

# SERVIDOR WEB CON SOPORTE DE SERVLETS Y JSPS

Trabajo final CESI

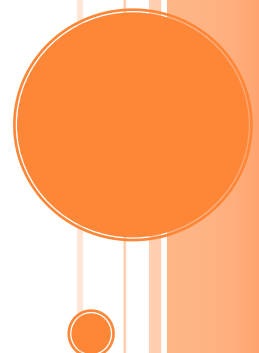
Martin Paez Anguita

Adil Hirchi

José María Martín Santaella

Grupo 17

Tipo 2 (21 horas de trabajo)



# 1 Índice General

1. Definición del Sistema .....	2
1.1    Objetivos y definición del sistema .....	2
1.1.1    Objetivos .....	2
1.1.2    Hardware .....	2
1.1.3    Software .....	3
1.1.4    Instalación y configuración.....	3
1.2    Servicios y sus posibles resultados .....	13
1.2.1    Servicios Tomcat .....	13
1.2.2    Posibles Resultados .....	14
1.3    Métricas .....	14
1.4    Parámetros .....	15
1.4.1    Limitaciones Hardware .....	15
1.4.2    Limitaciones Software.....	15
2    Evaluación del Sistema.....	16
2.1    Técnicas de evaluación .....	16
2.1.1    Herramientas para la evaluación .....	16
2.2    Carga de trabajo.....	17
2.3    Diseño de Experimentos.....	17
2.3.1    Configuración y subida de la Carga de trabajo .....	18
2.3.2    Ejecuciones .....	20
2.4    Análisis de los resultados .....	22
3    Conclusiones y discusión.....	24
3.1.1    Bibliografía .....	24
3.2    Cuestiones .....	25

# CAPÍTULO 1

## 1. DEFINICIÓN DEL SISTEMA

En este trabajo vamos a manejar Tomcat un servicio de apache 2 que funciona como contenedor web, es un servicio de código abierto que nos va a permitir usar Servlet de java, con el cual vamos a poder configurar y evaluar distintos servicios de un sistema.

### 1.1 Objetivos y definición del sistema

#### 1.1.1 Objetivos

Este trabajo consistirá en descargar, instalar, configurar y evaluar el servicio Tomcat en Ubuntu server, con este servicio manejaremos distintas técnicas de evaluación en base al uso de CPU, E/S Disco, E/S Red y uso de RAM. Este servicio será un soporte para Servlet y GPS ya que no es un servidor de aplicaciones

**Apache Tomcat** es un módulo de apache que funciona como un contenedor de servlets desarrollado bajo el proyecto Jakarta permite lanzar aplicaciones servlets y JSPs bajo apache sin mayores dificultades.

#### 1.1.2 Hardware

Las especificaciones hardware de nuestra máquina son:

- Intel i7-3770k procesador de cuatro núcleos
- Disco mecánico de 10 GB
- Memoria Ram de 2 GB
- Tarjeta gráfica AMD 3 GB

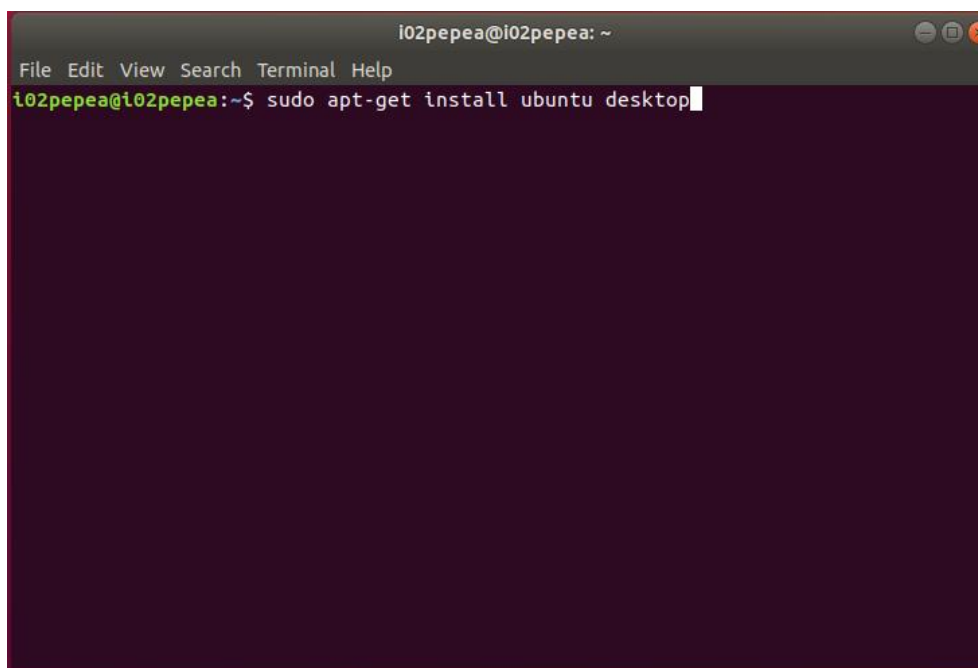
Con este hardware realizaremos la instalación del servidor.

### 1.1.3 Software

- El servicio a instalar será Tomcat 8.5.35
- Se instalará en Ubuntu server para alojar el servidor web
- Se usarán distintos programas para la monitorización del servicio y sus pruebas

### 1.1.4 Instalación y configuración

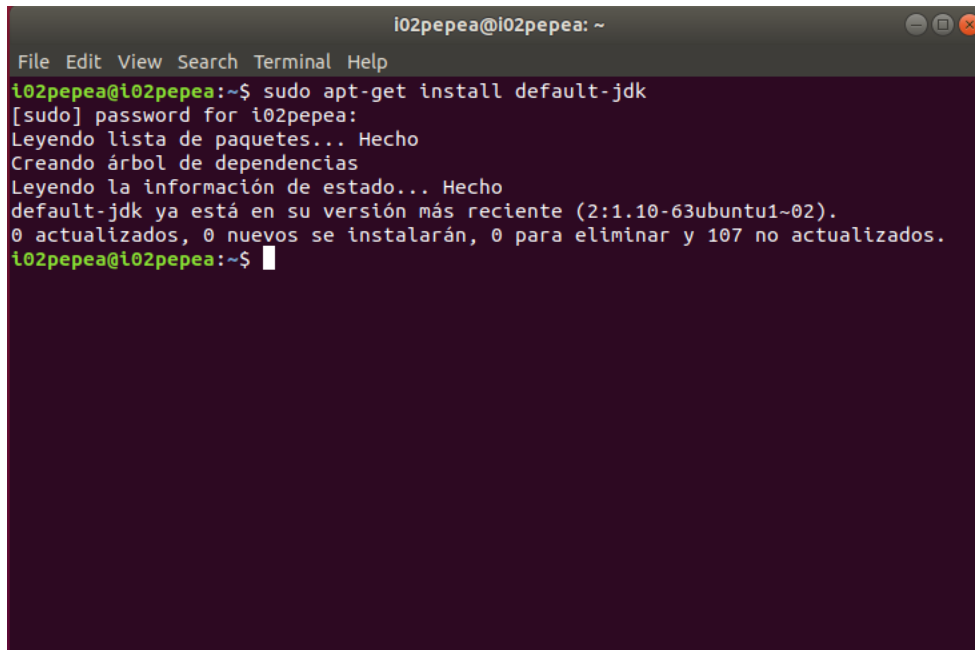
Dado que ya teníamos instalado Ubuntu server de prácticas anteriores solo nos va a hacer falta descargar, instalar y configurar Tomcat. Para que fuese más cómodo la instalación del servicio hemos instalado un escritorio en Ubuntu server ya que con el escritorio es más dinámico su configuración.



