

Soft Blue Soft Blue Soft Blue Soft Blue Soft Blue

Java Server Faces

Tags e Componentes JSF

Soft Blue

www.softblue.com.br

Todos os direitos de cia reservados. No  permitida a distribuio fsica ou eletrnica deste material sem a permisso expressa e por escrito do autor.

Tpicos Abordados

- Tags do JSF
- JSF HTML Tags e Core JSF Tags
 - Tags sem representao grfica
 - Textos e imagens
 - Caixas de texto
 - Campos hidden
 - Checkboxes, Radio buttons, Menus e Listboxes
 - Botes e Links
 - Painis
- JavaScript e Componentes JSF
- Resources

Tags do JSF

- Pginas JSF so compostas por tags
- JSF suporta 6 bibliotecas de tags

Nome	Namespace	Prefixo
HTML	http://java.sun.com/jsf/html	h:
Core	http://java.sun.com/jsf/core	f:
Facelets	http://java.sun.com/jsf/facelets	ui:
Composite Components	http://java.sun.com/jsf/composite	composite:
JSTL Core	http://java.sun.com/jsp/jstl/core	c:
JSTL Functions	http://java.sun.com/jsp/jstl/functions	fn:

- As tags padro do HTML tmbm so suportadas

Exemplo de Página JSF

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

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

  <h:head>
    ...
  </h:head>

  <h:body>
    ...
  </h:body>

</html>
```

Referencia as JSF
HTML tags

Referencia as tags
JSF Core

O prefixo é utilizado na
referência de determinada tag

Tags sem Representação Gráfica

Cabeçalho da página

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

Corpo da página

```
<h:body>
...
</h:body>
```

Formulários

```
<h:form>
...
</h:form>
```

Tags sem Representação Gráfica

Referência a arquivos JavaScript

```
<h:outputScript name="script.js" library="script" />
```

Referência a arquivos CSS

```
<h:outputStylesheet name="style.css" library="css" />
```

Textos e Imagens

Exibe texto na página

```
<h:outputText value="Boa tarde, #{bean.nome}!" />
```

Exibe texto na página, possibilitando a formatação do mesmo

```
<h:outputFormat value="{0}, {1}!">  
  <f:param value=#{bean.mensagem}" />  
  <f:param value=#{bean.nome}" />  
</h:outputFormat>
```

Exibe uma imagem

```
<h:graphicImage url="/images/figura.jpg" />
```

Caixas de Texto

Caixa de digitação de texto

```
<h:inputText value=#{bean.texto}" />
```

Texto na caixa de texto

Caixa de digitação de senha

```
<h:inputSecret value=#{bean.texto}" />
```

Caixa de digitação de texto em múltiplas linhas

```
<h:inputTextarea value=#{bean.itens}" rows="3" cols="10" />
```

Item 1
Item 2
Item 3

Campos Hidden

Campo que pertence ao formulário mas não é exibido

```
<h:inputHidden value=#{bean.id}" />
```

Seleção de Itens: Checkboxes

Checkbox para um único item

```
<h:selectBooleanCheckbox value="#{bean.ativo}" />
```

Ativo: ☒

Mapeado para um boolean no bean

Checkbox para múltiplos itens

```
<h:selectManyCheckbox value="#{bean.alimentos}">
  <f:selectItem itemValue="1" itemLabel="Macarrão" />
  <f:selectItem itemValue="2" itemLabel="Feijão" />
  <f:selectItem itemValue="3" itemLabel="Arroz" />
</h:selectManyCheckbox>
```

☒ Macarrão ☒ Feijão ☐ Arroz

Mapeado para um array ou coleção no bean

Utilize coleções apenas se os valores passados ao bean forem strings

Seleção de Itens: Checkboxes

Checkbox para múltiplos itens com lista dinâmica

```
<h:selectManyCheckbox value="#{bean.alimentos}">
  <f:selectItems value="#{bean.listaAlimentos}" />
</h:selectManyCheckbox>
```

☒ Macarrão ☒ Feijão ☐ Arroz

A lista de elementos é fornecida pelo bean...

