

Universidad de Costa Rica

Laboratorio de Transmisión de Datos

Informe 1:
Transmisión de datos entre circuitos integrados

Prof. Ing. Teodoro Willink, M.Sc.

Marco Vásquez Ovares - B17032

Franco Castro Chaves - C01886

Grupo: 01

II - 2025

Índice

1. Presentación	4
2. Equipo de Ingeniería	5
3. Equipo	5
4. Objetivos	6
4.1. Objetivo general	6
4.2. Objetivos específicos	6
5. Metodología	6
6. Implementación de hardware	7
7. Implementación de software	9
7.1. Diagrama de flujo	9
7.2. Definición de pines del sensor ultrasónico	10
7.3. Variables globales	10
7.4. Definición de pines para LEDs indicadores	10
7.5. Configuración inicial (setup())	10
7.6. Generación del pulso ultrasónico	11
7.7. Medición del eco y cálculo de la distancia	11
7.8. Envío de datos al monitor serie	11
7.9. Apagado de LEDs	12
7.10. Encendido de LEDs según rango de distancia	12
7.11. Retardo entre lecturas	12
8. Desempeño obtenido	13
8.1. Resultados	14
9. Presupuesto	19
9.1. Desglose de horas trabajadas	19
9.2. Tarifa de referencia	19
9.3. Estimación de costo de mano de obra	19
9.4. Materiales y recursos	19
9.5. Costo total del proyecto	20
10. Cronograma	21
11. Conclusiones	22
12. Recomendaciones	23
Referencias	24

Índice de figuras

1.	Representación conceptual del prototipo R2-D2 detectando un obstáculo mediante sensor ultrasónico.	4
2.	Secuencia de comunicación digital entre el Arduino UNO y el sensor ultrasónico HC-SR04. El pin TRIG recibe un pulso de $10\mu s$ en nivel alto, lo que provoca la emisión de 8 ciclos a 40kHz. El pin ECHO permanece en alto durante un tiempo proporcional a la distancia, codificando el resultado como un ancho de pulso digital.	7
3.	Diseño y funcionamiento del hardware	8
4.	Diagrama de flujo	9
5.	Montaje físico del sistema experimental.	14
6.	Activación del LED verde para distancias aproximadas de 160 cm.	15
7.	Activación del LED amarillo para distancias cercanas a 110 cm.	16
8.	Activación del LED rojo para distancias de aproximadamente 6,5 cm.	17
9.	Activación del LED azul cuando el sensor registra distancias fuera del rango definido.	18

Índice de tablas

1.	Lista de componentes	5
2.	Horas de trabajo	19
3.	Materiales	19
4.	Cronograma	21

1. Presentación

El equipo **Industrial Automaton**, con sede en los astilleros de Naboo, se complace en presentar este anteproyecto para atender los Términos de Referencia del curso IE0528 - *Laboratorio de Transmisión de Datos*.

Inspirados en la tradición de innovación galáctica, hemos decidido emprender el desarrollo de un prototipo experimental bajo el nombre en clave **R2-D2**. Como primer paso en su evolución tecnológica, nuestro prototipo incorpora un **sistema de percepción basado en ultrasonido**, que le permitirá reconocer su entorno inmediato y ubicarse frente a obstáculos.

Para ello, conectaremos un **microcontrolador de propósito general** (Arduino UNO, basado en el ATmega328P) con un **periférico específico de medición** (sensor ultrasónico HC-SR04). Esta integración ejemplifica la comunicación entre dos circuitos integrados y constituye la base del estudio solicitado.

La información de distancia obtenida por el sensor será procesada en tiempo real por el microcontrolador, que activará un conjunto de indicadores luminosos (LEDs de distintos colores) para representar visualmente el rango de proximidad. De esta manera, el prototipo contará con una “percepción inicial” que servirá como fundamento para futuras mejoras de navegación autónoma.

Con este anteproyecto, **Industrial Automaton** reafirma su compromiso con la exploración, la creatividad y la rigurosidad técnica.

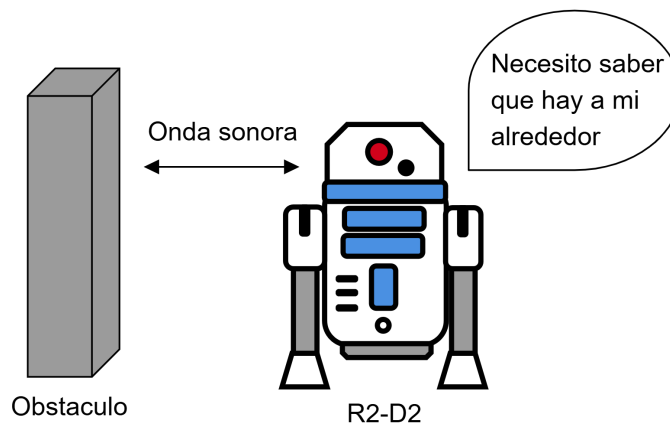


Figura 1: Representación conceptual del prototipo R2-D2 detectando un obstáculo mediante sensor ultrasónico.

2. Equipo de Ingeniería

El desarrollo de este estudio está a cargo de los siguientes profesionales:

- **Ing. Franco Castro Chaves**

Ingeniero eléctrico con énfasis en electrónica y telecomunicaciones. Actualmente cursa la licenciatura en Sistemas de Comunicación. Apasionado por el arte y la música, fuentes que inspiran su creatividad e innovación. Durante su periodo como estudiante participó en el Laboratorio de Investigación Biomédica como investigador y en proyectos interdisciplinarios como ElectrizarTE, donde la ciencia y la tecnología se fusionan con expresiones artísticas. En la actualidad se desempeña como ingeniero en el área de Tecnologías de la Información, y ha desarrollado un creciente interés en el mundo del software, complementando así su formación técnica con nuevas competencias digitales.

- **Ing. Marco Vásquez Ovares**

Ingeniero eléctrico con formación académica en el área de la ingeniería eléctrica y con conocimientos complementarios en programación, sistemas embebidos e ingeniería de computadoras. Mi perfil combina tanto la parte técnica de la electricidad y la electrónica, como el desarrollo de soluciones basadas en microcontroladores y entornos de software. Esta preparación me permite abordar proyectos de forma integral, desde el diseño del hardware hasta la implementación de algoritmos y aplicaciones prácticas en sistemas embebidos.

