SENAI

(SERVIÇO NACIONAL DE APRENDIZAGEM INDUSTRIAL)

**Eduardo da Costa, Leonardo Gonçalves Dias,**

**Vinícius Pupo Estevam**

**TÍTULO: S.A (Situação de Aprendizagem)**

Joinville

2024

**SUMÁRIO**

[1. Componentes 3](#_Toc170157139)

[2. Sistema de Trava de Segurança 9](#_Toc170157140)

[2.1 Componentes: 9](#_Toc170157141)

[2.2 Aplicabilidade: 10](#_Toc170157142)

[2.3 Viabilidade de Implantação: 10](#_Toc170157143)

[2.4 Custo total do projeto: 10](#_Toc170157144)

[2.5 R$180,80. 10](#_Toc170157145)

[3. Sistema de Alerta de Movimento 10](#_Toc170157146)

[3.1 Componentes: 10](#_Toc170157147)

[3.2 Aplicabilidade: 11](#_Toc170157148)

[3.3 Viabilidade da Implantação: 11](#_Toc170157149)

[3.4 Custo total do projeto: 11](#_Toc170157150)

[3.5 R$190,90. 11](#_Toc170157151)

[4. Projeto Escolhido: Sensor de Temperatura 12](#_Toc170157152)

[4.1 Componentes: 12](#_Toc170157153)

[4.2 Aplicabilidade: 12](#_Toc170157154)

[4.3 Viabilidade da Implantação: 12](#_Toc170157155)

[4.4 Custo total do projeto: 13](#_Toc170157156)

[4.5 Justificativa da Escolha: 13](#_Toc170157157)

[5. Anexos 13](#_Toc170157158)

[5.1 Anexo 1 Sistema de Trava de Segurança 13](#_Toc170157159)

[5.2 Anexo 2: Sistema de Alerta de Movimento 17](#_Toc170157160)

[5.3 Anexo 3: Sensor de Temperatura 20](#_Toc170157161)

[6. Referências 24](#_Toc170157162)

[6.1 Link 1 https://www.tinkercad.com/things/6vdQJ1oTDA9-copy-of-sensor-de-temperatura-/editel?tenant=circuits 24](#_Toc170157163)

[6.2 Link 2 https://www.tinkercad.com/things/9d1s5Jga6y7-copy-of-sensor-de-presenca/editel?tenant=circuits 24](#_Toc170157164)

[6.3 Link 3https://www.tinkercad.com/things/4x2rKkvfGYW-copy-of-sensor-de-temperatura/editel?tenant=circuits 24](#_Toc170157165)

LISTA DE ILUSTRAÇÕES

[Figura 1 - Arduino Uno 3](#_Toc169990597)

[Figura 2 - Protoboard 4](#_Toc169990598)

[Figura 3 - Jumpers 4](#_Toc169990599)

[Figura 4 - Resistores 5](#_Toc169990600)

[Figura 5 - LCD 5](#_Toc169990601)

[Figura 6 – LED 6](#_Toc169990602)

[Figura 7 - Buzzer 6](#_Toc169990603)

[Figura 8 - Sensor de PIR 7](#_Toc169990604)

[Figura 9 - Sensor de Temperatura 7](#_Toc169990605)

[Figura 10 - O Teclado Matricial 4x4 8](#_Toc169990606)

[Figura 11 - O Micro Servo Motor SG90 8](#_Toc169990607)

[Figura 12 - O sensor de vibração 9](#_Toc169990608)

[Figura 13 - Sistema de Trava de Segurança 9](#_Toc169990609)

[Figura 14 - Alerta de Movimento 11](#_Toc169990610)

[Figura 15 - Sensor de Temperatura 12](#_Toc169990611)

[Figura 16 - Imagem do Projeto Montado 13](#_Toc169990612)

# Componentes

Os componentes são aqueles que permitem com que os projetos possuam funcionalidades. Neste tópico, serão explicados os componentes utilizados nos três projetos que serão apresentados posteriormente.

