

# Qbo + ROS

Ariel Mora

CI-2657 - Robótica

Universidad de Costa Rica

# Agenda

- Qbo
- Qbo: Ejemplos
- ROS
- ROS: Caso de uso
- ROS: Funcionamiento
- Qbo + ROS
- Qbo + ROS: Aplicaciones
- Qbo + ROS: Material

# Qbo

- Robot de propósito general de bajo costo
- Plataforma para el desarrollo de Robótica e Inteligencia Artificial
- Open Source
- Modificable en hardware como en software



# Qbo: Especificaciones

## Hardware:

- Intel Core i3-2120T (2.6 GHz)
- 4 GB DDR3
- 128 GB SSD
- Intel 6200 WiFi Adapter

## Software:

- SO: Open-Qbo

## Actuadores - Sensores:

- 2 Webcams
- 2 Ultrasónicos
- 1 Infrarojo
- 3 Micrófonos
- 2 Parlantes
- 3 Llantas -> 2 motores DC
- Servos
- ASUS Xtion (Kinect)

# Qbo: Características

- Visión Estereoscópica
- Reconocimiento y rastreo de caras
- Reconocimiento de gestos
- Reconocimiento y rastreo de objetos
- Reconocimiento y síntesis de palabras
- Mapeado de objetos
- Detección de objetos

## Qbo: Auto reconocimiento



# Qbo: Conversación



# ROS

- Robot Operating System
- Framework
  - Bibliotecas
  - Herramientas
  - Convenciones
- ¿Por qué usar ROS?
  - Complejidad computacional
  - Reutilización de software
  - Desarrollo colaborativo



Open Source Robotics Foundation

# ROS: “Arcos-Bot”



Nota: Controlador equivale a Computadora (PC)

- 1 Controlador: Base omnidireccional
- 2 Controladores: Brazos robóticos
- 2 Controladores: Manos robóticas
- 1 Controlador: Cabeza
- Kinect: Ojos
- Sensores: infrarrojos, ultrasónicos, cámaras, ...

6 Controladores + n Sensores + Controladores Principales

Diferentes ciclos de ejecución

# ROS: “Arcos-Bot”



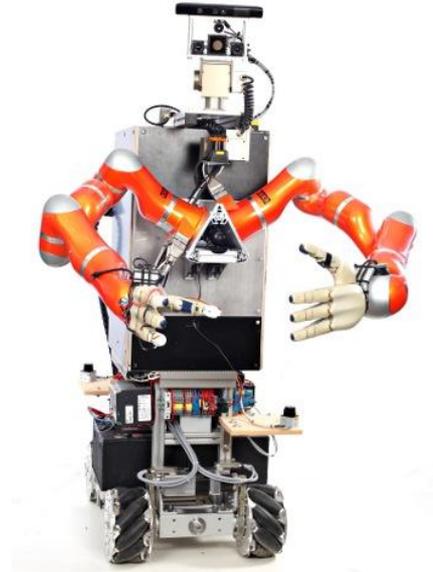
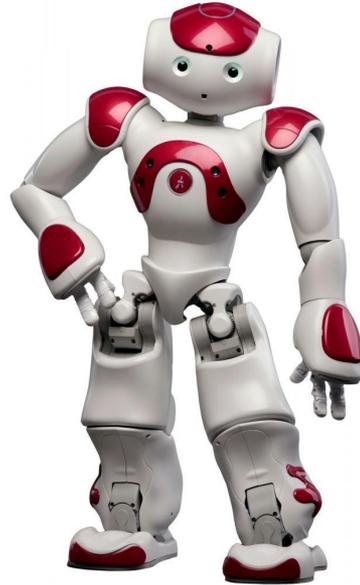
- Reconocimiento y detección de objetos
- Planeamiento de rutas
- Toma de decisiones
- Simulación

## ROS

- Comunicación entre controladores
- Reutilización de algoritmos
- Abstracción de hardware
- Ejecución en tiempo real