A continuación, deberíamos instalar apache2, pero ya lo teníamos instalado y configurado de otras prácticas, el comando sería:

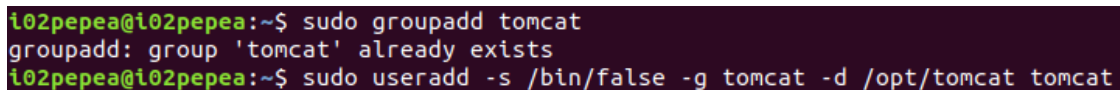
- sudo apt-get install apache2

Tomcat requiere que Java esté instalado en el servidor para poder ejecutar cualquier código de aplicación web Java. Podemos satisfacer este requisito instalando OpenJDK con apt-get.



```
i02pepea@i02pepea: ~  
File Edit View Search Terminal Help  
i02pepea@i02pepea:~$ sudo apt-get install default-jdk  
[sudo] password for i02pepea:  
Leyendo lista de paquetes... Hecho  
Creando árbol de dependencias  
Leyendo la información de estado... Hecho  
default-jdk ya está en su versión más reciente (2:1.10-63ubuntu1~02).  
0 actualizados, 0 nuevos se instalarán, 0 para eliminar y 107 no actualizados.  
i02pepea@i02pepea:~$
```

Por razones de seguridad, Tomcat debe ejecutarse como un usuario sin privilegios (es decir, no como root). Crearemos un nuevo usuario y grupo que ejecutará el servicio Tomcat.



```
i02pepea@i02pepea:~$ sudo groupadd tomcat  
groupadd: group 'tomcat' already exists  
i02pepea@i02pepea:~$ sudo useradd -s /bin/false -g tomcat -d /opt/tomcat tomcat
```

La mejor manera de instalar Tomcat es descargar la última versión de su web e instalarla manualmente.

Hay versiones más actualizadas, pero nosotros vamos a instalar la versión 8.5.35 ya que tiene muy buena estabilidad y los servicios que nosotros necesitamos.

The screenshot shows the Apache Tomcat 8.5.35 download page. The sidebar on the left contains links for Home, Download, Documentation, Problems, and Misc. The main content area displays the version 8.5.35 and provides links to download binary distributions (Core, Deployer, Extras, Embedded) and source code distributions. The page also includes a 'Quick Navigation' section with links to KEYS, 8.5.35, Browse, and Archives.

Instalares Tomcat en /opt/Tomcat , primero crearemos el escritorio y después moveremos los archivos descargados de Tomcat al directorio que acabamos de crear

```

102pepea@102pepea:~$ sudo groupadd tomcat
groupadd: group 'tomcat' already exists
102pepea@102pepea:~$ sudo mkdir /opt/tomcat

```

```
i02pepea@i02pepea:~$ sudo tar xzvf apache-tomcat-8*tar.gz -C /opt/tomcat --strip-components=1
```

El usuario de Tomcat que configuramos necesita tener acceso a la instalación de Tomcat. Lo configuraremos ahora.

```
i02pepea@i02pepea:~$ sudo chgrp -R tomcat /opt/tomcat
```

A continuación, le daremos al grupo tomcat acceso de lectura al directorio conf y a todos sus contenidos, y ejecutaremos el acceso al propio directorio:

```
i02pepea@i02pepea:~$ sudo chmod -R g+r conf
```

```
i02pepea@i02pepea:~$ sudo chown -R tomcat webapps/ work/ temp/ logs/
```

Ahora que se han configurado los permisos adecuados, podemos crear un archivo de servicio systemd para gestionar el proceso de Tomcat.

Queremos poder ejecutar Tomcat como un servicio, así que vamos a configurar el archivo de servicio systemd.

Para que funcione correctamente copiaremos la dirección del path de /jre y lo añadiremos el siguiente archivo que debemos crear para la configuración de Tomcat.

```
i02pepea@i02pepea: ~
File Edit View Search Terminal Help
GNU nano 2.9.3 /etc/systemd/system/tomcat.service

[Unit]
Description=Apache Tomcat Web Application Container
After=network.target

[Service]
Type=forking

Environment=JAVA_HOME=/usr/lib/jvm/java-1.11.0-openjdk-amd6
Environment=CATALINA_PID=/opt/tomcat/temp/tomcat.pid
Environment=CATALINA_HOME=/opt/tomcat
Environment=CATALINA_BASE=/opt/tomcat
Environment='CATALINA_OPTS=-Xms512M -Xmx1024M -server -XX:+UseParallelGC'
Environment='JAVA_OPTS=-Djava.awt.headless=true -Djava.security.egd=file:/dev/.$

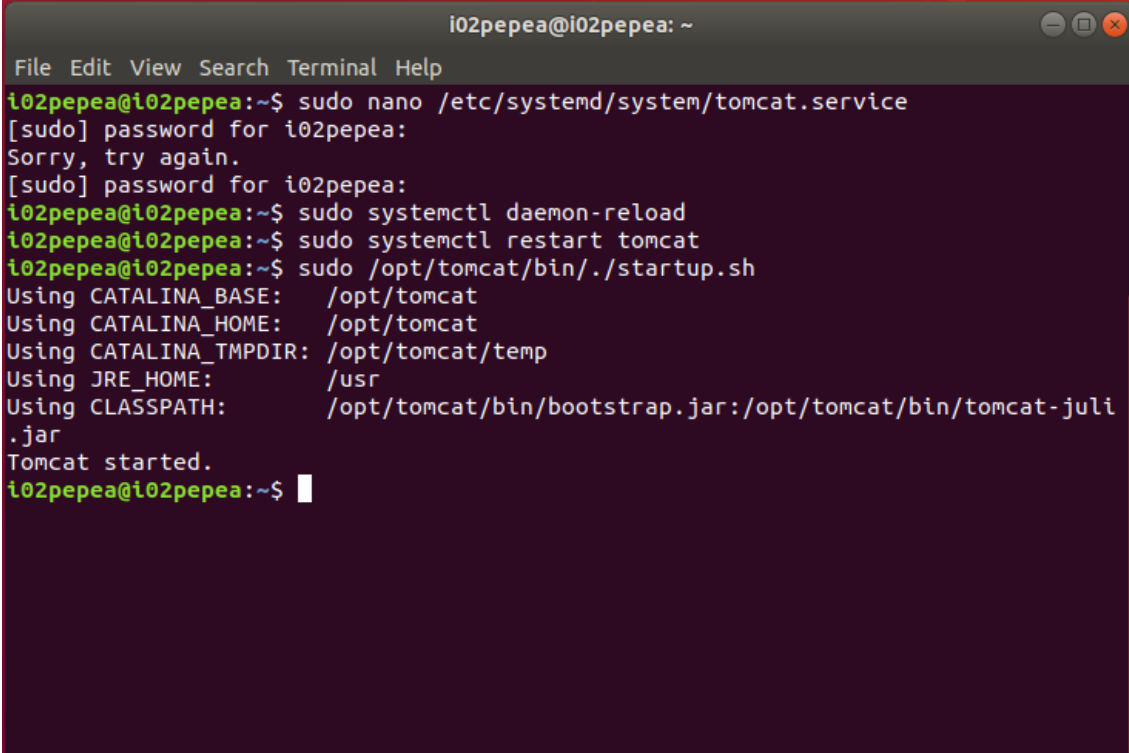
ExecStart=/opt/tomcat/bin/startup.sh
ExecStop=/opt/tomcat/bin/shutdown.sh

User=tomcat
Group=tomcat

[ Read 25 lines ]
^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify ^C Cur Pos
^X Exit ^R Read File ^\ Replace ^U Uncut Text ^T To Spell ^_ Go To Line
```

