Desenvolvimento de Serviços com Spring Boot

Etapa/Aula 02

Professor(a): Flávio Neves
E-mail: flavio.neves@prof.infnet.edu.br

Roteiro da Aula

- Apresentação do professor e alunos
- Apresentação do conteúdo da disciplina
- Roteiro da aula da semana
- Introdução ao Spring Boot

Apresentação do Professor

- NOME: Flávio da Silva Neves
- Graduação: Licenciatura em Computação - UPE
- Mestrado: Ciência da Computação – CIN - UFPE
- Doutorado: Ciência da Computação – CIN – UFPE
- Experiência no Profissional?
  - Ensino?
  - Mercado de computação?

Apresentação do Aluno

- Nome ?
- Estado / Município ?
- Trabalha ?
- Gosta de?
- Principais Habilidades com computação?
- Principais dificuldades com computação?
- Conhecimentos em Spring Boot??

Competências Abordadas na Disciplina

- Criar aplicações web a partir do Spring Boot Initializer
- Desenvolver RESTful APIs com Spring Boot
- Implementar persistência de dados com JPA, Redis e MongoDB
- Implementar testes em aplicações SpringBoot
- Implementar medidas de segurança em aplicações Spring Boot
- Realizar o deploy em aplicações SpringBoot

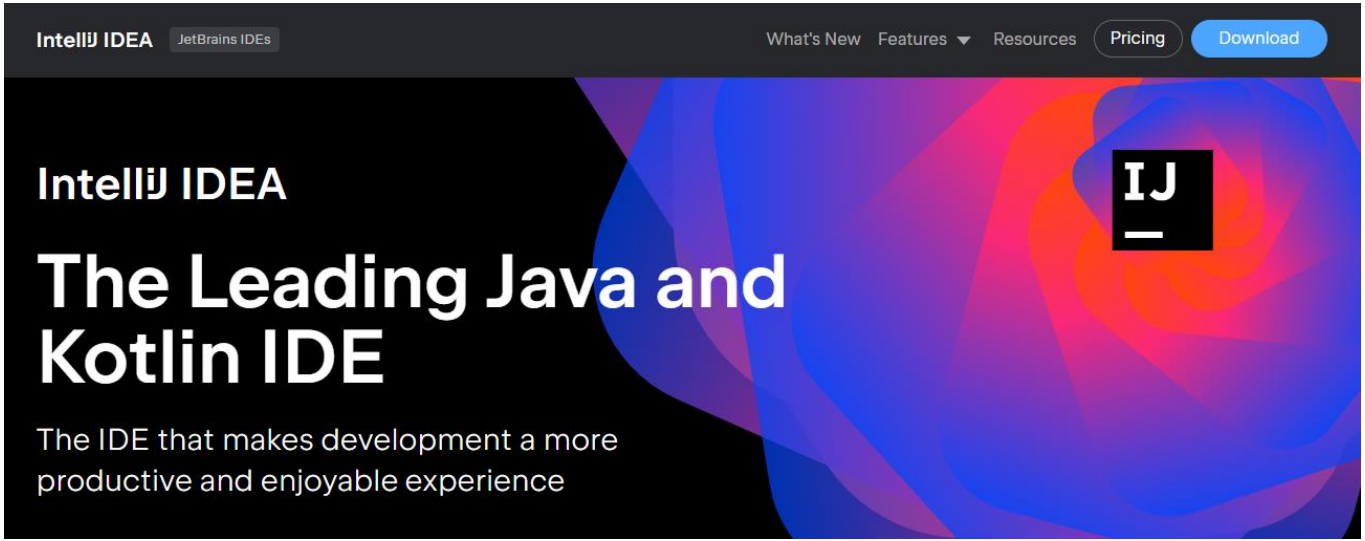
Semana 01

Roteiro da Aula

- IDE Recomendada
- Escolher entre Maven e Gradle como ferramenta de construção do projeto.
- Iniciar projetos através da interface web do Spring Initializr
- Iniciar projetos por meio da linha de comando com o Spring Boot CLI
- Aplicar técnicas de gerenciamento de dependências desde a criação do projeto até o desenvolvimento e a manutenção
- Utilizar o recurso de autoconfiguração para reduzir ou eliminar o código repetitivo
- Configurar a IDE para trabalhar em projetos com Spring Boot

IDE Recomendada

- IntelliJ IDEA



https://www.jetbrains.com/idea/?var=1

Maven X Gradle

- Maven e Gradle são ferramentas de compilação usadas em Java e outros projetos baseados em JVM para:
- Gerenciar dependências,
- Criar e empacotar aplicativos; e
- Executar outras tarefas relacionadas.

