

# Desenvolvimento Web II

## Aula 02 - Conhecendo as Tags do JSF

# Apresentação

---

Olá, seja bem-vindo(a) à nossa segunda aula sobre o framework JSF. Agora que já entendemos o que é o JavaServer Faces (JSF), quais as suas características e o ciclo de vida de uma aplicação desenvolvida com o mesmo, iremos agora nos aprofundar um pouco mais conhecendo suas principais tags e como realizar conversões de dados. Essas tags são muito importantes, pois servem para adicionarmos nas nossas aplicações o que representam os componentes HTML das páginas JSF. Ao final dessa aula, você poderá praticar um pouco mais o que aprendemos em uma aplicação básica, vamos lá?

## Objetivos

- Conhecer as principais TAGs do JSF;
- Entender como fazer conversões de dados no JSF.

# TAGS

---

O JavaServer Faces provê um conjunto de tags para que possamos definir nossa árvore de componentes das páginas JSF (facelets). Para facilitar o entendimento, vamos a um exemplo: se eu quero adicionar um campo para receber texto (input text) do HTML em uma página JSF, eu preciso utilizar a tag disponibilizada por esse framework que representa esse componente. No caso desse exemplo, a tag é a `h:inputText`. Detalharemos as principais tags, que estão agrupadas nos seguintes tipos: core e HTML. Mas antes de utilizar uma tag, precisamos informar à página JSF (facelet) qual o namespace da mesma. Para a utilização das tags JSF Core precisamos informar o namespace definido na Listagem 1. No caso da tags JSF HTML, precisamos informar o namespace definido na Listagem 2.

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

**Listagem 1** - Namespace JSF Core

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

**Listagem 2** - Namespace JSF HTML.

Observe que você poderá ter em sua página JSF tags de diferentes conjuntos de tags, ou seja, usar as tags JSF Core não exclui a possibilidade de utilizar também as tags JSF HTML. A realidade, na prática, é que você utilizará em sua página JSF diferentes tags pertencentes a diferentes conjuntos de tags. Por esse motivo, terá de incluir em suas páginas JSF diferentes namespaces, conforme apresentado na Listagem 3.

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

**Listagem 3** - Namespaces JSF HTML e Core na mesma página JSF.

# Tags JSF HTML

---

As tags presentes no conjunto JSF HTML fornecem ao desenvolvedor a possibilidade de incluir componentes HTML nas páginas JSF. Dentre as principais tags temos:

## h:inputText

Renderiza uma entrada do tipo texto (input com type= "text"). Utilizada, por exemplo, em formulários para obter o login do usuário, nome, etc. A Listagem 4 apresenta um exemplo de utilização da mesma e sua respectiva renderização em código HTML;

```
1 <h:inputText value="Hello World!" />
```

Renderiza:

```
1 <input type="text" value="Hello World!" />
```

**Listagem 4** - Tag h:inputText e sua renderização em código HTML.

## h:inputSecret

Renderiza uma entrada do tipo senha (input com type= "password"). Utilizada, por exemplo, em formulários para obter a senha do usuário. A Listagem 5 apresenta um exemplo de utilização da mesma e sua respectiva renderização em código HTML;

```
1 <h:inputSecret value="password" />
```

Renderiza:

```
1 <input type="password" value="password" />
```

**Listagem 5** - Tag h:inputSecret e sua renderização em código HTML.

## h:inputTextarea

Renderiza um campo de área de texto. Utilizado, por exemplo, em formulários para obter uma descrição. A Listagem 6 apresenta um exemplo de utilização da mesma e sua respectiva renderização em código HTML;

```
1 <h:inputTextarea row="10" col="10" value="Hello World Text Area!!!" />
```

Renderiza:

```
1 <textarea> Hello World Text Area!!!</textarea>
```

**Listagem 6** - Tag h:inputTextarea e sua renderização em código HTML.

## h:selectBooleanCheckbox

Renderiza uma caixa de seleção. Utilizada, por exemplo, em formulários para o usuário informar se concorda com uma licença. A Listagem 7 apresenta um exemplo de utilização da mesma e sua respectiva renderização em código HTML;

```
1 <h:selectBooleanCheckbox value="De acordo com a licença." id="chkAceita" />
```

Renderiza:

```
1 <input type="checkbox" checked="checked">De acordo com a licença.</input>
```

