

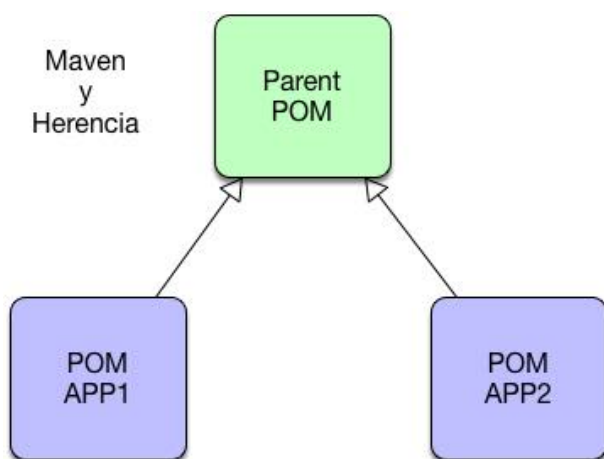
Blog sobre Java EE

Estás aquí: [Inicio](#)/[Java EE](#)/[Maven](#)/Maven Parent POM y uso de librerías

Maven Parent POM y uso de librerías

10 noviembre, 2017 por [Cecilio Álvarez Caules](#) — [Deja un comentario](#)

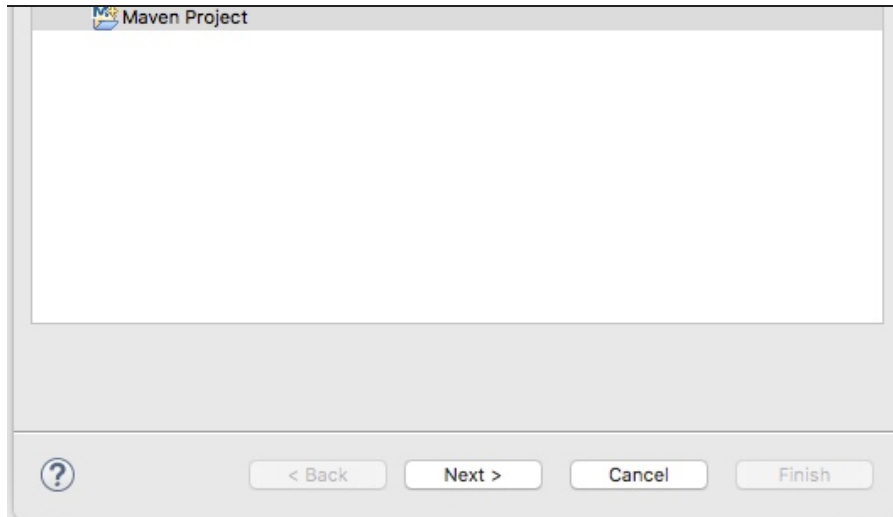
El concepto de **Maven Parent POM** es muy común en cuanto comenzamos a realizar aplicaciones un poco complejas **compuestas por varios proyectos**. La mayor parte de estos proyectos comparten el 90% de las librerías. De ahí surge la necesidad de compartir librerías entre varios proyectos. Ese es el problema que **nos solventan los Maven Parent POM**. Vamos a ver como utilizarlos a través de un ejemplo muy sencillo **con dos proyectos web que utilizan JSTL**. Ambos proyectos comparten las mismas librerías por lo tanto podemos definir una estructura de Maven jerárquica.



Maven Parent POM

Lo primero que vamos a hacer es crear un proyecto **en Eclipse con Maven**. Para ello abrimos el entorno de desarrollo y seleccionamos proyecto Maven.

Este sitio web utiliza cookies para que usted tenga la mejor experiencia de usuario. Si continúa navegando está dando su consentimiento para la aceptación de las mencionadas cookies y la aceptación de nuestra [política de cookies](#), pinche el enlace para mayor información.