3. Equipo

Tabla 1: Lista de componentes

Componente	Cantidad	Observaciones
Arduino UNO	1	Unidad de procesamiento (ATmega328P)
Sensor ultrasónico HC-SR04	1	Sensor de distancia por ultrasonido
LED color azul	1	Indicador: fuera de rango
LED color rojo	1	Indicador: rango cercano
LED color amarillo	1	Indicador: rango medio
LED color verde	1	Indicador: rango lejano
Resistencias 220Ω	4	Limitadoras de corriente para LEDs
Protoboard	1	Plataforma de prototipado
Cables de conexión	1 set	Interconexión de componentes

4. Objetivos

4.1. Objetivo general

- Diseñar e implementar un sistema de medición de distancia utilizando un sensor ultrasónico y un microcontrolador Arduino, con el fin de visualizar los resultados y analizar su desempeño en la transmisión y procesamiento de datos.

4.2. Objetivos específicos

- Configurar el sensor ultrasónico HC-SR04 y el Arduino para la medición de distancias.
- Implementar la lógica de adquisición y procesamiento de datos en Arduino.
- Visualizar las mediciones en un dispositivo de salida (LEDs y monitor serial).
- Analizar el desempeño del sistema en cuanto a precisión y estabilidad.

5. Metodología

El desarrollo del proyecto se llevará a cabo en las siguientes etapas:

1. **Investigación preliminar:** Como parte de la etapa de investigación preliminar, se consultaron diversas fuentes teóricas y prácticas sobre el funcionamiento del sensor ultrasónico HC-SR04 y la plataforma Arduino UNO. Entre ellas se incluyen:
 1. Tutorial interactivo en línea sobre el sensor HC-SR04 en Tinkercad: <https://www.tinkercad.com/things/1E3hXimmjCt-sensor-ultrasonico-hc-sr04>
 2. Video explicativo de Arduino en YouTube: <https://www.youtube.com/watch?v=SGHFbNRu1BA&t=384s>
 3. Guía práctica en *Arduino Project Hub*: <https://projecthub.arduino.cc/Isaac100/getting-started-with-the-hc-sr04-ultrasonic-sensor-7cabe1>
 4. Hoja técnica del sensor ultrasónico HC-SR04
 5. Hoja técnica de la tarjeta Arduino UNO R3
2. **Diseño de hardware:** conexión física entre Arduino y el sensor ultrasónico, así como los LEDs indicadores.
3. **Diseño de software:** programación en Arduino IDE para el control del sensor, la medición de distancias y la activación de salidas.
4. **Pruebas de funcionamiento:** validación de la lectura de distancias y comportamiento del sistema.
5. **Evaluación de desempeño:** análisis de la precisión de las mediciones en diferentes condiciones.

6. Implementación de hardware

El diseño de hardware del proyecto se fundamenta en la **integración entre el microcontrolador Arduino UNO (ATmega328P)** y el sensor ultrasónico **HC-SR04**, estableciendo un canal de transmisión de datos digitales basado en señales de control. La comunicación se realiza mediante dos pines: **TRIG** (entrada del sensor) y **ECHO** (salida del sensor).

El proceso inicia cuando el Arduino aplica un pulso digital de **10 μs en nivel alto ($\approx 5\text{ V}$)** sobre el pin **TRIG**, el cual ordena al HC-SR04 emitir una ráfaga de **8 ciclos ultrasónicos a 40 kHz**. Posteriormente, el sensor mantiene su pin **ECHO** en nivel alto durante un intervalo de tiempo proporcional a la duración del eco recibido. Dicho intervalo constituye la **codificación digital de la distancia**, ya que el ancho del pulso en **ECHO** representa el doble del tiempo de vuelo de la onda acústica. El Arduino mide este ancho de pulso en microsegundos y, a través de la relación:

$$d[\text{cm}] = \frac{t_{\text{echo}}[\mu\text{s}]}{58} \quad (1)$$

convierte el tiempo en una magnitud espacial. De esta forma, la transmisión de datos entre ambos circuitos integrados no se efectúa mediante un protocolo de bytes estructurados (como UART o I2C), sino a través de la **duración de una señal digital**, lo que constituye un esquema sencillo pero efectivo de comunicación maestro-esclavo.

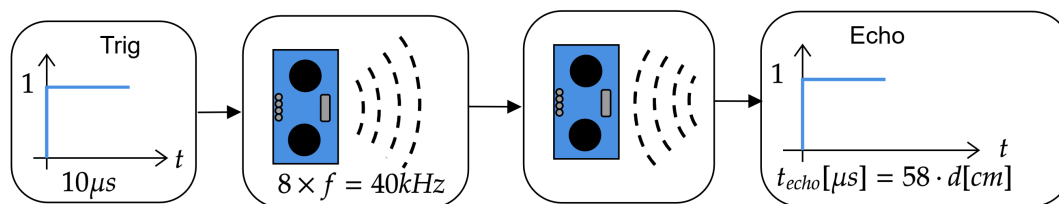


Figura 2: Secuencia de comunicación digital entre el Arduino UNO y el sensor ultrasónico HC-SR04. El pin **TRIG** recibe un pulso de $10\mu\text{s}$ en nivel alto, lo que provoca la emisión de 8 ciclos a 40kHz . El pin **ECHO** permanece en alto durante un tiempo proporcional a la distancia, codificando el resultado como un ancho de pulso digital.

Para la visualización de los resultados, se emplean LEDs indicadores conectados a pines digitales configurados como salidas. Cada LED representa un estado distinto según la distancia medida (por ejemplo: verde para distancia segura, amarillo para precaución y rojo para distancia crítica). Se utilizan resistencias en serie con los LEDs para limitar la corriente y proteger los componentes.

El sistema se alimenta a través del puerto USB del Arduino, lo que simplifica la implementación y reduce el uso de fuentes externas. Este diseño sencillo y modular facilita tanto la comprensión como la posible expansión del proyecto en el futuro, por ejemplo, añadiendo una pantalla LCD o comunicación con otros dispositivos.

