

LABORATORIO 6

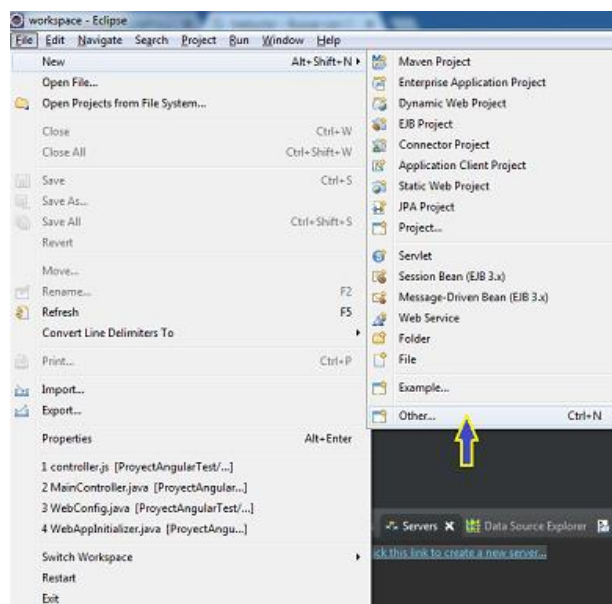
Uno de los contenedores más populares en el mundo de la programación, es el nacido bajo el proyecto Jakarta en la Apache Software Foundation, muy conocido como Apache Tomcat o simplemente Tomcat.

Tomcat es un contenedor de Servlet, mas sin embargo, a veces suele confundirse como un servidor de aplicaciones ya que puede funcionar como servidor web por sí mismo, pero no es así, aunque hoy en día, es utilizado en entornos de alta demanda y flujo intenso de tráfico.

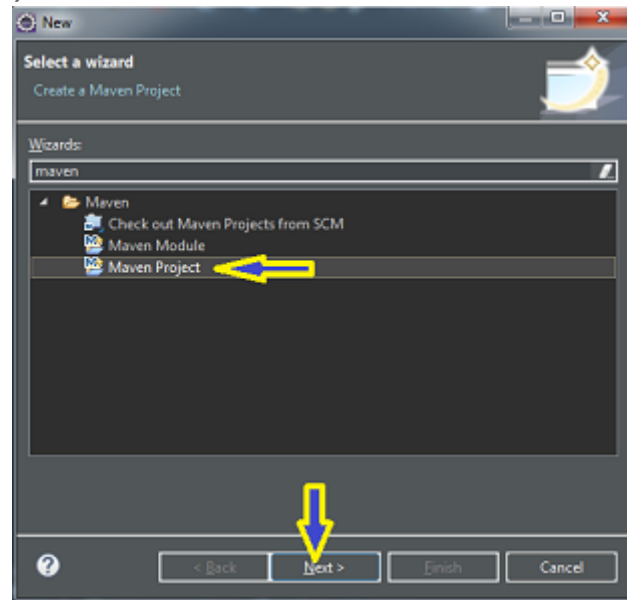
En este post explicaremos como integrar el plugin de tomcat para trabajarlo sobre un proyecto Java Web, ya que, con la llegada de Maven es más práctico hoy en día, compilar, desplegar y ejecutar una aplicación en el mundo de la web.

