

# Robot de bajo coste para el mantenimiento de carreteras

Julia López Augusto

[j.lopeza.2020@alumnos.urjc.es](mailto:j.lopeza.2020@alumnos.urjc.es)



Trabajo Fin de Grado

xx de diciembre de 2024



(CC) Julia López Augusto

*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 Diseño y construcción del robot

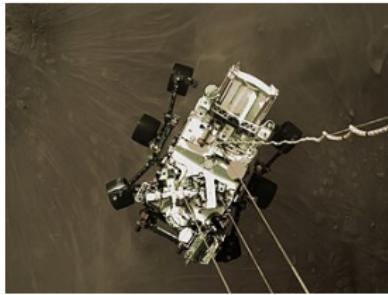
5 Soporte software del robot

6 Experimentos

7 Conclusiones

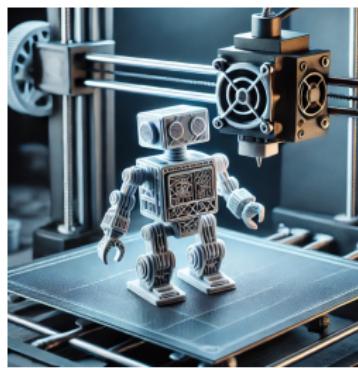
# *Introducción*

# Robots de campo



# *Objetivos*

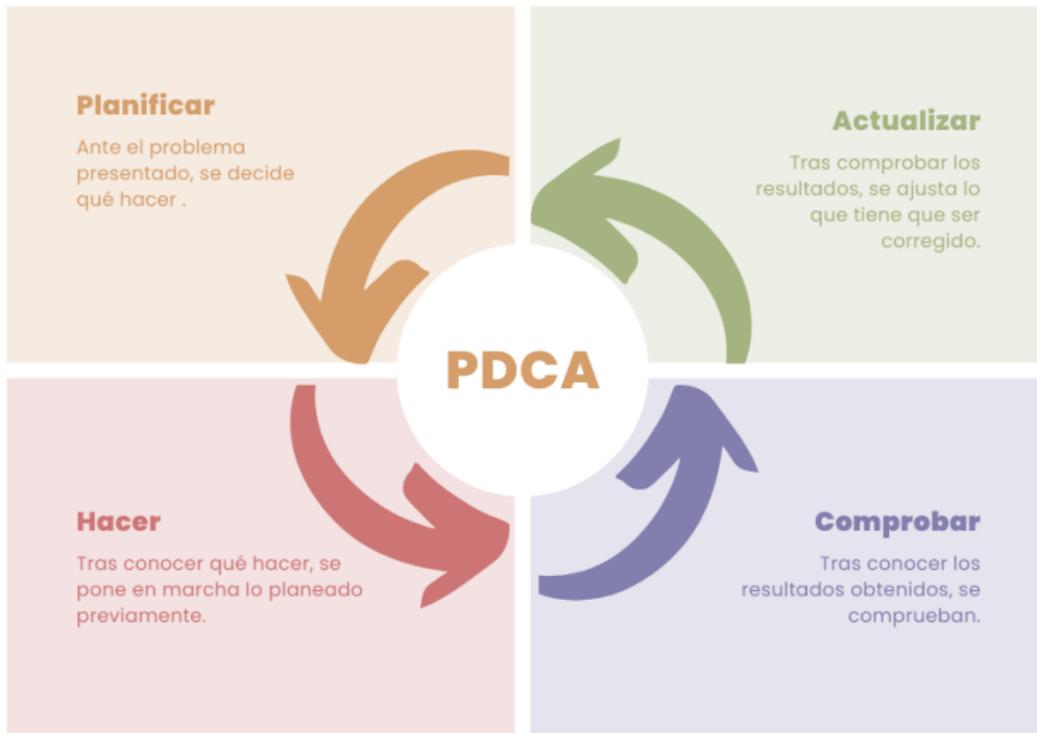
# Descripción del problema



# Requisitos

- ① Coste inferior a 250€.
- ② Piezas impresas en una impresora convencional.
- ③ Ubuntu como sistema operativo.
- ④ No se requiere tarjeta gráfica dedicada para entrenar modelos.
- ⑤ Modelos adaptados a las limitaciones hardware.
- ⑥ Integración con ROS 2.

# Metodología



# Plan de trabajo



Repository

Wiki

## DIARIO

En este apartado se va a detallar los mínimos avances obtenidos según se desarrolle la investigación. Dentro desplegable podrás encontrar toda la información al respecto.

1. [Septiembre 2023](#)
2. [Octubre 2023](#)
3. [Noviembre 2023](#)
4. [Febrero 2024](#)
5. [Marzo 2024](#)
6. [Abril 2024](#)
7. [Junio 2024](#)
8. [Julio 2024](#)
9. [Agosto 2024](#)
10. [Septiembre 2024](#)
11. [Octubre 2024](#)
12. [Noviembre 2024](#)

## EVOLUCIÓN PROYECTO

En este apartado se van a incluir todos los datos acerca del proyecto que se incluirán en la memoria final.

1. [Estado del arte](#)
2. [Hardware
  - \[Materiales
    - \\[Raspberry pi 4\\]\\(#\\)
    - \\[Motores\\]\\(#\\)
    - \\[Cámaras\\]\\(#\\)
    - \\[Módulo GPS\\]\\(#\\)\]\(#\)
  - \[Estructura
    - \\[Impresión 3D\\]\\(#\\)\]\(#\)](#)
3. [Software
  - \[Motores\]\(#\)
  - \[Módulo GPS\]\(#\)
  - \[Cámaras\]\(#\)](#)

# *Plataforma de desarrollo*

# Hardware

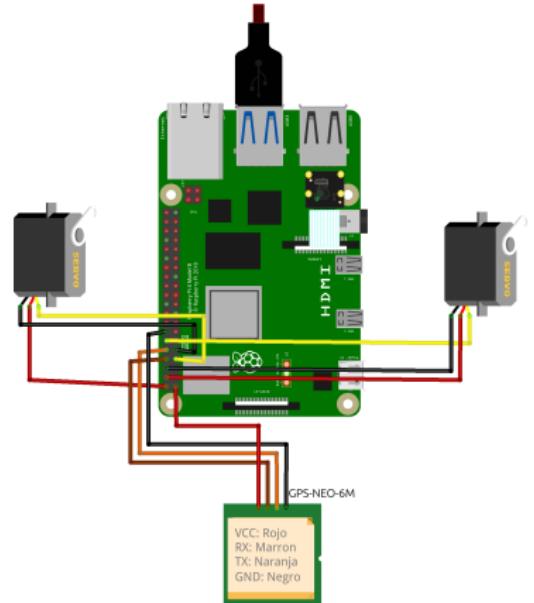


# Software

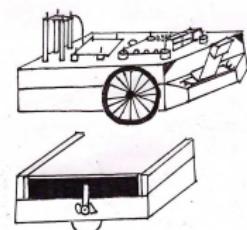


# *Diseño y construcción del robot*

# Geometría del robot



# Bocetos, maquetas y planos



MEDIDAS PLATEAU BASIC



MEDIDAS PUCA RESPIRATOR



MEDIDAS BASE: Ancho = 2 mm

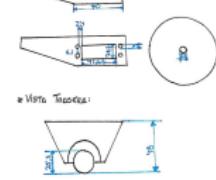
\* Vista Frontal:



\* Vista Lateral:



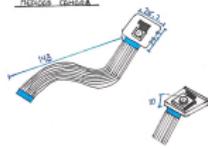
\* Vista Trasera:



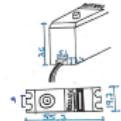
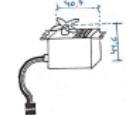
MEDIDAS MÓDULO GPS



