

Practica 4: Documentación

Ejercicio 1

Instalación de Phoronix Ubuntu

Si bien Phoronix Test Suite está disponible en los repositorios predeterminados de Ubuntu, y por tanto podríamos instalarlo fácilmente con

```
sudo apt-get install phoronix -test -suite
```

, los paquetes en los repositorios son probablemente muy viejos. por lo tanto, es aconsejable obtener la última versión de la página de descarga oficial. En el momento de realización de esta práctica, la versión estable es la 10.2.2, por tanto ejecutaremos

```
wget http://phoronix-test-suite.com/releases/repo/pts.debian/files/phoronix-test-suite_10.2.2_all.deb
```

Y tras esto realizaremos la instalación utilizando

```
sudo dpkg -i phoronix-test-suite_10.2.0_all.deb
```

Tras lo anterior utilizaremos el comando

```
sudo apt -f install
```

, para resolver cualquier problema de dependencias.

```
momid@momid:~$ wget http://phoronix-test-suite.com/releases/repo/pts.debian/files/phoronix-test-suite_10.2.2_all.deb
--2021-05-13 14:29:41-- http://phoronix-test-suite.com/releases/repo/pts.debian/files/phoronix-test-suite_10.2.2_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: 1091024 (1.0M) [application/x-debian-package]
Saving to: 'phoronix-test-suite_10.2.2_all.deb'

phoronix-test-suite_10.2 100%[=====] 1.04M 1006KB/s in 1.1s

2021-05-13 14:29:42 (1006 KB/s) - 'phoronix-test-suite_10.2.2_all.deb' saved [1091024/1091024]

momid@momid:~$ sudo dpkg -i phoronix-test-suite_10.2.2_all.deb
[sudo] password for momid:
Selecting previously unselected package phoronix-test-suite.
(Reading database ... 74736 files and directories currently installed.)
Preparing to unpack phoronix-test-suite_10.2.2_all.deb ...
Unpacking phoronix-test-suite (10.2.2) ...
Setting up phoronix-test-suite (10.2.2) ...
Processing triggers for mime-support (3.64ubuntu1) ...
Processing triggers for man-db (2.9.1-1) ...
momid@momid:~$ sudo apt -f install
Reading package lists... Done
Building dependency tree
Reading state information... Done
0 upgraded, 0 newly installed, 0 to remove and 97 not upgraded.
momid@momid:~$
```

Para comprobar que hemos instalado Phoronix con éxito podemos utilizar el comando

```
phoronix-test-suite help
```

, y tras aceptar la licencia de uso nos mostrará una lista de los comandos disponibles.

Instalación Phoronix CentOS

El único requerimiento obligatorio para Phoronix Test Suite es PHP CLI, luego lo instalamos con el comando

```
sudo yum install php-cli wget
```

También debemos instalar wget, para tras esto descargarnos el paquete desde la página.

Tras esto procedemos a descargarnos el paquete de instalación desde la página de Phoronix Test Suite, con el comando

```
wget https://phoronix-test-suite.com/releases/phoronix-test-suite-10.2.2.tar.gz
```

Ahora lo descomprimos con

```
tar -xzf phoronix-test-suite-10.2.2.tar.gz
```

y tras esto nos metemos en la carpeta y ejecutamos en el instalador, con los comando

```
cd phoronix-test-suite
```

```
sudo ./install-sh
```

Una vez hecho esto podemos comprobar que está correctamente instalado con

```
phoronix-test-suite help
```

En el caso de que nos ponga que se requiere alguna extensión de PHP para el funcionamiento hacemos

```
sudo yum install php-extensión
```

Tras instalar todas las dependencias que se nos exigen volvemos a probar el comando anterior y debería funcionar.

Benchmarking

compress-7zip

En primer lugar se ha elegido un benchmark de procesador denominado **7-Zip Compression**, el cual tiene un tiempo medio de ejecución de un minuto y medio, y cuya única dependencia es el compilador de C/C++. Para correr el test ejecutamos el comando

```
phoronix-test-suite benchmark compress-7zip
```

Una vez terminada la ejecución de los Benchmark, como es un benchmark que calcula la velocidad de compresión del procesador, nos muestra el número de compresiones realizadas en las tres ejecuciones que se han realizado.

UbuntuServer

Por su lado UbuntuServer nos ha proporcionado los siguientes datos

```

    Started Run 10 @ 14:12:58 *
    Started Run 11 @ 14:13:46 *
    Started Run 12 @ 14:14:34 *
    Started Run 13 @ 14:15:22 *
    Started Run 14 @ 14:16:10 *
    Started Run 15 @ 14:16:58 *

Compress Speed Test:
    3027
    3553
    3613
    3437
    3401
    3397
    3358
    3447
    3332
    3223
    3172
    3159
    3224
    3158
    3205

Average: 3314 MIPS
Deviation: 4.94%
Samples: 15

Comparison to 9,479 OpenBenchmarking.org samples since 4 January 2012; median result: 21412. Box
plot of samples:
[-#####!#####*#####-----*-*-*-----|*]
    ^ Apple M1: 38899    ARMv8 Neoverse-N1: 185312 ^          AMD EPYC 7713: 290812 ^
    AMD Ryzen Threadripper 3990X: 179399 ^
    2 x Intel Xeon Gold 6230N: 172157 ^
    POWER9: 165140 ^

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

Do you want to view the text results of the testing (Y/n): y
compress_Ubuntu
compress-7zip UbuntuServer

run:

    Processor: Intel Core i5-8300H (1 Core), Motherboard: Oracle VirtualBox v1.2, Chipset: Intel
    440FX 82441FX PMC, Memory: 4096MB, 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: 5.4.0-67-generic (x86_64), Compiler: GCC 9.3.0, File-System: ext4,
    Screen Resolution: 2048x2048, System Layer: Oracle VMWare

    7-Zip Compression 16.02
    Compress Speed Test
    MIPS > Higher Is Better
    run . 3314 |=====

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

```

Vemos que en la mejor de las ejecuciones obtiene una puntuación de 3613, mientras que en la peor obtiene una puntuación de 3027.

Además si cogemos la carpeta que contiene los resultados del test y la movemos a la carpeta `/var/www/html/` entonces podremos consultar en el navegador la siguiente información

System Information

compress_Ubuntu


Phoronix Test Suite 10.2.2	
Intel Core i5-8300H (1 Core)	Processor
Oracle VirtualBox v1.2	Motherboard
Intel 440FX 82441FX PMC	Chipset
4096MB	Memory
2 x 11GB VBOX HDD	Disk
VMware SVGA II	Graphics
Intel 82801AA AC 97 Audio	Audio
2 x Intel 82540EM	Network
Ubuntu 20.04	OS
5.4.0-67-generic (x86_64)	Kernel
GCC 9.3.0	Compiler
ext4	File System
2048x2048	Screen Resolution
Oracle VMware	System Layer

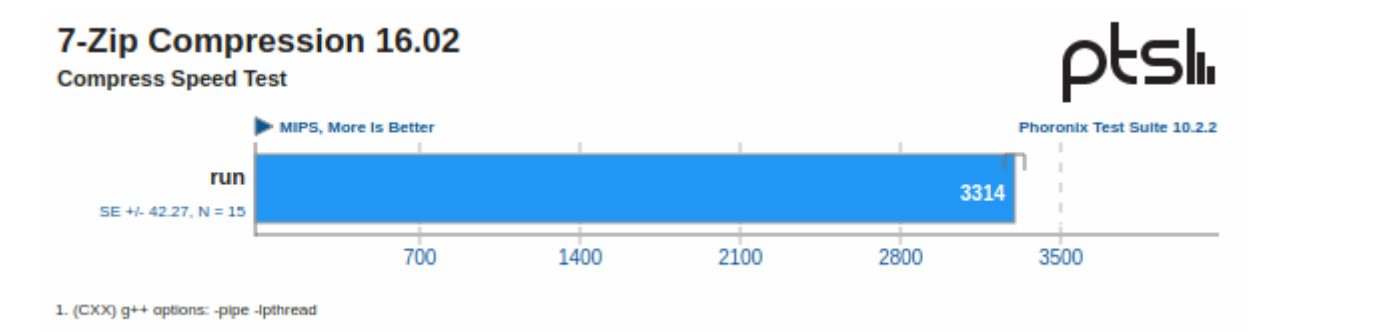
- Transparent Huge Pages: madvise

--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-
--enable-libstdc++-debug
--enable-libstdc++-time=yes
--enable-multilib --enable-multilib
--enable-nls --enable-objc-gc=auto
--enable-offload-targets=nvptx-none=/build/gcc-9-HskZE
--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-libstdc++-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

- itlb_multihit: KVM: Vulnerable + !tft:
Mitigation of PTE Inversion + mds:
Mitigation of Clear buffers; SMT Host
state unknown + meltdown: Mitigation of
PTI + spec_store_bypass: Vulnerable +
spectre_v1: Mitigation of
usercopy/swapgs barriers and __user
pointer sanitization + spectre_v2:
Mitigation of Full generic retpoline

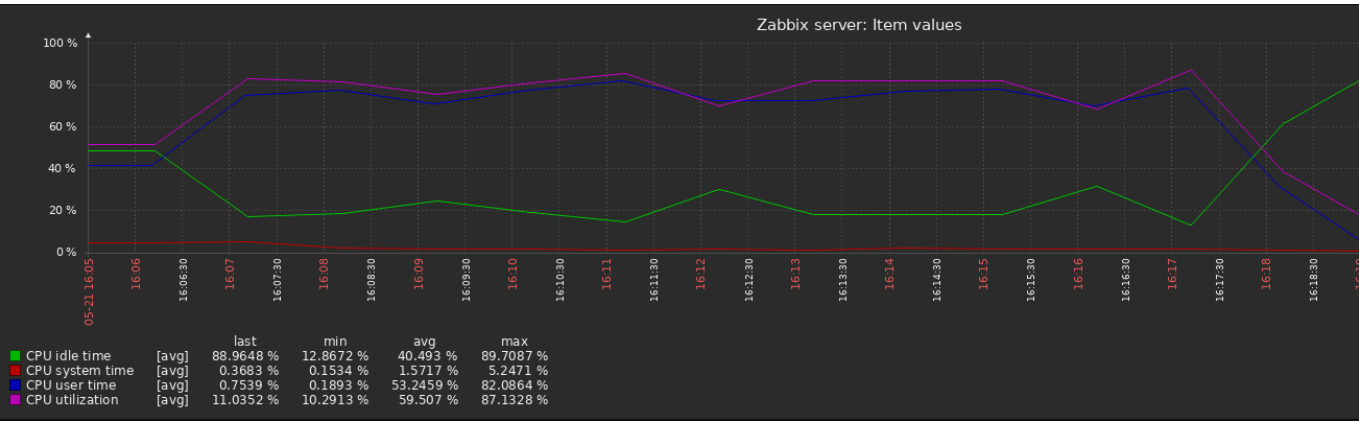
Results Overview

compress_Ubuntu		
	run	
compress-7zip: Compress Speed Test	3314	
Standard Error	42.27	
Standard Deviation	4.94%	
Phoronix Test Suite 10.2.2		



Podemos ver que en todas las ejecuciones realizadas, la media de UbuntuServer son 3314 puntos

Por otro lado, como configuré zabbix para que monitorizase el uso del procesador podemos ver cual es el uso del mismo



Si nos fijamos en los datos comprendidos entre la 16:05 y las 16:18 vemos que se ve un aumento significativo del uso de la CPU, como cabría esperar durante el tiempo de ejecución del Benchmark.

Aclarar que las horas no coinciden, porque la máquina virtual tiene una hora incorrecta configurada, sin embargo, los minutos si coinciden. También se observa que no pasa en ningún momento la utilización del 90%, luego zabbix no nos va a monitorizar ningún problema, pues el umbral que impone zabbix es el 90% de utilización de la CPU.

CentOS

Por su parte CentOS nos ha proporcionado los siguientes datos

```

Started Run 10 @ 10:12:03 *
Started Run 11 @ 10:12:46 *
Started Run 12 @ 10:13:29 *
Started Run 13 @ 10:14:14 *
Started Run 14 @ 10:14:59 *
Started Run 15 @ 10:15:43 *

Compress Speed Test:
3682
3628
3952
3995
3956
3857
3821
3865
3809
3834
3945
3608
3559
3685
3564

Average: 3784 MIPS
Deviation: 3.99%
Samples: 15

Comparison to 9,479 OpenBenchmarking.org samples since 4 January 2012; median result: 21412. Box
plot of samples:
[#####!#####*#####-----*--*--*-----| * ]
      ^ Apple M1: 38899      ARMv8 Neoverse-N1: 185312 ^      AMD EPYC 7713: 290812 ^
      AMD Ryzen Threadripper 3990X: 179399 ^
      2 x Intel Xeon Gold 6230N: 172157 ^
      POWER9: 165140 ^

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

Do you want to view the text results of the testing (Y/n): y
compress_CentOS
compress-7zip CentOS

run:

Processor: Intel Core i5-8300H (1 Core), Motherboard: Oracle VirtualBox v1.2, Chipset: Intel
440FX 82441FX PMC, Memory: 4096MB, Disk: 2 x 9GB VBOX HDD, Graphics: VMware SVGA II, Audio: Intel 8
2801AA AC 97 Audio, Network: 2 x Intel 82540EM

OS: CentOS Linux 8, Kernel: 4.18.0-193.el8.x86_64 (x86_64), File-System: xfs, Screen Resolut
ion: 2048x2048, System Layer: Oracle VMware

7-Zip Compression 16.02
Compress Speed Test
MIPS > Higher Is Better
run . 3784 |=====

Would you like to upload the results to OpenBenchmarking.org (y/n): _

```

Vemos que la mejor de las ejecuciones da 3995 puntos y la peor da 3559 puntos. Además repitiendo el proceso de mover la carpeta de resultados podemos observar que


System Information

compress_CentOS

Phoronix Test Suite 10.2.2		Phoronix Test Suite 10.2.2
Intel Core i5-8300H (1 Core)	Processor	
Oracle VirtualBox v1.2	Motherboard	
Intel 440FX 82441FX PMC	Chipset	
4096MB	Memory	
2 x 9GB VBOX HDD	Disk	
VMware SVGA II	Graphics	
Intel 82801AA AC 97 Audio	Audio	
2 x Intel 82540EM	Network	
CentOS Linux 8	OS	
4.18.0-193.el8.x86_64 (x86_64)	Kernel	
xfs	File-System	
2048x2048	Screen Resolution	
Oracle VMware	System Layer	

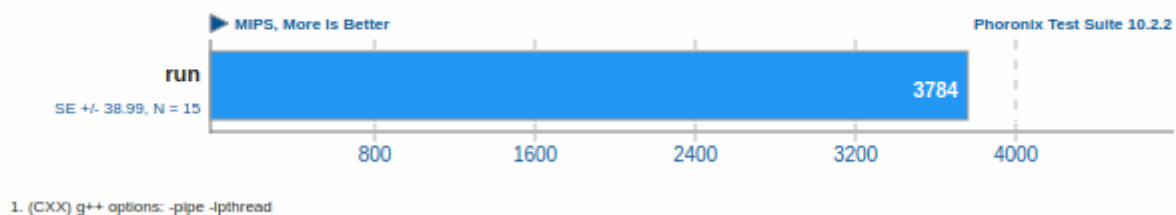
- Transparent Huge Pages: always
- SELinux + itlb_multihit: KVM: Vulnerable + 11tf: Mitigation of PTE Inversion + mds: Mitigation of Clear buffers; SMT Host state unknown + meltdown: Mitigation of PTI + spec_store_bypass: Vulnerable + spectre_v1: Mitigation of usercopy/swapgs barriers and __user pointer sanitization + spectre_v2: Mitigation of Full generic retpoline STIBP: disabled RSB filling + tsx_async_abort: Not affected

Results Overview

compress_CentOS		
	run	
compress-7zip: Compress Speed Test	3784	
Standard Error	38.99	
Standard Deviation	3.99%	
Phoronix Test Suite 10.2.2		

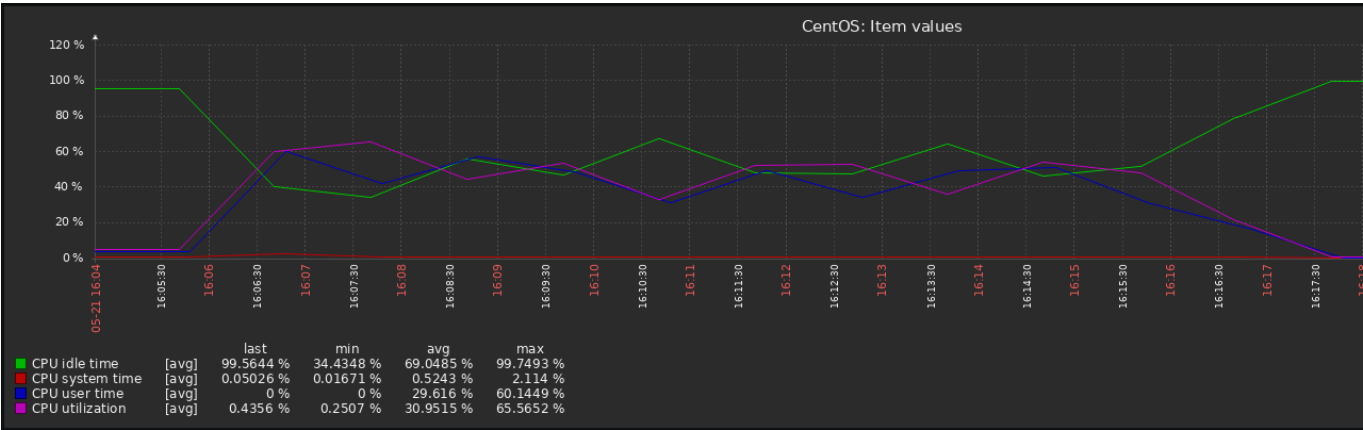
7-Zip Compression 16.02

Compress Speed Test



Podemos ver que la media de las ejecuciones de CentOS son 3784 puntos

Por otro lado, Zabbix nos muestra lo siguiente en cuanto a utilización de la CPU.



