

1



2

### Java para Web - JEE (Java Enterprise Edition)

O Java EE é um ambiente independente da plataforma, centrado em Java que cria e implementa aplicativos corporativos baseados na Web online. O Java EE inclui muitos componentes do Java Standard Edition (Java SE). A plataforma Java EE consiste em um conjunto de servidores APIs e protocolos que fornecem a funcionalidade para desenvolver aplicativos multicamadas com base na Web.

O JEE consiste em várias especificações de como deve ser implantado um software e a sua infraestrutura.

As especificações do JEE consistem em algumas principais APIs (Application Programming Interface): Java Servlets, Java Server Faces (JSF) e JavaServer Pages (JSP) considerado em desuso ultimamente. Essas são especificações essenciais para se trabalhar com web. Mesmo utilizando outros frameworks para a criação, é essencial aprender essas especificações, pois são pilares consistentes para o desenvolvimento.

**Download:** <http://www.oracle.com/technetwork/pt/java/index.html>

3

### Java – JRE x JDK (qual instalar?)

**JRE (Java Runtime Environment)**  
Formado pela JVM e pela biblioteca de classes Java utilizadas para execução de aplicações java, estas bibliotecas são chamadas de APIs Java. Portanto para rodarmos uma aplicação java é necessário instalarmos uma JRE no computador onde o software foi instalado.

**JDK (Java Development Kit)**  
É um conjunto de ferramentas necessárias para realizar o desenvolvimento de aplicações java e inclui a JRE e ferramentas de programação, como:  
javac – compilador  
jar – empacotador  
javadoc – ferramenta para geração de documentação

4

### Servlet

Como é comum na plataforma Java, foi definida uma especificação para padronizar a interface dos recursos oferecidos pelos Web Containers., chamada Servlet. Ver <https://jcp.org/en/jsr/detail?id=369>.

O nome "servlet" vem da ideia de um pequeno servidor cujo objetivo é receber chamadas HTTP, processá-las e devolver uma resposta ao cliente.

Cada servlet é, portanto, um objeto Java que recebe tais requisições (**request**) e produz algo (**response**), como uma página HTML dinamicamente gerada.

Ver <https://jcp.org/en/jsr/detail?id=366>.

5

### Java Server Faces (JSF)

JSF é uma tecnologia que nos permite criar aplicações Java para Web utilizando componentes visuais pré-prontos, de forma que o desenvolvedor não se preocupe com Javascript e HTML. Basta adicionar os componentes (calendários, tabelas, formulários) e eles serão renderizados e exibidos em formato html.

O JSF é uma especificação para o desenvolvimento de aplicações Web, seguindo o padrão Model View Controller (MVC) em Java, essa especificação surgiu como uma alternativa ao Struts, que na época era o principal framework para implementar aplicações nesse padrão de projeto.

6

## Java Server Faces (JSF)

Existem diversos frameworks para construção de interfaces ricas para o JSF, como o Mojarra, RichFaces, o IceFaces, PrimeFaces.

O JSF ainda tem a vantagem de ser uma especificação do Java EE, isto é, todo servidor de aplicações Java tem que vir com uma implementação dela e há diversas outras disponíveis. A implementação mais famosa do JSF e também a implementação de referência, é a Oracle Mojarra disponível em <http://javaserverfaces.java.net/>. Outra implementação famosa é a MyFaces da Apache Software Foundation em <http://myfaces.apache.org/>.

7

## Java Server Faces (JSF)

O JSF usa XHTML (HTML extensível) como tecnologia para as views do JSF, que por sua vez, é bem mais rígida que as JSPs e HTML básico ao não permitir erros de fechamento de tags, ou ausência de atributos obrigatórios, por exemplo.

O controlador é um servlet chamado Faces Servlet. Ele é responsável por ligar a view ao model, fazendo esse gerenciamento de capturar as requisições, interpretar e devolver a resposta para a view.