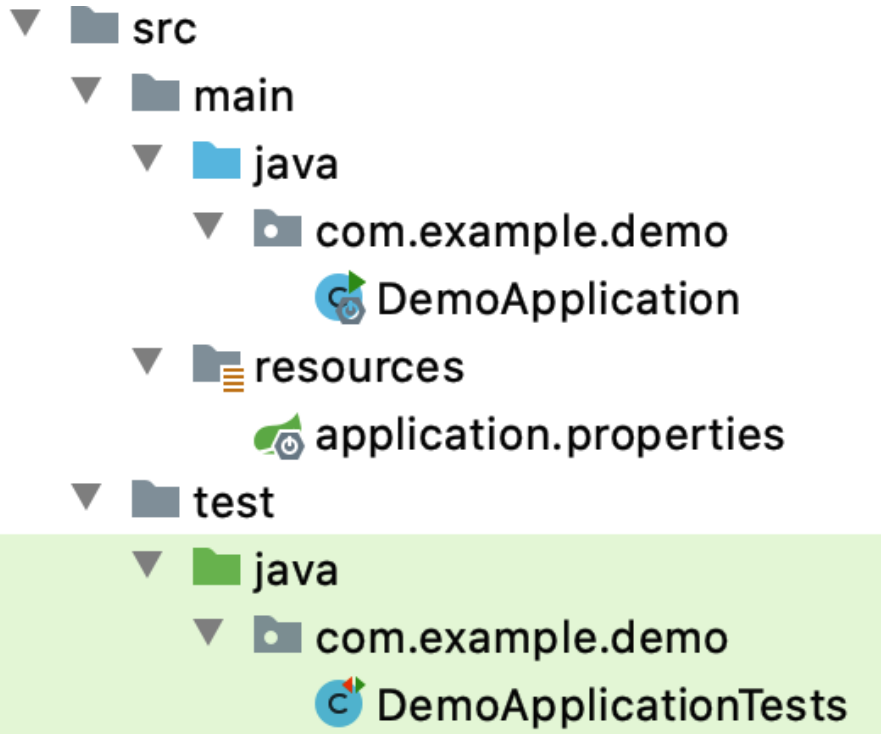
Maven

- Maven é uma escolha popular e sólida para um sistema de automação de construção.
- Já existe há algum tempo, tendo tido o seu início em 2002 e tornando-se um projeto de nível superior na Apache Software Foundation em 2003.
- A sua abordagem declarativa conceitualmente mais simples do que as alternativas
- Basta criar um ficheiro formatado em XML chamado pom.xml com as dependências e plug-ins desejados.

Maven

- O Maven cria e espera uma estrutura de projeto específica por convenção.
- Não é recomendado alterar a estrutura, a menos que esteja preparado para lutar contra a sua ferramenta de construção, uma missão contraproducente se alguma vez existiu.
- Para a grande maioria dos projetos, a estrutura convencional do Maven funciona perfeitamente, portanto, não é algo que você provavelmente precisará alterar.

Maven


src
main
java
com.example.demo
DemoApplication
resources
application.properties
test
java
com.example.demo
DemoApplicationTests

Gradle

- O Gradle é outra opção popular para a construção de projetos Java Virtual Machine (JVM).
- Lançado pela primeira vez em 2008, o Gradle utiliza uma linguagem específica de domínio (DSL) para produzir um arquivo de compilação build.gradle que é mínimo e flexível.
- Segue um exemplo de um arquivo de compilação do Gradle para um aplicativo Spring Boot.

Gradle

```
plugins {
        id 'org.springframework.boot' version '2.4.0'
        id 'io.spring.dependency-management' version '1.0.10.RELEASE
        id 'java'
}

group = 'com.example'
version = '0.0.1-SNAPSHOT'
sourceCompatibility = '11'

repositories {
        mavenCentral()
}

dependencies {
        implementation 'org.springframework.boot:spring-boot-starter
        testImplementation 'org.springframework.boot:spring-boot-star
}
```

Gradle

- O Gradle permite que o programador opte por utilizar as linguagens de programação Groovy ou Kotlin para uma DSL.
- Também oferece várias funcionalidades destinadas a reduzir o tempo de espera pela construção de um projeto.

Mave ou Gradle?

- Em geral, ambas as abordagens são capazes e amplamente utilizadas em um projeto, e a escolha entre as duas depende das necessidades e preferências específicas da equipe do projeto.
- O Maven pode ser uma escolha melhor para projetos que exigem uma configuração de compilação mais padronizada e uma abordagem mais simples.
- O Gradle pode ser preferido para compilações mais complexas e personalizáveis.

Visão Geral do Spring Boot

- Spring Boot é um framework Java de código aberto amplamente utilizado para desenvolvimento de aplicativos empresariais.
- Ele fornece um ambiente de tempo de execução para aplicativos Java que permite que eles sejam executados de forma independente sem precisar de um servidor de aplicativos externo.
- O Spring Boot oferece muitos recursos e benefícios para o desenvolvimento de aplicativos, incluindo um conjunto abrangente de bibliotecas, facilidade de uso e configuração automática.

