



DIGITAL  
INNOVATION  
ONE

# Spring(boot) Framework

---

Gleyson Sampaio  
Instrutor Java

# Aula 2: Springboot

## Spring Framework



# Spring Framework

## Projeto Springboot

- O que é Springboot ?
- Configurações manuais
- Starters
- Principais Starters
- Configuração de fábrica



# Springboot

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



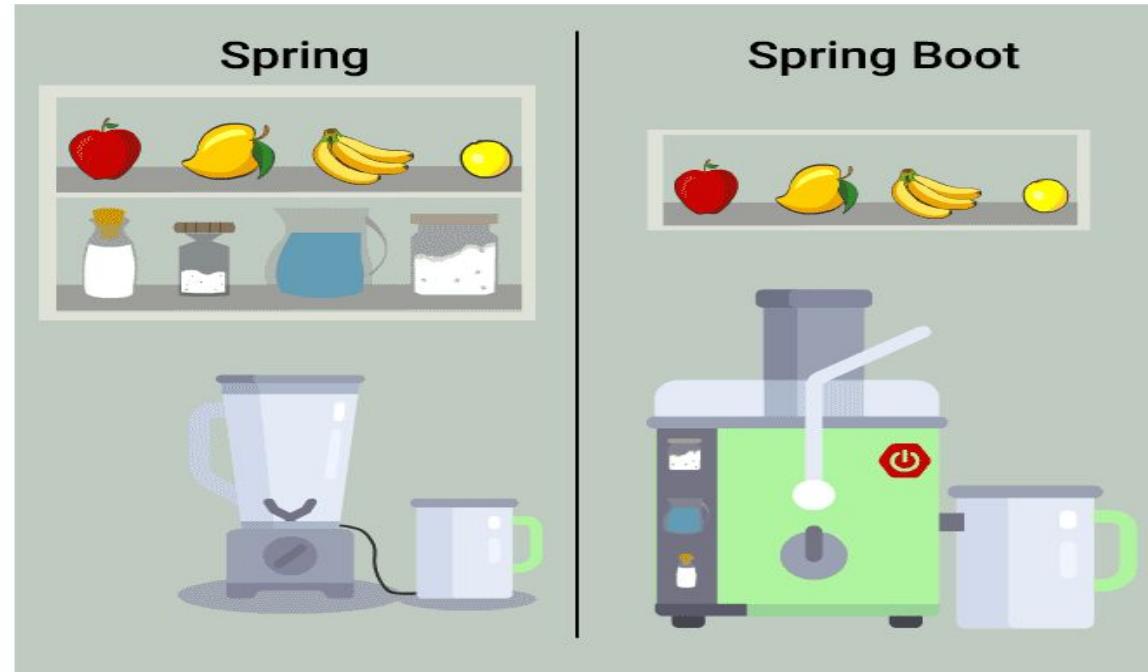
# Antes do Springboot

Desafios com a configuração do projeto.

- Dependência individual
- Verbosidade
- Incompatibilidade de versões
- Complexidade de gestão
- Configurações complexas e repetitivas



# Springboot





# Springboot

Dado que a maior parte das configurações necessárias para o início de um projeto são sempre as mesmas, por que não iniciar um projeto com todas estas configurações já definidas?



# Starters

```
<dependencies>
    <!-- Spring -->
    <dependency>
        <groupId>org.springframework</groupId>
        <artifactId>spring-core</artifactId>
        <version>${springframework.version}</version>
    </dependency>
    <dependency>
        <groupId>org.springframework</groupId>
        <artifactId>spring-context</artifactId>
        <version>${springframework.version}</version>
    </dependency>
    <dependency>
        <groupId>org.springframework</groupId>
        <artifactId>spring-tx</artifactId>
        <version>${springframework.version}</version>
    </dependency>
    <dependency>
        <groupId>org.springframework</groupId>
        <artifactId>spring-orm</artifactId>
        <version>${springframework.version}</version>
    </dependency>
    <dependency>
        <groupId>org.hibernate</groupId>
        <artifactId>hibernate-core</artifactId>
        <version>${hibernate.version}</version>
    </dependency>
    <!-- MySQL -->
    <dependency>
        <groupId>mysql</groupId>
        <artifactId>mysql-connector-java</artifactId>
        <version>${mysql.connector.version}</version>
    </dependency>
    <!-- Joda-Time -->
    <dependency>
        <groupId>joda-time</groupId>
        <artifactId>joda-time</artifactId>
        <version>${joda-time.version}</version>
    </dependency>
    <!-- To map JodaTime with database type -->
    <dependency>
        <groupId>org.Jadira.usertype</groupId>
        <artifactId>usertype.core</artifactId>
        <version>3.0.0.CR1</version>
    </dependency>
</dependencies>
```



