

Spring Boot

Edilson Silva



Spring Boot



Spring Boot é um módulo Spring que fornece o recurso RAD (Rapid Application Development) para o framework Spring.

O que é Spring Boot



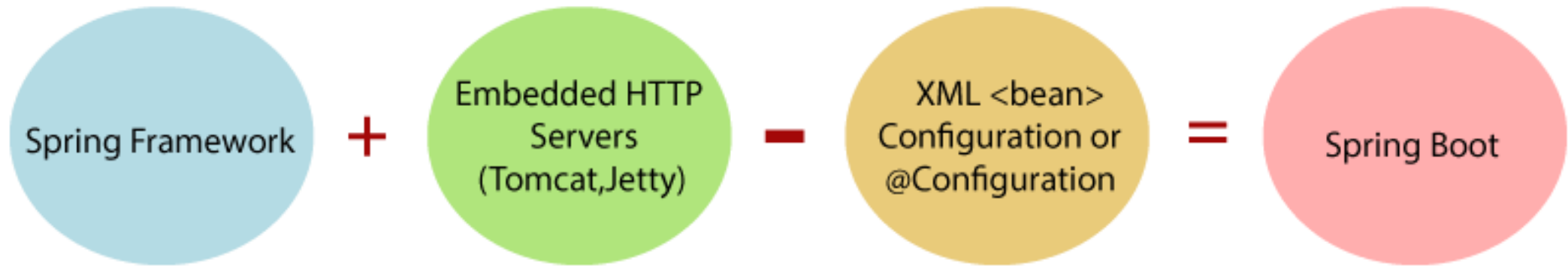
O Spring Boot é um projeto construído sobre o Spring Framework. Ele fornece uma maneira mais fácil e rápida de instalar, configurar e executar aplicativos simples e baseados na web. É um módulo Spring que fornece o recurso RAD (Rapid Application Development) para o Spring Framework.

O que é Spring Boot



Ele é usado para criar um aplicativo autônomo baseado em Spring que você pode simplesmente executar porque precisa de uma configuração Spring mínima.

Estrutura



Resumindo, Spring Boot é a combinação de Spring Framework e Embedded Servers.

No Spring Boot, não há exigência de configuração XML (descritor de implantação). Ele usa a convenção sobre o paradigma de design de software de configuração, o que significa que diminui o esforço do desenvolvedor. Podemos usar Spring STS IDE ou Spring Initializer para desenvolver aplicativos Spring Boot Java.

Por que devemos usar o Spring Boot Framework?

Devemos usar o Spring Boot Framework porquê:

- A abordagem de injeção de dependência é usada no Spring Boot.
- Ele contém recursos poderosos de gerenciamento de transações de banco de dados.
- Simplifica a integração com outros frameworks Java como JPA/Hibernate ORM, Struts, etc.
- Reduz o custo e o tempo de desenvolvimento do aplicativo.

Por que devemos usar o Spring Boot Framework?

Juntamente com o Spring Boot Framework, muitos outros projetos irmãos do Spring ajudam a criar aplicativos que atendem às necessidades de negócios modernos. Existem os seguintes projetos irmãos do Spring

Projetos Spring

- Spring Data:** Simplifica o acesso aos dados dos bancos de dados relacionais e NoSQL.
- Spring Batch:** Fornece processamento em lote poderoso.
- Spring Security:** É uma estrutura de segurança que fornece segurança robusta aos aplicativos.
- Spring Social:** Suporta integração com redes sociais como o LinkedIn.
- Spring Integration:** É uma implementação dos Padrões de Integração Corporativa. Ele facilita a integração com outros aplicativos corporativos usando mensagens leves e adaptadores declarativos.

Vantagens Spring

- Ele cria aplicativos Spring independentes que podem ser iniciados usando Java -jar.
- Ele testa aplicativos da web facilmente com a ajuda de diferentes servidores HTTP incorporados, como Tomcat, Jetty, etc.
- Não precisamos implantar arquivos WAR.Ele fornece POMs 'iniciantes' opinativos para simplificar nossa configuração Maven.
- Ele fornece recursos prontos para produção, como métricas, verificações de integridade e configuração externa.
- Não há nenhum requisito para configuração de XML.
- Ele oferece uma ferramenta CLI para desenvolver e testar o aplicativo Spring Boot.Oferece o número de plug-ins.

Vantagens Spring

•**Spring:** Spring Framework é o framework de desenvolvimento de aplicativos Java mais popular. A principal característica do Spring Framework é a **Injeção de dependência** ou **Inversão de Controle** (IoC). Com a ajuda do Spring Framework, podemos desenvolver um aplicativo **fracamente** acoplado. É melhor usar se o tipo de aplicativo ou as características forem puramente definidas.

Vantagens Spring

- Spring Boot:** Spring Boot é um módulo do Spring Framework. Ele nos permite construir um aplicativo autônomo com configurações mínimas ou zero. É melhor usar se quisermos desenvolver um aplicativo simples baseado em Spring ou serviços RESTful.

Arquitetura do Spring Boot

Spring Boot é um módulo do Spring Framework. Ele é usado para criar aplicativos baseados em Spring autônomos e de nível de produção com esforços mínimos. Ele é desenvolvido sobre o núcleo do Spring Framework.

O Spring Boot segue uma arquitetura em camadas na qual cada camada se comunica com a camada diretamente abaixo ou acima (estrutura hierárquica) dela.

Arquitetura do Spring Boot

Antes de entender a **Arquitetura Spring Boot** , devemos conhecer as diferentes camadas e classes presentes nela. Existem **quatro** camadas no Spring Boot são as seguintes:

- **Camada de apresentação**
- **Camada de Negócios**
- **Camada de Persistência**
- **Camada de banco de dados**

