



Instituto Superior de Engenharia

Politécnico de Coimbra

DEPARTAMENTO DE / DEPARTMENT OF
INFORMÁTICA E SISTEMAS

Minesweeper

Autor / Author

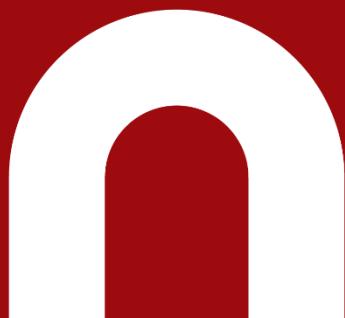
**Tiago Fernandes Lopes – a2023144639,
Engenharia Informática**

Autor / Author

**Daniel Duarte Silva – a2023144551,
Engenharia Informática**

Autor / Author

**Guilherme Alexandre Neves Martins – a2023144573,
Engenharia Informática**



INSTITUTO POLITÉCNICO
DE COIMBRA

INSTITUTO SUPERIOR
DE ENGENHARIA
DE COIMBRA

Coimbra, junho de 2024

ÍNDICE

1	Introdução	3
2	Componentes	4
3	Funcionalidades.....	5
4	Soluções utilizadas	8
5	Conclusão	9
6	Bibliografia.....	9

ÍNDICE DE FIGURAS

Figura 2-1 Lista de componentes	4
Figura 2-2 Diagrama de componentes.....	4
Figura 3-1 Estado inicial do tabuleiro	5
Figura 3-2 Tamanho do tabuleiro 16x16.....	5
Figura 3-3 Exemplo de um jogo.....	6
Figura 3-4 Pop-up a informar que o jogador perdeu.....	6
Figura 3-5 Mostrar todas as células ao perder	7
Figura 3-6 Pop-up para adicionar o nome a lista de melhores tempos	7
Figura 3-7 Lista de tempo dos melhores tempos	8

1 INTRODUÇÃO

O trabalho prático de Linguagens Script trata-se de uma recriação do jogo Minesweeper (Campo minado) utilizando React e Typescript. O jogo tem um objetivo simples que é “abrir” todas as células do tabuleiro sem selecionar uma mina. Quando o utilizador clica em uma mina o jogo acaba e o utilizador perde. O utilizador pode marcar células onde ele tem a certeza de estar uma mina, com uma bandeira para organizar o seu jogo. Para além de marcar uma célula com a presença de mina também pode marcar a célula como “possível presença de mina” sempre que não tiver certeza.

A aplicação está dividida em componentes funcionais para uma melhor gestão do projeto e utiliza hooks para a gestão do estado do jogo.

2 COMPONENTES

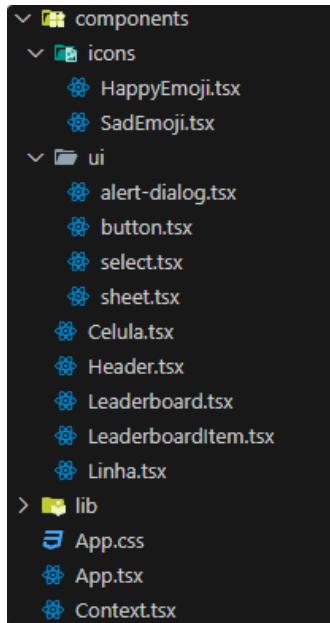


Figura 2-1 Lista de componentes

Os componentes de maior importância são:

- App
- Celula
- Header
- Context (é um componente especial será explicado brevemente)

Nestes componentes está toda a lógica do jogo desde o temporizador até à seleção das células. O diagrama de componentes da aplicação tem este aspecto:

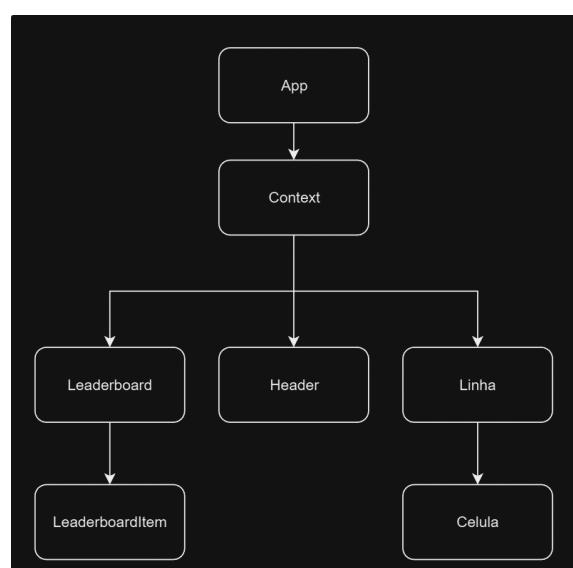


Figura 2-2 Diagrama de componentes

3 FUNCIONALIDADES

O tabuleiro tem como predefinição 9 linhas e 9 colunas e um total de 10 minas.

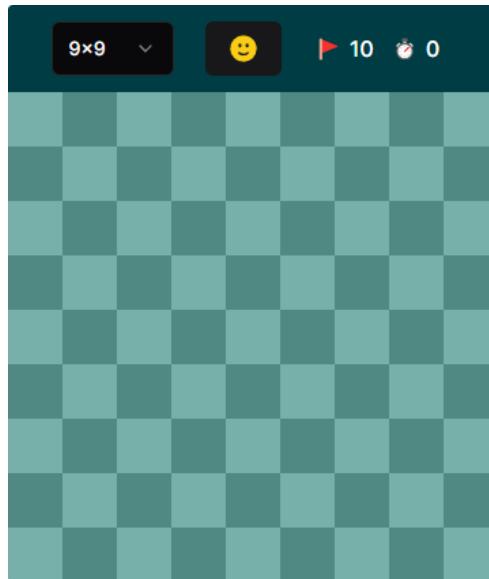


Figura 3-1 Estado inicial do tabuleiro

A dificuldade do jogo (tamanho do tabuleiro e quantidade de minas) pode ser alterada no selecionador no canto esquerdo do tabuleiro. Existem 3 dificuldades:

- 9x9, com 10 minas;
- 16x16, com 40 minas;
- 16x30, com 99 minas.

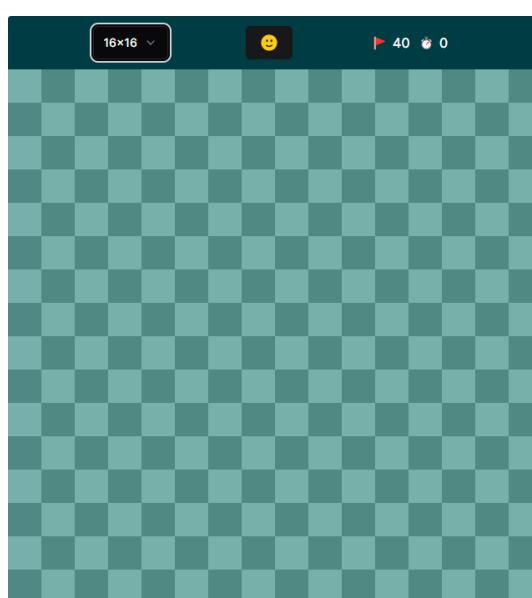


Figura 3-2 Tamanho do tabuleiro 16x16

Para reiniciar o jogo o utilizador clica no botão do meio do cabeçalho, este foi inspirado pelo jogo original de Minesweeper. No cabeçalho também temos a duração do jogo e a quantidade de bandeiras que o utilizador pode colocar.

