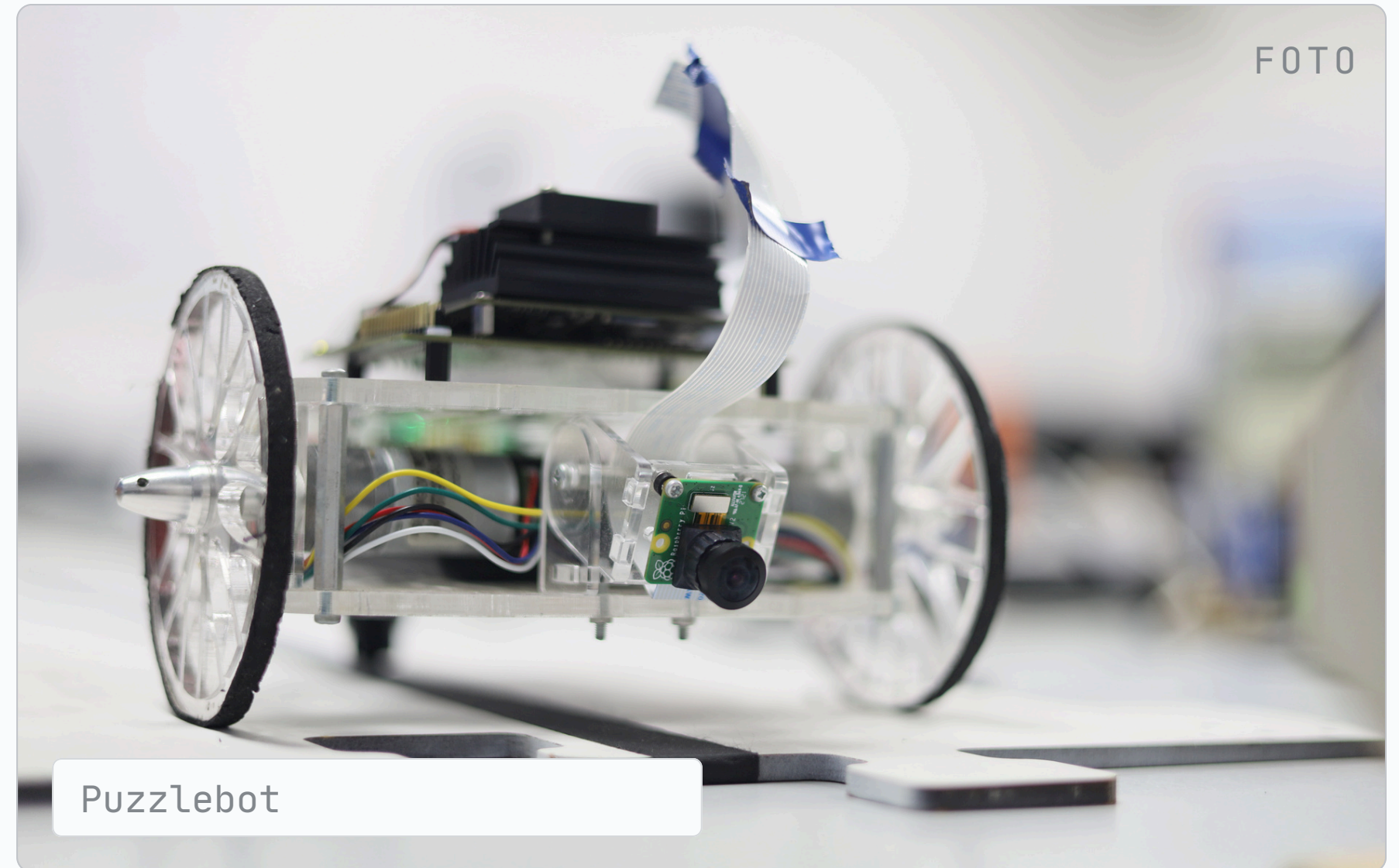




Robot Móvil Autónomo

Seguimiento de línea, detección de
semáforos y señales de tráfico.

OMAR • ARTURO • RICARDO



50%

Seguimiento de línea

20%

Semáforos

30%

Señales

01 • OBJETIVO

Objetivo del Prototipo

Un robot que recorre una pista a escala de forma **autónoma**, obedece **semáforos** y reacciona a **señales de tráfico** — sin teleoperación durante el intento.

01 Recorre la pista

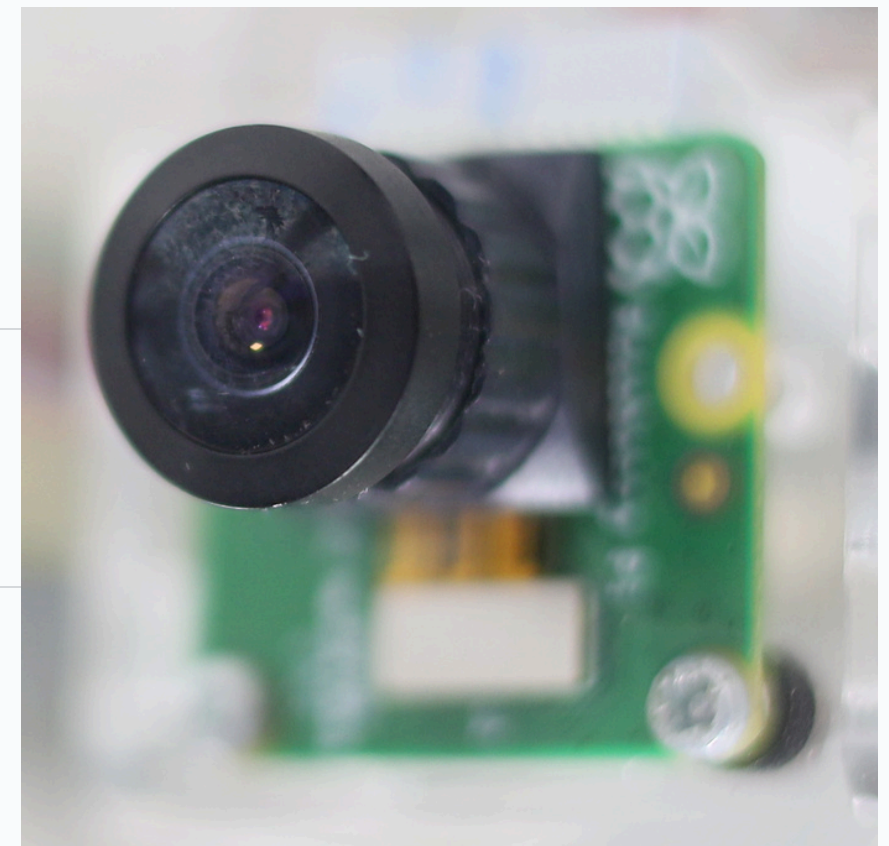
Avanza siguiendo la línea con ambas ruedas sobre el MDF. Es la base de la calificación.

02 Obedece semáforos

Detecta el estado del semáforo y actúa: se detiene en rojo, reduce en amarillo, avanza en verde.

03 Responde a señales

Reconoce Stop, Give Way, Work in Progress y flechas, y ejecuta la acción correspondiente.



Arquitectura del Sistema

Una cadena **percepción** → **detección** → **decisión** → **control** sobre ROS2. Cada acción del robot tiene un tópico que la respalda.

