

FIAP GRADUAÇÃO

SISTEMAS DE INFORMAÇÃO

MICROSERVICE AND WEB ENGINEERING

Prof^a. Aparecida Castello Rosa
profaparecida.rosa@fiap.com.br

Material de apoio:
PROF. PEDRO IVO CORREIA
PROF. LUCAS FURLANETO

■ Agenda

- Criar projeto utilizando Spring initializr
- Criar as camadas Controller e View

Objetivos

- Criar projeto produto-mvc utilizando spring initializr
- Importar o projeto no Eclipse, STS ou **IntelliJ IDEA**
- Analisar as dependências no pom.xml
- Criar camada controller
- Criar camada View
- Entender a interação entre View e Controller
- Criar a Home page da aplicação

Criando projeto produto-mvc - Spring

Criando projeto produto-mvc - Spring



The image shows the Spring Initializr web interface. It is a form for creating a new Spring project. The form is divided into several sections, each with a red border. The 'Project' section has radio buttons for 'Gradle - Groovy', 'Gradle - Kotlin', and 'Maven' (selected). The 'Language' section has radio buttons for 'Java' (selected), 'Kotlin', and 'Groovy'. The 'Spring Boot' section has radio buttons for '3.3.0 (SNAPSHOT)', '3.3.0 (M1)', '3.2.3 (SNAPSHOT)', '3.2.2' (selected), and '3.1.9 (SNAPSHOT)', '3.1.8'. The 'Project Metadata' section has text input fields for 'Group' (br.com.fiap), 'Artifact' (produto-mvc), 'Name' (produto-mvc), 'Description' (Cadastro de Produtos), and 'Package name' (br.com.fiap.produto-mvc). It also has radio buttons for 'Packaging' ('Jar' selected, 'War' unselected) and 'Java' ('21' unselected, '17' selected). The 'Dependencies' section has a button 'ADD ... CTRL + B' and lists 'Spring Web' (WEB), 'Thymeleaf' (TEMPLATE ENGINES), and 'Spring Boot DevTools' (DEVELOPER TOOLS). At the bottom, there are three buttons: 'GENERATE CTRL + G' (with a mouse cursor), 'EXPLORE CTRL + SPACE', and 'SHARE...'.

Project

☐ Gradle - Groovy ☒ Gradle - Kotlin ☐ Maven

Language

☒ Java ☐ Kotlin ☐ Groovy

Spring Boot

☐ 3.3.0 (SNAPSHOT) ☐ 3.3.0 (M1) ☐ 3.2.3 (SNAPSHOT) ☒ 3.2.2 ☐ 3.1.9 (SNAPSHOT) ☐ 3.1.8

Project Metadata

Group

Artifact

Name

Description

Package name

Packaging ☒ Jar ☐ War

Java ☐ 21 ☒ 17

Dependencies ADD ... CTRL + B

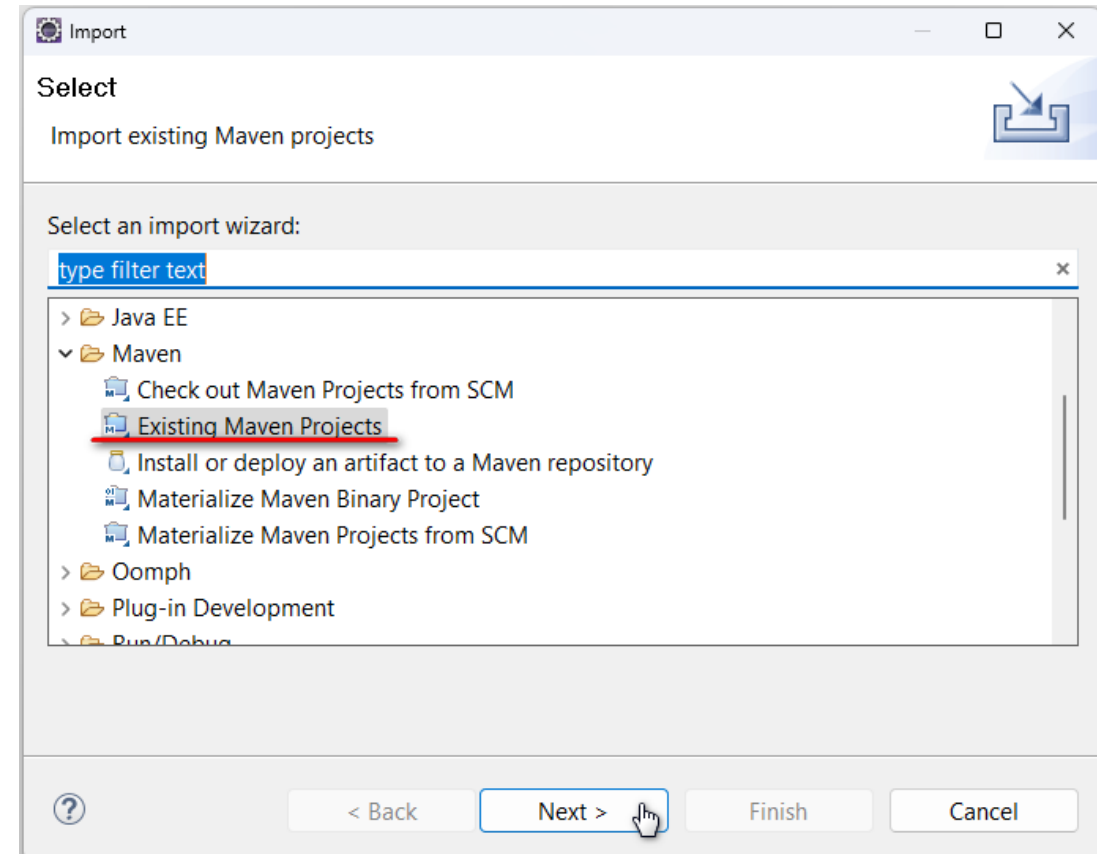
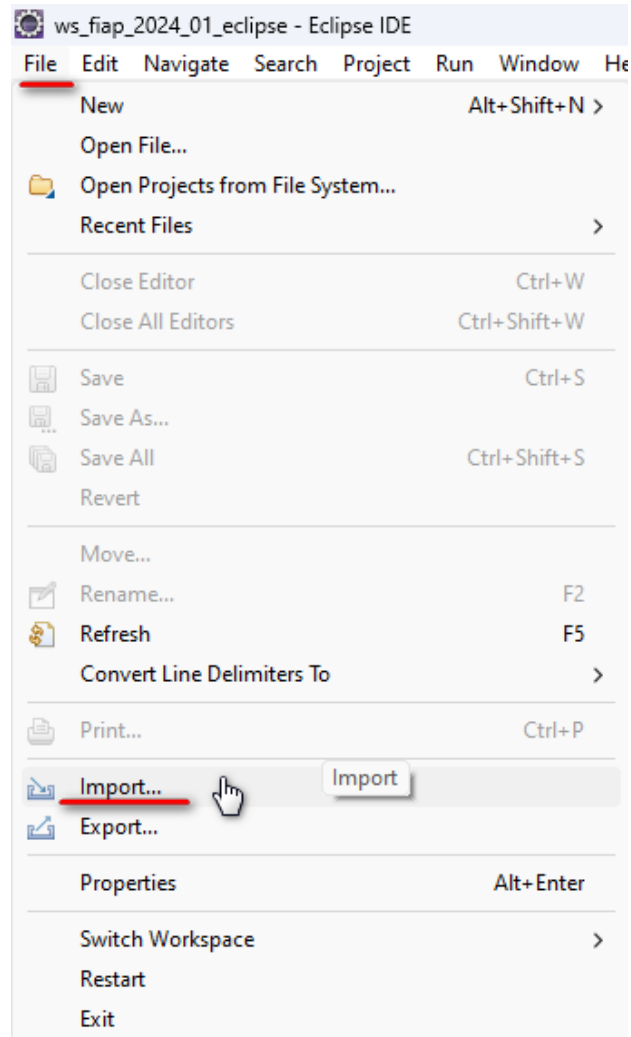
Spring Web WEB
Build web, including RESTful, applications using Spring MVC. Uses Apache Tomcat as the default embedded container.

Thymeleaf TEMPLATE ENGINES
A modern server-side Java template engine for both web and standalone environments. Allows HTML to be correctly displayed in browsers and as static prototypes.

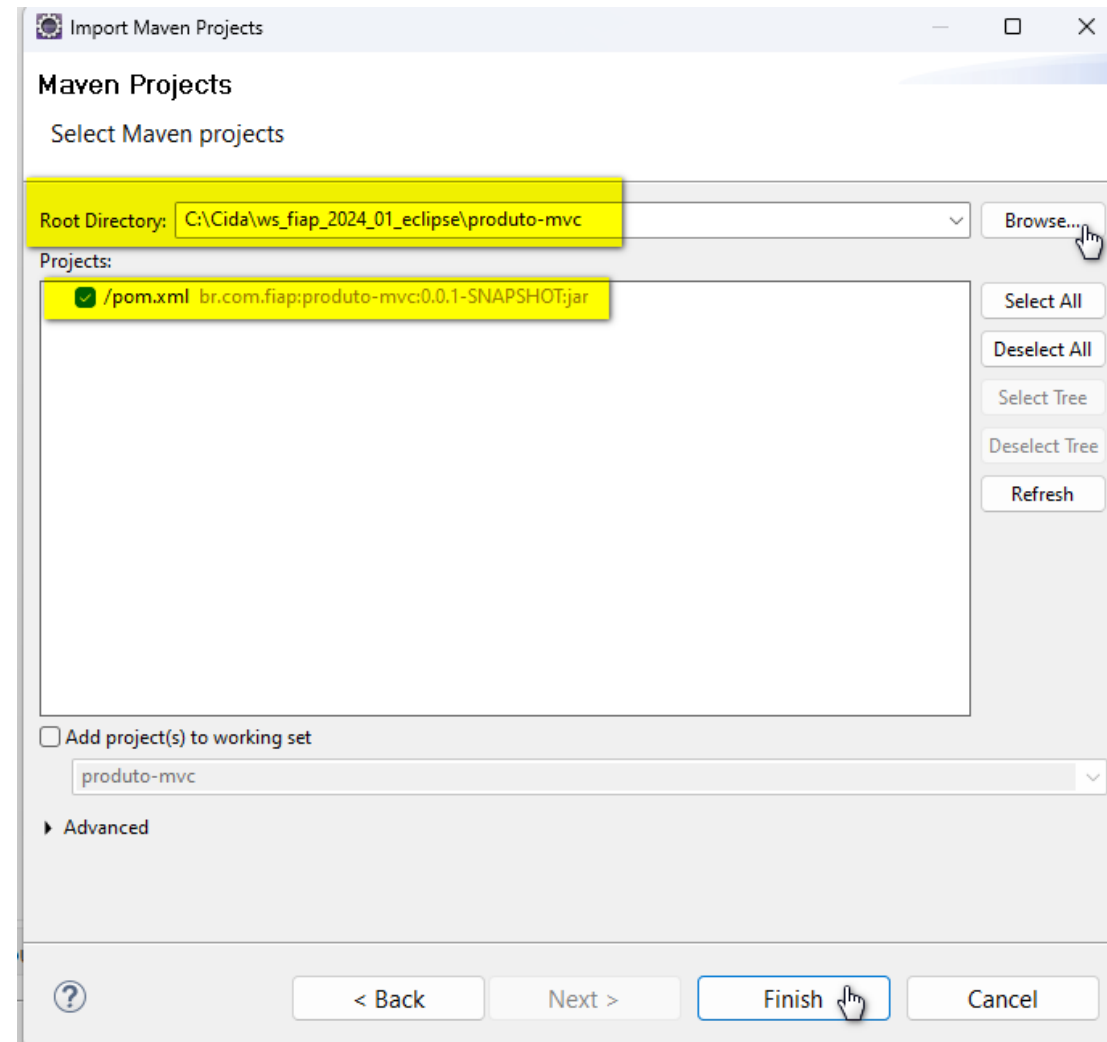
Spring Boot DevTools DEVELOPER TOOLS
Provides fast application restarts, LiveReload, and configurations for enhanced development experience.

