

Gestión de proyectos con Maven

Víctor Orozco - @tuxtor

8 de octubre de 2020

Academik



ACADEMIK

¿Como programar en Java?

1. Crear un proyecto (Objetivo, framework y SCM)
2. Programar el proyecto
3. Ejecutar el proyecto*

¿Como programar en Java?

1. **Crear un proyecto (Objetivo, framework y SCM)**
2. Programar el proyecto
3. Ejecutar el proyecto*

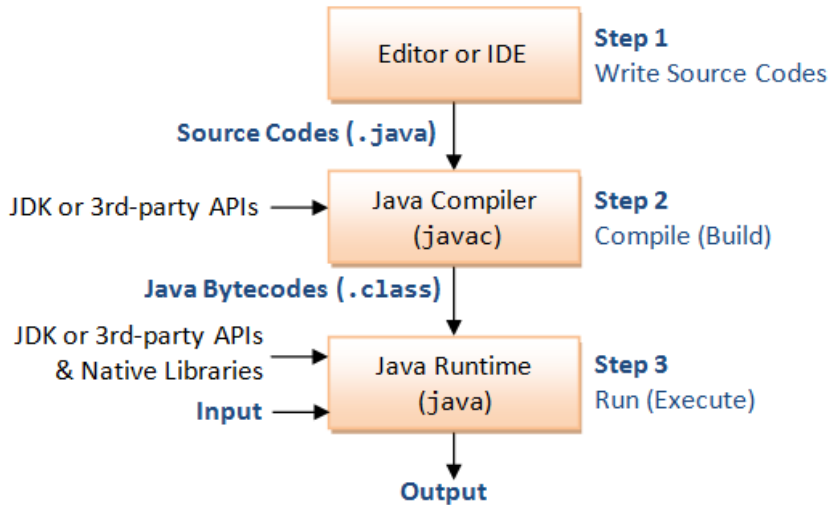


ACADEMIK

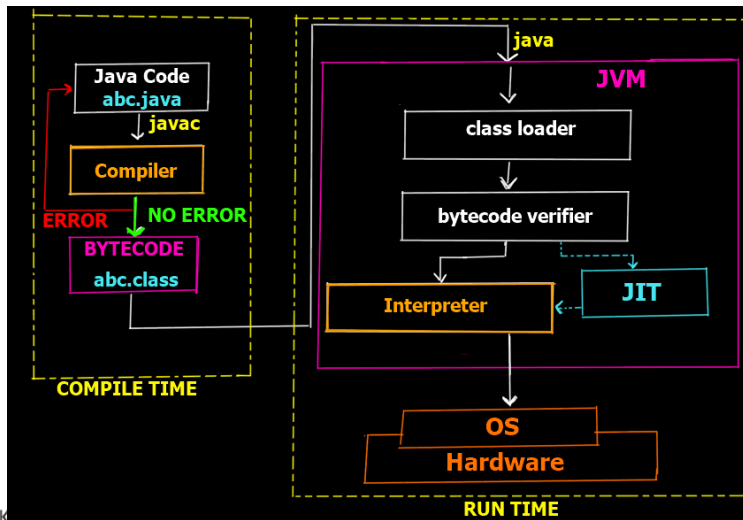
Proyectos en Maven



Compilador



Compilador



- ¿Dependencias?
- ¿Proceso de ensamblaje?
- ¿Proyecto final? ¿.jar, .war, .ear

- ¿Dependencias?
- ¿Proceso de ensamblaje?
- ¿Proyecto final? ¿.jar, .war, .ear

Task runner + Gestor de dependencias

- Project Object Model (Declarativo)
- Construcción, reportes, documentación
- Build tool
- Arquetipos
- Ensamblado manual
- Bill of materials (BOM)
- Starter
- .jar, .war, .ear

- groupId: Identifica y da pertenencia al paquete/biblioteca/proyecto hacia una empresa/organización
- artifactId: Identificador primario del artefacto
- version: Versión del paquete

```
mvn archetype:generate  
  -DgroupId=com.nabenik -DartifactId=demo1
```