Creando un proyecto Java Web desde Eclipse con Maven

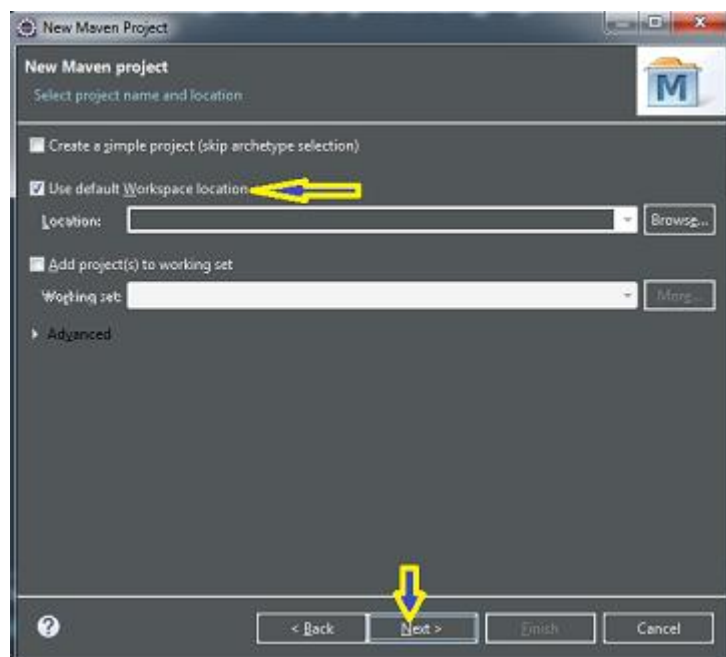
Damos clic en File(Archivo) -> New(Nuevo) -> Other(Otro)



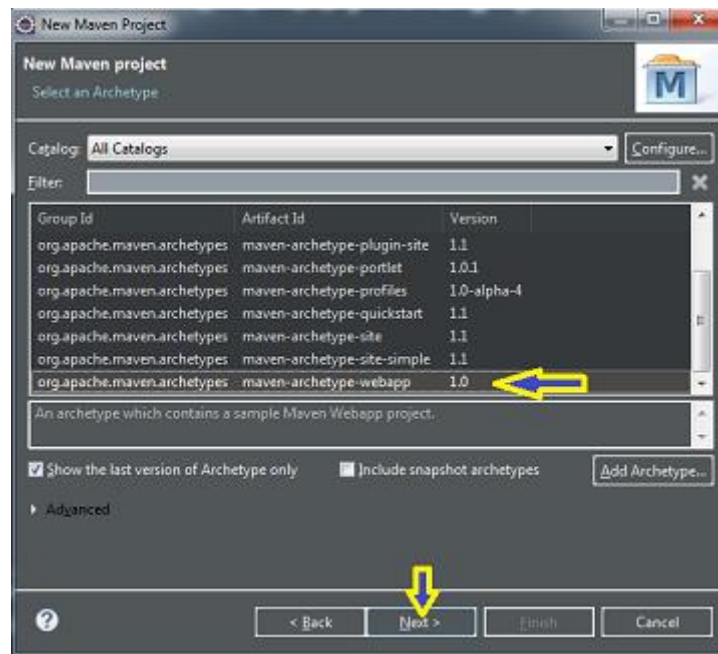
Nos aparecerá una ventana para la creación del tipo de proyecto que queremos crear, escribiremos en el Wizards la palabra "maven" y nos aparecerán las opciones de lo que queremos crear con maven, seleccionamos **Maven Project** y le damos clic en Next(Siguiente)



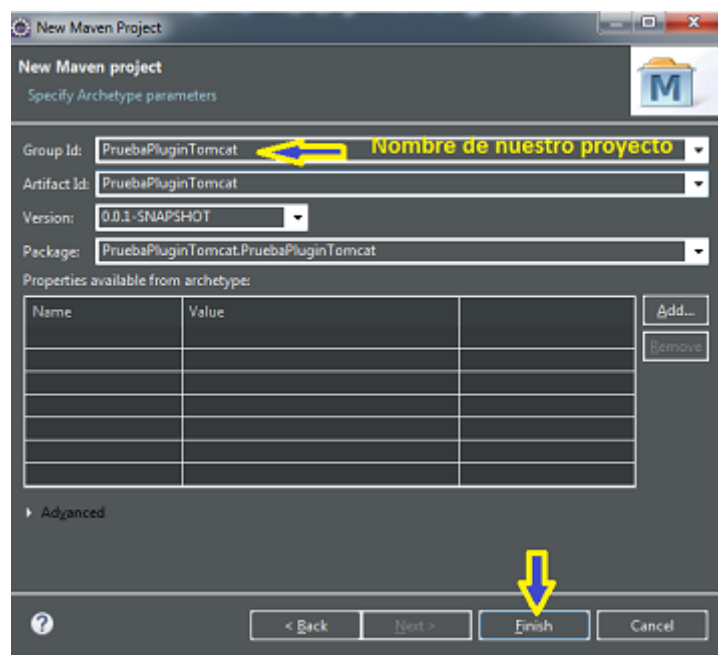
Nos aparecerá otra ventana, la cual dejaremos como esta, ya que, si presionamos el check que dice "Créate a simple project", crearemos un simple proyecto de java, y lo que buscamos es crear un Java Web, entonces daremos clic en Next(Siguiente)