Si nos fijamos en la franja horaria entre las 16:04 y las 16:17, que son la hora de inicio y finalizaci3n del test, vemos un pico en la uiltilizaci3n del mismo. Esta vez podemos observar que aunque la utilizaci3n de la CPU ha crecido bastante al empezar el test no ha pasado del 66%, mucho menos en comparaci3n con la utilizaci3n de la CPU obtenida en Ubuntu.

Conclusiones

Vemos que el resultado obtenido por UbuntuServer es peor que el obtenido por CentOS, a pesar de que la utilizaci3n del procesador ha sido mayor (el mejor resultado obtenido por CentOS es un 10.57% mejor que el mejor resultado obtenido por UbuntuServer, siendo la media de CentOS un 14.18% mejor que la media de UbuntuServer). Esto se puede deber entre otras cosas a que como ambas m1quinas tienen asignado un 3nico n3cleo y, sin embargo, UbuntuServer tiene m1s funciones que atender, pues es el encargado de monitorizar con Zabbix, la carga extra se deba a la carga de trabajo que pueda suponer Zabbix y la correspondiente monitorizaci3n que mantiene. Tambi3n podemos observar que en comparaci3n con otros resultados a este mismo test obtenidos por procesadores dedicados a servidores nuestro resultado es muy bajo.

Por ejemplo, fij1ndonos en el resultado obtenido por un AMD EPYC vemos que su resultado es de 298812, es decir, casi 70 veces mejor que el mejor de nuestros resultados.

cryptsetup

Se trata de un benchmark que mide el rendimiento en criptograf1a del sistema. Es un test cuya 3nica dependencia es tener instalado cryptsetup y tiene una duraci3n media de dos minutos y medio. Adem1s se trata de un test de sistema en contraposici3n al test de procesador que ya hemos realizado.

UbuntuServer

UbuntuServer nos ha dado por la terminal los siguientes resultados

9 / 35

```
Cryptsetup
AES-XTS 512b Encryption
MiB/s > Higher Is Better
run . 1569.8 |=====
```

```
Cryptsetup
AES-XTS 512b Decryption
MiB/s > Higher Is Better
run . 1536.2 |=====
```

```
Cryptsetup
Serpent-XTS 512b Encryption
MiB/s > Higher Is Better
run . 573.5 |=====
```

```
Cryptsetup
Serpent-XTS 512b Decryption
MiB/s > Higher Is Better
run . 583.2 |=====
```

```
Cryptsetup
Twofish-XTS 512b Encryption
MiB/s > Higher Is Better
run . 314.6 |=====
```

```
Cryptsetup
Twofish-XTS 512b Decryption
MiB/s > Higher Is Better
run . 316.8 |=====
```

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

Se puede apreciar que la media son **316.8 MiB/s** y la posición en la clasificación.

Por otro lado en la interfaz web se nos muestra

cryptsetup_result_Ubuntu

Phoronix Test Suite 10.2.2	
Intel Core i5-8300H (1 Core)	Processor
Oracle VirtualBox v1.2	Motherboard
Intel 440FX 82441FX PMC	Chipset
4096MB	Memory
2 x 11GB VBOX HDD	Disk
VMware SVGA II	Graphics
Intel 82801AA AC 97 Audio	Audio
2 x Intel 82540EM	Network
Ubuntu 20.04	OS
5.4.0-67-generic (x86_64)	Kernel
GCC 9.3.0	Compiler
ext4	File-System
2048x2048	Screen Resolution
Oracle VMware	System Layer

- Transparent Huge Pages: madvise
 - itlb_multihit: KVM: Vulnerable + !itf: Mitigation of PTE Inversion + mds: Mitigation of Clear buffers; SMT Host state unknown + meltdown: Mitigation of PTI + spec_store_bypass: Vulnerable + spectre_v1: Mitigation of usercopy/swaps barriers and __user pointer sanitization + spectre_v2: Mitigation of Full generic retpoline STIBP: disabled RSB filling + srbds: Unknown: Dependent on hypervisor status + tsx_async_abort: Not affected

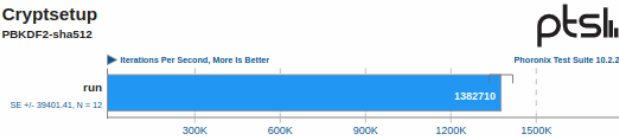
Results Overview

cryptsetup_result_Ubuntu		
ptsli	run	
cryptsetup: PBKDF2-sha512	1382710	
cryptsetup: PBKDF2-whirlpool	579643	
cryptsetup: AES-XTS 256b Encryption	1694.0	
cryptsetup: AES-XTS 256b Decryption	1646.6	
cryptsetup: Serpent-XTS 256b Encryption	590.3	
cryptsetup: Serpent-XTS 256b Decryption	599.5	
cryptsetup: Twofish-XTS 256b Encryption	320.9	
cryptsetup: Twofish-XTS 256b Decryption	317.7	
cryptsetup: AES-XTS 512b Encryption	1569.8	
cryptsetup: AES-XTS 512b Decryption	1536.2	
cryptsetup: Serpent-XTS 512b Encryption	573.5	
cryptsetup: Serpent-XTS 512b Decryption	583.2	
cryptsetup: Twofish-XTS 512b Encryption	314.6	
cryptsetup: Twofish-XTS 512b Decryption	316.8	

Phoronix Test Suite 10.2.2

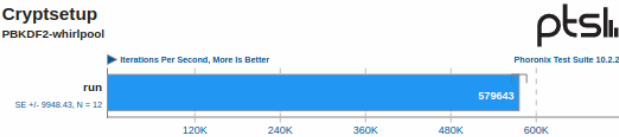
Cryptsetup

PBKDF2-sha512



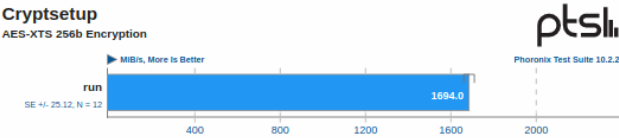
Cryptsetup

PBKDF2-whirlpool



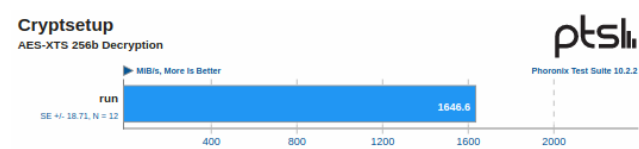
Cryptsetup

AES-XTS 256b Encryption



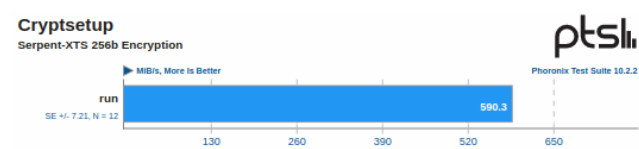
Cryptsetup

AES-XTS 256b Decryption



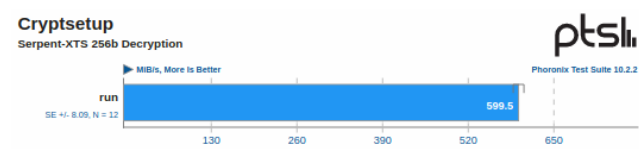
Cryptsetup

Serpent-XTS 256b Encryption



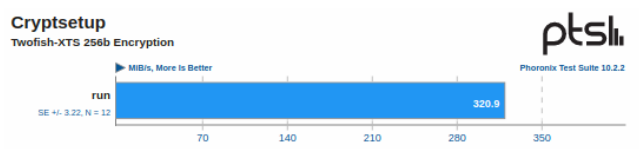
Cryptsetup

Serpent-XTS 256b Decryption



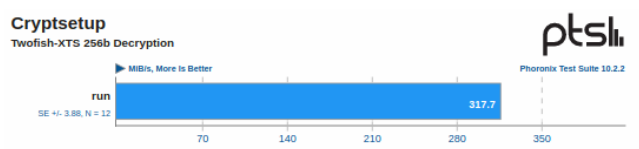
Cryptsetup

Twofish-XTS 256b Encryption



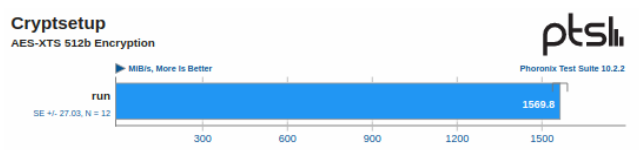
Cryptsetup

Twofish-XTS 256b Decryption



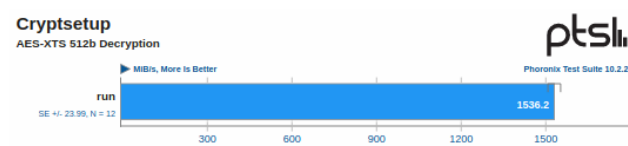
Cryptsetup

AES-XTS 512b Encryption



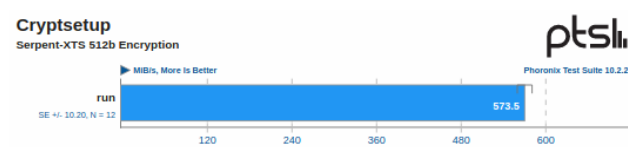
Cryptsetup

AES-XTS 512b Decryption



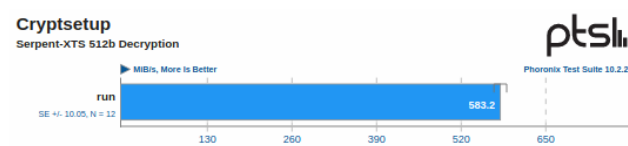
Cryptsetup

Serpent-XTS 512b Encryption



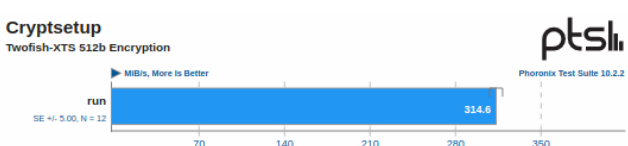
Cryptsetup

Serpent-XTS 512b Decryption



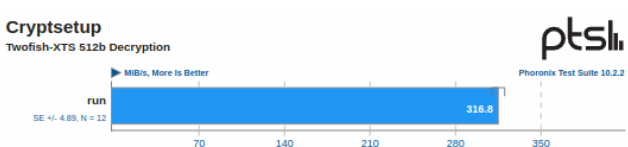
Cryptsetup

Twofish-XTS 512b Encryption



Cryptsetup

Twofish-XTS 512b Decryption



Donde nos da información del sistema, una vista general de los resultados obtenidos y un gráfico con los resultados obtenidos para cada tipo de encriptación. Además podemos ver la carga extra que se ha puesto sobre la CPU en zabbix.



```
Cryptsetup
AES-XTS 512b Encryption
MiB/s > Higher Is Better
run . 1650.4 |=====
```

```
Cryptsetup
AES-XTS 512b Decryption
MiB/s > Higher Is Better
run . 1611.0 |=====
```

```
Cryptsetup
Serpent-XTS 512b Encryption
MiB/s > Higher Is Better
run . 594.6 |=====
```

```
Cryptsetup
Serpent-XTS 512b Decryption
MiB/s > Higher Is Better
run . 593.8 |=====
```

```
Cryptsetup
Twofish-XTS 512b Encryption
MiB/s > Higher Is Better
run . 326.8 |=====
```

```
Cryptsetup
Twofish-XTS 512b Decryption
MiB/s > Higher Is Better
run . 323.8 |=====
```

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

, donde se puede apreciar que la media obtenida es **323.8 MiB/s**, quedando en un percentil más alto que UbuntuServer, aunque se debe destacar que se han realizado tres ejecuciones más que en UbuntuServer. Por su parte la interfaz web nos muestra lo siguiente


System Information

cryptsetup_result_CentOS

Phoronix Test Suite 10.2.2		Phoronix Test Suite 10.2.2
Intel Core i5-8300H (1 Core)	Processor	
Oracle VirtualBox v1.2	Motherboard	
Intel 440FX 82441FX PMC	Chipset	
4096MB	Memory	
2 x 9GB VBox HDD	Disk	
VMware SVGA II	Graphics	
Intel 82801AA AC 97 Audio	Audio	
2 x Intel 82540EM	Network	
CentOS Linux 8	OS	
4.18.0-193.el8.x86_64 (x86_64)	Kernel	
xfs	File-System	
2048x2048	Screen Resolution	
Oracle VMware	System Layer	

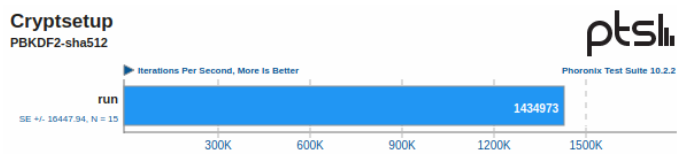
- Transparent Huge Pages: always
- SELinux + itlb_multihit: KVM: Vulnerable + !1tf: Mitigation of PTE Inversion + mds: Mitigation of Clear buffers; SMT Host state unknown + meltdown: Mitigation of PTI + spec_store_bypass: Vulnerable + spectre_v1: Mitigation of usercopy/swaps barriers and __user pointer sanitization + spectre_v2: Mitigation of Full generic retpoline STIBP: disabled RSB filling + tsx_async_abort: Not affected

Results Overview

cryptsetup_result_CentOS		
	run	
cryptsetup: PBKDF2-sha512	1434973	
cryptsetup: PBKDF2-whirlpool	593364	
cryptsetup: AES-XTS 256b Encryption	1709.9	
cryptsetup: AES-XTS 256b Decryption	1723.9	
cryptsetup: Serpent-XTS 256b Encryption	592.1	
cryptsetup: Serpent-XTS 256b Decryption	589.4	
cryptsetup: Twofish-XTS 256b Encryption	321.9	
cryptsetup: Twofish-XTS 256b Decryption	328.5	
cryptsetup: AES-XTS 512b Encryption	1650.4	
cryptsetup: AES-XTS 512b Decryption	1611.0	
cryptsetup: Serpent-XTS 512b Encryption	594.6	
cryptsetup: Serpent-XTS 512b Decryption	593.8	
cryptsetup: Twofish-XTS 512b Encryption	326.8	
cryptsetup: Twofish-XTS 512b Decryption	323.8	
Phoronix Test Suite 10.2.2		

