Desarrollo backend/frontend con Spring Boot.

By Alejandro Ambroa

Lo que vamos a ver en esta presentación

Spring boot

Esta presentación cubrirá los conceptos básicos de Spring y Spring boot. Veremos cómo desarrollar una aplicación cliente/servidor típica. Para ello usaremos algunos módulos de Spring y diversas bibliotecas:

Aplicación de ejemplo - Incidencias.

Se trata de un aplicación con funcionalidad básica que mostrará incidencias y las recibirá mediante websocket y servicio REST.

¿Qué es Spring Framework?

Escrito inicialmente por Rod Johnson, fue lanzado por primera vez en el mes de Junio del año 2003 bajo la licencia **Apache 2.0**, siendo una plataforma Java de código abierto. Convirtiéndose desde entonces en el framework más popular para Java empresarial, para crear **código de alto rendimiento**, liviano y reutilizable. Actualmente es casi un estándar *de facto* en el mundo Java.

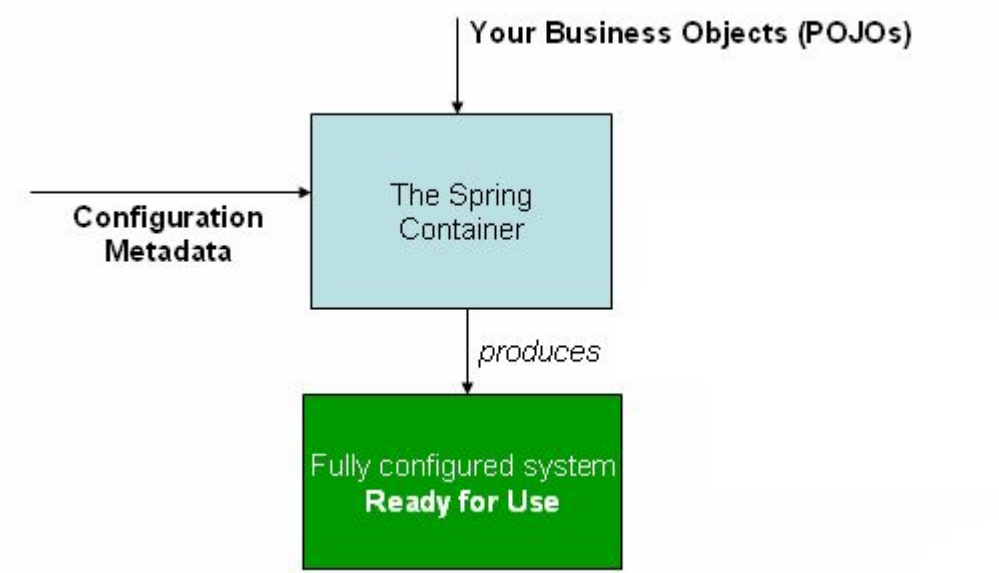
¿Qué es Spring boot?

Spring Boot es una infraestructura que forma parte de Spring Framework y permite construir y mantener una aplicación Spring de forma mucho más sencilla y rápida.

Está sobre todo pensada para el desarrollo de aplicaciones standalone.

Tanto Spring como Spring boot tienen una amplísima y detallada documentación. Spring es, hoy día, un estándar de facto en el mundo Java.

Spring se basa en un contenedor IoC



Spring puede configurarse mediante ficheros XML descriptivos o Anotaciones Java

Usando ficheros XML, definimos los beans (clases Java) y sus dependencias así:

El uso de ficheros XML, sobre todo en proyectos grandes, puede hacerse engorroso y a veces difícil de mantener. A estos ficheros se les llama “Spring context file”

```
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
	xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:context="http://www.springframework.org/schema/context"
	xsi:schemaLocation="http://www.springframework.org/schema/beans
	http://www.springframework.org/schema/beans/spring-beans.xsd
	http://www.springframework.org/schema/context
	http://www.springframework.org/schema/context/spring-context.xsd">
	<!-- <context:component-scan base-package="com.in28minutes.spring.basics"/> -->
	<bean id="xmlStringBean1" class="java.lang.String">
		<constructor-arg value="stringBean1" />
	</bean>
	<bean id="xmlStringBean2" class="java.lang.String">
		<constructor-arg value="stringBean2" />
	</bean>
</beans>
```

Para evitar esto, Spring comenzó a hacer un uso en profundidad de las anotaciones Java. El contendor IoC de Spring examina el classpath de la aplicación y analiza las anotaciones para actuar de forma determinada.

Con la siguiente anotación, Spring instancia la clase una vez y la almacena en su contenedor IoC. Así mismo gestiona todas las dependencias mediante la anotación Autowired.

```
@Service

public class IncidencesServiceImpl implements IncidencesService {

    @Autowired

    private IncidenceRepository incidenceRepository;
```

La anotación *Service* le indicará al contenedor IoC de Spring que debe instanciar como Singleton la clase *IncidencesServiceImpl*.

