> *Ficha Prática Nº6 (Jogo de Memória – Top 10 e LocalStorage)*

Notas:

- Esta ficha tem como objetivo implementar o código necessário para apresentar, atualizar e gravar o top10.
- O resultado final da ficha apresenta-se na figura 1.

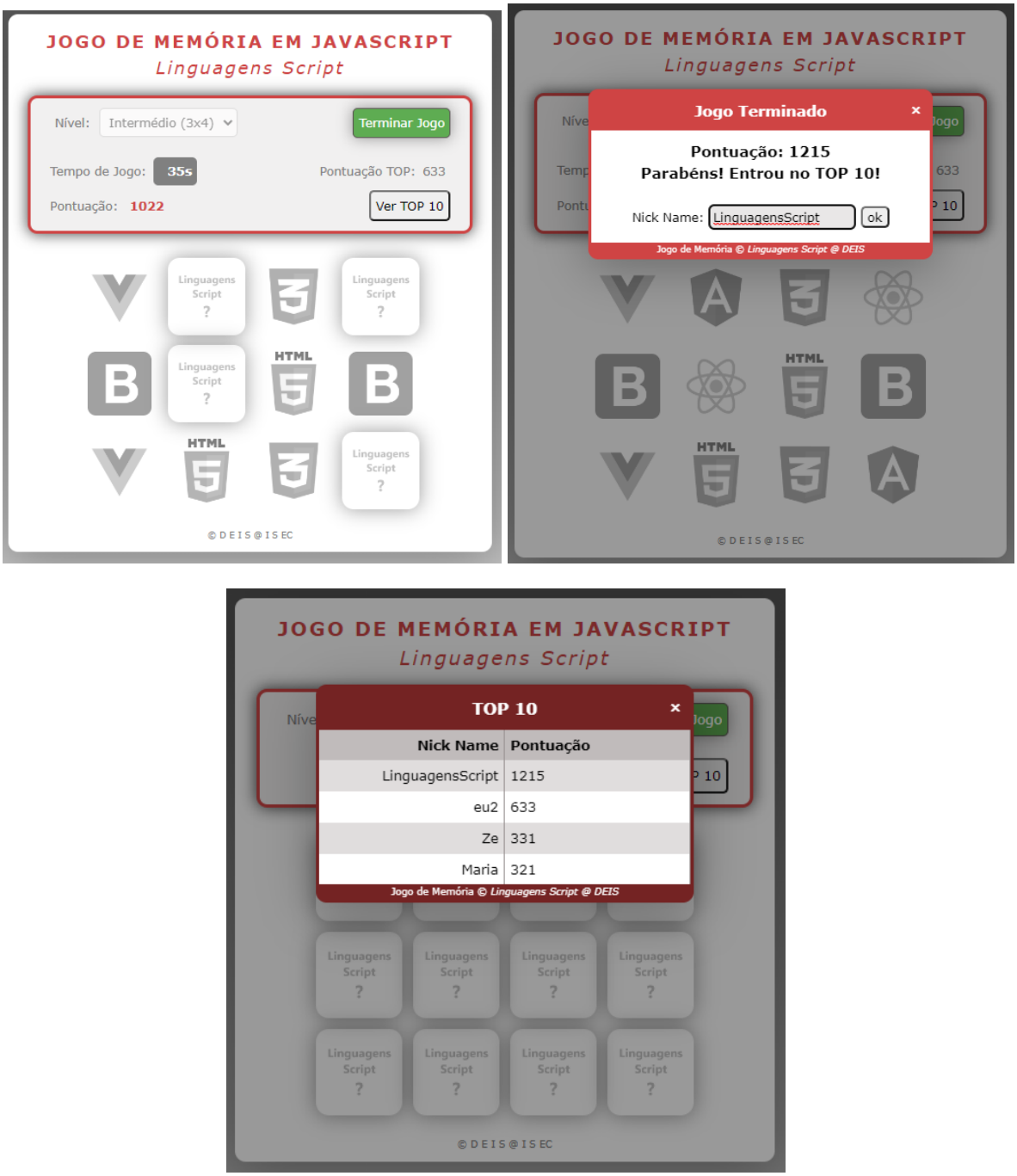

Figura 1 – Jogo de Memória em JavaScript – Imagens da aplicação – Top 10

> *Preparação do ambiente*

a. Efetue o download e descompacte o ficheiro **ficha6.zip** disponível no *inforestudante*.