Nuevamente nos aparecerá otra ventana donde nos muestra los tipos de proyecto que podemos crear con los arquetipos de Maven, nosotros escogeremos “maven-archetype-webapp” y daremos clic en Next(Siguiente)



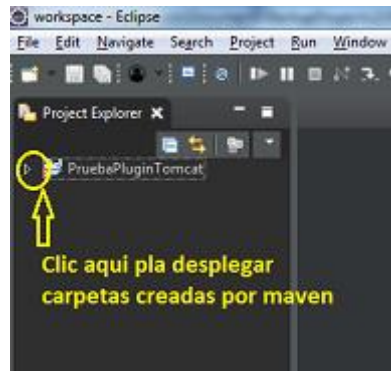
Por ultimo nos muestra otra ventana, donde ingresaremos el nombre de nuestro proyecto



Y daremos clic en Finish(Finalizar)

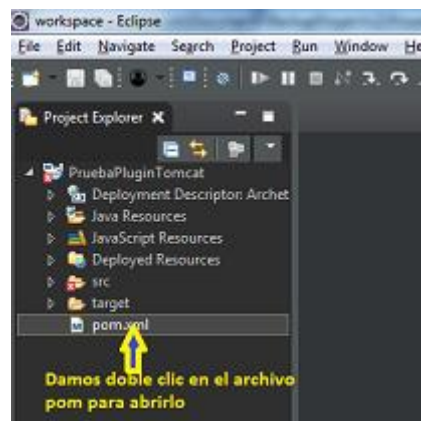
Hasta este punto, en la imagen de abajo podemos observar que se ha creado nuestro proyecto.

Para poder ver las carpetas creadas por Maven solo damos clic en la flechita del proyecto