Por que usar Spring Boot?

- Simplificar o desenvolvimento de aplicativos Java;
- Elimina a necessidade de configurar manualmente bibliotecas e estruturas;
- Facilita a integração de várias bibliotecas e estruturas;
- Simplifica a criação de testes automatizados.

Principais Recursos do Spring Boot

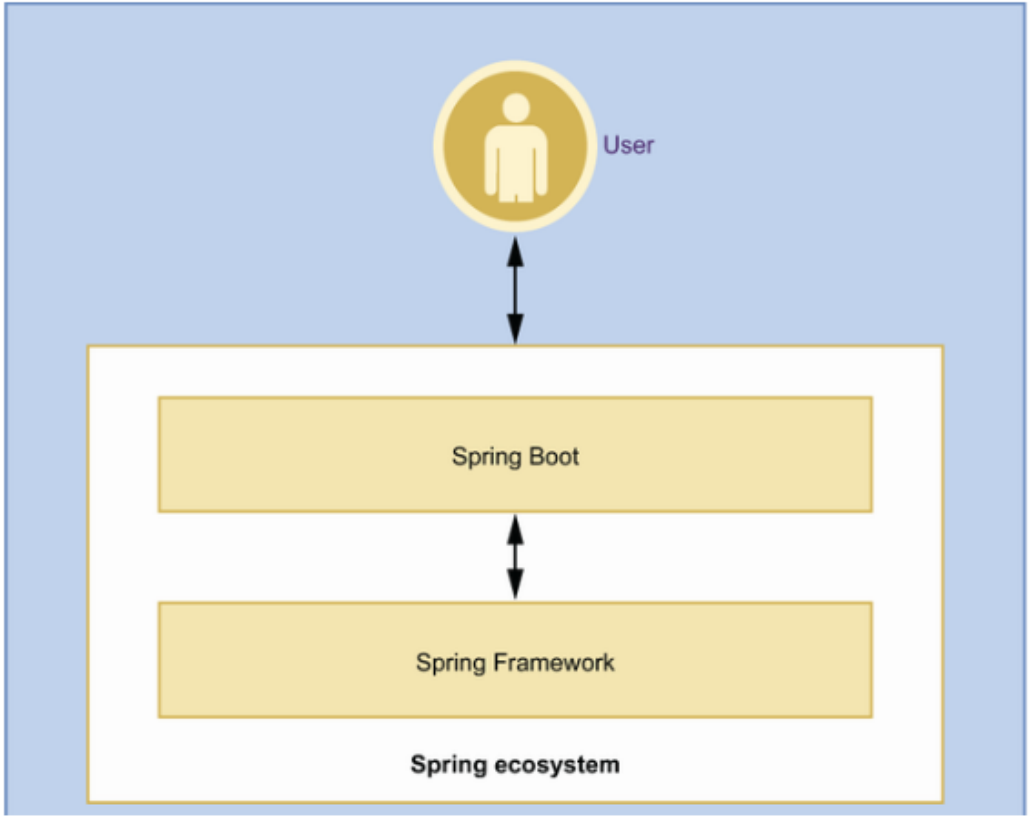
- Configuração automática:
  - O Spring Boot fornece configuração automática para várias bibliotecas e estruturas, o que significa que ele pode detectar automaticamente as dependências do aplicativo e configurá-las sem que o desenvolvedor precise fazer isso manualmente.
- Embutido Tomcat:
  - O Spring Boot incorpora o Tomcat em seu ambiente de tempo de execução, o que significa que os desenvolvedores não precisam se preocupar com a configuração e gerenciamento de um servidor de aplicativos externo.

Principais Recursos do Spring Boot

- Suporte ao desenvolvimento de API RESTful:
  - O Spring Boot oferece suporte ao desenvolvimento de API RESTful, permitindo que os desenvolvedores criem rapidamente APIs RESTful e integrem várias bibliotecas, como Spring Data e Spring Security.
- Bibliotecas de terceiros:
  - O Spring Boot permite que os desenvolvedores usem bibliotecas de terceiros facilmente, fornecendo um mecanismo de gerenciamento de dependências que gerencia automaticamente as versões das bibliotecas e garante que elas sejam compatíveis.

Componentes Principais do Spring Boot

- Spring Boot Starter;
  - **Spring Core;**
  - **Spring Web;**
  - **Spring Web MVC;**
  - **Lombok;**
  - **Servlet API.**
- Spring Boot AutoConfigurator
- Spring Boot Actuator



1 Inicialização da bota Spring

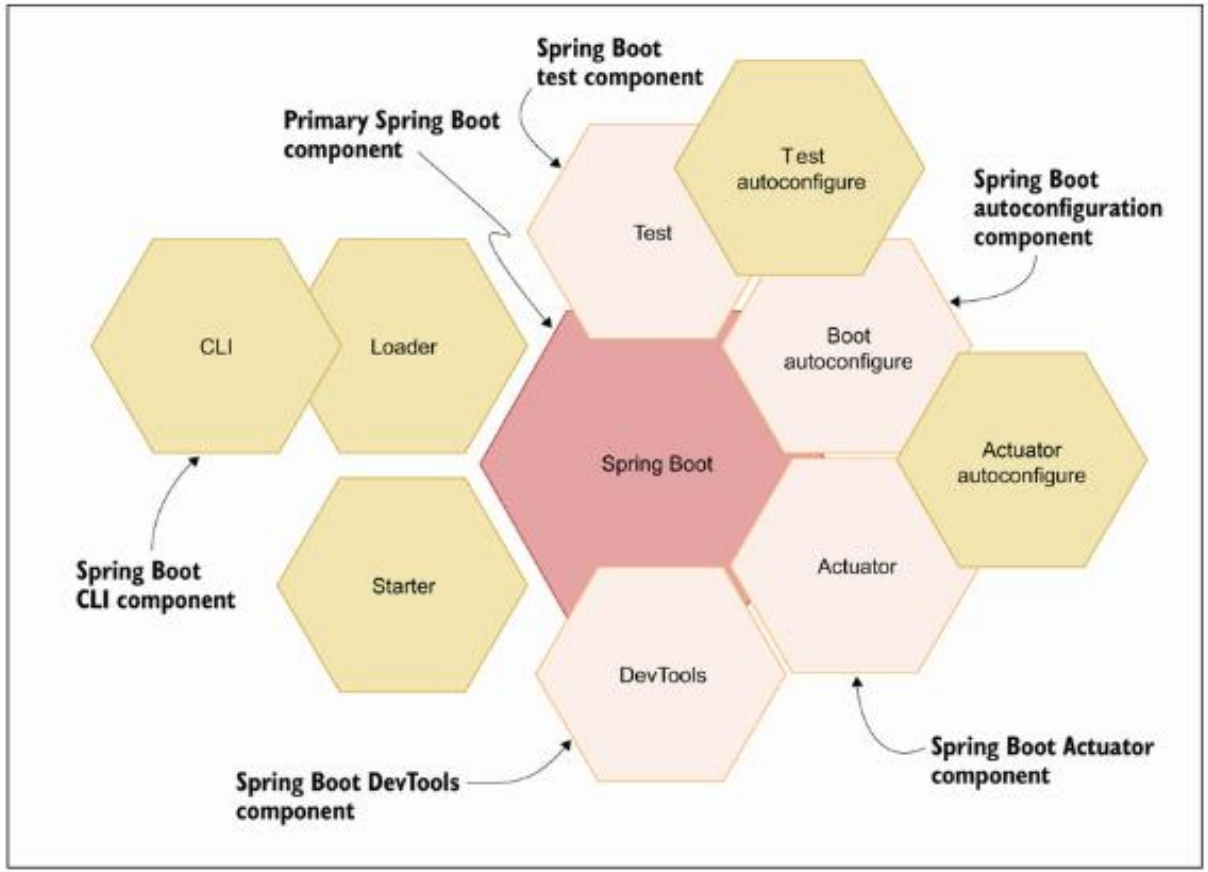
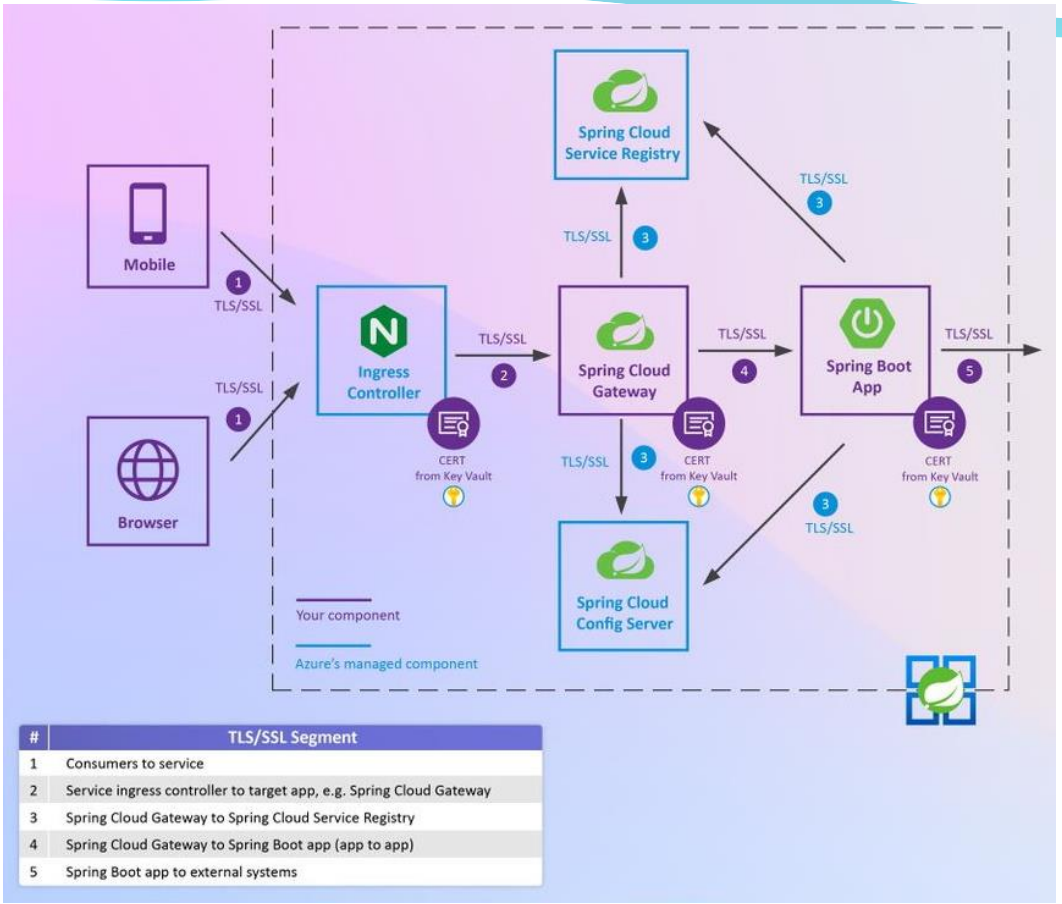


