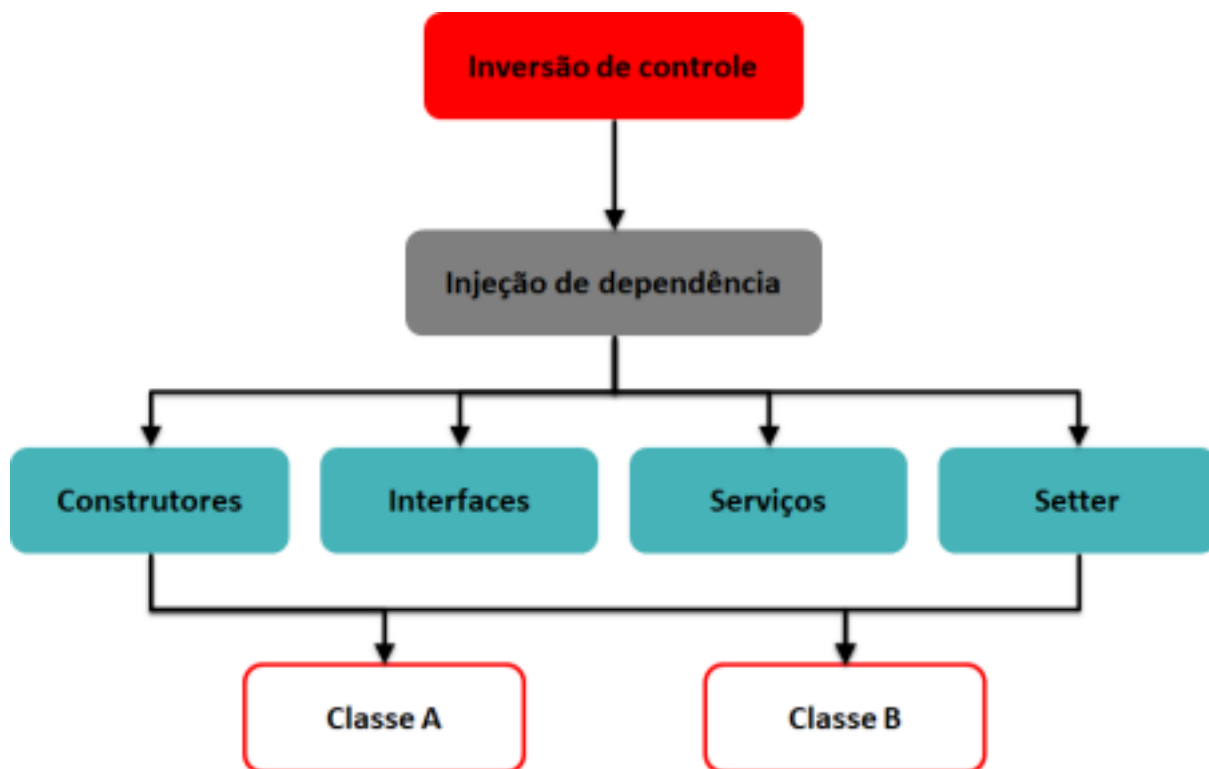


Inversão de controle

Os dois conceitos IoC e injeção de dependência estão relacionados um ao outro, mas não de forma simétrica.

Na programação orientada a objetos, existem várias formas de implementar o padrão IoC. Um deles é a injeção de dependência, mas nem todo padrão IoC é a injeção de dependência.



A inversão de controle ou inversion of control em inglês (IoC) refere-se a qualquer projeto de software cuja finalidade seja à necessidade de querer controlar o seu o fluxo de execução, de forma automática e transparente, ou seja, abrir mão desse fluxo a um “agente externo”, geralmente um framework.

Container de inversão de controle do Spring

Um assunto chave no Spring é o **container de inversão de controle**, que é compatível com a interface **ApplicationContext**. O Spring cria esse “espaço” em nossa aplicação onde algumas instâncias de objetos podem ser colocadas, tais como grupos de conexão de banco de dados, clientes HTTP, etc. Esses objetos chamamos de **beans** e podem ser usados posteriormente em outras partes da aplicação, geralmente por meio de uma interface para abstrair seu código de implementações específicas.

Um **bean** é um objeto que o Spring gerencia em tempo de execução com o **container de inversão de controle**. Eles são criados e adicionados a um “repositório de objetos” para que possam ser utilizados posteriormente.

O mecanismo para se referir a um desses beans do **ApplicationContext** em outras classes é o que chamamos de **injeção de dependência** e, no Spring, isso é possível por meio de configuração XML ou anotações de código.

Bean

No Spring, os objetos que formam a espinha dorsal da aplicação e são gerenciados pelo **container de inversão de controle** são chamados de **beans**.

Um **bean** é um objeto instanciado, montado e gerenciado por um **container de inversão de controle**. Em outras palavras, um bean é simplesmente um dos muitos objetos em nossa aplicação.

Container de inversão de controle no Spring

A interface `org.springframework.context.ApplicationContext` representa o container de inversão de controle e é responsável por instanciar, configurar e montar os beans.

O container obtém suas instruções sobre quais objetos instanciar, configurar e montar lendo os metadados de configuração são renderizados em XML ou por meio de anotações.