


# PRÁCTICA 1

## Despliegue de una aplicación con Tomcat y Maven

En esta primera práctica de la unidad vamos a realizar el despliegue de una aplicación java (archivo con extensión .war) usando el servidor de aplicaciones Tomcat.

Lo primero que haremos será realizar la configuración e instalación de Tomcat en nuestro sistema Debian.

1. Vamos a instalar el cortafuegos y abrir el puerto por el que Tomcat trabaja:  
*sudo apt install ufw, sudo ufw allow 8080*

A screenshot of a terminal window titled 'debian@debian: ~'. The terminal shows the execution of two commands: 'sudo apt install ufw' and 'sudo ufw allow 8080'. The output for the first command shows that ufw is already installed and up-to-date. The output for the second command shows that the rule for port 8080 is being skipped because it already exists.

```
debian@debian:~$ sudo apt install ufw
[sudo] contraseña para debian:
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
ufw ya está en su versión más reciente (0.36.2-1).
0 actualizados, 0 nuevos se instalarán, 0 para eliminar y 0 no actualizados.
debian@debian:~$ sudo ufw allow 8080
Skipping adding existing rule
Skipping adding existing rule (v6)
debian@debian:~$
```

2. Actualizamos los repositorios e instalamos java: *sudo apt update, sudo apt install default-jdk*

```
debian@debian: ~
Archivo  Editar  Pestañas  Ayuda

debian@debian:~$ sudo apt update
Des:1 http://security.debian.org/debian-security bookworm-security InRelease [48,0 kB]
Obj:2 http://deb.debian.org/debian bookworm InRelease
Des:3 http://deb.debian.org/debian bookworm-updates InRelease [52,1 kB]
Des:4 http://security.debian.org/debian-security bookworm-security/main Sources [60,8 kB]
Des:5 http://security.debian.org/debian-security bookworm-security/non-free-firmware Sources [796 B]
Des:6 http://security.debian.org/debian-security bookworm-security/main amd64 Packages [103 kB]
Des:7 http://security.debian.org/debian-security bookworm-security/main Translation-en [62,5 kB]
Des:8 http://security.debian.org/debian-security bookworm-security/non-free-firmware amd64 Packages [688 B]
Descargados 328 kB en 1s (643 kB/s)
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
Se pueden actualizar 7 paquetes. Ejecute «apt list --upgradable» para verlos.
debian@debian:~$
```

```
debian@debian: ~
Archivo  Editar  Pestañas  Ayuda

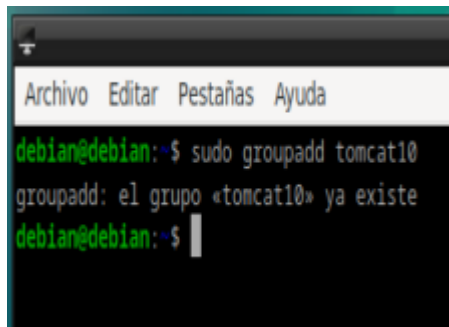
debian@debian:~$ sudo apt install default-jdk
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
default-jdk ya está en su versión más reciente (2:1.17-74).
0 actualizados, 0 nuevos se instalarán, 0 para eliminar y 7 no actualizados.
debian@debian:~$
```

3. Instalamos Tomcat 10: *sudo apt install tomcat10*

```
debian@debian: ~
Archivo  Editar  Pestañas  Ayuda

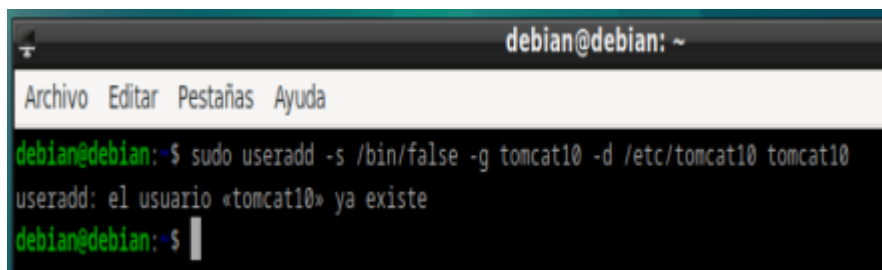
debian@debian:~$ sudo apt install tomcat10
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
tomcat10 ya está en su versión más reciente (10.1.6-1+deb12u1).
0 actualizados, 0 nuevos se instalarán, 0 para eliminar y 0 no actualizados.
debian@debian:~$
```