8

## XHTML

**XHTML (eXtensible Hypertext Markup Language)** - reformulação da linguagem de marcação HTML, baseada em XML. Combina as tags de marcação HTML com regras da XML. Necessária a utilização do **Doctype** - indica para o navegador, qual especificação do código se deve utilizar. No HTML5 não especifica um DTD, como nas versões anteriores. A responsabilidade do navegador saber qual especificação usar. No XHTML pode-se usar 3 tipos de doctypes:

**Strict** - Para escrever código limpo, livre de misturas entre estilo e conteúdos, 100% xhtml.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
```

**Transitional** - Junta elementos estruturais com elementos que o W3C quer tornar obsoleto na linguagem. Usado para manter a compatibilidade com browsers.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

**Frameset** - Usa-se em documentos que contenham frames.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Frameset//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-frameset.dtd">
```

9

## XHTML

### Principais diferenças entre HTML e XHTML

- ✓ Não esquecer de fechar tags: Ex.: <p> </p> ou Tags "Solitárias" Ex.: <br />
- ✓ Usar letras minúsculas, é case sensitive para tags e atributos Ex.: <div><p>Aqui um texto</p></div>
- ✓ Se não houver conteúdo a tag deve ser autofechada. Ex.: <hr /> <br />
- ✓ Não esquecer das aspas nos atributos Ex.: <td rowspan="3"> <input checked="checked" />
- ✓ Tags HTML, HEAD e BODY são de inclusão obrigatória.
- ✓ As tags devem estar convenientemente aninhadas. Exemplo:  
Errado: <div><em><p>Aqui um texto negrito</em></p></div>  
Certo: <div><em><p>Aqui um texto negrito</p></em></div>

Ver: <http://www.tutorialspoint.com/xhtml/>  
[http://www.w3schools.com/html/html\\_xhtml.asp](http://www.w3schools.com/html/html_xhtml.asp)

10

## XHTML

```
<!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:ui="http://java.sun.com/jsf/facelets"
xmlns:h="http://java.sun.com/jsf/html"
xmlns:f="http://java.sun.com/jsf/core">
```

Declaração  
para  
os  
namespaces

```
<h:head>
<h:outputStylesheet library="css" name="estilo.css"/>
<title>Olá</title>
</h:head>
<h:body>
<h1>Teste Exemplo JSF</h1>
<h:graphicImage library="images" name="imagem1.jpg" />
</h:body>
</html>
```

11

## Diferenças entre JSP e JSF

A rigor são tecnologias de escopo distinto: JSP (Java Server Pages) é framework para construir visões ou uma tecnologia que ajuda os desenvolvedores de software a criarem páginas web geradas dinamicamente baseadas em HTML, XML ou outros tipos de documentos. Enquanto JSF é framework para toda a camada de apresentação, baseado em (mas não limitado a) MVC.

Mas na prática JSF substitui o JSP, já que junto com o JSF 2.0 veio o Facelets, este sim com o objetivo de substituir o JSP pois também é um framework para construção de visões e é todo orientado ao JSF.

12

## Eclipse

Eclipse é um IDE para desenvolvimento Java, porém suporta várias outras linguagens a partir de plugins como C/C++, PHP, ColdFusion, Python, Scala e plataforma Android. Ele foi feito em Java e segue o modelo open source de desenvolvimento de software

Download: <https://eclipse.org/downloads/>

13

## Servidores de Aplicação

Como fazer o "download do Java EE"? O Java EE é apenas um "grande PDF", uma especificação, detalhando quais especificações fazem parte deste. Para usar o software, é necessário fazer o download de uma implementação dessas especificações.

Um servidor de aplicação (Application Server) é um servidor/serviço que disponibiliza um ambiente para a instalação e execução de certas aplicações.

A própria Sun/Oracle desenvolve uma dessas implementações, o Glassfish que é open source e gratuito.

14

## Servidores de Aplicação

Exemplo de alguns dos servidores de aplicação (necessário verificar qual deles é compatível com a versão do Java utilizada):

- ✓ Oracle/Sun, GlassFish Server Open Source Edition 4.0, gratuito;
- ✓ RedHat, JBoss Application Server 7.x, gratuito
- ✓ Oracle/BEA, Oracle WebLogic Server 8.x
- ✓ IBM, IBM WebSphere Application Server
- ✓ SAP, SAP NetWeaver Application Server ou SAP Web Application Server, Java EE 6 Web Profile
- ✓ Apache TomCat 8.x, gratuito

15

## Servidor de Aplicação - Apache Tomcat

- ✓ Implementação de referência para Java
- ✓ Escrito em Java
- ✓ Devido à sua simplicidade e fácil instalação, é comumente utilizado em cursos para desenvolvimento web

Download: <https://tomcat.apache.org/download-60.cgi>

16

## Softwares que serão utilizados (1)

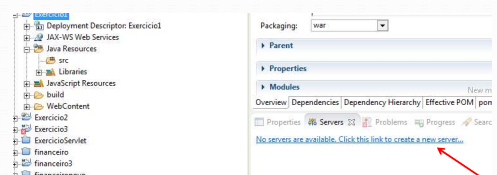
✓ **Copiar** os seguintes arquivos do público para a sua área de desktop (de preferência), e **descompactá-los**:

- ✓ **Apache TomCat**
- ✓ **Eclipse**
- ✓ **DB (Derby – um gerenciador de banco de dados simples)**

17

## Softwares que serão utilizados (2)

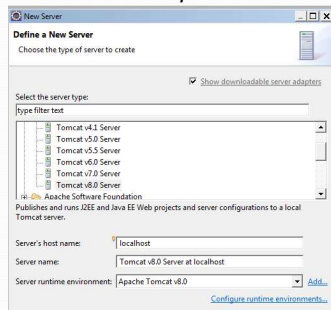
Criar um novo servidor Apache no Eclipse, clicar na opção indicada.



18

## Softwares que serão utilizados (3)

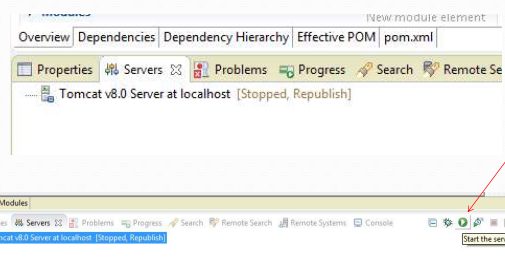
Escolher a versão do Apache e Finish.



19

## Softwares que serão utilizados (4)

O servidor está "parado", clicar em start. Se "startar", Apache está ok.



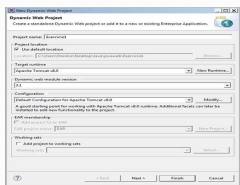
20

## Exercício 1 – Primeiro Projeto Java Web(1)

Disponibilizar como Atividade em no GITHUB.

→ Seu usuário/PWEB/Atividade12

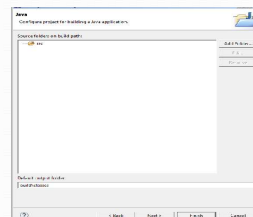
- ✓ File \ New \ **Dynamic Web Project** (utilizado para aplicações Web, possui configurações para o ambiente de servidores).
- ✓ Project Name: **Exercicio1**
- ✓ Selecionar: **Dynamic web module version: 3.1**
- ✓ Next



21

## Exercício 1 – Primeiro Projeto Java Web(2)

- ✓ Manter a mesma estrutura e
- ✓ Next

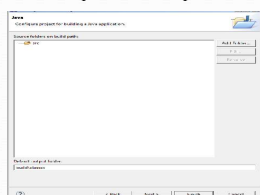


22

## Exercício 1 – Primeiro Projeto Java Web (3)

- ✓ Nessa tela **selecionar Generate web.xml ...** e Finish.

→ O arquivo web.xml serve para fazer diversas configurações, por exemplo definir a página principal da aplicação (quando selecionada essa opção será gerado o "esqueleto" do arquivo web.xml).



