



UNIVERSIDAD DE GRANADA

Práctica 4

INGENIERÍA DE SERVIDORES

Lukas Häring García 3º C

Contents

1	Phoronix	2
2	JMeter	4
2.1	Instalación en Ubuntu Server	4
2.2	Setup de JMeter	5
3	Referencias Bibliográficas	11

1 Phoronix

Phoronix Test Suite es la plataforma de pruebas y evaluación comparativa más completa disponible para Linux

El primer test, es para la CPU, consiste en resolver un número de sudokus y devuelve la media de los tiempos.

El tiempo medio obtenido es de 25 segundos aproximadamente en nuestra máquina virtual de Ubuntu.

```
el 440FX- 82441FX PMC, Memory: 4096MB, Disk: 2 x 11GB VBOX HDD, Graphics: VMware SVGA II, Audio: Intel  
el 82801AA AC 97 Audio, Network: Intel 82540EM Gigabit  
  
Software:  
OS: Ubuntu 16.04, Kernel: 4.4.0-131-generic (x86_64), File-System: ext4, Screen Resolution: 2048x2048, System Layer: Oracle VMware  
  
Would you like to save these test results (Y/n): Y  
Enter a name to save these results under: sudoku_test  
Enter a unique name to describe this test run / configuration: sudoku  
  
If you wish, 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 16.04 via the Phoronix Test Suite.  
New Description:  
  
Sudoku 0.4:  
pts/sudoku-1.0.1  
Test 1 of 1  
Estimated Trial Run Count: 3  
Estimated Time To Completion: 3 Minutes  
Started Run 1 @ 18:27:12  
Started Run 2 @ 18:27:43  
Started Run 3 @ 18:28:10 [Std. Dev: 1.54%]  
  
Test Results:  
24.387434959412  
25.117774963379  
24.9432117939  
  
Average: 24.82 Seconds  
  
Would you like to upload the results to OpenBenchmarking.org (Y/n): _
```

Figure 1: Test CPU, Resolución de sudoku

El segundo test se trata de introducir una gran cantidad de enteros en la RAM y sacarlos, así comprobar la velocidad de nuestra memoria (ancho de banda).

El test es realizado una única vez, con un tiempo estimado de 11 minutos.

El resultado obtenido fue de 9052.47 MB/s de ancho de banda.

```
Recently Saved Test Results:
- sudokuttest [Today]

    Enter a name to save these results under: ranspeed

The name of the saved file cannot be the same as a test/suite: ranspeed
    Enter a name to save these results under: ranspeed

The name of the saved file cannot be the same as a test/suite: ranspeed
    Enter a name to save these results under: ranspeedtest
    Enter a unique name to describe this test run / configuration: ranspeed

If you wish, 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 16.04 via the Phoronix Test Suite.

New Description:

RAMspeed SMP 3.5.0:
  pts/ranspeed-1.4.2 [Type: Average - Benchmark: Integer]
  Test 1 of 1
  Estimated Trial Run Count:    1
  Estimated Time To Completion: 11 Minutes
  Started Run 1 @ 18:35:22

  Test Results:
    9052.47

  Average: 9052.47 MB/s

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

lhg@ubuntu:~$
```

Figure 2: Test memoria RAM, Entrada y salida de valores enteros

2 JMeter

Apache JMeter es un software de código abierto, diseñada para cargar el comportamiento funcional de las pruebas y medir el rendimiento.

2.1 Instalación en Ubuntu Server

Suponiendo que tenemos la **primera práctica** acabada.

Descargamos **docker-compose** en nuestra máquina,

```
> apt install docker-compose
```

Una vez instalado, nos vamos a realizar un **git pull** del proyecto de *David Palomar*.

```
> git clone https://github.com/davidPalomar-ugr/iseP4JMeter.git
```

Ahora tendremos una carpeta llamada "*iseP4JMeter*", tras entrar en ella, podemos ejecutar el *Docker*.

