

Desarrollo de Aplicaciones Empresariales con Spring Framework Core 5

ISC. Ivan Venor García Baños



spring



Agenda

1. Presentación
2. Objetivos
3. Contenido
4. Despedida



3. Contenido

- i. Introducción a Spring Framework
- ii. Spring Core
- iii. Spring AOP
- iv. Spring JDBC – Transaction
- v. Spring ORM – Hibernate 5
- vi. Spring Data JPA
- vii. Fundamentos Spring MVC y Spring REST
- viii. Fundamentos Spring Security
- ix. Seguridad en Servicios REST
- x. Introducción Spring Boot**



X. Introducción a Spring Boot.



X. Introducción a Spring Boot.

10.1. Spring Boot vs Spring

10.2. Spring Initializr

10.3. Spring Boot CLI

10.4. BOM

10.5. Starters

10.6. Creando un jar ejecutable

10.7. DevTools y Live Reload



10.1. Spring Boot vs Spring

- Spring Framework es el Framework defacto para el desarrollo de aplicaciones Java empresariales.
- Sin embargo requiere:
 - Amplia experiencia en el manejo de dependencias del proyecto de Spring e integración de proveedores externos como Hibernate, Quartz, Jackson, JavaMail, etc.
 - Configuración desde cero. Es necesario configurar todos los beans de infraestructura (boilerplate code).
 - Spring Framework es un conjunto de librerías, módulos, buenas prácticas, clases, etc, sin embargo no define como deben de ser configuradas dada su alta flexibilidad.

by Pivotal™

10.1. Spring Boot vs Spring

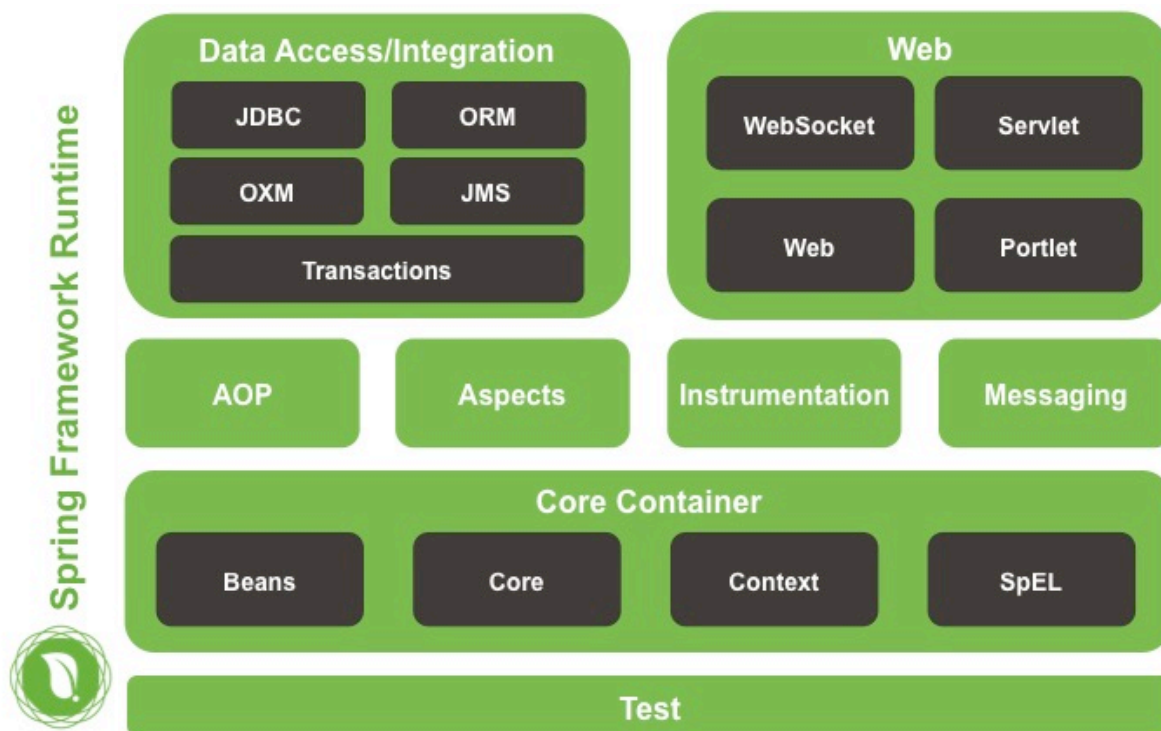
- Spring Framework:
 - No exige un estandar de desarrollo y ni de configuración.
 - Es altamente flexible.
- Integrar otros frameworks y sub-proyectos de Spring cada vez se hace más complicado.



ring
by Pivotal™

10.1. Spring Boot vs Spring

- Spring Framework módulos.



10.1. Spring Boot vs Spring

- Spring Framework:



by Pivotal™

10.1. Spring Boot vs Spring

- Spring Boot es un proyecto “umbrella” que esta construido sobre Spring Framework.
- Ha sido desarrollado por la misma comunidad de Spring Framework en el cuál colaboran los más exitosos desarrolladores Java open-source de la industria aplicando las mejores prácticas de desarrollo.
- Spring Boot es:
 - Opinonado debido a que está basado en un estandar de desarrollo dirigido por sus colaboradores.
 - Hacer “pair-programming” con el equipo de Spring Boot.



