

INGENIERÍA DE SERVIDORES (21-22)
Grado en Ingeniería Informática
Universidad de Granada

Arturo Alonso Carbonero

Práctica 4 : Benchmarking y Ajuste del Sistema

phoronix



ÍNDICE

1. Introducción.

2. Ejercicio 1 – Phoronix.

2.1. Phoronix en Ubuntu Server.

2.1.1. Instalación.

2.1.2. Tests.

2.2. Phoronix en CentOS.

2.2.1. Instalación.

2.2.2. Tests.

2.3. Apache Benchmark.

3. Ejercicio 2 – JMeter.

3.1. Instalación y puesta en funcionamiento.

3.2. JMeter.

3.2.1. Instalación de JMeter en el host anfitrión.

3.2.2. Uso de JMeter.

4. Referencias.

1. Introducción

Esta práctica consiste en dos ejercicios relacionados con el benchmarking, es decir, analizar desde diferentes puntos de vista el funcionamiento de un servidor.

Para la realización de la misma, partiré del mismo escenario original de la práctica anterior, esto es, pila lamp y el servicio SSH instalados pero sin fail2ban, sin Zabbix y sin Ansible. Si bien se podrían comparar los resultados ofrecidos por Zabbix con algunos de los resultados de los tests de phoronix, por ejemplo, no considero que sea extremadamente relevante, por lo que prescindiré de llevar a cabo dicha comparación.

Las direcciones IP de las maquinas son las utilizadas a lo largo del curso:

CentOs → 192.168.56.110

Ubuntu Server → 192.168.56.105

El usuario que utilizaré será el empleado en las sesiones de prácticas anteriores y se construye de la siguiente forma: Dos primeras letras del segundo apellido + Dos primeras letras del primer apellido + Primera letra del nombre:

Arturo Alonso Carbonero → alcaa.

En el host anfitrión utilizaré Ubuntu 18.04 LTS, es decir, una terminal de Ubuntu para Windows. El usuario por lo tanto diferirá del de las máquinas virtuales.

2. Ejercicio 1 – Phoronix

2.1. Phoronix en Ubuntu Server.

2.1.1. Instalación.

Para comenzar a utilizar Phoronix en la máquina con Ubuntu Server, es necesario realizar la instalación del mismo. Para ello, hemos de seguir los pasos que a continuación se muestran, comenzando por la obtención del repositorio.

```
alcaa@alcaa:~$ wget http://phoronix-test-suite.com/releases/repo/pts.debian/files/phoronix-test-suite_7.8.0_all.deb
--2021-12-14 10:50:24-- http://phoronix-test-suite.com/releases/repo/pts.debian/files/phoronix-test-suite_7.8.0_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: 555634 (543K) [application/x-debian-package]
Saving to: 'phoronix-test-suite_7.8.0_all.deb'

phoronix-test-suite_7.8. 100%[=====] 542.61K 212KB/s in 2.6s

2021-12-14 10:50:27 (212 KB/s) - 'phoronix-test-suite_7.8.0_all.deb' saved [555634/555634]
```

Desempaquetamos el contenido con el comando **dpkg** y la opción **-i** para instalarlo.

```
alcaa@alcaa:~$ sudo dpkg -i phoronix-test-suite_7.8.0_all.deb
Selecting previously unselected package phoronix-test-suite.
(Reading database ... 122024 files and directories currently installed.)
Preparing to unpack phoronix-test-suite_7.8.0_all.deb ...
Unpacking phoronix-test-suite (7.8.0) ...
Setting up phoronix-test-suite (7.8.0) ...
Processing triggers for man-db (2.9.1-1) ...
Processing triggers for mime-support (3.6.4ubuntu1) ...
```

Ya disponemos de Phoronix en la máquina. Para listar los tests que Phoronix ofrece ejecutamos el comando **phoronix-test-suite list-tests**. Al ser la primera vez que ejecutamos el comando, nos pedirá que aceptemos los términos de Phoronix. Tras esto, podemos ver la lista de los tests disponibles tal y como se aprecia en la imagen siguiente.

```
"continue" targeting switch is equivalent to "break". Did you mean to use "continue 2"? in /usr/share
e/phoronix-test-suite/pts-core/objects/pts_Graph/pts_graph_core.php:211

Phoronix Test Suite v7.8.0
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/ashes-escalation      - Ashes of the Singularity: Escalation Graphics
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/batman-knight         - Batman: Arkham Knight        Graphics
pts/batman-origins        - Batman: Arkham Origins       Graphics
pts/betsy                 - Betsy GPU Compressor         Graphics
pts/bioshock-infinite     - BioShock Infinite           Graphics
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

alcaa@alcaa:~$ phoronix-test-suite list-tests | more
```

Para probar un test, ejecutamos el comando **phoronix-test-suite run nombreTest**. Sin embargo, la primera vez no podremos llevar a cabo esta acción con éxito, obteniendo el siguiente error:

```
alcaa@alcaa:~$ phoronix-test-suite run pts/core-latency
"continue" targeting switch is equivalent to "break". Did you mean to use "continue 2"? in /usr/share/phoronix-test-suite/pts-core/objects/pts_Graph/pts_graph_core.php:211

[PROBLEM] Failed to find ZIP support for extracting file: /home/alcaa/.phoronix-test-suite/openbenchmarking.org/pts/core-latency-1.0.0.zip. Install PHP ZIP support or the unzip utility.
Updated OpenBenchmarking.org Repository Index
pts: 476 Distinct Tests, 84 Suites

[PROBLEM] Failed to find ZIP support for extracting file: /home/alcaa/.phoronix-test-suite/openbenchmarking.org/pts/core-latency-1.0.0.zip. Install PHP ZIP support or the unzip utility.

[PROBLEM] Invalid Argument: pts/core-latency

CORRECT SYNTAX:
phoronix-test-suite run [Test | Suite | OpenBenchmarking ID | Test Result] ...

See available tests to run by visiting OpenBenchmarking.org or running:

    phoronix-test-suite list-tests

Tests can be installed by running:

    phoronix-test-suite install <test-name>
```

Es necesario instalar PHP ZIP para poder ejecutar tests.

```
alcaa@alcaa:~$ sudo apt-get install php-zip
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  libzip5 php7.4-zip
The following NEW packages will be installed:
  libzip5 php-zip php7.4-zip
0 upgraded, 3 newly installed, 0 to remove and 132 not upgraded.
Need to get 70.2 kB of archives.
After this operation, 250 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 libzip5 amd64 1.5.1-0ubuntu1 [46.7 kB]
Get:2 http://es.archive.ubuntu.com/ubuntu focal-updates/universe amd64 php7.4-zip amd64 7.4.3-4ubuntu2.7 [21.5 kB]
Get:3 http://es.archive.ubuntu.com/ubuntu focal/universe amd64 php-zip all 2:7.4+75 [1996 B]
Fetched 70.2 kB in 1s (76.9 kB/s)
Selecting previously unselected package libzip5:amd64.
(Reading database ... 122583 files and directories currently installed.)
Preparing to unpack .../libzip5_1.5.1-0ubuntu1_amd64.deb ...
Unpacking libzip5:amd64 (1.5.1-0ubuntu1) ...
Selecting previously unselected package php7.4-zip.
Preparing to unpack .../php7.4-zip_7.4.3-4ubuntu2.7_amd64.deb ...
Unpacking php7.4-zip (7.4.3-4ubuntu2.7) ...
Selecting previously unselected package php-zip.
Preparing to unpack .../php-zip_2%3a7.4+75_all.deb ...
Unpacking php-zip (2:7.4+75) ...
Setting up libzip5:amd64 (1.5.1-0ubuntu1) ...
Setting up php7.4-zip (7.4.3-4ubuntu2.7) ...
Creating config file /etc/php/7.4/mods-available/zip.ini with new version
Setting up php-zip (2:7.4+75) ...
Processing triggers for libc-bin (2.31-0ubuntu9) ...
Processing triggers for libapache2-mod-php7.4 (7.4.3-4ubuntu2.7) ...
Processing triggers for php7.4-cli (7.4.3-4ubuntu2.7) ...
```

Instalación de php-zip

2.1.2. Tests.

Una vez instalada dicha función podemos comenzar a hacer uso de los tests. Antes de poder ejecutar uno, es necesario instalarlo. Para ello, ejecutamos el comando **phoronix-test-suite install nombreTest**. Sin embargo, a través de el comando para ejecutar el test podemos instalarlo, ya que en caso de no existir, nos preguntará si deseamos realizar dicha acción.

```
alcaa@alcaa:~$ phoronix-test-suite run pts/sudoku  
"continue" targeting switch is equivalent to "break". Did you mean to use "continue 2"? in /usr/share/phoronix-test-suite/pts-core/objects/pts_Graph/pts_graph_core.php:211  
[PROBLEM] pts/sudoku-1.0.1 is not installed.  
Would you like to stop and install these tests now (Y/n): Y
```

Instalación de un test desde run

Una vez instalado y ejecutado, obtenemos una pantalla como la siguiente:

```
"continue" targeting switch is equivalent to "break". Did you mean to use "continue 2"? in /usr/share/phoronix-test-suite/pts-core/objects/pts_Graph/pts_graph_core.php:211  
  
Phoronix Test Suite v7.8.0  
System Information  
  
PROCESSOR:           Intel Core i5-8250U @ 1.80GHz  
Core Count:          1  
Extensions:           SSE 4.2 + AVX2 + AVX + RDRAND + FSGSBASE  
Cache Size:           6144 KB  
  
GRAPHICS:             VMware SVGA II  
Screen:               2048x2048  
  
MOTHERBOARD:          Oracle VirtualBox v1.2  
BIOS Version:         VirtualBox  
Chipset:               Intel 440FX- 82441FX PMC  
Audio:                 Intel 82801AA AC 97 Audio  
Network:               Intel 82540EM Gigabit  
  
MEMORY:               1024MB  
  
DISK:                  116GB VBOX HDD  
File-System:           ext4  
Mount Options:         reltime rw  
Disk Scheduler:        MQ-DEADLINE  
  
OPERATING SYSTEM:     Ubuntu 20.04  
Kernel:                5.4.0-91-generic (x86_64)  
Compiler:              GCC 9.3.0  
System Layer:           Oracle VMWare  
Security:               KPTI + usercopy/swaps barriers and __user pointer sanitization + Full generic  
retpoline STIBP: disabled RSB filling Protection  
  
Would you like to save these test results (Y/n):
```

Información del sistema

Phoronix nos permitirá guardar los resultados obtenidos.

```
Would you like to save these test results (Y/n): Y  
Enter a name to save these results under: ejemplo  
Enter a unique name to describe this test run / configuration:  
  
If desired, enter a new description below to better describe this result set / system configuration  
under test.  
Press ENTER to proceed without changes.  
  
Current Description: Oracle VMWare testing on Ubuntu 20.04 via the Phoronix Test Suite.  
New Description:
```

Tras decidir si guardamos o no los resultados y, en caso afirmativo, la forma en la que lo haremos, obtendremos el resultado de la ejecución del test que hayamos ejecutado. Dependiendo del test, tomará más o menos tiempo (del orden de minutos).

En la imagen siguiente se puede ver el resultado de la ejecución del test **Sudoku**. Este test se encarga de medir el tiempo que tarda el procesador en ejecutar 100 sudokus.

```
Sudoku 0.4:
nts/sudoku-1.0.1
Test 1 of 1
Estimated Trial Run Count: 3
Estimated Time To Completion: 6 Minutes [10:06 UTC]
Started Run 1 @ 10:01:25
Started Run 2 @ 10:01:58
Started Run 3 @ 10:02:31

Total Time:
32.050171852112
31.300653934479
30.268803119659

Average: 31.21 Seconds
Deviation: 2.87%
```

El resultado ofrecido indica que el test se ha realizado una vez (azul), el número estimado de pruebas que se van a realizar (verde), el tiempo total estimado (naranja), la hora a la que se ha realizado cada prueba (rosa), el tiempo que ha tardado cada prueba (amarillo) y el tiempo medio de todas ellas (rojo). El campo *Deviation* (gris) indica la desviación estándar entre cada prueba. Si entre prueba y prueba supera el 3,5% (valor ajustable) se ejecutará una prueba extra para aumentar la precisión de los resultados.

