

**Linguagens Script**

Relatório

Sopa de letras

**Da autoria de**

Carlos Manuel Pascoal Pinto nº2021155627

Hugo Miguel Rainho Marques nº2021138225

Alexandre José Leonardo Ferreira nº2021138219

**Docente**

Cristiana Areias

15 de junho de 2022

**Introdução**

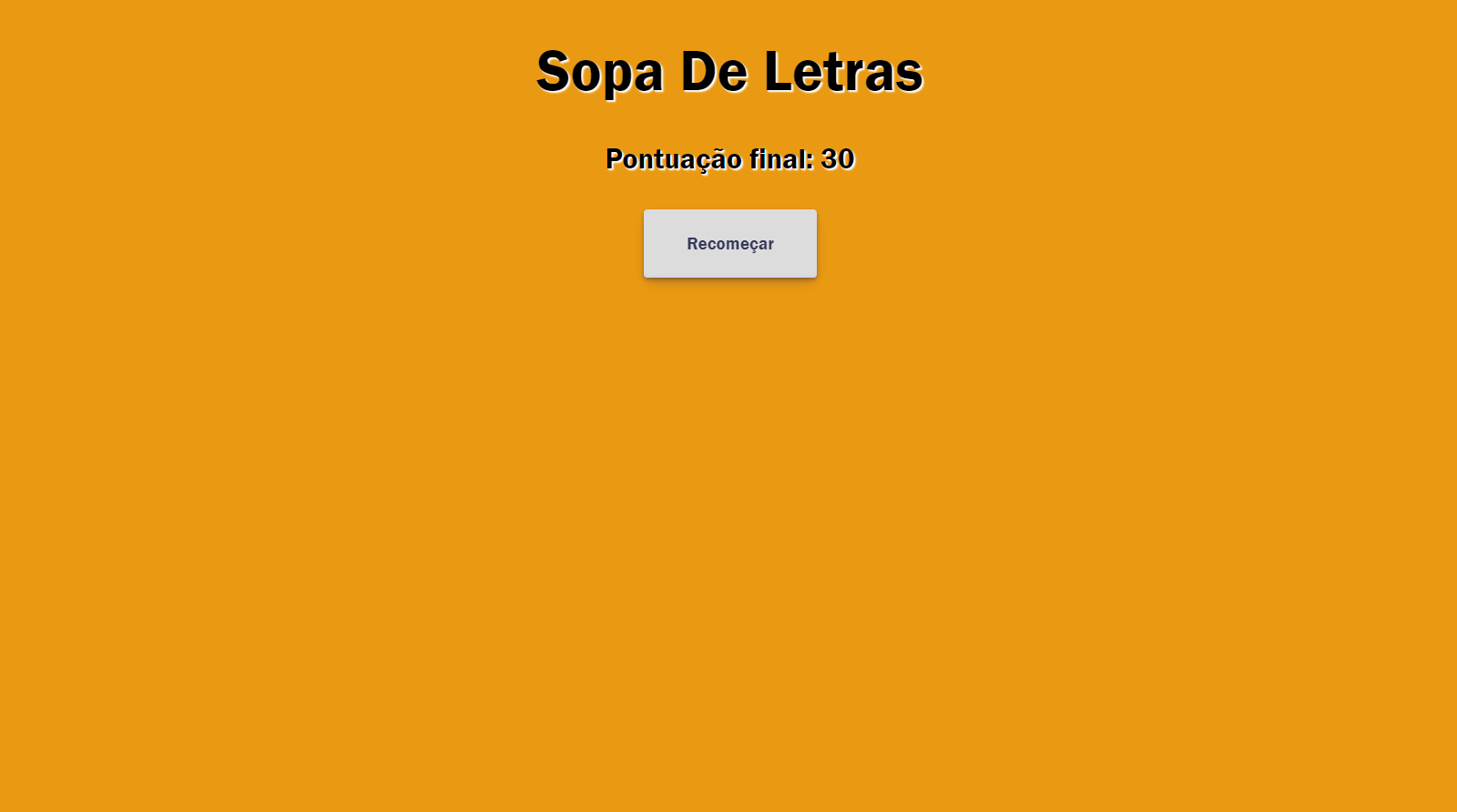
Neste trabalho prático temos como principal objetivo desenvolver uma aplicação em React, com base em Java Script. Utilizando os conhecimentos adquiridos nas aulas, construímos uma réplica de numa típica sopa de letras de forma a demonstrar o nosso domínio de Java Script, React e por consequência HTML e CSS.

