



UNIVERSIDAD DE GRANADA

## Práctica 4

INGENIERÍA DE SERVIDORES

*Lukas Häring García 3º C*

# Contents

<b>1</b>	<b>JMeter</b>	<b>2</b>
1.1	Instalación en Ubuntu Server . . . . .	2
1.2	Setup de JMeter . . . . .	3
<b>2</b>	<b>Referencias Bibliográficas</b>	<b>9</b>

# 1 JMeter

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

## 1.1 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
```

## 1.2 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

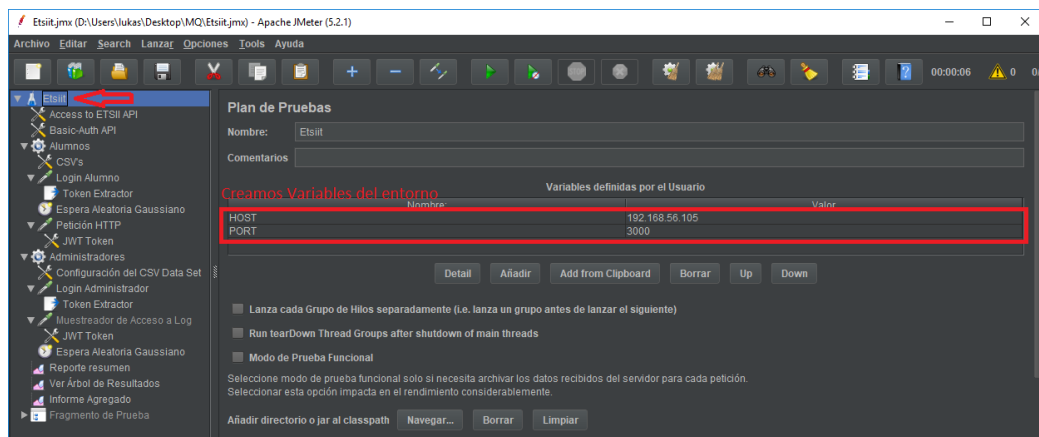


Figure 1: Primer paso, creación de variables del entorno

HTTP de nuestro archivo.

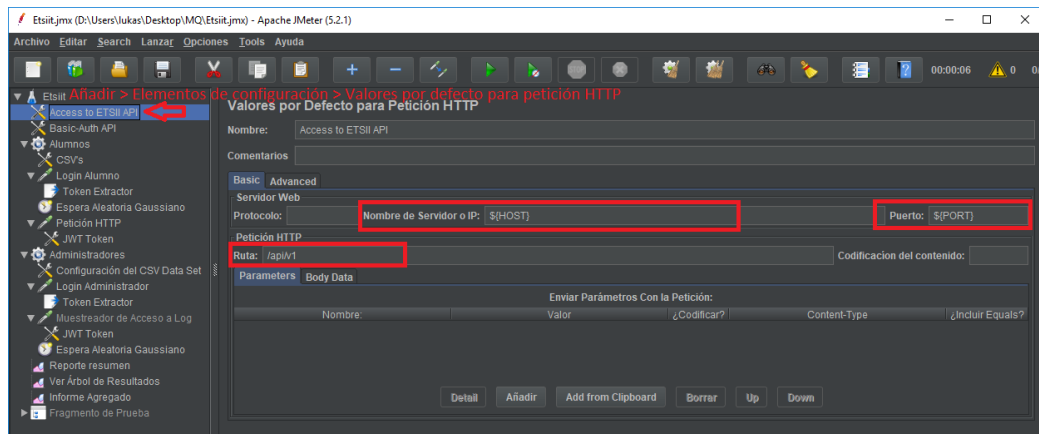


