



Certified



Corporation®

DESARROLLO DE APLICACIONES JEE CON SPRING FRAMEWORK

Dirección Nacional de Educación Continua

13-11-2021 Módulo 1



INTRODUCCIÓN

- La presente tiene como objetivo explicar las capacidades de un entorno de trabajo, sus funciones y capacidades, además explicar en detalle algunos de los módulos del framework de spring, con la intención de preparar al lector para generar soluciones empresariales simples y efectivas, conociendo la arquitectura y con esto ser un aporte al momento de diseñar una aplicación, aplicando los conocimientos de este documento.



¿QUÉ ES UN GESTOR DE PROYECTOS (IDE)?

Un entorno de desarrollo integrado (IDE) es un sistema de software para el diseño de aplicaciones que combina herramientas comunes para desarrolladores en una sola interfaz de usuario gráfica (GUI), la sigla en inglés Integrated Development Environment





SPRING
Framework

SPRING FRAMEWORK

- Spring Framework proporciona un modelo integral de programación y configuración para aplicaciones empresariales modernas basadas en Java, en cualquier tipo de plataforma de implementación.



SPRING
Framework

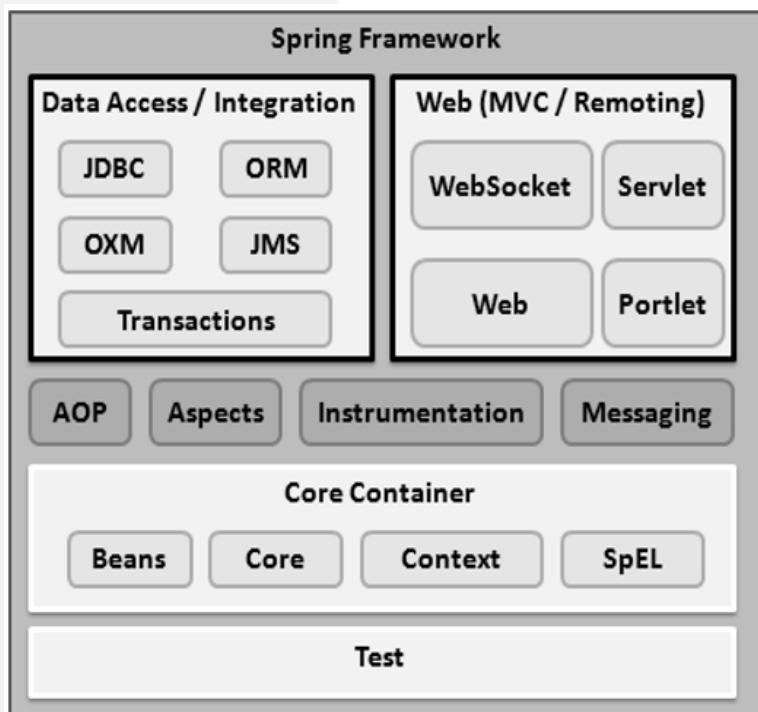
¿POR QUÉ SPRING FRAMEWORK?

- Spring es el marco de desarrollo de aplicaciones más popular para Java empresarial. Millones de desarrolladores de todo el mundo utilizan Spring Framework para crear código de alto rendimiento, fácilmente comprobable y reutilizable.
- Spring es ligero en cuanto a tamaño y transparencia.
- Las características principales de Spring Framework se pueden utilizar para desarrollar cualquier aplicación Java, pero existen extensiones para crear aplicaciones web sobre la plataforma Java EE.
- Spring Framework tiene como objetivo facilitar el uso del desarrollo J2EE(JEE since 2017) y promueve las buenas prácticas de programación al permitir un modelo de programación basado en POJO.



SPRING
Framework

ARQUITECTURA SPRING FRAMEWORK



- Spring es un framework modular que cuenta con una arquitectura dividida en capas o módulos, como se muestra en la Figura, lo cual permite tomar y ocupar únicamente las partes que interesen para el proyecto y juntarlas con gran libertad.

