

ISEP4-FULL-DPR.pdf



danielsp10



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



UNIVERSIDAD
DE GRANADA

INGENIERÍA DE SERVIDORES

EJERCICIO ENTREGABLE:
PRÁCTICA 4 – BENCHMARK Y JMETER

Doble Grado en Ingeniería Informática y Matemáticas (UGR)

Pérez Ruiz, Daniel

Uber

Dirección postal Granada | Tfno. 958 00 00 00 | Fax: 958 00 00 00 | correo@ugr.es | www.ugr.es

Contenido

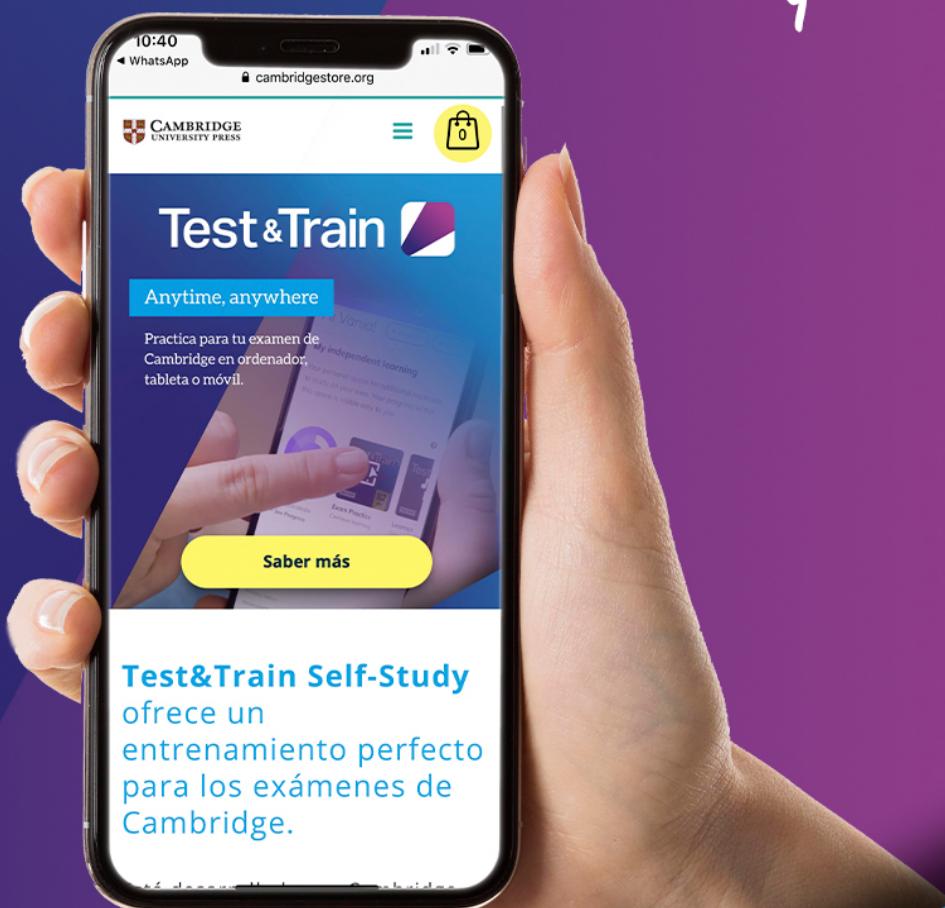
INGENIERÍA DE SERVIDORES: GRUPO 2 - Práctica 4.....	3
1.0 Enunciado del Ejercicio.....	3
1.1 Primeros Pasos : Instalación de Phoronix en CentOS	3
1.2 Primeros Pasos : Instalación de Phoronix en Ubuntu Server.....	5
1.3 Elección de Benchmarks para ejecutar en los S.O.....	6
<i>Ejecución del Benchmark de RAMspeed en CentOS:.....</i>	7
<i>Ejecución del Benchmark de RAMspeed en Ubuntu Server:.....</i>	11
<i>Tabla comparativa de Resultados Benchmark 1:.....</i>	14
<i>Ejecución del Benchmark de Systemd-Boot-Total en CentOS:.....</i>	15
<i>Ejecución del Benchmark de Systemd-Boot-Total en Ubuntu Server:.....</i>	17
<i>Tabla comparativa de Resultados Benchmark 2:.....</i>	18
2.0 Enunciado del Ejercicio.....	19
2.1 Primeros pasos: Instalación de JMeter en Máquina Host.....	19
2.2 Primeros pasos: Instalación de Docker en Ubuntu Server.....	20
2.3 Levantar el servicio iseP4Jmeter	23
2.4 Configuración del test con JMeter.....	25
2.5 Probar rendimiento del test.....	32

Test&Train

Practica online tu examen de inglés con
Test & Train de Cambridge:
www.testandtrain.es

-5%
DTO.

Código:
WUOT&T



y consigue ya tu
↓

**B2
FIRST**

**C1
ADVANCED**

#CambridgeTestandTrain

INGENIERÍA DE SERVIDORES: GRUPO 2 - Práctica 4

Ejercicio 1: Ejecución de Benchmarks.

1.0 Enunciado del Ejercicio

Una vez que haya indagado sobre los benchmarks disponibles, seleccione como mínimo dos de ellos y proceda a ejecutarlos en Ubuntu y CentOS. Comente las diferencias

1.1 Primeros Pasos : Instalación de Phoronix en CentOS

Phoronix es una plataforma que permite ejecutar un conjunto de benchmarks bajo la agrupación openbenchmarking.org. La aplicación puede instalarse a través de los gestores de paquetes ya vistos en los guiones anteriores. En primer lugar, obtendremos el programa del repositorio oficial y lo instalaremos en CentOS:

```
[root@localhost perud]# wget https://phoronix-test-suite.com/releases/phoronix-test-suite-10.4.0.tar.gz
--2021-05-19 11:19:05-- https://phoronix-test-suite.com/releases/phoronix-test-suite-10.4.0.tar.gz
Resolviendo phoronix-test-suite.com (phoronix-test-suite.com)... 192.211.48.82
Conectando con phoronix-test-suite.com (phoronix-test-suite.com)[192.211.48.82]:443... conectado.
Petición HTTP enviada, esperando respuesta... 200 OK
Longitud: 2196031 (2,1M) [application/x-gzip]
Grabando a: "phoronix-test-suite-10.4.0.tar.gz"

phoronix-test-suite-10.4 100%[=====] 2,09M 1,81MB/s en 1,2s
2021-05-19 11:19:07 (1,81 MB/s) - "phoronix-test-suite-10.4.0.tar.gz" guardado [2196031/2196031]
```

Tras descomprimir el fichero, nos movemos a la carpeta y ejecutamos el script de instalación.

```
[root@localhost perud]# tar xvfz phoronix-test-suite-10.4.0.tar.gz
```

```
[root@localhost perud]# cd phoronix-test-suite/
[root@localhost phoronix-test-suite]# ./install-sh
```

```
[root@localhost phoronix-test-suite]# ls
AUTHORS deploy install_macos phoronix-test-suite README.md
ChangeLog documentation install-sh phoronix-test-suite.bat release-highlights.md
COPYING install.bat ob-cache pts-core
[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/
```

Mostramos la información de nuestro sistema con Phoronix ejecutando la orden:

```
[perud@localhost ~]$ phoronix-test-suite system-info >> infoSystem.txt$
```

Vemos el contenido de este fichero de datos volcado:

```
Phoronix Test Suite v10.4.0
System Information

PROCESSOR: Intel Core i7-7500U
Core Count: 1
Extensions: SSE 4.2 + AUXZ + AUX + RDRAND + FSGSBASE
Cache Size: 4896 MB
Core Family: Kaby/Coffee/Whiskey Lake

GRAPHICS: VMware SVGA II
Screen: 2048x2048

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

MEMORY: 2048MB

DISK: 2 x 9GB VBOX HDD
File-System: xfs
Mount Options: attr2 inode64 noquota relatime rw seclabel
Disk Scheduler: MQ-DEADLINE
Disk Details: Block Size: 4096

OPERATING SYSTEM: CentOS Linux 8
Kernel: 4.18.0-193.el8.x86_64 (x86_64)

System Layer: Oracle VMware
Security: SELinux
```

```
* itlb_multihit: KUM: Vulnerable
+ l1tf: 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
```

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

Uber



Daniel Pérez Ruiz - perud

GRUPO 2 - PRÁCTICA 4

Pág: 5

1.2 Primeros Pasos : Instalación de Phoronix en Ubuntu Server

Ahora hacemos los mismos pasos que hemos hecho en CentOS para hacer lo mismo con Ubuntu Server. Primero, descargamos el paquete de instalación:

```
root@perud:/home/perud# wget http://phoronix-test-suite.com/releases/repo/pts.debian/files/phoronix-test-suite_10.4.0_all.deb
--2021-05-19 15:38:30-- http://phoronix-test-suite.com/releases/repo/pts.debian/files/phoronix-test-suite_10.4.0_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: 1164944 (1.1M) [application/x-debian-package]
Saving to: 'phoronix-test-suite_10.4.0_all.deb'

phoronix-test-suite_10.4 100%[=====] 1.11M 1.35MB/s in 0.8s
2021-05-19 15:38:31 (1.35 MB/s) - 'phoronix-test-suite_10.4.0_all.deb' saved [1164944/1164944]
```

```
root@perud:/home/perud# dpkg -i phoronix-test-suite_10.4.0_all.deb
Selecting previously unselected package phoronix-test-suite.
(Reading database ... 74737 files and directories currently installed.)
Preparing to unpack phoronix-test-suite_10.4.0_all.deb ...
Unpacking phoronix-test-suite (10.4.0) ...
Setting up phoronix-test-suite (10.4.0) ...
Processing triggers for mime-support (3.64ubuntu1) ...
Processing triggers for man-db (2.9.1-1) ...
```

Mostramos la información de nuestro sistema con la orden específica para ello:

```
Phoronix Test Suite v10.4.0
System Information

PROCESSOR: Intel Core i7-7500U
Core Count: 1
Extensions: SSE 4.2 + AVX2 + AVX + RDRAND + FSGSBASE
Cache Size: 4 MB
Core Family: Kaby/Coffee/Whiskey Lake

GRAPHICS: VMware SVGA II
Screen: 2048x2048

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

MEMORY: 4096MB

DISK:
File-System: 2 x 11GB VBOX HDD
Mount Options: ext4
Disk Scheduler: mq-deadline
Disk Details: Block Size: 4096

OPERATING SYSTEM: Ubuntu 20.04
Kernel: 5.4.0-67-generic (x86_64)

System Layer: Oracle VMware
Security: intel_multihit: KVM: Vulnerable
```

```
OPERATING SYSTEM: Ubuntu 20.04
Kernel: 5.4.0-67-generic (x86_64)

System Layer: Oracle VMware

Security:
  + itlb_multihit: KVM: Vulnerable
  + i1tf: 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
  + srbds: Unknown: Dependent on hypervisor status
  + tsx_async_abort: Not affected
```

