

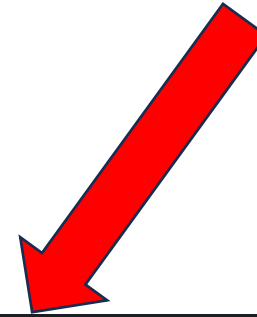
# POO2

Encontro 05 – Criando Projetos Spring Boot

Procedimentos de criação

Codificação Básica

<https://spring.io/> ←

A screenshot of the Spring website homepage. The header features the Spring logo and navigation links: 'Why Spring', 'Learn', 'Projects', 'Academy', 'Solutions', and 'Community'. The 'Projects' dropdown menu is open, showing a list of project categories. The main content area has a large heading 'Spring makes Java simple.' and two buttons: 'WHY SPRING' and 'QUICKSTART'. At the bottom, there is a 'NEWS' button and a link to the 'State of Spring Survey'.

spring<sup>®</sup> by VMware Tanzu

Why Spring ▾ Learn ▾ Projects ▾ Academy ▾ Solutions ▾ Community ▾

Spring makes Java simple.

WHY SPRING QUICKSTART

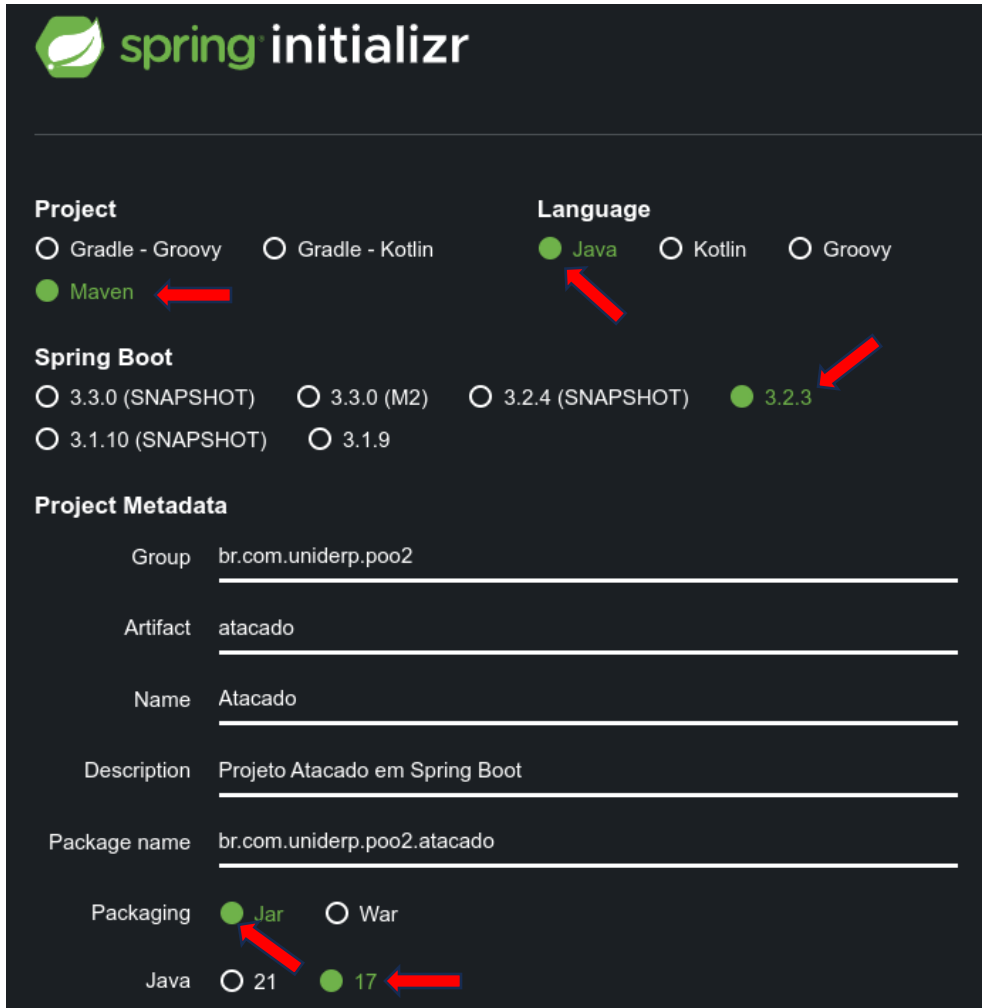
Overview  
Spring Boot  
Spring Framework  
Spring Cloud  
Spring Cloud Data Flow  
Spring Data  
Spring Integration  
Spring Batch  
Spring Security  
View all projects

