

## HOJA DE COMANDOS ROS2:

### } 1. COMANDOS BÁSICOS

**ros2:** Comprueba que ya están activos tanto el **underlay** (/opt/ros/humble) como el **overlay** (~/**<my\_workspace>**).

**ros2 (estructura):** **ros2 <command> <verb> [ <params> | <option> ] \***.

**ros2 pkg list:** Muestra la lista de todos los paquetes disponibles.

**ros2 pkg executables <package>:** Muestra los ejecutables de un paquete específico.

**ros2 run <paquete> <executable>:** Ejecuta un programa de un paquete específico.

**ros2 launch <my\_package> <executable>:** Ejecuta un programa de un paquete específico mediante launcher.

**ros2 node list:** Lista todos los nodos que se están ejecutando en ese momento.

**ros2 topic list:** Lista todos los topics que se están utilizando en ese momento.

**ros2 node info <node>:** Lista más información acerca de un nodo específico.

**ros2 topic info <topic>:** Lista más información acerca de un topic específico (tipo de mensaje, número de publicadores, número de suscriptores).

**ros2 topic info --verbose <topic>:** Muestra la información de un topic específico pero mostrando información de las QoS.

**ros2 topic pub -r <freq> <topic> <message\_type> <message>:** Publica en un topic un mensaje específico a una frecuencia específica.

**ros2 interface list:** Lista todos los tipos de mensajes válidos en el sistema.

**ros2 interface show <message>:** Lista el formato de mensaje y los campos que puede recibir.

**ros2 topic echo <topic>:** Lista los mensajes publicados de un topic específico, con la opción **--no-arr** no se muestra el contenido de los arrays de datos.

**ros2 run rqt\_graph rqt\_graph:** Muestra el grafo computacional de aquello que esté ejecutándose.

**ros2 run rqt\_console rqt\_console:** Herramientas para ver los mensajes publicados en un topic.

### } 2. COMANDOS DE EJECUCIÓN, LAUNCHERS Y TESTS

**ros2 run <my\_package> <executable> --ros-args --log-level <level>:** Configura el log para establecer otro nivel mínimo de severidad al mostrarse en la salida estándar.

**ros2 run <package> <executable> --ros-args -p <variable>:=<value>:** Ejecuta el programa estableciendo un valor específico a cada parámetro.

**ros2 run <package> <executable> --ros-args --params-file**

**install/<my\_package>/share/<my\_package>/config/params.yaml:** Ejecuta el programa pasándole como argumento el fichero que contiene todos los parámetros.

**ros2 run teleop\_twist\_keyboard teleop\_twist\_keyboard --ros-args -r cmd\_vel:=key\_vel:** Lanzar teleoperador.

**ros2 run rviz2 rviz2:** Lanzar rviz2 (muestra información geométrica y sensorial).

**ros2 run br2\_fsm\_bumpgo\_cpp bumper --ros-args -r output\_vel:=/nav\_vel -r input\_scan:=/scan\_raw -p use\_sim\_time:=true:** Ejecución del programa de la máquina de estados, con **use\_sim\_time:=true** toma el tiempo del topic /clock, correspondiente al tiempo del simulador en vez de el de nuestra máquina.

**ros2 launch br2\_tiago sim.launch.py world:=<world>:** Lanza el simulador en un mundo específico.

**ros2 launch br2\_navigation tiago\_navigation.launch.py:** Lanza la navegación.

**ros2 launch nav2\_bringup tb3\_simulation.launch.py:** Lanza la navegación, donde el paquete **nav2\_bringup** se encuentra en **/opt/ros/humble/share/nav2\_bringup** que incluye launchers, mapas y parámetros para la simulación con el TurtleBot3 (por defecto).

### } 3. COMANDOS DE LIFECYCLE NODES

**ros2 lifecycle nodes:** Muestra qué nodos de control de vida (LifeCycle) se están ejecutando en ese momento.

**ros2 lifecycle get <node>:** Comprueba en qué estado se encuentra un nodo específico.

**ros2 lifecycle list <node>:** Muestra las transiciones disponibles en el estado actual.

**ros2 lifecycle set <node> activate:** Activa un nodo específico.

**ros2 lifecycle set <node> deactivate:** Desactiva un nodo específico.

**ros2 run groot Groot:** Herramienta para crear y desarrollar Behavior Trees.

**build/<my\_package>/tests/<node> --ros-args -r input\_scan:=/scan\_raw -r output\_vel:=/key\_vel -p**

**use\_sim\_time:=true:** Testeo de un nodo de un BT.