Circuito eletrônico com fios

Descrição gerada automaticamente

Figura 1 - Arduino Uno

Placa Arduino Uno: Placa microcontroladora utilizada para receber e enviar sinais aos componentes por meio de pinos 14 de entrada e saída digital e 6 de entrada analógica, podendo realizar a interação com software ao ser conectado ao computador por meio de um cabo USB.



Figura 2 - Protoboard

Protoboard: a protoboard é uma placa de ensaio que serve como um protótipo de um aparelho eletrônico, com uma matriz de contatos que possibilita construir circuitos de teste sem que haja necessidade de solda e, assim, garantindo segurança e agilidade em diferentes atividades.



Figura 3 - Jumpers

Jumpers: Pequenos fios condutores utilizados para conexões entre dois pontos em um circuito.

Colher de metal

Descrição gerada automaticamente com confiança média

Figura 4 - Resistores

Resistores: Dispositivos utilizados para controle da passagem de corrente elétrica em circuitos, através da conversão de energia elétrica em energia térmica.

Interface gráfica do usuário, Texto

Descrição gerada automaticamente

Figura 5 - LCD

LCD: Os aparelhos de tela de cristal líquido (ou Liquid Crystal Display, no original em inglês que dá origem à sigla LCD) funcionam a partir de uma iluminação traseira (backlight) do tipo fluorescente por trás da tela para formar a imagem.

Uma imagem contendo Linha do tempo

Descrição gerada automaticamente

Figura 6 – LED

LED: O LED é um diodo semicondutor (junção P-N), que quando é energizado, emite luz visível – por isso o nome "LED" (Diodo Emissor de Luz). A luz não é monocromática (como em um laser), mas consiste em uma banda espectral relativamente estreita e é produzida pelas interações energéticas do elétron.

Em preto e branco

Descrição gerada automaticamente com confiança média

Figura 7 - Buzzer

Buzzer: É um componente eletrônico que converte um sinal elétrico em onda sonora, este dispositivo é utilizado para sinalização sonora, sendo aplicado em computadores, despertadores, carros, entre outros.

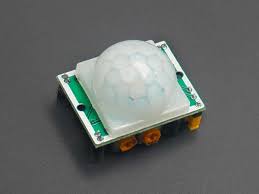


Figura 8 - Sensor de PIR

Sensor de PIR: O Sensor PIR funciona detectando variações no calor emitido por objetos em seu campo de visão. Ele contém uma lente especial que segmenta o campo de visão em várias zonas. Quando um objeto se move dentro dessas zonas, ele altera a quantidade de radiação infravermelha detectada por cada segmento.



Figura 9 - Sensor de Temperatura

O Sensor de Temperatura: TMP36 é um circuito integrado medidor de temperatura que possui encapsulamento TO-92 e que tem aparência de um transistor de 3 terminais.

Uma imagem contendo Quadro de comunicações

Descrição gerada automaticamente

Figura 10 - O Teclado Matricial 4x4

O Teclado Matricial 4x4: foi desenvolvido para facilitar a entrada de dados em projetos microcontrolados. Este teclado possui 16 teclas, onde 10 delas são números, 4 são letras e 2 são caracteres especiais.

Uma imagem contendo Quadro de comunicações

Descrição gerada automaticamente

Figura 11 - O Micro Servo Motor SG90

O Micro Servo Motor SG90: é um motor muito utilizado em aplicações para robótica, nos sistemas microcontroladores, como, por exemplo, Arduino, PIC e AVR.

Desenho de uma pessoa

Descrição gerada automaticamente com confiança baixa

Figura 12 - O sensor de vibração

O sensor de vibração: é um equipamento extremamente usado na indústria. Máquinas que apresentam alguma oscilação durante seu funcionamento podem ser identificadas de forma rápida e eficiente pelo sensor de vibração, o que acaba por garantir que não ocorram maiores danos subsequentes.

# Sistema de Trava de Segurança

## Componentes:

2x LED, 1x Placa Arduino Uno, 1x Protoboard, Jumpers, 2x resistores de 1k Ohms, 1x Buzzer, 1x teclado4x4, 1x micro servo.

Diagrama

Descrição gerada automaticamente

Figura 13 - Sistema de Trava de Segurança

## Aplicabilidade:

O sistema de trava de segurança baseado em IoT tem ampla aplicabilidade em áreas que necessitam de controle de acesso seguro, como residências, escritórios, e instalações industriais. Ele permite gerenciar o acesso a portas, armários, e áreas restritas, garantindo que apenas pessoas autorizadas possam entrar.

## Viabilidade de Implantação:

A viabilidade do sistema é alta devido à simplicidade e confiabilidade dos componentes envolvidos. Teclados 4x4, micro servos, buzzers e LEDs são fáceis de integrar e programar. A instalação e manutenção são relativamente simples, tornando o sistema acessível para uma ampla variedade de usuários e cenários de uso.

## Custo total do projeto:

## R$180,80.

# Sistema de Alerta de Movimento

## Componentes:

1x Arduino uno, 1x Protoboard , 1x resistor 1k Ohms, 1x LED 5mm vermelho, 1x Arduino Uno R3, Jumpers macho/macho, 1x Buzzer Piezo, 1x Sensor PIR, 1x Motor vibracall.

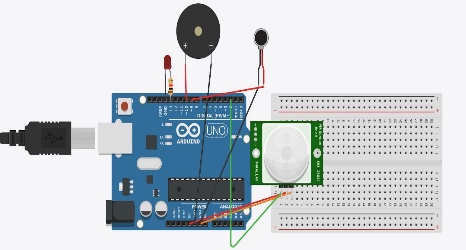


Figura 14 - Alerta de Movimento

## Aplicabilidade:

A aplicabilidade do sistema de sensor de alerta de movimento é vasta e inclui áreas como segurança residencial, monitoramento de áreas restritas em indústrias, e automação de iluminação em espaços comerciais. Esse sistema pode detectar intrusões, ativar alarmes, e acionar mecanismos de resposta rápida, aumentando a segurança e eficiência em diversos ambientes.

## Viabilidade da Implantação:

A viabilidade do sistema de sensor de alerta de movimento é alta, graças à disponibilidade e acessibilidade dos componentes utilizados, como o buzzer, LED, motor de vibração e sensor PIR. Além disso, a simplicidade de integração e a necessidade de pouca manutenção contribuem para a relação custo-benefício positiva, facilitando sua adoção em diferentes ambientes industriais e residenciais.

## Custo total do projeto:

## R$190,90.

# Projeto Escolhido: Sensor de Temperatura

## Componentes:

1x Arduino uno, 1x Protoboard, 1x Sensor de temperatura, 1x Buzzer, 2x Leds, 2x resistores de 1k,  *1x LCD i2c.*

Tela de um aparelho eletrônico

Descrição gerada automaticamente com confiança média

Figura 15 - Sensor de Temperatura

## Aplicabilidade:

sistema de sensor de temperatura é amplamente aplicável em diversas áreas, como controle de ambientes industriais, monitoramento de sistemas de climatização em edifícios comerciais, e gerenciamento de condições em instalações agrícolas. Ele ajuda a garantir que as condições térmicas permaneçam dentro de parâmetros seguros e eficientes, prevenindo danos a equipamentos e produtos.

## Viabilidade da Implantação:

A viabilidade do sistema é alta devido à simplicidade e à robustez dos componentes utilizados. Sensores de temperatura, buzzers, LEDs e displays LCD são amplamente disponíveis e bem documentados, facilitando a sua integração em sistemas maiores. A instalação e a manutenção são simples, o que torna o sistema acessível para uma ampla gama de usuários, desde técnicos a operadores menos experientes.

## Custo total do projeto:

R$190,66.

## Justificativa da Escolha:

Este projeto foi escolhido porque era o mais adequado para uma aplicação em grande escala. Seu preço baixo e a possibilidade de melhorias, permitindo a incorporação de práticas sustentáveis nos locais onde será instalado, o tornam uma excelente escolha.

