UNIVERSIDADE FEDERAL DO CEARÁ

11, Julho 2025

Sistemas Distribuídos - 01A - 2025.1

Lab. Spring Boot

Monalisa Silva Bezerra - 535614

Professor: Dr. Antonio Rafael Braga

1. Objetivo

Implementar um micro serviço básico em Spring Boot que responda a requisições HTTP em dois endpoints:

- *GET /* -> mensagem de boas-vindas.
- *GET /nome/{nome}* -> mensagem personalizada com o nome do usuário.

2. Etapas Realizadas

2.1 Instalação das ferramentas

- Java 21.0.7 LTS.
- Maven.

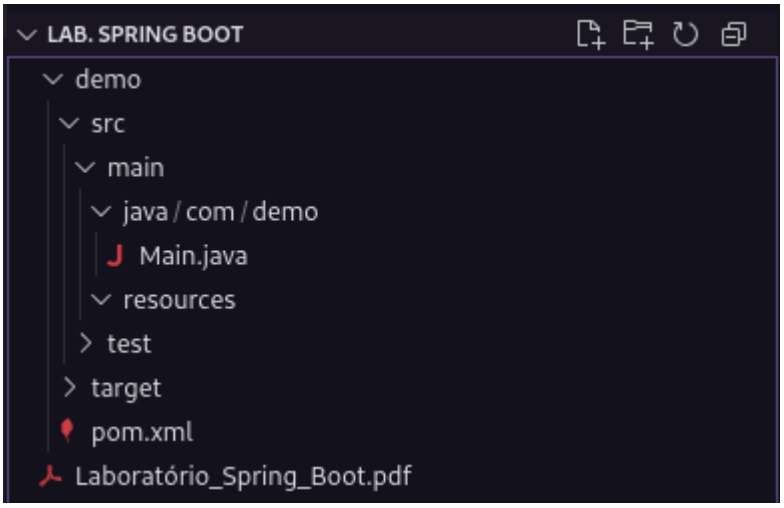
2.2 Criação do projeto

I. No VS Code, abriu-se a *Command Palette* (Ctrl + Shift + P) e executou-se “Maven: Generate from Archetype”, escolhendo o archetype padrão.