La anotación Autowired inyectará de forma automática una instancia de *IncidenceRepository.*

Con todo esto, a veces es engorroso montar una aplicación Spring. Sólo debemos tener en cuenta el gran número de módulos Spring (o API's) que existen para añadir funcionalidades:

- Spring Data (para acceso a datos, JPA)
- Spring Security (securizar la aplicación)
- Spring Integration (integración de sistemas y tests)
- Spring Cloud
- Spring MVC
- Spring WebFlux

....

Spring Boot nos permitirá incluir aquellos módulos que necesitemos en nuestra de manera sencilla.

Aplicación de ejemplo: Incidencias.

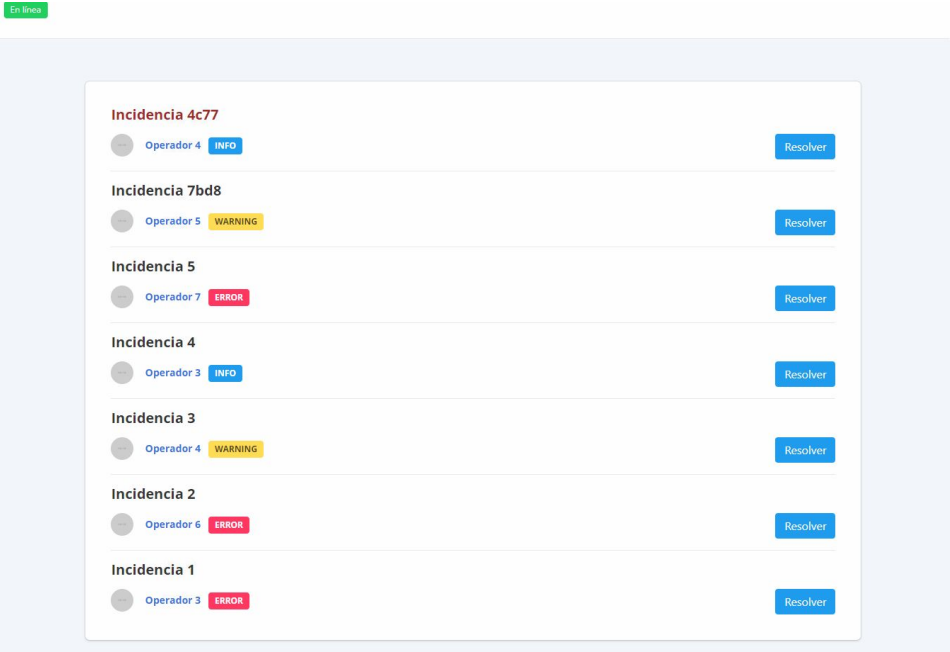
Aplicación de ejemplo.

La aplicación es extremadamente simple. Consiste en una sola página Web donde se muestra un listado de incidencias que genera el backend en tiempo real y un botón que activa su resolución.

Esta mecánica tan simple se usa para cubrir ejemplos básicos de:

- Organización de bibliotecas de dependencias para desarrollo de software en Windows 10 usando la herramienta **Scoop** (https://scoop.sh/)
- Organización de un proyecto Spring Boot.
- Uso de un proyecto Spring Boot en Eclipse.
- Uso de los aspectos básicos de **Spring Boot** como Configuraciones, Controladores, vistas con **Thymeleaf**
- Capa de datos mediante JPA y repositorios, usando base de datos H2
- Arquitectura de servicios.
- Controladores **REST** básicos.

Aplicación de ejemplo: Incidencias.



Fuera de tecnologías Spring Boot, el Frontend usa *Vue* como framework frontend y [*Bulma*](https://bulma.io/) como framework CSS. El front ha sido generado usando vue-cli, usando *ES6* y *Webpack*.

Aplicación de ejemplo. Incidencias.

La aplicación lanzará cada cierto tiempo una incidencia mediante un WebSocket y será recibida en tiempo real por el usuario de la web-
Cada incidencia se guarda en Base de datos con su estado.

Existirá un botón Resolver que pondrá la incidencia a estado *Resolved*.

Este botón lanzará una petición REST al servidor para que actualice el registro en Base de datos.

Aplicación de ejemplo: Incidencias.

Las herramientas que necesitaremos en caso de que queramos ejecutar la aplicación de ejemplo son:

1. JDK 8. Puede ser la de Oracle o bien OpenJDK
2. Maven. Un gestor de proyectos Java.
3. Spring CLI. Se trata de una utilidad en línea de comandos que nos permite hacer scafolding de diferentes tecnologías Spring. Lo usaremos en esta presentación.
4. Git
5. Usaremos Eclipse como IDE

Aplicación de ejemplo: Incidencias.

