

# **SENAI Joinville Norte I**

*Leonardo da Silva da Cruz*

*Nicolas Vinicius Jenichen*

*Luis Fernando dos Santos*

## **TRABALHO** **"SITUAÇÃO APRENDIZAGEM"**

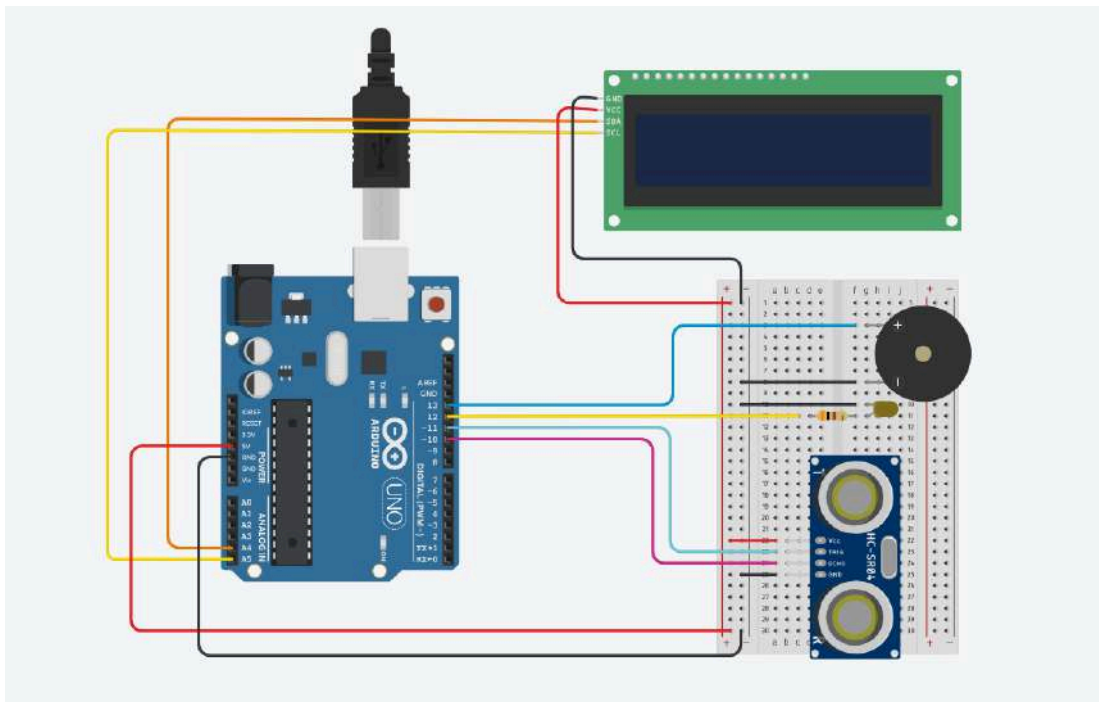
*JOINVILLE/SC*

*2024*

# PROJETOS

## *Leonardo da Silva da Cruz:*

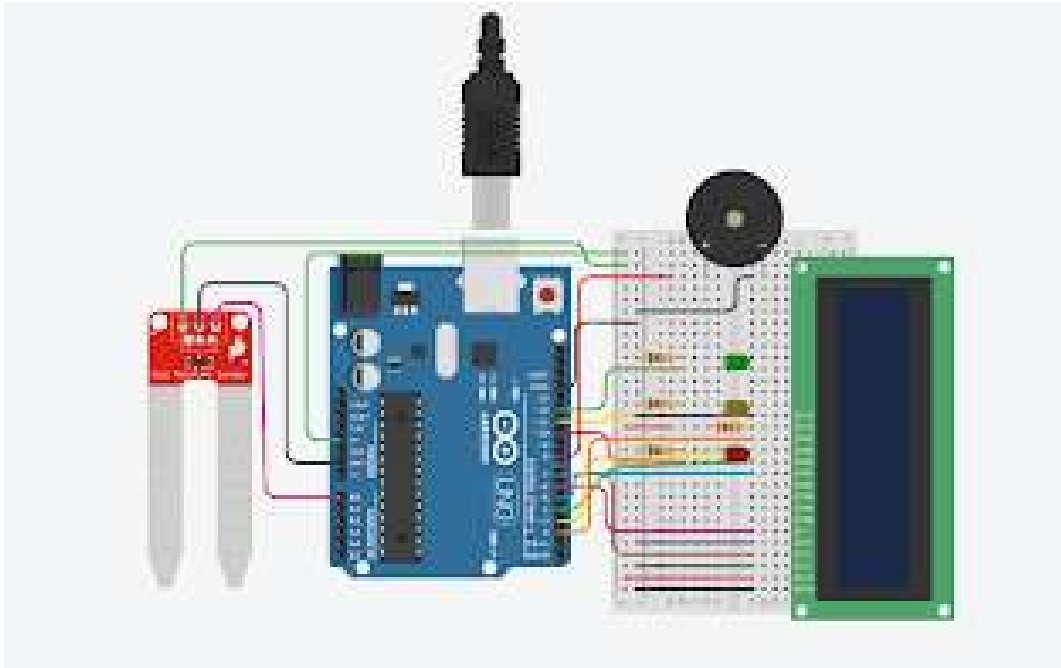
- **Nome:** S.P.A.D.(Sensor de proximidade com alerta de distância).
- **Componentes:** Arduino Uno R3, LCD 16x2 I2C, Placa de ensaio, Piezo Buzzer, Led, Resistor e Sensor de distância ultrassônico.
