

## Melhorando a tela do carrinho

### Transcrição

Nesta aula queremos deixar nosso Carrinho completamente funcional. Depois disso iremos um pouco mais além e veremos como seria realizar o cadastro do nosso usuário para finalizar a compra.

Na parte da quantidade do código xhtml do Carrinho fazemos:

```
<input class="formularioDoCarrinho-item-quantidade" type="number" min="0" jsf:value="#{carrinhoItem.quantidade}" />
```

Como o CDI vai saber quais itens estão no carrinho? `itens.quantidade` é um *array*, então precisamos de um índice que nos é entregue por meio do `varStatus` :

```
<ui:repeat value="#{carrinhoCompras.items}" var="carrinhoItem" varStatus="loop">
```

Desse modo conseguimos ter o status (índice) atual em que o loop está iterado, logo:

```
<input class="formularioDoCarrinho-item-quantidade" type="number" min="0" jsf:value="#{carrinhoItem.quantidade}" />
```

Precisamos modificar também o botão "Atualizar":

```
<button class="formularioDoCarrinho-atualizar" type="submit">Atualizar</button>
```

Agora vejamos o *form*, se existe uma *action*, ela será chamada, pois o botão é do tipo *submit*, então podemos tirá-la:

```
<form class="formularioDoCarrinho" method="post">
```

Basta mexermos no CSS do botão, retirando o link da imagem procedente da Casa do Código e substituindo com o do projeto:

```
.formularioDoCarrinho-atualizar {  
    background-image: url(../imagens/refresh.png)  
    ...  
}
```

De fato temos o botão de atualizar:

### Seu carrinho

Item	Preço	Qty	Total
 <div>SEO Prático</div>	R\$ 59.00	<input type="text" value="1"/>	R\$ 59.00
<a href="#">FINALIZAR COMPRA</a>			<b>R\$ 59.00</b>

Porém ele ainda não está funcional. Se incrementamos a quantidade de itens e clicamos nele, os valores não atualizam. O JSF não sabe qual é o formulário referente a esse input:

```
<form class="formularioDoCarrinho" method="post" jsf:id="formCarrinho">
```

E no botão:

```
<button class="formularioDoCarrinho-atualizar" type="submit" jsf:id="btnAtualizar">Atualizar</button>
```

Atualizamos a página e testamos:

### Seu carrinho

Item	Preço	Qty	Total
 <div>SEO Prático</div>	R\$ 59.00	<input type="text" value="5"/>	R\$ 295.00
<a href="#">FINALIZAR COMPRA</a>			<b>R\$ 295.00</b>

De fato, está funcionando! Mesmo adicionando outro item:

### Seu carrinho

Item	Preço	Qty	Total
 <div>Test-Driven Development</div>	R\$ 59.00	<input type="text" value="10"/>	R\$ 590.00
 <div>SEO Prático</div>	R\$ 59.00	<input type="text" value="5"/>	R\$ 295.00
<a href="#">FINALIZAR COMPRA</a>			<b>R\$ 885.00</b>

Vamos implementar o link de Remover com uma `action`, que ainda não funciona:

```
<a jsf:action="#{carrinhoComprasBean.remove(carrinhoItem)}">
```

No `CarrinhoComprasBean.java`:

```
public void remover(CarrinhoItem item) {
    carrinho.remover(item);
}
```

No CarrinhoCompras.java :

```
public void remover(CarrinhoItem item) {
    this.itens.remove(item);
}
```

Testando, de fato o item é removido.

O próximo passo é fazer com que a sacolinha do lado direito superior mostre a quantidade de itens adicionados no Carrinho.

