

INGENIERÍA DE SERVIDORES (2022-2023)

GRADO EN INGENIERÍA INFORMÁTICA

UNIVERSIDAD DE GRANADA

Memoria Práctica 4

Diego de Luis Ballesteros

23 de diciembre de 2022

Índice

1. Instalación de Phoronix en Ubuntu y Rocky y ejecución de dos benchmarks en cada máquina para comentar las diferencias.

- 1.1 Instalación de Phoronix en Ubuntu
- 1.2 Ejecución de benchmarks en Ubuntu
- 1.3 Instalación de Phoronix en Rocky
- 1.4 Ejecución de benchmarks en Rocky
- 1.5 Diferencias
- 1.6 Referencias

2. Test de JMeter sobre dos contenedores

- 2.1 Enunciado del ejercicio
- 2.2 Instalación de Docker, Docker-compose y la aplicación de git.
- 2.3 Instalación de JMeter en nuestro ordenador (Windows 10)
- 2.4 Creación del test de JMeter
- 2.5 Prueba del Test de JMeter
- 2.6 Referencias

1. Instalación de Phoronix en Ubuntu y Rocky y ejecución de dos benchmarks en cada máquina para comentar las diferencias

1.1 Instalación de Phoronix en Ubuntu

Phoronix test suite es una herramienta automatizada de prueba y evaluación de código abierto. Puede usarse para comparar el rendimiento de diferentes máquinas virtuales. En este caso la usaremos para comparar el rendimiento entre una de Ubuntu y otra de Rocky.

Phoronix está disponible en los repositorios de Ubuntu, pero probablemente son viejos así que vamos a añadir con wget (figura 1.2) contenido para instalar la última versión estable de Phoronix, en estos momentos, la 10.8.4 (figura 1.1). Una vez hecha la descarga podremos realizar la instalación con ayuda de el comando dpkg (figura 1.2).

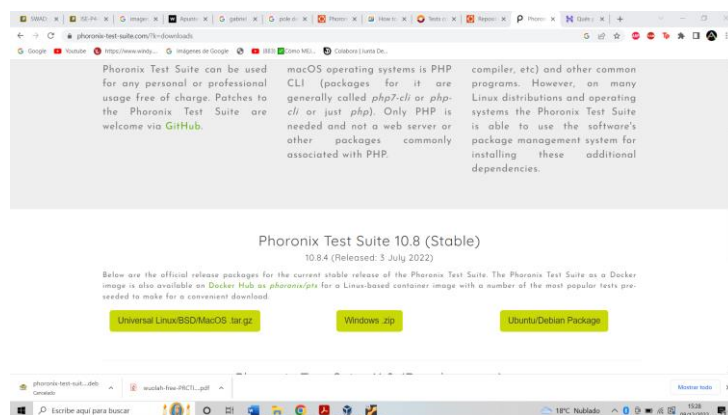


Figura 1.1

```
diegolb@ubuntu:~$ wget http://phoronix-test-suite.com/releases/repo/pts.debian/files/phoronix-test-suite_10.8.4_all.deb
--2022-12-08 14:30:03-- http://phoronix-test-suite.com/releases/repo/pts.debian/files/phoronix-test-suite_10.8.4_all.deb
Resolving phoronix-test-suite.com (phoronix-test-suite.com)... 192.211.48.82
Connecting to phoronix-test-suite.com (phoronix-test-suite.com)[192.211.48.82]:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 2352160 (2.2M) [application/x-debian-package]
Saving to: 'phoronix-test-suite_10.8.4_all.deb'

phoronix-test-suite_10.8 100%[=====] 2.24M 435KB/s in 5.4s
2022-12-08 14:30:09 (423 KB/s) - 'phoronix-test-suite_10.8.4_all.deb' saved [2352160/2352160]
```

```
diegolb@ubuntu:~$ sudo dpkg -i phoronix-test-suite_10.8.4_all.deb
```



Figura 1.2

1.2 Ejecución de benchmarks en ubuntu

Ahora vamos a listar los diferentes benchmarks (que se pueden ejecutar con Phoronix) con el comando **phoronix-test-suite list-tests | less** (figura 1.3). En concreto yo voy a usar dos: el pts/sudokut (solucionador de problemas de rompecabezas escrito en Tcl) y pts/pybench. Este último informa tiempos de prueba promedio para diferentes funciones, como BuiltinFunctionCalls y NestedForLoops. El resultado total proporciona una estimación aproximada del rendimiento promedio de Python en un sistema determinado. Este perfil de prueba ejecuta PyBench cada vez durante 20 rondas. Para ejecutar sudokut podemos hacer **phoronix-test-suite benchmark sudokut**. Para el pybench con el mismo comando. Resultados: figura 1.4 (sudokut), figura 1.5 (pybench). Puedo ver los resultados del sudokut en <https://openbenchmarking.org/result/2212089-NE-SUDOKUTDI63> y los del pybench en <https://openbenchmarking.org/result/2212080-NE-DIEGO521515>

Nota: los test son uno para procesador y otro para el sistema. Iba a usar pts/sysbench (una herramienta de referencia multiproceso programable basada en LuaJIT), pero me da error al usarlo porque faltan muchos paquetes.

pts/sudokut	ESG [1msSudokuESG [0m	Processor
pts/sunflow	ESG [1msSunflow Rendering SystemESG [0m	System
pts/svt-av1	ESG [1msSVT-AV1ESG [0m	Processor
pts/svt-hevc	ESG [1msSVT-HEVCESG [0m	Processor
pts/svt-vp9	ESG [1msSVT-VP9ESG [0m	Processor
pts/swet	ESG [1msSwetESG [0m	Processor
pts/synthmark	ESG [1msGoogle SynthMarkESG [0m	Processor
pts/sysbench	ESG [1msSysbenchESG [0m	System
pts/system-decompress-bzip2	ESG [1msSystem BZIP2 DecompressionESG [0m	Processor

Figura 1.3

```
SudoKut 0.4:
pts/sudokut-1.0.1
Test 1 of 1
Estimated Trial Run Count:    3
Estimated Time To Completion: 2 Minutes [15:20 UTC]
Started Run 1 @ 15:18:38
Started Run 2 @ 15:19:08
Started Run 3 @ 15:19:37

Total Time:
25.725
25.47
25.527

Average: 25.574 Seconds
Deviation: 0.52%

Comparison of 6,332 OpenBenchmarking.org samples since 26 February 2011; median result: 29.64 Se
conds. Box plot of samples:
[ * * * |-----x-----*#####*~*~* ]
                                     This Result (62nd Percentile): 25.574 ^
      ^ ARMv7 rev 0: 236           Intel Pentium 4 2.80GHz: 89 ^ Intel Core i9-12900H: 6.946 ^
      ^ ARMv7 Cortex-A7: 256       Intel Core i7-6700K: 10.2 ^
                                   Intel Core i7-4600U: 15.6 ^
                                   Intel Core i7 720Q: 30.16 ^

Do you want to view the text results of the testing (Y/n):
```



