

Sistema de Asistencia de Estacionamiento con ESP32 para Vehículos

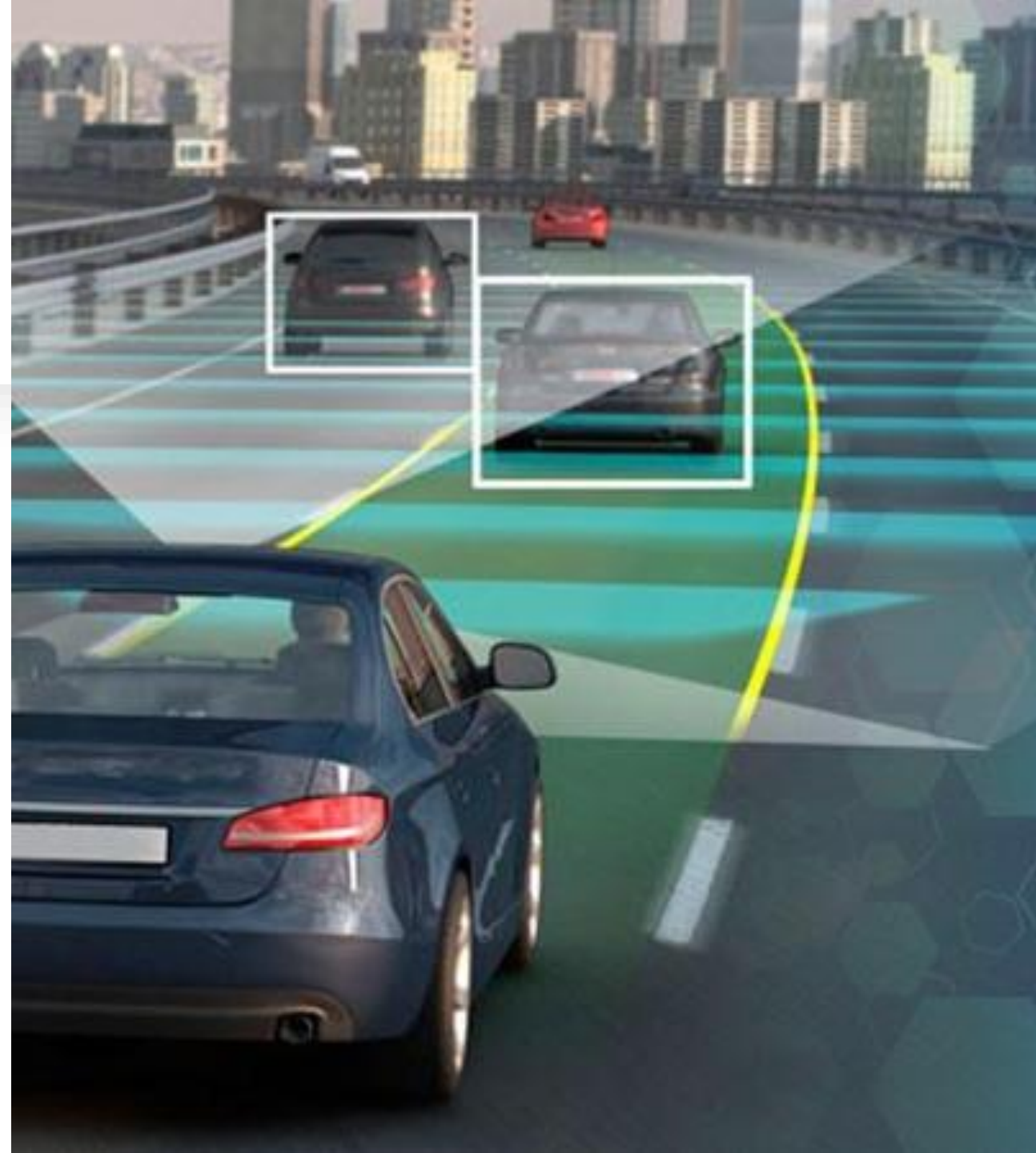
Introducción

- En comunidades urbanas como El Refugio en Tijuana, muchas familias utilizan vehículos sin sensores avanzados de reversa o advertencia.
- Esto incrementa el riesgo de accidentes al estacionarse o maniobrar en zonas angostas con peatones, niños o mascotas cerca.
- Con herramientas asequibles como el ESP32 y sensores ultrasónicos, los estudiantes pueden diseñar soluciones tecnológicas a problemas reales de su entorno.



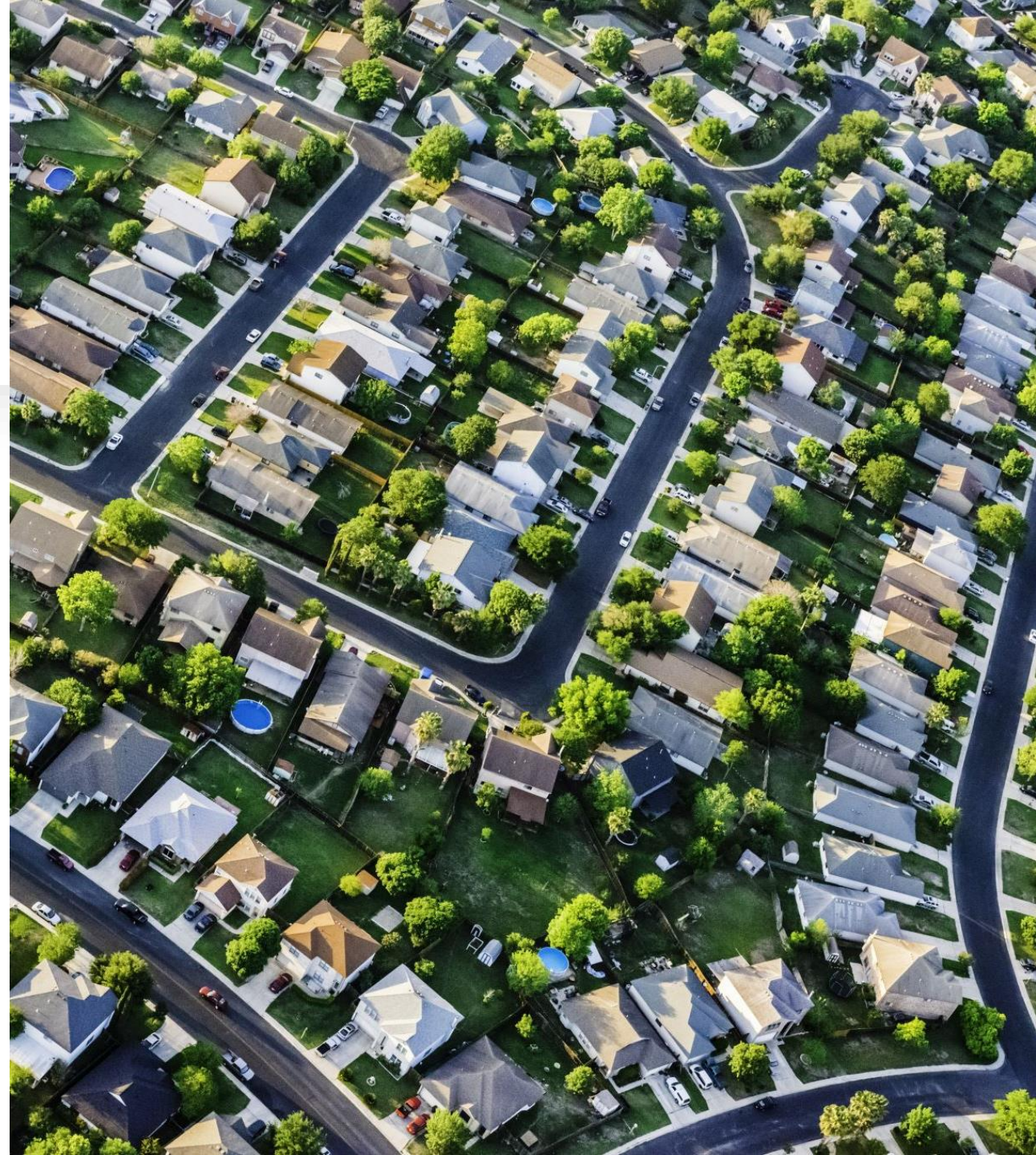
Accidentes viales en Tijuana

- Según datos del INEGI, en 2023 se registraron más de 15,000 accidentes de tránsito terrestre en zonas urbanas y suburbanas en México, con un incremento del 24% respecto al año anterior en Tijuana. ([INEGI, 2025](#))
- Muchos de estos accidentes ocurrieron en vialidades de alta velocidad y zonas residenciales sin infraestructura inteligente, como lo reporta. ([imparcial, 2024](#))



Condiciones urbanas en El Refugio, Tijuana

- El Refugio cuenta con más de 14,000 viviendas, muchas de ellas en zonas de alta densidad y con calles estrechas. ([Youtube,2025](#))
- El crecimiento urbano acelerado en Tijuana ha llevado a la proliferación de asentamientos en terrenos poco aptos para urbanización, como laderas y cañones, lo que incrementa la vulnerabilidad ante accidentes y desastres. ([Pueblos América](#))



