

INGENIERÍA DE SERVIDORES (2021-2022)
GRADO EN INGENIERÍA INFORMÁTICA
UNIVERSIDAD DE GRANADA

Memoria Práctica 4

Adrián Acosa Sánchez

29 de diciembre de 2021

Índice

1	Ejercicio 1: Phoronix	3
1.1	Instalación de Phoronix	3
1.1.1	Instalación de Phoronix en Ubuntu	3
1.1.2	Lista de los Benchmarks disponibles en Phoronix (Ubuntu)	4
1.1.3	Ejecución de Benchmarks en Ubuntu	5
1.1.4	Instalación de Phoronix en CentOS	11
1.1.5	Lista de los Benchmarks disponibles en Phoronix (CentOS)	13
1.1.6	Ejecución de Benchmarks en centos	14
2	Ejercicio 2: JMeter	19
2.1	Instalación de JMeter (Ubuntu)	19
2.2	Instalación de JMeter en nuestro Host	20
2.2.1	Parametrizar el Host y el Puerto en el Test Plan	21
2.2.2	Hacer dos grupos de hebras distintos para alumnos y administradores para simular el acceso concurrente	22
2.2.3	Extracción del token JWT con expresiones regulares	26
2.2.4	Esperas aleatorias a cada grupo de hebras	27
2.2.5	Peticiones HTTP de los login de Alumno y Administrador	28
2.2.6	Muestreo para simular el acceso de administradores	29
3	Bibliografía	30

1. Ejercicio 1: Phoronix

1.1. Instalación de Phoronix

1.1.1. Instalación de Phoronix en Ubuntu

Para descargar Phoronix en ubuntu tenemos que usar el comando `wget` para descargarlo de la página oficial:

```
adrianas@ubuntu:~$ wget http://phoronix-test-suite.com/releases/repo/pts.debian/files/phoronix-test-suite_7.8.0_all.deb
--2021-12-15 14:26:04-- 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  695KB/s   in 0.8s

2021-12-15 14:26:05 (695 KB/s) - 'phoronix-test-suite_7.8.0_all.deb' saved [555634/555634]

adrianas@ubuntu:~$ _
```

Figura 1: Ejecución del comando `wget` para descargar Phoronix

Para poder instalar archivos `.deb` tenemos que antes instalar una herramienta llamada *gdebi* – *core* de los repositorios oficiales de Ubuntu:

```
adrianas@ubuntu:~$ sudo apt-get install gdebi-core
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following NEW packages will be installed:
  gdebi-core
0 upgraded, 1 newly installed, 0 to remove and 130 not upgraded.
Need to get 116 kB of archives.
After this operation, 876 kB of additional disk space will be used.
Get:1 http://es.archive.ubuntu.com/ubuntu focal/universe amd64 gdebi-core all 0.9.5.7+nmu3 [116 kB]
Fetched 116 kB in 0s (341 kB/s)
Selecting previously unselected package gdebi-core.
(Reading database ... 68441 files and directories currently installed.)
Preparing to unpack .../gdebi-core_0.9.5.7+nmu3_all.deb ...
Unpacking gdebi-core (0.9.5.7+nmu3) ...
Setting up gdebi-core (0.9.5.7+nmu3) ...
Processing triggers for man-db (2.9.1-1) ...
adrianas@ubuntu:~$
```

Figura 2: Instalacion de la herramienta para instalar paquetes `.deb`

Una vez tenemos el archivo `.deb` y la herramienta que nos permite instalar archivos con esta extensión, procedemos a instalar el paquete de phoronix:

```
sudo gdebi phoronix-test-suite_7.8.0_all.deb
```

1.1.2. Lista de los Benchmarks disponibles en Phoronix (Ubuntu)

Para poder ver la lista de benchmarks disponibles con el paquete phoronix, ejecutamos el siguiente comando:

```
phoronix-test-suite list-tests
```

```
pts/caffe - Caffe System
pts/cassandra - Apache Cassandra System
pts/chia-vdf - Chia Blockchain VDF Processor
pts/civilization-vi - Civilization VI Graphics
pts/cl-mem - cl-mem Graphics
pts/clomp - CLOMP Processor
pts/cloudsuite-da - CloudSuite Data Analytics System
pts/cloudsuite-ga - CloudSuite Graph Analytics System
pts/cloudsuite-ma - CloudSuite In-Memory Analytics System
pts/cloudsuite-ms - CloudSuite Media Streaming System
pts/cloudsuite-ws - CloudSuite Web Serving System
pts/cloverleaf - CloverLeaf Processor
pts/clpeak - clpeak System
pts/coh2 - Company of Heroes 2 Graphics
pts/cond-cl - CoMD OpenCL System
pts/compilebench - Compile Bench Disk
pts/compress-7zip - 7-Zip Compression Processor
pts/compress-gzip - Gzip Compression Processor
pts/compress-lz4 - LZ4 Compression Processor
pts/compress-pbzip2 - Parallel BZIP2 Compression Processor
pts/compress-rar - RAR Compression System
pts/compress-xz - XZ Compression Processor
pts/compress-zstd - Zstd Compression Processor
pts/core-latency - Core-Latency Processor
pts/coremark - Coremark Processor
pts/couchdb - Apache CouchDB System
pts/cp2k - CP2K Molecular Dynamics Processor
pts/cpp-perf-bench - CppPerformanceBenchmarks System
pts/cpuminer-opt - Cpuminer-Opt Processor
pts/crafty - Crafty Processor
pts/cryptopp - Crypto++ Processor
pts/csgo - Counter-Strike: Global Offensive Graphics
pts/ctx-clock - ctx_clock System
pts/cuda-mini-nbody - CUDA Mini-Nbody Graphics
pts/cyclictst - Cyclictst System
pts/cython-bench - Cython Benchmark Processor
adrianas@ubuntu:~$ phoronix-test-suite list-tests | more_
```

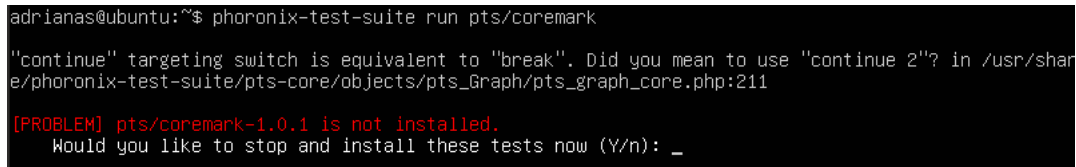
Figura 3: Listado de todos los benchmark proporcionados por Phoronix

1.1.3. Ejecución de Benchmarks en Ubuntu

Para poder ejecutar un benchmark de los proporcionados por Phoronix, tenemos que ejecutar el siguiente comando:

```
phoronix-test-suite run pts/coremark
```

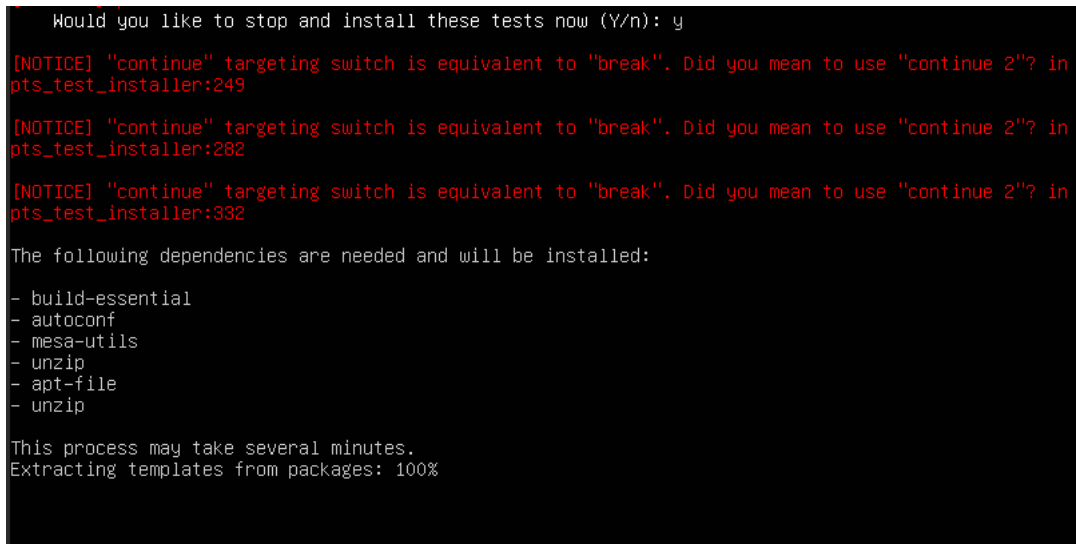
Cuando ejecutemos un benchmark, si este no está instalado te pedirá que lo instales. Así que le decimos que sí como en la siguiente imagen:



```
adrianas@ubuntu:~$ phoronix-test-suite run pts/coremark
"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/coremark-1.0.1 is not installed.
Would you like to stop and install these tests now (Y/n): _
```

