



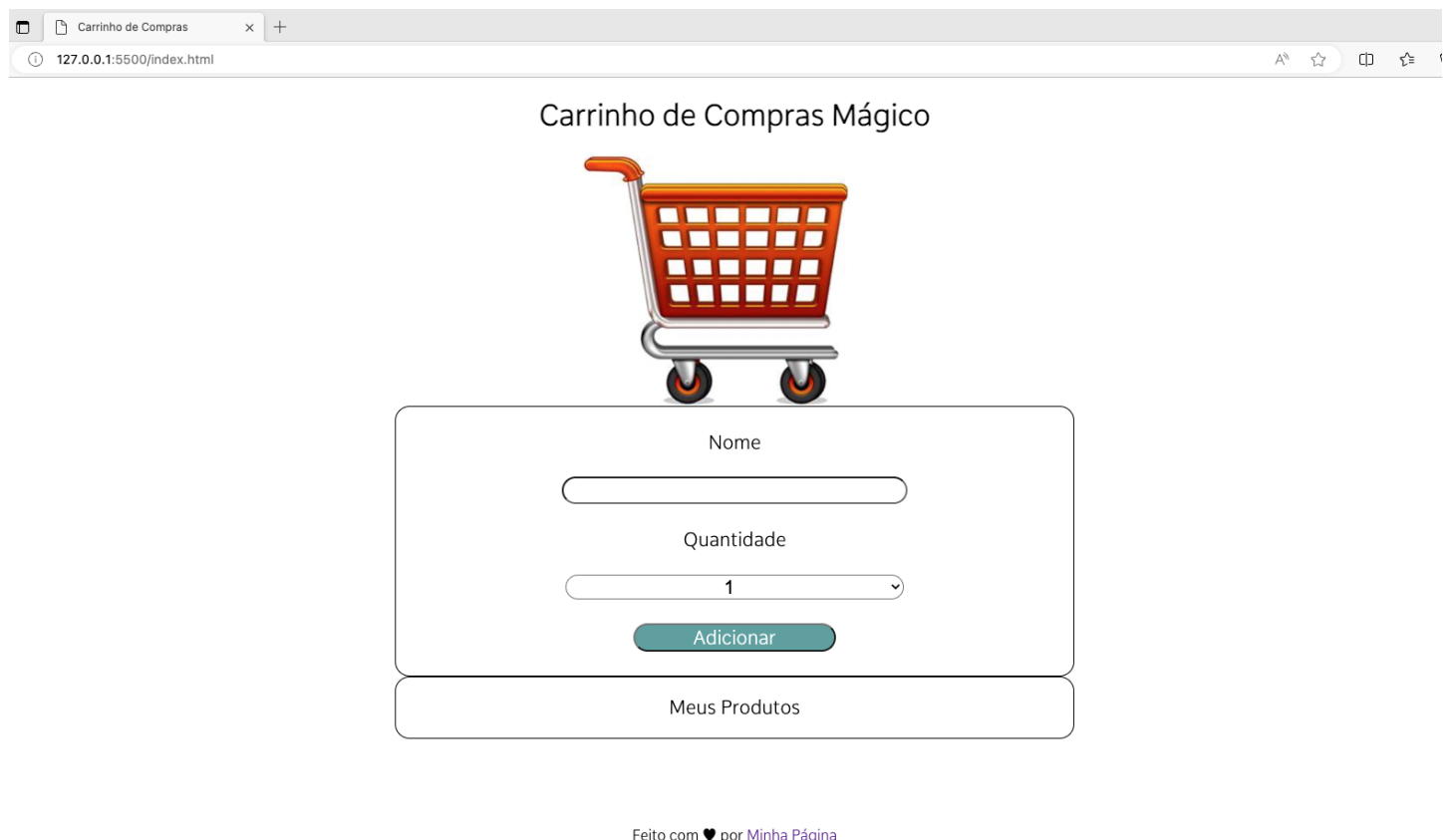
Instruções Gerais:

- 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, sendo o máximo 5 (cinco). Após inserido no carrinho, a lista de produtos é atualizada para mostrar o novo produto inserido e a quantidade.

Figura 1 apresenta a página inicial do carrinho. Nessa Figura tem-se também a barra do navegador, para auxiliar nas proporções.



A imagem mostra a interface de um navegador web com uma aba intitulada "Carrinho de Compras". O endereço da página é "127.0.0.1:5500/index.html". O título da página é "Carrinho de Compras Mágico".

Centralizado na tela há um ícone de um carrinho de compras laranja e prateado. Abaixo dele, há um formulário com os seguintes campos:

- Um campo de texto rotulado "Nome".
- Um campo de texto rotulado "Quantidade" com o valor "1" selecionado em uma lista suspensa.
- Um botão verde rotulado "Adicionar".

Abaixo do formulário, há uma seção rotulada "Meus Produtos".

Na base da página, há um rodapé que diz: "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 [>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.](link para sua página)

Descrição do funcionamento:

1 - O(a) usuário(a) precisa inserir os valores “Nome” e selecionar uma “Quantidade” (1 até 5) 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.

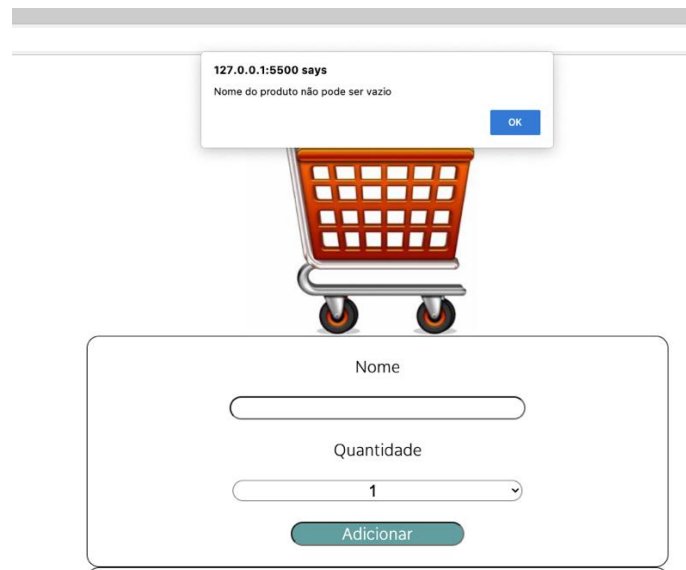


Figura 2

2 - Após inserido um nome válido e selecionada a quantidade (1 é o valor padrão) 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.

Carrinho de Compras Mágico



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. Observe que há um alinhamento nos botões, e que esse possui uma largura menor que o botão “Adicionar”. A Figura 4 mostra a lista atualizada ao se remover o Playstation 5. O elemento inteiro é removido.

Carrinho de Compras Mágico



Nome

Porta Copos Gamer

Quantidade

3

Adicionar

Meus Produtos

- Alienware. Quantidade: 1
- Porta Copos Gamer. Quantidade: 3

Feito com ♥ por [Minha Página](#)

Figura 4

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">
```

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 1-4. 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*, *bulma* e outros.

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

Informações do HTML:

1 – As imagens necessárias se encontram anexada a tarefa.

3 – 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 entity code “bull;” gera bullet points
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 inspetor 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.
7. A propriedade “value” permite recuperar valor do `<input>`
8. Utilize `Math.Random()` para geração de números aleatórios.