Uma imagem contendo eletrônico, circuito

Descrição gerada automaticamente

Figura 16 - Imagem do Projeto Montado

# Anexos

## Anexo 1 Sistema de Trava de Segurança

#include <Keypad.h>

#include <Servo.h>

Servo servo\_Motor;

char\* password = "586";

int posicao = 0;

const byte linha = 4;

const byte coluna = 4;

char keys[linha][coluna] = {

{'1','2','3','A'},

{'4','5','6','B'},

{'7','8','9','C'},

{'\*','0','#','D'}

};

byte linPins[linha] = {9,8,7,6};

byte colPins[coluna] = {5,4,3,2};

Keypad keypad = Keypad(makeKeymap(keys), linPins, colPins, linha, coluna);

const int ledvermelho = 12;

const int ledverde = 11;

const int buzzerPin = 10; // Definindo o pino do buzzer

void setup() {

pinMode(ledvermelho, OUTPUT);

pinMode(ledverde, OUTPUT);

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

servo\_Motor.attach(13);

setLocked(true);

}

void loop() {

char key = keypad.getKey();

if (key == '\*' || key == '#') {

posicao = 0;

setLocked(true);

}

if (key == password[posicao]) {

posicao++;

}

if (posicao == 3) {

setLocked(false);

}

delay(100);

}

void setLocked(int locked) {

if (locked) {

digitalWrite(ledvermelho, HIGH);

digitalWrite(ledverde, LOW);

servo\_Motor.write(0);

tone(buzzerPin, 1000, 500); // Som indicando senha incorreta

delay(500); // Pequeno intervalo para distinguir os sons

noTone(buzzerPin); // Desliga o buzzer

} else {

digitalWrite(ledverde, HIGH);

digitalWrite(ledvermelho, LOW);

servo\_Motor.write(90);

tone(buzzerPin, 2000, 1000); // Som indicando senha correta

delay(1000); // Pequeno intervalo para distinguir os sons

noTone(buzzerPin); // Desliga o buzzer

}

}

## Anexo 2: Sistema de Alerta de Movimento

int ledPin = 13;

int inputPin = 2;

int pirState = LOW;

int val = 0;

int pinSpeaker = 10;

int motorPin = 9;

void setup(){

pinMode(ledPin, OUTPUT);

pinMode(inputPin, INPUT);

pinMode(pinSpeaker, OUTPUT);

pinMode(motorPin, OUTPUT);

Serial.begin(9600);

}

void loop() {

val = digitalRead(inputPin);

if (val == HIGH) {

digitalWrite(ledPin, HIGH);

digitalWrite(motorPin, HIGH);

playTone(300, 160);

delay(150);

if (pirState == LOW) {

Serial.println("Motion detected!");

pirState = HIGH;

}

} else {

digitalWrite(ledPin, LOW);

digitalWrite(motorPin, LOW);

playTone(0, 0);

delay(300);

if (pirState == HIGH) {

Serial.println("Motion ended!");

pirState = LOW;

}

}

}

void playTone(long duration, int freq) {

duration \*= 1000;

int period = (1.0 / freq) \* 1000000;

long elapsed\_time = 0;

while (elapsed\_time < duration) {

digitalWrite(pinSpeaker, HIGH);

delayMicroseconds(period / 2);

digitalWrite(pinSpeaker, LOW);

delayMicroseconds(period / 2);

elapsed\_time += (period);

}

}

## Anexo 3: Sensor de Temperatura

//### BIBLIOTECA UTILIZADAS ###

// Biblioteca LCD

#include <LiquidCrystal.h>

// Inicializa a biblioteca LCD

LiquidCrystal LCD(12, 11, 5, 4, 3, 2);

//### DEFINIR OS PINO UTILIZADOS ###

// Define o PINO ANALÓGICO A0 como entrada do Sensor de Temperatura

const int SensorTempPino = A0;