9

INDUSTRIA,
INNOVACIÓN E
INFRAESTRUCTURA



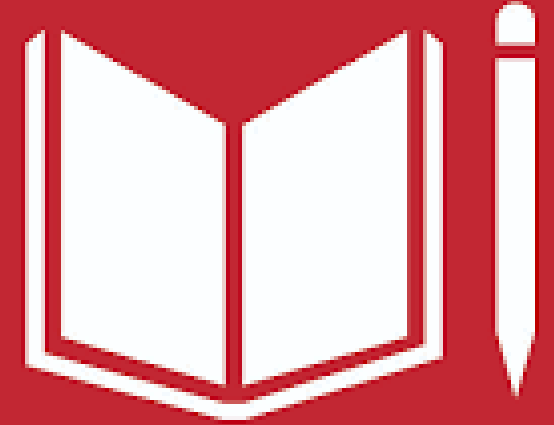
11

CIUDADES Y
COMUNIDADES
SOSTENIBLES



4

EDUCACIÓN
DE CALIDAD



¿A qué ODS contribuye?

- ODS 11 – Ciudades y comunidades sostenibles
 - Mejora la seguridad vial urbana
- ODS 9 – Industria, innovación e infraestructura
 - Promueve soluciones accesibles de ingeniería
- ODS 4 – Educación de calidad
 - Integra habilidades STEM con impacto real

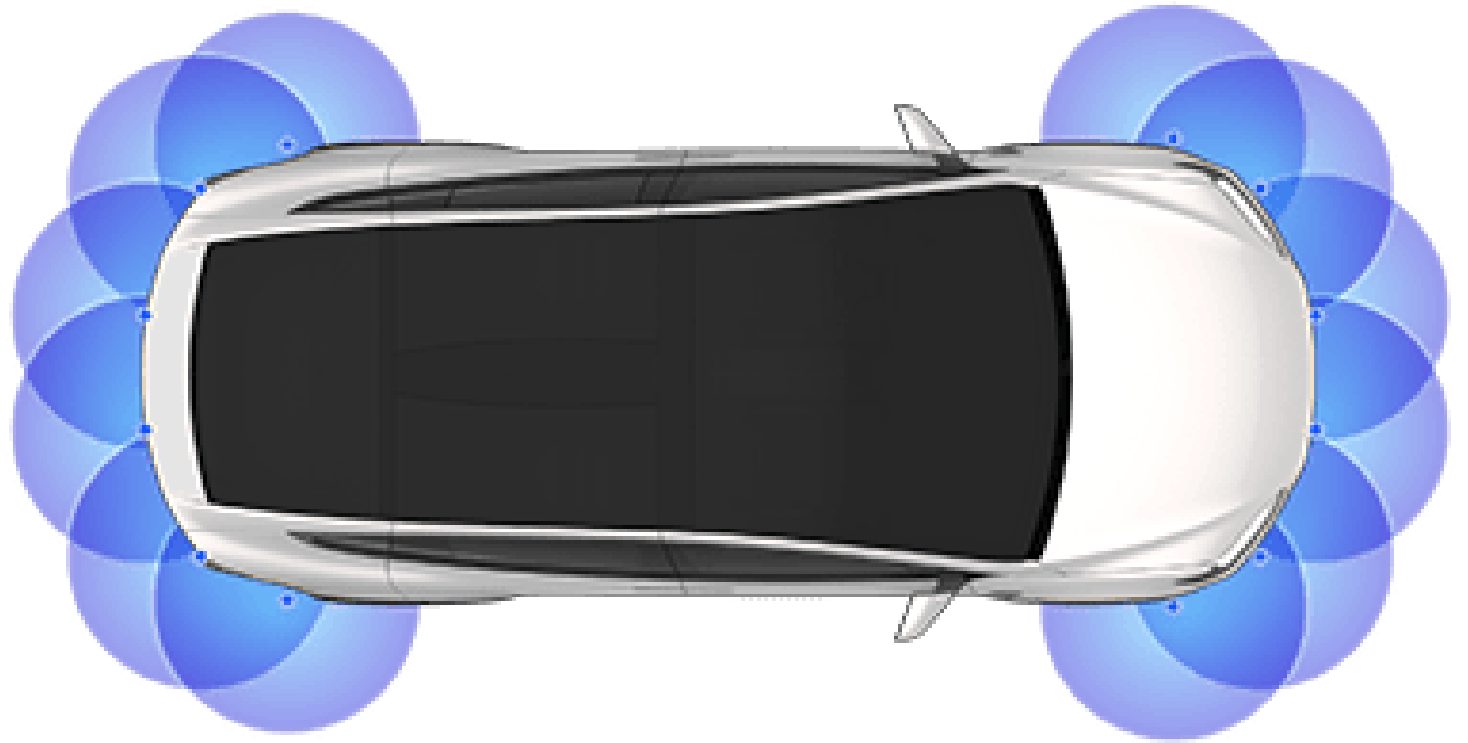


Propuesta de solución

- Desarrollar un **sistema de asistencia de estacionamiento** usando **ESP32, sensor ultrasónico, microservo, LED y App Inventor**, orientado a zonas residenciales como *El Refugio, Tijuana*. El sistema detecta obstáculos a corta distancia, activa alertas visuales, y permite monitoreo desde una app móvil. Busca resolver problemas de visibilidad y prevención de accidentes, aplicando habilidades técnicas con impacto social.

