



## Java WebDeveloper - Aula 06b - Noite Seg, Qua, Sex - Noite


---



### Tema da aula:

---

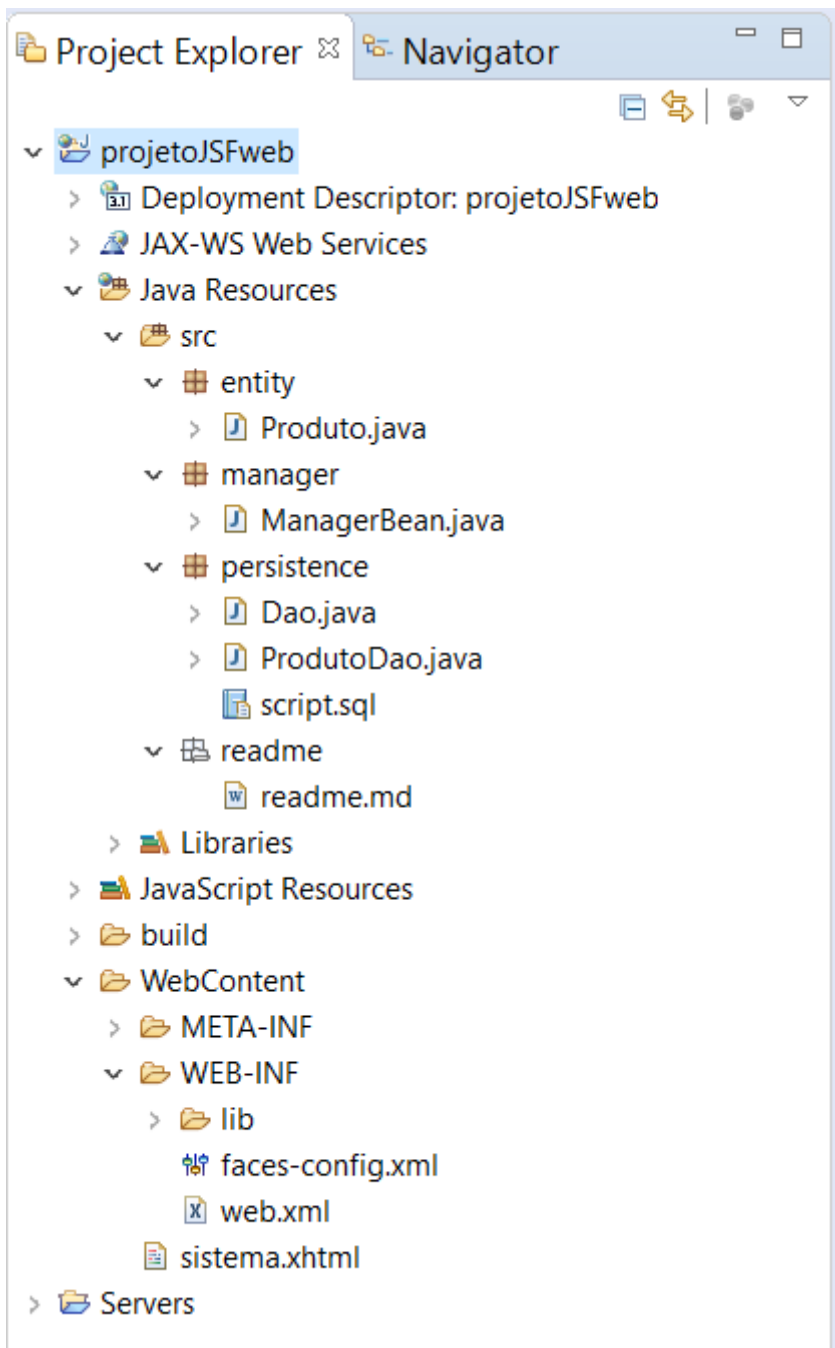
Projeto JSF, Serializable, Script, JDBC, MySQL, Dao, Configuração JSF, Lista, Tema PrimeFaces, Panel, Datalist, PanelGrid, Messages, Growl, DataTable, setPropertyActionListener, Dialog, ajax

—  Prof Edson Belém - [profedsonbelem@gmail.com](mailto:profedsonbelem@gmail.com) (<mailto:profedsonbelem@gmail.com>) 🕒 Segunda, Abr 13, 2020



### Estrutura do projeto depois de finalizado:

---



## Bibliotecas utilizadas (JSF):

---

▼ 📁 lib

- 📄 all-themes-1.0.10.jar
- 📄 commons-beanutils-1.8.0.jar
- 📄 commons-collections-3.2.1.jar
- 📄 commons-digester-2.0.jar
- 📄 commons-logging-1.1.1.jar
- 📄 javax.faces-2.0.11.jar
- 📄 jstl.jar
- 📄 mysql-connector-java-5.1.23-bin.jar
- 📄 primefaces-5.0-sources.jar
- 📄 primefaces-5.0.jar
- 📄 standard.jar



## Readme.md (<http://Readme.md>)

JSF (Concorrente ==> JEE Servlet)(EspecificaÃ§Ã£o)