Maven - Arquetipo

```
1. java
e a single project with junit, Mockito and slf4j dependencies.)
2222: remote -> uk.ac.ebi.gxa:atlas-archetype (Archetype for generating a custom
Atlas webapp)
2223: remote -> uk.ac.gate:gate-plugin-archetype (Maven archetype to create a ne
w GATE plugin project.)
2224: remote -> uk.ac.gate:gate-pr-archetype (Maven archetype to create a new GA
TE plugin project including a sample PR class (an empty LanguageAnalyser).)
2225: remote -> uk.ac.nactem.argo:argo-analysis-engine-archetype (An archetype w
hich contains a sample Argo (UIMA) Analysis Engine)
2226: remote -> uk.ac.nactem.argo:argo-reader-archetype (An archetype which cont
ains a sample Argo (UIMA) Reader)
2227: remote -> uk.ac.rdg.resc:edal-ncwms-based-webapp (-)
2228: remote -> uk.co.nemstix:basic-javaee7-archetype (A basic Java EE7 Maven ar
chetype)
2229: remote -> uk.co.solong:angular-spring-archetype (So Long archetype for RES
Tful spring services with an AngularJS frontend. Includes debian deployment)
2230: remote -> us.fatehi:schemacrawler-archetype-maven-project (-)
2231: remote -> us.fatehi:schemacrawler-archetype-plugin-command (-)
```



```
mvn archetype:generate -DgroupId=com.nabenik  
-DarchetypeArtifactId=maven-archetype-quickstart  
-DartifactId=demo1
```

Maven - Arquetipo

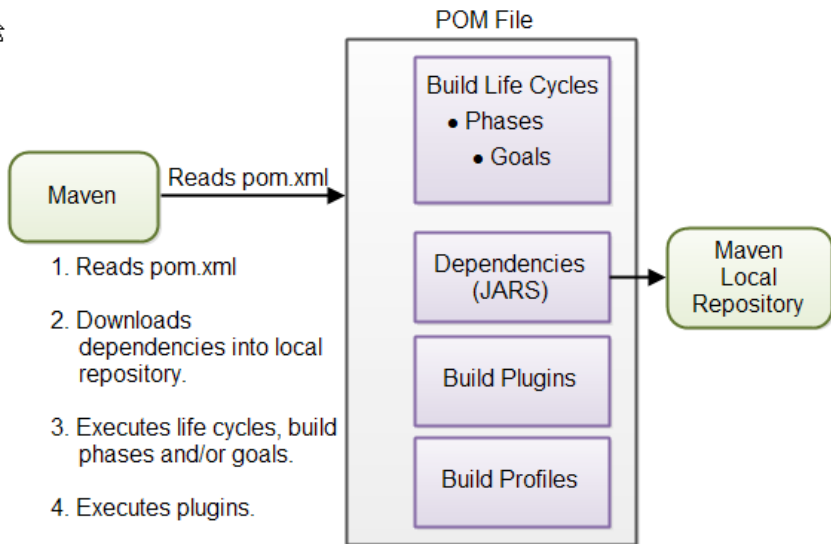
```
1. bash
millenium-falcon-2:demo1 tuxtor$ tree
.
├─ pom.xml
└─ src
    ├─ main
    │   └─ java
    │       └─ com
    │           └─ nabenik
    │               └─ App.java
    └─ test
        └─ java
            └─ com
                └─ nabenik
                    └─ AppTest.java

9 directories, 3 files
millenium-falcon-2:demo1 tuxtor$
```

