

Java Enterprise Edition - JEE

06. Java Server Pages - JSP



Esp. Márcio Palheta
gtalk: marcio.palheta@gmail.com



Agenda

- O que é JSP;
- Suas vantagens e desvantagens;
- Escrever arquivos JSP com scriptlets;
- Usar Expression Language;
- O que são taglibs;
- Exercício – Agenda de compromissos;



O que é uma página JSP?

- É uma página HTML comum que pode conter código JAVA;
- Deve possuir a extensão `.jsp`
- O código JAVA deve ser incluso entre tags especiais: `<% código java aqui %>`
- Exemplo: Declaração de uma variável
 - `<%String msg = "Bem vindo!"%>`
- O código entre `<%..%>` é um scriptlet



A variável implícita **out**

- Páginas JSP possuem variáveis implícitas;
- A variável **out** (do tipo `JspWriter`) permite a impressão de objetos através do método **println**:
 - `<% out.println(nome); %>
`
 - OU: `<%=nome %>
`



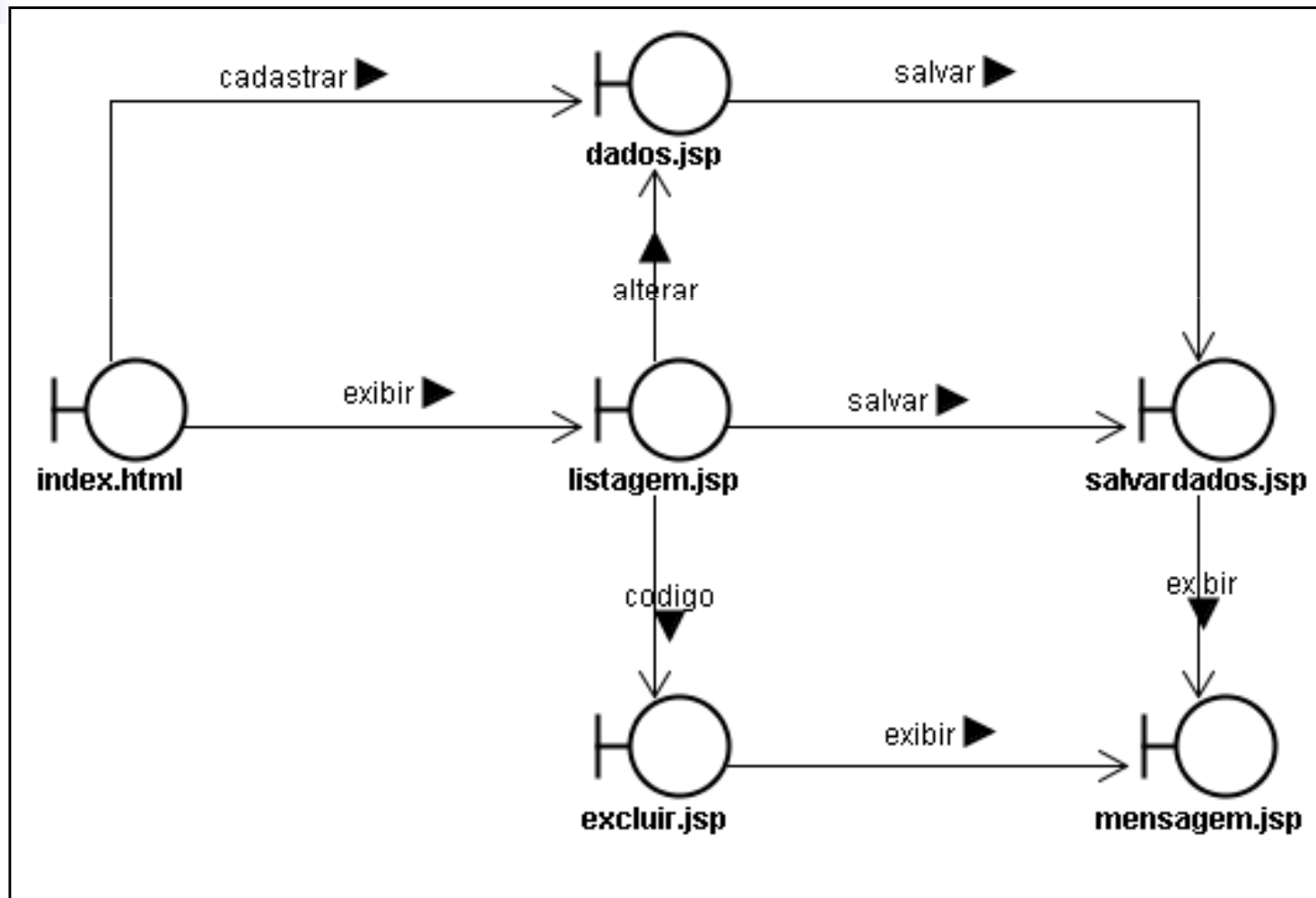
Primeira página JSP

- Crie a página bemvindo.jsp e inclua o trecho de código a seguir:

```
10<body>
11    <%-- Comentarios da JSP aqui --%>
12    <%
13        String mensagem = "Bem vindo!";
14    %>
15    Temos duas maneiras diferentes de escrever em uma página:<br><br>
16    Primeira:<br>
17    <% out.println(mensagem) ; %><br>
18    Segunda:<br>
19    <%=mensagem %><br>
20    Tudo foi <%= "Executado" %>
21</body>
```

