Servidor web con soporte de servlets y jsps

Trabajo final CESI

Martin Paez Anguita

Adil Hirchi

José María Martin Santaella

Grupo 17

Tipo 2 (21 horas de trabajo)

Índice General

[1. Definición del Sistema 2](#_Toc532159923)

[1.1 Objetivos y definición del sistema 2](#_Toc532159924)

[1.1.1 Objetivos 2](#_Toc532159925)

[1.1.2 Hardware 2](#_Toc532159926)

[1.1.3 Software 3](#_Toc532159927)

[1.1.4 Instalación y configuración 3](#_Toc532159928)

[1.2 Servicios y sus posibles resultados 13](#_Toc532159929)

[1.2.1 Servicios Tomcat 13](#_Toc532159930)

[1.2.2 Posibles Resultados 14](#_Toc532159931)

[1.3 Métricas 14](#_Toc532159932)

[1.4 Parámetros 15](#_Toc532159933)

[1.4.1 Limitaciones Hardware 15](#_Toc532159934)

[1.4.2 Limitaciones Software 15](#_Toc532159935)

[2 Evaluación del Sistema 16](#_Toc532159936)

[2.1 Técnicas de evaluación 16](#_Toc532159937)

[2.1.1 Herramientas para la evaluación 16](#_Toc532159938)

[2.2 Carga de trabajo 17](#_Toc532159939)

[2.3 Diseño de Experimentos 17](#_Toc532159940)

[2.3.1 Configuración y subida de la Carga de trabajo 18](#_Toc532159941)

[2.3.2 Ejecuciones 20](#_Toc532159942)

[2.4 Análisis de los resultados 22](#_Toc532159943)

[3 Conclusiones y discusión 24](#_Toc532159944)

[3.1.1 Bibliografía 24](#_Toc532159945)

[3.2 Cuestiones 25](#_Toc532159946)

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.

## Objetivos y definición del sistema

### 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.

### 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.

### Software

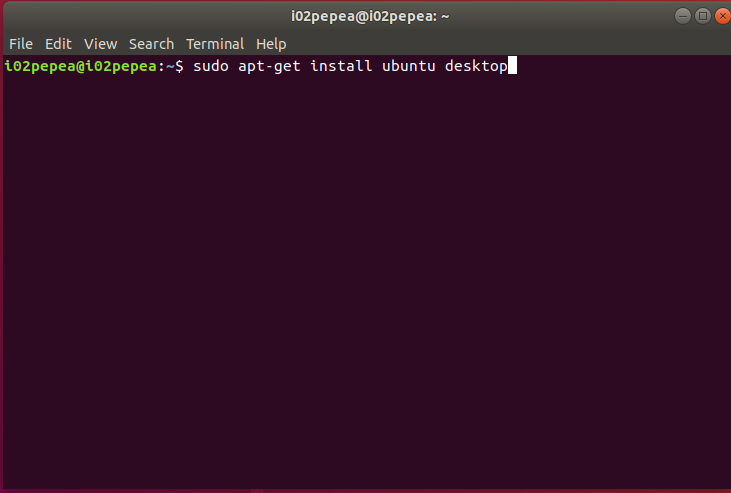
-El servicio a instalar será Tomcat 8.5.35

-Se instalará en Ubuntu server para alojar el servidor web

-Se usarán distintos programas para la monitorización del servicio y sus pruebas

### Instalación y configuración

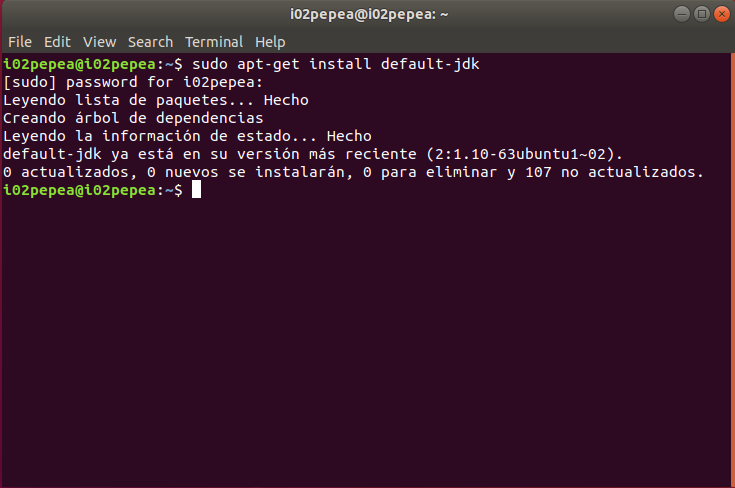
Dado que ya teníamos instalado Ubuntu server de prácticas anteriores solo nos va a hacer falta descargar, instalar y configurar Tomcat. Para que fuese más cómodo la instalación del servicio hemos instalado un escritorio en Ubuntu server ya que con el escritorio es más dinámico su configuración.



