

INSTALACION DE APACHE TOMCAT 9 EN UBUNTU (VPS) Y CONFIGURACION

1º) Instalamos OpenJDK:

```
sudo apt install default-jdk
```

2º) Creamos un usuario y de camino el directorio para Tomcat

```
sudo useradd -m -U -d /opt/tomcat -s /bin/false tomcat
```

3º) Instalamos wget y unzip, se puede hacer sin unzip pero en los siguientes pasos tendremos que usar otro metodo de descompresión como el **tar**

```
sudo apt install unzip wget
```

4º) Nos vamos al directorio **/tmp** y usamos el **wget** para descargar tomcat en su ultima versión

```
cd /tmp
```

```
wget http://ftp.cixug.es/apache/tomcat/tomcat-9/v9.0.16/bin/apache-tomcat-9.0.16.zip
```

5º) Una vez descargado, lo extraemos y movemos los archivos extraidos a **opt/tomcat/**

```
unzip apache-tomcat-*.zip
```

```
sudo mv apache-tomcat-*/ /opt/tomcat/
```

6º) Tomcat necesita permisos en la carpeta **/opt/tomcat** y que los scripts dentro de **/opt/tomcat/bin/** sean ejecutables.

```
sudo chown -R tomcat: /opt/tomcat
```

```
sudo chmod +x /opt/tomcat/bin/*.sh
```

7º) Creamos un nuevo archivo de unidad llamado **tomcat.service** dentro del directorio **/etc/systemd/system/** con el siguiente contenido

```
cd /etc/systemd/system/
```

```
sudo nano tomcat.service
```

```
[Unit]
Description=Tomcat 9 servlet container
After=network.target

[Service]
Type=forking

User=tomcat
Group=tomcat

Environment="JAVA_HOME=/usr/lib/jvm/default-java"
Environment="JAVA_OPTS=-Djava.security.egd=file:///dev/urandom"

Environment="CATALINA_BASE=/opt/tomcat"
Environment="CATALINA_HOME=/opt/tomcat"
Environment="CATALINA_PID=/opt/tomcat/temp/tomcat.pid"
Environment="CATALINA_OPTS=-Xms512M -Xmx1024M -server -XX:+UseParallelGC"

ExecStart=/opt/tomcat/bin/startup.sh
ExecStop=/opt/tomcat/bin/shutdown.sh

[Install]
WantedBy=multi-user.target
```

8º) Notificaremos a systemd que creamos un nuevo archivo de unidad e iniciaremos el servicio Tomcat ejecutando, si no da errores habilitaremos el servicio Tomcat para que se inicie automáticamente en el arranque.

```
sudo systemctl daemon-reload
```

```
sudo systemctl start tomcat          // Iniciamos Tomcat
```

```
sudo systemctl status tomcat         // Comprobamos errores
```

```
sudo systemctl enable tomcat         // Habilitamos Tomcat en el arranque
```

9º) Abrimos el puerto 8080 en este caso para poder acceder remotamente a nuestro Tomcat

```
sudo ufw allow 8080/tcp
```

10º) Creamos un usuario que tenga acceso a la interfaz de administración web añadiendo las siguientes líneas a **/opt/tomcat/conf/tomcat-users.xml**

```
sudo nano /opt/tomcat/conf/tomcat-users.xml
```

```
<role rolename="admin-gui"/>
<role rolename="manager-gui"/>
<user username="admin" password="admin123" roles="admin-gui,manager-gui"/>
</tomcat-users>
```

11º) Permitimos el acceso a la interfaz de administración web desde el exterior comentando las siguientes líneas del archivo **/opt/tomcat/webapps/manager/META-INF/context.xml** y **/opt/tomcat/webapps/host-manager/META-INF/context.xml**

```
sudo nano /opt/tomcat/webapps/manager/META-INF/context.xml
```

```
sudo nano /opt/tomcat/webapps/host-manager/META-INF/context.xml
```

```
<!--
    BARBA
    <Valve className="org.apache.catalina.valves.RemoteAddrValve"
        allow="127\.\d+\.\d+\.\d+|::1|0:0:0:0:0:0:0:1" />
-->
```

12º) Ya está, lo unico que nos queda es probarlo accediendo a

<http://vps612580.ovh.net:8080/manager/html>

EJECUCION DE APLICACIONES WEB DE EJEMPLO

Seguimos las instrucciones que nos indica <http://vps612580.ovh.net:8080/docs/appdev/sample/>

