

UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO  
FACULTAD DE CIENCIAS  
TALLER DE ROBÓTICA 2024-1



---

## Detección de paneles solares en ambientes virtuales

---

### **Integrantes y correos:**

**Del Moral Morales Francisco Emmanuel**

fcoemmdmm@ciencias.unam.mx

**Liprandi Cortes Rodrigo**

godites@ciencias.unam.mx

## 1. Arquitectura del proyecto

El proyecto se conforma de 3 subsistemas para poder realizar la detección de los paneles y los movimientos del robot dentro del ambiente virtual.

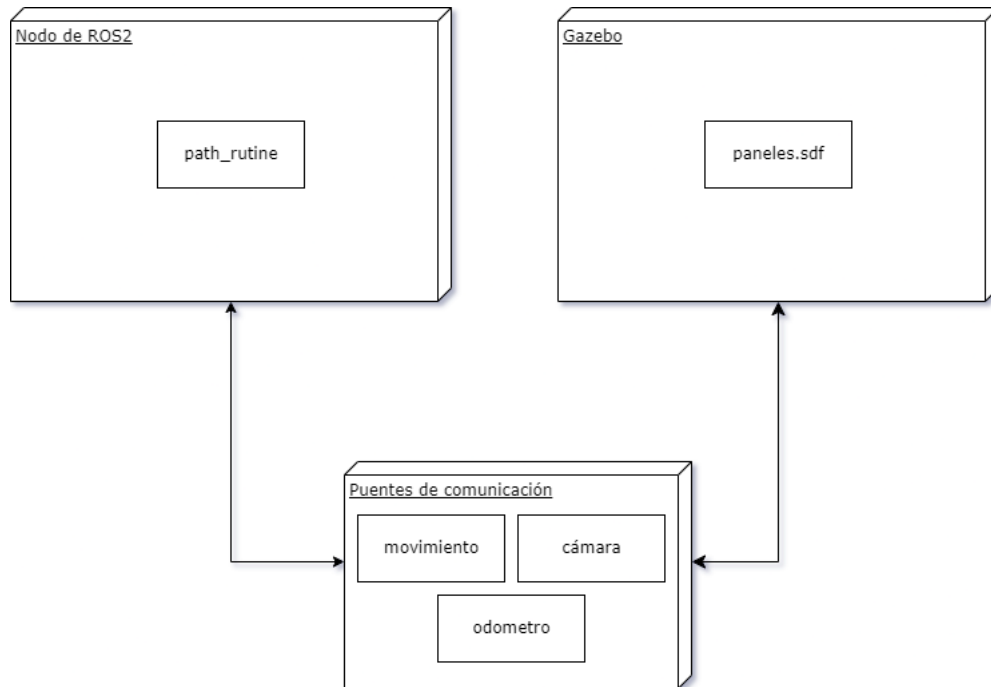


Figura 1: Arquitectura del sistema representado como bloques

A continuación se muestra el árbol de directorios del proyecto:

```

Ambiente-Paneles
├── ambiente_panel
│   ├── src
│   │   ├── camera_images
│   │   ├── movement
│   │   ├── panel_solar
│   │   ├── robot
│   │   └── paneles.sdf
├── ros2_ws
│   ├── src
│   │   ├── ros_gz
│   │   └── ros_gz_bridge
  
```

### 1.1. Mundo virtual en Gazebo

El ambiente virtual se conforma de un archivo *.sdf*, que representa nuestro mundo virtual, que a su vez usa modelos en formato *.dae* para crear las mallas de los paneles solares y de el robot.

1. *paneles.sdf* - Mundo virtual, este es el archivo principal de nuestro ambiente virtual. Aquí definimos

propiedades del mundo así como los enlaces, colisiones y articulaciones del robot. También definimos propiedades de la interfaz para poder visualizar la cámara y otra información desde Gazebo.

2. **panel\_solar** - Directorio donde se encuentran los archivos del modelo de los paneles solares, estos fueron hechos en Blender y exportados a stl y dae.
  - *panel\_solar.dae* archivo usado para crear las colisiones de los paneles solares.
  - *panel\_solar.stl* archivo usado para la parte visual de los paneles solares.
3. **robot** - Directorio donde se encuentran los archivos del modelo del robot
  - *main.body.dae* - Archivo donde se define la parte visual del cuerpo del robot.
  - *wheel.dae* - Archivo donde se define la parte visual de las llantas del robot.

## 1.2. Puentes de comunicación entre ROS2 y Gazebo

Los puentes nos sirven para establecer una comunicación entre ROS2 y Gazebo, de esta forma podemos enviar y recibir datos o comandos de ambas plataformas.

- **ros\_gz\_bridge** es el nodo que provee los puentes y permiten el intercambio de mensajes entre ROS2 y Gazebo, sólo tiene soporte para ciertos tipos de mensajes pero para este proyecto sólo necesitamos 3 tipos de mensajes:
  1. Mensajes de tipo Twist para poder mover al robot.
  2. Mensajes de tipo Image para la cámara virtual.
  3. Mensajes de tipo TF para conocer la posición del robot.

## 1.3. Nodo de ROS2 para ejecutar la rutina de movimiento y detección de paneles solares

El nodo usa mensajes de tipo Twist para el movimiento del robot. Además se usa la biblioteca OpenCV para detectar los paneles solares.

# 2. Instalación

Se recomienda el uso de Ubuntu 22.04 ya que en este sistema operativo se desarrolló el proyecto y las tecnologías mencionadas abajo funcionan bien en este sistema.

## 2.1. Requerimientos

### ■ Python

Los nodos de ROS 2 para el movimiento están hechos en python, además se requiere de la biblioteca **Open CV** para la detección de paneles solares.

```
1 sudo apt update # Actualizar los paquetes del sistema
2 sudo apt install python3 # Instala python
3 pip install opencv-python # Instala OpenCV
4 pip install numpy # Dependencia
5 pip install empy # Dependencia
6 pip install lark # Dependencia
```

### ■ ROS 2

Se requiere de una instalación de ROS 2 para poder ejecutar el nodo de movimiento y reconocimiento de paneles solares, en el desarrollo de este proyecto se usó la distribución **Humble**.

En este enlace se encuentra un tutorial de cómo instalarlo.

### ■ Gazebo

Se requiere instalar Gazebo para poder visualizar el mundo virtual y poder obtener información del robot como su posición y las imágenes de la cámara virtual. En este proyecto se usó Gazebo Ignition.

En este enlace se encuentra un tutorial de cómo instalarlo.

## 2.2. Descargar el proyecto

El código se encuentra actualmente en el siguiente repositorio: <https://github.com/ShadoRoca/Ambiente-Paneles>

Para poder descargarlo:

```
1 git clone https://github.com/ShadoRoca/Ambiente-Paneles
```

## 2.3. Compilación

Compilar los paquetes de ROS

```
1 # Terminal 1
2 cd ambiente_panel
3 source /opt/ros2/humble
4 colcon build
```

```
1 # Terminal 2
2 cd ros2_ws
3 source /opt/ros2/humble
4 colcon build
```

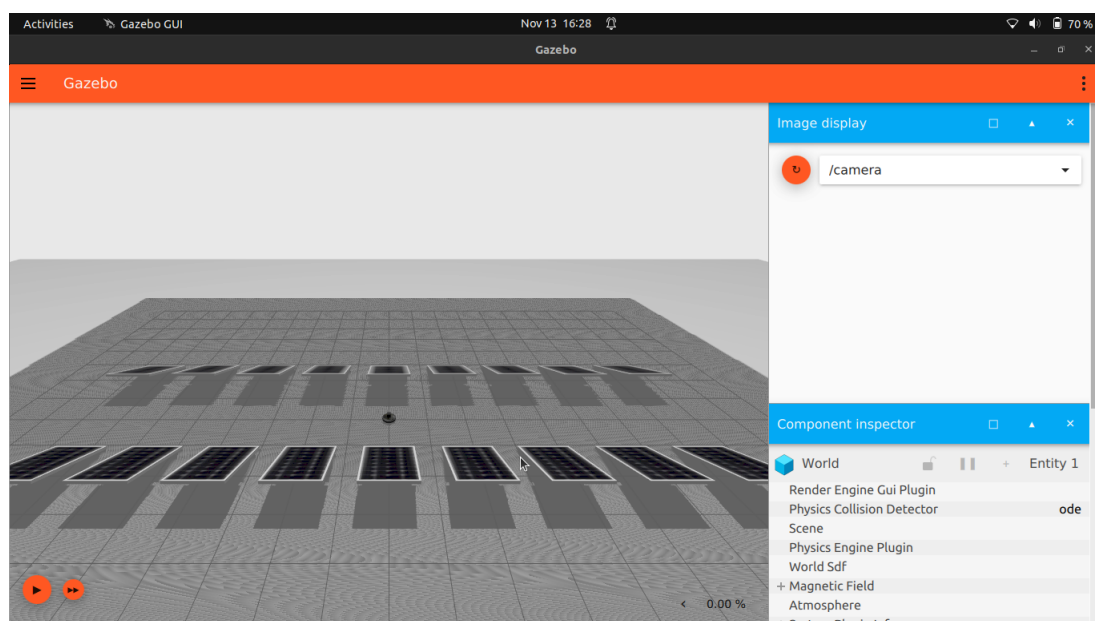
## 3. Ejecución del proyecto

Usar nuevas terminales para los siguientes pasos

### 3.1. Ambiente virtual

1. Iniciar el ambiente virtual

```
1 # Terminal 1
2 cd ambiente_panel/src
3 ign gazebo paneles.sdf
```



2. Dar click en el ícono de play en la esquina inferior izquierda para iniciar la simulación

### 3.2. Puentes de comunicación

```
1 # Terminal 2
2 cd ros2_ws
3 source /opt/ros2/humble
4 source install/setup.bash
5
6 ros2 run ros_gz_bridge parameter_bridge /commands/velocity@geometry_msgs
  /msg/Twist@ignition.msgs.Twist
```

```
1 # Terminal 3
2 cd ros2_ws
3 source /opt/ros2/humble
4 source install/setup.bash
5
6 ros2 run ros_gz_bridge parameter_bridge /camera@sensor_msgs/msg/
  Image@ignition.msgs.Image
```

```
1 # Terminal 4
2 cd ros2_ws
3 source /opt/ros2/humble
4 source install/setup.bash
5
6 ros2 run ros_gz_bridge parameter_bridge /model/kobuki_standalone/
  tf@tf2_msgs/msg/TFMessage@gz.msgs.Pose_V
```

### 3.3. Rutina de movimiento y detección de paneles

```
1 # Terminal 5
2 cd ambiente_panel
3 source /opt/ros2/humble
4 source install/setup.bash
5
6 ros2 run movement path_routine
```