**Listagem 7** - Tag h:selectBooleanCheckbox e sua renderização em código HTML.

## h:selectManyCheckbox

Renderiza um grupo de caixas de seleção. Utilizada, por exemplo, em formulários de questões onde o usuário pode escolher mais de uma opção. A Listagem 8 apresenta um exemplo de utilização da mesma e sua respectiva renderização em código HTML;

```
1 <h:selectManyCheckbox value="#{userData.data}">  
2 <f:selectItem itemValue="1" itemLabel="Item 1" />  
3 <f:selectItem itemValue="2" itemLabel="Item 2" />  
4 </h:selectManyCheckbox>
```

Renderiza:

```
1 <table>
2   <tr>
3     <td><input value="1" type="checkbox" checked="checked" />
4     <label> Item 1</label>
5   </td>
6   <td><input value="2" type="checkbox" checked="checked" />
7     <label> Item 2</label>
8   </td>
9 </tr>
10 </table>
```

**Listagem 8** - Tag `h:selectManyCheckbox` e sua renderização em código HTML.

## `h:selectOneRadio`

Renderiza um botão de opção. Utilizada, por exemplo, em formulários para a escolha do sexo do usuário. A Listagem 9 apresenta um exemplo de utilização da mesma e sua respectiva renderização em código HTML;

```
1 <h:selectOneRadio value="#{userData.data}">
2   <f:selectItem itemValue="1" itemLabel="Masculino" />
3   <f:selectItem itemValue="2" itemLabel="Feminino" />
4 </h:selectOneRadio>
```

Renderiza:

```
1 <table>
2   <tr>
3     <td><input type="radio" checked="checked" value="1" />
4     <label> Masculino </label></td>
5     <td><input type="radio" value="2" />
6     <label> Feminino </label></td>
7   </tr>
8 </table>
```

**Listagem 9** - Tag `h:selectOneRadio` e sua renderização em código HTML.

## h:selectOneMenu

Renderiza uma caixa de combinação. Utilizada, por exemplo, em formulários para a escolha da cidade do usuário. A Listagem 10 apresenta um exemplo de utilização da mesma e sua respectiva renderização em código HTML;

```
1 <h:selectOneMenu value="#{userData.data}">
2   <f:selectItem itemValue="1" itemLabel="Natal" />
3   <f:selectItem itemValue="2" itemLabel="Parnamirim" />
4 </h:selectOneMenu>
```

Renderiza:

```
1 <select>
2   <option value="1">Natal</option>
3   <option value="2">Parnamirim</option>
4 </select>
```

**Listagem 10** - Tag h:selectOneMenu e sua renderização em código HTML.

## h:outputText

Renderiza um texto. Utilizada, por exemplo, para exibir algum texto para o usuário. A Listagem 11 apresenta um exemplo de utilização da mesma e sua respectiva renderização em código HTML;

```
1 <h:outputText value="Aprendendo tags!" />
```

Renderiza: Aprendendo Tags!

**Listagem 11** - Tag h:outputText e sua renderização em código HTML.

## Tags JSF HTML

---

### h:graphicImage

Renderiza uma imagem. A Listagem 12 apresenta um exemplo de utilização da mesma e sua respectiva renderização em código HTML;

```
1 <h:graphicImage value="https://imd.ufrn.br/portal/assets/images/UFRN_LOGO-01.svg"/>
```

Renderiza:

```
1 
```

**Listagem 12** - Tag h:graphicImage e sua renderização em código HTML.

## h:commandButton

Renderiza um botão (input com type= "text"). A Listagem 13 apresenta um exemplo de utilização da mesma e sua respectiva renderização em código HTML;

```
1 <h:commandButton value="Click aqui!" onclick="alert('Bem-vindo ao IMD!');" />
```

Renderiza:

```
1 <input type="submit" value="Click aqui!" onclick="alert('Bem-vindo ao IMD!');" />
```

**Listagem 13** - Tag h:commandButton e sua renderização em código HTML.

## h:commandLink

Renderiza um link. A Listagem 14 apresenta um exemplo de utilização da mesma e sua respectiva renderização em código HTML;

```
1 <h:commandLink value="Página 1" action="pagina1" />
```

