

Practica-4-Resuelta.pdf



Zukii



Ingeniería de Servidores



3º Doble Grado en Ingeniería Informática y Administración y Dirección de Empresas



**Escuela Técnica Superior de Ingenierías Informática y de Telecomunicación
Universidad de Granada**

Práctica 4 - Resuelta

(Los exámenes de esta práctica y la 3, los tengo resueltos en otro documento en mi perfil @Zukii)

Práctica 4

Asignatura: Ingeniería de Servidores
Profesor: David Palomar Saéz

Índice:

0) Aclaraciones previas.....	2
1) Ejercicio 1: Benchmarks en Ubuntu y CentOS.....	2
1.0) Enunciado del Ejercicio.....	2
1.1) Instalación de Phoronix en Ubuntu.....	2
1.2) Elección de los benchmarks a ejecutar.....	3
1.3) Prueba en Ubuntu de los Benchmarks.....	4
1.4) Instalación de Phoronix en CentOS.....	7
1.5) Prueba en CentOS de los Benchmarks.....	8
1.6) Diferencias vistas.....	12
2) Ejercicio 2: Test de JMeter sobre dos contenedores.....	12
2.0) Enunciado del Ejercicio.....	12
2.1) Instalación de Docker, Docker-compose y la aplicación en Ubuntu Server.....	13
2.2) Instalación de JMeter en el Sistema Anfitrión (W10)	17
2.3) Creación del Test de JMeter.....	18
2.4) Prueba del Test de JMeter.....	24

0) Aclaraciones previas

@Zukii on Wuolah

Para la realización de los ejercicios, partiremos de las máquinas virtuales de Ubuntu y CentOS usadas en la práctica 3. Además, el prompt utilizado en las máquinas virtuales y conectados mediante ssh fue: “> export PS1='alvaroVR@h - \t - \d \$'”.

Sin nada más, a priori, que comentar, comenzaremos con la práctica:

1) Ejercicio 1: Benchmarks en Ubuntu y CentOS

1.0) Enunciado del Ejercicio

Una vez que haya indagado sobre los benchmarks disponibles, seleccione como mínimo dos de ellos y proceda a ejecutarlos en Ubuntu y CentOS. Comente las diferencias.

1.1) Instalación de Phoronix en Ubuntu

Para la instalación, tendremos que hacerlo sin apt, ya que si hacemos “> apt search phoronix” no nos sale ningún resultado. En nuestro caso, seguiremos los pasos de la siguiente web:

<https://ubunlog.com/phoronix-test-suite-una-herramienta-para-benchmark-multiplataforma/>

Copiaremos los comandos que aparecen en la web, conectados mediante ssh a la máquina Ubuntu desde el anfitrión:

```
alvaroVR@alvaro - 13:47:02 - Thu Dec 09 $ wget http://phoronix-test-suite.com/releases/pts.debian/files/phoronix-test-suite_8.6.0_all.deb
--2021-12-09 13:47:02-- http://phoronix-test-suite.com/releases/pts.debian/files/phoronix-test-suite_8.6.0_all.deb
Connecting to phoronix-test-suite.com (phoronix-test-suite.com)[192.211.48.82]:80... connected.
HTTP request sent, awaiting response... 200 OK
length: 603468 (589K) [application/x-debian-package]
Saving to: 'phoronix-test-suite_8.6.0_all.deb'

phoronix-test-suite_8.6.0_all 100%[=====] 589.32K 212KB/s in 2.8s

2021-12-09 13:47:06 (212 KB/s) - 'phoronix-test-suite_8.6.0_all.deb' saved [603468/603468]

alvaroVR@alvaro - 13:47:06 - Thu Dec 09 $ sudo dpkg -i phoronix-test-suite_8.6.0_all.deb
Selecting previously unselected package phoronix-test-suite.
(Reading database ... 111272 files and directories currently installed.)
Preparing to unpack phoronix-test-suite_8.6.0_all.deb ...
Unpacking phoronix-test-suite (8.6.0) ...
Setting up phoronix-test-suite (8.6.0) ...
Processing triggers for mime-support (3.6ubuntu1) ...
Processing triggers for man-db (2.9.1-1) ...
alvaroVR@alvaro - 13:47:17 - Thu Dec 09 $ sudo apt -f install
Reading package lists... Done
Building dependency tree
Reading state information... Done
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
alvaroVR@alvaro - 13:47:25 - Thu Dec 09 $ sudo apt update -f
E: Command line option '-f' [from -f] is not understood in combination with the other options.
alvaroVR@alvaro - 13:49:02 - Thu Dec 09 $ sudo apt -f update
E: Command line option '-f' [from -f] is not understood in combination with the other options.
alvaroVR@alvaro - 13:49:08 - Thu Dec 09 $ sudo apt update
Hit:1 http://repo.zabbiix.com/zabbiix/2.0/ubuntu focal InRelease
Hit:2 http://es.archive.ubuntu.com/ubuntu focal InRelease
Hit:3 http://es.archive.ubuntu.com/ubuntu focal-updates InRelease
Hit:4 http://es.archive.ubuntu.com/ubuntu focal-backports InRelease
Hit:5 http://es.archive.ubuntu.com/ubuntu focal-security InRelease
Reading package lists... Done
Building dependency tree
Reading state information... Done
All packages are up to date.
alvaroVR@alvaro - 13:49:15 - Thu Dec 09 $ export PS1='alvaroVR@h - \t - \d $'
```

Y con esto tendremos ya instalado phoronix-test-suite.

1.2) Elección de los benchmarks a ejecutar

Para ver qué benchmarks podemos hacer, ponemos el comando “> *phoronix-test-suite list-available-tests*” y nos aparecerá una lista de los posibles test.

Esta lista tiene tres columnas: una con el nombre del test, otra con una breve descripción de lo que trata el test y por último, el componente al que va enfocado el benchmark.

Así, si ejecutamos el comando y vemos el resultado, tiene que salir algo tal que así:

```
alvaroVR@alvaro - 14:10:29 - Thu Dec 09 $ phoronix-test-suite list-available-tests

Phoronix Test Suite v8.6.0
AN OUTDATED VERSION OF THE PHORONIX TEST SUITE IS INSTALLED.
THE VERSION IN USE IS 8.6.0 (8600), BUT THE LATEST IS PTS-CORE 10610.
VISIT HTTPS://WWW.PHORONIX-TEST-SUITE.COM/ TO UPDATE THIS SOFTWARE.

Available Tests

pts/ai-benchmark          - AI Benchmark Alpha          System
pts/aircrack-ng           - Aircrack-ng                 Processor
pts/amg                   - Algebraic Multi-Grid Benchmark Processor
pts/aobench               - AOBench                     Processor
pts/aom-av1               - AOM AV1                     Processor
pts/apache                - Apache HTTP Server          System
pts/apache-siege          - Apache Siege                 System
pts/appleseed              - Appleseed                    System
pts/arrayfire              - ArrayFire                    Processor
pts/askap                  - ASKAP                         System
pts/asmfish               - asmFish                      Processor
pts/astcenc                - ASTC Encoder                 System
pts/avifenc                - libavif avifenc              Processor
pts/basemark              - Basemark GPU                 System
pts/basis                  - Basis Universal              System
pts/blake2                 - BLAKE2                       Processor
pts/blender               - Blender                       System
pts/blogbench              - BlogBench                     Disk
pts/blosc                  - C-Blosc                       Processor
pts/bork                   - Bork File Encrypter          Processor
pts/botan                  - Botan                         Processor
pts/brl-cad                - BRL-CAD                      System
pts/build-apache           - Timed Apache Compilation     Processor
pts/build-clash            - Timed Clash Compilation       Processor
pts/build-eigen            - Timed Eigen Compilation       Processor
pts/build-erlang           - Timed Erlang/OTP Compilation  Processor
pts/build-ffmpeg           - Timed FFmpeg Compilation      Processor
pts/build-gcc              - Timed GCC Compilation         Processor
pts/build-gdb              - Timed GDB GNU Debugger Compilation Processor
pts/build-godot            - Timed Godot Game Engine Compilation Processor
pts/build-imagemagick       - Timed ImageMagick Compilation Processor
pts/build-linux-kernel     - Timed Linux Kernel Compilation Processor
```

(Nota, esta captura y otras no aparecen enteras porque no caben).

Ahora, lo que tocaría, sería seleccionar los tests que vamos a probar. En mi caso, he optado por un test enfocado en el procesador y otro en la memoria. Los elegidos son los siguientes:

```
pts/stressapptest          - Stressful Application Test    Memory
pts/sudokut                 - Sudoku                        Processor
```

Para más información podemos ver:

<https://openbenchmarking.org/test/pts/stressapptest>
<https://openbenchmarking.org/test/pts/sudokut>

En resumen, el primero trata de comprobar la estabilidad de la RAM y el segundo sobre cuánto tarda el ordenador en resolver 100 sudokus.

1.3) Prueba en Ubuntu de los Benchmarks

Ahora tendremos que instalar el benchmark que vamos a usar con el comando:

"> *phoronix-test-suite install <nombre>*". Así, ejecutaremos los comandos:
"> *phoronix-test-suite install pts/stressapptest*" y "> *phoronix-test-suite install pts/sudokut*"

Así, instalamos el primer benchmark:

```
alvaroVR@alvaro - 14:48:46 - Thu Dec 09 $ sudo phoronix-test-suite install pts/stressapptest

Phoronix Test Suite v8.6.0
AN OUTDATED VERSION OF THE PHORONIX TEST SUITE IS INSTALLED.
THE VERSION IN USE IS 8.6.0 (8600), BUT THE LATEST IS PTS-CORE 10610.
VISIT HTTPS://WWW.PHORONIX-TEST-SUITE.COM/ TO UPDATE THIS SOFTWARE.

The following dependencies are needed and will be installed:

- build-essential
- autoconf
- mesa-utils
- unzip
- apt-file

This process may take several minutes.
Extracting templates from packages: 100%
Reading package lists...
Building dependency tree...
Reading state information...
unzip is already the newest version (6.0-25ubuntu1).
unzip set to manually installed.
The following additional packages will be installed:
  automake autotools-dev binutils binutils-common binutils-x86-64-linux-gnu
  cpp cpp-9 dpkg-dev fakeroot g++ g++-9 gcc gcc-9 gcc-9-base
```

Y procedemos a ejecutar el benchmark con:

"> *phoronix-test-suite run pts/stressapptest*"

```
Memory Test Configuration
1: 1
2: 2
3: 4
4: 8
5: Test All Options
** Multiple items can be selected, delimit by a comma. **
Threads: 2

1: 2MB
2: 64MB
3: 256MB
4: 1GB
5: Test All Options
** Multiple items can be selected, delimit by a comma. **
RAM To Test: 2

1: 30 Seconds
2: 1 Minute
3: 3 Minute
4: 10 Minutes
5: 30 Minutes
6: 1 Hour
7: Test All Options
** Multiple items can be selected, delimit by a comma. **
Duration: 2
```

Luego, nos pide determinar unos parámetros para el benchmark, como son el número de hebras (2), la RAM que asignamos (64MB) y la duración (1m):

Posteriormente nos pedirá si queremos guardar los resultados. En mi caso, acepto y los guardo en un archivo llamado “stressappbench2”. También nos da la opción de poner una descripción única para el resultado y una descripción más extensa.

```
Would you like to save these test results (Y/n): y

Recently Saved Test Results:
stressappbench [Today]

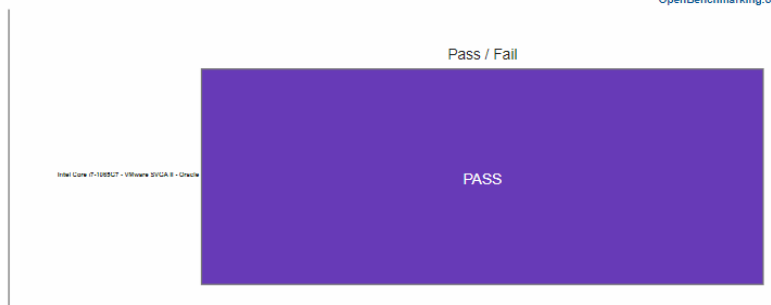
Enter a name for the result file: stressappbench2
Enter a unique name to describe this test run / configuration:
```

Al finalizar el test, nos preguntará si queremos ver los resultados. En mi caso, acepto y luego te pregunta si quieres subir tus resultados a OpenBenchmarking.org (cosa que recomiendo de cara a ayudar a la comunidad y que luego, mediante una URL podemos ver el resultado del benchmark. En mi caso: <https://openbenchmarking.org/result/2112125-SP-STRESSAPP75>).

