



Grupo de Usuários Java do Ceará

cejug 

Anatomia do JSF – JavaServer Faces

Rafael Ponte

rponte@gmail.com



Agenda

- Quem sou eu?
- Desenvolvimento de Aplicações Web
- O que é JavaServer Faces?
- Por que usar JSF?
- Overview da tecnologia
- Conclusão





Quem sou eu?

- **Rafael Ponte** – SCJP 1.4 e estudando para SCWCD
- Sócio fundador da empresa de consultoria e desenvolvimento *Triadworks Agile Software Development*
- Atualmente trabalho como analista programador na IVIA
- Trabalha com desenvolvimento Java/JEE desde 2005
- Entusiasta Java e JSF
- Moderador da lista de discussão *JavaServer Faces Brazilian Group*
- Técnico em Desenvolvimento de Software pelo CEFET-CE
- Ex-aluno frustrado do curso de bacharelado em Matemática da UFC 😊



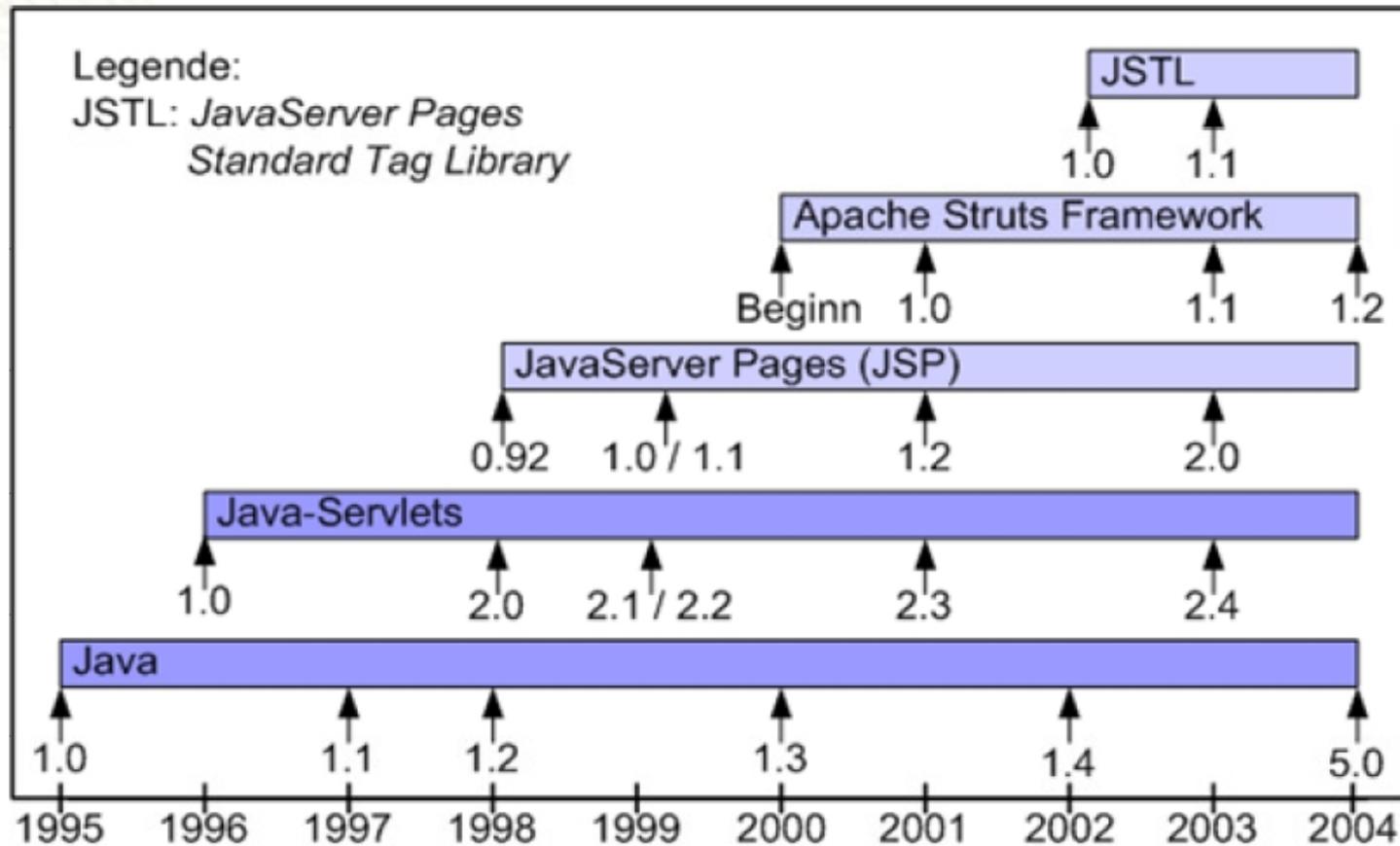
Desenvolvimento de Aplicações Web



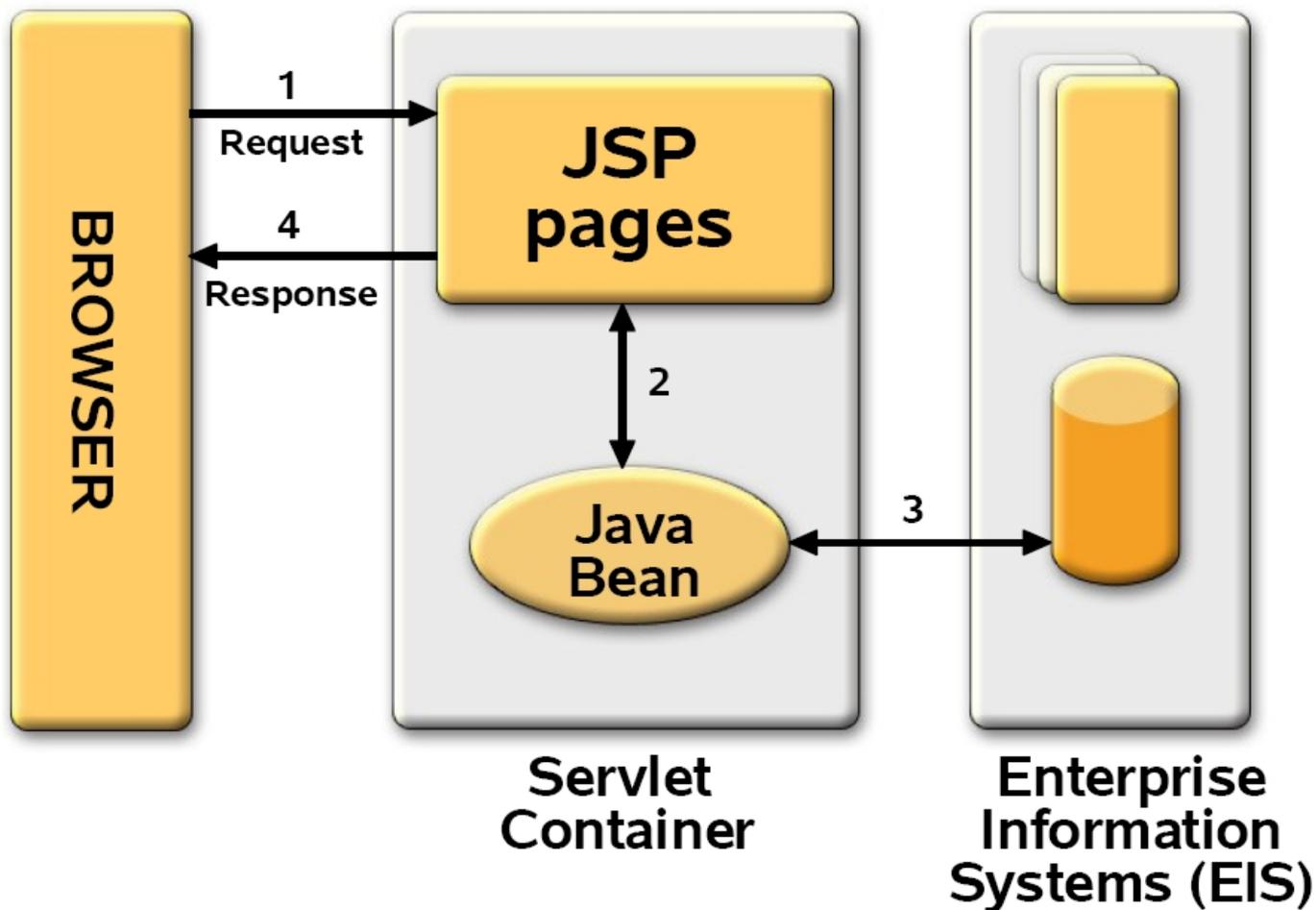
Desenvolvimento Web (geralmente)

- Aplicações web tornam-se mais e mais importantes
- Mais e mais complexidade
 - Ajax, validação (server vs. client), ...
- Exigência mais elevada do cliente durante os anos
 - Interfaces ricas (fácil de usar)
 - Prazos cada vez mais curtos
- Desenvolvimento considerado difícil
- A bola da vez..

Desenvolvimento Web (Java)

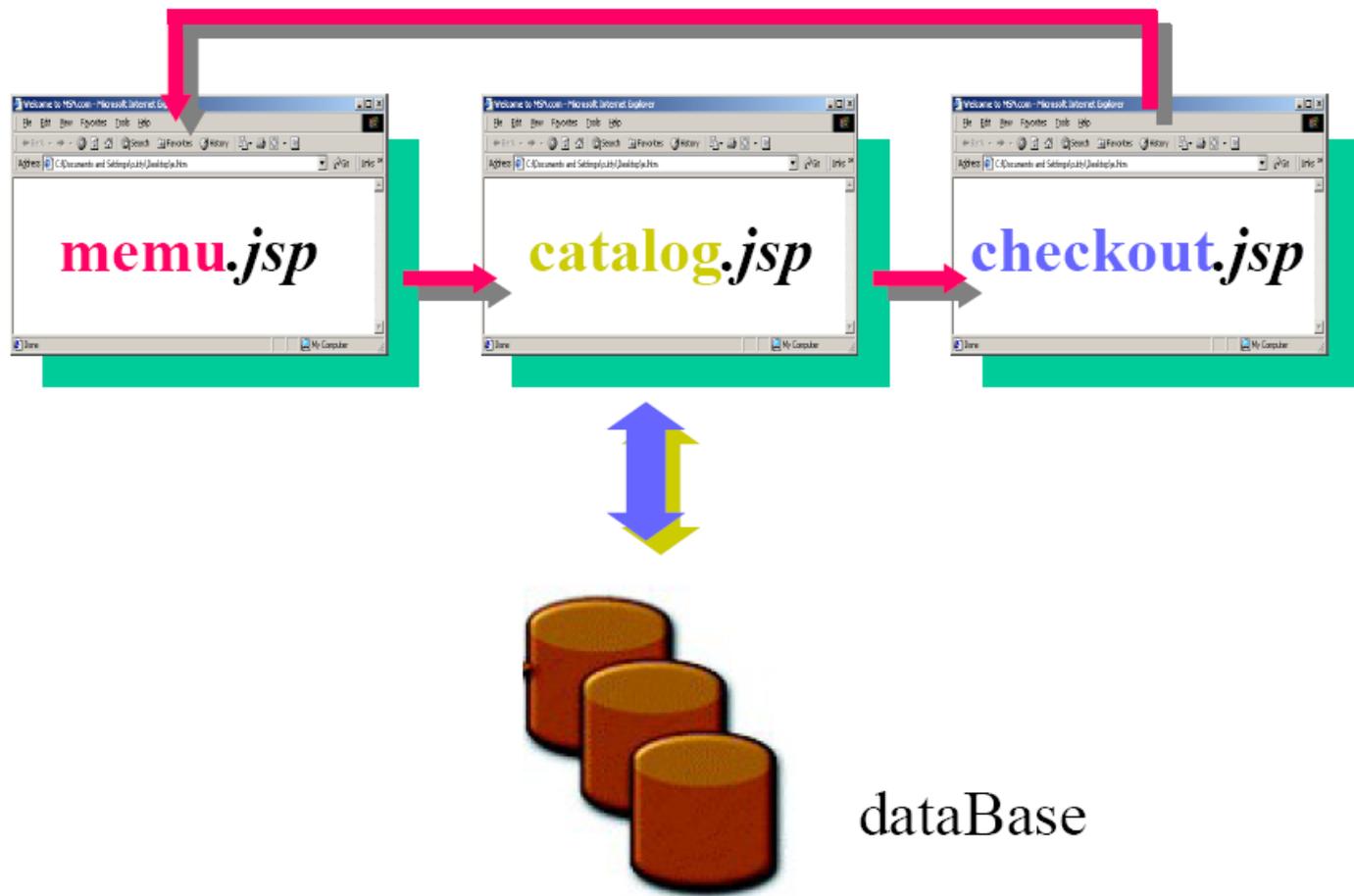


Model 1 (centrado em páginas)



