

# Manual de instalación de Docker y ejecución del entorno LIMO-Humble

UruBots Team

13 de noviembre de 2025

## Índice

1. Introducción	2
2. Instalación de dependencias básicas	2
3. Configuración del repositorio oficial de Docker	2
4. Instalación de Docker	3
5. Configuración de permisos para el usuario	3
6. Descarga del proyecto LIMO-Humble	3
7. Construcción de la imagen Docker	3
8. Ejecución del contenedor	3
9. Acceso al entorno	4
10. Posibles errores y soluciones	4
11. Guía para utilizar el robot LIMO en simulación	5
12. Conclusión	6

# 1. Introducción

Este manual describe paso a paso cómo instalar Docker en un sistema **Ubuntu 20.04 o superior**, configurar los permisos necesarios y ejecutar la imagen del proyecto **limo-humble-docker**, que contiene un entorno de trabajo con ROS2 Humble, una interfaz gráfica accesible vía navegador y todos los paquetes de simulación del robot LIMO.

## 2. Instalación de dependencias básicas

Comando:

```
sudo apt update
```

**Descripción:** Actualiza la lista de paquetes disponibles desde los repositorios del sistema para asegurar que las versiones más recientes estén disponibles.

```
sudo apt install -y ca-certificates curl gnupg lsb-release
```

**Descripción:** Instala herramientas esenciales para descargar y validar paquetes de software firmados digitalmente.

## 3. Configuración del repositorio oficial de Docker

```
sudo mkdir -p /etc/apt/keyrings
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | \
sudo gpg --dearmor -o /etc/apt/keyrings/docker.gpg
```

**Descripción:** Crea el directorio donde se guardarán las claves GPG y descarga la clave de autenticación oficial de Docker, necesaria para verificar la integridad de los paquetes.

```
echo \
"deb [arch=$(dpkg --print-architecture) \
signed-by=/etc/apt/keyrings/docker.gpg] \
https://download.docker.com/linux/ubuntu \
$(lsb_release -cs) stable" | \
sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
```

**Descripción:** Agrega el repositorio oficial de Docker a la lista de fuentes de software del sistema.

## 4. Instalación de Docker

```
sudo apt update  
sudo apt install -y docker-ce docker-ce-cli containerd.io docker-compose-plugin
```

**Descripción:** Actualiza nuevamente los índices de los repositorios e instala Docker Engine, el cliente CLI, el motor de contenedores containerd y el complemento docker-compose.

## 5. Configuración de permisos para el usuario

```
sudo usermod -aG docker $USER  
newgrp docker
```

**Descripción:** Agrega al usuario actual al grupo docker para permitir la ejecución de comandos Docker sin necesidad de usar sudo. El comando newgrp docker actualiza la sesión actual con los nuevos permisos.

## 6. Descarga del proyecto LIMO-Humble

```
git clone https://github.com/UruBots/limo-humble-docker.git  
cd limo-humble-docker
```

**Descripción:** Descarga el repositorio oficial del proyecto y accede al directorio donde se encuentra el Dockerfile necesario para construir la imagen.

## 7. Construcción de la imagen Docker

```
docker build -t limo-humble .
```

**Descripción:** Crea una imagen Docker llamada limo-humble a partir del Dockerfile del proyecto. El proceso puede demorar varios minutos dependiendo de la velocidad de conexión y del hardware disponible.

## 8. Ejecución del contenedor

```
docker run -it --rm \  
-p 6080:80 \  
--name limo_humble_container \  
limo-humble
```

**Descripción:** Ejecuta un contenedor basado en la imagen creada. El parámetro -p 6080:80 permite acceder al entorno gráfico desde el navegador ingresando a <http://localhost:6080>. El flag -rm elimina el contenedor automáticamente al cerrarlo, evitando acumulación de instancias viejas.

## 9. Acceso al entorno

Una vez que el contenedor está en ejecución, abrir el navegador y dirigirse a:

**http://localhost:6080**

Desde allí se podrá acceder al escritorio remoto del entorno ROS2 Humble y ejecutar nodos o simulaciones.

## 10. Posibles errores y soluciones

### Error 1: Docker no arranca o genera conflictos durante la instalación

En caso de que el servicio Docker falle al iniciarse o se presenten errores durante la instalación, se puede reinstalar completamente con los siguientes comandos:

```
sudo apt remove -y docker-ce docker-ce-cli containerd.io docker-compose-plugin  
sudo rm -rf /var/lib/docker /var/lib/containerd  
sudo apt install -y docker-ce docker-ce-cli containerd.io docker-compose-plugin
```

**Descripción:** Elimina completamente las instalaciones previas de Docker y vuelve a instalar los componentes necesarios desde cero. Esto soluciona problemas de configuración dañada o versiones conflictivas.

### Error 2: Docker no inicia automáticamente o presenta fallas al ejecutar comandos

Si el demonio de Docker no está activo, puede reiniciarse con los siguientes comandos:

```
sudo systemctl restart docker  
sudo systemctl status docker
```

**Descripción:** Reinicia el servicio principal de Docker y muestra su estado actual. Debe indicar **Active: active (running)** si el sistema se encuentra funcionando correctamente.

## 11. Guía para utilizar el robot LIMO en simulación

Este apartado explica los pasos necesarios para lanzar el robot LIMO dentro del entorno de simulación con Gazebo. Asegúrese de ejecutar estos comandos desde una terminal dentro del contenedor Docker o en un entorno ROS2 Humble correctamente configurado.

### Paso 1: Cargar el entorno de trabajo

```
source install/setup.bash
```

**Descripción:** Carga las configuraciones del espacio de trabajo (`workspace`) generado tras la compilación con `colcon build`. Esto permite que ROS2 reconozca los paquetes y launch files disponibles.

### Paso 2: Solucionar posibles errores de Gazebo

```
source /usr/share/gazebo/setup.bash
```

**Descripción:** Carga las variables de entorno necesarias para ejecutar Gazebo. En algunos casos, este paso evita errores relacionados con modelos o rutas faltantes durante el arranque de la simulación.

### Paso 3: Lanzar el robot LIMO en Gazebo

```
ros2 launch limo_description gazebo_models_diff.launch.py
```

**Descripción:** Inicia la simulación de LIMO en Gazebo utilizando el modelo diferencial. Aparecerá una ventana de Gazebo con el robot en un entorno vacío o preconfigurado según el archivo de lanzamiento.

### Paso 4: Publicar velocidades en el robot

```
ros2 topic pub -r 5 /cmd_vel geometry_msgs/msg/Twist \
"{linear: {x: 0.3, y: 0.0, z: 0.0}, angular: {x: 0.0, y: 0.0, z: 0.0}}"
```

**Descripción:** Publica mensajes de velocidad en el tópico `/cmd_vel`. En este ejemplo, el robot se mueve hacia la derecha a 0.3 m/s.

### Consejos adicionales

- Use Ctrl + C para detener la publicación o cerrar la simulación.
- Verifique que el tópico `/cmd_vel` esté activo con:

```
ros2 topic list
```

- Si Gazebo se queda en Preparing your world..., fuerce el renderizado por software:

```
LIBGL_ALWAYS_SOFTWARE=1 gazebo
```

## 12. Conclusión

Con estos pasos se completa la instalación y ejecución del entorno limo-humble-dock permitiendo disponer de un sistema de desarrollo reproducible para proyectos con ROS2 y simulación.