

Materia:

Programación De Robots Industriales.

Integrantes:

Alvarado Contreras Cesar Omar.

Marcos Manzo Torrez.

Grado y grupo:

6°A

Practica 1:

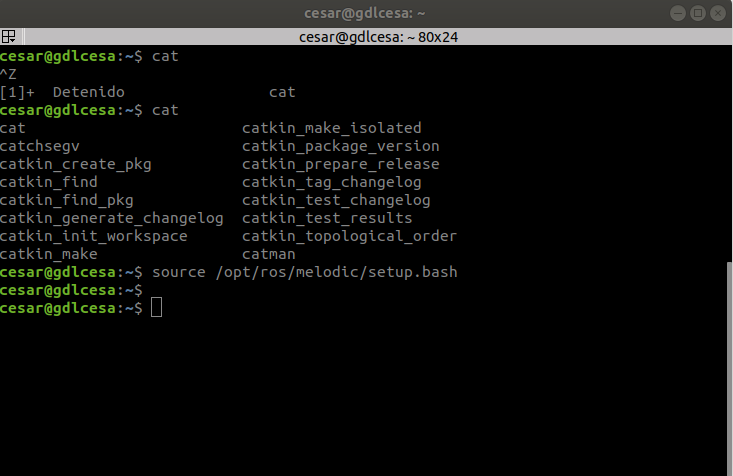
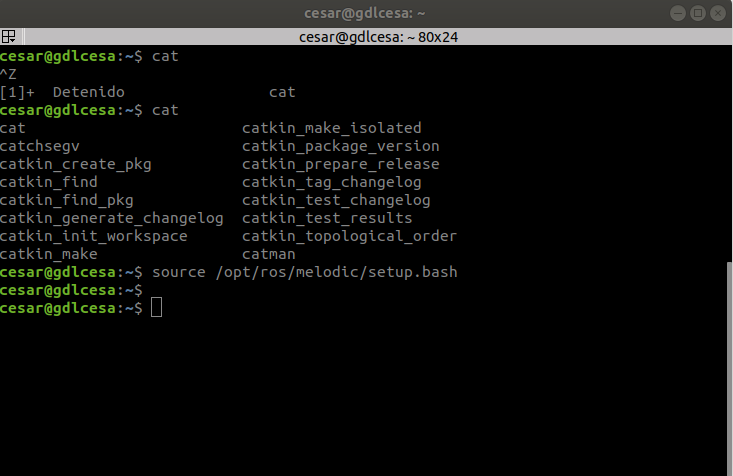
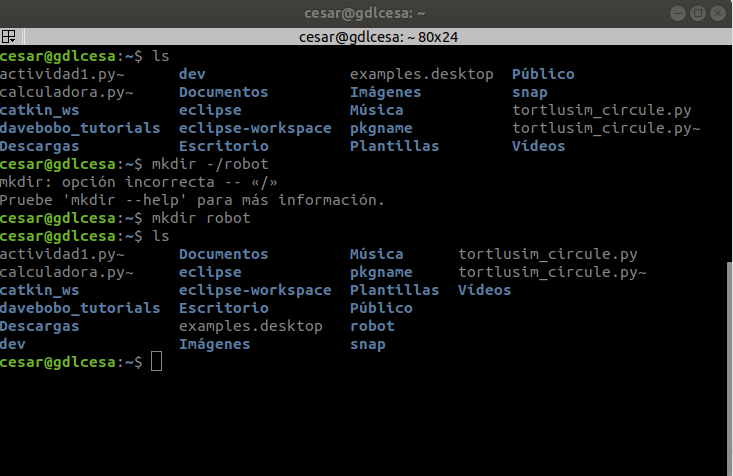
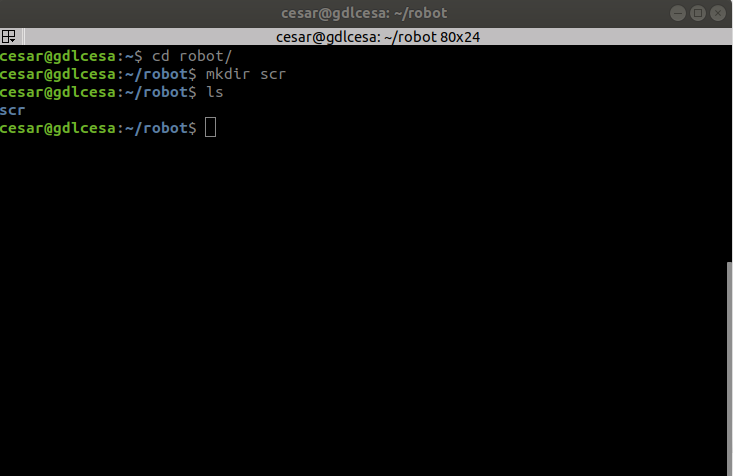
Manipulador industrial de robots en modo manual.

