



**Armando Alaniz**

JUG-Nicaragua

15 de diciembre de 2018



# Acerca de mí

- Java (Mas de 10 años)
- Microsoft SQL Server, Integration Services, Analysis Services.
- Pasatiempos:
  - Natación, Krav Maga (IKMF), Jiujitsu



# Agenda *Maven*<sup>TM</sup>

- Qué es Maven?
- POM
- Dependencies
- Build Life-cycle

# Maven's Objectives

- Making the build process easy
- Providing a uniform build system
- Providing quality project information
- Providing guidelines for best practices development
- Allowing transparent migration to new features

# Project Object Model (POM)

```
1. <project xmlns="http://maven.apache.org/POM/4.0.0"
2.   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
3.   xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
4.     http://maven.apache.org/xsd/maven-4.0.0.xsd">
5.   <modelVersion>4.0.0</modelVersion>
6.
7.   <!-- The Basics -->
8.   <groupId>...</groupId>
9.   <artifactId>...</artifactId>
10.  <version>...</version>
11.  <packaging>...</packaging>
12.  <dependencies>...</dependencies>
13.  <parent>...</parent>
14.  <dependencyManagement>...</dependencyManagement>
15.  <modules>...</modules>
16.  <properties>...</properties>
17.
18.  <!-- Build Settings -->
19.  <build>...</build>
20.  <reporting>...</reporting>
```

```
21.
22.   <!-- More Project Information -->
23.   <name>...</name>
24.   <description>...</description>
25.   <url>...</url>
26.   <inceptionYear>...</inceptionYear>
27.   <licenses>...</licenses>
28.   <organization>...</organization>
29.   <developers>...</developers>
30.   <contributors>...</contributors>
31.
32.   <!-- Environment Settings -->
33.   <issueManagement>...</issueManagement>
34.   <ciManagement>...</ciManagement>
35.   <mailingLists>...</mailingLists>
36.   <scm>...</scm>
37.   <prerequisites>...</prerequisites>
38.   <repositories>...</repositories>
39.   <pluginRepositories>...</pluginRepositories>
40.   <distributionManagement>...</distributionManagement>
41.   <profiles>...</profiles>
42. </project>
```

# Maven Coordinates

- groupId: org.postgresql
- artifactId: postgresql
- version: 42.2.5

# packaging

- pom
- jar
- war
- maven-plugin
- ejb
- ear
- rar

# Dependencies

- my project can only depend upon Maven artifacts
- This forces you to depend solely on dependencies that Maven can manage

```
1. <project xmlns="http://maven.apache.org/POM/4.0.0"
2.   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
3.   xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
4.                       https://maven.apache.org/xsd/maven-4.0.0.xsd">
5.   ...
6.   <dependencies>
7.     <dependency>
8.       <groupId>junit</groupId>
9.       <artifactId>junit</artifactId>
10.      <version>4.12</version>
11.      <type>jar</type>
12.      <scope>test</scope>
13.      <optional>true</optional>
14.    </dependency>
15.    ...
16.  </dependencies>
17.  ...
18. </project>
```



# Dependencies

- classifier
- type
- scope
  - compile: default scope. Compile dependencies are available in all classpaths. Transitive
  - provided: available on the compilation and test classpath , and is not transitive
  - runtime: not required for compilation, but is for execution. It is in the runtime and test classpaths, but not the compile classpath
  - test: test compilation, not transitive
  - system: similar to provided except that you have to provide the JAR which contains it explicitly
- systemPath: is used only if the the dependency scope is system
- optional: Marks optional a dependency when this project itself is a dependency
- mvn dependency:tree

# Dependencies: Exclusions

```
1. <project xmlns="http://maven.apache.org/POM/4.0.0"
2.   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
3.   xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
4.     https://maven.apache.org/xsd/maven-4.0.0.xsd">
5.   ...
6.   <dependencies>
7.     <dependency>
8.       <groupId>org.apache.maven</groupId>
9.       <artifactId>maven-embedder</artifactId>
10.      <version>2.0</version>
11.      <exclusions>
12.        <exclusion>
13.          <groupId>org.apache.maven</groupId>
14.          <artifactId>maven-core</artifactId>
15.        </exclusion>
16.      </exclusions>
17.    </dependency>
18.    ...
19.  </dependencies>
20.  ...
21. </project>
```

```
1. <project xmlns="http://maven.apache.org/POM/4.0.0"
2.   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
3.   xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
4.     https://maven.apache.org/xsd/maven-4.0.0.xsd">
5.   ...
6.   <dependencies>
7.     <dependency>
8.       <groupId>org.apache.maven</groupId>
9.       <artifactId>maven-embedder</artifactId>
10.      <version>3.1.0</version>
11.      <exclusions>
12.        <exclusion>
13.          <groupId>*</groupId>
14.          <artifactId>*</artifactId>
15.        </exclusion>
16.      </exclusions>
17.    </dependency>
18.    ...
19.  </dependencies>
20.  ...
21. </project>
```