PERCEPCIÓN

VisionCore

Imagen de cámara →
procesamiento de línea

`/line_error`

`/line_detected`

`/crossing_detected`

DETECCIÓN

Detector YOLO

Señales y semáforos

`/detected_sign` `/sign_area`

`/detected_color`

DECISIÓN

ControllerOd

Combina línea, señal, área, color
y cruce

→ `velocidad y giro`

CONTROL

Robot

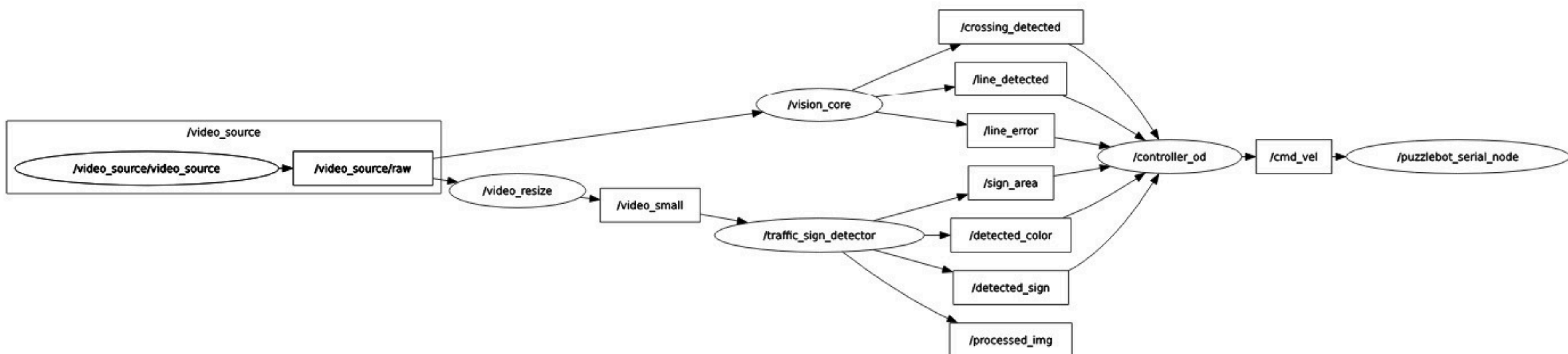
Movimiento físico, sin
teleoperación

`/cmd_vel`

“El controlador toma decisiones y envía velocidades a `/cmd_vel` sin teleoperación durante el intento.”