- **Código Fonte do projeto no anexo A;**
- **Imagem:**



- **Aplicabilidade:** Veículos, segurança, vigilância e indústria.
- **Viabilidade da implantação:** Ótimo, visto que é barato, simples e eficaz.
- **Custo:** R\$150,00.

## ***Nicolas Vinicius Jenichen***

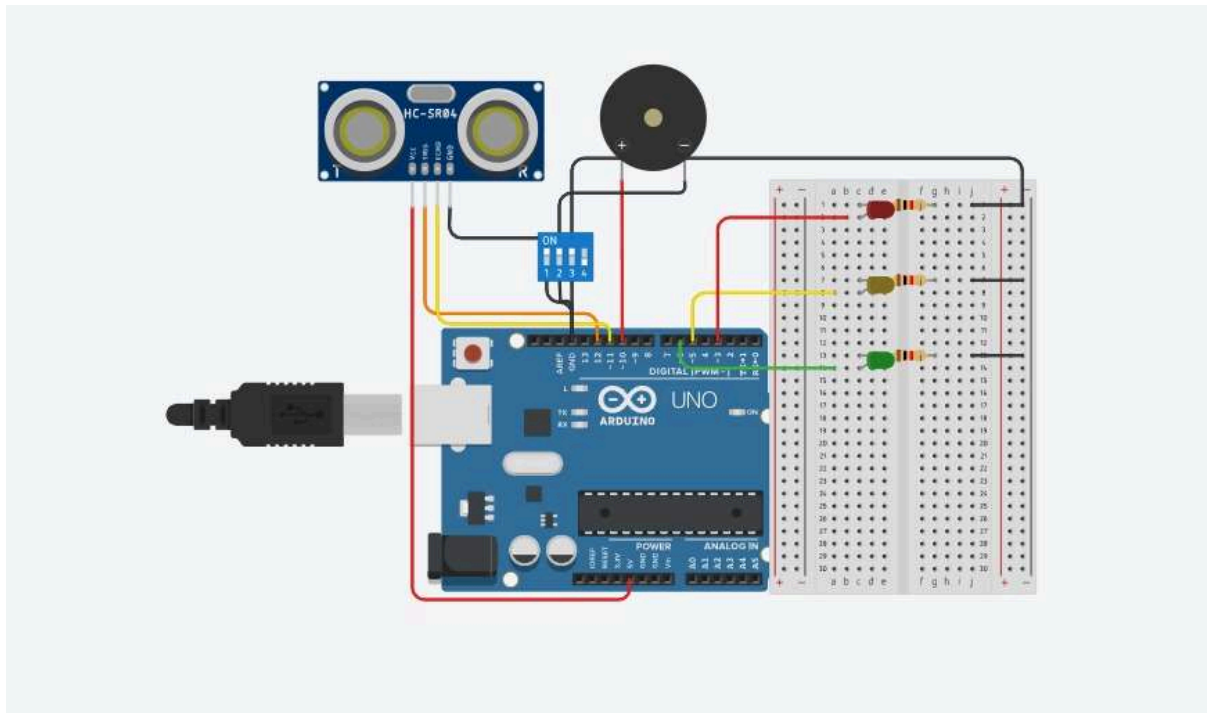
- **Nome:** Medidor de chuva para o RS
- **Componentes:** Arduino Uno R3, LCD 16x2, Placa de ensaio, Piezo Buzzer, Led, Resistor e Sensor de Umidade do Solo.
- **Código Fonte do projeto no anexo B;**
- **Imagem:**