Si procedemos a observar los resultados, nos da información sobre dónde se ha realizado el benchmark (nuestra máquina virtual) y lo importante de este benchmark, que es comprobar si ha pasado la prueba de estabilidad, cosa que lo hace como podemos observar en la web al final (o encima de donde le damos a ver los resultados del test, como luego haremos en CentOS).

Stressful Application Test

Threads: 2 - RAM To Test: 64MB - Duration: 1 Minute



```
Do you want to view the text results of the testing (Y/n): y
stressappbench2
Oracle VMare testing on Ubuntu 20.04 via the Phoronix Test Suite.

Intel Core i7-1065G7 - VMware SVGA II - Oracle:
Processor: Intel Core i7-1065G7 (1 Core), Motherboard: Oracle VirtualBox v1.2, Chipset: Intel 440FX 82441FX PMC, Memory: 2048MB, Disk: 11GB VBOX HDD, Graphics: VMware SVGA II, Audio: Intel 82801AA AC 97
Audio, Network: 2 x Intel 82540EM

OS: Ubuntu 20.04, Kernel: 5.4.0-90-generic (x86_64), Compiler: GCC 9.3.0, File-System: ext4, Screen Resolution: 2048x2048, System Layer: Oracle VMare

Stressful Application Test
Threads: 2 - RAM To Test: 64MB - Duration: 1 Minute
Pass / Fail
Intel Core i7-1065G7 - VMware SVGA II - Oracle .

Would you like to upload the results to OpenBenchmarking.org (y/n): y
Would you like to attach the system logs (lspci, dmesg, lsusb, etc) to the test result (y/n): y
Results Uploaded To: https://openbenchmarking.org/result/2112125-SP-STRESSAPP75

alvaroVR@alvaro - 16:14:17 - Sun Dec 12 $
```

Ahora, instalamos el segundo benchmark:

```
alvaroVR@alvaro - 16:58:19 - Sun Dec 12 $ phoronix-test-suite install pts/sudoku

Phoronix Test Suite v8.6.0
AN OUTDATED VERSION OF THE PHORONIX TEST SUITE IS INSTALLED.
THE VERSION IN USE IS 8.6.0 (8600), BUT THE LATEST IS PTS-CORE 10610.
VISIT HTTPS://WWW.PHORONIX-TEST-SUITE.COM/ TO UPDATE THIS SOFTWARE.

The following dependencies are needed and will be installed:

- tcl

This process may take several minutes.
Reading package lists...
Building dependency tree...
Reading state information...
The following additional packages will be installed:
  libtcl8.6 tcl8.6
Suggested packages:
  tcl-tclreadline
The following NEW packages will be installed:
  libtcl8.6 tcl tcl8.6
0 upgraded, 3 newly installed, 0 to remove and 0 not upgraded.
Need to get 922 kB of archives.
After this operation, 4197 kB of additional disk space will be used.
Get:1 http://es.archive.ubuntu.com/ubuntu focal/main amd64 libtcl8.6 amd64 8.6.10+dfsg-1 [902 kB]
Get:2 http://es.archive.ubuntu.com/ubuntu focal/main amd64 tcl8.6 amd64 8.6.10+dfsg-1 [14.8 kB]
Get:3 http://es.archive.ubuntu.com/ubuntu focal/universe amd64 tcl amd64 8.6.9+1 [5112 B]
Fetched 922 kB in 3s (355 kB/s)
Selecting previously unselected package libtcl8.6:amd64.
(Reading database ... 118790 files and directories currently installed.)
Preparing to unpack .../libtcl8.6_8.6.10+dfsg-1_amd64.deb ...
Unpacking libtcl8.6:amd64 (8.6.10+dfsg-1) ...
Selecting previously unselected package tcl8.6.
Preparing to unpack .../tcl8.6_8.6.10+dfsg-1_amd64.deb ...
Unpacking tcl8.6 (8.6.10+dfsg-1) ...
Selecting previously unselected package tcl.
Preparing to unpack .../archives/tcl_8.6.9+1_amd64.deb ...
```

Ahora probamos a ejecutarlo con: “> *phoronix-test-suite run pts/sudoku*”

Guardaremos los resultados en un archivo llamado “sudukutbench” y luego subiremos los resultados a OpenBenchmarking.org (Link para ver los resultados de una forma más bonita: <https://openbenchmarking.org/result/2112128-SP-SUDOKUTBE27>).

Si vemos los resultados veremos algo tal que así:

```
Sudoku 0.4:
pts/sudoku-1.0.1
Test 1 of 1
Estimated Trial Run Count: 3
Estimated Time To Completion: 2 Minutes [17:11 UTC]
Started Run 1 @ 17:10:04
Started Run 2 @ 17:10:17
Started Run 3 @ 17:10:30
Started Run 4 @ 17:10:43 *

Total Time:
12.71982216835
11.803972959518
12.138022899628
12.37749804364

Average: 12.26 Seconds
Deviation: 3.15%

Result compared to 199,282 OpenBenchmarking.org samples; median: 30.93. Box plot of sampling:
[-----#-----]
This Result (99th Percentile): 12.26 ^

2 x Intel Xeon 2.80GHz: 77 ^          AMD Athlon 64 X2 3800+: 56.69 ^          2 x Intel Xeon E5-2620 0: 25.54 ^ Intel Core i3-2100: 22.13 ^ Intel Core i5-11600K: 8.115 ^
SPARC64-VII sparcv9: 82 ^          2 x Intel Xeon X5550: 37.48 ^          2 x Intel Xeon X5550: 37.48 ^          Intel Core i7-7700K: 9.51 ^
1 x Intel Xeon 2.60GHz: 85          2 x Intel Xeon E5420: 38.61 ^          Intel Core i7-5557U: 13.59 ^

Do you want to view the text results of the testing (Y/n): y
sudukutbench
Oracle VMare testing on Ubuntu 20.04 via the Phoronix Test Suite.

Intel Core i7-1065G7:
Processor: Intel Core i7-1065G7 (1 Core), Motherboard: Oracle VirtualBox v1.2, Chipset: Intel 440FX 8241FX PMC, Memory: 2048MB, Disk: 11GB VBOX HDD, Graphics: VMware SVGA II, Audio: Intel 82801AA AC 97
Audio, Network: 2 x Intel 82540EM

OS: Ubuntu 20.04, Kernel: 5.4.0-90-generic (x86_64), Compiler: GCC 9.3.0, File System: ext4, Screen Resolution: 2048x2048, System Layer: Oracle VMare

Sudoku 0.4
Total Time
Seconds < Lower Is Better
Intel Core i7-1065G7 - 12.26 |-----#-----
```

Sobre los resultados podemos comentar que hemos tardado de media en cada sudoku 12.26 segundos, con una desviación típica del 3%. También podemos ver como este resultado se encuentra el percentil 99:


```
Average: 12.26 Seconds
Deviation: 3.15%
Result compared to 199,282 OpenBenchmarking.org samples; median: 30.93. Box plot of sampling:
[-----#-----]
This Result (99th Percentile): 12.26 ^
2 x Intel Xeon 2.80GHz: 77 ^ AMD Athlon 64 X2 3800+: 56.69 ^ 2 x Intel Xeon E5-2620 v4: 15.54 ^ Intel Core i3-2100: 22.13 ^ Intel Core i5-11400: 9.115 ^
SPARC64-VII sparcv9: 82 ^ 2 x Intel Xeon X5550: 17.89 ^ 2 x Intel Xeon E5420: 38.61 ^ Intel Core i7-7700K: 8.51 ^
2 x Intel Xeon 2.60GHz: 85 Intel Core i7-5557U: 13.59 ^
```

Hecho esto, podemos dar por concluido la parte de Ubuntu de este ejercicio.

1.4) Instalación de Phoronix en CentOS (También me conecto con ssh desde el anfitrión)

Para la instalación seguiremos la siguiente web:

<https://arstech.net/phoronix-test-suite/>

Primero, instalamos las dependencias necesarias (wget, php-cli y php-xml y bzip2):

```
alvaroVR@localhost ~ 04:52:37 - Thu Dec 09 $ sudo yum install wget php-cli php-xml bzip2
CentOS-8 - AppStream
CentOS-8 - AppStream
CentOS-8 - Base
CentOS-8 - Base
CentOS-8 - Extras
Zabbix Official Repository - x86_64
Zabbix Official Repository - x86_64
Zabbix Official Repository non-supported - x86_64
Package bzip2-1.0.6-26.el8.x86_64 is already installed.
Dependencies resolved.

=====
Package Architecture Version Repository Size
=====
Installing:
php-cli x86_64 7.2.24-1.module_el8.2.0+313+b04d0a66 AppStream 3.1 M
php-xml x86_64 7.2.24-1.module_el8.2.0+313+b04d0a66 AppStream 188 k
wget x86_64 1.19.5-10.el8 AppStream 734 k
=====
```