[plugin cookies](#)[ACEPTAR](#)

Una vez seleccionado Maven , pulsamos continuar y crear Artefacto simple .Nos aparece una nueva ventana en donde debemos configurar nuestro proyecto Maven.

A screenshot of the 'New Maven Project' configuration window. The title bar says 'New Maven Project'. Below the title bar, it says 'New Maven project' and 'Configure project'. There is a Maven logo icon. The window is divided into sections: 'Artifact' and 'Parent Project'. In the 'Artifact' section, there are fields for 'Group Id' (com.arquitecturajava), 'Artifact Id' (POMPadre), 'Version' (1), 'Packaging' (pom), 'Name', and 'Description'. In the 'Parent Project' section, there are fields for 'Group Id', 'Artifact Id', and 'Version', along with 'Browse...' and 'Clear' buttons. At the bottom, there is an 'Advanced' section (collapsed) and a footer with a question mark icon and buttons for '< Back', 'Next >', 'Cancel', and 'Finish'.

En este proyecto vamos a definir las librerías que serán compartidas **por los proyectos hijos**. En este caso voy a optar por las librerías de JSTL que son clásicas.

```
1 <project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3
2   <modelVersion>4.0.0</modelVersion>
3   <groupId>com.arquitecturajava</groupId>
4   <artifactId>POMPadre</artifactId>
5   <version>1</version>
6   <packaging>pom</packaging>
7   <dependencies>
8     <dependency>
9       <groupId>org.apache.taglibs</groupId>
10      <artifactId>taglibs-standard-impl</artifactId>
11      <version>1.2.5</version>
12    </dependency>
13    <dependency>
14      <groupId>javax.servlet.jsp.jstl</groupId>
15      <artifactId>jstl-api</artifactId>
16      <version>1.2</version>
17    </dependency>
18    <dependency>
19      <groupId>javax.servlet</groupId>
20      <artifactId>javax.servlet-api</artifactId>
21      <version>3.0.1</version>
22      <scope>provided</scope>
23    </dependency>
24  </dependencies>
25 </project>
```

Realizada esta operación vamos a crear **con el mismo formato un proyecto hijo**. En este caso será un proyecto Web:

Group Id: com.arquitecturajava

Este sitio web utiliza cookies para que usted tenga la mejor experiencia de usuario. Si continúa navegando está dando su consentimiento para la aceptación de las mencionadas cookies y la aceptación de nuestra [política de cookies](#), pinche el enlace para mayor información.

[plugin cookies](#)

ACEPTAR

Una vez creado el proyecto, el fichero **POM.xml** contendrá una referencia al **POM padre**:

```

1  <project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3
2  <modelVersion>4.0.0</modelVersion>
3
4  <artifactId>WebMaven2</artifactId>
5
6  <packaging>war</packaging>
7  <parent>
8  <groupId>com.arquitecturajava</groupId>
9  <artifactId>POMPadre</artifactId>
10 <version>1</version>
11 </parent>
12 </project>

```

Podemos crear una página JSP en nuestro proyecto web que haga uso de las librerías de **JSTL**.

```

1  <%@ page language="java" contentType="text/html; charset=UTF-8" pageEncodin
2  <%@ taglib uri = "http://java.sun.com/jsp/jstl/core" prefix = "c" %>
3  <!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.
4  <html>
5  <head>
6  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
7  <title>Insert title here</title>
8  </head>
9  <body>
10 <c:out value="hola jstl"/>
11 </body>
12 </html>

```

Este sitio web utiliza cookies para que usted tenga la mejor experiencia de usuario. Si continúa navegando está dando su consentimiento para la aceptación de las mencionadas cookies y la aceptación de nuestra [política de cookies](#), pinche el enlace para mayor información.

[plugin cookies](#)[ACEPTAR](#)

Acabamos de construir un proyecto Maven que delega en un parent POM para definir la estructura de librerías común que puede ser compartida entre varios podríamos crear otro igual.

