

# Prática 02:

## Programando ESP32 com interfaces

---

Disciplina: **Introdução à Internet das Coisas - IMD0902**

Prof. Heitor Florencio

Prof. Leonardo Augusto

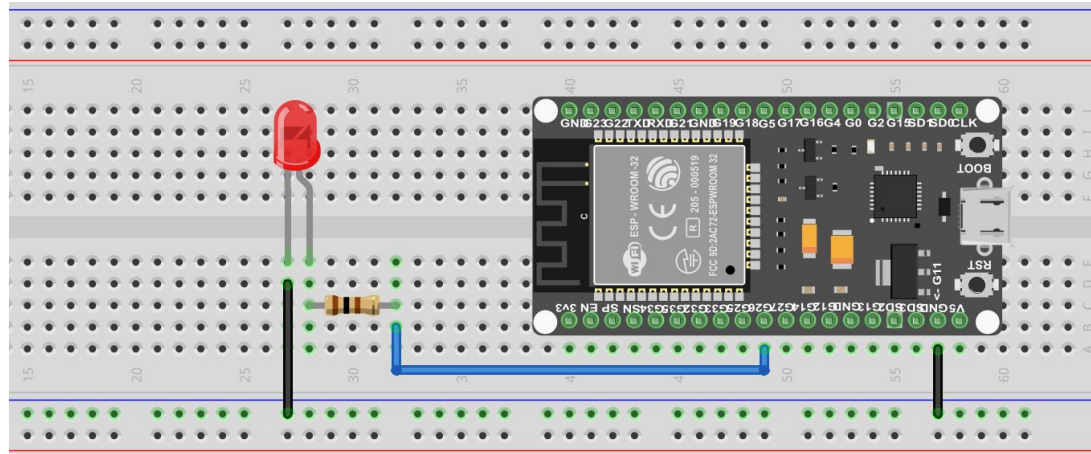
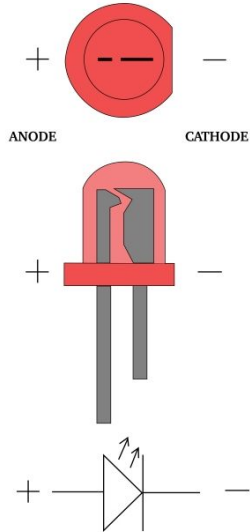
Aula:  
**Prática 02: Programando  
ESP32 com interfaces**

## Tópicos

- Experimento 01: Acionar um LED.
- Experimento 02: Detectar proximidade de objetos.
- Experimento 03: Controlar ativação do LED ou buzzer a partir da distância de objetos.

# Experimento 01: Acionar LED

- Objetivo: Acionar um LED a partir do pino GPIO26 do ESP32.
- Requisitos funcionais:
  - O microcontrolador ESP32 deve manter o LED ativado por 100 ms e, em seguida, desativado por 100 ms. Esse ciclo de ativação e desativação deve ser repetido.



# Experimento 01: Acionar LED

- Código:



```
sketch_pratica01_aciona-led | Arduino 1.8.13
Arquivo  Editar  Sketch  Ferramentas  Ajuda

sketch_pratica01_aciona-led

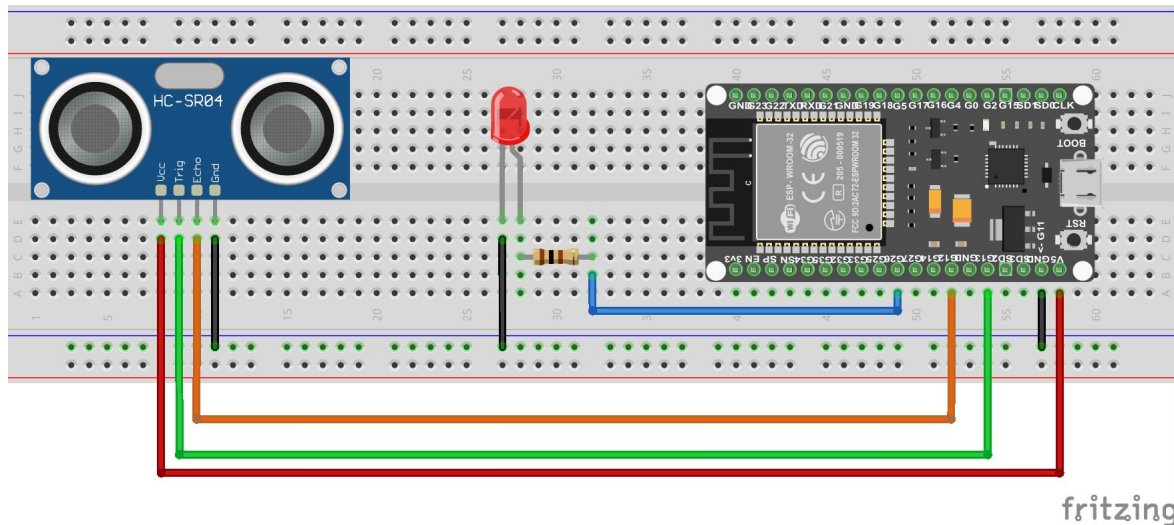
int led = 26;

void setup() {
  pinMode(led, OUTPUT);
}

void loop() {
  digitalWrite(led, HIGH);
  delay(100);
  digitalWrite(led, LOW);
  delay(100);
}
```