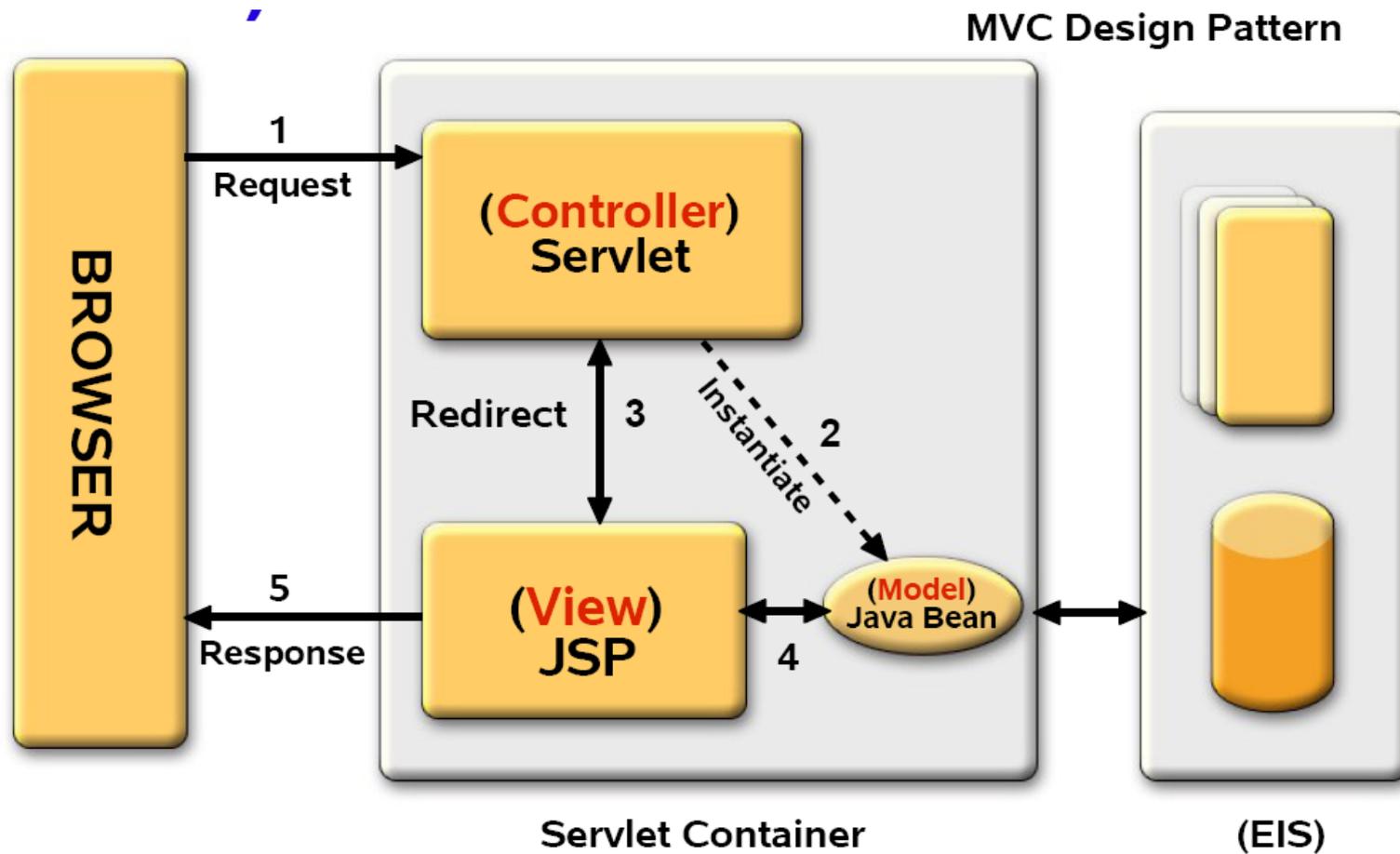
Introdução

Model 1 (Arquitetura centrada em páginas)



Introdução

Model 2 (MVC)



Surgiram diversos frameworks MVC baseado no Model 2

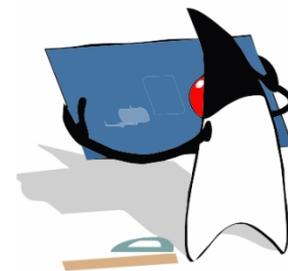
- Apache Struts
- WebWork (Struts2)
- Spring MVC
- Tapestry
- E muito outros..
 - E continuam surgindo mais

Problemas

- Java não nos fornece o bastante para desenvolvermos aplicações web com produtividade
- Dificuldade para integrar vários frameworks (algumas vez nem é possível)
- Falta de padrão para um framework web
- Ausência de uma API padrão para construir componentes web java
- E a solução?



JavaServer Faces! ☺





O que é JavaServer Faces?



O que é JavaServer Faces?

JavaServer Faces é...

- ... uma especificação
- ... um framework baseado em componentes de interface com o usuário
- ... um framework dirigido à eventos
- ... padrão de mercado
- ... RAD
- ... um framework MVC

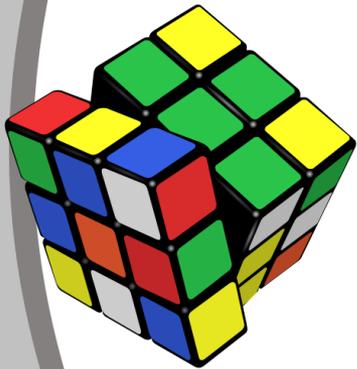


O que é JavaServer Faces?

JavaServer Faces é...



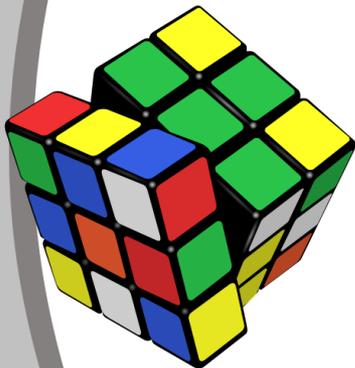
- ... uma especificação
 - JSR 127 (JSF 1.0 e JSF 1.1) – 2004
 - JSF 1.0 – não alcançou o sucesso esperado
 - JSF 1.1 – corrigiu os erros da versão anterior
 - JSR 252 (JSF 1.2) - 2006 (Java EE 5.0)
 - JSF 1.2 – melhor compatibilidade com JSP 2.1 e correção de bugs
 - JSR 314 (JSF 2.0)
 - Arquitetura, produtividade, ferramentas RAD, AJAX, melhores componentes UI..
 - A pergunta é.. quando.. 2007, 2008 ? (Java EE 6.0 ?)



O que é JavaServer Faces?

JavaServer Faces é...

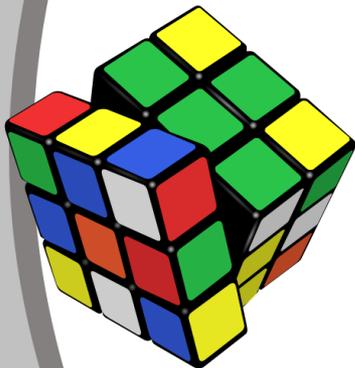
- ... um framework baseado em componentes de interface com o usuário
 - JSF é construído sob o conceito de componentes
 - Os componentes são a base para a construção da interface com o usuário
 - Existem componentes padrões equivalentes a quase todas as tags HTML e diversos componentes auxiliares
 - Existem diversas bibliotecas de componentes disponíveis comerciais e open source
 - Os componentes são expressos através de tags em um JSP e posteriormente transformados em HTML



O que é JavaServer Faces?

JavaServer Faces é...

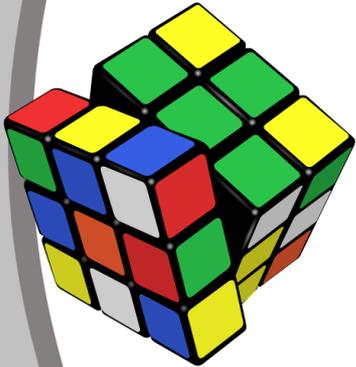
- ... um framework dirigido à eventos
 - Possui um modelo para manipulação de eventos
 - Componentes geram determinados eventos
 - Faz chamada à um método (ação no servidor)
 - O estado da aplicação web muda devido ao evento disparado pelo usuário
 - Um evento pode ser um click em um botão, focar sobre um input ou mesmo alterar o valor de uma combo-box



O que é JavaServer Faces?

JavaServer Faces é...

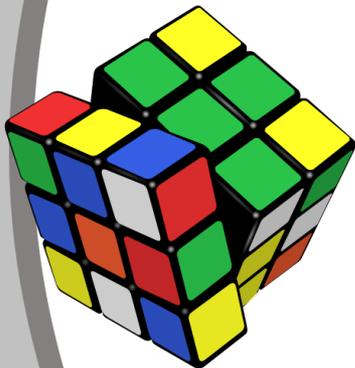
- ... padrão de mercado
 - É uma especificação (JCP)
 - Java EE 5.0
 - Suporte de Big Players
 - Sun, Apache, IBM, Oracle, BEA Systems, Borland, ...
 - IDEs (NetBeans, Eclipse, MyEclipse, JDeveloper, ...)
 - Implementações (Sun (R.I), Apache, IBM, ...)
 - Conjuntos de componentes (ADF Faces, Apache MyFaces, WoodStock, IceFaces, JBoss RichFaces, ...)



O que é JavaServer Faces?

JavaServer Faces é...

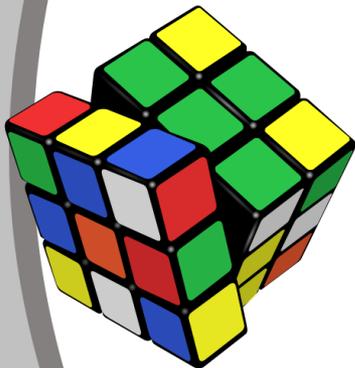
- ... RAD (**R**apid **A**pplication **D**evelopment)
- 4 camadas:
 - Arquitetura de componentes básica
 - Conjunto de componentes padrão
 - Infra-estrutura da aplicação
 - Ferramenta RAD
- JSF padroniza os primeiros três pontos e permite a criação de ferramentas RAD
- Utiliza-se IDEs *drag-and-drop* para construir aplicações:
 - Sun Studio Creator, NetBeans, JDeveloper etc



O que é JavaServer Faces?

JavaServer Faces é...

- ... um framework MVC para construir interfaces com o usuário para aplicações web. JSF inclui:
 - Um conjunto de APIs para:
 - representar os componentes UI e gerenciar o estado destes componentes
 - manipular os eventos e validação de entrada do usuário
 - definir e controlar a navegação entre as páginas
 - dar suporte a internacionalização e acessibilidade
 - Duas bibliotecas de tags JavaServer Pages (JSP) para expressar uma interface JavaServer Faces usando uma página JSP





Por que usar JSF?





Por que usar JSF?

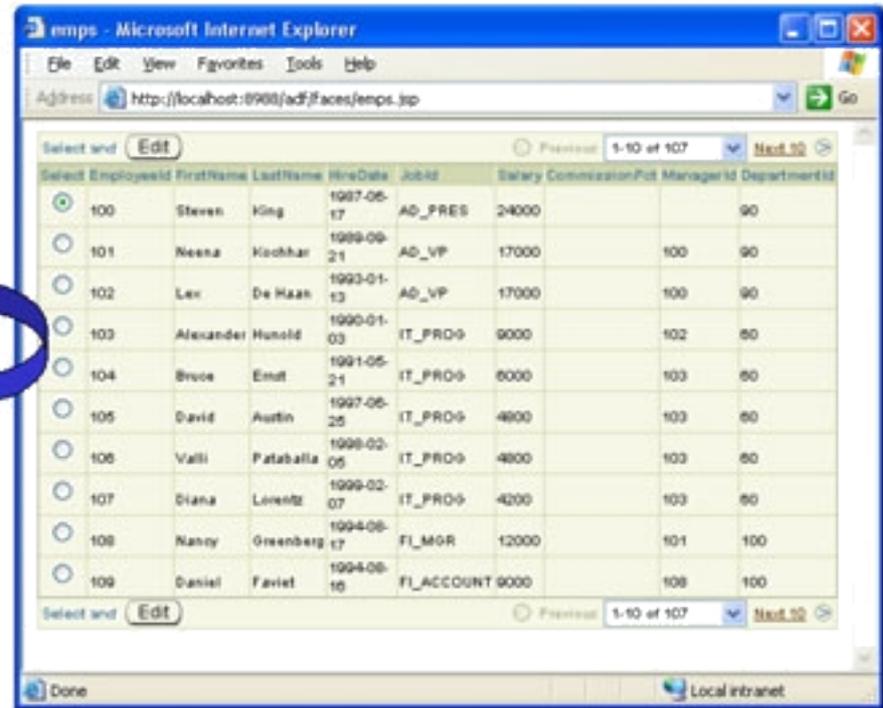
- É um padrão de mercado
- Presente na especificação JEE5.0 e em breve JEE6.0
- Suporte de Big Players
- Fácil de usar 😊 E produtivo também!
 - Fornece separação de funções que envolvem a construção de aplicações Web
 - Alta abstração para desenvolvimento de aplicação web
 - Modelo de programação dirigido a eventos (oposto ao modelo HTTP request/response)
 - Baixa curva de aprendizagem da equipe
 - Vários pontos de extensão (converters, validators, listeners etc)
 - Componentes já prontos!

Por que usar JSF?

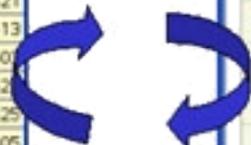
- Reusabilidade e estrutura de componentes extensível
- Acessibilidade (browsers, celulares, pda's, ...)



Wireless Client



HTML Browser





Por que usar JSF?

