



PONTIFICIA UNIVERSIDAD CATÓLICA DE CHILE
ESCUELA DE INGENIERÍA
Dirección de Investigación e Innovación
Programa IPre de Investigación en Pregrado

Proyecto Robot-X Caleuche

Ignacio Segura ^a, Giancarlo Troni ^b

^a *Major en Ingeniería Mecánica, Escuela de Ingeniería, Pontificia Universidad Católica de Chile. Segundo año,*
isegura@uc.cl

^b *Departamento de Departamento de Ingeniería Mecánica y Metalúrgica, Escuela de Ingeniería, Pontificia Universidad Católica de Chile. Profesor adjunto, gtroni@uc.cl*

Resumen

Con el objetivo de servir como una herramienta de detección de errores para el equipo Caleuche dentro del marco de la competencia internacional “The Maritime RobotX Challenge”, se ha dado inicio al desarrollo de una herramienta de monitoreo, cuyo propósito radica en la detección temprana de posibles fallos que el catamarán pueda experimentar durante la competencia, principalmente asociados a problemas de conexión entre los sensores y el computador principal, problemática identificada como crucial en ediciones previas del evento.

Con la finalidad de programar una interfaz gráfica capaz de mostrar simultáneamente los datos provenientes de los distintos componentes y alertar sobre posibles fallas, se utilizaron **GRAFANA** y **FOXGLOVE** como herramientas de visualización de datos. Estos sistemas de software permiten la interpretación de datos obtenidos mediante sensores a través de la creación de un **DASHBOARD** programado, haciendo uso de las capacidades gráficas inherentes a dichos programas.

Debido a la falta de documentación de *Grafana* para su integración con el sistema **ROS2 HUMBLE**, se optó por *Foxglove*, una opción que, además de solventar la carencia documental, facilita la visualización de cámaras y LIDARS. Tras su implementación con el simulador VRX disponible para la competencia, se inició la investigación con los sensores previstos para su uso en la competición. Los aspectos técnicos de *ROS 2 Humble* resultaron ineficaces en la interpretación por parte de *Grafana*.

Como resultado del proceso investigativo, se logró desarrollar con éxito un dashboard capaz de registrar, detectar y visualizar tempranamente errores vinculados a la conexión entre los sensores a bordo y el computador principal en tierra. Este dashboard incluye datos relacionados con la dirección



PONTIFICIA UNIVERSIDAD CATÓLICA DE CHILE
ESCUELA DE INGENIERÍA
Dirección de Investigación e Innovación
Programa IPre de Investigación en Pregrado

del viento, GNSS, nubes de puntos generadas por un láser-escáner, así como transmisiones en tiempo real de las cámaras incorporadas.

Palabras Clave: Robótica, Sensores, *Dashboard*, *Computer vision*, Simulación.



PONTIFICIA UNIVERSIDAD CATÓLICA DE CHILE
ESCUELA DE INGENIERÍA
Dirección de Investigación e Innovación
Programa IPre de Investigación en Pregrado

1. Introducción

El equipo universitario Caleuche se prepara para la nueva edición del *Robot-X Challenge 2024*, una competencia centrada en la programación y control de un WAM-V (*Wave Adaptive Modular Vehicle*) con el fin de cumplir metas específicas. Para lograr estos objetivos se emplean diversos sensores cuya desconexión o falla podría representar un problema grave, por lo tanto, una detección temprana es necesaria tanto en ambientes de prueba como durante la competencia.

Tradicionalmente, se utiliza la herramienta integrada en ROS 2 (*Robot Operating System*):

ros2 topic echo

para monitorear los estados de los sensores. Sin embargo, debido a la considerable cantidad de estos, se hace necesaria la automatización del proceso, incluyendo una rápida notificación del estado de cada uno. Dado el siguiente contexto, se da inicio al desarrollo de un *dashboard* que cumpla los requerimientos planteados.



PONTIFICIA UNIVERSIDAD CATÓLICA DE CHILE
ESCUELA DE INGENIERÍA
Dirección de Investigación e Innovación
Programa IPre de Investigación en Pregrado

2. Experimentación o metodología

El primer desafío en el desarrollo de la tecnología requerida se centra en la familiarización con **UBUNTU 22**, el sistema operativo utilizado para el manejo **ROS** y **ROS 2 Humble**, la versión utilizada por el equipo durante el desarrollo de la competencia. Posterior a esta etapa inicial, se inicia la búsqueda de la herramienta adecuada para la implementación del *dashboard*.

El primer software analizado a profundidad fue *Grafana*. A pesar de contar con una gran gama de opciones para graficar y tiempos de respuesta ajustables, su viabilidad se vio obstaculizada ante la falta de documentación que indique su conexión con el sistema **ROS 2 Humble**. Aunque se halla la documentación pertinente para su correcta implementación con **ROS**, no se encontró información oficial para **ROS 2**.



Figura 1.- Demo realizada en *Grafana*

En este contexto, surge el análisis de *Foxglove Studio*, una herramienta orientada a la creación de *dashboards* para robots. Dentro de su documentación se encuentra la información completa para su conexión con los diferentes **TÓPICOS** de **ROS 2**. Además, a diferencia de *Grafana*, permite la visualización de imágenes, nubes de puntos y mapas.

Para visualizar los tópicos publicados por **ROS 2**, se hizo necesaria la instalación de *foxglove_bridge*, componente suscrita a los tópicos. Tras su instalación para la distribución de *Humble*, el código permite leer y transmitir los tópicos entregados al *dashboard* de la aplicación para su representación gráfica.



PONTIFICIA UNIVERSIDAD CATÓLICA DE CHILE
ESCUELA DE INGENIERÍA
Dirección de Investigación e Innovación
Programa IPre de Investigación en Pregrado

El proceso de utilización de *foxglove_bridge* se lleva a cabo mediante la siguiente línea de código:

```
ros2 launch foxglove_bridge foxglove_bridge_launch.xml port:= <número>
```

Dentro de la aplicación de Foxglove se debe seleccionar '*Open connection*' y, dentro de la opción *Foxglove WebSocket*, seleccionar el mismo puerto utilizado en la línea de código anterior.

