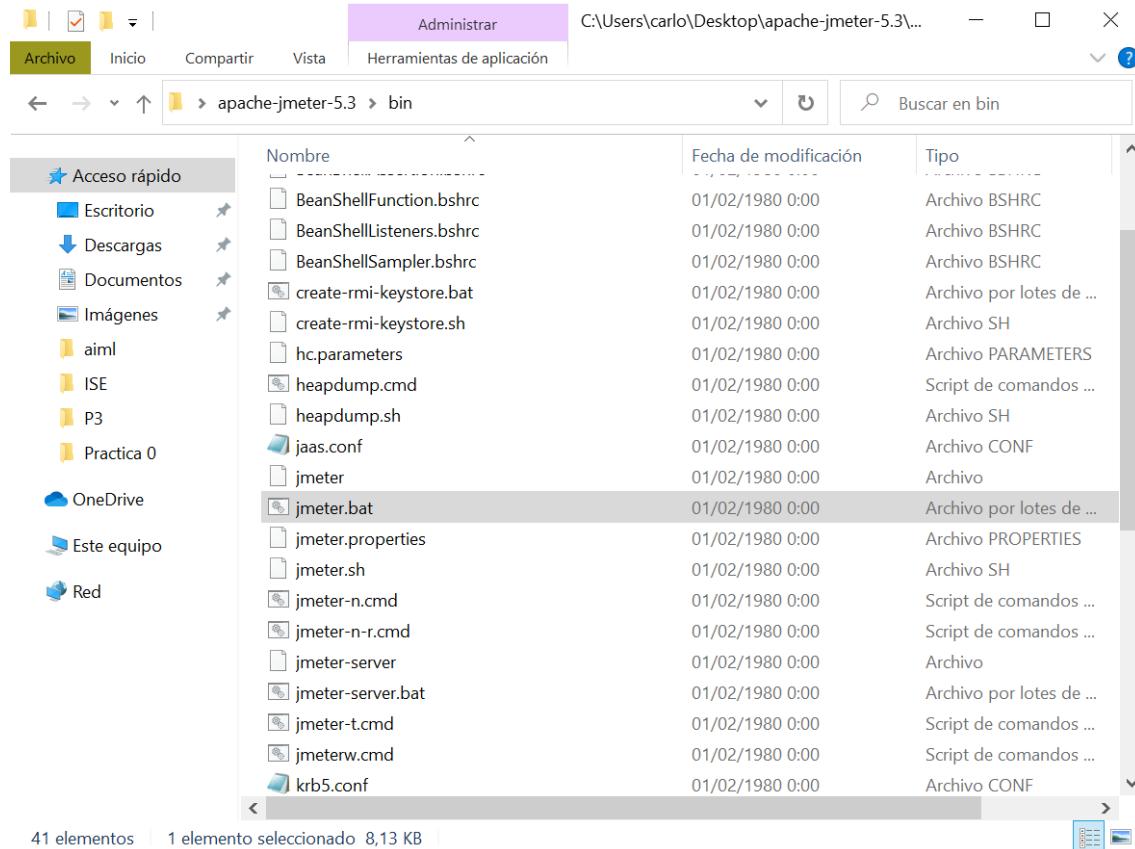


Lo primero que vamos a hacer es instalar JMeter en la máquina host para ello nos metemos en su [página web](#) y lo descargamos.



Ahora vamos a descargar tanto Docker como Docker Compose en nuestra máquina virtual con Ubuntu Server.

Primero vamos a instalar todas las dependencias necesarias con

```
apt-get install apt-transport-https ca-certificates curl gnupg-agent software-properties-common
```

Acto seguido instalaremos la key GPG oficial de Docker para validar la integridad de lo que hemos instalado.

