

TAREA 3.2 PRUEBAS DE FUNCIONAMIENTO Y RENDIMIENTO CON TOMCAT

Despliegue de
Aplicaciones Web

Contenido

FECHA DE INICIO: 19/12/2024 FECHA DE FINALIZACIÓN ESPERADA: 8/01/2025	2
OBJETIVOS	2
ENUNCIADO.....	2
Requisitos	10
Pasos de la práctica 1. Preparar el entorno.....	11
2. Instalar y configurar JMeter	11
4. Ejecutar la prueba	16
5. Analizar los resultados.....	16
6. Optimizar y repetir pruebas	18
DOCUMENTACIÓN	20

PRÁCTICA 3.2

PRUEBAS DE RENDIMIENTO DE APLICACIÓN WEB DESPLEGADA EN TOMCAT

FECHA DE INICIO: 19/12/2024 **FECHA DE FINALIZACIÓN ESPERADA:** 8/01/2025

RA ASOCIADO: RA3. Implanta aplicaciones web en servidores de aplicaciones evaluando y aplicando criterios de configuración para su funcionamiento seguro. **CEs ASOCIADOS:** d, e, f, g, h.

OBJETIVOS

- Aprender a implantar aplicaciones web en servidores de aplicaciones.
- Habitarse a la descripción del proceso.
- Creación de archivo war con netbeans en Linux usando Ant
- Evaluar el rendimiento de una aplicación web desplegada en un servidor Tomcat.
- Identificar el tiempo de respuesta, errores y capacidad de carga.

ENUNCIADO

Se desarrollará la práctica en un Ubuntu Server que disponga de un servidor de Tomcat instalado y en un Ubuntu Desktop que tenga un IDE Netbeans y un servidor Tomcat sólo para desarrollo para crear el war.

Se facilitan en la plataforma los archivos ejecutables para la instalación del jdk, de netbeans y de tomcat, así como los archivos java (servlets) y jsp necesarios para la construcción del proyecto. Sólo debes cambiar el código para hacer que tras eliminar un artículo de la lista vuelva a mostrar el listado de artículos actualizado. Modifica esto tras construir el proyecto y desplegar en Tomcat.

Se deberá instalar el jdk, Tomcat 10 y Netbeans en el el Ubuntu Desktop. En este equipo se deberá crear el proyecto y generar el archivo war a partir de los archivos facilitados de la aplicación.

En Tomcat (tanto en el servidor como en el Ubuntu Desktop) habrá un usuario con capacidad de gestión de aplicaciones a través de la interfaz web, con credenciales: **useradmin / 4321**.

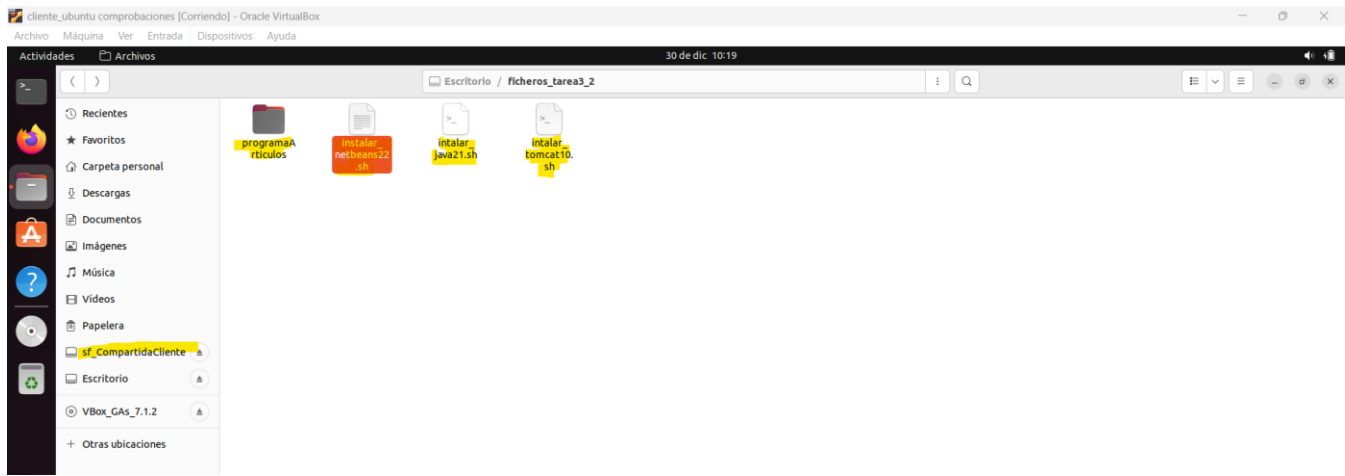
PRIMERA PARTE – Instalación de scripts en la máquina cliente para desarrollo

Para usar los scripts de instalación del JDK, de Netbeans y de Tomcat 10:

1. **Guarda cada script en Linux en un archivo separado (puedes copiarlos como están o usar nano e incluir su contenido):**

- Para el JDK: instalar_java21.sh
- Para NetBeans: instalar_netbeans22.sh
- Para Tomcat 10: instalar_tomcat10.sh

He creado una carpeta compartida desde mi Windows al cliente de Ubuntu, y así puedo acceder a el contenido facilitado:

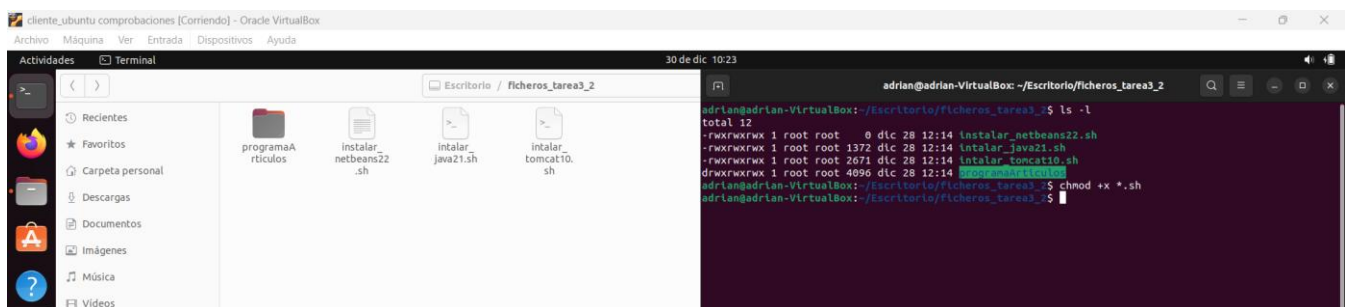


2. **Hazlos ejecutables:**

- `chmod +x instalar_java21.sh instalar_netbeans22.sh instalar_tomcat10.sh`

Accedemos a la terminal en el directorio donde tenemos los archivos y les hacemos ejecutables a todos, utilizando el comando `chmod +x *.sh`

(He modificado los scripts. Estaban incorrectos debido a problemas con los espacios, los nombres de los archivos y errores en su funcionamiento. Además, uno de los scrils pts estaba vacío. Los he corregido y ahora deberían funcionar correctamente.)



3. Ejecuta cada script con permisos de administrador:

- `sudo ./instalar_java21.sh`
- `sudo ./instalar_netbeans22.sh`
- `sudo ./instalar_tomcat10.sh`

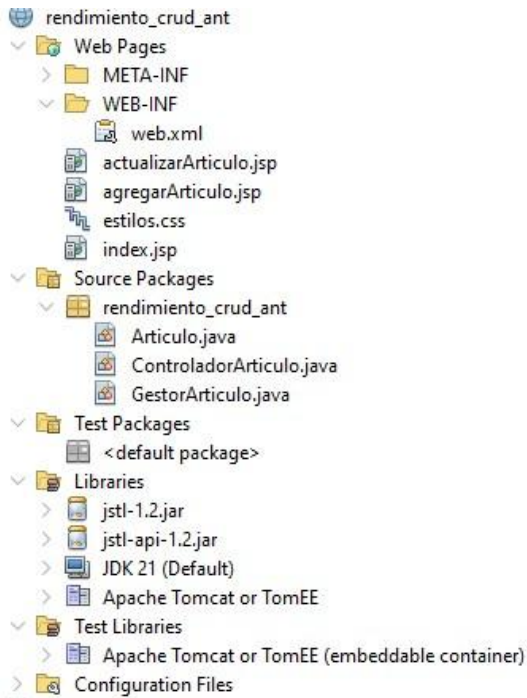
Utilizamos el comando `sudo` para ejecutar cada uno de los scripts:

```
adrian@adrian-VirtualBox:~/Descargas$ sudo ./instalar_java21.sh
sudo ./instalar_tomcat10.sh
sudo ./instalar_netbeans22.sh
```

SEGUNDA PARTE – generar archivo war.

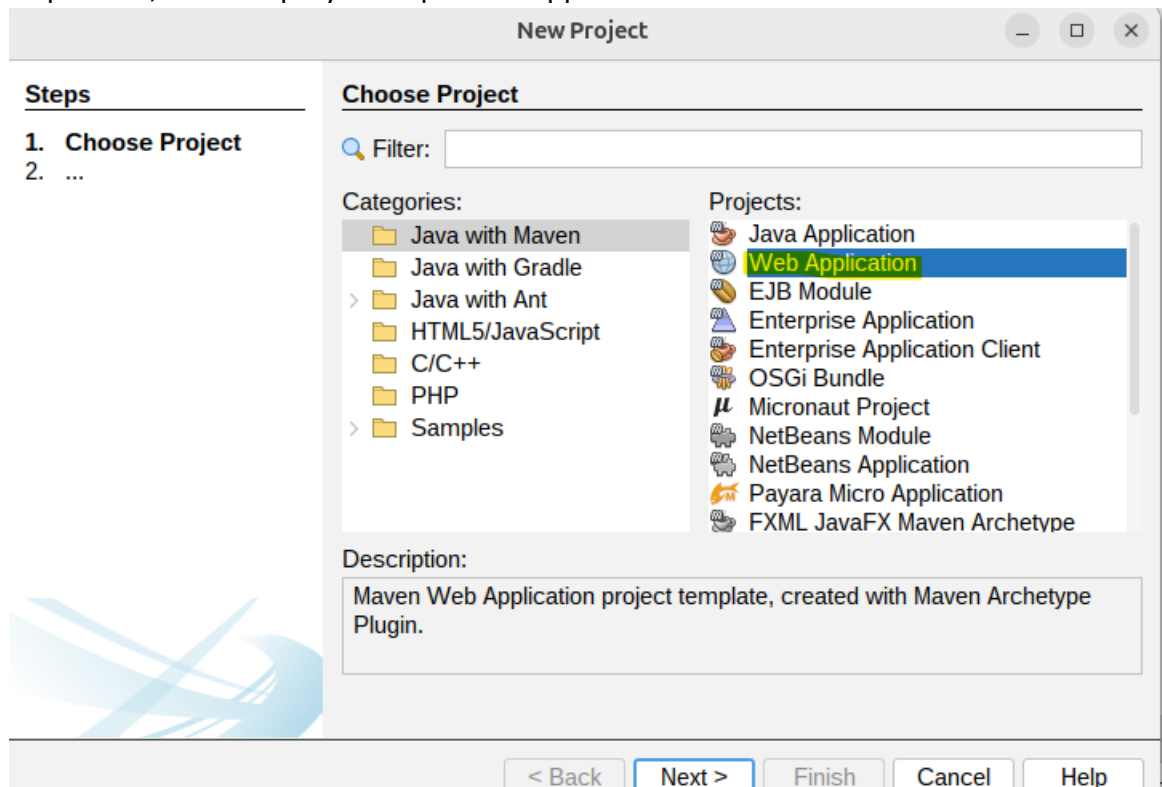
Para la asociación del servidor en Netbeans usaremos como Home y como Base ***/opt/tomcat***. Construye el proyecto de la aplicación con Netbeans, que desplegarás en el servidor Tomcat creado en la práctica anterior. Crea el archivo war en netbeans a partir de los servlets y JSP facilitados.

