hasta el manejo y control de robots móviles y de brazo robótico, explicaremos cada paso para adentrarnos en el vasto mundo de la robótica y la ingeniería.

Estamos seguros de que este aprendizaje práctico nos brindará una comprensión más profunda de la robótica y nos permitirá desarrollar habilidades valiosas en este campo en constante evolución.

## **1. Software Necesario**

Para comenzar, necesitaremos instalar el sistema operativo Ubuntu en nuestra PC y luego configurar el framework ROS (Robot Operating System). Aquí encontraremos una guía paso a paso para realizar estas instalaciones y asegurarnos de contar con todos los paquetes y herramientas necesarios para trabajar con ErCoT.

### **1.1. Descargue e instale Ubuntu en su Pc**

#### ***1.1.1. Descargar imagen de Ubuntu***

Descargue la imagen de escritorio Ubuntu 18.04 LTS o Ubuntu 20.04 LTS adecuada para su PC desde los enlaces a continuación. De los siguientes enlaces:

- [Ubuntu 20.04 LTS Desktop image \(64-bit\)](#)
- [Ubuntu 18.04 LTS Desktop image \(64-bit\)](#)

#### ***1.1.2. Instalación de Ubuntu***

Siga las instrucciones a continuación para instalar Ubuntu en el PC. Puede ser directamente en Disco duro o máquina virtual



- Install Ubuntu desktop

### 1.2. Instalación del framework ROS en el Pc Estático (Master\_pc)

Para la instalación de ROS es preciso identificar la versión de Linux que se está ejecutando en su computador e instalar la versión de Linux compatible con esta.

Linux	Versión
18.04	ROS Melodic Morenia
20.04	ROS Noetic Ninjemys

Abra la terminal con Ctrl+Alt+T e ingrese los siguientes comandos uno a la vez.

```
$ sudo apt update
$ sudo apt upgrade
$ wget https://raw.githubusercontent.com/ROBOTIS-GIT/robotis\_tools/master/install\_ros\_noetic.sh
$ chmod 755 ./install_ros_noetic.sh
$ bash ./install_ros_noetic.sh
```

### 1.3. Posterior a la instalación de ROS es necesario instalar los siguientes paquetes:

*Los siguientes comandos están configurados para la versión Noetic de ROS, si su versión es Melodic, reemplace Melodic por Noetic en los campos que corresponda.*

- En una terminal de comandos Linux ejecute las siguientes líneas de instrucción, con el fin de actualizar e instalar los paquetes necesarios para el manejo de ErCoT:

```
$ sudo apt-get update
$ sudo apt-get upgrade
$ sudo apt-get install ros-noetic-joy ros-noetic-teleop-twistjoy
ros-noetic-teleop-twist-keyboard ros-noetic-laser-proc
ros-noetic-rgbd-launch ros-noetic-depthimage-to-laserscan
ros-noetic-rosserial-arduino ros-noetic-rosserial-python
ros-noetic-rosserial-server ros-noetic-rosserial-client
ros-noetic-rosserial-msgs ros-noetic-amcl ros-noetic-map-server
ros-noetic-move-base ros-noetic-urdf ros-noetic-xacro
ros-noetic-compressed-image-transport ros-noetic-rqt-image-view
ros-noetic-gmapping ros-noetic-navigation
ros-noetic-interactive-markers
$ sudo apt install ros-noetic-dynamixel-sdk
$ sudo apt install ros-noetic-turtlebot3-msgs
$ sudo apt install ros-noetic-turtlebot3
$ sudo apt install
```



Para realizar la operación remota de los equipos en red se hace necesario instalar un servidor ssh (Secure Shell):

```
$ sudo apt-get install openssh-server
```

Es necesario sincronizar la hora de los equipos en red, para ello instalamos el siguiente protocolo:

```
$ sudo apt-get install ntpdate
```

Por último, es necesario incorporar el paquete `usb_cam` para acceder a la cámara que se utilizará en el componente de visión por computador, se instalará así:

```
$ cd catkin_ws/src  
$ git clone https://github.com/bosch-ros-pkg/usb_cam.git  
$ cd ..  
$ catkin_make
```