NOTA:

Os alunos que concluíram a resolução da ficha anterior, devem continuar nesse projeto. Os restantes alunos devem usar o código disponibilizado em ficha6.zip, no entanto, *o código disponibilizado não contém a resolução de todas as fichas anteriores, apenas o necessário para esta ficha, permitindo apenas o jogo no nível básico.*

Figura 2 - Jogo (inicio)

b. Inicie o *Visual Studio Code*, abra **a pasta da ficha no *workspace*** e visualize a página **index.html** no browser (recorra à extensão "*Live Server*"), no qual terá o aspeto da figura 2.

Parte I – Apresentação do TOP 10

1> Nesta fase, pretende-se apresentar a lista do TOP 10, quando se clica no botão Ver TOP 10, como se apresenta na figura seguinte.

Figura 3 - Top 10

a. Defina um array de objetos, global, de nome **`topGamers`** e inicialize-o com dois objetos. Cada objeto é constituído pelas propriedades `nickname` e `points`. Pode usar os valores apresentados na figura acima.

Abaixo um exemplo para definir um array de objetos:

```
let nomeArray = [
                  { nomePropriedade: 'exemplo1', nomePropriedade2: 1 },
                  { nomePropriedade: 'exemplo2', nomePropriedade2: 2 }
                ]
```

b. Implemente a função `getTop10`

- Na função, declare a variável `infoTop` que deve aceder ao elemento da página cujo id é `infoTop`. Será necessário para escrever a lista dos jogadores do TOP 10.

- Implemente o ciclo que percorra o array **topGamers** e construa uma *string* com os valores dos elementos existentes no array, no seguinte formato nickName – points.
- Com recurso à variável `infoTop` e à propriedade `innerHTML`, escreva a *string* construída no painel do top10, de forma a ficar com apresentada na figura seguinte.

- Adicione um *event listener* ao **btTop** (não necessita de definir btTop pois já existe essa constante definida em *modal.js*), de forma a invocar a função `getTop10` quando há um click no botão TOP10.
- Confirme no **browser** o comportamento.

c. Como pode verificar, o aspeto da janela TOP 10, não em o aspeto pretendido. Assim, implemente os seguintes passos:

- Coloque em comentário a código implementado na alínea anterior, mais propriamente, a criação da string e apresentação dos jogadores no top10.

- De forma a ficar com o aspeto pretendido, pretende-se criar dinamicamente elementos HTML, manipulando o DOM, de forma a criar a estrutura apresentada abaixo.

```
▼<div class="info" id="infoTop">
  ▼<div>
     <p>Nick Name</p>
     <p>Pontuação</p>
   </div>
  ▼<div>
     <p>Ze</p>
     <p>331</p>
   </div>
  ▼<div>
     <p>Maria</p>
     <p>321</p>
   </div>
</div>
```

- Como pode verificar, cada jogador (objecto) corresponde a um bloco DIV com dois parágrafos, um para o nickname e outro para a pontuação. Assim, substitua o código anterior de forma a criar estes elementos dinamicamente com recurso aos métodos e propriedades para manipulação do DOM como efetuado para criação do painel dinâmico na ficha 5. Os métodos e propriedades necessárias são:
  - `createElement()` - método que permite criar um elemento.
  - `appendChild()` - método que permite anexar um elemento (nó) como o último filho de um elemento.

- `elemento.cloneNode(true)` – método que permite criar uma cópia de um elemento e de todos os filhos, os atributos e valores. Retorna o elemento clonado.
- `textContent` – permite introduzir o texto no elemento p
- `firstChild /lastChild` – permitem aceder ao primeiro / ultimo filho de um elemento

- Confirme no browser a visualização do TOP 10 que já deverá ter o aspeto desejado.