Si se guarda el resultado, obtendremos tras la ejecución un enlace donde podremos observar el resultado desde la web de Phoronix.

```
Sudoku 0.4
Total Time
Seconds < Lower Is Better
Intel Core i5-8250U .. 70.86 |=====

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 r

Results Uploaded To: https://openbenchmarking.org/result/2112155-F0-SEJ31616593
```

En las siguientes imágenes se observa el contenido de dicho enlace, donde podemos ver información sobre el sistema donde se ha ejecutado el test y los resultados del mismo, entre otros.

Oracle VMware testing on Ubuntu 20.04 via the Phoronix Test Suite.

Compare your own system(s) to this result file with the [Phoronix Test Suite](#) by running the command: `phoronix-test-suite benchmark 2112155-FO-SEJ31616593`
[Jump To Table - Results](#)

Statistics

Graph Settings Table

☐ Remove Outliers Before Calculating Averages ☐ Prefer Vertical Bar Graphs ☐ Show Detailed System Result Table

Run Management

RESULT	PERFORMANCE PER	DATE	TEST
IDENTIFIER	DOLLAR	RUN	DURATION
Intel Core i5-8250U		December 15	

Refresh Results

sej

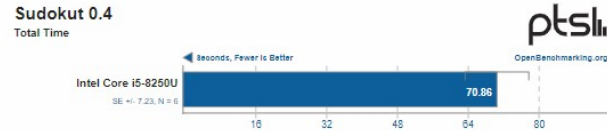
OpenBenchmarking.org	Phoronix Test Suite 10.6.1
Intel Core i5-8250U @ 1.80GHz (1 Core)	Processor
Oracle VirtualBox v1.2	Motherboard
Intel 440FX- 82441FX PMC	Chipset
1024MB	Memory
11GB VBOX HDD	Disk
VMware SVGA II	Graphics
Intel 82801AA AC 97 Audio	Audio
Intel 82540EM Gigabit	Network
Ubuntu 20.04	OS
5.4.0-91-generic (x86_64)	Kernel
GCC 9.3.0	Compiler
ext4	File-System
2048x2048	Screen Resolution
Oracle VMware	System Layer

Sej Benchmarks

- KPTI + usercopy/swaps barriers and __user pointer sanitization + Full generic retpoline STIBP: disabled
RSB filling Protection

Sudoku

This is a test of Sudoku, which is a Sudoku puzzle solver written in Tcl. This test measures how long it takes to solve 100 Sudoku puzzles. [Learn more via the OpenBenchmarking.org test page.](#)



Para finalizar el ejercicio, ejecutaré un test más en la máquina de Ubuntu Server. En este caso probaré el test **cachebench**. Este test está diseñado para realizar una prueba del rendimiento del ancho de banda de la memoria y la caché.

```
CacheBench:
pts/cachebench-1.1.2 [Test: Read]
Test 1 of 1
Estimated Trial Run Count:      3
Estimated Time To Completion: 3 Minutes [09:30 UTC]
  Started Run 1 @ 09:27:52
  Started Run 2 @ 09:30:00
  Started Run 3 @ 09:32:06

Test: Read:
2686.2398094762
2734.2343341429
2587.0596418571

Average: 2669.18 MB/s
Deviation: 2.81%
```

Resultados de cachebench

Nota: el formato de los resultados es el mismo que en el test anterior.

2.2. Phoronix en CentOS.

2.2.1. Instalación.

En el caso de CentOS el proceso de instalación es algo diferente con respecto a Ubuntu Server. Antes de proceder con el mismo, debemos instalar ciertas funciones que no vienen por defecto en CentOS.

```
lcalca@localhost ~]$ sudo yum install wget php-cli php-xml bzip2
```

Una vez instalados **wget**, **php-cli**, **php-xml** y **bzip2**, podemos comenzar con la instalación de Phoronix. Desde este punto es similar a la instalación en Ubuntu Server. Obtenemos el repositorio, desempaquetamos el contenido e instalamos Phoronix ejecutando el instalador obtenido tal y como se muestra en las imágenes.

```
lcalca@localhost ~]$ wget https://phoronix-test-suite.com/releases/phoronix-test-suite-8.4.1.tar.gz
--2021-12-15 04:41:15-- https://phoronix-test-suite.com/releases/phoronix-test-suite-8.4.1.tar.gz
Resolviendo phoronix-test-suite.com (phoronix-test-suite.com)... 192.211.48.82
Conectando con phoronix-test-suite.com (phoronix-test-suite.com)[192.211.48.82]:443... conectado.
Petición HTTP enviada, esperando respuesta... 200 OK
Longitud: 873706 (853K) [application/x-gzip]
Grabando a: "phoronix-test-suite-8.4.1.tar.gz"

phoronix-test-suite-8.4. 100%[=====>] 853,23K 740KB/s en 1,2s
2021-12-15 04:41:15 (740 KB/s) - "phoronix-test-suite-8.4.1.tar.gz" guardado [873706/873706]
```

Obtención del repositorio

```
lcalca@localhost ~]$ tar xvfz phoronix-test-suite-8.4.1.tar.gz
```

Desempaquetar contenido

```
lcalca@localhost ~]$ cd phoronix-test-suite/
lcalca@localhost phoronix-test-suite]$ 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/
```

Instalación de Phoronix

Para listar los tests ejecutamos el comando **phoronix-test-suite list-available-tests**. Obtendremos de nuevo un error. En este caso, será necesario instalar la extensión JSON para PHP (indicado en verde).

```
lcalca@localhost ~]$ phoronix-test-suite list-available-tests

The following PHP extensions are REQUIRED:

JSON      JSON support is required for OpenBenchmarking.org.

The following PHP extensions are OPTIONAL but recommended:

GD         The GD library is recommended for improved graph rendering.
POSIX      POSIX support is highly recommended.
```

```

[alcaa@localhost ~]$ sudo yum install php-json.x86_64
Ultima comprobacion de caducidad de metadatos hecha hace 17:06:23, el mar 14 dic 2021 11:47:36 EST.
Dependencias resueltas.
=====
Paquete          Arquitectura  Versión                               Repositorio  Tam.
=====
Instalando:
php-json          x86_64        7.2.24-1.module_el8.2.0+313+b04d0a66  AppStream    73 k
Resumen de la transacción
=====
Instalar 1 Paquete
Tamaño total de la descarga: 73 k
Tamaño instalado: 44 k
¿Está de acuerdo [s/N]?: s
Descargando paquetes:
php-json-7.2.24-1.module_el8.2.0+313+b04d0a66.x86_64.rpm          435 kB/s | 73 kB      00:00
-----
Total                                                                189 kB/s | 73 kB      00:00
Ejecutando verificación de operación
Verificación de operación exitosa.
Ejecutando prueba de operaciones
Prueba de operación exitosa.
Ejecutando operación
Preparando      :                               1/1
Instalando      : php-json-7.2.24-1.module_el8.2.0+313+b04d0a66.x86_64 1/1
Ejecutando scriptlet: php-json-7.2.24-1.module_el8.2.0+313+b04d0a66.x86_64 1/1
Verificando     : php-json-7.2.24-1.module_el8.2.0+313+b04d0a66.x86_64 1/1
Instalado:
php-json-7.2.24-1.module_el8.2.0+313+b04d0a66.x86_64
¡Listo!

```

Instalación de json-php

Desde este punto podremos utilizar Phoronix con normalidad.

```

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.

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-enlang          - Timed Enlang/GTP Compilation Processor

[alcaa@localhost ~]$ phoronix-test-suite list-available-tests | more

```

Lista de los tests disponibles

2.2.2. Tests.

La ejecución de los tests funciona igual que en Ubuntu Server y los resultados de los mismos se muestran de la misma forma. En las siguientes imágenes se muestra el resultado de la ejecución de los mismos tests que en el apartado de Ubuntu Server.

```
System Information

PROCESSOR:          Intel Core i5-8250U
Core Count:         1
Extensions:          SSE 4.2 + AUX2 + AUX + RDRAND + FSGSBASE
Cache Size:          6144 KB

GRAPHICS:            VMware SVGA II
Screen:              2048x2048

MOTHERBOARD:         Oracle VirtualBox v1.2
BIOS Version:         VirtualBox
Chipset:              Intel 440FX 82441FX PMC
Audio:               Intel 82801AA AC 97 Audio
Network:              Intel 82540EM Gigabit

MEMORY:              818MB

DISK:                9GB VBOX HDD
File-System:          xfs
Mount Options:         attr2 inode64 noquota relatime rw seclabel
Disk Scheduler:        MQ-DEADLINE

OPERATING SYSTEM:     CentOS Linux 8
Kernel:               4.18.0-193.el8.x86_64 (x86_64)
System Layer:          Oracle VMware
Security:              SELinux + KPTI + usercopy/swaps barriers and __user pointer sanitization + Full generic retpoline STIBP: disabled RSB filling + PTE Inversion
```

Información del Sistema

```
Sudoku 0.4:
pts/sudoku-1.0.1
Test 1 of 1
Estimated Trial Run Count:    3
Estimated Time To Completion: 2 Minutes [05:01 EST]
    Started Run 1 @ 04:59:26
    Started Run 2 @ 05:00:03
    Started Run 3 @ 05:00:40

Total Time:
    36.203042984009
    36.260391950607
    36.381829977036

Average: 36.28 Seconds
Deviation: 0.25%
```

Resultado de sudoku en CentOS

```
CacheBench:
pts/cachebench-1.1.2 [Test: Read]
Test 1 of 1
Estimated Trial Run Count: 3
Estimated Time To Completion: 3 Minutes [05:33 EST]
  Started Run 1 @ 05:30:45
  Started Run 2 @ 05:32:51
  Started Run 3 @ 05:34:57
  Started Run 4 @ 05:37:03 *

Test: Read:
  2472.2062067619
  2674.9227648571
  2553.1822613333
  2585.7201204762

Average: 2571.51 MB/s
Deviation: 3.26%
```

Resultado de cachebench en CentOS

Los resultados son similares en ambas máquinas, siendo CentOS generalmente algo más lento que Ubuntu Server. Como se puede ver, los tiempos obtenidos son algo mayores en CentOS.

2.3. Apache Benchmark.

Para comenzar a monitorizar con Apache Benchmark es necesario instalar previamente apache2 en Ubuntu Server y httpd en CentOS, lo cual se llevó a cabo en la práctica 2, por lo que no se mostrará. Esto es debido a que utilizaremos el comando **ab** para generar carga en el servicio http de los servidores desde el host anfitrión. Para ello, descargamos Apache Benchmark en el host anfitrión en <https://www.apachelounge.com/download/>.

Apache Lounge
Webmasters

Apache 2.4 VS16 Windows Binaries and Modules

Apache Lounge has provided up-to-date Windows binaries and popular third-party modules for more than 15 years. We have hundreds of thousands of satisfied users: small and big companies as well as home users. Always build with up to date dependencies and latest compilers, and tested thorough. The binaries are referenced by the ASF, Microsoft, PHP etc. and more and more software is packaged with our binaries and modules.

