



COTI Informática

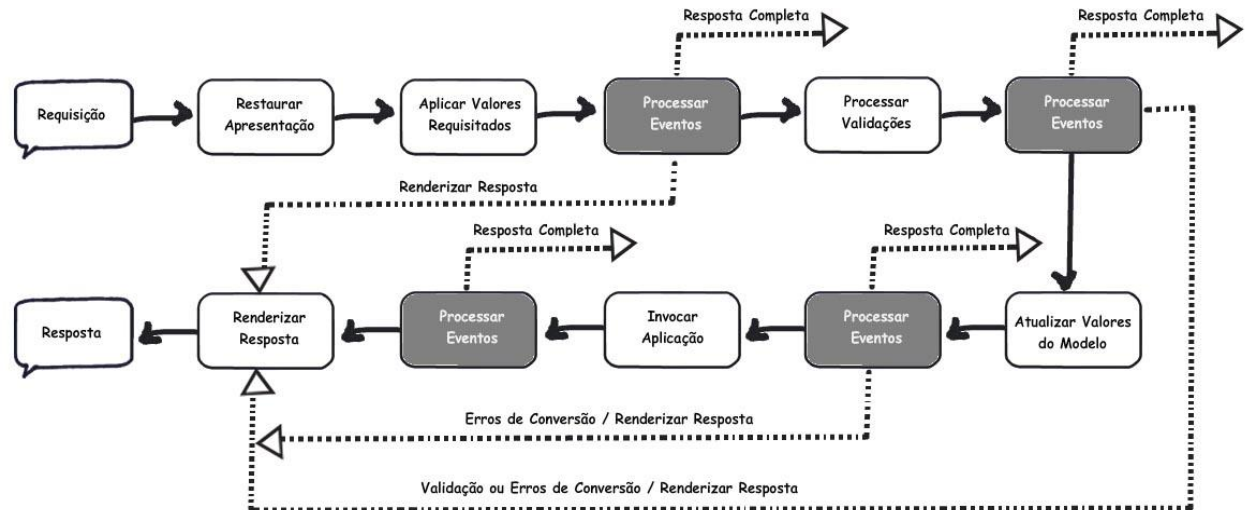
# Jsf2 e Primefaces

## Apostila de Componentes

---

COTI Informática - Av. Rio Branco, 185 - Sala 904  
Rio de Janeiro - RJ Tel. 21-2262-9043  
[www.cotiinformatica.com.br](http://www.cotiinformatica.com.br)

## Ciclo de Vida do Jsf



### Fase 01

### Restaurar Apresentação

Se a página requisitada já foi exibida antes, é recuperada toda a árvore de componentes para a página requisitada. Caso esteja sendo exibida pela primeira vez é construída uma nova árvore de componentes. No caso da página já ter sido exibida antes, todos os componentes serão configurados com seu estado anterior. É fácil notar isso no processo de validação, onde os valores digitados inicialmente no formulário serão mantidos.

Se a requisição não possuir dados solicitados, a implementação JSF pula para a fase Render Response. Normalmente isso acontece quando a página é requisitada a primeira vez.

Nesta fase o request vem através do controller do FacesContext. O Controller examina o request e extrai o ID da visão (tela, view) que é determinado pelo nome da página JSP.

O Framework do JSF utiliza esse ID para localizar os componentes para a tela atual. Se a visão ainda não existe ela será criada pelo framework, se ela já existir será apenas reutilizada pelo controller. A visão possui os componentes GUI's (Graphical User Interface).

Esta fase possui 3 instâncias: nova visão, visão inicial e o retorno (postback) da visão. Cada instância dessa é tratada de forma diferente.

Na nova visão, o JSF constroi a visão para o Faces e vincula esta página aos handlers e validators. A visão é salva no FacesContext como um objeto. O FacesContext contém todas as informações de como se deverá manipular este GUI para a requisição corrente. O FacesContext guarda as visões na propriedade ViewRoot o qual possui todos os componentes da visão atual.

Na visão inicial, o JSF cria uma visão em branco. Esta visão será populada de acordo com o evento solicitado pelo usuário. Da visão inicial o JSF avança diretamente para a fase de Render Response.

No caso do postback (o usuário retornou a página que acessou anteriormente), a visão já existe sendo necessário apenas ser restaurada.

### Fase 02

#### Aplicar valores requisitados

Nesta fase ocorrem iterações sobre os objetos da árvore de componentes. São extraídos todos os valores digitados pelo usuário e guardados esses valores nos seus respectivos componentes.

O propósito desta fase é fazer com que cada componente recupere seu estado corrente. O Componente deve primeiro ser criado ou recuperado a partir do FacesContext, seguido por seus valores. Os valores dos componentes são geralmente recuperados dos parâmetros de request, no entanto eles podem ser recuperados dos cabeçalhos ou cookies gerados.

Se um componente não está com seu atributo Immediate setado para True, o valor será apenas convertido. Se o campo for Inteiro, o valor será convertido para um inteiro. Se a conversão falhar será adicionado no FacesContext uma mensagem de erro que será exibida durante a fase de Render response.

Se um componente estiver com seu atributo immediate setado para true o valor será convertido para o tipo específico do atributo e logo após será validado, o valor convertido será guardado no componente.

É nesta fase onde ocorrem as conversões de valores, ou seja, onde os Converter's criados pelo usuário serão utilizados ou os conversores padrão do JSF.

### Fase 03

#### Processar Validações

A String com o valor enviado é convertida para o tipo correto do objeto. Caso ocorra algum erro de validação mensagem de erro é adicionada no FacesContext, o componente é marcado como inválido e a implementação JSF invoca a fase Render Response. Caso contrário o ciclo de vida continua normalmente.

É nesta fase onde ocorrem as validações. Esta validação pode ser criada pelo desenvolvedor ou "obtida" diretamente do JSF. Os valores são validados de acordo com as regras de validação da aplicação. Se um valor estiver errado é gerado uma mensagem de erro e adicionada no FacesContext e o componente é marcado como inválido. Se algum componente estiver marcado como inválido o JSF avançará automaticamente para a fase de Render Response que irá exibir o erro de validação.

Se não houver nenhum erro de validação o JSF irá avançar para a fase de Update Model values.

**Fase 04****Atualizar Valores do Modelo**

Sua principal atividade é atribuir os valores informados pelo usuário no formulário, para as respectivas propriedades associadas aos BackBeans (também conhecidos como Managed Beans). Pode haver erro na conversão, fazendo com que o JSF dispare um erro de tempo de execução.

Esta é a fase onde é atualizado o valor no lado do servidor, atualizando a propriedade dentro do seu Backing Bean. Somente as propriedades que estão sendo manipulados do componente é que serão atualizadas. Note que esta fase só irá acontecer após a fase de validação, o que lhe garante que os dados que estão indo para o seu Bean são válidos a nível de "tela" pois os mesmos poderão ser inválidos a nível de regras de negócio de sua aplicação.

**Fase 05****Invocar Aplicação**

Nesta fase o controlador (controller) do JSF chama o método associado no submit, disparando assim a camada de regras de negócio da aplicação. Todos os valores foram validados e carregados nas fases anteriores, por isso poderemos usá-los conforme necessitar. Geralmente é retornado uma String de resultado, que é passada para o handle de navegação.

Nesta fase o Controller do JSF invoca a aplicação para manipular o envio do form. O valor do componente sempre estará convertido, validado e aplicado aos objetos de modelo, então vc poderá utilizá-los para realizar suas regras de negócio.

Nesta fase vc especifica um "outcome" ou valor de saída que irá servir para continuar o fluxo lógico da aplicação. Você faz isso definindo uma "saída" quando houver sucesso no envio de um form e irá dar um retorno desta saída.

Por exemplo, uma saída "sucesso" irá levar o usuário para a próxima página. Mas para essa navegação poder acontecer é necessário que seja criado um mapeamento dentro do arquivo faces-config.xml com este "outcome".

**Fase 06****Renderizar Resposta**

Esta fase codifica a resposta e envia-a de volta ao navegador. A casa nova requisição o ciclo recomeça do zero.

### Parte 1: Componentes de seleção e entrada de dados

#### Configuração do Jsf 2.0 com Primefaces:

##### web.xml

```
<?xml version="1.0" encoding="UTF-8"?>

<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns="http://java.sun.com/xml/ns/javaee"
  xmlns:web="http://java.sun.com/xml/ns/javaee/web-app_2_5.xsd"
  xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
    http://java.sun.com/xml/ns/javaee/web-app_3_0.xsd"
  version="3.0">

  <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>

  <context-param>
    <param-name>javax.faces.STATE_SAVING_METHOD</param-name>
    <param-value>client</param-value>
  </context-param>
  <context-param>
    <param-name>javax.servlet.jsp.jstl.fmt.localizationContext</param-name>
    <param-value>resources.application</param-value>
  </context-param>
  <listener>
    <listener-class>com.sun.faces.config.ConfigureListener</listener-class>
  </listener>

  <welcome-file-list>
    <welcome-file>index.jsp</welcome-file>
  </welcome-file-list>

</web-app>
```

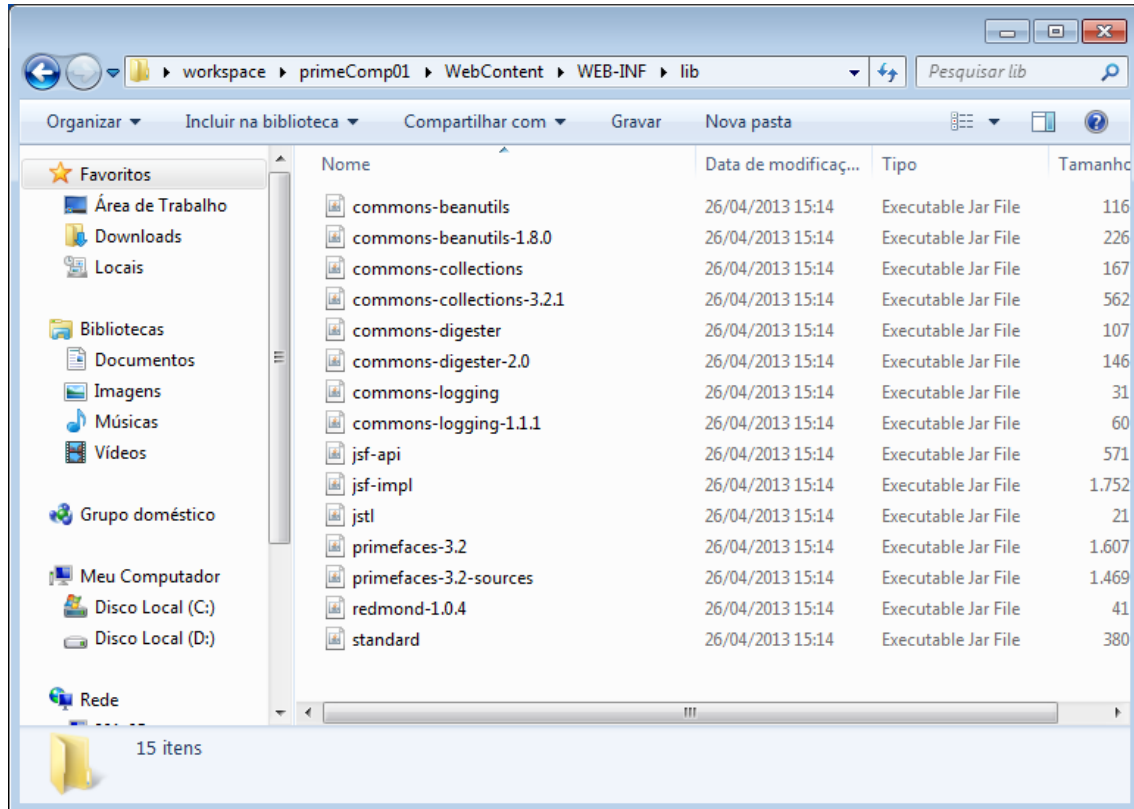
##### faces-config.xml

```
<?xml version="1.0" encoding="UTF-8"?>

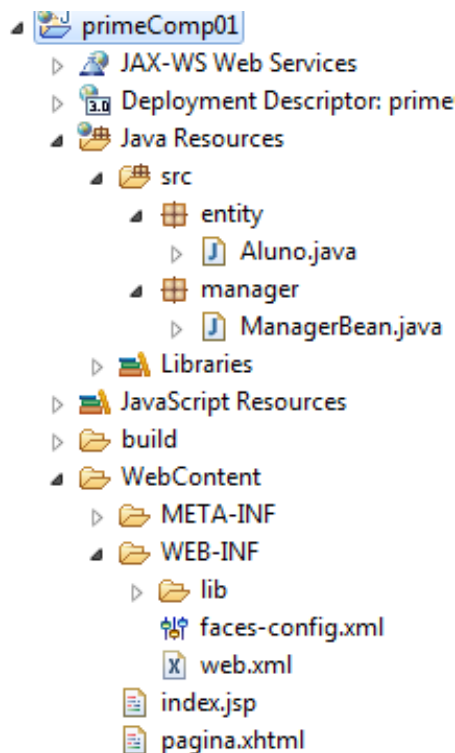
<faces-config xmlns="http://java.sun.com/xml/ns/javaee"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
    http://java.sun.com/xml/ns/javaee/web-facesconfig_2_0.xsd"
  version="2.0">

</faces-config>
```

## Bibliotecas do Jsf 2.0



## Estrutura de arquivos do Projeto...



### Classe Aluno (**Entidade**)

```
package entity;

import java.util.Arrays;

public class Aluno {

    private Integer idAluno;
    private String nome;
    private String sexo;
    private String estadoCivil;
    private String[] cursos;

    public Aluno() {

    }

    public Aluno(Integer idAluno, String nome, String sexo,
                  String estadoCivil,
                  String[] cursos) {
        super();
        this.idAluno = idAluno;
        this.nome = nome;
        this.sexo = sexo;
        this.estadoCivil = estadoCivil;
        this.cursos = cursos;
    }

    @Override
    public String toString() {
        return "Aluno [idAluno=" + idAluno + ", nome=" + nome
            + ", sexo=" + sexo + ", estadoCivil=" + estadoCivil
            + ", cursos=" + Arrays.toString(cursos) + "]";
    }

    public Integer getIdAluno() {
        return idAluno;
    }

    public void setIdAluno(Integer idAluno) {
        this.idAluno = idAluno;
    }

    public String getNome() {
        return nome;
    }

    public void setNome(String nome) {
        this.nome = nome;
    }

    public String getSexo() {
        return sexo;
    }
}
```

```
public void setSexo(String sexo) {
    this.sexo = sexo;
}

public String getEstadoCivil() {
    return estadoCivil;
}

public void setEstadoCivil(String estadoCivil) {
    this.estadoCivil = estadoCivil;
}

public String[] getCursos() {
    return cursos;
}

public void setCursos(String[] cursos) {
    this.cursos = cursos;
}
}
```

### ManagerBean (Gerenciador)

```
package manager;

import java.util.ArrayList;
import java.util.List;

import javax.faces.application.FacesMessage;
import javax.faces.bean.ManagedBean;
import javax.faces.bean.RequestScoped;
import javax.faces.context.FacesContext;
import javax.faces.model.SelectItem;

import entity.Aluno;

@ManagedBean(name = "ManagerBean")
@RequestScoped
public class ManagerBean {

    private Aluno aluno;

    private List<SelectItem> sexos;
    private List<SelectItem> estadosCivil;
    private List<SelectItem> cursos;

    public ManagerBean() {
        aluno = new Aluno();
    }

    public void cadastrar() {

        FacesContext fc = FacesContext.getCurrentInstance();
        fc.addMessage("form1", new FacesMessage(aluno.toString()));

    }
}
```



```
public Aluno getAluno() {
    return aluno;
}

public void setAluno(Aluno aluno) {
    this.aluno = aluno;
}

public List<SelectItem> getSexos() {

    sexos = new ArrayList<SelectItem>();

    sexos.add(new SelectItem("M", "Masculino"));
    sexos.add(new SelectItem("F", "Feminino"));

    return sexos;
}

public void setSexos(List<SelectItem> sexos) {
    this.sexos = sexos;
}

public List<SelectItem> getEstadosCivil() {

    estadosCivil = new ArrayList<SelectItem>();

    estadosCivil.add(new SelectItem("Casado", "Casado"));
    estadosCivil.add(new SelectItem("Solteiro", "Solteiro"));
    estadosCivil.add(new SelectItem("Viúvo", "Viúvo"));
    estadosCivil.add(new SelectItem("Divorciado", "Divorciado"));

    return estadosCivil;
}

public void setEstadosCivil(List<SelectItem> estadosCivil) {
    this.estadosCivil = estadosCivil;
}

public List<SelectItem> getCursos() {

    cursos = new ArrayList<SelectItem>();

    cursos.add(new SelectItem("Java", "WebDeveloper Java"));
    cursos.add(new SelectItem("Oracle", "Oracle DBA"));
    cursos.add(new SelectItem("Design", "WebDesign HTML5"));
    cursos.add(new SelectItem("PHP", "WebDeveloper PHP"));

    return cursos;
}

public void setCursos(List<SelectItem> cursos) {
    this.cursos = cursos;
}

}
```

### pagina.xhtml (View)

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<!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:f="http://java.sun.com/jsf/core"
      xmlns:p="http://primefaces.org/ui">
  <h:head>
    <meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1" />
    <title>COTI Informática</title>
  </h:head>
  <h:body>
    <p:panel header="Componentes de Seleção Único e Múltiplo">

