



Spring Framework

Por João Bosco de Barros Freitas, MSc

Antes de começarmos:

- Neste slide faremos um resumo do framework Spring.
- Para uma leitura completa consulte :
 - <https://spring.io/guides>
 - <https://docs.spring.io/spring/docs/4.2.x/spring-framework-reference/html/>

Breve história do Spring

- Novembro de 2002, Rod Johnson publicou, o livro: *Expert One-on-One J2EE Design and Development*.
- Objetivo:
 - Mostrar a complexidade das aplicações usando o guia do J2EE; e
 - Apresentar uma alternativa
 - mais simples,
 - menos custosa,
 - com tempo de entrega mais realista,
 - mais gerenciável e
 - com melhor performance.

Breve história do Spring

- Além do texto, mais de 30 mil linhas de código Java do Spring Framework.
- **Principais conceitos e fundamentos do Spring foram definidos nessas linhas:**
 - Container de IoC – *Inversion of Control*, ou Inversão de Controle – com um BeanFactory,
 - ApplicationContext e um DI, ou *Dependency Injection* – Injeção de Dependência,
 - Fundamentos **do Spring MVC com Controller, HandlerMapping e associados**,
 - O JdbcTemplate e o conceito de acesso a dados agnóstico à tecnologia.

Qual a diferença para o EE tradicional?

- **Spring é uma plataforma Java que disponibiliza** uma infraestrutura para o desenvolvimento de aplicações corporativas.
- O **Spring cuida da infraestrutura** , cabendo ao desenvolvedor cuidar de sua aplicação/negócio.
- Muito parecido com o proposto pelo Java EE tradicional, não? A diferença está em como o Spring faz. Em uma única expressão, ele não é intrusivo!

Ao contrário do desenvolvimento JEE tradicional, podemos desenvolver toda a aplicação utilizando apenas POJO, classes simples. Não há necessidade de se implementar ou estender nenhum artefato alienígena aos nossos requisitos, como EJB.

Princípio Hollywood



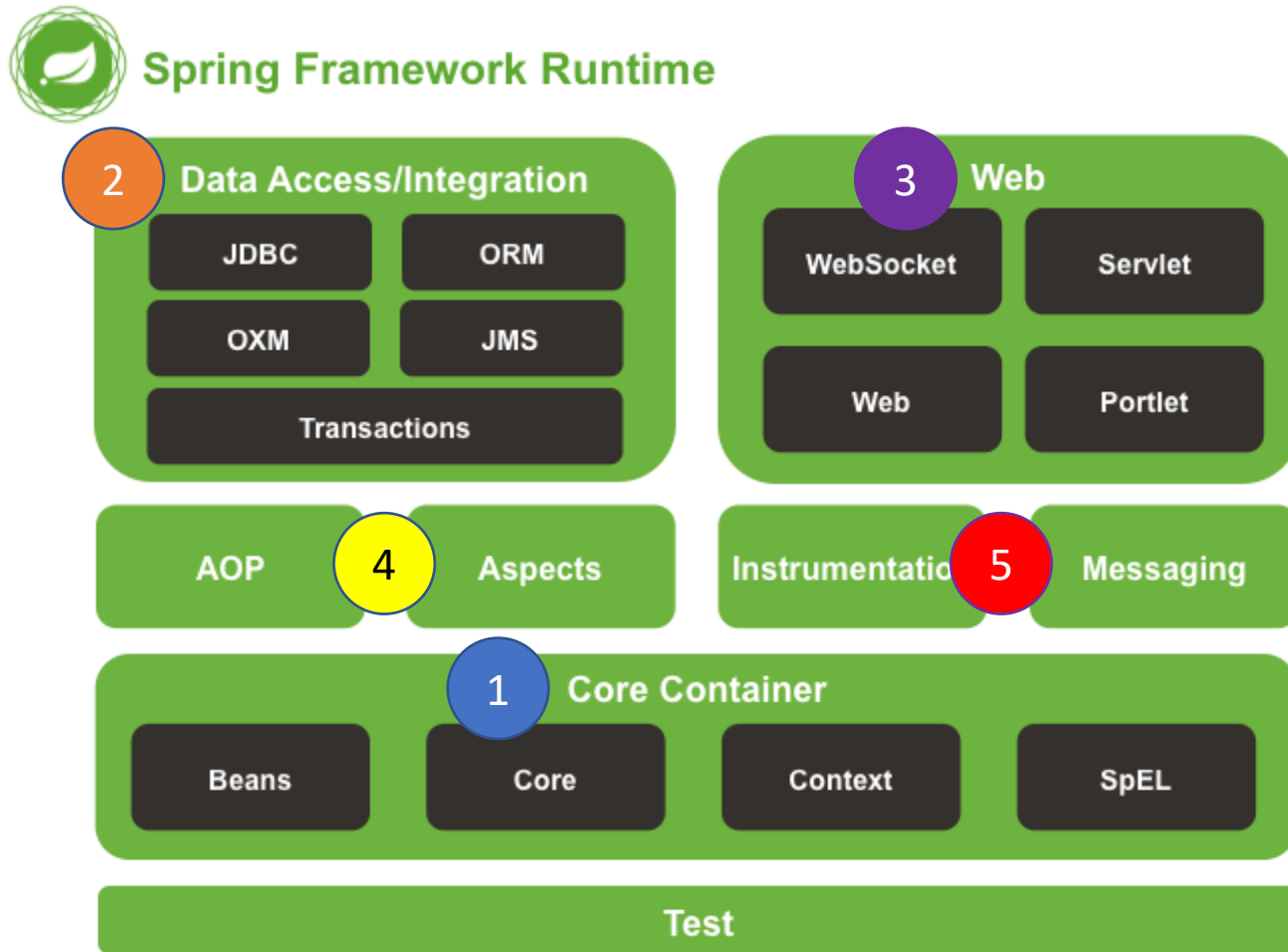
Uma característica importante de um *framework* é que os métodos definidos pelo usuário para estender o framework são invocados pelo próprio *framework*, ao invés de serem chamados pelo aplicativo do usuário.