Con este comando `sudo apt-key fingerprint 0EBFCD88` como se indica en la [página de Docker](#) comprobamos si la key es la correcta

```

[carlosgs@p2-ubuntu: ~]
Desempaquetando python3-software-properties (0.96.20.9) sobre (0.96.20.7) ...
Procesando disparadores para man-db (2.7.5-1) ...
Procesando disparadores para libc-bin (2.23-0ubuntu10) ...
Procesando disparadores para dbus (1.10.6-1ubuntu3.3) ...
Configurando libcurl3-gnutls:amd64 (7.47.0-1ubuntu2.14) ...
Configurando curl (7.47.0-1ubuntu2.14) ...
Configurando apt-transport-https (1.2.32ubuntu0.1) ...
Configurando ca-certificates (20190110~16.04.1) ...
Configurando libassuan0:amd64 (2.4.2-2) ...
Configurando pinentry-curses (0.9.7-3) ...
Configurando libnpth0:amd64 (1.2-3) ...
Configurando gnupg-agent (2.1.11-6ubuntu2.1) ...
Configurando python3-software-properties (0.96.20.9) ...
Configurando software-properties-common (0.96.20.9) ...
Procesando disparadores para libc-bin (2.23-0ubuntu10) ...
Procesando disparadores para ca-certificates (20190110~16.04.1) ...
Updating certificates in /etc/ssl/certs...
10 added, 31 removed; done.
Running hooks in /etc/ca-certificates/update.d...
done.
carlosgs@p2-ubuntu:~$ sudo curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -
[sudo] password for carlosgs:
OK
carlosgs@p2-ubuntu:~$ sudo apt-key fingerprint 0EBFCD88
pub   4096R/0EBFCD88 2017-02-22
      Huella de clave = 9DC8 5822 9FC7 DD38 854A E2D8 8D81 803C 0EBF CD88
uid            Docker Release (CE deb) <docker@docker.com>
sub   4096R/F273FCDB 2017-02-22
carlosgs@p2-ubuntu:~$
```

Lo siguiente que debemos hacer es añadir el repositorio con

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

```

[carlosgs@p2-ubuntu: ~]
Configurando python3-software-properties (0.96.20.9) ...
Configurando software-properties-common (0.96.20.9) ...
Procesando disparadores para libc-bin (2.23-0ubuntu10) ...
Procesando disparadores para ca-certificates (20190110~16.04.1) ...
Updating certificates in /etc/ssl/certs...
10 added, 31 removed; done.
Running hooks in /etc/ca-certificates/update.d...
done.
carlosgs@p2-ubuntu:~$ sudo curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -
[sudo] password for carlosgs:
OK
carlosgs@p2-ubuntu:~$ sudo apt-key fingerprint 0EBFCD88
pub   4096R/0EBFCD88 2017-02-22
      Huella de clave = 9DC8 5822 9FC7 DD38 854A E2D8 8D81 803C 0EBF CD88
uid            Docker Release (CE deb) <docker@docker.com>
sub   4096R/F273FCDB 2017-02-22
carlosgs@p2-ubuntu:~$ sudo add-apt-repository \ "deb [arch=amd64] https://download.docker.com/linux/ubuntu \ $(lsb_release -cs) \ stable"
Error: « deb [arch=amd64] https://download.docker.com/linux/ubuntu \ xenial \ stable» es incorrecto
carlosgs@p2-ubuntu:~$ sudo add-apt-repository \ "deb [arch=amd64] https://download.docker.com/linux/ubuntu \ $(lsb_release -cs) \ stable"
Error: « deb [arch=amd64] https://download.docker.com/linux/ubuntu \ xenial \ stable» es incorrecto
carlosgs@p2-ubuntu:~$ sudo add-apt-repository
Error: se necesita un repositorio como argumento
carlosgs@p2-ubuntu:~$ sudo add-apt-repository \
> "deb [arch=amd64] https://download.docker.com/linux/ubuntu \ $(lsb_release -cs) \ stable"
carlosgs@p2-ubuntu:~$
```

Por último, hacemos un `apt-get update` instalamos docker con `apt-get install docker-ce Docker-ce-cli containerd.io`

Ahora usamos el comando `docker run hello-world` para comprobar que todo funciona correctamente

```

Selezionare carlosgs@p2-ubuntu: ~
carlosgs@p2-ubuntu:~$ sudo docker run hello-world
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/

```

Una vez tenemos instalado Docker vamos a instalar Docker Compose. Para ello lo primero que vamos a hacer es instalar el paquete con

```

sudo curl -L "https://github.com/docker/compose/releases/download/1.25.5/dockercompose-
$(uname -s)-$(uname -m)" -o /usr/local/bin/docker-compose

```

Ahora damos permisos de ejecución donde hemos instalado el paquete con

```

sudo chmod +x /usr/local/bin/docker-compose