- <http://localhost:8080/jspteste/bemvindo.jsp>

Estudo de caso: Cadastro de empresas



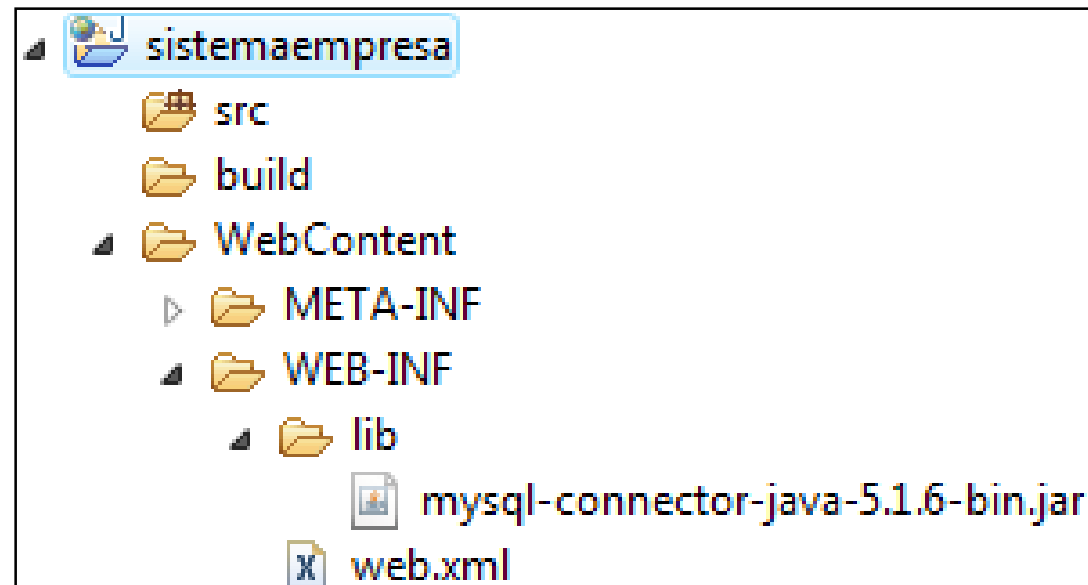


Exercício 01

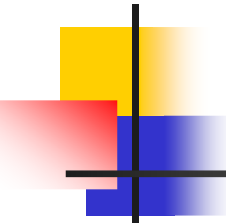
- 01.Criar o projeto WEB: [sistemaempresa](#)
- 02.Criar uma ConnectionFactory;
- 03.Criar o bean Empresa;
- 04.Criar a classe EmpresaDAO;
- 05.Criar métodos para cadastrar, alterar, listar, consultar e excluir;
- 06.Criar a página index.html
- Acessar:
 - <http://localhost:8080/sistemaempresa/>

01. Projeto web: sistemaempresa

- Crie um novo Dynamic Web Project;
- Copie o driver mysql para a pasta ...\\WebContent\\WEB-INF\\lib

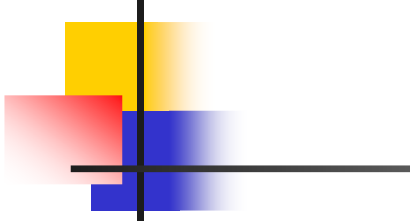


02. ConnectionFactory



```
1 package br.fucapi.curso.jee.model.connection;
2 import java.sql.Connection;
3 import java.sql.DriverManager;
4 import java.sql.SQLException;
5 public class ConnectionFactory {
6     public static Connection getConnection() {
7         Connection connection = null;
8         try {
9             //Definicao do driver JDBC
10             Class.forName("com.mysql.jdbc.Driver");
11             //Criacao da conexao
12             connection = DriverManager.getConnection(
13                 "jdbc:mysql://localhost/cursojee"
14                 "root", "");
15         } catch (ClassNotFoundException e) {
16             e.printStackTrace();
17         } catch (SQLException e) {
18             e.printStackTrace();
19         }
20         //Retorno da conexao gerada
21         return connection;
22     }
23 }
```