Luego descargamos, descomprimos e instalamos Phoronix-test-suite:

```
alvaroVR@localhost ~ 04:56:29 - Thu Dec 09 $ wget https://phoronix-test-suite.com/releases/phoronix-test-suite-8.4.1.tar.gz
--2021-12-09 04:56:30-- https://phoronix-test-suite.com/releases/phoronix-test-suite-8.4.1.tar.gz
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|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 873706 (853K) [application/x-gzip]
Saving to: 'phoronix-test-suite-8.4.1.tar.gz'

phoronix-test-suite-8.4.1.tar 100%[=====>] 853.23K 15.1KB/s in 25s

2021-12-09 04:56:56 (33.5 KB/s) - 'phoronix-test-suite-8.4.1.tar.gz' saved [873706/873706]

alvaroVR@localhost ~ 04:56:56 - Thu Dec 09 $ tar xvfz phoronix-test-suite-8.4.1.tar.gz
phoronix-test-suite/
phoronix-test-suite/deploy/
phoronix-test-suite/deploy/phoromatic-upstart/
phoronix-test-suite/deploy/phoromatic-upstart/phoromatic-server.conf
phoronix-test-suite/deploy/phoromatic-upstart/phoromatic-client.conf
phoronix-test-suite/deploy/phoromatic-upstart/phoromatic-client.alt.conf
phoronix-test-suite/deploy/rpm-package/
phoronix-test-suite/deploy/rpm-package/build-package-rpm.php
phoronix-test-suite/deploy/farm-system-customizations/
phoronix-test-suite/deploy/farm-system-customizations/intel-xorg-headless.conf
phoronix-test-suite/deploy/farm-system-customizations/radeon-xorg-headless.conf
phoronix-test-suite/deploy/farm-system-customizations/ubuntu-initial-setup.sh
phoronix-test-suite/deploy/deployments.md
phoronix-test-suite/deploy/phoromatic-initd/
phoronix-test-suite/deploy/phoromatic-initd/phoromatic-client
phoronix-test-suite/deploy/phoromatic-rcd/
phoronix-test-suite/deploy/phoromatic-rcd/phoromatic-client
phoronix-test-suite/deploy/deb-package/
phoronix-test-suite/deploy/deb-package/build-package-deb.php
```



```
alvaroVR@localhost - 04:57:04 - Thu Dec 09 $ cd phoronix-test-suite
alvaroVR@localhost - 04:57:11 - Thu Dec 09 $ ./install-sh
ERROR: /usr is not writable. Run this installer as root or specify a different directory prefix as the first argument sent to this script.
alvaroVR@localhost - 04:57:17 - Thu Dec 09 $ sudo ./install-sh
which: no xdg-mime in (/sbin:/bin:/usr/sbin:/usr/bin)

Phoronix Test Suite Installation Completed

Executable File: /usr/bin/phoronix-test-suite
Documentation: /usr/share/doc/phoronix-test-suite/
Phoronix Test Suite Files: /usr/share/phoronix-test-suite/
```

Luego, si probamos a poner cualquier comando, nos da un error, de que nos hace falta la extensión de JSON para PHP:

```
alvaroVR@localhost - 04:59:46 - Thu Dec 09 $ phoronix-test-suite

The following PHP extensions are REQUIRED:

JSON      JSON support is required for OpenBenchmarking.org.
```

Para arreglar esto, buscamos la extensión que nos hace falta con yum poniendo: “> yum search json | grep php” y luego esta, la instalamos con “> sudo yum install php-json.x86_64”.

```
alvaroVR@localhost - 05:00:40 - Thu Dec 09 $ yum search json | grep php
Last metadata expiration check: 0:00:49 ago on Thu Dec 9 05:00:37 2021.
php-json.x86_64 : Javascript Object Notation extension for PHP
alvaroVR@localhost - 05:01:27 - Thu Dec 09 $ yum apt install php-json.x86_64
No such command: apt. Please use /usr/bin/yum --help
It could be a YUM plugin command, try: 'yum install --df-command(apt)'.
alvaroVR@localhost - 05:01:44 - Thu Dec 09 $ yum install php-json.x86_64
Error: This command has to be run under the root user.
alvaroVR@localhost - 05:01:53 - Thu Dec 09 $ sudo yum install php-json.x86_64
Last metadata expiration check: 0:00:59 ago on Thu Dec 9 04:53:00 2021.
Dependencies resolved.
=====
Package                               Architecture      Version           Repository        Size
=====
Installing:
php-json                               x86_64            7.2.24-1.module_el8.2.0+313+b04d0a66 AppStream         73 k
Transaction Summary
-----
Install 1 Package
Total download size: 73 k
Installed size: 44 k
Is this ok [y/N]: y
Downloading Packages:
php-json-7.2.24-1.module_el8.2.0+313+b04d0a66.x86_64.rpm                276 kB/s | 73 kB  00:00
-----
Total                                                                    86 kB/s | 73 kB  00:00
```

Si luego volvemos a probar a poner algún comando de phoronix-test-suite, ya si nos dejaría. Lo único a destacar, es que al ser la primera vez que ejecutamos, nos preguntará si aceptamos los términos de phoronix y si habilitamos el modo anónimo para las estadísticas. (ambas las acepto).

1.5) Prueba en CentOS de los Benchmarks

Ahora, para la prueba, necesitamos instalar los mismos benchmarks que hemos usado en Ubuntu. Los comandos para la instalación son los mismos que los usados en Ubuntu.

Para el primer benchmark, cuando procedemos a instalarlo, nos da un error de que nos falta glibc-static:

```
Error: Unable to find a match: glibc-static
```

Para arreglarlo, procedemos a intentar buscar información sobre este error y vemos como hay un hilo en stackoverflow sobre su resolución:

<https://stackoverflow.com/questions/60238534/add-libraries-for-static-compilation-centos-8>

Así, vemos que lo solucionamos poniendo los siguientes comandos:

```
> sudo dnf config-manager --enable PowerTools
> sudo dnf install glibc-static
```

```
alvaroVR@localhost - 05:37:14 - Thu Dec 09 $ sudo dnf config-manager --enable PowerTools
alvaroVR@localhost - 05:37:17 - Thu Dec 09 $ sudo dnf install glibc-static
CentOS-8 - PowerTools                               3.4 MB/s | 2.3 MB    00:00
Last metadata expiration check: 0:00:01 ago on Thu Dec  9 05:37:26 2021.
Dependencies resolved.
=====
Package                                Architecture      Version            Repository          Size
=====
Installing:
glibc-static                           x86_64             2.28-164.el8       PowerTools           2.1 M
Upgrading:
glibc                                   x86_64             2.28-164.el8       BaseOS                3.6 M
glibc-all-langpacks                   x86_64             2.28-164.el8       BaseOS                25 M
glibc-common                           x86_64             2.28-164.el8       BaseOS                1.3 M
libxcrypt                              x86_64             4.1.1-6.el8        BaseOS                73 k
Installing dependencies:
glibc-devel                            x86_64             2.28-164.el8       BaseOS                1.0 M
glibc-headers                          x86_64             2.28-164.el8       BaseOS                480 k
kernel-headers                         x86_64             4.18.0-348.2.1.el8_5 BaseOS                8.3 M
libxcrypt-devel                        x86_64             4.1.1-6.el8        BaseOS                25 k
libxcrypt-static                       x86_64             4.1.1-6.el8        PowerTools            56 k
=====
```

Ahora sí, ya podemos instalar el benchmark:

```
alvaroVR@localhost - 05:37:50 - Thu Dec 09 $ phoronix-test-suite install pts/stressapptest

Phoronix Test Suite v8.4.1
AN OUTDATED VERSION OF THE PHORONIX TEST SUITE IS INSTALLED.
THE VERSION IN USE IS 8.4.1 (8410), BUT THE LATEST IS PTS-CORE 10610.
VISIT HTTPS://WWW.PHORONIX-TEST-SUITE.COM/ TO UPDATE THIS SOFTWARE.

The following dependencies are needed and will be installed:
- gcc
- gcc-c++
- make
- autoconf
- automake
- glibc-static
- patch
- expat-devel

This process may take several minutes.
Last metadata expiration check: 0:00:45 ago on Thu Dec  9 05:37:26 2021.
Package glibc-static-2.28-164.el8.x86_64 is already installed.
Dependencies resolved.
=====
Package      Arch      Version            Repo          Size
=====
Installing:
autoconf     noarch    2.69-29.el8        AppStream     710 k
=====
```

Y probamos su ejecución con los mismos parámetros que en Ubuntu:

```
alvaroVR@localhost - 05:38:46 - Thu Dec 09 $ phoronix-test-suite run pts/stressapptest

Phoronix Test Suite v8.4.1
AN OUTDATED VERSION OF THE PHORONIX TEST SUITE IS INSTALLED.
THE VERSION IN USE IS 8.4.1 (8410), BUT THE LATEST IS PTS-CORE 10610.
VISIT HTTPS://WWW.PHORONIX-TEST-SUITE.COM/ TO UPDATE THIS SOFTWARE.

Stressful Application Test:
pts/stressapptest-1.0.1
Memory Test Configuration
1: 1
2: 2
3: 4
4: 8
5: Test All Options
** Multiple items can be selected, delimit by a comma. **
Threads: 2

1: 2MB
2: 64MB
3: 256MB
4: 1GB
5: Test All Options
** Multiple items can be selected, delimit by a comma. **
RAM To Test: 2

1: 30 Seconds
2: 1 Minute
3: 3 Minute
4: 10 Minutes
5: 30 Minutes
6: 1 Hour
7: Test All Options
** Multiple items can be selected, delimit by a comma. **
Duration: 2
```

Al ejecutarlo, al final, veremos los resultados (Link para verlo por internet: <https://openbenchmarking.org/result/2112120-SK-STRESSAPP48>):

```
Stressful Application Test:
pts/stressapptest-1.0.1 [Threads: 2 - RAM To Test: 64MB - Duration: 1 Minute]
Test 1 of 1
Estimated Trial Run Count: 1
Estimated Time To Completion: 14 Minutes [06:05 EST]
Started Run 1 @ 05:52:33
[NOTICE] Undefined: min_result in pts_test_result_parser:462
[NOTICE] Undefined: max_result in pts_test_result_parser:462
Final: PASS (Pass / Fail)
```

Y vemos, como pasa la prueba de estabilidad.

Ahora para el segundo, procedemos a instalar el benchmark con “> *phoronix-test-suite install pts/sudoku*”.

```
alvaroVR@localhost - 05:18:08 - Thu Dec 09 $ phoronix-test-suite install pts/sudoku
```

```
Phoronix Test Suite v8.4.1
AN OUTDATED VERSION OF THE PHORONIX TEST SUITE IS INSTALLED.
THE VERSION IN USE IS 8.4.1 (8410), BUT THE LATEST IS PTS-CORE 10610.
VISIT HTTPS://WWW.PHORONIX-TEST-SUITE.COM/ TO UPDATE THIS SOFTWARE.
```

The following dependencies are needed and will be installed:

```
- tcl
```

```
This process may take several minutes.
Last metadata expiration check: 0:26:06 ago on Thu Dec 9 04:53:00 2021.
Dependencies resolved.
```

Package	Architecture	Version	Repository	Size
=====				
Installing:				
tcl	x86_64	1:8.6.8-2.el8	BaseOS	1.1 M

Transaction Summary

=====

Install 1 Package

Total download size: 1.1 M

Installed size: 4.2 M

Downloading Packages:

Ahora probamos a ejecutarlo con: `> phoronix-test-suite run pts/sudoku`

Guardaremos los resultados en un archivo llamado “sudukutbench” y luego subiremos los resultados a OpenBenchmarking.org (Link para ver los resultados de una forma más bonita: <https://openbenchmarking.org/result/2112124-SK-SUDOKUTBE26>).

```
Sudoku 0.4:
pts/sudoku-1.0.1
Test 1 of 1
Estimated Trial Run Count: 3
Estimated Time To Completion: 2 Minutes [05:22 EST]
Started Run 1 @ 05:20:17
Started Run 2 @ 05:21:07
Started Run 3 @ 05:21:46
Started Run 4 @ 05:22:04 *
Started Run 5 @ 05:22:22 *
Started Run 6 @ 05:22:39 *
Started Run 7 @ 05:22:57 *
Started Run 8 @ 05:23:14 *
Started Run 9 @ 05:23:30 *
Started Run 10 @ 05:23:47 *
Started Run 11 @ 05:24:04 *
Started Run 12 @ 05:24:21 *

Total Time:
49.153222084045
38.12140583992
16.897970199585
17.288281917572
15.937389135361
16.221601963043
16.044873952866
15.826927900314
15.828149795532
16.166773080826
15.888979911804
15.974460840225

Average: 20.78 Seconds
Deviation: 52.66%
```

Podemos ver en los resultados como hemos tardado de media 20.78 segundos por cada sudoku, aunque viendo la secuencia, podemos ver que hay una gran desviación entre las muestras. Por ejemplo, el por caso es 49s y el mejor es 16s.