Renderiza:

```
1 <a href="#">Página 1</a>
```

**Listagem 14** - Tag h:commandLink e sua renderização em código HTML.



## h:message

Renderiza uma mensagem para um componente na página JSF. Utilizada, por exemplo, para as mensagens de validações de campos em formulários. A Listagem 15 apresenta um exemplo de utilização da mesma e sua respectiva renderização em código HTML;

```
1 <h:inputText id="nomeUsuario" size="20" label="nomeUsuario" required="true" />
2 <h:message for="nomeUsuario" style="color:red" />
```

Renderiza (em caso de validação):

```
1 <span style="color:red">nomeUsuario: Validation Error: Value is required</span>
```

**Listagem 15** - Tag h:message e sua renderização em código HTML.

## h:messages

Renderiza mensagens para todos os componentes na página JSF. Também utilizada para mensagens de validações em formulários. A Listagem 16 apresenta um exemplo de utilização da mesma e sua respectiva renderização em código HTML;

```
1 <h:messages style="color:red;margin:8px;" />
```

Renderiza:

```
1 <ul style="color:red;margin:8px;">
2   <li> nomeUsuario: Validation Error:
3     Length is greater than allowable maximum of '20' </li>
4   <li> senha: Validation Error:
5     Length is less than allowable minimum of '5' </li>
6 </ul>
```

**Listagem 16** - Tag h:messages e sua renderização em código HTML.

## f:convertDateTime

Realiza a conversão de uma data e hora para um formato definido. Utilizada para facilitar a leitura das datas pelos usuários. A Listagem 17 apresenta um exemplo de utilização da mesma e sua respectiva renderização em código HTML;

```
1 <h:outputText value="#{item.data}">
2 <f:convertDateTime pattern="dd/MM/yyyy HH:mm"/>
3 </h:outputText>
```

Renderiza: 17/10/2014

**Listagem 17** - Tag f:convertDateTime e sua renderização em código HTML.

## f:validateLength

Realiza a validação de uma quantidade de caracteres especificada. Utilizada, por exemplo, para validar o tamanho mínimo de caracteres de um campo em um formulário. A Listagem 18 apresenta um exemplo de utilização da mesma;

```
1 <h:inputSecret value="#{usuarioMBean.senha}">
2 <f:validateLength minimum="8" maximum="16" />
3 </h:inputSecret>
```

**Listagem 18** - Tag f:validateLength.

## f:selectItem

Especifica um item para as tags de botão de opção, caixa de combinação, etc. A Listagem 19 apresenta um exemplo de utilização da mesma;

```
1 <h:selectOneMenu value="#{usuarioMBean.estado}">
2 <f:selectItem itemValue="1" itemLabel="RN" />
3 <f:selectItem itemValue="2" itemLabel="PB" />
4 </h:selectOneMenu>
```

**Listagem 19** - Tag f:selectItem

## f:selectItems

Especifica um conjunto de itens para as tags de botão de opção, caixa de combinação, etc. A Listagem 20 apresenta um exemplo de utilização da mesma;

```
1 <h:selectOneMenu value="#{usuarioMBean.estado}" >
2 <f:selectItems value="#{estadosMBean.estados}" var="uf"
3 itemLabel="#{uf.nome}" itemValue="#{uf.id}" />
4 </h:selectOneMenu>
```

