Diego Fernandes Martinez - 2350947

- Usar um Style Guide - apresentar o uso de pelo menos 10 regras do style escolhido (sugerido AirBnb)

Foi usado AirBnb.

Regras:

1. Use const for all of your references; avoid using var.

Todas as referências agora são declaradas com const;

2. Use single quotes '' for strings.

Usada para todas as strings que não são templates;

1 & 2:

```
6       const namePrompt = window.prompt('Qual é o seu nome?');
```

3. When programmatically building up strings, use template strings instead of concatenation.

```
9           alert(`Bem vindo(a) ${namePrompt}`);
```

Usada para todas as string programadas;

4. Functions with multiline signatures, or invocations, should be indented just like every other multiline list in this guide: with each item on a line by itself, with a trailing comma on the last item.

```
24  // getElementById & charCode
25  function verify(event) {
26      // x declarado como var na 1ª avaliação
27      const x = event.charCode;
28
29      if (x === 13) {
30          // user declarado como var na 1ª avaliação
31          const user = document.getElementById('name').value;
32
33          // pss declarado como var na 1ª avaliação
34          const pss = document.getElementById('password').value;
35
36          if (pss === '12345') {
37              alert(`Bem vindo(a) de volta ${user}`);
38          }
39      }
40  }
```

Funções indentadas como no guia;

5. Use === and !== over == and !=.

```
29      if (x === 13) {
```

```
8      if (namePrompt !== null && namePrompt !== '') {
```

Operadores usados;

6. Use shortcuts for booleans, but explicit comparisons for strings and numbers.

```
20          if (c) clearInterval(x);
```

```
29      if (x === 13) {
```

Atalho usado;

7. Use braces with all multiline blocks. Brackets usados;
8. Use // for single line comments. Place single line comments on a newline above the subject of the comment. Put an empty line before the comment unless it's on the first line of a block. Feito;
9. Start all comments with a space to make it easier to read. Feito;
10. Place 1 space before the leading brace. Feito;

7, 8, 9 & 10:

```
24  // getElementById & charCode
25  function verify(event) {
26      // x declarado como var na 1ª avaliação
27      const x = event.charCode;
28
29      if (x === 13) {
30          // user declarado como var na 1ª avaliação
31          const user = document.getElementById('name').value;
32
33          // pss declarado como var na 1ª avaliação
34          const pss = document.getElementById('password').value;
35
36          if (pss === '12345') {
37              alert(`Bem vindo(a) de volta ${user}`);
38          }
39      }
40  }
```

- Usar um lint - mostrar a correção de pelo menos 5 problemas informados pelo lint (sugerido JSHint - usar o arquivo .jshintrc disponível no moodle)

Foi usado JSHint.

Problemas corrigidos:

1. Missing "use strict" statement. (E007)

```
2   'use strict';
```
"use strict" inserido na 2ª linha;

2. 'namePrompt' is not defined. (W117)

```
6       const namePrompt = window.prompt('Qual é o seu nome?');
```
namePrompt declarado como constante;

3. Expected '!==' and instead saw '!='. (W116)

```
7       if (namePrompt !== null && namePrompt !== '') {
```
'!=' usados substituídos por '!==';

4. Line is too long. (W101)

```
15      const c = window.confirm('Para continuar a navegar neste site deve aceitar os cookies 🍪');
16      //c declarado com var na 1ª avaliação
```
Comentário movido para a próxima linha;

5. Missing semicolon. (W033)

```
19                  clearInterval(x);
```
Ponto e vírgula inserido;

- Usar strict mode

Strict mode faz mudanças na semântica do JavaScript:

1. Troca os erros silenciosos do JS por lançar erros.
2. Pode ser melhor otimizado em algumas engines do que código non strict.
3. Proíbe o uso de sintaxe que provavelmente será definida em futuras versões do JS.

Strict mode pode ser usado para um script inteiro se declarado antes de qualquer outra declaração no script, ou em funções individuais se declarado no início da função.

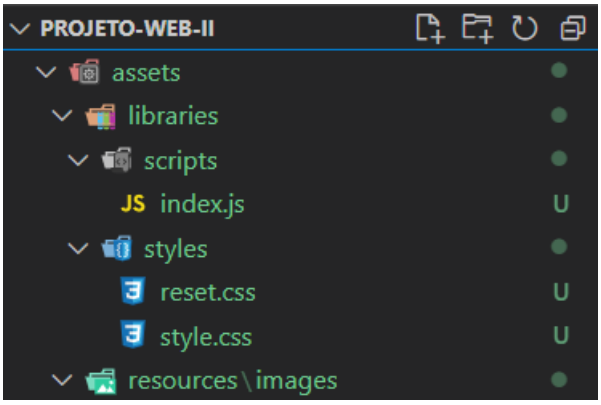
Foi usado nesse script para o script inteiro.

```
2   'use strict';
```

- Usar pasta assets e subpastas resources e libraries para organizar o código

É usado para melhor organização dos arquivos de um projeto.

Usado neste projeto.



- Usar let ou const ao invés de var

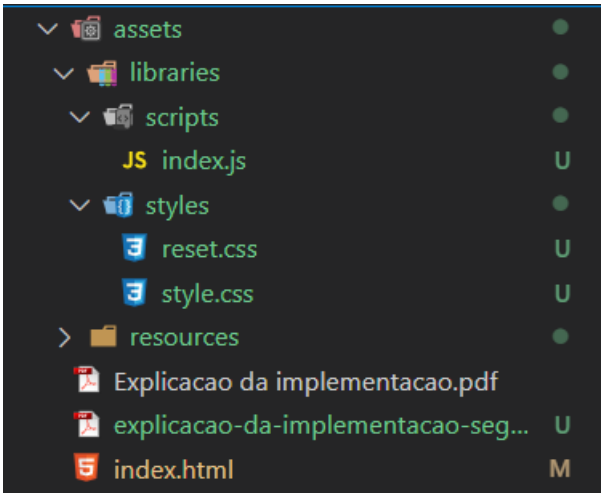
var é o jeito antigo de declarar variáveis e é mantido por compatibilidade. let é mais apropriado por se comportar como muitos esperam que o var se comporte. const deve ser usado caso a variável não for mudada.

Nesse script foi usado const em todas as variáveis.

- Nomes de arquivos minúsculos e separados por hífen (dashed-case)

Usado para melhor organização dos arquivos e uso no código.

Neste projeto foram usados nomes de arquivos em minúsculo, e com hífen no pdf da 2ª avaliação.



- Ler e escrever em elementos input com a propriedade value

A propriedade value define ou retorna o valor em um campo de texto.

Foi usado para escrever o nome dos campos antes do mouse focar no campo, e também para validar a senha no JS e imprimir o nome do usuário.

```
onfocus="this.value=''" value="Usuário">
' onfocus="this.value=''" value="Senha (Dica: 1 a 5)"
```

- Alterar o conteúdo de elementos div ou p com a propriedade innerHTML ou textContent

A propriedade innerHTML define ou retorna o HTML contido em um elemento.

Foi usado no botão dropdown de categorias.

```
47              categories.innerHTML(
48                  `<button>
49                      <p>Categorias</p>
50                  </button>
51                  <div class="dropdown-content">
52                      <p>Ação</p>
53                      <p>Aventura</p>
54                      <p>RPG</p>
55                      <p>Estratégia</p>
56                      <p>Plataforma</p>
57                  </div>`
58              );
```