1.6) Diferencias vistas

Las diferencias que hemos podido observar, han sido sobre todo en el segundo benchmark (el de los sudokus) ya que el primero, ambos han pasado la prueba de estabilidad de la RAM con los mismo parámetros. Si tuviésemos que destacar algo del primero podemos destacar que para CentOS nos ha hecho falta biblioteca para la compilación estática pero ninguna diferencia notable más.

Respecto al segundo, podemos observar como en CentOS, la media y la desviación es más alta respecto a Ubuntu que es más “equilibrado”. También podemos observar que los resultados en CentOS se muestran de forma diferente (al menos los que salen inmediatamente después de ejecutar el benchmark) y no nos aparece la comparativa entre procesadores ni el percentil al que pertenece el resultado.

2) Ejercicio 2: Test de JMeter sobre dos contenedores

2.0) Enunciado del Ejercicio

Tras probar un test básico para una web, 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:

- *El test debe tener parametrizados el Host y el Puerto en el Test Plan (puede hacer referencia usando \$param).*
- *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.*
- *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.*
- *El muestreo para simular el acceso de los administradores lo debe coger el archivo apiAlumnos.log (usando un Access Log Sampler).*

- Use una expresión regular (Regular Expression Extractor) para extraer el token JWT que hay que añadir a la cabecera de las peticiones (usando HTTP Header Manager)

2.1) Instalación de Docker, Docker-compose y la aplicación en Ubuntu Server

(Nos volvemos a conectar desde el anfitrión con ssh a Ubuntu Server).

Empezaremos con Docker y procederemos como viene en el PDF de la Práctica 4:

Primero, añadimos la llave para validar el repositorio y añadimos el repositorio:

```
alvarov@alvarov - 11:19:51 - Mon Dec 13 $ curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -
alvarov@alvarov - 11:43:18 - Mon Dec 13 $ curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -C
alvarov@alvarov - 11:43:45 - Mon Dec 13 $ sudo add-apt-repository
Error: need a repository as argument
alvarov@alvarov - 11:43:46 - Mon Dec 13 $ "deb [arch=amd64] https://download.docker.com/linux/ubuntu
$ {lsb_release -cs} stable"
-bash: deb [arch=amd64] https://download.docker.com/linux/ubuntu
focal stable: No such file or directory
alvarov@alvarov - 11:43:46 - Mon Dec 13 $ sudo add-apt-repository
Error: need a repository as argument
alvarov@alvarov - 11:43:49 - Mon Dec 13 $ "deb [arch=amd64] https://download.docker.com/linux/ubuntu
$ {lsb_release -cs} stable"
-bash: deb [arch=amd64] https://download.docker.com/linux/ubuntu
focal stable: No such file or directory
alvarov@alvarov - 11:43:49 - Mon Dec 13 $ "C
alvarov@alvarov - 11:43:50 - Mon Dec 13 $ sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu ${lsb_release -cs} stable"
sudo: add-apt: command not found
alvarov@alvarov - 11:44:38 - Mon Dec 13 $ sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu ${lsb_release -cs} stable"
Get:1 https://download.docker.com/linux/ubuntu focal InRelease [57.7 kB]
Hit:2 http://es.archive.ubuntu.com/ubuntu focal InRelease
Hit:3 http://es.archive.ubuntu.com/ubuntu focal-updates InRelease
Hit:4 http://es.archive.ubuntu.com/ubuntu focal-backports InRelease
Hit:5 http://es.archive.ubuntu.com/ubuntu focal-security InRelease
Hit:6 http://repo.zabbix.com/zabbix/5.0/ubuntu focal InRelease
Get:7 https://download.docker.com/linux/ubuntu focal/stable amd64 Packages [12.9 kB]
Get:8 https://download.docker.com/linux/ubuntu focal/stable amd64 Contents (deb) [1343 B]
Fetched 72.0 kB in 2s (39.6 kB/s)
Reading package lists... Done
alvarov@alvarov - 11:44:50 - Mon Dec 13 $
```

Actualizamos la lista de repositorios, buscamos repositorio de docker y lo instalamos:

```
alvaroVR@alvaro - 11:46:18 - Mon Dec 13 $ 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
Hit:4 http://es.archive.ubuntu.com/ubuntu focal-backports InRelease
Hit:5 http://es.archive.ubuntu.com/ubuntu focal-security InRelease
Hit:6 http://repo.zabbix.com/zabbix/5.0/ubuntu focal InRelease
Reading package lists... Done
Building dependency tree
Reading state information... Done
All packages are up to date.
alvaroVR@alvaro - 11:46:27 - Mon Dec 13 $ apt search docker-ce
Sorting... Done
Full Text Search... Done
docker-ce/focal 5:20.10.11~3-0~ubuntu-focal amd64
  Docker: the open-source application container engine

docker-ce-cli/focal 5:20.10.11~3-0~ubuntu-focal amd64
  Docker CLI: the open-source application container engine

docker-ce-rootless-extras/focal 5:20.10.11~3-0~ubuntu-focal amd64
  Rootless support for Docker.

alvaroVR@alvaro - 11:46:42 - Mon Dec 13 $ sudo apt install docker-ce
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  containerd.io docker-ce-cli docker-ce-rootless-extras docker-scan-plugin pigz slurp4netns
Suggested packages:
  aufs-tools cgroupfs-mount | cgroup-lite
The following NEW packages will be installed:
  containerd.io docker-ce docker-ce-cli docker-ce-rootless-extras docker-scan-plugin pigz slurp4netns
```

Comprobamos que el servicio está iniciado y añadimos mi usuario (alvaro) al grupo docker:

```
alvaroVR@alvaro - 11:52:31 - Mon Dec 13 $ systemctl status docker
* docker.service - Docker Application Container Engine
   Loaded: loaded (/lib/systemd/system/docker.service; enabled; vendor preset: enabled)
   Active: active (running) since Mon 2021-12-13 11:47:36 UTC; 5min ago
     TriggeredBy: ● docker.socket
       Docs: https://docs.docker.com
    Main PID: 60854 (dockerd)
       Tasks: 7
         Memory: 28.5M
        CGroup: /system.slice/docker.service
               └─60854 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock

Dec 13 11:47:35 alvaro dockerd[60854]: time="2021-12-13T11:47:35.112378101Z" level=warning msg="Your kernel does not support CPU realtime scheduler"
Dec 13 11:47:35 alvaro dockerd[60854]: time="2021-12-13T11:47:35.112544225Z" level=warning msg="Your kernel does not support cgroup blkio weight"
Dec 13 11:47:35 alvaro dockerd[60854]: time="2021-12-13T11:47:35.112608089Z" level=warning msg="Your kernel does not support cgroup blkio weight_device"
Dec 13 11:47:35 alvaro dockerd[60854]: time="2021-12-13T11:47:35.113050562Z" level=info msg="Loading containers: start."
Dec 13 11:47:35 alvaro dockerd[60854]: time="2021-12-13T11:47:35.582821802Z" level=info msg="Default bridge (docker0) is assigned with an IP address 172.17.0.8/16. Daemon option --bip can be used to set a preferred IP address"
Dec 13 11:47:35 alvaro dockerd[60854]: time="2021-12-13T11:47:35.987912730Z" level=info msg="Loading containers: done."
Dec 13 11:47:36 alvaro dockerd[60854]: time="2021-12-13T11:47:36.270405035Z" level=info msg="Docker daemon" commit=847da18 graphdriver(s)=overlay2 version=20.10.11
Dec 13 11:47:36 alvaro dockerd[60854]: time="2021-12-13T11:47:36.271103034Z" level=info msg="Daemon has completed initialization"
Dec 13 11:47:36 alvaro systemd[1]: Started Docker Application Container Engine.
Dec 13 11:47:36 alvaro dockerd[60854]: time="2021-12-13T11:47:36.394624474Z" level=info msg="API listen on /run/docker.sock"
alvaroVR@alvaro - 11:52:42 - Mon Dec 13 $ sudo usermod -sG docker su_nombre_de_usu
usermod: user "su_nombre_de_usu" does not exist
alvaroVR@alvaro - 11:53:08 - Mon Dec 13 $ sudo usermod -sG docker alvaro
```

Probamos un comando para ver si funciona correctamente docker:

@Zukii on Wuolah

```
alvaroVR@alvaro - 11:55:26 - Mon Dec 13 $ docker run hello-world
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
2db29710123e: Pull complete
Digest: sha256:cc15c5b292d8525effc0f89cb299f1804f3a725c8d05e158653a563f15e4f685
Status: Downloaded newer image for hello-world:latest

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/

alvaroVR@alvaro - 11:55:40 - Mon Dec 13 $
```

Y con esto damos por concluida la instalación de Docker. Ahora seguimos con docker-compose.