**Listagem 20** - Tag f:selectItems

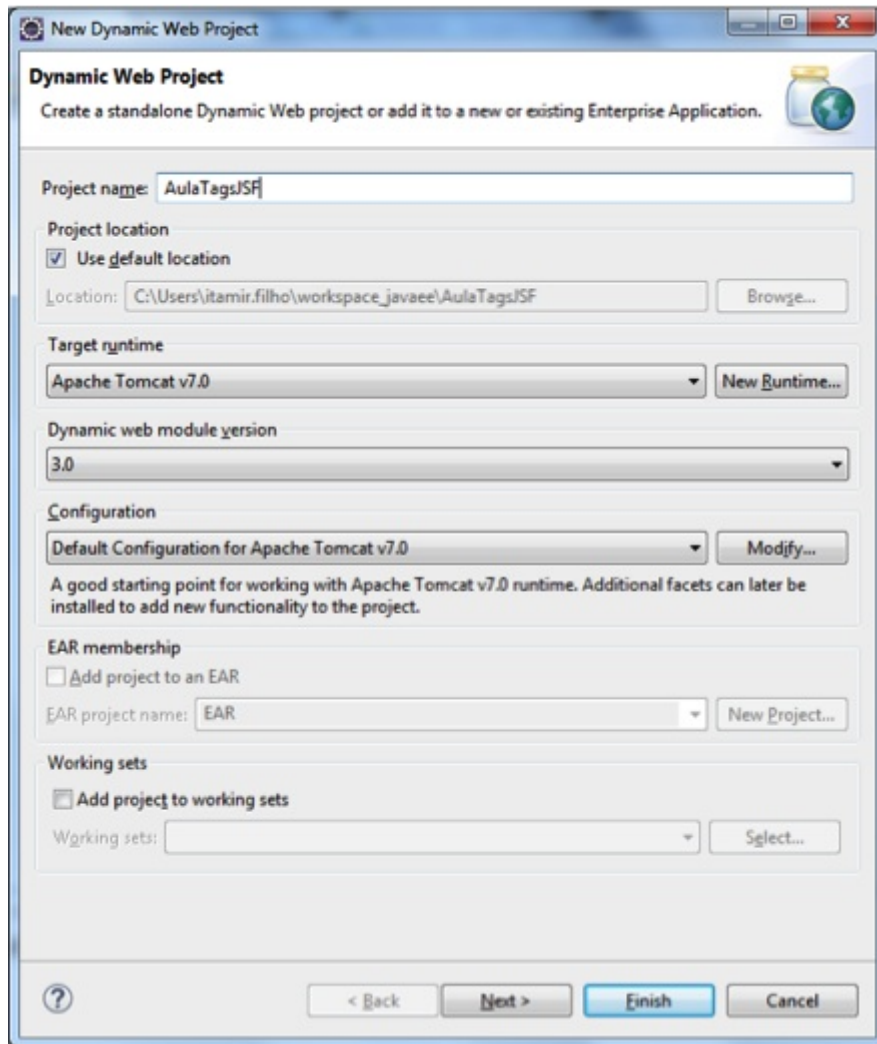
## Criando uma Segunda Aplicação Web

---

Agora que já conhecemos as principais tags do framework JSF, vamos criar a nossa segunda aplicação web para colocar em prática nosso conhecimento. Essa aplicação será um formulário para cadastro de pessoas criado com as tags que nós estudamos, incluindo campos, validações e etc. Então vamos lá!

Para começar, vamos criar um projeto dinâmico web chamado AulaTagsJSF. Você poderá seguir os mesmos passos definidos na aula 1 para criação de um projeto JSF, em que será necessário definir o nome do projeto, o servidor de aplicação de destino. A Figura 1 apresenta a janela de criação de projetos dinâmicos web do eclipse com as configurações desse nosso projeto.

**Figura 01** - Janela para criação do projeto AulaTagsJSF.



**Fonte:** Autoria Própria

Criado o projeto, coloque dentro da pasta lib o jar do JSF (javax.faces-2.2.8.jar). Agora criaremos um arquivo chamado web.xml, que deverá ser adicionado na pasta WEB-INF. Esse arquivo conterá a definição da servlet do JSF, que chamaremos de Faces Servlet, e o seu mapeamento. O conteúdo do arquivo web.xml é apresentado na Listagem 21. Está achando isso familiar? Sim! Esses são os passos que utilizamos para criar o nosso primeiro projeto com JSF, e serão os passos que você utilizará para criar seus projetos JSF.

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
3 xmlns="http://java.sun.com/xml/ns/javaee"
4 xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
5 http://java.sun.com/xml/ns/javaee/web-app_3_0.xsd" version="3.0">
6
7 <display-name>AulaTagsJSF</display-name>
8   <welcome-file-list>
9     <welcome-file>index.jsf</welcome-file>
10  </welcome-file-list>
11
12 <servlet>
13   <servlet-name>Faces Servlet</servlet-name>
14   <servlet-class>javax.faces.webapp.FacesServlet</servlet-class>
15   <load-on-startup>1</load-on-startup>
16 </servlet>
17
18 <servlet-mapping>
19   <servlet-name>Faces Servlet</servlet-name>
20   <url-pattern>*.jsf</url-pattern>
21 </servlet-mapping>
22
23 </web-app>

```

**Listagem 21** - Código do web.xml da aplicação AulaTagsJSF.

Vamos agora criar os seguintes pacotes:

- br.ufrn.imd.dominio: que iremos adicionar a entidade Pessoa;
- br.ufrn.imd.controllers: que iremos adicionar o controller CadastroMBean.

A classe Pessoa terá os atributos: nome, endereço, data de nascimento e sexo. A Listagem 22 apresenta o código dessa classe.

