

ADS - ANÁLISE E DESENVOLVIMENTO DE SISTEMAS

Desenvolvimento Web

Prof Me Cristiano Camilo – aula2



Objetivos de Aula

- Introdução ao desenvolvimento de aplicações Web
- Introdução ao JSF

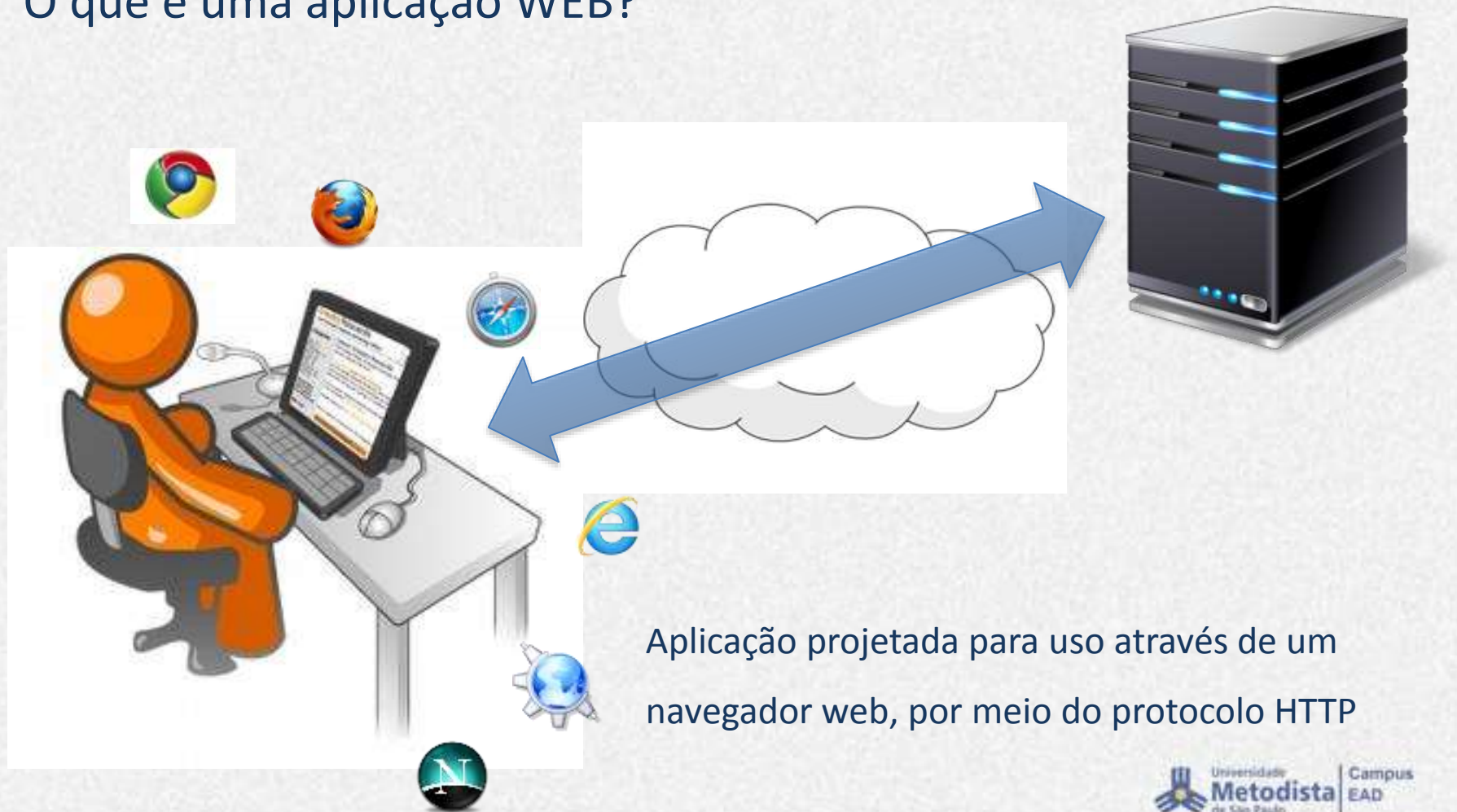


Introdução



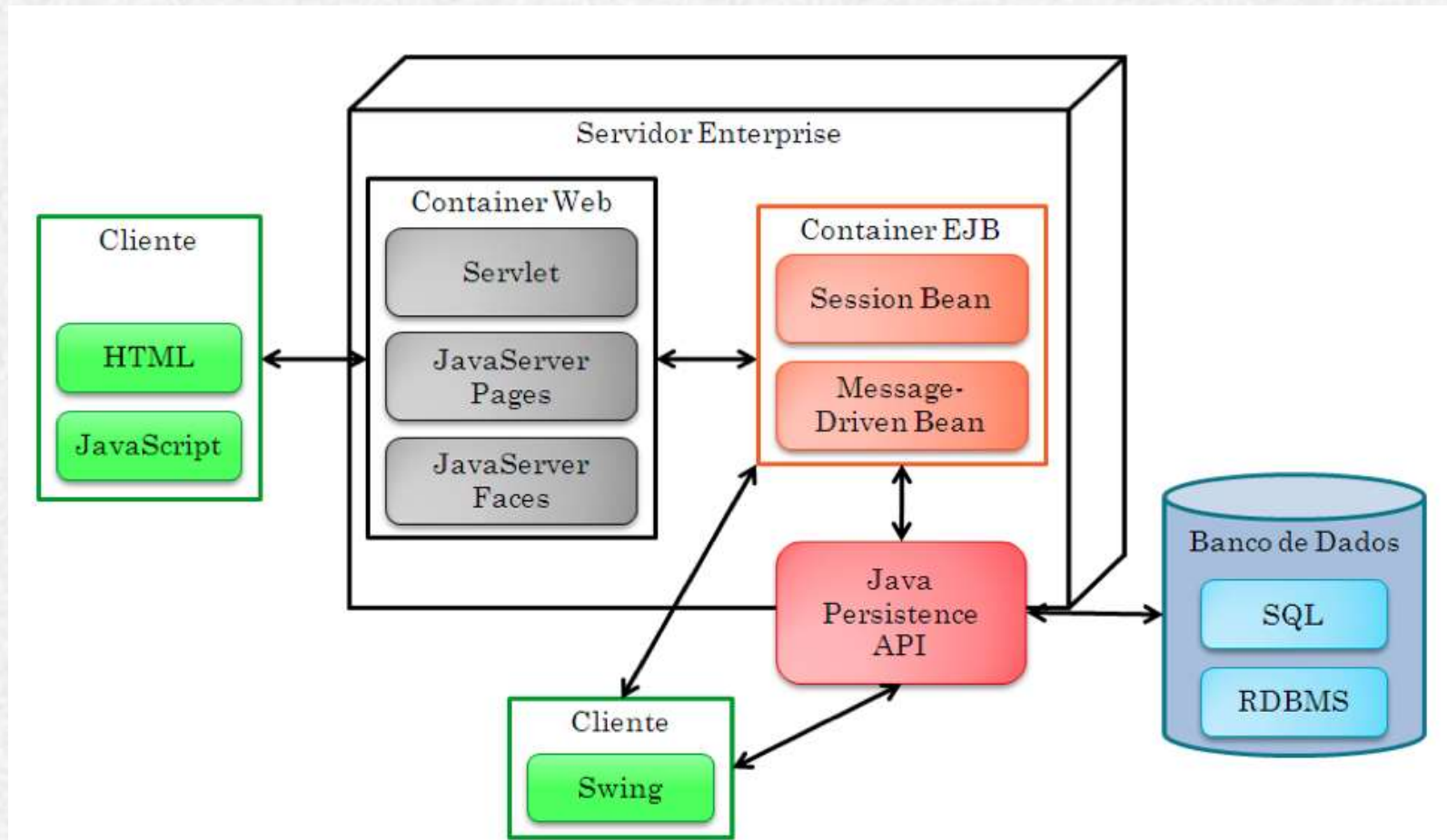
Introdução

O que é uma aplicação WEB?



Introdução

Modelo de arquitetura de uma aplicação:



Java Server Faces 2.0



Java Server Faces