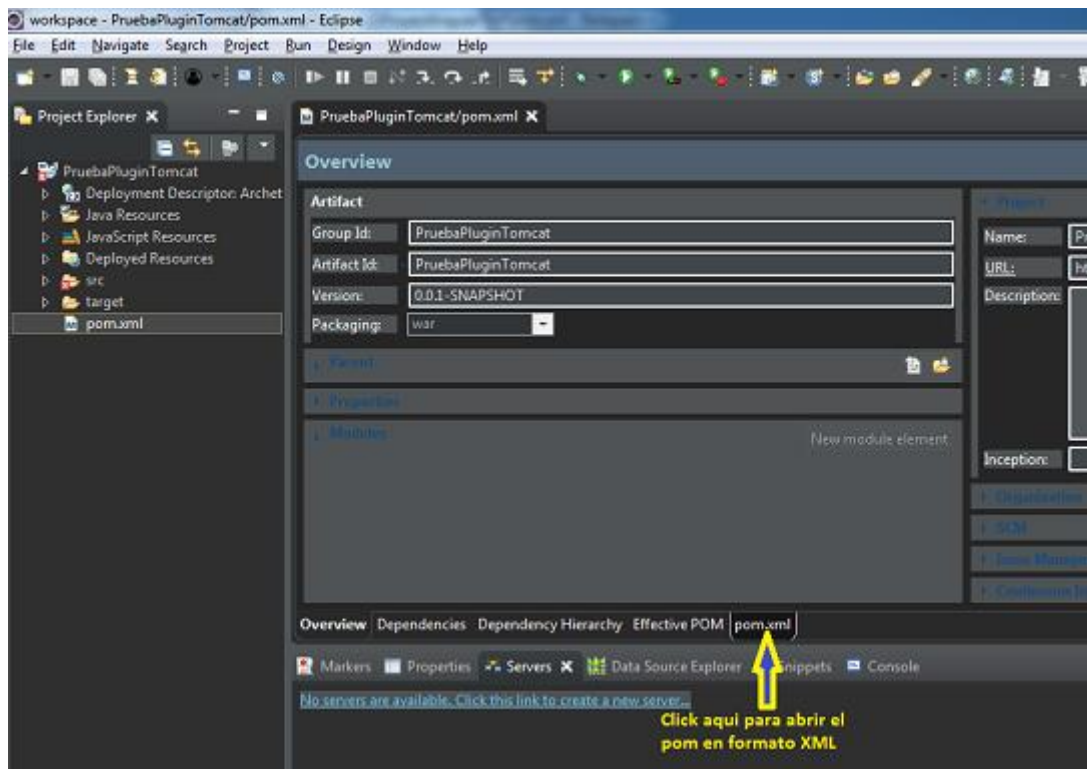
Y se desplegarán los archivos y carpetas creadas por Maven.

Daremos doble clic en el archivo Pom de maven para poder ver su contenido

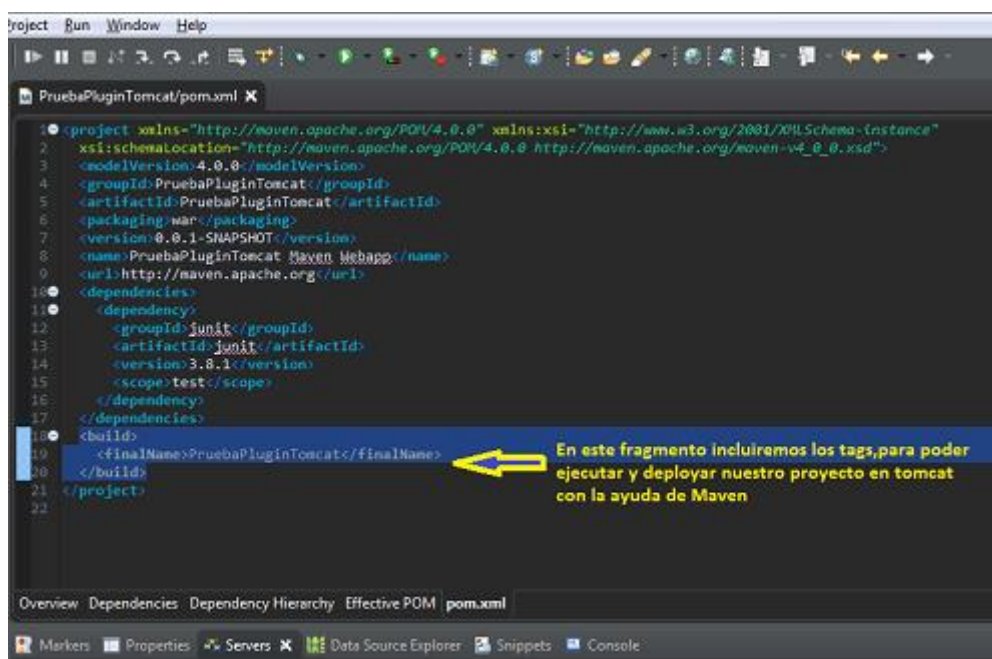


Observamos que eclipse nos abre el editor, pero no nos muestra el archivo en formato xml.

