

## RELATÓRIO LINGUAGENS SCRIPT – “Ultimate Tic-Tac-Toe”

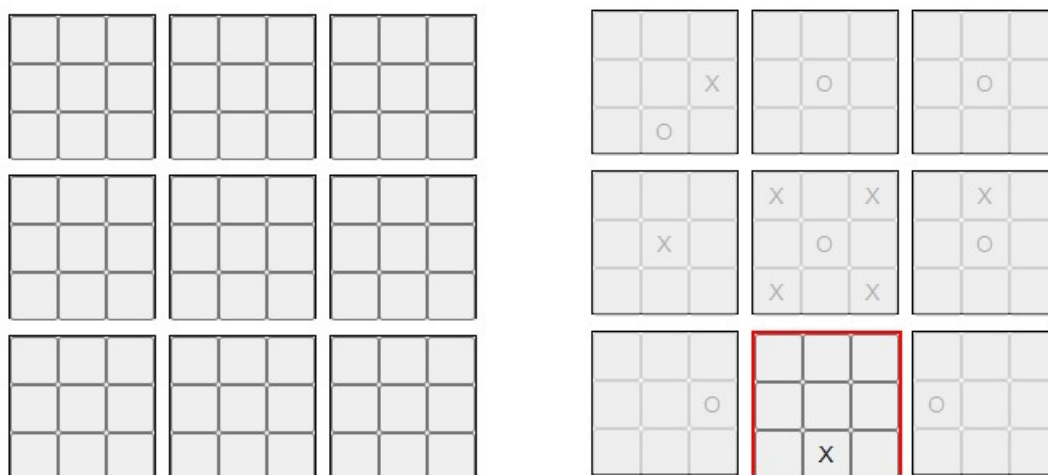


Figura 1 – Interface do jogo Ultimate Tic-Tac-Toe

### > RESUMO

Este trabalho prático tem como objetivo o desenvolvimento de uma aplicação em ReactJS, na qual os alunos devem aplicar todos os conhecimentos adquiridos ao longo das aulas. Como tal, devem demonstrar o domínio na tecnologia React assim como das tecnologias necessárias ao desenvolvimento de uma solução coerente e consistente, nomeadamente JavaScript, HTML e CSS.

### > TEMA

O tema do trabalho prático é uma variante do tradicional jogo do galo, designado como “Ultimate Tic-Tac-Toe”. O jogo está organizado por 9 mini-tabuleiros do jogo do galo, dispostos numa grelha 3x3, sendo que o vencedor do jogo é aquele que fizer 3 em linha (seja na horizontal, vertical ou diagonal) do tabuleiro geral, ou aquele que tiver mais tabuleiros quando o tempo de jogo acabar.

## > **EQUIPA DE TRABALHO**

O trabalho prático foi realizado pela seguinte equipa:

- Dinis Meireles de Sousa Falcão | [a2020130403@isec.pt](mailto:a2020130403@isec.pt)
- David Pedro de Seco Rodrigues | [a2019130152@isec.pt](mailto:a2019130152@isec.pt)

## > **COMPONENTES**

O jogo é acessível através do browser e disponibiliza as seguintes funcionalidades:

- Solicitar o nome dos jogadores;
- Decide de forma aleatória o primeiro jogador bem como o símbolo associado, e apresenta essa informação;
- Apresentação do tabuleiro geral de jogo;
- Identificação do jogador que deve efetuar a jogada;
- Seleção da posição do mini-tabuleiro com o rato;
- A escolha do mini-tabuleiro:
  - Opção 2: não é livre, sendo determinada pela jogada anterior do adversário. Isto é, a posição escolhida pelo adversário identifica o mini-tabuleiro onde deve ser continuado o jogo.
- Identificação do vencedor de um mini-tabuleiro, com a colocação do símbolo do vencedor em todos os quadrados do mini-tabuleiro;
- Quando um mini-tabuleiro fica fechado em termos de vitória ou empate, este fica bloqueado, sem permitir novas jogadas;
- 1 nível de jogo: 1 contra 1 jogador;
- Tempo de jogo limitado, sendo o vencedor quem tiver mais vitórias nos tabuleiros pequenos;
- Identificação de fim de jogo, quando todos os mini tabuleiros estiverem concluídos, assim como o vencedor;
- Permitir jogar novamente.

## > **LIMITAÇÕES CONHECIDAS**

Não implementámos a navegação pelas células disponíveis com o teclado, visto que era opcional. Além disto, também optámos por não fazer o nível de jogo “1 contra computador”.

## > **DESAFIOS**

Inicialmente, custou-nos um pouco perceber qual seria a melhor estratégia para realizar este trabalho. Após perceber que o trabalho não seria tão complexo como pensávamos, penso que o nosso principal desafio foi a separação do vencedor por esgotamento do tempo e do vencedor por concluir um dos “combos”.

