
**UNIVERSIDADE FEDERAL DE LAVRAS**

**GAC116 - Programação Web**

**Discentes:**
Gabriel
Rafael Furtado Morais

# Relatório Prático

**LAVRAS-MG**

**2025**

# Spring Framework (SpringBoot)

Em nossa terceira etapa do trabalho escolhemos refazer o backend da aplicação de e-commerce de jogos com a tecnologia Spring Framework (SpringBoot) por já termos conhecimento da linguagem Java e por nenhum dos integrantes do grupo ter usado essa ferramenta tão elogiada pelo mercado de trabalho. A seguir apresentamos o diferencial do Spring e vamos expor nossa opinião sobre a utilização do django e do Spring.

## Container de Inversão de Controle (IoC)

Na programação imperativa tradicional, o fluxo de execução e a criação de objetos são controlados explicitamente pelo desenvolvedor. Por exemplo, se uma classe A depende de uma classe B, a classe A é responsável por instanciar a classe B (via operador new). Isso gera um acoplamento forte entre as classes. O Spring inverte essa função através do princípio de Inversão de Controle (IoC). No contexto do Spring, o IoC é materializado pelo IoC Container, frequentemente representado pela interface ApplicationContext. O Container é responsável por:

- Instanciação: Criar os objetos da aplicação.
- Configuração: Definir as propriedades desses objetos.
- Montagem: Gerenciar as dependências entre os objetos.
- Ciclo de Vida: Gerenciar todo o ciclo de vida, desde a criação até a destruição dos objetos.

Dessa forma, a aplicação não busca suas dependências; ela as recebe automaticamente do container, isso garante um baixo acoplamento e facilitando testes unitários.

## Spring Beans

Dentro do ecossistema Spring, um objeto que é instanciado, montado e gerenciado pelo IoC Container é denominado Bean. Entretanto, nem todo objeto em uma aplicação Java é um Bean, apenas aqueles cuja gestão é delegada ao Spring. Os Beans são criados com base em definições de configuração, que no Spring Boot são majoritariamente feitas através de anotações nas classes. As principais anotações (estereótipos) que sinalizam ao Container para transformar uma classe em um Bean incluem:

- **@Component**: Um componente genérico gerenciado pelo Spring.
- **@Service**: Indica que a classe contém lógica de negócio.
- **@Repository**: Indica que a classe interage com o banco de dados (Data Access Object), habilitando tradução de exceções de persistência.
- **@Controller** / **@RestController**: Indica que a classe lida com requisições HTTP na camada web.

Por padrão, os Beans no Spring são gerenciados sob o escopo Singleton, o que significa que o Container cria uma única instância de cada Bean e a reutiliza em toda a aplicação, otimizando o uso de memória.

## Injeção de Dependência (DI):

A Injeção de Dependência é o padrão de projeto utilizado pelo IoC Container para implementar a Inversão de Controle. É o processo pelo qual o Container fornece as dependências que o Bean precisa no momento que é criado. Quando o Spring Boot inicializa, ele realiza a varredura de componentes (*Component Scan*), identifica quais Beans precisam de outros Beans para funcionar e realiza a "injeção". A prática recomendada é a Injeção via Construtor, onde as dependências são declaradas como parâmetros no construtor da classe. Isso garante que o objeto nunca seja instanciado em um estado inválido ou incompleto.

## Spring Boot

Enquanto o Spring Framework fornece o Container e o gerenciamento de Beans, o Spring Boot atua como um facilitador através da Auto-Configuração (Auto-Configuration). Ao iniciar a aplicação (anotada com @SpringBootApplication), o Spring Boot:

- Analisa o classpath (bibliotecas presentes no projeto).
- Deduz a intenção do desenvolvedor baseada nas dependências encontradas.
- Configura e registra automaticamente Beans de infraestrutura no Container.

# Principais diferenças entre o Django e Spring Framework

Em nossos trabalhos encontramos algumas  diferenças entre os frameworks que impactaram como realizamos o e-commerce de jogos nas duas tecnologias. Quando fizemos a aplicação utilizando o Django tivemos poucos problemas, a organização do projeto, feita pelo Django, é sucinta e com pouco tempo conseguimos entender o fluxo de trabalho, contamos com diversos funcionalidades pré-prontas como a sessão de administrador, que já veio criado pelo framework, o que facilitou bastante a completude do Checkpoint 2. Além disso, a linguagem de programação Python, em nossa visão, colabora para uma maior facilidade pois é mais “beginner-friendly”. Por fim o django fornece uma forma de fazer o front end da aplicação simples e poderosa, podendo ser utilizado para ambos (frontend e backend) sem tanta perda de qualidade e desempenho da aplicação, ainda mais aliada a um ferramenta de estilização como o Bootstrap.

Em contrapartida o Spring Framework, juntamente com o Spring Boot, necessitou de uma ampla configuração de pacotes, versões, escolhas de gerenciador de

pacotes e escolha de dependências que precisávamos para começar  a desenvolver o mesmo projeto do e-commerce. A estrutura de organização de pastas e o entendimento de cada parte do Spring Framework nos levou mais tempo do que em relação ao Django. Não obstante, todos os recursos já prontos que bastavam serem usados por nós no projeto agora precisavam ser implementados, como área de administrador, o Spring também utiliza uma forma bem peculiar de *annotations* no qual cada um tem seu propósito, entre outros diversos fatores. Portanto no nosso entendimento concluímos que, para aqueles já acostumados em fazer projetos e sistemas web a melhor ferramenta seria, a depender do caso, o Spring Framework que possibilita uma maior liberdade para se criar tais sistemas o mais customizados e específicos possíveis ao preço de uma maior complexidade de configuração e manutenção, já para aqueles que estão começando a fazer seus primeiros sistemas web, não sabem a fundo os conceitos que regem tais sistemas e o que precisam para poderem funcionar a melhor alternativa sem dúvida seria o Django, não é por menos que a disciplina o utiliza. Ele é fácil de entender, de configurar e ensina bem cada um dos conceitos que um sistema web necessita