Figure 2: Segundo paso, valores por defecto

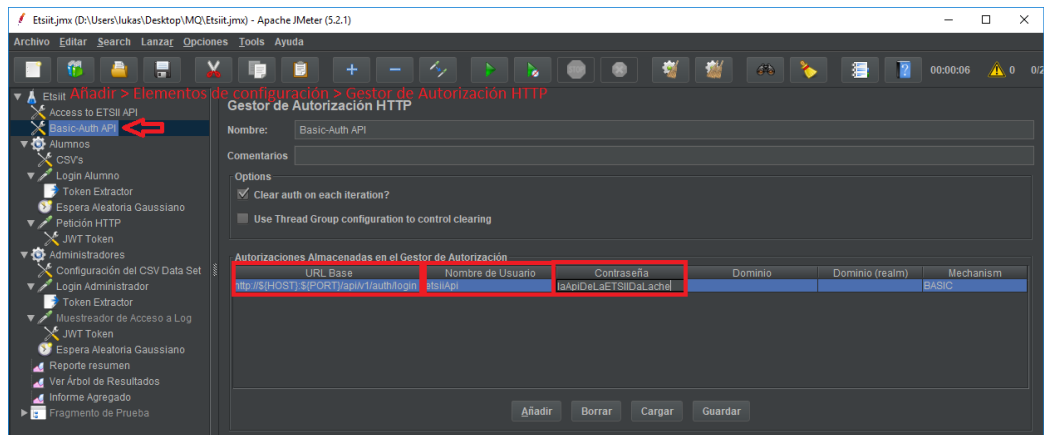


Figure 3: Tercer paso, Autorización HTTP

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

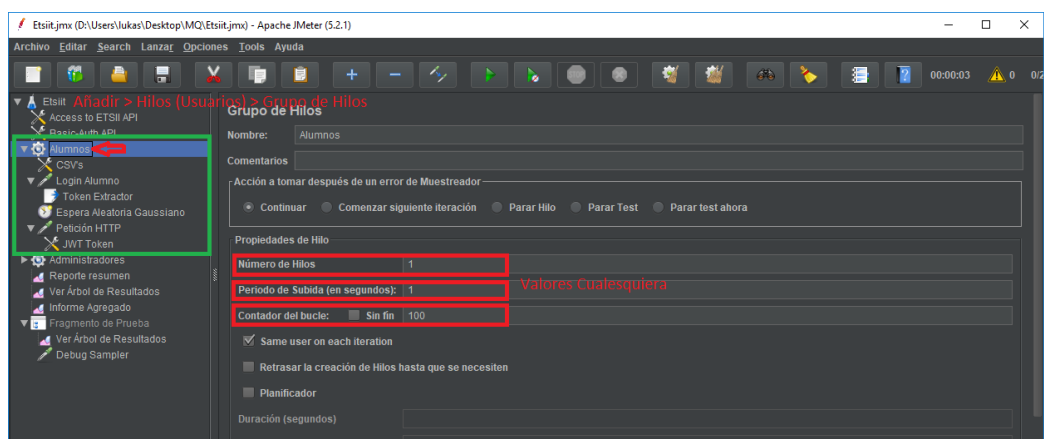


Figure 4: Primer paso, creación de variables del entorno

Asignamos para cada grupo de hebras, su CSV correspondiente.

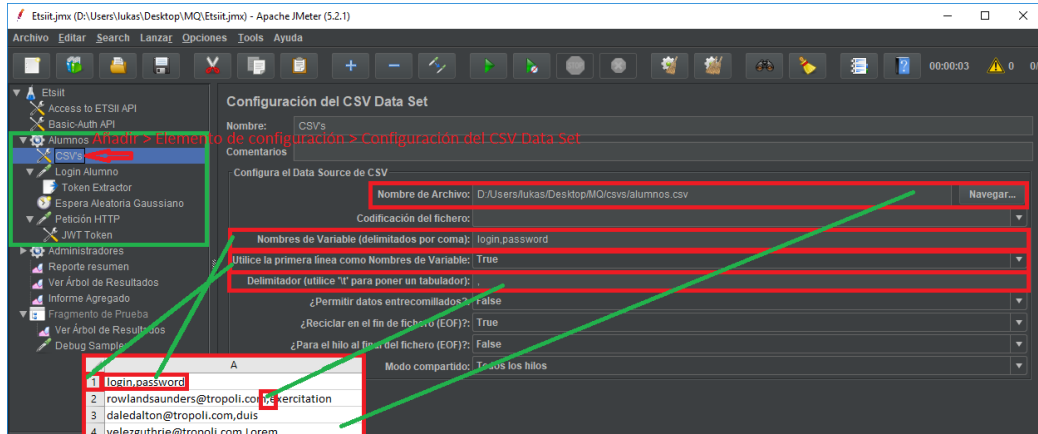


Figure 5: Quinto paso, asignar archivos CSV

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

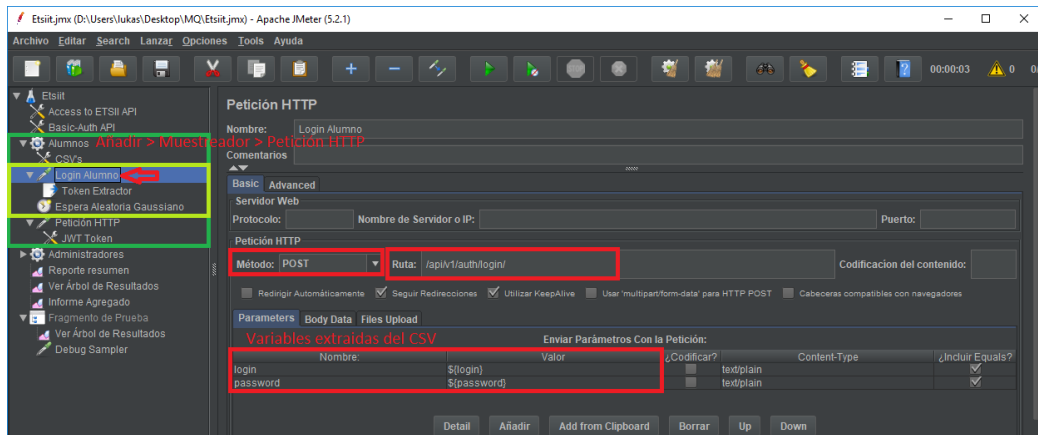


Figure 6: Sexto, Peticiones HTTP de cada muestra

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

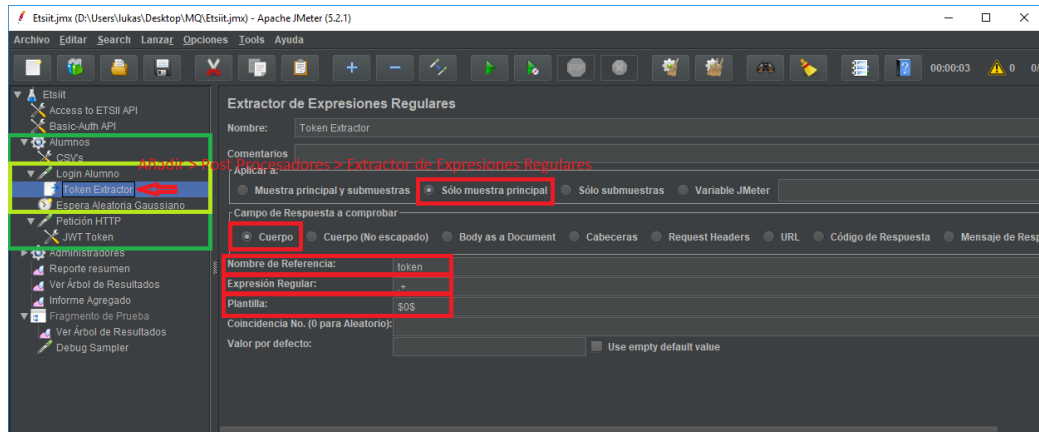