# ROS: Robots

- Aldebaran Nao
- Tum Rosie



# ROS: Robots

- iRobot Roomba
  
- Lego NXT



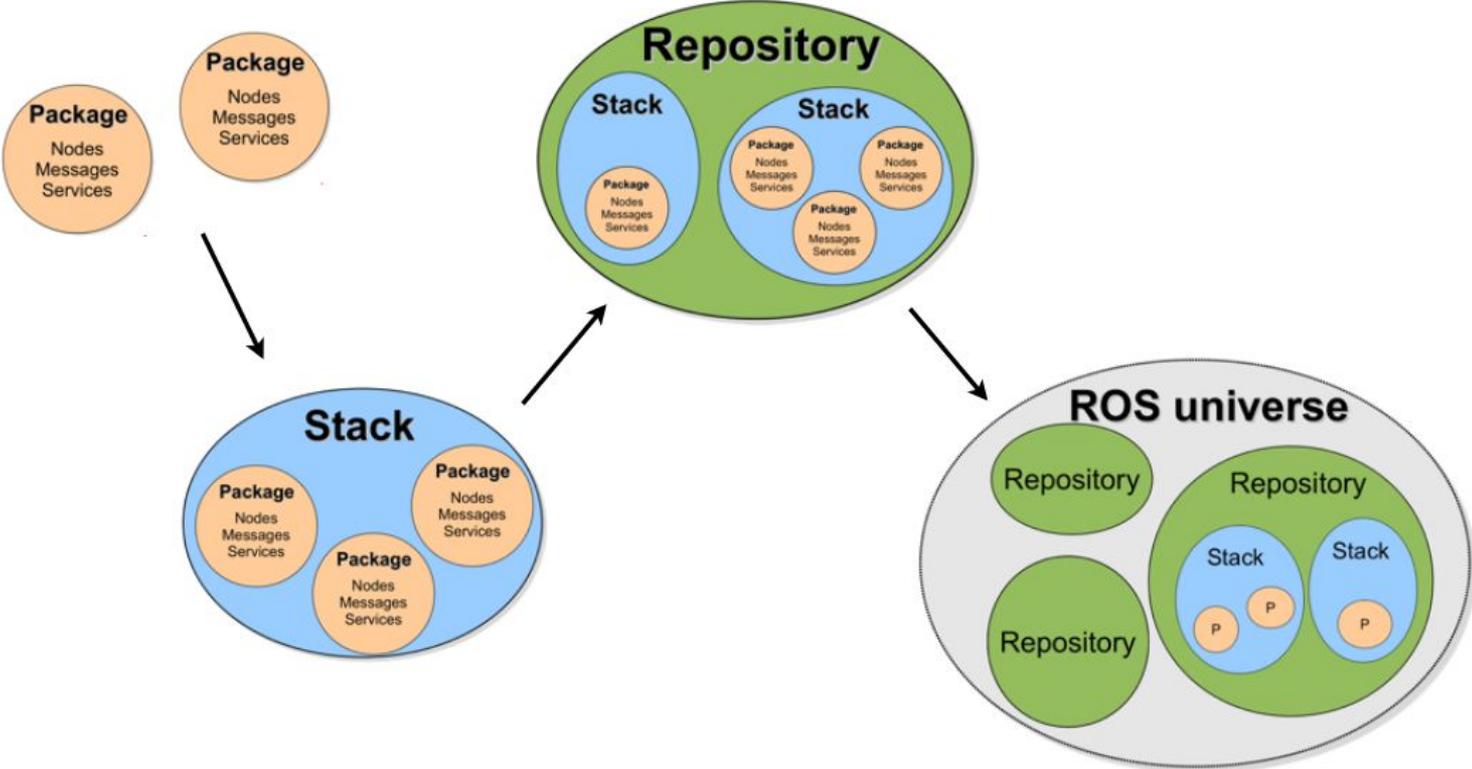
# ROS: Funcionamiento

- Paquetes
- Nodos
- Topics
- Servicios

# ROS: Funcionamiento

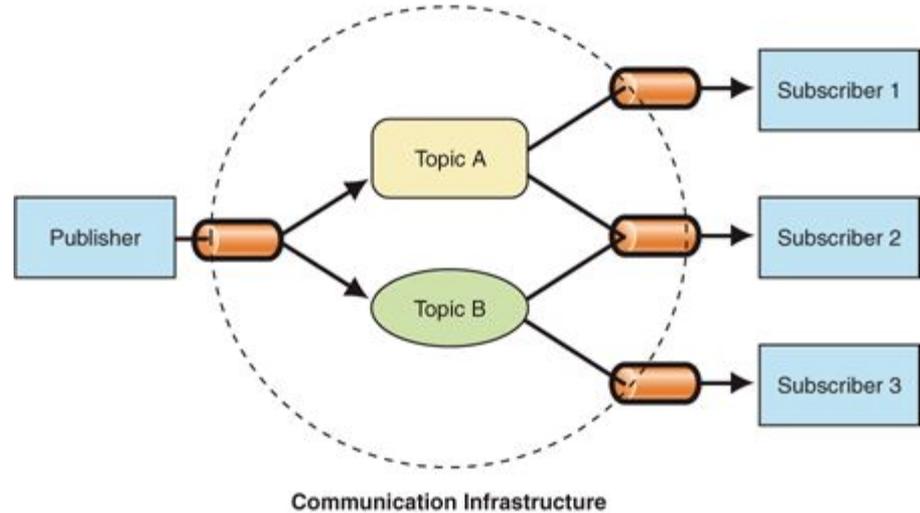
- Paquetes
  - Bloque de construcción
  - Reusable
    - Complejo para ser útil
    - Simple para ser usado
  - Contiene
    - Nodos
    - Topics
    - Servicios