El proyecto en Netbeans tendrá la siguiente estructura:

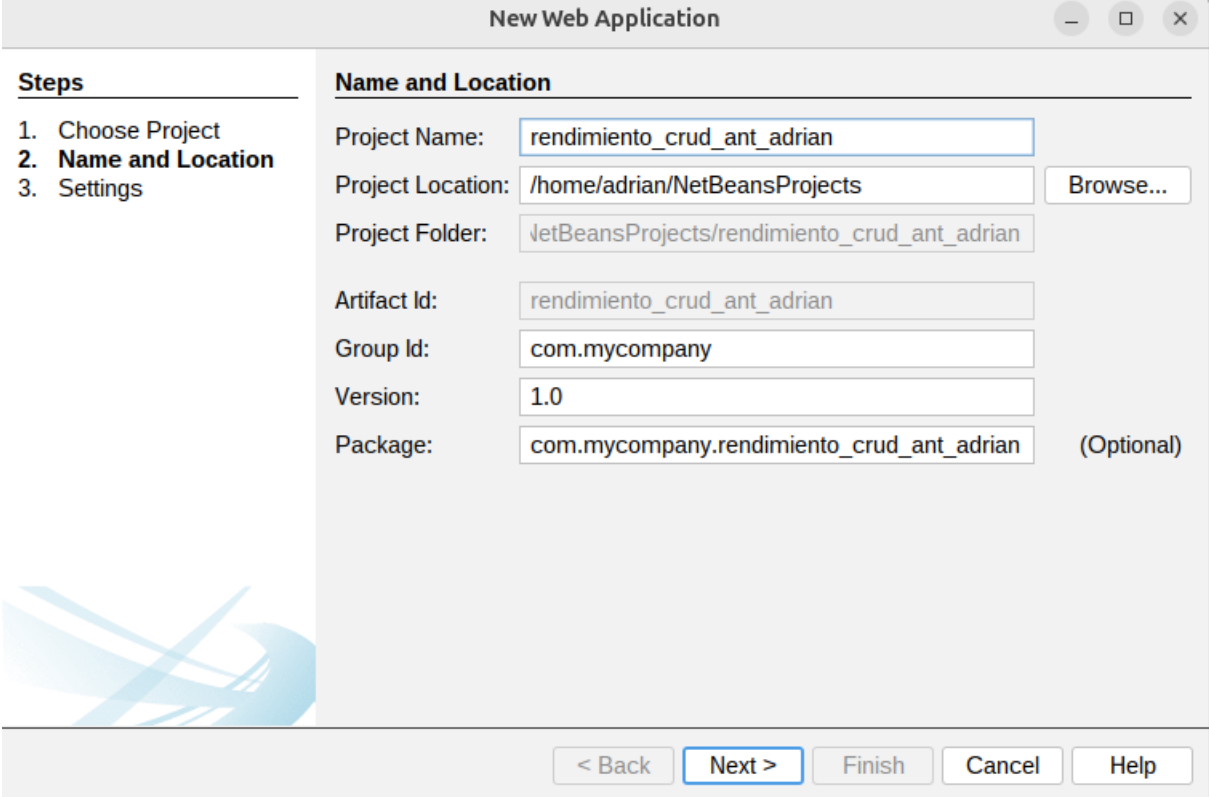


Pasos que he realizado:

Lo primero, crear un proyecto tipo Web Application:

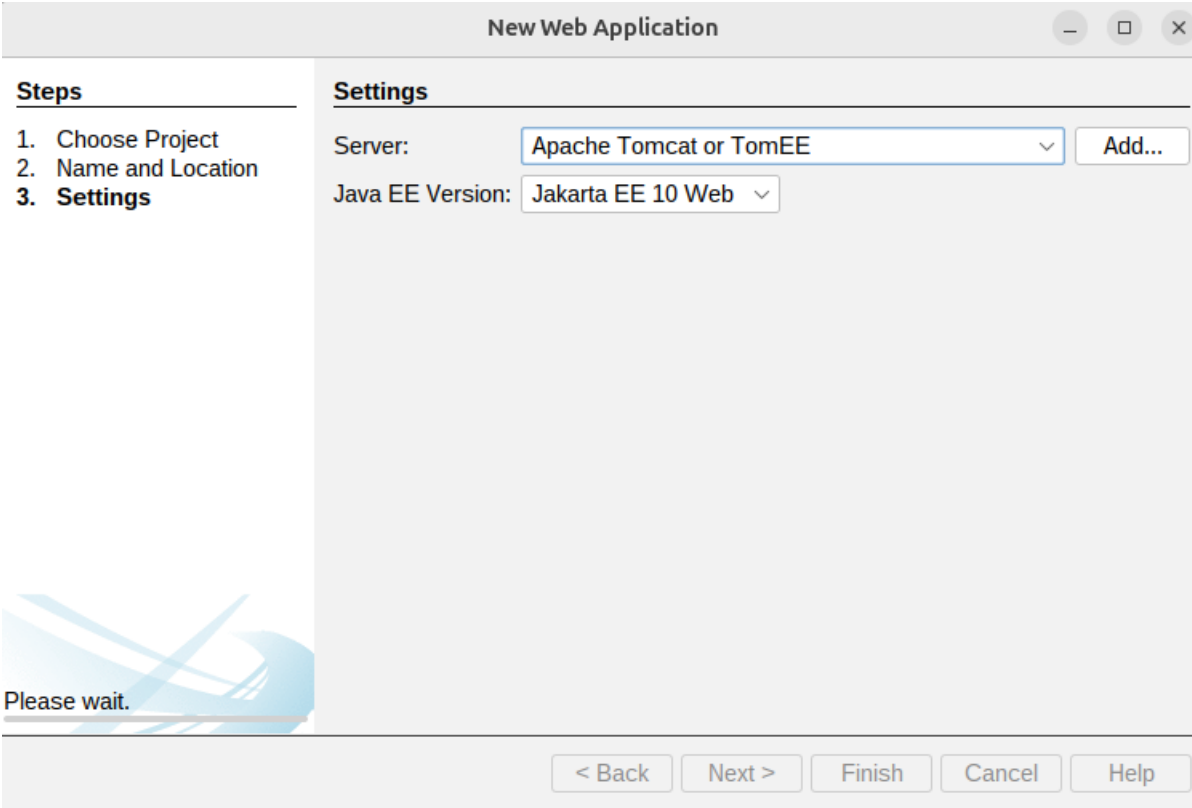


Le damos un nombre:

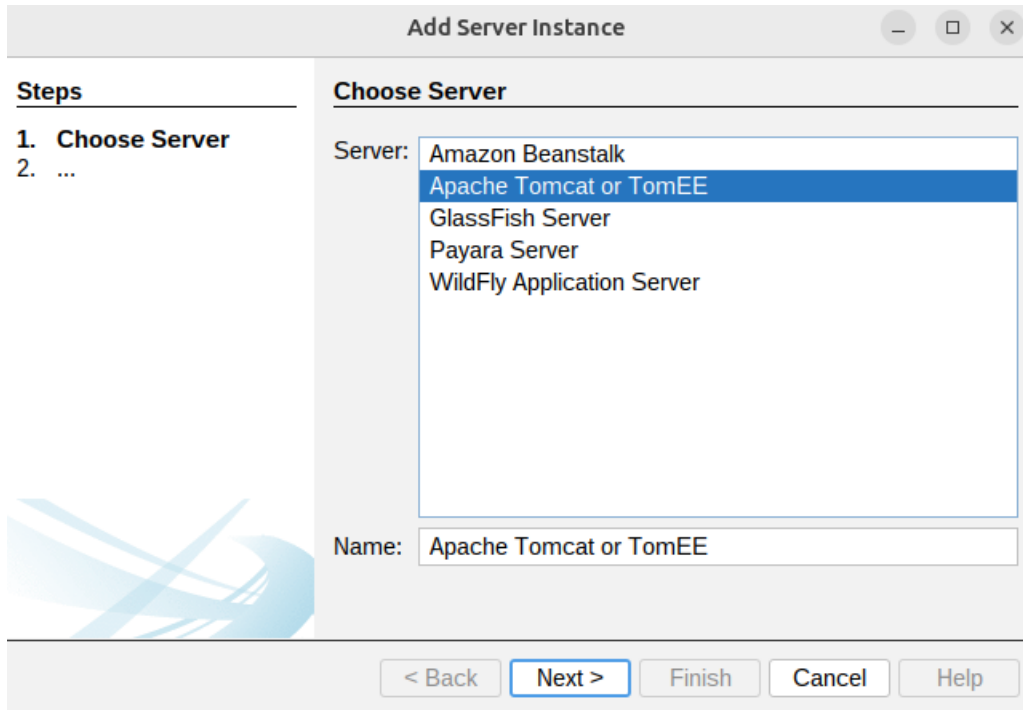


The dialog box is titled "New Web Application". On the left, a "Steps" list shows: 1. Choose Project, 2. **Name and Location**, and 3. Settings. The main area is titled "Name and Location" and contains several input fields: "Project Name" with the value "rendimiento_crud_ant_adrian", "Project Location" with "/home/adrian/NetBeansProjects" and a "Browse..." button, "Project Folder" with "NetBeansProjects/rendimiento_crud_ant_adrian", "Artifact Id" with "rendimiento_crud_ant_adrian", "Group Id" with "com.mycompany", "Version" with "1.0", and "Package" with "com.mycompany.rendimiento_crud_ant_adrian" (marked as optional). At the bottom are buttons for "< Back", "Next >", "Finish", "Cancel", and "Help".

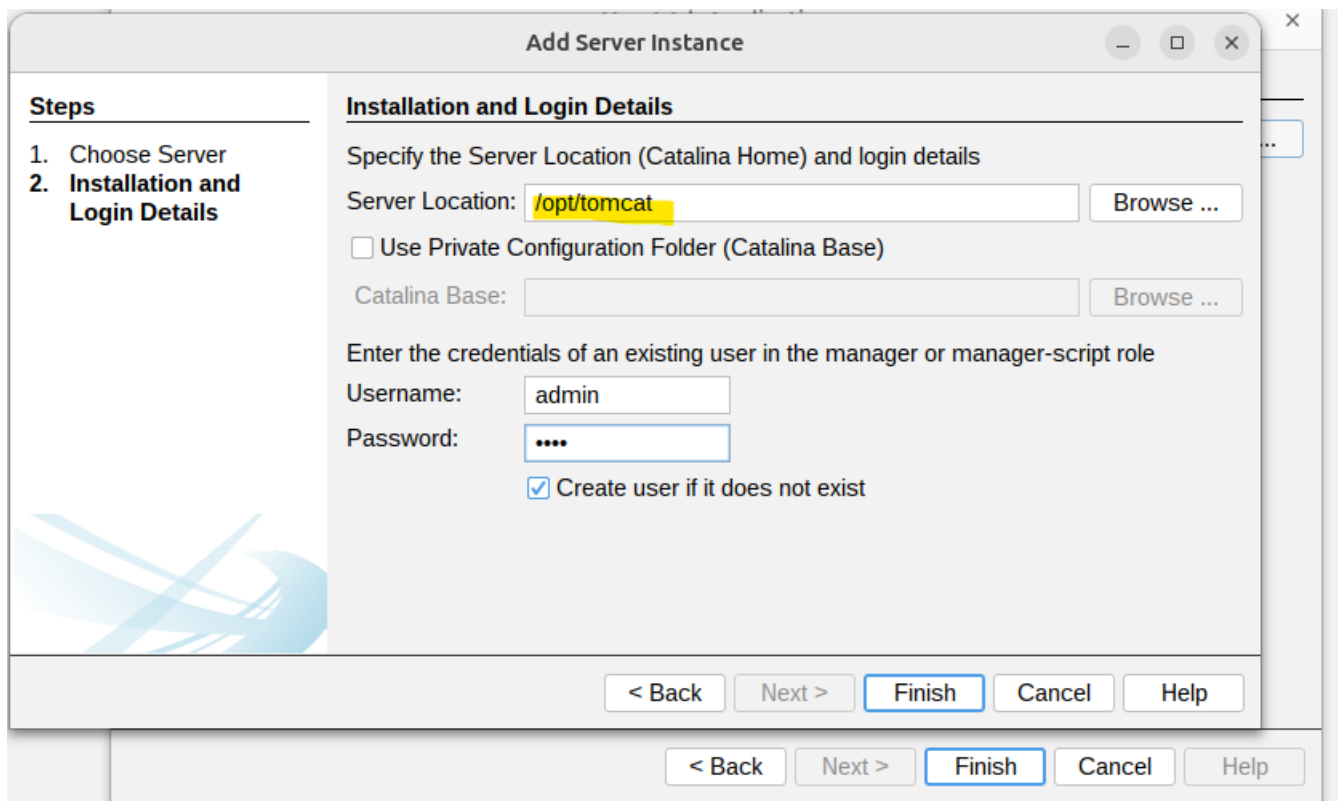
Y seleccionamos Apache como tipo de servidor:



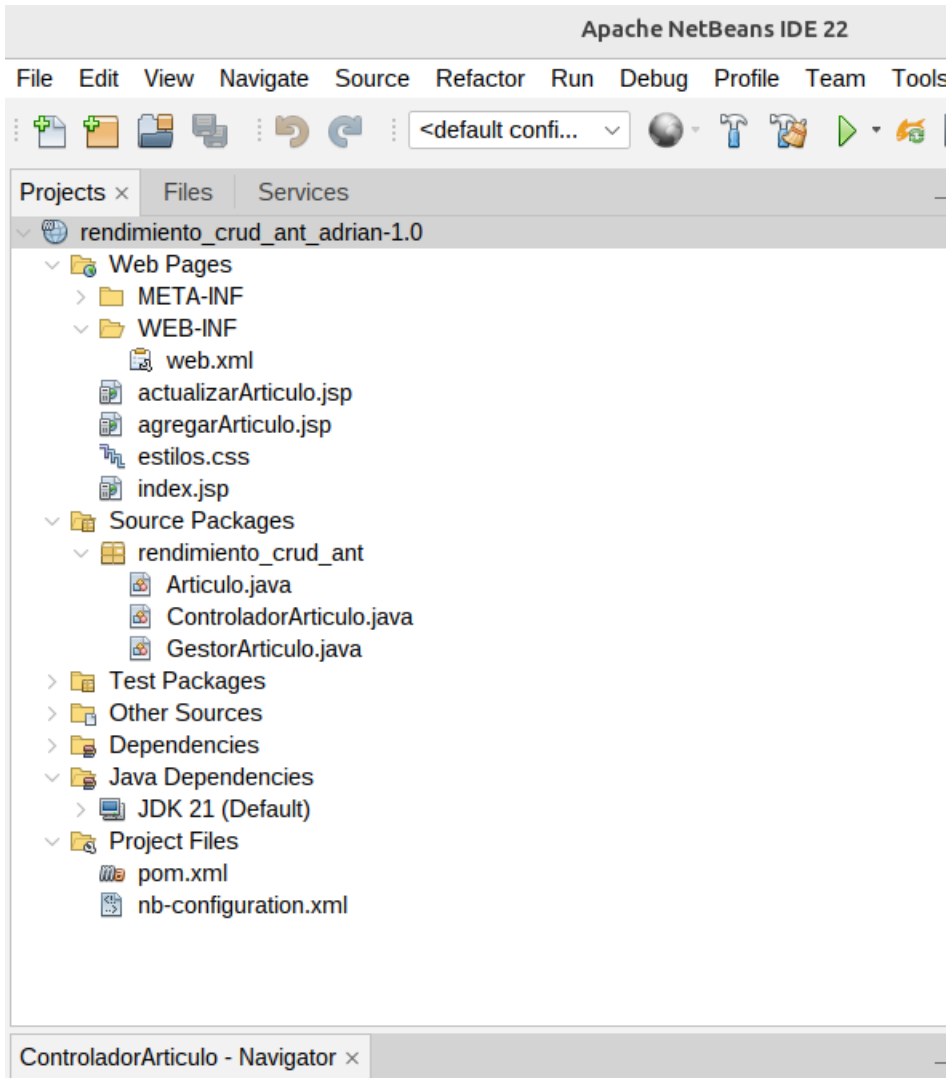
The dialog box is titled "New Web Application". On the left, a "Steps" list shows: 1. Choose Project, 2. Name and Location, and 3. **Settings**. The main area is titled "Settings" and contains two dropdown menus: "Server" with "Apache Tomcat or TomEE" selected and an "Add..." button, and "Java EE Version" with "Jakarta EE 10 Web" selected. At the bottom are buttons for "< Back", "Next >", "Finish", "Cancel", and "Help". A "Please wait." message is visible in the bottom left corner of the dialog area.



Elegimos como base para nuestro proyecto “/opt/tomcat”, e indicamos el nombre de usuario y la contraseña del usuario administrador:



Mi estructuración:



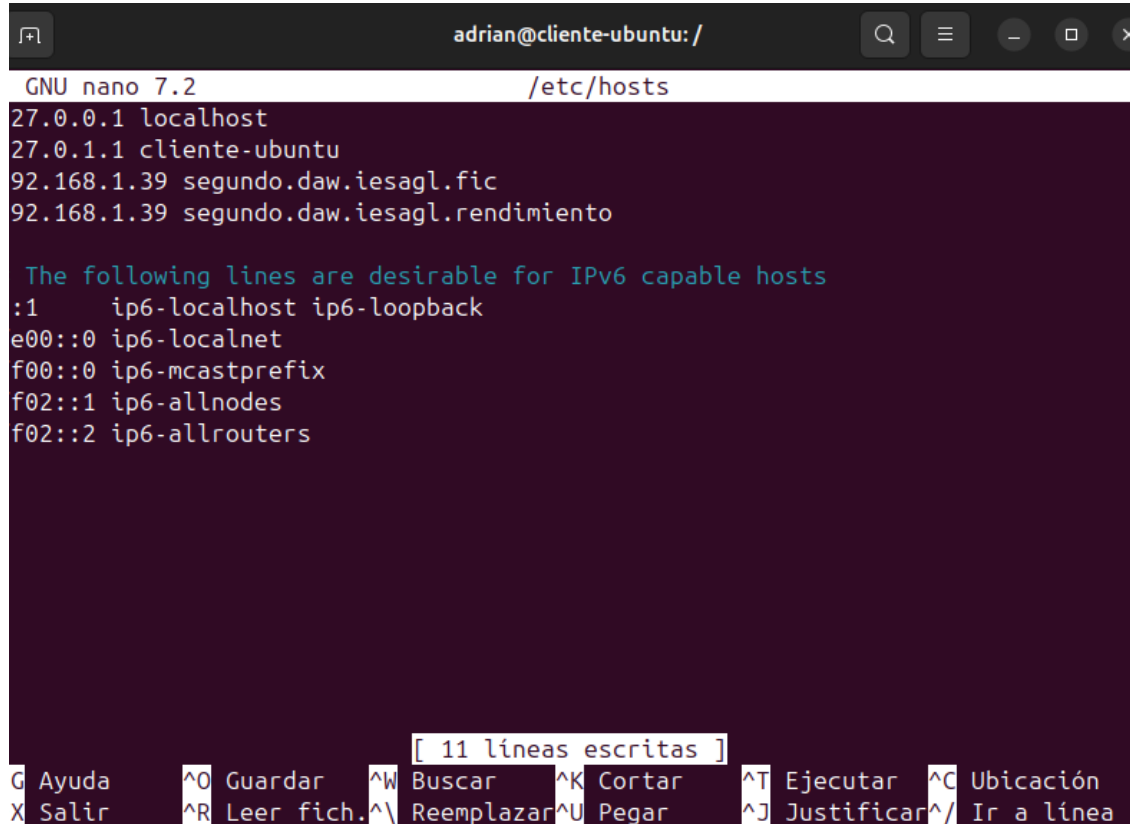
TERCERA PARTE - Desplegar la aplicación.

Despliega la aplicación en el servidor Tomcat creado en la práctica anterior.

Se utilizará como nombre de dominio: **segundo.daw.iesagl.rendimiento**.

El servidor tendrá al menos una de sus interfaces en MODO PUENTE, de forma que se pueda acceder desde cualquier equipo del aula. Puedes utilizar el servidor creado en la práctica 1.

Agregamos la IP de nuestro servidor seguida del nombre de dominio que se indica en la práctica:



The screenshot shows a terminal window with the nano text editor open to the file `/etc/hosts`. The terminal title bar indicates the user is `adrian@cliente-ubuntu`. The file content is as follows:

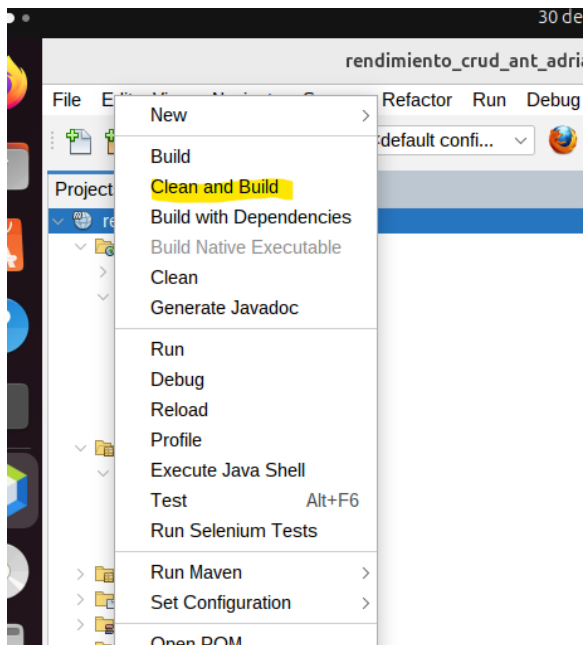
```
GNU nano 7.2 /etc/hosts
27.0.0.1 localhost
27.0.1.1 cliente-ubuntu
92.168.1.39 segundo.daw.iesagl.fic
92.168.1.39 segundo.daw.iesagl.rendimiento

The following lines are desirable for IPv6 capable hosts
::1          ip6-localhost ip6-loopback
fe00::0      ip6-localnet
ff00::0      ip6-mcastprefix
ff02::1      ip6-allnodes
ff02::2      ip6-allrouters
```

At the bottom of the terminal, a status bar shows `[11 líneas escritas]` and a keyboard shortcuts menu with the following options:

G Ayuda	^O Guardar	^W Buscar	^K Cortar	^T Ejecutar	^C Ubicación
X Salir	^R Leer fich.	^E Reemplazar	^U Pegar	^J Justificar	^_ Ir a línea

