



Curso de Especialização em Desenvolvimento Java

Desenvolvimento WEB com JAVA

Prof. Emílio Dias
emiliodias@gmail.com
<http://www.github.com/emiliodias>

Conteúdo da disciplina

Hands-on:

Desenvolvendo um blog pessoal

Revisão sobre MVC

Revisão Geral

MVC component based vs MVC action based

Apresentação de alguns frameworks MVC Java

Struts, Spring MVC, JSF, Apache Wicket, Google Web Toolkit – GWT,
Thymeleaf, Freemaker, Velocity.

Spring MVC + Thymeleaf

Discussões gerais sobre segurança

Autenticação com Spring Security

Discussões conceituais gerais e pragmáticas sobre a Web

Web Services, REST

Avaliação

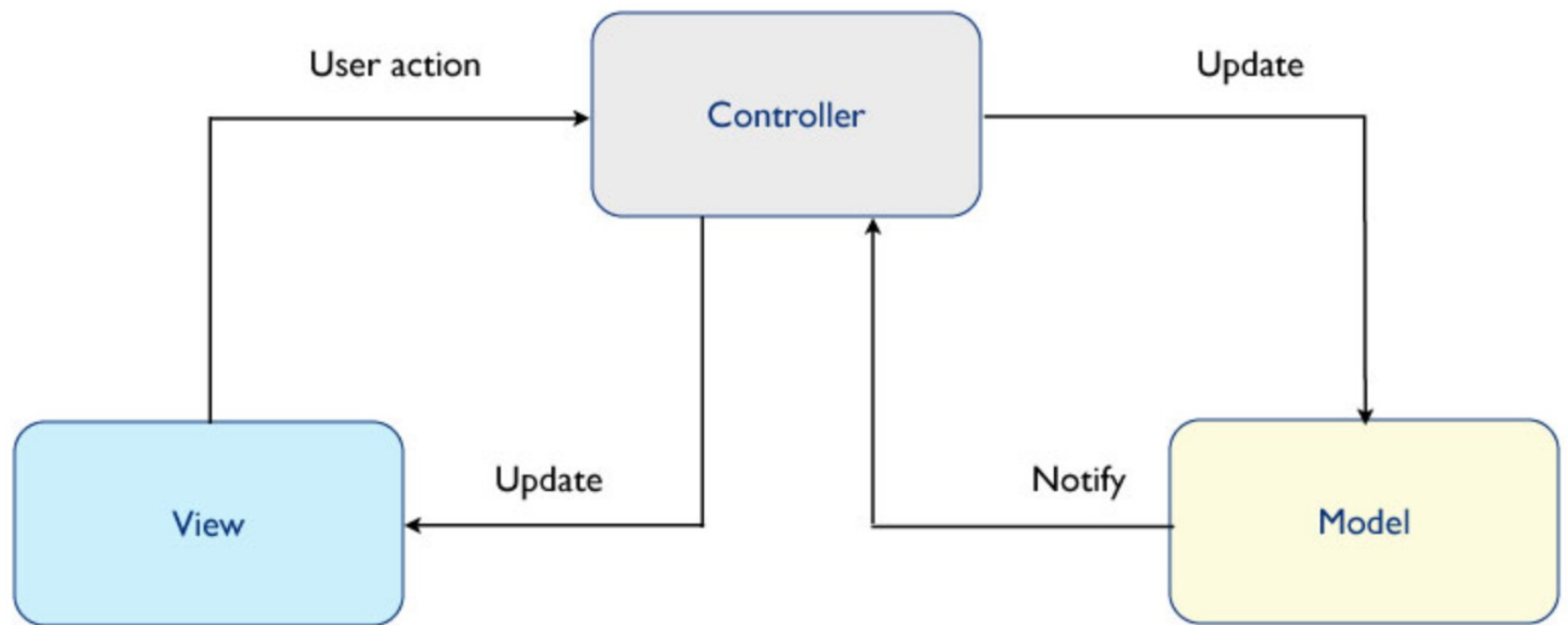
- 100 pontos baseados em:
 - Entrega de um projeto final utilizando as tecnologias discutidas durante a disciplina.

