

SENAI NORTE I
TÉCNICO DESENVOLVIMENTO DE SISTEMAS
INTERNET DAS COISAS (IOT)

SITUAÇÃO APRENDIZAGEM

Gabriel da Cunha
Gabriel dos Santos
Jean Nack Oliveira
Murilo Celso Birckholz

Joinville, Maio de 2024

SUMÁRIO

1. INTRODUÇÃO.....	5
2. PROJETO: ALARME DE ZONA DE PERIGO.....	
5 2.1. Componentes utilizados.....	5
2.2. Código fonte: [Anexo 1].....	8
2.3. Projeto desenvolvido no Tinkercad.....	8
2.4. Aplicabilidade.....	8
2.5. Viabilidade da implantação.....	8
2.6. Custo total do projeto.....	9
3. PROJETO: SENSOR DE TEMPERATURA.....	10
3.1. Componentes utilizados.....	10
3.2. Código fonte: [Anexo 2].....	12
3.3. Projeto desenvolvido no Tinkercad.....	13
3.4. Aplicabilidade.....	13
3.5. Viabilidade da implantação.....	14
3.6. Custo total do projeto.....	15
4. PROJETO: UNIDADE DE MONITORAMENTO DE AMBIENTE.....	15
4.1. Componentes utilizados.....	15
4.2. Código fonte: [Anexo 3].....	19
4.3. Projeto desenvolvido no Tinkercad.....	19
4.4. Aplicabilidade.....	19
4.5. Viabilidade da implantação.....	20
4.6. Custo total do projeto.....	21
5. PROJETO: CONTROLE DE NÍVEL DE ÁGUA.....	21
5.1. Componentes utilizados.....	21
5.2. Código fonte: [Anexo 4].....	24
5.3. Projeto desenvolvido no Tinkercad.....	24
5.4. Aplicabilidade.....	24
5.5. Viabilidade da implantação.....	25
5.6. Custo total do projeto.....	25
6. PROJETO ESCOLHIDO: ALARME DE ZONA DE PERIGO.....	26
6.1. Foto do projeto.....	26
6.2. Justificativa.....	26
7. EXPLICAÇÃO DOS COMPONENTES UTILIZADOS.....	27
7.1. Resistor.....	27
7.2. Buzzer.....	28
7.3. Display LCD 16x2 (I2C).....	28
7.4. Sensor de Distância Ultrassônico.....	28
7.5. Placa de Ensaio.....	28
7.6. Jumpers.....	28
7.7. Arduino Uno R3.....	29

7.8. Motor DC.....	29
7.9. Sensor de Temperatura (LM35).....	29
7.10. Sensor de Gás MQ-5 GLP.....	29
7.11. Sensor de Luz do Ambiente.....	29
7.12. Potenciômetro.....	30
7.13. Sensor de Umidade.....	30
8. CONCLUSÃO.....	30
9. ANEXOS.....	30
9.1. Anexo 1: Código Fonte do Projeto Alarme de Zona de Perigo.....	30
9.2. Anexo 2: Código Fonte do Projeto Sensor de Temperatura.....	33
9.3. Anexo 3: Código Fonte do Projeto Unidade de Monitoramento de Ambiente.....	35
9.4. Anexo 4: Código Fonte do Projeto Controle de Nível de Água.....	37

LISTA DE FIGURAS

Figura 1 - Arduino Uno R3 5
 Figura 2 - Placa de Ensaio 5
 Figura 3 - Buzzer 6
 Figura 4 - Resistores de 300 ohms 6
 Figura 5 - Led Vermelho, Verde e Amarelo 6
 Figura 6 - Sensor de distância ultrassônico 7
 Figura 7 - Display LCD 16x2 (I2C) 7
 Figura 8 - Jumpers 7
 Figura 9 - Protótipo do projeto Alarme de Zona de Perigo 8
 Figura 10 - Arduino Uno R3 10
 Figura 11 - Placa de Ensaio 10
 Figura 12 - Sensor de Temperatura (LM35) 10
 Figura 13 - Led Verde, Azul, Vermelho e Amarelo 11
 Figura 14 - Buzzer 11
 Figura 15 - Motor DC 11
 Figura 16 - Resistores de 1K ohms 12
 Figura 17 - Jumpers 12
 Figura 18 - Protótipo do projeto Alarme de Zona de Perigo 13
 Figura 19 - Arduino Uno R3 16
 Figura 20 - Sensor de Temperatura (LM35) 16
 Figura 21 - Sensor de Gás MQ-5 GLP 16
 Figura 22 - Sensor de Luz do Ambiente 17
 Figura 23 - Display LCD 16x2 (I2C) 17
 Figura 24 - Potenciômetro 17
 Figura 25 - Resistor 220 Ohm 18
 Figura 26 - Resistor 4,7K Ohm 18
 Figura 27 - Resistor de 1K ohms 18
 Figura 28 - Placa de Ensaio 19
 Figura 29 - Protótipo do projeto Unidade de Monitoramento de Ambiente 19
 Figura 30 - Arduino Uno R3 21
 Figura 31 - Placa de Ensaio 22
 Figura 32 - Sensor de Umidade 22
 Figura 33 - Display LCD 16x2 (I2C) 22
 Figura 34 - Potenciômetro 23
 Figura 35 - Resistor de 1K ohms 23
 Figura 36 - Resistor 220 Ohm 23
 Figura 37 - Protótipo do projeto Unidade de Monitoramento de Ambiente 24
 Figura 38 - Projeto Desenvolvido na Prática 26

1. INTRODUÇÃO

Este relatório apresenta os projetos desenvolvidos pela equipe utilizando o componente Arduino. Cada membro da equipe elaborou um projeto individual, utilizando no mínimo quatro componentes. Dentre esses projetos, um foi escolhido para implementação prática coletiva. O objetivo é demonstrar a aplicabilidade e a

viabilidade de soluções IoT em diferentes contextos.