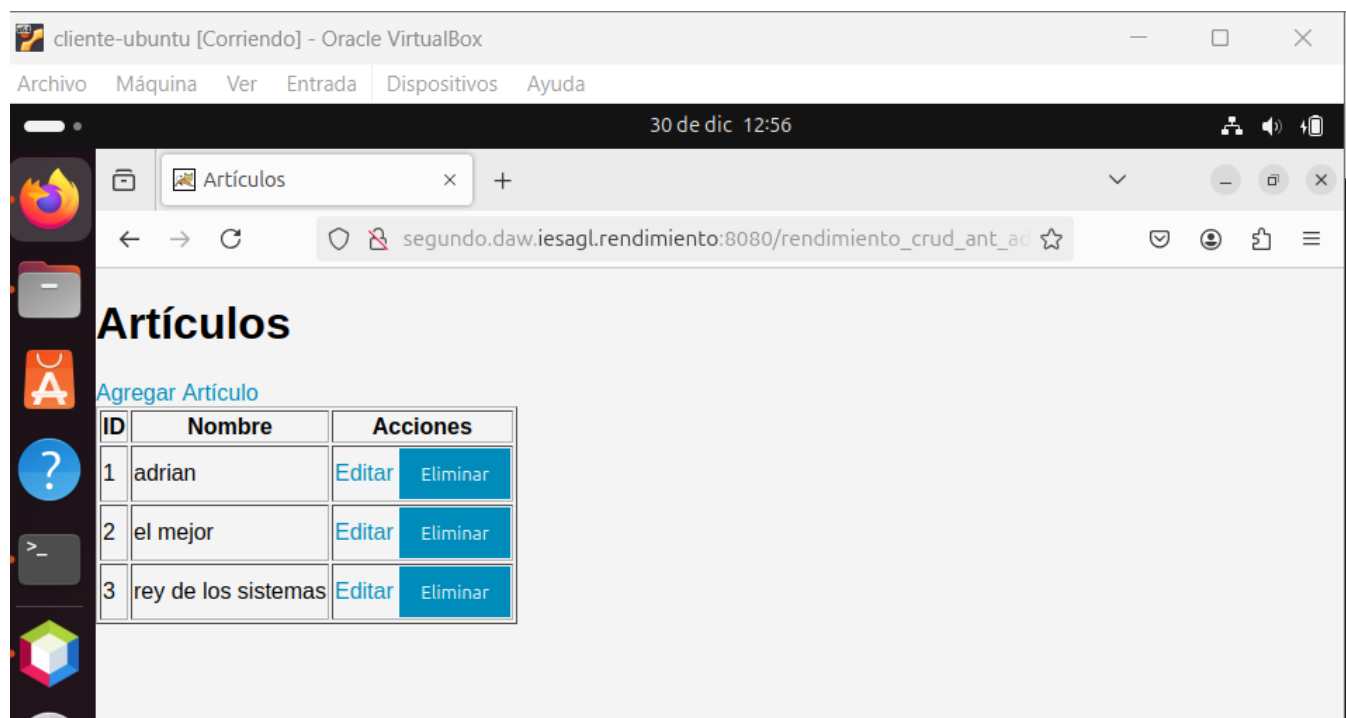
Hacemos clean and build en nuestro proyecto:



Y utilizamos el comando scp para copiar nuestro archivo WAR al servidor, y desplegarlo en Tomcat:

```
adrian@cliente-ubuntu: ~/NetBeansProjects/rendimiento_crud_ant_adria...
adrian@cliente-ubuntu:~/NetBeansProjects/rendimiento_crud_ant_adrian/target$ scp
/home/adrian/NetBeansProjects/rendimiento_crud_ant_adrian/target/rendimiento_cr
ud_ant_adrian-1.0.war adrian@192.168.1.39:/opt/tomcat/webapps/
The authenticity of host '192.168.1.39 (192.168.1.39)' can't be established.
ED25519 key fingerprint is SHA256:ARUPLJIqhFnwqyJjTKXuVTQaufavrl7iy1nCZgHdsLM.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.1.39' (ED25519) to the list of known hosts.
adrian@192.168.1.39's password:
rendimiento_crud_ant_adrian-1.0.war      100% 8823    9.5MB/s   00:00
adrian@cliente-ubuntu:~/NetBeansProjects/rendimiento_crud_ant_adrian/target$
```

Abrimos en un navegador el dominio que hemos configurado, para verificar que funciona correctamente:



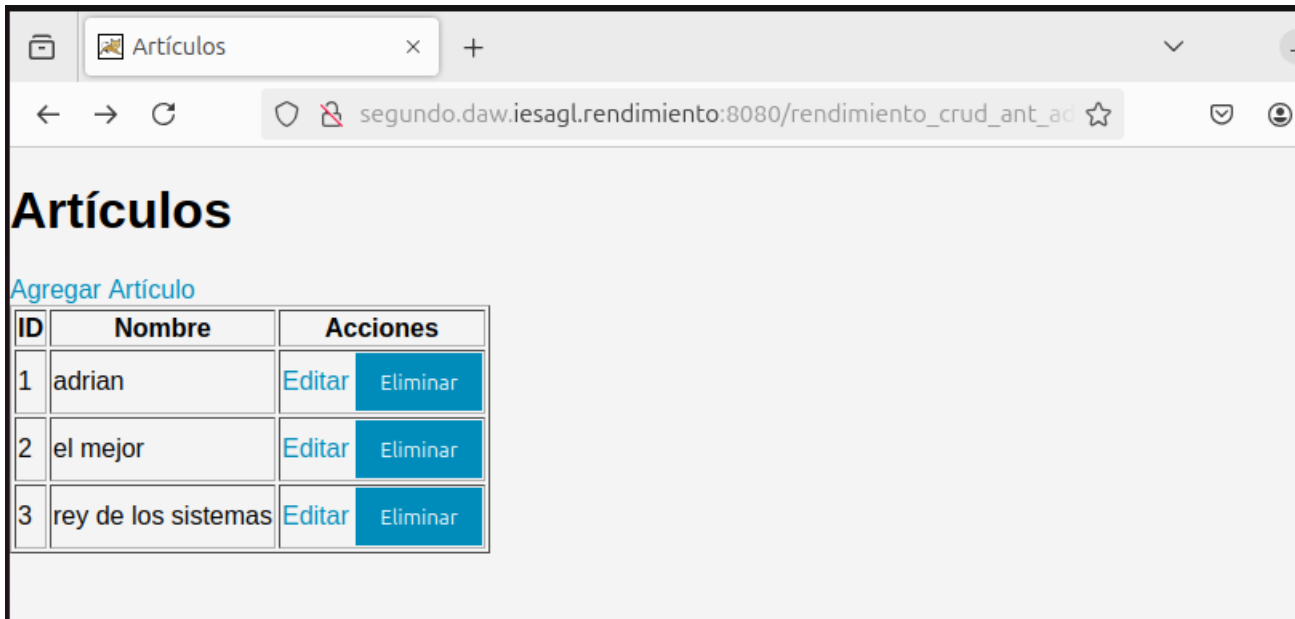
CUARTA PARTE- Pruebas de rendimiento con Apache JMeter

Requisitos

1. Un servidor con Tomcat instalado y la aplicación web desplegada.
2. Apache JMeter instalado en el equipo que realizará las pruebas.
3. Configuración de red que permita el acceso al servidor Tomcat desde JMeter.

Pasos de la práctica 1. Preparar el entorno

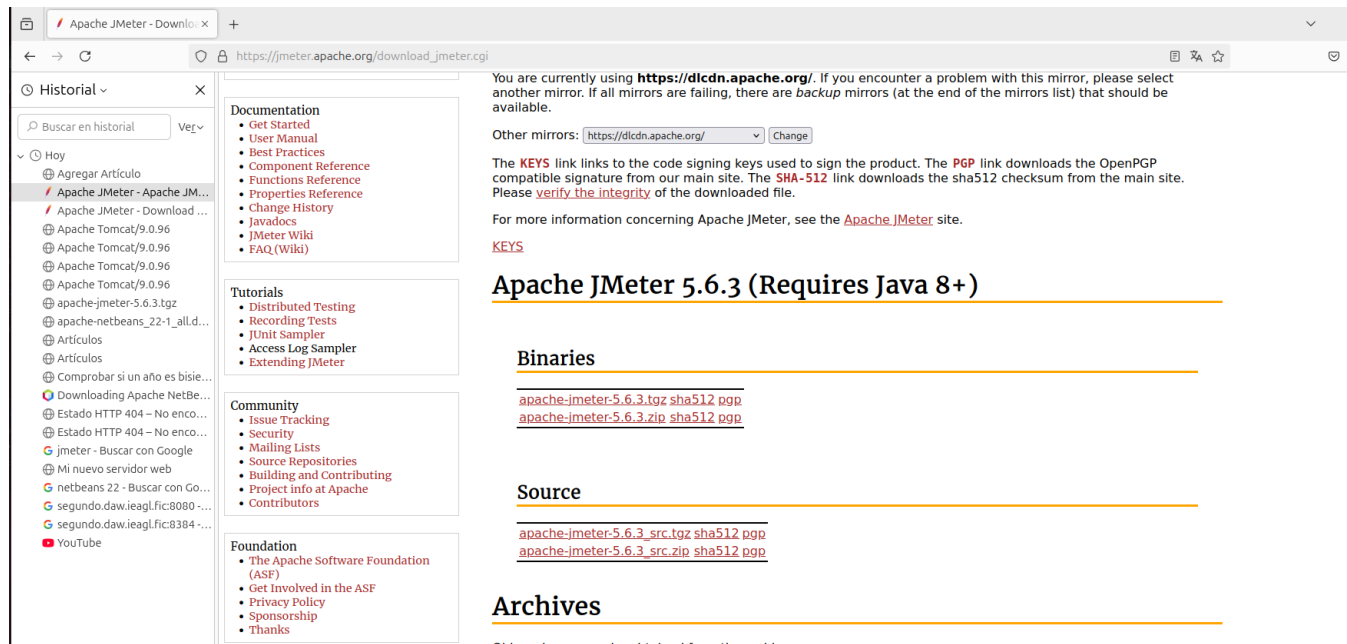
- Asegúrate de que Tomcat esté en funcionamiento y que la aplicación sea accesible desde un navegador.
- Verifica la URL base de la aplicación web (por ejemplo, `http://<server_ip>:8080/app`).



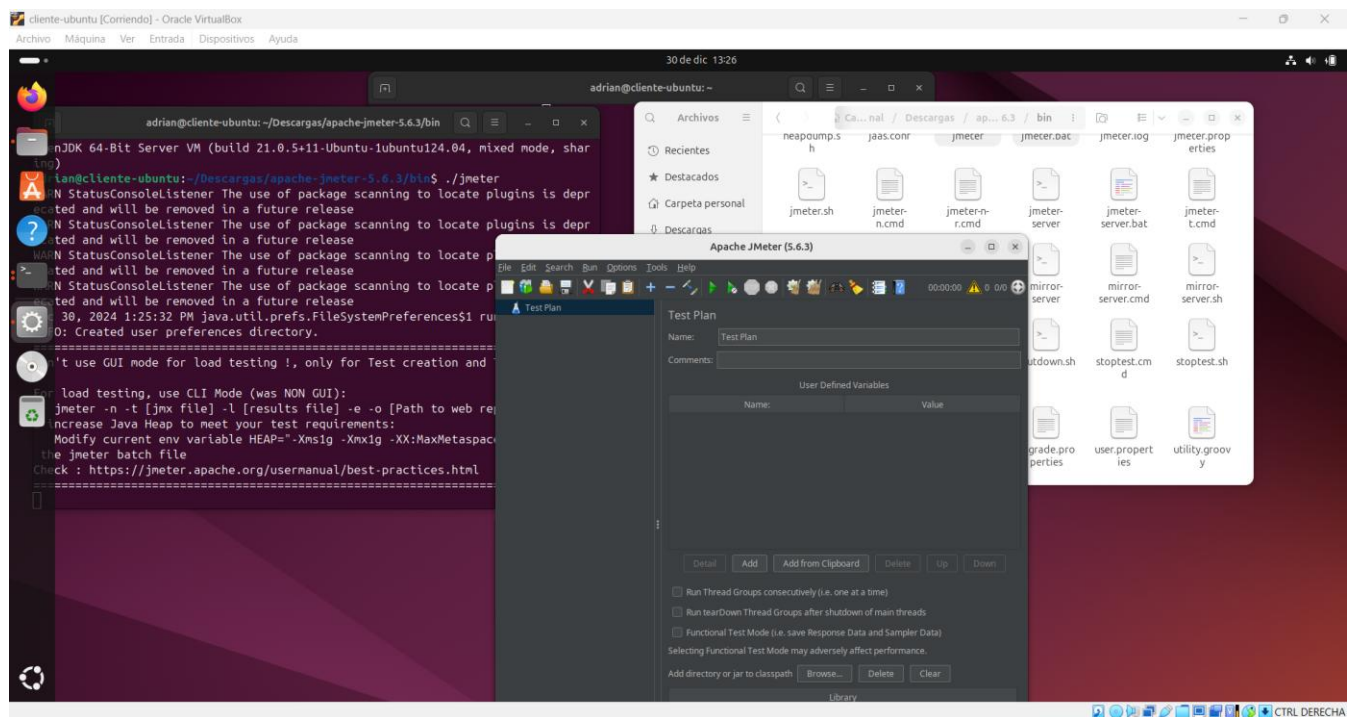
2. Instalar y configurar JMeter

- Descarga Apache JMeter desde [su página oficial](#).
- Extrae el archivo descargado y ejecuta el archivo `jmeter.bat` (Windows) o `jmeter` (Linux/MacOS) para abrir la interfaz gráfica.