# Experimento 02: Detectar proximidade de objetos

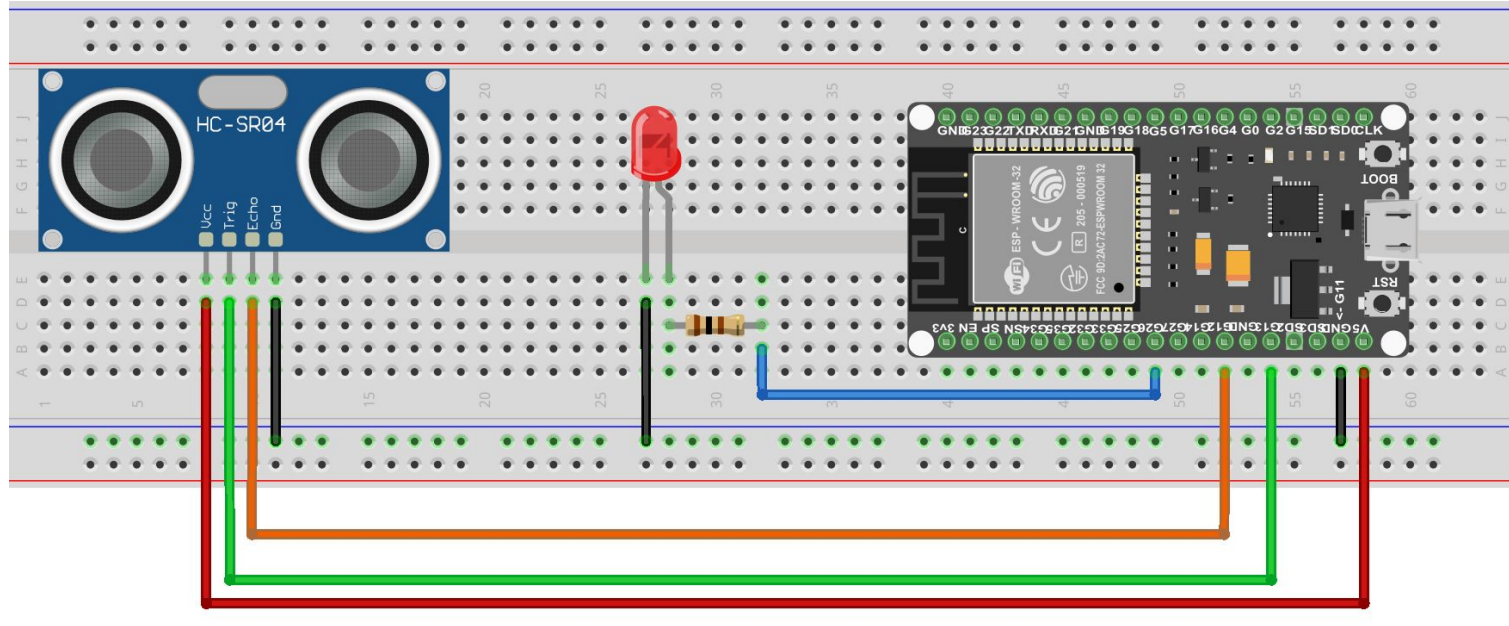
- Objetivo: Medir distância de objetos a partir do sensor HC-SR04.
- Requisitos funcionais:
  - O microcontrolador ESP32 deve ler o valor do sensor ultrassônico de proximidade HC-SR04 a cada 1 segundo.
- Diagrama:



# Experimento 02: Detectar proximidade de objetos

→ Pino GPIO13 conectado ao pino Trigger do sensor.

→ Pino GPIO12 conectado ao pino Echo do sensor.

