

JMeter ~ José Santos Salvador

JMeter nos permite hacer test complejos, se define como una aplicación diseñada para cargar test funcional de comportamiento y de rendimiento. Permite crear concurrencia real debido a la posibilidad de usar varias hebras dentro de la misma CPU, así como distribuir la creación de carga de la máquina.

Nota: añadir que este segundo ejercicio lo realicé en las mismas máquinas virtuales pero el HOST era distinto (esta vez era con mi portátil). Al igual que en otros ejercicios, hago ssh desde el host a las MV para poder copiar y pegar comandos.

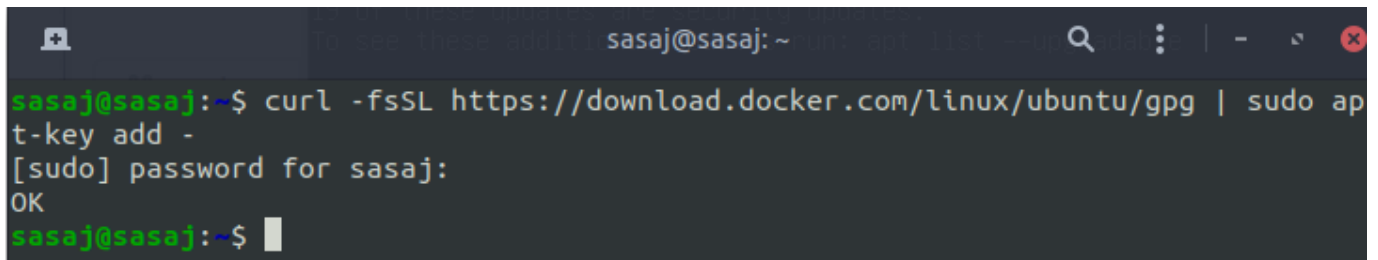
Lo que vamos a realizar es un test básico para una web usando Jmeter, para hacer un test sobre la aplicación que ejecuta sobre dos contenedores (usamos docker-compose en la máquina virtual para esto)

Instalación docker Ubuntu-server

El primer paso para este ejercicio, es descargar docker en la MV.

Añadimos llave GPG para validar el repositorio

```
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -
```

A terminal window with a dark background. The prompt is 'sasaj@sasaj:~'. The command 'curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -' is entered. The output shows '[sudo] password for sasaj:' followed by 'OK'. The prompt returns to 'sasaj@sasaj:~\$'.

```
sasaj@sasaj:~$ curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -  
[sudo] password for sasaj:  
OK  
sasaj@sasaj:~$
```

Añadimos repositorio

```
sudo add-apt-repository  
"deb [arch=amd64] https://download.docker.com/linux/ubuntu  
$(lsb_release -cs) stable"
```

```
sasaj@sasaj:~$ sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu
> $(lsb_release -cs) stable"
Hit:1 http://es.archive.ubuntu.com/ubuntu focal InRelease
Get:2 http://es.archive.ubuntu.com/ubuntu focal-updates InRelease [114 kB]
Get:3 http://es.archive.ubuntu.com/ubuntu focal-backports InRelease [101 kB]
Get:4 http://es.archive.ubuntu.com/ubuntu focal-security InRelease [109 kB]
Get:5 http://repo.zabbix.com/zabbix/5.0/ubuntu focal InRelease [7123 B]
Get:6 https://download.docker.com/linux/ubuntu focal InRelease [36.2 kB]
Get:7 http://es.archive.ubuntu.com/ubuntu focal-updates/main amd64 Packages [698
kB]
Get:8 http://es.archive.ubuntu.com/ubuntu focal-updates/main Translation-en [175
kB]
Get:9 http://es.archive.ubuntu.com/ubuntu focal-updates/main amd64 c-n-f Metadat
a [11.3 kB]
Get:10 http://es.archive.ubuntu.com/ubuntu focal-updates/restricted amd64 Packag
```

Actualizamos lista de repositorios

```
sudo apt update
```

```
sasaj@sasaj:~$ sudo apt update
Hit:1 http://es.archive.ubuntu.com/ubuntu focal InRelease
Hit:2 https://download.docker.com/linux/ubuntu focal InRelease
Hit:3 http://es.archive.ubuntu.com/ubuntu focal-updates InRelease
Hit:4 http://repo.zabbix.com/zabbix/5.0/ubuntu focal 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
120 packages can be upgraded. Run 'apt list --upgradable' to see them.
sasaj@sasaj:~$
```

Buscamos el repositorio de docker y lo instalamos

```
apt search docker-ce
sudo apt install docker-ce
```

```
sasaj@sasaj:~$ apt search docker-ce
Sorting... Done
Full Text Search... Done
docker-ce/focal 5:20.10.0~3-0~ubuntu-focal amd64
  Docker: the open-source application container engine

docker-ce-cli/focal 5:20.10.0~3-0~ubuntu-focal amd64
  Docker CLI: the open-source application container engine

docker-ce-rootless-extras/focal 5:20.10.0~3-0~ubuntu-focal amd64
  Rootless support for Docker.

sasaj@sasaj:~$ sudo apt install docker-ce
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages were automatically installed and are no longer required:
  linux-headers-5.4.0-48 linux-headers-5.4.0-48-generic
  linux-image-5.4.0-48-generic linux-modules-5.4.0-48-generic
  linux-modules-extra-5.4.0-48-generic
```

Añadimos el usuario al grupo docker

```
sudo usermod -aG docker sasaj
```

Reiniciamos el bash y ejecutamos ahora sin sudo

```
docker info
```

```
sasaj@sasaj:~$ docker info
Client:
 Context:    default
 Debug Mode: false
 Plugins:
  app: Docker App (Docker Inc., v0.9.1-beta3)
  buildx: Build with BuildKit (Docker Inc., v0.4.2-docker)

Server:
 Containers: 0
  Running: 0
  Paused: 0
  Stopped: 0
 Images: 0
 Server Version: 20.10.0
 Storage Driver: overlay2
  Backing Filesystem: extfs
  Supports d_type: true
  Native Overlay Diff: true
 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: runc io.containerd.runc.v2 io.containerd.runtime.v1.linux
 Default Runtime: runc
 Init Binary: docker-init
 containerd version: 269548fa27e0089a8b8278fc4fc781d7f65a939b
 runc version: ff819c7e9184c13b7c2607fe6c30ae19403a7aff
 init version: de40ad0
 Security Options:
  apparmor
  seccomp
   Profile: default
 Kernel Version: 5.4.0-53-generic
 Operating System: Ubuntu 20.04.1 LTS
 OSType: linux
 Architecture: x86_64
 CPUs: 2
 Total Memory: 1.941GiB
 Name: sasaj
 ID: WUGB:52YQ:F2CB:KMUU:GYDU:P53J:YFJT:BISY:PT2E:Z742:VJ7A:ZNWQ
 Docker Root Dir: /var/lib/docker
 Debug Mode: false
 Registry: https://index.docker.io/v1/
 Labels:
 Experimental: false
 Insecure Registries:
  127.0.0.0/8
 Live Restore Enabled: false

WARNING: No swap limit support
WARNING: No blkio weight support
```

Y ahora probamos el contenedor tipico de hello-world y al no estar en el sistema, se descarga

```
docker run hello-world
```

```
sasaj@sasaj:~$ docker run hello-world
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
0e03bdcc26d7: Pull complete
Digest: sha256:1a523af650137b8accdaed439c17d684df61ee4d74feac151b5b337bd29e7eec
Status: Downloaded newer image for hello-world:latest

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/

sasaj@sasaj:~$
```

Instalacion docker-compose Ubuntu-server

Para organizar y orquestar los contenedores de este ejercicio (en este caso uno para la BD y otro para la aplicación en sí) usamos docker-compose, también es muy usado Kubernetes

Instalamos docker-compose

```
apt install docker-compose
```

```
sasaj@sasaj:~$ sudo apt install docker-compose
[sudo] password for sasaj:
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages were automatically installed and are no longer required:
  linux-headers-5.4.0-48 linux-headers-5.4.0-48-generic linux-image-5.4.0-48-generic
  linux-modules-5.4.0-48-generic linux-modules-extra-5.4.0-48-generic
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
  cgroudfs-mount python3-cached-property python3-docker python3-dockerpty python3-docopt
  python3-texttable python3-websocket
Recommended packages:
  docker.io
The following NEW packages will be installed:
  cgroudfs-mount docker-compose python3-cached-property python3-docker python3-dockerpty
  python3-texttable python3-websocket
0 upgraded, 8 newly installed, 0 to remove and 120 not upgraded.
2 not fully installed or removed.
Need to get 268 kB of archives.
After this operation, 1649 kB 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 cgroudfs-mount all 1.4 [63
kB]
Get:2 http://es.archive.ubuntu.com/ubuntu focal/universe amd64 python3-cached-property al
kB]
Get:3 http://es.archive.ubuntu.com/ubuntu focal/universe amd64 python3-websocket all 0.53
3 kB]
Get:4 http://es.archive.ubuntu.com/ubuntu focal/universe amd64 python3-docker all 4.1.0-1
Get:5 http://es.archive.ubuntu.com/ubuntu focal/universe amd64 python3-dockerpty all 0.4.
Get:6 http://es.archive.ubuntu.com/ubuntu focal/universe amd64 python3-docopt all 0.6.2-2
kB]
Get:7 http://es.archive.ubuntu.com/ubuntu focal/universe amd64 python3-texttable all 1.6.
Get:8 http://es.archive.ubuntu.com/ubuntu focal/universe amd64 docker-compose all 1.25.0-
```

Ahora probamos a lanzar docker-compose y su version con

```
docker-compose
docker-compose --version
```

docker-compose --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

Commands:

build	Build or rebuild services
bundle	Generate a Docker bundle from the Compose file
config	Validate and view the Compose file
create	Create services
down	Stop and remove containers, networks, images, and volumes
events	Receive real time events from containers
exec	Execute a command in a running container
help	Get help on a command
images	List images
kill	Kill containers
logs	View output from containers
pause	Pause services
port	Print the public port for a port binding
ps	List containers
pull	Pull service images
push	Push service images
restart	Restart services
rm	Remove stopped containers
run	Run a one-off command
scale	Set number of containers for a service
start	Start services
stop	Stop services
top	Display the running processes
unpause	Unpause services
up	Create and start containers
version	Show the Docker-Compose version information

```
sasa]@sasa]~$ docker-compose --version
docker-compose version 1.25.0, build unknown
sasa]@sasa]~$
```

Me da un error en initramfs por falta de espacio y siguiendo la referencia⁴, borro kernels antiguos


```
sasaj@sasaj:~$ sudo apt --purge autoremove
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages will be REMOVED:
  linux-headers-5.4.0-48* linux-headers-5.4.0-48-generic* linux-image-5.4.0-48-generic*
  linux-modules-5.4.0-48-generic* linux-modules-extra-5.4.0-48-generic*
0 upgraded, 0 newly installed, 5 to remove and 120 not upgraded.
2 not fully installed or removed.
After this operation, 360 MB disk space will be freed.
Do you want to continue? [Y/n] Y
(Reading database ... 170773 files and directories currently installed.)
Removing linux-headers-5.4.0-48-generic (5.4.0-48.52) ...
□
```

Instalacion Jmeter Ubuntu (host)

Siguiendo referencia [3](#)

Comprobé que tenía la versión correcta de java y javac

```
java -version
javac -version
```

```
juse@juse-OMEN-by-HP-Laptop:~$ java -version
java version "1.8.0_221"
Java(TM) SE Runtime Environment (build 1.8.0_221-b11)
Java HotSpot(TM) Server VM (build 25.221-b11, mixed mode)
juse@juse-OMEN-by-HP-Laptop:~$ javac -version
javac 1.8.0_221
juse@juse-OMEN-by-HP-Laptop:~$ █
```

Nos bajamos jmeter con wget (usado ya en otros ejercicios) con la versión que queramos

```
wget http://www.gtlib.gatech.edu/pub/apache/jmeter/binaries/
<Apache_JMETER_VERSION>
```

En mi caso he usado la 5.2 (por no usar la más nueva por si tenía errores) [1](#)

```
wget http://www.gtlib.gatech.edu/pub/apache/jmeter/binaries/apache-
jmeter-5.2.tgz
```



```
juse@juse-OMEN-by-HP-Laptop:~$ wget http://www.gtlib.gatech.edu/pub/apache/jmeter/binaries/apache-jmeter-5.2.tgz
--2020-12-14 20:50:39-- http://www.gtlib.gatech.edu/pub/apache/jmeter/binaries/apache-jmeter-5.2.tgz
Resolviendo www.gtlib.gatech.edu (www.gtlib.gatech.edu)... 128.61.111.9, 128.61.111.11, 128.61.111.10, ..
Conectando con www.gtlib.gatech.edu (www.gtlib.gatech.edu)[128.61.111.9]:80... conectado.
Petición HTTP enviada, esperando respuesta... 200 OK
Longitud: 62106843 (59M) [application/x-gzip]
Guardando como: "apache-jmeter-5.2.tgz"

apache-jmeter-5.2.tgz      100%[=====>] 59,23M  10,5MB/s   en 6,1s

2020-12-14 20:50:46 (9,71 MB/s) - "apache-jmeter-5.2.tgz" guardado [62106843/62106843]
```

