

# Robótica Computacional

Introdução a ROS 2

# Exemplos de Robôs

## UVD Robot

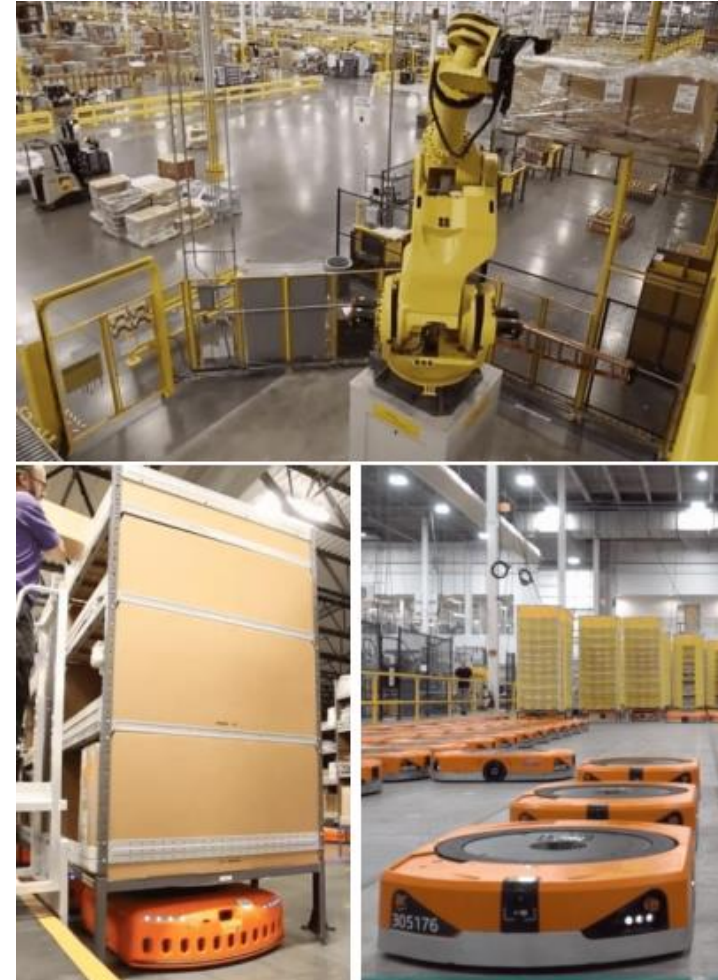
- Luz ultravioleta
  - Câmera 3D
    - Localização
- Sensores de profundidade
  - Detector de obstáculos
- Comunicação wireless
- Inteligência
  - Precisa saber onde o robô já visitou
  - Evitar pessoas!



# Exemplos de Robôs

## Amazon Robotics Fulfillment Center

- Robôs
  - Fanuc M-2000iA
  - Robotic Drive Unit
- Sensores
  - Lidar (2D / 3D)
  - Câmera RGB
  - Distância (Ultrasom ou IR)
- Inteligência
  - Deve entregar a “pod” do ponto A até o ponto B
  - Evitar muitos outros robôs





# Mas e se o ambiente não for estruturado?



# Competição da DARPA Subterranean Challenge

## Objetivo

- Navegar em um ambiente perigoso
- Encontrar objetos de interesse
- Resgate
- Gás
- Artefatos

## Como vencer uma competição dessas?

- Gerar mapas do ambiente
- Manter localização do robô
- Detectar objetos de interesse e mapear sua localização
- Coordenar diversos tipos de robô



# Vencedor: Team Cerberus

## Robôs:

- Spot (payload)
  - four-legged
  - 360o câmera
  - LIDAR
  - Microphone
- Seekur Jr: (mapeamento)
  - six-wheeled
  - LIDAR
  - GPS
- DroneSense (exploração)
  - 360o câmera
  - LIDAR
  - Detector de gás
- Flyability Gimball (exploração)
  - 360o câmera
  - LIDAR

**Como administrar essas informações?**





# ROS - Robot Operating System

“open-source robotics middleware platform that provides a set of software libraries and tools for building robot applications.”