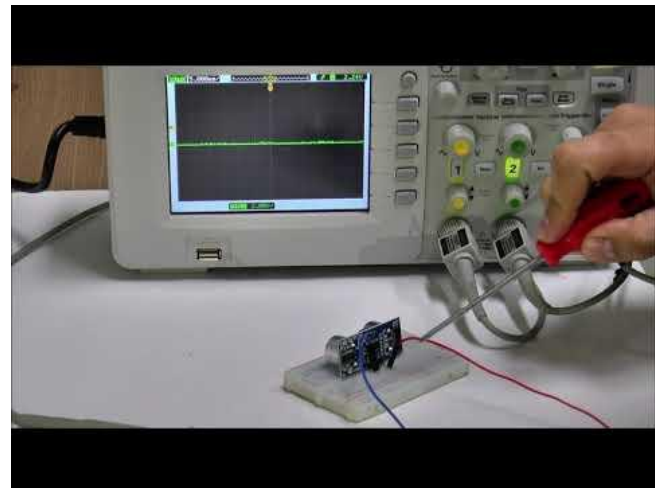
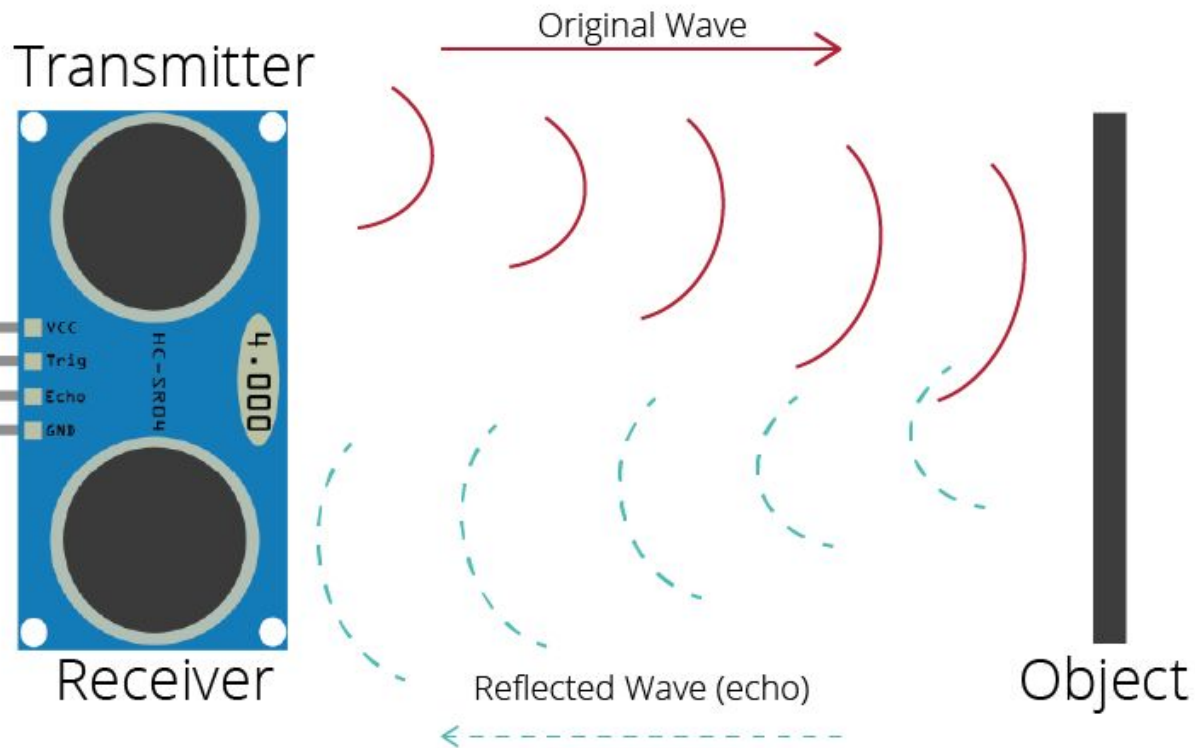


# Sensor ultrassônico HC-SR04



Power Supply	5V DC
Working Current	15 mA
Working Frequency	40 kHz
Maximum Range	4 meters
Minimum Range	2 cm
Measuring Angle	15°
Resolution	0.3 cm
Trigger Input Signal	10uS TTL pulse
Echo Output Signal	TTL pulse proportional to the distance range
Dimensions	45mm x 20mm x 15mm