Nota: En diferentes pruebas he tenido que realizar un reinicio previo de la máquina antes de abrir el *Docker*.

```
iseP4JMeter > docker-compose up -d
```

Podemos realizar una prueba previa, que realizará una petición "HTTP" para comprobar que nos devuelve un resultado.

```
iseP4JMeter > sh pruebaEntorno.sh
```

2.2 Setup de JMeter

Se ha realizado la práctica en el sistema operativo **Windows**, pero en realidad no existe ninguna diferencia.

Comentar previamente que se utiliza el click derecho para "Añadir" un nuevo elemento en nuestro entorno. Añadimos previamente la configuración

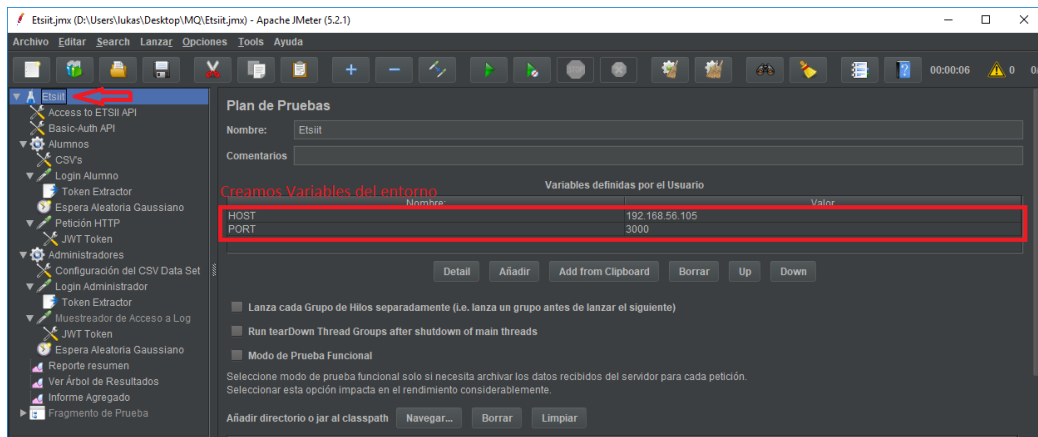


Figure 3: Primer paso, creación de variables del entorno

HTTP de nuestro archivo.