DEVELOPMENT TOOLS  
Spring Tools 4  
Spring Initializr ↗

NEWS | State of Spring Survey



# <https://start.spring.io/> (parte 1)



**spring initializr**

**Project**  
☐ Gradle - Groovy ☐ Gradle - Kotlin ☒ **Maven**

**Language**  
☒ **Java** ☐ Kotlin ☐ Groovy

**Spring Boot**  
☐ 3.3.0 (SNAPSHOT) ☐ 3.3.0 (M2) ☐ 3.2.4 (SNAPSHOT) ☒ **3.2.3**  
☐ 3.1.10 (SNAPSHOT) ☐ 3.1.9

**Project Metadata**

Group

Artifact

Name

Description

Package name

Packaging ☒ **Jar** ☐ War

Java ☐ 21 ☒ **17**

Project: Maven  
Language: Java  
Spring Boot: 3.2.3

Group: br.com.uniderp.poo2  
Artifact: atacado  
Name: Atacado  
Description: Projeto Atacado em Spring Boot.  
Package name: br.com.uniderp.poo2.atacado  
Packaging: Jar  
Java: 17

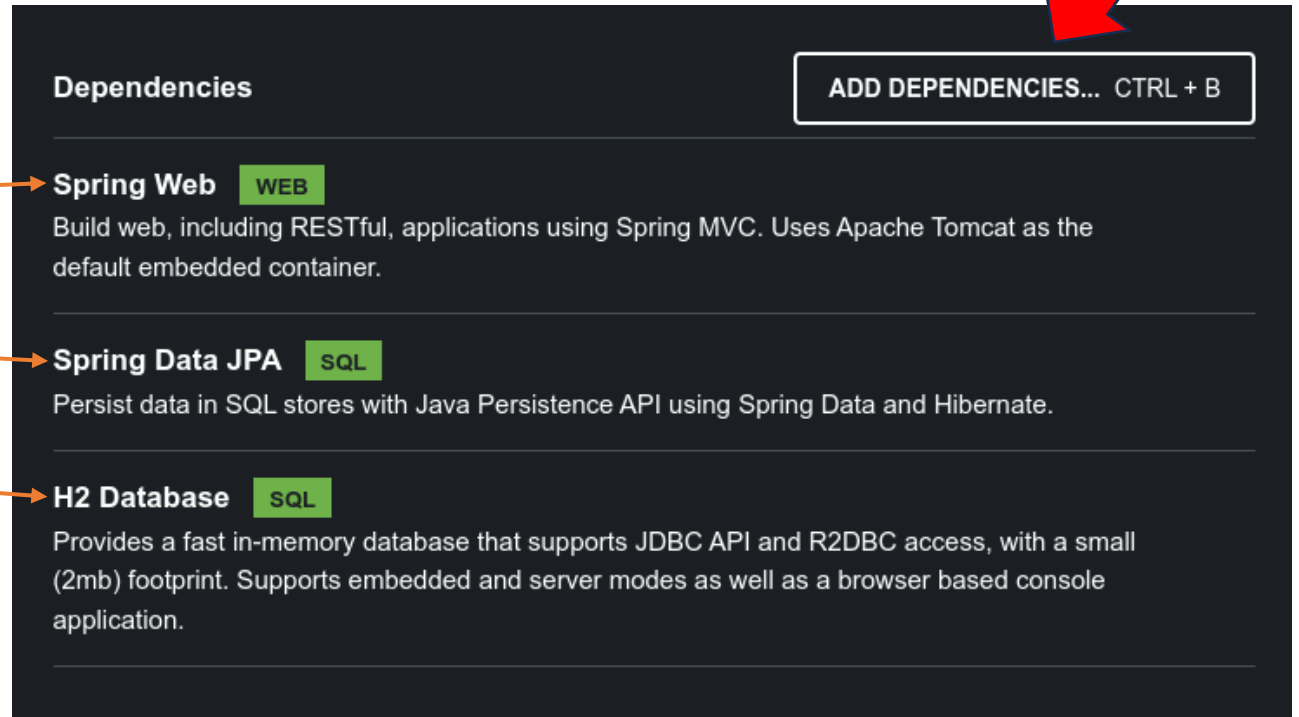
**ATENÇÃO: ainda não terminamos, precisamos adicionar as dependências.**

