





## Adicionando página do carrinho

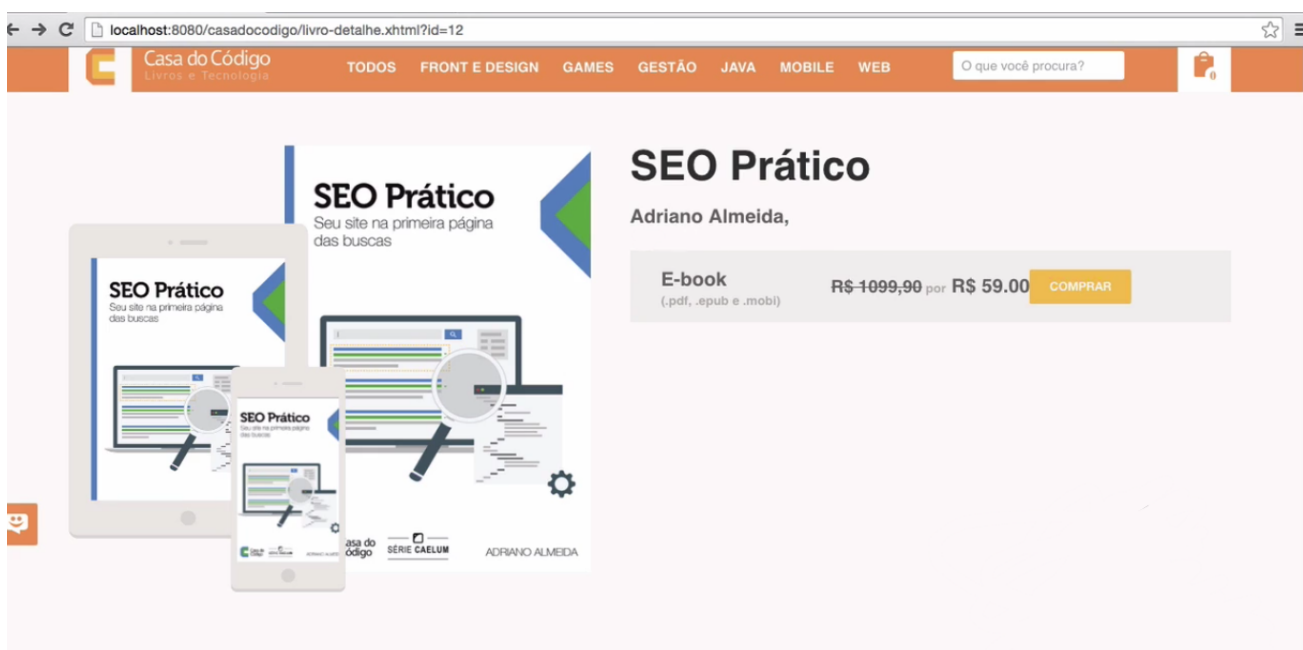
### Transcrição

O nosso carrinho deve ser preenchido com os itens que selecionamos ao clicarmos em "Comprar". O item aparece no Carrinho com sua imagem, seu nome, preço, quantidade, total por item e o botão para excluir. Como podemos ver no exemplo da Casa do Código:

Seu carrinho				
Item	Preço	Qtd	Total	
 Java SE 8 Programmer I: O guia para sua certificação Oracle Certified Associate - E-book	R\$39,90	1	R\$39,90	
 FINALIZAR COMPRA		 R\$39,90		

Para o curso implementaremos apenas o item em si e o botão para o removermos.

Neste momento no nosso sistema temos



Queremos que o botão de comprar faça com que o item vá para a tela do Carrinho. Começemos pelo início: por essa tela propriamente dita!

Vocês já terão disponível para download na hora do exercício. Colocamos os arquivos `carrinho.xhtml`, `refresh.png` e `trash.png` dentro da webapp. Você verá que ainda falta um arquivo, o `style-carrinho.css`. Este arquivo não iremos mexer, mas se você quiser, pode usá-lo para estilizar a tela do Carrinho. Não é o foco aqui alterarmos o CSS.

No `carrinho.xhtml` já foram inseridas todas as bibliotecas, inclusive o `passthrough`. Dessa maneira conseguimos ter uma tela preparada para nossos trabalhos, como já vimos anteriormente.

Nós só iremos mexer aqui dentro do `<tr>`

```

<tr>
  <td class="formularioDoCarrinho-item">
    <a href="https://www.casadocodigo.com.br...">
      
    </a>
    ...
  </td>
  ...
</tr>

```

E fazer com que ele se repita. Afinal para cada item adicionado no carrinho, uma linha a mais apareça. Temos já o componente com o qual já trabalhamos, que é o `<ui:repeat>`. Ele abrirá antes e fechará depois do `<tr>`:

```

<ui:repeat>
  <tr>
    <td class="formularioDoCarrinho-item">
      <a href="https://www.casadocodigo.com.br...">
        
      </a>
      ...
    </td>
    ...
  </tr>
</ui:repeat>

```

Qual o valor que queremos que se repita? Ora, dos itens que vieram do Bean, do nosso Carrinho de Compras. Então dentro do `<ui:repeat>`:

```

<ui:repeat value="#{carrinhoComprasBean.itens}" var="carrinhoItem">

```

Temos então um item se repetindo desde que no `carrinhoComprasBean` apareçam alguns itens. Hora de modificar as linhas para generalizarmos:

```

<td class="formularioDoCarrinho-item">
  <h2 class="formularioDoCarrinho-item-titulo">#{carrinhoItem.livro.titulo}</h2>
</td>
<td class="formularioDoCarrinho-item-preco">R$ #{carrinhoItem.livro.preco}</td>
<td class="formularioDoCarrinho-item">
  <input class="formularioDoCarrinho-item-quantidade" type="number" min="0" value="#{carrinhoItem.quantidade}">
</td>
<td class="formularioDoCarrinho-item formularioDoCarrinho-item-precoTotal" title="Preço unitário">

```

Vamos trabalhar agora com o botão "Remover":

```

/...

/...

