Práctica 4:

Benchmarking y Ajuste del Sistema

Práctica realizada por Juan Sánchez Rodríguez

Sistemas operativos utilizados:

- Ubuntu Server
- CentOS 7

Sesión 1:

Instalación Phoronix:

Ubuntu:

```
jsr@ubuntu:~$ sudo apt install -y phoronix-test-suite php-zip
```

CentOS:

```
[root@localhost jrodriguez]# yum install -y phoronix-test-suite_
```

Para ver los tests disponibles debemos usar la siguiente instrucción (vale en ambos sistemas):

#> phoronix-test-suite list-available-tests

Ahora vamos a probar un par de tests de la lista, en este caso “ramspeed” y “sudokut”, veremos que al ejecutar la instrucción se deberán instalar algunas dependencias, este proceso puede durar varios minutos:

```
jsr@ubuntu:~$ phoronix-test-suite benchmark pts/ramspeed

The following dependencies are needed and will be installed:

- build-essential
- autoconf
- mesa-utils
- unzip

This process may take several minutes.
Extrayendo plantillas para los paquetes: 100%
_
```

Una vez ejecutado nos preguntará por la configuración, en mi caso he seleccionado en ambas máquinas el type “Add” y el benchmark “Integer” (Dura unos 11 minutos), aquí podemos ver la pantalla que nos sale una vez seleccionadas las opciones en CentOS (en Ubuntu server es similar):

```
  PROCESSOR:            Intel Core i7-4720HQ
    Core Count:         1
    Extensions:         SSE 4.2 + AVX2 + AVX + RDRAND + FSGSBASE
    Cache Size:         6144 KB
    Microcode:          0x0

  GRAPHICS:             VMware SVGA II
    Screen:             2048x2048

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

  MEMORY:               1024MB

  DISK:                 9GB VBOX HDD
    File-System:        xfs
    Mount Options:      attr2 inode64 noquota relatime rw seclabel
    Disk Scheduler:     CFQ

  OPERATING SYSTEM:     CentOS Linux 7
    Kernel:             3.10.0-514.el7.x86_64 (x86_64)
    Compiler:           GCC 4.8.5 20150623
    System Layer:       KVM VMware
    Security:           SELinux

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

RAMspeed SMP 3.5.0:
    pts/ramspeed-1.4.2 [Type: Add - Benchmark: Integer]
    Test 1 of 1
    Estimated Trial Run Count:    1
    Estimated Time To Completion: 11 Minutes [09:55 CET]
        Started Run 1 @ 09:45:00_
```

Y aquí están los resultados del test en Ubuntu Server y en CentOS, respectivamente:

```
RAMspeed SMP 3.5.0:
    pts/ramspeed-1.4.2 [Type: Add - Benchmark: Integer]
    Test 1 of 1
    Estimated Trial Run Count:    1
    Estimated Time To Completion: 11 Minutes
        Started Run 1 @ 09:45:01

    Test Results:
        7242.57

    Average: 7242.57 MB/s

jsr@ubuntu:~$
```

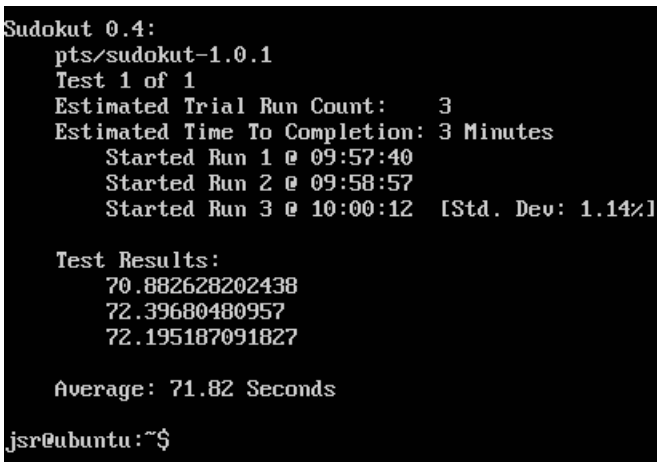
```
RAMspeed SMP 3.5.0:
    pts/ramspeed-1.4.2 [Type: Add - Benchmark: Integer]
    Test 1 of 1
    Estimated Trial Run Count:    1
    Estimated Time To Completion: 11 Minutes [09:55 CET]
        Started Run 1 @ 09:45:00

    Type: Add - Benchmark: Integer:
        7398.81

    Average: 7398.81 MB/s

    Result compared to 3,008 OpenBenchmarking.org samples since 2 January 2000; median result: 16795
. Box plot of samples:
    [      |------------*----####################*!#################----------*-------*---| *      ]
                        ^ This Result (17th Percentile): 7399
                2 x 8192 MB DDR4-2133MHz: 16165 ^ 8 x 8192 MB DDR4-2400MHz Samsung: 31119 ^
                                                  12 x 8192 MB DDR4-2666MT: 28949 ^
                                          8 x 16384 MB DDR4-1866MHz: 26398 ^

[root@localhost jrodriguez]# _
```

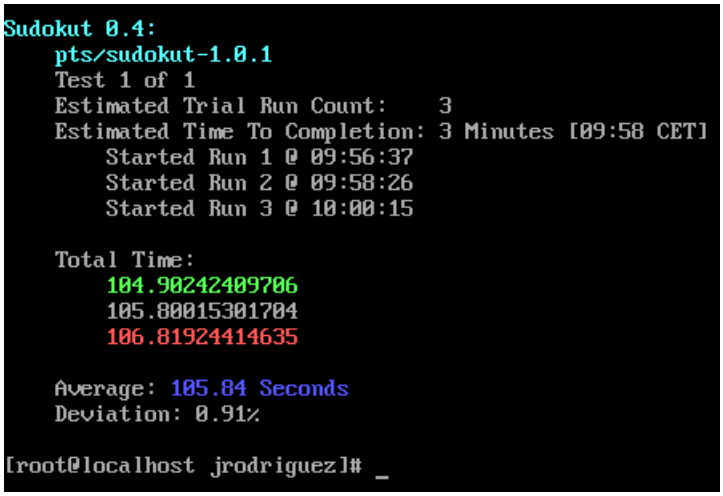
El promedio de velocidad de nuestra memoria RAM en Ubuntu Server es de 7242.57 MB/s y en CentOS es de 7398.81 MB/s por lo que podemos comprobar que la de CentOS es más alta.

