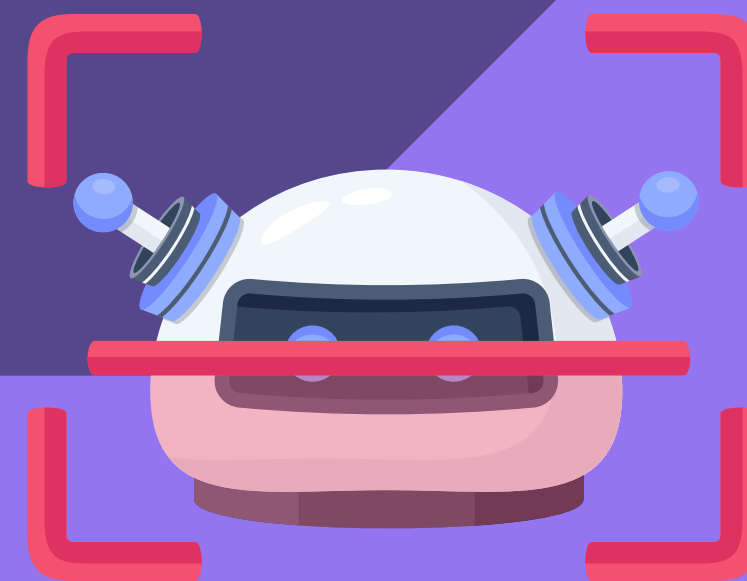


# PROYECTO DE ROBÓTICA

## Integrantes:

- Williams Ccompí Guerrero
- Víctor Ormeño Salazar
- Martín Rosales Figueroa





# CONTENIDO

01

Introducción

02

Objetivos

03

Metodología

04

Modelo  
robótico Final

05

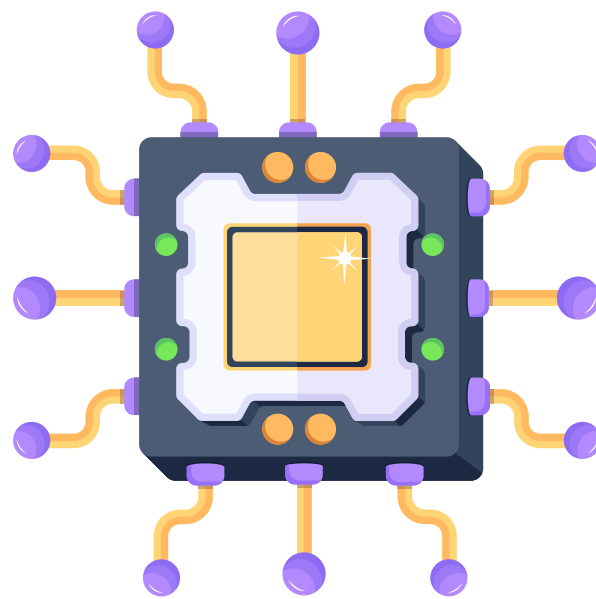
Resultados

06

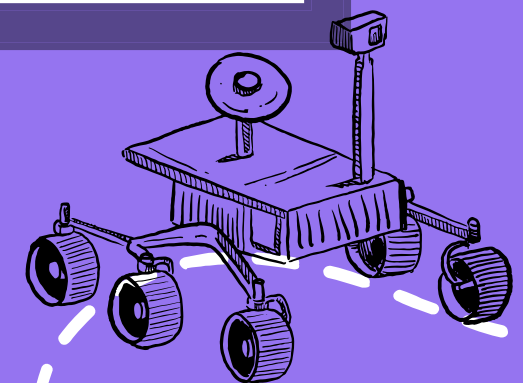
Conclusión

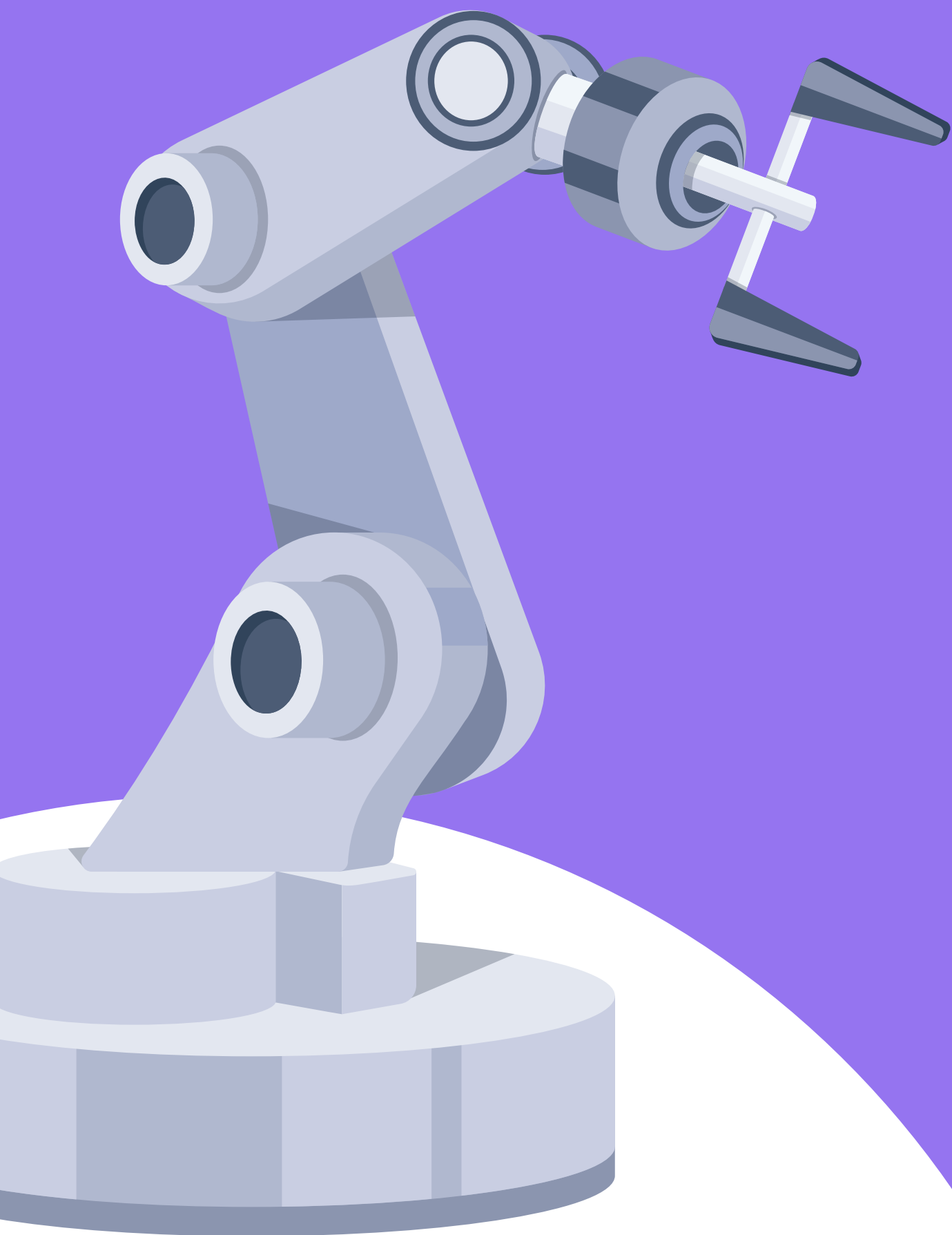


# INTRODUCCIÓN



En el ámbito de la robótica móvil, la capacidad de seguir trayectorias específicas es crucial para aplicaciones en automatización industrial y vehículos autónomos. Este proyecto presenta el diseño e implementación de un carrito autónomo que sigue líneas negras en una carretera simulada mediante procesamiento de imágenes.





# OBJETIVOS DEL PROYECTO

1

Diseñar e implementar un carrito autónomo capaz de seguir líneas negras en una carretera simulada utilizando tecnología basada en Arduino, sensores y una cámara.

2

Evaluar la eficacia del carrito autónomo mediante pruebas en diferentes tipos de carreteras divididas por niveles (básico, intermedio, avanzado) y verificar su capacidad para detectar la carretera, evitar desvíos y maniobrar en espacios difíciles con curvas extrañas.