4. Creamos un grupo de usuarios para usar Tomcat: *sudo groupadd tomcat10*



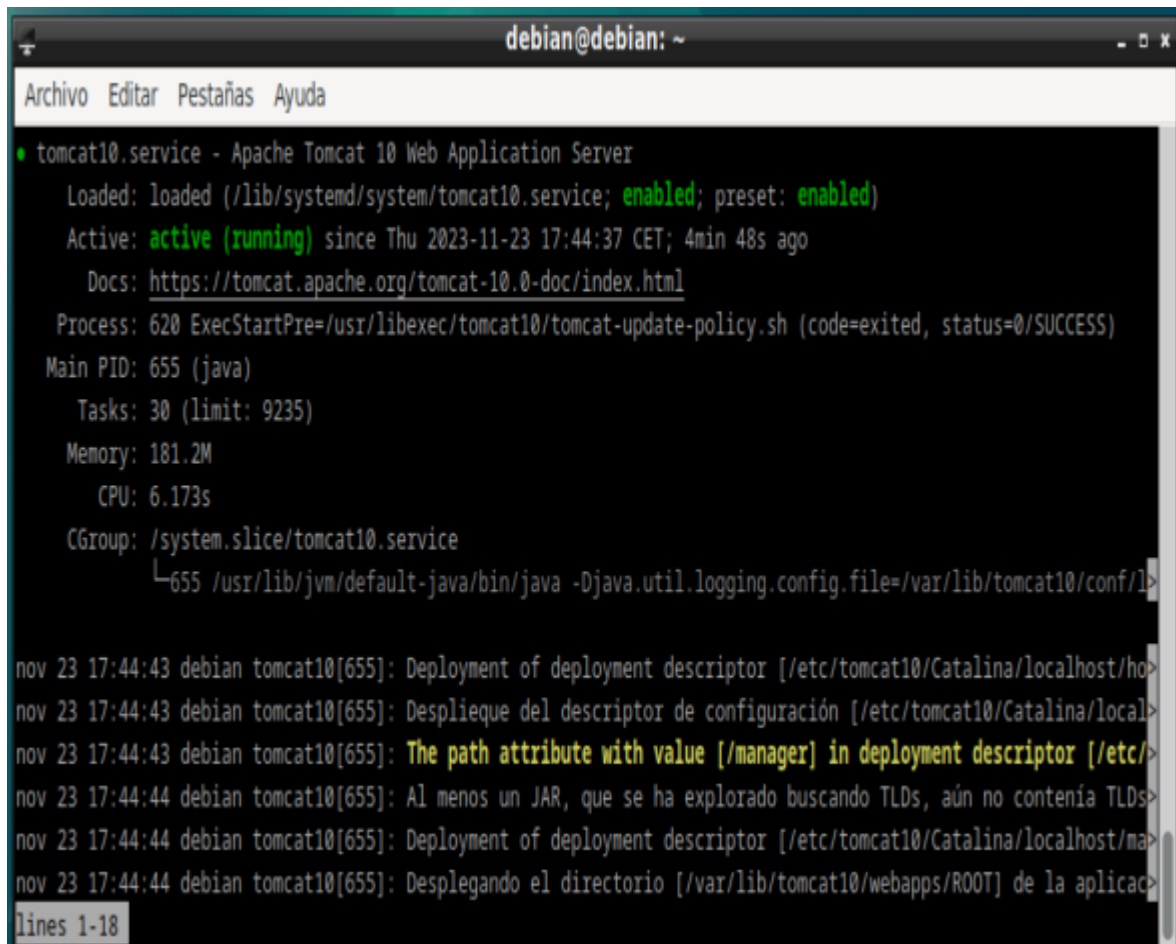
```
debian@debian:~$ sudo groupadd tomcat10
groupadd: el grupo «tomcat10» ya existe
debian@debian:~$
```

5. Creamos un usuario: *sudo useradd -s /bin/false -g tomcat10 -d /etc/tomcat10 tomcat10*



```
debian@debian:~$ sudo useradd -s /bin/false -g tomcat10 -d /etc/tomcat10 tomcat10
useradd: el usuario «tomcat10» ya existe
debian@debian:~$
```

6. Iniciamos y vemos si está corriendo: *sudo systemctl start tomcat10*, *sudo systemctl status tomcat10*. Ve a localhost:8080 y haz una captura del resultado (se debe indicar que tomcat está funcionando correctamente).



```
debian@debian:~$ sudo systemctl status tomcat10
● tomcat10.service - Apache Tomcat 10 Web Application Server
   Loaded: loaded (/lib/systemd/system/tomcat10.service; enabled; preset: enabled)
   Active: active (running) since Thu 2023-11-23 17:44:37 CET; 4min 48s ago
     Docs: https://tomcat.apache.org/tomcat-10.0-doc/index.html
   Process: 620 ExecStartPre=/usr/libexec/tomcat10/tomcat-update-policy.sh (code=exited, status=0/SUCCESS)
   Main PID: 655 (java)
      Tasks: 30 (limit: 9235)
     Memory: 181.2M
        CPU: 6.173s
    CGroup: /system.slice/tomcat10.service
            └─655 /usr/lib/jvm/default-java/bin/java -Djava.util.logging.config.file=/var/lib/tomcat10/conf/l>

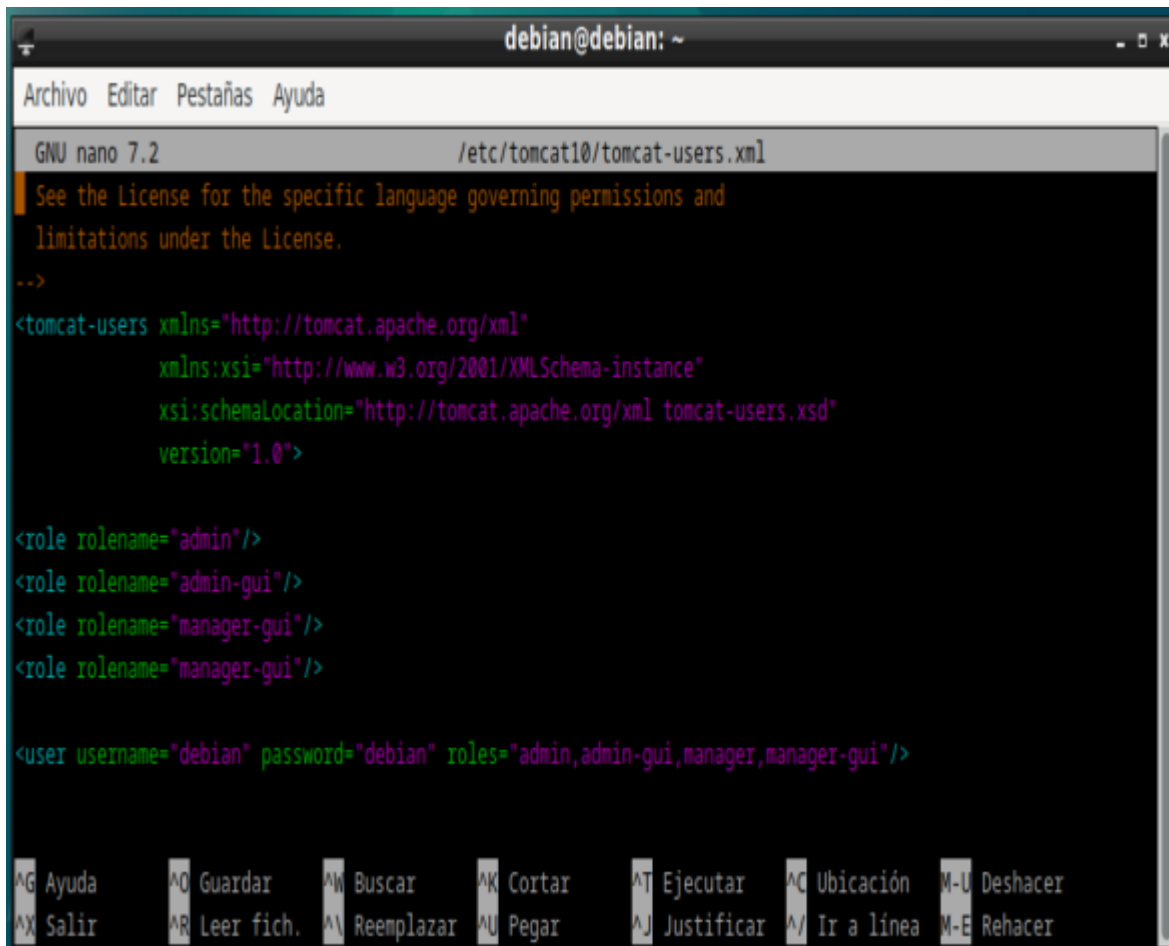
nov 23 17:44:43 debian tomcat10[655]: Deployment of deployment descriptor [/etc/tomcat10/Catalina/localhost/ho>
nov 23 17:44:43 debian tomcat10[655]: Despliegue del descriptor de configuración [/etc/tomcat10/Catalina/local>
nov 23 17:44:43 debian tomcat10[655]: The path attribute with value [/manager] in deployment descriptor [/etc/>
nov 23 17:44:44 debian tomcat10[655]: Al menos un JAR, que se ha explorado buscando TLDs, aún no contenía TLDs>
nov 23 17:44:44 debian tomcat10[655]: Deployment of deployment descriptor [/etc/tomcat10/Catalina/localhost/ma>
nov 23 17:44:44 debian tomcat10[655]: Desplegando el directorio [/var/lib/tomcat10/webapps/ROOT] de la aplicac>
lines 1-18
```