```

Y ejecutamos `docker-compose --version` para asegurarnos de que vamos bien

```

Selezionare carlosgs@p2-ubuntu: /usr/local/bin
/usr/local/bin/docker-compose: 5: /usr/local/bin/docker-compose: request: not found
ls
docker-compose
carlosgs@p2-ubuntu: /usr/local/bin$ ./docker-compose
./docker-compose: linea 1: '$\r': orden no encontrada
./docker-compose: linea 2: head: No existe el archivo o el directorio
./docker-compose: linea 3: head: No existe el archivo o el directorio
./docker-compose: linea 4: meta: No existe el archivo o el directorio
./docker-compose: linea 5: mddot: orden no encontrado
./docker-compose: linea 5: title: No existe el archivo o el directorio
./docker-compose: linea 5: title: No existe el archivo o el directorio
./docker-compose: linea 6: meta: No existe el archivo o el directorio
./docker-compose: linea 6: style: No existe el archivo o el directorio
./docker-compose: linea 7: style: No existe el archivo o el directorio
./docker-compose: linea 8: body: orden no encontrado
./docker-compose: linea 9: background-color: orden no encontrada
./docker-compose: linea 10: error sintáctico cerca del elemento inesperado '('
./docker-compose: linea 10:     color: rgba(0, 0, 0, 0.5);
carlosgs@p2-ubuntu: /usr/local/bin$ nano docker-compose
Use wq para volver a nano. Use wq! para volver a nano.
[1]+  Detenido      nano docker-compose
carlosgs@p2-ubuntu: /usr/local/bin$ vi docker-compose
[No se ha escrito nada al disco desde el último cambio]
/bin/bash: q: orden no encontrada
El intérprete de órdenes devolvió 127
Pulse INTRO o escriba una orden para continuar
carlosgs@p2-ubuntu: /usr/local/bin$ sudo rm docker-compose
carlosgs@p2-ubuntu: /usr/local/bin$ sudo curl -L "https://github.com/docker/compose/releases/download/1.25.5/docker-compose-$(uname -s)-$(uname -m)" -o /usr/local/bin/docker-
-compose
% Total    % Received % Xferd  Average Speed   Time   Time   Current
          Dload  Upload Total Spent   Spent    Left  Speed
100  638  100  638    0     0  509      0  0:00:01  0:00:01  ---:--- 509
100 16.7M  100 16.7M    0     0 1678K      0  0:00:10  0:00:10  ---:--- 3163K
carlosgs@p2-ubuntu: /usr/local/bin$ sudo chmod +x /usr/local/bin/docker-compose
carlosgs@p2-ubuntu: /usr/local/bin$ sudo docker-compose --version
docker-compose version 1.25.5, build 8a1c60f6
carlosgs@p2-ubuntu: /usr/local/bin$ 

```

Lo siguiente que vamos a hacer es descargarnos la API que se nos da con

```

git clone https://github.com/davidPalomar-uqr/iseP4JMeter.git

```

Y nos metemos en `/usr/local/bin/iseP4JMeter`

```
Seleccinar carlosgs@p2-ubuntu: /usr/local/bin/iseP4JMeter
carlosgs@p2-ubuntu: /usr/local/bin$ sudo git clone https://github.com/davidPalomar-ugr/iseP4JMeter.git
[sudo] password for carlosgs:
Clonar en /iseP4JMeter...
remote: Enumerating objects: 8, done.
remote: Counting objects: 100% (8/8), done.
remote: Compressing objects: 100% (6/6), done.
remote: Total 3774 (delta 1), reused 4 (delta 1), pack-reused 3766
Receiving objects: 100% (3774/3774), 7.77 MiB | 1.49 MiB/s, done.
Resolving deltas: 100% (702/702), done.
Comprobando la conectividad... hecho.
Extrayendo archivos: 100% (73/73), done.
carlosgs@p2-ubuntu: /usr/local/bin$ cd /iseP4JMeter
-bash: cd: /iseP4JMeter: No existe el archivo o el directorio
carlosgs@p2-ubuntu: /usr/local/bin$ cd iseP4JMeter/
carlosgs@p2-ubuntu: /usr/local/bin/iseP4JMeter$
```