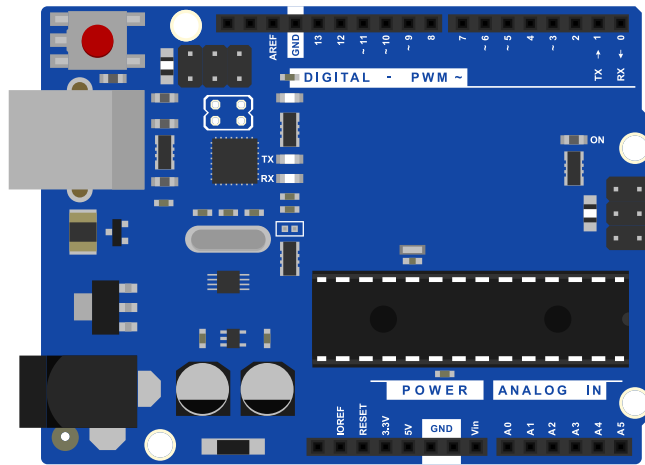
3

Validar la viabilidad y flexibilidad del enfoque propuesto para la navegación autónoma del carrito, destacando su potencial para aplicaciones en el mundo real, como sistemas de transporte autónomos, robots de servicio y aplicaciones industriales.



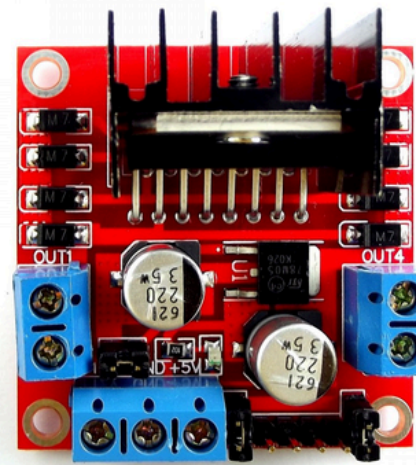
# METODOLOGÍA

## a) Componentes principales



### Arduino UNO

Placa de microcontrolador de código abierto basado en el microchip ATmega 328P.



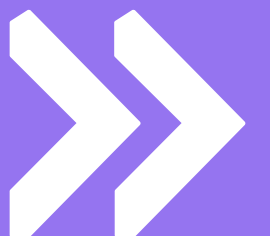
### Puente H

Circuito electrónico que permite controlar la dirección de un motor de corriente continua (CC).

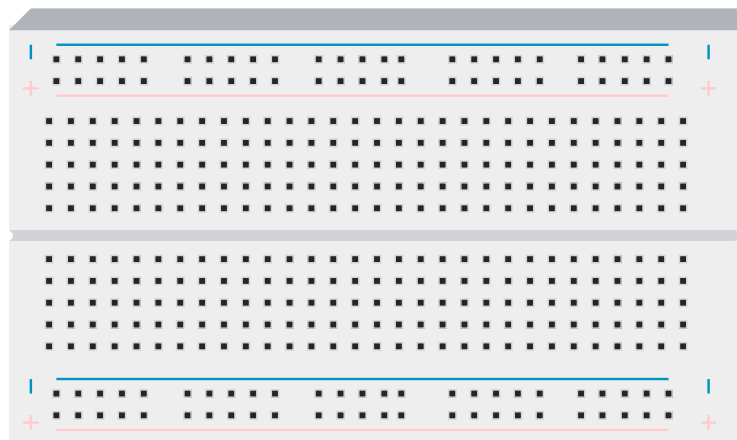


### Cables Hembra y Macho

Son dos tipos de conectores que se utilizan para unir dos o más componentes electrónicos.

