

```
1: /*****
2: **                                     Exercício 4                                     **
3: **                                     **                                     **
4: **      Exercício 4: Jogo das senhas                                     **
5: **      Ao digitar uma senha de 4 dígitos no teclado do kit Pic       **
6: **      Genius, o programa verifica se a senha digitada é a senha     **
7: **      correta, se sim, pisca os leds 3 vezes, e da uma mensgaem no  **
8: **      display lcd, se não, emite um sinal sonoro no buzzer e uma     **
9: **      mensgaem de mensagem incorreta no display lcd.                 **
10: **                                     **                                     **
11: **      Arquivo: exercicio4.c                                           **
12: **      Compilador: MikroC PRO PIC v.6.4.0                             **
13: **      UFLA - Lavras/MG - 14/11/2024                                  **
14: *****/
15:
16: // Conexoes LCD do kit PICGenios
17: sbit LCD_RS at RE2_bit;
18: sbit LCD_EN at RE1_bit;
19: sbit LCD_D4 at RD4_bit;
20: sbit LCD_D5 at RD5_bit;
21: sbit LCD_D6 at RD6_bit;
22: sbit LCD_D7 at RD7_bit;
23:
24: sbit LCD_RS_Direction at TRISE2_bit;
25: sbit LCD_EN_Direction at TRISE1_bit;
26: sbit LCD_D4_Direction at TRISD4_bit;
27: sbit LCD_D5_Direction at TRISD5_bit;
28: sbit LCD_D6_Direction at TRISD6_bit;
29: sbit LCD_D7_Direction at TRISD7_bit;
30:
31: #define atraso 50
32:
33: char senhaDigitada[5]; // Array para armazenar senha digitada (4 dígitos + '\0' pa
    ara finalizar a string)
34:
35: // Função para mostrar texto no LCD
36: void LCD(char texto[16]) {
37:     Lcd_Out(1,1,texto);
38: }
39:
40: // Função para ler tecla digitada e retornar 1 se uma tecla foi pressionada, ou
    se não
41: char teclaDigitada(int j) {
42:     char saida;
43:     char str[2]; // Array para converter `saida` em string
44:
45:     portb.rb0 = 0;           // habilita primeira coluna do teclado
46:     delay_ms(atraso);
47:     if (portd.rd1 == 0)      saida = '7';
48:     else if (portd.rd2 == 0) saida = '4';
49:     else if (portd.rd3 == 0) saida = '1';
50:     else saida = '\0';
51:     portb.rb0 = 1;           // desabilita primeira coluna do teclado
52:     delay_ms(atraso);
53:
54:     portb.rb1 = 0;           // habilita segunda coluna do teclado
55:     delay_ms(atraso);
56:     if (saida == '\0' && portd.rd0 == 0)      saida = '0';
57:     else if (saida == '\0' && portd.rd1 == 0)  saida = '8';
58:     else if (saida == '\0' && portd.rd2 == 0)  saida = '5';
59:     else if (saida == '\0' && portd.rd3 == 0)  saida = '2';
60:     portb.rb1 = 1;           // desabilita segunda coluna do teclado
```

```
61:    delay_ms(atraso);
62:
63:    portb.rb2 = 0;           // habilita terceira coluna do teclado
64:    delay_ms(atraso);
65:    if (saida == '\0' && portd.rd1 == 0)      saida = '9';
66:    else if (saida == '\0' && portd.rd2 == 0)  saida = '6';
67:    else if (saida == '\0' && portd.rd3 == 0)  saida = '3';
68:    portb.rb2 = 1;           // desabilita terceira coluna do teclado
69:    delay_ms(atraso);
70:
71:    if (saida != '\0') {
72:        str[0] = saida; // Converte caractere para string
73:        str[1] = '\0'; // Finaliza string
74:
75:        Lcd_Out(2, j + 1, str); // Exibe o dígito no LCD
76:        senhaDigitada[j] = saida; // Armazena o dígito na senha digitada
77:        return 1; // Retorna 1 para indicar que uma tecla foi pressionada
78:    }
79:
80:    return 0; // Retorna 0 se nenhuma tecla foi pressionada
81: }
82:
83: //Função para piscar os leds
84: void ascendeLed(){
85:     portb = 240;
86:     delay_ms(250);
87:     portb = 0;
88:     delay_ms(250);
89: }
90:
91: void main() {
92:     char senhaCorreta[5] = "9999"; // Senha correta para comparação
93:     int i, j; //variáveis para controle de loop
94:
95:     ADCON0 = 0x00; // Desabilita o conversor A/D
96:     ADCON1 = 0x06; // Configura todos pinos da porta A para digital
97:     // trisc e' uma posicao (registrador) na memoria de dados
98:     // do microcontrolador. Configura o pino RC1 como saida para
99:     // comunicacao com o buzzer.
100:    trisc = 0xFD; // 0b11111101
101:
102:    // Inicia com buzzer desligado.
103:    portc.rc1 = 0;
104:
105:    trisd = 0x0F; // Configura pinos RD0,RD1,RD2,RD3 da PORTD como entradas (t
teclado) e RD4,RD5,RD6,RD7 como saída (LCD)
106:    trisb = 0; // Configura pinos RB0,RB1,RB2 da PORTB como saída (teclado) e
o restante como entrada
107:    portb = 7; // inicializa pinos RB0,RB1,RB2 com o valor 1 (para o teclado)
e o restante como 0 (LEDs).
108:
109:    Lcd_Init(); // Inicializa o Display
110:    delay_ms(atraso);
111:    Lcd_Cmd(_LCD_CLEAR); // Limpa o Display
112:    delay_ms(atraso);
113:    Lcd_Cmd(_LCD_CURSOR_OFF); // Desabilita o cursor
114:    delay_ms(atraso);
115:
116:    LCD("Digite a senha:");
117:
118:    j = 0; // Índice para exibir números na posição correta
119:    do {
```

```
120:     if (teclaDigitada(j)) { // Chama teclaDigitada e só incrementa se uma tecla
    foi pressionada
121:         j++;
122:         if (j >= 4) { // Após digitar 4 dígitos, verifica a senha
123:             senhaDigitada[4] = '\0'; // Finaliza a string senhaDigitada
124:
125:             // Compara a senha digitada com a senha correta
126:             if (strcmp(senhaDigitada, senhaCorreta) == 0) { //se for correta
127:                 delay_ms(200);
128:                 Lcd_Cmd(_LCD_CLEAR);
129:                 LCD("Senha correta!");
130:                 ascendeLed();
131:                 ascendeLed();
132:                 ascendeLed();
133:             } else { //se for incorreta
134:                 delay_ms(200);
135:                 Lcd_Cmd(_LCD_CLEAR);
136:                 LCD("Senha incorreta!");
137:                 portc.rc1 = 1;
138:                 delay_ms(500);
139:                 portc.rc1 = 0;
140:             }
141:
142:             delay_ms(500); // Espera meio segundo para mostrar a mensagem
143:             Lcd_Cmd(_LCD_CLEAR); // Limpa o display
144:
145:             LCD("Digite a senha:"); // Reinicia a solicitação de senha
146:             j = 0; // Reinicia o índice para a próxima entrada de senha
147:         }
148:     }
149:     delay_ms(atraso);
150: } while(1);
151: } // fim do programa
152:
```