02

Templates reutilizáveis com Facelets

Downloads

A imagem utilizada durante essa aula pode ser baixada: aqui (http://s3.amazonaws.com/caelum-online-public/JSF/logo.png)

Caso queira começar o treinamento a partir desse vídeo, pode baixar o projeto aqui (http://s3.amazonaws.com/caelum-online-public/JSF/livraria-capitulo8.zip) e importá-lo no Eclipse. Só baixe este arquivo se **não tiver feito os exercícios dos capítulos anteriores**.

Código duplicado na view

Com a página **livro.xhtml** em edição, queremos adicionar o logo da **Caelum** em seu topo através da *tag* `<h:graphicImage />`. É necessário indicar a localização da imagem, mas primeiro entenderemos como o JSF fornece acesso a este recurso.

Por padrão, no JSF 2.0, imagens, scripts e arquivos de CSS devem estar dentro de um diretório chamado **resources**, que fica dentro da pasta ***WebContent***. A pasta ainda não existe, logo, vamos criá-la. Agora, dentro da pasta **resources**, criaremos a pasta **img** e nela gravaremos o nosso logo.

Voltando a tag `<h:graphicImage />`, definiremos dois atributos. O primeiro é o `library`. Nele, preencheremos com o valor *img*, a pasta que contém nosso logo dentro de resources. O segundo é o `name`. Nele definimos o nome do arquivo, em nosso caso, **logo.png**. Vamos iniciar o servidor e verificar o resultado. O logo é exibido como esperado.

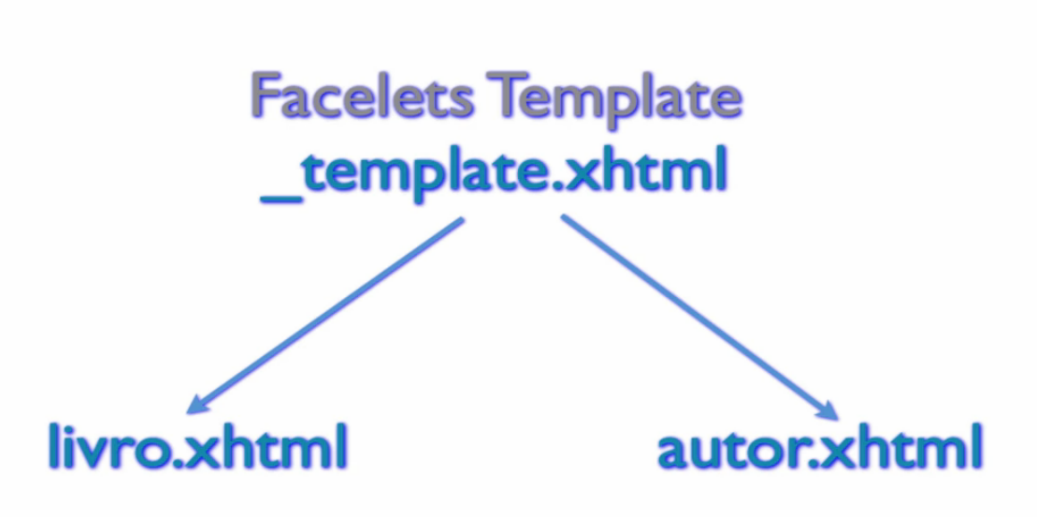
A ideia é termos o mesmo logo também em **autor.xhtml**. Ou seja, teremos um cabeçalho em todas as páginas. Repare que podemos fazer isto facilmente copiando e colando o código da página **livro.xhtml** para a página **autor.xhtml**.

O que aconteceria se além da imagem, nosso cabeçalho passasse a mostrar a informação do usuário logado no sistema? Com certeza isso nos daria problemas de manutenção, já que precisaríamos nos lembrar de atualizar o cabeçalho em todas as páginas.

Usando templates com Facelets

Para resolver problemas como este que o JSF vem com o mecanismo de ***templates*** através de ***facelets***. Isto é, um molde que pode ser aproveitado por outras páginas. Toda vez que este molde for atualizado, todas as páginas que o seguirem também serão atualizadas.