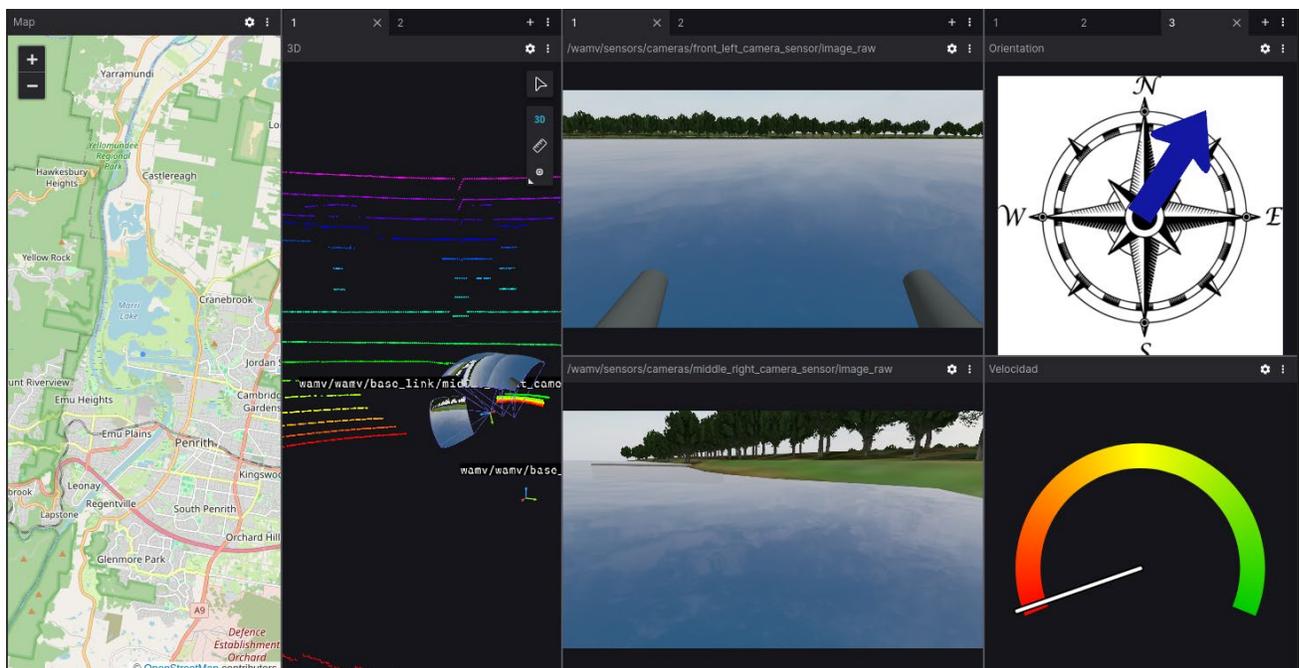


Figura 2.- DEMO realizada en Foxglove

Para realizar la primera prueba, se empleó el **sistema de simulación de VRX**. Cabe destacar que, durante este proceso, el computador utilizado sufrió un error considerable con *ROS* lo que requirió su formateo. Este incidente facilitó la instalación del simulador, lo que permitió finalizar la primera DEMO.

Adicionalmente, se identificó la necesidad de recibir notificaciones de actividad por parte de cada sensor. Actualmente en el laboratorio se cuenta con 3, un *GNSS*, una cámara y un *LIDAR*. Ninguno



PONTIFICIA UNIVERSIDAD CATÓLICA DE CHILE
ESCUELA DE INGENIERÍA
Dirección de Investigación e Innovación
Programa IPre de Investigación en Pregrado

de estos sensores posee un tópicos por defecto que indique su estado, por lo que se hace imprescindible la creación de un nuevo tópicos que provea esta información.

En consecuencia, se desarrolla un primer código en *PYTHON* que recibe información de un tópicos predeterminado en intervalos regulares de tiempo. En caso de no recibir datos dentro de este periodo, aparece un mensaje de error.

El código inicial funciona de la siguiente manera: al recibir un mensaje por parte del tópicos al que se ha suscrito, inicia un contador manejado por la función *datetime*. Simultáneamente, otra función verifica si el tiempo transcurrido desde la recepción del último mensaje supera un valor definido (0.1 segundos en este primer experimento). En otras palabras, cada vez que el nodo suscrito publique información, el código publica en un nuevo tópicos un valor de 1; si no se recibe información dentro del tiempo establecido, publica un valor de 0.

En el laboratorio, se identificaron dos dificultades: en primer lugar, la clase de tópicos que publica es

sensor_msgs/msg/Imu

en segundo lugar, la fiabilidad de su *QoS* (*Quality of Service*) es *'BEST_EFFORT'*. El código utilizado dentro del simulador no puede leer la clase tipo *IMU*, además, su fiabilidad es *'RELIABLE'*, Se observa en la documentación de ROS 2 como ambas son incompatibles. Se añade dentro del código un nuevo *QoS*, además de importar el tópicos *Imu*.



PONTIFICIA UNIVERSIDAD CATÓLICA DE CHILE
ESCUELA DE INGENIERÍA
Dirección de Investigación e Innovación
Programa IPre de Investigación en Pregrado

Finalmente, el resultado producido por el *LIDAR* se debe de observar de la siguiente manera:

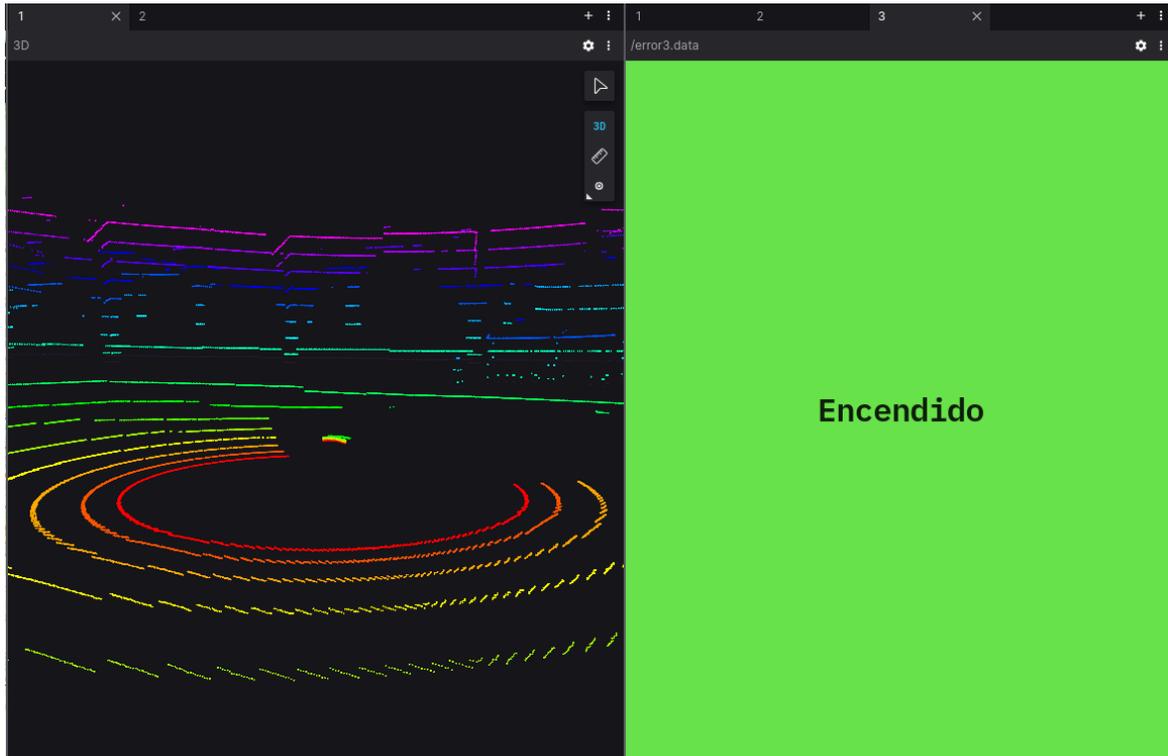


Figura 3.- DEMO LIDAR realizada en Foxglove

Debido a dificultades a la hora de acceder al laboratorio durante el fin de noviembre y comienzos de diciembre, se utiliza nuevamente el **sistema de simulación de VRX** con su tópico Imu integrado, de todas formas, se puede observar como la fiabilidad del *QoS* del tópico creado ha cambiado.



PONTIFICIA UNIVERSIDAD CATÓLICA DE CHILE
ESCUELA DE INGENIERÍA
Dirección de Investigación e Innovación
Programa IPre de Investigación en Pregrado

3. Resultados y discusión

El desarrollo del *dashboard* fue exitoso, logrando conectar la información de *ROS 2* a un único espacio con el cual trabajar, además, el uso de *Foxglove* permite conectar los tópicos que sean necesarios para el futuro del equipo. Entre las ventajas observadas en contraste a *Grafana* se encuentran la visualización de imágenes, mapas y nubes de puntos generadas por los diversos sensores, llegando a competir con el sistema *RVIZ* utilizado comúnmente durante la competencia.

Finalmente, el código encargado de entregar el estado del sensor medido posee las características requeridas para su correcto funcionamiento, y la solución entregada es capaz de ser extrapolada a cualquier tópico requerido. Cabe destacar que durante la investigación fueron utilizados únicamente 3 sensores, y cambios mínimos pueden ser requeridos para su extrapolación a distintos tipos de sensores.



PONTIFICIA UNIVERSIDAD CATÓLICA DE CHILE
ESCUELA DE INGENIERÍA
Dirección de Investigación e Innovación
Programa IPre de Investigación en Pregrado

4. Conclusiones

Durante el desarrollo de la investigación fue necesario el aprendizaje de *ROS2*, *C++*, *Ubuntu*, *Python* implementado en robótica y *Foxglove*. La creación del dashboard fue exitosa, utilizando la herramienta *Foxglove* como base y un código en *Python3* se logra finalmente el cometido. Debido a la utilización de datos obtenidos mayoritariamente del simulador, resulta importante realizar una prueba exhaustiva en cada uno de los sensores reales, debido a diferentes errores que puedan presentar a la hora de ser conectados, a la vez que se debe examinar la eficacia del funcionamiento del sistema en tiempo real.