Figura 4: Ejecución de un benchmark en Ubuntu

También nos pedirá que instalemos sus dependencias necesarias para la correcta ejecución del benchmark:



```
Would you like to stop and install these tests now (Y/n): y
[NOTICE] "continue" targeting switch is equivalent to "break". Did you mean to use "continue 2"? in pts_test_installer:249
[NOTICE] "continue" targeting switch is equivalent to "break". Did you mean to use "continue 2"? in pts_test_installer:282
[NOTICE] "continue" targeting switch is equivalent to "break". Did you mean to use "continue 2"? in pts_test_installer:332
The following dependencies are needed and will be installed:
- build-essential
- autoconf
- mesa-utils
- unzip
- apt-file
- unzip
This process may take several minutes.
Extracting templates from packages: 100%
```

Figura 5: Dependencias del benchmark

Una vez instalado nos enseñará información sobre nuestro hardware y empezará a preguntarnos si queremos guardar los resultados cuando termine de ejecutar dicho benchmark. En mi caso le digo que sí y le pongo un nombre (la descripción la dejo vacía porque es una prueba):

```

"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:      AMD Ryzen 5 3600X 6-Core @ 3.79GHz
  Core Count:    1
  Extensions:    SSE 4.2 + AVX2 + AVX + RDRAND + FSGSBASE
  Cache Size:    512 KB
  Microcode:     0x6000626

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:           21GB VBOX HDD
  File-System:    ext4
  Mount Options:  relatime 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:        usercopy/swaps barriers and __user pointer sanitization + Full AMD retpoline
STIBP: disabled RSB filling Protection

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

```

Figura 6: Información adicional sobre la ejecución del Benchmark

Una vez terminamos de completar toda la información pedida para la ejecución del benchmark, se empezará a ejecutar y cuando termine nos preguntará si queremos ver los resultados:

```
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:

[NOTICE] "continue" targeting switch is equivalent to "break". Did you mean to use "continue 2"? in
pts_test_installer:249

[NOTICE] "continue" targeting switch is equivalent to "break". Did you mean to use "continue 2"? in
pts_test_installer:282

[NOTICE] "continue" targeting switch is equivalent to "break". Did you mean to use "continue 2"? in
pts_test_installer:332

Coremark 1.0:
pts/coremark-1.0.1
Test 1 of 1
Estimated Trial Run Count: 3
Estimated Time To Completion: 7 Minutes [14:52 UTC]
Started Run 1 @ 14:46:36
Started Run 2 @ 14:46:52
Started Run 3 @ 14:47:15

CoreMark Size 666 - Iterations Per Second:
33573.946617
33428.046131
33707.865169

Average: 33569.95 Iterations/Sec
Deviation: 0.42%

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

Figura 7: Resultados del Benchmark

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

AMD Ryzen 5 3600X 6-Core:

Processor: AMD Ryzen 5 3600X 6-Core @ 3.79GHz (1 Core), Motherboard: Oracle VirtualBox v1.2,
Chipset: Intel 440FX- 82441FX PMC, Memory: 1024MB, Disk: 21GB VBOX HDD, Graphics: VMware SVGA II, A
udio: Intel 82801AA AC 97 Audio, Network: Intel 82540EM Gigabit

OS: Ubuntu 20.04, Kernel: 5.4.0-91-generic (x86_64), Compiler: GCC 9.3.0, File-System: ext4,
Screen Resolution: 2048x2048, System Layer: Oracle VMware

Coremark 1.0
CoreMark Size 666 - Iterations Per Second
Iterations/Sec > Higher Is Better
AMD Ryzen 5 3600X 6-Core .. 33569.95 |=====

Would you like to upload the results to OpenBenchmarking.org (Y/n):
```

Figura 8: Resultados numéricos del Benchmark

Nos pregunta también si queremos verlo en una página web donde aparecen los datos más ordenados. En esa página sale la siguiente información:

test_coremark

OpenBenchmarking.org	Phoronix Test Suite 10.6.1
AMD Ryzen 5 3600X 6-Core @ 3.79GHz (1 Core)	Processor
Oracle VirtualBox v1.2	Motherboard
Intel 440FX- 82441FX PMC	Chipset
1024MB	Memory
21GB 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

Test_coremark Benchmarks

```
--build=x86_64-linux-gnu --disable-vtable-verify
--disable-werror --enable-checking=release
--enable-clocale=gnu --enable-default-pie
--enable-gnu-unique-object
--enable-languages=c,ada,c++,go,brig,d,fortran,objc,obj-c++,gm2
--enable-libstdcxx-debug --enable-libstdcxx-time=yes
--enable-multiarch --enable-multilib --enable-nls
--enable-objc-gc=auto
--enable-offload-targets=nvptx-none=/build/gcc-9-HskZEa/gcc-9-9.3.0/debi
--enable-plugin --enable-shared --enable-threads=posix
--host=x86_64-linux-gnu --program-prefix=x86_64-linux-gnu-
--target=x86_64-linux-gnu --with-abi=m64 --with-arch=32=i686
--with-default-libstdcxx-abi=new
--with-gcc-major-version-only
--with-multilib-list=m32,m64,mx32
--with-target-system-zlib=auto --with-tune=generic
--without-cuda-driver -v
```

System Logs

```
- usercopy/swaps barriers and __user pointer sanitization +
Full AMD retooling STIBP: disabled RSB fillino Protection
```

Coremark

This is a test of EEMBC CoreMark processor benchmark. [Learn more via the OpenBenchmarking.org test page.](#)



Figura 9: Resultados del Benchmark en página web

Ejecutamos ahora un benchmark de git:

```
adrianas@ubuntu:~$ phoronix-test-suite run pts/git
"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/git-1.1.0 is not installed.
Would you like to stop and install these tests now (Y/n): y
[NOTICE] "continue" targeting switch is equivalent to "break". Did you mean to use "continue 2"? in pts_test_installer:249
[NOTICE] "continue" targeting switch is equivalent to "break". Did you mean to use "continue 2"? in pts_test_installer:282
[NOTICE] "continue" targeting switch is equivalent to "break". Did you mean to use "continue 2"? in pts_test_installer:332
Phoronix Test Suite v7.8.0
To Install: pts/git-1.1.0
Determining File Requirements .....
Searching Download Caches .....
1 Test To Install
pts/git-1.1.0:
  Test Installation 1 of 1
  Installing Test @ 15:24:40
```

Figura 10: Instalación del Benchmark de Git

```
"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:      AMD Ryzen 5 3600X 6-Core @ 3.79GHz
Core Count:     1
Extensions:     SSE 4.2 + AVX2 + AVX + RDRAND + FSGSBASE
Cache Size:     512 KB
Microcode:      0x6000626

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:           21GB VBOX HDD
File-System:    ext4
Mount Options:  relatime 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:       usercopy/swaps barriers and __user pointer sanitization + Full AMD retpoline
STIBP: disabled RSB filling Protection

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

Figura 11: Ejecución del Benchmark de Git

```

Time To Complete Common Git Commands:
51.222301006317
50.533879041672
50.493579149246

Average: 50.75 Seconds
Deviation: 0.81%

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

AMD Ryzen 5 3600X 6-Core - VMWare SVGA II - Oracle:

Processor: AMD Ryzen 5 3600X 6-Core @ 3.79GHz (1 Core), Motherboard: Oracle VirtualBox v1.2,
Chipset: Intel 440FX- 82441FX PMC, Memory: 1024MB, Disk: 21GB VBOX HDD, Graphics: VMWare SVGA II, A
Audio: Intel 82801AA AC 97 Audio, Network: Intel 82540EM Gigabit

OS: Ubuntu 20.04, Kernel: 5.4.0-91-generic (x86_64), Compiler: GCC 9.3.0, File-System: ext4,
Screen Resolution: 2048x2048, System Layer: Oracle VMWare

