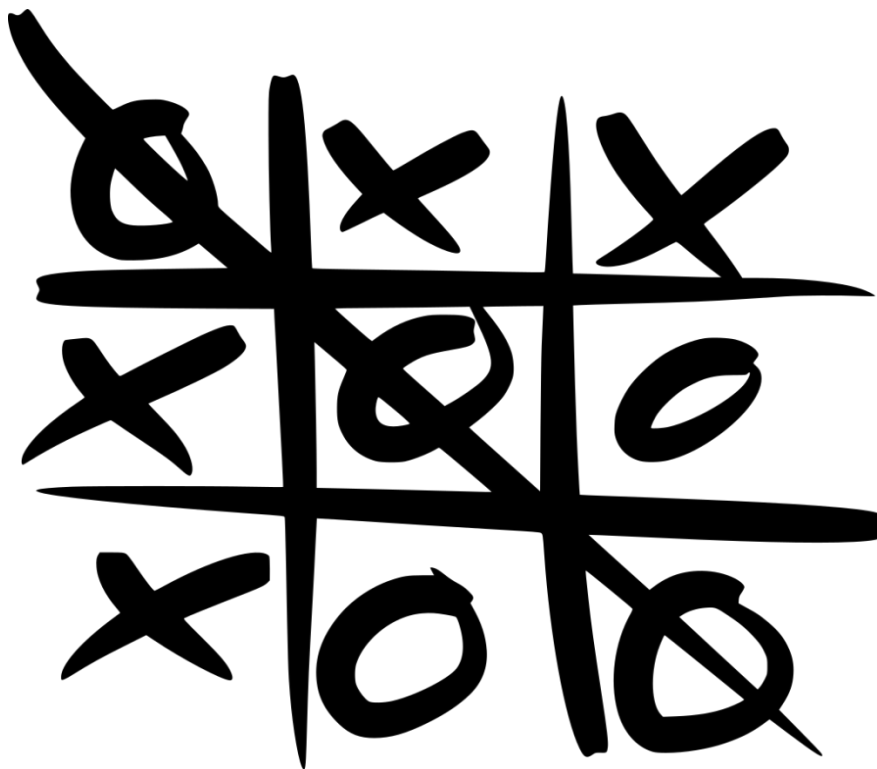


# Jogo da Velha

## Introdução

Nesta atividade, você terá a oportunidade de desenvolver a implementação do Jogo da Velha utilizando os conceitos de Tipos Abstratos de Dados (TADs) opacos, com alocação dinâmica de memória. Os TADs opacos garantem que a estrutura interna do tipo seja ocultada, promovendo uma maior encapsulação e modularidade no código.

O Jogo da Velha, também conhecido como "Tic-Tac-Toe" em inglês, é um jogo de tabuleiro tradicional projetado para dois jogadores. O objetivo principal é ser o primeiro a formar uma sequência de três símbolos idênticos em uma grade 3x3.



## Funcionamento do programa

O tabuleiro é composto por uma matriz 3 x 3, na qual cada célula é representada por coordenadas X e Y. No início, todas as posições estão preenchidas com o caractere "-", indicando que estão desocupadas. O jogador 1 é responsável pela primeira jogada, utilizando o símbolo "x", enquanto o jogador 2 utiliza o símbolo "0". As jogadas dos

jogadores são alternadas, com o jogador 2 aguardando a jogada do jogador 1. Após cada jogada, o programa exibirá o estado atualizado do tabuleiro.

```
Jogador 1
Digite uma posicao (x e y):
Jogada [1,1]!
    ---
    -X-
    ---

Jogador 2
Digite uma posicao (x e y):
```

O programa deverá processar todas as jogadas até que o jogo não possa mais prosseguir. Nesse ponto, o programa identificará o jogador vencedor ou declarará um empate. O usuário terá a opção de iniciar uma nova partida digitando 's', ou encerrar o programa digitando 'n'.

```
Jogador 1
Digite uma posicao (x e y):
Jogada [0,2]!
    00X
    0X0
    XXX
JOGADOR 1 Venceu!
Jogar novamente? (s,n)
```

## Regras de Funcionamento

O programa terá de ser projetado para lidar com jogadas inválidas, como jogadas fora dos limites do tabuleiro, jogadas em posições já ocupadas ou jogadas com formato incorreto. Para ilustrar, exemplos de jogadas inválidas são:

1. Fora do tabuleiro: Como o tabuleiro é uma matriz 3 x 3, somente coordenadas 0, 1 ou 2 são válidas.

```
Jogador 1
Digite uma posicao (x e y):
Posicao invalida (FORA DO TABULEIRO - [3,3] )!
Jogador 1
Digite uma posicao (x e y):
```

2. Repetidas: Não é permitido seleccionar uma célula já ocupada.

```
Jogador 2
Digite uma posicao (x e y):
Jogada [2,1]!
    00X
    0X0
    -XX

Jogador 1
Digite uma posicao (x e y):
Posicao invalida (OCUPADA - [2,0] )!
Jogador 1
Digite uma posicao (x e y):
```

3. Formato incorreto: As coordenadas da jogada devem ser inseridas em um formato válido, consistindo de valores para a coordenada X e Y.

## Informações Adicionais

Esta tarefa é realizada individualmente, com o intuito de demonstrar sua compreensão dos conceitos e sua habilidade em criar trechos de código correspondentes.

O código deve seguir as interfaces definidas nos arquivos ".h" fornecidos junto com o exercício. Esses arquivos não devem ser modificados, pois contêm as especificações do problema a ser resolvido.

Além disso, é crucial enfatizar que você será responsável pela manipulação correta da memória durante a execução do programa. Isso inclui a alocação dinâmica e a liberação de memória conforme necessário, garantindo que não haja vazamentos de memória. Qualquer erro relacionado à manipulação de memória, detectado pelo Valgrind, resultará em um decréscimo na nota atribuída à atividade.

Definição dos formatos de entrada e saída:

- Entrada: O programa inicia lendo a primeira jogada do jogador 1 (coordenadas X e Y no tabuleiro). Após a validação dessa jogada, o programa lê a jogada do jogador 2 e assim sucessivamente até que o jogo seja concluído. Em seguida, ele aguarda a entrada do caractere 's' (para indicar uma nova partida) ou 'n' (para encerrar o programa).

- Saída: A cada jogada, o programa exibirá na saída padrão o estado atualizado do tabuleiro, incluindo as jogadas realizadas. Ao final de cada partida, o programa anunciará o vencedor, se houver um. Em caso de jogada inválida, a razão da invalidade será informada.

Exemplos detalhados de formatos de entrada e saída podem ser encontrados nos arquivos de exemplo fornecidos com o exercício.