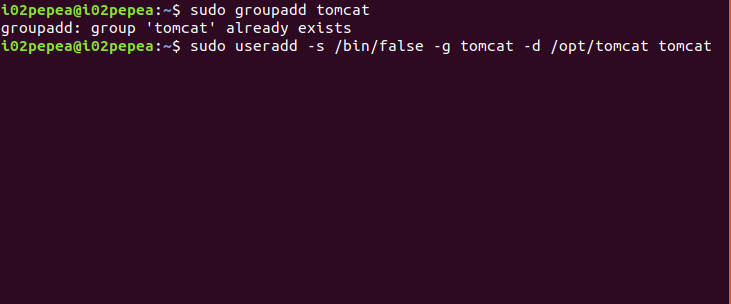
A continuación, deberíamos instalar apache2, pero ya lo teníamos instalado y configurado de otras prácticas, el comando seria:

-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.

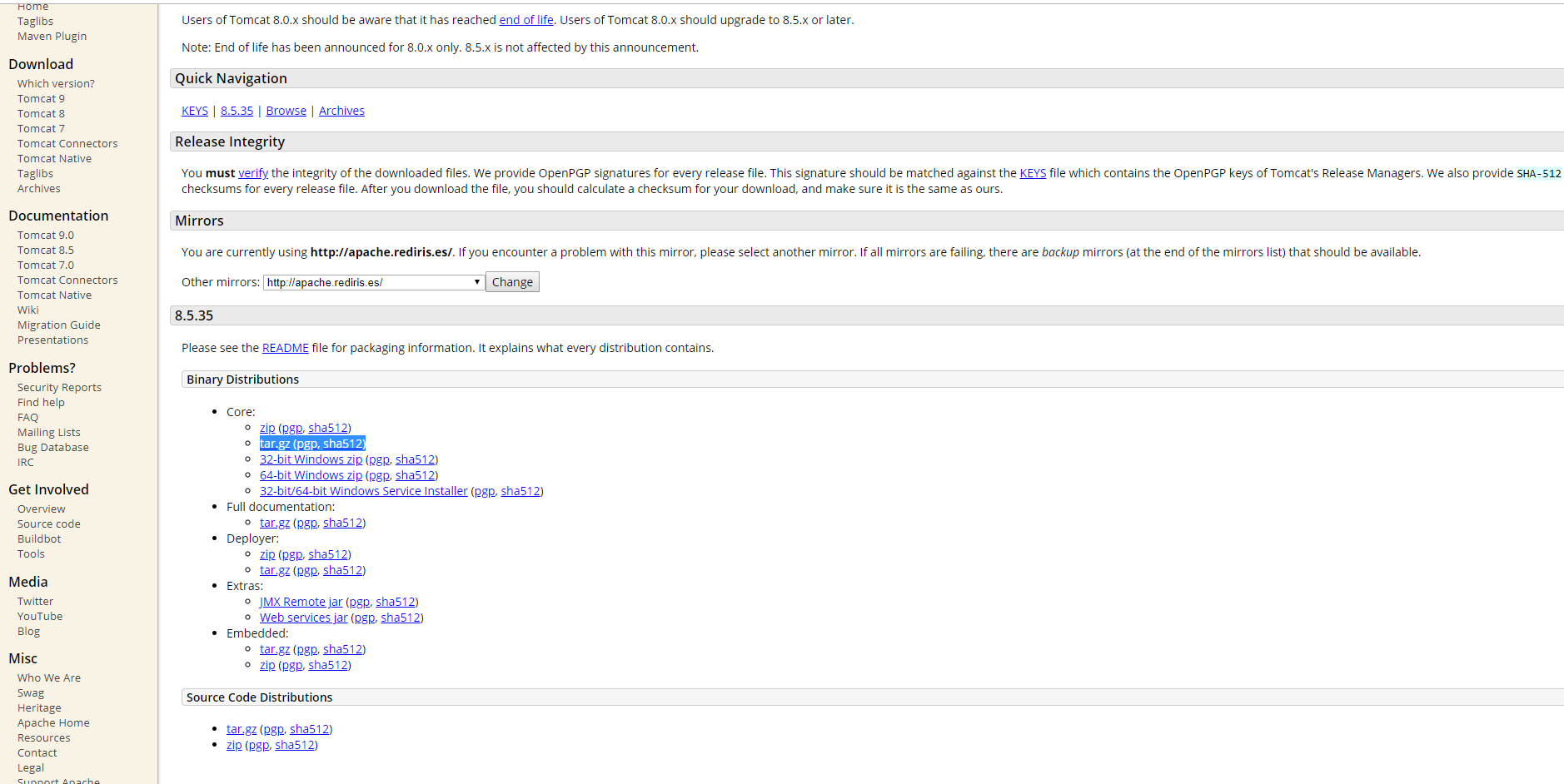


Por razones de seguridad, Tomcat debe ejecutarse como un usuario sin privilegios (es decir, no como root). Crearemos un nuevo usuario y grupo que ejecutará el servicio Tomcat.



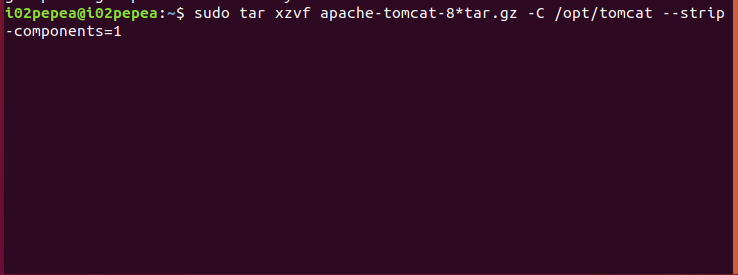
La mejor manera de instalar Tomcat es descargar la última versión de su web e instalarla manualmente.