// Define o PINO 8 para o alerta de temperatura baixa

const int AlertaTempBaixa = 8;

// Define o PINO 13 para o alerta de temperatura alta

const int AlertaTempAlta = 13;

// Define o pino do BUZZER

const int PiezoPino = 7;

//### DEFINIR OS LIMITE DE TEMPERATURA ###

// Define temperatura baixa como abaixo de 16 graus Celsius

const int TempBaixa = 16;

// Define temperatura alta como acima de 25 graus Celsius

const int TempAlta = 25;

//### PINOS DE SAÍDA ###

void setup() {

// Define o pino de alerta de temperatura baixa como saída

pinMode(AlertaTempBaixa, OUTPUT);

// Define o pino de alerta de temperatura alta como saída

pinMode(AlertaTempAlta, OUTPUT);

// Define o pino do piezo como saída

pinMode(PiezoPino, OUTPUT);

// ### CONFIGURANDO DISPLAY LCD ###

// Define a quantidade de colunas e linhas do LCD

LCD.begin(16, 2);

// Imprime a mensagem no LCD

LCD.print("Temperatura:");

// Muda o cursor para a primeira coluna e segunda linha do LCD

LCD.setCursor(0, 1);

// Imprime a mensagem no LCD

LCD.print(" C F");

}

//### AÇÕES PARA CALCULAR A TEMPERATURA ###

void loop() {

// Faz a leitura da tensão no Sensor de Temperatura

int SensorTempTensao = analogRead(SensorTempPino);

// Converte a tensão lida em Graus Celsius

float TemperaturaC = ((SensorTempTensao \* 5.0) / 1024 - 0.5) \* 100.0;

// Converte a temperatura em Graus Celsius para Fahrenheit

float TemperaturaF = (TemperaturaC \* 9.0 / 5.0) + 32.0;

// Muda o cursor para a primeira coluna e segunda linha do LCD

LCD.setCursor(0, 1);

// Imprime a temperatura em Graus Celsius

LCD.print(TemperaturaC);

// Muda o cursor para a décima coluna e segunda linha do LCD

LCD.setCursor(9, 1);

// Imprime a temperatura em Graus Fahrenheit

LCD.print(TemperaturaF);

//### LIGA OS LED ###

// Acende ou apaga os alertas luminosos de temperatura baixa e alta

if (TemperaturaC >= TempAlta) {

digitalWrite(AlertaTempBaixa, LOW);

digitalWrite(AlertaTempAlta, HIGH);

// ### AÇÕES DO BUZZER ###

// Ativa o piezo para temperatura alta

for (int i = 0; i < 10; i++) {

tone(PiezoPino, 500); // Frequência do som em Hz

delay(100); // Liga o buzzer por 100ms

noTone(PiezoPino);

delay(100); // Desliga o buzzer por 100ms

}

} else if (TemperaturaC < TempBaixa) {

digitalWrite(AlertaTempBaixa, HIGH);

digitalWrite(AlertaTempAlta, LOW);

// Ativa o piezo para temperatura baixa

for (int i = 0; i < 10; i++) {

tone(PiezoPino, 350); // Frequência do som em Hz

delay(50); // Liga o buzzer por 100ms

noTone(PiezoPino);

delay(50); // Desliga o buzzer por 100ms

}

} else {

digitalWrite(AlertaTempBaixa, LOW);

digitalWrite(AlertaTempAlta, LOW);

// Desativa o buzzer

noTone(PiezoPino);

}

// Aguarda 1 segundo antes da próxima leitura

delay(1000);

}

# Referências

## Link 1 <https://www.tinkercad.com/things/6vdQJ1oTDA9-copy-of-sensor-de-temperatura-/editel?tenant=circuits>

## Link 2 <https://www.tinkercad.com/things/9d1s5Jga6y7-copy-of-sensor-de-presenca/editel?tenant=circuits>

## Link 3<https://www.tinkercad.com/things/4x2rKkvfGYW-copy-of-sensor-de-temperatura/editel?tenant=circuits>