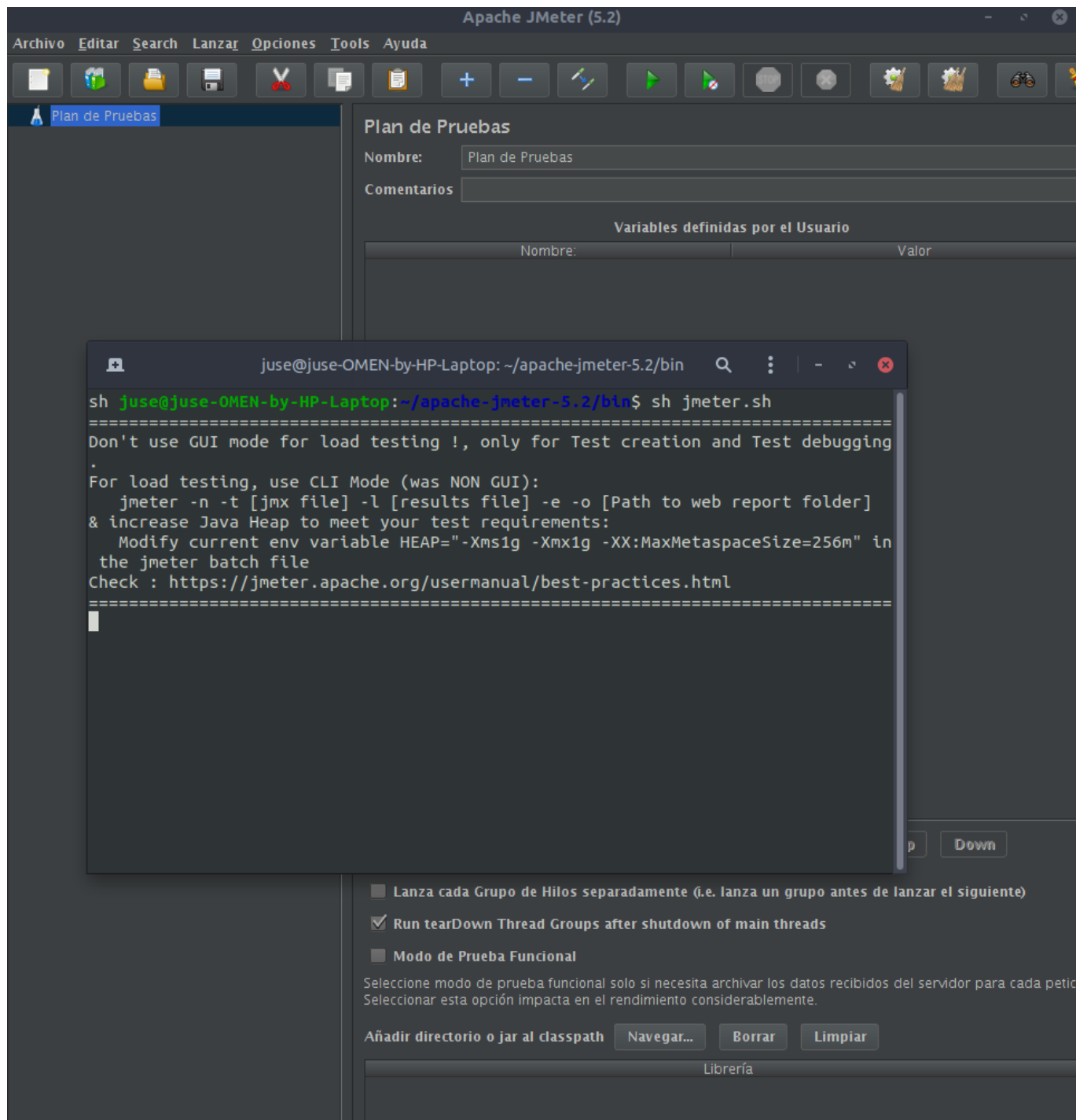
Extraemo el archivo con

```
tar xf apache-jmeter-5.2.tgz
```

```
juse@juse-OMEN-by-HP-Laptop:~$ tar xf apache-jmeter-5.2.tgz
juse@juse-OMEN-by-HP-Laptop:~$ ls
Android                               commaCT-2020.01.0    mozilla.pdf          Sync
7-linux                             Descargas           Música               thrift-0.13.0
Desktop                             Documentos         netbeans-8.2        vcpkg-master
dotfiles                           Escritorio          NetBeansProjects    Videos
examples.desktop                   glassfish-4.1.1     node_modules        'VirtualBox VMs'
Imágenes                           lampstack-7.3.12-0  Plantillas          Visual_Paradigm_16.0
snap                                public              raku                 zef
cd apache-jmeter-5.2/
apache-jmeter-5.2$ ls
LICENSE  licenses  NOTICE  printable_docs  README.md
juse@juse-OMEN-by-HP-Laptop:~/apache-jmeter-5.2$ cd bin/
juse@juse-OMEN-by-HP-Laptop:~/apache-jmeter-5.2/bin$ ls
ApacheJMeter.jar      jaas.conf          krb5.conf           stoptest.cmd
BeanShellAssertion.bshrc  jmeter            log4j2.xml          stoptest.sh
BeanShellFunction.bshrc  jmeter.bat        mirror-server       system.properties
BeanShellListeners.bshrc  jmeter-n.cmd      mirror-server.cmd   templates
BeanShellSampler.bshrc   jmeter-n-r.cmd    mirror-server.sh    threaddump.cmd
create-rmi-keystore.bat  jmeter.properties reportgenerator.properties  threaddump.sh
examples               jmeter-server     report-template     upgrade.properties
hc.parameters           jmeter-server.bat rmi_keystore.jks    user.properties
heapdump.cmd            jmeter.sh         saveservice.properties  utility.groovy
heapdump.sh             jmeter-t.cmd      shutdown.cmd
juse@juse-OMEN-by-HP-Laptop:~/apache-jmeter-5.2/bin$ sh jmeter.sh
=====
Don't use GUI mode for load testing !, only for Test creation and Test debugging.
For load testing, use CLI Mode (was NON GUI):
    jmeter -n -t [jmx file] -l [results file] -e -o [Path to web report folder]
& increase Java Heap to meet your test requirements:
    Modify current env variable HEAP="-Xms1g -Xmx1g -XX:MaxMetaspaceSize=256m" in the jmeter batch file
Check : https://jmeter.apache.org/usermanual/best-practices.html
=====
```

Lanzamos jmeter con

```
sh jmeter.sh
```



Instalación iseP4JMeter en Ubuntu-Server

Durante la realización de esta parte, tenía el docker-compose up lanzado en la MV para poder hacer así las peticiones de forma correcta.

Siguiendo la referencia [2](#) en el github de ISEJmeter, procedemos a realizar la instalación del servicio en la MV

Clonamos el repo

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

Cambiamos a ese repositorio

```
cd iseP4JMeter
```

```
sasaj@sasaj:~$ git clone https://github.com/davidPalomar-ugr/iseP4JMeter.git
Cloning into 'iseP4JMeter'...
remote: Enumerating objects: 4, done.
remote: Counting objects: 100% (4/4), done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 3778 (delta 0), reused 0 (delta 0), pack-reused 3774
Receiving objects: 100% (3778/3778), 7.78 MiB | 2.18 MiB/s, done.
Resolving deltas: 100% (706/706), done.
sasaj@sasaj:~$ cd iseP4JMeter
sasaj@sasaj:~/iseP4JMeter$ docker-compose up
```

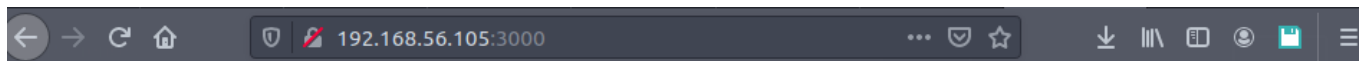
Y lanzamos el servicio con el orquestador docker-compose up

```
sasaj@sasaj:~$ ls
iseP4JMeter  mon RAID.py  phoronix-test-suite_10.0.1_all.deb  zabbix-release_5.0-1+focal_all.deb
sasaj@sasaj:~$ cd iseP4JMeter/
sasaj@sasaj:~/iseP4JMeter$ sudo [ 103.115130] cloud-init[1348]: Cloud-init v. 20.2-45-g5f7825e2-0ubuntu1~20.04.1 running 'modules:final' at Tue, 15 Dec 2020 07:41:25 +0000. Up 102.51 seconds.
[ 103.115695] cloud-init[1348]: Cloud-init v. 20.2-45-g5f7825e2-0ubuntu1~20.04.1 finished at Tue, 15 Dec 2020 07:41:26 +0000. Datasource DataSourceNone. Up 103.10 seconds
[ 103.116035] cloud-init[1348]: 2020-12-15 07:41:26,593 - cc_final_message.py[WARNING]: Used fallback datasource
[sudo] password for sasaj:
Starting isep4jmeter_mongodb_1 ... done
Starting isep4jmeter_nodejs_1 ... done
Starting isep4jmeter_mongoddbinit_1 ... done
Attaching to isep4jmeter_mongodb_1, isep4jmeter_mongoddbinit_1, isep4jmeter_nodejs_1
```

Siguiendo el README del repo, lanzamos la petición get

```
http://192.168.56.105:3000/
```

y obtenemos de la API



ETSII Alumnos API

Descripción de la API Restful:

POST /api/v1/auth/login

Parametros:

login:<emailUsuario>

password:<secreto>

Seguridad:

Acceso protegido con BasicAuth (etsiiApi:laApiDeLaETSII DaLache)

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 vál:

Retorna:

Objeto Json con perfil de alumno

En la MV se obtiene esa petición GET y se le da respuesta como si ha visto en la captura de arriba

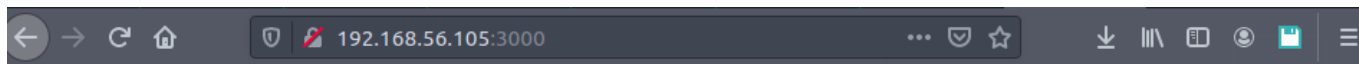
```
ount":0}}
ise4jmeter_mongodbinit_1 exited with code 0
nodejs_1 | GET / 200 10.334 ms - 843
nodejs_1 | GET /stylesheets/style.css 200 4.255 ms - 111
nodejs_1 | GET /favicon.ico 404 1.766 ms - 41
```

Configuración Alumnos

Ahora dentro de JMeter en Plan de Pruebas, en la pestaña de Variables definidas por el usuario, añadimos la IP que vamos a analizar que es de Ubuntu server (192.168.56.105) y el puerto (3000) según el github de ISEJmeter².

Para no perder la configuración cada vez que apagamos el pc, lo guardamos en un archivo .jmx en el icono guardar

Siguiendo la imagen de github ² me dispuse a cambiar el nombre del archivo para tenerlo igual. Le puse ETSIIT porque consideré que era una errata.



ETSII Alumnos API

Descripción de la API Restful:

POST /api/v1/auth/login

Parametros:

login:<emailUsuario>

password:<secreto>

Seguridad:

Acceso protegido con BasicAuth (etsiiApi:laApiDeLaETSII DaLache)

Retorna:

JWT Token

GET /api/v1/alumnos/alumno/<email>

Seguridad:

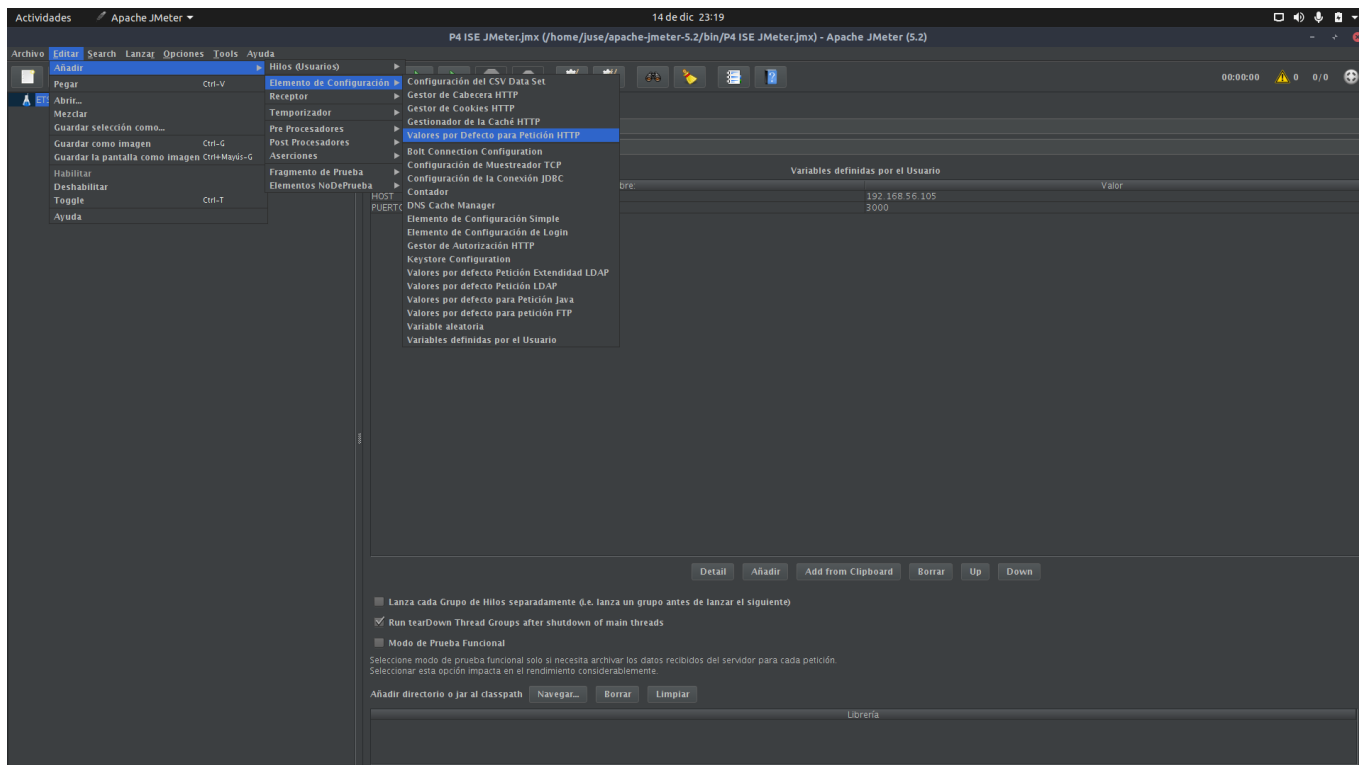
Token JWT valido en cabecera estandar authorization: Bearer <token>