Josh Long (龙之春, जोश) •
@starbuxman

Following

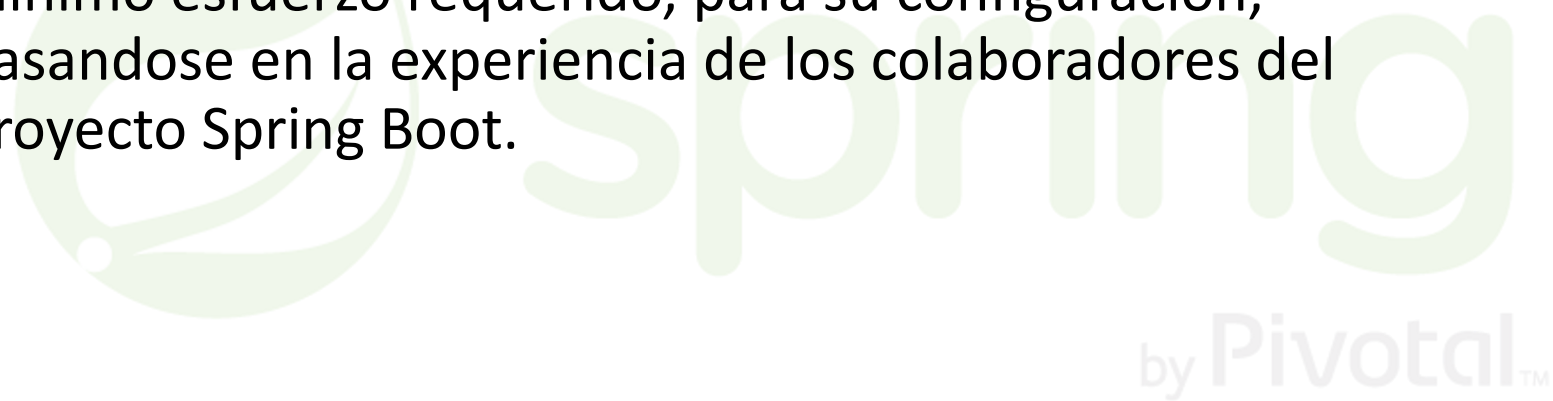
a reminder: @SpringBoot lets you pair-program with the #Spring team.

7:28 PM - 21 Sep 2017 from San Antonio, TX

pivotal™

10.1. Spring Boot vs Spring

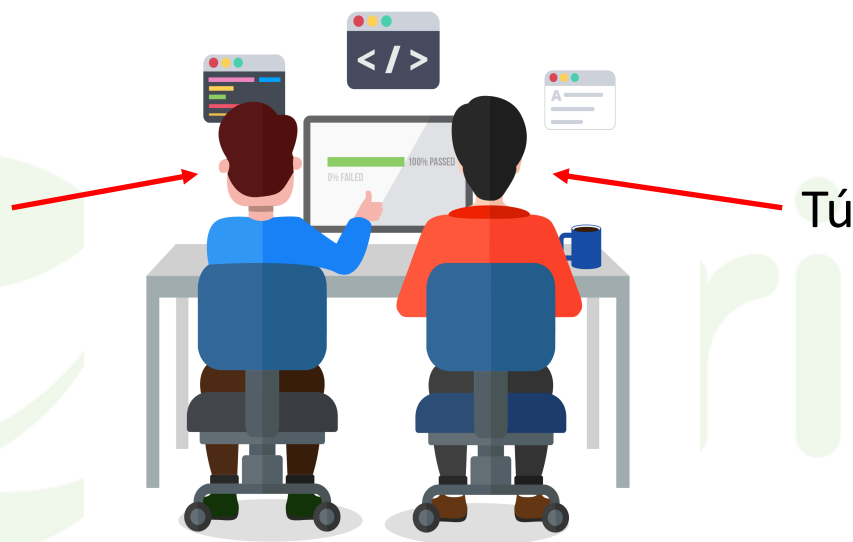
- El principal objetivo de Spring Boot es:
 - Permitirle a los desarrolladores crear aplicaciones productivas basadas en Spring Framework con el mínimo esfuerzo requerido, para su configuración, basandose en la experiencia de los colaboradores del proyecto Spring Boot.



10.1. Spring Boot vs Spring

- Spring Boot te permite hacer “pair-programming” con el equipo de desarrollo de Spring.
- Convención sobre configuración.

Alguien experto
en Spring Framework

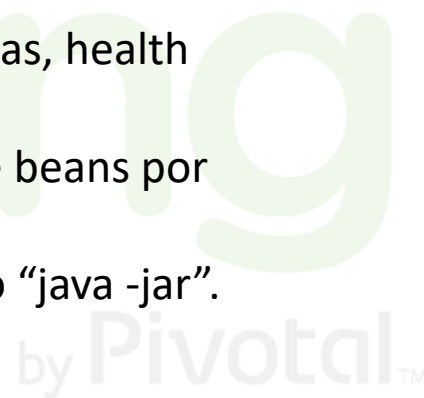


Tú

ring
by Pivotal™

10.1. Spring Boot vs Spring

- Spring Boot:
 - Permite crear aplicaciones standalone y web desde un simple método “main”.
 - Permite el despliegue de servidores web (Servlet containers) embebidos como Tomcat, Jetty y Undertow (No requiere despliegue de archivos WAR).
 - Provee dependencias “opinionadas” de terceros por default.
 - Genera los beans necesarios para el aplicativo simplemente escaneando las librerías detectadas en el classpath.
 - Provee características productivas generales tal como: métricas, health checks y configuración externalizada.
 - No es un generador de código y no requiere configuración de beans por XML.
 - Fácil de ejecutar aplicaciones Spring Boot mediante comando “java -jar”.