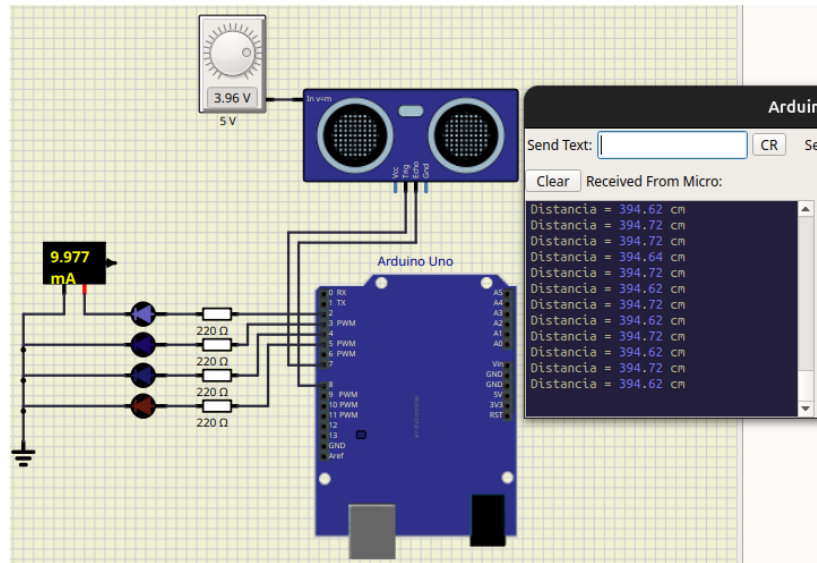


Figura 3: Diseño y funcionamiento del hardware
Elaboración propia

El cálculo de la resistencia se realiza aplicando la ley de Ohm:

$$R = \frac{V_{cc} - V_f}{I_f} \quad (2)$$

donde:

- $V_{cc} = 5V$ es el voltaje de salida de los pines digitales del Arduino.
- $V_f \approx 2,0V$ corresponde al voltaje típico de polarización directa de un LED estándar.
- $I_f \approx 15mA = 0,015A$ es la corriente recomendada para un brillo adecuado sin sobrecargar el microcontrolador.

Sustituyendo valores:

$$R = \frac{5V - 2V}{0,015A} \quad (3)$$

$$R \approx 200\Omega \quad (4)$$

El valor comercial más cercano es **220 Ω**, que proporciona una corriente ligeramente menor ($\approx 13,6mA$), garantizando brillo suficiente y mayor seguridad tanto para el LED como para la salida del microcontrolador.

7. Implementación de software

7.1. Diagrama de flujo

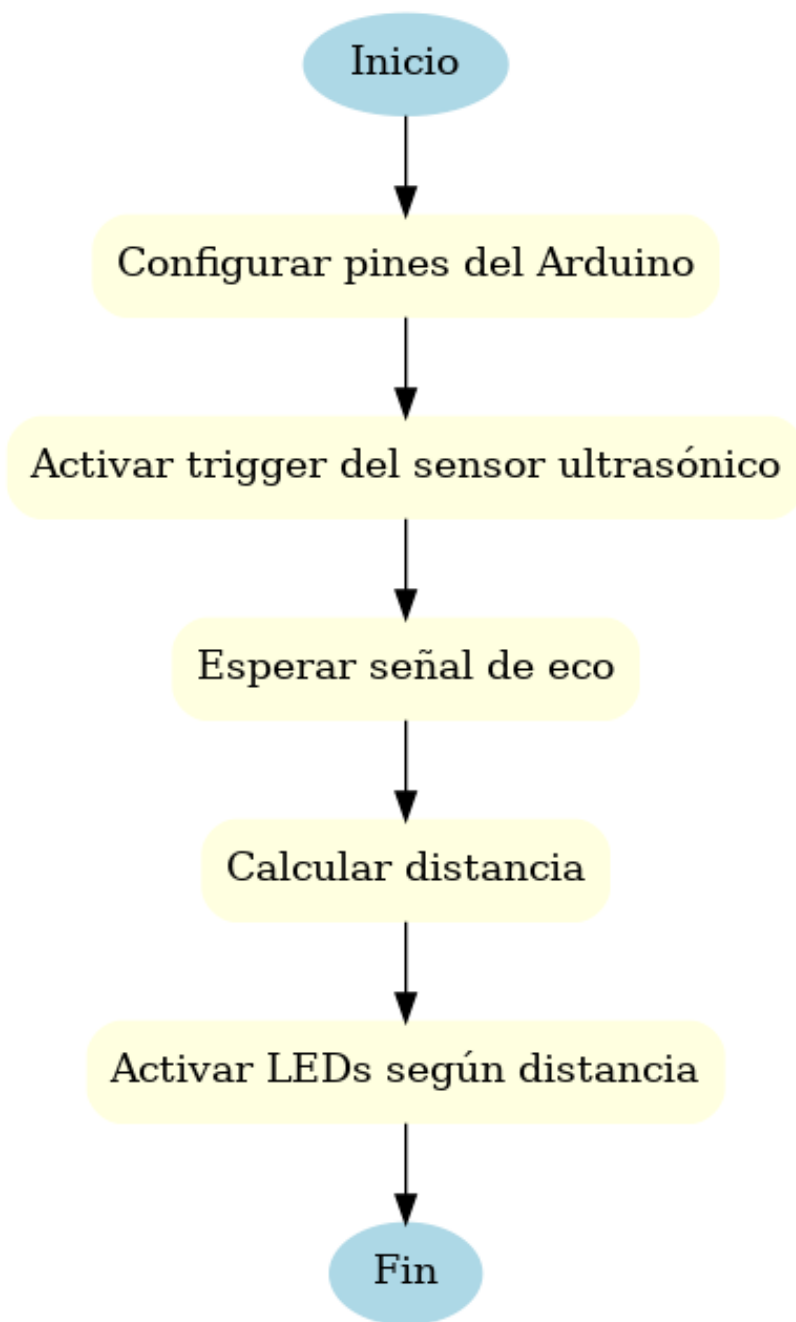


Figura 4: Diagrama de flujo
Elaboración propia

7.2. Definición de pines del sensor ultrasónico

```
const int trigger = 7;    // Pin digital para enviar el pulso (Trigger)
const int echo = 8;       // Pin digital para recibir el eco (Echo)
```

Estos pines definen la conexión con el sensor ultrasónico HC-SR04:

- **trigger**: se usa para enviar el pulso ultrasónico.
- **echo**: mide el tiempo que tarda en volver la señal reflejada.

7.3. Variables globales

```
float dist;    // Variable que almacena la distancia medida en cm
```

- **dist**: guarda el valor calculado de la distancia entre el sensor y un objeto en centímetros.

7.4. Definición de pines para LEDs indicadores

```
const int LED_AZUL = 2;
const int LED_ROJO = 3;
const int LED_AMARILLO = 4;
const int LED_VERDE = 5;
```

Cada LED se conecta a un pin del Arduino:

- Azul → fuera de rango o sin detección.
- Rojo → distancia corta (<30 cm).
- Amarillo → rango medio (30–150 cm).
- Verde → rango lejano (150–310 cm).