Para ello daremos clic en la pestaña pom.xml como se muestra en la siguiente figura



Ahora si podemos visualizar el archivo pom.xml de nuestro proyecto



De la figura anterior podemos observar que Maven, nos construye un xml con las características que tiene nuestro proyecto, como el nombre y la versión, pero aunado a eso, Maven nos instala la configuración para realizar pruebas unitarias a través de junit, del cual estaremos hablando en otro post.

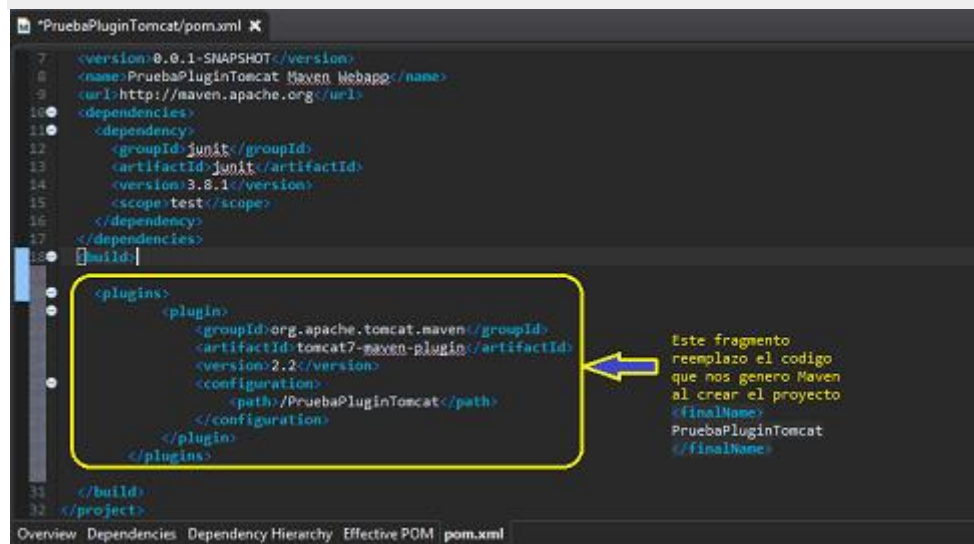
Otra cosa que podemos observar es que también nos instala el tag , el cual se utilizara para construir lo necesario para levantar el proyecto, en este tag, vamos a incorporar el plugin de Tomcat para asi deployar nuestro aplicativo.

Hasta aquí hemos creado un proyecto con Maven.

Integrando el plugin de Tomcat en un proyecto Maven

Para integrar Tomcat como un componente de nuestro proyecto Maven, bastaría con insertar dentro de los tags