Figura 1.4`

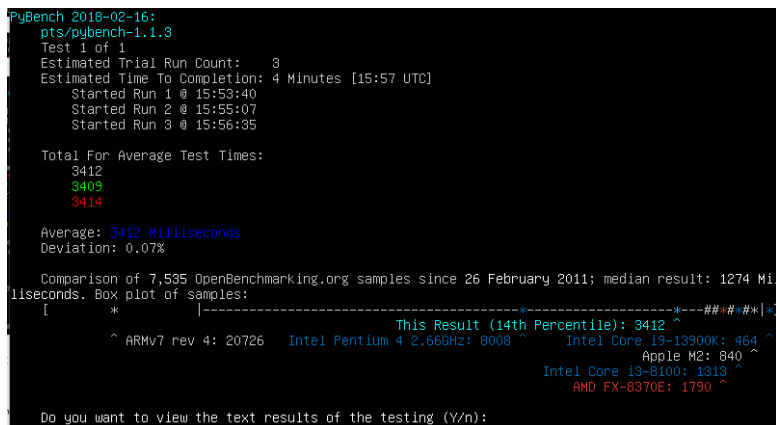


Figura 1.5

Primero de todo tenemos que instalar varias dependencias con **sudo dnf install wget php-cli php-xml bzip2**. Luego pasamos a coger el paquete comprimido de Phoronix usando wget (figura 1.6), lo descomprimiremos (hemos de instalar tar porque si no, no se puede) y lo instalaremos ejecutando el script de instalación (figura 1.7).

Figura 1.6

Figura 1.7

Procedemos a ejecutar los mismos benchmarks que en Ubuntu. En concreto pybench (figura 1.8) y sudokut (figura 1.9).

Los resultados de pybench se pueden ver en <https://openbenchmarking.org/result/2212084-NE-DIEGOL74861> y los resultados de sudokut se pueden ver en <https://openbenchmarking.org/result/2212088-NE-SUDOKUTDI31>

```
PyBench 2018-02-16:
pts/pybench-1.1.3
Test 1 of 1
Estimated Trial Run Count: 3
Estimated Time To Completion: 4 Minutes [03:31 CET]
Started Run 1 @ 03:28:23
Started Run 2 @ 03:30:13
Started Run 3 @ 03:32:03

Total For Average Test Times:
4321
4320
4314

Average: 4318 Milliseconds
Deviation: 0.03%

Comparison of 7,535 OpenBenchmarking.org samples since 26 February 2011; median result: 1274 Milliseconds. Box plot of samples:
-----
^ ARMv7 rev 4: 28726 Intel Pentium 4 2.66GHz: 8880 Intel Core i9-13980K: 464 Apple M2: 848 Intel Core i3-5180: 1313 AMD FX-8370E: 1798

Do you want to view the text results of the testing (Y/n):
```

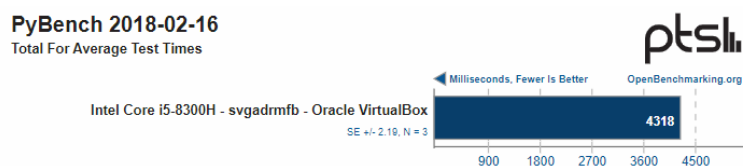


Figura 1.8

```
Sudoku 0.4:
pts/sudokut-1.0.1
Test 1 of 1
Estimated Trial Run Count: 3
Estimated Time To Completion: 2 Minutes [03:38 CET]
Started Run 1 @ 03:36:47
Started Run 2 @ 03:37:24
Started Run 3 @ 03:38:02

Total Time:
33.812
33.282
33.054

Average: 33.383 Seconds
Deviation: 1.16%

Comparison of 6,332 OpenBenchmarking.org samples since 26 February 2011; median result: 29.64 Seconds. Box plot of samples:
-----
^ ARMv7 Cortex-A7: 256 Intel Pentium 4 2.80GHz: 89 Intel Core i9-12900H: 6.946 Intel Core i7-6700K: 18.2 Intel Core i7-4580U: 15.6 Intel Core i7-3517U: 21.45

Do you want to view the text results of the testing (Y/n):
```



Figura 1.9

1.5 Diferencias

Podemos ver en el caso de sudokut que Ubuntu está mejor optimizado dado que tarda 25 segundos y rocky tarda 33.28 al menos para este benchmark. Por otro lado, para pybench, también tenemos que Ubuntu tarda menos, en nuestro caso 3412 segundos y para rocky 4318. Por tanto, a lo mejor no es información suficiente, pero en Ubuntu el procesador ejecuta con más velocidad los programas que en rocky de momento.

1.6 Referencias

Aquí tenemos las referencias para la instalación en Ubuntu y Rocky de Phoronix

- 1- <https://www.phoronix-test-suite.com/?k=downloads>
- 2- <https://arstech.net/phoronix-test-suite/>
- 3- <https://ubunlog.com/phoronix-test-suite-una-herramienta-para-benchmark-multiplataforma/>

Aquí está la referencia de la página donde podemos ver qué hace cada prueba.

- 4- <https://openbenchmarking.org/tests>

2. Test de JMeter sobre dos contenedores

2.1 Enunciado del ejercicio

tras probar un test básico para una web [5], utilizaremos Jmeter para hacer un test sobre una aplicación que ejecuta sobre dos contenedores (uno para la BD y otro para la aplicación en sí). El código está disponible en <https://github.com/davidPalomar-ugr/iseP4JMeter> donde se dan detalles sobre cómo ejecutar la aplicación en una de nuestras máquinas virtuales. El test de Jmeter debe incluir los siguientes elementos:

1. El test debe tener parametrizados el Host y el Puerto en el Test Plan (puede hacer referencia usando \$param)
2. Debe hacer dos grupos de hebras distintos para simular el acceso de los alumnos y los administradores. Las credenciales de alumno y administrador se cogen de los archivos: alumnos.csv y administrador.csv respectivamente.
3. Añadimos esperas aleatorias a cada grupo de hebras (Gaussian Random Timer) El login de alumno, su consulta de datos (recuperar datos alumno) y login del administrador son peticiones HTTP.
4. El muestreo para simular el acceso de los administradores lo debe coger el archivo apiAlumnos.log (usando un Acces Log Sampler) .
5. Use una expresión regular (Regular Expresión Extractor) para extraer el token JWT que hay que añadir a la cabecera de las peticiones (usando HTTP Header Manager).

