

Flujo de datos en multirobótica

Unai Sanz Conejo

u.sanz.2019@alumnos.urjc.es



Trabajo fin de grado

26 de junio de 2024



(CC) Unai Sanz Conejo

Este trabajo se entrega bajo licencia CC BY-NC-SA. Usted es libre de (a) compartir: copiar y redistribuir el material en cualquier medio o formato; y (b) adaptar: remezclar, transformar y crear a partir del material. El licenciador no puede revocar estas libertades mientras cumpla con los términos de la licencia.

Contenidos

- 1 Introducción
- 2 Objetivos
- 3 Plataforma de desarrollo
- 4 Arquitectura
- 5 Desarrollo
- 6 Pruebas y experimentos
- 7 Conclusiones

Introducción

Robótica móvil



Robótica educativa y de bajo coste



a)



b)



c)

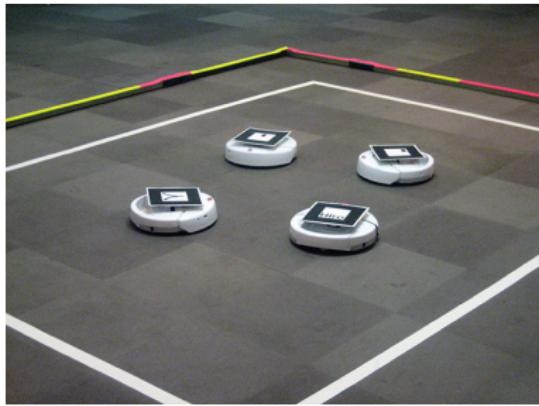
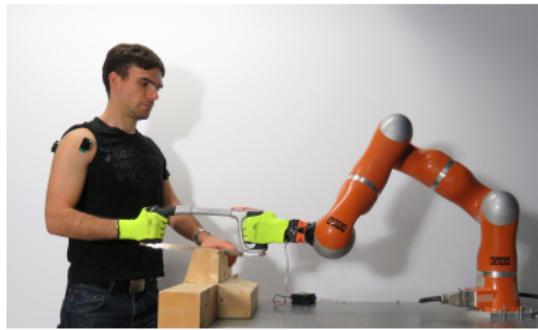


d)



e)

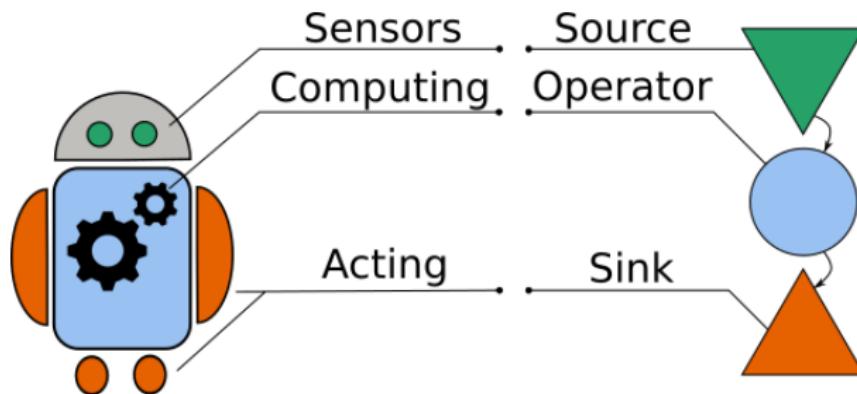
Robótica colaborativa



Flujos de datos en robótica

Comparación Robots con Flujos de datos

- Sensores = nodo Source.
- Cómputo = nodo Operator.
- Actuadores = nodo Sink.