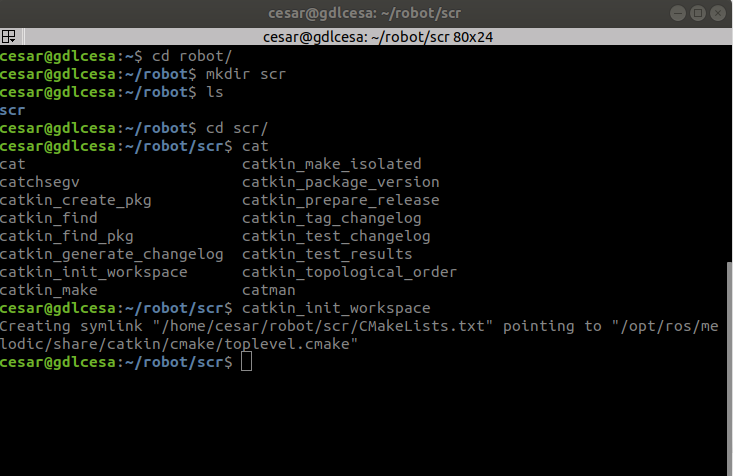
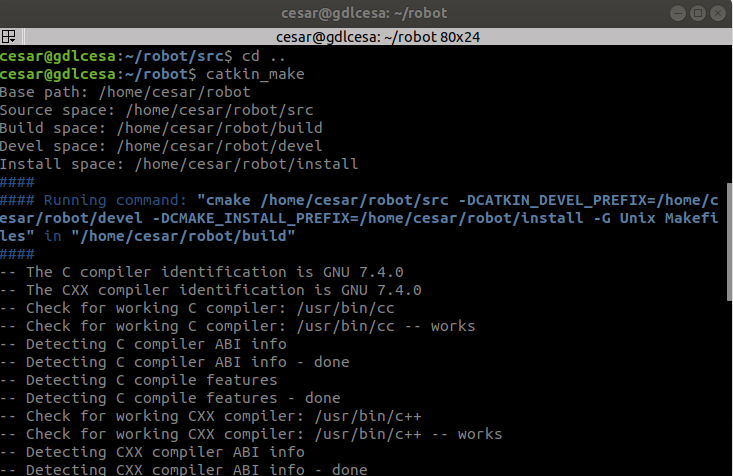
**Objetivo.**

Construcción y simulación de articulaciones de un robot (brazo robótico) con Ros y Gazebo.

|  |
| --- |
| Materiales. |
| ° Computadora con: |
| ° Software Gazebo. |
| ° Software ROS. |
| ° Software Eclipse. |
| ° Software Rende. |