Alumnos solo pueden solicitar sus datos. Administradores pueden solicitar cualquier alumno vál:

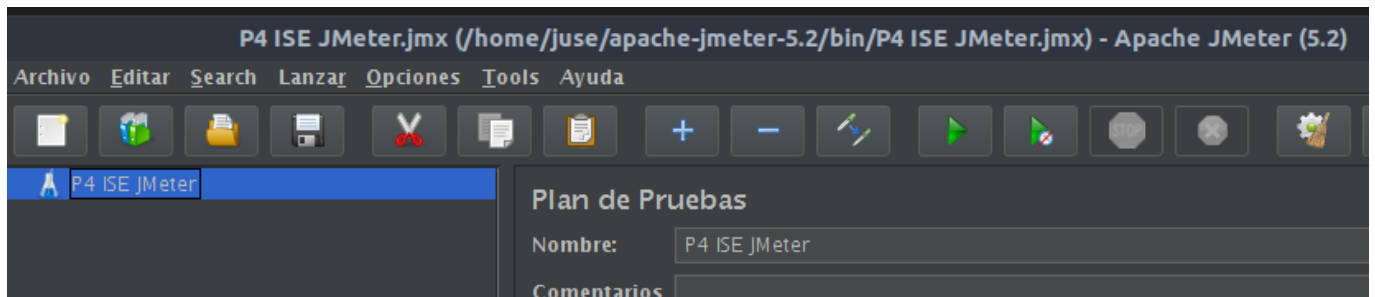
Retorna:

Objeto Json con perfil de alumno

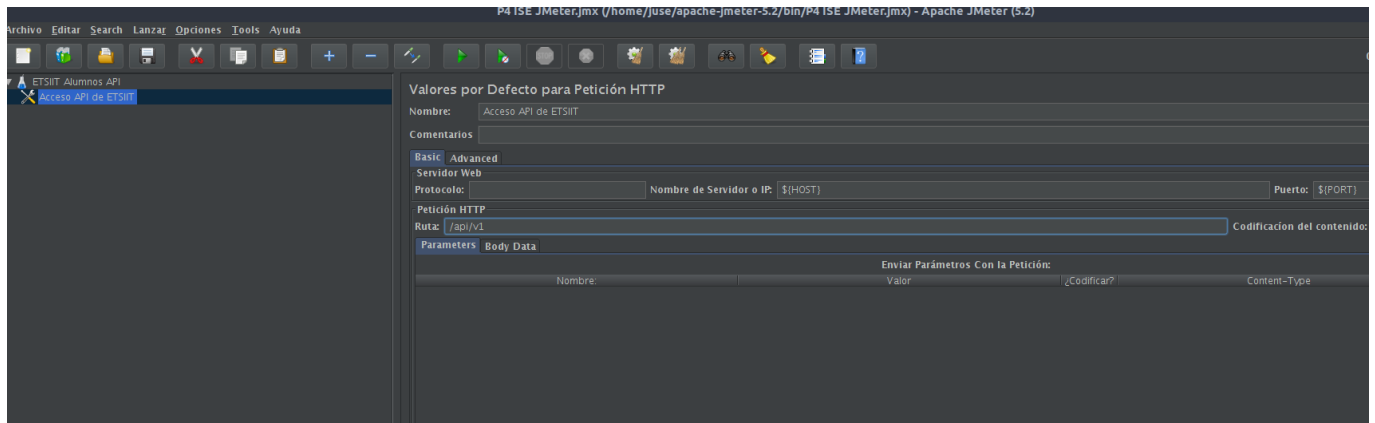
Ahora añadimos los valores por defecto para las peticiones HTTP. Editar > Añadir > Elemento de Configuración



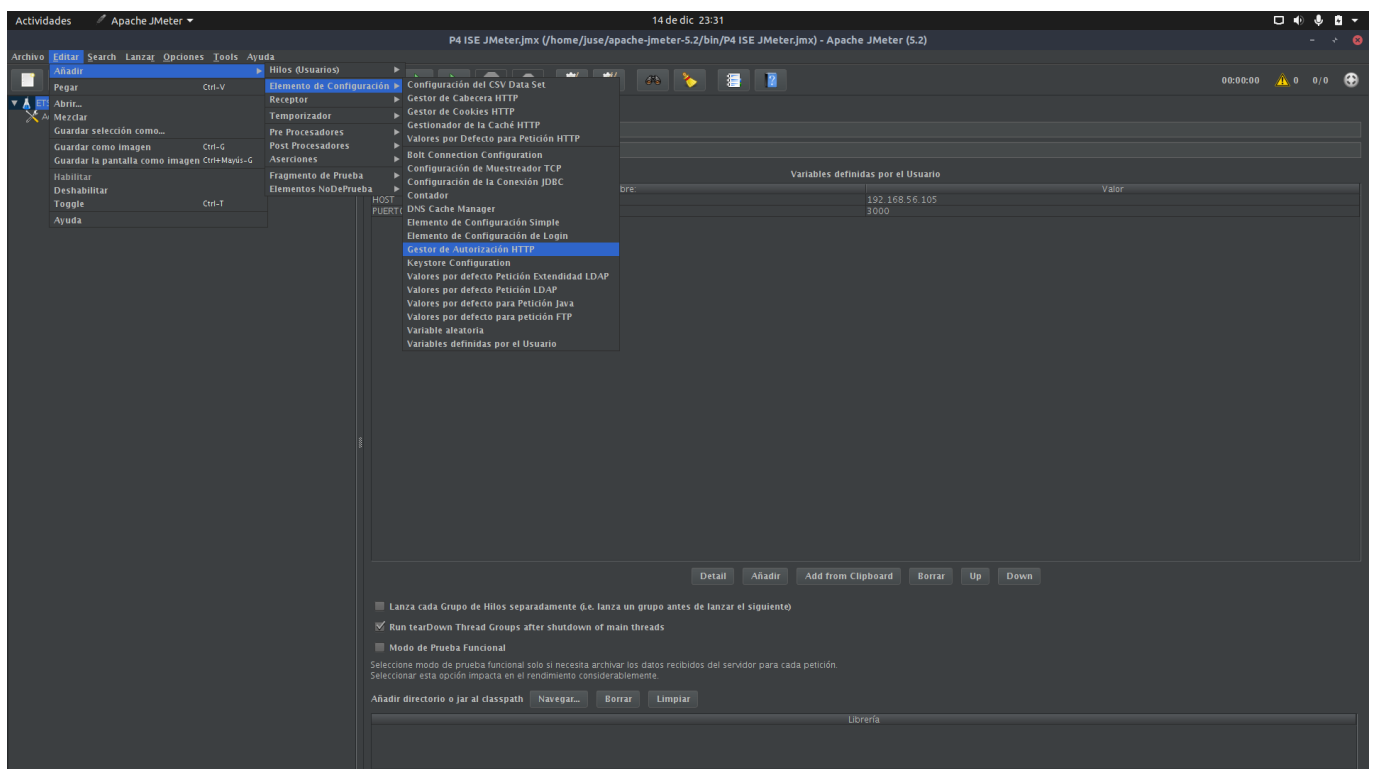
La ruta la sabemos porque al hacer <http://192.168.56.105:3000/> de la API de ETSIIIT pues vemos la ruta `api/v1/` como se puede ver en la foto



Con \$HOST variable HOST que hemos configurado al inicio y con \$PUERTO igual.



Ahora añadimos las credenciales. Editar > Añadir > Elemento de configuracion > Gestor de Autorización HTTP




tal y como sale al meterte desde el navegador en la API de ISE (mirar captura de arriba), la contraseña es: laApiDeLaETSIIaLache, te dice que protegica con BasicAuth (Base64)⁵ y utilizando la página <https://www.base64decode.org/> nos la convierte

Encode to Base64 format

Simply enter your data then push the encode button.

laApiDeLaETSIIaLache

 To encode binaries (like images, documents, etc.) use the file upload form a


UTF-8

Destination character set.

LF (Unix)

Destination newline separator.

- ☐ Encode each line separately (useful for multiple entries).
- ☐ Split lines into 76 character wide chunks (useful for MIME).
- ☐ Perform URL safe encoding (uses Base64URL format).

 Live mode OFF

Encodes in real-time when you type or paste (suppor

> ENCODE <

Encodes your data into the textarea below.

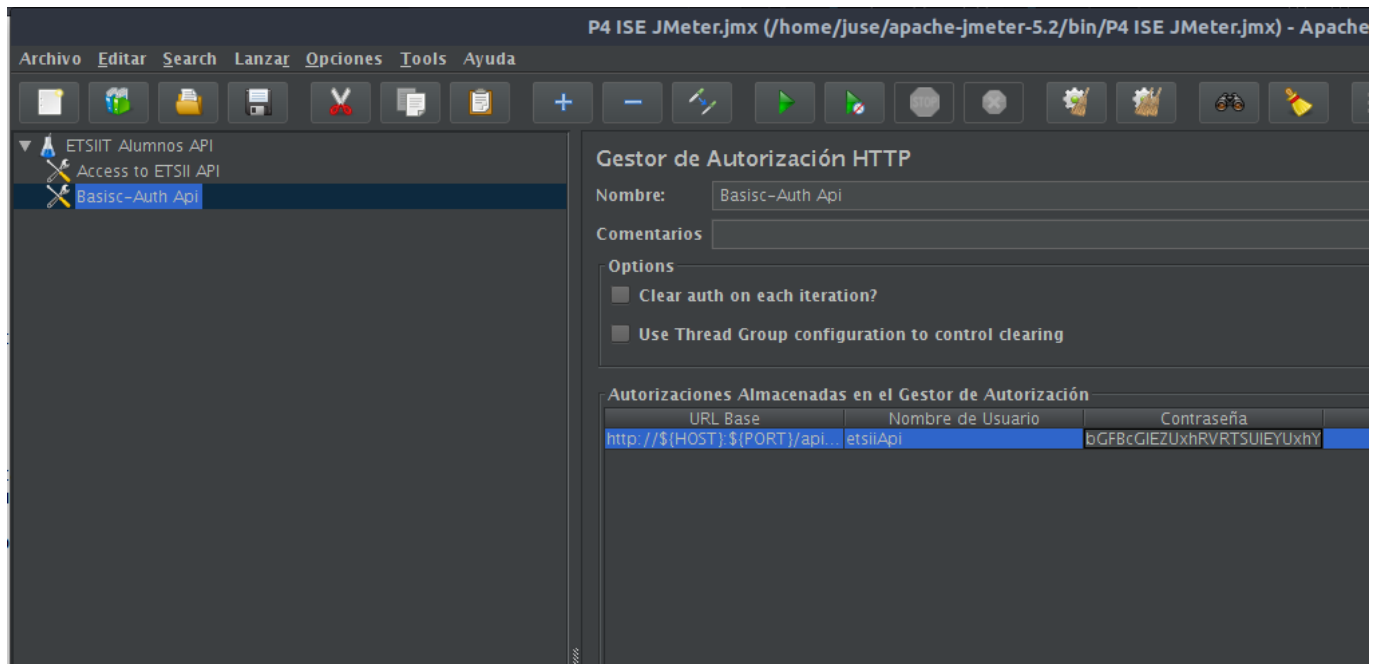
bGFBcGIEZUxhRVRTSUIEYUxhY2hl

nos devuelve bGFBcGIEZUxhRVRTSUIEYUxhY2hl

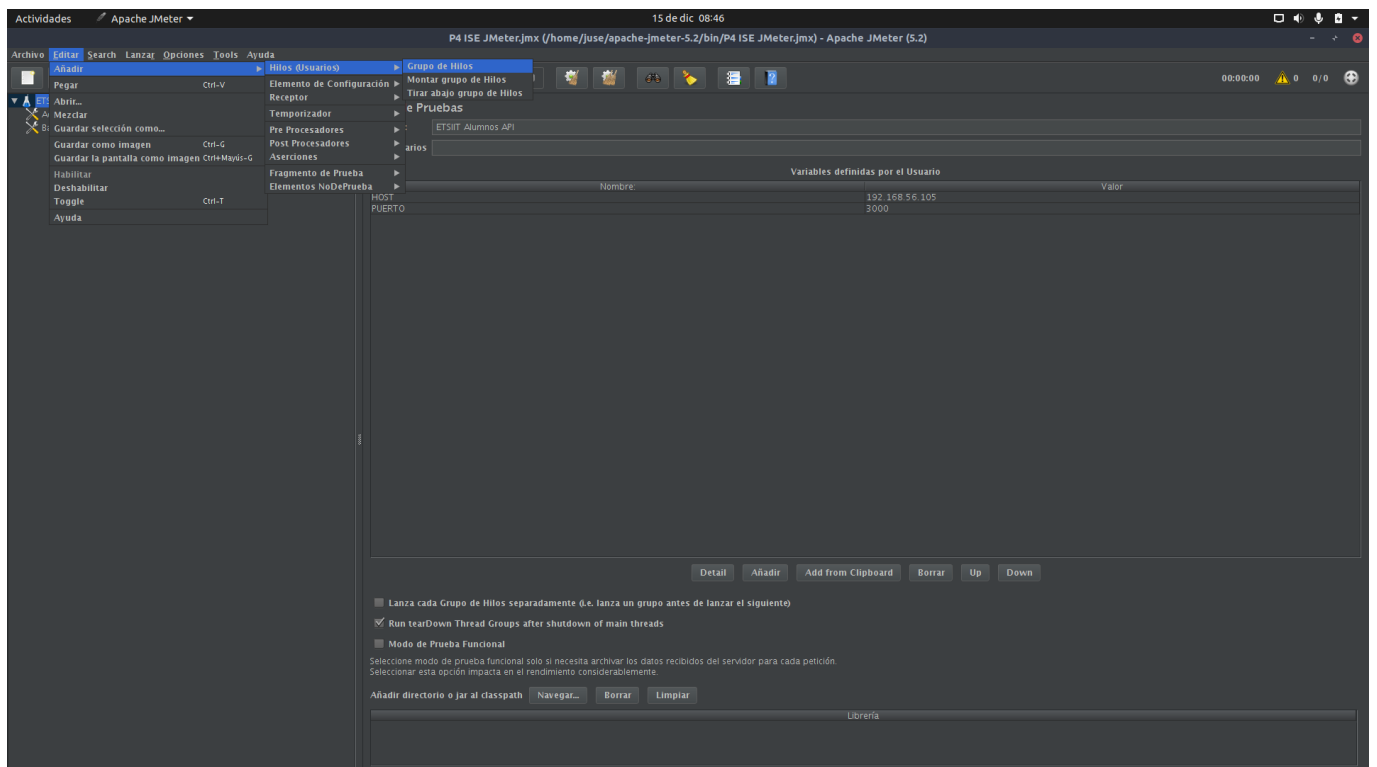
De esto no terminaba de estar seguro y decidí dejar la contraseña tal cual de laApiDeLaETSIIaLache suponiendo que BasicAuth protege de forma interna, ya que es un metodo HTTP user agent.

