

Laboratorio Virtual de Robótica Industrial.

Especificación de Recursos de Software

El desarrollo del laboratorio virtual de robótica industrial está basado en dos amplios pilares de recursos de software tales como las librerías que corresponden a ROS y las librerías de desarrollo de aplicaciones de Qt-Creator. Ambos poseen licencias open-source para su uso y oficialmente ambas son muy utilizadas para el desarrollo de sistemas robóticos bajo ROS.

1. ROS

Dentro del actual desarrollo del laboratorio se está utilizando la versión Full de ROS

Install >> ros-kinetic-desktop-full

Al usar esta versión se instalan librerías y aplicaciones para desarrollo y visualización de robots entre ellas destacan.

- RVIZ
- Gazebo
- URDF library
- SDF library

1.1 Modelación de robots.

ROS posee la capacidad de simular robots antes de pasar a un escenario real siendo estas una de las capacidades por la cual muchos roboticistas deciden utilizar ROS entre otros Frameworks o Middlewares.

URDF: Es una extensión de archivo ocupado para la modelación de robots, este lenguaje de marcado es parecido al lenguaje HTML solo que con variaciones las cuales, en vez de definir el marco estructural de una página web, este describe el marco de referencia de árbol de un robot.

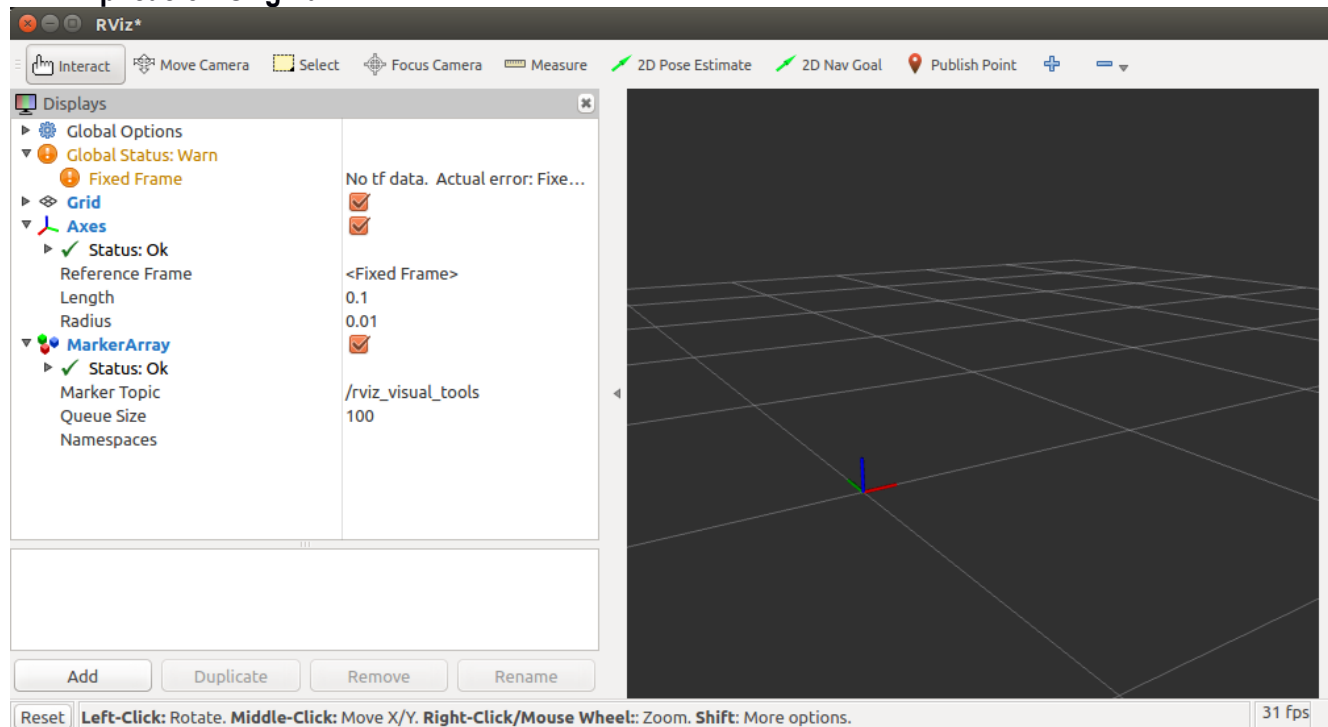
SDF: Es una extensión de archivo que es ocupado para la modelación de escenarios y modelación de robots dentro de un solo archivo. En este se puede definir muchas variables relacionadas con el escenario, tales como gravedad, inercia, fricción y entre otras variables que influyen en los cuerpos dentro de los escenarios virtuales.

