

# Instalación de Tomcat y Maven para despliegue de aplicación Java

## Tabla de Contenido

1. Introducción .....	1
2. Instalación de Tomcat .....	2
2.1. Permitir el acceso a nuestro servidor desde otro ordenador.....	2
2.2. Despliegue manual mediante la GUI de administración.....	3
3. Despliegue con Maven.....	3
3.1. Instalación de Maven.....	3
3.2. Configuración de Maven.....	4
3.3. Despliegue .....	7
4. Tarea .....	8
5. Cuestiones .....	8
6. Referencias.....	9
7. Evaluación .....	9

## 1. Introducción

Si consultamos el apartado de [versiones de Tomcat](#) en su página oficial, nos daremos cuenta de que no vamos a usar la última versión, para esta práctica, sino la 9. La pregunta es casi inevitable:

### ¿Por qué?

En el enlace anterior vemos como desde su versión 9, Tomcat da soporte a Java 8 y superiores. Sin embargo, a partir de la versión 10.1.x, da soporte a Java 11 y superiores.

### ¿Qué significa esto?

En Java 9 se introdujeron novedades como un nuevo sistema de módulos (Jigsaw), [entre otras](#).

En Java 11 se dio un paso más al haber renombrado completamente las rutas de paquetes [javax](#). a [jakarta](#).. Oracle, a pesar de haber hecho público el desarrollo de Java, no hizo lo mismo con el nombre de Java.

Así las cosas, resulta que Java 8 puede que a día de hoy aún sea la más usada en proyectos reales. Dicho esto, podría realizarse un proceso de migración de un proyecto de Java 8 a Java 11 y utilizarlo en Tomcat 10.

No obstante, para Java 8 su soporte para uso comercial (pagando) acabó [en Marzo de 2022](#), pero para uso no comercial sigue hasta 2030.

En conclusión, no es raro encontrarse en el mundo real un proyecto a desplegar realizado en Java 8.

Podría realizarse una migración y los conceptos de despliegue que veremos seguirían aplicando. Así las cosas, por facilidad en la realización de las prácticas utilizaremos Tomcat 9 y el plugin oficial de Maven para Tomcat 7 para el despliegue (luego veremos el motivo).

## 2. Instalación de Tomcat

Esta práctica es muy sencilla y va a consistir en realizar la instalación del servidor de aplicaciones Tomcat 9, en una máquina virtual corriendo Debian 11 Bullseye.



Se puede hacer tanto con el administrador de paquetes `apt` como de forma manual. La forma más recomendable por su sencillez es la primera.

Para ello, y como sugerencia, podéis apoyaros en [este tutorial online](#), aunque sóis libres de consultar tantas fuentes como deseéis.

Obviamente, debéis utilizar vuestro propios usuarios y contraseña.

### 2.1. Permitir el acceso a nuestro servidor desde otro ordenador

Tomcat no permite por defecto que se acceda a algunos paneles de control desde un ordenador distinto al `localhost`. Para solucionarlo sustituiremos el fichero `context.xml` del directorio `/usr/share/tomcat9-admin/host-manager/META-INF/` por el siguiente:

Fichero `/usr/share/tomcat9-admin/host-manager/META-INF/context.xml`

```
<?xml version="1.0" encoding="UTF-8"?>
<Context antiResourceLocking="false" privileged="true" >
  <CookieProcessor className="org.apache.tomcat.util.http.Rfc6265CookieProcessor"
    sameSiteCookies="strict" />
</Context>
```

```
<Valve className="org.apache.catalina.valves.RemoteAddrValve"
      allow="\d+\.\d+\.\d+\.\d+" />
<Manager
sessionAttributeValueClassNameFilter="java\.lang\.(?:Boolean|Integer|Long|Number|String)|org\.apache\.catalina\.filters\.CsrfPreventionFilter\$LruCache(?:\$1)?|java\.util\.(?:Linked)?HashMap"/>
</Context>
```

Recargaremos el servidor:

```
sudo systemctl restart tomcat9
```

## 2.2. Despliegue manual mediante la GUI de administración

Realizaremos el despliegue manual de una aplicación ya previamente empaquetada en formato WAR. Para ello:

1. Nos logueamos con el usuario previamente creado.
2. Buscamos la sección que nos permite desplegar un WAR manualmente, seleccionamos nuestro archivo y lo desplegamos.

