

Curso de Spring
Framework

Desarrollo con Maven y Spring

Micael Gallego

micael.gallego@gmail.com

@micael_gallego

Francisco Gortázar

patxi.gortazar@gmail.com

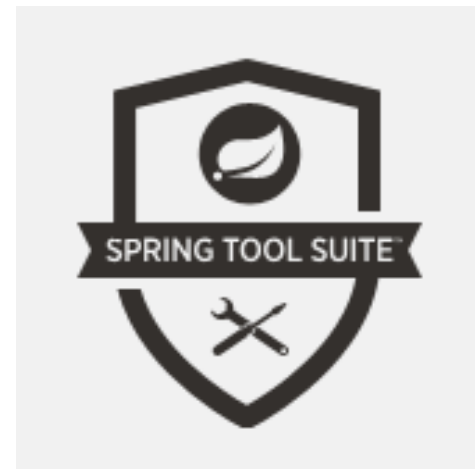
@fgortazar

- Sistema de **gestión de dependencias** (*librerías*) y sus versiones
- Sistema de **construcción de proyectos** (de código a entregable .zip)
- Estructura única de proyecto compatible con todos los **entornos de desarrollo** y sistemas de **integración continua**

- Maven gestiona muchos aspectos de la **construcción** de la aplicación
 - Descarga automática de dependencias
 - Compilación
 - Ejecución de tests
 - Publicación de la aplicación construida (binarios)
 - Documentación
 - ...

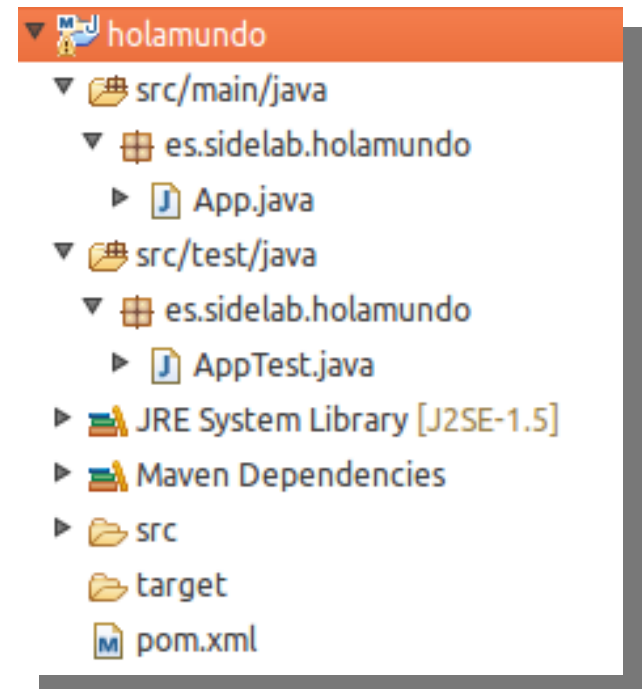
- **¿Cómo usar Maven?**
 - **Desde el entorno de desarrollo:** Maven está integrado en los IDEs Java más importantes (eclipse, netbeans, IntelliJ...)
 - **Desde la línea de comandos:** Sin necesidad de usar un IDE. Ideal para construcción de proyectos de forma automatizada

- **Spring Tool Suite (STS)**
 - En la asignatura usaremos una versión de **Eclipse** diseñada para desarrollo de aplicaciones **Spring**
 - <http://spring.io/tools>
 - Proporciona facilidades para trabajar con aplicaciones **Spring**
 - Está basado en la última versión de Eclipse