Para ello, accedemos a la página oficial de apache para descargar Apache JMeter:



Una vez lo hayamos descargado y extraído, navegamos a la carpeta bin y ejecutamos JMeter con el comando `./jmeter`, lo cual nos abrirá la interfaz gráfica de JMeter:



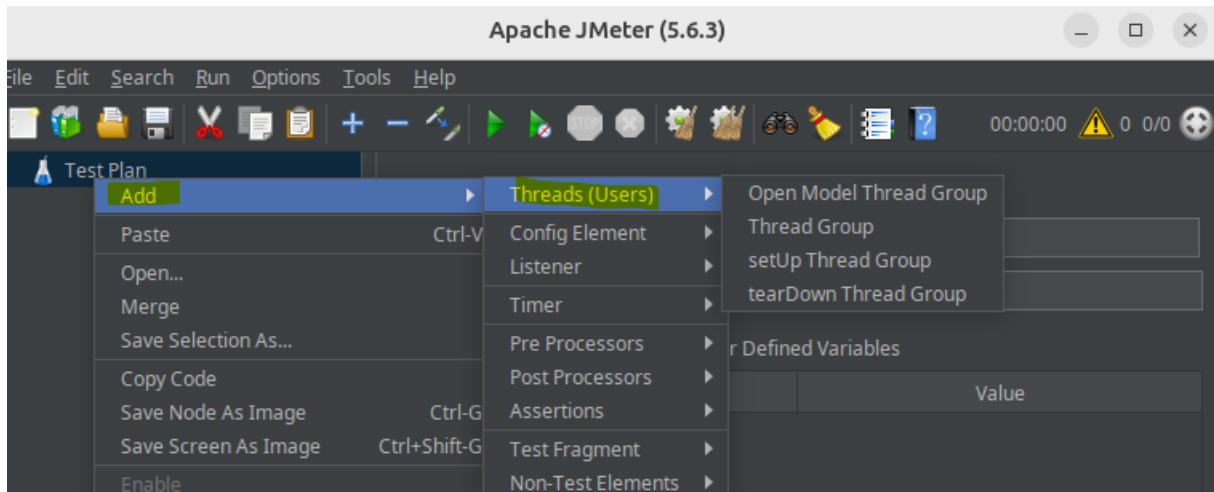
3. Diseñar el plan de prueba en JMeter 1.

Crear un grupo de usuarios:

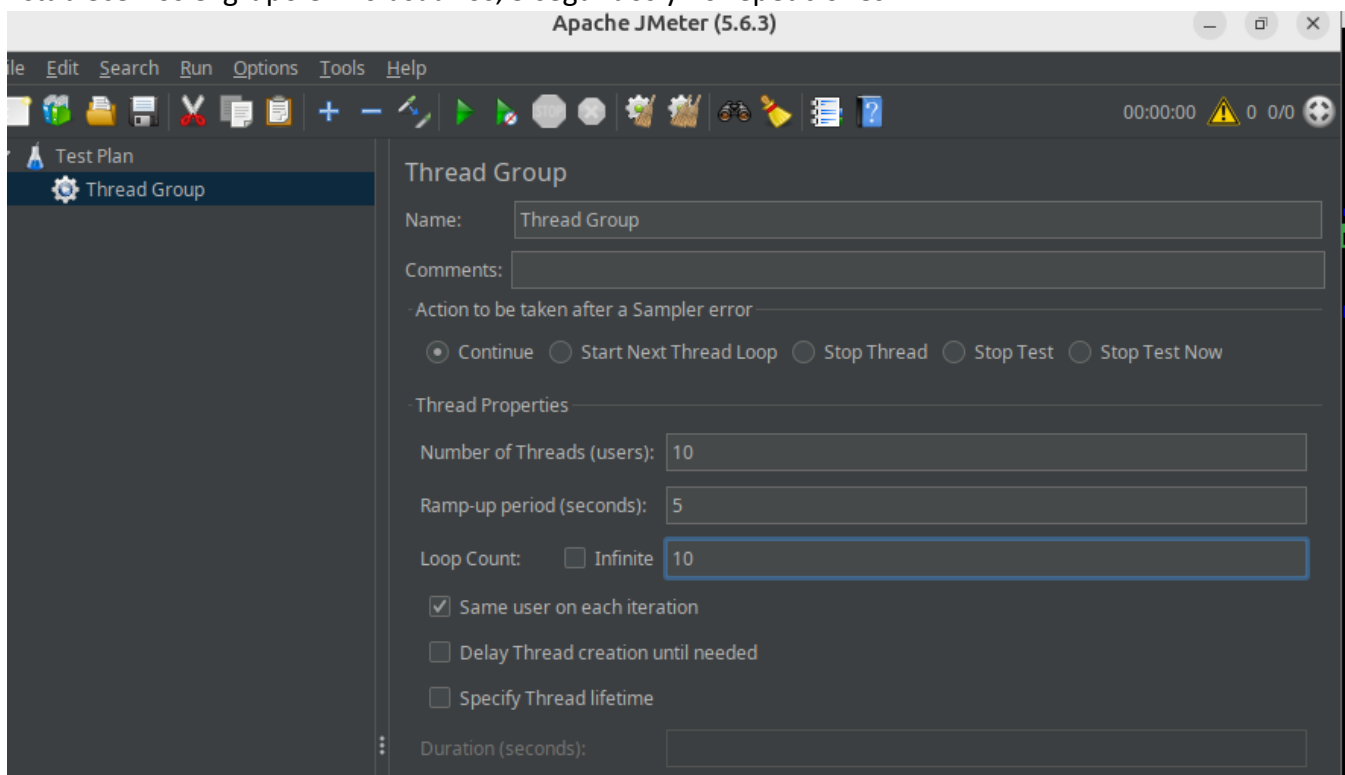
- Haz clic derecho en Test Plan > Add > Threads (Users) > Thread Group.
- Configura:

- Número de Threads (Usuarios): Por ejemplo, 10 (incrementa progresivamente en pruebas posteriores).
- Ramp-Up Period: 5 (segundos).
- Loop Count: 10 (cantidad de repeticiones).

Siguiendo los pasos que se nos indica, creamos un grupo de usuarios:



Establecemos el grupo en 10 usuarios, 5 segundos y 10 repeticiones:

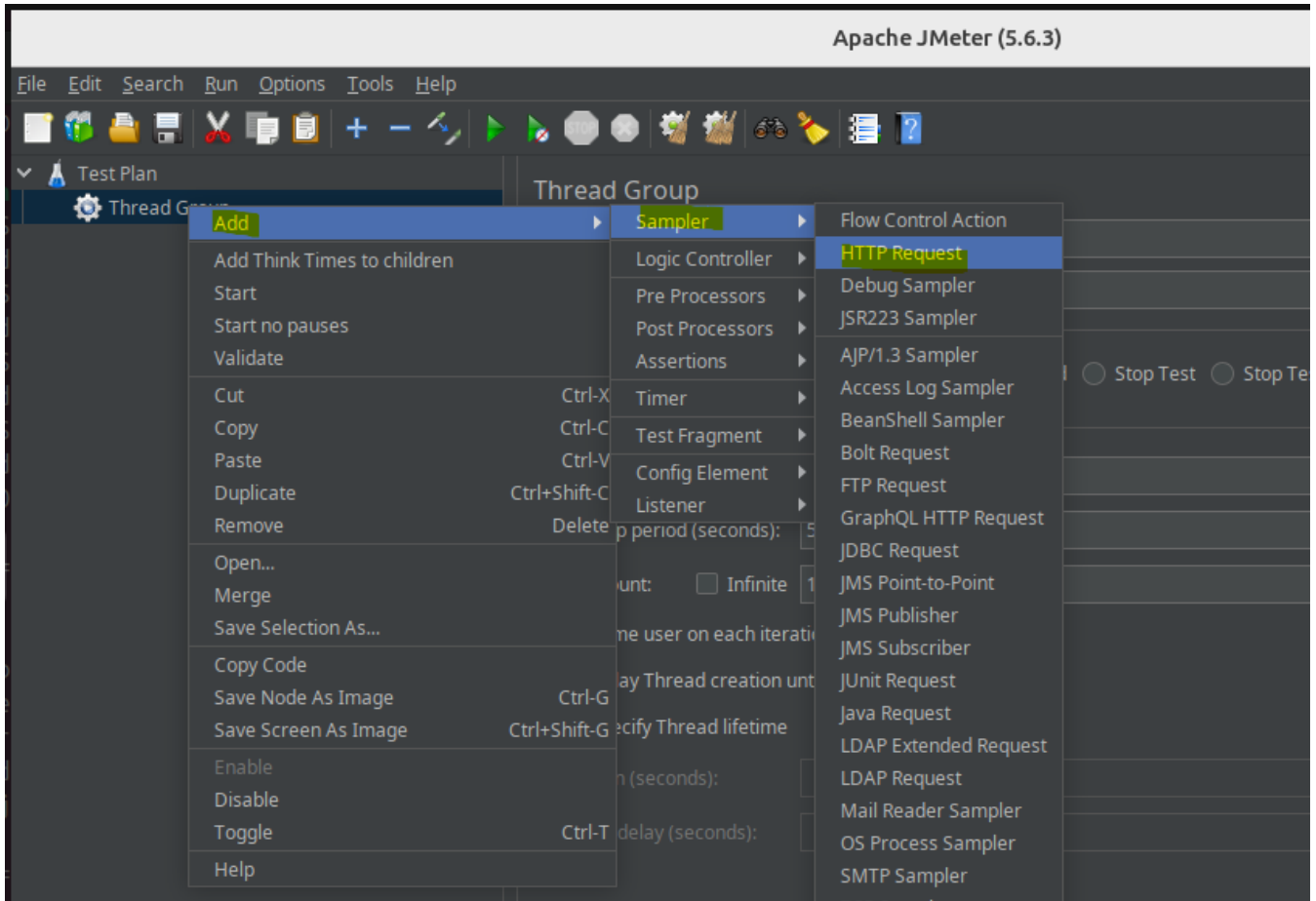


2. Añadir un sampler HTTP Request:

- Haz clic derecho en Thread Group > Add > Sampler > HTTP Request.

- Configura:
 - Server Name or IP: IP o dominio del servidor Tomcat.
 - Path: Ruta relativa de la aplicación (por ejemplo, /app/home). □
 - Método HTTP: GET o POST según la funcionalidad.