Hands-on

Hands-on

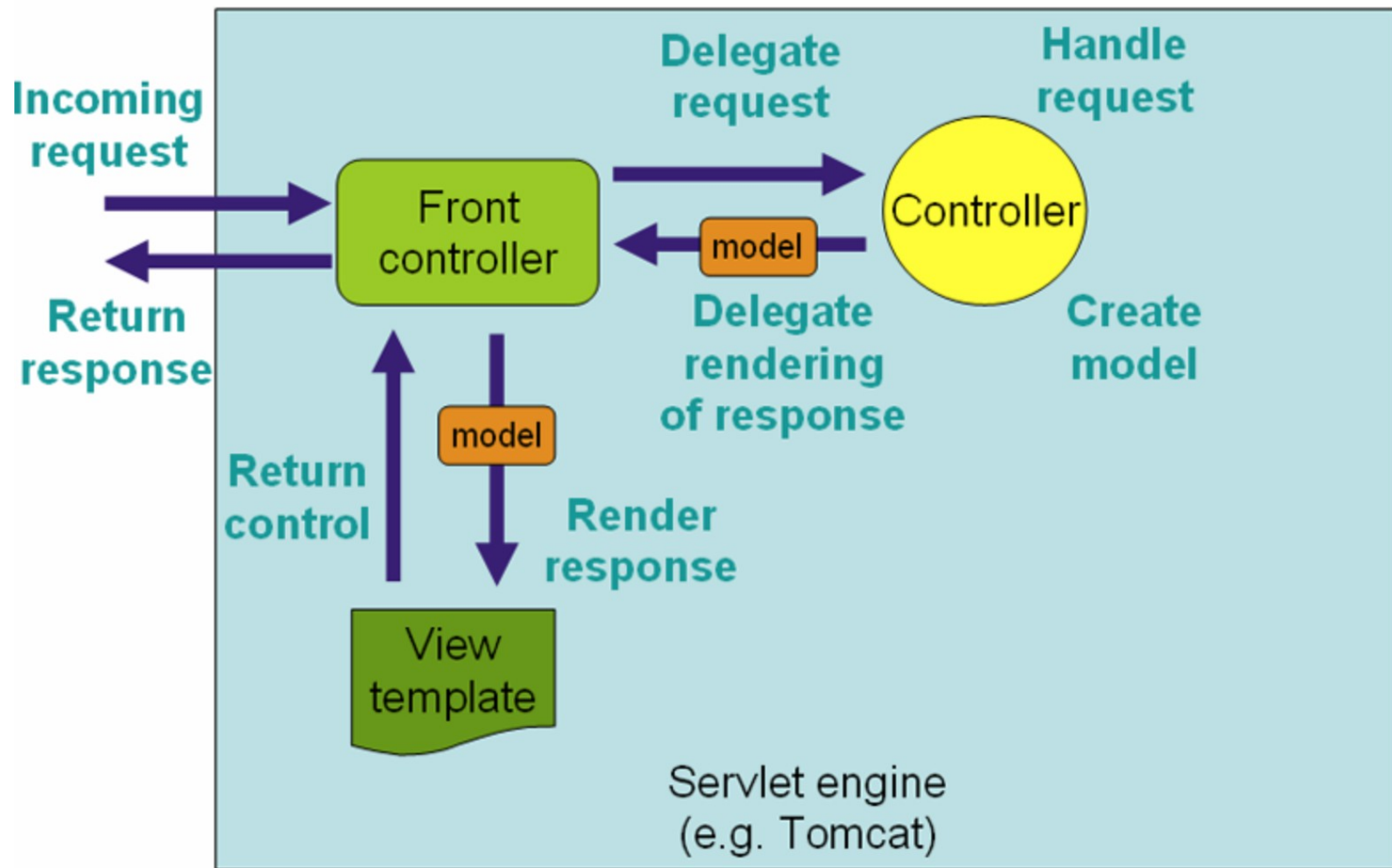
- O conteúdo deste tópico pode ser encontrado em:
 - <https://github.com/emiliodias/posjava/tree/master/blog>