7. Vamos a dar acceso a un usuario al servidor de aplicaciones, para ello modificaremos el archivo de configuración: `sudo nano /etc/tomcat10/tomcat-users.xml`

8. Al final del archivo, hemos de añadir las siguientes líneas:

```
<tomcat-users>
<role rolename="admin"/>
<role rolename="admin-gui"/>
<role rolename="manager"/>
<role rolename="manager-gui"/>
```

```
<user username="tu_usuario" password="tu_contraseña"
roles="admin,admin-gui,manager,manager-gui"/>
</tomcat-users>
```



```
debian@debian: ~
Archivo Editar Pestañas Ayuda
GNU nano 7.2 /etc/tomcat10/tomcat-users.xml
See the License for the specific language governing permissions and
limitations under the License.
...>
<tomcat-users xmlns="http://tomcat.apache.org/xml"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://tomcat.apache.org/xml tomcat-users.xsd"
  version="1.0">

  <role rolename="admin"/>
  <role rolename="admin-gui"/>
  <role rolename="manager-gui"/>
  <role rolename="manager-gui"/>

  <user username="debian" password="debian" roles="admin,admin-gui,manager,manager-gui"/>

Ayuda  Guardar  Buscar  Cortar  Ejecutar  Ubicación  M-U  Deshacer
Salir  Leer fich.  Reemplazar  Pegar  Justificar  Ir a línea  M-E  Rehacer
```

9. Vamos a instalar el administrador web: `sudo apt install tomcat10-admin`

```

debian@debian: ~
Archivo Editar Pestañas Ayuda

debian@debian:~$ sudo apt install tomcat10-admin
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
tomcat10-admin ya está en su versión más reciente (10.1.6-1+deb12u1).
0 actualizados, 0 nuevos se instalarán, 0 para eliminar y 7 no actualizados.
debian@debian:~$