7.5. Configuración inicial (setup())

```
void setup() {
  Serial.begin(9600);

  pinMode(trigger, OUTPUT);
  pinMode(echo, INPUT);

  pinMode(LED_AZUL, OUTPUT);
  pinMode(LED_ROJO, OUTPUT);
  pinMode(LED_AMARILLO, OUTPUT);
  pinMode(LED_VERDE, OUTPUT);
}
```

- Se inicia la comunicación serial a 9600 baudios para mostrar datos en el monitor serie.
- Se configuran los pines del sensor (`trigger` como salida y `echo` como entrada).
- Los pines de los LEDs se configuran como salidas digitales.

7.6. Generación del pulso ultrasónico

```
digitalWrite(trigger, LOW);  
delayMicroseconds(5);  
digitalWrite(trigger, HIGH);  
delayMicroseconds(10);  
digitalWrite(trigger, LOW);
```

Se envía un pulso corto al pin `trigger`:

- Bajo \rightarrow espera inicial.
- Alto (10 μ s) \rightarrow dispara el pulso ultrasónico.
- Bajo \rightarrow finaliza el pulso.

7.7. Medición del eco y cálculo de la distancia

```
dist = pulseIn(echo, HIGH);  
dist = dist / 58;
```

- `pulseIn(echo, HIGH)` mide el tiempo (en microsegundos) que tarda la señal en regresar.
- Se divide entre 58 para convertirlo aproximadamente a centímetros, según la velocidad del sonido.

7.8. Envío de datos al monitor serie

```
Serial.print("Distancia = ");  
Serial.print(dist);  
Serial.println(" cm");
```

Muestra en pantalla la distancia medida, útil para verificar valores en la PC.

7.9. Apagado de LEDs

```
digitalWrite(LED_AZUL, LOW);  
digitalWrite(LED_ROJO, LOW);  
digitalWrite(LED_AMARILLO, LOW);  
digitalWrite(LED_VERDE, LOW);
```

Antes de decidir qué LED encender, todos se apagan para evitar que queden varios prendidos al mismo tiempo.

7.10. Encendido de LEDs según rango de distancia

```
if (dist == 0 || dist > 310) {  
    digitalWrite(LED_AZUL, HIGH);  
}  
else if (dist < 30) {  
    digitalWrite(LED_ROJO, HIGH);  
}  
else if (dist >= 30 && dist <= 150) {  
    digitalWrite(LED_AMARILLO, HIGH);  
}  
else if (dist > 150 && dist <= 310) {  
    digitalWrite(LED_VERDE, HIGH);  
}
```

Dependiendo del valor de `dist`, se enciende un solo LED:

- Azul → fuera de rango o sin detección.
- Rojo → muy cerca.
- Amarillo → distancia media.
- Verde → distancia lejana.

7.11. Retardo entre lecturas

```
delay(200);
```

Introduce una pausa de 200 milisegundos para estabilizar las lecturas y evitar parpadeo excesivo de los LEDs.

8. Desempeño obtenido

En el desarrollo de sistemas embebidos y aplicaciones de monitoreo mediante microcontroladores, uno de los aspectos fundamentales es la forma en que los dispositivos periféricos, como sensores o actuadores, se comunican con la unidad de control principal. En este proyecto, se implementa un sistema basado en un Arduino UNO y un sensor analógico, donde la transmisión de información se realiza a través de un protocolo de comunicación de tipo analógico. A diferencia de otros protocolos de comunicación digital como UART, SPI o I2C, este tipo de comunicación no requiere un intercambio de datos en formato binario con sincronización de bits o bytes, sino que se basa en la variación continua de voltaje a través de una línea de señal que es interpretada directamente por el convertidor analógico-digital (ADC) del microcontrolador.

Este método se conoce comúnmente como protocolo **Trigger/Echo** o **Time-of-Flight (ToF)** por ultrasonido, y consiste en intercambiar información a través de señales digitales simples de pulso, donde la duración de los mismos transmite la información.

La elección de este protocolo analógico responde a la necesidad de simplificar la implementación y garantizar una conexión confiable entre el sensor y el microcontrolador. El Arduino UNO, al contar con un ADC interno de 10 bits, puede convertir el valor de voltaje que entrega el sensor en una señal digital interpretable dentro del rango de 0 a 1023. De esta manera, no se requiere mayor complejidad en el manejo de buses de datos, relojes de sincronización o direcciones de dispositivos, como ocurre en protocolos más avanzados. Este enfoque resulta adecuado cuando se trabaja con sensores cuyo principio de funcionamiento es emitir una señal proporcional a la magnitud medida, como voltaje en función de temperatura, luz o presión.

Comparado con protocolos de comunicación digital como SPI o I2C, la transmisión analógica presenta ventajas y desventajas que deben ser tomadas en cuenta según la aplicación. Entre las ventajas principales se encuentra la simplicidad, ya que basta con una única línea de señal y una conexión a tierra compartida para transmitir la información. Esto reduce la cantidad de pines utilizados y facilita la integración del sistema. Además, al no depender de la configuración de registros de comunicación, la implementación resulta más rápida y accesible para proyectos o prototipos iniciales. Sin embargo, una de las limitaciones más evidentes de este protocolo analógico es la susceptibilidad al ruido eléctrico y a las pérdidas de señal, especialmente cuando se requiere transmitir a través de distancias relativamente largas. Esto puede generar errores en la medición, ya que pequeñas variaciones en el voltaje pueden ser interpretadas como cambios en la magnitud física que se está midiendo.

Si bien protocolos como I2C y SPI permiten la transmisión de datos digitales de manera más robusta y con mayores velocidades, su implementación requiere una configuración más detallada y un mayor número de líneas de comunicación. Por ejemplo, SPI ofrece una alta velocidad de transferencia y comunicación full-duplex, pero a costa de utilizar más pines y dificultar la escalabilidad cuando se incorporan múltiples dispositivos. I2C, por su parte, resuelve el problema de la escalabilidad mediante el uso de direcciones en un bus compartido, aunque con una menor velocidad en comparación con SPI. UART, en cambio, es muy eficiente para comunicación punto a punto y resulta ampliamente utilizado en módulos de comunicación como Bluetooth o GPS, pero no es tan adecuado para sensores que entregan señales analógicas directamente.

