

# MEMORIA PRACTICA 4 ISE

Javier Ramírez Pulido B2

## PHORONIX Y PRUEBAS

### - UBUNTU:

Lo primero será obtener el paquete de Phoronix con la sentencia:

`$ wget http://phoronix-testsuite.com/releases/repo/pts.debian/files/phoronix-test-suite_8.6.0_all.deb`

```
root@javierrp:/home/javierrp# wget http://phoronix-test-suite.com/releases/repo/pts.debian/files/pho
ronix-test-suite_8.6.0_all.deb
--2020-12-08 23:07:31-- http://phoronix-test-suite.com/releases/repo/pts.debian/files/phoronix-test-
-suite_8.6.0_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: 603468 (589K) [application/x-debian-package]
Saving to: 'phoronix-test-suite_8.6.0_all.deb'

phoronix-test-suite_8.6. 100%[=====] 589.32K 710KB/s in 0.8s
2020-12-08 23:07:32 (710 KB/s) - 'phoronix-test-suite_8.6.0_all.deb' saved [603468/603468]

root@javierrp:/home/javierrp#
```

Utilizamos dkpg para instalar Phoronix-test-suite

`$ dkpg -i phoronix-test-suite_8.6.0_all.deb`

```
root@javierrp:/home/javierrp# dpkg -i phoronix-test-suite_8.6.0_all.deb
Selecting previously unselected package phoronix-test-suite.
(Reading database ... 83782 files and directories currently installed.)
Preparing to unpack phoronix-test-suite_8.6.0_all.deb ...
Unpacking phoronix-test-suite (8.6.0) ...
Setting up phoronix-test-suite (8.6.0) ...
Processing triggers for mime-support (3.64ubuntu1) ...
Processing triggers for man-db (2.9.1-1) ...
root@javierrp:/home/javierrp# _
```

Phoronix cuenta con una extensa lista de diferentes tests que pasaremos a mostrar. Es recomendable pensar bien qué opciones queremos antes de descargarlos porque tenemos que estar seguros de poder encontrarlos en CentOS para comparar resultados.

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

```
javierrp@javierrp:~$ phoronix-test-suite list-available-tests _
```

Y este es un ejemplo de una parte de lo que se muestra:

pts/y-cruncher	- Y-Cruncher	Processor
pts/yafaray	- YafaRay	Processor
system/apache	- Apache Benchmark	
system/blender	- Blender	System
system/blogbench	- BlogBench	Disk
system/clpeak	- clpeak	System
system/compress-lzma	- LZMA Compression	
system/compress-pbzip2	- PBZIP2 Compression	
system/compress-zstd	- Zstd Compression	Processor
system/cryptsetup	- Cryptsetup	System
system/darktable	- Darktable	System
system/dbench	- Dbench	Disk
system/ethminer	- Ethereum Ethminer	System
system/fio	- Flexible IO Tester	Disk
system/gegl	- GEGL	System
system/gimp	- GIMP	System
system/gmic	- G'MIC	System
system/gnupg	- GnuPG	Processor
system/hugin	- Hugin	System
system/inkscape	- Inkscape	System
system/iozone	- IOzone	Disk
system/libreoffice	- LibreOffice	System
system/mpv	- MPV	System
system/ocrmypdf	- OCRMyPDF	System
system/octave-benchmark	- GNU Octave Benchmark	System
system/openssl	- OpenSSL	
system/opm	- Open Porous Media	Processor
system/rawtherapee	- RawTherapee	System
system/redis	- Redis	
system/rsvg	- librsvg	System
system/selenium	- Selenium	System
system/selenium-top-sites	- Time To Load + View Popular Websites	System
system/sqlite	- SQLite	
system/tesseract-ocr	- Tesseract OCR	System
system/wireguard	- WireGuard + Linux Networking Stack Stress Test Netwo	

Existe un test que calcula el tiempo que se tira el sistema resolviendo 100 sudokus llamado “sudokut”, por lo que mostramos la misma lista de antes, pero seleccionando solo aquellos con ese nombre:

```
javierrp@javierrp:~$ phoronix-test-suite list-available-tests | grep sudokut
pts/sudokut - Sudokut Processor
```

Y el otro test estará relacionado con el rendimiento de la criptografía de la máquina. Es se ha escogido al azar de la lista (comprobando que también estará en CentOS) y es *cryptsetup*:

```
javierrp@javierrp:~$ phoronix-test-suite list-available-tests | grep cryptsetup
system/cryptsetup - Cryptsetup System
```

Pasamos a la instalación de los benchmarks seleccionados:

```
javierrp@javierrp:~$ phoronix-test-suite install pts/sudokut

[PROBLEM] Failed to find ZIP support for extracting file: /home/javierrp/.phoronix-test-suite/pts/sudokut-1.0.1.zip. Install PHP ZIP support or the unzip utility.
Updated OpenBenchmarking.org Repository Index
pts: 428 Distinct Tests, 74 Suites

[PROBLEM] Failed to find ZIP support for extracting file: /home/javierrp/.phoronix-test-suite/pts/sudokut-1.0.1.zip. Install PHP ZIP support or the unzip utility.

[PROBLEM] Invalid Argument: pts/sudokut

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

Popular Tests:
  build-mpplayer      botan      postmark    pgbench
  mafft               build-apache  phpbench    hammer
Possible Suggestions:
- pts-desktop-live [Suite]

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>
```

Proceso que nos da un fallo sobre PHP por falta de dependencias que será resuelto con la siguiente orden:

```
javierrp@javierrp:~$ sudo apt install php-zip
```

Y volvemos a intentar instalarlo, ahora sin errores.

```
javierrp@javierrp:~$ phoronix-test-suite install sudokut

Phoronix Test Suite v8.6.0
AN OUTDATED VERSION OF THE PHORONIX TEST SUITE IS INSTALLED.
THE VERSION IN USE IS 8.6.0 (8600), BUT THE LATEST IS PTS-CORE 10000.
VISIT HTTPS://WWW.PHORONIX-TEST-SUITE.COM/ TO UPDATE THIS SOFTWARE.

The following dependencies are needed and will be installed:

- tcl
- mesa-utils
- unzip
- apt-file

This process may take several minutes.
```

```

Setting up libregex-perl (0.36-1) ...
Setting up libdrm-amdgpu1:amd64 (2.4.101-2) ...
Setting up libwayland-client0:amd64 (1.18.0-1) ...
Setting up mesa-vulkan-drivers:amd64 (20.0.8-0ubuntu1~20.04.1) ...
Setting up tcl8.6 (8.6.10+dfsg-1) ...
Setting up liblist-moreutils-perl (0.416-1build5) ...
Setting up libgl1-mesa-dri:amd64 (20.0.8-0ubuntu1~20.04.1) ...
Setting up apt-file (3.2.2) ...
The system-wide cache is empty. You may want to run 'apt-file update'
as root to update the cache.
Setting up tcl (8.6.9+1) ...
Setting up libglx-mesa0:amd64 (20.0.8-0ubuntu1~20.04.1) ...
Setting up libglx0:amd64 (1.3.1-1ubuntu0.20.04.1) ...
Setting up libgl1:amd64 (1.3.1-1ubuntu0.20.04.1) ...
Setting up mesa-utils (8.4.0-1build1) ...
Processing triggers for libc-bin (2.31-0ubuntu9) ...
Processing triggers for man-db (2.9.1-1) ...
Processing triggers for mime-support (3.64ubuntu1) ...
  To Install: pts/sudokut-1.0.1

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

1 Test To Install
  1 File To Download [0.02MB]
  1MB Of Disk Space Is Needed

pts/sudokut-1.0.1:
  Test Installation 1 of 1
  1 File Needed [0.02 MB]
  Downloading: sudokut0.4-1.tar.bz2
  Downloading .....
  Installation Size: 0.1 MB
  Installing Test @ 23:21:04

javierrp@javierrp:~$

```

Y el benchmark restante:

```

javierrp@javierrp:~$ phoronix-test-suite install system/cryptsetup

Phoronix Test Suite v8.6.0
AN OUTDATED VERSION OF THE PHORONIX TEST SUITE IS INSTALLED.
THE VERSION IN USE IS 8.6.0 (8600), BUT THE LATEST IS PTS-CORE 10000.
VISIT HTTPS://WWW.PHORONIX-TEST-SUITE.COM/ TO UPDATE THIS SOFTWARE.

  To Install: system/cryptsetup-1.0.0

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

1 Test To Install
  1MB Of Disk Space Is Needed

system/cryptsetup-1.0.0:
  Test Installation 1 of 1
  Installation Size: 1 MB
  Installing Test @ 23:21:57

javierrp@javierrp:~$ _

```

Ahora que tenemos ambos tests, nos pasamos a CentOS y lo configuramos antes de la obtención de resultados.

- **CENTOS:**

Para prevenir los errores que tuvimos en Ubuntu, instalo una serie de dependencias relacionadas con PHP para Phoronix:

```
[root@localhost ~]# yum install wget php-cli php-xml bzip2
```

```
wget          x86_64      1.19.5-10.el8      AppStream
Instalando dependencias:
libxslt       x86_64      1.1.32-5.el8       BaseOS

Resumen de la transacción
=====
Instalar 3 Paquetes

Tamaño total de la descarga: 1.1 M
Tamaño instalado: 3.9 M
¿Está de acuerdo [s/N]? : s
Descargando paquetes:
(1/3): libxslt-1.1.32-5.el8.x86_64.rpm           870 kB/s | 250 kB
(2/3): php-xml-7.2.24-1.module_el8.2.0+313+b04d0a66.x86_64.rpm 572 kB/s | 188 kB
(3/3): wget-1.19.5-10.el8.x86_64.rpm           1.3 MB/s | 734 kB
-----
Total                                           1.3 MB/s | 1.1 MB
Ejecutando verificación de operación
Verificación de operación exitosa.
Ejecutando prueba de operaciones
Prueba de operación exitosa.
Ejecutando operación
Preparando      :
Instalando      : libxslt-1.1.32-5.el8.x86_64
Instalando      : php-xml-7.2.24-1.module_el8.2.0+313+b04d0a66.x86_64
Instalando      : wget-1.19.5-10.el8.x86_64
Ejecutando scriptlet: wget-1.19.5-10.el8.x86_64
Verificando     : php-xml-7.2.24-1.module_el8.2.0+313+b04d0a66.x86_64
Verificando     : wget-1.19.5-10.el8.x86_64
Verificando     : libxslt-1.1.32-5.el8.x86_64

Instalado:
libxslt-1.1.32-5.el8.x86_64      php-xml-7.2.24-1.module_el8.2.0+313+b04d0a66.x86_64
wget-1.19.5-10.el8.x86_64

¡Listo!
[root@localhost ~]# _
```

Además, PHP necesita una extensión con el nombre de JSON, por lo que antes de continuar la dejamos instalada:

```
[root@localhost phoronix-test-suite]# yum install php-json
```

```

[root@localhost phoronix-test-suite]# yum install php-json
Última comprobación de caducidad de metadatos hecha hace 0:53:58, el mar 08 dic 2020
Dependencias resueltas.
=====
Paquete          Arquitectura Versión                               Repositorio
=====
Instalando:
php-json         x86_64      7.2.24-1.module_el8.2.0+313+b04d0a66 AppStream
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      288 kB/s | 73 kB
-----
Total                                                         115 kB/s | 73 kB
Ejecutando verificación de operación
Verificación de operación exitosa.
Ejecutando prueba de operaciones
Prueba de operación exitosa.
Ejecutando operación
  Preparando      :
  Instalando      : php-json-7.2.24-1.module_el8.2.0+313+b04d0a66.x86_64
  Ejecutando scriptlet: php-json-7.2.24-1.module_el8.2.0+313+b04d0a66.x86_64
  Verificando     : php-json-7.2.24-1.module_el8.2.0+313+b04d0a66.x86_64

Instalado:
  php-json-7.2.24-1.module_el8.2.0+313+b04d0a66.x86_64

¡Listo!
[root@localhost phoronix-test-suite]#

```

Y ahora ya si pasamos a descargar el repositorio de Phoronix:

```

[root@localhost ~]# wget http://phoronix-test-suite.com/releases/phoronix-test-suite-8.6.0.tar.gz
--2020-12-08 18:26:09-- http://phoronix-test-suite.com/releases/phoronix-test-suite-8.6.0.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]:80... conectado.
Petición HTTP enviada, esperando respuesta... 200 OK
Longitud: 885180 (864K) [application/x-gzip]
Grabando a: "phoronix-test-suite-8.6.0.tar.gz"

phoronix-test-suite-8.6. 100%[=====>] 864,43K 924KB/s en 0,9s

2020-12-08 18:26:10 (924 KB/s) - "phoronix-test-suite-8.6.0.tar.gz" guardado [885180/885180]

[root@localhost ~]#

```

Descomprimos el .tar.gz que contiene el ejecutable:

```

[root@localhost ~]# tar xvfz phoronix-test-suite-8.6.0.tar.gz _

```

```

phoronix-test-suite/pts-core/commands/list_missing_dependencies.php
phoronix-test-suite/pts-core/commands/network_info.php
phoronix-test-suite/pts-core/commands/search.php
phoronix-test-suite/pts-core/commands/rename_identifier_in_result_file.php
phoronix-test-suite/pts-core/commands/list_installed_suites.php
phoronix-test-suite/pts-core/commands/debug_self_test.php
phoronix-test-suite/pts-core/commands/wins_and_losses.php
phoronix-test-suite/pts-core/web-interface/
phoronix-test-suite/pts-core/web-interface/favicon.ico
phoronix-test-suite/pts-core/web-interface/js/
phoronix-test-suite/pts-core/web-interface/js/pts-web-socket.js
phoronix-test-suite/pts-core/web-interface/js/pts-web-interface.js
phoronix-test-suite/pts-core/web-interface/js/pts-web-functions.js
phoronix-test-suite/pts-core/web-interface/index.php
phoronix-test-suite/pts-core/web-interface/assets/
phoronix-test-suite/pts-core/web-interface/assets/pts-mobile-interface.css
phoronix-test-suite/pts-core/web-interface/assets/pts-web-logo.png
phoronix-test-suite/pts-core/web-interface/assets/pts-web-interface.css
phoronix-test-suite/pts-core/web-interface/assets/background.png
phoronix-test-suite/pts-core/web-interface/web-interfaces/
phoronix-test-suite/pts-core/web-interface/web-interfaces/pts_webui_test.php
phoronix-test-suite/pts-core/web-interface/web-interfaces/pts_webui_results.php
phoronix-test-suite/pts-core/web-interface/web-interfaces/pts_webui_result.php
phoronix-test-suite/pts-core/web-interface/web-interfaces/pts_webui_tests.php
phoronix-test-suite/pts-core/web-interface/web-interfaces/pts_webui_system.php
phoronix-test-suite/pts-core/web-interface/web-interfaces/pts_webui_loader.php
phoronix-test-suite/pts-core/web-interface/web-interfaces/pts_webui_main.php
phoronix-test-suite/pts-core/web-interface/web-interfaces/pts_webui_search.php
phoronix-test-suite/pts-core/web-interface/web-interfaces/pts_webui_component.php
phoronix-test-suite/pts-core/web-interface/web-interfaces/pts_webui_intro.php
phoronix-test-suite/pts-core/web-interface/html/
phoronix-test-suite/pts-core/web-interface/html/about.html
phoronix-test-suite/pts-core/web-interface/html/benchmark.html
phoronix-test-suite/pts-core/web-interface/html/test_queue.html
phoronix-test-suite/pts-core/web-interface/html/early.html
phoronix-test-suite/pts-core/web-interface/html/settings.html
[root@localhost ~]# _

```

Accedemos al directorio “Phoronix-test-suite” y ejecutamos el script que viene, encargado de instalar Phoronix:

```

[root@localhost ~]# cd phoronix-test-suite/
[root@localhost phoronix-test-suite]# ./install-sh
which: no xdg-mime in (/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/root/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/

[root@localhost phoronix-test-suite]#

```

Y como hicimos en Ubuntu, mostramos el listado de benchmarks disponibles:

```

[root@localhost phoronix-test-suite]# phoronix-test-suite list-available-tests

```

Mostrando, entre otras cosas, elementos como:

pts/y-cruncher	- Y-Cruncher	Processor
pts/yafaray	- Yafaray	Processor
system/apache	- Apache Benchmark	
system/blender	- Blender	System
system/blogbench	- BlogBench	Disk
system/clpeak	- clpeak	System
system/compress-lzma	- LZMA Compression	
system/compress-pbzip2	- PBZIP2 Compression	
system/compress-zstd	- Zstd Compression	Processor
system/cryptsetup	- Cryptsetup	System
system/darktable	- Darktable	System
system/dbench	- Dbench	Disk
system/ethminer	- Ethereum Ethminer	System
system/fio	- Flexible IO Tester	Disk
system/gegl	- GEGL	System
system/gimp	- GIMP	System
system/gmic	- G'MIC	System
system/gnupg	- GnuPG	Processor
system/hugin	- Hugin	System
system/inkscape	- Inkscape	System
system/iozone	- IOzone	Disk
system/libreoffice	- LibreOffice	System
system/mpv	- MPV	System
system/ocrmypdf	- OCRMyPDF	System
system/octave-benchmark	- GNU Octave Benchmark	System
system/openssl	- OpenSSL	
system/opm	- Open Porous Media	Processor
system/rawtherapee	- RawTherapee	System
system/redis	- Redis	
system/rsvg	- librsvg	System
system/selenium	- Selenium	System
system/selenium-top-sites	- Time To Load + View Popular Websites	System
system/sqlite	- SQLite	
system/tesseract-ocr	- Tesseract OCR	System
system/wireguard	- WireGuard + Linux Networking Stack Stress Test Netwo	

Mostramos los que hemos descargado en Ubuntu y los encontramos de la siguiente manera:

```
[root@localhost phoronix-test-suite]# phoronix-test-suite list-available-tests | grep sudokut
pts/sudokut - Sudokut Processor
[root@localhost phoronix-test-suite]# phoronix-test-suite list-available-tests | grep cryptsetup
system/cryptsetup - Cryptsetup System
[root@localhost phoronix-test-suite]# _
```

