

# Práctica 4 - L2

## Parte Práctica:

### En AlmaLinux:

`cd opt/jmeter`, aquí es donde tenemos instalado jmeter.  
`/bin/jmeter.sh` con este archivo .sh podemos comprobar si nos funciona jmeter y lo hemos instalado correctamente.

### En PC host (Windows en mi caso):

En el navegador introducimos con el docker corriendo en Debian la ip del debian:3000.

Abrimos jmeter, guardamos el TestPlan-1, en Test Plan Add -> Threads -> Threads Group. Threads Group Add -> Config Element -> HTTP Authorization Manager. Dentro del Http Authorization Manager añadimos y ponemos de Base URL el `http://ipDebian:3000/`, de usuario y contraseña ponemos lo que pone en la página de ip Debian:3000

Threads Group -> Add -> Sampler -> HTTP Request. En HTTP Request, el protocolo es http, la ip es la de Debian, el puerto 3000. de HTTP Request cambiamos GET por POST y ponemos lo que viene en la página de ipDebian:3000 (`/api/v1/autenti...`)

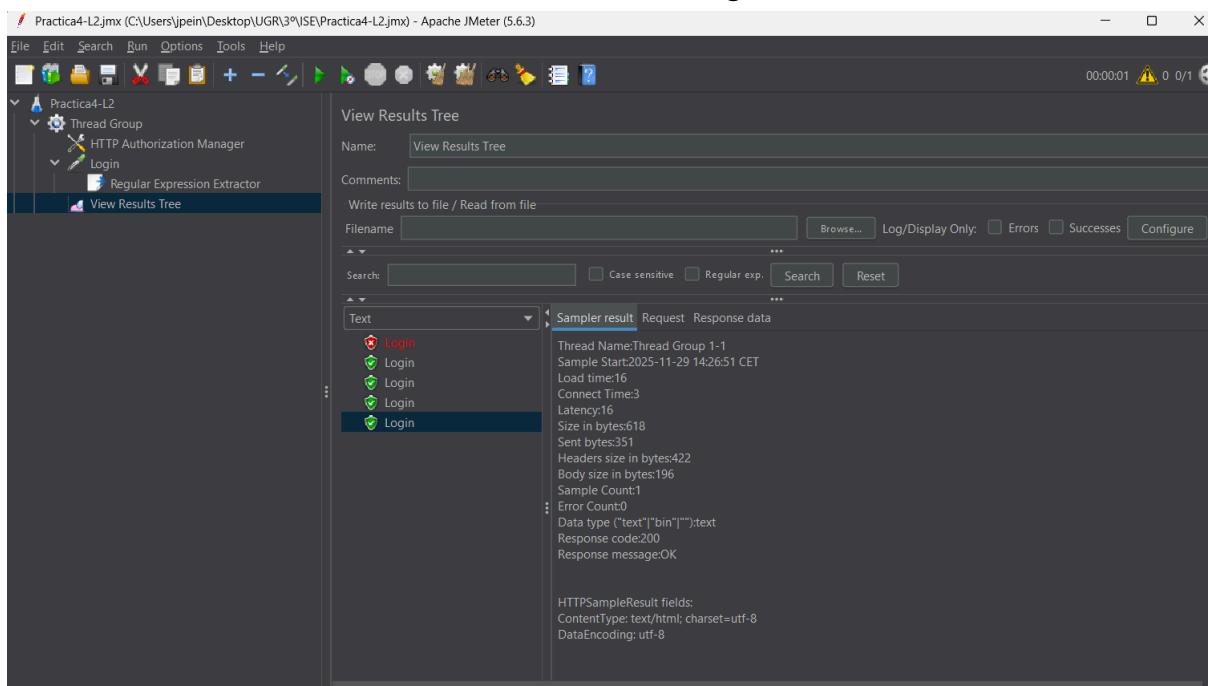
### En Debian:

Introducimos el comando `sudo docker exec -it ise-p4app-mongodb-1` dentro hacemos `show dbs`, `use etsii`, `show collections`, `db.usuarios.find()` `db.alumnos.find()`, con estos dos últimos comando veremos los datos de los usuarios y los alumnos.

## En PC host (Windows en mi caso):

En HTTP Request, el Name será **Login**, en Parameters ponemos de nombre **login** y de value el **correo electrónico** de un usuario que hayamos visto en **db.usuarios.find()** de Debian, añadimos otro parameter y le ponemos de name **password** y de value la **contraseña del usuario** que hemos elegido.

En Login -> Add -> Post Processor -> Regular Expression Extractor. En Regular Expression Extractor de name created variable ponemos **token** y de regular expresion **El token es:(.+)**, en template ponemos **\$1\$** y en match ponemos **1**. En Thread Group -> Add -> Listener-> View Result Tree. Si en View Result Tree sale Login en verde.



En View Result Tree vemos estadísticas en el Request podemos ver los detalles de la petición, y en Response data podemos ver el token que nos devolvieron.

