

Lab 3 - Primefaces

Neste laboratório iremos conhecer alguns componentes do primefaces e construiremos toda interface final do projeto Banco que está sendo desenvolvida ao longo do curso.

Para informações adicionais, tutoriais e documentação acesse o site <http://www.primefaces.org/>.

Exercícios

Exercício 1: Adicionando Primefaces ao projeto.

Exercício 2: Modificando url-pattern e definindo um tema primefaces.

Exercício 3: Definindo templates e criando a tela de login.

Exercício 4: Criar todas as páginas referente ao projeto Banco.

Exercício 1 -Adicionando Primefaces ao projeto

1. Modifique o **pom.xml** do projeto *Banco-Framework* adicionando o repositório do primefaces.

```
<repositories>
    <repository>
        <id>prime-repo</id>
        <name>PrimeFaces Maven Repository</name>
        <url>http://repository.primefaces.org</url>
        <layout>default</layout>
    </repository>
</repositories>
```

2. Agora ainda no **pom.xml** adicione o primefaces em suas dependências.

```
<dependency>
    <groupId>org.primefaces</groupId>
    <artifactId>primefaces</artifactId>
    <version>4.0</version>
</dependency>
```

3. Todas as páginas que serão criadas e usarão o primefaces terão que declarar as seguintes tags:

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

Exercício 2 -Modificando url-pattern e definindo um tema do primefaces

1. Para adicionar um tema modifique o **pom.xml** de *Banco-Framework* adicionando a seguinte dependência:

```
<dependency>
    <groupId>org.primefaces.themes</groupId>
    <artifactId>all-themes</artifactId>
    <version>1.0.10</version>
</dependency>
```

Na tag <artifactId/> estamos informando que qualquer tema do primefaces poderá ser utilizado.

2. Agora modifique o **web.xml** de *Banco-Framework* para declarar qual tema iremos utilizar.

```
<context-param>
  <param-name>primefaces.THEME</param-name>
  <param-value>bootstrap</param-value>
</context-param>
```

Você deve colocar o tema selecionado dentro da tag `<param-value/>`.

Nesse caso o tema selecionado foi o *bootstrap*, mas você pode colocar qualquer tema do primefaces, inclusive criar um tema no site: <http://jqueryui.com/themeroller/>

3. Para modificar a url-pattern/url de acesso, edite o arquivo **web.xml** conforme o código abaixo:

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

<welcome-file-list>
  <welcome-file>/index.jsf</welcome-file>
</welcome-file-list>
```

<servlet> define os servlets da aplicação e seus parâmetros.

<servlet-mapping> estamos informando ao web container quais tipos de servlets serão invocados e que tipo de URL deverá ser chamado, nesse caso serão URLs com extensão **.jsf**.

<welcome-file-list> estamos informando qual arquivo ou arquivos devem ser invocados automaticamente pelo servidor caso nenhum outro arquivo tenha sido especificado.

Anteriormente a url de acesso era **http://localhost:8080/SuaAplicacao/faces/suaPagina.xhtml**
agora será **http://localhost:8080/SuaAplicacao/suaPagina.jsf**

4. Para testar, crie uma página index.xhtml com seguinte código:

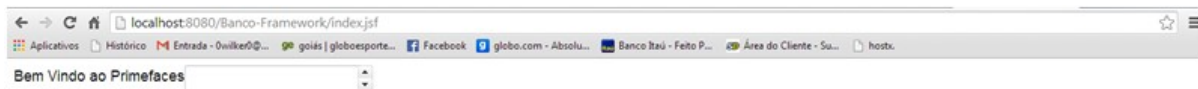
```
<html xmlns="http://www.w3.org/1999/xhtml"
  xmlns:h="http://java.sun.com/jsf/html"
  xmlns:f="http://java.sun.com/jsf/core"
  xmlns:p="http://primefaces.org/ui">

  <h:head>

  </h:head>
  <h:body>
    <p:outputLabel value="Bem Vindo ao Primefaces"/>
    <p:spinner />
  </h:body>
</html>
```

Da forma que o primefaces foi declarado em `xmlns:p="http://primefaces.org/ui"`
Todos os componentes do primefaces são acessados com `<p:ComponenteQueForUtilizar />`

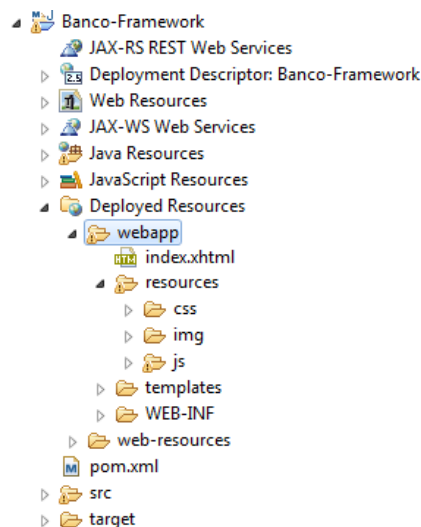
5. Agora execute no servidor tomcat acessando a seguinte url: <http://localhost:8080/Banco-Framework/index.jsf>



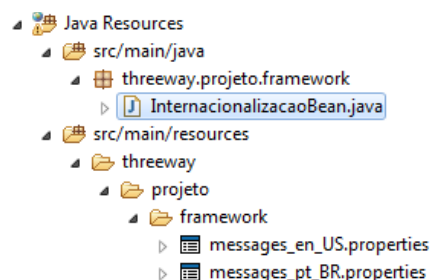
6. Brinque um pouco mude o tema, procure em <http://www.primefaces.org/themes>, veja o quanto é simples mudar o tema de um sistema desenvolvido em Primefaces.

Exercício 3-Definindo templates e criando a tela de login

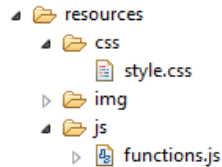
1. Crie os diretórios *templates*, *resources*, *resources/css*, *resources/js* e *resources/img* dentro de *webapp*.



2. Internacionalize sua aplicação como já foi feito anteriormente, criando os arquivos *messages* e a classe **InternacionalizacaoBean.java**, depois faça o mapeamento no **faces-config.xml**. Utilize os arquivos *properties* e *internacionalizacaoBean* criados no projeto WEB. Sempre que for preciso, crie as propriedades no *properties*, sabemos que você já sabe como fazer por isso não descreveremos como.



3. Crie os arquivos **style.css**, **functions.js** dentro de seus respectivos pacotes.



O arquivo **style.css** irá conter os estilos dos elementos das nossas páginas, enquanto **function.js** poderá conter algum código javascript caso queiramos adicionar à aplicação.

4. Modifique o arquivo **style.css** que acabamos de criar para adequações a página. Caso haja dúvida sobre algum aspecto desse arquivo pergunte ao professor.

```
*,html,body,div,dl,dt,dd,ul,ol,li,h1,h2,h3,h4,h5,h6,pre,form,label,fieldset,input,p,blockquote,th,td {
    margin: 0;
}
html {
    font-family: Arial, Helvetica, sans-serif;
    font-size: 12px; /*color:#565656*/
    color: #000;
    margin-left: 0px;
}
body {
    line-height: 1.5;
    border: 0;
    font-size: 12px;
    margin: 0 0 0 0;
    padding: 0;
}
.vertical-offset-100 {
    padding-top: 100px;
}
.painelLogin {
    width: 310px;
    height: 170px;
    margin-left: auto;
    margin-right: auto;
    margin-top: 18%;
}

.internacionalizacao {
    padding: 6px 4px 0px;
    border-radius: 3px;
    border: 1px solid;
    text-align: right !important;
}
.textRight {
    text-align: right !important;
}

.ui-panelgrid, .ui-panelgrid td, .ui-panelgrid tr {
    border-style: none !important
}
.ui-widget, .ui-datatable, .ui-datatable-empty-message, span, label {
    font-size: 12px !important;
}