```

1 package br.ufrn.imd.dominio;
2 import java.util.Date;
3
4 /**
5  * Classe simplificada que representa uma pessoa.
6  * @author itamir.filho
7  */
8 public class Pessoa {
9
10     private String nome;
11     private String endereco;
12     private Date dataNascimento;
13     private String sexo;
14
15     public String getNome() {
16         return nome;
17     }
18
19     public void setNome(String nome) {
20         this.nome = nome;
21     }
22
23     public String getEndereco() {
24         return endereco;
25     }
26
27     public void setEndereco(String endereco) {
28         this.endereco = endereco;
29     }
30
31     public Date getDataNascimento() {
32         return dataNascimento;
33     }
34
35     public void setDataNascimento(Date dataNascimento) {
36         this.dataNascimento = dataNascimento;
37     }
38
39     public String getSexo() {
40         return sexo;
41     }
42
43     public void setSexo(String sexo) {
44         this.sexo = sexo;
45     }
46
47 }

```

**Listagem 22** - Código da classe Pessoa.

O controller `CadastroMBean` terá como atributo um objeto da classe `Pessoa` e um método chamado `cadastrar`. A Listagem 23 apresenta o código da classe `CadastroMBean`. Observe que vamos utilizar a anotação `@ManagedBean` e `@SessionScope` do pacote `javax.faces.bean` para definir que essa classe é um controller (MBean) do JSF e está no escopo de sessão. **(POPUP: objetos associados à sessão dos usuários existirão até que os mesmos sejam retirados ou a sessão expire)**

Observe que o método `cadastrar`, que estará associado a um botão presente na nossa página JSF, retorna a String `"/sucesso.jsf"`. Essa String retornada indica para qual página JSF o usuário será direcionado após o clique nesse botão. Nesse caso, o usuário será direcionado para a página presente na pasta `WebContent` de nome `"sucesso"`.

```

1 package br.ufrn.imd.controllers;
2
3 import javax.faces.bean.ManagedBean;
4 import javax.faces.bean.SessionScoped;
5
6 import br.ufrn.imd.dominio.Pessoa;
7
8 /**
9  * MBean para realização do cadastro da pessoa.
10  * @author itamir.filho
11  */
12 @ManagedBean
13 @SessionScoped
14 public class CadastroMBean {
15
16     //atributo pessoa
17     Pessoa pessoa;
18
19     /**
20      * Construtor
21      */
22     public CadastroMBean() {
23         pessoa = new Pessoa();
24     }
25
26     public Pessoa getPessoa() {
27         return pessoa;
28     }
29
30     public void setPessoa(Pessoa pessoa) {
31         this.pessoa = pessoa;
32     }
33
34     /**
35      * Método para realização do cadastro.
36      * @return
37      */
38
39     public String cadastrar() {
40         return "/sucesso.jsf";
41     }
42
43 }

```

**Listagem 23** - Código da classe CadastroMBean.

Vamos agora à criação de duas páginas JSF: form.xhtml e sucesso.xhtml, apresentadas respectivamente nas Listagens 24 e 25. Lembre-se que elas precisam estar dentro da pasta WebContent. Na página “form”, teremos o formulário que



utilizando as tags do JSF obterão os seguintes dados do usuário: nome (campo obrigatório), endereço (campo obrigatório), data de nascimento e sexo. Na página “sucesso”, iremos apenas apresentar os dados capturados.