Instalamos docker-compose y probamos que funciona:

```
alvaro@alvaro - 11:55:40 - Mon Dec 13 $ sudo apt install docker-compose
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  python3-cached-property python3-docker python3-dockerpty python3-dockerpty python3-dockerpty python3-texttable python3-websocket
Recommended packages:
  docker.io
The following NEW packages will be installed:
  docker-compose python3-cached-property python3-docker python3-dockerpty python3-dockerpty python3-dockerpty python3-texttable python3-websocket
0 upgraded, 7 newly installed, 0 to remove and 0 not upgraded.
Need to get 262 kB of archives.
After this operation, 1616 kB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://es.archive.ubuntu.com/ubuntu focal/universe amd64 python3-cached-property all 1.5.1-4 [10.9 kB]
Get:2 http://es.archive.ubuntu.com/ubuntu focal/universe amd64 python3-websocket all 0.53.0-2ubuntu1 [32.3 kB]
Get:3 http://es.archive.ubuntu.com/ubuntu focal/universe amd64 python3-docker all 4.1.0-1 [83.8 kB]
Get:4 http://es.archive.ubuntu.com/ubuntu focal/universe amd64 python3-dockerpty all 0.4.1-2 [11.1 kB]
Get:5 http://es.archive.ubuntu.com/ubuntu focal/universe amd64 python3-dockerpty all 0.6.2-2.2ubuntu1 [19.7 kB]
Get:6 http://es.archive.ubuntu.com/ubuntu focal/universe amd64 python3-texttable all 1.6.2-2 [11.0 kB]
Get:7 http://es.archive.ubuntu.com/ubuntu focal/universe amd64 docker-compose all 1.25.0-1 [92.7 kB]
Fetched 262 kB in 1s (210 kB/s)
Selecting previously unselected package python3-cached-property.
(Reading database ... 119293 files and directories currently installed.)
Preparing to unpack .../0-python3-cached-property_1.5.1-4_all.deb ...
Unpacking python3-cached-property (1.5.1-4) ...
Selecting previously unselected package python3-websocket.
Preparing to unpack .../1-python3-websocket_0.53.0-2ubuntu1_all.deb ...
Unpacking python3-websocket (0.53.0-2ubuntu1) ...
Selecting previously unselected package python3-docker.
Preparing to unpack .../2-python3-docker_4.1.0-1_all.deb ...
Unpacking python3-docker (4.1.0-1) ...
Selecting previously unselected package python3-dockerpty.
Preparing to unpack .../3-python3-dockerpty_0.4.1-2_all.deb ...
Unpacking python3-dockerpty (0.4.1-2) ...
Selecting previously unselected package python3-dockerpty.
Preparing to unpack .../4-python3-dockerpty_0.6.2-2.2ubuntu1_all.deb ...
Unpacking python3-dockerpty (0.6.2-2.2ubuntu1) ...
Selecting previously unselected package python3-texttable.
Preparing to unpack .../5-python3-texttable_1.6.2-2_all.deb ...
Unpacking python3-texttable (1.6.2-2) ...
Selecting previously unselected package docker-compose.
Preparing to unpack .../6-docker-compose_1.25.0-1_all.deb ...
Unpacking docker-compose (1.25.0-1) ...
Setting up python3-cached-property (1.5.1-4) ...
Setting up python3-websocket (0.53.0-2ubuntu1) ...
Setting up python3-docker (4.1.0-1) ...
Setting up python3-dockerpty (0.4.1-2) ...
Setting up python3-dockerpty (0.6.2-2.2ubuntu1) ...
Setting up python3-texttable (1.6.2-2) ...
Setting up docker-compose (1.25.0-1) ...
```

```
alvaroVR@alvaro - 12:12:10 - Mon Dec 13 $ docker-compose --version
docker-compose version 1.25.0, build unknown
alvaroVR@alvaro - 12:12:14 - Mon Dec 13 $ docker-compose
Define and run multi-container applications with Docker.

Usage:
  docker-compose [-f <arg>...] [options] [COMMAND] [ARGS...]
  docker-compose -h|--help

Options:
  -f, --file FILE             Specify an alternate compose file
                              (default: docker-compose.yml)
  -p, --project-name NAME     Specify an alternate project name
                              (default: directory name)
  --verbose                   Show more output
  --log-level LEVEL           Set log level (DEBUG, INFO, WARNING, ERROR, CRITICAL)
  --no-ansi                   Do not print ANSI control characters
  -v, --version               Print version and exit
  -H, --host HOST             Daemon socket to connect to

  --tls                       Use TLS; implied by --tlsverify
  --tlscacert CA_PATH         Trust certs signed only by this CA
  --tlscert CLIENT_CERT_PATH  Path to TLS certificate file
  --tlskey TLS_KEY_PATH       Path to TLS key file
  --tlsverify                 Use TLS and verify the remote
```

Y por último instalamos la aplicación para el test de JMeter mediante git clone:

```
alvaroVR@alvaro - 12:12:20 - Mon Dec 13 $ git clone https://github.com/davidPalomar-ugr/iseP4JMeter.git
Cloning into 'iseP4JMeter'...
remote: Enumerating objects: 3797, done.
remote: Counting objects: 100% (23/23), done.
remote: Compressing objects: 100% (14/14), done.
remote: Total 3797 (delta 9), reused 15 (delta 7), pack-reused 3774
Receiving objects: 100% (3797/3797), 7.79 MiB | 2.81 MiB/s, done.
Resolving deltas: 100% (715/715), done.
alvaroVR@alvaro - 12:15:44 - Mon Dec 13 $ cd iseP4JMeter
alvaroVR@alvaro - 12:17:11 - Mon Dec 13 $ pwd
/home/alvaro/iseP4JMeter
```

Luego ponemos: “> *sudo docker-compose up*” y nos da dos problemas.

Primeramente, nos da un problema de espacio. Vemos con “>*df -h /var*” como tenemos /var lleno y esto causa problemas. Lo que haremos será añadir otro disco, crear un nuevo volumen lógico y mover /var a este como hicimos en la práctica 1. (No pongo las instrucciones pero son muy sencillas).

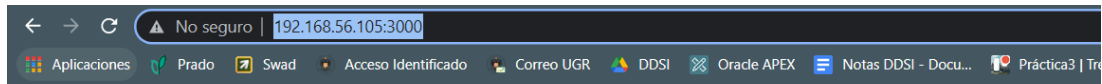
Arreglado esto, un problema de abortado por instrucciones ilegales además de un aviso de incompatibilidad con el procesador:

```
alvaroVR@alvaro - 18:17:04 - Mon Dec 13 $ docker-compose up
Starting isep4jmeter_mongodb_1 ... done
Starting isep4jmeter_mongodbinit_1 ... done
Starting isep4jmeter_nodejs_1 ... done
Attaching to isep4jmeter_mongodb_1, isep4jmeter_mongodbinit_1, isep4jmeter_nodejs_1
mongodb_1
mongodb_1 | WARNING: MongoDB 5.0+ requires a CPU with AVX support, and your current system does not appear to have that!
mongodb_1 | see https://jira.mongodb.org/browse/SERVER-54407
mongodb_1 | see also https://www.mongodb.com/community/forums/t/mongodb-5-0-cpu-intel-g4650-compatibility/116610/2
mongodb_1 | see also https://github.com/docker-library/mongo/issues/485#issuecomment-891991814
mongodbinit_1
mongodbinit_1 | WARNING: MongoDB 5.0+ requires a CPU with AVX support, and your current system does not appear to have that!
mongodbinit_1 | see https://jira.mongodb.org/browse/SERVER-54407
mongodbinit_1 | see also https://www.mongodb.com/community/forums/t/mongodb-5-0-cpu-intel-g4650-compatibility/116610/2
mongodbinit_1 | see also https://github.com/docker-library/mongo/issues/485#issuecomment-891991814
isep4jmeter_mongodb_1 exited with code 132
mongodbinit_1 | Illegal instruction (core dumped)
mongodbinit_1 | Illegal instruction (core dumped)
nodejs_1
nodejs_1 | > etsii@1.0.0 start
nodejs_1 | > node ./bin/www
```

[illegible][illegible]

```
alvaroVR@alvaro - 21:38:18 - Mon Dec 13 $ sudo ufw allow 3000/tcp
Rule updated
Rule updated (v6)
alvaroVR@alvaro - 21:38:22 - Mon Dec 13 $ _
```

Podemos observar la URL:



ETSII Alumnos API

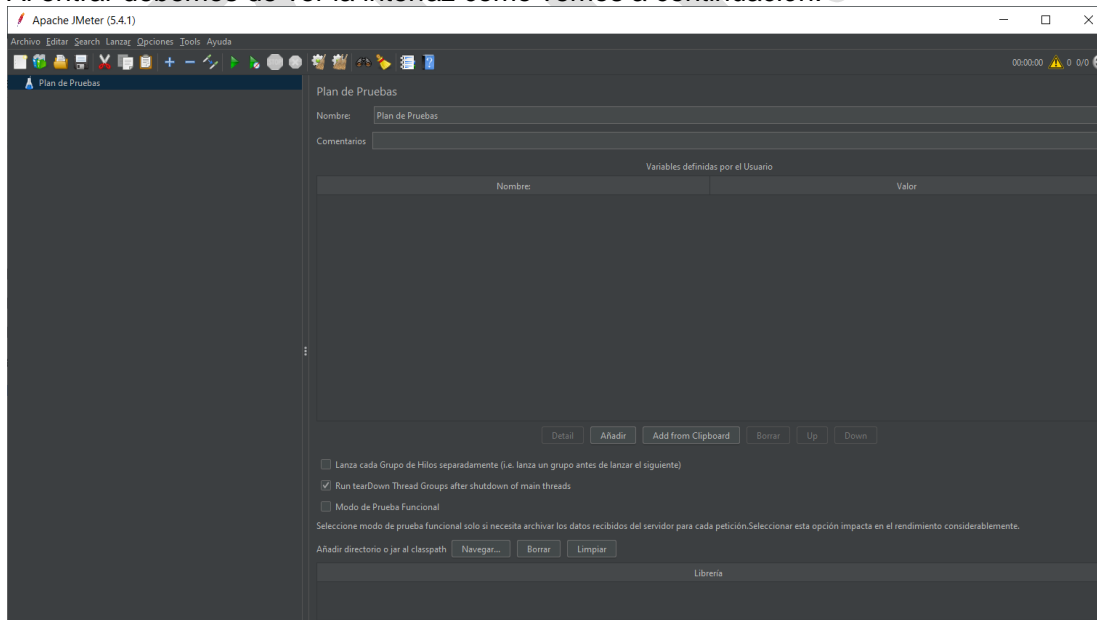
Descripción de la API Restful:

- POST /api/v1/auth/login
 - Parametros:
 - login:<emailUsuario>
 - password:<secreto>
 - Seguridad:
 - Acceso protegido con BasicAuth (etsiiApi:laApiDeLaETSIIIDLache)
 - 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 cualquier alumno válido
 - Retorna:
 - Objeto json con perfil de alumno

2.2) Instalación de JMeter en el Sistema Anfitrión (W10)

Para la instalación accedemos a la página de JMeter y luego a la sección de descargas (https://jmeter.apache.org/download_jmeter.cgi). Aquí, descargamos un binario comprimido (en mi caso el .zip). Luego, descomprimos y dentro de la carpeta “bin” tenemos la aplicación “ApacheJMeter” que debemos abrirla con Java.

Al entrar debemos de ver la interfaz como vemos a continuación:



2.3) Creación del Test de JMeter

(Como al entrar por primera vez tenemos un test sin nada, no hace falta crear otro test)

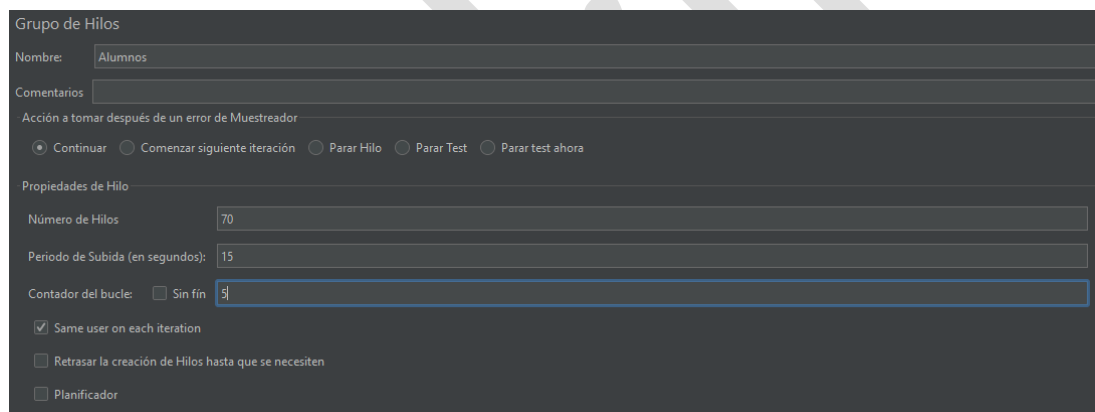
Lo primero que haremos será **parametrizar el host y el puerto** tal y como dice el enunciado. Esto lo hacemos en el TestPlan, dándole a *añadir* y *rellenando los dos campos*. También cambiaremos el nombre al plan y quedaría así:



Nombre	Valor
HOST	192.168.56.105
PORT	3000

Luego haremos la parte de **crear dos grupos de hebras distintos** que simulan a los alumnos y los administradores. Para esto, damos *click derecho sobre plan de pruebas* -> *Añadir* -> *Hilos* -> *Grupos de Hilos* y ya al crear el grupo modificamos el nombre.