En Login, en vez de poner en parameters lo que hemos puesto, podemos dentro de Body Data podemos pasarlo como un json, ejem:

```
{  
    "login" : "emaildelusuario"  
    "password" : "contraseña del usuario"  
}
```

además tendríamos que en Login->Add->Config Element->HTTP Header Manager.

HTTP Header Manager en Name Content-Type, y en value /applications/json.

Para la segunda parte que es la de consultar alumno, haremos Thread Group -> Add -> Http Request.

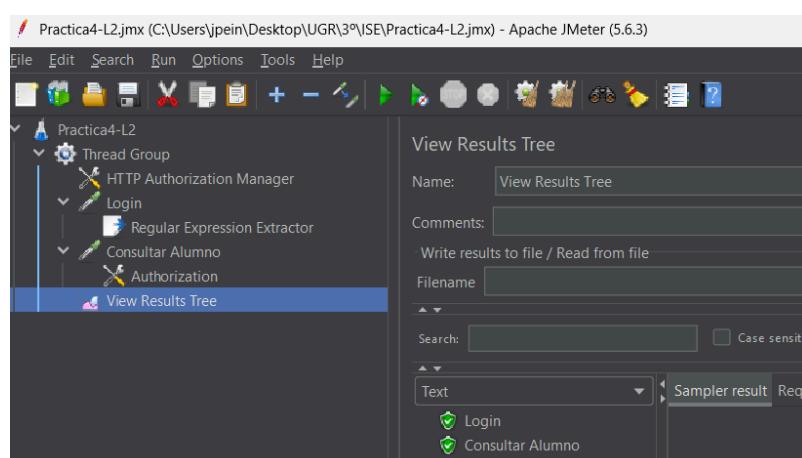
En HTTP Request, name: Consultar alumno, protocol http, IP: ip de Debian, puerto: 3000.

En parameters: name email, value el de un alumno, (otro parameter) name

En Consultar alumno -> Add-> Config Element-> HTTP Header Manager.

En HTTP Header Manager name Authorization, value Bearer \${token}

En Thread Group ->Add-> Result Tree. Y dentro de este vemos si están en verde, el Consultar alumno nunca estará verde(válido) si el login ha fallado.



File -> Save as: Guardamos la configuración del fichero con el nombre carga.

Ahora editamos y en Thread Group cambiamos Number of threads a 50, lo de en medio a 1 y loop count 100. Deshabilitamos Result tree para no consumir demasiados recursos.

Ahora tenemos que pasarle el archivo **carga** a Alma, con **scp -P 22022 carga.jmx jpeiper@192.168.56.110:/home/jpeiper**. Con esto tendríamos copiado el archivo en Alma.

## En AlmaLinux:

Comprobamos que está el archivo, **ls /home/jpeiper**.

**cp /home/jpeiper/carga.jmx /opt** para copiarlo en la carpeta donde tengo el jmeter.

**jmeter/bin/jmeter.sh -n** (sin entorno gráfico) **-t carga.jmx -l result.jtl**  
(para que nos guarde los resultados en este archivo)

## En Debian:

Con **docker stats** vemos el uso de cpu, etc de cada docker, con esto podemos ver mientras se ejecuta el jmeter que están trabajando.

## **En AlmaLinux:**

Lo importante es que los porcentaje de la derecha tras ejecutar sean 0% que significa 0% de errores.

## **En PC host (Windows en mi caso):**

**scp -P 22022 jpeiper@192.168.56.110:/opt/result.jtl .** para copiar desde alma el resultado en Windows en el archivo en el que estemos. con **File -> Open ->** Abrimos el archivo **result.jtl** y podemos ver los datos y estadísticas, la columna del error es de las más importantes.

## TAREA:

**Hay que comprobar que cuando haces el Login con un usuario si pides los datos de otro alumno distinto al que has hecho el login que Login se ponga en Verde, y datos del alumno en Rojo.**

The screenshot shows the JMeter interface with a 'View Results Tree' listener selected. The tree view displays two requests: 'Login' (green checkmark) and 'Consultar Alumno' (red shield icon). The 'Sampler result' tab is selected, showing detailed metrics for both requests. The 'Response data' tab is also visible, showing the response body for each request.

**Sampler result**

Thread Name: Thread Group 1-1  
Sample Start: 2025-11-30 12:51:21 CET  
Load time: 22  
Connect Time: 0  
Latency: 22  
Size in bytes: 522  
Sent bytes: 470  
Headers size in bytes: 428  
Body size in bytes: 94  
Sample Count: 1  
Error Count: 1  
Data type ("text"|"bin"|""): text  
**Response code: 403**  
Response message: Forbidden

**HTTPSampleResult fields:**  
Content-Type: text/html; charset=utf-8  
DataEncoding: utf-8

This screenshot shows the same JMeter setup as the first one, but the response body for the 'Consultar Alumno' request is different. It now contains the text: '<h1>Solo un administrador puede consultar los datos de otro alumno</h1><h2></h2><pre></pre>'.