Buttons: GENERATE CTRL + G, EXPLORE CTRL + SPACE, SHARE...

Importando o projeto no Eclipse e STS

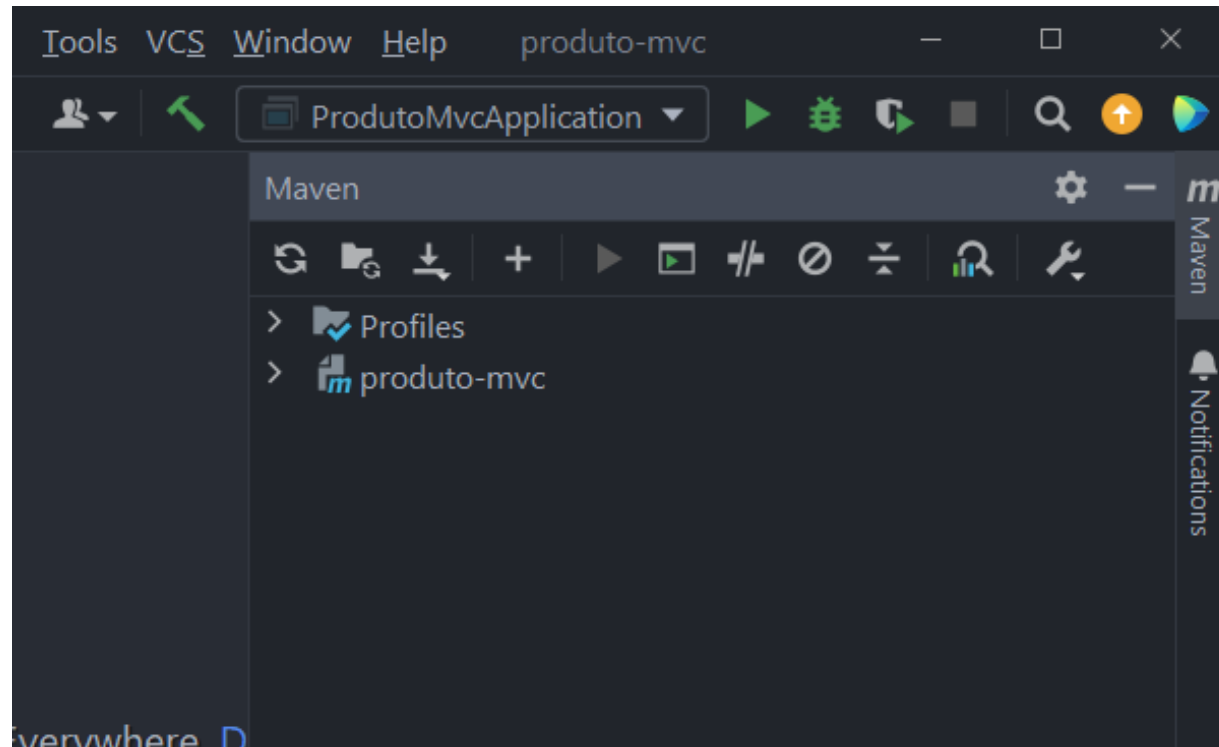


Importando o projeto no Eclipse e STS

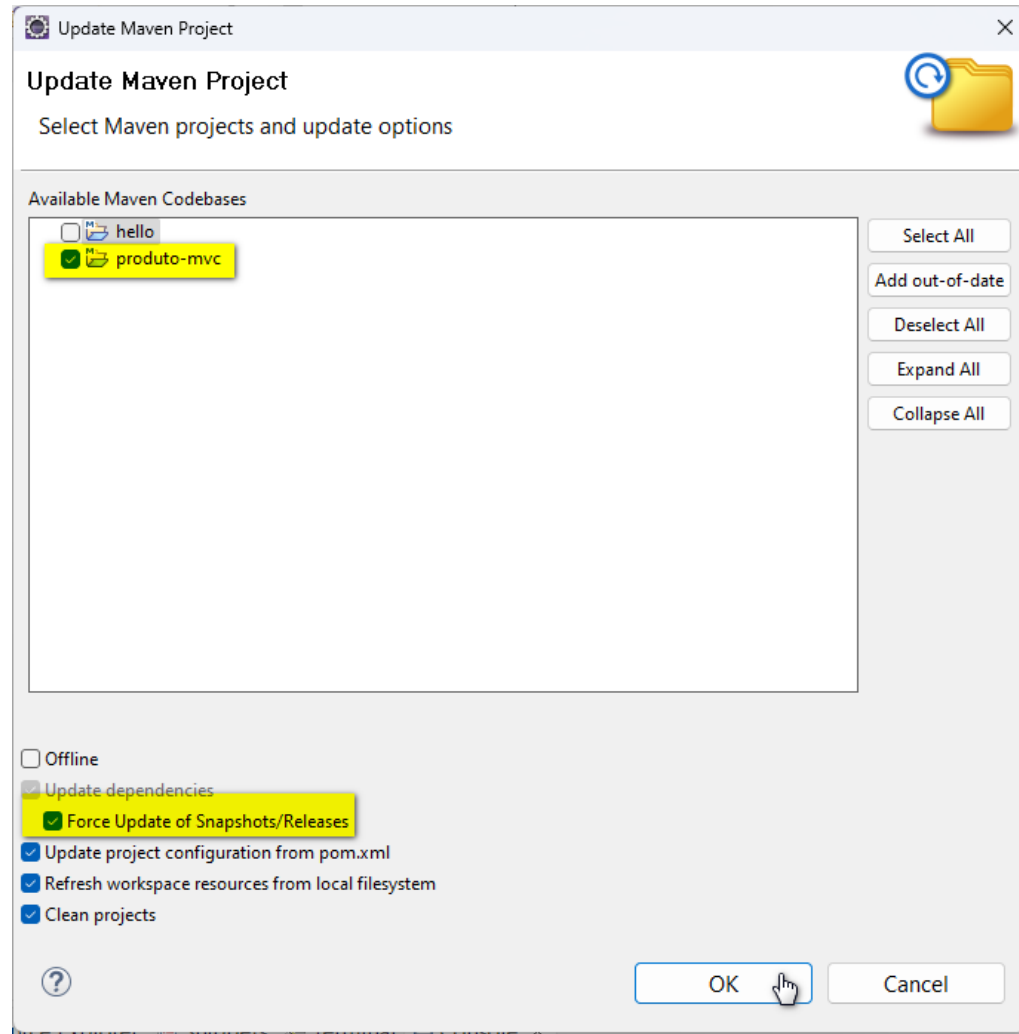
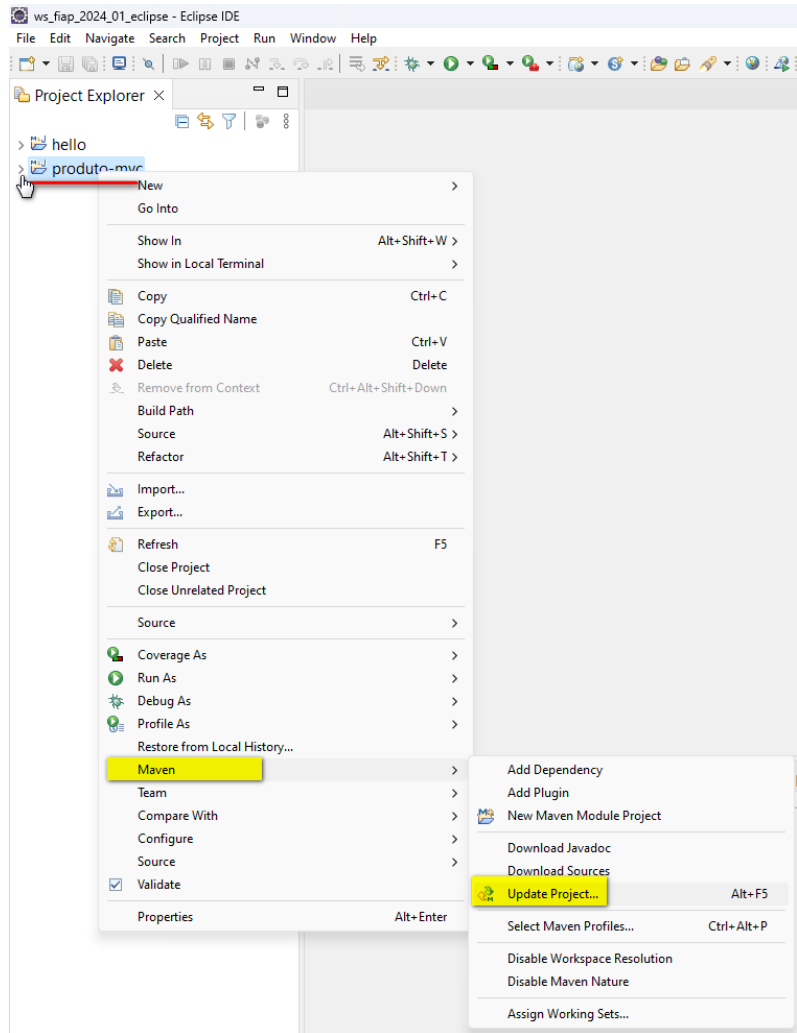


■ Atualizar Maven – IntelliJ IDEA

- Quando realizamos modificações no **pom.xml**, é necessário atualizá-lo, usando o Maven.