Git
Time To Complete Common Git Commands
Seconds < Lower Is Better
AMD Ryzen 5 3600X 6-Core - VMWare SVGA II - Oracle .. 50.75 |=====
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/2112152-F0-TESTGIT6672
adrianas@ubuntu:~$ _

```

Figura 12: Resultados del Benchmark de Git

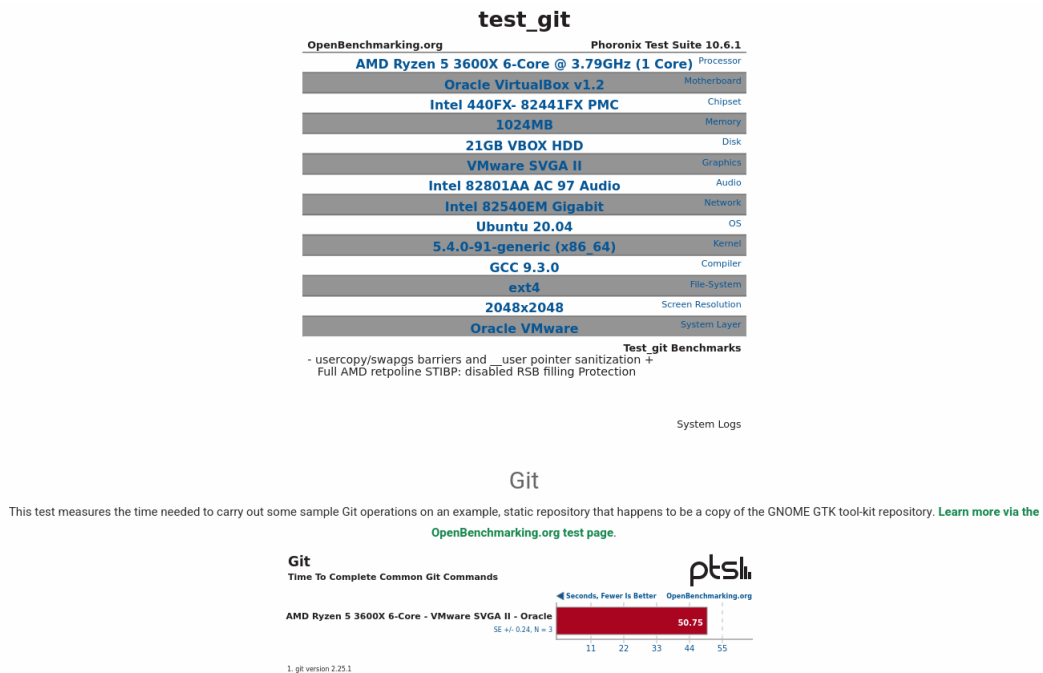


Figura 13: Resultados de Benchmark de Git en página web

1.1.4. Instalación de Phoronix en CentOS

La instalación en CentOS es muy parecida a la de Ubuntu. Comenzamos por instalar los paquetes que aparecen en la siguiente imagen:

```
[adrianas@localhost ~]$ sudo dnf install wget php-cli php-xml
CentOS-8 - AppStream                220 kB/s | 8.2 MB    00:38 A
CentOS-8 - Base                     2.4 MB/s | 3.5 MB    00:01
CentOS-8 - Extras                   23 kB/s | 10 kB     00:00
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
Installing dependencies:
libxslt                x86_64    1.1.32-6.el8                          BaseOS          250 k
php-common              x86_64    7.2.24-1.module_el8.2.0+313+b04d0a66 AppStream       661 k
Enabling module streams:
httpd                  2.4
nginx                  1.14
php                    7.2

Transaction Summary
=====
Install 5 Packages

Total download size: 4.9 M
Installed size: 20 M
Is this ok [y/N]: _
```

Figura 14: Instalación de paquetes necesarios para Phoronix

Tras esto tendremos que descargar el paquete de phoronix con wget desde su repositorio oficial:

```
[adrianas@localhost ~]$ wget https://phoronix-test-suite.com/releases/phoronix-test-suite-8.4.1.tar.gz
--2021-12-15 10:02:10-- 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. 100%[=====>] 853.23K  1.09MB/s   in 0.8s

2021-12-15 10:02:12 (1.09 MB/s) - 'phoronix-test-suite-8.4.1.tar.gz' saved [873706/873706]

[adrianas@localhost ~]$
```

Figura 15: Descarga de Phoronix de su repositorio oficial

Lo descomprimos y ejecutamos el archivo install-sh contenido en la carpeta de Phoronix:

```
phoronix-test-suite/pts-core/external-test-dependencies/xml/lede-packages.xml
phoronix-test-suite/pts-core/external-test-dependencies/xml/pclinuxos-packages.xml
phoronix-test-suite/pts-core/external-test-dependencies/xml/mandrivalinux-packages.xml
phoronix-test-suite/pts-core/external-test-dependencies/xml/brew-packages.xml
phoronix-test-suite/pts-core/external-test-dependencies/xml/void-packages.xml
phoronix-test-suite/pts-core/external-test-dependencies/xml/openindiana-packages.xml
phoronix-test-suite/pts-core/external-test-dependencies/xml/ubuntu-packages.xml
phoronix-test-suite/pts-core/external-test-dependencies/xml/microsoft-packages.xml
phoronix-test-suite/pts-core/external-test-dependencies/xml/alpine-packages.xml
phoronix-test-suite/pts-core/external-test-dependencies/xml/angstrom-packages.xml
phoronix-test-suite/pts-core/external-test-dependencies/xml/termux-packages.xml
phoronix-test-suite/pts-core/external-test-dependencies/xml/opensuse-packages.xml
phoronix-test-suite/pts-core/external-test-dependencies/xml/xsl/
phoronix-test-suite/pts-core/external-test-dependencies/xml/xsl/pts-exdep-viewer.xsl
phoronix-test-suite/pts-core/external-test-dependencies/xml/xsl/pts-generic-exdep-viewer.xsl
phoronix-test-suite/pts-core/external-test-dependencies/xml/opensolaris-packages.xml
phoronix-test-suite/pts-core/external-test-dependencies/xml/gentoo-packages.xml
phoronix-test-suite/pts-core/external-test-dependencies/xml/optware-packages.xml
phoronix-test-suite/pts-core/external-test-dependencies/xml/midnightbsd-packages.xml
phoronix-test-suite/pts-core/external-test-dependencies/xml/generic-packages.xml
phoronix-test-suite/pts-core/external-test-dependencies/xml/zenwalk-packages.xml
phoronix-test-suite/pts-core/external-test-dependencies/xml/pardus-packages.xml
phoronix-test-suite/pts-core/external-test-dependencies/xml/solus-packages.xml
phoronix-test-suite/pts-core/hooks/
phoronix-test-suite/pts-core/hooks/startup/
phoronix-test-suite/pts-core/hooks/startup/template.sh
phoronix-test-suite/pts-core/openbenchmarking.org/
phoronix-test-suite/pts-core/openbenchmarking.org/schemas/
phoronix-test-suite/pts-core/openbenchmarking.org/schemas/test-profile-downloads.xsd
phoronix-test-suite/pts-core/openbenchmarking.org/schemas/test-profile.xsd
phoronix-test-suite/pts-core/openbenchmarking.org/schemas/test-suite.xsd
phoronix-test-suite/pts-core/openbenchmarking.org/schemas/results-parser.xsd
phoronix-test-suite/pts-core/openbenchmarking.org/schemas/types.xsd
phoronix-test-suite/pts-core/openbenchmarking.org/schemas/result-file.xsd
phoronix-test-suite/pts-core/openbenchmarking.org/openbenchmarking-mime.xml
phoronix-test-suite/phoronix-test-suite.bat
adrianas@localhost ~$ tar xvfz phoronix-test-suite-8.4.1.tar.gz _
```

Figura 16: Descompresión del archivo descargado de Phoronix

```
adrianas@localhost ~$ cd phoronix-test-suite/
adrianas@localhost phoronix-test-suite$ ls
AUTHORS      deploy      install-sh  pts-core
ChangeLog    documentation  phoronix-test-suite  README.md
COPYING      install.bat  phoronix-test-suite.bat  release-highlights.md
adrianas@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/
adrianas@localhost phoronix-test-suite$
```

Figura 17: Ejecución del instalador de Phoronix

1.1.5. Lista de los Benchmarks disponibles en Phoronix (CentOS)

Una vez ejecutado todo lo anterior procedemos a listar todos los Benchmarks disponibles en CentOS

```
[adrianas@localhost phoronix-test-suite]$ phoronix-test-suite list-available-test

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.
SQLite3    SQLite3 is required when running a Phoromatic server.
POSIX      POSIX support is highly recommended.

[adrianas@localhost phoronix-test-suite]$ _
```

Figura 18: Error de listados Benchmarks disponibles de Phoronix en CentOS

Pero nos da un error ya que no tenemos todas las dependencias necesarias para ejecutarlos. Luego instalamos las dependencias como aparece en la siguiente imagen:

```
[adrianas@localhost phoronix-test-suite]$ sudo dnf install php-json.x86_64
Last metadata expiration check: 0:04:37 ago on Wed 15 Dec 2021 09:59:56 AM EST.
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]: _
```

Figura 19: Instalación de dependencias necesarias

Una vez hecho esto ya podremos listar todos los Benchmarks disponibles con la siguiente orden:

```
phoronix-test-suite list-available-tests
```

1.1.6. Ejecución de Benchmarks en centos

Para comparar los rendimientos entre ambos sistemas operativos voy a ejecutar los mismos test que en Ubuntu. En las siguientes fotos aparecen los datos de ambos Benchmarks: En mi caso, para poder ejecutar el benchmark coremark tuve problemas de dependencias para poder ejecutarlo. Así que tuve que habilitar los PowerTools de los repositorios de CentOS para poder instalar dichas dependencias:

```
[adrianas@localhost phoronix-test-suite]$ phoronix-test-suite run pts/coremark

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.