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.



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

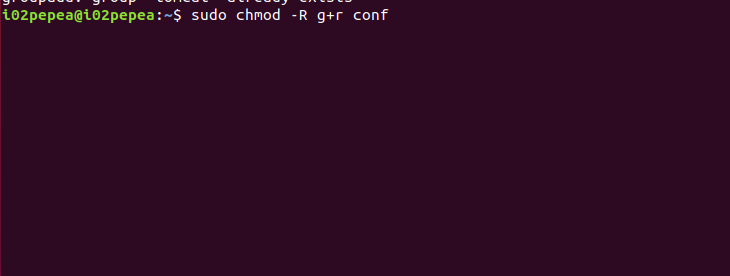


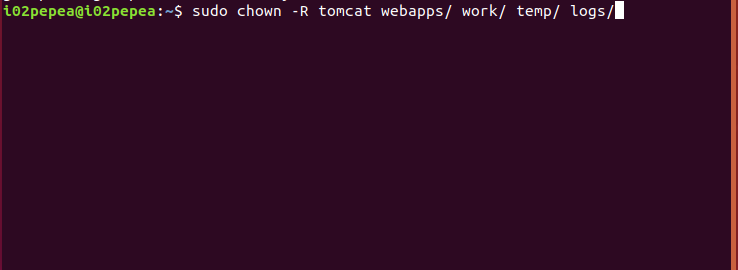


El usuario de Tomcat que configuramos necesita tener acceso a la instalación de Tomcat. Lo configuraremos ahora.



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:

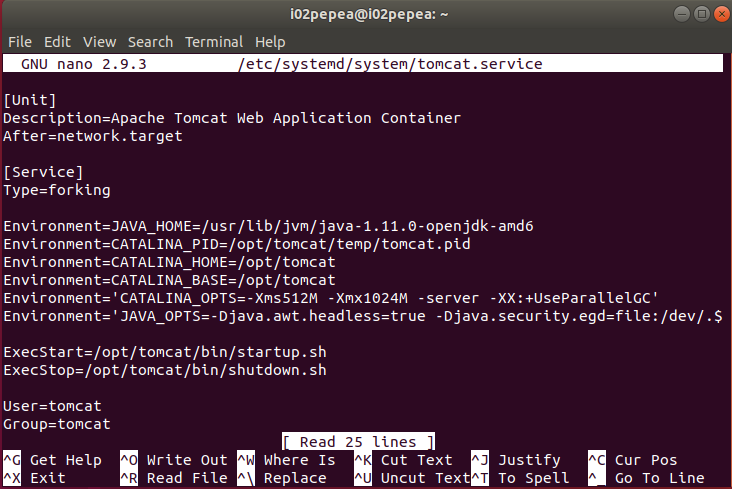




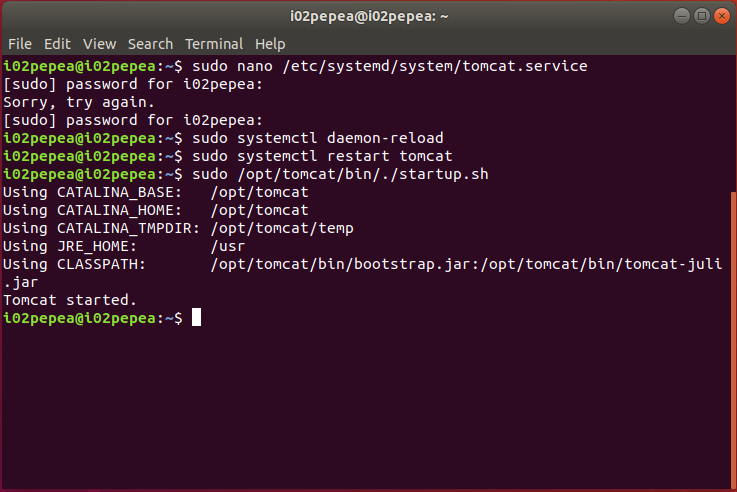
Ahora que se han configurado los permisos adecuados, podemos crear un archivo de servicio systemd para gestionar el proceso de Tomcat.

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

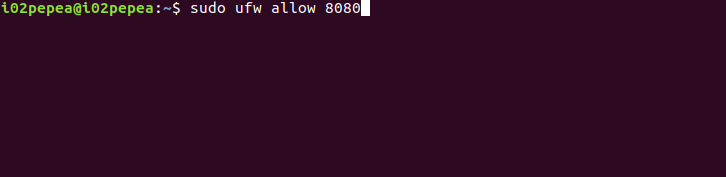
Para que funcione correctamente copiaremos la dirección del path de /jre y lo añadiremos el siguiente archivo que debemos crear para la configuración de Tomcat.



