

Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Norte
Campus Natal–Central

Diretoria Acadêmica de Gestão e Tecnologia da Informação

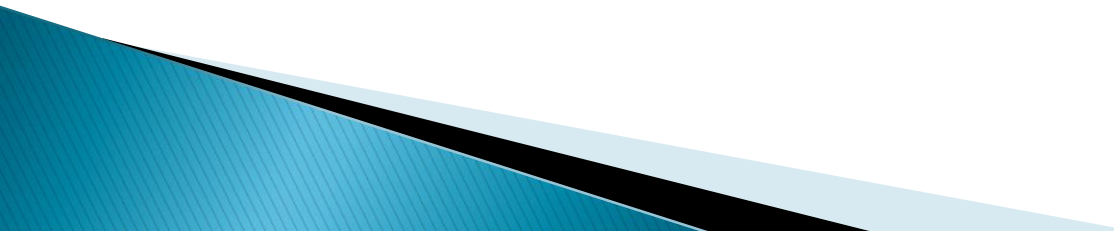
Tecnologia em Análise e Desenvolvimento de Sistemas

Java Server Faces

Componentes Web do JEE

Prof. Fellipe Aleixo (*fellipe.aleixo@ifrn.edu.br*)

O que é um Framework?

- ▶ Framework provê uma solução para uma família de problemas semelhantes
 - ▶ Utiliza conjunto de classes flexíveis e extensíveis
 - ▶ Permite construção de várias aplicações com pouco esforço
- 

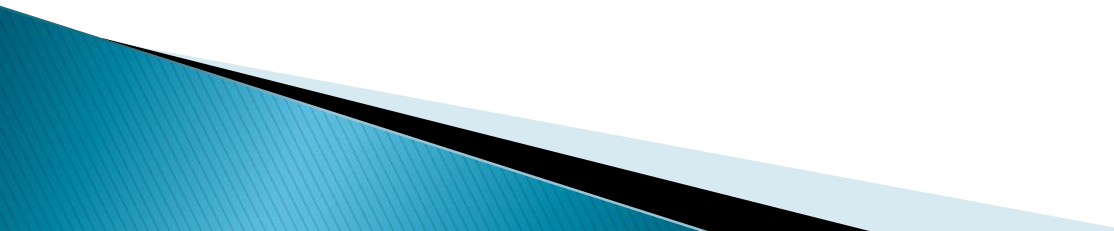
O que é o JSF?

- ▶ É um *framework* que
 - Criação de interfaces de usuário (GUI) – *server side*
 - Define um conjunto de componentes de UI
 - Permite a ligação de componentes de UI a objetos Java (*beans*)
- ▶ Analogia com Swing
- ▶ Faz parte da especificação JEE

O que é JSF?

- ▶ Um conjunto de APIs para:
 - representação de componentes de UI
 - gerenciamento do estado de tais componentes
 - manipulação de eventos
 - validação entrada
 - definição do fluxo de navegação
 - suporte à internacionalização e acessibilidade
- ▶ Uma biblioteca de *tags* customizada para páginas JSP permite a expressão dos componentes JSF em tais páginas

Elementos do JSF

- ▶ Modelo componentes
 - ▶ Conjunto componentes de UI pré-fabricados
 - ▶ Modelo programação orientado à eventos
 - ▶ Permite desenvolvedores independentes fornecerem componentes adicionais
- 

Exemplos de Componentes de UI – Padrão

Application Field Group	
☒ New Group ☐ Existing Group	
New Group	

Application Field Type	
☐ New Field Type ☒ Existing Field Type	
Name	Cell Phone Number v

Applicaition Field Display									
Select Checkboxes v	<table border="1"><thead><tr><th>Name</th><th></th></tr></thead><tbody><tr><td>One</td><td>☒</td></tr><tr><td>Two</td><td>☐</td></tr><tr><td>Three</td><td>☐</td></tr></tbody></table>	Name		One	☒	Two	☐	Three	☐
Name									
One	☒								
Two	☐								
Three	☐								

Exemplos de Componentes de UI – Customizados



JSF Messages:

Application Map:

weblogic.servlet.WebAppComponentMBean	[Caching Stub]Proxy for mydomain:ApplicationComponentMBean,ui_dir,Location=myserver,Name=accessweb-ui,Type=ApplicationComponentMBean
com.sun.faces.HTML_BASIC	com.sun.faces.renderkit.RenderKitImpl@1911e61
weblogic.servlet.WebAppComponentRuntimeMBean	weblogic.servlet.internal.WebAppRuntimeMBeanImpl,mydomain:ApplicationRuntime=myserver__apps,ui_dir,Location=myserver,Name=myserver__myserver,ServerRuntime=myserver,Type=WebAppComponentRuntimeMBean,mydomain:Location=myserver,Name=myserver__myserver,ui_dir,ServerRuntime=myserver,Type=ApplicationRuntimeMBean
com.sun.faces.ApplicationAssociate	com.sun.faces.application.ApplicationAssociate@1911e61
javax.servlet.context.tempdir	C:\bea\user_projects\domains\mydomain\myserver\ui_dir_accessweb-ui\public
com.sun.faces.OneTimeInitialization	com.sun.faces.OneTimeInitialization

Session Map:

dsa_notify	com.ihc.issa.accessweb.ui.dsa.NotificationMgrBean
javax.faces.request.charset	UTF-8
/AdminApplication.jsp	javax.faces.component.UIViewRoot@119e003
admin_template	com.ihc.issa.accessweb.ui.admin.TemplateBean