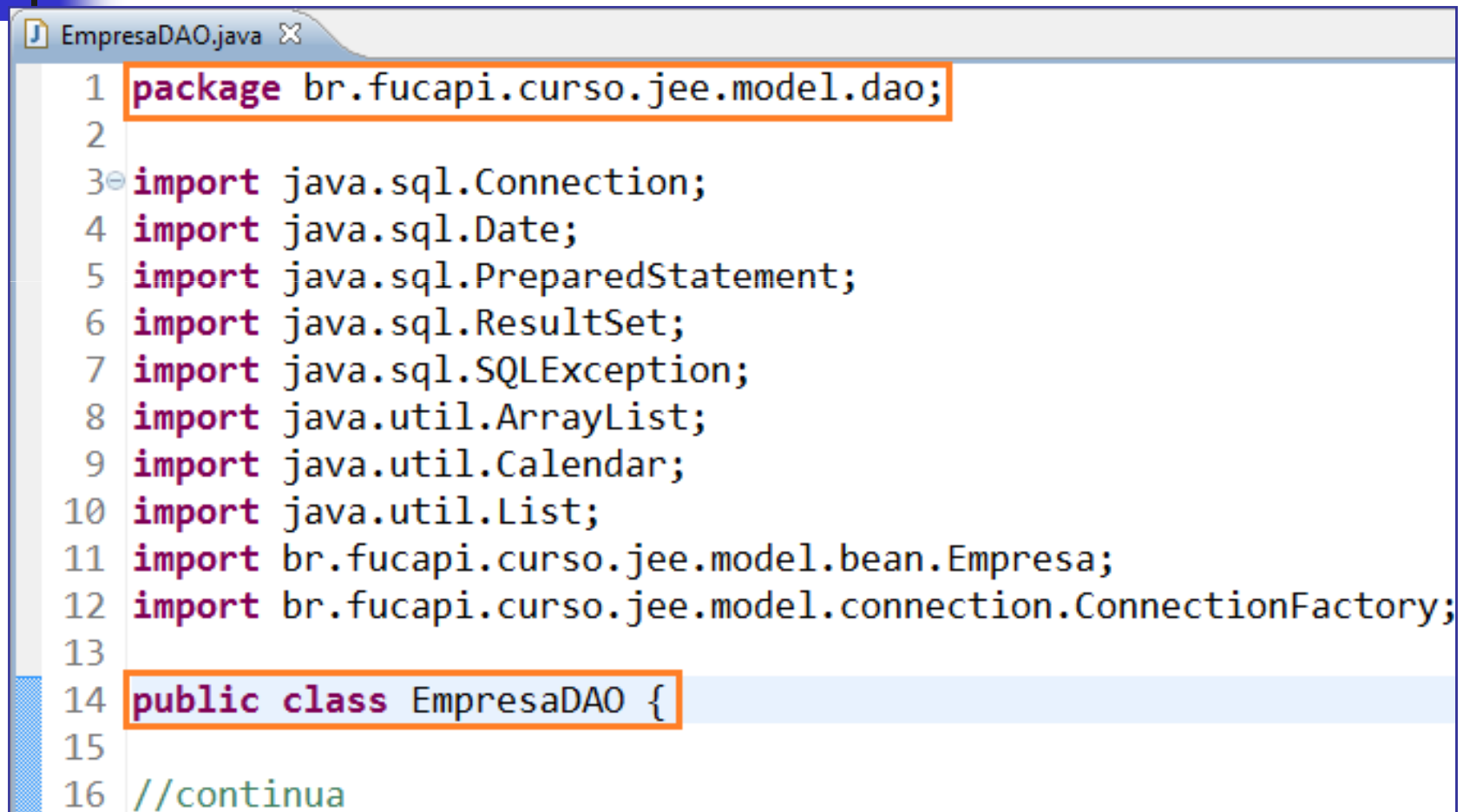
03. Classe entidade: Empresa



```
package br.fucapi.curso.jee.model.bean;
import java.util.Calendar;

public class Empresa {
    private Long id;
    private String cnpj;
    private String razaoSocial;
    private String endereco;
    private String telefone;
    private String site;
    private String email;
    private Calendar dataCriacao;
    @Override
    public String toString() {
        return cnpj+" - "+razaoSocial;
    }
    //Metodos de get e set aqui
```

04. A classe EmpresaDAO.java



```
1 package br.fucapi.curso.jee.model.dao;
2
3 import java.sql.Connection;
4 import java.sql.Date;
5 import java.sql.PreparedStatement;
6 import java.sql.ResultSet;
7 import java.sql.SQLException;
8 import java.util.ArrayList;
9 import java.util.Calendar;
10 import java.util.List;
11 import br.fucapi.curso.jee.model.bean.Empresa;
12 import br.fucapi.curso.jee.model.connection.ConnectionFactory;
13
14 public class EmpresaDAO {
15
16 //continua
```



Método EmpresaDAO.listar()

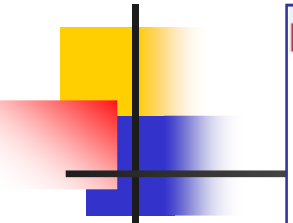
```
public List<Empresa> listar() {  
    //Criação da conexão com o banco de dados  
    Connection connection = ConnectionFactory.getConnection();  
    //Criação do comando SQL  
    String sql = "Select * from empresa";  
    //Criação do objeto de execução de comandos SQL  
    PreparedStatement stmt = null;  
    1 //Criação da Collection a ser retornada  
    List<Empresa> lista = new ArrayList<Empresa>();  
    2 //Criação do objeto que recebe o resultado da execução SQL  
    ResultSet resultSet = null;  
  
    //Continua...
```

Continuação...

//Continua...

```
try {
    stmt = connection.prepareStatement(sql);
    resultSet = stmt.executeQuery();
    while (resultSet.next()) {
        Empresa empresa = new Empresa();
        //Carga do objeto Empresa
        empresa.setId(resultSet.getLong("id"));
        empresa.setCnpj(resultSet.getString("cnpj"));
        empresa.setRazaoSocial(resultSet.getString("razaosocial"));
        empresa.setEndereco(resultSet.getString("endereco"));
        empresa.setTelefone(resultSet.getString("telefone"));
        empresa.setSite(resultSet.getString("site"));
        empresa.setEmail(resultSet.getString("email"));
        Calendar data = Calendar.getInstance();
        data.setTime(resultSet.getDate("dtcriacao"));
        empresa.setDataCriacao(data);
        lista.add(empresa);
    }
} catch (SQLException e) {
    e.printStackTrace();
} finally {
    closeConnection(connection, stmt);
}
return lista;
}
```

Método EmpresaDAO.alterar()



```
public void alterar(Empresa empresa) {
    Connection connection = ConnectionFactory.getConnection();
    String sql = "Update empresa set "
        + "cnpj = ?, razaosocial = ?, endereco = ?, "
        + "telefone = ?, site = ?, email = ?, dtcriacao = ? "
        + "Where id = ?";
    PreparedStatement stmt = null;
    try {
        stmt = connection.prepareStatement(sql);
        stmt.setString(1, empresa.getCnpj());
        stmt.setString(2, empresa.getRazaoSocial());
        stmt.setString(3, empresa.getEndereco());
        stmt.setString(4, empresa.getTelefone());
        stmt.setString(5, empresa.getSite());
        stmt.setString(6, empresa.getEmail());
        stmt.setDate(7, new Date(empresa.getDataCriacao().
                                getTimeInMillis()));
        stmt.setLong(8, empresa.getId());
        stmt.execute();
    } catch (SQLException e) {
        e.printStackTrace();
    } finally{
        closeConnection(connection, stmt);
    }
}
```

