

AULA 1 – PROJETO MÃO NA MASSA (CARRINHO DE COMPRAS)

OBJETIVO DA AULA

Compreender como utilizar um *template* pronto.

APRESENTAÇÃO

Nesta aula vamos dar o nosso primeiro passo para a construção de um pequeno projeto. A ideia aqui é construirmos juntos um carrinho de compras usando HTML, CSS, JavaScript, ou seja, todos os assuntos abordados durante a nossa disciplina. Além disso, vamos usar também o *Bootstrap*, pois faremos uso de um *template* pronto para facilitar a nossa vida. *Bootstrap* nada mais é que um *framework front-end* cujo objetivo é facilitar a criação de sites e/ou aplicações responsivas de forma rápida e fácil. Nosso *template* será este:



My T-shirt

LOGIN

HELP

SHOPPING BAG

2 items

PRODUCTS	PRICE	QUANTITY	TOTAL	
 <div>COTTON T-SHIRT Art.No: 091091001 Color: White Size: M</div>	\$9.99	<div>- 2 +</div>	\$19.98	×
 <div>WHITE T-SHIRT Art.No: 056289891 Color: White Size: XL</div>	\$20.9	<div>- 2 +</div>	\$41.8	×

GO BACK


SUBTOTAL \$61.78

CONTINUE

Baseado neste *template*, faremos as modificações necessárias para personalizar o nosso carrinho, tendo como fonte de inspiração a lojinha da Gran Cursos.

1. INTRODUÇÃO

Nosso carrinho ficará assim:






Loja

Gran

[ENTRAR](#)
[AJUDA](#)

LOJINHA DA GRANCURSOS

3 itens

PRODUTO	PREÇO	QUANTIDADE	TOTAL	
<div>  <div> <div>Caneca Eu Dou Conta</div> <div>Art.No: 091091001</div> <div>Color: Branco</div> </div> </div>	R\$45,00	<div> <div>-</div> <div>2</div> <div>+</div> </div>	R\$90,00	×
<div>  <div> <div>Moleskine</div> <div>Art.No: 056289891</div> <div>Color: Vermelho</div> </div> </div>	R\$39,90	<div> <div>-</div> <div>1</div> <div>+</div> </div>	R\$39,90	×
<div>  <div> <div>Caneca Mágica</div> <div>Art.No: 091091001</div> <div>Color: Bege</div> </div> </div>	R\$49,90	<div> <div>-</div> <div>1</div> <div>+</div> </div>	R\$49,90	×

VOLTAR

SUBTOTAL R\$179,80

CONTINUAR

Para isso, temos que seguir os seguintes passos:

- Criar um documento HTML chamado index.html;
- Criar um documento CSS chamado carrinho.css;
- Criar um documento JavaScript chamado carrinho.js;
- Fazer uso de um template pronto e copiar os códigos de HTML e CSS;
- Colar os códigos de HTML dentro do index.html;
- Colar os códigos de CSS dentro do carrinho.css;
- Estabelecer os links com os documentos de CSS e Javascript;
- Estabelecer os links com CDN's utilizados pelo *Bootstrap*;
- Programar o carrinho de compras;

Utilizarei o VSCode como IDE, mas você pode escolher a que quiser. Na *Unidade 1* deixei uma lista das IDE's mais utilizadas. O *template* que utilizaremos está disponível em: <https://bbbootstrap.com/snippets/bootstrap-ecommerce-cart-page-template-91851305#>. Neste link você encontrará o *preview* do *template* que vamos utilizar, o código de HTML e CSS, além dos recursos necessários para fazer o carrinho funcionar.

CONSIDERAÇÕES FINAIS

Nesta aula vimos como será o passo a passo para construirmos o nosso carrinho de compras. O próximo passo é criar o nosso documento `index.html`, que receberá o código que copiaremos do site do nosso *template*. Deixo em materiais complementares diversos links de como fazer um carrinho de compras. Veja que existem formas diferentes de se fazer. Espero que você explore e teste todos os exemplos disponibilizados aqui.

MATERIAIS COMPLEMENTARES

Carrinho de compras: <https://github.com/williandutras/carrinho-de-compras/blob/master/index.html>.

Carrinho de compras com Javascript – SimpleCart: https://odesenvolvedor.com.br/carrinho-de-compras-com-javascript-simplecart_1721.html.

Calculando o total de um carrinho de compras com Javascript/Jquery: <https://nxh101.wordpress.com/2015/08/15/calculando-o-total-de-um-carrinho-de-compras-com-javascriptjquery/>.

Criando um Carrinho de Compras (Vídeo Completo): <https://youtu.be/byk3xJof8kY>.

REFERÊNCIAS

FERREIRA, Elcio; EIS, Diego *HTML5 Curso W3C Escritório Brasil*. 2015. Disponível em: <<https://www.w3c.br/pub/Cursos/CursoHTML5/html5-web.pdf>>. Acesso em: 26 out. 2022.

AULA 2 – MÃO NA MASSA – HTML

OBJETIVO DA AULA

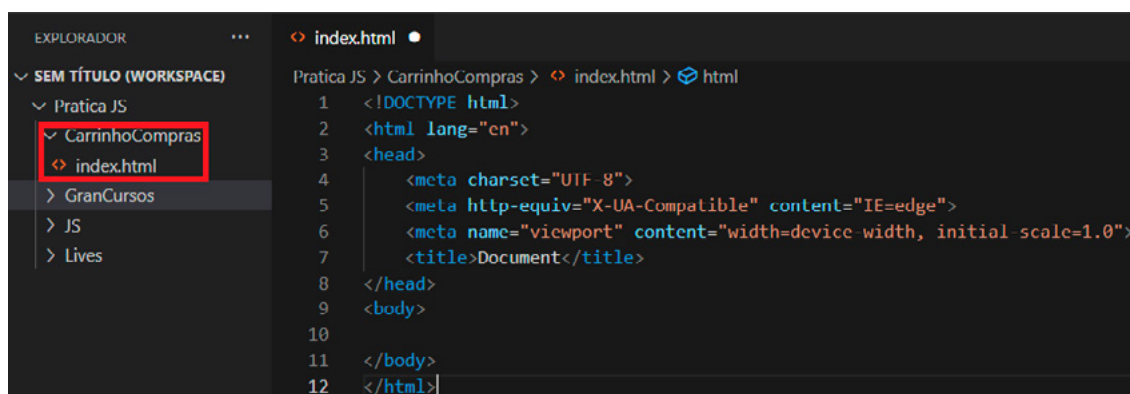
Aplicar os conceitos de HTML no projeto.

APRESENTAÇÃO

Relembrando, HTML (*Hypertext Markup Language*) é uma linguagem de marcação de hipertexto utilizada para construção de páginas web. Segundo Ferreira e Eis (2015), um hipertexto nada mais é do que um conjunto de elementos (palavras, imagens, vídeos, áudio, documentos etc.) ligados por conexões. Juntos, estes elementos formam uma grande rede de informação. “Para distribuir informação de uma maneira global, é necessário haver uma linguagem entendida universalmente por diversos meios de acesso. O HTML se propõe a ser esta linguagem” (FERREIRA e EIS, 2015). Nesta aula criaremos o nosso index.html e faremos as modificações necessárias para personalizar nosso carrinho de compras. Vamos lá!

1. HTML

1º passo: criar o documento HTML com sua estrutura básica.



Repare que criei um diretório chamado CarrinhoCompras e coloquei o index.html dentro deste diretório.

2º passo: colar dentro do <body> o código de HTML do nosso *template*.



Fonte: <https://bbbootstrap.com/snippets/bootstrap-ecommerce-cart-page-template-91851305#>.

```

53         </div>
54     </div>
55     <div class="pl-md-0 pl-1"><b>$20.9</b></div>
56     <div class="pl-md-0 pl-2">
57         <span class="fa fa-minus-square text-secondary"></span><span class="px-md-3 px-1">2</span><span class="fa
58             fa-plus-square text-secondary"></span>
59     </div>
60     <div class="pl-md-0 pl-1"><b>$41.8</b></div>
61     <div class="close">&times;</div>
62 </div>
63 </div>
64 </div>
65 </div>
66 <div class="container bg-light rounded-bottom py-4" id="zero-pad">
67     <div class="row d-flex justify-content-center">
68         <div class="col-lg-10 col-12">
69             <div class="d-flex justify-content-between align-items-center">
70                 <div>
71                     <button class="btn btn-sm bg-light border border-dark">GO BACK</button>
72                 </div>
73                 <div class="px-md-0 px-1" id="footer-font">
74                     <b class="pl-md-4">SUBTOTAL</b><span class="pl-md-4">$61.78</span></div>
75             </div>

```

```

76         <div>
77             <button class="btn btn-sm bg-dark text-white px-lg-5 px-3">CONTINUE</button>
78         </div>
79     </div>
80 </div>
81 </div>
82 </div>
83 </body>
84 </html>

```

Acrescentamos o código na linha 10 e agora temos 84 linhas de código. Vou usar sempre as linhas como referência para que vocês possam fazer as modificações aí também. Bom, como o código já está pronto, só precisamos ajustá-lo: vamos traduzir os pequenos textos, incluir nossas imagens e “linkar” o HTML com os arquivos externos. Você deve ter reparado que toda linha de código começa com uma div e que todas elas têm uma classe (class). Só perceberemos a diferença que essas classes farão no visual da nossa página quando “linkarmos” com os arquivos externos.

3º passo: alterar o texto do <title> para **Carrinho de Compras**, linha 7.

```

7     <title>Carrinho de Compras</title>

```

4º passo: substituir o título <h4> com o texto **My T-shirt**

14 <div class="pt-1"><h4>My T-shirt</h4></div> pelo logotipo da lojinha Gran Cursos, linha 14. Para isso, vamos usar a tag passando o endereço da imagem.

```

14 <div></div>

```

Para seguir o padrão do código que copiamos, criamos uma <div>, dentro dela criamos a tag e passamos o endereço da imagem: <https://loja.grancursosonline.com.br/wp-content/themes/base-framework/assets/logo-loja-gran-no-background.png>. Além disso, alteramos a altura (height) e a largura (width) da imagem.

5º passo: trocar LOGIN e HELP por ENTRAR e AJUDA, linhas 15 e 16.

```

15 <div class="ml-auto p-2"><a href="#" class="text-dark text-decoration-none" id="mobile-font">ENTRAR</a></div>
16 <div class="p-2"><a href="#" class="text-dark text-decoration-none" id="mobile-font">AJUDA</a></div>

```

6º passo: alterar todo o texto do cabeçalho do nosso carrinho, linhas 19-26.

```

18 <div class="d-flex flex-column pt-4">
19     <div><h5 class="text-uppercase font-weight-normal">Lojinha da GranCursos</h5></div>
20     <div class="font weight normal">3 itens</div>
21 </div>
22 <div class="d-flex flex-row px-lg-5 mx-lg-5 mobile" id="heading">
23     <div class="px-lg-5 mr-lg-5" id="produc">PRODUTOS</div>
24     <div class="px-lg-5 ml-lg-5" id="prc">PREÇO</div>
25     <div class="px-lg-5 ml-lg-1" id="quantity">QUANTIDADE</div>
26     <div class="px-lg-5 ml-lg-3" id="total">TOTAL</div>
27 </div>

```

Mudamos para 3 itens porque vamos acrescentar um produto a mais no nosso carrinho.

7º passo: alterar a imagem e os dados do primeiro produto, linhas 30-42. Além disso vamos remover a linha (linha 35) que representa o tamanho do produto, pois nossa caneca não tem tamanho P-M-G. Este é o link da imagem do nosso primeiro produto: https://grancursos-blog-loja-prd.s3.amazonaws.com/wp-content/uploads/2022/10/01150955/900x900_9.jpg.

```

45 <div class="d-flex flex-row align-items-center">
46 <div></div>
47 <div class="d-flex flex-column pl-md-3 pl-1">
48 <div><h6>Moleskine</h6></div>
49 <div><span class="pl-2"><b>R$39,90</b></span></div>
50 <div><span class="pl-3">Vermelho</span></div>
51 <div><span class="pl-4">XL</span></div>
52 </div>
53 </div>
54 <div class="pl-md-0 pl-1"><b>R$39,90</b></div>
55 <div class="pl-md-0 pl-2">
56 <span class="fa fa-minus-square text-secondary"></span><span class="px-md-3 px-1">1</span><span class="fa
fa-plus-square text-secondary"></span>
57 </div>
58 <div class="pl-md-0 pl-1"><b>R$39,90</b></div>
59 <div class="close">&times;</div>
60 </div>

```

Por enquanto, os valores e a quantidade vão ficar estáticos, vamos torná-los dinâmicos com o JavaScript.

8º passo: alterar a imagem e os dados do segundo produto, linhas 46-58. Além disso, vamos remover a linha (linha 51) que representa o tamanho do produto, pois nossa agenda não tem tamanho P-M-G. Este é o link da imagem do nosso segundo produto: https://grancursos-blog-loja-prd.s3.amazonaws.com/wp-content/uploads/2022/10/01151137/900x900_11.jpg.

```

45 <div class="d-flex flex-row align-items-center">
46 <div></div>
47 <div class="d-flex flex-column pl-md-3 pl-1">
48 <div><h6>Moleskine</h6></div>
49 <div><span class="pl-2"><b>R$39,90</b></span></div>
50 <div><span class="pl-3">Vermelho</span></div>
51 <div><span class="pl-4">XL</span></div>
52 </div>
53 </div>
54 <div class="pl-md-0 pl-1"><b>R$39,90</b></div>
55 <div class="pl-md-0 pl-2">
56 <span class="fa fa-minus-square text-secondary"></span><span class="px-md-3 px-1">1</span><span class="fa
fa-plus-square text-secondary"></span>
57 </div>
58 <div class="pl-md-0 pl-1"><b>R$39,90</b></div>
59 <div class="close">&times;</div>
60 </div>

```

9º passo: acrescentar mais um produto, para isso, vamos copiar todo o código do produto 1 (linhas 29-42) e colar na linha 60. Feito isso, vamos alterar a imagem e os dados do terceiro produto. Este é o link da imagem do nosso terceiro produto: https://grancursos-blog-loja-prd.s3.amazonaws.com/wp-content/uploads/2022/10/01151653/900x900_19.jpg. Com este acréscimo, agora temos um total de 96 linhas de código.

```

60 <div class="d-flex flex-row justify-content-between align-items-center pt-lg-4 pt-2 pb-3 border-bottom mobile">
61   <div class="d-flex flex-row align-items-center">
62     <div></div>
63     <div class="d-flex flex-column pl-md-3 pl-1">
64       <div><h6>Caneca Mágica</h6></div>
65       <div>Código<span class="pl-2">091091001</span></div>
66       <div>Cor:<span class="pl-3">Bege</span></div>
67     </div>
68   </div>
69   <div class="pl-md-0 pl-1"><b>R$49,90</b></div>
70   <div class="pl-md-0 pl-2">
71     <span class="fa fa-minus-square text-secondary"></span><span class="px-md-3 px-1">1</span><span class="fa fa-plus-square text-secondary"></span>
72   </div>
73   <div class="pl-md-0 pl-1"><b>R$49,90</b></div>
74   <div class="close">&times;</div>
75 </div>

```

10º passo: para concluir, vamos alterar o texto dos botões que ficam na parte de baixo do carrinho, linhas 82-88.

```

81 <div>
82   <button class="btn btn-sm bg-light border border-dark">VOLTAR</button>
83 </div>
84 <div class="px-md-0 px-1 id="footer-font">
85   <b class="pl-md-4">SUBTOTAL</b><span class="pl-md-4">R$179,80</span></div>
86 </div>
87 <div>
88   <button class="btn btn-sm bg-dark text-white px-lg-5 px-3">CONTINUAR</button>
89 </div>

```

CONSIDERAÇÕES FINAIS

Nesta aula fizemos algumas alterações no nosso código de HTML para personalizar o carrinho com os produtos da lojinha da Gran Cursos. Inserimos imagens novas, alteramos o texto e excluímos algumas linhas que não vão nos servir. Neste momento, estamos trabalhando com valores estáticos, mas não se preocupe, na aula de JavaScript vamos tornar estes valores dinâmicos, ou seja, o usuário vai aumentar ou diminuir a quantidade e o subtotal será alterado automaticamente. Se você se perdeu no passo a passo descrito acima, pode acessar o nosso código neste link: <https://github.com/GRANCodigo/PraticaDeProgramacao/blob/1ed3e8e69489cb82c3c195f444fe66d3ced98f53/Unidade1Aula5a>. Após todas essas modificações sua página deve estar mais ou menos assim:



[ENTRAR](#)
[AJUDA](#)

Lojinha da GranCursos

3 itens
PRODUTOS
PREÇO
QUANTIDADE
TOTAL



Caneca Eu dou Conta

Código:091091001
Cor:Branco
RS45,00
2
RS90,00



Moleskine

Código:056289891
Cor:Vermelho
RS39,90
1
RS39,90
x



Caneca Mágica

Código:091091001
Cor:Bege
RS49,90
1

RS49,90
x

[VOLTAR](#)

SUBTOTALRS179,80

[CONTINUAR](#)

MATERIAIS COMPLEMENTARES

Criando uma Loja Virtual – Definindo o processo de compra e incluindo e removendo itens no carrinho – 5: https://www.macoratti.net/Cursos/webmatrix/wbm_ecm5.htm.

Curso de JS. Aula 06. Pizzaria da Nonna. Carrinho de compras: <https://www.youtube.com/watch?v=tczhRdyj1nM>.

REFERÊNCIAS

FERREIRA, Elcio; EIS, Diego *HTML5 Curso W3C Escritório Brasil*. 2015. Disponível em: <<https://www.w3c.br/pub/Cursos/CursoHTML5/html5-web.pdf>>. Acesso em: 26 out. 2022.

AULA 3 – MÃO NA MASSA – BOOTSTRAP

OBJETIVO DA AULA

Aprender a utilizar o Bootstrap.

APRESENTAÇÃO

Na aula anterior, alteramos todo o código de HTML para atender a nossa necessidade. Você deve ter percebido que todos os produtos estão agrupados do lado esquerdo da nossa página e que não possui nenhuma formatação. Nesta aula vamos aprender como vincular o HTML ao Bootstrap.

1. BOOTSTRAP

Antes de mais nada, vamos entender o que é o Bootstrap. Para deixar nossa página HTML com uma aparência legal temos que customizar cores, fontes, tamanhos, margens... e muita das vezes utilizamos o CSS para fazer isso, porém essa tarefa acaba se tornando muito trabalhosa. Sendo assim, vamos ver como estilizar usando Bootstrap. Como já foi dito anteriormente, Bootstrap é um *framework front-end* capaz de nos fornecer estruturas de CSS de forma simples e rápida. Um *framework* fornece uma série de códigos prontos para serem reutilizados por uma determinada tecnologia, tornando o desenvolvimento dos códigos mais padronizado e seguro, além de facilitar a vida dos desenvolvedores.

VOCÊ SABIA?

Originalmente, o Bootstrap foi desenvolvido para o Twitter por um grupo de desenvolvedores liderados por Mark Otto e Jacob Thornton Logo e se tornou uma das estruturas de *front-end* e projetos de código aberto mais populares do mundo.

