


# Interfaces e Tabelas de Dados

Prof. Leonardo Vianna do Nascimento  
Disciplina de Desenvolvimento de Sistemas I

# JSF e CSS

---

- Componentes HTML presentes em páginas XHTML podem ser formatados normalmente
- Componentes JSF podem ser formatados utilizando seletores de ID ou seletores de classe
  - Outros seletores não podem ser usados, pois componentes JSF são convertidos em um ou mais elementos HTML

# Seletores de ID

---

- É possível formatar qualquer componente JSF em CSS utilizando seletores de ID

```
<html xmlns="http://www.w3.org/1999/xhtml"
      xmlns:h="http://xmlns.jcp.org/jsf/html">
  <h:head>
    <title>Facelet Title</title>
    <style type="text/css">
      #texto {color: red;}
    </style>
  </h:head>
  <h:body>
    <h:outputText id="texto" value="JSF com CSS"/>
  </h:body>
</html>
```

# Seletores de Classe

---

- Pode-se definir classes para componentes JSF utilizando o atributo *styleClass*

```
<html xmlns="http://www.w3.org/1999/xhtml"
      xmlns:h="http://xmlns.jcp.org/jsf/html">
  <h:head>
    <title>Facelet Title</title>
    <style type="text/css">
      .texto {color: red;}
    </style>
  </h:head>
  <h:body>
    <h:outputText styleClass="texto" value="JSF com CSS"/>
  </h:body>
</html>
```

# CSS Inline

---

- É possível utilizar o atributo *style* em todos os componentes JSF

```
<html xmlns="http://www.w3.org/1999/xhtml"
      xmlns:h="http://xmlns.jcp.org/jsf/html">
  <h:head>
    <title>Facelet Title</title>
  </h:head>
  <h:body>
    <h:outputText style="color:red;"
                  value="JSF com CSS"/>
  </h:body>
</html>
```

# CSS Externo

---

- É possível incluir arquivos CSS externos da maneira tradicional, mas isso pode nos causar diversos problemas
  - Ao mover os arquivos XHTML, teremos que alterar manualmente o caminho especificado para os arquivos CSS
  - Futuramente utilizaremos *Facelets* para estruturar nossas páginas, o que dificultará e muito a vinculação tradicional de recursos CSS

# Recursos

---

- A partir do JSF 2.0 existe uma solução melhor para arquivos externos CSS, Javascript e imagens
  - É possível colocá-los em uma pasta chamada *resources* dentro do diretório raiz da aplicação
  - Os subdiretórios dentro de *resources* indicam bibliotecas de recursos
    - Subpasta *css* para arquivos CSS
    - Subpasta *javascript* para arquivos Javascript
    - Subpasta *images* para arquivos de imagens

# Definindo CSS Externo

---

- Para relacionar um documento XHTML com um arquivo CSS externo utilizando *resources*, devemos efetuar os seguintes passos:
  - Criar uma pasta *resources* dentro da aba *Páginas Web* do projeto
  - Dentro dessa pasta criar uma subpasta chamada *css*
  - Criar o arquivo CSS dentro dessa subpasta
  - Utilizar o componente *outputStylesheet* para linkar com o arquivo CSS



# *outputStyleSheet*

---

Nome da subpasta de  
*resources* considerada aqui



```
<h:outputStyleSheet library="css" name="style.css"/>
```



Nome do arquivo CSS a  
ser carregado

# Exemplo 1

---

- Mostra o uso de estilos CSS para formatar uma página JSF

# Script e Imagens

---

- Os componentes *outputScript* e *graphicImage* podem ser usados de forma similar para inserir, respectivamente, um arquivo Javascript ou uma imagem

```
<h:outputScript library="javascript" name="arq.js" />
```

```
<h:graphicImage library="imaged" name="foto.jpg"/>
```

# Tag *dataTable*

---

- Gera uma tabela preenchida com dados de uma lista de objetos
- Atributos Básicos
  - *id, binding, rendered, styleClass*
- Atributos HTML de tabelas
  - *bgcolor, border, cellpadding, cellspacing, dir, frame, rules, summary, lang, style, title, width*
- Eventos DHTML
  - *onclick, ondblclick, onkeydown, onkeypress, onkeyup, onmousedown, onmousemove, onmouseout, onmouseover, onmouseup*

# Dados de uma Tabela

---

- A origem dos dados deve ser definida no atributo *value* de um componente *dataTable* que aceita:
  - Um objeto Java qualquer
  - Um array
  - Um objeto `java.util.List`
  - Um objeto `java.sql.ResultSet`
  - Um objeto de `javax.servlet.jsp.jstl.sql.Result`
  - Um objeto de `javax.faces.model.DataModel`

# Dados de uma Tabela

---

- O atributo *var* permite especificar um nome para cada item a ser exibido (como no componente *repeat* do Facelets)
- Outros atributos
  - *first*
    - Posição do primeiro item a ser mostrado (começando em zero)
  - *rows*
    - Número de linhas exibidas na tabela
    - Se for zero, serão mostradas todas as linhas da tabela

# Colunas

---

- Somente componentes *column* podem ser especificados dentro de um *dataTable*
  - Qualquer outro componente será ignorado
- Um *dataTable* pode possuir diversos componentes *column*, onde cada um especifica uma coluna da tabela
- Atributos básicos de componentes *column*
  - *id*, *binding*, *rendered*, *styleClass*

# Exemplo 2

---

- Mostra um projeto com uma página listando diversos nomes de pessoas em um *dataTable*
  - Veja a construção das colunas
    - Podemos especificar nelas qualquer componente HTML ou JSF que será exibido naquela coluna em todas as linhas



# Cabeçalhos de Colunas

Dentro deste componente *facet* com nome *header* especificamos o que deve aparecer no cabeçalho da coluna

```
<h:column>
```

```
  <f:facet name="header">
```

```
    Primeiro Nome"
```

```
  </f:facet>
```

```
  #{n.primeiro}
```

```
</h:column>
```

O que estiver fora de *facet* irá aparecer no conteúdo da coluna nas diversas linhas. Neste exemplo estamos mandando mostrar o título de cada filme

# Rodapés de Colunas

---

Conteúdo da coluna

```
<h:column>
```

```
#{nome.ultimo}
```

```
<f:facet name="footer">
```

```
  Rodapé da coluna
```

```
</f:facet>
```

```
</h:column>
```

Dentro de um *facet* com nome *footer*  
especificamos o conteúdo do rodapé da coluna

# Título da Tabela

---

Dentro deste componente *facet* com nome *caption* vai o título da tabela



```
<h:dataTable ...>
```

```
  <f:facet name="caption">
```

```
    Lista de Nomes
```

```
  </f:facet>
```

```
  ...
```

```
</h:dataTable>
```

## Exemplo 3

---

- Modificação do Exemplo 2 acrescentando um título à tabela, um cabeçalho e um rodapé para as colunas

# Componentes em Facets

---

- É possível colocar vários componentes em um *facet* desde que sejam agrupados em um *panelGroup*
  - Caso contrário, somente o primeiro componente será mostrado

# Tabelas e Estilos

---

- O componente *dataTable* possui atributos que especificam classes CSS para
  - A tabela como um todo (*styleClass*)
  - Cabeçalhos e rodapés de colunas (*headerClass* e *footerClass*)
  - Colunas individuais (*columnClasses*)
  - Linhas individuais (*rowClasses*)
- **OBSERVAÇÃO**
  - Os atributos *columnClasses* e *rowClasses* são mutuamente exclusivos (*columnClasses* tem prioridade)

# Exemplo 4

---

- Nesse exemplo utilizamos os atributos *styleClass*, *headerClass* e *columnClasses*
  - Note que é possível especificar mais de uma classe em *columnClasses*
  - As classes são aplicadas alternadamente nas colunas
  - Como somente duas classes foram especificadas, apenas as duas primeiras colunas serão formatadas

# Exemplo 5

---

- Reimplementação do exemplo anterior utilizando *rowClasses* em vez de *columnClasses*
  - Note que, como o número de classes é menor que o número de linhas, o JSF repete as classes alternadamente



# Exemplo 6

---

- Vamos modificar o formulário de consulta de usuários, onde todos os usuários serão listados em uma tabela
  - Em cada linha foram colocados links para consulta e exclusão de usuários

# Primefaces

---

- Biblioteca de componentes adicionais para o JSF
- Contém componentes mais sofisticados
  - Muitos implementam o conceito de RIA (*Rich Internet Applications*)
    - Componentes visuais mais sofisticados

# Primefaces

---

- Documentação
  - Link disponível no Moodle
- Mostruário de componentes
  - Link disponível no Moodle

# Primefaces e Netbeans

---

- Basta incluir a biblioteca do Primefaces no projeto

# Exemplo 7

---

- Modificação do exemplo do cadastro de usuários utilizando componentes Primefaces
  - InputText e InputMask
  - CommandButton e Button
  - DataTable e Column
  - PanelGrid
  - OutputLabel
  - Layout, LayoutUnit, Menu e MenuItem
  - Dialog