Ahora para hacer el test “sudokut” ejecutamos en ambas máquinas la instrucción (este test es bastante más corto):

$> phoronix-test-suite benchmark pts/sudokut

Aquí están los resultados del test en Ubuntu server y CentOS respectivamente:





Sudokut es una prueba donde se solucionan rompecabezas de Sudokus en Tcl, mide el tiempo que tardan en resolver los mismos. En Ubuntu Server vemos que tarda 71.82 segundos mientras que en CentOS tarda 105.84, podemos ver que Ubuntu Server lo hace mejor.

Ahora ejecutaremos los test desde un contenedor Docker, para ello debemos instalar Docker

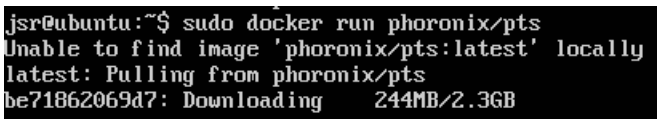
```
jsr@ubuntu:~$ sudo apt install docker.io
```

Después usamos:

$> sudo systemctl start docker

$> sudo systemctl enable docker

Ahora debemos descargar y arrancar el contenedor de phoronix:



Y por último ejecutamos los tests.

Apache Benchmark

En nuestro caso usamos las siguientes instrucciones desde el HOST:

Para Ubuntu Server: Host $> ab -n 1000 -c 100 192.168.56.105/

Para CentOS: Host $> ab -n 1000 -c 100 192.168.56.110/

Los resultados mostrados serán los siguientes:

Ubuntu server

```
Benchmarking 192.168.56.105 (be patient)
Completed 100 requests
Completed 200 requests
Completed 300 requests
Completed 400 requests
Completed 500 requests
Completed 600 requests
Completed 700 requests
Completed 800 requests
Completed 900 requests
Completed 1000 requests
Finished 1000 requests


Server Software:        Apache/2.4.18
Server Hostname:        192.168.56.105
Server Port:            80

Document Path:          /
Document Length:        11321 bytes

Concurrency Level:      100
Time taken for tests:   0.941 seconds
Complete requests:      1000
Failed requests:        0
Total transferred:      11595000 bytes
HTML transferred:       11321000 bytes
Requests per second:    1062.49 [#/sec] (mean)
Time per request:       94.119 [ms] (mean)
Time per request:       0.941 [ms] (mean, across all concurrent requests)
Transfer rate:          12030.81 [Kbytes/sec] received

Connection Times (ms)
              min  mean[+/-sd] median   max
Connect:        0    0   1.1      0      13
Processing:    52   87  17.1     83     170
Waiting:        0   86  16.6     82     163
Total:         53   88  17.2     83     170

Percentage of the requests served within a certain time (ms)
  50%     83
  66%     90
  75%     97
  80%    100
  90%    114
  95%    121
  98%    126
  99%    134
 100%    170 (longest request)
```

CentOS

```
Benchmarking 192.168.56.110 (be patient)
Completed 100 requests
Completed 200 requests
Completed 300 requests
Completed 400 requests
Completed 500 requests
Completed 600 requests
Completed 700 requests
Completed 800 requests
Completed 900 requests
Completed 1000 requests
Finished 1000 requests


Server Software:        Apache/2.4.6
Server Hostname:        192.168.56.110
Server Port:            80

Document Path:          /
Document Length:        694 bytes

Concurrency Level:      100
Time taken for tests:   0.979 seconds
Complete requests:      1000
Failed requests:        0
Total transferred:      876000 bytes
HTML transferred:       694000 bytes
Requests per second:    1021.18 [#/sec] (mean)
Time per request:       97.926 [ms] (mean)
Time per request:       0.979 [ms] (mean, across all concurrent requests)
Transfer rate:          873.58 [Kbytes/sec] received

Connection Times (ms)
              min  mean[+/-sd] median   max
Connect:        0    0   0.8      0      10
Processing:    55   92  18.2     84     221
Waiting:        1   92  17.8     84     132
Total:         55   92  18.3     84     221

Percentage of the requests served within a certain time (ms)
  50%     84
  66%    101
  75%    108
  80%    112
  90%    118
  95%    120
  98%    128
  99%    130
 100%    221 (longest request)
```

ApacheBenchmark nos sirve para medir el rendimiento de nuestro servidor generando una serie de llamadas a las diferentes IPs distribuidas en hilos en nuestro caso estamos haciendo 1000 llamadas en 100 hilos. En los resultados debemos fijarnos en la columna “mean”, fila “Total”, donde podemos ver la suma de los tiempos de conexión y procesamiento. Podemos ver que en Ubuntu Server es 88ns y en CentOS 92ns, por lo que en Ubuntu Server es menor.