- Suporte cross-browser ☺
- Segurança
 - OWASP (Open Web Application Security Project)
 - cross-site scripting, session hijacking, execução de métodos, ...
 - Natureza orientada a componentes, validação e pontos de extensão
- Grande atividade da comunidade em fóruns, listas de discussão, blogs, portais, revistas, livros etc
- Integração com diversos frameworks
 - Spring, Spring-Annotation, Spring-Webflow, Hibernate, EJB3, JPA, JBoss Seam, Acegi, JAAS, Struts, Struts2, Shale, Tiles, Facelets, Sitemesh, DWR, EXT, Crank, Beehive-Webflow, GWT, Portlets e muitos outros

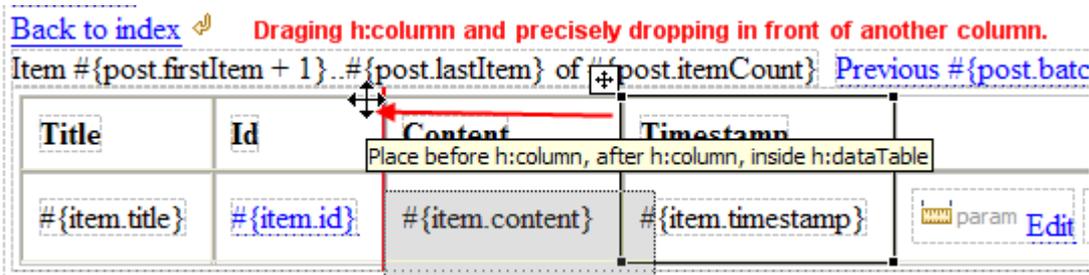
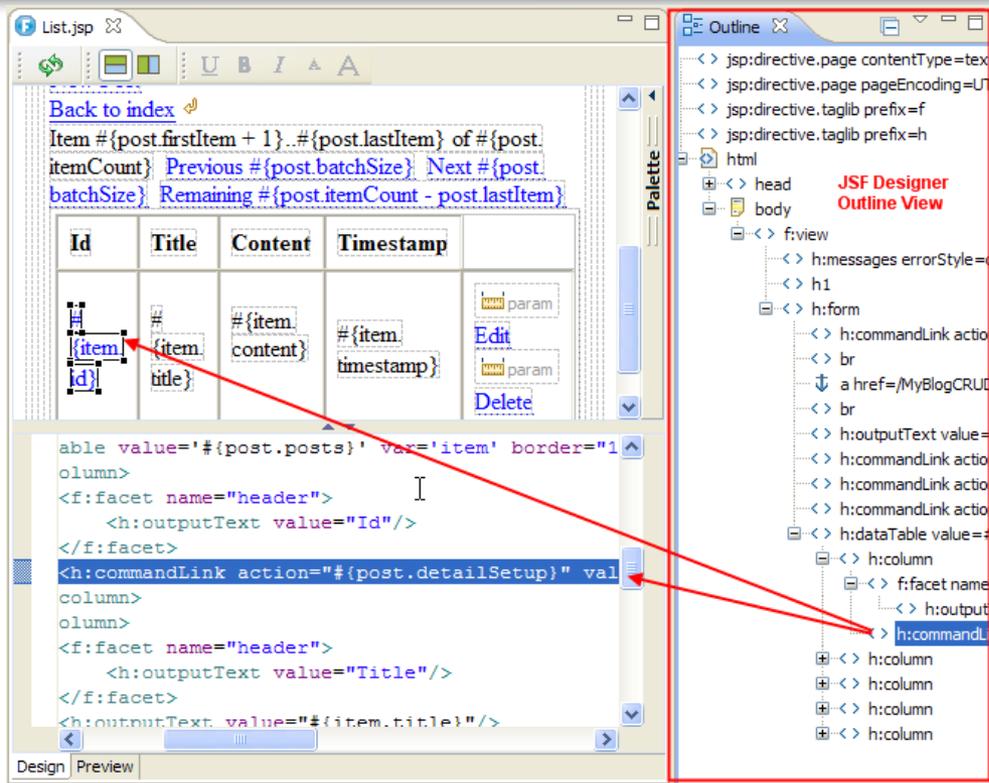
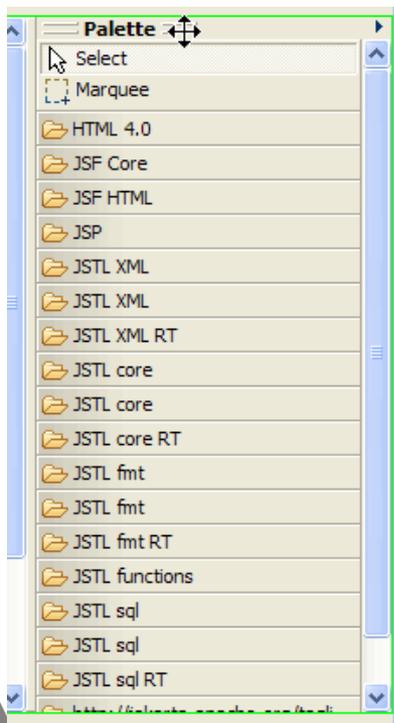


Por que usar JSF?

- Testes unitários
 - Mock objects
 - Shale testing framework
- Templates
 - Tiles e Sitemesh
 - Facelets
- IDEs e WYSIWYG (What you see is what you get)
 - Eclipse (**MyEclipse**, Europa etc)
 - **NetBeans (Visual WebPack)**
 - Sun Java Studio Creator
 - Oracle JDeveloper
 - **JSFToolbox (Dreamweaver)**
 - IntelliJ IDEA
 - JBuilder
 - IBM Rational App/Web Developer
 - Red Hat Developer Studio (antigo Exadel Studio)

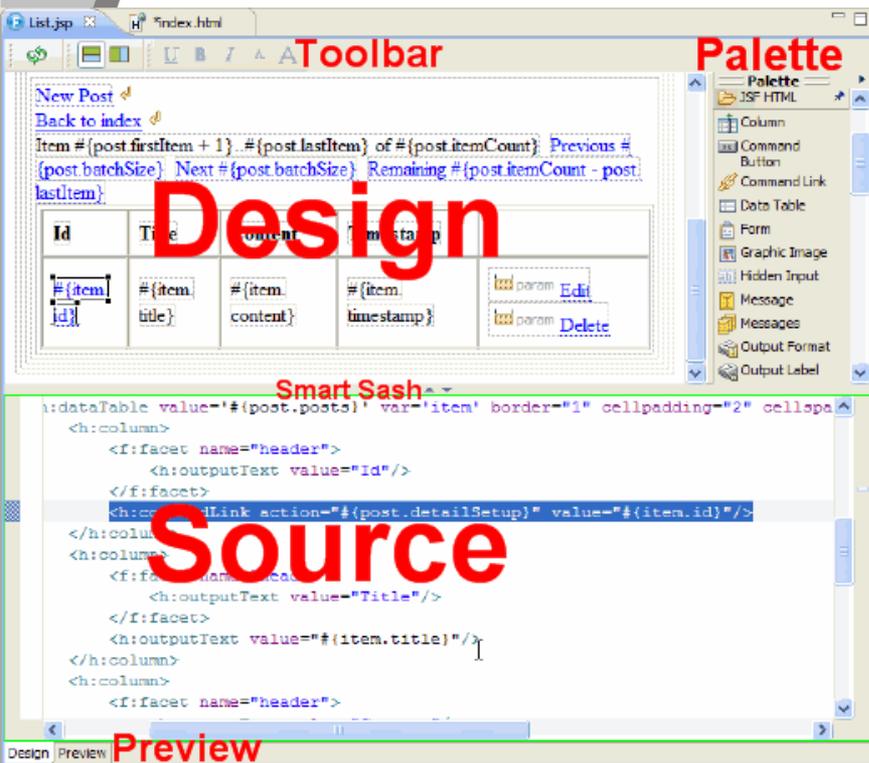
Por que usar JSF?

MyEclipse (1)



Por que usar JSF?

MyEclipse (2)



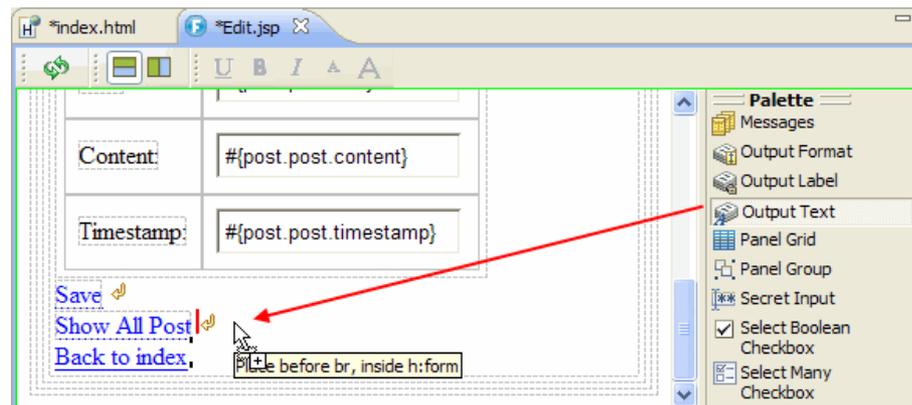
Design

Source

```

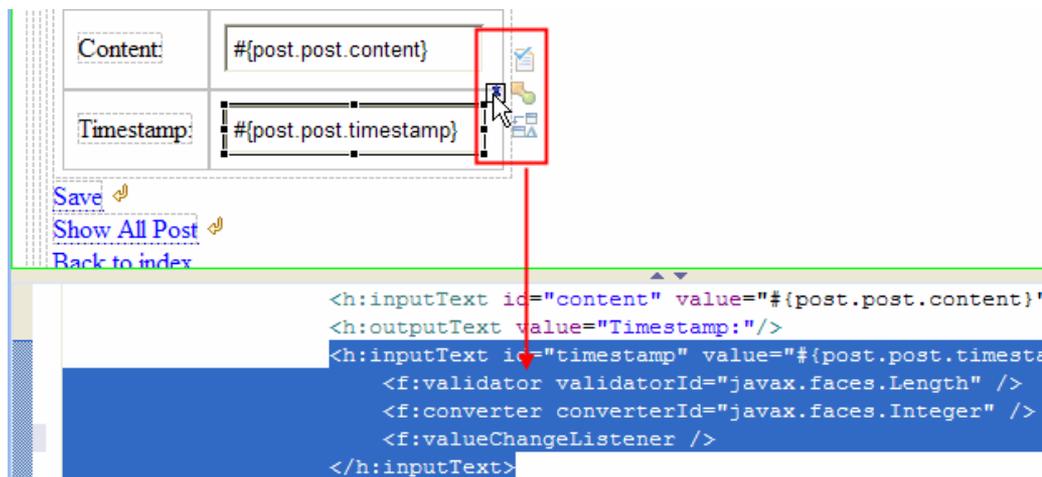
<table border="1" cellpadding="2" cellspacing="1">
  <thead>
    <tr>
      <th>Id</th>
      <th>Title</th>
      <th>Content</th>
      <th>Timestamp</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td>#{item.id}</td>
      <td>#{item.title}</td>
      <td>#{item.content}</td>
      <td>#{item.timestamp}</td>
      <td>
        <input type="button" value="Edit" />
        <input type="button" value="Delete" />
      </td>
    </tr>
  </tbody>
</table>

```



Design

Place before br, inside h:form



Source

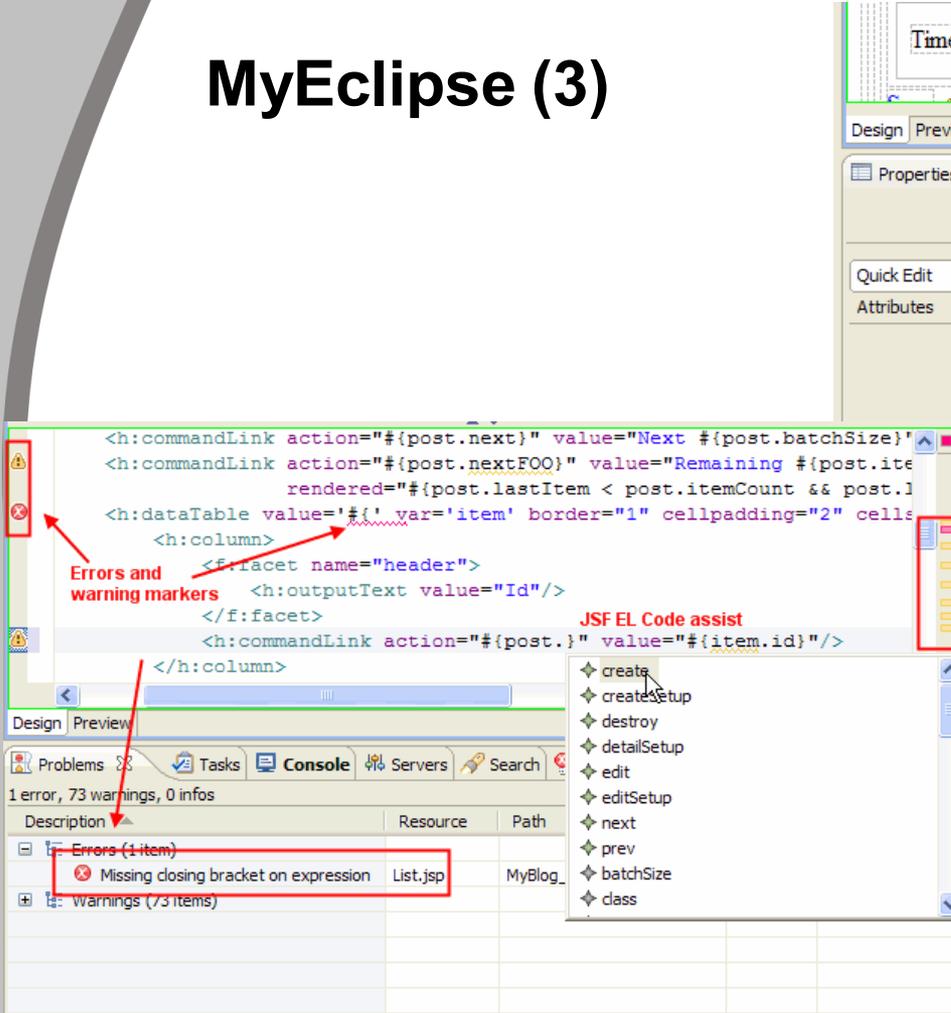
```

<h:inputText id="content" value="#{post.post.content}"/>
<h:outputText value="Timestamp:" />
<h:inputText id="timestamp" value="#{post.post.timestamp}"/>
<f:validator validatorId="javax.faces.Length" />
<f:converter converterId="javax.faces.Integer" />
<f:valueChangeListener />
</h:inputText>

```

Por que usar JSF?