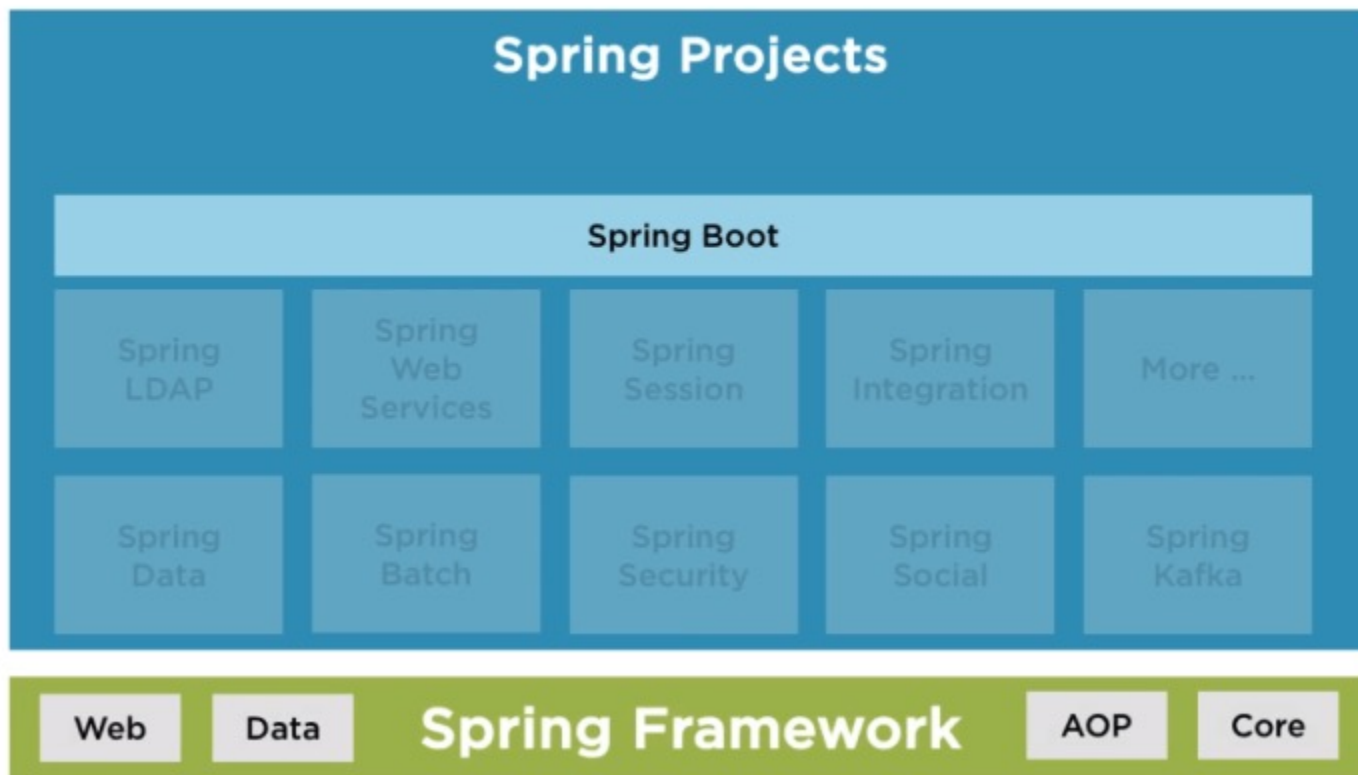


10.1. Spring Boot vs Spring

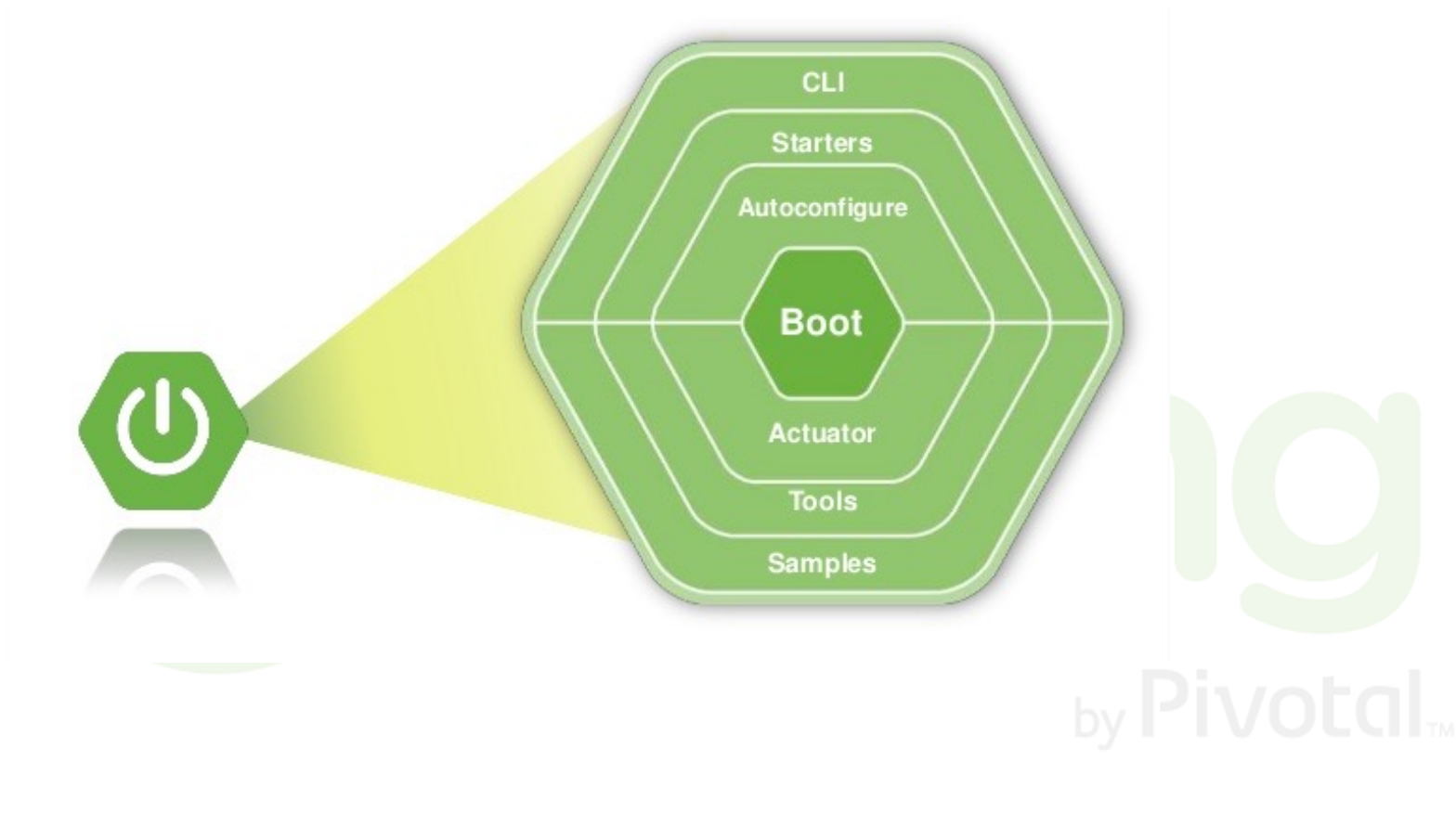
- Spring Boot.