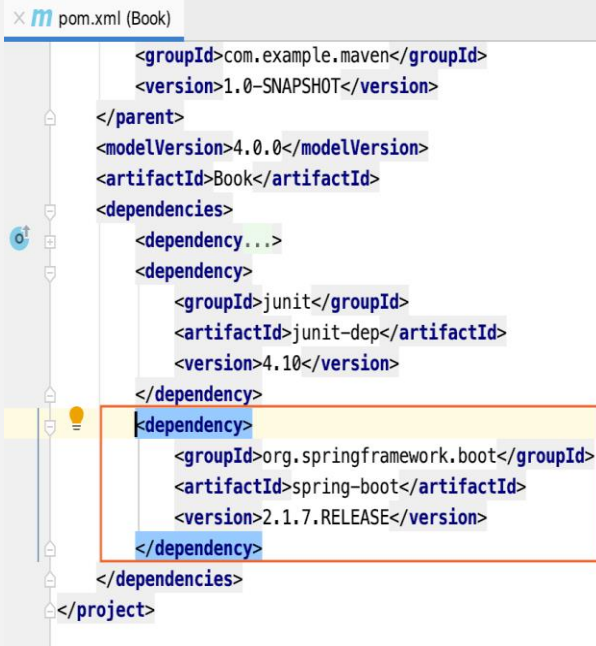
MAVEN

- **Maven2** es una herramienta open-source, que se creó en 2001 con el objetivo de simplificar los procesos de build (compilar y generar ejecutables a partir del código fuente). Con el fin de optimizar los tiempos de construcción asociados a los proyectos y evitar analizar qué partes de código se deben compilar, qué librerías utiliza el código, dónde incluirlas, y qué dependencias de compilación había en el proyecto.



SPRING
Framework

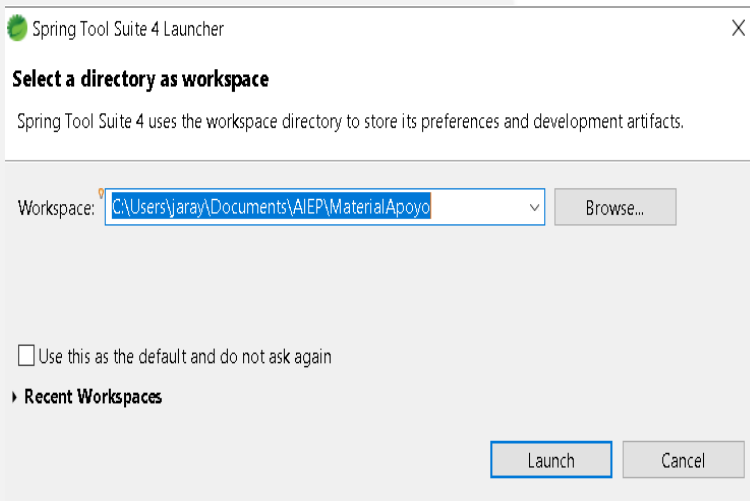
MODELO OBJETO PROYECTO POM.XML



- El modelo de objetos del proyecto (POM) archivo en formato XML (lenguaje de marcado extensible). La función es similar al archivo build.xml de ant, con funciones más potentes. Este archivo se utiliza para administrar: código fuente, archivos de configuración, información y roles del desarrollador, sistema de seguimiento de problemas, información de la organización, autorización del proyecto, URL del proyecto, dependencias del proyecto