<https://start.spring.io/> (parte 1)

Spring Web

Spring Data JPA

H2 Database



The screenshot shows the 'Dependencies' panel of the Spring Start IDE. A red arrow points to the 'ADD DEPENDENCIES... CTRL + B' button in the top right corner. Three orange arrows point from the text labels on the left to the dependency entries: 'Spring Web' points to the first entry, 'Spring Data JPA' points to the second, and 'H2 Database' points to the third. The dependency entries are listed with their category (WEB or SQL) in a green box and a description.

**Dependencies** ADD DEPENDENCIES... CTRL + B

- Spring Web** WEB  
Build web, including RESTful, applications using Spring MVC. Uses Apache Tomcat as the default embedded container.
- Spring Data JPA** SQL  
Persist data in SQL stores with Java Persistence API using Spring Data and Hibernate.
- H2 Database** SQL  
Provides a fast in-memory database that supports JDBC API and R2DBC access, with a small (2mb) footprint. Supports embedded and server modes as well as a browser based console application.

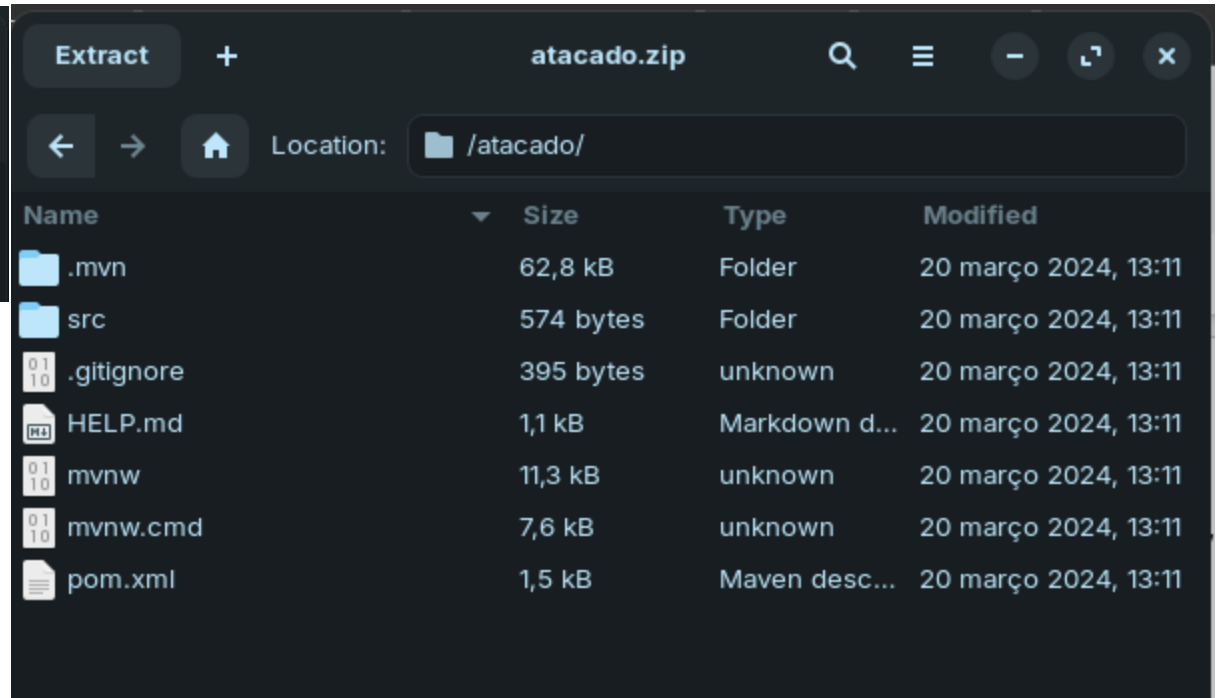
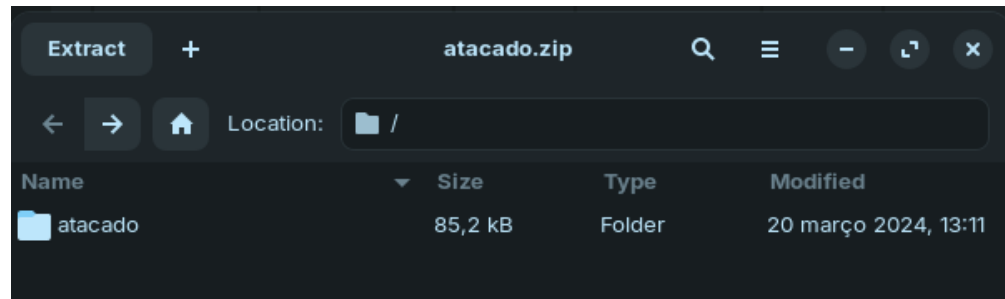
GENERATE CTRL + ↵