23

## Exercício 1 – Primeiro Projeto Java Web (4)

- ✓ Estrutura criada:



**src** - códigos fonte Java (.java)

**build** - onde o Eclipse compila as classes (.class). (outro tipo de projeto essas classes seriam compiladas na pasta bin)

**libraries** - bibliotecas utilizadas na aplicação, Apache, JRE, ...

Há uma pasta **WebContent** que por padrão é a pasta raiz, o que estiver nessa pasta poderá acessado na URL do projeto, por exemplo a página principal (html,xhtml.jsp,jsf), arquivos imagens e etc. Dentro dela há outra pasta chamada **WEB-INF**, que é muito importante pois contém configurações e recursos necessários para o projeto rodar.

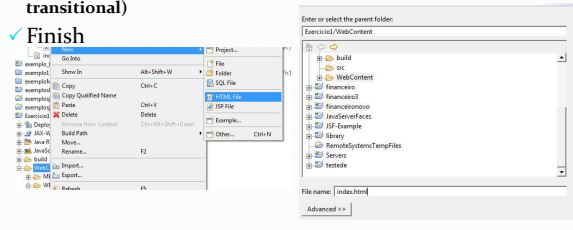
**WebContent/WEB-INF/lib/** - bibliotecas .jar

\*\* No Java EE, existe o conceito de **contextos Web** para diferenciar sites ou projetos distintos em um mesmo servidor, é como se fosse uma pasta virtual, por padrão esse contexto usa o mesmo nome do projeto, no caso Exercício. Mas se por algum motivo precisar alterar clicar em Project / Properties / Web Project Settings e mudar a opção **Context Root**.

24

## Exercício 1 – Primeiro Projeto Java Web (5)

- ✓ Botão direito em cima da pasta **WebContent** e escolher **New\HTML\File**
- ✓ Escolher o nome: **index.html** (escolher opção **transitional**)
- ✓ **Finish**



25

## Exercício 1 – Primeiro Projeto Java Web (6)

- ✓ Inclua o **title** e o **h1** no arquivo **index.html** e salve esse arquivo.

```

1 <!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www
2 <html>
3 <head>
4 <meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
5 <title>Primeiro Projeto</title>
6 </head>
7 <body>
8 <h1>Boa noite!!!</h1>
9 </body>
10 </html>

```

**\*\* ctrl + shift + f – ideta o código**

26

## Exercício 1 – Primeiro Projeto Java Web (7)

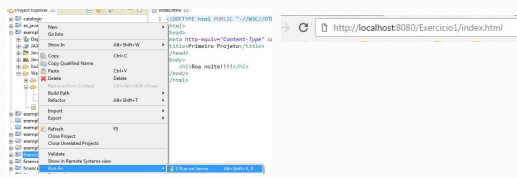
**Para executar :**

**Opção 1:**

- ✓ Botão direito em cima de Exercício1
- ✓ Run as \ Run on Server \ Escolher o Tomcat e Finish

**Opção 2:**

- ✓ No navegador digite: <http://localhost:8080/Exercicio1/index.html>



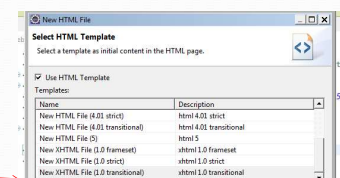
27

## Exercício 2 – Segundo Projeto(1)

Disponibilizar como Atividade em no GITHUB.

→ Seu usuario/PWEB/Atividade13

- ✓ Os procedimentos são os mesmos do exercícios anterior até a parte 4. No item 5, quando for criar a página principal ao invés de escolher formato HTML, escolher **XHTML**.
- ✓ Botão direito em cima da pasta **WebContent** e escolher **New\HTML\File**
- ✓ Escolher o nome: **index.xhtml**
- ✓ NEXT
- ✓ Selecciona New XHTML File (1.0 Transitional)
- ✓ Finish



28

## Exercício 2 – Segundo Projeto (2)

- ✓ Acertar o conteúdo do arquivo **index.xhtml**
- ✓ Inserir **title** e **h1**

```

1 <?xml version="1.0" encoding="UTF-8" ?>
2 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3
3 <html xmlns="http://www.w3.org/1999/xhtml">
4 <head>
5 <meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1" />
6 <title>Segundo Projeto</title>
7 </head>
8 <body>
9 <h1>Olha aqui o título da Página</h1>
10 </body>
11 </html>

