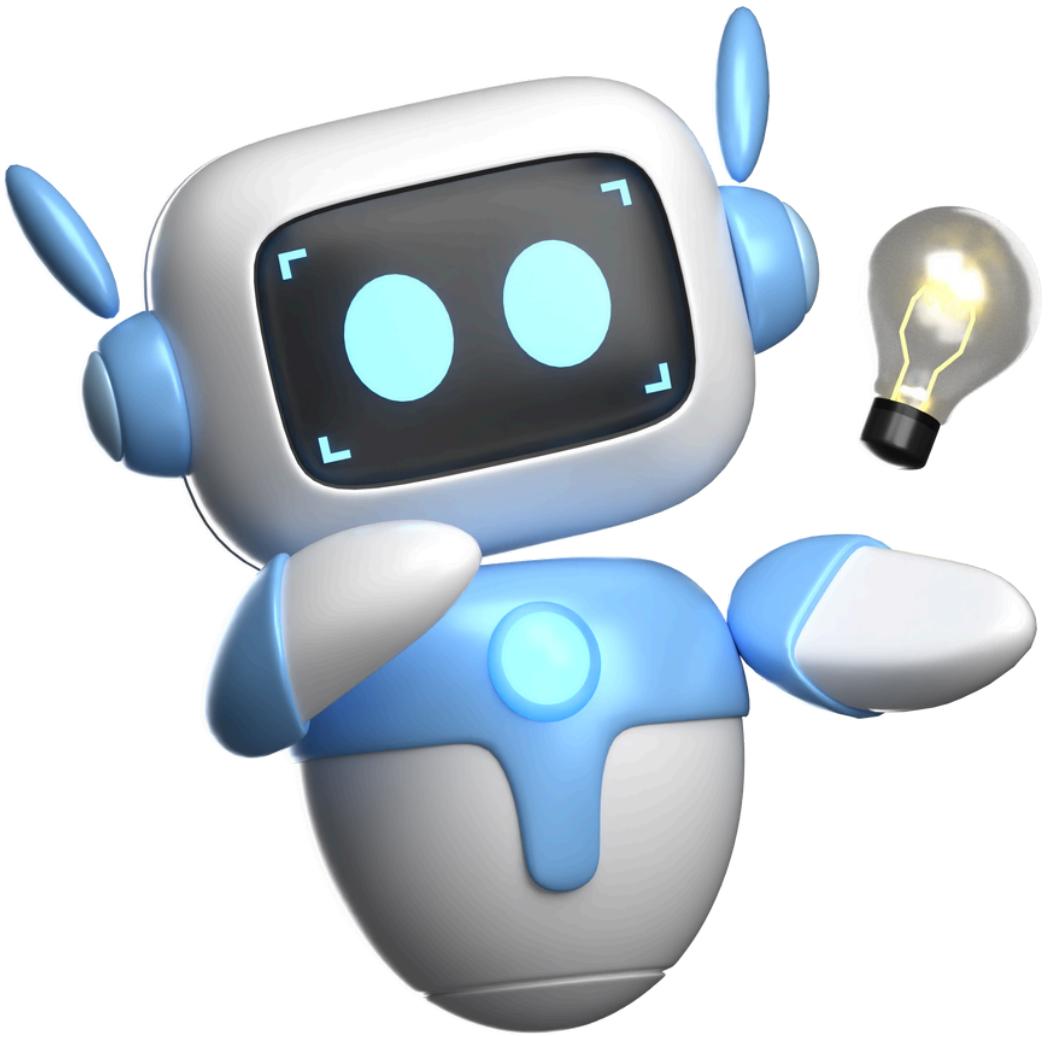




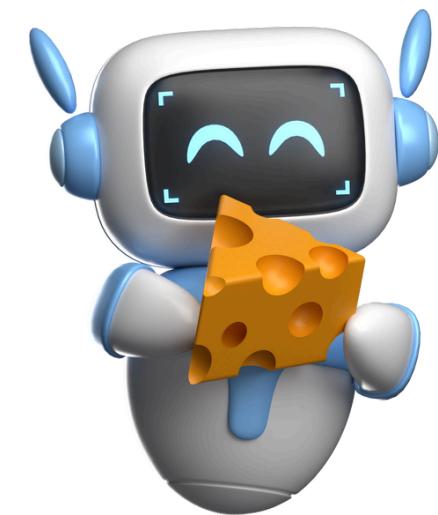
Proyecto final

Laura Rairan, Eduard Meza, Maria
Fernanda Rodriguez

Este proyecto presenta un robot 2WD basado en ESP32, diseñado para la detección de obstáculos y el control avanzado. Integra comunicación segura MQTT con TLS, una API REST para control remoto y una aplicación Android con modos de mando y voz (IA).