## > **DESCRIÇÃO DO CÓDIGO**

### ○ **App.js**

A função Square é uma componente do tipo botão que representa cada quadrado do tabuleiro. Recebe várias propriedades, como value (valor do quadrado), onSquareClick (função de clique no quadrado), isPlayable (indicador se o quadrado é jogável) e winner (indicador se o quadrado faz parte da sequência vencedora).

A função App é a componente principal do jogo. Utiliza vários Hooks para gerir o estado do jogo. gameStarted: estado que indica se o jogo foi iniciado ou não; xIsNext: estado que indica se é a vez do jogador X; squares: estado que representa o tabuleiro do jogo. É uma matriz 2D (9x9) que armazena os valores dos quadrados; activeBoard: estado que indica o tabuleiro ativo, ou seja, o tabuleiro em que o próximo jogador deve jogar; timerX e timerO: estados que representam os tempos de cada jogador; isAgainstComputer: estado que indica se o jogador está a jogar contra o computador; overallWinner: estado que representa o vencedor geral do jogo.

A função useEffect é usada para atualizar o tempo de cada jogador e verificar se o jogo terminou. Se o jogo estiver em andamento e não houver vencedor geral ou vencedor em algum tabuleiro específico, o cronômetro é incrementado a cada segundo. Se um jogador exceder x segundos, o jogo é encerrado.

A função handleClick é chamada quando um jogador clica num quadrado. Verifica se o jogo já terminou ou se o quadrado selecionado é jogável. Em seguida, atualiza o estado do tabuleiro, verifica se houve um vencedor no tabuleiro específico e define o próximo tabuleiro ativo. Se estiver a jogar contra o computador, a função makeComputerMove é chamada após um atraso de 500ms.

A função resetGame reinicia o jogo, redefinindo os estados iniciais.

A função makeComputerMove é onde o movimento do computador seria implementado, mas a implementação não foi realizada.

A função **calculateWinner** verifica se há um vencedor num tabuleiro específico. Ela recebe uma matriz de tabuleiros e o índice do tabuleiro ativo (se houver). Verifica todas as combinações vencedoras possíveis (linhas, colunas e diagonais) e retorna o vencedor, se houver.

A função **calculateOverallWinner** verifica se há um vencedor geral no jogo. Recebe uma matriz de tabuleiros e verifica as combinações vencedoras nas primeiras posições dos tabuleiros (índice = 0). Retorna o vencedor geral, se houver.

A componente **App** renderiza a estrutura do jogo.

### ○ **control-panel.component.jsx**

A função **ControlPanel** é uma componente **React** que representa o painel de controlo do jogo Ultimate Tic-Tac-Toe. Ele mostra os campos de entrada para os nomes dos jogadores e um botão para atribuir símbolos aleatórios aos jogadores. Utiliza os **Hooks useState** para gerir o estado dos nomes dos jogadores e os símbolos atribuídos a cada jogador.

A função **handleSubmit** é chamada quando o formulário é submetido. Gera um valor aleatório entre 'X' e 'O' usando a função **Math.random()** e **Math.floor()**. Atribui o símbolo gerado ao jogador 1 e o símbolo oposto ao jogador 2. Mostra na consola os nomes dos jogadores.

A componente **ControlPanel** é responsável por exibir o painel de controlo do jogo, permitindo que os jogadores insiram os seus nomes e atribuam símbolos aleatórios.

## > **INTERFACE**

The screenshot displays the user interface for the 'Ultimate Tic-Tac-Toe' game. At the top, the header reads 'LINGUAGENS SCRIPT 2022/2023' and 'LICENCIATURA EM ENGENHARIA INFORMÁTICA'. Below this is a light purple box containing the title 'JOGO ULTIMATE TIC-TAC-TOE'. Inside this box, there are two rows for player information: 'JOGADOR 1:' with a text input and 'Símbolo Aleatório:' with a dropdown menu, and 'JOGADOR 2:' with a text input and 'Símbolo Aleatório:' with a dropdown menu. A green button labeled 'Atribui Símbolos' is positioned below these inputs. Below the purple box is a button labeled 'Iniciar Jogo'. Underneath that, it says 'Player Turn: X'. The main game area consists of a 3x3 grid of smaller 3x3 grids, all currently empty. Below the grid is a button labeled 'Jogar contra o computador'. At the bottom of the grid area is a button labeled 'Reiniciar Jogo'. On the left side of the interface, there is a small box with a timer: 'Player 1: 0 seconds' and 'Player 2: 0 seconds'. At the very bottom of the page, the text '©2022-2023' is visible.

**FIM**