[PROBLEM] pts/coremark-1.0.1 is not installed.
Would you like to stop and install these tests now (Y/n): y

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.
Error: Unable to find a match: glibc-static
Last metadata expiration check: 0:12:28 ago on Wed 15 Dec 2021 09:59:56 AM EST.
No match for argument: glibc-static

There are dependencies still missing from the system:
- Compiler / Development Libraries

1: Ignore missing dependencies and proceed with installation.
2: Skip installing the tests with missing dependencies.
3: Re-attempt to install the missing dependencies.
4: Quit the current Phoronix Test Suite process.
Missing dependencies action: _
```

Figura 20: Dependencias de pts/coremark

```
[adrianas@localhost phoronix-test-suite]$ sudo dnf config-manager --enable PowerTools
```

Figura 21: Habilitando los PowerTools de yum

Una vez hecho esto, comenzamos con la ejecución de ambos benchmarks. En primer lugar el benchmark de coremark:

```
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 CentOS Linux 8 via the Phoronix Test Suite.
New Description:

Coremark 1.0:
  pts/coremark-1.0.1
  Test 1 of 1
  Estimated Trial Run Count: 3
  Estimated Time To Completion: 7 Minutes [10:20 EST]
    Started Run 1 @ 10:14:00
    Started Run 2 @ 10:14:22
    Started Run 3 @ 10:14:44

  CoreMark Size 666 - Iterations Per Second:
    33694.614477
    33560.800984
    33978.933062

  Average: 33744.78 Iterations/Sec
  Deviation: 0.63%

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

Figura 22: Ejecución del benchmark coremark

```
Do you want to view the text results of the testing (Y/n): y
test_coremark
Oracle VMware testing on CentOS Linux 8 via the Phoronix Test Suite.

AMD Ryzen 5 3600X 6-Core:

  Processor: AMD Ryzen 5 3600X 6-Core (1 Core), Motherboard: Oracle VirtualBox v1.2, Chipset:
  Intel 440FX 82441FX PMC, Memory: 818MB, Disk: 21GB VBox HDD, Graphics: VMware SVGA II, Audio: Intel
  82801AA AC 97 Audio, Network: Intel 82540EM Gigabit

  OS: CentOS Linux 8, Kernel: 4.18.0-193.el8.x86_64 (x86_64), Compiler: GCC 8.5.0 20210514, Fi
  le-System: xfs, Screen Resolution: 2048x2048, System Layer: Oracle VMware

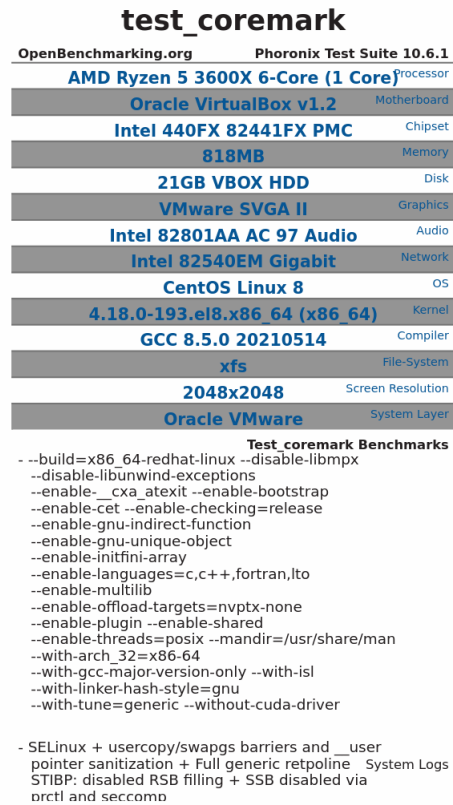
Coremark 1.0
CoreMark Size 666 - Iterations Per Second
Iterations/Sec > Higher Is Better
AMD Ryzen 5 3600X 6-Core .. 33745 |=====

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/2112157-SK-TESTCOREM85

ladrianas@localhost phoronix-test-suite1$ _
```

Figura 23: Resultados del benchmark coremark



Coremark

This is a test of EEMBC CoreMark processor benchmark. [Learn more via the OpenBenchmarking.org test page.](#)



Figura 24: Resultados de la página web del test coremark

Y ahora la ejecución del benchmark de Git:

```
adrianas@ubuntu:~$ phoronix-test-suite run pts/git
"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/git-1.1.0 is not installed.
  Would you like to stop and install these tests now (Y/n): y
[NOTICE] "continue" targeting switch is equivalent to "break". Did you mean to use "continue 2"? in pts_test_installer:249
[NOTICE] "continue" targeting switch is equivalent to "break". Did you mean to use "continue 2"? in pts_test_installer:282
[NOTICE] "continue" targeting switch is equivalent to "break". Did you mean to use "continue 2"? in pts_test_installer:332
Phoronix Test Suite v7.8.0

  To Install:   pts/git-1.1.0

  Determining File Requirements .....
  Searching Download Caches .....

  1 Test To Install

pts/git-1.1.0:
  Test Installation 1 of 1
  Installing Test @ 15:24:40
```

Figura 25: Instalación del benchmark de Git

```
Time To Complete Common Git Commands:
53.400220023529
53.244813919067
53.051034021701

Average: 53.23 Seconds
Deviation: 0.33%

Do you want to view the text results of the testing (Y/n): y
test_git
Oracle VMware testing on CentOS Linux 8 via the Phoronix Test Suite.

AMD Ryzen 5 3600X 6-Core - VMware SUGA II - Oracle:

  Processor: AMD Ryzen 5 3600X 6-Core (1 Core), Motherboard: Oracle VirtualBox v1.2, Chipset:
  Intel 440FX 82441FX PMC, Memory: 818MB, Disk: 21GB VBox HDD, Graphics: VMware SUGA II, Audio: Intel
  82801AA AC 97 Audio, Network: Intel 82540EM Gigabit

  OS: CentOS Linux 8, Kernel: 4.18.0-193.el8.x86_64 (x86_64), Compiler: GCC 8.5.0 20210514, Fi
  le-System: xfs, Screen Resolution: 2048x2048, System Layer: Oracle VMware

Git
Time To Complete Common Git Commands
Seconds < Lower Is Better
AMD Ryzen 5 3600X 6-Core - VMware SUGA II - Oracle .. 53.23 |=====

  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/2112156-SK-TESTGIT1032

[adrianas@localhost phoronix-test-suite]$_
```