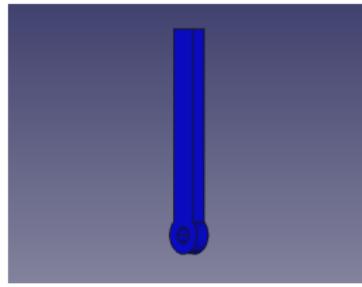
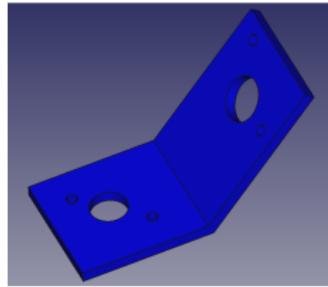
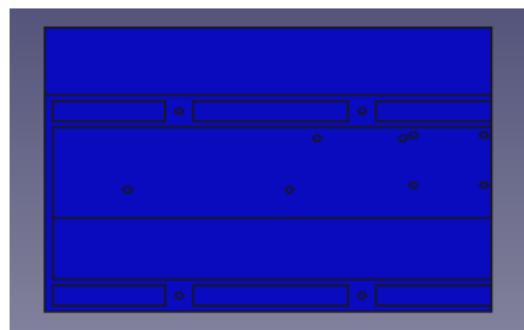
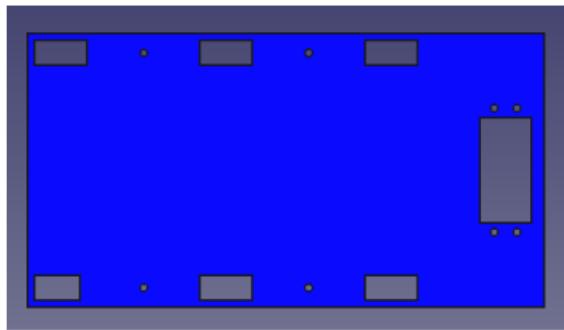
MEDIDAS CÁMARA



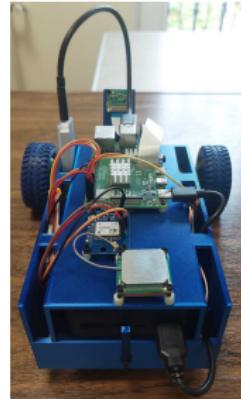
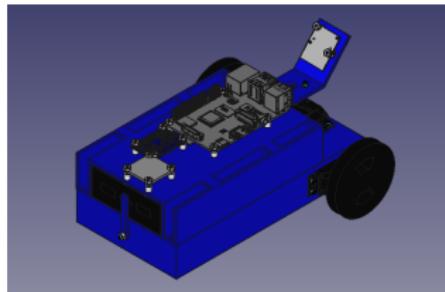
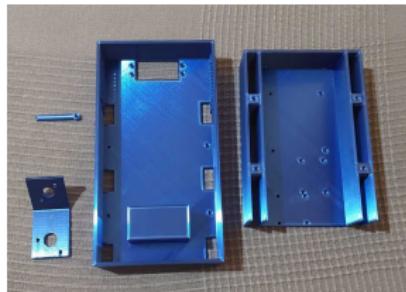
MEDIDAS SERVO