Exemplos de Componentes de UI – Código Aberto

4 - 6 of 7



ID	First Name	Family Name
4	Hans	Mueller
5	Amadeus	Mozart
6	Harry	Potter

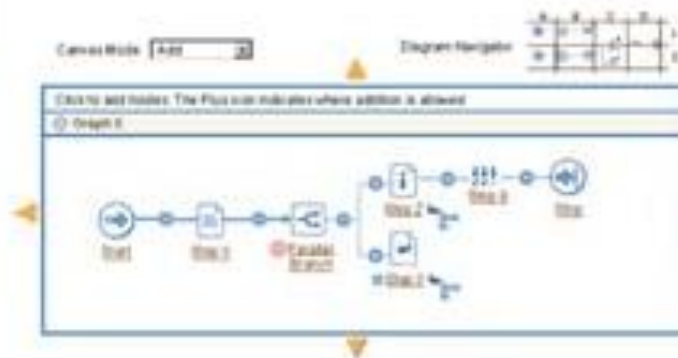


<		February 2004					>
Mon	Tues	Wed	Thur	Fri	Sat	Sun	
26	27	28	29	30	31	1	
2	3	4	5	6	7	8	
9	10	11	12	13	14	15	
16	17	18	19	20	21	22	
23	24	25	26	27	28	29	

Exemplos de Componentes de UI – Proprietários

Show All Details | Hide All Details | Previous | 1-3 of 6 | Next >

Details	Name	Age
▼ Hide	Sally	24
Detailed Information		
	Name	Sally
	Age	24
	Title	Doctor
	Patients	38
	Hospital	UCSF
▶ Show	Willy	25
▶ Show	Kelly	20



136-255-136

Root = Computer > Hardware > CDROM

Product Name that
spans 2 lines

Home

Shopping Cart

Processing: Baking the cake

Using processingLayout element.

In this demo we use the processingLayout element. It is intended to make it easier to create the BLAF Processing Template.

Cancel

- ✓ All the ingredients have been mixed.
- ✓ The oven has been preheated to 400 degrees.
- ✓ The pan has been greased and has the batter in it.
- ✓ The pan is in the oven.
- ✓ The cake is almost done.

Cancel



QuickTime™

▶ [UIX Components](#)

▼ [UIX Controller](#)

◦ [Architecture](#)

◦ [Release](#)

▶ [Architecture](#)

▶ [Release](#)



Home



Help

Finance

Switch Applications



Visited Step1

Visited Step2

Visited Step3

Active Step4

Next Step5

Next Step6

Código dos Componentes de UI

- ▶ Sintaxe para um componente de caixa de texto:

```
<h:inputText/>
```

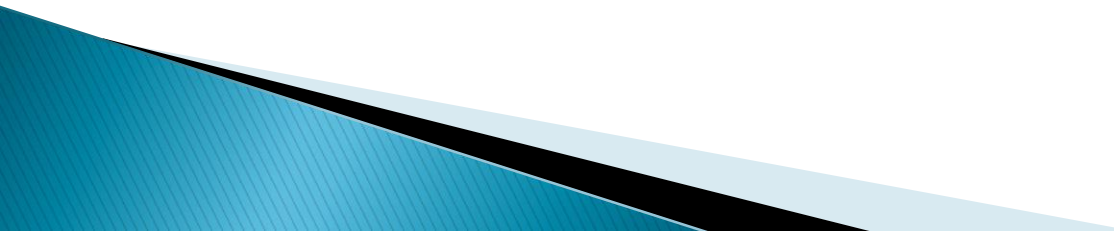
- ▶ Sintaxe para o componente de calendário:

```
<v:calendar/>
```