Figura 1.2 Componentes do Spring Boot



#	TLS/SSL Segment
1	Consumers to service
2	Service ingress controller to target app, e.g. Spring Cloud Gateway
3	Spring Cloud Gateway to Spring Cloud Service Registry
4	Spring Cloud Gateway to Spring Boot app (app to app)
5	Spring Boot app to external systems

Lombok

- Lombok é uma biblioteca Java que gera automaticamente os construtores, getter, setter, toString e outros com base na presença de algumas anotações na classe Plain Old Java Object (POJO).
- Tudo que você precisa fazer é usar a anotação apropriada na classe POJO.
- Por exemplo, para gerar um método getter para todas as variáveis de membro na classe POJO, você pode especificar `@Getter` uma anotação na classe.

Spring Initializr

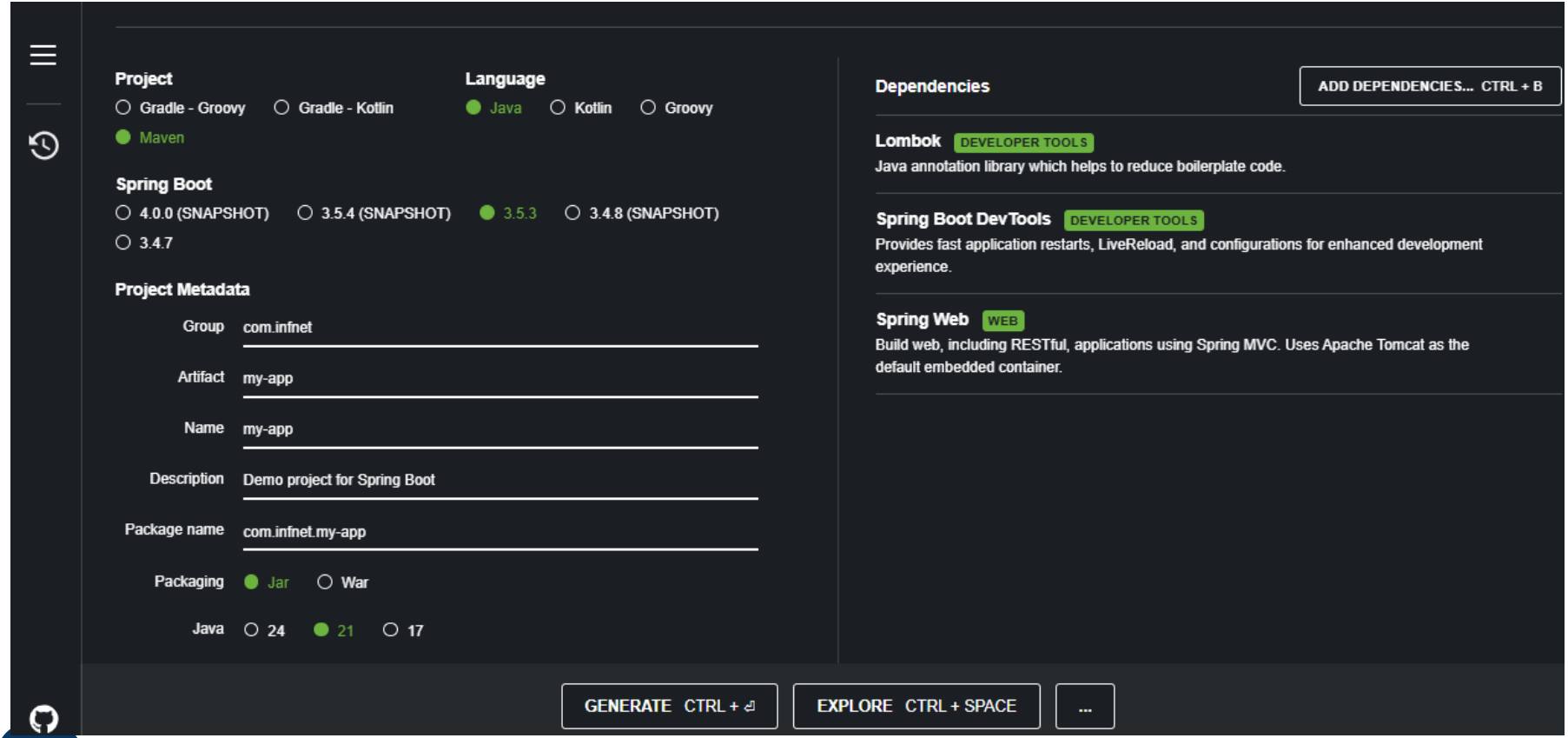
- O Spring Initializr fornece uma API extensível para gerar projetos baseados na JVM com implementações para vários conceitos comuns.
- Também oferece várias funcionalidades destinadas a reduzir o tempo de espera pela construção de um projeto.