      <h:form id="form1">

        <h:panelGrid columns="2" style="padding: 6px;">

          <h:outputText value="Id do Aluno:"/>
          <p:spinner value="#{ManagerBean.aluno.idAluno}" size="5"/>

          <h:outputText value="Nome do Aluno:"/>
          <p:inputText value="#{ManagerBean.aluno.nome}"/>

          <h:outputText value="Sexo:"/>
          <p:selectOneRadio value="#{ManagerBean.aluno.sexo}">
            <f:selectItems value="#{ManagerBean.sexos}"/>
          </p:selectOneRadio>

          <h:outputText value="Estado Civil:"/>
          <p:selectOneMenu value="#{ManagerBean.aluno.estadoCivil}">
            <f:selectItems value="#{ManagerBean.estadosCivil}"/>
          </p:selectOneMenu>

          <h:outputText value="Curso(s):"/>
          <p:selectManyCheckbox value="#{ManagerBean.aluno.cursos}"
            layout="pageDirection">
            <f:selectItems value="#{ManagerBean.cursos}"/>
          </p:selectManyCheckbox>

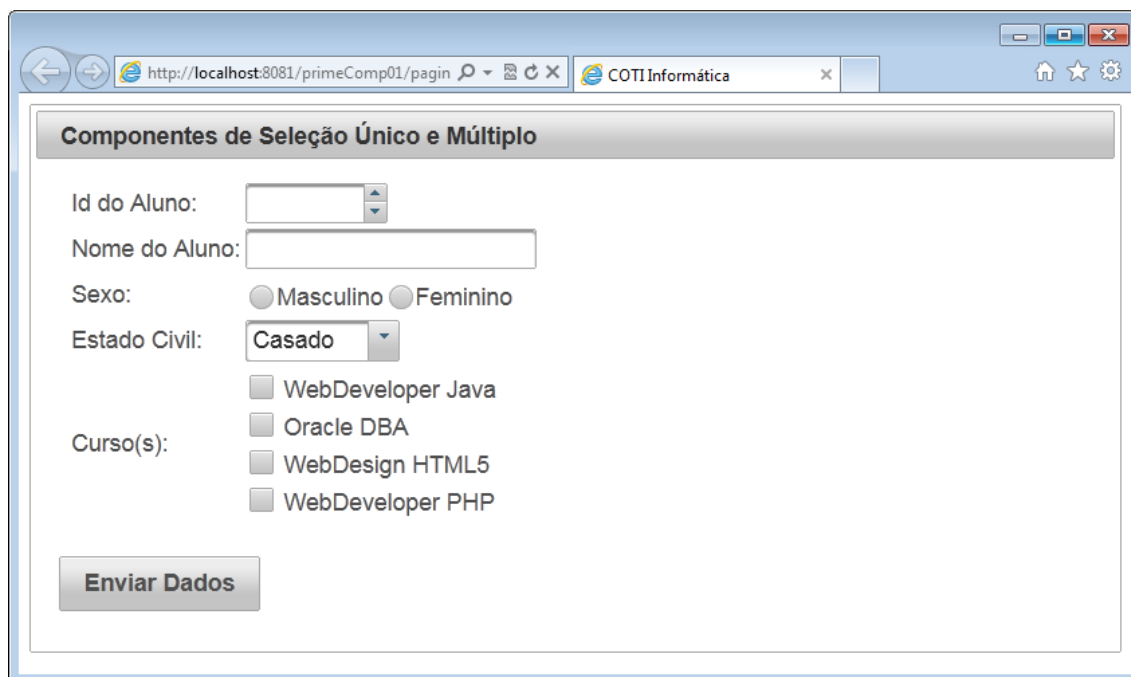
        </h:panelGrid>

        <p>
          <p:commandButton value="Enviar Dados"
            action="#{ManagerBean.cadastrar}"
            ajax="true" update="form1"/>
        </p>

        <h:messages/>

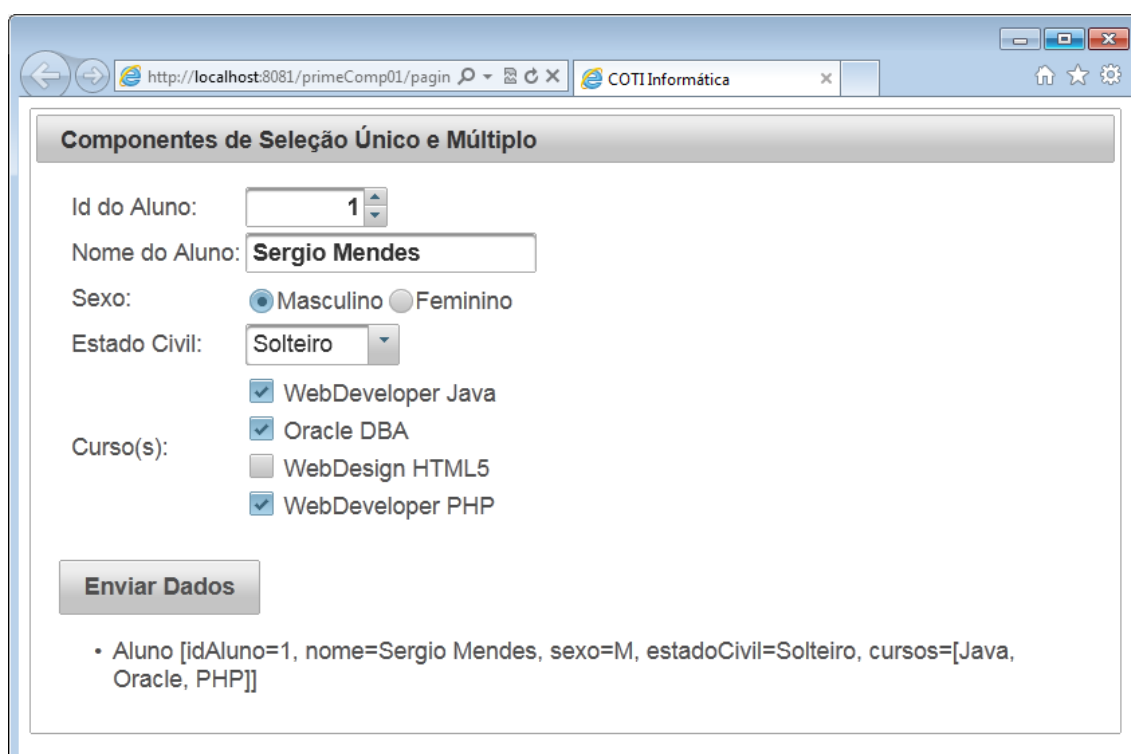
      </h:form>
    </p:panel>
  </h:body>
</html>
```

Resultado...



The screenshot shows a web browser window with the URL `http://localhost:8081/primeComp01/pagin`. The page title is "Componentes de Seleção Único e Múltiplo". The form contains the following fields and options:

- Id do Aluno:** A spinner box with the value 1.
- Nome do Aluno:** A text box containing "Sergio Mendes".
- Sexo:** Radio buttons for "Masculino" (selected) and "Feminino".
- Estado Civil:** A dropdown menu showing "Casado".
- Curso(s):** A list of checkboxes for "WebDeveloper Java", "Oracle DBA", "WebDesign HTML5", and "WebDeveloper PHP", all of which are currently unchecked.
- Enviar Dados:** A button at the bottom of the form.



The screenshot shows the same web browser window after the form has been submitted. The form fields are now populated with the following data:

- Id do Aluno:** A spinner box with the value 1.
- Nome do Aluno:** A text box containing "Sergio Mendes".
- Sexo:** Radio buttons for "Masculino" (selected) and "Feminino".
- Estado Civil:** A dropdown menu showing "Solteiro".
- Curso(s):** A list of checkboxes for "WebDeveloper Java", "Oracle DBA", "WebDesign HTML5", and "WebDeveloper PHP". The checkboxes for "WebDeveloper Java", "Oracle DBA", and "WebDeveloper PHP" are now checked, while "WebDesign HTML5" remains unchecked.
- Enviar Dados:** A button at the bottom of the form.

Below the form, a message is displayed:

- Aluno [idAluno=1, nome=Sergio Mendes, sexo=M, estadoCivil=Solteiro, cursos=[Java, Oracle, PHP]]

## Parte 2: Campo SelectCheckBoxMenu

### Configuração do Jsf 2.0 com Primefaces:

#### web.xml

```
<?xml version="1.0" encoding="UTF-8"?>

<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns="http://java.sun.com/xml/ns/javaee"
  xmlns:web="http://java.sun.com/xml/ns/javaee/web-app_2_5.xsd"
  xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
    http://java.sun.com/xml/ns/javaee/web-app_3_0.xsd"
  version="3.0">

  <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>

  <context-param>
    <param-name>javax.faces.STATE_SAVING_METHOD</param-name>
    <param-value>client</param-value>
  </context-param>
  <context-param>
    <param-name>javax.servlet.jsp.jstl.fmt.localizationContext</param-name>
    <param-value>resources.application</param-value>
  </context-param>
  <listener>
    <listener-class>com.sun.faces.config.ConfigureListener</listener-class>
  </listener>

  <welcome-file-list>
    <welcome-file>index.jsp</welcome-file>
  </welcome-file-list>

</web-app>
```

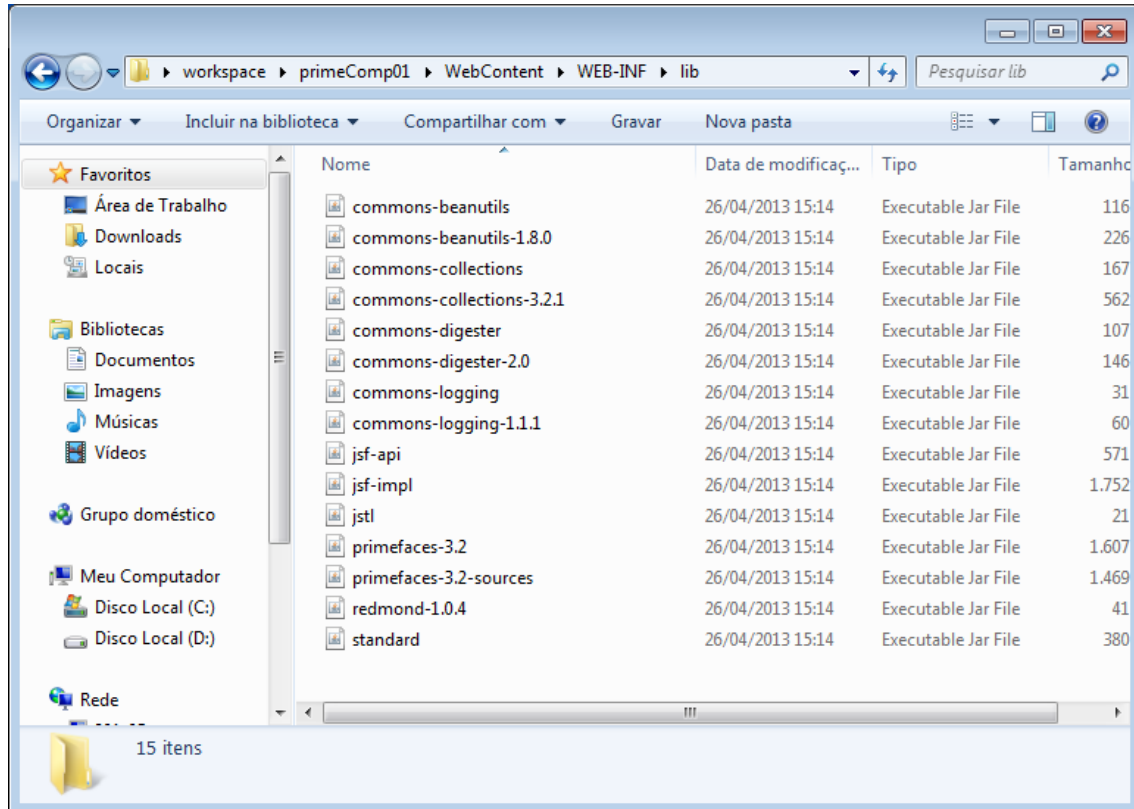
#### faces-config.xml

```
<?xml version="1.0" encoding="UTF-8"?>

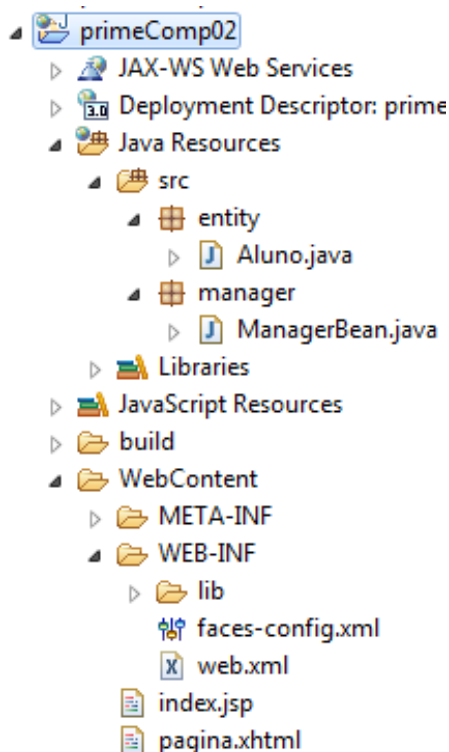
<faces-config xmlns="http://java.sun.com/xml/ns/javaee"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
    http://java.sun.com/xml/ns/javaee/web-facesconfig_2_0.xsd"
  version="2.0">

</faces-config>
```

## Bibliotecas do Jsf 2.0



## Estrutura de arquivos do Projeto...



### Classe Aluno (**Entidade**)

```
package entity;

import java.util.List;

public class Aluno {

    private Integer idAluno;
    private String nome;
    private List<String> cursos;

    public Aluno() {

    }

    public Aluno(Integer idAluno, String nome, List<String> cursos) {
        super();
        this.idAluno = idAluno;
        this.nome = nome;
        this.cursos = cursos;
    }

    @Override
    public String toString() {
        return "Aluno [idAluno=" + idAluno + ", nome="
            + nome + ", cursos=" + cursos + "]";
    }

    public Integer getIdAluno() {
        return idAluno;
    }

    public void setIdAluno(Integer idAluno) {
        this.idAluno = idAluno;
    }

    public String getNome() {
        return nome;
    }

    public void setNome(String nome) {
        this.nome = nome;
    }

    public List<String> getCursos() {
        return cursos;
    }

    public void setCursos(List<String> cursos) {
        this.cursos = cursos;
    }

}
```

### ManagerBean (Gerenciador)

```
package manager;

import java.util.LinkedHashMap;
import java.util.Map;

import javax.faces.application.FacesMessage;
import javax.faces.bean.ManagedBean;
import javax.faces.bean.RequestScoped;
import javax.faces.context.FacesContext;

import entity.Aluno;

@ManagedBean(name="ManagerBean")
@RequestScoped
public class ManagerBean {

    private Aluno aluno;
    private Map<String, String> cursos;

    public ManagerBean() {
        aluno = new Aluno();
    }

    public void cadastrar() {
        FacesContext fc = FacesContext.getCurrentInstance();
        fc.addMessage("form1", new FacesMessage(aluno.toString()));
    }

    public Aluno getAluno() {
        return aluno;
    }

    public void setAluno(Aluno aluno) {
        this.aluno = aluno;
    }

    public Map<String, String> getCursos() {

        cursos = new LinkedHashMap<String, String>();

        cursos.put("WebDeveloper em Java", "Java");
        cursos.put("WebDeveloper em C#", "C#");
        cursos.put("Banco de dados Oracle", "Oracle");
        cursos.put("Banco de dados SqlServer", "SqlServer");
        cursos.put("WebDesign HTML5", "Design");

        return cursos;
    }

    public void setCursos(Map<String, String> cursos) {
        this.cursos = cursos;
    }