Mas o que é JSF ?

- JavaServer Faces (JSF) - framework para desenvolvimento de aplicações web.

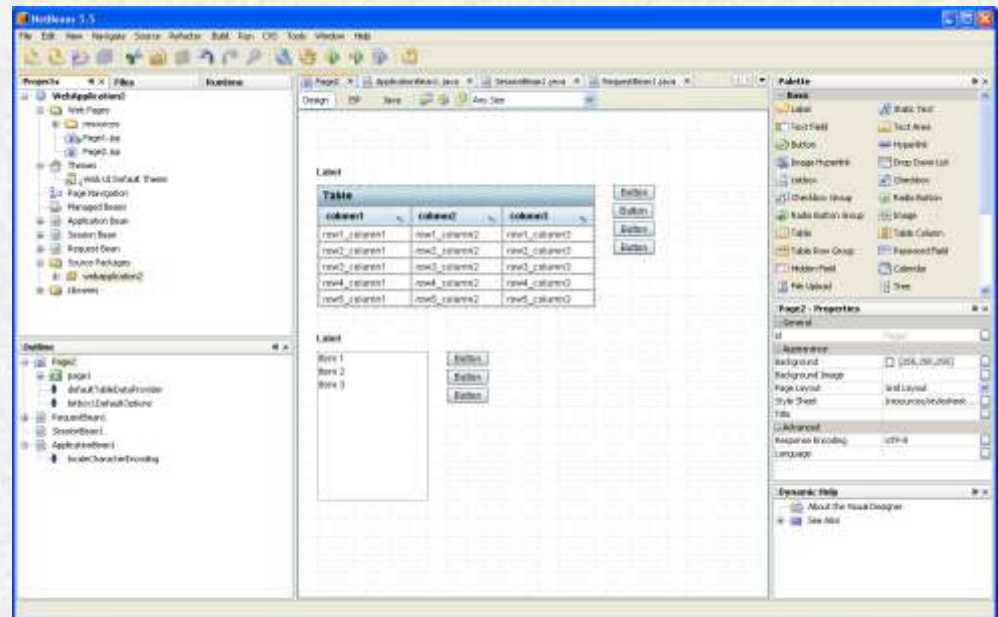
“Um Framework ou arcabouço conceitual é um conjunto de conceitos usado para resolver um problema de um domínio específico”



Java Server Faces

Mas o que é JSF ?

- Desenvolvimento baseado em componentes de interface gráfica similar ao Swing.



Java Server Faces

Mas o que é JSF ?