2. PROJETO: ALARME DE ZONA DE PERIGO

2.1. Componentes utilizados



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

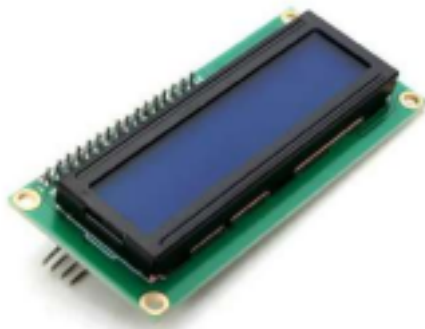


Figura 7 - Display LCD 16x2 (I2C)



Figura 8 - Jumpers

2.2. Código fonte: [Anexo 1]

2.3. Projeto desenvolvido no Tinkercad

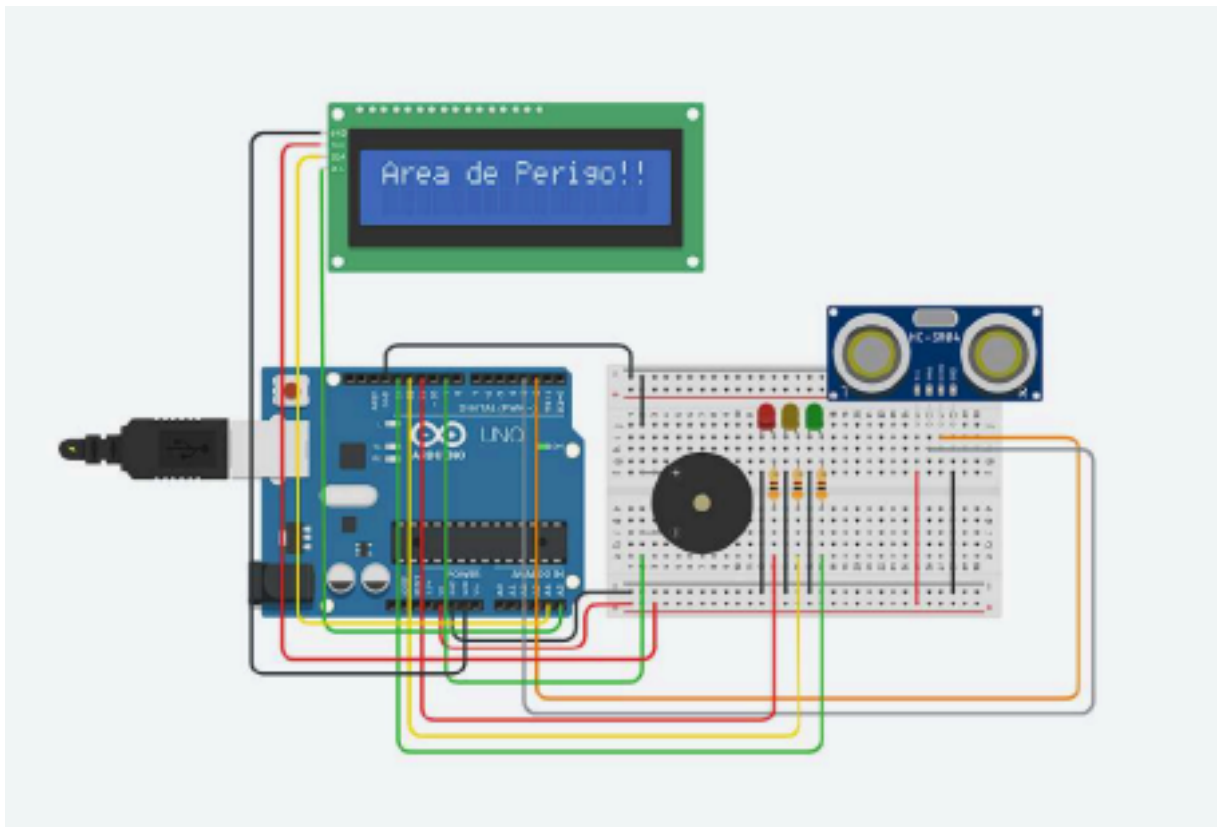


Figura 9 - Protótipo do projeto Alarme de Zona de Perigo

2.4. Aplicabilidade

O projeto "Alarme de Zona de Perigo" é altamente aplicável em ambientes industriais, canteiros de obras, fábricas e áreas de construção onde há zonas com diferentes níveis de risco. Ele pode ser utilizado para aumentar a segurança dos trabalhadores, avisando-os visualmente e auditivamente sobre os perigos presentes em determinadas áreas. Além disso, pode ser aplicado em escolas e playgrounds para manter crianças afastadas de áreas potencialmente perigosas.

2.5. Viabilidade da implantação

A viabilidade da implantação do projeto é alta, considerando os seguintes fatores:

- **Facilidade de Implementação:** Utilizando componentes comuns e de fácil acesso, a montagem do sistema é relativamente simples e pode ser realizada sem a necessidade de conhecimentos técnicos avançados.
- **Manutenção:** O sistema requer pouca manutenção, sendo necessário apenas

verificar periodicamente o funcionamento dos LEDs, do buzzer e do sensor ultrassônico.

- **Escalabilidade:** O projeto pode ser facilmente expandido para cobrir áreas maiores ou adicionar mais zonas de perigo, integrando mais sensores e LEDs.
- **Custo:** O custo total é acessível, tornando viável a implantação em diferentes locais sem grandes investimentos.