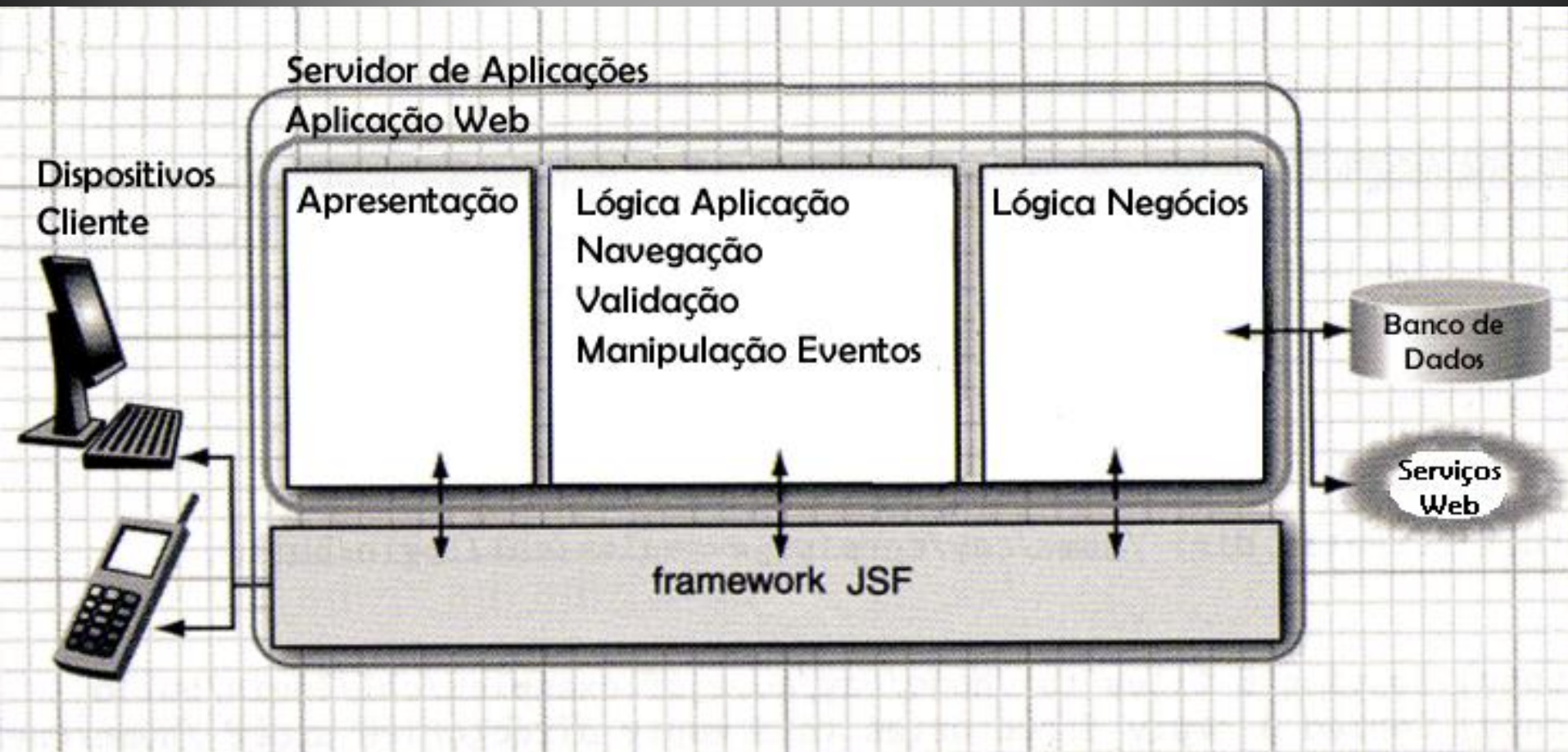
- ▶ Componente de calendário com data definida:

```
<v:calendar value="#{bean.date}"/>
```

Serviços do JSF

- ▶ Iteração com dispositivos clientes
 - ▶ União entre **apresentação visual**, **lógica da aplicação** e **lógica de negócios** de uma aplicação Web – MVC
 - ▶ Se restringe apenas à camada de apresentação
 - ▶ Persistência banco de dados, serviços Web estão fora do escopo
- 

Serviços do JSF



Serviços JSF: MVC

- ▶ Conecta visão ao modelo
- ▶ Componente de UI pode ser ligado ao Bean

```
<h:inputText value="#{curso.descricao}" />
```

- ▶ Processa eventos de ação e mudança de valores operando como controlador

```
<a4j:commandButton id="cbtGravar" immediate="false"  
    action="#{manterUsuariosController.incluirUsuario}"  
    styleClass="botoes" value="#{msg['link.gravar']}"  
    title="#{msg['link.gravar']}"  
    oncomplete="#{manterUsuariosController.exibeMensagemModal}"  
    reRender="mpAviso, panIncluiUsuarios">  
</a4j:commandButton>
```


Serviços JSF: Manipulação de Eventos

- ▶ *Binding* de métodos

```
<h:outputText value="#{bundle.welcome}"  
    rendered="#{bean.showWelcome}"/>
```

- ▶ “Ouvintes” de mudança de valores

```
<h:selectOneMenu value="#{bean.type}" onchange="submit()"  
    immediate="true" valueChangeListener="#{bean.change}">  
    <f:selectItems value="#{bean.types}">  
</h:selectOneMenu>
```

- ▶ Execução de comandos

```
<h:commandButton action="#{bean.submit}"/>
```

- ▶ “Ouvintes” de ações

```
<h:commandButton actionListener="#{bean.listen}"  
    action="#{bean.submit}"/>
```

Serviços JSF: Validação de Entrada

▶ Validadores padrão

```
<h:inputText id="card" value="#{bean.card}" required="true">  
    <f:validateLength minimum="13"/>  
</h:inputText>
```

- `f:validateDoubleRange`
- `f:validateLongRange`
- `f:validateLength`

▶ Validadores customizados

▶ Erros de validação

- `<h:message for="card"/>`
- `<h:messages/>`

Serviços JSF: Validação de Entrada

▶ Exemplo:

```
<h:inputText value="#{aluno.matricula}" required="true">  
    requiredMessage=" Informe a matricula do aluno!"  
    validatorMessage=" Matricula invalida!">  
    <f:validateLength minimum="10" />  
</h:inputText>
```


Serviços JSF: Conversores

▶ Conversores padrão

```
<h:inputText value="bean.date">  
    <f:convertDateTime pattern="MM/dd/yyyy"/>  
</h:inputText>
```

- `f:convertNumber`
- `f:convertDateTime`

▶ Conversores customizados

```
<h:inputText value="bean.user" converter="com.ihc.CustomConverter">
```

Serviços JSF: Internacionalização

► Definição do faces-config.xml

```
<application>
  <message-bundle>messages</message-bundle>
  <locale-config>
    <!-- Linguagem Português Brasil como padrão -->
    <default-locale>pt_BR</default-locale>
    <supported-locale>pt_BR</supported-locale>
    <supported-locale>en_US</supported-locale>
  </locale-config>
  <resource-bundle>
    <base-name>messages</base-name>
    <!-- Variável utilizada no jsp<!-->
    <var>msg</var>
  </resource-bundle>
</application>
```

Utilizando a Internacionalização

- ▶ Criação de um arquivo `message.properties` e descrição da seguinte maneira

```
msg.nomeSistema=MiniCurso de JavaServer Faces
```

- ▶ Utilizando a mensagem criada no componente `outputText` de um arquivo `.jspx`

```
<f:loadBundle basename="message" var="msg"/>
```

```
...
```

```
<h:outputText value="#{msg.nomeSistema}" />
```

Navegação de Páginas

- Configuração de navegação similar ao Struts

```
<navigation-rule>  
  <from-view-id> /index.jsp</from-view-id>  
  <navigation-case>  
    <from-outcome>Success</from-outcome>  
    <to-view-id> /thankyou.jsp</to-view-id>  
  </navigation-case>  
  <navigation-case>  
    <from-outcome>Error</from-outcome>  
    <to-view-id> /error.jsp</to-view-id>  
  </navigation-case>  
</navigation-rule>
```

Managed Beans

- ▶ Segunda sessão do [faces-config.xml](#)
- ▶ Serve como cola entre os componentes de UI e a lógica de negócio da aplicação

Managed Beans

- ▶ São **JavaBeans** simples (obedece algumas regras)
 - Possuem um construtor vazio (ou omite todos)
 - Não possuem atributos públicos
 - Definem métodos acessadores e modificadores públicos, segundo o padrão “getXxx” e “setXxx”
 - Se a classe possui um método “getNome” que retorna uma String, presume-se a existência de um atributo String denominado de “nome”
 - Atributos booleanos usam “isXxxx” ao invés de “getXxx”
 - Mais informações em <http://java.sun.com/beans/docs/>

Managed Beans

- ▶ *Beans* JSF também possuem métodos de ação
 - Invocados por JSF em resposta a uma ação ou evento disparado pelo usuário
 - Contêm o código que manipula o modelo de dados por trás da aplicação
 - No Struts estes são chamados de “classes de ação” e precisam ser subclasses de “Action”

Managed Beans

- ▶ Podem ser classificados em tipos (sem diferenças técnicas entre eles)
 - *Model beans* – focam no modelo de dados
 - *Backing beans* – focam na UI

Managed Beans

- ▶ Possuem quatro escopos possíveis (mesmos dos JSP/servlet)
 - **Application** – a aplicação inteira compartilha uma única instância
 - **Session** – uma nova instância é criada quando um novo usuário inicia o acesso
 - **Request** – uma nova instância é criada para cada requisição
 - Sem escopo – acessados por outros *beans* e removidos pelo “*garbage collector*” quando não são mais referenciadas por outro objetos

Managed beans

```
<managed-bean>
```

```
  <managed-bean-name>library</managed-bean-name>
```

```
  <managed-bean-class>
```

```
    com.oreilly.jent.jsf.library.model.Library</managed-bean-class>
```

```
  <managed-bean-scope>application</managed-bean-scope>
```

```
</managed-bean>
```

```
<managed-bean>
```

```
  <managed-bean-name>usersession</managed-bean-name>
```

```
  <managed-bean-class>
```

```
    com.oreilly.jent.jsf.library.session.UserSession</managed-bean-class>
```

```
  <managed-bean-scope>session</managed-bean-scope>
```

```
</managed-bean>
```

```
<managed-bean>
```

```
  <managed-bean-name>loginform</managed-bean-name>
```

```
  <managed-bean-class>
```

```
    com.oreilly.jent.jsf.library.backing.LoginForm</managed-bean-class>
```

```
  <managed-bean-scope>request</managed-bean-scope>
```

```
</managed-bean>
```

Managed beans

```
<managed-bean>  
  <managed-bean-name>addbookform</managed-bean-name>  
  <managed-bean-class>  
    com.oreilly.jent.jsf.library.backing.AddBookForm</managed-bean-class>  
  <managed-bean-scope>request</managed-bean-scope>  
</managed-bean>
```

```
<managed-bean>  
  <managed-bean-name>bookdisplayform</managed-bean-name>  
  <managed-bean-class>  
    com.oreilly.jent.jsf.library.backing.BookDisplayForm  
  </managed-bean-class>  
  <managed-bean-scope>session</managed-bean-scope>  
  <managed-property>  
    <property-name>library</property-name>  
    <value>#{library}</value>  
  </managed-property>  
</managed-bean>
```

JSPX x JSP

► Extensão JSPX

- Formato XML para criar páginas JSF (bem formado)
- Força a separação do código da camada de visão

```
<?xml version="1.0" encoding="UTF-8"?>
<jsp:root xmlns:jsp="http://java.sun.com/JSP/Page" version="2.0"
  xmlns:f="http://java.sun.com/jsf/core"
  xmlns:h="http://java.sun.com/html">
  <jsp:output omit-xml-declaration="true" doctype-root-element="HTML"
    doctype-system="http://www.w3.org/TR/html4/loose.dtd" doctype-
public="-//W3C//DTD HTML 4.01 Transitional//EN"/>
  <jsp:directive.page contentType="text/html; charset=utf_8"/>
  <f:view>
    <html>
    <body>
    <h:outputText styleClass="fontePanel"
      value="#{msg['painel.autenticacao']}" />
    </body>
    </html>
  </f:view>
</jsp:root>
```

JSPX x JSP

- ▶ Extensão JSP
 - Possibilita a utilização *scriptlets*

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
pageEncoding="ISO-8859-1"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
<%@ taglib uri="http://richfaces.org/a4j" prefix="a4j"%>
<%@ taglib uri="http://richfaces.org/rich" prefix="rich"%>
<%@ taglib uri="http://java.sun.com/jsf/html" prefix="h"%>
<%@ taglib uri="http://java.sun.com/jsf/core" prefix="f"%>
<html>
<head>
<title>Hello World RichFaces</title>
</head>
<body>
<f:view>
<form name="form" method="get" action="2.jsp">
<input type="text" name="nome">
<input type="submit">
</h:form>
</f:view>
<%
String variavel = request.getParameter("nome");
if (variavel == "minhasenha") {
    out.println("certo");
} else {
    out.println("errado");
}
%>
</body>
</html>
```

Regras de Navegação e Definição dos Beans

```
<?xml version='1.0' encoding='UTF-8'?>
<faces-config version="1.2" xmlns="http://java.sun.com/xml/ns/javaee"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
    http://java.sun.com/xml/ns/javaee/web-facesconfig_1_2.xsd">
  <lifecycle>
    <phase-listener>br.com.miniCursoJSF.utils.FaseListener</phase-listener>
  </lifecycle>
  <!-- ***** Arquivo de Mensagens com Internacionalização ***** -->
  <application>
    <message-bundle>messages</message-bundle>
    <locale-config>
      <!-- Linguagem Português Brasil como padrão -->
      <default-locale>pt_BR</default-locale>
      <supported-locale>pt_BR</supported-locale>
      <supported-locale>en_US</supported-locale>
    </locale-config>
    <resource-bundle>
      <base-name>messages</base-name>
      <!-- Variável utilizada no jsp<var>msg</var> -->
    </resource-bundle>
  </application>

  <!-- ***** Mapeamento das Classes (Beans) ***** -->
  <managed-bean>
    <description>Menu da aplicacao</description>
    <managed-bean-name>menu</managed-bean-name>
    <managed-bean-class>br.com.miniCursoJSF.utils.Menu</managed-bean-class>
    <managed-bean-scope>application</managed-bean-scope>
  </managed-bean>
```

Regras de Navegação e Definição dos Beans

```
<managed-bean>|
  <description>Mensagens</description>
  <managed-bean-name>mensagem</managed-bean-name>
  <managed-bean-class>br.com.miniCursoJSF.utils.Mensagem</managed-bean-class>
  <managed-bean-scope>session</managed-bean-scope>
</managed-bean>

<managed-bean>
  <description>Form Manter Usuarios</description>
  <managed-bean-name>manterUsuariosForm</managed-bean-name>
  <managed-bean-class>br.com.miniCursoJSF.form.ManterUsuariosForm</managed-bean-class>
  <managed-bean-scope>request</managed-bean-scope>
</managed-bean>

<managed-bean>
  <description>Controlador Manter Usuarios</description>
  <managed-bean-name>manterUsuariosController</managed-bean-name>
  <managed-bean-class>br.com.miniCursoJSF.controller.ManterUsuariosController</managed-bean-class>
  <managed-bean-scope>session</managed-bean-scope>
</managed-bean>

<!-- ***** Navegação ***** -->
<navigation-rule>
  <from-view-id>*/</from-view-id>
  <navigation-case>
    <from-outcome>paginaLogin</from-outcome>
    <to-view-id>/login.jsp</to-view-id>
    <redirect />
  </navigation-case>
</navigation-rule>
```


Regras de Navegação e Definição dos Beans

```
<navigation-rule>
  <from-view-id>/login.jsp</from-view-id>
  <navigation-case>
    <from-outcome>paginaPrincipal</from-outcome>
    <to-view-id>/index.jsp</to-view-id>
    <redirect />
  </navigation-case>
</navigation-rule>

<!-- Manter Dados Usuarios -->
<navigation-rule>
  <display-name>Página de Exibir Usuarios</display-name>
  <from-view-id>/jsp/usuario/exibirUsuarios.jsp</from-view-id>
  <navigation-case>
    <display-name>Página Principal</display-name>
    <from-outcome>cancelar</from-outcome>
    <to-view-id>/index.jsp</to-view-id>
    <redirect />
  </navigation-case>
  <navigation-case>
    <display-name>Página de Consulta de Usuarios</display-name>
    <from-outcome>consultarUsuario</from-outcome>
    <to-view-id>/jsp/usuario/consultarUsuario.jsp</to-view-id>
    <redirect />
  </navigation-case>
  <navigation-case>
    <display-name>Página de Inclusao de Usuarios</display-name>
    <from-outcome>incluirUsuario</from-outcome>
    <to-view-id>/jsp/usuario/incluirUsuario.jsp</to-view-id>
    <redirect />
  </navigation-case>
</faces-config>
```

Configuração da Aplicação no web.xml

```
<?xml version="1.0" encoding="UTF-8"?>|
<web-app version="2.5" xmlns="http://java.sun.com/xml/ns/javaee"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
    http://java.sun.com/xml/ns/javaee/web-app_2_5.xsd">
  <display-name>MiniCursoJSF</display-name>
  <description>Mini Curso de JavaServer Faces</description>

  <filter-mapping>
    <filter-name>MyFacesExtensionsFilter</filter-name>
    <servlet-name>Faces Servlet</servlet-name>
  </filter-mapping>

  <filter-mapping>
    <filter-name>MyFacesExtensionsFilter</filter-name>
    <url-pattern>/faces/myFacesExtensionResource/*</url-pattern>
  </filter-mapping>

  <context-param>
    <param-name>javax.faces.CONFIG_FILES</param-name>
    <param-value>/WEB-INF/faces-config.xml</param-value>
  </context-param>
  <context-param>
    <param-name>javax.faces.DEFAULT_SUFFIX</param-name>
    <param-value>.jspx</param-value>
  </context-param>
```

Configuração da Aplicação no web.xml

```
<context-param>
  <param-name>com.sun.faces.validateXml</param-name>
  <param-value>true</param-value>
</context-param>
<context-param>
  <param-name>javax.faces.STATE_SAVING_METHOD</param-name>
  <param-value>client</param-value>
</context-param>
<context-param>
  <param-name>org.apache.myfaces.ALLOW_JAVASCRIPT</param-name>
  <param-value>true</param-value>
</context-param>
<context-param>
  <param-name>org.apache.myfaces.PRETTY_HTML</param-name>
  <param-value>true</param-value>
</context-param>
<context-param>
  <param-name>org.apache.myfaces.DETECT_JAVASCRIPT</param-name>
  <param-value>>false</param-value>
</context-param>
<context-param>
  <param-name>org.apache.myfaces.AUTO_SCROLL</param-name>
  <param-value>true</param-value>
</context-param>
<context-param>
  <param-name>org.richfaces.SKIN</param-name>
  <param-value>orange</param-value>
</context-param>
<context-param>
  <param-name>org.richfaces.CONTROL_SKINNING</param-name>
  <param-value>enable</param-value>
</context-param>
```

Configuração da Aplicação no web.xml

```
<filter>
  <display-name>RichFaces Filter</display-name>
  <filter-name>richfaces</filter-name>
  <filter-class>org.ajax4jsf.Filter</filter-class>
</filter>
<filter-mapping>
  <filter-name>richfaces</filter-name>
  <servlet-name>Faces Servlet</servlet-name>
  <dispatcher>REQUEST</dispatcher>
  <dispatcher>FORWARD</dispatcher>
  <dispatcher>INCLUDE</dispatcher>
</filter-mapping>
<servlet>
  <servlet-name>Faces Servlet</servlet-name>
  <servlet-class>javax.faces.webapp.FacesServlet</servlet-class>
  <load-on-startup>1</load-on-startup>
</servlet>
<servlet-mapping>
  <servlet-name>Faces Servlet</servlet-name>
  <url-pattern>/app/*</url-pattern>
</servlet-mapping>

<session-config>
  <session-timeout>30</session-timeout>
</session-config>
<welcome-file-list>
  <welcome-file>app/login.jsp</welcome-file>
</welcome-file-list>
</web-app>
```

Exemplo de Componente RichFaces

```
<f:view>
  <h:form>
    <h:panelGrid columns="2">
      <h:outputText value="Type the Text:" />
      <h:inputText value="#{bean.text}" />
      <a4j:support event="onkeyup" reRender="repeater" />
    </h:panelGrid>
    <h:outputText value="Text in the AJAX Response:" />
    <h:outputText id="repeater" value="#{bean.text}" />
  </h:form>
</f:view>
```

a4j:support adds AJAX capability to existing components

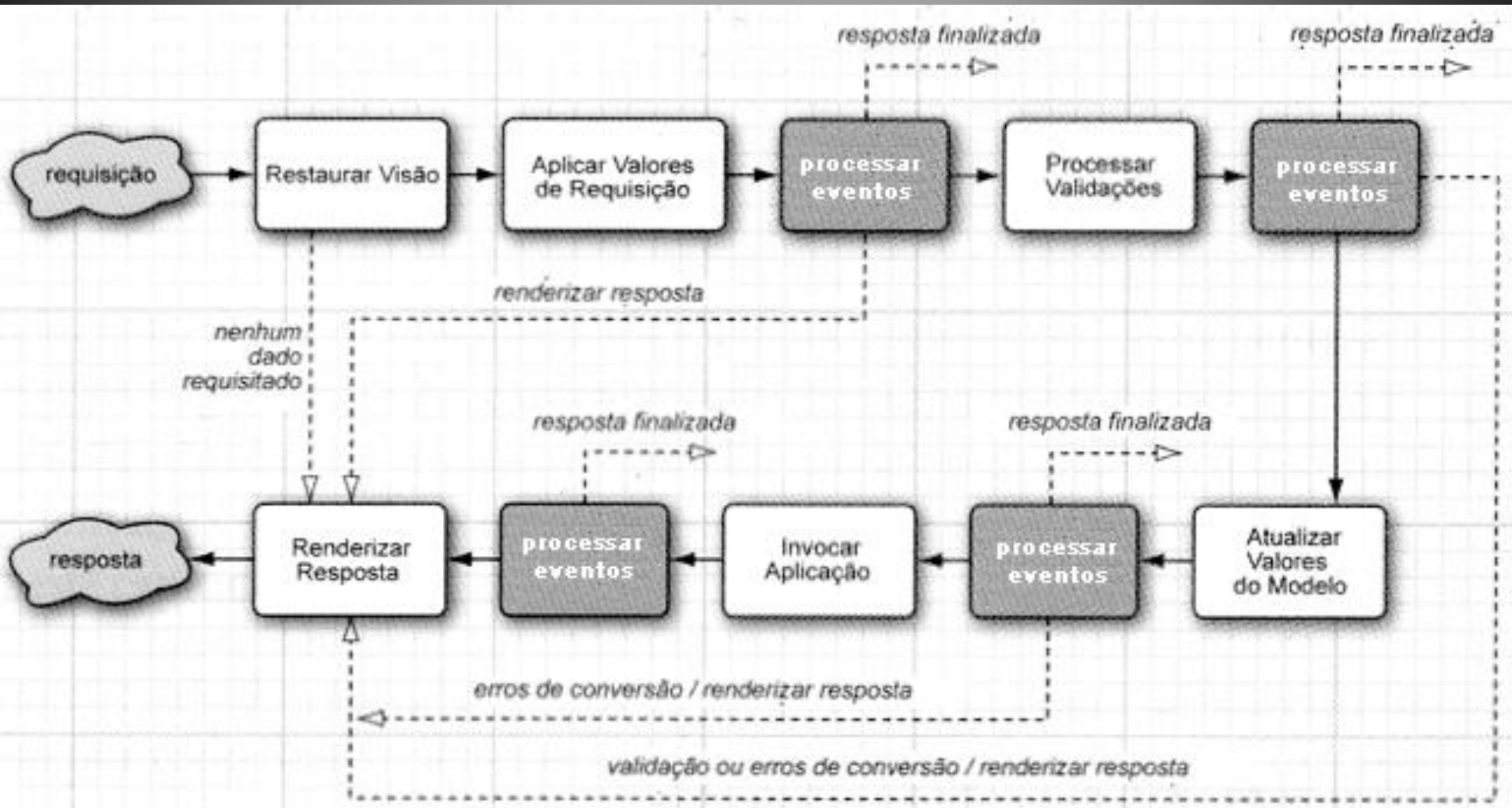
Fires AJAX request on this event

Points to component(s) to be re-rendered

Ciclo de Vida do JSF

- ▶ JSF cria a “árvore de componentes”
 - Cada elemento corresponde a um componente da interface com o usuário
- ▶ Passos para processamento de uma requisição:
 1. Recupera a visão (JSP + árvore de componentes)
 2. Mapeia os “valores” da requisição (HTTP → UI)
 3. Processa as validações
 4. Atualiza os valores do modelo