- Utiliza o padrão de projeto Model-View-Controller (MVC) para facilitar o desenvolvimento



Java Server Faces

Mas o que é JSF ?

- Dirigido a eventos



Java Server Faces

Resumindo...

“JavaServer Faces (JSF) é um framework de interface de usuário para desenvolvimento de aplicações web Java. Ele possui um design para facilitar significativamente o trabalho de escrever e manter aplicações web que rodam em servidores de aplicações Java e renderizam suas interfaces de usuário (telas) de volta para o cliente (navegador) solicitante.” [JCP]

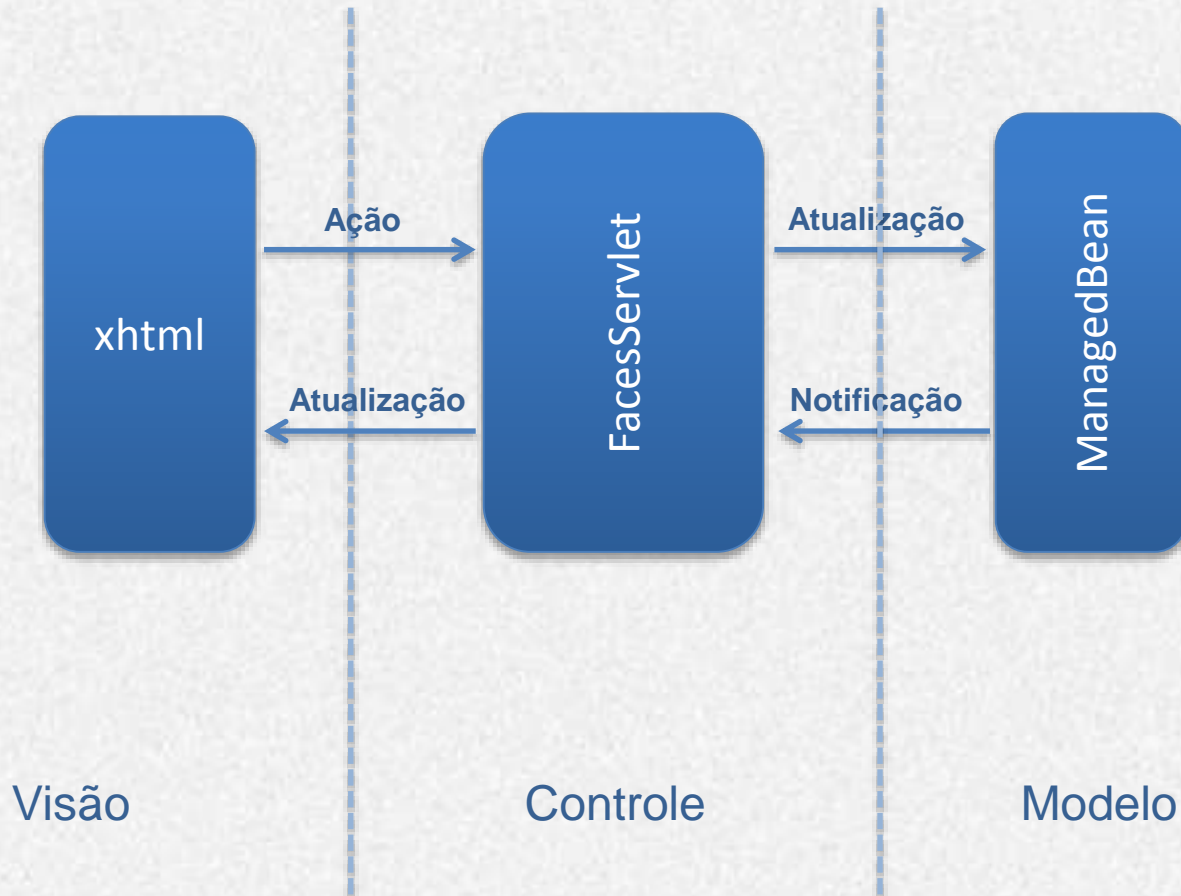
Java Server Faces

Modelo MVC no detalhe...

- O **View** é representado pelos arquivos .xhtml, que serão responsáveis apenas pela renderização das páginas.
- O **Model** é representado pelo ManagedBean, que é uma classe que executa as ações da página e possui os atributos que serão apresentados pela tela.
- E o **Controller** é gerenciado pelo JSF, mais especificamente pela classe FacesServlet, que redireciona as chamadas de páginas, faz as conversões das tags JSF para HTML, entre outras funcionalidades.

Java Server Faces

Modelo MVC no detalhe...



Java Server Faces

Histórico do JSF...

- JSF 1.0 (2004)
 - JSF 1.1 (2004)
 - JSF 1.2 (2006)
 - JSF 2.0 (2009)
 - JSF 2.1 (2010)
 - JSF 2.2 (2013)
- JSP para composição de templates
- Facelets para composição de templates

Versão mais atual: JSF 2.2.9, Dezembro de 2014

Java Server Faces

Porque utilizar JSF ?