La decisión de utilizar un protocolo analógico en este caso se justifica porque el objetivo principal es mostrar de manera sencilla y práctica cómo un sistema de comunicación entre un microcontrolador y un sensor puede ser implementado para resolver necesidades de un cliente. Este enfoque permite exponer con claridad la interacción entre hardware y software sin la necesidad de recurrir a configuraciones avanzadas de buses de comunicación digital. Asimismo, en un entorno de prototipo inicial,

el empleo de un canal analógico permite concentrarse en la adquisición de datos y en la posterior aplicación de los mismos.

En conclusión, el protocolo de comunicación utilizado en este proyecto, basado en señales analógicas, constituye una solución adecuada cuando se requiere simplicidad, bajo costo de implementación y facilidad de integración con un microcontrolador como el Arduino UNO. Aunque presenta limitaciones frente a protocolos digitales más sofisticados, su uso sigue siendo altamente válido en sistemas donde la prioridad es la adquisición directa de variables físicas, la rapidez en la implementación y la claridad en la demostración de conceptos de comunicación embebida.

8.1. Resultados

Se realizó el montaje experimental de acuerdo con el diseño descrito en la Figura 3 y utilizando los componentes listados en la Tabla 1, durante la sesión de laboratorio efectuada el día miércoles 27 de agosto de 2025 en el aula Multiversa de la Escuela de Ingeniería Eléctrica de la Universidad de Costa Rica.

En la Figura 5 se muestra el montaje físico del sistema, en el cual se aprecia la conexión del microcontrolador *Arduino UNO*, el sensor ultrasónico HC-SR04 y los cuatro diodos emisores de luz (LEDs) que funcionan como indicadores de rango. La comunicación entre el microcontrolador y el sensor se verificó a través del *Serial Monitor* de Arduino IDE, donde se registraron las mediciones en tiempo real, y mediante la respuesta visual de los LEDs, los cuales se activaban según el rango de distancia definido en el código.

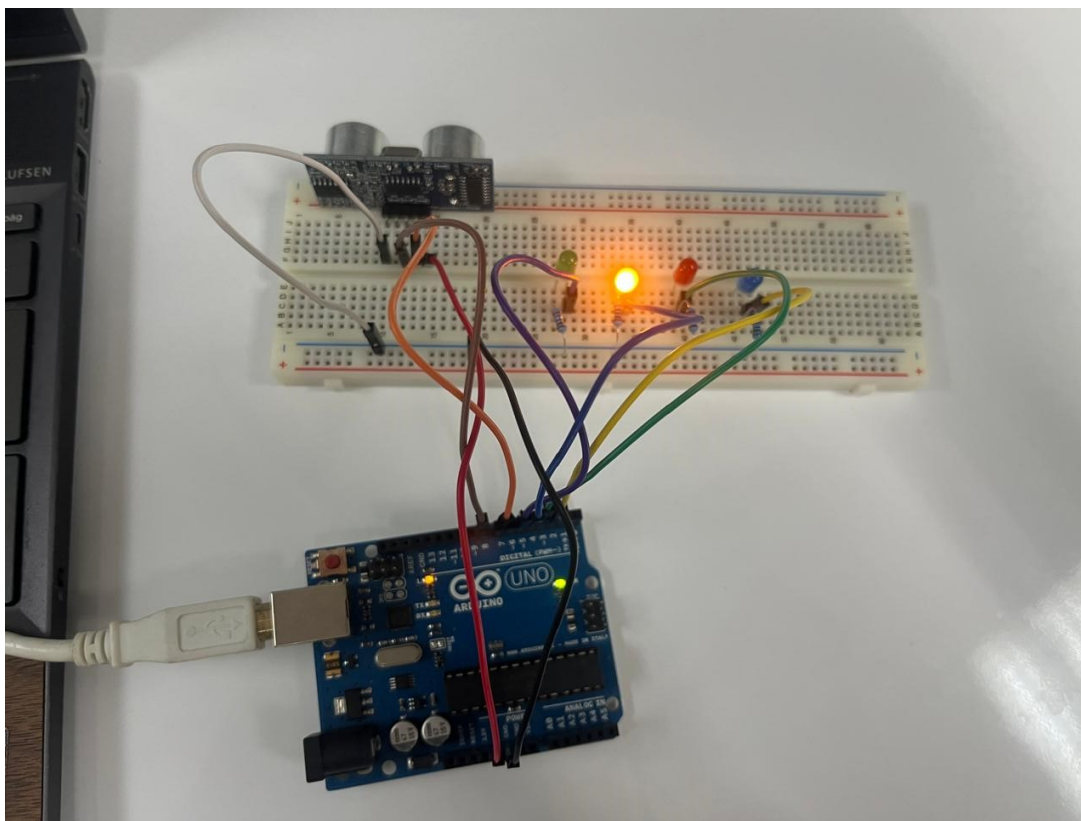


Figura 5: Montaje físico del sistema experimental.

Elaboración propia

En la Figura 6 se observa la activación del LED verde cuando el sensor registra un objeto a una distancia aproximada de 165 cm. Este comportamiento coincide con el umbral de rango definido en el programa para distancias entre 150 y 310 cm. El *Serial Monitor* confirma lecturas estables en dicho intervalo, validando la correcta transmisión de datos entre los circuitos integrados.