## Descriptor de dependência

```
<parent>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-parent</artifactId>
    <version>2.3.4.RELEASE</version>
    <relativePath/> <!-- lookup parent from repository -->
</parent>
<dependencies>
    <dependency>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-data-jpa</artifactId>
    </dependency>
    <!-- MySQL -->
    <dependency>
        <groupId>mysql</groupId>
        <artifactId>mysql-connector-java</artifactId>
        <version>${mysql.connector.version}</version>
    </dependency>
</dependencies>
```



# Starters

## Benefícios

- Coesão
- Versões compatíveis
- Otimização do tempo
- Configuração simples
- Foco no negócio

# Alguns Starters

## Listagem de alguns starters mais utilizados

### **Spring-boot-starter-\***

● **data-jpa:** Integração ao banco de dados via JPA - Hibernate.

- **data-mongodb:** Interação com banco de dados MongoDB.
- **web:** Inclusão do container Tomcat para aplicações REST.
- **web-services:** Webservices baseados na arquitetura SOAP.

# Alguns Starters

## Listagem de alguns starters mais utilizados

### **Spring-boot-starter-\***

**batch:** Implementação de JOBS de processos.

- **test:** Disponibilização de recursos para testes unitários como JUnit
- **openfeign:** Client HTTP baseado em interfaces
- **actuator:** Gerenciamento de monitoramento da aplicação.



# Para saber mais

- <https://www.fusion-reactor.com/blog/the-difference-between-spring-framework-vs-spring-boot/>
- <https://dev.to/eduwyre/settling-spring-vs-spring-boot-debate-8ek>
- <https://www.reply.com/solidsoft-reply/en/content/webservices-soap-and-rest-a-simple-introduction>
- <https://www.geeksforgeeks.org/difference-between-spring-and-spring-boot/>



DIGITAL  
INNOVATION  
ONE

# Spring(boot) Framework

---

Gleyson Sampaio  
Instrutor Java

# Aula 2: Primeiros Passos

## Springboot



# Primeiros passos

- Criando um projeto com initializr
- importando o projeto maven no IntelliJ
- Conhecendo a estrutura spring boot
- Bean e CommandLineRunner



DIGITAL  
INNOVATION  
ONE

# Spring(boot) Framework

---

Gleyson Sampaio  
Instrutor Java

# Aula 2: Beans x Components

Springboot



# Quem são eles?

- Quando usar @Bean ?
- Quando usar @Component?
- Implementar a IoC e DI



DIGITAL  
INNOVATION  
ONE

# Spring(boot) Framework

---

Gleyson Sampaio  
Instrutor Java

# Aula 2: Singleton ou Prototype ?

Springboot



# Quantos irei precisar?

- Conceito de Scope
- Configurando objeto Singleton
- Configurando objetos Prototype



DIGITAL  
INNOVATION  
ONE

# Spring(boot) Framework

---

Gleyson Sampaio  
Instrutor Java

# Aula 2: Properties Value

## Springboot



# Nem tudo é =

- O poderoso application.properties
- @Value
- Default value



DIGITAL  
INNOVATION  
ONE

# Spring(boot) Framework

---

Gleyson Sampaio  
Instrutor Java

# Aula 2: Configuration Properties

Springboot



# É sério isso ?

- O poderoso application.properties
- @ConfigurationProperties (prefix)



