

# EXERCÍCIO DE PROGRAMAÇÃO WEB: JOGO DA VELHA (TIC TAC TOE)

## Objetivo

Desenvolver um jogo da velha funcional utilizando **JavaScript puro**, juntamente com **HTML** e **CSS**, praticando conceitos de:

- Manipulação do DOM
- Eventos (Event Listeners)
- Estruturas condicionais
- Lógica de jogo
- Separação de responsabilidades (HTML, CSS e JS)

## Requisitos do projeto

Você deve implementar **um jogo da velha (Tic Tac Toe)** com as seguintes funcionalidades e regras:

## Regras e funcionalidades detalhadas

### 1. Interface do Jogo

- A interface deve conter uma **grade 3x3** de células (9 casas).
- Cada célula deve ser clicável.
- As células devem ser representadas visualmente (ex: por uma borda, cor, etc.).
- O jogo pode conter:
  - Um **título** ("Jogo da Velha" ou "Tic Tac Toe")
  - Um espaço para exibir **mensagens** (ex: "Vez do jogador X", "Jogador O venceu!", "Empate", etc.)

### 2. Jogabilidade

- Dois jogadores jogam alternadamente, usando "X" e "O".
- O primeiro jogador sempre joga com "**X**".
- Ao clicar em uma célula vazia:
  - O símbolo do jogador atual ("X" ou "O") deve ser inserido.

- A jogada deve ser registrada e a célula não pode mais ser clicada.
- Após cada jogada, o sistema deve:
  - Verificar se há um **vencedor**
  - Verificar se houve um **empate**
  - Alternar para o próximo jogador
  - Atualizar a mensagem de status no topo ou abaixo do tabuleiro
  - Um botão de "**Reiniciar Jogo**"

### 3. Condições de Vitória

O jogo deve detectar quando um dos jogadores vence. Um jogador vence se conseguir formar uma linha com três símbolos iguais (X ou O):

- **Horizontal:** qualquer linha com 3 símbolos iguais
- **Vertical:** qualquer coluna com 3 símbolos iguais
- **Diagonal:** duas diagonais possíveis com 3 símbolos iguais

### 4. Empate

- Se todas as 9 casas forem preenchidas **sem vencedor**, o jogo deve declarar **empate**.

### 5. Reiniciar o Jogo

- Ao clicar no botão "Reiniciar Jogo", o tabuleiro deve ser limpo:
  - Todas as casas ficam vazias
  - O jogo volta para o estado inicial
  - O primeiro jogador volta a ser o "X"
  - A mensagem de status é redefinida

## Tecnologias e restrições

- **HTML + CSS + JavaScript Puro (Vanilla JS)** somente.
- **Não é permitido** o uso de bibliotecas externas como jQuery, React, Bootstrap, etc.
- O código deve estar separado em arquivos (ex. index.html, style.css, script.js)

## Sugestões de estilo (Opcional)

- Destaque visual para o jogador atual (ex: cor diferente na mensagem)
- Destaque para células vencedoras (cor de fundo diferente, por exemplo)
- Fonte amigável e layout centralizado
- Animações simples (opcional)

## Prazo de entrega

- Data limite: **12 de setembro de 2025**
- Entregar o projeto via repositório Git obrigatoriamente.

## CrITÉRIOS de avaliação

Critério	Pontos
Funcionalidade completa do jogo	40 pts
Verificação de vitória e empate	20 pts
Código organizado e bem estruturado	15 pts
Separação de HTML, CSS e JS	10 pts
Responsividade ou estilo visual extra	5 pts
Comentários e boas práticas	10 pts
Total	100 pts

## Dicas

- Comece pelo HTML, criando a estrutura do tabuleiro.
- Use `addEventListener` para capturar cliques nas células.
- Crie uma função que verifica o estado do jogo após cada jogada.
- Mantenha o código modular, com funções pequenas e específicas.
- Procurar para SOLID