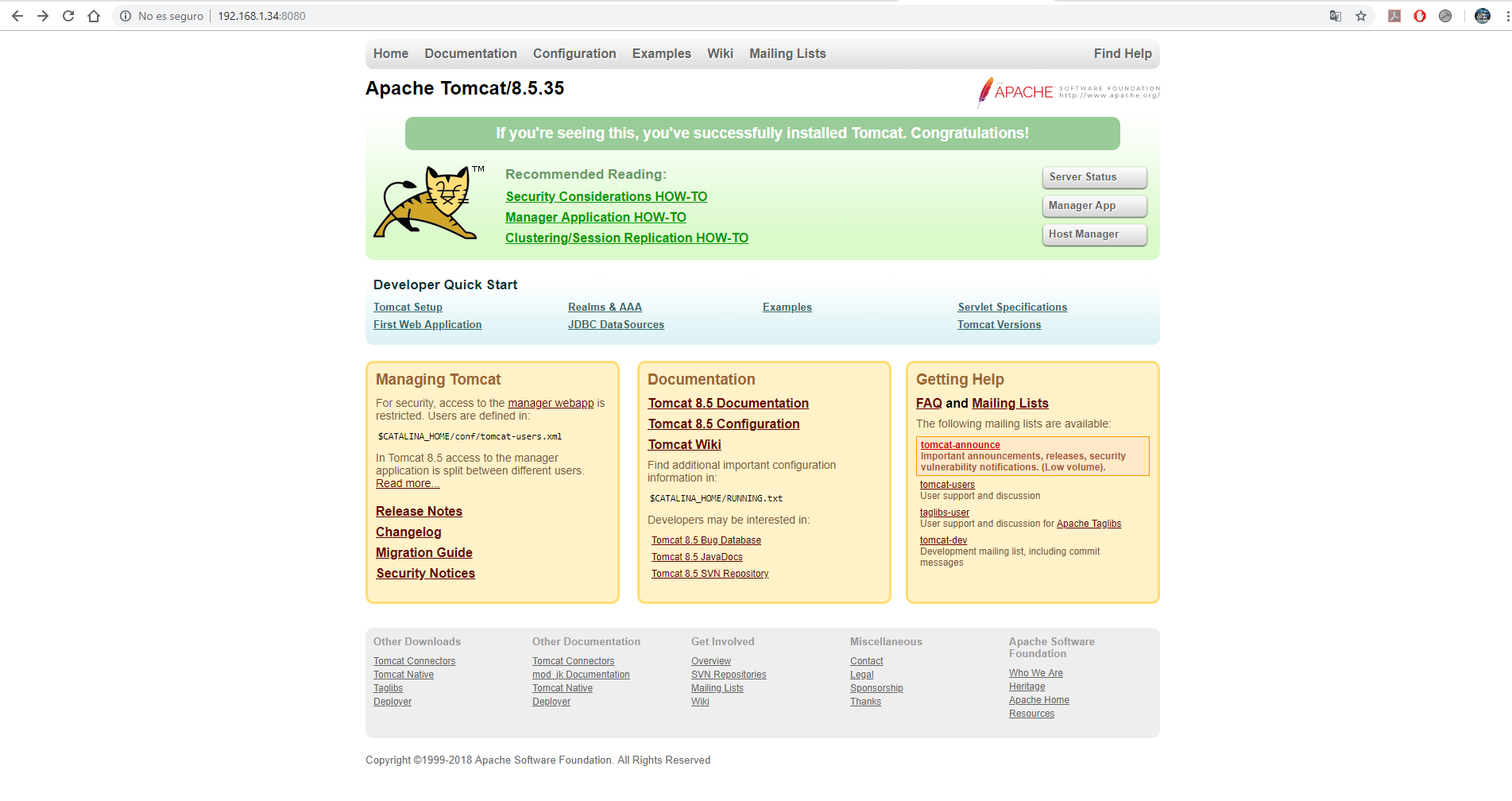
A continuación, recargaremos el proceso systemd para que conozca nuestro archivo de servicio, después reiniciaremos el servicio Tomcat:



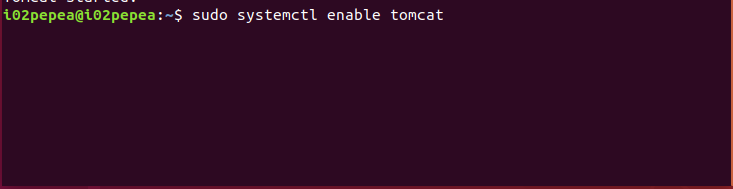
Ahora que el servicio de Tomcat esta encendido habilitaremos el puerto:



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:

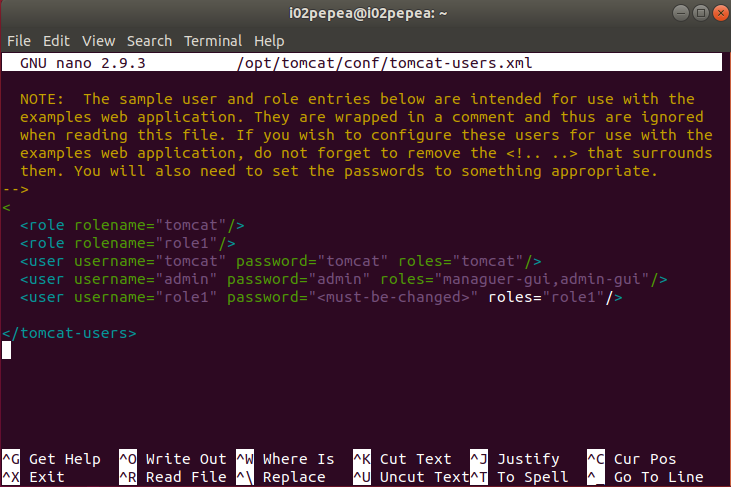


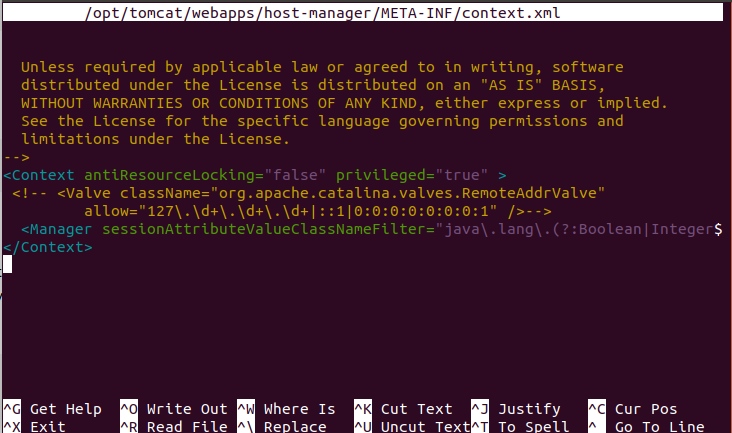
Si todo ha funcionado correctamente, a continuación, habilitaremos Tomcat para que se encienda automáticamente:



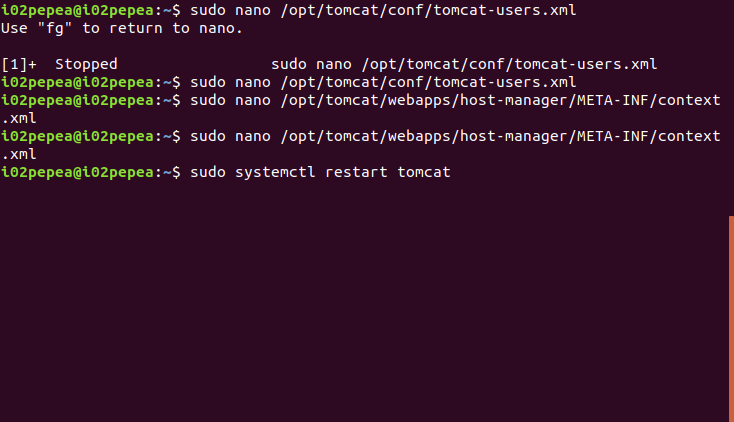
Una vez instalado Tomcat procederemos a configurar el servicio para poder manejarlo y realizar las tareas necesarias.

Para poder manejar Tomcat debemos añadir usuarios y darles privilegios.

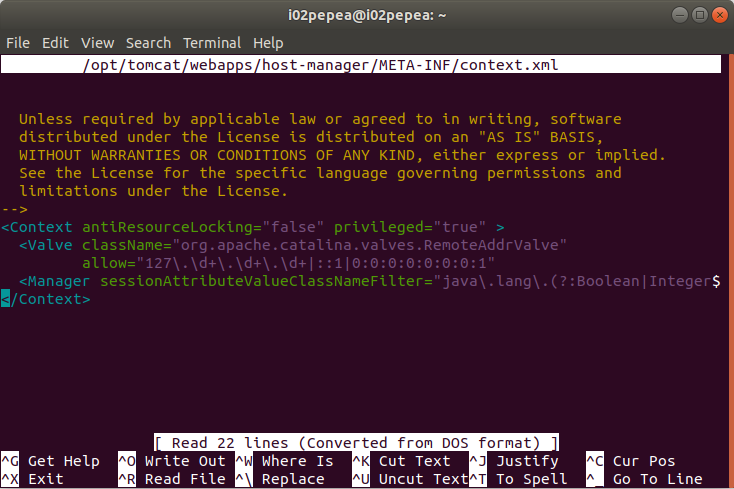




Después de insertar los usuarios y darles privilegios procederemos a reiniciar el servicio para que se realicen los cambios.



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:

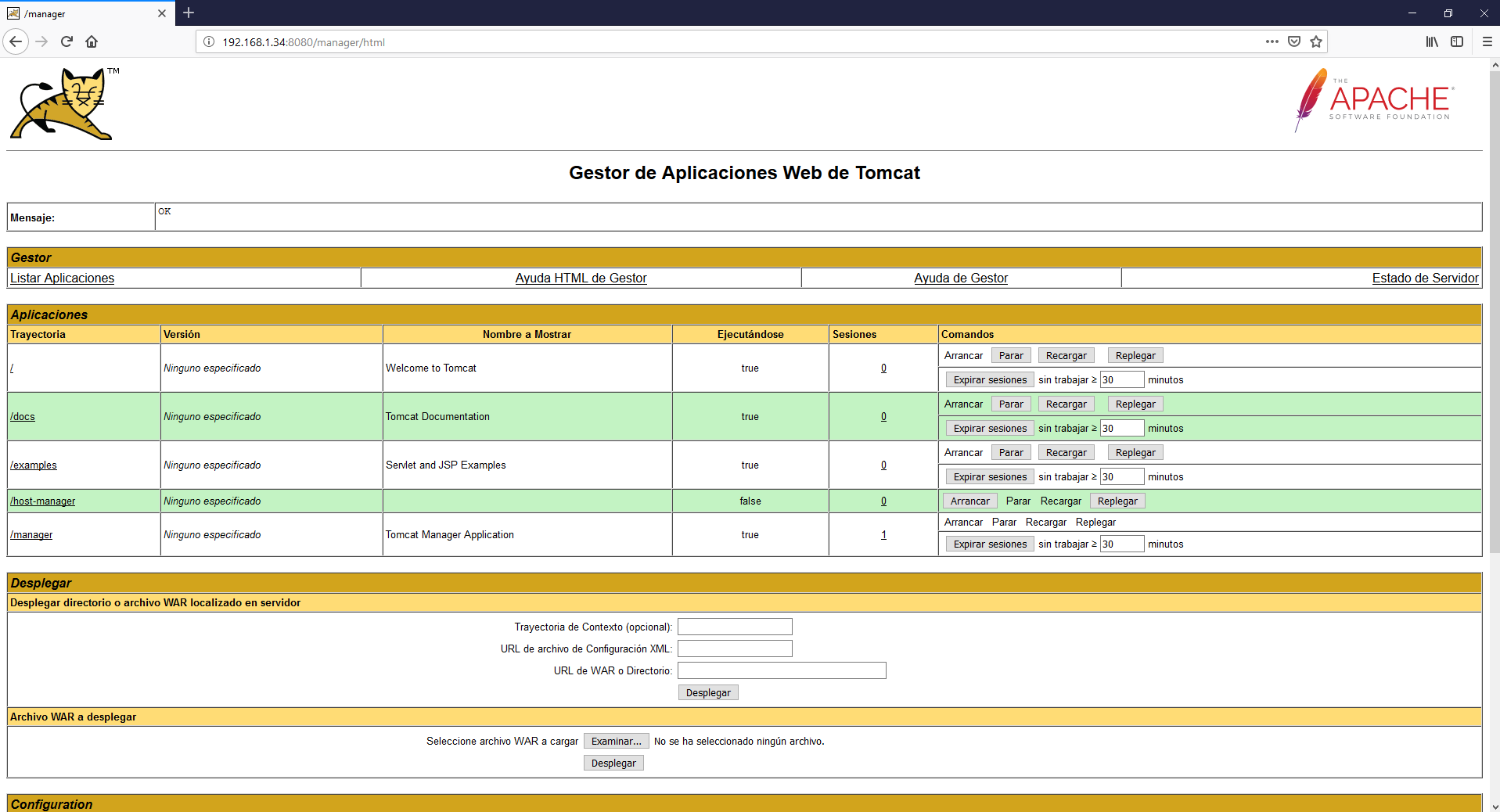


Y después volveremos a usar este comando, para que todos los cambios hagan efecto.