```

**\*\* O elemento raiz do documento deve conter (recomendação) uma declaração xmlns para o namespace(para não acontecer ambiguidades) em XHTML. O namespace em XHTML é definido como: <http://www.w3.org/1999/xhtml>. O cabeçalho de tipo Content-Type é usado para indicar o tipo de mídia do recurso.**

29

## Exercício 2 – Segundo Projeto (3)

- ✓ Acertar o conteúdo do arquivo **web.xml** (Web Content\Web-INF), incluindo também o **index.xhtml**.

```

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 http://
5 id="WebApp_ID" version="3.1">
6 <display-name>Exercicio2</display-name>
7 <welcome-file-list>
8 <welcome-file>index.xhtml</welcome-file>
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
17 </web-app>

```

30



## Exercício 2 – Segundo Projeto(4)

- ✓ Formas de Execução
- ✓ Botão direito em cima de Exercicio2 e Run as \ Run on Server \ Escolher o Tomcat e Finish
- Para acessar o projeto pelo navegador use o endereço:  
**<http://localhost:8080/Exercicio2/>**

31

## MAVEN

O Maven é uma ferramenta de gerenciamento, construção e implantação de projetos, que ajuda no processo de gerenciamento de dependências e no de build, geração de relatórios e de documentação.

Existem outros por ex. Gradle (utilizado no desenvolvimento Android), Ant

32

## MAVEN (POM.XML – 1)

A unidade básica de configuração do Maven é um arquivo chamado **pom.xml** (*Project Object Model*), que deve ficar na raiz do seu projeto. Nele é declarada a estrutura, dependências e características do projeto, ele fica na raiz do projeto para poder chamar as targets de build do projeto.

**Download:**  
**<https://http://maven.apache.org/download.cgi?Preferred=http%3A%2F%2Fftp.unicamp.br%2Fpub%2Fapache%2F#>**

33

## MAVEN (POM.XML - 2)

```
<project>
  <modelVersion>4.0.0</modelVersion>
  <groupId>br.com.javaparaweb</groupId>
  <artifactId>teste</artifactId>
  <version>1.0</version>
</project>
```

- ✓ Contém apenas a identificação do projeto, e uma informação a mais: modelVersion, que é a identificação da versão do arquivo pom.xml e deve ser sempre 4.0.0. A identificação do projeto consiste em três informações:
  - ✓ groupId: um identificador da empresa/grupo ao qual o projeto pertence. Geralmente o nome do site da empresa/grupo ao contrário. Exs.: br.com.javaparaweb, br.fatec.javaparaweb
  - ✓ artifactId: o nome do projeto. Ex: teste.
  - ✓ version: a versão atual do projeto. Ex: 1.0-SNAPSHOT.

34

## MAVEN - Dependências

Essas informações são usadas em muitos lugares, como o controle de dependências. Por exemplo, para dizer que o MYSQL é uma dependência da sua aplicação é só acrescentar no pom as linhas:

```
<dependency>
  <groupId>mysql</groupId>
  <artifactId>mysql-connector-java</artifactId>
  <version>5.1.38</version>
</dependency>
```

**\*\* Quando necessário, o Maven vai baixar o jar do MYSQL 5.1.38, e todas as suas dependências, e vai colocá-las no classpath da aplicação durante os builds, testes, etc. Ou seja, não precisa entrar no site do MYSQL, baixar um zip com vários jars e ter que procurar quais jars devem ser colocados no classpath.**

35

## MAVEN (Servlets)

No mundo dos Servlets, o mapeamento é feito num arquivo especial, chamado web.xml e que deve ficar localizado dentro do diretório WEB-INF.