2.2 Instalación de Docker, Docker-compose y la aplicación de git

Para la instalación de Docker y de Docker-compose seguiremos el guión de la práctica 4 tal y como viene.

Primero añadiremos la llave para validar el repositorio usando curl, añadiremos el repositorio, actualizaremos la lista de repositorios y buscaremos el repositorio de Docker y lo instalaremos (figura 2.1).

```
status: inactive
diegolb@ubuntu:~$ curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -
[sudo] password for diegolb:
OK

Get:33 http://es.archive.ubuntu.com/ubuntu focal-security/restricted amd64 Packages [1383 kB]
Get:34 http://es.archive.ubuntu.com/ubuntu focal-security/restricted Translation-en [195 kB]
Get:35 http://es.archive.ubuntu.com/ubuntu focal-security/restricted amd64 c-n-f Metadata [596 B]
Get:36 http://es.archive.ubuntu.com/ubuntu focal-security/universe amd64 Packages [777 kB]
Get:37 http://es.archive.ubuntu.com/ubuntu focal-security/universe Translation-en [150 kB]
Get:38 http://es.archive.ubuntu.com/ubuntu focal-security/universe amd64 c-n-f Metadata [16.9 kB]
Get:39 http://es.archive.ubuntu.com/ubuntu focal-security/multiverse amd64 Packages [22.2 kB]
Get:40 http://es.archive.ubuntu.com/ubuntu focal-security/multiverse Translation-en [5464 B]
Get:41 http://es.archive.ubuntu.com/ubuntu focal-security/multiverse amd64 c-n-f Metadata [516 B]
Fetched 266 MB in 6min 20s (700 kB/s)
Reading package lists... Done
diegolb@ubuntu:~$ sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu
$(lsb_release -cs) stable"

Reading package lists... Done
diegolb@ubuntu:~$ sudo apt update
Hit:1 http://es.archive.ubuntu.com/ubuntu focal InRelease
Hit:2 https://download.docker.com/linux/ubuntu focal InRelease
Hit:3 http://es.archive.ubuntu.com/ubuntu focal-updates InRelease
Get:4 http://repo.zabbix.com/zabbix/5.0/ubuntu focal InRelease [4958 B]
Hit:5 http://es.archive.ubuntu.com/ubuntu focal-backports InRelease
Hit:6 http://es.archive.ubuntu.com/ubuntu focal-security InRelease
Fetched 4958 B in 1s (5004 B/s)
Reading package lists... Done
Building dependency tree
Reading state information... Done

122 packages can be upgraded. Run 'apt list --upgradable' to see them.
diegolb@ubuntu:~$ sudo apt install docker-ce
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages were automatically installed and are no longer required:
  fakeroot libalgorithm-diff-perl libalgorithm-diff-xs-perl libalgorithm-diff-perl libatomic1
  libc-dev-bin libcc1-0 libcrypt-dev libdpkg-perl libfontconfig1 libgomp1 libitm1 liblsan0
  libquadmath0 libtsan0 libubsan1 manpages
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
  containerd.io docker-ce-cli docker-ce-rootless-extras docker-sc
Suggested packages:
```

```
diegolb@ubuntu:~$ sudo apt search docker-ce
```

```
diegolb@ubuntu:~$ sudo apt install docker-ce
```

Figura 2.1

Ahora comprobamos el estado del servicio (ver que está iniciado y activado...) (figura 2.2)


```
diegolb@ubuntu:~$ sudo systemctl enable --now docker.service
Synchronizing state of docker.service with SysV service script with /lib/systemd/syste
l.
Executing: /lib/systemd/systemd-sysv-install enable docker
diegolb@ubuntu:~$ systemctl status docker.service
● docker.service - Docker Application Container Engine
   Loaded: loaded (/lib/systemd/system/docker.service; enabled; vendor preset: enab
   Active: active (running) since Thu 2022-12-08 18:41:07 UTC; 4s ago
   TriggeredBy: ● docker.socket
     Docs: https://docs.docker.com
    Main PID: 37912 (dockerd)
       Tasks: 7
      Memory: 84.2M
     CGroup: /system.slice/docker.service
            └─37912 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd
Dec 08 18:41:07 ubuntu dockerd[37912]: time="2022-12-08T18:41:07.144274329Z" level=war
Dec 08 18:41:07 ubuntu dockerd[37912]: time="2022-12-08T18:41:07.144361135Z" level=war
Dec 08 18:41:07 ubuntu dockerd[37912]: time="2022-12-08T18:41:07.144443288Z" level=war
Dec 08 18:41:07 ubuntu dockerd[37912]: time="2022-12-08T18:41:07.144870555Z" level=info
Dec 08 18:41:07 ubuntu dockerd[37912]: time="2022-12-08T18:41:07.355643508Z" level=info
Dec 08 18:41:07 ubuntu dockerd[37912]: time="2022-12-08T18:41:07.540095472Z" level=info
Dec 08 18:41:07 ubuntu dockerd[37912]: time="2022-12-08T18:41:07.720080281Z" level=info
Dec 08 18:41:07 ubuntu dockerd[37912]: time="2022-12-08T18:41:07.720535100Z" level=info
Dec 08 18:41:07 ubuntu systemd[1]: Started Docker Application Container Engine.
Dec 08 18:41:07 ubuntu dockerd[37912]: time="2022-12-08T18:41:07.799215921Z" level=info
diegolb@ubuntu:~$
```

Figura 2.2

Ahora tenemos que añadir nuestro nombre de usuario al grupo docker con **sudo usermod -aG Docker diegolb**. Vamos a probar varios comandos para ver que todo está correcto (figura 2.3). En concreto sudo docker run hello-world (también docker info).

```
diegolb@ubuntu:~$ sudo docker run hello-world

Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
1. The Docker client contacted the Docker daemon.
2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
   (amd64)
3. The Docker daemon created a new container from that image which runs the
   executable that produces the output you are currently reading.
4. The Docker daemon streamed that output to the Docker client, which sent it
   to your terminal.

To try something more ambitious, you can run an Ubuntu container with:
$ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:
https://hub.docker.com/

For more examples and ideas, visit:
https://docs.docker.com/get-started/
```