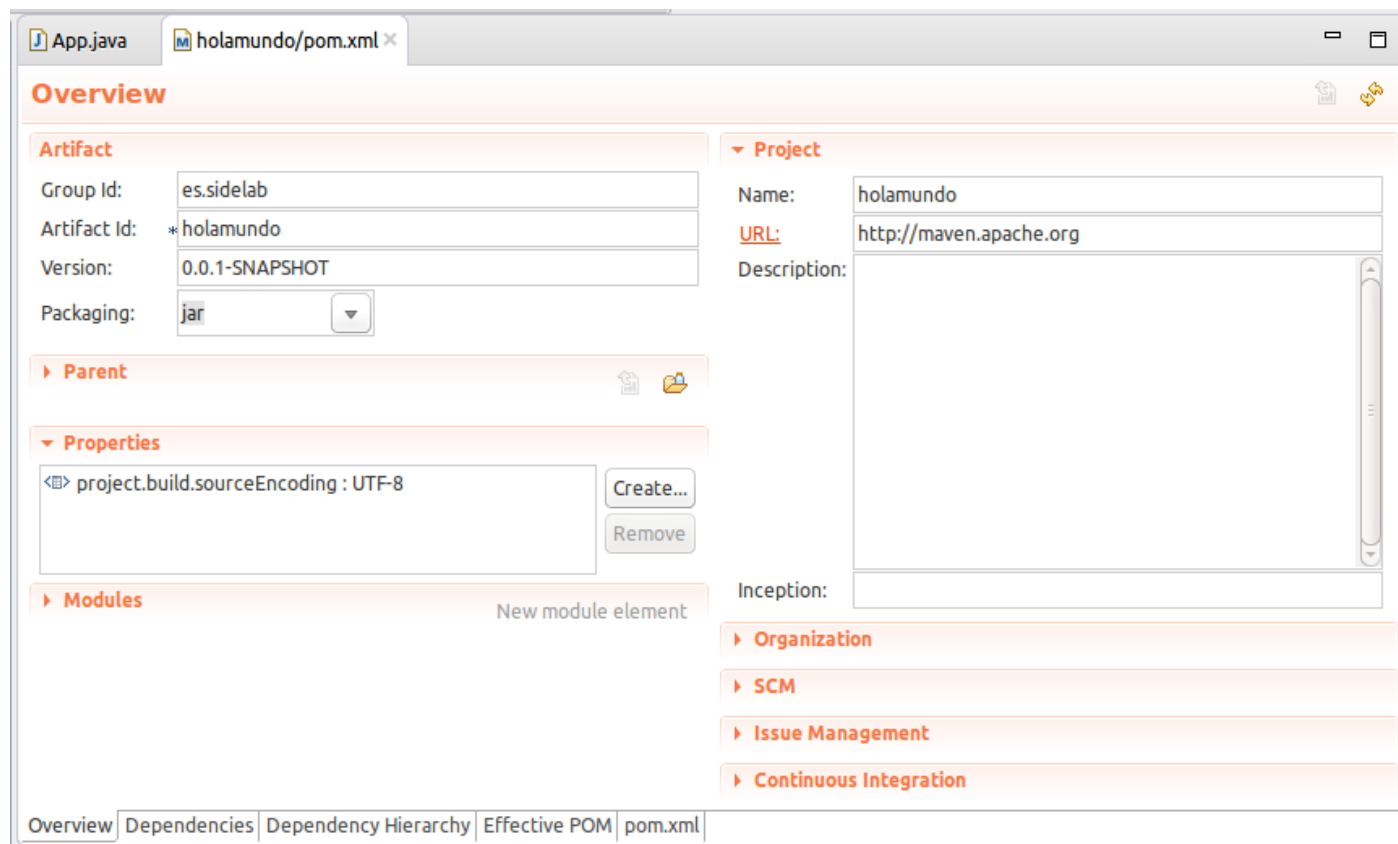


- **Cómo crear un proyecto Maven en Eclipse**
 - Nuevo proyecto > Maven > Maven Project
 - Dejar la plantilla que aparece por defecto seleccionada (**maven-archetype-quickstart**)
 - Indicar el nombre del proyecto:
 - GroupId: es.sidelab
 - ArtifactId: holamundo

- La mayoría de los proyectos Maven tienen la siguiente **estructura**
 - **src/main/java**: Código de la aplicación
 - **src/test/java**: Código de los tests
 - **pom.xml**: Fichero de descripción del proyecto (nombre, dependencias, configuraciones, etc...)