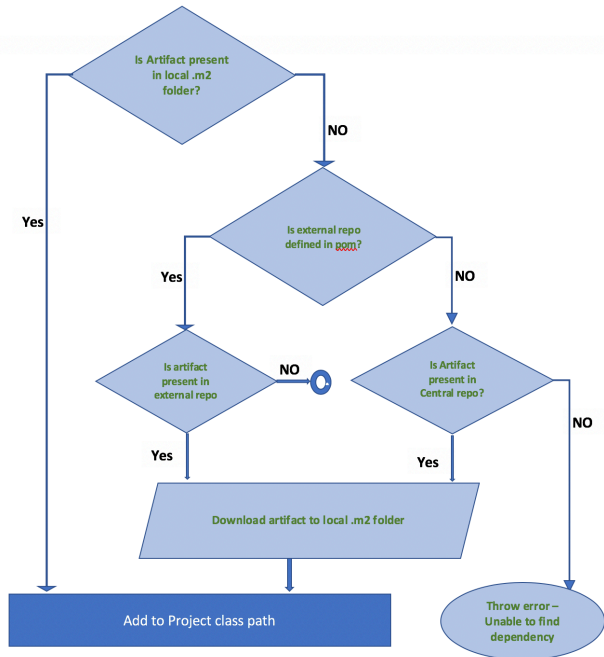


Estructura de carpetas

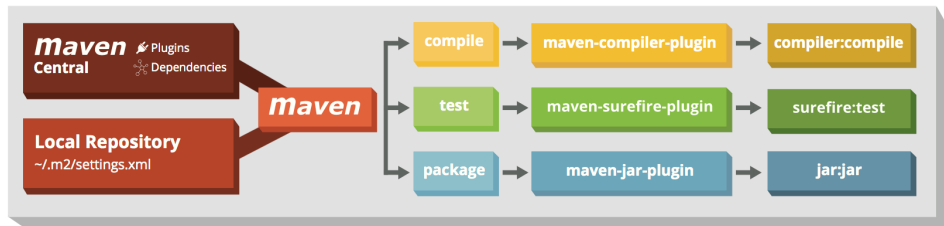
- `src/main/java` - Código fuente
- `src/main/resources` - Recursos del proyecto (se copian de forma integra)
- `src/main/webapp` - Recursos web y/o basados en Servlet
- `src/test/*` - Entorno de pruebas
- `target` - Archivos generados por Maven



All executed according to selected build profile.



Maven

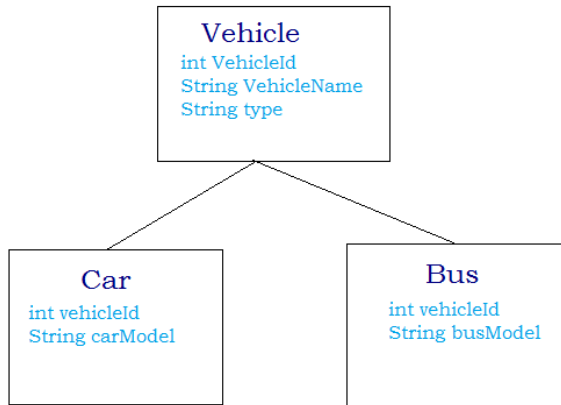


- `clean` - Elimina el directorio `target`
- `validate` - Valida si el proyecto es correcto
- `compile` - Compila los archivos fuente y almacena el resultado en `target/classes`
- `test` - Ejecuta tests
- `package` - Toma los objetos compilados y los empaca en formatos distribuibles e.g. JAR, WAR
- `verify` - Ejecuta tests adicionales sobre el paquete para control de calidad
- `install` - Instala el paquete en el repositorio local
- `deploy` - Despliega el paquete en un repositorio remoto



Maven - Ejercicio 1

Mediante Maven ejecute la creación de un proyecto denominado demo2 y en el mismo implemente un programa que permita la fabricación de objetos automóvil de acuerdo al siguiente diagrama de clases:



- Fases - Ejecutadas por plugins intercambiables
- Goal - Un paso dentro de un plugin que se ejecuta dentro de una fase
- Los Goals ejecutan el trabajo dentro de las fases

```
<build>  
  <plugins>  
    . . .  
  </plugins>  
</build>
```

Maven Plugins - Dependency

```
mvn dependency:analyze
```

```
mvn dependency:tree
```

```
<plugin>
  <groupId>org.apache.maven.plugins</groupId>
  <artifactId>maven-compiler-plugin</artifactId>
  <version>3.6.1</version>
  <configuration>
    <source>1.8</source>
    <target>1.8</target>
  </configuration>
</plugin>
```

Maven Plugins - Jar

```
<plugin>
  <groupId>org.apache.maven.plugins</groupId>
  <artifactId>maven-jar-plugin</artifactId>
  <version>3.1.0</version>
  ...
  <configuration>
    <archive>
      <manifest>
        <addClasspath>true</addClasspath>
        <mainClass>fully.qualified.MainClass</mainClass>
      </manifest>
    </archive>
  </configuration>
  ...
</plugin>
```



Compile y empaque el proyecto, posteriormente ejecute la clase `App.java` contenida en el arquetipo de java

Ejecute los ejercicios 1 y 2 en su IDE de preferencia

Utilice el starter de Helidon para generar una API web con MicroProfile

Automatice la creación de un container en Docker con Maven



**Oracle
Groundbreakers**



ORACLE®
Certified Professional
Java SE 8 Programmer

ORACLE®
Certified Associate
Java SE 8 Programmer



- vorozco@nabenik.com
- @tuxtor
- <http://vorozeo.com>
- <http://tuxtor.shekalug.org>



This work is licensed under Creative
Commons Attribution-
NonCommercial-ShareAlike 3.0
Guatemala (CC BY-NC-SA 3.0 GT).

(CC BY-NC-SA3.0 GT)



ACADEMIK

Escríbenos a cursos@academik.io

www.academik.io

(CC BY-NC-SA3.0 GT)