

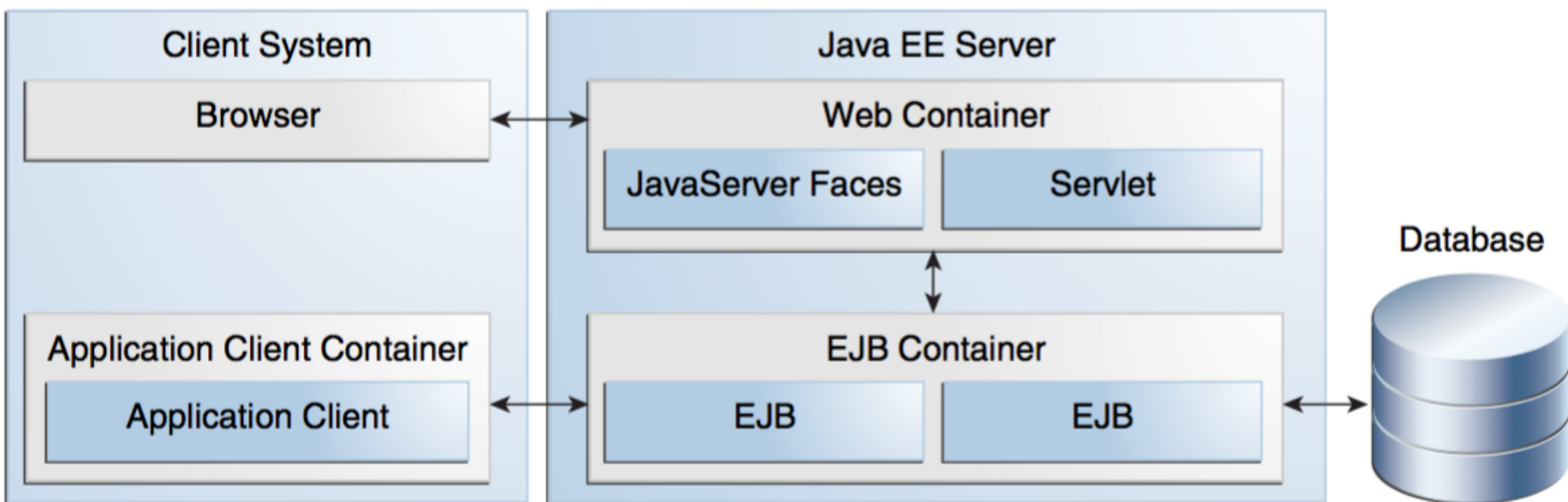
Curso Desenvolvedor Java



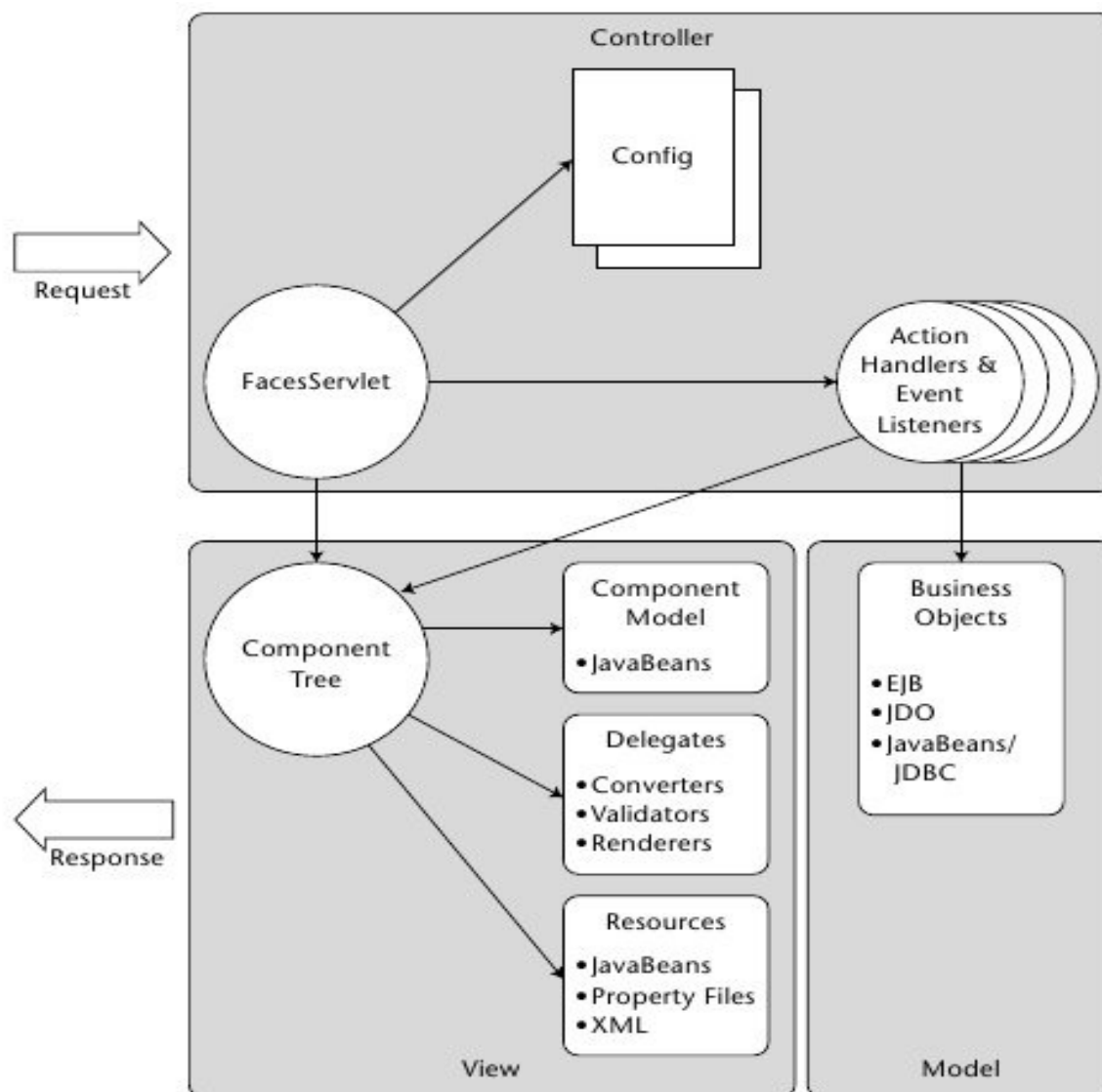
JAVA™

JSF

O JFS no Container Web



JSF Model-View-Controller



Para que seja possível a utilização de JSF em uma aplicação WEB é necessário que o Faces Servlet seja configurado no arquivo de configuração da aplicação, o web.xml.

```
<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>*.xhtml</url-pattern>
</servlet-mapping>
```

Nesta configuração também é informado qual o padrão de URL para que o Faces Servlet identifique as páginas que ele irá processar.

JSF Components Tags

| Tag | Functions | Rendered As | Appearance |
|------------------------------|---|---|--|
| <code>h:column</code> | Represents a column of data in a data component | A column of data in an HTML table | A column in a table |
| <code>h:commandButton</code> | Submits a form to the application | An HTML <code><input type=value></code> element for which the type value can be "submit", "reset", or "image" | A button |
| <code>h:commandLink</code> | Links to another page or location on a page | An HTML <code><a href></code> element | A link |
| <code>h:dataTable</code> | Represents a data wrapper | An HTML <code><table></code> element | A table that can be updated dynamically |
| <code>h:form</code> | Represents an input form (inner tags of the form receive the data that will be submitted with the form) | An HTML <code><form></code> element | No appearance |
| <code>h:graphicImage</code> | Displays an image | An HTML <code></code> element | An image |
| <code>h:inputSecret</code> | Allows a user to input a string without the actual string appearing in the field | An HTML <code><input type="password"></code> element | A field that displays a row of characters instead of the actual string entered |
| <code>h:inputText</code> | Allows a user to input a string | An HTML <code><input type="text"></code> element | A field |
| <code>h:inputTextarea</code> | Allows a user to enter a multiline string | An HTML <code><textarea></code> element | A multirow field |
| <code>h:message</code> | Displays a localized message | An HTML <code></code> tag if styles are used | A text string |