Para Visualización de los robots en 3D se ocupa RVIZ y Gazebo.

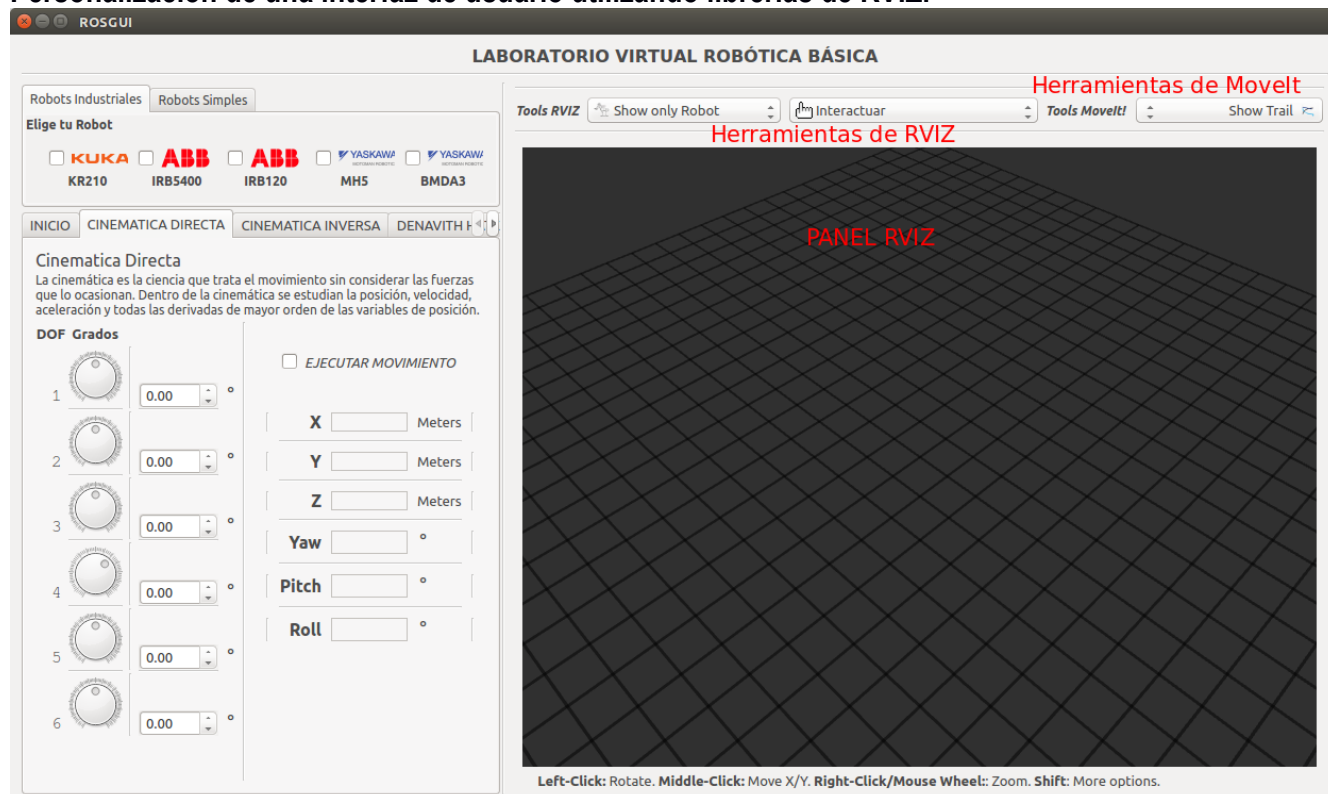
	URDF	SDF
Gazebo	x	x
RVIZ	x	

RVIZ: es una aplicación de visualización de robots la cual puede ser personalizada y modificada para un uso específico. Utilizando su código fuente.