**Sampler result**

Thread Name: Thread Group 1-1  
Sample Start: 2025-11-30 12:51:21 CET  
Load time: 22  
Connect Time: 0  
Latency: 22  
Size in bytes: 522  
Sent bytes: 470  
Headers size in bytes: 428  
Body size in bytes: 94  
Sample Count: 1  
Error Count: 1  
Data type ("text"|"bin"|""): text  
**Response code: 403**  
Response message: Forbidden

**HTTPSampleResult fields:**  
Content-Type: text/html; charset=utf-8  
DataEncoding: utf-8

Como podemos observar, nos dice que solo un administrador puede consultar los datos de otro alumno.

## Comprobar que al hacer Login con un administrador y pides los datos de otro alumno te salen ambas en verde.

The screenshot shows the JMeter interface with a Thread Group containing an HTTP Authorization Manager and a Login sampler. The Login sampler is selected. The 'Basic' tab of the configuration panel is shown, with the 'Protocol [http]: http' and 'Server Name or IP: 192.168.56.105' fields filled. The 'HTTP Request' section shows a POST method and a path of '/api/v1/authentication/login'. Under the 'Parameters' tab, there are two entries: 'login' with value 'stuartkirby@etsii.ugr.es' and 'password' with value 'tempor'. The 'View Results Tree' tab is also visible on the right.

The screenshot shows the JMeter interface with a Thread Group containing a Login sampler and a Consultar Alumno sampler. The Consultar Alumno sampler is selected. The 'Basic' tab of the configuration panel is shown, with the 'Protocol [http]: http' and 'Server Name or IP: 192.168.56.105' fields filled. The 'HTTP Request' section shows a POST method and a path of '/api/v1/alumnos/alum/'. Under the 'Parameters' tab, there is one entry: 'email' with value 'meredithwhitaker@tropoli.com'. The 'View Results Tree' tab is also visible on the right.

The screenshot shows the 'View Results Tree' panel. It lists two samplers: 'Login' and 'Consultar Alumno', both marked with green checkmarks. The 'Text' tab is selected. The 'Sampler result' pane displays the results for the 'Login' sampler, including the thread name, sample start time, load time, connect time, latency, size in bytes, sent bytes, headers size, body size, sample count, error count, data type, response code, and response message. The 'Request' and 'Response data' tabs are also visible.

Como podemos observar, con un administrador podemos acceder a la información de otro alumno.

## Ver con qué número en Number Threads, Ramp-up y Loop Count, da de porcentaje de error en result.jtl >0%.

The screenshot shows the Apache JMeter 5.6.3 interface. In the top window, a 'Thread Group' is selected. The configuration is as follows:

- Name: Thread Group
- Comments:
- Action to be taken after a Sampler error:
  - Continue (radio button selected)
  - Start Next Thread Loop
  - Stop Thread
  - Stop Test
  - Stop Test Now
- Thread Properties:
  - Number of Threads (users): 10000
  - Ramp-up period (seconds): 1
  - Loop Count:
    - Infinite (checkbox checked)
    - 1 (checkbox unselected)
  - Duration (seconds):
- Advanced options:
  - Same user on each iteration (checkbox checked)
  - Delay Thread creation until needed (checkbox unselected)
  - Specify Thread lifetime (checkbox unselected)

Below this window is a terminal window showing the command and its output:

```
[jpeiper@myjmeterserver jmeter]$ jmeter/bin/jmeter.sh -n -t Practica4-L2-Intentar-Romper.jmx -l result.jtl
WARN StatusConsoleListener The use of package scanning to locate plugins is deprecated and will be removed in a future release
WARN StatusConsoleListener The use of package scanning to locate plugins is deprecated and will be removed in a future release
WARN StatusConsoleListener The use of package scanning to locate plugins is deprecated and will be removed in a future release
WARN StatusConsoleListener The use of package scanning to locate plugins is deprecated and will be removed in a future release
Creating summariser <summary>
Created the tree successfully using Practica4-L2-Intentar-Romper.jmx
Starting standalone test @ 2025 Dec 1 08:52:21 CET (1764575541033)
Waiting for possible Shutdown/StopTestNow/HeapDump/ThreadDump message on port 4445
summary +  922 in 00:00:09 =  106.1/s Avg:  785 Min:   14 Max:  4115 Err:      0 (0.00%) Active: 17 Started: 478 Finished: 461
summary +  5255 in 00:00:30 = 172.7/s Avg:  1967 Min:     6 Max:  8381 Err:      3 (0.06%) Active: 588 Started: 3576 Finished: 2988
summary =  6177 in 00:00:39 = 157.8/s Avg:  1791 Min:     6 Max:  8381 Err:      3 (0.05%)
```

Como podemos observar, con 10000 Number Threads, 1 de Ramp-up y 1 de Loop Count, hemos conseguido que tenga un error > 0%.