`cd /opt/tomcat/webapps`

`wget http://vps612580.ovh.net:8080/docs/appdev/sample/sample.war`

Una vez hecho esto, vamos a nuestro ejemplo <http://vps612580.ovh.net:8080/samplePre/>

(Le he cambiado el nombre a **samplePre** por la siguiente actividad)

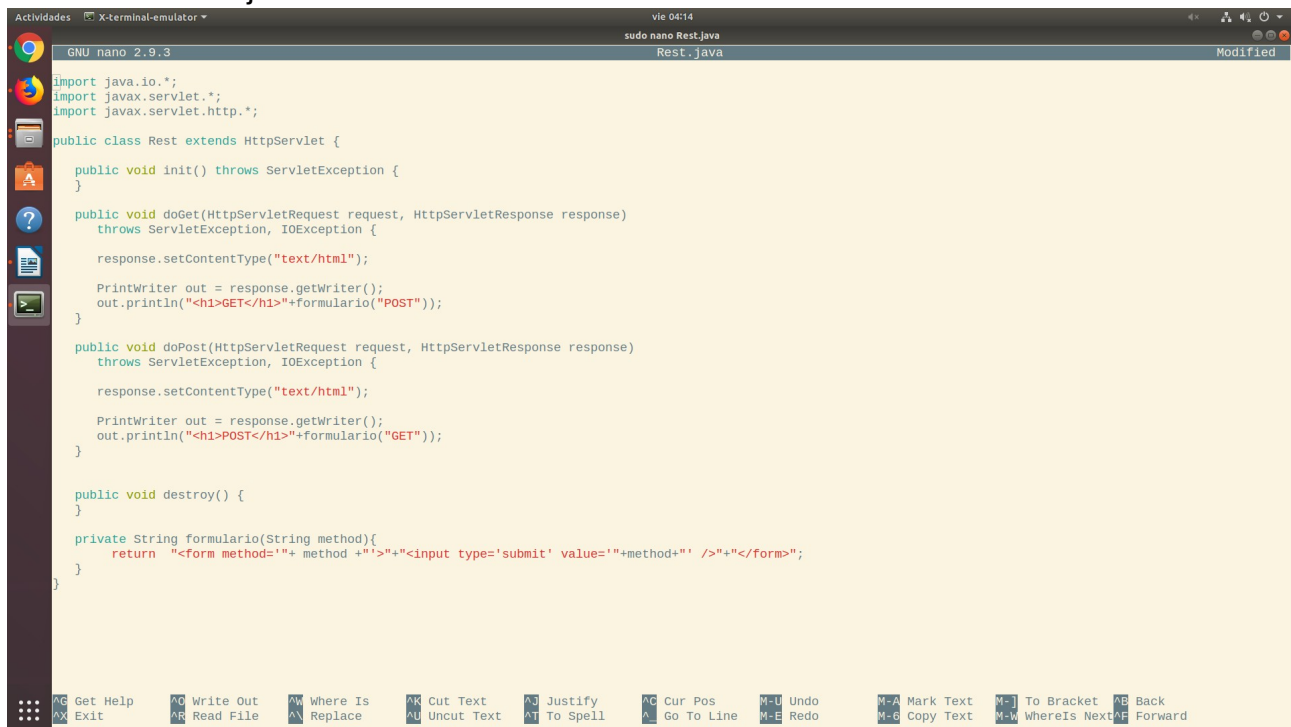
Tenemos mas webs de ejemplos en: <http://vps612580.ovh.net:8080/examples/>

CAMBIAR EL SERVLET DE EJEMPLO

(<http://vps612580.ovh.net:8080/sample/>) PARA QUE MUESTRE UN MENSAJE PERSONALIZADO POR EL ALUMNO ANTE UNA PETICION GET

`cd /opt/tomcat/webapps/classes/mypackage`

`sudo nano Hello.java`



```
GNU nano 2.9.3
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;

public class Rest extends HttpServlet {

    public void init() throws ServletException {
    }

    public void doGet(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        response.setContentType("text/html");
        PrintWriter out = response.getWriter();
        out.println("<h1>GET</h1>"+formulario("POST"));
    }

    public void doPost(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        response.setContentType("text/html");
        PrintWriter out = response.getWriter();
        out.println("<h1>POST</h1>"+formulario("GET"));
    }

    public void destroy() {
    }

    private String formulario(String method){
        return "<form method='"+ method +"'>"+<input type='submit' value='"+method+"' />"+</form>";
    }
}
```