# Piezas diseñadas

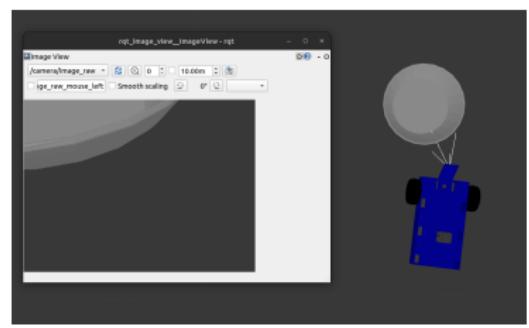
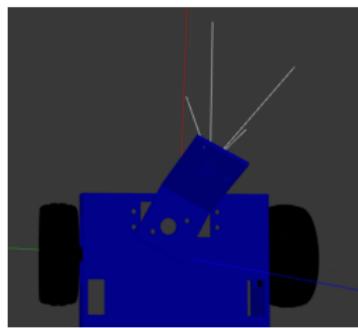
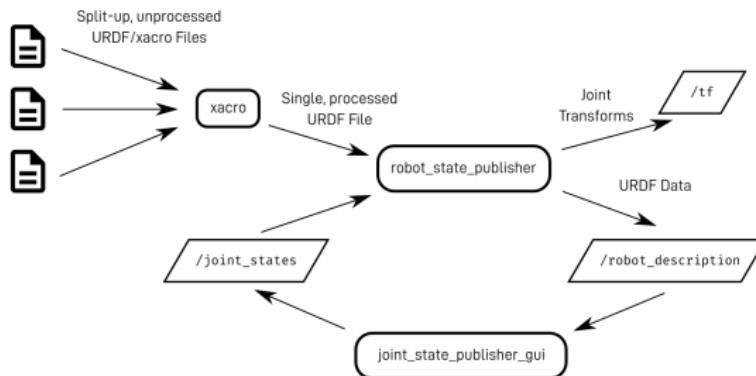


# Impresión y montaje



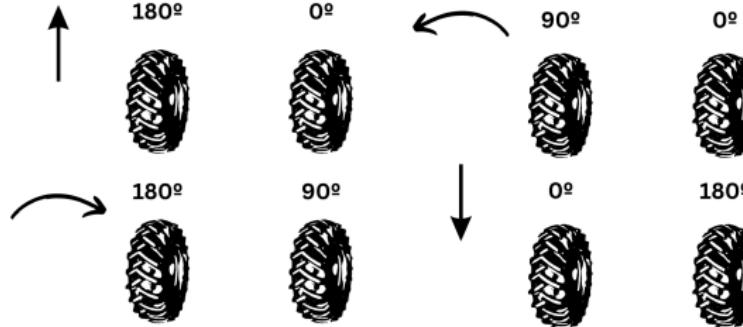
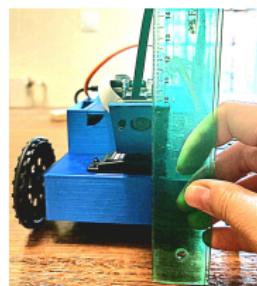
# *Soporte software del robot*

# Simulación

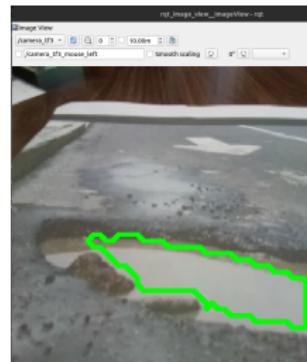
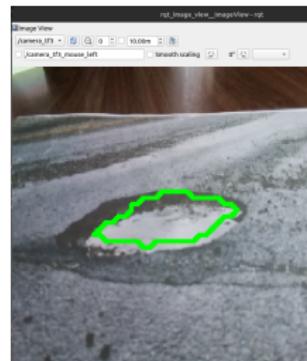
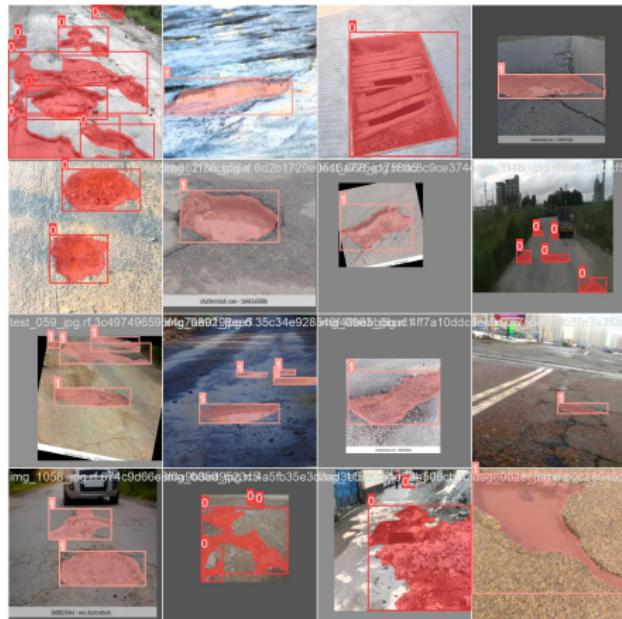


