

# Introducción a ROS con Deep Learning Robot

Diego Cabrera Mendieta

<sup>1</sup>Departamento de Ingeniería Mecánica  
Universidad Politécnica Salesiana

<sup>2</sup>Departamento de Ciencias de la Computación e Inteligencia Artificial  
Universidad de Sevilla

ROS-DL Robot, 2016

- 1 Introducción a ROS
- 2 Deep Learning Robot y ROS
- 3 Ejemplos de aplicación
- 4 Conclusiones

## Generalidades:

- Código abierto.
- Meta-SO.
- Similar a otros framework para robots, Player, YARP, Orocos, etc.
- En tiempo de ejecución ROS genera un grafo de procesos.
- Puede distribuirse en varias máquinas.
- No es un framework para tiempo real.
- válido para plataformas Unix.

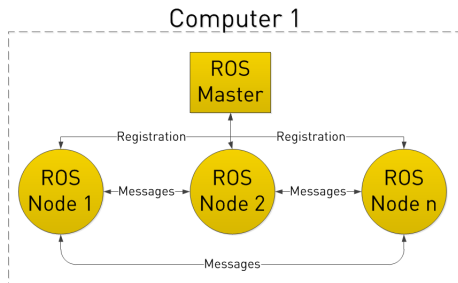


Figure : ROS en tiempo de ejecución

## Procesamiento de imágenes en un robot + visualización remota.

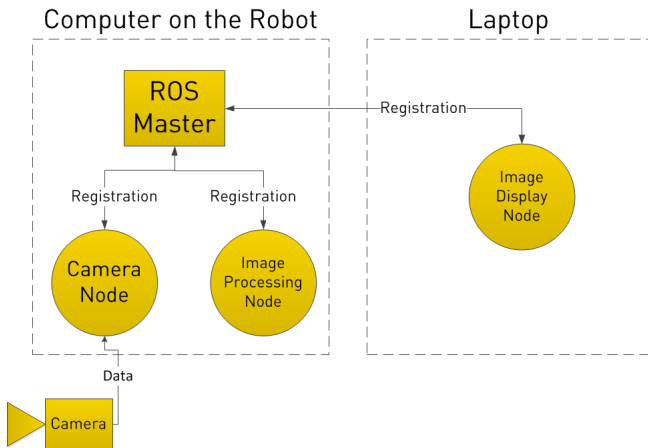


Figure : Suscripción de los nodos en el Maestro.

# Ejemplo con ROS

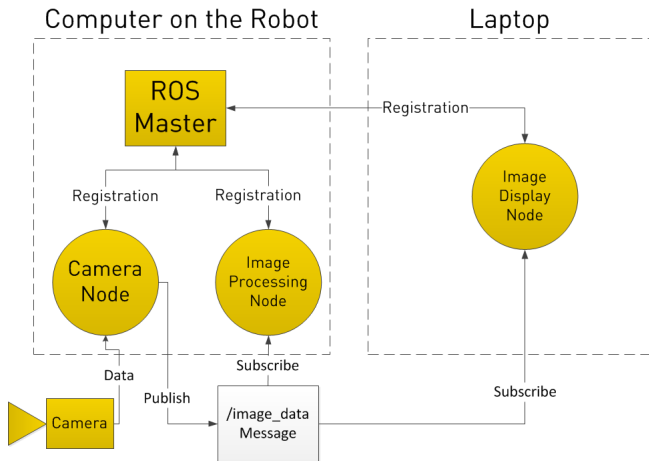


Figure : Publicación y Suscripción en un tema.

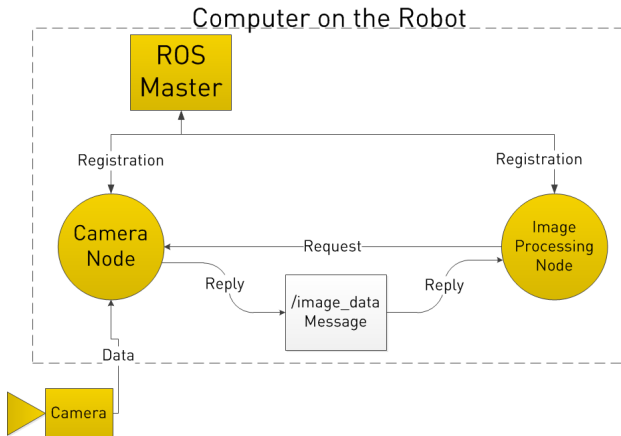


Figure : Petición por servicio.