- Facilidade de desenvolver aplicações web:
 - Desenvolvimento das telas
 - Atribuição de valores a objetos
 - Conversão
 - Validação
 - Internacionalização
 - Template

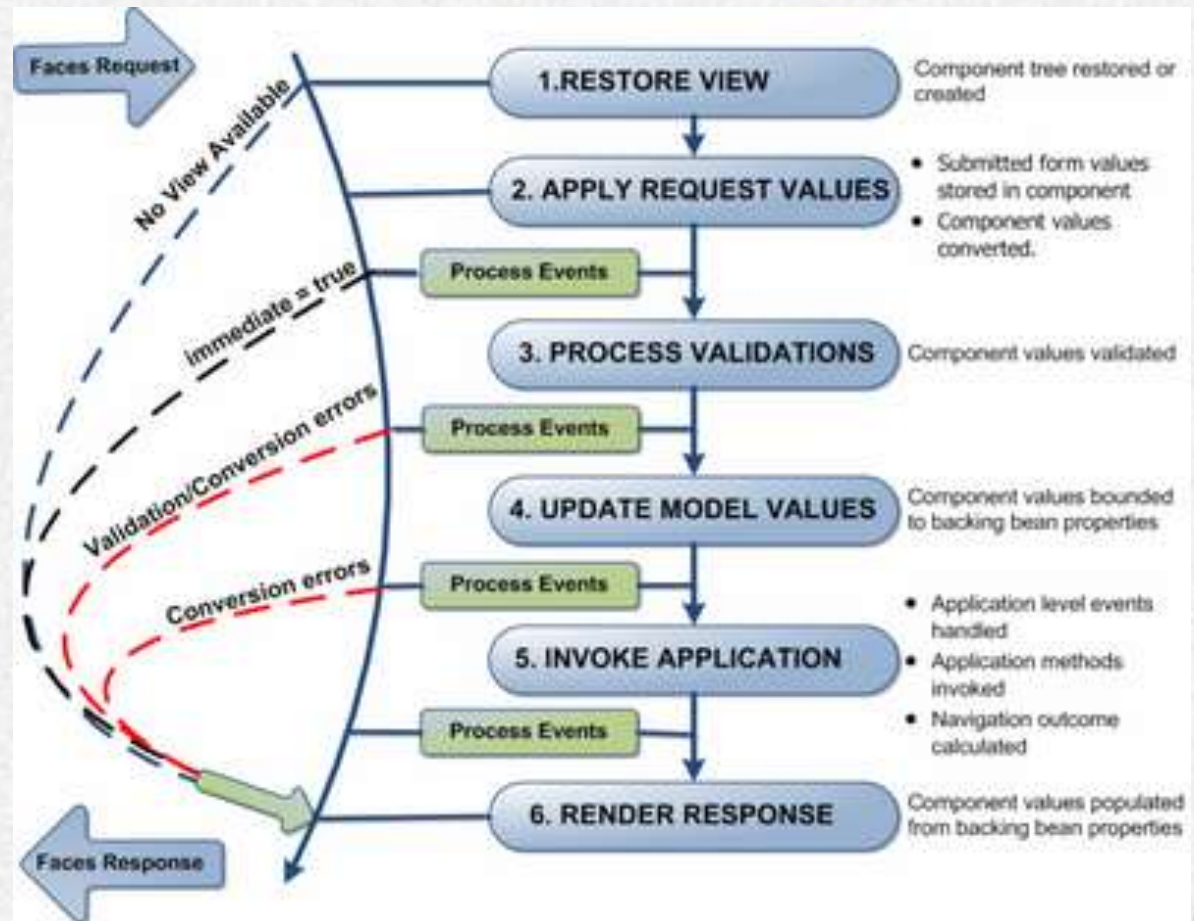
Java Server Faces

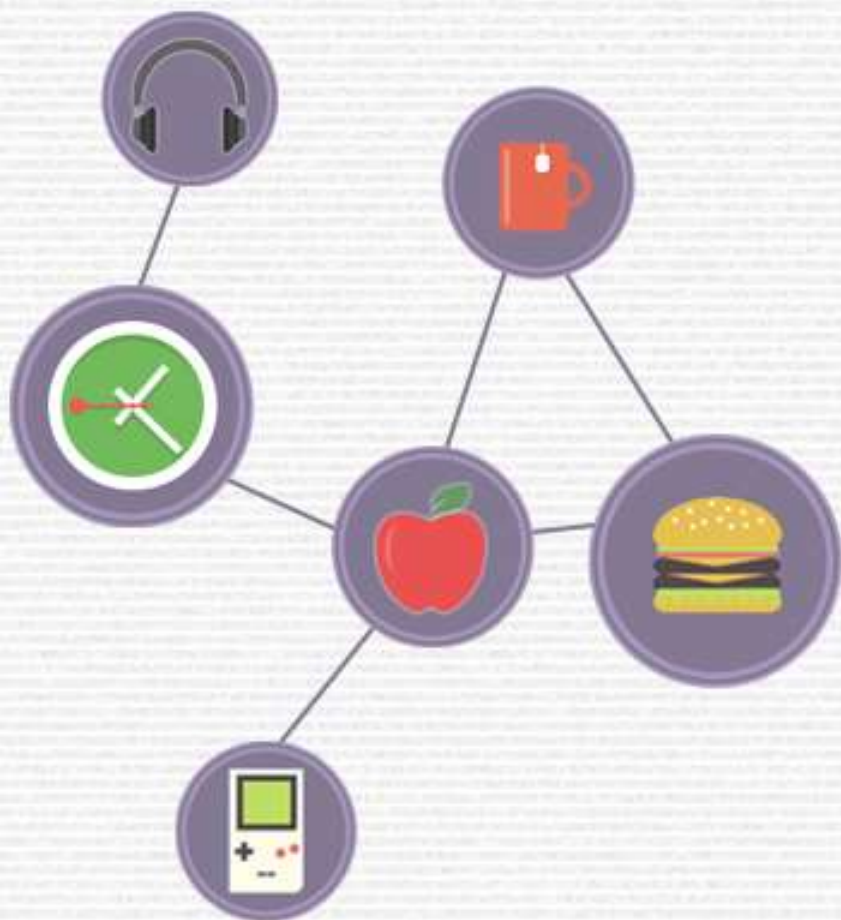
Porque utilizar JSF ?