Una vez aquí hacemos `sudo docker-compose up`

```
Seleccinar carlosgs@p2-ubuntu: /usr/local/bin/iseP4JMeter
mongod_1 2020-06-03T20:14:57.563+0000 I INDEX [conn8] index build: collection scan done. scanned 1000 total records in 0 seconds
mongod_1 2020-06-03T20:14:57.578+0000 I INDEX [conn8] index build: inserted 1000 keys from external sorter into index in 0 seconds
mongod_1 2020-06-03T20:14:57.587+0000 I INDEX [conn8] index build: done building index id_1 on ns etsi.usuarios
mongodinit_1 {
    "createdCollectionAutomatically" : false,
    "numIndexesBefore" : 1,
    "numIndexesAfter" : 2,
    "ok" : 1
}
mongod_1 2020-06-03T20:14:57.599+0000 I NETWORK [conn8] end connection 172.18.0.4:53348 (0 connections now open)
mongod_1 2020-06-03T20:14:57.634+0000 I NETWORK [listener] connection accepted from 172.18.0.4:53350 #9 (1 connection now open)
mongod_1 2020-06-03T20:14:57.636+0000 I NETWORK [conn9] received client metadata from 172.18.0.4:53350 conn9: { driver: { name: "mongo-go-driver", version: "v1.2.1" }, os: { type: "linux", architecture: "amd64" }, platform: "go1.12.17" }
mongod_1 2020-06-03T20:14:57.639+0000 I NETWORK [listener] connection accepted from 172.18.0.4:53352 #10 (2 connections now open)
mongod_1 2020-06-03T20:14:57.644+0000 I NETWORK [conn10] received client metadata from 172.18.0.4:53352 conn10: { driver: { name: "mongo-go-driver", version: "v1.2.1" }, os: { type: "linux", architecture: "amd64" }, platform: "go1.12.17", application: { name: "mongoimport" } }
mongodinit_1 2020-06-03T20:14:57.646+0000 connected to: mongodb://mongodb/
mongodinit_1 2020-06-03T20:14:57.657+0000 20 document(s) imported successfully. 0 document(s) failed to import.
mongod_1 2020-06-03T20:14:57.664+0000 I NETWORK [conn9] end connection 172.18.0.4:53350 (1 connection now open)
mongod_1 2020-06-03T20:14:57.666+0000 I NETWORK [conn10] end connection 172.18.0.4:53352 (0 connections now open)
nodejs_1 (node:17) DeprecationWarning: current URL string parser is deprecated, and will be removed in a future version. To use the new parser, pass option { useNewUrlParser: true } to MongoClient.connect.
nodejs_1 (node:17) DeprecationWarning: current Server Discovery and Monitoring engine is deprecated, and will be removed in a future version. To use the new Server Discover and Monitoring engine, pass option { useUnifiedTopology: true } to the MongoClient constructor.
mongod_1 2020-06-03T20:14:57.777+0000 I NETWORK [listener] connection accepted from 172.18.0.3:41626 #11 (1 connection now open)
mongod_1 2020-06-03T20:14:57.991+0000 I NETWORK [conn11] received client metadata from 172.18.0.3:41626 conn11: { driver: { name: "nodejs", version: "3.5.8" }, os: { type: "Linux", name: "linux", architecture: "x64", version: "4.4.0-131-generic" }, platform: "'Node.js v8.17.0 , LE (legacy)'" }
iseP4JMeter_mongodinit_1 exited with code 0
```

Ahora abrimos el navegador de nuestro anfitrión y ponemos **192.168.82.105:3000** que es la dirección de la maquina de Ubuntu Server y el puerto que usa esta API y nos saldrá lo siguiente.



ETSII Alumnos API

```
Descripción de la API Restful:  
POST /api/v1/auth/login  
Parametros:  
    login:<emailusuario>  
    password:<secreto>  
Seguridad:  
    Acceso protegido con BasicAuth (etsiiApi:laApiDeLaETSIIDaLache)  
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 valido  
Retorna:  
    Objeto Json con perfil de alumno
```

Lo que indica que de momento todo va bien.

Vamos a probar el script que nos habéis dado para que esta funcionando.