RVIZ Aplicación Original



Personalización de una interfaz de usuario utilizando librerías de RVIZ.



Una vez obtenida la aplicación de visualización de los datos del robot, pueden cargarse a este escenario los modelos de robots con URDF en este caso para la aplicación RVIZ

Dentro de la Aplicación personalizada se cargan plugins de visualización para RVIZ tales como.

PLUGIN_RVIZ_ROBOT_MODEL

A este plugin se le define el robot a cargarse a la pantalla o al panel de RVIZ.

Para esto se carga vía parámetros de ROS el archivo URDF. De tal forma que si deseamos cambiar un robot u otro del panel solo basta con actualizar el parámetro del nodo ROS de la GUI.

Otro plugin que permite la visualización de Robots es el Plugin de MoveIt!.

PLUGIN_MOVEIT_ROBOT_MODEL

Este permite mostrar el robot generado por las librerías de planeación de MoveIt en el cual se puede mostrar las trayectorias de un robot. Cabe destacar que el uso de MoveIt permite agilizar el proceso de desarrollo de software aplicado a la Cinemática.

Todo esto resuelve la visualización de los modelos de robots industriales o de modelos simples de Robots.

1.2 Cinemática Directa

En la descripción del laboratorio virtual se ha mencionado la importancia de la cinemática para el análisis de un robot.

Para esto se decidió incorporar la librería KDL que permite el análisis de un robot a partir de un archivo URDF.

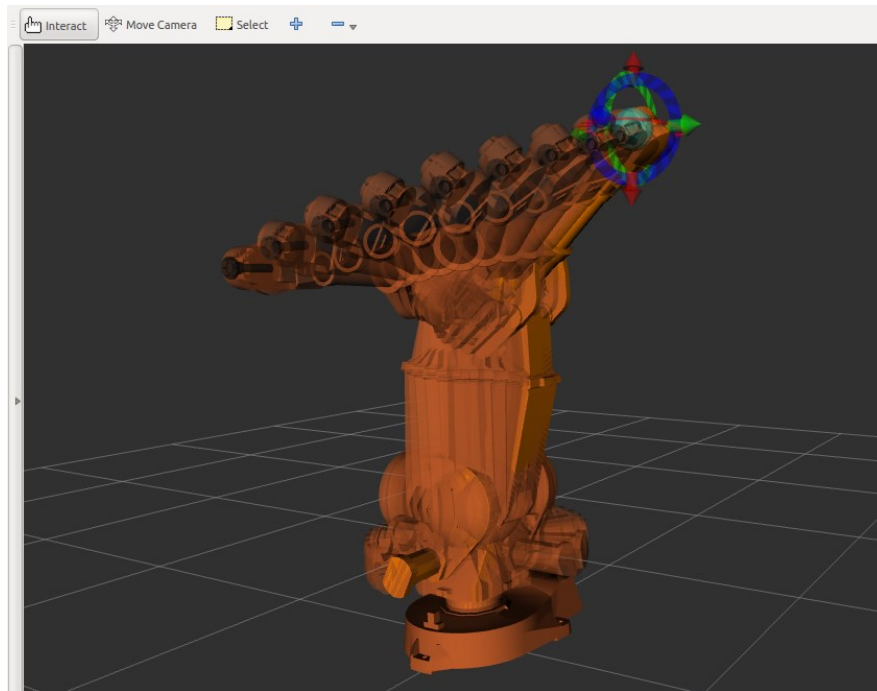
Conocer el valor del Efecto final de acuerdo a la posición actual de cada joint, conocer el grado de apertura de cada robot y conocer la matriz de transformación homogénea característica de cada robot.

Para esto se aplican diferentes algoritmos ya propios de la librería.

1.3 Cinemática Inversa

La cinemática inversa forma parte de lo propuesto en el trabajo monográfico pero este punto representa mayor complejidad al utilizar los algoritmos ya propuestos en la librería utilizada en la cinemática directa KDL. De tal forma que para este problema se ha elegido el uso de las librerías de MoveIt! Las cuales además de utilizar las librerías nativas de KDL incorporan una mejor solución al problema de la cinemática inversa, esto es así porque MoveIt! Es el producto de incorporar diferentes métodos del estado del arte de la planeación del movimiento de un Robot (Motion Planning).