```
private SelectItem[] listaAlimentos = {
    new SelectItem("1", "Macarrão"),
    new SelectItem("2", "Feijão"),
    new SelectItem("3", "Arroz")
};
```

Seleção de Itens: Checkboxes

Checkbox para múltiplos itens com lista dinâmica de qualquer tipo

```
<h:selectManyCheckbox value="#{bean.alimentos}">
  <f:selectItems value="#{bean.listaAlimentos}"
    var="a" itemValue="#{a.id}" itemLabel="#{a.nome}" />
</h:selectManyCheckbox>
```

☒ Macarrão ☒ Feijão ☐ Arroz

A lista de elementos é fornecida pelo bean...

```
private List<Alimento> listaAlimentos =
    Arrays.asList(new Alimento[] {
        new Alimento(1, "Macarrão"),
        new Alimento(2, "Feijão"),
        new Alimento(3, "Arroz")
    });
```

A classe Alimento possui os atributos id e nome, e os getters/setters correspondentes

Seleção de Itens: Radio Buttons

Radio button para seleção de um item entre vários

```
<h:selectOneRadio value="#{bean.alimento}">
  <f:selectItem itemValue="1" itemLabel="Macarrão" />
  <f:selectItem itemValue="2" itemLabel="Feijão" />
  <f:selectItem itemValue="3" itemLabel="Arroz" />
</h:selectOneRadio>
```

☐ Macarrão ☒ Feijão ☐ Arroz

A tag `f:selectItems` também pode ser utilizada com radio buttons

Seleção de Itens: Menus

Caixa de seleção de um único item

```
<h:selectOneMenu value="#{bean.alimento}">
  <f:selectItem itemLabel="Selecione" noSelectionOption="true" />
  <f:selectItem itemValue="1" itemLabel="Macarrão" />
  <f:selectItem itemValue="2" itemLabel="Feijão" />
  <f:selectItem itemValue="3" itemLabel="Arroz" />
</h:selectOneMenu>
```

Feijão
Selecione
Macarrão
Feijão
Arroz

A tag `f:selectItems` também pode ser utilizada com menus

Seleção de Itens: Listboxes

Lista de seleção de um único item

```
<h:selectOneListBox value="#{bean.alimento}">
  <f:selectItem itemValue="1" itemLabel="Macarrão" />
  <f:selectItem itemValue="2" itemLabel="Feijão" />
  <f:selectItem itemValue="3" itemLabel="Arroz" />
</h:selectOneListBox>
```

Macarrão
Alimento: Feijão
Arroz

Lista de seleção de múltiplos itens

```
<h:selectManyListBox value="#{bean.alimentos}">
  <f:selectItem itemValue="1" itemLabel="Macarrão" />
  <f:selectItem itemValue="2" itemLabel="Feijão" />
  <f:selectItem itemValue="3" itemLabel="Arroz" />
</h:selectManyListBox>
```

Macarrão
Feijão
Alimento: Arroz

A tag `f:selectItems` também pode ser utilizada com listboxes

Botões

Botão que envia uma requisição ao servidor (POST)

```
<h:commandButton value="Processar Pedido"
  action="#{bean.processar}" />
```

Processar Pedido

Botão que envia uma requisição ao servidor (GET)

```
<h:button value="Processar Pedido" outcome="page" />
```

Processar Pedido

Links

Link que envia uma requisição ao servidor (POST)

```
<h:commandLink action="#{bean.processar}">
  Processar Pedido
</h:commandLink>
```

Processar Pedido

Link que envia uma requisição ao servidor (GET)

```
<h:link value="Processar Pedido" outcome="page" />
```

Processar Pedido

Link que não submete dados ao servidor

```
<h:outputLink value="http://www.softblue.com.br">
  Site da Softblue
</h:outputLink>
```

Site da Softblue

Paineis

- São utilizados para organizar componentes em tabelas
- Definidos pela tag ***h:panelGrid***

Cadastro de Cliente	
Nome:	<input type="text"/>
E-mail:	<input type="text"/>
Endereço:	<input type="text"/>
Telefone:	<input type="text"/>
<input type="button" value="OK"/>	

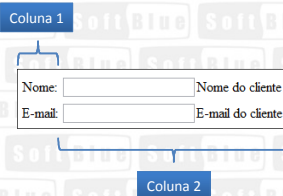
Header
Tabela com 2 colunas
Footer

Paineis

