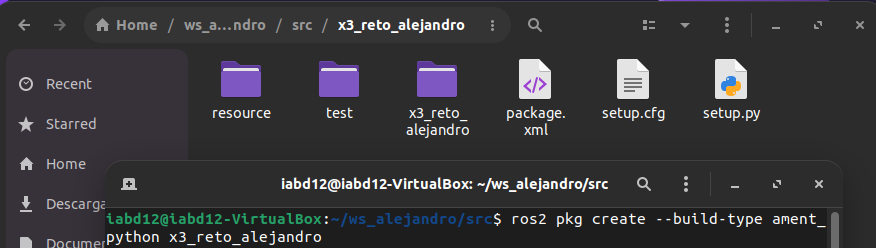
RETO X3

ROS2 y RVIZ2

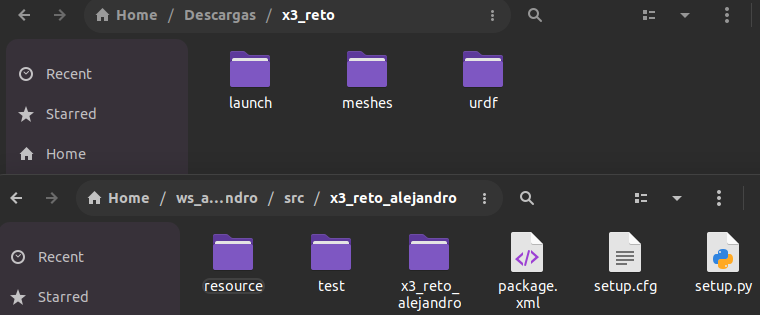
Alejandro Cruz Aguilar

Como ya tenemos el entorno preparado desde la práctica anterior, no hace falta explicarlo.

Simplemente empezaremos desde el momento en que creamos el directorio que guardara todo lo que tenga que ver con el modelado del robot:

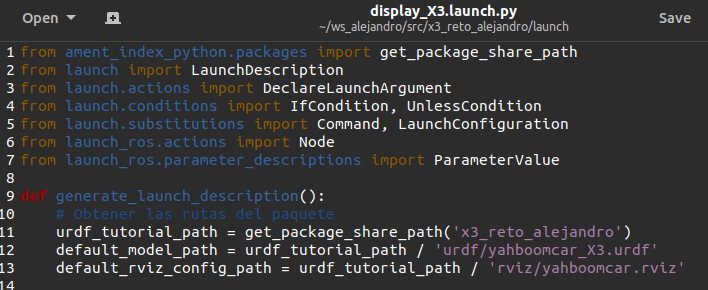


Ahora, con el archivo del reto x3 descargado, nos copiamos esas carpetas en nuestro proyecto.



Tenemos que cambiar en el lanzador, para que identifique nuestro paquete (proyecto). Buscamos la linea “urdf\_tutorial\_path”, y cambiamos la linea para que sea así:

urdf\_tutorial\_path = get\_package\_share\_path('<nombre\_proyecto\_x3>')



De nuevo, volvemos a cambiar el archivo “package.xml”, ya que viene por defecto, con nuestras necesidades:

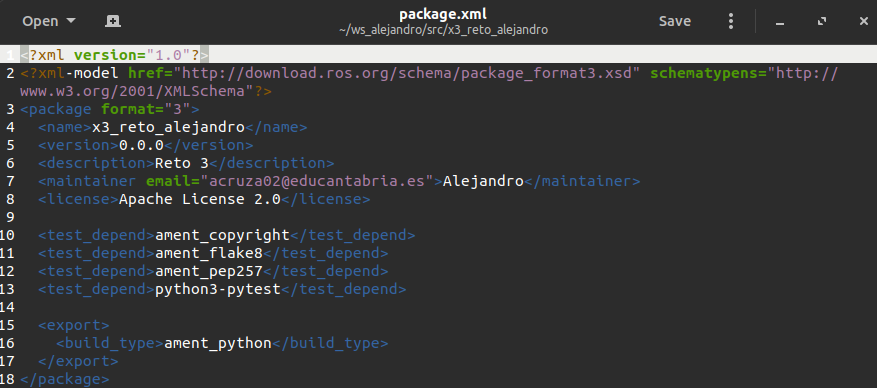
<name>x3\_reto\_alejandro</name>

<version>0.0.0</version>

<description>Reto 3</description>

<maintainer email="acruza0z@educantabria.es">Alejandro</maintainer>

<license>Apache License 2.0</license>



Y, de nuevo, volvemos a cambiar el setup.py, pero muy atentos, ya que no es igual.