JSF Components Tags

| Tag | Functions | Rendered As | Appearance |
|--------------------------------------|--|--|--------------------|
| <code>h:messages</code> | Displays localized messages | A set of HTML <code></code> tags if styles are used | A text string |
| <code>h:outputFormat</code> | Displays a formatted message | Plain text | Plain text |
| <code>h:outputLabel</code> | Displays a nested component as a label for a specified input field | An HTML <code><label></code> element | Plain text |
| <code>h:outputLink</code> | Links to another page or location on a page without generating an action event | An HTML <code><a></code> element | A link |
| <code>h:outputText</code> | Displays a line of text | Plain text | Plain text |
| <code>h:panelGrid</code> | Displays a table | An HTML <code><table></code> element with <code><tr></code> and <code><td></code> elements | A table |
| <code>h:panelGroup</code> | Groups a set of components under one parent | A HTML <code><div></code> or <code></code> element | A row in a table |
| <code>h:selectBooleanCheckbox</code> | Allows a user to change the value of a Boolean choice | An HTML <code><input type="checkbox"></code> element | A check box |
| <code>h:selectManyListbox</code> | Allows a user to select multiple items from a set of items all displayed at once | An HTML <code><select></code> element | A box |
| <code>h:selectOneMenu</code> | Allows a user to select one item from a set of items | An HTML <code><select></code> element | A menu |
| <code>h:selectOneRadio</code> | Allows a user to select one item from a set of items | An HTML <code><input type="radio"></code> element | A group of options |

Pass-Through HTML5 Elements

| HTML5 Element Name | Identifying Attribute | Facelets Tag |
|--------------------|-----------------------|-------------------------|
| a | jsf:action | h:commandLink |
| a | jsf:value | h:outputLink |
| body | | h:body |
| button | | h:commandButton |
| form | | h:form |
| head | | h:head |
| img | | h:graphicImage |
| input | type="button" | h:commandButton |
| input | type="checkbox" | h:selectBooleanCheckbox |
| input | type="color" | h:inputText |
| input | type="date" | h:inputText |
| input | type="email" | h:inputText |
| input | type="number" | h:inputText |
| input | type="password" | h:inputSecret |
| input | type="submit" | h:commandButton |
| label | | h:outputLabel |
| link | | h:outputStylesheet |
| script | | h:outputScript |
| select | | h:selectOneListbox |
| textarea | | h:inputTextArea |

Facelets para Templates

| Tag | Function |
|-----------------------------|---|
| <code>ui:component</code> | Defines a component that is created and added to the component tree. |
| <code>ui:composition</code> | Defines a page composition that optionally uses a template. Content outside of this tag is ignored. |
| <code>ui:debug</code> | Defines a debug component that is created and added to the component tree. |
| <code>ui:decorate</code> | Similar to the composition tag but does not disregard content outside this tag. |
| <code>ui:define</code> | Defines content that is inserted into a page by a template. |
| <code>ui:fragment</code> | Similar to the component tag but does not disregard content outside this tag. |
| <code>ui:include</code> | Encapsulates and reuses content for multiple pages. |
| <code>ui:insert</code> | Inserts content into a template. |
| <code>ui:param</code> | Used to pass parameters to an included file. |
| <code>ui:repeat</code> | Used as an alternative for loop tags, such as <code>c:forEach</code> or <code>h:dataTable</code> . |
| <code>ui:remove</code> | Removes content from a page. |

| Tag Library | URI | Prefix | Example | Contents |
|---------------------------------------|---|--------|---|--|
| JavaServer Faces Facelets Tag Library | http://xmlns.jcp.org/jsf/facelets | ui: | ui:component ui:insert | Tags for templating |
| JavaServer Faces HTML Tag Library | http://xmlns.jcp.org/jsf/html | h: | h:head h:body h:outputText h:inputText | JavaServer Faces component tags for all UIComponent objects |
| JavaServer Faces Core Tag Library | http://xmlns.jcp.org/jsf/core | f: | f:actionListener f:attribute | Tags for JavaServer Faces custom actions that are independent of any particular render kit |
| Pass-through Elements Tag Library | http://xmlns.jcp.org/jsf | jsf: | jsf:id | Tags to support HTML5-friendly markup |
| Pass-through Attributes Tag Library | http://xmlns.jcp.org/jsf/passthrough | p: | p:type | Tags to support HTML5-friendly markup |
| Composite Component Tag Library | http://xmlns.jcp.org/jsf/composite | cc: | cc:interface | Tags to support composite components |
| JSTL Core Tag Library | http://xmlns.jcp.org/jsp/jstl/core | c: | c:forEach c:catch | JSTL 1.2 Core Tags |
| JSTL Functions Tag Library | http://xmlns.jcp.org/jsp/jstl/functions | fn: | fn:toUpperCase fn:toLowerCase | JSTL 1.2 Functions Tags |