Sesión 2:

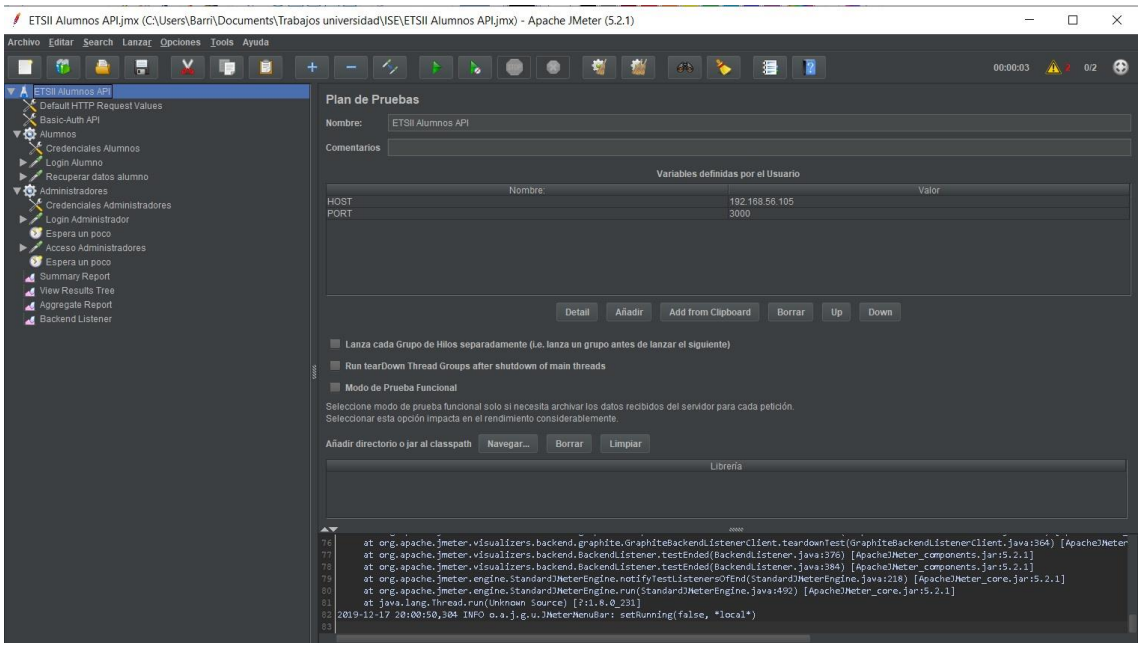
En esta sesión debemos instalar y usar JMeter, debemos instalarlo en nuestro Host, y en nuestra máquina virtual usar el siguiente comando para obtener el repositorio concreto de esta práctica:

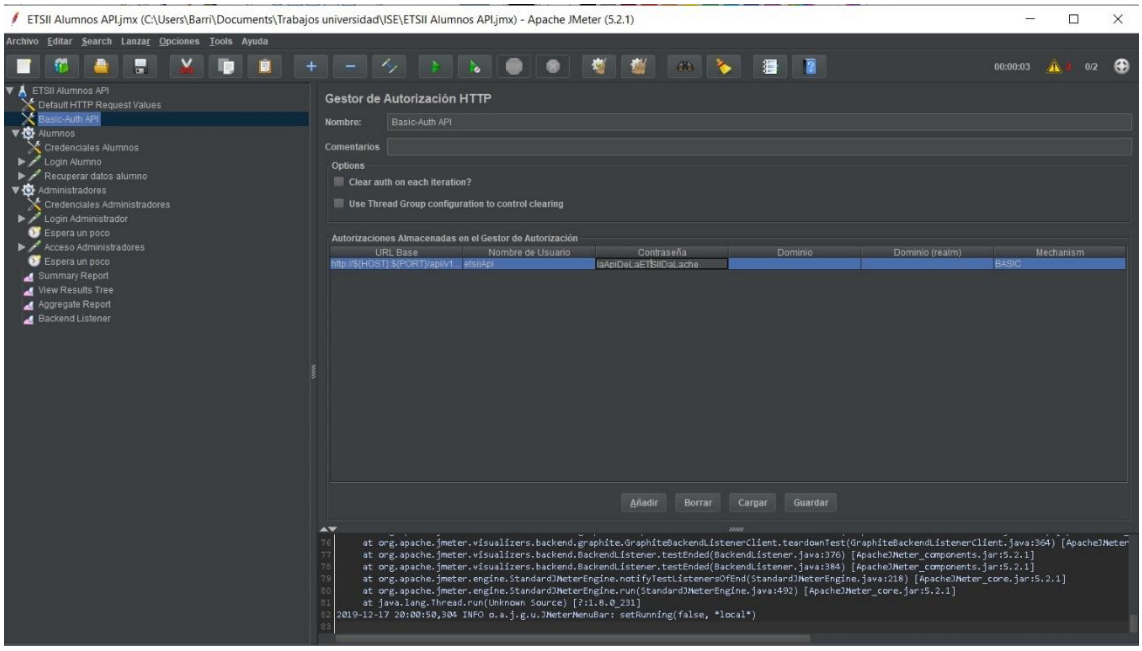
```
jsr@ubuntu:~$ git clone https://github.com/davidPalomar-ugr/iseP4JMeter.git
Clonar en «iseP4JMeter»...
remote: Enumerating objects: 3766, done.
remote: Total 3766 (delta 0), reused 0 (delta 0), pack-reused 3766
Receiving objects: 100% (3766/3766), 7.75 MiB | 1.69 MiB/s, done.
Resolving deltas: 100% (701/701), done.
Comprobando la conectividad… hecho.
Extrayendo archivos: 100% (72/72), done.
```

