

Oficina Spring Boot

Alunos IFSP Votuporanga

Outubro 2024

1 O que é Spring Boot?

Spring Boot é um *ecossistema* open-source baseado em Java, que facilita o desenvolvimento de aplicativos Java com o framework Spring. Ele foi criado para simplificar o processo de configuração e execução de aplicações Spring, permitindo que os desenvolvedores construam rapidamente aplicações robustas e produtivas.

1.1 Quais são as vantagens de usar um framework?

- **Produtividade aumentada:** Frameworks fornecem componentes e ferramentas prontos para uso, permitindo que os desenvolvedores se concentrem nas funcionalidades da aplicação ao invés de lidar com configurações de baixo nível.
- **Manutenção facilitada:** A estrutura organizada e padronizada de um framework ajuda a manter o código mais legível e fácil de entender por diferentes desenvolvedores, além de simplificar futuras atualizações e correções.
- **Segurança:** Muitos frameworks incluem funcionalidades de segurança pré-embutidas, como proteção contra injeção de SQL, XSS (Cross-Site Scripting), entre outros. Isso ajuda a proteger as aplicações sem que o desenvolvedor precise implementar todas as medidas manualmente.
- **Testabilidade:** A maioria dos frameworks promove boas práticas para o desenvolvimento de código testável, além de fornecer integração com bibliotecas de teste, facilitando a criação de testes automatizados e garantindo a qualidade do software.
- **Comunidade e suporte:** Frameworks populares possuem grandes comunidades, com uma ampla documentação, tutoriais e fóruns de suporte, o que facilita a resolução de problemas e o aprendizado de novas funcionalidades.
- **Escalabilidade:** Frameworks são desenvolvidos para suportar o crescimento das aplicações, oferecendo soluções estruturadas para gerenciar grandes volumes de dados, requisições e serviços distribuídos.
- **Reutilização de código:** Frameworks promovem a reutilização de componentes e padrões já testados e validados, reduzindo a duplicação de código e minimizando erros.
- **Padronização:** O uso de um framework incentiva o desenvolvimento dentro de padrões estabelecidos, resultando em código mais consistente e de fácil integração entre equipes de desenvolvimento.

- **Integração simplificada:** Frameworks costumam oferecer fácil integração com outras tecnologias e bibliotecas, facilitando a construção de sistemas complexos e interconectados.
- **Redução do tempo de desenvolvimento:** Através de ferramentas e funcionalidades pré-definidas, o uso de um framework pode acelerar consideravelmente o ciclo de desenvolvimento, diminuindo o tempo necessário para lançar uma aplicação.

2 Como começar?

A melhor forma de aprender algo é através da documentação oficial:


- [Visão Geral](#)
- [Guia Inicial](#)
- [Todos os Guias](#)

3 Como inicializar nosso projeto?

1. Use o [Gerador Oficial](#) para criar um projeto.
2. Selecione os seguintes parâmetros:
 - **Project:** Maven
 - **Language:** Java
 - **Spring Boot:** 3.3.4 (**SEM** snapshot)
3. Insira os seguintes metadados:
 - **Group:** br.com
 - **Artifact:** cadastroalunos
 - **Name:** cadastroalunos
 - **Description:** Uma API Rest de cadastro de alunos.
 - **Package name:** br.com.cadastroalunos
 - **Packaging:** Jar
 - **Java:** Java 17 (A depender da versão instalada nos computadores)
4. Clique em **ADD DEPENDENCIES** (Ctrl + B) e adicione as seguintes dependências:
 - **Spring Web**
 - **PostgreSQL Driver**
 - **Spring Data JPA**
 - **Spring Boot DevTools**
 - **Lombok**

Ao final desse processo você deve ter um formulário semelhante ao representado na [Figura 1](#).

5. Clique em **GENERATE** (Ctrl + Enter) e o projeto deve ser baixado no formato de um arquivo compactado. Salve-o em um local de fácil acesso (De preferência em um diretório que não necessite de privilégios de administrador).

 **spring initializr**

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

Spring Boot
☐ 3.4.0 (SNAPSHOT) ☐ 3.4.0 (M3) ☐ 3.3.5 (SNAPSHOT) ☒ **3.3.4**
☐ 3.2.11 (SNAPSHOT) ☐ 3.2.10

Project Metadata
Group
Artifact
Name
Description
Package name
Packaging ☒ **Jar** ☐ War
Java ☐ 23 ☐ 21 ☒ **17**

Dependencies ADD DEPENDENCIES... CTRL + B

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

PostgreSQL Driver **SQL**
A JDBC and R2DBC driver that allows Java programs to connect to a PostgreSQL database using standard, database independent Java code.

Spring Data JPA **SQL**
Persist data in SQL stores with Java Persistence API using Spring Data and Hibernate.

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

Lombok **DEVELOPER TOOLS**
Java annotation library which helps to reduce boilerplate code.

