

Sumário

- 1. Tutorial: Jogo da Velha (Tic-Tac-Toe) com React
 - 1.1. Objetivo
 - 1.2. Estrutura do Projeto
 - 1.3. Exemplo de Código Completo
 - 1.4. Conceitos Principais
 - 1.5. Resultado Esperado
 - 1.6. Referência e Atribuição
-

1. Tutorial: Jogo da Velha (Tic-Tac-Toe) com React

1.1. Objetivo

Construir o clássico **Jogo da Velha** com React, aprendendo:

- Criação e composição de **componentes funcionais**
 - Uso de **estado (useState)**
 - Passagem de **props**
 - **Levantamento de estado** (lifting state up)
 - Gerenciamento de **histórico de jogadas** e “time travel”
-

1.2. Estrutura do Projeto

Componente	Responsabilidade
Square	Representa um quadrado do tabuleiro
Board	Controla o estado dos quadrados e lógica do jogo
Game	Gerencia histórico e renderiza o tabuleiro

1.3. Exemplo de Código Completo

```
import React from 'react';
import ReactDOM from 'react-dom/client';

function Square({ value, onClick }) {
  return (
    <button className="square" onClick={onClick}>
      {value}
    </button>
  );
}

function Board({ squares, onPlay, xIsNext }) {
  /
```

```
const nextValue = xIsNext ? "X" : "0";
const winner = calculateWinner(squares);
const status = winner
? `Vencedor: ${winner}`
: `Próximo jogador: ${nextValue}`;

function handleClick(i) {
  if (squares[i] || winner) return;
  const nextSquares = squares.slice();
  nextSquares[i] = nextValue;
  onPlay(nextSquares);
}

return (
  <>
  <div className="status">{status}</div>
  <div className="board-row">
    {[0, 1, 2].map(i => (
      <Square key={i} value={squares[i]} onClick={() => handleClick(i)} />
    )))
  </div>
  <div className="board-row">
    {[3, 4, 5].map(i => (
      <Square key={i} value={squares[i]} onClick={() => handleClick(i)} />
    )))
  </div>
  <div className="board-row">
    {[6, 7, 8].map(i => (
      <Square key={i} value={squares[i]} onClick={() => handleClick(i)} />
    )))
  </div>
  </>
);
}

function Game() {
  const [history, setHistory] = React.useState([Array(9).fill(null)]);
  const [currentMove, setCurrentMove] = React.useState(0);
  const xIsNext = currentMove % 2 === 0;
  const currentSquares = history[currentMove];

  function handlePlay(nextSquares) {
    const nextHistory = [...history.slice(0, currentMove + 1), nextSquares];
    setHistory(nextHistory);
    setCurrentMove(nextHistory.length - 1);
  }

  function jumpTo(move) {
    setCurrentMove(move);
  }

  const moves = history.map(_ , move) => {
    const description = move ? `Ir para jogada #${move}` : "Início do jogo";
    return (

```

```
<li key={move}>
    <button onClick={() => jumpTo(move)}>{description}</button>
</li>
);
});

return (
    <div className="game">
        <div className="game-board">
            <Board
                squares={currentSquares}
                onPlay={handlePlay}
                xIsNext={xIsNext}
            />
        </div>
        <div className="game-info">
            <ol>{moves}</ol>
        </div>
    </div>
);
}

function calculateWinner(squares) {
    const lines = [
        [0,1,2],[3,4,5],[6,7,8],
        [0,3,6],[1,4,7],[2,5,8],
        [0,4,8],[2,4,6]
    ];
    for (let [a,b,c] of lines) {
        if (squares[a] && squares[a] === squares[b] && squares[a] === squares[c]) {
            return squares[a];
        }
    }
    return null;
}

const root = ReactDOM.createRoot(document.getElementById("root"));
root.render(<Game />);
```

1.4. Conceitos Principais

- **Componentização** e hierarquia de componentes
- **Props** e callbacks de eventos (`onClick`)
- **Gerenciamento de estado** com `useState`
- **Imutabilidade** de arrays (`slice()`)
- **Lifting state up** para controle centralizado
- **Renderização condicional** e `map()` para listas
- **Histórico e time travel** no React

1.5. Resultado Esperado

- Um tabuleiro 3x3 exibido no navegador.
 - Alternância entre os jogadores “X” e “O”.
 - Indicação do próximo jogador ou vencedor.
 - Lista de jogadas anteriores clicáveis.
-

1.6. Referência e Atribuição

Este conteúdo foi adaptado com fins educacionais a partir do tutorial oficial: **“Tutorial: Tic-Tac-Toe” – React Docs (pt-BR)** Disponível em: <https://pt-br.react.dev/learn/tutorial-tic-tac-toe> © Meta Platforms / React Team. Utilização mediante autorização para fins de ensino.

Aqui terminamos.

Bons estudos!
Prof. João Ferreira