The binaries, are build with the sources from ASF at httpd.apache.org, contains the latest patches and latest dependencies like zlib, openssl etc. which makes the downloads here mostly more actual then downloads from other places. The binaries **do not run** on XP and 2003. Runs on: 7 SP1, Vista SP2, 8/8.1, 10, 11 Server 2008 SP2 / R2 SP1, Server 2012 / R2, Server 2016/2019/2022.

Build with the latest Windows® Visual Studio C++ 2019 aka VS16. VS16 has improvements, fixes and optimizations over VC15 in areas like Performance, MemoryManagement, New standard conformance features, Code generation and Stability. For example code quality tuning and improvements done across different code generation areas for "speed". And makes more use of latest processors and supported Windows editions (win7 and up) internal features.

VS16 is backward compatible, see [Compatibility VS16](#). You can use a VC15/14 module inside a VS16 binary, for example PHP VC15/14 as module.

Be sure you installed latest 14.30.30704 Visual C++ Redistributable for Visual Studio 2015-2022 : vc_redist_x64 or vc_redist_x86 see [Redistributable](#)

Apache 2.4 binaries VS16
[Info & Changelog](#)

Apache 2.4.51 Win64
● [httpd-2.4.51-win64-VS16.zip](#) 07 Oct '21 10.409K
PGP Signature (Public [PGP key](#)), SHA1-SHA512 [checksums](#)

Apache 2.4.51 Win32
● [httpd-2.4.51-win32-VS16.zip](#) 07 Oct '21 9.502K
PGP Signature (Public [PGP key](#)), SHA1-SHA512 [checksums](#)

To be sure that a download is intact and has not been tampered with, use PGP, see [PGP Signature](#)

Apache 2.4 modules VS16

Mail for the PGP signatures and/or SHA checksums to verify the contents of a file.

Keep Server Online
If you find the downloads useful, please express your satisfaction with a donation.

Una vez descargado, busquemos el directorio **Apache24/conf** y modificamos el valor **[ServerName]** en el fichero de configuración **httpd.conf**.

Disco local (C:) > Este equipo > Disco local (C:)

Nombre	Fecha de modifica...	Tipo
Apache24	07/10/2021 16:33	Carpeta de
apex_21.1	07/11/2021 12:57	Carpeta de

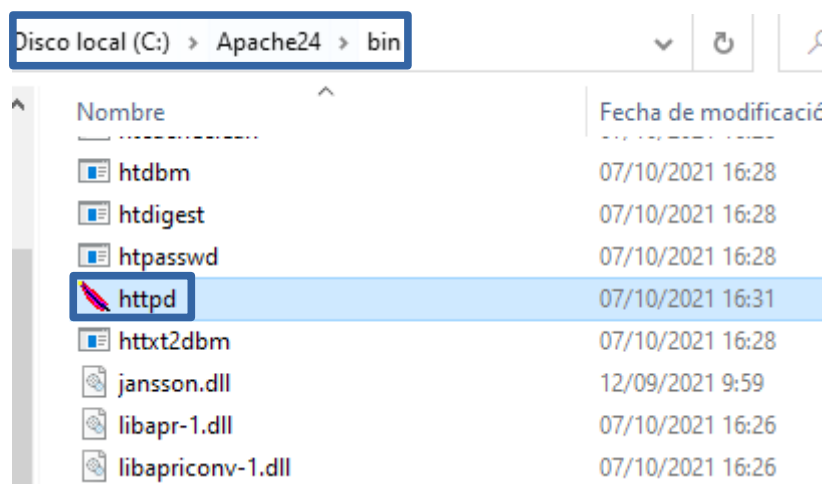
Disco local (C:) > Apache24 > conf

Nombre	Fecha de modificaci
extra	07/10/2021 16:33
original	07/10/2021 16:33
charset.conv	07/10/2021 16:31
httpd.conf	07/10/2021 16:31
magic	07/10/2021 16:31
mime.types	07/10/2021 16:31
openssl.cnf	24/08/2021 15:38

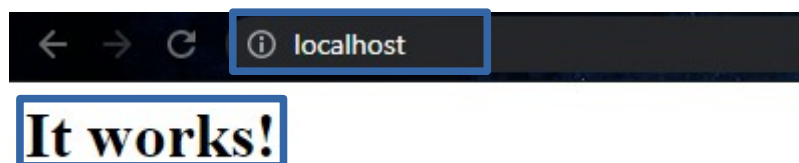
```
# in which case these default settings will be overridden for the
# virtual host being defined.
#
#
# ServerAdmin: Your address, where problems with the server should be
# e-mailed. This address appears on some server-generated pages, such
# as error documents. e.g. admin@your-domain.com
#
ServerAdmin admin@example.com
#
# ServerName gives the name and port that the server uses to identify itself.
# This can often be determined automatically, but we recommend you specify
# it explicitly to prevent problems during startup.
#
# If your host doesn't have a registered DNS name, enter its IP address here.
#
ServerName localhost:80
#
# Deny access to the entirety of your server's filesystem. You must
# explicitly permit access to web content directories in other
# <Directory> blocks below.
#
```

Valor modificado

Ejecutamos el archivo ejecutable de la siguiente imagen y comprobamos desde el navegador que está funcionando de forma correcta.



Fichero ejecutable



Comprobación desde el navegador

Comprobamos que apache2 y httpd están respectivamente activos en Ubuntu Server y CentOS mediante el comando **systemctl status nombreServicio**.

```
alcaa@alcaa:~$ systemctl status apache2
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor preset: ena
   Active: active (running) since Wed 2021-12-15 09:21:32 UTC; 1h 43min ago
     Docs: https://httpd.apache.org/docs/2.4/
   Process: 654 ExecStart=/usr/sbin/apachectl start (code=exited, status=0/SUCCESS)
  Main PID: 760 (apache2)
    Tasks: 7 (limit: 1072)
   Memory: 16.8M
   CGroup: /system.slice/apache2.service
           └─ 760 /usr/sbin/apache2 -k start
              764 /usr/sbin/apache2 -k start
              768 /usr/sbin/apache2 -k start
              769 /usr/sbin/apache2 -k start
              776 /usr/sbin/apache2 -k start
              777 /usr/sbin/apache2 -k start
             2867 /usr/sbin/apache2 -k start

Dec 15 09:21:31 alcaa systemd[1]: Starting The Apache HTTP Server...
Dec 15 09:21:32 alcaa apachectl[726]: AH00558: apache2: Could not reliably determine
Dec 15 09:21:32 alcaa systemd[1]: Started The Apache HTTP Server.
```

Ubuntu Server

```
alcaa@localhost ~]$ systemctl status httpd
● httpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; enabled; vendor preset
 Drop-In: /usr/lib/systemd/system/httpd.service.d
           └─ php-fpm.conf
   Active: active (running) since Wed 2021-12-15 04:21:55 EST; 1h 44min ago
     Docs: man:httpd.service(8)
  Main PID: 862 (httpd)
   Status: "Total requests: 11; Idle/Busy workers 100/0;Requests/sec: 0.00176; B
    Tasks: 213 (limit: 5019)
   Memory: 25.9M
   CGroup: /system.slice/httpd.service
           └─ 862 /usr/sbin/httpd -DFOREGROUND
              948 /usr/sbin/httpd -DFOREGROUND
              950 /usr/sbin/httpd -DFOREGROUND
              951 /usr/sbin/httpd -DFOREGROUND
              992 /usr/sbin/httpd -DFOREGROUND

dic 15 04:21:53 localhost.localdomain systemd[1]: Starting The Apache HTTP Server
dic 15 04:21:55 localhost.localdomain httpd[862]: AH00558: httpd: Could not reli
dic 15 04:21:55 localhost.localdomain systemd[1]: Started The Apache HTTP Server
dic 15 04:21:55 localhost.localdomain httpd[862]: Server configured, listening o
```

CentOs

A continuación, desde el host anfitrión, ejecutamos el comando **ab** para generar la carga. En este caso el comando será el siguiente:

ab -c 1 -n 100 http://192.168.56.105/ (Ubuntu Server)

ab -c 1 -n 100 http://192.168.56.110/ (CentOs)

-c → Indica la pseudoconurrencia, es decir, el número de paquetes que se emitirán.

-n → Indica el número de peticiones a realizar.


```

alonsoarturo@DESKTOP-UKJ4G5Q:/mnt/c/Apache24/conf$ ab -c 1 -n 100 http://192.168.56.105/
This is ApacheBench, Version 2.3 <$Revision: 1807734 $>
Copyright 1996 Adam Twiss, Zeus Technology Ltd, http://www.zeustech.net/
Licensed to The Apache Software Foundation, http://www.apache.org/

Benchmarking 192.168.56.105 (be patient).....done


Server Software:      Apache/2.4.41
Server Hostname:      192.168.56.105
Server Port:          80

Document Path:        /
Document Length:      10918 bytes

Concurrency Level:    1
Time taken for tests:  0.949 seconds
Complete requests:    100
Failed requests:       0
Total transferred:    1119200 bytes
HTML transferred:     1091800 bytes
Requests per second:  105.43 [#/sec] (mean)
Time per request:     9.485 [ms] (mean)
Time per request:     9.485 [ms] (mean, across all concurrent requests)
Transfer rate:        1152.28 [Kbytes/sec] received

Connection Times (ms)
              min    mean[+/-sd] median    max
Connect:        0      1   1.6      0      15
Processing:      5      9   4.4      6      27
Waiting:        3      8   3.9      6      21
Total:          5      9   5.0      7      29

Percentage of the requests served within a certain time (ms)
 50%      7
 66%     11
 75%     12
 80%     14
 90%     15
 95%     18
 98%     27
 99%     29
100%     29 (longest request)

```

Resultado para Ubuntu Server

```

alonsoarturo@DESKTOP-UKJ4G5Q:/mnt/c/Apache24/conf$ ab -c 1 -n 100 http://192.168.56.110/
This is ApacheBench, Version 2.3 <$Revision: 1807734 $>
Copyright 1996 Adam Twiss, Zeus Technology Ltd, http://www.zeustech.net/
Licensed to The Apache Software Foundation, http://www.apache.org/

Benchmarking 192.168.56.110 (be patient).....done


Server Software:      Apache/2.4.37
Server Hostname:      192.168.56.110
Server Port:          80

Document Path:        /
Document Length:      78562 bytes

Concurrency Level:     1
Time taken for tests:  2.465 seconds
Complete requests:     100
Failed requests:       73
   (Connect: 0, Receive: 0, Length: 73, Exceptions: 0)
Total transferred:    7873569 bytes
HTML transferred:     7856269 bytes
Requests per second:  40.58 [#/sec] (mean)
Time per request:     24.645 [ms] (mean)
Time per request:     24.645 [ms] (mean, across all concurrent requests)
Transfer rate:        3119.87 [Kbytes/sec] received


Connection Times (ms)
              min      mean[+/-sd]  median    max
Connect:        0        1   0.2         1      1
Processing:    14       24   6.3        20     41
Waiting:        6       14   4.6        12     28
Total:         15       25   6.4        21     42


Percentage of the requests served within a certain time (ms)
 50%      21
 66%      26
 75%      31
 80%      32
 90%      35
 95%      36
 98%      39
 99%      42
100%      42 (longest request)

```

Resultado para CentOS

Además de información de carácter general en lo que al servidor respecta, como el nombre, el software o el puerto utilizado, **ab** ofrece información sobre la carga que puede ser interesante. Los campos más relevantes son:

- **Time taken for tests:** Tiempo total de la prueba.
- **Requests per second:** Número de peticiones por segundo.
- **Time per request:** Tiempo por petición en mili-segundos.
- **Transfer rate:** Frecuencia de transferencia en KB/seg.
- **Connection times:** Tiempos relacionados con la conexión y el procesamiento.