De este modo cubrimos:

- Organización de un proyecto Spring Boot
- Uso de los aspectos básicos de Spring Boot como Configuraciones, Controladores y vistas con **Thymeleaf**
- Uso de STOMP sobre WebSockets.
- Capa de datos mediante JPA y repositorios, usando base de datos H2
- Arquitectura de servicios.
- Controladores REST básicos.
- Biblioteca Quartz para generar eventos periódicos.

Aplicación de ejemplo: Incidencias.

No entraremos en profundidad sobre la instalación de estas herramientas, una de las formas más rápidas en Ubuntu es:

- Instalar una implementación de Java y la JDK (en este caso la OpenJdk 8)

```
sudo apt-get install default-jdk
```

- Instalar Maven

```
sudo apt-get install maven
```

- Instalar Spring CLI

Spring no está en los repositorios Ubuntu debido a incompatibilidades de licencia. Para poder instalarlo vamos a añadir primero un gestor de paquetes para la JVM, SDKMAN! (https://sdkman.io/)

```
curl -s "https://get.sdkman.io" | bash
```

Ahora podemos instalar springboot, el cual incluye Spring CLI

```
sdk install springboot
```

Aplicación de ejemplo: Incidencias.

En Windows la instalación es más incómoda, debido a la falta de gestores de paquetes oficiales. Podemos instalar todo a mano (descargando la JDK, maven y springboot)

También podemos usar Scoop, un package manager para Windows que está ganando popularidad. Scoop necesita tener un cliente git instalado.

- Instalamos Scoop. Para ello abrimos un Powershell y escribimos:

```
iex (new-object net.webclient).downloadstring('https://get.scoop.sh')
```

- Una vez instalado, añadir bucket Java

```
scoop bucket add java
```

- Instalar JDK

```
scoop install oraclejdk
```

- Instalar maven

```
scoop install maven
```

- Instalar springboot

```
scoop install springboot
```

Aplicación de ejemplo: Incidencias.

Generación de la aplicación

Clonar repositorio

```
git clone https://github.com/aleamb/peumconf2018-app.git
```

Generar frontend

```
npm install
npm run build
```

Compilar Backend

```
mvn package
```

Ejecutar:

```
java -jar .\target\peumconf2018-spring-boot-app-0.0.1-SNAPSHOT.jar
```

Abrir un navegador web y conectarse a http://localhost:8080

Aplicación de ejemplo: Incidencias.

Una aplicación Spring Boot puede ser creada de la siguiente forma:

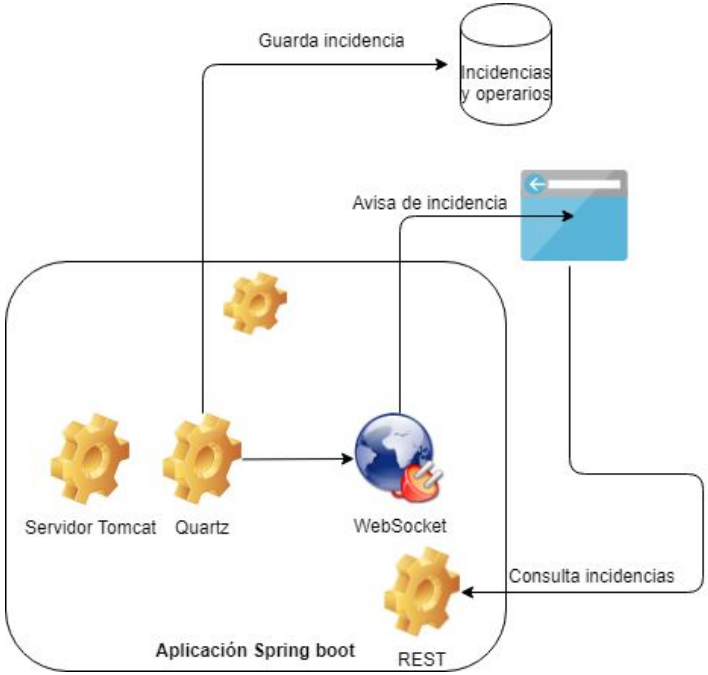
```
spring init -g=peumconf -a=spring-boot-app -name=spring-boot-app
```

Y arrancada con:

```
mvn spring-boot:run
```

Todas las funcionalidades que vayamos necesitando, se irán añadiendo a la aplicación en forma de dependencias.

Análisis de la Aplicación (viendo el código)



Gracias!

¿Preguntas?

Correo de contacto: jandroz@gmail.com