También modificamos el número de hilos (usuarios), el período de subida (lo que tarda en alcanzar el máximo número de peticiones) y modificamos las iteraciones del bucle:



Nombre: Alumnos

Comentarios:

Acción a tomar después de un error de Muestreador:

☒ Continuar ☐ Comenzar siguiente iteración ☐ Parar Hilo ☐ Parar Test ☐ Parar test ahora

Propiedades de Hilo:

Número de Hilos: 70

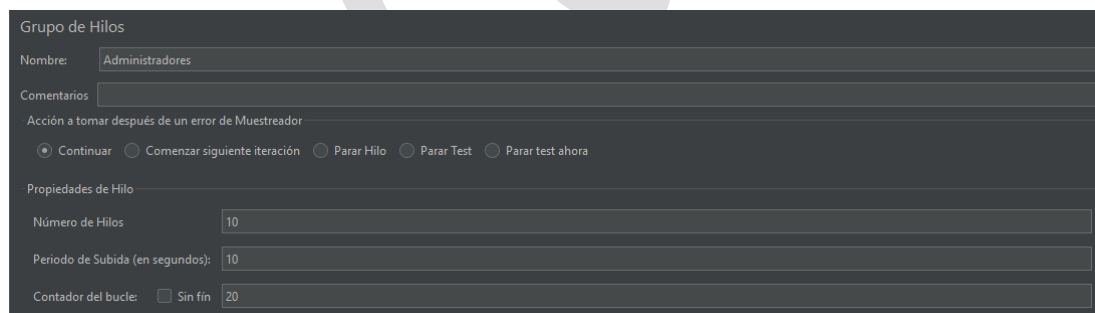
Periodo de Subida (en segundos): 15

Contador del bucle: ☐ Sin fin 5

☒ Same user on each iteration

☐ Retrasar la creación de Hilos hasta que se necesiten

☐ Planificador



Nombre: Administradores

Comentarios:

Acción a tomar después de un error de Muestreador:

☒ Continuar ☐ Comenzar siguiente iteración ☐ Parar Hilo ☐ Parar Test ☐ Parar test ahora

Propiedades de Hilo:

Número de Hilos: 10

Periodo de Subida (en segundos): 10

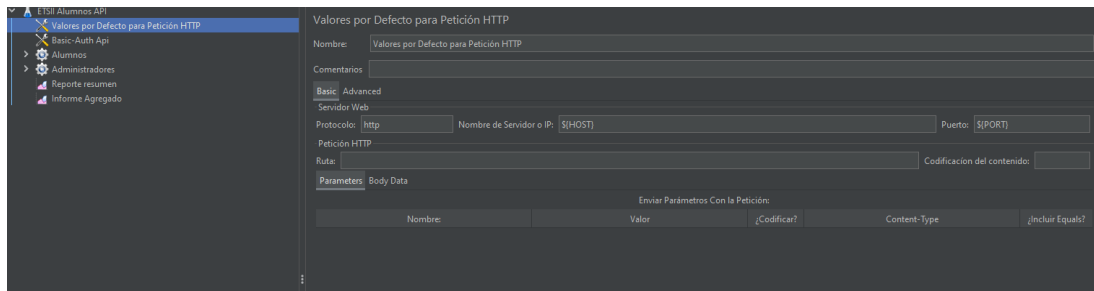
Contador del bucle: ☐ Sin fin 20

☒ Same user on each iteration

☐ Retrasar la creación de Hilos hasta que se necesiten

☐ Planificador

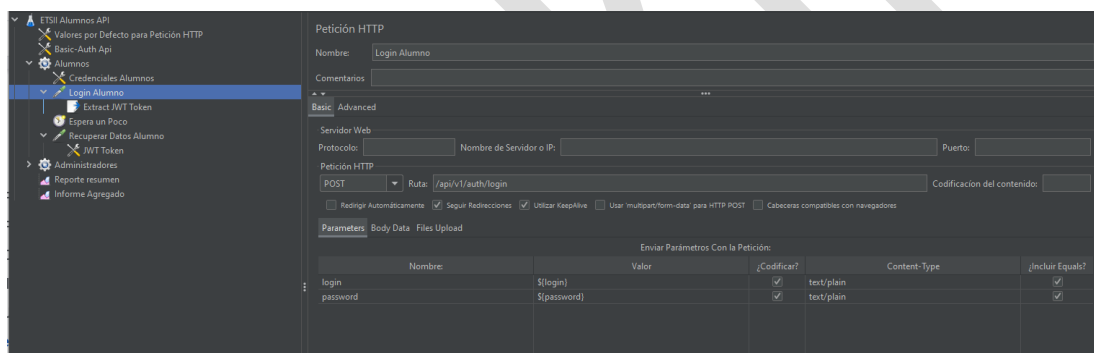
Lo siguiente que haremos será poner unos **valores por defecto a las peticiones http**. Hay que darle al *click derecho sobre ETSII Alumnos API* -> *Añadir* -> *Elemento de Configuración* -> *Valores por defecto para petición HTTP*. Esto lo ponemos encima de los dos grupos de hebras porque es algo común a ambas. Dentro de los valores por defecto, modificamos el campo de puerto al parámetro \$(PORT) (definido al principio), la IP del host a \$(HOST) y que las peticiones se harán con el protocolo http:



Ahora, simularemos las **peticiones HTTP de acceso de los alumnos**. Para esto, crearemos una petición HTTP. Esto lo hacemos haciendo *click derecho sobre Alumnos -> Añadir -> Muestreador -> Petición HTTP*.

Haremos la petición a la ruta `/api/v1/auth/login`, de tipo POST (como vimos con la URL `192.168.56.105:3000`), y pasamos los parámetros de usuario y contraseña (codificados) para el acceso (Posteriormente definidos).

Luego, en estas peticiones, modificaremos los parámetros. Los usuarios deberían quedar tal que así:

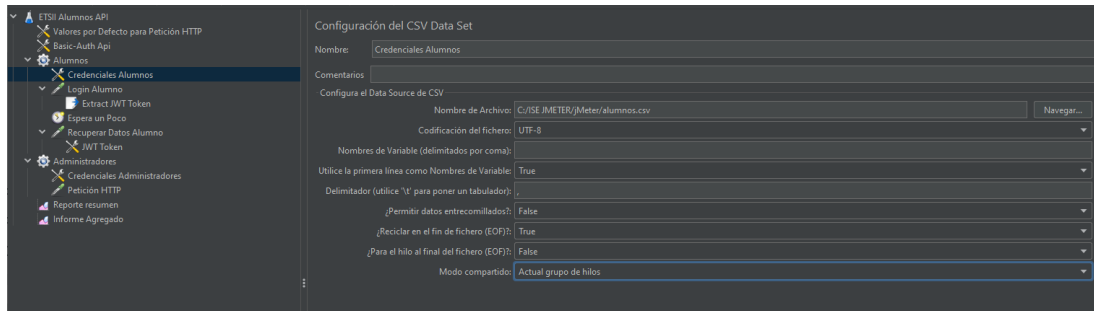


(El protocolo, el host y el puerto ya lo hemos especificado en “Valores por defecto de HTTP”)

Pero ojo, previamente, deberemos haber leído del archivo `alumnos.csv` para darle valor a las variables de login y password. Esto lo hacemos haciendo *click derecho sobre Login Alumnos -> Añadir -> Elemento de Configuración -> Configuración del CSV Data Set y lo ponemos encima de “Login Alumnos”*.

Como vamos a requerir tener el archivo `.csv` en nuestro sistema anfitrión, que es donde tenemos JMeter, lo descargamos de github.

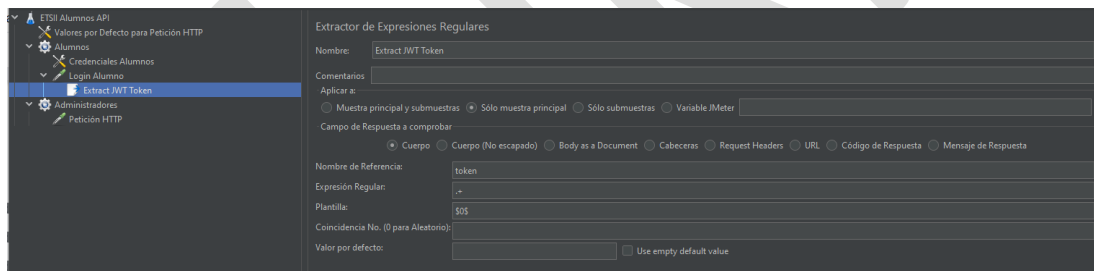
Una vez descargado y con una ruta definida, ponemos la ruta y modificamos unos cuantos parámetros de la configuración como activar que la primera línea tiene el nombre de las variables (login y password para asignar los valores a `$(login)` y `$(password)`).



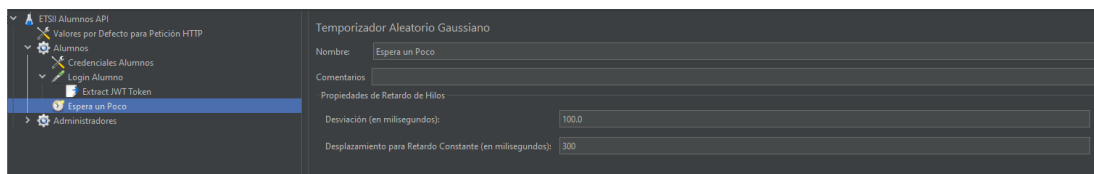
Si la petición POST que hemos puesto antes ha ido correctamente, la API nos devolverá el **token JWT** del usuario que hizo la petición. Este token viene en la respuesta a la petición por lo que tenemos que comprobar mediante expresiones regulares si es un token JWT. Lo necesitaremos para realizar peticiones de tipo GET, por lo que tendremos que guardar.

Haremos *click derecho sobre Login Alumnos* -> *Añadir* -> *PostProcesadores* -> *Extractor de Expresiones Regulares*. Y tendremos que modificar los campos de “Nombre de Referencia”, “Expresión Regular” y “Plantilla” que hacen referencia al nombre de la variable, a la expresión con la que extraemos y lo que guardamos en la variable respectivamente.

