



Instruções Gerais:

- Apenas discentes presentes na aula poderão entregar.
- Entregar a solução completa em formato ZIP pela tarefa do SIGAA.
- **Horário de Entrega definido na Tarefa do SIGAA**
- Formato de Entrega: Coloque a solução completa (arquivos HTML/CSS/JS) em um pasta chamada “prova1”, em seguida compacte esse diretório em formato “zip” (“**prova1.zip**”). A extensão “.zip” é obrigatória.
- Permitida consulta apenas aos documentos com prefixo “CONSULTA” que se encontra no SIGAA na aula da Prova.
- Distribuição da nota: HTML – 15%, CSS - 25%, JS - 60%.

Tarefa Única (10 pontos)

A tarefa consiste no desenvolvimento de um simulador de carrinhos de compras de um e-commerce. O(a) usuário(a) deverá inserir o nome do produto e pressionar “Adicionar”. Após adicionado, cada produto terá um botão para aumentar a quantidade e outro para remover a quantidade, ambos em passos de uma unidade. Após inserido no carrinho, a lista de produtos é atualizada para mostrar o novo produto inserido e a quantidade. Se a quantidade de um produto atingir zero (0), este será removido da lista. A Figura 1 apresenta a página inicial do carrinho.

Carrinho de Compras Mágico

Nome

Meus Produtos

Feito com ♥ por [Minha Página](#)

Figura 1

Inicialmente na área “Meus Produtos” não há nenhum item. Há também um rodapé com os dizeres “Feito com ♥ por [link para sua página](#) `target="_blank"` >Minha Página”. No campo “link para sua página”, coloque um link que te identifique de alguma forma profissional como: GitHub, LinkedIn, Lattes, Página Pessoal, Projeto de Extensão etc.

Descrição do funcionamento:

1 - O(a) usuário(a) precisa inserir um “Nome” antes de clicar no botão “Adicionar”. Caso nenhum nome seja inserido, a aplicação irá informar “**Nome do produto não pode ser vazio**” (ou outra frase de sua preferência que leve ao mesmo significado) através de um *alert()* como mostrado na Figura 2. A aplicação não avança enquanto não houver um nome válido.

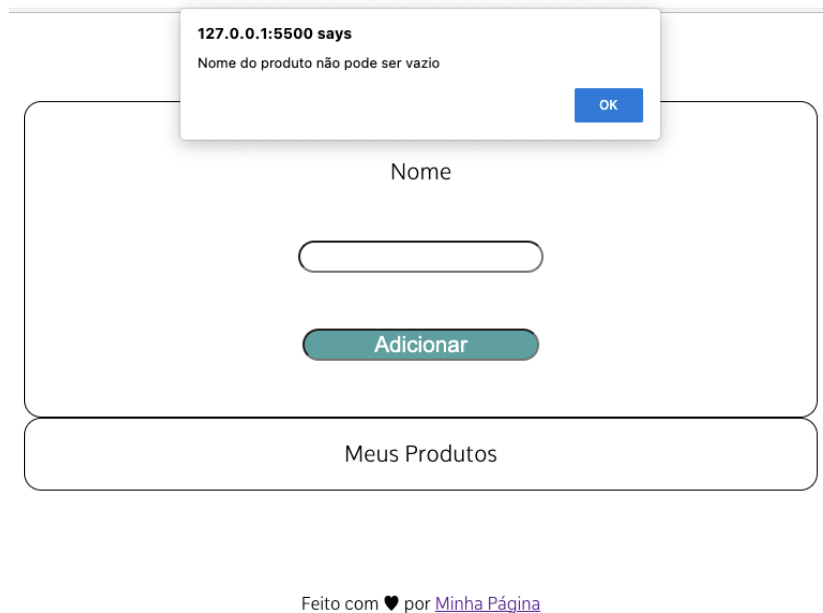


Figura 2

2 - Após inserido um nome válido a aplicação irá adicionar esse produto na lista após o(a) usuário(a) clicar no botão “Adicionar”. conforme mostrado na Figura 3. Ao adicionar um produto, por padrão, a quantidade será “1” para cada. No exemplo adicionamos Playstation 5, Alienware, e Porta Copos Gamers. A lista é gerada dinamicamente conforme se adicionam produtos.



Figura 3

Cada produto é gerado com seu nome e a quantidade padrão que é “1”. Tem-se também dois botões que permitem modificar a quantidade de cada produto. Para a avaliação do CSS não será considerado o rigor do espaçamento horizontal entre os elementos da lista, basta que se encontrem na mesma linha.

3 – Ao pressionar o botão “+” ou “-” a quantidade do produto é modificada. Como exemplo, a Figura 4 mostra a lista atualizada com 3 (três) Playstation 5 e 3 (três) Porta Copos Gamer.



Figura 4

4 – Se um elemento atingir a quantidade 0 (zero) este é removido da lista. A Figura 5 apresenta a quantidade atualizada de 2 (dois) Playstation 5 e o Porta Copos Gamer foi removido pois atingiu a quantidade 0 (zero).



Figura 5

Informações do CSS:

1 - Para replicar a fonte, use o seguinte comando, dentro da *tag* head (para baixar a Fonte)

```
<link rel="preconnect" href="https://fonts.googleapis.com">  
<link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>  
<link href="https://fonts.googleapis.com/css2?family=Comme:wght@200&display=swap" rel="stylesheet">
```

Para modificar a fonte utilize: `font-family: 'Comme', sans-serif;`

2 – Os discentes estão livres para implementarem/organizarem o CSS da forma que acharem melhor para atender ao apresentando nas Figuras 1-5. Não é obrigatório acertar de forma precisa medidas, cores etc. O objetivo é replicar o posicionamento e algumas proporções.

3 – Não é permitido utilizar nenhuma biblioteca para implementar o CSS, tais como *bootstrap*, *tailwind* e outros.

4 – É mandatório fazer a estilização em arquivo externo. **Estilização interna ou *inline* será desconsiderada.**

5 – É permitido fazer estilização dinâmica com JavaScript.

Informações do HTML:

1 – As imagens necessárias se encontram anexada a tarefa.

3 – Observe as Figuras 1-5 para identificar quais elementos se encontram na tela. Botões? Entradas? Textos? Imagens? Rótulos? Títulos? Sub-títulos?

Informações do JavaScript:

1 – Utilizar código JavaScript puro, sem auxílio de nenhuma biblioteca externa.

Informações Gerais:

1 - Permitida consulta apenas aos materiais com prefixo “CONSULTA” que se encontram no SIGAA.

2 – Entrega em formato “.zip”, com os três arquivos fontes (HTML/CSS/JS).

Dicas:

1. O “bullet point” pode ser representado pelo entity code “•”
2. Utilize `console.log()` para ver no terminal “strings” ou elementos HTML em formato literal
3. Utilize `console.dir()` para ver as propriedades dos elementos HTML em formato objeto (lembre-se da aula sobre DOM). Analise as propriedades.
4. Quando um evento é disparado, o elemento pode ser recuperado e suas propriedades podem ser exploradas. Veja o código abaixo:

```
const exemploFunc = (e) => {  
    console.log(e.target); //Elemento literal que disparou o evento  
    console.dir(e.target); //Elemento em formato objeto. É possível ver todas as propriedades  
}
```
5. Lembre-se que o DOM é uma árvore, e é possível navegar por essa a partir de qualquer elemento HTML e encontrar um ancestral (*parentNode*) ou descendente (*children*). Algumas propriedades são *arrays*. Explore o `console.dir()` para visualizar as propriedades que você pode manipular.
6. Explore o *inspector* no navegador para visualizar os elementos e informações do CSS
7. Use agrupadores para ajudar na organização.
8. Desenhe bordas para visualizar os elementos com mais facilidade durante a estilização.
9. A propriedade “value” permite recuperar o valor do `<input>` e `<select>`

10. Utilize `Math.Random()` para geração de números aleatórios.
11. CSS Dica Extra: Considerando um contêiner flex, se deseja que o espaçamento entre os elementos tenha um alinhamento mais estável use o comando “flex: 1 1 0” ou “flex: 1” nos descendentes do contêiner flex.
12. Use a *tag* `<link>`, dentro do `<head>`, para conectar a página HTML ao CSS externo.
13. Use a *tag* `<script>`, antes de finalizar o `</body>`, para conectar a página HTML ao JavaScript externo.

Boa prova para todos 😊