De nuevo, seguimos los pasos indicados para añadir un sampler HTTP request:



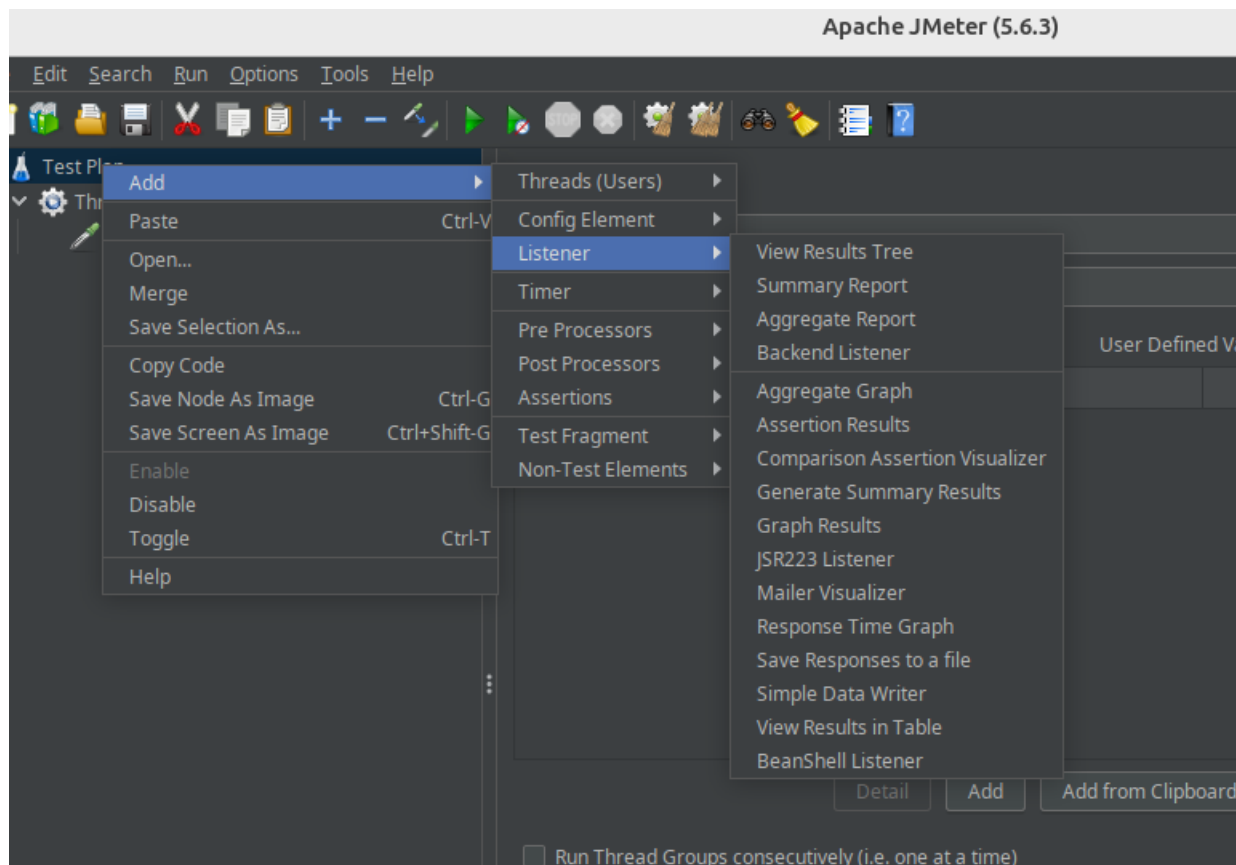
Y lo configuramos de la siguiente forma:

The screenshot shows the 'HTTP Request' configuration window in Apache JMeter. The 'Name' field is set to 'HTTP Request'. The 'Basic' tab is selected. Under 'Web Server', the 'Protocol' is 'http', the 'Server Name or IP' is '/rendimiento_crud_ant_adrian-1.0/', and the 'Port Number' is '8080'. The 'HTTP Request' section shows the 'Method' as 'GET' and the 'Path' as 'http://segundo.daw.iesagl.rendimiento:8080/rendimiento_crud_ant_adrian-1.0/'. There are checkboxes for 'Redirect Automatically', 'Follow Redirects', 'Use KeepAlive', 'Use multipart/form-data', and 'Browser-compatible headers'. Below this, there are tabs for 'Parameters', 'Body Data', and 'Files Upload'. The 'Parameters' tab is active, showing a table with columns: 'Name', 'Value', 'URL Encode?', 'Content-Type', and 'Include Equals?'. The table is currently empty.

3. Añadir un listener para visualizar resultados:

- Haz clic derecho en Test Plan > Add > Listener > View Results in Table.

Por último, añadimos el listener:



4. Ejecutar la prueba

- Presiona el botón verde "Iniciar" en la parte superior.
- Observa los resultados en el listener View Results in Table o en gráficos como Graph Results.

5. Analizar los resultados

- **Tiempo de respuesta promedio:** Observa los valores registrados en el listener.
- **Errores:** Verifica si hubo respuestas no esperadas (códigos HTTP diferentes a 2xx).
- **Capacidad máxima:** Incrementa el número de Threads (usuarios) y observa el punto en el que el tiempo de respuesta aumenta considerablemente o aparecen errores.

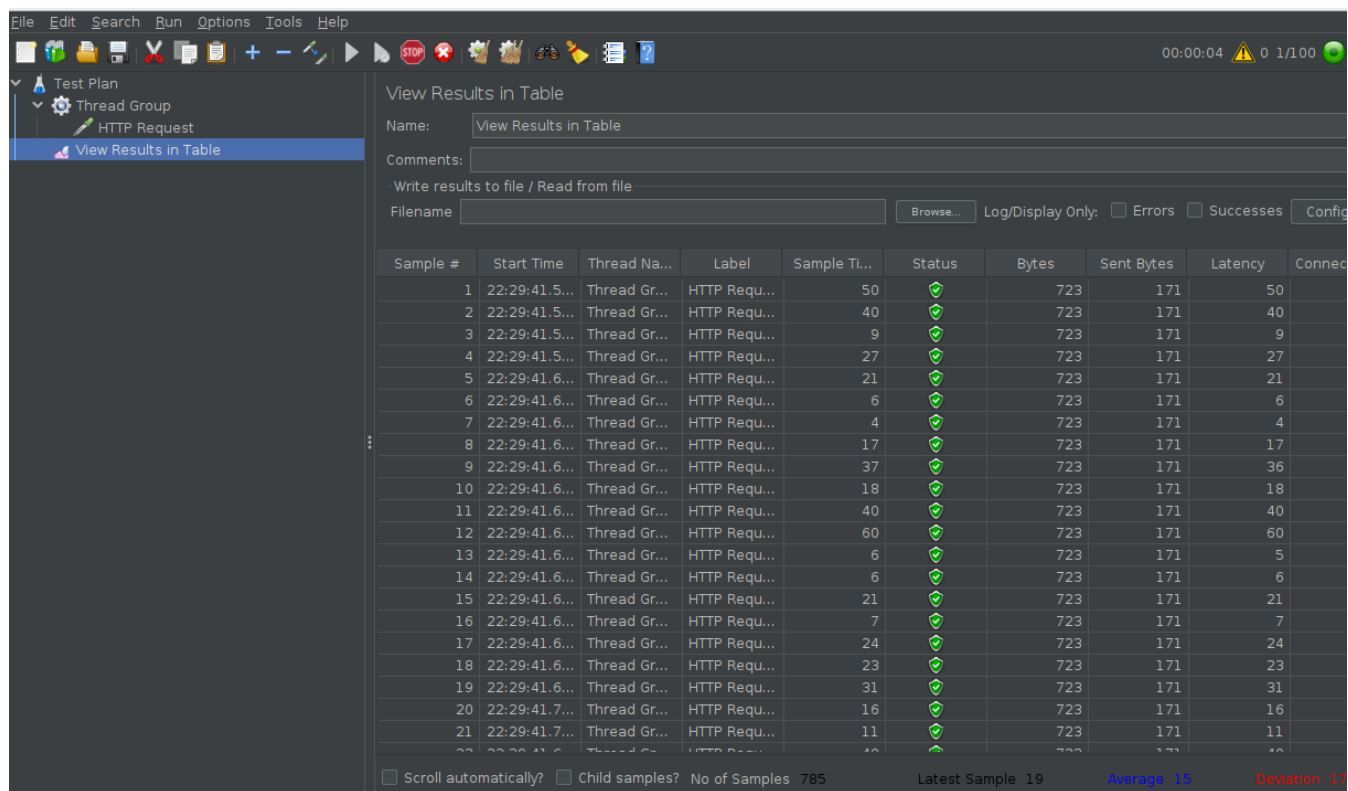
Pulsamos el botón verde para iniciar la prueba, lo cual nos mostrará lo siguiente:

Sample #	Start Time	Thread Na...	Label	Sample Ti...	Status	Bytes	Sent Bytes	Latency	Connec
1	22:26:24.7...	Thread Gr...	HTTP Requ...	66	✓	723	171	60	
2	22:26:24.7...	Thread Gr...	HTTP Requ...	4	✓	723	171	4	
3	22:26:24.7...	Thread Gr...	HTTP Requ...	5	✓	723	171	5	
4	22:26:24.8...	Thread Gr...	HTTP Requ...	5	✓	723	171	5	
5	22:26:24.8...	Thread Gr...	HTTP Requ...	5	✓	723	171	5	
6	22:26:24.8...	Thread Gr...	HTTP Requ...	5	✓	723	171	5	
7	22:26:24.8...	Thread Gr...	HTTP Requ...	4	✓	723	171	4	
8	22:26:24.8...	Thread Gr...	HTTP Requ...	4	✓	723	171	4	
9	22:26:24.8...	Thread Gr...	HTTP Requ...	4	✓	723	171	4	
10	22:26:24.8...	Thread Gr...	HTTP Requ...	3	✓	723	171	3	
11	22:26:24.9...	Thread Gr...	HTTP Requ...	7	✓	723	171	7	
12	22:26:24.9...	Thread Gr...	HTTP Requ...	10	✓	723	171	10	
13	22:26:25.0...	Thread Gr...	HTTP Requ...	17	✓	723	171	17	
14	22:26:25.0...	Thread Gr...	HTTP Requ...	10	✓	723	171	10	
15	22:26:25.0...	Thread Gr...	HTTP Requ...	8	✓	723	171	8	
16	22:26:25.0...	Thread Gr...	HTTP Requ...	4	✓	723	171	4	
17	22:26:25.0...	Thread Gr...	HTTP Requ...	3	✓	723	171	3	
18	22:26:25.0...	Thread Gr...	HTTP Requ...	4	✓	723	171	3	
19	22:26:25.0...	Thread Gr...	HTTP Requ...	4	✓	723	171	4	
20	22:26:25.0...	Thread Gr...	HTTP Requ...	3	✓	723	171	3	
21	22:26:25.5...	Thread Gr...	HTTP Requ...	9	✓	723	171	9	

☐ Scroll automatically?
 ☐ Child samples?
 No of Samples: 100
 Latest Sample: 3
 Average: 8
 Deviation: 9

Tal y como se puede ver en la captura, el tiempo de respuesta para la prueba con 10 usuarios fue muy bajo (de 8 ms), y no hubo ningún error.