# Configuración del robot real

$$\begin{bmatrix} 497,66 & 0 & 325,3 \\ 0 & 502,16 & 240,18 \\ 0 & 0 & 1 \end{bmatrix}$$



# Detección y obtención del contorno del bache



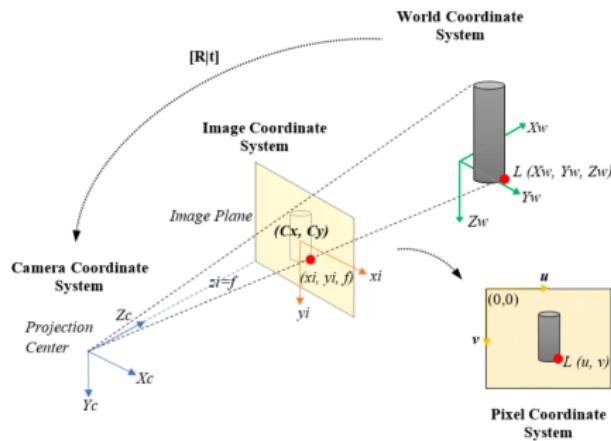
Vídeo de la demostración

# Modelo de cámara pinhole

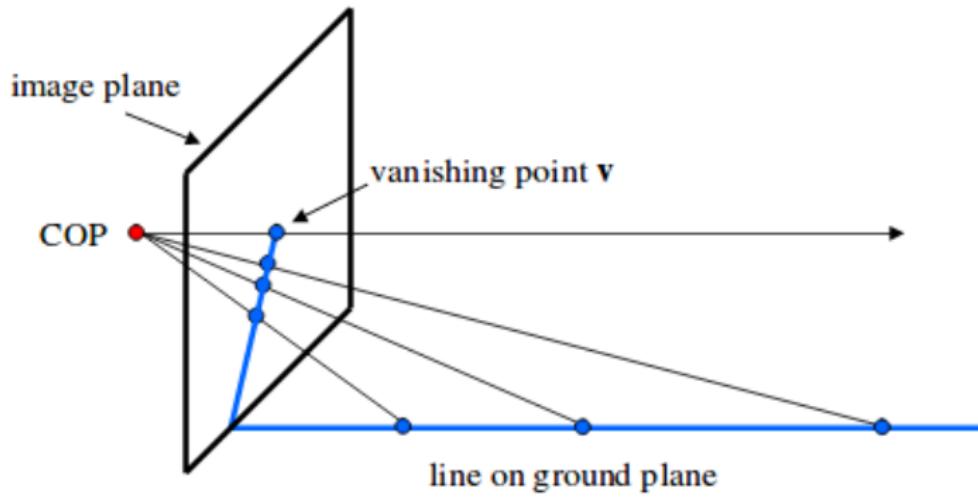
$$\begin{matrix} \text{Factor de escala para homogeneizar } u \text{ y } v \\ \text{Coordenadas del punto proyectivo en pixels} \end{matrix} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & [s] & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{22} & t_z \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

Parámetros intrínsecos      Parámetros extrínsecos

Distancia focal x      Distancia focal y      Ángulo de giro      Centro imagen x      Centro imagen y      Centro de un punto en el mundo real



