



Componentes do JSF



Famílias de componentes

■ Comando

- *HtmlCommandButton*: um botão de submit
- *HtmlCommandLink*: um hiperlink com ação

■ Dados

- *HtmlDataTable*: tabela associada a uma fonte de dados

■ Gráficos

- *HtmlGraphicImage*: uma imagem

■ Mensagem

- *HtmlMessage*: exibe mensagens para a um componente

Famílias de componentes

■ Mensagens

- *HtmlMessages*: lista de mensagens

■ Entrada de dados

- *HtmlInputHidden*: campo oculto html
- *HtmlInputSecret*: campo de senha
- *HtmlInputText*: campo de entrada de dados simples
- *HtmlInputTextArea*: campo de entrada de texto

■ Seleção verdadeiro/falso

- *HtmlSelectBooleanCheckbox*: um checkbox

Famílias de componentes

■ Itens de seleção

- ***UISelectItem***: representam um item de uma lista ou combo
- ***UISelectItems***: representam vários itens

■ Saída de dados

- ***HtmlOutputFormat***: texto formatado (<**h:outputFormat**>)
- ***HtmlOutputLabel***: rótulos de campos (<**h:outputLabel**>)
- ***HtmlOutputLink***: hiperlink que não produz ação
- ***HtmlOutputText***: texto simples (<**h:outputText**>)
- ***UIOutput***: texto simples com marcação HTML (<**f:verbatim**>)

Famílias de componentes

■ Painel

- *HtmlPanelGrid*: tabela com campos configuráveis
- *HtmlPanelGroup*: agrupa componentes para aplicar padrões comuns ou aninhamento

■ Seleção única

- *HtmlSelectOneRadio*: grupo de botões de radio
- *HtmlSelectOneListbox*: lista de seleção única
- *HtmlSelectOneMenu*: combobox de seleção única

Famílias de componentes

- Seleção múltipla
 - *HtmlSelectManyCheckbox*: checkboxes agrupados
 - *HtmlSelectManyListbox*: lista de seleção múltipla
 - *HtmlSelectManyMenu*: lista de seleção múltipla
- Raiz da árvore de componentes
 - *UIViewRoot*: elemento raiz da árvore de componentes de uma visão

Propriedades

■ Propriedades comuns:

Propriedade	Tipo	Valor default	Obrigatório
<i>id</i>	String	não tem	sim
<i>value</i>	Object	não tem	sim
<i>rendered</i>	boolean	true	não
<i>converter</i>	Converter	não tem	não
<i>styleClass</i>	String	não tem	não
<i>binding</i>	String	não tem	não

Propriedades

- Propriedades pass-through
 - Não são validadas (passam direto para o código HTML)

```
<h:inputText id="form:nome" tabindex="3"  
size="20" maxlength="100" value="Fred"/>
```



```
<input type="text" id="form:nome"  
tabindex="3" size="20" maxlength="100" value="Fred"/>
```


Saída de dados

- Componentes de saída com formato

```
<h:outputText value="Dados <i>não</i> inseridos:"/>
```



Dados <i>não</i> inseridos:

```
<h:outputText value="Dados <i>não</i> inseridos:"  
escape="false"/>
```



Dados *não* inseridos:

Saída de dados

■ Componentes de saída simples

```
<f:verbatim escape="true">
  <servlet>
    <servlet-name>FrontController</servlet-name>
    <servlet-class>br.fred.Front</servlet-class>
  </servlet>
</f:verbatim>
```



```
&lt;servlet&gt;
  &lt;servlet-name&gt;FrontController&lt;/servlet-name&gt;
  &lt;servlet-class&gt;br.fred.Front&lt;/servlet-class&gt;
&lt;/servlet&gt;
```

Saída de dados

- **f:verbatim** também é usado para aninhar tags JSTL dentro de containers JSF

```
<h:panelGrid columns="2" border="1">
  <f:verbatim>
    <c:forEach items="{user.favoriteSites}" var="site">
      <c:out value="{site}, "/>
    </c:forEach>
  </f:verbatim>
  <f:verbatim escape="false">
    <p><font color="red">Texto <u>HTML</u>
    simples.</font></p>
  </f:verbatim>
</h:panelGrid>
```

Saída de dados

- Rótulos de componentes de entrada

```
<h:outputLabel for="userNameInput" accesskey="N">  
  <h:outputText value="Seu nome: "/>  
</h:outputLabel>  
<h:inputText id="userNameInput"/>
```

Browser

Seu nome:

Saída de dados

- Textos formatados:

```
<h:outputFormat value="Olá {0}. Seu perfil é {1}.  
{0}, seu browser é: {2}.">  
  <f:param value="#{user.name}"/>  
  <f:param value="programador"/>  
  <f:param value="#{header['User-Agent']}"/>  
</h:outputFormat>
```

Saída de dados

- Textos formatados:

- Date format

{**número**, **date**, padrão/estilo}

{1, date, medium}

{1, date, dd.MM.yyyy}

Dentro do value

- Choice format

{**número**, **choice**, padrão}

{1, choice, 0#nenhum|1uma|2#duas}

Saída de dados

■ Hiperlinks

```
<h:outputLink value="docs/requisitos.htm">  
  <h:outputText value="Documento de Requisitos"/>  
</h:outputFormat>
```



```
<a href="docs/requisitos.htm">Documento de Requisitos</a>
```

```
<h:outputLink value="listarTurma.do">  
  <h:outputText value="Listar turma"/>  
  <f:param name="cod" value="100"/>  
</h:outputFormat>
```

Mensagens de um componente

- Severidade da mensagem
 - Info: mensagem de informação ao usuário
 - Warn: um aviso de possibilidade de erro
 - Error: um erro ocorreu (ex. validação)
 - Fatal: um erro grave ocorreu

```
<h:inputText id="login" value="#{bb.login}"/>  
<h:message for="login"  
  showDetail="true" showSummary="true"  
  warnStyle="color: green"  
  infoStyle="color: blue"  
  errorStyle="color: red"/>
```


Mensagens de um componente

- Num método do *backing bean*

```
if (login é inválido) {  
    FacesMessage msg =  
        new FacesMessage("Login ou senha inválidos.");  
    FacesContext fc = FacesContext.getCurrentInstance();  
    fc.addMessage("msgLoginInvalido", msg);  
}  
...
```

Mensagens

- Exibe diversas mensagens de diferentes fontes, independentemente de componente

```
<h:messages  
  globalOnly="true"  
  warnStyle="color: green"  
  infoStyle="color: blue"  
  errorStyle="color: red"/>
```

Painéis

■ HtmlPanelGroup

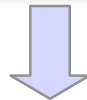
- Agrupa e aplica estilos comuns a diversos componentes
- Não produz HTML, exceto quando um estilo css ou um id é definido (nestes casos produz um ``)
- Usado para agrupar elementos em `<f:facet`

■ HtmlPanelGrid:

- Define layouts e estilos
- É renderizado para HTML num elemento `<table>`

Painéis

```
<h:panelGroup id="titulo">  
  <h:graphicImage url="images/seta.gif"/>  
  <h:outputText value="Consulta por nome"/>  
</h:panelGroup>
```



```
<span id="formTal:titulo">  
    
  Consulta por nome  
</span>
```

Painéis

```
<h:panelGrid columns="2" cellpadding="1"
border="1" width="40%">
  <h:outputLabel value="Nome:"/>
  <h:inputText id="login" value="#{bb.login}"/>
  <h:outputLabel value="Senha:"/>
  <h:inputText id="login" value="#{bb.login}"/>
  <h:commandButton value="Entrar" action=.../>
</h:panelGrid>
```

Nome:	<input type="text"/>
Senha:	<input type="text"/>
<input type="button" value="Entrar"/>	

Componentes de entrada de dados

■ Registrando listeners

- Componentes de entrada → eventos *value-change*

```
<h:inputText value="#{produtoBean.nome}"  
  valueChangeListener="#{produtoBean.nameChanged}"/>
```

- Componentes de comando → eventos *action*

```
<h:commandButton value="Gravar"  
  actionListener="#{produtoBean.salve}"/>
```

```
<h:commandButton value="Gravar">  
  <f:actionListener type="br.meu.ListenerTal" />  
  <f:actionListener type="br.meu.ListenerQual" />  
</h:commandButton>
```