Objetivo general

- Desarrollar un sistema de asistencia para estacionamiento usando un **ESP32**, un **sensor ultrasónico HC-SR04**, un **LED de advertencia**, un **microservo** y una **app móvil** que muestre la distancia en tiempo real, simule la reversa con un **pushbutton** y active una señal visual de peligro.



Hardware



Cantidad	Elemento	Descripción
1	ESP32	Microcontrolador con Wi-Fi integrado para gestionar sensores y conectividad
1	Sensor HC-SR04	Sensor ultrasónico para medir distancia a objetos
1	Microservo SG90	Motor pequeño de rotación controlada para levantar o mover señal visual
1	LED rojo	Indicador de advertencia cuando un objeto se encuentra muy cerca
1	Pushbutton	Botón para activar manualmente el sistema (simula marcha atrás)
1	Resistencia 220Ω	Limitador de corriente para el LED
1	Protoboard	Base de conexiones rápida y reutilizable para el circuito
Varios	Cables Dupont	Cables macho-macho o hembra-macho para conexiones
1	Cable USB	Alimentación y programación del ESP32 desde la PC

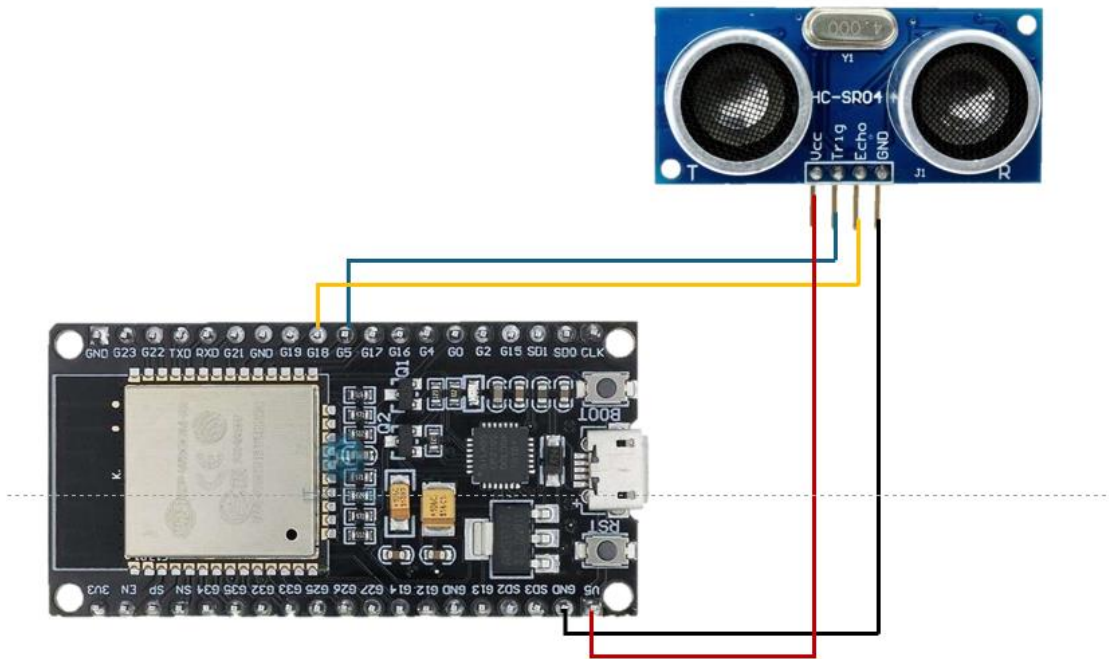
Software



Cantidad	Elemento	Descripción
1	Arduino IDE	Plataforma para programar y cargar el código al ESP32
1	App Inventor 2	Entorno visual para crear la app móvil de monitoreo y control
1	Navegador web	Herramienta para acceder a App Inventor y monitorear el ESP32 vía IP
1	Celular Android	Dispositivo para instalar y probar la app móvil
1	Librería WiFi.h	Librería de Arduino para conexión del ESP32 a red Wi-Fi
1	Librería Servo.h	Librería para controlar el microservo desde el ESP32