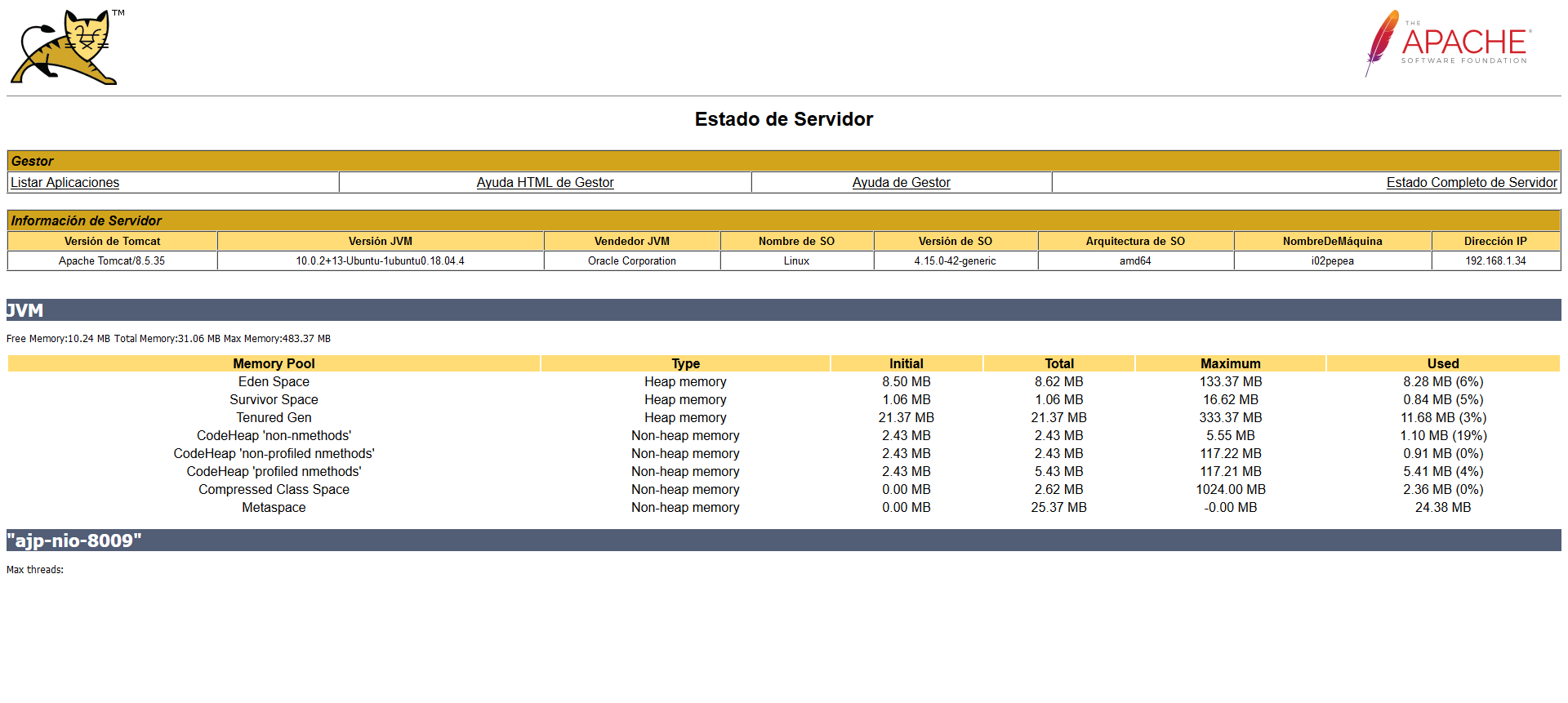
-Sudo systemctl restart Tomcat

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

-Manager-gui



-Estado del servidor:



## Servicios y sus posibles resultados

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

### 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 envió de datos.
* **Envió 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.

### 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.

## 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.

## 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:

### 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 maquina como la otra ya que, si en la maquina anfitriona estas usando el hardware en alto rendimiento, afectara a la máquina virtual y limitara las cargas de prueba que realices.

### 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

# 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.

## 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

### 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 muestro el número de transferencias de entra/salida del dispositivo.

## Carga de trabajo

La carga de trabajo consistirá es 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.

## Diseño de Experimentos

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

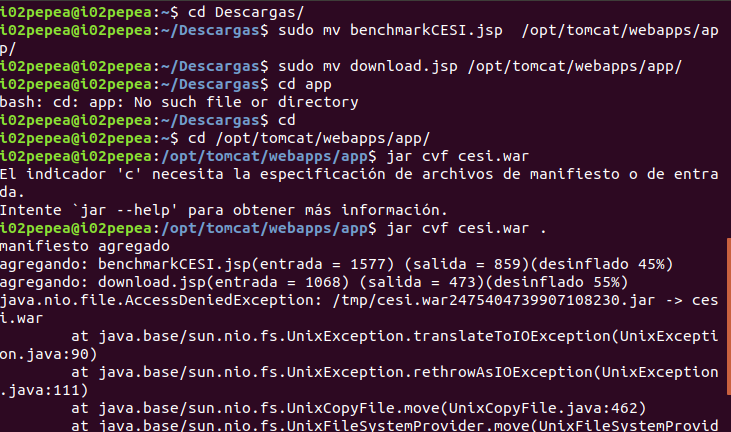
Nuestro principal objetivo es monitorizar la actividad de las métricas explicadas anteriormente con la monitorización correspondiente.

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,

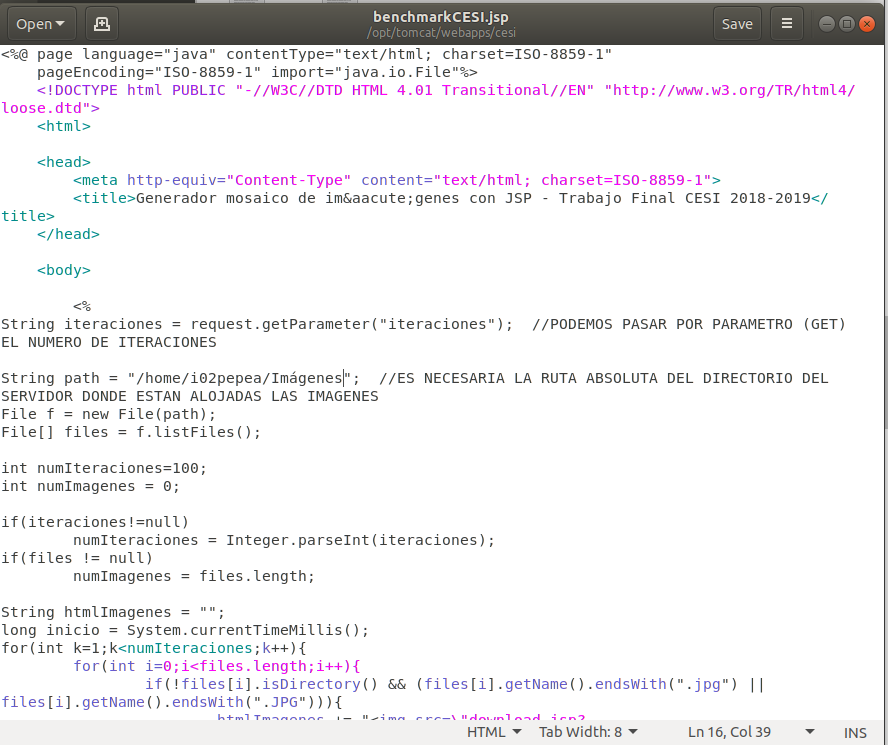
Las localizaciones de los archivos se van a encontrar en el servidor Tomcat instalado y configurado anteriormente, comenzamos.