# Parte um

# Configuracao

#pacotes

entity, manager, persistence (view)

view --> Manger

tela --> Classe

xhtml

##



## Produto.java

```
1 package entity;
2
3 import java.io.Serializable;
4
5 public class Produto implements Serializable {
6
7     private static final long serialVersionUID = 1L;
8     private Integer id;
9     private String nome;
10    private String autor;
11    private Double preco;
12
13    public Produto() {
14    }
15
16    public Produto(Integer id, String nome, String autor, Double preco) {
17        this.id = id;
18        this.nome = nome;
19        this.autor = autor;
20        this.preco = preco;
21    }
22
23    @Override
24    public String toString() {
25        return "Produto [id=" + id + ", nome=" + nome + ", autor="
26            + autor + ", preco=" + preco + "]";
27    }
28
29    public Integer getId() {
30        return id;
31    }
32
33    public void setId(Integer id) {
34        this.id = id;
35    }
36
37    public String getNome() {
38        return nome;
39    }
40
41    public void setNome(String nome) {
42        this.nome = nome;
43    }
44
45    public String getAutor() {
46        return autor;
47    }
48
49    public void setAutor(String autor) {
50        this.autor = autor;
51    }
52
53    public Double getPreco() {
54        return preco;
```

```

55         }
56
57         public void setPreco(Double preco) {
58             this.preco = preco;
59         }
60
61     }

```

## script.sql

```

1  mysql -u root -p
2
3  create database BDJSFUM;
4  use BDJSFUM;
5
6  create table produto(
7      id int primary key auto_increment,
8      nome varchar (50) unique ,
9      autor varchar (50),
10     preco double
11 );
12
13 show tables;
14
15 insert into produto (id,nome,autor,preco) values
16     (null, 'Java Certificacao', 'kathy sierra',200),
17     (null, 'Java OO', 'Douglas Rocha',80),
18     (null, 'Java EE', 'Cleiton Sampaio',70);
19
20 select * from produto;

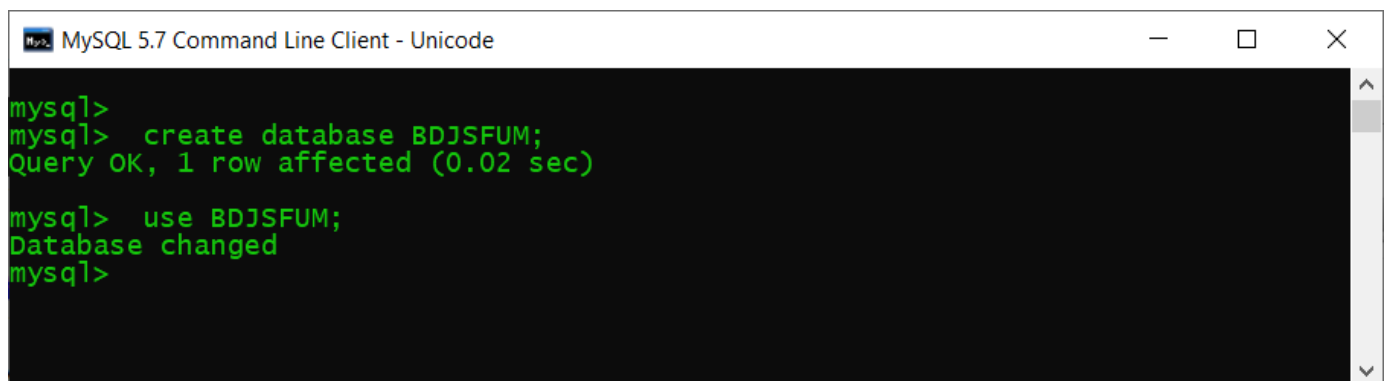
```

 Visualizando no banco.

```

1  create database BDJSFUM;
2  use BDJSFUM;

```



The screenshot shows a terminal window titled "MySQL 5.7 Command Line Client - Unicode". The prompt is "mysql>". The user enters "create database BDJSFUM;" and the response is "Query OK, 1 row affected (0.02 sec)". Then the user enters "use BDJSFUM;" and the response is "Database changed". The prompt remains "mysql>".

```

mysql>
mysql> create database BDJSFUM;
Query OK, 1 row affected (0.02 sec)

mysql> use BDJSFUM;
Database changed
mysql>

```

```

1      create table produto(
2          id int primary key auto_increment,
3          nome varchar (50) unique ,
4          autor varchar (50),
5          preco double
6      );
7
8      show tables;

```

```

mysql>
mysql> create table produto(
-> id int primary key auto_increment,
-> nome varchar (50) unique ,
-> autor varchar (50),
-> preco double
-> );
Query OK, 0 rows affected (0.08 sec)

mysql>
mysql> show tables;
+-----+
| Tables_in_bdjsfum |
+-----+
| produto           |
+-----+
1 row in set (0.00 sec)

```

```

1      insert into produto (id,nome,autor,preco) values
2          (null, 'Java Certificacao', 'kathy sierra',200),
3          (null, 'Java OO', 'Douglas Rocha',80),
4          (null, 'Java EE', 'Cleiton Sampaio',70);
5
6      select * from produto;

```

```

mysql>
mysql> insert into produto (id,nome,autor,preco) values
-> (null, 'Java Certificacao', 'kathy sierra',200),
-> (null, 'Java OO', 'Douglas Rocha',80),
-> (null, 'Java EE', 'Cleiton Sampaio',70);
Query OK, 3 rows affected (0.02 sec)
Records: 3  Duplicates: 0  Warnings: 0

mysql>
mysql> select * from produto;
+----+-----+-----+-----+
| id | nome          | autor          | preco |
+----+-----+-----+-----+
| 1  | Java Certificacao | kathy sierra  | 200   |
| 2  | Java OO         | Douglas Rocha  | 80    |
| 3  | Java EE         | Cleiton Sampaio | 70    |
+----+-----+-----+-----+
3 rows in set (0.00 sec)

mysql>