Fonte: <https://www.alura.com.br/artigos/bootstrap#:~:text=Bootstrap%20%C3%A9%20um%20framework%20front,dispositivos%20m%C3%B3veis%20da%20mesma%20forma.>

O Bootstrap possui uma grande variedade de classes que podem ser aplicadas em diferentes elementos. Entenda mais sobre as classes neste site: <https://diegomariano.com/bootstrap/#:~:text=Bootstrap%20apresenta%20uma%20s%C3%A9rie%20classes,declaradas%20por%20meio%20das%20classes%20%E2%80%9C>

Falando em elementos, o Bootstrap oferece estilos aos elementos da página, ou seja:

(...) só de utilizar o framework, ele já aplicará uma estilização padrão aos elementos básicos. Assim, as fontes e seus tamanhos serão os mesmos em todas as páginas, as cores de links, a distância entre linhas em parágrafos e em muitas outras características de elementos (NOLETO, 2022).

Isso tudo é muito legal, mas como faço para aplicar o Bootstrap na minha página? O Bootstrap pode ser instalado e configurado de diferentes formas, veja: <https://blog.betrybe.com/bootstrap/>. No nosso caso, vamos instalá-lo via CDN (*Content Delivery Network*), ou seja, vamos utilizar os links CDN's disponíveis no nosso *template*.

PREVIEW	HTML	CSS	JAVASCRIPT	RESOURCES
CDN				ACTION
		https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css		Copy CDN
		https://stackpath.bootstrapcdn.com/bootstrap/4.0.0/js/bootstrap.bundle.min.js		Copy CDN
		https://cdnjs.cloudflare.com/ajax/libs/jquery/3.2.1/jquery.min.js		Copy CDN
		https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-awesome.min.css		Copy CDN

1.1. O QUE É UM CDN?

CDN significa Rede de Distribuição de Conteúdo, ou seja:

(...) é uma rede de servidores que armazenam réplicas do conteúdo de outros sites na memória (*cache*) e depois os entrega aos visitantes, baseando-se na localização geográfica para conectá-los ao servidor mais próximo e mais rápido, reduzindo o tempo de transferência dos dados (latência) (FRANCO, 2020).

Como pode ser visto na imagem acima, o Bootstrap tem CDN para armazenamento de arquivos CSS e JavaScript. No CDN que corresponde ao.css temos as classes que serão utilizadas para estilizar os elementos básicos de HTML além das configurações necessários para criar layouts responsivos. Como alguns recursos e componentes (botões, janelas modais, mensagens de alerta...) do Bootstrap necessitam da execução de funções do JavaScript, o uso do CDN referente ao.js se faz necessário.

Agora que já sabemos os conceitos básicos e como o Bootstrap funciona, vamos vinculá-lo ao nosso projeto. É bem simples! Vamos lá...

1º passo: na página do nosso *template* temos os links dos recursos de CDN's necessários para estilizar o nosso carrinho, vamos inclui-lo na área comportamental do nosso código <head> através das tags <link> e <script>.


```

1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta http-equiv="X-UA-Compatible" content="If=edge">
6   <meta name="viewport" content="width=device-width, initial-scale=1.0">
7   <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css">
8   <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-awesome.min.css">
9   <script src="https://stackpath.bootstrapcdn.com/bootstrap/4.0.0/js/bootstrap.bundle.min.js"></script>
10  <script src="https://cdnjs.cloudflare.com/ajax/libs/jquery/3.2.1/jquery.min.js"></script>
11 </head>
12 <title>Carrinho de Compras</title>

```




Repare que para vincular os CDN's de.css usamos a tag <link> e para vincular os CDN's de.js usamos a tag <script>.

2º passo: conferir se agora o *template* foi aplicado. Neste momento nosso carrinho deve estar assim:


ENTRAR AJUDA

LOJINHA DA GRANCURSOS

3 itens

	PRODUTOS	PREÇO	QUANTIDADE	TOTAL	
	Caneca Eu dou Conta Código: 091091001 Cor: Branco	R\$45,00	- 2 +	R\$90,00	×
	Moleskine Código: 056289891 Cor: Vermelho	R\$39,90	- 1 +	R\$39,90	×
	Caneca Mágica Código: 091091001 Cor: Bege	R\$49,90	- 1 +	R\$49,90	×

VOLTAR
SUBTOTAL R\$179,80
CONTINUAR

Muito diferente da nossa primeira versão, não é mesmo? Antes tínhamos todos os produtos agrupados do lado esquerdo da página. Ao inserir os CDN's no código, o *layout* do *template* foi aplicado. Se você se perdeu no passo a passo descrito acima, pode acessar o nosso código neste link: <https://github.com/GRANCodigo/PraticaDeProgramacao/blob/4ab6a3d20771511591c439fd6a0e544eb524a46d/Unidade4Aula1b>.

CONSIDERAÇÕES FINAIS

Nesta aula, vimos que o Bootstrap ajuda muito na padronização e manutenção do seu código, reduzindo consideravelmente o tempo dedicado ao desenvolvimento. Porém, alguns desenvolvedores não gostam de utilizar o *framework* pelo fato de ter muitas classes e divs para estilizar os elementos, deixando o código poluído e muito grande. É uma questão de gosto mesmo, até porque existem diversos outros frameworks disponíveis (*Vue.js*, *React*, *Angular*), você pode optar por utilizar ou não, lembrando que cada um deles possui suas vantagens e desvantagens.

MATERIAIS COMPLEMENTARES

Site oficial do Bootstrap: <https://getbootstrap.com/>.

REFERÊNCIAS

FRANCO, Felipe. *2 maneiras simples de instalar o Bootstrap em um projeto Angular*. 2020. Disponível em: <https://www.fabricadecodigo.com/bootstrap-instalacao-angular/>. Acesso em 28 nov. 2022.

NOLETO, Cairo. *Bootstrap: o que é, como usar e para que serve esse framework*. 2022. Disponível em: <https://blog.betrybe.com/bootstrap/>. Acesso em: 28 nov. 2022.

AULA 4 – MÃO NA MASSA – CSS

OBJETIVO DA AULA

Aplicar os conceitos de CSS no projeto.

APRESENTAÇÃO

Nesta aula, vamos explorar o `o.css` disponível no nosso *template*. A Folha de Estilos em Cascata (CSS) é uma linguagem baseada em regras utilizada para aplicar estilos (cor de fundo, borda, cor de texto, tamanho da fonte...) ao seu documento. As características visuais aplicadas aos elementos são representadas pelo que chamamos de **propriedades** e essas propriedades podem ser combinadas, trazendo possibilidades de personalização quase infinitas.