Finalmente, se realizará la documentación del proyecto con el fin de ser replicado por cada siguiente miembro del equipo con facilidad.



PONTIFICIA UNIVERSIDAD CATÓLICA DE CHILE
ESCUELA DE INGENIERÍA
Dirección de Investigación e Innovación
Programa IPre de Investigación en Pregrado

❖ **Agradecimientos**

A Cristian Nova por su asesoramiento en *ROS 2*, el simulador del equipo, introducción a los diferentes sensores del laboratorio e idea de ocupar *Foxglove*.

A Octavio Águila por su organización todas las semanas para llevar constancia de los avances del grupo, además de su introducción en el uso de *rosbags*.

A Tomas Contreras por su introducción en *Ubuntu 22*, *Docker* e idea de utilizar *Grafana*.

A Javiera Fuentes por su coordinación y dirección, por siempre tener una disposición a ayudar y resolver dificultades o dudas, además de siempre estar al tanto de mis avances.



PONTIFICIA UNIVERSIDAD CATÓLICA DE CHILE
ESCUELA DE INGENIERÍA
Dirección de Investigación e Innovación
Programa IPre de Investigación en Pregrado

❖ **GLOSARIO**

DASHBOARD: Tablero de mandos.

FOXGLOVE: Herramienta de diagnóstico y visualización para robots.

GRAFANA: Software de código abierto que permite la visualización de diversos datos.

IMU: Unidad de medición inercial, datos de movimiento.

PYTHON: Lenguaje de programación orientado a objetos.

QOS: Rendimiento promedio de un servicio o mensaje (tópico).

ROS2 (Humble): Software de código abierto para diseñar y controlar robots.

TÓPICO/TOPIC: Mensajes utilizados por ROS para comunicar diversos sensores o componentes.

UBUNTU: Sistema operativo de código abierto de la distribución Linux.



PONTIFICIA UNIVERSIDAD CATÓLICA DE CHILE
ESCUELA DE INGENIERÍA
Dirección de Investigación e Innovación
Programa IPre de Investigación en Pregrado

❖ **Referencias**

Enterprise open source and Linux. (s/f). Ubuntu. Recuperado el 10 de diciembre de 2023, de <https://ubuntu.com/>

Grafana documentation. (s/f). Grafana Labs. Recuperado el 10 de diciembre de 2023, de <https://grafana.com/docs/grafana/latest/>

Introduction. (s/f). Foxglove.dev. Recuperado el 10 de diciembre de 2023, de <https://docs.foxglove.dev/docs/introduction/>

QOS python code for create_subscriber - ROS Answers: Open Source Q&A Forum. (s/f). Ros.org. Recuperado el 10 de diciembre de 2023, de

https://answers.ros.org/question/360676/qos-python-code-for-create_subscriber/

Quality of Service settings — ROS 2 Documentation: Rolling documentation. (s/f). Ros.org. Recuperado el 10 de diciembre de 2023, de

<https://docs.ros.org/en/rolling/Concepts/Intermediate/About-Quality-of-Service-Settings.html>

RobotX. (2019, agosto 20). RobotX. <https://robotx.org/>

ROS 2 documentation — ROS 2 documentation: Humble documentation. (s/f).

Ros.org. Recuperado el 10 de diciembre de 2023, de

<https://docs.ros.org/en/humble/index.html>

ROS Foxglove bridge. (s/f). Foxglove.dev. Recuperado el 10 de diciembre de 2023, de <https://docs.foxglove.dev/docs/connecting-to-data/ros-foxglove-bridge>

Understanding topics — ROS 2 Documentation: Foxy documentation. (s/f). Ros.org.

Recuperado el 10 de diciembre de 2023, de

<https://docs.ros.org/en/foxy/Tutorials/Beginner-CLI-Tools/Understanding-ROS2-Topics/Understanding-ROS2-Topics.html>

VRX Competition 2023. (2023, abril 13). RobotX. <https://robotx.org/programs/vrx-2023/>

What is python? Executive summary. (s/f). Python.org. Recuperado el 10 de diciembre de 2023, de <https://www.python.org/doc/essays/blurb/>