

 02

Melhorando nossa Área Administrativa

Transcrição

Se você ainda não tem o código que fizemos no primeiro módulo, pode baixar através do link

<https://s3.amazonaws.com/caelum-online-public/java-ee-webapp/alura-casadocodigo-modulo1-projeto-final.zip>

(<https://s3.amazonaws.com/caelum-online-public/java-ee-webapp/alura-casadocodigo-modulo1-projeto-final.zip>)

Vamos começar melhorando nosso sistema. Já temos a Home e também o Detalhe do livro, mas nossa área de administração não está legal, temos apenas uma tabela comum na nossa lista e também nosso formulário está bem simples.

Para melhorar essa administração, faremos uso do Bootstrap que pode ser baixado em <http://getbootstrap.com/getting-started/#download> (<http://getbootstrap.com/getting-started/#download>).

Quando baixar, descompacte e dentro da pasta descompactada você deve estar vendo a estrutura com três novas pastas, sendo elas `css`, `fonts` e `js`. Copei o conteúdo da pasta `css` baixada para dentro da nossa pasta `src/main/webapp/resources/css`. Faça o mesmo procedimento para a pasta `js` que também já temos em nosso projeto, dentro de `src/main/webapp/resources/js`. Já a pasta `fonts` que não temos ainda no projeto, copie a pasta inteira para `src/main/webapp/resources`.

Para utilizar, abra o arquivo `lista.xhtml` e dentro dele vamos fazer uso do Bootstrap. Dentro da tag `<html>`, adicione o código abaixo, mantendo o conteúdo existente dentro da tag `<h:body>`:

```
<h:head>
</h:head>

<h:body>

    <!-- Mantenha o código da lista aqui! -->

</h:body>
```

As novas tags `<h:head>` e `<h:body>` servem para que o JSF possa identificar corretamente onde é o cabeçalho e corpo da página. É importante usá-los ao invés dos normais HTML que podem gerar problemas no JSF.

Uma vez que estamos dentro do JSF, precisamos utilizar uma tag diferente para declarar nossos CSS. Essa tag é a `<h:outputStylesheet>`. Com ela, declaramos os atributos `library` como sendo a pasta onde estão nossos CSSs, e também o nome do arquivo com o atributo `name`. O JSF 2.2 já considera a pasta `resources` como sendo padrão, então não precisamos declará-la.

Se você for até o sistema (se seu servidor não estiver rodando, coloque-o para subir), perceba que tivemos uma mudança bem sutil até o momento.

Vamos para nossa tag `<h:datatable>` e dentro dela, vamos adicionar um atributo chamado `class` que servirá para definir as classes CSS que nossa tabela terá. No atributo `class` coloque os valores `table-bordered table-striped`. Agora nossa tabela já estará bem melhor.

Caso você nunca tenha usado **Bootstrap**, basta acessar o site <http://getbootstrap.com/css/> (<http://getbootstrap.com/css/>) e no menu lateral direito, escolher quais tags você quer melhorar o css dele. Estamos usando a opção **tables**. Faça o teste com outros classes.

Uma outra vantagem do Bootstrap é poder aproveitar os templates formados que ele já possui. Acesse o endereço <http://getbootstrap.com/examples/starter-template/> (<http://getbootstrap.com/examples/starter-template/>) e com o botão direito do mouse, selecione a opção **Exibir código fonte da página**. Perceba que o template possui uma tag div como a seguir `<div class="container">`, então já vamos adicionar esse div logo abaixo da tag `<h:body>`. Antes do fechamento do `</h:body>` feche a div.

Vamos repetir as alterações que fizemos para a `lista.xhtml` também para o arquivo `form.xhtml`. Dentro do `form.xhtml`, você deve ter o resultado parecido com esse:

```
<h:head>
    <h:outputStylesheet library="css" name="bootstrap.min.css" />
</h:head>

<h:body>
    <div class="container">
        <!-- MANTER O CÓDIGO DO FORMULÁRIO AQUI -->
    </div>
</h:body>
```

Acesse o formulário e perceba que ele já está bem melhor e mais bem formatado.

Ainda vamos melhorar essa nossa administração nos próximos vídeos.