

## Adicionando os itens do carrinho na sessão do usuário

### Transcrição

O problema que estamos tendo é que toda vez que clicamos no botão "Comprar", ele parece perder a referência e na URL não aparece mais o *id*. Isso acontece porque quando baixamos todo o código que veio direto da Casa do Código, o botão veio com o seguinte:

```
<button class="adicionarAoCarrinho-botaoComprar" jsf:action="#{carrinhoComprasBean.add(livroDet:
</button>
```

Tanto o `name` quanto o `value` são atributos que o JSF vai preencher em tempo de execução da nossa página, logo precisamos excluí-los:

```
<button class="adicionarAoCarrinho-botaoComprar" jsf:action="#{carrinhoComprasBean.add(livroDet:
</button>
```

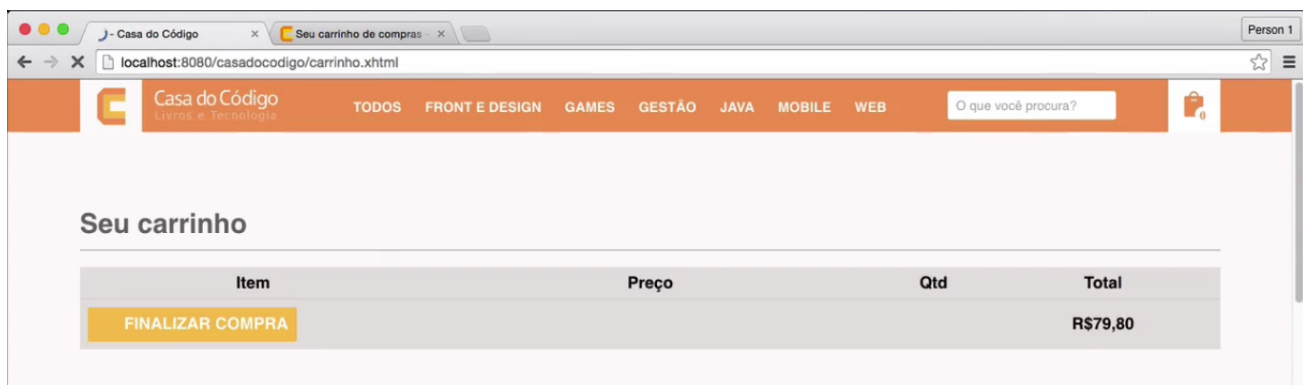
Um outro detalhe que pode ser facilmente resolvido é que o `livroDetalheBean.livro.id` não está sendo preenchido dessa forma, mas apenas `livroDetalheBean.id`, logo:

```
... jsf:action="#{carrinhoComprasBean.add(livroDetalheBean.id)}" ...
```

Ele já está preenchido pela metadata:

```
<f:metada>
  <f:viewParam value="#{livroDetalheBean.id}" name "id" />
  <f:viewAction action="#{livroDetalheBean.carregaDetalhe()}" />
</f:metadata>
```

O `name` é muito importante para o JSF, pois é assim que este controla os componentes. Desse modo, ao clicarmos em comprar, de fato vamos direto para o Carrinho de Compras.



Perceba porém que o Carrinho ainda aparece vazio.

O Carrinho de Compras é um elemento que mesmo saindo do site, ao voltarmos, mantém os itens selecionados.

Adicionar o item é um tipo de requisição que esperamos que permaneça no sistema, mesmo se o usuário encerre sua sessão. O próprio CTI tem uma `@session` :

`@SessionScoped`

```
public class CarrinhoCompras {

    // ...

}
```

O `@SessionScoped` fará com que possamos utilizar o Carrinho como sendo *escopo de sessão*. O usuário terá os itens salvos durante toda a sessão. Também precisamos implementar o *Serializable*:

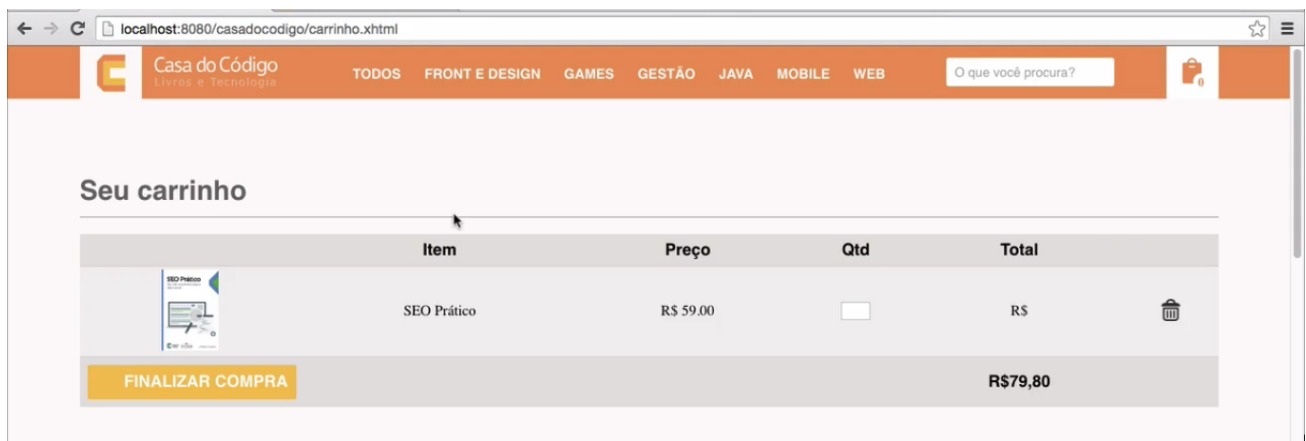
```
public class CarrinhoCompras implements Serializable {

}
```

Uma outra possibilidade interessante é usar o componente `@Named`. No `CarrinhoComprasBean.java`, o `getItens()` pode ser chamado diretamente pelo Carrinho. Com o `@Named` inserido, na tela do Carrinho podemos deixar:

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

Lá pela linha 276. Assim o `CarrinhoCompras` pode acessar direto os itens. Vejamos se conseguimos resolver nosso problema:



De fato, tudo ocorreu bem, porém a quantidade e o valor total de um único item vieram vazios. Então em `CarrinhoItem.java` fazemos no construtor:

```
public CarrinhoItem(Livro livro) {
    this.livro = livro;
    this.quantidade = 1;
}
```

Desse modo a quantidade recebe como padrão o valor 1