Como se puede observar en las imágenes, el tiempo empleado en CentOS en el campo **Time taken for tests** es de 2,465 segundos, frente a los 0,949 segundos de Ubuntu Server. Esto quiere decir que CentOS es más lento que Ubuntu Server.

3. Ejercicio 2 – JMeter

3.1. Instalación y puesta en funcionamiento.

Para instalar y utilizar JMeter, es necesario instalar **docker** y **docker-compose** en nuestra máquina con Ubuntu Server.

```
alcaa@alcaa:~$ sudo apt install docker docker-compose
```

Tras ello comprobamos si la instalación se ha llevado a cabo de forma adecuada.

```
alcaa@alcaa:~$ docker -v
Docker version 20.10.7, build 20.10.7-0ubuntu5~20.04.2
alcaa@alcaa:~$ docker-compose -v
docker-compose version 1.25.0, build unknown
```

-v para mostrar la versión

A continuación, obtenemos el repositorio de GitHub para la práctica indicado en el guión de la misma. El enlace se encuentra en el apartado de referencias.

```
alcaa@alcaa:~$ git clone http://github.com/davidPalomar-ugr/iseP4JMeter
Cloning into 'iseP4JMeter'...
warning: redirecting to https://github.com/davidPalomar-ugr/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 | 4.57 MiB/s, done.
Resolving deltas: 100% (715/715), done.
```

Con esto, aparece en nuestra máquina el directorio **iseP4JMeter**, donde dispondremos de los ficheros necesarios para la práctica. Nos situamos en dicho directorio y ejecutamos el comando **sudo docker-compose up** para levantar el servicio. Si no está activado, lo activamos mediante el comando **systemctl** y, posteriormente, lo levantamos (la primera vez se demorará unos minutos ya que descargará varios contenedores).

```
alcaa@alcaa:~$ systemctl status docker
• docker.service - Docker Application Container Engine
  Loaded: loaded (/lib/systemd/system/docker.service; enabled; vendor preset: enabled)
  Active: inactive (dead)
  TriggeredBy: • docker.socket
  Docs: https://docs.docker.com
alcaa@alcaa:~$ sudo systemctl enable docker
alcaa@alcaa:~$ sudo systemctl start docker
alcaa@alcaa:~$ systemctl status docker
• docker.service - Docker Application Container Engine
  Loaded: loaded (/lib/systemd/system/docker.service; enabled; vendor preset: enabled)
  Active: active (running) since Wed 2021-12-15 11:21:27 UTC; 1s ago
  TriggeredBy: • docker.socket
  Docs: https://docs.docker.com
  Main PID: 6941 (dockerd)
  Tasks: 8
  Memory: 117.9M
  CGroup: /system.slice/docker.service
          └─6941 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock

Dec 15 11:21:27 alcaa dockerd[6941]: time="2021-12-15T11:21:27.011787798Z" level=warning msg="Your
Dec 15 11:21:27 alcaa dockerd[6941]: time="2021-12-15T11:21:27.011984180Z" level=warning msg="Your
Dec 15 11:21:27 alcaa dockerd[6941]: time="2021-12-15T11:21:27.012034746Z" level=warning msg="Your
Dec 15 11:21:27 alcaa dockerd[6941]: time="2021-12-15T11:21:27.015577137Z" level=info msg="Loading
Dec 15 11:21:27 alcaa dockerd[6941]: time="2021-12-15T11:21:27.251013489Z" level=info msg="Default
Dec 15 11:21:27 alcaa dockerd[6941]: time="2021-12-15T11:21:27.432548776Z" level=info msg="Loading
Dec 15 11:21:27 alcaa dockerd[6941]: time="2021-12-15T11:21:27.667602171Z" level=info msg="Docker
Dec 15 11:21:27 alcaa dockerd[6941]: time="2021-12-15T11:21:27.673647026Z" level=info msg="Daemon
Dec 15 11:21:27 alcaa systemd[1]: Started Docker Application Container Engine.
Dec 15 11:21:27 alcaa dockerd[6941]: time="2021-12-15T11:21:27.767867153Z" level=info msg="API lis

alcaa@alcaa:~/iseP4JMeter$ sudo docker-compose up
```

Si el proceso se ha realizado correctamente, el servidor enviará mensajes de forma periódica tras ejecutar el comando **sudo docker-compose up** tal y como se muestra en la siguiente imagen, lo cual indica que está activo y funcionando.

```
mongodb_1 | {"t":{"date":"2021-12-16T10:28:01.178+00:00"},"s":"I", "c":"STORAGE", "id":22430, "ctx":"Checkpoint", "msg":"WiredTiger message", "attr":{"message":"[1639650481:178877] [1:0x7fb06c4e3700], WT_SESSION.checkpoint: [WT_VERB_CHECKPOINT_PROGRESS] saving checkpoint snapshot min: 2382, snapshot max: 2382 snapshot count: 0, oldest timestamp: (0, 0) , meta checkpoint timestamp: (0, 0) base write gen: 821"}}
mongodb_1 | {"t":{"date":"2021-12-16T10:29:01.220+00:00"},"s":"I", "c":"STORAGE", "id":22430, "ctx":"Checkpoint", "msg":"WiredTiger message", "attr":{"message":"[1639650541:220512] [1:0x7fb06c4e3700], WT_SESSION.checkpoint: [WT_VERB_CHECKPOINT_PROGRESS] saving checkpoint snapshot min: 2384, snapshot max: 2384 snapshot count: 0, oldest timestamp: (0, 0) , meta checkpoint timestamp: (0, 0) base write gen: 821"}}
mongodb_1 | {"t":{"date":"2021-12-16T10:30:01.261+00:00"},"s":"I", "c":"STORAGE", "id":22430, "ctx":"Checkpoint", "msg":"WiredTiger message", "attr":{"message":"[1639650601:261908] [1:0x7fb06c4e3700], WT_SESSION.checkpoint: [WT_VERB_CHECKPOINT_PROGRESS] saving checkpoint snapshot min: 2386, snapshot max: 2386 snapshot count: 0, oldest timestamp: (0, 0) , meta checkpoint timestamp: (0, 0) base write gen: 821"}}
```

Mensajes del servidor

Activamos el puerto de escucha en el firewall si es necesario. Para JMeter es el puerto 3000.

```
alcaa@alcaa:~$ sudo ufw allow 3000/tcp
Rule added
Rule added (v6)
alcaa@alcaa:~$ sudo ufw status
Status: active

To Action From
--
22022 ALLOW Anywhere
80/tcp ALLOW Anywhere
443/tcp ALLOW Anywhere
10051/tcp ALLOW Anywhere
3000/tcp ALLOW Anywhere
22022 (v6) ALLOW Anywhere (v6)
80/tcp (v6) ALLOW Anywhere (v6)
443/tcp (v6) ALLOW Anywhere (v6)
10051/tcp (v6) ALLOW Anywhere (v6)
3000/tcp (v6) ALLOW Anywhere (v6)
```

Finalmente, accedemos al servidor desde el navegador en el host anfitrión a través del puerto de escucha para comprobar que hemos tenido éxito.

← → ↻ ⚠ No es seguro | 192.168.56.105:3000

ETSII Alumnos API

Descripción de la API Restful:

POST /api/v1/auth/login

Parametros:

login:<emailUsuario>

password:<secreto>

Seguridad:

Acceso protegido con BasicAuth (etsiiApi:laApiDeLaETSIIIDaLache)

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

Resultado esperado (en verde)

3.2. JMeter.

3.2.1. Instalación de JMeter en el host anfitrión.

Puesto que JMeter está hecho en Java, será necesario tenerlo instalado en nuestro host anfitrión. Si no es así, lo descargamos. Dependiendo de la versión de JMeter, la versión de Java de la que debemos disponer será una u otra. En mi caso, utilizaré la versión mostrada en la siguiente imagen (Java 11), aunque la versión utilizada de JMeter (5.4.2) funciona con versiones inferiores (Desde Java 8 en adelante) tal y como se indica en la web.

```
blonsoarturo@DESKTOP-UKJ4G5Q:~$ java -version
openjdk version "11.0.13" 2021-10-19
OpenJDK Runtime Environment (build 11.0.13+8-Ubuntu-0ubuntu1.18.04)
OpenJDK 64-Bit Server VM (build 11.0.13+8-Ubuntu-0ubuntu1.18.04, mixed mode, sharing)
```

Procedemos con la descarga de JMeter desde la web oficial.

Apache JMeter 5.4.2 (Requires Java 8+)

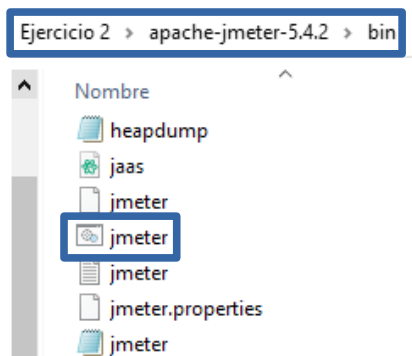
Binaries

[apache-jmeter-5.4.2.tgz sha512 pgp](#)
[apache-jmeter-5.4.2.zip sha512 pgp](#)

Source

[apache-jmeter-5.4.2_src.tgz sha512 pgp](#)
[apache-jmeter-5.4.2_src.zip sha512 pgp](#)

Una vez descargado JMeter, nos dirigimos al directorio del mismo y, en el directorio **bin**, buscamos el fichero con extensión **‘.bat’ jmeter**.



Al arrancar JMeter en este punto me he topado con un error que me impedía realizar acciones como guardar cambios o cargar ficheros locales a la aplicación. La solución ha sido cambiar la línea 112 del fichero anterior tal y como se aprecia en la imagen siguiente. En el apartado de referencias se encuentra el enlace a la solución.

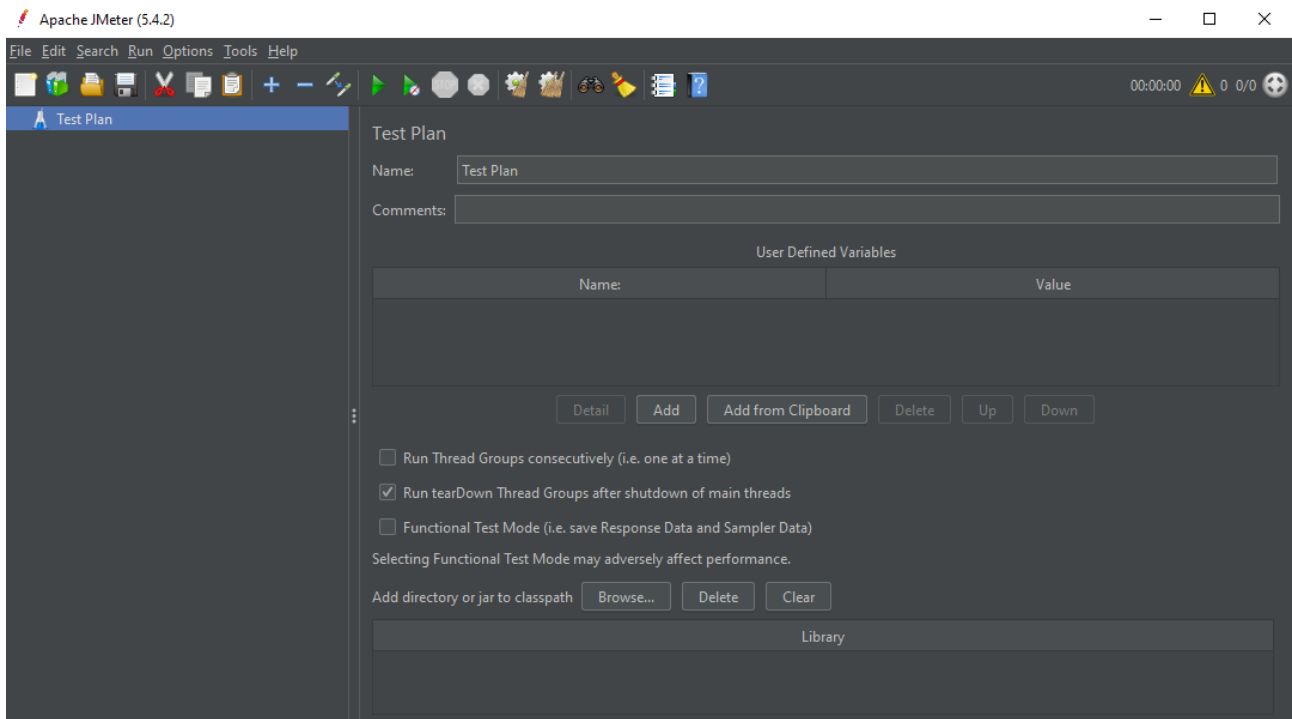
```
jmeter: Bloc de notas
Archivo Edición Formato Ver Ayuda

set ERRORLEVEL=2
goto pause
)

rem Check if version is from OpenJDK or Oracle Hotspot JVM prior to 9 containing 1.${version}.x
rem JAVAVER will be equal to "9.0.4" (quotes are part of the value) for Oracle Java 9
rem JAVAVER will be equal to "1.8.0_161" (quotes are part of the value) for Oracle Java 8
rem so we extract 2 chars starting from index 1
IF "%JAVAVER:~1,2%"=="1." (
    set JAVAVER=%JAVAVER:~1,2%
    for /f "delims=. tokens=1-3" %%v in ("%JAVAVER%") do (
        set current_minor=%%w
    )
) else (
    rem Java 9 at least
    set current_minor=9
    set JAVA9_OPTS=-add-opens java.desktop/sun.awt=ALL-UNNAMED -add-opens java.desktop/sun.swing=ALL-UNNAMED -add-opens
)

for /f "delims=. tokens=1-3" %%v in ("%MINIMAL_VERSION%") do (
    set minimal_minor=%%w
)
```