Funciones Destacadas

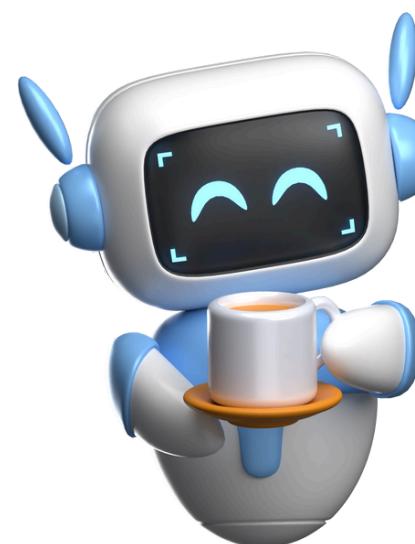
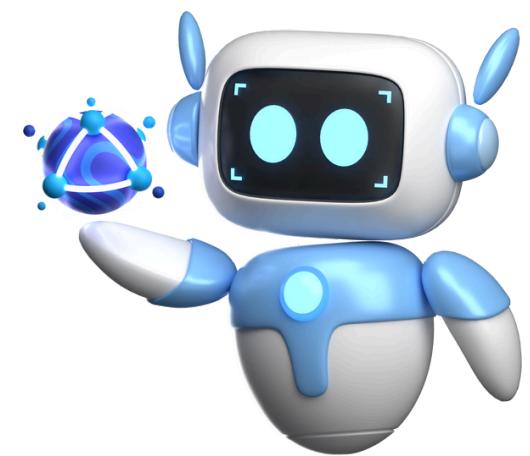


Detección de Obstáculos

Sensor ultrasónico HC-SR04 para frenado automático del robot.

API REST en ESP32

Control remoto mediante endpoints para revisar el estado y mover el robot.

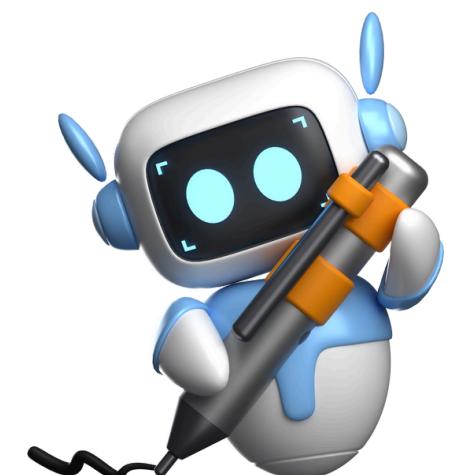


MQTT Seguro con TLS

Publicación de telemetría en un broker en la nube (HiveMQ Cloud o Mosquitto) usando cifrado TLS.

App Android

Control por botones y por voz con integración de IA (OpenAI).

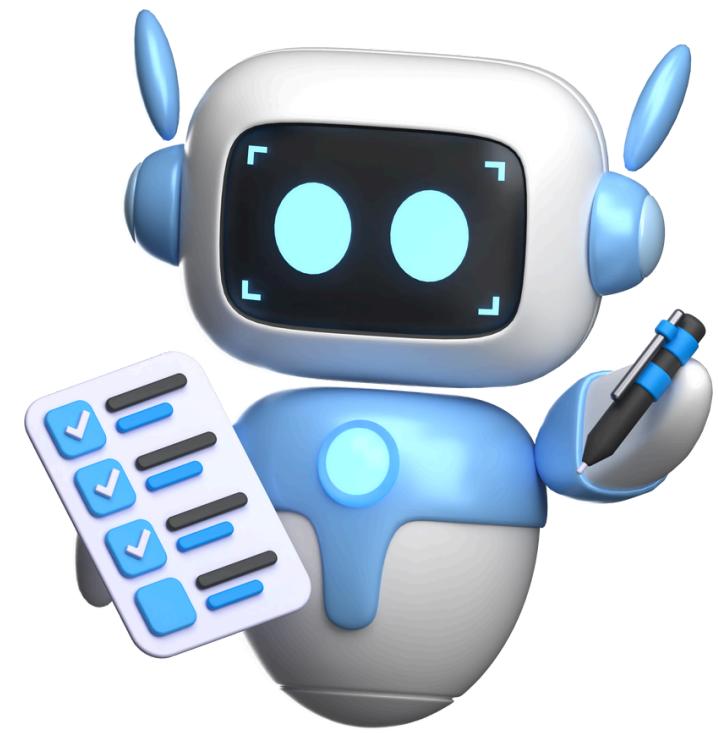


Componentes principales

ESP32 2WD
Control de motores,
lectura del sensor HC-
SR04, servidor HTTP y
cliente MQTT seguro.

Broker MQTT (TLS/SSL)
Servidor en la nube que
recibe la telemetría y
distribuye mensajes entre
el robot y la app.

App Android
Pantalla principal, modo
mando por MQTT/REST
y modo voz con IA.



Modo voz con IA

Transcripción de Audio

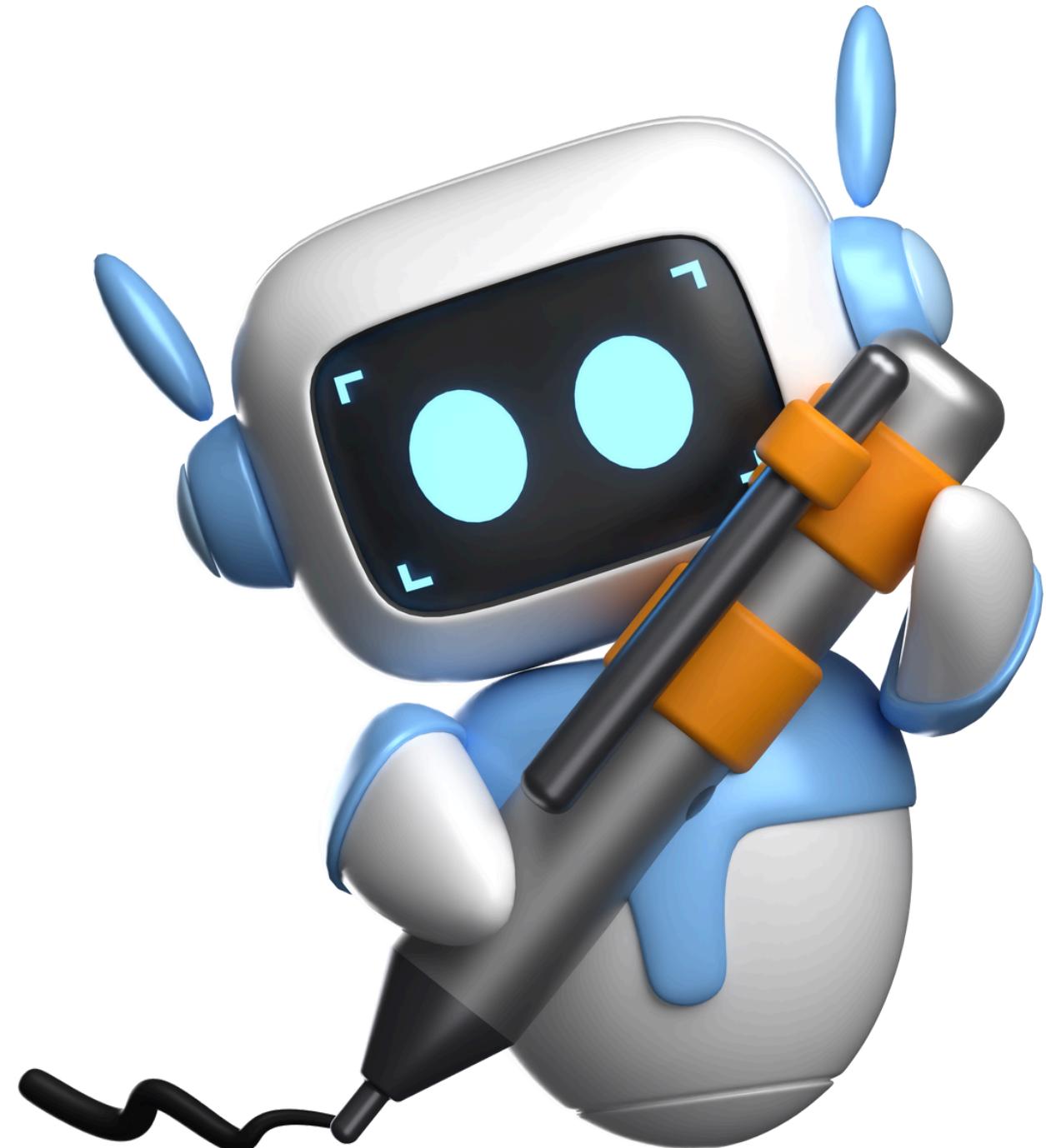
- 1 La app envía la grabación a un modelo de OpenAI para convertir voz en texto.