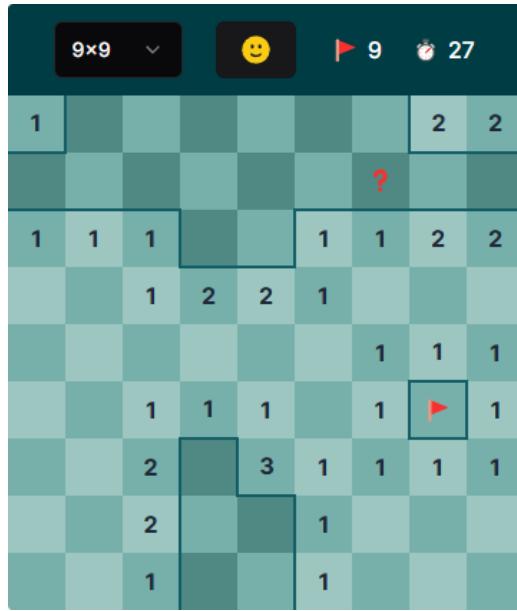


Figura 3-3 Exemplo de um jogo

O utilizador seleciona células ao clicar com o botão esquerdo do rato, se a célula for vazia serão abertas células até encontrar células com minas adjacentes. O botão direito do rato permite alternar entre os estados da célula:

- presença de mina, representada por ;
- possível presença de mina, representada por ;
- normal.

Se o utilizador perder então aparecerá uma mensagem a informar que perdeu e são mostradas todas as células do tabuleiro.

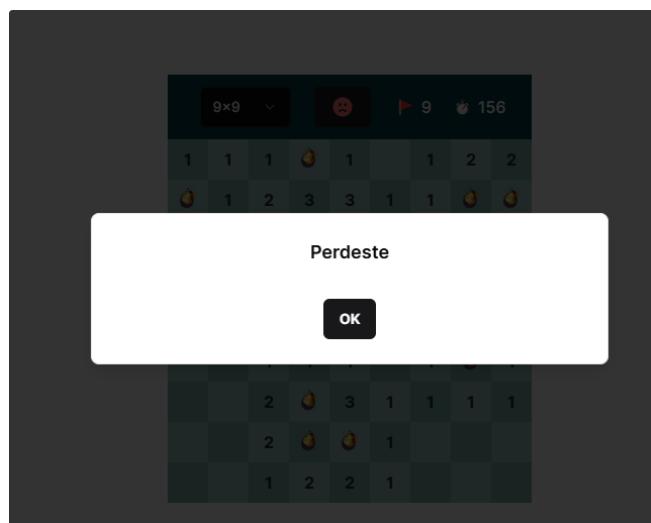


Figura 3-4 Pop-up a informar que o jogador perdeu

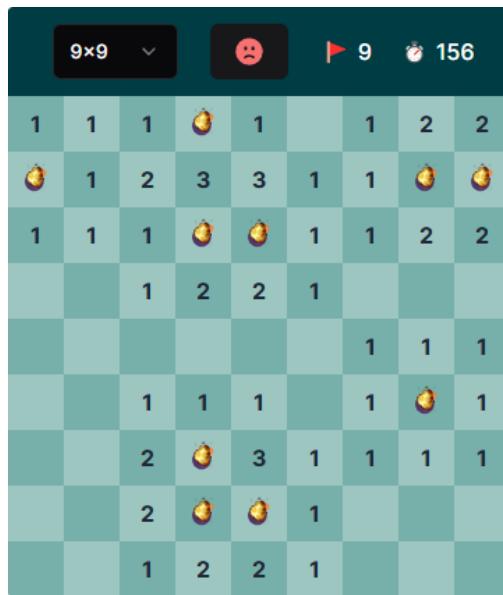


Figura 3-5 Mostrar todas as células ao perder

Quando o jogador ganha é pedido que introduza o seu nome para adicionar a uma lista dos melhores tempos por dificuldade. Caso o jogador não pretenda adicionar o seu nome pode apenas cancelar.

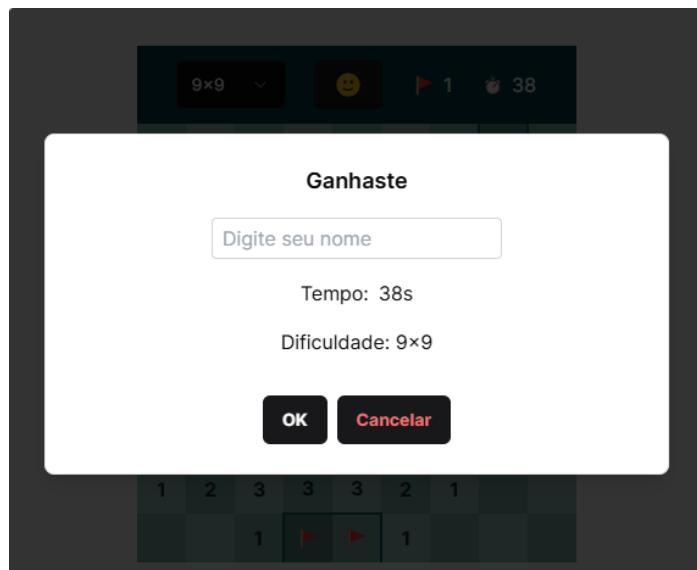


Figura 3-6 Pop-up para adicionar o nome a lista de melhores tempos

A lista de melhores tempos é acessível ao clicar no botão “Leaderboard” no lado direito da página. Existe um seletor de dificuldade e os jogadores estão ordenados pelo tempo que demoraram a acabar o jogo.

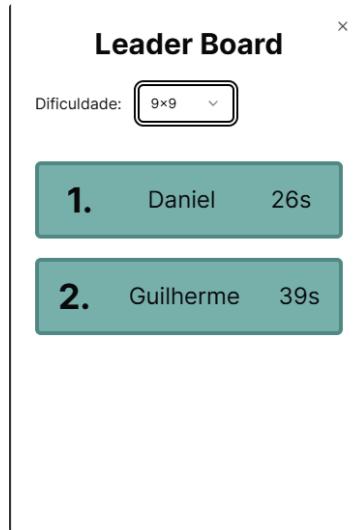


Figura 3-7 Lista de tempo dos melhores tempos

4 SOLUÇÕES UTILIZADAS

O Typescript foi utilizado para fornecer uma melhor experiência de desenvolvimento do jogo. Diminui a possibilidade de utilizar métodos inexistentes ou usar propriedades com nome inválido.

A gestão do estado do jogo é feita com o hook “useReducer”. Preferimos este hook como alternativa a vários hooks “useState”, pois apesar de ser necessário a escrita de mais código inicialmente, com o aumento de complexidade da aplicação as interações ficam todas na mesma função. Esta decisão ajudou bastante nos testes e “debug” do jogo. Utilizamos hooks “useState” na aplicação, mas para situações mais simples como por exemplo o temporizador.

O componente Context é extremamente importante pois nele está o estado “global”. Um “Context” permite que um componente receba informação sem ser necessário enviar por “props”, ou seja, temos um estado “global” ao qual podemos aceder com o seguinte hook:

```
useContext(MinesweeperContext)
```

Este estado “global” é utilizado em toda a aplicação para que os componentes filho: Celula, Header, etc., possam atualizar o estado.

O algoritmo utilizado para selecionar as células vazias é o “Breadth First Search” onde percorre todas as células á volta de uma célula e adiciona-as a uma lista de espera. A lista de espera é percorrida e sempre que uma célula estiver vazia as células vizinhas são adicionadas à lista de espera e assim sucessivamente. Todas as células percorridas são selecionadas.

5 CONCLUSÃO

Este jogo foi um bom desafio para testar as nossas capacidades de React e aprender mais sobre o uso do Typescript. A utilização de um algoritmo de pesquisa também nos forneceu novos conhecimentos úteis para o nosso futuro.

6 BIBLIOGRAFIA

Scaling Up with Reducer and Context – React. (n.d.). React.dev.

<https://react.dev/learn/scaling-up-with-reducer-and-context>

useReducer – React. (n.d.). React.dev. <https://react.dev/reference/react/useReducer>

Wikipedia Contributors. (2019, May 28). Breadth-first search. Wikipedia; Wikimedia

Foundation. https://en.wikipedia.org/wiki/Breadth-first_search



**Instituto Superior
de Engenharia**

Politécnico de Coimbra