DIGITAL  
INNOVATION  
ONE

# Java Persistence API

---

Gleyson Sampaio  
Instrutor Java

# Aula 2: Conceito de ORM e JPA

## Springboot



# ORM e JPA

- O que é ORM ?
- Java Persistence API
- Mapeamentos
- EntityManager

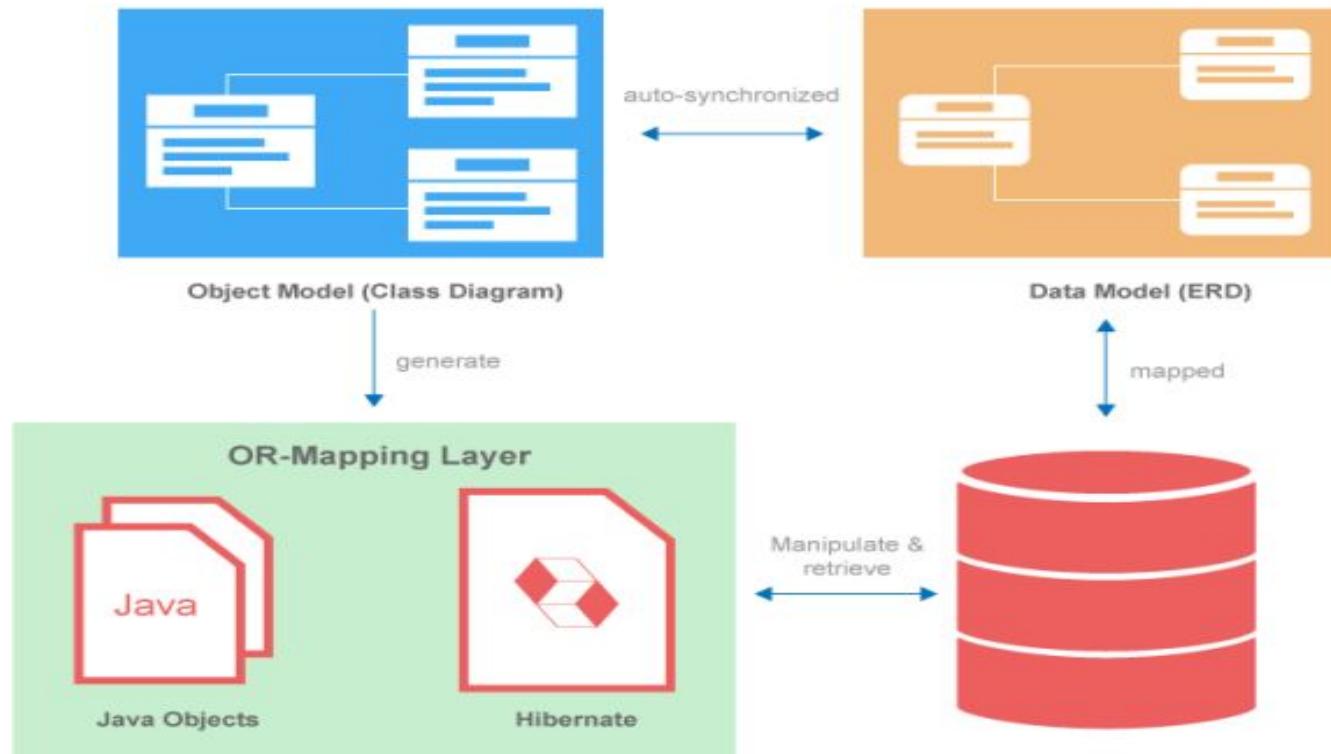
# O que é ORM?

Object-Relational Mapping, Em português, mapeamento objeto-relacional, é um recurso para aproximar o paradigma da orientação a objetos ao contexto de banco de dados relacional.