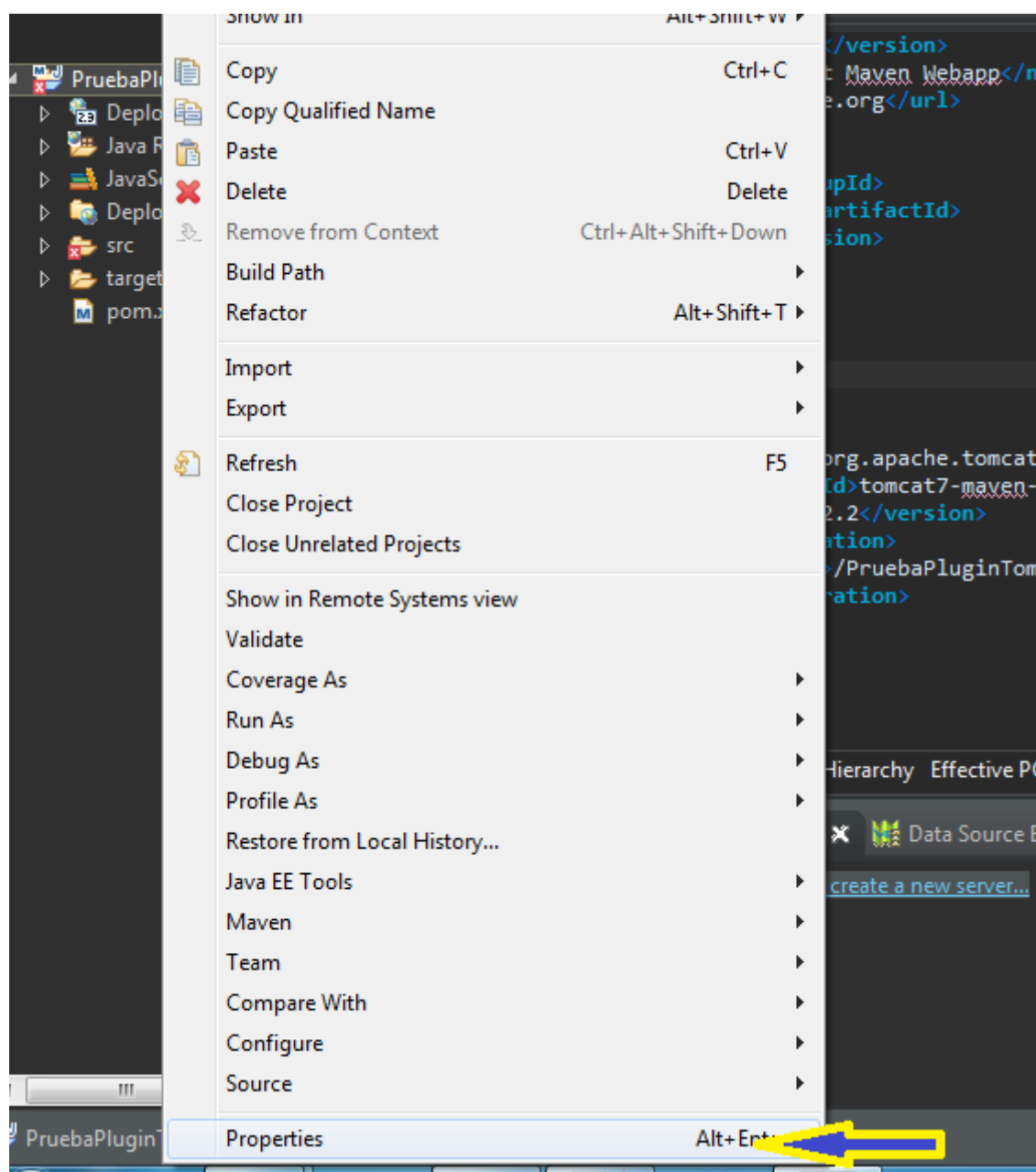
```
<plugins>
  <groupId>org.apache.tomcat.maven</groupId>
  <artifactId>tomcat7-maven-plugin</artifactId>
  <version>2.2</version>
  <configuration>
    <port>9090</port>
    <path>/</path>
    <contextReloadable>true</contextReloadable>
  </configuration>
</plugin>
</plugins>
```