Tras estos pasos, se nos listará la aplicación ya desplegada como un directorio más y podremos acceder a ella.

Documenta el despliegue manual de la aplicación que os podéis descargar para tal efecto en la actividad (archivo .war).

## 3. Despliegue con Maven

### 3.1. Instalación de Maven

Para instalar Maven en nuestro Debian tenemos, de nuevo, dos opciones:

- Instalación mediante gestor de paquetes APT
- Instalación manual

La primera, recomendada, es mucho más sencilla y automatizada (establece todos los paths y

variables de entorno), aunque con la segunda se podría conseguir un paquete más actualizado.

Ambos métodos vienen explicados [aquí](https://www.how2shout.com/linux/how-to-install-apache-maven-on-debian-11-bullseye/)

Si decidimos seguir el primer método, el más sencillo, vemos que es tan simple como actualizar los repositorios e instala Maven:

```
sudo apt-get update
sudo apt-get install maven
```

Para comprobar que todo ha ido correctamente, podemos ver la versión instalada de Maven:

```
mvn --v
```

## 3.2. Configuración de Maven

Para poder realizar despliegues en nuestro Tomcat previamente instalado, necesitamos realizar la configuración adecuada para Maven. Ya sabemos que esto en Linux significa editar los archivos de configuración adecuados. Vamos a ello.

### 3.2.1. Roles

En primer lugar necesitamos asegurarnos de que en el apartado anterior de la práctica hemos añadido todos los usuarios necesarios, así como sus respectivos roles. Debemos añadir el rol de **manager-script** para permitir que Maven se autentique contra Tomcat y pueda realizar el despliegue.

Los roles utilizados por Tomcat vienen detallados en [su documentación](#), que merece ser consultada:

You can find the role names in the [web.xml](#) file of the Manager web application. The available roles are:

- **manager-gui** — Access to the HTML interface.
- **manager-status** — Access to the "Server Status" page only.
- **manager-script** — Access to the tools-friendly plain text interface that is described in this document, and to the "Server Status" page.
- **manager-jmx** — Access to JMX proxy interface and to the "Server Status" page.

En dicha documentación se nos indica que, por temas de seguridad, es recomendable no otorgar los roles de **manager-script** o **manager-jmx** al mismo usuario que tenga el rol de **manager-gui**.



Tendremos dos usuarios, uno para la GUI y otro exclusivamente para hacer los deploys de Maven.

Así las cosas, modificamos el archivo `/etc/tomcat9/tomcat-users.xml` acorde a nuestras necesidades (los nombres de usuario y contraseña deberán ser los que elijáis para vosotros):

Extracto del fichero `/etc/tomcat9/tomcat-users.xml`

```
...
```

```

<role rolename="admin"/>
<role rolename="admin-gui"/>
<role rolename="manager"/>
<role rolename="manager-gui"/>
<role rolename="manager-status"/>
<role rolename="manager-script"/>
<role rolename="manager-jmx"/>

<user username="alumno"
  password="1234"
  roles="admin, admin-gui, manager, manager-gui"
/>
<user username="deploy" ①
  password="1234"
  roles="manager-script"
/>
...

```

① Usuario y contraseña para el despliegue

### 3.2.2. settings.xml

Editar el archivo `/etc/maven/settings.xml` para indicarle a Maven, un identificador para el servidor sobre el que vamos a desplegar (no es más que un nombre, ponedle el nombre que consideréis), así como las credenciales. Todo esto se hará dentro del bloque **servers** del XML:

*Extracto del fichero `/etc/maven/settings.xml`*

```

...
<servers>
  <server>
    <id>Tomcat</id>
    <username>deploy</username> ①
    <password>1234</password>
  </server>
</servers>
...

