

# Sistema de Controle de Acesso (fechadura eletrônica)

## 1. Descrição do Projeto

Este projeto consiste na implementação de um sistema de controle de acesso utilizando um Arduino, um teclado matricial, um display LCD com interface I2C e um servo motor. O sistema permite que um usuário digite uma senha no teclado para abrir uma porta controlada pelo servo motor. Além disso, possui um modo de administração que permite alterar a senha de acesso.

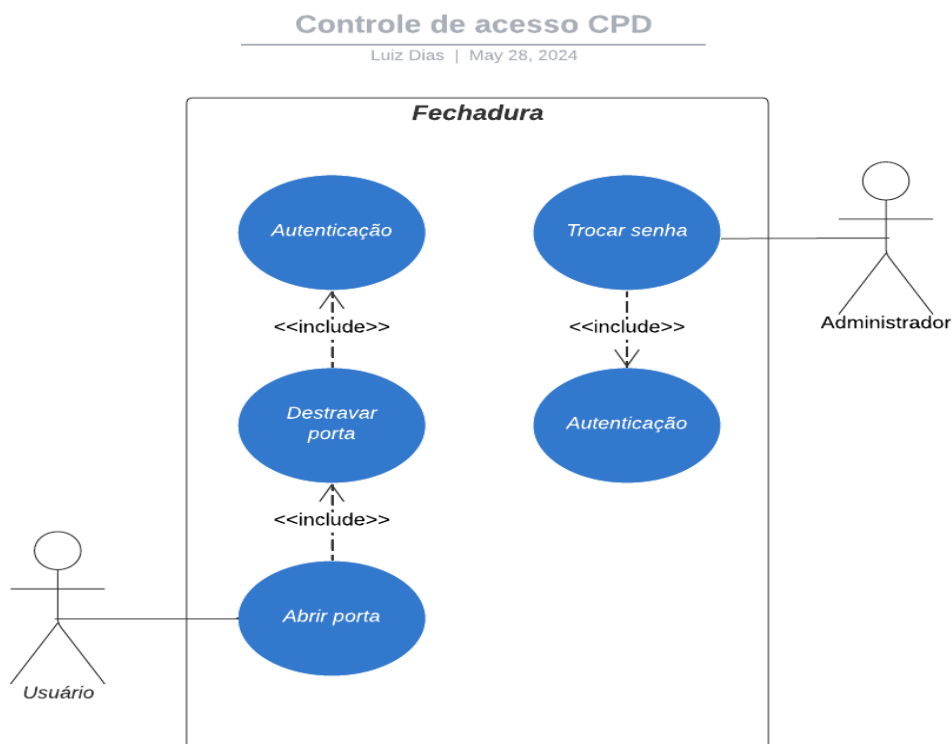
## 2. Componentes Utilizados

- Arduino Uno
- Display LCD I2C 16x2 (LiquidCrystal\_I2C), baseado em PCF8574
- Teclado Matricial 4x4 (Keypad)
- Servo Motor (Servo)
- Fios Jumper

## 3. Conexões

- Display LCD I2C:
  - Conectado ao barramento I2C (SDA, SCL) do Arduino.
- Teclado Matricial:
  - Linhas conectadas aos pinos digitais 6, 7, 8 e 9 do Arduino.
  - Colunas conectadas aos pinos digitais 10, 11, 12 e 13 do Arduino.
- Servo Motor:
  - Sinal conectado ao pino analógico A0 do Arduino.
  - Alimentação conectada aos pinos 5V e GND do Arduino.

## 4. Diagrama



## 5. Código Fonte

### 5.1. Inclusão de bibliotecas:

```
include <LiquidCrystal_I2C.h>
include <Keypad.h>
include <Servo.h>
```

Aqui, estamos incluindo as bibliotecas necessárias para o funcionamento do código. LiquidCrystal\_I2C.h é utilizada para controlar um display LCD via I2C, Keypad.h para a leitura de um teclado matricial, e Servo.h para controlar um servo motor.

### 5.2. Definição de variáveis e objetos:

```
LiquidCrystal_I2C lcd(0x20, 16, 2);
const byte numLinhasTeclado = 4;
const byte numColunasTeclado = 4;
char senhaCorreta[5] = "1234";
char entradaUsuario[5];
byte contadorEntradas = 0;

byte pinosLinhasTeclado[numLinhasTeclado] = {6, 7, 8, 9};
byte pinosColunasTeclado[numColunasTeclado] = {10, 11, 12, 13};
char senhaAdm[] = "0800";

char teclasTeclado[numLinhasTeclado][numColunasTeclado] = {
    {'1', '2', '3', 'A'},
    {'4', '5', '6', 'B'},
    {'7', '8', '9', 'C'},
    {' ', '0', ' ', 'D'}
};

Servo servoMotor;
Keypad teclado = Keypad(makeKeymap(teclasTeclado), pinosLinhasTeclado,
pinosColunasTeclado, numLinhasTeclado, numColunasTeclado);
```

Aqui, estamos definindo variáveis e objetos necessários para o funcionamento do sistema.

- lcd: objeto do tipo LiquidCrystal\_I2C para controlar o display LCD.
- numLinhasTeclado e numColunasTeclado: dimensões do teclado matricial.
- senhaCorreta: senha de acesso ao sistema.
- entradaUsuario: vetor para armazenar a senha digitada pelo usuário.
- contadorEntradas: contador para rastrear o número de teclas pressionadas.
- pinosLinhasTeclado e pinosColunasTeclado: mapeamento dos pinos do teclado matricial.
- senhaAdm: senha de administrador.
- teclasTeclado: mapeamento das teclas do teclado matricial.
- servoMotor: objeto do tipo Servo para controlar o servo motor.
- teclado: objeto do tipo Keypad para a leitura do teclado matricial.

### 5.3. Função setup():

```
void setup() {  
  lcd.init();  
  lcd.backlight();  
  inicial();  
  servoMotor.attach(A0, 544, 2400);  
  servoMotor.write(0);  
}
```

Esta função é executada uma vez quando o Arduino é inicializado. Aqui, estamos inicializando o display LCD, configurando a tela inicial, conectando o servo motor a um pino e definindo a posição inicial do servo motor.

### 5.4. Função loop():

```
void loop() {  
  char teclaDigitada = teclado.getKey();  
  if ((teclaDigitada != 0) && (contadorEntradas < 4)) {  
    entradaUsuario[contadorEntradas] = teclaDigitada;  
    limparSenha();  
    inicial();  
  }  
}
```

Esta função é executada continuamente após a inicialização. Aqui, estamos lendo as teclas pressionadas no teclado, armazenando as teclas pressionadas, exibindo as teclas no LCD, verificando se a senha foi inserida corretamente e tomando ações apropriadas com base nisso.

### 5.5. Função inicial():

```
void inicial() {  
  lcd.clear();  
  lcd.setCursor(0, 0);  
  lcd.print("  Sala CPD ");  
  lcd.setCursor(0, 1);  
  lcd.print("Senha: ");  
}
```

Esta função é responsável por configurar a tela inicial do LCD. Ela faz o seguinte:

- Limpa o LCD para garantir que não haja informações anteriores na tela.
- Define o cursor na posição (0, 0), que é a primeira linha e a primeira coluna do display.
- Exibe a mensagem "Sala CPD" no display, centralizada horizontalmente.
- Define o cursor na posição (0, 1), que é a segunda linha e a primeira coluna do display.
- Exibe a mensagem "Senha: " no display, indicando ao usuário que ele deve digitar a senha.

## 5.6. Função limparSenha():

```
void limparSenha() {  
    for (int i = 0; i < 5; i++) {  
        entradaUsuario[i] = '\0';  
    }  
    contadorEntradas = 0;  
}
```

Esta função é responsável por limpar a senha digitada pelo usuário, resetando o vetor de entrada e o contador de entradas. Ela faz o seguinte:

- Utiliza um loop for para percorrer cada posição do vetor entradaUsuario.
- Atribui o valor '\0' (caractere nulo) a cada posição do vetor, limpando assim a senha digitada anteriormente.
- Reseta o contador de entradas, garantindo que a próxima senha digitada comece a ser armazenada a partir da primeira posição do vetor.

## 5.7. Função alterarSenha():

```
void alterarSenha() {  
    lcd.clear();  
    lcd.setCursor(0, 0);  
    lcd.print("Nova Senha: ");  
    delay(1000);  
  
    char novaSenhaUsuario[5];  
    byte contadorNovaSenha = 0;  
  
    while (contadorNovaSenha < 4) {  
        char teclaNovaSenha = teclado.getKey();  
        if (teclaNovaSenha) {  
            novaSenhaUsuario[contadorNovaSenha] = teclaNovaSenha;  
            lcd.setCursor((contadorNovaSenha + 6), 1);  
            lcd.print(teclaNovaSenha);  
            contadorNovaSenha++;  
        }  
    }  
  
    novaSenhaUsuario[4] = '\0';  
    strcpy(senhaCorreta, novaSenhaUsuario);  
  
    lcd.clear();  
    lcd.setCursor(0, 0);  
    lcd.print("Senha Alterada!");  
    delay(2000);  
}
```

Esta função permite ao usuário alterar a senha. Ela faz o seguinte:

- Limpa o LCD para uma nova tela.
- Exibe a mensagem "Nova Senha: " no display, indicando ao usuário que ele deve digitar a nova senha.
- Aguarda um segundo para dar tempo ao usuário de se preparar para digitar a nova senha.
- Inicia um loop while para permitir que o usuário digite a nova senha.
- A cada tecla pressionada, a função atualiza a tela do LCD exibindo a tecla pressionada e armazenando a nova senha no vetor novaSenhaUsuario.
- Quando todas as quatro teclas da nova senha são digitadas, a função copia a nova senha para a variável senhaCorreta.
- Exibe uma mensagem de confirmação no display, indicando que a senha foi alterada com sucesso.

## 6. Funcionamento

### 1. Inicialização:

- O LCD é inicializado e a tela de boas-vindas é exibida.
- O servo motor é configurado para a posição inicial (0 graus).

### 2. Entrada da Senha:

- O usuário digita a senha usando o teclado matricial.
- A primeira tecla digitada é exibida no LCD, as demais são substituídas por asteriscos.

### 3. Verificação da Senha:

- Se a senha estiver correta, o servo motor se move para 90 graus, liberando a porta e uma mensagem de "Porta Liberada!" é exibida no LCD.
- Se a senha estiver incorreta, uma mensagem de "Senha Incorreta!" é exibida no LCD.
- Se a senha for de administrador, entra no modo de administrador para poder alterar a senha de acesso.

### 4. Reinicialização:

- Após 3 segundos, o sistema é reinicializado, trancando a porta e voltando à tela de boas-vindas.

## 7. Link do Projeto

Você pode acessar o projeto no Tinkercad através do seguinte link: [Controle de Acesso](#)

Você pode acessar o projeto no GitHub através do seguinte link: [GitHub](#)