MyEclipse (3)

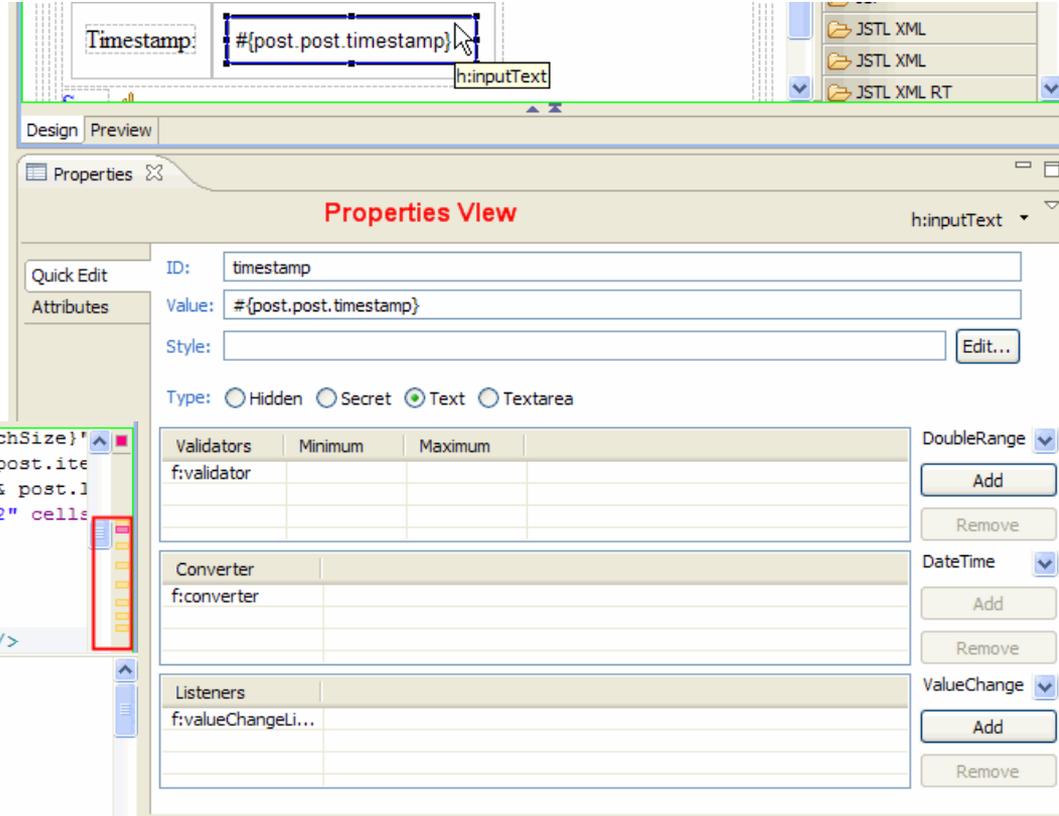


The screenshot shows the MyEclipse IDE with a JSP file open. The code contains several JSF tags. A red box highlights error and warning markers on the left side of the editor. A red arrow points to a specific error in the Problems view: "Missing closing bracket on expression" in the file "List.jsp" at resource "MyBlog...". A JSF EL Code assist popup is visible, showing a list of methods: create, createSetup, destroy, detailSetup, edit, editSetup, next, prev, batchSize, and class.

```

<h:commandLink action="#{post.next}" value="Next #{post.batchSize}"
<h:commandLink action="#{post.nextFOO}" value="Remaining #{post.item
    rendered="#{post.lastItem < post.itemCount && post.l
<h:dataTable value="#{var='item' border="1" cellpadding="2" cells
    <h:column>
        <f:facet name="header">
            <h:outputText value="Id"/>
        </f:facet>
        <h:commandLink action="#{post.}" value="#{item.id}"/>
    </h:column>

```



The screenshot shows the Properties View for an `h:inputText` component. The ID is "timestamp" and the Value is `#{post.post.timestamp}`. The Type is set to "Text". The view includes sections for Validators, Converter, and Listeners, each with an "Add" button.

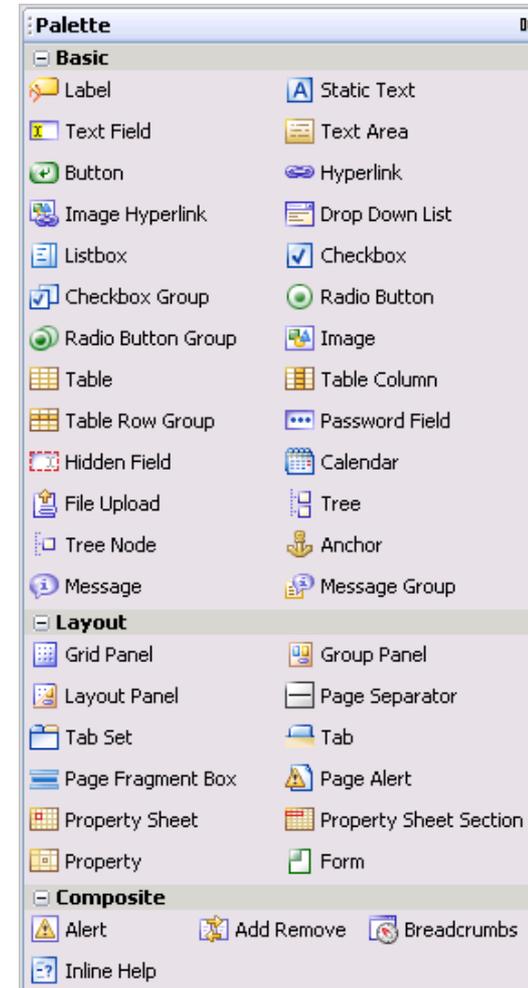
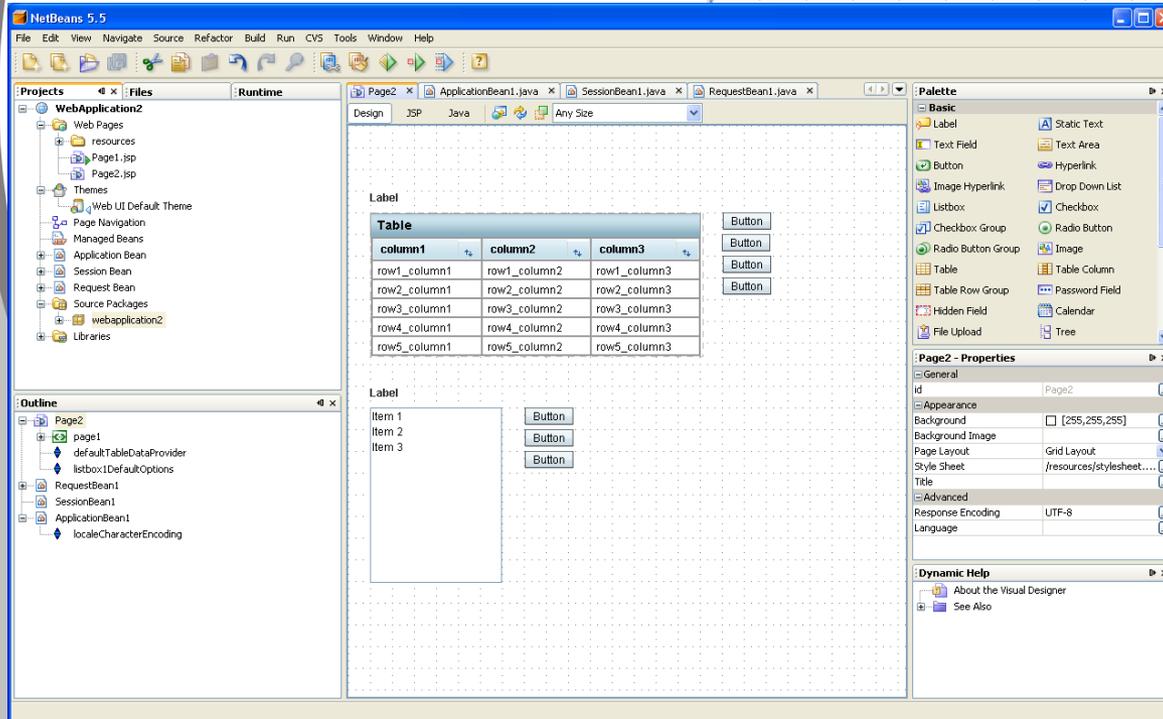
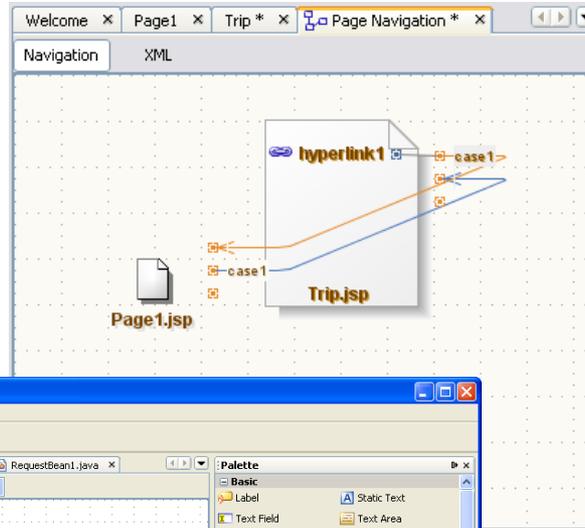
Validators	Minimum	Maximum
f:validator		

Converter	Minimum	Maximum
f:converter		

Listeners	ValueChange
f:valueChangeli...	

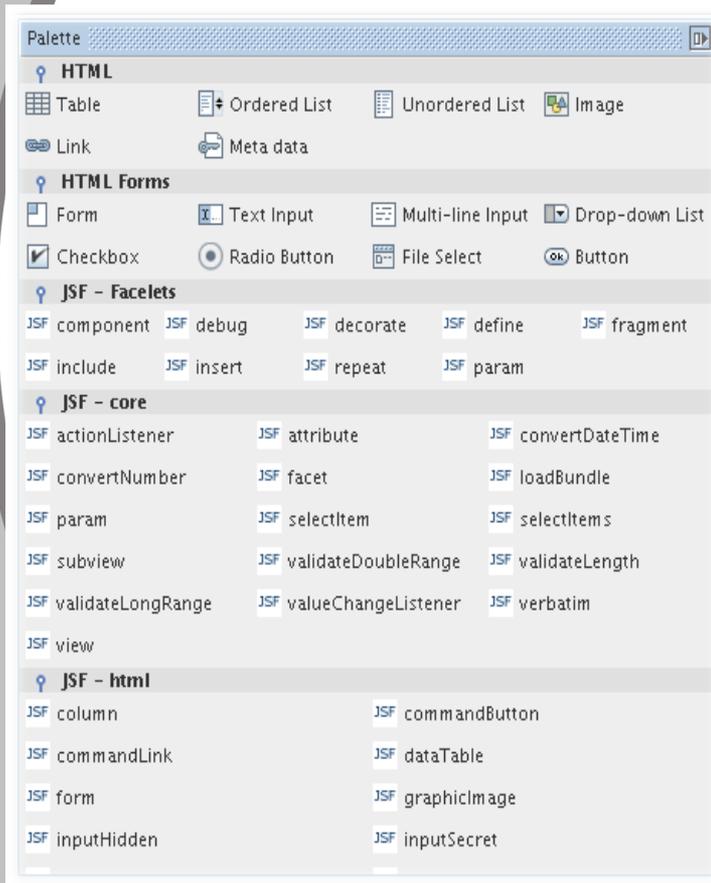
Por que usar JSF?

NetBeans (1)



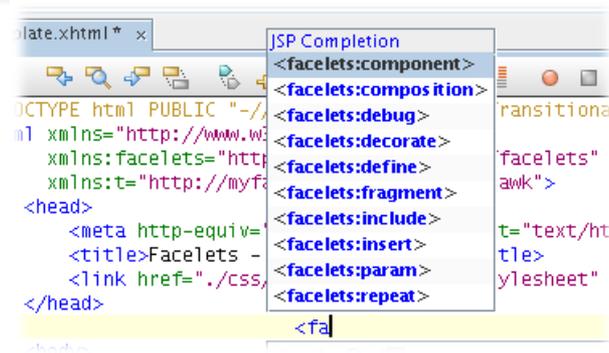
Por que usar JSF?

