

# Spring framework com Spring Boot

Framework open source desenvolvido para a plataforma Java baseado nos padrões de projetos inversão de controle e injeção de dependência.

Sua estrutura é composta por módulos afins de reduzir a complexidade no desenvolvimento aplicações simples ou corporativa.

## Spring Módulos

É baseado no módulo core.

Possui o módulo de gerenciamento e armazenamento de dados.

Possui o módulo Web.

## Spring versus Java EE

O Java EE era muito complicado com procedimentos burocráticos, apenas para chegar a alguns resultados. Então usar o Spring era um caminho mais simples e fácil de evoluir. Porém agora existe a versão 5 do Java EE e a comunidade volta na discussão Spring versus Java EE. Portanto, atualmente, tanto o Java EE quanto o Spring são úteis para o desenvolvimento – apenas vai depender da empresa qual utilizar.

## Inversão de Controle

O propósito é minimizar o acoplamento do código. Inversion of Control ou IoC, trata-se do redirecionamento do fluxo de execução de um código retirando parcialmente o controle sobre ele e delegando-o para um container.

Com o IoC, o Container cria todos os objetos que é preciso criar.

## Injeção de Dependências

É um padrão de desenvolvimento com a finalidade de manter baixo o nível de acoplamento entre módulos de um sistema.

A aplicação, ao ser inicializada, carrega o Container, que tem objetos reais, que serão injetadas na Interface.

## Beans

Bean é um objeto que é instanciado (criado), montado e gerenciado por um container através do princípio da inversão de controle.

## **Scopes**

Há escopos com tarefas diferentes. Alguns estão entre os contextos HTTP (escopos de requisições, de sessão e global).

### *Singleton*

Container define apenas uma instancia do objeto.

### *Prototype*

Será criado um novo objeto a cada solicitação ao container.

### *HTTP*

Request: Um bean será criado para cada requisição HTTP.

Session: Um bean será criado para a sessão de usuário.

Global: Ou Application Scope cria um bean para o ciclo de vida do contexto da aplicação.

## **Autowired**

Anotação (indicação) onde deverá ocorrer uma injeção automática de dependência.

## **Spring framework – projeto Springboot**

Enquanto o Spring Framework é baseado no padrão de injeção de dependência, o Springboot foca na configuração automática, na produtividade.

Inicia o projeto com todas as configurações necessários para o início de um projeto já definidas.

Starters são descritores de dependência. Seus benefícios são: coesão, versões compatíveis, otimização do tempo, configuração simples e foco no negócio.

Alguns starters mais utilizados são: data-jpa (integração ao banco de dados via jpa), data-mongodb, web, web-services, batch, test, openfeign, actuator.

## ***Primeiros passos***

Foi criado e inicializado uma aplicação Spring Boot por meio do site Spring Initializr.

No projeto, não pode existir comandos depois do SpringApplication.run() na classe Application. É preciso criar outra classe como componente e implementar CommandLineRunner.

Através do `@Autowired` é realizado a injeção de dependências. Por exemplo, um objeto que é do tipo de outra classe – Calculadora calculadora. (Isso com o `@Autowired` na linha acima)

### ***Beans X Component***

É utilizado o Bean quando não tem acesso ao código fonte.

É utilizado o Component quando é uma classe que tem a finalidade de ser provida através do conceito de injeção de dependências.

### **ORM**

Mapeamento de objeto relacional. É um recurso para aproximar o paradigma da orientação a objetos ao contexto de banco de dados relacional.

Seu uso é realizado através do mapeamento de objeto para uma tabela por uma biblioteca ou framework.

Há uma camada de implementação.

### **JPA**

É uma especificação baseada em interfaces, que através de um framework realiza operações de persistência de objetos em Java.

Além do Hibernate, existem outros movimentos, como: eclipse link, top link (da oracle), etc.

*Mapeamento:* Aspectos das anotações de mapeamento do JPA: Identificação, definição, relacionamento, herança e persistência.

### **Repository**

É um padrão de projeto similar ao DAO (Data Access Object) no sentido de que seu objetivo é abstrair o acesso a dados de forma genérica a partir de seu modelo.

Spring Data JPA facilita a implementação do padrão Repository através de AOP (Aspect Oriented Programming – programação orientada a aspectos). Ele mesmo se encarrega de interpretar a assinatura de um método para montar a JPQL (SQL pelo JPA) correspondente.