Etapa 1: Conexión del Sensor Ultrasonico (HC-SR04)

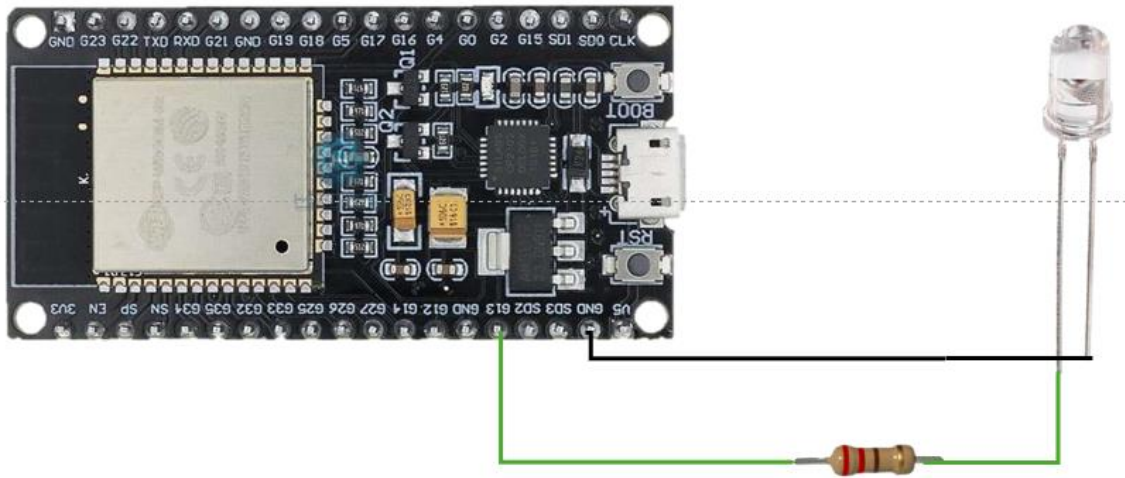
- **Prueba esperada:** Leer distancia por monitor serial y que los valores cambien al acercar la mano.
- [CODIGO 01 SENSOR ULTRASONICO](#)



Pin del Sensor	Conecta a ESP32
VCC	5V
GND	GND
Trig	GPIO 5
Echo	GPIO 18

Etapa 2: Agregar el LED de Advertencia

- **Prueba esperada:** Si la distancia < 30 cm, el LED se enciende. Si > 30 cm, se apaga.
- CODIGO DE PRUEBA: [02 LED](#)



Elemento

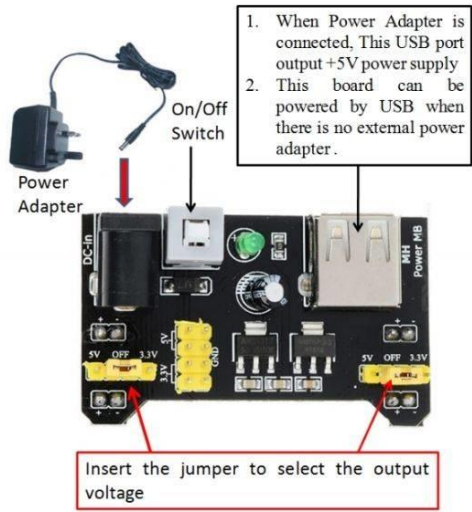
Ánodo (pata larga)
del LED

Cátodo (pata corta)

Conecta a ESP32

GPIO 13 a través de
resistencia de $220\ \Omega$

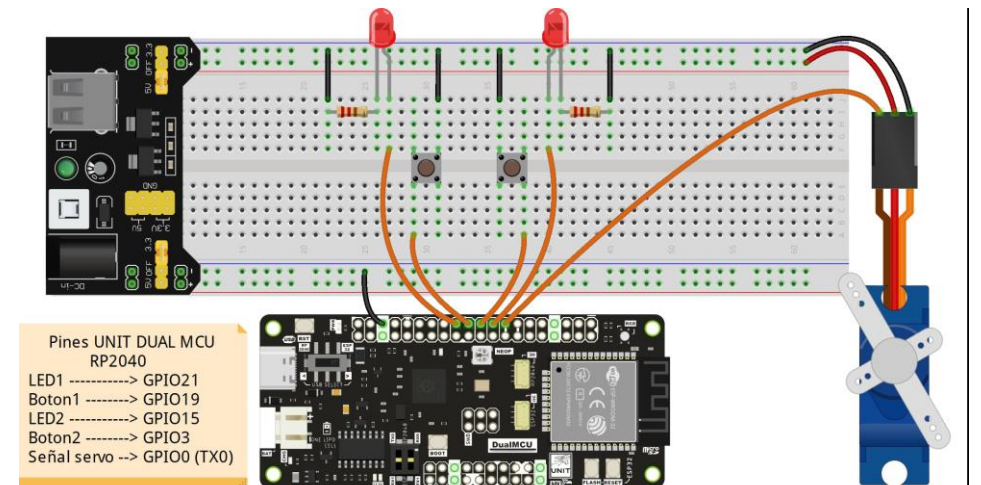
GND



CABLE DEL SERVO	CONECTAR A ESP32
Marrón (GND)	GND
Rojo (VCC)	5V
Naranja (Señal)	GPIO 25