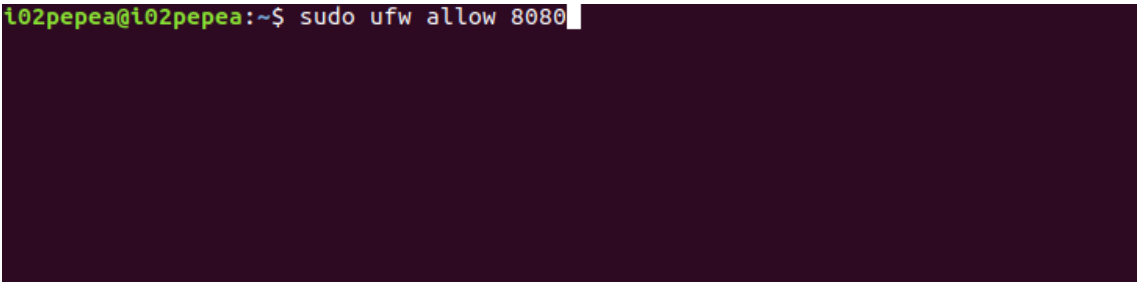


A continuación, recargaremos el proceso systemd para que conozca nuestro archivo de servicio, después reiniciaremos el servicio Tomcat:

A terminal window titled 'i02pepea@i02pepea: ~' with a menu bar (File, Edit, View, Search, Terminal, Help). The terminal shows the following commands and output:

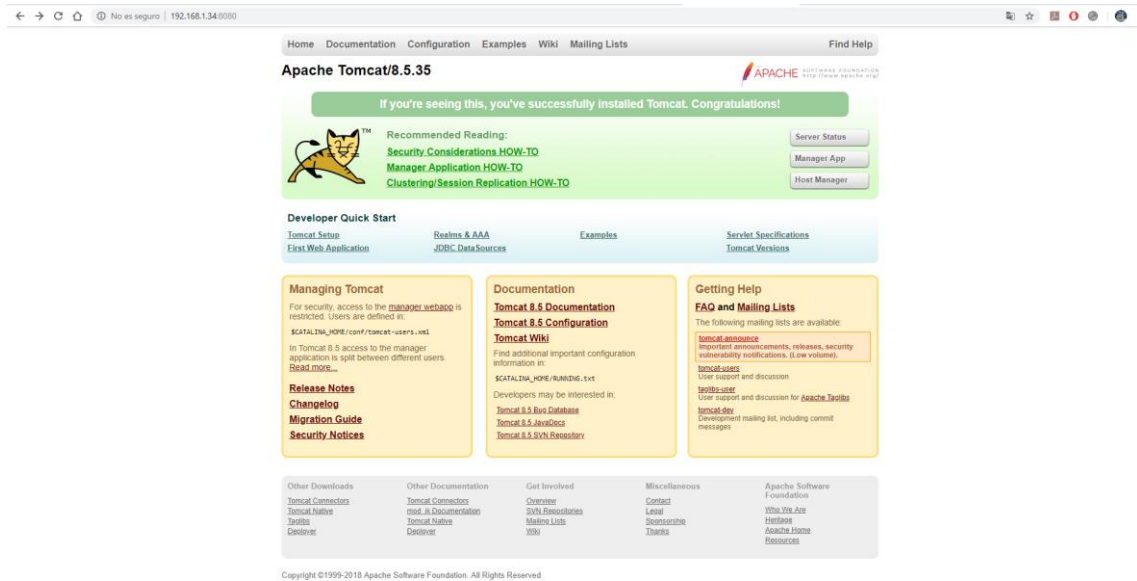
```
i02pepea@i02pepea:~$ sudo nano /etc/systemd/system/tomcat.service
[sudo] password for i02pepea:
Sorry, try again.
[sudo] password for i02pepea:
i02pepea@i02pepea:~$ sudo systemctl daemon-reload
i02pepea@i02pepea:~$ sudo systemctl restart tomcat
i02pepea@i02pepea:~$ sudo /opt/tomcat/bin/./startup.sh
Using CATALINA_BASE:   /opt/tomcat
Using CATALINA_HOME:   /opt/tomcat
Using CATALINA_TMPDIR: /opt/tomcat/temp
Using JRE_HOME:        /usr
Using CLASSPATH:       /opt/tomcat/bin/bootstrap.jar:/opt/tomcat/bin/tomcat-juli.jar
Tomcat started.
i02pepea@i02pepea:~$
```

Ahora que el servicio de Tomcat esta encendido habilitaremos el puerto:

A terminal window showing the command to allow port 8080:

```
i02pepea@i02pepea:~$ sudo ufw allow 8080
```

Con el firewall modificado, podremos acceder a la página de bienvenida predeterminada yendo al dominio o dirección IP seguida de :8080 en un navegador web:



Si todo ha funcionado correctamente, a continuación, habilitaremos Tomcat para que se encienda automáticamente:

```
i02pepea@i02pepea:~$ sudo systemctl enable tomcat
```

Una vez instalado Tomcat procederemos a configurar el servicio para poder manejarlo y realizar las tareas necesarias.

