

## Informe Técnico – Configuración y Ejecución del Proyecto ROS

### 1. Diagnóstico y reparación de hardware

Durante la etapa inicial del proyecto se detectó un problema de comunicación entre el sensor LiDAR SICK LMS511 y el computador. Tras realizar un análisis de continuidad eléctrica, se identificó que el conector del sensor presentaba daños por sobrecalentamiento, impidiendo su correcta detección por el sistema. El componente defectuoso fue reparado exitosamente, restableciendo la comunicación entre el sensor y el equipo de cómputo.

#### 1.1 Asignacion de IP a maquina virtual (CADA VEZ QUE LA MAQUINA VIRTUAL SE APAGA)

Se debe hacer un nuevo direccionamiento de la ip de la interfaz de la VirtualBox para poder obtener internet y comunicación con el sensor:

```
ip link show #muestra las interfaces utilizadas  
sudo ip addr flush dev enp0s8 #verificar que interfaz que se comunica con el sensor  
sudo ip addr add 192.168.0.20/24 #pone ip nueva  
sudo ip link set enp0s8 up #Enciende la tarjeta de red con la nueva ip
```

### 2. Gestión del repositorio y dependencias

Instalar dependencias para conexión y generación de consultas con la base de datos:

#### 2.1 Actualizar gestor de paquetes

```
sudo apt update
```

#### 2.2 Instalar SQLAlchemy y el driver de SQL server:

```
pip3 install sqlalchemy pymssql
```

Posteriormente, se procedió a la migración completa del proyecto, incluyendo archivos fuente y dependencias, hacia un repositorio gestionado mediante GitHub Desktop. Como parte de la validación del entorno, se ejecutó el archivo de prueba ejemplo.py, confirmando el correcto funcionamiento del proyecto en el nuevo repositorio.

Es importante destacar que cada vez que se desee agregar un nuevo archivo ejecutable en Python (.py), este debe ubicarse dentro de la carpeta truck\_positioning. Adicionalmente, es obligatorio registrar dicho archivo en el archivo setup.py, específicamente dentro de la sección entry\_points, siguiendo la siguiente estructura:

```
'nombre_elegido = truck_positioning.nombre_archivo:main'
```

Este procedimiento asegura que ROS2 pueda reconocer y ejecutar correctamente los nodos definidos.

### 3. Proceso de compilación y ejecución

Para la correcta compilación y ejecución del proyecto se deben utilizar múltiples terminales, siguiendo el procedimiento descrito a continuación:

#### 3.1 Compilación del paquete

En una primera terminal, se debe acceder al directorio raíz del proyecto:

```
cd ~/Documents/GitHub/ProyectoROS
```

Luego, se compila específicamente el paquete truck\_positioning utilizando el siguiente comando:

```
colcon build --packages-select truck_positioning
```

Una vez finalizada la compilación, se debe cargar el entorno de trabajo generado:

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

#### 3.2 Configuración y ejecución del sensor LiDAR

Antes de ejecutar cualquier comando ROS2, es indispensable cargar el entorno base de ROS Humble:

```
source /opt/ros/humble/setup.bash
```

Posteriormente, se lanza el nodo de configuración del sensor LiDAR LMS511 mediante el siguiente comando:

```
ros2 launch sick_scan_xd sick_lms_5xx.launch.py hostname:=192.168.0.2
```

Este comando permite configurar parámetros críticos del sensor, tales como la dirección IP, el puerto de comunicación, la frecuencia de escaneo y el sistema de coordenadas.

#### 3.3 Ejecución del nodo del proyecto

En una segunda terminal, se repite el proceso de carga del entorno y se ejecuta el nodo correspondiente al archivo previamente registrado en setup.py:

```
ros2 run truck_positioning nombre_archivo_sobrenombr_en_setup.py
```

### 3.4 Visualización de la nube de puntos

En una tercera terminal, se inicializa la herramienta RViz2, la cual permite visualizar y configurar la nube de puntos generada por el sensor:

```
rviz2
```

### 3.5 Visualización del grafo de nodos

Finalmente, en una cuarta terminal se ejecuta la herramienta rqt\_graph, que permite analizar gráficamente la comunicación entre nodos, identificando publicadores, suscriptores y los tópicos utilizados:

```
rqt_graph
```

Este procedimiento facilita la verificación del correcto flujo de información dentro del sistema ROS2.