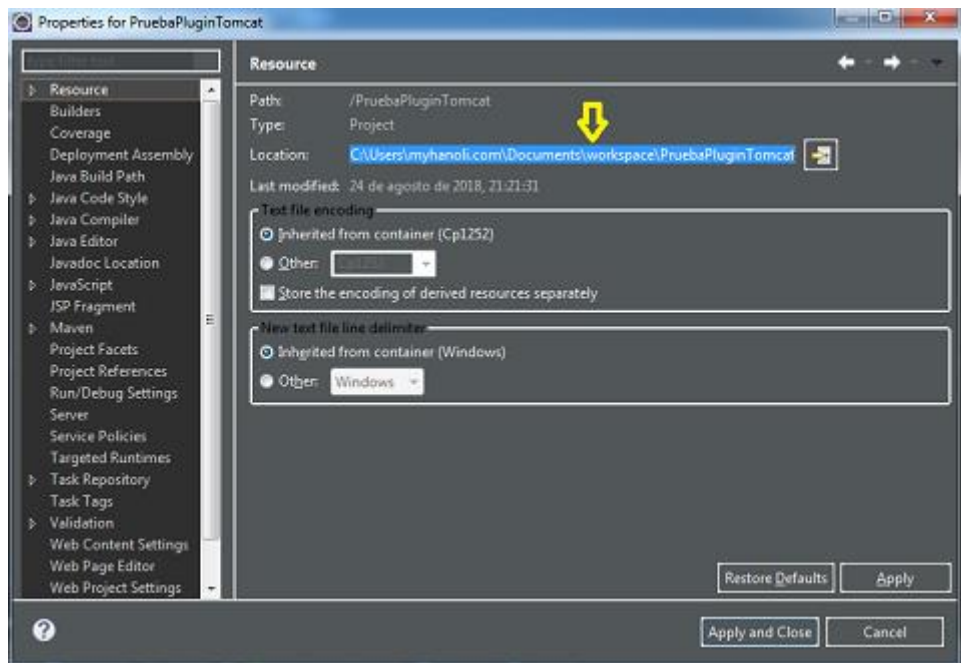
Extrayendo ruta de nuestro proyecto

Lo primero que tenemos que hacer es extraer la ruta donde se encuentra nuestro proyecto, podemos hacerlo de la forma que queramos, sin embargo les comparto como yo lo hago.

Damos clic derecho en el proyecto para que nos despliegue las acciones y daremos clic en properties o propiedades, como se muestra en la figura



Nos mostrara la pantalla de propiedades de nuestro proyecto, seleccionamos la ruta de nuestro proyecto como se muestra en la figura y le damos copiar.



Finalmente abrimos un cmd (Linea de comando de windows) y escribimos el comando `cd` y pegamos nuestra ruta,posteriormente damos un enter para colocarnos en la ruta de nuestro proyecto.

Compilando nuestro proyecto

Una vez dentro de nuestro proyecto, es el turno de que Maven haga su trabajo.

Para ello escribiremos el comando

```
1 | mvn clean install
```

como se muestra en la siguiente figura


```
Microsoft Windows [Versión 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. Reservados todos los derechos.

C:\Users\myhanoli.com>cd C:\Users\myhanoli.com\Documents\workspace\PruebaPluginTomcat

C:\Users\myhanoli.com\Documents\workspace\PruebaPluginTomcat>mvn clean install
```

Y observamos que la compilación termine correctamente

```
Results :
Tests run: 0, Failures: 0, Errors: 0, Skipped: 0

[INFO] --- maven-war-plugin:2.1.1:war (default-war) @ PruebaPluginTomcat ---
[INFO] Packaging webapp
[INFO] Assembling webapp [PruebaPluginTomcat] in [C:\Users\myhanoli.com\Documents\workspace\PruebaPluginTomcat\target\PruebaPluginTomcat]
[INFO] Processing war project
[INFO] Copying webapp resources [C:\Users\myhanoli.com\Documents\workspace\PruebaPluginTomcat\src\main\webapp]
[INFO] Webapp assembled in [34 ms]
[INFO] Building war: C:\Users\myhanoli.com\Documents\workspace\PruebaPluginTomcat\target\PruebaPluginTomcat.war
[WARNING] Warning: selected war files include a WEB-INF/web.xml which will be ignored
[INFO] --- maven-install-plugin:2.3.1:install (default-install) @ PruebaPluginTomcat ---
[INFO] Installing C:\Users\myhanoli.com\Documents\workspace\PruebaPluginTomcat\target\PruebaPluginTomcat.war to C:\Users\myhanoli.com\.m2\repository\PruebaPluginTomcat\PruebaPluginTomcat\0.0.1-SNAPSHOT\PruebaPluginTomcat-0.0.1-SNAPSHOT.war
[INFO] Installing C:\Users\myhanoli.com\Documents\workspace\PruebaPluginTomcat\pom.xml to C:\Users\myhanoli.com\.m2\repository\PruebaPluginTomcat\PruebaPluginTomcat\0.0.1-SNAPSHOT\PruebaPluginTomcat-0.0.1-SNAPSHOT.pom
[INFO] BUILD SUCCESS
[INFO] Total time: 1.470s
[INFO] Finished at: Fri Aug 24 21:41:11 CDT 2018
[INFO] Final Memory: 6M/15M
[INFO] C:\Users\myhanoli.com\Documents\workspace\PruebaPluginTomcat>
```