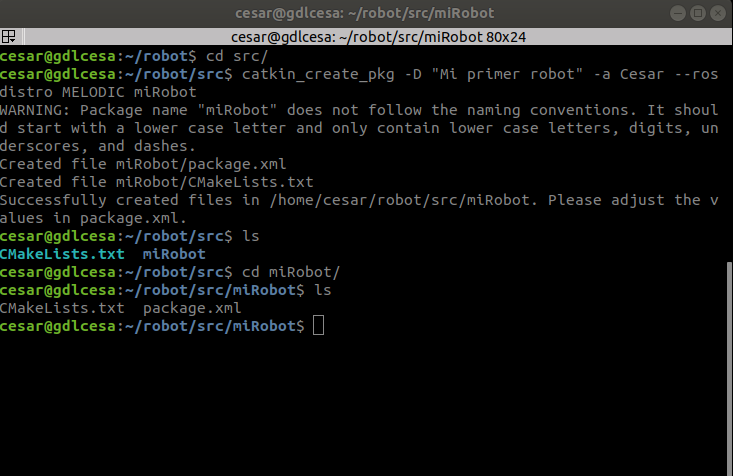
**Procedimiento.**

1. Primero creamos el entorno de trabajo para nuestro robot usamos **$cat (tabulador)**.
2. Correspondiente mente ejecutamos ros **$source /*opt/*ros/melodic/setup.bash**.
3. Conseguido esto vamos a decidir la carpeta que contendrá nuestro entorno de trabajo, creamos la carpeta ~/robot con el siguiente comando **$mkdir robot**.
4. Dentro de esta carpeta /robot crearemos otra carpeta *robot/*src con el mismo comando **$mkdir src**. Pero para ello vamos a ubicarnos en la carpeta robot con **$cd robot**.
5. Creamos el entorno de trabajo catkin **$catkin\_init\_workspace**.

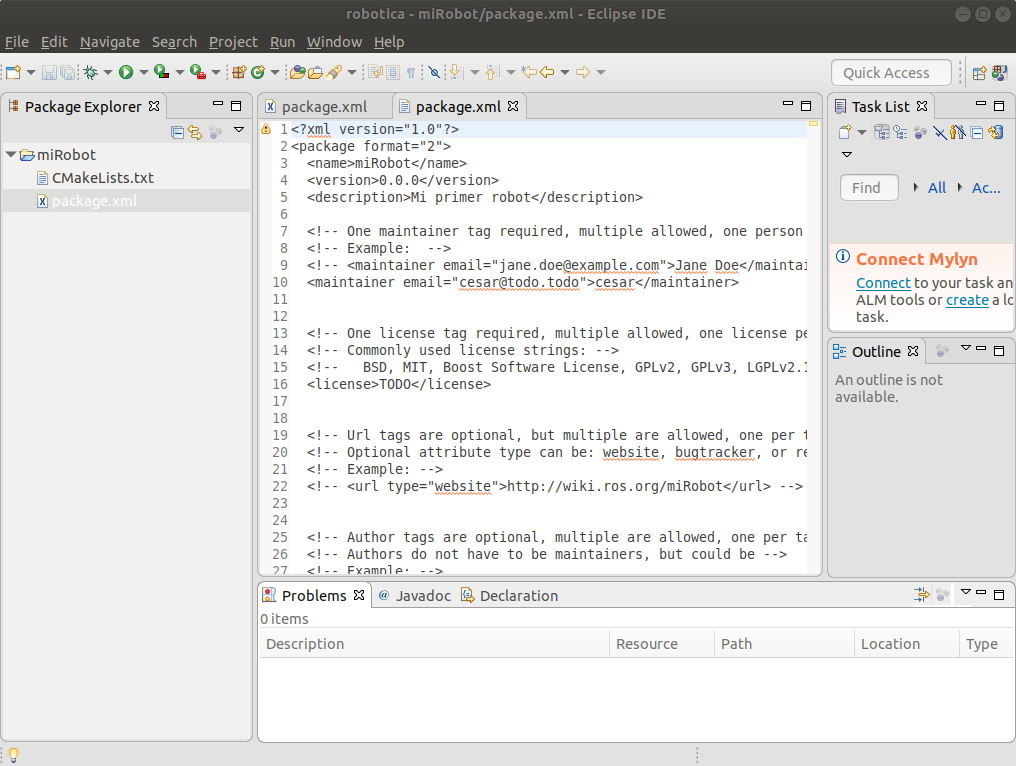
Regresamos a la carpeta robot **$cd ..** seguido creamos nuestro entorno de trabajo para el robot donde se desarrollará **$catkin\_make**.