Figura 2.3

Con esto hemos finalizado la instalación de docker, ahora vamos a seguir con docker-compose. Para ello vamos a instalar docker-compose y vamos a comprobar que funciona (figura 2.4).

```
Setting up python3-docopt (0.6.2-2ubuntu1) ...
Setting up python3-websocket (0.53.0-2ubuntu1) ...
update-alternatives: using /usr/bin/python3-wsdump to provide /usr/bin/
Setting up python3-dockerpty (0.4.1-2) ...
Setting up python3-docker (4.1.0-1) ...
Setting up docker-compose (1.25.0-1) ...
Processing triggers for man-db (2.9.1-1) ...
diegolb@ubuntu:~$ sudo apt install docker-compose_
```

```
diegolb@ubuntu:~$ docker-compose -v
docker-compose version 1.25.0, build unknown
diegolb@ubuntu:~$
```

```
rm          Remove stopped containers
run         Run a one-off command
scale       Set number of containers for a service
start       Start services
stop        Stop services
top         Display the running processes
unpause     Unpause services
up          Create and start containers
version     Show the Docker-Compose version information
diegolb@ubuntu:~$ docker-compose
```

Figura 2.4

Por último, instalaremos la aplicación de git para el test de JMeter mediante git clone.

```
diegolb@ubuntu:~$ git clone https://github.com/davidPalomar-ugr/iseP4JMeter.git
Cloning into 'iseP4JMeter'...
remote: Enumerating objects: 3816, done.
remote: Counting objects: 100% (42/42), done.
remote: Compressing objects: 100% (33/33), done.
remote: Total 3816 (delta 21), reused 14 (delta 7), pack-reused 3774
Receiving objects: 100% (3816/3816), 7.82 MiB | 855.00 KiB/s, done.
Resolving deltas: 100% (727/727), done.
```

Figura 2.5

Ahora hacemos un **sudo docker-compose up** (figura 2.7). Me faltaba espacio, por tanto, he decidido crear un nuevo disco (le añado el volumen físico) con su grupo de volúmenes y añadiéndole el volumen lógico. En este he puesto el /var para poder tener todo el espacio que necesito (lsblk en figura 2.6).

```
diegolb@ubuntu:~/iseP4JMeter$ lsblk
NAME        MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
loop0        7:0      0   63.2M 1 loop /snap/core20/1695
loop1        7:1      0   55.6M 1 loop /snap/core18/2632
loop2        7:2      0   91.8M 1 loop /snap/lxd/23991
loop3        7:3      0   63.2M 1 loop /snap/core20/1738
loop4        7:4      0   49.6M 1 loop /snap/snapd/17883
loop5        7:5      0   55.6M 1 loop /snap/core18/2620
loop6        7:6      0   91.9M 1 loop /snap/lxd/24061
sda          8:0      0    10G 0 disk 
├─sda1        8:1      0     1M 0 part 
├─sda2        8:2      0    400M 0 part 
├─└─md0        9:0      0    399M 0 raid1 /boot
│   └─sda3        8:3      0     9.6G 0 part 
│       └─md1        9:1      0     9.6G 0 raid1 
│           └─dm_crypt-0 253:1      0     9.6G 0 crypt 
│               └─vgraid1-swap 253:2      0      1G 0 lvm [SWAP]
│                   └─vgraid1-home 253:3      0    500M 0 lvm /home
│                       └─vgraid1-raiz 253:4      0     8.1G 0 lvm /
sdb          8:16     0    10G 0 disk 
├─sdb1        8:17     0     1M 0 part 
├─sdb2        8:18     0    400M 0 part 
├─└─md0        9:0      0    399M 0 raid1 /boot
│   └─sdb3        8:19     0     9.6G 0 part 
│       └─md1        9:1      0     9.6G 0 raid1 
│           └─dm_crypt-0 253:1      0     9.6G 0 crypt 
│               └─vgraid1-swap 253:2      0      1G 0 lvm [SWAP]
│                   └─vgraid1-home 253:3      0    500M 0 lvm /home
│                       └─vgraid1-raiz 253:4      0     8.1G 0 lvm /
sdc          8:32     0    10G 0 disk 
└─varD-var    253:0      0     8G 0 lvm /var
sr0         11:0     1     61M 0 rom
```

Figura 2.6

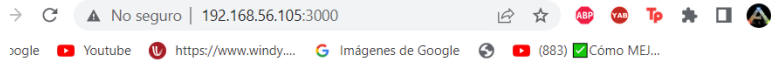
Se me había olvidado ejecutarlo en segundo plano, así que lo hago ahora con la opción **-d**.

```
mongod_1 | {"t":{"$date":"2022-12-08T19:38:30.232+00:00"},"s":"I", "c":"","id":20883,
, "ctx":"conn39", "msg":"Interrupted operation as its client disconnected", "attr":{"opId":1341}}
mongod_1 | {"t":{"$date":"2022-12-08T19:38:30.232+00:00"},"s":"I", "c":"NETWORK", "id":22944,
, "ctx":"conn38", "msg":"Connection ended", "attr":{"remote":"172.18.0.4:35170", "uid":"4572a842-68c
d-4c9c-abdb-48ca8fd3b61c", "connectionId":38, "connectionCount":6}}
mongod_1 | {"t":{"$date":"2022-12-08T19:38:30.234+00:00"},"s":"I", "c":"NETWORK", "id":22944,
, "ctx":"conn39", "msg":"Connection ended", "attr":{"remote":"172.18.0.4:35180", "uid":"e91ffd77-65d
e-471f-91a8-0c11812ca9e7", "connectionId":39, "connectionCount":5}}
mongod_1 | {"t":{"$date":"2022-12-08T19:38:30.235+00:00"},"s":"I", "c":"NETWORK", "id":22944,
, "ctx":"conn40", "msg":"Connection ended", "attr":{"remote":"172.18.0.4:35192", "uid":"28951125-bed
d-4f77-864b-49a936d19071", "connectionId":40, "connectionCount":4}}
mongod_1 | {"t":{"$date":"2022-12-08T19:38:30.304+00:00"},"s":"I", "c":"STORAGE", "id":22260,
, "ctx":"TimestampMonitor", "msg":"Removing drop-pending Idents with drop timestamps before timesta
mp", "attr":{"timestamp":{"timestamp":{"t":"0.1":"0}}}}
mongod_1 | {"t":{"$date":"2022-12-08T19:38:30.304+00:00"},"s":"I", "c":"STORAGE", "id":22237,
, "ctx":"TimestampMonitor", "msg":"Completing drop for ident", "attr":{"ident":"internal-20--1368007
912899941227", "dropTimestamp":{"timestamp":{"t":"0.1":"0}}}}
mongod_1 | {"t":{"$date":"2022-12-08T19:38:30.308+00:00"},"s":"I", "c":"STORAGE", "id":22237,
, "ctx":"TimestampMonitor", "msg":"Completing drop for ident", "attr":{"ident":"internal-19--1368007
912899941227", "dropTimestamp":{"timestamp":{"t":"0.1":"0}}}}
iseP4JMeter_mongodbin1_1 exited with code 0
^CGracefully stopping... (press Ctrl+C again to force)
Stopping iseP4JMeter_nodejs_1 ... done
Stopping iseP4JMeter_mongodbin1_1 ... done
diegolb@ubuntu:~/iseP4JMeter$ sudo docker-compose up -d_
```