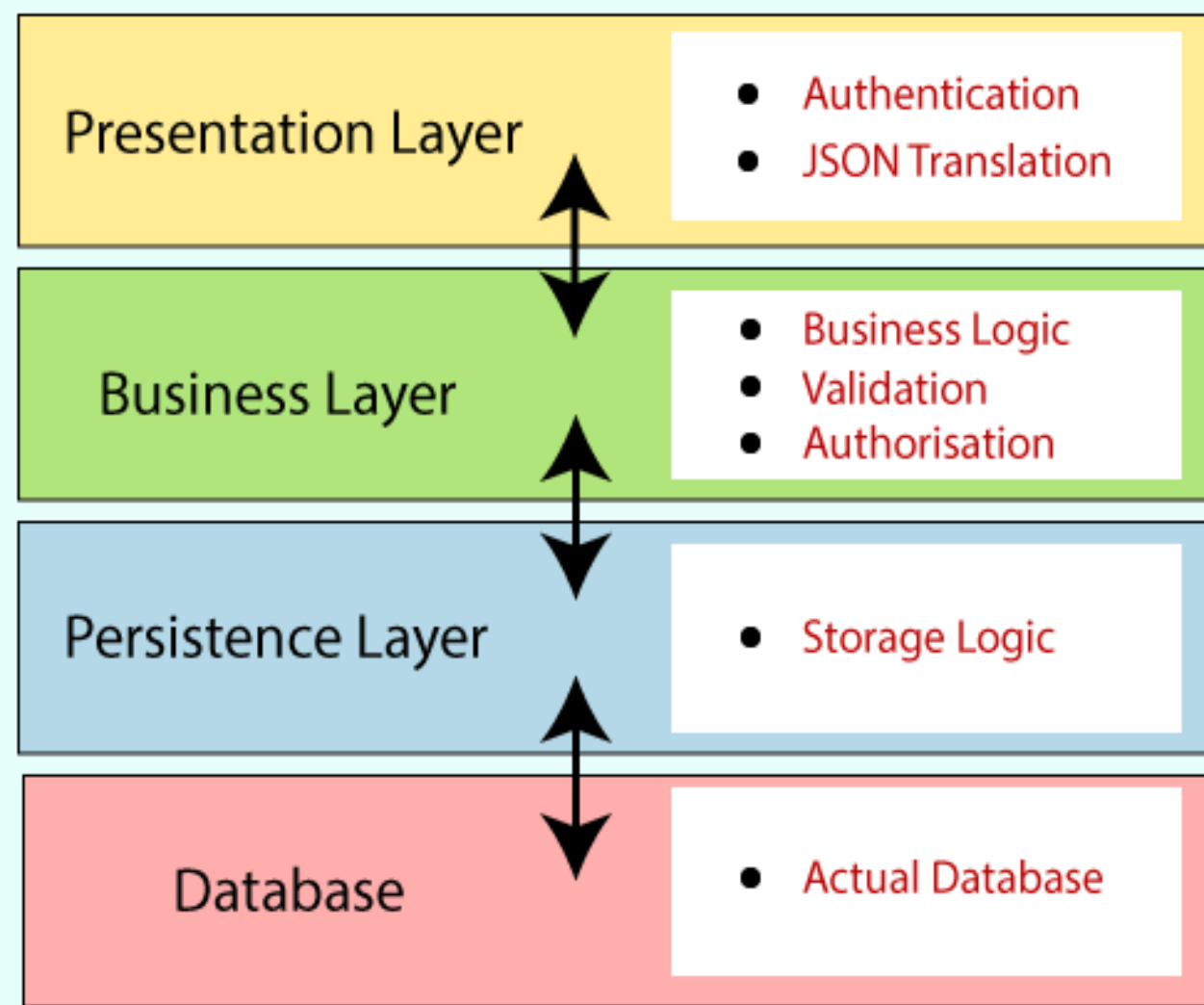
Arquitetura do Spring Boot

Camada de Apresentação: A camada de apresentação lida com as solicitações HTTP, traduz o parâmetro JSON em objeto, autentica a solicitação e a transfere para a camada de negócios. Resumindo, consiste em **visualizações**, ou seja, parte do frontend.

Camada de negócios: A camada de negócios lida com toda a **lógica de negócios**. Consiste em classes de serviço e usa serviços fornecidos por camadas de acesso a dados. Ele também executa **autorização** e **validação**.

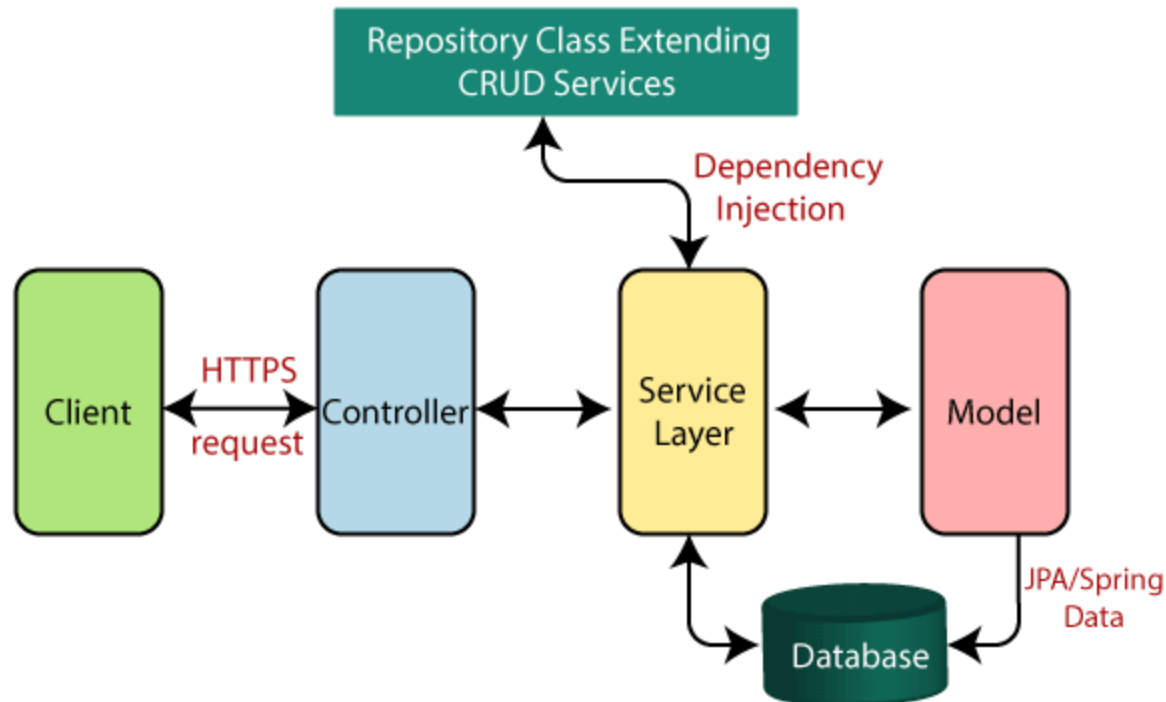
Camada de persistência: A camada de persistência contém toda a **lógica de armazenamento** e converte objetos de negócios de e para linhas do banco de dados.

Camada de Banco de Dados: Na camada de banco de dados, as operações **CRUD** (criar, recuperar, atualizar, excluir) são executadas.



Arquitetura do Spring Boot

Spring Boot flow architecture



- Agora temos classes validadoras, classes de visualização e classes utilitárias.
- O Spring Boot usa todos os módulos do Spring MVC, Spring Data, etc. A arquitetura do Spring Boot é a mesma do Spring MVC, exceto por uma coisa: não há necessidade de classes DAO e DAOImpl **no** Spring **boot** .
- Cria uma camada de acesso a dados e executa a operação CRUD.
- O cliente faz as solicitações HTTP (PUT ou GET).
- A solicitação vai para o controlador e o controlador mapeia essa solicitação e a manipula. Depois disso, ele chama a lógica de serviço, se necessário. m erro ocorrer

