

**A D S -ANÁLISE E DESENVOLVIMENTO DE SISTEMAS**

## **Desenvolvimento Web**

**Prof. Me. Cristiano Camilo – Aula 3**



# Aula de hoje...

- Facelets
- Internacionalização



# Facelets



# Facelets

O que são Templates?



Ambiente estabelecido como modelo,  
permitindo criar conteúdos de forma rápida.

# Facelets

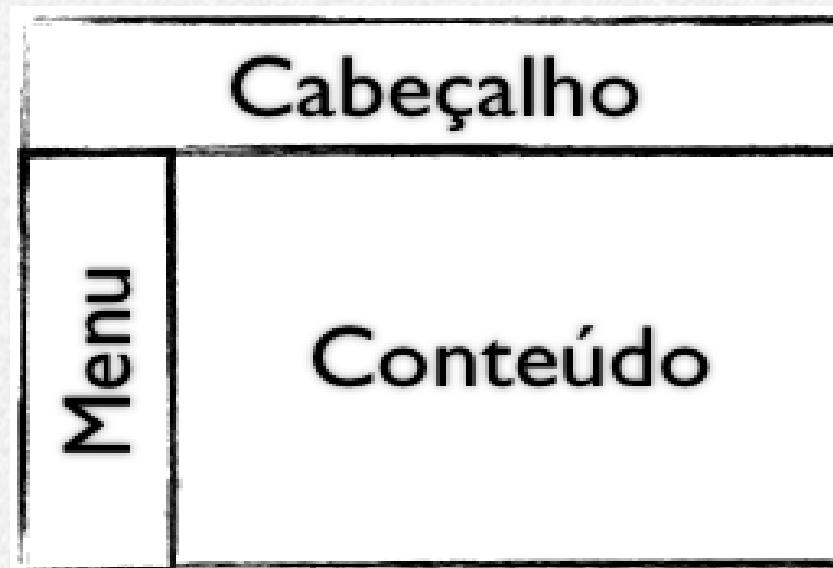
O que são Facelets?



Templates são layouts de páginas que podem ser reaproveitadas a partir do momento em que uma página padrão (template) é criada.

# Facelets

Podemos, assim, criar uma página com uma estrutura mais básica que será aproveitada e customizada por outras páginas.



# Facelets

Na versão 2.0 do JSF, foi incorporado o projeto Facelets que possui tags para criação de templates. Para utilizá-lo, basta importar a biblioteca de tags <http://java.sun.com/jsf/facelets>, por exemplo:

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

# Facelets

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

As Tags da biblioteca Facelets:

- insert
- composition
- define
- component
- decorate
- fragment
- debug
- include
- param
- remove
- repeat

# Facelets

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

**<ui:insert>**

- A tag **insert** permite delimitar áreas que podem ser alteradas pelas páginas que seguirão este template.

- Exemplo:

```
<ui:insert name="conteudo">
    <h1>Template Web</h1>
</ui:insert>
<ui:insert name="conteudo">
    Aula de JSF 2.0 - criando templates de página
</ui:insert>
<ui:insert name="rodape">
    Universidade Metodista de São Paulo
</ui:insert>
```

# Facelets

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

## <ui:composition>

- A tag **composition** é utilizada para definir qual template uma página está utilizando.

- Exemplo:

```
<html xmlns="http://www.w3.org/1999/xhtml"
      xmlns:h="http://java.sun.com/jsf/html"
      xmlns:ui="http://java.sun.com/jsf/facelets">
<ui:composition template="/principal.xhtml">
    Universidade Metodista de São Paulo
</ui:composition >
</html>
```

# Facelets

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

## <ui:define>

- A tag **define** é utilizada para substituir as áreas de um template (tag insert) com novo conteúdo.

- Exemplo:

```
<html xmlns="http://www.w3.org/1999/xhtml"
      xmlns:h="http://java.sun.com/jsf/html"
      xmlns:ui="http://java.sun.com/jsf/facelets">
<ui:composition template="/principal.xhtml">
    <ui:define name="conteudo">
        <h1> Cadastro de pessoas </h1>
    </ui:define >
</ui:composition >
</html>
```

# Facelets

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

## <ui:include>

- A tag **include** é utilizada para se adicionar um arquivo xhtml dentro de outro

- Exemplo:

```
<h:body>
```

```
    <div id="page">
```

```
        <div id="header">
```

```
            <ui:insert name="header" >
```

```
                <ui:include
```

```
src="/template/cabecalhoPadrao.xhtml" />
```

```
            </ui:insert>
```

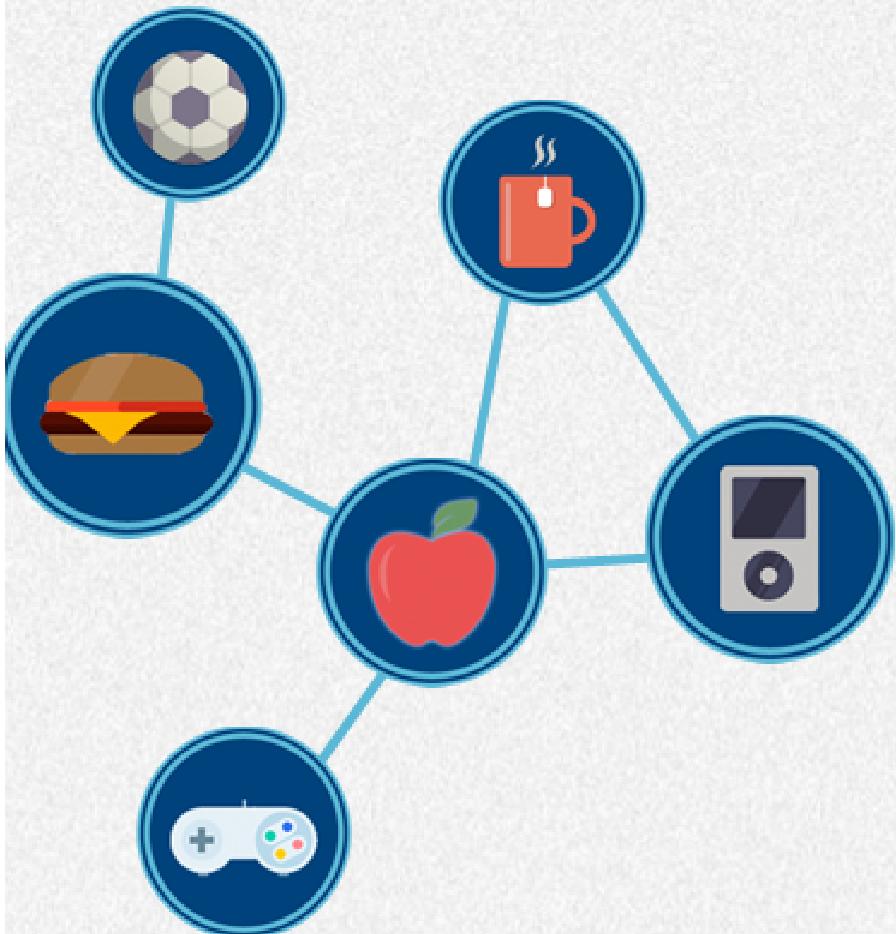
```
        </div>
```

```
</h:body>
```

# Exemplo



Exemplo de  
criação e utilização  
de Facelets



# Intervalo

# Facelets

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

## <ui:component>

- A tag **component** é idêntica à tag composition, com a diferença que ela não gera um template e adiciona o componente diretamente à árvore.

- Exemplo:

```
<html xmlns="http://www.w3.org/1999/xhtml"
      xmlns:h="http://java.sun.com/jsf/html"
      xmlns:ui="http://java.sun.com/jsf/facelets">
<ui:composition template="/principal.xhtml">
    Universidade Metodista de São Paulo
</ui:composition >
</html>
```

# Facelets

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

## <ui:decorate>

- A tag **decorate** é idêntica à tag composition, mas não desconsidera todo o texto fora de sua tag, sendo muito eficaz quando se deseja decorar algo.

- Exemplo:

```
<html xmlns="http://www.w3.org/1999/xhtml"
      xmlns:h="http://java.sun.com/jsf/html"
      xmlns:ui="http://java.sun.com/jsf/facelets">
<ui:composition template="/principal.xhtml">
    Universidade Metodista de São Paulo
</ui:composition >
</html>
```

# Facelets

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

## <ui:fragment>

- A tag **decorate** é idêntica à tag component, mas não desconsidera todo o texto fora da tag.

- Exemplo:

```
<html xmlns="http://www.w3.org/1999/xhtml"
      xmlns:h="http://java.sun.com/jsf/html"
      xmlns:ui="http://java.sun.com/jsf/facelets">
<ui:composition template="/principal.xhtml">
    Universidade Metodista de São Paulo
</ui:composition >
</html>
```

# Facelets

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

## <ui:debug>

- A tag **debug** tem a função de exibir a árvore de componentes do JSF e as variáveis de escopo.

- Exemplo:

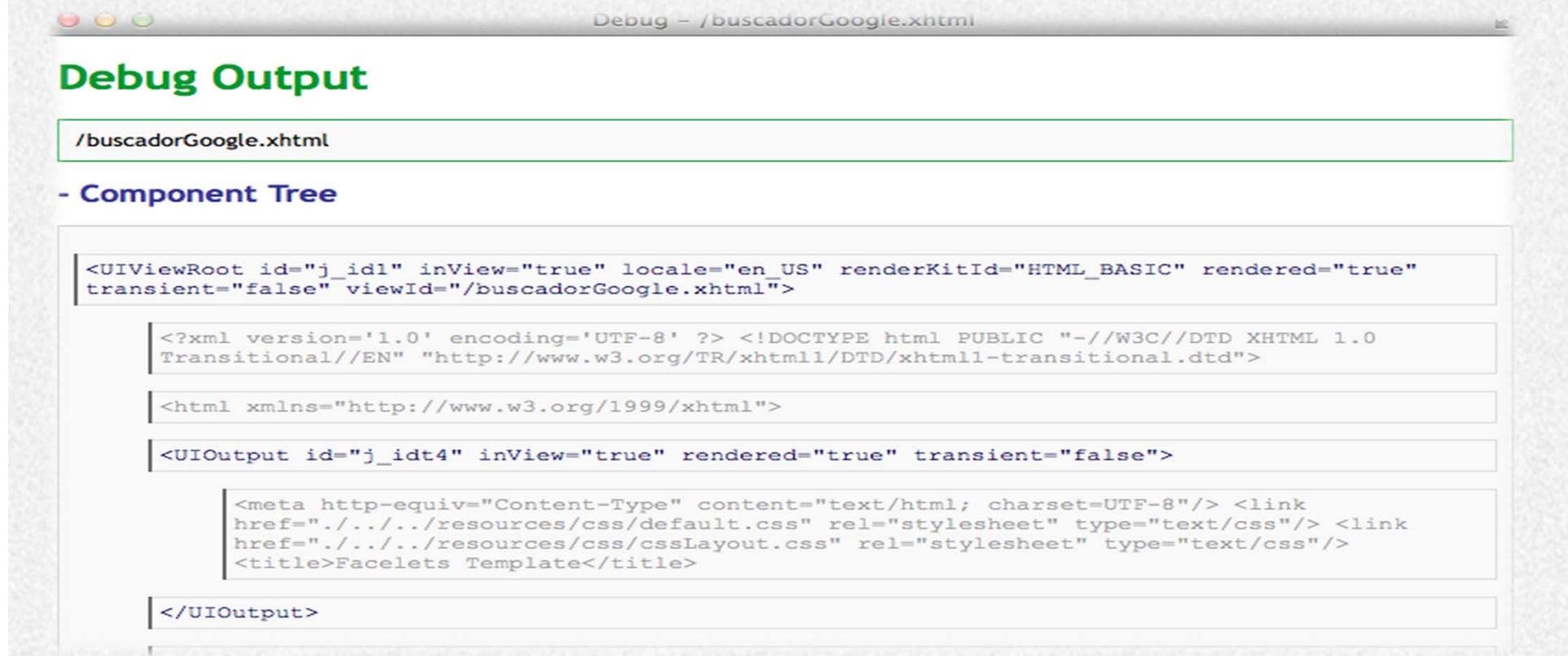
```
<html xmlns="http://www.w3.org/1999/xhtml"
      xmlns:h="http://java.sun.com/jsf/html"
      xmlns:ui="http://java.sun.com/jsf/facelets">
    <ui:composition template="/principal.xhtml">
        Universidade Metodista de São Paulo
    </ui:composition >
    <ui:debug hotkey="p" rendered="true"/>
</html>
```

# Facelets

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

**<ui:debug hotkey="p" rendered="true"/>**

Neste caso, a sequência de atalho para o debug é "CTRL + SHIFT + P", mas o padrão é "CTRL + SHIFT + D"



The screenshot shows a Java IDE interface with a title bar "Debug - /buscadorGoogle.xhtml". Below the title bar, there's a green header bar with the text "Debug Output". Underneath, a code editor displays the XML structure of the Facelets page. The code includes the UIViewRoot component and its children: the XML declaration, the html element, and the UIOutput component which contains meta, link, and title tags.

```
<UIViewRoot id="j_id1" inView="true" locale="en_US" renderKitId="HTML_BASIC" rendered="true"
transient="false" viewId="/buscadorGoogle.xhtml">

<?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">

<UIOutput id="j_idt4" inView="true" rendered="true" transient="false">

<meta http-equiv="Content-Type" content="text/html; charset=UTF-8"/> <link
href=".//resources/css/default.css" rel="stylesheet" type="text/css"/> <link
href=".//resources/css/cssLayout.css" rel="stylesheet" type="text/css"/>
<title>Facelets Template</title>

</UIOutput>
```

# Facelets

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

## <ui:param>

- A tag **param** serve para que se passem parâmetros para uma página ligada por include ou para um template.

- Exemplo com include:

```
<ui:insert name="header" >
  <ui:include src="/template/common/cabecalho.xhtml">

    <ui:param name="tagLine" value="Parâmetros com JSF! " />

  </ui:include>
</ui:insert>
```

# Facelets

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

**<ui:param>**

```
<?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:ui="http://java.sun.com/jsf/facelets">
<body>
    <ui:composition>
        <h2>Tag Line : #{tagLine}</h2>
    </ui:composition>
</body>
</html>
```

# Facelets

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

## <ui:param>

- A tag **param** serve para que se passem parâmetros para uma página ligada por include ou para um template.
- Exemplo com composition:

```
<ui:composition template="template/common/layoutComum.xhtml">  
    <ui:param name="nomeDoArquivo" value="default.xhtml" />  
</ui:composition>
```

# Facelets

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

<ui:param>

```
<?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"
      xmlns:ui="http://java.sun.com/jsf/facelets">
<h:body>
    <div id="content">
        Arquivo : #{nomeDoArquivo}
    </div>
</h:body>
</html>
```

# Facelets

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

## <ui:remove>

- A tag **remove** faz com que todos os componentes dentro dela não sejam renderizados.

- Exemplo:

```
<body>
```

```
    <h:form>
```

```
        <h:outputText value="O botão abaixo não será exibido" />
```

```
        <ui:remove>
```

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

```
        </ui:remove>
```

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

```
</body>
```

# Facelets

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

## <ui:repeat>

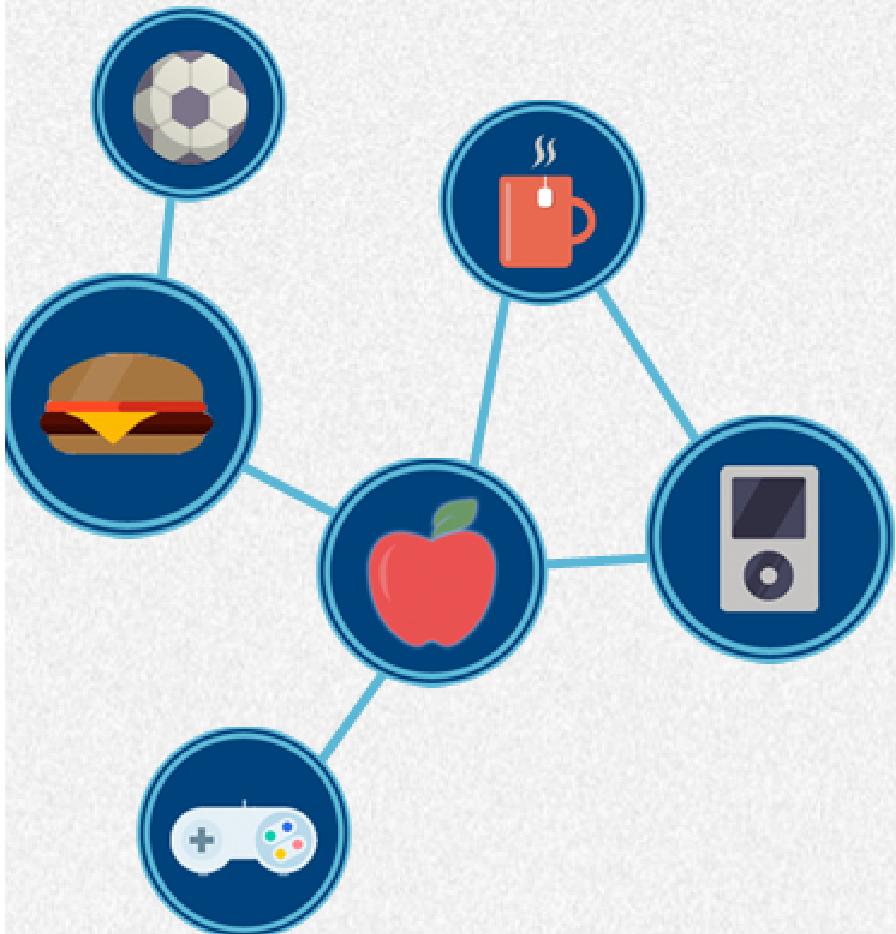
- A tag **repeat** determina um laço de repetição. Pode substituir as tags h:DataTable e c:forEach
- Exemplo:

```
<table>
<ui:repeat var="o" value="#{pedido_pedidosList}" varStatus="status">
    <tr>
        <td>#{o.numeroDoPedido}</td>
        <td>#{o.produto}</td>
        <td>#{o.preco}</td>
        <td>#{o.quantidade}</td>
    </tr>
</ui:repeat>
</table>
```

# Exemplo



Exemplo de  
utilização de mais  
tags Facelets



# Intervalo

# Internacionalização



# Internacionalização

Mas o que é Internacionalização?

O conceito de desenvolver uma aplicação multi-idioma e sem interferir drasticamente no código, separando todos os textos que são apresentados nas telas do mesmo.



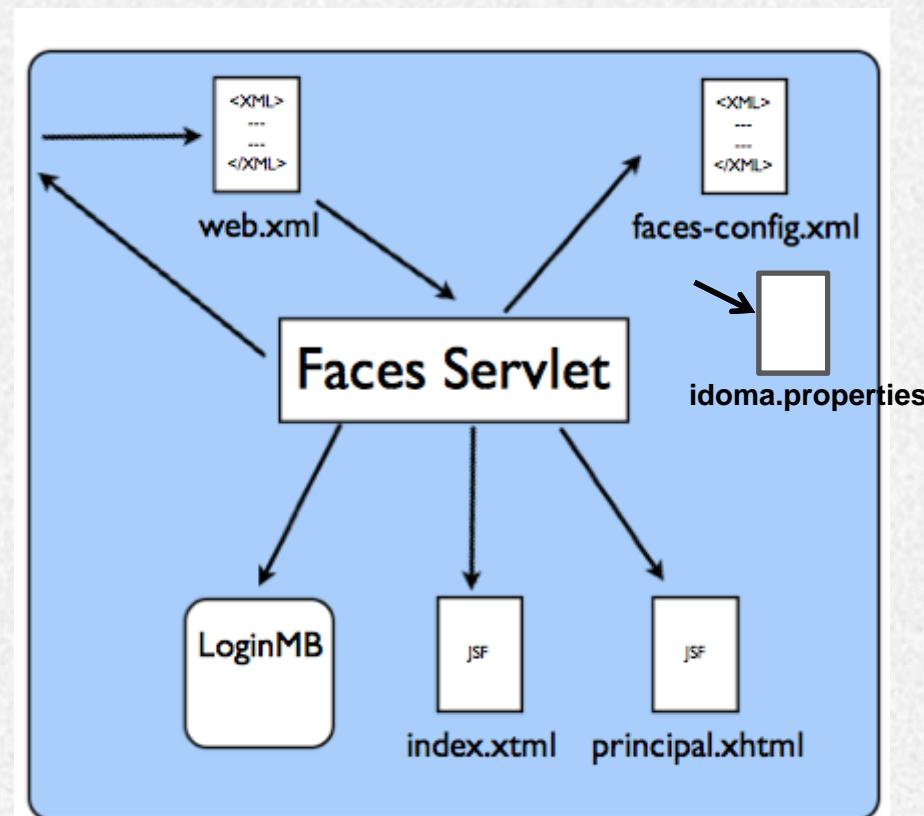
# Internacionalização

## Internacionalização em JSF:

- A i18n é uma funcionalidade que o framework JSF da para a aplicação poder ser desenvolvida em mais de um idioma.
- Utilizamos arquivos .properties para guardar todos os textos que serão apresentados em tela.

# Internacionalização

Estrutura básica de uma aplicação JSF com internacionalização:



# Internacionalização

Com isso, a exibição da página depende do consumidor:



# Internacionalização

O que é um arquivo .properties?

- É um arquivo que trabalha com o conceito de associação de chaves e valores:

Exemplo:

nome=Cristiano

sobrenome=Camilo

endereco.logradouro=Rua do Sacramento

endereco.cep=09641000

Podemos utilizar  
nomes com  
pontos!

# Internacionalização

Exemplo de arquivos de propriedades:

livraria\_pt.properties

```
1 titulo=Livraria Web
2 pesquisar=Pesquisar
3 novo=Novo
4 salvar=Salvar
5
6 menu.cadastroLivro=Cadastro de Livro
7 menu.livrosVendidos=Livros Vendidos
8
9 livro.cadastroLivro=Cadastro de Livro
10 livro.autor=Autor
11 livro.isbn=ISBN
12 livro.titulo=Titulo
13 livro.editora=Editora
14 livro.numPaginas=N m p ginas
15 livro.ano=Ano
16 livro.preco=Pre o
17
18 venda.vendaLivros=Venda de Livros
19 venda.cliente=Cliente
20 venda.titulo=Titulo
21 venda.preco=Pre o
22 venda.total=Total
```

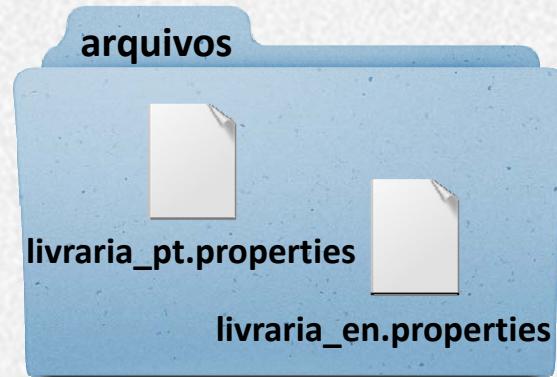
livraria\_en.properties

```
1 titulo=Bookstore Web
2 pesquisar=Search
3 novo>New
4 salvar=Save
5
6 menu.cadastroLivro=Register Books
7 menu.livrosVendidos=Book Sales
8
9 livro.cadastroLivro=Register Books
10 livro.autor=Author
11 livro.isbn=ISBN
12 livro.titulo=Title
13 livro.editora=Publisher
14 livro.numPaginas=Pages
15 livro.ano=Year
16 livro.preco=Price
17
18 venda.vendaLivros=Book Sales
19 venda.cliente=Client
20 venda.titulo=Title
21 venda.preco=Price
22 venda.total=Total
```

# Internacionalização

Preparando sua aplicação para trabalhar com internacionalização por meio do arquivo faces-config.xml

```
<application>
<resource-bundle>
    <base-name>arquivos.livraria</base-name>
    <var>m</var>
</resource-bundle>
    <locale-config> <default-locale>pt</default-locale>
    <supported-locale>pt</supported-locale>
    <supported-locale>en</supported-locale>
</locale-config>
```



# Internacionalização

## Utilizando a internacionalização em páginas JSF

```
<h:outputText value="#{m['livro.titulo']}
```

```
<f:selectItem itemLabel="#{m['livro.autor']}" itemValue="#{m['livro.autor']}
```

```
<h:commandButton value="#{m['salvar']}" action="#{livroMB.doSalvarLivro}"/>
```

```
<h:commandLink value="#{m['menu.cadastroLivro']}" action="#{menuMB.doCadastroLivro}"/>
```

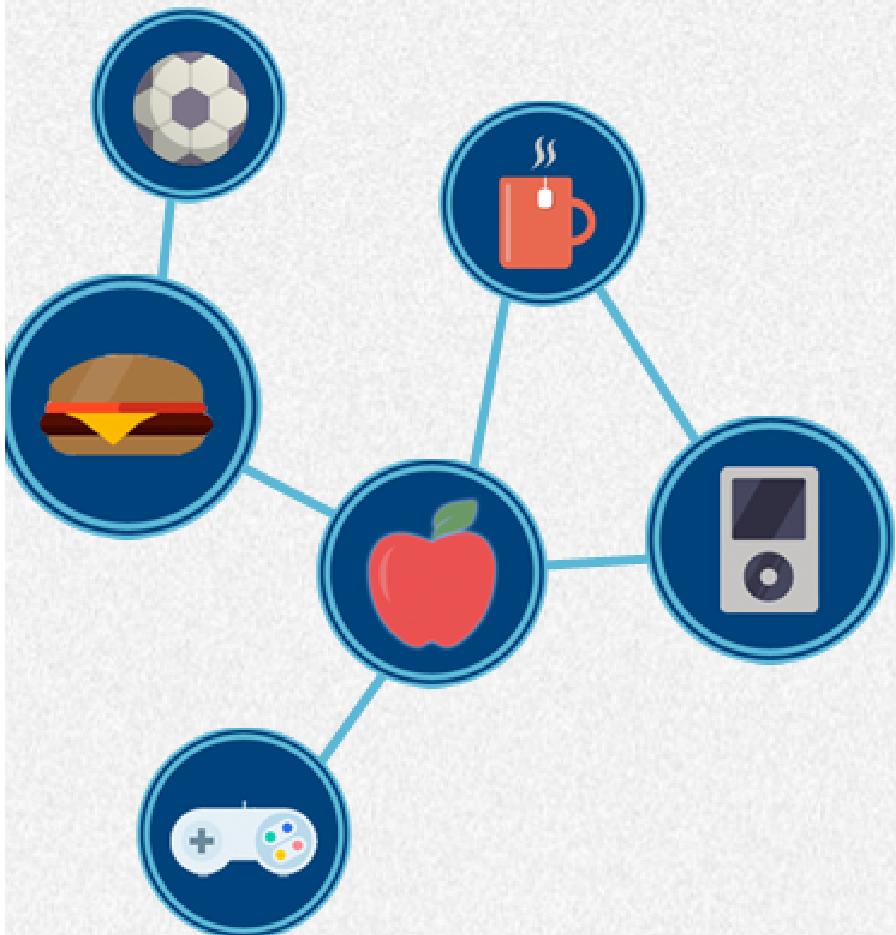


Variável definida  
no faces-config

# Exemplo



Exemplo de aplicação  
com  
internacionalização



Fim