

Paginando a tabela

Transcrição

O próximo recurso que iremos utilizar é o [Paginator](#) (<http://www.primefaces.org/showcase/ui/data/datatable/paginator.xhtml>), pois teremos muitos livros e não podemos mostrar todos esses livros numa página só, por isso podemos paginar a tabela. E para fazer isso é bem simples, basta adicionar a tabela o atributo `paginator="true"` e o atributo `rows` com o número de linhas que queremos exibir por página:

```
<p:dataTable value="#{livroBean.livros}" var="livro" id="tabelaLivros"
    paginator="true" rows="5">
    <!-- conteúdo da tabela -->
</p:dataTable>
```

Filtrando os elementos da tabela

Por último, aplicaremos o [Filter](#) (<http://www.primefaces.org/showcase/ui/data/datatable/filter.xhtml>) para o usuário poder buscar o título do livro que ele quiser. Para isso, temos que adicionar o atributo `filterBy` na coluna que queremos (Livros), dizendo por qual valor queremos filtrar, mas não só isso, precisamos também definir um `filterMatchMode`. Nesse atributo precisamos definir que, conforme o usuário vai digitando o que procura, onde esse valor tem que aparecer, se o título deve começar ou terminar com o valor digitado, se deve conter o valor ou esse valor tem que ser exato. É isso que definimos nesse atributo, e no nosso caso queremos que o título comece com o que o usuário for digitando, logo seu valor será `startsWith`:

```
<f:facet name="header">Livros</f:facet>
<p:column headerText="Título" sortBy="#{livro.titulo}"
    filterBy="#{livro.titulo}" filterMatchMode="startsWith">
    <h:outputText value="#{livro.titulo}" />
</p:column>
```

Perfeito, agora conseguimos filtrar nossos livros, e com isso terminamos de "primefacear" a página `livro.xhtml`, mas ainda podemos fazer mais melhorias, fiquem à vontade para testá-las.

No próximo capítulo melhoraremos a navegação da nossa aplicação, criando um menu para essa tarefa, até lá!