### Configuración y subida de la Carga de trabajo

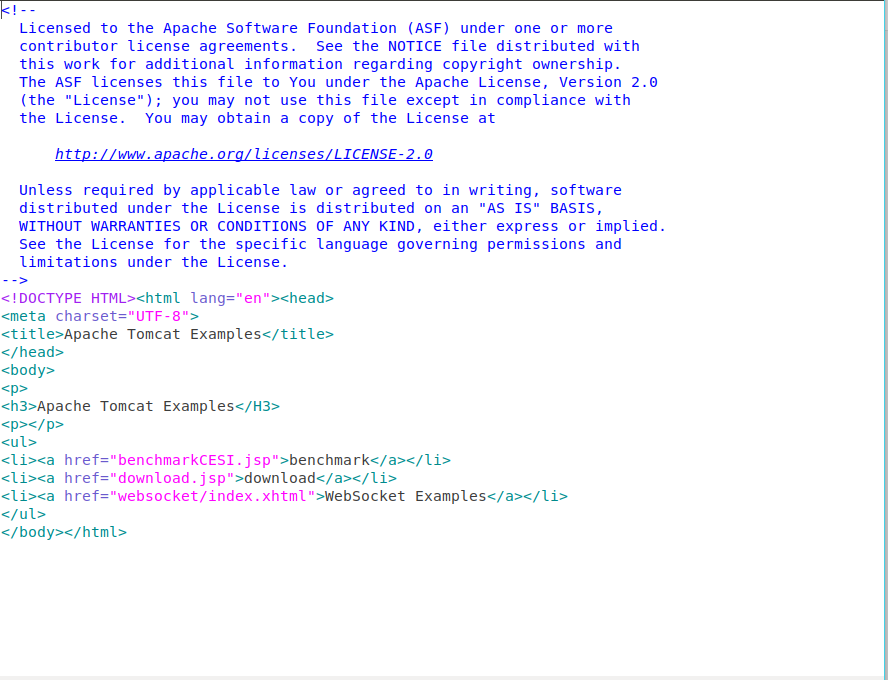
En primer lugar, compilaremos los archivos jsp para que el servidor los pueda leer:



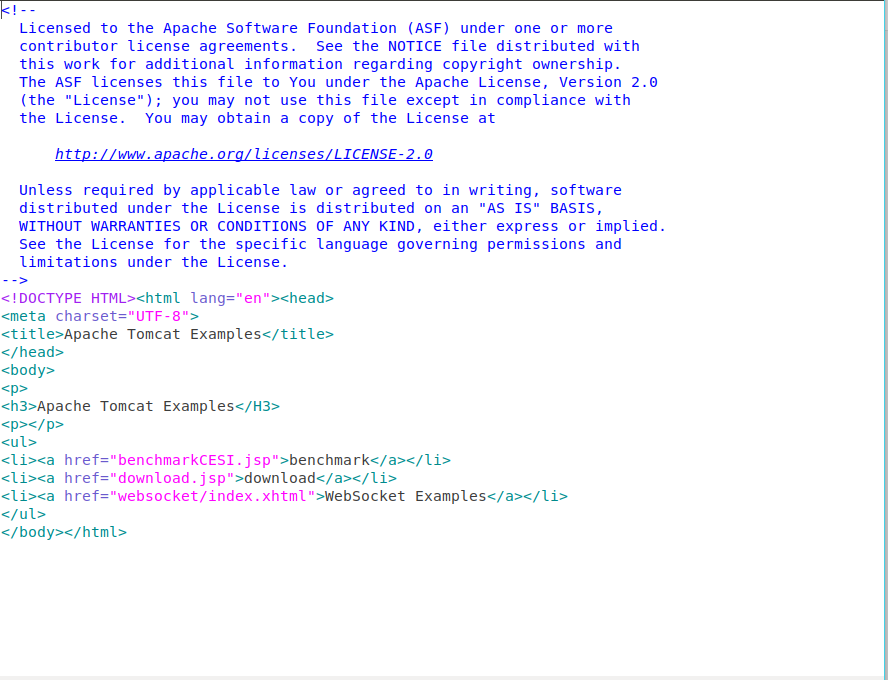
Después modificaremos el archivo benchmarkCESI para poner la ruta donde se encuentran nuestras imágenes:



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



Aquí podemos observar como se ha configurado correctamente:



Y comprobamos que funciona para poder monitorizar:



### 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:

## 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

# 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.

### 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://www.ecured.cu/Servidor_Tomcat>

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

## 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.