Java Server Faces

O Ciclo de vida do JSF...





Intervalo



Universidade
Metodista
de São Paulo

Campus
EAD

Java Server Faces

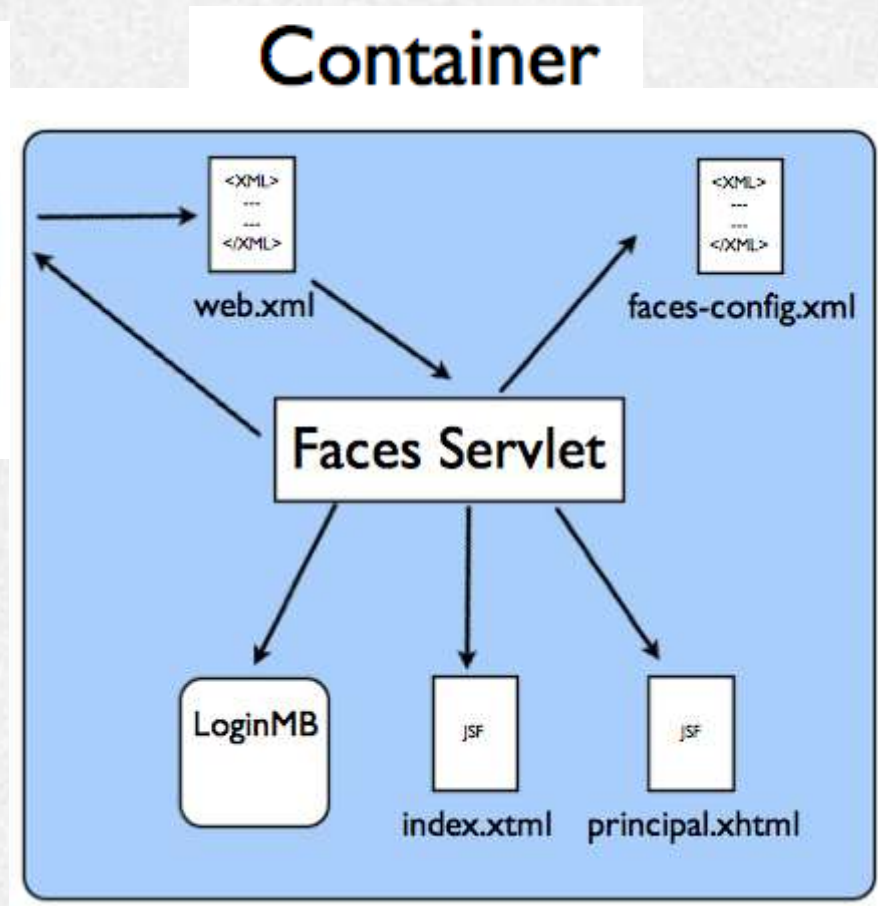
Exemplo de uma aplicação JSF:

```
<?xml version='1.0' encoding='UTF-8' ?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml"
  xmlns:h="http://java.sun.com/jsf/html">
  <h:head>
    <title>Hello World!</title>
  </h:head>
  <h:body>
    <h1><h:outputText value="Hello World com JSF 2.0"/></h1>
  </h:body>
</html>
```



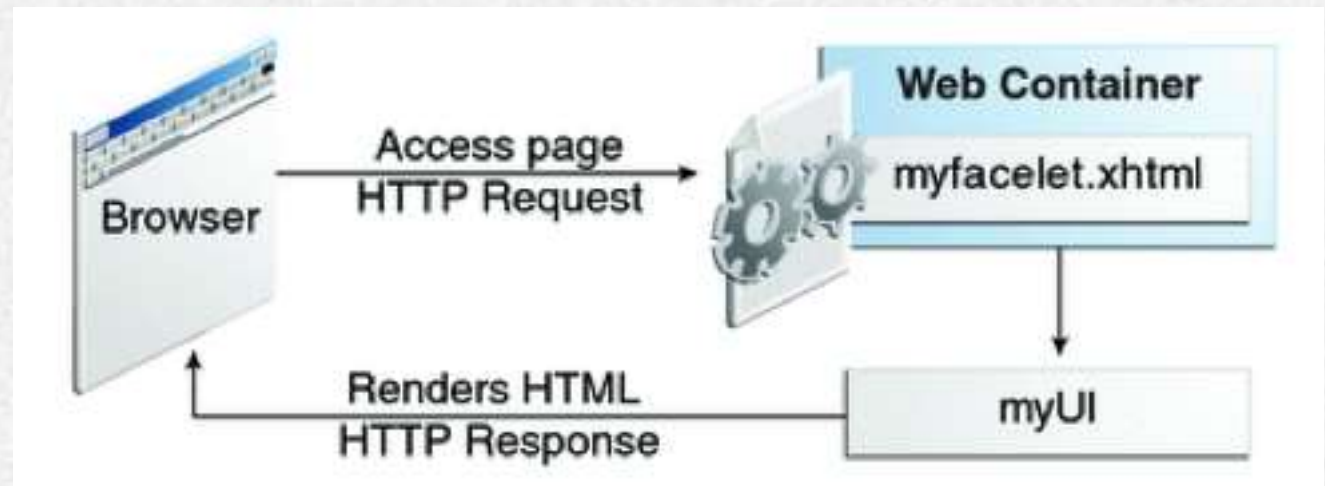
Java Server Faces

Estrutura básica de uma aplicação JSF



Java Server Faces

Quando o cliente (browser) solicita uma página JSF, o FacesServlet processa os componentes JSF da página e uma resposta HTML ou XHTML é retornada.