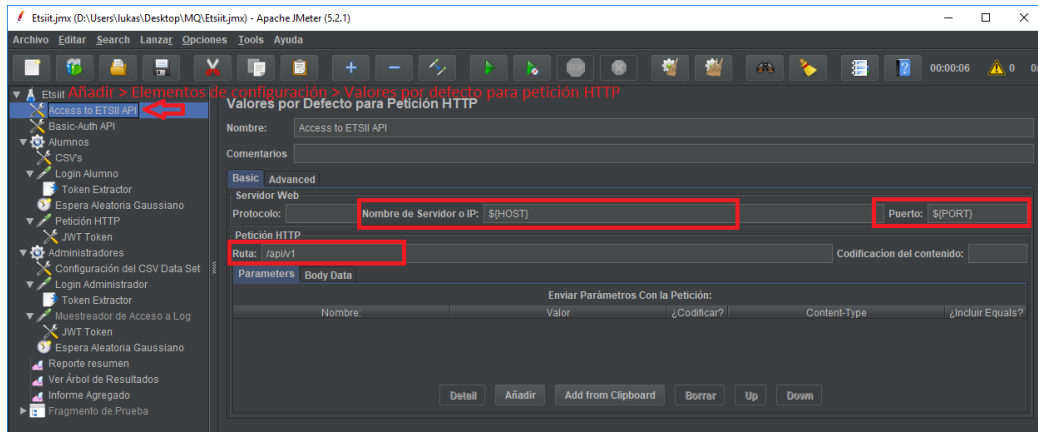


Figure 4: Segundo paso, valores por defecto

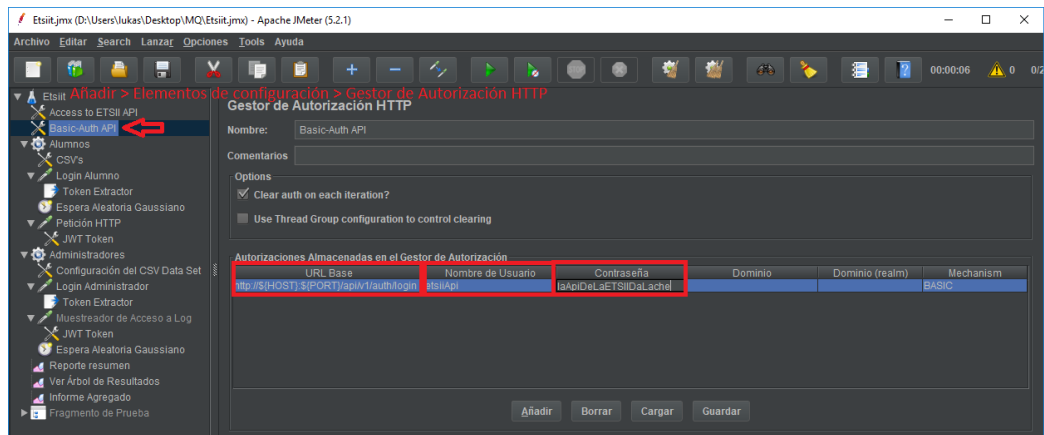


Figure 5: Tercer paso, Autorización HTTP

Creamos **dos grupos de hebras** "Alumnos" y "Administradores", cuyos valores son modificables.

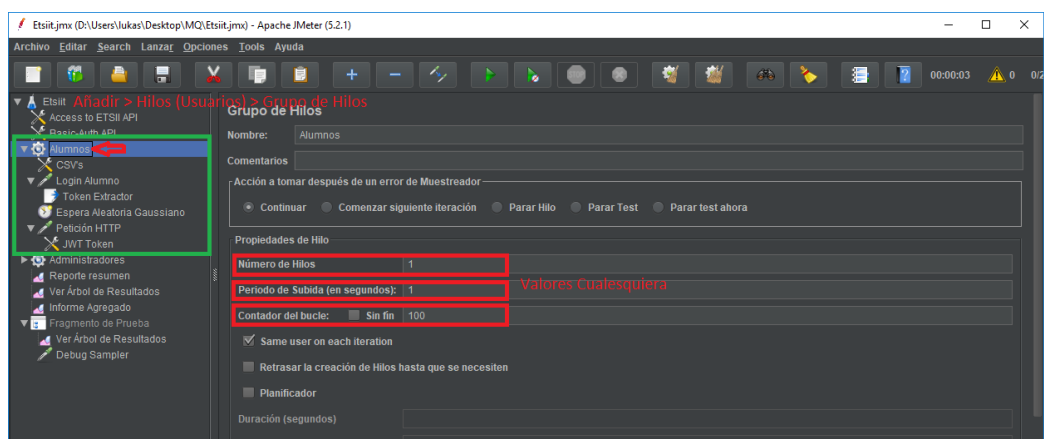


Figure 6: Primer paso, creación de variables del entorno

Asignamos para cada grupo de hebras, su CSV correspondiente.

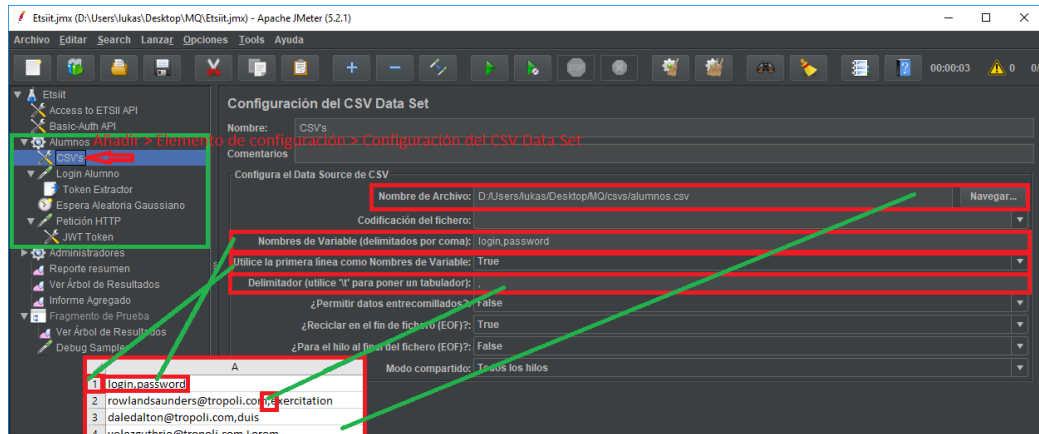


Figure 7: Quinto paso, asignar archivos CSV

Cada entrada del CSV va a realizar una petición HTTP, por lo que vamos a utilizar dichos valores para generarla.