.deposito {
    background: green !important ;
}

.saque {
    background: red !important ;
}

.transferencia {
    background: blue !important ;
}
```

5. Crie a página **login.xhtml** utilizando primefaces e JSF de acordo com o código abaixo:

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

<h:head>
<title>#{msg['titulo.cabecalho']} </title>
<link rel="stylesheet" type="text/css" href="#{facesContext.externalContext.requestContextPath}/resources/css/style.css" />
<script src="#{facesContext.externalContext.requestContextPath}/resources/js/functions.js"> </script>
</h:head>

<h:body>
<ui:include src="templates/topoPagina.xhtml" />

<h:form id="formLogin">

    <p:panel styleClass="painelLogin" header="#{msg['realizarLogin']}">

        <h:panelGrid styleClass="textRight" columns="2" cellpadding="5">

            <h:outputLabel value="#{msg['login']}: " />
            <p:inputText />

            <h:outputLabel value="#{msg['senha']}: " />
            <p:password />

            <p:spacer width="70" height="0"/>
            <p:commandButton value="#{msg['entrar']}" />

        </h:panelGrid>

    </p:panel>

</h:form>
</h:body>
</html>
```

Perceba que estamos adicionando os arquivos **style.css** e **function.js** no começo da página, no cabeçalho, dentro das tags **<link>** e **<script>**.

6. Crie a página **topoPagina.xhtml** de modo que o mesmo implemente a internacionalização vista anteriormente. Crie dentro do pacote templates de acordo com o código abaixo:

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

<p:panel>
<h:panelGrid columns="2" width="100%">
    <h3>#{msg['titulo.cabecalho']} </h3>

    <h:form id="formInternacionalizacao" style="text-align: right !important">

        <h:outputLabel value="#{msg['saudacao']}: " />

        <p:spacer width="15" />

        <p:commandLink styleClass="internacionalizacao" update="@all"
            action="#{internacionalizacaoBean.portugueseLocale()}">
            <p:graphicImage value="resources/img/brasil.png" />
        </p:commandLink>

        <p:spacer width="5" />

        <p:commandLink styleClass="internacionalizacao" update="@all"
            action="#{internacionalizacaoBean.englishLocale()}">
            <p:graphicImage value="resources/img/estados_unidos.png" />
        </p:commandLink>

        <p:spacer width="15" />

        <p:commandLink action="login" value="#{msg['botao.sair']}" />

    </h:form>
</h:panelGrid>
</p:panel>
</html>
```

Criamos dois botões que irão aplicar a mudança de linguagem na nossa página de acordo com os métodos em InternacionalizacaoBean e o arquivo messages.

Perceba que estamos incluindo o **topoPagina.xhtml** em **login.xhtml** (`<ui:include />`).

7. Execute a tela de login e veja o resultado:

Projeto Banco 3Way NetWorks

Bem vindo, Sr (a)   Sair



8. Crie a página **template.xhtml** utilizando primefaces e JSF de acordo com o código abaixo. Utilize o componente `<p:layout />`.

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

<h:head>
<title>#{msg['titulo.cabecalho']}

```

```
</p:layoutUnit>



<p:layoutUnit style="border:none !important" size="15%" position="east"/>
<p:layoutUnit style="border:none !important" size="15%" position="west"/>

</p:layout>
</h:body>
</html>
```


Veja que a página possui a tag `<ui:include name="conteudo" />`, onde poderá ser inserido uma página que for definida como composition. Seu funcionamento foi explicado no laboratório anterior.