1.3 Elección de Benchmarks para ejecutar en los S.O

Para la realización de esta parte del ejercicio, voy a utilizar los siguientes benchmarks:

- **RAMSpeed:** [Prueba de Memoria RAM]. Este benchmark comprueba el rendimiento de la memoria RAM de un sistema.
 - Ejecución del Benchmark: [phoronix-test-suite benchmark ramspeed](#)
- **Systemd-Boot-Total:** [Prueba de Procesador] Utiliza la herramienta systemd-analyze para detectar el tiempo total empleado en el boot.
 - Ejecución del Benchmark: [phoronix-test-suite benchmark systemd-boot-total](#)

Ejecución del Benchmark de RAMspeed en CentOS:

Primero se necesitan instalar ciertas dependencias necesarias para la ejecución de este benchmark.

```
[perud@localhost ~] $ phoronix-test-suite benchmark ramspeed

Evaluating External Test Dependencies ......

The following dependencies are needed and will be installed:

- gcc
- gcc-c++
- make
- autoconf
- automake
- patch
- expat-devel

This process may take several minutes.
```

Tras haber instalado todas las dependencias y el benchmark, procedemos a ejecutar todos los test que nos ofrece para coma flotante.

```
Phoronix Test Suite v10.4.0

Installed: pts/ramspeed-1.4.3

RAMspeed SMP 3.5.0:
pts/ramspeed-1.4.3
Memory Test Configuration
1: Copy
2: Scale
3: Add
4: Triad
5: Average
6: Test All Options
** Multiple items can be selected, delimit by a comma. **
Type: 6

1: Integer
2: Floating Point
3: Test All Options
** Multiple items can be selected, delimit by a comma. **
Benchmark: 2_
```

CentOS [Prueba 1] : Sumar números en coma flotante.

```
RAMspeed SMP 3.5.0:
pts/ramspeed-1.4.3 [Type: Add - Benchmark: Floating Point]
Test 1 of 5
Estimated Trial Run Count: 3
Estimated Test Run-Time: 10 Minutes
Estimated Time To Completion: 47 Minutes [07:15 EDT]
Started Run 1 @ 06:29:24
Started Run 2 @ 06:32:35
Started Run 3 @ 06:35:40

Type: Add - Benchmark: Floating Point:
10073.11
10275.95
10268.24

Average: 10205.77 MB/s
Deviation: 1.13%

Comparison to 3,341 OpenBenchmarking.org samples since 26 February 2011; median result: 17523. Box plot of samples:
[-----*#####
^ This Result (25th Percentile): 10206
8 x 16384 MB DDR4-2133MHz: 18862 ^
16 x 32 GB DDR4-2400MT: 34784 ^
8 x 8192 MB DDR4-2400MHz Samsung: 31438 ^
12 x 16384 MB DDR4-2933MT: 30651 ^
2 x 8192 MB DDR4-3200MT: 28691 ^
```

CentOS [Prueba 2] : Copiar números en coma flotante.

```
RAMspeed SMP 3.5.0:
pts/ramspeed-1.4.3 [Type: Copy - Benchmark: Floating Point]
Test 2 of 5
Estimated Trial Run Count: 3
Estimated Test Run-Time: 10 Minutes
Estimated Time To Completion: 37 Minutes [07:15 EDT]
Started Run 1 @ 06:38:53
Started Run 2 @ 06:42:07
Started Run 3 @ 06:45:19

Type: Copy - Benchmark: Floating Point:
8871.78
9101.62
8717.27

Average: 8896.89 MB/s
Deviation: 2.17%

Comparison to 3,439 OpenBenchmarking.org samples since 26 February 2011; median result: 14708. Box plot of samples:
[-----*#####
^ This Result (23rd Percentile): 8897
1 x 7908 MB RAM: 18672 ^
12 x 32 GB DDR4-3200MT: 30779 ^
12 x 16384 MB DDR4-2933MT: 29346 ^
6 x 32 GB DDR4-2933MT: 27909 ^
4 x 16384 MB DDR4-3200MT: 27080 ^
```

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

Uber



Daniel Pérez Ruiz - perud

GRUPO 2 - PRÁCTICA 4

Pág: 9

CentOS [Prueba 3] : Multiplicación escalar por número en coma flotante.

```
RAMspeed SMP 3.5.0:
  pts/ramspeed-1.4.3 [Type: Scale - Benchmark: Floating Point]
  Test 1 of 1
  Estimated Trial Run Count: 3
  Estimated Time To Completion: 10 Minutes [07:24 EDT]
    Started Run 1 @ 07:15:05
    Started Run 2 @ 07:18:05
    Started Run 3 @ 07:21:06

  Type: Scale - Benchmark: Floating Point:
    8932.38
    8973.99
    9021.49

  Average: 8975.95 MB/s
  Deviation: 0.50%

  Comparison to 3,324 OpenBenchmarking.org samples since 26 February 2011: median result: 14812. Box plot of samples:
  [-----*#####*!#####-----*---|*-----*-----]
      ^ This Result (23rd Percentile): 8976
      12 x 8192 MB DDR3-1600MHz 689911-071: 19831 ^
      4 x 8192 MB DDR4-3200MT: 32096 ^
      12 x 16384 MB DDR4-2933MT: 29571 ^
      8 x 16384 MB DDR4-2666MT: 27673 ^
      4 x 16384 MB DDR4-3200MT: 27151 ^
```

CentOS [Prueba 4] : Operación TRIAD. Suma + producto por escalar en coma flotante. $[c = a+q^*b]$.

```
RAMspeed SMP 3.5.0:
  pts/ramspeed-1.4.3 [Type: Triad - Benchmark: Floating Point]
  Test 1 of 2
  Estimated Trial Run Count: 3
  Estimated Test Run-Time: 10 Minutes
  Estimated Time To Completion: 19 Minutes [07:47 EDT]
    Started Run 1 @ 07:29:23
    Started Run 2 @ 07:32:36
    Started Run 3 @ 07:35:51

  Type: Triad - Benchmark: Floating Point:
    9008.73
    9254.89
    9285.55

  Average: 9183.06 MB/s
  Deviation: 1.65%

  Comparison to 3,362 OpenBenchmarking.org samples since 26 February 2011: median result: 17283. Box plot of samples:
  [-----*#####*!#####-----*---|*-----*-----]
      ^ This Result (22nd Percentile): 9183
      16 x 4896 MB DDR3-1600MHz Samsung: 20979 ^
      4 x 8192 MB DDR4-3200MT: 34592 ^
      16 x 32 GB DDR4-2400MT: 31162 ^
      6 x 16384 MB DDR4-2666MHz Micron 18ASF2G72PDZ-2G6D1: 28196 ^
      6 x 16384 MB DDR4-2933MT: 27390 ^
```

CentOS [Prueba 5] : Media en coma flotante.

```
RAMspeed SMP 3.5.0:
pts/ramspeed-1.4.3 [Type: Average - Benchmark: Floating Point]
Test 1 of 1
Estimated Trial Run Count: 3
Estimated Time To Completion: 10 Minutes [08:07 EDT]
    Started Run 1 @ 07:58:28
    Started Run 2 @ 08:01:27
    Started Run 3 @ 08:04:26

Type: Average - Benchmark: Floating Point:
  9902.8
  9857.07
  9886.11

Average: 9881.99 MB/s
Deviation: 0.23%

Comparison to 4,512 OpenBenchmarking.org samples since 26 February 2011; median result: 13637. Box plot of samples:
[ *-----#####
^ This Result (35th Percentile): 9882
  ^ 1 x 512: 1887      2 x 8192 MB DDR4-2133MT: 20003 ^ 4 x 4096 MB DDR4-3200MT: 31600 ^
  16 x 32 GB DDR4-2400MT: 29838 ^
  4 x 16384 MB DDR4-3200MT: 29188 ^
  6 x 32 GB DDR4-2933MT: 28220 ^
```

Comparativa de datos para CentOS:

Sistema	CentOS 8	
Cantidad RAM	2GB - 2046MB	
Prueba 1: Add	10205.77 MB/s	Percentil 25
Prueba 2: Copy	8896.89 MB/s	Percentil 23
Prueba 3: Scale	8975.95 MB/s	Percentil 23
Prueba 4: Triad	9183.06 MB/s	Percentil 22
Prueba 5: Average	9881.99 MB/s	Percentil 35

Ejecución del Benchmark de RAMspeed en Ubuntu Server:

Al igual que en CentOS, instalamos las dependencias del benchmark, y utilizamos los mismos test: **todos, sobre coma flotante.**

```
perud@perud:~$ phoronix-test-suite benchmark ramspeed

    Evaluating External Test Dependencies ......

The following dependencies are needed and will be installed:

- build-essential
- autoconf
- mesa-utils
- vulkan-tools
- unzip
- apt-file

This process may take several minutes.
```

Ubuntu Server [Prueba 1] : Sumar números en coma flotante.

```
RAMspeed SMP 3.5.0:
  pts/ramspeed-1.4.3 [Type: Add - Benchmark: Floating Point]
  Test 1 of 1
  Estimated Trial Run Count:   3
  Estimated Time To Completion: 10 Minutes [13:43 UTC]
    Started Run 1 @ 13:34:14
    Started Run 2 @ 13:37:16
    Started Run 3 @ 13:40:16

  Type: Add - Benchmark: Floating Point:
    10954.25
    10951.41
    10990.62

  Average: 10965.43 MB/s
  Deviation: 0.20%

  Comparison to 3,341 OpenBenchmarking.org samples since 26 February 2011; median result: 17523. Box plot of samples:
  [-----##*#####!###*#####-----*---| * ]
    ^ This Result (27th Percentile): 10965
      8 x 16384 MB DDR4-2133MHz: 18862 ^
      16 x 32 GB DDR4-2400MT: 34784 ^
      8 x 8192 MB DDR4-2400MHz Samsung: 31438 ^
      12 x 16384 MB DDR4-2933MT: 30651 ^
      2 x 8192 MB DDR4-3200MT: 28691 ^
```

Ubuntu Server [Prueba 2] : Copiar n meros en coma flotante.

```
RAMspeed SMP 3.5.0:  
pts/ramspeed-1.4.3 [Type: Copy - Benchmark: Floating Point]  
Test 1 of 1  
Estimated Trial Run Count: 3  
Estimated Time To Completion: 9 Minutes [13:53 UTC]  
Started Run 1 @ 13:44:33  
Started Run 2 @ 13:47:36  
Started Run 3 @ 13:50:36  
  
Type: Copy - Benchmark: Floating Point:  
9507.65  
9718.18  
9702.4  
  
Average: 9642.74 MB/S  
Deviation: 1.22%  
  
Comparison to 3,439 OpenBenchmarking.org samples since 26 February 2011; median result: 14708. B  
ox plot of samples:  
[-----*# ##### !#####-*-----*-----*-----|* ]  
^ This Result (26th Percentile): 9643  
1 x 7908 MB RAM: 18672 ^ 12 x 32 GB DDR4-3200MT: 30779 ^  
12 x 16384 MB DDR4-2933MT: 29346 ^  
6 x 32 GB DDR4-2933MT: 27909 ^  
4 x 16384 MB DDR4-3200MT: 27080 ^
```

Ubuntu Server [Prueba 3] : Escalar n meros en coma flotante.