Además, que utilizando MoveIt se puede utilizar el plugin de visualización del robot e incorporarlo dentro del panel personalizado de RVIZ mencionado anteriormente (**PLUGIN_MOVEIT_ROBOT_MODEL**) incorporando en esta visualización de los datos del modelo del robot en 3D como la trayectoria y otros.



1.3 Espacio de Trabajo del robot (workspace)

Para generar el espacio de trabajo característico de cada robot se utilizara la librería Releaux que posee un plugin para generar el espacio de trabajo dentro de la pantalla o panel de RVIZ

Plugin_workspace_visualization/ReachabilityMap

Para esto también se actualiza el modelo del robot desde los parámetros del servidor de ROS, De tal forma que si deseamos cambiar un robot u otro del panel solo basta con actualizar el parámetro del nodo ROS de la GUI.

1.4 Simulaciones con Gazebo

Gazebo simulador permite cargar un escenario desde el archivo SDF, es diferente la forma de manipular un escenario desde Gazebo comparado con RVIZ, debido a que todas las visualizaciones de los modelos dentro de RVIZ son de carácter estático es decir que no se puede ver sólidos dinámicos (Dentro de Gazebo se puede simular una Un tornillo cayendo desde una mesa) esto sin necesidad de programación previa, desde el momento que cargue todos los modelos al escenario de Gazebo estos rebotan, balancean y se vuelven dinámicos.

Por lo tanto si se desea obtener estas capacidades de “realismo” es necesario utilizar gazebo.

Para el control de los robots o escenarios de Gazebo se hace mediante plugins propios de Gazebo que permite extraer los datos del escenario procesarlos y mandar ordenes o comandos para que estos comiencen a interactuar con el escenario.

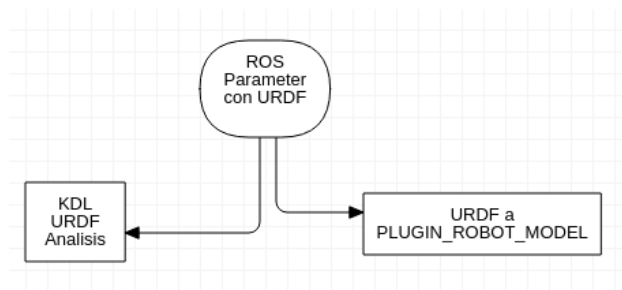
1.5 Nodos de ROS.

Se entiende que con ROS es necesario el amplio uso de los nodos en un sistema robótico para compartir la información.

Se pudo observar que en la interfaz personalizada de RVIZ casi no se hace mucha mención de estos.

1.5.1 La interfaz personalizada: es un nodo que se registra al ROSMaster pero que no comparte información ni recibe información de otros nodos. La única forma que interactúa nuestra interfaz con ROS es mediante el servicio de Parámetros de ROS.

Los Parámetros pueden ser información que está disponible para muchos nodos o simplemente para un solo nodo como es el caso del desarrollo de nuestra interfaz. Estos permiten registrar en el ROSMaster cierta información estática tal como un URDF luego esa información es ocupada para diferentes procesos dentro del mismo nodo que requieren la misma información, permitiendo registrar y eliminar el registro cuando sea necesario (Actualizar nuevo modelo de Robot).



1.5.2 Nodos con Gazebo: Con Gazebo es diferente el proceso de desarrollo ya que de forma general se puede ver a Gazebo como tener el modelo de un robot con sus escenarios y otros recursos, pero de carácter virtual. Para esto es necesario abrir nodos que mande información de ROS hacia el modelo virtual del robot y nodos que además escuchen lo que está pasando en el escenario de Gazebo (Sensores virtuales de Gazebo)

1.5.3 Nodo Con Joystick: ROS posee un conjunto de Drivers para conectar periféricos a una aplicación, para esto se lanza un nodo de ROS que permite comunicarse bajo los estándares definidos por el protocolo que soporta el periférico. En la mayoría de los casos con los controles genéricos y otros la comunicación es usando USB HID. Estos datos que escucha nuestro nodo serán utilizados para ejecutar movimientos desde el Joystick al Robot.