```

① Atentos al usuario y contraseña de despliegue

### 3.2.3. Generar una aplicación

Generaremos una aplicación de ejemplo en nuestro directorio personal:

```
cd
```

```
$ mvn archetype:generate -DgroupId=org.zaidinvergeles -DartifactId=tomcat-war
-deployment -DarchetypeArtifactId=maven-archetype-webapp -DinteractiveMode=fa
```

Podéis sustituir los valores de **groupId** y **artifactId** (este será el nombre de la aplicación) por lo que queráis.

Tras muchas líneas veremos:

```
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
```

Se habrá creado un directorio llamado **tomcat-war-deployment**. Entraremos en él:

```
$ cd tomcat-war-deployment
```

### 3.2.4. POM

Ahora debemos modificar el **POM** del proyecto para que haga referencia a que el despliegue se realice con el plugin de Maven para Tomcat.



No existen plugins **oficiales** para Tomcat más allá de la versión 7 del servidor. No obstante, el plugin para Tomcat 7 sigue funcionando correctamente con Tomcat 9. Otra opción sería utilizar el plugin [Cargo](#)

Fichero **pom.xml**

```
<project xmlns="http://maven.apache.org/POM/4.0.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/maven-
v4_0_0.xsd">
  <modelVersion>4.0.0</modelVersion>
  <groupId>org.zaidinvergeles</groupId>
  <artifactId>tomcat-war-deployment</artifactId>
  <packaging>war</packaging>
  <version>1.0-SNAPSHOT</version>
  <name>tomcat-war-deployment Maven Webapp</name>
  <url>http://maven.apache.org</url>
  <dependencies>
    <dependency>
      <groupId>junit</groupId>
      <artifactId>junit</artifactId>
      <version>3.8.1</version>
      <scope>test</scope>
    </dependency>
```

```
</dependencies>
<build>
  <finalName>tomcat-war-deployment</finalName> ①
</build>
</project>
```

① Modificaremos esta parte

Donde lo que añadimos es el bloque

```
<build>
  <finalName>tomcat-war-deployment</finalName> ①
  <plugins>
    <plugin>
      <groupId>org.apache.tomcat.maven</groupId>
      <artifactId>tomcat7-maven-plugin</artifactId>
      <version>2.2</version>
      <configuration>
        <url>http://localhost:8080/manager/text</url> ②
        <server>Tomcat</server> ③
        <path>/despliegue</path> ④
      </configuration>
    </plugin>
  </plugins>
</build>
```

① Nombre final del ejecutable .jar que se va a generar

② URL del servidor Tomcat donde se hará el despliegue. Como en nuestro caso Maven y Tomcat están en el mismo servidor, la URL corresponde a localhost. Esta URL debe ir seguida por `/manager/text`, tal y como leemos en la [documentación del plugin](#)

③ Nombre del server donde se va a desplegar la aplicación. El nombre debe ser consistente con lo que hayamos puesto en el `settings.xml` del paso anterior.

④ Nombre que la aplicación utilizará en el path de la URL

### 3.3. Despliegue

Tras generar esta aplicación, los comandos finales que se utilizan en Maven para desplegar, volver a desplegar o desplegar una aplicación, son:

```
mvn tomcat7:deploy
mvn tomcat7:redploy
mvn tomcat7:undeploy
```

Así pues, tras el despliegue con Maven nos indicará que todo ha ido correctamente con un mensaje de **BUILD SUCCESS**, tal que así:

```
default: [INFO] Deploying war to http://localhost:8080/despliegue
default: Uploading: http://localhost:8080/manager/text/deploy?path=%2Fmyapp
Uploaded: http://localhost:8080/manager/text/deploy?path=%2Fmyapp (3 KB at 334.1
KB/sec)
```

Y, accediendo a través de la GUI, debemos ver que la aplicación está desplegado y que podemos acceder a ella perfectamente:

<http://servidor:8080/despliegue>

## 4. Tarea

Realizar el despliegue con la aplicación de prueba.

Repetir el despliegue pero esta vez con otra aplicación que no es la de prueba. Más adelante ya hablaremos de **git** pero de momento, usaremos los comandos que veremos a continuación.