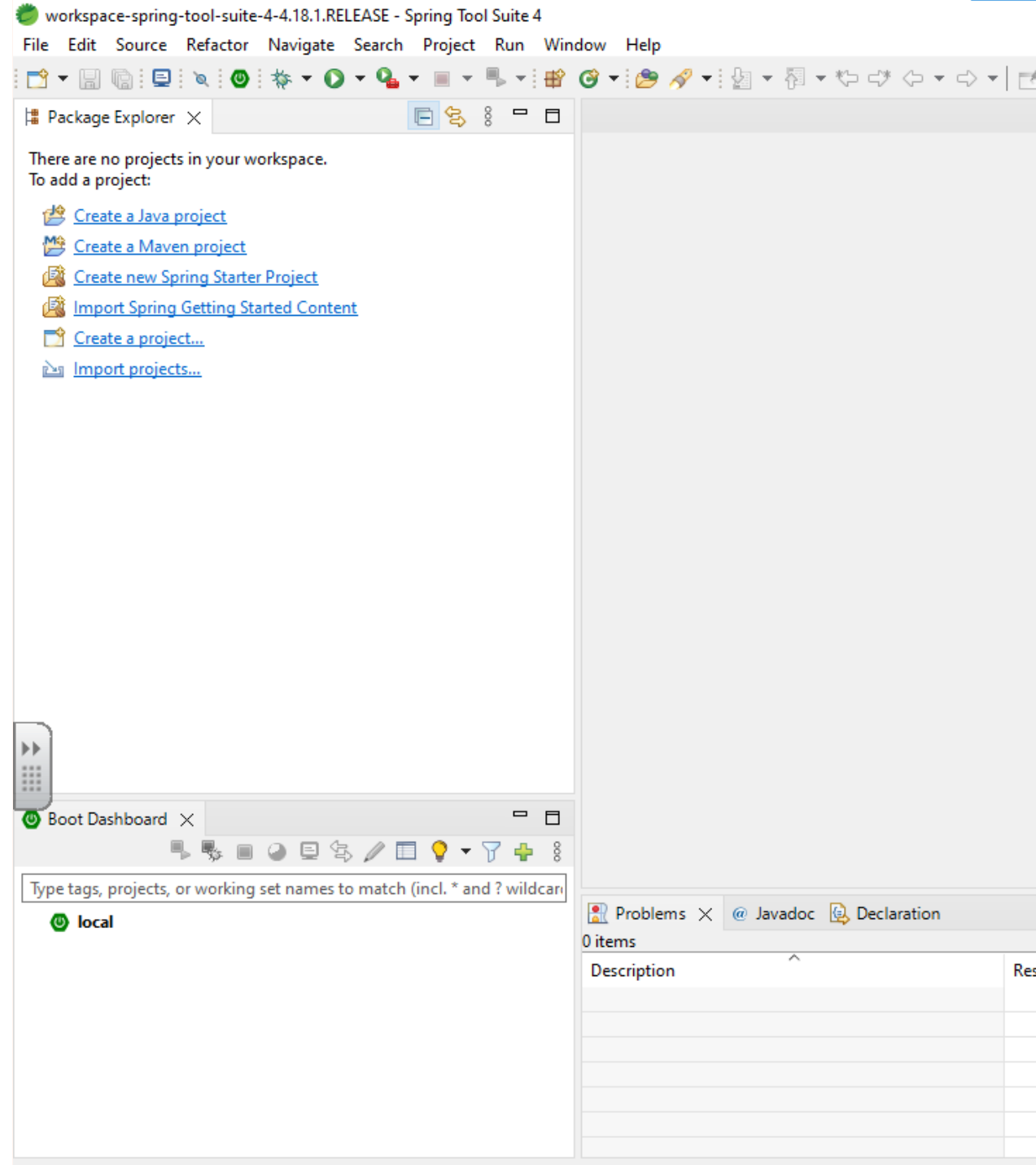
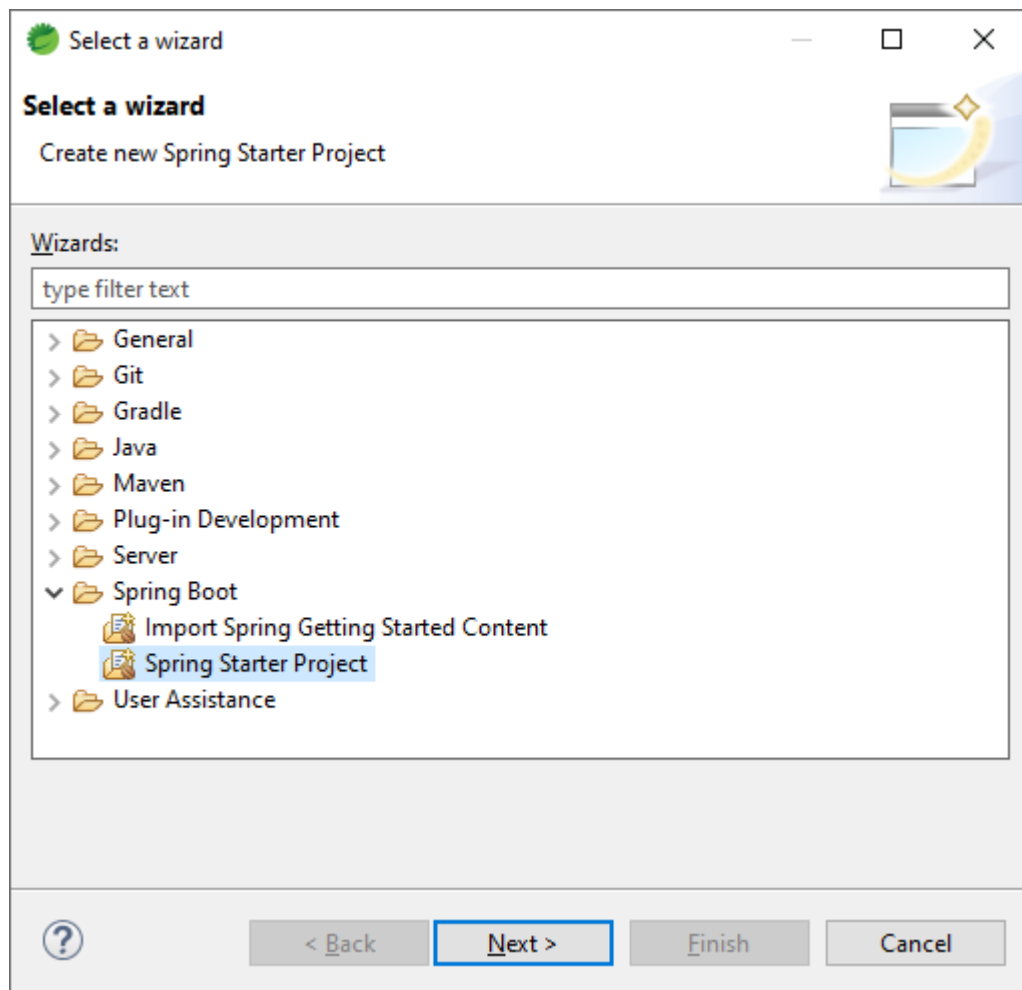
Desenvolvendo um projeto

Edilson Silva



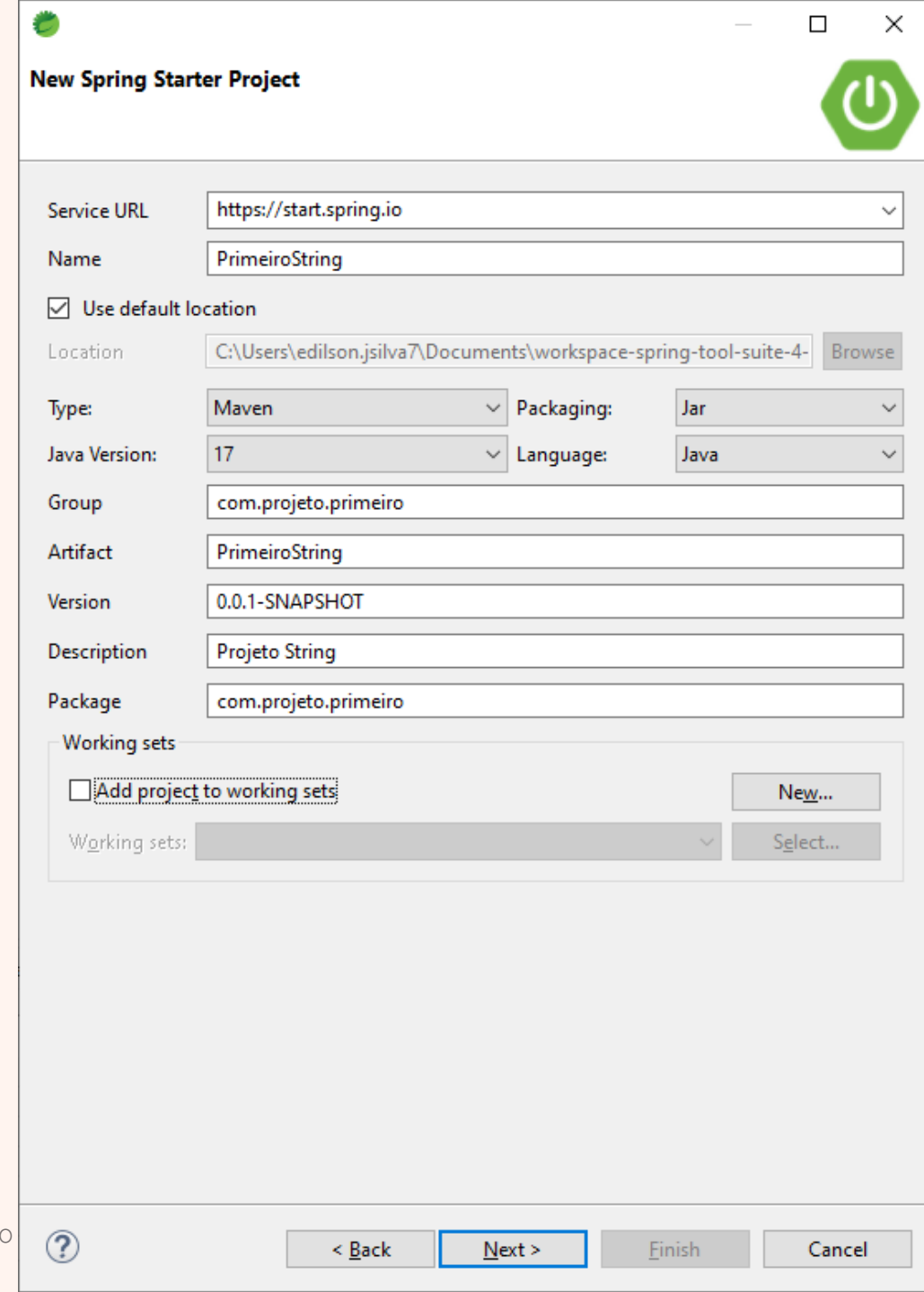
Criar um novo projeto Spring

Menu File -> New -> Other



Criar um novo projeto Spring

Faça as configurações exatamente como está sendo mostrada a seguir



New Spring Starter Project

Service URL:

Name:

☒ Use default location

Location:

Type: Packaging:

Java Version: Language:

Group:

Artifact:

Version:

Description:

Package:

Working sets

☐ Add project to working sets

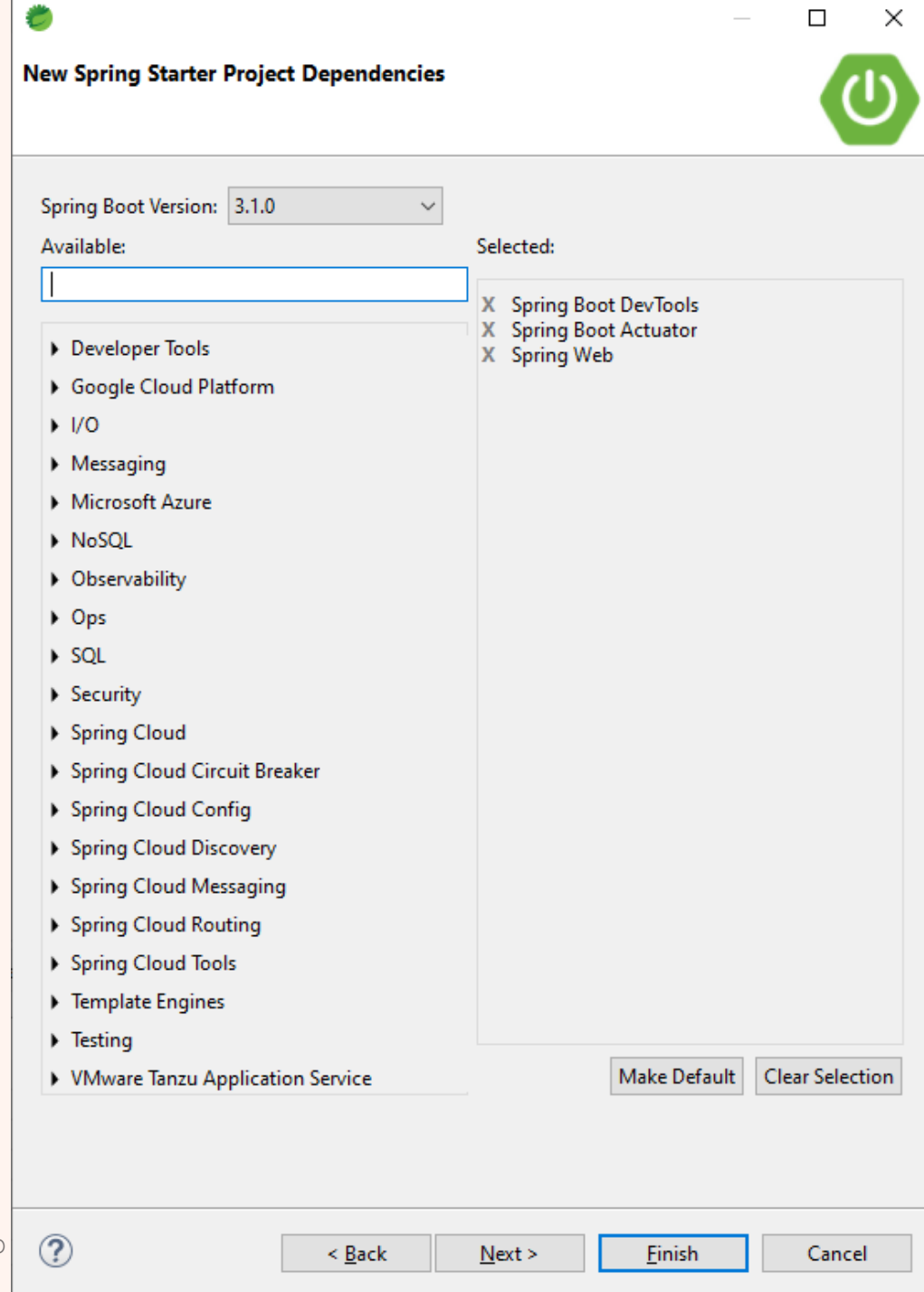
Working sets:

Criar um novo projeto Spring

Digite as dependências que serão utilizadas.

As dependências estão ao lado direito da tela.

Clique em Next e a tela posterior Finish



New Spring Starter Project Dependencies

Spring Boot Version: 3.1.0

Available:

Selected:

- X Spring Boot DevTools
- X Spring Boot Actuator
- X Spring Web

Developer Tools

Google Cloud Platform

I/O

Messaging

Microsoft Azure

NoSQL

Observability

Ops

SQL

Security

Spring Cloud

Spring Cloud Circuit Breaker

Spring Cloud Config

Spring Cloud Discovery

Spring Cloud Messaging

Spring Cloud Routing

Spring Cloud Tools

Template Engines

Testing

VMware Tanzu Application Service

Make Default

Clear Selection

< Back

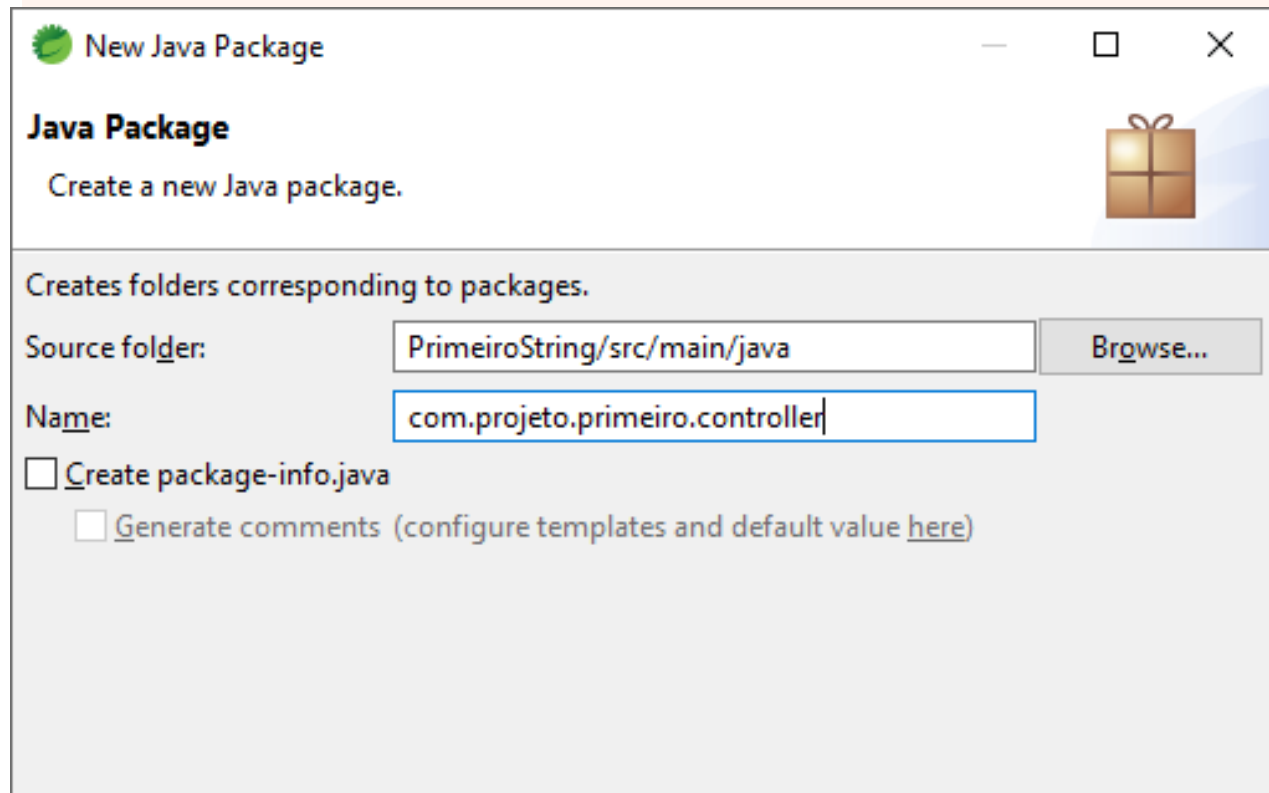
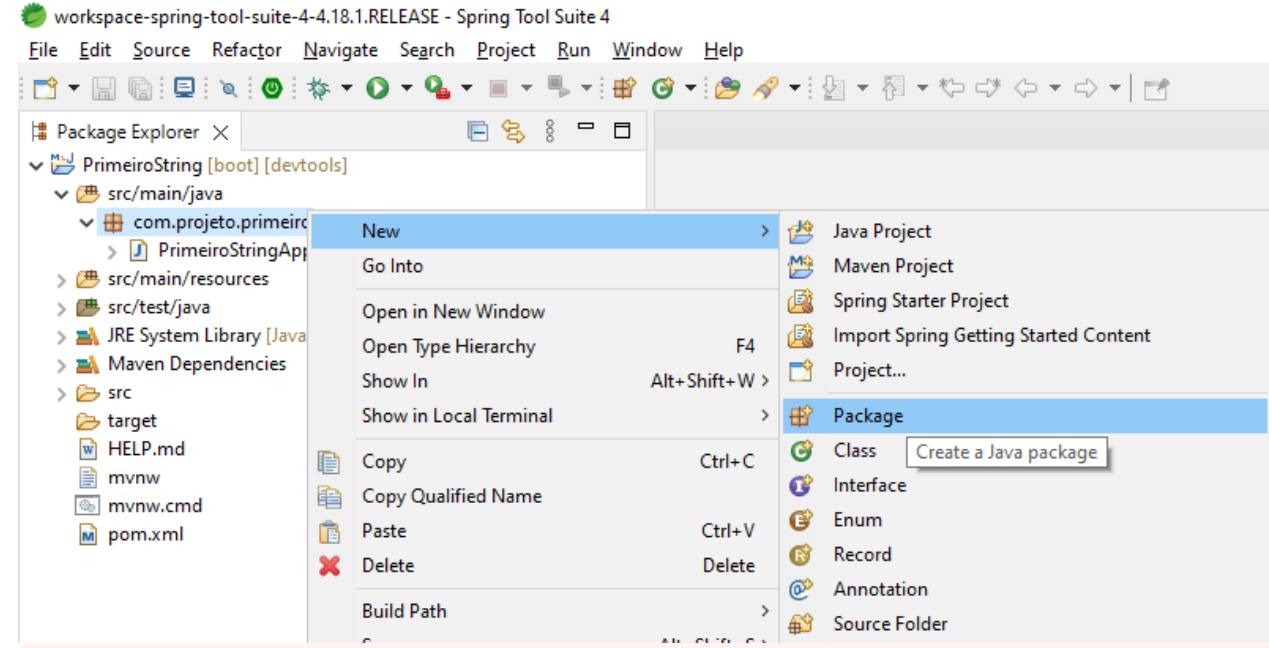
Next >

Finish

Cancel

Criar um novo projeto Spring

Crie um novo package no diretório com.projeto.primeiro com o nome de com.projeto.primeiro.controller



Criar um novo projeto Spring

Crie uma classe com o nome PrimeiroController no pacote controller

New Java Class

Java Class
Create a new Java class.

Source folder: PrimeiroString/src/main/java [Browse...](#)

Package: com.projeto.primeiro.controller [Browse...](#)

☐ Enclosing type: [Browse...](#)

Name: PrimeiroController

Modifiers: ☒ public ☐ package ☐ private ☐ protected
☐ abstract ☐ final ☐ static
☒ none ☐ sealed ☐ non-sealed ☐ final

Superclass: java.lang.Object [Browse...](#)

Interfaces: [Add...](#) [Remove](#)

Which method stubs would you like to create?

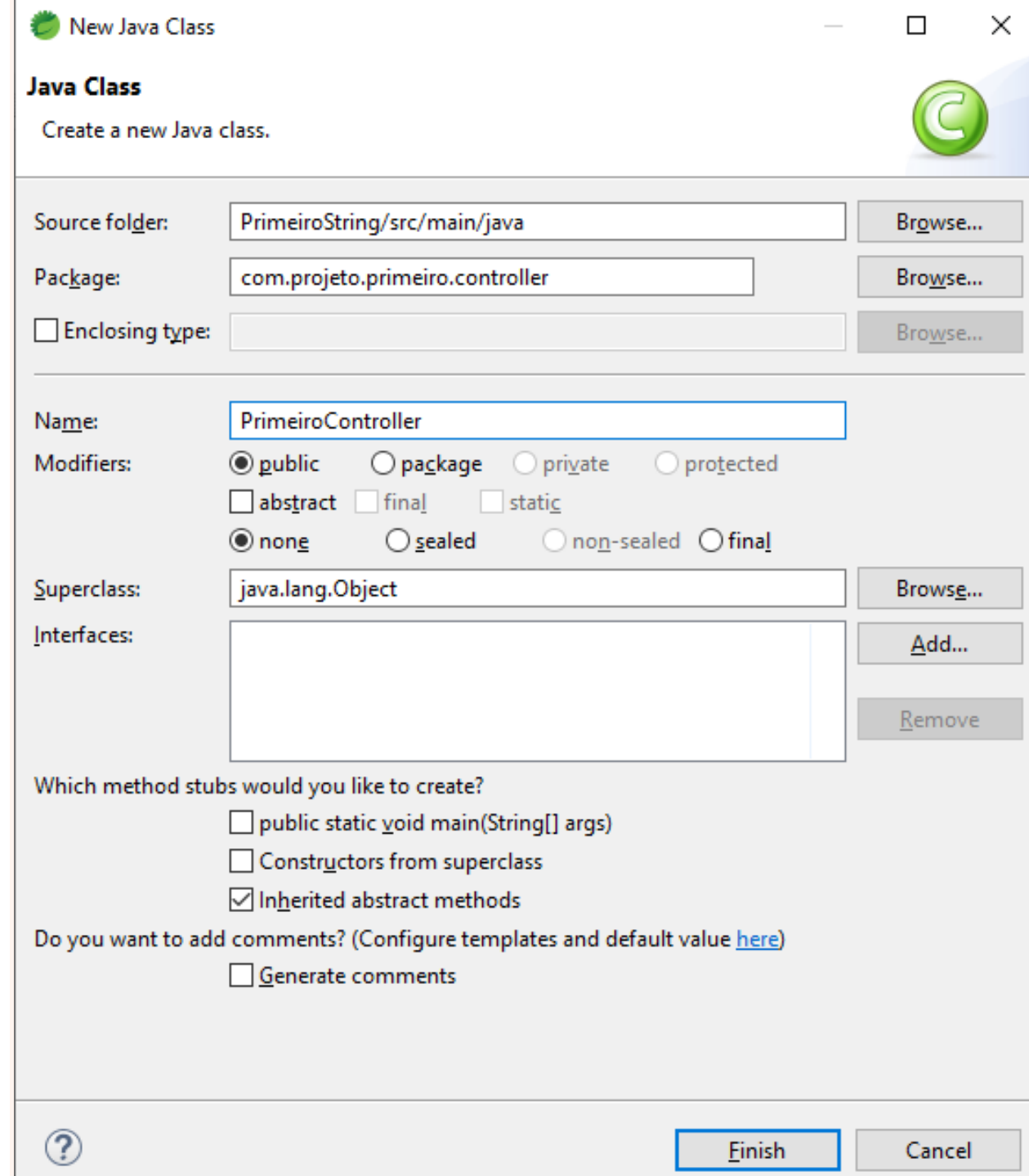
☐ public static void main(String[] args)
☐ Constructors from superclass
☒ Inherited abstract methods

Do you want to add comments? (Configure templates and default value [here](#))
☐ Generate comments

[?](#) [Finish](#) [Cancel](#)

Criar um novo projeto Spring

Crie uma classe com o nome PrimeiroController no pacote controller



New Java Class

Java Class

Create a new Java class.