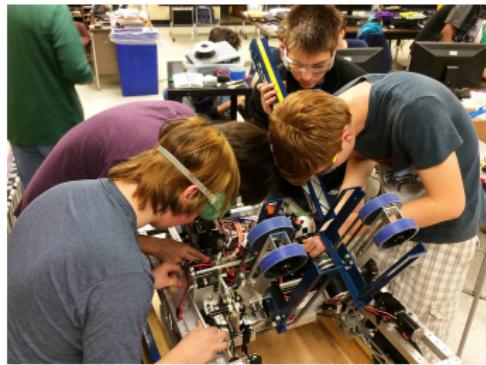
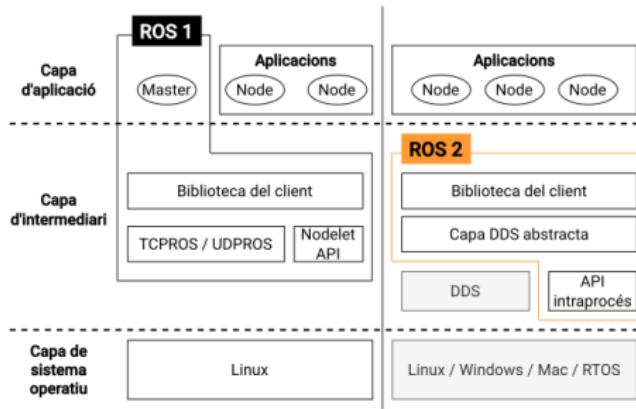


Objetivos

Descripción del problema

Problemas

- Escalón de aprendizaje en robótica.
 - Dificultad de ROS.
- Congestion de red
 - Generada por DDS.



Requisitos

Requisitos

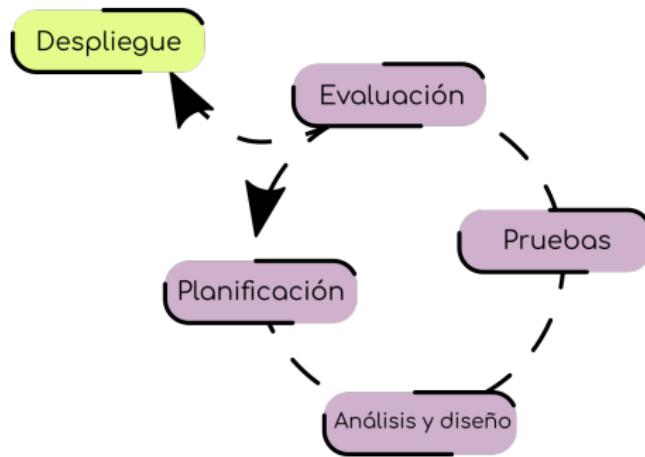
- Ubuntu 22.04 LTS en todas las máquinas.
- Compatibilidad entre herramientas software.
- Facilidad de reproducción y despliegue.
- Sencillez de las aplicaciones robóticas desarrolladas.
- Hardware económico.

Objetivos

- Desarrollo de forma de programación de flujos de datos junto a ROS2.
- Desarrollo de una aplicación robótica que lo demuestre.
- Sencillez de código y hardware económico.
- Solución a la brecha de aprendizaje.
- Reducción de congestión.

Metodología

- Ciclo de desarrollo software **iterativo**.
 - Actualizaciones en **YouTube** y **GitHub**
<https://github.com/RoboticsURJC/tfg-unai>.
- **Reuniones** semanales.
- Desarrollo de la **memoria**.

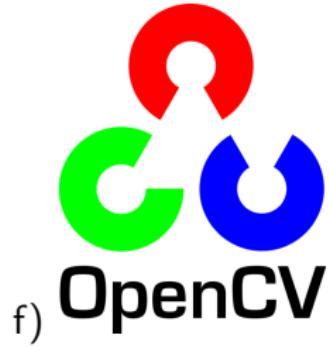
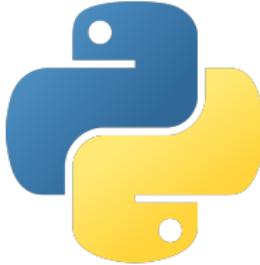