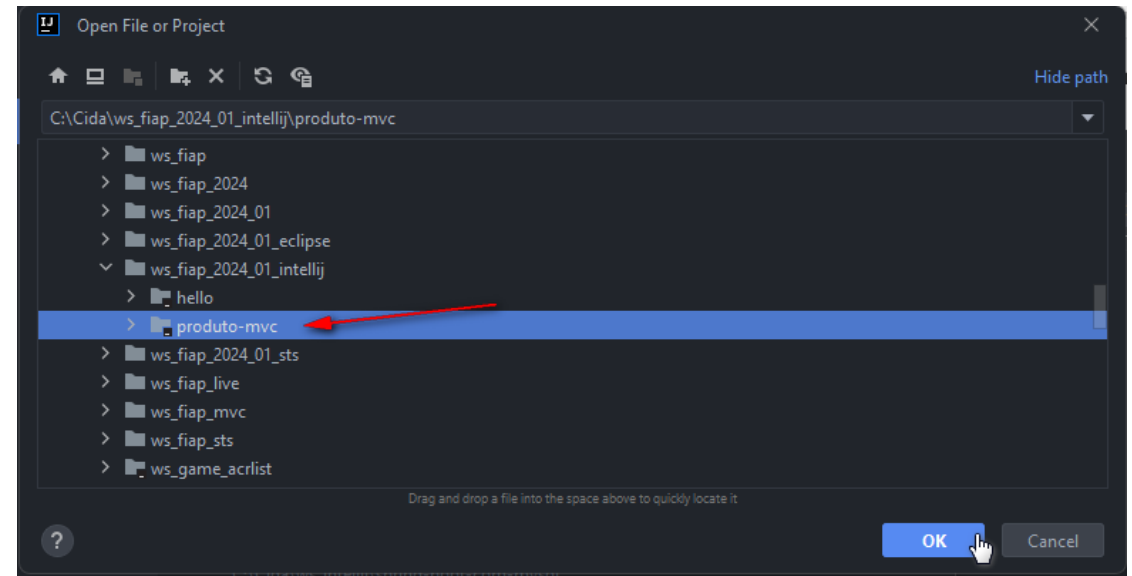
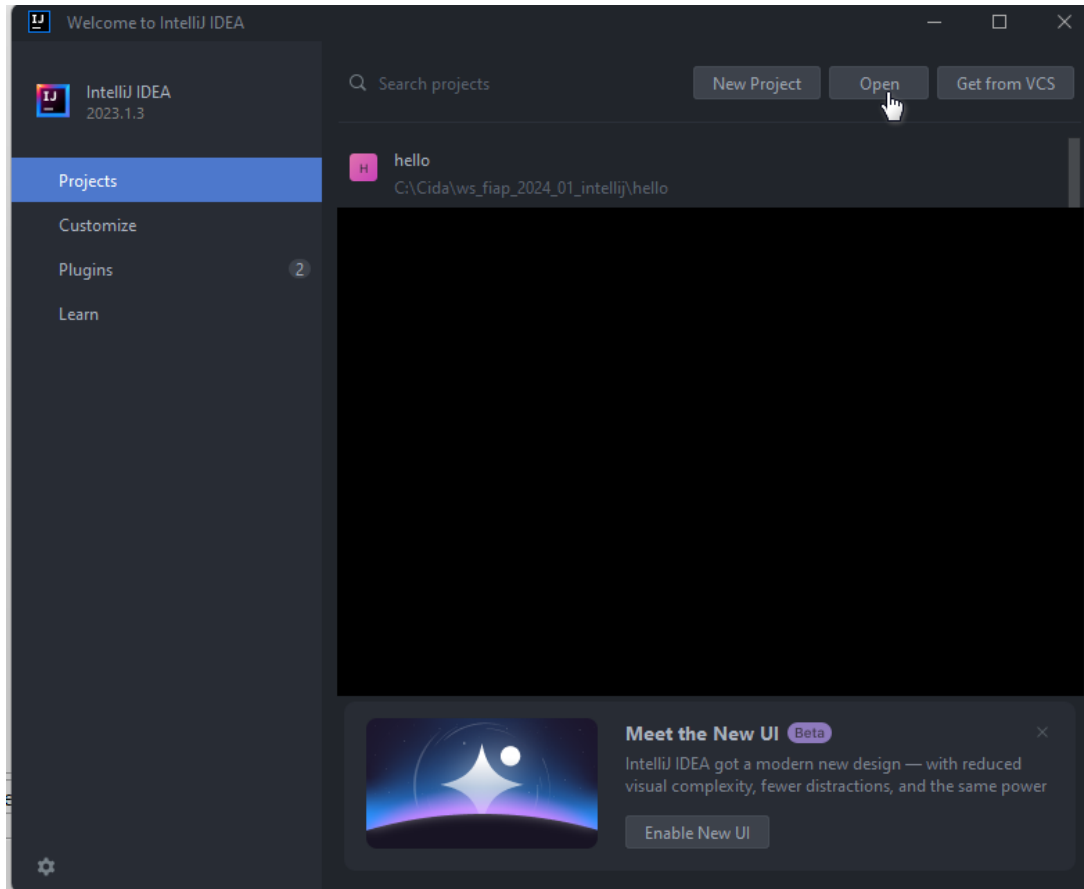


Atualizar Maven – Eclipse e STS

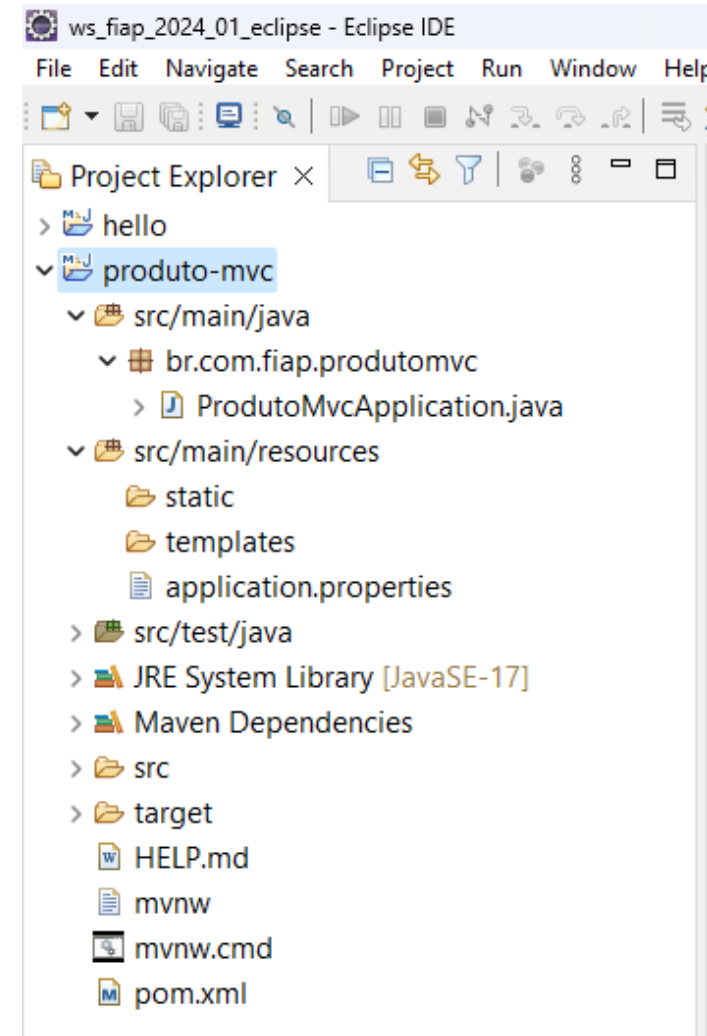
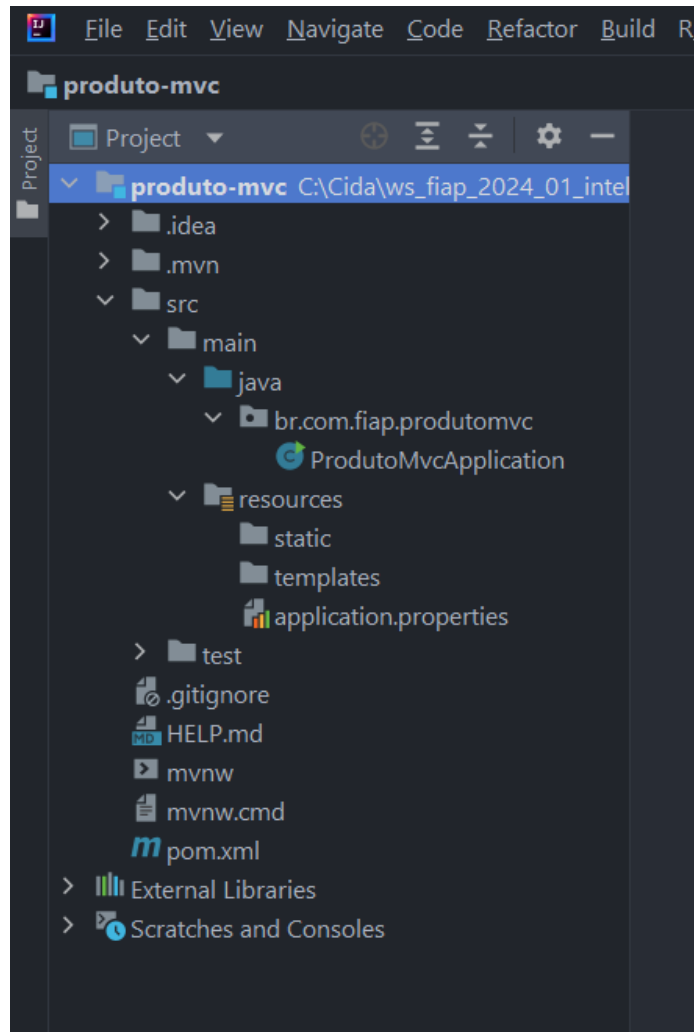


IntelliJ IDEA

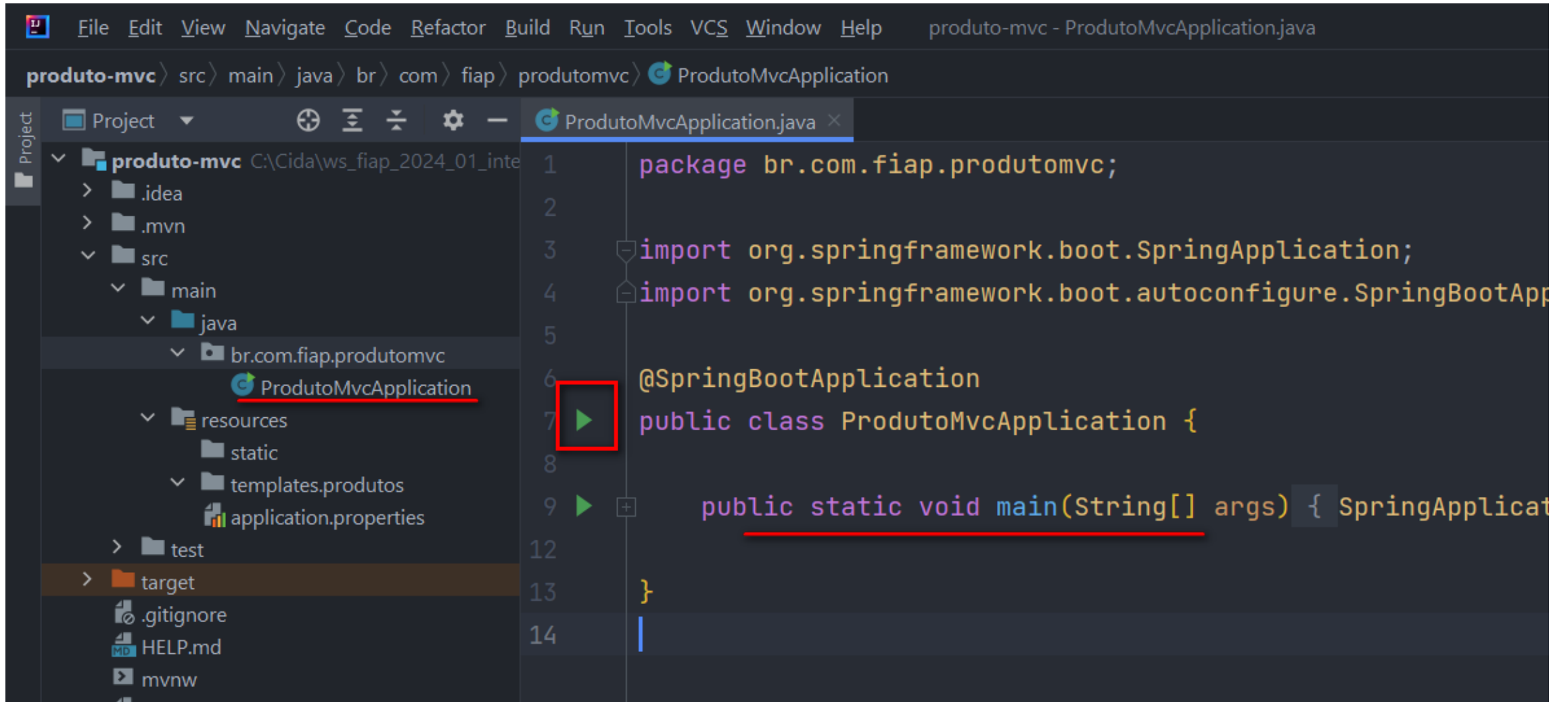
Abrindo o projeto com IntelliJ IDEA



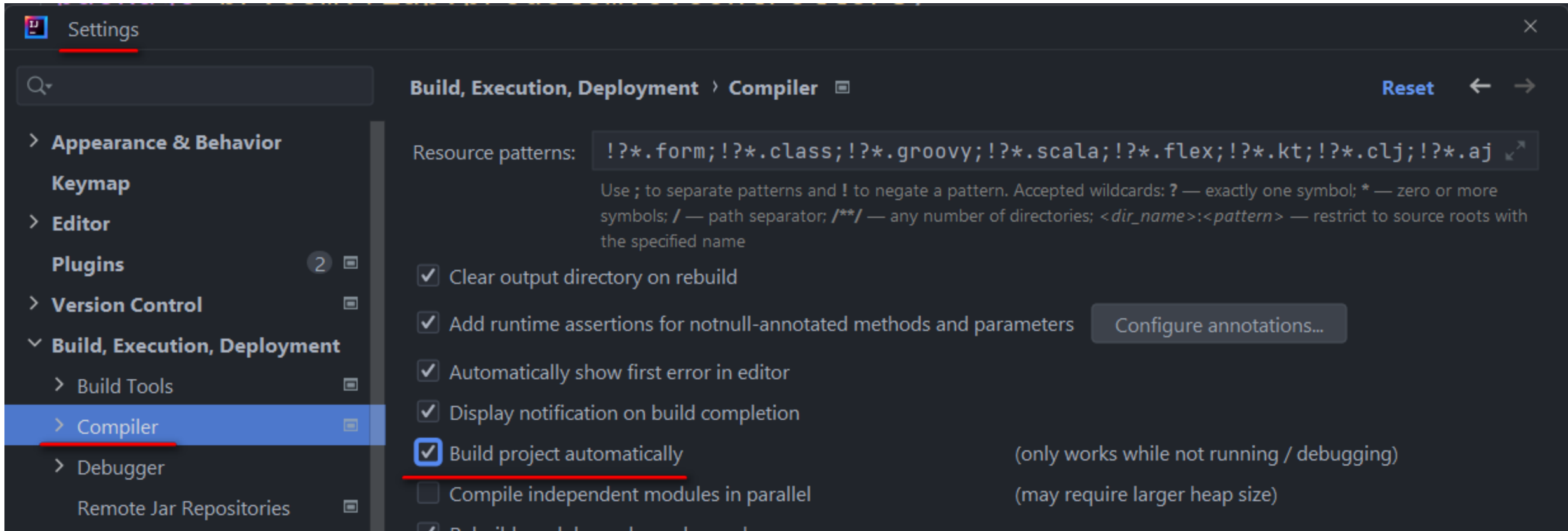
IntelliJ x Eclipse e STS



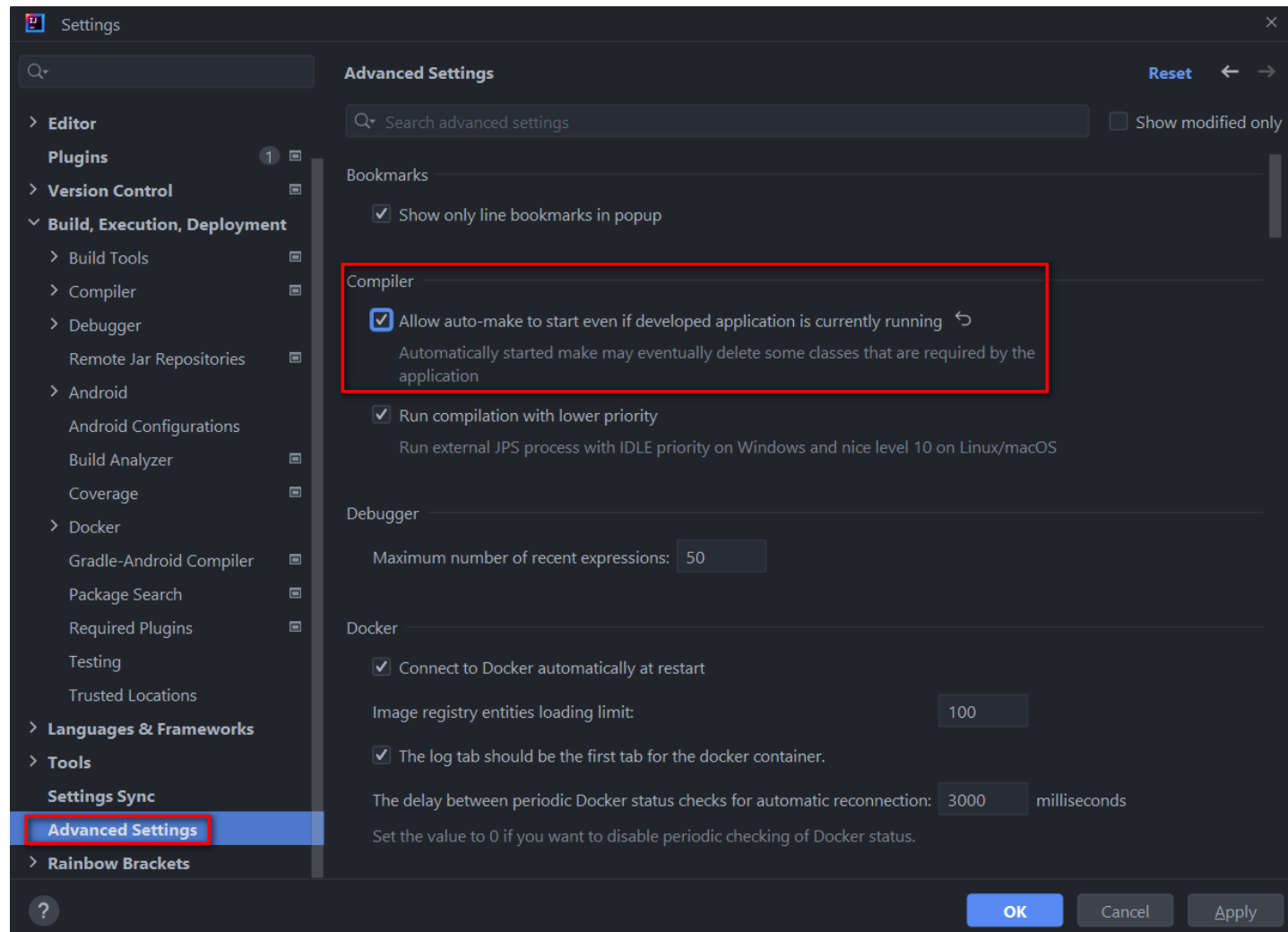
Executar a aplicação IntelliJ IDEA



Configurações IntelliJ para DevTools

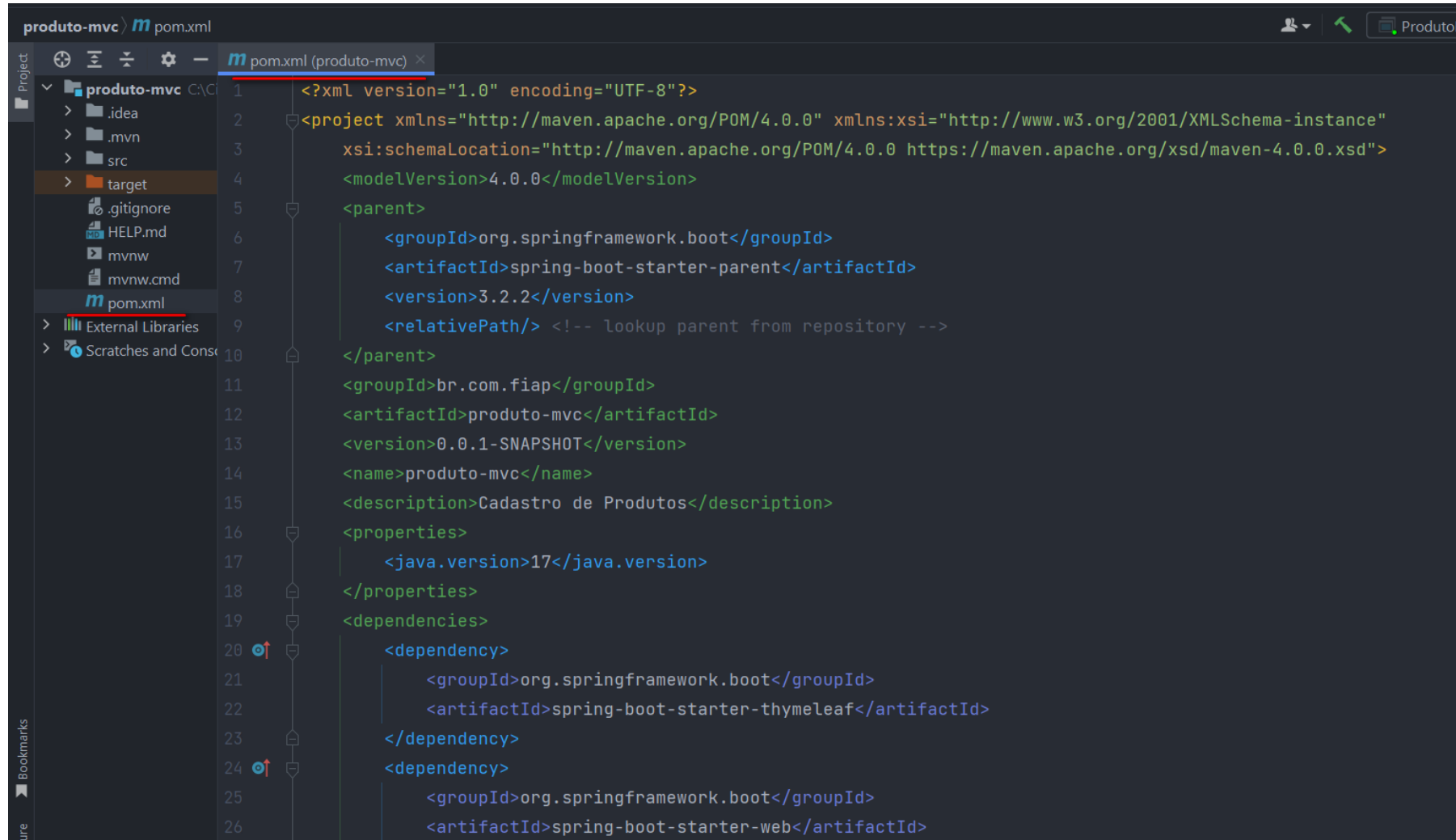


Configuração IntelliJ para DevTools



pom.xml

Examinar pom.xml



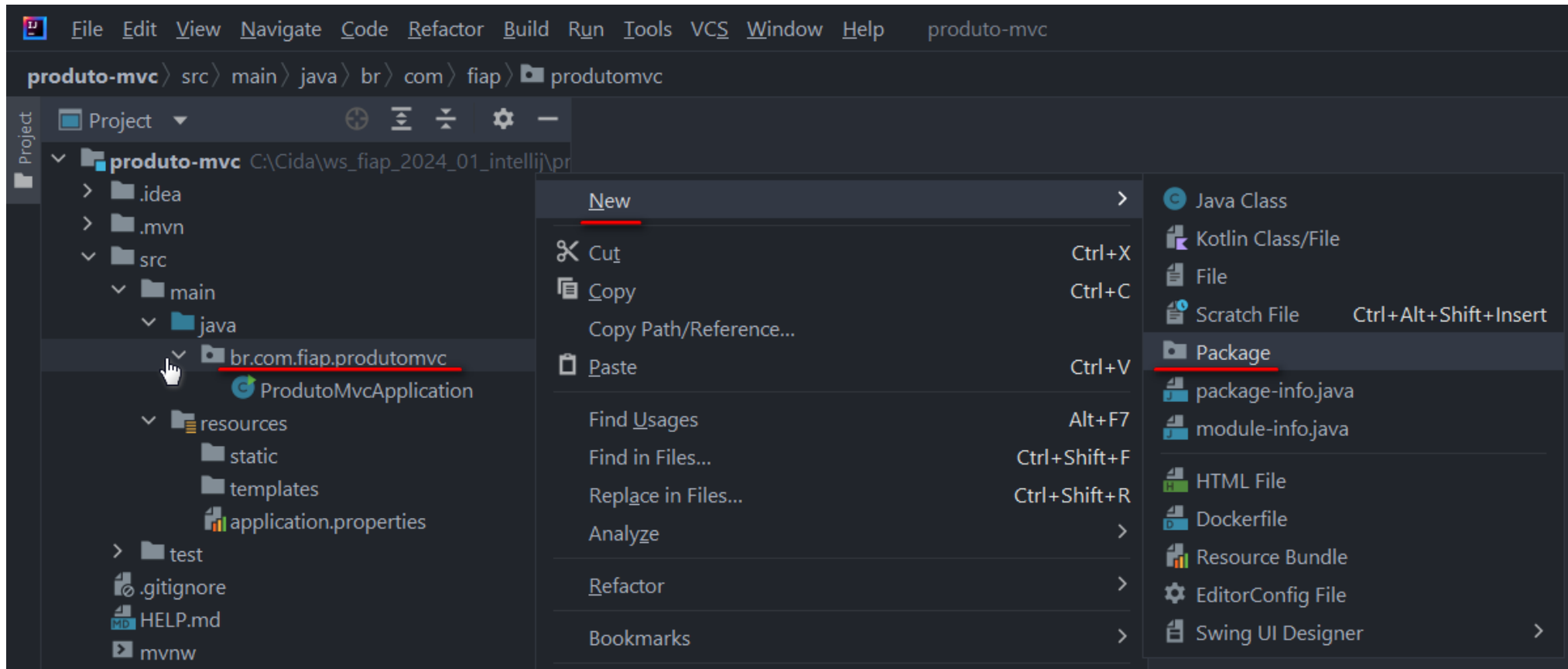
The screenshot shows an IDE window with the file 'pom.xml' open. The left sidebar displays the project structure for 'produto-mvc', including folders like '.idea', '.mvn', 'src', 'target', and files like '.gitignore', 'HELP.md', 'mvnw', 'mvnw.cmd', and 'pom.xml'. The main editor area shows the XML content of 'pom.xml' with line numbers 1 through 26. The XML defines a Maven project with the following details:

- XML version: 1.0, encoding: UTF-8
- Project namespace: http://maven.apache.org/POM/4.0.0
- XSI schema location: http://maven.apache.org/POM/4.0.0 https://maven.apache.org/xsd/maven-4.0.0.xsd
- Model version: 4.0.0
- Parent: org.springframework.boot:spring-boot-starter-parent:3.2.2
- Group ID: br.com.fiap
- Artifact ID: produto-mvc
- Version: 0.0.1-SNAPSHOT
- Name: produto-mvc
- Description: Cadastro de Produtos
- Properties: java.version=17
- Dependencies: spring-boot-starter-thymeleaf and spring-boot-starter-web

Controller

Criando a camada controller

Criar Package Controllers

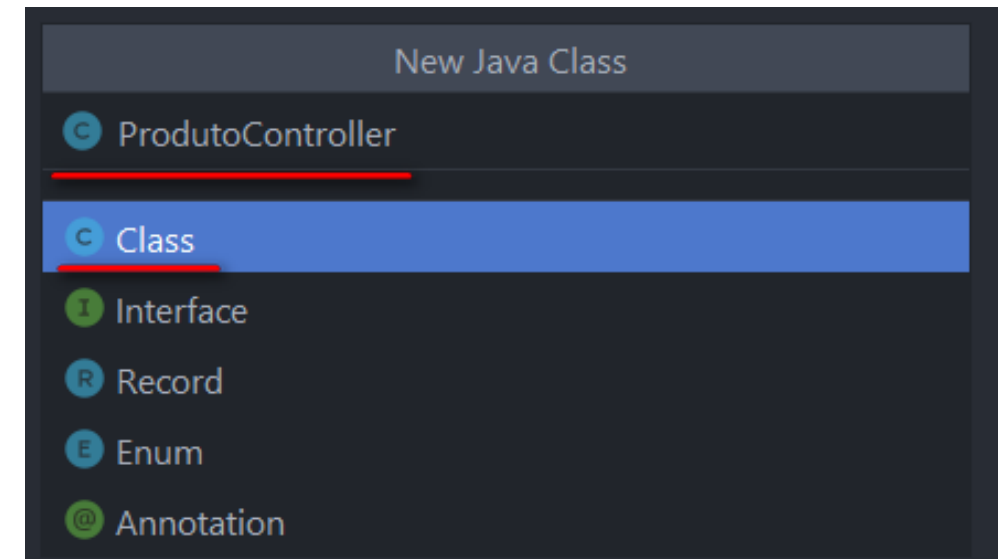
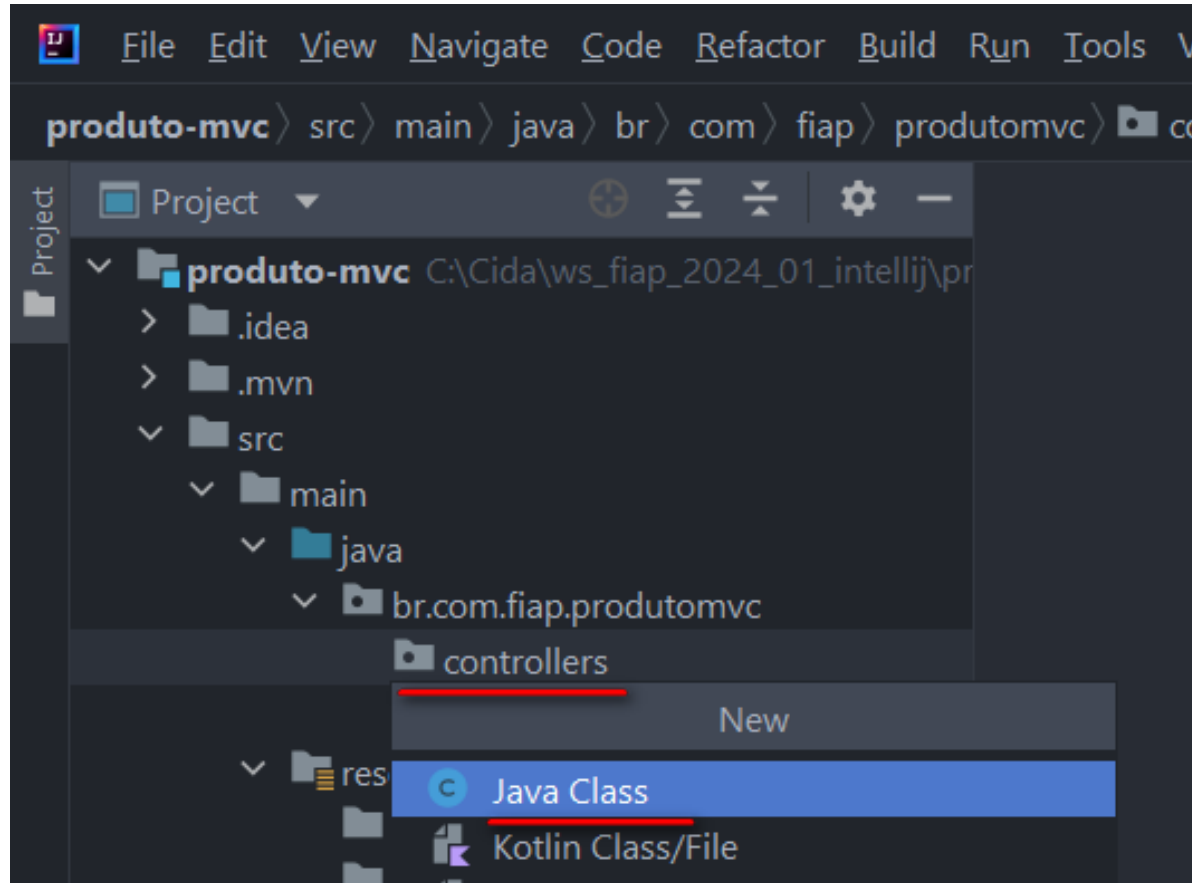


New Package

`br.com.fiap.producomvc.controllers`

■ Criar Class ProdutoController

Tecla de atalho: ALT + Insert



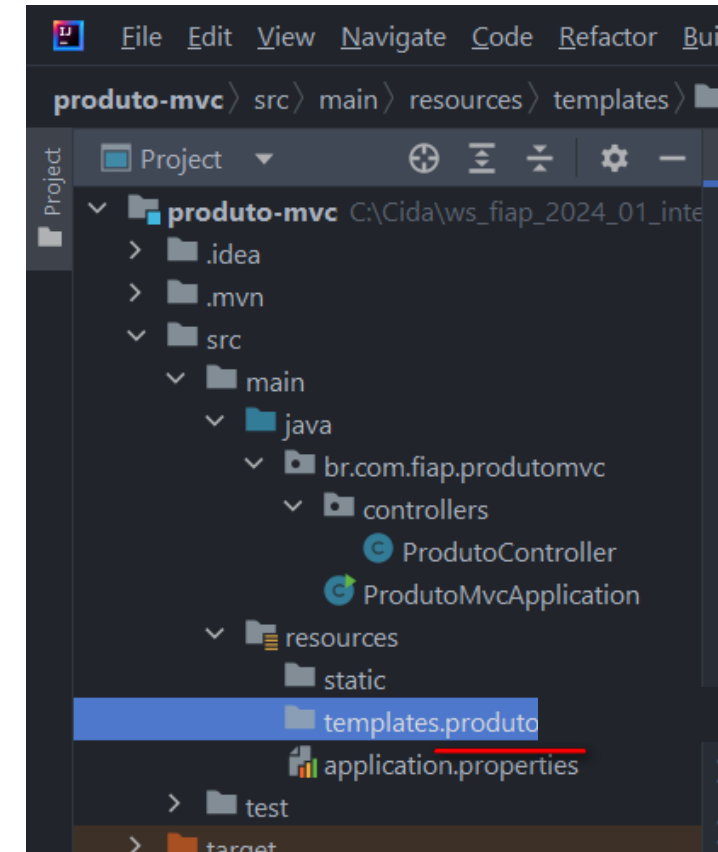
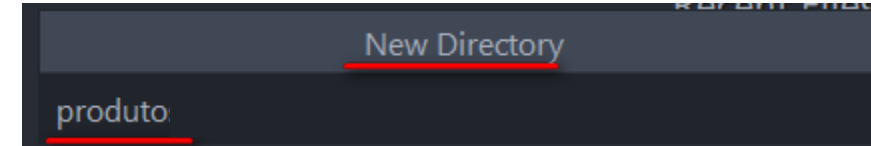
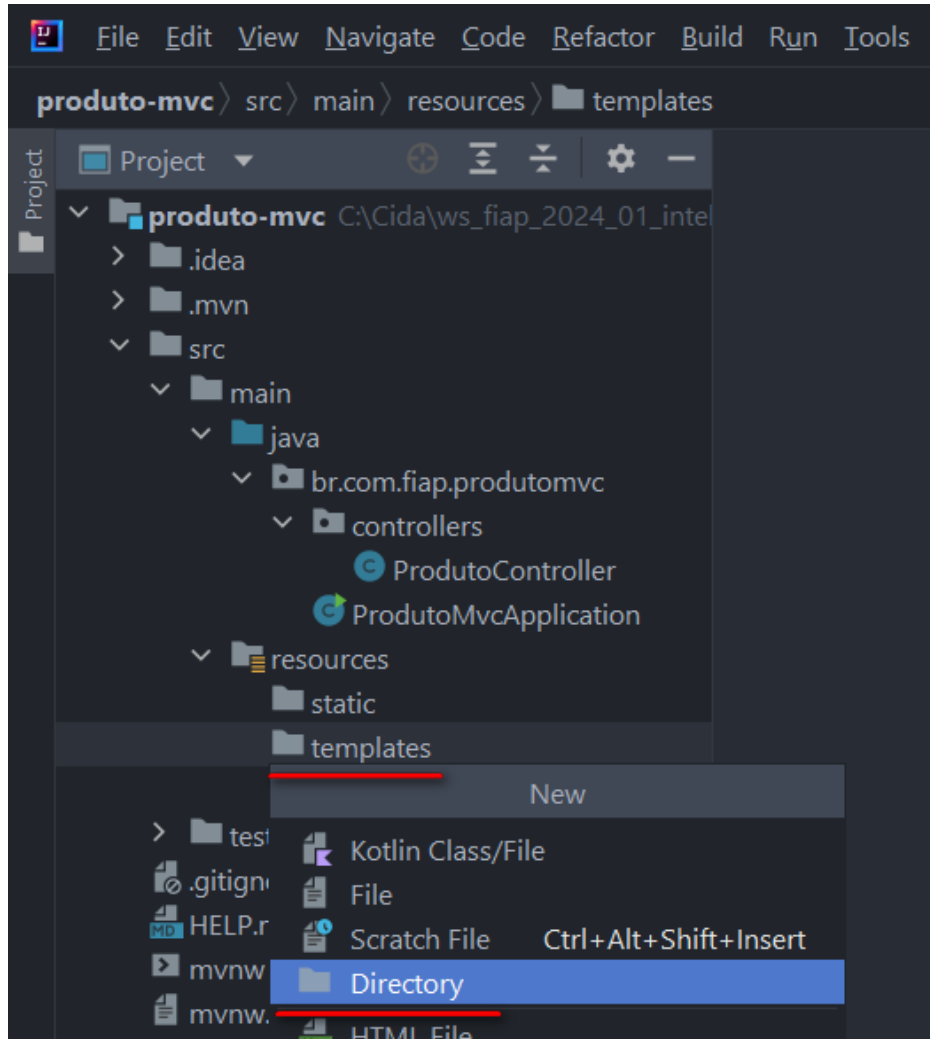
■ Criar Class ProdutoController

```
ProdutoController.java x
1  package br.com.fiap.produtomvc.controllers;
2
3  import org.springframework.stereotype.Controller;
4  import org.springframework.web.bind.annotation.GetMapping;
5  import org.springframework.web.bind.annotation.PostMapping;
6  import org.springframework.web.bind.annotation.RequestMapping;
7
8  //URL - localhost:8080/produtos
9
10 @Controller // Gerenciado pelo Spring
11 @RequestMapping("/produtos") //mapeando URL
12 public class ProdutoController {
13
14     // Método ou verbo HTTP
15     @GetMapping
16     public String adicionarProduto(){
17
18         // As Views ficam dentro da pasta templates
19         //retorna caminho da página cadastro HTML
20         return "produto/novo-produto";
21     }
22 }
```

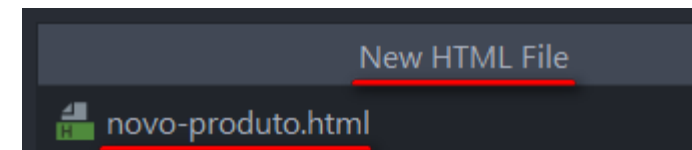
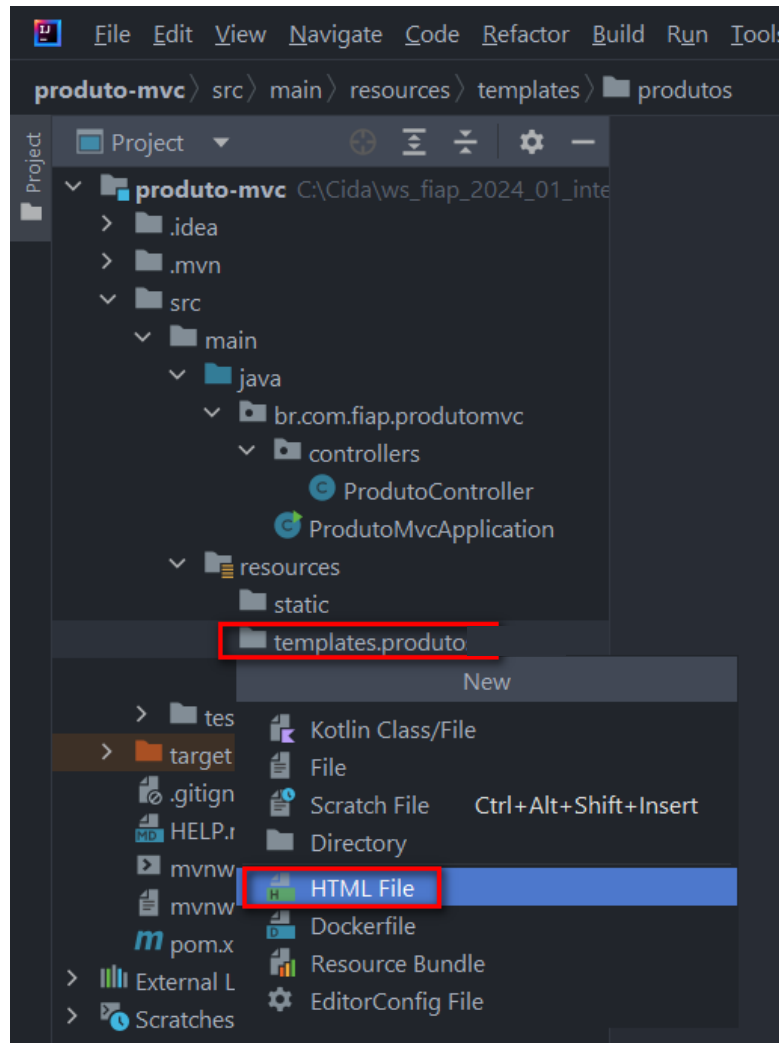
View

Criando a camada View

■ Criar diretório produtos em Templates



■ Criar página cadastro.html

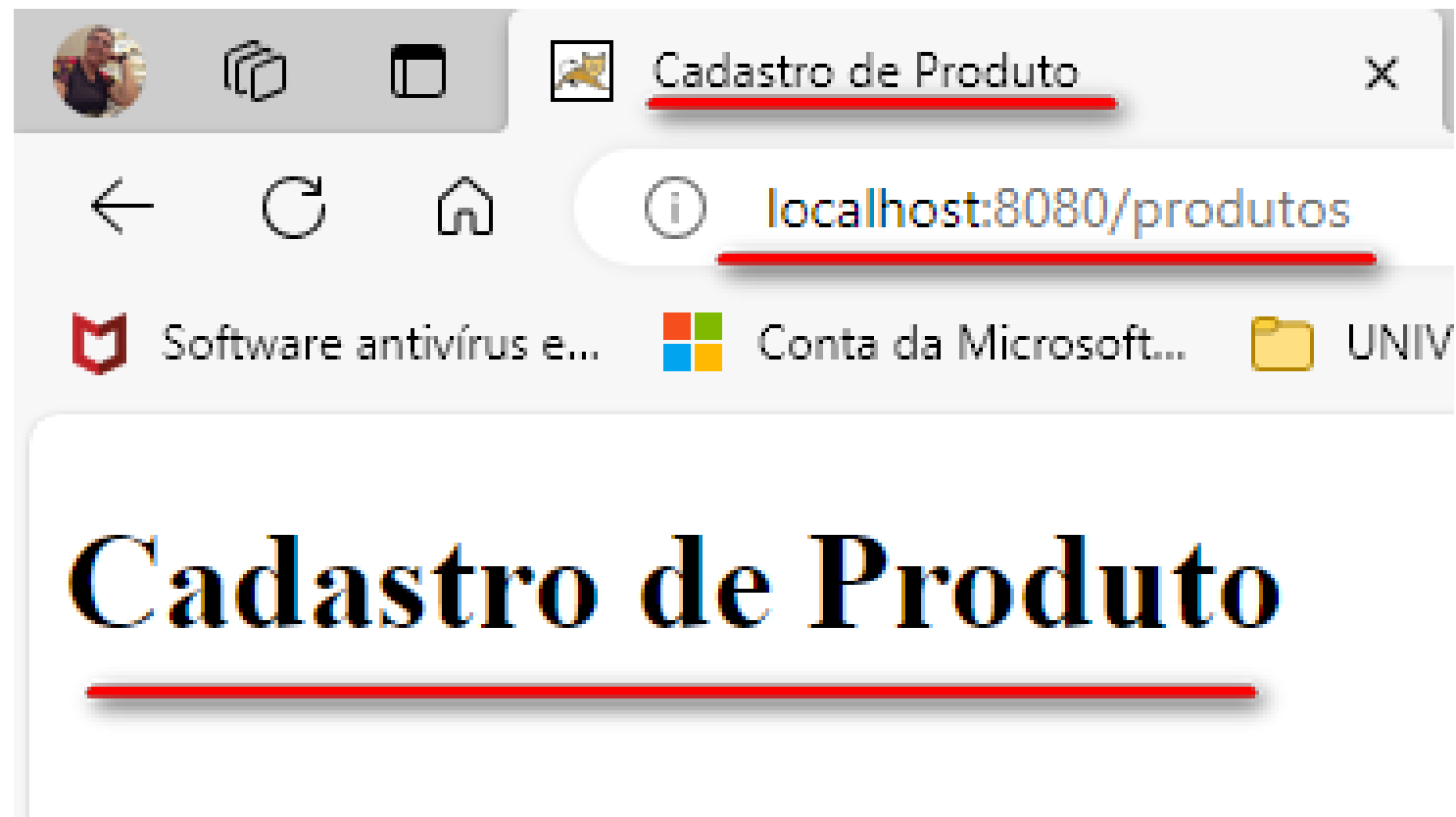


■ Criar página cadastro.html

```
1      <!DOCTYPE html>
2      <html lang="pt-br">
3      <head>
4          <meta charset="UTF-8">
5          <title>Cadastro de Produto</title>
6      </head>
7      <body>
8          <h1>Cadastro de Produto</h1>
9
10     </body>
11 </html>
```

■ Executar a aplicação

<http://localhost:8080/produtos>



■ Criar o formulário para cadastro de produtos FIAP

```
1  <!DOCTYPE html>
2  <html lang="pt-br">
3  <head>
4      <meta charset="UTF-8">
5      <title>Cadastro de Produto</title>
6  </head>
7  <body>
8      <h1>Cadastro de Produto</h1>
9      <form name="form-cadastro" method="post">
10         <label for="nome">Nome do Produto:</label>
11         <input id="nome" name="nome" placeholder="Nome do produto"> <br />
12         <label for="descricao">Descrição:</label>
13         <input id="descricao" name="descricao" placeholder="Descrição do produto">
14         <br />
15         <label for="valor">Valor - R$:</label>
16         <input type="number" id="valor" name="valor" placeholder="Valor do produto">
17         <br /><br />
18         <input type="submit" value="Cadastrar">
19     </form>
20 </body>
21 </html>
```

Produto

- id : Long
- nome : String
- descricao : String
- valor : Double

Testar a aplicação

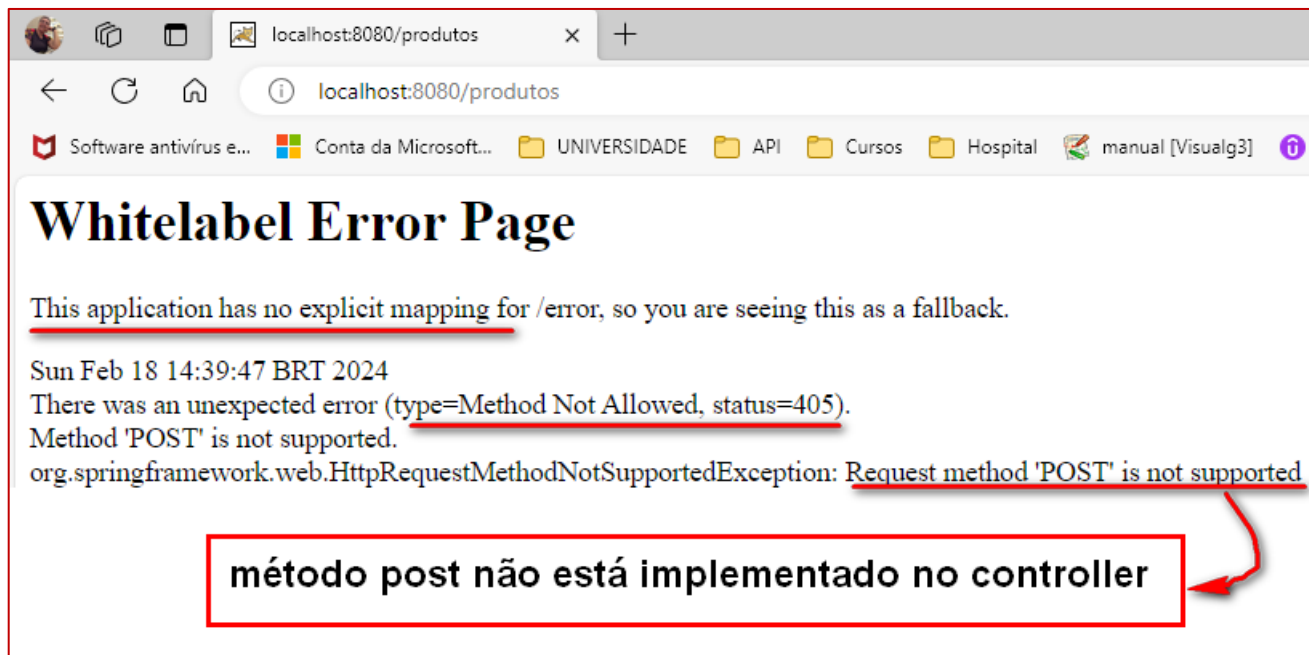


Cadastro de Produto

Nome do Produto:

Descrição:

Valor - R\$:



Whitelabel Error Page

This application has no explicit mapping for /error, so you are seeing this as a fallback.

Sun Feb 18 14:39:47 BRT 2024

There was an unexpected error (type=Method Not Allowed, status=405).

Method 'POST' is not supported.

org.springframework.web.HttpRequestMethodNotSupportedException: Request method 'POST' is not supported

método post não está implementado no controller

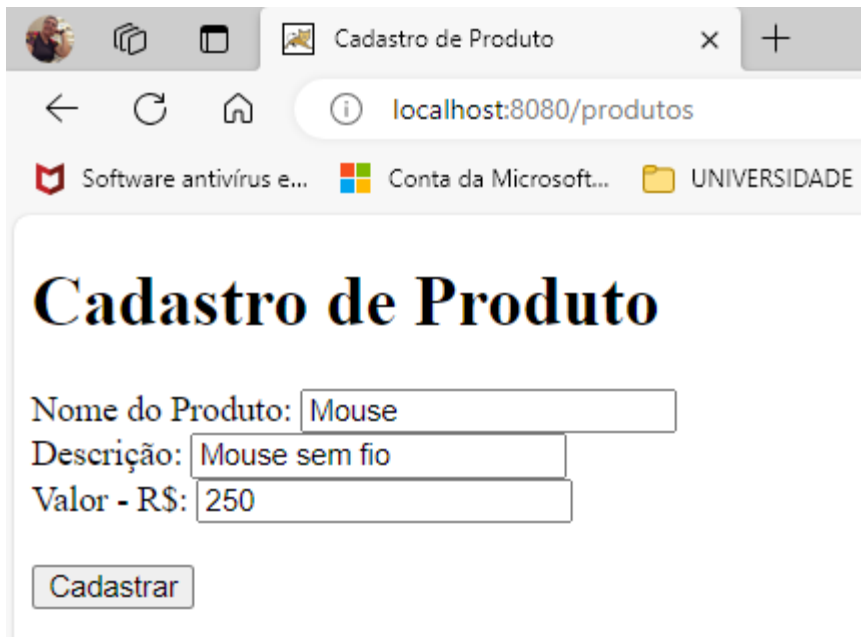
Alterar class ProdutoController

- Implementando o método PostMapping – provisoriamente

```
10 // -- código omitido --
11 @Controller // Gerenciado pelo Spring
12 @RequestMapping("/produtos") //mapeando URL
13 public class ProdutoController {
14
15     // Método ou verbo HTTP
16     @GetMapping
17     public String adicionarProduto(){
18
19         // As Views ficam dentro da pasta templates
20         //retorna caminho da página cadastro HTML
21         return "produto/novo-produto";
22     }
23
24     // receber dados da View para cadastrar produto
25     @PostMapping()
26     public String insertProduto(){
27         //provisório
28         return "redirect:/produtos";
29     }
30 }
```

Testar a aplicação

- Depois de clicar no botão Cadastrar, provisoriamente, carregamos novamente a página novo-produto.html, pelo redirecionamento da rota.

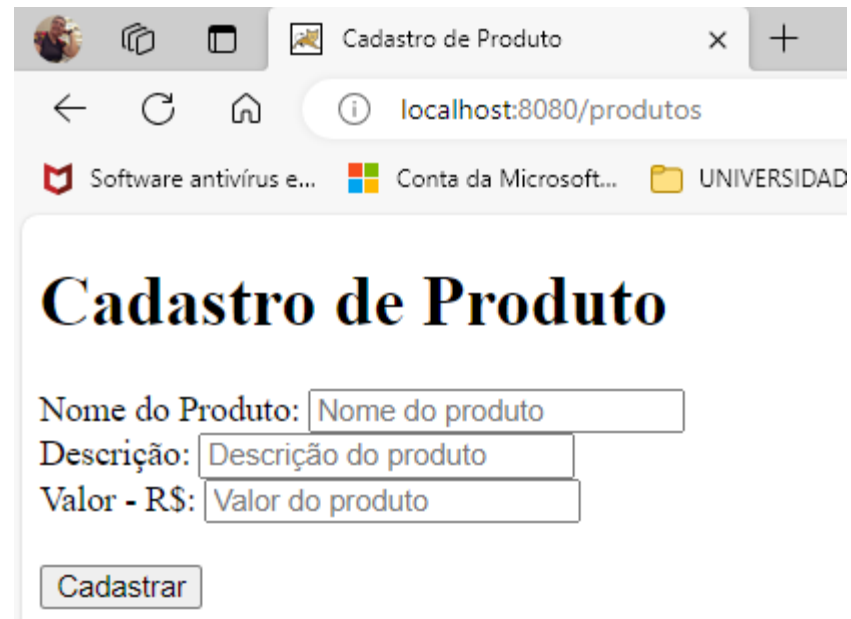


Cadastro de Produto

Nome do Produto:

Descrição:

Valor - R\$:



Cadastro de Produto

Nome do Produto:

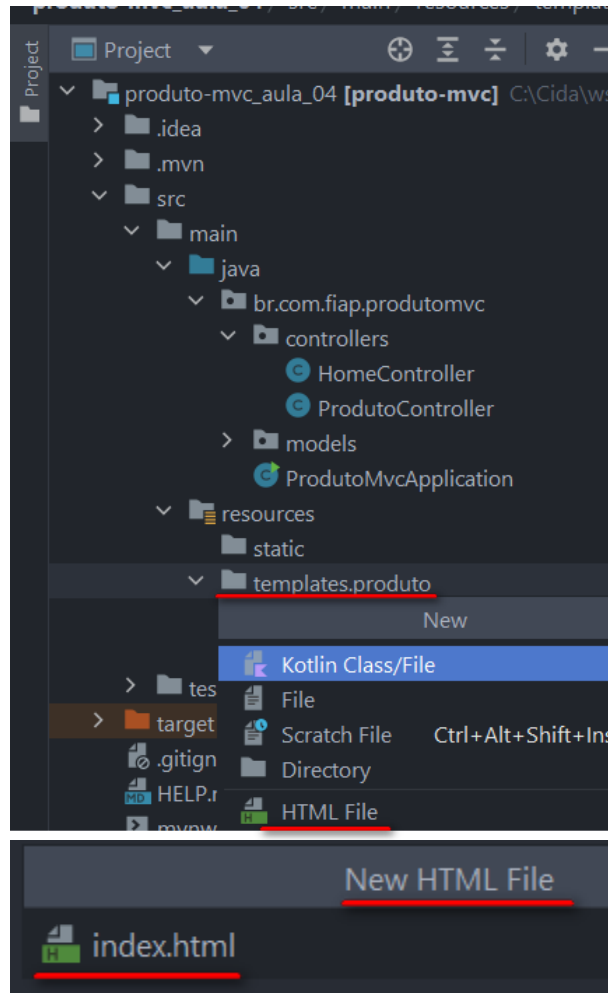
Descrição:

Valor - R\$:

Exercício

Adicionar a Home page do projeto

■ Criar a View index

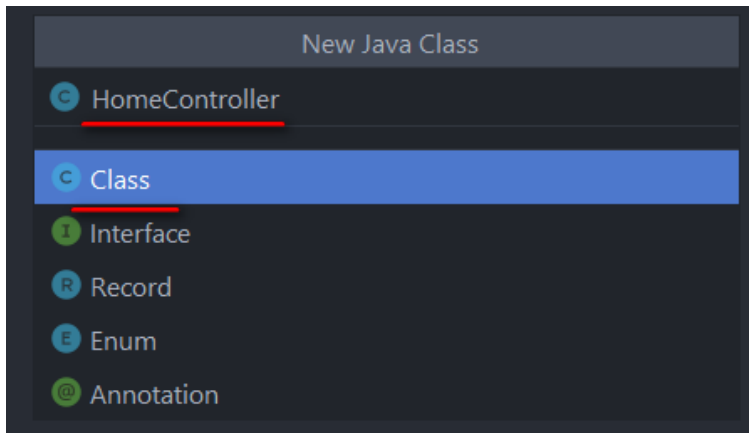
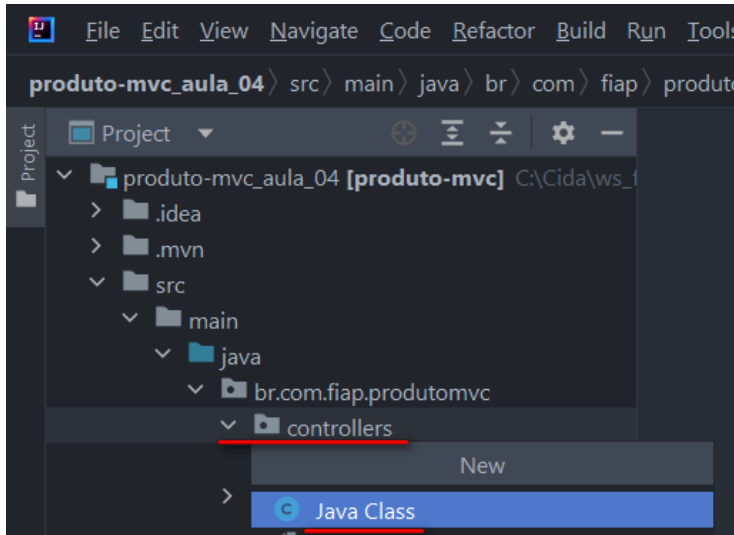


`<html lang="pt-br" xmlns:th="http://thymeleaf.org">`

```
1  <!DOCTYPE html>
2  <html lang="pt-br" xmlns:th="http://thymeleaf.org">
3  <head>
4      <meta charset="UTF-8">
5      <title>Home</title>
6  </head>
7  <body>
8      <div>
9          <h2 th:text="${msg}"></h2>
10         <a href="/produtos"><button>Novo produto</button></a>
11
12     </div>
13 </body>
14 </html>
```

URL - localhost:8080/produtos

Criar HomeController



```
HomeController.java x
1 package br.com.fiap.produtomvc.controllers;
2
3 import org.springframework.stereotype.Controller;
4 import org.springframework.ui.Model;
5 import org.springframework.web.bind.annotation.RequestMapping;
6
7 @Controller
8 public class HomeController {
9
10     @RequestMapping("/") // URL raiz - localhost:8080/
11     public String index(Model model){
12         model.addAttribute("msg", "Bem-vindo(a) à FIAP Developers");
13         return "/produto/index";
14     }
15 }
```

I Model

- No Spring MVC, o **Model** funciona como um contêiner que contém os dados da aplicação. Esses dados podem estar em qualquer formato, como objetos, strings, informações do banco de dados, etc.
- É necessário colocar a interface do **Model** na parte do controlador da aplicação. O objeto `HttpServletRequest` lê as informações fornecidas pelo usuário e as passa para a interface do **Model**. Assim, uma **View** (página de visualização) acessa os dados da parte **model**.
- O **Model** é utilizado para **passar valores** para renderizar **Views**.

Testar aplicação





Copyright © 2024
Prof^a. Aparecida de Fátima Castello Rosa

Copyright © 2020
Prof^o. Pedro Ivo Correia e Prof^o. Lucas Furlaneto

Todos direitos reservados. Reprodução ou divulgação total ou parcial deste documento é expressamente proibido sem o consentimento formal, por escrito, do Professor (autor).