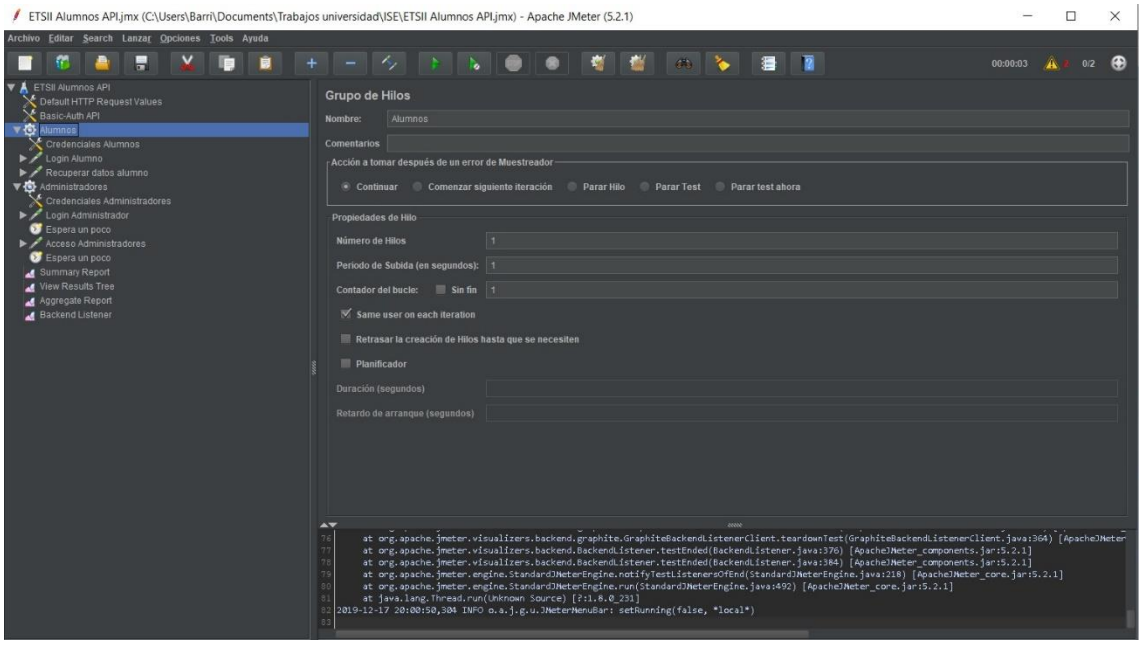
Después debemos comprobar que tenemos instalado “docker-compose”, en el caso de que no lo tengamos debemos instalarlo y usar los siguientes comandos (el “-d” nos sirve para ejecutarlo en segundo plano y así tener a mano el terminal):

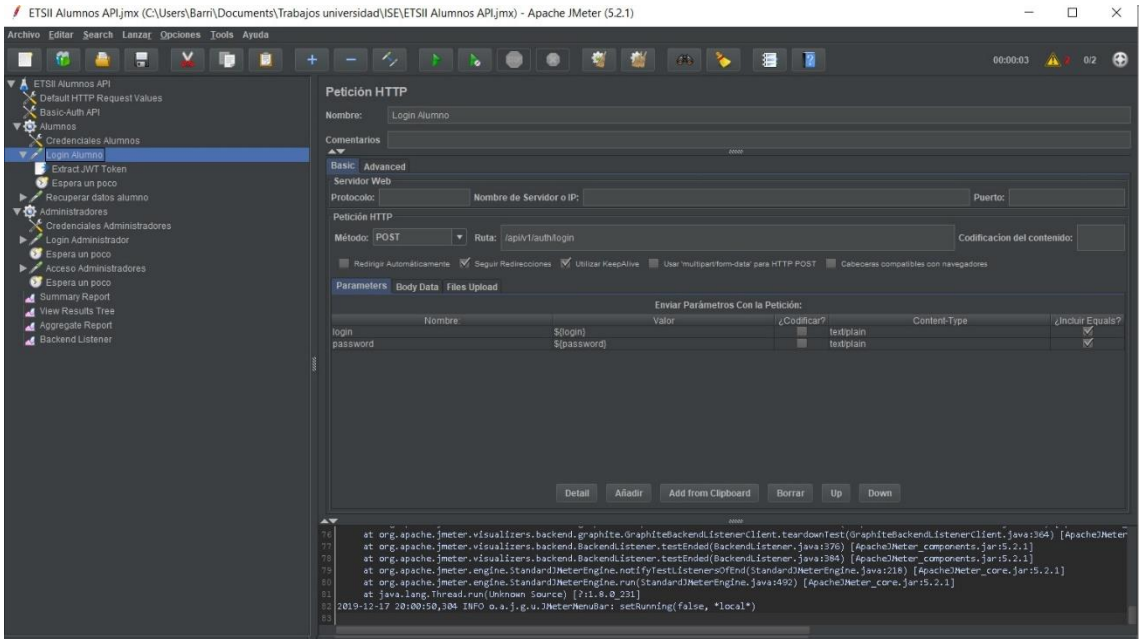
```
jsr@ubuntu:~$ cd iseP4JMeter/
jsr@ubuntu:~/iseP4JMeter$ sudo docker-compose up -d_
```

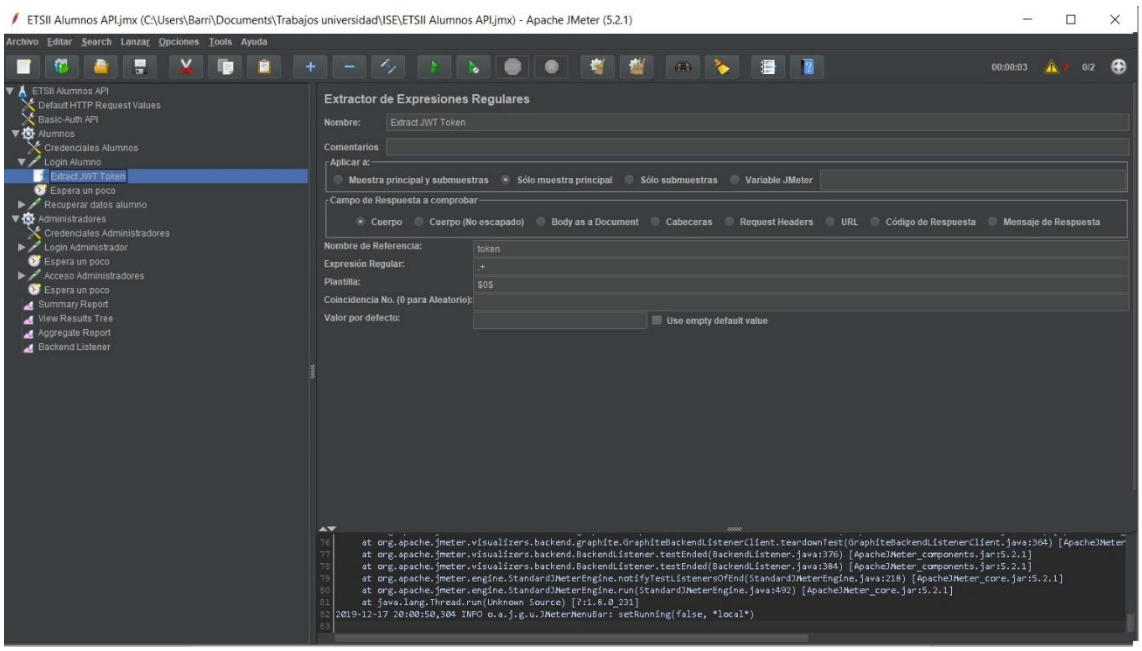
Una vez hecho esto descargamos en nuestro Host los archivos y abrimos el JMeter, después debemos conseguir la estructura explicada en el manual de la práctica encontrado en Github. En la última clase defendí la misma así que no considero necesario poner absolutamente todas las pantallas y he capturado las más importantes y la solución final.