Tras esto, ya podemos acceder a JMeter con normalidad.

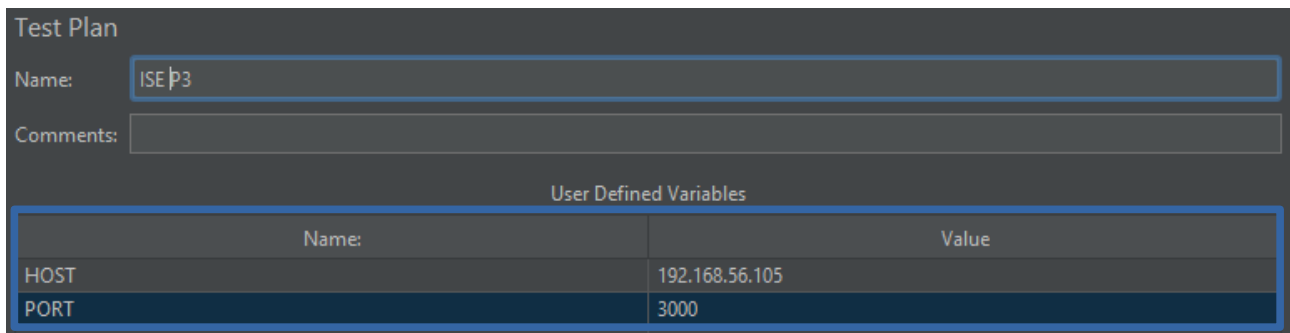


Menú de JMeter

3.2.2. Uso de JMeter.

A continuación se muestra el proceso de uso de JMeter para simular carga real en el servidor http de la máquina con Ubuntu Server:

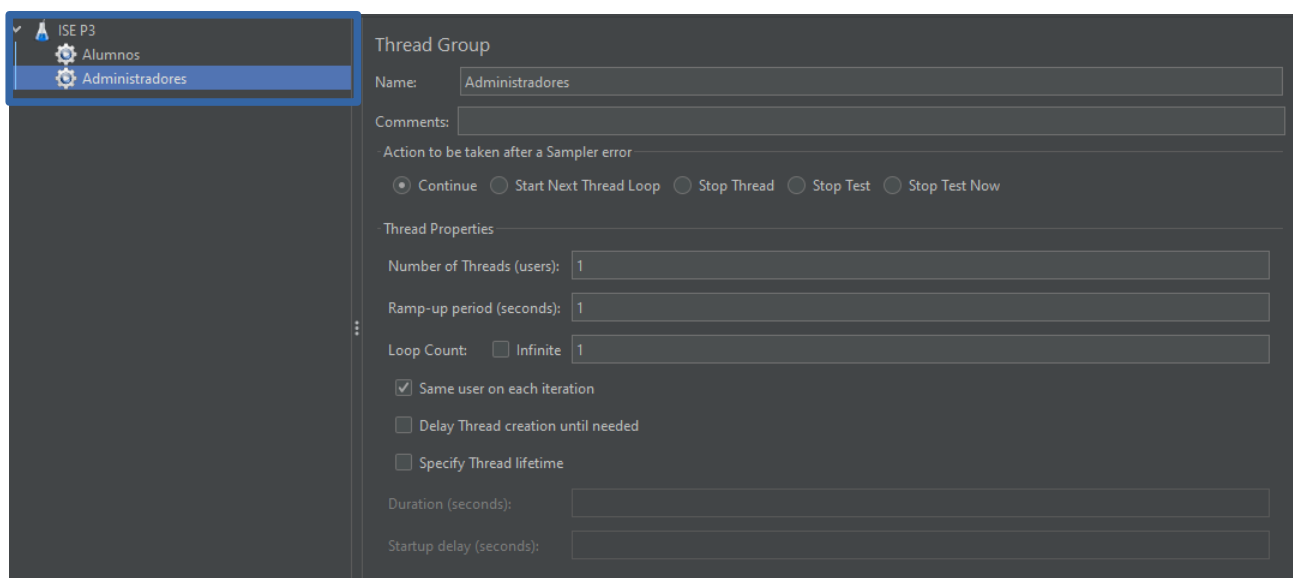
1. Parametrizamos el Host y el Puerto.



The screenshot shows the 'Test Plan' configuration in JMeter. The 'Name' field is set to 'ISE p3'. Below it, the 'User Defined Variables' table is visible, containing two entries:

Name:	Value
HOST	192.168.56.105
PORT	3000

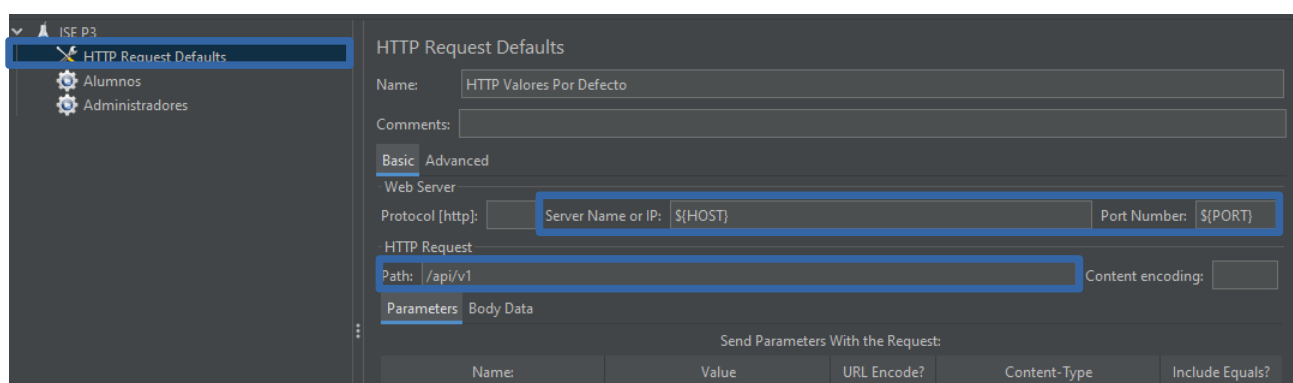
2. Creamos dos grupos de hebras para simular una situación real donde hay varios tipos de usuario. Concretamente, uno para los alumnos y otro para los administradores.



The screenshot shows the 'Thread Group' configuration for 'Administradores'. The 'Name' field is set to 'Administradores'. The 'Thread Properties' section is expanded, showing the following settings:

- Number of Threads (users): 1
- Ramp-up period (seconds): 1
- Loop Count: ☐ Infinite 1
- ☒ Same user on each iteration
- ☐ Delay Thread creation until needed
- ☐ Specify Thread lifetime
- Duration (seconds):
- Startup delay (seconds):

3. Añadimos valores por defecto para todas las peticiones de tipo HTTP. Concretamente, el host, el puerto y la ruta.



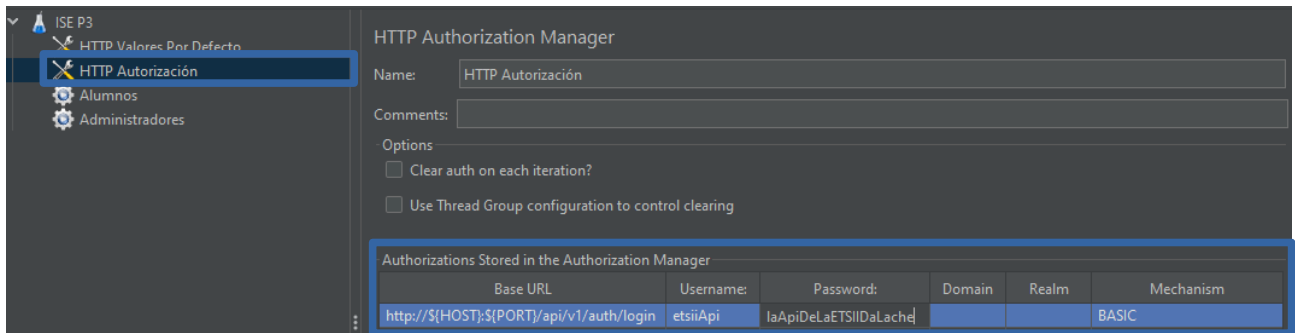
The screenshot shows the 'HTTP Request Defaults' configuration. The 'Name' field is set to 'HTTP Valores Por Defecto'. The 'Basic' tab is selected, showing the following settings:

- Protocol (http): Server Name or IP: \${HOST} Port Number: \${PORT}
- Path: /api/v1 Content encoding:
- Parameters Body Data

Below the configuration, there is a table for 'Send Parameters With the Request':

Name:	Value	URL Encode?	Content-Type	Include Equals?
-------	-------	-------------	--------------	-----------------

4. Creamos la autorización indicando la contraseña **laApiDeLaETSIIDaLache**, el URL, el nombre de la Api y 'BASIC' en el campo 'Mechanism'.