MVC



Spring MVC

Spring MVC

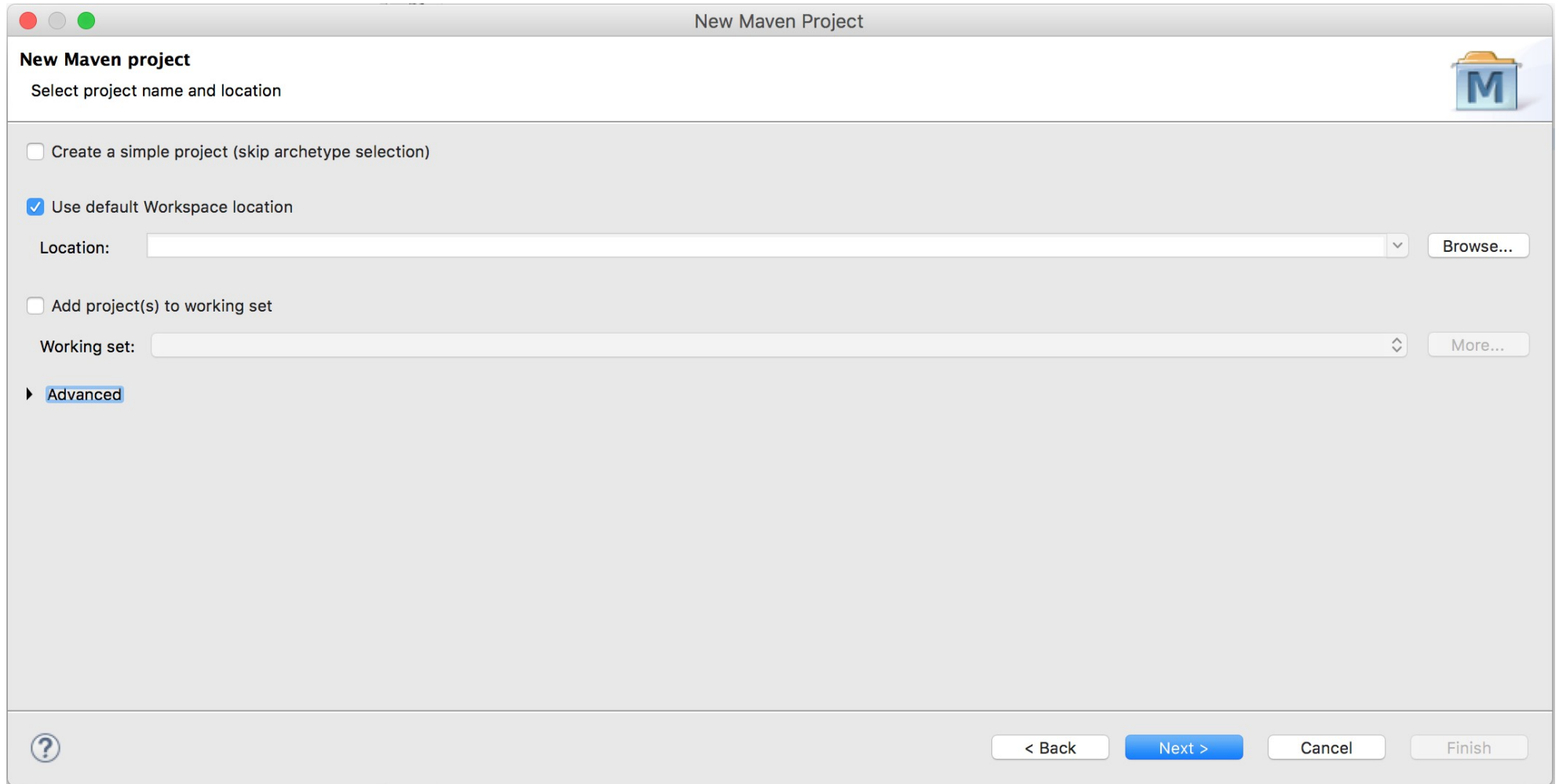


Criando seu projeto Spring-MVC

Criando seu projeto Spring-MVC

- Pré-requisitos:
 - Apache Tomcat ≥ 7
 - Apache Maven
 - Eclipse

Novo projeto Maven



The image shows a 'New Maven Project' dialog box with a title bar containing standard macOS window controls (red, yellow, green buttons) and the text 'New Maven Project'. The main content area is titled 'New Maven project' and has a subtitle 'Select project name and location'. In the top right corner of the main area is a small icon of a folder with a blue 'M' on it. The main area contains several options: a checkbox 'Create a simple project (skip archetype selection)' which is unchecked; a checkbox 'Use default Workspace location' which is checked; a 'Location:' label followed by a text input field and a 'Browse...' button; a checkbox 'Add project(s) to working set' which is unchecked; a 'Working set:' label followed by a text input field and a 'More...' button; and a section header 'Advanced' preceded by a right-pointing triangle icon. At the bottom of the dialog is a footer bar with a question mark icon on the left and four buttons on the right: '< Back' (disabled), 'Next >' (active/highlighted in blue), 'Cancel' (disabled), and 'Finish' (disabled).

New Maven Project

New Maven project
Select project name and location

☐ Create a simple project (skip archetype selection)

☒ Use default Workspace location

Location:

☐ Add project(s) to working set

Working set:

▶ **Advanced**

Selezione o archetype web-app

New Maven Project

Select an Archetype

Catalog: All Catalogs Configure...

Filter: web ✕

Group Id	Artifact Id	Version
org.apache.maven.archetypes	maven-archetype-webapp	1.0

☒ Show the last version of Archetype only ☐ Include snapshot archetypes Add Archetype...

► Advanced


? < Back Next > Cancel Finish

Configure os parâmetros adequados

New Maven Project

New Maven project

Specify Archetype parameters



Group Id:

br.com.unitri.desenvolvimentoweb

▼

Artifact Id:

springmvc

▼

Version:

0.0.1-SNAPSHOT

▼

Package:

br.com.unitri.desenvolvimentoweb.springmvc

▼


Properties available from archetype:

Name	Value	

Add...