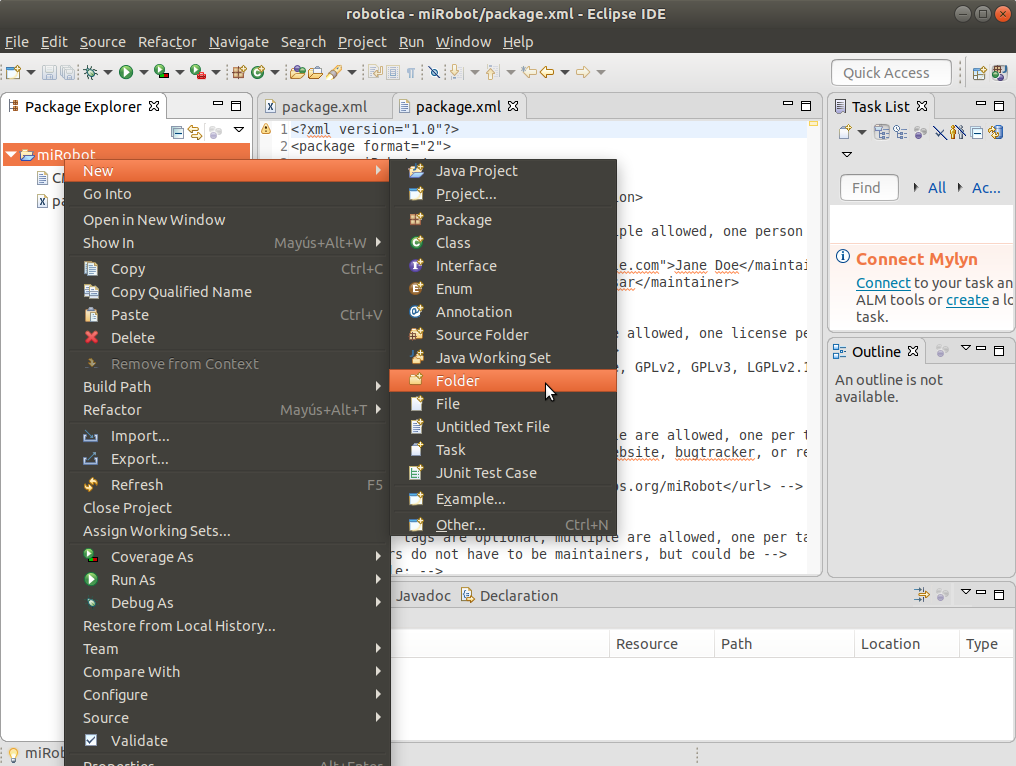
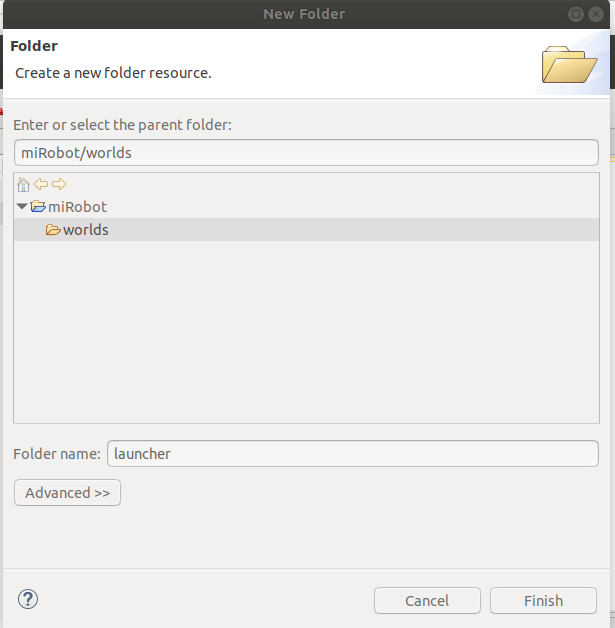
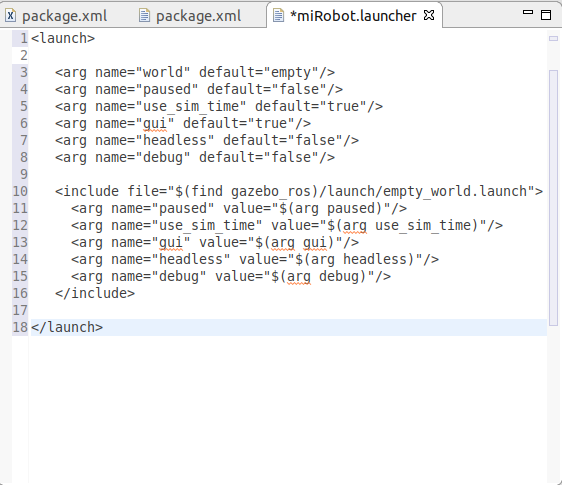
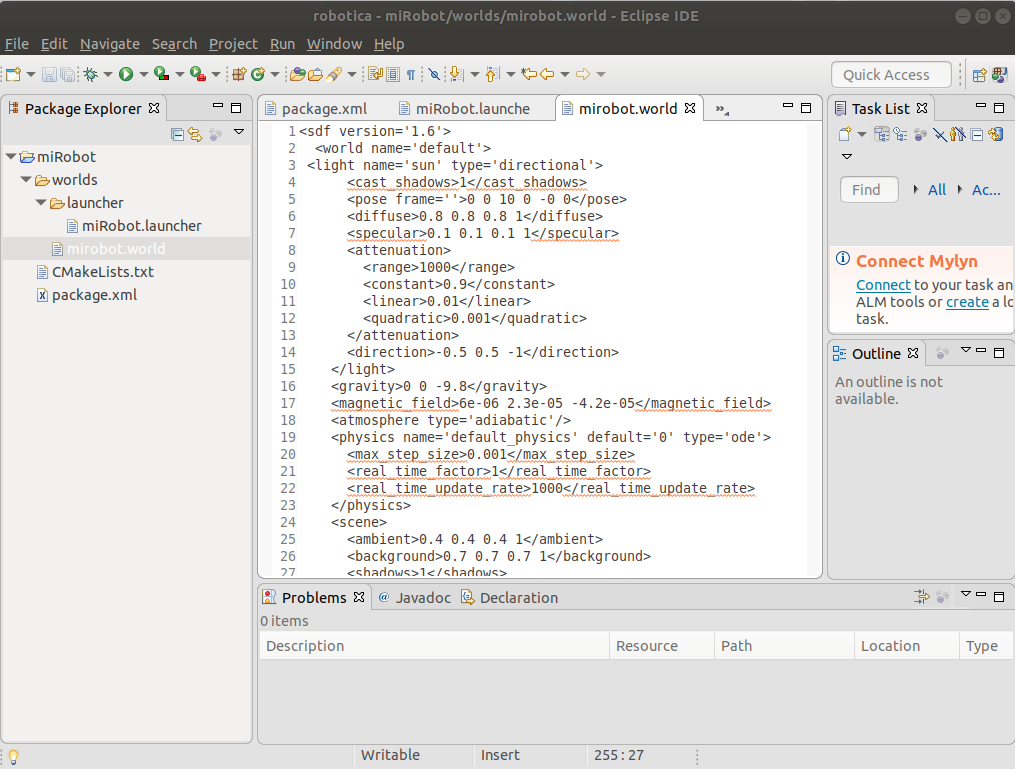
1. Con esto tendremos nuestro entorno de desarrollo para el robot.