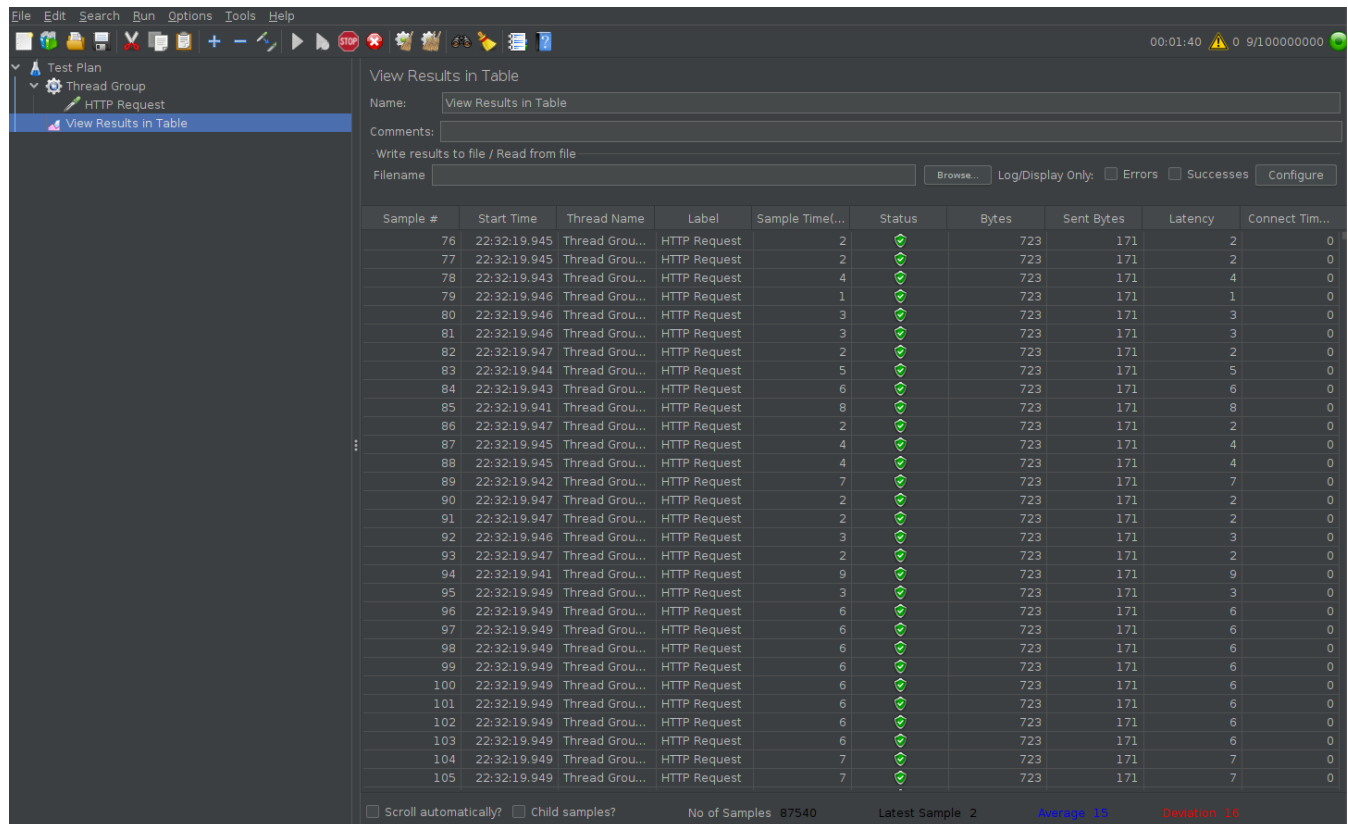
Repetimos las pruebas, pero esta vez con 100 usuarios:



Sample #	Start Time	Thread Name	Label	Sample Time	Status	Bytes	Sent Bytes	Latency	Connect Time
1	22:29:41.5...	Thread Gr...	HTTP Requ...	50	✓	723	171	50	
2	22:29:41.5...	Thread Gr...	HTTP Requ...	40	✓	723	171	40	
3	22:29:41.5...	Thread Gr...	HTTP Requ...	9	✓	723	171	9	
4	22:29:41.5...	Thread Gr...	HTTP Requ...	27	✓	723	171	27	
5	22:29:41.6...	Thread Gr...	HTTP Requ...	21	✓	723	171	21	
6	22:29:41.6...	Thread Gr...	HTTP Requ...	6	✓	723	171	6	
7	22:29:41.6...	Thread Gr...	HTTP Requ...	4	✓	723	171	4	
8	22:29:41.6...	Thread Gr...	HTTP Requ...	17	✓	723	171	17	
9	22:29:41.6...	Thread Gr...	HTTP Requ...	37	✓	723	171	36	
10	22:29:41.6...	Thread Gr...	HTTP Requ...	18	✓	723	171	18	
11	22:29:41.6...	Thread Gr...	HTTP Requ...	40	✓	723	171	40	
12	22:29:41.6...	Thread Gr...	HTTP Requ...	60	✓	723	171	60	
13	22:29:41.6...	Thread Gr...	HTTP Requ...	6	✓	723	171	5	
14	22:29:41.6...	Thread Gr...	HTTP Requ...	6	✓	723	171	6	
15	22:29:41.6...	Thread Gr...	HTTP Requ...	21	✓	723	171	21	
16	22:29:41.6...	Thread Gr...	HTTP Requ...	7	✓	723	171	7	
17	22:29:41.6...	Thread Gr...	HTTP Requ...	24	✓	723	171	24	
18	22:29:41.6...	Thread Gr...	HTTP Requ...	23	✓	723	171	23	
19	22:29:41.6...	Thread Gr...	HTTP Requ...	31	✓	723	171	31	
20	22:29:41.7...	Thread Gr...	HTTP Requ...	16	✓	723	171	16	
21	22:29:41.7...	Thread Gr...	HTTP Requ...	11	✓	723	171	11	

Esta vez, el tiempo de respuesta aumentó ligeramente, a 15 ms, y no hay errores aún.

Por último, he hecho una prueba con una cantidad mucho mayor de usuarios, con 100000000:



Sample #	Start Time	Thread Name	Label	Sample Time	Status	Bytes	Sent Bytes	Latency	Connect Time
76	22:32:19.945	Thread Grou...	HTTP Request	2	✓	723	171	2	0
77	22:32:19.945	Thread Grou...	HTTP Request	2	✓	723	171	2	0
78	22:32:19.943	Thread Grou...	HTTP Request	4	✓	723	171	4	0
79	22:32:19.946	Thread Grou...	HTTP Request	1	✓	723	171	1	0
80	22:32:19.946	Thread Grou...	HTTP Request	3	✓	723	171	3	0
81	22:32:19.946	Thread Grou...	HTTP Request	3	✓	723	171	3	0
82	22:32:19.947	Thread Grou...	HTTP Request	2	✓	723	171	2	0
83	22:32:19.944	Thread Grou...	HTTP Request	5	✓	723	171	5	0
84	22:32:19.943	Thread Grou...	HTTP Request	6	✓	723	171	6	0
85	22:32:19.941	Thread Grou...	HTTP Request	8	✓	723	171	8	0
86	22:32:19.947	Thread Grou...	HTTP Request	2	✓	723	171	2	0
87	22:32:19.945	Thread Grou...	HTTP Request	4	✓	723	171	4	0
88	22:32:19.945	Thread Grou...	HTTP Request	4	✓	723	171	4	0
89	22:32:19.942	Thread Grou...	HTTP Request	7	✓	723	171	7	0
90	22:32:19.947	Thread Grou...	HTTP Request	2	✓	723	171	2	0
91	22:32:19.947	Thread Grou...	HTTP Request	2	✓	723	171	2	0
92	22:32:19.946	Thread Grou...	HTTP Request	3	✓	723	171	3	0
93	22:32:19.947	Thread Grou...	HTTP Request	2	✓	723	171	2	0
94	22:32:19.941	Thread Grou...	HTTP Request	9	✓	723	171	9	0
95	22:32:19.949	Thread Grou...	HTTP Request	3	✓	723	171	3	0
96	22:32:19.949	Thread Grou...	HTTP Request	6	✓	723	171	6	0
97	22:32:19.949	Thread Grou...	HTTP Request	6	✓	723	171	6	0
98	22:32:19.949	Thread Grou...	HTTP Request	6	✓	723	171	6	0
99	22:32:19.949	Thread Grou...	HTTP Request	6	✓	723	171	6	0
100	22:32:19.949	Thread Grou...	HTTP Request	6	✓	723	171	6	0
101	22:32:19.949	Thread Grou...	HTTP Request	6	✓	723	171	6	0
102	22:32:19.949	Thread Grou...	HTTP Request	6	✓	723	171	6	0
103	22:32:19.949	Thread Grou...	HTTP Request	6	✓	723	171	6	0
104	22:32:19.949	Thread Grou...	HTTP Request	7	✓	723	171	7	0
105	22:32:19.949	Thread Grou...	HTTP Request	7	✓	723	171	7	0

El tiempo de respuesta sigue sin aumentar incluso con un número tan exagerado de usuarios (se ha mantenido en 15 ms), y tampoco ha habido errores.

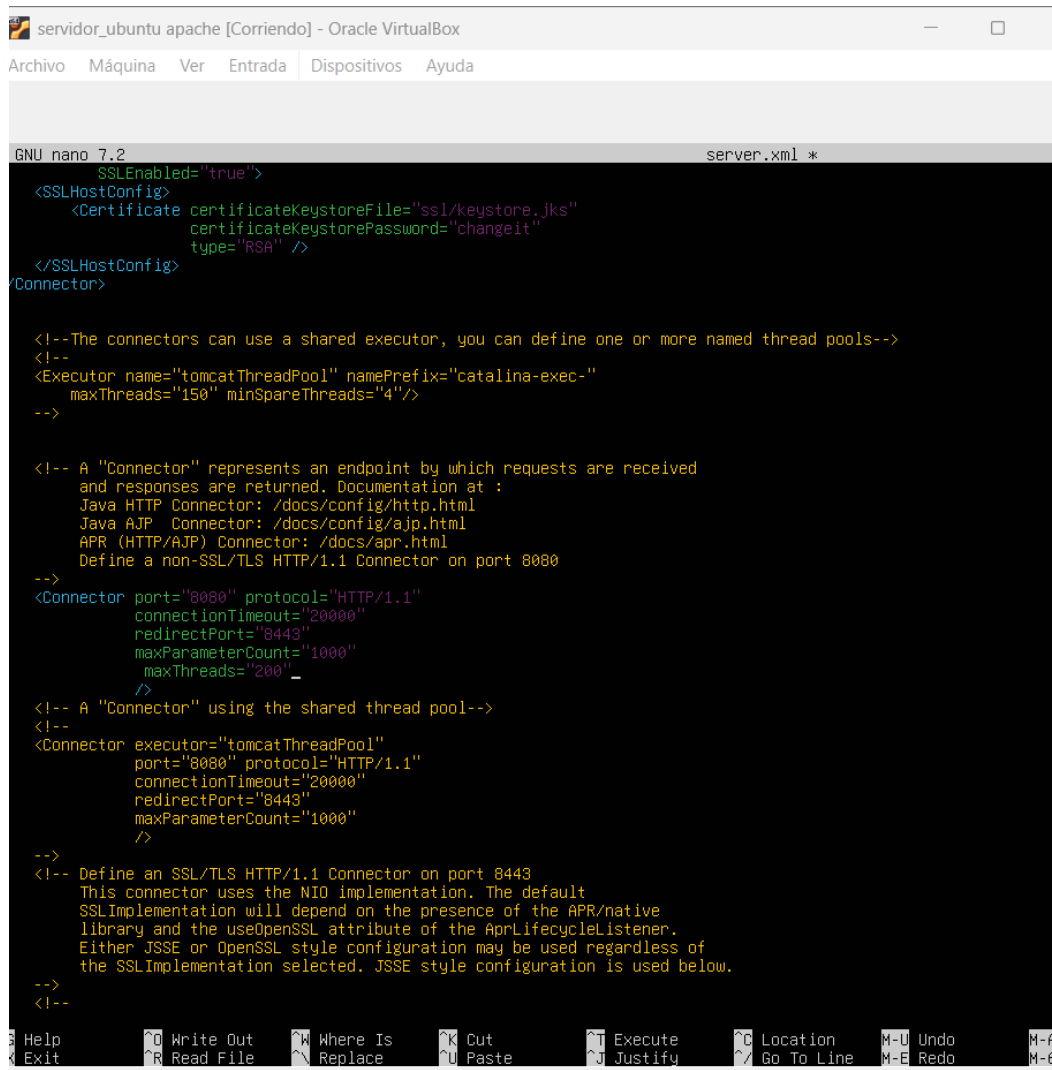
6. Optimizar y repetir pruebas

- Ajusta configuraciones en Tomcat si es necesario:
 - Modifica el archivo server.xml para aumentar el tamaño del pool de conexiones:

```
<Connector port="8080" protocol="HTTP/1.1"
    connectionTimeout="20000"
    maxThreads="200" />
```

- Prueba con diferentes escenarios de carga para identificar el límite de rendimiento.

