

2.1 Spring Tool Suite: nuestro entorno de trabajo

STS (Spring Tool Suite), es un entorno de desarrollo basado en Eclipse que está preparado para el desarrollo de aplicaciones Spring. Proporciona un entorno *ready-to-use* para implementar, depurar, ejecutar y desplegar aplicaciones Spring, incluyendo integraciones con Git, Maven, AspectJ, así como todas las funcionalidades de las últimas versiones de eclipse.

La última versión disponible la podemos encontrar en la propia web de spring (<https://spring.io/tools/sts>). Si pinchamos sobre el enlace, podemos acceder a la descarga del entorno, que viene empaquetado como un fichero zip. Dentro del mismo, podemos encontrar una carpeta con todo el software, y el ejecutable, llamado STS . exe.

Sin embargo, en los primeros compases del curso, trabajaremos con Spring al estilo Maven, lo que nos proporcionará algunas ventajas:

- Podremos trabajar con proyectos *legacy* (heredados), que estén configurados como proyectos Maven.
- Nos permitirá implementar proyectos sin el uso de la herramienta específica STS, y así poder utilizar cualquier otro IDE de uso general (el propio Eclipse, Netbeans, JDeveloper...).

2.2 Una palabra más sobre *Maven*

A día de hoy encontramos a muchos desarrolladores de Spring que utilizan Eclipse, y que comienzan los desarrollos de Spring como proyectos Maven. Maven nos permite gestionar las dependencias a través de un fichero de configuración, `pom.xml`. A continuación podemos encontrar las dependencias Maven más usuales (y para la versión actual) para el trabajo con Spring.

Dependencia	¿Para qué sirve?
<code><dependency><groupId>org.springframework</groupId><artifactId>spring-core</artifactId><version>4.3.2.RELEASE</version></dependency></code>	Esta es una de las dependencias más básicas. Si queremos construir un proyecto básico con Spring, que nos permita usar el <i>IoC container</i> y definir nuestros propios beans, esta es la dependencia que debemos usar.
<code><dependency><groupId>org.springframework</groupId><artifactId>spring-webmvc</artifactId><version>4.3.2.RELEASE</version></dependency></code>	Esta dependencia nos permitirá trabajar con todos los componentes para implementar aplicaciones web siguiendo el patrón Modelo-Vista-Controlador
<code><dependency><groupId>org.springframework</groupId><artifactId>spring-aop</artifactId><version>4.3.2.RELEASE</version></dependency></code>	Esta dependencia sirve para poder utilizar la programación orientada a aspectos (AOP).
<code><dependency><groupId>org.springframework</groupId><artifactId>spring-data-</code>	Esta dependencia, algo más específica, nos permitirá trabajar con el submódulo de Spring Data para JPA

Dependencia

¿Para qué sirve?

```
jpa</artifactId></dependency>
```

A lo largo del curso trabajaremos con algunas dependencias más, que no son propias de Spring, pero sí de otros elementos que necesitaremos, como la interfaz de usuario, el proveedor de ORM (*Object Relational Mapping*), la base de datos...:

Dependencia

¿Para qué sirve?

```
<dependency><groupId>com.h2database</groupId><artifactId>h2</artifactId></dependency>
```

Base de datos relacional *embebible* en un proyecto, implementada íntegramente en Java. Es muy útil para la fase de desarrollo o para sistemas que no necesitan un sistema gestor de base de datos profesional de alto rendimiento.

```
<dependency><groupId>org.hibernate</groupId><artifactId>hibernate-entitymanager</artifactId></dependency>
```

Hibernate es uno de los motores de persistencia más utilizado a día de hoy en proyectos software.

```
<dependency><groupId>log4j</groupId><artifactId>log4j</artifactId><version>1.2.14</version></dependency>
```

Log 4 Java es uno de los múltiples sistemas de *logging* disponibles para Spring, que nos permitirá configurar rápidamente este aspecto de nuestros proyectos.

Durante todas las lecciones, se proporcionarán las dependencias necesarias o el fichero `pom.xml` completo necesarios para poder realizar los ejemplos.