Etapa 3: Integrar el microservo (SG90)

- **Consejo:** Aunque muchos SG90 funcionan con 5V, algunos ESP32 no dan suficiente corriente por el pin 5V. Si se comporta errático, te puedo sugerir usar una fuente externa o capacitores.
- CODIGO PRUEBA: [03 SERVO](#)

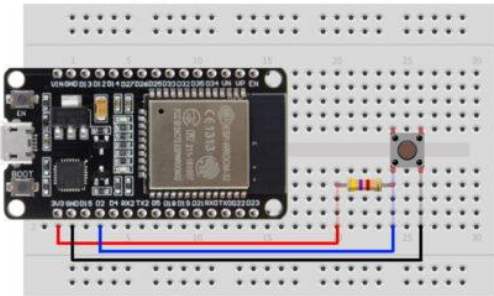


Etapa 4: Pushbutton de activación

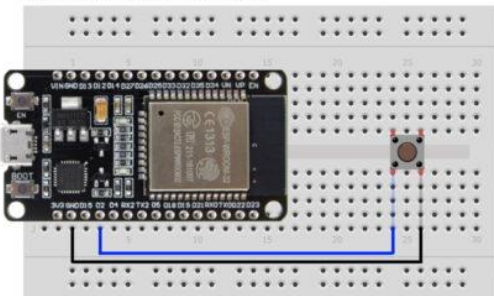
- **PULL-DOWN:** Cuando conectas algo (como voltaje al presionar un botón), cambia a HIGH.
- **PULL-UP:** el pin se conecta internamente a **3.3 V** (HIGH) hasta que tú, por ejemplo con un botón, lo mandas a tierra (LOW).
- Se usará el modo **INPUT_PULLUP** para evitar el uso de resistencias externas. El sistema se activará solo cuando el botón esté presionado (estado LOW).
- CODIGO DE PRUEBA: [04 Push Button](#)

PULL-UP

CON RESISTENCIA EXTERNA

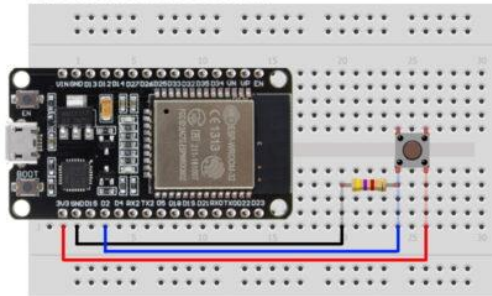


CON RESISTENCIA INTERNA

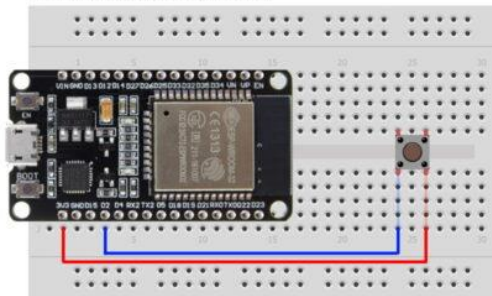


PULL-DOWN

CON RESISTENCIA EXTERNA



CON RESISTENCIA INTERNA



Pin del botón	Conectar a ESP32
Un lado	GPIO 19
Otro lado	GND

Área del sistema	Resultado esperado
Pushbutton	Si está presionado, el sistema se activa. Si no, todo permanece desactivado.
Sensor ultrasónico	Lee y calcula la distancia en centímetros con precisión cada 200 ms.
LED rojo	Se enciende automáticamente si la distancia es menor o igual a 30 cm.
Microservo	Gira a 90° si hay peligro. Regresa a 0° si no lo hay.
Monitor Serial	Muestra continuamente el valor de la distancia leída y el estado actual del sistema.
Servidor HTTP	Entrega respuestas JSON en dos rutas:

Probar el código estructurado

- **Cargar directamente ese código completo** a tu ESP32 desde el Arduino IDE, pero asegúrate de cumplir con estas **4 condiciones clave**:
 - **Seleccionar placa correcta** → En el menú de Arduino IDE → Herramientas → “ESP32 Dev Module” (o el modelo específico que uses)
 - **Puerto correcto seleccionado** → Herramientas → Puerto → Asegúrate de elegir el COM adecuado
 - **Tener las librerías** → Necesitas: WiFi.h (viene con ESP32), y Servo.h (instalada desde el Gestor de Librerías)
 - **Configurar tu red Wi-Fi** → En el código, cambia `const char* ssid = "TU_SSID"` y `password` por los de tu red

CODIGO DE PRUEBA: [05 CODIGO ESTRUCTURADO](#)

Copiar esa IP:



Aplicación web móvil embebida en el ESP32

Etapa 1: ¿Qué hará el sistema?