Componentes de entrada de dados

- `HtmlForm`
- `HtmlInputText`
- `HtmlInputSecret`
- `HtmlInputHidden`
- `HtmlInputTextArea`

Os nomes já dizem tudo!

```
<h:inputText id="nome"
  value="#{produtoBean.nome}"
  valueChangeListener="#{produtoBean.nameChanged}"
  tabIndex="2"/>
```

Itens de seleção

■ UISelectItem

- Representa um único item de um conjunto de itens
- Usados por listboxes, botões de radio e checkboxes
- Não geram HTML, deixam isso para o componente pai

```
<h:componentePai>  
  <f:selectItem itemValue="0" itemLabel="PB"/>  
  <f:selectItem itemValue="1" itemLabel="PE"/>  
  <f:selectItem itemValue="2" itemLabel="RN"/>  
  <f:selectItem itemValue="3" itemLabel="SE"/>  
</h:componentePai>
```


Itens de seleção

■ UISelectItems

- Representa uma coleção de itens (SelectItem)
- Podem ser combinados com SelectItem simples
- Propriedade **value**: um SelectItem, um array ou collection de SelectItem ou um Map<String,String>

```
<h:componentePai>  
  <f:selectItem itemValue="0" itemLabel="PB"/>  
  <f:selectItem itemValue="1" itemLabel="PE"/>  
  <f:selectItem itemValue="2" itemLabel="RN"/>  
  <f:selectItems value="#{bbean.outrosEstados}" />  
</h:componentePai>
```

Componentes de seleção múltipla

■ HtmlSelectManyCheckbox

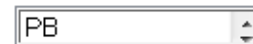
- Grupo de checkboxes organizados numa <table>
- Pode-se definir a direção (horizontal ou vertical)

■ HtmlSelectManyListbox

- Lista de opções
- Permite seleção múltipla

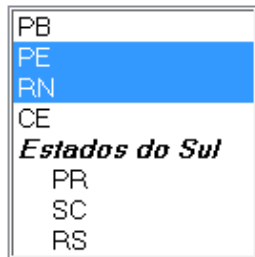
■ HtmlSelectManyMenu

- Lista de elementos em que apenas um por vez pode ser visualizado



Componentes de seleção múltipla

■ HtmlSelectManyListbox



```
SelectItem esul[] = new SelectItem[3];
esul[0] = new SelectItem("Parana", "PR");
esul[1] = new SelectItem("Sta. Catarina", "SC");
esul[2] = new SelectItem("R.G. do Sul", "RS");
sul = new SelectListGroup("Estados do Sul");
sul.setSelectItems(esul);
```

```
<h:selectManyListbox>
```

```
<f:selectItem itemValue="0" itemLabel="PB"/>
```

```
<f:selectItem itemValue="1" itemLabel="PE"/>
```

```
<f:selectItem itemValue="2" itemLabel="RN"/>
```

```
<f:selectItem itemValue="3" itemLabel="CE"/>
```

```
<f:selectItems value="#{bbean.sul}" />
```

```
</ h:selectManyListbox >
```

Componentes de seleção simples

■ HtmlSelectOneRadio

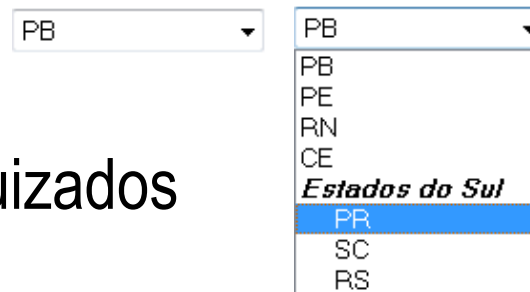
- Grupo de botões de radio numa tabela (<table>)
- Pode-se definir a direção (horizontal ou vertical)

■ HtmlSelectOneListbox

- Lista com seleção de apenas um único item

■ HtmlSelectOneMenu

- Um combobox
- Os itens podem ser hierarquizados



Componentes de comando

■ `HtmlCommandButton`

- Especifique `type="reset"` para botão de reset

```
<h:commandButton value="salvar"  
  action="#{bb.salvar}" />
```

■ `HtmlCommandLink`

```
<h:commandLink action="#{bb.salvar}" >  
  <h:outputText value="Salvar" />  
</h:commandLink>
```