Generación de Comandos

- 2 Otro modelo de OpenAI transforma ese texto en comandos simples para el robot.

Ejecución y Telemetría

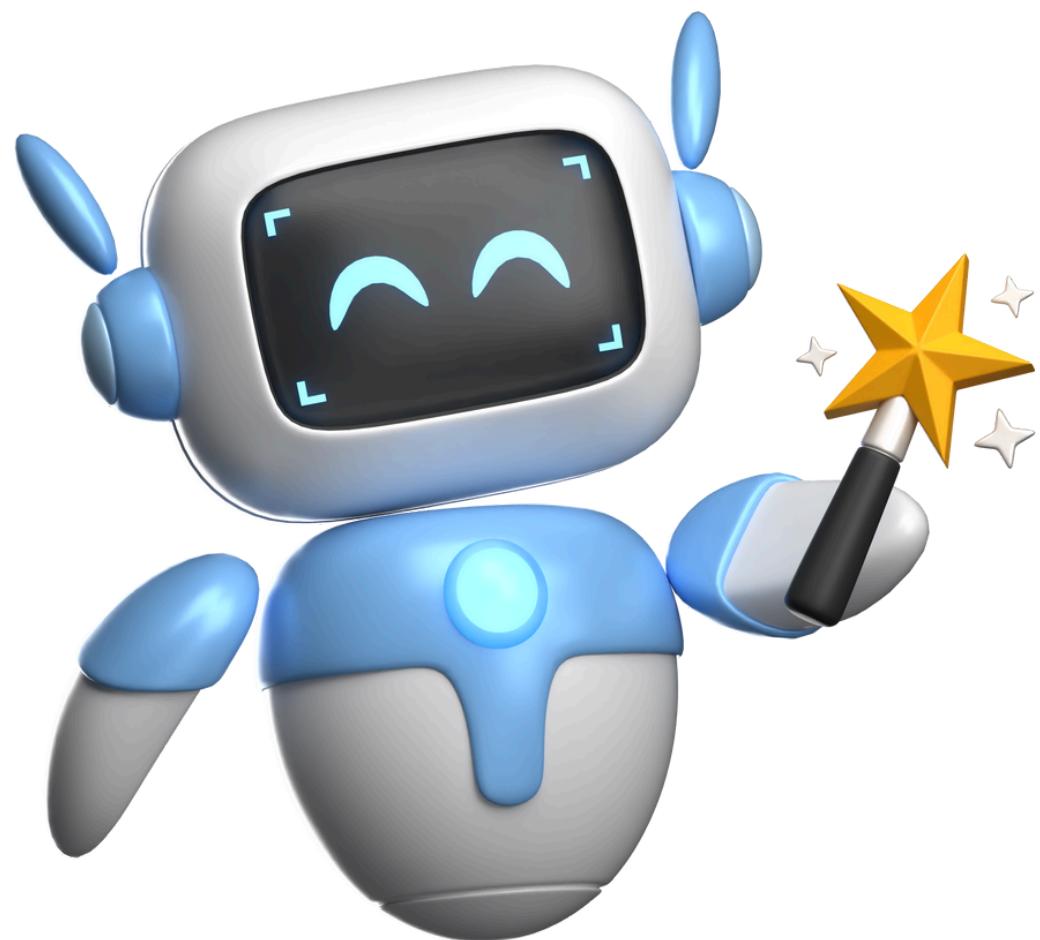
- 3 La app envía los comandos al ESP32 y muestra la telemetría recibida por MQTT.



App Android: mando y modo voz

Sección Modo Mando (ControlActivity)	Sección Modo Voz (VoiceActivity)
Interfaz con botones de dirección (adelante, atrás, izquierda, derecha).	Botón de micrófono para grabar instrucciones habladas.
Botón para comprobar el estado del robot (healthcheck).	Envía el audio a OpenAI para transcribir y convertir a comandos del robot.
Muestra mensajes de conexión, telemetría y comandos enviados.	Ejecuta los comandos uno a uno y muestra la distancia medida en tiempo real.
Usa MQTT para recibir la IP y los datos del robot, y REST para enviar órdenes.	Permite controlar el robot de forma más natural, solo hablando.

Conclusión



Este proyecto nos permitió integrar en un solo sistema varias áreas que normalmente se ven por separado: electrónica, programación embebida, redes, seguridad, aplicaciones móviles e inteligencia artificial. Logramos que un robot 2WD sea capaz de moverse de forma segura, detectar obstáculos, enviar telemetría a la nube y recibir órdenes desde una app Android, incluso por voz usando modelos de OpenAI.