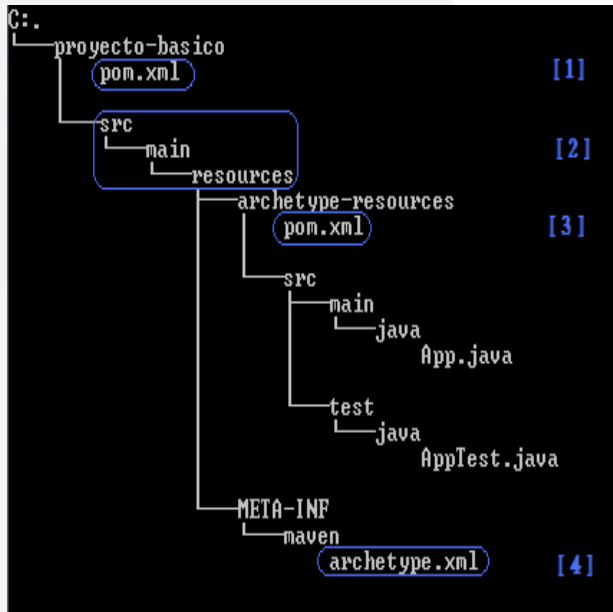
CREACIÓN DE PROYECTOS



- Para la creación de proyectos utilizaremos el IDE Spring Tool Suite4 (STS), para lo cual se debe generar un Workspace.
- Luego debemos elegir un arquetipo Maven para la generación de los proyectos.

¿QUÉ ES UN ARQUETIPO?

- Los arquetipos Maven, puede decirse que son plantillas, parametrizadas o configuradas para utilizar determinadas tecnologías que los desarrolladores utilizan como base para escribir y organizar el código de la aplicación.



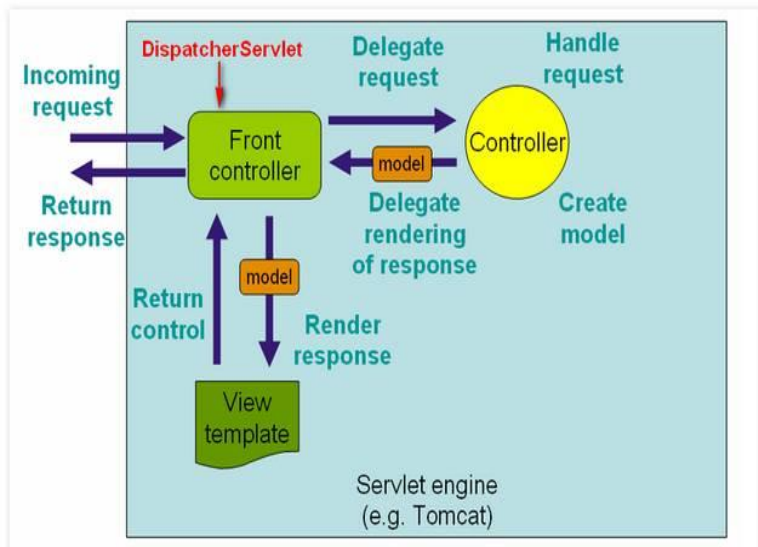
SPRING
Framework



ARQUETIPO JAVA

- El IDE tiene la capacidad de poder importar o crear un proyecto directamente desde un arquetipo Maven, algunos datos que se solicitan en esta creación es.
- **Group ID**, corresponde al sistema base de package
- **Paquetización**, debe guardar relación con lo que hace el proyecto.
- **ArtifactID**, será el nombre del compilado (jar,war,ear) debe guardar relación con lo que hace el proyecto y si obedece a un patrón por ejemplo: AIEP-DAO-Clientes.
- **Versión**, es la versión base de creación del compilado.
- **Package**: es la estructura del proyecto a esta altura solo se define el base, aunque se puede editar, lo recomendado es dejarlo idéntico al group-id.



