

Situação Aprendizagem *de IOT*

GRUPO: GABRIEL CUNHA, JEAN, MURILO E GABRIEL DOS SANTOS

Finalidade do projeto

- **Prevenir acidentes em áreas perigosas;**
- **Fornecer um alerta imediato em situações de risco;**
- **Aumentar a segurança em ambientes industriais, construção civil, e outros.**

Justificativa do projeto

- **Segurança e Prevenção de Acidentes;**
- **Simplicidade e Efetividade;**
- **Viabilidade de Implementação;**
- **Escalabilidade e Adaptabilidade;**
- **Custo-Benefício;**
- **Contribuição para a Sociedade.**

Onde utilizar

- **Ambientes Industriais;**
- **Canteiros de Obras e Fábricas;**
- **Escolas e Playgrounds.**

COMPONENTES



Figura 1 - Arduino Uno R3



Figura 2 - Placa de Ensaio



Figura 3 - Buzzer

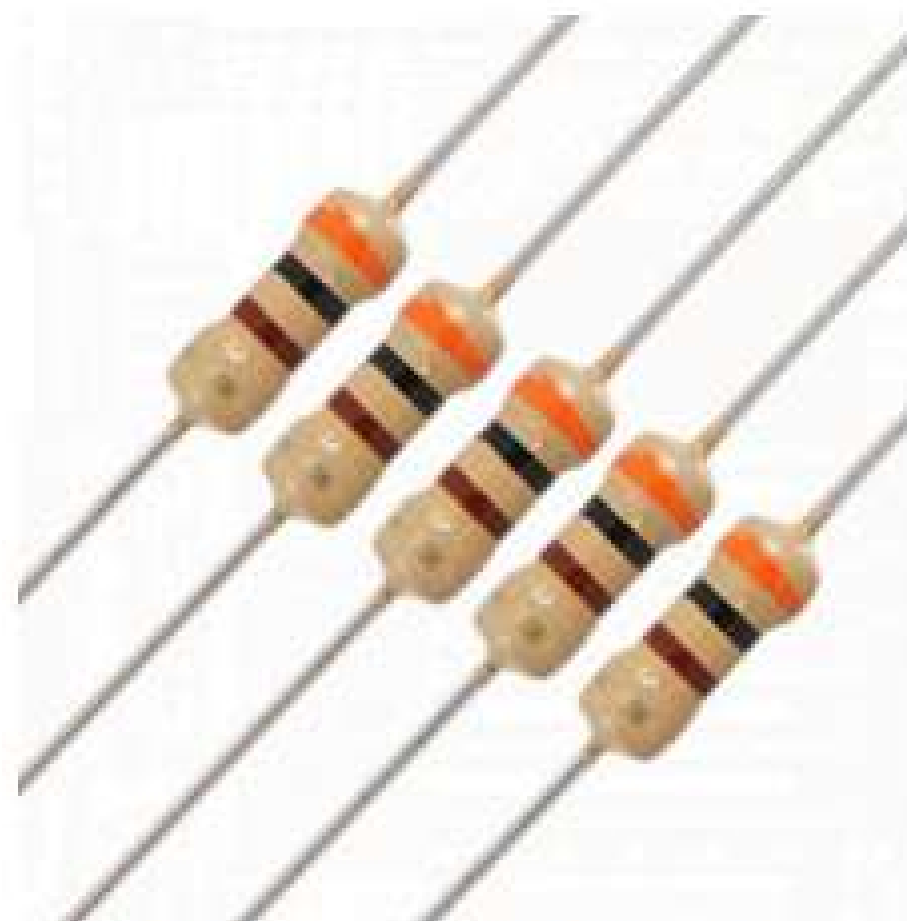


Figura 4 - Resistores de 300 ohms



Figura 5 - Led Vermelho, Verde e Amarelo



Figura 6 - Sensor de distância ultrassônico

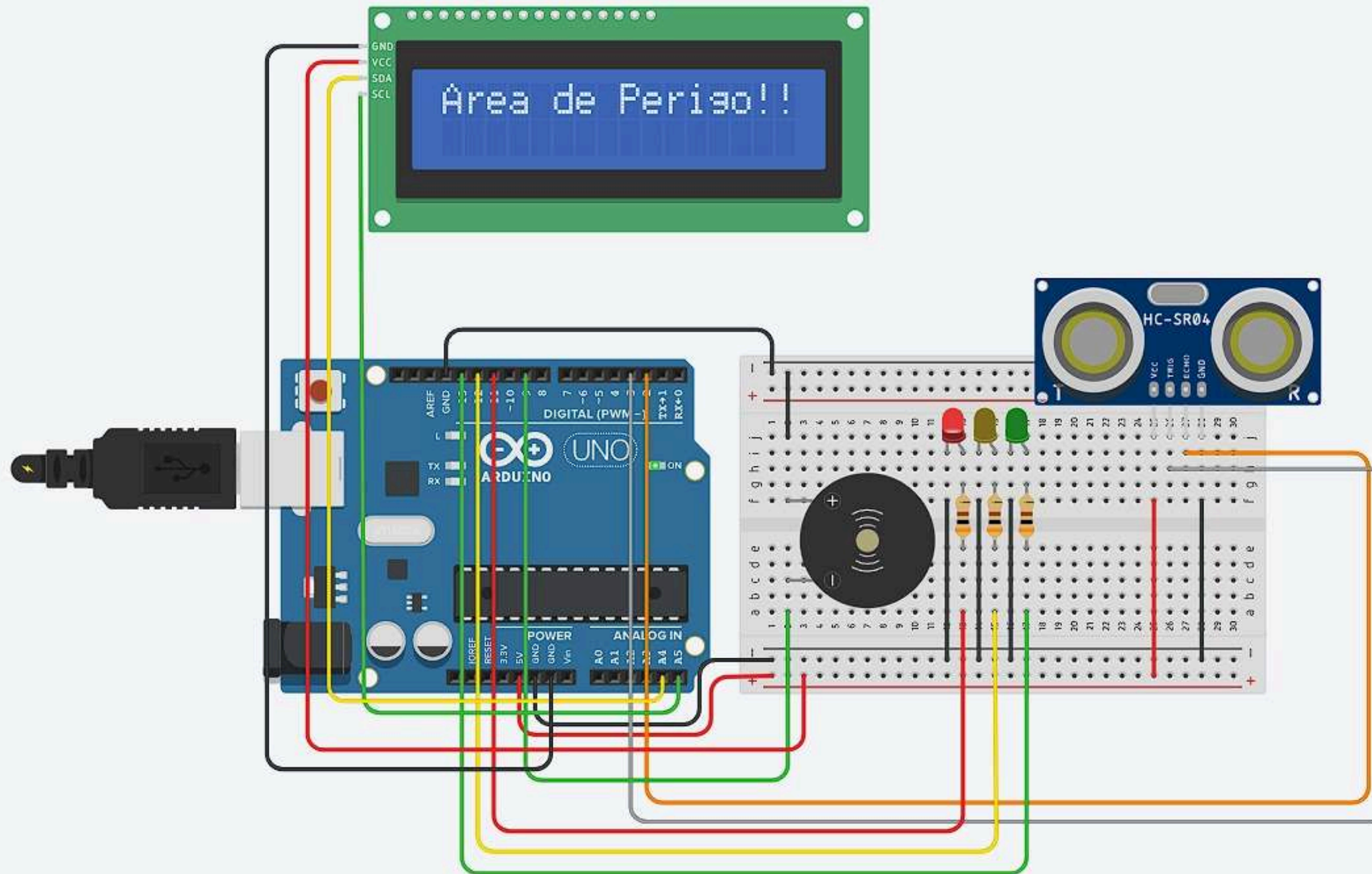


Figura 7 - Display LCD 16x2 (I2C)



Figura 8 - Jumpers

Tinkercad do projeto



Código do projeto

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>

// Inicializa o LCD no endereço 0x20 com 16 colunas e 2 linhas
LiquidCrystal_I2C lcd(0x20, 16, 2);

const int ECHO = 2; // Pino 2 recebe o pulso do sensor ultrassônico
const int TRIG = 3; // Pino 3 emite o pulso do sensor ultrassônico
const int ledVermelho = 11;
const int ledAmarelo = 12;
const int ledVerde = 13;
const int pino_buzzer = 9;

long duracao = 0; // Armazena a duração do pulso
long distancia = 0; // Armazena a distância calculada
String ultimoAlerta = ""; // Armazena a última mensagem exibida no LCD

void setup() {
  Serial.begin(9600); // Inicia a comunicação serial a 9600 bps

  pinMode(ECHO, INPUT); // Configura o pino ECHO como entrada
  pinMode(TRIG, OUTPUT); // Configura o pino TRIG como saída

  pinMode(ledVermelho, OUTPUT); // Configura o pino do LED vermelho como saída