# Hipótesis suelo

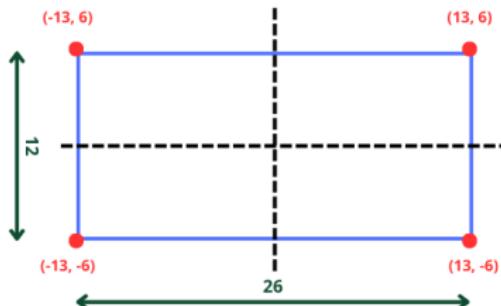


# Algoritmo de la lazada

$$A = \frac{1}{2} \left| \sum_{i=1}^{n-1} x_i y_{i+1} + x_n y_1 - \sum_{i=1}^{n-1} x_{i+1} y_i - x_1 y_n \right|$$

Donde  $A$  es el área del polígono,  $n$  es el número de lados del polígono y  $(x_i, y_i)$ ,  $i = 1, 2, \dots, n$  son los vértices del polígono de forma alternativa.

Ejemplo teórico:



Área del rectángulo = base x altura  
 $A = 26 \times 12 = 312$

Algoritmo de la lazada:

$$\begin{array}{r} (13, 6) \\ (13, -6) \\ \hline \end{array} \quad \begin{array}{r} (-13, -6) \\ (-13, 6) \\ \hline \end{array} \quad \frac{1}{2} \quad \text{pos - neg}$$

$$\begin{aligned} \text{pos} &= 13 \cdot (-6) + 13 \cdot (-6) + (-13) \cdot 6 + (-13) \cdot 6 = -312 \\ \text{neg} &= 13 \cdot 6 + (-13) \cdot (-6) + (-13) \cdot (-6) + (-13) \cdot 6 = 312 \end{aligned}$$

$$\frac{1}{2} \quad \boxed{-312 - 312} = 312$$

# Interfaz web

192.168.5.73:8000/index.html

### Mapa de Seguimiento GPS

The map displays a green polygon representing a tracked area. A blue marker indicates the current position. A tooltip shows the calculated area: "Área calculada: 614.0255737304688 mm²". The map includes street names like "Calle Matadero" and "Calle Ronda", and a green shaded area labeled "Parque de los espagos". To the right, there's a vertical road labeled "Cerro de los Quirones" and "Cerro de los Cisneros".

### Mando Controlador

A control interface featuring a central square button with a blue arrow pointing up, surrounded by four smaller square buttons with blue arrows pointing up, down, left, and right respectively.

# Detección de líneas

## Distintos casos de detección de líneas de PiBot

Detecta 1 línea:

- Va pegado al arcén
- Va por la línea central



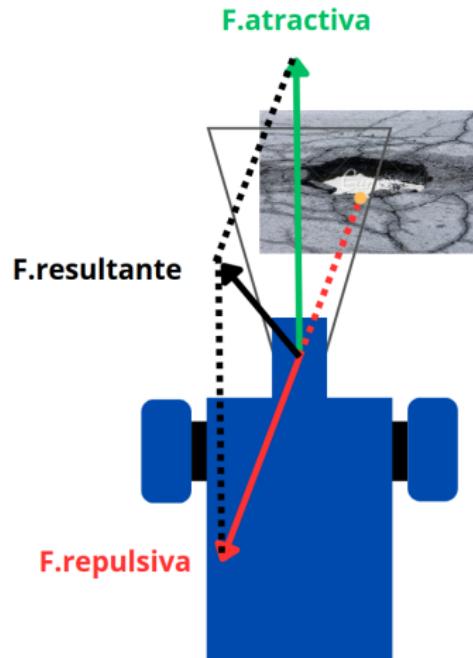
Detected 2 lines when  
going between the two  
lanes



No detecta líneas:

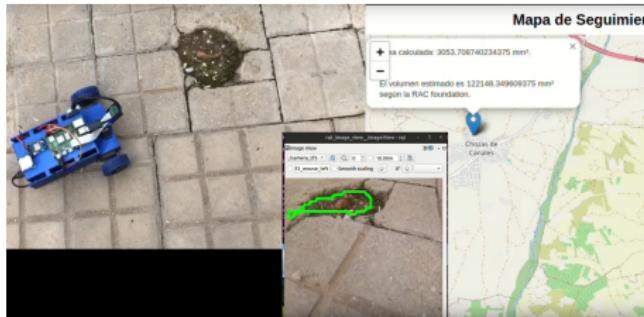
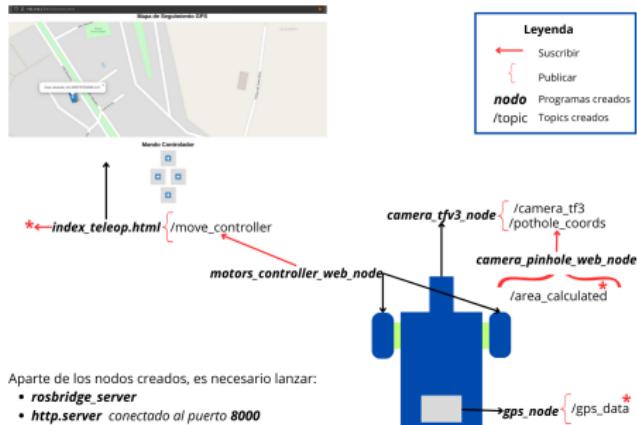
- Going between the two lanes when there is no central line
- When it is outside the road

## VFF

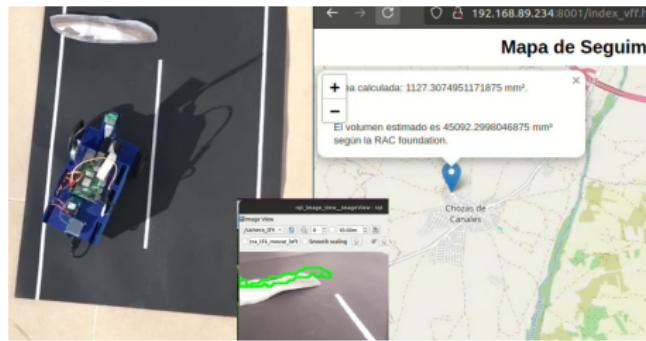
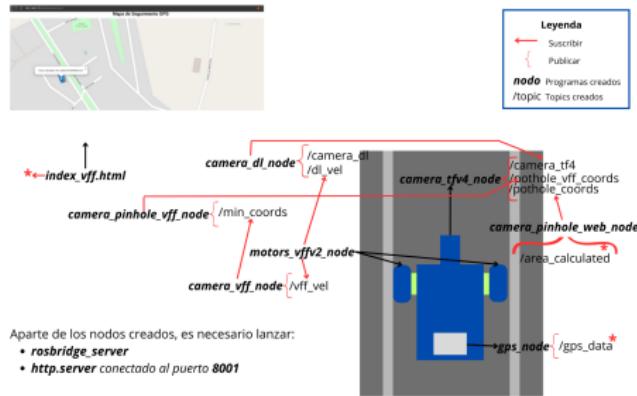


# *Experimentos*

# Teleoperado



# Autónomo



# *Conclusiones*

# Habilidades desarrolladas

- FreeCAD.
- Mecánica y ensamblaje de piezas.
- Crear un robot en simulación.
- ROS 2 Control.
- Modelo de aprendizaje supervisado en una Raspberry Pi.
- Integración de ROS 2 con páginas web.
- Generar documentación de calidad en LaTeX.

# Líneas futuras

- Soporte software al robot en simulación.
- Integrar el motor de la cámara.
- Modificar la altura cámara.
- Mantener a PiBotJ en la última versión.
- Implantar una versión más robusta de PiBotJ como un asistente real.

# Robot de bajo coste para el mantenimiento de carreteras

Julia López Augusto

[j.lopeza.2020@alumnos.urjc.es](mailto:j.lopeza.2020@alumnos.urjc.es)



Trabajo Fin de Grado

xx de diciembre de 2024