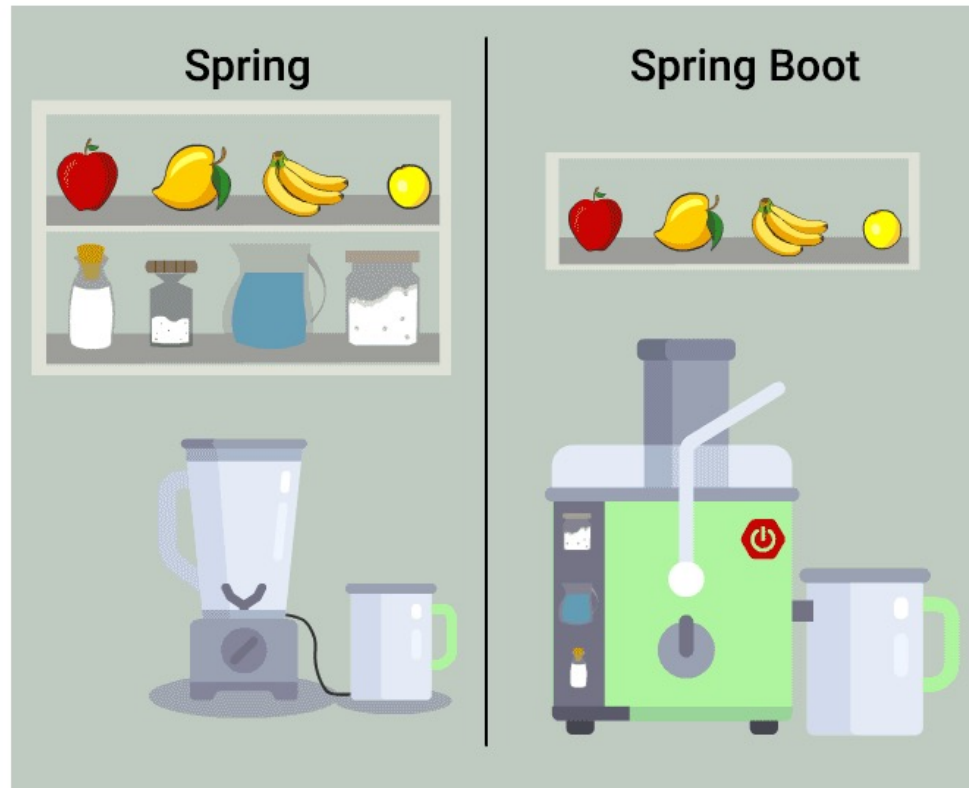
10.1. Spring Boot vs Spring



10.1. Spring Boot vs Spring

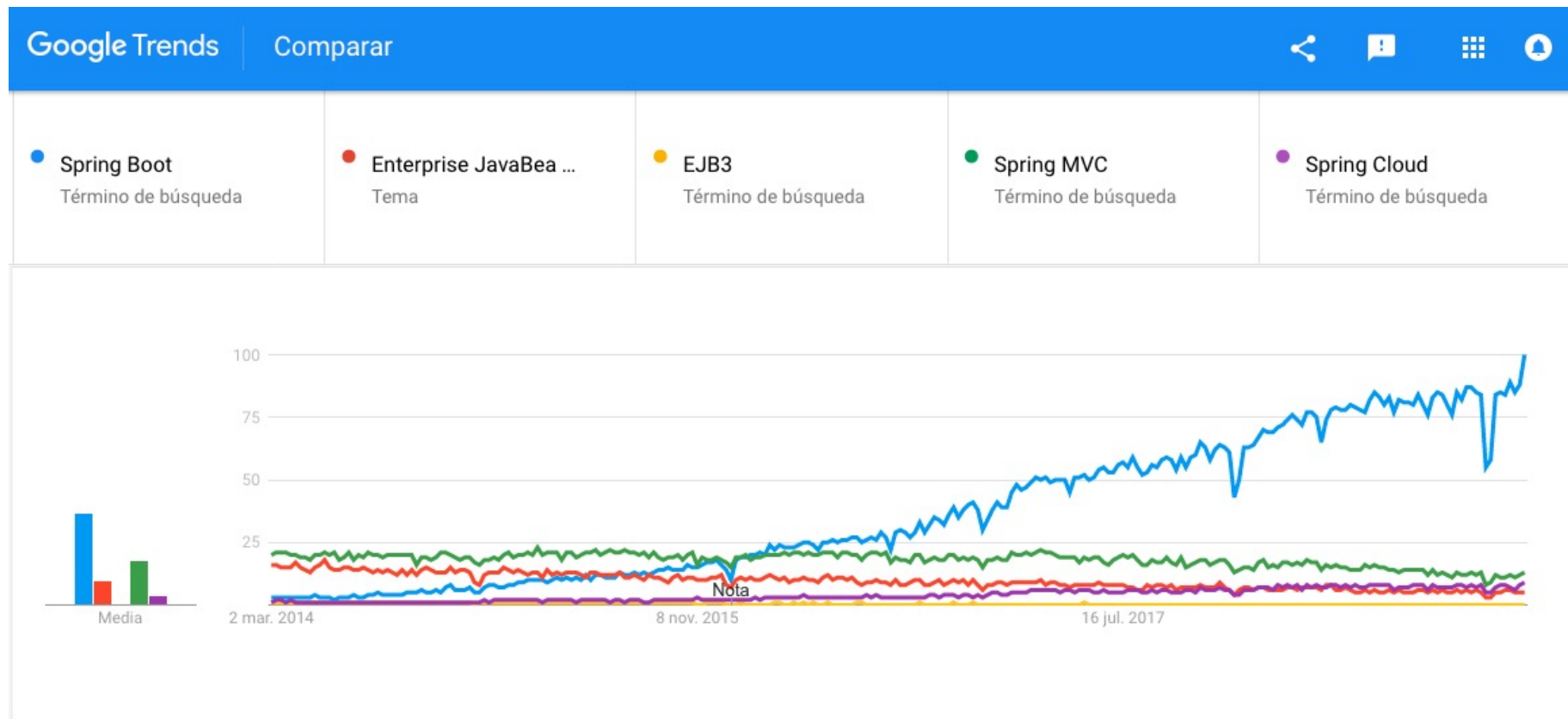


10.1. Spring Boot vs Spring



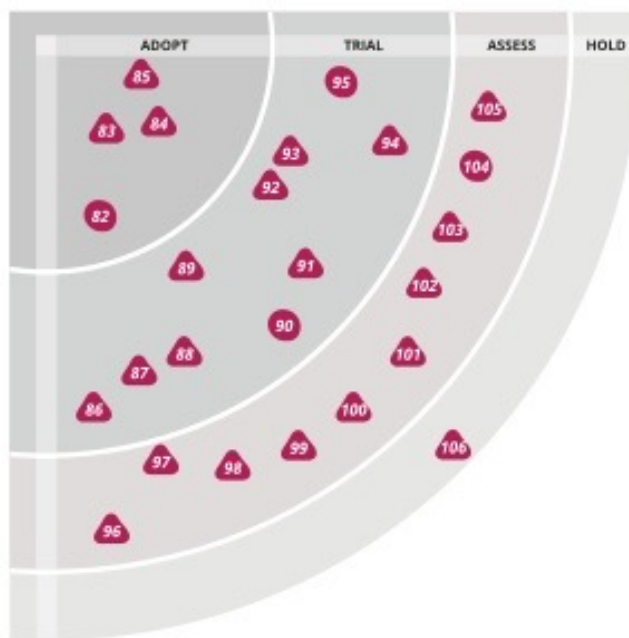
ng
by Pivotal™

10.1. Spring Boot vs Spring



10.1. Spring Boot vs Spring

ThoughtWorks Technology Radar



LANGUAGES & FRAMEWORKS

ADOPT

- 82. ES6
- 83. React.js
- 84. Spring Boot
- 85. Swift

TRIAL

- 86. Butterknife new
- 87. Dagger new
- 88. Dapper new
- 89. Ember.js
- 90. Enlive
- 91. Fetch new
- 92. React Native
- 93. Redux new
- 94. Robolectric new
- 95. SignalR

ASSESS

- 96. Alamofire new
- 97. AngularJS
- 98. Aurelia new
- 99. Cylon.js new
- 100. Elixir
- 101. Elm
- 102. GraphQL new
- 103. Immutable.js new
- 104. OkHttp
- 105. Recharts new