```
<h:panelGrid columns="2">
  <f:facet name="header">
    <h:outputText value="Cadastro de Cliente" />
  </f:facet>
  <h:outputText value="Nome:" />
  <h:inputText value="#{bean.nome}" />
  <h:outputText value="E-mail:" />
  <h:inputText value="#{bean.email}" />
  <h:outputText value="Endereço:" />
  <h:inputText value="#{bean.endereco}" />
  <h:outputText value="Telefone:" />
  <h:inputText value="#{bean.telefone}" />
  <f:facet name="footer">
    <h:commandButton value="OK" action="#{bean.cadastrar}" />
  </f:facet>
</h:panelGrid>
```

Agrupando Componentes

- Ao utilizar painéis, é possível agrupar componentes em uma mesma coluna
- A tag **h:panelGroup** é utilizada



Agrupando Componentes

```
<h:panelGrid columns="2">
  <h:outputText value="Nome:" />
  <h:panelGroup>
    <h:inputText value="#{bean.nome}" />
    <h:outputText value="Nome do cliente"></h:outputText>
  </h:panelGroup>
  <h:outputText value="E-mail:" />
  <h:panelGroup>
    <h:inputText value="#{bean.email}" />
    <h:outputText value="E-mail do cliente"></h:outputText>
  </h:panelGroup>
</h:panelGrid>
```

ID de um Componente

- Todo componente pode ter um ID associado
- Um componente com ID pode ser referenciado:
 - Por outros componentes JSF
 - No código Java
 - Em chamadas JavaScript

```
<h:inputText id="txtNome" value="#{bean.nome}" />
```

Renderização de Componentes

- Todo componente pode ser renderizado ou não
- Isto é controlado pelo atributo **rendered**

```
<h:inputText value="#{bean.nome}" rendered="false" />
```

O componente não
estará presente na tela

É possível utilizar value
expressions

JavaScript e Componentes JSF

- É possível integrar componentes JSF com código JavaScript

```
<h:form id="pageForm">
  <h:panelGrid columns="3">
    <h:outputText value="Nome:" />
    <h:inputText id="txtNome" value="#{bean.nome}" />
    <h:commandButton
      value="Excluir"
      action="#{bean.processar}"
      onclick="return excluir(this.form);"
    />
  </h:panelGrid>
</h:form>
```

ID do
formulário

ID da caixa
de texto

Chama uma função
JavaScript

JavaScript e Componentes JSF

```
function excluir(form) {  
    var nome = form["pageForm:txtNome"].value;  
    if (confirm("Excluir o nome " + nome + "?")) {  
        return true;  
    }  
    return false;  
}
```

O componente deve ser referenciado como `idForm:idComp`

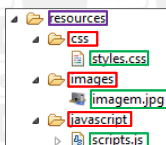
Se a tag `h:form` tiver o atributo `prependId="false"`, os componentes não recebem o ID do formulário como prefixo

Outros Componentes JSF

- O JSF possui um conjunto nativo de componentes simples
- Possibilita que novos componentes sejam criados
- Algumas bibliotecas de componentes gratuitas
 - ICEFaces
 - <http://www.icefaces.org>
 - PrimeFaces
 - <http://www.primefaces.org>
 - RichFaces
 - <http://www.jboss.org/richfaces>
 - Apache Tomahawk
 - <http://myfaces.apache.org/tomahawk>

Resources

- Recurso introduzido no JSF 2.x
- Permite referenciar recursos, que podem estar separados em bibliotecas
 - Imagens, arquivos CSS, arquivos JavaScript



Referenciando Resources

- Imagens

```
<img name="imagem.jpg" library="images" />
```

- Arquivos CSS

```
<link type="text/css" name="styles.css" library="css" />
```

- Arquivos JavaScript

```
<script name="scripts.js" library="javascript" />
```

O atributo **target** indica onde a chamada ao arquivo será colocada (**head** ou **body**)

Colocando em Prática...



Agora que você já aprendeu a teoria, acesse as vídeo-aulas práticas e pratique os assuntos abordados neste módulo!

[Clique aqui para acessar as vídeo-aulas práticas](#)