# A project cannot be downloaded from the central Maven repository

1. `mvn install:install-file -Dfile=non-maven-proj.jar -DgroupId=some.group -DartifactId=non-maven-proj -Dversion=1 -Dpackaging=jar`
2. `deploy:deploy-file`
3. Set the dependency scope to system and define a `systemPath`

# Repositories

- local: `${user.home}/.m2/repository`. Se puede personalizar en el archivo `settings.xml` en `$M2_HOME/conf`
  - <https://stackoverflow.com/questions/6891984/maven-local-repository-location-for-linux-and-windows>
- remote: Central (Maven Central US repository)
  - <https://search.maven.org/>
  - <https://mvnrepository.com/repos>

# Using a Repository Manager

- <https://maven.apache.org/repository-management.html>
  - [Apache Archiva](#) (open source)
  - [CloudRepo](#) (commercial)
  - [Cloudsmith Package](#) (commercial)
  - [JFrog Artifactory Open Source](#) (open source)
  - [JFrog Artifactory Pro](#) (commercial)
  - [Sonatype Nexus OSS](#) (open source)
  - [Sonatype Nexus Pro](#) (commercial)
  - [packagecloud.io](#) (commercial)
- 
- <https://archiva.apache.org/index.cgi>

# Using Mirrors for Repositories

- <https://maven.apache.org/guides/mini/guide-mirror-settings.html>

```
1. <settings>
2.   ...
3.   <mirrors>
4.     <mirror>
5.       <id>UK</id>
6.       <name>UK Central</name>
7.       <url>http://uk.maven.org/maven2</url>
8.       <mirrorOf>central</mirrorOf>
9.     </mirror>
10.  </mirrors>
11.  ...
12. </settings>
```

```
1. <settings>
2.   ...
3.   <mirrors>
4.     <mirror>
5.       <id>internal-repository</id>
6.       <name>Maven Repository Manager running on repo.mycompany.com</name>
7.       <url>http://repo.mycompany.com/proxy</url>
8.       <mirrorOf>*</mirrorOf>
9.     </mirror>
10.  </mirrors>
11.  ...
12. </settings>
```

# Repositories

- mvn dependency:list-repositories
- <https://stackoverflow.com/questions/2877436/how-can-i-ask-maven-for-a-list-of-the-default-repositories>
  - mvn help:evaluate, \${project.repositories}

# Inheritance

```
1. <project xmlns="http://maven.apache.org/POM/4.0.0"
2.   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
3.   xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
4.       https://maven.apache.org/xsd/maven-4.0.0.xsd">
5.   <modelVersion>4.0.0</modelVersion>
6.
7.   <groupId>org.codehaus.mojo</groupId>
8.   <artifactId>my-parent</artifactId>
9.   <version>2.0</version>
10.  <packaging>pom</packaging>
11. </project>
```

```
1. <project xmlns="http://maven.apache.org/POM/4.0.0"
2.   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
3.   xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
4.       https://maven.apache.org/xsd/maven-4.0.0.xsd">
5.   <modelVersion>4.0.0</modelVersion>
6.
7.   <parent>
8.     <groupId>org.codehaus.mojo</groupId>
9.     <artifactId>my-parent</artifactId>
10.    <version>2.0</version>
11.    <relativePath>../my-parent</relativePath>
12.  </parent>
13.
14.  <artifactId>my-project</artifactId>
15. </project>
```



# Aggregation (or Multi-Module)

```
1. <project xmlns="http://maven.apache.org/POM/4.0.0"
2.   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
3.   xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
4.     https://maven.apache.org/xsd/maven-4.0.0.xsd">
5.   <modelVersion>4.0.0</modelVersion>
6.
7.   <groupId>org.codehaus.mojo</groupId>
8.   <artifactId>my-parent</artifactId>
9.   <version>2.0</version>
10.  <packaging>pom</packaging>
11.
12.  <modules>
13.    <module>my-project</module>
14.    <module>another-project</module>
15.    <module>third-project/pom-example.xml</module>
16.  </modules>
17. </project>
```

# Dependency Management

- Used by a POM to help manage dependency information across all of its children
- Is a mechanism for centralizing dependency information

# Dependency Management

```
1. <project>
2.   ...
3.   <dependencies>
4.     <dependency>
5.       <groupId>group-a</groupId>
6.       <artifactId>artifact-a</artifactId>
7.       <version>1.0</version>
8.       <exclusions>
9.         <exclusion>
10.          <groupId>group-c</groupId>
11.          <artifactId>excluded-artifact</artifactId>
12.        </exclusion>
13.      </exclusions>
14.    </dependency>
15.    <dependency>
16.      <groupId>group-a</groupId>
17.      <artifactId>artifact-b</artifactId>
18.      <version>1.0</version>
19.      <type>bar</type>
20.      <scope>runtime</scope>
21.    </dependency>
  </dependencies>
</project>
```

```
1. <project>
2.   ...
3.   <dependencies>
4.     <dependency>
5.       <groupId>group-c</groupId>
6.       <artifactId>artifact-b</artifactId>
7.       <version>1.0</version>
8.       <type>war</type>
9.       <scope>runtime</scope>
10.    </dependency>
11.    <dependency>
12.      <groupId>group-a</groupId>
13.      <artifactId>artifact-b</artifactId>
14.      <version>1.0</version>
15.      <type>bar</type>
16.      <scope>runtime</scope>
17.    </dependency>
18.  </dependencies>
19. </project>
```

