



Spring

DEVinHouse

Parcerias para desenvolver a sua carreira

SENAI

<LAB365>

AGENDA

- Spring
- Maven
- Primeiro Projeto

O Spring Framework fornece um modelo abrangente de programação e configuração para aplicativos empresariais modernos baseados em Java - em qualquer tipo de plataforma de implementação.

Um elemento-chave do Spring é o suporte de infraestrutura no nível do aplicativo: o Spring se concentra no "encanamento" dos aplicativos corporativos para que as equipes possam se concentrar na lógica de negócios no nível do aplicativo, sem vínculos desnecessários com ambientes de implantação específicos.



Spring Boot

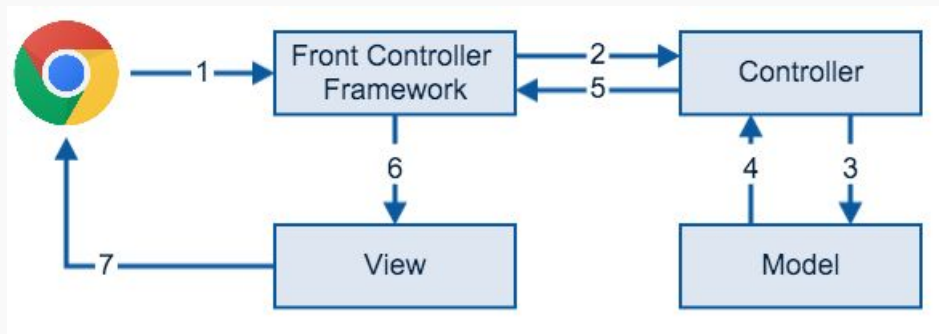
O Spring Boot é um framework Java open source que tem como objetivo facilitar esse processo em aplicações Java. Consequentemente, ele traz mais agilidade para o processo de desenvolvimento, uma vez que devs conseguem reduzir o tempo gasto com as configurações iniciais.

Essencialmente, o Spring Boot pode ser considerado um plugin para a ferramenta de building, seja ela o Maven ou o Gradle. Seus principais objetivos são gerenciar dependências de maneira opinativa e automática, e simplificar a execução do projeto em tempo de desenvolvimento e depuração.



Spring MVC

O **MVC** funciona como um padrão de arquitetura de software que melhora a conexão entre as camadas de dados, lógica de negócio e interação com usuário. Spring MVC é uma estrutura da web baseada em Model View e Controller amplamente usada para desenvolver aplicações web.



O Apache **Maven** é uma ferramenta de automação e gerenciamento de projetos Java, embora também possa ser utilizada com outras linguagens. Ela fornece às equipes de desenvolvimento uma forma padronizada de automação, construção e publicação de suas aplicações, agregando agilidade e qualidade ao produto final.



Spring Initializr

O **Spring Initializr** é um recurso disponibilizado para gerar um projeto de inicialização com os recursos necessários de acordo com o projeto.

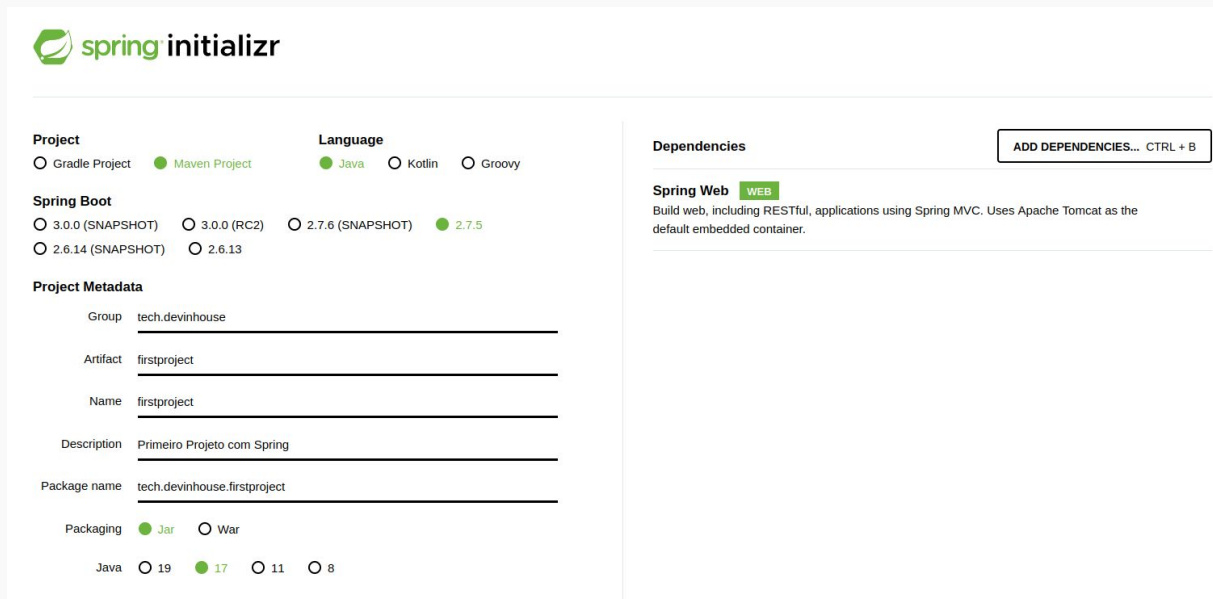
Para isso, basta acessar o endereço:

<https://start.spring.io/>

E configurar de acordo com as dependências necessárias.

Primeiro Projeto

Vamos configurar nosso primeiro projeto baixando um starter pelo Spring Initializr com as seguintes dependências:



The image shows the Spring Initializr web interface. It is divided into several sections for configuring a new project. The 'Project' section has radio buttons for 'Gradle Project', 'Maven Project' (selected), and 'Language' with options for 'Java' (selected), 'Kotlin', and 'Groovy'. The 'Spring Boot' section has radio buttons for versions: '3.0.0 (SNAPSHOT)', '3.0.0 (RC2)', '2.7.6 (SNAPSHOT)', '2.7.5' (selected), and '2.6.14 (SNAPSHOT)', '2.6.13'. The 'Project Metadata' section contains input fields for 'Group' (tech.devinhouse), 'Artifact' (firstproject), 'Name' (firstproject), 'Description' (Primeiro Projeto com Spring), and 'Package name' (tech.devinhouse.firstproject). The 'Packaging' section has radio buttons for 'Jar' (selected) and 'War'. At the bottom, there are version indicators for 'Java' with options 19, 17 (selected), 11, and 8. On the right, the 'Dependencies' section has a button 'ADD DEPENDENCIES... CTRL + B' and a section for 'Spring Web' with a 'WEB' tag and a description: 'Build web, including RESTful, applications using Spring MVC. Uses Apache Tomcat as the default embedded container.'

Project

☐ Gradle Project ☒ Maven Project

Language

☒ Java ☐ Kotlin ☐ Groovy

Spring Boot

☐ 3.0.0 (SNAPSHOT) ☐ 3.0.0 (RC2) ☐ 2.7.6 (SNAPSHOT) ☒ 2.7.5 ☐ 2.6.14 (SNAPSHOT) ☐ 2.6.13

Project Metadata

Group

Artifact

Name

Description

Package name

Packaging

☒ Jar ☐ War

Java ☐ 19 ☒ 17 ☐ 11 ☐ 8

Dependencies

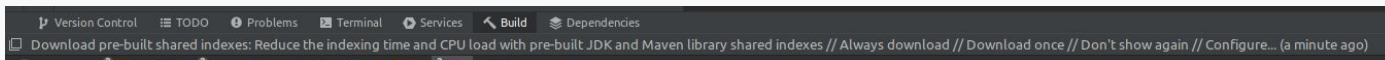
[ADD DEPENDENCIES... CTRL + B](#)

Spring Web WEB

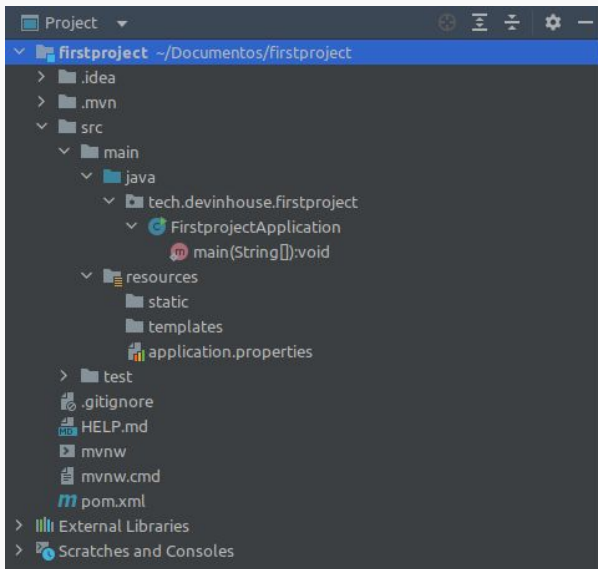
Build web, including RESTful, applications using Spring MVC. Uses Apache Tomcat as the default embedded container.

Primeiro Projeto

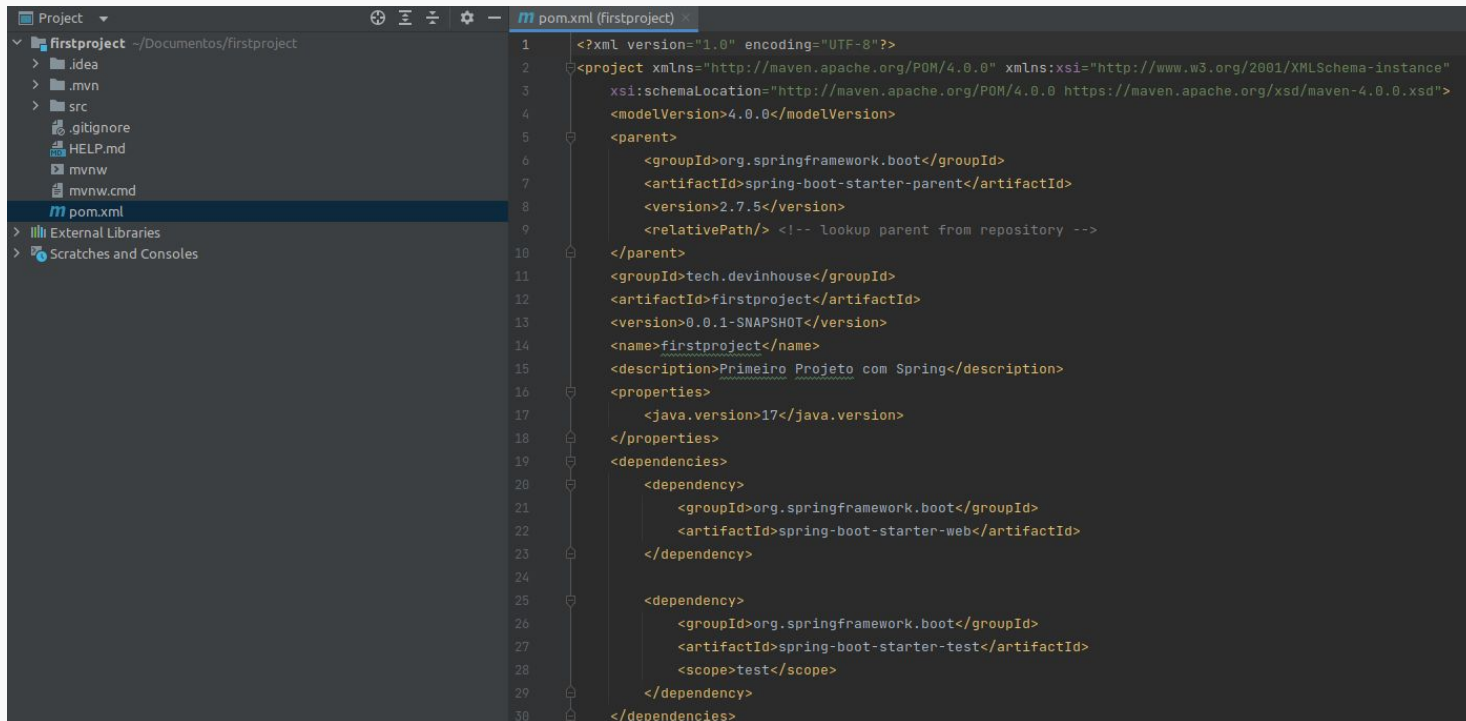
Após o download, abra pelo IntelliJ e aguarde a instalação das dependências do Maven.



Após conclusão, teremos a seguinte estrutura de arquivos:



Primeiramente vamos falar do arquivo pom.xml na estrutura raiz do projeto:



The screenshot shows an IDE with a project named 'firstproject' at the path '~/Documentos/firstproject'. The file explorer on the left lists the project structure, including folders like '.idea', '.mvn', 'src', and files like '.gitignore', 'HELP.md', 'mvnw', and 'mvnw.cmd'. The 'pom.xml' file is selected and its content is displayed in the editor. The XML content defines a Maven project with a parent POM, project details, properties, and dependencies.

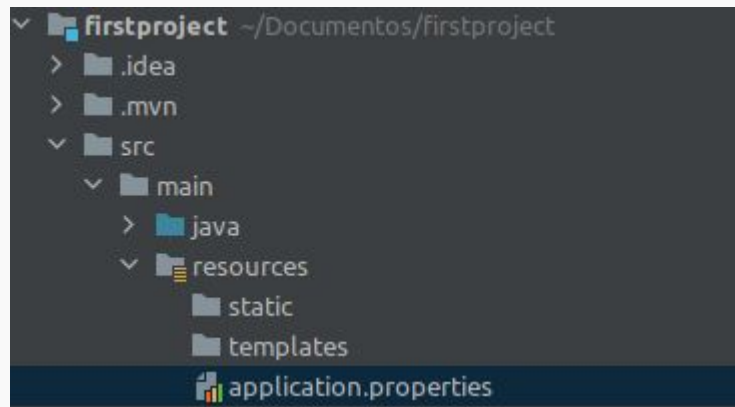
```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
3     xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 https://maven.apache.org/xsd/maven-4.0.0.xsd">
4     <modelVersion>4.0.0</modelVersion>
5     <parent>
6         <groupId>org.springframework.boot</groupId>
7         <artifactId>spring-boot-starter-parent</artifactId>
8         <version>2.7.5</version>
9         <relativePath/> <!-- lookup parent from repository -->
10    </parent>
11    <groupId>tech.devinhouse</groupId>
12    <artifactId>firstproject</artifactId>
13    <version>0.0.1-SNAPSHOT</version>
14    <name>firstproject</name>
15    <description>Primeiro Projeto com Spring</description>
16    <properties>
17        <java.version>17</java.version>
18    </properties>
19    <dependencies>
20        <dependency>
21            <groupId>org.springframework.boot</groupId>
22            <artifactId>spring-boot-starter-web</artifactId>
23        </dependency>
24
25        <dependency>
26            <groupId>org.springframework.boot</groupId>
27            <artifactId>spring-boot-starter-test</artifactId>
28            <scope>test</scope>
29        </dependency>
30    </dependencies>
```

O **pom.xml** é o arquivo que contém as configurações e dependências do projeto pelo Maven. Nele adicionamos as dependências de bibliotecas externas, configurações de estrutura, versionamento e etc.

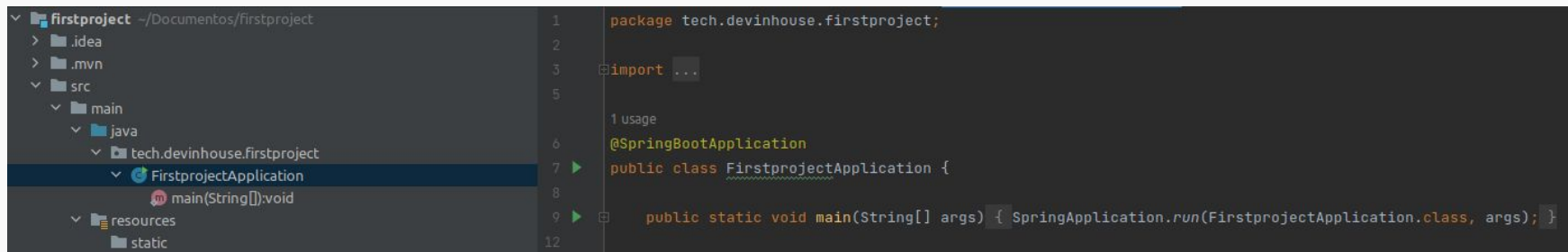
Para mais detalhes, você pode consultar o funcionamento desse arquivo pelo site oficial do Apache Maven em:

<https://maven.apache.org/guides/introduction/introduction-to-the-pom.html>

Outro arquivo importante é o “application.properties” que se encontra em: “*src/main/resources*”. Nele iremos configurar as bibliotecas utilizadas como por exemplo o nosso banco de dados.



Por fim, na pasta java teremos o package do projeto, neste caso, *"tech.devinhouse.firstproject"* que conterá o arquivo de inicialização do Spring Boot.



The screenshot displays an IDE with a project structure on the left and the source code of `FirstprojectApplication.java` on the right. The project structure shows a package `tech.devinhouse.firstproject` under the `src/main/java` directory. The source code on the right is as follows:

```
1 package tech.devinhouse.firstproject;
2
3 import ...
4
5 1 usage
6 @SpringBootApplication
7 public class FirstprojectApplication {
8
9     public static void main(String[] args) { SpringApplication.run(FirstprojectApplication.class, args); }
10
11
12 }
```

Primeiro Controller

Vamos criar uma classe na package "tech.devinhouse.firstproject.controller" chamado "PessoasController" adicionado as anotações:

- @RestController: Define que a classe corresponderá a um controller.
- @RequestMapping("/pessoas"): Define uma rota do projeto que aquele controller apontará.
- @GetMapping(): Declarado em um método que aceitará a requisição via método GET.

```
@RestController
@RequestMapping("/pessoas")
public class PessoaController {

    @GetMapping()
    public String getPets() { return "Primeiro Controller!"; }

}
```

AVALIAÇÃO DOCENTE

O que você está achando das minhas aulas neste conteúdo?

[Clique aqui](#) ou escaneie o QRCode ao lado para avaliar minha aula.

Sinta-se à vontade para fornecer uma avaliação sempre que achar necessário.





DEVinHouse

Parcerias para desenvolver a sua carreira

OBRIGADO!



<LAB365>