Remove

► Advanced



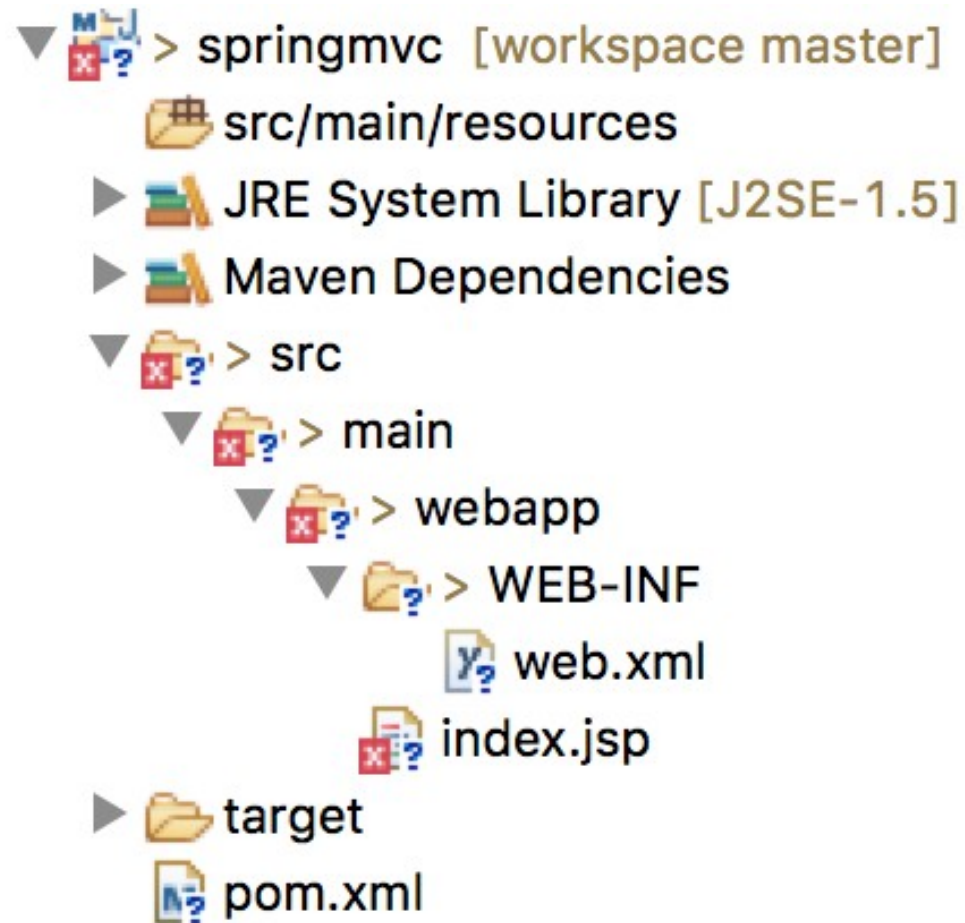
< Back

Next >

Cancel

Finish

Estrutura do seu projeto



pom.xml

```
1 <project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
2   xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/maven-v4_0_0.xsd">
3   <modelVersion>4.0.0</modelVersion>
4   <groupId>br.com.unitri.desenvolvimentoweb</groupId>
5   <artifactId>springmvc</artifactId>
6   <packaging>war</packaging>
7   <version>0.0.1-SNAPSHOT</version>
8   <name>springmvc Maven Webapp</name>
9   <url>http://maven.apache.org</url>
10  <dependencies>
11    <dependency>
12      <groupId>junit</groupId>
13      <artifactId>junit</artifactId>
14      <version>3.8.1</version>
15      <scope>test</scope>
16    </dependency>
17  </dependencies>
18  <build>
19    <finalName>springmvc</finalName>
20  </build>
21 </project>
22
```