 5. Invoca a aplicação (ações solicitadas pelo usuário)
 6. “Renderiza” a resposta

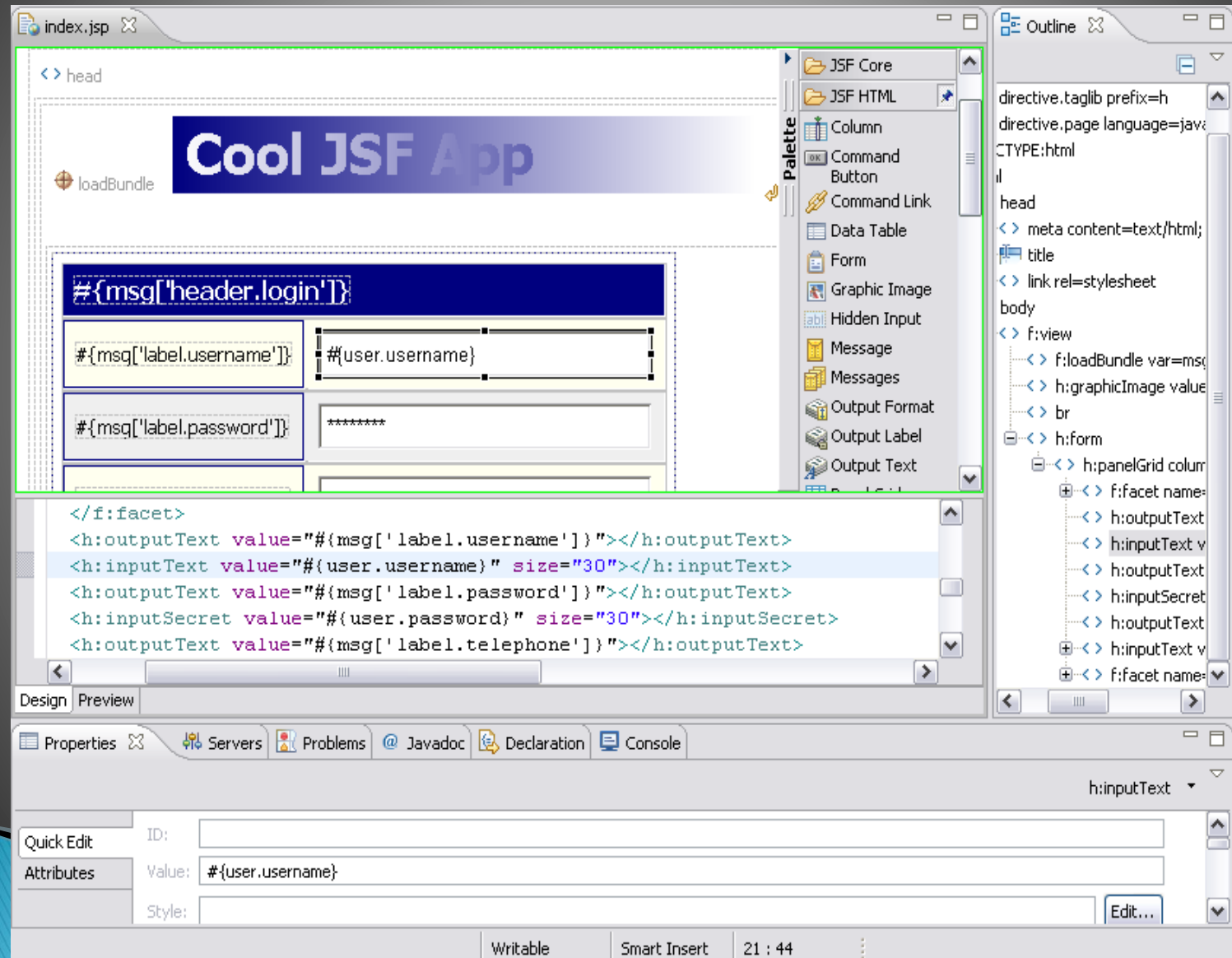
Ciclo de Vida do JSF



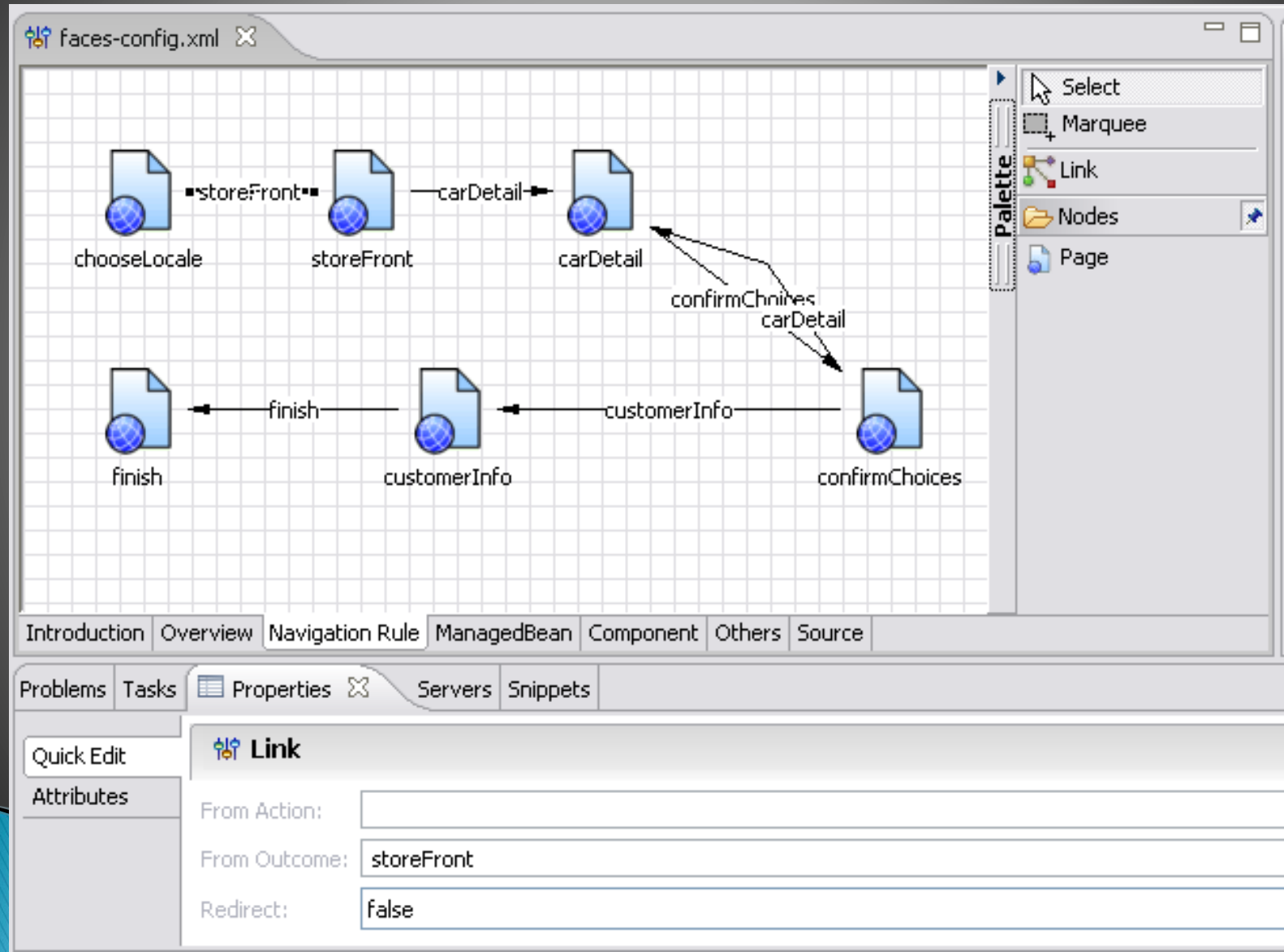
Desenvolvendo JSF em IDEs como o Eclipse

- » A maior parte do desenvolvimento se dá de forma gráfica (arrastar e soltar)

Eclipse Web Page Editor



Eclipse Faces Configuration Editor



Referências

- ▶ Tutorial JEE 7:
 - <http://docs.oracle.com/javaee/7/tutorial/doc>
- ▶ Site da Implementação oficial do *JavaServer Faces*
 - <https://javaserverfaces.java.net/>
- ▶ Componentes e recursos
 - <http://www.jsfcentral.com/>
 - <http://www.jamesholmes.com/JavaServerFaces/>
 - <http://myfaces.apache.org>
 - <http://www.jboss.org/richfaces>
 - <http://primefaces.org>
- ▶ Material UFCG (Jacques)
 - <http://www.dsc.ufcg.edu.br/~jacques/cursos/daca/html/jsf/jsf.htm>