GENERATE CTRL + G

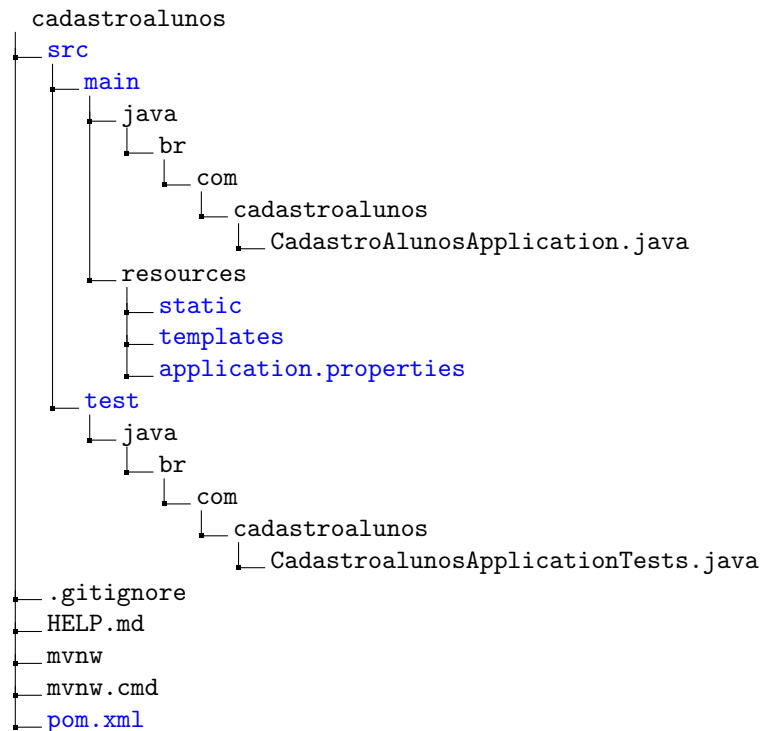
EXPLORE CTRL + SPACE

SHARE...

Figure 1: Exemplo de bootstrapping da nossa aplicação

4 Estrutura de um Projeto com Spring Boot

Ao abrir o projeto que você acabou de gerar no Visual Studio Code, você deve se deparar com uma estrutura como a seguinte:



4.1 Source

Diretório principal da aplicação. É onde todo o código fonte deve ser inserido, diretório pai dos diretórios Main e Test.

4.2 Main

Diretório onde deve residir a lógica da aplicação, é dentro dele que serão colocadas classes Controllers, Entidades e Repositórios.

4.3 Static

Onde devem ser colocados os arquivos estáticos, por exemplo Imagens, páginas HTML e folhas de estilo CSS.

4.4 Templates

Onde devem ser colocados os templates para renderização das páginas WEB. Alguns exemplos de engines de template suportadas pelo Spring são: Thymeleaf, Mustache e JSP's (Java Server Pages)

4.5 Application.Properties

Esse é o arquivo onde a maior parte da configuração da nossa aplicação será feita, nele podemos indicar os detalhes da conexão com o banco de dados. Comportamento das bibliotecas instaladas e chaves de encriptação para senhas.

4.6 Test

Diretório onde serão escritos os testes automatizados da aplicação para garantir a funcionalidade e qualidade do código escrito pelos desenvolvedores que colaboram no projeto.

4.7 Pom.xml

Arquivo onde serão declarados os metadados do projeto, os nomes e as versões de pacotes baixados pelo Maven.

5 Como rodar um projeto Spring Boot?

Existem diversas formas de executar um projeto Spring. A depender da IDE utilizada é provável que um simples click no botão com ícone de play já seja suficiente. De qualquer forma, vocês aprenderão a rodar um projeto utilizando a ferramenta de build Maven.

1. Abra um novo terminal do VSCode (Caso prefira abrir um terminal do próprio S.O. será necessário estar no diretório que contém o arquivo "pom.xml" para que o processo funcione)
2. Digite o seguinte comando:

```
1 $ mvn spring-boot:run
```

6 Principais Anotations do Spring Boot

6.1 @Service

A anotação `@Service` é usada para marcar uma classe Java como um *Service*, ou seja, uma camada de serviço na arquitetura da aplicação. Essa anotação indica que a classe contém a lógica de negócio da aplicação.

6.2 @Repository

A anotação `@Repository` é usada para marcar uma classe Java como um *Repository*, ou seja, uma camada de acesso a dados. Essa anotação indica que a classe contém a lógica de interação com o banco de dados.

6.3 @Entity

A anotação `@Entity` é usada para marcar uma classe Java como uma entidade, ou seja, uma representação de uma tabela no banco de dados. Essa anotação indica que a classe será mapeada para uma tabela no banco de dados.

6.4 @Table

A anotação `@Table` é usada em conjunto com a anotação `@Entity` para especificar o nome da tabela no banco de dados que a entidade será mapeada.

6.5 @Autowired

A anotação `@Autowired` é usada para injetar dependências em uma classe Java. Essa anotação indica que o Spring deve criar uma instância do objeto e injetá-lo na classe que o está usando.