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

Uber



Daniel Pérez Ruiz - perud

GRUPO 2 - PRÁCTICA 4

Pág: 13

*Ubuntu Server [Prueba 4] : Operación TRIAD. Suma + producto por escalar en coma flotante. [c = a+q*b].*

```
RAMspeed SMP 3.5.0:  
  pts/ramspeed-1.4.3 [Type: Triad - Benchmark: Floating Point]  
  Test 1 of 1  
  Estimated Trial Run Count:  3  
  Estimated Time To Completion: 9 Minutes [14:22 UTC]  
    Started Run 1 @ 14:13:42  
    Started Run 2 @ 14:16:43  
    Started Run 3 @ 14:19:43  
  
  Type: Triad - Benchmark: Floating Point:  
    10016.91  
    10072.96  
    10083.37  
  
  Average: 10057.55 MB/s  
  Deviation: 0.35%  
  
  Comparison to 3,362 OpenBenchmarking.org samples since 26 February 2011; median result: 17283. Box plot of samples:  
  [-----*-----#*****#!#####-----*----| *      *      ]  
          ^ This Result (25th Percentile): 10058  
    16 x 4096 MB DDR3-1600MHz Samsung: 20979 ^ 4 x 8192 MB DDR4-3200MT: 34592 ^  
    16 x 32 GB DDR4-2400MT: 31162 ^  
    6 x 16384 MB DDR4-2666MHz Micron 18ASF2G72PD2-2G6D1: 28196 ^  
    6 x 16384 MB DDR4-2933MT: 27390 ^
```

Ubuntu Server [Prueba 5] : Media en coma flotante.

```
RAMspeed SMP 3.5.0:  
  pts/ramspeed-1.4.3 [Type: Average - Benchmark: Floating Point]  
  Test 1 of 1  
  Estimated Trial Run Count:  3  
  Estimated Time To Completion: 9 Minutes [14:33 UTC]  
    Started Run 1 @ 14:24:45  
    Started Run 2 @ 14:27:50  
    Started Run 3 @ 14:30:51  
  
  Type: Average - Benchmark: Floating Point:  
    9563.85  
    9775.18  
    9816.93  
  
  Average: 9718.65 MB/s  
  Deviation: 1.40%  
  
  Comparison to 4,512 OpenBenchmarking.org samples since 26 February 2011; median result: 13637. Box plot of samples:  
  [*|-----#*****#!#####-----*----|*      *      ]  
          ^ This Result (34th Percentile): 9719  
    ^ 1 x 512: 1887    2 x 8192 MB DDR4-2133MT: 20003 ^ 4 x 4096 MB DDR4-3200MT: 31600 ^  
    16 x 32 GB DDR4-2400MT: 29838 ^  
    4 x 16384 MB DDR4-3200MT: 29188 ^  
    6 x 32 GB DDR4-2933MT: 28220 ^
```

Comparativa de datos para Ubuntu Server:

Sistema	Ubuntu Server 20.04	
Cantidad RAM	4GB - 4096MB	
Prueba 1: Add	10965.43 MB/s	Percentil 27
Prueba 2: Copy	9642.74 MB/s	Percentil 26
Prueba 3: Scale	8359.04 MB/s	Percentil 21
Prueba 4: Triad	10057.55 MB/s	Percentil 25
Prueba 5: Average	9718.65 MB/s	Percentil 34

Tabla comparativa de Resultados Benchmark 1:

Sistema	CentOS 8	Ubuntu Server 20.04
Cantidad RAM	2GB - 2048MB	4GB - 4096MB
Prueba 1: Add	10205.77 MB/s	10965.43 MB/s
Prueba 2: Copy	8896.89 MB/s	9642.74 MB/s
Prueba 3: Scale	8975.95 MB/s	8359.04 MB/s
Prueba 4: Triad	9183.06 MB/s	10057.55 MB/s
Prueba 5: Average	9881.99 MB/s	9718.65 MB/s

Los resultados son ligeramente mejores en la máquina de Ubuntu que en la máquina con CentOS, debido probablemente a que se dispone de una cantidad de memoria mayor para operaciones en la de Ubuntu.

Ejecución del Benchmark de Systemd-Boot-Total en CentOS:

No es necesario instalar dependencias a la hora de ejecutar este benchmark puesto que utiliza la herramienta `systemd-analyze` ya integrada. No tenemos disponible la comparativa con *Loader* ni con *Firmware*.

```
perud@localhost ~]$ phoronix-test-suite benchmark systemd-boot-total

Evaluating External Test Dependencies . . . . .  
  
Phoronix Test Suite v10.4.0  
  
To Install: pts/systemd-boot-total-1.0.6  
  
Determining File Requirements . . . . .  
Searching Download Caches . . . . .  
  
1 Test To Install  
 1 Second Estimated Install Time  
  
pts/systemd-boot-total-1.0.6:  
  Test Installation 1 of 1  
  Installing Test @ 08:51:56  
  
Systemd Total Boot Time:  
pts/systemd-boot-total-1.0.6  
System Test Configuration  
 1: Total  
 2: Userspace  
 3: Kernel  
 4: Loader  
 5: Firmware  
 6: Test All Options  
 ** Multiple items can be selected, delimit by a comma. **  
Test:
```

CentOS [Prueba 1] : Comprobación del tiempo total requerido.

CentOS [Prueba 2] : Comprobación del tiempo boot para el kernel.

```
Systemd Total Boot Time:
pts/systemd-boot-total-1.0.6 [Test: Kernel]
Test 2 of 5
Estimated Trial Run Count: 1
Started Run 1 @ 08:52:57

Test: Kernel:
982

Average: 982 ms

Comparison to 1,095 OpenBenchmarking.org samples since 16 August 2016; median result: 2217. Box
plot of samples:
[-----*-----#####
This Result (74th Percentile): 982 ^
Intel Xeon E5-1680 v3: 6617 ^ Intel Xeon E3-1260L v5: 700 ^
Intel Core i3-7100: 1404 ^
Intel Core i9-9900KS: 1941 ^
AMD Ryzen 3 1200: 2536 ^
```

CentOS [Prueba 3] : Comprobación del tiempo boot para el espacio de usuario.

```
Systemd Total Boot Time:
pts/systemd-boot-total-1.0.6 [Test: Userspace]
Test 5 of 5
Estimated Trial Run Count: 1
Started Run 1 @ 08:53:28

Test: Userspace:
34388

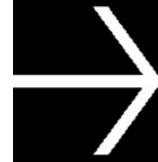
Average: 34388 ms

Comparison to 609 OpenBenchmarking.org samples since 16 August 2016; median result: 19882. Box p
lot of samples:
[-----*-----#####
This Result (8th Percentile): 34388 ^
Intel Core i5-8400: 21572 ^
Intel Xeon E3-1235L v5: 22313 ^
Intel Core i5-6500: 24383 ^
Intel Core i5-9400F: 26324 ^
```

Sistema	CentOS 8	
Procesador	Intel Core I7-5500U	
Prueba 1: Total	Media: 35370 ms	Percentil 40
Prueba 2: Kernel	Media: 982 ms	Percentil 74
Prueba 3: UserSpace	Media: 34388 ms	Percentil 8

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

Uber



Daniel Pérez Ruiz - perud

GRUPO 2 - PRÁCTICA 4

Pág: 17

Ejecución del Benchmark de Systemd-Boot-Total en Ubuntu Server:

```
perud@perud:~$ phoronix-test-suite benchmark systemd-boot-total

Evaluating External Test Dependencies ......

Phoronix Test Suite v10.4.0

Installed: pts/systemd-boot-total-1.0.6

Systemd Total Boot Time:
pts/systemd-boot-total-1.0.6
System Test Configuration
 1: Total
 2: Userspace
 3: Kernel
 4: Loader
 5: Firmware
 6: Test All Options
 * Multiple items can be selected, delimit by a comma. **
Test: _
```

Ubuntu Server [Prueba 1] : Comprobación del tiempo total requerido.

```
Systemd Total Boot Time:
pts/systemd-boot-total-1.0.6 [Test: Total]
Test 1 of 1
Estimated Trial Run Count: 1
Started Run 1 @ 13:06:24

Test: Total:
99100

Average: 99100 ms

Comparison to 1,156 OpenBenchmarking.org samples since 11 August 2016; median result: 28061. Box
plot of samples:
[-----*-----#*#*#*#*]
                                         This Result (19th Percentile): 99100 ^
                                         Intel Core i5-2400: 11952 ^
                                         Intel Core i5-9400F: 29380 ^
                                         Intel Core i5-8400: 42965 ^
```

Ubuntu Server [Prueba 2] : Comprobación del tiempo boot para el kernel.

```
Systemd Total Boot Time:
pts/systemd-boot-total-1.0.6 [Test: Kernel]
Test 2 of 5
Estimated Trial Run Count: 1
Estimated Time To Completion: 1 Minute [13:11 UTC]
Started Run 1 @ 13:11:18

Test: Kernel:
51474

Average: 51474 ms
```

Ubuntu Server [Prueba 3] : Comprobación del tiempo boot para el espacio de usuario.

```
Systemd Total Boot Time:  
  pts/systemd-boot-total-1.0.6 [Test: Userspace]  
Test 1 of 1  
Estimated Trial Run Count: 1  
    Started Run 1 @ 13:09:00  
  
Test: Userspace:  
  47626  
  
Average: 47626 ms  
  
Comparison to 609 OpenBenchmarking.org samples since 17 August 2016; median result: 19882. Box p  
lot of samples:  
[-----*---*#*#*#!#####-----|]  
^ This Result (2nd Percentile): 47626  
          Intel Core i5-8400: 21572 ^  
          Intel Xeon E3-1235L v5: 22313 ^  
          Intel Core i5-6500: 24383 ^  
          Intel Core i5-9400F: 26324 ^
```

Sistema	Ubuntu Server	
Procesador	Intel Core I7-5500U	
Prueba 1: Total	Media: 99100 ms	Percentil 19
Prueba 2: Kernel	Media: 51474 ms	Percentil 74
Prueba 3: UserSpace	Media: 47626 ms	Percentil 2

Tabla comparativa de Resultados Benchmark 2:

Procesador	Intel Core I7-5500U	
Sistema	CentOS 8	Ubuntu Server
Prueba 1: Total	Media: 35370 ms	Media: 99100 ms
Prueba 2: Kernel	Media: 982 ms	Media: 51474 ms
Prueba 3: UserSpace	Media: 34388 ms	Media: 47626 ms

Podemos ver que hay una gran diferencia con respecto a los datos obtenidos en CentOS que los de Ubuntu, siendo bastante peores los de este segundo sistema. Esto se puede deber a que CentOS es un sistema más ligero que Ubuntu.

