Sumário

---

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

1.1. Objetivo

Criar um jogo interativo de **Jogo da Velha** usando React, aplicando os principais conceitos:

- Componentes funcionais
- Props
- Estado (`useState`)
- “Lifting State Up” (compartilhamento de estado entre componentes)

1.2. Estrutura do Projeto

O projeto terá os seguintes componentes:

Componente	Função
Square	Representa um quadrado do tabuleiro
Board	Contém 9 quadrados e controla a lógica de jogadas
Game	Gerencia o histórico e a navegação entre jogadas

1.3. Exemplo de Código Completo

```
<!DOCTYPE html>
<html lang="pt-BR">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>React - Jogo da Velha</title>
```

```
    <script src="https://unpkg.com/react@17/umd/react.development.js"></script>
    <script src="https://unpkg.com/react-dom@17/umd/react-dom.development.js">
</script>
    <script src="https://unpkg.com/@babel/standalone/babel.min.js"></script>
    <style>
      .board-row {
        display: flex;
      }
      .square {
        width: 60px;
        height: 60px;
        background: #fff;
        border: 1px solid #999;
        font-size: 24px;
        font-weight: bold;
        text-align: center;
        line-height: 60px;
        cursor: pointer;
      }
      .status {
        margin-bottom: 10px;
        font-weight: bold;
      }
    </style>
  </head>
  <body>
    <div id="root"></div>

    <script type="text/babel">
      function Square({ value, onSquareClick }) {
        return (
          <button className="square" onClick={onSquareClick}>
            {value}
          </button>
        );
      }

      function Board({ xIsNext, squares, onPlay }) {
        function handleClick(i) {
          if (squares[i] || calculateWinner(squares)) return;

          const nextSquares = squares.slice();
          nextSquares[i] = xIsNext ? "X" : "O";
          onPlay(nextSquares);
        }

        const winner = calculateWinner(squares);
        let status;
        if (winner) {
          status = "Vencedor: " + winner;
        } else {
          status = "Próximo jogador: " + (xIsNext ? "X" : "O");
        }
```

```
      return (
        <>
          <div className="status">{status}</div>
          <div className="board-row">
            <Square value={squares[0]} onSquareClick={() => handleClick(0)} />
            <Square value={squares[1]} onSquareClick={() => handleClick(1)} />
            <Square value={squares[2]} onSquareClick={() => handleClick(2)} />
          </div>
          <div className="board-row">
            <Square value={squares[3]} onSquareClick={() => handleClick(3)} />
            <Square value={squares[4]} onSquareClick={() => handleClick(4)} />
            <Square value={squares[5]} onSquareClick={() => handleClick(5)} />
          </div>
          <div className="board-row">
            <Square value={squares[6]} onSquareClick={() => handleClick(6)} />
            <Square value={squares[7]} onSquareClick={() => handleClick(7)} />
            <Square value={squares[8]} onSquareClick={() => handleClick(8)} />
          </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((squares, move) => {
        let description;
        if (move > 0) {
          description = "Ir para jogada #" + move;
        } else {
          description = "Ir para o início do jogo";
        }
        return (
          <li key={move}>
            <button onClick={() => jumpTo(move)}>{description}</button>
          </li>
        );
      });

      return (
        <div className="game">
```

```
            <div className="game-board">
              <Board xIsNext={xIsNext} squares={currentSquares} onPlay=
{handlePlay} />
            </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 i = 0; i < lines.length; i++) {
          const [a, b, c] = lines[i];
          if (squares[a] && squares[a] === squares[b] && squares[a] ===
squares[c]) {
            return squares[a];
          }
        }
        return null;
      }

      ReactDOM.render(<Game />, document.getElementById("root"));
    </script>
  </body>
</html>
```

1.4. Explicação do Código

1.4.1. `Square`

Renderiza um botão que exibe "X", "O" ou vazio. Recebe as props `value` e `onSquareClick`.

1.4.2. `Board`

Organiza os 9 quadrados e contém a lógica de jogadas. Determina o vencedor com `calculateWinner` e exibe o status atual do jogo.

1.4.3. `Game`

Gerencia o histórico de jogadas (`history`) e permite navegar entre estados anteriores. Usa a técnica de "time travel" para reverter o estado do jogo.

1.4.4. `calculateWinner`

Função auxiliar que verifica combinações vencedoras (linhas, colunas e diagonais).

1.5. Conceitos Praticados

- Componentes funcionais e reutilização
- Hooks: `useState`
- Props e callbacks entre componentes
- Imutabilidade do estado (`slice()`)
- "Lifting state up"
- Renderização condicional
- Mapeamento de listas (`map()`)

1.6. Resultado Esperado

- O tabuleiro exibe 9 quadrados.
- Cada clique alterna entre "X" e "O".
- O jogo mostra qual jogador é o próximo ou quem venceu.
- O histórico de jogadas é exibido e permite "viajar no tempo".

1.7. Referência

Tutorial oficial: React - Jogo da Velha (Tic Tac Toe)

Aqui terminamos.

Bons estudos!
Prof. João Ferreira