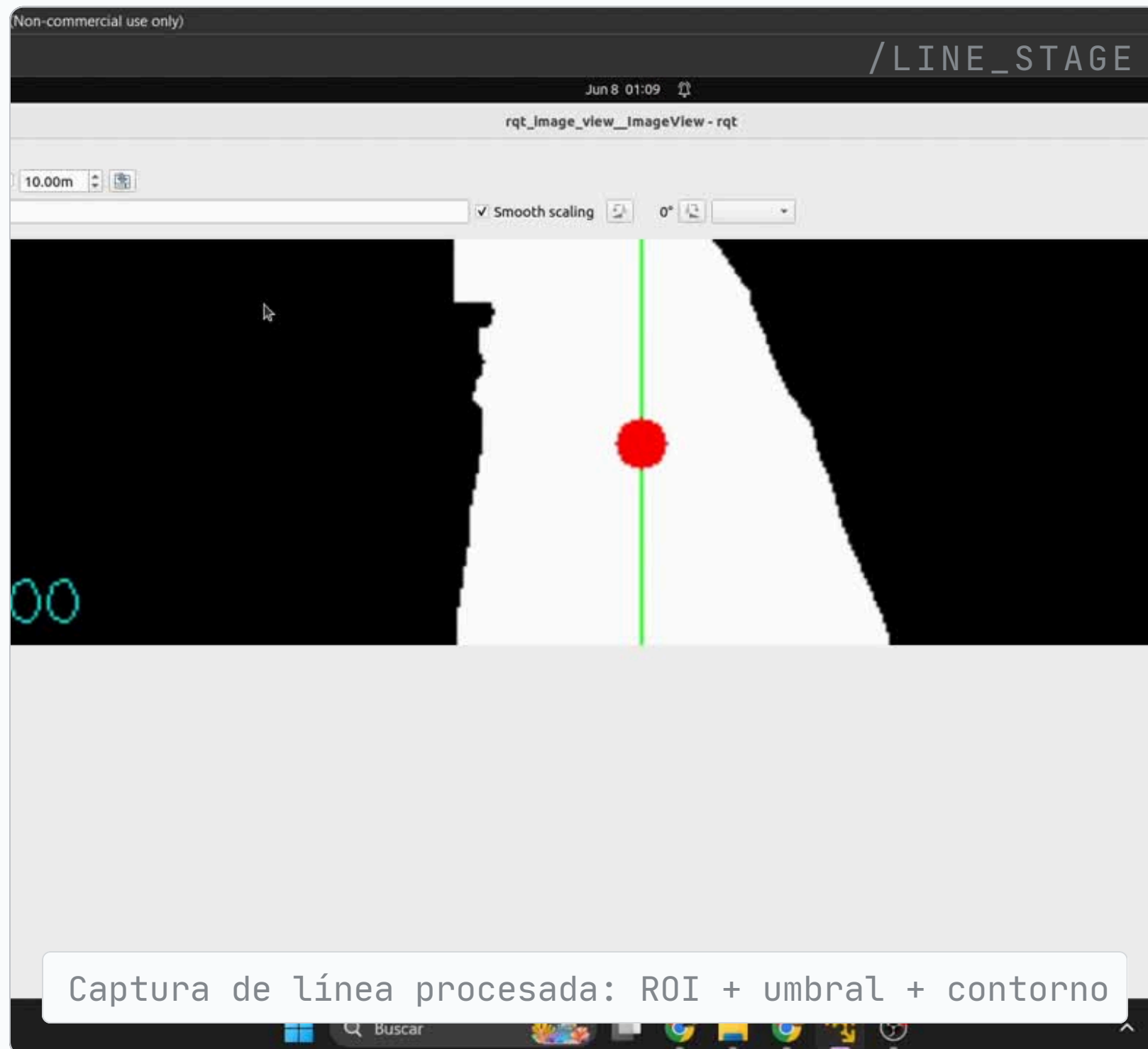
07 • EN VIVO

Nodos



Seguimiento de Línea

50% de la rúbrica



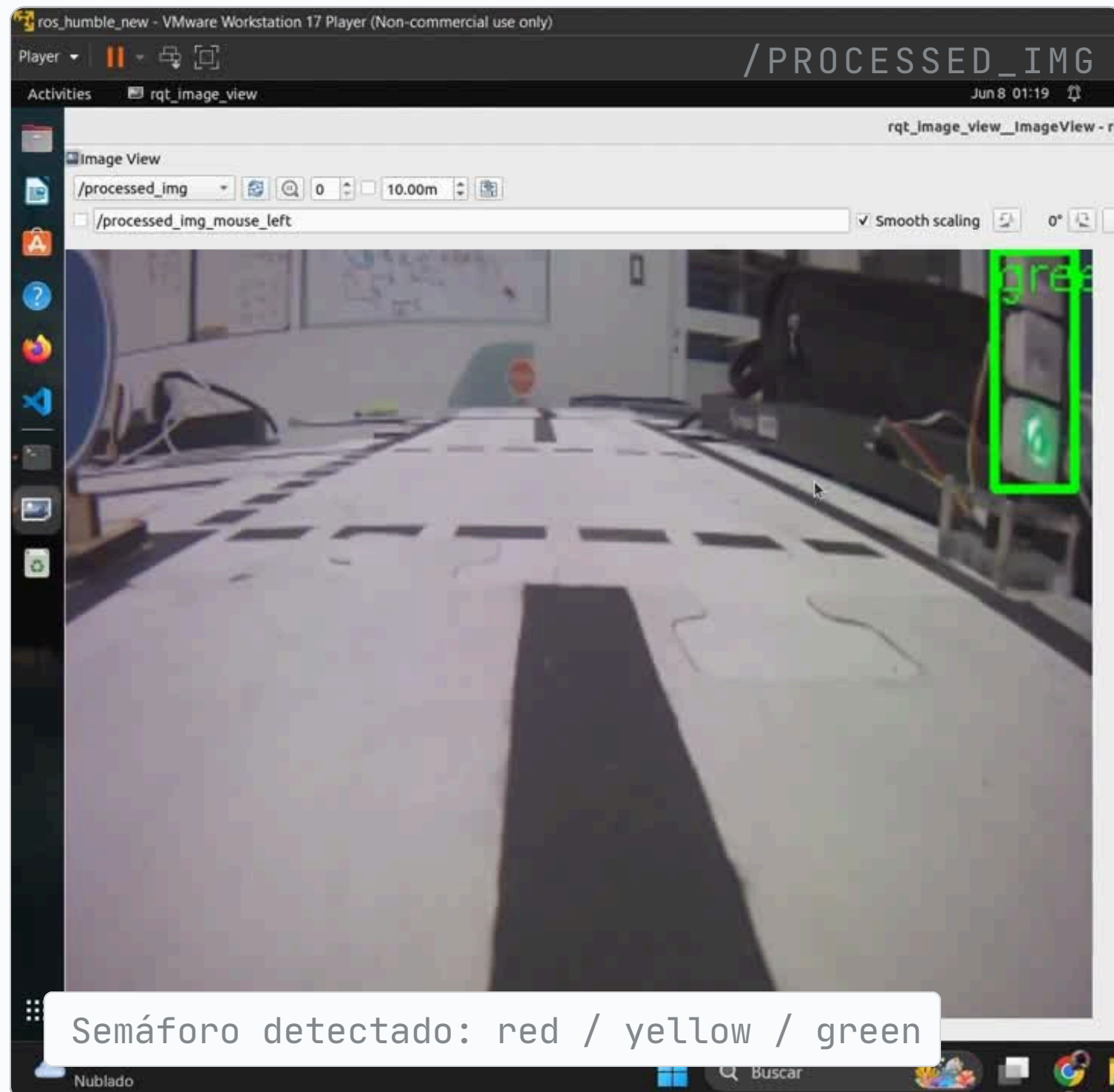
- ROI y umbral**
Se recorta la región de interés y se binariza la imagen para aislar la línea.
- Contorno y error**
Se calcula el desvío del centro de la línea: el `/line_error`.
- Control P**
El controlador corrige con ganancia proporcional para mantener el rumbo.

$$\omega = K_p \times \text{error}$$

`/line_error``/line_detected``/crossing_detected`

Detección de Semáforos

20% de la rúbrica

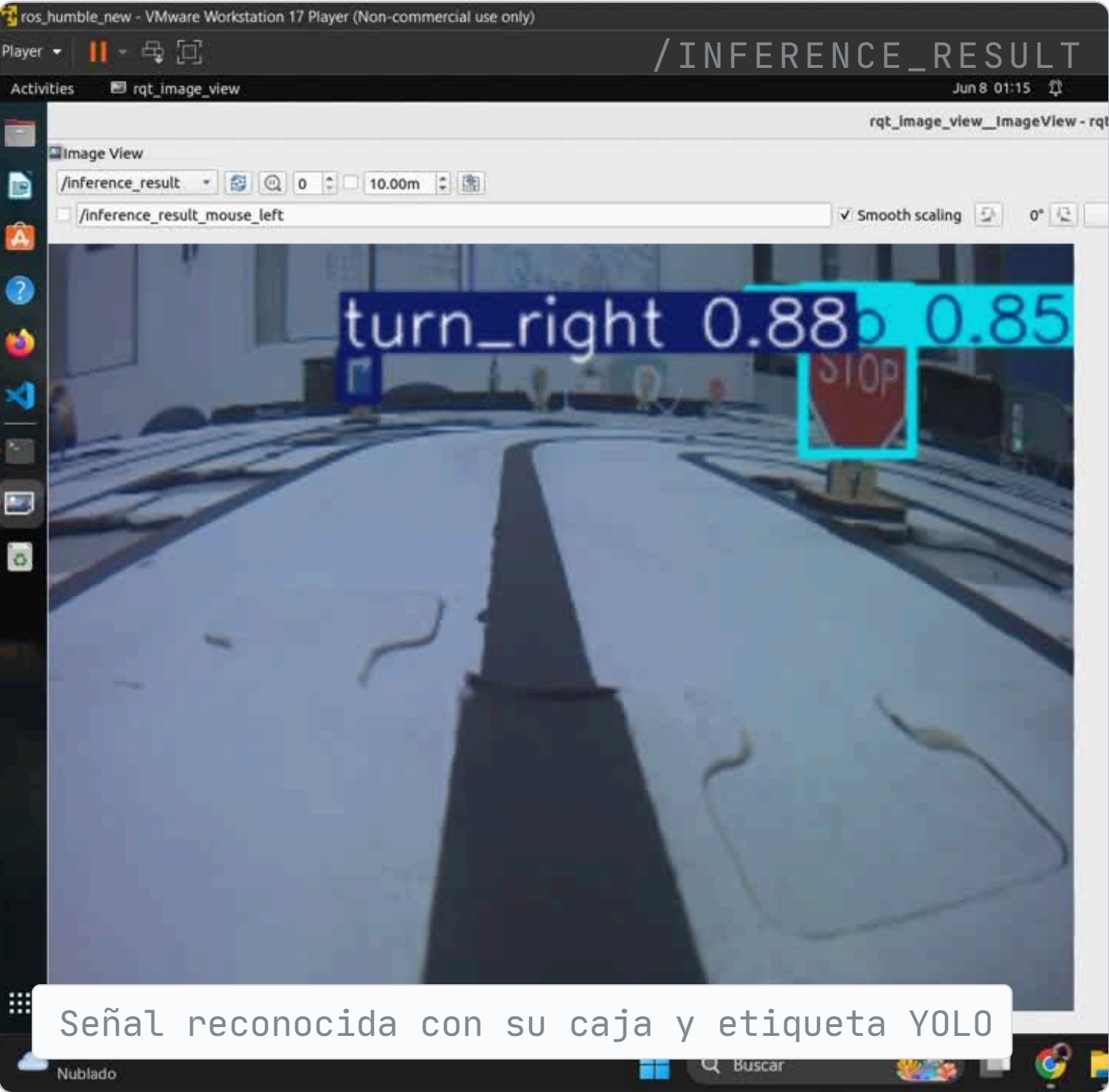


YOLO clasifica el estado y lo publica en `/detected_color`. El controlador reacciona físicamente:

rojo**Se detiene por completo.****amarillo****Reduce velocidad — factor `slow_factor = 0.50`.****verde****Avanza con normalidad.**`/detected_color`

DetECCIÓN de Señales

30% de la rúbrica



YOLO publica la clase en `/detected_sign` y su tamaño en `/sign_area`. Cada señal dispara una acción con tiempo:

SEÑAL	ACCIÓN
Stop	Detención durante 10 s
Give Way	Reduce velocidad 5 s
Work in Progress	Reduce velocidad 10 s
Flechas	Ejecuta el giro antes del cruce

El área mínima evita falsos positivos a distancia.

