

PROYECTO POSTURA

MATERIALES:

- ARDUINO UNO
- SENSOR ULTRASONIDO
- LED
- BUZZER
- JUMPER
- PROTOBOARD

CONEXIONES

ULTRASONIDO

VCC A		NEGATIVO PROTOBOARD
TRIGER A		4 DIGITAL PWM ARDUINO
ECHO A		5 DIGITAL PWM ARDUINO
GND A		POSITIVO PROTOBOARD

LED

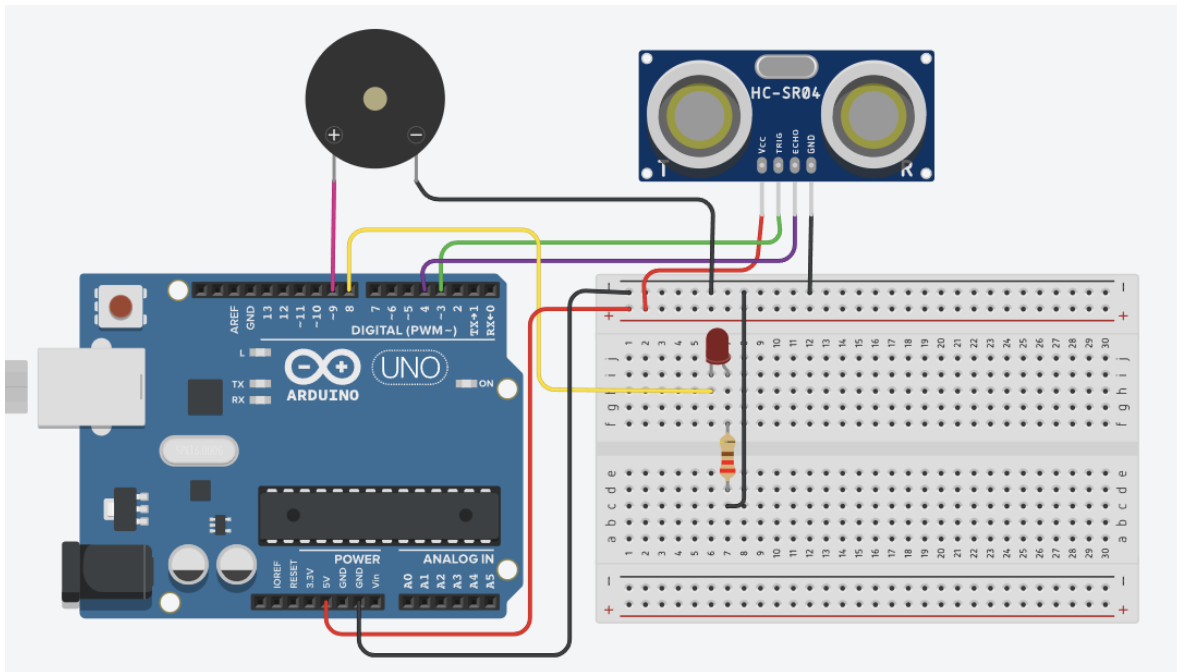
S A		8 DIGITAL PWM
V	A	NEGATIVO PROTOBOARD
G	A	POSITIVO PROTOBOARD

BUZZER

S	A	9 DIGITAL PWM ARDUINO
V	A	NEGATIVO PROTOBOARD
G	A	POSITIVO PROTOBOARD

PROTOBOARD

NEGATIVO	A	5V ARDUINO POWER
POSITIVO	A	GND ARDUINO POWER



CODIGO.

```
// Sensor HC-SR04 + LED + Buzzer
```

```
// Pines sensor
```

```
const int trigPin = 3;
```

```
const int echoPin = 4;
```

```
// LED y buzzer
```

```
const int led = 8;
```

```
const int buzzer = 9;
```

```
long duracion;
```

```
float distancia;
```

```
void setup() {
```

```
    Serial.begin(9600);
```

```
    pinMode(trigPin, OUTPUT);
```

```
    pinMode(echoPin, INPUT);
```

```
    pinMode(led, OUTPUT);
```

```
    pinMode(buzzer, OUTPUT);
```

```
}
```

```
void loop() {
```

```

    distancia = medirDistancia();

    Serial.print("Distancia: ");
    Serial.println(distancia);

    if (distancia < 10) {
        // ✓ POSTURA CORRECTA
        digitalWrite(led, LOW);
        noTone(buzzer);
    }
    else {
        // ✗ POSTURA INCORRECTA
        digitalWrite(led, HIGH);
        tone(buzzer, 1000); // sonido continuo
    }

    delay(300);
}

// Función medir distancia
float medirDistancia() {
    digitalWrite(trigPin, LOW);
    delayMicroseconds(5);

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

    duracion = pulseIn(echoPin, HIGH, 30000);

    if (duracion == 0) return 999;

    return duracion * 0.034 / 2;
}

```