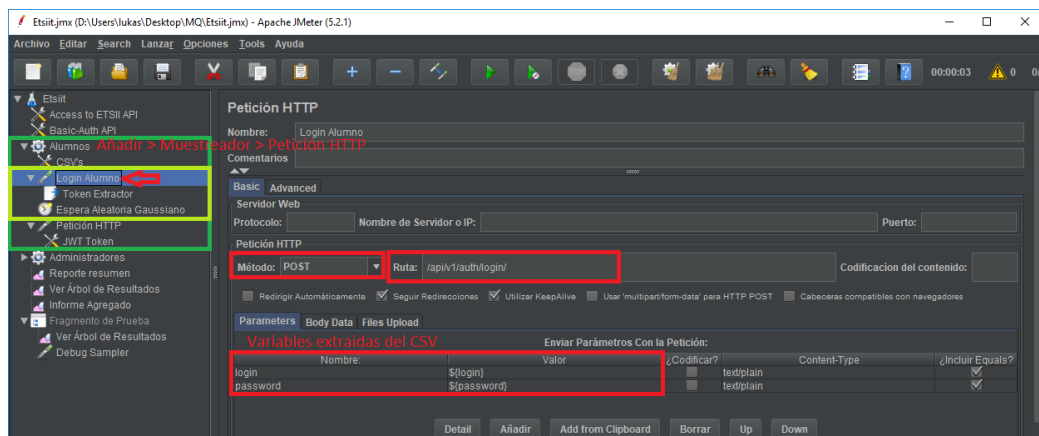


Figure 8: Sexto, Peticiones HTTP de cada muestra

Cada petición a su vez, nos va a devolver un "token" correspondiente.

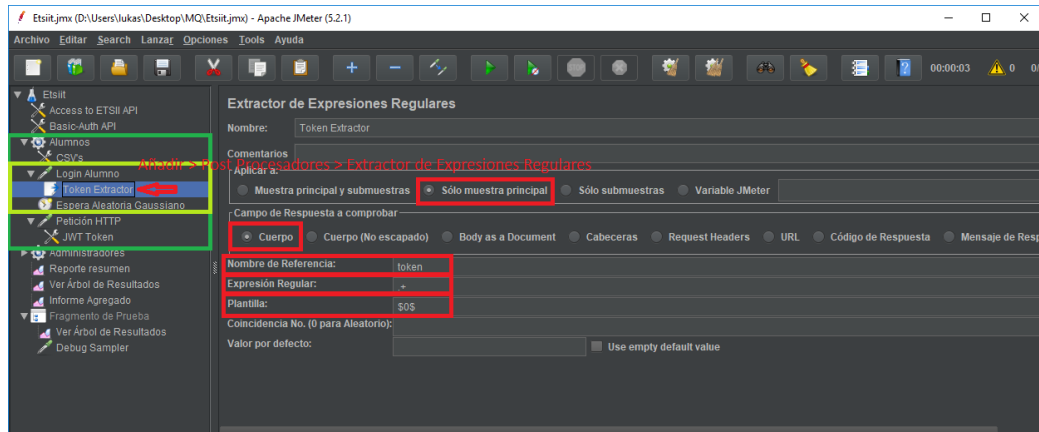


Figure 9: Séptimo, Extracción del token

Se realizará una **espera aleatoria** después de dicha extracción.

Vamos a utilizar el token extraído para extraer finalmente el **token de Acceso** a dicho usuario.