9. Execute o template pela url <http://localhost:8080/Banco-Framework/templates/template.jsf>.

Projeto Banco 3Way NetWorks

Bem vindo, Sr.(a)   [Sair](#)

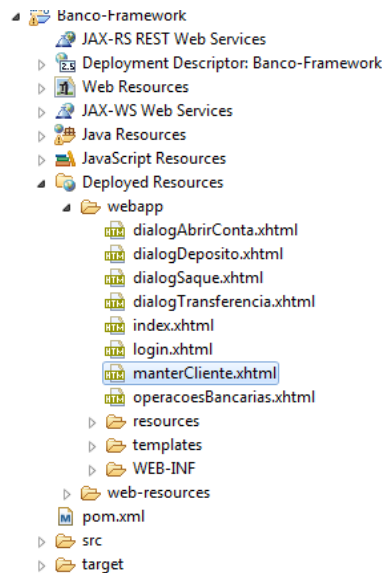
Banco: Agência:

 Manter Cliente

 Operações Bancárias

Exercício 4 - Criar todas as páginas referente a aplicação Banco

1. Crie as seguintes páginas: **manterCliente.xhtml**, **operacoesBancarias.xhtml**, **dialogAbrirConta.xhtml**, **dialogSaque.xhtml**, **dialogDeposito.xhtml** e **dialogTransferencia.xhtml**, essas serão modificadas nos passos posteriores para concluir o exercício.



2. Modifique a página **manterCliente.xhtml** utilizando o template já criado de acordo com o código abaixo:

```
<ui:composition xmlns="http://www.w3.org/1999/xhtml"
xmlns:f="http://java.sun.com/jsf/core"
xmlns:h="http://java.sun.com/jsf/html"
xmlns:p="http://primefaces.org/ui"
xmlns:ui="http://java.sun.com/jsf/facelets"
template="templates/template.xhtml">

<ui:define name="conteudo">

<h:form id="manterClientesForm">

    <p:panel header="">
        <p:panelGrid style="width:100%;">

            <p:row>
                <p:column styleClass="textRight">
                    <h:outputLabel value="#{msg['form.cliente.nome']}:" />
                </p:column>

                <p:column>
                    <p:inputText style="width:100%;" size="45" />
                </p:column>

                <p:column styleClass="textRight">
                    <h:outputLabel value="#{msg['form.cliente.telefone']}:" />
                </p:column>

                <p:column>
                    <p:inputMask style="width:100%;" mask="(99)9999-9999" />
                </p:column>
            </p:row>

            <p:row>
                <p:column styleClass="textRight">
                    <h:outputLabel value="#{msg['form.cliente.endereco']}:" />
                </p:column>

                <p:column colspan="3">
                    <p:inputText style="width:100%;" />
                </p:column>
            </p:row>
        </p:panelGrid>
    </p:panel>
</h:form>
</ui:define>
</ui:composition>
```



```

</p:row>
<p:row>
  <p:column styleClass="textRight">
    <h:outputLabel value="#{msg['form.cliente.rg']}:" />
  </p:column>
  <p:column>
    <p:inputText style="width:100%;" />
  </p:column>
  <p:column styleClass="textRight">
    <h:outputLabel value="#{msg['form.cliente.cpf']}:" />
  </p:column>
  <p:column>
    <p:inputMask style="width:100%;" mask="999.999.999-99" />
  </p:column>
</p:row>
</p:panelGrid>
</p:panel>
<p:panel>
  <p:dataTable emptyMessage="#{msg['tabela.cliente.empty']}" styleClass="fonte">
    <f:facet name="header">
      #{msg['topoTabelaCliente']}
    </f:facet>
    <p:column headerText="#">
    </p:column>
    <p:column headerText="#{msg['form.cliente.nome']}">
    </p:column>
    <p:column headerText="#{msg['form.cliente.endereco']}">
    </p:column>
    <p:column headerText="#{msg['form.cliente.telefone']}">
    </p:column>
    <p:column headerText="#{msg['form.cliente.rg']}">
    </p:column>
    <p:column headerText="#{msg['form.cliente.cpf']}">
    </p:column>
    <p:column headerText="#{msg['tabela.acoes']}">
    </p:column>
  </p:dataTable>
</p:panel>
</h:form>
</ui:define>
</ui:composition>

```

A página `manterCliente.xhtml` é um `composition` que contém um formulário com os campos dos dados do cliente e uma tabela (tag `<p:dataTable />`) com a lista de clientes já cadastrados. Perceba que no começo da página declaramos ela como “conteudo”. O mesmo será feito com a página `operacoesBancarias.xhtml`.