2.6. Custo total do projeto

O custo total do projeto é composto pelos seguintes componentes:

- Placa de Ensaio: R\$ 20,00
- Arduino Uno R3: R\$ 80,00
- LED Amarelo: R\$ 1,00
- LED Vermelho: R\$ 1,00
- LED Verde: R\$ 1,00
- Buzzer: R\$ 5,00
- Sensor de Distância Ultrassônico: R\$ 15,00
- Display LCD 16x2 (I2C): R\$ 40,00
- Resistores de 300 ohms: R\$ 0,60 (3 resistores a R\$ 0,20 cada)
- Jumpers: R\$ 10,00 (1 Pacote)

Portanto, o custo total estimado do projeto é de R\$ 173,60

3. PROJETO: SENSOR DE TEMPERATURA

3.1. Componentes utilizados



Figura 10 - Arduino Uno R3

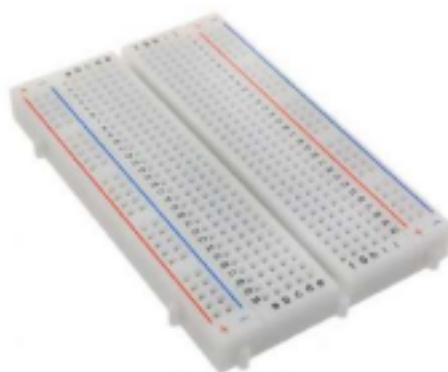


Figura 11 - Placa de Ensaio



Figura 12 - Sensor de Temperatura (LM35)



Figura 13
- Led Verde, Azul, Vermelho e Amarelo



Figura 14 - Buzzer



Figura 15 - Motor DC



Figura 16 - Resistores de 1K ohms



Figura 17 - Jumpers

3.2. Código fonte: [Anexo 2]

3.3. Projeto desenvolvido no Tinkercad

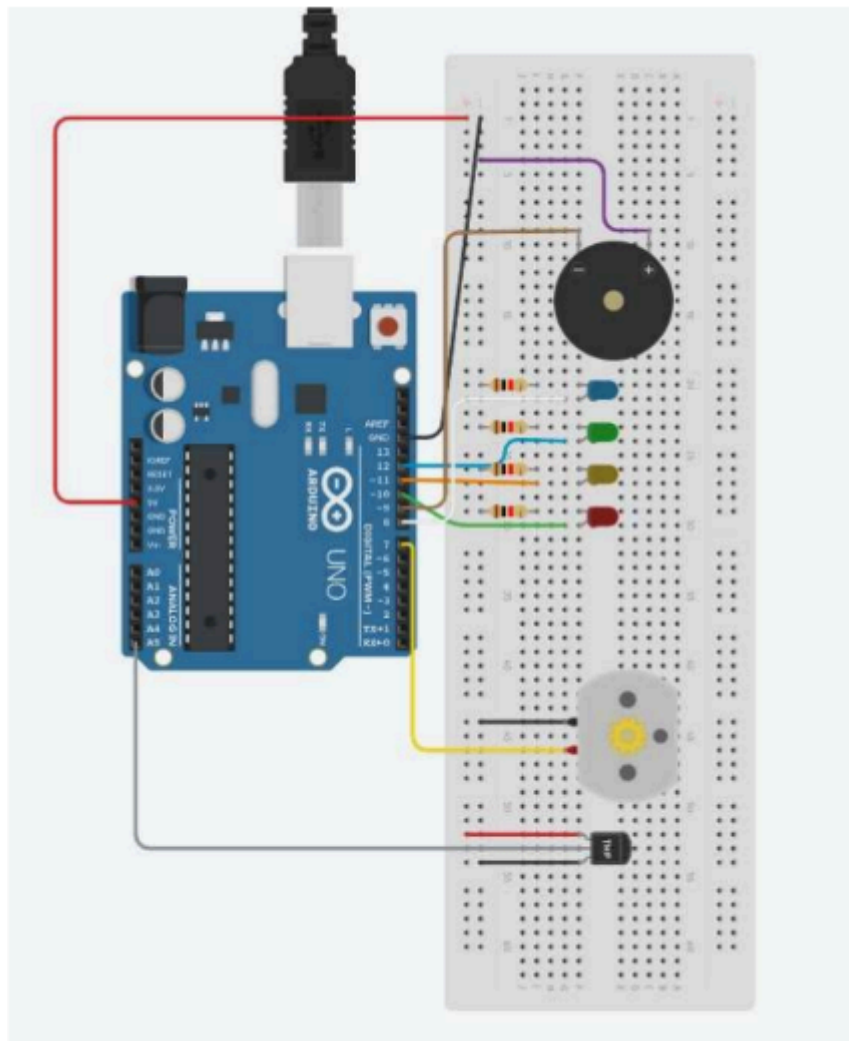


Figura 18 - Protótipo do projeto Alarme de Zona de Perigo

3.4. Aplicabilidade

Este sistema de sensor de temperatura tem uma ampla gama de aplicações industriais. Em ambientes controlados como laboratórios, data centers e salas limpas, o controle de temperatura é crucial. Por exemplo, em um laboratório, a precisão da temperatura pode ser vital para a realização de experimentos sensíveis. Nos data centers, um controle rigoroso da temperatura ajuda a evitar o superaquecimento dos servidores, garantindo o funcionamento contínuo e eficiente dos sistemas. Na indústria alimentícia e farmacêutica, o controle de sistemas de refrigeração é essencial para garantir a integridade dos produtos.

Este sistema pode monitorar e ajustar automaticamente a temperatura, evitando a deterioração de alimentos e medicamentos. Em incubadoras, tanto para ovos quanto para culturas biológicas, o controle preciso da temperatura aumenta significativamente a taxa de sucesso, otimizando os resultados para agricultores e pesquisadores. Nas linhas de produção industrial, o monitoramento térmico é essencial para processos como fundição, moldagem e fabricação de plásticos. A manutenção de temperaturas específicas garante a qualidade e uniformidade dos produtos finais, reduzindo o risco de defeitos e desperdícios. Na agricultura, especialmente em estufas, o controle climático é fundamental. O sistema pode manter uma temperatura ideal, promovendo o crescimento saudável das plantas e resultando em maior produtividade e qualidade das colheitas.

3.5. Viabilidade da implantação

A implementação deste sistema na indústria é viável e vantajosa por vários motivos. Em termos de custo-benefício, o sistema é acessível, com um custo total de menos de R\$ 200,00. Isso permite que pequenas e médias empresas possam adotar a tecnologia sem grandes investimentos iniciais. A facilidade de integração é outro ponto forte. O sistema é modular e pode ser facilmente incorporado a processos industriais existentes. Componentes como o Arduino são amplamente documentados e suportados, o que simplifica a adoção e integração, reduzindo a necessidade de treinamento extensivo. A manutenção do sistema é simples e econômica. Os componentes utilizados são facilmente substituíveis, o que minimiza o tempo de inatividade e reduz os custos operacionais.

Esta simplicidade de manutenção é crucial para operações industriais que necessitam de alta disponibilidade. Além disso, o sistema é escalável. Ele pode ser expandido para monitorar múltiplos pontos de temperatura e controlar várias saídas, permitindo a adaptação para sistemas mais complexos sem a necessidade de reestruturação significativa. A confiabilidade do sistema é garantida pelo uso de componentes de qualidade e uma programação robusta. Isso o torna adequado para ambientes industriais onde a precisão e a continuidade do funcionamento são

cruciais. A confiabilidade é essencial para garantir que os processos industriais não sejam interrompidos e que os produtos finais mantenham a qualidade desejada.

3.6. Custo total do projeto

O custo total do projeto é composto pelos seguintes componentes:

- Placa de Ensaio: R\$ 20,00
- Arduino Uno R3: R\$ 80,00
- Sensor de Temperatura (LM35): R\$ 20,00
- LED Azul: R\$ 1,00
- LED Verde: R\$ 1,00
- LED Amarelo: R\$ 1,00
- LED Vermelho: R\$ 1,00
- Buzzer: R\$ 5,00
- Motor DC: R\$ 8,00
- Resistores de 1K ohms: R\$ 0,80 (4 resistores a R\$ 0,20 cada)
- Jumpers: R\$ 10,00 (1 Pacote)

Portanto, o custo total estimado do projeto é de R\$ 187,80

4. PROJETO: UNIDADE DE MONITORAMENTO DE AMBIENTE

4.1. Componentes utilizados



Figura 19 - Arduino Uno R3



Figura 20 - Sensor de Temperatura(LM35)



Figura 21 - Sensor de Gás MQ-5 GLP



Figura 22 - Sensor de Luz do Ambiente

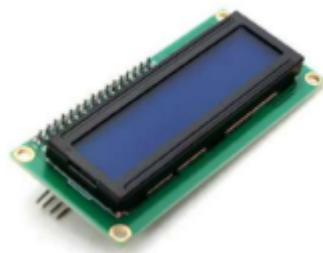


Figura 23 - Display LCD 16x2 (I2C)



Figura 24 - Potenciômetro



Figura 25 - Resistor 220 Ohm



Figura 26 - Resistor 4,7K Ohm



Figura 27 - Resistor de 1K ohms



Figura 28 - Placa de Ensaio

4.2. Código fonte: [Anexo 3]

4.3. Projeto desenvolvido no Tinkercad

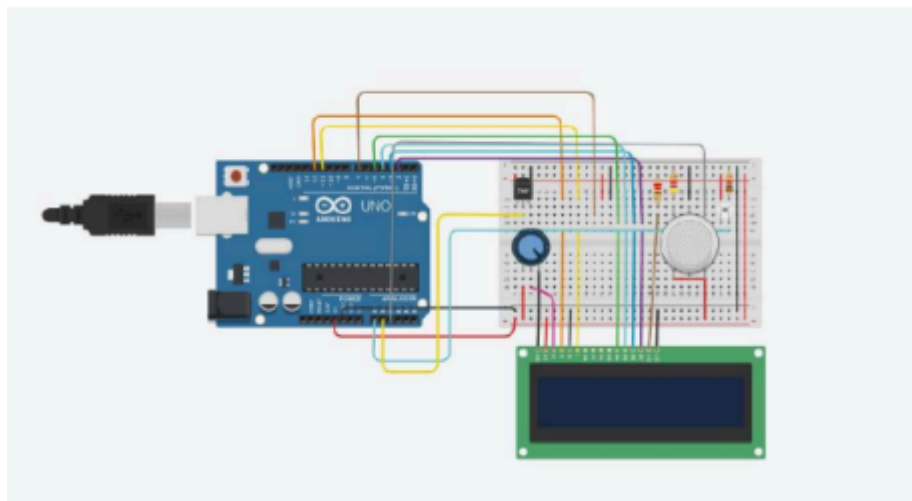


Figura 29 - Protótipo do projeto Unidade de Monitoramento de Ambiente

4.4. Aplicabilidade

O projeto de monitoramento de ambiente por meio de sensores tem uma ampla aplicabilidade em diferentes setores:

- **Controle de Qualidade do Ar em Ambientes Fechados:** Monitora poluentes, umidade e temperatura em locais internos como escritórios e residências,

alertando sobre condições adversas.

