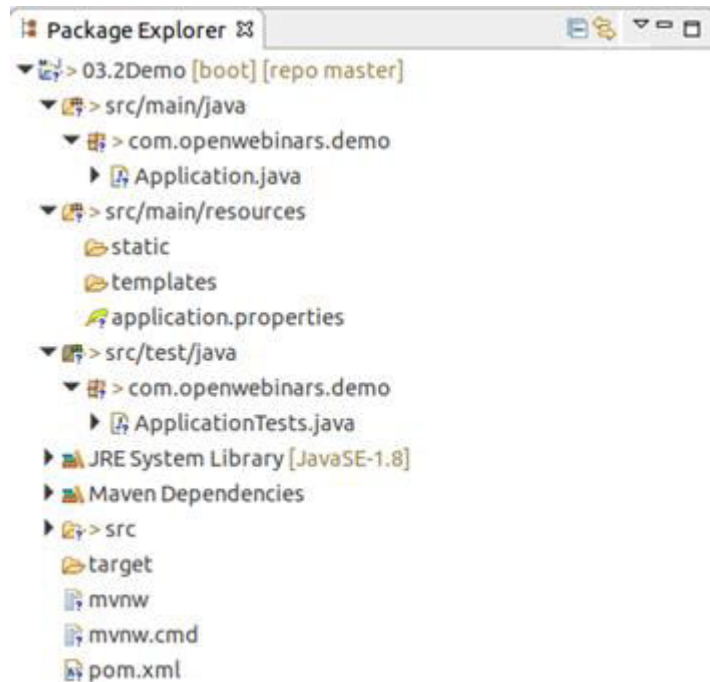


ESTRUCTURA DE UN PROYECTO CON SPRING BOOT



PROYECTO GENERADO CON SPRING INITIALIZR



POM.XML

- ▶ Datos de nuestro proyecto
- ▶ *Pom parent*
- ▶ Propiedades
- ▶ Dependencias
- ▶ Plugins

CÓDIGO FUENTE

▼  > src/main/java

▼  > com.openwebinars.demo

▶  Application.java



Nuestro código de aplicación

▼  > src/main/resources



Recursos: imágenes, plantillas, ficheros de propiedades, sql, ...

📁 static

📁 templates

📄 application.properties

▼  > src/test/java

▼  > com.openwebinars.demo



Test

▶  ApplicationTests.java

LIBRERÍAS

- ▶ JRE System Library [JavaSE-1.8]
- ▼ Maven Dependencies
 - ▶ spring-boot-starter-web-2.1.0.RELEASE.jar - /home/luismi/.m2/repo
 - ▶ spring-boot-starter-2.1.0.RELEASE.jar - /home/luismi/.m2/reposit
 - ▶ spring-boot-2.1.0.RELEASE.jar - /home/luismi/.m2/repository/org
 - ▶ spring-boot-autoconfigure-2.1.0.RELEASE.jar - /home/luismi/.m2/r
 - ▶ spring-boot-starter-logging-2.1.0.RELEASE.jar - /home/luismi/.m2/
 - ▶ logback-classic-1.2.3.jar - /home/luismi/.m2/repository/ch/qos/lo
 - ▶ logback-core-1.2.3.jar - /home/luismi/.m2/repository/ch/qos/logb
 - ▶ log4j-to-slf4j-2.11.1.jar - /home/luismi/.m2/repository/org/apache
 - ▶ log4j-api-2.11.1.jar - /home/luismi/.m2/repository/org/apache/log
 - ▶ jul-to-slf4j-1.7.25.jar - /home/luismi/.m2/repository/org/slf4j/jul-t
 - ▶ javax.annotation-api-1.3.2.jar - /home/luismi/.m2/repository/java
 - ▶ snakeyaml-1.23.jar - /home/luismi/.m2/repository/org/yaml/snake
 - ▶ spring-boot-starter-json-2.1.0.RELEASE.jar - /home/luismi/.m2/repo
 - ▶ jackson-databind-2.9.7.jar - /home/luismi/.m2/repository/com/fasterxml

Nuestra versión de la JVM

Librerías asociadas a cada una de las dependencias Maven que hemos incluido antes.

```
<dependencies>
  <dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-web</artifactId>
  </dependency>

  <dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-test</artifactId>
    <scope>test</scope>
  </dependency>
</dependencies>
```

CLASE MAIN

```
@SpringBootApplication  
public class Application {  
  
    public static void main(String[] args) {  
        SpringApplication.run(Application.class, args);  
    }  
}
```

← Anotación **mágica** que
autoconfigura nuestra aplicación

↑

Comienza a ejecutar Spring, el servidor de Tomcat y tras eso nuestra aplicación.

EJECUCIÓN DEL PROYECTO

Varias posibilidades

- ▶ Desde el IDE: *Run As > Spring Boot App*
- ▶ Desde el terminal
 - ▶ Ejecutar: `mvn spring-boot:run`
 - ▶ Generar jar:
 - ▶ `mvn package`
 - ▶ `cd target`
 - ▶ `java -jar project-name.jar`