II. Foi definido:

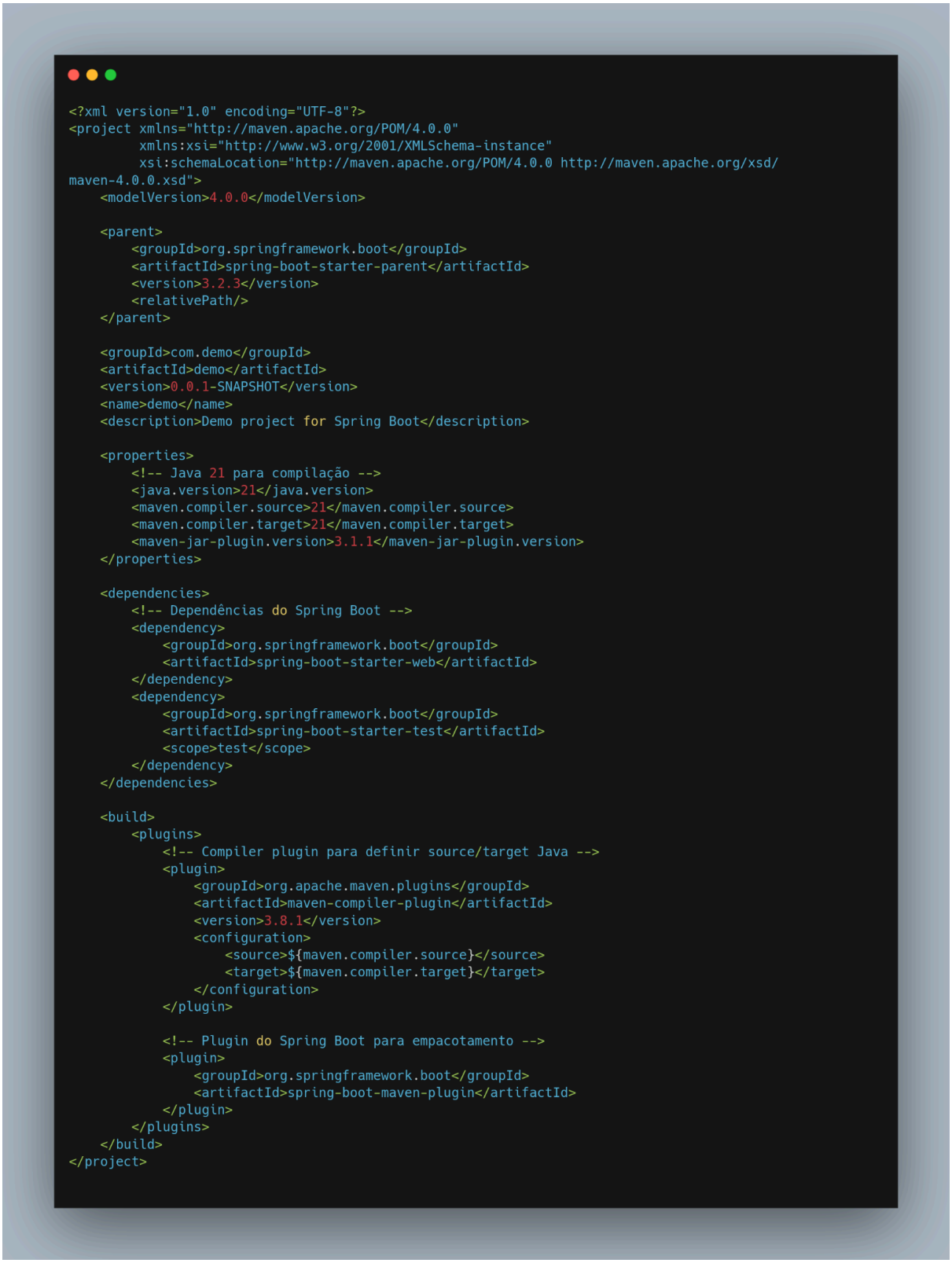
- *GroupId*: com.demo
- *ArtifactId*: demo

III. Foi confirmada a criação da estrutura de diretórios padrão:



2.3 Configuração do pom.xml

Abriu-se o pom.xml gerado e ajustou-se com as dependências necessárias de Web e Test. O conteúdo ficou conforme abaixo:



```xml
<?xml version="1.0" encoding="UTF-8"?>
<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>

    <parent>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-parent</artifactId>
        <version>3.2.3</version>
        <relativePath/>
    </parent>

    <groupId>com.demo</groupId>
    <artifactId>demo</artifactId>
    <version>0.0.1-SNAPSHOT</version>
    <name>demo</name>
    <description>Demo project for Spring Boot</description>

    <properties>
        <!-- Java 21 para compilação -->
        <java.version>21</java.version>
        <maven.compiler.source>21</maven.compiler.source>
        <maven.compiler.target>21</maven.compiler.target>
        <maven-jar-plugin.version>3.1.1</maven-jar-plugin.version>
    </properties>

    <dependencies>
        <!-- Dependências do Spring Boot -->
        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter-web</artifactId>
        </dependency>
        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter-test</artifactId>
            <scope>test</scope>
        </dependency>
    </dependencies>

    <build>
        <plugins>
            <!-- Compiler plugin para definir source/target Java -->
            <plugin>
                <groupId>org.apache.maven.plugins</groupId>
                <artifactId>maven-compiler-plugin</artifactId>
                <version>3.8.1</version>
                <configuration>
                    <source>${maven.compiler.source}</source>
                    <target>${maven.compiler.target}</target>
                </configuration>
            </plugin>

            <!-- Plugin do Spring Boot para empacotamento -->
            <plugin>
                <groupId>org.springframework.boot</groupId>
                <artifactId>spring-boot-maven-plugin</artifactId>
            </plugin>
        </plugins>
    </build>
</project>
```

2.4 Desenvolvimento do Código-Fonte

Na pasta com.demo, foi criado a classe Main.java, contendo:

- Anotação `@SpringBootApplication` para auto-configuração.