3. Execute a página **manterCliente.xhtml** e veja o resultado

Projeto Banco 3Way NetWorks

Bem vindo, Sr.(a)

Sair

Banco: Agência:

Manter Cliente

Operações Bancárias

Nome:

Telefone:

Endereço:

Registro Geral:

CPF:

Clientes Cadastrados

#	Nome	Endereço	Telefone	Registro Geral	CPF	Ações
Nenhum Cliente Cadastrado						

4. Agora modifique a página **operacoesBancarias.xhtml** que acabamos de criar de acordo com o código abaixo:

```
<ui:composition xmlns="http://www.w3.org/1999/xhtml"
xmlns:f="http://java.sun.com/jsf/core"
xmlns:h="http://java.sun.com/jsf/html"
xmlns:p="http://primefaces.org/ui"
xmlns:ui="http://java.sun.com/jsf/facelets"
template="templates/template.xhtml">

<ui:define name="conteudo">

<h:form id="opBancariasForm">
    <p:panel header="">

        <h:panelGrid columns="8" cellpadding="5">
            <h:outputLabel value="Nº Conta:" />
            <h:outputLabel value="#" />
            <p:spacer width="10" />
            <h:outputLabel value="Data Abertura:" />
            <h:outputLabel value="#" />
            <p:spacer width="10" />
            <h:outputLabel value="Saldo:" />
            <h:outputLabel value="#" />
        </h:panelGrid>

        <p:spacer height="20" />
        <p:separator />
        <p:spacer height="20" />

        <h:panelGrid columns="4" cellpadding="2">
            <p:commandButton value="Abrir Conta" onclick="PF('dialogAbrirConta').show();" />
            <p:commandButton value="Deposito" onclick="PF('dialogDeposito').show();" />
            <p:commandButton value="Saque" onclick="PF('dialogSaque').show();" />
            <p:commandButton value="Transferencia" onclick="PF('dialogTransf').show();" />
        </h:panelGrid>

    </p:panel>

    <p:panel>

        <p:dataTable emptyMessage="#{msg['tabela.transacao.empty']}" styleClass="fonte">
            <f:facet name="header">
                #{msg['topoTabelaTransacao']}
            </f:facet>
            <p:column headerText="#">
            </p:column>
            <p:column headerText="#{msg['form.transacao.tipoTransacao']}">
            </p:column>
            <p:column headerText="#{msg['form.transacao.contaCredito']}">
            </p:column>
            <p:column headerText="#{msg['form.transacao.contaDebito']}">
            </p:column>
            <p:column headerText="#{msg['data']}">
            </p:column>
            <p:column headerText="#{msg['valor']}">
            </p:column>
            <p:column headerText="#{msg['tabela.acoes']}">
            </p:column>
        </p:dataTable>
    </p:panel>
</h:form>

<h:form id="dialogForm">

    <ui:include src="dialogDeposito.xhtml" />
    <ui:include src="dialogSaque.xhtml" />
    <ui:include src="dialogTransferencia.xhtml" />
    <ui:include src="dialogAbrirConta.xhtml" />



</h:form>

</ui:define>
</ui:composition>
```

Assim como em `manterCliente.xhtml`, a página `operacoesBancarias.xhtml` possui um formulário que contém as opções de operações bancárias e uma tabela contendo as operações feitas pelo cliente selecionado. Perceba que estamos incluindo 4 dialogs nessa página. Ao incluir essas dialogs na página, podemos chama-las pelos *Command Buttons* referente a cada operação bancária. Basta utilizar o método `onClick` e **`PF(dialog).show()`** .

