

UNIVERSIDADE FEDERAL DO AMAZONAS – UFAM  
INSTITUTO DE CIÊNCIAS EXATAS E TECNOLOGIA – ICET  
BACHARELADO EM ENGENHARIA DE SOFTWARE

Trabalho Prático IPII

ITACOATIARA-AM  
2020

**WALESON AZEVEDO PESSOA DE MELO**

MATRÍCULA: 21954228

**João Victor Neves Machado da Silva**

MATRÍCULA: 21955314

Trabalho Prático IP II

Trabalho referente ao assunto de Introdução á Programação II, apresentado como requisito para obtenção de nota parcial do segundo período, do segundo semestre de 2020, solicitado pelo professor Alternei Brito, ministrante da disciplina de IP II, no Curso de Bacharelado em Engenharia de Software, do Instituto de Ciências Exatas e Tecnologia – ICET.

ITACOATIARA-AM  
2020

## Introdução

O jogo da velha é um jogo ou passatempo popular. Possui regras simples, sem grandes dificuldade para ser aprendido. O jogo da velha é constituído de três linhas e três colunas, onde dois jogadores colocam seus símbolos (Geralmente um circulo e um x) em espaços em branco entre essas linhas, com a finalidade de preencher ou as linhas horizontais, verticais ou diagonais com o mesmo símbolo antes que o outro jogador faça isso primeiro.

## Objetivos

O intuito deste trabalho é implementar este jogo na linguagem de programação C, usando as estruturas de dados de forma lógica para chegar ao resultado final, pondo em pratica todos os conceitos aprendidos na matéria de Introdução à Programação II (IPII).

## Ideias Gerais da Aplicação

Usar como uma tabela de jogo uma matriz, onde dois jogadores podem dar as coordenadas de onde será marcado o seu símbolo, as coordenadas serão o número da linha e da coluna respetivamente.

Exemplo: O jogador 1 terá seu número marcado na tabela e o jogador 2 também logo após informar a linha e coluna, como na imagem abaixo.

```
-----JOGO DA VELHA-----
1      0      0
0      2      0
0      0      0

----Jogador 1-----
Linha: █
```

Quando um jogador faz a sequencia de números ele é informado que ganhou o jogo. Caso dê empate (deu Velha) o programa mostra uma mensagem, e então pergunta se deseja prosseguir, se sim, o programa reinicia a tabela, se não, o programa mostra um relatório com as informações de quantas partidas foram jogadas, quantas vezes cada jogador ganhou e a porcentagem de vitoria de cada jogador.

## Detalhes da Implementação

### 1- Estruturas de Dados Utilizada

A estrutura de dados utilizada foi matriz, para formar a tabela do jogo. O Jogador informa a linha e a coluna onde quer marcar, começando do um até o três.

## 2- Funções

```
void pr_tabuleiro(int tabuleiro[3][3]);
```

Função para inicializar o tabuleiro, passamos por parâmetro o tabuleiro que será inicializado com o valor zero em todas as posições, ele serve também para reiniciar o tabuleiro a cada jogada.

Parâmetros:

tabuleiro[3][3]: Matriz que serve como o tabuleiro do jogo.

Retorno: Void.

```
void p_tabuleiro(int tabuleiro[3][3]);
```

Função para mostrar o tabuleiro, um laço de repetição é feito para passar por todas as posições do tabuleiro o mostrar o valor contido.

Parâmetros:

tabuleiro[3][3]: Matriz que serve como o tabuleiro do jogo.

Retorno: Void.

```
void informacoes(int jog_1, int jog_2, int jogadas);
```

Função para mostrar o relatório do jogo quando o mesmo é finalizado.

Parâmetros:

jog\_1: Número de vitórias do jogador 1;

jog\_2: Número de vitórias do jogador 2;

jogadas: Número total de partidas jogadas.

Retorno: Void.

```
int verificacao(int tabuleiro[3][3], int jogador);
```

Função para fazer as verificações de quem ganhou o jogo. Nela está implementada as regras de vitória do jogo.

Parâmetros:

tabuleiro[3][3]: Matriz que serve como o tabuleiro do jogo;

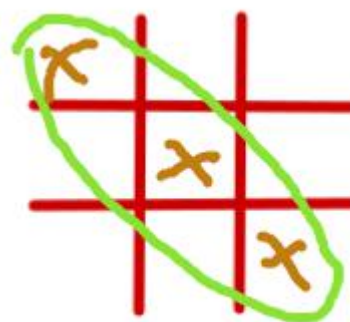
jogador: Número do jogador.