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns="http://xmlns.jcp.org/xml/ns/javaee"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee
    http://xmlns.jcp.org/xml/ns/javaee/web-app_3_1.xsd"
  version="3.1">
  <display-name>JavaServerFaces</display-name>
  <!-- Change to "Production" when you are ready to deploy -->
  <context-param>
    <param-name>javax.faces.PROJECT_STAGE</param-name>
    <param-value>Development</param-value>
  </context-param>
  <!-- Welcome page -->
  <welcome-file-list>
    <welcome-file>default.xhtml</welcome-file>
  </welcome-file-list>
```

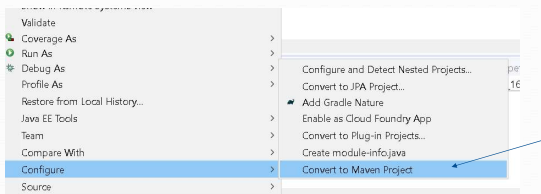
Em um projeto MAVEN, a página \*.html ou \*.xhtml deve ficar dentro da pasta webapp

36

## MAVEN (Servlets)

Se o projeto foi criado como Dynamic Project é possível convertê-lo para Maven.

Usar botão direito mouse\



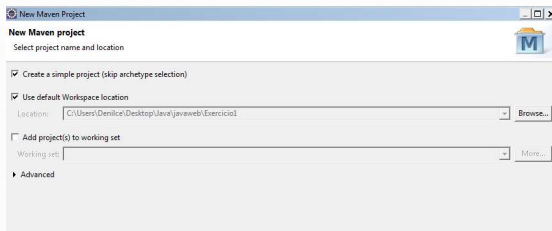
37

## Exercício 3 – Terceiro Projeto Java Web (1)

Disponibilizar como Atividade no GITHUB.

→ Seu usuário/PWEB/Atividade4

- ✓ File \ New \ Maven Project
- ✓ Project Name: Exercício3
- ✓ Selecionar: Create a simple ...
- ✓ Next

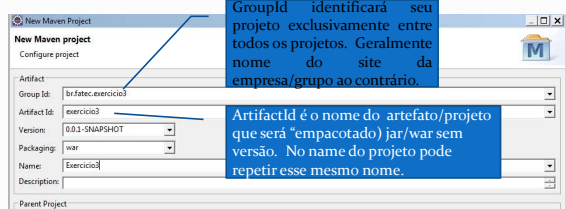


38

## Exercício 3 – Terceiro Projeto Java Web (2)

- ✓ Group id -> br.fatec.exercicio3 (padrão: domínio+nome+o que faz)
- ✓ Artifact id -> Exercício3
- ✓ Packaging -> war → TEM QUE SER ESSA EXTENSÃO PARA PODER RODAR NO SERVIDOR DE APLICAÇÃO (WEB APPLICATION RESOURCE)

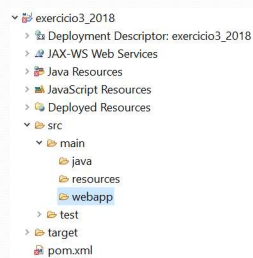
✓ Name: Exercício3



39

## Exercício 3 – Terceiro Projeto Java Web (3)

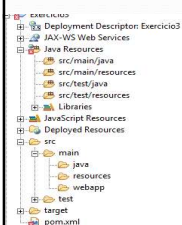
✓ Estrutura do projeto Maven



40

## Exercício 3 – Terceiro Projeto Java Web (8)

✓ A estrutura do projeto Maven é diferente.



Java Resources: nessa pasta onde fica a estrutura (própria) do Maven.

src/main/java - Diretório onde está o código fonte Java da Aplicação e/ou Biblioteca

src/main/resources - Arquivos de configuração e outros arquivos devem ficar nesta pasta

src/main: nessa pasta ficam uma cópia dos arquivos do src/main/java (exibidas como elas são exibidas na estrutura de pastas).

src/main/webapp - Pasta para conteúdo web. Aqui se encontra todos os arquivos fonte da parte web (html/jsp/xhtml)

pom.xml: para um projeto maven, esse é o arquivo onde ficam todas as configurações pertinente a geração de build, testes, jars de terceiros, etc;

target: é o diretório onde fica tudo que é gerado: JARs, WARs, etc.

41

## Exercício 3 – Terceiro Projeto Java Web (4)

✓ No pom.xml aparecem os dados da aplicação



42

## Exercício 3 – Terceiro Projeto Java Web (5)

✓ **POM.XML** - Incluir as dependências e repositórios (repositórios acessados pelo Maven para baixar as dependências)