Ejecutamos `./pruebaEntorno.sh`

P4 UBUNTU [Corriendo] - Oracle VM VirtualBox

Archivo Máquina Ver Entrada Dispositivos Ayuda

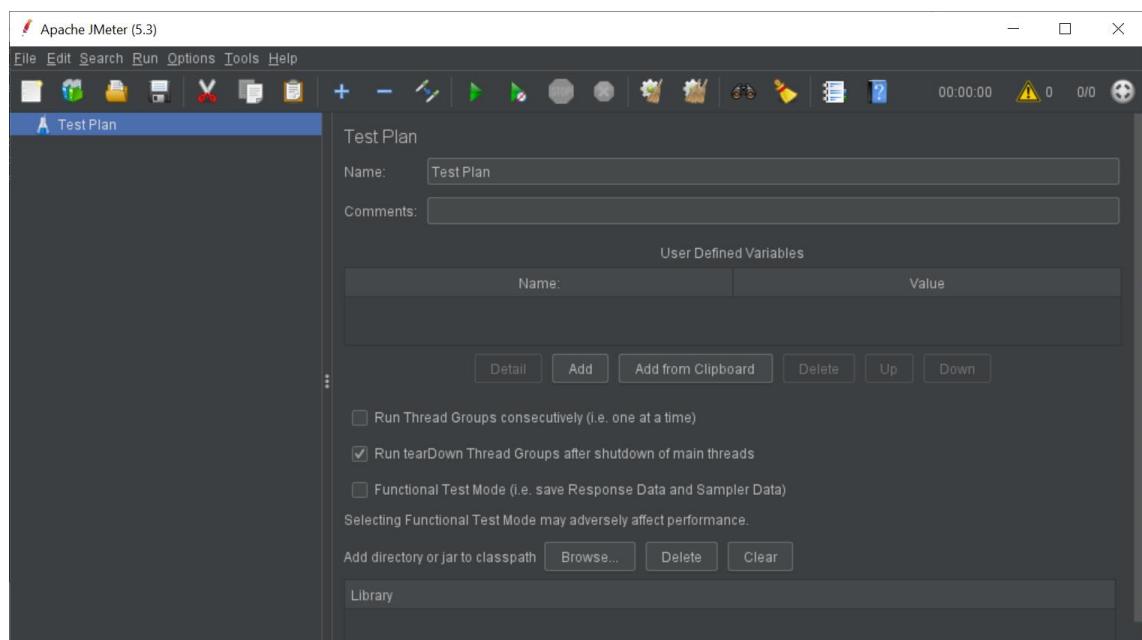
```
New release '18.04.4 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

ssh
usage: ssh [-1246AaCfGgKkMNnqsTtUvXxYy] [-b bind_address] [-c cipher_spec]
           [-D [bind_address:]port] [-E log_file] [-e escape_char]
           [-F configfile] [-I pkcs11] [-i identity_file] [-L address]
           [-l login_name] [-m mac_spec] [-O ctl_cmd] [-o option] [-p port]
           [-Q query_option] [-R address] [-S ctl_path] [-W host:port]
           [-w local_tun[:remote_tun]] [user@]hostname [command]
carlosgs@p2-ubuntu:~$ sudo install openssh-server
[sudo] password for carlosgs:
install: falta el operando archivo de destino después de 'openssh-server'
Pruebe 'install --help' para más información.
carlosgs@p2-ubuntu:~$ sudo apt install openssh-server
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
openSSH-server ya está en su versión más reciente (1:7.2p2-4ubuntu2.8).
0 actualizados, 0 nuevos se instalarán, 0 para eliminar y 178 no actualizados.
carlosgs@p2-ubuntu:~$ cd /usr/local/bin/iseP4JMeter/
carlosgs@p2-ubuntu:/usr/local/bin/iseP4JMeter$ ./pruebaEntorno.sh
{"_id": "5ed804bcc4e9b2c932cb7ed", "nombre": "Mari", "apellidos": "Fletcher Weiss", "sexo": "female", "email": "mariweiss@tropoli.com", "fechaNacimiento": "1992-04-04T00:00:00.000Z", "comentarios": "Aliquip dolor laboris ullamco id ex labore. Ipsum eiusmod ut aliquip non cillum deserunt sunt commodo anim ad nisi excepteur eu deserunt. Sit sunt proident Lorem irure irure minim adipisicing cillum. Nostrud officia in proident velit velit sit fugiat pariatur quis ad laboris minim dolor elit. Sint velit pariatur commodo sint veniam exercitation. Duis proident minim consequat consectetur sint et tempor labore culpa esse. Exercitation laborum non esse mollit tempor ea dolor minim adipisicing mollit in aliqua.\\nUllamco adipisicing excepteur commodo sunt nulla quis sunt velit Lorem pariatur sunt ad do incididunt. In eu nostrud ullamco laboris eu minim. Consequat sit et eiusmod officia ex sit minim sit laborum quis laborum labore non. Dolor nulla ut pariatur reprehenderit minim dolore consequat sunt aliquip ipsum esse. Excepteur consequat fugiat elit et nisi dolore aute minim nostrud et.\\r\\n", "cursos": [{"curso": 1, "media": 5.2}, {"curso": 2, "media": 9.1}], "usuario": 10}carlosgs@p2-ubuntu:/usr/local/bin/iseP4JMeter$ _
```