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



# O que é ORM?





# JPA

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



DIGITAL  
INNOVATION  
ONE

# JPA - Implementações

HIBERNATE



ECLIPSE LINK

eclipse)link

TOP LINK

ORACLE  
TOPLINK

OPEN JPA

OpenJPA



# Mapeamento

Vamos conhecer os aspectos das anotações de mapeamento do JPA

- Identificação
- Definição
- Relacionamento
- Herança
- Persistência



# Mapeamento na prática

```
import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;
import javax.persistence.Table;

@Entity
@Table(name="tb_usuario")
public class Usuario {

    @Id
    @GeneratedValue(strategy=GenerationType.AUTO)
    @Column(name="id_usuario")
    private Long id;

    private String nome;

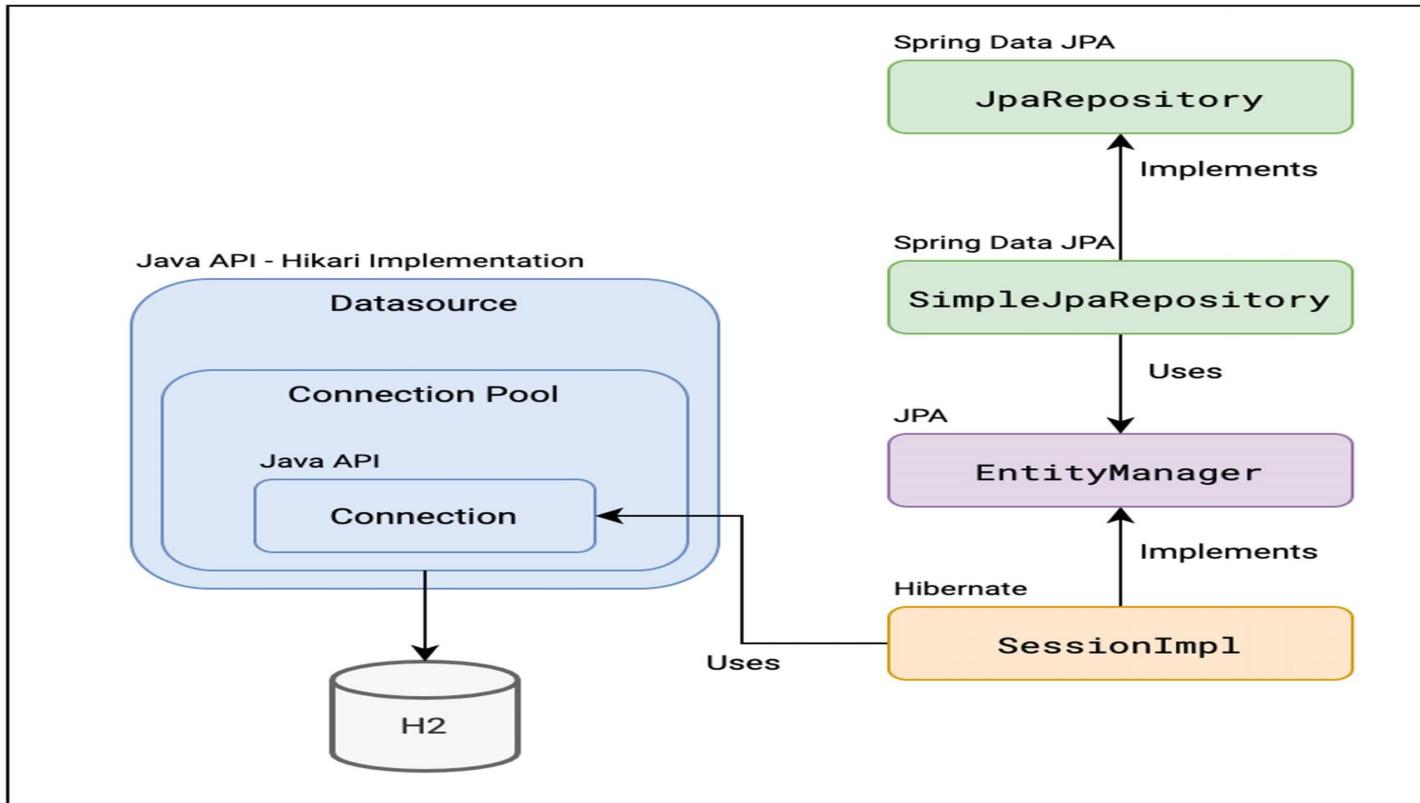
    @Column(name="login_usuario")
    private String login;

    @Column(name="senha_usuario")
    private String senha;

}
```



# EntityManager



# Dúvidas?

- > Fórum do curso
- > Comunidade online (discord)