HOLD

- 106. JSPatch new

ThoughtWorks Technology Radar April '16
<https://www.thoughtworks.com/radar>





X. Introducción a Spring Boot.

10.1. Spring Boot vs Spring

10.2. Spring Initializr

10.3. Spring Boot CLI

10.4. BOM

10.5. Starters

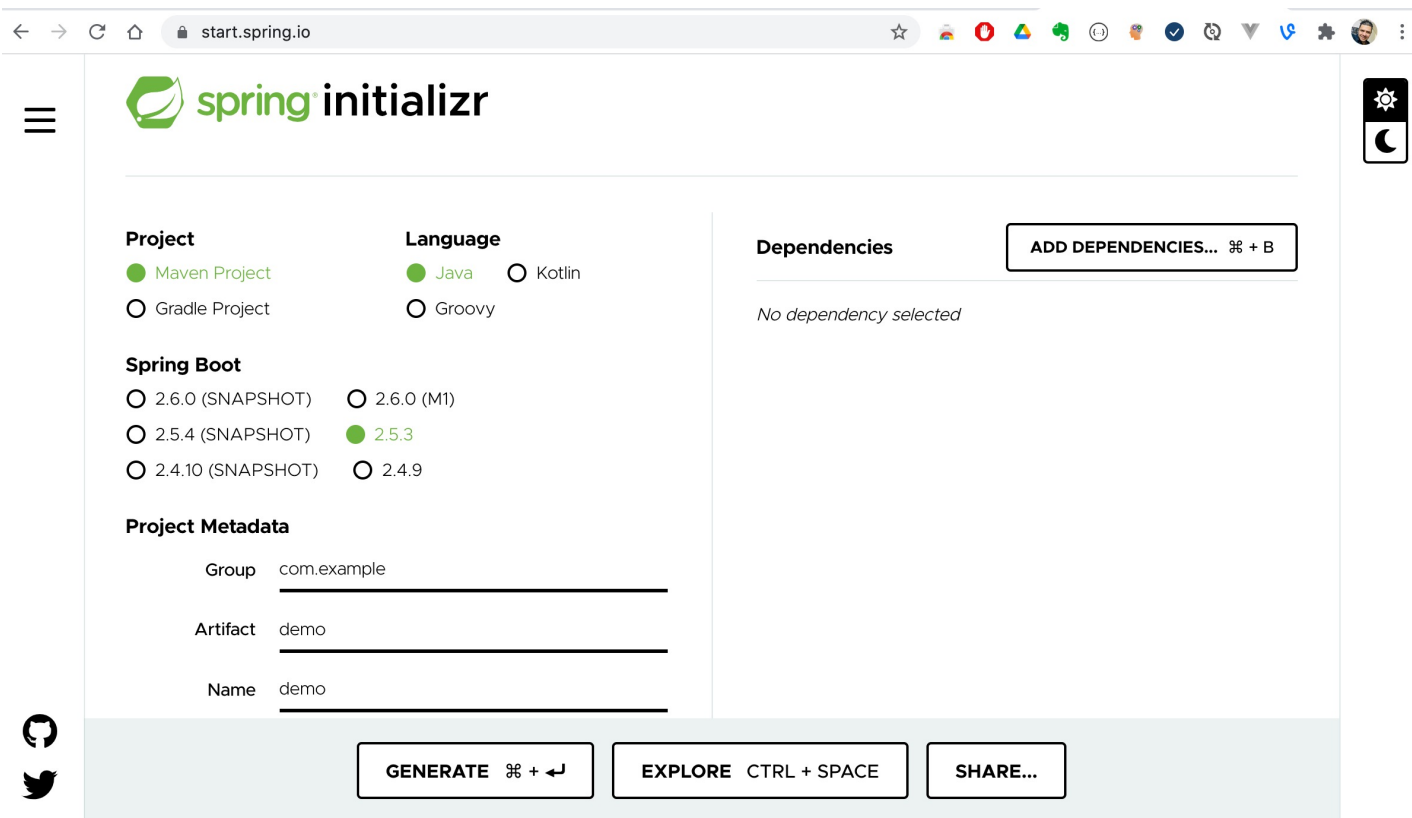
10.6. Creando un jar ejecutable

10.7. DevTools y Live Reload



10.2. Spring Initializr

- Es una interface web para inicializar un proyecto con Spring Boot:



The screenshot shows the Spring Initializr web interface in a browser. The URL bar shows 'start.spring.io'. The interface has a sidebar with a hamburger menu and a dark mode toggle. The main content area is divided into three sections: Project, Language, and Spring Boot. The Project section has radio buttons for 'Maven Project' (selected) and 'Gradle Project'. The Language section has radio buttons for 'Java' (selected) and 'Kotlin', and 'Groovy'. The Spring Boot section has radio buttons for versions: '2.6.0 (SNAPSHOT)', '2.6.0 (M1)', '2.5.4 (SNAPSHOT)', '2.5.3' (selected), '2.4.10 (SNAPSHOT)', and '2.4.9'. Below these is the 'Project Metadata' section with input fields for 'Group' (com.example), 'Artifact' (demo), and 'Name' (demo). To the right of these sections is a 'Dependencies' section with a button 'ADD DEPENDENCIES... ⌘ + B' and the text 'No dependency selected'. At the bottom are three buttons: 'GENERATE ⌘ + ↵', 'EXPLORE CTRL + SPACE', and 'SHARE...'. On the right side of the interface, there is a large, faint 'd' logo and the text 'otal™'.