- **Monitoramento Ambiental em Espaços Públicos:** Fornece informações sobre a qualidade do ar e as condições ambientais em locais públicos como parques, praças e áreas de lazer, promovendo conscientização sobre questões relacionadas à poluição.
- **Agricultura Inteligente:** Monitora condições ambientais em estufas e ambientes de cultivo, auxiliando os agricultores a otimizar o crescimento das plantas ao ajustar o ambiente conforme necessário.
- **Eficiência Energética em Edifícios:** Controla a iluminação artificial com base na luminosidade ambiente, contribuindo para a redução do consumo de energia em edifícios comerciais e residenciais.

Essas aplicações demonstram a versatilidade do projeto e sua capacidade de se adaptar a diferentes necessidades e ambientes.

4.5. Viabilidade da implantação

O projeto demonstra uma viabilidade significativa de implantação, apesar da necessidade de adquirir os sensores de gás e de temperatura externamente, uma vez que não estão disponíveis no SENAI. A montagem rápida e fácil do projeto permite sua aplicação em diversos ambientes, desde residenciais até industriais, contanto que haja uma fonte de energia compatível com o Arduino UNO R3.

Além disso, a acessibilidade e disponibilidade geral dos componentes necessários para o projeto contribuem para sua viabilidade, já que os materiais são amplamente encontrados em lojas de eletrônicos e online, a preços acessíveis. Isso facilita a replicação do projeto em diferentes locais e contextos, sem a necessidade de recursos especializados ou investimentos significativos.

Considerando a simplicidade da montagem e a flexibilidade de aplicação, o projeto demonstra ser viável tanto para iniciantes quanto para profissionais, sendo uma solução prática e eficaz para monitoramento ambiental em diversos cenários.

4.6. Custo total do projeto

O custo total do projeto é composto pelos seguintes componentes:

- Arduino Uno R3: R\$ 80,00
- Sensor de Temperatura (LM35): R\$ 20,00
- Sensor de Gás MQ-5 GLP: R\$ 30,00
- Sensor de Luz do Ambiente: R\$ 10,00
- Display LCD 16x2: R\$ 40,00
- Potenciômetro: R\$ 5,00
- Resistor 220 Ohm: R\$ 0,50
- Resistor 4720 Ohm: R\$ 0,50
- Resistor 1k Ohm: R\$ 0,50
- Placa de Ensaio Pequena: R\$ 20,00

Portanto, o custo total estimado do projeto é de R\$ 206,00

5. PROJETO: CONTROLE DE NÍVEL DE ÁGUA

5.1. Componentes utilizados



Figura 30 - Arduino Uno R3



Figura 31 - Placa de Ensaio

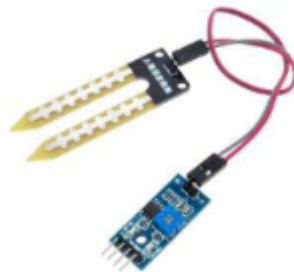


Figura 32 - Sensor de Umidade

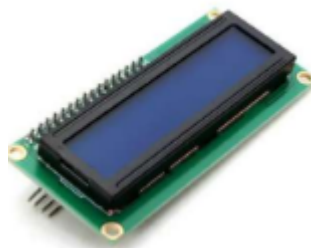


Figura 33 - Display LCD 16x2 (I2C)



Figura 34 - Potenciômetro



Figura 35 - Resistor de 1K ohms



Figura 36 - Resistor 220 Ohm

5.2. Código fonte: [Anexo 4]

5.3. Projeto desenvolvido no Tinkercad

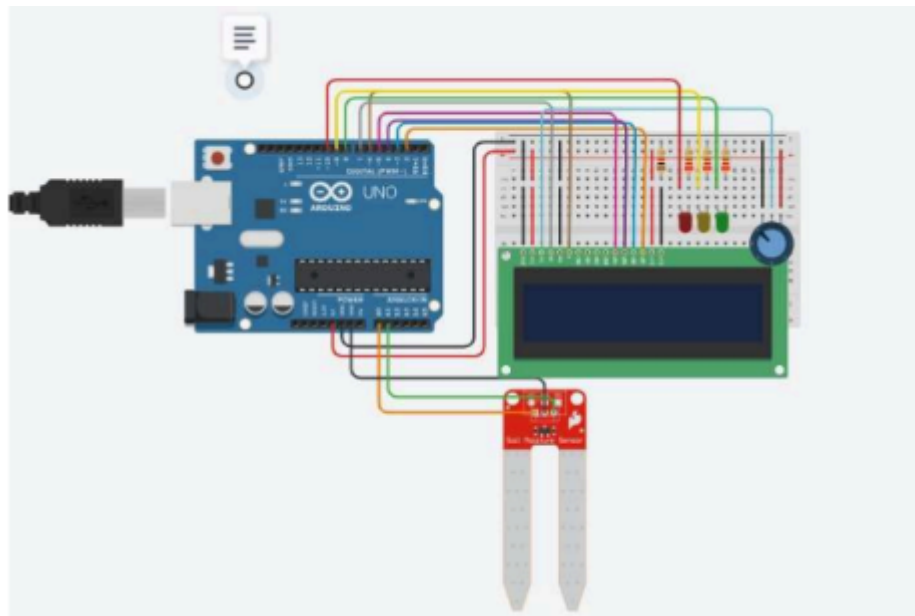


Figura 37 - Protótipo do projeto Unidade de Monitoramento de Ambiente

5.4. Aplicabilidade

A aplicabilidade desse projeto é voltada para o monitoramento de níveis em tanques industriais, sendo útil em diversos setores:

- **Postos de gasolina:** Monitoramento de combustível para evitar desabastecimento e vazamentos.
- **Indústrias químicas:** Controle de substâncias químicas para segurança e eficiência.