Ahora añadimos a esa autorizacion el usuario y contraseña que nos indica en la API para acceder a ello, miramos la captura de mas arriba y vemos que es con POST /api/v1/auth/login

```
http://${HOST}:${PUERTO}/api/v1/auth/login
```

Creamos el grupo de hilo para simular. Editar > Añadir > Hilos(Usuarios) > Grupos de Hilos



Ponemos 4 hilos y con 50 iteraciones

Grupo de Hilos

Nombre:

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

Periodo de Subida (en segundos):

Contador del bucle: ☐ Sin fin

☒ Same user on each iteration

☐ Retrasar la creación de Hilos hasta que se necesiten

☐ Planificador

Duración (segundos)

Retardo de arranque (segundos)

Ahora creamos la petición http que van a hacer ese grupo de hilos, por lo tanto es post al ser login, usamos las variables de HOST y PUERTO y luego añadimos login y pass que las necesitaremos para hacer el post de los alumnos.

Editar > Añadir > Muestrador > Petición HTTP

Petición HTTP

Nombre:

Comentarios

Basic **Advanced**

Servidor Web

Protocolo: Nombre de Servidor o IP: Puerto:

Petición HTTP

Método: Ruta: Codificación del contenido

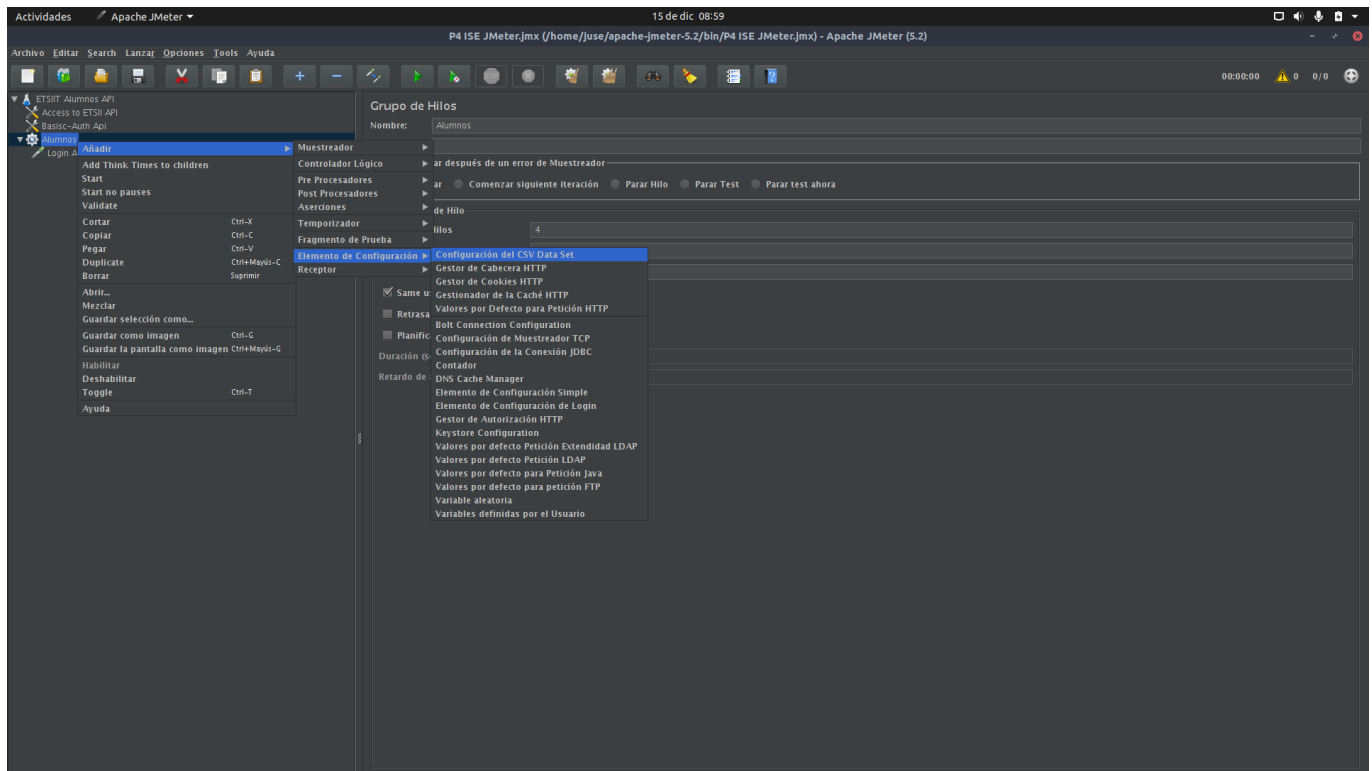
☐ Redirigir Automáticamente
 ☒ Seguir Redirecciones
 ☒ Utilizar KeepAlive
 ☐ Usar 'multipart/form-data' para HTTP POST
 ☐ Cabeceras compatibles con navegadores

Parameters **Body Data** **Files Upload**

Enviar Parámetros Con la Petición:

Nombre:	Valor	¿Codificar?	Content-Type
login	<input type="text" value="\${login}"/>	<input checked="" type="checkbox"/>	text/plain
pass	<input type="text" value="\${pass}"/>	<input checked="" type="checkbox"/>	text/plain

Ahora añadimos un elemento de configuración para las credenciales del archivo csv como nos dicen. Editar > Añadir > Elemento de Configuración > Configuración del CSV Data Set



Le indicas como leer del archivo y de donde (ruta). El delimitador es la coma, la primera línea es para el nombre de las variables por lo tanto lo ponemos a TRUE.

Configuración del CSV Data Set

Nombre: Credenciales Alumnos

Comentarios:

Configura el Data Source de CSV

Nombre de Archivo:	/home/juse/Escritorio/Cuarto/1°Cuatri/ISE/Practicas/P4/iseP4JMeter/jMeter/alumnos.csv
Codificación del fichero:	
Nombres de Variable (delimitados por coma):	
Utilice la primera línea como Nombres de Variable:	<input checked="" type="checkbox"/> True
Delimitador (utilice '\t' para poner un tabulador):	,
¿Permitir datos entrecomillados?:	<input type="checkbox"/> False
¿Reciclar en el fin de fichero (EOF)?:	<input type="checkbox"/> False
¿Para el hilo al final del fichero (EOF)?:	<input checked="" type="checkbox"/> True
Modo compartido:	Actual grupo de hilos

Cuando un usuario hace Login, se le devuelve un token (indicado en la API) y por tanto necesitamos procesarlo. Para ello usaremos expresiones regulares.

Buscando en internet y comparando [8](#) encontré esta expresión regular [6](#) y asegurando su validez con referencia [7](#) acabé utilizando:

```
^[A-Za-z0-9-_=]+\.\.[A-Za-z0-9-_=]+\.[A-Za-z0-9-_.+/=]*$
```

Basicamente tiene tres partes, separadas entre si por puntos estas partes aceptan cualquier letra, simbolo de ahí la parte de la expresion regular de `^[A-Za-z0-9-_=]+` repetida tres veces con `.` entre si (el punto). Este

token nos los devolverá si la petición POST ha sido ejecutada correctamente, viene en el mensaje de la respuesta a la petición.

Para añadirla sobre la peticion login alumno

click derecho Login Alumno > Añadir > PostProcesadores > Extractor de Expresiones Regulares.

Extractor de Expresiones Regulares

Nombre:

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 ☐ Requ

Nombre de Referencia:

Expresión Regular:

Plantilla:

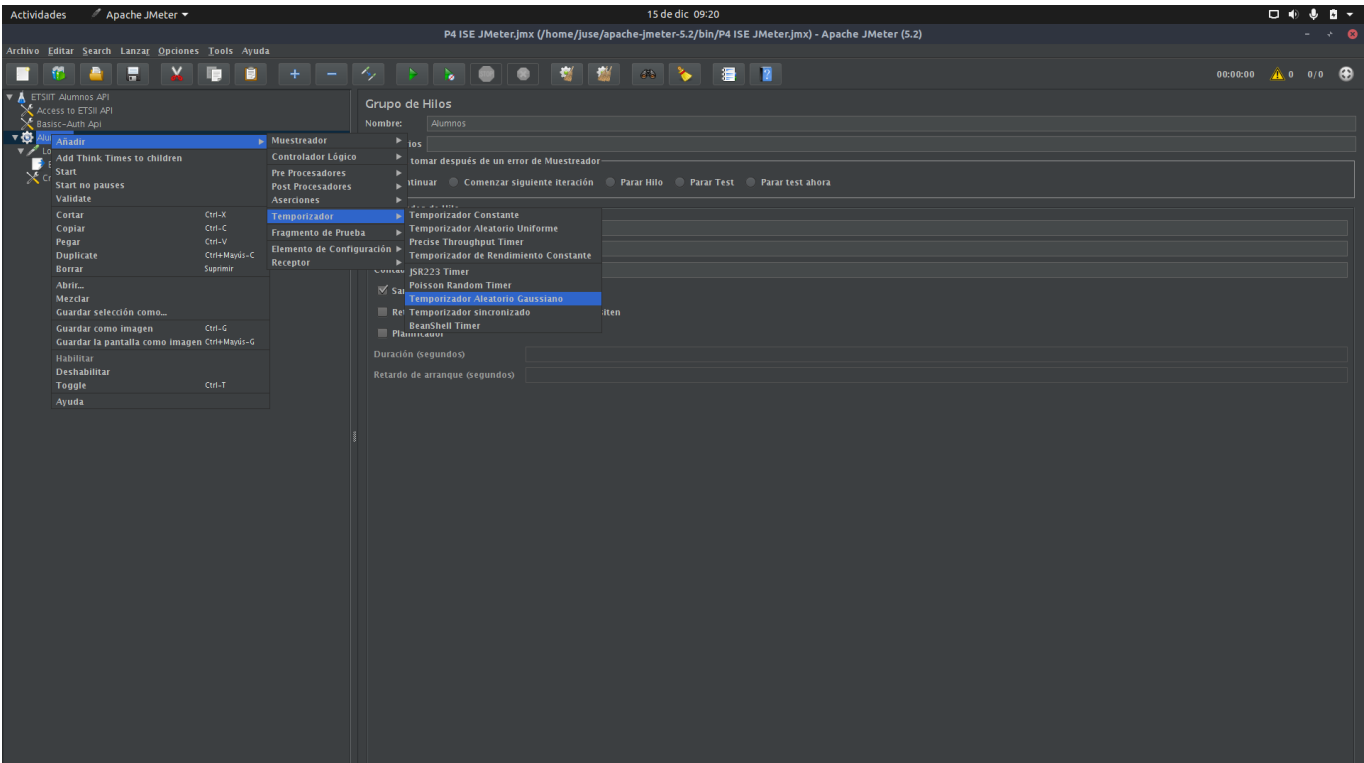
Coincidencia No. (0 para Aleatorio):

Valor por defecto: ☐ Use empty default value

y en nombre de la referencia ponemos el nombre donde se guardara la variable

Después añadimos un temporizador tal y como nos piden

Editar > Añadir > Temporizador > Temporizador Aleatorio Gaussiano



Con la sesión ya iniciada hacemos un GET con nuestro login (Petición HTTP) llamado Recuperar Datos Alumno.