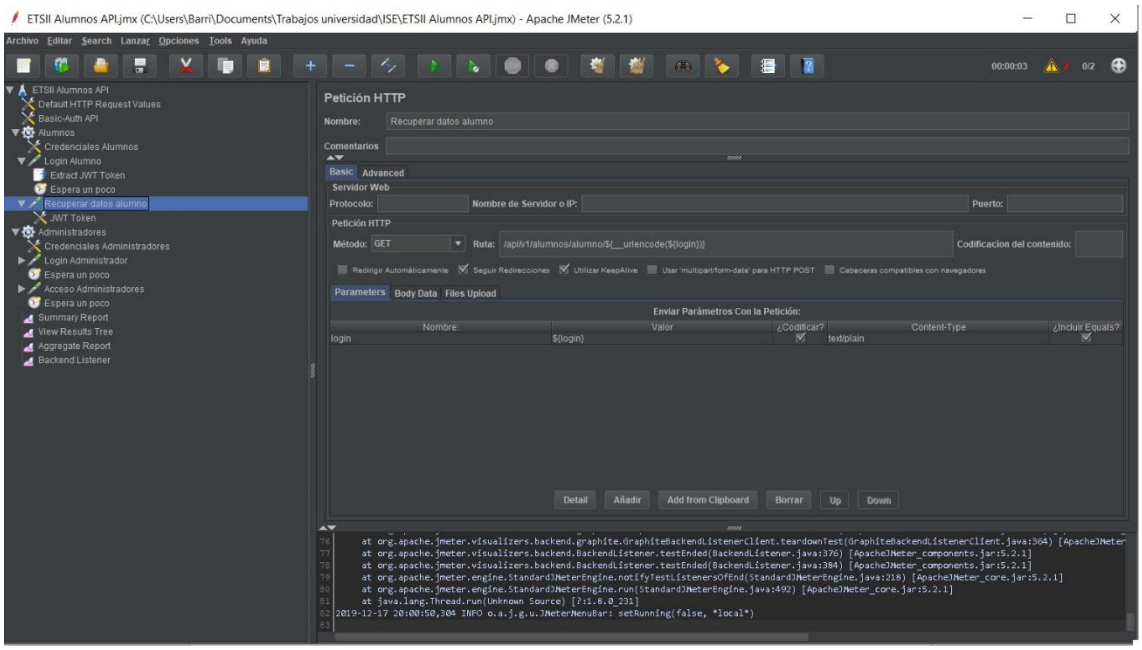


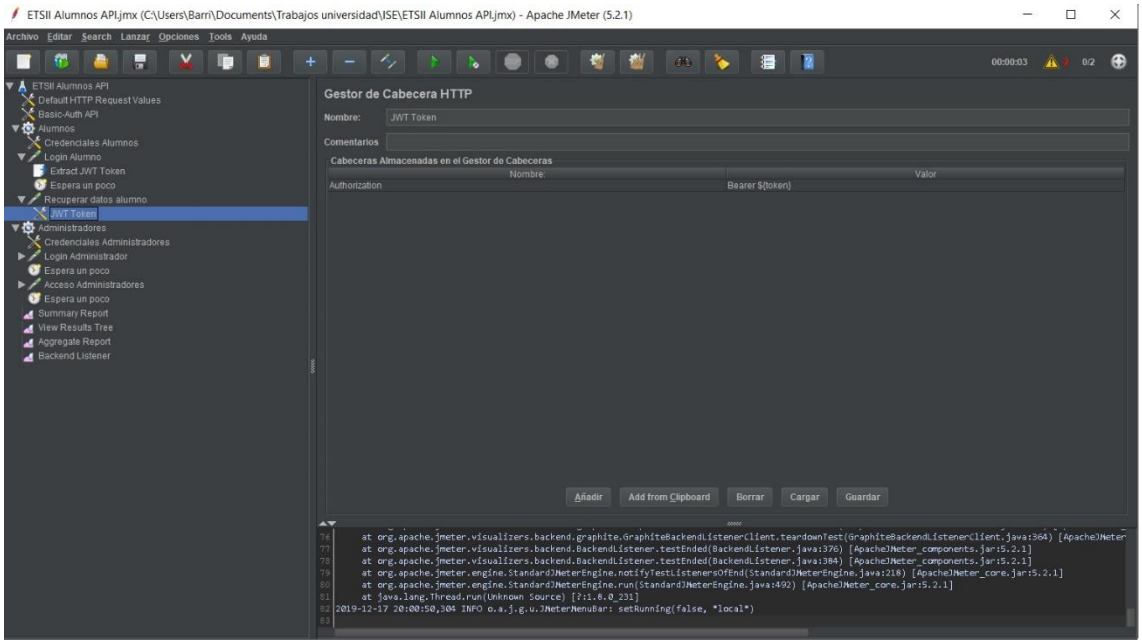

ETSII Alumnos API.jmx (C:\Users\Barri\Documents\Trabajos universidad\ISE\ETSII Alumnos API.jmx) - Apache JMeter (5.2.1)
Archivo Editar Search Lanzar Opciones Tools Ayuda
ETSII Alumnos API
Default HTTP Request Values
Basic-Auth API
Alumnos
Credenciales Alumnos
Login Alumno
Recuperar datos alumno
Administradores
Credenciales Administradores
Login Administrador
Espera un poco
Acceso Administradores
Espera un poco
Summary Report
View Results Tree
Aggregate Report
Backend Listener
Gestor de Autorización HTTP
Nombre: Basic-Auth API
Comentarios
Options
Clear auth on each iteration?
Use Thread Group configuration to control clearing
Autorizaciones Almacenadas en el Gestor de Autorización
URL Base
Nombre de Usuario
Contraseña
Dominio
Dominio (realm)
Mechanism
http://${HOST}:${PORT}/apiv1
etsiiApi
laApiDeLaETSIIDaLache
BASIC
Añadir
Borrar
Cargar
Guardar
at org.apache.jmeter.visualizers.backend.graphite.GraphiteBackendListenerClient.teardownTest(GraphiteBackendListenerClient.java:364) [ApacheJMeter
at org.apache.jmeter.visualizers.backend.BackendListener.testEnded(BackendListener.java:376) [ApacheJMeter_components.jar:5.2.1]
at org.apache.jmeter.visualizers.backend.BackendListener.testEnded(BackendListener.java:384) [ApacheJMeter_components.jar:5.2.1]
at org.apache.jmeter.engine.StandardJMeterEngine.notifyTestListenersOfEnd(StandardJMeterEngine.java:218) [ApacheJMeter_core.jar:5.2.1]
at org.apache.jmeter.engine.StandardJMeterEngine.run(StandardJMeterEngine.java:492) [ApacheJMeter_core.jar:5.2.1]
at java.lang.Thread.run(Unknown Source) [?:1.8.0_231]
2019-12-17 20:00:50,304 INFO o.a.j.g.u.JMeterMenuBar: setRunning(false, *local*)