Adicione a dependência do spring-webmvc

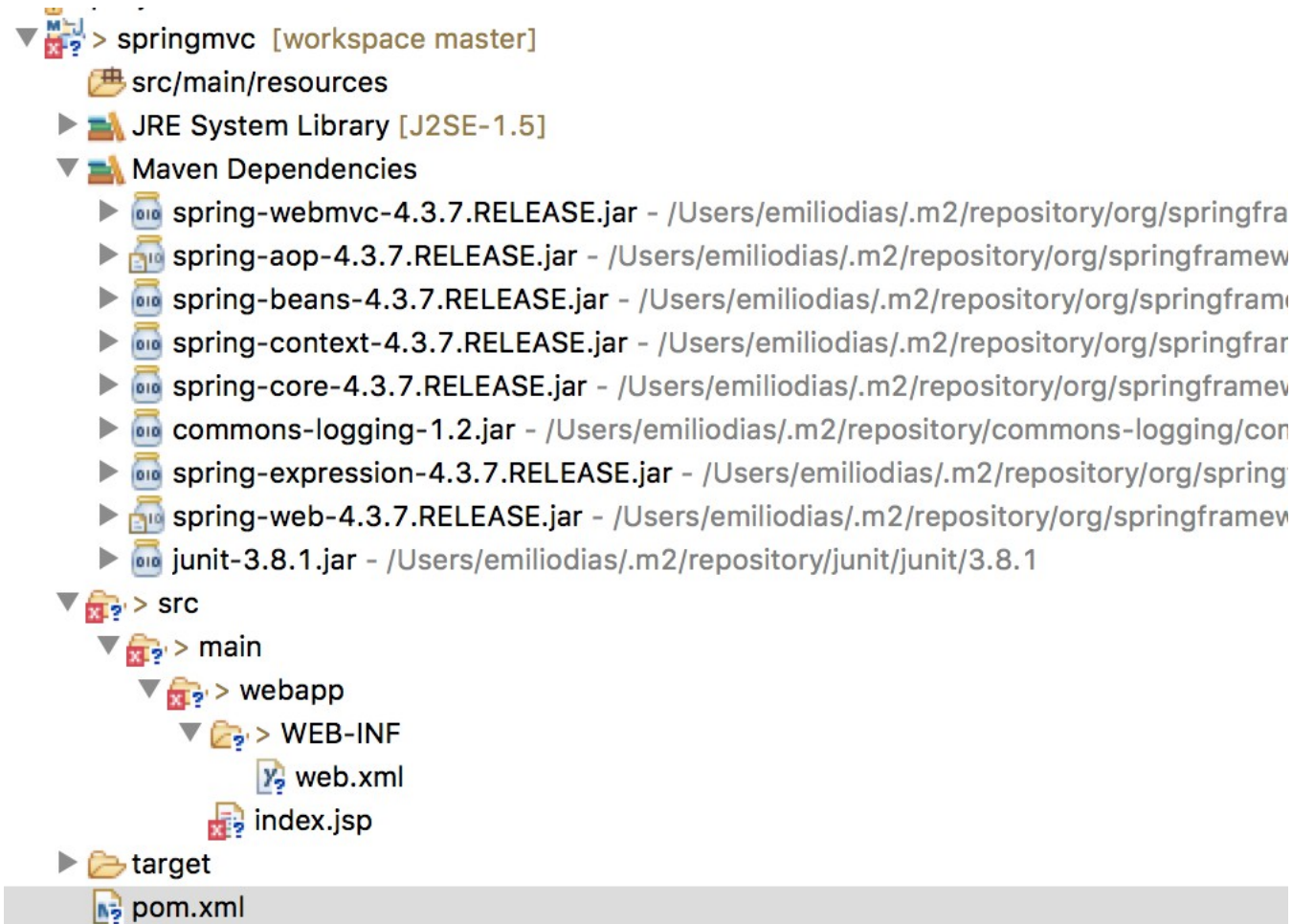
```
1 <project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
2   xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/maven-v4_0_0.xsd">
3   <modelVersion>4.0.0</modelVersion>
4   <groupId>br.com.unitri.desenvolvimentoweb</groupId>
5   <artifactId>springmvc</artifactId>
6   <packaging>war</packaging>
7   <version>0.0.1-SNAPSHOT</version>
8   <name>springmvc Maven Webapp</name>
9   <url>http://maven.apache.org</url>
10
11   <properties>
12     <spring.version>4.3.7.RELEASE</spring.version>
13   </properties>
14
15   <dependencies>
16
17     <dependency>
18       <groupId>org.springframework</groupId>
19       <artifactId>spring-webmvc</artifactId>
20       <version>${spring.version}</version>
21     </dependency>
22
23     <dependency>
24       <groupId>junit</groupId>
25       <artifactId>junit</artifactId>
26       <version>3.8.1</version>
27       <scope>test</scope>
28     </dependency>
29
30   </dependencies>
31
32   <build>
33     <finalName>springmvc</finalName>
34
35     <plugins>
36       <plugin>
37         <artifactId>maven-compiler-plugin</artifactId>
38         <version>2.3.2</version>
39         <configuration>
40           <source>1.7</source>
41           <target>1.7</target>
42           <encoding>UTF-8</encoding>
43         </configuration>
44       </plugin>
45     </plugins>
46
47   </build>
48
49 </project>
```

É bom deixar explícito a versão de compilação que você deseja usar no seu projeto.

```
<dependency>
<groupId>org.springframework</groupId>
<artifactId>spring-webmvc</artifactId>
<version>${spring.version}</version>
</dependency>
```

```
<plugins>
<plugin>
<artifactId>maven-compiler-
plugin</artifactId>
<version>2.3.2</version>
<configuration>
<source>1.7</source>
<target>1.7</target>
<encoding>UTF-8</encoding>
</configuration>
</plugin>
</plugins>
```