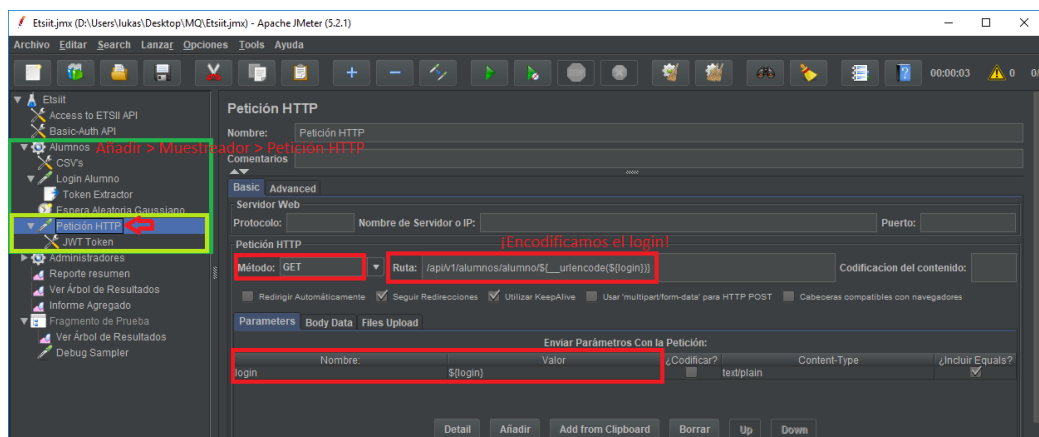


Figure 10: Octavo paso, petición del token JWT

Añadimos a la cabecera la **autenticación** de tipo **Bearer** con el **token** obtenido.

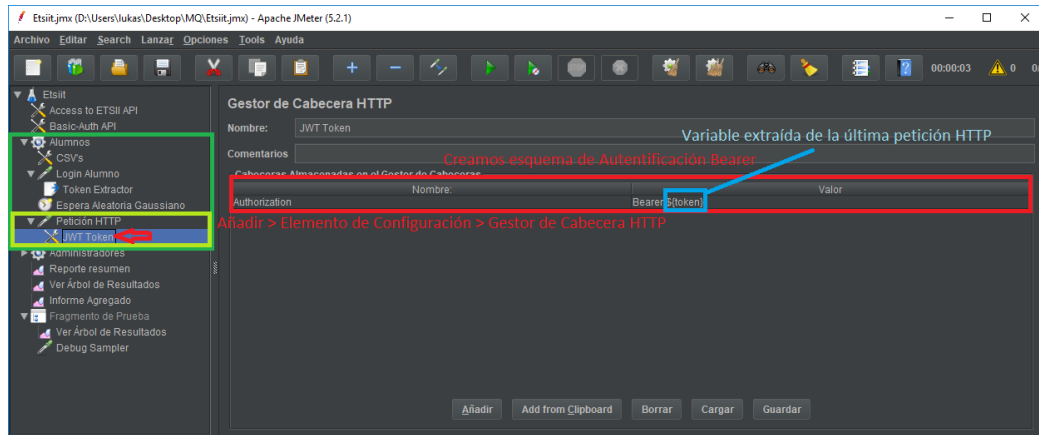


Figure 11: Noveno paso, creación de variables del entorno

Para los administradores, a diferencia de los Alumnos de una única , van a realizar todas las peticiones que tiene el archivo **TCP Log** que vamos a utilizar sobre los Alumnos.

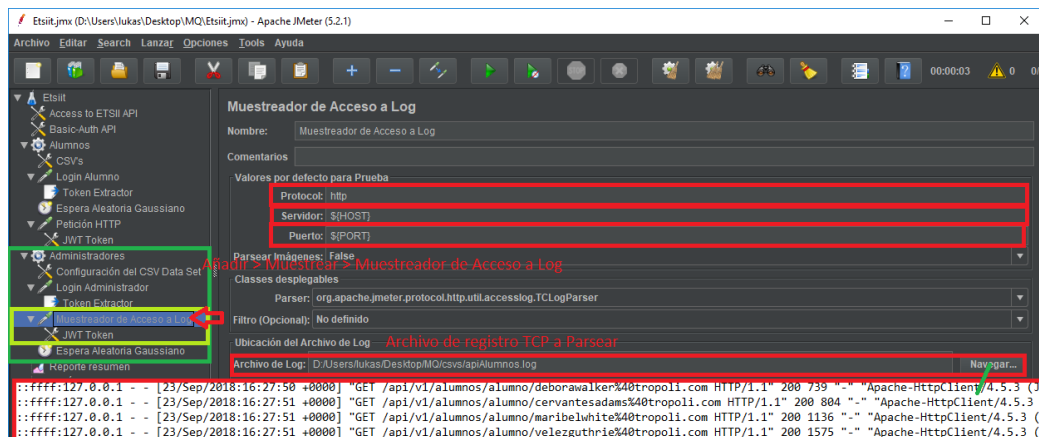


Figure 12: Décimo paso, Muestreo TCP Log

Una vez acabado, podemos Añadir **Receptores** para visualizar los resultados. Podemos comprobar que no ha ocurrido ningún fallo viendo el árbol de peticiones, existen 5 códigos de resolución de una petición

1. **1**** Respuestas informativas
2. **2**** Respuestas satisfactorias
3. **3**** Redirecciones
4. **4**** Errores de cliente
5. **5**** Errores de servidor

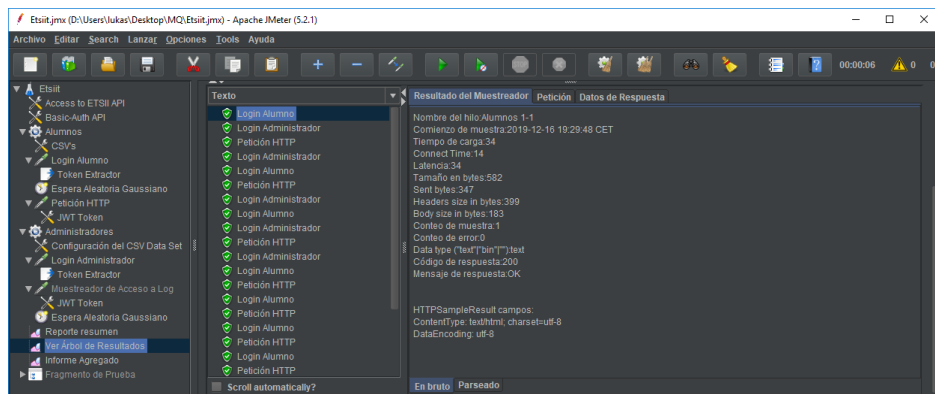


Figure 13: Árbol de peticiones

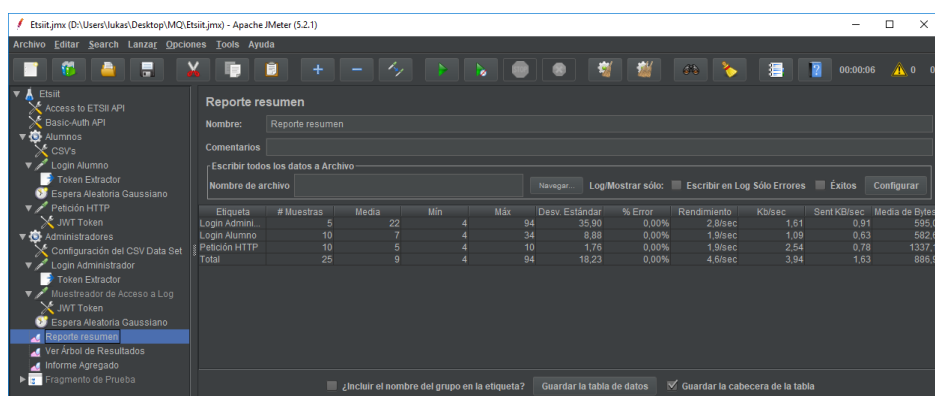


Figure 14: Resumen de las peticiones

3 Referencias Bibliográficas

References

- [1] Códigos de estado de respuesta HTTP <https://developer.mozilla.org/es/docs/Web/HTTP/Status>