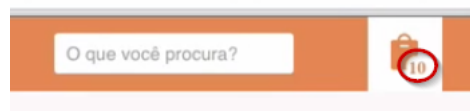


Vamos mexer no text :

```
<text text-decoration="none" text-anchor="middle" x="19.5" y="30" class="sacola-contador">0</te;
```

Perceba que modificando o número - agora está "0" - modifica-se a imagem:

```
<text text-decoration="none" text-anchor="middle" x="19.5" y="30" class="sacola-contador">10</t;
```



É exatamente aqui que queremos mudar:

```
<text text-decoration="none" text-anchor="middle" x="19.5" y="30" class="sacola-contador">#{cari
```

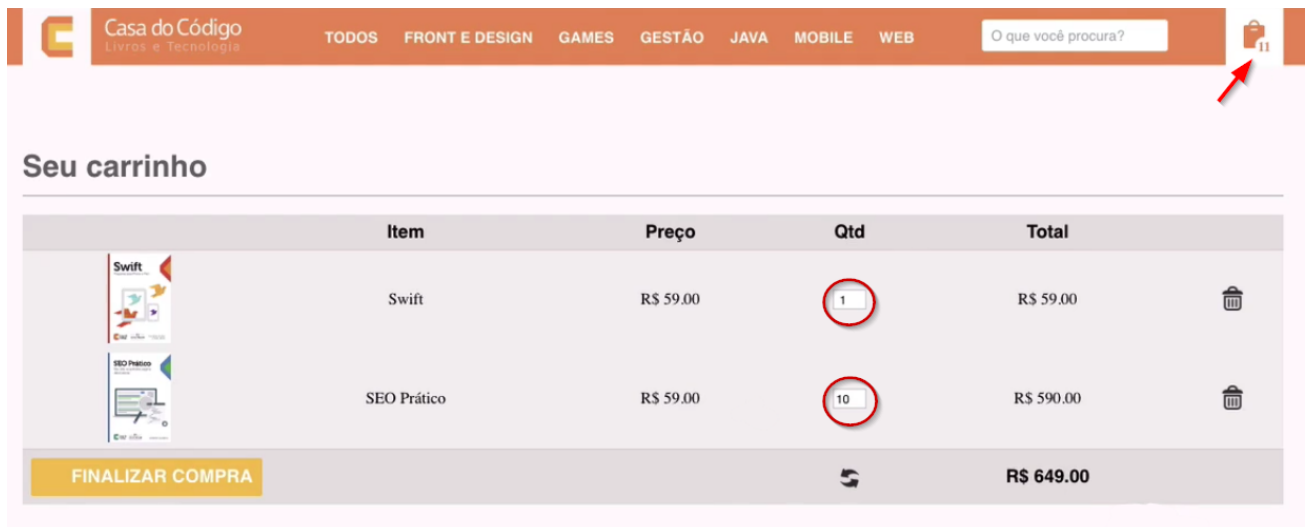
Também vamos mudar o título, um pouco a cima do código:

```
<title id="sacolaLabel">
  Você tem #{carrinhoCompras.getQuantidadeTotal()} itens na sacola
</title>
```




E dentro do `CarrinhoCompras.java`, usamos a nova API de *streams* do Java 8:

```
public Integer getQuantidadeTotal() {
    return itens.stream().mapToInt(item -> item.getQuantidade()).sum();
}
```

Com esse método conseguimos pegar o total de itens em relação à quantidade, inclusive dentro de cada item. Vejamos:



**Seu carrinho**

Item	Preço	Qtd	Total
 Swift	R\$ 59.00	1	R\$ 59.00
 SEO Prático	R\$ 59.00	10	R\$ 590.00
<b>FINALIZAR COMPRA</b>			<b>R\$ 649.00</b>

De fato, conseguimos fazer funcionar!

Se você tiver qualquer dúvida sobre Java 8 e como usar API de streams, sugerimos fazer o curso da Alura de [Java 8](https://cursos.alura.com.br/course/java8-lambdas) (<https://cursos.alura.com.br/course/java8-lambdas>) do instrutor [Paulo Silveira](https://cursos.alura.com.br/user/paulo-silveira) (<https://cursos.alura.com.br/user/paulo-silveira>).