```

10. Accede ahora a localhost:8080/manager/html, y muestra una captura de la interfaz gráfica una vez te has autenticado.

**Gestor de Aplicaciones Web de Tomcat**

Mensaje: OK

**Gestor**

[Listar Aplicaciones](#) [Ayuda HTML de Gestor](#) [Ayuda de Gestor](#) [Estado de Servidor](#)

Aplicaciones					
Ruta	Versión	Nombre a Mostrar	Ejecutándose	Sesiones	Comandos
/	Ninguno especificado		true	0	Arrancar Parar Recargar Replegar Expirar sesiones sin trabajar ≥ 30 minutos
/host-manager	Ninguno especificado	Tomcat Host Manager Application	true	0	Arrancar Parar Recargar Replegar Expirar sesiones sin trabajar ≥ 30 minutos
/manager	Ninguno especificado	Tomcat Manager Application	true	1	Arrancar Parar Recargar Replegar Expirar sesiones sin trabajar ≥ 30 minutos

**Desplegar**

Desplegar directorio o archivo WAR localizado en servidor

Trayectoria de Contexto (opcional):

Version (for parallel deployment):

11. Mira cómo hacer el despliegue y aplícalo al archivo .war que hay en la Práctica 1 de esta Unidad.
12. Añade una captura de la aplicación desplegada y otra del aspecto de la tabla de aplicaciones de la interfaz de Tomcat.



## Sample "Hello, World" Application

This is the home page for a sample application used to illustrate the source directory organization of a web application utilizing the principles outlined in the Application Developer's Guide.

To prove that they work, you can execute either of the following links:

- To a [JSP page](#).
- To a [servlet](#).

Ahora vamos a utilizar Maven para realizar la compilación y el despliegue de un proyecto web en Tomcat.

1. Instalamos Maven: *sudo apt update*, *sudo apt install maven*

```
debian@debian:~$ sudo apt update
Obj:1 http://security.debian.org/debian-security bookworm-security InRelease
Obj:2 http://deb.debian.org/debian bookworm InRelease
Obj:3 http://deb.debian.org/debian bookworm-updates InRelease
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
Se pueden actualizar 7 paquetes. Ejecute «apt list --upgradable» para verlos.
debian@debian:~$ sudo apt install maven
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes adicionales:
  libaopalliance-java libapache-pom-java libatinject-jsr330-api-java libcdi-api-java libcommons-cli-java
  libcommons-io-java libcommons-lang3-java libcommons-parent-java liberror-prone-java
  libgeronimo-annotation-1.3-spec-java libgeronimo-interceptor-3.0-spec-java libguava-java libguice-java
  libjansi-java libjsr305-java libmaven-parent-java libmaven-resolver-java libmaven-shared-utils-java
  libmaven3-core-java libplexus-cipher-java libplexus-classworlds-java libplexus-component-annotations-java
```

2. Vemos que se ha instalado: *mvn --v*

A continuación debemos realizar una serie de configuraciones de manera que Maven se pueda autenticar en Tomcat para realizar los despliegues. Debemos utilizar el rol manager-script para poder realizar despliegues con Maven a través de la línea de comandos.

```

debian@debian:~$ mvn --v
Apache Maven 3.8.7
Maven home: /usr/share/maven
Java version: 17.0.9, vendor: Debian, runtime: /usr/lib/jvm/java-17-openjdk-amd64
Default locale: es_ES, platform encoding: UTF-8
OS name: "linux", version: "6.1.0-13-amd64", arch: "amd64", family: "unix"
debian@debian:~$

```

3. Vamos a crearnos un nuevo usuario que tenga este rol de manager-script en el archivo que configuramos en el punto 8. Debes añadir lo siguiente al archivo:

```

<role rolename="manager-script"/>
<user username="tu_usuario_d" password="tu_contraseña"
roles="manager-script"/>

```

```

<role rolename="manager-script"/>
<user username="debian" password="debian" roles="manager-script"/>