}
```

### pagina.xhtml (View)

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<!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:f="http://java.sun.com/jsf/core"
      xmlns:p="http://primefaces.org/ui">
  <h:head>
    <meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1" />
    <title>COTI Informática</title>
  </h:head>
  <h:body>

    <p:panel header="Componentes de Seleção Único e Múltiplo">

      <h:form id="form1">

        <h:panelGrid columns="2" style="padding: 6px;">

          <h:outputText value="Id do Aluno:" />
          <p:spinner value="#{ManagerBean.aluno.idAluno}" size="5"/>

          <h:outputText value="Nome do Aluno:" />
          <p:inputText value="#{ManagerBean.aluno.nome}" />

          <h:outputText value="Curso(s):" />
          <p:selectCheckboxMenu value="#{ManagerBean.aluno.cursos}"
            label="Selecione o Curso"
            filter="true" filterText="Filter"
            filterMatchMode="startsWith"
            panelStyle="width:400px">

            <f:selectItems value="#{ManagerBean.cursos}" />

          </p:selectCheckboxMenu>

        </h:panelGrid>

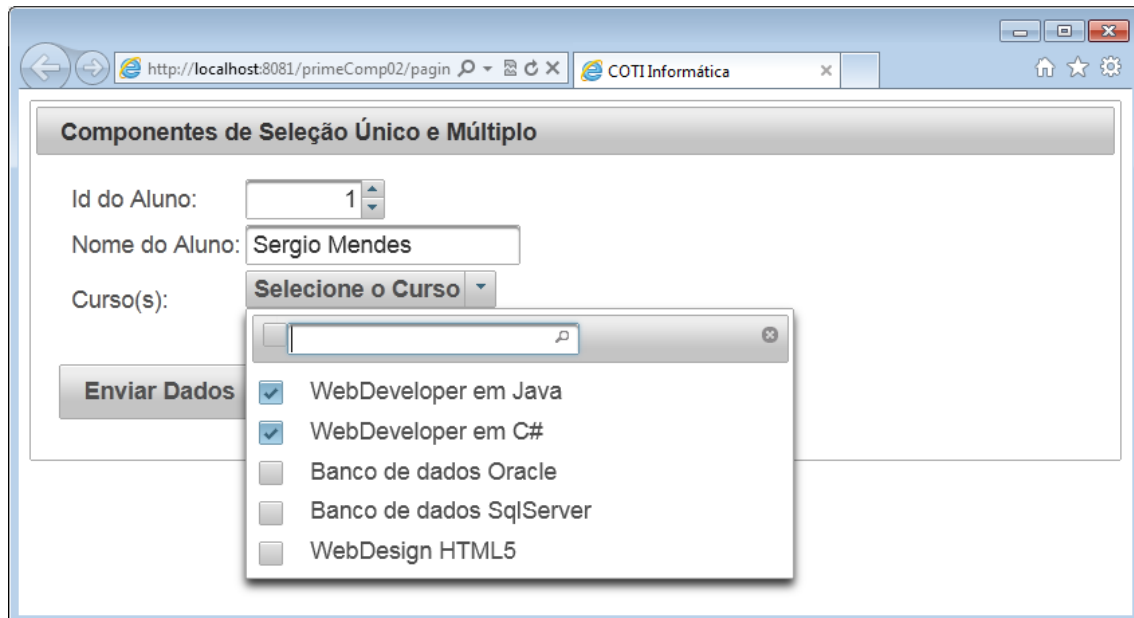
        <p>
          <p:commandButton value="Enviar Dados"
            action="#{ManagerBean.cadastrar}"
            ajax="true" update="form1"
            oncomplete="janela.show()" />
        </p>

        <p:dialog widgetVar="janela"
          header="Dados Enviados" modal="true">
          <h:messages />
        </p:dialog>

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



Resultado...



Componentes de Seleção Único e Múltiplo

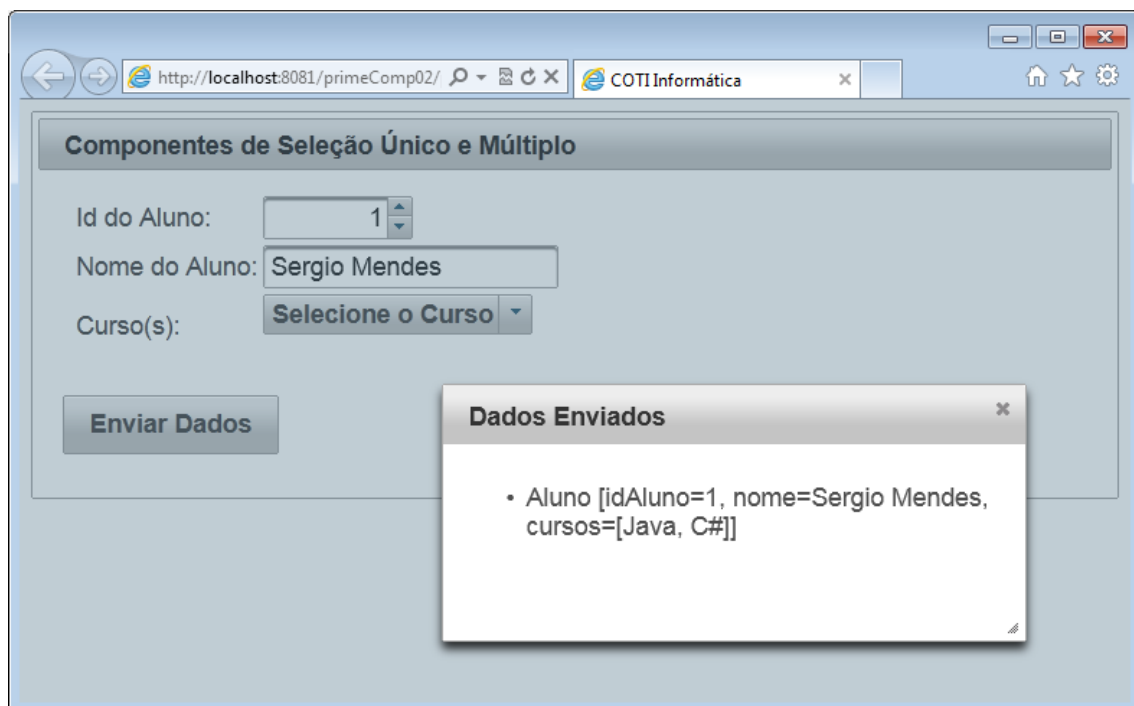
Id do Aluno: 1

Nome do Aluno: Sergio Mendes

Curso(s): Selezione o Curso

Enviar Dados

- ☒ WebDeveloper em Java
- ☒ WebDeveloper em C#
- ☐ Banco de dados Oracle
- ☐ Banco de dados SqlServer
- ☐ WebDesign HTML5



Componentes de Seleção Único e Múltiplo

Id do Aluno: 1

Nome do Aluno: Sergio Mendes

Curso(s): Selezione o Curso

Enviar Dados

**Dados Enviados**

- Aluno [idAluno=1, nome=Sergio Mendes, cursos=[Java, C#]]

## Parte 3: Galeria de Imagens

### Configuração do Jsf 2.0 com Primefaces:

#### web.xml

```
<?xml version="1.0" encoding="UTF-8"?>

<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns="http://java.sun.com/xml/ns/javaee"
  xmlns:web="http://java.sun.com/xml/ns/javaee/web-app_2_5.xsd"
  xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
    http://java.sun.com/xml/ns/javaee/web-app_3_0.xsd"
  version="3.0">

  <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>

  <context-param>
    <param-name>javax.faces.STATE_SAVING_METHOD</param-name>
    <param-value>client</param-value>
  </context-param>
  <context-param>
    <param-name>javax.servlet.jsp.jstl.fmt.localizationContext</param-name>
    <param-value>resources.application</param-value>
  </context-param>
  <listener>
    <listener-class>com.sun.faces.config.ConfigureListener</listener-class>
  </listener>

  <welcome-file-list>
    <welcome-file>index.jsp</welcome-file>
  </welcome-file-list>

</web-app>
```

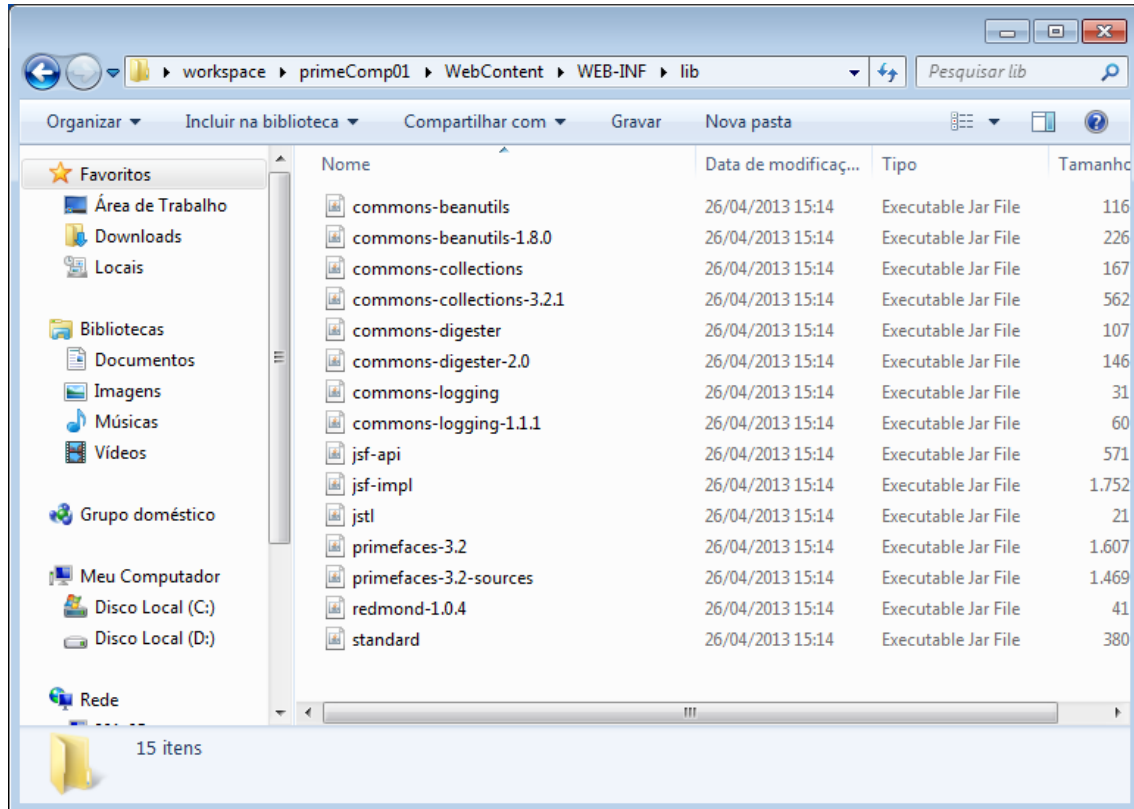
#### faces-config.xml

```
<?xml version="1.0" encoding="UTF-8"?>

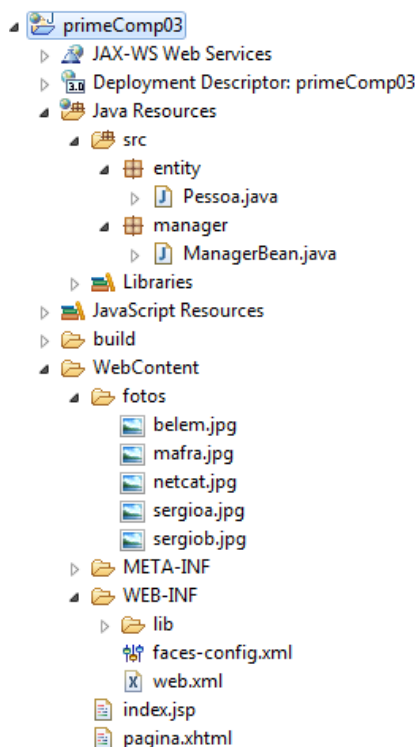
<faces-config xmlns="http://java.sun.com/xml/ns/javaee"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
    http://java.sun.com/xml/ns/javaee/web-facesconfig_2_0.xsd"
  version="2.0">

</faces-config>
```

## Bibliotecas do Jsf 2.0



## Estrutura de arquivos do Projeto...



### Classe Pessoa (**Entidade**)

```
package entity;

public class Pessoa {

    private Integer idPessoa;
    private String nome;
    private String sexo;
    private String foto;

    public Pessoa() {

    }

    public Pessoa(Integer idPessoa, String nome, String sexo, String foto){
        super();
        this.idPessoa = idPessoa;
        this.nome = nome;
        this.sexo = sexo;
        this.foto = foto;
    }

    @Override
    public String toString() {
        return "Pessoa [idPessoa=" + idPessoa + ", nome="
            + nome + ", sexo=" + sexo + ", foto=" + foto + "]";
    }

    public Integer getIdPessoa() {
        return idPessoa;
    }

    public void setIdPessoa(Integer idPessoa) {
        this.idPessoa = idPessoa;
    }

    public String getNome() {
        return nome;
    }

    public void setNome(String nome) {
        this.nome = nome;
    }

    public String getSexo() {
        return sexo;
    }

    public void setSexo(String sexo) {
        this.sexo = sexo;
    }

    public String getFoto() {
        return foto;
    }
}
```

```
        public void setFoto(String foto) {  
            this.foto = foto;  
        }  
    }  
}
```

### ManagerBean (Gerenciador)

```
package manager;  
  
import java.util.ArrayList;  
import java.util.List;  
  
import javax.faces.bean.ManagedBean;  
import javax.faces.bean.RequestScoped;  
  
import entity.Pessoa;  
  
@ManagedBean(name="ManagerBean")  
@RequestScoped  
public class ManagerBean {  
  
    private List<Pessoa> lista;  
  
    public ManagerBean() {  
    }  
  
    public List<Pessoa> getLista() {  
  
        lista = new ArrayList<Pessoa>();  
  
        lista.add(new Pessoa(1, "Rodrigo NetCat",  
                               "Feminino", "netcat.jpg"));  
  
        lista.add(new Pessoa(2, "Mafra Velho",  
                               "Feminino", "mafra.jpg"));  
  
        lista.add(new Pessoa(3, "Belem",  
                               "Masculino", "belem.jpg"));  
  
        lista.add(new Pessoa(4, "Sergio B",  
                               "Masculino", "sergiob.jpg"));  
  
        lista.add(new Pessoa(5, "Sergio A",  
                               "Masculino", "sergioa.jpg"));  
  
        return lista;  
    }  
  
    public void setLista(List<Pessoa> lista) {  
        this.lista = lista;  
    }  
  
}
```

### pagina.xhtml (**View**)

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<!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:f="http://java.sun.com/jsf/core"
      xmlns:p="http://primefaces.org/ui">
  <h:head>
    <meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1" />
    <title>COTI Informática</title>
  </h:head>
  <h:body>

    <p:panel header="Componente de Galeria de Imagens">

      <h:form id="form1">

        <p:galleria value="#{ManagerBean.Lista}" var="item"
          panelWidth="340" panelHeight="460"
          frameWidth="48" frameHeight="65">

          <p:graphicImage value="/fotos/#{item.foto}"
            title="#{item.nome}"/>

          <f:facet name="content">
            <h:panelGrid columns="2" cellpadding="5">
              <f:facet name="header">

                <p:graphicImage value="/fotos/#{item.foto}"
                  width="320" height="320"/>
              </f:facet>

              <h:outputText value="Id da Pessoa: " />
              <h:outputText value="#{item.idPessoa}"/>

              <h:outputText value="Nome: " />
              <h:outputText value="#{item.nome}"/>

              <h:outputText value="Sexo: " />
              <h:outputText value="#{item.sexo}"/>

            </h:panelGrid>
          </f:facet>
        </p:galleria>
      </h:form>
    </p:panel>
  </h:body>
</html>
```

Resultado...



## Parte 4: Componentes de Gráficos

### Configuração do Jsf 2.0 com Primefaces:

#### web.xml

```
<?xml version="1.0" encoding="UTF-8"?>

<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns="http://java.sun.com/xml/ns/javaee"
  xmlns:web="http://java.sun.com/xml/ns/javaee/web-app_2_5.xsd"
  xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
    http://java.sun.com/xml/ns/javaee/web-app_3_0.xsd"
  version="3.0">

  <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>

  <context-param>
    <param-name>javax.faces.STATE_SAVING_METHOD</param-name>
    <param-value>client</param-value>
  </context-param>
  <context-param>
    <param-name>javax.servlet.jsp.jstl.fmt.localizationContext</param-name>
    <param-value>resources.application</param-value>
  </context-param>
  <listener>
    <listener-class>com.sun.faces.config.ConfigureListener</listener-class>
  </listener>

  <welcome-file-list>
    <welcome-file>index.jsp</welcome-file>
  </welcome-file-list>

</web-app>
```

#### faces-config.xml

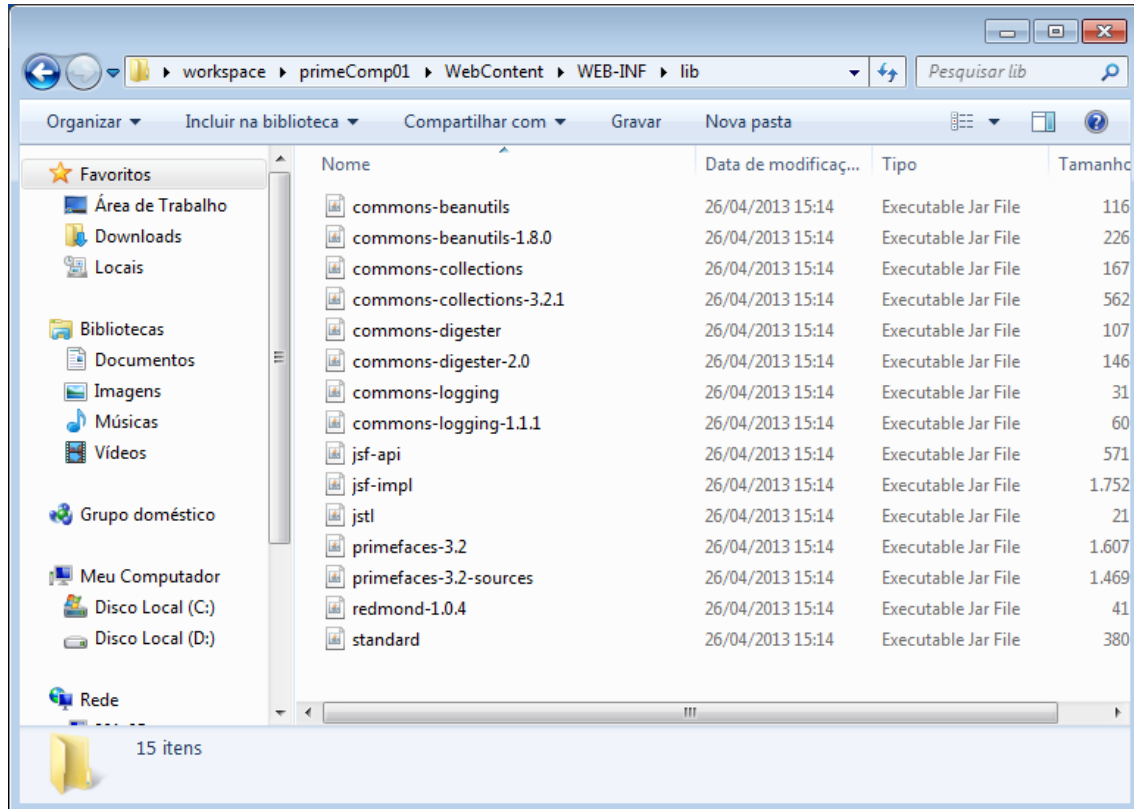
```
<?xml version="1.0" encoding="UTF-8"?>

<faces-config 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-facesconfig_2_0.xsd"
  version="2.0">

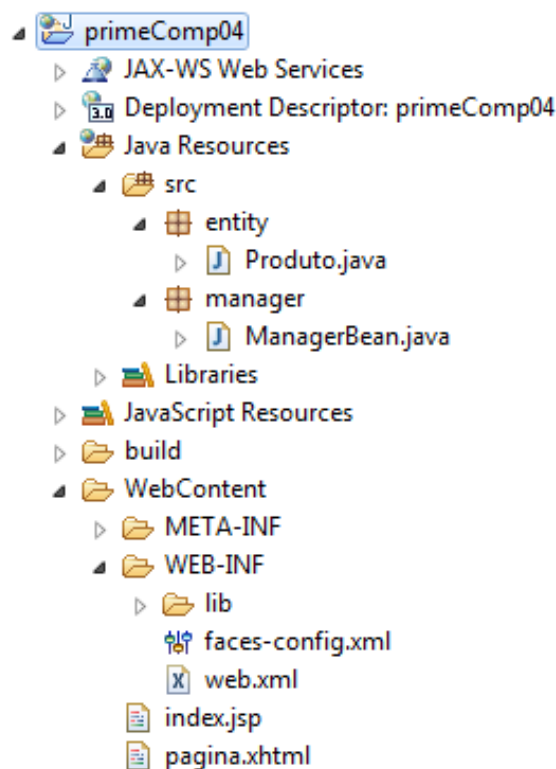
</faces-config>
```