1. CSS

Na aula anterior, vinculamos o Bootstrap através de CDN's, lembra? Sendo assim, toda vez que a página for carregada, o servidor mais próximo da localidade da sua localidade se encarrega de fornecer o estilo da página. Por esse motivo, você poderá perceber uma certa lentidão no carregamento, dependendo da velocidade da sua internet. Esse modo de instalação é bem simples, pois não requer de nenhuma habilidade especial, porém o código é meio "fechado", não sendo possível fazer alteração de nenhuma variável ou classe do Bootstrap. Com isso, é uma prática comum termos, além do Bootstrap, um arquivo `de.css` com formatações mais específicas. Veja abaixo o `o.css` disponível no nosso *template*:

```
1  body{
2    background-color: #eeeeee;
3  }
4
5  .footer-background{
6    background-color: rgb(204, 199, 199);
7  }
8
9  @media(max-width:991px){
10    #heading{
11      padding-left: 50px;
12    }
13    #prc{
14      margin-left: 70px;
15      padding-left: 110px;
16    }
17    #quantity{
18      padding-left: 48px;
19    }
20    #produc{
21      padding-left: 40px;
22    }
23    #total{
24      padding-left: 54px;
25    }
26  }
```

```

27 @media(max-width:767px){
28     .mobile{
29         font-size: 10px;
30     }
31     h5{
32         font-size: 14px;
33     }
34     h6{
35         font-size: 9px;
36     }
37     #mobile-font{
38         font-size: 11px;
39     }
40     #prc{
41         font-weight: 700;
42         margin-left: -45px;
43         padding-left: 105px;
44     }
45     #quantity{
46         font-weight: 700;
47         padding-left: 6px;
48     }
49     #produc{
50         font-weight: 700;
51         padding-left: 0px;
52     }

```

```

53     #total{
54         font-weight: 700;
55         padding-left: 9px;
56     }
57     #image{
58         width: 60px;
59         height: 60px;
60     }
61     .col{
62         width: 100%;
63     }
64     #zero-pad{
65         padding: 2%;
66         margin-left: 10px;
67     }
68     #footer-font{
69         font-size: 12px;
70     }
71     #heading{
72         padding-top: 15px;
73     }
74 }

```

Repare que este código usa muitas propriedades que vimos na Unidade 1, lembra? São elas:

- **width**: define a largura de um elemento que pode ser definido com valor fixo usando pixel (px) ou valor dinâmico usando porcentagem (%);
- **height**: define altura de um elemento que pode ser definido com valor fixo usando pixel (px) ou valor dinâmico usando porcentagem (%);
- **margin**: define o espaçamento externo de um elemento.
- **padding**: define o espaçamento interno, ou seja, espaço entre a borda e o conteúdo.
- **background-color**: define a cor de fundo do elemento.
- **font-size**: define o tamanho da fonte. Esta propriedade permite tais unidades: px (pixels), em (unidade relativa), pt (pontos), %, entre outras.
- **font-weight**: define a intensidade ou espessura da fonte. Esta propriedade pode ter os seguintes valores: **normal** | **bold** | **bolder** | **lighter**, entre outros.

DESTAQUE

Quando as propriedades *padding* e *margin* recebem um único valor (`padding: 2px;`) alteram os quatro lados de um elemento: superior, direita, inferior e esquerda. Porém é possível especificar o valor de forma individual e separadamente (`padding-top: 2px; padding-right: 4px; padding-bottom: 6px; padding-left: 8px;`) ou em conjunto (`padding: 2px 4px 6px 8px;`), passando os 4 valores de uma vez.

Outra coisa que você deve ter percebido nesse código é que boa parte dele faz parte de *Media Queries* (linhas 9 e 27).

Media query, nada mais é do que uma estrutura do CSS que permite aplicar estilizações específicas para uma página web de acordo com certas condições, fazendo assim com esta página possa se adequar ao layout de tela em diferentes tamanhos e tipos de mídia (MEDIA, 2022).

Sua sintaxe é basicamente composta pela expressão `@media` seguida de uma condição entre `()`, veja:

```
@media (condição) {
/* Estilizações CSS */
}
```

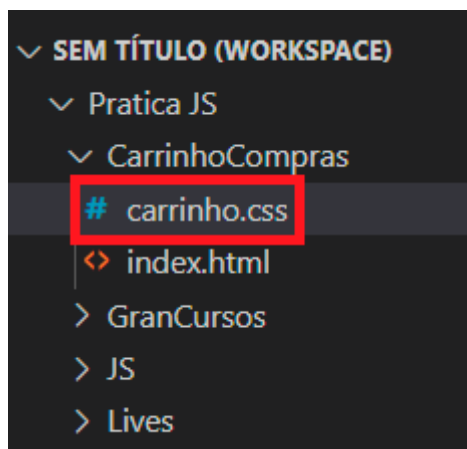

Esta condição resultará num resultado verdadeiro ou falso. Quando uma *media query* é verdadeira, as regras de estilos correspondentes são aplicadas, caso contrário as regras não serão aplicadas.

DESTAQUE

Saiba mais sobre como criar expressões media query em: <https://www.treinaweb.com.br/blog/media=-queries-o-que-sao-e-como-usar-no-css#:~:text=O%20que%20s%C3%A3o%20Media%20Queries,tamanhos%20e%20tipos%20de%20m%C3%ADdia.>

No nosso código estamos fazendo uso do que chamamos de *media features*, ela define a largura da janela. Usamos o max-width para definir a largura máxima de uma condição, definindo se a estilizações serão aplicadas conforme o tamanho da tela. Bom, vamos então vincular o css ao nosso código.

1º passo: crie um documento externo de.css.



2º passo: copie o código de css disponível no nosso template e cole neste documento que você criou.




Fonte: <https://bbbootstrap.com/snippets/bootstrap-ecommerce-cart-page-template-91851305#>.