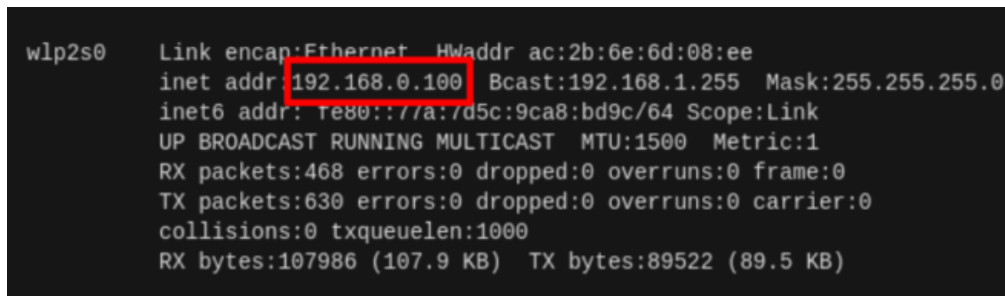
## **2. Configuración de Red ROS**

Una vez que tengamos nuestro entorno listo, explicaremos cómo configurar la red de ROS para establecer una conexión fluida entre el Master\_pc y el Turtlebot\_pc, permitiendo el control y la navegación del robot móvil de forma remota. A través de la teleoperación, aprenderemos a manejar el Turtlebot3 con un joystick en Arduino, mientras mapeamos el entorno para comprender su estructura y construir un mapa detallado.

### **2.1. Configuración del Master\_pc**

Conecte su PC a la red Wifi “Turtlebot” y encuentre la dirección IP asignada con el siguiente comando:

```
$ ifconfig
```



```
wlp2s0  Link encap:Ethernet  HWaddr ac:2b:6e:6d:08:ee  
        inet addr:192.168.0.100  Bcast:192.168.1.255  Mask:255.255.255.0  
        inet6 addr: fe80::7a:7d5c:9ca8:bd9c/64 Scope:Link  
        UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1  
        RX packets:468 errors:0 dropped:0 overruns:0 frame:0  
        TX packets:630 errors:0 dropped:0 overruns:0 carrier:0  
        collisions:0 txqueuelen:1000  
        RX bytes:107986 (107.9 KB)  TX bytes:89522 (89.5 KB)
```

Abra el archivo y actualice la configuración de IP de ROS con el siguiente comando:

```
$ gedit -bashrc
```

Al final del archivo agregue las siguientes líneas:

- export ROS\_HOSTNAME=192.168.1.100
- export ROS\_MASTER\_URI=http://{ROS\_HOSTNAME}:11311

*Tenga en cuenta que la dirección indicada con rojo debe ser la hallada en el Master\_pc.*

Finalmente guarde los cambios del documento y cierre el mismo.



## **2.2. Configuración Turtlebot\_pc**

En una nueva ventana de comandos y haciendo uso del *ssh server* accederemos de forma remota al Turtlebot\_pc con el siguiente comando:

```
$ ssh -Y turtlebot@192.168.1.13
```

A continuación, el sistema le solicitará una clave de acceso, digite la siguiente sin puntos ni espacios:

```
$ carlitosbot
```

luego abra el archivo y actualice la configuración de IP de ROS con el siguiente comando:

```
$ gedit -bashrc
```

Agregue las siguientes líneas de código al final del documento:

- `export ROS_HOSTNAME=192.168.1.13`
- `export ROS_MASTER_URI=http://192.168.1.100:11311`

*Tenga en cuenta que la dirección indicada con rojo debe ser la hallada en el Master\_pc.*



### **3. Teleoperación y Mapeo del Turtlebot3**

#### **3.1. Conexión con el Turtlebot3**

En una ventana de comandos del Master\_pc ejecute el núcleo de ROS, así:

```
$ roscore
```

Para establecer conexión con el turtlebot3 ejecute la siguiente línea de código en una nueva pestaña de comandos:

```
$ ssh -Y turtlebot@192.168.1.13
```

A continuación, el sistema le solicitará una clave de acceso, digite la siguiente sin puntos ni espacios:

```
$ carlitosbot
```

Al tener acceso al Turtlebot\_pc ejecute el paquete Bringup del robot, para iniciar la operación del robot móvil, así:

```
$ roslaunch turtlebot3_bringup turtlebot3_robot.launch
```

#### **3.2. Teleoperación del Turtlebot3**

*Conecte el joystick de Arduino a uno de los puertos USB disponibles en el Master\_pc.*

Posteriormente ejecute la siguiente línea de código, para configurar la comunicación serial de Arduino con ROS, así:

```
$ rosrund serial_python serial_node.py /dev/ttyUSB0
```

Ahora ejecute las siguientes líneas de código para lanzar el nodo ROS que permitirá la manipulación del robot con a través del joystick, así:

```
$ rosrund tdg_pkg teleop_joy.py
```

Ahora podemos navegar a través de la pista haciendo uso del joystick.