Desplegando nuestro proyecto en Tomcat

Para que podamos visualizar nuestro proyecto en el navegador, basta con escribir el comando de maven “mvn tomcat7:run” una vez que la compilación haya terminado

```
[INFO] BUILD SUCCESS
[INFO] Total time: 1.470s
[INFO] Finished at: Fri Aug 24 21:41:11 CDT 2018
[INFO] Final Memory: 6M/15M
[INFO] C:\Users\myhanoli.com\Documents\workspace\PruebaPluginTomcat>mvn tomcat7:run
```

Y nos mostrara que tomcat se ha levantado correctamente y se ha cargado nuestro proyecto en Tomcat

```

C:\Users\ayhanoli.com\Documents\workspace\PruebaPluginTomcat>mvn tomcat7:run
[INFO] Scanning for projects...
[INFO]
[INFO] -----
[INFO] Building PruebaPluginTomcat Maven Webapp 0.0.1-SNAPSHOT
[INFO] -----
[INFO] >>> tomcat7-maven-plugin:2.2:run (default-cli) @ PruebaPluginTomcat >>>
[INFO]
[INFO] --- maven-resources-plugin:2.5:resources (default-resources) @ PruebaPluginTomcat ---
[debug] execute contextualize
[WARNING] Using platform encoding (Cp1252 actually) to copy filtered resources,
i.e. build is platform dependent!
[INFO] Copying 0 resource
[INFO]
[INFO] --- maven-compiler-plugin:2.3.2:compile (default-compile) @ PruebaPluginTomcat ---
[INFO] No sources to compile
[INFO]
[INFO] <<< tomcat7-maven-plugin:2.2:run (default-cli) @ PruebaPluginTomcat <<<
[INFO]
[INFO] --- tomcat7-maven-plugin:2.2:run (default-cli) @ PruebaPluginTomcat ---
[INFO] Running war on http://localhost:8080/PruebaPluginTomcat
[INFO] Using existing Tomcat server configuration at C:\Users\ayhanoli.com\Documents\workspace\PruebaPluginTomcat\target\tomcat
[INFO] Create webapp with contextPath: /PruebaPluginTomcat
ago 24, 2018 9:52:50 PM org.apache.coyote.AbstractProtocol init
INFORMACION: Initializing ProtocolHandler ["http-bio-8080"]
ago 24, 2018 9:52:50 PM org.apache.catalina.core.StandardService startInternal
INFORMACION: Starting service Tomcat
ago 24, 2018 9:52:50 PM org.apache.catalina.core.StandardEngine startInternal
INFORMACION: Starting Servlet Engine: Apache Tomcat/7.0.47
ago 24, 2018 9:52:52 PM org.apache.coyote.AbstractProtocol start
INFORMACION: Starting ProtocolHandler ["http-bio-8080"]

```

Probando nuestro aplicativo en el navegador

Para poder probar nuestro aplicativo abriremos la ruta

`http://localhost:9090`

La colocaremos en nuestro navegador y podremos observar(si todo se hizo correctamente), nuestro proyecto corriendo en tomcat

Aquí tienes un repo con el código:

<https://github.com/reshmabidikar/learnjava-webapp-example/blob/master/pom.xml>