Java Server Faces

O Arquivo **web.xml**:

Deve estar contido dentro da pasta WEB-INF !

Nele, podemos informar ao Container:

- Página principal da aplicação
- Parâmetros adicionais para aplicação
- Faces Servlet

Java Server Faces

Exemplo de um arquivo **web.xml**:

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app version="3.0" 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_3_0.xsd">
  <context-param>
    <param-name>javax.faces.PROJECT_STAGE</param-name>
    <param-value>Development</param-value>
  </context-param>
  <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>/faces/*</url-pattern>
  </servlet-mapping>
  <session-config>
    <session-timeout>
      30
    </session-timeout>
  </session-config>
  <welcome-file-list>
    <welcome-file>faces/index.xhtml</welcome-file>
  </welcome-file-list>
</web-app>
```


Java Server Faces

O Arquivo **faces-config.xml**:

Deve estar contido dentro da pasta WEB-INF !

Nele, podemos informar ao Container:

- Arquivo de internacionalização “.properties”
- Classes que são Managed Bean (não necessário no JSF 2.0)
- Regras de navegação entre páginas
- Conversores
- Componentes adicionais e de terceiros

Java Server Faces

A black laptop is centered in the image. Its screen is white and displays the text "Hello World em JSF" in a black, sans-serif font. Behind the laptop, there is a faint, gray silhouette of a person wearing a suit and tie, with their hands positioned as if holding the laptop. The background is a light gray with a subtle grid pattern.

Hello World em JSF

Java Server Faces

O Managed Bean

- O Managed Bean serve para controlar a lógica de negócio das telas.
- Dentro do Managed Bean temos:
 - Atributos que podem representar os dados dinâmicos das telas.
 - Métodos que representam ações ou eventos da tela.

Java Server Faces

Para se criar um Managed Bean, no JSF 2.0, basta utilizar a anotação `@ManagedBean`

```
package br.metodista.ads4;  
  
import javax.faces.bean.ManagedBean;  
  
@ManagedBean  
public class ExemploMB {  
  
}
```

Java Server Faces

Para se referenciar um Managed Bean em uma página xmhtml

```
<h:outputText id="senhaText" value="Senha: " />
<h:inputSecret id="senha" label="Senha"
               value="#{loginMB.usuario.senha}" required="true" />
```



```
@ManagedBean
public class LoginMB {
    private Usuario usuario = new Usuario();

    public Usuario getUsuario() {
        return usuario;
    }

    public void setUsuario(Usuario usuario) {
        this.usuario = usuario;
    }
}
```

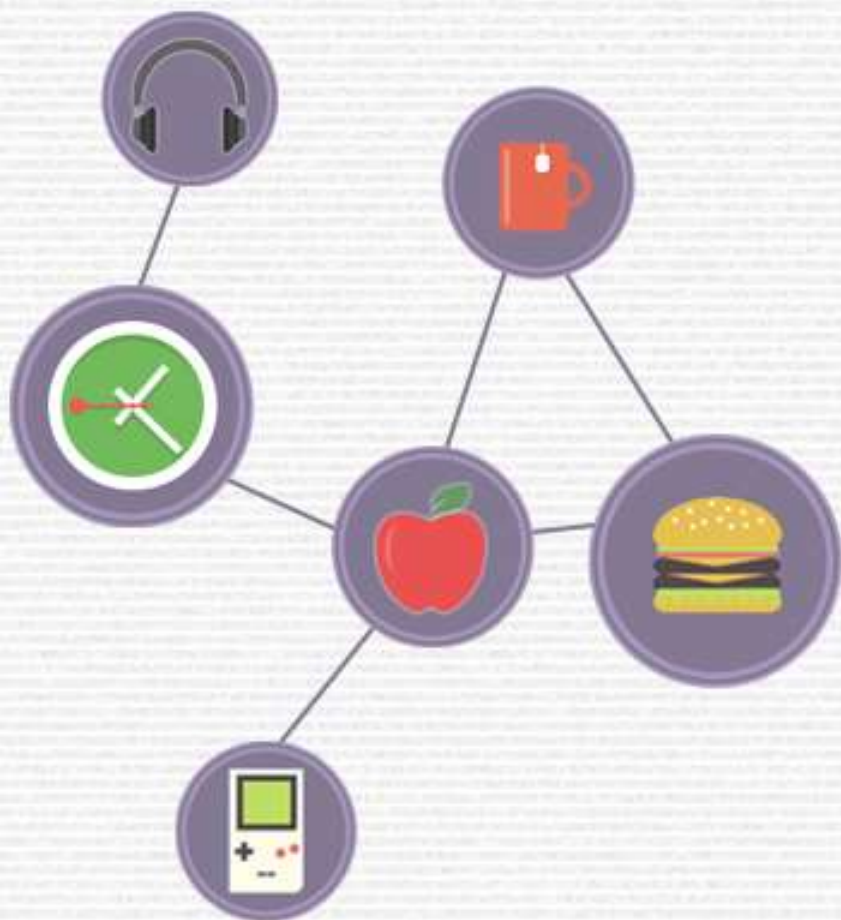
Java Server Faces

Devolvendo mensagens e redirecionamentos para o xhtml:

```
public String doEfetuarLogin() {  
    if("sakurai".equals(usuario.getLogin())  
        && "123".equals(usuario.getSenha()))  
        return "principal";  
    else {  
        FacesMessage msg = new FacesMessage("Usuário ou senha inválido!");  
        FacesContext.getCurrentInstance().addMessage("erro", msg);  
        return null;  
    }  
}
```

Página de
retorno

Mensagem
de retorno



Intervalo



Universidade
Metodista
de São Paulo

Campus
EAD

Java Server Faces

Bibliotecas de tags (taglibs)

É um conjunto de tags ou marcações, que podem ser utilizadas no desenvolvimento.



Java Server Faces

Algumas Bibliotecas de tags

<http://java.sun.com/jsf/html>

Possui os componentes básicos para renderização de telas em HTML.

<http://java.sun.com/jsf/core>

Possui os componentes básicos para desenvolver as telas em JSF independente da forma de renderização.

<http://java.sun.com/jsf/facelets>

A biblioteca facelets possui de templates e componentes

Java Server Faces

Biblioteca: <http://java.sun.com/jsf/html>

Exemplo de tags:

- form
- inputText
- outputText
- commandButton
- commandLink
- messages
- selectOneMenu
- panelGrid
- panelGroup
- dataTable
- column

Java Server Faces

Biblioteca: <http://java.sun.com/jsf/html>

<h:form>

- A tag **form** é utilizada para definir uma área que possui um formulário, onde podemos ter campos de texto, botões, links e outros.

- Exemplo:

```
<h:form id="formMenu">
```

 Conteúdo do formulário...

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

Java Server Faces

Biblioteca: <http://java.sun.com/jsf/html>

<h:inputText>

- A tag **inputText** é utilizada para definir um campo de texto onde o usuário pode entrar com alguma informação.

- Exemplo:

```
<h:inputText id="titulo" value="#{livroMB.livro.titulo}"  
size="100" maxlength="150" />
```


Java Server Faces

Biblioteca: <http://java.sun.com/jsf/html>

<h:inputSecret>

- A tag **inputSecret** é utilizada para definir um campo de texto onde o usuário pode entrar com alguma informação, mas o que é digitado não aparece legível na tela.

- Exemplo:

```
<h:inputSecret id="senha"
value="#{loginMB.usuario.senha}" size="100"
maxlength="30" />
```

Java Server Faces

Biblioteca: <http://java.sun.com/jsf/html>

<h:outputText>

- A tag **outputText** é utilizada para definir um texto (label).

- Exemplo:

```
<h:outputText id="ola"  
value="Olá #{loginMB.usuario.nome}" />
```

Java Server Faces

Biblioteca: <http://java.sun.com/jsf/html>

<h:commandLink>

- A tag **commandLink** é utilizada para definir um link.

- Exemplo:

```
<h:commandLink id="linkLivro" value="Cadastro de Livro"  
action="#{menuMB.doCadastroLivro}" />
```


Java Server Faces

Biblioteca: <http://java.sun.com/jsf/html>

<h:commandButton>

- A tag **commandButton** é utilizada para definir um botão.

- Exemplo:

```
<h:commandButton id="botaoSalvar" value="Salvar"  
action="#{livroMB.doSalvarLivro}" />
```

Java Server Faces

Biblioteca: <http://java.sun.com/jsf/html>

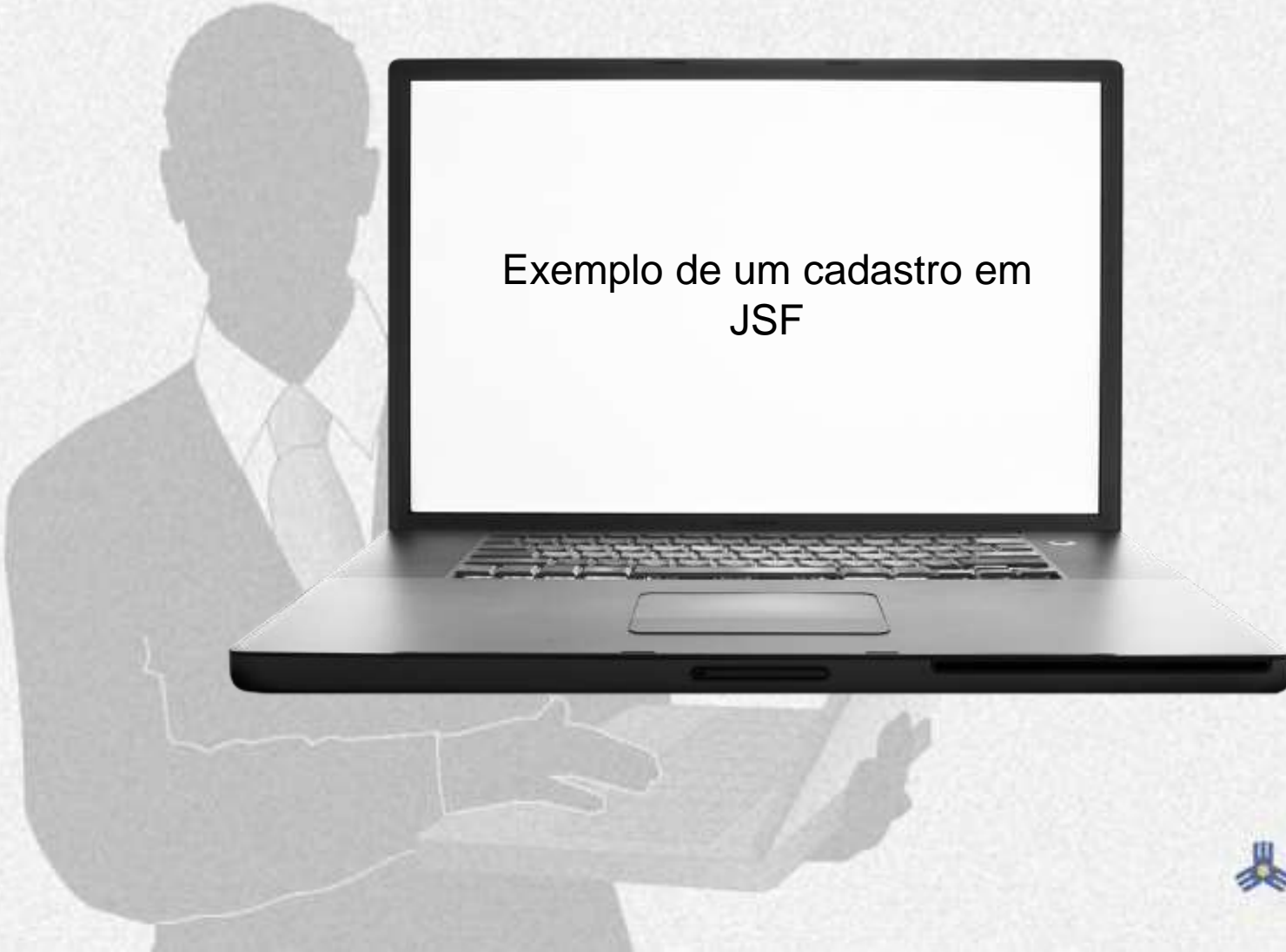
<h:messages>

- A tag **messages** é utilizada para imprimir mensagens de informação, aviso ou erro.

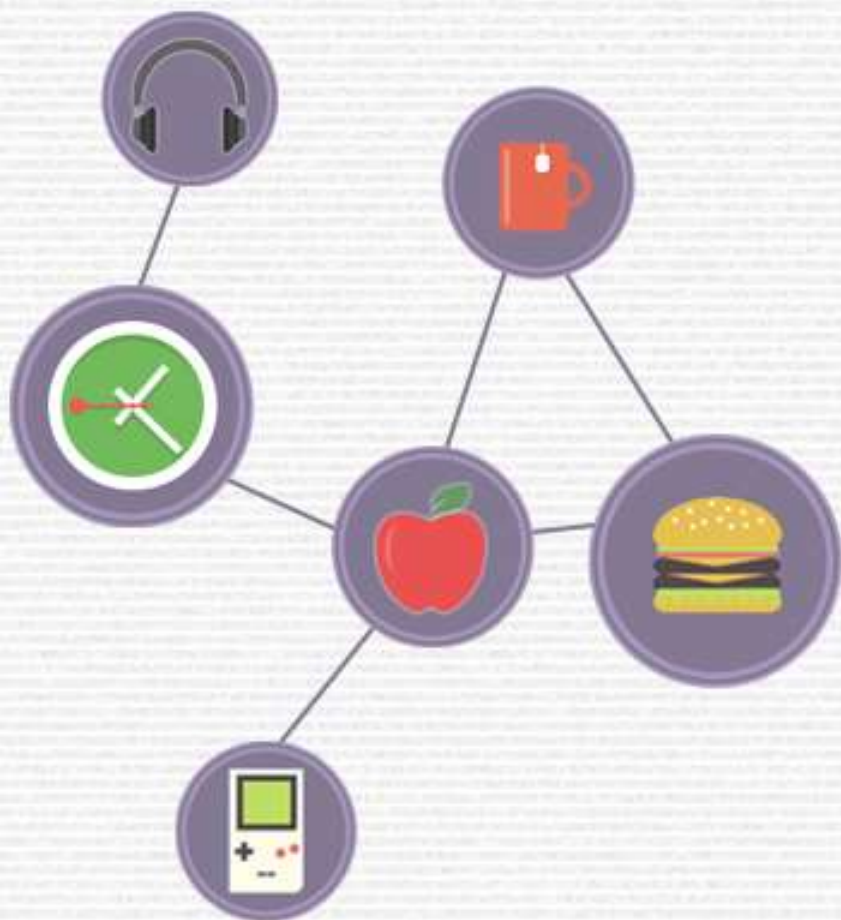
- Exemplo:

```
<h:messages id="mensagens" showSummary="true"  
layout="table" />
```

Exemplo



Exemplo de um cadastro em
JSF



FIM



Universidade
Metodista
de São Paulo

Campus
EAD