Petición HTTP

Nombre: Petición HTTP

Comentarios

Basic Advanced

Servidor Web

Protocolo: http Nombre de Servidor o IP: \${HOST} Puerto: \${PORT}

Petición HTTP

Método: GET Ruta: /api/v1/alumnos/alumno/\${login} Codificación

☐ 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
login	\${login}	<input type="checkbox"/>	text/plain
pass	\${pass}	<input type="checkbox"/>	text/plain

Para entender por qué se pone bearer en el token he usado la referencia [9](#)

Ponemos una cabecera HTTP con el valor del token en la petición

P4 ISE JMeter.jmx (/home/juse/apache-jmeter-5.2/bin/P4 ISE JMeter.jmx) - Apache JMeter (5.2)

Archivo Editar Search Lanzar Opciones Tools Ayuda

Gestor de Cabecera HTTP

Nombre: JWT Token

Comentarios

Cabeceras Almacenadas en el Gestor de Cabeceras

Nombre:	Valor
Authorization	Bearer\${alumnoToken}

Tenía errores en configuración SCV Data Set donde no tenía puesto el nombre de las variables delimitado por comas; login, pass en mi caso

Otro error que tenía era que ponía \${PORT} cuando la había definido como PUERTO en las variables definidas por el usuario

Administradores

Al igual que con los usuarios, creamos un grupo de hilos y le indicamos de donde coge las credenciales (con CSV Data Set Configuration)

Grupo de Hilos

Nombre:

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:

Periodo de Subida (en segundos):

Contador del bucle: ☐ Sin fin

☒ Same user on each iteration
☐ Retrasar la creación de Hilos hasta que se necesiten
☐ Planificador

Duración (segundos)

Retardo de arranque (segundos)

Le indicas como coger los datos del CSV, las variables que son, la primera linea la ignora y la ruta

Configuración del CSV Data Set

Nombre:

Comentarios

Configura el Data Source de CSV

Nombre de Archivo:
 Codificación del fichero:
 Nombres de Variable (delimitados por coma):
 Utilice la primera línea como Nombres de Variable: ☒
 Delimitador (utilice '\t' para poner un tabulador):
 ¿Permitir datos entrecomillados?: ☐
 ¿Reciclar en el fin de fichero (EOF)?: ☐
 ¿Para el hilo al final del fichero (EOF)?: ☒
 Modo compartido:

Añadimos la petición http al igual que con usuario

Petición HTTP

Nombre:

Comentarios

Basic Advanced

Servidor Web

Protocolo: Nombre de Servidor o IP:

Petición HTTP

Método: Ruta:

☐ 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?	Contenido	
login	<input style="border: 1px solid #ccc;" type="text" value="\${login}"/>	<input checked="" type="checkbox"/>	text/plain	
pass	<input style="border: 1px solid #ccc;" type="text" value="\${pass}"/>	<input checked="" type="checkbox"/>	text/plain	

Creamos el extractor de token (lo he copiado de alumno ya que es igual, editar > copiar y luego editar > pegar en login Administrador)

Extractor de Expresiones Regulares

Nombre:

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
- ☐ Requiere

Nombre de Referencia:

Expresión Regular:

Plantilla:

Coincidencia No. (0 para Aleatorio):

Valor por defecto: ☐ Use empty default value

Metemos el temporizador

Temporizador Aleatorio Gaussiano

Nombre:

Comentarios:

Propiedades de Retardo de Hilos:

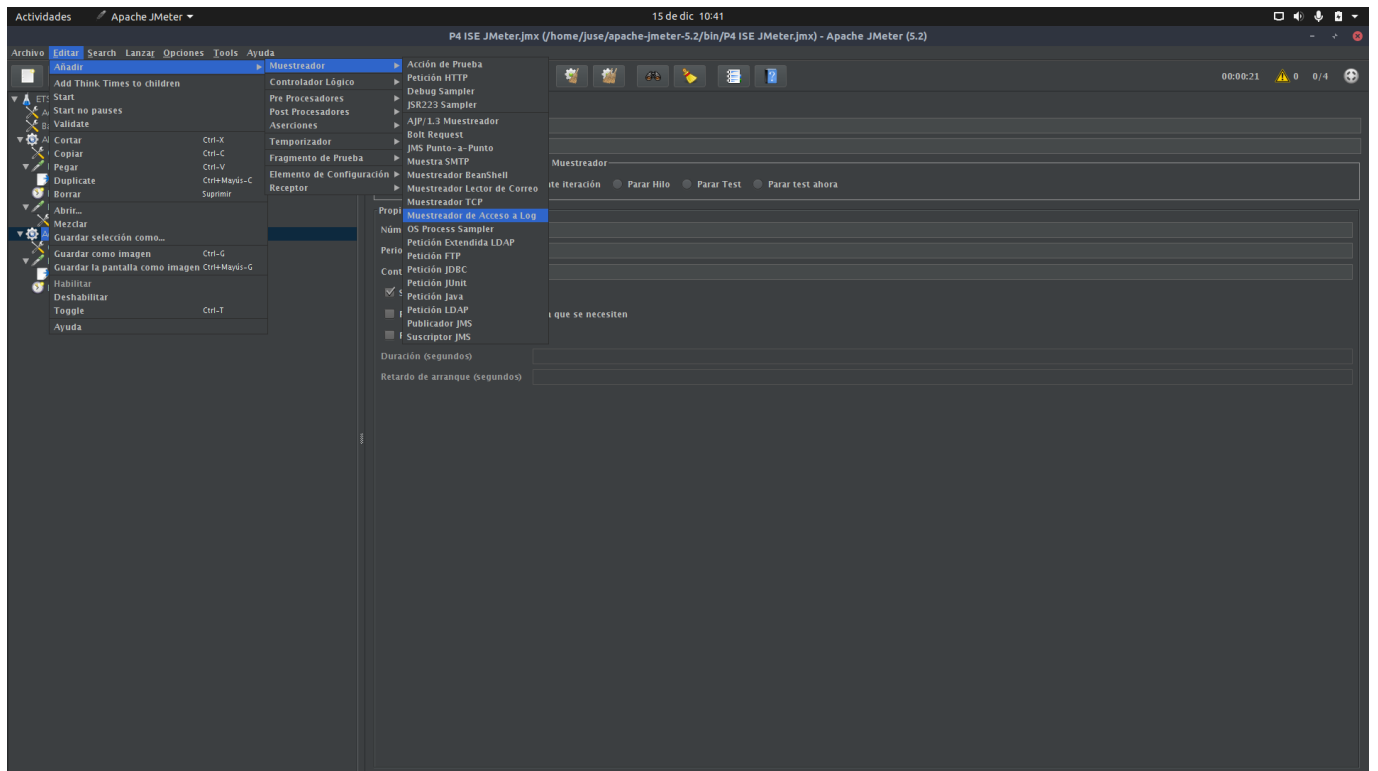
Desviación (en milisegundos):

Desplazamiento para Retardo Constante (en milisegundos):

Me di cuenta buscando por internet, que es se puede hacer lo mismo que con la expresion regular de arriba pero con .+ (así aseguras que sea vacio con ese +)y tambien te aseguras que sea un JWT Token, sin embargo mantuve la que te tenía porque funcionaba. (<https://www.regextester.com/105777> aqui se comprueba)

Ahora y tal y como se indica en el pdf de la practica, añadimos el Access Log Sampler ()

Añadir > muestreador > muestreador de acceso a log



Indicamos archivo del log, puerto y host

Muestreador de Acceso a Log

Nombre: Acceso Administradores

Comentarios:

Valores por defecto para Prueba

Protocol: http

Servidor: \${HOST}

Puerto: \${PUERTO}

Parsear Imágenes: False

Clases desplegables

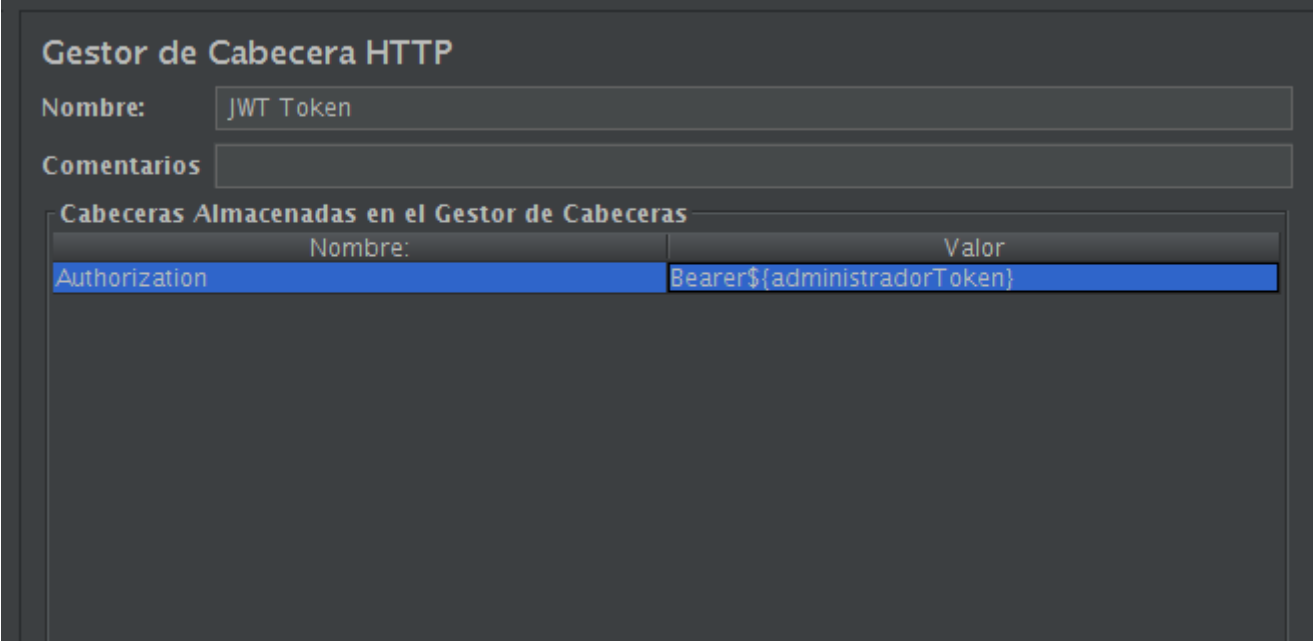
Parser: org.apache.jmeter.protocol.http.util.accesslog.TCLogParser

Filtro (Opcional): No definido

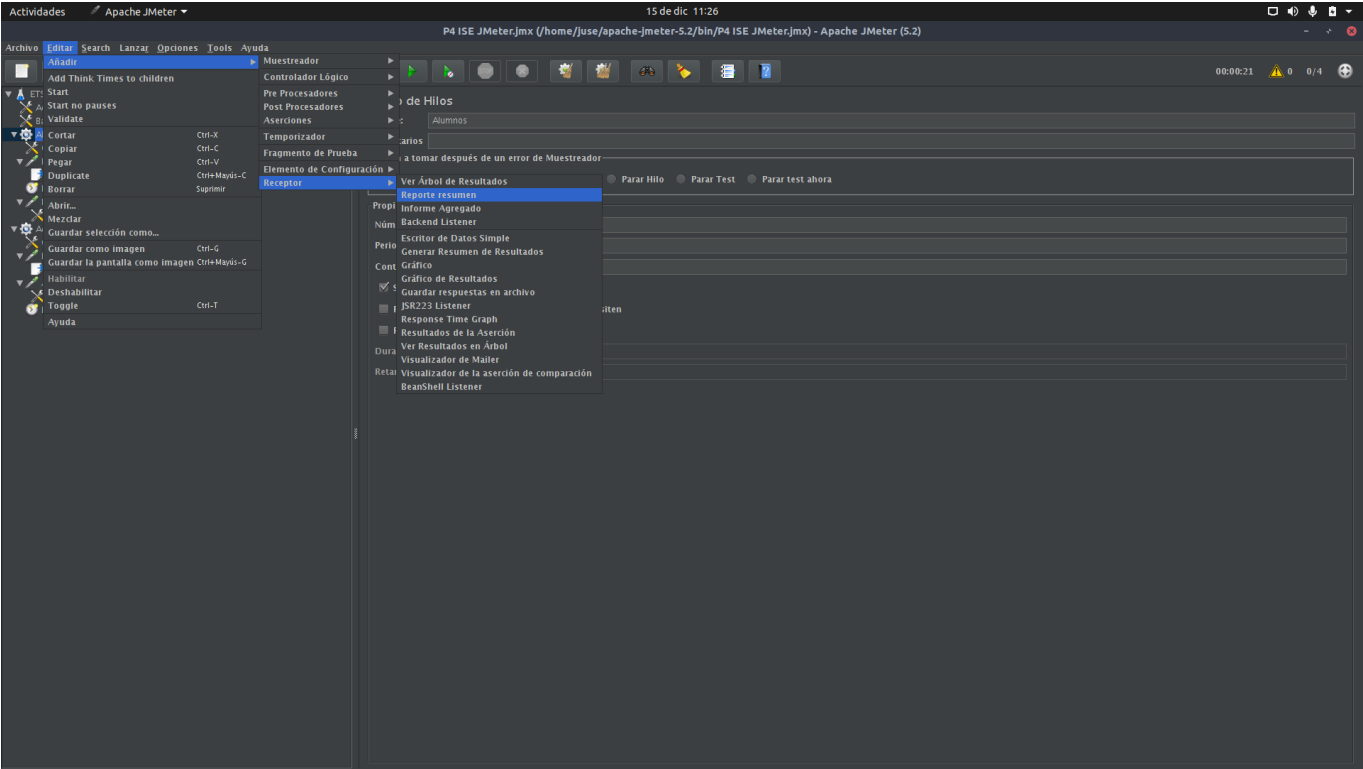
Ubicación del Archivo de Log

Archivo de Log: /home/juse/Escritorio/Cuarto/1°Cuatri/ISE/Practicas/P4/iseP4JMeter/jMeter/apiAlumnos.log

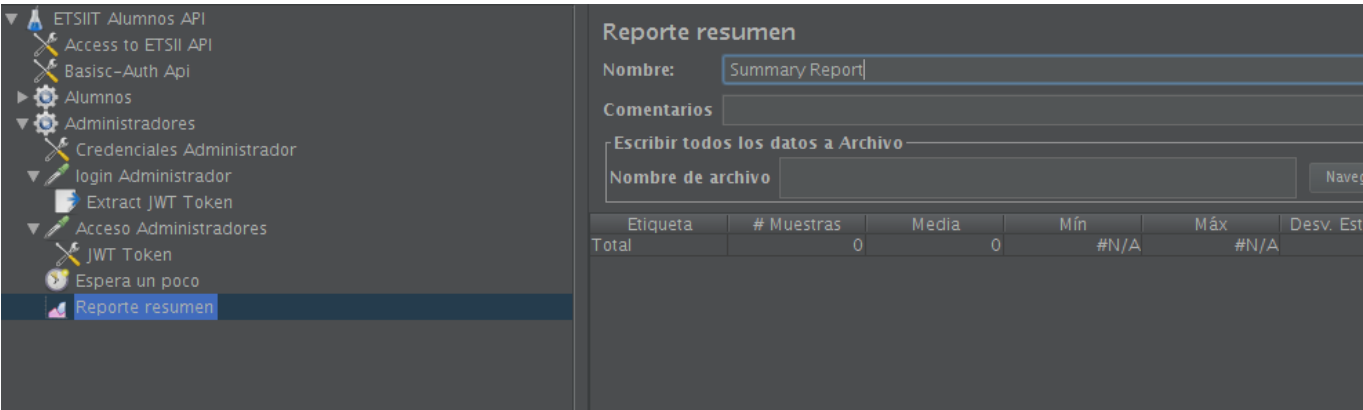
Para hacer uso de las peticiones que hay en el archivo .log, necesitamos el token de administrador, para eso hacemos igual que con usuarios pero con administradorToken