O Framework geralmente desempenha o papel do programa principal na coordenação e no sequenciamento da atividade do aplicativo (ciclo de vida).

Essa inversão de controle dá aos frameworks o poder de servir como esqueletos extensíveis. Os métodos fornecidos pelo usuário ajustam os algoritmos genéricos definidos na estrutura para uma aplicação específica.

Adaptado pelo autor à partir de Ralph Johnson e Brian Foote

Estrutura do Spring



O Framework Spring consiste em diversos recursos que estão organizados em cerca de 20 módulos.

Esses módulos são agrupados em Core Container, Data Access / Integration, Web, AOP (Programação Orientada a Aspectos), Instrumentação e Teste, conforme mostrado no diagrama ao lado.

Fonte: <https://docs.spring.io/spring/docs/3.0.x/spring-framework-reference/html/overview.html>

Estrutura do Spring 1

- Os módulos Core e Beans[1] fornecem as partes principais da estrutura, incluindo os recursos Inversão de Controle (IoC) e Injeção de Dependência.
- O BeanFactory é uma implementação sofisticada do “Pattern Factory”[2].
 - Permite desacoplar as dependências de uma classe da lógica de sua aplicação.

[1]-Beans são classes java

[2]-Para entender o Design Pattern Factory, visite
https://pt.wikipedia.org/wiki/Factory_Method

Estrutura do Spring

- O módulo Contexto usa a infraestrutura fornecida pelos módulos Core e Beans:
 - Permite acessar objetos por meio do seu nome em um estilo semelhante à JNDI[1].

[1] A **JNDI** ou **Java Naming and Directory Interface** é uma API para acesso a serviços de diretórios. Ela permite que aplicações cliente descubram e obtenham dados ou objetos através de um nome.

Estrutura do Spring

- A camada Acesso a Dados / Integração consiste nos módulos JDBC, ORM, OXM, JMS e Transaction.
 - **JDBC** fornece uma camada de abstração que elimina a necessidade de codificação JDBC
 - **ORM** fornece camadas de integração para APIs populares de mapeamento relacional de objeto, incluindo JPA, JDO, Hibernate.
 - **OXM** fornece uma camada de abstração que suporta implementações de mapeamento de objeto / XML para JAXB, Castor, XMLBeans, JiBX e XStream.
 - **Java Messaging Service (JMS)** contém recursos para produzir e consumir mensagens.
 - **Transaction** suporta o gerenciamento de transações programáticas e declarativas

Estrutura do Spring

- A camada da Web consiste nos módulos Web, Web-Servlet, Web-Struts e Web-Portlet.
- Web fornece recursos de integração, como funcionalidade de upload de arquivos de várias partes e a inicialização do contêiner IoC usando servlet listeners.
- O módulo spring-webmvc fornece uma separação clara entre o código do modelo de domínio e os formulários da Web e integra-se a todos os outros recursos do Spring Framework. Contém a implementação do Web Services REST para aplicativos da Web.
- O módulo spring-webmvc-portlet fornece a implementação MVC a ser usada em um ambiente de portlet.