2. IDE Qt-Creator.

Qt es un framework multiplataforma orientado a objetos para el desarrollo de software en escritorio, sistemas móviles y sistemas embebidos. Las plataformas soportadas por Qt incluyen Linux, Windows, Android, iOS y otros.

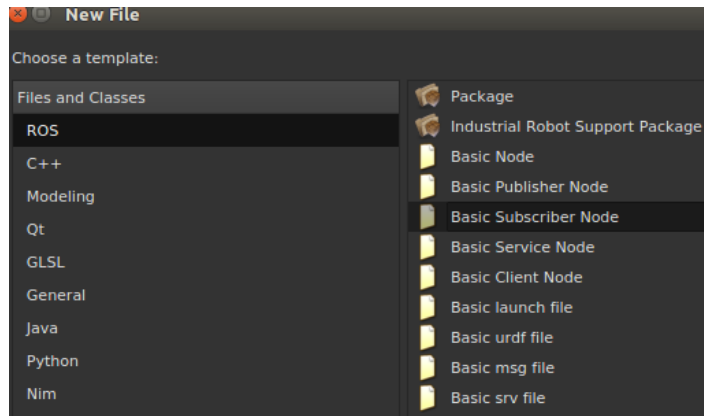
Qt no es un lenguaje de programación por sí solo, es un framework escrito en C++ (Conjunto de librerías y convenciones).

2.1 QT Con ROS

Uno de las mayores ventajas de Qt Creator es que permite que un equipo de desarrolladores comparta un proyecto a través de diferentes plataformas de desarrollo (Microsoft Windows, Mac OS X, y Linux) con una herramienta común para desarrollo y depurado.

Para la programación de los nodos de ROS es necesario utilizar C++ o python. Actualmente El IDE de Qt-Creator es el oficial utilizado por desarrolladores ROS.

Permite compilar con Cmake que es el compilador oficial de ROS, ejecutar nodos desde el IDE y Posee plantillas de creación de Nodos, Launch o URDF.



Utilizando QT-Creator se pueden crear simples nodos que pueden ser meramente nativos bajo ROS es decir solo utilizando librerías de ROS, esto se puede extender utilizando las librerías de Qt-Creator y crear procesos que lancen un nodo de ROS que “Maten” un proceso de ROS entre otras funcionalidades, todo esto se podría diseñar bajo una interfaz de usuario amigable con botones, textos, ventanas etc.

```
#include "ros/ros.h"
#include "ros/callback_queue.h"
#include "ros/subscribe_options.h"
#include "std_msgs/String.h"
```

```
#include <QMDiArea>
#include <QtGui>
#include <QMainWindow>
#include <ros/ros.h>
#include <kdl_parser/kdl_parser.hpp>
```