- **Aplicabilidade:** Monitorar o crescimento da água.
- **Viabilidade da implantação:** Boa, visto que esse projeto também serve para medir a umidade do solo.
- **Custo:** R\$160,00.

## ***Luis Fernando dos Santos***

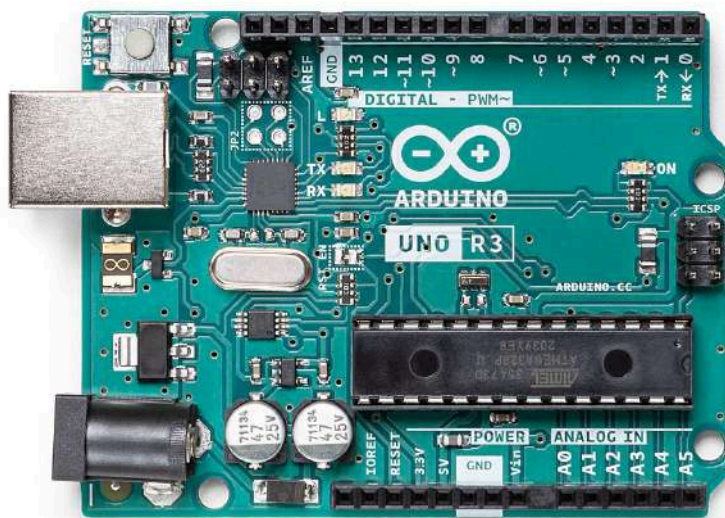
- **Nome:** Sensor de ré.
- **Componentes:** Arduino Uno R3, DIP Switch 4 Vias, Placa de ensaio, Piezo Buzzer, Led, Resistor e Sensor de Umidade do Solo.
- **Código fonte do projeto no Anexo C;**
- **Imagem:**