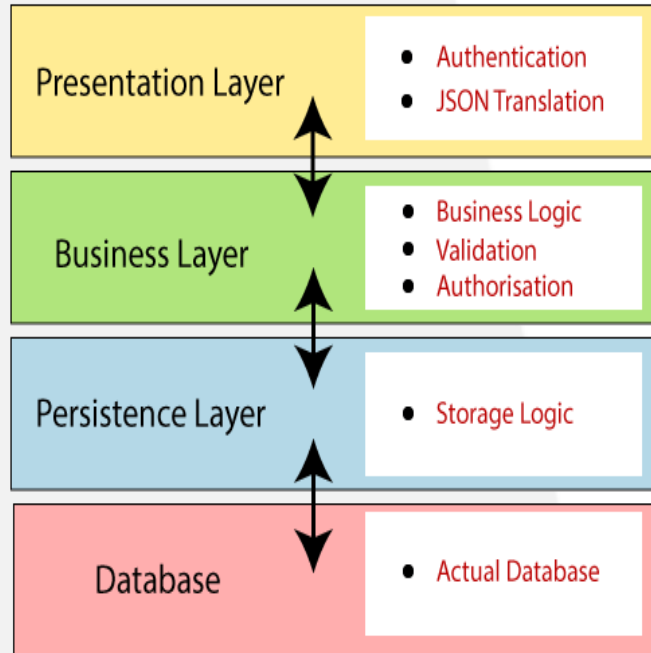


SPRING MVC

- En la aplicación web Spring MVC, consta de 3 componentes estándar MVC (Modelo, Vistas, Controlador):
1. Modelos: objetos de dominio que son procesados por la capa de servicio (lógica empresarial) o la capa persistente (operación de la base de datos).
 2. Vistas: muestra datos, normalmente es una página JSP escrita con la biblioteca de etiquetas estándar de Java (JSTL).
 3. Controladores: mapeo de URL e interactúa con la capa de servicio para el procesamiento comercial y devuelve un modelo.



SPRING
Framework



SPRING BOOT

- *Spring Boot es un módulo de Spring Framework. Se utiliza para crear aplicaciones independientes basadas en Spring de grado de producción con un esfuerzo mínimo.*
- *Spring de grado de producción autoconfigurable. Con él, los desarrolladores pueden comenzar rápidamente sin perder tiempo preparando y configurando su aplicación Spring.*
- *Spring Boot está construido sobre el Framework Spring y viene con muchas dependencias que se pueden conectar a la aplicación Spring. Algunos ejemplos son Spring Kafka, Spring LDAP, Spring Web Services y Spring Security.*

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

DEPENDENCIAS MAVEN

- *Uno de los puntos fuertes de maven son las dependencias. En nuestro proyecto podemos decirle a maven que necesitamos un jar (por ejemplo, log4j o el conector de MySQL) y maven es capaz de ir a internet, buscar esos jar y bajárselos automáticamente. Es más, si alguno de esos jar necesitara otros jar para funcionar, maven "tira del hilo" y va bajándose todos los jar que sean necesarios*



SPRING
Framework

REPOSITORIO LOCAL Y REMOTO

- *Un repositorio en Maven contiene artefactos de compilación y dependencias de diversos tipos:*
- *El repositorio local es un directorio en la computadora donde se ejecuta Maven. Almacena en caché las descargas remotas y contiene artefactos de compilación temporales que aún no ha publicado.*
- *Los repositorios remotos se refieren a cualquier otro tipo de repositorio, al que se accede mediante una variedad de protocolos como `file://` y `https://`. Estos repositorios pueden ser un repositorio verdaderamente remoto configurado por un tercero para proporcionar sus artefactos para su descarga (por ejemplo, repo.maven.apache.org). Otros repositorios "remotos" pueden ser repositorios internos configurados en un archivo o servidor HTTP dentro de su empresa, usados para compartir artefactos privados entre equipos de desarrollo y para lanzamientos.*