2> Pretende-se apresentar a pontuação TOP quando se inicia o jogo, como apresentado na figura abaixo.

a. Para isso, implemente a função `getTopPoints` que deve ser invocada na função `startGame`. A função getTopPoints, deve apenas obter os pontos do primeiro elemento no array e escrever no elemento cujo ID é `pointsTop`.

b. Confirme o comportamento no browser.

Parte II – Verifica se entra no TOP 10

3> Nesta fase, pretende-se verificar se o utilizador entra no TOP 10. Se entrar é solicitado o nick name como apresentado na figura seguinte.

Para isso, implemente os seguintes passos:

a. Implemente a função `getLastPoints` que deverá retornar a pontuação da último jogador existente na lista dos topGamers.

b. Implemente a função `checkTop10` que permite verificar se o jogador entra ou não no top 10 e, se entrar, deve ser apresentada a caixa para especificar o seu **`nickname`**.

- A função, que deve ser invocada na função `stopGame`

- Para um utilizador poder entrar no TOP10, deve garantir uma das condições: se ainda não existirem 10 jogadores na lista ou então, se existir, a pontuação é superior à ultima. Assim, usando as funções realizadas nas alíneas a) e b) implemente a função. Para mostrar a caixa de dialogo do nickname, declare a variável `nick` que deve aceder ao elemento da página cujo id é `nickname` e recorra à propriedade `display = 'block'`. Além disso, no elemento com id `messageGameOver`, recorra à propriedade `innerHTML` para acrescentar ao que já existe nesse elemento, o texto `"<br>Parabéns! Entrou no TOP 10!"`.

c. Confirme o comportamento no browser.

Parte III – Gravar dados no TOP 10

4> Para gravar o nickname e os dados do utilizador, implemente os seguintes passos:

a. Implemente a função `saveTop10` que deve implementar os seguintes passos:

- Criar um objeto com o nickname e pontuação. Para obter o nickname, deve aceder ao valor do elemento com id `inputNick`.
- Verificar se o jogador já se encontra na lista do topGamers e se sim, atualizar a sua pontuação caso seja inferior à pontuação obtida no jogo. Se não existir, adicionar o elemento no fim do array. DICA: Use o método **map**
- Ordenar o array da seguinte forma:

```
topGamers.sort(function (a, b) { return b.points - a.points });
```

- Se existirem mais de 10 elementos, elimine o último.

b. Implemente o *eventListener* que invoque a função anterior, quando existir um click no botão cujo id é `okTop`. Para além de invocar a função `saveTop10`, também deve fechar a janela modal (display=none) e além disso, invocar a função `reset`.

c. Confirme o comportamento no browser e verifique na consola se não existe nenhum erro.

Parte IV – Gravar e Obter dados no LocalStorage

5> A propriedade *localStorage* permite armazenar/aceder a dados locais, persistindo os dados ainda que se feche a janela do browser. Isto é, permite a gravação de dados no cliente. O exemplo abaixo apresenta como se pode armazenar e aceder a dados simples, ou objetos.

```
localStorage.setItem('meuDado', '1234');
localStorage.getItem('meuDado')

localStorage.setItem('meuObjecto',
JSON.stringify({prop:'1',prop:'2'}));
meuObjecto = JSON.parse(meuObjecto)
```

No contexto da ficha, embora na realidade não aconteça e não faz sentido armazenar o TOP10 localmente, pretende-se exercitar esta propriedade.

6> Assim, implemente os seguintes passos:

a. Na função `saveTop10,` armazene o nickname e também o top10.

b. Implemente a função `getLocalStorage` que deverá obter os dados armazenados com o localStorage, nomeadamente o nickname e a lista do TOP10.

c. Por forma a que a função anterior seja executada apenas uma vez, transforme-a numa uma IIFE (*Immediately Invoked Function Expression*) como apresentado abaixo:

```
(function(){
    //...
})();
```

d. **Verifique no browser o comportamento o jogo, que deverá estar completo.**