# Sensor ultrassônico HC-SR04





# Experimento 02: Detectar proximidade de objetos

- Código:

```
sketch_pratica02_sensor-distancia $  
  
int trigPin = 13;  
int echoPin = 12;  
  
//define sound speed in cm/uS  
#define SOUND_SPEED 0.034  
  
long duracaoPulso;  
float distanciaCm;  
  
void setup() {  
    Serial.begin(115200);  
    pinMode(trigPin, OUTPUT);  
    pinMode(echoPin, INPUT);  
}
```

# Experimento 02: Detectar proximidade de objetos

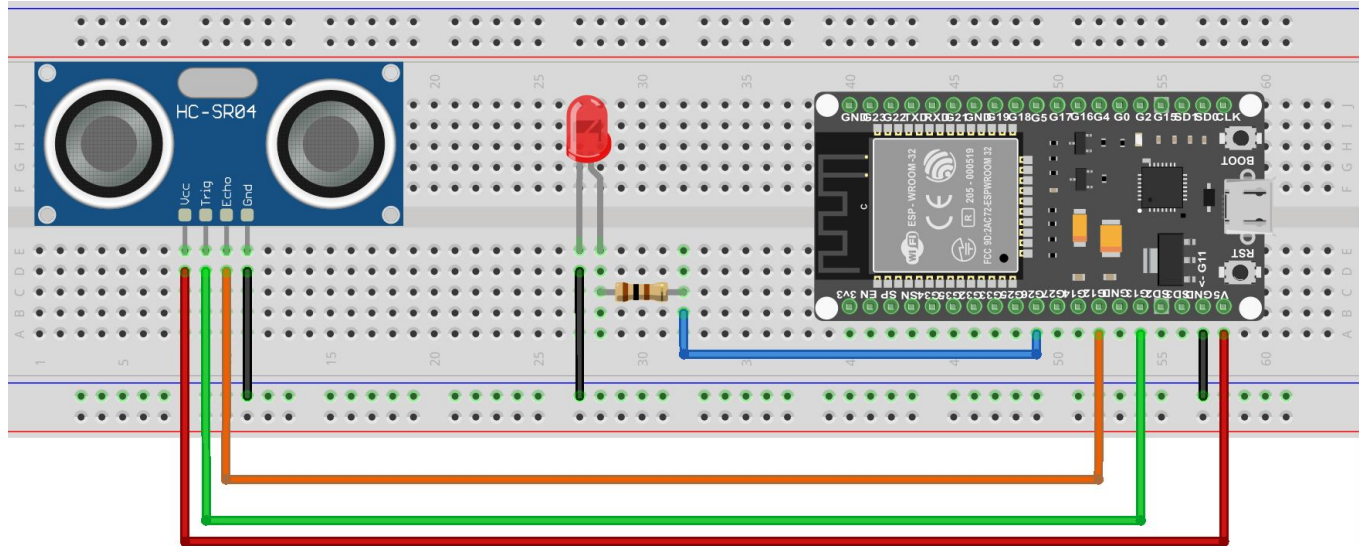
- Código:

```
sketch_pratica02_sensor-distancia$  
  
int trigPin = 13;  
int echoPin = 12;  
  
//define sound speed in cm/uS  
#define SOUND_SPEED 0.034  
  
long duracaoPulso;  
float distanciaCm;  
  
void setup() {  
    Serial.begin(115200);  
    pinMode(trigPin, OUTPUT);  
    pinMode(echoPin, INPUT);  
}
```

```
void loop() {  
    // Envio do sinal de trig  
    digitalWrite(trigPin, LOW);  
    delayMicroseconds(2);  
    // Seta o pino Trig alto por 10 ms  
    digitalWrite(trigPin, HIGH);  
    delayMicroseconds(10);  
    digitalWrite(trigPin, LOW);  
  
    // Calcular o tempo que o pino fica alto em ms  
    duracaoPulso = pulseIn(echoPin, HIGH);  
  
    // Calcula a distância  
    distanciaCm = duracaoPulso * SOUND_SPEED / 2;
```

# Experimento 03: Controlar ativação do LED ou buzzer a partir da distância de objetos

- Objetivo: Controlar ativação do LED ou buzzer a partir da distância de objetos.
- Requisitos funcionais:
  - O microcontrolador ESP32 deve ativar/desativar o LED ou o buzzer em uma frequência maior quando o objeto estiver próximo.
- Diagrama:



## Experimento 03: Controlar ativação do LED ou buzzer a partir da distância de objetos

- Código:

**Desenvolva  
seu código!**

# Dúvidas?

**Prof Heitor Florencio**  
**IMD/UFRN**  
**Sala 103 - nPITI/IMD**  
**heitorm@imd.ufrn.br**

---