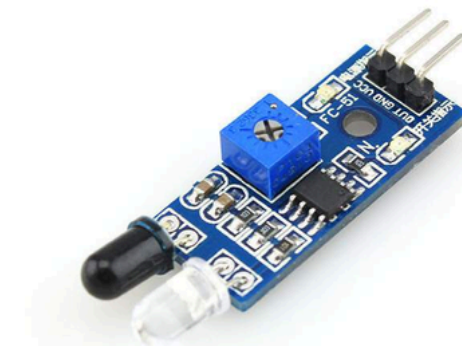


# METODOLOGÍA



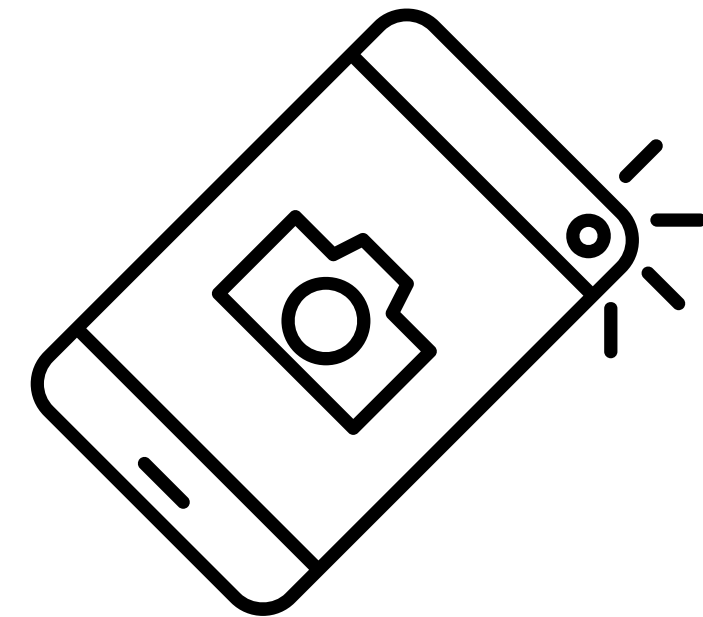
## Protoboard

Placa de pruebas o bread board, es una herramienta fundamental en electrónica que permite construir circuitos de forma temporal y sin necesidad de soldar componentes.



## Sensor IR

Detecta la presencia de objetos a distancias cortas. Tiene diferentes usos como detector de obstáculos en carritos.



## Cámara de Celular

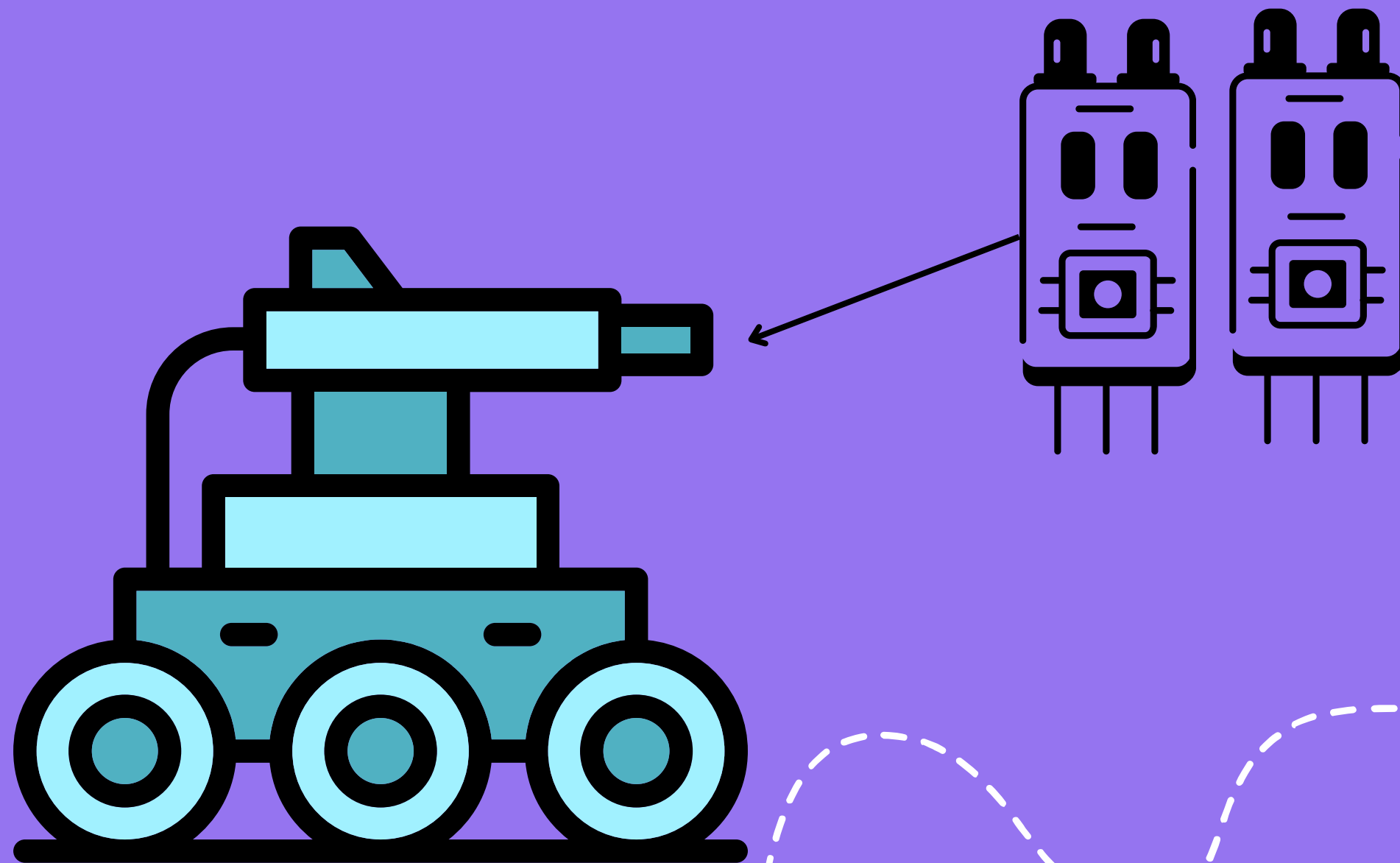
Usada para transmitir imágenes del camino por donde pasará el carrito.