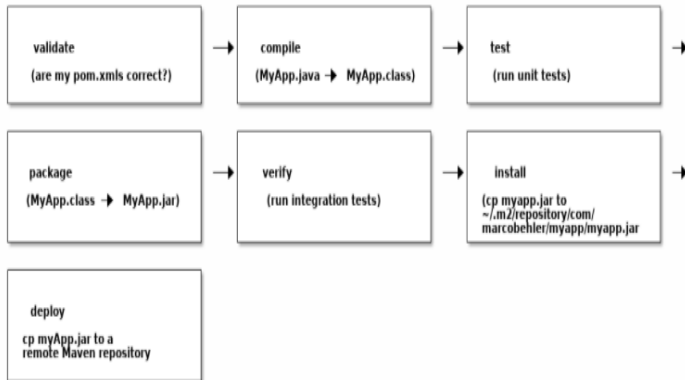


CICLO VIDA COMPILACIÓN

- En Maven se definen tres ciclos de build del software con una serie de etapas diferenciadas. Por ejemplo, el ciclo de vida de la compilación tiene las etapas de:
 - **Validación (validate):** Validar que el proyecto es correcto.
 - **Compilación (compile).**
 - **Test (test):** Probar el código fuente usando un framework de pruebas unitarias.
 - **Empaquetar (package):** Empaquetar el código compilado y transformarlo en algún formato tipo .jar o .war.
 - **Pruebas de integración (integration-test):** Procesar y desplegar el código en algún entorno donde se puedan ejecutar las pruebas de integración.
 - **Verificar** que el código empaquetado es válido y cumple los criterios de calidad (verify).
 - **Instalar** el código empaquetado en el repositorio local de Maven, para usarlo como dependencia de otros proyectos (install).
 - **Desplegar** el código a un entorno (deploy).



COMPILACIÓN



- Para la compilación utilizaremos el entorno de desarrollo, se debe ejecutar el comando “mvn clean install”
- El comando **clean** realiza una limpieza de las librerías previamente instaladas en el directorio .M2
- El comando **install**, vuelve a compilar y ejecuta el ciclo de vida nuevamente.



EJECUCIÓN DE PRUEBAS

- Para la ejecución de las pruebas se debe incluir en nuestro POM las librerías necesarias para la ejecución nuestras clases tipo test.
- Con el comando “package o install ” Maven ejecutara los test cases declarados en la carpeta src/test/java/ de la imagen indicada.



SPRING
Framework

LIMPIEZA

- El comando **Clean**, genera la eliminación del artefacto en el repositorio local y elimina las clases previamente compiladas en la carpeta .M2

uipo > Disco local (C:) > Usuarios > .m2 > repository

Nombre	Fecha de modificación	Tipo	Tamaño
antir	13-01-2021 12:31	Carpeta de archivos	
aopalliance	13-01-2021 12:33	Carpeta de archivos	
asm	13-01-2021 12:33	Carpeta de archivos	
avalon-framework	25-01-2021 9:55	Carpeta de archivos	
backport-util-concurrent	13-01-2021 12:28	Carpeta de archivos	
ch	13-01-2021 12:28	Carpeta de archivos	
cl	09-11-2021 14:21	Carpeta de archivos	
classworlds	14-01-2021 16:47	Carpeta de archivos	
com	13-07-2021 16:24	Carpeta de archivos	
commons-beanutils	19-01-2021 17:29	Carpeta de archivos	
commons-chain	25-01-2021 9:55	Carpeta de archivos	
commons-cli	19-01-2021 13:21	Carpeta de archivos	
commons-codec	14-01-2021 16:47	Carpeta de archivos	
commons-collections	19-01-2021 17:29	Carpeta de archivos	
commons-configuration	25-01-2021 9:59	Carpeta de archivos	
commons-digester	19-01-2021 17:29	Carpeta de archivos	
commons-fileupload	03-05-2021 12:48	Carpeta de archivos	
commons-httpclient	25-01-2021 9:55	Carpeta de archivos	
commons-io	13-01-2021 12:27	Carpeta de archivos	
commons-jxpath	25-01-2021 9:59	Carpeta de archivos	
commons-lang	19-01-2021 13:22	Carpeta de archivos	