Nos clonamos el repositorio:

```
git clone https://github.com/cameronmcnz/rock-paper-scissors.git
```

Nos situamos dentro de él:

```
cd rock-paper-scissors
```

Y cambiamos de rama:

```
git checkout patch-1
```

Tras esto debemos proceder exactamente igual que en el caso anterior, con la ventaja de que ya tenemos configurados los usuarios de Tomcat y los parámetros de Maven.

Así pues, sólo habría que añadir el bloque `<plugin>...</plugin>` adecuado para poder hacer nuestro despliegue.



Documenta, incluyendo capturas de pantallas, el proceso que has seguido para realizar el despliegue de esta nueva aplicación, así como el resultado final.

## 5. Cuestiones

Habéis visto que los archivos de configuración que hemos tocado contienen contraseñas en texto plano, por lo que cualquiera con acceso a ellos obtendría las credenciales de nuestras herramientas.

En principio esto representa un gran riesgo de seguridad, ¿sabrías razonar o averiguar por qué esto



está diseñado de esta forma?

## 6. Referencias

1. [Tutorial Tomcat I](<https://www.how2shout.com/linux/install-apache-tomcat-10-on-debian-11-linux/>)
2. [Tutorial Tomcat II](<https://www.linuxtechi.com/how-to-install-apache-tomcat-on-debian/>)
3. [Tutorial Tomcat para Ubuntu](<https://www.digitalocean.com/community/tutorials/how-to-install-apache-tomcat-10-on-ubuntu-20-04>)
4. [Instalación Maven](<https://www.how2shout.com/linux/how-to-install-apache-maven-on-debian-11-bullseye/>)
5. [JSF 3.0 en Tomcat 10 con Java 11](<https://www.nestoralmeida.com/jsf-3-0-en-tomcat-10-con-java-11/#9-renombrar-javax-a-jakarta>)
6. [Migración de Java 8 a Java 11](<https://www.adictosaltrabajo.com/2019/05/21/migracion-de-java-8-a-java-11/>)
7. [Install and configure jdk11 + Tomcat + Maven under Linux system](<https://developpaper.com/install-and-configure-jdk11-tomcat-maven-under-linux-system/>)
8. [Step-by-step Maven Tomcat WAR file deploy example](<https://www.theserverside.com/video/Step-by-step-Maven-Tomcat-WAR-file-deploy-example>)
9. [How to deploy the java application to Tomcat 9 webserver using Maven](<https://devopspilot.com/content/tomcat/tutorials/how-to-deploy-java-application-to-tomcat-using-maven.html>)
10. [How to Install Apache Maven on Debian 11 Bullseye](<https://www.how2shout.com/linux/how-to-install-apache-maven-on-debian-11-bullseye/>)
11. [How to Deploy a WAR File to Tomcat](<https://www.baeldung.com/tomcat-deploy-war>)
12. [Migrate Maven Projects to Java 11](<https://winterbe.com/posts/2018/08/29/migrate-maven-projects-to-java-11-jigsaw/>)
13. [How to configure Tomcat 9.0 in Maven](<https://programmer.group/how-to-configure-tomcat-9.0-in-maven.html>)
14. [Github: cameronmcnz/rock-paper-scissors](<https://github.com/cameronmcnz/rock-paper-scissors>)
15. [Why are plain text passwords in the config files?](<https://cwiki.apache.org/confluence/display/TOMCAT/Password>)
16. [How to avoid storing passwords in the clear for tomcat's server.xml Resource definition of a DataSource?](<https://stackoverflow.com/questions/129160/how-to-avoid-storing-passwords-in-the-clear-for-tomcats-server-xml-resource-def>)

## 7. Evaluación

Criterio	Puntuación
----------	------------

Despliegue manual de un <b>.war</b> en Tomcat correcto y bien documentado	<b>1 punto</b>
Despliegue del <b>.war</b> de prueba utilizando maven correcto y bien documentado	<b>0.25 puntos</b>
Despliegue con maven desde un repositorio de Github correcto y bien documentado	<b>4.75 puntos</b>
Cuestiones	<b>2 puntos</b>
Se ha prestado especial atención al formato del documento, utilizando la plantilla actualizada y haciendo un correcto uso del lenguaje técnico	<b>2 puntos</b>