# METODOLOGÍA



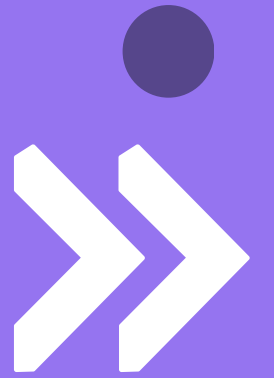
## b) Instalación de los sensores IR



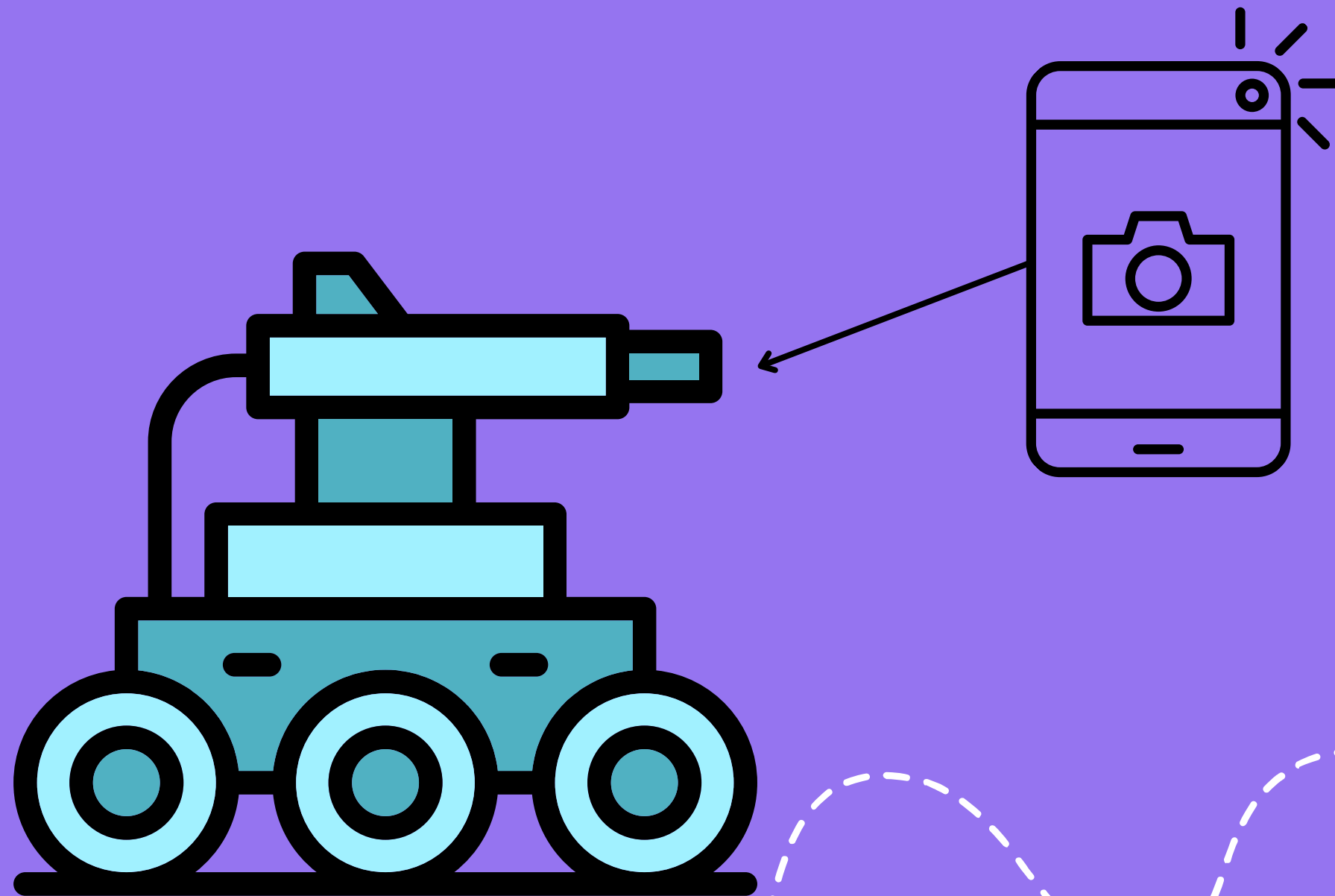
Teniendo en consideración aspectos como el punto ciego y la distancia de detección óptima, se procede a colocar los sensores IR en la parte delantera del carrito. Esta ubicación estratégica permite al sistema detectar con eficacia las líneas de trayectoria y tomar decisiones de navegación en función de la información recopilada por los sensores.