```

4. Ahora debemos editar el archivo de configuración de Maven, localizado en `/etc/maven/settings.xml` para indicarle el servidor sobre el que se tiene que realizar los despliegues (Tomcat). Debe quedar un bloque similar al de la imagen:

```

<servers>
  <server>
    <id>mi-servidor-Tomcat</id>
    <username>jesus-d</username>
    <password>xxx</password>
  </server>
</servers>

```

```

<server>
  <id>mi-servidor-Tomcat</id>
  <username>alejandro-perinan</username>
  <password>debian</password>
</server>

```

5. A continuación vamos a clonar un proyecto de un repositorio de GitHub. Para lo que ejecutaremos el siguiente comando: `git clone https://github.com/cameronmcnz/rock-paper-scissors.git`



```

debian@debian:~$ git clone https://github.com/cameronmcnz/rock-paper-scissors.git
Clonando en 'rock-paper-scissors'...
remote: Enumerating objects: 999, done.
remote: Counting objects: 100% (28/28), done.
remote: Compressing objects: 100% (19/19), done.
remote: Total 999 (delta 11), reused 22 (delta 7), pack-reused 971
Recibiendo objetos: 100% (999/999), 29.14 MiB | 4.77 MiB/s, listo.
Resolviendo deltas: 100% (219/219), listo.
debian@debian:~$

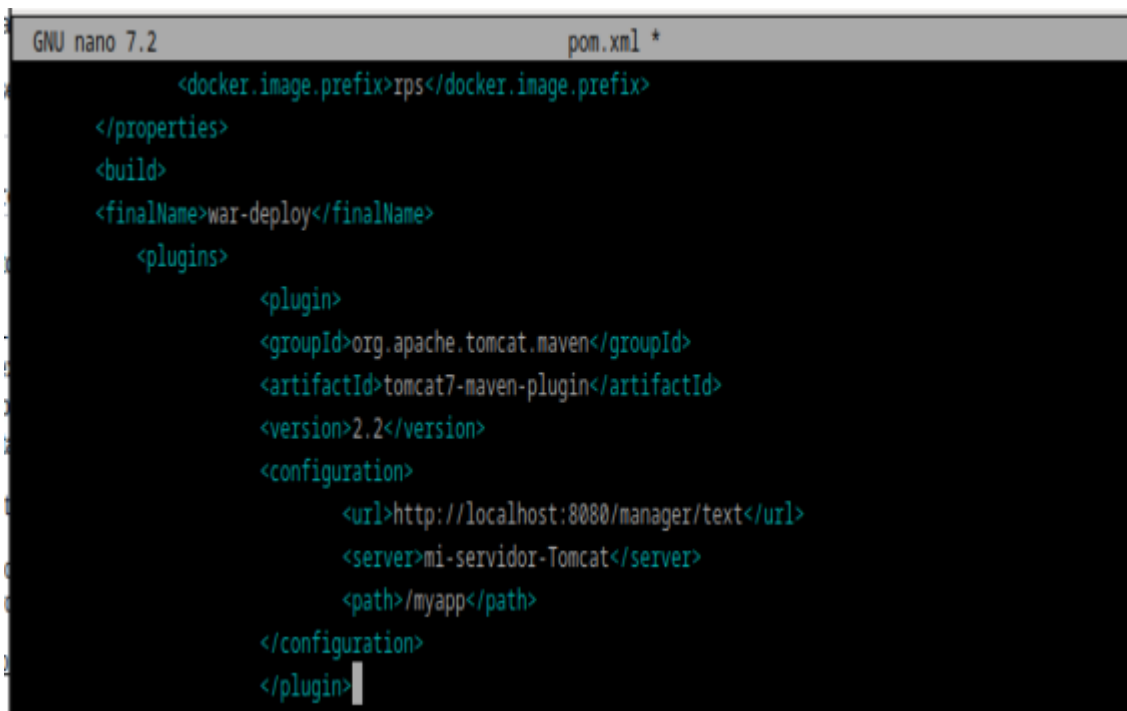
```