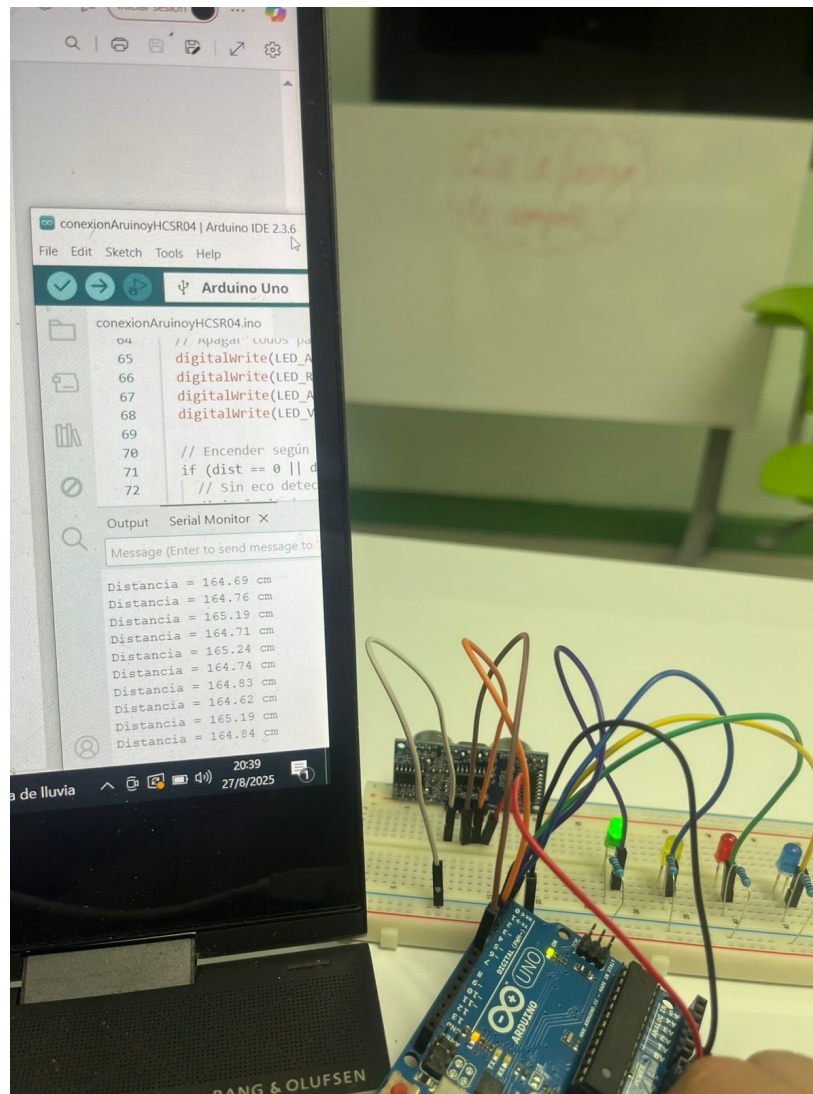


Figura 6: Activación del LED verde para distancias aproximadas de 160 cm.

Elaboración propia

En la Figura 7 se muestra la activación del LED amarillo al detectar un objeto a una distancia cercana a 110 cm. Este resultado confirma que el sistema responde adecuadamente en el rango medio definido (30–150 cm). Las mediciones reportadas en el *Serial Monitor* presentan una variación mínima ($\pm 0,5$ cm), lo que demuestra un comportamiento estable y confiable en este intervalo.

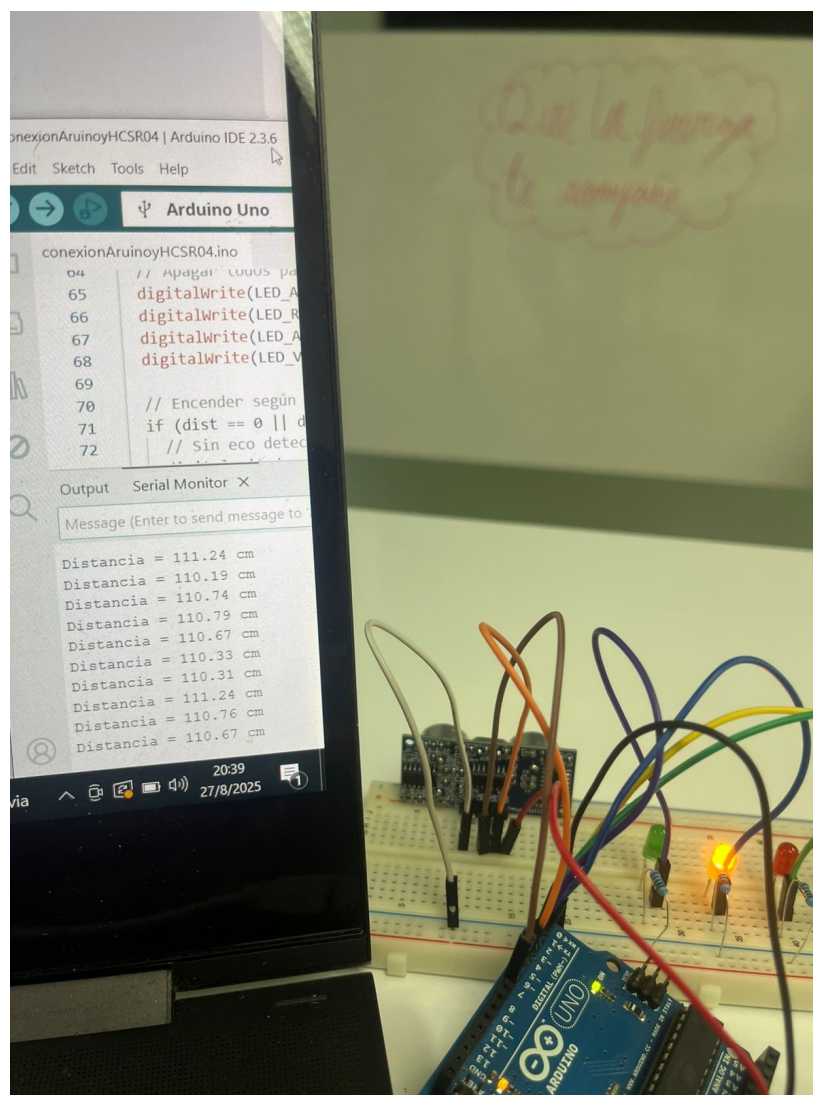


Figura 7: Activación del LED amarillo para distancias cercanas a 110 cm.

Elaboración propia

Por su parte, en la Figura 8 se evidencia la activación del LED rojo al colocar un objeto a una distancia de aproximadamente 6,5 cm. Las lecturas obtenidas en el *Serial Monitor* se mantienen consistentes, lo cual valida el funcionamiento del sistema en el rango crítico de proximidad (distancias menores a 30 cm).