Captura de pantalla con las librerías de ROS y Librerías de ROS y QT-Creator

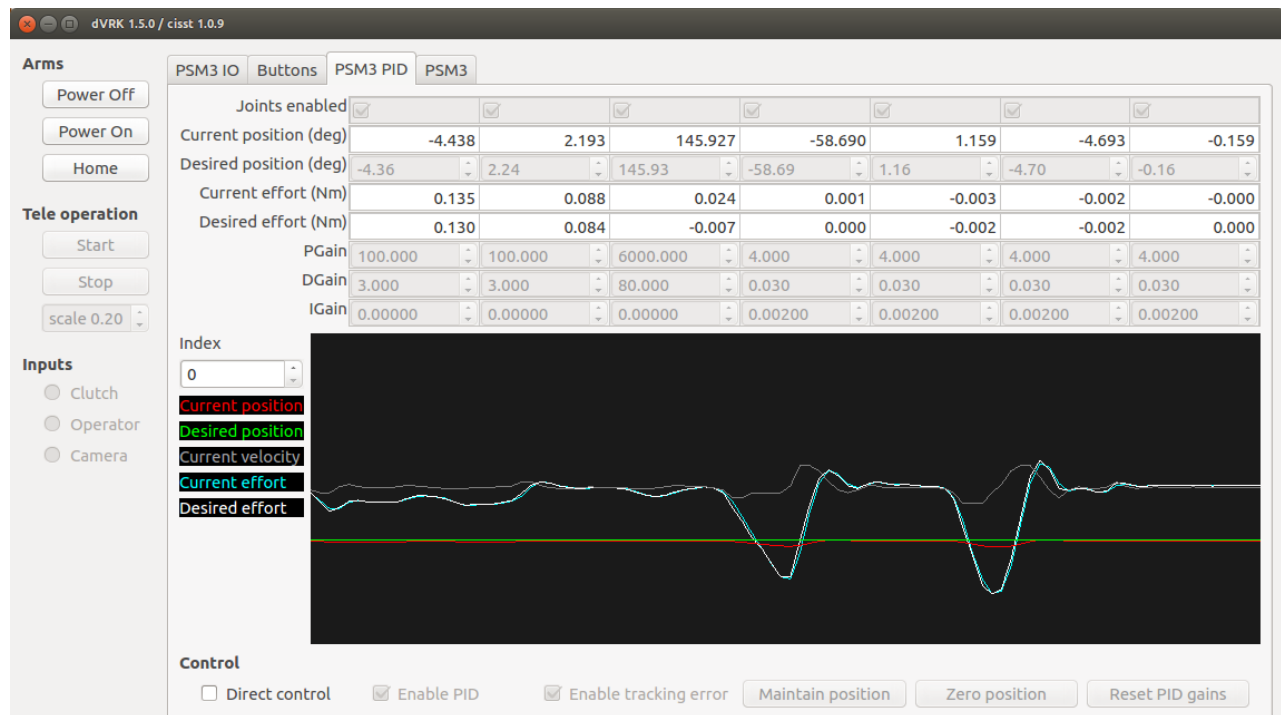
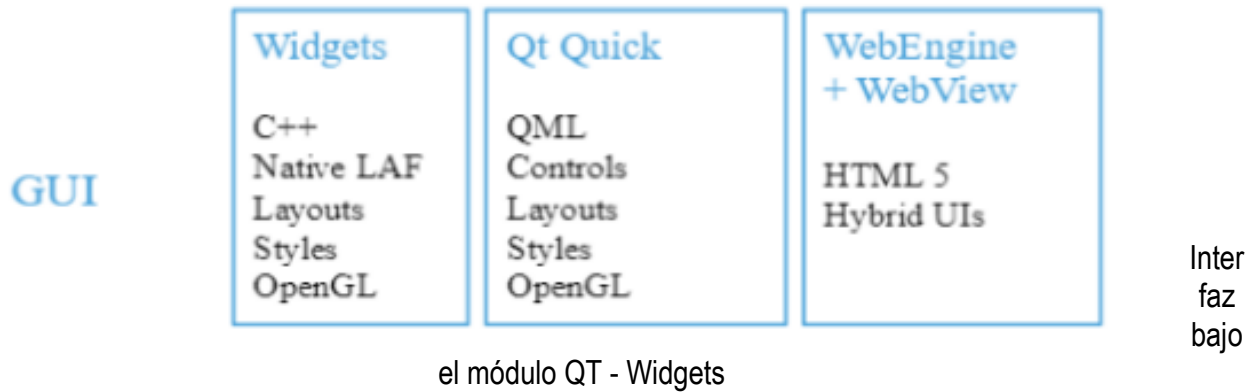
Incluyendo Qt-Creator se puede tener un botón en una GUI y cada vez que este sea presionado mande información a otro nodo de ROS, como se muestra en la imagen siguiente.

```
MainDialog::MainDialog(QWidget *parent): QDialog(parent){
    this->bAvanzar = new QPushButton(this);
    connect(bAvanzar, SIGNAL(pressed()), this, SLOT(avanzar()));

void MainDialog::avanzar(){
    enviarRos("e /home/ray/robot/src/miRobot/programas/avanzar");
}
```