Para poder manejar Tomcat debemos añadir usuarios y darles privilegios.

```
i02pepea@i02pepea: ~
File Edit View Search Terminal Help
GNU nano 2.9.3 /opt/tomcat/conf/tomcat-users.xml

NOTE: The sample user and role entries below are intended for use with the
examples web application. They are wrapped in a comment and thus are ignored
when reading this file. If you wish to configure these users for use with the
examples web application, do not forget to remove the <!-- .. --> that surrounds
them. You will also need to set the passwords to something appropriate.
-->
<
<role rolename="tomcat"/>
<role rolename="role1"/>
<user username="tomcat" password="tomcat" roles="tomcat"/>
<user username="admin" password="admin" roles="manager-gui,admin-gui"/>
<user username="role1" password="<must-be-changed>" roles="role1"/>
</tomcat-users>

^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify ^C Cur Pos
^X Exit ^R Read File ^\ Replace ^U Uncut Text ^T To Spell ^_ Go To Line
```

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

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.
-->
<Context antiResourceLocking="false" privileged="true" >
<!-- <Valve className="org.apache.catalina.valves.RemoteAddrValve"
allow="127\.\d+\.\d+\.\d+|::1|0:0:0:0:0:0:1" />-->
<Manager sessionAttributeValueClassNameFilter="java\.lang\.(?:Boolean|Integer$
</Context>

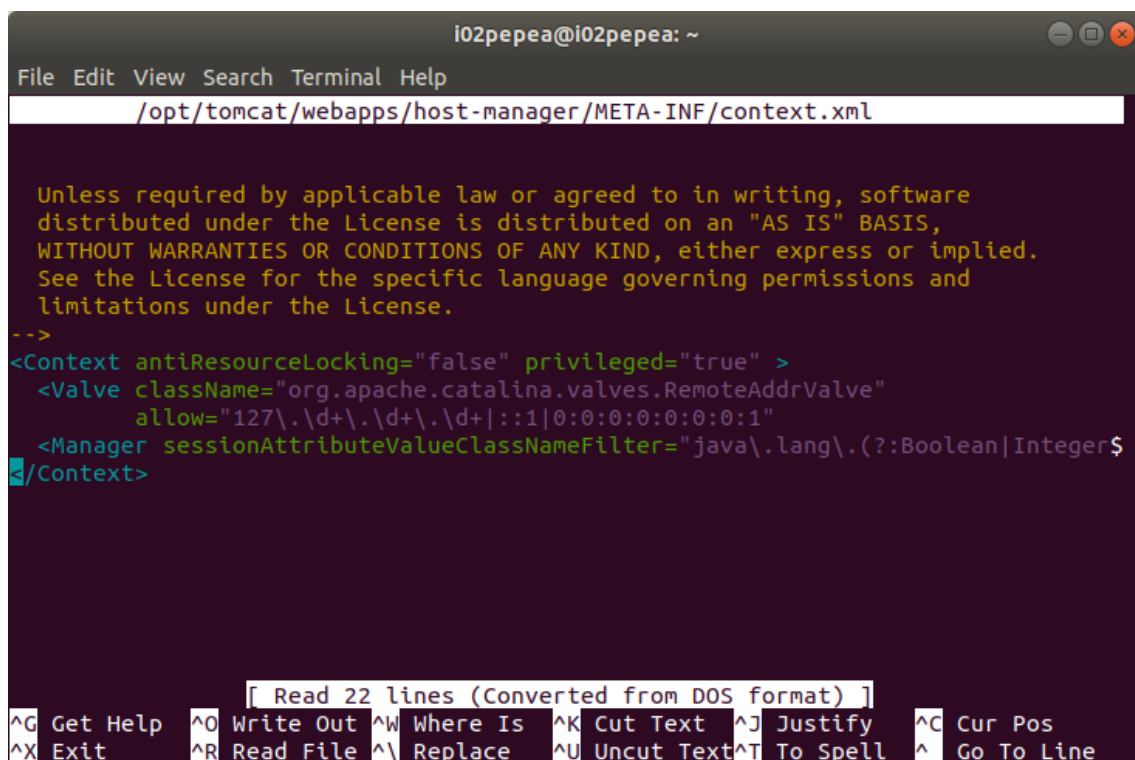
^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify ^C Cur Pos
^X Exit ^R Read File ^\ Replace ^U Uncut Text ^T To Spell ^_ Go To Line
```

Después de insertar los usuarios y darles privilegios procederemos a reiniciar el servicio para que se realicen los cambios.

```
i02pepea@i02pepea:~$ sudo nano /opt/tomcat/conf/tomcat-users.xml
Use "fg" to return to nano.

[1]+  Stopped                  sudo nano /opt/tomcat/conf/tomcat-users.xml
i02pepea@i02pepea:~$ sudo nano /opt/tomcat/conf/tomcat-users.xml
i02pepea@i02pepea:~$ sudo nano /opt/tomcat/webapps/host-manager/META-INF/context.xml
i02pepea@i02pepea:~$ sudo nano /opt/tomcat/webapps/host-manager/META-INF/context.xml
i02pepea@i02pepea:~$ sudo systemctl restart tomcat
```

Ademas, comentaremos la restricción de la dirección IP para permitir conexiones desde cualquier lugar. Alternativamente, si queremos permitir el acceso sólo a conexiones procedentes de su propia dirección IP, puede añadir su dirección IP pública a la lista:



```
i02pepea@i02pepea: ~
File Edit View Search Terminal Help
/opt/tomcat/webapps/host-manager/META-INF/context.xml

