**1 – Fazer a configuração do ambiente**

1. Java
2. Maven
3. Eclipse
4. Postgresql ou outro BD
5. Heidi ou outra ferramenta para manipulação simplificada de BD
6. Postman ou outra ferramenta para manipulação de operações com JSON
7. GIT
8. Extensão JSONView no Chrome

**2 – Criar um projeto Maven**

**3 – Adicionar no POM do Projeto**

<packaging>jar</packaging>

<!-- define como um projeto tipo Spring Boot -->

<parent>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-parent</artifactId>

<version>2.2.2.RELEASE</version>

</parent>

<!-- define as propriedades do Java e de codificação -->

<properties>

<java.version>13</java.version>

<project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>

<project.report.outputEncoding>UTF-8</project.report.outputEncoding>

</properties>

<dependencies>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-data-rest</artifactId>

</dependency>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-test</artifactId>

<scope>test</scope>

</dependency>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-data-jpa</artifactId>

</dependency>

<dependency>

<groupId>org.postgresql</groupId>

<artifactId>postgresql</artifactId>

</dependency>

<dependency>

<groupId>junit</groupId>

<artifactId>junit</artifactId>

</dependency>

</dependencies>

<build>

<plugins>

<plugin>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-maven-plugin</artifactId>

</plugin>

</plugins>

</build>

**4 – Criar o arquivo application.properties na pasta resource**

**5 – Adicionar as seguintes configurações:**

## Spring DATASOURCE (DataSourceAutoConfiguration & DataSourceProperties)

spring.datasource.driverClassName=org.postgresql.Driver

spring.datasource.url=jdbc:postgresql://localhost/test\_spring

spring.datasource.username=postgres

spring.datasource.password=admin

# The SQL dialect makes Hibernate generate better SQL for the chosen database

spring.jpa.properties.hibernate.dialect= org.hibernate.dialect.PostgreSQLDialect

# Hibernate ddl auto (create, create-drop, validate, update)

spring.jpa.hibernate.ddl-auto=update

spring.jpa.show-sql=false

#server.port = 8090

**6 – Criar um pacote Java raiz no Projeto**

**7 – Criar a Classe Startup com a annotation @SpringBootApplication, na classe deverá ser criado o método main indicando que a classe é o ponto de entrada da aplicação Spring Boot**

**8 – No método main da classe Startup adicionar a seguinte linha de código:**

SpringApplication.run(Startup.class, args);

**9 – Crie os pacotes na raíz: exception, model, controller, repository, services e view**

**10 – No pacote model crie o pacote mock apenas para armazenar uma classe de teste de funcionamento denominada Greeting que poderá receber do controller respectivo um id (long) e um content (String), os campos devem estar encapsulados e final por padrão e os métodos acessadores bem como o construtor com passagem de parâmetros devem ser criados**

**11 – Crie a classe GreetingController com a annotation @RestController, nessa classe criaremos um método chamado greetingParam com a annotation @RequestMapping passando a url que poderá ser chamada com o uso de parâmetros e outro greetingPath com a mesma anotação de cima que poderá ser chamado com a especificação do caminho**

**12 – Crie uma classe Pessoa com a anotação @Entity definindo os campos uma uma Entity JPA qualquer**

**13 -**