

## Práctico 3.b - ROS2.

1. Instalar el proyecto Francesca desde el repositorio de gitlab:  
[https://gitlab.fing.edu.uy/jvisca/francesca\\_ws](https://gitlab.fing.edu.uy/jvisca/francesca_ws)  
Para acceder a el deberá estar logueado con el usuario de Facultad.
2. Levantar el robot en modo simulador (ver el README.md del proyecto), y aplicar lo aprendido en el práctico 3.a
3. Responder las siguientes preguntas:
  - a. Haga que el robot se mueva en un círculos usando el tópico `/cmd_vel` desde la línea de comando. ¿El movimiento simulado se parece a lo que esperaban?
  - b. Mueva el robot usando el Nodo de control teclado. ¿En qué difieren las perspectivas dadas por Gazebo y RViz2?
  - c. Mientras el robot se mueve, usar RViz2 para visualizar los Links correspondientes a las ruedas. ¿Están de acuerdo con el movimiento simulado resultante? ¿Las ruedas “patinan”?
  - d. ¿Cómo debe invocar la simulación para que se activen el Lidar y la cámara simulados?
  - e. Genere el árbol *tf* completo (con los sensores adicionales) de Francesca usando

```
ros2 run tf2_tools view_frames
```

- f. ¿Qué tópicos generan el Lidar y la cámara? Visualízenlos en RViz2.
- g. Identifique en el modelo URDF del robot dónde están definidas las ubicaciones de ambos sensores, e intente moverlos. Por ejemplo, haga que la cámara apunte a un lado y no al frente.
- h. Levante el módulo de odometría visual basada en el Lidar utilizando

```
ros2 run rtabmap_odom icp_odometry --ros-args -r odom:=odom_icp
```

- Compárela con la odometría basada en los encoders de las ruedas. ¿Cuál parece más confiable en sus pruebas?
- i. Edite los scripts de launch de francesca para agregar la opción de levantar el nodo de odometría visual, de forma similar a como se levantan el Lidar o la cámara.
  - j. Edite el mundo usado en el simulador para agregar algunos objetos más.