Figura 26: Ejecución del benchmark de Git

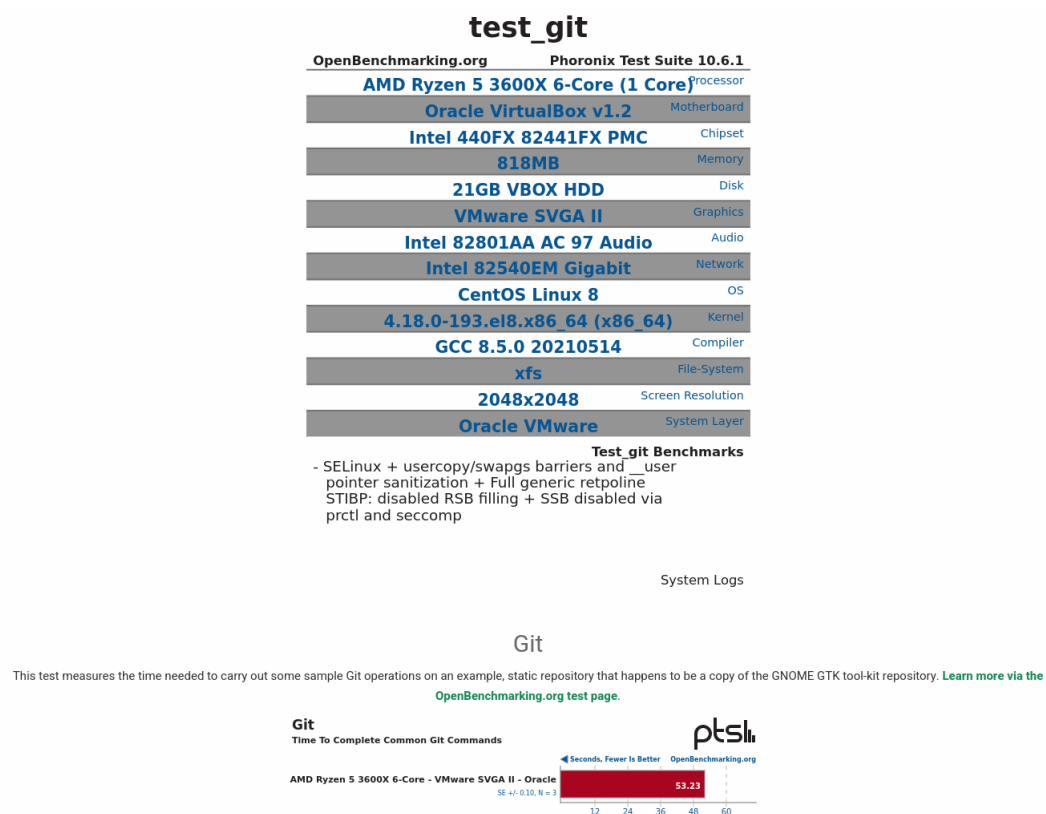


Figura 27: Resultados de la página web del test Git

Los resultados para coremark son para CentOS 33745 y para Ubuntu 33569. Los resultados para Git son para CentOS 53.23 y para Ubuntu 50.75. Como podemos ver, no hay una diferencia clara entre ambos sistemas ya que los valores de los benchmarks tienen diferencias de muy poca puntuación (tiene sentido ya que se ejecutan sobre el mismo Host).

2. Ejercicio 2: JMeter

2.1. Instalación de JMeter (Ubuntu)

Antes de instalar JMeter tenemos que instalar docker y docker-compose para poder descargar el repositorio que se proporciona:

```
sudo apt-get install docker docker-compose
```

Ahora vamos a descargar el repositorio de JMeter proporcionado para hacer la práctica con el siguiente comando:



```
adrianas@ubuntu:~$ git clone http://github.com/davidPalomar-ugr/iseP4JMeter
Clonando en 'iseP4JMeter'...
warning: redirigiendo a 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
Recibiendo objetos: 100% (3797/3797), 7.79 MiB | 2.84 MiB/s, listo.
Resolviendo deltas: 100% (715/715), listo.
adrianas@ubuntu:~$
```

Figura 28: Descarga de la carpeta con JMeter

Una vez descargada la carpeta, iniciamos el servicio con el comando:

```
sudo docker-compose up
sudo docker-compose down
```

up para levantar el servicio y down para pararlo. La ejecución de docker ensucia mucho la pantalla con muchos outputs. Para que no que no imprima nada por pantalla, ejecutamos el siguiente comando:



```
adrianas@ubuntu:~/iseP4JMeter$ sudo docker-compose up 2> cerr.txt > cout.txt &
```

Figura 29: Levantar el servicio sin outputs

La primera vez tardará un rato ya que tendrá que descargar todos los contenedores. Una vez termine, tendremos que habilitar el puerto 3000 ya que el servicio lo requiere. Esto, como ya sabemos, lo haremos con el comando:

```
ufw enable
ufw allow 3000/tcp
```

Para ver si todo se ha iniciado correctamente, lo comprobamos en un navegador poniendo la IP de nuestro servidor seguido del puerto 3000 y tendría que salir un resultado como el siguiente:



Figura 30: Pantalla de inicio del servicio

2.2. Instalación de JMeter en nuestro Host

El único requisito que tiene JMeter para permitir su ejecución es tener Java instalado y lo podemos comprobar con la ejecución del comando:

```
java --version
```

Una vez comprobamos que tenemos Java instalado, nos vamos a la página oficial de JMeter y lo descargamos. Al descomprimirlo nos vamos a su carpeta y ejecutamos el archivo cuyo nombre es ApacheJMeter.jar:

```
adrianas@debian:~/Descargas/apache-jmeter-5.4.1/bin$ java -jar ApacheJMeter.jar
```

Figura 31: Ejecución con Java de JMeter.

Al abrirlo nos aparecerá una ventana como la siguiente:

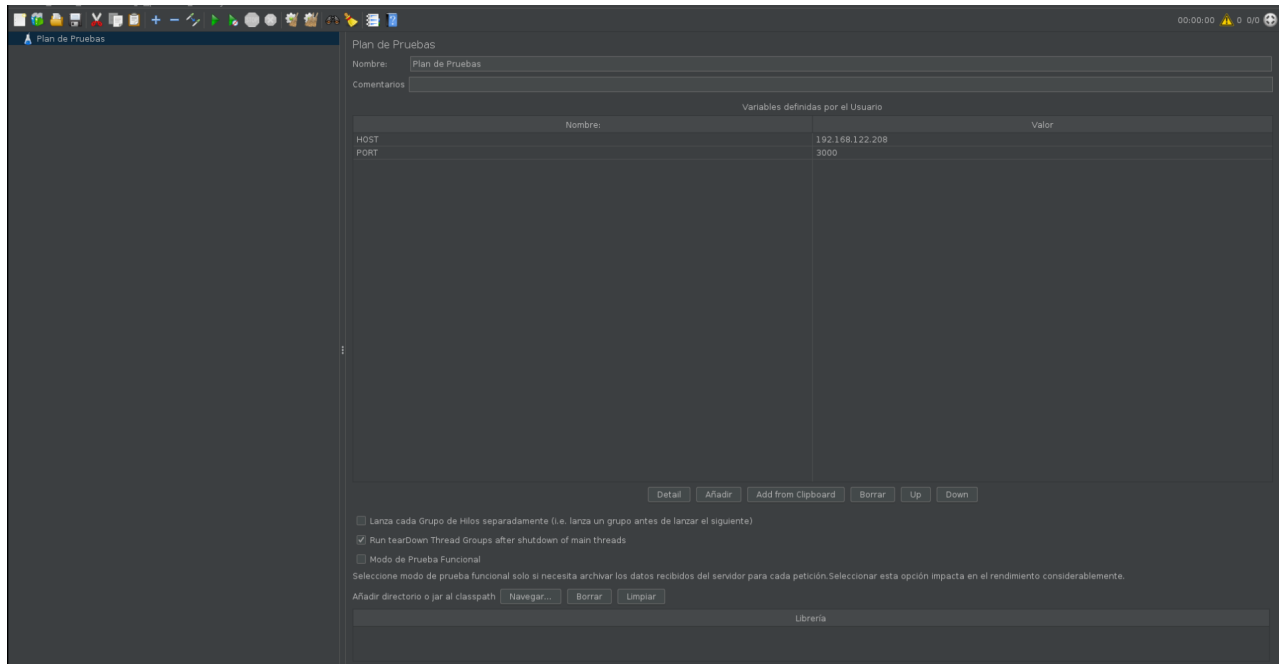


Figura 32: Pantalla inicial de JMeter

2.2.1. Parametrizar el Host y el Puerto en el Test Plan

Para lograr parametrizar el Host y el Puerto tenemos que añadir al nodo principal los datos de nuestra IP y el Puerto:

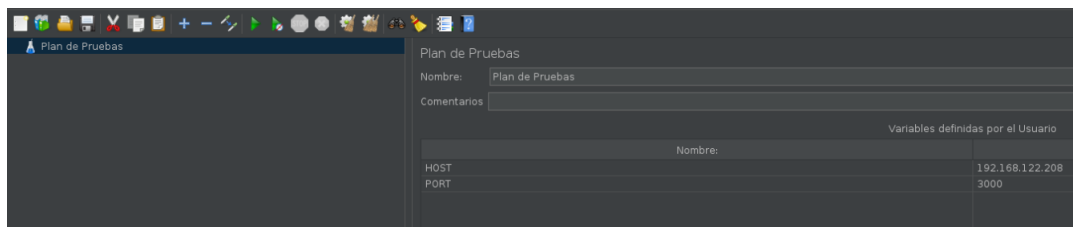


Figura 33: Añadimos las variables de HOST y PORT

2.2.2. Hacer dos grupos de hebras distintos para alumnos y administradores para simular el acceso concurrente

Para ello lo primero que tenemos que hacer es ver los archivos .csv que se encuentran en la carpeta iseP4JMeter/jMeter. Ahí encontraremos los archivos alumnos.csv y administradores.csv. Miramos su contenido y cogemos una cuenta de cada uno de los archivos:

```
adrianas@ubuntu:~/iseP4JMeter$ cd jMeter/
adrianas@ubuntu:~/iseP4JMeter/jMeter$ ls -l
total 536
-rw-r--r-- 1 adrianas adrianas 673 dic 16 16:29 administradores.csv
-rw-r--r-- 1 adrianas adrianas 31922 dic 16 16:29 alumnos.csv
-rw-r--r-- 1 adrianas adrianas 511803 dic 16 16:29 apiAlumnos.log
adrianas@ubuntu:~/iseP4JMeter/jMeter$ cat administradores.csv
login,password
stuartkirby@etsii.ugr.es,tempor
suarezgraves@etsii.ugr.es,laborum
janablake@etsii.ugr.es,excepteur
osbornecantrell@etsii.ugr.es,sint
dickersonlowe@etsii.ugr.es,labore
harrisongomez@etsii.ugr.es,ad
garrettbeck@etsii.ugr.es,occaecat
pottschase@etsii.ugr.es,proident
kaitlinsavage@etsii.ugr.es,qui
buchanancrosby@etsii.ugr.es,irure
annmendoza@etsii.ugr.es,minim
fuentesrandolph@etsii.ugr.es,dolor
cookecooley@etsii.ugr.es,do
hogancarney@etsii.ugr.es,proident
watsonwalters@etsii.ugr.es,nulla
ramonaeaton@etsii.ugr.es,anim
fowlervazquez@etsii.ugr.es,deserunt
lanecurry@etsii.ugr.es,voluptate
bakerreilly@etsii.ugr.es,proident
claricecamacho@etsii.ugr.es,nostrud
adrianas@ubuntu:~/iseP4JMeter/jMeter$
```

Figura 34: Archivo administradores.csv

```

hoodholmes@tropoli.com,eu
hahnroberts@tropoli.com,minim
toniaewing@tropoli.com,anim
vasquezstevens@tropoli.com,non
tommiessolis@tropoli.com,aute
fayehampton@tropoli.com,aliqua
ellisalexander@tropoli.com,dolor
ninamorrison@tropoli.com,cupidatat
keithsantos@tropoli.com,fugiat
velmasutton@tropoli.com,consectetur
tonimcleod@tropoli.com,eiusmod
searsosborne@tropoli.com,sunt
lulabennett@tropoli.com,voluptate
jaynesanchez@tropoli.com,minim
snyderkirk@tropoli.com,magna
vargasdejesus@tropoli.com,ut
ruthiegalloway@tropoli.com,ex
rayalston@tropoli.com,fugiat
haleykirkland@tropoli.com,officia
maestrickland@tropoli.com,deserunt
ashleesloan@tropoli.com,incididunt
cantrellanderson@tropoli.com,velit
gillrice@tropoli.com,deserunt
gingergould@tropoli.com,voluptate
jacquelynbell@tropoli.com,laborum
workmansanford@tropoli.com,sint
watsonpitts@tropoli.com,elit
gallagherreyes@tropoli.com,enim
luciledouglas@tropoli.com,ipsum
rhodasilva@tropoli.com,eiusmod
hortonhayes@tropoli.com,prehenderit
wallerhardy@tropoli.com,laboris
coleknight@tropoli.com,laboris
aldavega@tropoli.com,ea
madeleinerandolph@tropoli.com,dolor
dionnefrazier@tropoli.com,amet
meyerdillard@tropoli.com,incididunt
cherryhodes@tropoli.com,do
pollardsweet@tropoli.com,laborum
altheamadden@tropoli.com,nisi
mcfaddencurry@tropoli.com,consectetur
minervamcneil@tropoli.com,cillum
margretdonovan@tropoli.com,non
hernandezmayer@tropoli.com,nisi
luciariley@tropoli.com,ullamco
lizamonroe@tropoli.com,ut
farrellguthrie@tropoli.com,nostrud
adrianas@ubuntu:~/iseP4JMeter/iMeter$ cat alumnos.csv

```

Figura 35: Archivo alumnos.csv

Una vez tenemos ambas cuentas, vamos a JMeter y añadimos dos grupos de hebras que simularán el acceso de los administradores y de los alumnos.

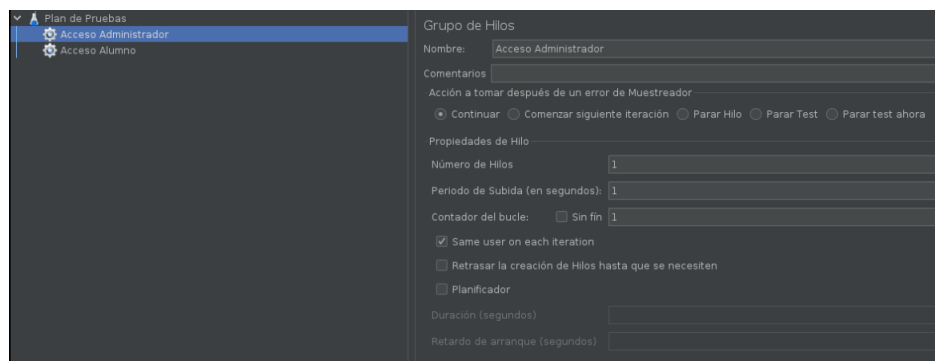


Figura 36: Añadimos dos grupos de hebras

Yo he dejado todo por defecto en cada uno de los grupos de hebras. Añadimos un elemento de configuración HTTP por defecto para que en las peticiones que añadiremos a

continuación tengan la IP y el Puerto configurados por defecto.

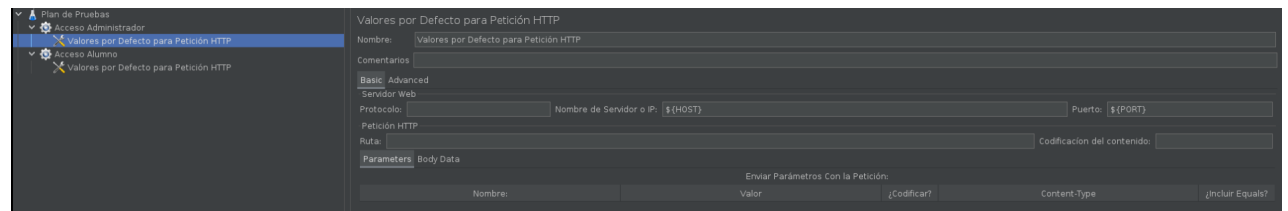


Figura 37: Valores por defecto para las Peticiones HTTP

En la imagen añado un elemento a cada hebra, aunque lo más adecuado es añadirlo en general al plan de pruebas (lo hago más adelante).

Lo siguiente que tenemos que hacer es añadir una petición HTTP a cada una de las hebras para simular un login de un administrador y de un alumno:

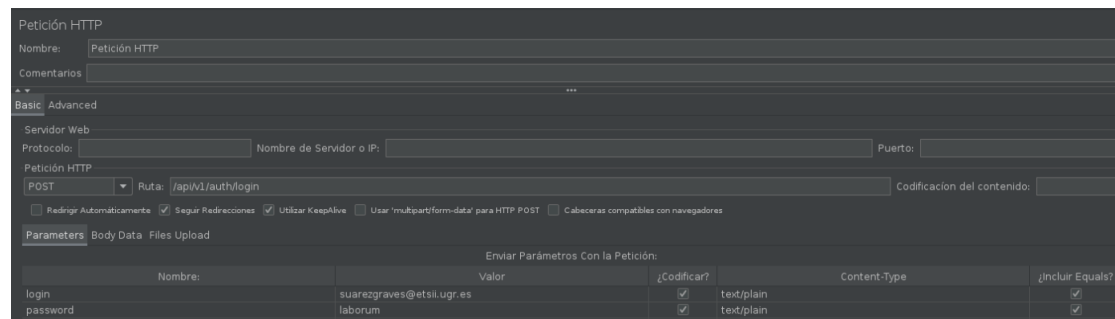


Figura 38: Login de administrador

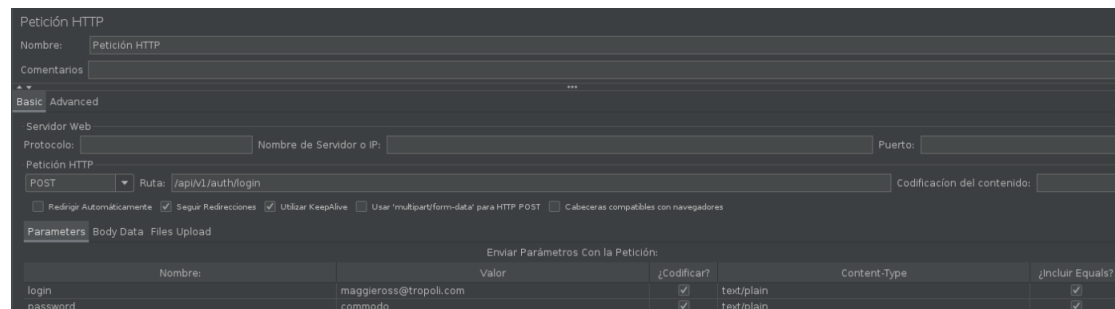


Figura 39: Login de usuario

Para poder ver los resultados de ambas peticiones añadimos un árbol de resultados por defecto y ejecutamos el plan de pruebas:

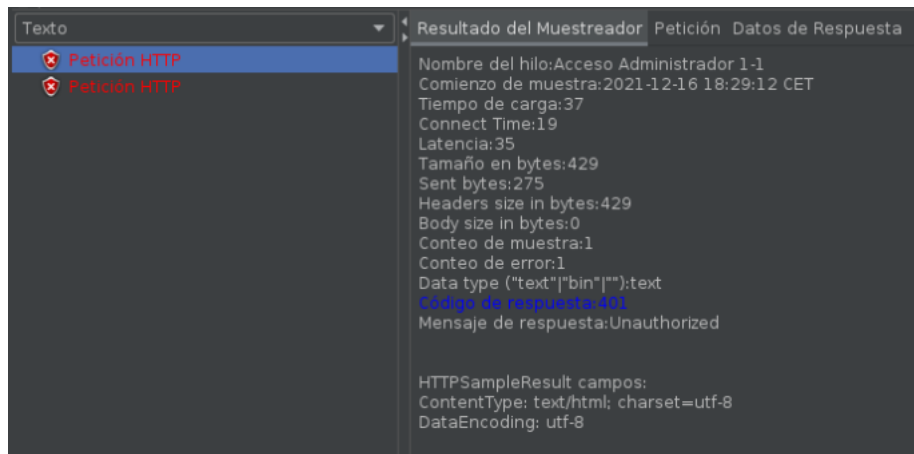


Figura 40: Ejecución del plan de pruebas

Como vemos nos da error en ambos login. Esto es porque no tenemos autorización. El acceso está protegido con un BasicAuth donde el usuario es etsiiApi y la contraseña laApiDeLaETSIIDaLache. Esto aparece en la página de inicio que vimos antes. Para solucionar este error añadimos en cada uno un gestor de autorización HTTP con los datos ya mencionados:

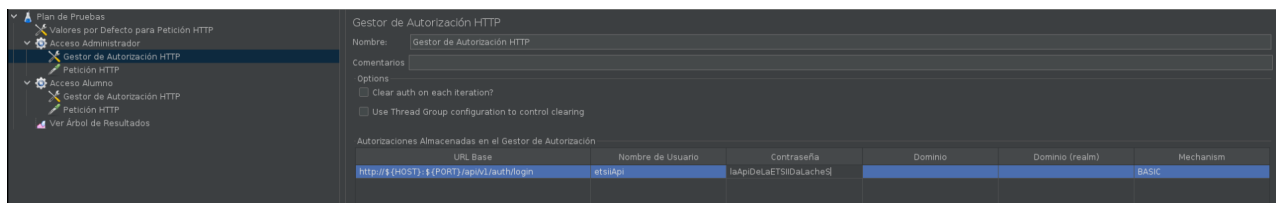


Figura 41: Configuración del gestor de Autorización HTTP

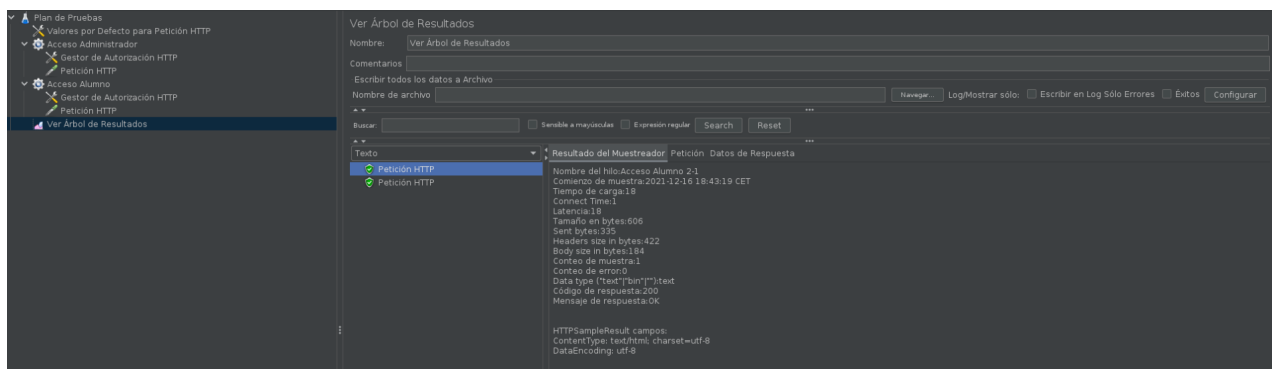


Figura 42: Resultados correctos de los login

2.2.3. Extracción del token JWT con expresiones regulares

Para poder guardar el token que nos devuelven ambos login tenemos que crear un extractor de expresiones regulares como aparece en la siguiente imagen:

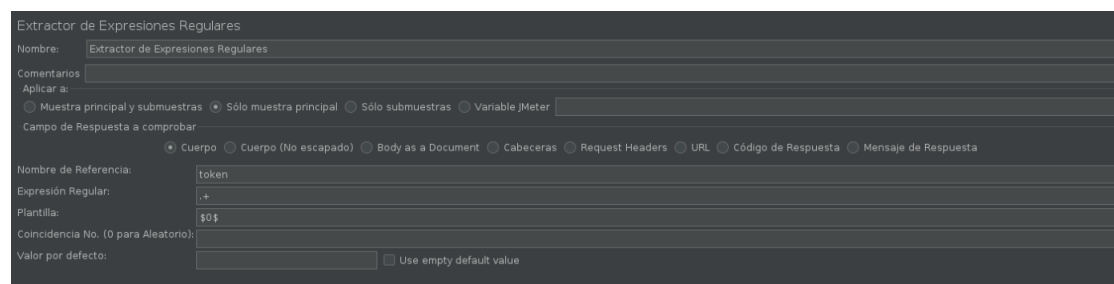


Figura 43: Extractor de expresiones regulares

Para poder recibir la información usando un GET y el token hace falta que creamos otra petición HTTP donde pondremos solo la ruta del correo del alumno que queremos mostrar:

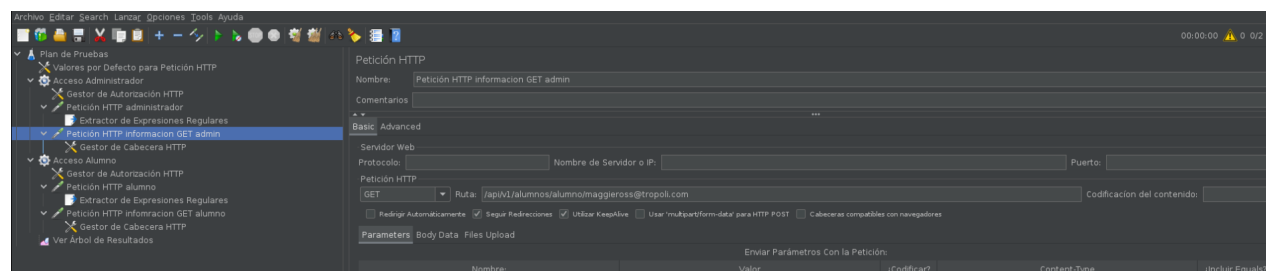


Figura 44: Petición HTTP GET

También tenemos que indicar el método de seguridad usando el token que se ha recibido. Esto se hace con un gestor de cabecera HTTP, el cual se lo pondremos a la petición HTTP GET del administrador.



Figura 45: Gestor de cabecera HTTP

Y podemos ver que funciona correctamente:

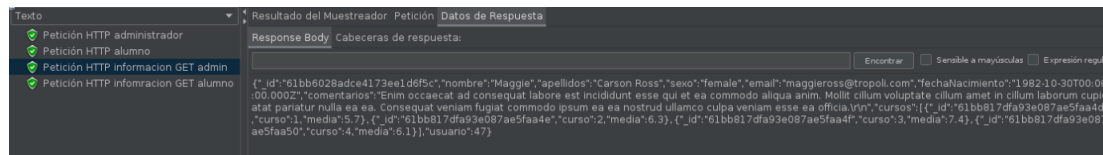


Figura 46: Ejecución del plan de pruebas para comprobar si funciona

Pero para comprobar que funcione correctamente, si le indicamos un correo de un alumno en vez de un administrador pasa lo siguiente:

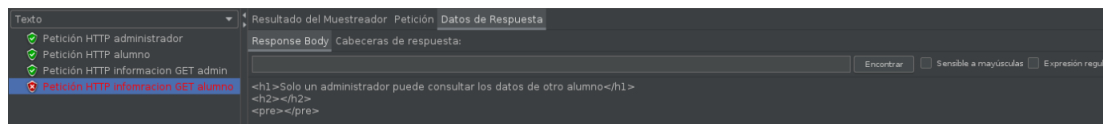


Figura 47: Cambio de correo por el de un alumno

2.2.4. Esperas aleatorias a cada grupo de hebras

Con el objetivo de simular un entorno más realista, le añadimos a cada uno de los grupos de hebras un temporizador aleatorio Gaussiano como se ve en la siguiente imagen:

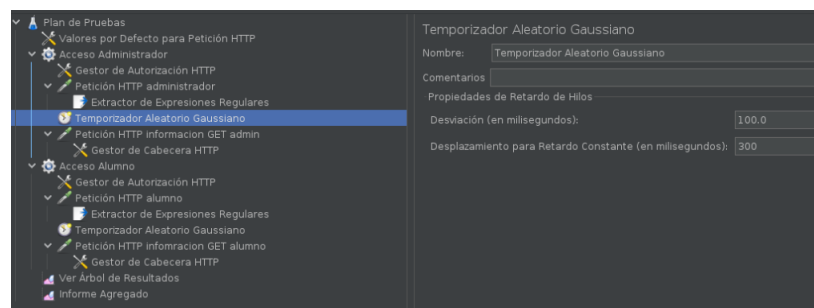


Figura 48: Adición de temporizador aleatorio Gaussiano

2.2.5. Peticiones HTTP de los login de Alumno y Administrador

Para ello lo primero que tenemos que hacer es añadir un elemento llamado CSV Data Set donde se accederá a los usuarios.

En la imagen hay un fallo y es que el archivo no se accede desde donde pongo. Tuve que descargar el repositorio en el host y buscar el archivo manualmente.

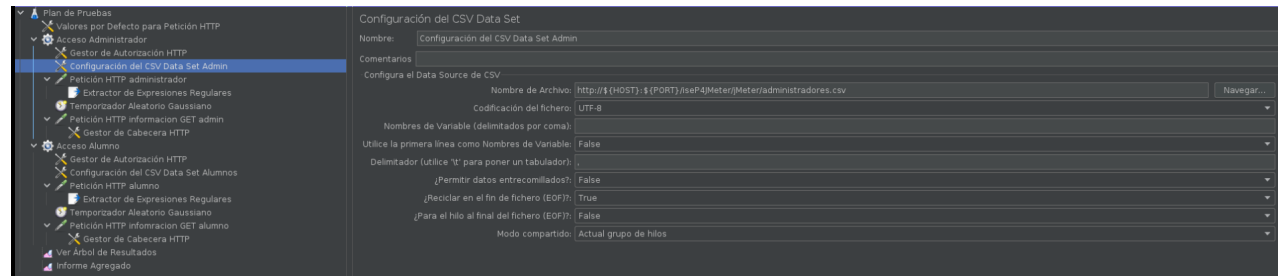


Figura 49: Configuración del CSV Data Set

Y lo mismo para los alumnos. Luego hacemos una petición HTTP GET donde pondremos la siguiente información de login:

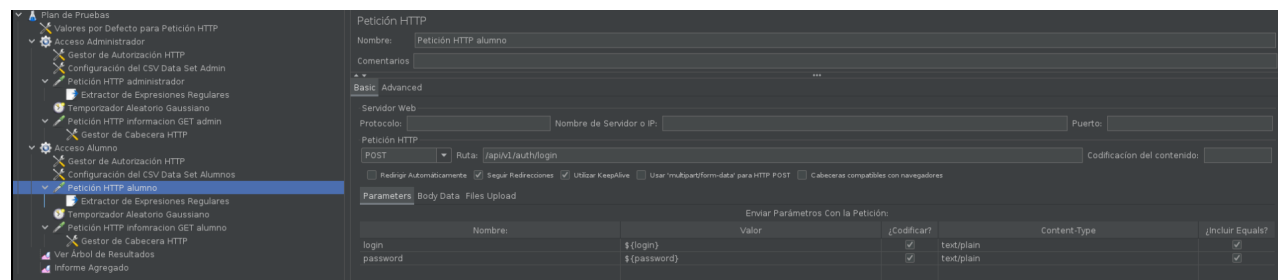


Figura 50: Login de alumnos

Y al probar el plan de tests:

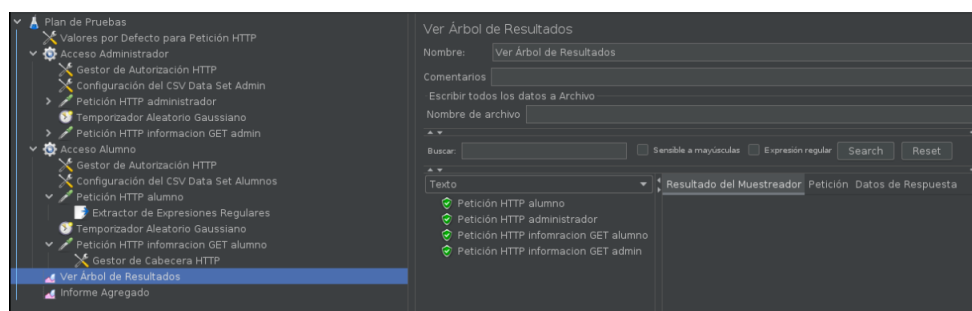


Figura 51: Prueba de logins

2.2.6. Muestreo para simular el acceso de administradores

Lo primero que tenemos que hacer es deshabilitar la petición GET del administrador.

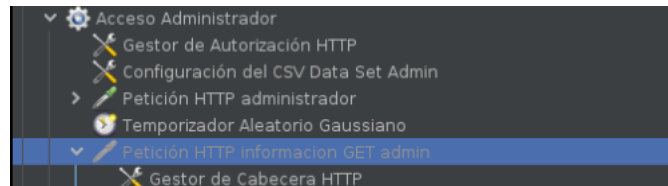


Figura 52: Deshabilitamos la petición GET del administrador

Luego tenemos que añadir un muestreador de Acceso a Log:

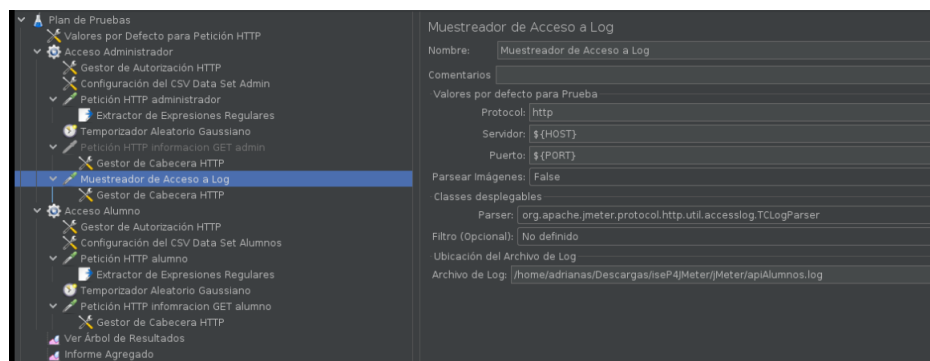


Figura 53: Muestreador de Acceso a Log

Y por último añadimos la cabecera donde le pasamos el token de la sesión de administrador y con esto acabaríamos la práctica. El resultado sería el siguiente:

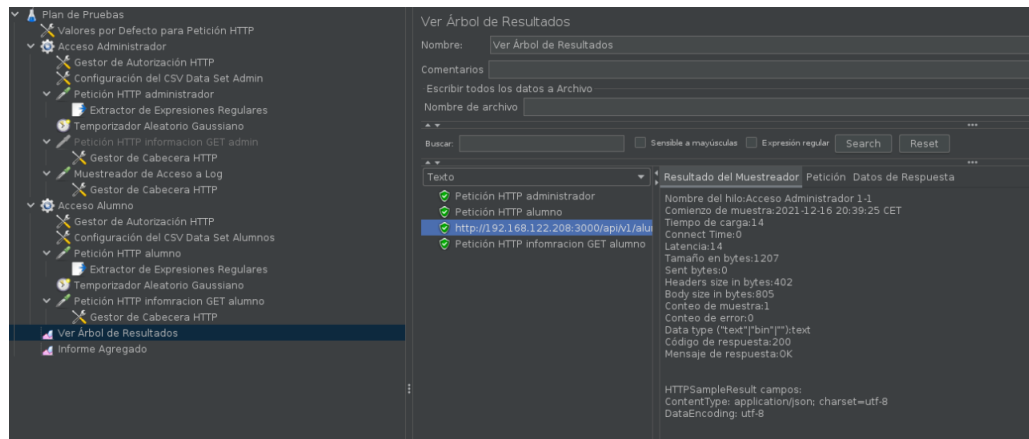


Figura 54: Árbol de resultados final

3. Bibliografía

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

² <https://github.com/phoronix-test-suite/phoronix-test-suite/blob/master/documentation/phoronix-test-suite.md>

³ <https://github.com/davidPalomar-ugr/iseP4JMeter>

¹Descarga de JMeter

²Documentación de Phoronix

³Repositorio proporcionado para la práctica