Source folder: PrimeiroString/src/main/java [Browse...](#)

Package: com.projeto.primeiro.controller [Browse...](#)

☐ Enclosing type: [Browse...](#)

Name: PrimeiroController

Modifiers: ☒ public ☐ package ☐ private ☐ protected
☐ abstract ☐ final ☐ static
☒ none ☐ sealed ☐ non-sealed ☐ final

Superclass: java.lang.Object [Browse...](#)

Interfaces: [Add...](#) [Remove](#)

Which method stubs would you like to create?

☐ public static void main(String[] args)
☐ Constructors from superclass
☒ Inherited abstract methods

Do you want to add comments? (Configure templates and default value [here](#))
☐ Generate comments

[?](#) [Finish](#) [Cancel](#)

Criar um novo projeto Spring

Vamos usar anotações neste projeto

Spring Boot Annotations é uma forma de metadados que fornece dados sobre um programa. Em outras palavras, as anotações são usadas para fornecer informações **suplementares** sobre um programa. Não faz parte do aplicativo que desenvolvemos. Não tem efeito direto na operação do código que eles anotam. Não altera a ação do programa compilado.

```
PrimeiroController.java X
1 package com.projeto.primeiro.controller;
2
3 import org.springframework.web.bind.annotation.GetMapping;
4 import org.springframework.web.bind.annotation.RestController;
5
6 @RestController
7 public class PrimeiroController {
8
9     @GetMapping("/dados")
10    public String dados() {
11        return "Olá, tudo bem!";
12    }
13 }
14
```


Criar um novo projeto Spring

Vamos usar anotações neste projeto

@Required: Aplica-se ao método **bean** setter. Ele indica que o bean anotado deve ser preenchido no momento da configuração com a propriedade necessária, caso contrário, lançará uma exceção **BeanInitializationException**.

@Autowired: Spring fornece autoligação baseada em anotação fornecendo anotação **@Autowired**. Ele é usado para ligar automaticamente o spring bean em métodos setter, variável de instância e construtor. Quando usamos a anotação **@Autowired**, o contêiner de mola liga automaticamente o bean por tipo de dados correspondente.

@Configuration: É uma anotação em nível de classe. A classe anotada com **@Configuration** usada por Spring Containers como fonte de definições de bean.

@ComponentScan: É usado quando queremos escanear um pacote em busca de beans. É usado com a anotação **@Configuration**. Também podemos especificar os pacotes básicos para verificar os componentes do Spring.

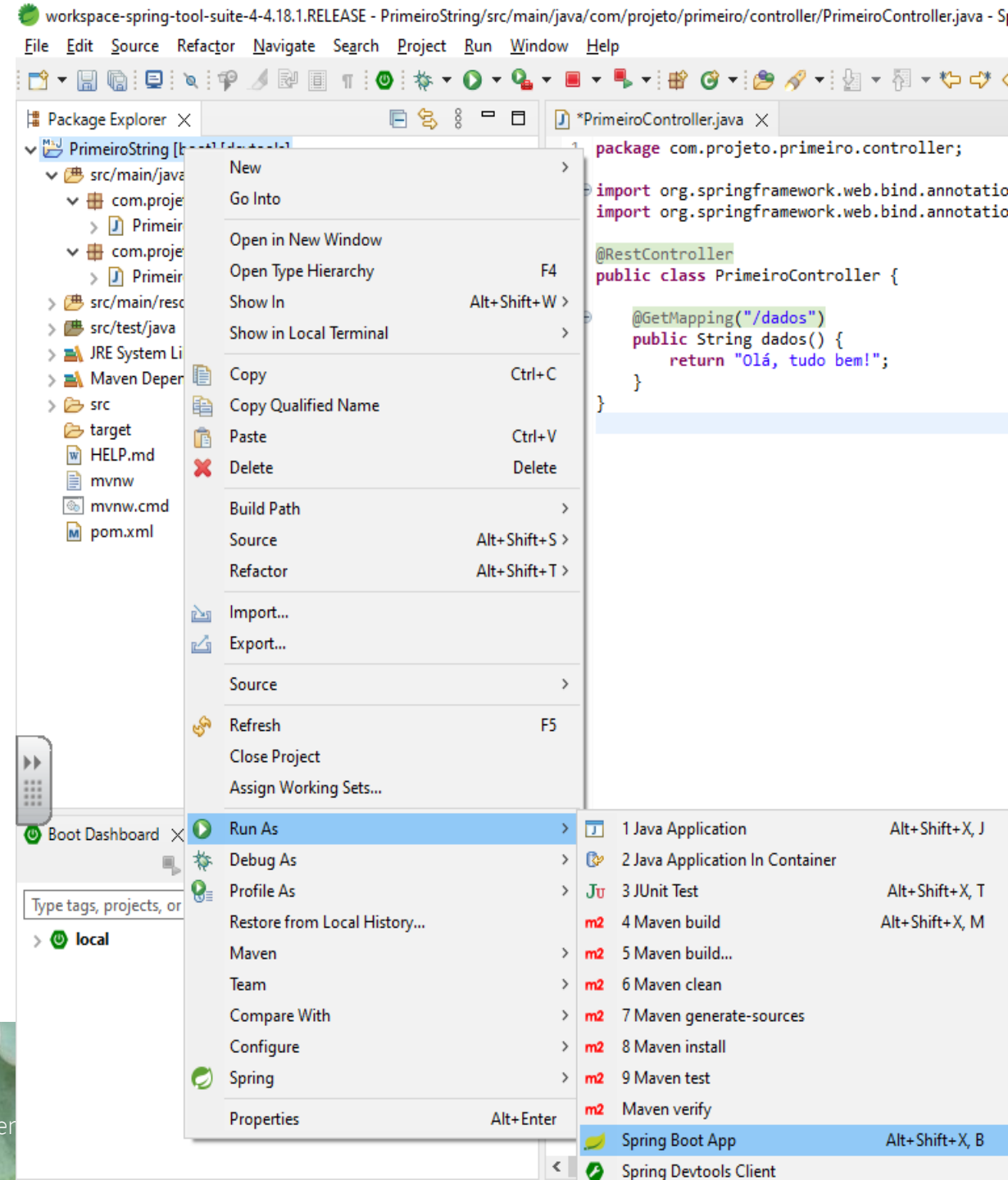
Vamos ao código do arquivo PrimeiroController

 *PrimeiroController.java ×

```
1 package com.projeto.primeiro.controller;
2
3 import org.springframework.web.bind.annotation.GetMapping;
4 import org.springframework.web.bind.annotation.RestController;
5
6 @RestController
7 public class PrimeiroController {
8
9     @GetMapping("/dados")
10    public String dados() {
11        return "Olá, tudo bem!";
12    }
13 }
14
```

