

FATEC IPIRANGA

PASTOR ENÉAS TOGNINI

**ANÁLISE E DESENVOLVIMENTO DE
SISTEMAS**

PROGRAMAÇÃO ESTRUTURADA E MODULAR

PROFESSOR CARLOS HENRIQUE VERISSIMO PEREIRA

MILENA MITIE AOKI

**SÃO PAULO, SP
2024**

Fatec
Ipiranga

CP
Centro
Paula Souza

SUMÁRIO

ATIVIDADE N1-5 - REFATORANDO PARA MODULARIZAR CÓDIGO

1	PROGRAMAÇÃO EM LINGUAGEM C.....	3
2	DIAGRAMA DE BLOCOS DA SOLUÇÃO.....	9

PROGRAMAÇÃO EM LINGUAGEM C

```
#include <stdio.h>

#include <string.h>

// Estrutura que representa uma peça no tabuleiro
typedef struct {
    char identificador[4]; // Ex: "BR", "BP1", "PR", etc.
} Peca;

// Função para inicializar o tabuleiro com peças na posição inicial
void inicializarTabuleiro(Peca tabuleiro[8][8]) {
    // Preencher o tabuleiro com "X" para casas vazias
    for (int i = 0; i < 8; i++) {
        for (int j = 0; j < 8; j++) {
            strcpy(tabuleiro[i][j].identificador, "X");
        }
    }

    // Configuração das peças brancas
    strcpy(tabuleiro[0][0].identificador, "BT1"); strcpy(tabuleiro[0][1].identificador,
"BC1");
    strcpy(tabuleiro[0][2].identificador, "BB1"); strcpy(tabuleiro[0][3].identificador,
"BD");
    strcpy(tabuleiro[0][4].identificador, "BR"); strcpy(tabuleiro[0][5].identificador,
"BB2");
    strcpy(tabuleiro[0][6].identificador, "BC2"); strcpy(tabuleiro[0][7].identificador,
"BT2");
    for (int i = 0; i < 8; i++) {
```

```

        sprintf(tabuleiro[1][i].identificador, "BP%d", i + 1);
    }

    // Configuração das peças pretas
    strcpy(tabuleiro[7][0].identificador, "PT1"); strcpy(tabuleiro[7][1].identificador,
"PC1");
    strcpy(tabuleiro[7][2].identificador, "PB1"); strcpy(tabuleiro[7][3].identificador,
"PD");
    strcpy(tabuleiro[7][4].identificador, "PR"); strcpy(tabuleiro[7][5].identificador,
"PB2");
    strcpy(tabuleiro[7][6].identificador, "PC2"); strcpy(tabuleiro[7][7].identificador,
"PT2");
    for (int i = 0; i < 8; i++) {
        sprintf(tabuleiro[6][i].identificador, "PP%d", i + 1);
    }
}

// Função para imprimir o tabuleiro
void imprimirTabuleiro(Peca tabuleiro[8][8]) {
    char letrasColunas[] = "abcdefgh";

    // Imprime as letras das colunas
    printf(" ");
    for (int i = 0; i < 8; i++) {
        printf("%4c", letrasColunas[i]);
    }
    printf("\n");

    // Imprime o tabuleiro linha por linha

```

```

for (int i = 0; i < 8; i++) {
    printf("%d ", 8 - i); // Número da linha
    for (int j = 0; j < 8; j++) {
        printf("%4s", tabuleiro[i][j].identificador);
    }
    printf("\n");
}
printf("\n");
}

```

// Função para mover uma peça de uma posição para outra

```

void moverPeca(Peca tabuleiro[8][8], int origemLinha, int origemColuna, int
destinoLinha, int destinoColuna) {
    strcpy(tabuleiro[destinoLinha][destinoColuna].identificador,
tabuleiro[origemLinha][origemColuna].identificador);
    strcpy(tabuleiro[origemLinha][origemColuna].identificador, "...");
}

```

// Funções que representam as jogadas específicas do Xequê Pastor

// Jogada #1: Brancas jogam Peão do Rei (e2 -> e4)

```

void jogada1Brancas(Peca tabuleiro[8][8]) {
    printf("Jogada #1: Brancas jogam Peão do Rei (e2 -> e4)\n");
    moverPeca(tabuleiro, 1, 4, 3, 4);
    imprimirTabuleiro(tabuleiro);
}

```

// Jogada #1: Pretas jogam Peão do Rei (e7 -> e5)

```

void jogada1Pretas(Peca tabuleiro[8][8]) {

```

```
printf("Jogada #1: Pretas jogam Peão do Rei (e7 -> e5)\n");  
moverPeca(tabuleiro, 6, 4, 4, 4);  
imprimirTabuleiro(tabuleiro);  
}
```

```
// Jogada #2: Brancas jogam Bispo (f1 -> c4)  
void jogada2Brancas(Peca tabuleiro[8][8]) {  
    printf("Jogada #2: Brancas jogam Bispo (f1 -> c4)\n");  
    moverPeca(tabuleiro, 0, 5, 3, 2);  
    imprimirTabuleiro(tabuleiro);  
}
```

```
// Jogada #2: Pretas jogam Cavalo (b8 -> c6)  
void jogada2Pretas(Peca tabuleiro[8][8]) {  
    printf("Jogada #2: Pretas jogam Cavalo (b8 -> c6)\n");  
    moverPeca(tabuleiro, 7, 1, 5, 2);  
    imprimirTabuleiro(tabuleiro);  
}
```

```
// Jogada #3: Brancas jogam Dama (d1 -> h5)  
void jogada3Brancas(Peca tabuleiro[8][8]) {  
    printf("Jogada #3: Brancas jogam Dama (d1 -> h5)\n");  
    moverPeca(tabuleiro, 0, 3, 4, 7);  
    imprimirTabuleiro(tabuleiro);  
}
```

```
// Jogada #3: Pretas jogam Cavalo (g8 -> f6)  
void jogada3Pretas(Peca tabuleiro[8][8]) {
```

```

printf("Jogada #3: Pretas jogam Cavalo (g8 -> f6)\n");
moverPeca(tabuleiro, 7, 6, 5, 5);
imprimirTabuleiro(tabuleiro);
}

// Jogada #4: Xequê Mate (Dxf7#)
void jogada4XequêMate(Peca tabuleiro[8][8]) {
    printf("Jogada #4: Xequê Mate - Brancas capturam Peão do Rei (Dxf7#)\n");
    moverPeca(tabuleiro, 4, 7, 6, 5);
    imprimirTabuleiro(tabuleiro);
}

// Função principal que coordena as jogadas
int main() {
    Peca tabuleiro[8][8];

    // Inicializa o tabuleiro
    inicializarTabuleiro(tabuleiro);

    printf("Tabuleiro inicial:\n");
    imprimirTabuleiro(tabuleiro);

    // Simulação do Xequê Pastor
    jogada1Brancas(tabuleiro);
    jogada1Pretas(tabuleiro);
    jogada2Brancas(tabuleiro);
    jogada2Pretas(tabuleiro);
    jogada3Brancas(tabuleiro);

```

```
jogada3Pretas(tabuleiro);  
jogada4XequeMate(tabuleiro);  
  
printf("Fim do jogo: Xeque Mate!\n");  
  
return 0;  
}
```


DIAGRAMA DE BLOCOS DA SOLUÇÃO