El ESP32 creará su propia **página web**

La app se abrirá desde el navegador del celular (ej. Chrome)

Desde ahí podrás:

- Tocar un botón “Medir distancia”
- Ver el resultado: “Distancia: XX cm”
- Encender el LED si el valor está debajo del umbral

Todo esto se hace con HTML + JavaScript incrustado en el código del ESP32.

Parte 1 – Probar conexión

Objetivo

- Que el ESP32 se conecte a tu red Wi-Fi
- Que te muestre una IP en el Monitor Serial

Código de Prueba: 06
CONEXIÓN-PRUEBA

Parte 2: Mostrar una página desde el ESP32

- **Objetivo de esta parte**
 - Que el ESP32 cree una **página web interactiva**
 - Que tú puedas abrir esa página desde tu celular
 - Que la página tenga un botón (por ahora sin función real, solo decorativo)
 - Que todo se sirva directamente desde el ESP32 usando su IP
- Código de Prueba: [07 WebServer](#)
- ¿Qué hace este código?
 - Se conecta al Wi-Fi y te muestra su IP como antes
 - Sirve una página que puedes visitar desde tu celular escribiendo <http://192.168.1.67>
 - Al presionar el botón, el mensaje cambiará a: Botón presionado correctamente

Botón Virtual



Presionar

Botón presionado correctamente



Sistema de Medición

Sistema encendido



Activo




Distancia: 29 cm


Parte 3: incluir el sensor ultrasónico


- Botón virtual con dos estados:
 - “Sistema encendido” / “Sistema apagado”
 - Cuando está encendido, mide distancia y la muestra en la página
 - Cuando está apagado, no hace medición y muestra “Sin lectura”
 - Código de prueba: [08 Ultrasónico](#)

Sistema de Medición

Sistema encendido


 Activo


 Distancia: 32 cm


 Zona segura. LED apagado.

Sistema de Medición

Sistema encendido

 Activo

 Distancia: 2 cm




 ¡Distancia peligrosa! LED encendido.

Parte 4: Encender el LED si la distancia es menor a cierto umbral

• Objetivo de esta etapa

- Encender el LED (pin 13) **solo si** la distancia medida es menor, por ejemplo, a 30 cm
- Mientras el sistema esté encendido (desde el botón virtual), el LED reaccionará según cada lectura automática
- Si la distancia es mayor, el LED permanece apagado
- Código de prueba: [09 LED](#)

- **Parte 1: Personalizar colores según estado**

- Vamos a usar colores dinámicos para indicar:
 -  **Peligro** (distancia menor al umbral)
 -  **Zona segura** (distancia mayor)
 -  **Sistema apagado**
- Eso lo lograremos con **JavaScript + estilos CSS + respuestas del ESP32.**

- **Parte 2: Ajustar la interfaz para celular**

- Para que se vea bien en pantalla pequeña:
- Usamos **meta etiquetas de visualización**
- Estilizamos con CSS para que los textos y botones se adapten al ancho del celular
- Código prueba: [10 Adaptado](#)



Sistema de Medición

Sistema encendido

 **Activo**



Distancia: 16 cm




¡Distancia peligrosa! LED encendido.



Sistema de Medición

Sistema encendido

 **Activo**



Distancia: 59 cm



Zona segura. LED apagado.

Mostrar una **tabla** que se actualiza en vivo con cada lectura

- **Paso 1 – Guardar el historial en el navegador**
 - Lo haremos **desde la interfaz web**, para que sea liviano y sin modificar el ESP32 aún. Usamos **JavaScript** para almacenar las lecturas en una lista, y mostrarla en una tabla que crece cada segundo.
- Código de prueba: [11 HISTORIAL](#)

 Sistema de Medición

Sistema encendido

● Activo

📏 Distancia: 51 cm

✅ Zona segura. LED apagado.