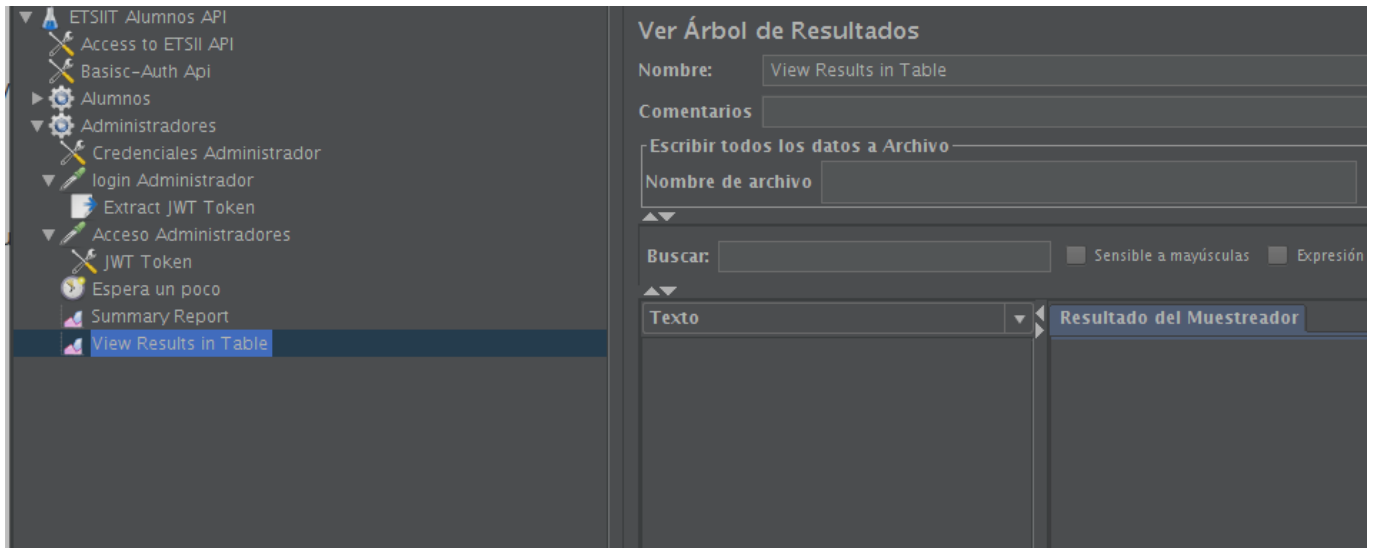
Aqui añadimos, reporte resumen, informe agregado y ver resultados en árbol.



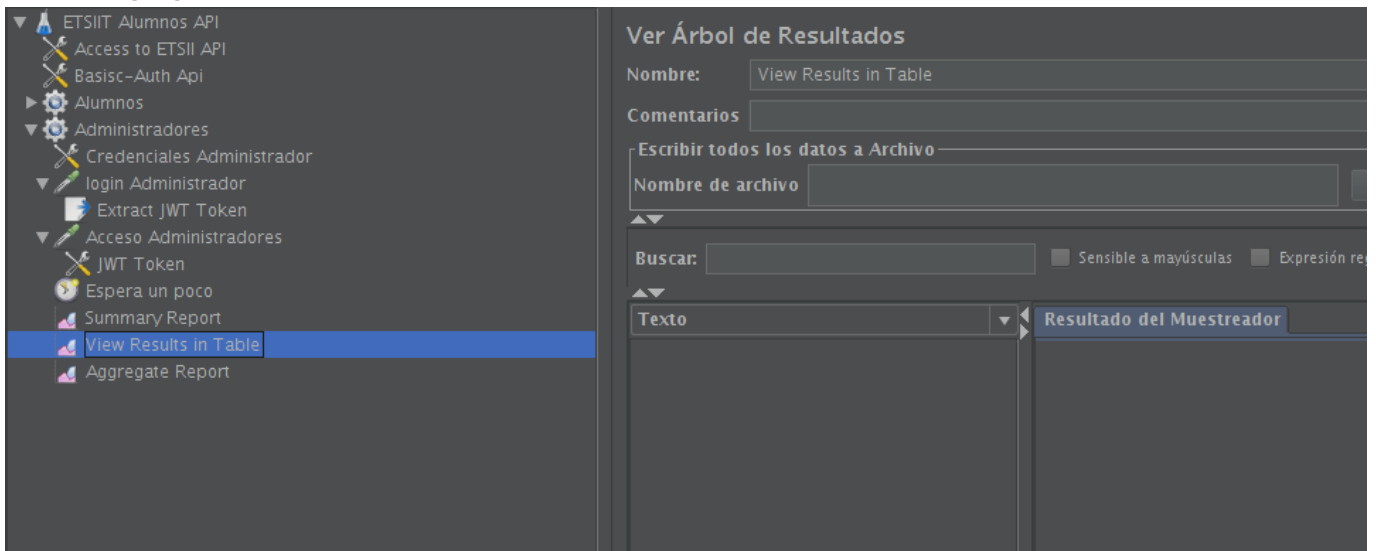
resumen del reporte



resultado en table



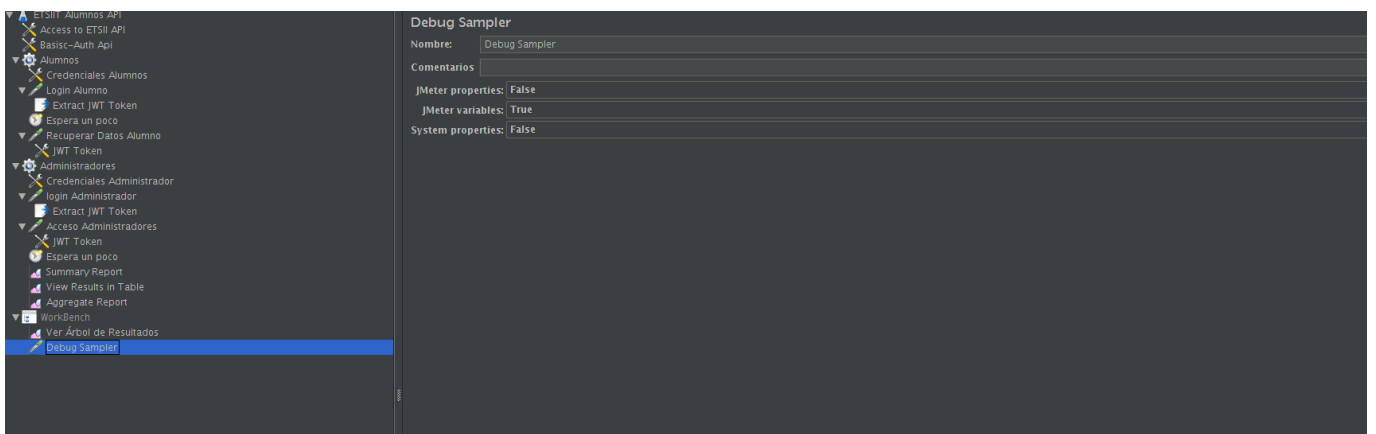
informe agregado



ahora añadimos el fragmento de prueba (añadir > fragmento de prueba > fragmento de prueba)

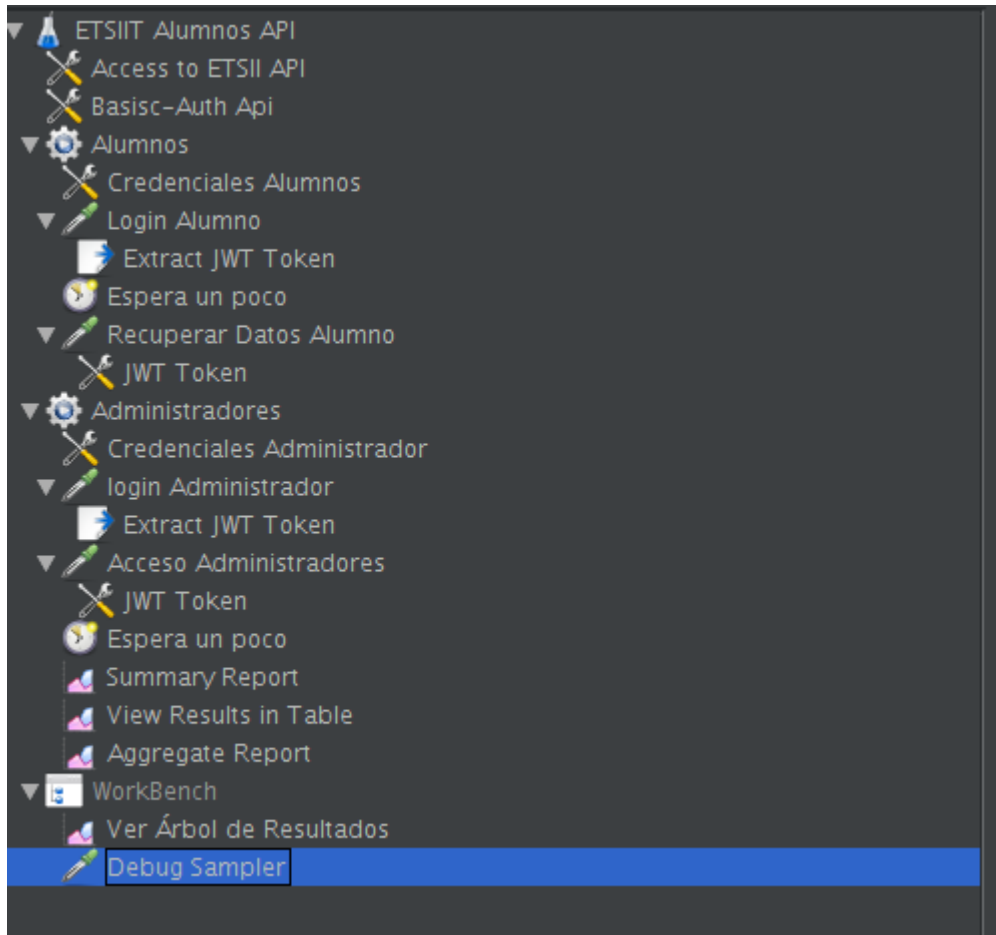
Despues creamos un arbol de resultados y un debug sampler para poder ver los resultados desde el programa de jmeter.

añadir > muestreador > debug sampler

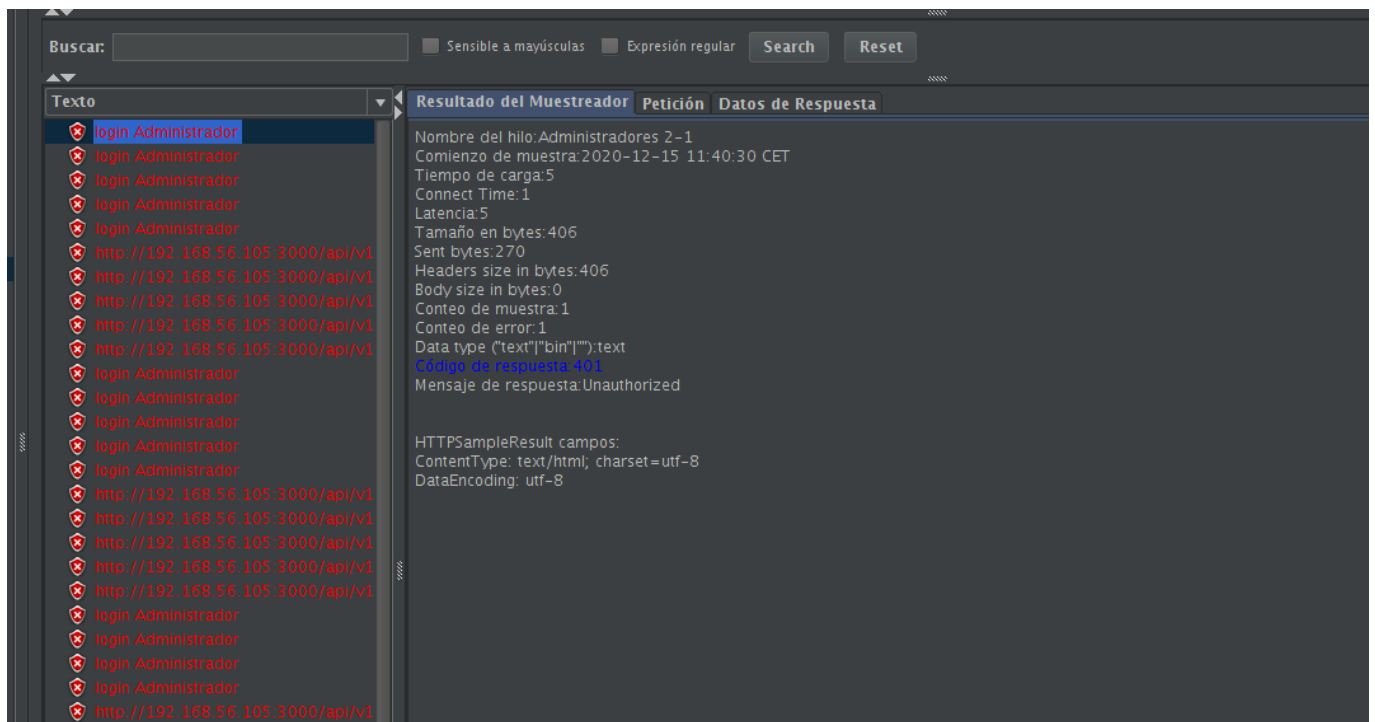


Lanzar Prueba

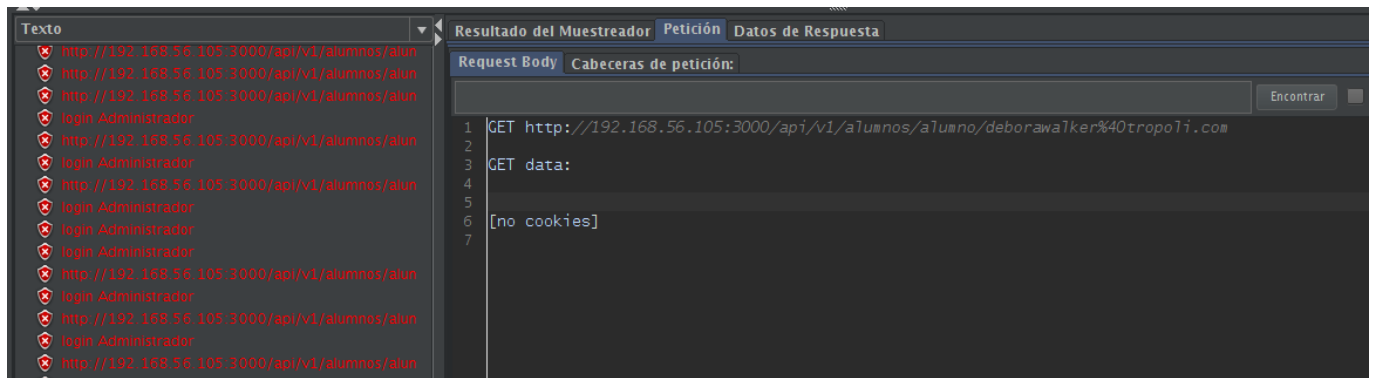
la prueba nos quedaría entonces como



Tras lanzar la prueba con jmeter, me devolvía un error



Cambie el nombre variables de las credenciales de administradores. Miré dentro de view results in table, en petición por si no cogía bien los valores del archivo csv



En response body no me salía nada, entonces cambié la expresión regular. Me di cuenta de que el error estaba en credenciales administrador que tenía puesto para todos los hilos cuando debería de ser para el actual grupo de hilos.