5. Execute a página e veja o resultado:

Projeto Banco 3Way NetWorks

Bem vindo, Sr.(a)   Sair

Banco: Agência:

Manter Cliente

Operações Bancárias

Nº Conta: # Data Abertura: # Saldo: #

Abrir Conta

Deposito

Saque

Transferencia

Historico de Transações

#	Tipo Transação	Titular/Conta Crédito	Titular/Conta Débito	Data	Valor	Ações
Nenhuma Transação Cadastrada						

6. Agora vamos modificar a página **`dialogDeposito.xhtml`**, para que esta seja uma dialog chamada dentro da página **`operacoesBancarias.xhtml`**.

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

<p:dialog id="idDialogDeposito" appendTo="@{body}" header="Deposito" widgetVar="dialogDeposito"
modal="true" closable="false" showEffect="explode" hideEffect="bounce" resizable="false">

<h:panelGrid columns="2" cellpadding="5">
<h:outputLabel value="#{msg['valor']}:" />
<p:inputText />
</h:panelGrid>

<p:spacer height="5"/>
<p:separator />

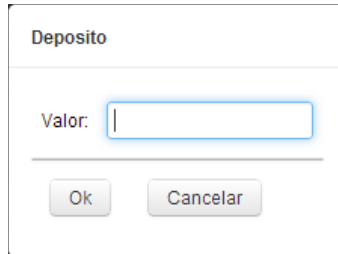
<h:panelGrid columns="2" cellpadding="10">
<p:commandButton value="Ok" />
<p:commandButton value="Cancelar" onclick="PF('dialogDeposito').hide();"/>
</h:panelGrid>

</p:dialog>
</html>
```

Ao declarar a dialog, estamos inserindo todas as propriedades: id, efeito ao ser aberta e fechada, se pode ser redimensionada, título, etc. Você pode encontrar mais detalhes sobre essas propriedades no site do primefaces.

Veja que no método `onClick` do segundo *Command Button* utilizando **`PF(dialog).hide()`** para que a dialog seja fechada.

7. Execute a página **operacoesBancarias.xhtml** e clique no botão *Deposito* para ver o resultado do código:



8. Agora vamos modificar a página **dialogSaque.xhtml**, para que esta seja uma dialog chamada dentro da página **operacoesBancarias.xhtml**.

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

<p:dialog id="idDialogSaque" appendTo="@{body}" header="Saque" widgetVar="dialogSaque"
modal="true" showEffect="explode" hideEffect="bounce" closable="false" resizable="false">

<h:panelGrid columns="2" cellpadding="5">
  <h:outputLabel value="#{msg['valor']}:" />
  <p:inputText />
</h:panelGrid>

<p:spacer height="5"/>
<p:separator />

<h:panelGrid columns="2" cellpadding="10">
  <p:commandButton value="Ok" />
  <p:commandButton value="Cancelar" onclick="PF('dialogSaque').hide();"/>
</h:panelGrid>

</p:dialog>
</html>
```

9. Execute a página **operacoesBancarias.xhtml** e clique no botão *Saque* para ver o resultado do código:



10. Agora vamos modificar a página **dialogTransferencia.xhtml**, para que esta seja uma dialog chamada dentro da página **operacoesBancarias.xhtml**.

```
<html xmlns="http://www.w3.org/1999/xhtml"
```