📄 Historial de Lecturas

#	Distancia (cm)	Estado	Hora
1	36	✅ Zona segura. LED apagado.	10:36:48 p.m.
2	36	✅ Zona segura. LED apagado.	10:36:49 p.m.
3	36	✅ Zona segura. LED apagado.	10:36:50 p.m.
4	36	✅ Zona segura. LED apagado.	10:36:51 p.m.
5	9	⚠️ ¡Distancia peligrosa! LED encendido.	10:36:52 p.m.
6	10	⚠️ ¡Distancia peligrosa! LED encendido.	10:36:53 p.m.
7	10	⚠️ ¡Distancia peligrosa! LED encendido.	10:36:54 p.m.
8	10	⚠️ ¡Distancia peligrosa! LED encendido.	10:36:55 p.m.
9	7	⚠️ ¡Distancia peligrosa! LED encendido.	10:36:56 p.m.
10	7	⚠️ ¡Distancia peligrosa! LED encendido.	10:36:57 p.m.
11	7	⚠️ ¡Distancia peligrosa! LED encendido.	10:36:58 p.m.

Sensor Web Interactivo con Historial y Exportación

Código de prueba: [12 Exportar](#)

A	B	C	D	E
#	Distancia (cm)	Estado	Hora	
1	50	âœ… Zona s	10:40:20 p.m.	
2	50	âœ… Zona s	10:40:21 p.m.	
3	51	âœ… Zona s	10:40:22 p.m.	
4	27	âšš ĩ, Â;Dista	10:40:23 p.m.	
5	7	âšš ĩ, Â;Dista	10:40:24 p.m.	
6	7	âšš ĩ, Â;Dista	10:40:25 p.m.	
7	7	âšš ĩ, Â;Dista	10:40:26 p.m.	
8	7	âšš ĩ, Â;Dista	10:40:27 p.m.	
9	7	âšš ĩ, Â;Dista	10:40:28 p.m.	
10	7	âšš ĩ, Â;Dista	10:40:29 p.m.	
11	7	âšš ĩ, Â;Dista	10:40:30 p.m.	
12	7	âšš ĩ, Â;Dista	10:40:31 p.m.	
13	7	âšš ĩ, Â;Dista	10:40:32 p.m.	
14	7	âšš ĩ, Â;Dista	10:40:33 p.m.	
15	7	âšš ĩ, Â;Dista	10:40:34 p.m.	
16	7	âšš ĩ, Â;Dista	10:40:35 p.m.	
17	7	âšš ĩ, Â;Dista	10:40:36 p.m.	
18	7	âšš ĩ, Â;Dista	10:40:41 p.m.	
19	7	âšš ĩ, Â;Dista	10:40:41 p.m.	
20	7	âšš ĩ, Â;Dista	10:40:41 p.m.	
21	7	âšš ĩ, Â;Dista	10:40:41 p.m.	
22	7	âšš ĩ, Â;Dista	10:40:42 p.m.	
23	7	âšš ĩ, Â;Dista	10:40:42 p.m.	
24	7	âšš ĩ, Â;Dista	10:40:43 p.m.	
25	7	âšš ĩ, Â;Dista	10:40:44 p.m.	
26	7	âšš ĩ, Â;Dista	10:40:45 p.m.	
27	7	âšš ĩ, Â;Dista	10:40:46 p.m.	

30	7	⚠ ¡Distancia peligrosa! LED encendido.	10:40:49 p.m.
31	7	⚠ ¡Distancia peligrosa! LED encendido.	10:40:50 p.m.
32	7	⚠ ¡Distancia peligrosa! LED encendido.	10:40:51 p.m.
33	7	⚠ ¡Distancia peligrosa! LED encendido.	10:40:52 p.m.
34	7	⚠ ¡Distancia peligrosa! LED encendido.	10:40:53 p.m.
35	7	⚠ ¡Distancia peligrosa! LED encendido.	10:40:54 p.m.
36	7	⚠ ¡Distancia peligrosa! LED encendido.	10:40:55 p.m.
37	7	⚠ ¡Distancia peligrosa! LED encendido.	10:40:56 p.m.
 Exportar a Excel			

Referencias

- INEGI. (2023). Estadísticas de accidentes de tránsito terrestre en zonas urbanas y suburbanas. Instituto Nacional de Estadística y Geografía. Recuperado de <https://www.inegi.org.mx/temas/accidentes/>
- El Imparcial. (2023, 10 de enero). Crecen 24% los accidentes viales en Tijuana: faltan sensores y prevención. Recuperado de <https://www.elimparcial.com/tijuana>
- Baja News. (2025, 1 de febrero). Atlas de riesgo: Colonias en Tijuana con calles angostas y alta densidad poblacional. Recuperado de <https://bajanews.mx/noticias/37689/Industrial>
- News BC. (2025, 1 de febrero). Presentaron el Atlas de Riesgo de Tijuana. Recuperado de <https://www.industrialnewsbc.com/2025/02/01/presentaron-atlas-de-riesgo-de-tijuana/>