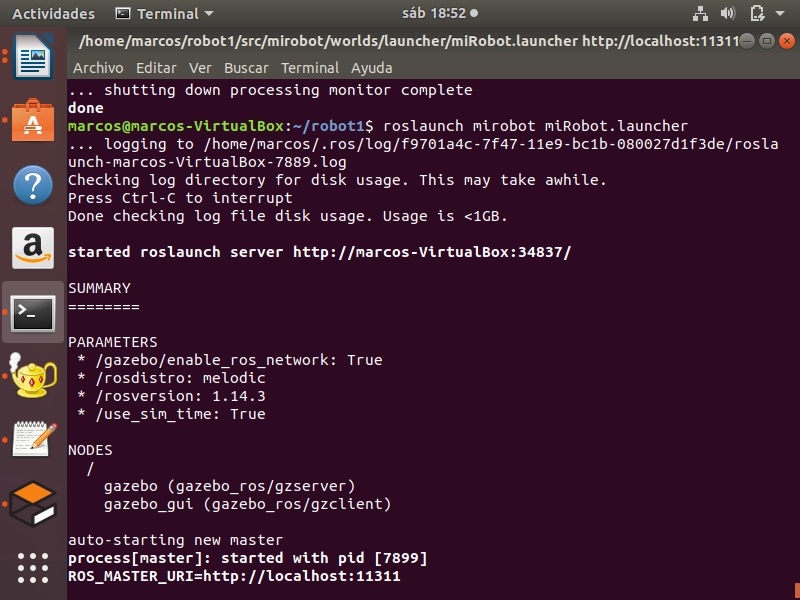
Para empezar a diseñar nuestro robot crearemos unos cuanto paquetes y librerías que iremos predefiniendo, iremos a nuestra carpeta /src y crearemos un paquete básico para programar **$catkin\_create\_pkg –rosdistro MELODIC miRobot**. El cual nos crea lo que se muestra en la siguiente imagen.