1

2

3

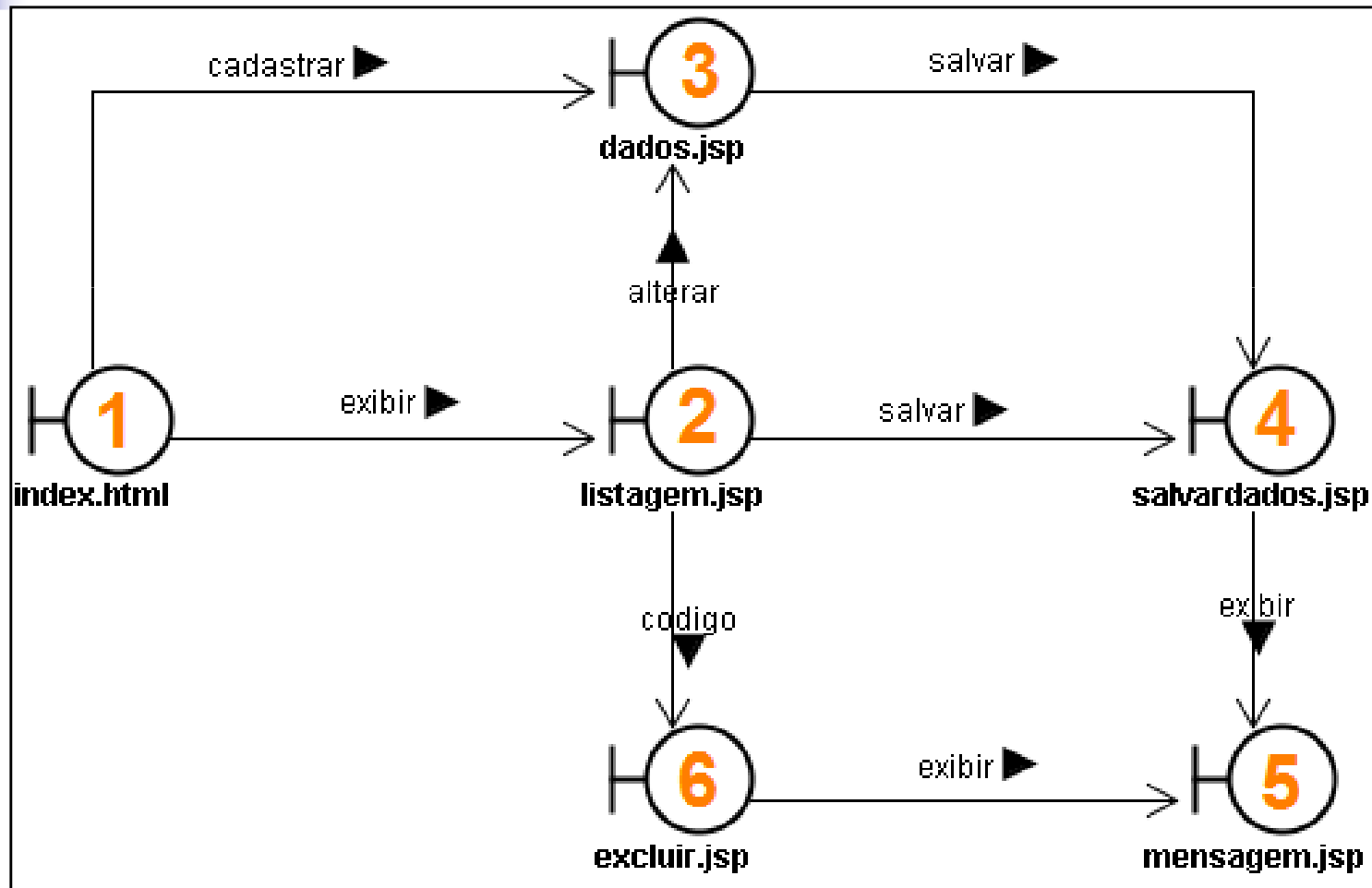
EmpresaDAO.consultar()

```
public Empresa consultar(Long id) {
    Empresa empresa = null;
    Connection connection = ConnectionFactory.getConnection();
    String sql = "Select * from empresa where id = ?";
    PreparedStatement stmt = null;
    ResultSet resultSet = null;
    try {
        stmt = connection.prepareStatement(sql);
        stmt.setLong(1, id);
        resultSet = stmt.executeQuery();
        if (resultSet.next()) {
            empresa = new Empresa();
            empresa.setId(resultSet.getLong("id"));
            empresa.setCnpj(resultSet.getString("cnpj"));
            empresa.setRazaoSocial(resultSet.getString("razaosocial"));
            empresa.setEndereco(resultSet.getString("endereco"));
            empresa.setTelefone(resultSet.getString("telefone"));
            empresa.setSite(resultSet.getString("site"));
            empresa.setEmail(resultSet.getString("email"));
            empresa.setDataCriacao(Calendar.getInstance());
            empresa.getDataCriacao().setTime(resultSet.getDate("dtcriacao"));
        }
    } catch (SQLException e) {
        e.printStackTrace();
    } finally {
        closeConnection(connection, stmt);
    }
    return empresa;
}
```