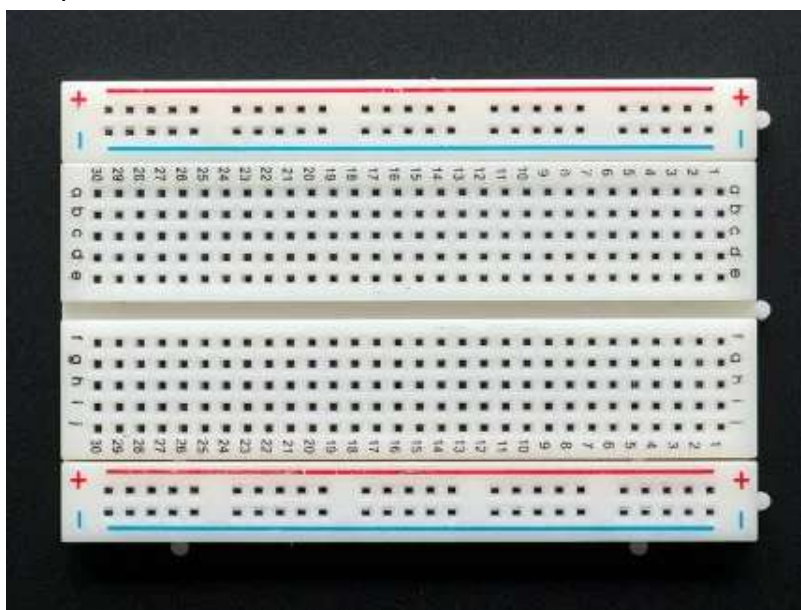
- **Aplicabilidade:** Veículos automotores.
- **Viabilidade da implantação:** Ótima, barata e funcional.
- **Custo:** R\$135,00.

# COMPONENTES UTILIZADOS NOS PROJETOS

- 1. Arduino Uno R3:** Uma placa de microcontrolador usada para criar projetos de IoT e eletrônicos. Ele fornece uma plataforma de desenvolvimento para criar dispositivos interativos controlados por computador.



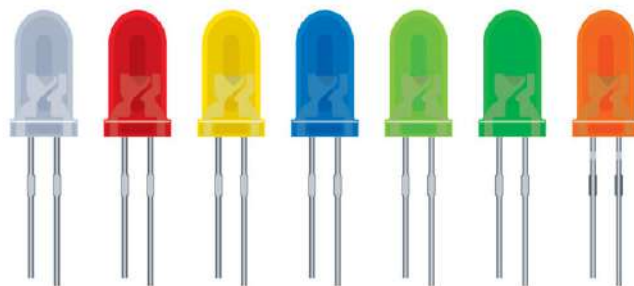
- 2. Placa de Ensaio:** É usada para prototipagem rápida de circuitos eletrônicos. Ela permite conectar componentes eletrônicos sem a necessidade de soldagem, facilitando o teste e a criação de circuitos temporários.



- 3. Sensor de distância ultrassônico:** É usado para medir a distância entre o sensor e um objeto usando ondas sonoras de alta frequência. Ele é comumente utilizado em projetos de IoT para detecção de obstáculos, monitoramento de níveis de líquidos e controle de distância em robótica, entre outros fins.



- 4. Led:** LED é um componente eletrônico usado para emitir luz quando uma corrente elétrica passa por ele.



**5. Resistor:** é um componente eletrônico usado para limitar a corrente elétrica em um circuito, controlar a voltagem, proteger outros componentes e dividir a tensão.



**6. Piezo Buzzer:** É um dispositivo utilizado para produzir som através da vibração de um material piezoelétrico quando uma corrente elétrica é aplicada a ele. Ele é comumente usado em projetos eletrônicos para fornecer feedback sonoro, como alarmes, notificações e indicadores de eventos.



**7. LCD 16x2 I2C:** O LCD 16x2 I2C é um tipo de display de cristal líquido com capacidade para exibir 16 caracteres em 2 linhas. A interface I2C permite uma conexão mais simples com microcontroladores, exigindo menos pinos de comunicação. Ele é comumente usado em projetos de IoT e eletrônicos para exibir informações, como textos, números e gráficos simples.





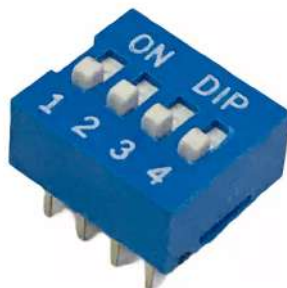
- 8. Sensor de umidade do solo:** O sensor de umidade do solo é usado para medir a quantidade de água presente no solo, fornecendo informações essenciais para o monitoramento e controle da irrigação em agricultura de precisão e em sistemas de jardinagem automatizados.



- 9. LCD 16x2:** O LCD 16x2 (Liquid Crystal Display) é um tipo de tela que pode exibir até 16 caracteres em duas linhas. Ele é comumente usado em projetos de eletrônica e IoT para exibir informações como texto, números, ou até mesmo gráficos simples.



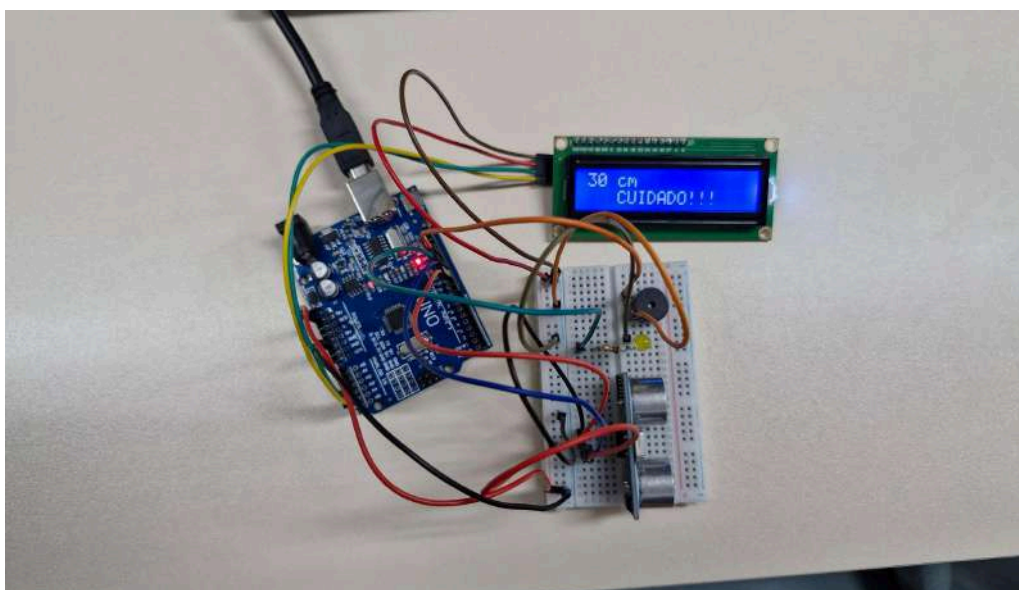
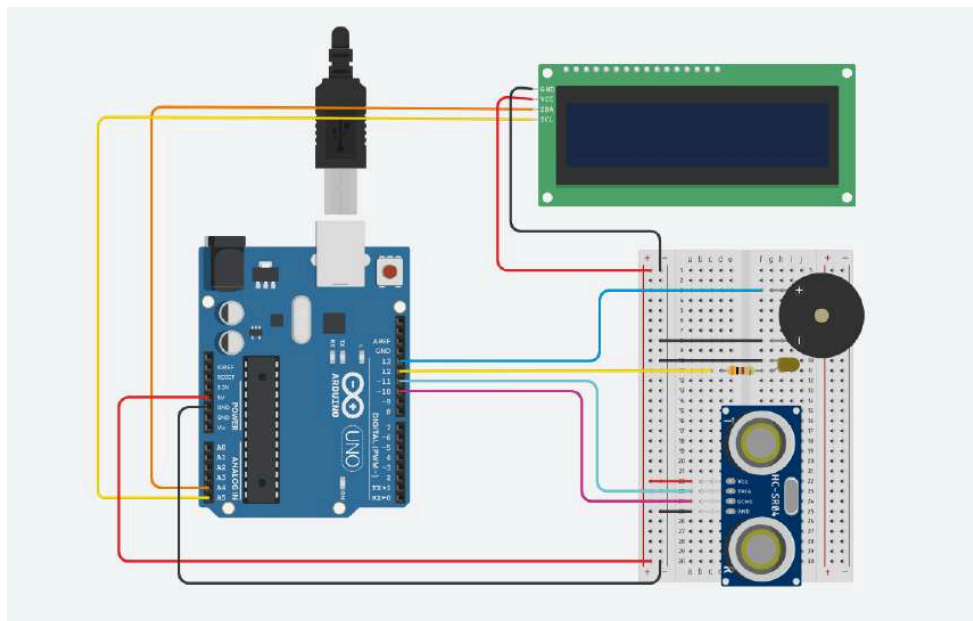
- 10. DIS DPST x4:** O DIS DPST x4 refere-se a um dispositivo de chaveamento que consiste em quatro interruptores de pólo-único-pólo-duplo (DPST), onde cada interruptor tem dois conjuntos de contatos que podem abrir e fechar simultaneamente. Esse tipo de dispositivo é frequentemente usado em sistemas de controle de energia ou em circuitos de comutação onde múltiplas cargas precisam ser controladas independentemente.