- **pom.xml**: Configuración del proyecto



- **pom.xml**: Configuración del proyecto



```
1 <project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
2   xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd"
3   <modelVersion>4.0.0</modelVersion>
4
5   <groupId>es.sidelab</groupId>
6   <artifactId>holamundo</artifactId>
7   <version>0.0.1-SNAPSHOT</version>
8   <packaging>jar</packaging>
9
10  <name>holamundo</name>
11  <url>http://maven.apache.org</url>
12
13  <properties>
14    <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
15  </properties>
16
17  <dependencies>
18    <dependency>
19      <groupId>junit</groupId>
20      <artifactId>junit</artifactId>
21      <version>3.8.1</version>
22      <scope>test</scope>
23    </dependency>
24  </dependencies>
25 </project>
```

Poner la vista de código fuente

- **pom.xml:** Configuración del proyecto
 - **groupId:** Organización, familia
 - **artifactId:** Nombre del proyecto
 - **version:** Versión del proyecto (especialmente útil para librerías)
 - **packaging:** Tipo de aplicación (jar es una aplicación normal)
 - **name:** Nombre “bonito” del proyecto (para documentación)
 - **url:** Web del proyecto (para documentación)

```
<groupId>es.sidelab</groupId>  
<artifactId>holamundo</artifactId>  
<version>0.0.1-SNAPSHOT</version>  
<packaging>jar</packaging>
```

```
<name>holamundo</name>  
<url>http://maven.apache.org</url>
```

- **pom.xml:** Configuración del proyecto
 - **properties:**
 - ▮ Configuraciones generales del proyecto
 - ▮ Versión de Java (Si no se pone nada es la 1.5)
 - ▮ Codificación de los ficheros fuente

```
<properties>
  <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
  <maven.compiler.source>1.8</maven.compiler.source>
  <maven.compiler.target>1.8</maven.compiler.target>
</properties>
```

- **pom.xml**: Configuración del proyecto
 - **dependencies**:
 - ▮ Dependencias (**librerías**)
 - ▮ Cada librería está identificada por su **groupId**, **artifactId** y **versión** (coordenadas)
 - ▮ Se pueden poner tantas dependencias como se quiera

```
<dependencies>
  <dependency>
    <groupId>junit</groupId>
    <artifactId>junit</artifactId>
    <version>3.8.1</version>
    <scope>test</scope>
  </dependency>
</dependencies>
```

- **pom.xml:** Configuración del proyecto
 - **Cuidado!** Algunos cambios en el fichero pom.xml no se reflejan en eclipse de forma automática
 - Cuando se hace un cambio y eclipse no se actualiza con esos cambios, se tiene que indicar explícitamente
 - ▮ Botón derecho proyecto > Maven > Update Project...

- **Construir un proyecto Maven**
 - **De forma automática:**
 - ▮ Eclipse se descarga las librerías y compila el código.
 - ▮ Eclipse no empaqueta la aplicación en un .jar
 - **De forma manual:**
 - ▮ Se pueden ejecutar operaciones Maven desde eclipse o desde la línea de comandos (si está maven instalado en el sistema)