Plataforma de desarrollo

Hardware



Software

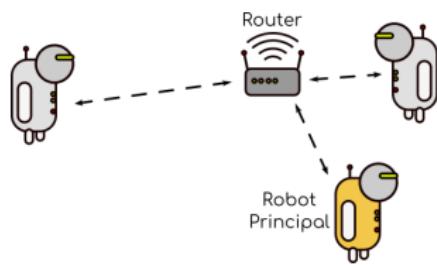


Arquitectura

Topología hardware

Características

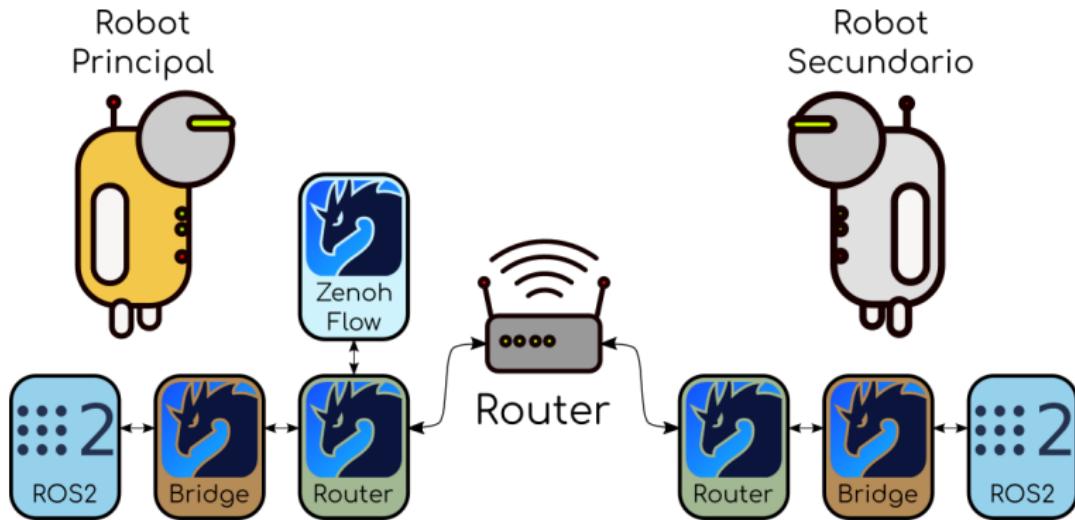
- Red **sin** necesidad de acceso a **Internet**.
- Topología **centralizada** alrededor del portátil.
- Máquina más potente que correrá todo el software.



Topología software

Características

- División en nodos.
- Modularización del código.

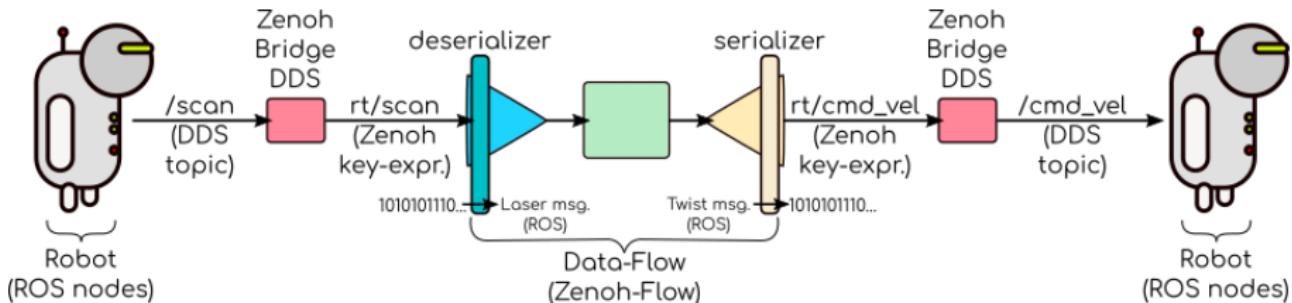


Desarrollo

Programación de flujo de datos

Componentes software

- Zenoh (protocolo de comunicaciones).
- Zenoh-Flow (framework para programación de flujos de datos).
- Zenoh-bridge-DDS (traductor Zenoh-DDS).