Otros artículos relacionados

1. [Utilizando Maven Profiles](#)
2. [Maven y Dependencias \(I\)](#)
3. [Maven y Dependencias \(II\)](#)
4. [Maven y Eclipse \(II\)](#)
5. [Maven](#)



Archivada en: [Maven](#)

Deja un comentario

Tu dirección de correo electrónico no será publicada. Los campos obligatorios están marcados con *

Comentario

Nombre *

Este sitio web utiliza cookies para que usted tenga la mejor experiencia de usuario. Si continúa navegando está dando su consentimiento para la aceptación de las mencionadas cookies y la aceptación de nuestra [política de cookies](#), pinche el enlace para mayor información.

plugin cookies

ACEPTAR

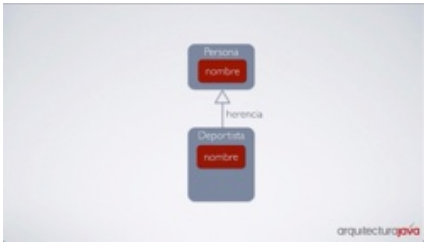
PUBLICAR COMENTARIO

BUSCAR

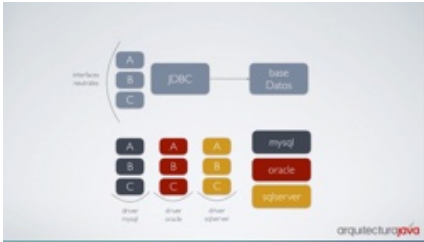
Buscar en este sitio ...

Mis Cursos de Java Gratuitos

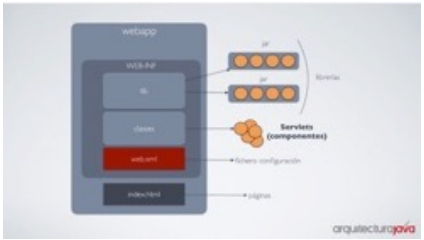
Java Herencia



Java JDBC



Servlets



Este sitio web utiliza cookies para que usted tenga la mejor experiencia de usuario. Si continúa navegando está dando su consentimiento para la aceptación de las mencionadas cookies y la aceptación de nuestra [política de cookies](#), pinche el enlace para mayor información.

[plugin cookies](#)

ACEPTAR

POPULAR

- [Java Stream map y estadísticas](#)
- [Maven Parent POM y uso de librerías](#)
- [Spring GetMapping ,PostMapping etc](#)
- [Java Stream Context y simplificación de Streams](#)
- [Mis Cursos de Java para desarrolladores](#)
- [Spring Boot WAR sin Microservicios](#)
- [Spring 5 Hello World](#)
- [Eclipse Git , Repositorios locales y remotos](#)
- [Java Overload y preguntas de certificacion](#)
- [JPA DTO \(Data Transfer Object\) y JPQL](#)

CONTACTO

contacto@arquitecturajava.com

LO MAS LEIDO

Este sitio web utiliza cookies para que usted tenga la mejor experiencia de usuario. Si continúa navegando está dando su consentimiento para la aceptación de las mencionadas cookies y la aceptación de nuestra [política de cookies](#), pinche el enlace para mayor información.

[plugin cookies](#)[ACEPTAR](#)

[Ejemplo de JPA , introduccion \(1\)](#)

[¿Cuales son las certificaciones Java?](#)

[¿Qué es Gradle?](#)

[Usando el patron factory](#)

[Introducción a Servicios REST](#)

[Ejemplo de Java Singleton \(Patrones y ClassLoaders\)](#)

[Java Stream map y estadísticas](#)

[Uso de Java Generics \(I\)](#)

[REST JSON y Java](#)

[Mis Libros](#)

[Comparando java == vs equals](#)

[Spring MVC Configuración \(I\)](#)

[Java Override y encapsulación](#)

[Spring GetMapping ,PostMapping etc](#)

[¿Qué es un Microservicio?](#)