Red Neuronal — YOLOv8

30% de la rúbrica

**SEÑALES**

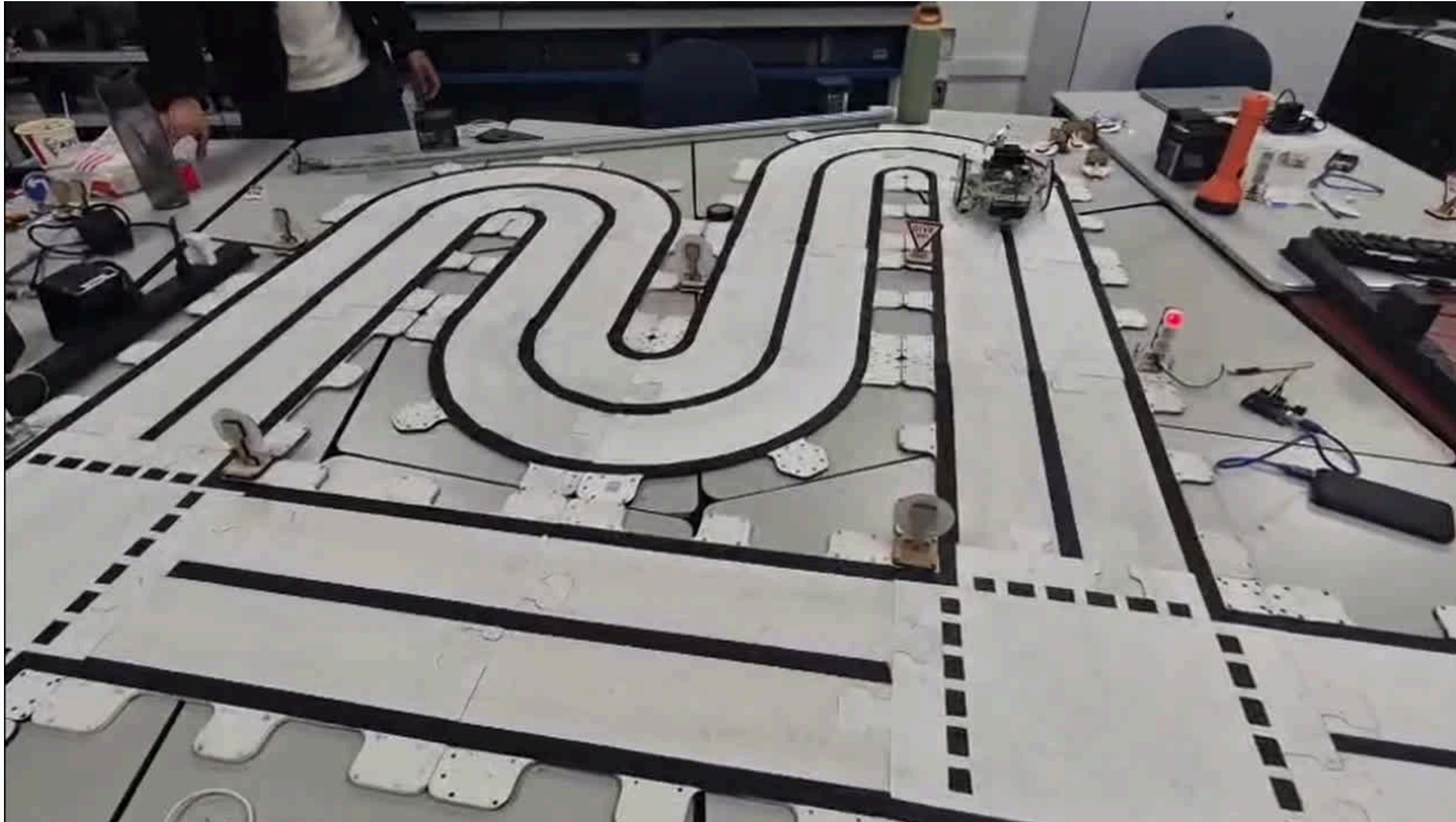
- Modelo: YOLOv8m
- Dataset: 2,103 imágenes · 6 clases
- Entrenamiento: 300 épocas

**SEMÁFOROS**

- Modelo: YOLOv8s · Transfer Learning
- Dataset: 155 imágenes · 4 clases
- Entrenamiento: 50 épocas

`fliplr=0` en señales – voltear una flecha cambia su clase

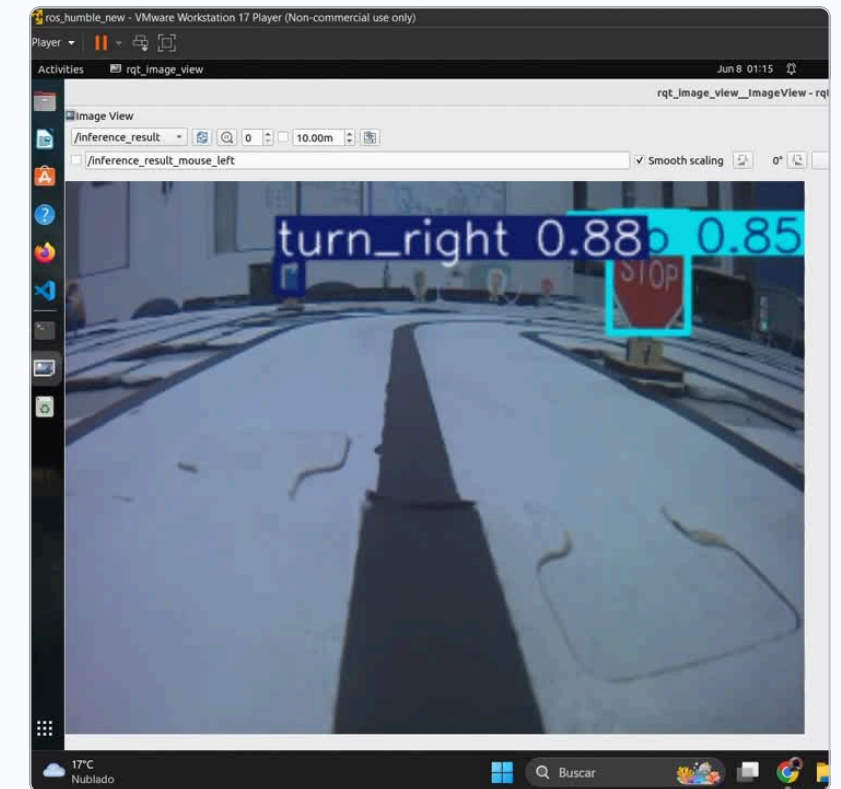
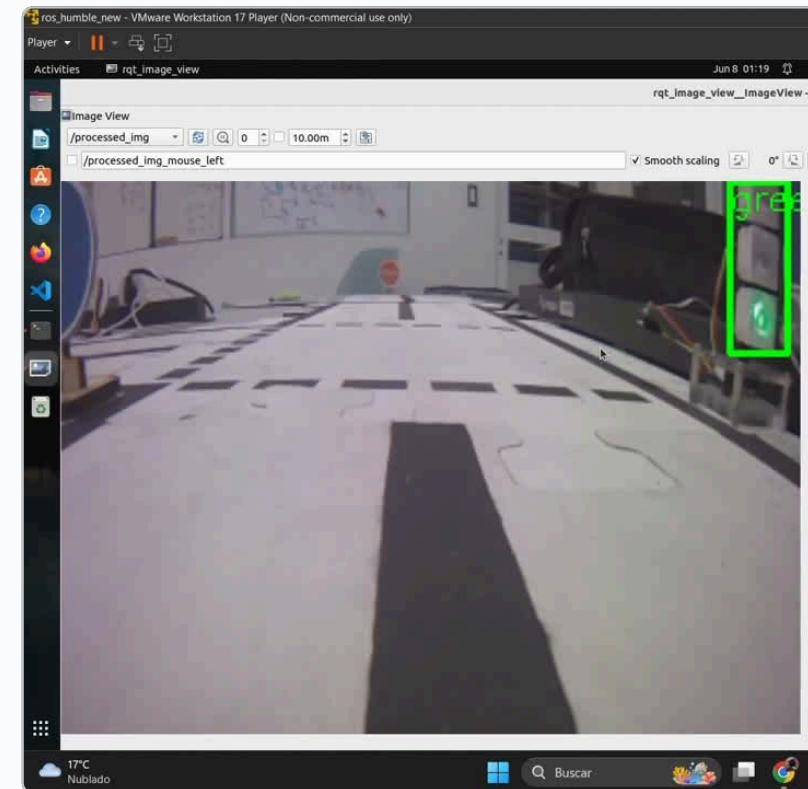
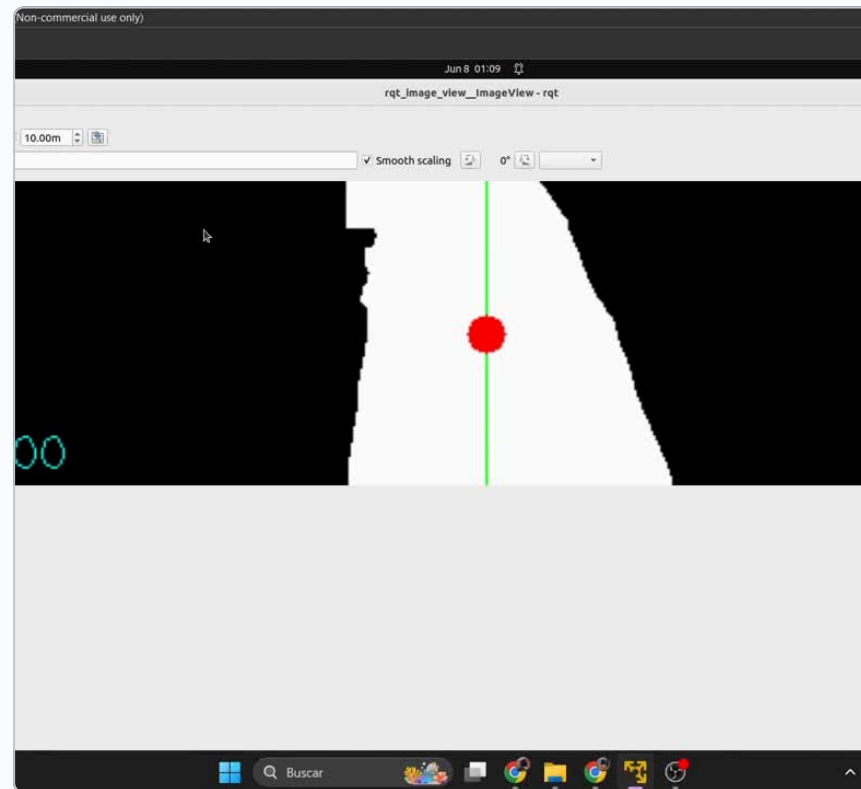
Resultado de las pruebas



Riesgos y Mitigación (Pruebas)

SÍNTOMA	CAUSA PROBABLE	AJUSTE RECOMENDADO
Se sale en curvas	Velocidad alta o ganancia agresiva	Bajar <code>v_recta</code> y <code>v_curva</code> ; revisar <code>kp_high</code> , <code>kd</code> .
No ve la línea	Umbral / ROI / iluminación	Ajustar <code>thresh_val</code> , ROI y luz; revisar <code>/line_stage</code> .
Stop no dura 10 s	Parámetro o estado no activado	Confirmar <code>stop_duration = 10.0</code> y detección de <code>stop</code> .
Give Way / Work no reducen	<code>slow_factor</code> o detección no llega	Confirmar <code>/detected_sign</code> , <code>/sign_area</code> , <code>slow_factor = 0.50</code> .
No detecta semáforo	Área o confianza baja	Revisar <code>light_conf</code> , <code>light_min_area</code> , distancia y luz.
Flecha no ejecuta giro	No se bloquea antes del cruce	Revisar <code>arrow_min_area</code> , <code>arrow_confirm</code> , <code>turn_duration</code> , <code>turn_w</code> .

Conclusion



- Robot autónomo funcional: sigue línea, obedece semáforos y señales sin teleoperación
- Dos redes YOLO entrenadas con transfer learning para detección en tiempo real
- Arquitectura ROS2 modular: cada subsistema opera y se depura de forma independiente

References

Ultralytics. (2023). YOLOv8. <https://ultralytics.com>

Open Robotics. (2022). ROS 2 Humble Hawksbill. <https://docs.ros.org>

OpenCV Team. (2023). OpenCV – Open Source Computer Vision Library.
<https://opencv.org>

Manchester Robotics Ltd. (2025). Puzzlebot Platform Documentation.

Gracias

Seguimiento de línea, detección de semáforos y señales de tráfico.

OMAR • ARTURO • RICARDO



50%

Seguimiento de línea

20%

Semáforos

30%

Señales