O template nada mais é do que uma página *xhtml* utilizando *facelets*. Por convenção, seu nome inicia com *underscore* (_), sendo assim, vamos criar nosso primeiro *template*.



Configurando um template

Com a pasta *WebContent* selecionada, vamos utilizar o atalho do Eclipse `Control + 3` seguido do texto *New file*. Apenas uma opção aparece. Tecle `Enter` para selecionar. O Eclipse solicita o nome do arquivo. Usaremos **_template.xhtml**.

Começaremos pela *tag* `<html>` . Nela, importaremos as bibliotecas que utilizaremos. Primeiro a do JSF, segundo a biblioteca do *facelet* com o *namespace* `xmlns:ui="http://java.sun.com/jsf/facelets"` . Por padrão, temos a *tag* `<h:head>` e `<h:body>` .

```
<html xmlns="http://www.w3.org/1999/xhtml"
      xmlns:f="http://java.sun.com/jsf/core"
      xmlns:h="http://java.sun.com/jsf/html"
      xmlns:ui="http://java.sun.com/jsf/facelets">

      <h:head />
      <h:body>
      </h:body>
</html>
```

Com essa estrutura mínima, vamos criar uma **div** com `id` ***cabecalho***. Precisamos mover a *tag* `<h:graphicImage />` da página **livro.xhtml** para o *template*, abrindo a página, cortando e colando em nosso template, dentro da **div cabecalho**.

Faz todo sentido aproveitarmos o logo em todas as nossas páginas, mas queremos também padronizar a exibição do título da página através da *tag* **h1**. Repare que a definição de seu conteúdo não é responsabilidade do *template*, mas de quem utilizá-lo.

Para isso, utilizaremos a *tag* `<ui:insert>` com o **name** *titulo* como conteúdo da *tag* **h1**. Esta *tag* indicará para as páginas que utilizarem o *template* que há uma área com `name titulo` que pode ser preenchida:

```
<html xmlns="http://www.w3.org/1999/xhtml"
      xmlns:f="http://java.sun.com/jsf/core"
      xmlns:h="http://java.sun.com/jsf/html"
      xmlns:ui="http://java.sun.com/jsf/facelets">

      <h:head />
      <h:body>
            <div id="cabecalho">
```

```
                    <h:graphicImage library="img" name="logo.png"/>
                    <h1><ui:insert name="titulo"></ui:insert></h1>
            </div>
        </h:body>
</html>
```

Não é apenas o título que muda de página para a página, mas o conteúdo da página. Sendo assim, vamos adicionar uma nova **div** com *id conteudo* e dentro dela a *tag* `<ui:insert>` com **name** *conteudo*:

```
<html xmlns="http://www.w3.org/1999/xhtml"
        xmlns:h="http://java.sun.com/jsf/html"
        xmlns:f="http://java.sun.com/jsf/core"
        xmlns:ui="http://java.sun.com/jsf/facelets">

    <h:head/>
    <h:body>
        <div id="cabecalho">
            <h:graphicImage library="img" name="logo.png"/>
            <h1><ui:insert name="titulo"></ui:insert></h1>
        </div>

        <div id="conteudo">
            <ui:insert name="conteudo">
            </ui:insert>
        </div>
    </h:body>
</html>
```

Associação da página com o template

Falta fazer a associação da página **livro.xhtml** com o *template*. Isto é feito envolvendo todo o conteúdo da página pela *tag* **ui:composition**. Agora precisamos indicar pelo atributo **template**, o nosso *template* **_template.xhtml**. Como estamos utilizando *facelet* na página, precisamos também importa-lo:

```
<html xmlns="http://www.w3.org/1999/xhtml"
    xmlns:h="http://java.sun.com/jsf/html"
    xmlns:f="http://java.sun.com/jsf/core"
    xmlns:ui="http://java.sun.com/jsf/facelets">

  <ui:composition template="_template.xhtml">
     <!--codigo omitido -->
```

Após indicar que a nossa página utiliza um *template*, precisamos definir as áreas em aberto do *template*. A primeira delas é **<ui:insert name="titulo">**. Para cada **ui:insert** do *template*, podemos ter um **ui:define** na página que o utiliza, adicionando a *tag* **<ui:define name="titulo">** com o conteúdo **Novo Livro**. Podemos remover a *tag* **h1**, pois ela agora vem do *template*.