### Bibliotecas do Jsf 2.0



### Estrutura de arquivos do Projeto...



### Classe Produto (**Entidade**)

```
package entity;

public class Produto {

    private Integer idProduto;
    private String nome;
    private Integer quantidade;

    public Produto() {

    }

    public Produto(Integer idProduto, String nome, Integer quantidade) {
        super();
        this.idProduto = idProduto;
        this.nome = nome;
        this.quantidade = quantidade;
    }

    @Override
    public String toString() {
        return "Produto [idProduto=" + idProduto + ", nome=" + nome
            + ", quantidade=" + quantidade + "]";
    }

    public Integer getIdProduto() {
        return idProduto;
    }

    public void setIdProduto(Integer idProduto) {
        this.idProduto = idProduto;
    }

    public String getNome() {
        return nome;
    }

    public void setNome(String nome) {
        this.nome = nome;
    }

    public Integer getQuantidade() {
        return quantidade;
    }

    public void setQuantidade(Integer quantidade) {
        this.quantidade = quantidade;
    }

}
```

### ManagerBean (Gerenciador)

```
package manager;

import java.util.ArrayList;
import java.util.List;

import javax.faces.bean.ManagedBean;
import javax.faces.bean.RequestScoped;

import org.primefaces.model.chart.CartesianChartModel;
import org.primefaces.model.chart.ChartSeries;
import org.primefaces.model.chart.PieChartModel;

import entity.Produto;

@ManagedBean(name = "ManagerBean")
@RequestScoped
public class ManagerBean {

    private List<Produto> lista;

    private PieChartModel grafico1;
    private CartesianChartModel grafico2;

    public ManagerBean() {

        lista = new ArrayList<Produto>();

        lista.add(new Produto(1, "Mouse", 30));
        lista.add(new Produto(2, "Monitor", 25));
        lista.add(new Produto(3, "Notebook", 10));
        lista.add(new Produto(4, "Impressora", 8));
        lista.add(new Produto(5, "Teclado", 20));
        lista.add(new Produto(6, "PenDrive", 45));

    }

    public PieChartModel getGrafico1() {

        grafico1 = new PieChartModel();

        for (Produto p : lista) {
            grafico1.set(p.getNome(), p.getQuantidade());
        }

        return grafico1;
    }

    public void setGrafico1(PieChartModel grafico1) {
        this.grafico1 = grafico1;
    }
}
```

```

public CartesianChartModel getGrafico2() {

    grafico2 = new CartesianChartModel();

    ChartSeries produtos = new ChartSeries();
    produtos.setLabel("Produtos");

    for (Produto p : lista) {
        produtos.set(p.getNome(), p.getQuantidade());
    }

    grafico2.addSeries(produtos);

    return grafico2;
}

public void setGrafico2(CartesianChartModel grafico2) {
    this.grafico2 = grafico2;
}
}

```

### pagina.xhtml (View)

```

<?xml version="1.0" encoding="ISO-8859-1" ?>
<!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:f="http://java.sun.com/jsf/core"
      xmlns:p="http://primefaces.org/ui">
<h:head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1" />
<title>COTI Informática</title>
</h:head>
<h:body>
    <p:panel header="Componentes de Gráficos">

        <h:form id="form1">

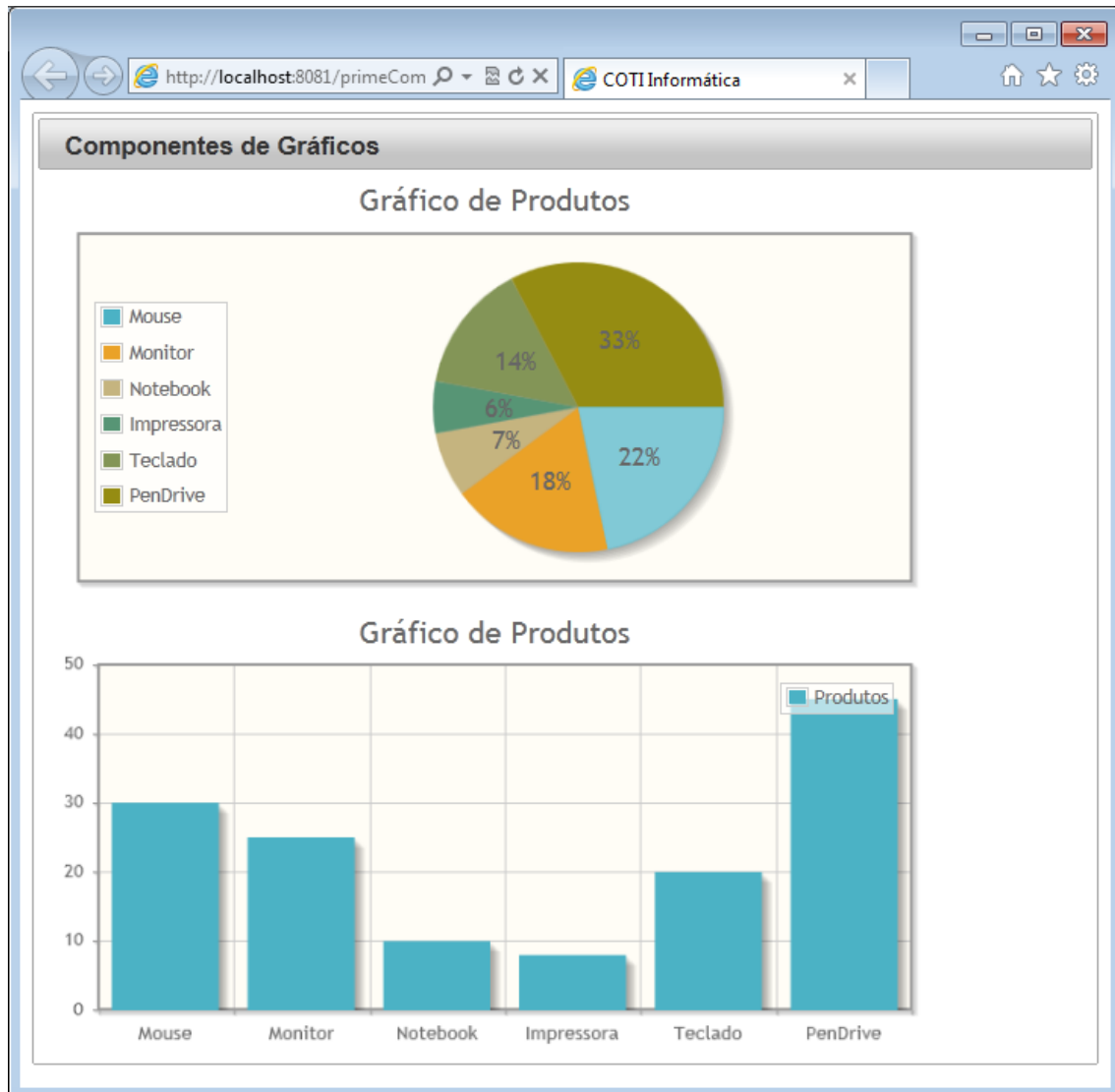
            <p:pieChart value="#{ManagerBean.grafico1}"
                title="Gráfico de Produtos"
                legendPosition="w"
                style="width:600px; height:300px"
                showDataLabels="true" />

            <p:barChart value="#{ManagerBean.grafico2}"
                title="Gráfico de Produtos"
                legendPosition="ne"
                style="width:600px; height:300px"
                showDatatip="true"/>

        </h:form>
    </p:panel>
</h:body>
</html>

```

Resultado...



### Parte 5: Captcha

#### Configuração do Jsf 2.0 com Primefaces:

##### web.xml

```
<?xml version="1.0" encoding="UTF-8"?>

<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns="http://java.sun.com/xml/ns/javaee"
  xmlns:web="http://java.sun.com/xml/ns/javaee/web-app_2_5.xsd"
  xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
    http://java.sun.com/xml/ns/javaee/web-app_3_0.xsd"
  version="3.0">

  <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>

  <context-param>
    <param-name>javax.faces.STATE_SAVING_METHOD</param-name>
    <param-value>client</param-value>
  </context-param>
  <context-param>
    <param-name>javax.servlet.jsp.jstl.fmt.localizationContext</param-name>
    <param-value>resources.application</param-value>
  </context-param>
  <listener>
    <listener-class>com.sun.faces.config.ConfigureListener</listener-class>
  </listener>

  <welcome-file-list>
    <welcome-file>index.jsp</welcome-file>
  </welcome-file-list>

</web-app>
```

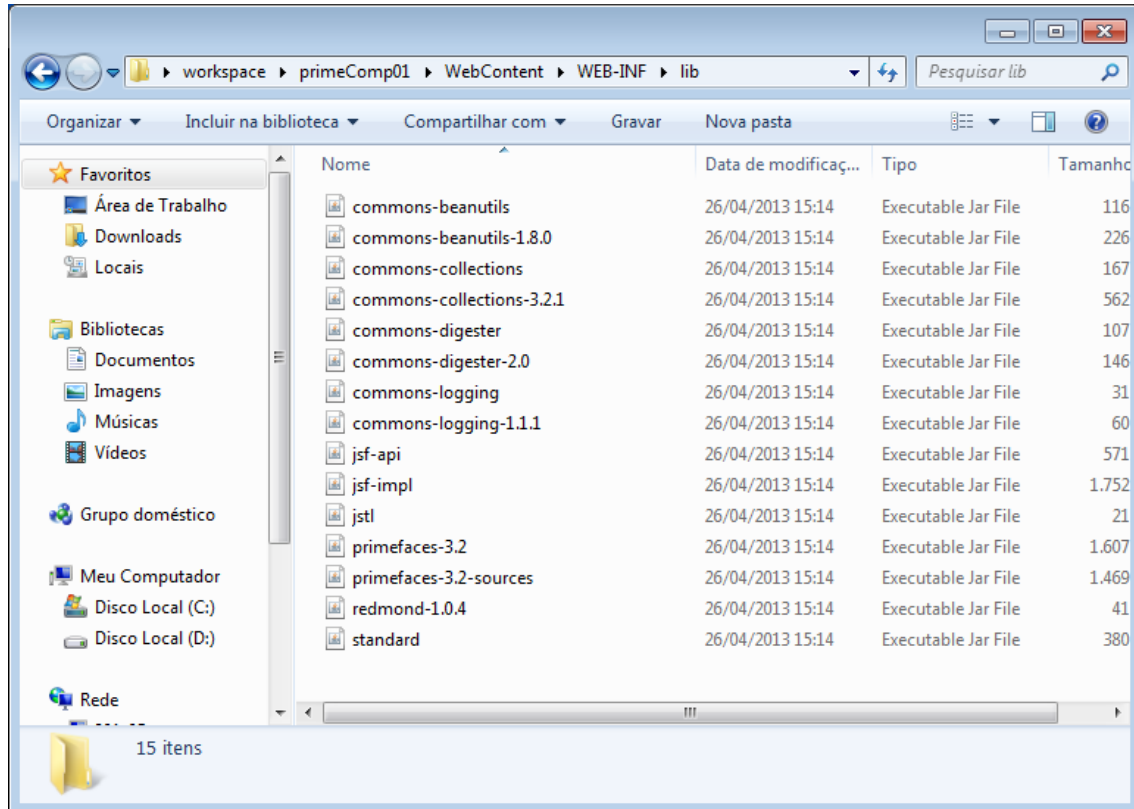
##### faces-config.xml

```
<?xml version="1.0" encoding="UTF-8"?>

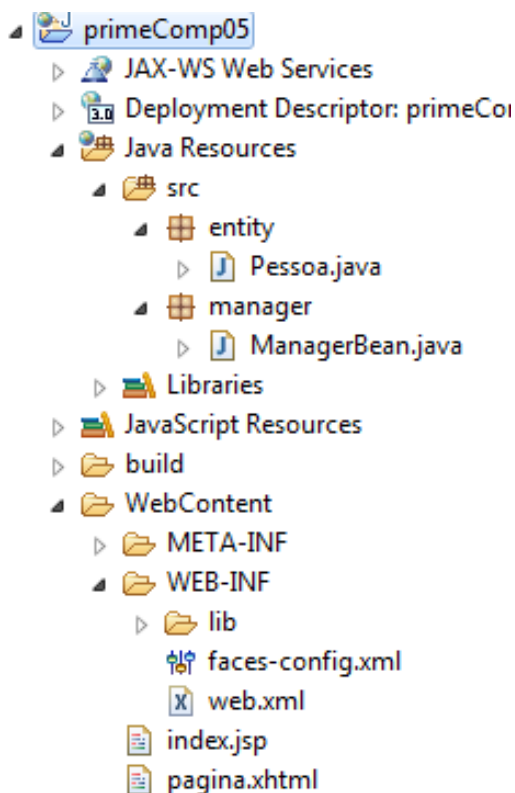
<faces-config xmlns="http://java.sun.com/xml/ns/javaee"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
    http://java.sun.com/xml/ns/javaee/web-facesconfig_2_0.xsd"
  version="2.0">

</faces-config>
```

### Bibliotecas do Jsf 2.0



### Estrutura de arquivos do Projeto...



### Classe Pessoa (**Entidade**)

```
package entity;

public class Pessoa {

    private Integer idPessoa;
    private String nome;

    public Pessoa() {

    }

    public Pessoa(Integer idPessoa, String nome) {
        super();
        this.idPessoa = idPessoa;
        this.nome = nome;
    }

    public Integer getIdPessoa() {
        return idPessoa;
    }

    public void setIdPessoa(Integer idPessoa) {
        this.idPessoa = idPessoa;
    }

    public String getNome() {
        return nome;
    }

    public void setNome(String nome) {
        this.nome = nome;
    }

    @Override
    public String toString() {
        return "Pessoa [idPessoa=" + idPessoa + ", nome=" + nome + "];"
    }

}
```

### ManagerBean (**Gerenciador**)

```
package manager;

import javax.faces.application.FacesMessage;
import javax.faces.bean.ManagedBean;
import javax.faces.bean.RequestScoped;
import javax.faces.context.FacesContext;

import entity.Pessoa;
```



```
@ManagedBean(name = "ManagerBean")
@RequestScoped
public class ManagerBean {

    private Pessoa pessoa;

    public ManagerBean() {
        pessoa = new Pessoa();
    }

    public void resgatar() {

        FacesContext.getCurrentInstance().addMessage("form1",
            new FacesMessage(pessoa.toString()));
        pessoa = new Pessoa();

    }

    public Pessoa getPessoa() {
        return pessoa;
    }

    public void setPessoa(Pessoa pessoa) {
        this.pessoa = pessoa;
    }

}
```

### pagina.xhtml (View)

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<!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:f="http://java.sun.com/jsf/core"
    xmlns:p="http://primefaces.org/ui">
<h:head>
    <meta http-equiv="Content-Type"
        content="text/html; charset=ISO-8859-1" />
    <title>COTI Informática</title>
</h:head>
<h:body>

    <p:panel header="Componente de Captcha">

        <h:form id="form1">

            <h:panelGrid columns="2">

                <h:outputText value="Id da Pessoa:"/>
                <p:inputText
                    value="#{ManagerBean.pessoa.idPessoa}"/>

                <h:outputText value="Nome:"/>
                <p:inputText
                    value="#{ManagerBean.pessoa.nome}"/>

            </h:panelGrid>

        </h:form>

    </p:panel>

</h:body>
```

```

        </h:panelGrid>

        <p:captcha label="Captcha" />
        <br/><br/>

        <p:commandButton value="Enviar Dados" ajax="false"
            action="#{ManagerBean.resgatar}" />

        <p:messages/>

    </h:form>

</p:panel>

</h:body>
</html>

```

### Configurando o Captcha do Primefaces no web.xml...