- **Tareas básicas de Maven**

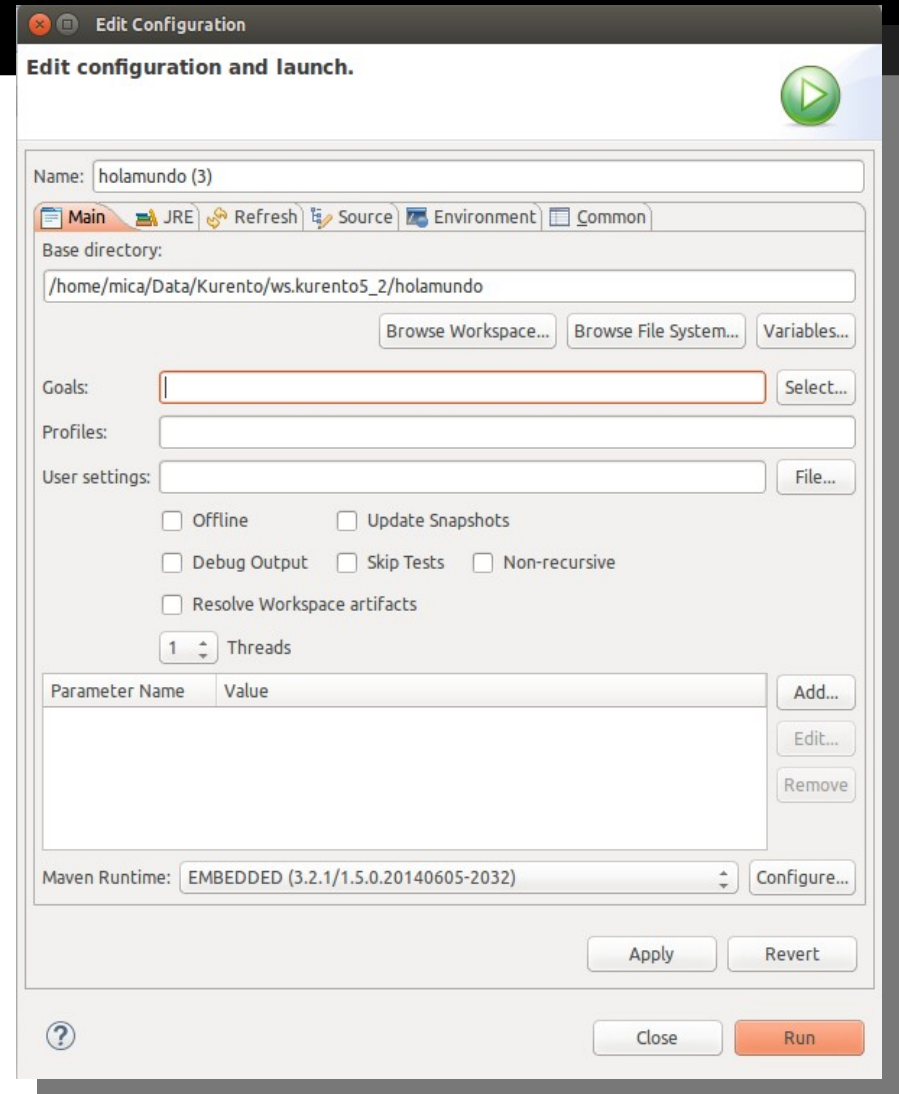
- **compile:** Compila el código
- **test:** Ejecuta los tests del proyecto
- **package:** Empaqueta el proyecto, habitualmente generando un fichero .jar
- **install:** Instala el paquete generado para que esté accesible a otros proyectos de la misma máquina
- **deploy:** Publica el paquete generado para que esté accesible para otros desarrolladores del equipo o públicamente en Internet
- **clean:** Limpia la carpeta de los ficheros generados (binarios, zips, etc...)

- Las tareas tienen **dependencias** entre ellas. Por ejemplo, al ejecutar **install**, se ejecuta antes automáticamente **compile**, **test** y **package**.

- La tarea **clean** es **independiente** de las demás

DESARROLLO WEB CON SPRING Maven

- Ejecutar tareas Maven desde eclipse
 - Botón derecho > Run As
 - **mvn clean**
 - **mvn install**
 - **mvn build...:** Permite crear una tarea nueva configurable
 - **mvn build:** Permite seleccionar una tarea previamente ejecutada



- **Repositorio de dependencias (librerías)**
 - Cuando Maven **construye un proyecto**, las dependencias se cargan desde un repositorio
 - **Repositorio en la red**
 - ▮ Existen varios repositorios públicos en Internet (**Maven Central Repository**)
 - ▮ Es habitual que las empresas tengan un **repositorio privado** para toda la organización en su red local
 - **Repositorio en la máquina de desarrollo ~/.m2/repository**
 - ▮ Guarda las librerías que se han **descargado** para que estén disponibles para construir de nuevo el proyecto o para **otros proyectos**
 - ▮ Guarda las librerías que **construyen** para que estén disponibles para otros proyectos