`sudo javac -cp ./opt/tomcat/lib/servlet-api.jar Hello.java`

En este momento, me dí cuenta de que la resolución de nombres completamente cualificados estaba rota, por lo que moví el Hello.class a la carpeta /opt/tomcat/webapps/sample/WEB-INF/classes y modifiqué el archivo /opt/tomcat/webapps/sample/WEB-INF/web.xml quedando este de esta forma.



```
GNU nano 2.9.3
sudo nano web.xml
web.xml

<?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/j2ee http://java.sun.com/xml/ns/j2ee/web-app_2_4.xsd"
  version="2.4">

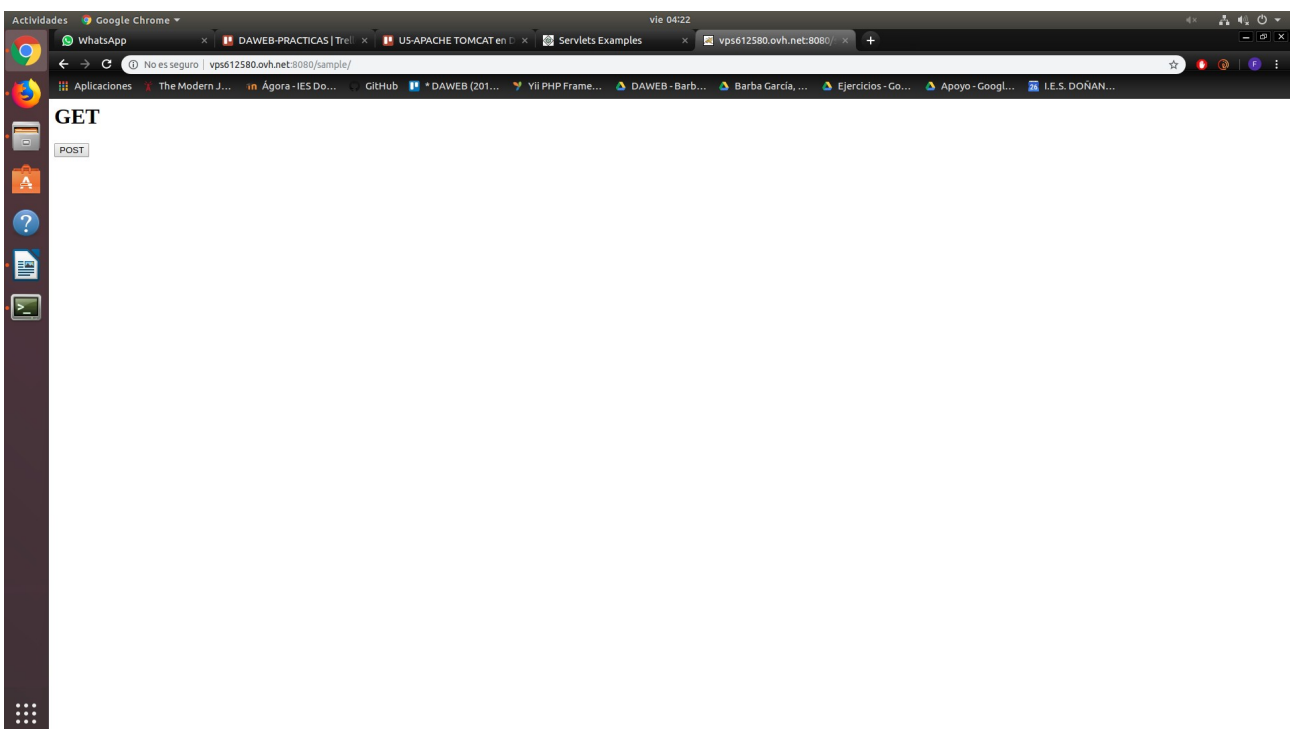
  <display-name>Hello, World Application</display-name>
  <description>
    This is a simple web application with a source code organization
    based on the recommendations of the Application Developer's Guide.
  </description>

  <servlet>
    <servlet-name>Hello</servlet-name>
    <servlet-class>Hello</servlet-class>
  </servlet>

  <servlet-mapping>
    <servlet-name>Hello</servlet-name>
    <url-pattern>*</url-pattern>
  </servlet-mapping>

</web-app>
```

Este seria el resultado, mostrando un titulo h1 con el contenido GET en caso de petición GET y POST en caso de petición POST



IDEM ANTERIOR ANTE UNA PETICION POST (MENSAJE DISTINTO)

En el codigo del ejercicio anterior ya está hecho.

