

[1.- Actualizar ordenador 1](#_Toc367385886)

[2.- Instalar Gazebo 1](#_Toc1703276909)

[3.- Preparar entorno ROS2 2](#_Toc1747297495)

[4.- Lanzar Gazebo 2](#_Toc1222296531)

[5.- Dependencias y paquetes Gazebo/ROS2 3](#_Toc1763146519)

[6.- Modificar nuestro modelo. 4](#_Toc1016784837)

[7.- Instalar dependencias con rosdep 6](#_Toc254326243)

[8.- Primer lanzamiento del modelo 7](#_Toc1001437935)

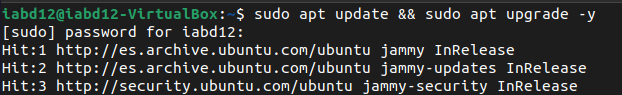
[9.- Arreglar el modelo 8](#_Toc1097456748)

[10.- Finalizacion del modelo 9](#_Toc1755942036)

## 1.- Actualizar ordenador

Obviamente, haces de hacer cualquier cosa, habra que actualizar por completo el ordenador. Para ello usaremos:

sudo apt update && sudo apt upgrade -y



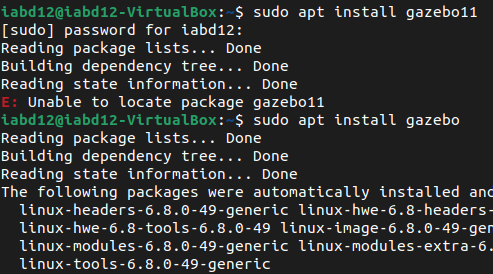
## 2.- Instalar Gazebo

Ahora, con el ordenador actualizado, hay que instalar gazebo. Para ello, puedes probar con:

sudo apt install gazebo11

Si este no funciona, simplemente con:

sudo apt install gazebo



## 3.- Preparar entorno ROS2

Ya con gazebo instalado, tendremos que configurar el entorno de ROS2 para que funcione:

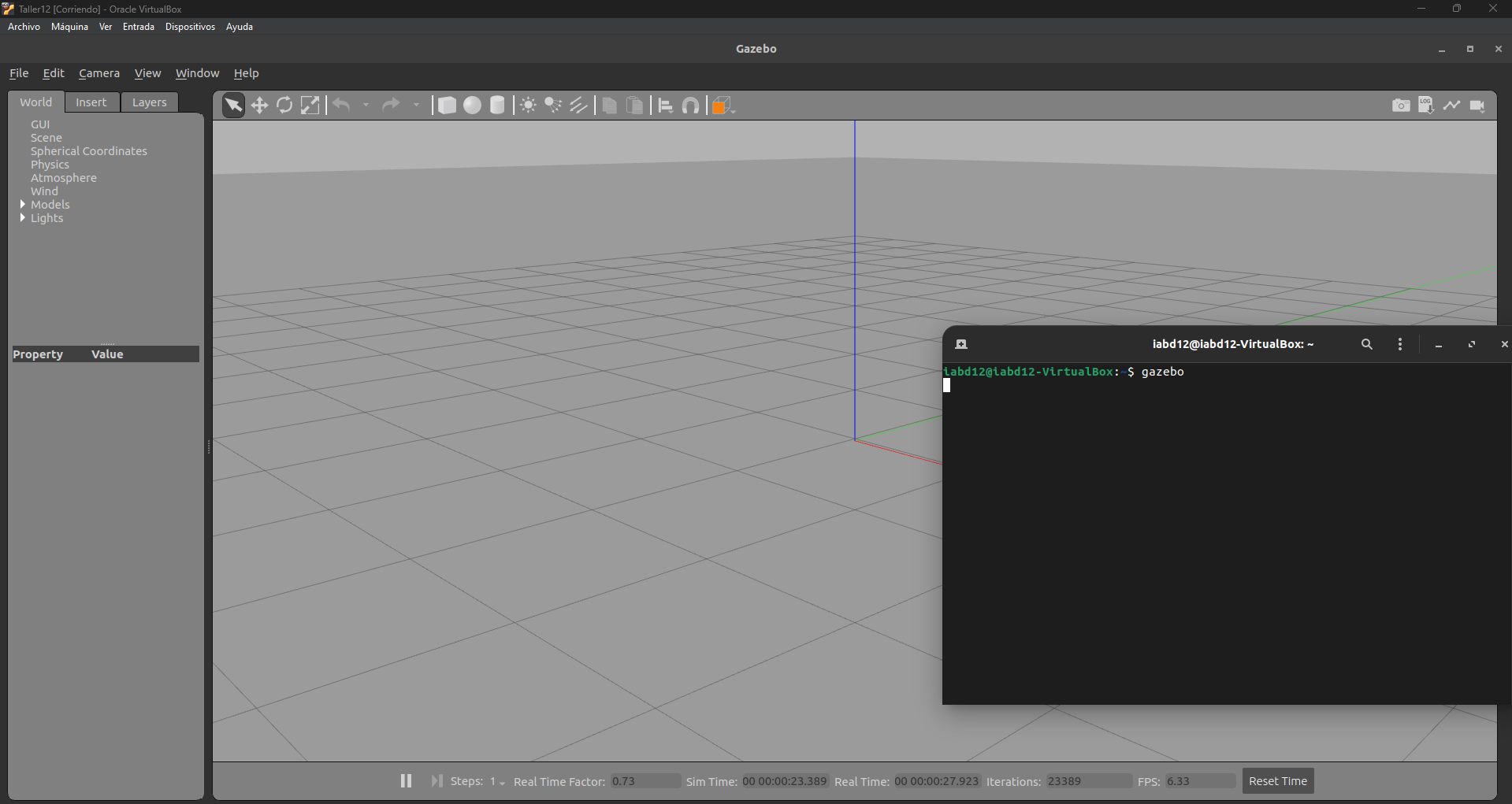
source /opt/ros/humble/setup.bash



## 4.- Lanzar Gazebo

Con el entorno ya preparado y configurado. Probamos a lanzar gazebo:

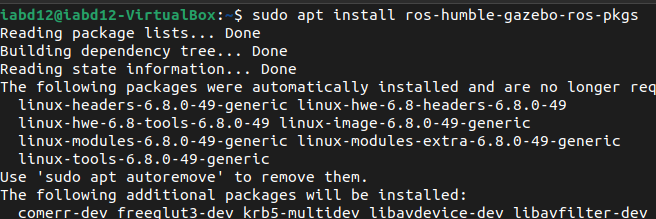
gazebo



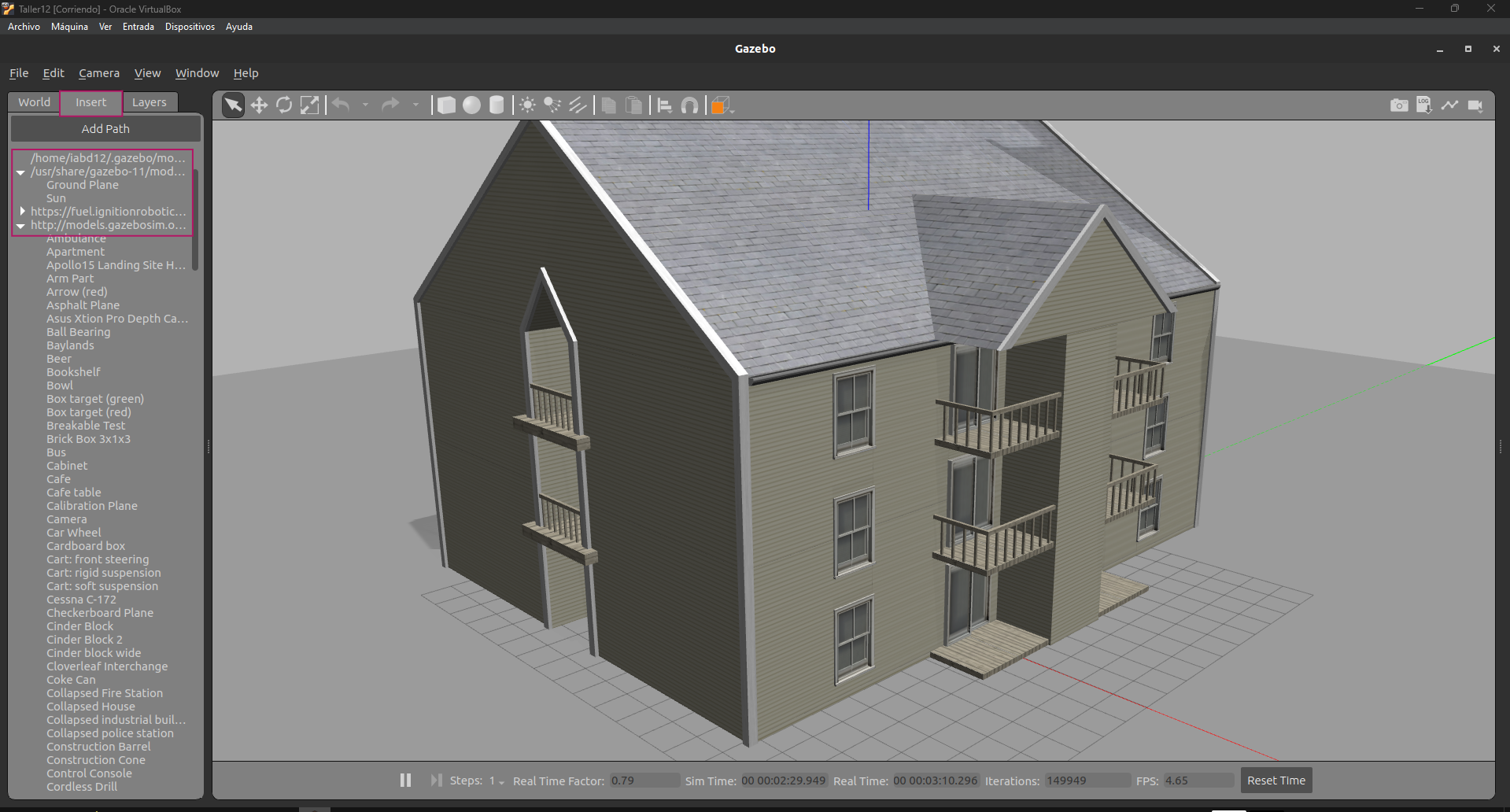
## 5.- Dependencias y paquetes Gazebo/ROS2

Si conseguimos lanzar gazebo sin problemas, podremos instalar las dependencias y paquetes necesarios para lo siguiente en la tarea. Para ello:

sudo apt install ros-humble-gazebo-ros-pkgs -y



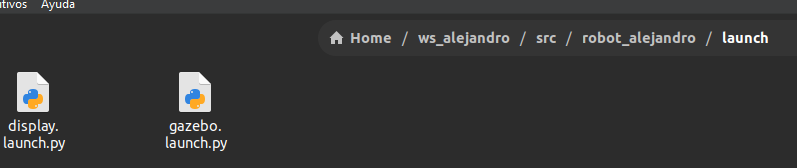
Después de instalarlo, si volvemos a entrar en Gazebo, podremos cargar modelos dados por la misma compañía. Entrando en “Insert” y eligiendo uno de los nombres que aparecen podremos hacerlo perfectamente.



## 6.- Modificar nuestro modelo.

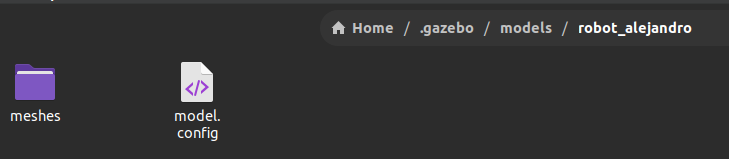
Ahora, tendremos que actualizar nuestro modelo personal, para poder visualizarlo con gazebo. Para ello, descargaremos el archivo “gazebo.launch.py” dado, y lo pondremos en la carpeta “launch” de nuestro modelo. La ruta debería ser así:

~/home/<carpeta\_workspace>/src/<nombre\_modelo>/launch



Aparte, en la nueva carpeta que se nos ha creado con gazebo, tendremos que crear una carpeta que se llame “models”, y dentro de este, otra que se llame como nuestro modelo. Dentro de este necesitaremos pegar la carpeta “meshes” de nuestro modelo de ROS2, y crear un archivo llamado “model.config” el cual tendrá nuestra configuración.