EmpresaDAO.excluir()

```
154 public void excluir(long id) { 1
155     //Criação da conexão
156     Connection connection = ConnectionFactory.getConnection();
157     2 //criação da String SQL a ser executada
158     String sql = "Delete from empresa Where id = ?";
159     //Criação do objeto a executar o comando SQL
160     PreparedStatement stmt = null;
161     try {
162         //Configuração do statement
163         stmt = connection.prepareStatement(sql);
164         //Carga dos parâmetros da instrução SQL
165         stmt.setLong(1, id);
166         3 //Execução do comando de exclusão
167         stmt.execute();
168     } catch (SQLException e) {
169         e.printStackTrace();
170     } finally{
171         //Encerramento da conexão
172         closeConnection(connection, stmt);
173     }
174 }
```


Exercício 02: Criação das telas no diretório WebContent

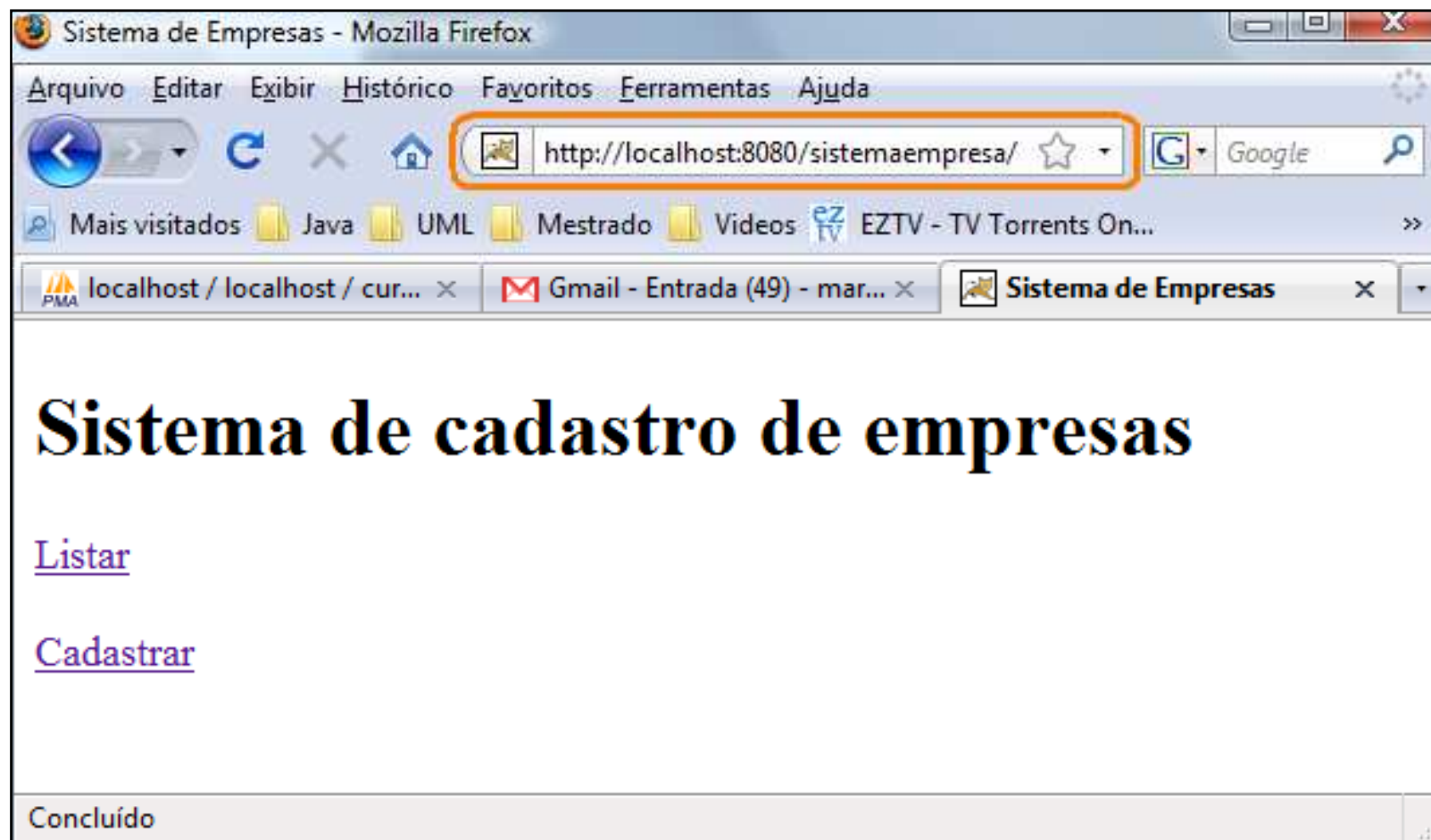




2.1 index.html

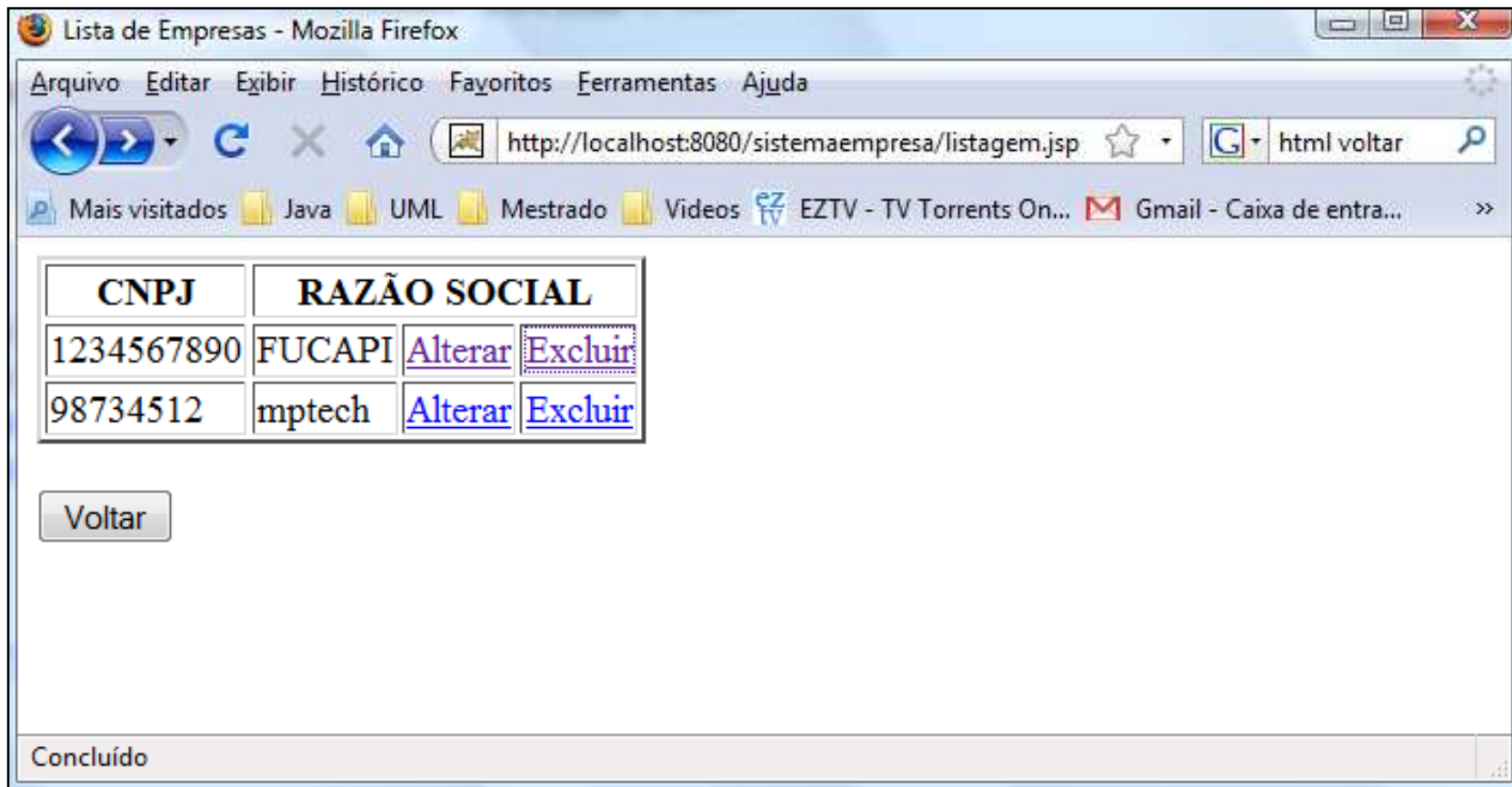
```
index.html X
1<html>
2<head>
3<meta http-equiv="Content-Type"
4      content="text/html; charset=ISO-8859-1">
5<title>Sistema de Empresas</title>
6</head>
7<body>
8    <h1>Sistema de cadastro de empresas</h1>
9    <a href="listagem.jsp">Listar</a><br><br>
10   <a href="dados.jsp">Cadastrar</a>
11</body>
12</html>
```

2.1 Página publicada

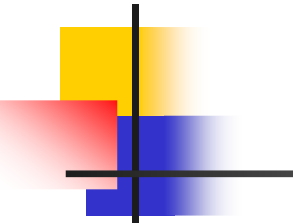


```
<% EmpresaDAO dao = new EmpresaDAO();  
List<Empresa> lista = dao.listar();  
for(Empresa empresa: lista){ %>
```

2.2 Página publicada



2.3 dados.jsp



```
<%@page import="br.fucapi.curso.jee.model.bean.Empresa,
                br.fucapi.curso.jee.model.dao.EmpresaDAO" %>
<html>
<head><title>Dados da empresa</title></head>
<%
    String id = request.getParameter("id");
    EmpresaDAO dao = null;
    Empresa empresa = null;
    if(id!=null){
        dao = new EmpresaDAO();