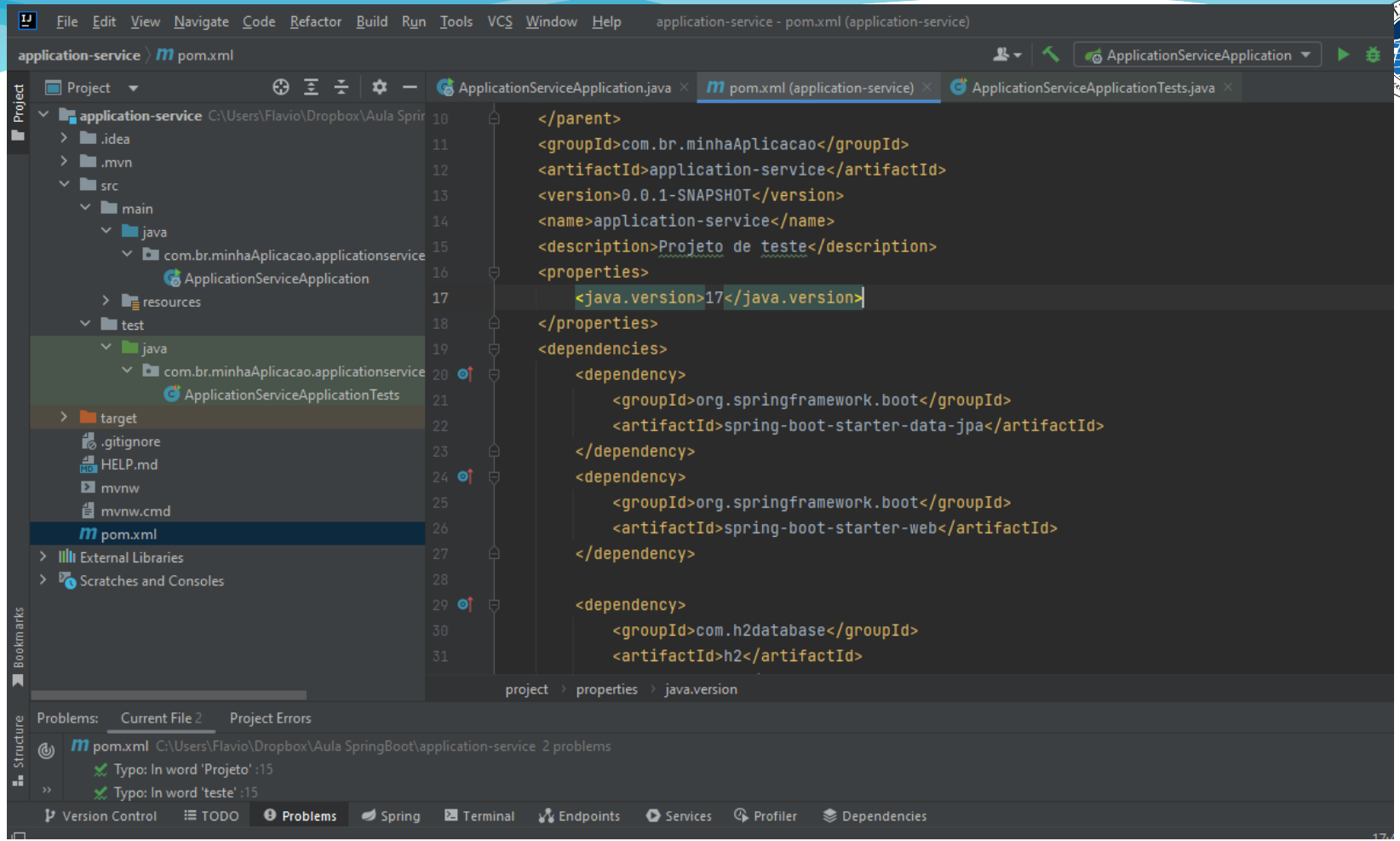
https://start.spring.io/

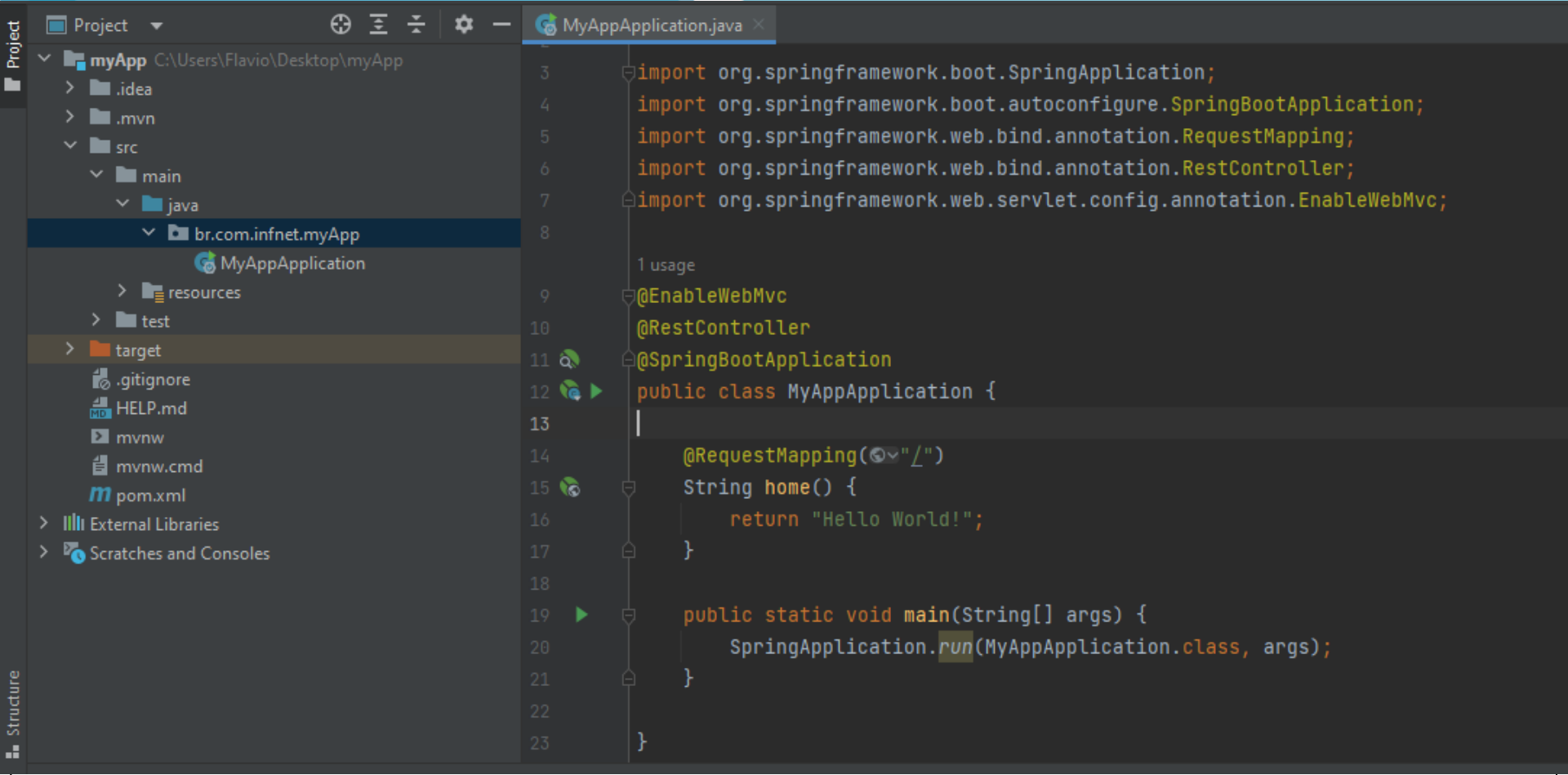
O que preciso instalar na máquina?