```



## Dao.java

```
1  package persistence;
2
3  import java.sql.Connection;
4  import java.sql.DriverManager;
5  import java.sql.PreparedStatement;
6  import java.sql.ResultSet;
7
8  public class Dao {
9
10     Connection con;
11     PreparedStatement stmt;
12     ResultSet rs;
13
14     public void open() throws Exception {
15         Class.forName("com.mysql.jdbc.Driver");
16         con = DriverManager.getConnection
17             ("jdbc:mysql://localhost:3306/BDJSFUM", "root", "coti");
18     }
19
20     public void close() throws Exception {
21         con.close();
22     }
23
24 }
```



## ProdutoDao.java

```

1  package persistence;
2
3  import java.util.ArrayList;
4  import java.util.List;
5  import entity.Produto;
6
7
8  public class ProdutoDao extends Dao {
9      public void create(Produto p) throws Exception{
10         open();
11         stmt= con.prepareStatement
12             ("insert into produto values (null,?,?,?)");
13         stmt.setString(1, p.getNome());
14         stmt.setString(2, p.getAutor());
15         stmt.setDouble(3, p.getPreco());
16         stmt.execute();
17         close();
18     }
19
20
21     public List<Produto> findAll() throws Exception{
22         open();
23         stmt = con.prepareStatement("select * from produto");
24         rs = stmt.executeQuery();
25         List<Produto> lista =new ArrayList<Produto>();
26         while(rs.next()) {
27             Produto p = new Produto();
28             p.setId(rs.getInt(1));
29             p.setNome(rs.getString(2));
30             p.setAutor(rs.getString(3));
31             p.setPreco(rs.getDouble(4));
32             lista.add(p);
33         }
34         close();
35         return lista;
36     }
37
38
39     public Produto findByCode(Integer cod) throws Exception{
40         open();
41         stmt = con.prepareStatement
42             ("select * from produto where id = ?");
43         stmt.setInt(1, cod);
44         rs = stmt.executeQuery();
45         Produto produto = null;
46         if (rs.next()) {
47             produto = new Produto(
48                 rs.getInt("id"),
49                 rs.getString("nome"),
50                 rs.getString("autor"),
51                 rs.getDouble("preco")
52             );
53         }
54         close();

```




```

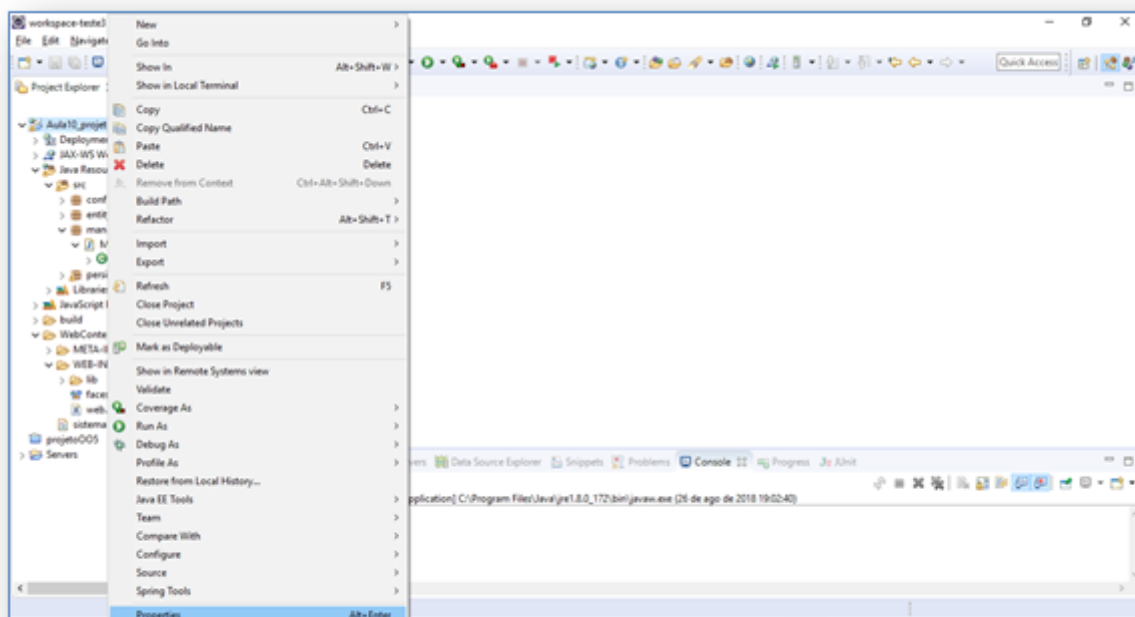
55         return produto;
56     }
57
58
59     public void update(Produto p) throws Exception{
60         open();
61         stmt= con.prepareStatement
62         ("update produto set nome=?, autor=?, preco=? where id=?");
63         stmt.setString(1, p.getNome());
64         stmt.setString(2, p.getAutor());
65         stmt.setDouble(3, p.getPreco());
66         stmt.setInt(4, p.getId());
67         stmt.execute();
68         close();
69     }
70
71
72     public void delete(Integer cod) throws Exception{
73         open();
74         stmt= con.prepareStatement
75         ("delete from produto where id=?");
76         stmt.setInt(1, cod);
77         stmt.execute();
78         close();
79     }
80
81
82     public static void main(String[] args) {
83         try {
84             Produto p = new Produto(null,"Nodejs","belem", 20.);
85             ProdutoDao dao = new ProdutoDao();
86             dao.create(p);
87             dao.findAll().stream().forEach(x->{
88                 System.out.println(x.getNome() + ", " + x.getPreco());
89             }
90             );
91         }catch(Exception ex) {
92             ex.printStackTrace();
93         }
94     }
95 }