        empresa = dao.consultar(Long.parseLong(id));
    }else{
        empresa = new Empresa();
    }
%>
<body>
    <h1>
        <%if(id==null){ %>
            Cadastrar empresas
        <%}else{ %>
            Alterar dados da empresa
        <%} %>
    </h1>
    <!-- Continua... -->
```

2.3 dados.jsp (continuação...)

```
<form action="salvardados.jsp" method="post">
<table>
  <%if(empresa.getId()!=null){ %>
    <tr><td>ID: </td>
      <td><input type="text" name="edId" value="<%=empresa.getId()%>" /></td></tr>
  <%} %>
  <tr><td>CNPJ: </td><td>
    <input type="text" name="edCNPJ" value="<%=empresa.getCnpj()%>" /></td></tr>
  <tr><td>RAZÃO SOCIAL: </td>
    <td><input type="text" name="edRazaoSocial" value="<%=empresa.getRazaoSocial()%>" /></td></tr>
  <tr><td>ENDEREÇO: </td>
    <td><input type="text" name="edEndereco" value="<%=empresa.getEndereco()%>" /></td></tr>
  <tr><td>TELEFONE: </td>
    <td><input type="text" name="edTelefone" value="<%=empresa.getTelefone()%>" /></td></tr>
  <tr><td>SITE: </td>
    <td><input type="text" name="edSite" value="<%=empresa.getSite()%>" /></td></tr>
  <tr><td>EMAIL: </td>
    <td><input type="text" name="edEmail" value="<%=empresa.getEmail()%>" /></td></tr>
  <tr><td>Data de Criação: </td>
    <td><input type="text" name="edData" value="<%=empresa.getDataCriacao()%>" /></td></tr>
  <tr>
    <td><input type="submit" value="Enviar dados" /></td>
    <td><input type="button" value="Cancelar" onclick="history.go(-1)" /></td></tr>
</table>
</form></body></html>
```

2.3 Página publicada



A screenshot of a web browser window. The title bar shows 'Dados da empresa'. The address bar contains 'localhost:8080/sistemaempresa/dados.jsp'. The page title is 'Cadastrar empresas'. The form contains several input fields for company registration, followed by 'Enviar dados' and 'Cancelar' buttons.