equipo > Disco local (C:) > Usuarios > .m2 > repository

Nombre	Fecha de modificación	Tipo	Tamaño
antir	13-01-2021 12:31	Carpeta de archivos	
aopalliance	13-01-2021 12:33	Carpeta de archivos	
asm	13-01-2021 12:33	Carpeta de archivos	
avalon-framework	25-01-2021 9:55	Carpeta de archivos	
backport-util-concurrent	13-01-2021 12:28	Carpeta de archivos	
ch	13-01-2021 12:28	Carpeta de archivos	
classworlds	14-01-2021 16:47	Carpeta de archivos	
com	13-07-2021 16:24	Carpeta de archivos	
commons-beanutils	19-01-2021 17:29	Carpeta de archivos	
commons-chain	25-01-2021 9:55	Carpeta de archivos	
commons-cli	19-01-2021 13:21	Carpeta de archivos	
commons-codec	14-01-2021 16:47	Carpeta de archivos	
commons-collections	19-01-2021 17:29	Carpeta de archivos	
commons-configuration	25-01-2021 9:59	Carpeta de archivos	
commons-digester	19-01-2021 17:29	Carpeta de archivos	
commons-fileupload	03-05-2021 12:48	Carpeta de archivos	
commons-httpclient	25-01-2021 9:55	Carpeta de archivos	
commons-io	13-01-2021 12:27	Carpeta de archivos	
commons-jxpath	25-01-2021 9:59	Carpeta de archivos	
commons-lang	19-01-2021 13:22	Carpeta de archivos	
commons-logging	19-01-2021 13:21	Carpeta de archivos	
commons-validator	19-01-2021 17:29	Carpeta de archivos	
de	19-01-2021 17:28	Carpeta de archivos	





```
[INFO] -----> c:\dep\ejemp\java\AEP-Ejemplos <-----
[INFO] Building AEP-Ejemplos 0.0.1-SNAPSHOT
[INFO] [ jar ]
[INFO]
[INFO] --- maven-clean-plugin:3.1.0:clean (default-clean) @ AEP-Ejemplos ---
[INFO] Deleting C:\Users\jony\Documents\AIEP\MaterialAep\AEP-Ejemplos\target
[INFO]
[INFO] --- maven-resources-plugin:3.1.0:resources (default-resources) @ AEP-Ejemplos ---
[INFO] Using 'UTF-8' encoding to copy filtered resources.
[INFO] skip non existing resourceDirectory C:\Users\jony\Documents\AIEP\MaterialAep\AEP-Ejemplos\src\main\resources
[INFO]
[INFO] --- maven-compiler-plugin:3.8.0:compile (default-compile) @ AEP-Ejemplos ---
[INFO] Changes detected - recompiling the module!
[INFO] Compiling 1 source file to C:\Users\jony\Documents\AIEP\MaterialAep\AEP-Ejemplos\target\classes
[INFO]
[INFO] --- maven-resources-plugin:3.1.0:testResources (default-testResources) @ AEP-Ejemplos ---
[INFO] Using 'UTF-8' encoding to copy filtered resources.
[INFO] skip non existing resourceDirectory C:\Users\jony\Documents\AIEP\MaterialAep\AEP-Ejemplos\src\test\resources
[INFO]
[INFO] --- maven-compiler-plugin:3.8.0:testCompile (default-testCompile) @ AEP-Ejemplos ---
[INFO] Changes detected - recompiling the module!
[INFO] Compiling 1 source file to C:\Users\jony\Documents\AIEP\MaterialAep\AEP-Ejemplos\target\test-classes
[INFO]
[INFO] --- maven-surefire-plugin:2.22.1:test (default-test) @ AEP-Ejemplos ---
[INFO] Downloading from : https://repo.maven.apache.org/maven2/org/apache/maven/surefire/surefire-junit4-2.22.1.jar
[INFO] Downloaded from : https://repo.maven.apache.org/maven2/org/apache/maven/surefire/surefire-junit4-2.22.1.jar (3.1 kB at 3.8 MB/s)
[INFO] Downloading from : https://repo.maven.apache.org/maven2/org/apache/maven/surefire/surefire-providers-2.22.1.jar
[INFO] Downloaded from : https://repo.maven.apache.org/maven2/org/apache/maven/surefire/surefire-providers-2.22.1.jar (2.5 kB at 11 MB/s)
[INFO] Downloading from : https://repo.maven.apache.org/maven2/org/apache/maven/surefire/surefire-junit4-2.22.1.jar
[INFO] Downloaded from : https://repo.maven.apache.org/maven2/org/apache/maven/surefire/surefire-junit4-2.22.1.jar (55 kB at 228 MB/s)
[INFO]
[INFO] -----> T E S T S <-----
[INFO]
[INFO] Running c:\dep\ejemp\java\AppTest
[INFO] Tests run: 1, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.003 s - in c:\dep\ejemp\java\AppTest
[INFO]
[INFO] Results:
[INFO]
[INFO] Tests run: 1, Failures: 0, Errors: 0, Skipped: 0
[INFO]
[INFO]
[INFO] --- maven-jar-plugin:3.1.0:jar (default-jar) @ AEP-Ejemplos ---
[INFO] Building jar: C:\Users\jony\Documents\AIEP\MaterialAep\AEP-Ejemplos\target\AEP-Ejemplos-0.0.1-SNAPSHOT.jar
[INFO]
[INFO] --- maven-install-plugin:3.1.0:install (default-install) @ AEP-Ejemplos ---
[INFO] Installing C:\Users\jony\Documents\AIEP\MaterialAep\AEP-Ejemplos\target\AEP-Ejemplos-0.0.1-SNAPSHOT.jar to C:\Users\jony\localrepository\jony\AEP-Ejemplos\0.0.1-SNAPSHOT\AEP-Ejemplos-0.0.1-SNAPSHOT.jar
[INFO] Installing C:\Users\jony\Documents\AIEP\MaterialAep\AEP-Ejemplos\pom.xml to C:\Users\jony\localrepository\jony\AEP-Ejemplos\0.0.1-SNAPSHOT\AEP-Ejemplos-0.0.1-SNAPSHOT.pom
[INFO]
[INFO] -----> BUILD SUCCESS <-----
[INFO]
[INFO] Total time: 3.807 s
[INFO] Finished at: 2022-11-07T04:21:39-05:00
[INFO] ----->
```