- **Tratamento de água:** Garantia de níveis adequados em cada etapa do tratamento.
- **Alimentícia e bebidas:** Gerenciamento de ingredientes líquidos e produtos finais.
- **Farmacêutica:** Precisão no controle de soluções e suspensões. ●
- Agrícola:** Otimização de tanques de irrigação e fertilizantes líquidos. ●
- Usinas de energia:** Controle de combustível e água de resfriamento. ●
- Embarcações:** Monitoramento de combustível e água de lastro. ●
- Petroquímicas:** Gestão de petróleo e derivados.

O projeto oferece soluções customizáveis para monitoramento em tempo real, automação e integração com sistemas de gestão.

5.5. Viabilidade da implantação

A viabilidade do projeto é bastante prática, especialmente considerando que todos os componentes necessários estão disponíveis no SENAI. Aqui estão os aspectos que reforçam essa viabilidade:

5.6. Custo total do projeto

O custo total do projeto é composto pelos seguintes componentes:

- Arduino Uno R3: R\$ 80,00
- Display LCD 16x2: R\$ 40,00
- Potenciômetro: R\$ 5,00
- Resistor 220 Ohm: R\$ 0,50
- Resistor 1k Ohm: R\$ 0,50
- Sensor de Umidade: R\$ 20,00
- Placa de Ensaio: R\$ 20,00

Portanto, o custo total estimado do projeto é de R\$ 166,00.

6. PROJETO ESCOLHIDO: ALARME DE ZONA DE PERIGO

6.1. Foto do projeto

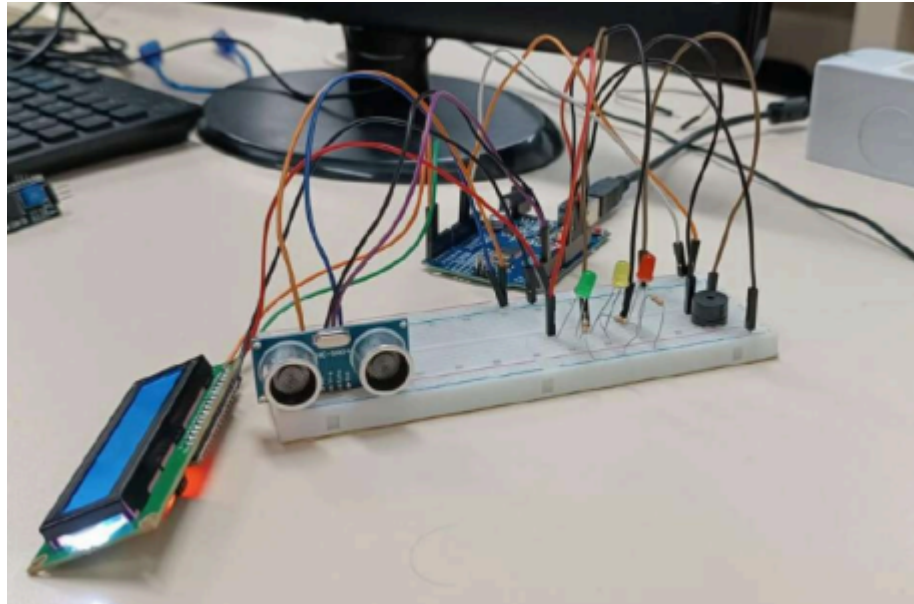


Figura 38 - Projeto Desenvolvido na Prática

6.2. Justificativa

O projeto "Alarme de Zona de Perigo" foi escolhido entre todos os projetos desenvolvidos por nossa equipe devido à sua relevância prática, impacto potencial e viabilidade de implementação. Aqui estão os principais motivos que justificaram nossa escolha:

- **Segurança e Prevenção de Acidentes:** O projeto é altamente aplicável em ambientes industriais, comerciais e públicos onde a segurança é uma prioridade. A capacidade de identificar e alertar sobre áreas de risco e perigo pode prevenir acidentes e salvar vidas. O sistema de sinalização com LEDs e alertas sonoros é uma solução eficiente para manter pessoas seguras em locais potencialmente perigosos.
- **Simplicidade e Efetividade:** Apesar de utilizar componentes simples e acessíveis, o projeto oferece uma solução robusta e eficaz para a detecção

de perigos. A combinação de LEDs, um buzzer e um sensor ultrassônico proporciona um sistema de alerta claro e imediato, que pode ser facilmente entendido e seguido por qualquer pessoa.

- **Viabilidade de Implementação:** O projeto "Alarme de Zona de Perigo" é de fácil montagem e não exige componentes caros ou difíceis de encontrar. A sua implementação é prática e rápida, tornando-o uma excelente opção para ambientes onde a instalação rápida é necessária. Além disso, a manutenção do sistema é simples e de baixo custo.
- **Escalabilidade e Adaptabilidade:** O sistema pode ser facilmente escalado e adaptado para diferentes ambientes e necessidades. Pode ser utilizado em diversos contextos, como fábricas, canteiros de obras, armazéns e áreas públicas. Sua flexibilidade permite que seja ajustado para detectar diferentes tipos de riscos, tornando-o uma solução versátil.
- **Custo-Benefício:** O custo total do projeto é relativamente baixo em comparação com o valor que proporciona em termos de segurança e prevenção de acidentes. A relação custo-benefício torna o projeto acessível para uma ampla gama de aplicações, desde pequenas empresas até grandes indústrias.
- **Contribuição para a Sociedade:** Ao melhorar a segurança em diversos ambientes, o projeto contribui para o bem-estar da sociedade como um todo. A prevenção de acidentes e a promoção de um ambiente seguro são benefícios inestimáveis que justificam plenamente a escolha deste projeto.

Diante desses fatores, o "Alarme de Zona de Perigo" foi selecionado por sua capacidade de proporcionar um impacto positivo significativo, sua fácil implementação e manutenção, além de sua aplicabilidade em diversos contextos, garantindo um ambiente mais seguro e protegido para todos.

