

# Spring Boot + Injeção de Dependência

Mergulhe no mundo do Spring Boot e aprenda como a injeção de dependência simplifica a construção de aplicações robustas e escaláveis. Explore os conceitos, técnicas e boas práticas para dominar essa poderosa ferramenta.



by João Choma

# Introdução ao Spring Boot

Spring Boot é um framework para desenvolver aplicações Java com facilidade e velocidade. Simplifica a configuração e oferece convenções para iniciar projetos rapidamente.

1

## Início Rápido

O Spring Boot fornece uma estrutura pronta para uso, acelerando o desenvolvimento de aplicações.

2

## Convenções

Convenções simplificam a configuração, reduzindo o tempo gasto com configurações complexas.

3

## Autoconfiguração

O Spring Boot configura automaticamente as dependências necessárias para sua aplicação.



# Princípios da Injeção de Dependência

A injeção de dependência (DI) é um padrão de design que promove o desacoplamento de classes. Em vez de criar instâncias de classes diretamente, elas são injetadas.

## Desacoplamento

As classes não dependem diretamente da implementação de suas dependências, tornando o código mais flexível e testável.

## Reutilização

Componentes podem ser facilmente reutilizados em diferentes partes da aplicação, sem a necessidade de modificações.

## Testabilidade

A injeção de dependências facilita o teste unitário de componentes, permitindo a substituição de dependências reais por mocks.

# Configuração de Dependências no Spring Boot

No Spring Boot, a configuração de dependências é realizada através do arquivo pom.xml, usando o Maven, ou build.gradle, usando o Gradle.

Dependências	Exemplo
Spring Web	org.springframework.boot:spring-boot-starter-web
Spring Data JPA	org.springframework.boot:spring-boot-starter-data-jpa



A diagram illustrating the Spring bean lifecycle. It features three rectangular boxes: 'Spring bean' at the top, 'bntalln' in the middle, and 'Desscton' at the bottom. A solid white arrow points from 'Spring bean' to 'bntalln', and another solid white arrow points from 'bntalln' to 'Desscton'. A dashed white arrow points from 'Desscton' back to 'Spring bean', completing a cycle.

Spring bean

bntalln

Desscton

# Gerenciamento de Ciclo de Vida dos Beans

O Spring gerencia o ciclo de vida dos beans, que são as classes que compõem sua aplicação. O ciclo de vida inclui etapas como criação, inicialização e destruição.

1

Construção

O bean é instanciado e seus atributos são inicializados.

2

Inicialização

O bean é configurado e preparado para uso.

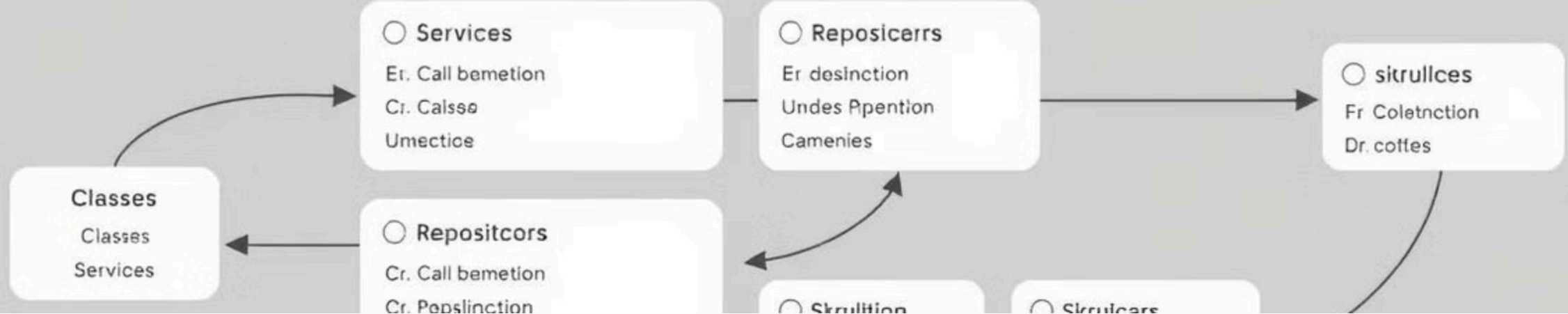
3

Destruição

O bean é removido da aplicação quando não é mais necessário.