Figure 7: Séptimo, Extracción del token

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.

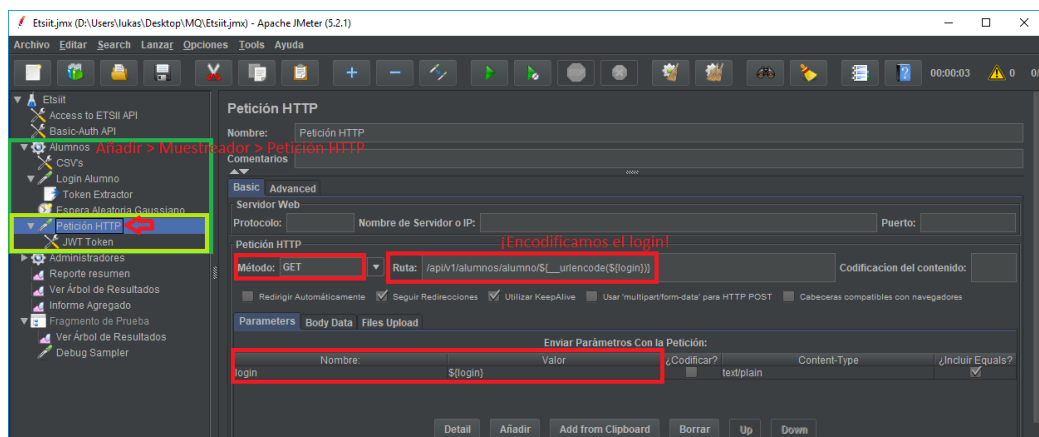
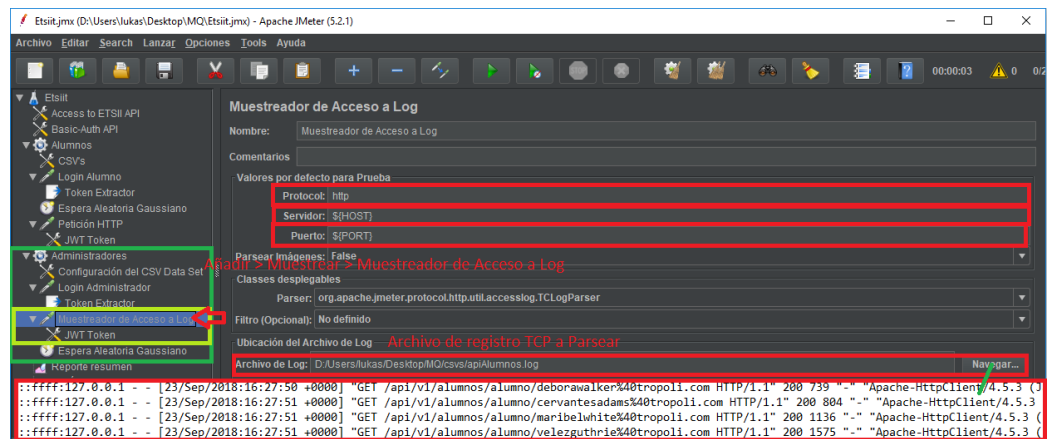


Figure 8: Octavo paso, petición del token JWT

El screenshot shows the Apache JMeter GUI. On the left, the 'Test Plan' tree is visible, with 'Petición HTTP' selected. The main panel displays the 'Gestor de Cabecera HTTP' configuration. The 'Nombre' field is set to 'JWT Token', and the 'Valor' field is set to 'Bearer \${token}'. A red box highlights the 'Nombre' and 'Valor' fields, and a blue arrow points to the '\${token}' variable. A red box also highlights the 