6. Si nos situamos dentro del directorio que se ha generado (cd rock-paper-scissors), veremos que hay un archivo pom.xml. Este es el archivo que debemos configurar para poder realizar el despliegue del proyecto Maven sobre Tomcat.
7. Dentro del pom.xml, debemos añadir el plugin correspondiente de Tomcat. Lo haremos dentro del bloque “pugins”, que a su vez está dentro del bloque “build”. Debe quedar algo similar a la imagen.

```

<build>
  <finalName>war-deploy</finalName>
  <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>
        <server>mi-servidor-Tomcat</server>
        <path>/myapp</path>
      </configuration>
    </plugin>
  </plugins>
</build>

```



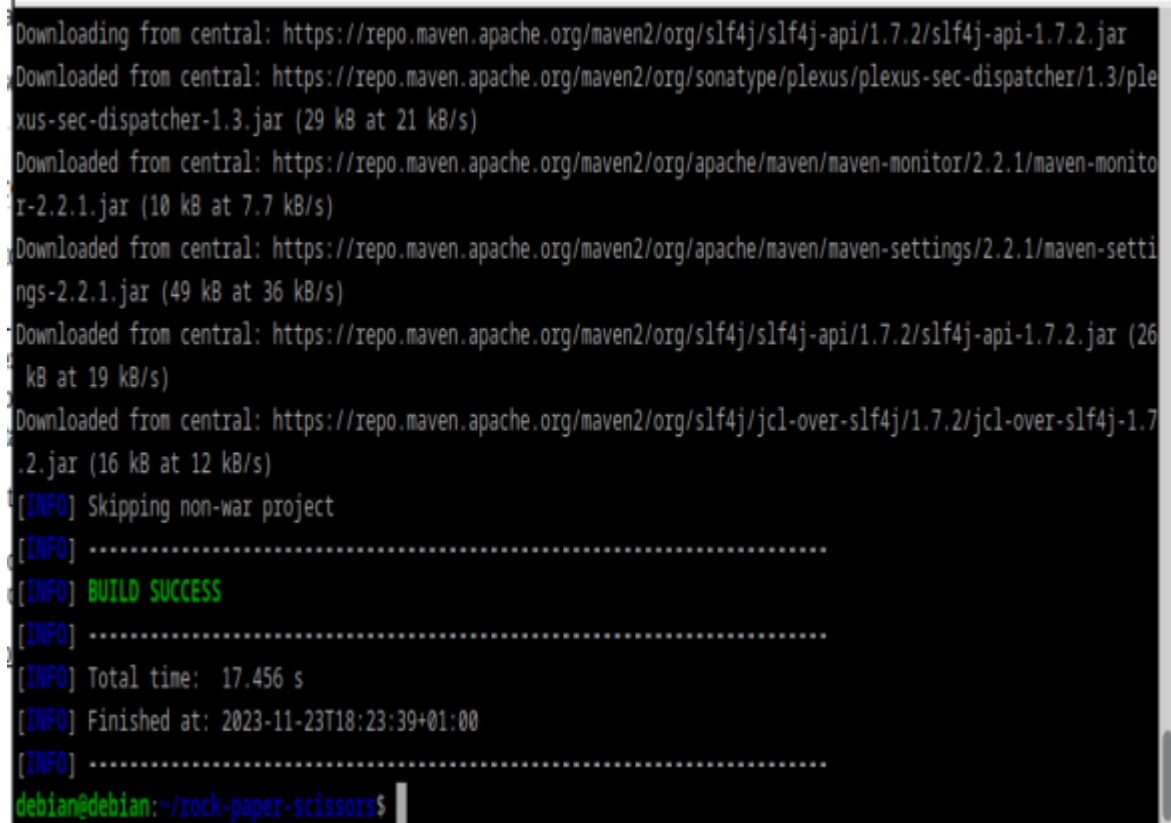
```

GNU nano 7.2                                pom.xml *
      <docker.image.prefix>rps</docker.image.prefix>
    </properties>
    <build>
      <finalName>war-deploy</finalName>
      <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>
            <server>mi-servidor-Tomcat</server>
            <path>/myapp</path>
          </configuration>
        </plugin>
      </plugins>
    </build>
  </project>