# PROJETO ESCOLHIDO PARA A APRESENTAÇÃO

O projeto escolhido foi o **"SPAD(Sensor de proximidade com alerta de distância)"** devido às suas características de simplicidade, ser baixo custo e ter diversas aplicações, podendo ser aplicado em automóveis como um assistente de estacionamento e de detecção de ponto cego, em robótica, auxiliando robôs com a finalidade de evitar colisões e navegar com segurança em ambientes industriais, em construções evitando que máquinas pesadas e operários se colidem, etc.



## **ANEXO A - S.P.A.D.(Sensor de proximidade com alerta de distância).**

```
#include <LiquidCrystal_I2C.h>
#define col 16
#define lin 2
#define ende 0x20
#define buzzerPin 13

int ECHO = 10;
int TRIG = 11;
int YELLOW = 12;
long duracao = 0;
long distancia = 0;
LiquidCrystal_I2C led(ende, col, lin);

void setup () {
    Serial.begin(9600);
    pinMode(ECHO,INPUT);
    pinMode(TRIG,OUTPUT);
    pinMode(YELLOW, OUTPUT);
    pinMode(buzzerPin,OUTPUT);
    led.init();
    led.backlight();
    led.clear();
}

void loop () {
    digitalWrite(TRIG,LOW);
    delay(10);
    digitalWrite(TRIG,HIGH);
    delay(10);
    digitalWrite(TRIG,LOW);
    duracao = pulseIn(ECHO,HIGH);
    distancia = duracao/58;
    led.setCursor(0, 0);
    led.print(distancia);
    led.print(" cm");
    delay(300);
    led.clear();

    if(distancia <= 85) {
        led.setCursor(0, 1);
        led.print("CUIDADO");
        digitalWrite(YELLOW, HIGH);
        tone(buzzerPin,500);
    }
    else {
        digitalWrite(YELLOW, LOW);
        noTone(buzzerPin);
    }
}
```