Dependências adicionadas ao seu classpath



Configure seu web.xml

Adicione o conteúdo abaixo entre as tags <web-app>

```
<listener>
<listener-class>org.springframework.web.context.ContextLoaderListener</listener-class>
</listener>

<context-param>
<param-name>contextConfigLocation</param-name>
<param-value>/WEB-INF/classes/spring/ApplicationContext.xml</param-value>
</context-param>

<servlet>
<servlet-name>DispatcherServlet</servlet-name>
<servlet-class>org.springframework.web.servlet.DispatcherServlet</servlet-class>
<init-param>
<param-name>contextConfigLocation</param-name>
<param-value></param-value>
</init-param>
<load-on-startup>1</load-on-startup>
</servlet>

<servlet-mapping>
<servlet-name>DispatcherServlet</servlet-name>
<url-pattern>/</url-pattern>
</servlet-mapping>
```

Configure seu web.xml

Remova o trecho:

```
<!DOCTYPE web-app PUBLIC  
"-//Sun Microsystems, Inc.//DTD Web Application 2.3//EN"  
"http://java.sun.com/dtd/web-app_2_3.dtd" >
```

Configure seu web.xml

Deixe a tag “web-app” conforme abaixo:

```
<web-app 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-app_2_5.xsd"
  Version="2.5">
```

ApplicationContext.xml

Dentro do diretório src/main/resources, faça:

- Crie um folder chamado spring em /src/main/resources.
- Crie um arquivo com o nome ApplicationContext.xml. Copie e cole o conteúdo abaixo:

```
<beans xmlns="http://www.springframework.org/schema/beans"
  xmlns:mvc="http://www.springframework.org/schema/mvc"
  xmlns:context="http://www.springframework.org/schema/context"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

  xsi:schemaLocation="
    http://www.springframework.org/schema/beans
    http://www.springframework.org/schema/beans/spring-beans.xsd
    http://www.springframework.org/schema/mvc
    http://www.springframework.org/schema/mvc/spring-mvc.xsd
    http://www.springframework.org/schema/context
    http://www.springframework.org/schema/context/spring-context.xsd">

  <context:component-scan base-package="br.com.unitri.desenvolvimentoweb" />
  <mvc:annotation-driven />

  <bean class="org.springframework.web.servlet.view.InternalResourceViewResolver" >

    <property name = "prefix" value = "/WEB-INF/views/" />
    <property name = "suffix" value = ".jsp" />

  </bean>

</beans>
```

Estrutura básica de pacotes

- Crie um folder chamado src/main/java
- Dentro do diretório src/main/java, crie um pacote com o nome:

br.com.unitri.desenvolvimentoweb.controllers

HelloWorld.java

No pacote criado anteriormente, crie uma classe com o nome HelloWorld, e copie e cole o conteúdo abaixo:

```
package br.com.unitri.desenvolvimentoweb.controllers;

import org.springframework.stereotype.Controller;
import org.springframework.ui.ModelMap;
import org.springframework.web.bind.annotation.RequestMapping;

@Controller
public class HelloWorld {

    @RequestMapping("/helloworld")
    public String helloWorld(ModelMap model) {

        return "helloWorld";
    }

}
```


HelloWorld.jsp

Crie um diretório chamado views, dentro de src/main/webapp/WEB-INF, e após isso, crie um arquivo chamado HelloWorld.jsp com o conteúdo abaixo:

```
<%@ page contentType="text/html; charset=UTF-8"%>
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>
<html>
<head>
<meta charset="UTF-8">
</head>
<body>

<h2>Hello World Pós Java Unitri!!!</h2>

</body>
</html>
```

Compilando

Execute a compilação do seu projeto pelo próprio Eclipse, ou se preferir, utilize os comandos Maven abaixo em sua linha de comando:

- mvn clean compile package

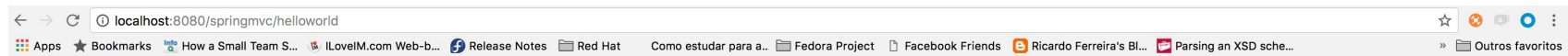
Este comando irá gerar um artefato war, copie-o e cole no diretório webapps do seu Tomcat. Após executar este processo sua aplicação já estará apta para acesso.

Testando

Após o start do Tomcat, abra o navegador de sua preferência e digite:

http://localhost:8080/springmvc/helloworld

Você deverá obter um resultado igual ao abaixo:



Hello World Pós Java Unitri!!!