Ahora, el “data\_files”, debe contener exactamente lo siguiente:

data\_files=[

('share/ament\_index/resource\_index/packages', ['resource/' + package\_name]),

('share/' + package\_name, ['package.xml']),

(os.path.join('share', package\_name, 'urdf'), glob('urdf/\*')),

(os.path.join('share', package\_name, 'launch'), glob('launch/\*')),

(os.path.join('share', package\_name, 'meshes'), glob('meshes/\*.STL')),

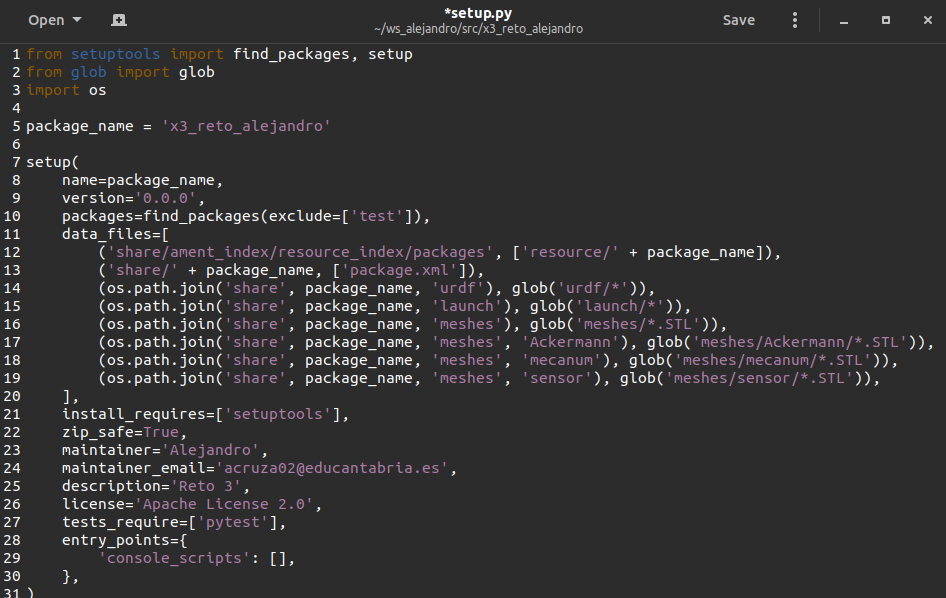
(os.path.join('share', package\_name, 'meshes', 'Ackermann'), glob('meshes/Ackermann/ \*. STL'))

(os.path.join('share', package\_name, 'meshes', 'mecanum'), glob('meshes/mecanum/ \*. STL')),

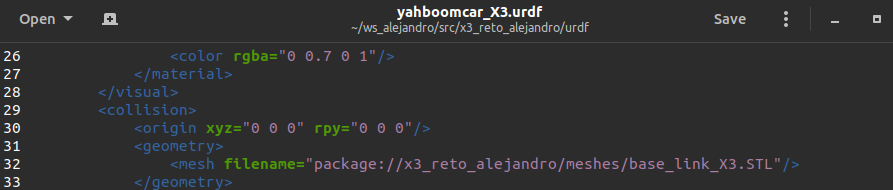
(os.path.join('share', package\_name, 'meshes', 'sensor'), glob('meshes/sensor/ \*. STL'))

]

Esto es, debido a que dentro de la carpeta de “meshes”, tenemos unas subcarpetas, las cuales, al leerlas, no cuentan como un archivo valido, así que por ello tenemos que leerlas por separado.

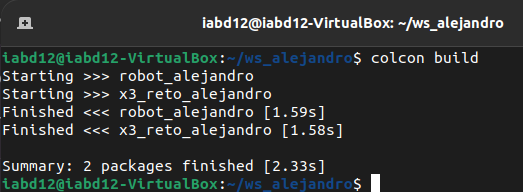


Por si acaso, en el urdf, debemos cambiar la ruta del paquete, si es que tenemos un nombre distinto, en todas las líneas en las que ponga “<mesh filename=”” />”, para que la ruta sea “package://<nombre\_proyecto>/---”



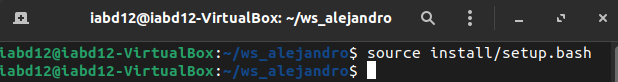
Al terminar de cambiarlo, volvemos a la carpeta raíz y hacemos un “colcon build”

colcon build



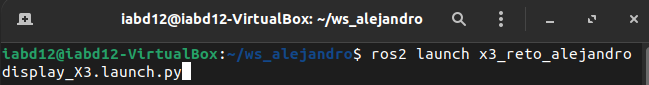
También, para actualizar, lanzamos “source install/setup.bash”

source install/setup.bash



Y lanzamos el modelo:

ros2 launch <nombre\_proyecto\_x3> <nombre\_python\_launcher\_x3>



Ahora aquí dentro, lo más probable es que haya el mismo error que en el anterior manual. En ese mismo esta la solución.