Retorno: Int. Se o jogador ganhar a função retorna o número do jogador, se não retorna zero.

Verificações feitas:

1- Diagonal Principal:

```
for(i = 0; i < 3; i++){  
    if(tabuleiro[i][i] == jogador){  
        cont += 1;
```



```

    }

    if(cont >= 3){
        vencedor_partida = jogador;
    }
}

```

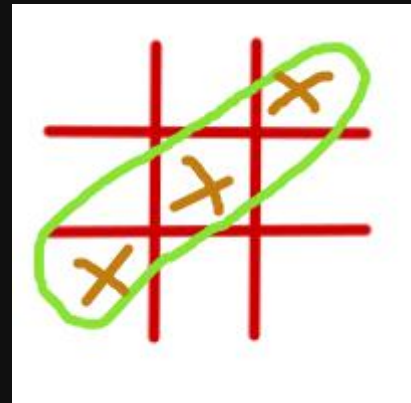
2- Diagonal Secundaria:

`j = 2;` // Definindo j como 2 para começar no canto superior direito de nossa matriz.

```

for(i = 0; i < 3; i++){
    if(tabuleiro[i][j] == jogador){
        cont += 1;
        j -= 1;
    }
    if(cont >= 3){
        vencedor_partida = jogador;
    }
}

```

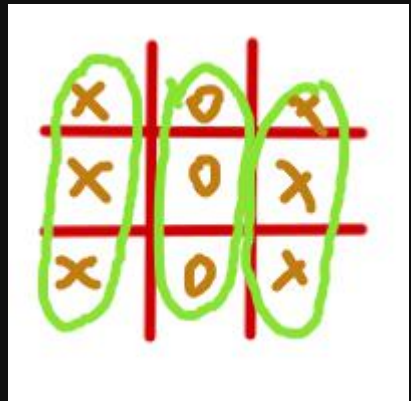


3- Vertical:

```

for(j = 0; j < 3; j++){
    cont = 0;
    for(i = 0; i < 3; i++){
        if(tabuleiro[i][j] == jogador){
            cont += 1;
        }
    }
    if(cont >= 3){
        vencedor_partida = jogador;
    }
}

```

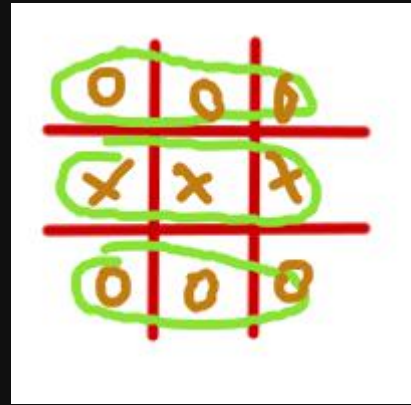


4- Horizontal:

```

for(i = 0; i < 3; i++){
    cont = 0;
    for(j = 0; j < 3; j++){
        if(tabuleiro[i][j] == jogador){
            cont += 1;
        }
    }
    if(cont >= 3){
        vencedor_partida = jogador;
    }
}

```



```

int inserir(int tabuleiro[3][3], int linha, int coluna, int jogador);

```

Função para inserir no tabuleiro. Depois de passar as coordenadas a função verifica se o local indicado é válido e se o local já está preenchido, se sim, uma mensagem é exibida avisando que o local é inválido para marcar, caso contrário o local é marcado.

Parâmetros:

tabuleiro[3][3]: Matriz que serve como o tabuleiro do jogo;

linha: Linha onde o jogador quer marcar;

coluna: Coluna onde o jogador quer marcar;

jogador: Número do jogador.

Retorno: Int. Retorna zero se o local for marcado com sucesso ou um se o local não for válido.

```

int jogar(int tabuleiro[3][3], int *partidas);

```

Função que executa a jogada. Ela pede as coordenadas e passa para a função ***inserir***, logo após inserir chama a função de ***verificacao***.

Parâmetros:

tabuleiro[3][3]: Matriz que serve como o tabuleiro do jogo;

\*partidas: Ponteiro que serve para atualizar o número de jogadas por partidas, para que quando der velha (empate) a partida acabe.

Retorno: Int. Se o jogador ganhar a função retorna o número do jogador, se as jogadas acabarem a função retorna três.

```
int main()
```

Função principal do programa. Contem o código que faz as chamadas das funções.

**\*O código contem explicações em forma de comentário.**