CTRL DERECHA

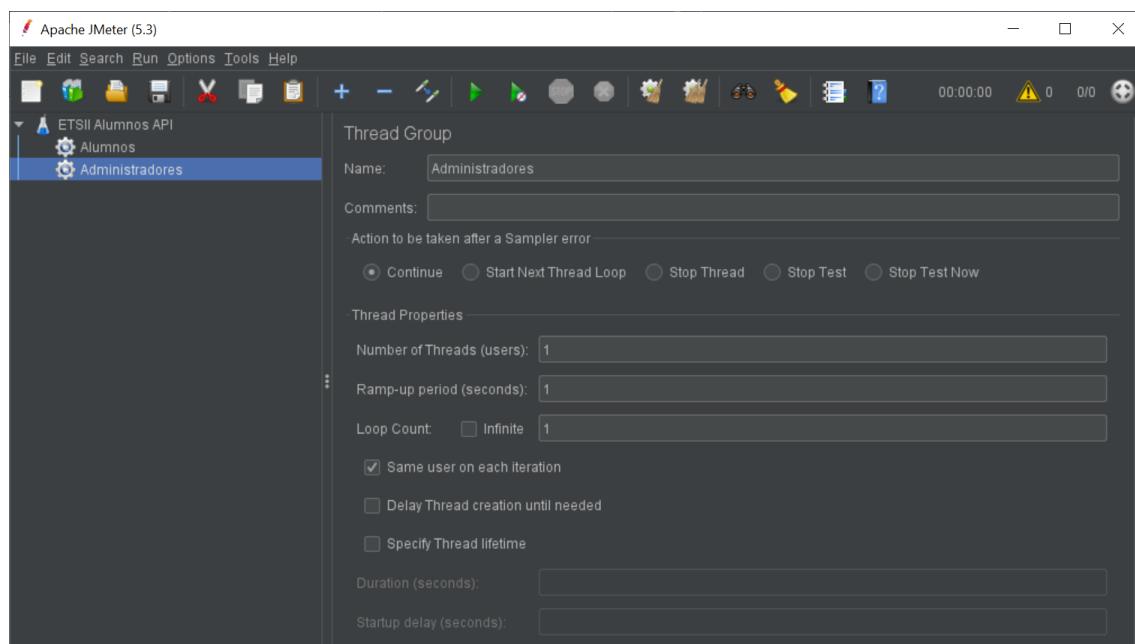
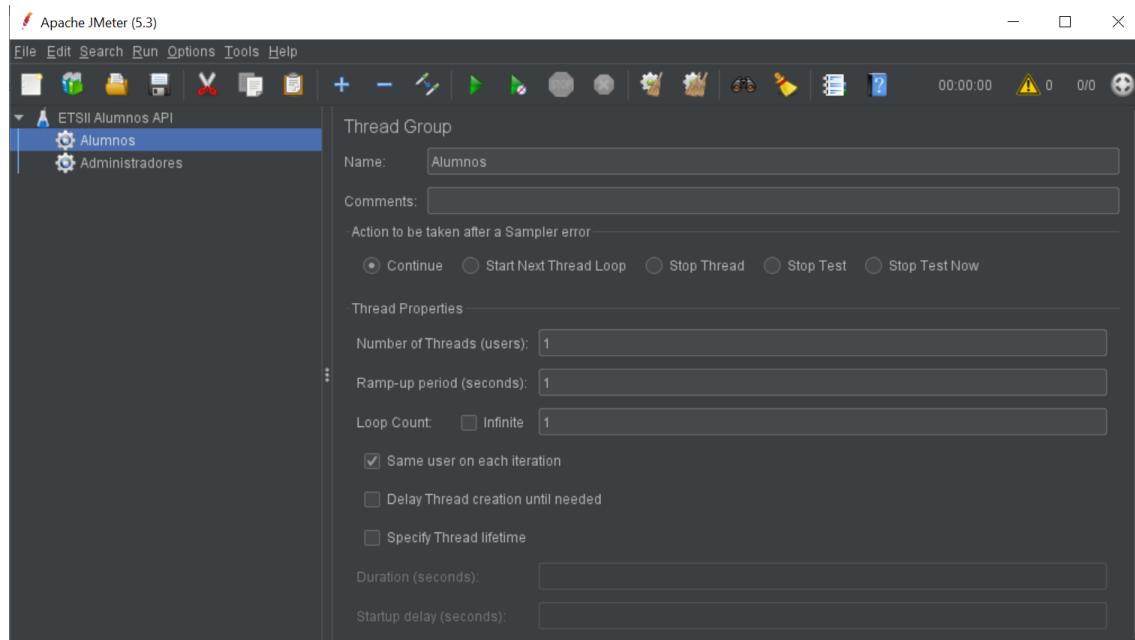
Esto es lo que debe de salir así que bien.

Ahora vamos a volver a la maquina host a usar JMeter, al estar en Windows 10 nosotros ejecutamos el archivo jmeter.bat y se nos abre lo siguiente.



Vamos primero a crear un nuevo test pinchando en *File/New* y le ponemos el nombre que queramos.

Ahora vamos a crear dos grupos, uno de Alumnos y otro de Admins. Para crear cada grupo pinchamos con el botón derecho en el test, le damos a *add/Thread(User)/ThreadGroup* Le cambiamos los nombres y dejamos los valores por defecto quedando algo así.



Ahora nos vamos a Alumno/Add/Sampler/"Http Request", le cambiamos el nombre que queramos, añadimos la ip de nuestro server y el puerto que usa, cambiamos el método a POST y ponemos el path indicado en la API, tambien en la documentación buscamos el login de algún alumno y lo ponemos en parámetros, Quedadno algo así.