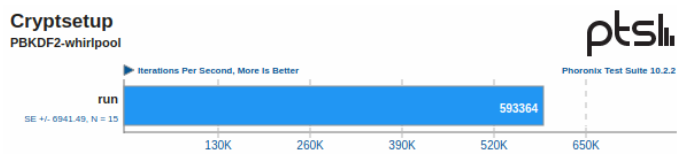
Cryptsetup

PBKDF2-sha512



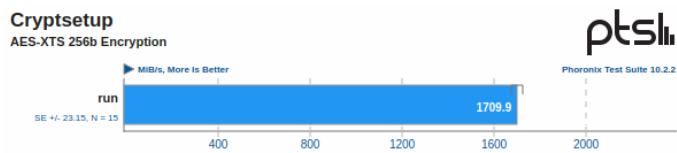
Cryptsetup

PBKDF2-whirlpool



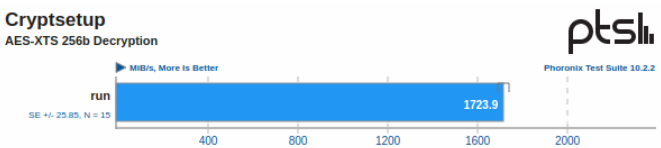
Cryptsetup

AES-XTS 256b Encryption



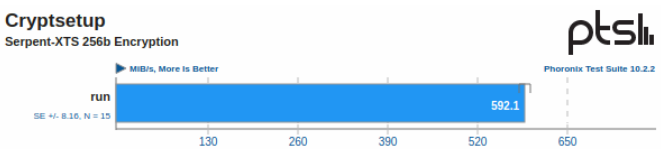
Cryptsetup

AES-XTS 256b Decryption



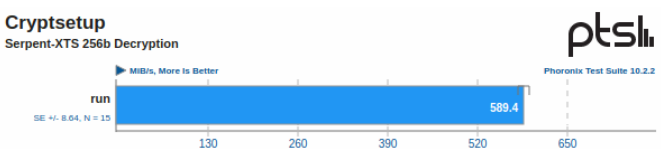
Cryptsetup

Serpent-XTS 256b Encryption



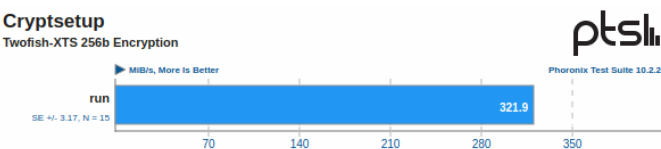
Cryptsetup

Serpent-XTS 256b Decryption



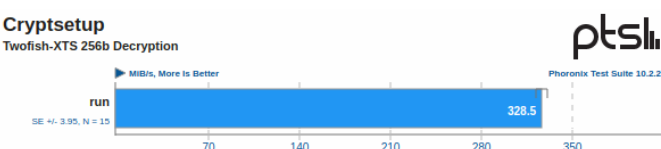
Cryptsetup

Twofish-XTS 256b Encryption



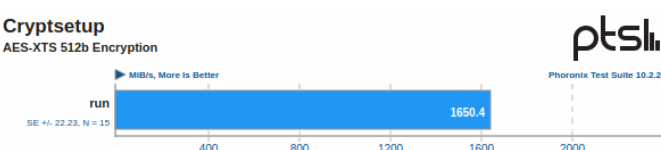
Cryptsetup

Twofish-XTS 256b Decryption



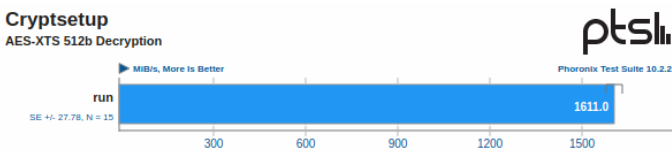
Cryptsetup

AES-XTS 512b Encryption



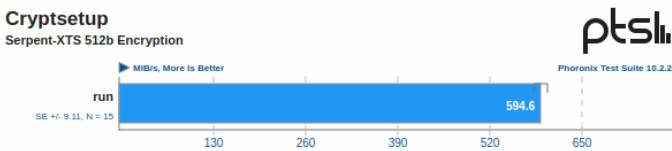
Cryptsetup

AES-XTS 512b Decryption



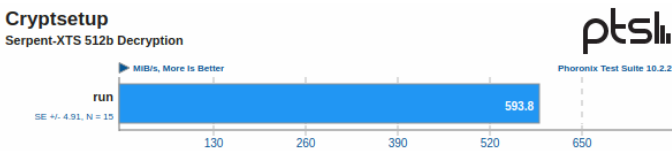
Cryptsetup

Serpent-XTS 512b Encryption



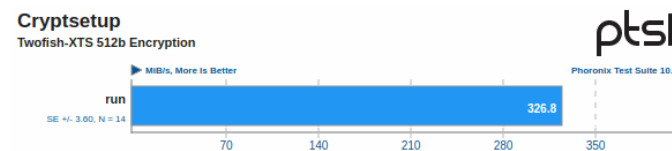
Cryptsetup

Serpent-XTS 512b Decryption



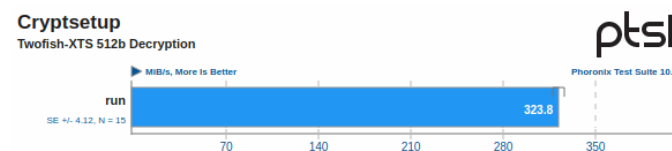
Cryptsetup

Twofish-XTS 512b Encryption



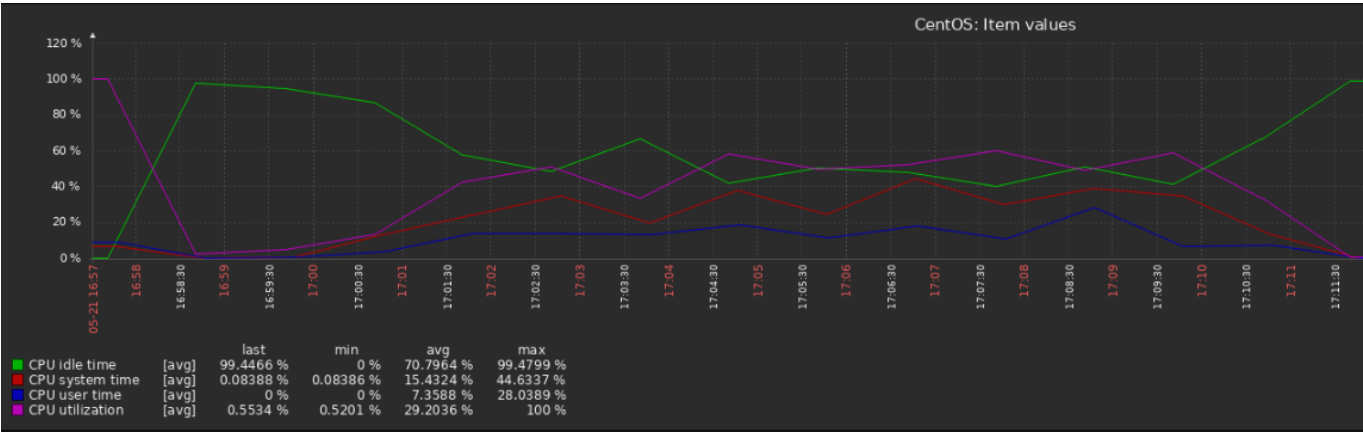
Cryptsetup

Twofish-XTS 512b Decryption



, donde nos muestra la información del sistema sobre el que se ha realizado el Benchmark, una vista general de los resultado y los resultados obtenidos de forma particular para cada una de las encriptaciones disponibles.

Por otro lado, zabbix nos muestra



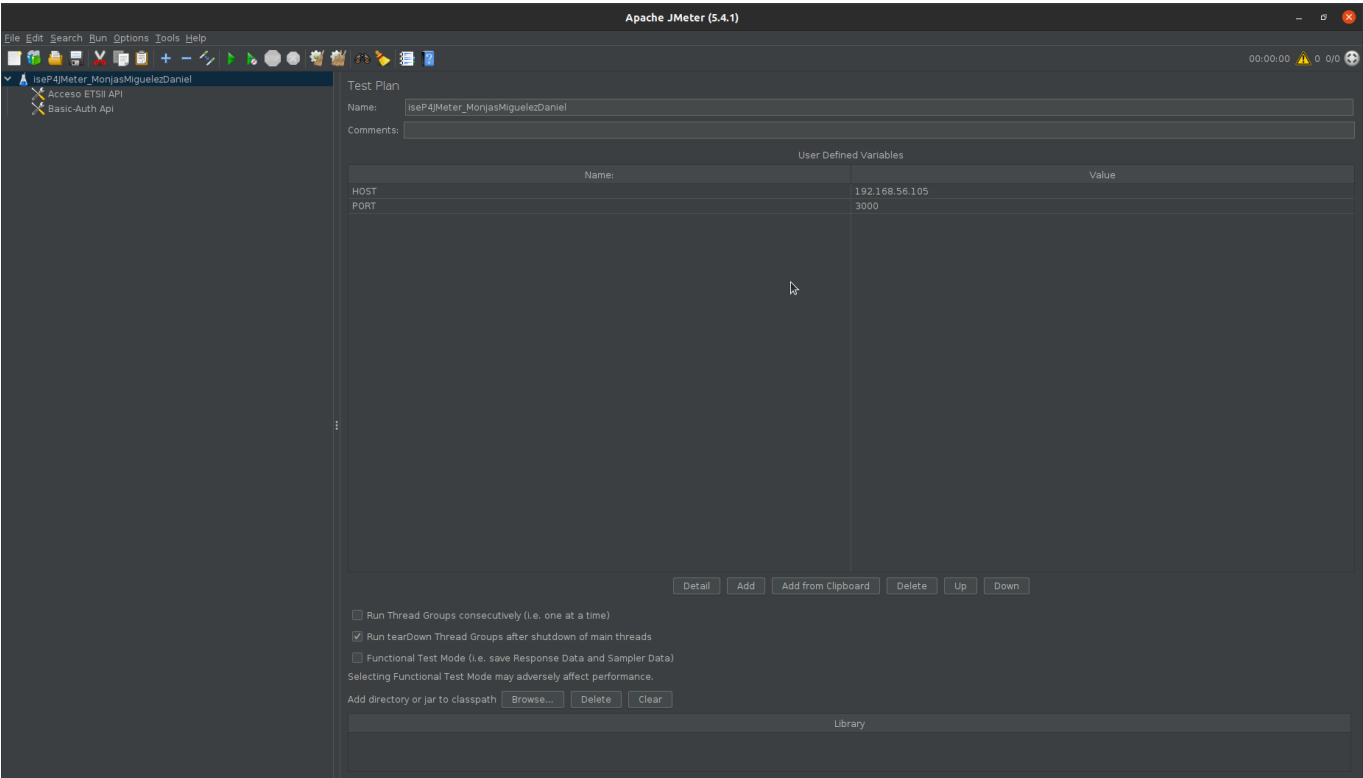
Nos muestra que tras un pico del 100% que tiene debido a la instalación de dependencias que se ha realizado, tras esto parte de una utilización cercana al 0%, que tras empezar el test sube rápidamente hasta una utilización entrono al 60%, muy similar a la obtenida en el test anterior.

Conclusiones

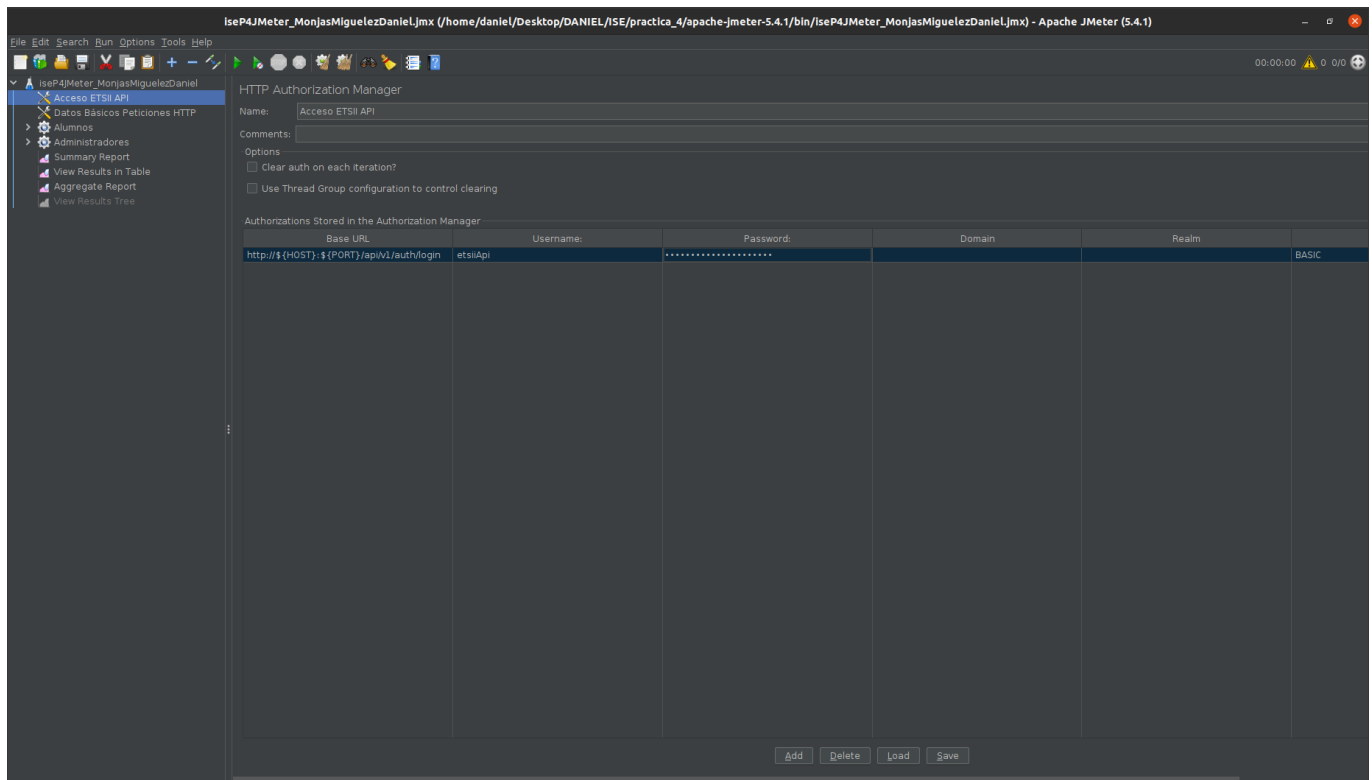
De nuevo CentOS nos proporciona unos resultados un poquito mejor que el obtenido por UbuntuServer (entorno al 2.21%), con una utilización de la CPU menor. De nuevo se considera que esa utilización de CPU mayor en UbuntuServer se debe a la tarea por parte de zabbix de monitorizar tanto a si mismo como a CentOS.

Ejercicio 2

Una vez he instalado la aplicación iseP4JMeter en la máquina virtual e instalado JMeter en mi ordenador abro la aplicación y defino dos variables de usuario



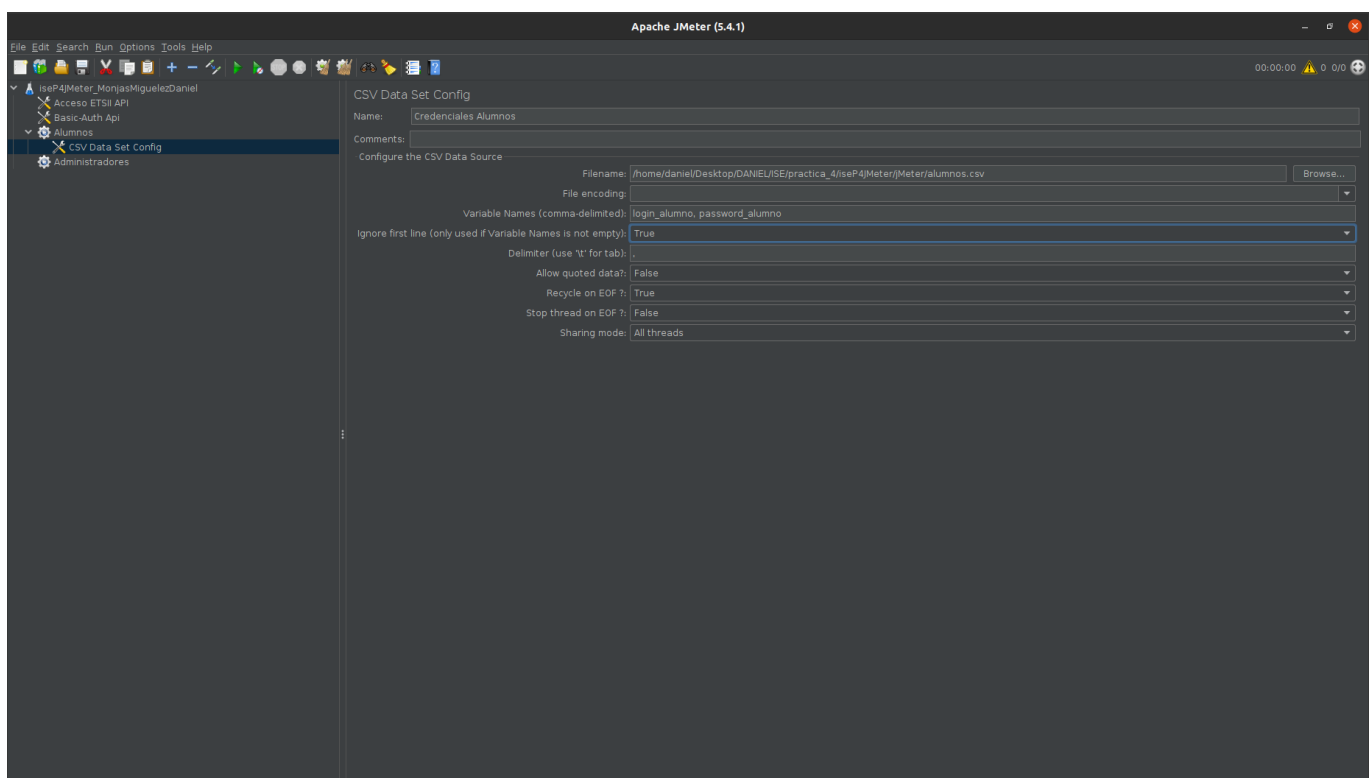
Tras esto añado un HTTP Authorization Manager que lo denomino Acceso ETSII API y le añado los siguiente parámetros



Con esto tenemos guardados en dos variables la IP del servidor y el puerto. También tenemos que en los mensajes HTTP, en la cabecera, se ha especificado **Content-Type: application/x-www-form-urlencoded** y además se ha definido la autenticación básica para acceder a la API de la ETSII.

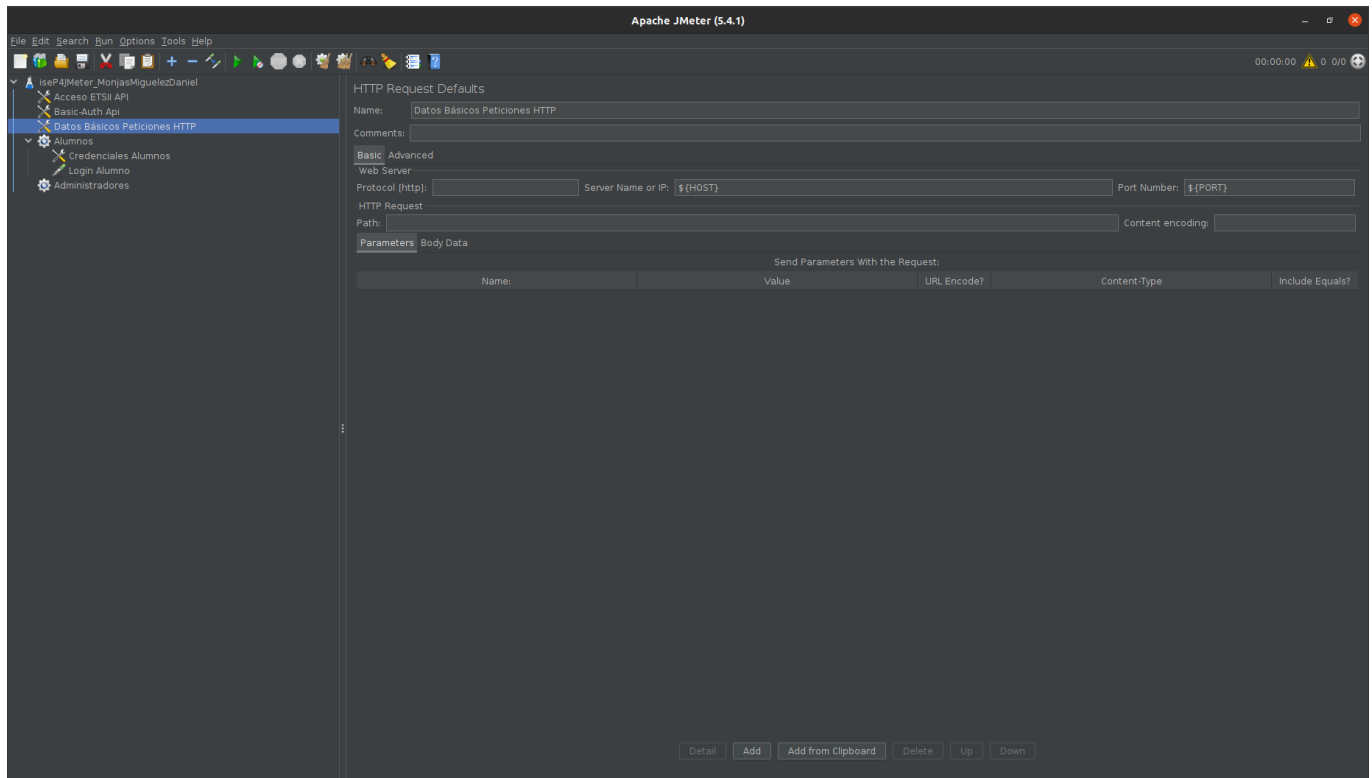
Ahora añadido al test plan dos grupos de hebras, uno que denominaremos Alumnos y otro que denominaremos Administradores, dejando de momento los parámetros que traen por defecto.

Al grupo Alunos, le hacemos **Add->Config Element->CSV Data Set Config** al cual denominaremos **Credenciales Alumnos** y cuyos campos rellenaremos como sigue



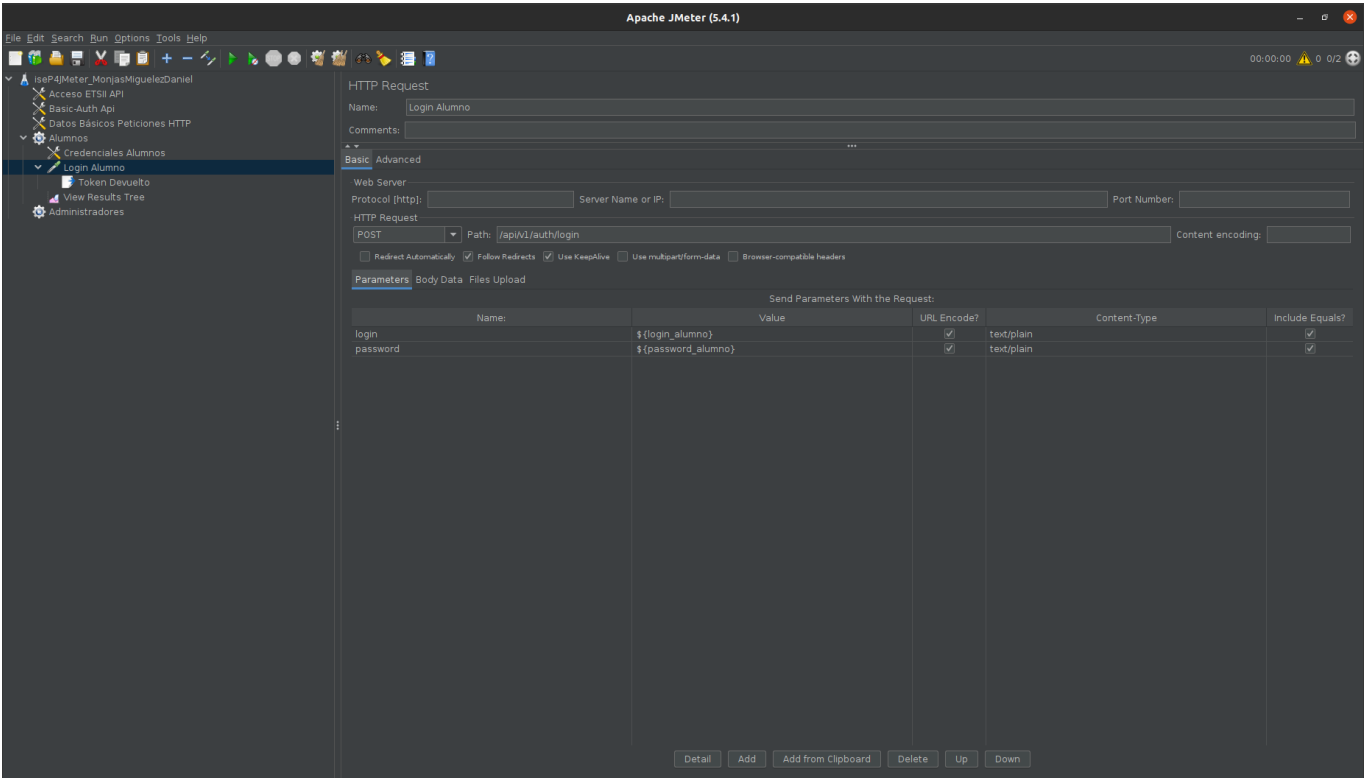
Es especialmente importante cambiar el campo de ignorar primera línea a true, pues la primera línea tiene escrito login y password, lo cual nos dará un error.

Tras esto creamos un **HTTP Request** en el grupo Alumnos haciendo **Add->Sampler->HTTP Request**, y la denominamos Login Alumno. Como más adelante crearemos varias peticiones HTTP todas dirigidas al mismo host y al mismo puerto vamos a crear en el test plan un **HTTP Request Defaults** el cual rellenamos así



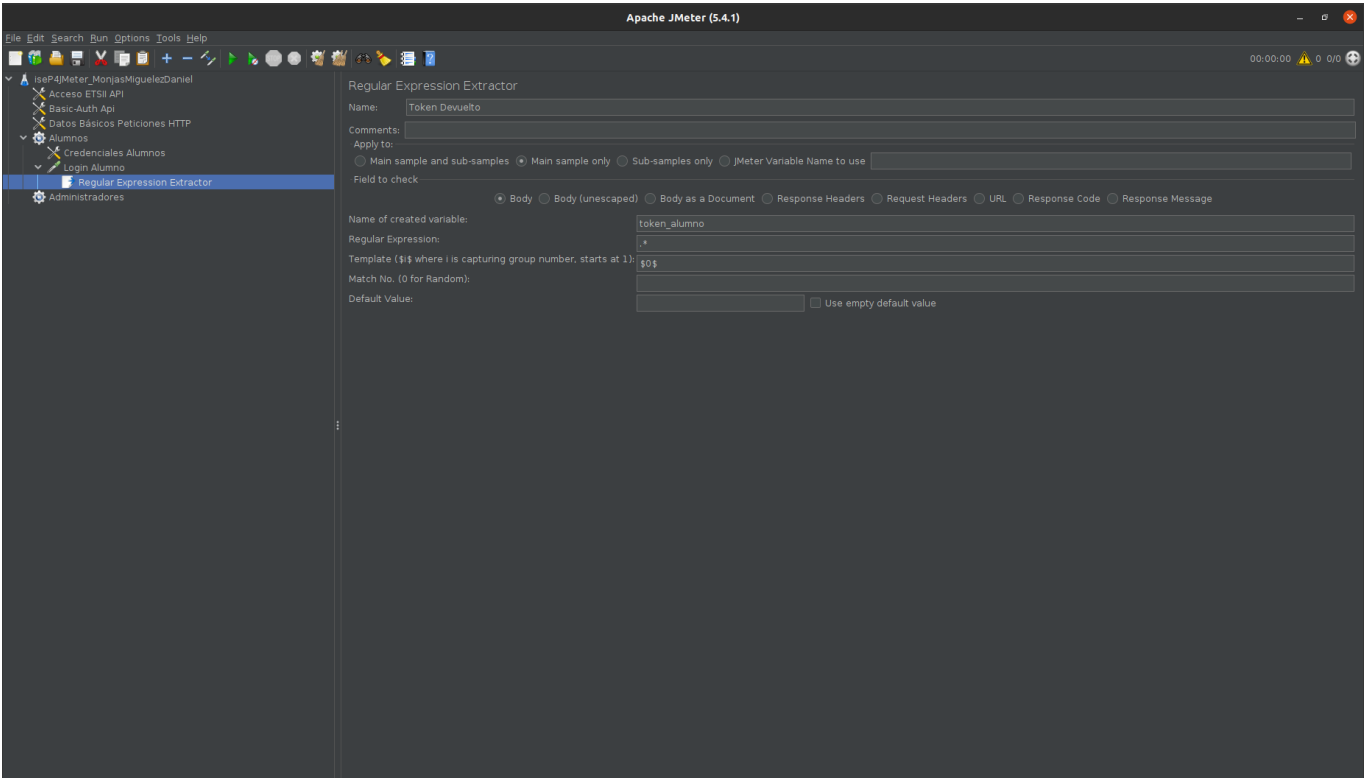
Con esto tendremos que todas las peticiones HTTP irán dirigidas a la dirección contenida en la variables HOST y al puerto contenido en la variable PORT.

Siguiendo con Login Alumno, lo rellenamos como sigue,



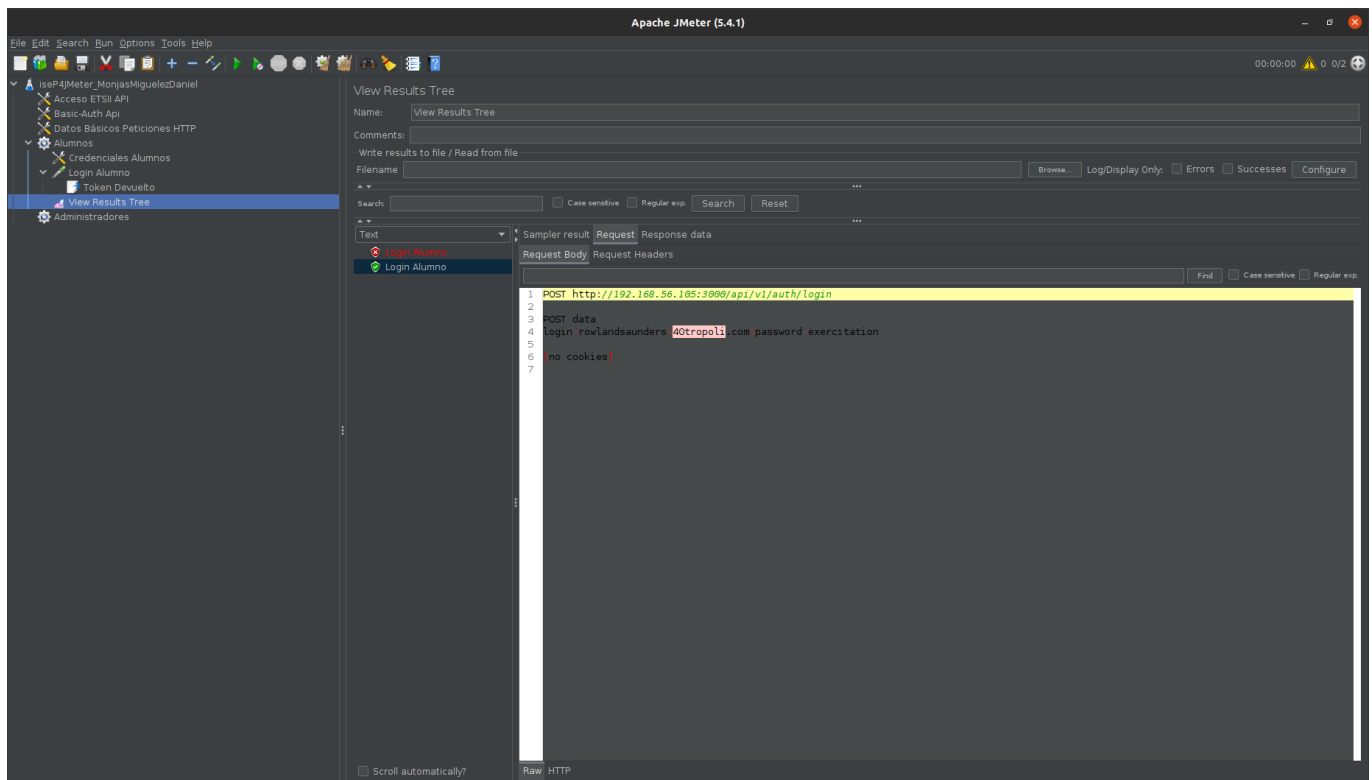
, donde nuestra HTTP Request tendrá dos parámetros, login y password, cuyos valores son obtenidos del archivos de credenciales alumnos.csv. Además vemos que en path vamos a hacer login.

Ahora veamos para extraer el Token que nos devuelve la API al HTTP Request le añadimos un **Regular Expression Extractor**, para ello hacemos sobre Login Alumno **Add->Post Processors->Regular Expression Extractor**, al cual denominaremos Token Devuelto y rellenaremos sus campos como sigue



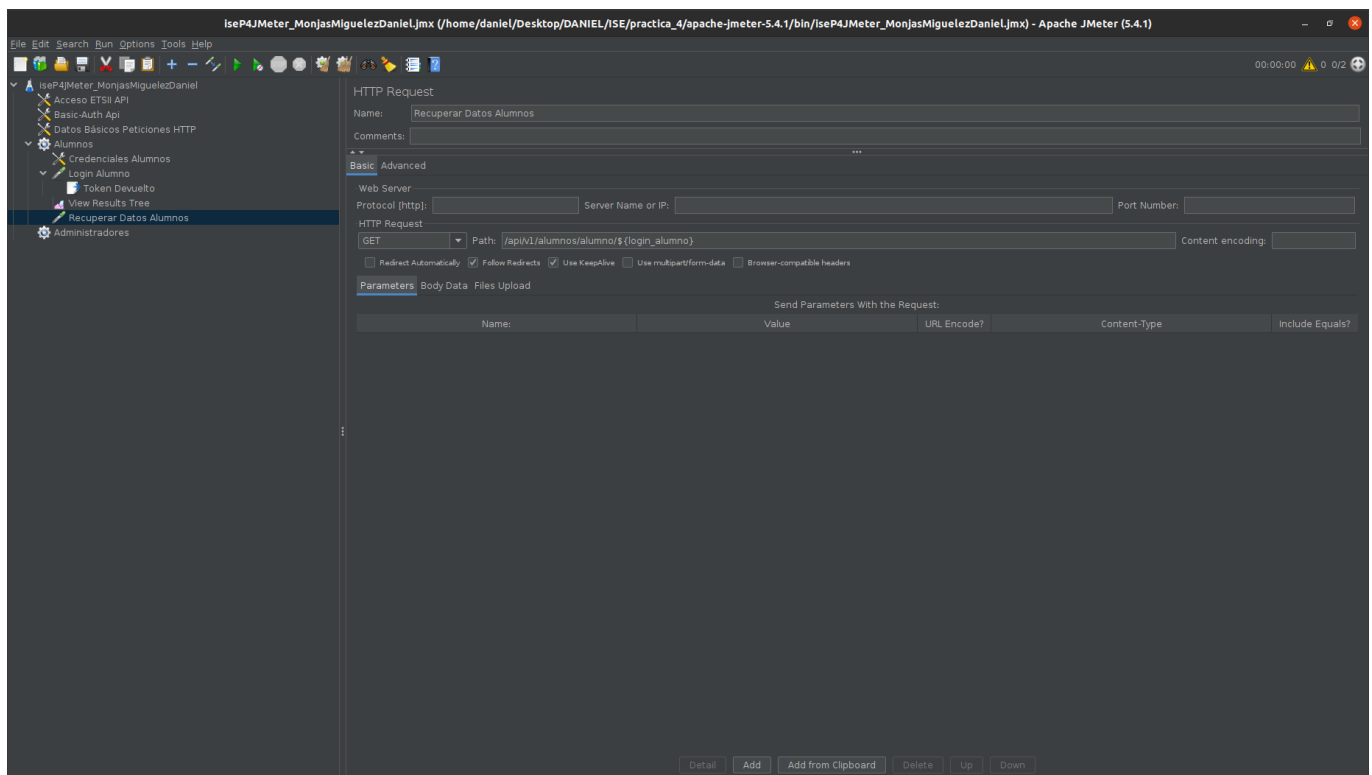
, donde hemos indicado que se almacene en la variable **token_alumno** el valor de lo que devuelva la API, la expresión regular indica que se coja todo y el template que se cojan todos los subgrupos.

Para comprobar si de momento funciona todo bien le añadimos al grupo Alumnos un Listener View Results Tree, para ello hacemos sobre Alumno **Add->Listener->View Results Tree** y le damos a ejecutar el test.

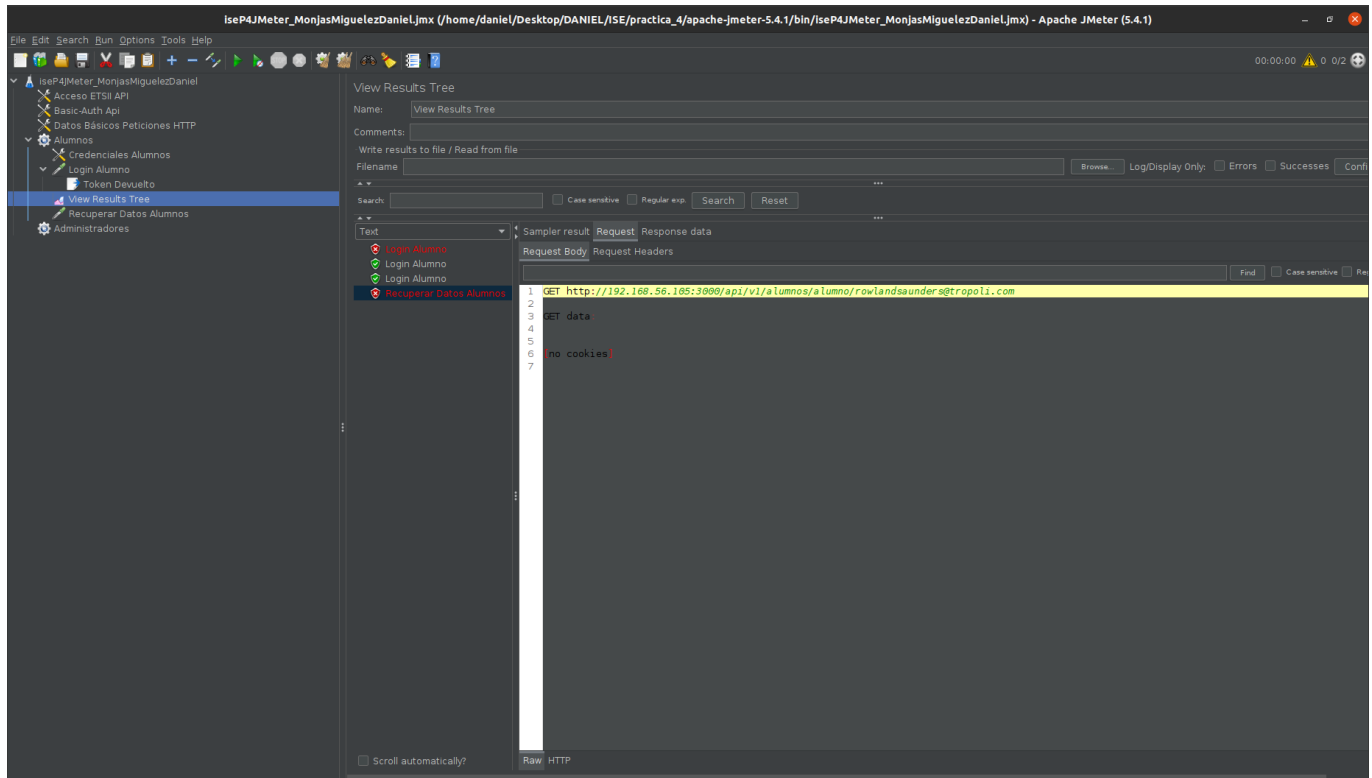


Si nos da error depuramos viendo que ha podido fallar. Por ejemplo, mi primer login ha dado error porque no había indicado el path en login alumno.

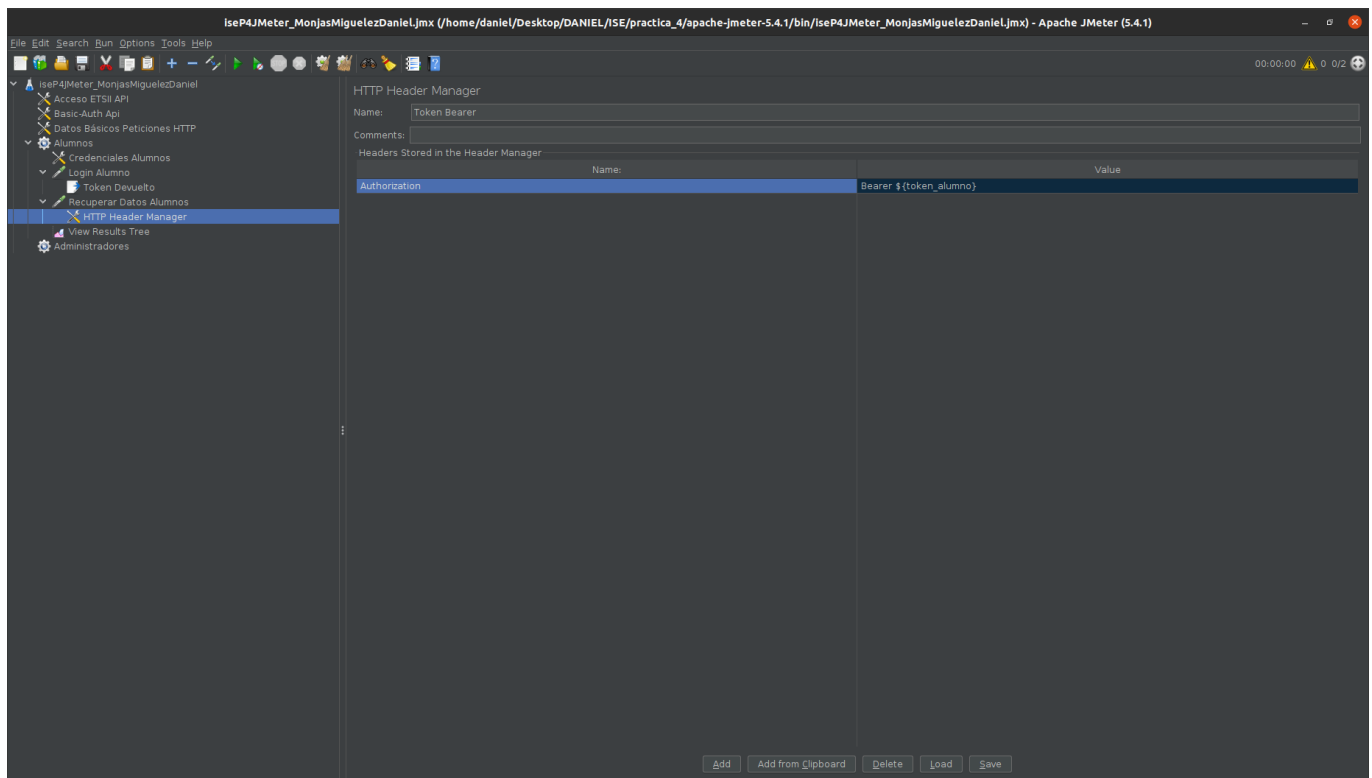
Ahora hacemos otra petición HTTP, esta vez para recoger los datos del alumno concreto y lo denominaremos Recuperar Datos Alumnos, y lo rellenamos como sigue



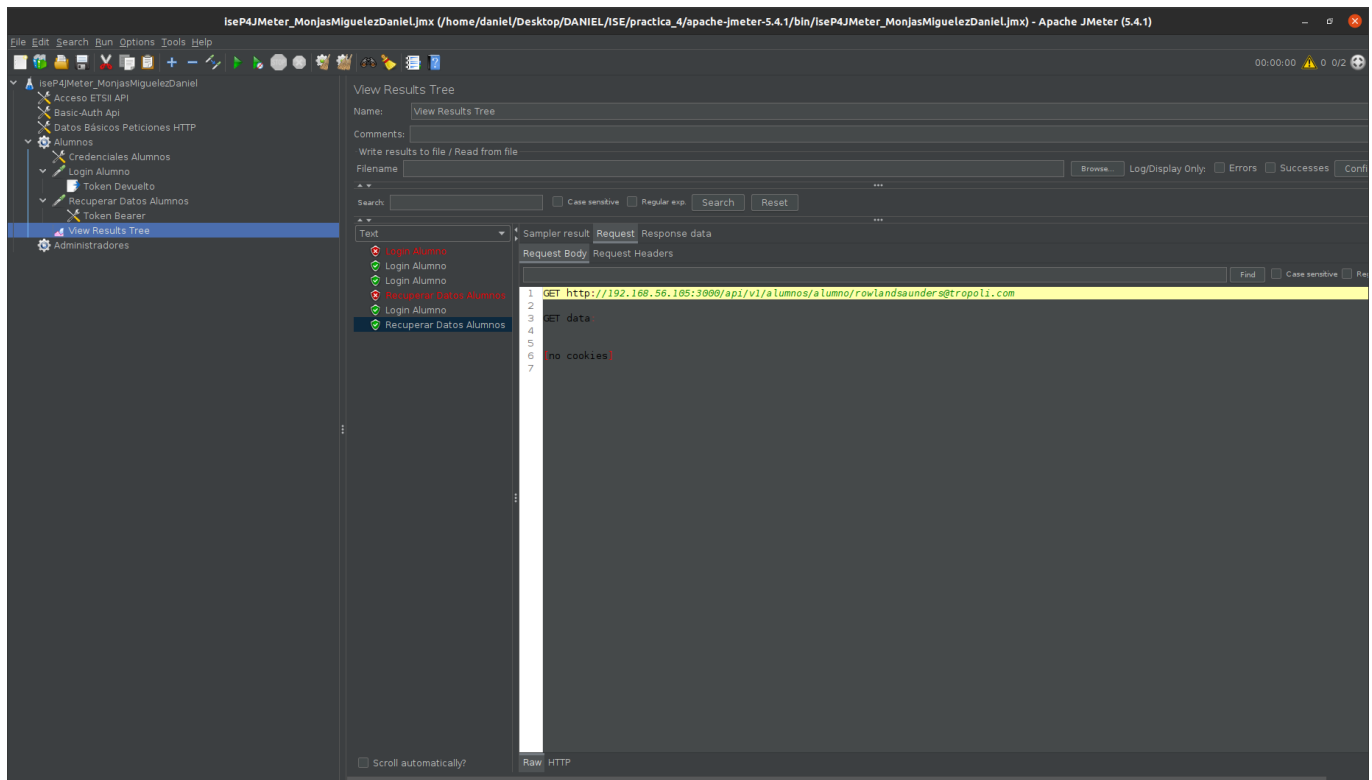
Sin embargo, si ahora probamos el test nos dará error



, esto se debe a que no hemos pasado en la cabecera el token que nos ha devuelto en el login, para añadirlo sobre la petición HTTP Recuperar Datos Alumnos añado un HTTP Header Manager y lo rellenamos como sigue

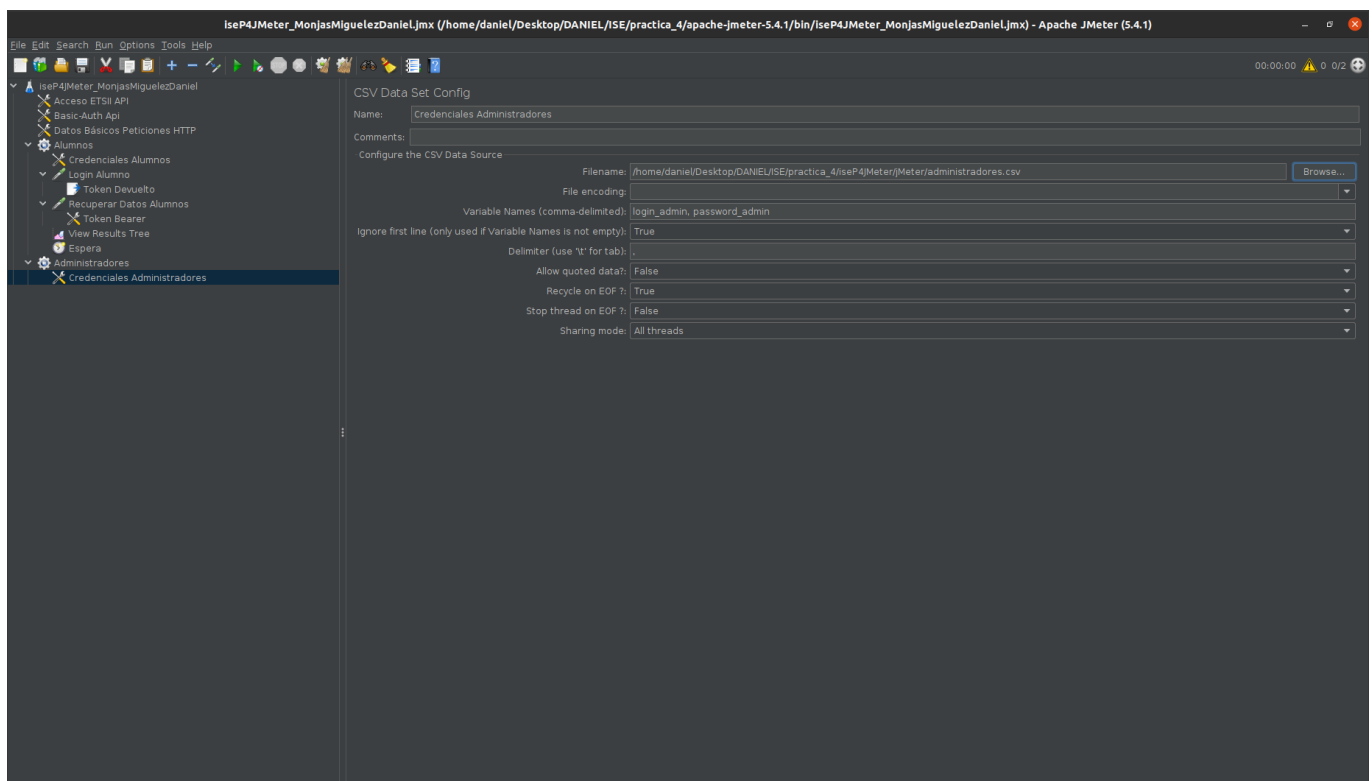


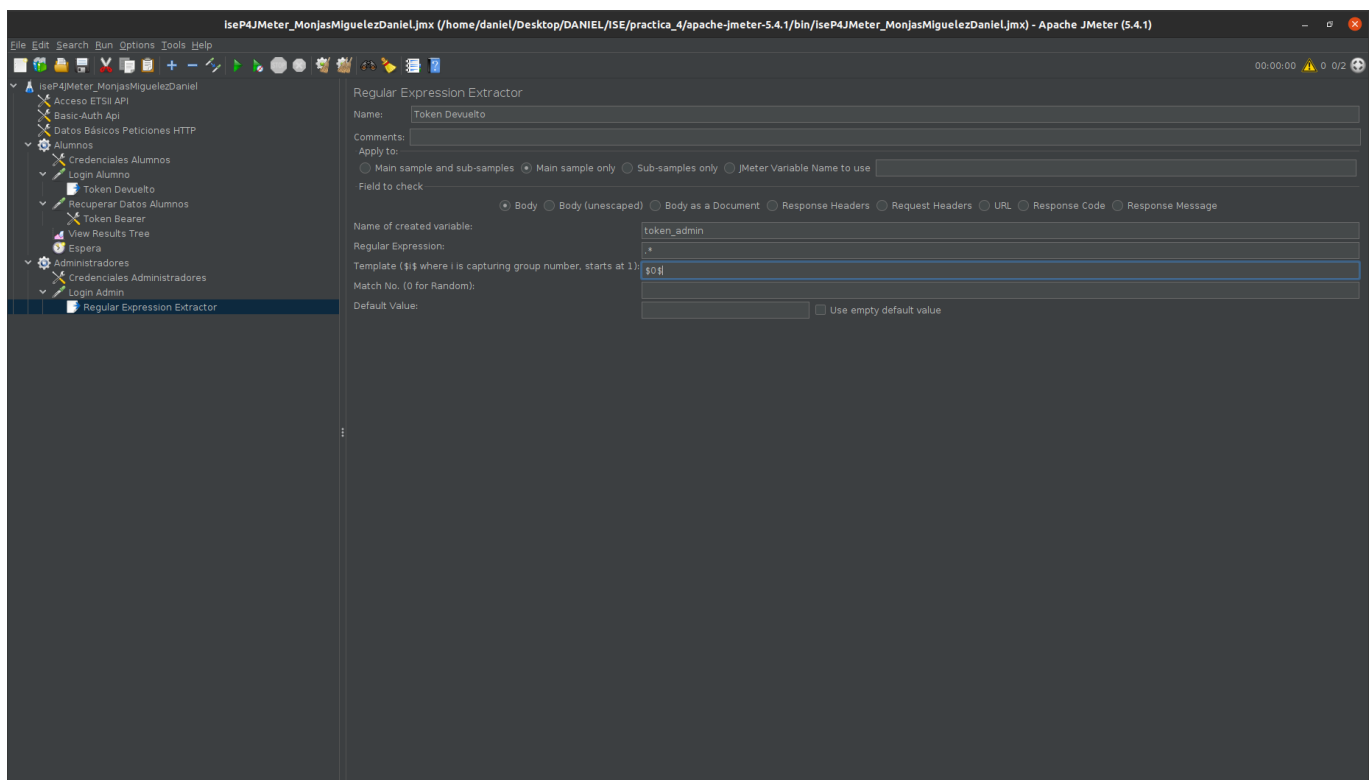
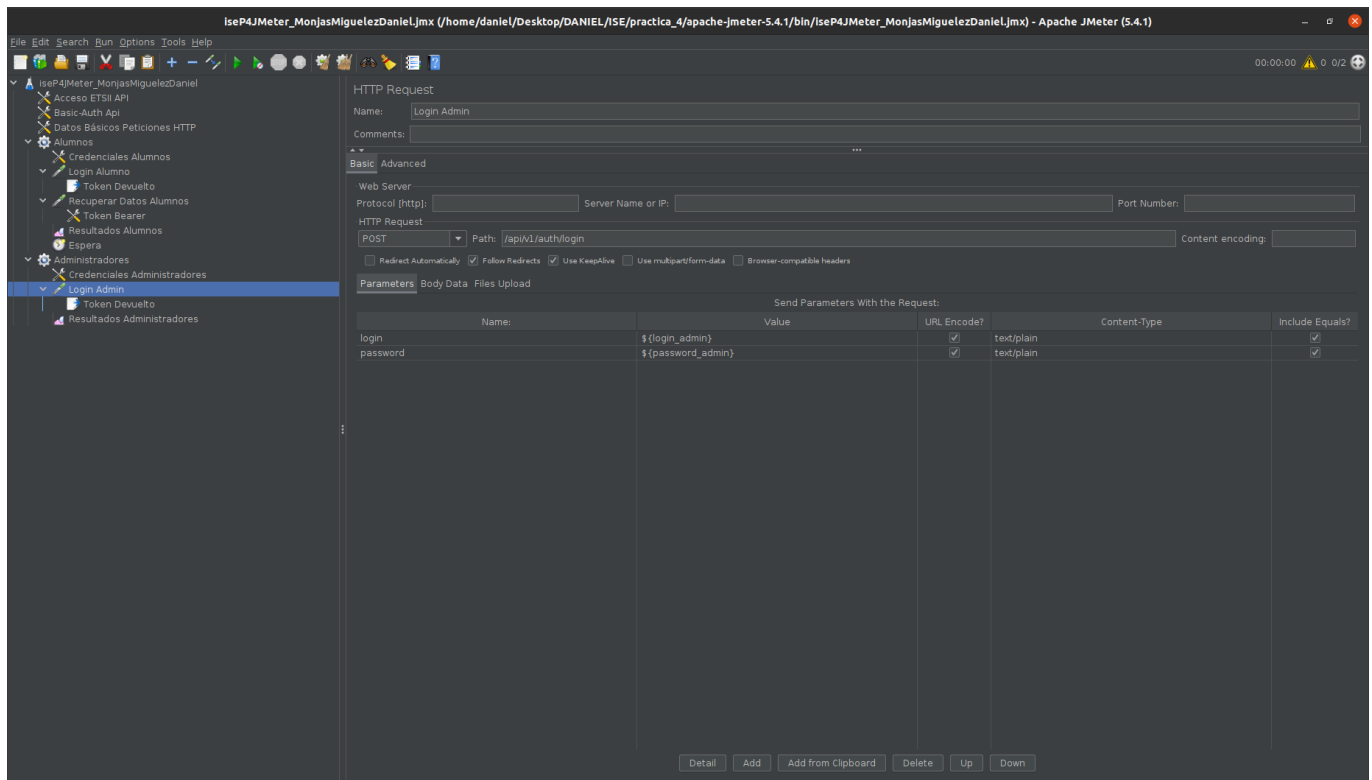
Y ahora si probamos de nuevo el test ya debería funcionar.



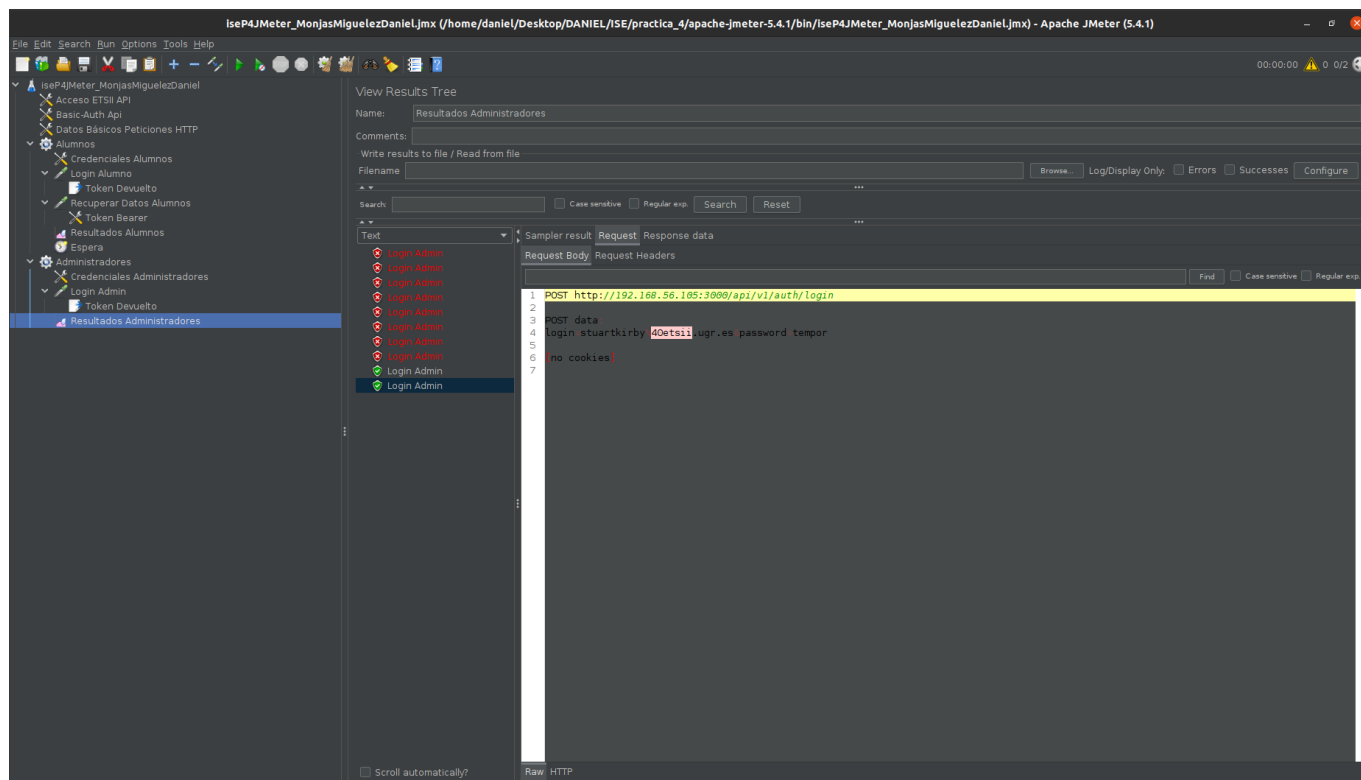
Efectivamente. Por último añadimos la espera que nos indica el enunciado añadiendo a alumnos un timer haciendo **Add->Timer->Gaussian Random Timer** que denominamos espera.

Para el grupo de administradores repetimos el proceso, cambiando el archivo de credenciales.

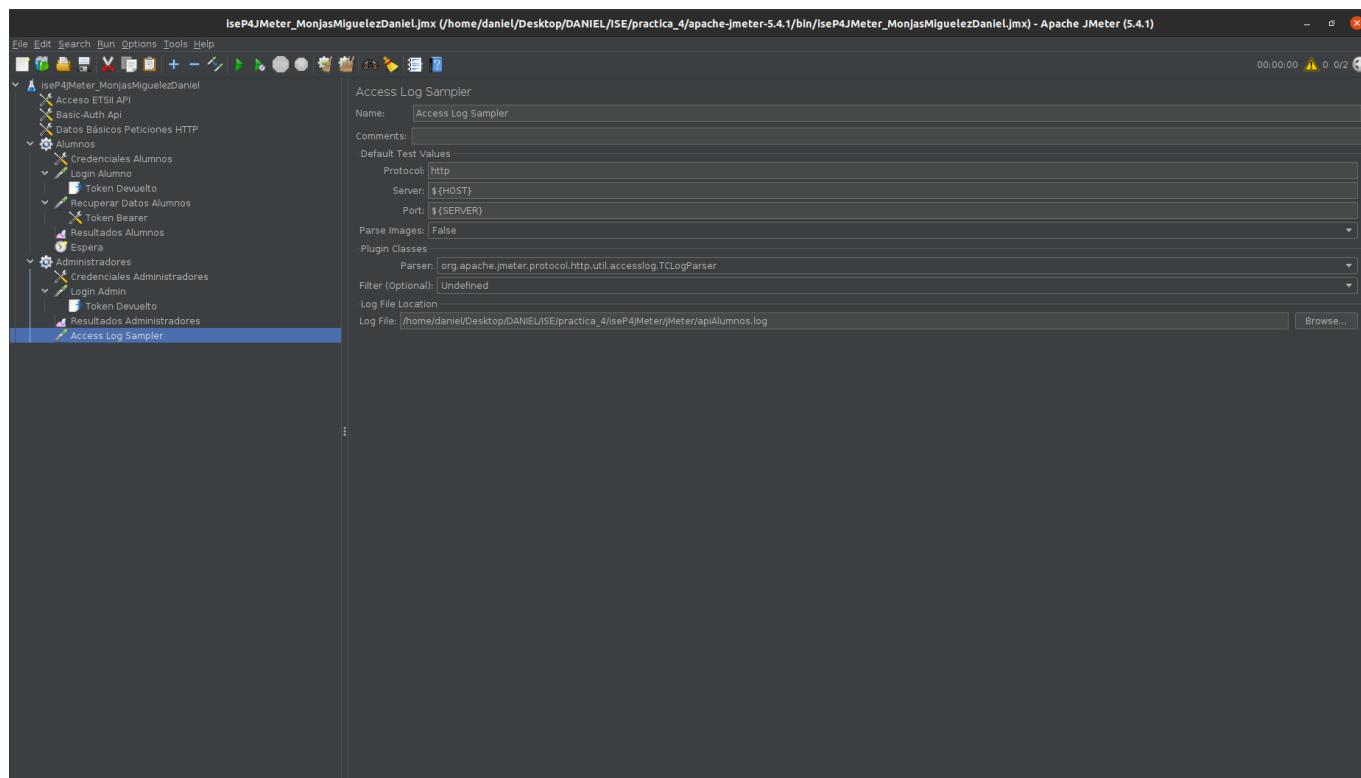




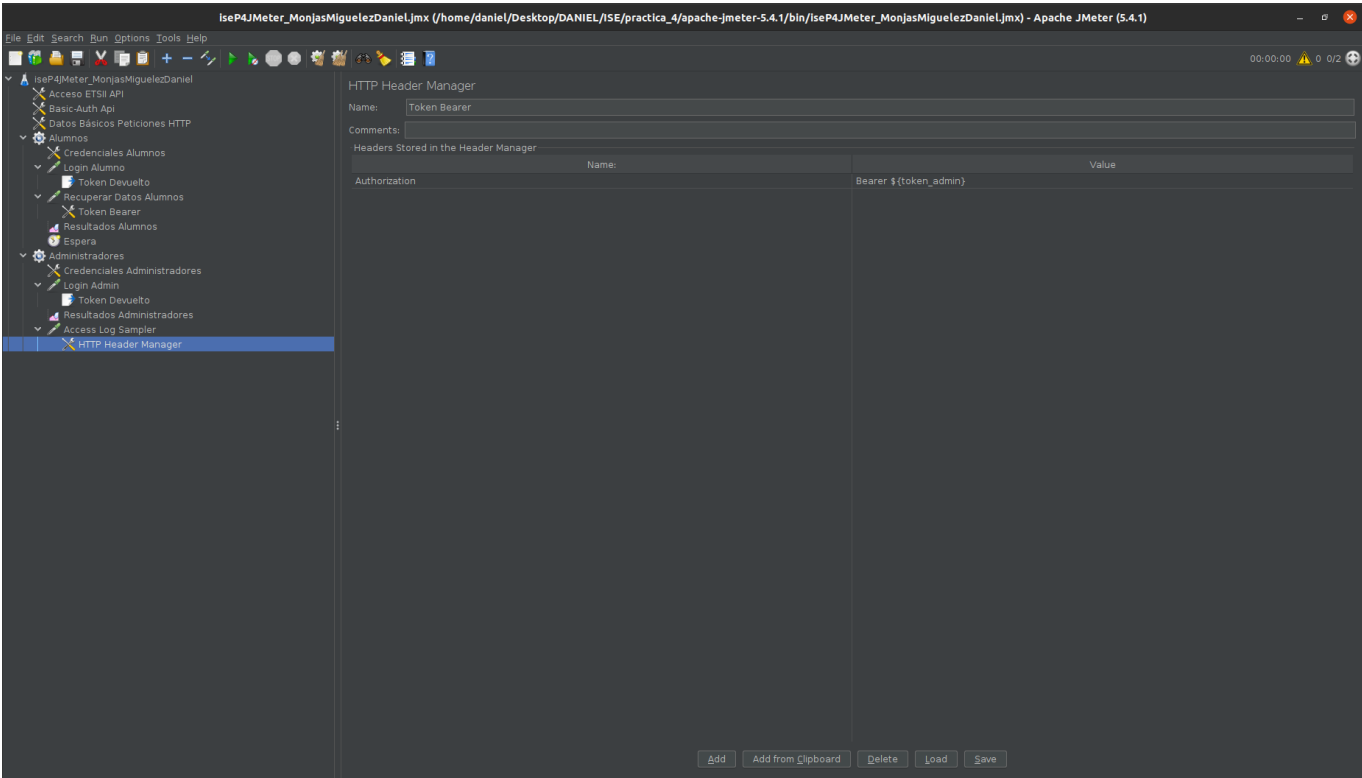
Con esto ya podemos probar si el login es correcto



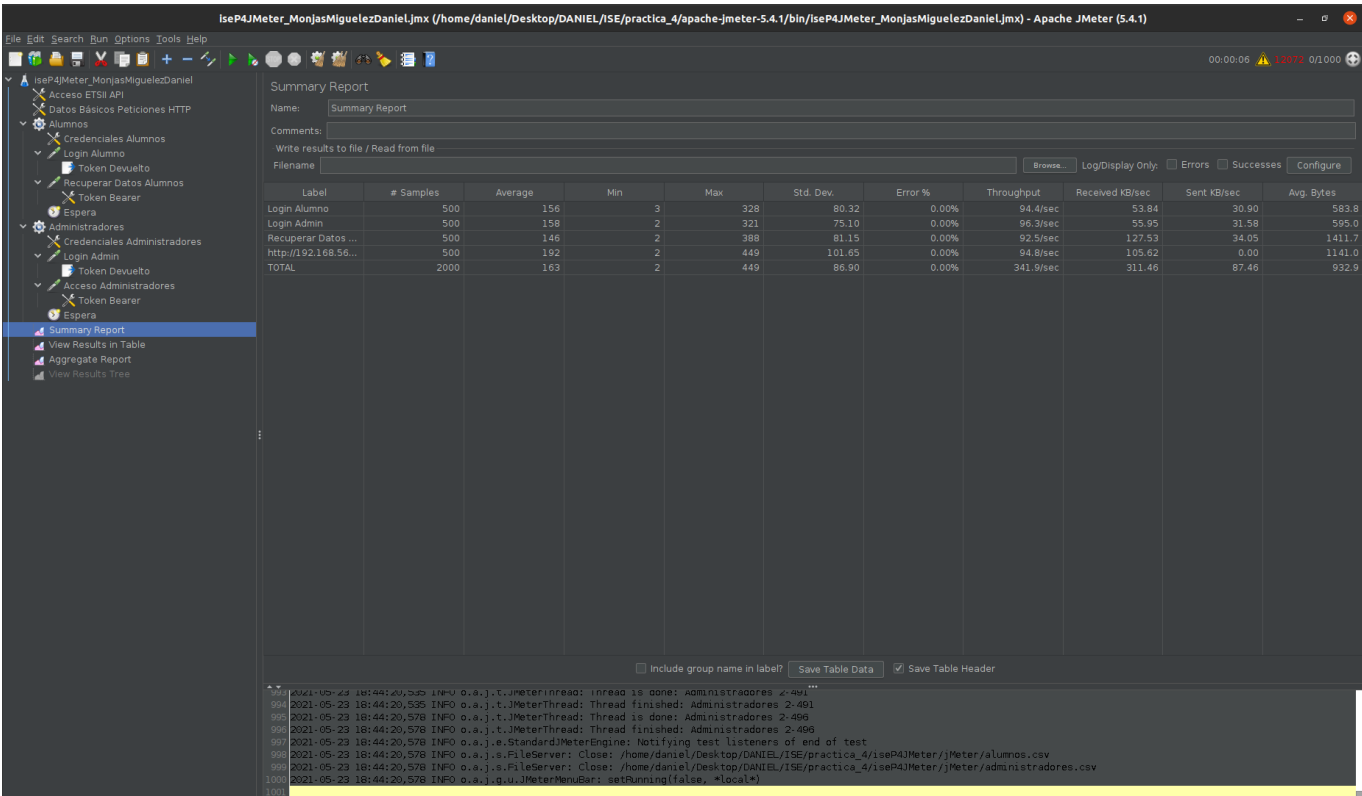
Al principio me había dado error, pues la cabecera del HTTP Request había puesto GET, en lugar de POST. Llegados a este punto es donde varía de alumnos. En lugar de crear un nuevo HTTP Request, creamos un Acceso Log Samples, para ello en administradores hacemos **Add->Sampler->Access Log Sampler** que rellenamos como sigue



, y al igual que hicimos con Recuperar Datos Alumnos tenemos que hacer un HTTP Header Manager al que le pasamos el token que nos devuelve la API al identificarnos como administradores. Y lo rellenamos como sigue



Una vez hecho esto ya tendríamos nuestro test listo para ejecutar. Antes de ejecutar al test le añadimos test listenerse **Add->Listener->Summary Report**, **Add->Listener->View Results in Table** y **Add->Listener->Aggregate Report** (podemos añadir también un View Results Tree, pero aconsejo desactivarlo, pues con muchas peticiones ralentiza bastante), con los cuales comprobaremos que nuestro test es correcto. Además en alumnos y en administradores vamos a actualizar los parámetros Number of Threads y Ramp-up period, a 500 y 5 respectivamente, de forma que nuestro test mandará 100 peticiones por segundo durante cinco segundos de alumnos y análogo con administradores. Con esto podemos ejecutar nuestro test.



View Results in Table

Name:

Comments:

Write results to file / Read from file

Filename Log/Display Only: ☐ Errors ☐ Successes

Sample #	Start Time	Thread Name	Label	Sample Time(ms)	Status	Bytes	Sent Bytes	Latency	Connect Time(ms)
265	18:44:15.768	Administradores 2-69	Login Admin	111	✓	595	334	111	1
266	18:44:15.782	Administradores 2-75	Login Admin	97	✓	595	336	97	1
267	18:44:15.798	Alumnos 1-49	Recuperar Datos AL...	149	✓	1008	376	149	0
268	18:44:15.772	Alumnos 1-60	Recuperar Datos AL...	134	✓	2283	374	134	0
269	18:44:15.831	Alumnos 1-68	Recuperar Datos AL...	77	✓	1985	377	77	0
270	18:44:15.792	Administradores 2-51	http://192.168.56.1...	117	✓	1141	0	117	0
271	18:44:15.817	Alumnos 1-112	Login Alumno	94	✓	584	336	94	1
272	18:44:15.834	Alumnos 1-92	Login Alumno	77	✓	583	342	77	0
273	18:44:15.806	Alumnos 1-110	Login Alumno	116	✓	584	337	116	1
274	18:44:15.824	Alumnos 1-121	Login Alumno	98	✓	584	336	98	1
275	18:44:15.806	Alumnos 1-111	Login Alumno	117	✓	584	339	117	0
276	18:44:15.823	Alumnos 1-102	Login Alumno	100	✓	583	334	100	0
277	18:44:15.841	Alumnos 1-97	Login Alumno	82	✓	583	335	82	0
278	18:44:15.816	Alumnos 1-90	Login Alumno	109	✓	583	333	109	0
279	18:44:15.818	Alumnos 1-105	Login Alumno	107	✓	584	334	107	0
280	18:44:15.838	Alumnos 1-104	Login Alumno	87	✓	584	343	87	1
281	18:44:15.804	Administradores 2-45	Login Admin	122	✓	595	337	122	1
282	18:44:15.832	Administradores 2-78	Login Admin	95	✓	595	336	95	1
283	18:44:15.812	Alumnos 1-79	Recuperar Datos AL...	135	✓	2054	374	135	0
284	18:44:15.792	Administradores 2-38	http://192.168.56.1...	156	✓	1141	0	156	0
285	18:44:15.823	Administradores 2-42	http://192.168.56.1...	126	✓	1141	0	126	0
286	18:44:15.820	Administradores 2-39	http://192.168.56.1...	131	✓	1141	0	131	0
287	18:44:15.819	Administradores 2-41	http://192.168.56.1...	132	✓	1141	0	132	0
288	18:44:15.831	Administradores 2-30	http://192.168.56.1...	121	✓	1141	0	121	0
289	18:44:15.854	Alumnos 1-74	Recuperar Datos AL...	98	✓	1520	375	98	0
290	18:44:15.824	Administradores 2-25	http://192.168.56.1...	129	✓	1141	0	129	0
291	18:44:15.850	Alumnos 1-130	Login Alumno	113	✓	584	339	113	1
292	18:44:15.854	Administradores 2-83	Login Admin	110	✓	595	336	110	0
293	18:44:15.865	Alumnos 1-71	Login Alumno	99	✓	583	340	99	0
294	18:44:15.845	Administradores 2-91	Login Admin	120	✓	595	333	120	1

☐ Scroll automatically? ☐ Child samples? No of Samples: 2000 Latest Sample: 3 Average: 143 Deviation: 84

```

2021-05-23 18:44:20,526 INFO o.a.j.t.JMeterThread: Thread is done: Administradores 2-491
994 2021-05-23 18:44:20,535 INFO o.a.j.t.JMeterThread: Thread finished: Administradores 2-491
995 2021-05-23 18:44:20,578 INFO o.a.j.t.JMeterThread: Thread is done: Administradores 2-496
996 2021-05-23 18:44:20,578 INFO o.a.j.t.JMeterThread: Thread finished: Administradores 2-496
997 2021-05-23 18:44:20,578 INFO o.a.j.e.StandardJMeterEngine: Notifying test listeners of end of test
998 2021-05-23 18:44:20,578 INFO o.a.j.s.FileServer: Close: /home/daniel/Desktop/DANIEL/ISE/practica_4/iseP4JMeter/jMeter/alumnos.csv
999 2021-05-23 18:44:20,578 INFO o.a.j.s.FileServer: Close: /home/daniel/Desktop/DANIEL/ISE/practica_4/iseP4JMeter/jMeter/administradores.csv
1000 2021-05-23 18:44:20,578 INFO o.a.j.g.u.JMeterMenuBar: setRunning(false, *local*)

```

Aggregate Report

Name:

Comments:

Write results to file / Read from file

Filename Log/Display Only: ☐ Errors ☐ Successes

Label	# Samples	Average	Median	90% Line	95% Line	99% Line	Min	Maximum	Error %	Throughput	Received KB/sec	Sent KB/sec
Login Alumno	500	156	142	271	290	314	3	328	0.00%	94.4/sec	53.84	30.90
Login Admin	500	158	146	262	285	314	2	321	0.00%	96.3/sec	55.95	31.58
Recuperar Dat...	500	146	137	257	299	356	2	388	0.00%	92.5/sec	127.53	34.05
http://192.168...	500	192	182	328	365	429	2	449	0.00%	94.8/sec	105.62	0.00
TOTAL	2000	163	153	286	311	375	2	449	0.00%	341.9/sec	311.46	87.46

☐ Include group name in label? ☒ Save Table Header

```

2021-05-23 18:44:20,526 INFO o.a.j.t.JMeterThread: Thread is done: Administradores 2-491
994 2021-05-23 18:44:20,535 INFO o.a.j.t.JMeterThread: Thread finished: Administradores 2-491
995 2021-05-23 18:44:20,578 INFO o.a.j.t.JMeterThread: Thread is done: Administradores 2-496
996 2021-05-23 18:44:20,578 INFO o.a.j.t.JMeterThread: Thread finished: Administradores 2-496
997 2021-05-23 18:44:20,578 INFO o.a.j.e.StandardJMeterEngine: Notifying test listeners of end of test
998 2021-05-23 18:44:20,578 INFO o.a.j.s.FileServer: Close: /home/daniel/Desktop/DANIEL/ISE/practica_4/iseP4JMeter/jMeter/alumnos.csv
999 2021-05-23 18:44:20,578 INFO o.a.j.s.FileServer: Close: /home/daniel/Desktop/DANIEL/ISE/practica_4/iseP4JMeter/jMeter/administradores.csv
1000 2021-05-23 18:44:20,578 INFO o.a.j.g.u.JMeterMenuBar: setRunning(false, *local*)

```

En el Summary Report se puede apreciar que se han realizado 500 peticiones de Login Alumno, Login Administradores, Recuperar Datos y luego un http://dir_ip que se corresponde con Acceso Administradores. También se aprecia que el porcentaje de error en las cuatro filas es 0 así como el total.

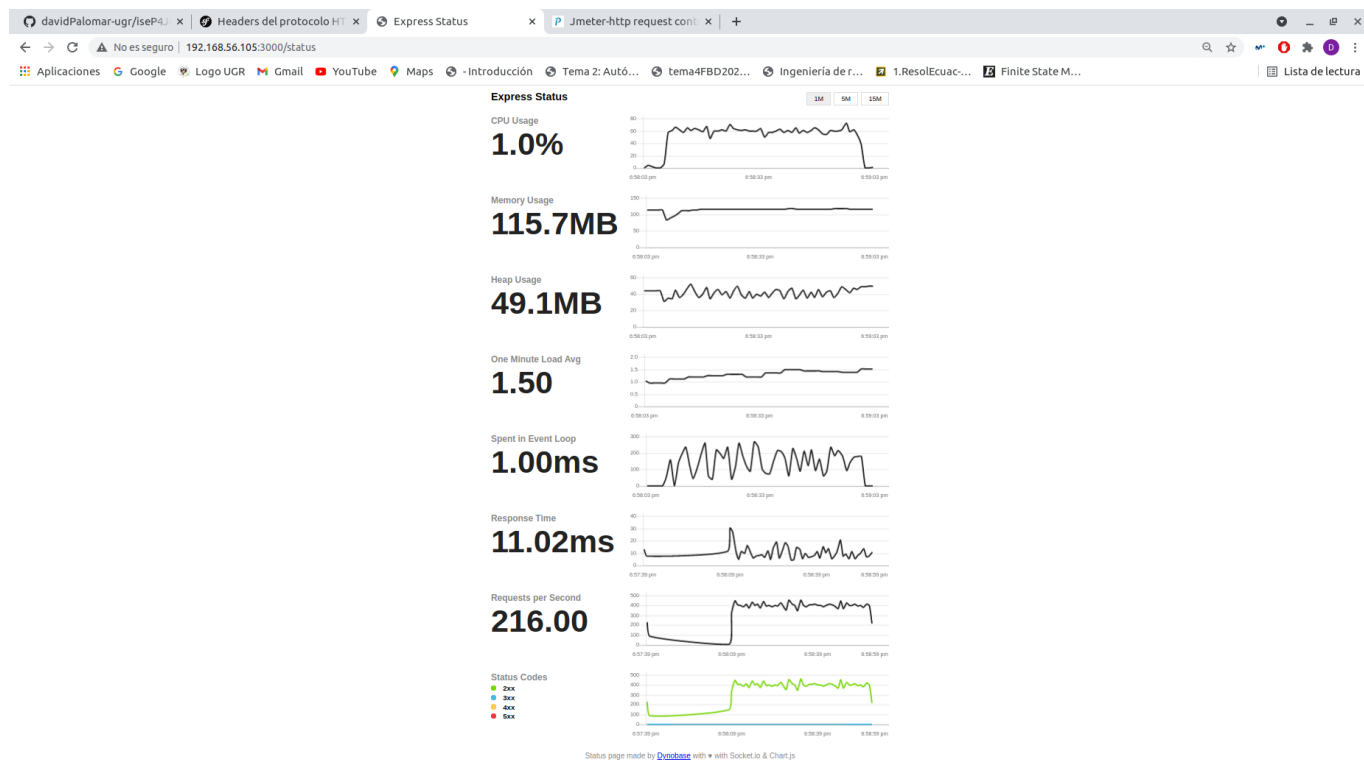
Si vamos a View Results in Table vemos que de nuevo nos muestra status en verde en todas las peticiones aunque solo se vean unas pocas en la imagen

en cuanto al Agregate Report podemos apreciar lo mismo, que el error obtenido es 0 y otras información como los KB/sec recibidos y enviados, etc.

Recaltar que se ha eliminado el HTTP Header Manager Basic-Auth Api que se ven en varias imágenes, pues el único parámetro que incluía era `Content-Type: application/x-www-form-urlencoded`, que es el por defecto en JMeter, y porque este generaba error en Acceso Administradores.

Por último veamos como estresa nuestra aplicación (aumentaré un poco el número de peticiones y segundos), para ello vamos en el navegador a la siguiente dirección

<http://192.168.56.105:3000/status>



Para esta prueba he establecido un test en el que acceden 100 usuarios de cada tipo (admin y alumnos) cada segundo durante 50 segundos. Vemos que la utilización de la CPU se encuentra entorno al 75%, luego nuestro servidor podría servir unas pocas más peticiones por segundo. También se ve que el status code por lo general es del tipo 2xx, es decir, de ejecución correcta.

Con esto se concluye el ejercicio 2.

Ejercicio Opcional. Ejecutar un test de Phoronix usando Docker en nuestro ordenador anfitrión

En primer lugar tenemos que instalar Docker en nuestra máquina, para ello se recomienda seguir los pasos de <https://help.wnpower.com/hc/es/articles/360048910771-Cómo-instalar-Docker-en-tu-servidor-con-Ubuntu>

Tras esto procedemos a descargar una imagen de Docker llamada phoronix/pts con el comando

```
docker pull phoronix/pts
```

```
daniel@daniel-XPS-15-9570:~/Desktop/DANIEL/ISE/practica_4/phoronix-test-suite$ sudo docker pull phoronix/pts
Using default tag: latest
latest: Pulling from phoronix/pts
83ee3a23efb7: Pull complete
db98fc6f11f0: Pull complete
f611acd52c6c: Pull complete
110ccc1b06f4: Pull complete
5ecb394aea70: Pull complete
c0477717ed6e: Pull complete
3a2e3b56e24b: Pull complete
Digest: sha256:9723cd8eb5516aa3e0c1e65c03bdf0fe1969016aba8368d46c48bbf3dfae0d0f
Status: Downloaded newer image for phoronix/pts:latest
docker.io/phoronix/pts:latest
```

Una vez instalado lanzamos el docker con el comando

```
sudo docker run -it phoronix/pts
```

Con esto llegaremos a una pantalla similar a esta

```
daniel@daniel-XPS-15-9570: ~
Updated Test: system/selenium v1.0.22 Selenium
Updated OpenBenchmarking.org Repository Index
git: 8 Distinct Tests, 9 Test Versions
Phoronix Test Suite v10.2.1
Interactive Shell

Generating Shell Cache...
Refreshing OpenBenchmarking.org Repository Cache...

PROCESOR: Intel Core i5-8300H @ 4.00GHz
Core Count: 4
Thread Count: 8
Extensions: SSE 4.2 + AVX2 + AVX + RDRAND + FSGSBASE
Cache Size: 8 MB
Microcode: 0x0e
Core Family: Kaby/Coffee/Whiskey Lake
Scaling Driver: intel_pstate powersave

GRAPHICS: i915drmfb
Frequency: 1000MHz
Display Driver: i915drmfb
Monitor: ASUS VS229
Screen: 1920x1080

MOTHERBOARD: Dell GYMHT6
BIOS Version: 1.10.2
Audio: Realtek ALC3266

MEMORY: 20GB

DISK: SK hynix PC401 NVMe 256GB
File-System: overlayfs
Disk Scheduler: NONE

OPERATING SYSTEM: Ubuntu 20.04.1 LTS
Kernel: 5.8.0-53-generic (x86_64)
System Layer: Docker
Security:
+ itlb_multihit: KVM: Mitigation of VMX disabled
+ l1tf: Mitigation of PTE Inversion; VMX: conditional cache flushes SMT vulnerable
+ mds: Mitigation of Clear buffers; SMT vulnerable
+ meltdown: Mitigation of PTI
+ spec_store_bypass: Mitigation of SSB disabled via prctl and seccomp
+ spectre_v1: Mitigation of usercopy/swapgs barriers and __user pointer sanitization
+ spectre_v2: Mitigation of Full generic retpoline IBPB: conditional IBRS_FW STIBP: conditional RSB filling
+ srbds: Mitigation of Microcode
+ tsx_async_abort: Not affected

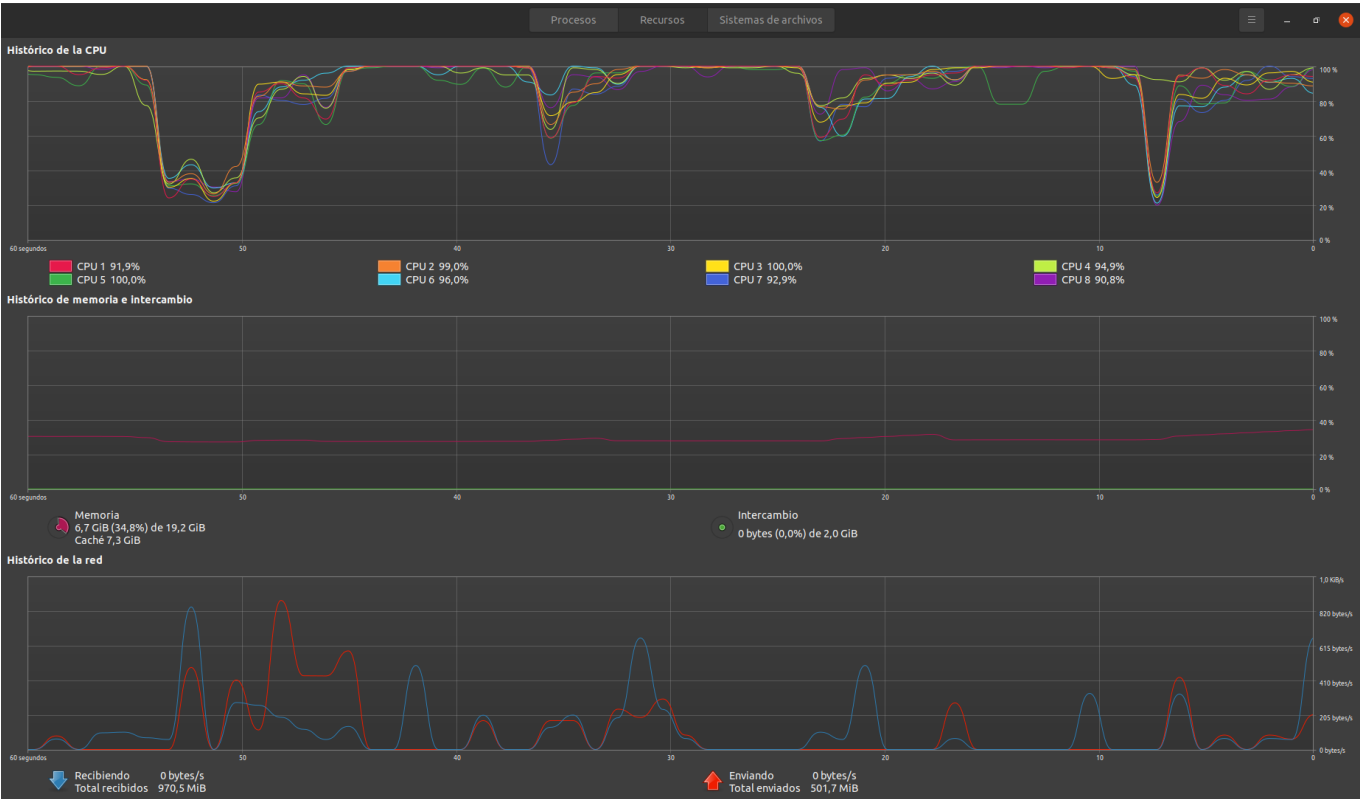
CPU Temperature: 48.00 C CPU Usage (Summary): 20.55 % Memory Usage: 2831 MB System Temperature: 25.0 C System Uptime 24 h

Phoronix Test Suite command to run or help for all possible options, commands for a quick overview of options, interactive for a guided experience, system-info to view system hardware/software information, exit to exit. For new users, benchmark is the simplest and most important sub-command. Tab auto-completion support available.
```

Y dese aquí ejecutamos el comando

```
benchmark compress-7zip
```

, que se corresponde con un test de procesador ejecutado en el ejercicio 1. Abrimos el monitor del sistema para comprobar la utilización del procesador durante la ejecución del test.





```
daniel@daniel-XPS-15-9570: ~  
+ spectre_v1: Mitigation of usercopy/swapgs barriers and __user pointer sanitization  
+ spectre_v2: Mitigation of Full generic retpoline IBPB: conditional IBRS_FW STIBP: conditional RSB filling  
+ srbds: Mitigation of Microcode  
+ tsx_async_abort: Not affected  
  
Would you like to save these test results (Y/n): Y  
Enter a name for the result file: compress_pc  
Enter a unique name to describe this test run / configuration: run  
  
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: Docker testing on Ubuntu 20.04.1 LTS via the Phoronix Test Suite.  
New Description: compress-7zip pc personal  
  
[Performance Tip] The powersave CPU scaling governor is currently in use. It's possible to obtain greater performance if using the performance governor.  
To change behavior, run:  
echo performance | tee /sys/devices/system/cpu/cpu*/cpufreq/scaling_governor  
Reference: https://openbenchmarking.org/result/1706268-TN-CPUGOVERN32  
  
To stop showing performance tips, run: phoronix-test-suite unload-module perf_tips  
Continuing in 5 seconds or press CTRL-C to stop the testing process.  
  
7-Zip Compression 16.02:  
pts/compress-7zip-1.7.1  
Test 1 of 1  
Estimated Trial Run Count: 3  
Estimated Time To Completion: 5 Minutes [22:57 UTC]  
Started Run 1 @ 22:53:26  
Started Run 2 @ 22:54:31  
Started Run 3 @ 22:55:36  
  
Compress Speed Test:  
13928  
13342  
13768  
  
Average: 13679 MIPS  
Deviation: 2.21%  
  
Comparison to 9,490 OpenBenchmarking.org samples since 4 January 2012; median result: 21418. Box plot of samples:  
[ |----- * * * * *  
^ This Result (36th Percentile): 13679  
Intel Xeon W-2155: 57290 ^ AMD Ryzen 9 5950X: 108808 ^ AMD EPYC 7402: 138897 ^ 2 x AMD EPYC 7352: 102815 ^ 2 x Intel Xeon Platinum 8280: 241699 ^ AMD EPYC 7713: 298812 ^  
2 x Intel Xeon Gold 5118: 97374 ^ AMD Ryzen Threadripper 3990X: 179399 ^ AMD EPYC 7602: 267323 ^  
AMD EPYC 7302: 95552 ^ 2 x Intel Xeon Gold 6152: 176748 ^ AMD EPYC 7702: 262153 ^  
2 x AMD EPYC 7E32: 91444 ^ 2 x Intel Xeon Gold 6230N: 172157 ^ 2 x AMD EPYC 7502: 255912 ^  
  
Do you want to view the text results of the testing (Y/n): ]
```

```
Do you want to view the text results of the testing (Y/n): Y  
compress_pc  
compress-7zip pc personal  
  
run:  
  
Processor: Intel Core i5-8300H @ 4.00GHz (4 Cores / 8 Threads), Motherboard: Dell 0YMMT6 (1.16.2 BIOS), Memory: 20GB, Disk: SK hynix PC401 NVMe 256GB, Graphics: i915drmfb (1000MHz), Audio: Realtek ALC3266, Monitor: ASUS VS229  
  
OS: Ubuntu 20.04.1 LTS, Kernel: 5.8.0-53-generic (x86_64), Display Driver: i915drmfb, Compiler: GCC 9.3.0, File-System: overlayfs, Screen Resolution: 1920x1080, System Layer: Docker  
  
7-Zip Compression 16.02  
Compress Speed Test  
MIPS > Higher Is Better  
run . 13679 ]=====]  
  
Would you like to upload the results to OpenBenchmarking.org (y/n): ]
```

En las tres imágenes del monitor del sistema se ve que las ocho hebras de mi portátil se encuentran en valores cercanos o iguales al 100% de utilización. Al inicio de las segunda y tercera imágenes se ven dos

grandes surcos de utilización de la CPU, uno en cada una de las imágenes, esto se debe a la finalización de las ejecuciones de una de las pruebas y el comienzo de la siguiente.

Si nos fijamos estrictamente en las puntuaciones obtenidas, mi ordenador ha obtenido unas puntuaciones muy superiores a las que nos han proporcionado ambas máquinas virtuales, en particular el mejor resultado por mi ordenador es 3.85 veces mejor que el obtenido en el mejor resultado de la máquina UbuntuServer, y 3.48 veces mejor que el mejor resultado obtenido por la máquina CentOS.

El resultado era algo evidente, pues mi portátil utiliza los 8 núcleos en lugar de el núcleo único que utilizan las máquinas virtuales.