**Funcionalidades**

* **Escolha de nível:**
  + Encontra-se disponível um sistema de alteração de níveis que além de alterar o número de letras, altera o número de palavras a encontrar (escolhidas aleatoriamente de uma lista presente no ficheiro de constantes).
    - É possível jogar nos níveis:
      * Básico: 14\*14 Letras.
      * Intermédio: 16\*16 Letras.
      * Avançado: 18\*18 Letras.
* **Disposição de palavras:**
  + As palavras encontram-se dispostas em todos os sentidos e nas direções:
    - Horizontal
    - Vertical
    - Diagonal
* **Temporizador:**
  + Ao longo do jogo, aplica-se um temporizador regressivo que para além de variar consoante o nível escolhido, faz o jogo terminar quando termina.
* **Pontuação:**
  + Quanto ao sistema de pontuação (sendo que não foram dadas indicações a seguir para o seu desenvolvimento), foi criado um algoritmo que:
    - Obtém os segundos em falta para o término do jogo, multiplica por 1.5, divide por 10000 e após arredondar soma aos pontos já conseguidos.
* **Lista de palavras:**
  + Juntamente com o puzzle da sopa de letras é gerada uma lista de palavras a encontrar. Esta tem algumas propriedades especificas:
    - O número de palavras depende da dificuldade escolhida (4, 7 e 10 palavras respetivamente consoante o aumento dos níveis).
    - É ainda possível adicionar palavras à lista sendo que o desenvolvimento deste algoritmo foi feito de maneira a ser possível adicionar palavras num dos níveis e estas continuarem a estar incluídas na lista, mesmo após a alteração do nível.
    - As palavras adicionadas não passam pelo processo de escolha aleatória e assim, encontram-se sempre incluídas nas palavras a encontrar.
    - Cada vez que uma das palavras é encontrada, aplica-se uma regra de CSS de forma a riscar a palavra na lista, como é possível ver na figura seguinte.

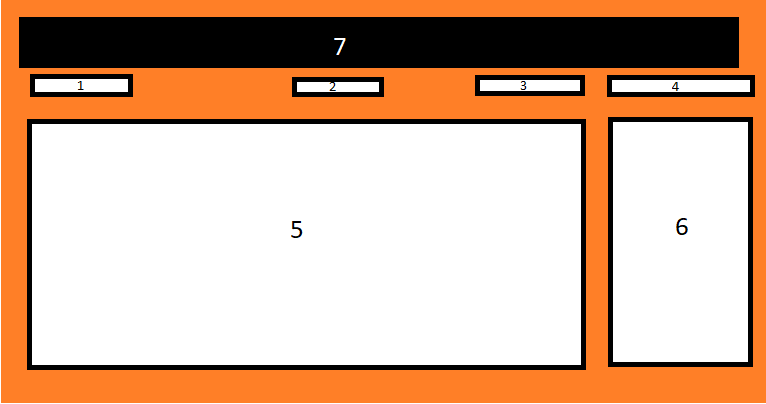


*Figura 1-Interface da sopa de letras*



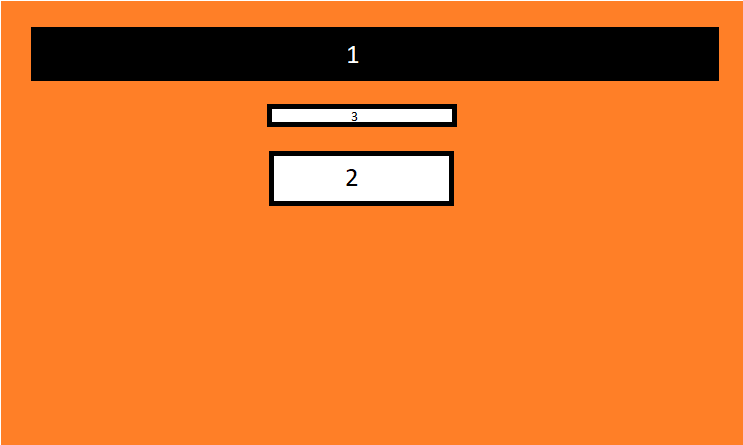
*Figura 2-Ecra de fim de jogo*

**Componentes**



*Figura 3-Divisão em componentes*

1. **Nível-**Escolha do nível de dificuldade.
2. **Timer-**Tempo de jogo restante.
3. **Pontuação-**Pontos obtidos até ao momento.
4. **Adiciona palavra-**Lugar onde é possível adicionar palavras à lista.
5. **Puzzle-**Tabuleiro da sopa de letras.
6. **Lista-**Lista das palavras a encontrar e encontradas.
7. **Header-**Titulo do programa.



*Figura 4-components do ecrã final*

1. **Header-**Contém o título
2. **Recomeçar-** Botão para voltar a jogar
3. **Pontuação-**Pontuação total obtida.

**Desafios**

Ao longo do desenvolvimento do jogo, a maior dificuldade encontrada foi ao desenvolver a função “escreve” que faz todos os cálculos necessários para colocar as palavras no puzzle de forma correta (sem se cortarem e só se cruzarem se partilharem a(s) letra do cruzamento).

**Conclusão**

Com este trabalho aumentamos o nosso conhecimento da biblioteca React e consequentemente desenvolvemos ainda o nosso conhecimento em Java Script.

Além disto acabamos por desenvolver um programa útil e funcional de maneira que poderemos sempre retornar ao trabalho desenvolvido na UC de JS para resolver sopas de letras.