# ROS: Funcionamiento



# ROS: Funcionamiento

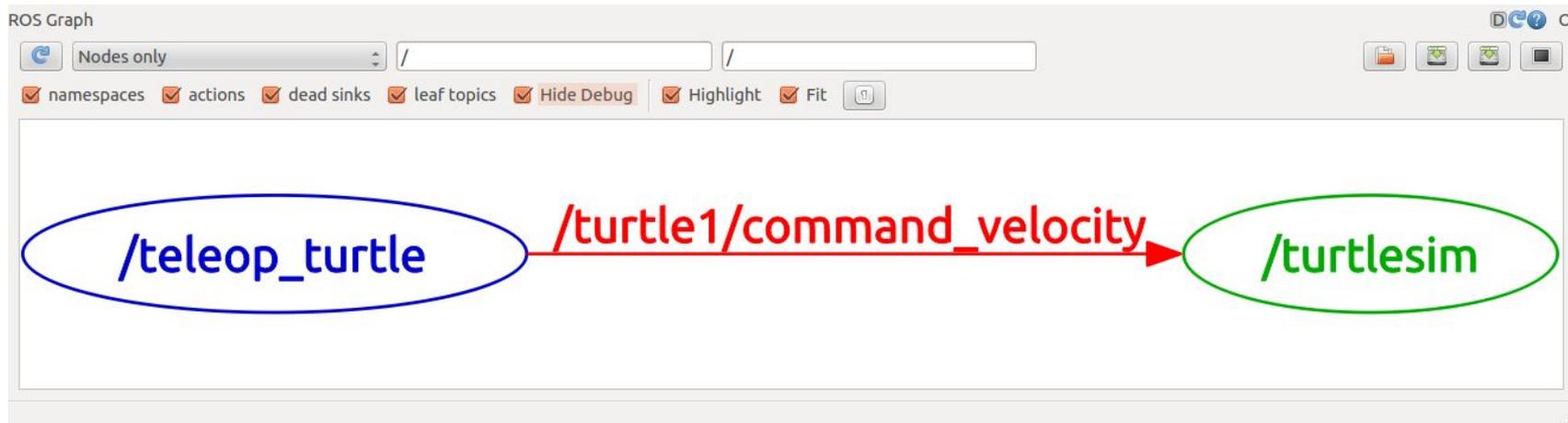
- Patrón Publish - Subscribe
  - Patrón de mensajes
  - Publisher
  - Subscriber(s)
  - Topic



# ROS: Funcionamiento

- Nodos
  - Unidades -> Funcionalidad específica
  - Tipo:
    - Publisher
    - Subscriber
  - Pueden ser ambos
- Topics
  - Mensajes