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.
-->
<Context antiResourceLocking="false" privileged="true" >
  <Valve className="org.apache.catalina.valves.RemoteAddrValve"
    allow="127\.\d+\.\d+\.\d+|::1|0:0:0:0:0:0:0:1"
  <Manager sessionAttributeValueClassNameFilter="java\.lang\.(?:Boolean|Integer$
/Context>
```

[ Read 22 lines (Converted from DOS format) ]

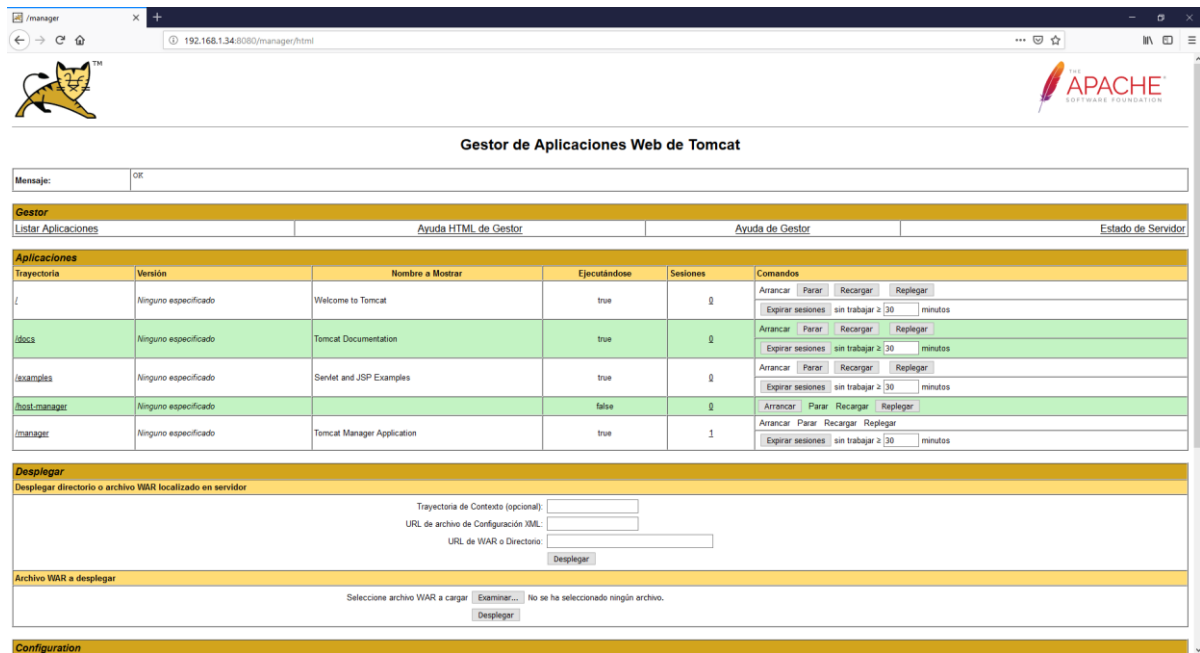
^G Get Help	^O Write Out	^W Where Is	^K Cut Text	^J Justify	^C Cur Pos
^X Exit	^R Read File	^\ Replace	^U Uncut Text	^T To Spell	^_ Go To Line

Y después volveremos a usar este comando, para que todos los cambios hagan efecto.

-Sudo systemctl restart Tomcat

Ya tendríamos todo configurado para poder usar el servidor Tomcat como podemos observar en las siguientes imágenes:

-Manager-gui



**Gestor de Aplicaciones Web de Tomcat**

Mensaje: OK

**Gestor**

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

Aplicaciones	Versión	Nombre a Mostrar	Ejecutándose	Sesiones	Comandos
/	Ninguno especificado	Welcome to Tomcat	true	0	Arrancar Parar Recargar Replegar Expirar sesiones sin trabajar > 30 minutos
/docs	Ninguno especificado	Tomcat Documentation	true	0	Arrancar Parar Recargar Replegar Expirar sesiones sin trabajar > 30 minutos
/examples	Ninguno especificado	Servlet and JSP Examples	true	0	Arrancar Parar Recargar Replegar Expirar sesiones sin trabajar > 30 minutos
/host-manager	Ninguno especificado		false	0	Arrancar Parar Recargar Replegar
/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):

URL de archivo de Configuración XML:

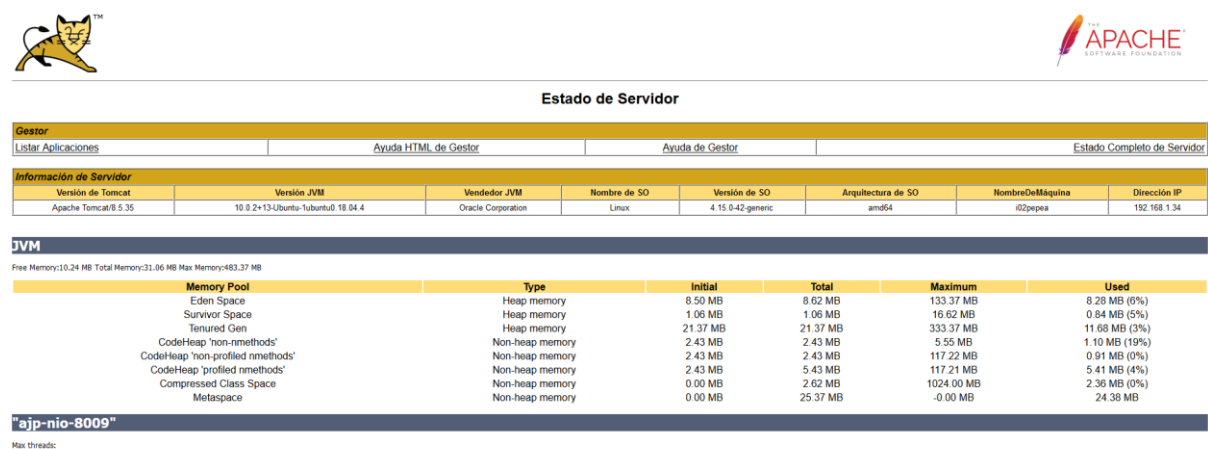
URL de WAR o Directorio:

**Archivo WAR a desplegar**

Seleccione archivo WAR a cargar  No se ha seleccionado ningún archivo.

**Configuration**

-Estado del servidor:



**Estado de Servidor**

**Gestor**

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

**Información de Servidor**

Versión de Tomcat	Versión JVM	Vendedor JVM	Nombre de SO	Versión de SO	Arquitectura de SO	NombreDeMáquina	Dirección IP
Apache Tomcat/8.5.35	10.0.2+13-Ubuntu18.04.4	Oracle Corporation	Linux	4.15.0-42-generic	amd64	i02pepa	192.168.1.34

**JVM**

Free Memory:10.24 MB Total Memory:31.06 MB Max Memory:483.37 MB

Memory Pool	Type	Initial	Total	Maximum	Used
Eden Space	Heap memory	8.50 MB	8.62 MB	133.37 MB	8.28 MB (6%)
Survivor Space	Heap memory	1.06 MB	1.06 MB	16.62 MB	0.84 MB (5%)
Tenured Gen	Heap memory	21.37 MB	21.37 MB	333.37 MB	11.68 MB (3%)
CodeHeap 'non-nmethods'	Non-heap memory	2.43 MB	2.43 MB	5.55 MB	1.10 MB (19%)
CodeHeap 'non-profiled nmethods'	Non-heap memory	2.43 MB	2.43 MB	117.22 MB	0.91 MB (0%)
CodeHeap 'profiled nmethods'	Non-heap memory	2.43 MB	5.43 MB	117.21 MB	5.41 MB (4%)
Compressed Class Space	Non-heap memory	0.00 MB	2.62 MB	1024.00 MB	2.36 MB (0%)
Metaspace	Non-heap memory	0.00 MB	25.37 MB	-0.00 MB	24.38 MB

**"ajp-nio-8009"**

Max threads:

## 1.2 Servicios y sus posibles resultados

A continuación, vamos a explicar los posibles servicios y funcionalidades de Tomcat y sus posibles resultados.

### 1.2.1 Servicios Tomcat

Tomcat es el servicio que hemos instalado el cual vamos a configurar y utilizar, este servicio nos ofrece las siguientes funcionalidades:

- El **balanceo de carga** es el proceso que permite dividir la cantidad de trabajo que debe realizar un ordenador, entre dos o más ordenadores, de manera que se obtiene más resultados de trabajo en la misma cantidad de tiempo.
- **Enviar Datos a Páginas.** La aplicación permite enviar un mensaje a través de distintos métodos. La información es recibida por la página dinámica.
- **Página Dinámica.** Página JSP que despliega la información enviada por la página de envío de datos.
- **Envío de Mail usando Servlet.** En esta página se puede enviar un correo electrónico utilizando javax.mail.

La jerarquía de directorios de instalación de Tomcat incluye:

- **bin** - arranque, cierre, y otros scripts y ejecutables.
- **common** - clases comunes que pueden utilizar Catalina y las aplicaciones web.
- **conf** - ficheros XML y los correspondientes DTD para la configuración de Tomcat.
- **logs** - logs de Catalina y de las aplicaciones.
- **server** - clases utilizadas solamente por Catalina.
- **shared** - clases compartidas por todas las aplicaciones web.
- **webapps** - directorio que contiene las aplicaciones web.
- **work** - almacenamiento temporal de ficheros y directorios.

### 1.2.2 Posibles Resultados

Los resultados consistirán en la evaluación de las cargas de prueba que el profesor ha proporcionado, que, a su vez, monitorizaremos para ver cómo actúan las distintas métricas del sistema.

Los posibles resultados serán:

- Insertar un benchmark proporcionado por el profesor en el servidor.
- Monitorización desde Ubuntu Server la carga de trabajo con alguna herramienta del sistema.
- Analizar los resultados del sistema

Una vez realizadas las operaciones sacaremos conclusiones de todas las pruebas realizadas.

## 1.3 Métricas

Las métricas de este trabajo consistirán en una serie de criterios y magnitudes para comparar las distintas prestaciones del servicio.

Las métricas serán las siguientes:

- **Uso de CPU.** Con esta métrica analizaremos el porcentaje de la CPU usado para la ejecución de los procesos.
- **Entrada/Salida Disco (Kb/s).** Con esta métrica monitorizaremos la velocidad de transferencia en la entrada y salida del disco duro.
- **Entrada/Salida Red (Kb/s).** Con esta métrica monitorizaremos el uso de red del servidor Tomcat cuando hagamos las pruebas.
- **Uso de memoria RAM.** Con esta métrica analizaremos el porcentaje de la memoria RAM que el sistema necesite para los test.

## 1.4 Parámetros

En este apartado nos vamos a centrar en los distintos parámetros que pueden afectar a las prestaciones, como son las características de las limitaciones hardware y software.

-Tipo de limitaciones:

### 1.4.1 Limitaciones Hardware

Este tipo de limitaciones son aquellas que presentan los componentes del hardware como lo son la CPU, la memoria RAM, la tarjeta de red, disco duro, etc.

Al tener nuestro servicio instalado en una máquina virtual el sistema está compartiendo recursos que pueden limitar tanto una máquina como la otra ya que, si en la máquina anfitriona estas usando el hardware en alto rendimiento, afectará a la máquina virtual y limitará las cargas de prueba que realices.

### 1.4.2 Limitaciones Software

Este tipo de limitaciones son aquellas que presenta el sistema operativo durante la ejecución de un programa.

Cuando estemos realizando la ejecución de la monitorización de las cargas de prueba, los distintos procesos pueden consumir mucho hardware, lo que puede llevar a una ralentización del sistema y afectar a la monitorización.



## CAPÍTULO 2

### 2 EVALUACIÓN DEL SISTEMA

En esta sección vamos a realizar las distintas pruebas para las métricas del sistema propuestas con su correspondiente monitorización de cada una de ellas donde se analizarán los resultados del trabajo realizado.

#### 2.1 Técnicas de evaluación

Estas técnicas consistirán en una serie de pasos para el estudio del sistema que hemos instalado como pueden ser:

- Subida de la carga
- Ejecución de la carga
- Monitorización

##### 2.1.1 Herramientas para la evaluación

Para la evaluación del sistema, realizaremos la monitorización con la herramienta Sar de Linux ya que, viene incluida en el sistema y es sencilla de usar.

Esta herramienta se puede usar en el terminal y cuando subamos la carga y ejecutemos, usaremos los distintos comandos que tiene Sar para la monitorización de la carga.

Sar tiene distintas opciones para monitorización del hardware, las principales que vamos a usar son:

- **sar -u 2.** Este comando monitoriza el uso de la CPU cada dos segundos.
- **sar -n DEV.** Este comando monitoriza el uso de la red.

- **sar -r.** Este comando monitoriza el uso de la RAM.
- **sar -A.** Este comando monitoriza todos los modos.
- **sar -d.** Este comando muestra la entrada/salida de las ejecuciones.
- **sar -b.** Este comando muestra el número de transferencias de entrada/salida del dispositivo.

## 2.2 Carga de trabajo

La carga de trabajo consistirá en una serie de archivos que proporciona el profesor. El archivo contiene subcarpetas con imágenes. El objetivo de la carga es estresar el sistema subiendo distintas imágenes al servidor Tomcat.

## 2.3 Diseño de Experimentos

El trabajo consiste en monitorizar el servidor cuando se carguen los archivos del trabajo, se realizarán una serie de trabajos que estresarán el sistema para poder ver el rendimiento del hardware y software con este servicio web.

Nuestro principal objetivo es monitorizar la actividad de las métricas explicadas anteriormente con la monitorización correspondiente.

Se realizarán una serie de pasos antes de comenzar con la monitorización, como guardar y subir la carga de trabajo proporcionada por el profesor al servidor Tomcat,

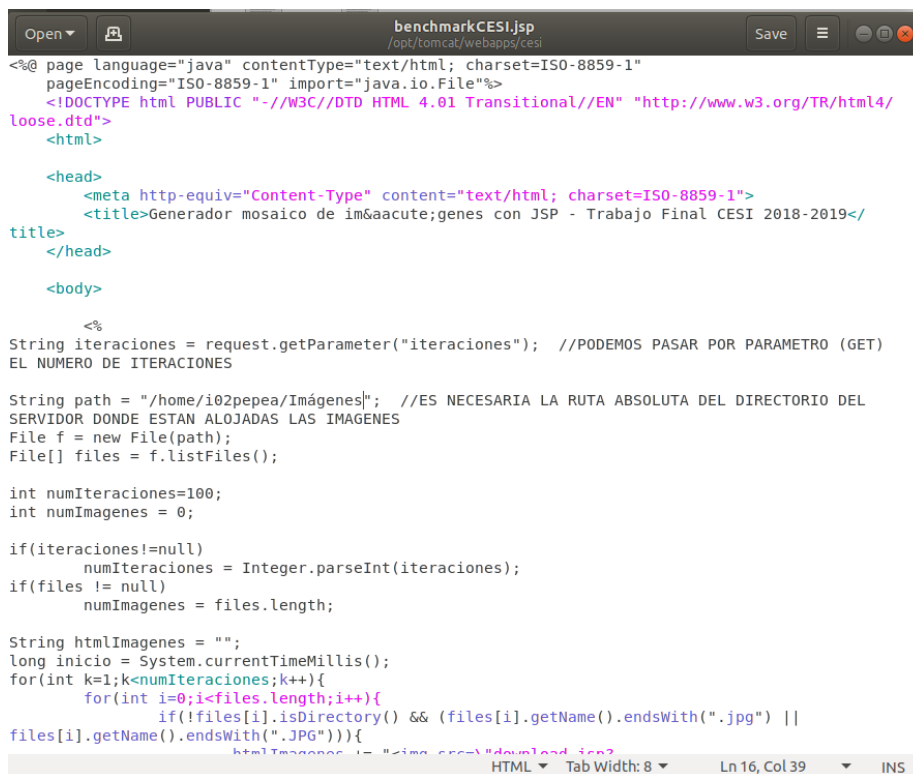
Las localizaciones de los archivos se van a encontrar en el servidor Tomcat instalado y configurado anteriormente, comenzamos.

### 2.3.1 Configuración y subida de la Carga de trabajo

En primer lugar, compilaremos los archivos jsp para que el servidor los pueda leer:

```
i02pepea@i02pepea:~$ cd Descargas/
i02pepea@i02pepea:~/Descargas$ sudo mv benchmarkCESI.jsp /opt/tomcat/webapps/ap
p/
i02pepea@i02pepea:~/Descargas$ sudo mv download.jsp /opt/tomcat/webapps/app/
i02pepea@i02pepea:~/Descargas$ cd app
bash: cd: app: No such file or directory
i02pepea@i02pepea:~/Descargas$ cd
i02pepea@i02pepea:~$ cd /opt/tomcat/webapps/app/
i02pepea@i02pepea:/opt/tomcat/webapps/app$ jar cvf cesi.war
El indicador 'c' necesita la especificación de archivos de manifiesto o de entra
da.
Intente `jar --help' para obtener más información.
i02pepea@i02pepea:/opt/tomcat/webapps/app$ jar cvf cesi.war .
manifiesto agregado
agregando: benchmarkCESI.jsp(entrada = 1577) (salida = 859)(desinflado 45%)
agregando: download.jsp(entrada = 1068) (salida = 473)(desinflado 55%)
java.nio.file.AccessDeniedException: /tmp/cesi.war2475404739907108230.jar -> ces
i.war
    at java.base/sun.nio.fs.UnixException.translateToIOException(UnixExcepti
on.java:90)
    at java.base/sun.nio.fs.UnixException.rethrowAsIOException(UnixException
.java:111)
    at java.base/sun.nio.fs.UnixCopyFile.move(UnixCopyFile.java:462)
    at java.base/sun.nio.fs.UnixFileSystemProvider.move(UnixFileSystemProvid
```

Después modificaremos el archivo benchmarkCESI para poner la ruta donde se encuentran nuestras imágenes:



```
benchmarkCESI.jsp
/opt/tomcat/webapps/cesi

<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1" import="java.io.File"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/
loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>Generador mosaico de imágenes con JSP - Trabajo Final CESI 2018-2019</
title>
</head>
<body>
<%
String iteraciones = request.getParameter("iteraciones"); //PODEMOS PASAR POR PARAMETRO (GET)
EL NUMERO DE ITERACIONES

String path = "/home/i02pepea/Imágenes"; //ES NECESARIA LA RUTA ABSOLUTA DEL DIRECTORIO DEL
SERVIDOR DONDE ESTAN ALOJADAS LAS IMAGENES
File f = new File(path);
File[] files = f.listFiles();

int numIteraciones=100;
int numImágenes = 0;

if(iteraciones!=null)
    numIteraciones = Integer.parseInt(iteraciones);
if(files != null)
    numImágenes = files.length;

String htmlImágenes = "";
long inicio = System.currentTimeMillis();
for(int k=1;k<numIteraciones;k++){
    for(int i=0;i<files.length;i++){
        if(!files[i].isDirectory() && (files[i].getName().endsWith(".jpg") ||
files[i].getName().endsWith(".JPG"))){
            htmlImágenes += "<img src=\"" + download.jsp?";
            "
```

Editamos el index para poder utilizar la carga desde el servicio Tomcat:

```
<!--
Licensed to the Apache Software Foundation (ASF) under one or more
contributor license agreements. See the NOTICE file distributed with
this work for additional information regarding copyright ownership.
The ASF licenses this file to You under the Apache License, Version 2.0
(the "License"); you may not use this file except in compliance with
the License. You may obtain a copy of the License at

    http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.
-->
<!DOCTYPE HTML><html lang="en"><head>
<meta charset="UTF-8">
<title>Apache Tomcat Examples</title>
</head>
<body>
<p>
<h3>Apache Tomcat Examples</H3>
<p></p>
<ul>
<li><a href="benchmarkCESI.jsp">benchmark</a></li>
<li><a href="download.jsp">download</a></li>
<li><a href="websocket/index.xhtml">WebSocket Examples</a></li>
</ul>
</body></html>
```

Aquí podemos observar como se ha configurado correctamente:

```
<!--
Licensed to the Apache Software Foundation (ASF) under one or more
contributor license agreements. See the NOTICE file distributed with
this work for additional information regarding copyright ownership.
The ASF licenses this file to You under the Apache License, Version 2.0
(the "License"); you may not use this file except in compliance with
the License. You may obtain a copy of the License at