- Java 21
- IntelliJ IDEA

Como Começar com o Spring Boot




File Edit View Navigate Code Refactor Build Run Tools VCS Window Help
application-service - pom.xml (application-service)
application-service
pom.xml
ApplicationServiceApplication
Project
application-service C:\Users\Flavio\Dropbox\Aula Sprir
.idea
.mvn
src
main
java
com.br.minhaAplicacao.applicationservice
ApplicationServiceApplication
resources
test
java
com.br.minhaAplicacao.applicationservice
ApplicationServiceApplicationTests
target
.gitignore
HELP.md
mvnw
mvnw.cmd
pom.xml
External Libraries
Scratches and Consoles
ApplicationServiceApplication.java
pom.xml (application-service)
ApplicationServiceApplicationTests.java
</parent>
<groupId>com.br.minhaAplicacao</groupId>
<artifactId>application-service</artifactId>
<version>0.0.1-SNAPSHOT</version>
<name>application-service</name>
<description>Projeto de teste</description>
<properties>
<java.version>17</java.version>
</properties>
<dependencies>
<dependency>
<groupId>org.springframework.boot</groupId>
<artifactId>spring-boot-starter-data-jpa</artifactId>
</dependency>
<dependency>
<groupId>org.springframework.boot</groupId>
<artifactId>spring-boot-starter-web</artifactId>
</dependency>
<dependency>
<groupId>com.h2database</groupId>
<artifactId>h2</artifactId>
project
properties
java.version
Problems:
Current File 2
Project Errors
pom.xml C:\Users\Flavio\Dropbox\Aula SpringBoot\application-service 2 problems
Typo: In word 'Projeto' :15
Typo: In word 'teste' :15
Version Control
TODO
Problems
Spring
Terminal
Endpoints
Services
Profiler
Dependencies


Project
myApp C:\Users\Flavio\Desktop\myApp
.idea
.mvn
src
main
java
br.com.infnet.myApp
MyAppApplication
resources
test
target
.gitignore
HELP.md
mvnw
mvnw.cmd
pom.xml
External Libraries
Scratches and Consoles
MyAppApplication.java
import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RestController;
import org.springframework.web.servlet.config.annotation.EnableWebMvc;
1 usage
@EnableWebMvc
@RestController
@SpringBootApplication
public class MyAppApplication {
@RequestMapping("/")
String home() {
return "Hello World!";
}
public static void main(String[] args) {
SpringApplication.run(MyAppApplication.class, args);
}
}

Algumas anotações

- Spring Boot Annotations é uma forma de metadados que fornece dados sobre um programa.
- Por outras palavras, as anotações são utilizadas para fornecer informações suplementares sobre um programa. Não é uma parte da aplicação que desenvolvemos.
- Não tem um efeito direto no funcionamento do código que anotam.

Algumas anotações

- **@Configuration:**
  - Essa é uma anotação a nível de Classe que diz para o Container de Inversão de Controle do Spring que essa classe é a fonte de beans (dependencias requeridas por outras classes no projeto).

Algumas anotações

- **@Bean:**
  - Essa anotação é no nível do Método/Função e indica que esse método/função cria e retorna um “bean” que pode ser usado como dependência em outras classes do projeto.
  - E vale lembrar que temos dois tipos de injeção de dependência no Spring: por tipo (Default) ou por nome.

Algumas anotações

- **@RestController/@Controller:**
  - Anotação de nível de classe que indica que a classe que carrega essa anotação precisará ser exposta como um bean de controlador e injetada nos consumidores que indicaram que precisam dele.
  - Os controladores que carregam essa anotação fornecem suporte para especificar coisas específicas de Controller, como o caminho, o verbo HTTP, o corpo da solicitação, o parâmetro de consulta, o parâmetro do caminho, os tipos de conteúdo aceitos, etc.

Algumas anotações

- **@SpringBootApplication:**
  - A anotação @SpringBootApplication do Spring Boot é utilizada para marcar uma classe de configuração que declara um ou mais métodos @Bean e também aciona a autoconfiguração e a verificação de componentes.
  - É o mesmo que declarar uma classe com as anotações @Configuration, @EnableAutoConfiguration e @ComponentScan.

Algumas anotações

- **@EnableWebMvc:**
  - Diz ao Spring para importar configurações MVC. Com a configuração automática do Spring Boot, essa anotação pode ser omitida.

Algumas anotações

- **@RequestMapping:**
  - Serve como o coração do mapeamento de solicitações HTTP em aplicações Spring.
  - Ela é a ponte que liga as requisições dos usuários aos métodos dos seus controladores, encarregados de processá-las.

Spring Boot CLI

- Spring Boot CLI (Command Line Interface) é um software Spring Boot para executar e testar aplicativos Spring Boot a partir do prompt de comando.
- O Spring Boot usando CLI, usa internamente os componentes Spring Boot Starter e Spring Boot AutoConfigurate para resolver todas as dependências e executar o aplicativo.
- Ele contém internamente Groovy e Grape (JAR Dependency Manager) para adicionar Spring Boot Defaults e resolver todas as dependências automaticamente.

Desenvolvimento de Serviços com Spring Boot

Etapa/Aula 02

Professor(a): Flávio Neves
E-mail: flavio.neves@prof.infnet.edu.br