Modificamos el archivo server.xml, para agregarle, dentro del Connector port, el atributo maxThreads="200", y el connectionTimeout="20000":



```
servidor_ubuntu apache [Corriendo] - Oracle VirtualBox
Archivo  Máquina  Ver  Entrada  Dispositivos  Ayuda

GNU nano 7.2 server.xml *
    SSLEnabled="true">
<SSLHostConfig>
  <Certificate certificateKeystoreFile="ssl/keystore.jks"
    certificateKeystorePassword="changeit"
    type="RSA" />
</SSLHostConfig>
</Connector>

<!--The connectors can use a shared executor, you can define one or more named thread pools-->
<!--
<Executor name="tomcatThreadPool" namePrefix="catalina-exec-"
  maxThreads="150" minSpareThreads="4"/>
-->

<!-- A "Connector" represents an endpoint by which requests are received
and responses are returned. Documentation at :
Java HTTP  Connector: /docs/config/http.html
Java AJP   Connector: /docs/config/ajp.html
APR (HTTP/AJP) Connector: /docs/apr.html
Define a non-SSL/TLS HTTP/1.1 Connector on port 8080
-->
<Connector port="8080" protocol="HTTP/1.1"
  connectionTimeout="20000"
  redirectPort="8443"
  maxParameterCount="1000"
  maxThreads="200"
/>
<!-- A "Connector" using the shared thread pool-->
<!--
<Connector executor="tomcatThreadPool"
  port="8080" protocol="HTTP/1.1"
  connectionTimeout="20000"
  redirectPort="8443"
  maxParameterCount="1000"
/>
-->
<!-- Define an SSL/TLS HTTP/1.1 Connector on port 8443
This connector uses the NIO implementation. The default
SSLImplementation will depend on the presence of the APR/native
library and the useOpenSSL attribute of the AprLifecycleListener.
Either JSSE or OpenSSL style configuration may be used regardless of
the SSLImplementation selected. JSSE style configuration is used below.
-->
<!--
Help  Write Out  Where Is  Cut  Execute  Location  M-U Undo  M-A
Exit  Read File  Replace  ^U Paste  ^J Justify  ^_ Go To Line  M-E Redo  M-6
```

Una vez ejecutada la optimización, repetimos las pruebas:

Sample #	Start Time	Thread Name	Label	Sample Time(...)	Status	Bytes	Sent Bytes	Latency	Connect Tim...
1	22:39:21.154	Thread Grou...	HTTP Request	39	Success	723	171	39	31
2	22:39:21.194	Thread Grou...	HTTP Request	1	Success	723	171	1	0
3	22:39:21.196	Thread Grou...	HTTP Request	9	Success	723	171	9	0
4	22:39:21.205	Thread Grou...	HTTP Request	2	Success	723	171	2	0
5	22:39:21.207	Thread Grou...	HTTP Request	3	Success	723	171	3	0
6	22:39:21.210	Thread Grou...	HTTP Request	5	Success	723	171	5	0
7	22:39:21.215	Thread Grou...	HTTP Request	3	Success	723	171	3	0
8	22:39:21.218	Thread Grou...	HTTP Request	2	Success	723	171	2	0
9	22:39:21.220	Thread Grou...	HTTP Request	1	Success	723	171	1	0
10	22:39:21.221	Thread Grou...	HTTP Request	1	Success	723	171	1	0
11	22:39:21.635	Thread Grou...	HTTP Request	8	Success	723	171	8	1
12	22:39:21.643	Thread Grou...	HTTP Request	4	Success	723	171	4	0
13	22:39:21.647	Thread Grou...	HTTP Request	12	Success	723	171	12	0
14	22:39:21.659	Thread Grou...	HTTP Request	12	Success	723	171	12	0
15	22:39:21.671	Thread Grou...	HTTP Request	2	Success	723	171	2	0
16	22:39:21.673	Thread Grou...	HTTP Request	2	Success	723	171	2	0
17	22:39:21.675	Thread Grou...	HTTP Request	1	Success	723	171	1	0
18	22:39:21.676	Thread Grou...	HTTP Request	2	Success	723	171	2	0
19	22:39:21.678	Thread Grou...	HTTP Request	2	Success	723	171	2	0
20	22:39:21.680	Thread Grou...	HTTP Request	1	Success	723	171	1	0
21	22:39:22.190	Thread Grou...	HTTP Request	10	Success	723	171	10	7
22	22:39:22.200	Thread Grou...	HTTP Request	3	Success	723	171	3	0
23	22:39:22.203	Thread Grou...	HTTP Request	3	Success	723	171	3	0
24	22:39:22.206	Thread Grou...	HTTP Request	5	Success	723	171	5	0
25	22:39:22.211	Thread Grou...	HTTP Request	34	Success	723	171	34	0
26	22:39:22.245	Thread Grou...	HTTP Request	5	Success	723	171	5	0
27	22:39:22.250	Thread Grou...	HTTP Request	3	Success	723	171	3	0
28	22:39:22.253	Thread Grou...	HTTP Request	1	Success	723	171	1	0
29	22:39:22.254	Thread Grou...	HTTP Request	2	Success	723	171	2	0
30	22:39:22.256	Thread Grou...	HTTP Request	13	Success	723	171	13	0

☐ Scroll automatically? ☐ Child samples? No of Samples: 80 Latest Sample: 2 Average: 5 Deviation: 4

En la prueba con 5 usuarios, el tiempo de respuesta disminuye hasta 5 ms.

En la prueba con 100 usuarios, también disminuye, quedando en 10 ms.

Sample #	Start Time	Thread Name	Label	Sample Time(...)	Status	Bytes	Sent Bytes	Latency	Connect Tim...
1	22:40:04.596	Thread Grou...	HTTP Request	44	Success	723	171	44	18
2	22:40:04.585	Thread Grou...	HTTP Request	37	Success	723	171	37	29
3	22:40:04.662	Thread Grou...	HTTP Request	3	Success	723	171	3	0
4	22:40:04.655	Thread Grou...	HTTP Request	11	Success	723	171	11	0
5	22:40:04.665	Thread Grou...	HTTP Request	2	Success	723	171	2	0
6	22:40:04.666	Thread Grou...	HTTP Request	23	Success	723	171	23	0
7	22:40:04.667	Thread Grou...	HTTP Request	25	Success	723	171	25	0
8	22:40:04.652	Thread Grou...	HTTP Request	52	Success	723	171	52	29
9	22:40:04.692	Thread Grou...	HTTP Request	13	Success	723	171	13	0
10	22:40:04.689	Thread Grou...	HTTP Request	17	Success	723	171	17	0
11	22:40:04.705	Thread Grou...	HTTP Request	3	Success	723	171	3	0
12	22:40:04.704	Thread Grou...	HTTP Request	4	Success	723	171	4	0
13	22:40:04.706	Thread Grou...	HTTP Request	3	Success	723	171	3	0
14	22:40:04.708	Thread Grou...	HTTP Request	2	Success	723	171	2	0
15	22:40:04.708	Thread Grou...	HTTP Request	2	Success	723	171	2	0
16	22:40:04.709	Thread Grou...	HTTP Request	3	Success	723	171	3	0
17	22:40:04.710	Thread Grou...	HTTP Request	3	Success	723	171	3	0
18	22:40:04.709	Thread Grou...	HTTP Request	4	Success	723	171	4	1
19	22:40:04.710	Thread Grou...	HTTP Request	4	Success	723	171	4	0
20	22:40:04.712	Thread Grou...	HTTP Request	2	Success	723	171	2	0
21	22:40:04.713	Thread Grou...	HTTP Request	4	Success	723	171	4	0
22	22:40:04.714	Thread Grou...	HTTP Request	3	Success	723	171	3	0
23	22:40:04.713	Thread Grou...	HTTP Request	5	Success	723	171	5	0
24	22:40:04.714	Thread Grou...	HTTP Request	4	Success	723	171	4	0
25	22:40:04.717	Thread Grou...	HTTP Request	9	Success	723	171	9	0
26	22:40:04.718	Thread Grou...	HTTP Request	19	Success	723	171	19	0
27	22:40:04.726	Thread Grou...	HTTP Request	11	Success	723	171	11	0
28	22:40:04.718	Thread Grou...	HTTP Request	26	Success	723	171	26	0
29	22:40:04.737	Thread Grou...	HTTP Request	21	Success	723	171	21	0
30	22:40:04.744	Thread Grou...	HTTP Request	16	Success	723	171	16	0

☐ Scroll automatically? ☐ Child samples? No of Samples: 1000 Latest Sample: 1 Average: 10 Deviation: 10

En la prueba con 100000000 usuarios, el tiempo de respuesta no mejora, debido a los límites físicos del hardware y la capacidad del sistema.

View Results in Table

Name: View Results in Table

Comments:

Write results to file / Read from file

Filename: Log/Display Only: ☐ Errors ☐ Successes

Sample #	Start Time	Thread Name	Label	Sample Time...	Status	Bytes	Sent Bytes	Latency	Connect Tim...
1	22:41:36.419	Thread Grou...	HTTP Request	26	✓	723	171	26	11
2	22:41:36.418	Thread Grou...	HTTP Request	28	✓	723	171	28	6
3	22:41:36.419	Thread Grou...	HTTP Request	28	✓	723	171	28	5
4	22:41:36.414	Thread Grou...	HTTP Request	39	✓	723	171	39	10
5	22:41:36.428	Thread Grou...	HTTP Request	27	✓	723	171	27	1
6	22:41:36.425	Thread Grou...	HTTP Request	30	✓	723	171	30	3
7	22:41:36.414	Thread Grou...	HTTP Request	41	✓	723	171	41	10
8	22:41:36.419	Thread Grou...	HTTP Request	37	✓	723	171	37	5
9	22:41:36.455	Thread Grou...	HTTP Request	2	✓	723	171	2	0
10	22:41:36.449	Thread Grou...	HTTP Request	9	✓	723	171	9	0
11	22:41:36.446	Thread Grou...	HTTP Request	13	✓	723	171	13	0
12	22:41:36.446	Thread Grou...	HTTP Request	13	✓	723	171	13	0
13	22:41:36.455	Thread Grou...	HTTP Request	5	✓	723	171	5	0
14	22:41:36.453	Thread Grou...	HTTP Request	7	✓	723	171	7	0
15	22:41:36.455	Thread Grou...	HTTP Request	6	✓	723	171	6	0
16	22:41:36.457	Thread Grou...	HTTP Request	4	✓	723	171	4	0
17	22:41:36.430	Thread Grou...	HTTP Request	31	✓	723	171	31	27
18	22:41:36.458	Thread Grou...	HTTP Request	4	✓	723	171	4	0
19	22:41:36.456	Thread Grou...	HTTP Request	6	✓	723	171	6	2
20	22:41:36.459	Thread Grou...	HTTP Request	3	✓	723	171	3	0
21	22:41:36.459	Thread Grou...	HTTP Request	3	✓	723	171	3	0
22	22:41:36.460	Thread Grou...	HTTP Request	6	✓	723	171	6	0
23	22:41:36.456	Thread Grou...	HTTP Request	10	✓	723	171	10	1
24	22:41:36.461	Thread Grou...	HTTP Request	6	✓	723	171	6	0
25	22:41:36.467	Thread Grou...	HTTP Request	2	✓	723	171	2	0
26	22:41:36.462	Thread Grou...	HTTP Request	7	✓	723	171	7	0
27	22:41:36.462	Thread Grou...	HTTP Request	7	✓	723	171	7	0
28	22:41:36.466	Thread Grou...	HTTP Request	4	✓	723	171	4	0
29	22:41:36.460	Thread Grou...	HTTP Request	10	✓	723	171	10	0
30	22:41:36.462	Thread Grou...	HTTP Request	8	✓	723	171	8	0

☐ Scroll automatically? ☐ Child samples? No of Samples 41544 Latest Sample 39 Average 16 Deviation 16

DOCUMENTACIÓN

Deberás documentar los procedimientos indicando:

- Los pasos realizados (comandos, modificaciones a ficheros de configuración y rutas de los mismos, etc.).
- capturas de pantalla que demuestren que se han logrado los objetivos planteados.
- Presenta gráficos de tiempo de respuesta y errores para diferentes niveles de carga.
- Describe los límites de la aplicación y posibles mejoras.