<https://start.spring.io/>

10.2. Spring Initializr

Practica 36. Spring Initializr

- Inicializar un proyecto Spring Boot mediante Spring Initializr.
- Generar un proyecto con las siguientes características:
 - Maven Project con Java y usando Spring Boot 2.4.x (o la más actual, NO SNAPSHOT)
 - Group: com.truper.springboot
 - Artifact: 7-Spring-Initializr
 - Dependencias: Web
 - Package Name: com.truper.springboot
- Generar proyecto e importar en eclipse / STS.
- Implementar un @RestController simple, ejecutar el proyecto y abrir navegador <http://localhost:8080>



X. Introducción a Spring Boot.

10.1. Spring Boot vs Spring

10.2. Spring Initializr

10.3. Spring Boot CLI

10.4. BOM

10.5. Starters

10.6. Creando un jar ejecutable

10.7. DevTools y Live Reload



10.3. Spring Boot CLI

- Herramienta de línea de comandos para crear un prototipo de proyecto basado en Spring Boot “production-ready”; es similar a Spring Initializr.
- Requiere instalación:
 - <https://docs.spring.io/spring-boot/docs/current/reference/html/getting-started-installing-spring-boot.html#getting-started-installing-the-cli>



10.3. Spring Boot CLI

- Spring Boot con Spring Boot CLI



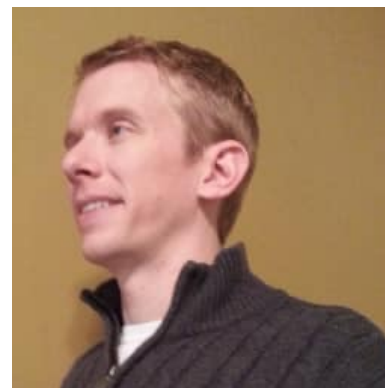
Rob Winch

@rob_winch

@Controller

```
class ThisWillActuallyRun {  
    @RequestMapping("/")  
    @ResponseBody  
    String home() {  
        "Hello World!"  
    }  
}
```

15:12 - 6 ago. 2013



ring
by Pivotal™

https://twitter.com/rob_winch/status/364871658483351552?lang=es

10.3. Spring Boot CLI

- Spring Boot CLI

@RestController

```
public class AlumnosController {  
    @RequestMapping("/")  
    public String saludo() {  
        return "Bienvenidos al curso de Spring Boot";  
    }  
}
```

Ejecuta:

spring run AlumnosController.java

by Pivotal™



X. Introducción a Spring Boot.

10.1. Spring Boot vs Spring

10.2. Spring Initializr

10.3. Spring Boot CLI

10.4. BOM

10.5. Starters

10.6. Creando un jar ejecutable

10.7. DevTools y Live Reload



spring
by Pivotal™

10.4. BOM – Bill of Materials

- **¿Qué es un POM? (Maven)**
 - Project Object Model.
 - Archivo XML que contiene información y configuración del proyecto, utilizado por Maven, para importar sus dependencias.
- **¿Qué es un BOM?**
 - Bill of Materials (Lista de materiales).
 - Es un tipo especial de POM que es utilizado para controlar las versiones de las dependencias de un proyecto.
 - Provee un lugar centralizado para administrar las versiones de dichas dependencias.
 - Provee flexibilidad para agregar una dependencia evitando la preocupación referente a la versión compatible de la misma con el resto de dependencias.

by Pivotal™

10.4. BOM – Bill of Materials

- **¿Cómo utilizar un BOM en nuestro proyecto?**
 - Hacer que nuestro proyecto herede de un POM padre (BOM).
 - Un proyecto puede heredar de un solo proyecto padre, no es eficiente en definiciones de proyectos amplios y complejos (por ejemplo: Requiere Spring Cloud).
 - Importar uno o varios BOMs tal como sea requerido.



10.4. BOM – Bill of Materials

```
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">
```

```
<modelVersion>4.0.0</modelVersion>
```

```
<groupId>com.truper.springboot</groupId>
```

```
<artifactId>1-introduccion</artifactId>
```

```
<version>0.0.1-SNAPSHOT</version>
```

```
<parent>
```

```
<groupId>org.springframework</groupId>
```

```
<artifactId>spring-framework-bom</artifactId>
```

```
<version>5.1.5.RELEASE</version>
```

```
<type>pom</type>
```

```
<scope>import</scope>
```

```
</parent>
```

```
...
```

Heredar de POM (BOM)
padre



Spring
by Pivotal™

10.4. BOM – Bill of Materials

```
...  
<dependencies>  
    <dependency>  
        <groupId>org.springframework</groupId>  
        <artifactId>spring-context</artifactId>  
    </dependency>  
    <dependency>...</dependency>  
</dependencies>  
</project>
```

No requiere especificar versión.



10.4. BOM – Bill of Materials

```
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">
```

```
<modelVersion>4.0.0</modelVersion>
```

```
<groupId>com.truper.springboot</groupId>
```

```
<artifactId>1-introduccion</artifactId>
```

```
<version>0.0.1-SNAPSHOT</version>
```

```
<dependencies>
```

```
<dependency>
```

```
<groupId>org.springframework</groupId>
```

```
<artifactId>spring-context</artifactId>
```

```
</dependency>
```

```
<dependency>...</dependency>
```

```
</dependencies>
```

```
...
```

No requiere
especificar versión.