#### **3.3. Mapeo haciendo uso del Turtlebot3 y Rviz**

En una nueva ventana de comandos ejecute el nodo SLAM para el robot móvil, de esta manera se visualizará el entorno por medio del LiDar, así:

```
$ roslaunch turtlebot3_slam turtlebot3_slam.launch
```



Desplace el robot a través de la pista (De inicio a fin y viceversa) para que el robot identifique el entorno, y construya el mapa de lo que está observando a su alrededor.

En una nueva ventana de comandos ejecute la siguiente instrucción, que permitirá almacenar el mapa del entorno construido por el robot:

```
$ roslaunch map_server map_saver -f ~/map
```



## 4. Navegación autónoma del Turtlebot3

### 4.1. Requisitos previos:

Para realizar la navegación autónoma usted debe tener lo siguiente:

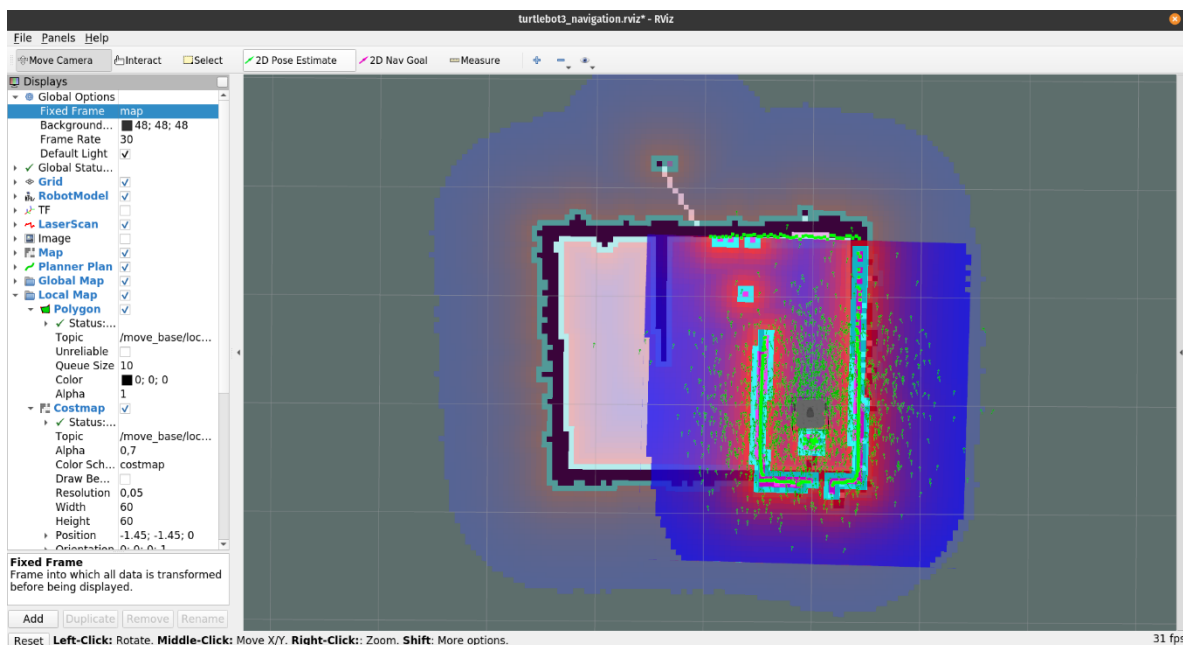
- Conectar la red ROS (entre el Master\_pc y Turtlebot3\_pc).
- Ejecutar el roscore en Master\_pc.
- Ejecutar el bringup robot en Turtlebot3\_pc.
- Tener el mapa del entorno (creado por medio de SLAM)

### 4.2. Navegación autónoma a través de Rviz

En una nueva ventana de comandos ejecutar la siguiente línea de código para ejecutar el Rviz y el entorno de navegación autónoma:

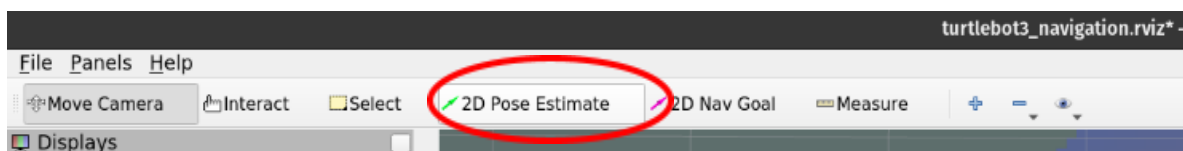
```
$ roslaunch turtlebot3_navigation  
turtlebot3_navigation.launch map_file:=$HOME/map.yaml
```

Se abrirá una ventana como esta:



En la parte superior se encuentran dos iconos de gran importancia,

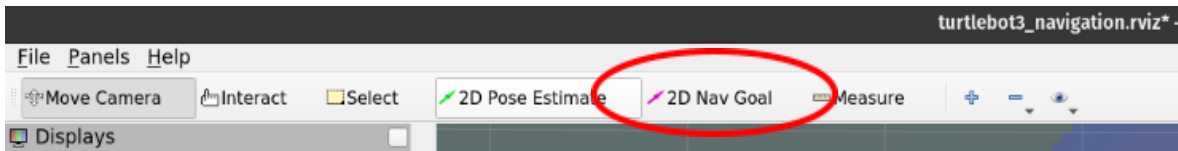
- El primero es para darle la ubicación inicial al robot