```

1 <html xmlns="http://www.w3.org/1999/xhtml"
2 xmlns:h="http://java.sun.com/jsf/html"
3 xmlns:f="http://java.sun.com/jsf/core">
4
5 <h:head>
6   <title>Formulário</title>
7 </h:head>
8
9 <h:body>
10 <h:messages/>
11 <h:form>
12
13 <table>
14   <tr>
15     <td>Nome:</td>
16     <td><h:inputText value="#{cadastroMBean.pessoa.nome}" size="50" required="true" required
17   </tr>
18   <tr>
19     <td>Endereço:</td>
20     <td><h:inputTextarea value="#{cadastroMBean.pessoa.endereco}" cols="50" rows="4" require
21   </tr>
22   <tr>
23     <td>Data de nascimento: </td>
24     <td>
25       <h:inputText value="#{cadastroMBean.pessoa.dataNascimento}" size="5" maxlength="10">
26       <f:convertDateTime pattern="dd/MM/yyyy" />
27       </h:inputText>
28     </td>
29   </tr>
30   <tr>
31     <td>Sexo: </td>
32     <td>
33       <h:selectOneRadio value="#{cadastroMBean.pessoa.sexo}">
34       <f:selectItem itemValue="M" itemLabel="Masculino" />
35       <f:selectItem itemValue="F" itemLabel="Feminino" />
36       </h:selectOneRadio>
37     </td>
38   </tr>
39   <tr>
40     <td colspan="2" align="center">
41       <h:commandButton value="Cadastrar" action="#{cadastroMBean.cadastrar}" />
42     </td>
43   </tr>
44 </table>
45
46 </h:form>
47 </h:body>
48 </html>

```

**Listagem 24** - Código da página form.xhtml.

```

1 <html xmlns="http://www.w3.org/1999/xhtml"
2 xmlns:h="http://java.sun.com/jsf/html"
3 xmlns:f="http://java.sun.com/jsf/core">
4
5 <h:head>
6   <title>Sucesso </title>
7 </h:head>
8
9 <h:body>
10
11 <table>
12   <tr>
13     <td>Nome: </td>
14     <td><h:outputText value="#{cadastroMBean.pessoa.nome}" /> </td>
15   </tr>
16   <tr>
17     <td>Endereço: </td>
18     <td><h:outputText value="#{cadastroMBean.pessoa.endereco}" /> </td>
19   </tr>
20   <tr>
21     <td>Data de nascimento: </td>
22     <td>
23       <h:outputText value="#{cadastroMBean.pessoa.dataNascimento}">
24       <f:convertDateTime pattern="dd/MM/yyyy" />
25       </h:outputText>
26     </td>
27   </tr>
28   <tr>
29     <td>Sexo: </td>
30     <td>
31       <h:outputText value="#{cadastroMBean.pessoa.sexo}" />
32     </td>
33   </tr>
34 </table>
35 </h:body>
36 </html>

```

**Listagem 25** - Código da página sucesso.xhtml.

Após a codificação das classes e páginas JSF, apresentadas nas Listagens 22, 23, 24 e 25, vamos iniciar o servidor de aplicação Apache Tomcat e acessar nossa aplicação pela URL: <http://localhost:8080/AulaTagsJSF/form.jsf> . Ao acessar, a página apresentada na Figura 2 é exibida, e ao preenchermos e clicarmos no botão de cadastrar obtemos a página apresentada na Figura 3.

**Figura 02** - Janela para criação do projeto AulaTagsJSF.

Formulário

localhost:8080/AulaTagsJSF/form.jsf

Aplicativos Favoritos

Nome:

Endereço:

Data de nascimento:

Sexo: ☐ Masculino ☐ Feminino

**Fonte:** Autoria Própria

**Figura 03** - Página de sucesso.

Sucesso

localhost:8080/AulaTagsJSF/form.jsf

Aplicativos Favoritos

Nome: Itamir

Endereço: Rua das casas.

Data de nascimento: 17/10/2010

Sexo: M

**Fonte:** Autoria Própria

# Resumo

---

E com isso chegamos ao fim da nossa aula. Observe que já conhecemos sobre o framework JSF e suas principais Tags. Agora você já pode começar a desenvolver aplicações web com formulários! Na próxima aula continuaremos apresentando mais sobre o mesmo para que você possa criar aplicações web ainda mais completas. Vamos em frente!

## Referências

---

Tag Library Documentation Generator - Generated Documentation. Disponível em: <https://docs.oracle.com/javaee/6/javaxserverfaces/2.0/docs/pdl/docs/facelets/>.

Tutorials Point. **Basic JSF Tags.** Disponível em: [http://www.tutorialspoint.com/jsf/jsf\\_basic\\_tags.htm](http://www.tutorialspoint.com/jsf/jsf_basic_tags.htm). Acesso em: 27 março de 2015.

JSF Tags. Disponível em: <http://www.horstmann.com/corejsf/jsf-tags.html>. Acesso em: 27 março de 2015.

JSF Tool Box. **Tag Reference.** Disponível em: <http://www.jsftoolbox.com/documentation/help/12-TagReference/core/index.jsf>. Acesso em: 27 março de 2015.