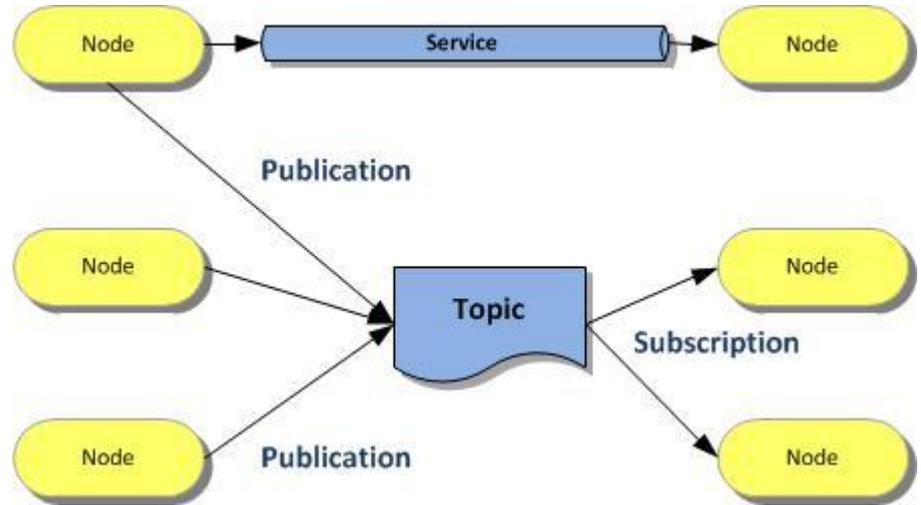
# ROS: Funcionamiento



- Nodos
  - Publisher (Azul)
  - Subscriber (Verde)
  - Topic (Rojo)

# ROS: Funcionamiento

- Services
  - Comunicación bidireccional
  - Síncrona
  - Uno a uno



# Qbo + ROS

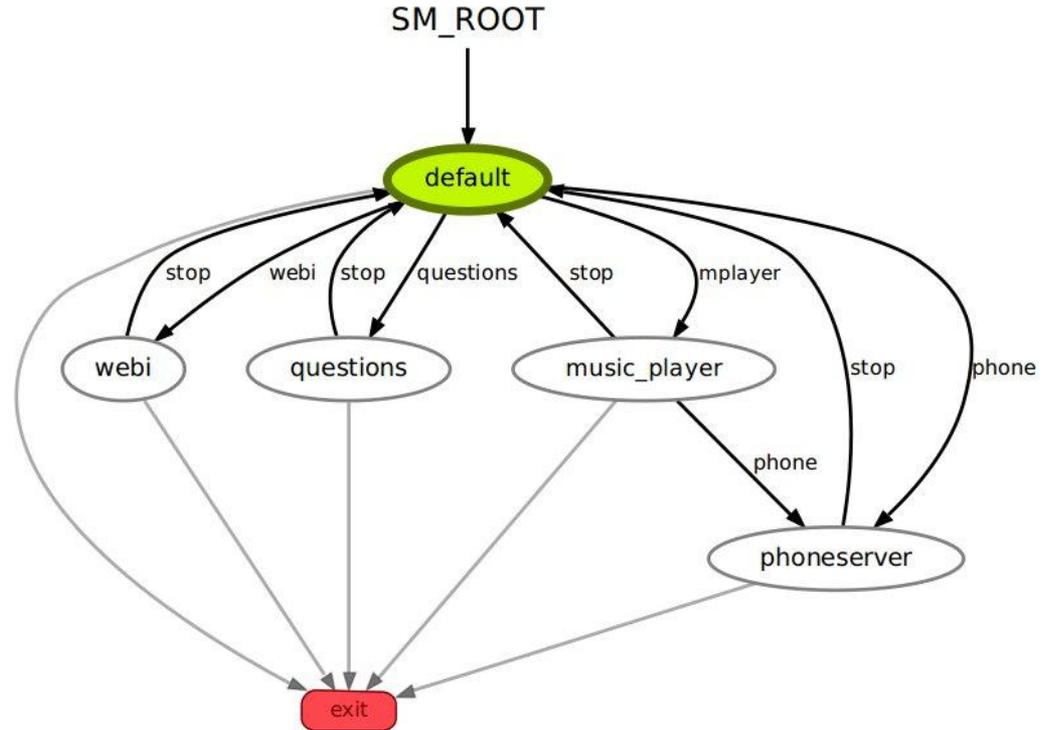
- Qbo: ROS instalado
- Modularización en Nodos
- Nodos
  - qbo\_face\_tracking: detecta y rastrea un rostro
  - qbo\_face\_following: mueve la cabeza y la base hacia el rostro detectado
  - qbo\_face\_recognition: aprende y reconoce rostros
  - qbo\_object\_recognition: aprende y reconoce objetos
  - qbo\_listen: reconoce palabras y oraciones
  - qbo\_talk: síntesis de voz

# Qbo + ROS: Aplicaciones

- qbo\_questions
  - ejemplos de ROS
  - varios módulos
  - inter-relación entre módulos

# Qbo + ROS: Aplicaciones

- qbo\_brain
  - intermediario para servicios



# Qbo + ROS: Material

- O’Kane, J. [A Gentle Introduction to ROS](#).
- [ROS Tutorials](#).
  - Python: [Rospy](#).
  - C/C++: [Roscpp](#).
- [OpenQbo](#).
  - [Tutorial: Primera aplicación](#).
  - [Qbo Apps](#).