# Exemplos Práticos de Injeção de Dependência

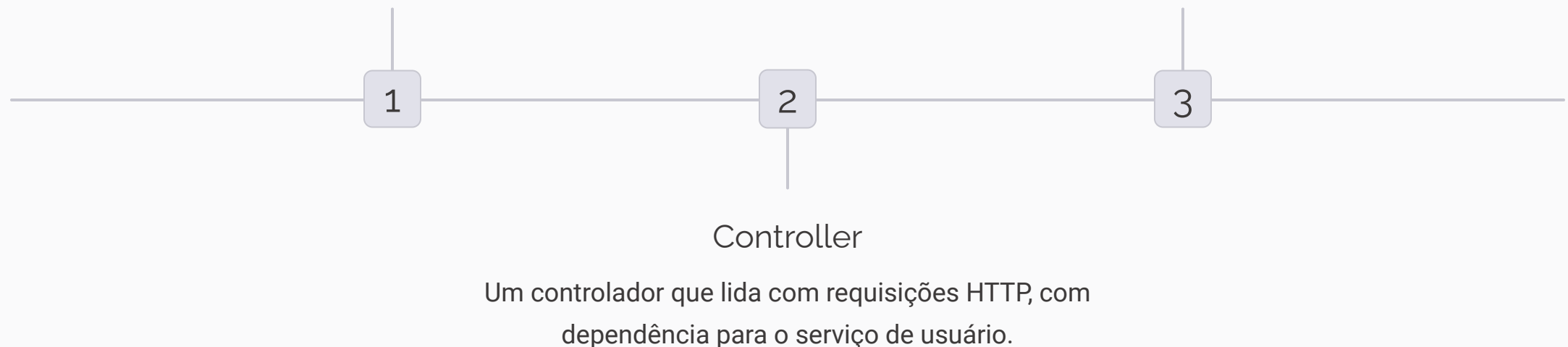
A injeção de dependência é amplamente utilizada em aplicações Spring Boot. Veja exemplos práticos de como essa técnica é aplicada em diferentes cenários.

## Serviço de Usuário

Um serviço que gerencia dados de usuários, com dependências para repositório e validação.

## Repositório de Usuário

Um repositório que interage com o banco de dados para operações de persistência de dados.



# Boas Práticas e Dicas de Implementação

Siga boas práticas para otimizar a implementação da injeção de dependência em aplicações Spring Boot. Mantenha o código organizado, testável e escalável.

## Testes Unitários

Escreva testes unitários para validar o comportamento de seus componentes.

## Injeção por Construtor

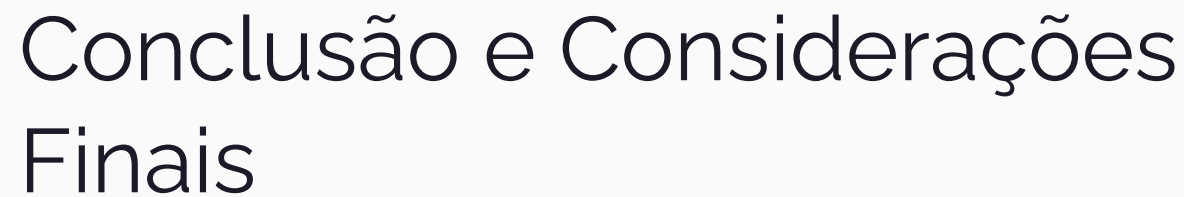
Injete dependências por meio do construtor da classe, garantindo que as dependências sejam obrigatórias.

## Interfaces

Utilize interfaces para desacoplar as dependências, facilitando a substituição de implementações.







Compreender e aplicar os princípios da injeção de dependência leva à construção de aplicações Spring Boot robustas, fáceis de manter e escaláveis.