```

<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns="http://java.sun.com/xml/ns/javaee"
    xmlns:web="http://java.sun.com/xml/ns/javaee/web-app_2_5.xsd"
    xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
        http://java.sun.com/xml/ns/javaee/web-app_3_0.xsd"
    version="3.0">
    <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>

    <context-param>
        <description>State saving method: 'client' or 'server'
            (=default). See JSF Specification 2.5.2</description>
        <param-name>javax.faces.STATE_SAVING_METHOD</param-name>
        <param-value>client</param-value>
    </context-param>
    <context-param>
        <param-name>
            javax.servlet.jsp.jstl.fmt.localizationContext</param-name>
        <param-value>resources.application</param-value>
    </context-param>

    <context-param>
        <param-name>primefaces.PRIVATE_CAPTCHA_KEY</param-name>
        <param-value>
            6LfYJOESAAAAAPig0XEEIEeohCvYJ0sy6B_Ql6lF
        </param-value>
    </context-param>

```

```
<context-param>
    <param-name>primefaces.PUBLIC_CAPTCHA_KEY</param-name>
    <param-value>
        6LfyJOESAAAAAP_THL03VJlq4m6BY8Yd_n674FoP
    </param-value>
</context-param>

<listener>
    <listener-class>
        com.sun.faces.config.ConfigureListener</listener-class>
</listener>

<welcome-file-list>
    <welcome-file>index.jsp</welcome-file>
</welcome-file-list>

</web-app>
```

Resultado...



## Parte 6: Download e Exportação de dados

### Configuração do Jsf 2.0 com Primefaces:

#### web.xml

```
<?xml version="1.0" encoding="UTF-8"?>

<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns="http://java.sun.com/xml/ns/javaee"
  xmlns:web="http://java.sun.com/xml/ns/javaee/web-app_2_5.xsd"
  xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
    http://java.sun.com/xml/ns/javaee/web-app_3_0.xsd"
  version="3.0">

  <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>

  <context-param>
    <param-name>javax.faces.STATE_SAVING_METHOD</param-name>
    <param-value>client</param-value>
  </context-param>
  <context-param>
    <param-name>javax.servlet.jsp.jstl.fmt.localizationContext</param-name>
    <param-value>resources.application</param-value>
  </context-param>
  <listener>
    <listener-class>com.sun.faces.config.ConfigureListener</listener-class>
  </listener>

  <welcome-file-list>
    <welcome-file>index.jsp</welcome-file>
  </welcome-file-list>

</web-app>
```

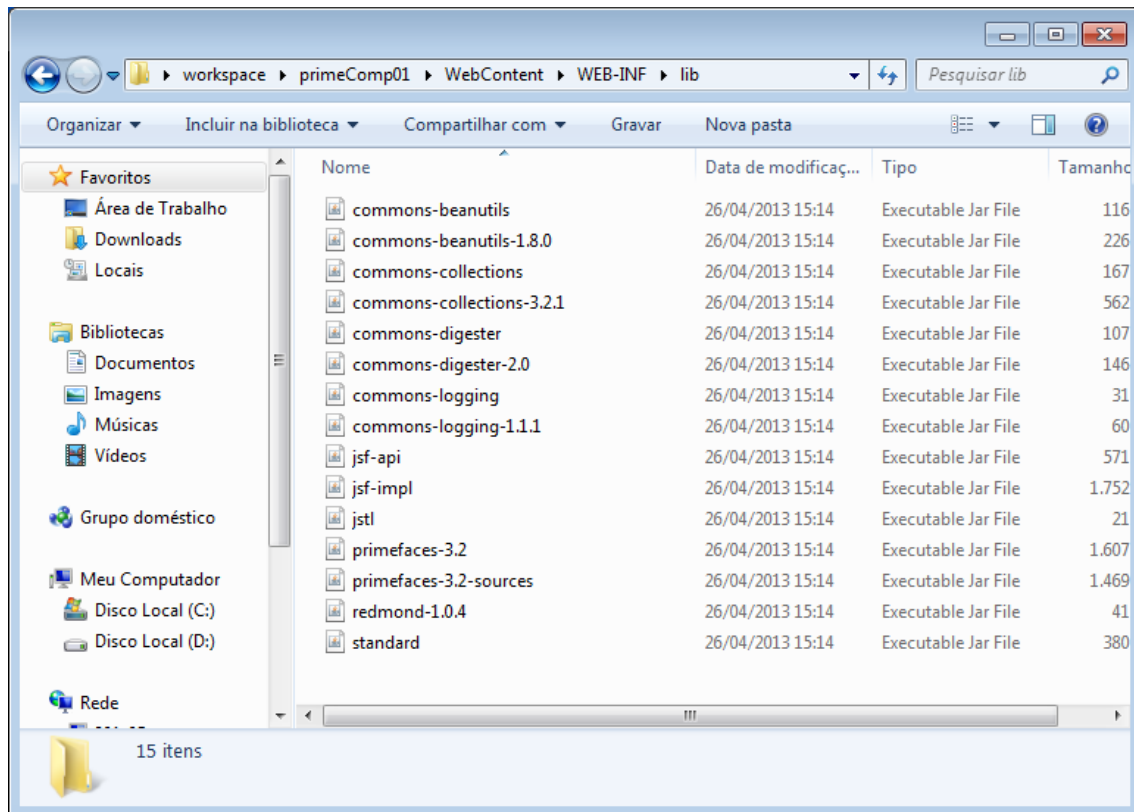
#### faces-config.xml

```
<?xml version="1.0" encoding="UTF-8"?>

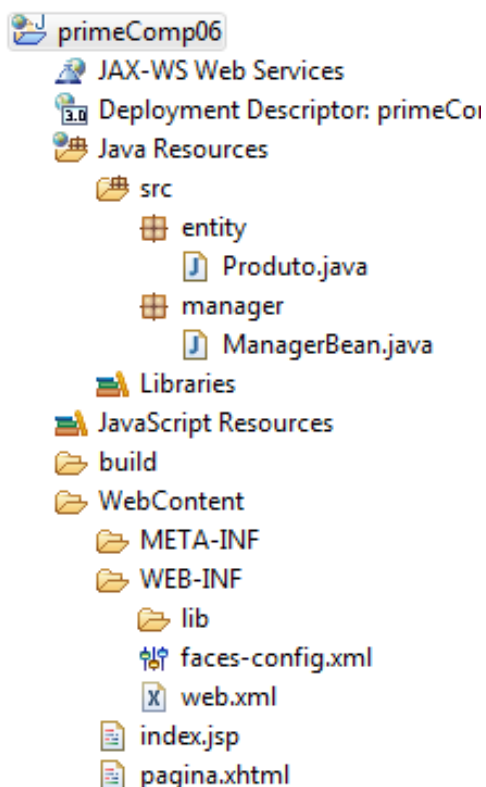
<faces-config xmlns="http://java.sun.com/xml/ns/javaee"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
    http://java.sun.com/xml/ns/javaee/web-facesconfig_2_0.xsd"
  version="2.0">

</faces-config>
```

### Bibliotecas do Jsf 2.0



### Estrutura de arquivos do Projeto...



### Classe Produto (**Entidade**)

```
package entity;

public class Produto {

    private Integer idProduto;
    private String nome;
    private Double preco;
    private Integer quantidade;

    public Produto() {

    }

    public Produto(Integer idProduto, String nome, Double preco,
        Integer quantidade) {
        super();
        this.idProduto = idProduto;
        this.nome = nome;
        this.preco = preco;
        this.quantidade = quantidade;
    }

    public Integer getIdProduto() {
        return idProduto;
    }

    public void setIdProduto(Integer idProduto) {
        this.idProduto = idProduto;
    }

    public String getNome() {
        return nome;
    }

    public void setNome(String nome) {
        this.nome = nome;
    }

    public Double getPreco() {
        return preco;
    }

    public void setPreco(Double preco) {
        this.preco = preco;
    }

    public Integer getQuantidade() {
        return quantidade;
    }

    public void setQuantidade(Integer quantidade) {
        this.quantidade = quantidade;
    }
}
```

```
@Override
public String toString() {
    return "Produto [idProduto=" + idProduto + ", nome=" + nome
        + ", preco=" + preco + ", quantidade="
        + quantidade + "]\n";
}

}
```

### ManagerBean (Gerenciador)

```
package manager;

import java.util.ArrayList;
import java.util.List;

import javax.faces.bean.ManagedBean;
import javax.faces.bean.RequestScoped;

import entity.Produto;

@ManagedBean(name = "ManagerBean")
@RequestScoped
public class ManagerBean {

    private List<Produto> lista;

    public ManagerBean() {

    }

    public List<Produto> getList() {

        lista = new ArrayList<Produto>();

        lista.add(new Produto(1, "Cenoura", 10.0, 10));
        lista.add(new Produto(2, "Aipim", 2.0, 20));
        lista.add(new Produto(3, "Beringela", 3.0, 15));
        lista.add(new Produto(4, "Nabo", 5.0, 12));
        lista.add(new Produto(5, "Pepino", 3.0, 30));
        lista.add(new Produto(6, "Calabresa", 4.0, 40));
        lista.add(new Produto(7, "Paio", 2.0, 6));
        lista.add(new Produto(8, "Banana", 3.0, 20));

        return lista;
    }

    public void setLista(List<Produto> lista) {
        this.lista = lista;
    }

}
```

### pagina.xhtml (View)

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<!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:f="http://java.sun.com/jsf/core"
      xmlns:p="http://primefaces.org/ui">
  <h:head>

    <meta http-equiv="Content-Type"
          content="text/html; charset=ISO-8859-1" />
    <title>COTI Informática</title>

  </h:head>
  <h:body>

    <p:panel header="Componente de Exportação de dados">

      <h:form id="form1">

        <p:dataTable id="tabela" value="#{ManagerBean.Lista}"
                     var="p" paginator="true" rows="5">

          <p:column headerText="Código">
            <h:outputText value="#{p.idProduto}"/>
          </p:column>

          <p:column headerText="Nome do Produto">
            <h:outputText value="#{p.nome}"/>
          </p:column>

          <p:column headerText="Preço">
            <h:outputText value="#{p.preco}">
              <f:convertNumber type="currency"/>
            </h:outputText>
          </p:column>

          <p:column headerText="Quantidade">
            <h:outputText value="#{p.quantidade}"/>
          </p:column>

          <p:column headerText="Total">
            <h:outputText
              value="#{p.preco * p.quantidade}">
              <f:convertNumber type="currency"/>
            </h:outputText>
          </p:column>

        </p:dataTable>

        <p:commandButton value="Excel" ajax="false">
          <p:dataExporter type="xls" target="tabela" fileName="produtos" />
        </p:commandButton>

      </h:form>

    </p:panel>

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



```
<p:commandButton value="PDF" ajax="false">
    <p:dataExporter type="pdf" target="tabela" fileName="produtos" />
</p:commandButton>

<p:commandButton value="CSV" ajax="false">
    <p:dataExporter type="csv" target="tabela" fileName="produtos" />
</p:commandButton>

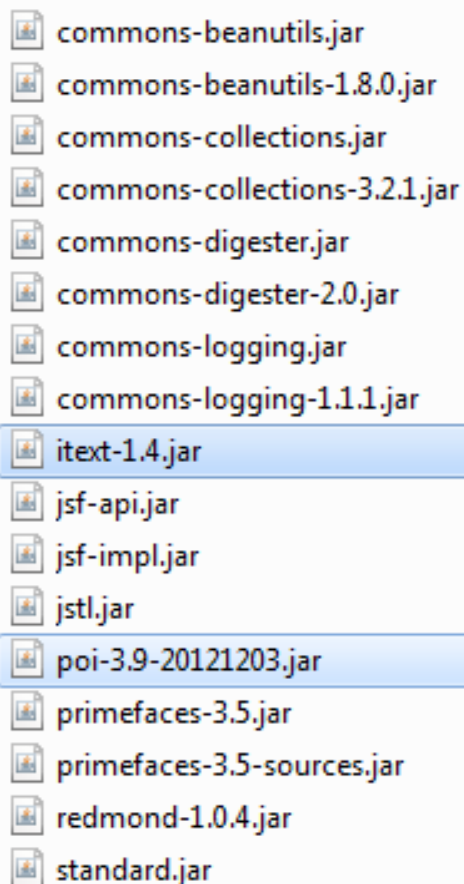
<p:commandButton value="XML" ajax="false">
    <p:dataExporter type="xml" target="tabela" fileName="produtos" />
</p:commandButton>

</h:form>

</p:panel>

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

### Bibliotecas do Apache POI e IText...



- commons-beanutils.jar
- commons-beanutils-1.8.0.jar
- commons-collections.jar
- commons-collections-3.2.1.jar
- commons-digester.jar
- commons-digester-2.0.jar
- commons-logging.jar
- commons-logging-1.1.1.jar
- itext-1.4.jar
- jsf-api.jar
- jsf-impl.jar
- jstl.jar
- poi-3.9-20121203.jar
- primefaces-3.5.jar
- primefaces-3.5-sources.jar
- redmond-1.0.4.jar
- standard.jar

Resultado...

COTI Informática x

localhost:8081/primeComp06/pagina.jsf

**Componente de Exportação de dados**

Código	Nome do Produto	Preço	Quantidade	Total
1	Cenoura	R\$ 10,00	10	R\$ 100,00
2	Aipim	R\$ 2,00	20	R\$ 40,00
3	Beringela	R\$ 3,00	15	R\$ 45,00
4	Nabo	R\$ 5,00	12	R\$ 60,00
5	Pepino	R\$ 3,00	30	R\$ 90,00

Excel PDF CSV XML

produtos (1).pdf - Foxit Reader

Arquivo Editar Exibir Ferramentas Comentários Formulários Ajuda

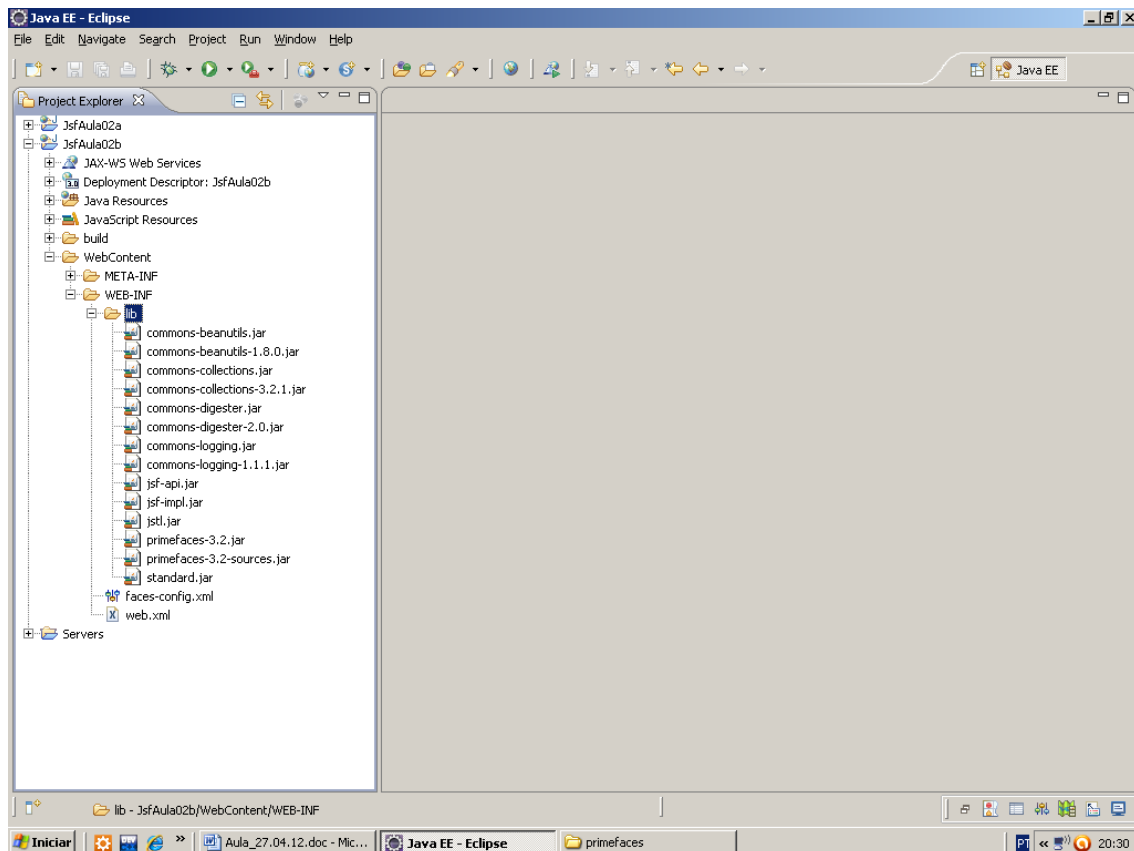
1	Cenoura	R\$ 10,00	10	R\$ 100,00
2	Aipim	R\$ 2,00	20	R\$ 40,00
3	Beringela	R\$ 3,00	15	R\$ 45,00
4	Nabo	R\$ 5,00	12	R\$ 60,00
5	Pepino	R\$ 3,00	30	R\$ 90,00
6	Calabresa	R\$ 4,00	40	R\$ 160,00
7	Paio	R\$ 2,00	6	R\$ 12,00
8	Banana	R\$ 3,00	20	R\$ 60,00

