

## **Trabalho Prático em React – “Sopa de Letras”**

### **> Âmbito**

Este trabalho prático tem como objetivo o desenvolvimento de uma aplicação em **React JS**, 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 nas tecnologias necessárias ao desenvolvimento de uma solução coerente e consistente, nomeadamente JavaScript, HTML e CSS.

Os alunos devem organizar-se, obrigatoriamente, **em grupos de 3 alunos**. Qualquer alteração a esta regra, deverá ser previamente acordada com a docente responsável pela disciplina, através do endereço de email [cris@isec.pt](mailto:cris@isec.pt). Todos os alunos devem colaborar na implementação do trabalho, e na implementação de componentes React, sob pena de, na defesa, lhes ser atribuído 0 na nota do trabalho.

Todos os detalhes do trabalho encontram-se especificados nas secções seguintes e, como tal, **sugere-se a leitura com a devida atenção** sob pena de não implementarem o trabalho na sua totalidade, por descuido. Recorda-se que o trabalho tem um peso de 8 valores da nota final da disciplina, como especificado na FUC de Linguagens Script.

Valoriza-se a atuação autónoma do aluno na construção de uma solução para o tema proposto, tendo em consideração que esta deve cumprir, no mínimo, os requisitos gerais apresentados.

### **> Tema**

O tema do trabalho prático é o tradicional jogo **“Sopa de Letras”**, no qual deve ser apresentada **uma lista de palavras** que devem ser encontradas numa grelha composta por diferentes letras, entre elas, as palavras a encontrar. As palavras podem estar distribuídas na horizontal, vertical ou diagonal, e em várias direções, isto é:

- esquerda → direita
- direita → esquerda
- cima → baixo
- baixo → cima.

As figuras seguintes **apresentam um exemplo do jogo** a implementar, não estando nessas figuras a opção de seleção do nível de jogo e outras das funcionalidades referidas na secção seguinte.



Figura 1 – Exemplos do Jogo Sopa de Letras

## > Funcionalidades

O jogo a implementar deverá ser acessível através do browser e deverá disponibilizar, pelo menos, as seguintes **funcionalidades**:

- Apresentação de uma lista de palavras aleatórias (de entre uma lista disponível, definida num *array* ou num ficheiro);
- Apresentação da grelha que incluam as palavras em posição aleatória;
- Seleção da palavra na grelha (com ajuda do rato e/ou teclado);
- 3 níveis de jogo (simples, intermédio e avançado) em que a dimensão da grelha e número de palavras varia;
- Tempo de jogo limitado, dependendo do nível;
- Pontuação (metodologia ao critério dos alunos);
- Identificação de fim de jogo;
- Permitir jogar novamente;
- Permitir a especificação de outras palavras, para além das já existentes no array.

## > **Implementação**

A implementação do trabalho, como anteriormente referido, deverá ser efetuado em React, recorrendo à versão atual, versão 18. Em relação à implementação, é dada a liberdade ao aluno para seguir a estratégia que considerar melhor, no entanto, especificam-se abaixo algumas características e limitações que devem ter em consideração.

### → **Método**

Os alunos devem criar um projeto de forma a facilitar a gestão de dependências de todos os módulos necessários à implementação de uma aplicação react. Assim, recomenda-se o uso da aplicação `create-react-app` para criação do projeto, mas é dada liberdade aos alunos para usarem outra abordagem, desde que devidamente explicada no relatório.

Em relação à criação dos componentes, o React permite a implementação dos componentes usando abordagens diferentes, seja *functional components* (em conjunto com hooks) ou *class components*, no entanto, o trabalho deverá ser implementado, **na sua maioria**, com componentes funcionais. Os alunos que assim o desejarem, poderão também explorar componentes de classe (alerta-se, no entanto, que quem seguir esta opção, não será considerada como extra na nota do trabalho prático).

### → **Interface**

Fica à responsabilidade do aluno a especificação da interface da aplicação. Esta não tem de ter o aspeto do exemplo apresentado nas figuras anteriores. Deve haver uma preocupação no sentido de desenvolver uma aplicação visualmente agradável com uma imagem consistente. Poderão ser usadas bibliotecas e *frameworks* CSS que considerarem úteis e possam facilitar a implementação da interface, como por exemplo, Bootstrap, Tailwind CSS, entre outros. No entanto, **não é permitido** recorrer a bibliotecas e/ou frameworks que disponibilizem componentes React já implementados, como por exemplo, *Material UI*, *React Bootstrap*, *Semantic UI*, *React Toolbox*, *Ant Design*, *React Foundations*, entre outros. Todos os componentes necessários à implementação do jogo devem ser implementados pelos alunos.

Nesse sentido, sempre que o aluno utilize componentes previamente implementados por terceiros (bibliotecas/frameworks, entre outros), será considerado que o aluno não implementou esse componente nem as respetivas funcionalidades.

## > **Relatório**

O trabalho prático deve ser acompanhado de um pequeno relatório (que não ultrapasse as dez páginas) onde devem especificar as funcionalidades implementadas bem como as soluções utilizadas no desenvolvimento da aplicação. Além disso, devem ser especificados os componentes, ou outros elementos que tenham sido utilizados no desenvolvimento da aplicação.

## > **Condições e Data de Entrega**

A **data limite** para entrega do trabalho prático é o **dia 18 de Junho de 2022, sábado, até às 23:59**. A entrega do trabalho prático é realizada em formato digital, **no inforestudante, apenas por um elemento do grupo**, devendo ainda especificar no momento da submissão os outros elementos do grupo de trabalho. O trabalho prático deve ser submetido num ficheiro **ZIP** com o nome **LS2122\_numAluno1\_numAluno2\_NumAluno3.ZIP**.

**NOTA IMPORTANTE:** Antes de criar o ficheiro ZIP, os alunos devem **obrigatoriamente** remover a pasta “*node\_modules*” e o ficheiro “*package-lock.json*” existente na estrutura do projeto. Para além do projeto implementado, que deve incluir o código fonte (sem a pasta), o ficheiro Zip deve incluir o relatório em formato pdf.

Recomenda-se que os alunos testem o trabalho que submetem, sob pena, de não entregarem a versão correta ou pretendida, como frequentemente acontece....

## > **Defesa**

O trabalho terá uma defesa obrigatória, que será efetuada na semana de 20 a 24 de junho, através de uma pré-inscrição no *moodle*. **A não comparência** à defesa na data (dia e hora) combinada, implica reprovação do aluno à disciplina.

## > **CrITÉrios de Avaliação**

- Aspeto Gráfico (15% da percentagem total do trabalho);
- Apresentação de uma lista de palavras aleatórias; (5%)
- Preenchimento da grelha com palavras e letras de forma aleatória; (20%)

- Encontrar palavra na grelha e retirar/destacar que já foi encontrada na lista: (30% opção 2)
  - **(opção 1)** Selecionando apenas a primeira e a última letra, e esta ser depois destacada na grelha.
  - **(opção 2)** Selecionando a primeira letra e arrastando o rato na direção certa até à última letra com destaque de forma gradual na grelha (**opção mais valorizada**).
- Pontuação do Jogo (5%)
- Identificação de fim de jogo – Temporizador (5%)
- Vários níveis de jogos (10%)
- Permitir a especificação de novas palavras (10%);
- *Critérios gerais a aplicar à nota final:*
  - Organização do código e metodologia usada na implementação da aplicação, nomeadamente, forma de estruturação da aplicação em vários componentes React, o uso correto de hooks (quando aplicável) na aplicação, entre outros.
  - Defesa do aluno.