



Spring Boot



**Certified
Developer**
The Ultimate Tech Degree

DigitalHouse >
Coding School



Temas

1

**Spring Boot
Initializr**

2

@Annotations

3

**Definindo a
classe principal**



1 | Spring Boot Initializr



Spring Boot Initializr

Spring Initializr é um pequeno utilitário web que nos permite criar um esqueleto de um projeto Spring Boot com as opções que desejamos configurar.

Permite escolher o provedor de dependências (Maven, Gradle, etc.), a linguagem a utilizar (Java, Kotlin, etc.), o tipo de empacotamento (Jar ou War), as dependências/bibliotecas de que precisamos, entre outras configurações iniciais.



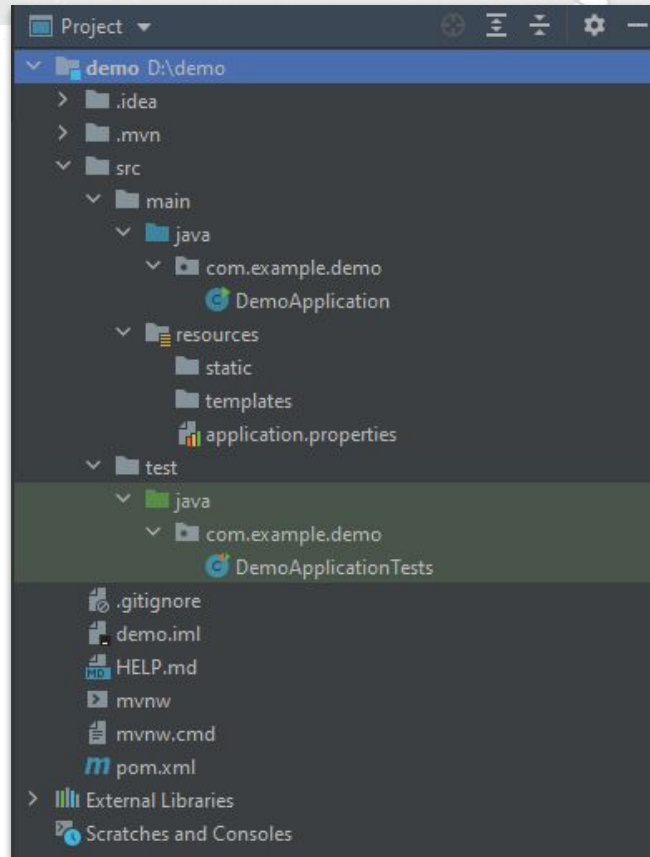
Pode ser usado a partir do [site oficial](#), ou baixando um plugin para IntelliJ IDEA ou a IDE que estamos usando.



Estrutura do Projeto

Os seguintes arquivos e diretórios são criados. Entre os mais importantes estão:

- pom.xml
- application.properties.
- pasta de destino (após construir o projeto).
- profile (definir ambiente de trabalho).



2 | @Annotations



Por que estamos interessados em @Annotations?

Com o Spring Boot, é possível executar uma aplicação Spring com muito pouca configuração, a qual está na forma de [anotações](#).

Uma anotação é usada para fornecer ao compilador ou às ferramentas de desenvolvimento informações para fazer algo.

Uma anotação sempre vem antes de uma classe, método ou declaração e tem a seguinte aparência:

@EstaEUmaAnotacao





Por que estamos interessados em @Annotations?

Pode ter elementos como parâmetros com valor para fornecer mais informações, por exemplo:

```
@Autor(nome = "DH", data = "14/03/2019")  
public class MinhaClasseExemplo() {  
    ...  
}
```

Podemos colocar múltiplas anotações:

```
@Autor(nome = "DH", data = "14/03/2019")  
@Deprecated  
@OutraAnotacao("valor")  
public class MinhaClasseExemplo() {  
    ...  
}
```




Quais são as mais comuns?

@Deprecated

Nós a usamos para indicar que um método está obsoleto. Imagine que existem vários programadores trabalhando no mesmo projeto e temos um método que será eliminado no futuro, então você simplesmente adiciona a anotação @Deprecated e pronto.

@Override

Informa ao compilador que substituirá o método da classe pai. Isso é aplicável quando há herança de classe. Se sua classe tem um **extends**, você pode sobrescrever seu código, colocando em seu método que está na classe filha com esta anotação.

@SuppressWarnings

Diz ao compilador para remover as advertências que podem ser exibidas, ou seja, não mostrará mais os avisos que este método pode gerar. Será útil quando você não estiver mais interessado em alguns avisos que conhece.

3

Definindo a classe principal



SpringApplication

Como já sabemos, toda aplicação Java deve conter uma classe principal com um método **main**.

Este método, no caso de implementação de uma aplicação Spring, deve chamar o método **run** da classe **SpringApplication**.

A seguir, definiremos de forma muito fácil a classe principal de nossa aplicação e vamos ver como o Spring Boot nos ajuda a simplificar as **annotations** que tínhamos com o Spring Framework.





@Configuration indica que a classe em que está contém a configuração principal do projeto.

A anotação **@EnableAutoConfiguration** indica que será aplicada a configuração automática do starter que usamos. Só deve ser adicionada em um local, sendo muito comum colocá-la na classe main.

Terceiro, a etiqueta **@ComponentScan** ajuda a localizar elementos marcados com outras anotações quando necessário.

```
@Configuration
@EnableAutoConfiguration
@ComponentScan
public class Application {

    public static void main(String[] args) throws Exception {
        SpringApplication.run(Application.class, args);
    }
}
```



Para não preencher nossa classe com anotações, ao contrário de usar apenas Spring Framework, o **Spring Boot** nos permite substituir as etiquetas `@Configuration`, `@EnableAutoConfiguration` e `@ComponentScan` por **@SpringBootApplication**, que abrange o resto.

```
@SpringBootApplication = {  
  
    @Configuration  
    @EnableAutoConfiguration  
    @ComponentScan  
  
}
```



DigitalHouse>