## Parte 7: Criação de Templates com Facelets

### Facelets (Criação de Layouts)

Recurso que permite a criação de templates em páginas Jsf  
Foi incorporado a partir da versão 2.0 do Jsf

- Facelets – Templates
- PrimeFaces



### web.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="http://java.sun.com/xml/ns/javaee"
xmlns:web="http://java.sun.com/xml/ns/javaee/web-app_2_5.xsd"
xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
http://java.sun.com/xml/ns/javaee/web-app_2_5.xsd" version="2.5">

  <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>
```

```
<context-param>
  <param-name>javax.faces.STATE_SAVING_METHOD</param-name>
  <param-value>client</param-value>
</context-param>
<context-param>
  <param-name>javax.servlet.jsp.jstl.fmt.localizationContext</param-name>
  <param-value>resources.application</param-value>
</context-param>
<listener>
  <listener-class>com.sun.faces.config.ConfigureListener</listener-class>
</listener>
<session-config>
  <session-timeout>30</session-timeout>
</session-config>
<welcome-file-list>
  <welcome-file>index.jsp</welcome-file>
</welcome-file-list>
</web-app>
```

---

- **Arquivo CSS**

```
body
{
    font-family: verdana;
    font-size: 9pt;
    padding: 30px;
    background-color: #CCC;
}

#topo
{
    width: 0 auto;
    height: 0 auto;
    border: 1px solid #000;
    background-color: #FAFAFA;
}

#menu
{
    width: 0 auto;
    height: 0 auto;
    border: 1px solid #000;
    background-color: #FAFAFA;
}

#principal
{
    width: 0 auto;
    height: 0 auto;
    border: 1px solid #000;
    background-color: #FFF;
}
```

## topo.xhtml

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<!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:f="http://java.sun.com/jsf/core"
      xmlns:p="http://primefaces.org/ui"
      xmlns:ui="http://java.sun.com/jsf/facelets">
  <h:head>
    <meta http-equiv="Content-Type"
          content="text/html; charset=ISO-8859-1" />
    <title></title>
  </h:head>
  <h:body>

    <!-- Área destinada ao topo da página -->

    <ui:composition>

      <p:panel header="COTI Informática - Projeto Jsf Facelets"
              style="background-color: #FAFAFA;">
        <p:graphicImage value="/templates/imagens/logo.jpg"/>
      </p:panel>

    </ui:composition>

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

## menu.xhtml

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<!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:f="http://java.sun.com/jsf/core"
      xmlns:p="http://primefaces.org/ui"
      xmlns:ui="http://java.sun.com/jsf/facelets">
  <h:head>
    <meta http-equiv="Content-Type"
          content="text/html; charset=ISO-8859-1" />
    <title></title>
  </h:head>
  <h:body>

    <!-- Área destinada ao menu da página -->

    <ui:composition>

      <h:form id="menu">

        <p:toolbar>
```

```
<p:toolbarGroup>

    <p:commandButton
        value="Cadastrar Usuário"/>

    <p:menuButton value="Consultar">
        <p:menuitem value="Produtos"/>
        <p:menuitem value="Pedidos"/>
        <p:menuitem value="Clientes"/>
        <p:menuitem value="Estoque"/>
    </p:menuButton>

</p:toolbarGroup>

</p:toolbar>

</h:form>

</ui:composition>

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

### Criando uma página do sistema baseado no layout

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<!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:f="http://java.sun.com/jsf/core"
    xmlns:p="http://primefaces.org/ui"
    xmlns:ui="http://java.sun.com/jsf/facelets">
    <h:head>
    <meta http-equiv="Content-Type"
        content="text/html; charset=ISO-8859-1" />
    <title>COTI Informática</title>

    <link rel="stylesheet"
        href="templates/styles/default.css"
        type="text/css"/>

    </h:head>
    <h:body>

        <div id="topo">
            <!-- Importar o conteudo da página topo.xhtml -->
            <ui:insert name="topo">
                <ui:include src="/templates/topo.xhtml"/>
            </ui:insert>
        </div>

        <div id="menu">
            <!-- Importar o conteudo da página menu.xhtml -->
            <ui:insert name="menu">
                <ui:include src="/templates/menu.xhtml"/>
            </ui:insert>
        </div>
    </h:body>
</html>
```

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

<div id="principal">

    <p:spacer height="300"/>

</div>

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

---

## Criando a página contendo a Janela popup

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<!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:f="http://java.sun.com/jsf/core"
      xmlns:p="http://primefaces.org/ui"
      xmlns:ui="http://java.sun.com/jsf/facelets">
<h:head>

<meta http-equiv="Content-Type"
      content="text/html; charset=ISO-8859-1" />

<title></title>
</h:head>

<h:body>

    <ui:composition>

        <p:dialog widgetVar="janela"
                  header="Pesquisa de Produtos"
                  modal="true">

            <h:form>

                <h:outputText value="Relação de Produtos cadastrados"/>

            </h:form>

        </p:dialog>

    </ui:composition>

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

---

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<!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:f="http://java.sun.com/jsf/core"
      xmlns:p="http://primefaces.org/ui"
      xmlns:ui="http://java.sun.com/jsf/facelets">

<h:head>

<meta http-equiv="Content-Type"
      content="text/html; charset=ISO-8859-1" />

<title></title>

</h:head>

<h:body>

    <ui:composition>

        <p:dialog widgetVar="janela"
                  header="Pesquisa de Produtos"
                  modal="true">

            <h:form>

                <h:outputText
                    value="Relação de Produtos cadastrados"/>

                <br/><br/>
                <p:commandButton value="Excluir"
                                onclick="janela2.show()" />

            </h:form>

        </p:dialog>

        <p:dialog widgetVar="janela2"
                  header="Confirmação"
                  modal="true">

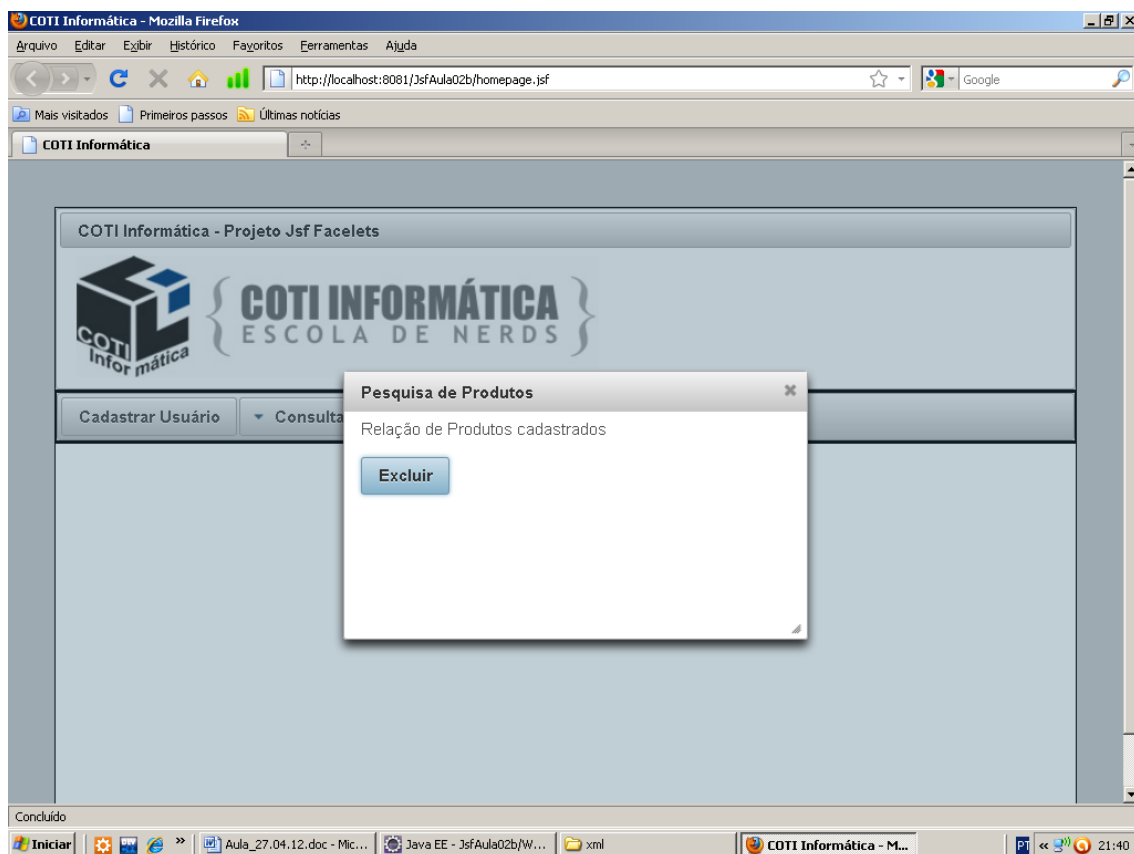
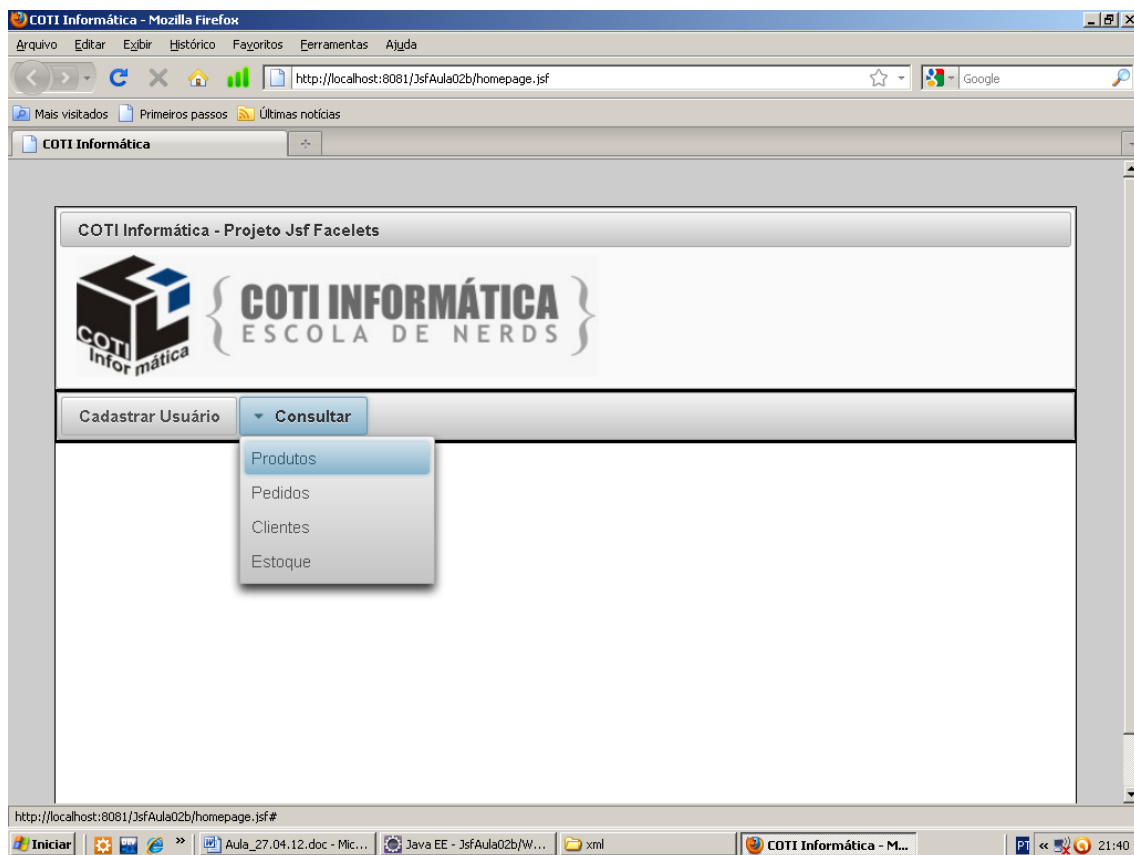
            <h:form>
                <p:commandButton value="Confirmar"/>
                <p:commandButton value="Cancelar"
                                onclick="janela2.hide()" />
            </h:form>

        </p:dialog>

    </ui:composition>

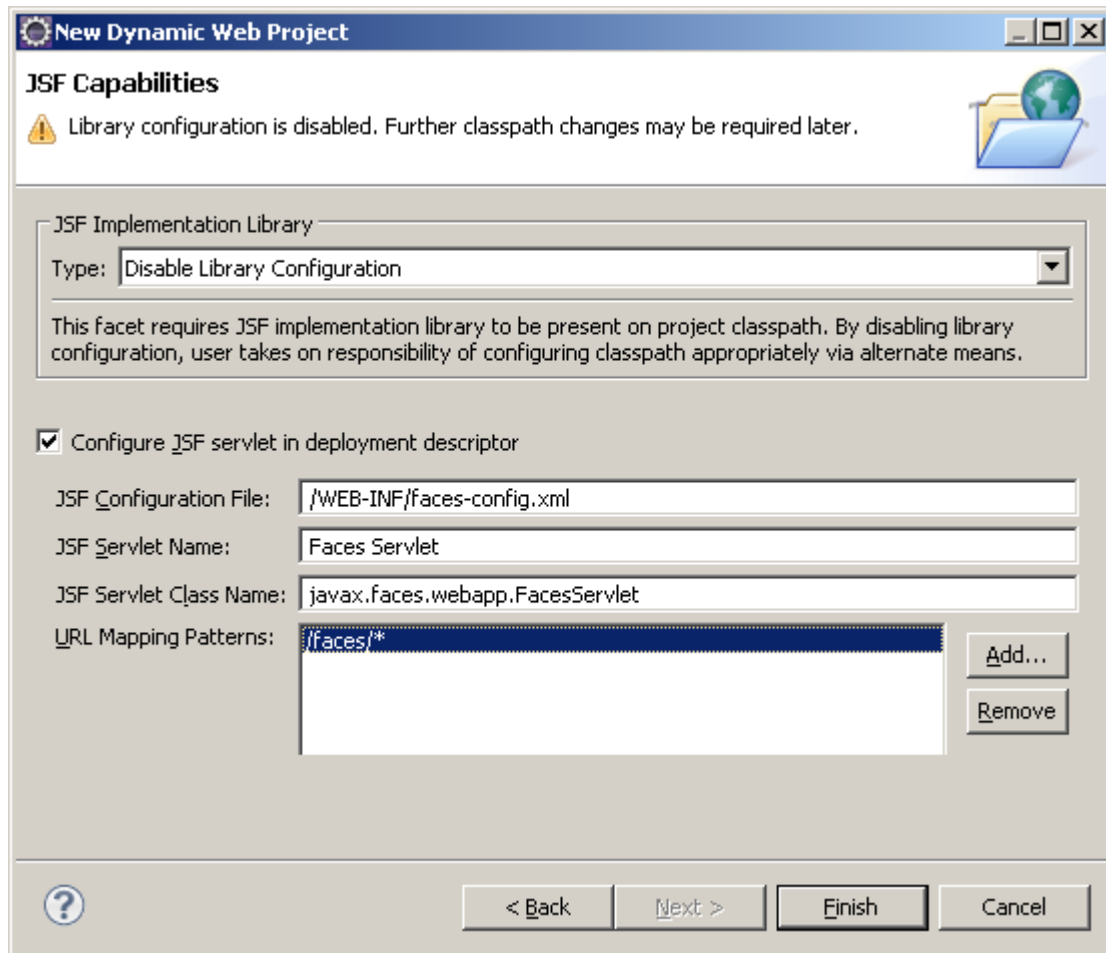
</h:body>
</html>
```





## Parte 8: Upload de Imagens e Carrossel de imagens

- Criando o Projeto



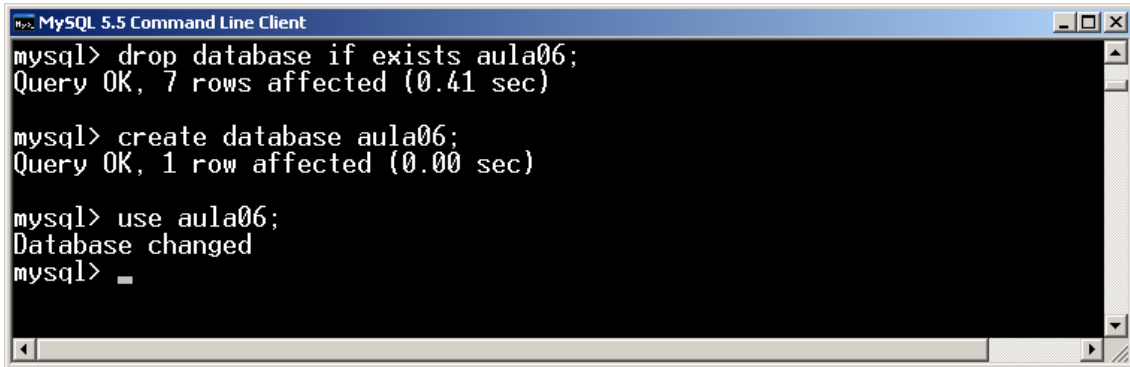
## Configurando o Hibernate

config/hibernate.cfg.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE hibernate-configuration PUBLIC "-//Hibernate/Hibernate Configuration DTD
3.0//EN" "http://hibernate.sourceforge.net/hibernate-configuration-3.0.dtd">
<hibernate-configuration>
  <session-factory>
    <property name="hibernate.dialect">
      org.hibernate.dialect.MySQLDialect</property>
    <property name="hibernate.connection.driver_class">
      com.mysql.jdbc.Driver</property>
    <property name="hibernate.connection.url">
      jdbc:mysql://localhost:3306/aula06</property>
    <property name="hibernate.connection.username">root</property>
    <property name="hibernate.connection.password">coti</property>
  </session-factory>
</hibernate-configuration>
```

```
<property name="hibernate.show_sql">true</property>
<property name="hibernate.format_sql">true</property>

<mapping class="entity.Produto"/>
</session-factory>
</hibernate-configuration>
```



```
mysql> drop database if exists aula06;
Query OK, 7 rows affected (0.41 sec)

mysql> create database aula06;
Query OK, 1 row affected (0.00 sec)

mysql> use aula06;
Database changed
mysql> _
```

```
package entity;

import java.util.Date;
import javax.persistence.*;

@Entity
@Table
public class Produto {

    @Id
    @GeneratedValue(strategy=GenerationType.AUTO)
    @Column
    private Integer idProduto;

    @Column
    private String nome;

    @Temporal(TemporalType.DATE)
    @Column
    private Date dataCadastro;

    @Column
    private String foto;

    public Produto() {

    }

    public Integer getIdProduto() {
        return idProduto;
    }

    public void setIdProduto(Integer idProduto) {
        this.idProduto = idProduto;
    }

    public String getNome() {
        return nome;
    }

}
```

```
public void setNome(String nome) {
    this.nome = nome;
}

public Date getDataCadastro() {
    return dataCadastro;
}

public void setDataCadastro(Date dataCadastro) {
    this.dataCadastro = dataCadastro;
}

public String getFoto() {
    return foto;
}

public void setFoto(String foto) {
    this.foto = foto;
}
}
```

### HibernateUtil.java

```
package persistence;

import org.hibernate.SessionFactory;
import org.hibernate.cfg.AnnotationConfiguration;
import org.hibernate.cfg.Configuration;
import org.hibernate.tool.hbm2ddl.SchemaExport;

public class HibernateUtil {
    private static final SessionFactory sessionFactory;

    static {
        try {

            sessionFactory = new
                AnnotationConfiguration().configure(

                    "config/mysql_hibernate.cfg.xml").buildSessionFactory();
        } catch (Throwable ex) {

            System.err.println("Initial SessionFactory creation
                failed." + ex);
            throw new ExceptionInInitializerError(ex);
        }
    }

    public static SessionFactory getSessionFactory() {
        return sessionFactory;
    }

    //Método 'Main' para exportar o script das entidades para o
    banco de dados
    public static void main(String[] args) {

        Configuration cfg = new AnnotationConfiguration();
```

```
        cfg.configure("config/mysql_hibernate.cfg.xml");

        new SchemaExport(cfg).create(true, true);
    }
}

drop table if exists Produto

create table Produto (
    idProduto integer not null auto_increment,
    dataCadastro date,
    foto varchar(255),
    nome varchar(255),
    primary key (idProduto)
)

package persistence;

import java.util.List;

import org.hibernate.Query;
import org.hibernate.Session;
import org.hibernate.Transaction;

import entity.Produto;

public class ProdutoDao {

    Session session;
    Transaction transaction;
    Query query;

    public void create(Produto p) throws Exception{

        //Obter uma conexão -> HibernateUtil
        session = HibernateUtil.getSessionFactory().openSession();
        //Iniciar uma transação
        transaction = session.beginTransaction();
        //Cadastrar o Produto
        session.save(p);
        //Commit
        transaction.commit();
    }

    public List<Produto> findAll() throws Exception{

        session = HibernateUtil.getSessionFactory().openSession();
        query = session.createQuery("select p from Produto as p");
        return query.list();
    }
}
```

### Tomahawk

Componente para o Jsf que permite a realização de recursos e eventos diversificados, neste caso, usará para realizar o Upload de imagens.

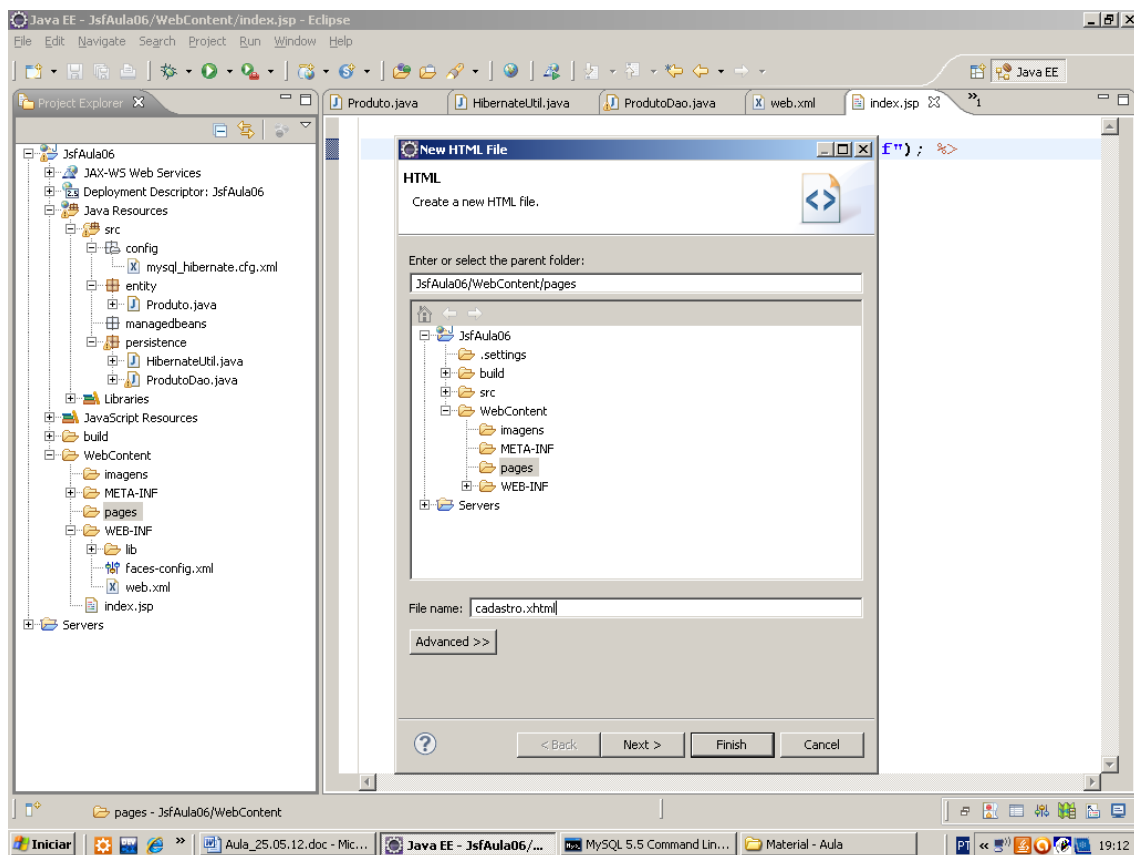
#### web.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns="http://java.sun.com/xml/ns/javaee"
  xmlns:web="http://java.sun.com/xml/ns/javaee/web-app_2_5.xsd"
  xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
    http://java.sun.com/xml/ns/javaee/web-app_2_5.xsd"
  version="2.5">
  <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>
  <context-param>
    <param-name>javax.faces.STATE_SAVING_METHOD</param-name>
    <param-value>client</param-value>
  </context-param>
  <context-param>
    <param-
name>javax.servlet.jsp.jstl.fmt.localizationContext</param-name>
    <param-value>resources.application</param-value>
  </context-param>
  <listener>
    <listener-
class>com.sun.faces.config.ConfigureListener</listener-class>
  </listener>

  <filter>
    <filter-name>Extensions Filter</filter-name>
    <filter-class>
      org.apache.myfaces.webapp.filter.ExtensionsFilter
    </filter-class>
    <init-param>
      <param-name>uploadMaxFileSize</param-name>
      <param-value>100m</param-value>
    </init-param>
    <init-param>
      <param-name>uploadThresholdSize</param-name>
      <param-value>100k</param-value>
    </init-param>
    <init-param>
      <param-name>uploadRepositoryPath</param-name>
      <param-value>c:/temp</param-value>
    </init-param>
  </filter>
  <filter-mapping>
    <filter-name>Extensions Filter</filter-name>
```

```
<servlet-name>Faces Servlet</servlet-name>
</filter-mapping>
<session-config>
    <session-timeout>30</session-timeout>
</session-config>
<welcome-file-list>
    <welcome-file>index.jsp</welcome-file>
</welcome-file-list>
```

```
</web-app>
```



Para que formulários HTML suportem campos de upload de arquivos é necessário que estes formulários possuam um atributo denominado **enctype** com o valor **multipart/form-data**

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<!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:f="http://java.sun.com/jsf/core"
    xmlns:p="http://primefaces.org/ui"
    xmlns:t="http://myfaces.apache.org/tomahawk">
<h:head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1" />
<title>Insert title here</title>
</h:head>
```

```
<h:body>

    <p:panel header="Cadastro de Produtos">

        <h:form id="formcadastro"
            enctype="multipart/form-data">

            <h:panelGrid columns="2">

                <h:outputLabel value="Nome do Produto:"
                    for="nome"/>
                <p:inputText id="nome"/>

                <h:outputLabel value="Data de Cadastro:"
                    for="data"/>
                <p:calendar id="data"/>

                <h:outputLabel value="Foto:" for="foto"/>
                <t:inputFileUpload id="foto"/>

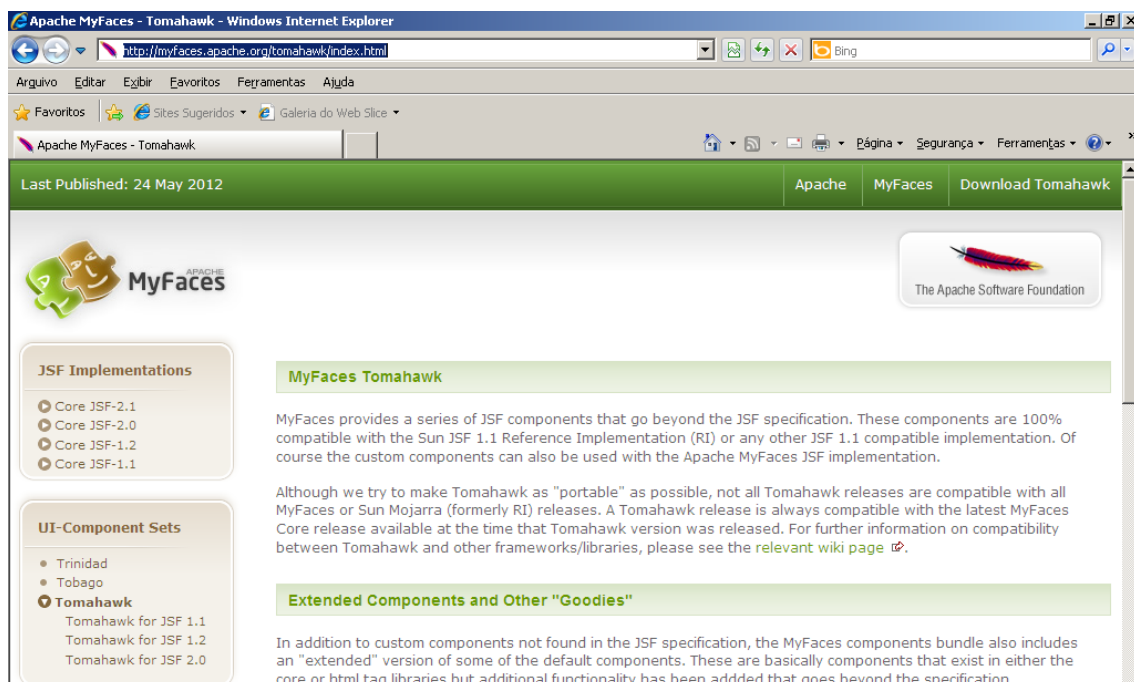
            </h:panelGrid>

            <p:commandButton value="Cadastrar Produto"
                ajax="false"/>

            <p:messages/>

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

<http://myfaces.apache.org/tomahawk/index.html>



Apache MyFaces - Tomahawk - Windows Internet Explorer

http://myfaces.apache.org/tomahawk/index.html

Apache MyFaces Tomahawk

Last Published: 24 May 2012

Apache MyFaces Download Tomahawk

**JSF Implementations**

- Core JSF-2.1
- Core JSF-2.0
- Core JSF-1.2
- Core JSF-1.1

**UI-Component Sets**

- Trinidad
- Tobago
- Tomahawk**
  - Tomahawk for JSF 1.1
  - Tomahawk for JSF 1.2
  - Tomahawk for JSF 2.0

**MyFaces Tomahawk**

MyFaces provides a series of JSF components that go beyond the JSF specification. These components are 100% compatible with the Sun JSF 1.1 Reference Implementation (RI) or any other JSF 1.1 compatible implementation. Of course the custom components can also be used with the Apache MyFaces JSF implementation.

Although we try to make Tomahawk as "portable" as possible, not all Tomahawk releases are compatible with all MyFaces or Sun Mojarra (formerly RI) releases. A Tomahawk release is always compatible with the latest MyFaces Core release available at the time that Tomahawk version was released. For further information on compatibility between Tomahawk and other frameworks/libraries, please see the [relevant wiki page](#).

**Extended Components and Other "Goodies"**

In addition to custom components not found in the JSF specification, the MyFaces components bundle also includes an "extended" version of some of the default components. These are basically components that exist in either the core or html tag libraries but additional functionality has been added that goes beyond the specification.



### ManagedBean

```
package managedbeans;

import javax.faces.bean.ManagedBean;
import javax.faces.bean.RequestScoped;
import org.apache.myfaces.custom.fileupload.UploadedFile;
import entity.Produto;

@ManagedBean(name="mbproduto")
@RequestScoped
public class ManagedBeanProduto {

    //Atributo -> Entidade
    private Produto produto;

    //Atributo para resgatar o arquivo enviado no campo de
    Upload
    private UploadedFile foto;

    //Construtor default
    public ManagedBeanProduto() {
        produto = new Produto();
    }

    //Método para cadastrar Produto
    public void cadastrar(){

    }

    public Produto getProduto() {
        return produto;
    }

    public void setProduto(Produto produto) {
        this.produto = produto;
    }

    public UploadedFile getFoto() {
        return foto;
    }

    public void setFoto(UploadedFile foto) {
        this.foto = foto;
    }
}
```

---

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<!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:f="http://java.sun.com/jsf/core"
xmlns:p="http://primefaces.org/ui"
xmlns:t="http://myfaces.apache.org/tomahawk">
<h:head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1"
/>
<title>Insert title here</title>
</h:head>

<h:body>

    <p:panel header="Cadastro de Produtos">

        <h:form id="formcadastro"
            enctype="multipart/form-data">

            <h:panelGrid columns="2">

                <h:outputLabel value="Nome do Produto:"
                    for="nome"/>
                <p:inputText id="nome"
                    value="#{mbproduto.produto.nome}"/>

                <h:outputLabel value="Data de Cadastro:"
                    for="data"/>
                <p:calendar id="data"
                    value="#{mbproduto.produto.dataCadastro}"/>

                <h:outputLabel value="Foto:" for="foto"/>
                <t:inputFileUpload id="foto"
                    value="#{mbproduto.foto}"/>

            </h:panelGrid>

            <p:commandButton value="Cadastrar Produto"
                ajax="false"
                action="#{mbproduto.cadastrar}"/>

            <p:messages/>

        </h:form>
    </p:panel>

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

---

```
package util;
```

```
import java.io.File;
import java.io.FileOutputStream;
import java.io.OutputStream;
```



## Jsf2 e Primefaces – Apostila de Componentes

COTI Informática - Av. Rio Branco, 185 - Sala 904 - Rio de Janeiro – RJ  
Tel. 21-2262-9043 [www.cotiinformatica.com.br](http://www.cotiinformatica.com.br)

```
import java.text.SimpleDateFormat;
import java.util.Date;

import org.apache.commons.io.IOUtils;
import org.apache.myfaces.custom.fileupload.UploadedFile;

public class Upload {

    private static File file;
    private static OutputStream out;

    //Método para prover o Upload do Arquivo
    public static String SalvarFoto(UploadedFile foto) throws
        Exception{

        //Definir o Nome do Arquivo
        String nomeArquivo = new
        SimpleDateFormat("yyyyMMddHHmmss").
            format(new Date()) + ".jpg";

        //Definir o caminho do arquivo
        String path = "C:\\Documents and Settings\\Coti
        Informatica\\Desktop\\workspace\\JsfAula\\
        WebContent\\imagens\\";

        //Salvar o Arquivo
        file = new File(path + nomeArquivo);

        out = new FileOutputStream(file);
        IOUtils.copy(foto.getInputStream(), out);

        return nomeArquivo;
    }
}
```

---

```
package managedbeans;

import javax.faces.application.FacesMessage;
import javax.faces.bean.ManagedBean;
import javax.faces.bean.RequestScoped;
import javax.faces.context.FacesContext;
import org.apache.myfaces.custom.fileupload.UploadedFile;
import persistence.ProdutoDao;
import util.Upload;
import entity.Produto;

@ManagedBean(name="mbproduto")
@RequestScoped
public class ManagedBeanProduto {

    //Atributo -> Entidade
    private Produto produto;

    //Atributo para resgatar o arquivo enviado no campo de Upload
    private UploadedFile foto;