The screenshot shows the Apache JMeter 5.3 interface with a test plan named "Test.jmx". The left sidebar shows a tree structure with "Alumnos" selected. In the main panel, an "HTTP Request" sampler is selected. The "Basic" tab is active. The "Web Server" section has "Protocol [http]" set to "http", "Server Name or IP:" set to "192.168.82.105", and "Port Number:" set to "3000". The "HTTP Request" section has "Method" set to "POST" and "Path" set to "/api/v1/auth/login". The "Parameters" tab is selected, showing two parameters: "login" with value "rowlandsauders@tropoli..." and "password" with value "exercitation". Both parameters have "Content-Type" set to "text/plain" and "Include Equals?" checked.

Vamos a añadir la autorización, para ello le damos a

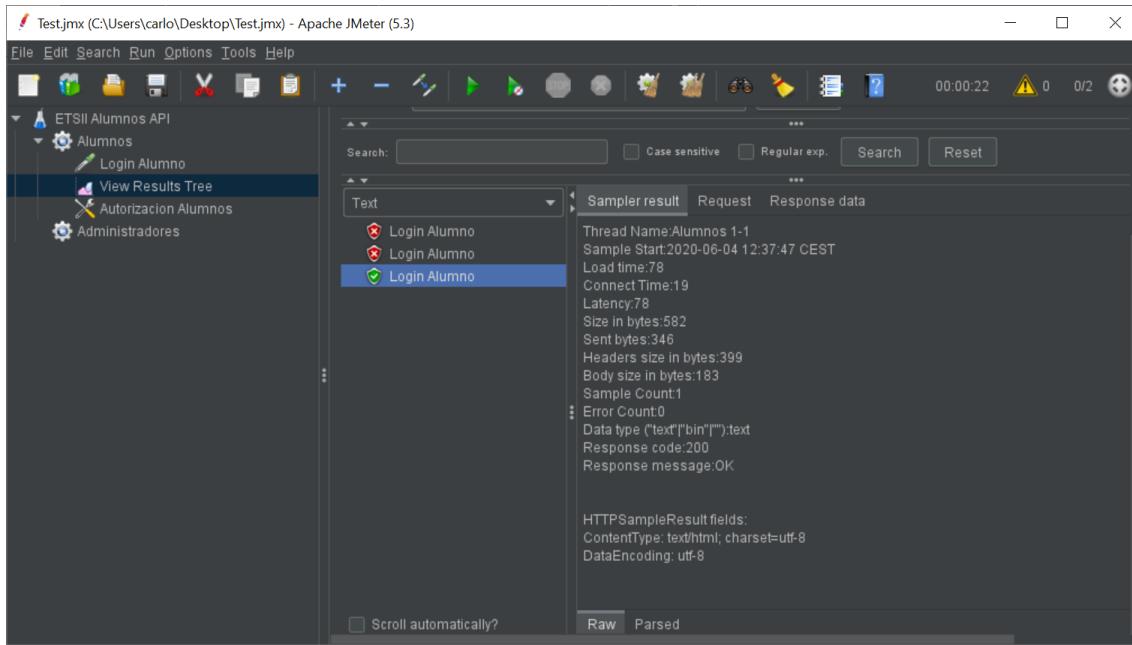
Alumnos/Add/Config Element/"HTTP Authorization Manager"

Le ponemos el nombre que queramos y le damos a add, ahí ponemos la ip de nuestro server, en nuestro caso 192.168.82.105:3000, el usuario “etsiiApi” y la contraseña “laApiDeLaETSIIDaLache” tal y como aparece [Aquí](#).

The screenshot shows the Apache JMeter 5.3 interface with the same test plan and sidebar. A new "HTTP Authorization Manager" element is selected in the tree. The "Name:" field is set to "Autorizacion Alumnos". The "Options" section has two checkboxes: "Clear auth on each iteration?" and "Use Thread Group configuration to control clearing", both unchecked. The "Authorizations Stored in the Authorization Manager" table shows one entry: "Base URL" is "http://192.168.82.105:3000/", "Username" is "etsiiApi", and "Password" is "laApiDeLaETSIIDaLache". The "Mechanism" column is "BASIC". At the bottom, there are "Add", "Delete", "Load", and "Save" buttons.

