

PROGRAMAÇÃO PARA WEB II

AULA 6

Profa. Silvia Bertagnolli

NAVEGANDO
COM JSF


TAREFAS

1. Abra o arquivo ListaExercicios_Aula18 que está no Moodle

CRIANDO O LOGIN

O Login é uma página JSF que permite a entrada no sistema

Esta página chama o método `usuarioBean.validar()`, que valida se o usuário é válido ou inválido



Identificador:

Senha:

VALIDANDO O USUÁRIO

```
public String validar() {  
    if(usuario.validar()){  
        this.usuarioLogado = usuario;  
        usuario = new Usuario();  
        return "menu";  
    }  
    return "invalido";  
}
```

Para que é usado o atributo **usuarioLogado**?

MENU

Olá sbertagnolli, seja bem-vindo ao nosso sistema!

[Cadastrar usuário](#)

[Listar usuários](#)

[Sair](#)

O MENU

```
<h:panelGrid columns="1">
    <h:outputLabel style="font-size: 20px;"
        value="Olá {usuarioBean.usuarioLogado.identificador},
            seja bem-vindo ao nosso sistema!"/>
</h:panelGrid>
<h:panelGrid columns="1">
    <h:link value="Cadastrar usuário" outcome="cadastrar"/>
</h:panelGrid>
```


LINKS

`<h:link />`

`<h:commandLink />`

`<h:outputLink />`

São usados para renderizar um href do HTML

EXEMPLOS: H:LINK

```
<h:link value="Login page" outcome="login" />
```

É equivalente a:

```
<a href="/JavaServerFaces/faces/login.xhtml">Login page</a>
```

EXEMPLOS: H:LINK

```
<h:link value="Página de Login + Param " outcome="login" >  
<f:param name="identificador" value="sbertagnolli" />  
</h:link>
```

É equivalente a:

```
<a href="/JavaServerFaces/faces/login.xhtml?  
identificador=sbertagnolli"> Página de Login + Param</a>
```

EXEMPLOS: H:LINK

```
<h:link outcome="login" >
```

```
<h:graphicImage library="images" name="cadeado.png" />
```

```
</h:link>
```

É equivalente a:

```
<a href="/JavaServerFaces/faces/login.xhtml">  </a>
```

CADASTRO

TELA DE CADASTRO

Cadastrar Usuário

Nome:

Identificador:

Senha:

@POSTCONSTRUCT

Ao executar o código sem as linhas abaixo ocorre um `NullPointerException`. Explique por que isso ocorre.

```
@PostConstruct  
  
public void carregaLista() {  
    lista = usuario.listar();  
}
```

Para que serve o `PostConstruct`?

LISTAGEM

Nome	Identificador		
Angelo da Silva	asilva	Excluir	Editar
Mariana da Silva	msilva	Excluir	Editar
Agnes Silveira2	asilveira	Excluir	Editar
Pedro Gomes	pgomes	Excluir	Editar
Marcela Ramos	mramos	Excluir	Editar

Menu

LISTANDO DADOS

```
<h:dataTable border="1" value="#{usuarioBean.lista}"
var="user">
  <h:column>
    <f:facet name="header">Nome</f:facet>#{user.nome}
  </h:column>
  <h:column>
    <f:facet name="header">Identificador</f:facet>
    #{user.identificador}
  </h:column>
  <h:column>
```

LISTANDO DADOS

```
<h:column>
    <h:commandButton value="Excluir"
                      action="#{usuarioBean.excluir(user)}/>
</h:column>
<h:column>
    <h:commandButton value="Editar"
                      action="#{usuarioBean.editar(user)}/>
</h:column>
</h:dataTable>
```

COMMANDBUTTON

Usado como um botão, onde:

value = corresponde ao texto que será exibido

action = ação que será realizada

onClick = evento acionado quando o botão é clicado

```
<h:column>  
    <h:commandButton value="Excluir"  
                      action="#{usuarioBean.excluir(user)}/>  
</h:column>
```

```
<h:commandButton value = "Clique aqui" onclick =  
"alert('Olá');" />
```

TRABALHO FINAL

TRABALHO

Usando as páginas desenvolvidas no trabalho anterior faça:

- 1) A tela de login chamar um menu que conterà os itens da aula de hoje, e o identificador do usuário que está logado
- 2) Monte a navegação entre as páginas de modo que os dados carregados estejam corretos
- 3) A tabela de listagem deve conter um botão para excluir e um botão para editar – use imagens para fazer esses botões