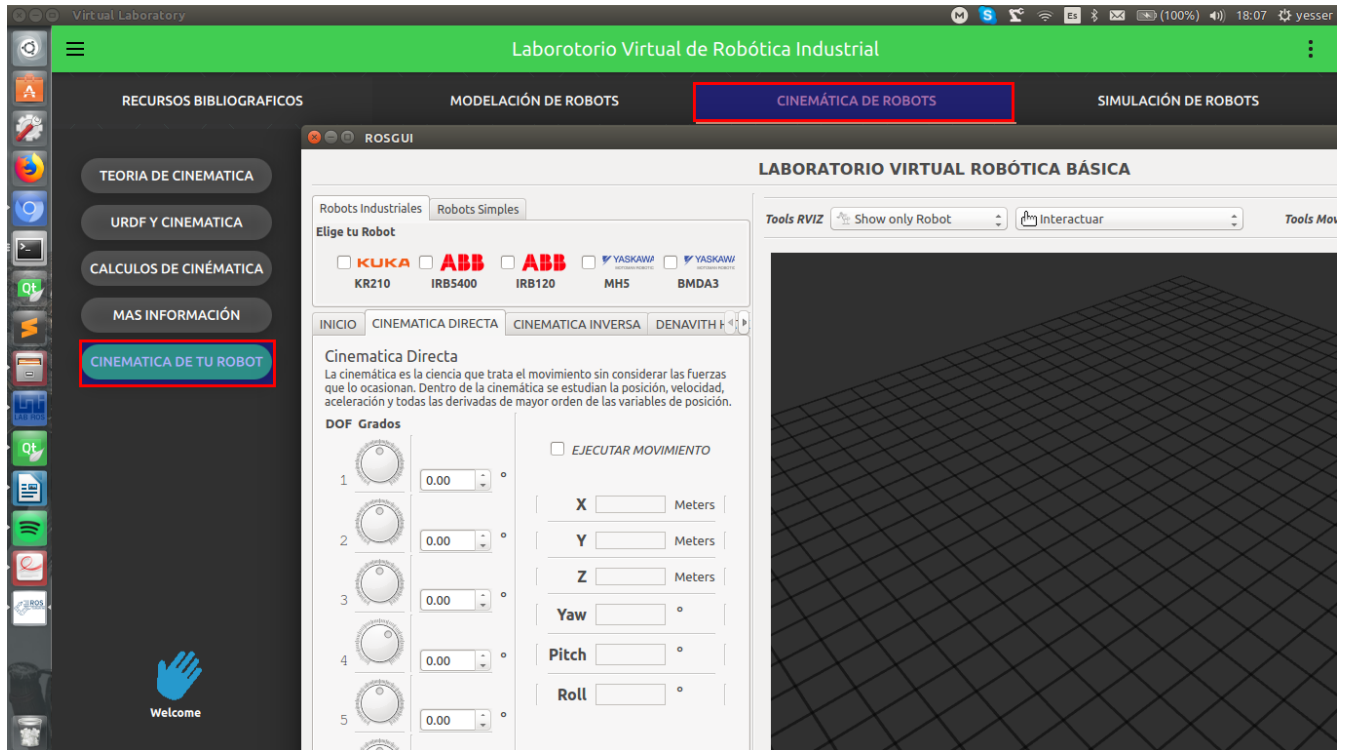
2.2 QWidgets

En la interfaz personalizada de RVIZ para mostrar las prácticas de cinemática directa e inversa. Se utiliza uno de los módulos de QT-Creator denominado Qwidgets. En esta se pueden encontrar botones, sliders, Combobox, dialogos, TabBars, PlotXY, etc. Las cuales forman parte del diseño de la interfaz para mostrar información o enviar información desde una aplicación amigable.



2.3 QML

La interfaz de inicio principal del laboratorio virtual de Robótica industrial está diseñada bajo el módulo de QML de Qt, esto debido a que cuenta con animaciones fluidas, mayores opciones de recursos para la aplicación debido que con QT se puede agregar otros lenguajes de programación como JavaScript, Programación WEB y otros recursos que hacen de QML una opción bastante robusta.



La anterior captura de muestra la pantalla de inicio que permitirá lanzar otros nodos con aplicaciones de **Modela tu Robot**, **Cinemática de tu robot** y **Simulaciones de Gazebo**, la interfaz cuenta con recursos Web (QTWebEngine) que son ocupados para mostrar PDF dentro de la interfaz y también mostrar el escenario WEB del simulador Gazebo.

Actualmente se esta trabajando en la incorporación del simulador gazebo Web dentro de la interfaz QML esto se logra por el uso de HTML, CSS y JavaScript. Con HTML se describe cada uno de los ítem, con CSS se agrega forma y mejoras en la vista de los ítems y con javascript se agrega la lógica de programación de la pagina WEB.

Esta pagina Web se carga con el motor de QTWebEngine a uno de los ítems de QML dentro de la aplicación.