1. Una vez hecho esto ejecutamos el programa eclipse.

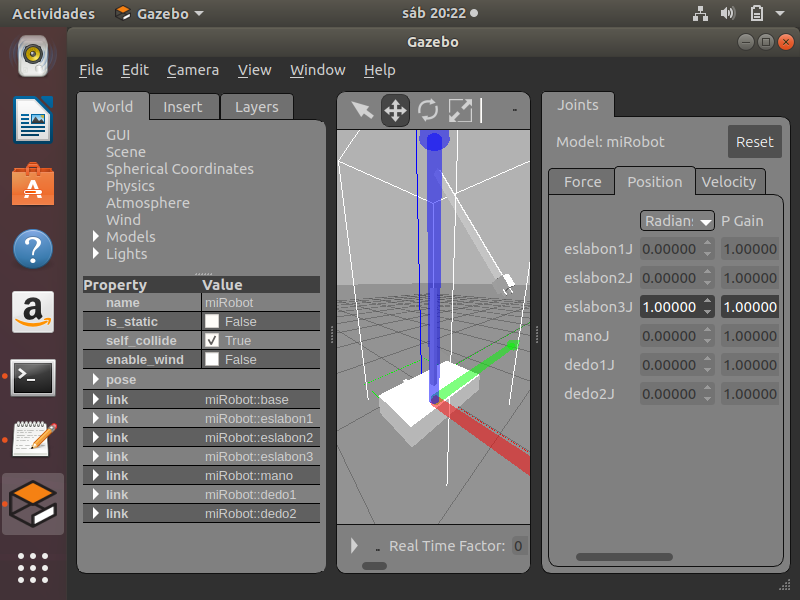
Creamos un proyecto general sin código de java, c, c+, etc.

El cual a la hora de crear lo le diremos que lo guarde en la ruta /*robot*/src/miRobot.

1. Que dando algo así cuando terminemos de dar la ruta, Creamos una carpeta llamada **/worlds** como se muestra en la siguiente imagen.
2. Dentro de la carpeta **/worlds** se creará otra llamada **/launcher**.
3. El cual nos ayudara para crear un **file** lanzador para Ros. Que contendrá el siguiente código.
4. Una vez creado este archivo lo siguiente es crear un file en la carpeta /worlds que se llamara mirobot.world. Este archivo será el entorno grafico 3D, de nuestro robot el cual podremos darle atributos, diseño, escenario, etc. el cual nos quedaría de esta forma.
5. Una vez terminado todo este proceso podremos ejecutar nuestro robot con **$roslaunch miRobot miRobot.launcher.** El cual nos abrirá ROS y Gazebo para visualizar nuestro robot, veremos algo parecido cuando ejecutemos el comando en la terminal como en la siguiente imagen.



**Resultados.**

El resultado de este procedimiento fue la visualización de un robot (brazo robótico) con eslabones, grados de libertas y como le afecta la gravedad a nuestro maquina cuando se le aplica gravedad.

**Conclusiones.**

Manzo:

Al haber generado nuestro robot, con sus seis piezas diferentes articuladas de diferente manera, y al haber añadido de la misma forma las uniones entre ellas, un piso y una gravedad pudimos observar como el robot se caía por su misma inercia. Pudimos observar las variantes que podemos dar al robot, que van desde el tamaño de las piezas hasta el color y material de las mismas.

Cabe resaltar que se puede dar un movimiento al robot, ya que hasta el momento no hemos aprendido como manejar esas situaciones, debido a que necesitamos agregar un peso a las piezas de la misma forma que añadimos una fuerza con la cual se detendrá la pieza a manera de suspensión y con esto poder irle asignando distintas variables numéricas y obtener el movimiento deseado.

Tras ser la primera práctica, nos llevamos los principios básicos del uso de ros, utilizando la herramienta gazebo que nunca habíamos trabajado con ella ni su entorno. Creo que el aprendizaje mas significativo que nos llevamos de esta práctica es el saber utilizar las aplicaciones por primera vez, el saber el para qué y por qué de las cosas y como dar variantes para crear un entorno de trabajo propio y funcional.

Cesar:

al generar el entorno de trabajo para catkin se generaron errores con el entorno de trabajo para crear nuestros robots por lo cual se instalaron algunos programas que se necesitaron para poder crear nuestro entorno de miRobot para poder ejecutarlo.