```

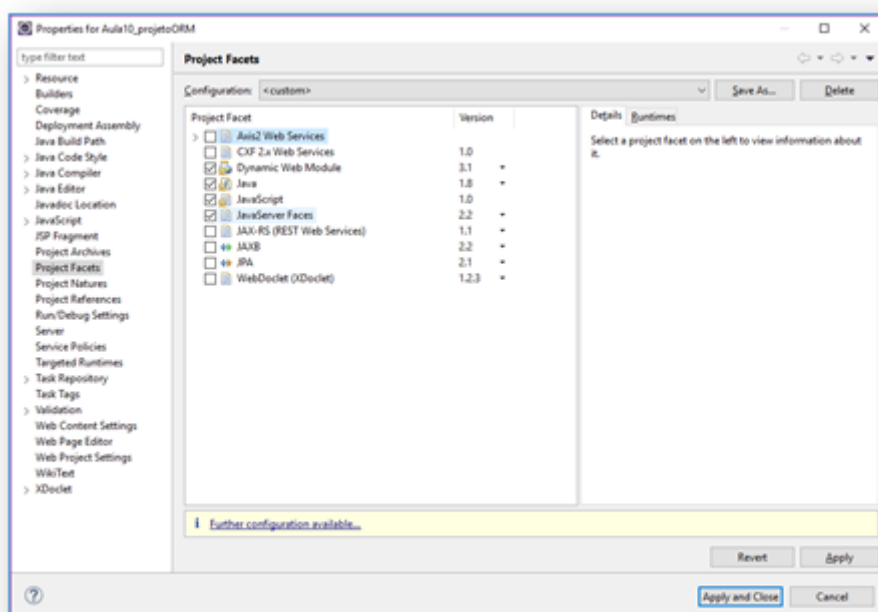



**Para habilitar o JSF no projeto:**

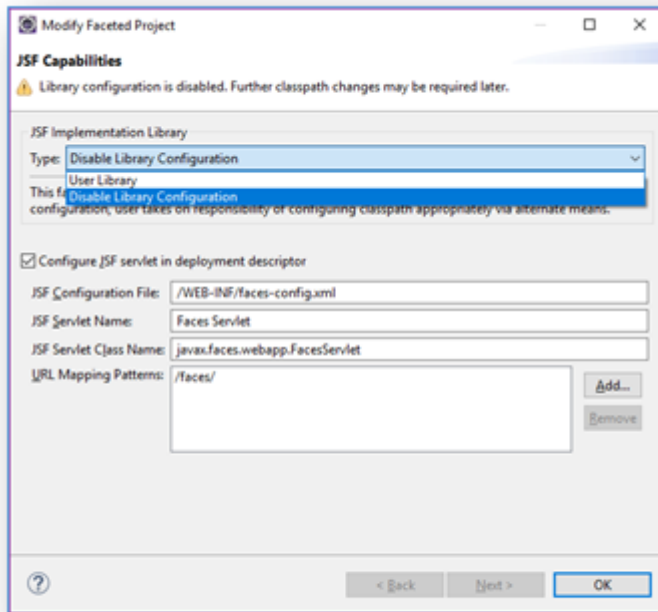
 Clicar no projeto com o botão direito.




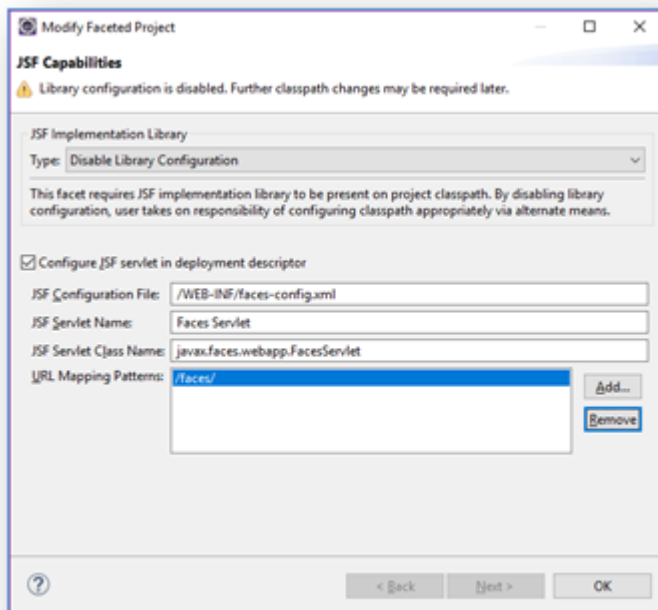
 Clicar em "project facets" → marcar "java server faces" -> clicar no link "further configuration"




