

## Creación de un archivo WAR para Tomcat

---

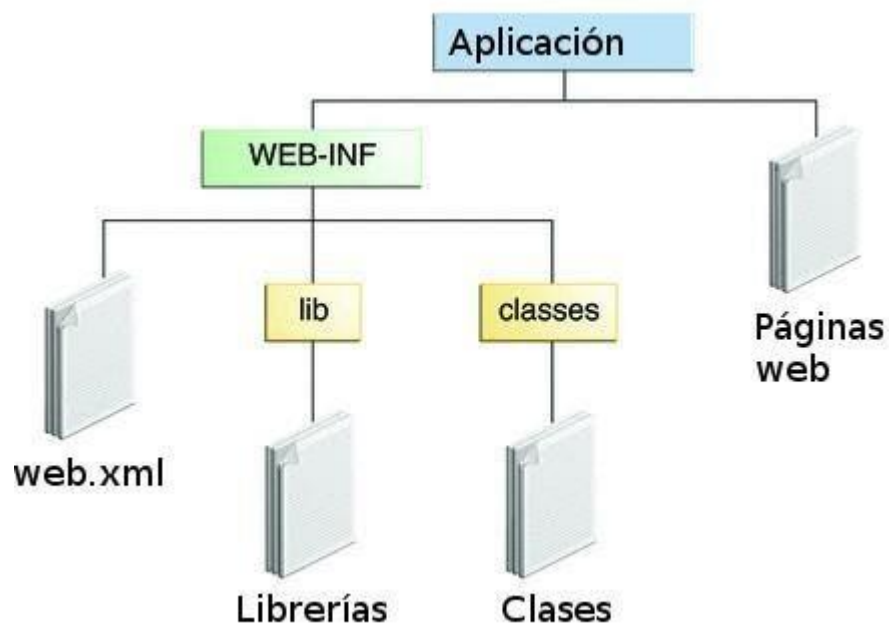
El servidor de aplicaciones Tomcat nos puede mostrar una serie de ejemplos instalando su paquete correspondiente:

```
sudo apt-get install tomcat9-examples
```

Dentro del sistema de archivos los ejemplos quedan instalados en `/usr/share/tomcat9-examples/examples` y podemos verlos y ejecutarlos desde el *manager webapp*.

```
localhost:8080
```

Entre estos ejemplos nos encontramos con servlets, aplicaciones JSP, JavaScript..., pero si investigamos en el sistema de archivos podemos ver que estas aplicaciones y cualquier otra desplegada en Tomcat debe cumplir con una estructura de directorios tal como muestra la siguiente imagen:



Para conseguir tal estructura debemos ejecutar las siguientes órdenes:

```
mkdir ejemplo
mkdir ejemplo/WEB-INF
mkdir ejemplo/WEB-INF/classes
```

```
mkdir ejemplo/WEB-INF/lib
```

Podemos crearlo en cualquier directorio porque luego se desplegará con *manager webapp*.

Suponiendo que *ejemplo* sea el nombre de la aplicación.

A continuación vamos a crear una sencilla aplicación en un archivo llamado *index.jsp* dentro del directorio *ejemplo* con el siguiente contenido:

```
<html>
<head><title>EJEMPLO DE APLICACIÓN WEB</title>
<script language="Javascript">
    function popup(){ alert("Ejemplo de la UT3");}

</script></head>
<body><h1 align=center>DESPLIEGUE DE APLICACIONES WEB</h1>
<div align=center>
<form><input type="button" value="UNIDAD 3"
    onclick="popup()" "></form>
</body></html>
```

El siguiente paso sería crear el archivo *web.xml* dentro del directorio indicado anteriormente. Este archivo es un XML con el descriptor del despliegue, y en una de sus formas más sencillas podría ser algo así:

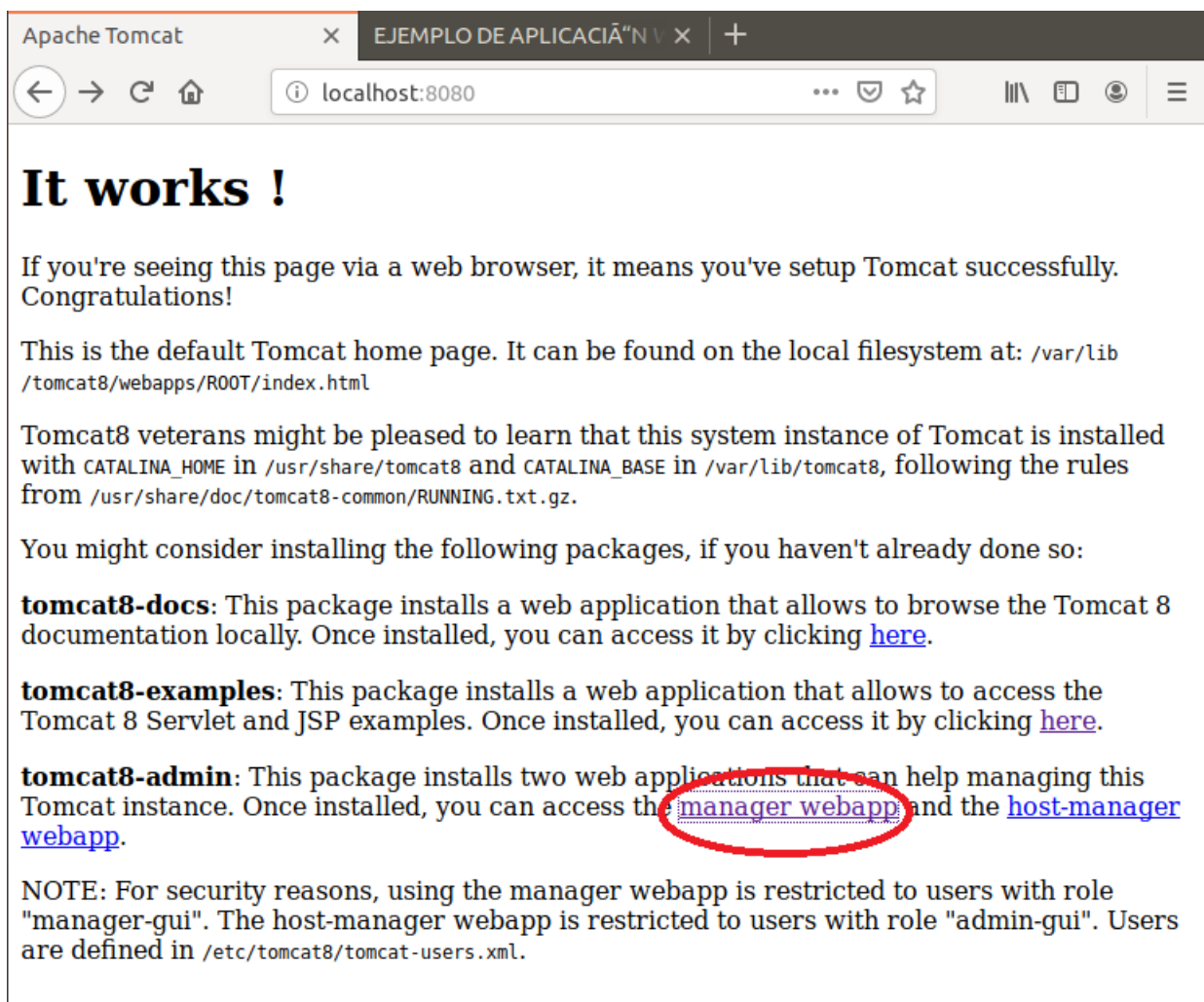
```
<?xml version="1.0" encoding="ISO-8859-1"?>
<web-app xmlns="http://java.sun.com/xml/ns/j2ee"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
http://java.sun.com/xml/ns/javaee/web-app_2_5.xsd" version="2.5">
    <display-name>Descriptor Aplicacion ejemplo</display-name>
    <description>
        Un descriptor web.xml muy sencillo.
    </description>
</web-app>
```

Al terminar de crear nuestra aplicación podemos crear el archivo WAR compilando de la

siguiente manera desde dentro de la carpeta ejemplo:

```
sudo jar cvf ejemplo.war .
```

En este momento ya tenemos el archivo WAR que podemos desplegarlo mediante el *manager webapp*.



Apache Tomcat x EJEMPLO DE APLICACIÓN x +

← → ↺ 🏠 ⓘ localhost:8080 ... 📄 ⚙️ ☰

## It works !

If you're seeing this page via a web browser, it means you've setup Tomcat successfully. Congratulations!

This is the default Tomcat home page. It can be found on the local filesystem at: `/var/lib/tomcat8/webapps/ROOT/index.html`

Tomcat8 veterans might be pleased to learn that this system instance of Tomcat is installed with `CATALINA_HOME` in `/usr/share/tomcat8` and `CATALINA_BASE` in `/var/lib/tomcat8`, following the rules from `/usr/share/doc/tomcat8-common/RUNNING.txt.gz`.

You might consider installing the following packages, if you haven't already done so:

**tomcat8-docs:** This package installs a web application that allows to browse the Tomcat 8 documentation locally. Once installed, you can access it by clicking [here](#).

**tomcat8-examples:** This package installs a web application that allows to access the Tomcat 8 Servlet and JSP examples. Once installed, you can access it by clicking [here](#).

**tomcat8-admin:** This package installs two web applications that can help managing this Tomcat instance. Once installed, you can access the [manager webapp](#) and the [host-manager webapp](#).

NOTE: For security reasons, using the manager webapp is restricted to users with role "manager-gui". The host-manager webapp is restricted to users with role "admin-gui". Users are defined in `/etc/tomcat8/tomcat-users.xml`.

**Deploy**

**Deploy directory or WAR file located on server**

Context Path (required):

XML Configuration file URL:

WAR or Directory URL:

Deploy

**WAR file to deploy**

Select WAR file to upload

Browse...

No file selected.

Deploy

Lo desplegamos con manager y ya aparecerá en la lista de aplicaciones desplegadas.  
A continuación podremos acceder a ella:

