# 1️⃣ Definición de pines y variables

const int trigPin = 9; const int echoPin = 10;

* Aquí decimos que:
  + El pin **TRIG** del sensor (el que envía la señal) está conectado al pin digital 9 de Arduino.
  + El pin **ECHO** (el que recibe la señal) está conectado al pin digital 10.
* const int significa que son números constantes que no van a cambiar.

long duracion; // Tiempo de ida y vuelta del sonido en microsegundos float distancia; // Distancia calculada en centímetros

* duracion guarda el tiempo total que tarda el sonido en **ir y volver** hasta el objeto.
* distancia guarda el resultado de la fórmula d=t/59d = t/59.



# 2️⃣ Configuración inicial (setup)

void setup() {

Serial.begin(9600); // Velocidad de comunicación con la PC pinMode(trigPin, OUTPUT); // TRIG es salida (manda pulso) pinMode(echoPin, INPUT); // ECHO es entrada (recibe pulso)

}

* Serial.begin(9600) permite enviar datos a la PC para verlos en el **Monitor Serie**.
* pinMode(...) indica si un pin es **entrada** (INPUT) o **salida** (OUTPUT).



# 3️⃣ Bucle principal (loop)

Este código se repite sin parar. a) Preparar el pin TRIG

digitalWrite(trigPin, LOW); delayMicroseconds(2);

* Aseguramos que el TRIG esté **apagado** antes de empezar.
* Esperamos 2 microsegundos para que el sensor esté listo.

## b) Enviar pulso ultrasónico

digitalWrite(trigPin, HIGH); delayMicroseconds(10); digitalWrite(trigPin, LOW);

* Encendemos el TRIG durante **10 μs** (esto hace que el sensor emita un “clic” ultrasónico).
* Luego lo apagamos.

## c) Medir tiempo de ida y vuelta

duracion = pulseIn(echoPin, HIGH);

* pulseIn mide cuánto tiempo el pin **ECHO** está en estado **HIGH**.
* Ese tiempo corresponde al viaje de ida y vuelta del sonido.

## d) Calcular la distancia

distancia = duracion / 59.0;

* Usamos la fórmula d=t/59d = t/59 que vimos antes.
* El resultado es en **centímetros**.

## e) Mostrar en la pantalla de la PC

Serial.print("Distancia: ");

Serial.print(distancia); Serial.println(" cm");

* Serial.print envía texto o números al **Monitor Serie** de Arduino.
* Esto nos deja ver la distancia en tiempo real.

## f) Pequeña pausa

delay(500);

• Espera 0.5 segundos antes de la próxima medición (evita que se llene la pantalla muy rápido).



**Resumen de lo que hace el Arduino en cada ciclo:**

1. Apaga TRIG para reiniciar el sensor.
2. Enciende TRIG 10 μs → el sensor envía un pulso de sonido.
3. Mide cuánto tarda en regresar ese sonido.
4. Aplica la fórmula para convertir tiempo en centímetros.
5. Muestra el resultado en la PC.
6. Espera medio segundo y repite.