# METODOLOGÍA



## c) Instalación de la cámara

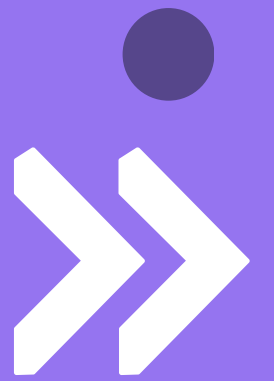


Se adquiere un celular Android de tamaño estándar que servirá como dispositivo de captura de imágenes. Con la ayuda de una montura adecuada, la cámara se instala en la parte superior del carrito, asegurando una posición óptima para la captura de imágenes del entorno circundante.



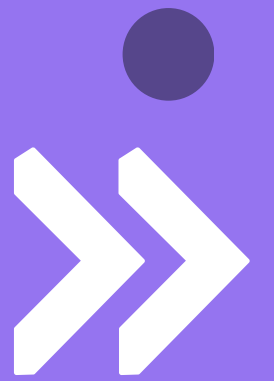
# METODOLOGÍA

## Prueba de cámara



# METODOLOGÍA

## Fragmento de código

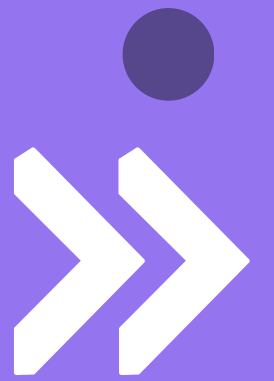


```
1 def extraerCentro(frame):
2     dimH = frame.shape[1]
3     dimV = frame.shape[0]
4     rango = 150
5
6     # Aplicamos un filtro negro:
7     imagenInvert = abs(250 - frame)
8     imagenInvert[imagenInvert < rango] = 0
9     imagenInvert[imagenInvert >= rango] = 250
10
11     # Establecemos la mitad vertical
12     mitadV = dimV // 2
13     pistaInicio = 0
14     pistaFin = dimH
15
16     # Encontrar los bordes de la pista
17     for j in range(dimH):
18         pixel = imagenInvert[mitadV][j]
19         if pixel == 250 and pistaInicio == 0:
20             pistaInicio = j
21         if pixel == 0 and pistaInicio != 0:
22             pistaFin = j-1
23         break
24
25     # Retorno las coordenadas del centro (altura, ancho)
26     puntoH = (pistaFin + pistaInicio) // 2
27     #print("Coordenadas horizontales : ", (puntoH))
28     return (mitadV, puntoH)
```

```
1 def dibujarCruz(frame, puntoH, puntoV, mensaje):
2     grosor = 5
3     s = frame.copy()
4     # linea vertical
5     s[:,puntoH - grosor :puntoH + grosor ,:] = 255
6     s[puntoV - grosor :puntoV + grosor,: ,:] = 255
7
8     ## Dibujar el rango permitido para mantenerse en el centro:
9     # Escribir texto sobre el frame
10    cv2.putText(s,
11               mensaje,
12               (50, 50),
13               cv2.FONT_HERSHEY_SIMPLEX,
14               1,
15               (255, 255, 255),
16               2,
17               cv2.LINE_AA)
18    return s
```

```
1 def decision(puntoActual, d):
2     f,c = d
3     columnas = c//2
4     margen = 20
5     minimo = columnas - margen
6     maximo = columnas + margen
7     if(puntoActual > maximo):
8         # Gira a la derecha :
9         return("Gira a la derecha")
10    elif (puntoActual < minimo):
11        # Gira a la izquierda :
12        return("Gira a la izquierda")
13    else:
14        # Está en el centro
15        return("sigue derecho")
```

# MODELO ROBÓTICO FINAL



# RESULTADOS



01

El carrito autónomo demostró ser capaz de cumplir con las expectativas establecidas, ya que sus sensores IR lograron detectar la carretera y los espacios que no son de color negro, permitiéndole evitar desvíos y mantenerse en la trayectoria deseada.

02

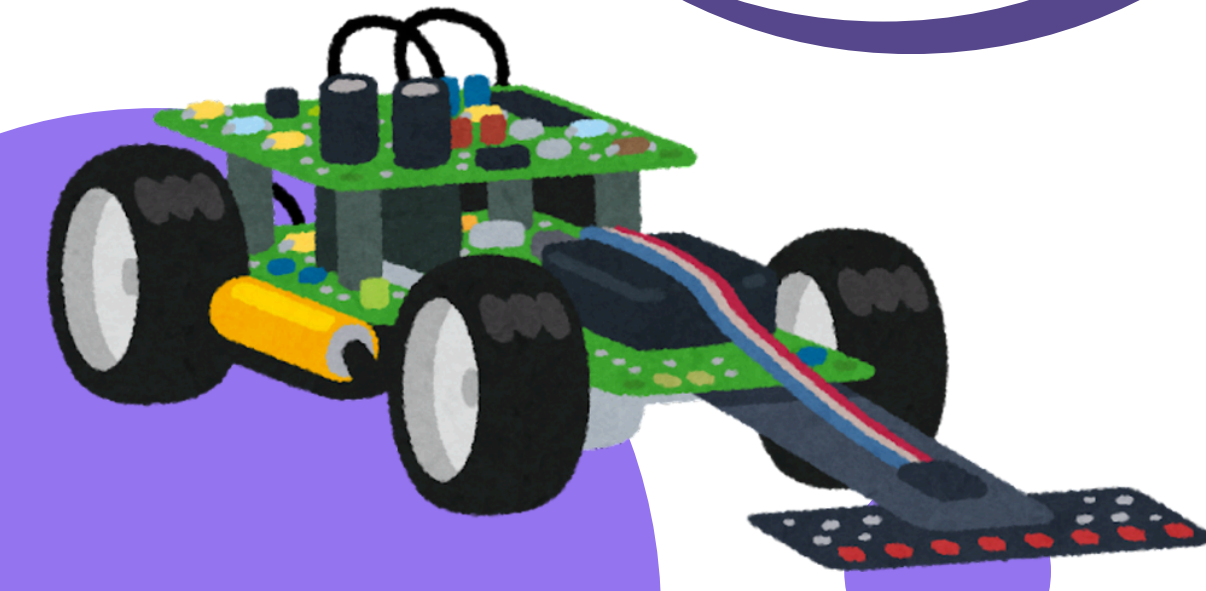
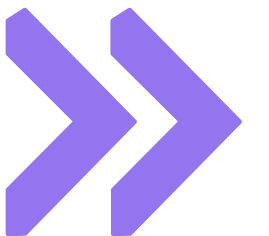
Se observó una excelente velocidad de respuesta y una buena maniobrabilidad por parte del carrito autónomo durante las pruebas realizadas en diferentes tipos de carreteras, lo que indica un rendimiento satisfactorio en situaciones variadas y desafiantes.

03

El sistema implementado en el carrito demostró una eficacia notable al seguir líneas de trayectoria precisas, incluyendo líneas rectas, curvas, zigzags y puntos, lo que refleja la correcta implementación del método descrito en el documento.