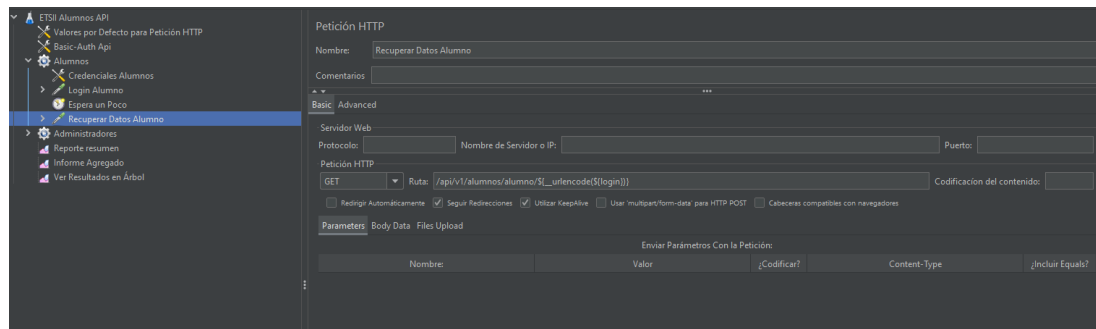
Así, nos debe salir tal y como vemos en la siguiente imagen:



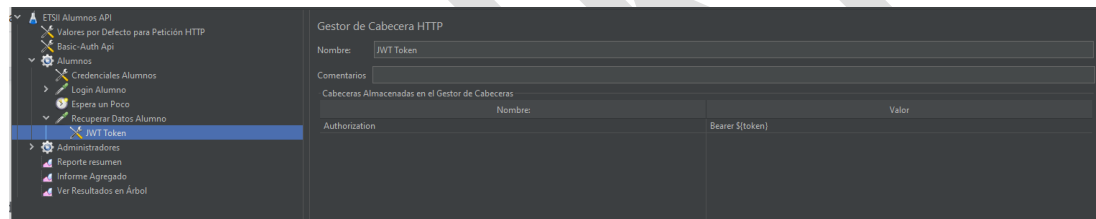
Ahora, siguiendo el esquema dado, ponemos una **pausa aleatoria Gaussiana**. Se hace haciendo *click derecho en Alumnos* -> *Añadir* -> *Temporizador* -> *Temporizador Aleatorio Gaussiano*. Podemos modificar los parámetros (los ms) para aparentar el comportamiento real de los usuarios, simulando pausas aleatorias desde 0ms a los ms especificados en los parámetros, pero en mi caso, lo dejo por defecto. Algo así deberíamos acabar teniendo:



Tras habernos autenticado con la primera petición http (Login Alumno), ahora haremos una **petición pero de tipo GET**. Con esta, recuperaremos datos del alumno (el mismo que se “loggeo”). Las diferencias serán el tipo y la ruta (/api/v1/auth/\${login}). Hacemos *click derecho sobre ETSII Alumnos API -> Añadir -> Muestreador -> Petición HTTP* y debemos observar:

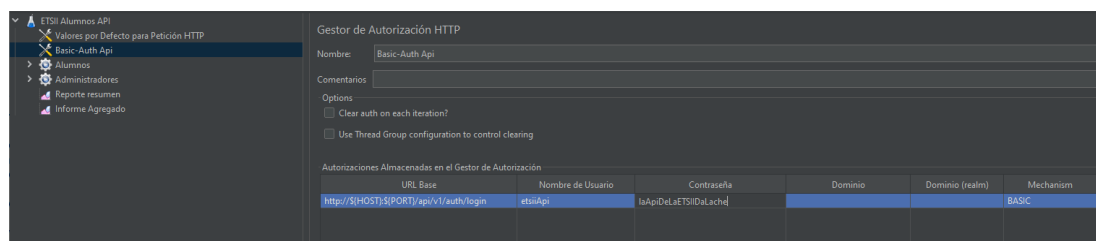


Luego, y por último respecto a los alumnos, como es necesario autenticarse para realizar una petición de tipo GET, crearemos un **Gestor de Cabeceras HTTP** que usará el **token JWT** conseguido previamente **para autenticarnos**. En este añadiremos la cabecera “Authorization” y como valor Bearer \${token}. Lo conseguimos haciendo *click derecho sobre Alumnos -> Añadir -> Elementos de Configuración -> Gestor de cabecera HTTP* (lo colocamos debajo de la petición HTTP GET) y finalmente, debemos ver algo que así:



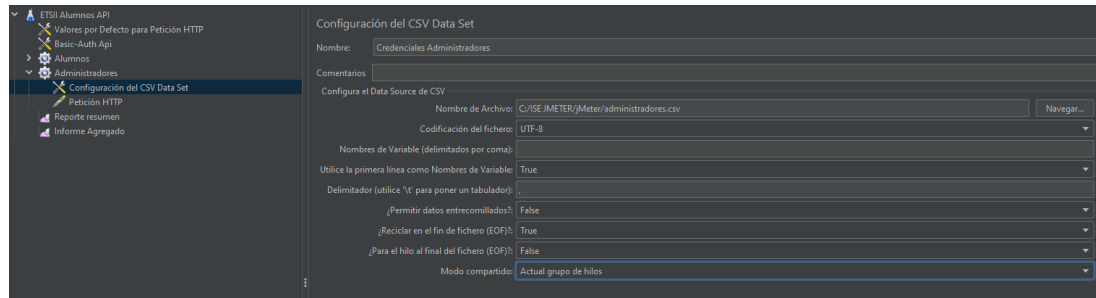
Y con esto podemos dar por terminada la parte de los alumnos.

Ahora, procederemos con los administradores, pero previo a esto, realizaremos la **autorización para acceder a la API**. Esto será común y previo a la parte de alumnos y administradores. Añadiremos un gestor de autorización http haciendo *click derecho sobre ETSII Alumnos API -> Añadir -> Elementos de Configuración -> Gestor de Autorización HTTP*. Debe quedar tal que así:

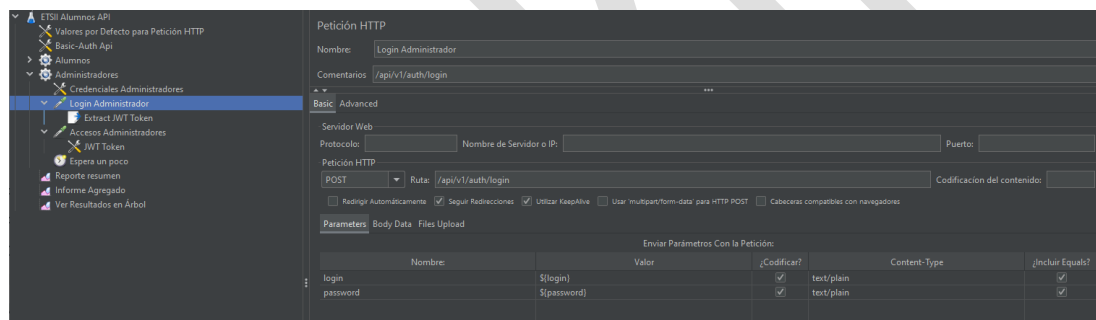


Y ahora sí, volvemos con los **administradores**. Estos serán parecidos a los alumnos, es por ello, que seré más breve en las cosas que ya hayan aparecido y haya comentado con los alumnos.

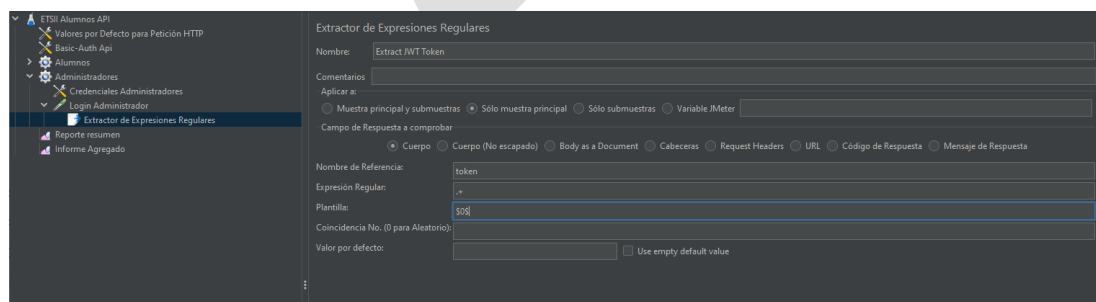
Empezaremos con el **ingreso de las credenciales de los administradores**. Esto lo haremos igual que como hicimos antes, es decir, leyendo de un archivo .csv (administradores.csv). Haremos *click derecho sobre Administradores -> Añadir -> Elemento de Configuración -> Configuración del CSV Data Set*. Así, debe salir:



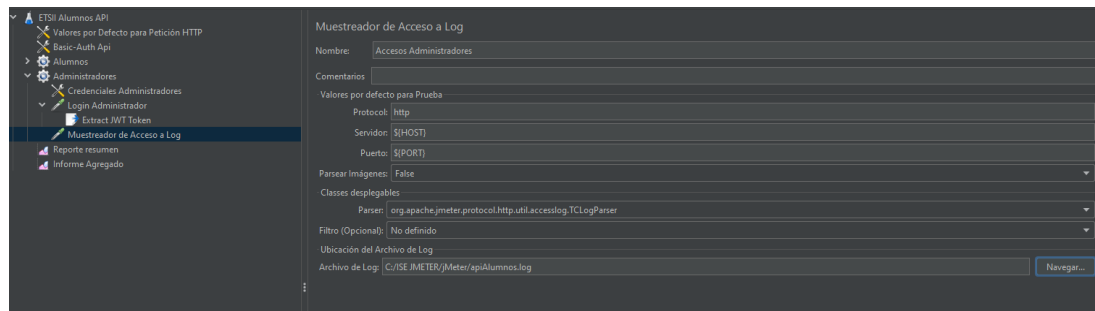
Luego hacemos el **login de los administradores**. Para esto hacemos *click derecho sobre Administradores -> Añadir -> Muestreador -> Petición HTTP*. Y nos quedaría prácticamente igual que los alumnos:



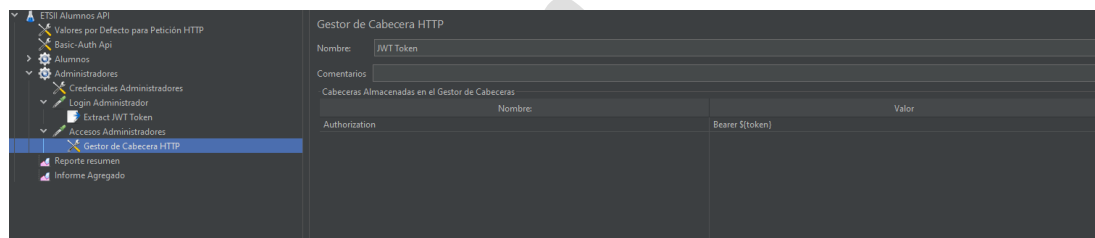
Hecho esto, deberemos **obtener el Token JWT** que se recibe tras el login. Hay que hacer *click derecho sobre Login Administradores -> Añadir -> PostProcesadores -> Extractor de Expresiones Regulares*. El resultado debe ser:



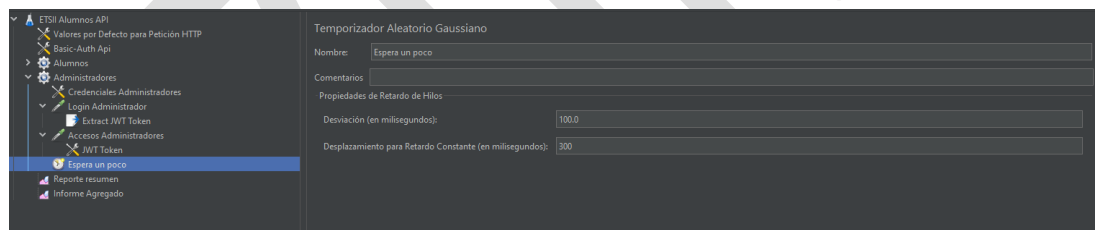
Luego hacemos el **Acceso de los Administradores** para obtener los datos de diversos alumnos. Esto lo hacemos con un muestreador de acceso a log, que lo añadimos haciendo *click derecho sobre Administradores -> Añadir -> Muestreador -> Muestreador de Acceso a log*. Esto lo tenemos que dejar tal que así:



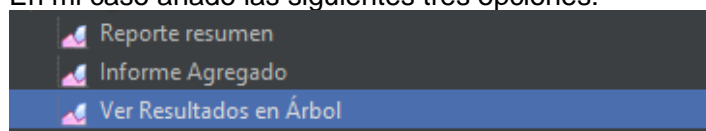
Al igual que con los alumnos, ahora añadimos el gestor de cabecera que **usará el token JWT** que teníamos de antes. Lo añadimos haciendo *click derecho sobre Administradores -> Añadir -> Elementos de Configuración -> Gestor de cabecera HTTP*. El resultado sería:



Luego añadimos una **espera aleatoria** (*click derecho en Administrador -> Añadir -> Temporizador -> Temporizador Aleatorio Gaussiano*):



Y por último, añadimos lo necesario para poder visualizar los resultados. Esto lo hacemos haciendo *click derecho en ETSII Alumnos API -> Añadir -> Receptor -> ---*. En mi caso añado las siguientes tres opciones:



2.4) Prueba del Test de JMeter

Así, si probamos a ejecutar el test, y visualizar los resultados podremos ver:

Archivo Editar Search Lanzar Opciones Tools Ayuda

00:00:08 0 25/80

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 Mue...	Estado	Bytes	Sent Bytes	Latency	Connect Time(ms)
1	13:31:17.558	Alumnos 1-1	Login Alumno	16	✓	605	345	15	5
2	13:31:17.591	Administradores...	Login Administr...	11	✓	618	327	11	3
3	13:31:17.636	Alumnos 1-2	Login Alumno	11	✓	605	333	11	3
4	13:31:17.758	Administradores...	http://192.168.56...	10	✓	1207	0	10	0
5	13:31:17.865	Alumnos 1-1	Recuperar Datos ...	12	✓	1170	382	11	0
6	13:31:17.869	Alumnos 1-3	Login Alumno	21	✓	605	336	21	4
7	13:31:17.892	Alumnos 1-2	Recuperar Datos ...	9	✓	1603	377	9	0
8	13:31:17.933	Alumnos 1-4	Login Alumno	18	✓	605	336	18	6
9	13:31:18.039	Administradores...	Login Administr...	18	✓	618	339	18	0
10	13:31:18.142	Alumnos 1-1	Login Alumno	11	✓	606	336	11	0
11	13:31:18.145	Alumnos 1-4	Recupera Datos ...	16	✓	1111	379	16	0
12	13:31:18.152	Administradores...	http://192.168.56...	19	✓	1305	0	19	0
13	13:31:18.262	Alumnos 1-5	Login Alumno	10	✓	606	338	10	3
14	13:31:18.311	Alumnos 1-3	Recuperar Datos ...	15	✓	2072	379	15	0
15	13:31:18.388	Alumnos 1-4	Login Alumno	13	✓	606	337	13	0
16	13:31:18.394	Administradores...	Login Administr...	24	✓	618	338	24	2
17	13:31:18.451	Alumnos 1-2	Login Alumno	10	✓	606	332	9	0
18	13:31:18.467	Alumnos 1-1	Recuperar Datos ...	15	✓	1286	377	15	0
19	13:31:18.481	Administradores...	Login Administr...	8	✓	618	339	8	0
20	13:31:18.550	Alumnos 1-6	Login Alumno	16	✓	606	335	15	2
21	13:31:18.597	Alumnos 1-7	Login Alumno	9	✓	606	339	9	2
22	13:31:18.641	Administradores...	http://192.168.56...	22	✓	1605	0	22	0
23	13:31:18.646	Alumnos 1-5	Recuperar Datos ...	27	✓	1540	381	27	0
24	13:31:18.659	Alumnos 1-1	Login Alumno	18	✓	606	336	18	0
25	13:31:18.688	Alumnos 1-2	Recuperar Datos ...	9	✓	1659	377	9	0
26	13:31:18.697	Alumnos 1-4	Recuperar Datos ...	9	✓	1600	380	9	0

☐ Scroll automatically? ☐ Child samples? No. de Muestras: 442 Última Muestra: 9 **Media: 15** **Desviación: 11**

Archivo Editar Search Lanzar Opciones Tools Ayuda

00:00:22 0 0/80

Reporte resumen

Nombre: Reporte resumen

Comentarios:

Escribir todos los datos a Archivo

Nombre de archivo: Navegar... Log/Mostrar sólo: Escribir en Log Sólo Errores Éxitos Configurar

Etiqueta	# Muestras	Media	Min	Max	Desv. Estándar	% Error	Rendimiento	Kb/sec	Sent KB/sec	Media de Bytes
Login Alumno	375	22	6	1064	55.74	0.00%	6.2/min	0.06	0.03	606.6
Recuperar Datos Alumno	368	20	5	112	17.07	0.00%	6.1/min	0.15	0.04	1515.4
Login Administrador	206	22	6	1090	75.72	0.00%	3.4/min	0.03	0.02	618.0
http://192.168.56.105:30...	12	26	8	108	27.53	0.00%	12.0/hour	0.00	0.00	1207.0
http://192.168.56.105:30...	12	18	8	45	10.56	0.00%	12.0/hour	0.00	0.00	1305.0
http://192.168.56.105:30...	10	37	8	110	31.61	0.00%	1.1/sec	1.73	0.00	1605.6
http://192.168.56.105:30...	10	23	9	56	14.63	0.00%	1.1/sec	2.19	0.00	2044.9
http://192.168.56.105:30...	10	47	11	120	34.51	0.00%	1.1/sec	1.20	0.00	1142.0
http://192.168.56.105:30...	10	27	10	49	12.57	0.00%	1.1/sec	1.66	0.00	1575.0
http://192.168.56.105:30...	10	22	8	43	11.65	0.00%	1.0/sec	1.27	0.00	1258.0
http://192.168.56.105:30...	10	27	13	60	13.80	0.00%	59.4/min	1.05	0.00	1082.0
http://192.168.56.105:30...	10	28	9	91	23.25	0.00%	1.0/sec	1.50	0.00	1512.0
http://192.168.56.105:30...	10	29	11	62	13.69	0.00%	1.0/sec	1.08	0.00	1083.0
http://192.168.56.105:30...	10	42	8	110	33.11	0.00%	1.0/sec	1.15	0.00	1131.0
http://192.168.56.105:30...	10	29	8	114	33.00	0.00%	1.1/sec	1.70	0.00	1631.0
http://192.168.56.105:30...	10	36	8	80	27.31	0.00%	1.0/sec	1.24	0.00	1219.0
http://192.168.56.105:30...	10	29	8	72	22.41	0.00%	1.0/sec	2.17	0.00	2175.0
http://192.168.56.105:30...	10	19	8	46	12.07	0.00%	1.0/sec	1.23	0.00	1242.0
http://192.168.56.105:30...	10	27	7	55	16.87	0.00%	1.0/sec	1.51	0.00	1512.0
http://192.168.56.105:30...	10	20	10	48	11.71	0.00%	58.3/min	1.01	0.00	1063.0
http://192.168.56.105:30...	10	16	8	47	11.20	0.00%	1.0/sec	1.29	0.00	1308.0
http://192.168.56.105:30...	10	12	8	21	3.44	0.00%	1.0/sec	1.06	0.00	1071.0
http://192.168.56.105:30...	10	12	9	25	4.96	0.00%	1.0/sec	1.69	0.00	1691.0
Total	1153	22	5	1090	47.14	0.00%	19.2/min	0.32	0.09	1037.3

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

Y en la máquina ubuntu podemos ver las peticiones que realiza JMeter:

```
nodejs_1 | GET /api/v1/alumnos/alumno/blankenshipchapman%40tropoli.com 200 5.621 ms - 1288
nodejs_1 | POST /api/v1/auth/login 200 5.269 ms - 196
nodejs_1 | POST /api/v1/auth/login 200 5.198 ms - 196
nodejs_1 | POST /api/v1/auth/login 200 5.864 ms - 196
nodejs_1 | GET /api/v1/alumnos/alumno/blankenshipchapman%40tropoli.com 200 5.547 ms - 1288
nodejs_1 | POST /api/v1/auth/login 200 7.155 ms - 196
nodejs_1 | GET /api/v1/alumnos/alumno/janieking%40tropoli.com 200 6.041 ms - 840
nodejs_1 | GET /api/v1/alumnos/alumno/woodardoneill%40tropoli.com 200 4.917 ms - 1109
nodejs_1 | GET /api/v1/alumnos/alumno/bentleysharpe%40tropoli.com 200 5.061 ms - 906
nodejs_1 | POST /api/v1/auth/login 200 5.554 ms - 196
nodejs_1 | POST /api/v1/auth/login 200 6.110 ms - 196
nodejs_1 | POST /api/v1/auth/login 200 5.593 ms - 196
nodejs_1 | GET /api/v1/alumnos/alumno/woodardoneill%40tropoli.com 200 5.539 ms - 1109
nodejs_1 | GET /api/v1/alumnos/alumno/kristinearnold%40tropoli.com 200 6.004 ms - 661
nodejs_1 | GET /api/v1/alumnos/alumno/staceytownsend%40tropoli.com 200 6.710 ms - 669
nodejs_1 | POST /api/v1/auth/login 200 5.616 ms - 196
nodejs_1 | POST /api/v1/auth/login 200 7.731 ms - 196
nodejs_1 | POST /api/v1/auth/login 200 7.146 ms - 196
nodejs_1 | GET /api/v1/alumnos/alumno/blankenshipchapman%40tropoli.com 200 5.168 ms - 1288
nodejs_1 | GET /api/v1/alumnos/alumno/bentleysharpe%40tropoli.com 200 5.956 ms - 906
nodejs_1 | GET /api/v1/alumnos/alumno/kristinearnold%40tropoli.com 200 5.952 ms - 661
nodejs_1 | POST /api/v1/auth/login 200 8.982 ms - 196
nodejs_1 | GET /api/v1/alumnos/alumno/bentleysharpe%40tropoli.com 200 5.398 ms - 906
nodejs_1 | POST /api/v1/auth/login 200 5.072 ms - 196
nodejs_1 | POST /api/v1/auth/login 200 5.325 ms - 196
nodejs_1 | GET /api/v1/alumnos/alumno/staceytownsend%40tropoli.com 200 6.636 ms - 669
nodejs_1 | GET /api/v1/alumnos/alumno/staceytownsend%40tropoli.com 200 5.852 ms - 669
nodejs_1 | POST /api/v1/auth/login 200 7.755 ms - 196
nodejs_1 | POST /api/v1/auth/login 200 4.471 ms - 196
nodejs_1 | GET /api/v1/alumnos/alumno/blankenshipchapman%40tropoli.com 200 6.992 ms - 1288
nodejs_1 | GET /api/v1/alumnos/alumno/blankenshipchapman%40tropoli.com 200 6.515 ms - 1288
mongod_1 | {"t":{"date":"2021-12-16T12:31:39.925+00:00"},"s":"I", "c":"STORAGE", "id":22430
, "ctx":"Checkpointner", "msg":"WiredTiger message", "attr":{"message":["1639657899:925044"] [1:0x7f672
8d96700], WT_SESSION.checkpoint: [WT_VERB_CHECKPOINT_PROGRESS] saving checkpoint snapshot min: 2261,
snapshot max: 2261 snapshot count: 0, oldest timestamp: (0, 0) , meta checkpoint timestamp: (0, 0)
base write gen: 249"}}}
```

Hecho esto, damos por concluido el ejercicio, y por tanto la práctica.
16/12/2021