NetBeans (2)

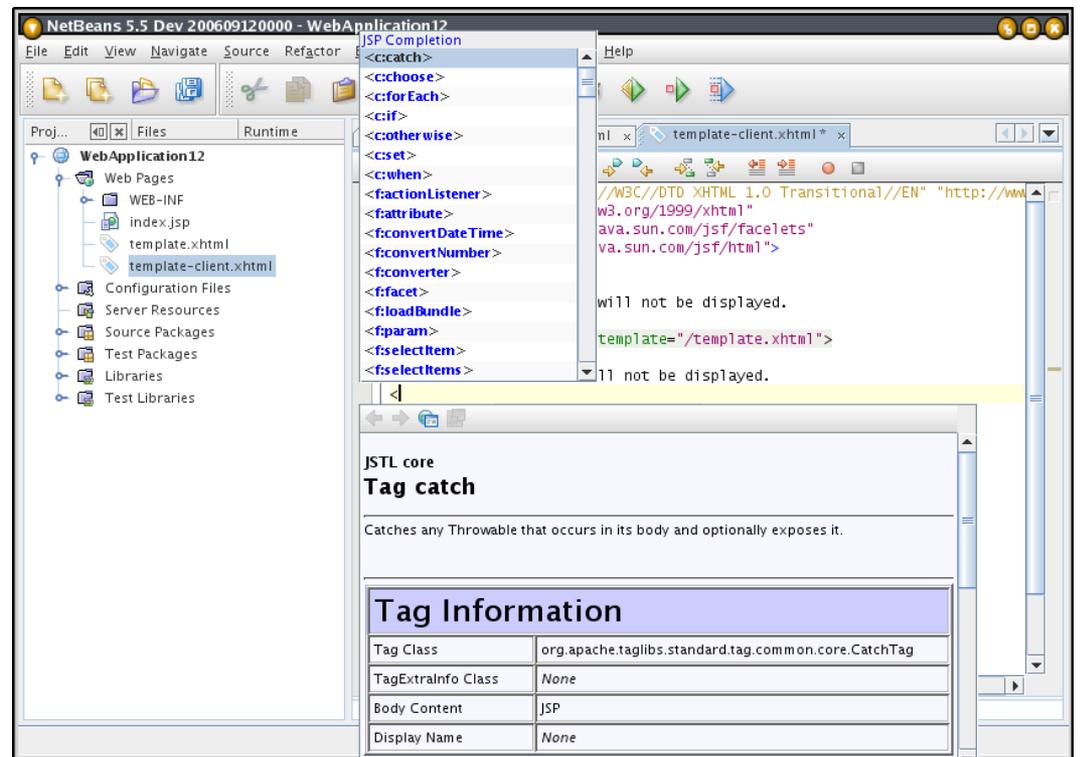


The image shows the NetBeans Palette with several categories of components:

- HTML**
 - Table
 - Ordered List
 - Unordered List
 - Image
 - Link
 - Meta data
- HTML Forms**
 - Form
 - Text Input
 - Multi-line Input
 - Drop-down List
 - Checkbox
 - Radio Button
 - File Select
 - OK Button
- JSF - Facelets**
 - JSF component
 - JSF debug
 - JSF decorate
 - JSF define
 - JSF fragment
 - JSF include
 - JSF insert
 - JSF repeat
 - JSF param
- JSF - core**
 - JSF actionListener
 - JSF attribute
 - JSF convertDateTime
 - JSF convertNumber
 - JSF facet
 - JSF loadBundle
 - JSF param
 - JSF selectItem
 - JSF selectItems
 - JSF subview
 - JSF validateDoubleRange
 - JSF validateLength
 - JSF validateLongRange
 - JSF valueChangeListener
 - JSF verbatim
 - JSF view
- JSF - html**
 - JSF column
 - JSF commandButton
 - JSF commandLink
 - JSF dataTable
 - JSF form
 - JSF graphicImage
 - JSF inputHidden
 - JSF inputSecret



A close-up of the JSP Completion popup in NetBeans. It shows a list of facelets tags: <facelets:component>, <facelets:composition>, <facelets:debug>, <facelets:decorate>, <facelets:define>, <facelets:fragment>, <facelets:include>, <facelets:insert>, <facelets:param>, and <facelets:repeat>. The background shows a snippet of XML code for a facelet component.



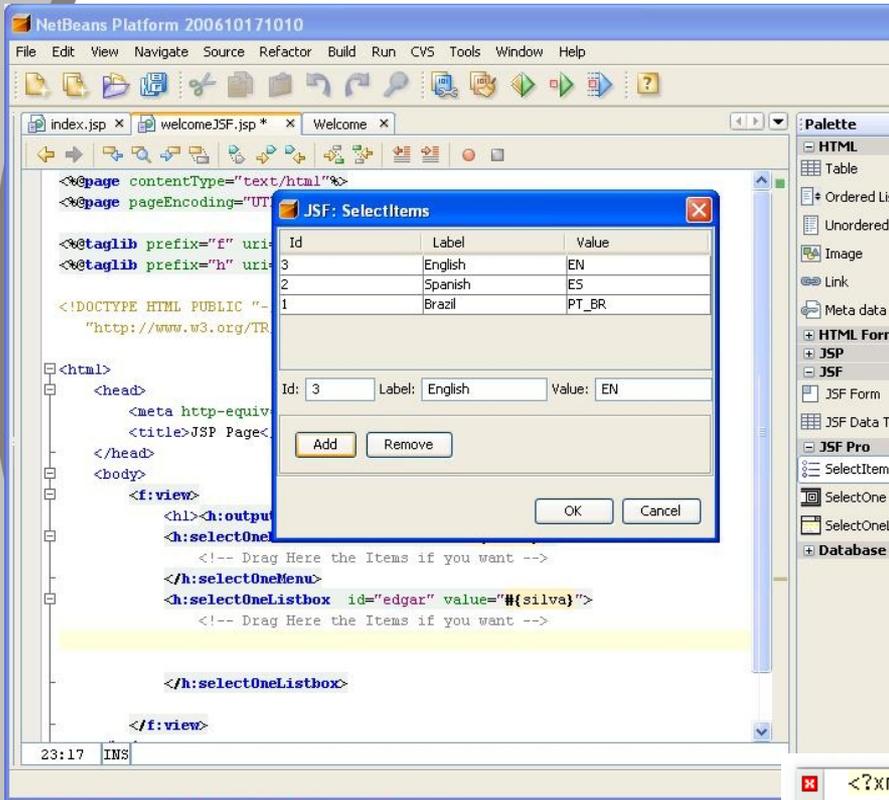
The screenshot shows the NetBeans IDE with the following elements:

- Project Explorer:** Shows a project named 'WebApplication12' with a 'Web Pages' folder containing 'index.jsp', 'template.xhtml', and 'template-client.xhtml'.
- Source Editor:** Displays the 'template-client.xhtml' file with a JSP completion popup for the <ccatch> tag. The code includes various JSF tags like <ccatch>, <ccchoose>, <cfor Each>, <cif>, <cother wise>, <cset>, <cwhen>, <factionListener>, <fattribute>, <fconvertDate Time>, <fconvertNumber>, <fconverter>, <ffacet>, <floadBundle>, <fparam>, <fselectItem>, and <fselectItems>.
- Help Window:** Shows the documentation for the 'JSTL core Tag catch' tag, which catches any Throwable that occurs in its body and optionally exposes it.
- Tag Information Table:**

Tag Information	
Tag Class	org.apache.taglibs.standard.tag.common.core.CatchTag
TagExtraInfo Class	None
Body Content	JSP
Display Name	None

Por que usar JSF?

NetBeans (3)



NetBeans Platform 200610171010

File Edit View Navigate Source Refactor Build Run CVS Tools Window Help

index.jsp x welcomeJSF.jsp * x Welcome x

JSF: SelectItems

Id	Label	Value
3	English	EN
2	Spanish	ES
1	Brazil	PT_BR

Id: 3 Label: English Value: EN

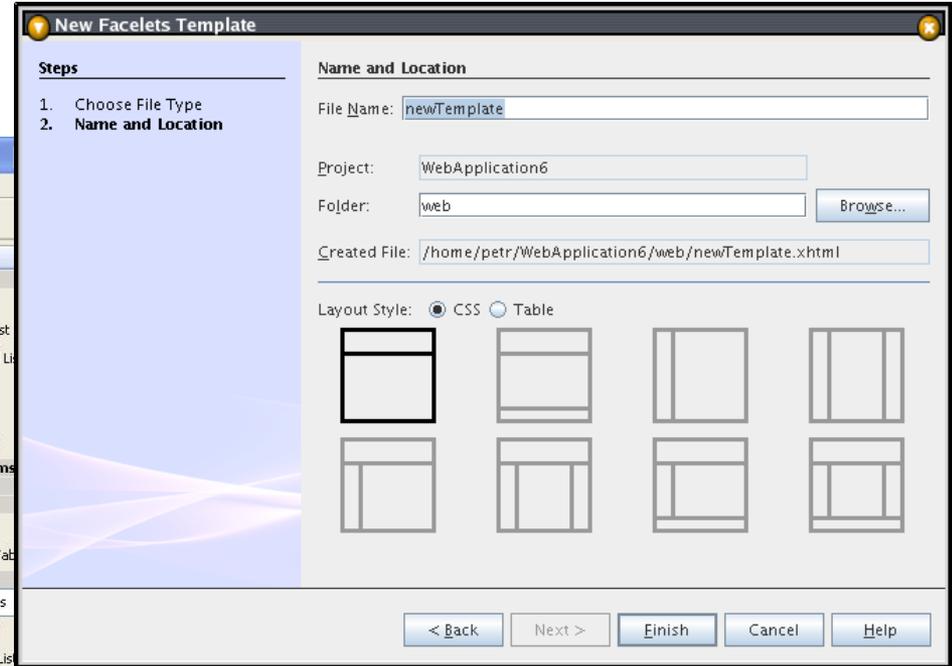
Add Remove

OK Cancel

```

<?xml version='1.0' encoding='UTF-8' ?>
<@page contentType="text/html" />
<@page pageEncoding="UTF-8" />
<@taglib prefix="f" uri="http://java.sun.com/jsf/core" />
<@taglib prefix="h" uri="http://java.sun.com/jsf/html" />
<!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" />
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
<title>JSP Page</title>
</head>
<body>
<f:view>
<h:outputText value="Welcome" />
<h:selectOneMenu value="Language" />
<!-- Drag Here the Items if you want -->
</h:selectOneMenu>
<h:selectOneListbox id="edgar" value="#{silva}" />
<!-- Drag Here the Items if you want -->
</h:selectOneListbox>
</f:view>

```



New Facelets Template

Steps

- Choose File Type
- Name and Location

Name and Location

File Name: newTemplate

Project: WebApplication6

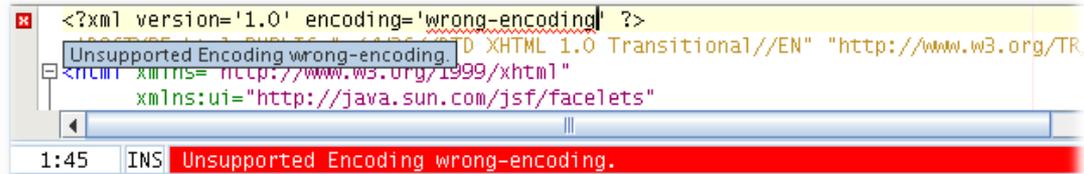
Folder: web Browse...

Created File: /home/petr/WebApplication6/web/newTemplate.xhtml

Layout Style: CSS Table

Layout templates: 8 icons showing different grid and table structures.

< Back Next > Finish Cancel Help



```

<?xml version='1.0' encoding='wrong-encoding' ?>
<html xmlns="http://www.w3.org/1999/xhtml" />
<h:selectOneMenu value="Language" />
<!-- Drag Here the Items if you want -->
</h:selectOneMenu>
<h:selectOneListbox id="edgar" value="#{silva}" />
<!-- Drag Here the Items if you want -->
</h:selectOneListbox>
</f:view>

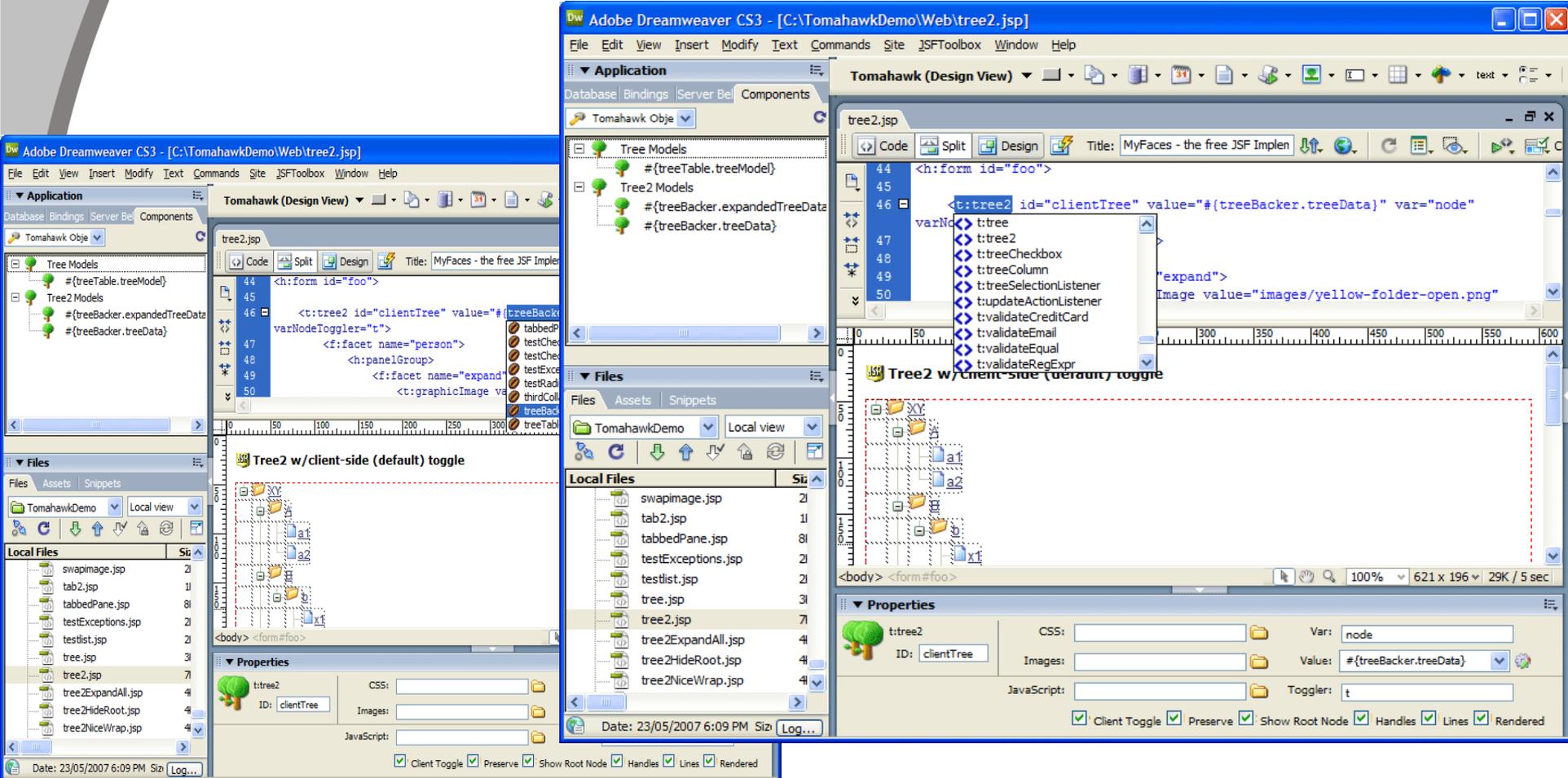
```

Unsupported Encoding wrong-encoding.