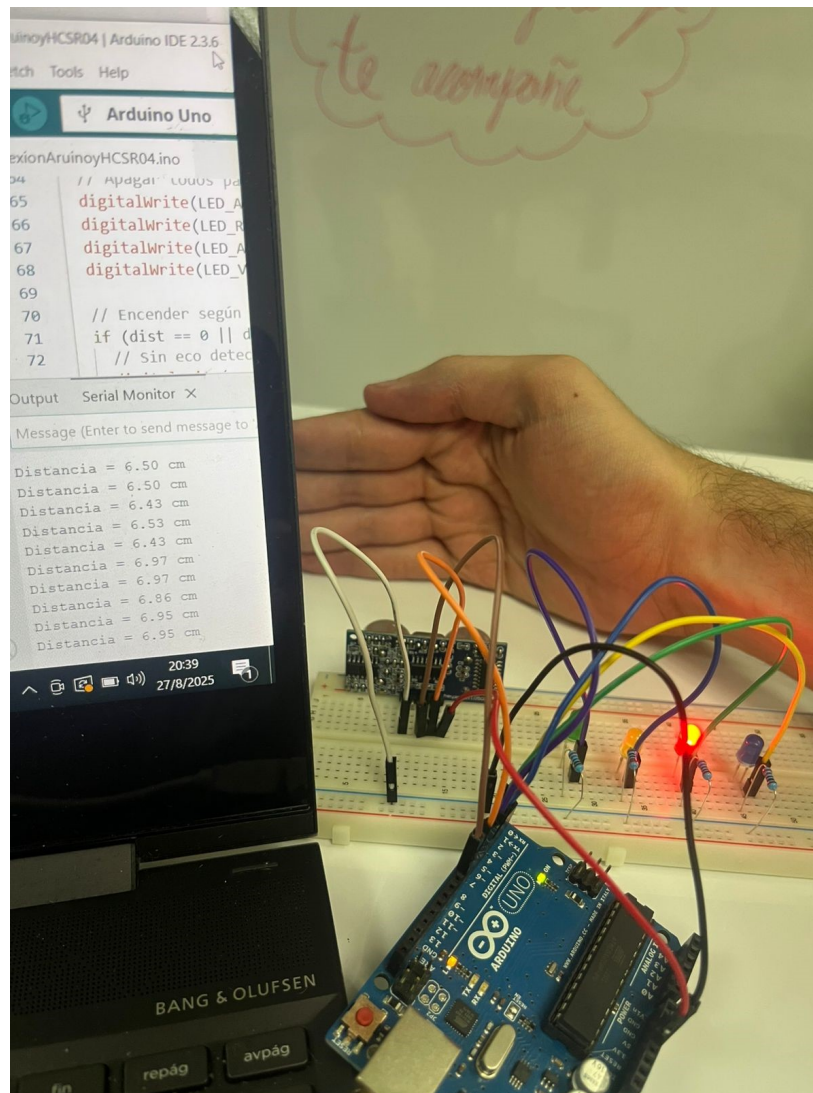


Figura 8: Activación del LED rojo para distancias de aproximadamente 6,5 cm.

Elaboración propia

Finalmente, cuando el sensor se orienta hacia espacios abiertos sin obstáculos cercanos, se enciende el LED azul, como se observa en la Figura 9. En este caso, el *Serial Monitor* registra valores superiores a los 2300 cm, que exceden el rango de operación definido en el código. Este resultado confirma que el sistema distingue correctamente los casos de “fuera de rango” o ausencia de eco detectable.

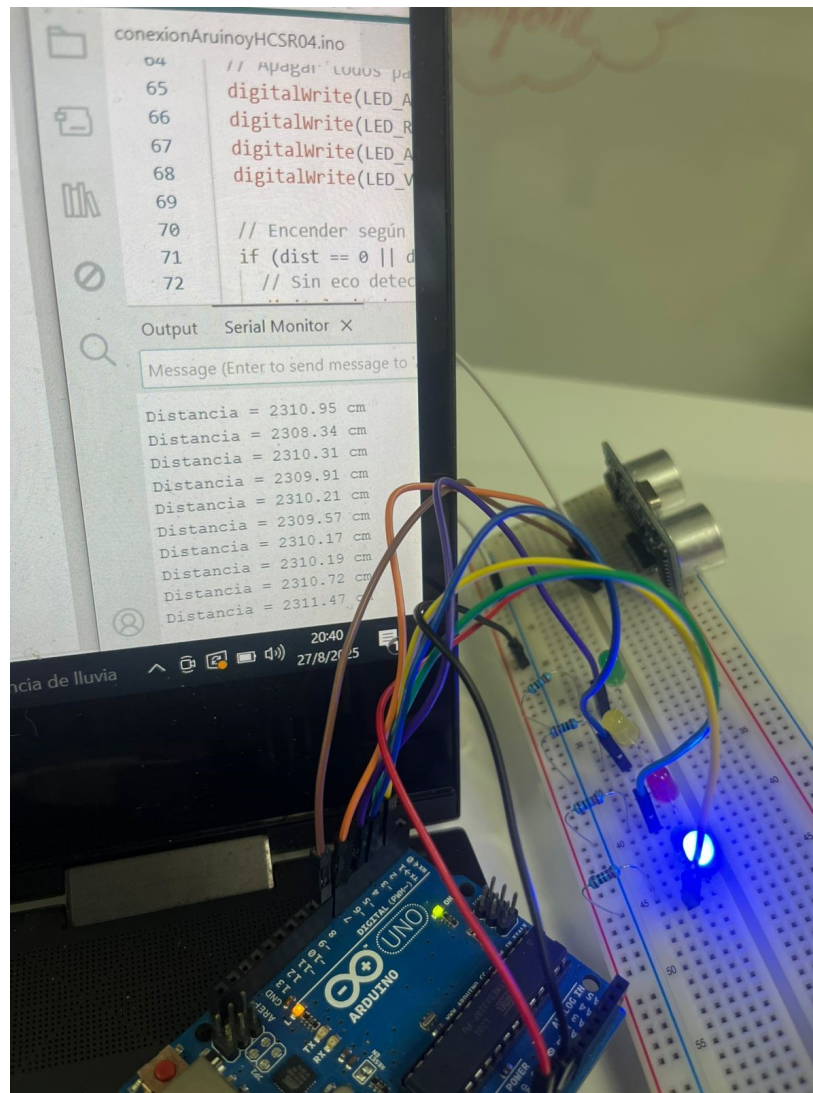


Figura 9: Activación del LED azul cuando el sensor registra distancias fuera del rango definido.

Elaboración propia

En síntesis, todas las pruebas realizadas fueron exitosas. Se validó que la comunicación entre el microcontrolador y el sensor ultrasónico permite medir distancias de manera confiable, y que la representación de los rangos mediante LEDs constituye un mecanismo efectivo de visualización. Los resultados confirman la correcta implementación de la comunicación digital entre los circuitos integrados.

9. Presupuesto

9.1. Desglose de horas trabajadas

Tabla 2: Horas de trabajo

Actividad	Horas estimadas
Investigación y planificación	8 h
Diseño del sistema (hardware/software)	10 h
Desarrollo de firmware / programación	16 h
Pruebas y validaciones	10 h
Documentación y presentación final	6 h
Total	50 h

9.2. Tarifa de referencia

La tarifa establecida es de ₡4,000 por hora.