Ejercicio 2: Uso de jmeter para test sobre contenedores.

2.0 Enunciado del Ejercicio

Tras probar un test básico para una web, utilizaremos Jmeter para hacer un test sobre una aplicación que ejecuta sobre dos contenedores (uno para la BD y otro para la aplicación en sí). El código está disponible en <https://github.com/davidPalomar-ugr/iseP4JMeter> donde se dan detalles sobre cómo ejecutar la aplicación en una de nuestras máquinas virtuales. El test de Jmeter debe incluir los siguientes elementos:

- El test debe tener parametrizados el Host y el Puerto en el Test Plan (puede hacer referencia usando \$param).
- Debe hacer dos grupos de hebras distintos para simular el acceso de los alumnos y los administradores. Las credenciales de alumno y administrador se cogen de los archivos: alumnos.csv y administrador.csv respectivamente.
- Añadimos esperas aleatorias a cada grupo de hebras (Gaussian Random Timer).
- El login de alumno, su consulta de datos (recuperar datos alumno) y login del administrador son peticiones HTTP.
- El muestreo para simular el acceso de los administradores lo debe coger el archivo apiAlumnos.log (usando un Acces Log Sampler).
- Use una expresión regular (Regular Expression Extractor) para extraer el token JWT que hay que añadir a la cabecera de las peticiones (usando HTTP Header Manager).

2.1 Primeros pasos: Instalación de JMeter en Máquina Host.

En primer lugar, lo que necesitamos es instalar el programa JMeter, que será el entorno de desarrollo principal del ejercicio en el host anfitrión de las máquinas virtuales, en mi caso estoy utilizando Windows 10 64 bits con la siguiente información:



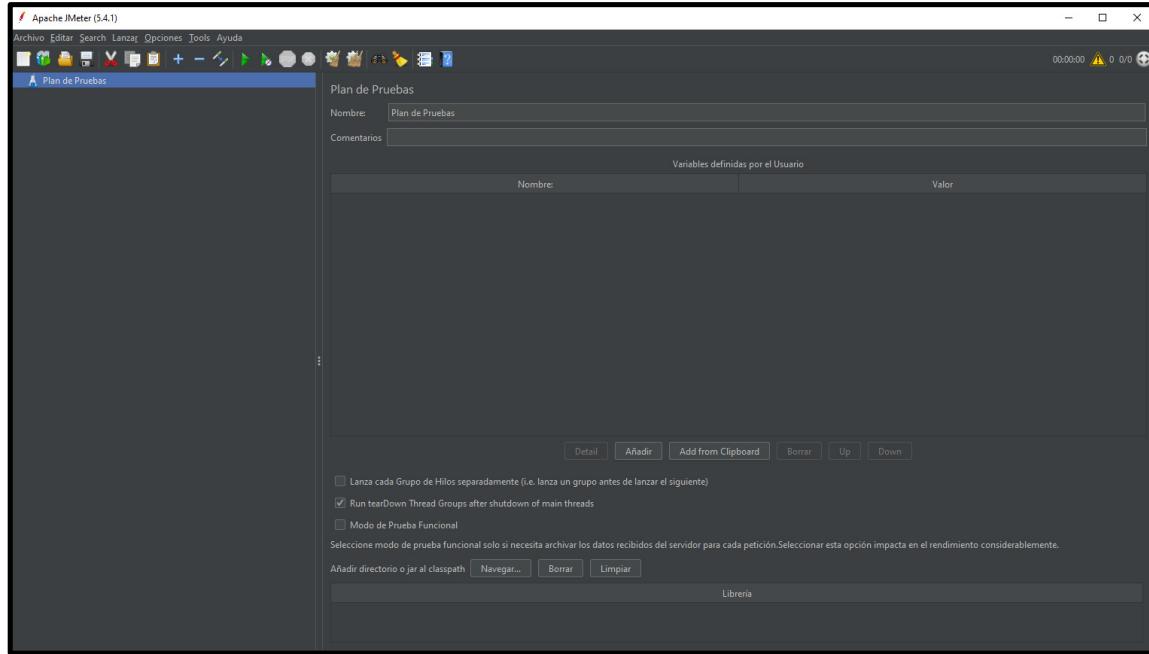
Tras irnos a la página de descarga de JMeter, descargamos el paquete de los binarios y lo instalamos:

Apache JMeter 5.4.1 (Requires Java 8+)

Binaries

[apache-jmeter-5.4.1.tgz sha512 pgp](#)
[apache-jmeter-5.4.1.zip sha512 pgp](#)

Esta es la interfaz del programa. De momento dejaremos aparcada esta sección.



2.2 Primeros pasos: Instalación de Docker en Ubuntu Server.

A continuación, instalaremos la herramienta **Docker** y **Docker-Compose** en la máquina de Ubuntu siguiendo los pasos marcados por el guión de la práctica. Para simplificar esta operación utilizaremos una sesión de ssh para copiar y pegar las órdenes:

Añadir la clave GPG para validar el repositorio de docker, y después lo añadimos:

```
root@perud:/home/perud# curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -
OK
root@perud:/home/perud# sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu $(lsb_release -cs) stable"
Get:1 https://download.docker.com/linux/ubuntu focal InRelease [36.2 kB]
Hit:2 http://es.archive.ubuntu.com/ubuntu focal InRelease
Get:3 https://download.docker.com/linux/ubuntu/focal/stable amd64 Packages [9166 B]
Get:4 http://es.archive.ubuntu.com/ubuntu focal-updates InRelease [114 kB]
Get:5 https://download.docker.com/linux/ubuntu/focal/stable amd64 Contents (deb) [1343 B]
Hit:6 http://repo.zabbix.com/zabbix/5.0/ubuntu focal InRelease
Get:7 http://es.archive.ubuntu.com/ubuntu focal-backports InRelease [101 kB]
Get:8 http://es.archive.ubuntu.com/ubuntu focal-security InRelease [114 kB]
Get:9 http://es.archive.ubuntu.com/ubuntu focal amd64 Contents (deb) [40.9 MB]
Get:10 http://es.archive.ubuntu.com/ubuntu focal-updates/main amd64 Packages [988 kB]
Get:11 http://es.archive.ubuntu.com/ubuntu focal-updates amd64 Contents (deb) [44.5 MB]
Get:12 http://es.archive.ubuntu.com/ubuntu focal-updates/universe amd64 Packages [777 kB]
Get:13 http://es.archive.ubuntu.com/ubuntu focal-backports amd64 Contents (deb) [8260 B]
Get:14 http://es.archive.ubuntu.com/ubuntu focal-security amd64 Contents (deb) [38.8 MB]
Fetched 126 MB in 3min 56s (535 kB/s)
Reading package lists... Done
```

Actualizamos los repositorios:

```
root@perud:/home/perud# apt update
Hit:1 https://download.docker.com/linux/ubuntu focal InRelease
Hit:2 http://es.archive.ubuntu.com/ubuntu focal InRelease
Hit:3 http://repo.zabbix.com/zabbix/5.0/ubuntu focal InRelease
Hit:4 http://es.archive.ubuntu.com/ubuntu focal-updates InRelease
Hit:5 http://es.archive.ubuntu.com/ubuntu focal-backports InRelease
Hit:6 http://es.archive.ubuntu.com/ubuntu focal-security InRelease
Reading package lists... Done
Building dependency tree
Reading state information... Done
93 packages can be upgraded. Run 'apt list --upgradable' to see them.
```

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

Uber



Daniel Pérez Ruiz - perud

GRUPO 2 - PRÁCTICA 4

Pág: 21

Buscamos el paquete docker-ce (edición de la comunidad).

```
root@perud:/home/perud# apt search docker-ce
Sorting... Done
Full Text Search... Done
docker-ce/focal 5:20.10.6~3-0~ubuntu-focal amd64
  Docker: the open-source application container engine

docker-ce-cli/focal 5:20.10.6~3-0~ubuntu-focal amd64
  Docker CLI: the open-source application container engine

docker-ce-rootless-extras/focal 5:20.10.6~3-0~ubuntu-focal amd64
  Rootless support for Docker.
```

Instalamos docker y docker-compose:

```
root@perud:/home/perud# apt install docker-ce docker-compose
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  containerd.io docker-ce-cli docker-ce-rootless-extras docker-scan-plugin git git-man libcurl3-gnutls liberror-perl pigz
  python3-cached-property python3-docker python3-dockerpty python3-docopt python3-texttable python3-websocket slirp4netns
Suggested packages:
  aufs-tools cgroupfs-mount | cgroup-lite git-daemon-run | git-daemon-sysvinit git-doc git-el git-email git-gui gitk gitweb git-cvs
  git-mediawiki git-svn
Recommended packages:
  docker.io
The following NEW packages will be installed:
  containerd.io docker-ce docker-ce-cli docker-ce-rootless-extras docker-compose docker-scan-plugin git git-man libcurl3-gnutls
  liberror-perl pigz python3-cached-property python3-docker python3-dockerpty python3-docopt python3-texttable python3-websocket
  slirp4netns
0 upgraded, 18 newly installed, 0 to remove and 93 not upgraded.
Need to get 114 MB of archives.
After this operation, 506 MB of additional disk space will be used.
Do you want to continue? [Y/n] Y
Get:1 http://es.archive.ubuntu.com/ubuntu focal/universe amd64 pigz amd64 2.4-1 [57.4 kB]
Get:2 https://download.docker.com/linux/ubuntu/focal/stable amd64 containerd.io amd64 1.4.4-1 [28.3 MB]
Get:3 http://es.archive.ubuntu.com/ubuntu focal/universe amd64 python3-cached-property all 1.5.1-4 [10.9 kB]
Get:4 http://es.archive.ubuntu.com/ubuntu focal/universe amd64 python3-websocket all 0.53.0-2ubuntu1 [32.3 kB]
Get:5 http://es.archive.ubuntu.com/ubuntu focal/universe amd64 python3-docker all 4.1.0-1 [83.4 kB]
Get:6 http://es.archive.ubuntu.com/ubuntu focal/universe amd64 python3-dockerpty all 0.4.1-2 [11.1 kB]
Get:7 http://es.archive.ubuntu.com/ubuntu focal/universe amd64 python3-docopt all 0.6.2-2.2ubuntu1 [19.7 kB]
Get:8 http://es.archive.ubuntu.com/ubuntu focal/universe amd64 python3-texttable all 1.6.2-2 [11.0 kB]
Get:9 http://es.archive.ubuntu.com/ubuntu focal/universe amd64 docker-compose all 1.25.0-1 [92.7 kB]
Get:10 http://es.archive.ubuntu.com/ubuntu focal-updates/main amd64 libcurl3-gnutls amd64 7.68.0-1ubuntu2.5 [232 kB]
Get:11 http://es.archive.ubuntu.com/ubuntu focal/main amd64 liberror-perl all 0.17029-1 [26.5 kB]
Get:12 http://es.archive.ubuntu.com/ubuntu focal-updates/main amd64 git-man all 1:2.25.1-1ubuntu3.1 [884 kB]
Get:13 https://download.docker.com/linux/ubuntu/focal/stable amd64 docker-ce-cli amd64 5:20.10.6~3-0~ubuntu-focal [41.4 MB]
33% [12 git-man 47.6 kB/884 kB 5%] [13 docker-ce-cli 0 B/41.4 kB]
```

Comprobamos en primer lugar el servicio de docker, y después hacemos una prueba de funcionamiento:

```
root@perud:/home/perud# systemctl status docker
● docker.service - Docker Application Container Engine
   Loaded: loaded (/lib/systemd/system/docker.service; enabled; vendor preset: enabled)
   Active: active (running) since Sat 2021-05-22 15:30:04 UTC; 29s ago
     TriggeredBy: ● docker.socket
     Docs: https://docs.docker.com
       Main PID: 4061 (dockerd)
         Tasks: 8
        Memory: 40.0M
        CGroup: /system.slice/docker.service
                  └─4061 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock

May 22 15:30:02 perud dockerd[4061]: time="2021-05-22T15:30:02.928467924Z" level=warning msg="Your kernel does not support CPU realtime"
May 22 15:30:02 perud dockerd[4061]: time="2021-05-22T15:30:02.928563633Z" level=warning msg="Your kernel does not support cgroup blkio"
May 22 15:30:02 perud dockerd[4061]: time="2021-05-22T15:30:02.928648116Z" level=warning msg="Your kernel does not support cgroup blkio"
May 22 15:30:02 perud dockerd[4061]: time="2021-05-22T15:30:02.928869660Z" level=info msg="Loading containers: start."
May 22 15:30:03 perud dockerd[4061]: time="2021-05-22T15:30:03.512828509Z" level=info msg="Default bridge (docker0) is assigned with an IP from the bridge interface eth0"
May 22 15:30:03 perud dockerd[4061]: time="2021-05-22T15:30:03.644737764Z" level=info msg="Loading containers: done."
May 22 15:30:04 perud dockerd[4061]: time="2021-05-22T15:30:04.077015640Z" level=info msg="Docker daemon" commit=e8728dd2 graphdriver(s)
May 22 15:30:04 perud dockerd[4061]: time="2021-05-22T15:30:04.078428018Z" level=info msg="Daemon has completed initialization"
May 22 15:30:04 perud systemd[1]: Started Docker Application Container Engine.
May 22 15:30:04 perud dockerd[4061]: time="2021-05-22T15:30:04.135554699Z" level=info msg="API listen on /run/docker.sock"
lines 1-21/21 (END)
```

Previamente añadimos el usuario al grupo docker:

```
root@perud:/home/perud# usermod -aG docker perud
root@perud:/home/perud# groups perud
perud : perud adm cdrom sudo dip plugdev lxd docker
```

Comprobamos la información de docker y ejecutamos un script de prueba:

```
root@perud:/home/perud# docker info ; docker run hello-world
Client:
  Context:    default
  Debug Mode: false
  Plugins:
    app: Docker App (Docker Inc., v0.9.1-beta3)
    buildx: Build with BuildKit (Docker Inc., v0.5.1-docker)
    scan: Docker Scan (Docker Inc., v0.7.0)

Server:
  Containers: 0
  Running: 0
  Paused: 0
  Stopped: 0
  Images: 0
  Server Version: 20.10.6
  Storage Driver: overlay2
    Backing Filesystem: extfs
    Supports d_type: true
    Native Overlay Diff: true
    userxattr: false
  Logging Driver: json-file
  Cgroup Driver: cgroupfs
  Cgroup Version: 1
  Plugins:
    Volume: local
    Network: bridge host ipvlan macvlan null overlay
    Log: awslogs fluentd gcplogs gelf journald json-file local logentries splunk syslog
  Swarm: inactive
  Runtimes: io.containerd.runtime.v1.linux runc io.containerd.runc.v2
  Default Runtime: runc
  Init Binary: docker-init
  containerd version: 05f951a3781f4f2c1911b05e61c160e9c30eaa8e
  runc version: 12644e614e25b05da6fd08a38ffa0cfe1903fdec
  init version: de40ad0
  Security Options:
    apparmor
    seccomp
      Profile: default
  Kernel Version: 5.4.0-67-generic
  Operating System: Ubuntu 20.04.1 LTS
  OSType: linux
  Architecture: x86_64
  CPUs: 1
  Total Memory: 3.844GiB
  Name: perud
  ID: ZWMN:7C7A:MWTB:A247:H03D:YP4Z:VK5N:IIFZ:ESWE:UPGF:4NBL:JZDH
  Docker Root Dir: /var/lib/docker
  Debug Mode: false
  Registry: https://index.docker.io/v1/
```

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

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

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

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

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

Comprobamos ahora el estado de docker-compose y la versión instalada:

```
root@perud:/home/perud# docker-compose
Define and run multi-container applications with Docker.

Usage:
  docker-compose [-f <arg>...] [options] [COMMAND] [ARGS...]
  docker-compose -h|--help

Options:
  -f, --file FILE           Specify an alternate compose file
                            (default: docker-compose.yml)
  -p, --project-name NAME   Specify an alternate project name
                            (default: directory name)
  --verbose                 Show more output
  --log-level LEVEL         Set log level (DEBUG, INFO, WARNING, ERROR, CRITICAL)
  --no-ansi                  Do not print ANSI control characters
  -v, --version               Print version and exit
  -H, --host HOST            Daemon socket to connect to

  --tls                      Use TLS; implied by --tlsverify
  --tlscacert CA_PATH        Trust certs signed only by this CA
  --tlscert CLIENT_CERT_PATH Path to TLS certificate file
  --tlskey TLS_KEY_PATH      Path to TLS key file
  --tlsverify                Use TLS and verify the remote
  --skip-hostname-check      Don't check the daemon's hostname against the
                            name specified in the client certificate
  --project-directory PATH   Specify an alternate working directory
                            (default: the path of the Compose file)
  --compatibility             If set, Compose will attempt to convert keys
                            in v3 files to their non-Swarm equivalent
  --env-file PATH             Specify an alternate environment file
```

```
root@perud:/home/perud# docker-compose --version
docker-compose version 1.25.0, build unknown
```

2.3 Levantar el servicio iseP4Jmeter

Descargamos el repositorio de la aplicación iseP4Jmeter y lo activamos con la correspondiente orden de docker compose.

```
root@perud:/home/perud# git clone https://github.com/davidPalomar-ugr/iseP4JMeter.git
Cloning into 'iseP4JMeter'...
remote: Enumerating objects: 3782, done.
remote: Counting objects: 100% (8/8), done.
remote: Compressing objects: 100% (8/8), done.
remote: Total 3782 (delta 3), reused 0 (delta 0), pack-reused 3774
Receiving objects: 100% (3782/3782), 7.78 MiB | 6.03 MiB/s, done.
Resolving deltas: 100% (709/709), done.
```

```
root@perud:/home/perud# cd iseP4JMeter/
root@perud:/home/perud/iseP4JMeter# docker-compose up
Creating network "isep4jmeter_default" with the default driver
Pulling mongodb (mongo)...latest: Pulling from library/mongo
01bf7da0a088c: Pull complete
f3b4a5f15c7a: Pull complete
57ffbe87baa1: Pull complete
77d5e5c7eab9: Pull complete
43798cf18b45: Pull complete
67349a81f435: Pull complete
590845b1f17c: Pull complete
1f2ff17242ce: Pull complete
6f11b2ce0594: Pull complete
91532386f4ec: Pull complete
705ef0ab262e: Pull complete
e6238126b609: Pull complete
Digest: sha256:8b35c0a75c2dbf23110ed2485feca567ec9ab743feee7a0d7a148f806daf5e86
Status: Downloaded newer image for mongo:latest
```

```
Building mongodbinit
Step 1/5 : FROM mongo
--> 07630e791de3
Step 2/5 : COPY ./scripts/* /tmp/
--> 6a8a740c38a0
Step 3/5 : RUN chmod 755 /tmp/initializeMongoDB.sh
--> Running in 46b55d415ef2
Removing intermediate container 46b55d415ef2
--> 41dd6294883d
Step 4/5 : WORKDIR /tmp
--> Running in b434335b7a22
Removing intermediate container b434335b7a22
--> 9512d7f76e20
Step 5/5 : CMD ./initializeMongoDB.sh
--> Running in dc7ca007a726
Removing intermediate container dc7ca007a726
--> e17c71706c40
Successfully built e17c71706c40
Successfully tagged isep4jmeter_mongodbinit:latest
WARNING: Image for service mongodbinit was built because it did not already exist. To rebuild this image you must use `docker-compose
ld` or `docker-compose up --build`.
```

Tras terminar el despliegue podemos ya crear el test de comprobación.

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

Uber



Daniel Pérez Ruiz - perud

GRUPO 2 - PRÁCTICA 4

Pág: 25

2.4 Configuración del test con JMeter.

Aquí se indican los pasos a seguir para configurar el test pedido en el ejercicio.

Si hacemos **docker-compose up**, podemos acceder al servicio en el navegador con:
<http://192.168.56.105:3030>

ETSII Alumnos API

```
Descripción de la API Restful:  
POST /api/v1/auth/login  
    Parámetros:  
        login:<emailusuario>  
        password:<secreto>  
    Seguridad:  
        Acceso protegido con BasicAuth (etsiiApi:laApiDeLaETSIIDaLache)  
    Retorna:  
        JWT Token  
  
GET /api/v1/alumnos/alumno/<email>  
    Seguridad:  
        Token JWT valido en cabecera estandar authorization: Bearer <token>  
        Alumnos solo pueden solicitar sus datos. Administradores pueden solicitar cualquier alumno válido  
    Retorna:  
        Objeto Json con perfil de alumno
```

Paso 1: Parametrización de HOST y PUERTO.

Plan de Pruebas	
Nombre:	TestE2P4_perud
Comentarios:	Test confeccionado para el ejercicio 2 de la práctica 4 de ISE - Daniel Pérez Ruiz
Variables definidas por el Usuario	
HOST	Nombre: 192.168.56.105
PORt	Valor: 3000

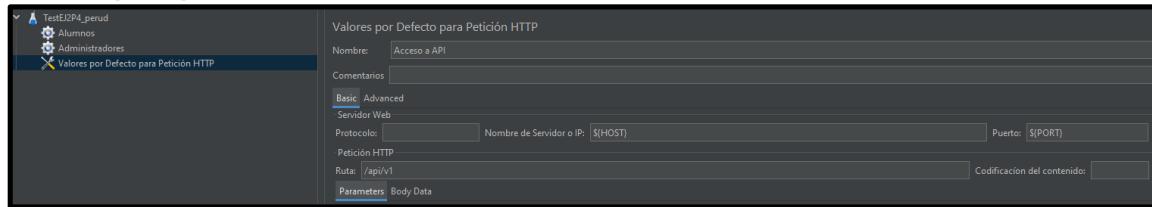
Paso 2: Creación de grupos de hebras. Se ha creado un grupo para simular a los alumnos (con un total de 5 hebras), y otro grupo para simular a los administradores (con un total de 3 hebras).

Para crear un grupo de hebras: **Añadir – Hilos – Grupo de Hilos.**

Grupo de Hilos	
Nombre:	Alumnos
Comentarios:	Simulación de una hebra dedicada al login de los Alumnos
Acción a tomar después de un error de Muestreador:	<input checked="" type="radio"/> Continuar <input type="radio"/> Comenzar siguiente iteración <input type="radio"/> Parar Hilo <input type="radio"/> Parar Test <input type="radio"/> Parar test ahora
Propiedades de Hilo	
Número de Hilos:	5
Periodo de Subida (en segundos):	1
Contador del bucle:	<input type="checkbox"/> Sin fin <input checked="" type="checkbox"/> Same user on each iteration

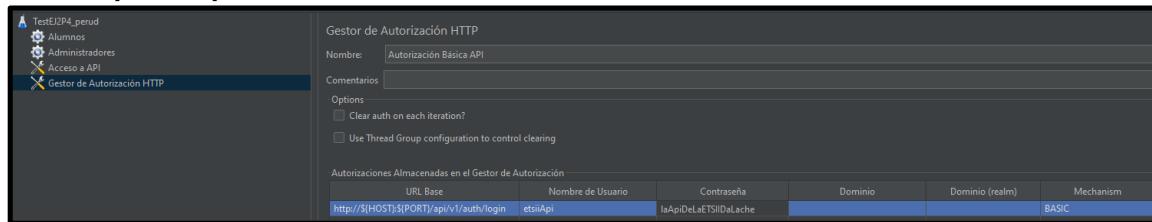
*Paso 3: A continuación, añadimos ahora la petición de acceso al servidor. Gracias a que tenemos el host y puerto parametrizados, basta con hacer referencia a esos valores. Lo que necesitamos crear aquí es una configuración para los **Valores por defecto de la petición HTTP**.*

*Para crear petición por defecto HTTP: **Añadir – Elemento de Configuración – Valores por defecto para petición HTTP**.*



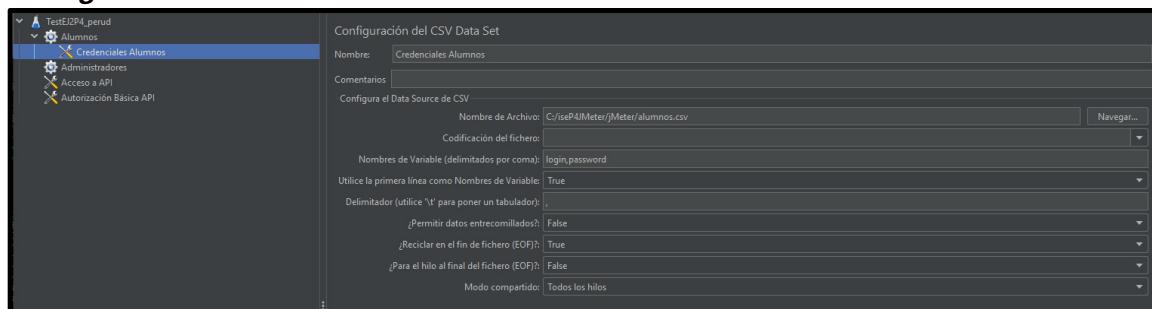
Paso 4: Definimos la autorización básica a la API con los parámetros indicados en la página de github.

*Para configurar la autorización HTTP: **Añadir – Elemento de Configuración – Valores por defecto para la petición HTTP**.*



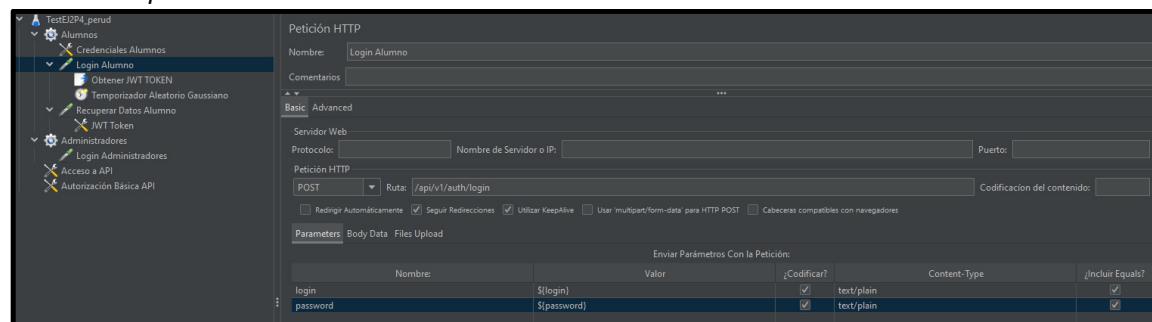
*Paso 5: Definimos las credenciales para los alumnos. Le decimos en los parámetros que el fichero que queremos coger es **alumnos.csv** cuyos campos son **usuario,contraseña**.*

*Para especificar los datos de los alumnos: **Añadir – Elemento de configuración – Configuración del CSV Data Set***



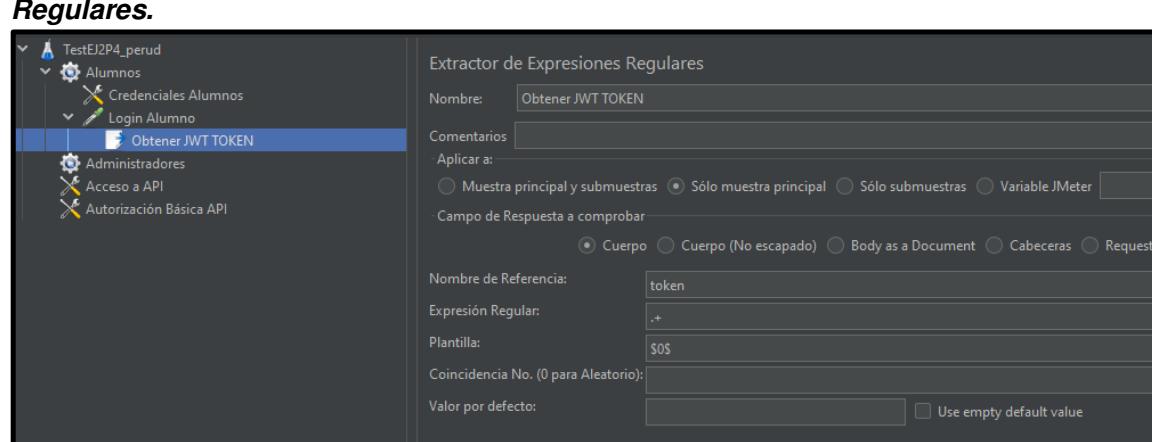
Paso 6: Creamos una petición POST para los alumnos. Indicamos la ruta donde se procede a la autenticación dentro del objeto de muestreo de petición HTTP.

Para crear petición HTTP: Añadir – Muestreador – Petición HTTP.



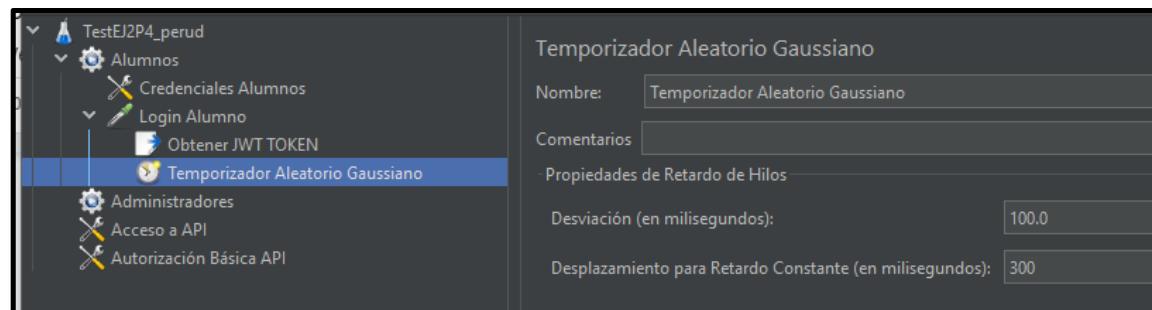
*Paso 7: Según el diagrama cuando hacemos una autenticación válida el servidor nos devuelve un mensaje OK además de un token. Recuperaremos esa información añadiendo un nuevo objeto específico para ello, concretamente, el **Extractor de Expresiones Regulares**.*

Para configurar un EER: Añadir – Post Procesadores – Extractor de Expresiones Regulares.



Paso 8: Para simular los accesos aleatorios de una manera más realista, añadimos un temporizador que siga una distribución Gaussiana de números aleatorios.

Para configurar temporizador: Añadir – Temporizador – Temporizador Aleatorio Gaussiano.



Paso 9: Recuperamos los datos de los alumnos configurando otro item de Petición HTTP con GET.

Para configurar petición HTTP: Añadir – Muestreador – Petición HTTP

The screenshot shows the 'Petición HTTP' (HTTP Request) configuration screen. The 'Nombre' (Name) field is set to 'Recuperar Datos Alumno'. The 'Protocolo' (Protocol) dropdown is set to 'GET' and the 'Ruta' (Path) is '/api/v1/alumnos/alumno/\${_urlencodecode\${{login}}}'.

Nombre:	Valor:	¿Codificar?:	Content-Type:	¿Incluir Equals?:
login	\$(login)	<input checked="" type="checkbox"/>	text/plain	<input checked="" type="checkbox"/>

Paso 10: En caso de que el token deje de ser válido por algún motivo, el gestor de cabecera HTTP nos ayuda a resolver este tipo de problemas.

Para configurar gestor de cabecera: Añadir – Elemento de Configuración – Gestor de Cabecera HTTP.

The screenshot shows the 'Gestor de Cabecera HTTP' (HTTP Header Manager) configuration screen. The 'Nombre' (Name) field is set to 'JWT Token'. In the 'Cabeceras Almacenadas en el Gestor de Cabeceras' (Stored Headers in the Header Manager) section, there is a single entry for 'Authorization' with the value 'Bearer \${token}'.

Terminada la configuración del grupo de alumnos ahora realizaremos los mismos pasos para el grupo de administradores.

Paso 11: Definimos el login para administradores.

Para configurar petición HTTP: Añadir – Muestreador – Petición HTTP

The screenshot shows the 'Petición HTTP' (HTTP Request) configuration screen. The 'Nombre' (Name) field is set to 'Login Administradores'. The 'Protocolo' (Protocol) dropdown is set to 'POST' and the 'Ruta' (Path) is '/api/v1/auth/login'.

Nombre:	Valor:	¿Codificar?:	Content-Type:	¿Incluir Equals?:
login	\$(login)	<input checked="" type="checkbox"/>	text/plain	<input checked="" type="checkbox"/>
password	\$(password)	<input checked="" type="checkbox"/>	text/plain	<input checked="" type="checkbox"/>

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



Daniel Pérez Ruiz - perud

GRUPO 2 - PRÁCTICA 4

Pág: 29

Paso 12: Definimos las credenciales de los administradores.

Para especificar los datos de los administradores: **Añadir – Elemento de configuración – Configuración del CSV Data Set**

Paso 13: Recuperamos el token de los administradores con el Extractor de Expresiones Regulares.

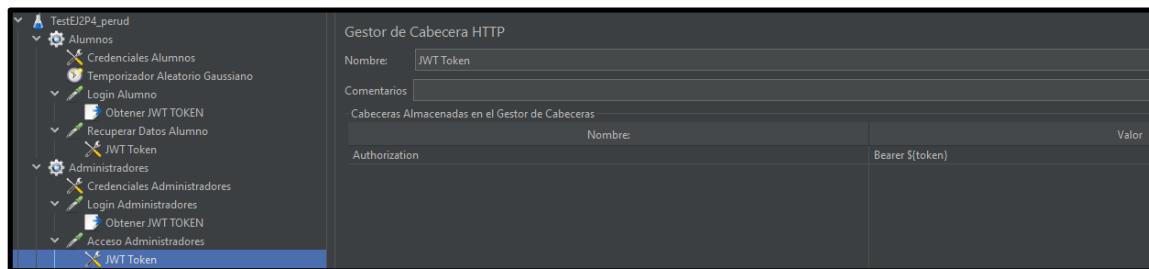
Para configurar un EER: **Añadir – Post Procesadores – Extractor de Expresiones Regulares.**

Paso 14: Como los administradores pueden ver quién entra al sistema, es necesario en el grupo de los administradores un item de muestreo al log de accesos.

Para configurar muestreo a log: **Añadir – Muestreador – Muestreador de Acceso a Log**

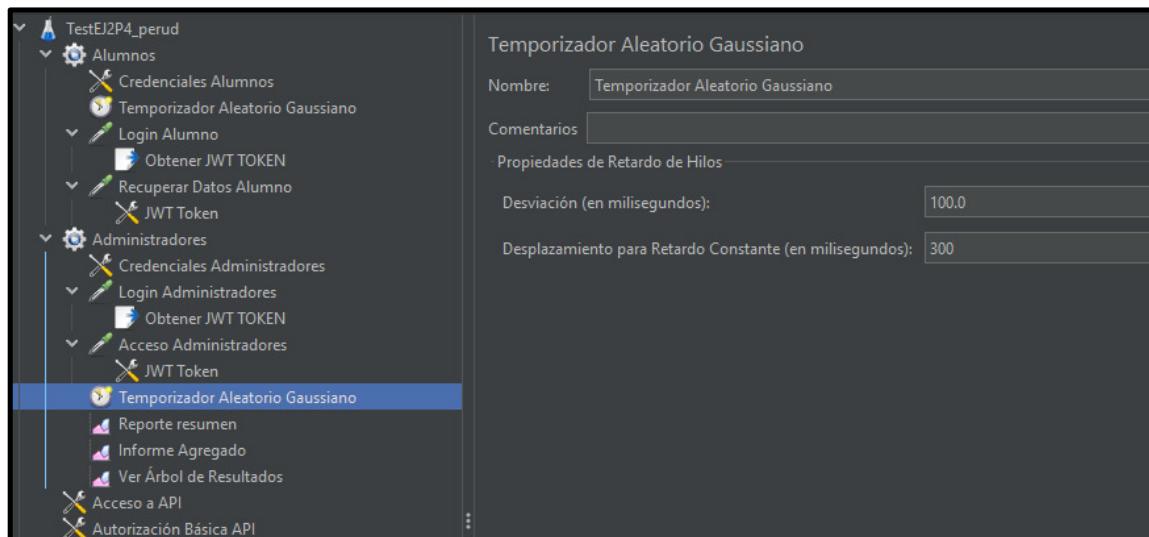
Paso 15: Nuevamente, con el gestor de cabeceras, podemos resolver problemas asociados con el token.

Para configurar gestor de cabecera: Añadir – Elemento de Configuración – Gestor de Cabecera HTTP.

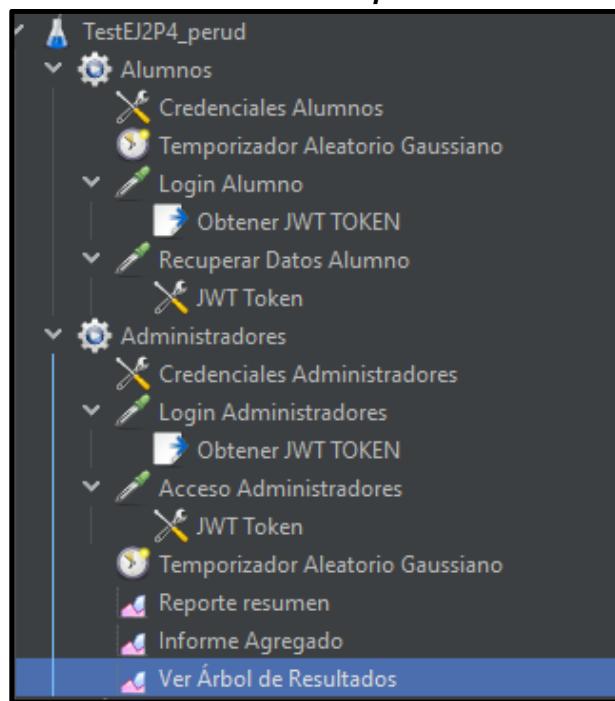


Paso 16: Añadimos el temporizador al grupo.

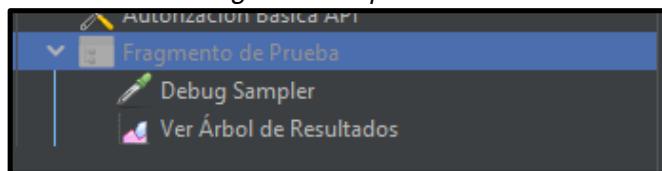
Para configurar temporizador: Añadir – Temporizador – Temporizador Aleatorio Gaussiano.



Paso 17: Añadimos items de generación de informe según la imagen del git: **Ver árbol de resultados, Reporte Resumen, e Informe Agregado.**
Todos estos items se encuentran en **Añadir – Receptor.**



Paso 18: Finalmente añadimos un fragmento de prueba.



2.5 Probar rendimiento del test.

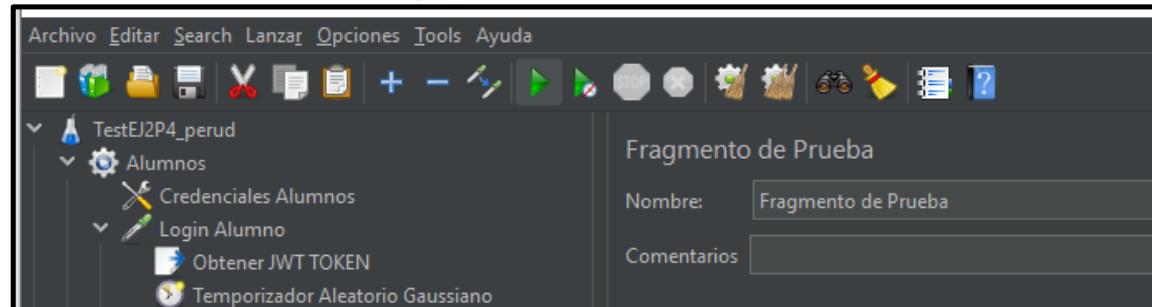
Utilizamos el docker-compose up para activar el servicio.

```

mongodb_1    | {"t": {"$date": "2021-05-22T18:32:17.242+00:00"}, "s": "I", "c": "NETWORK", "id": 51800
, "ctx": "conn10", "msg": "client metadata", "attr": {"remote": "172.20.0.4:37114", "client": "conn10", "do
c": {"driver": {"name": "mongo-go-driver", "version": "v1.4.2"}, "os": {"type": "linux", "architecture": "amde
4"}, "platform": "go1.15.8", "application": {"name": "mongoimport"}}}}
mongodbinit_1 | 2021-05-22T18:32:17.246+0000  connected to: mongodb://mongodb/
mongodb_1    | {"t": {"$date": "2021-05-22T18:32:17.246+00:00"}, "s": "I", "c": "NETWORK", "id": 51800
, "ctx": "conn12", "msg": "client metadata", "attr": {"remote": "172.20.0.4:37118", "client": "conn12", "do
c": {"driver": {"name": "mongo-go-driver", "version": "v1.4.2"}, "os": {"type": "linux", "architecture": "amde
4"}, "platform": "go1.15.8", "application": {"name": "mongoimport"}}}}
mongodbinit_1 | 2021-05-22T18:32:17.253+0000  20 document(s) imported successfully. 0 document(s)
Failed to import.
mongodb_1    | {"t": {"$date": "2021-05-22T18:32:17.262+00:00"}, "s": "I", "c": "-", "id": 20883
, "ctx": "conn11", "msg": "Interrupted operation as its client disconnected", "attr": {"opId": 1106}}
mongodb_1    | {"t": {"$date": "2021-05-22T18:32:17.262+00:00"}, "s": "I", "c": "NETWORK", "id": 22944
, "ctx": "conn10", "msg": "Connection ended", "attr": {"remote": "172.20.0.4:37114", "connectionId": 10, "c
onnectionCount": 2}}
mongodb_1    | {"t": {"$date": "2021-05-22T18:32:17.263+00:00"}, "s": "I", "c": "NETWORK", "id": 22944
, "ctx": "conn11", "msg": "Connection ended", "attr": {"remote": "172.20.0.4:37116", "connectionId": 11, "c
onnectionCount": 1}}
mongodb_1    | {"t": {"$date": "2021-05-22T18:32:17.264+00:00"}, "s": "I", "c": "NETWORK", "id": 22944
, "ctx": "conn12", "msg": "Connection ended", "attr": {"remote": "172.20.0.4:37118", "connectionId": 12, "c
onnectionCount": 0}}
nodejs_1     | (node:18) DeprecationWarning: current URL string parser is deprecated, and will be
removed in a future version. To use the new parser, pass option { useNewUrlParser: true } to MongoClient.connect.
nodejs_1     | (node:18) [MONGODB DRIVER] Warning: Current Server Discovery and Monitoring engine
is deprecated, and will be removed in a future version. To use the new Server Discover and Monitorin
g engine, pass option { useUnifiedTopology: true } to the MongoClient constructor.
mongodb_1    | {"t": {"$date": "2021-05-22T18:32:17.430+00:00"}, "s": "I", "c": "NETWORK", "id": 22943
, "ctx": "listener", "msg": "Connection accepted", "attr": {"remote": "172.20.0.3:36876", "connectionId": 13, "c
onnectionCount": 1}}
mongodb_1    | {"t": {"$date": "2021-05-22T18:32:17.506+00:00"}, "s": "I", "c": "NETWORK", "id": 51800
, "ctx": "conn13", "msg": "client metadata", "attr": {"remote": "172.20.0.3:36876", "client": "conn13", "do
c": {"driver": {"name": "nodejs", "version": "3.6.6"}, "os": {"type": "Linux", "name": "linux", "architecture": "x64"}, "version": "5.4.0-67-generic"}, "platform": "Node.js v8.17.0, LE (legacy)"}}
isep4jmeter_mongodbinit_1 exited with code 0
`[S]`S

```

Iniciamos el test con el botón de play:



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

Uber



Daniel Pérez Ruiz - perud

GRUPO 2 - PRÁCTICA 4

Pág: 33

Podemos ver que se realiza correctamente el test. Aquí se muestran capturas que lo corroboran con los receptores que hemos incorporado.

The screenshot shows the JMeter Test Plan interface. On the left, the tree view displays a test plan named 'TestEJ2P4_perud' containing several samplers under 'Alumnos' and 'Administradores' groups, along with a 'Ver Árbol de Resultados' sampler which is currently selected. On the right, the 'Ver Árbol de Resultados' dialog is open, showing the results of the execution. The results table has columns for 'Nombre' (Name), 'Comentarios' (Comments), and 'Texto' (Text). The 'Texto' column lists numerous successful requests, all marked with a green checkmark, indicating they were executed correctly. Examples of the listed requests include 'Login Administradores', 'http://192.168.56.105:3000/api/v1/alumnos/alumno/debon', and various URLs for 'cerval' and 'marib' users.

Nombre	Comentarios	Texto
Ver Árbol de Resultados	Escribir todos los datos a Archivo	<ul style="list-style-type: none">✓ Login Administradores✓ Login Administradores✓ http://192.168.56.105:3000/api/v1/alumnos/alumno/debon✓ Login Administradores✓ http://192.168.56.105:3000/api/v1/alumnos/alumno/debon✓ Login Administradores✓ Login Administradores✓ http://192.168.56.105:3000/api/v1/alumnos/alumno/cerval✓ http://192.168.56.105:3000/api/v1/alumnos/alumno/debon✓ Login Administradores✓ http://192.168.56.105:3000/api/v1/alumnos/alumno/cerval✓ Login Administradores✓ Login Administradores✓ http://192.168.56.105:3000/api/v1/alumnos/alumno/marib✓ http://192.168.56.105:3000/api/v1/alumnos/alumno/marib✓ http://192.168.56.105:3000/api/v1/alumnos/alumno/cerval✓ Login Administradores✓ Login Administradores

Para el primer login de Administrador. Vemos que el resultado del muestreador es que se ha obtenido el mensaje OK con código 200 tal y como se indica la práctica.

Ver Árbol de Resultados

Nombre: Ver Árbol de Resultados

Comentarios:

Escribir todos los datos a Archivo

Nombre de archivo:

Buscar: ...

Sensible a mayúsculas Expresión regular Search Reset

Resultado del Muestreador Página Datos de Respuesta

Texto

Resultados de muestra (16)

1. Login Administradores
http://192.168.56.105:3000/api/v1/alumnos/alumno/debor
2. Login Administradores
http://192.168.56.105:3000/api/v1/alumnos/alumno/debor
3. Login Administradores
http://192.168.56.105:3000/api/v1/alumnos/alumno/debor
4. Login Administradores
http://192.168.56.105:3000/api/v1/alumnos/alumno/debor
5. Login Administradores
http://192.168.56.105:3000/api/v1/alumnos/alumno/cervai
6. Login Administradores
http://192.168.56.105:3000/api/v1/alumnos/alumno/debor
7. Login Administradores
http://192.168.56.105:3000/api/v1/alumnos/alumno/cervai
8. Login Administradores
http://192.168.56.105:3000/api/v1/alumnos/alumno/cervai
9. Login Administradores
http://192.168.56.105:3000/api/v1/alumnos/alumno/marib
10. Login Administradores
http://192.168.56.105:3000/api/v1/alumnos/alumno/debor
11. Login Administradores
http://192.168.56.105:3000/api/v1/alumnos/alumno/debor
12. Login Administradores
http://192.168.56.105:3000/api/v1/alumnos/alumno/debor
13. Login Administradores
http://192.168.56.105:3000/api/v1/alumnos/alumno/debor
14. Login Administradores
http://192.168.56.105:3000/api/v1/alumnos/alumno/debor
15. Login Administradores
http://192.168.56.105:3000/api/v1/alumnos/alumno/debor
16. Login Administradores
http://192.168.56.105:3000/api/v1/alumnos/alumno/debor

Nombre del hilo: Administradores 2-1
Comienzo de muestra: 2021-05-22 20:51:03 CEST
Tiempo de carga: 6
Connect Time: 2
Latencia: 6
Tamaño en bytes: 595
Sent bytes: 335
Headers size in bytes: 399
Body size in bytes: 196
Conteo de muestra: 1
Conteo de error: 0
Data type ("text"|"bin"|""): text
Código de respuesta: 200
Mensaje de respuesta: OK

HTTPSampleResult campos:
ContentType: text/html; charset=utf-8
DataEncoding: utf-8

Esta es la petición realizada al servidor.

Resultado del Muestreador Petición Datos de Respuesta

Texto

Resultados de muestra (16)

Request Body Cabeceras de petición:

```
POST http://192.168.56.105:3000/api/v1/auth/Login
Content-Type: application/x-www-form-urlencoded
Content-Length: 10
User-Agent: Apache-HttpClient/4.5.13 (Java/1.8.0_292)
Host: 192.168.56.105:3000
Connection: Keep-Alive
Accept-Encoding: gzip, deflate
Accept: */*
```

Encontrar Sensible a mayúsculas Expresión regular

1. POST http://192.168.56.105:3000/api/v1/auth/Login
2.
3. POST data
4. login stuartkirby 40etsii ugr.es@password temporal
5.
6. [no cookies]
7.

Aquí se ve la respuesta enviada al cliente, con el token de autenticación.

Resultado del Muestreador Petición Datos de Respuesta

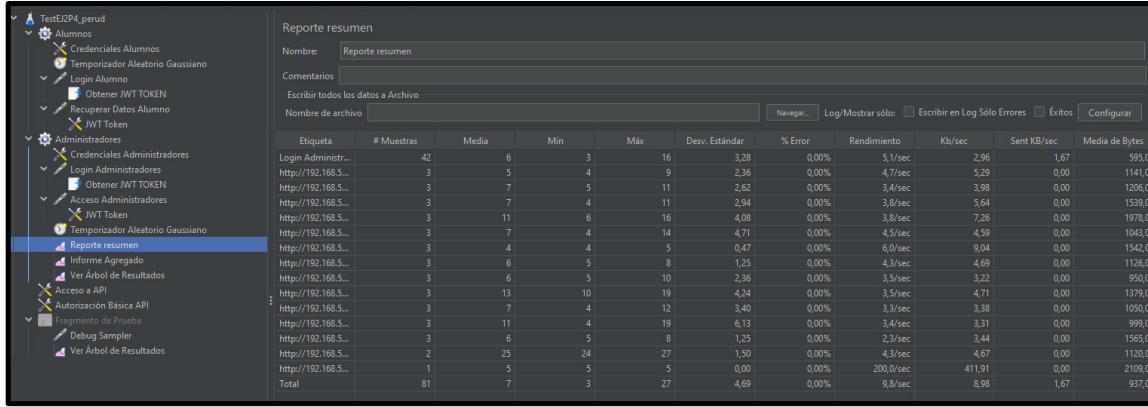
Texto

Resultados de muestra (16)

Response Body Cabeceras de respuesta:

```
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJzdWIjOiJwMDAsImIhdCI6MTYyMTcwOTQ2MjYwIiBmJmljoxNjIxNzA5NDAzLCIiHAIoE2MjE3MTMxMjMs...  
bmIzdHUhZG9yIn0.1W-m1a6n3WG-wb-Wbb9bjHeTEKGLx34VzaXppwLw
```

Con este reporte añadido en formato de tabla, se nos indican los accesos realizados por el test y el porcentaje de error en la autenticación. Comprobamos que es del 0.00%.



The screenshot shows the JMeter interface. On the left, there's a tree view of the test plan with nodes like 'Alumnos', 'Administradores', and 'Reporte resumen'. The 'Reporte resumen' node is selected. On the right, there's a 'Reporte resumen' panel with fields for 'Nombre' (Reporte resumen), 'Comentarios', and 'Escribir todos los datos a Archivo'. Below this is a table with columns: Etiquetas, # Muestras, Media, Min, Máx, Desv. Estándar, % Error, Rendimiento, Kb/sec, Sent Kb/sec, and Media de Bytes. The table contains data for various HTTP requests, showing metrics like 0.00% error and 0.00% sent KB/sec. A total row at the bottom shows 81 samples and 0.00% error.

Etiquetas	# Muestras	Media	Min	Máx	Desv. Estándar	% Error	Rendimiento	Kb/sec	Sent Kb/sec	Media de Bytes
Login Administr...	42	6	3	16	3,28	0,00%	5,1/sec	2,96	1,67	595,0
http://192.168.5...	3	5	4	9	2,36	0,00%	4,7/sec	5,29	0,00	1141,0
http://192.168.5...	3	7	5	11	2,62	0,00%	3,4/sec	3,98	0,00	1206,0
http://192.168.5...	3	7	4	11	2,94	0,00%	3,8/sec	5,64	0,00	1539,0
http://192.168.5...	3	11	6	16	4,08	0,00%	3,8/sec	7,26	0,00	1978,0
http://192.168.5...	3	7	4	14	4,71	0,00%	4,5/sec	4,59	0,00	1043,0
http://192.168.5...	3	4	4	5	0,47	0,00%	6,0/sec	9,04	0,00	1542,0
http://192.168.5...	3	6	5	8	1,25	0,00%	4,3/sec	4,69	0,00	1126,0
http://192.168.5...	3	6	5	10	2,36	0,00%	3,5/sec	3,22	0,00	950,0
http://192.168.5...	3	13	10	19	4,24	0,00%	3,5/sec	4,71	0,00	1379,0
http://192.168.5...	3	7	4	12	3,40	0,00%	3,3/sec	3,38	0,00	1050,0
http://192.168.5...	3	11	4	19	6,13	0,00%	3,4/sec	3,31	0,00	999,0
http://192.168.5...	3	6	5	8	1,25	0,00%	2,3/sec	3,44	0,00	1565,0
http://192.168.5...	2	25	24	27	1,50	0,00%	4,3/sec	4,67	0,00	1120,0
Total	81	7	3	27	4,69	0,00%	9,8/sec	8,98	1,67	937,0