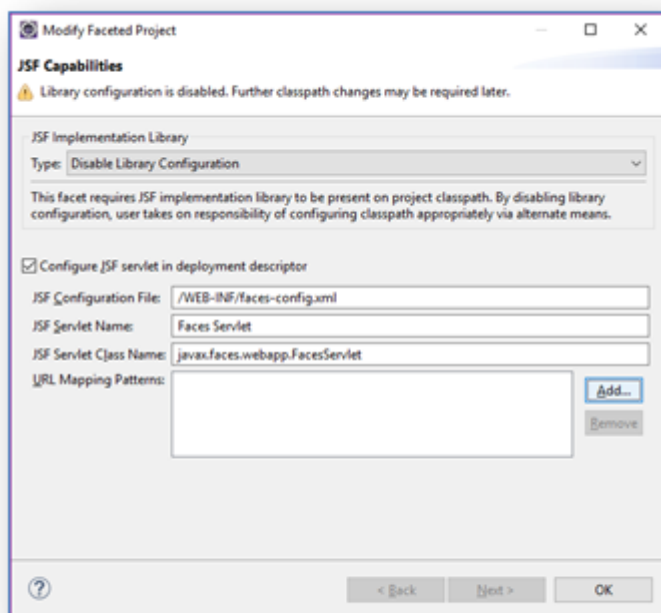
 Clicar em type e selecionar "disable library"




 Clicar em "url mapping" → clicar em "remove"

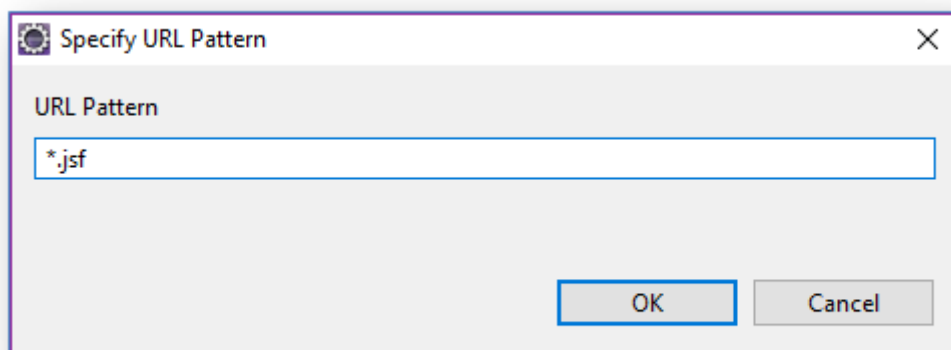


 Clicar em "add"




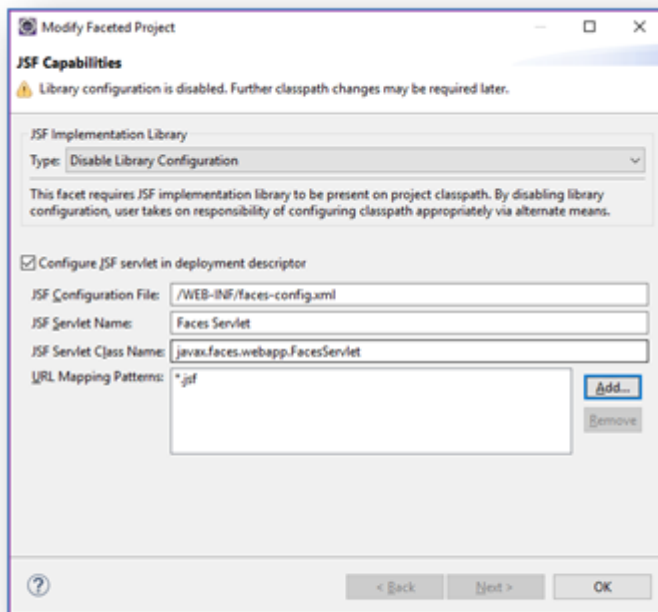
The screenshot shows the 'Modify Faceted Project' dialog box with the 'JSF Capabilities' tab selected. A warning message at the top states: 'Library configuration is disabled. Further classpath changes may be required later.' Below this, the 'JSF Implementation Library' section shows 'Type: Disable Library Configuration'. A note explains that this facet requires JSF implementation library to be present on project classpath. The 'Configure JSF servlet in deployment descriptor' checkbox is checked. Fields for 'JSF Configuration File' (set to '/WEB-INF/faces-config.xml'), 'JSF Servlet Name' (set to 'Faces Servlet'), and 'JSF Servlet Class Name' (set to 'javax.faces.webapp.FacesServlet') are visible. The 'URL Mapping Patterns' list is empty, and the 'Add...' button is highlighted with a blue border. 'Remove' and 'OK' buttons are also present at the bottom.


 Escrever a "url pattern" → ok

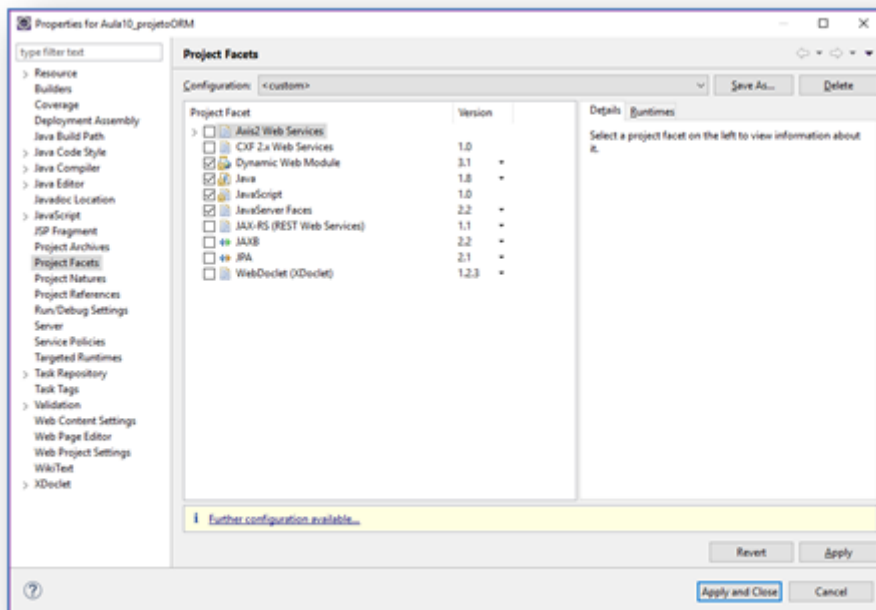


The screenshot shows the 'Specify URL Pattern' dialog box. The 'URL Pattern' text field contains the text '\*.jsf'. The 'OK' button is highlighted with a blue border, and the 'Cancel' button is also visible.

 Clicar em "ok"



 Clicar em "apply and close"



 ManagerBean.java

```
1 package manager;
2
3 import java.util.List;
4
5 import javax.faces.application.FacesMessage;
6 import javax.faces.bean.ManagedBean;
7 import javax.faces.bean.RequestScoped;
8 import javax.faces.context.FacesContext;
9
10 import entity.Produto;
11 import persistence.ProdutoDao;
12
13 @ManagedBean(name = "mb")
14 @RequestScoped
15 public class ManagerBean {
16
17     private Produto produto; // unidade
18     private List<Produto> produtos; // grupo (consulta do Banco)
19
20     public ManagerBean() {
21         this.produto = new Produto();
22     }
23
24
25     public Produto getProduto() {
26         return produto;
27     }
28
29     public void setProduto(Produto produto) {
30         this.produto = produto;
31     }
32
33     // lista populada ...
34     public List<Produto> getProdutos() {
35         try {
36             produtos = new ProdutoDao().findAll();
37         } catch (Exception ex) {
38
39         }
40         return produtos;
41     }
42
43     public void setProdutos(List<Produto> produtos) {
44         this.produtos = produtos;
45     }
46
47
48     public void gravar() {
49         FacesContext fc = FacesContext.getCurrentInstance();
50         try {
51             new ProdutoDao().create(produto);
52             this.produto = new Produto();
53             fc.addMessage(null, new FacesMessage
54                 ("Dados gravados ..."));
55         }
```

```

55         } catch (Exception ex) {
56             ex.printStackTrace();
57             fc.addMessage(null, new FacesMessage
58                 ("Error ..." + ex.getMessage()));
59         }
60     }
61
62
63     public void excluir() {
64         FacesContext fc = FacesContext.getCurrentInstance();
65         try {
66             new ProdutoDao().delete(produto.getId());
67             this.produto = new Produto();
68             fc.addMessage(null, new FacesMessage
69                 ("Dados Excluidos ..."));
70         } catch (Exception ex) {
71             ex.printStackTrace();
72             fc.addMessage(null, new FacesMessage
73                 ("Error ..." + ex.getMessage()));
74         }
75     }
76
77
78     public void alterar() {
79         System.out.println("Olha quem estou alterando" + produto);
80         FacesContext fc = FacesContext.getCurrentInstance();
81         try {
82             new ProdutoDao().update(produto);
83             this.produto = new Produto();
84             fc.addMessage(null, new FacesMessage
85                 ("Dados Alterados ..."));
86
87         } catch (Exception ex) {
88             ex.printStackTrace();
89             fc.addMessage(null, new FacesMessage
90                 ("Error ..." + ex.getMessage()));
91         }
92     }
93 }

```

```
1  <?xml version="1.0" encoding="UTF-8"?>
2  <web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
3      xmlns="http://xmlns.jcp.org/xml/ns/javaee"
4      xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee
5      http://xmlns.jcp.org/xml/ns/javaee/web-app_3_1.xsd"
6      id="WebApp_ID" version="3.1">
7      <display-name>projetoJSFweb</display-name>
8      <welcome-file-list>
9          <welcome-file>index.html</welcome-file>
10         <welcome-file>index.htm</welcome-file>
11         <welcome-file>index.jsp</welcome-file>
12         <welcome-file>default.html</welcome-file>
13         <welcome-file>default.htm</welcome-file>
14         <welcome-file>default.jsp</welcome-file>
15     </welcome-file-list>
16     <servlet>
17         <servlet-name>Faces Servlet</servlet-name>
18         <servlet-class>javax.faces.webapp.FacesServlet
19         </servlet-class>
20         <load-on-startup>1</load-on-startup>
21     </servlet>
22     <servlet-mapping>
23         <servlet-name>Faces Servlet</servlet-name>
24         <url-pattern>*.jsf</url-pattern>
25     </servlet-mapping>
26
27     <context-param>
28         <param-name>primefaces.THEME</param-name>
29         <param-value>blitzer</param-value>
30     </context-param>
31
32 </web-app>
```