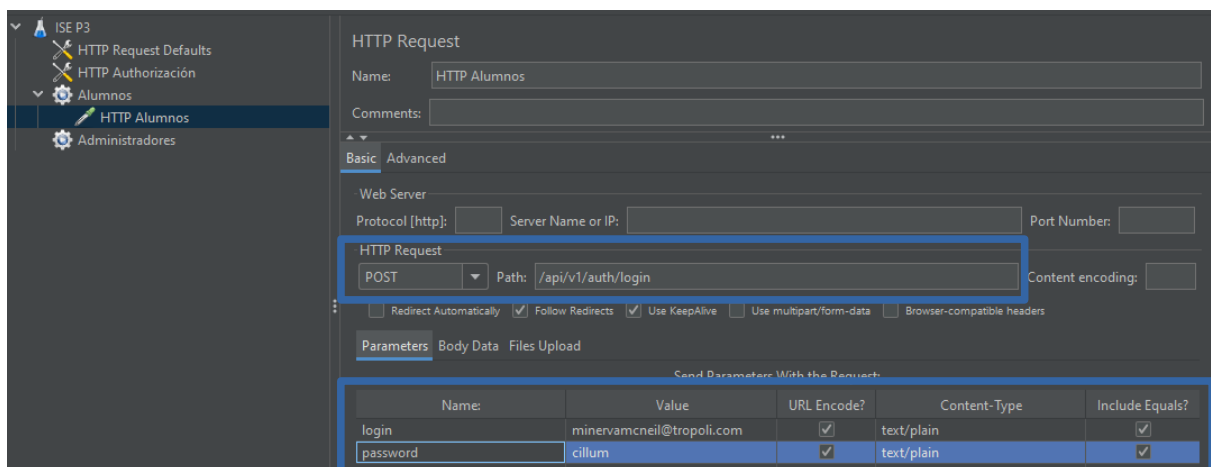
5. Para simular el acceso de un alumno y de un administrador, creamos una petición HTTP (de tipo POST) para cada grupo de hebras indicando un usuario que se encuentre en los ficheros **alumnos.csv** y **administradores.csv**.

`kaitlinsavage@etsii.ugr.es,qui`

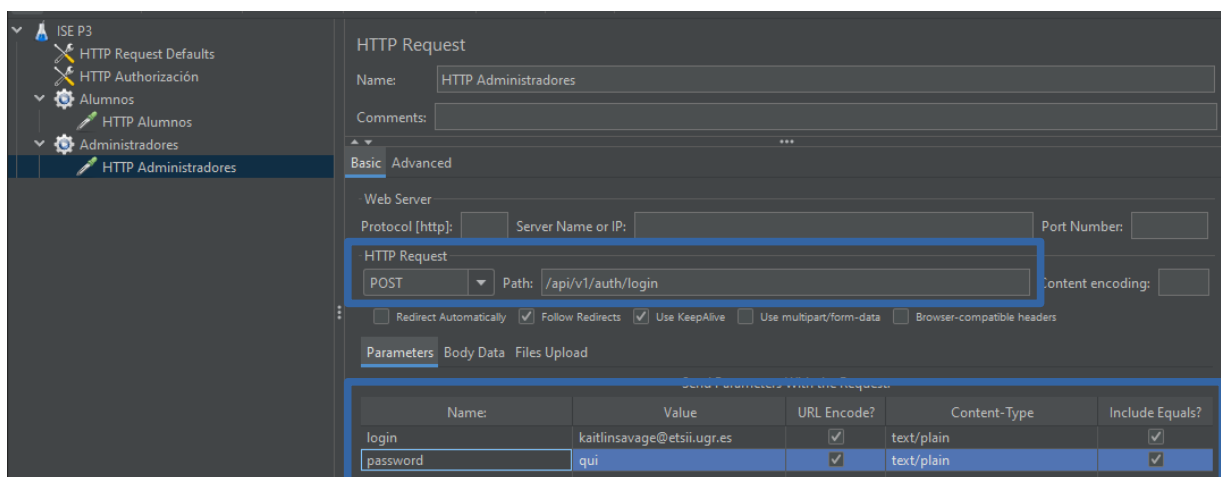
Administrador

`minervamcneil@tropoli.com,cillum`

Alumno

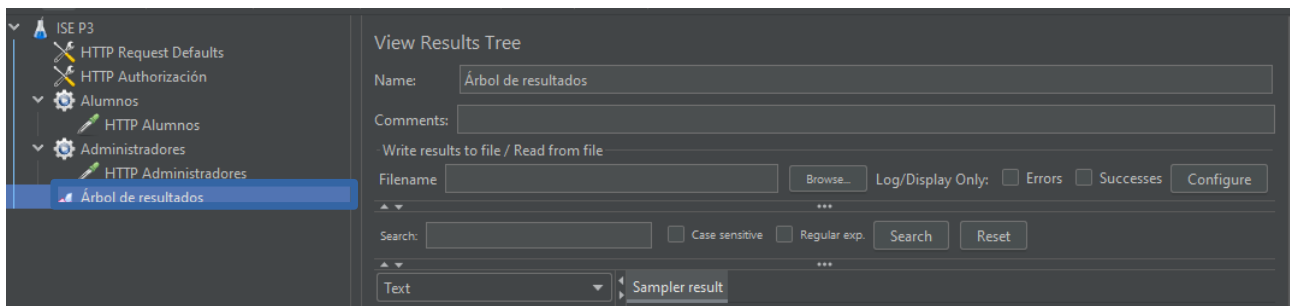


Petición de Alumnos



Petición de Administradores.

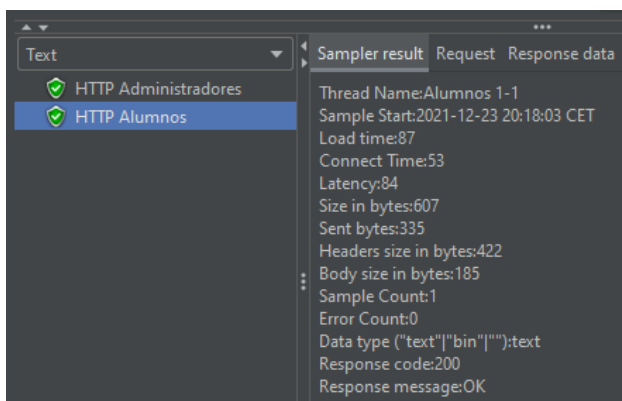
6. Creamos un árbol de resultados para visualizarlos de forma cómoda y probamos a ejecutar lo que llevamos hasta ahora para comprobar que funciona.



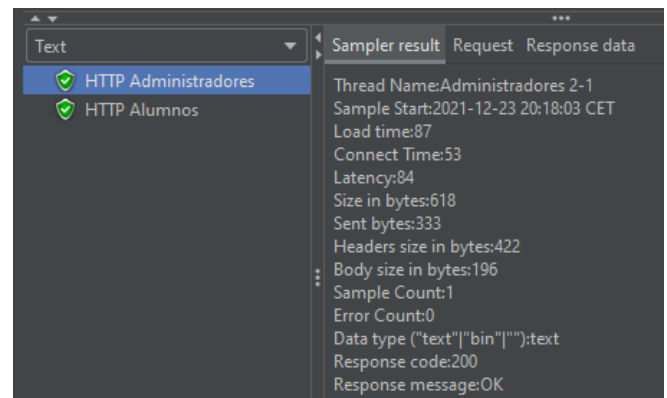
Árbol de resultados



Botones para ejecutar



Alumnos

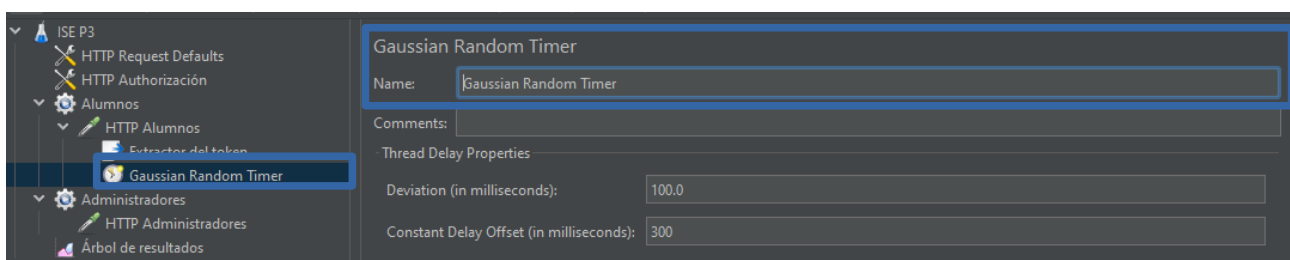


Administradores

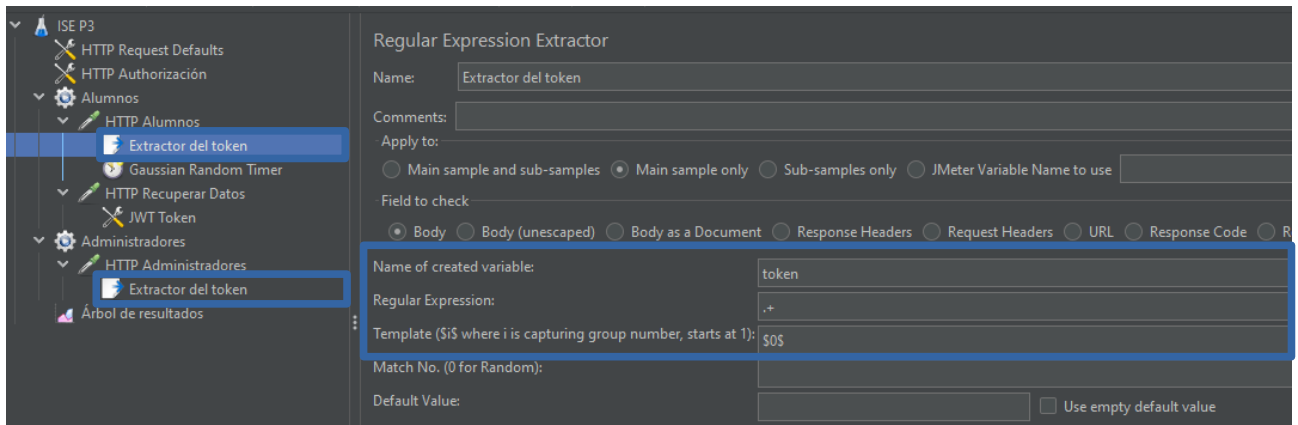
```
nodejs_1 | POST /api/v1/auth/login 200 6.651 ms - 185
nodejs_1 | POST /api/v1/auth/login 200 14.071 ms - 196
```

Mensajes del servidor (Ubuntu Server)

7. Añadimos a la petición de los Alumnos un temporizador para simular la espera entre petición y petición (de tipo Gaussiano), así como un extractor de expresiones regulares (en este caso la expresión regular `‘.+’`) para el token de las peticiones de inicio de sesión de los alumnos y de los administradores.