O JSF utiliza páginas XHTML e para configurarmos os prefixos utilizados pelo JSF precisamos registra-los no elemento HTML.

```
<html lang="pt-br"  
  xmlns="http://www.w3.org/1999/xhtml"  
  xmlns:h="http://xmlns.jcp.org/jsf/html"  
  xmlns:f="http://xmlns.jcp.org/jsf/core"  
  xmlns:jsf="http://xmlns.jcp.org/jsf">
```

O Managed Bean e o JSF

```
@ManagedBean(name="cadaluno")
public class CadAluno {
    @EJB
    private AlunoDao dao;

    private String nome;
    private String matricula;

    public String getNome() {
        return nome;
    }

    public void setNome(String nome) {
        this.nome = nome;
    }

    public String getMatricula() {
        return matricula;
    }

    public void setMatricula(String matricula) {
        this.matricula = matricula;
    }

    public String salvar() {
    }
}
```

```
<h:head>
<meta charset="UTF-8" />
<title>Cadastro Aluno</title>
</h:head>
<h:body>
    <f:view>
        <h:messages />
        <h:form>
            <h3>Cadastro de Alunos</h3><hr />

            <h:outputLabel value="Nome: " />
            <h:inputText value="#{cadaluno.nome}" size="20" />

            <h:outputLabel value="Matricula: " />
            <h:inputText size="8" value="#{cadaluno.matricula}" />

            <h:commandButton value="Enviar" action="#{cadaluno.salvar}" />
            <h:commandButton value="Voltar" action="menu.xhtml" />
        </h:form>
    </f:view>
</h:body>
</h:html>
```

Os métodos get e set do ManagedBean são utilizados pela página JSF.

Um `<h:selectOneMenu/>` é implementado segundo o exemplo ao lado.

```
<h:selectOneMenu value="#{cadaluno.curso}">
  <f:selectItems value="#{cadaluno.cursos}" var="item"
    itemLabel="#{item.nome}" itemValue="#{item.id}" />
</h:selectOneMenu>
```

Dos objetos retornados pelo método `getCursos()` são selecionados o label e o valor com os atributos `itemLabel` e `itemValue` respectivamente.

No atributo `value` em `selectOneMenu` é informado que receberá o valor do objeto selecionado.

```
private Integer curso;

public Integer getCurso() {
    return curso;
}

public void setCurso(Integer curso) {
    this.curso = curso;
}

public List<Curso> getCursos() {
    return dao.getCursos();
}
```

Um `<h:dataTable/>` é implementado de forma similar ao `for` do JSP, porém cada coluna deve ser definida utilizando o elemento `<h:column/>`.

```
private ListDataModel<Aluno> alunos;

public ListDataModel<Aluno> getAlunos() {
    alunos = new ListDataModel<>(dao.getAlunos());
    return alunos;
}

@SuppressWarnings("unchecked")
public void remove() {
    for (Aluno item : (List<Aluno>) alunos.getWrappedData()) {
        if (item.isDel())
            dao.delAluno(item.getId());
    }
}
```

```
<h:dataTable value="#{cadaluno.alunos}" var="aluno">
    <h:column>
        <f:facet name="header">Del</f:facet>
        <h:selectBooleanCheckbox value="#{aluno.del}" />
    </h:column>

    <h:column>
        <f:facet name="header">Nome</f:facet>
        <h:outputText value="#{aluno.nome}" />
    </h:column>
</h:dataTable>
```

Uma coleção do tipo `ListDataModel` deve ser utilizada para conter os objetos de um `<h:dataTable/>`.

É através do método `getWrappedData` que obtemos os dados retornados pela página JSF para efetuar ações.