  pinMode(ledAmarelo, OUTPUT); // Configura o pino do LED amarelo como saída
  pinMode(ledVerde, OUTPUT); // Configura o pino do LED verde como saída

  pinMode(pino_buzzer, OUTPUT); // Configura o pino do buzzer como saída

  lcd.init(); // Inicializa o LCD
  lcd.backlight(); // Liga a luz de fundo do LCD
}
```


Código do projeto

```
void loop() {  
  // Envia um pulso de disparo ao sensor ultrassônico  
  digitalWrite(TRIG, LOW);  
  delay(10);  
  
  digitalWrite(TRIG, HIGH);  
  delay(10);  
  
  digitalWrite(TRIG, LOW);  
  
  // Mede a duração do pulso recebido pelo pino ECHO  
  duracao = pulseIn(ECHO, HIGH);  
  
  // Calcula a distância em cm  
  distancia = duracao / 58;  
  String alertaAtual;  
  
  // Verifica a distância e atualiza LEDs e mensagem do LCD conforme a zona de segurança  
  if (distancia > 335) {  
    digitalWrite(ledVerde, HIGH);  
    digitalWrite(ledAmarelo, LOW);  
    digitalWrite(ledVermelho, LOW);  
  
    alertaAtual = "  Area Segura";  
  }  
  else if ((distancia < 335) && (distancia >= 165)) {  
    digitalWrite(ledVerde, LOW);  
    digitalWrite(ledAmarelo, HIGH);  
    digitalWrite(ledVermelho, LOW);  
  
    alertaAtual = "  Area de Risco";  
  }  
}
```


Código do projeto

```
}  
else if (distancia < 165) {  
    digitalWrite(ledVerde, LOW);  
    digitalWrite(ledAmarelo, LOW);  
    digitalWrite(ledVermelho, HIGH);  
  
    tone(pino_buzzer, 1200); // Emite som pelo buzzer  
    delay(250);  
    digitalWrite(ledVermelho, LOW);  
    noTone(pino_buzzer);  
    delay(250);  
    alertaAtual = "Area de Perigo!!";  
}  
  
// Atualiza o LCD somente se a mensagem mudou  
if (alertaAtual != ultimoAlerta) {  
    lcd.clear();  
    lcd.print(alertaAtual);  
    ultimoAlerta = alertaAtual; // Atualiza a última mensagem exibida  
}  
  
// Imprime a distância no monitor serial  
Serial.print("Distancia em cm: ");  
Serial.println(distancia);  
}
```

Obrigado!