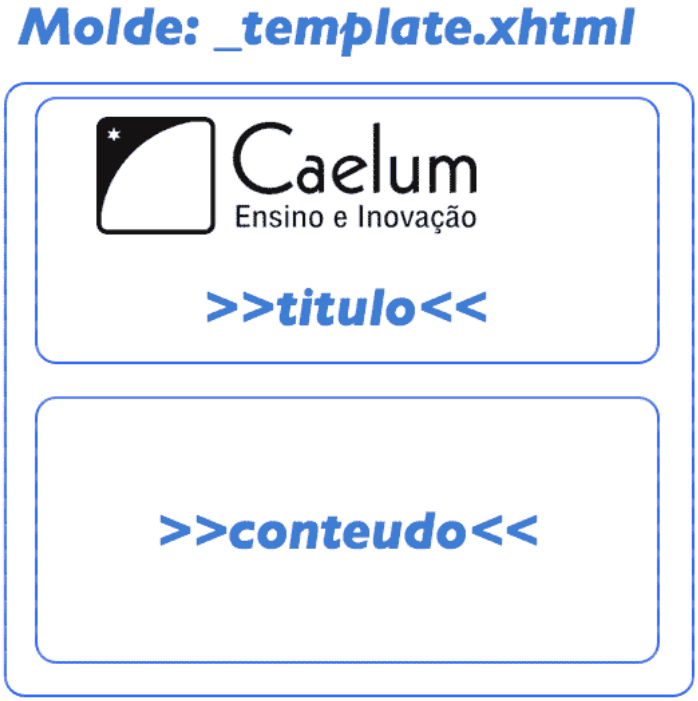
Por fim, precisamos colocar todo o conteúdo da página dentro da *tag* `<ui:define>`, mas agora com o **name** *conteudo*.

```
<html xmlns="http://www.w3.org/1999/xhtml"
    xmlns:h="http://java.sun.com/jsf/html"
```

```
    xmlns:f="http://java.sun.com/jsf/core"
    xmlns:ui="http://java.sun.com/jsf/facelets">

<ui:composition template="_template.xhtml">
    <ui:define name="titulo">
            Novo Livro
    </ui:define>
    <ui:define name="conteudo">
        <h:form>
                <!--codigo omitido -->
        </h:form>
                <!--codigo omitido -->
    </ui:define>
</ui:composition>
</html>
```

Pronto, terminamos as alterações necessárias. Vamos visualizar no navegador. Conforme esperado, nossa página autor herdou o logo do *template*, ao mesmo tempo em que personalizou o título e o conteúdo da página.



Aplicando o template nas demais páginas

Faremos a mesma coisa com a página **autor.xhtml**. Voltando ao Eclipse, abrindo a página para edição, importando o *facelets*. Agora, vamos colocar todo o conteúdo da página dentro da *tag* `<ui:composition>` , sem esquecer de definir o nosso template(**_template.xhtml**). Agora falta definir o título, por isso utilizaremos a *tag* `<ui:define name="titulo">` , colocando o nosso título *Novo Autor*. Agora falta envolver todo o conteúdo da página na *tag* `<ui:define name="conteudo">` .

```
<html xmlns="http://www.w3.org/1999/xhtml"
    xmlns:h="http://java.sun.com/jsf/html"
    xmlns:f="http://java.sun.com/jsf/core"
    xmlns:ui="http://java.sun.com/jsf/facelets">
```

```
<ui:composition template="_template.xhtml">
    <ui:define name="titulo">
            Novo Autor
        </ui:define>
    <ui:define name="conteudo">
        <h:form id="autor">
                <!--codigo omitido -->
        </h:form>
    </ui:define>
</ui:composition>
</html>
```

Agora, visualizando a página, vemos que o logo também é apresentado. Criamos um *template* usando *facelets*, que foi reutilizado tanto pela página livro, quanto autor, assim, reaproveitando o código e facilitando a manutenção.