Instalamos primero Sudokut:

```
[root@localhost phoronix-test-suite]# phoronix-test-suite install pts/sudokut
```



```

Descargando paquetes:
tcl-8.6.8-2.el8.x86_64.rpm                2.5 MB/s | 1.1 MB    00:00
-----
Total                                1.7 MB/s | 1.1 MB    00:00
Ejecutando verificación de operación
Verificación de operación exitosa.
Ejecutando prueba de operaciones
Prueba de operación exitosa.
Ejecutando operación
  Preparando      :                               1/1
  Instalando      : tcl-1:8.6.8-2.el8.x86_64      1/1
  Ejecutando scriptlet: tcl-1:8.6.8-2.el8.x86_64  1/1
  Verificando     : tcl-1:8.6.8-2.el8.x86_64      1/1

Instalado:
  tcl-1:8.6.8-2.el8.x86_64

¿Listo?
  To Install:    pts/sudokut-1.0.1

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

  1 Test To Install
    1 File To Download [0.02MB]
    1MB Of Disk Space Is Needed

  pts/sudokut-1.0.1:
    Test Installation 1 of 1
    1 File Needed [0.02 MB]
    Downloading: sudokut0.4-1.tar.bz2 [0.02MB]
    Downloading .....
    Installation Size: 0.1 MB
    Installing Test @ 19:13:36

```

Y ahora Cryptsetup:

```

[root@localhost phoronix-test-suite]# phoronix-test-suite install cryptsetup

Phoronix Test Suite v8.6.0
AN OUTDATED VERSION OF THE PHORONIX TEST SUITE IS INSTALLED.
THE VERSION IN USE IS 8.6.0 (8600), BUT THE LATEST IS PTS-CORE 10000.
VISIT HTTPS://WWW.PHORONIX-TEST-SUITE.COM/ TO UPDATE THIS SOFTWARE.

  To Install:    system/cryptsetup-1.0.0

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

  1 Test To Install
    1MB Of Disk Space Is Needed

  system/cryptsetup-1.0.0:
    Test Installation 1 of 1
    Installation Size: 1 MB
    Installing Test @ 19:14:28

[root@localhost phoronix-test-suite]# _

```

## - UBUNTU

Una vez instalados ambos tanto en CentOS como en Ubuntu, volvemos a esta última máquina para la realización de las pruebas. Con la sentencia:

```
$ phoronix-test-suite system-info
```

Podemos ver algunas características sobre el sistema, interesantes al tenerlo en cuenta en los resultados y las comparaciones.

```
THE VERSION IN USE IS 8.6.0 (8600), BUT THE LATEST IS PTS-CORE 10000.
VISIT HTTPS://WWW.PHORONIX-TEST-SUITE.COM/ TO UPDATE THIS SOFTWARE.

System Information

PROCESSOR:           Intel Core i5-8250U
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:               2 x Intel 82540EM

MEMORY:               1024MB

DISK:                  2 x 11GB VBOX HDD
File-System:           ext4
Mount Options:         relatime rw
Disk Scheduler:        MQ-DEADLINE

OPERATING SYSTEM:      Ubuntu 20.04
Kernel:                5.4.0-54-generic (x86_64)
System Layer:          Oracle VMware
Security:              KPTI
                      + usercopy/swaps barriers and __user pointer sanitization
                      + Full generic retpoline STIBP: disabled RSB filling
                      + PTE Inversion

javierrp@javierrp:~$
```

Ejecutamos el primer benchmark, que en este caso será Sudoku:

```
javierrp@javierrp:~$ phoronix-test-suite run pts/sudoku
```

Y nos preguntará si queremos guardar los resultados en un archivo y qué nombre le pondremos a este:

```
Would you like to save these test results (Y/n): y
Enter a name for the result file: ubuntu_sudoku
```

Siendo este el resultado de ejecutarlo:

```
Sudoku 0.4:
pts/sudoku-1.0.1
Test 1 of 1
Estimated Trial Run Count: 3
Estimated Time To Completion: 2 Minutes [00:29 UTC]
  Started Run 1 @ 00:27:32
  Started Run 2 @ 00:28:05
  Started Run 3 @ 00:28:38

Total Time:
  32.238322019577
  31.373245000839
  30.778419017792

Average: 31.46 Seconds
Deviation: 2.83%

Seconds < Lower Is Better
Intel Core i5-8250U . 31.46 |=====
ubuntu_sudoku ..... 30.69 |=====
```

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

ubuntu_sudoku:

  Processor: Intel Core i5-8250U (1 Core), Motherboard: Oracle VirtualBox v1.2, Chipset: Intel
  440FX 82441FX PMC, Memory: 1024MB, Disk: 2 x 11GB VBOX HDD, Graphics: VMware SVGA II, Audio: Intel
  82801AA AC 97 Audio, Network: 2 x Intel 82540EM

  OS: Ubuntu 20.04, Kernel: 5.4.0-54-generic (x86_64), File-System: ext4, Screen Resolution: 2
  048x2048, System Layer: Oracle VMware
```

```
Intel Core i5-8250U:

  Processor: Intel Core i5-8250U (1 Core), Motherboard: Oracle VirtualBox v1.2, Chipset: Intel
  440FX 82441FX PMC, Memory: 1024MB, Disk: 2 x 11GB VBOX HDD, Graphics: VMware SVGA II, Audio: Intel
  82801AA AC 97 Audio, Network: 2 x Intel 82540EM

  OS: Ubuntu 20.04, Kernel: 5.4.0-54-generic (x86_64), File-System: ext4, Screen Resolution: 2
  048x2048, System Layer: Oracle VMware

Sudoku 0.4
Total Time
Seconds < Lower Is Better
ubuntu_sudoku ..... 30.69 |=====
Intel Core i5-8250U . 31.46 |=====
```

Ahora pasamos a ejecutar el otro benchmark, "Cryptsetup", guardando los datos en un archivo que, en mi caso y por error, he llamado cryptsetupcentos, pero dejar claro que seguimos en Ubuntu:

```
javierrp@javierrp:~$ phoronix-test-suite run cryptsetup
```

```
Phoronix Test Suite v8.6.0
AN OUTDATED VERSION OF THE PHORONIX TEST SUITE IS INSTALLED.
THE VERSION IN USE IS 8.6.0 (8600), BUT THE LATEST IS PTS-CORE 10000.
VISIT HTTPS://WWW.PHORONIX-TEST-SUITE.COM/ TO UPDATE THIS SOFTWARE.
```

## System Information

```

PROCESSOR:           Intel Core i5-8250U
  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:            2 x Intel 82540EM

MEMORY:              1024MB

DISK:                2 x 11GB VBOX HDD
  File-System:         ext4
  Mount Options:       relatime rw
  Disk Scheduler:      MQ-DEADLINE

OPERATING SYSTEM:    Ubuntu 20.04
  Kernel:              5.4.0-54-generic (x86_64)
  System Layer:        Oracle VMWare
  Security:            KPTI

```

```
Estimated Time To Completion: 3 Minutes [00:40 UTC]
Started Run 1 @ 00:37:59
Started Run 2 @ 00:40:25
Started Run 3 @ 00:41:02
```

```
PBKDF2-sha512:
1038194
1018034
1028015
```

Average: 1028081 Iterations Per Second  
Deviation: 0.98%

```
Result compared to 99 OpenBenchmarking.org samples; median: 1645272. Box plot of sampling:  
[-----*-----#####!####*-***-*-*-*-*]  
This Result (9th Percentile): 1028081 ^  
  
Intel Core i9-10900K: 2019102 ^  
Intel Core i5-10600K: 1902104 ^  
AMD Ryzen 9 3900X 12-Core: 1791848 ^
```

```

PBKDF2-whirlpool:
    498372
    485451
    486352

```

Average: 490058 Iterations Per Second  
Deviation: 1.47%

[illegible]

```

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

Intel Core i5-8250U - VMware SVGA II - Oracle:

Processor: Intel Core i5-8250U (1 Core), Motherboard: Oracle VirtualBox v1.2, Chipset: Intel
440FX 82441FX PMC, Memory: 1024MB, Disk: 2 x 11GB VBOX HDD, Graphics: VMware SVGA II, Audio: Intel
82801AA AC 97 Audio, Network: 2 x Intel 82540EM

OS: Ubuntu 20.04, Kernel: 5.4.0-54-generic (x86_64), File-System: ext4, Screen Resolution: 2
048x2048, System Layer: Oracle VMware

Cryptsetup 2.2.2
PBKDF2-sha512
Iterations Per Second > Higher Is Better
Intel Core i5-8250U - VMware SVGA II - Oracle . 1028081 |=====

Cryptsetup
PBKDF2-whirlpool
Iterations Per Second > Higher Is Better
Intel Core i5-8250U - VMware SVGA II - Oracle . 490058 |=====

```

## - CENTOS

Ahora pasamos a centos a realizar la ejecución de ambos tests aquí también. El primero de ellos volverá a ser Sudokut:

```

[root@localhost phoronix-test-suite]# phoronix-test-suite run pts/sudokut

Phoronix Test Suite v8.6.0
AN OUTDATED VERSION OF THE PHORONIX TEST SUITE IS INSTALLED.
THE VERSION IN USE IS 8.6.0 (8600), BUT THE LATEST IS PTS-CORE 10000.
VISIT HTTPS://WWW.PHORONIX-TEST-SUITE.COM/ TO UPDATE THIS SOFTWARE.

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:               2 x Intel 82540EM

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:                4.18.0-193.28.1.el8_2.x86_64 (x86_64)
System Layer:          Oracle VMware
Security:              SELinux

```

```

Sudoku 0.4:
pts/sudoku-1.0.1
Test 1 of 1
Estimated Trial Run Count: 3
Estimated Time To Completion: 2 Minutes [19:45 EST]
  Started Run 1 @ 19:44:01
  Started Run 2 @ 19:44:38
  Started Run 3 @ 19:45:15

Total Time:
  36.776849031448
  36.125168085098
  36.068422079086

Average: 36.32 Seconds
Deviation: 1.08%

Result compared to 15,136 OpenBenchmarking.org samples; median: 30.76. Box plot of sampling:
[ * * * * |-----*-----##*!#-----* * ]
                                This Result (27th Percentile): 36.32 ^
                                ^ ARMv7 rev 1: 154
                                ^ Intel Celeron 550: 161
                                AMD Athlon 64 X2 4000: 62 ^ Intel Core i7-9700K: 8.627 ^
                                Intel Core i7-5557U: 13.593 ^
                                Intel Xeon E5-2667 v4: 18.783 ^

```

```

centossudoku
Oracle VMware testing on CentOS Linux 8 via the Phoronix Test Suite.

Intel Core i5-8250U:

  Processor: Intel Core i5-8250U (1 Core), Motherboard: Oracle VirtualBox v1.2, Chipset: Intel
  440FX 82441FX PMC, Memory: 818MB, Disk: 9GB VBOX HDD, Graphics: VMware SUGA II, Audio: Intel 82801A
  A AC 97 Audio, Network: 2 x Intel 82540EM

  OS: CentOS Linux 8, Kernel: 4.18.0-193.28.1.el8_2.x86_64 (x86_64), File-System: xfs, Screen
  Resolution: 2048x2048, System Layer: Oracle VMware

  Sudoku 0.4
  Total Time
  Seconds < Lower Is Better
  Intel Core i5-8250U . 36.32 |=====

```

Y ahora pasamos con Cryptsetup, tanto su ejecución como resultados.

```

[root@localhost phoronix-test-suite]# phoronix-test-suite run system/cryptsetup

Phoronix Test Suite v8.6.0
AN OUTDATED VERSION OF THE PHORONIX TEST SUITE IS INSTALLED.
THE VERSION IN USE IS 8.6.0 (8600), BUT THE LATEST IS PTS-CORE 10000.
VISIT HTTPS://WWW.PHORONIX-TEST-SUITE.COM/ TO UPDATE THIS SOFTWARE.

System Information

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

GRAPHICS:      VMware SUGA II
Screen:        2048x2048

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

```



desviación típica, relacionada con dispersión de los datos con respecto de la media, obtenemos que en Ubuntu es de un 2.33% y en Centos un 1.08%.

Para el benchmark de *cryptsetup*, en Ubuntu el resultado tras las dos pruebas que realiza es de 1028081 iteraciones (con una desviación de 0.98%) y 490058 iteraciones (con una desviación de 1.47%) respectivamente, mientras que en centos las pruebas resultaron en 1064701 iteraciones (con una desviación de 1.48%) y 485123 iteraciones (con una desviación de 2.91%). En cuanto a desviación, Ubuntu se encuentra mucho menos disperso con respecto a la media, pero en potencial de encriptación, centOS parece estar en ventaja.

## DOCKER UBUNTU

Seguiremos el material proporcional por la asignatura para la instalación en Ubuntu Server de Docker.

Lo primero será añadir una llave GPG para el repositorio:

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

Y ahora sí añadimos el repositorio y actualizamos la lista de estos:

```
javierrp@javierrp:~$ sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu $(lsb_release -cs) stable"
```

```
javierrp@javierrp:~$ sudo apt update
Hit:1 http://es.archive.ubuntu.com/ubuntu focal InRelease
Ign:2 http://ppa.launchpad.net/ansible/ansible/ubuntu focal InRelease
Hit:3 https://download.docker.com/linux/ubuntu focal InRelease
Hit:4 http://es.archive.ubuntu.com/ubuntu focal-updates InRelease
Hit:5 http://es.archive.ubuntu.com/ubuntu focal-backports InRelease
Err:6 http://ppa.launchpad.net/ansible/ansible/ubuntu focal Release
104 Not Found [IP: 91.188.85.88 80]
```

Buscamos Docker:

```
javierrp@javierrp:~$ apt search docker-ce
Sorting... Done
Full Text Search... Done
docker-ce/focal 5:19.03.14~3-0~ubuntu-focal amd64
  Docker: the open-source application container engine
docker-ce-cli/focal 5:19.03.14~3-0~ubuntu-focal amd64
  Docker CLI: the open-source application container engine
javierrp@javierrp:~$
```



Y una vez que lo encontramos, lo instalamos:

```
javierrp@javierrp:~$ sudo apt install docker-ce
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  aufs-tools cgroupfs-mount containerd.io docker-ce-cli pigz
The following NEW packages will be installed:
  aufs-tools cgroupfs-mount containerd.io docker-ce docker-ce-cli pigz
0 upgraded, 6 newly installed, 0 to remove and 97 not upgraded.
Need to get 91.5 MB of archives.
After this operation, 413 MB of additional disk space will be used.
Do you want to continue? [Y/n]
```

Y al comprobar el estado del servicio veremos lo siguiente:

```
javierrp@javierrp:~$ 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 2020-12-09 01:07:17 UTC; 2min 19s ago
 TriggeredBy: ● docker.socket
    Docs: https://docs.docker.com
   Main PID: 15092 (dockerd)
      Tasks: 8
     Memory: 38.0M
    CGroup: /system.slice/docker.service
            └─15092 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock

Dec 09 01:07:16 javierrp dockerd[15092]: time="2020-12-09T01:07:16.374355529Z" level=warning msg="Y>
Dec 09 01:07:16 javierrp dockerd[15092]: time="2020-12-09T01:07:16.374362584Z" level=warning msg="Y>
Dec 09 01:07:16 javierrp dockerd[15092]: time="2020-12-09T01:07:16.374374554Z" level=warning msg="Y>
Dec 09 01:07:16 javierrp dockerd[15092]: time="2020-12-09T01:07:16.374546462Z" level=info msg="Load>
Dec 09 01:07:16 javierrp dockerd[15092]: time="2020-12-09T01:07:16.749915925Z" level=info msg="Defa>
Dec 09 01:07:17 javierrp dockerd[15092]: time="2020-12-09T01:07:17.112549378Z" level=info msg="Load>
Dec 09 01:07:17 javierrp dockerd[15092]: time="2020-12-09T01:07:17.361227624Z" level=info msg="Dock>
Dec 09 01:07:17 javierrp dockerd[15092]: time="2020-12-09T01:07:17.361604028Z" level=info msg="Daem>
Dec 09 01:07:17 javierrp systemd[1]: Started Docker Application Container Engine.
Dec 09 01:07:17 javierrp dockerd[15092]: time="2020-12-09T01:07:17.558600850Z" level=info msg="API >
lines 1-21/21 (END)
```

Para dar permisos añadimos nuestros usuarios al grupo de superusuarios de docker:

```
javierrp@javierrp:~$ sudo usermod -aG docker javierrp
javierrp@javierrp:~$
```

Para comprobar el correcto funcionamiento pasamos a ejecutar el programa hello-world:

```
javierrp@javierrp:~$ sudo docker run hello-world
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
0e03bdcc26d7: Pull complete
Digest: sha256:e7c70bb24b462baa86c102610182e3efcb12a04854e8c582838d92970a09f323
Status: Downloaded newer image for hello-world:latest

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

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

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

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

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

javierrp@javierrp:~$
```

## DOCKER-COMPOSE UBUNTU

Pasamos a la instalación de Docker-compose:

```
javierrp@javierrp:~$ sudo apt install docker-compose
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  python3-cached-property python3-docker python3-dockerpty python3-docopt python3-texttable
  python3-websocket
Recommended packages:
  docker.io
The following NEW packages will be installed:
  docker-compose python3-cached-property python3-docker python3-dockerpty python3-docopt
  python3-texttable python3-websocket
0 upgraded, 7 newly installed, 0 to remove and 97 not upgraded.
Need to get 262 kB of archives.
After this operation, 1616 kB of additional disk space will be used.
Do you want to continue? [Y/n] _
```

Y comprobamos la versión instalada, que deberá mostrar:

```
javierrp@javierrp:~$ docker-compose --version
docker-compose version 1.25.0, build unknown
javierrp@javierrp:~$ _
```

## JMETER UBUNTU

Clonamos el repositorio de David Plomar:

```
javierrp@javierrp:~$ git clone https://github.com/davidPalomar-ugr/iseP4JMeter.git
Cloning into 'iseP4JMeter'...
remote: Enumerating objects: 4, done.
remote: Counting objects: 100% (4/4), done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 3778 (delta 0), reused 0 (delta 0), pack-reused 3774
Receiving objects: 100% (3778/3778), 7.78 MiB | 4.67 MiB/s, done.
Resolving deltas: 100% (706/706), done.
javierrp@javierrp:~$
```

Nos movemos al directorio “iseP4JMeter” y dentro levantamos la aplicación (la opción -d es para dejarlo como demonio en segundo plano):

```
javierrp@javierrp:~/iseP4JMeter$ docker-compose up -d
Starting isep4jmeter_mongodb_1 ... done
Starting isep4jmeter_mongodbininit_1 ... done
Starting isep4jmeter_nodejs_1 ... done
javierrp@javierrp:~/iseP4JMeter$
```

Sabemos que JMeter usará el puerto 3000, por lo que permitimos con el cortafuegos el uso de ese puerto.

```
root@javierrp:/home/javierrp/iseP4JMeter# ufw allow 3000
Rule added
Rule added (v6)
```

### JMETER EN WINDOWS:

Para la instalación de JMeter en Windows vamos al siguiente enlace:

[Apache JMeter - Download Apache JMeter](#)

Y descargamos el .zip de la versión 5.4:

# Apache JMeter 5.4 (Requires Java 8+)

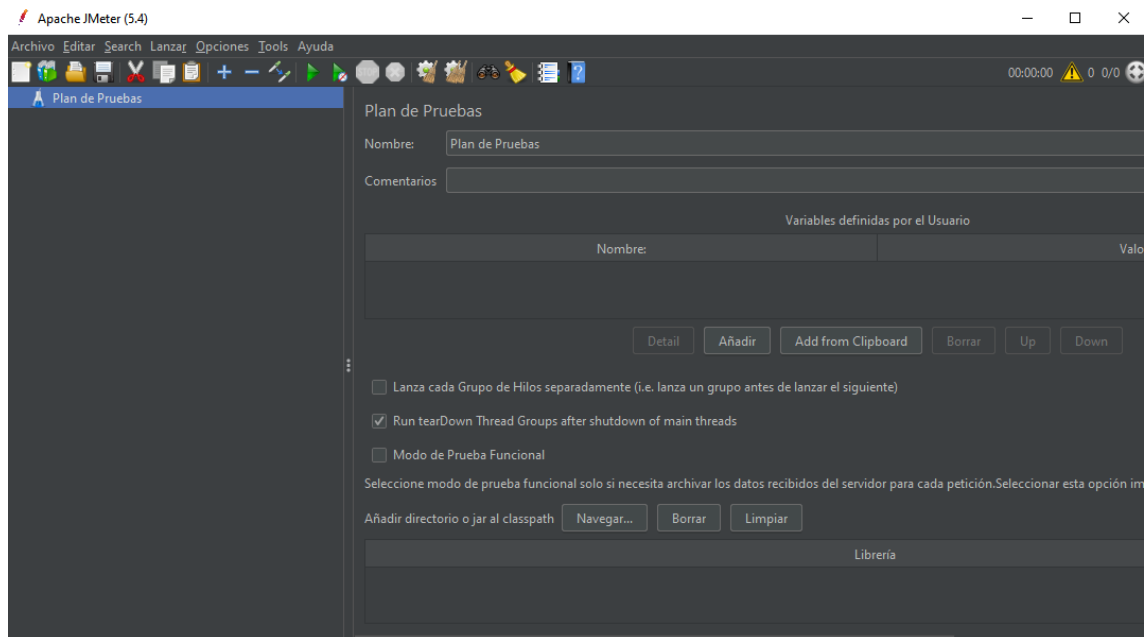
## Binaries

[apache-jmeter-5.4.tgz](#) [sha512](#) [pgp](#)  
[apache-jmeter-5.4.zip](#) [sha512](#) [pgp](#)

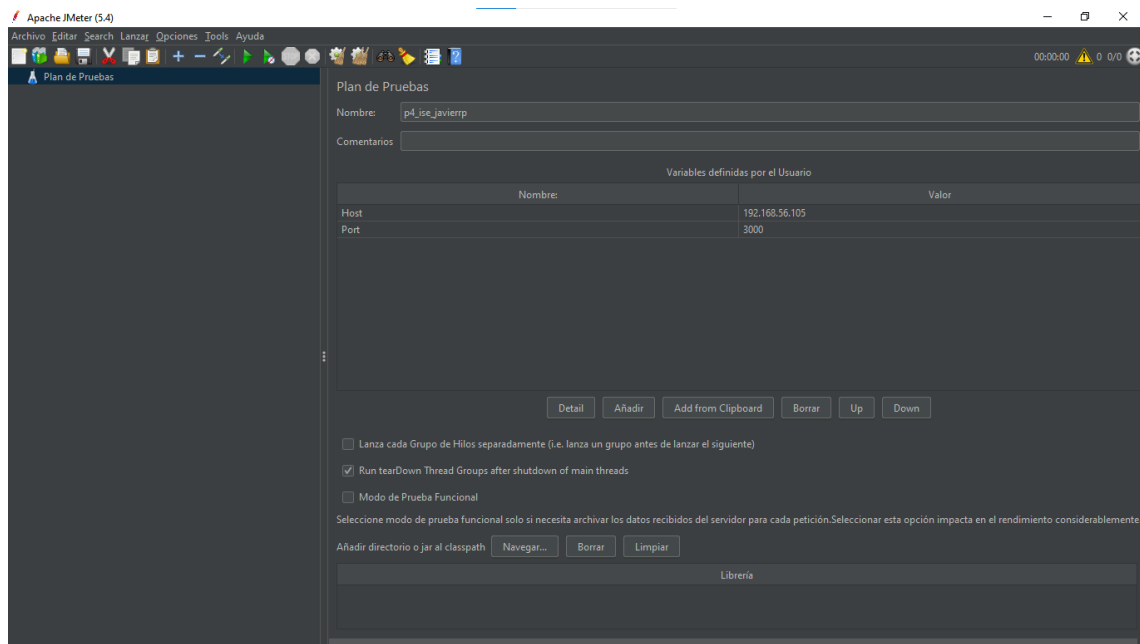
Y en la terminal (cmd) de Windows vamos a donde tengas descomprimido lo que acabamos de descargar (en mi caso `C:/Users/xaviv/Downloads/apache-jmeter-5.4/bin`) y lo ejecutamos con la orden 'java', lo cual abrirá el programa automáticamente.

```
C:\Users\xaviv\Downloads\apache-jmeter-5.4\bin>java -jar ApacheJMeter.jar
```

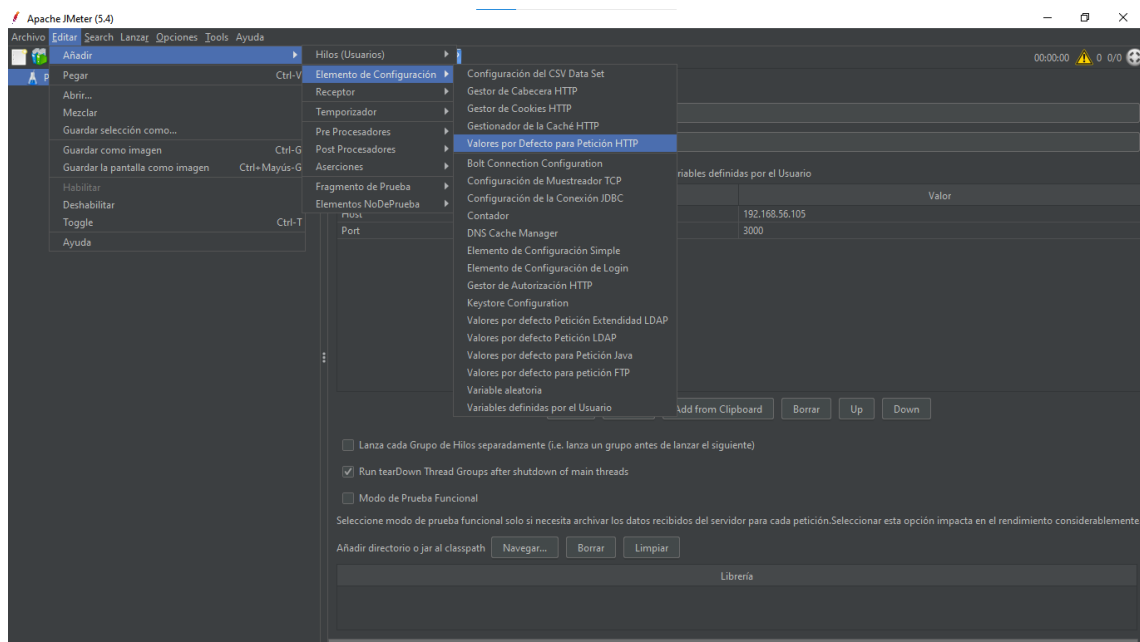
Siendo esto lo que se abrirá de primeras:

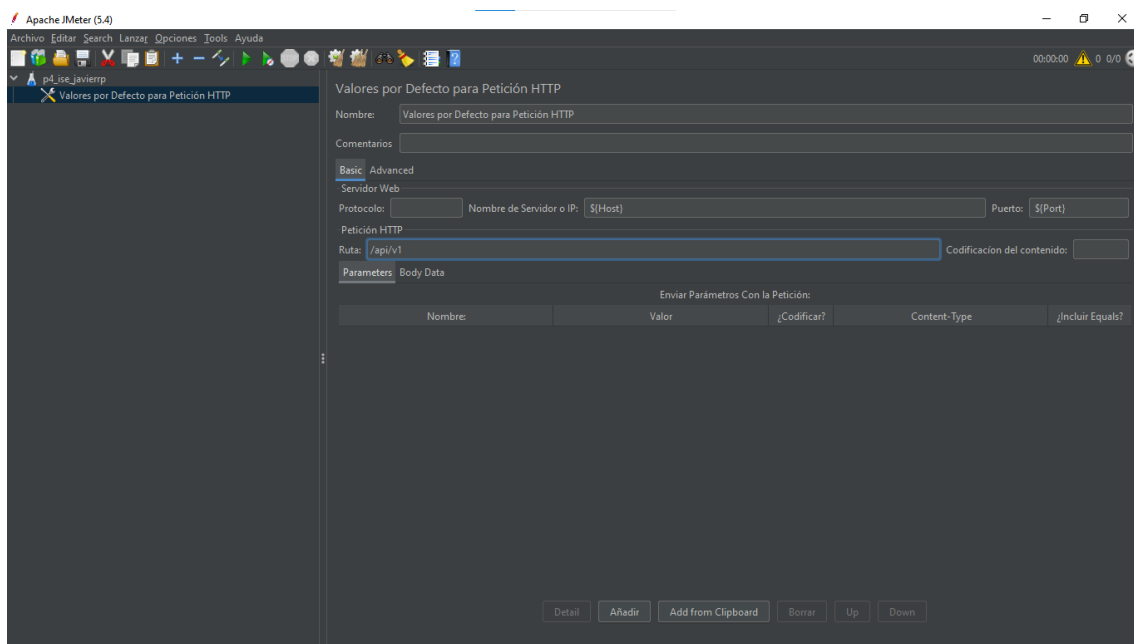


Siendo esta la configuración que estableceremos en la página principal que se abre al principio.

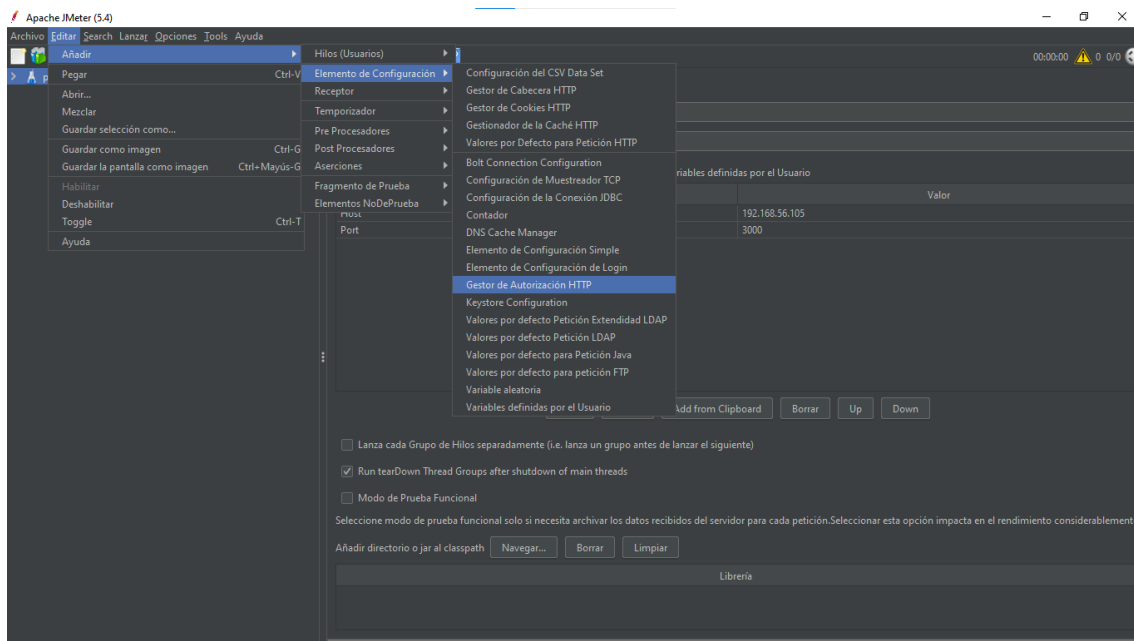


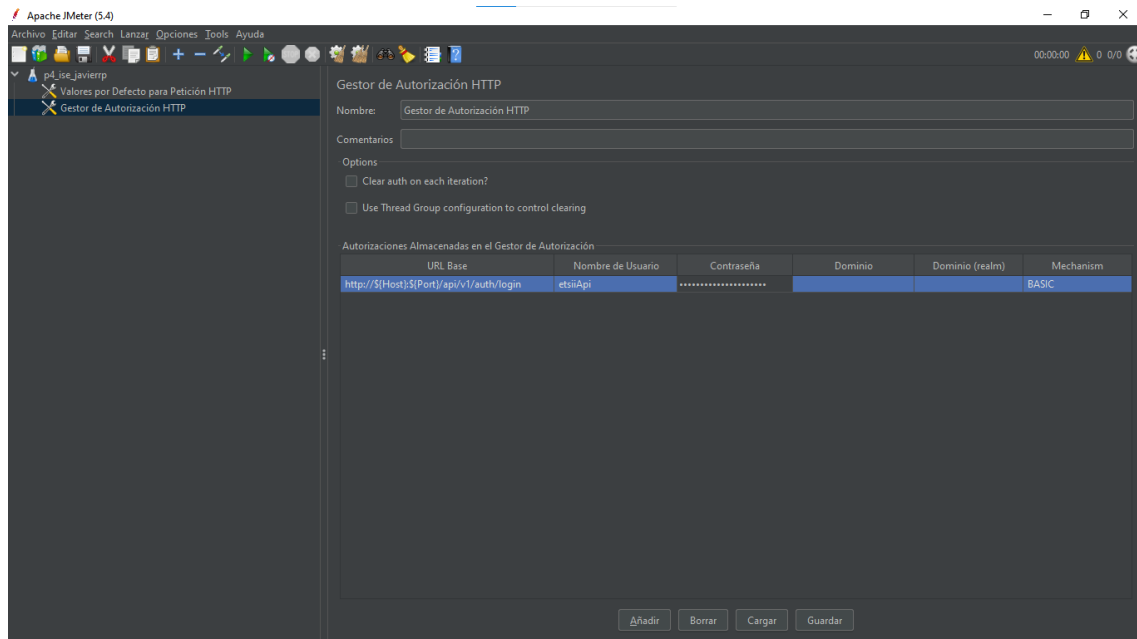
A continuación, dejo unas capturas de los lugares a los que hay que acceder y la configuración que hay que poner en cada sitio. Es importante remarcar el lugar en el que nos encontramos (que se ve en la parte lateral izquierda) cuando nos dirigimos a los diferentes apartados.



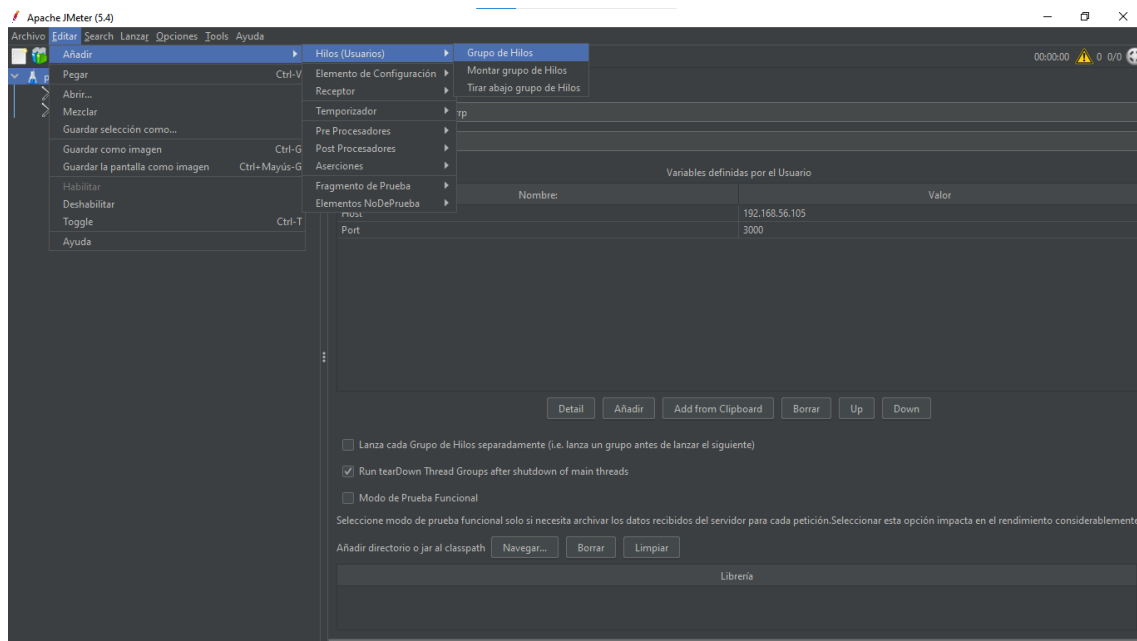


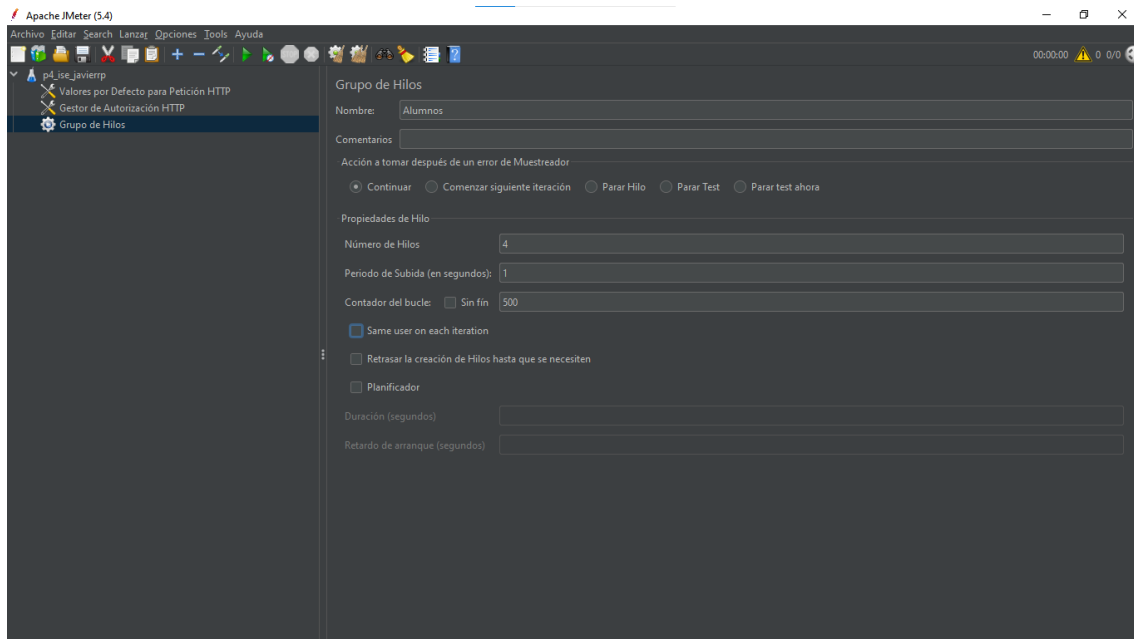
En “p4\_ise\_javierrp” vamos a *Editar -> Añadir -> Elemento de Configuración -> Gestor de autorización HTTP*



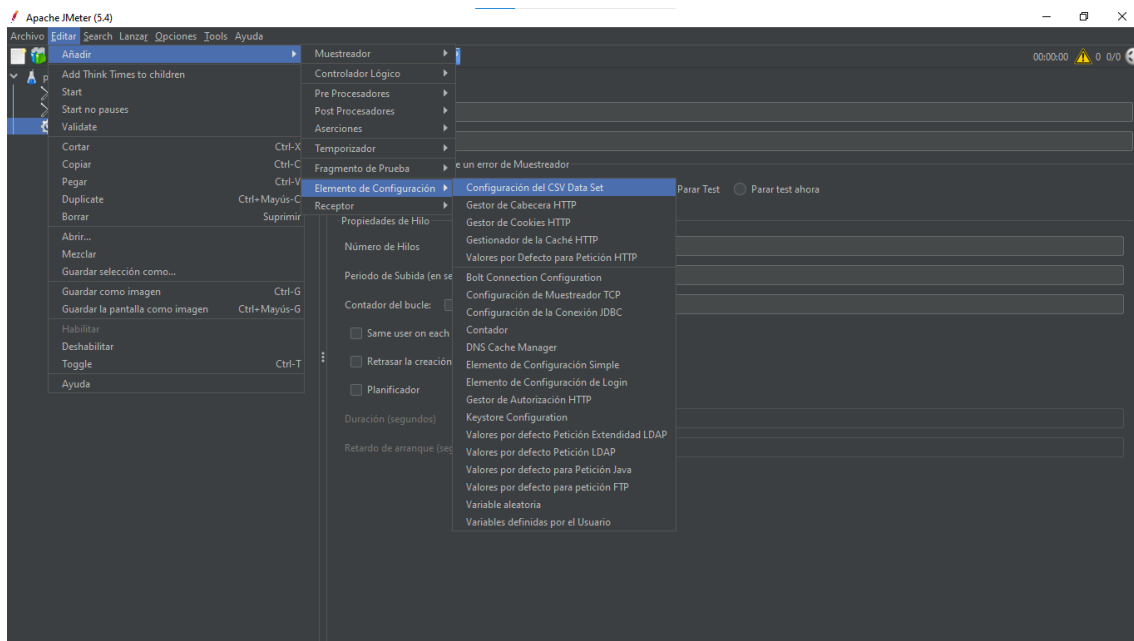


En “p4\_ise\_javierrp” vamos a *Editar -> Añadir -> Hilos (Usuarios) -> Grupo de hilos*

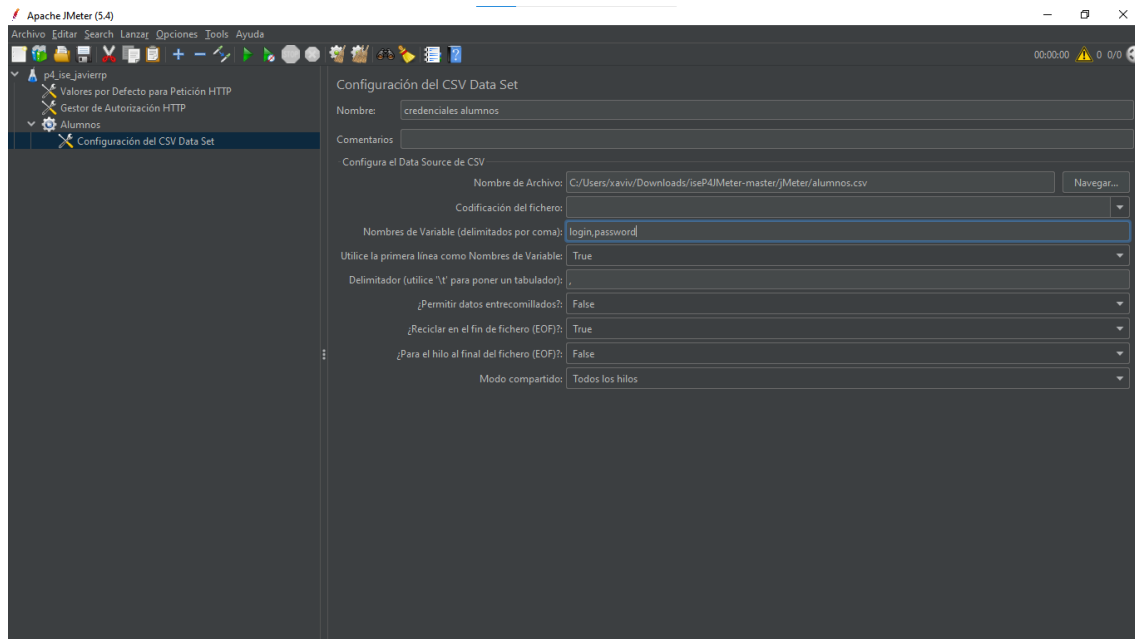




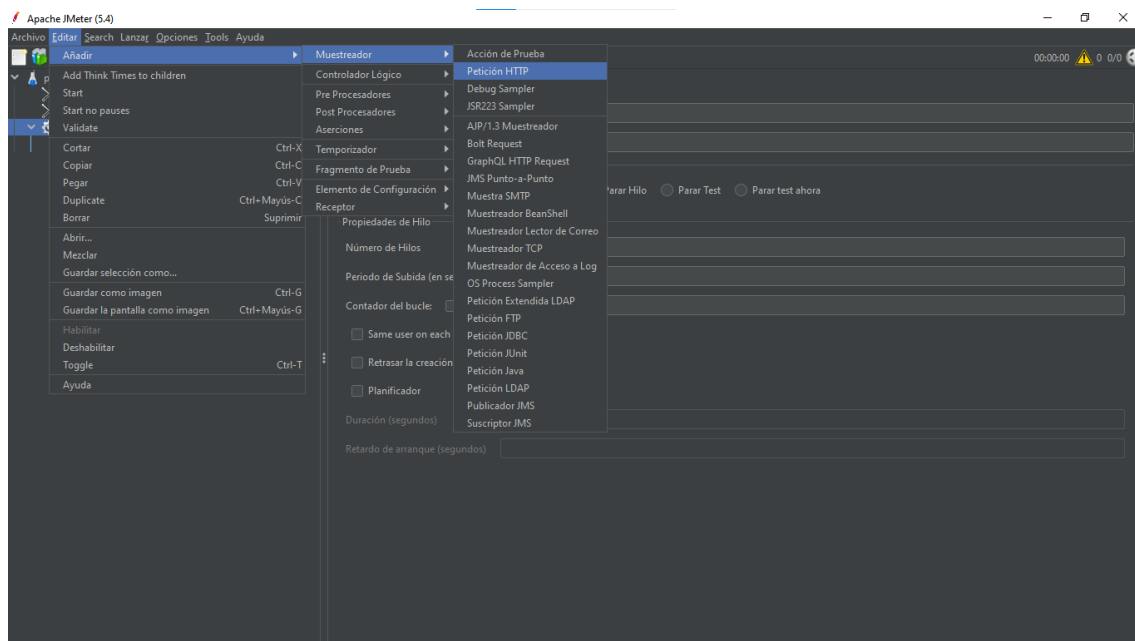
En “grupo de hilos” (o “alumnos” si los cambios ya se han guardado) vamos a *Editar -> Añadir -> Elemento de Configuración -> Configuración del CSV Data Set*

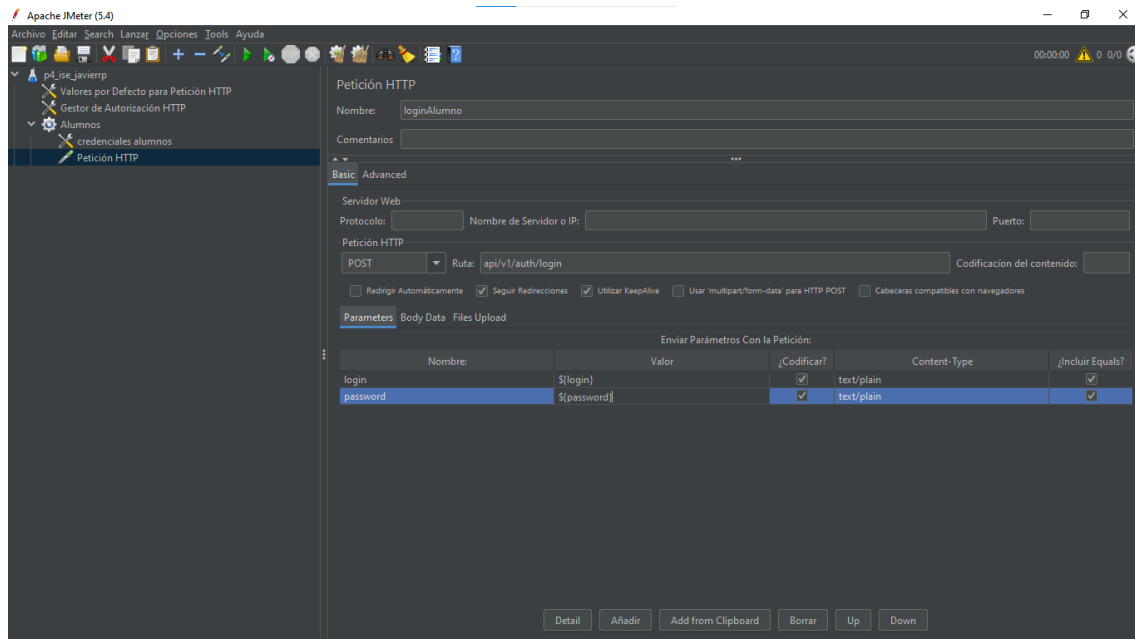




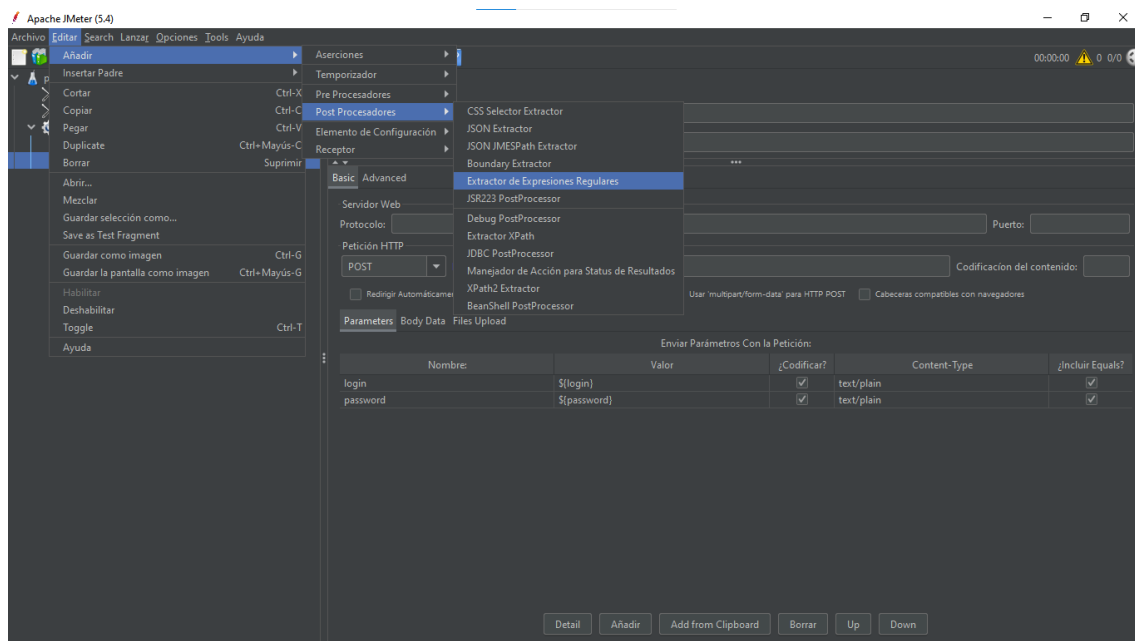


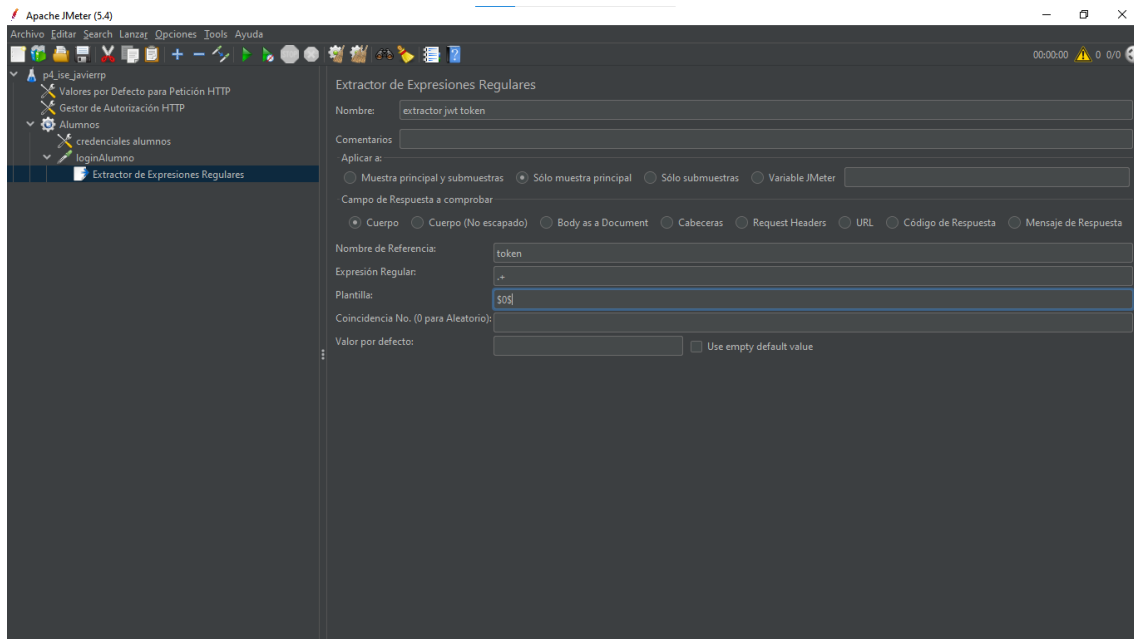
En “alumnos” vamos a *Editar -> Añadir -> Muestreador -> Petición HTTP*



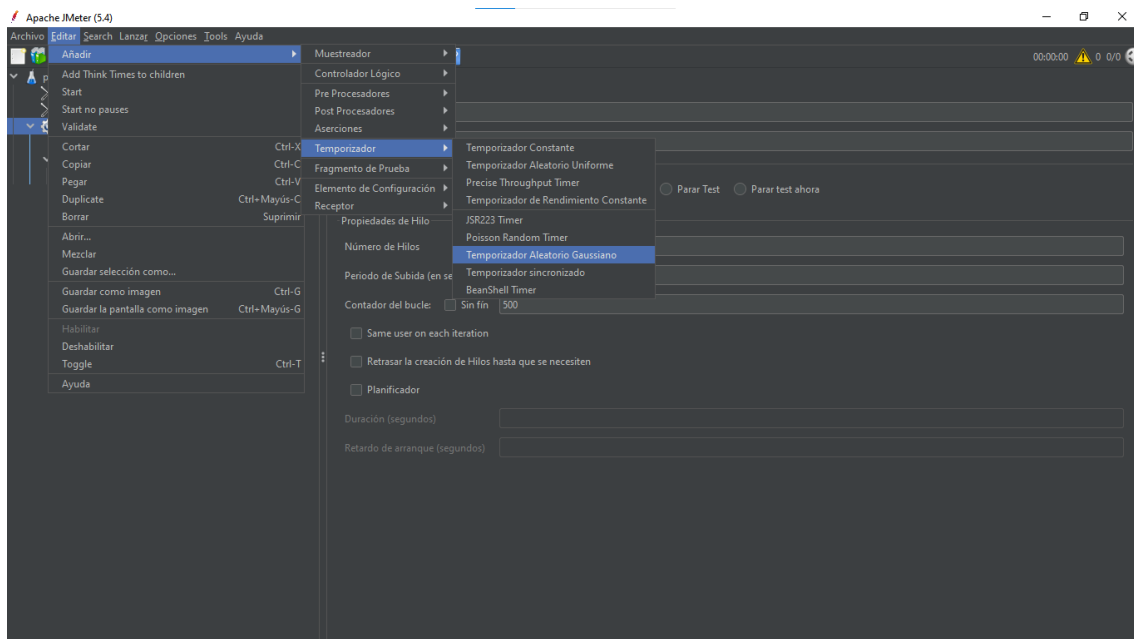


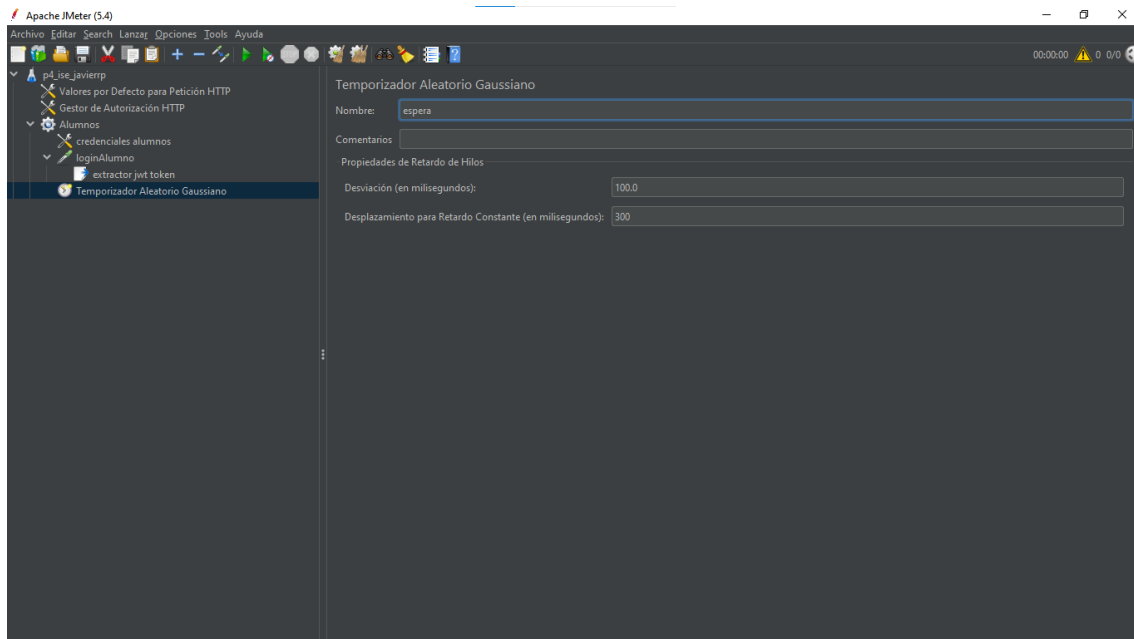
En “petición HTTP” (o en “loginAlumno” si se han guardado los cambios) vamos a *Editar -> Añadir -> Post Procesadores -> Extractor de Expresiones Regulares*



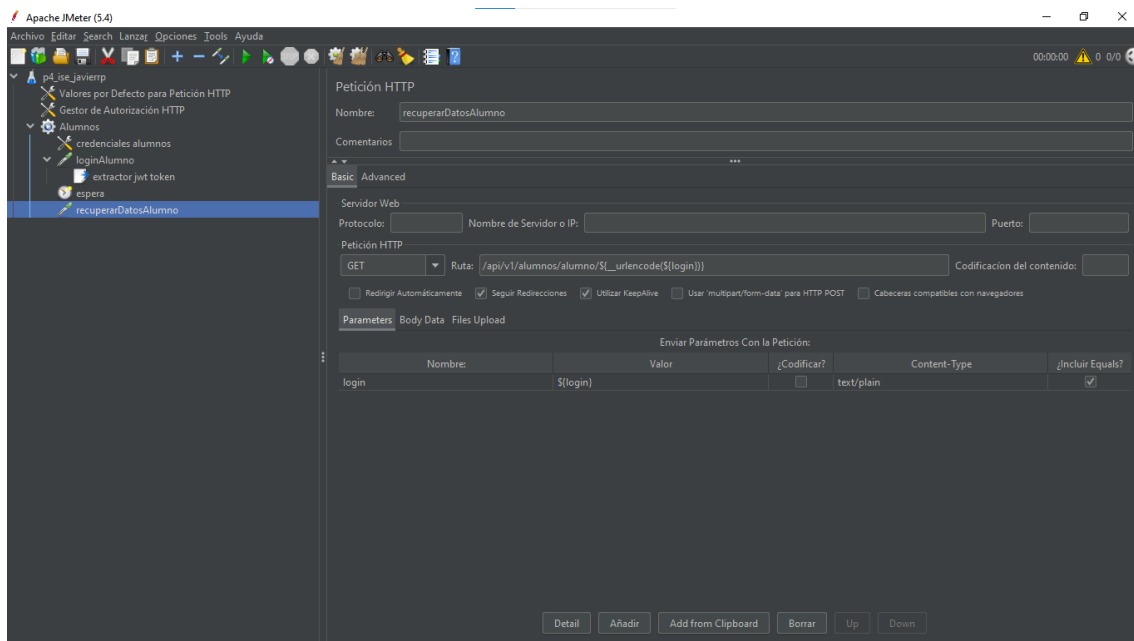


En “alumnos” vamos a *Editar -> Añadir -> Temporizador -> Temporizador aleatorio Gaussiano autorización HTTP*

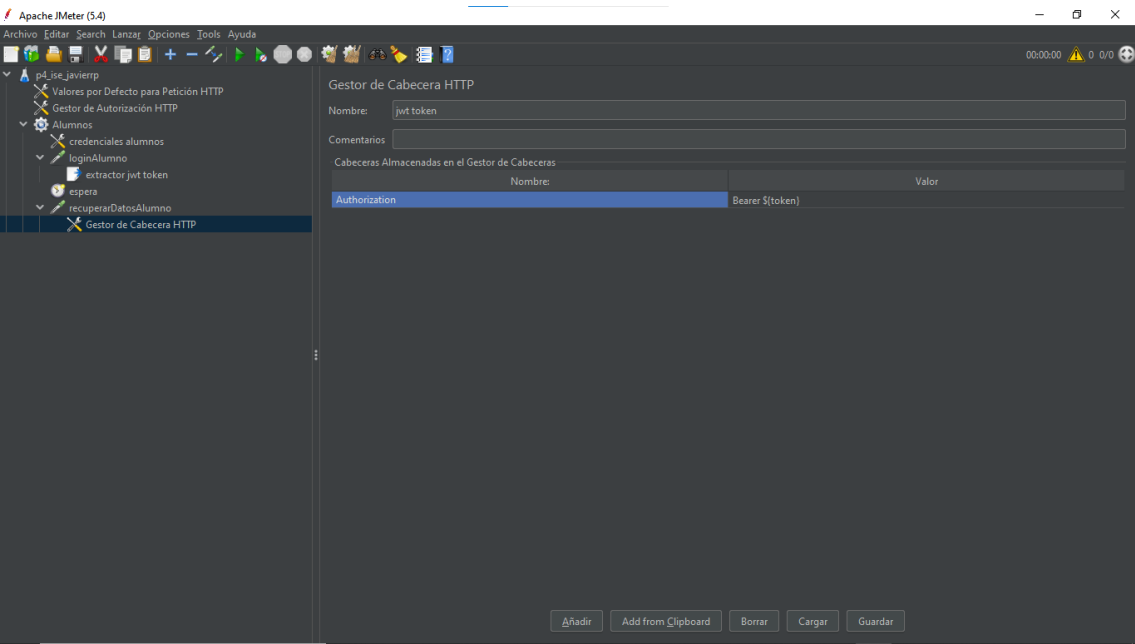
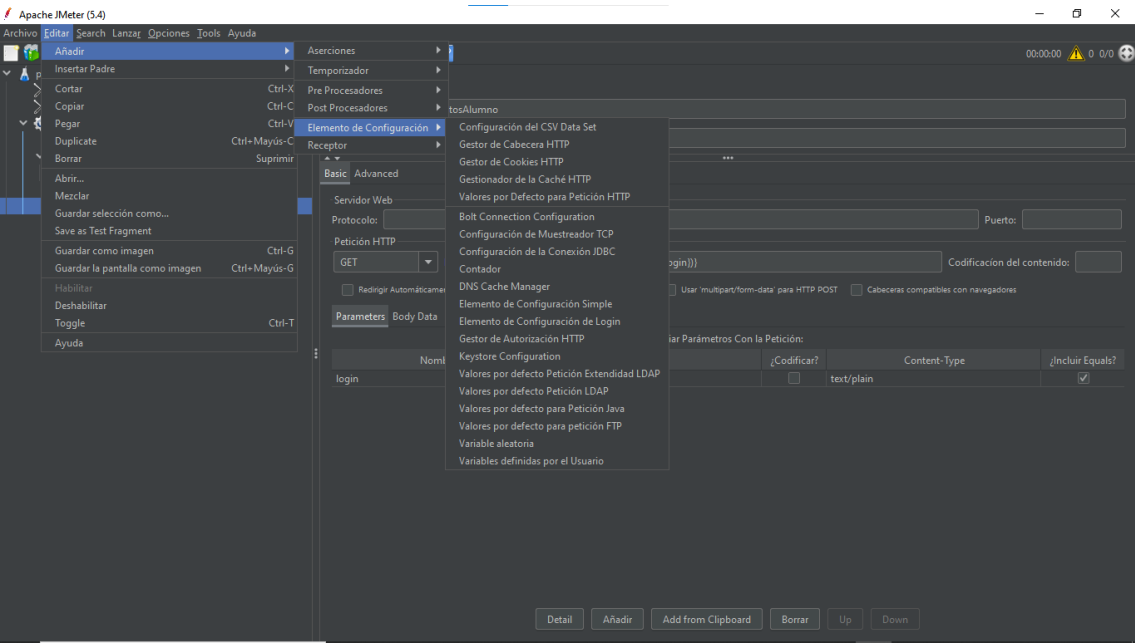




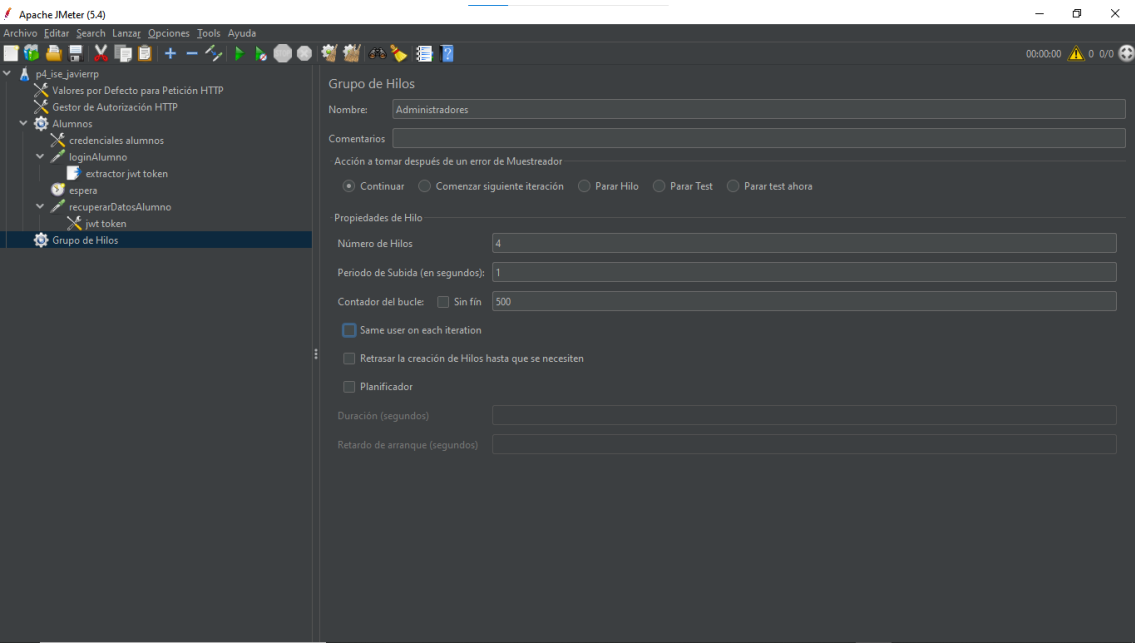
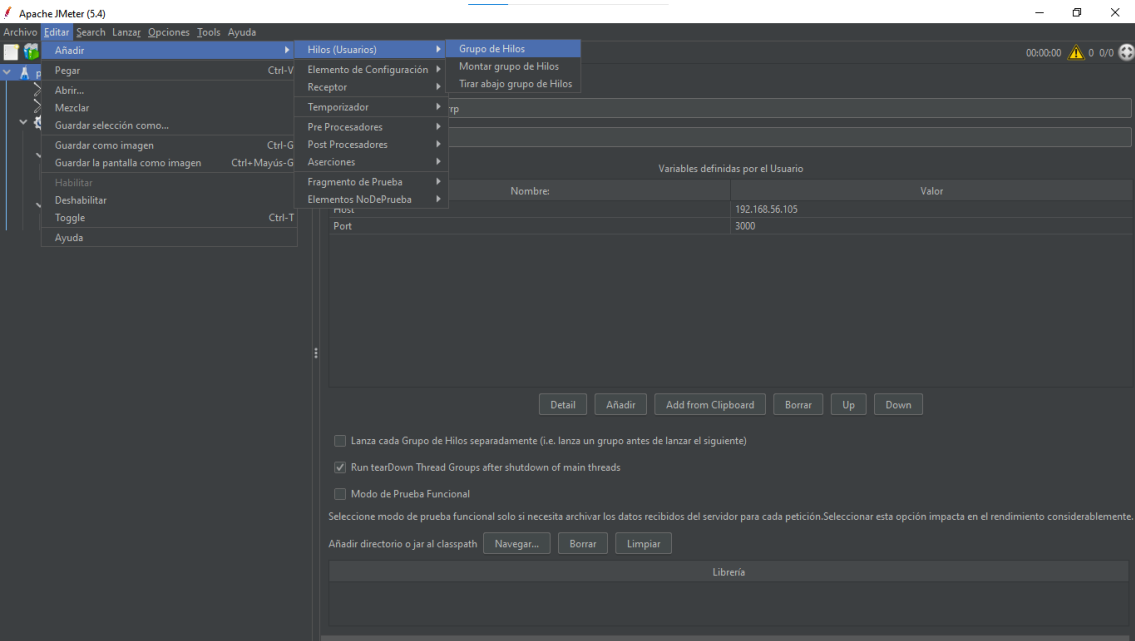
En “alumnos” vamos a *Editar -> Añadir -> Muestreador -> Petición HTTP*

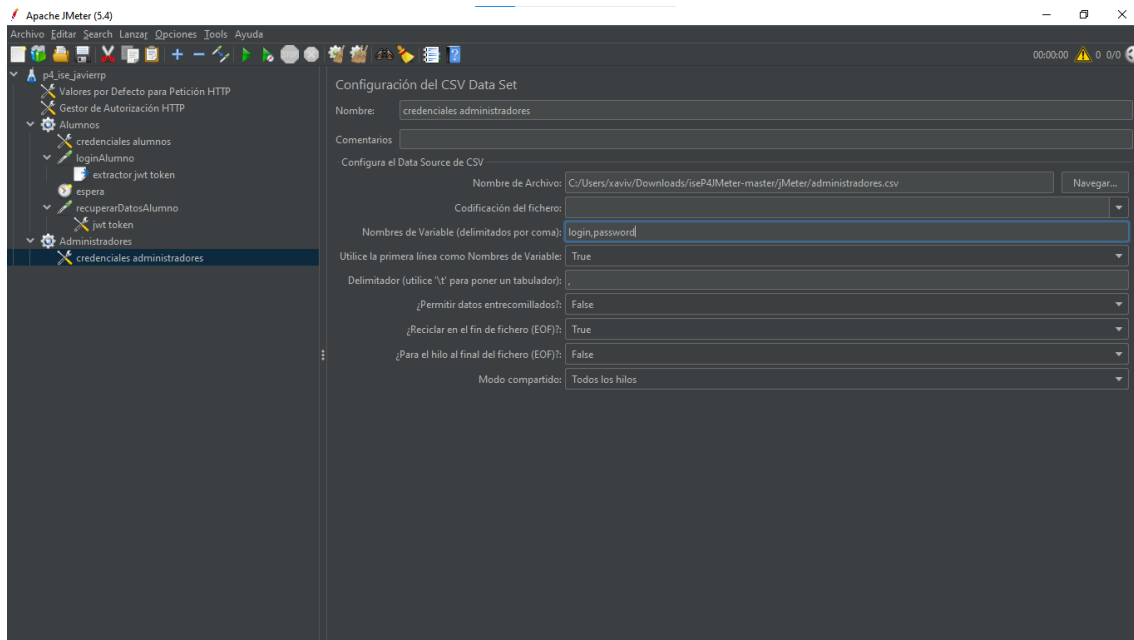
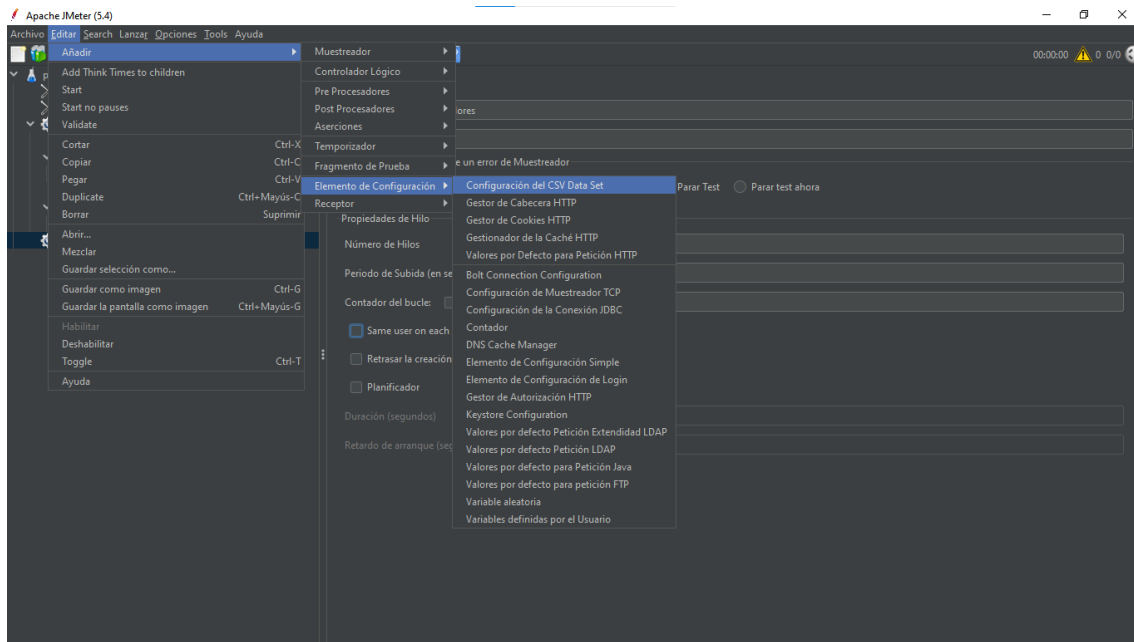


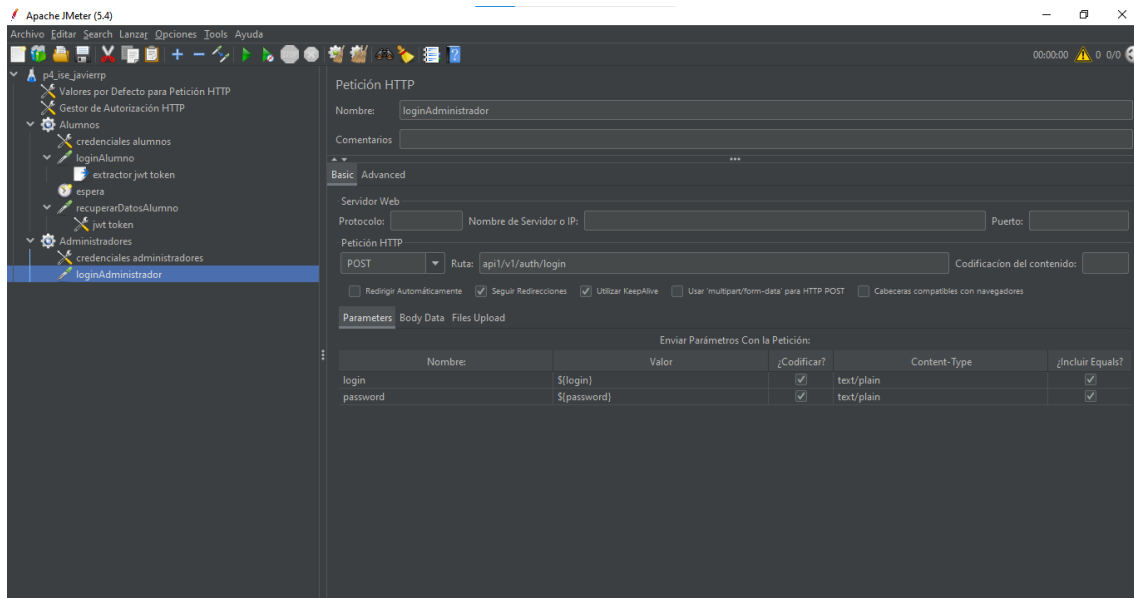
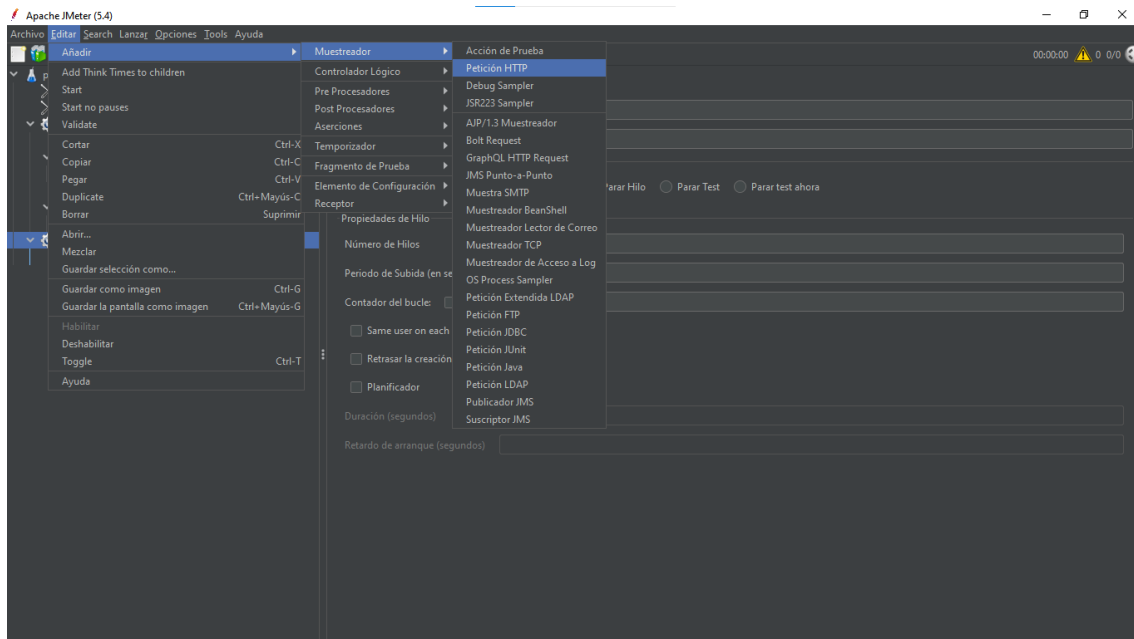
En “recuperarDatosAlumnos” vamos a *Editar -> Añadir -> Muestreador -> Petición HTTP*



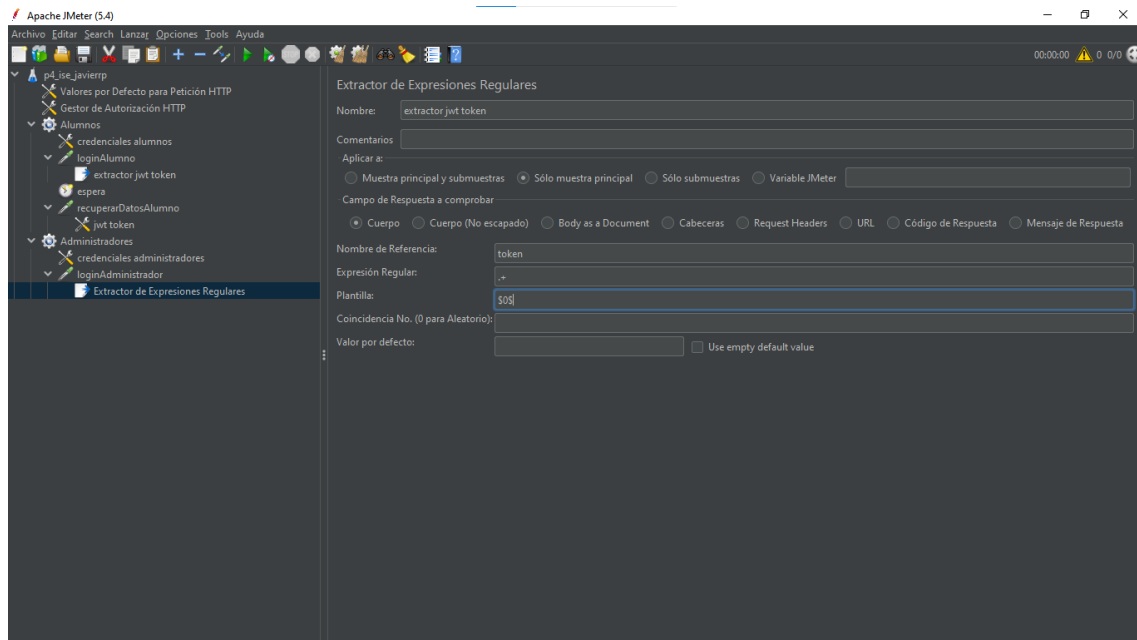
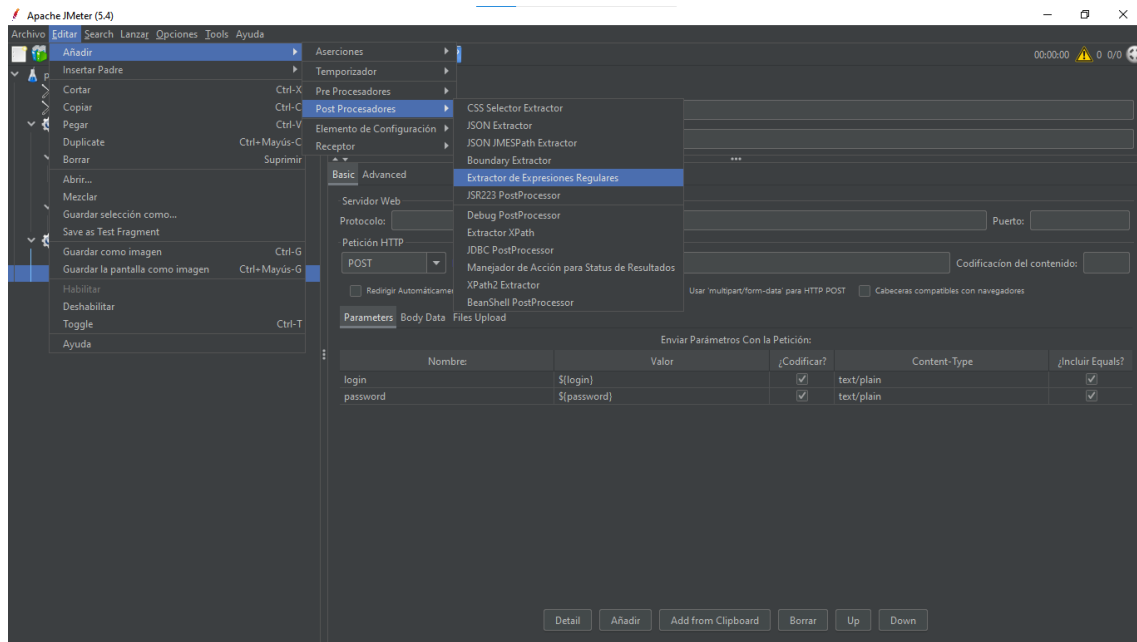
Y tras toda esta configuración queda terminada la parte de los alumnos, ahora toca hacer la de los administradores, que en cuanto a procedimiento es tan similar que solamente pondré las capturas con los lugares a los que acceder, las cosas que añadir y la configuración que dejar:

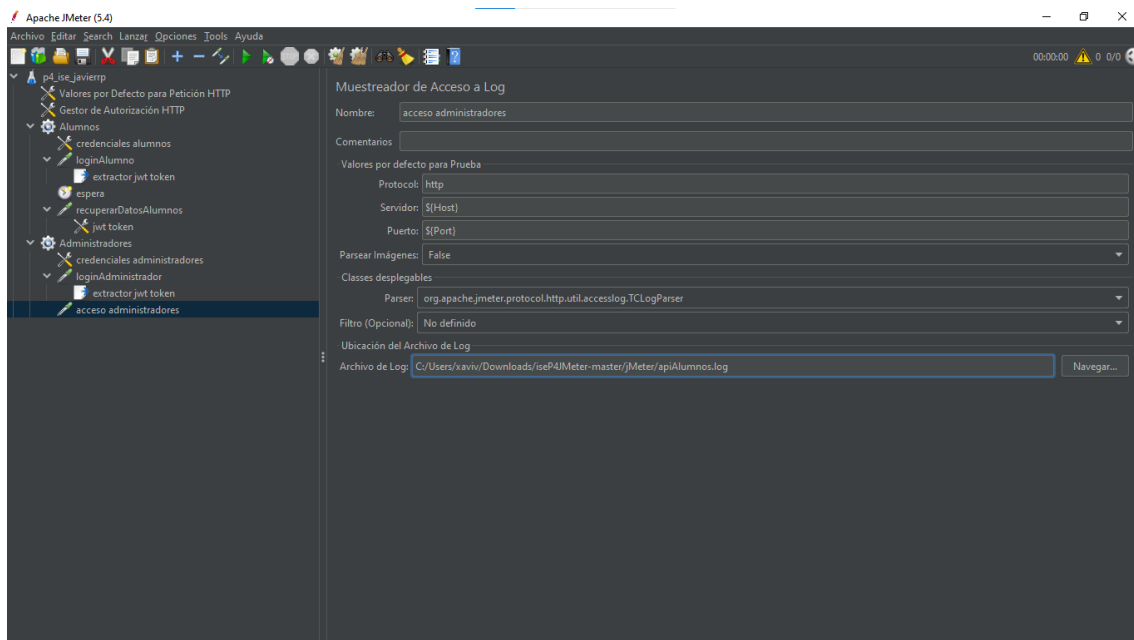
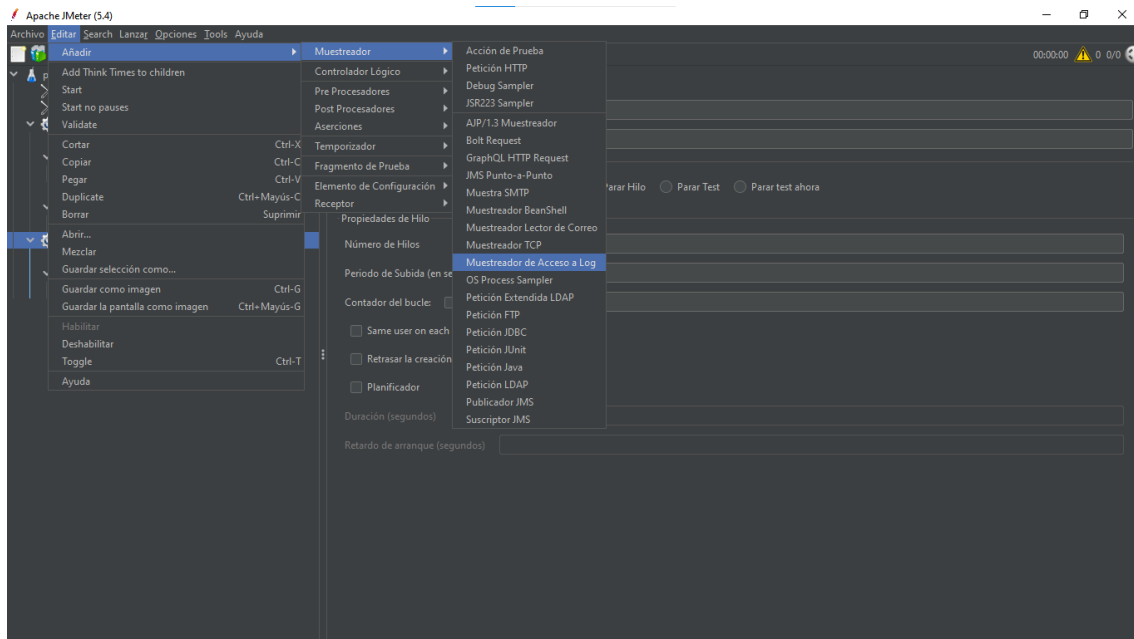


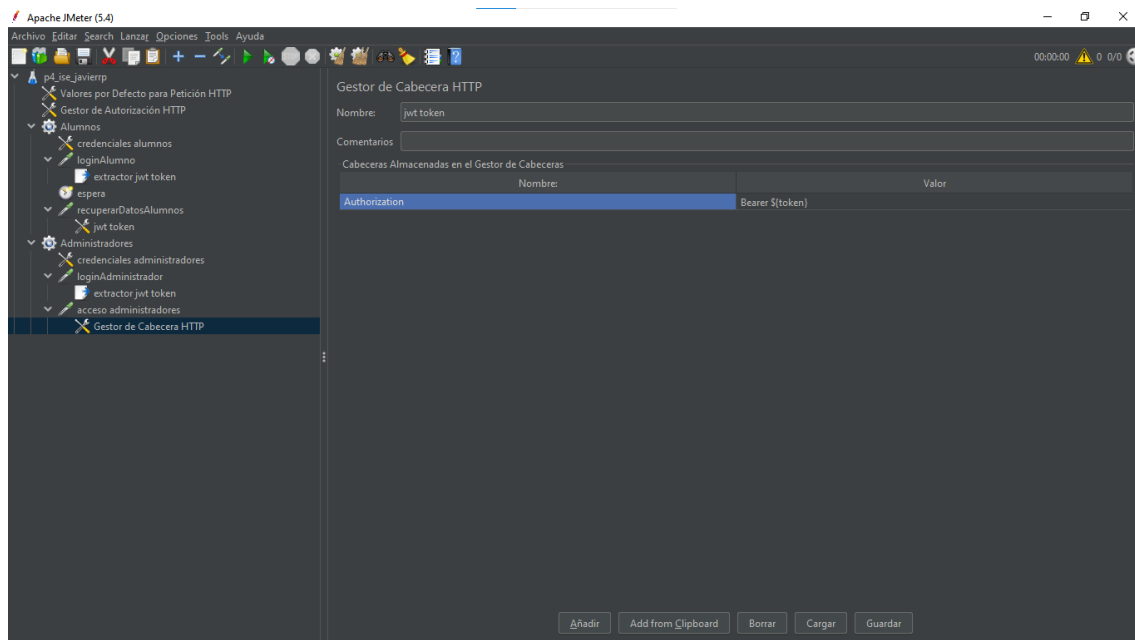
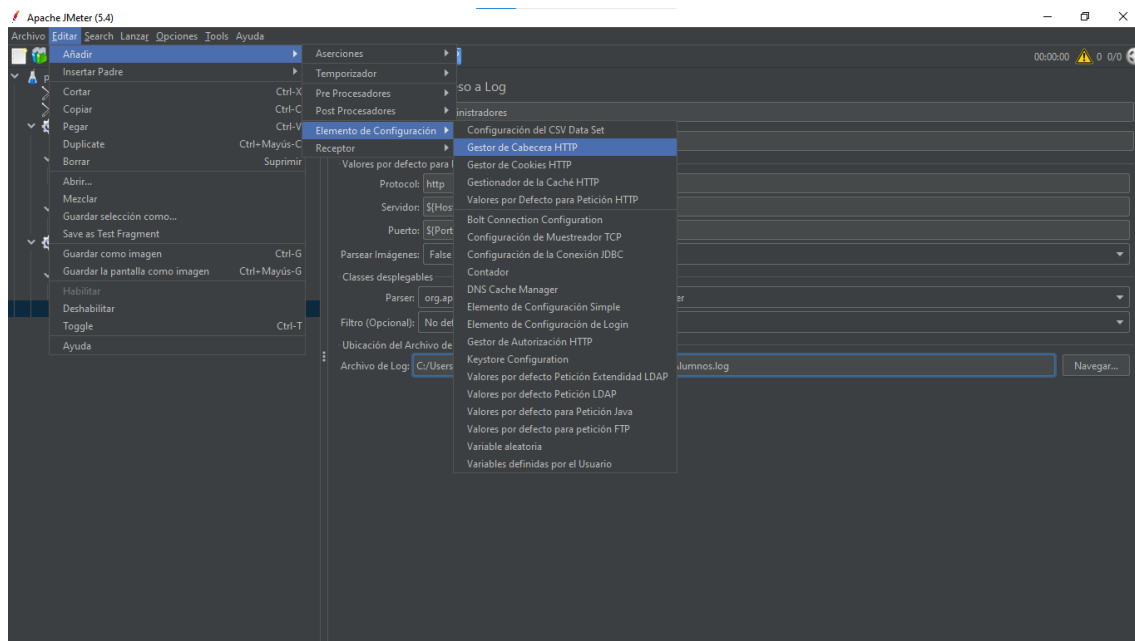


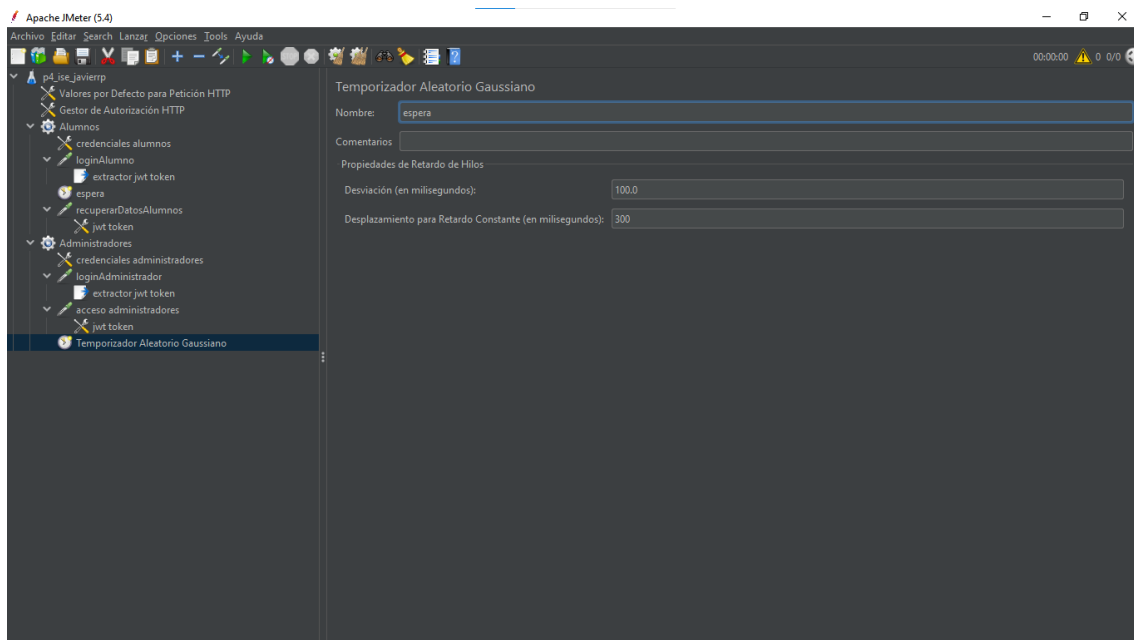
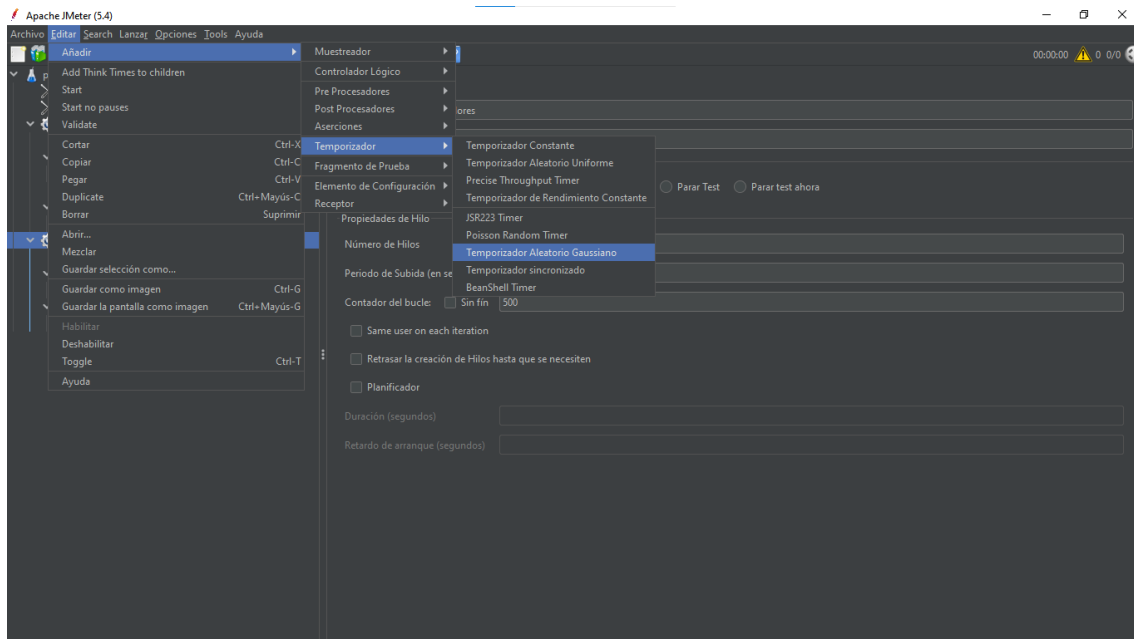




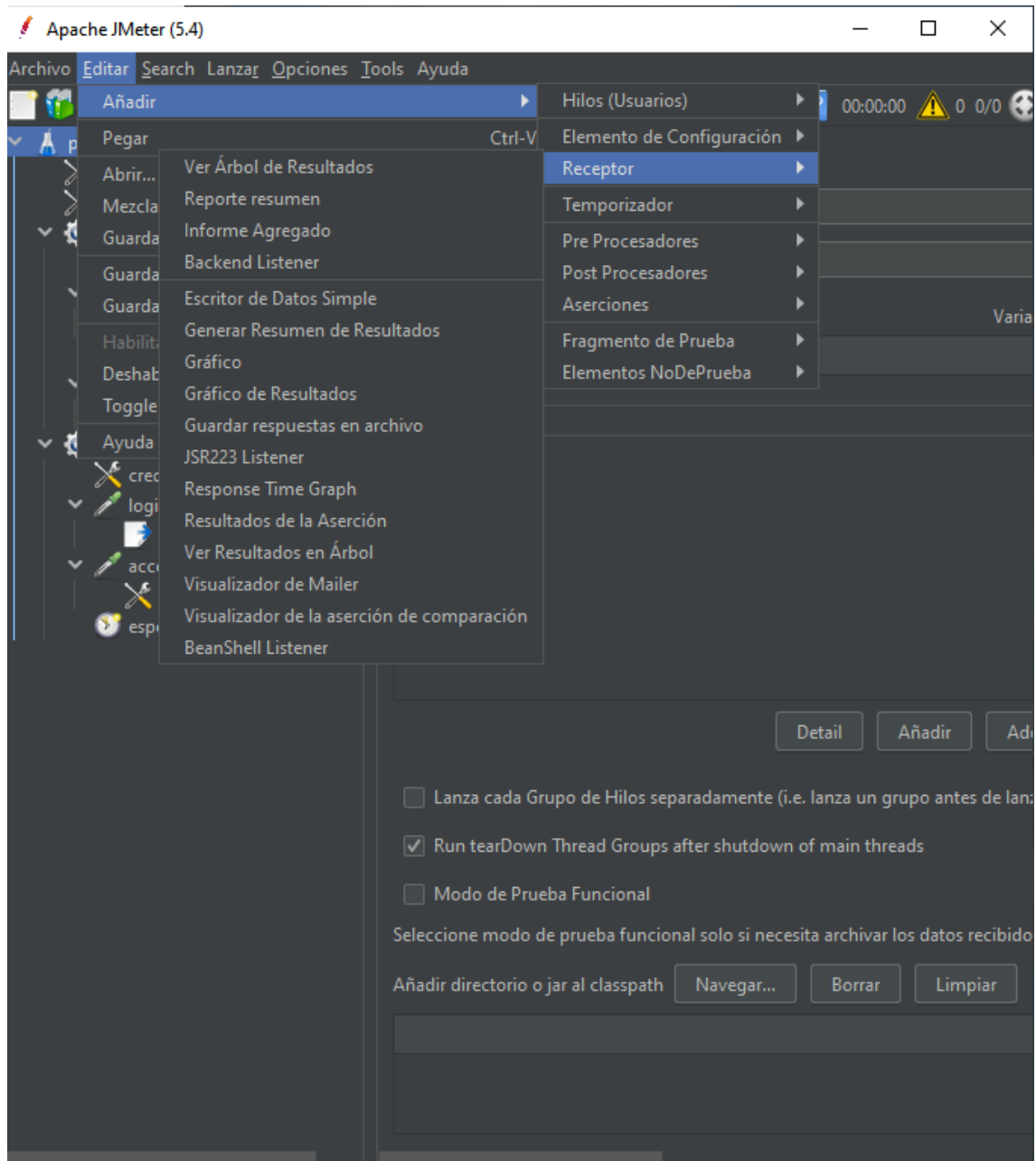


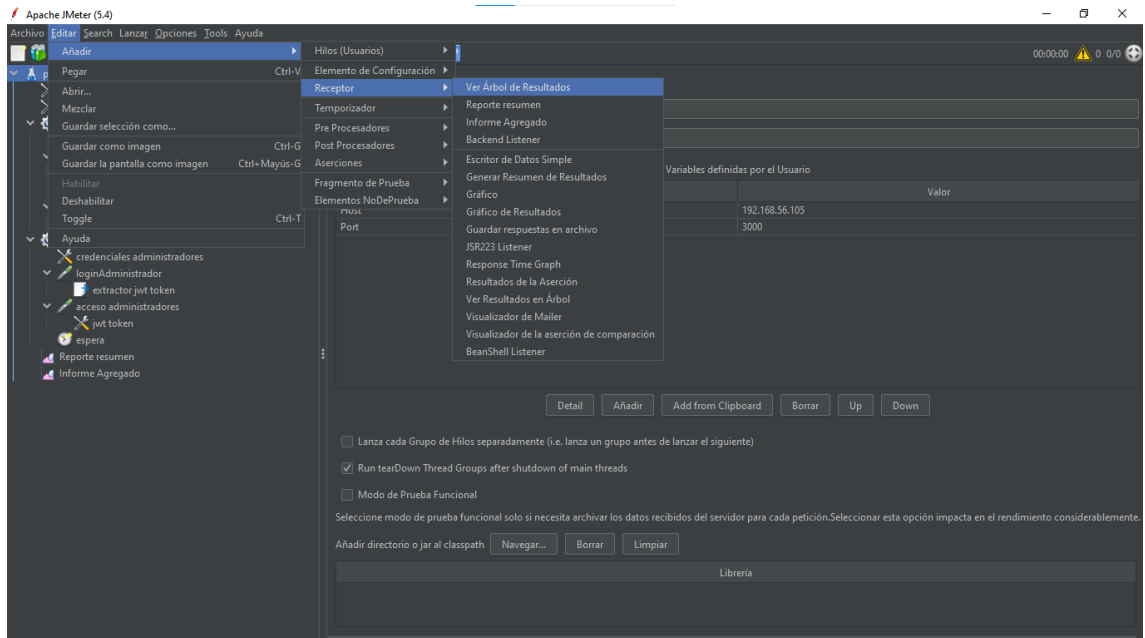
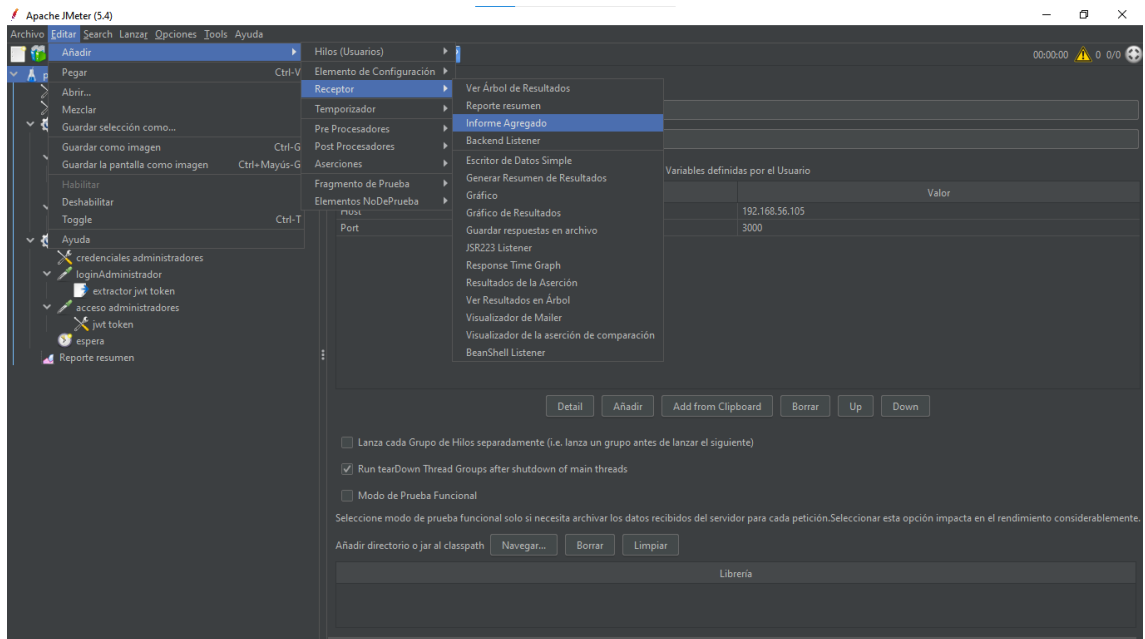




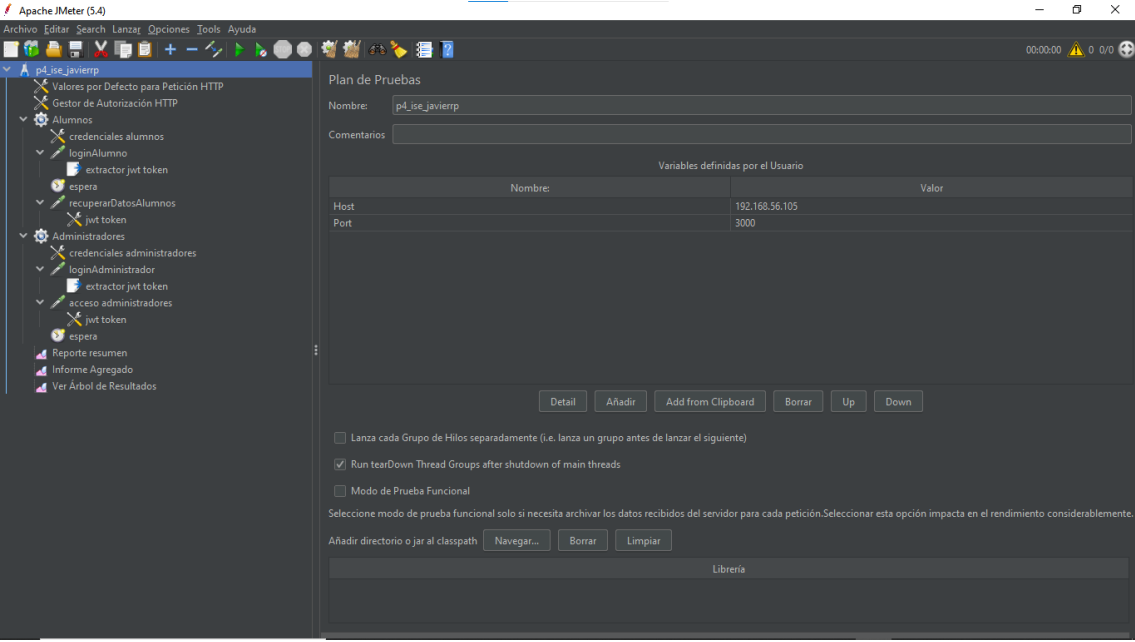


Y aquí habría finalizado también la configuración de los administradores. Por último, se añaden algunos elementos que nos informen de cómo han ido los datos, accesos y acciones que se realicen sobre Ubuntu.

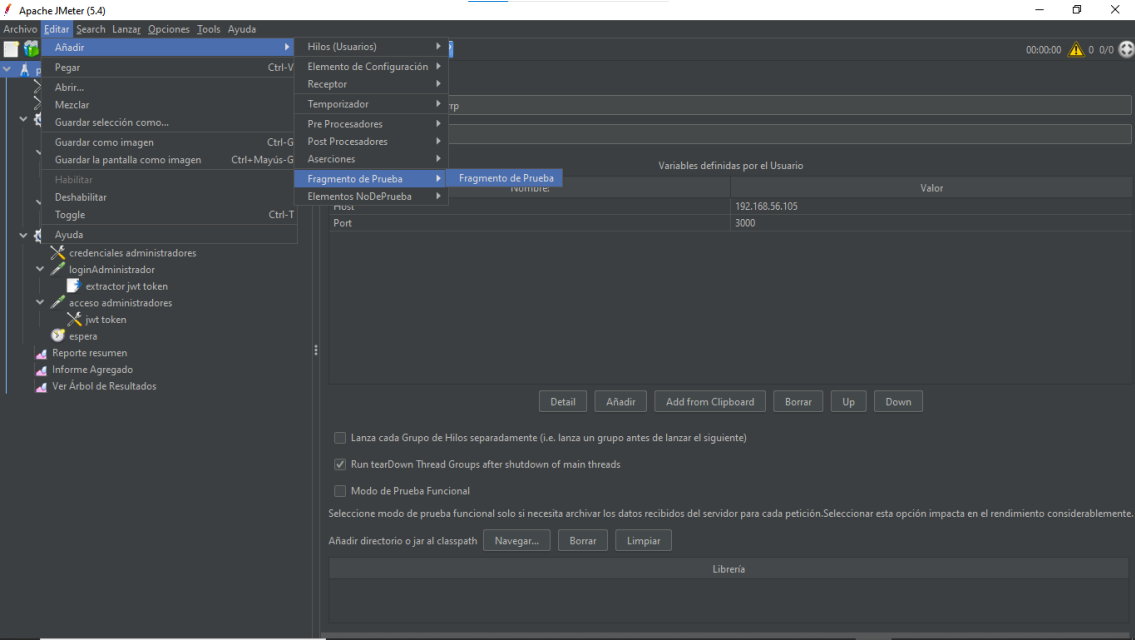


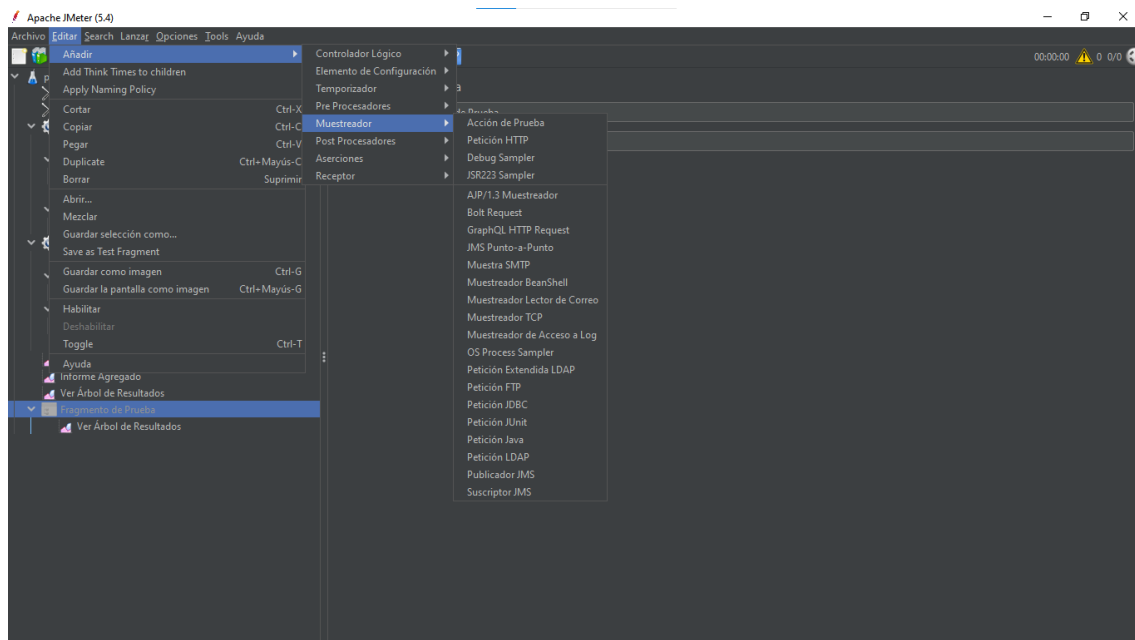
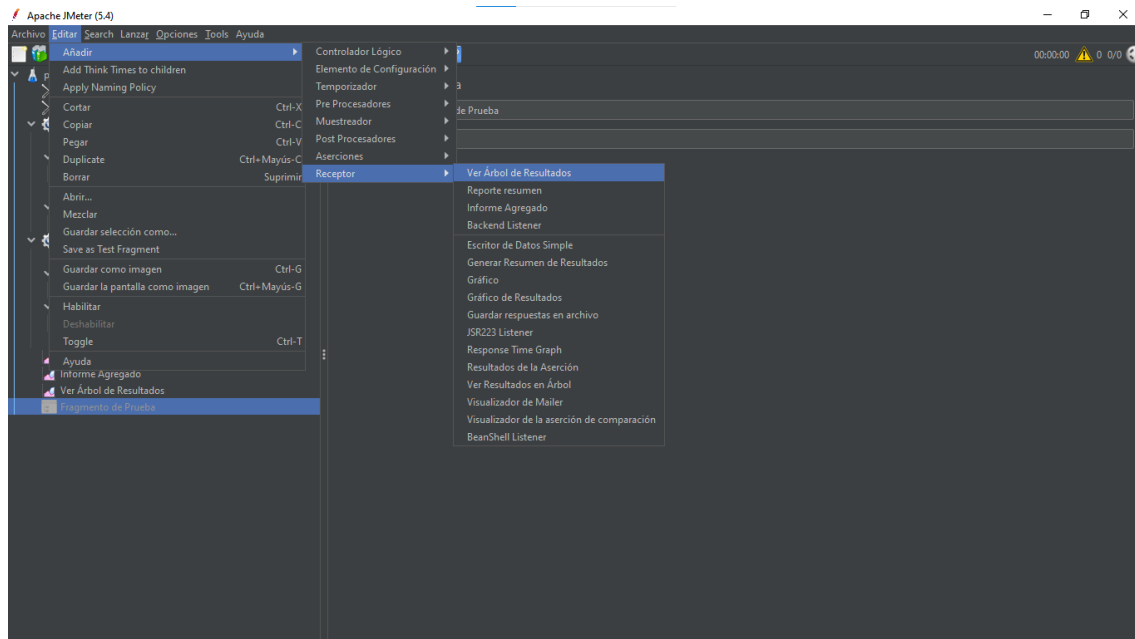


Quedando una configuración final de este modo para poder observar bien el árbol y entender a qué nivel debe haber quedado anidado:



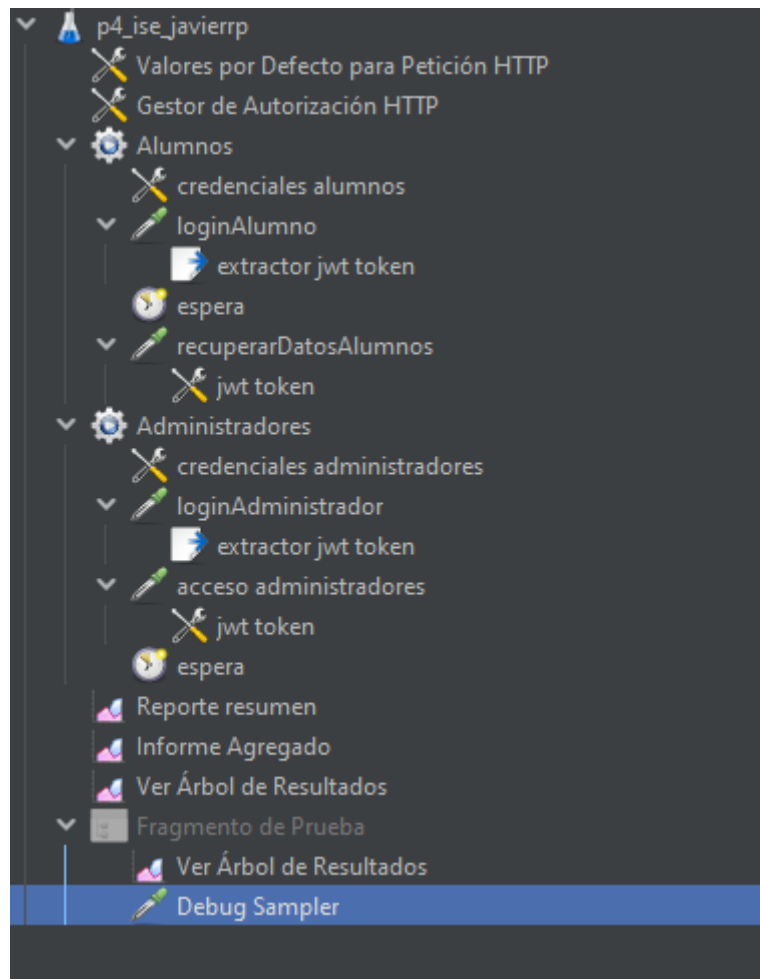
Añadimos un fragmento de prueba para ver algunos resultados.



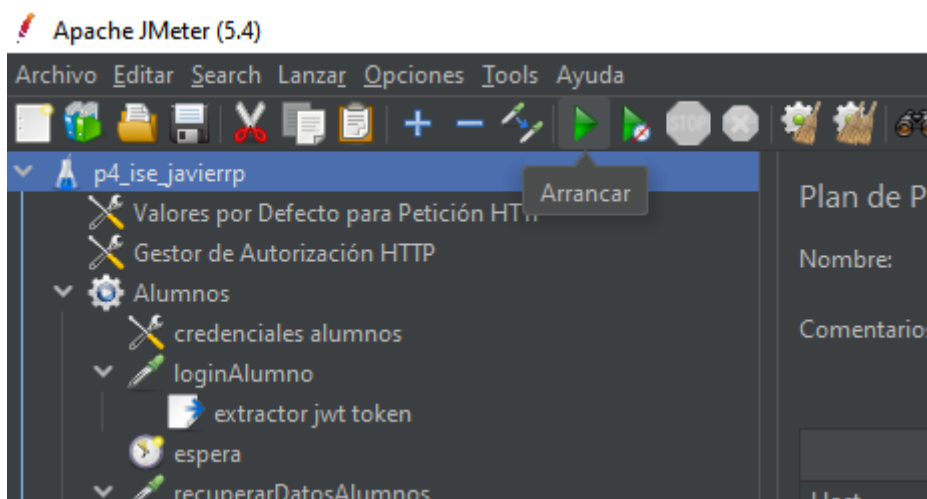




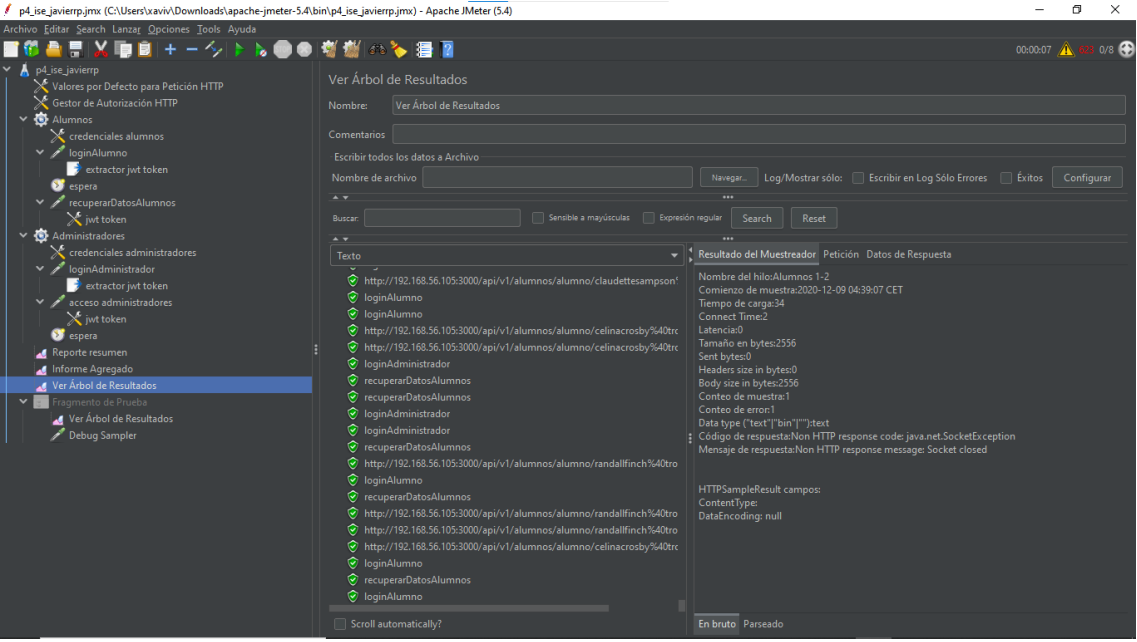
## CONFIGURACIÓN FINAL



Lo ejecutamos:



Y estas son algunas de las imágenes de los resultados que obtengo:



p4\_ise\_javerrp.jmx (C:\Users\xavi\Downloads\apache-jmeter-5.4.0\bin\p4\_ise\_javerrp.jmx) - Apache JMeter (5.4)

Archivo Editor Search Listas Opciones Tools Ayuda

00:02:16 8/8

pd\_ise\_javerrp

- Valores por Defecto para Petición HTTP
- Gestor de Autorización HTTP
- Alumnos
  - credenciales alumnos
  - loginAlumno
  - extractor jwt token
  - espera
  - recuperarDatosAlumnos
  - jwt token
- Administradores
  - credenciales administradores
  - loginAdministrador
  - extractor jwt token
  - acceso administradores
  - jwt token
  - espera
  - Reporte resumen
  - Informe Agregado**
  - Ver Árbol de Resultados
  - Fragmento de Prueba
  - Ver Árbol de Resultados
  - Debug Sampler

### Informe Agregado

Nombre: Informe Agregado

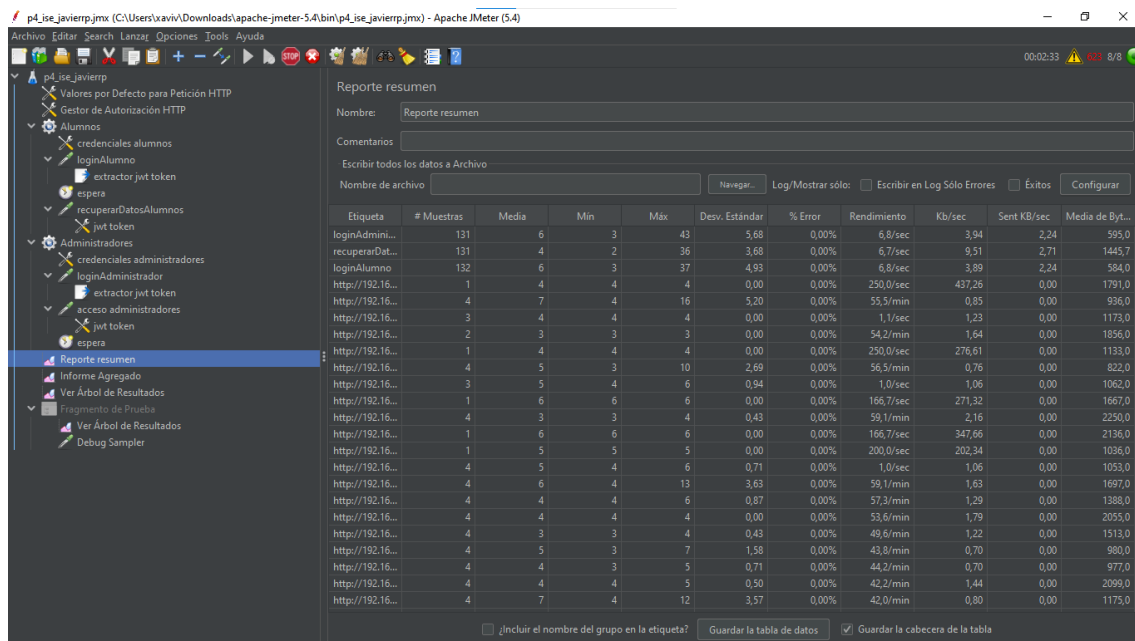
Comentarios

Escribir todos los datos a Archivo

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

Etiqueta	# Muestras	Media	Mediana	90% Line	95% Line	99% Line	Min	Máx.	% Error	Rendim...	Kb/sec	Sent Kb/sec
loginAlu...	127	6	5	12	16	30	4	34	0,00%	6,7/sec	3,84	2,21
loginAdm...	127	7	5	14	20	24	3	34	0,00%	6,7/sec	3,91	2,22
https://192...	2	20	3	37	37	37	3	37	0,00%	54,5/min	0,87	0,00
recuperar...	126	5	4	11	13	20	2	22	0,00%	6,8/sec	9,37	2,73
https://192...	4	4	4	6	6	6	4	6	0,00%	1,1/sec	1,58	0,00
https://192...	2	12	10	15	15	15	10	15	0,00%	50,4/min	1,17	0,00
https://192...	1	18	18	18	18	18	18	18	0,00%	55,6/sec	92,45	0,00
https://192...	4	10	6	20	20	20	3	20	0,00%	1,0/sec	1,58	0,00
https://192...	4	6	4	9	9	9	4	9	0,00%	55,7/min	0,93	0,00
https://192...	2	8	5	11	11	11	5	11	0,00%	54,2/min	1,92	0,00
https://192...	1	9	9	9	9	9	9	9	0,00%	111,1/sec	101,67	0,00
https://192...	4	5	4	8	8	8	4	8	0,00%	55,7/min	1,43	0,00
https://192...	4	4	4	6	6	6	4	6	0,00%	53,9/min	1,55	0,00
https://192...	4	6	5	11	11	11	3	11	0,00%	54,3/min	1,11	0,00
https://192...	4	5	5	7	7	7	4	7	0,00%	56,3/min	0,88	0,00
https://192...	4	6	4	15	15	15	4	15	0,00%	56,4/min	1,65	0,00
https://192...	4	8	5	16	16	16	4	16	0,00%	51,9/min	1,07	0,00
https://192...	4	4	4	6	6	6	3	6	0,00%	54,2/min	1,85	0,00
https://192...	4	4	4	5	5	5	4	5	0,00%	54,6/min	1,26	0,00
https://192...	4	4	4	8	8	8	3	8	0,00%	51,1/min	0,98	0,00
https://192...	4	4	4	5	5	5	4	5	0,00%	51,8/min	1,03	0,00
https://192...	4	7	6	10	10	10	3	10	0,00%	51,3/min	1,41	0,00
https://192...	4	4	4	7	7	7	4	7	0,00%	51,5/min	1,38	0,00

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



Y por último, vemos la página a la que accedemos si buscamos en el navegador nuestra IP especificando el puerto de JMeter.



## ETSII Alumnos API

Descripción de la API Restful:

POST /api/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