10.4. BOM – Bill of Materials

```
...  
<dependencyManagement>  
  <dependencies>  
    <dependency>  
      <groupId>org.springframework</groupId>  
      <artifactId>spring-framework-bom</artifactId>  
      <version>5.1.5.RELEASE</version>  
      <type>pom</type>  
      <scope>import</scope>  
    </dependency>  
  </dependencies>  
</dependencyManagement>  
</project>
```

Importación de BOM



X. Introducción a Spring Boot.

10.1. Spring Boot vs Spring

10.2. Spring Initializr

10.3. Spring Boot CLI

10.4. BOM

10.5. Starters

10.6. Creando un jar ejecutable

10.7. DevTools y Live Reload



spring

by Pivotal™

10.5. Starters

- Los “starters” son dependencias (jar’s) vacíos, que únicamente contienen definiciones de dependencias relacionadas a alguna técnica o funcionalidad particular.
- Son un conjunto de descriptores de dependencias que se incluirán en el proyecto si dicho “starter” es incluido en el proyecto.
- Los “starters” evitan que el desarrollador tenga que preocuparse por incluir todas las dependencias relacionadas a una funcionalidad, implementada con Spring en particular.
- Si el proyecto requiere Spring Data JPA para implementar acceso a datos, únicamente es necesario incluir la dependencia “spring-boot-starter-data-jpa” al proyecto.

10.5. Starters

- Existen múltiples “starters” proveídos por la comunidad de Spring Boot listos para ser incluidos en un proyecto Java empresarial con Spring Boot:
 - <https://docs.spring.io/spring-boot/docs/current/reference/htmlsingle/#spring-boot-starter-tomcat>
 - spring-boot-starter
 - spring-boot-starter-activemq
 - spring-boot-starter-amqp
 - spring-boot-starter-aop
 - spring-boot-starter-batch
 - spring-boot-starter-cache
 - spring-boot-starter-data-elasticsearch
 - spring-boot-starter-data-jdbc
 - spring-boot-starter-data-jpa
 - spring-boot-starter-data-mongodb
 - spring-boot-starter-data-redis
 - spring-boot-starter-web
 - spring-boot-starter-web-services
 - spring-boot-starter-webflux
 - spring-boot-starter-data-rest
 - spring-boot-starter-oauth2-client
 - spring-boot-starter-oauth2-resource-server
- Y muchos más ...



X. Introducción a Spring Boot.

10.1. Spring Boot vs Spring

10.2. Spring Initializr

10.3. Spring Boot CLI

10.4. BOM

10.5. Starters

10.6. Creando un jar ejecutable

10.7. DevTools y Live Reload



10.6. Creando un jar ejecutable

- Spring Boot genera aplicaciones autocontenidas que incluyen todo lo necesario (dependencias) para poder ejecutarse, también son llamadas aplicaciones “fat jar/uber jar”.
- Por default Spring Boot genera aplicaciones empaquetadas como jar
- Existen 3 modos de generar aplicaciones Spring Boot autocontenidas:
 - Spring CLI mediante comando “spring jar <nombre-de-jar> <archivos>”
 - Maven mediante spring-boot-maven-plugin starter y comando: “mvn package”
 - Gradle mediante “spring-boot” plugin y comando “gradle build” (fuera de alcance).

10.6. Creando un jar ejecutable

- Ventajas de crear una aplicación autocontenida:
 - No requiere más dependencias para ejecutarse.
 - Se ejecuta mediante “java -jar”.
 - Se ejecuta donde sea que contenga una JVM.
 - No requiere servidor de aplicaciones.
 - Facilita el diseño de arquitecturas distribuidas.
 - Ideal para arquitecturas de Microservicios.
 - Ideal para empaquetar y desplegar en la nube. (PaaS y/o IaaS).





X. Introducción a Spring Boot.

10.1. Spring Boot vs Spring

10.2. Spring Initializr

10.3. Spring Boot CLI

10.4. BOM

10.5. Starters

10.6. Creando un jar ejecutable

10.7. DevTools y Live Reload



spring
by Pivotal™

10.7. DevTools y Live Reload

- Spring Boot 1.3+, a través de **spring-boot-devtools**, agrega funcionalidades que facilitan el desarrollo entre las cuales destacan:
 - Valores de propiedades por default:
 - `spring.thymeleaf.cache`
 - Entre otras
 - Reinicio automático de la aplicación.
 - Spring Boot DevTools reinicia la aplicación en automático al detectar cambios en los archivos del classpath (anteriormente usado JRebel o Spring Loaded).
 - Live Reload (Recarga activa, web-browser):
 - Spring Boot DevTools expone un servidor Live Reload para recargar la página web desde el navegador de forma automática. Requiere plugin en el navegador (Live Reload client).
 - Puenteo para Debug Remoto (Remote Debug Tunneling):
 - Puentea JDWP (Java Debug Wire Protocol) logs sobre HTTP exponiendo puerto 80 y 443.
 - Reinicio y actualización remota (Remote Update and Restart):
 - Monitoreo de cambios en archivos del classpath de la aplicación de forma remota, utilizado en conjunto con Live Reload.

10.2. Spring Initializr

Practica 37. Spring Boot. Spring MVC – Spring Data Mongo – Spring Security.

- Inicializar un proyecto Spring Boot mediante Spring Tool Suite.
- Generar un proyecto con las siguientes características:
 - Maven Project con Java y usando Spring Boot 2.4.x (o la más actual, NO SNAPSHOT)
 - Group: com.truper.springboot
 - Artifact: 8-spring-boot-intro
 - Dependencias: Data MongoDB, Embedded Mongo, Security, Web, DevTools y Lombok
 - Package Name: com.truper.springboot
- Generar proyecto e importar en eclipse / STS.
- Implementar una aplicación web completa, con seguridad y acceso a datos con MongoDB (guiado por instructor).