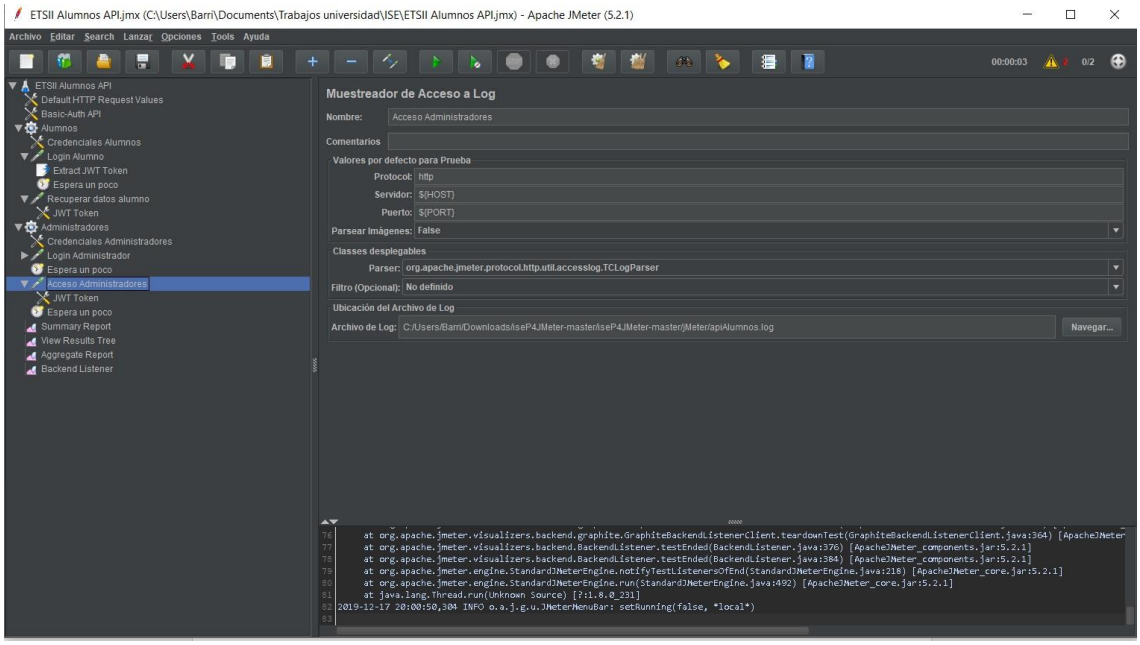

ETSII Alumnos API.jmx (C:\Users\Barri\Documents\Trabajos universidad\ISE\ETSII Alumnos API.jmx) - Apache JMeter (5.2.1)
Archivo Editar Search Lanzar Opciones Tools Ayuda
ETSII Alumnos API
Default HTTP Request Values
Basic-Auth API
Alumnos
Credenciales Alumnos
Login Alumno
Recuperar datos alumno
Administradores
Credenciales Administradores
Login Administrador
Espera un poco
Acceso Administradores
Espera un poco
Summary Report
View Results Tree
Aggregate Report
Backend Listener
Grupo de Hilos
Nombre: Alumnos
Comentarios
Acción a tomar después de un error de Muestreador
Continuar
Comenzar siguiente iteración
Parar Hilo
Parar Test
Parar test ahora
Propiedades de Hilo
Número de Hilos 1
Periodo de Subida (en segundos): 1
Contador del bucle: Sin fin 1
Same user on each iteration
Retrasar la creación de Hilos hasta que se necesiten
Planificador
Duración (segundos)
Retardo de arranque (segundos)
at org.apache.jmeter.visualizers.backend.graphite.GraphiteBackendListenerClient.teardownTest(GraphiteBackendListenerClient.java:364) [ApacheJMeter
at org.apache.jmeter.visualizers.backend.BackendListener.testEnded(BackendListener.java:376) [ApacheJMeter_components.jar:5.2.1]
at org.apache.jmeter.visualizers.backend.BackendListener.testEnded(BackendListener.java:384) [ApacheJMeter_components.jar:5.2.1]
at org.apache.jmeter.engine.StandardJMeterEngine.notifyTestListenersOfEnd(StandardJMeterEngine.java:218) [ApacheJMeter_core.jar:5.2.1]
at org.apache.jmeter.engine.StandardJMeterEngine.run(StandardJMeterEngine.java:492) [ApacheJMeter_core.jar:5.2.1]
at java.lang.Thread.run(Unknown Source) [?:1.8.0_231]
2019-12-17 20:00:50,304 INFO o.a.j.g.u.JMeterMenuBar: setRunning(false, *local*)