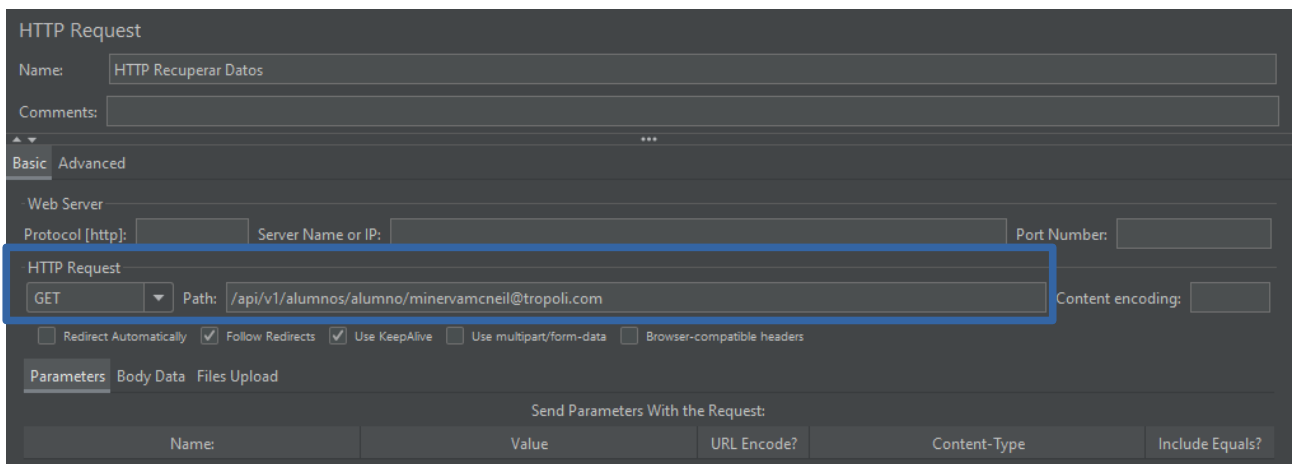


Temporizador

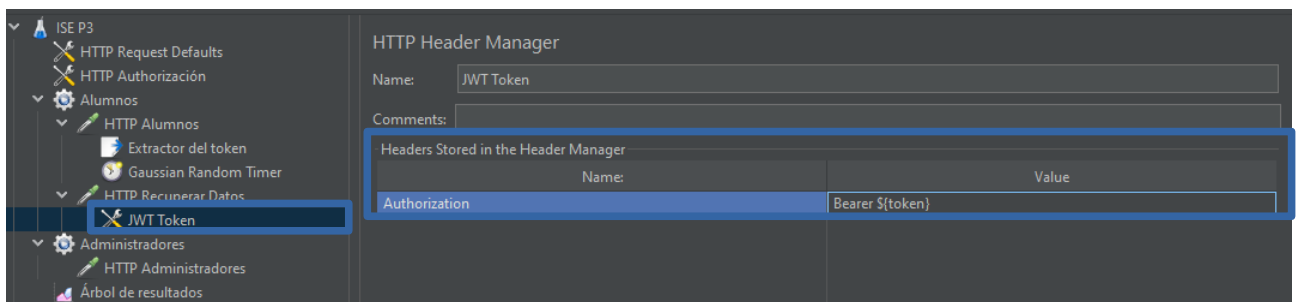


Extractor del Token

8. Creamos una petición HTTP de tipo GET para recuperar los datos de los alumnos y un gestor de cabecera para la misma para que el token sirva aunque la aplicación nos rediriga a otro nombre de dominio.

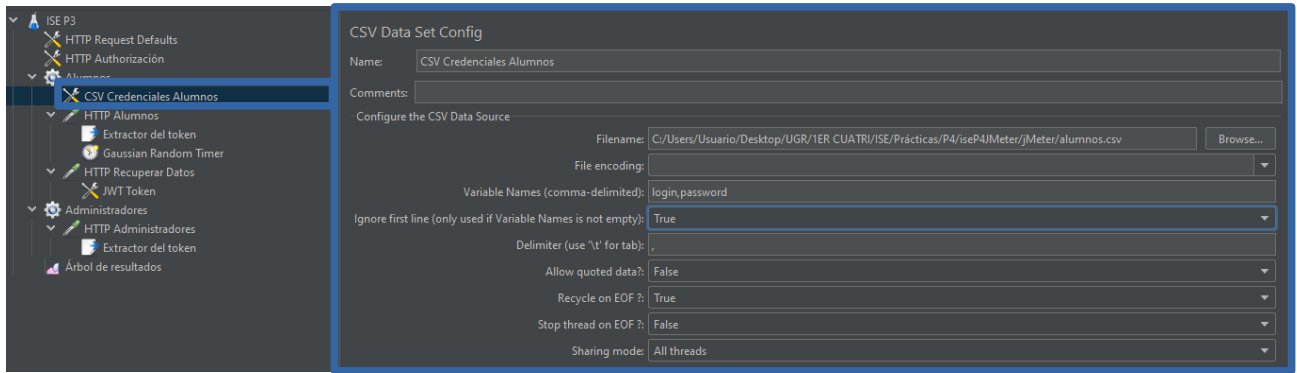


Petición de tipo GET

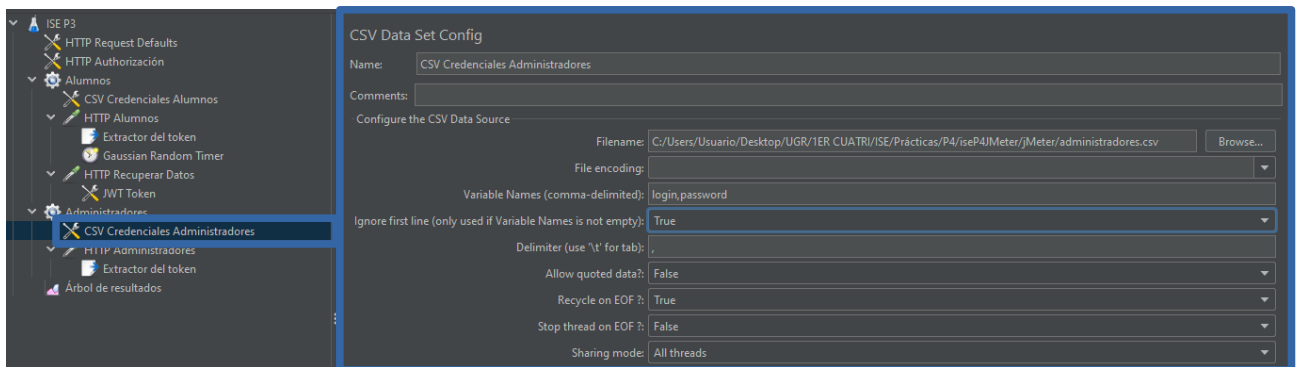


Gestor de cabecera de la petición GET

9. Realizamos la configuración de las credenciales de los alumnos y de los administradores.

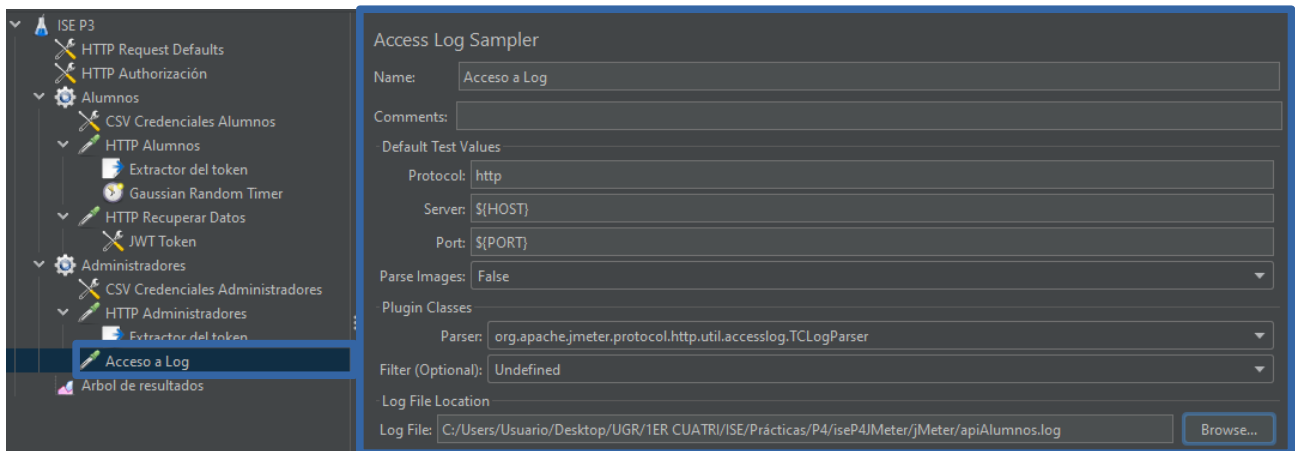


Alumnos

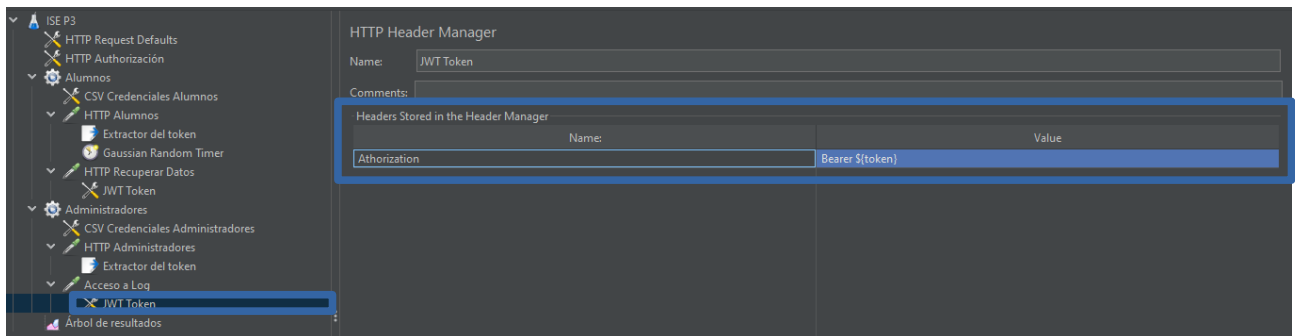


Administradores

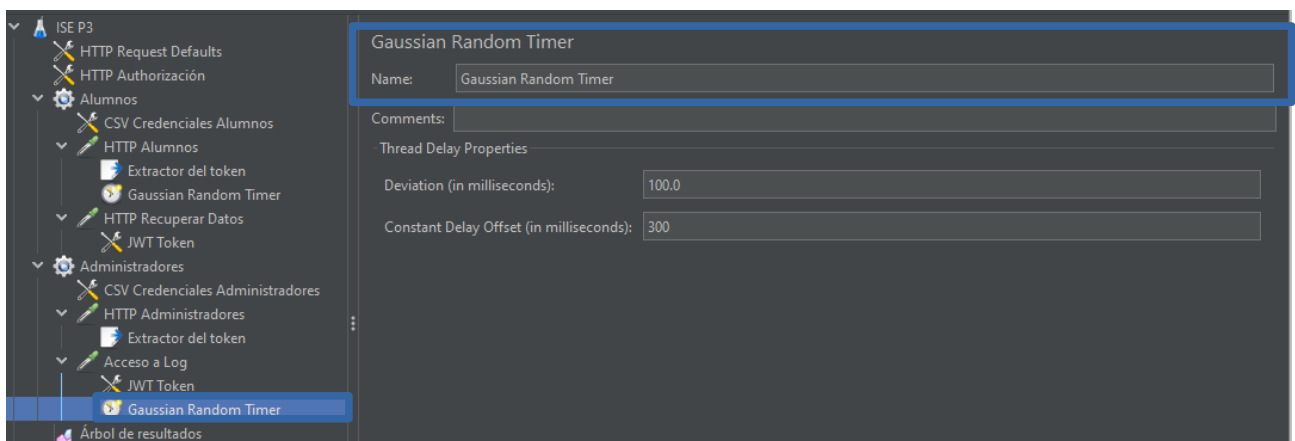
10. Configuramos el muestreador de acceso a log indicando el fichero **apiAlumnos.log** del directorio de la práctica para mostrar el log de los accesos al sistema. Además, añadimos al mismo un gestor de cabecera igual que el del punto 8 para la petición GET de alumnos así como un temporizador del mismo tipo.