Figura 2.7

Ahora habilitamos el puerto 3000 que es el que usa la aplicación. Una vez habilitado veremos que la página funciona.

```
Starting JMeter...
diegolb@ubuntu:~/iseP4JMeter$ sudo ufw allow 3000/tcp
Rules updated
Rules updated (v6)
```



ETSII Alumnos API

Descripción de la API Restful:

- POST /api/v1/auth/login
 - Parametros:
 - login:<emailUsuario>
 - password:<secreto>
 - Seguridad:
 - Acceso protegido con BasicAuth (etsiiApi:laApiDeLaETSIIaLache)
 - Retorna:
 - JWT Token
- GET /api/v1/alumnos/alumno/<email>
 - Seguridad:
 - Token JWT valido en cabecera estandar authorization: Bearer <token>
 - Alumnos solo pueden solicitar sus datos. Administradores pueden solicitar cualq
 - Retorna:
 - Objeto Json con perfil de alumno

Figura 2.8

Ahora también podemos ejecutar el pruebaEntorno.sh para ver si hemos lanzado bien la aplicación (figura 2.9).

```
diegolb@ubuntu:~/iseP4JMeter$ ./pruebaEntorno.sh
{"id":"63923d2c2120744f460f916d","nombre":"Mari","apellidos":"Fletcher Weiss","sexo":"female","email":"mariweiss@tropoli.com","fechaNacimiento":"1992-04-04T00:00:00.000Z","comentarios":"Aliquip dolor laboris ullamco id ex labore, Ipsum eiusmod ut aliquip non cillum deserunt sunt commodo anim ad nisi excepteur eu deserunt. Sit sunt proident Lorem irure irure minim adipisicing cillum. Nostrud officia in proident velit velit sit fugiat pariatur quis ad laboris minim dolor elit. Sint velit pariatur commodo sint veniam exercitation. Duis proident minim consequat consectetur sint et tempor labore culpa esse. Exercitation laborum non esse mollit tempor ea dolor minim adipisicing mollit in aliqua. \r\nUllamco adipisicing excepteur commodo sunt nulla quis sunt velit Lorem pariatur sunt ad do incididunt. In eu nostrud ullamco laboris eu minim. Consequat sit et eiusmod officia ex sit minim sit laborum quis laborum labore non. Dolor nulla ut pariatur reprehenderit minim dolore consequat sunt aliquip ipsum esse. Excepteur consequat fugiat elit et nisi dolore aute minim nostrud et.\r\n","cursos":[{"id":"639245b6328afd80712aa9e6","curso":1,"media":5.2},{id":"639245b6328afd80712aa9e7","curso":2,"media":9.1}],"usuario":10}diegolb@ubuntu:~/iseP4JMeter$
```

Figura 2.9

2.3 Instalación de JMeter en nuestro ordenador (Windows 10)

Ahora queremos instalar JMeter, para ello vamos a entrar en la página web de JMeter (referencias) y luego a la sección de descargas. Una vez quí, descargamos un .zip. Luego lo descomprimiremos, entramos en la carpeta bin y abrimos la aplicación ApacheJMeter con java. Una vez entramos vemos una interfaz en la que pone “Plan de Pruebas” (figura 2.10)

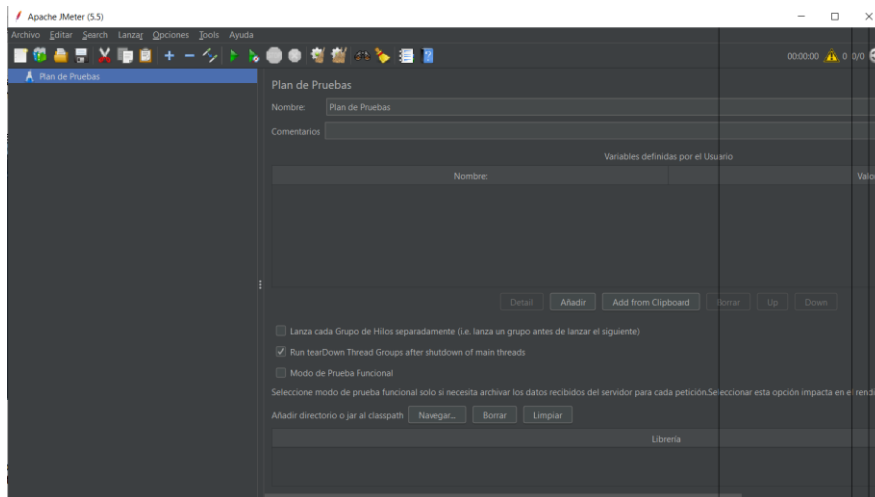


Figura 2.10

2.4 Creación del test de JMeter

Como ya tenemos creado un test sin nada podemos usar este. Le cambiaremos el nombre a ETSII Alumnos API. Ahora vamos a añadir la ip del host y el puerto (192.168.56.105:3000) tal y como viene en el primer punto del enunciado.

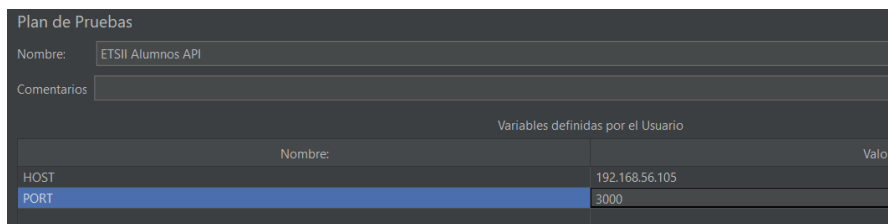


Figura 2.11

Ahora vamos a crear dos grupos diferentes de hebras que van a simular a los alumnos y a los administradores. Para ello hay que hacer click sobre el plan de pruebas, darle a añadir, a hilos y luego a grupos de hilos. Por otro lado, vamos también a modificar el número de hilos (usuarios), el periodo de subida (el tiempo que se demora en alcanzar el máximo número de peticiones) y las iteraciones del bucle (figura 2.12).