sistema.xhtml



```

1  <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
2  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
3
4  <html xmlns="http://www.w3.org/1999/xhtml"
5      xmlns:ui="http://java.sun.com/jsf/facelets"
6      xmlns:h="http://java.sun.com/jsf/html"
7      xmlns:f="http://java.sun.com/jsf/core"
8      xmlns:p="http://primefaces.org/ui">
9
10 <h:head>
11 </h:head>
12
13 <h:body>
14
15     <p:panel header="Cadastro de Livro">
16         <h:form id="form1">
17             <p:panelGrid columns="2">
18                 <h:outputText value="Nome" />
19                 <p:inputText value="#{mb.produto.nome}" />
20                 <h:outputText value="Autor" />
21                 <p:inputText value="#{mb.produto.autor}" />
22                 <h:outputText value="Preco" />
23                 <p:inputText value="#{mb.produto.preco}" />
24                 <p:commandButton value="Gravar"
25                     action="#{mb.gravar}"
26                     update=":form1,:form2,:form3"
27                     ajax="true">
28                     </p:commandButton>
29             </p:panelGrid>
30
31             <p:messages></p:messages>
32             <p:growl></p:growl>
33
34         </h:form>
35     </p:panel>
36
37
38     <p:panel header="Consulta dos Livros">
39         <h:form id="form2">
40             <p:dataList value="#{mb.produtos}" var="item">
41                 #{item.nome} , #{item.preco}
42             </p:dataList>
43         </h:form>
44     </p:panel>
45
46
47     <p:panel header="DataTable">
48         <h:form id="form3">
49
50             <p:dataTable value="#{mb.produtos}" var="linha">
51
52                 <p:column headerText="Id" sortBy="#{linha.id}">
53                     <h:outputText value="#{linha.id}" />
54

```

```

55
56     <p:column headerText="Nome" filterBy="#{linha.nome}">
57         <h:outputText value="#{linha.nome}" />
58     </p:column>
59
60     <p:column headerText="Autor" sortBy="#{linha.autor}">
61         <h:outputText value="#{linha.autor}" />
62     </p:column>
63
64     <p:column headerText="Preco">
65         <h:outputText value="#{linha.preco}" />
66     </p:column>
67
68     <p:column headerText="Excluir">
69         <p:commandButton value="Excluir" ajax="true"
70             update=":form3, :form2"
71             action="#{mb.excluir}">
72             <f:setPropertyActionListener value="#{linha}"
73                 target="#{mb.produto}" />
74         </p:commandButton>
75     </p:column>
76
77     <p:column headerText="Alterar">
78         <p:commandButton value="Editar"
79             ajax="true"
80             update=":form5"
81             onclick="PF('edita1').show()">
82             <f:setPropertyActionListener value="#{linha}"
83                 target="#{mb.produto}" />
84         </p:commandButton>
85     </p:column>
86 </p:dataTable>
87 </h:form>
88 </p:panel>
89
90 <p:dialog header="Alterar" widgetVar="edita1" modal="true">
91     <h:form id="form5">
92         <h:inputHidden value="#{mb.produto.id}" />
93
94         <p:panelGrid columns="2">
95             <h:outputText value="Nome" />
96             <p:inputText value="#{mb.produto.nome}" />
97
98             <h:outputText value="Autor" />
99             <p:inputText value="#{mb.produto.autor}" />
100
101             <h:outputText value="Preco" />
102             <p:inputText value="#{mb.produto.preco}" />
103
104             <p:commandButton value="Alterar"
105                 onclick="PF('edita1').hide()"
106                 update=":form2, :form3"
107                 ajax="true"
108                 action="#{mb.alterar}">
109         </p:commandButton>