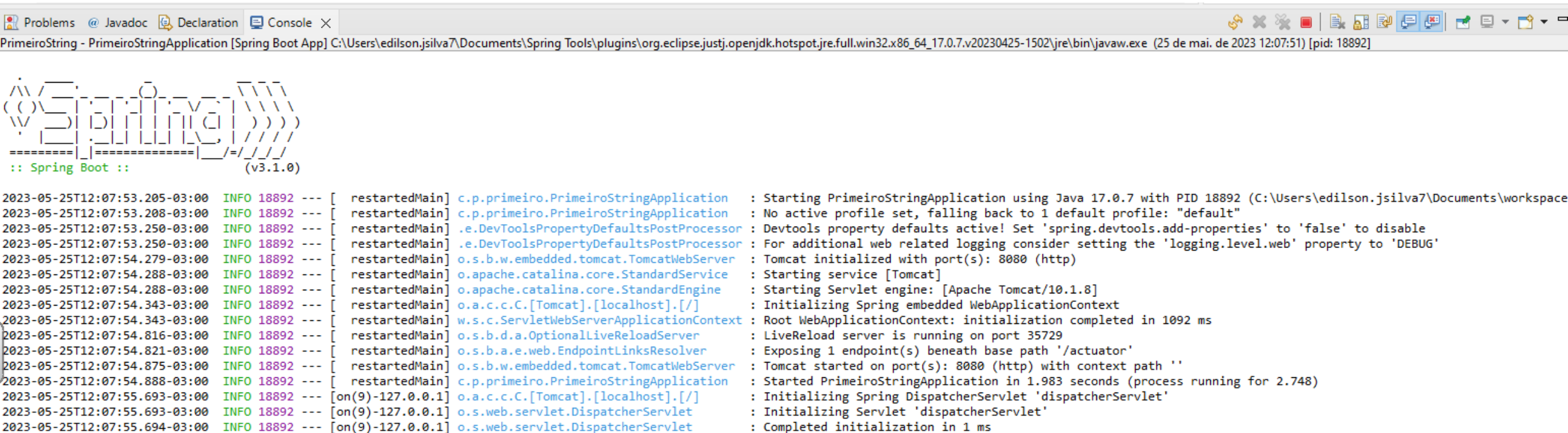
Vamos ao código do arquivo PrimeiroController

Para testar basta clicar com o botão direito do mouse sobre o nome do projeto e escolher a opção Run->SpringBoot App



Vamos ao código do arquivo PrimeiroController

O resultado da execução. Agora abra o navegador e digite `http://127.0.0.1:8080/dados`



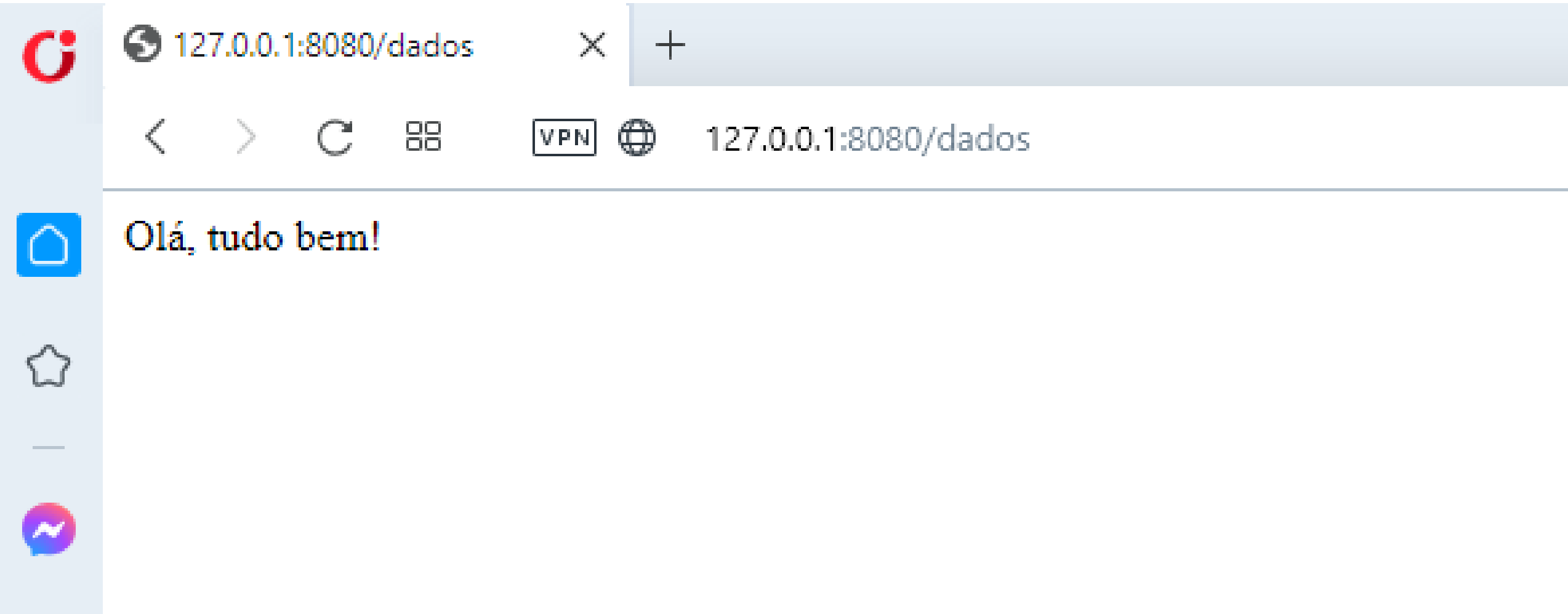
```
Problems @ Javadoc Declaration Console X
PrimeiroString - PrimeiroStringApplication [Spring Boot App] C:\Users\edilson.jsilva7\Documents\Spring Tools\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.7.v20230425-1502\jre\bin\javaw.exe (25 de mai. de 2023 12:07:51) [pid: 18892]

:: Spring Boot :: (v3.1.0)

2023-05-25T12:07:53.205-03:00 INFO 18892 --- [ restartedMain] c.p.primeiro.PrimeiroStringApplication : Starting PrimeiroStringApplication using Java 17.0.7 with PID 18892 (C:\Users\edilson.jsilva7\Documents\workspace
2023-05-25T12:07:53.208-03:00 INFO 18892 --- [ restartedMain] c.p.primeiro.PrimeiroStringApplication : No active profile set, falling back to 1 default profile: "default"
2023-05-25T12:07:53.250-03:00 INFO 18892 --- [ restartedMain] .e.DevToolsPropertyDefaultsPostProcessor : Devtools property defaults active! Set 'spring.devtools.add-properties' to 'false' to disable
2023-05-25T12:07:53.250-03:00 INFO 18892 --- [ restartedMain] .e.DevToolsPropertyDefaultsPostProcessor : For additional web related logging consider setting the 'logging.level.web' property to 'DEBUG'
2023-05-25T12:07:54.279-03:00 INFO 18892 --- [ restartedMain] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat initialized with port(s): 8080 (http)
2023-05-25T12:07:54.288-03:00 INFO 18892 --- [ restartedMain] o.apache.catalina.core.StandardService : Starting service [Tomcat]
2023-05-25T12:07:54.288-03:00 INFO 18892 --- [ restartedMain] o.apache.catalina.core.StandardEngine : Starting Servlet engine: [Apache Tomcat/10.1.8]
2023-05-25T12:07:54.343-03:00 INFO 18892 --- [ restartedMain] o.a.c.c.C.[Tomcat].[localhost].[/] : Initializing Spring embedded WebApplicationContext
2023-05-25T12:07:54.343-03:00 INFO 18892 --- [ restartedMain] w.s.c.ServletWebServerApplicationContext : Root WebApplicationContext: initialization completed in 1092 ms
2023-05-25T12:07:54.816-03:00 INFO 18892 --- [ restartedMain] o.s.b.d.a.OptionalLiveReloadServer : LiveReload server is running on port 35729
2023-05-25T12:07:54.821-03:00 INFO 18892 --- [ restartedMain] o.s.b.a.e.web.EndpointLinksResolver : Exposing 1 endpoint(s) beneath base path '/actuator'
2023-05-25T12:07:54.875-03:00 INFO 18892 --- [ restartedMain] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat started on port(s): 8080 (http) with context path ''
2023-05-25T12:07:54.888-03:00 INFO 18892 --- [ restartedMain] c.p.primeiro.PrimeiroStringApplication : Started PrimeiroStringApplication in 1.983 seconds (process running for 2.748)
2023-05-25T12:07:55.693-03:00 INFO 18892 --- [on(9)-127.0.0.1] o.a.c.c.C.[Tomcat].[localhost].[/] : Initializing Spring DispatcherServlet 'dispatcherServlet'
2023-05-25T12:07:55.693-03:00 INFO 18892 --- [on(9)-127.0.0.1] o.s.web.servlet.DispatcherServlet : Initializing Servlet 'dispatcherServlet'
2023-05-25T12:07:55.694-03:00 INFO 18892 --- [on(9)-127.0.0.1] o.s.web.servlet.DispatcherServlet : Completed initialization in 1 ms
```

Vamos ao código do arquivo PrimeiroController

Veja o resultado



Vamos ao código do arquivo PrimeiroController

Mude a classe PrimeiroController e crie novos endpoints

```
@GetMapping("/listaprodutos")  
public List<String> produtos(){  
    List<String> lst = new ArrayList<String>();  
    lst.add("Mouse");  
    lst.add("Teclado");  
    lst.add("Telefone");  
    return lst;  
}
```

Vamos ao código do arquivo PrimeiroController

Mude a classe PrimeiroController e crie novos endpoints

```
@GetMapping("/listanprodutos")
public List<List<String>> nprodutos(){
    List<List<String>> lst = new ArrayList<List<String>>();

    List<String> categoria = new ArrayList<String>();
    categoria.add("Informática");
    categoria.add("Material escritório");
    categoria.add("Escolar");

    List<String> produto = new ArrayList<String>();
    produto.add("Mouse");
    produto.add("Teclado");
    produto.add("Telefone");

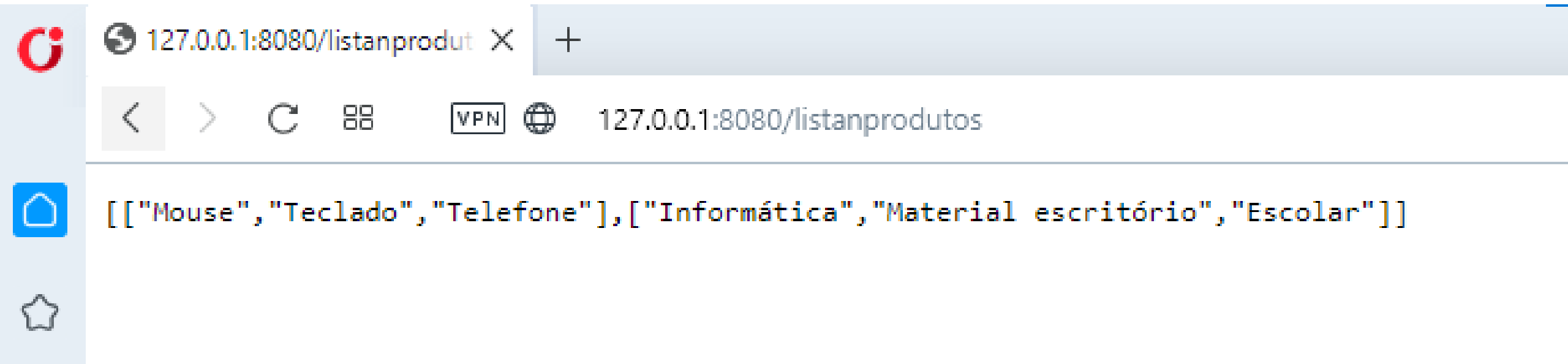
    lst.add(produto);
    lst.add(categoria);

    return lst;
}
```



Vamos ao código do arquivo PrimeiroController

Agora é testar usando os novos end points criados



Tente fazer

Seguindo a mesma ideia dos slides anteriores, crie novos end point chamados:

- /products

- /images

e teste

/products

```
PrimeiroController.java X
44 @GetMapping("/products")
45 public List<String> products(){
46     List<String> lst = new ArrayList<String>();
47
48     lst.add("id: 1");
49     lst.add("title: iPhone 9");
50     lst.add("description: An apple mobile which is nothing like apple");
51     lst.add("discountPercentage: 12.96");
52     lst.add("rating: 4.69");
53     lst.add("stock: 94");
54     lst.add("brand: Apple");
55     lst.add("category: smartphones");
56     lst.add("thumbnail: https://i.dummyjson.com/data/products/1/thumbnail.jpg");
57
58     lst.add("id: 2");
59     lst.add("title: iPhone X");
60     lst.add("description: SIM-Free, Model A19211 6.5-inch Super Retina HD display with OLED technology A12 Bionic chip with ...");
61     lst.add("price: 899");
62     lst.add("discountPercentage: 17.94");
63     lst.add("rating: 4.44");
64     lst.add("stock: 34");
65     lst.add("brand: Apple");
66     lst.add("category: smartphones");
67     lst.add("thumbnail: https://i.dummyjson.com/data/products/2/thumbnail.jpg");
68
69     lst.add("id: 3");
70     lst.add("title: Samsung Universe 9");
71     lst.add("description: Samsung's new variant which goes beyond Galaxy to the Universe");
72     lst.add("price: 1249");
73     lst.add("discountPercentage: 15.46");
74     lst.add("rating: 4.09");
75     lst.add("stock: 36");
76     lst.add("brand: Samsung");
77     lst.add("category: smartphones");
78     lst.add("thumbnail: https://i.dummyjson.com/data/products/3/thumbnail.jpg");
79
80     lst.add("id: 4");
81     lst.add("title: OPPO F19");
82     lst.add("description: OPPO F19 is officially announced on April 2021");
83     lst.add("price: 280");
84     lst.add("discountPercentage: 17.91");
85     lst.add("rating: 4.3");
86     lst.add("stock: 123");
87     lst.add("brand: OPPO");
88     lst.add("category: smartphones");
89     lst.add("thumbnail: https://i.dummyjson.com/data/products/4/thumbnail.jpg");
90     return lst;
```

/images

```
@GetMapping("/images")
public List<String> images(){
    List<String> lst = new ArrayList<String>();

    lst.add("https://i.dummyjson.com/data/products/1/1.jpg");
    lst.add("https://i.dummyjson.com/data/products/1/2.jpg");
    lst.add("https://i.dummyjson.com/data/products/1/3.jpg");
    lst.add("https://i.dummyjson.com/data/products/1/4.jpg");
    lst.add("https://i.dummyjson.com/data/products/1/thumbnail.jpg");

    lst.add("https://i.dummyjson.com/data/products/2/1.jpg");
    lst.add("https://i.dummyjson.com/data/products/2/2.jpg");
    lst.add("https://i.dummyjson.com/data/products/2/3.jpg");
    lst.add("https://i.dummyjson.com/data/products/2/thumbnail.jpg");

    lst.add("https://i.dummyjson.com/data/products/4/1.jpg");
    lst.add("https://i.dummyjson.com/data/products/4/2.jpg");
    lst.add("https://i.dummyjson.com/data/products/4/3.jpg");
    lst.add("https://i.dummyjson.com/data/products/4/4.jpg");
    lst.add("https://i.dummyjson.com/data/products/4/thumbnail.jpg");

    lst.add("https://i.dummyjson.com/data/products/6/1.png");
    lst.add("https://i.dummyjson.com/data/products/6/2.jpg");
    lst.add("https://i.dummyjson.com/data/products/6/3.png");
    lst.add("https://i.dummyjson.com/data/products/6/4.jpg");

    return lst;
}
```


Teste suas rotas