## ANEXO B - Medidor de chuva para o RS.

```
#include <LiquidCrystal.h>

LiquidCrystal lcd(7, 6, 5, 4, 3, 2);

#define ver 10 // definindo a entrada 2 como LED vermelho
#define ama 11 // definindo a entrada 4 como LED azul
#define verd 12 // definindo a entrada 7 como LED verde
#define umid A0 // definindo a entrada A0 como o nível de umidade
#define buzzerPin 8 // definindo o pino 8 como buzzer

int umidade = 0;

void setup() {
  pinMode(umid, INPUT);
  pinMode(ver, OUTPUT);
  pinMode(ama, OUTPUT);
  pinMode(verd, OUTPUT);
  pinMode(buzzerPin, OUTPUT); // configurando o pino do buzzer como saída
  Serial.begin(9600);

  // Configuração do LCD
  lcd.begin(16, 2);
  lcd.print("Umidade: ");
}

void loop() {
  umidade = analogRead(umid);
  Serial.println(umidade);

  // Atualizar o display LCD
  lcd.setCursor(9, 0); // posicionar cursor na coluna 9 da primeira linha
  lcd.print(" "); // limpar a área da umidade
  lcd.setCursor(9, 0); // reposicionar o cursor
  lcd.print(umidade); // exibir o valor da umidade

  if (umidade < 275) {
    digitalWrite(ver, HIGH);
    digitalWrite(ama, LOW);
    digitalWrite(verd, LOW);
    tone(buzzerPin, 1000, 200); // produz um som com frequência de 1000Hz
    lcd.setCursor(0, 1);
    lcd.print("Pouca umidade");
  } else if (umidade >= 275 && umidade < 600) {
    digitalWrite(ver, LOW);
    digitalWrite(ama, HIGH);
    digitalWrite(verd, LOW);
    tone(buzzerPin, 2000, 500); // produz um som com frequência de 2000Hz
    lcd.setCursor(0, 1);
    lcd.print("media umidade");
  } else {
    digitalWrite(ver, LOW);
    digitalWrite(ama, LOW);
    digitalWrite(verd, HIGH);
    tone(buzzerPin, 4000); // produz um som com frequência de 4000Hz
    lcd.setCursor(0, 1);
    lcd.print("grande umidade");
  }

  delay(1000); // aguarda 1 segundo
}
```

## ANEXO C - Sensor de ré.

```
int distance = 0;
double cm = 0;
int n = 0;

long readUltrasonicdistance(int triggerPin, int echoPin)
{
    pinMode(triggerPin, OUTPUT);
    digitalWrite(triggerPin, LOW);
    delayMicroseconds(1);
    digitalWrite(triggerPin, HIGH);
    delayMicroseconds(3);
    digitalWrite(triggerPin, LOW);
    pinMode(echoPin, INPUT);
    return pulseIn(echoPin, HIGH);
}

void calculaDistancia()
{
    distance = 300;
    cm = 0.017233 * readUltrasonicdistance(12, 11);
    Serial.print(cm);
    Serial.print("cm\n");
}

void setup()
{
    Serial.begin(11500);
    pinMode(3, OUTPUT);
    pinMode(5, OUTPUT);
    pinMode(6, OUTPUT);
    pinMode(10, OUTPUT);
}

void loop()
{
    calculaDistancia();

    do {
        digitalWrite(6, LOW);
        digitalWrite(5, LOW);
        digitalWrite(3, LOW);
        noTone(10);
        calculaDistancia();
    } while (cm > distance);
    do {
        digitalWrite(6, HIGH);
        digitalWrite(5, LOW);
        digitalWrite(3, LOW);
        tone(10, 100);
        delay(300);
        digitalWrite(6, LOW);
        noTone(10);
        delay(900);
        calculaDistancia();
    } while (cm <= distance && cm > distance - 100);
    do {
        digitalWrite(6, HIGH);
        digitalWrite(5, HIGH);
        digitalWrite(3, LOW);
        tone(10, 700);
        delay(300);
    }
```

```

    digitalWrite(6, LOW);
    digitalWrite(5, LOW);
    noTone(10);
    delay(500);
    calculaDistancia();
} while (cm <= distance - 100 && cm > distance - 200);
do {
    digitalWrite(6, HIGH);
    digitalWrite(5, HIGH);
    digitalWrite(3, HIGH);
    tone(10, 1500);
    delay(300);
    digitalWrite(6, LOW);
    digitalWrite(5, LOW);
    digitalWrite(3, LOW);
    noTone(10);
    delay(180);
    calculaDistancia();
} while (cm <= distance - 200 && cm > distance - 250);
do {
    digitalWrite(6, HIGH);
    digitalWrite(5, HIGH);
    digitalWrite(3, HIGH);
    tone(10, 2000);
    delay(100);
    calculaDistancia();
} while (cm <= distance - 250 && cm > distance - 300);
delay(1);
}

```