```



8. Indica a qué hacen referencia las líneas “server” y “path” dentro del bloque “configuration”. Mira un poco la estructura del pom e indica qué otro bloque se incluye.
9. Una vez realizada toda la configuración, podemos realizar la compilación y el despliegue del proyecto con este comando: *mvn tomcat7:deploy*
10. Muestra una captura con el mensaje que aparece en consola tras ejecutar el comando anterior.



```
Downloading from central: https://repo.maven.apache.org/maven2/org/slf4j/slf4j-api/1.7.2/slf4j-api-1.7.2.jar
Downloaded from central: https://repo.maven.apache.org/maven2/org/sonatype/plexus/plexus-sec-dispatcher/1.3/plexus-sec-dispatcher-1.3.jar (29 kB at 21 kB/s)
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/maven-monitor/2.2.1/maven-monitor-2.2.1.jar (10 kB at 7.7 kB/s)
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/maven-settings/2.2.1/maven-settings-2.2.1.jar (49 kB at 36 kB/s)
Downloaded from central: https://repo.maven.apache.org/maven2/org/slf4j/slf4j-api/1.7.2/slf4j-api-1.7.2.jar (26 kB at 19 kB/s)
Downloaded from central: https://repo.maven.apache.org/maven2/org/slf4j/jcl-over-slf4j/1.7.2/jcl-over-slf4j-1.7.2.jar (16 kB at 12 kB/s)
[INFO] Skipping non-war project
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 17.456 s
[INFO] Finished at: 2023-11-23T18:23:39+01:00
[INFO] -----
debian@debian:~/rock-paper-scissors$
```

11. ¿Se ha podido realizar el despliegue? ¿Por qué?  
Si, pone BUILD SUCCESS.
12. Prueba a usar el siguiente proyecto: *git clone <https://github.com/koddas/war-web-project>*

```

debian@debian:~/rock-paper-scissors$ git clone https://github.com/koddas/war-web-project
Clonando en 'war-web-project'...
remote: Enumerating objects: 140, done.
remote: Counting objects: 100% (51/51), done.
remote: Compressing objects: 100% (12/12), done.
remote: Total 140 (delta 43), reused 39 (delta 39), pack-reused 89
Recibiendo objetos: 100% (140/140), 20.38 KiB | 496.00 KiB/s, listo.
Resolviendo deltas: 100% (68/68), listo.
debian@debian:~/rock-paper-scissors$

```

13. Ahora modifica el pom correspondiente añadiendo lo siguiente:

```

<plugin>
  <groupId>org.apache.maven.plugins</groupId>
  <artifactId>maven-war-plugin</artifactId>
  <version>3.3.2</version>
  <configuration>
  </configuration>
</plugin>
<plugin>
  <groupId>org.apache.tomcat.maven</groupId>
  <artifactId>tomcat7-maven-plugin</artifactId>
  <version>2.2</version>
  <configuration>
    <url>http://localhost:8080/manager/text</url>
    <server>mi-servidor-Tomcat</server>
    <path>/webapp</path>
  </configuration>
</plugin>

```

```

<plugin>
  <groupId>org.apache.tomcat.maven</groupId>
  <artifactId>tomcat7-maven-plugin</artifactId>
  <version>2.2</version>
  <configuration>
    <url>http://localhost:8080/manager/text</url>
    <server>mi-servidor-Tomcat</server>
    <path>/myapp</path>
  </configuration>
</plugin>

```

```
<plugin>
<groupId>org.apache.maven.plugins</groupId>
<artifactId>maven-war-plugin</artifactId>
<version>3.3.2</version>
<configuration>
</configuration>
</plugin>
```

14. Comprueba si ahora se puede desplegar la aplicación web. Intenta abrirla desde la interfaz de Tomcat. Incluye una captura del aspecto de la aplicación desplegada.
- No se puede.