```

```

110         </p:panelGrid>
111         </h:form>
112     </p:dialog>
113 </h:body>
114
115 </html>

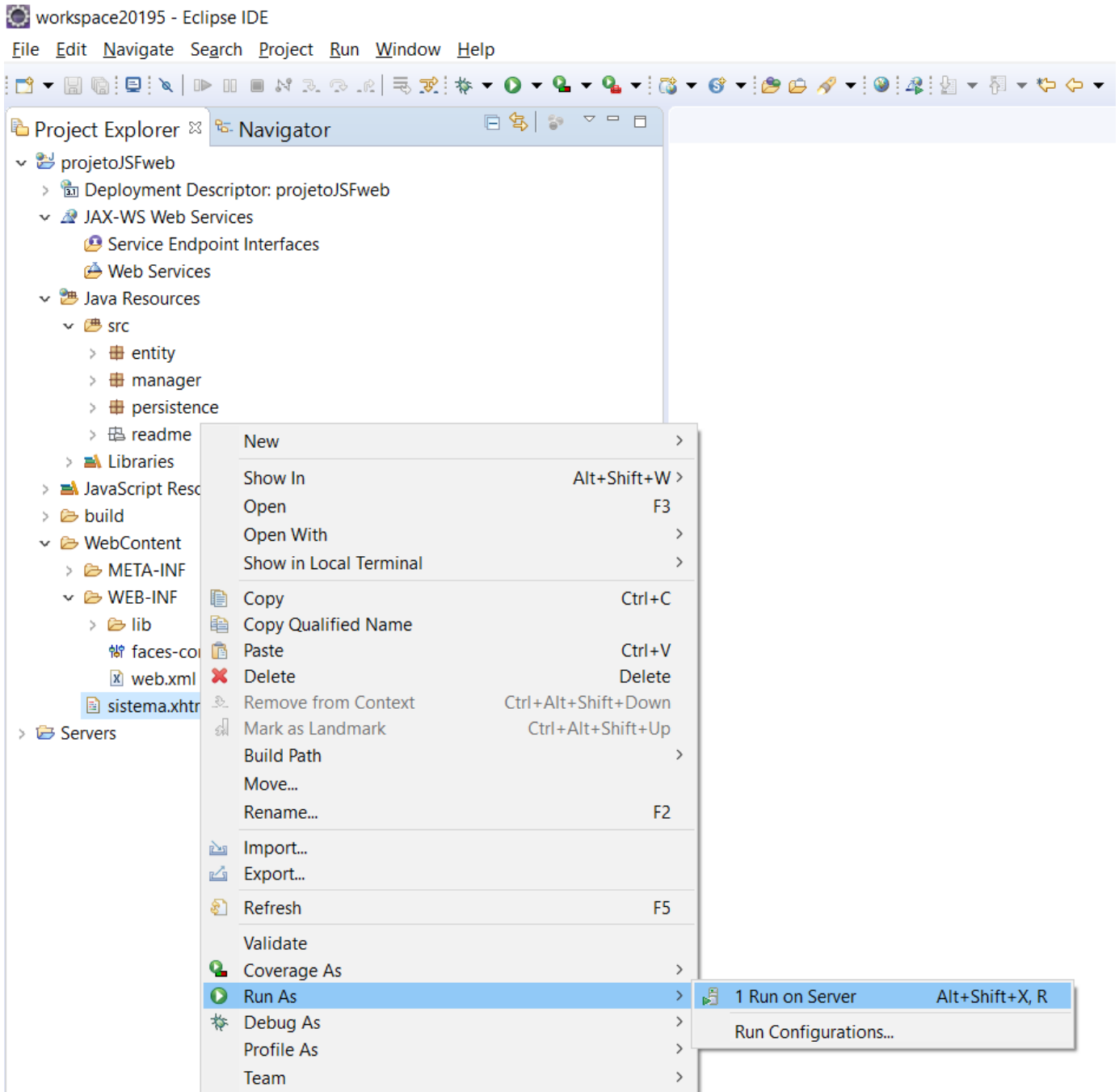
```




## Para rodar a página:




Clique na classe com o botão direito  run as  run on server



Selecionar o tomcat  Finish

 Run On Server

**Run On Server**  
Select which server to use



How do you want to select the server?

☒ Choose an existing server

☐ Manually define a new server


Select the server that you want to use:

type filter text

Server	State
▼ localhost	
Tomcat v8.0 Server at localhost	Stopped

Apache Tomcat v8.0 supports J2EE 1.2, 1.3, 1.4, and Java EE 5, 6, and 7 Web modules. Columns...


☐ Always use this server when running this project


 < Back

Next >

Finish

Cancel

 Aguardar abrir o browser. Podendo gravar, filtrar, alterar e excluir.

 <http://localhost:8080/projetoJSFweb/sistema.jsf>

localhost:8080/projetoJSFweb/sistema.jsf

### Cadastro de Livro

Nome

Autor

Preço

### Consulta dos Livros

- Java Certificacao , 200.0
- Java OO , 80.0
- Java EE , 70.0

### DataTable

Id	Nome	Autor	Preço	Excluir	Alterar
1	Java Certificacao	kathy sierra	200.0	<input type="button" value="Excluir"/>	<input type="button" value="Editar"/>
2	Java OO	Douglas Rocha	80.0	<input type="button" value="Excluir"/>	<input type="button" value="Editar"/>
3	Java EE	Cleiton Sampaio	70.0	<input type="button" value="Excluir"/>	<input type="button" value="Editar"/>

 Clicando em excluir o livro **Java EE**

localhost:8080/projetoJSFweb/sistema.jsf

### Cadastro de Livro

Nome

Autor

Preço

### Consulta dos Livros

- Java Certificacao , 200.0
- Java OO , 80.0

### DataTable

Id	Nome	Autor	Preço	Excluir	Alterar
1	Java Certificacao	kathy sierra	200.0	<input type="button" value="Excluir"/>	<input type="button" value="Editar"/>
2	Java OO	Douglas Rocha	80.0	<input type="button" value="Excluir"/>	<input type="button" value="Editar"/>

 Clicando em alterar o livro **Java OO**

localhost:8080/projetoJSFweb/sistema.jsf

### Cadastro de Livro

Nome

Autor

Preço

Gravar

### Consulta dos Livros

- Java Certificacao , 200.0
- Java OO , 80.0

### DataTable

Id	Nome			Excluir	Alterar
1	Java Certificacao	kathy sierra	200.0	Excluir	Editar
2	Java OO	Douglas Rocha	80.0	Excluir	Editar

**Alterar**  
Nome   
Autor   
Preço   
Alterar

✎ Alterando o nome do livro

localhost:8080/projetoJSFweb/sistema.jsf

### Cadastro de Livro

Nome

Autor

Preço

Gravar

### Consulta dos Livros

- Java Certificacao , 200.0
- Java OO , 80.0

### DataTable

Id	Nome			Excluir	Alterar
1	Java Certificacao	kathy sierra	200.0	Excluir	Editar
2	Java OO	Douglas Rocha	80.0	Excluir	Editar

**Alterar**  
Nome   
Autor   
Preço   
Alterar

✎ Nome foi alterado nas duas listas

## Consulta dos Livros

- Java Certificacao , 200.0
- Java Orientação a Objetos , 80.0

## DataTable

Id 	Nome
1	Java Certificacao
2	Java Orientação a Objetos