ETSII Alumnos API.jmx (C:\Users\Barri\Documents\Trabajos universidad\ISE\ETSII Alumnos API.jmx) - Apache JMeter (5.2.1)
Archivo Editar Search Lanzar Opciones Tools Ayuda
ETSII Alumnos API
Default HTTP Request Values
Basic-Auth API
Alumnos
Credenciales Alumnos
Login Alumno
Extract JWT Token
Espera un poco
Recuperar datos alumno
Administradores
Credenciales Administradores
Login Administrador
Espera un poco
Acceso Administradores
Espera un poco
Summary Report
View Results Tree
Aggregate Report
Backend Listener
Petición HTTP
Nombre: Login Alumno
Comentarios
Basic Advanced
Servidor Web
Protocolo: Nombre de Servidor o IP: Puerto:
Petición HTTP
Método: POST Ruta: /api/v1/auth/login Codificacion 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 ${login} text/plain
password ${password} text/plain
Detail Añadir Add from Clipboard Borrar Up Down
at org.apache.jmeter.visualizers.backend.graphite.GraphiteBackendListenerClient.teardownTest(GraphiteBackendListenerClient.java:364) [ApacheJMeter
at org.apache.jmeter.visualizers.backend.BackendListener.testEnded(BackendListener.java:376) [ApacheJMeter_components.jar:5.2.1]
at org.apache.jmeter.visualizers.backend.BackendListener.testEnded(BackendListener.java:384) [ApacheJMeter_components.jar:5.2.1]
at org.apache.jmeter.engine.StandardJMeterEngine.notifyTestListenersOfEnd(StandardJMeterEngine.java:218) [ApacheJMeter_core.jar:5.2.1]
at org.apache.jmeter.engine.StandardJMeterEngine.run(StandardJMeterEngine.java:492) [ApacheJMeter_core.jar:5.2.1]
at java.lang.Thread.run(Unknown Source) [?:1.8.0_231]
2019-12-17 20:00:50,304 INFO o.a.j.g.u.JMeterMenuBar: setRunning(false, *local*)