7. EXPLICAÇÃO DOS COMPONENTES UTILIZADOS

Os componentes utilizados no Arduino são diversos e podem ser classificados em categorias como sensores, atuadores, componentes de comunicação, módulos de alimentação, entre outros.

Os componentes servem para diversas funções que permitem a criação de projetos interativos e automatizados. Cada tipo de componente desempenha um papel específico no funcionamento do sistema.

7.1. Resistor

Resistores são componentes que transformam energia elétrica em térmica pelo efeito Joule. Eles não possuem polaridade e são usados para limitar corrente e dividir tensões em circuitos. A leitura de seus valores é feita através das listras coloridas que indicam resistência e tolerância.

7.2. Buzzer

Buzzers são dispositivos que produzem som ao receber uma corrente elétrica. Existem tipos piezoelétricos e eletromagnéticos, usados para fornecer feedback auditivo em alarmes e sistemas de aviso. São polarizados, com terminais positivo e negativo.

7.3. Display LCD 16x2 (I2C)

O display LCD 16x2 pode exibir até 32 caracteres em duas linhas. A versão I2C utiliza apenas dois pinos para comunicação, facilitando a conexão com microcontroladores. É ideal para exibir informações textuais em projetos eletrônicos.

7.4. Sensor de Distância Ultrassônico

Sensores ultrassônicos, como o HC-SR04, medem a distância através de ondas sonoras. Eles emitem um pulso ultrassônico e calculam a distância com base no tempo de retorno do eco. São usados em robótica, automação e sistemas de segurança.

7.5. Placa de Ensaio

Placas de ensaio, ou breadboards, permitem a construção de circuitos temporários sem solda. Elas possuem uma matriz de furos conectados internamente para facilitar a inserção e conexão de componentes e fios, sendo úteis para prototipagem e testes.

7.6. Jumpers

Jumpers são conectores usados para alterar configurações de circuitos eletrônicos. Eles permitem conexões rápidas e ajustáveis em placas de ensaio e PCBs, sendo essenciais para prototipagem e configuração de hardware.

7.7. Arduino Uno R3

O Arduino Uno R3 é uma plataforma de desenvolvimento baseada no microcontrolador ATmega328P. Possui 14 pinos digitais e 6 entradas analógicas, e é amplamente utilizado em prototipagem, automação e ensino devido à sua facilidade de uso e vasta comunidade de suporte.

7.8. Motor DC

Motores DC convertem energia elétrica em energia mecânica rotacional. Eles são amplamente utilizados em robótica, automação e brinquedos. A velocidade e direção de rotação podem ser controladas ajustando a tensão e a polaridade da corrente fornecida.

7.9. Sensor de Temperatura (LM35)

O LM35 é um sensor de temperatura linear que fornece uma saída de tensão proporcional à temperatura em graus Celsius. Ele é preciso, fácil de usar e não requer calibração externa, sendo ideal para aplicações de monitoramento de temperatura.

7.10. Sensor de Gás MQ-5 GLP

O MQ-5 é um sensor de gás que detecta gases inflamáveis como GLP, metano e

propano. Ele gera uma saída analógica proporcional à concentração do gás, sendo utilizado em sistemas de detecção de vazamentos e segurança ambiental.

7.11. Sensor de Luz do Ambiente

Sensores de luz ambiente detectam a intensidade da luz ao redor, ajustando automaticamente a iluminação de telas e sistemas de iluminação inteligente. Eles são usados em dispositivos móveis, automação residencial e iluminação pública.

7.12. Potenciômetro

Potenciômetros são resistores ajustáveis que alteram a resistência ao girar um eixo. São usados para controlar volumes, ajustar níveis de sinal e calibrar dispositivos em circuitos eletrônicos, permitindo ajustes precisos e manuais.

7.13. Sensor de Umidade

Sensores de umidade medem a quantidade de vapor d'água no ar. Eles são usados em sistemas de controle climático, agricultura e armazenamento para monitorar e manter níveis ideais de umidade, garantindo condições adequadas para diversos processos.

8. CONCLUSÃO

O desenvolvimento dos projetos IoT com Arduino demonstrou a acessibilidade, versatilidade e aplicabilidade desta tecnologia em diversas áreas. Com a vasta comunidade e recursos disponíveis, até mesmo iniciantes podem criar soluções inovadoras. A modularidade dos projetos e a diversidade de componentes facilitam a adaptação conforme necessário. Essa acessibilidade combinada com a integração com outras tecnologias sugere um futuro onde a IoT baseada em Arduino desempenha um papel vital em setores diversos, desde a indústria até o cotidiano, melhorando eficiência e qualidade de vida.

9. ANEXOS

9.1. Anexo 1: Código Fonte do Projeto Alarme de Zona de Perigo

```
#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
}
```

```

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";
    }
    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);
}

```

9.2. Anexo 2: Código Fonte do Projeto Sensor de Temperatura

```

#define Vermelho 10
#define Amarelo 11
#define Verde 12
#define Azul 8
#define pinoBuzzer 9
#define SensorPin A1
#define Temp A5
#define Motor 7

int leitura;
void setup() {
    // Inicialização da serial
    Serial.begin(9600);

    // Configuração dos pinos
    pinMode(A5, INPUT);
    pinMode(Azul, OUTPUT);
    pinMode(Vermelho, OUTPUT);
    pinMode(Amarelo, OUTPUT);
    pinMode(Verde, OUTPUT);
}