3º passo: Insira na linha 11 a tag que vinculará o HTML ao CSS.




```
11 <link rel="stylesheet" href="carrinho.css">
```

A princípio não teremos nenhuma mudança no visual da nossa página. Repare que vai continuar do mesmo jeito, mas ao redimensionar a tela você perceberá a diferença. Veja a janela redimensionada **com** o css:


[ENTRAR](#)
[AJUDA](#)

LOJINHA DA GRANCURSOS

3 itens


PRODUTO	PREÇO	QUANTIDADE	TOTAL
 <div> Caneca Eu Dou Conta Código: 091091001 Cor: Branco </div>	R\$45,00	2	R\$90,00
 <div> Moleskine Código: 056289891 Cor: Vermelho </div>	R\$39,90	1	R\$39,90
 <div> Caneca Mágica Código: 091091001 Cor: Bege </div>	R\$49,90	1	R\$49,90

VOLTAR

SUBTOTAL R\$179,80




CONTINUAR

Agora veja a janela redimensionada **sem** o css:


[ENTRAR](#) [AJUDA](#)

LOJINHA DA GRANCURSOS

3 itens

PRODUTOS	PREÇO	QUANTIDADE	TOTAL
 <p>Caneca Eu dou Conta Código: 091091001 Cor: Branco</p>	R\$45,00	2	R\$90,00 ✕
 <p>Moleskine Código: 056289891 Cor: Vermelho</p>	R\$39,90	1	R\$39,90 ✕
 <p>Caneca Mágica Código: 091091001 Cor: Bege</p>	R\$49,90	1	R\$49,90 ✕

[VOLTAR](#)

SUBTOTAL R\$179,80

[CONTINUAR](#)

Percebeu a diferença? É nítido, não é mesmo? Logo, podemos concluir que o css do nosso template é basicamente voltado para a dimensão da janela.

O conteúdo deste livro eletrônico é licenciado para GLEITON - 08303020692, vedada, por quaisquer meios e a qualquer título, a sua reprodução, cópia, divulgação ou distribuição, sujeitando-se aos infratores à responsabilização civil e criminal.

CONSIDERAÇÕES FINAIS

Nesta aula, vimos que podemos trabalhar com CSS e Bootstrap ao mesmo tempo. Além disso, aprendemos um conceito novo chamado *media query*. Conhecimento nunca é demais, não é mesmo? Exploramos o código do nosso *template* e vimos que muitas das propriedades utilizadas para estilizar o nosso carrinho foram aprendidas na *Unidade 1*. Legal, né? Com isso, temos certeza de que o conteúdo aprendido aqui é de fato utilizado na prática.

MATERIAIS COMPLEMENTARES

Usando Medias queries: https://developer.mozilla.org/pt-BR/docs/Web/CSS/Media_Queries/Using_media_queries.

REFERÊNCIAS

MEDIA. *Media queries, o que são e como usar no CSS?* 2022. Disponível em: <https://www.treinaweb.com.br/blog/media-queries-o-que-sao-e-como-usar-no-css#:~:text=O%20que%20s%C3%A3o%20Media%20Queries,tamanhos%20e%20tipos%20de%20m%C3%ADdia>. Acesso em: 30 nov. 2022.

AULA 5 – MÃO NA MASSA – JAVASCRIPT

OBJETIVO DA AULA

Aplicar os conceitos de JavaScript no projeto.

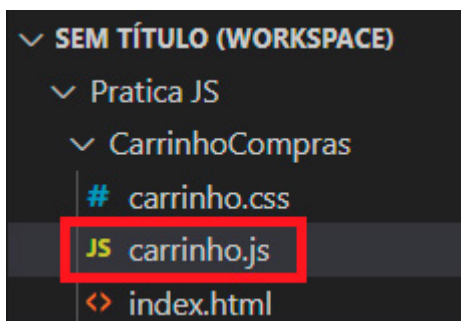
APRESENTAÇÃO

Agora sim! Chegamos na melhor parte do projeto! Vamos tornar este carrinho dinâmico e colocar tudo para funcionar? Até o momento, construímos todo o layout, porém os botões de + e - não estão funcionando e o cálculo dos valores também não. Sendo assim, nesta aula, vamos programar o nosso carrinho com JavaScript. Vem comigo!

1. JAVASCRIPT

Antes de começar a programar precisamos fazer algumas modificações no HTML. Será necessário criar vincular o arquivo externo de.js ao HTML e criar os identificadores (id) para os elementos que manipularemos no JavaScript.

1º passo: crie um documento externo de.js.



2º passo: Insira na linha 12 a tag que vinculará o HTML ao JS.

```
12      <script src="carrinho.js"></script>
```

3º passo: identificar os elementos de HTML que serão manipulados e criar seus respectivos id's. Na linha 34, temos o nosso primeiro produto. Vamos criar um id para o produto, para a quantidade e para o total.



Aqui está todas as modificações que fizemos no código que representa o primeiro produto:

```

34      <div id="produto0" class="d-flex flex-row justify-content-between align-items-center pt-lg-4 pt-2 pb-3
border-bottom mobile">
35        <div class="d-flex flex-row align-items-center">
36          <div></div>
37          <div class="d-flex flex-column pl-md-3 pl-1">
38            <div><h6>Caneca Eu dou Conta</h6></div>
39            <div>Código:<span class="pl-2">091091001</span></div>
40            <div>Cor:<span class="pl-3">Branco</span></div>
41          </div>
42        </div>
43        <div class="pl-md-0 pl-1"><b>R$45,00</b></div>
44        <div class="pl-md-0 pl-2">
45          <span class="fa fa-minus-square text-secondary" onclick="removerItem(0)">
46            </span><span class="px-md-3 px-1" id="quantidade0">0</span>
47          <span class="fa fa-plus-square text-secondary" onclick="adicionarItem(0)"></span>
48        </div>
49        <div class="pl-md-0 pl-1">
50          <b><span id="total0">R$</span></b>
51        </div>
52        <div class="close">&times;</div>
53      </div>

```

Temos três produtos no nosso carrinho, vamos nos referir ao primeiro como produto0, o segundo como produto1 e o terceiro como produto2. Por que optei por estes nomes? Porque vamos utilizar vetores para manipular estes elementos. Lembre que cada elemento do vetor possui um índice e que esse índice começa por 0? Então, por isso, optei por utilizar esta nomenclatura.