    //Construtor default
```

```
public ManagedBeanProduto() {
    produto = new Produto();
}

//Método para cadastrar Produto
public void cadastrar() {

    FacesContext context =
    FacesContext.getCurrentInstance();
    FacesMessage message = null;

    try {

        produto.setFoto(Upload.SalvarFoto(foto));

        ProdutoDao d = new ProdutoDao();
        d.create(produto);

        produto = new Produto();
        message = new FacesMessage("Produto cadastrado
                                   com sucesso.");

    }
    catch(Exception e) {

        message = new FacesMessage("Erro: " +
                                   e.getMessage());

        message.setSeverity(FacesMessage.SEVERITY_ERROR);

    }

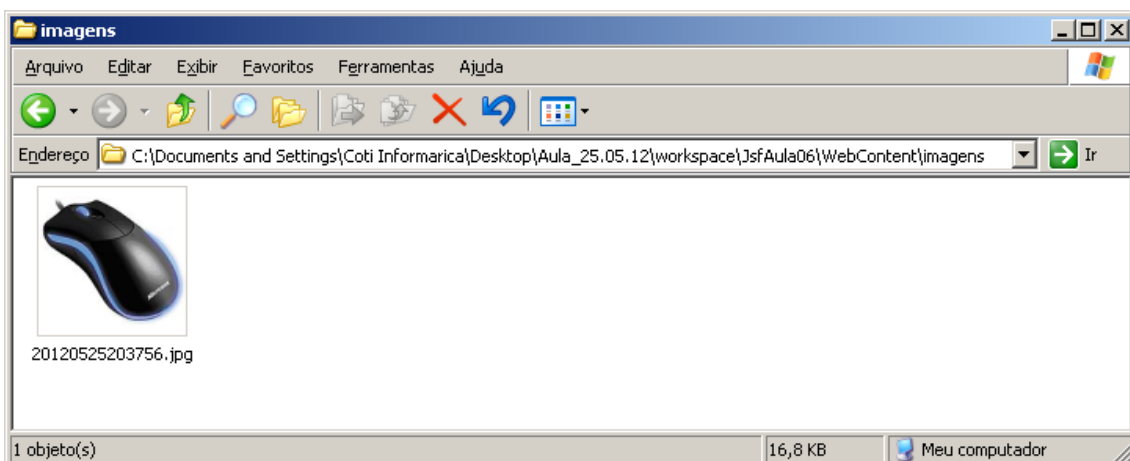
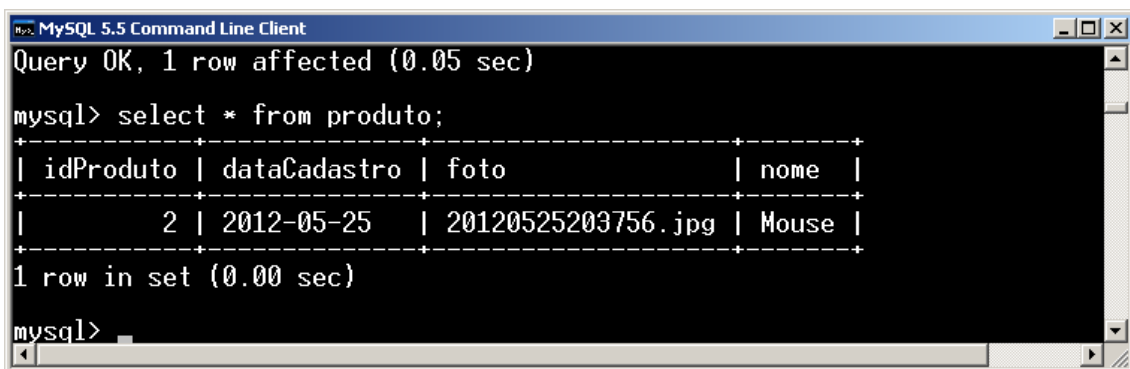
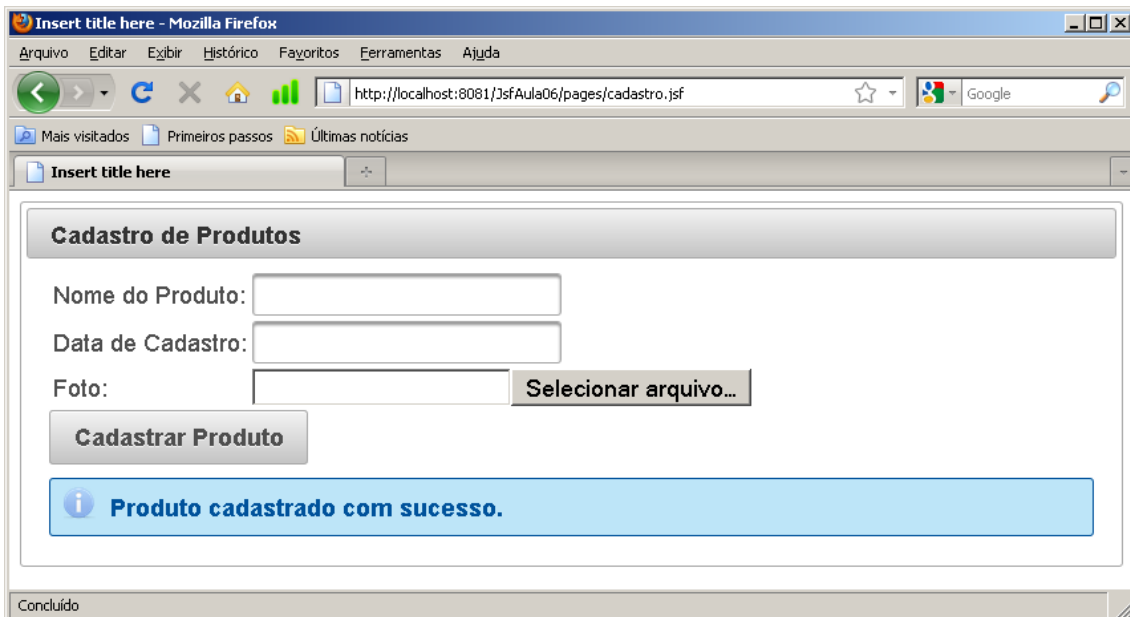
    context.addMessage("formcadastro", message);
}

public Produto getProduto() {
    return produto;
}

public void setProduto(Produto produto) {
    this.produto = produto;
}

public UploadedFile getFoto() {
    return foto;
}

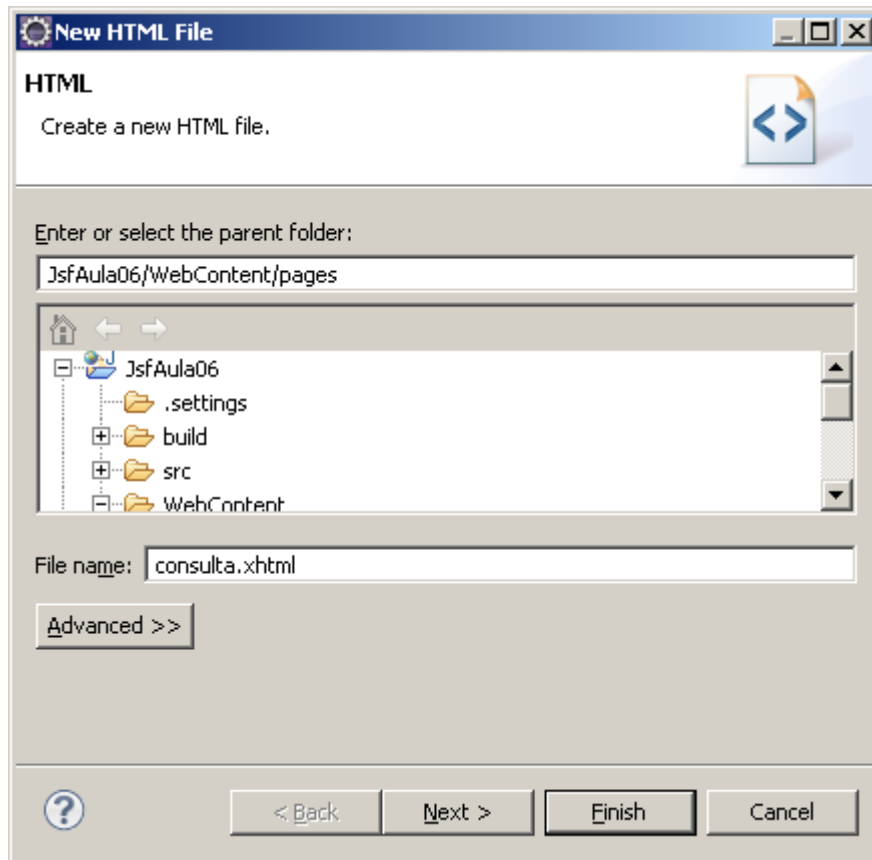
public void setFoto(UploadedFile foto) {
    this.foto = foto;
}
}
```



### Hibernate:

```

insert
into
    Produto
    (dataCadastro, foto, nome)
values
    (?, ?, ?)
  
```



```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<!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:f="http://java.sun.com/jsf/core"
      xmlns:p="http://primefaces.org/ui">
<h:head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1"
/>
<title>Insert title here</title>
</h:head>
<h:body>

    <p:panel header="Galeria de Produtos">

        <h:form id="formconsulta">

            <!-- Carrossel de Imagens -->
            <p:ring>
            </p:ring>

            <p:spacer height="60"/>

            <p>
                <p:commandLink value="Cadastrar Produto">
```

```
                action="cadastro.jsf"
                ajax="false"/>
            </p>
        </h:form>
    </p:panel>
</h:body>
</html>
```

### ManagedBean

```
package managedbeans;

import java.util.List;
import javax.faces.application.FacesMessage;
import javax.faces.bean.ManagedBean;
import javax.faces.bean.RequestScoped;
import javax.faces.context.FacesContext;
import org.apache.myfaces.custom.fileupload.UploadedFile;
import persistence.ProdutoDao;
import util.Upload;
import entity.Produto;

@ManagedBean(name="mbproduto")
@RequestScoped
public class ManagedBeanProduto {

    //Atributo -> Entidade
    private Produto produto;

    //Atributo para resgatar o arquivo enviado no campo de
    Upload
    private UploadedFile foto;

    //Atributo para armazenar uma lista de produtos
    private List<Produto> listagem;

    //Construtor default
    public ManagedBeanProduto() {
        produto = new Produto();
    }

    //Método para cadastrar Produto
    public void cadastrar() {

        FacesContext context =
            FacesContext.getCurrentInstance();
        FacesMessage message = null;

        try {

            produto.setFoto(Upload.SalvarFoto(foto));

            ProdutoDao d = new ProdutoDao();
```

```
d.create(produto);

produto = new Produto();
message = new FacesMessage("Produto cadastrado
                             com sucesso.");
}
catch(Exception e) {

    message = new FacesMessage("Erro: " +
                                e.getMessage());

    message.setSeverity(FacesMessage.SEVERITY_ERROR);
}

context.addMessage("formcadastro", message);
}

public Produto getProduto() {
    return produto;
}

public void setProduto(Produto produto) {
    this.produto = produto;
}

public UploadedFile getFoto() {
    return foto;
}

public void setFoto(UploadedFile foto) {
    this.foto = foto;
}

public List<Produto> getListagem() {

    try{

        ProdutoDao d = new ProdutoDao();
        listagem = d.findAll();

    }
    catch(Exception e){
        e.printStackTrace();
    }

    return listagem;
}

public void setListagem(List<Produto> listagem) {
    this.listagem = listagem;
}

}
```



```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<!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:f="http://java.sun.com/jsf/core"
      xmlns:p="http://primefaces.org/ui">
<h:head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1" />
<title>Insert title here</title>
</h:head>
<h:body>

    <p:panel header="Galeria de Produtos">

        <h:form id="formconsulta">

            <!-- Carrossel de Imagens -->
            <p:ring value="#{mbproduto.listagem}" var="p">

                <p:column>

                    <center>

                        <p:graphicImage
                            value="/imagens/#{p.foto}"
                            height="140" width="160"/>

                        <br/>

                        <p:commandButton value="Detalhes"
                            onclick="painel.show()"
                            actionListener="#{mbproduto.detalhes}">

                            <f:param name="produto"
                                id="produto" value="#{p}" />

                        </p:commandButton>

                    </center>

                </p:column>

            </p:ring>

            <p:spacer height="60"/>

            <p>
                <p:commandLink value="Cadastrar Produto"
                    action="cadastro.jsf"
                    ajax="false"/>
            </p>

        </h:form>

    </p:panel>
```

```
<!-- Janela modal -->
<p:dialog widgetVar="painel" header="Detalhes do Produto"
          modal="true">

</p:dialog>

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

---

```
package managedbeans;

import java.util.List;

import javax.faces.application.FacesMessage;
import javax.faces.bean.ManagedBean;
import javax.faces.bean.RequestScoped;
import javax.faces.component.UIParameter;
import javax.faces.context.FacesContext;
import javax.faces.event.ActionEvent;

import org.apache.myfaces.custom.fileupload.UploadedFile;

import persistence.ProdutoDao;
import util.Upload;
import entity.Produto;

@ManagedBean(name="mbproduto")
@RequestScoped
public class ManagedBeanProduto {

    //Atributo -> Entidade (Cadastro)
    private Produto produto;

    //Atributo para resgatar o arquivo enviado no campo de
    Upload
    private UploadedFile foto;

    //Atributo para armazenar uma lista de produtos
    private List<Produto> listagem;

    //Atributo -> Entidade (Detalhes)
    private Produto itemproduto;

    //Construtor default
    public ManagedBeanProduto() {
        produto = new Produto();
        itemproduto = new Produto();
    }

    //Método para cadastrar Produto
    public void cadastrar() {
```

```
FacesContext context =
FacesContext.getCurrentInstance();
FacesMessage message = null;

try {

    produto.setFoto(Upload.SalvarFoto(foto));

    ProdutoDao d = new ProdutoDao();
    d.create(produto);

    produto = new Produto();
    message = new FacesMessage("Produto cadastrado
                                com sucesso.");

}
catch(Exception e) {

    message = new FacesMessage("Erro: " +
                                e.getMessage());

    message.setSeverity(FacesMessage.SEVERITY_ERROR);

}

context.addMessage("formcadastro", message);

}

//Método para exibir um produto selecionado pelo componente ring
//Executado a partir de um ActionListener, isto significa que o
//Método recebe parâmetros da página (ActionEvent)
public void detalhes(ActionEvent event){

    try{

        //Resgatar o parâmetro (name='produto') enviado
        //pelo evento ActionListener
        UIParameter parametro = (UIParameter)
        event.getComponent().findComponent("produto");

        //Extrair o valor contido no parametro (Classe
        //Produto)
        itemproduto = (Produto) parametro.getValue();

    }
    catch(Exception e){
        e.printStackTrace();
    }

}
```

```
public Produto getProduto() {
    return produto;
}

public void setProduto(Produto produto) {
    this.produto = produto;
}

public UploadedFile getFoto() {
    return foto;
}

public void setFoto(UploadedFile foto) {
    this.foto = foto;
}

public List<Produto> getListagem() {

    try{

        ProdutoDao d = new ProdutoDao();
        listagem = d.findAll();

    }
    catch(Exception e){
        e.printStackTrace();
    }

    return listagem;
}

public void setListagem(List<Produto> listagem) {
    this.listagem = listagem;
}

public Produto getItemproduto() {
    return itemproduto;
}

public void setItemproduto(Produto itemproduto) {
    this.itemproduto = itemproduto;
}

}
```

---

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<!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:f="http://java.sun.com/jsf/core"
      xmlns:p="http://primefaces.org/ui">
```

```
<h:head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1" />
<title>Insert title here</title>
</h:head>
<h:body>

    <p:panel header="Galeria de Produtos">

        <h:form id="formconsulta">

            <!-- Carrossel de Imagens -->
            <p:ring value="#{mbproduto.listagem}" var="p">

                <p:column>

                    <center>
                        <p:graphicImage
                            value="/imagens/#{p.foto}"
                            height="140" width="160"/>

                        <br/>

                        <p:commandButton
                            value="Detalhes" onclick="painel.show()"
                            actionListener="#{mbproduto.detalhes}"
                            update="formdetalhes">

                                <f:param name="produto" id="produto"
                                    value="#{p}" />

                                </p:commandButton>

                            </center>

                        </p:column>

                    </p:ring>

                    <p:spacer height="60"/>

                    <p>
                        <p:commandLink value="Cadastrar Produto"
                            action="cadastro.jsf"
                            ajax="false"/>
                    </p>

                </h:form>

            </p:panel>

            <!-- Janela modal -->

            <p:dialog widgetVar="painel" header="Detalhes do Produto"
                modal="true">

                <h:form id="formdetalhes">

                    <center>
```

```

        <!-- Exibir a Imagem -->
        <p:graphicImage
            value="/imagens/#{mbproduto.itemproduto.foto}"
            height="180" width="200"/>

        <br/><br/>

        <strong>
            <h:outputText
                value="#{mbproduto.itemproduto.nome}"
                style="font-size: 16pt;"/>
        </strong>

        <br/><br/>

        Cadastrado em: <br/>

        <h:outputText
            value="#{mbproduto.itemproduto.dataCadastro}">

            <f:convertDateTime pattern="EEEE dd
                'de' MMMM 'de' yyyy"/>
        </h:outputText>
        <p>
            <p:commandButton value="Fechar
                Janela" onclick="painel.hide()"/>
        </p>
    </center>
</h:form>

</p:dialog>

</h:body>
</html>

```