1:45 INS Unsupported Encoding wrong-encoding.

Por que usar JSF?

JSFToolbox (1)



The screenshot displays the Adobe Dreamweaver CS3 interface with the JSFToolbox integrated into the design view. The main window shows the code for a JSP page, with the JSFToolbox components palette visible on the right. The components palette lists various JSF components such as `t:tree`, `t:tree2`, `t:treeCheckbox`, `t:treeColumn`, `t:treeSelectionListener`, `t:updateActionListener`, `t:validateCreditCard`, `t:validateEmail`, `t:validateEqual`, and `t:validateRegExpr`. The code editor shows the following snippet:

```

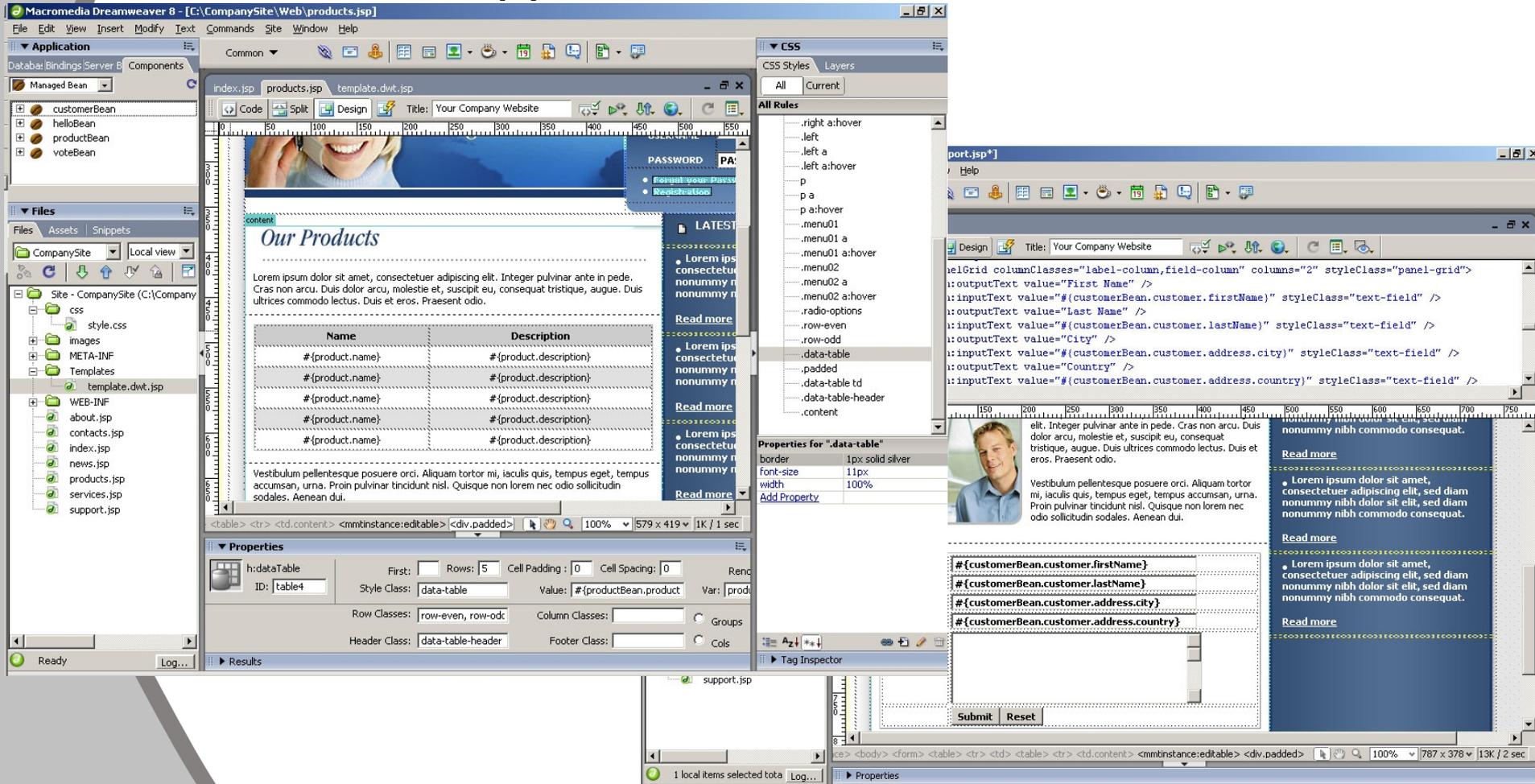
44 <h:form id="foo">
45
46 <t:tree2 id="clientTree" value="#{treeBacker.treeData}" var="node"
47 varNode="#{treeBacker.expandedTreeData}"
48 <f:facet name="person">
49 <h:panelGroup>
50 <f:facet name="expand">
51 <t:graphicImage value="#{treeBacker.expandImage}" alt="Expand icon" />
52 </f:facet>
53 </h:panelGroup>
54 </t:tree2>
55 </h:form>

```

The JSFToolbox also shows a list of local files, including `swapimage.jsp`, `tab2.jsp`, `tabbedPane.jsp`, `testExceptions.jsp`, `testlist.jsp`, `tree.jsp`, `tree2.jsp`, `tree2ExpandAll.jsp`, `tree2HideRoot.jsp`, and `tree2NiceWrap.jsp`. The Properties panel at the bottom shows the configuration for the selected `t:tree2` component, including its ID (`clientTree`), Value (`#{treeBacker.treeData}`), and Toggle (`t`).

Por que usar JSF?

JSFToolbox (2)



The screenshot shows the Macromedia Dreamweaver 8 interface. The main workspace displays a web page titled "Your Company Website" with a navigation menu, a "Our Products" section containing a table, and a registration form. The table has the following structure:

Name	Description
<input type="text" value="#{product.name}"/>	<input type="text" value="#{product.description}"/>
<input type="text" value="#{product.name}"/>	<input type="text" value="#{product.description}"/>
<input type="text" value="#{product.name}"/>	<input type="text" value="#{product.description}"/>
<input type="text" value="#{product.name}"/>	<input type="text" value="#{product.description}"/>

Below the table is a registration form with the following fields:

```

<input type="text" value="#{customerBean.customer.firstName}" styleClass="text-field" />
<input type="text" value="#{customerBean.customer.lastName}" styleClass="text-field" />
<input type="text" value="#{customerBean.customer.address.city}" styleClass="text-field" />
<input type="text" value="#{customerBean.customer.address.country}" styleClass="text-field" />
    
```

The interface includes several panels:

- Application:** Shows managed beans like customerBean, helloBean, productBean, and voteBean.
- Files:** Shows the project structure with folders like css, images, META-INF, and WEB-INF.
- Properties:** Shows properties for the selected h:dataTable, including First, Rows (5), Cell Padding, Cell Spacing, Row Classes, Column Classes, Header Class, and Footer Class.
- Tag Inspector:** Shows the current tag and its attributes.
- Properties for ".data-table":** Shows border, font-size, width, and other styling options.
- Code:** Shows the underlying HTML and JSP code for the selected elements.



Por que usar JSF?

- Grande variedade de componentes comerciais e open-source disponíveis
 - **Myfaces Tomahawk**
 - **Myfaces Sandbox**
 - Myfaces Tobago
 - Myfaces Trinidad
 - Myfaces Orchestra
 - ADF Faces
 - IceFaces
 - **WoodStock**
 - Net Advantage
 - WebGalileoFaces
 - **JBoss RichFaces**
 - JBoss Ajax4jsf
 - Backbase
 - Simplicia (ECruiser Suite)
 - QuipuKit
 - Blueprints
 - RCFaces
 - Jenia4Faces
 - ZK
 - G4JSF (GWT API)
 - YUI4JSF (Yahoo API)
 - Ext4JSF (Ext API)
 - Dynamic Faces
 - Crank

* Mais componentes? <http://www.jsfmatrix.net/>

Por que usar JSF?

MyFaces (1)

head 1	head 2	head 5
1-1	1-3	1-4 1-5
2-1	2-3	2-4 2-5
3-1	3-3	3-4 3-5
4-1	4-3	4-4 4-5
foot 1	foot 2	foot 3 foot 4

head 1	head 3	head 4
1-1	1-2 1-3	
2-1	2-2 2-3	
3-1	3-2 3-3	
4-1	4-2 4-3	
foot 1	foot 2	foot 3 foot 4 foot 5

País	Iso-Code	Cidades
AUSTRIA	AT	Show
AZERBAIJAN	AZ	Show
BAHAMAS	BS	Show
BAHRAIN		
BANGLADESH		
BARBADOS		

País	Iso-Code	Cidades
AUSTRIA	AT	Hide
Wien	<input type="checkbox"/>	Unselect
Graz	<input checked="" type="checkbox"/>	Unselect
Linz	<input type="checkbox"/>	Unselect
Salzburg	<input type="checkbox"/>	Unselect
AZERBAIJAN	AZ	Show
BAHAMAS	BS	Show
BAHRAIN	BH	Show
BANGLADESH	BD	Show
BARBADOS	BB	Sh

Tab1 Tab2 Tab3

A common paragraph

A checkbox

Tab 1 visible

Tab 2 visible

Tab 3 visible

Option 1
Option 3
another option 4
another option 6

>
<

Option 2
another option 5

< **Juli 2005** >

Mo	Di	Mi	Do	Fr	Sa	So
					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	30	31

08 Juli 2005

Juli 2005

Wk	Mo	Di	Mi	Do	Fr	Sa	So
26					1	2	3
27	4	5	6	7	8	9	10
28	11	12	13	14	15	16	17
29	18	19	20	21	22	23	24
30	25	26	27	28	29	30	31

Today is Fri, 8 Jul 2005

Por que usar JSF?



MyFaces (2)

MyFaces - The free JavaServer™ Faces Implementation (Version 1.0.4 beta)

Home Examples Documentation Options Info

- Sample 1
- Sample 2
- Validations
- Components
 - Master/Detail Example
 - Sortable Table
 - Selectboxes
 - File upload
 - Tabbed Pane
 - Calendar
 - Dynamic Lists
 - Tree
 - RdfTicker
 - DataScroller

its content open dialog action after dismiss

ViewId Dialog

This is the text from the dojo/dojoDialogView.jsp file

[close window](#)

Email

Product Information

- » Search Products
- » **Search Accessory**
- » Advanced Serach
- Online Shop
- Corporate Information
- Contact

New Hampshire

New Hampshire

New Jersey

New Mexico

New York

Header 1 Header 2 Header 3

9001  XY XY 9001

9001  A A 9001

9001  B B 9001

9001  C C 9001

Footer

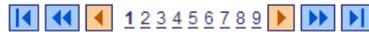
Show only manufacturers between M and Z

Id	Manufacturer	Model
1	Ford	Mondeo
2	Skoda	Octavia
3	Opel	Vectra
4	Hyundai	Elantra
5	Peugeot	406
6	Peugeot	307
7	Alfa	Romeo
8	Honda	Stream
9	Rover	25
10	Renault	Laguna
11	Mercedes	C-Klasse
12	Audi	A4
13	Rover	75
14	VW	Passat
15	Miteubishi	Carisma

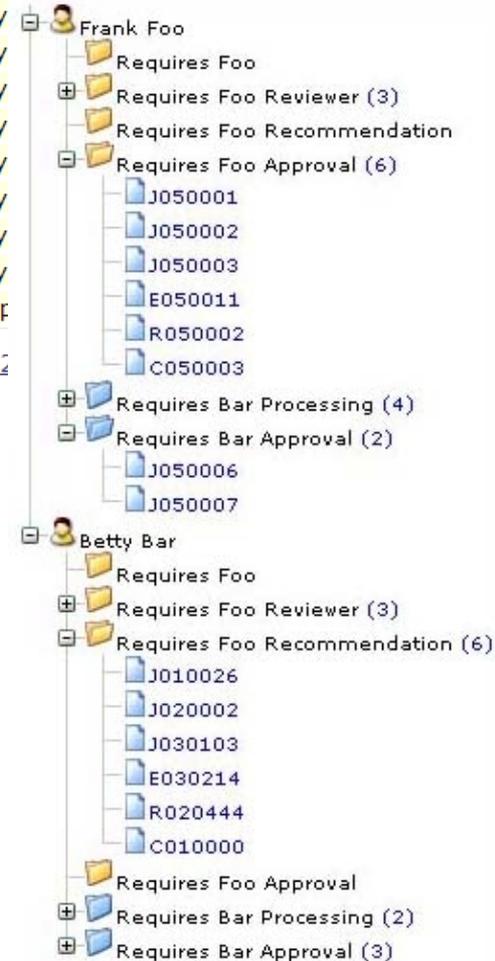
Por que usar JSF?