# Deep Learning Robot

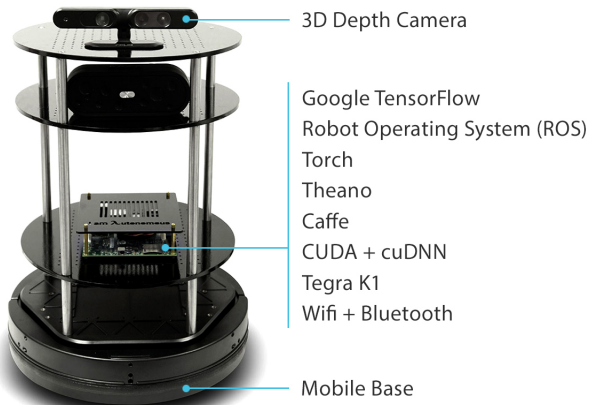


Figure : Partes del DL-Robot.

# Configuración de comunicación

- 1 Instalar ROS en la estación de trabajo.
- 2 Ingresar la estación de trabajo y el DL-Robot a una misma red.
- 3 Acceder a DL-Robot por ssh.
- 4 En DL-Robot agregar a `.bashrc`:  

```
ROS_MASTER_URI=http://localhost:11311  
ROS_HOSTNAME=IP_OF_DLBOT
```
- 5 En la estación de trabajo agregar a `.bashrc`:  

```
ROS_MASTER_URI=http://IP_OF_DLBOT:11311  
ROS_HOSTNAME=IP_OF_WS
```
- 6 Reiniciar la sesión ssh y la consola en la estación de trabajo.



- 1 En DL-Robot ejecutar:  
`roslaunch turtlebot_bringup minimal.launch`
- 2 En la estación de trabajo ejecutar:  
`roslaunch turtlebot_teleop keyboard_teleop.launch`
- 3 Conducir el DL-Robot.

- 1 En DL-Robot ejecutar:

```
roslaunch turtlebot_bringup minimal.launch  
roslaunch oppenni2_launch oppenni2.launch
```

- 2 En la estación de trabajo ejecutar:

```
roslaunch image_view image_view image:=/camera/rgb/image_raw
```

- 3 En la estación de trabajo ejecutar:

```
roslaunch image_view image_view image:=/camera/depth/image
```

- 4 En DL-Robot detener oppenni (Ctrl-C) y ejecutar:

```
roslaunch turtlebot_bringup 3dsensor.launch
```

- 5 En la estación de trabajo cerrar todo y nuevamente ejecutar:

```
roslaunch turtlebot_rviz_launchers view_robot.launch
```

- 6 En RViz bajo Display activar “Registered DepthCloud” y “Image”:

# Crear un nodo en python

- 1 En DL-Robot cerrar todo y ejecutar:  
`roslaunch turtlebot_bringup minimal.launch`
- 2 En la estación de trabajo crear el script "goforward.py" con su respectivo código:
- 3 En la estación de trabajo ejecutar:  
`python goforward.py`
- 4 En la estación de trabajo crear el script "goincircle.py" con su respectivo código:
- 5 En la estación de trabajo ejecutar:  
`python goincircle.py`
- 6 En la estación de trabajo ejecutar:  
`python draw_a_square.py`

# Construir un mapa

- 1 En DL-Robot cerrar todo y ejecutar:

```
roslaunch turtlebot_bringup minimal.launch  
roslaunch turtlebot_navigation gmapping_demo.launch
```

- 2 En la estación de trabajo ejecutar:

```
roslaunch turtlebot_rviz_launchers view_navigation.launch  
roslaunch turtlebot_teleop keyboard_teleop.launch
```

- 3 En DL-Robot cerrar ejecutar:

```
roslaunch map_server map_saver -f /tmp/my_map
```

- 1 En DL-Robot cerrar todo y ejecutar:

```
roslaunch turtlebot_bringup minimal.launch  
roslaunch turtlebot_navigation amcl_demo.launch map_file:=/
```

- 2 En la estación de trabajo ejecutar:

```
roslaunch turtlebot_rviz_launchers view_navigation.launch
```

- 3 En RViz con la herramienta "2D Pose Estimate" dar clic y arrastrar para ubicar la posición actual y la orientación del robot.
- 4 En RViz con la herramienta "2D Nav Goal" dar clic y arrastrar para ubicar la posición final y la orientación deseada del robot luego del movimiento.

# Moverse a una posición por código

- 1 No modificar nada del ejemplo anterior.
- 2 En la estación de trabajo crear el archivo `"go_to_specific_point_on_map.py"` con su código.
- 3 Modificar la línea 45 con la coordenada a la que deseamos que se dirija el robot.
- 4 Ejecutar `"go_to_specific_point_on_map.py"`.

# Conclusiones

# Gracias