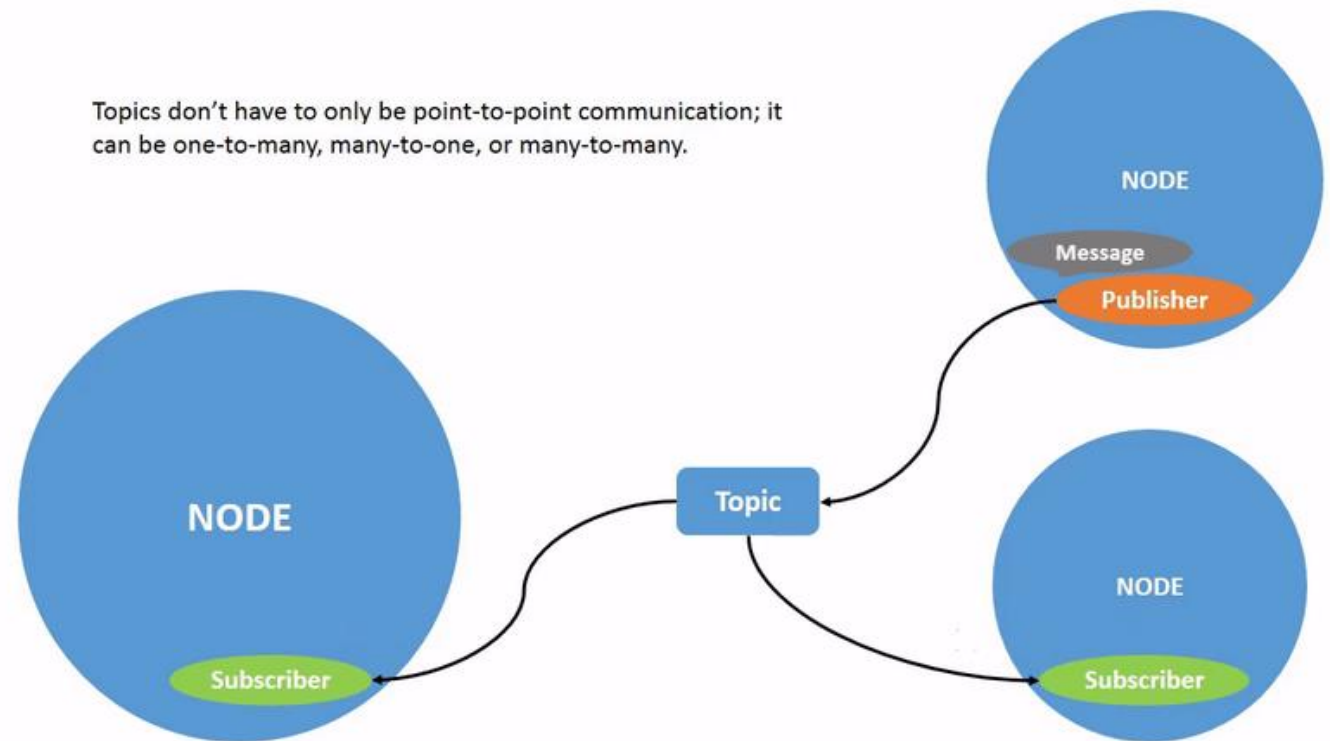
- Vantagens da ROS
  - Modular
  - Sistema de mensagem
  - Visualização
- **Versão: ROS 2 - Humble**



# ROS - Robot Operating System

## Sistema da ROS

- **Nodes (Nós):** Um nó na ROS é um programa que executa uma ação ⇔ Script do Python