Muestreador de acceso a log

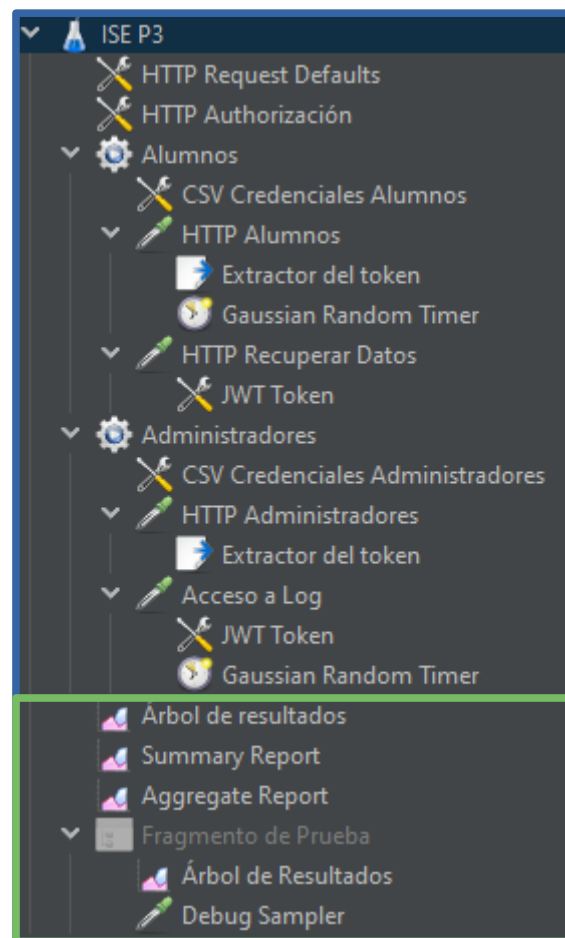


Gestor de cabecera



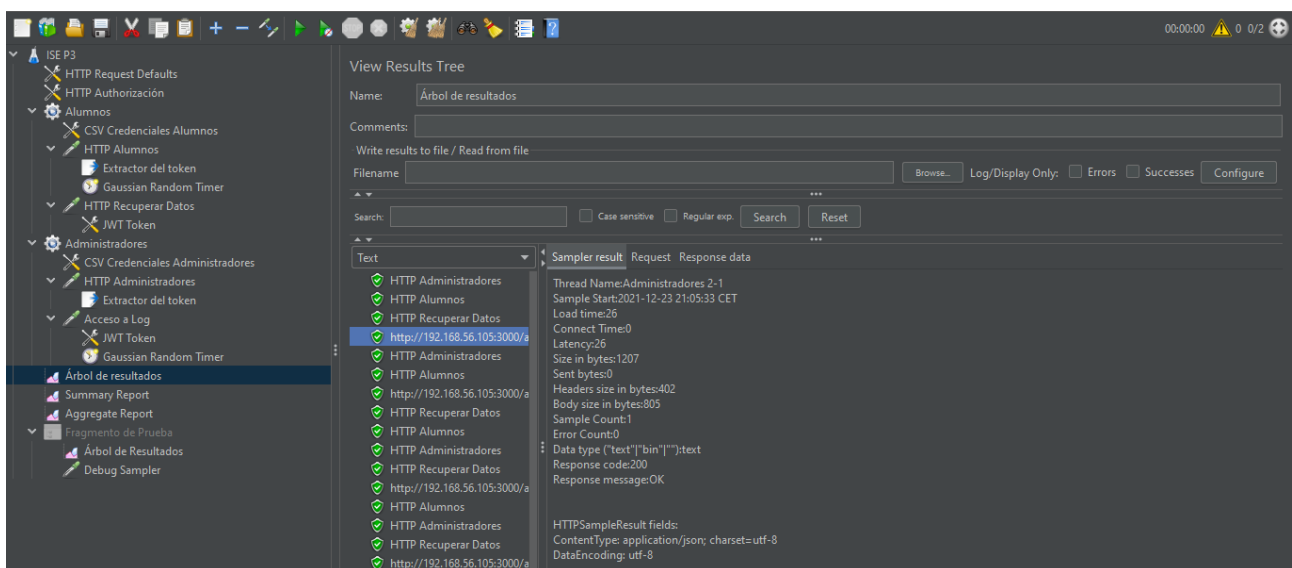
Temporizador

11. Estructura final tras añadir algunos elementos para mostrar los resultados (indicados en verde).

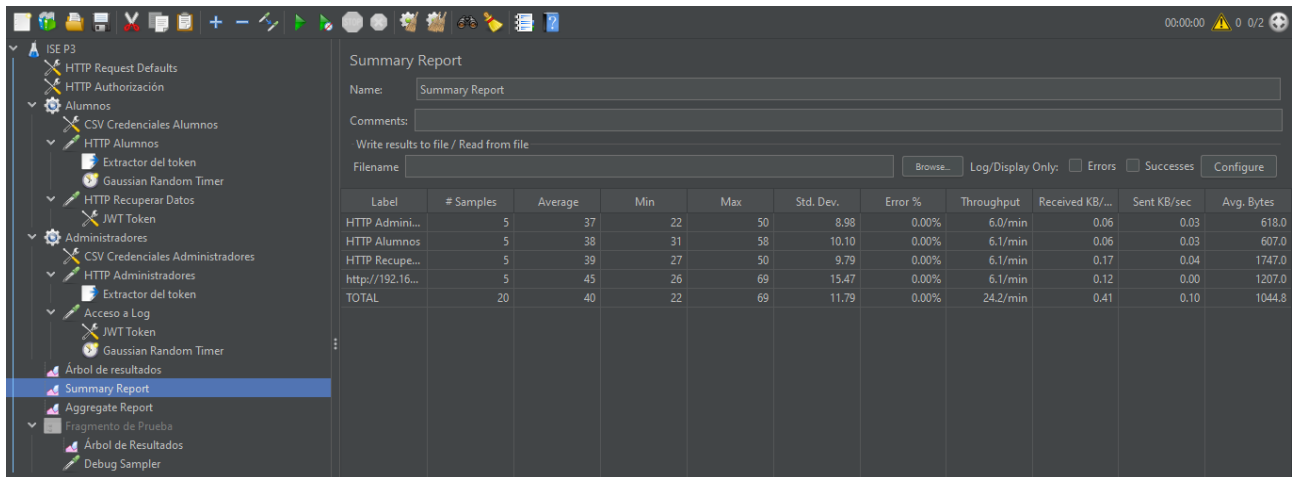


Estructura final

13. Muestra de los resultados obtenidos tras la ejecución del plan.



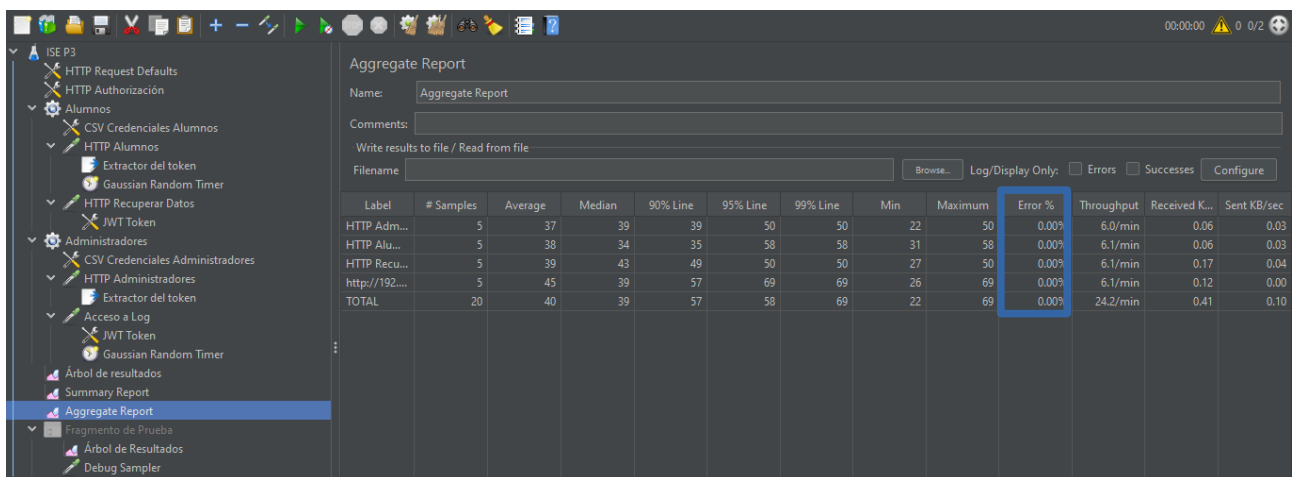
Árbol de resultados



The screenshot shows the JMeter Summary Report. The left sidebar lists various test components, with 'Summary Report' selected. The main panel displays a table with performance metrics for different test elements. All error percentages are 0.00%.

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/sec	Sent KB/sec	Avg. Bytes
HTTP Admini...	5	37	22	50	8.98	0.00%	6.0/min	0.06	0.03	618.0
HTTP Alumnos	5	38	31	58	10.10	0.00%	6.1/min	0.06	0.03	607.0
HTTP Recupe...	5	39	27	50	9.79	0.00%	6.1/min	0.17	0.04	1747.0
http://192.16...	5	45	26	69	15.47	0.00%	6.1/min	0.12	0.00	1207.0
TOTAL	20	40	22	69	11.79	0.00%	24.2/min	0.41	0.10	1044.8

Reporte resumen



The screenshot shows the JMeter Aggregate Report. The left sidebar lists various test components, with 'Aggregate Report' selected. The main panel displays a table with performance metrics for different test elements. All error percentages are 0.00%.

Label	# Samples	Average	Median	90% Line	95% Line	99% Line	Min	Maximum	Error %	Throughput	Received K...	Sent KB/sec
HTTP Adm...	5	37	39	39	50	50	22	50	0.00%	6.0/min	0.06	0.03
HTTP Alu...	5	38	34	35	58	58	31	58	0.00%	6.1/min	0.06	0.03
HTTP Recu...	5	39	43	49	50	50	27	50	0.00%	6.1/min	0.17	0.04
http://192...	5	45	39	57	69	69	26	69	0.00%	6.1/min	0.12	0.00
TOTAL	20	40	39	57	58	69	22	69	0.00%	24.2/min	0.41	0.10

Reporte agregado

Como se puede observar en las dos imágenes anteriores, indicado en verde, el porcentaje de error de cada petición es del 0,00%.

```
nodejs_1 | POST /api/v1/auth/login 200 27.909 ms - 196
nodejs_1 | POST /api/v1/auth/login 200 16.570 ms - 185
nodejs_1 | GET /api/v1/alumnos/alumno/minervamcneil@tropoli.com 200 57.749 ms - 1316
nodejs_1 | GET /api/v1/alumnos/alumno/deborawalker%40tropoli.com 200 28.468 ms - 805
```

Mensajes del servidor (Ubuntu Server)

4. Referencias

Phoronix y Apache Benchmark

- [Michael Larabel] phoronix.com: <https://www.phoronix.com/scan.php?page=article&item=docker-phoronix-pts&num=1>
- [Michael Larabel] phoronix.com: https://www.phoronix.com/scan.php?page=news_item&px=NzU2MA
- phoronix-test-suite.com: <https://www.phoronix-test-suite.com/?k=phoromatic>
- [bladernr] wiki.ubuntu.com: <https://wiki.ubuntu.com/PhoronixTestSuite#installing>
- openbenchmarking.org: <https://openbenchmarking.org/>
- openbenchmarking.org: <https://openbenchmarking.org/test/pts/sudokut>
- openbenchmarking.org: <https://openbenchmarking.org/test/pts/cachebench>
- [Michael Larabel] phoronix.com: https://www.phoronix.com/scan.php?page=news_item&px=Most-Downloaded-Benchmarks
- ArsTech.net: <https://arstech.net/phoronix-test-suite/>
- apachelounge.com: <https://www.apachelounge.com/download/>

JMeter

- jmeter.apache.com: https://jmeter.apache.org/download_jmeter.cgi
- jmeter.apache.org: <https://jmeter.apache.org/>
- [David Palomar] Repositorio de la P4: <https://github.com/davidPalomar-ugr/iseP4JMeter>
- oracle.com: <https://www.oracle.com/java/technologies/downloads/#jdk17-windows>
- [Dimitri T] StackExchange.com: <https://sqa.stackexchange.com/questions/47774/jmeter-illegalaccesserror>
- [Jérôme Loisel]: OctoPerf.com: <https://octoperf.com/blog/2018/03/29/jmeter-tutorial/#overview>