    http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.
-->
<!DOCTYPE HTML><html lang="en"><head>
<meta charset="UTF-8">
<title>Apache Tomcat Examples</title>
</head>
<body>
<p>
<h3>Apache Tomcat Examples</H3>
<p></p>
<ul>
<li><a href="benchmarkCESI.jsp">benchmark</a></li>
<li><a href="download.jsp">download</a></li>
<li><a href="websocket/index.xhtml">WebSocket Examples</a></li>
</ul>
</body></html>
```

Y comprobamos que funciona para poder monitorizar:



2.3.2 Ejecuciones

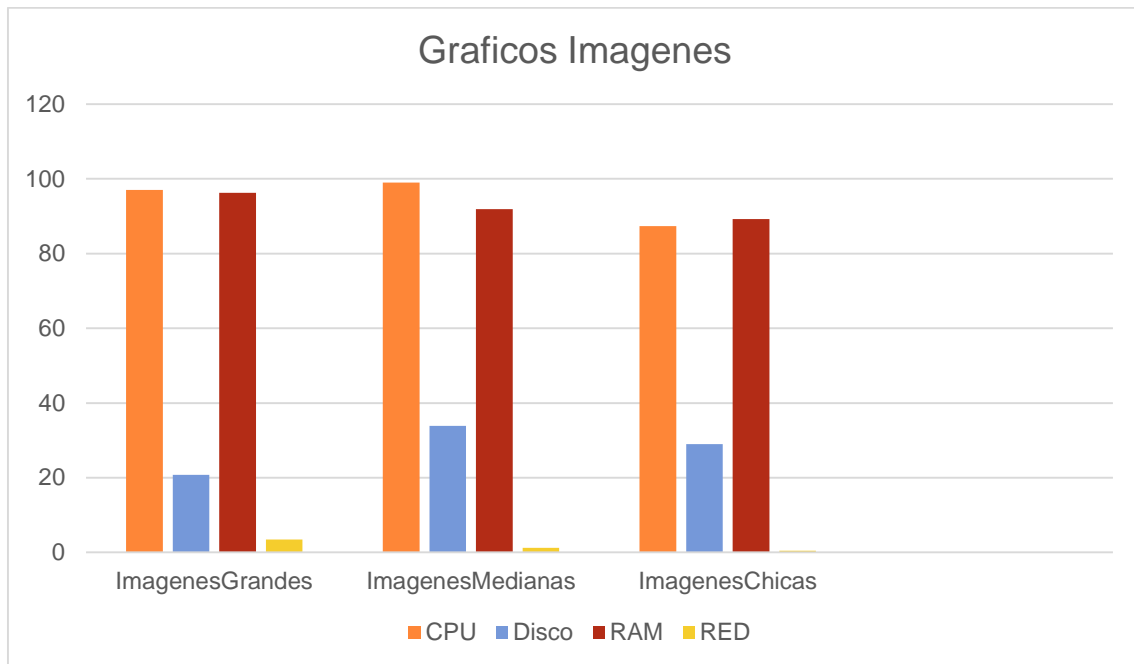
A continuación, mostraremos las ejecuciones de los tres tipos de archivos que ha facilitado el profesor según la métrica propuesta mediante tablas:

Imágenes Grandes				
	CPU %	Disco (Kb/s)	RAM %	Red (KB/s)
1	98	16,00	96.22	1,856
2	94	15.00	96.28	1,28
3	94	32,00	96.24	10,048
4	100	20,00	96.24	0,57
Media	97,00	20,75	96.25	3,43

Imágenes Medianas				
	CPU %	Disco (Kb/s)	RAM %	Red (KB/s)
1	100	16,33	91.87	0.00
2	94	76,77	91.86	0.512
3	99	25,33	91.86	0.192
4	98	17,23	91.86	0.832
Media	99	33,915	91.86	0.2088

Imágenes Medianas				
	CPU %	Disco (Kb/s)	RAM %	Red (KB/s)
1	95.45	36.00	89.25	0.128
2	99.50	44.00	89.25	0.516
3	99.00	16.00	89.60	0.09384
4	55.50	20.00	88.90	0.2001
Media	87.36	29	89.20	0.4565

A continuación, mostrares un gráfico con la media general de las cargas de las ejecuciones:



## 2.4 Análisis de los resultados

- CPU

Con respecto a la CPU para las imágenes grandes tenemos ocupado más porcentaje que al resto lo cual significa que las imágenes grandes necesitan más porcentaje de procesador en el servidor web para poder cargarlas

Por otra parte, podemos observar que para las imágenes pequeñas no necesitamos tanto porcentaje para poder cargarlas.

- Disco

Como podemos observar en la gráfica donde más porcentajes de disco se ha usado para entrada salidas ha sido en las imágenes medianas, pero en este apartado no podemos observar un gran cambio con respecto a otros.

- RAM

En este apartado podemos observar como las imágenes grandes requieren mayor uso de memoria RAM ya que tienen mayor tamaño que el resto al comienzo se puede observar en todas las ejecuciones que usan el 100% de esta memoria y va disminuyendo su uso dependiendo del tamaño de las imágenes.

- Red

Este apartado es también uno de los más similares con respecto a las otras métricas ya que hemos observado, que da igual el tipo de imágenes.

Al ejecutarlo sea cual sea el tipo de imagen, saltaban picos de Kb en cualquiera de las ejecuciones con los que no podemos concluir con precisión ventajas entre los tipos de imágenes.

- Tiempos

Este apartado no tiene datos concretos, pero hemos observado que las imágenes Grandes el servidor las carga más rápido que las pequeñas.



## CAPÍTULO 3

### 3 CONCLUSIONES Y DISCUSIÓN

Concluyendo con este trabajo podemos sacar algunas conclusiones teniendo en cuenta el número de ejecuciones de las cargas de trabajo.

La carga que mayor uso del hardware necesitaba era las imágenes grandes, ya que, al ser de mayor tamaño necesitaban más porcentaje de uso del sistema, pero no podemos sacar conclusiones concretas debido a que estamos en un servicio virtualizado.

Con esto podemos decir la utilidad del servicio Tomcat, un servicio gratuito de código abierto, el cual no necesita de una maquina muy potente para poder ejecutarse y llevar tareas algo pesadas en equipos de bajo rendimiento, si es verdad que al estar virtualizado no podemos dar una opinión concreta como hemos dicho anteriormente.

#### 3.1.1 Bibliografía

<https://www.arsys.es/blog/programacion/tomcat-servidores-cloud/>

<http://profesores.elo.utfsm.cl/~agv/elo330/2s03/projects/Tomcat/index.html>

<https://tomcat.apache.org/tomcat-8.5-doc/windows-service-howto.html>

<https://www.adictosaltrabajo.com/2012/06/27/tuning/>

<https://medium.com/@valerybriz/gu%C3%ADa-completa-para-montar-un-servlet-con-java-tomcat-y-ant-explicado-paso-a-paso-windows-b2ff203c50d3>

[https://programacion.net/articulo/desplegar\\_servlets\\_y\\_aplicaciones\\_web\\_en\\_tomcat\\_y\\_weblogic\\_server\\_175](https://programacion.net/articulo/desplegar_servlets_y_aplicaciones_web_en_tomcat_y_weblogic_server_175)

[https://www.ecured.cu/Servidor\\_Tomcat](https://www.ecured.cu/Servidor_Tomcat)

<https://www.digitalocean.com/community/tutorials/how-to-install-apache-tomcat-8-on-ubuntu-16-04>

## 3.2 Cuestiones

### **¿Es posible tomar mediciones más precisas con otras herramientas?**

Sí, hemos observado que con top pueden tomarse mediciones más precisas que con `sas`, pero ambas tienen muy buenas utilidades que investigando pueden aprovecharse mejor.

### **¿Se pueden administrar varios roles de usuario en Tomcat?**

Si es posible si modificas el archivo `sudo nano /opt/tomcat/conf/tomcat-users.xml` puedes agregar todo tipo de usuarios para que puedan acceder al servicio y sus archivos sin que pueden eliminar o parar procesos o cargas de trabajo.

### **¿Qué problema hay si le das permisos a la carpeta generada por el archivo war?**

Ha sido el mayor error que nos ha costado resolver, pero, al darle permisos con `chmod` el servicio Tomcat deniega el acceso y no puedes acceder al benchmark lo que hace que no cargue la configuración.