Sendo assim, nosso primeiro produto foi identificado por produto0 (*linha 34*), seguindo esta mesma linha de raciocínio a quantidade (*linha 46*) e o total (*linha 50*) também estão acompanhados do número 0. Repare que todos os que estavam numa única linha foram quebrados em 3 linhas para facilitar nosso entendimento, a *linha 45* representa o botão de +, a *linha 46* representa a quantidade (repare que zeramos o valor da quantidade, pois este valor será modificado dinamicamente pelo usuário), a *linha 47* representa o botão de -. Criamos também um novo para representar nosso subtotal (*linha 50*), identificamos como total0, apagamos o valor estático e deixamos somente o R\$. Por quê? Porque esse valor será calculado de acordo com a quantidade que o usuário irá escolher. Além disso, criamos um evento onclick para os botões + e - e associamos as funções removerItem(0) e adicionarItem(0), ou seja, ao clicarmos neste botões eles chamarão estas funções que iremos criar no JS. Passamos 0 por parâmetro para estas funções, pois estamos nos referindo ao produto0. Agora vamos fazer a mesma coisa para o produto1 e produto2. Aqui está todas as modificações que fizemos no produto1:

```

54      <div id="produto1" class="d-flex flex-row justify-content-between align-items-center pt-4 pb-3 mobile">
55        <div class="d-flex flex-row align-items-center">
56          <div></div>
57          <div class="d-flex flex-column pl-md-3 pl-1">
58            <div><h6>Moleskine</h6></div>
59            <div>Código:<span class="pl-2">056289891</span></div>
60            <div>Cor:<span class="pl-3">Vermelho</span></div>
61          </div>
62        </div>
63        <div class="pl-md-0 pl-1"><b>R$39,90</b></div>
64        <div class="pl-md-0 pl-2">
65          <span class="fa fa-minus-square text-secondary" onclick="removerItem(1)">
66            </span><span class="px-md-3 px-1" id="quantidade1">0</span>
67          <span class="fa fa-plus-square text-secondary" onclick="adicionarItem(1)"></span>
68        </div>
69        <div class="pl-md-0 pl-1">
70          <b><span id="total1">R$</span></b>
71        </div>
72        <div class="close">&times;</div>
73      </div>

```

Repare que agora passamos o valor 1 como parâmetro da função, pois estamos nos referindo ao produto1. Aqui estão todas as modificações que fizemos no produto2:

```

74 <div id="produto2" class="d-flex flex-row justify-content-between align-items-center pt-lg-4 pt-2 pb-3
border-bottom mobile">
75 <div class="d-flex flex-row align-items-center">
76 <div></div>
77 <div class="d-flex flex-column pl-md-3 pl-1">
78 <div><h6>Caneca Mágica</h6></div>
79 <div>Código:<span class="pl-2">091091001</span></div>
80 <div>Cor:<span class="pl-3">Bege</span></div>
81 </div>
82 </div>
83 <div class="pl-md-0 pl-1"><b>R$49,90</b></div>
84 <div class="pl-md-0 pl-2">
85 <span class="fa fa-minus-square text-secondary" onclick="removerItem(2)"></span>
86 <span class="px-md-3 px-1" id="quantidade2">0</span>
87 <span class="fa fa-plus-square text-secondary" onclick="adicionarItem(2)"></span>
88 </div>
89 <div class="pl-md-0 pl-1">
90 <b><span id="total2">R$</span></b>
91 </div>
92 <div class="close">&times;</div>
93 </div>

```

Repare que agora passamos o valor 2 como parâmetro da função, pois estamos nos referindo ao produto2. Para finalizar vamos identificar o subtotal:

```

105 <b class="pl-md-4">SUBTOTAL R$<span class="pl-md-4" id="valorTotalCompra"></span></b>

```

Repare que aqui também apagamos o valor estático e acrescentamos o R\$ ao lado o SUBTOTAL. Feito todas essas modificações, agora vamos criar os nossos scripts.

4º passo: criar os vetores, que armazenaram os valores referentes a quantidade, valor unitário e valor total por produto, no documento de JS.

```

1 var qtd = [0,0,0]; //quantidade de cada produto
2 var valorProduto = [45.00,39.90, 49.90]; //valor unitário de cada produto
3 var valorTotal = [0,0,0]; //valor total de cada produto

```

Repare que cada vetor tem 3 posições, cada uma dessas posições representa os nossos produtos respectivamente. A primeira posição de todos os vetores criados representa o produto0 e assim sucessivamente.

5º passo: criar a função adicionarItem(item):

```

5 function adicionarItem(item){
6   var quantidade = document.getElementById('quantidade' + item);
7   qtd[item] = qtd[item] + 1; //cálculo que adiciona +1 na quantidade
8   quantidade.innerHTML = qtd[item]; //exibe o valor da quantidade na tela

```

A função adicionarItem(item) recebe item como parâmetro, esse item é o valor que colocamos dentro dos () na chamada da função. Lembra que para cada botão de + e - adicionamos as funções adicionarItem e removerItem? A questão é, para cada produto passamos um valor diferente como parâmetro, no produto0 passamos adicionarItem(0) e removerItem(0), no produto1 passamos adicionarItem(1) e removerItem(1), no produto2 passamos adicionarItem(2) e removerItem(2). Para entender o código acima vamos imaginar que estamos adicionando o

produto0, logo receberemos como parâmetro o valor 0, concorda? Então, criamos uma variável (linha 6) chamada quantidade que vai receber o texto “quantidade” + item, ou seja, quantidade0. Na linha 7 estamos fazendo o cálculo que adicionará +1 na quantidade toda vez que pressionarmos o botão +. Na linha 8, estamos exibindo o valor escolhido pelo usuário na tela. Ele só é capaz de exibir o número no local correto, pois a variável quantidade recebeu o valor quantidade0 que é exatamente o id que criamos para a quantidade que representa o produto0.

```
46 <span class="px-md-3 px-1" id="quantidade0">0</span>
```

Agora o botão de + de todos os produtos devem estar funcionando, veja:

PRODUTOS	PREÇO	QUANTIDADE	TOTAL
 <p>Caneca Eu dou Conta Código: 091091001 Cor: Branco</p>	RS45,00	<div>- 2 +</div>	RS ×
 <p>Moleskine Código: 056289891 Cor: Vermelho</p>	RS39,90	<div>- 1 +</div>	RS ×
 <p>Caneca Mágica Código: 091091001 Cor: Bege</p>	RS49,90	<div>- 3 +</div>	RS ×

Porém, o valor total de cada produto não está sendo calculado. Vamos agora acrescentar ao nosso código a parte referente a este cálculo

```
5 function adicionarItem(item){
6   var quantidade = document.getElementById('quantidade' + item);
7   qtd[item] = qtd[item] + 1; //cálculo que adiciona +1 na quantidade
8   quantidade.innerHTML = qtd[item]; //exibe o valor da quantidade na tela
9
10  var total = document.getElementById("total" + item);
11  valorTotal[item] = Number.parseFloat(valorProduto[item]) * qtd[item];
12  //multiplica o valor unitário pela quantidade
13  total.innerHTML = valorTotal[item].toFixed(2); //exibe o valor total do produto na tela
14
15 }
```