EXPLORE CTRL + SPACE

SHARE...

# Arquivo atacado.zip

Após a geração, o arquivo será baixado automaticamente. Este arquivo contém o projeto Spring Boot, e tudo o que é necessário para iniciar a codificação do projeto.



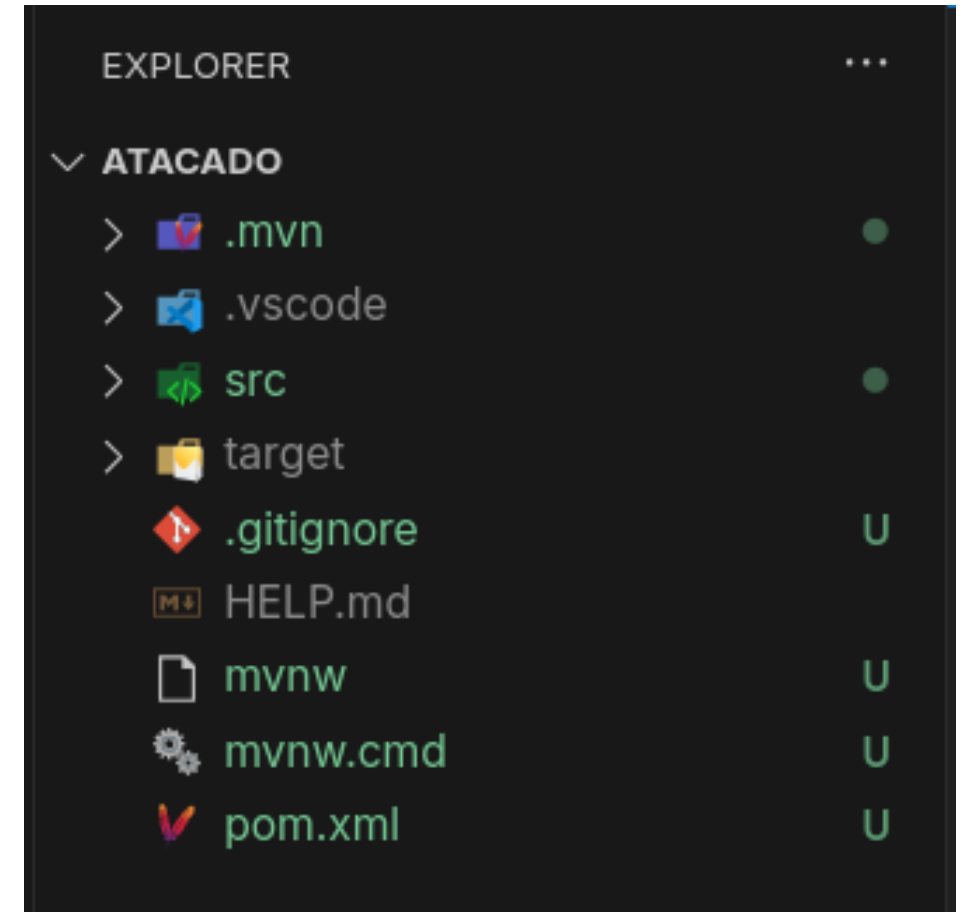
# VS Code

Lista de Extensões do VS Code:

- Extension Pack for Java (pacote com 6 extensões);
- Spring Boot Extension Pack (pacote com 3 extensões);
- Vscode-icons (extensão para visualização de ícones nas pastas e arquivos importantes);
- Live Server (Servidor web para testes);
- Prettier Code Formatter (formatador de código)

# VS Code

- Copie o arquivo atacado.zip para uma pasta mais adequada, de fácil acesso, e descompacte o arquivo.
- Abra o VS Code (configurado com as extensões), e abra a pasta atacado.
- Ao lado temos uma visão das pastas e arquivos do projeto.



# VS Code

## **ATENÇÃO:**

- O JDK deve estar instalado previamente, antes de abrir o projeto.
- Também deve estar configurado (JAVA\_HOME e PATH).
- A sugestão é utilizar o JDK 17 ou superior.



# VS Code – Projeto Atacado

Após abrir o projeto, o VS Code pode demorar um pouco, pois o Maven vai restaurar todos os arquivos e pacotes necessários, fazendo download dos recursos.

A partir daqui, vamos criar os arquivos necessários para executar a API REST.

# VS Code – Projeto Atacado

- Selecione a pasta Java, e abra até a pasta atacado.
- Na pasta atacado, vamos criar uma pasta denominada Entities.
- Na pasta Entities, vamos criar as classes:
- Classe
- Subclasse
- Produto

A seguir, temos o diagrama de classe.

# VS Code – Projeto Atacado

Classe
- codigo: Long - nome: String - dataDeInclusao: Date
+ Classe() + Classe(Long, String, Date)

Subclasse
- codigo: Long - codigoClasse: Long - nome: String - dataDeInclusao: Date
+ Subclasse() + Subclasse(Long, Long, String, Date)

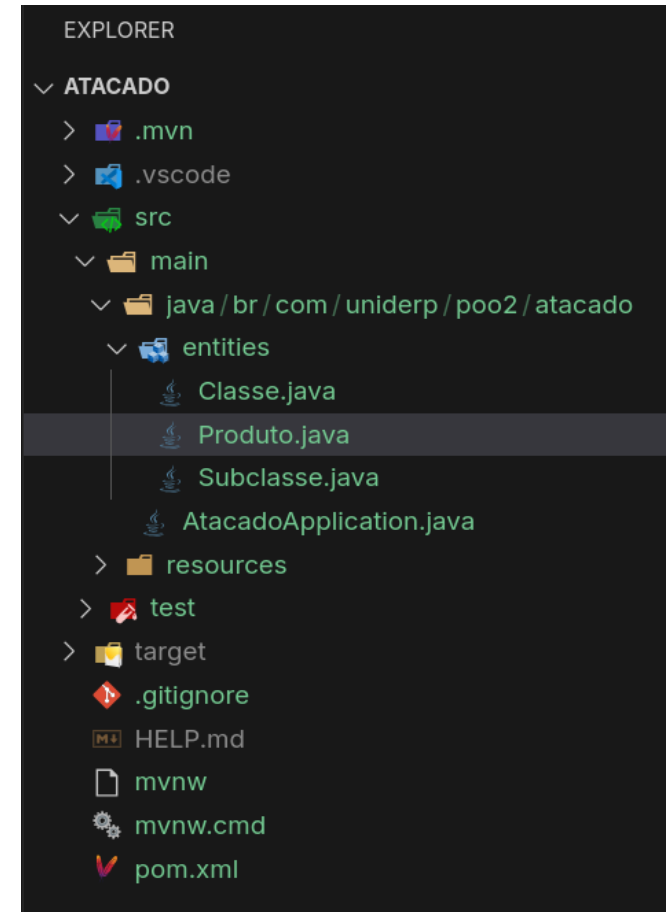
Produto
- codigo: Long - codigoSubclasse: Long - nome: String - preco: Double - dataDeInclusao: Date
+ Produto() + Produto(Long, Long, String, Double, Date)

# VS Code – Projeto Atacado

Após a codificação, ficaremos com algo semelhante a imagem ao lado.

Compare e perceba a organização dos arquivos no projeto.

Vamos prosseguir para a camada de Controller.



# VS Code – Projeto Atacado

- Selecione a pasta Java, e abra até a pasta atacado.
- Na pasta atacado, vamos criar uma pasta denominada Controllers.
- Na pasta Controllers, vamos criar as classes:
- ClasseController
- SubclasseController
- ProdutoController

Nesta aula, vamos trabalhar apenas com a classe ClasseController, para servir de exemplo.

# VS Code – Projeto Atacado

- Selecione a pasta Java, e abra até a pasta atacado.
- Na pasta atacado, vamos criar uma pasta denominada Controllers.
- Na pasta Controllers, vamos criar as classes:
- ClasseController
- SubclasseController
- ProdutoController

Nesta aula, vamos trabalhar apenas com a classe ClasseController, para servir de exemplo. Siga as orientações do professor.