Aun despues de esto, seguía con el error 401 error unauthorized, ese error es de no estar autorizado y no entendia por qué ese error ya que me coge de forma correcto los valores del archivo administradores.csv con su contraseña, la ruta está puesta tal y como se pide.

Probé con una extension de mozilla para hacer el post por si era cosa de jmeter, me instale otra version de jmeter con sudo apt-get install

The screenshot shows the 'HTTP request Maker' application window. It has a dark title bar with the application name and window controls. Below the title bar is a blue header with two tabs: 'Request' (selected) and 'Response'. The main content area is white and contains several sections: 'Target Site:' with a text input field containing 'about:newtab'; 'Method:' with a dropdown menu showing 'GET'; 'Request Headers:' with a text area containing 'Content-Type:application/x-www-form-urlencoded', 'User-Agent:Mozilla', and 'Accept:/*/*'; and 'Body Data:' with an empty text area. At the bottom of the form are two blue buttons: 'Load' and 'Submit'.

Viendo que no era nada de esto, concluí que tenía que ser cosa de la autenticación en la conexión con la API al inicio, ya que del archivo de administradores lo cogía todo bien.

Recordé que no había usado al final la codificación 64 de BasicAuth, decidí codificar los valores que se le pasaba por cabecera a la petición de la API, entonces añadí un gestor de cabecera http, y añadí una variable Authorization con valor

```
ZXRzaWlBcGk6bGFBCG1EZUxhRVRTSU1EYUxhY2hl
```

codificado de

```
etsiiApi: laApiDeLaETSIIDaLache
```

tal y como muestra la url de <http://192.168.56.105:3000/> con la pagina <https://decodebase64.com/>.

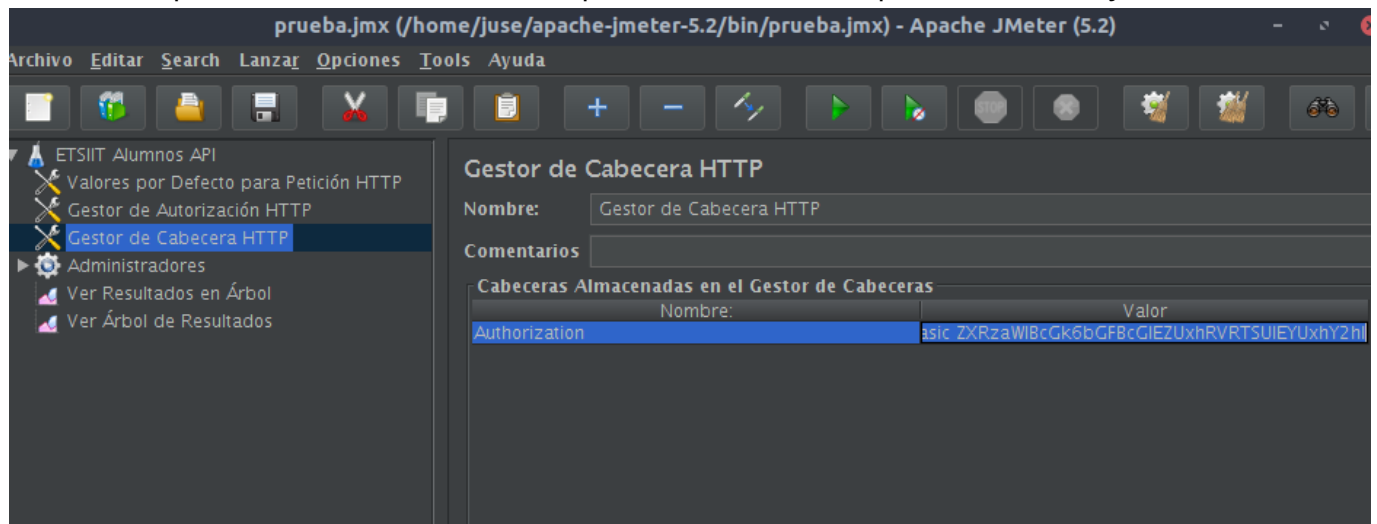
Tambien tenia en basi auth api

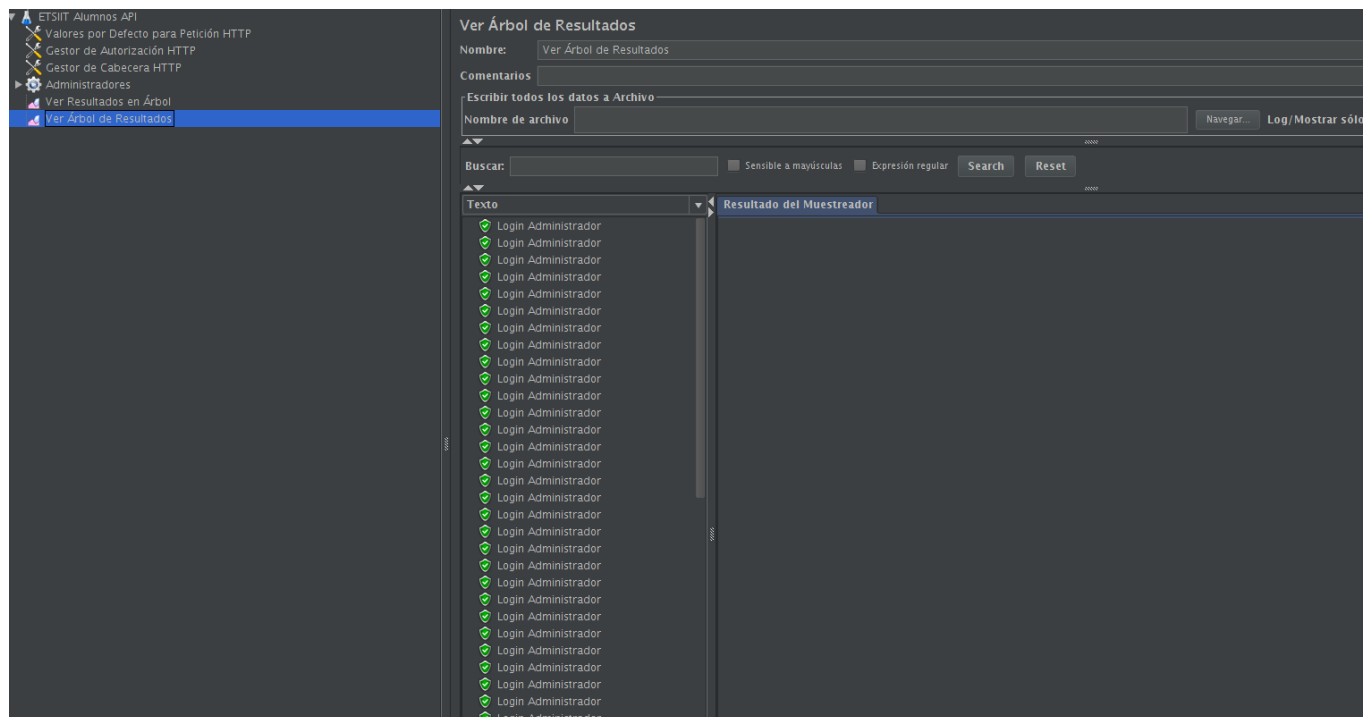
```
http://${HOST}:${PUERTO}/api/v1/auth/login
```

y era

```
http://${HOST}:${PUERTO}/api/auth/login
```

Realice unas pruebas mas basicas con ciertos parametros cambiados para encontrar mejor el error



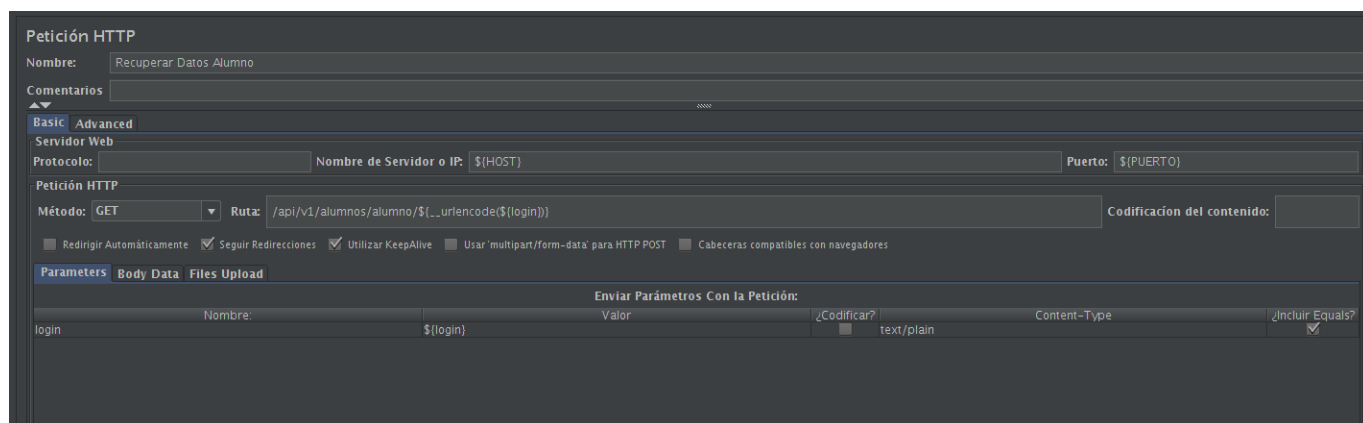


Despues lo pasé al archivo .jmx qal completo

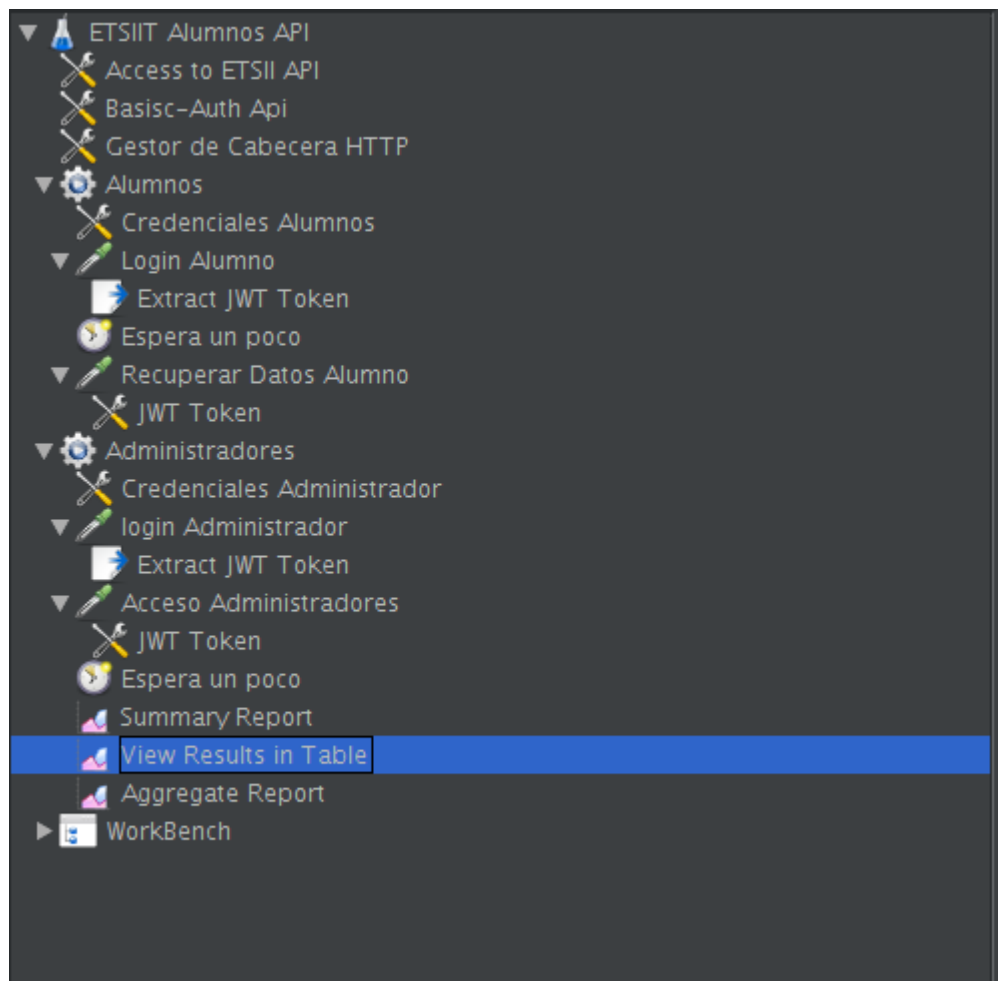
El error del login estaba arreglado, ahora me daba el error con las peticiones de apiAlumnos.log.