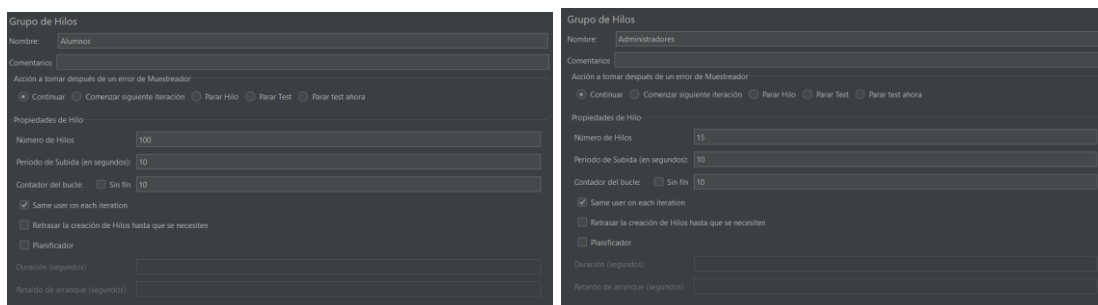


Figura 2.12

Ahora vamos a crear el gestor de autenticación de http. Para ello le damos click derecho al plan, le damos a insertar elemento de configuración y posteriormente al gestor de autenticación http. En este tenemos que añadir

la contraseña que se puede ver por ejemplo en el pruebaEntorno.sh (**laApiDeLaETSIIaLache**), el usuario que es **etsiiApi** y el URL base que es **http://\${HOST}:\${PORT}/api/v1/auth/login**. Con este gestor de autenticación podemos acceder como alumnos o como administradores.

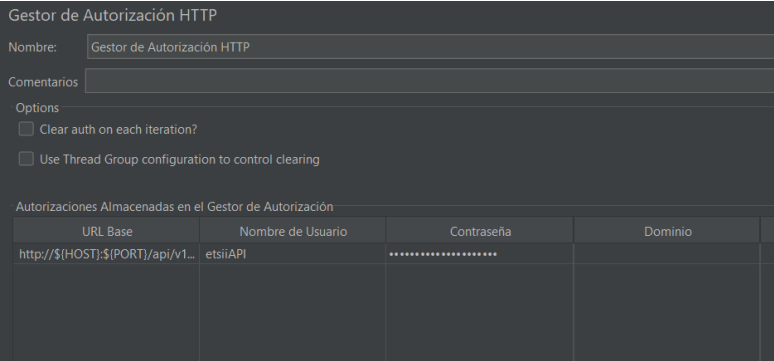


Figura 2.13

Luego dentro de cada uno de los hilos tenemos que crear la petición post, que nos devolverá una expresión regular que la utilizaremos para saber quien es el alumno o el administrador. Para ello le damos click derecho a alumnos y luego a administradores, le damos a añadir, muestreador, y petición http. La configuración la ponemos como en la **figura 2.14**. Ponemos login y password para que estos parámetros se rellenen directamente con los datos del fichero csv. Como hemos instalado JMeter en Windows, necesitamos descargar el repositorio git para tener los datos de alumnos y administradores.

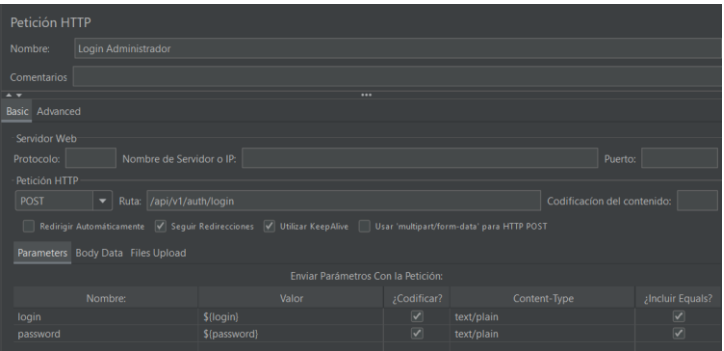
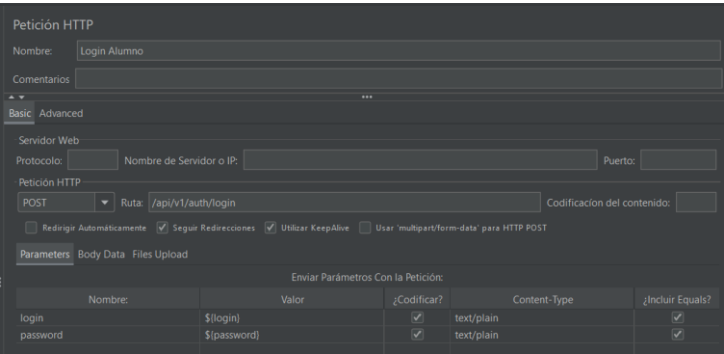


Figura 2.14

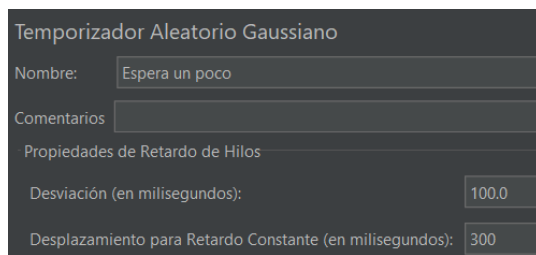
Ahora hay que añadir el extractor de expresiones regulares que enviará la expresión regular a un token. Para ello le damos click derecho a la petición http de ambos hilos, a añadir - post procesadores - extractor de expresiones regulares (figura 2.15). Aquí básicamente lo que ocurre es que se extrae uno a más caracteres al añadir “.+”.

Figura 2.15

Para que todo esto funcione, necesito crear la configuración del CSV y así poder extraer los usuarios y contraseñas de los alumnos y administradores. Para ello, le damos a añadir a los dos hilos - elemento de configuración - Configuración del CSV Data Set. Una vez creado, rellenamos como en la figura 2.16. Hay que añadir la dirección del fichero csv de windows.

Figura 2.16

Ahora podemos añadir también temporizadores de esperas aleatorias para los alumnos y para los administradores. Añadir - temporizador - temporizador aleatorio gaussiano.