'Comentarios' field with the text 'Creamos esquema de Autenticación Bearer'. A red box highlights the 'Comentarios' field with the text 'Creamos esquema de Autenticación Bearer'. A red box highlights the 'Comentarios' field with the text 'Creamos esquema de Autenticación Bearer'.

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



7



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

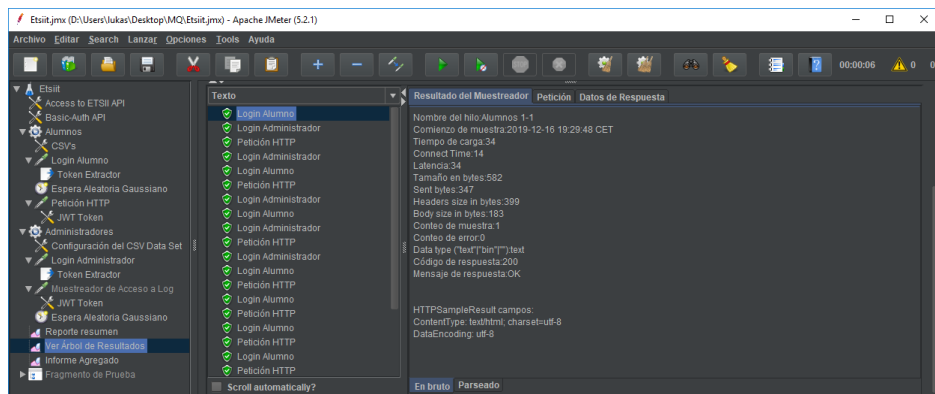


Figure 11: Árbol de peticiones

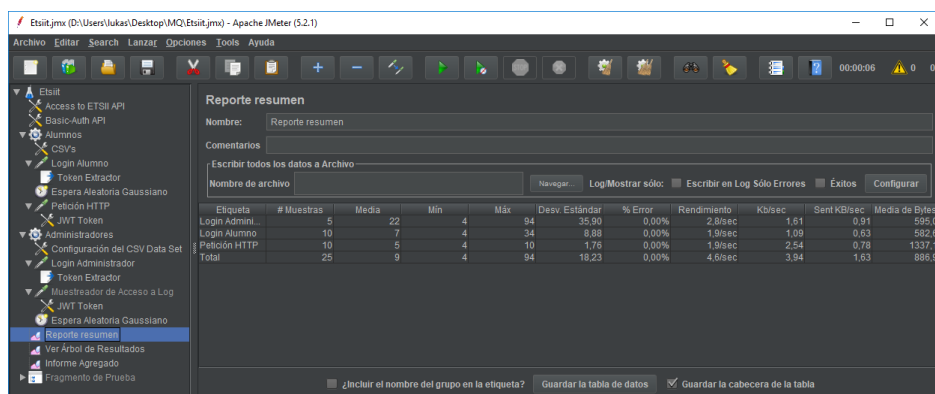


Figure 12: Resumen de las peticiones

## 2 Referencias Bibliográficas

### References

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