

/*

Título: Controle de Acesso com Arduino

Descrição:

Você foi contratado para desenvolver um sistema de controle de acesso simples usando Arduino. O sistema consiste em um teclado matricial de 4x4 e um display LCD 16x2. O objetivo é permitir que um usuário insira um código de acesso de 4 dígitos. Se o código estiver correto, uma mensagem de "Acesso Permitido" será exibida no LCD; caso contrário, será exibida uma mensagem de "Acesso Negado".

Requisitos:

- Caso não possua senhas cadastradas, permitir acesso sem senha
- Após cadastrar o usuário, permitir acesso somente com senha
- Use a tecla # para confirmar
- Use a tecla * para cancelar
- Utilize as letras para navegar entre as funcionalidades

Funcionalidades

1. Usuário Logado:

- Cadastrar novas senhas
- Excluir senhas
- Mostrar senhas cadastradas
- Logoff

2. Usuário deslogado

- Login

Funcionalidade Adicional (Opcional):

Implemente um mecanismo de bloqueio temporário após um número definido de tentativas fracassadas. Permita que o código de acesso seja alterado pelo usuário através de uma interface simples.

*/

```
#include <LiquidCrystal.h>
```

```
// conexão física nos pinos
```

```
LiquidCrystal lcd(0, 1, 2, 3, 4, 5);
```

```
const int rows[] = {6, 7, 8, 9};
```

```
const int cols[] = {10, 11, 12, 13};
```

```
// mapear o teclado
```

```
const char keys[4][4] = {{'1', '2', '3', 'A'},  
                          {'4', '5', '6', 'B'},  
                          {'7', '8', '9', 'C'},  
                          {'*', '0', '#', 'D'}};
```

```
// vars global
```

```
char choice = 'B';
```

```
//senhas
```

```
int password[4][10];
```

```
//senhas cadastradas
```

```
int setPassword = 0;
```

```

void setup()
{
    // inicialização do display lcd
    lcd.begin(16,2);
    // iniciação o teclado matricial
    for (int i=0; i<4; i++) {
        pinMode(rows[i], INPUT_PULLUP);
        pinMode(cols[i], OUTPUT);
        digitalWrite(cols[i], HIGH);
    }
    lcd.print("Entrar sem senha");
}

// main
void loop() {
    // chama a função que verifica e retorna se alguma tecla for pressionada
    char key = keyboardScan();
    // se retornar 0 = nulo, não tem tecla pressionada

    if (key > 0) {
        delay(500); // evitar considerar acionamentos sequenciais
    }
    login(key);
}

// login inicial
void login(char op) {
    // com senha
    if(op == 'A' && setPassword > 0){
        lcd.setCursor(0,0);
        lcd.print("Entrar");
        choice = op;
    } //sem senha
    else if(op == 'B' && setPassword < 1){
        lcd.setCursor(0,0);
        lcd.print("Entrar sem senha");
        choice = op;
    }
    // confirmacao
    if(op == '#' && choice == 'A')
        loginPassword();
    else if(op == '#' && choice == 'B')
        loginNoPassword();
}

// func opções de entrada
void loginPassword(){
    int digits = 0;
    int inputPassword[4];

    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print("Digite a senha");

    //entrada

```

```

while(digits < 4){
    char digit = keyboardScan();
    if (digit > 0) {
        delay(300);
    }
    if(digit > 0 && digit != '*' && digit != '#'){
        inputPassword[digits] = digit;
        digits++;
        lcd.setCursor(digits,1);
        lcd.print(digit);
        delay(300);
    }
}
lcd.clear();

//validação
bool validate = false;
for(int j=setPassword; j>=0;j--){
    for(int i=0; i<4; i++){
        if(inputPassword[i] != password[i][setPassword-j])
            break;
        else
            validate = true;
    }
    if(validate){
        lcd.print("Acesso Permitido");
        delay(300);
        choice = 'A';
        menu();
    }
    else{
        lcd.print("Acesso Negado");
        delay(500);
        lcd.clear();
        lcd.setCursor(0,0);
        lcd.print("Entrar");
        choice = 'A';
    }
}

// menu logado
void menu() {
    bool logoff = false;
    int digits=0;
    choice = 'A';
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print("Cadastrar");
    lcd.setCursor(0,1);
    lcd.print("novas senhas");

    while(!logoff){

        char digit = keyboardScan();
        if (digit > 0) {

```

```

    delay(300);
}

// opções
if(digit == 'A'){
    choice = 'A';
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print("Cadastrar");
    lcd.setCursor(0,1);
    lcd.print("novas senhas");

}
if(digit == 'B'){
    choice = 'B';
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print("Excluir senhas");
}
if(digit == 'C') {
    choice = 'C';
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print("Mostrar senhas");
}
if(digit == 'D') {
    choice = 'D';
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print("Logoff");
}
//mostrar

// logoff
if(choice == 'D' && digit == '#'){
    lcd.clear();
    lcd.print("Logoff...");
    delay(400);
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print("Entrar");
    choice = 'A';
    logoff = true;
}
}
}

// sem senha
void loginNoPassword() {
    // quantidade de digitos
    int digits = 0;

    lcd.clear();

```

```

lcd.print("Acesso Permitido");
delay(400);
lcd.clear();
while(digits < 4){
  lcd.setCursor(0,0);
  lcd.print("Cadastrar senha");
  // entrada
  char digit = keyboardScan();
  if (digit > 0) {
    delay(300);
  }
  if(digit > 0 && digit != '*' && digit != '#'){
    password[digits][setPassword] = digit;
    digits++;
    lcd.setCursor(digits,1);
    lcd.print(digit);
    delay(300);
  }
}
// sucesso no cadastro
lcd.clear();
lcd.setCursor(0,0);
lcd.print("Senha cadastrada");
lcd.setCursor(0,1);
lcd.print("com sucesso!");
delay(300);
// tela de inicio
lcd.clear();
lcd.setCursor(0,0);
lcd.print("Entrar");
choice = 'A';
setPassword++;
}

```

```

char keyboardScan() {
  // retorno da função - padrão retorna nulo
  char ret = 0;

  for (int i=0; i<4; i++) {
    // colocar o pino de saída (coluna) em nível lógico baixo
    digitalWrite(cols[i], LOW);
    // verificar em todas as entradas (linhas)
    // se tem algum sinal em nível lógico baixo (botão pressionado)
    for (int j=0; j<4; j++) {
      // verificar se tem sinal nos pinos de entrada (linhas) em nível lógico baixo
      if (digitalRead(rows[j]) == LOW) {
        ret = keys[j][i];
        // escreve a coluna e linha pressionada
        // lcd.print("Col: " + String(i) + " - Lin: " + String(j));
      }
    }
    // colocar o pino de saída (coluna) em nível lógico alto
    digitalWrite(cols[i], HIGH);
  }
  // retorno da função
}

```

```
    return ret;  
}
```