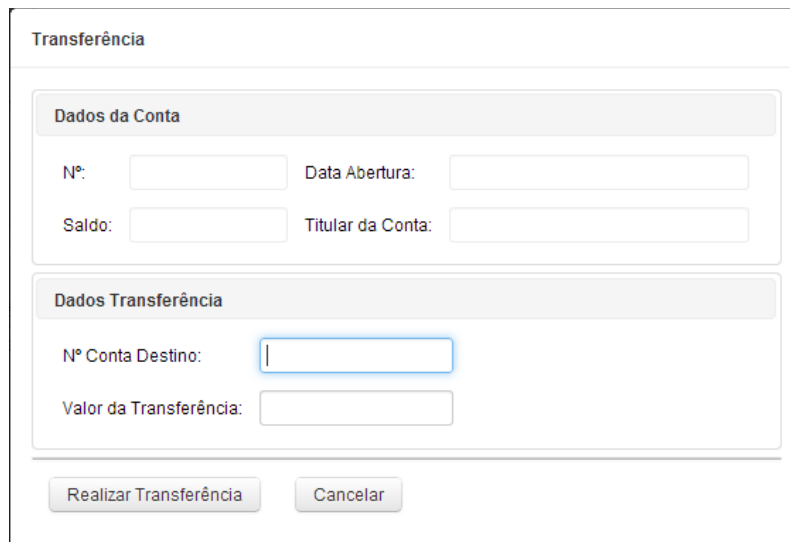
```

xmlns:h="http://java.sun.com/jsf/html"
xmlns:f="http://java.sun.com/jsf/core"
xmlns:p="http://primefaces.org/ui"
xmlns:ui="http://java.sun.com/jsf/facelets">

<p:dialog id="idDialogTransf" appendTo="@body" header="Transferência" widgetVar="dialogTransf"
    modal="true" showEffect="explode" hideEffect="bounce" closable="false" resizable="false">
<p:panel header="Dados da Conta">
    <h:panelGrid columns="4" cellpadding="5">
        <h:outputLabel value="Nº:" />
        <p:inputText size="15" disabled="true"/>
        <h:outputLabel value="Data Abertura:" />
        <p:inputText size="35" disabled="true"/>
        <h:outputLabel value="Saldo:" />
        <p:inputText size="15" disabled="true"/>
        <h:outputLabel value="Titular da Conta:" />
        <p:inputText size="35" disabled="true"/>
    </h:panelGrid>
</p:panel>
<p:spacer height="5" />
<p:panel header="Dados Transferência">
    <h:panelGrid columns="2" cellpadding="5">
        <h:outputLabel value="Nº Conta Destino:" />
        <p:inputText />
        <h:outputLabel value="Valor da Transferência:" />
        <p:inputText />
    </h:panelGrid>
</p:panel>
<p:spacer height="5" />
<p:separator />
<h:panelGrid columns="2" cellpadding="10">
    <p:commandButton value="Realizar Transferência" />
    <p:commandButton value="Cancelar" onclick="PF('dialogTransf').hide();"/>
</h:panelGrid>
</p:dialog>
</html>

```

11. Execute a página **operacoesBancarias.xhtml** e clique no botão *Transferencia* para ver o resultado do código:



12. Agora vamos modificar a página **dialogAbrirConta.xhtml**, para que esta seja uma dialog chamado dentro da página **operacoesBancarias.xhtml**.

```
<html xmlns="http://www.w3.org/1999/xhtml">
```

```
xmlns:h="http://java.sun.com/jsf/html"
xmlns:f="http://java.sun.com/jsf/core"
xmlns:p="http://primefaces.org/ui"
xmlns:ui="http://java.sun.com/jsf/facelets">

<p:dialog id="idDialogAbrirConta" appendTo="@body" header="Abertura de Conta" widgetVar="dialogAbrirConta"
        modal="true" closable="false" showEffect="explode" hideEffect="bounce" resizable="false">

    <h:panelGrid columns="2" cellpadding="5">
        <h:outputLabel value="Data Abertura:" />
        <p:inputText disabled="true"/>
        <h:outputLabel value="Numero da Conta:" />
        <p:inputText />
        <h:outputLabel value="Saldo Inicial:" />
        <p:inputText />
    </h:panelGrid>

    <p:spacer height="5"/>

    <p:separator />

    <h:panelGrid columns="2" cellpadding="10">
        <p:commandButton value="Abrir Conta" />
        <p:commandButton value="Cancelar" onclick="PF('dialogAbrirConta').hide();"/>
    </h:panelGrid>

</p:dialog>
</html>
```

13. Execute a página **operacoesBancarias.xhtml** e clique no botão *Abrir Conta* para ver o resultado do código:

