

# Practica-4-resuelta.pdf



PruebaAlien



Ingeniería de Servidores



3º Grado en Ingeniería Informática



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

**¡HAZTE  
BILINGÜE!**

958 261 159  
615 834 365

academia-granada.es

**CLASES DE INGLÉS**  
**B1 B2**  
**C1**  
**BASIC English**  
(NIVEL PRINCIPIANTE)

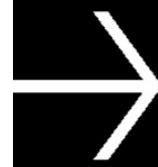
**CLASES DE FRANCÉS**  
**B1 B2**  
**DELF**  
**DELF**



**PUERTA  
REAL**  
Academia de Enseñanza

Cuanto más difícil  
sea el examen,  
más vas a disfrutar  
celebrándolo.

Uber



Practica 3

---

2021

---

Asignatura:

Ingeniería de Servidores

Uber

WUOLAH

# Índice

## Ejercicio 1

Pág. 2

|                                     |         |
|-------------------------------------|---------|
| - Instalación de Phoronix en Ubuntu | Pág. 2  |
| - Arrancar Benchmarks en Ubuntu     | Pág. 3  |
| - Instalación de Phoronix en CentOS | Pág. 6  |
| - Arrancar test en CentOS           | Pág. 6  |
| - Apache Benchmark                  | Pág. 10 |

## Ejercicio 2

Pág. 13

|   |         |
|---|---------|
| - Instalación de JMeter Ubuntu Server             | Pág. 13 |
| - Instalación de JMeter en nuestro host anfitrión | Pág. 15 |
| - Hacer test que parametrice el Host y el Puerto  | Pág. 15 |
| - Grupos de hebras alumnos y administradores      | Pág. 16 |
| - Expresión regular para extraer el token JWT     | Pág. 19 |
| - Esperas aleatorias (Gaussian Random Timer)      | Pág. 22 |
| - Uso de alumnos.csv y administrador.csv          | Pág. 22 |
| - Acceso desde el archivo apiAlumnos.log          | Pág. 24 |

## Practica 2

|                              |         |
|------------------------------|---------|
| - Cuando un disco duro falla | Pág. 26 |
|------------------------------|---------|

# Test&Train

Practica online  
tu examen de inglés  
[www.testandtrain.es](http://www.testandtrain.es)

-5%  
DTO.

Código:

**WUOT&T**



**B2  
FIRST**

**C1  
ADVANCED**

## Ejercicio 1

### Instalación de Phoronix en Ubuntu

Para instalar Phoronix tenemos que descargar el paquete usando el comando **wget**.

```
ruben@ub-server:~$ wget http://phoronix-test-suite.com/releases/repo/pts.debian/files/phoronix-test-suite_7.8.0_all.deb
--2020-12-29 12:26:55--  http://phoronix-test-suite.com/releases/repo/pts.debian/files/phoronix-test-suite_7.8.0_all.deb
Resolving phoronix-test-suite.com (phoronix-test-suite.com)... 192.211.48.82
Connecting to phoronix-test-suite.com (phoronix-test-suite.com)|192.211.48.82|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 555634 (543K) [application/x-debian-package]
Saving to: 'phoronix-test-suite_7.8.0_all.deb'

phoronix-test-suite_7.8. 100%[=====] 542.61K  751KB/s   in 0.7s

2020-12-29 12:26:56 (751 KB/s) - 'phoronix-test-suite_7.8.0_all.deb' saved [555634/555634]
```

Para instalar el paquete, tenemos que instalar un programa para descomprimir y ejecutar archivos .deb, **sudo apt-get install gdebi-core**

```
ruben@ub-server:~$ sudo apt-get install gdebi-core
[sudo] password for ruben:
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following NEW packages will be installed:
  gdebi-core
0 upgraded, 1 newly installed, 0 to remove and 78 not upgraded.
Need to get 116 kB of archives.
After this operation, 876 kB of additional disk space will be used.
Get:1 http://es.archive.ubuntu.com/ubuntu focal/universe amd64 gdebi-core all 0.9.5.7+nmu3 [116 kB]
Fetched 116 kB in 0s (266 kB/s)
Selecting previously unselected package gdebi-core.
(Reading database ... 68241 files and directories currently installed.)
Preparing to unpack .../gdebi-core_0.9.5.7+nmu3_all.deb ...
Unpacking gdebi-core (0.9.5.7+nmu3) ...
Setting up gdebi-core (0.9.5.7+nmu3) ...
Processing triggers for man-db (2.9.1-1) ...
ruben@ub-server:~$ _
```

Instalamos phoronix con el comando **sudo gdebi phoronix-test-suite\_7.8.0\_all.deb**. Y por último ejecutamos el programa.

```
for any direct or indirect damages for any use of
OpenBenchmarking.org -- including, without limitation, any lost
profits, business interruption, loss of programs, loss of
programmed data, or otherwise.

- For enterprise support, sponsorship, or other professional
inquiries, contact phoronix@phoronix.com. Community support can
be found in the Phoronix Forums at
http://www.phoronix.com/forums/.

- If you opt to submit your test results to OpenBenchmarking.org,
the final results as well as basic hardware and software details
(what is shown in the results viewer) will be shared and publicly
accessible through http://www.openbenchmarking.org/.

- Public bug reports, feature requests, and other issues can be
brought up in the Phoronix Test Suite forums, mailing list, or a
direct email to Phoronix Media.

Anonymous Usage Reporting / Statistics: If enabling the anonymous
usage reporting / statistics feature, some information about the
Phoronix Test Suite runs will be submitted to
OpenBenchmarking.org. This information is used for analytical
purposes, including but not limited to, determining the most
popular tests / suites and calculating average run-times for
different test profiles. The test results are not reported in
this process nor the installed software / hardware information,
but ambient information about the testing process. This
information is stored anonymously. More information on this
feature is available with the included documentation.

For more information on the Phoronix Test Suite and its features,
visit http://www.phoronix-test-suite.com/ or view the included
documentation.

Do you agree to these terms and wish to proceed (Y/n): y
Enable anonymous usage / statistics reporting (Y/n): y_
```

Nos pedirá que aceptemos los términos y si queremos que se informe de forma anónima, en mi caso le digo que si.

## Listado de benchmarks disponibles en Phoronix en Ubuntu

Para ver el listado de benchmark usamos el comando **phoronix-test-suite list-test**, y nos muestra los diferentes test que dispone phoronix.

```
system/blogbench          - BlogBench           Disk
system/clpeak              - Clpeak              System
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/glmark2             - GLmark2            Graphics
system/gmic                 - G'MIC               System
system/gnugp               - 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/open                - Open Porous Media Processor
system/rawtherapee          - RawTherapee         System
system/rsvg                 - librsvg             System
system/selenium             - Selenium            System
system/selenium-top-sites    - Time To Load + View Popular Websites System
system/tesseract-ocr        - Tesseract OCR      System
system/wireguard            - WireGuard + Linux Networking Stack Stress Test Network
git/aom-av1                 - AOM AV1             Processor
git/dav1                   - david               Processor
git/ravie                  - ravie               Processor
git/svt-av1                 - SVT-AV1             Processor
git/svt-hevc                - SVT-HEVC            Processor
git/svt-vp9                 - SVT-VP9             Processor
git/vpxenc                  - VP9 libvpx Encoding Processor
git/x265                   - x265               Processor
ruben@ub-server:~$ phoronix-test-suite list-tests
```

## Arrancar Benchmarks en Ubuntu

Para ejecutar los benchmarks disponibles en phoronix, es tan simple como poner **phoronix-test-suite run <nombre\_benchmark>**. En mi caso voy a ejecutar **pts/sudokut**, que hace un test del procesador. Como no esta instalado al ejecutarlo me pregunta si quiero instalarlo.

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

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

Una vez terminado la instalación, lo volvemos a ejecutar y nos muestra información del procesador, la grafica, placabase, etc... y nos pregunta si lo queremos guardar los resultados, en mi caso los guardo y nos pedirá unas cuantas cosas como el nombre del archivo, nombre a describir el test y la descripción.

```
Would you like to save these test results (Y/n): y
Enter a name to save these results under: test_sudokut
Enter a unique name to describe this test run / configuration:

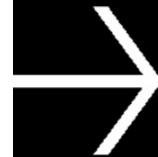
If desired, enter a new description below to better describe this result set / system configuration
under test.
Press ENTER to proceed without changes.

Current Description: Oracle VMWare testing on Ubuntu 20.04 via the Phoronix Test Suite.

New Description:
```

# Cuanto más difícil sea el examen, más vas a disfrutar celebrándolo.

# Uber



Al terminar nos hace el test del procesador:

```
Sudokut 0.4:  
pts/sudokut-1.0.1  
Test 1 of 1  
Estimated Trial Run Count: 3  
Estimated Time To Completion: 2 Minutes [17:11 UTC]  
    Started Run 1 @ 17:10:04  
    Started Run 2 @ 17:10:20  
    Started Run 3 @ 17:10:36  
  
Total Time:  
14.988317966461  
14.864675045013  
14.684154033661  
  
Average: 14.85 Seconds  
Deviation: 1.03%  
  
Do you want to view the text results of the testing (Y/n):
```

Donde muestra información del test, nos indica que el test se ha realizado 1 de 1 veces, ha realizado 3 pruebas, nos muestra el tiempo estimado que ha tardado en completarse, muestra la hora en la que se ha realizado las 3 pruebas **Started Run 1, 2 y 3**, nos muestra el tiempo medio en segundos en lo que el procesador ha sido capaz de realizar un test y por ultimo nos muestra el **Average** (promedio) y la desviación, que nos indica como de disperso están los datos, en mi caso están muy poco dispersos ya que solo hay un 1.03%, dando tiempos muy parecidos.

Ejecutamos otro test como por ejemplo **pts/encode-mp3**

```
LAME MP3 Encoding 3.100:  
pts/encode-mp3-1.7.4  
Test 1 of 1  
Estimated Trial Run Count: 3  
Estimated Time To Completion: 3 Minutes [18:31 UTC]  
    Started Run 1 @ 18:29:01  
    Started Run 2 @ 18:29:14  
    Started Run 3 @ 18:29:26  
  
WAV To MP3:  
11.163236141205  
10.895861148834  
11.369033813477  
  
Average: 11.14 Seconds  
Deviation: 2.13%  
  
Do you want to view the text results of the testing (Y/n):
```

Tambien podemos ver el resultado en la pagina que nos da, en mi caso es <https://openbenchmarking.org/result/2012293-FO-TESTENCOD25>:

```
Results Uploaded To: https://openbenchmarking.org/result/2012293-FO-TESTENCOD25  
ruben@ub-server:~$
```

## test\_encode-mp3

| OpenBenchmarking.org                         | Phoronix Test Suite 10.2.0m3 |
|--|------------------------------|
| <b>Intel Core i7-6700 @ 3.41GHz (1 Core)</b> | Processor                    |
| <b>Oracle VirtualBox v1.2</b>                | Motherboard                  |
| <b>Intel 440FX- 82441FX PMC</b>              | Chipset                      |
| <b>2048MB</b>                                | Memory                       |
| <b>2 x 16GB VBOX HDD</b>                     | Disk                         |
| <b>VMware SVGA II</b>                        | Graphics                     |
| <b>Intel 82801AA AC 97 Audio</b>             | Audio                        |
| <b>Intel 82540EM Gigabit</b>                 | Network                      |
| <b>Ubuntu 20.04</b>                          | OS                           |
| <b>5.4.0-56-generic (x86_64)</b>             | Kernel                       |
| <b>GCC 9.3.0</b>                             | Compiler                     |
| <b>ext4</b>                                  | File-System                  |
| <b>2048x2048</b>                             | Screen Resolution            |
| <b>Oracle VMWare</b>                         | System Layer                 |

### Test\_encode-mp3 Benchmarks

```

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

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

```

System Logs  
OPC Classification

### LAME MP3 Encoding

LAME is an MP3 encoder licensed under the LGPL. This test measures the time required to encode a WAV file to MP3 format. Learn more via the OpenBenchmarking.org test page.



## Instalación de Phoronix en CentOS

La instalación es parecida a la de Ubuntu, pero con algunos cambios. Para empezar, instalamos **wget**, **php-cli** y **php-xml** que no lo trae instalado CentOS y son necesarios para la instalación de Phoronix.

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

A continuación, nos descargamos el paquete de la pagina oficial usando **wget**  
<https://phoronix-test-suite.com/releases/phoronix-test-suite-8.4.1.tar.gz>.

```
[root@localhost ruben]# wget http://phoronix-test-suite.com/releases/phoronix-test-suite-8.4.1.tar.gz
--2020-12-07 06:23:19-- http://phoronix-test-suite.com/releases/phoronix-test-suite-8.4.1.tar.gz
Resolviendo phoronix-test-suite.com (phoronix-test-suite.com)... 192.211.48.82
Conectando con phoronix-test-suite.com (phoronix-test-suite.com)|192.211.48.82|:80... conectado.
Petición HTTP enviada, esperando respuesta... 200 OK
Longitud: 873706 (853K) [application/x-gzip]
Grabando a: "phoronix-test-suite-8.4.1.tar.gz"

phoronix-test-suite-8.4.1. 100%[=====] 853,23K  913KB/s   en 0,9s

2020-12-07 06:23:21 (913 KB/s) - "phoronix-test-suite-8.4.1.tar.gz" guardado [873706/873706]

[root@localhost ruben]# ls -l
total 856
-rw-r--r--. 1 root root 873706 dic  9  2018 phoronix-test-suite-8.4.1.tar.gz
[root@localhost ruben]#
```

Lo descomprimimos con **tar xvfz phoronix-test-suite-8.4.1.tar.gz**, accedemos a la carpeta **phoronix-test-suite** y ejecutamos el archivo **./install-sh**.

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

Phoronix Test Suite Installation Completed

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

## Arrancar test en CentOS

Una vez instalado, miramos la lista de test disponibles con el comando **phoronix-test-suite list-available-tests**. Pero nos falta instalarle a centos la extensión JSON para PHP, puesto que lo necesita.

```
[root@localhost phoronix-test-suite]# phoronix-test-suite list-available-tests
The following PHP extensions are REQUIRED:
JSON      JSON support is required for OpenBenchmarking.org.

The following PHP extensions are OPTIONAL but recommended:
GD        The GD library is recommended for improved graph rendering.
POSIX    POSIX support is highly recommended.

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

Para ello ejecutamos el siguiente comando **sudo yum install php-json.x86\_64**

```
[root@localhost phoronix-test-suite]# yum install php-json.x86_64
Última comprobación de caducidad de metadatos hecha hace 1:48:47, el lun 07 dic 2020 05:01:58 EST.
Dependencias resueltas.
=====
Paquete          Arquitectura  Versión           Repositorio      Tam.
=====
Instalando:
php-json        x86_64        7.2.24-1.module_e18.2.0+313+b04d0a66      AppStream       73 k

Resumen de la transacción
=====
Instalar 1 Paquete

Tamaño total de la descarga: 73 k
Tamaño instalado: 44 k
¿Está de acuerdo [s/N]? s
Descargando paquetes:
php-json-7.2.24-1.module_e18.2.0+313+b04d0a66.x86_64.rpm      318 kB/s | 73 kB  00:00
Total                                         182 kB/s | 73 kB  00:00
Ejecutando verificación de operación
Verificación de operación exitosa.
Ejecutando prueba de operaciones
Prueba de operación exitosa.
Ejecutando operación
  Preparando :                                                 1/1
  Instalando  : php-json-7.2.24-1.module_e18.2.0+313+b04d0a66.x86_64 1/1
  Ejecutando scriptlet: php-json-7.2.24-1.module_e18.2.0+313+b04d0a66.x86_64 1/1
  Verificando  : php-json-7.2.24-1.module_e18.2.0+313+b04d0a66.x86_64 1/1

Instalado:
  php-json-7.2.24-1.module_e18.2.0+313+b04d0a66.x86_64

¡Listo!
```

Ejecutamos otra vez el comando anterior para ver el listado, y ya nos muestra el listado de tests:

|                           |  |           |
|---------------------------|--|-----------|
| pts/y-cruncher            | - Y-Cruncher   | Processor |
| pts/yafaray               | - YafaRay  | Processor |
| system/apache             | - Apache Benchmark                                       | System    |
| system/blender            | - Blender  | Disk      |
| system/blogbench          | - BlogBench  | System    |
| system/clpeak             | - clpeak   | System    |
| system/compress-lzma      | - LZMA Compression                                       | Processor |
| system/compress-pbzip2    | - PBZIP2 Compression                                     | System    |
| system/compress-zstd      | - Zstd Compression                                       | System    |
| 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  | Processor |
| system/gnupg              | - GnuPG  | System    |
| 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  | Processor |
| system/opm                | - Open Porous Media                                      | System    |
| system/rawtherapee        | - RawTherapee  | System    |
| system/redis              | - Redis  | System    |
| system/rsvg               | - librsvg  | System    |
| system/selenium           | - Selenium   | System    |
| system/selenium-top-sites | - Time To Load + View Popular Websites                   | System    |
| system/sqlite             | - SQLite   | System    |
| system/tesseract-ocr      | - Tesseract OCR  | System    |
| system/wireguard          | - WireGuard + Linux Networking Stack Stress Test Network | Network   |

# Cuanto más difícil sea el examen, más vas a disfrutar celebrándolo.

# Uber



Y para ejecutar un test es igual que en Ubuntu **phoronix-test-suite run <nombre\_benchmark>**, en mi caso voy a ejecutar el mismo test que en Ubuntu para compararlo.

```
System Information

PROCESSOR: Intel Core i7-6700
Core Count: 1
Extensions: SSE 4.2 + FSGSBASE
Cache Size: 8192 KB
Microcode: 0xffffffff

GRAPHICS: VMware SVGA II
Screen: 2048x2048

MOTHERBOARD: Oracle VirtualBox v1.2
BIOS Version: VirtualBox
Chipset: Intel 400FX 82441FX PMC
Audio: Intel 82B01AA AC 97 Audio
Network: Intel 82540EM Gigabit

MEMORY: 818MB

DISK: 3 x 9GB VBOX HDD
File-System: ext4
Mount Options: relatime rw seclabel
Disk Scheduler: MQ-DEADLINE

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

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

Para ello nos muestra información del servidor.

Y a continuación hace el test:

```
Sudokut 0.4:
pts@sudokut-1.0.1
Test 1 of 1
Estimated Trial Run Count: 3
Estimated Time To Completion: 2 Minutes [07:00 EST]
Started Run 1 @ 06:58:32
Started Run 2 @ 06:58:52
Started Run 3 @ 06:59:11

Total Time:
18.350058078766
18.273214101791
17.900588888615

Average: 18.17 Seconds
Deviation: 1.32%

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

Como se puede ver el tiempo es mayor al de Ubuntu, los datos están un poco más dispersos que los de Ubuntu (1.03%(ubuntu) < 1.32 (centos)), pero aun así no están muy dispersos, con lo que los 3 tiempos son muy similares.

Ejecutamos otro test a ejecutar como por ejemplo **system/apache**

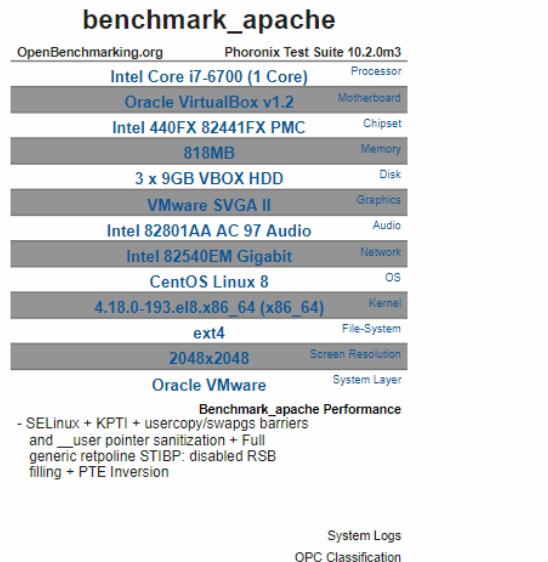
```
Apache Benchmark 2.4.29:
system/apache-1.1.0
Test 1 of 1
Estimated Trial Run Count: 3
Estimated Time To Completion: 9 Minutes [07:47 EST]
    Running Pre-Test Script @ 07:39:04
    Running Pre-Test Script @ 07:39:09
    Started Run 1 @ 07:39:09
    Started Run 2 @ 07:47:45
    Started Run 3 @ 07:51:34
    Started Run 4 @ 07:55:21 *
    Started Run 5 @ 07:59:08 *
    Started Run 6 @ 08:03:09 *
    Started Run 7 @ 08:07:22 *
    Started Run 8 @ 08:12:12 *
    Started Run 9 @ 08:16:50 *
    Running Post-Test Script @ 08:21:08
    Running Post-Test Script @ 08:21:14

Static Web Page Serving:
1948.11
4432.62
4457.31
4451.24
4280.03
3991.77
3479.71
3626.82
3913.84

Average: 3833.49 Requests Per Second
Deviation: 28.66%
```

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

Los resultados se pueden ver en esta pagina, ya que le he indicado que lo guarde en la pagina de openbenchmarking, <https://openbenchmarking.org/result/2012310-SK-BENCHMARK83>



System Logs  
OPC Classification

## Apache Benchmark

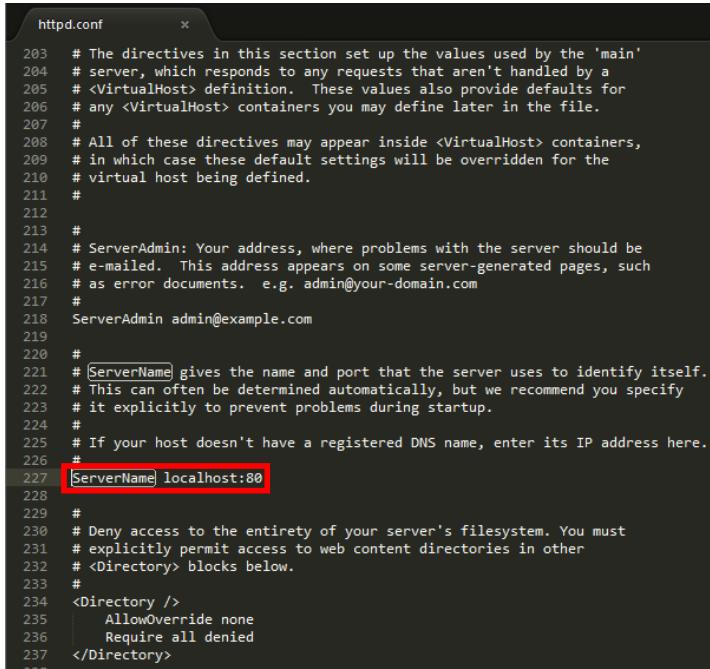
This is a test of ab, which is the Apache benchmark program, using the system's supplied Apache web server. Learn more via the OpenBenchmarking.org test page.



## Apache Benchmark

Este en mi caso ya los traen instalados en Ubuntu y centos, porque en la practica 2 instalamos apache y traen en la instalación el comando **ab**, pero para monitorizar las 2 maquinas virtuales tenemos que instalar un programa en mi host anfitrión **apache benchmarking**, que me lo descargare de esta pagina <https://www.apachelounge.com/download/>

Una vez descargado lo descomprimo en la carpeta raíz del sistema, en mi caso en C. Accedemos a la carpeta **Apache24/conf**, y modificamos el archivo **httpd.conf**. Por defecto escucha por el puerto 80.



```
httpd.conf *  
203 # The directives in this section set up the values used by the 'main'  
204 # server, which responds to any requests that aren't handled by a  
205 # <VirtualHost> definition. These values also provide defaults for  
206 # any <VirtualHost> containers you may define later in the file.  
207 #  
208 # All of these directives may appear inside <VirtualHost> containers,  
209 # in which case these default settings will be overridden for the  
210 # virtual host being defined.  
211 #  
212 #  
213 #  
214 # ServerAdmin: Your address, where problems with the server should be  
215 # e-mailed. This address appears on some server-generated pages, such  
216 # as error documents. e.g. admin@your-domain.com  
217 #  
218 ServerAdmin admin@example.com  
219 #  
220 #  
221 # [ServerName] gives the name and port that the server uses to identify itself.  
222 # This can often be determined automatically, but we recommend you specify  
223 # it explicitly to prevent problems during startup.  
224 #  
225 # If your host doesn't have a registered DNS name, enter its IP address here.  
226 #  
227 [ServerName] localhost:80  
228 #  
229 #  
230 # Deny access to the entirety of your server's filesystem. You must  
231 # explicitly permit access to web content directories in other  
232 # <Directory> blocks below.  
233 #  
234 <Directory />  
235 | AllowOverride none  
236 | Require all denied  
237 </Directory>
```

Descomentamos la variable **ServerName** y le cambiamos el nombre a **localhost:80**.

Ahora arrancamos apache en **C:/Apache24/bin/httpd.exe** y comprobamos que funciona



Ahora vamos a generar carga en las máquinas virtuales, empezando con Ubuntu, para ello usamos el comando **ab -c 1 -n 100 http://192.168.56.105**

- **-c:** la concurrencia, dicho de otro modo, el numero de paquetes que va a emitir, en mi caso de 1 en 1.
- **-n:** es el numero de peticiones que se envían
- La dirección del servidor

Comprobamos antes de ejecutar de que el servidor apache esta arrancado:

```
ruben@ub-server:~$ systemctl status apache2
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor preset: enabled)
   Active: active (running) since Thu 2020-12-31 11:23:43 UTC; 2min 27s ago
     Docs: https://httpd.apache.org/docs/2.4/
          Main PID: 38221 (apache2)
             Tasks: 6 (limit: 2282)
            Memory: 14.2M
           CGroup: /system.slice/apache2.service
                   ├─38221 /usr/sbin/apache2 -k start
                   ├─38255 /usr/sbin/apache2 -k start
                   ├─38256 /usr/sbin/apache2 -k start
                   ├─38257 /usr/sbin/apache2 -k start
                   ├─38258 /usr/sbin/apache2 -k start
                   └─38259 /usr/sbin/apache2 -k start

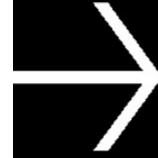
Dec 31 11:23:42 ub-server systemd[1]: Starting The Apache HTTP Server...
Dec 31 11:23:43 ub-server apachectl[38205]: AH00558: apache2: Could not reliably determine the server's fully qualified name, using 192.168.56.105 for ServerName
Dec 31 11:23:43 ub-server systemd[1]: Started The Apache HTTP Server.
lines 1-18/18 (END)
```

Y ejecutamos el comando:

```
C:\Apache24\bin>ab -c 1 -n 100 http://192.168.56.105/
This is ApacheBench, Version 2.3 <$Revision: 1879490 $>
Copyright 1996 Adam Twiss, Zeus Technology Ltd, http://www.zeustech.net/
Licensed to The Apache Software Foundation, http://www.apache.org/
Benchmarking 192.168.56.105 (be patient)....done
League of Subir WHOLA sub... Entrada Dispositivos Ayuda
Server Software: Apache/2.4.41
Server Hostname: 192.168.56.105
Server Port: 80
Document Path: /
Document Length: 10918 bytes
Concurrency Level: 1
Time taken for tests: 0.317 seconds
Complete requests: 100
Failed requests: 0
Total transferred: 1119200 bytes
HTML transferred: 1091800 bytes
Requests per second: 315.10 [#/sec] (mean)
Time per request: 3.174 [ms] (mean)
Time per request: 3.174 [ms] (mean, across all concurrent requests)
Transfer rate: 3443.91 [Kbytes/sec] received
Este equipo Musica
Connection Times (ms)
  min mean[+/-sd] median max
Connect: 0 1 0.5 1
Processing: 2 0.7 0.3 3
Waiting: 1 0.6 0.1 1
Total: 2 0.6 0.3 7
Percentage of the requests served within a certain time (ms)
  50% 3
  66% 3
  75% 3
  80% 4
  90% 4
  95% 4
  98% 5
  99% 7
 100% 7 (longest request)
C:\Apache24\bin> English class
```

# Cuanto más difícil sea el examen, más vas a disfrutar celebrándolo.

Uber



Como se puede ver nos muestra información del servidor, como el software que usa, el nombre del host, puerto, etc. Pero los datos más importantes son:

- **Time taken for tests:** Es el tiempo estimado que tardara
- **Requests per second:** Número de solicitudes por segundo
- **Time per request:** tiempo que tarda por solicitud
- **Transfer rate:** frecuencia o ratio transferido en KBytes/segundo
- **Connection times:** te muestra el tiempo mínimo, máximo, mediana, ... en conectarse, procesarse, esperar, y el total de todo.
- **Percentage of the requests server within a certain time:** nos muestra el porcentaje de solicitudes al servidor con respecto el tiempo determinado.

Con CentOS:

Comprobamos que esta arrancado

```
[root@localhost phoronix-test-suite]# systemctl status httpd
● httpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; enabled; vendor preset: disabled)
   Drop-In: /usr/lib/systemd/system/httpd.service.d
             └─php-fpm.conf
     Active: active (running) since Mon 2020-12-07 08:21:14 EST; 1h 29min ago
       Docs: man:httpd.service(8)
 Main PID: 89147 (httpd)
   Status: "Waiting, Listening on: port 80"
     Tasks: 213 (limit: 5619)
    Memory: 24.5M
      CGroup: /system.slice/httpd.service
              ├─89147 /usr/sbin/httpd -DFOREGROUND
              ├─89148 /usr/sbin/httpd -DFOREGROUND
              ├─89151 /usr/sbin/httpd -DFOREGROUND
              ├─89152 /usr/sbin/httpd -DFOREGROUND
              └─89153 /usr/sbin/httpd -DFOREGROUND

dic 07 08:21:13 localhost.localdomain systemd[1]: Starting The Apache HTTP Server...
dic 07 08:21:14 localhost.localdomain httpd[89147]: AH00558: MMap: Could not reliably determine the address of the mapping to use
dic 07 08:21:14 localhost.localdomain systemd[1]: Started The Apache HTTP Server.
dic 07 08:21:14 localhost.localdomain httpd[89147]: Server configured, listening on: port 80
lines 1-21 (END)
```

Y ejecutamos el mismo comando, pero para centos:

```
C:\Apache24\bin>ab -c 1 -n 100 http://192.168.56.110/
This is ApacheBench, Version 2.3 <$Revision: 1879490 $>
Copyright 1996 Adam Twiss, Zeus Technology Ltd, http://www.zeustech.net/
Licensed to The Apache Software Foundation, http://www.apache.org/
Benchmarking 192.168.56.110 (be patient)....done
Server Software:        Apache/2.4.37
Server Hostname:        192.168.56.110
Server Port:            80
Document Path:          /
Document Length:        16 bytes
Concurrency Level:      1
Time taken for tests:   0.485 seconds
Complete requests:      100
Failed requests:         0
Display usage information (this message and help for
Total transferred:      18900 bytes
TLS Server Name Indication (SNI) supported
HTML transferred:       1600 bytes
Specify SSL/TLS cipher suite (See options -k and -K)
Requests per second:    206.05 [#/sec] (mean)
Time per request:       4.853 [ms] (mean), T95 ~4.853 ms
Time per request:       4.853 [ms] (mean, across all concurrent requests)
Transfer rate:          38.03 [Kbytes/sec] received
Connection Times (ms)
              min  mean  [-+sd] median  max
Connect:        0    1  0.5 php-fpm 1
Processing:    2    4  0.4 active  3 running 97 since Mon 2020-12-07 08:21:14
Waiting:       2    4  0.3 available 134 serv 95 (B)
Total:         3    5  0.5 189.4147 (4httpd) 98
                           Steady. Running, Listening on port 80"
Percentage of the requests served within a certain time (ms)
 50%    4
 60%    4
 66%    4
 75%    4
 80%    4
 88%    5
 90%    5
 95%    5
 98%   11
 99%   98
100%  98 (longest request)
C:\Apache24\bin>systemctl status httpd
● httpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; enabled; vendor preset: disabled)
   Drop-In: /usr/lib/systemd/system/httpd.service.d
             └─php-fpm.conf
     Active: active (running) since Mon 2020-12-07 08:21:14 EST; 1h 29min ago
       Docs: man:httpd.service(8)
 Main PID: 89147 (httpd)
   Status: "Waiting, Listening on: port 80"
     Tasks: 213 (limit: 5619)
    Memory: 24.5M
      CGroup: /system.slice/httpd.service
              ├─89147 /usr/sbin/httpd -DFOREGROUND
              ├─89148 /usr/sbin/httpd -DFOREGROUND
              ├─89151 /usr/sbin/httpd -DFOREGROUND
              ├─89152 /usr/sbin/httpd -DFOREGROUND
              └─89153 /usr/sbin/httpd -DFOREGROUND

dic 07 08:21:14 localhost.localdomain systemd[1]: Starting The Apache HTTP Server...
dic 07 08:21:14 localhost.localdomain httpd[89147]: AH00558: MMap: Could not reliably determine the address of the mapping to use
dic 07 08:21:14 localhost.localdomain systemd[1]: Started The Apache HTTP Server.
dic 07 08:21:14 localhost.localdomain httpd[89147]: Server configured, listening on: port 80
lines 1-21 (END)
```

Realizado por Ruben Girela Castellón

Uber

Como se puede ver CentOS es un segundo y medio más lento que Ubuntu, el numero de solicitudes por segundo es menor que en Ubuntu, el tiempo que tarda por solicitud es de 1 ms mayor al de Ubuntu y como se puede ver en el connection Times Ubuntu Server tarda muchísimo menos en procesar y esperar que CentOS Server.

## Ejercicio 2

Instalación de JMeter Ubuntu Server

Para instalar JMeter, tenemos que instalar **docker** y **Docker-compose**

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

Una vez instalado comprobamos que están instalados:

```
ruben@ubuntu-server:~$ docker -v
Docker version 19.03.8, build afacb8b7f0
ruben@ubuntu-server:~$ docker-compose -v
docker-compose version 1.25.0, build unknown
ruben@ubuntu-server:~$
```

A continuación, nos vamos a <https://github.com/davidPalomar-ugr/iseP4JMeter>, donde nos descargaremos el repositorio JMeter. Para descargarlo usamos el comando **git clone**:

```
ruben@ubuntu-server:~$ git clone http://github.com/davidPalomar-ugr/iseP4JMeter
Cloning into 'iseP4JMeter'...
warning: redirecting to https://github.com/davidPalomar-ugr/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 | 196.00 KiB/s, done.
Resolving deltas: 100% (706/706), done.
Updating files: 100% (73/73), done.
ruben@ubuntu-server:~$
```

Una vez descargado accedemos a la carpeta **iseP4JMeter** para poder ejecutar el programa, para levantar el servicio, se usa el comando **sudo Docker-compose up**, la primera vez tarda, porque tiene que descargarse varios contenedores, ya que esta aplicación se basa en otros contenedores públicos que distribuyen en forma de una aplicación de contenedores Docker sobre Compose.

Si queremos levantar el servicio y que nos imprima la salida del servidor usamos el siguiente comando **sudo docker-compose up &**. Y si queremos parar el servicio es **sudo docker-compose down**.

```
ruben@ubuntu-server:~/iseP4JMeter$ sudo docker-compose down
[sudo] password for ruben:
Stopping isep4jmeter_nodejs_1 ... done
Removing isep4jmeter_mongodbinit_1 ... done
Removing isep4jmeter_nodejs_1      ... done
Removing isep4jmeter_mongodb_1     ... done
Removing network isep4jmeter_default
ruben@ubuntu-server:~/iseP4JMeter$ _
```

Y antes de terminar con el servidor, vamos a habilitar el puerto 3000, ya que en el repositorio github explica que para acceder al servicio hay que usar el puerto 3000 y como ya sabemos en Ubuntu salvo los especificados están bloqueados todos los puertos.

```

ruben@ubuntu-server:~/iseP4JMeter$ sudo ufw status
[sudo] password for ruben:
Status: active

To           Action    From
--          ----     ---
22022        ALLOW     Anywhere
80          ALLOW     Anywhere
443         ALLOW     Anywhere
10050/tcp    ALLOW     Anywhere
10051/tcp    ALLOW     Anywhere
22022 (v6)   ALLOW     Anywhere (v6)
80 (v6)      ALLOW     Anywhere (v6)
443 (v6)     ALLOW     Anywhere (v6)
10050/tcp (v6) ALLOW     Anywhere (v6)
10051/tcp (v6) ALLOW     Anywhere (v6)

ruben@ubuntu-server:~/iseP4JMeter$
```

Lo habilitamos:

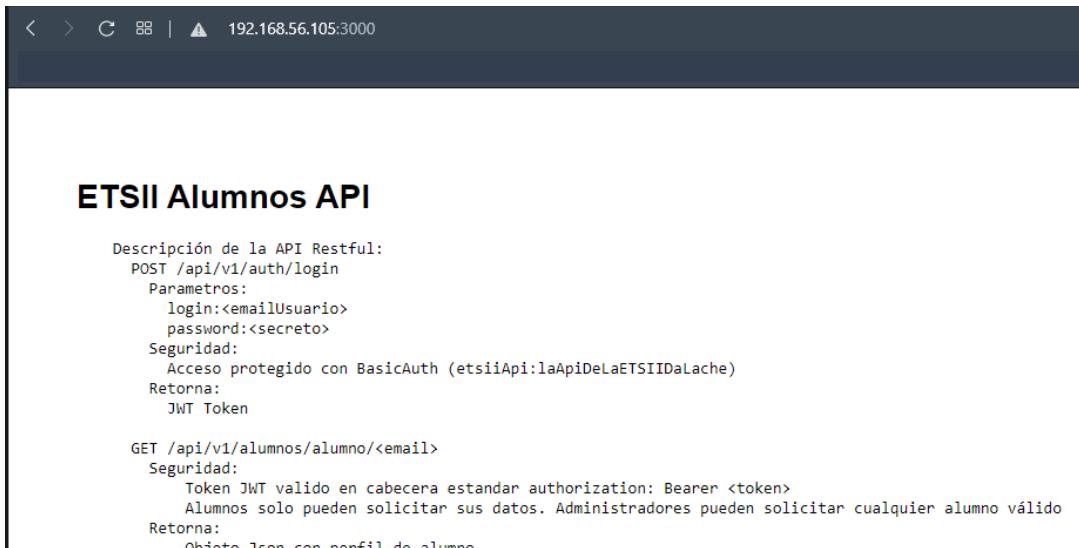
```

ruben@ubuntu-server:~/iseP4JMeter$ sudo ufw allow 3000/tcp
Rule added
Rule added (v6)
ruben@ubuntu-server:~/iseP4JMeter$ sudo ufw status
Status: active

To           Action    From
--          ----     ---
22022        ALLOW     Anywhere
80          ALLOW     Anywhere
443         ALLOW     Anywhere
10050/tcp    ALLOW     Anywhere
10051/tcp    ALLOW     Anywhere
3000/tcp     ALLOW     Anywhere
22022 (v6)   ALLOW     Anywhere (v6)
80 (v6)      ALLOW     Anywhere (v6)
443 (v6)     ALLOW     Anywhere (v6)
10050/tcp (v6) ALLOW     Anywhere (v6)
10051/tcp (v6) ALLOW     Anywhere (v6)
3000/tcp (v6) ALLOW     Anywhere (v6)

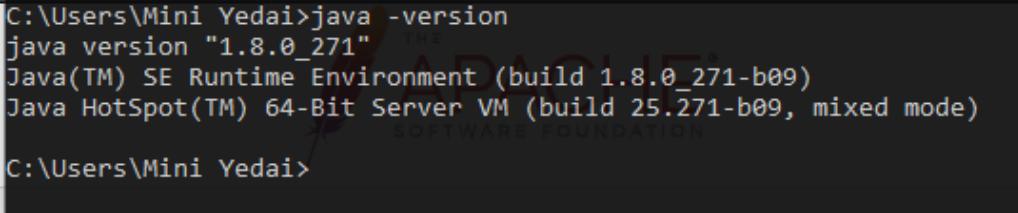
ruben@ubuntu-server:~/iseP4JMeter$
```

Para comprobar que esta arrancado podemos usar el comando **sudo Docker-compose ps** para administrar los diferentes contenedores que hay funcionando o ir a nuestro navegador y poner la dirección IP y el puerto 3000.



## Instalación de JMeter en nuestro host anfitrión

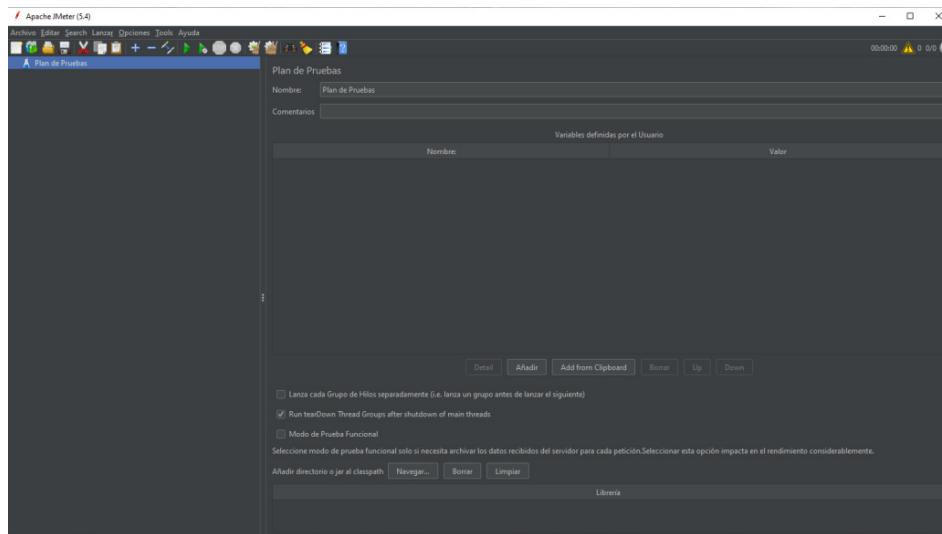
Para ello nos descargamos JMeter de la página oficial [https://jmeter.apache.org/download\\_jmeter.cgi](https://jmeter.apache.org/download_jmeter.cgi) y nos descargamos el paquete comprimido, lo descomprimimos y ya podemos usar el programa, ya que este programa es portable para cualquier sistema operativo ya que está hecho en Java, lo único que tenemos que tener instalado es Java.



```
C:\Users\Mini Yedai>java -version
java version "1.8.0_271"
Java(TM) SE Runtime Environment (build 1.8.0_271-b09)
Java HotSpot(TM) 64-Bit Server VM (build 25.271-b09, mixed mode)
C:\Users\Mini Yedai>
```

En mi caso tengo la versión 8 ya que uso el JMeter 5.4 que requiere de Java 8.

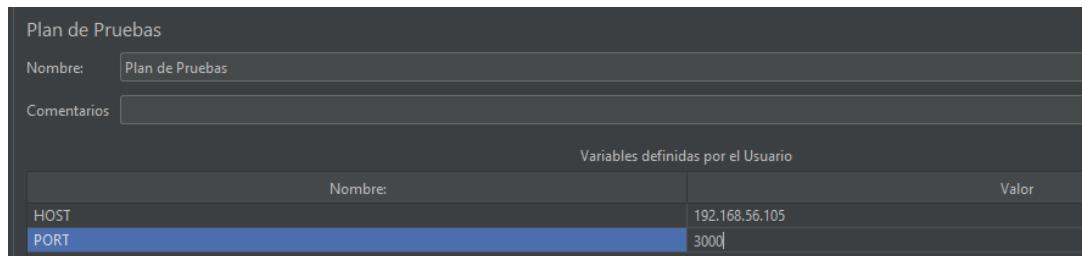
Para ejecutar el programa nos vamos a la carpeta **apache-jmeter-5.4/bin** y ejecutamos el programa **ApacheJMeter.jar**:



La primera vez que lo ejecutamos no tiene nada, donde empezamos en el nodo raíz **Test Plan**

### Hacer un test que parametrice el Host y el Puerto en el Test Plan

Para ello nos vamos al nodo raíz que es **Test Plan** y hay metemos los parámetros **Host** y **Port**.



# Cuanto más difícil sea el examen, más vas a disfrutar celebrándolo.



Hacer 2 grupos de hebras distintos para simular el acceso de los alumnos y los administradores

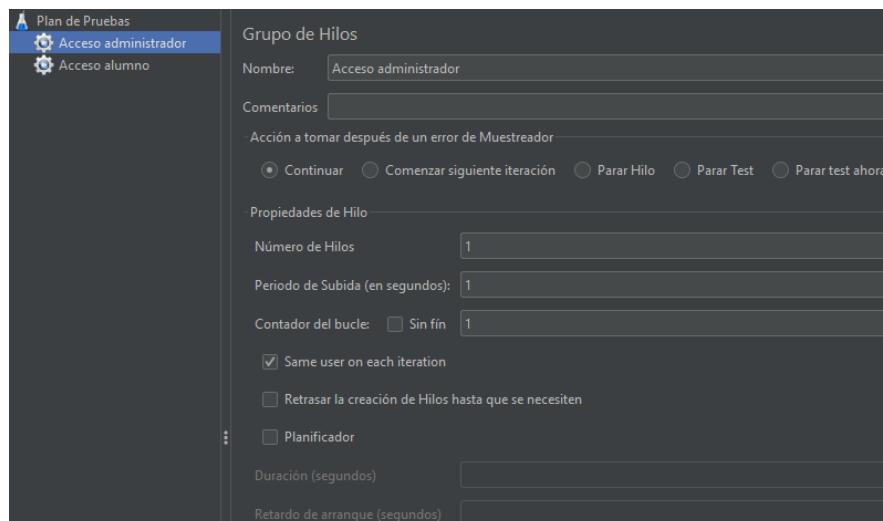
Antes de hacer nada vamos a ver 3 archivos que se encuentran en **iseP4JMeter/jMeter**, donde están las cuentas de los alumnos (**alumnos.csv**), administradores (**administradores.csv**) y un historial de accesos que han realizado los administradores (**apiAlumnos.log**).

```
ruben@ubuntu-server:~/iseP4JMeter/jMeter$ ls -l
total 536
-rw-rw-r-- 1 ruben ruben    673 Dec 23 18:15 administradores.csv
-rw-rw-r-- 1 ruben ruben  31922 Dec 23 18:15 alumnos.csv
-rw-rw-r-- 1 ruben ruben 511803 Dec 23 18:15 apiAlumnos.log
ruben@ubuntu-server:~/iseP4JMeter/jMeter$ _
```

Para ello vamos a coger una cuenta de administradores, con la que accederemos en JMeter, en mi caso [cookecooley@etsii.ugr.es](mailto:cookecooley@etsii.ugr.es) con la contraseña **do** y para el de alumnos [potterwooten@tropoli.com](mailto:potterwooten@tropoli.com) con la contraseña **dolor**.

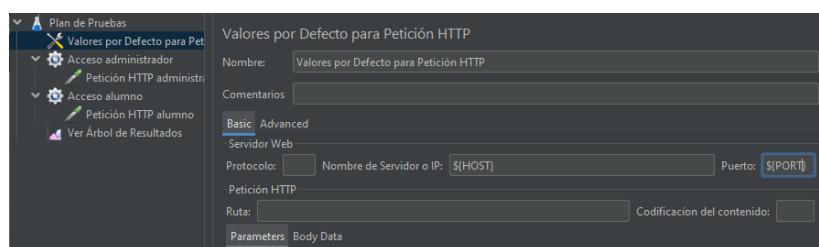
**cookecooley@etsii.ugr.es,do    potterwooten@tropoli.com,dolor**

Una vez sabido eso, nos vamos a JMeter y creamos 2 grupos de hebras como dice el título uno para los accesos de los alumnos y otro para los administradores, para ello pulsamos botón derecho sobre el nodo raíz, add o añadir, hilos o threads y thread group o grupo de hilos.



En mi caso he dejado la configuración por defecto en ambos grupos de hebras.

A continuación, vamos a añadir un elemento de configuración HTTP por defecto para las peticiones HTTP, donde meteremos el host y el puerto para no introducirlos en cada petición:



Ahora para simular el acceso de un usuario alumno y de un administrador, creamos 2 peticiones HTTP para cada grupo de hebras, donde introduciremos la ruta, el tipo de petición (GET, POST,...), el usuario y la contraseña:

Para el acceso como administrador

Petición HTTP

Nombre: Petición HTTP administrador

Comentarios

**Basic Advanced**

Servidor Web

Protocolo: Nombre de Servidor o IP: Puerto:

Petición HTTP

POST Ruta: /api/v1/auth/login Codificación del contenido:

Redirigir Automáticamente  Seguir Redirecciones  Utilizar KeepAlive  Usar 'multipart/form-data' para HTTP POST  Cabeceras compatibles con navegadores

**Parameters Body Data Files Upload**

Enviar Parámetros Con la Petición:

| Nombre:  | Valor                    | ¿Codificar?                         | Content-Type | ¿Incluir Equals?                    |
|----------|--------------------------|-------------------------------------|--------------|-------------------------------------|
| login    | cookecooley@etsii.ugr.es | <input checked="" type="checkbox"/> | text/plain   | <input checked="" type="checkbox"/> |
| password | do                       | <input checked="" type="checkbox"/> | text/plain   | <input checked="" type="checkbox"/> |

Para acceso como alumno:

Petición HTTP

Nombre: Petición HTTP alumno

Comentarios

**Basic Advanced**

Servidor Web

Protocolo: Nombre de Servidor o IP: Puerto:

Petición HTTP

POST Ruta: /api/v1/auth/login Codificación del contenido:

Redirigir Automáticamente  Seguir Redirecciones  Utilizar KeepAlive  Usar 'multipart/form-data' para HTTP POST  Cabeceras compatibles con navegadores

**Parameters Body Data Files Upload**

Enviar Parámetros Con la Petición:

| Nombre:  | Valor                    | ¿Codificar?                         | Content-Type | ¿Incluir Equals?                    |
|----------|--------------------------|-------------------------------------|--------------|-------------------------------------|
| login    | potterwooten@tropoli.com | <input checked="" type="checkbox"/> | text/plain   | <input checked="" type="checkbox"/> |
| password | dolor                    | <input checked="" type="checkbox"/> | text/plain   | <input checked="" type="checkbox"/> |

Para visualizar el resultado de los accesos creamos en cada grupo un árbol de resultados, para ver de forma más cómoda lo que está pasando.

Ver Árbol de Resultados

Nombre: Ver Árbol de Resultados

Comentarios

Escribir todos los datos a Archivo

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

Buscar: Sensible a mayúsculas: Expresión regular: Search Reset

Texto: Resultado del Muestreador Pregunta Datos de Respuesta

Si lo ejecutamos como esta nos devolverá el servidor un mensaje de error como este en ambos logins:

| Resultado del Muestreador                   |  |
|---|--|
| Nombre del hilo:Acceso administrador 1-1    |  |
| Comienzo de muestra:2020-12-27 13:28:00 CET |  |
| Tiempo de carga:92                          |  |
| Connect Time:88                             |  |
| Latencia:92                                 |  |
| Tamaño en bytes:406                         |  |
| Sent bytes:270                              |  |
| Headers size in bytes:406                   |  |
| Body size in bytes:0                        |  |
| Conteo de muestra:1                         |  |
| Conteo de error:1                           |  |
| Data type ("text" "bin" ""):text            |  |
| <b>Código de respuesta:401</b>              |  |
| Mensaje de respuesta:Unauthorized           |  |

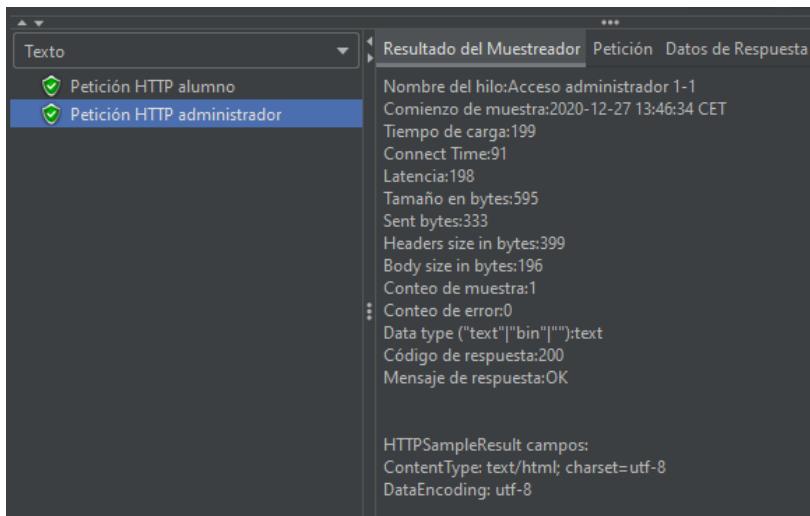
Esto se debe a que no tenemos autorización, porque no tenemos puesto el acceso protegido con BasicAuth (etsiiApi:laApiDeLaETSIIDaLache). Para añadírselo, le añadimos un nodo gestor de autorización HTTP para cada grupo de hebras que usen la petición HTTP tipo POST:

| Autorizaciones Almacenadas en el Gestor de Autorización |                   |            |         |                 |           |
|---|-------------------|------------|---------|-----------------|-----------|
| URL Base  | Nombre de Usuario | Contraseña | Dominio | Dominio (realm) | Mechanism |
| http://\${HOST}:\${PORT}/api/v1/auth/login              | etsiiApi          | *****      |         |                 | BASIC     |

En el que le añadimos una autorización con la URL del acceso, en mi caso del login, el usuario y contraseña de la API no de alumnos o administrador y el tipo de autorización, en mi caso BASIC.

Y si lo ejecutamos nos funciona:

| Resultado del Muestreador                   |  |
|---|--|
| Nombre del hilo:Acceso alumno 2-1           |  |
| Comienzo de muestra:2020-12-27 13:46:34 CET |  |
| Tiempo de carga:194                         |  |
| Connect Time:90                             |  |
| Latencia:194                                |  |
| Tamaño en bytes:583                         |  |
| Sent bytes:336                              |  |
| Headers size in bytes:399                   |  |
| Body size in bytes:184                      |  |
| Conteo de muestra:1                         |  |
| Conteo de error:0                           |  |
| Data type ("text" "bin" ""):text            |  |
| <b>Código de respuesta:200</b>              |  |
| Mensaje de respuesta:OK                     |  |

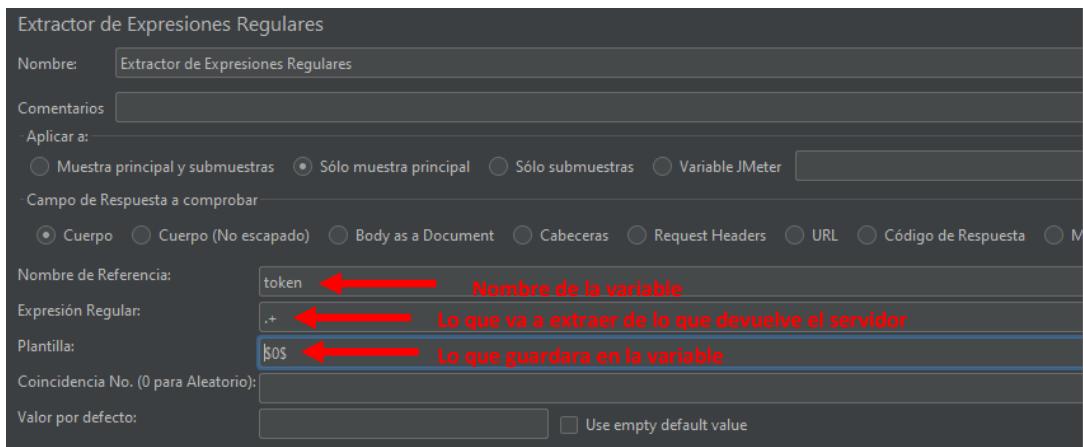


```
nodejs_1 | POST /api/v1/auth/login 200 100.656 ms - 184
nodejs_1 | POST /api/v1/auth/login 200 102.483 ms - 196
```

Devolviéndonos en ambos accesos un token.

### Usar expresiones regulares para extraer el token JWT

Pero si queremos hacer peticiones tipo GET con el token que nos devuelve tenemos que guardar ese token en una variable y usarlo, para ello creamos un extractor de expresiones regulares, de tal forma que aplicando una expresión regular sobre lo que nos devuelve el servidor obtenemos la información que necesitamos, en mi caso como solo devuelve el token pues la expresión regular es muy simple (.+).



Y por último para recibir la información usando GET y el token, creamos otra petición HTTP, donde añadimos únicamente la ruta donde le especificamos el correo del alumno a mostrar.

# Cuanto más difícil sea el examen, más vas a disfrutar celebrándolo.

Uber



Y a parte indicar el método de seguridad usando el token que ha recibido del servidor, para ello usaremos el gestor de cabecera HTTP, que solo le afectara a la petición GET HTTP del administrador:

The screenshot shows the 'Plan de Pruebas' interface. On the left, there's a tree view of test cases: 'Valores por Defecto para Petición HTTP', 'Acceso administrador' (which contains 'Gestor de Autorización HTTP', 'Petición HTTP administrador', 'Extractor de Expresiones Regulares', and 'Petición HTTP información GET administrador'), and 'Gestor de Cabecera HTTP'. On the right, the 'Gestor de Cabecera HTTP' configuration window is open. It has fields for 'Nombre' (set to 'Gestor de Cabecera HTTP') and 'Comentarios'. Below that is a table titled 'Cabeceras Almacenadas en el Gestor de Cabeceras' with columns 'Nombre' and 'Valor'. A single row is selected, showing 'Authorization' as the name and 'Bearer \${token}' as the value. This row is highlighted with a red border.

Donde le indicamos el nombre del campo **Authorization** y el valor del campo que es **Bearer** y el token.

Lo ejecutamos para ver que funciona:

The screenshot shows the 'Ver Árbol de Resultados' interface. On the left, it shows the same tree structure as the previous screenshot. On the right, the results are displayed in a table with columns 'Resultado del Muestreador', 'Petición', and 'Datos de Respuesta'. There are three rows: 'Petición HTTP alumno', 'Petición HTTP administrador', and 'Petición HTTP información GET alumno'. The 'Petición HTTP administrador' row is expanded, showing 'Petición HTTP información GET administrador' as a child. Below the table, there is a log terminal window displaying command-line output. The output includes logs from 'nodejs\_1' and 'mongodb\_1', showing database queries and responses related to the test execution. The log ends with a successful response from 'nodejs\_1'.

```
nodejs_1      | GET /api/v1/alumnos/alumno/potterwooten@tropoli.com 200 1039.874 ms - 1410
mongodb_1     | { "t": { "$date": "2020-12-27T16:36:08.387+00:00" }, "s": "I", "c": "COMMAND", "id": "51803
  "ctx": "conn26", "msg": "Slow query", "attr": { "type": "command", "ns": "etsii.alumnos", "command": { "find": "alumnos", "filter": { "email": "potterwooten@tropoli.com" }, "projection": {}, "limit": 1, "singleBatch": true, "batchSize": 1, "returnKey": false, "showRecordId": false, "lsid": { "id": "58b2a8c7-be5f-4f8e-af8a-c17b4dce5a39" }, "db": "etsii" }, "planSummary": "IXSCAN { email: 1 }", "keysExamined": 1, "docsExamined": 1, "cursorExhausted": true, "numYields": 1, "nreturned": 1, "queryHash": "F4C9CE3A", "planCacheKey": "214EB173", "reslen": 1519, "locks": { "ReplicationStateTransition": { "acquireCount": { "r": 2 } }, "Global": { "acquireCount": { "r": 1 } }, "Database": { "acquireCount": { "r": 2 } }, "Collection": { "acquireCount": { "r": 2 } }, "Mutex": { "acquireCount": { "r": 1 } } }, "storage": { "protocol": "op_msg", "durationMillis": 825 } }
nodejs_1      | GET /api/v1/alumnos/alumno/potterwooten@tropoli.com 200 1139.793 ms - 1410
```

Y como se puede ver nos devuelve la información del alumno indicado. Pero si le indico un correo de un alumno para acceder a la información de ese alumno como otro alumno. Ejemplo iniciamos sesión con [potterwooten@tropoli.com](mailto:potterwooten@tropoli.com) y queremos acceder a la información de la alumna [isabelward@tropoli.com](mailto:isabelward@tropoli.com)

The screenshot shows the 'Petición HTTP' configuration window. It has fields for 'Nombre' (set to 'Petición HTTP información GET alumno'), 'Comentarios', and tabs for 'Basic' and 'Advanced'. Under 'Basic', there are fields for 'Servidor Web' (Protocolo: [dropdown], Nombre de Servidor o IP: [input]), 'Petición HTTP' (Method: [dropdown] set to 'GET', Ruta: [input] set to '/api/v1/alumnos/alumno/isabelward@tropoli.com').

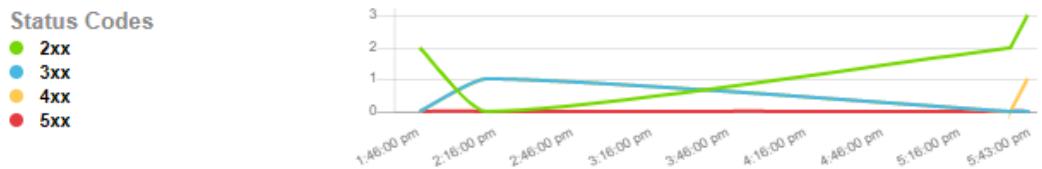
Realizado por Ruben Girela Castellón

Uber

The screenshot shows a browser developer tools Network tab. A failed request for 'Petición HTTP información GET alumno' is highlighted in red. The response body contains the text: '<h1>Solo un administrador puede consultar los datos de otro alumno</h1>'.

No nos dejara acceder a la información ya que no es administrador.

Si nos vamos a la gráfica del servidor 192.168.56.105:3000/status nos muestra los códigos que devuelve el servidor:

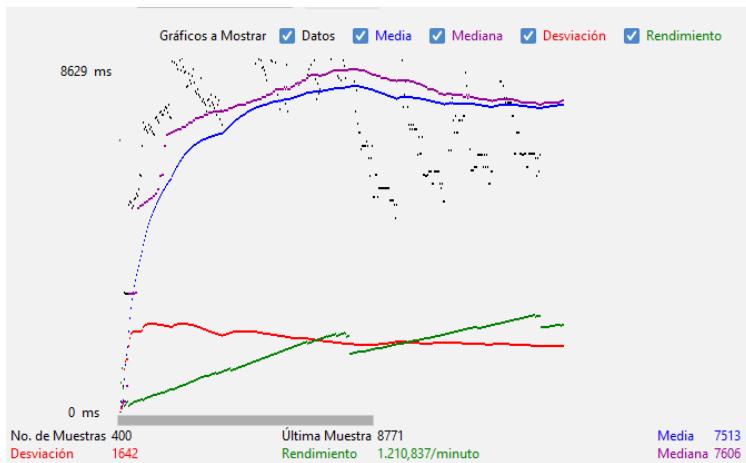


También podemos monitorizar con el programa JMeter, para ello añadimos un **informe agregado** en **añadir/receptor/ informe agregado**.

The screenshot shows the JMeter Test Plan interface. On the left, the test plan tree includes a 'Plan de Pruebas' node with several requests: 'Valores por Defecto para Petición HTTP', 'Acceso administrador' (with 'Gestor de Autorización HTTP'), 'Petición HTTP administrador' (with 'Extractor de Expresiones Regulares'), 'Petición HTTP información GET administrador' (with 'Gestor de Cabecera HTTP'), 'Acceso alumno' (with 'Gestor de Autorización HTTP'), 'Petición HTTP alumno' (with 'Extractor de Expresiones Regulares'), 'Petición HTTP información GET alumno' (with 'Gestor de Cabecera HTTP'), and 'Ver Árbol de Resultados'. On the right, the 'Informe Agregado' configuration dialog is open, showing a table of aggregated results for each request.

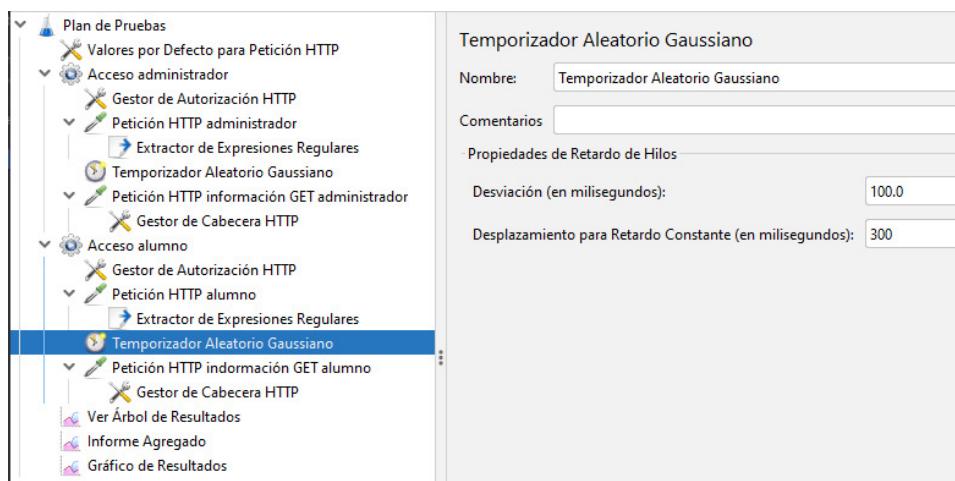
| Etiqueta    | # Mues. | Media | Mediana | 90% Line | 95% Line | 99% Line | Mín | Máx | % Error | Rendim... | Kb/sec | Sent KB... |
|-------------|---------|-------|---------|----------|----------|----------|-----|-----|---------|-----------|--------|------------|
| Petición... | 1       | 14    | 14      | 14       | 14       | 14       | 14  | 14  | 0,00%   | 71,4/sec  | 40,67  | 23,44      |
| Petición... | 1       | 27    | 27      | 27       | 27       | 27       | 27  | 27  | 0,00%   | 37,0/sec  | 21,52  | 12,04      |
| Petición... | 1       | 15    | 15      | 15       | 15       | 15       | 15  | 15  | 0,00%   | 66,7/sec  | 118,36 | 24,61      |
| Petición... | 1       | 314   | 314     | 314      | 314      | 314      | 314 | 314 | 0,00%   | 3,2/sec   | 5,65   | 1,21       |
| Total       | 4       | 92    | 15      | 314      | 314      | 314      | 14  | 314 | 0,00%   | 11,7/sec  | 13,75  | 4,10       |

E incluso graficas podemos ver en JMeter:



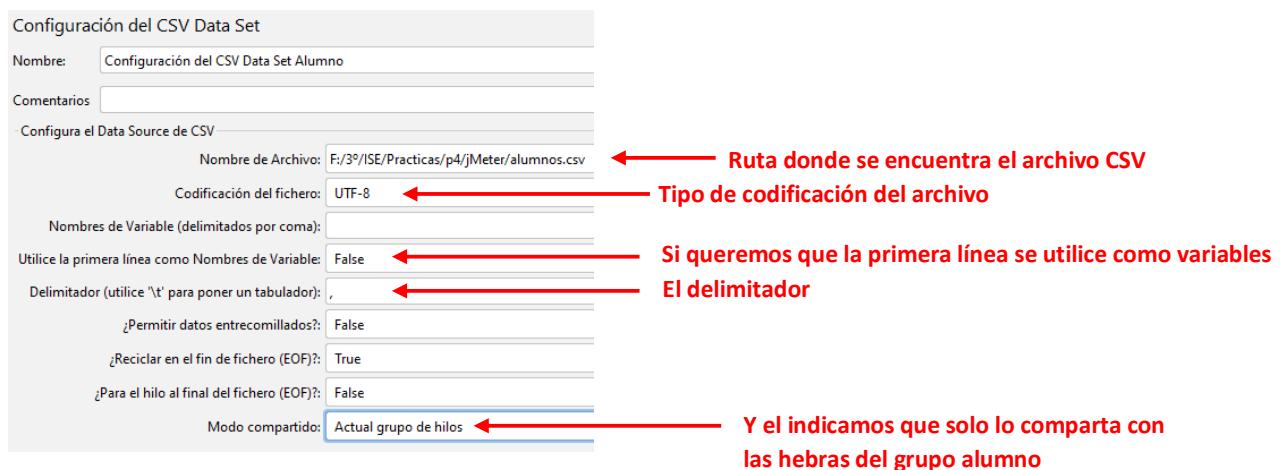
## Añadimos esperas aleatorias a cada grupo de hebras (Gaussian Random Timer)

Para ello nos vamos al grupo de hebras y añadimos el nodo temporizador aleatorio Gausiano:



Hacer las sesiones de alumno y administrador desde los archivos alumnos.csv y administrador.csv

Para ello vamos a crear un nodo configuración del CSW Data Set donde accederemos a los usuarios desde hay.



Y lo repetimos para el administrador, salvo el archivo que es **administradores.csv**.

Y por último modificamos las peticiones para iniciar sesión por las variables del archivo csv, en los 2 grupos de hebras.

Y además en el grupo de hebras de alumno modificamos también las peticiones GET para que no pida la información de un alumno, sino el suyo propio:

Como se puede ver funciona.

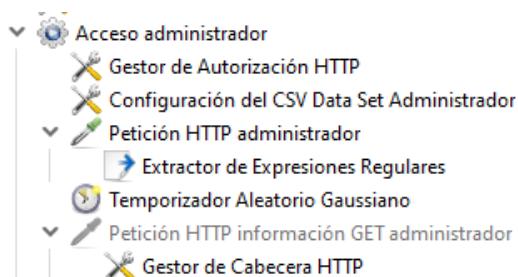
# Cuanto más difícil sea el examen, más vas a disfrutar celebrándolo.

Uber



Muestreo para simular el acceso de los administradores desde el archivo apiAlumnos.log

Para ello deshabilitamos la petición GET del grupo de hebras administrador con **Ctrl + T**.



De tal forma que se nos queda de forma gráfica con un gris claro, para entender que esta deshabilitado.

A continuación, creamos un nodo **muestreador de acceso a log**:

**Muestreador de Acceso a Log**

Nombre: Muestreador de Acceso a Log

Comentarios

Valores por defecto para Prueba

Protocol: http

Servidor: \${HOST}

Puerto: \${PORT}

Parsear Imágenes: False

Classes desplegables

Parser: org.apache.jmeter.protocol.http.util.accesslog.TCLogParser

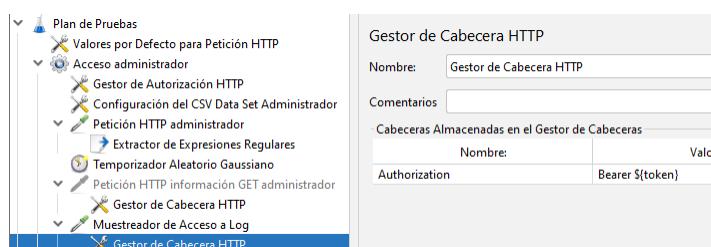
Filtro (Opcional): No definido

Ubicación del Archivo de Log

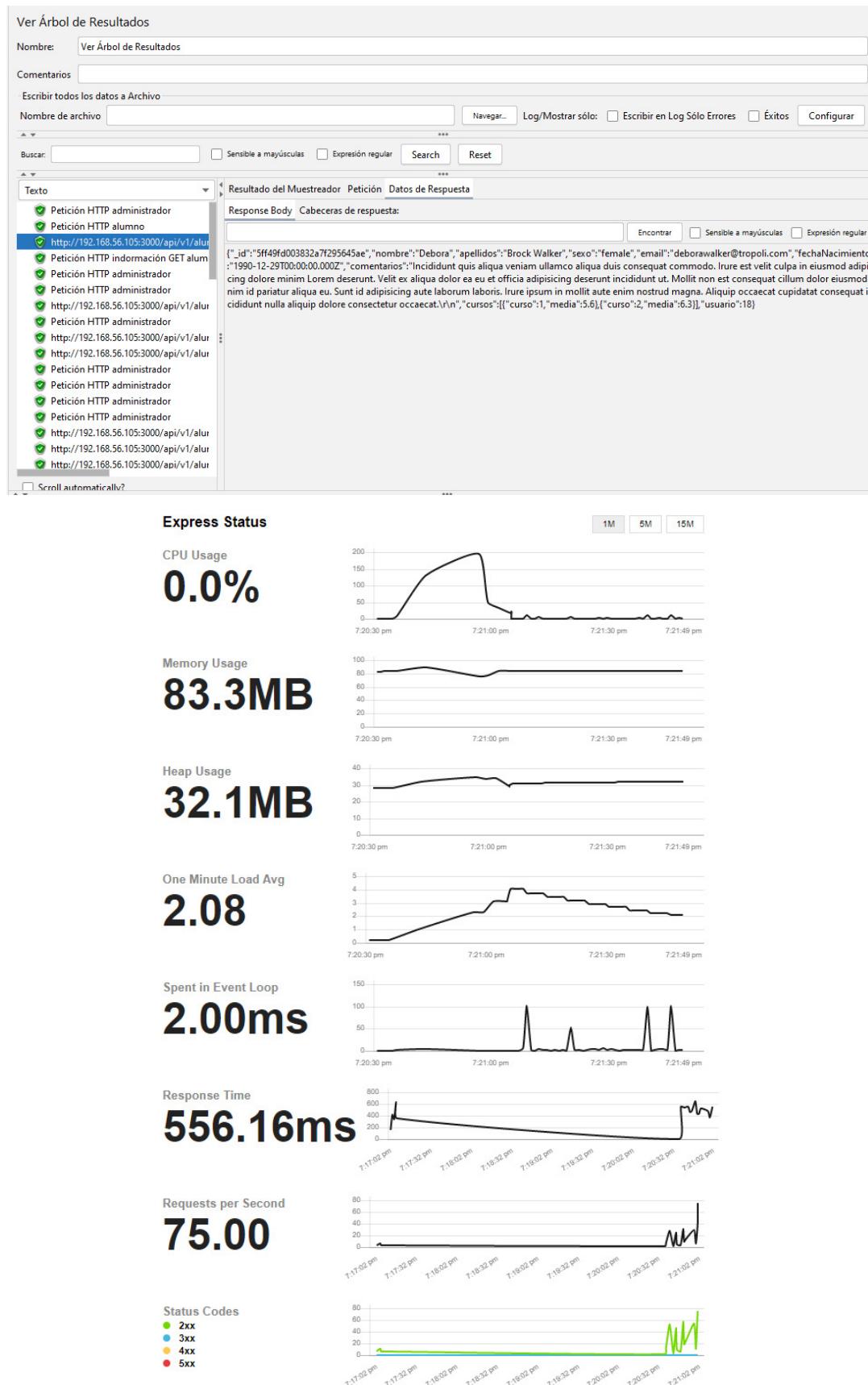
Archivo de Log: F:/3º/ISE/Prácticas/p4/jMeter/apiAlumnos.log

Donde indicamos el HOST y el PORT, puesto que los valores por defecto HTTP solo sirven para las peticiones HTTP y el archivo .log.

Y por último añadimos la cabecera donde le pasamos el token de la sesión:



## Resultado final:



Realizado por Ruben Girela Castellón

Con todo lo que has estudiado, como para suspender por llegar tarde. Uber

## Cuando un disco duro falla

Antes que nada, vamos a ver cómo podemos ver los raids y si están operativos, para ello accedemos a **cat /proc/mdstat**, donde nos mostrara información de los raids (si están activos, numero de discos conectados, etc)

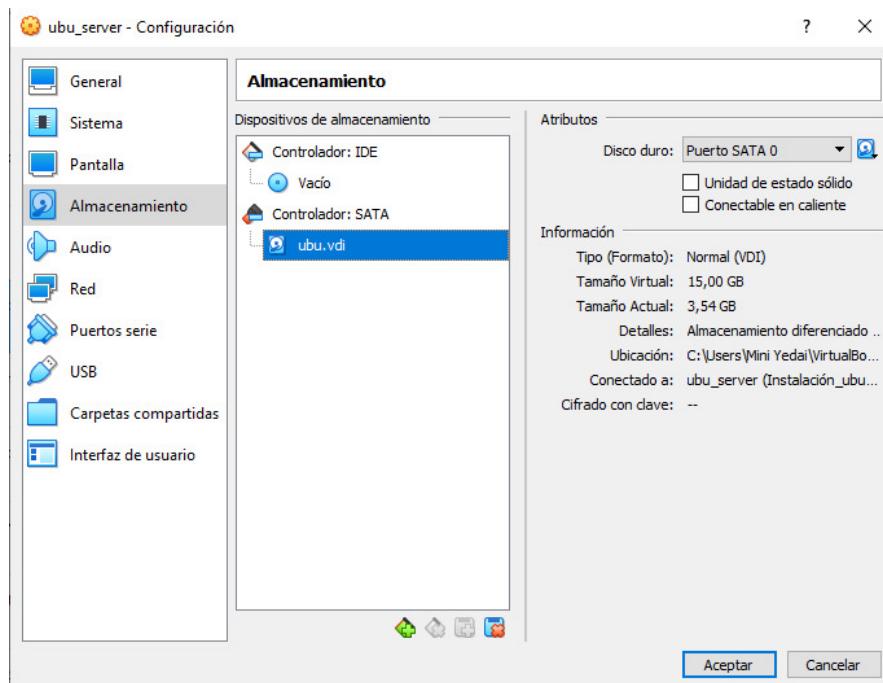
```
ruben@ub-server:~$ cat /proc/mdstat
Personalities : [raid1] [linear] [multipath] [raid0] [raid6] [raid5] [raid4] [raid10]
md1 : active raid1 sdb3[1] sda3[0]
      14695424 blocks super 1.2 [2/2] [UU]

md0 : active raid1 sda2[0] sdb2[1]
      1019904 blocks super 1.2 [2/2] [UU]

unused devices: <none>
ruben@ub-server:~$
```

En mi caso están levantados los raids 1, tienen 2 discos y ambos están funcionando (que no se han caído, roto, etc).

Para simular que un disco se ha caído en virtualbox, lo que vamos a hacer es quitar uno de los discos virtuales. Para ello nos vamos a la configuración de la máquina virtual y en almacenamiento quitamos uno de los discos virtuales:



Y a continuación ejecutamos la máquina virtual. Va a tardar un tiempo 5 minutos aprox., porque hace una comprobación de los discos y hasta que se dé cuenta de que hay un disco caído.

De tal forma que nos aparecerá esto:

```
found
  Cannot process volume group vg0
  Volume group "vg0" not found
  Cannot process volume group vg0
cryptsetup: Waiting for encrypted source device
  UUID=26a97d8a-cd56-42c7-8694-50a86203ce97...
[ 22.909362] random: crng init done
[ 22.909721] random: 7 urandom warning(s) missed due to ratelimiting
    ALERT! encrypted source device UUID=26a97d8a-cd56-42c7-8694-50a86203ce97 does not exist, can't unlock dm_crypt-0.
    Check cryptopts=source= bootarg: cat /proc/cmdline
        or missing modules, devices: cat /proc/modules; ls /dev
Dropping to a shell.

BusyBox v1.30.1 (Ubuntu 1:1.30.1-4ubuntu6.2) built-in shell (ash)
Enter 'help' for a list of built-in commands.

(initramfs) _
```

Lo que hace el sistema cuando hay algún fallo en el raid, ya sea que se ha caído un disco u otro motivo, se desactiva automáticamente. De tal forma que en vez de arrancar el grub, lo que lanza es arrancar un sistema operativo en RAM, que nos permite usar la consola de forma limitada.

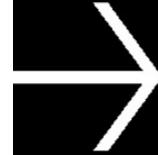
Para saber los comandos que podemos usar pulsamos 2 veces el tabulador y nos mostrara los comandos que podremos usar en la consola:

```
BusyBox v1.30.1 (Ubuntu 1:1.30.1-4ubuntu6.2) built-in shell (ash)
Enter 'help' for a list of built-in commands.

(initramfs)
[ dhclient      kbd_mode      ntfs-3g      sort
[[ dhclient-script   kill          nuke         stat
acpid          dmesg        kmod          openvt      static-sh
arch           dmsetup     ln            pdata_tools  stty
ash             du           loadfont     pidof       switch_root
awk             dumpe2fs    loadkeys     pivot_root  sync
basename       dumpkmap    loadkmap     plymouth    tail
blkid          echo         losetup     plymouthhd tee
blockdev       egrep        ls            poweroff   test
btrfs          env          lvm          printf     thin_check
busybox        expr         lzop        ps          touch
cache_check    false        mdadm       pwd         tr
cat             fbset        mdmon      readlink   true
chmod          fgrep        minips     reboot     tty
chroot         find         mkdir      reset      udevadm
chvt           fold         mke2fs     resume    umount
clear          fstrim     mkfifo      rm         uname
cmp             fstype      mknod      rmdir     uniq
cp              grep         mkswap     run-init   vgchange
cpio            gunzip     mktemp     run-parts wait-for-root
cryptroot-unlock gzip        modinfo    sed        wget
cryptsetup     halt        modprobe   seq        which
cut             head        more      setfont   wipefs
date           hostname   mount     setkeycodes yes
dd              hwclock   mount.fuse sg_inq
deallocvt      ifconfig   mount.ntfs sh
deluser        ip          mount.ntfs-3g sha512sum
devmem         ipconfig   mv        sleep
df              iscsistart nfsmount
```

# Cuanto más difícil sea el examen, más vas a disfrutar celebrándolo.

Uber



Para saber que ha pasado exactamente usamos el comando **dmesg**:

```
[ 4.270603] ata3: SATA link up 3.0 Gbps (SStatus 123 SControl 300)
[ 4.272117] ata3.00: ATA-6: VBOX HARDDISK, 1.0, max UDMA/133
[ 4.272549] ata3.00: 31457280 sectors, multi 128: LBA48 NCQ (depth 32)
[ 4.281847] ata3.00: configured for UDMA/133
[ 4.282367] scsi 2:0:0:0: Direct-Access      ATA      VBOX HARDDISK    1.0 PQ: 0 ANSI: 5
[ 4.282925] sd 2:0:0:0: [sda] 31457280 512-byte logical blocks: (16.1 GB/15.0 GiB)
[ 4.283261] sd 2:0:0:0: [sda] Write Protect is off
[ 4.283674] sd 2:0:0:0: [sda] Mode Sense: 00 3a 00 00
[ 4.283688] sd 2:0:0:0: Attached scsi generic sg1 type 0
[ 4.284040] sd 2:0:0:0: [sda] Write cache: enabled, read cache: enabled, doesn't support DPO or FUA
[ 4.343025] sda: sda1 sda2 sda3
[ 4.343749] sd 2:0:0:0: [sda] Attached SCSI disk
[ 4.613695] ata4: SATA link down (SStatus 0 SControl 300)
[ 4.837250] e1000 0000:00:08.0 eth0: (PCIe:33MHz:32-bit) 08:00:27:ed:cd:91
[ 4.837631] e1000 0000:00:08.0 eth0: Intel(R) PRO/1000 Network Connection
[ 5.534897] e1000 0000:00:08.0 eth1: (PCIe:33MHz:32-bit) 08:00:27:ac:bc:e0
[ 5.535411] e1000 0000:00:08.0 eth1: Intel(R) PRO/1000 Network Connection
[ 5.537398] e1000 0000:00:08.0 emp0s8: renamed from eth1
[ 5.541292] e1000 0000:00:03.0 emp0s8: renamed from eth0
[ 5.692018] raid6: sse2x4  gen() 15790 MB/s
[ 5.744027] raid6: sse2x4  xor() 10806 MB/s
[ 5.808084] raid6: sse2x2  gen() 17323 MB/s
[ 5.872437] raid6: sse2x2  xor() 11048 MB/s
[ 5.928027] raid6: sse2x1  gen() 10646 MB/s
[ 5.980381] raid6: sse2x1  xor()  8494 MB/s
[ 5.980781] raid6: using algorithm sse2x2 gen() 17323 MB/s
[ 5.981081] raid6: .... xor() 11048 MB/s, rmw enabled
[ 5.981406] raid6: using ssse3x2 recovery algorithm
[ 5.994618] xor: measuring software checksum speed
[ 6.100094] prefetch64-sse: 56690.000 MB/sec
[ 6.195957] generic_sse: 52796.000 MB/sec
[ 6.196399] xor: using function: prefetch64-sse (56690.000 MB/sec)
[ 6.208733] async_tx: api initialized (async)
[ 22.909362] random: crng init done
[ 22.909721] random: 7 urandom warning(s) missed due to ratelimiting
(initramfs)
```

Y como se puede ver es como si se hubiera interrumpido la ejecución del raid, sabemos que es el raid, pero no es muy preciso, entonces para ver más información sobre el raid hacemos **cat /proc/mdstat**, para ver el estado de los discos y del raid:

```
(initramfs) cat /proc/mdstat
Personalities : [linear] [multipath] [raid0] [raid1] [raid6] [raid5] [raid4] [raid10]
md0 : inactive sda2[0](S)
      1019904 blocks super 1.2

md1 : inactive sda3[0](S)
      14695424 blocks super 1.2

unused devices: <none>
(initramfs)
```

Y como podemos ver esta inactivos el raid, entonces para arreglar este problema, para que arranque con el disco no caído, usamos el comando **mdadm** que nos permite administrar los dispositivos. Para ello lo que vamos a hacer es arrancar el raid que esta inactivo usando el parámetro **-R**:

```
(initramfs) mdadm -R /dev/md0
[ 3597.363123] md/raid1:md0: active with 1 out of 2 mirrors
[ 3597.363412] md0: detected capacity change from 0 to 1044381696
mdadm: started array /dev/md0
(initramfs) [ 3597.380658] md0: p1

(initramfs) mdadm -R /dev/md1
[ 3621.777671] md/raid1:md1: active with 1 out of 2 mirrors
[ 3621.778005] md1: detected capacity change from 0 to 15048114176
mdadm: started array /dev/md1
(initramfs)
```

De tal forma que ya están activos:

```
(initramfs) cat /proc/mdstat
Personalities : [linear] [multipath] [raid0] [raid1] [raid6] [raid5] [raid4] [raid10]
md0 : active (auto-read-only) raid1 sda2[0]
      1019904 blocks super 1.2 [2/1] [U_]

md1 : active (auto-read-only) raid1 sda3[0]
      14695424 blocks super 1.2 [2/1] [U_]

unused devices: <none>
(initramfs)
```

Pero como podemos ver hay un disco caído, para ser exacto el segundo disco, ya que **[U\_]** nos indica los discos que están levantados o caídos. Para salir pulsamos **Ctrl + D** y el sistema continuara con su chequeo, pero esta vez con el raid activo.

```
(initramfs) done.
[Begin: Running /scripts/local-premount ... [ 3860.729391] Btrfs loaded, crc32c=crc32c-intel
]Scanning for Btrfs filesystems
[Begin: Waiting for suspend/resume device ... Begin: Running /scripts/local-block ... Please unlock disk dm_crypt-0:
```

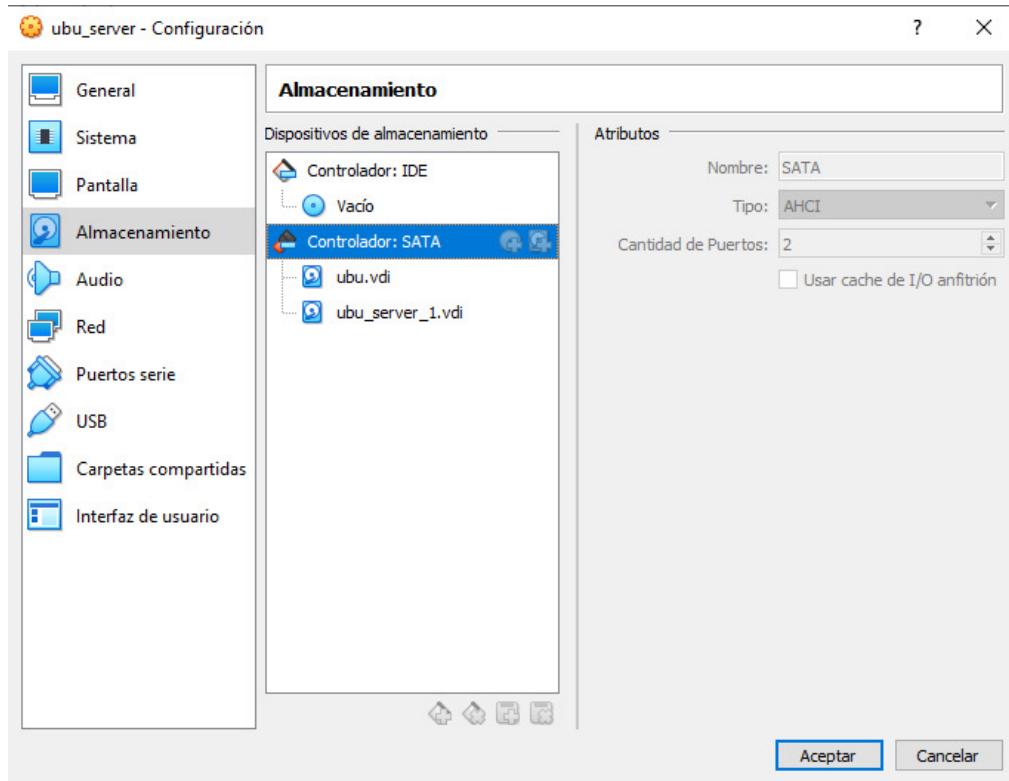
Y ya como ha detectado el raid nos pide la clave para acceder a la partición encriptada.

```
Last login: Thu Dec 17 08:48:28 UTC 2020 on tty1
ruben@ub-server:~$
```

```
ruben@ub-server:~$ lsblk
NAME      MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
loop0        7:0    0   55M  1 loop /snap/core18/1880
loop1        7:1    0 71.3M  1 loop /snap/lxd/16099
loop2        7:2    0 67.8M  1 loop /snap/lxd/18150
loop3        7:3    0 55.4M  1 loop /snap/core18/1944
loop4        7:4    0 31.1M  1 loop /snap/snapd/10492
loop5        7:5    0 29.9M  1 loop /snap/snapd/8542
sda         8:0    0   15G  0 disk 
└─sda1       8:1    0   1M  0 part 
└─sda2       8:2    0 997M  0 part 
  └─md0       9:0    0 996M  0 raid1 
    ├─md0p1    259:0  0 991M  0 part  /boot
    └─sda3       8:3    0   14G  0 part 
      └─md1       9:1    0   14G  0 raid1 
        ├─dm_crypt-0 253:0  0   14G  0 crypt 
          ├─vg0-lv--root 253:1  0   8G  0 lvm   /
          ├─vg0-lv--swap 253:2  0   2G  0 lvm   [SWAP]
          └─vg0-lv--home 253:3  0   4G  0 lvm   /home
sr0        11:0   1 1024M  0 rom
```

Y listo ya podemos acceder al sistema operativo. Normalmente cuando ocurre esto se sustituye el disco y se configura para el raid, por si se cayera el disco actual activo tengamos uno de respaldo.

Para ello nos vamos a la máquina virtual y añadimos un disco virtual nuevo, como el raid que usamos es RAID 1 el disco duro tiene que tener el mismo tamaño:



Vemos que el disco lo detecta el sistema operativo con **lsblk**:

```
ruben@ub-server:~$ lsblk
NAME      MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
loop0          7:0   0 55M  1 loop /snap/core18/1880
loop1          7:1   0 55.4M 1 loop /snap/core18/1944
loop2          7:2   0 31.1M 1 loop /snap/snapd/10492
loop3          7:3   0 67.8M 1 loop /snap/1xd/18150
loop4          7:4   0 71.3M 1 loop /snap/1xd/16099
loop5          7:5   0 29.9M 1 loop /snap/snapd/8542
sda           8:0   0 15G  0 disk 
└─sda1         8:1   0  1M  0 part 
└─sda2         8:2   0 997M 0 part 
  └─md0          9:0   0 996M 0 raid1 
    ├─md0p1     259:0  0 991M 0 part  /boot
    └─sda3         8:3   0 14G  0 part 
      └─md1          9:1   0 14G  0 raid1 
        ├─dm_crypt-0 253:0  0 14G  0 crypt 
          ├─vg0-lv--root 253:1  0  8G  0 lvm   /
          ├─vg0-lv--swap 253:2  0  2G  0 lvm   [SWAP]
          └─vg0-lv--home 253:3  0  4G  0 lvm   /home
sdb           8:16  0 10G  0 disk 
sr0          11:0   1 1024M 0 rom 

ruben@ub-server:~$
```

Ahora tenemos que particionar el disco en 3 como en el disco **sda**, con los mismos tamaños que indican, usando **sudo fdisk /dev/sdb**

```
[ruben@ub-server:~$ sudo fdisk /dev/sdb
[sudo] password for ruben:

Welcome to fdisk (util-linux 2.34).
Changes will remain in memory only, until you decide to write them.
Be careful before using the write command.

Device does not contain a recognized partition table.
Created a new DOS disklabel with disk identifier 0x455560ba.

Command (m for help):
```

Para crear las particiones pulsamos:

- **n**: para crear una partición nueva al disco seleccionado.
- **p**: para indicar que es primaria
- **Enter**, para que elija por defecto el numero de la partición
- **Enter**, para que elija por defecto el primer sector
- Y en vez de darle el ultimo sector, le damos el tamaño que va a tener esa partición **+1M** (le indico que va a ocupar 1 Mbyte esa partición)

Creación de las particiones:

```
Command (m for help): n
Partition type
  p  primary (0 primary, 0 extended, 4 free)
  e  extended (container for logical partitions)
Select (default p): p
Partition number (1-4, default 1):
First sector (2048-31457279, default 2048):
Last sector, +/-sectors or +/-size{K,M,G,T,P} (2048-31457279, default 31457279): +1M

Created a new partition 1 of type 'Linux' and of size 1 MiB.

Command (m for help): n
Partition type
  p  primary (1 primary, 0 extended, 3 free)
  e  extended (container for logical partitions)
Select (default p): p
Partition number (2-4, default 2):
First sector (4096-31457279, default 4096):
Last sector, +/-sectors or +/-size{K,M,G,T,P} (4096-31457279, default 31457279): +997M

Created a new partition 2 of type 'Linux' and of size 997 MiB.
```

Y para indicar que la última partición ocupe el resto le ponemos **-0**:

```
Command (m for help): n
Partition type
  p  primary (2 primary, 0 extended, 2 free)
  e  extended (container for logical partitions)
Select (default p): p
Partition number (3,4, default 3):
First sector (2045952-31457279, default 2045952):
Last sector, +/-sectors or +/-size{K,M,G,T,P} (2045952-31457279, default 31457279): -0

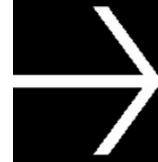
Created a new partition 3 of type 'Linux' and of size 14 GiB.

Command (m for help): w
The partition table has been altered.
Calling ioctl() to re-read partition table.
Syncing disks.

ruben@ub-server:~$ _
```

# Cuanto más difícil sea el examen, más vas a disfrutar celebrándolo.

Uber



Para guardar los cambios hechos le damos **w** y **Enter**, para guardar y salir.

```
ruben@ub-server:~$ lsblk
NAME      MAJ:MIN RM  SIZE RO TYPE  MOUNTPOINT
loop0      7:0    0   55M  1 loop  /snap/core18/1880
loop1      7:1    0 55.4M  1 loop  /snap/core18/1944
loop2      7:2    0 31.1M  1 loop  /snap/snapd/10492
loop3      7:3    0 67.8M  1 loop  /snap/lxd/18150
loop4      7:4    0 71.3M  1 loop  /snap/lxd/16099
loop5      7:5    0 29.9M  1 loop  /snap/snapd/8542
sda       8:0    0   15G  0 disk
└─sda1     8:1    0   1M  0 part
  └─sda2     8:2    0 997M  0 part
    └─md0      9:0    0 996M  0 raid1
      └─md0p1    259:0  0 991M  0 part  /boot
        └─sda3     8:3    0   14G  0 part
          └─md1      9:1    0   14G  0 raid1
            └─dm_crypt-0 253:0  0   14G  0 crypt
              ├─vg0-lv--root 253:1  0   8G  0 lvm   /
              ├─vg0-lv--swap 253:2  0   2G  0 lvm   [SWAP]
              └─vg0-lv--home 253:3  0   4G  0 lvm   /home
sdb       8:16   0   10G  0 disk
└─sdb1     8:17   0   1M  0 part
  └─sdb2     8:18   0 997M  0 part
    └─sdb3     8:19   0   9G  0 part
sr0      11:0   1 1024M 0 rom
ruben@ub-server:~$
```

Como podemos ver ya están las 3 particiones hechas, ahora le añadimos a cada multidevices un disco, en mi caso tenemos 2 multidevice:

- **md0**: que está asignado a la partición **sda2**, pues se lo asignamos también al **sdb2**
- **md1**: que está asignado a la partición **sda3**, pues se lo asignamos también al **sdb3**

Para ello usamos el comando:

```
mdadm /dev/md0 --add /dev/sdb2 y   mdadm /dev/md1 --add /dev/sdb3
```

```
ruben@ub-server:~$ sudo mdadm /dev/md0 --add /dev/sdb2
mdadm: added /dev/sdb2
ruben@ub-server:~$ sudo mdadm /dev/md1 --add /dev/sdb3
mdadm: added /dev/sdb3
ruben@ub-server:~$ lsblk
NAME      MAJ:MIN RM  SIZE RO TYPE  MOUNTPOINT
loop0      7:0    0   55M  1 loop  /snap/core18/1880
loop1      7:1    0 55.4M  1 loop  /snap/core18/1944
loop2      7:2    0 31.1M  1 loop  /snap/snapd/10492
loop3      7:3    0 67.8M  1 loop  /snap/lxd/18150
loop4      7:4    0 71.3M  1 loop  /snap/lxd/16099
loop5      7:5    0 29.9M  1 loop  /snap/snapd/8542
sda       8:0    0   15G  0 disk
└─sda1     8:1    0   1M  0 part
  └─sda2     8:2    0 997M  0 part
    └─md0      9:0    0 996M  0 raid1
      └─md0p1    259:0  0 991M  0 part  /boot
        └─sda3     8:3    0   14G  0 part
          └─md1      9:1    0   14G  0 raid1
            └─dm_crypt-0 253:0  0   14G  0 crypt
              ├─vg0-lv--root 253:1  0   8G  0 lvm   /
              ├─vg0-lv--swap 253:2  0   2G  0 lvm   [SWAP]
              └─vg0-lv--home 253:3  0   4G  0 lvm   /home
sdb       8:16   0   15G  0 disk
└─sdb1     8:17   0   1M  0 part
  └─sdb2     8:18   0 997M  0 part
    └─sdb3     8:19   0   9G  0 part
sr0      11:0   1 1024M 0 rom
ruben@ub-server:~$
```

Como se puede ver ya están añadidos los multidevices y no solo eso sino también que el segundo disco que nos detecta es el nuevo metido, en el que ya está cargado todo el contenido del otro disco.

```
ruben@ub-server:~$ cat /proc/mdstat
Personalities : [linear] [multipath] [raid0] [raid1] [raid5] [raid4] [raid10]
md0 : active raid1 sdb2[2] sda2[0]
      1019904 blocks super 1.2 [2/2] [UU]

md1 : active raid1 sdb3[2] sda3[0]
      14695424 blocks super 1.2 [2/2] [UU]

unused devices: <none>
ruben@ub-server:~$
```

Pero nos faltaría el grub en el segundo disco que lo haremos en la partición creada de 1Mbyte en mi caso **sdb1**, para ello usamos el comando **sudo grub-install /dev/sdb**:

```
ruben@ub-server:~$ sudo grub-install /dev/sdb
Installing for i386-pc platform.
Installation finished. No error reported.
ruben@ub-server:~$ _
```

Y ya funcionaria igual que el disco caído.