

Relatório Projeto Técnicas e Desenvolvimento de Algoritmos  
Jogo da Velha (3x3)

Grupo: Felipe Souto Maior Mendes – RGM: 32565895

Matteus Maia Resende Guimarães – RGM: 34128361

Rafaella Teotônio Veiga de Araújo – RGM: 32729308

Julliane Di Paula Oliveira Xavier – RGM: 32920270

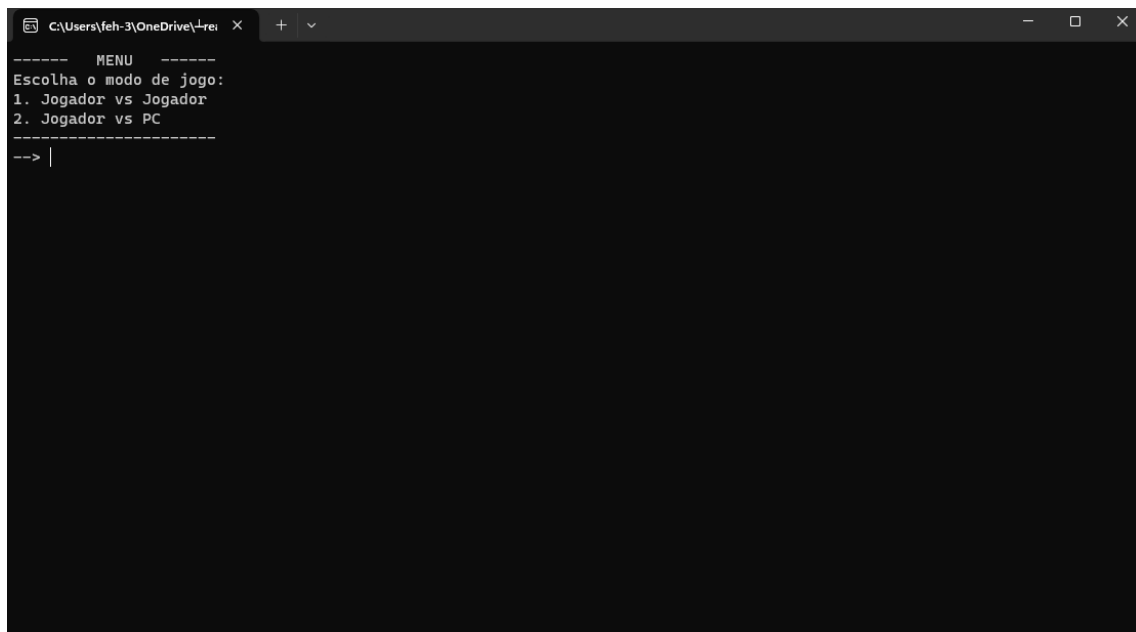
## INTRODUÇÃO

O algoritmo é baseado nas regras do jogo da velha, onde cada jogador deve preencher um espaço do tabuleiro com as letras ‘X’ ou ‘O’ de maneira alternada. O primeiro jogador a fazer uma sequência de 3 letras iguais na horizontal, vertical ou nas diagonais será o vencedor, ou o tabuleiro pode ser preenchido totalmente, de maneira que nenhum dos jogadores conseguirá fazer tal sequência, resultando em um empate. Além da forma tradicional de jogo, o algoritmo também permite que o usuário escolha jogar contra a máquina através de um simples menu de opções.

## RESULTADOS

A primeira dificuldade que encontramos foi na forma de alocar dinamicamente os espaços da matriz do tabuleiro 3x3. Depois de pesquisar, decidimos fazer do tabuleiro um ponteiro para uma matriz de caracteres, que aloca o seu espaço para a matriz e depois a inicializa com o valor ‘ ‘. Dessa forma, a lógica de alocação dinâmica dos espaços em branco na criação da matriz e na checagem dos vencedores se tornou muito parecida com a alocação estática, o que facilitou a implementação do algoritmo.

Outra particularidade foi na apresentação do jogo. Não queríamos que o terminal ficasse muito poluído ao terminar ao iniciar/terminar uma partida. Descobrimos que o Windows permite a utilização de uma função chamada *system(“cls”)*. Quando solicitada no código, ela limpa a janela do console, o que foi muito útil no decorrer da implementação.



*Figura 1: Menu Interativo Inicial*

```
C:\Users\feh-3\OneDrive\rei X + v - □ ×

  0 1 2
0  | | |
  -----
1  | | |
  -----
2  | | |

Jogador 1 (X) --> Informe a linha e a coluna: 0 0
  0 1 2
0 x | | |
  -----
1  | | |
  -----
2  | | |

Jogador 2 (O) --> Informe a linha e a coluna: 1 0
  0 1 2
0 x | | |
  -----
1  o | | |
  -----
2  | | |

Jogador 1 (X) --> Informe a linha e a coluna: |
```

Figura 2: Modo de Jogo (Jogador vs Jogador)

```
C:\Users\feh-3\OneDrive\rei X + v - □ ×

  0 1 2
0  | | |
  -----
1  | | |
  -----
2  | | |

Jogador 1 (X) --> Informe a linha e a coluna: 2 1
  0 1 2
0  | | |
  -----
1  | | |
  -----
2  | x |

PC:
  0 1 2
0  | | |
  -----
1  | | |
  -----
2  | x | o

Jogador 1 (X) --> Informe a linha e a coluna: |
```

Figura 3: Modo de Jogo (Jogador vs PC)

```
C:\Users\feh-3\OneDrive\rei x + v
0 1 2
0 x | |
-----
1 x | o | o
-----
2 x | x | o

Jogador 1 venceu!

Deseja jogar novamente? 1 = Sim / 0 = Não
--> |
```