```
<project xmlns="http://maven.apache.org/POM/4.0.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/
xsd/maven-4.0.0.xsd">
<modelVersion>4.0.0</modelVersion>
<groupId>br.fatec.exercicio3</groupId>
<artifactId>Exercicio3</artifactId>
<version>0.0.1-SNAPSHOT</version>
<packaging>war</packaging>
<name>Exercicio3</name>

<dependencies><!-- para funcionar o JavaServer Faces -->
<dependency>
<groupId>javax</groupId>
<artifactId>javaee-api</artifactId>
<version>8.0</version>
<scope>provided</scope>
</dependency>
```

Você encontra as informações sobre as dependências na web, depende da dependência que irá incluir.

43

## Exercício 3 – Terceiro Projeto Java Web (6)

```
<dependency>
<groupId>org.glassfish</groupId>
<artifactId>javax.faces</artifactId>
<version>2.2.13</version>
</dependency>
</dependencies>

<build>
<plugins>
<plugin><!-- para projeto vinculado ao java 8, por default é java 5 -->
<groupId>org.apache.maven.plugins</groupId>
<artifactId>maven-compiler-plugin</artifactId>
<version>3.2</version>
<configuration>
<source>1.8</source>
<target>1.8</target>
</configuration>
</plugin>
```

44

## Exercício 3 – Terceiro Projeto Java Web (7)

```
<plugin><!-- plugin para considerar os arquivos fonte padrao utf8 -->
<artifactId>maven-resources-plugin</artifactId>
<version>2.7</version>
</plugin>
</plugins>
</build>
<properties><!-- acentuação no projeto -->
<project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
</properties>

<repositories>
<!-- o repositório que será acessado para baixar dependências pode ser local ou remoto,
no caso está sendo usado remoto -->
<repository>
<id>java.net</id>
<url>https://maven.java.net/content/repositories/public/</url>
</repository>
</repositories>
</project>
```

\*\* Onde fica o repositório local: C:\Users\USUARIO\.m2\repository\javax\javaee-api\8.0

45

## Exercício 3 – Terceiro Projeto Java Web (9)

- ✓ Botão direito em cima da pasta **scr/main/webapp**
- ✓ File/New
- ✓ Criar arquivo: **index.xhtml**
- ✓ O código está no próximo slide

46

## Exercício 3 – Terceiro Projeto Java Web (10)

Arquivo **INDEX.xhtml**:

```
<!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">

<!-- As tags que representam os componentes do JSF estão em duas taglibs principais
(bibliotecas de tags): a core e a html.
A taglib html contém os componentes necessários
para montarmos nossa tela gerando o HTML adequado. Já a core possui diversos
componentes não visuais, como tratadores de eventos ou validadores.
Por ora, usaremos apenas os componentes da h:html
-->
<h:head>
<title>Testando Maven</title>
</h:head>
<h:body>
<h:outputText value="Olá, estamos usando o Maven..." />
</h:body>
</html>
```

47

## Exercício 3 – Terceiro Projeto Java Web (11)

- ✓ Botão direito em cima da pasta **scr/main/webapp**
- ✓ File/New/Folder
- ✓ Criar folder: **WEB-INF**
- ✓ Dentro desse folder criar o arquivo (new/file) **web.xml**
- ✓ O código está no próximo slide.

48



## Exercício 3 – Terceiro Projeto Java Web (12)

## ARQUIVO WEB.XML:

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="http://xmlns.jcp.org/xml/ns/javaee"
xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee
http://xmlns.jcp.org/xml/ns/javaee/web-app_3_1.xsd"
id="WebApp_ID" version="3.1">
<display-name>Exercicio3</display-name>

<!-- JSF mapping , O elemento servlet-class declara javax.faces.webapp.FacesServlet
fazendo com que este seja carregado inicialmente. -->

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

49

## Exercício 3 – Terceiro Projeto Java Web (13)

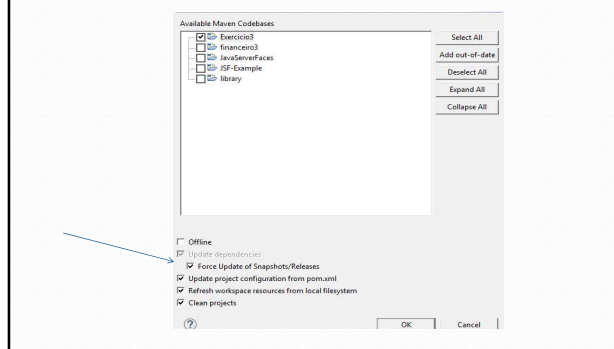
## Continuação ARQUIVO WEB.XML:

```
<welcome-file-list>
<welcome-file>index.xhtml</welcome-file>
</welcome-file-list>
</web-app>
```

50

## Exercício 3 – Terceiro Projeto Java Web (14)

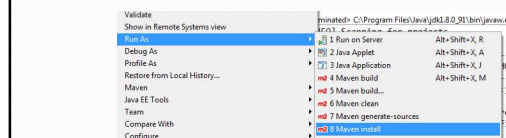
- ✓ Botão direito em cima do nome do projeto: **Maven** -> Update Project, escolher nome do projeto, Force ... e ok.



51

## Exercício 3 – Terceiro Projeto Java Web (15)

- ✓ Botão direito em cima do nome do projeto **Run as Maven Install**, para gerar o arquivo e se tiver algum erro, será mostrado no console.

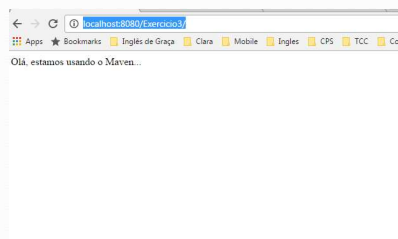


52

## Exercício 3 – Terceiro Projeto Java Web (16)

## Para executar:

- ✓ Botão direito em cima do nome do projeto **Run as On Server** ou  
✓ No Navegador <http://localhost:8080/Exercicio3/>



53

## Referências

- ✓ ANOTACOES. Disponível em: <http://www.devmedia.com.br/mapeamento-no-hibernate-com-anotacoes/29472>> Acesso: 30.Mai.2016.
- ✓ BERNARDO, Sérgio. Tutorial T5Tutorial do framework Tapestry5. Fatec Sorocaba. Setembro/2015.
- ✓ CAELUM. Apostila do curso FJ-21 - Java para Desenvolvimento Web. Disponível em: <https://www.caelum.com.br/apostila-java-web/> Acesso:30.Mai.2016
- ✓ HEINZELMANN, Décio L. MELO, Alexandre Altair de. Programação Java para Web. 2ª Ed. Novatec: São Paulo.
- ✓ HIBERNATE. Disponível em: <http://www.tutorialspoint.com/hibernate/>> Acesso: 30.Mai.2016.
- ✓ JSFJSP. Disponível em: <http://pt.stackoverflow.com/questions/81228/diferen%C3%A7a-entre-jsf-e-jsp>> Acesso: 30.Mai.2016.

54

## Referências

- ✓ MAPPING. Disponível em: <<http://www.mauda.com.br/?p=581>> Acesso: 30.Mai.2016.
- ✓ MAVEN. Disponível em: [http://www.w3ii.com/pt/maven/maven\\_repositories.html](http://www.w3ii.com/pt/maven/maven_repositories.html) Acesso: 30.Mai.2016.
- ✓ SILVA, Luiz Ricardo. Disponível em: <<http://luizricardo.org/2014/06/instalando-configurando-e-usando-o-maven-para-gerenciar-suas-dependencias-e-seus-projetos-java>> Acesso: 30.Mai.2016.
- ✓ XHTML. Disponível em: <<https://web.fe.up.pt/~mgioooo9/mestrado/goii/XHTML/criar-docs-xhtml.html>> Acesso: 29.Mai.2016
- ✓ WEBXML. Disponível em: <<http://www.mkyong.com/web-development/the-web-xml-deployment-descriptor-examples/>> Acesso: 29.Mai.2016