Dados da empresa

localhost:8080/sistemaempresa/dados.jsp

Cadastrar empresas

ID:

CNPJ:

RAZÃO SOCIAL:

ENDEREÇO:

TELEFONE:

SITE:

EMAIL:

Data de Criação:



2.4 salvardados.jsp

```
<%@page import="br.fucapi.curso.jee.model.bean.Empresa,
                br.fucapi.curso.jee.model.dao.EmpresaDAO,
                java.text.SimpleDateFormat, java.util.*" %>
<html> <head><title>Tela de sucesso</title></head>
<% String operacao = ""; EmpresaDAO dao = new EmpresaDAO();
    Empresa empresa = new Empresa();
    String id = request.getParameter("edId");
    empresa.setCnpj(request.getParameter("edCNPJ"));
    empresa.setRazaoSocial(request.getParameter("edRazaoSocial"));
    empresa.setEndereco(request.getParameter("edEndereco"));
    empresa.setTelefone(request.getParameter("edTelefone"));
    empresa.setSite(request.getParameter("edSite"));
    empresa.setEmail(request.getParameter("edEmail"));
    Date date = new SimpleDateFormat("dd/MM/yyyy").parse(
        request.getParameter("edData"));
    empresa.setDataCriacao(Calendar.getInstance());
    empresa.getDataCriacao().setTime(date);

    //Continua
```

2.4 continuação...

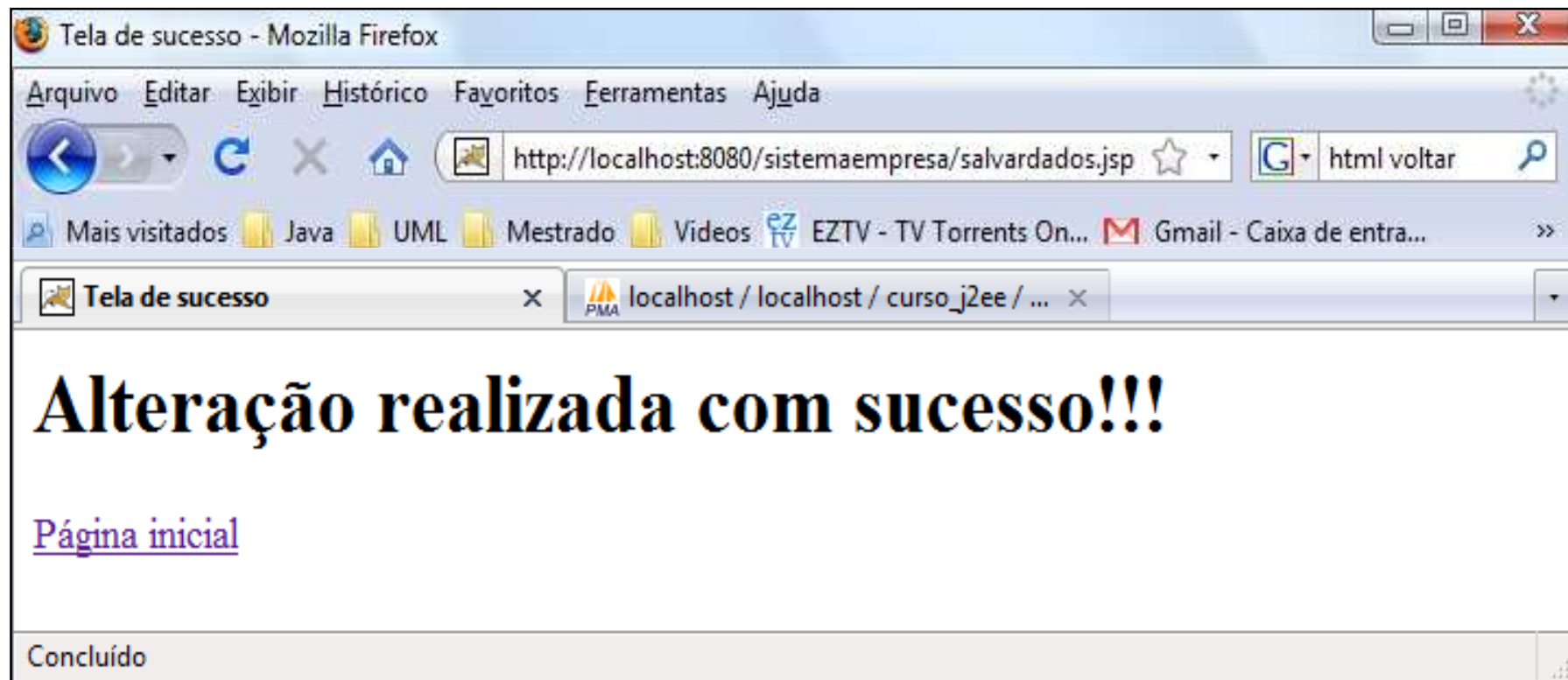
```
//Continua

if(id == null){
    dao.cadastrar(empresa);
    operacao = "Cadastro realizado ";
}else{
    empresa.setId(new Long(id));
    dao.alterar(empresa);
    operacao = "Alteração realizada ";
}

%>

<body>
    <h1><%=operacao %> com sucesso</h1>
    <a href="index.html"> Página inicial</a><br>
    <a href="listagem.jsp"> Página de listagem</a>
</body>
</html>
```