~/home/.gazebo/models/<nombre\_modelo>/meshes



Cambios para model.config:

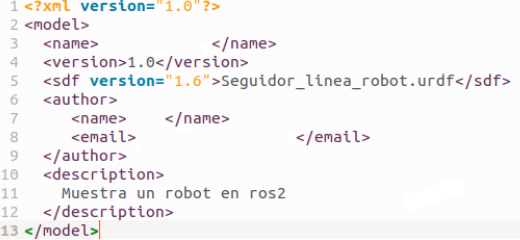
<name> “nombre\_modelo” </name>

<author>

<name> “nombre\_creador” </name>

<email> “email\_creador” </email>

</author>



## 7.- Instalar dependencias con rosdep

Por primera vez, sino lo habias hecho ya antes (yo si), deberas lanzar:

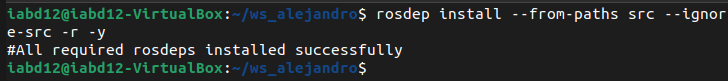
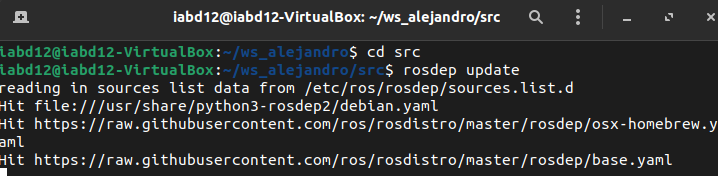
sudo rosdep init

Despues de lanzarlo o ya habiendolo lanzado:

rosdep update

Por ultimo, despues de actualizar:

rosdep install --from-paths src --ignore-src -r -y



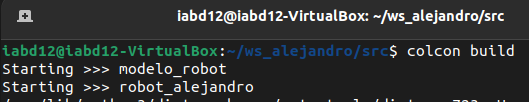
## 8.- Primer lanzamiento del modelo

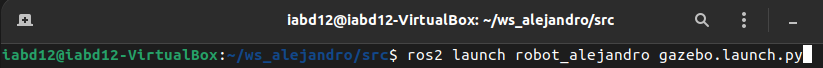
Ya con todo cambiado, probamos a lanzarlo por primera vez en nuestro workspace:

colcon build

source install/setup.bash

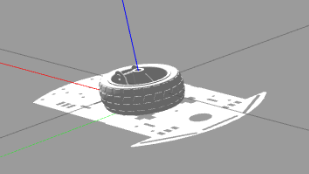
ros2 launch <nombre\_modelo> <nombre\_lanzador.py>





## 9.- Arreglar el modelo

Si todo ha ido como esperábamos, nos deberá aparecer esto:



Para ello, en el archivo “Seguidor\_linea\_robot.urdf” dentro de la carpeta “urdf” de nuestro modelo de ros2 (workspace). Tendremos que añadir lo siguiente en cada etiqueta que se llame “<inertial>”:

<inertial>

<inertia

ixx="0.0001"

ixy="0"

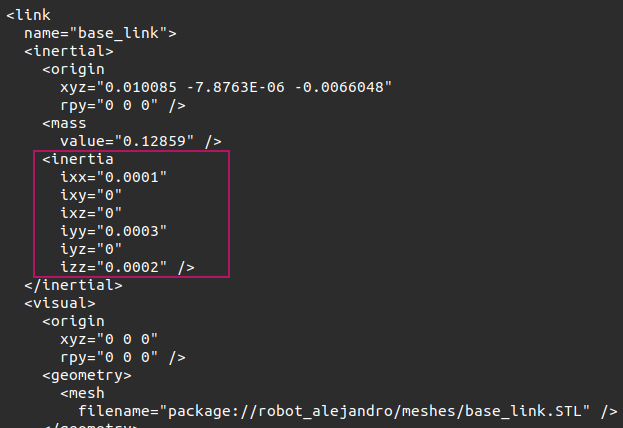
ixz="0"

iyy="0.0003"

iyz="0"

izz="0.0002" />

</inertial>

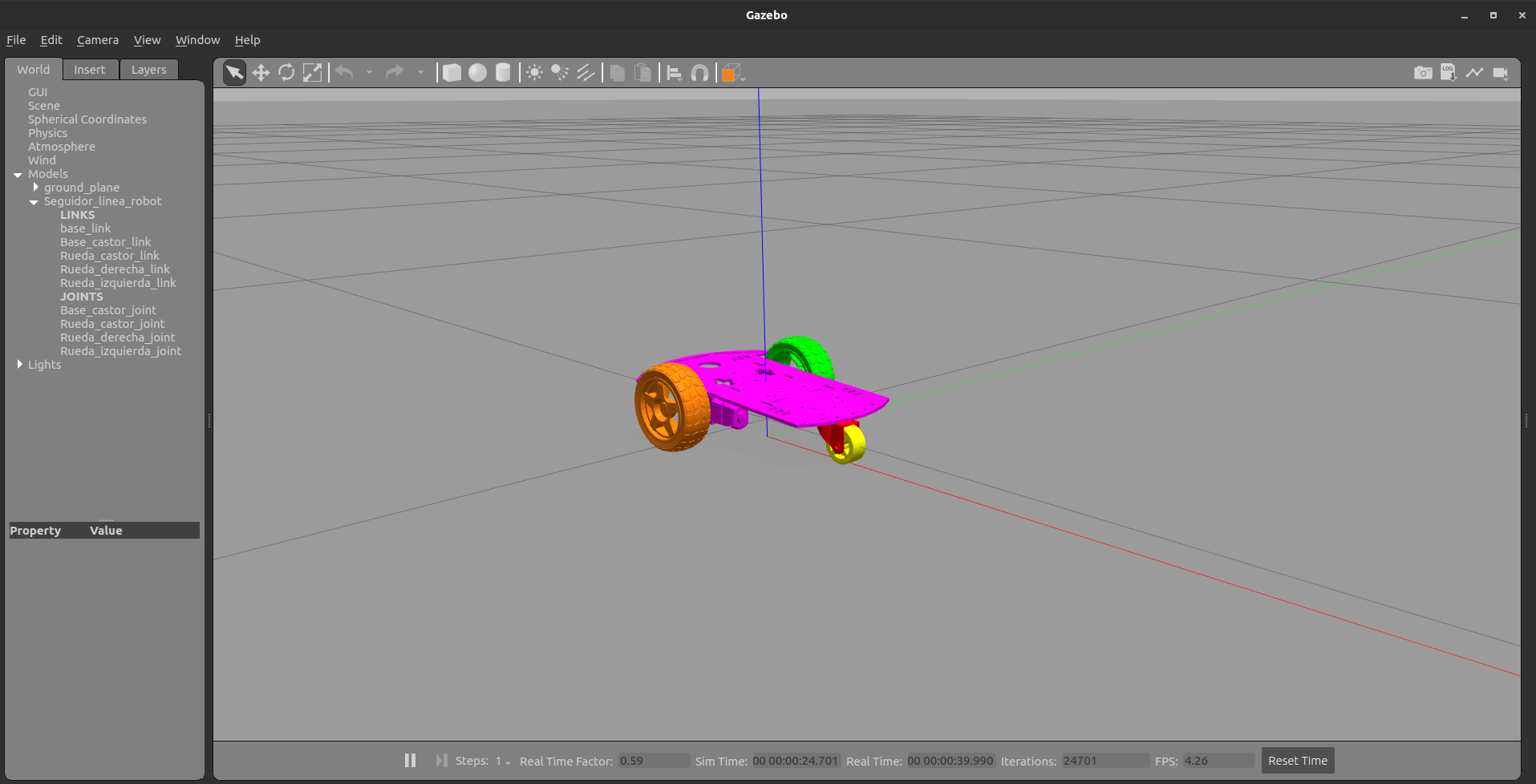


## 10.- Finalización del modelo

Cuando cambiemos el archivo por completo. Volveremos a hacer:

colcon build

Y lo volveremos a lanzar.



Ahora, podremos aplicar fuerzas al modelo para fuerza y torque y jugar un poco con ello.