- `@RestController` com dois métodos mapeados:
- Método `main()` que invoca `SpringApplication.run(...)`.

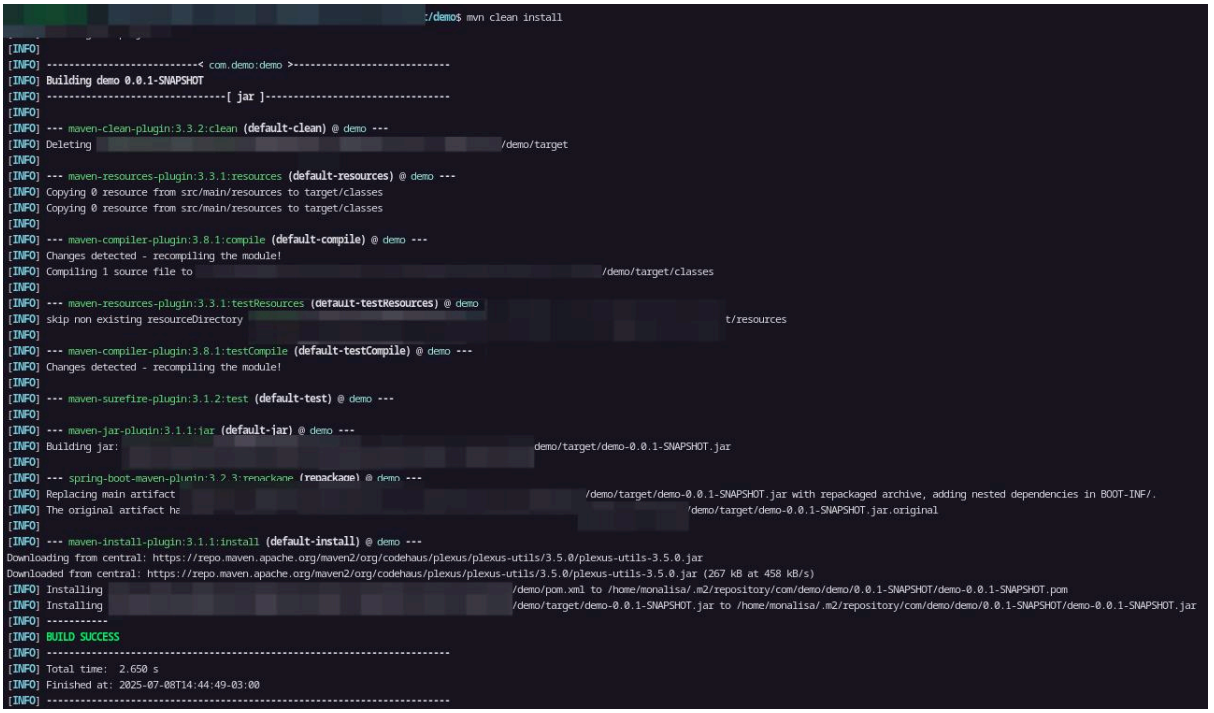
O código completo:

```java
package com.demo;

import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;
import org.springframework.web.bind.annotation.PathVariable;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RestController;

@RestController
@SpringBootApplication

public class Main {
    @RequestMapping("/")
    String home() {
        return "Olá, seja bem vindo ao teste de Spring Boot para disciplina de Sistemas Distribuídos";
    }

    @RequestMapping("/nome/{nome}")
    String piada(@PathVariable String nome) {
        return "Olá " + nome + ", você vai realmente conseguir passar nessa disciplina?";
    }
    public static void main(String[] args) {
        SpringApplication.run(Main.class, args);
    }
}
  return go(f, seed, [])
}
```

2.5 Build e Execução

No terminal, dentro da pasta raiz do projeto (`demo`), executou-se:

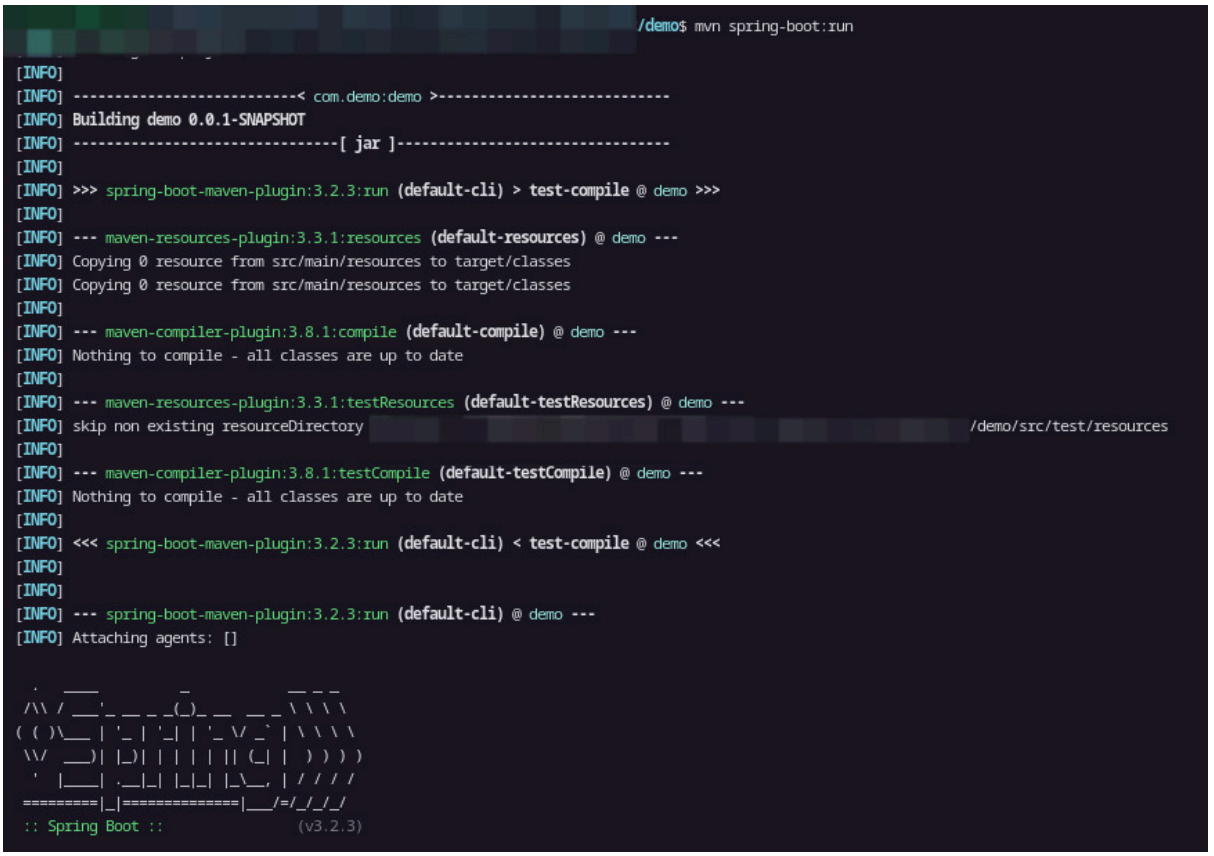
```
mvn clean install
```



A compilação ocorreu sem erros e os testes passaram corretamente.

Para iniciar a aplicação:

`mvn spring-boot:run`



Spring Boot iniciou o Tomcat embarcado na porta 8080 por padrão.

3. Resultados Obtidos

1. Compilação bem sucedida

- Módulo `demo` compilou sem falhas, gerando o JAR em `target/demo-0.0.1-SNAPSHOT.jar`.

2. Serviço HTTP ativo

- A aplicação está respondendo em `http://localhost:8080/` com a mensagem: “Olá, seja bem vindo ao teste de Spring Boot para disciplina de Sistemas Distribuídos”.
- Em `http://localhost:8080/nome/Monalisa`, retorna: “Olá Monalisa, você vai realmente conseguir passar nessa disciplina?”

4. Conclusão

O laboratório permitiu validar o fluxo de criação de um micro serviço Spring Boot, permitindo a conclusão da proposta de criar endpoints simples que servem como base para aplicações mais complexas em Sistemas Distribuídos.