Al igual que con administrador, en el nombre tengo que seguir la nomenclatura de github, que es login password y lo cambié. Además a la hora de hacer el login con alumno es necesario pasarle urlencode ya que al observar los errores y lo que se obtenía de las peticiones, hacía falta pasarle urlencode [10](#).

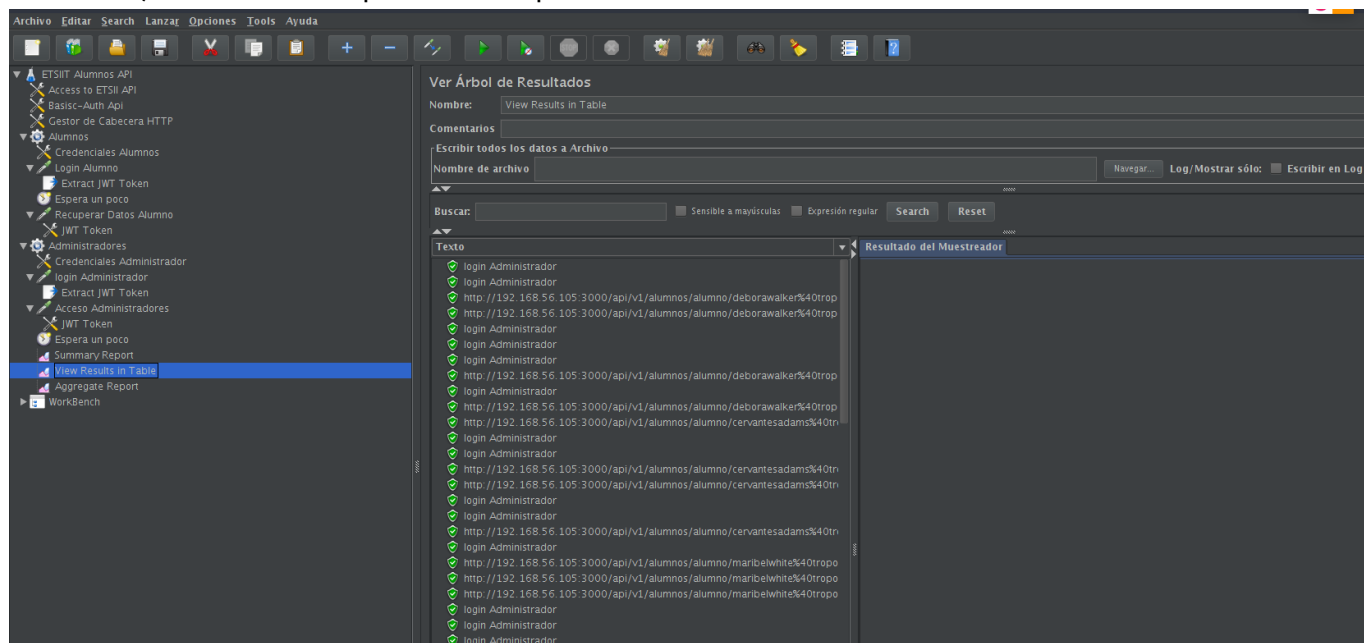
```
error -> /api/v1/alumnos/alumno/${__urlencode(${login})}
```



Al final quedaría la prueba como:



Al lanzarla, se obtiene la respuesta de las peticiones de forma correcta en las distintas vistas



Nombre de archivo					Navegar...		Log/Mostrar sólo: <input type="checkbox"/> Escribir en Log Sólo Errores <input type="checkbox"/> Éxitos		Configurar	
Etiqueta	# Muestras	Media	Mín	Máx	Desv. Estándar	% Error	Rendimiento	Kb/sec	Sent KB/sec	Media de Bytes
login Administrador	60	16	3	77	12,74	0,00%	6,5/sec	3,78	2,14	595
http //192.168.56...	4	16	14	18	1,79	0,00%	6,7/sec	7,44	0,00	1141
http //192.168.56...	4	17	7	27	7,33	0,00%	7,7/sec	9,09	0,00	1206
http //192.168.56...	4	18	15	26	4,32	0,00%	5,8/sec	8,76	0,00	1539
http //192.168.56...	4	15	11	19	3,34	0,00%	4,3/sec	8,25	0,00	1978
http //192.168.56...	4	6	5	9	1,64	0,00%	3,9/sec	3,96	0,00	1043
http //192.168.56...	4	14	7	17	4,06	0,00%	3,8/sec	5,74	0,00	1542
http //192.168.56...	4	25	13	58	18,82	0,00%	3,8/sec	4,17	0,00	1126
http //192.168.56...	4	39	23	83	25,15	0,00%	3,1/sec	2,88	0,00	950
http //192.168.56...	4	18	4	28	8,97	0,00%	2,5/sec	3,33	0,00	1379
http //192.168.56...	4	19	7	35	10,54	0,00%	2,4/sec	2,41	0,00	1050
http //192.168.56...	4	27	8	46	15,71	0,00%	3,4/sec	3,36	0,00	999
http //192.168.56...	4	30	25	40	5,77	0,00%	3,7/sec	5,71	0,00	1565
http //192.168.56...	4	11	4	19	5,41	0,00%	3,8/sec	4,13	0,00	1120
http //192.168.56...	4	26	5	85	33,80	0,00%	4,1/sec	8,48	0,00	2109
http //192.168.56...	4	8	6	11	1,92	0,00%	4,6/sec	5,13	0,00	1143
Total	120	18	3	85	14,52	0,00%	12,6/sec	11,86	2,09	960

Etiqueta	# Muestras	Media	Mediana	90% Line	95% Line	99% Line	Mín	Máx	% Error	Rendimiento	Kb/sec	Sent KB/sec
login Administra...	60	16	14	25	47	50	3	77	0,00%	6,5/sec	3,78	2,14
http://192.168...	4	16	15	18	18	18	14	18	0,00%	6,7/sec	7,44	0,00
http://192.168...	4	17	16	27	27	27	7	27	0,00%	7,7/sec	9,09	0,00
http://192.168...	4	18	16	26	26	26	15	26	0,00%	5,8/sec	8,76	0,00
http://192.168...	4	15	13	19	19	19	11	19	0,00%	4,3/sec	8,25	0,00
http://192.168...	4	6	5	9	9	9	5	9	0,00%	3,9/sec	3,96	0,00
http://192.168...	4	14	16	17	17	17	7	17	0,00%	3,8/sec	5,74	0,00
http://192.168...	4	29	14	50	50	50	13	50	0,00%	3,8/sec	4,17	0,00
http://192.168...	4	39	25	83	83	83	23	83	0,00%	3,6/sec	2,88	0,00
http://192.168...	4	18	17	28	28	28	4	28	0,00%	2,5/sec	3,33	0,00
http://192.168...	4	19	13	35	35	35	7	35	0,00%	2,4/sec	2,41	0,00
http://192.168...	4	27	16	46	46	46	8	46	0,00%	3,4/sec	3,36	0,00
http://192.168...	4	30	27	40	40	40	25	40	0,00%	3,7/sec	5,71	0,00
http://192.168...	4	11	10	19	19	19	4	19	0,00%	3,8/sec	4,13	0,00
http://192.168...	4	26	7	85	85	85	5	85	0,00%	4,1/sec	8,48	0,00
http://192.168...	4	8	8	11	11	11	6	11	0,00%	4,6/sec	5,13	0,00
Total	120	18	14	31	47	83	3	85	0,00%	12,6/sec	11,86	2,09

la info de los alumnos de las peticiones de `apiAlumnos.log`

The screenshot shows the Burp Suite interface. On the left, the 'Text' pane displays a list of HTTP requests. The 11th request is selected and highlighted in blue. It is a GET request to the URL: `http://192.168.56.105:3000/api/v1/alumnos/alumno/cervantesadams%40tr`. The right pane shows the 'Response Body' of this request, which is a JSON object containing user information.

Request List (Text Pane):

- login Administrador
- login Administrador
- http://192.168.56.105:3000/api/v1/alumnos/alumno/deborawalker%40tropo
- http://192.168.56.105:3000/api/v1/alumnos/alumno/deborawalker%40tropo
- login Administrador
- login Administrador
- login Administrador
- http://192.168.56.105:3000/api/v1/alumnos/alumno/deborawalker%40tropo
- login Administrador
- http://192.168.56.105:3000/api/v1/alumnos/alumno/deborawalker%40tropo
- http://192.168.56.105:3000/api/v1/alumnos/alumno/cervantesadams%40tr
- login Administrador
- login Administrador
- login Administrador
- http://192.168.56.105:3000/api/v1/alumnos/alumno/cervantesadams%40tr
- login Administrador
- login Administrador
- http://192.168.56.105:3000/api/v1/alumnos/alumno/cervantesadams%40tr
- login Administrador
- http://192.168.56.105:3000/api/v1/alumnos/alumno/maribelwhite%40tropo
- http://192.168.56.105:3000/api/v1/alumnos/alumno/maribelwhite%40tropo
- http://192.168.56.105:3000/api/v1/alumnos/alumno/maribelwhite%40tropo
- login Administrador
- login Administrador

Response Body (Selected Request):

```
{
  "id": "5fd89d0922ab1d2fcd08dfe1",
  "nombre": "Cervantes",
  "apellidos": "Alvarez Adams",
  "sexo": "male",
  "email": "cervantesadams@tropoli.com",
  "fechaNacimiento": "1984-03-06T00:00:00.000Z",
  "comentarios": "Eiusmod qui minim irure consequat eiusmod reprehenderit minim aliquip ea aliquip proident minim labore. Tempor consequat cillum reprehenderit qui in. Officia cillum sit cupidatat reprehenderit deserunt quis tempor amet sit adipisicing veniam ut. Nisi eu enim ipsum fugiat eu in ididunt Lorem quis ea Duis occaecat tempor. Do anim non aliquip veniam. Veniam qui ex et enim ut in occaecat et magna adipisicing mollit occaecat eiusmod. Officia cillum consectetur commodo tempor veniam fugiat sit laborum anim qui.\n(n",
  "curso": "1",
  "media": 7.1,
  "usuario": "14"
}
```

Con esto y como índice github, En la URL <http://:3000/status> se puede monitorizar el estado de carga del servidor NodeJS.

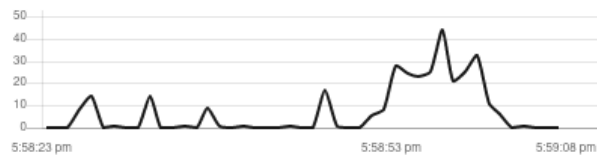
escribimos: `http://192.168.56.105:3000/status`

Express Status

1M 5M 15M

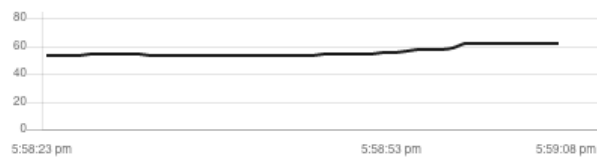
CPU Usage

0.0%



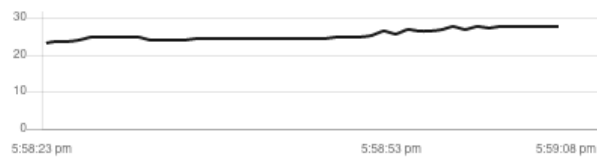
Memory Usage

62.0MB



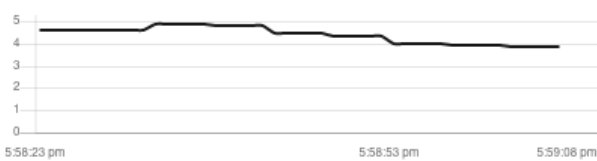
Heap Usage

27.7MB



One Minute Load Avg

3.82



Spent in Event Loop

5.00ms



Response Time

10.52ms



Requests per Second

22.00



Status Codes

- 2xx
- 3xx
- 4xx
- 5xx



Conclusiones

De este ejercicio no solo me voy con ver el potencial de JMeter para pruebas de carga, si no que he aprendido bastante de las peticiones HTTP sobre servicios (cosa que no hemos dado en toda la carrera ahora mismo). Además encuentro fascinante docker y docker-compose como herramientas para levantar y bajar microservicios, así como su gran verstaibilidad (tema del que intentaré hacer un trabajo opcional en enero).

Referencias

- (1):<http://www.gtlib.gatech.edu/pub/apache/jmeter/binaries/>
- (2):<https://github.com/davidPalomar-ugr/iseP4JMeter>

- (3):<https://medium.com/@cudakara/how-to-install-jmeter-in-ubuntu-20-04-7e4d30c42551>
- (4):<https://askubuntu.com/questions/827114/what-to-do-about-update-initramfs-errors-during-kernel-upgrade>
- (5):https://en.wikipedia.org/wiki/Basic_access_authentication
- (6):<https://github.com/auth0/node-jsonwebtoken/issues/162>
- (7):<https://www.regextester.com/105777>
- (8):<https://stackoverflow.com/questions/62750614/check-if-bearer-token-is-jwt-or-not>
- (9):<https://www.developerro.com/2019/03/12/jwt-api-authentication/>
- (10):https://www.w3schools.com/tags/ref_urlencode.ASP