Temporizador Aleatorio Gaussiano

Nombre:

Comentarios

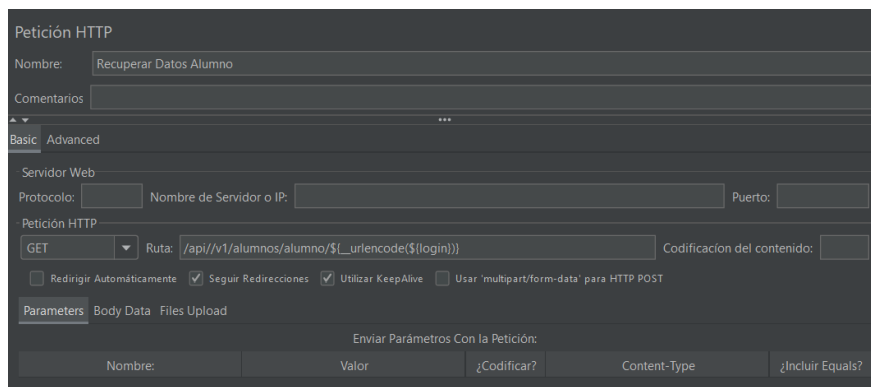
Propiedades de Retardo de Hilos

Desviación (en milisegundos):

Desplazamiento para Retardo Constante (en milisegundos):

Figura 2.17

Para recuperar los datos de los alumnos tenemos que hacer un get, que es otra petición http. Además, hay que añadir un header manager (gestor de cabecera http) para poder recuperar los datos del token. Para añadir el get tenemos que crear otra petición http como la del post con la configuración de la figura 2.18, y para el header manager (figura 2.19) le damos a añadir - elemento de configuración - gestor de cabecera http.



Petición HTTP

Nombre:

Comentarios

Basic Advanced

Servidor Web

Protocolo: Nombre de Servidor o IP: Puerto:

Petición HTTP

GET Codificación del contenido:

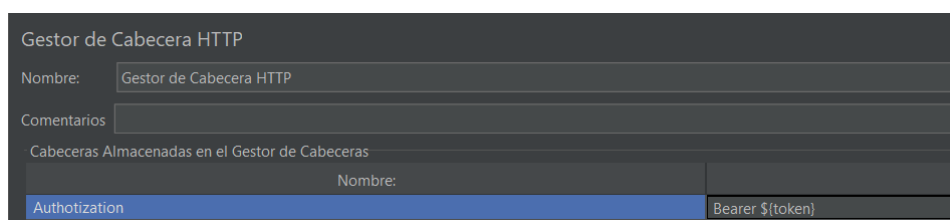
☐ Redirigir Automáticamente ☒ Seguir Redirecciones ☒ Utilizar KeepAlive ☐ Usar 'multipart/form-data' para HTTP POST

Parameters Body Data Files Upload

Enviar Parámetros Con la Petición:

Nombre:	Valor	¿Codificar?	Content-Type	¿Incluir Equals?

Figura 2.18



Gestor de Cabecera HTTP

Nombre:

Comentarios

Cabeceras Almacenadas en el Gestor de Cabeceras

Nombre:	Valor
Authorization	Bearer \${token}

Figura 2.19

Ahora vamos a añadir el archivo que registra las peticiones que los usuarios hacen a la aplicación web para que el muestreo para simular el acceso de los administradores lo coja el archivo apiAlumnos.log (figura 2.20). Para esto le damos a añadir en el hilo administradores - muestreador - muestreador de acceso log. Además, hay que añadir el gestor de cabecera http para obtener los datos de los administradores (figura 2.21).

Figura 2.20

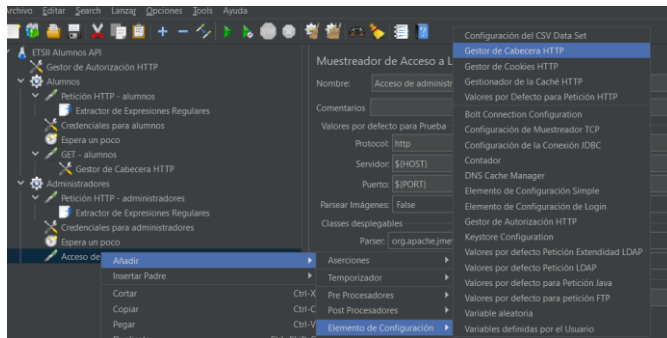


Figura 2.21

Por otro lado, vamos a añadir los valores por defecto para la petición http. Con esto, podremos asegurarnos de que todas las peticiones van a usar el mismo servidor y puerto. Para añadir esto tenemos que darle a añadir - elemento de configuración - valores por defecto para petición http.

Figura 2.22

Ahora vamos a añadir el apartado del análisis. En concreto añadiremos reporte resumen, informe agregado y ver resultados árbol. Finalmente, el JMeter quedará como en la figura 2.24.

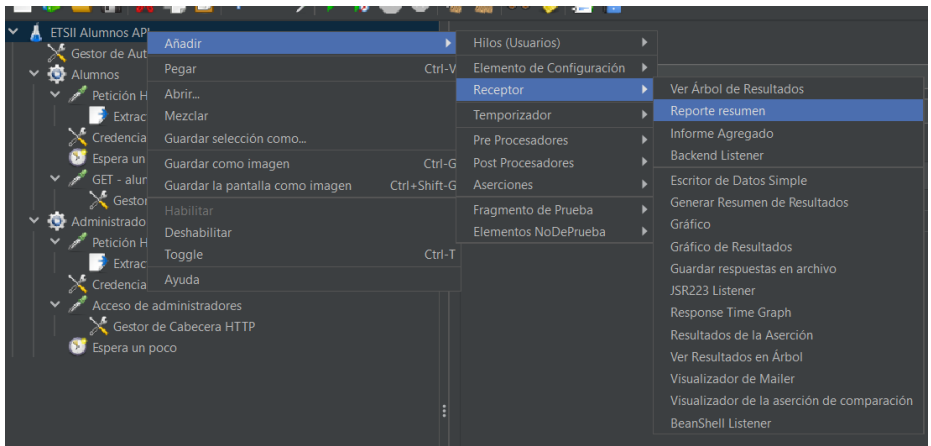


Figura 2.23

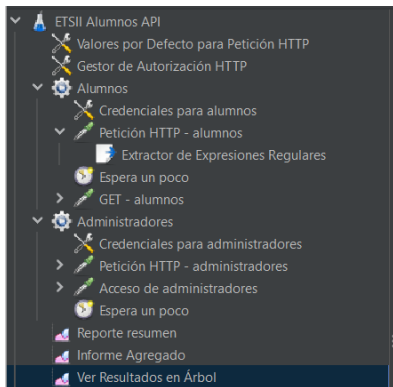


Figura 2.24

2.5 Prueba del test de JMeter

Ahora voy a pasar a mostrar los resultados de mis tests. Por un lado, estará el de reporte resumen (figura 2.25), luego el del informe agregado (figura 2.26) y luego el de ver resultados del árbol (figura 2.27).