MyFaces (3)

Index	Type	Model
100	Car Type 100	blue
101	Car Type 101	green
102	Car Type 102	blue
103	Car Type 103	green
104	Car Type 104	blue
105	Car Type 105	green
106	Car Type 106	blue
107	Car Type 107	green
108	Car Type 108	blue
109	Car Type 109	green



	Modell	Farbe
1	Car Type 1	blue
2	Car Ty	
3	Car Ty	
4	Car Ty	
5	Car Ty	
6	Car Ty	
7	Car Ty	
8	Car Ty	
9	Car Ty	
10	Car Ty	



This is the first textual text situation.

This is the second textual text situation.

This is the third textual text situation.

- Popup Text 3
- [MyFaces Homepage](#)
 - [MyFaces Homepage](#)
 - [MyFaces Homepage](#)
 - [MyFaces Homepage](#)
 - [MyFaces Homepage](#)

Groups	Items
Group 1	Item 0
	Item 1
	Item 2
Group 2	Item 3
	Item 4
	Item 5
Group 3	Item 6
	Item 7
	Item 8
Group 4	Item 9
	Item 10
	Item 11

Startseite

Beispiele

- Beispiel 1
- Beispiel 2

Komponenten

- Master/Detail Beispiel
- Sortierbare Liste**
- Selectboxen
- Datei hochladen
- Karteireiter
- Kalender

Dokumentation

Einstellungen

Info

Por que usar JSF?

RichFaces (1)

Sep 25, 2007

<< < September, 2007 > >>

Sun	Mon	Tue	Wed	Thu	Fri	Sat
35	26	27	28	29	30	1
36	2	3	4	5	6	8
37	9	10	11	12	13	15
38	16	17	18	19	20	22
39	23	24	25	26	27	29
40	30	1	2	3	4	6

Sep 25, 2007 (x) Today

Form Validation. Using rich:message

Name: ✓

Job: ✓

Address: ✗ Address: Validation Error: Value is required.

Zip: ✗ Zip: Validation Error: Value is required.

Make	Model	Price	Mileage
Toyota	Camry	35508	20644.0
Toyota	Camry	41638	56729.0
Toyota	Camry	42071	31879.0
Toyota	Avalon	30386	67623.0
Toyota	Avalon	17355	56027.0
Toyota	Avalon	19981	51882.0
Toyota	Avalon	24380	17850.0
Toyota	Avalon	49351	55399.0
Toyota	Avalon	25698	51172.0
Toyota	Avalon	37714	20052.0

A

- Abba
- Abbeville
- Acworth
- Adairsville
- Adel
- Adrian
- Ailey

- Ajax Support
- Resources/Beans Handling
- Ajax Output
- Ajax Miscellaneous
- Data Iteration
 - Column
 - Column Group
 - Data Definition List
 - Data Filter Slider
 - Data Grid
 - Data List
 - Data Ordered List
 - Data Table
 - Data Table Scroller
 - Repeat
 - Scrollable Data Table
- Drag-Drop Support
- Rich Menu
- Rich Trees
- Rich Output
- Rich Input
- Rich Miscellaneous

Por que usar JSF?

RichFaces (2)

- Chris Rea
 - The Road To Hell
 - The Road To Hell (Part 2)
 - Tell Me There's A Heaven
 - Looking For A Rainbow
 - Daytona
 - That's What They Always Say
 - You Must Be Evil
 - I Just Wanna Be With You
 - The Road To Hell (Part 1)
 - Let's Dance
 - Your Warm And Tender Love
 - Texas
- Bach, Johann Sebastian
- Baccara
 - Grand Collection
- David Miles Huber

Car Store			
Chevrolet Tahoe Price: 32451 Mileage: 71155.0 VIN: GFSISCAZUIHHWDS Stock: FJWDGM	Chevrolet Tahoe Price: 31296 Mileage: 40447.0 VIN: TZXGEYHKXDBMS Stock: BVSLWQ	Chevrolet Tahoe Price: 20161 Mileage: 12484.0 VIN: ARUVVMZTIZFUH Stock: OTRQR	Chevrolet Tahoe Price: 46262 Mileage: 18875.0 VIN: DTIATVCJHQNMC Stock: WLAYKJ
Chevrolet Tahoe Price: 31494 Mileage: 39782.0 VIN: ONULPEGVGRWESU Stock: PQLVTG	Ford Taurus Price: 31427 Mileage: 69920.0 VIN: BPAEEJIEFVZBQV Stock: YQZHIP	Ford Taurus Price: 17331 Mileage: 55816.0 VIN: ZVFNYLRJKNFWXU Stock: AVPCWEX	Ford Taurus Price: 50767 Mileage: 41967.0 VIN: TZSXZYMOAKAYLFX Stock: KUNWC
Ford Taurus Price: 38497 Mileage: 9168.0 VIN: NFLCPNPJOMVJKA Stock: EWYSOMZ	Ford Taurus Price: 49812 Mileage: 75289.0 VIN: EUPJYRDTFFPLVV Stock: HOMMRCU	Ford Taurus Price: 49307 Mileage: 76290.0 VIN: YKWMCHEQSICVH Stock: JHBQDS	Ford Taurus Price: 27199 Mileage: 29290.0 VIN: YXRXYMBLXXMBVFL Stock: IFYBDX

Applying Skin on non-RichFaces Components

[DeepMarine](#) [BlueSky](#) [JapanCherry](#)

Name:
 Job:

You have just entered:
 Name: rponte
 Job: developer

	Expenses			subtotals
	Meals	Hotels	Transport	
San Jose				
25-Aug-97	\$37.74	\$112.00	\$45.00	
26-Aug-97	\$27.28	\$112.00	\$45.00	
	\$65.02	\$224.00	\$90.00	\$379.02
Seattle				
27-Aug-97	\$96.25	\$109.00	\$36.00	
28-Aug-97	\$35.00	\$109.00	\$36.00	
	\$131.25	\$218.00	\$72.00	\$421.25
Totals	\$196.27	\$442.00	\$162.00	\$800.27

Por que usar JSF?

RichFaces (3)

- Group 1
- Group 2
 - Item 2.1
 - Item 2.2
 - Item 2.3
 - Group 2.4
 - Item 2.4.1
 - Item 2.4.2
 - Item 2.4.3
 - Item 2.5
- Group 3
 - Item 3.1
 - Item 3.2
 - Item 3.3

File menu with sub-menu:

- New
- Open
- Save As...
 - Text File
 - PDF File
- Close
- Exit

File menu with links:

- RichFaces Home Page
- RichFaces Forum

Option menu with sub-options:

- Option1
- Option2
 - Suboption2-1
 - Suboption2-2
- Option3
 - Group2
 - SuboptionG2-2-1
 - SuboptionG2-2-2

Source List and PHP Frameworks:

- Source List: Flexible Ajax, ajaxCFC, AJAXEngine, AjaxAC, MonoRail, AJAX AGENT, FastPage, JSMX, Symfony, PowerWEB
- PHP Frameworks: PAJAJ, AJAXEngine

Map interface showing 2D/3D view and a callout box:

See this location in bird's eye view

Switch to bird's eye view.

Aerial Hybrid

Microsoft® Virtual Earth™

Zoom: In Out



Fade Effect Click to Activate	BlindDown Effect Click to Activate	BlindUp Effect Click to Activate
Opacity Effect Click to Activate	SwitchOff Effect Click to Activate	DropOut Effect Click to Activate
Highlight Effect Click to Activate	Fold Effect Click to Activate	Squish Effect Click to Activate

Por que usar JSF?

WoodStock (1)

Browse Server

Server Name: inet175

Look In:

File Filter:

To apply a new field value, press Enter in that field.

Sort By:

Name	Size	Date/Time
snadm/	512	09/07/2006 14:29
spool/	512	09/07/2006 14:55
src/	512	09/07/2006 14:22
staroffice7/	512	09/07/2006 14:56
ucbinclude/	512	09/07/2006 15:01
vmsys/	512	09/07/2006 14:48
X/	512	09/07/2006 14:30
X11/	512	09/07/2006 14:30
X11R6/	512	09/07/2006 14:30
xpg4/	512	09/07/2006 14:31
xpg6/	512	08/31/2007 14:51
asukaid30473.jpg	34689	03/14/2007 07:20

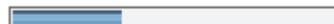
To select multiple items, use Shift-click or Control-click.

Selected Files:

Selected File(s) Path: /usr/asukaid30473.jpg

Date:
dd-MM-yyyy

Analyzing Host Barney



33% complete



▼ One

Label1 Contents on bucket 1

► Two

► Three

▼ Four

Contents on bucket 4

Node 0

- Node 10
 - Node 11
 - Node 12
- Node 20
- Node 30
 - Node 31
 - Node 32
- Node 40
- Node 50
 - Node 51
 - Node 52

Bubble with hyper link

Anchor example

Clicking on this anchor will redirect you to another page.

[Go to Sun](#)

[text example](#)

Por que usar JSF?

WoodStock (2)

popup window which appears on screen in response to a

Bubble with hyper link ✕

Anchor example

Clicking on this anchor will redirect you to another page.

[Go to Sun](#)

[text example](#)

Text Field
File Chooser
Orderable List
AddRemove
Calendar

Favorite Authors:

Available

- Harper Lee
- Ernest Hemingway
- John Steinbeck
- Michael Chabon
- Frank McCourt
- Thornton Wilder
- Philip Roth
- Michael Shaara

Add >

Add All >>

< Remove

<< Remove All

Move Up

Move Down

Selected

- Richard Russo
- Michael Cunningham

Table (1 - 5 of 20)

Filter:

	Last Name		First Name
<input type="checkbox"/>	Celestin		Horace
<input checked="" type="checkbox"/>	Nadeau		Andre
<input type="checkbox"/>	Frankart		Keith
<input type="checkbox"/>	Dunlap		Ron
<input type="checkbox"/>	Woodard		Dan

Page: of 4

Click here to open menu

- Index
- Hyperlink
- Tree
- Image

User's Role *

aaa

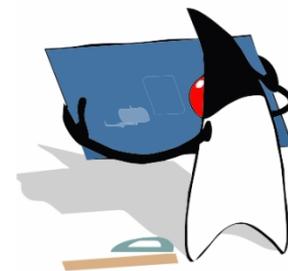
- DHCP Management
- Log Management
- Network Management

Assign Role:

Analyzing Host Barney



Overview da Tecnologia



Conceitos chaves do JSF

- Componentes
- Renderers
- Managed-beans
- Converters / Validators
- Navegação
- Eventos
- Ciclo de vida (Request Lifecycle)

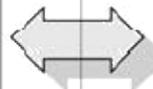


Componentes (1)

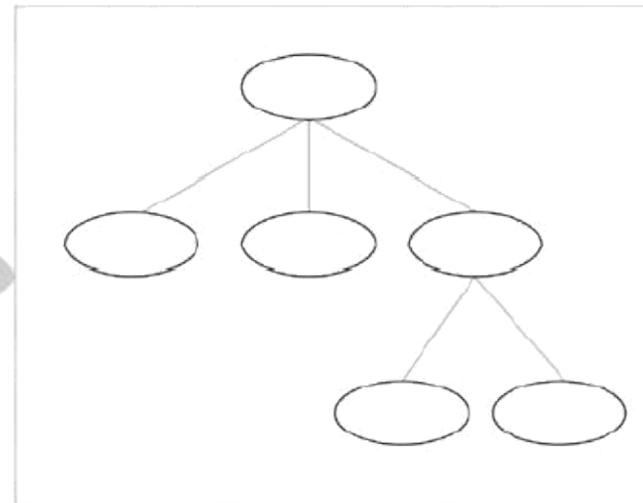
- Separa a lógica de negócio da apresentação
- Cada view é composta por uma hierarquia de componentes



Client side
HTML rendering of components

A diagram of a client-side HTML form. It features a text input field labeled "Name:" with a cursor inside. Below the input field are two buttons: "Cancel" and "Submit". The form is presented as a document with a folded top-right corner and a shadow, suggesting it's a rendered page.