*Figura 4: Tela final do jogo*

## APÊNDICE

```
#include <stdio.h>
#include <stdlib.h>
#include <locale.h>

char **criarTabuleiro(){
    int i,j;
    char **tabuleiro = (char **)malloc(3 * sizeof(char *));
    for (i = 0; i < 3; i++){
        tabuleiro[i] = (char *)malloc(3 * sizeof(char));
        for (j = 0; j < 3; j++){
            tabuleiro[i][j] = ' ';
        }
    }
    return tabuleiro;
}

void mostrarTabuleiro(char **tabuleiro){
    int i,j;
    printf("\n    0  1  2\n\n");
    for (i = 0; i < 3; i++){
        printf("%d  ", i);
        for (j = 0; j < 3; j++){
            printf(" %c ", tabuleiro[i][j]);
            if (j < 2){
                printf("|");
            }
        }
        printf("\n");
        if (i < 2) {
            printf(" ----- \n");
        }
    }
    printf("\n");
}
```

Figura 5: Funções para Criar e Inicializar o tabuleiro.

```

int checarVencedor(char **tabuleiro, char letra){
    int i;
    for (i = 0; i < 3; i++){
        if ((tabuleiro[i][0] == letra && tabuleiro[i][1] == letra && tabuleiro[i][2] == letra) ||
            (tabuleiro[0][i] == letra && tabuleiro[1][i] == letra && tabuleiro[2][i] == letra)){
            return 1;
        }
    }
    if ((tabuleiro[0][0] == letra && tabuleiro[1][1] == letra && tabuleiro[2][2] == letra) ||
        (tabuleiro[0][2] == letra && tabuleiro[1][1] == letra && tabuleiro[2][0] == letra)){
        return 1;
    }
    return 0;
}

int checarFinalDoJogo(char **tabuleiro){
    int i,j;
    for (i = 0; i < 3; i++){
        for (j = 0; j < 3; j++){
            if (tabuleiro[i][j] == ' '){
                return 0;
            }
        }
    }
    return 1;
}

int checarJogada(char **tabuleiro, int linha, int coluna){
    if (linha >= 0 && linha < 3 && coluna >= 0 && coluna < 3 && tabuleiro[linha][coluna] == ' '){
        return 1;
    }
    return 0;
}

```

Figura 6: Funções para checar vencedor, final do jogo e jogada válida.

```

void jogadorJoga(char **tabuleiro, char letra){
    int linha, coluna;
    do {
        printf("--> Informe a linha e a coluna: ");
        scanf("%d %d", &linha, &coluna);

        if (checarJogada(tabuleiro, linha, coluna) == 0){
            printf("Linha e Coluna inválidas\n\n");
        }

    } while (checarJogada(tabuleiro, linha, coluna) == 0);

    tabuleiro[linha][coluna] = letra;
}

void pcJoga(char **tabuleiro, char letra){
    int linha, coluna;
    do {
        linha = rand() % 3;
        coluna = rand() % 3;

    } while (checarJogada(tabuleiro, linha, coluna) == 0);

    tabuleiro[linha][coluna] = letra;
}

```

Figura 7: Funções para ler a jogada do jogador e fazer jogada da máquina.

```

int main(){
    setlocale(LC_ALL, "Portuguese");

    int op = 0;
    char **tabuleiro;
    int jogadorAtual;
    int fim;
    int jogarNovamente;

    do{
        printf("-----\t MENU\t-----\n");
        printf("Escolha o modo de jogo:\n");
        printf("1. Jogador vs Jogador\n");
        printf("2. Jogador vs PC\n");
        printf("-----\n");
        printf("--> ");
        scanf("%d", &op);

        switch (op){
            case 1:
                system("cls");
                break;
            case 2:
                system("cls");
                break;
            default:
                printf("\nModo inválido. Por favor, escolha novamente.\n\n");
                continue;
        }

        tabuleiro = criarTabuleiro();
        jogadorAtual = 0;
        fim = 0;

        do{
            mostrarTabuleiro(tabuleiro);

            if (op == 1 || (op == 2 && jogadorAtual == 0)){
                printf("Jogador %d (%c)", jogadorAtual + 1, (jogadorAtual == 0) ? 'X' : 'O');
                jogadorJoga(tabuleiro, (jogadorAtual == 0) ? 'X' : 'O');
            } else {
                printf("PC:\n");
                pcJoga(tabuleiro, 'O');
            }

            if (checarVencedor(tabuleiro, 'X')){
                system("cls");
                mostrarTabuleiro(tabuleiro);
                printf("Jogador 1 venceu!\n\n");
                fim = 1;
            }

            } else if (checarVencedor(tabuleiro, 'O')){
                system("cls");
                mostrarTabuleiro(tabuleiro);
                if (op == 1 || (op == 2 && jogadorAtual == 0)){
                    printf("Jogador 2 venceu!\n\n");
                } else {
                    printf("PC venceu!\n\n");
                }
                fim = 1;
            }

            } else if (checarFinalDoJogo(tabuleiro)){
                system("cls");
                mostrarTabuleiro(tabuleiro);
                printf("Deu Velha!\n");
                fim = 1;
            }

        }

        jogadorAtual = (jogadorAtual + 1) % 2;

    } while (!fim);

    free(tabuleiro);

    printf("Deseja jogar novamente? 1 = Sim / 0 = Não\n--> ");
    scanf("%d", &jogarNovamente);
    if (jogarNovamente != 1){
        return 0;
    }
    else{
        op = 0;
    }
    system("cls");

} while (op != 1 && op != 2);

return 0;
}

```

*Figuras 8 e 9: Função principal do jogo.*