9.3. Estimación de costo de mano de obra

$$50 \text{ horas} \times \text{₡}4,000/\text{hora} = \text{₡}200,000$$

9.4. Materiales y recursos

Tabla 3: Materiales

Componente	Cantidad	Proveedor	Costo (₡)
Arduino UNO	1	Industrial Automaton	
Protoboard	1	Industrial Automaton	
Sensor ultrasónico HC-SR04	1	Industrial Automaton	—
LED color azul	1	Industrial Automaton	—
LED color rojo	1	Industrial Automaton	—
LED color amarillo	1	Industrial Automaton	—
LED color verde	1	Industrial Automaton	—
Resistencias 220Ω	4	Industrial Automaton	—
Cables de conexión	1 set	Industrial Automaton	—

9.5. Costo total del proyecto

Por lo tanto el costo total del proyecto es de : **₡200,000**

10. Cronograma

Tabla 4: Cronograma

Fecha	Actividad
Miércoles (Día 1)	Revisión de los Términos de Referencia (TdR) y definición del alcance.
Jueves (Día 2)	Planteamiento de objetivos y metodología de trabajo.
Viernes (Día 3)	Diseño preliminar del hardware (esquemático y descripción).
Sábado (Día 4)	Selección de componentes y verificación de compatibilidad.
Domingo (Día 5)	Documentación de diseño de hardware.
Lunes (Día 6)	Diseño preliminar del software (flujo lógico y pseudocódigo).
Martes (Día 7)	Desarrollo del código base en Arduino.
Miércoles (Día 8)	Revisión y depuración del código inicial.
Jueves (Día 9)	Integración del hardware y software en el documento del anteproyecto.
Viernes (Día 10)	Redacción del apartado de desempeño esperado y presupuesto estimado.
Sábado (Día 11)	Revisión y ajustes al cronograma dentro del documento.
Domingo (Día 12)	Redacción de conclusiones parciales y revisión general.
Lunes (Día 13)	Corrección de estilo y preparación del documento final.
Martes (Día 14)	Entrega final del proyecto.

11. Conclusiones

El desarrollo del presente estudio permitió cumplir con el objetivo general de diseñar e implementar un sistema de medición de distancia basado en un sensor ultrasónico HC-SR04 y un microcontrolador Arduino UNO, verificando tanto la transmisión como el procesamiento de datos en tiempo real. La integración del hardware con el software desarrollado en la plataforma Arduino IDE demostró ser efectiva y consistente durante las pruebas realizadas en laboratorio.

En relación con los objetivos específicos, se concluye lo siguiente:

- **Configuración del sensor y microcontrolador:** La conexión física entre el sensor ultrasónico HC-SR04 y el Arduino UNO se ejecutó exitosamente, permitiendo la correcta adquisición de datos de distancia. La codificación digital mediante la duración del pulso en el pin `ECHO` se interpretó de manera confiable por el microcontrolador.
- **Implementación de la lógica de adquisición y procesamiento:** El programa desarrollado permitió generar pulsos de activación, capturar los ecos y convertirlos en mediciones de distancia. Las pruebas confirmaron la estabilidad del algoritmo y la correcta traducción del tiempo de vuelo de la señal acústica en valores espaciales.
- **Visualización de resultados:** La información obtenida fue mostrada tanto en el *Serial Monitor* como en el sistema de indicadores luminosos. Los LEDs respondieron de manera adecuada a los rangos de distancia definidos: rojo para proximidad, amarillo para rango medio, verde para rango lejano y azul para fuera de rango. Esto facilitó la interpretación visual inmediata del desempeño del sistema.
- **Análisis de desempeño:** Las mediciones presentaron un margen de error reducido (aproximadamente ± 1 cm en condiciones normales), lo que demuestra la precisión y estabilidad del sistema. Asimismo, la transición entre los distintos rangos se mostró fluida y coherente con la programación establecida.

En conclusión, el prototipo implementado valida de forma satisfactoria la comunicación entre circuitos integrados mediante señales digitales de tiempo, y constituye una base sólida para futuras extensiones, tales como la incorporación de interfaces gráficas, sistemas de almacenamiento de datos o integración en plataformas de robótica móvil.

12. Recomendaciones

A partir de la experiencia obtenida en el desarrollo e implementación del sistema de medición de distancia, se plantean las siguientes recomendaciones para futuros trabajos y mejoras del prototipo:

- **Ampliación de la interfaz de salida:** Incorporar una pantalla LCD o una interfaz gráfica en computadora para visualizar de forma más detallada las mediciones, permitiendo registrar valores numéricos con mayor claridad y en tiempo real.
- **Optimización del procesamiento de datos:** Implementar técnicas de filtrado digital (por ejemplo, promediado móvil o filtros de mediana) para reducir el efecto de posibles ruidos en la señal y mejorar la estabilidad de las lecturas en entornos no controlados.
- **Validación en condiciones diversas:** Realizar pruebas adicionales en ambientes con diferentes niveles de ruido acústico, variaciones de temperatura y superficies reflectivas para evaluar el desempeño del sensor y cuantificar el margen de error en escenarios reales.
- **Gestión energética:** Analizar opciones de alimentación alternativas, como baterías recargables o fuentes de bajo consumo, con el fin de hacer el prototipo independiente de la conexión USB y mejorar su portabilidad.

En síntesis, estas recomendaciones buscan potenciar el prototipo desarrollado, aportando nuevas funcionalidades y robustez al sistema, de manera que pueda evolucionar hacia aplicaciones más complejas en el ámbito de la robótica y los sistemas embebidos.

Referencias

1. Tinkercad, “Sensor ultrasónico HC-SR04 en simulador,” [En línea]. Disponible en: <https://www.tinkercad.com/things/1E3hXimmjCt-sensor-ultrasonico-hc-sr04>. [Accedido: 27-ago-2025].
2. Arduino, “Uso del sensor ultrasónico HC-SR04,” YouTube, [En línea]. Disponible en: <https://www.youtube.com/watch?v=SGHFbNRu1BA&t=384s>. [Accedido: 27-ago-2025].
3. Isaac100, “Getting started with the HC-SR04 ultrasonic sensor,” Arduino Project Hub, [En línea]. Disponible en: <https://projecthub.arduino.cc/Isaac100/getting-started-with-the-hc-sr04-ultrasonic-sensor-7cabe1>. [Accedido: 27-ago-2025].
4. Handson Technology, *HC-SR04 Ultrasonic Sensor Module User Guide*, Datasheet, 2019.
5. Arduino S.r.l., *Arduino UNO R3*, Datasheet, 2024.