Server side
JavaServer Faces view (component tree)



Componentes (2)



- Componentes podem ser adicionados na view programaticamente ou via template (JSP por default, ou Facelets para melhor desempenho e fácil desenvolvimento)
- Componentes são expressos em tags em uma página JSP
- Os componentes padrões são divididos em dois grupos
 - Faces Core `<f:view>`, `<f:loadBundle>`, `<f:param>`
 - HTML wrappers `<h:dataTable>`, `<h:selectOneMenu>`
- Componentes em sua maioria são de interface com o usuário
- Componente = class + [renderer] + tag handler (JSP)

Renderers



- Responsável por exibir o componente no cliente
- São agrupados em *render kits*
 - Render kit padrão é o HTML, mas podemos ter WML, SVG etc.
 - Efetua o *encode* (gera HTML do componente)
 - Efetua o *decode* (seta o componente com os dados da URL query string e do formulário)
- Resumindo
 - `<h:inputText id="login" size="16" />`
 - `<input type="text" id="myForm:login" name="myForm:login" size="16" />`

Managed beans



- POJO – Plain Old Java Objects
- “Linka” a view ao modelo (controller)
 - Funciona como um “espelho” da view
 - São acessados via Expression Language (EL) na JSP
 - Algo semelhante ao *Action+ActionForm* do Struts
- Declarado no faces-config.xml nos possíveis escopos
 - Application (uma instância por aplicação)
 - Session (uma instância por sessão/usuário)
 - Request (uma instância por request)
 - None (criado sob demanda)

JSF Value Binding



- “Binda” valor do componente ao managed-bean
- O binding ocorre através de JSF Expression Language (EL)
 - Semelhante a JSTL, porém usa-se ‘#{ }’ no lugar de ‘\${ }’
 - Pode-se executar métodos no modelo através de expressões
 - Não é processada imediatamente como na JSTL
 - A partir da JSF 1.2 utiliza-se a Unified Expression Language (JSP 2.1) que resolve problemas de incompatibilidade entre a JSF-EL e JSP-EL
- Para cada request o framework
 - Converte cada valor de entrada (String) em um tipo de objeto Java
 - Na exibição(output) converte do tipo de objeto Java para String

JSF Value Binding – Exemplo (1)



view.jsp

```
<h:selectOneMenu value="#{ logger.fiLevel}"  
    styleClass="portlet-form-field"  
    disabled="#{ !fiLogLevelView.hasPermModifyFiLevels}">  
    <f:selectItems value="#{fiLogLevelView.logLevels}"/>  
</h:selectOneMenu>
```

Objeto logger

```
public EnumLogLevel getFiLevel()  
{  
    return fiLevel;  
}  
public void setFiLevel(EnumLogLevel fiLevel)  
{  
    this.fiLevel = fiLevel;  
}
```

JSF Value Binding – Exemplo (2)



view.jsp

```
<h:selectOneMenu value="#{ logger.fiLevel } "  
    styleClass="portlet-form-field"  
    disabled="#{ !fiLogLevelView.hasPermModifyFiLevels }">  
    <f:selectItems value="#{fiLogLevelView.logLevels}"/>  
</h:selectOneMenu>
```

Managed-beans são registrados dentro do faces-config.xml

```
<managed-bean>  
    <managed-bean-name>fiLogLevelView</managed-bean-name>  
    <managed-bean-class>  
        product.mc.modloglevel.web.controller.FiLogLevelView  
    </managed-bean-class>  
    <managed-bean-scope>session</managed-bean-scope>  
</managed-bean>
```



Converters (1)

- Tem por finalidade converter e/ou formatar objetos do tipo Java para String e vice-versa
- Converters são bi-direcionais
 - Input converter: `getAsObject()`
 - Output converter: `getAsString()`
- JSF já fornece converters padrões para date / time, numbers etc
- Você pode implementar seu próprio converter
 - Basta implementar a interface `javax.faces.convert.Converter`
 - Registrar o converter no `faces-config.xml`
 - O converter pode atuar de forma explícita(id) ou implícita(class) de acordo com a sua configuração no `faces-config.xml`

Converters (2)



javax.faces.convert.Converter

```
public interface Converter
{
    Object getAsObject(FacesContext context,
                      UIComponent component,
                      String value) throws ConverterException;

    String getAsString(FacesContext context,
                      UIComponent component,
                      Object value) throws ConverterException;
}
```

Converters (3)



Converter é registrado dentro faces-config.xml, no exemplo abaixo estamos registrando um converter para que toda propriedade do tipo *ValuedTypesafeEnum* de qualquer bean use este converter

```
<converter>
  <converter-for-class>
    util.enums.ValuedTypesafeEnum
  </converter-for-class>
  <converter-class>
    util.web.faces.ValuedTypesafeEnumConverter
  </converter-class>
</converter>
```

Converters (4) - Exemplo



Converter declarado explicitamente

```
<h:inputText id="phone"
value="#{UserRegistration.user.phone}">
    <f:converter converterId="cejug.PhoneConverter" />
</h:inputText>
```

Ou implicitamente (converter mapeado para objetos do tipo *PhoneNumber*)

```
<h:inputText id="phone"
value="#{UserRegistration.user.phone}">
    [Olha mãe!! Nenhum converter 😊]
</h:inputText>
```

Validators (1)



- Tem por responsabilidade garantir que o valor informado pelo usuário seja válido
- Validator trabalha com objetos, não somente com Strings
- JSF já fornece validators padrões como required, length, range etc
- Você pode implementar seu próprio validator
 - Basta implementar a interface *javax.faces.validator.Validator*
 - Registrar o validator no faces-config.xml
 - Você pode associar um ou mais validators à um componente

Validators (2)



javax.faces.validator.Validator

```
public interface Validator extends EventListener {
    public static final String NOT_IN_RANGE_MESSAGE_ID =
        "javax.faces.validator.NOT_IN_RANGE";
    public void validate(FacesContext context,
        UIComponent component,
        Object value)
        throws ValidatorException;
}
```

Validators são registrados no faces-config.xml

```
<validator>
  <validator-id>prefsValidator</validator-id>
  <validator-class>com...PrefsValidator</validator-class>
</validator>
```

Validators (3) - Exemplo



Validação para obrigatoriedade e comprimento do campo

```
<h:inputText id="firstName"
value="#{UserRegistration.user.firstName}"
required="true">
    <f:validateLength minimum="2" maximum="25" />
</h:inputText>
```

Navegação (1)

- Navegação de uma página para outra
- Simples e poderoso sistema de navegação
- O framework seleciona a próxima view baseado
 - Na view anterior
 - Resultado(outcome) de uma action
 - Próprio evento (ação)
 - Qualquer combinação acima
- Permite wildcard ('*') para páginas de origem (*from-view-id*)



Navegação (2) - Exemplo



```
<navigation-rule>
  <from-view-id>/jsp/deposito/guincheiro/*</from-view-id>
  <navigation-case>
    <from-outcome>searchlist</from-outcome>
    <to-view-id>/jsp/deposito/guincheiro/searchList.jsp</to-view-id>
  </navigation-case>
  <navigation-case>
    <from-outcome>update</from-outcome>
    <to-view-id>/jsp/deposito/guincheiro/update.jsp</to-view-id>
  </navigation-case>
</navigation-rule>
```

Eventos (1)



- Componentes geram determinados eventos que por sua vez disparam ações no servidor (método(s) no managed-bean ou listeners)
- Você também pode implementar listeners de ação associados a outros componentes do formulário
- Os listeners mais comuns do JSF são
 - ActionListener
 - ValueChangeListener

Eventos (2) - Exemplo



Action event para navegação

```
<h:commandButton action="nextPage" value="Próximo" />
```

Action event que executa método no managed-bean

```
<h:commandLink action="#{cadastroBean.cadastrar}"  
value="Cadastrar" />
```

Método no managed-bean

```
public String cadastrar() {  
    // processa evento  
    return "listar"; // retorna para view "listar"  
}
```

Eventos (3) - Exemplo



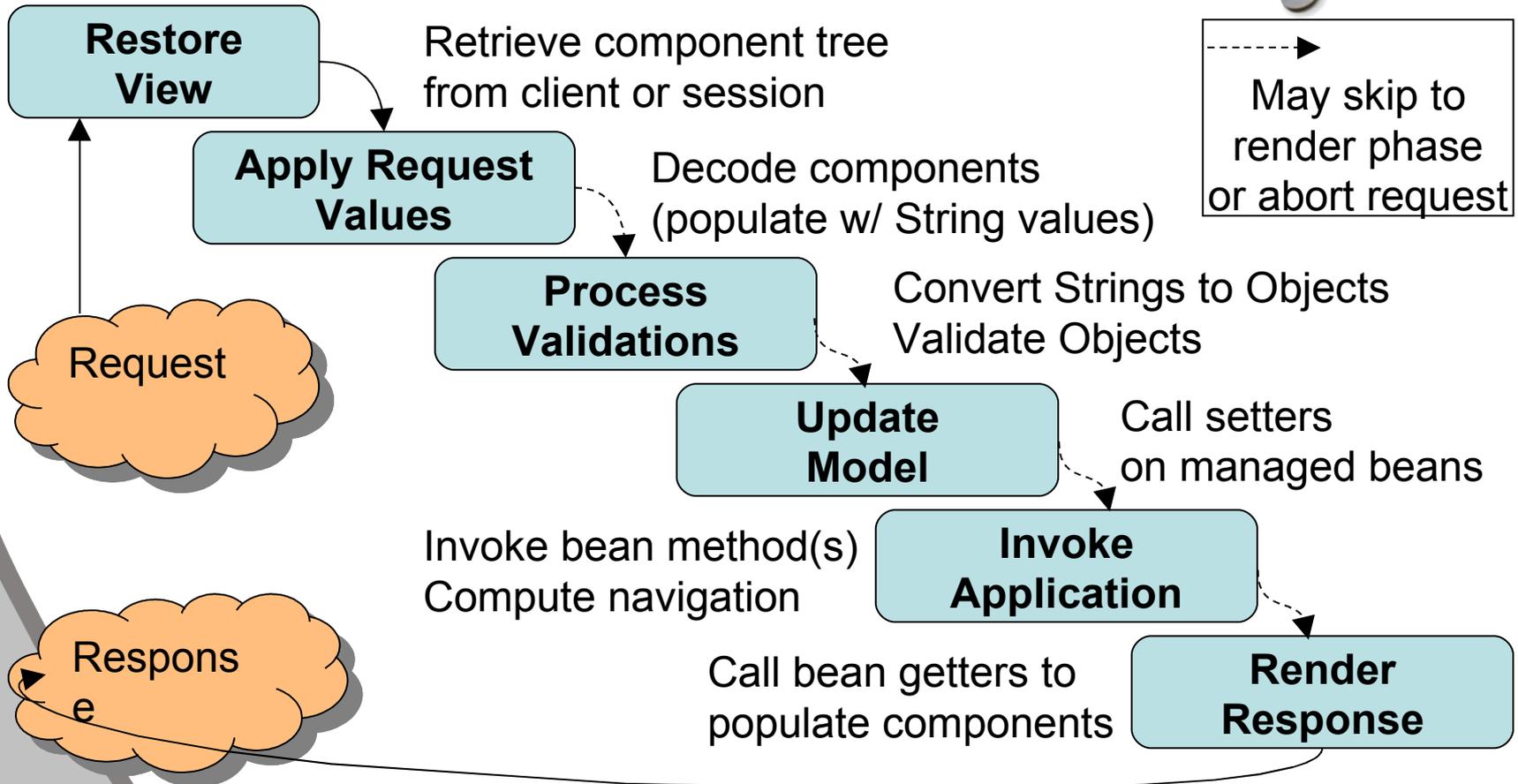
Value-change event que executa método após alterar o valor do componente

```
<h:selectOneMenu id="UF" value="#{cadastroBean.uf}"
  valueChangeListener="#{cadastroBean.carregaCidades}"
  onchange="submit();" />
  <f:selectItems value="#{cadastroBean.ufList}" />
</h:selectOneMenu>
```

Método no managed-bean

```
public void carregaCidades(ValueChangeEvent event) {
    // processa evento
    String uf = event.getNewValue(); // obtém novo valor
    this.cidadesList = Servico.carregaCidades(uf);
    // processa evento
}
```

JSF Request Lifecycle



JSF Extension Points



- Customizar componentes
- Phase listeners (antes, depois de qualquer fase)
- Customizar converters / validators
- Customizar renderers
- Customizar ActionListenerImpl para manipular eventos
- Decorar ou substituir view handler, navigation handler, state manager etc

JSF Configuração (1)



- faces-config.xml
- Arquivo XML para configurar recursos da aplicação
- Contém regras de navegação, customização e extensão de converters, validators, componentes etc
- Geralmente está localizado dentro de */WEB-INF/faces-config.xml*
- Pode ser dividido em diretórios e sub-diretórios ou dentro de arquivos jar
 - Seta *javax.faces.application.CONFIG_FILES* no web.xml
 - Ou então coloca-se *META-INF/faces-config.xml* dentro de um jar

JSF Configuração (2) - Exemplo



faces-config.xml (sem cabeçalho de arquivo xml)

```
<faces-config>
<!-- ***** -->
<!-- CONVERTERS -->
<!-- ***** -->
<converter>
  <converter-id>core.faces.CnpfCnpjConverter</converter-id>
  <converter-class>core.faces.converter.CnpfCnpjConverter</converter-class>
</converter>
<!-- ***** -->
<!-- MANAGED BEANS -->
<!-- ***** -->
<managed-bean>
  <managed-bean-name>reportBean</managed-bean-name>
  <managed-bean-class>core.faces.bean.ReportBean</managed-bean-class>
  <managed-bean-scope>request</managed-bean-scope>
</managed-bean>
<!-- ***** -->
<!-- NAVIGATION -->
<!-- ***** -->
<navigation-rule>
  <from-view-id>/jsp/deposito/tiporeq/searchList.jsp</from-view-id>
  <navigation-case>
    <from-outcome>deposito.tiporeq.update</from-outcome>
    <to-view-id>/jsp/deposito/tiporeq/update.jsp</to-view-id>
  </navigation-case>
</navigation-rule>
</faces-config>
```

O que seria uma típica aplicação JSF?

- Templates de visão (JSP ou Facelets)
- Managed-bean para cada view registrado dentro do faces-config.xml
- Regras de navegação dentro do faces-config.xml



- Meu objetivo! 😊
- JSF é a bala de prata para o desenvolvimento de aplicações web?!
- Vale a pena investir nesta tecnologia?
- Mercado em Fortaleza
- Por onde começar?
 - javasf - <http://groups.google.com/group/javasf>
 - cejug - <http://www.cejug.org/>

 Perguntas?





Obrigado! 😊