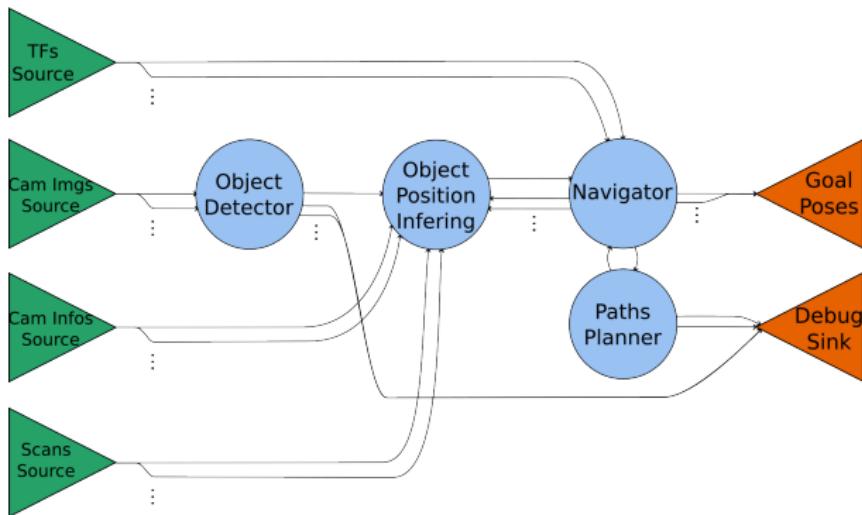


Pruebas y experimentos

Bases del proyecto

Trabajo previo

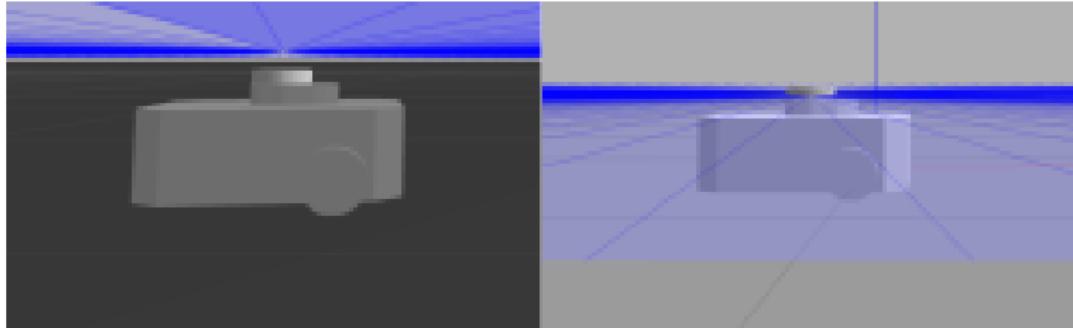
- Aplicaciones previas
- Actualizaciones.
- Mejoras y modificaciones.



Pruebas

Pruebas en simulación

- **Correcciones** de nodos de Zenoh-Flow.
 - nodo Navigator.
 - nodo ObjDetector.
- **Modificaciones** modelo 3D del Turtlebot 3.



Pruebas

Pruebas de laboratorio

- **Telecomunicaciones**, (ROS2, CycloneDDS, Zenoh API, [Zenoh](#), [Zenoh-Flow](#), [Zenoh-bridge-DDS](#)...)
 - **Modificaciones** para laboratorio
 - Mapas
 - Cambios de nombres de frames
 - Solución de bugs (nodo PathsPlanner)
 - **Escalabilidad** a 3 robots.
 - Pruebas con **detector de objetos** (nodo [ObjDetector](#)).
-
- <https://youtu.be/JNb0i57Nfjg>

Conclusiones

Objetivos cumplidos

- Desarrollo de forma de programación de flujos de datos junto a ROS2.
- Desarrollo de una aplicación robótica que lo demuestre.
- Sencillez de código y hardware económico.
- Solución a la brecha de aprendizaje.

Líneas futuras

- Actualización a versiones más estables.
- Paquete de instalación.
- Aplicación con interfaz sencilla e intuitiva.

Flujo de datos en multirobótica

Unai Sanz Conejo

u.sanz.2019@alumnos.urjc.es



Trabajo fin de grado

26 de junio de 2024