

/*

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};
```

```
// Mapeamento do teclado
```

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

```
// Definindo as senhas
```

```
String passwords[10] = {""}; // Até 10 senhas
```

```
int passwordCount = 0;
```

```
String input = "";
```

```
bool loggedIn = false;
```

```
bool isAdminRemoved = false; // Flag para verificar se a senha Admin foi removida
```

```
const int maxAttempts = 3;
```

```
int attempts = 0;
```

```
unsigned long blockTime = 0;
```

```
const unsigned long blockDuration = 15000; // 15 segundos
```

```
void setup() {
```

```
    // Inicialização do display LCD
```

```
    lcd.begin(16, 2);
```

```
    lcd.print("Digite a senha:");
```

```
    // Inicialização do teclado matricial
```

```
    for (int i = 0; i < 4; i++) {
```

```

        pinMode(rows[i], INPUT_PULLUP);
        pinMode(cols[i], OUTPUT);
        digitalWrite(cols[i], HIGH);
    }
}

void loop() {
    if (millis() < blockTime) {
        lcd.clear();
        lcd.print("Bloqueado!");
        delay(1000);
        return;
    } else if (millis() >= blockTime && blockTime != 0) {
        blockTime = 0; // Reset block time
        lcd.clear();
        lcd.print("Digite a senha:");
    }

    // 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) {
        handleKeyPress(key);
    }
}

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];
                // Aguardar para evitar acionamentos
sequenciais
                while (digitalRead(rows[j]) == LOW);
                delay(50); // Debounce delay
            }
        }

        // Colocar o pino de saída (coluna) em nível lógico
alto
        digitalWrite(cols[i], HIGH);
    }
    // retorno da função
    return ret;
}

```

```

void handleKeyPress(char key) {
    if (loggedIn) {
        switch (key) {
            case 'A':
                addPassword();
                break;
            case 'B':
                removePassword();
                break;
            case 'C':
                showPasswords();
                break;
            case 'D':
                logOff();

```

```

        break;
    default:
        handleInput(key);
        break;
    }
} else {
    handleInput(key);
}
}

void handleInput(char key) {
    if (key == '*' && input.length() > 0) {
        input = input.substring(0, input.length() - 1);
        lcd.setCursor(input.length(), 1);
        lcd.print(" ");
        lcd.setCursor(input.length(), 1);
    } else if (key == '#') {
        checkPassword();
    } else if (key >= '0' && key <= '9' && input.length() < 4)
    {
        input += key;
        lcd.setCursor(input.length() - 1, 1);
        lcd.print(key);
    }
}

void checkPassword() {
    if (input.length() == 4 || (input.length() == 0 &&
passwordCount == 0)) {
        lcd.clear();
        bool accessGranted = false;
        if (passwordCount == 0) {
            accessGranted = true;
        } else {
            for (int i = 0; i < passwordCount; i++) {
                if (input == passwords[i]) {
                    accessGranted = true;

```

```

        break;
    }
}

if (accessGranted) {
    lcd.print("Acesso Permitido");
    loggedIn = true;
} else {
    lcd.print("Acesso Negado");
    attempts++;
    if (attempts >= maxAttempts) {
        blockTime = millis() + blockDuration;
        attempts = 0;
    }
}

delay(2000); // Delay para mostrar a mensagem antes de
resetar
lcd.clear();
if (!loggedIn) {
    lcd.print("Digite a senha:");
} else {
    lcd.print("Menu:");
}
input = ""; // Resetar o input
}

}

void addPassword() {
    lcd.clear();
    lcd.print("Nova Senha:");
    input = "";
    while (true) {
        char key = keyboardScan();
        if (key == 'A' || key == 'B' || key == 'C' || key ==
'D') {
            handleKeyPress(key); // Navegar mesmo dentro do
menu

```

```

        return;
    }
    if (key == '*') {
        if (input.length() > 0) {
            input = input.substring(0, input.length() -
1);

            lcd.setCursor(input.length(), 1);
            lcd.print(" ");
            lcd.setCursor(input.length(), 1);
        }
    } else if (key == '#') {
        if (input.length() == 4) {
            if (passwordCount < 10) {
                passwords[passwordCount] = input;
                passwordCount++;
                lcd.clear();
                lcd.print("Senha Adicionada");
            } else {
                lcd.clear();
                lcd.print("Limite Atingido");
            }
        } else {
            lcd.clear();
            lcd.print("Senha Invalida");
        }
        delay(2000);
        lcd.clear();
        lcd.print("Menu:");
        return;
    } else if (key >= '0' && key <= '9' && input.length()
< 4) {
        input += key;
        lcd.setCursor(input.length() - 1, 1);
        lcd.print(key);
    }
}
}

```

```

void removePassword() {
    lcd.clear();
    lcd.print("Remover Senha:");
    input = "";
    while (true) {
        char key = keyboardScan();
        if (key == 'A' || key == 'B' || key == 'C' || key ==
'D') {
            handleKeyPress(key); // Navegar mesmo dentro do
menu

            return;
        }
        if (key == '*') {
            if (input.length() > 0) {
                input = input.substring(0, input.length() -
1);

                lcd.setCursor(input.length(), 1);
                lcd.print(" ");
                lcd.setCursor(input.length(), 1);
            }
        } else if (key == '#') {
            if (input.length() == 4) {
                for (int i = 0; i < passwordCount; i++) {
                    if (passwords[i] == input) {
                        if (i == 0 && !isAdminRemoved) {
                            lcd.clear();
                            lcd.print("Senha Admin nao pode
ser removida");

                            delay(2000);
                            lcd.clear();
                            lcd.print("Remover Senha:");
                            return;
                        } else {
                            for (int j = i; j < passwordCount
- 1; j++) {

```



```

        passwords[j] = passwords[j +
1];

        }
        passwords[passwordCount - 1] = "";
        passwordCount--;
        lcd.clear();
        lcd.print("Senha Removida");
        delay(2000);
        lcd.clear();
        lcd.print("Menu:");
        if (i == 0) {
            isAdminRemoved = true;
        }
        return;
    }
}

}
lcd.clear();
lcd.print("Senha Nao Encontrada");
} else {
    lcd.clear();
    lcd.print("Senha Invalida");
}
delay(2000);
lcd.clear();
lcd.print("Menu:");
return;
} else if (key >= '0' && key <= '9' && input.length()
< 4) {
    input += key;
    lcd.setCursor(input.length() - 1, 1);
    lcd.print(key);
}
}
}

void showPasswords() {

```

```
    lcd.clear();
    lcd.print("Senhas:");
    for (int i = 0; i < passwordCount; i++) {
        lcd.setCursor(0, 1);
        if (i == 0 && !isAdminRemoved) {
            lcd.print(passwords[i] + " - Admin");
        } else {
            lcd.print(passwords[i]);
        }
        delay(2000);
        lcd.clear();
        lcd.print("Senhas:");
    }
}

void logOff() {
    loggedIn = false;
    input = ""; // Resetar o input ao deslogar
    lcd.clear();
    lcd.print("Deslogado");
    delay(2000);
    lcd.clear();
    lcd.print("Digite a senha:");
}
```