```



```

pinMode(SensorPin, INPUT);
pinMode(pinoBuzzer, OUTPUT);
pinMode(Motor, OUTPUT);

// Estado inicial dos LEDs
digitalWrite(Azul, LOW);
digitalWrite(Vermelho, LOW);
digitalWrite(Amarelo, LOW);
digitalWrite(Verde, LOW);

}

void loop() {
    int tmp = analogRead(Temp);
    float voltage = (tmp * 5.0) / 1024;
    float milliVolt = voltage * 1000;
    float tmpCel = (milliVolt - 500) / 10;

    Serial.println(tmpCel);

    // Controle do motor
    if (tmpCel > 5 && tmpCel < 60) {
        Serial.println("Nivel De Temperatura: Baixo ");
        digitalWrite(Motor, HIGH);
        digitalWrite(Vermelho, LOW);
        digitalWrite(Amarelo, LOW);
        digitalWrite(Azul, LOW);
        digitalWrite(Verde, HIGH);
        noTone(pinoBuzzer);
    } else if (tmpCel > 60 && tmpCel < 100) {
        Serial.println("Nivel De Temperatura: Médio ");
        digitalWrite(Vermelho, LOW);
        digitalWrite(Amarelo, HIGH);
        digitalWrite(Verde, LOW);
        digitalWrite(Azul, LOW);
        digitalWrite(Motor, HIGH);
        noTone(pinoBuzzer);
    } else if (tmpCel > 100) {
        Serial.println("Nivel De Temperatura: Alto ");
    }
}

```

```

        digitalWrite(Vermelho, HIGH);
        digitalWrite(Amarelo, LOW);
        digitalWrite(Verde, LOW);
        digitalWrite(Azul, LOW);
        digitalWrite(Motor, LOW);
        tone(pinoBuzzer, 900);
        delay(300);
        noTone(pinoBuzzer);
        delay(300);
    } else if (tmpCel < 5) {
        Serial.println("Nivel De Temperatura: Muito baixo ");

        digitalWrite(Vermelho, LOW);
        digitalWrite(Amarelo, LOW);
        digitalWrite(Verde, LOW);
        digitalWrite(Azul, HIGH);
        digitalWrite(Motor, LOW);
        tone(pinoBuzzer, 20);
        delay(300);
        noTone(pinoBuzzer);
        delay(300);
    }

    delay(100); // Pequeno atraso para estabilidade
}

```

9.3. Anexo 3: Código Fonte do Projeto Unidade de Monitoramento de Ambiente

```

#include <LiquidCrystal.h>
#define sLuz A0
#define sTemp A1
#define sGas A2
LiquidCrystal lcd(12,11,5,4,3,2);

void setup() {
    lcd.begin(16, 2);
}

void loop() {
    nivelluz();
    limparTela();
}

```

```

    temperatura();
    limparTela();
    gas();
    limparTela();
}

void limparTela(){
    delay(2000);
    lcd.setCursor(0,0);
    lcd.print(" ");
    lcd.setCursor(0,1);
    lcd.print(" ");
    delay(1000);
}

void nivelluz(){
    int luz = analogRead(sLuz);
    luz=map(luz,0,471,0,10);
    lcd.setCursor(0,0);
    lcd.print("Nivel de luz:");
    delay(500);
    lcd.setCursor(0,1);
    lcd.print(luz);
    delay(200);
    lcd.print(" ");
}

void temperatura(){
    float temp=analogRead(sTemp)*5;
    float tempC=((temp/1023)-0.5)*100;
    lcd.setCursor(0,0);
    lcd.print("Temperatura: ");
    lcd.setCursor(0,1);
    lcd.print("No momento ");
    lcd.print(tempC);
}

void gas(){
    int gas = analogRead(sGas);
    gas = map(gas,300,750,0,100);
    lcd.setCursor(0,0);
    lcd.print("Nivel do Gas:");

```

```

    lcd.setCursor(0,1);
    lcd.print("Atual: ");
    lcd.print(gas);
    delay(200);

}

```

9.4. Anexo 4: Código Fonte do Projeto Controle de Nível de Água

```

#include <LiquidCrystal.h>
LiquidCrystal lcd(7, 6, 5, 4, 3, 2);

void setup() {
  lcd.begin(16, 2);
  pinMode(A0, OUTPUT);
  pinMode(A1, INPUT);
  pinMode(8, OUTPUT); // Verde
  pinMode(9, OUTPUT); // Amarelo
  pinMode(10, OUTPUT); // Vermelho
}

void loop() {

  int moisture;
  // Para inicializar o sensor
  digitalWrite(A0, HIGH);
  // Lê o valor do sensor de nível de água
  moisture = analogRead(A1);
  // Desliga o pino
  digitalWrite(A0, LOW);
  // Imprime o valor do sensor no console serial
  Serial.println(moisture);

  digitalWrite(8,LOW);
  digitalWrite(9,LOW);
  digitalWrite(10,LOW);

  // Variáveis para fazer o controle do volume de água do

```

```

reservatório
int lowThreshold = 200;
int mediumThreshold = 600;

if (moisture < lowThreshold) {
    digitalWrite(10, HIGH); // Acende o LED vermelho

    // Exibe "Baixo" no LCD

    lcd.setCursor(0, 1);
    lcd.print("Nivel de agua:");
    lcd.setCursor(0, 0);
    lcd.print("Baixo");

} else if (moisture < mediumThreshold) {
    digitalWrite(9, HIGH); // Acende o LED amarelo

    // Exibe "Médio" no LCD
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("Nivel de agua:");
    lcd.setCursor(0, 1);
    lcd.print("Medio");
} else {
    digitalWrite(8, HIGH); // Acende o LED verde
    // Exibe "Alto" no LCD
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("Nivel de agua:");
    lcd.setCursor(0, 1);
    lcd.print("Alto");
}

delay(100); // Aguarda um breve intervalo antes de repetir o loop
}

```