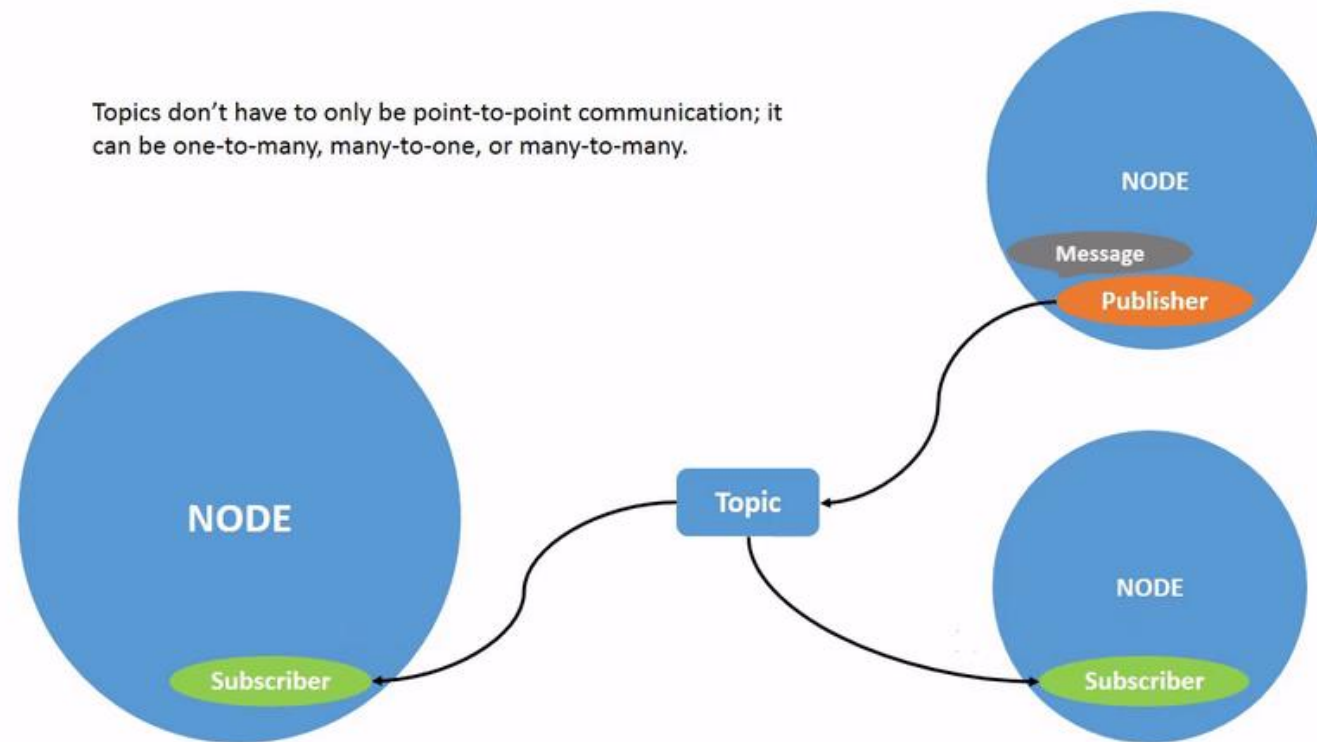




# ROS - Robot Operating System

## Sistema da ROS

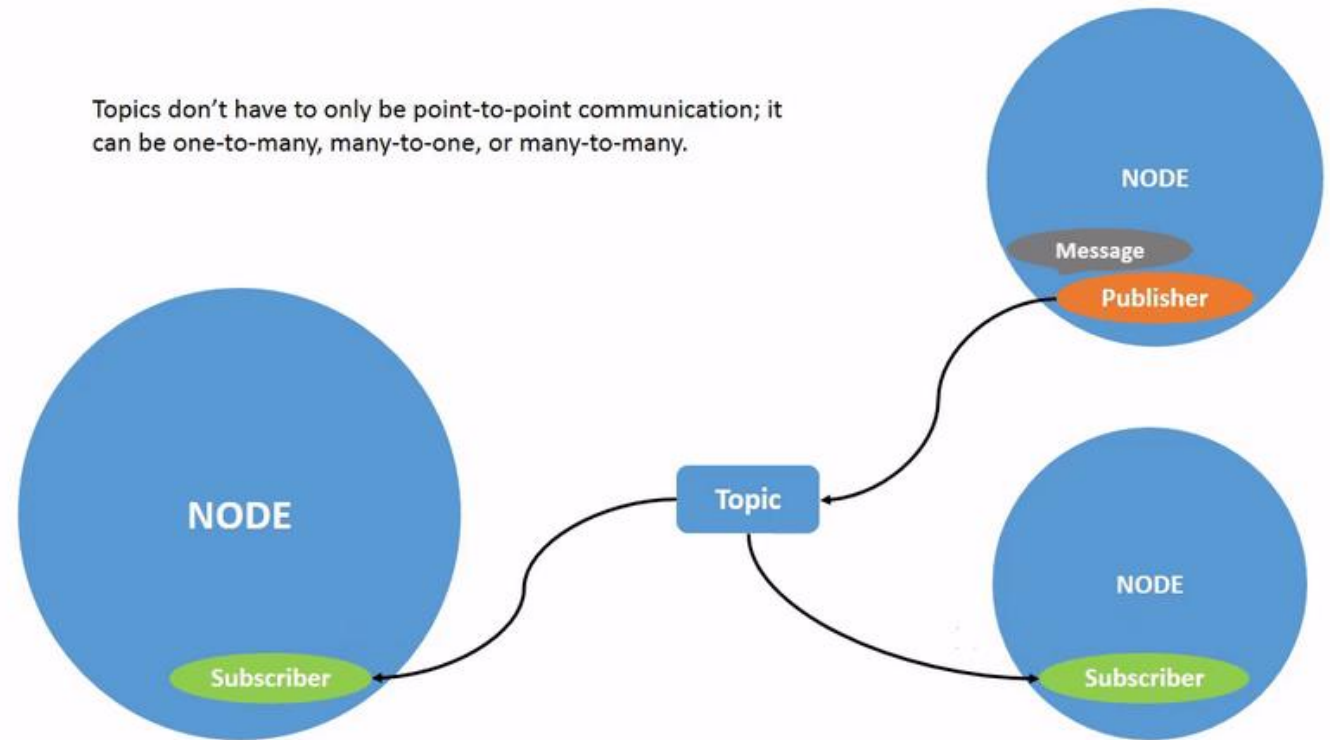
- **Topics (Tópicos):**  
Barramento onde trocamos informações entre nós.
- **Messages (Mensagens):**  
Estruturas de dados que carregam informações



# ROS - Robot Operating System

## Sistema da ROS

- **Publisher:** Nó que publica a mensagem.
- **Subscriber:** Nó que recebe a mensagem.



# Atividades

Agora estão prontos para seguir com as atividades individuais:

- Atividade 1 - Pacotes e comandos importantes no simulador.
- Atividade 2 - Navegando nos tópicos e mensagens da ROS 2.
- Atividade 3 - Criando e configurando um pacote na ROS 2.
- Atividade 4 - Criando um nó subscritor e publicador e configurando o pacote.