EJECUCIÓN

- Después de que la compilación de Maven sea exitosa, se puede ejecutar el jar directamente desde la carpeta.M2 o desde la carpeta \target\ classes con el siguiente comando java.



EMPAQUETAMIENTO

```
<artifactId>spring-jdbc</artifactId>  
<packaging>jar</packaging>  
<name>Spring Boot JDBC</name>  
<version>1.0</version>
```

- El comando **package**, genera el tipo de empaquetamiento indicado en el POM ya sea un Jar o War o Ear.



SPRING
Framework

DESCARGA DE GESTOR DE PROYECTOS

- Para descargar el entorno de desarrollo (IDE)
- Acceder a <https://spring.io/tools>
- Seleccionar la versión de STS acorde a su sistema operativo.
- Esperar la descarga.

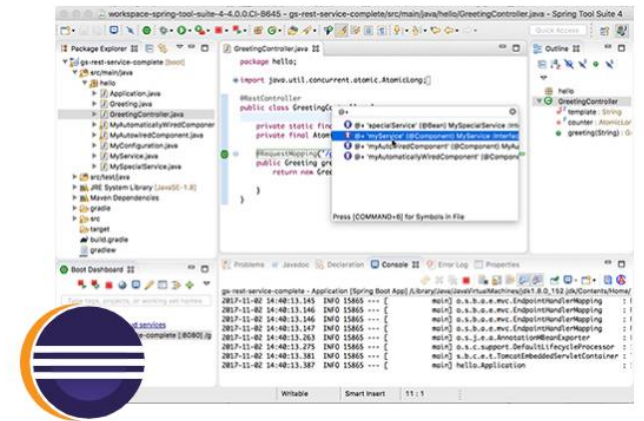
Spring Tools 4 for Eclipse

The all-new Spring Tool Suite 4.
Free. Open source.

4.12.1 - LINUX X86_64

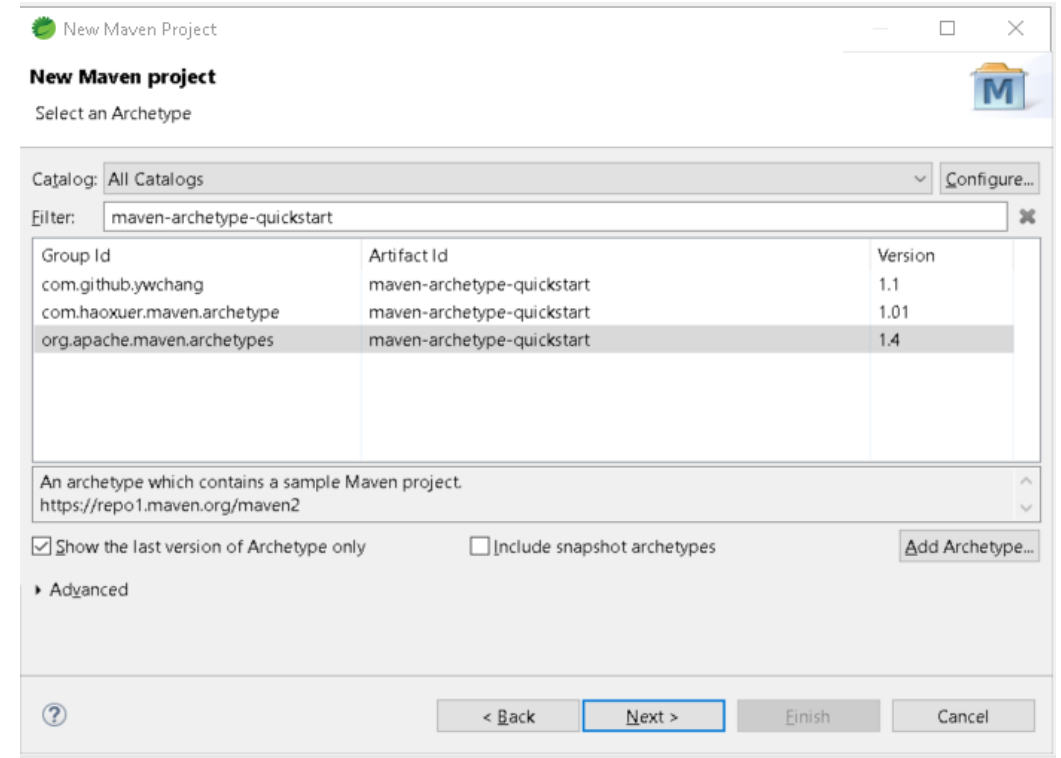
4.12.1 - MACOS X86_64

4.12.1 - WINDOWS X86_64



DESCARGA DE ARQUETIPO JAVA

- Profesor presenta paso a paso para descargar arquetipo
- Menú STS para creación proyecto Maven.
- Nombre Arquetipo. maven-archetype-quickstart
- Compilación y ejecución arquetipo
- Indica Carpeta .M2



PRUEBA PRÁCTICA NRO1

- El alumno deberá instalar la herramienta STS
- Luego crear un Worspace en STS
- Descargar el arquetipo Maven java estándar
- Crear una clase java que obtenga información desde un fichero (txt)
- Crear una clase TEST que valide el cumplimiento de la clase previamente realizada.
- Incluir las librerías necesarias en el POM.xml para ejecutar las pruebas
- Mostrar la consola con los principales paso a paso del ciclo de vida Maven, Clean, validate, compile, install.

PRUEBA PRÁCTICA NRO2

- El alumno debe generar un nuevo proyecto java spring boot,
- Se deben agregar las librerías asociadas a Spring Boot
- Crear una clase java que almacene en variable un listado obtenido desde fichero (txt)
- Luego recorrer dicha variable con toda la data obtenida.
- Crear una clase TEST que valide el cumplimiento de la clase previamente realizada.
- Incluir las librerías necesarias en el POM.xml para ejecutar las pruebas
- Levantar el proyecto SpringBoot desplegando el paso a paso.



MUCHAS GRACIAS

Certified



Corporation®