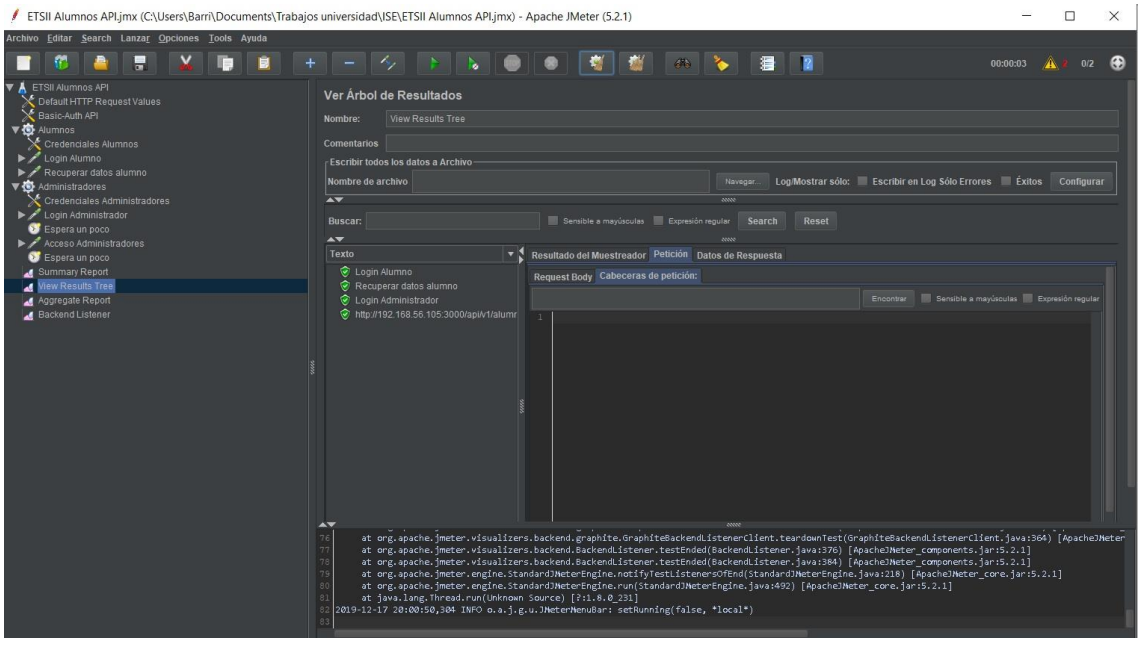

ETSII Alumnos API.jmx (C:\Users\Barri\Documents\Trabajos universidad\ISE\ETSII Alumnos API.jmx) - Apache JMeter (5.2.1)
Archivo Editar Search Lanzar Opciones Tools Ayuda
ETSII Alumnos API
Default HTTP Request Values
Basic-Auth API
Alumnos
Credenciales Alumnos
Login Alumno
Extract JWT Token
Espera un poco
Recuperar datos alumno
Administradores
Credenciales Administradores
Login Administrador
Espera un poco
Acceso Administradores
Espera un poco
Summary Report
View Results Tree
Aggregate Report
Backend Listener
Extractor de Expresiones Regulares
Nombre: Extract JWT Token
Comentarios
Aplicar a:
Muestra principal y submuestras Sólo muestra principal Sólo submuestras Variable JMeter
Campo de Respuesta a comprobar
Cuerpo Cuerpo (No escapado) Body as a Document Cabeceras Request Headers URL Código de Respuesta Mensaje de Respuesta
Nombre de Referencia: token
Expresión Regular: .+
Plantilla: 0
Coincidencia No. (0 para Aleatorio):
Valor por defecto:
Use empty default value
at org.apache.jmeter.visualizers.backend.graphite.GraphiteBackendListenerClient.teardownTest(GraphiteBackendListenerClient.java:364) [ApacheJMeter
at org.apache.jmeter.visualizers.backend.BackendListener.testEnded(BackendListener.java:376) [ApacheJMeter_components.jar:5.2.1]
at org.apache.jmeter.visualizers.backend.BackendListener.testEnded(BackendListener.java:384) [ApacheJMeter_components.jar:5.2.1]
at org.apache.jmeter.engine.StandardJMeterEngine.notifyTestListenersOfEnd(StandardJMeterEngine.java:218) [ApacheJMeter_core.jar:5.2.1]
at org.apache.jmeter.engine.StandardJMeterEngine.run(StandardJMeterEngine.java:492) [ApacheJMeter_core.jar:5.2.1]
at java.lang.Thread.run(Unknown Source) [?:1.8.0_231]
2019-12-17 20:00:50,304 INFO o.a.j.g.u.JMeterMenuBar: setRunning(false, *local*)


ETSII Alumnos API.jmx (C:\Users\Barri\Documents\Trabajos universidad\ISE\ETSII Alumnos API.jmx) - Apache JMeter (5.2.1)
Archivo Editar Search Lanzar Opciones Tools Ayuda
ETSII Alumnos API
Default HTTP Request Values
Basic-Auth API
Alumnos
Credenciales Alumnos
Login Alumno
Extract JWT Token
Espera un poco
Recuperar datos alumno
JWT Token
Administradores
Credenciales Administradores
Login Administrador
Espera un poco
Acceso Administradores
Espera un poco
Summary Report
View Results Tree
Aggregate Report
Backend Listener
Petición HTTP
Nombre: Recuperar datos alumno
Comentarios
Basic Advanced
Servidor Web
Protocolo: Nombre de Servidor o IP: Puerto:
Petición HTTP
Método: GET Ruta: /api/v1/alumnos/alumno/${__urlencode(${login})} Codificacion 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 ${login} text/plain
Detail Añadir Add from Clipboard Borrar Up Down
at org.apache.jmeter.visualizers.backend.graphite.GraphiteBackendListenerClient.teardownTest(GraphiteBackendListenerClient.java:364) [ApacheJMeter
at org.apache.jmeter.visualizers.backend.BackendListener.testEnded(BackendListener.java:376) [ApacheJMeter_components.jar:5.2.1]
at org.apache.jmeter.visualizers.backend.BackendListener.testEnded(BackendListener.java:384) [ApacheJMeter_components.jar:5.2.1]
at org.apache.jmeter.engine.StandardJMeterEngine.notifyTestListenersOfEnd(StandardJMeterEngine.java:218) [ApacheJMeter_core.jar:5.2.1]
at org.apache.jmeter.engine.StandardJMeterEngine.run(StandardJMeterEngine.java:492) [ApacheJMeter_core.jar:5.2.1]
at java.lang.Thread.run(Unknown Source) [?:1.8.0_231]
2019-12-17 20:00:50,304 INFO o.a.j.g.u.JMeterMenuBar: setRunning(false, *local*)


ETSII Alumnos API.jmx (C:\Users\Barri\Documents\Trabajos universidad\ISE\ETSII Alumnos API.jmx) - Apache JMeter (5.2.1)
Archivo Editar Search Lanzar Opciones Tools Ayuda
ETSII Alumnos API
Default HTTP Request Values
Basic-Auth API
Alumnos
Credenciales Alumnos
Login Alumno
Extract JWT Token
Espera un poco
Recuperar datos alumno
JWT Token
Administradores
Credenciales Administradores
Login Administrador
Espera un poco
Acceso Administradores
Espera un poco
Summary Report
View Results Tree
Aggregate Report
Backend Listener
Gestor de Cabecera HTTP
Nombre: JWT Token
Comentarios
Cabeceras Almacenadas en el Gestor de Cabeceras
Nombre: Valor
Authorization Bearer ${token}
Añadir Add from Clipboard Borrar Cargar Guardar
at org.apache.jmeter.visualizers.backend.graphite.GraphiteBackendListenerClient.teardownTest(GraphiteBackendListenerClient.java:364) [ApacheJMeter
at org.apache.jmeter.visualizers.backend.BackendListener.testEnded(BackendListener.java:376) [ApacheJMeter_components.jar:5.2.1]
at org.apache.jmeter.visualizers.backend.BackendListener.testEnded(BackendListener.java:384) [ApacheJMeter_components.jar:5.2.1]
at org.apache.jmeter.engine.StandardJMeterEngine.notifyTestListenersOfEnd(StandardJMeterEngine.java:218) [ApacheJMeter_core.jar:5.2.1]
at org.apache.jmeter.engine.StandardJMeterEngine.run(StandardJMeterEngine.java:492) [ApacheJMeter_core.jar:5.2.1]
at java.lang.Thread.run(Unknown Source) [?:1.8.0_231]
2019-12-17 20:00:50,304 INFO o.a.j.g.u.JMeterMenuBar: setRunning(false, *local*)



Y el resultado final, sin ningún error:



Referencias:

https://phoenixnap.com/kb/how-to-install-docker-on-ubuntu-18-04

https://octoperf.com/blog/2018/04/16/how-to-install-jmeter-windows/

https://stackoverflow.com/questions/6299948/how-to-run-apache-benchmark-load-test-in-windows

https://github.com/davidPalomar-ugr/iseP4JMeter/blob/master/README.md