

Sistemas Distribuídos - Relatório

Universidade Federal do Ceará – Campus de Quixadá

Prática: Introdução ao Spring Boot e Maven

Dupla: Mônica Maria Rodrigues

Maryana Moraes

Professor: Rafael Braga

1. Objetivo

O objetivo da prática foi configurar um ambiente de desenvolvimento em Java utilizando **Spring Boot** e **Maven**, criar um projeto básico, compilar o código-fonte e executar uma aplicação web simples, disponibilizando endpoints HTTP acessíveis via navegador ou linha de comando.

2. Fundamentação Teórica

O **Spring Boot** é um framework Java que facilita o desenvolvimento de aplicações web, oferecendo configuração automática, servidor embutido e gerenciamento simples de dependências.

O **Maven** é uma ferramenta de automação para:

- gerenciamento de dependências;
- Compilação;
- Empacotamento;
- execução de projetos Java.

O arquivo pom.xml define:

- dependências do projeto;
- versão do Java;
- Plugins;
- estrutura de build.

A classe principal anotada com `@SpringBootApplication` inicializa toda a aplicação, enquanto a anotação `@RestController` permite expor endpoints web através de métodos anotados com `@RequestMapping`.

3. Materiais e Métodos

Materiais

- Sistema operacional **Ubuntu**
- **Java JDK**
- **Maven**
- **VS Code**
- PDF da prática (fornecido pelo professor)
- Terminal Linux

Procedimentos Executados

1. Instalação e verificação da versão do Java:

```
java –version
```

```
sudo apt update
```

```
sudo apt install openjdk-11-jdk -y
```

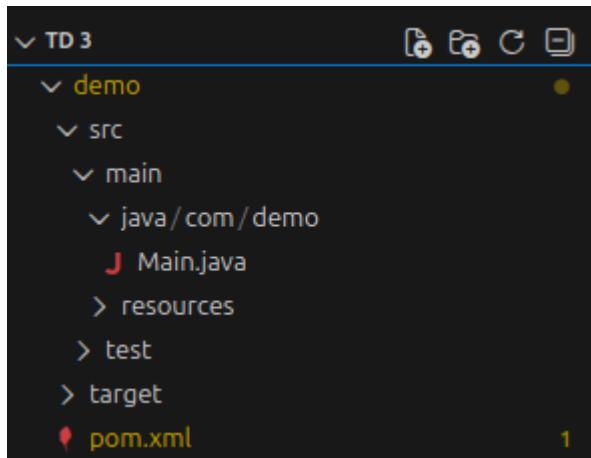
2. Instalação e verificação da versão do Maven:

```
sudo apt update
```

```
sudo apt install maven -y
```

```
mvn -v
```

3. Criação da estrutura do projeto:



4. Criação do arquivo pom.xml contendo as dependências do Spring Boot.

```
pom.xml 1 x J Main.java
demo > pom.xml
1  <?xml version="1.0" encoding="UTF-8"?>
2  <project xmlns="http://maven.apache.org/POM/4.0.0"
3  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-
5  4.0.0.xsd">
6  <modelVersion>4.0.0</modelVersion>
7  <parent>
8  <groupId>org.springframework.boot</groupId>
9  <artifactId>spring-boot-starter-parent</artifactId>
10 <version>2.1.6.RELEASE</version>
11 <relativePath/> <!-- lookup parent from repository -->
12 </parent>
13 <groupId>com.demo</groupId>
14 <artifactId>demo</artifactId>
15 <version>0.0.1-SNAPSHOT</version>
16 <name>demo</name>
17 <description>Demo project for Spring Boot</description>
18 <properties>
19 <java.version>1.8</java.version>
20 <maven-jar-plugin.version>3.1.1</maven-jar-plugin.version>
21 </properties>
22 <dependencies>
23 <dependency>
24 <groupId>org.springframework.boot</groupId>
25 <artifactId>spring-boot-starter-web</artifactId>
26 </dependency>
27 <dependency>
28 <groupId>org.springframework.boot</groupId>
29 <artifactId>spring-boot-starter-test</artifactId>
30 <scope>test</scope>
31 </dependency>
32 </dependencies>
33 <build>
34 <plugins>
35 <plugin>
36 <groupId>org.springframework.boot</groupId>
37 <artifactId>spring-boot-maven-plugin</artifactId>
38 </plugin>
39 </plugins>
40 </build>
41 </project>
```

5. Criação do arquivo Main.java com os endpoints.

The screenshot shows a code editor with two tabs: 'pom.xml' and 'Main.java'. The 'Main.java' tab is active, displaying the following Java code:

```
demo > src > main > java > com > demo > Main.java > Main > main(String[])
  1 package com.demo;
  2 import org.springframework.boot.SpringApplication;
  3 import org.springframework.boot.autoconfigure.SpringBootApplication;
  4 import org.springframework.web.bind.annotation.PathVariable;
  5 import org.springframework.web.bind.annotation.RequestMapping;
  6 import org.springframework.web.bind.annotation.RestController;
  7 @RestController
  8 @SpringBootApplication
  9 public class Main {
 10     @RequestMapping("/")
 11     String home() {
 12         return "Olá, seja bem vindo ao teste de Spring Boot para disciplina de Sistemas Distribuídos";
 13     }
 14     @RequestMapping("/nome/{nome}")
 15     String piada(@PathVariable String nome) {
 16         return "Olá " + nome + ", vocês vão realmente conseguir passar nessa disciplina?";
 17     }
 18     public static void main(String[] args) {
 19         SpringApplication.run(Main.class, args);
 20     }
 21 }
```

6. Compilação do projeto:

```
mvn clean install
```

7. Execução da aplicação:

```
mvn spring-boot:run
```

8. Testes no navegador:

- <http://localhost:8080/>
- <http://localhost:8080/nome/SeuNome>

4. Resultados Obtidos

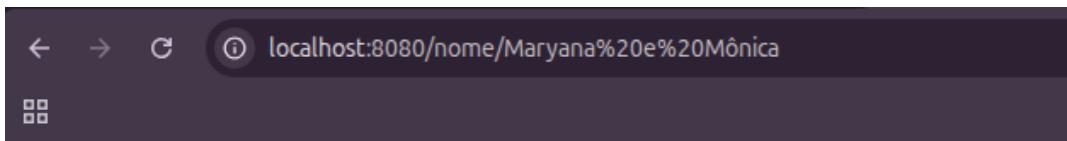
Após a compilação bem-sucedida, a aplicação foi iniciada no servidor embutido Tomcat, na porta **8080**.

Os endpoints responderam corretamente:

- Acessando <http://localhost:8080/>, a aplicação retornou a mensagem:



- Acessando <http://localhost:8080/nome/Maryana e Mônica>, retornou:



Também foi gerado o arquivo .jar dentro da pasta target.

6. Conclusão

A prática permitiu compreender a estrutura de um projeto Spring Boot e o uso de Maven como ferramenta de build. Verificou-se que o uso de dependências automáticas e o servidor embutido simplifica o desenvolvimento.

Além disso, foi possível observar a importância da organização da estrutura de pastas e da correta configuração do pom.xml para garantir a compatibilidade com o Java instalado.