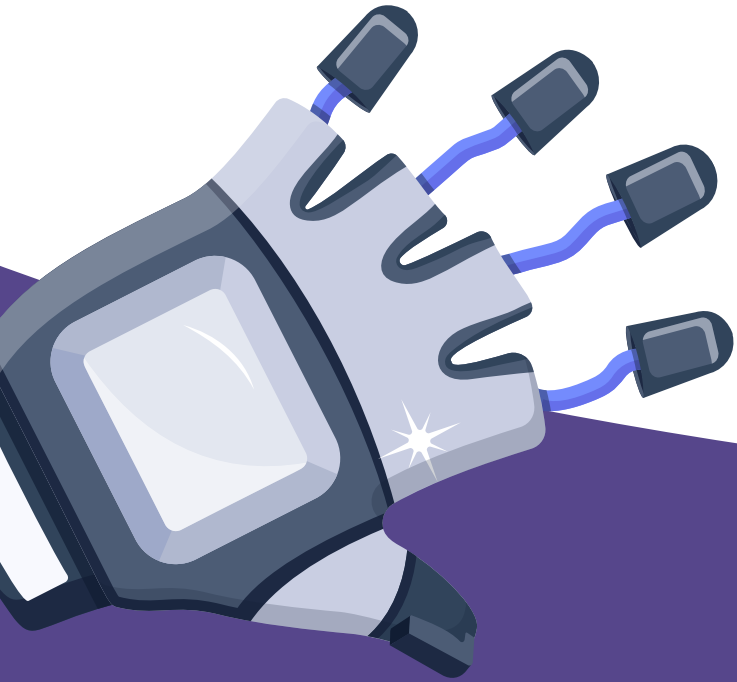
04

Los resultados de las simulaciones realizadas verificaron la fiabilidad y flexibilidad del enfoque propuesto para la navegación autónoma del carrito, mostrando su capacidad para adaptarse a una variedad de configuraciones de trayectoria.



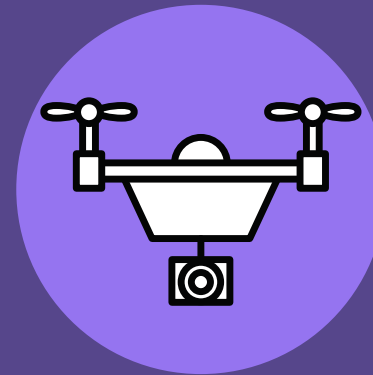


# CONCLUSIONES



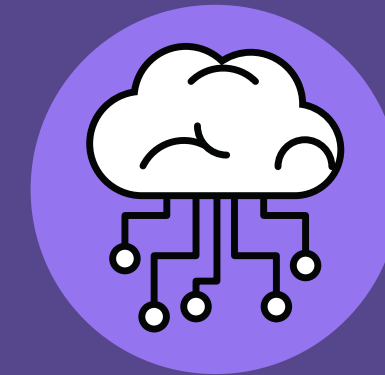
## Trayectorias

El proyecto ha demostrado la viabilidad de un vehículo autónomo capaz de seguir trayectorias predeterminadas mediante la detección de líneas negras a través del procesamiento de imágenes, utilizando tecnología basada en Arduino, sensores y una cámara.



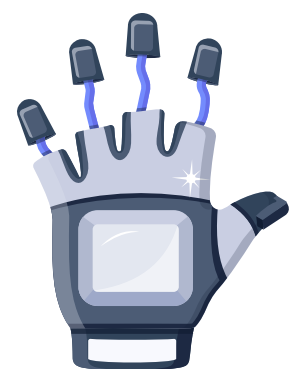
## Navegación Autónoma

El sistema de navegación autónoma implementado en el carrito ha logrado seguir con éxito diversos tipos de líneas de trayectoria, como rectas, curvas, zigzags y puntos, gracias al preprocesamiento de imágenes y una definición precisa de las líneas centrales de trayectoria.



## Simulación

Los resultados de las simulaciones realizadas confirman la fiabilidad y flexibilidad del enfoque propuesto, indicando que el carrito puede adaptarse a una variedad de configuraciones de trayectoria y mostrando su potencial para aplicaciones en sistemas de transporte autónomos, robots de servicio y entornos industriales.



**MUCHAS GRACIAS**

