



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 selecionar a quantidade através de dois botões, um para aumentar a quantidade e outro para diminuir. Após inserido no carrinho, a lista de produtos é atualizada para mostrar o novo produto inserido e a quantidade. Ao final temos também a quantidade total de produtos (não de itens por produto). A Figura 1 apresenta a página inicial do carrinho, com navegador em 100% de Zoom.

Carrinho de Compras Mágico

Nome

+

1

-

Adicionar

Meus Produtos

Total de Produtos: 0

Feito com ♥ por [Minha Página](#)

Figura 1

Inicialmente na área “Meus Produtos” não há nenhum item. Há 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 os valores “Nome” e definir a quantidade manipulando os botões “+” (verde) e “-” (vermelho). 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.

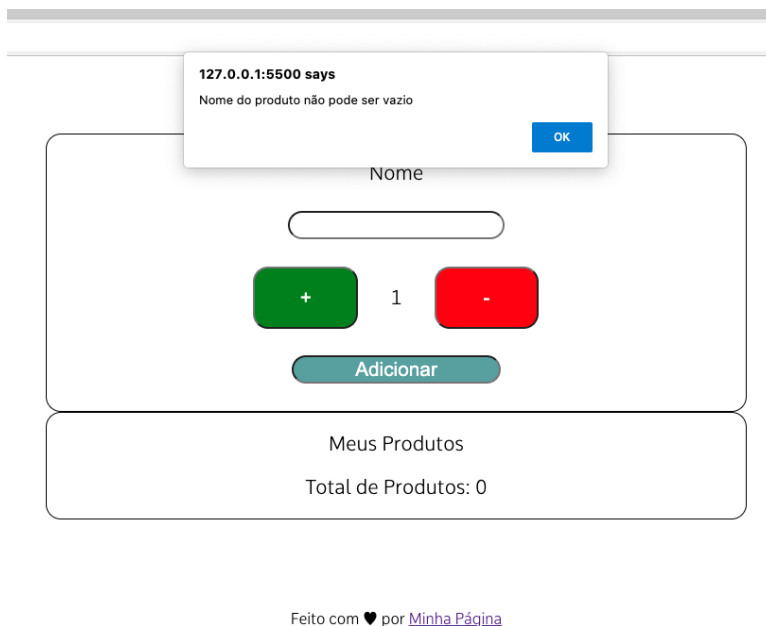


Figura 2

2 - Após inserido um nome válido e definida a quantidade (o padrão é 1) a aplicação irá adicionar esse produto na lista após o(a) usuário(a) clicar no botão “Adicionar” conforme mostrado na Figura 3. No exemplo adicionamos 2 (dois) Playstation 5, 1 (um) Alienware, e 3 (três) Porta Copos Gamers. A lista é gerada dinamicamente conforme se adicionam produtos. Observe também que a última linha mostrou “Total de Produtos: 3”. Isto é, adicionamos 3 (três) produtos diferentes e não a soma total dos itens, que seriam 6 (seis) itens.



Figura 3

3 – A lista criada permite que o produto seja removido. Todo elemento é criado com um botão com os dizeres “X” anexado. Ao apertar esse botão o elemento é removido da lista. A Figura 4 mostra a lista atualizada ao se remover o Playstation 5. O elemento inteiro é removido. Observe que a quantidade “Total de Produtos” foi atualizada para 2 (dois).

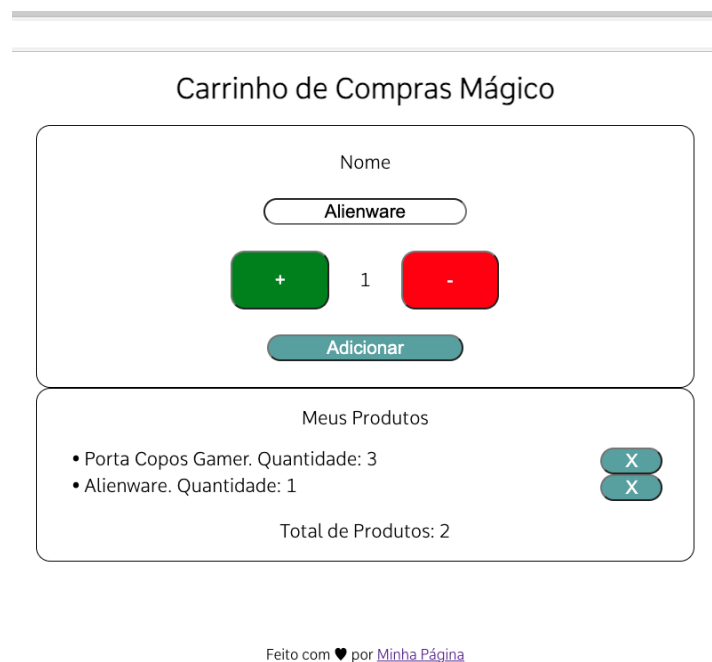


Figura 4

4 – Ao adicionar um novo produto, a quantidade padrão retorna para 1 (um). Isto é, ao se adicionar um novo produto, a quantidade do próximo é reiniciada para que o(a) usuário(a) possa definir a nova quantidade para o novo produto.

5 – Ao pressionar o botão “-” a quantidade do produto não pode ser inferior a 1 (um). Mesmo que continue pressionando o botão “-” o valor se manterá em 1 (um).

Informações do CSS:

1. Para replicarem a fonte, usem 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">
```

2. Para modificar a fonte utilize: **font-family: 'Comme', sans-serif;**
3. Os discentes estão livres para implementarem/organizarem o CSS da forma que acharem melhor para atender ao apresentando nas Figuras 1-4. Não é obrigatório acertar de forma precisa medidas, cores etc. O objetivo é replicar o posicionamento e algumas proporções.
4. Não é permitido utilizar nenhuma biblioteca para implementar o CSS, tais como *bootstrap*, *bulma* e outros.
5. É mandatório fazer a estilização em arquivo externo. **Estilização interna ou inline será desconsiderada.**
6. Estilização dinâmica pelo JavaScript também pode ser feita.

Informações do HTML:

1. As imagens necessárias se encontram anexada a tarefa.
2. Observe as Figuras 1-4 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. Quando um botão é disparado por um evento, esse pode ser exposto e se explorar todas as propriedades dos elementos HTML envolvidos. 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  
}
```
3. 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.
4. Explore o *inspector* no navegador para visualizar os elementos e informações do CSS
5. Use agrupadores para ajudar na organização
6. Desenhe bordas para ver melhor os elementos durante a estilização.

Boa prova para todos 😊