Assim como no cálculo da quantidade, criamos uma variável (linha 10) chamada total que vai receber o texto “quantidade” + item, ou seja, total0. Na linha 11 estamos fazendo o cálculo que multiplicará a quantidade pelo valor unitário do produto resultando no valor total de cada produto. Na linha 13, estamos exibindo o valor total calculado na tela. Ele só é capaz

de exibir o valor no local correto, pois a variável total recebeu o valor total0 que é exatamente o id que criamos para total que representa o produto0. O toFixed(2) foi usado para limitarmos o número de casas decimais.

```
50 <b><span id="total0">R$</span></b>
```

Agora, ao adicionar um produto ele calcula automaticamente o valor total do produto, veja:

PRODUTOS	PREÇO	QUANTIDADE	TOTAL
 <p>Caneca Eu dou Conta Código: 091091001 Cor: Branco</p>	R\$45,00	- 2 +	90.00 x
 <p>Moleskine Código: 056289891 Cor: Vermelho</p>	R\$39,90	- 1 +	39.90 x
 <p>Caneca Mágica Código: 091091001 Cor: Bege</p>	R\$49,90	- 3 +	149.70 x

6º passo: criar a função `removerItem(item)`, para isso, vamos copiar e colar o código do `adicionarItem(item)` e fazer uns pequenos ajustes:

```
17 function removerItem(item){
18     if(qtd[item] > 0){
19         var quantidade = document.getElementById('quantidade' + item);
20         qtd[item] = qtd[item] - 1; //cálculo que remove -1 na quantidade
21         quantidade.innerHTML = qtd[item]; //exibe o valor da quantidade na tela
22
23         var total = document.getElementById("total" + item);
24         valorTotal[item] = Number.parseFloat(valorProduto[item]) * qtd[item];
25         //multiplica o valor unitário pela quantidade
26         total.innerHTML = valorTotal[item].toFixed(2); //exibe o valor total do produto na tela
27     }
28 }
29
```

O cálculo é o mesmo, basta diminuirmos ao invés de somar na *linha 20*, pois estamos removendo um item. Além disso, acrescentamos um `if` para quantidade não ficar com o valor negativo. Agora o nosso botão de – também está funcionando e recalculando o valor automaticamente.

7º passo: criar a função `valorCompra()`, ela será responsável por calcular o valor total da compra


```
31 function valorCompra(){
32     var valorTotalCompra = document.getElementById('valorTotalCompra');
33     var valor = 0;
34     for(var i=0; i<valorTotal.length;i++){
35         valor+=valorTotal[i];
36     }
37     valorTotalCompra.innerHTML = valor.toFixed(2);
38 }
```

Para realizar a soma de todos os produtos adicionados no carrinho precisamos pegar os valores que estão presentes no array `valorTotal`, para isso utilizaremos a estrutura de repetição `for`, ela será responsável por percorrer o vetor, pegar valor total de cada um dos produtos e somá-los. O método `.length` foi utilizado para verificar o tamanho do array `valorTotal`. O resultado desta soma será armazenado na variável `valor` (linha 35), o laço é necessário porque cada produto tem seu valor em uma posição específica no array.

Bom, o subtotal é recalculado toda vez que você adiciona ou remove um item, concorda? Logo, você deve estar se perguntando: Por que esse código não está presente dentro do `adicionarItem()` e `removerItem()`? Para não termos que repetir o código dentro de cada uma das funções acima, criamos uma nova função e agora vamos chamá-la dentro das funções `adicionarItem()` e `removerItem()`, veja:

```
5 function adicionarItem(item){
6     var quantidade = document.getElementById('quantidade' + item);
7     qtd[item] = qtd[item] + 1; //cálculo que adiciona +1 na quantidade
8     quantidade.innerHTML = qtd[item]; //exibe o valor da quantidade na tela
9
10    var total = document.getElementById("total" + item);
11    valorTotal[item] = Number.parseFloat(valorProduto[item]) * qtd[item];
12    //multiplica o valor unitário pela quantidade
13    total.innerHTML = valorTotal[item].toFixed(2); //exibe o valor total do produto na tela
14
15    valorCompra();
16 }
17 function removerItem(item){
18     if(qtd[item] > 0){
19         var quantidade = document.getElementById('quantidade' + item);
20         qtd[item] = qtd[item] - 1; //cálculo que remove -1 na quantidade
21         quantidade.innerHTML = qtd[item]; //exibe o valor da quantidade na tela
22
23         var total = document.getElementById("total" + item);
24         valorTotal[item] = Number.parseFloat(valorProduto[item]) * qtd[item];
25         //multiplica o valor unitário pela quantidade
26         total.innerHTML = valorTotal[item].toFixed(2); //exibe o valor total do produto na tela
27
28         valorCompra();
29     }
30 }
```





Finalizamos o nosso carrinho! Uhuuuu! Agora vamos testar para ver se está tudo certo:



ENTRAR AJUDA

LOJINHA DA GRANCURSOS

3 itens

PRODUTOS	PREÇO	QUANTIDADE	TOTAL	
 <div> <div>Caneca Eu dou Conta</div> <div>Código: 091091001</div> <div>Cor: Branco</div> </div>	R\$45,00	2	90.00	×
 <div> <div>Moleskine</div> <div>Código: 056289891</div> <div>Cor: Vermelho</div> </div>	R\$39,90	1	39.90	×
 <div> <div>Caneca Mágica</div> <div>Código: 091091001</div> <div>Cor: Rosa</div> </div>	R\$49,90	3	149.70	×

VOLTAR

SUBTOTAL R\$ 279.60

CONTINUAR

O subtotal foi realizado com sucesso!

CONSIDERAÇÕES FINAIS

Nesta aula, inserimos no nosso projeto a parte da programação, ou seja, fizemos os botões + e - funcionar, calculamos o valor total de cada produto e também o total geral da compra.

MATERIAIS COMPLEMENTARES

Se você se perdeu no meio da explicação e não conseguiu evoluir no seu código, acesse este link <https://github.com/GRANCodigo/PraticaDeProgramacao/blob/ec06942681daf31c-23c466b318e6a72aa232fa2c/Unidade4Aula1c> para ter acesso ao código completo.

REFERÊNCIAS

DE MATOS, F. J. M. JavaScript Progressivo: *Curso completo de JavaScript para iniciantes*. 2015. Disponível em: <<https://www.javascriptprogressivo.net/>>. Acesso em: 20 nov. 2022.