Reporte resumen

Nombre:

Comentarios:

Escribir todos los datos a Archivo

Nombre de archivo: Log/Mostrar sólo: ☐ Escribir en Log Sólo Errores ☐ Éxitos

Etiqueta	# Muestras	Media	Min	Max	Desv. Estándar	% Error	Rendimiento	Kb/sec	Sent KB/sec	Media de Bytes
Login Adminis...	150	5	2	35	5,62	0,00%	10,3/sec	6,20	3,38	618,0
Login Alumno...	1000	4	1	40	4,49	0,00%	65,8/sec	38,98	21,57	606,9
Recuperar Dat...	1000	5	1	51	5,89	0,00%	64,3/sec	95,44	23,85	1521,2
http://192.168...	15	8	3	27	6,92	0,00%	1,6/sec	1,89	0,00	1207,0
http://192.168...	15	8	3	30	8,09	0,00%	1,6/sec	2,06	0,00	1305,0
http://192.168...	15	6	3	15	3,54	0,00%	1,7/sec	2,69	0,00	1605,0
http://192.168...	15	8	2	53	13,45	0,00%	1,7/sec	3,35	0,00	2044,0
http://192.168...	15	8	3	37	8,68	0,00%	1,7/sec	1,84	0,00	1142,0
http://192.168...	15	7	3	40	9,14	0,00%	1,6/sec	2,48	0,00	1575,0
http://192.168...	15	6	3	26	5,61	0,00%	1,6/sec	2,00	0,00	1258,0
http://192.168...	15	12	2	60	16,18	0,00%	1,6/sec	1,74	0,00	1082,0
http://192.168...	15	6	3	27	5,95	0,00%	1,7/sec	2,46	0,00	1512,0
http://192.168...	15	7	2	24	6,88	0,00%	1,6/sec	1,72	0,00	1083,0
Total	2300	5	1	60	5,68	0,00%	147,2/sec	151,76	47,91	1055,6

☐ ¿Incluir el nombre del grupo en la etiqueta? ☒ Guardar la cabecera de la tabla

Figura 2.25

Informe Agregado

Nombre: Informe Agregado

Comentarios

Escribir todos los datos a Archivo

Nombre de archivo

Navegar...

Log/Mostrar sólo:

☐ Escribir en Log Sólo Errores

☐ Éxitos

Configurar

Etiqueta	# Muestras	Media	Mediana	90% Line	95% Line	99% Line	Min	Max	% Error	Rendimiento	Kb/sec	Sent Kb/sec
Login Admin...	150	5	4	10	16	33	2	35	0,00%	10,3/sec	6,20	3,3
Login Alum...	1000	4	3	8	13	25	1	40	0,00%	63,8/sec	38,98	21,1
Recuperar D...	1000	5	3	10	16	35	1	51	0,00%	64,3/sec	95,44	23,8
http://192.1...	15	8	6	16	19	27	3	27	0,00%	1,6/sec	1,89	0,0
http://192.1...	15	8	6	12	27	30	3	30	0,00%	1,6/sec	2,06	0,0
http://192.1...	15	6	4	11	12	15	3	15	0,00%	1,7/sec	2,69	0,0
http://192.1...	15	8	3	11	27	53	2	53	0,00%	1,7/sec	3,35	0,0
http://192.1...	15	8	7	9	21	37	3	37	0,00%	1,7/sec	1,84	0,0
http://192.1...	15	7	5	8	16	40	3	40	0,00%	1,6/sec	2,48	0,0
http://192.1...	15	6	5	9	10	26	3	26	0,00%	1,6/sec	2,00	0,0
http://192.1...	15	12	6	21	40	60	2	60	0,00%	1,6/sec	1,74	0,0
http://192.1...	15	6	5	8	12	27	3	27	0,00%	1,7/sec	2,40	0,0
http://192.1...	15	7	3	16	20	24	2	24	0,00%	1,6/sec	1,72	0,0
Total	2300	5	3	9	16	32	1	60	0,00%	147,2/sec	151,76	47,0

☐ ¿Incluir el nombre del grupo en la etiqueta?

Guardar la tabla de datos

☒ Guardar la cabecera de la tabla

Figura 2.26

Ver Resultados en Árbol

Nombre: Ver Resultados en Árbol

Comentarios

Escribir todos los datos a Archivo

Nombre de archivo

Navegar...

Log/Mostrar sólo:

☐ Escribir en Log Sólo Errores

☐ Éxitos

Configurar

Muestra #	Tiempo de comi...	Nombre del hilo	Etiqueta	Tiempo de Mues...	Estado	Bytes	Sent Bytes	Latency	Connect Time...
76	18:30:54.169	Alumnos 1-3	Login Alumno	2		606	335	2	0
77	18:30:54.194	Administradores...	http://192.168.5...	4		1207	0	4	0
78	18:30:54.225	Alumnos 1-10	Recuperar Datos...	2		2322	378	2	0
79	18:30:54.263	Alumnos 1-19	Login Alumno	4		606	337	4	1
80	18:30:54.272	Administradores...	Login Administr...	3		618	337	3	1
81	18:30:54.282	Alumnos 1-9	Login Alumno	3		606	339	3	0
82	18:30:54.337	Alumnos 1-11	Recuperar Datos...	2		1046	378	2	0
83	18:30:54.342	Alumnos 1-20	Login Alumno	3		606	339	3	0
84	18:30:54.352	Administradores...	Login Administr...	5		618	338	5	0
85	18:30:54.352	Alumnos 1-1	Recuperar Datos...	7		1208	383	7	0
86	18:30:54.353	Alumnos 1-17	Recuperar Datos...	7		2201	379	7	0
87	18:30:54.386	Alumnos 1-3	Recuperar Datos...	4		1533	380	4	0
88	18:30:54.387	Alumnos 1-6	Login Alumno	3		606	341	3	0
89	18:30:54.393	Alumnos 1-16	Recuperar Datos...	2		1176	377	2	0
90	18:30:54.397	Alumnos 1-14	Login Alumno	3		606	335	3	0
91	18:30:54.418	Alumnos 1-4	Recuperar Datos...	3		1159	379	3	0
92	18:30:54.434	Alumnos 1-12	Login Alumno	3		606	332	3	0
93	18:30:54.442	Alumnos 1-10	Login Alumno	3		606	337	3	0
94	18:30:54.492	Administradores...	Login Administr...	4		618	335	4	0
95	18:30:54.497	Alumnos 1-2	Login Alumno	3		606	331	3	0

Figura 2.27

2.6 Referencias

Página web para descargar JMeter en el ordenador anfitrión.

- 1- https://jmeter.apache.org/download_jmeter.cgi

Para crear este test he leído el readme.md del repositorio y he explorado todas las diferentes opciones que se muestran dentro de este repositorio.