2.4 Página publicada

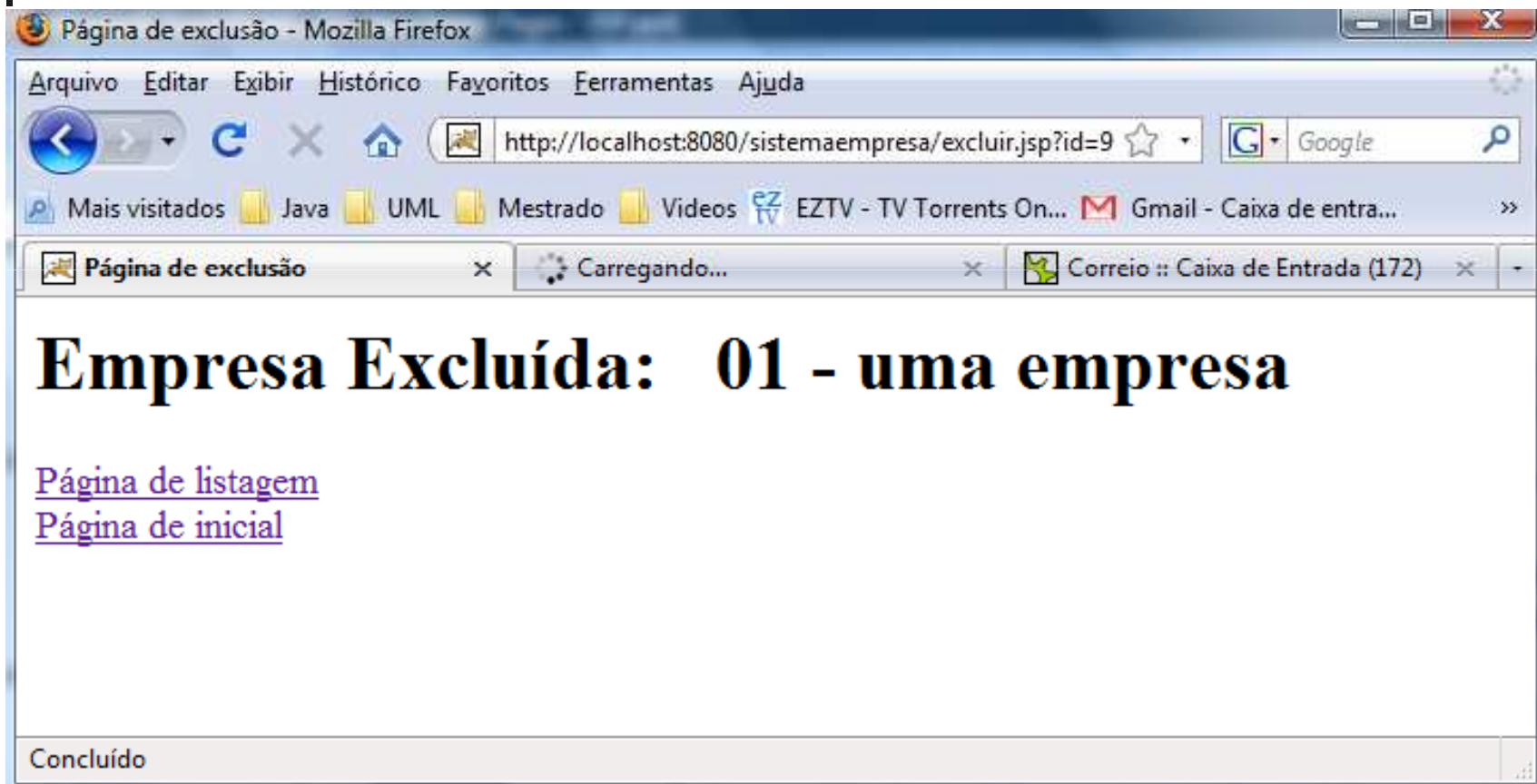




2.5 excluir.jsp

```
<%@page import="br.fucapi.curso.jee.model.bean.Empresa,
                br.fucapi.curso.jee.model.dao.EmpresaDAO" %>
<html> <head><title>Excluir empresa</title></head>
<%
    String id = request.getParameter("id");
    EmpresaDAO dao = new EmpresaDAO();
    Empresa empresa = dao.consultar(new Long(id));
    dao.excluir(empresa.getId());
%>
<body>
    <h1>Empresa Escluída: <br>
        <%=empresa.getCnpj()+" - "+empresa.getRazaoSocial() %>
    </h1>
    <a href="index.html"> Página inicial</a>
    <a href="listagem.jsp"> Página de listagem</a>
</body>
</html>
```

2.5 Página publicada





O que vem a seguir?

- JSTL;
- Fim dos scriptlets;
- A camada de controle;
- Fluxo de páginas;
- Servlets ou JSPs?
- Configurações;
- Arquitetura MVC;



Referências

- www.caelum.com.br
- Hall, Marty, "Core Servlets and Java Server Pages", Janeiro 2002, Sun Microsystems Press;
- <http://java.sun.com/j2ee/1.6/docs/tutorial/doc/index.html>
- <http://java.sun.com/products/jndi/docs.html>
- <http://java.sun.com/blueprints/corej2eepatterns/Patterns/index.html>

Java Enterprise Edition - JEE

06. Java Server Pages - JSP



Esp. Márcio Palheta
gtalk: marcio.palheta@gmail.com