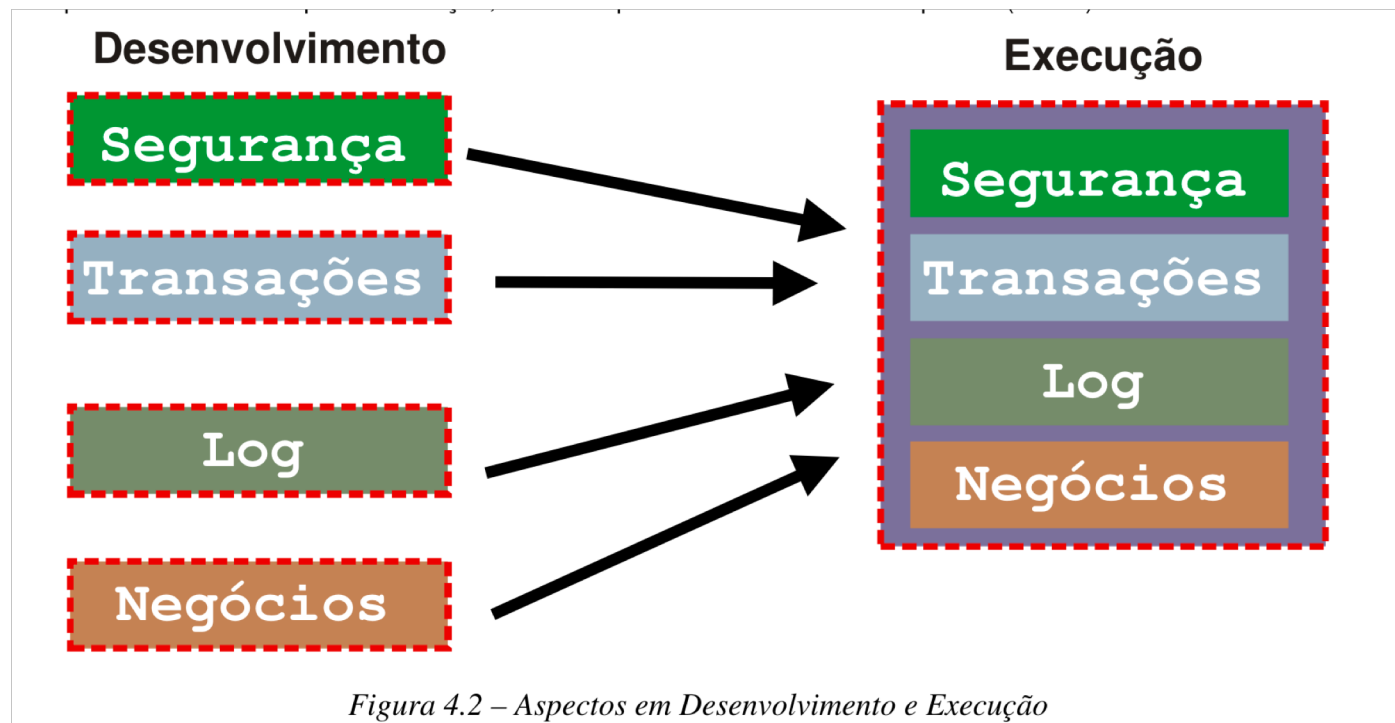
Estrutura do Spring

- A AOP ou “Programação Orientada a Aspectos” é utilizada pelo Spring para separar os conceitos(aspectos) transversais (ou ortogonais) da aplicação das regras de negócio em si.

Aspectos Transversais (Orthogonal Concern): são os diversos conceitos com que um código de aplicação tem que lidar, como regras de negócio, segurança e conexões com bancos de dados. A idéia por trás da AOP é tratar estes conceitos em lugares separados (entenda-se “classes separadas” se estamos falando de Java) enquanto estamos desenvolvendo e uni-las em tempo de execução. Estes conceitos que se integram ao código de negócios são chamados de Aspectos.

Autor: Phillip Calçado, disponível em <http://blog.flexdev.com.br/wp-content/uploads/spring/apostila-spring.pdf>

Estrutura do Spring 4



Autor: Phillip Calçado, disponível em <http://blog.flexdev.com.br/wp-content/uploads/spring/apostila-spring.pdf>

Estrutura do Spring

- O módulo de teste suporta o teste de unidade e teste de integração de componentes Spring com JUnit ou TestNG.
- O módulo de teste também fornece objetos Mock que você pode usar para testar seu código isoladamente.