```

Subindo um pouco no código, modificaremos a imagem do item no carrinho com uma URL padrão `/file`, que também usará o `contextPath`, assim como a página:

```
<td class="formularioDoCarrinho-item">
  <a href="#{request.contextPath}/livro-detalle.xhtml?id=#{carrinhoItem.livro.id}">
     getItens()`:

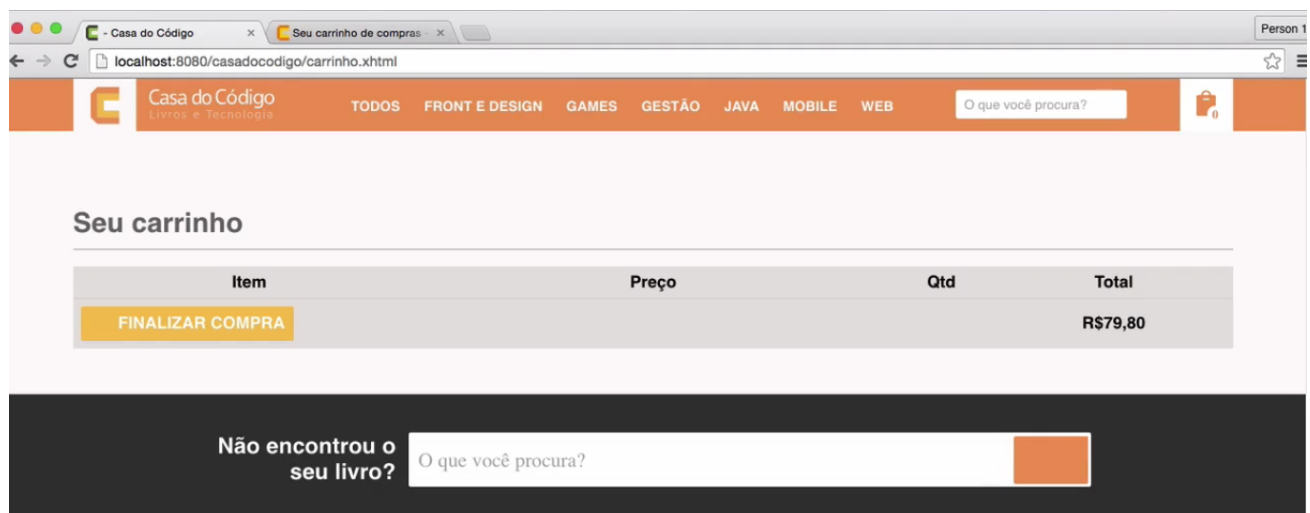
```
public List<CarrinhoItem> getItens() {
    return itens;
}
```

Temos um problema de tipos aqui. Temos tanto um `List<CarrinhoItem>` quanto um `Set<CarrinhoItem>`. Ambos não conversam diretamente, portanto devemos fazer:

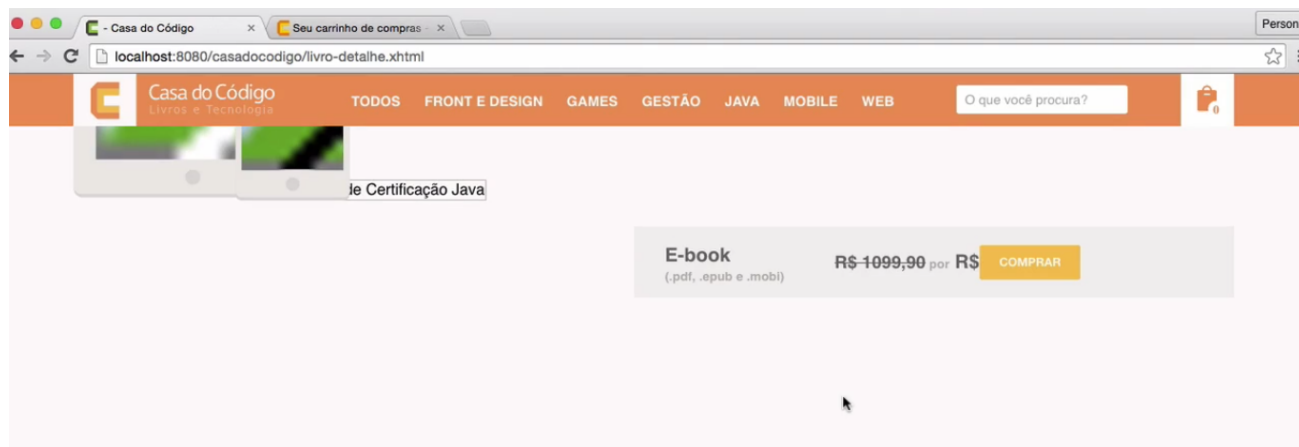
```
public List<CarrinhoItem> getItens() {
    return new ArrayList<CarrinhoItem>(itens);
}
```

É bom usarmos o `List` porque o `ui.repeat` só funciona através da interface `List`, diferente de quando usa um `Set`.

Vejamos enfim se nossa tela aparece. Publiquemos a aplicação no servidor:



O carrinho está vazio, mas está funcionando! Se clicarmos no botão "Comprar" na página de detalhe do livro, apresentará um erro:



Mas veremos a solução disso a diante.