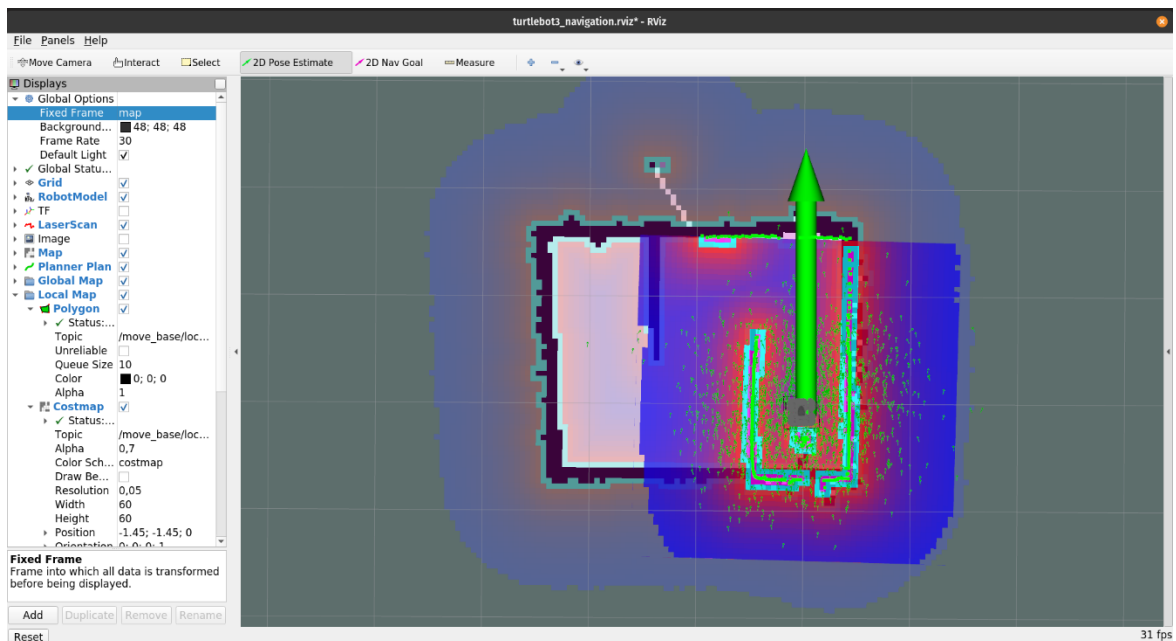




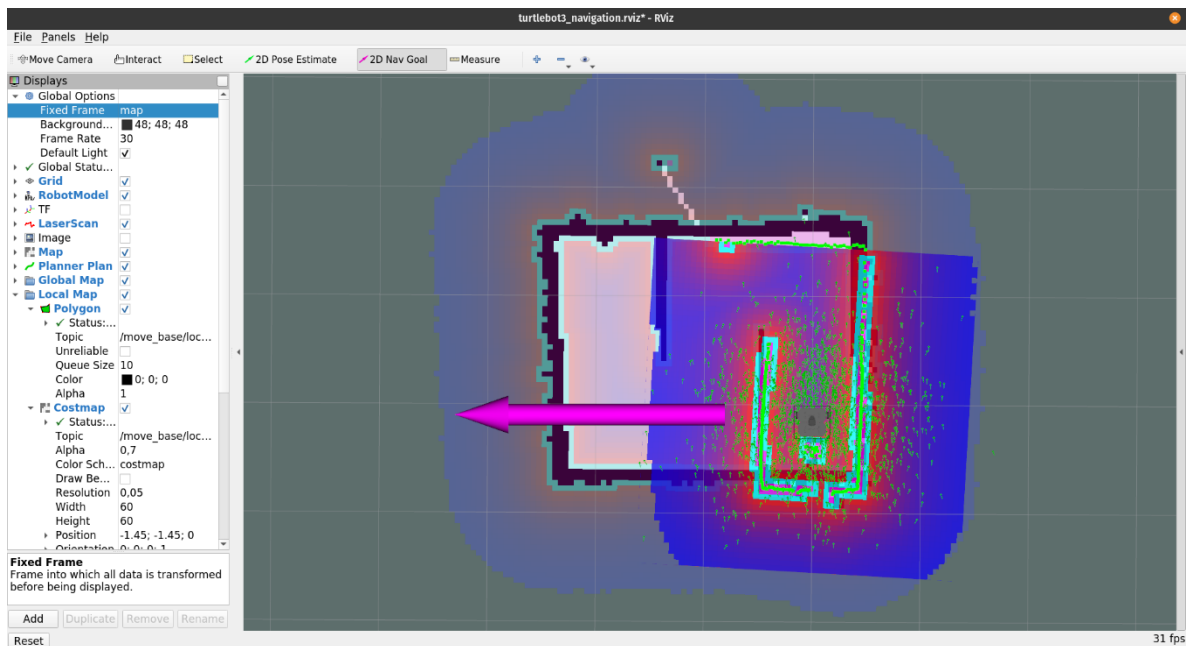
- El segundo es para darle la ubicación final u objetivo de la trayectoria al robot



Haciendo uso de la flecha verde ubique el robot en el entorno hacia la dirección que está apuntando, procure que lo que se detecta a través del lidar concuerde con el mapa previamente elaborado, así:



Haga clic en el mapa para establecer el destino del robot y arrastre la flecha purpura hacia la dirección hacia la que estará mirando el robot.



Ahora el robot se desplaza de forma autónoma según se le indique.



## **5. Recogida y colocación con Dobot Magician**

En esta guía, también abordaremos la recogida y colocación de un objeto sobre nuestro turtlebot, haciendo uso de un Dobot Magician, un fascinante robot de brazo robótico. Veremos los requisitos previos para su correcto funcionamiento y aprenderemos a conectarlo y ejecutar comandos de Python para realizar tareas de recogida y colocación de objetos.

### **5.1. Requisitos previos:**

- Tener alimentación de tensión en el Dobot
- Realizar correcta conexión de la bomba de vacío al Dobot
- Tener alimentación de tensión en el computador NUC
- Realizar conexión al computador NUC por medio del cable serial
- Encender el Dobot y el computador NUC
- Ubicar el robot móvil turtlebot3 en el punto de recogida

### **5.2. Ejecución del código en Python.**

Para establecer conexión con el Dobot\_pc ejecute la siguiente línea de código en una nueva pestaña de comandos:

```
$ ssh -Y turtlebot@192.168.1.14
```

A continuación, el sistema le solicitará una clave de acceso, digite la siguiente sin puntos ni espacios:

```
$ carlitosbot
```

Posterior a esto se indica la ruta de ubicación del robot y se ejecuta el script de Python necesario para tomar dos piezas y ubicarlas sobre el robot móvil. Con los siguientes comandos:

```
$ cd Dobot/  
$ python3 recogida.py
```

Una vez realizado el proceso de recogida y colocación se puede desplazar el robot móvil al punto final del circuito, siguiendo los pasos descritos para la navegación autónoma.