# Dependency Manag

- Parent POM

```
1. <project>
2.   ...
3.   <dependencyManagement>
4.     <dependencies>
5.       <dependency>
6.         <groupId>group-a</groupId>
7.         <artifactId>artifact-a</artifactId>
8.         <version>1.0</version>
9.
10.        <exclusions>
11.          <exclusion>
12.            <groupId>group-c</groupId>
13.            <artifactId>excluded-artifact</artifactId>
14.          </exclusion>
15.        </exclusions>
16.      </dependency>
17.
18.      <dependency>
19.        <groupId>group-c</groupId>
20.        <artifactId>artifact-b</artifactId>
21.        <version>1.0</version>
22.        <type>war</type>
23.        <scope>runtime</scope>
24.      </dependency>
25.
26.      <dependency>
27.        <groupId>group-a</groupId>
28.        <artifactId>artifact-b</artifactId>
29.        <version>1.0</version>
30.        <type>bar</type>
31.        <scope>runtime</scope>
32.      </dependency>
33.    </dependencies>
34.  </dependencyManagement>
35. </project>
```

# Dependency Management

```
1. <project>
2.   ...
3.   <dependencies>
4.     <dependency>
5.       <groupId>group-a</groupId>
6.       <artifactId>artifact-a</artifactId>
7.     </dependency>
8.
9.     <dependency>
10.      <groupId>group-a</groupId>
11.      <artifactId>artifact-b</artifactId>
12.      <!-- This is not a jar dependency, so we must specify type. -->
13.      <type>bar</type>
14.    </dependency>
15.  </dependencies>
16. </project>
```

```
1. <project>
2.   ...
3.   <dependencies>
4.     <dependency>
5.       <groupId>group-c</groupId>
6.       <artifactId>artifact-b</artifactId>
7.       <!-- This is not a jar dependency, so we must specify type. -->
8.       <type>war</type>
9.     </dependency>
10.
11.     <dependency>
12.       <groupId>group-a</groupId>
13.       <artifactId>artifact-b</artifactId>
14.       <!-- This is not a jar dependency, so we must specify type. -->
15.       <type>bar</type>
16.     </dependency>
17.   </dependencies>
18. </project>
```

# Dependency Management

- Minimal set of information for matching a dependency reference against a **dependencyManagement** section is actually {groupId, artifactId, type, classifier}

# Properties

- <https://books.sonatype.com/mvnref-book/reference/resource-filtering-sect-properties.html>
- `${env.PATH}`
- `${project.version}`
- `${settings.offline}`
- Java System Properties: `${java.home}`
- `someVar: ${someVar}`

```
1. <project>
2.   ...
3.   <properties>
4.     <maven.compiler.source>1.7</maven.compiler.source>
5.     <maven.compiler.target>1.7</maven.compiler.target>
6.     <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
7.     <project.reporting.outputEncoding>UTF-8</project.reporting.outputEncoding>
8.   </properties>
9.   ...
10. </project>
```

# Build Lifecycle

- <https://maven.apache.org/guides/introduction/introduction-to-the-lifecycle.html>
- Clean Lifecycle
- Default Lifecycle
- Site Lifecycle



# Clean Lifecycle

- pre-clean
- clean
- post-clean

# Default Lifecycle

- validate: validate the project is correct and all necessary information is available
- compile: compile the source code of the project
- test: test the compiled source code using a suitable unit testing framework
- package: take the compiled code and package it in its distributable format, such as a JAR
- verify: run any checks on results of integration tests to ensure quality criteria are met
- install: install the package into the local repository, for use as a dependency in other projects locally
- deploy: done in the build environment, copies the final package to the remote repository for sharing with other developers and projects.

# Site Lifecycle

- pre-site
- site
- post-site
- site-deploy

# Build Lifecycle

- The order of execution depends on the order in which the goal(s) and the build phase(s) are invoked
  - if a goal is bound to one or more build phases, that goal will be called in all those phases
  - A build phase can also have zero or more goals bound to it. If a build phase has no goals bound to it, that build phase will not execute. But if it has one or more goals bound to it, it will execute all those goals
- 
- mvn install
  - mvn clean deploy
  - mvn clean dependency:copy-dependencies package

# Gracias



 [Javanicaragua.org](http://Javanicaragua.org)

 @jug\_nicaragua

 jug-ni