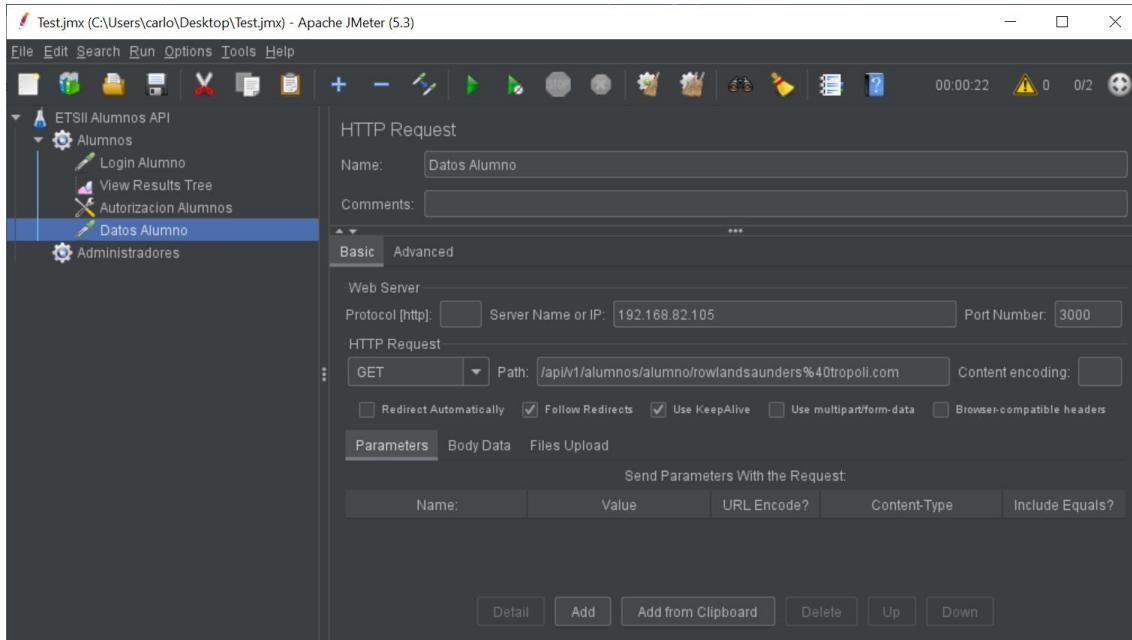
Vamos a comprobar que funciona, Le damos a *Alumnos/Add/Listener/"View Result Tree"* y ejecutamos el test pinchando en el triángulo verde.

Vemos que funciona



Ahora vamos a darle acceso a los alumnos a las calificaciones, para ello vamos a

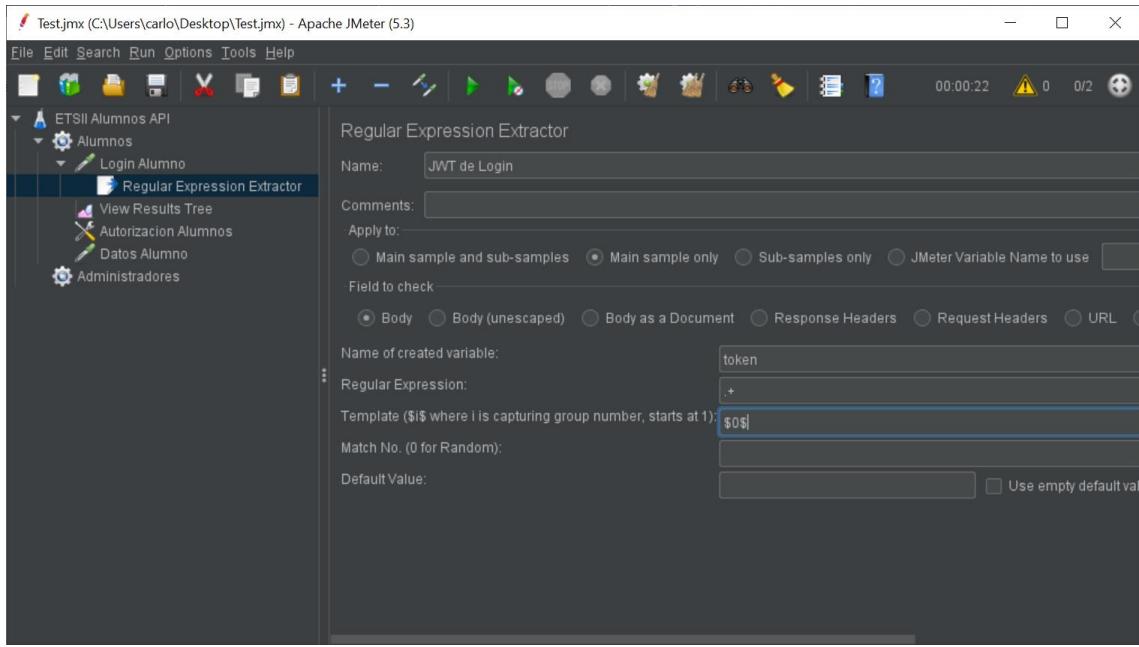
Alumnos/Add/Sampler/”HTTP Request” y modificamos los datos tal que así.



Vamos a comprobar que tenemos acceso a las calificaciones, para ello vamos a extraer el JWT y pasarlo a Datos Alumno. Para ello vamos a

“Login Alumno”/Add/”Post Processors”/”Regular Expression Extractor”

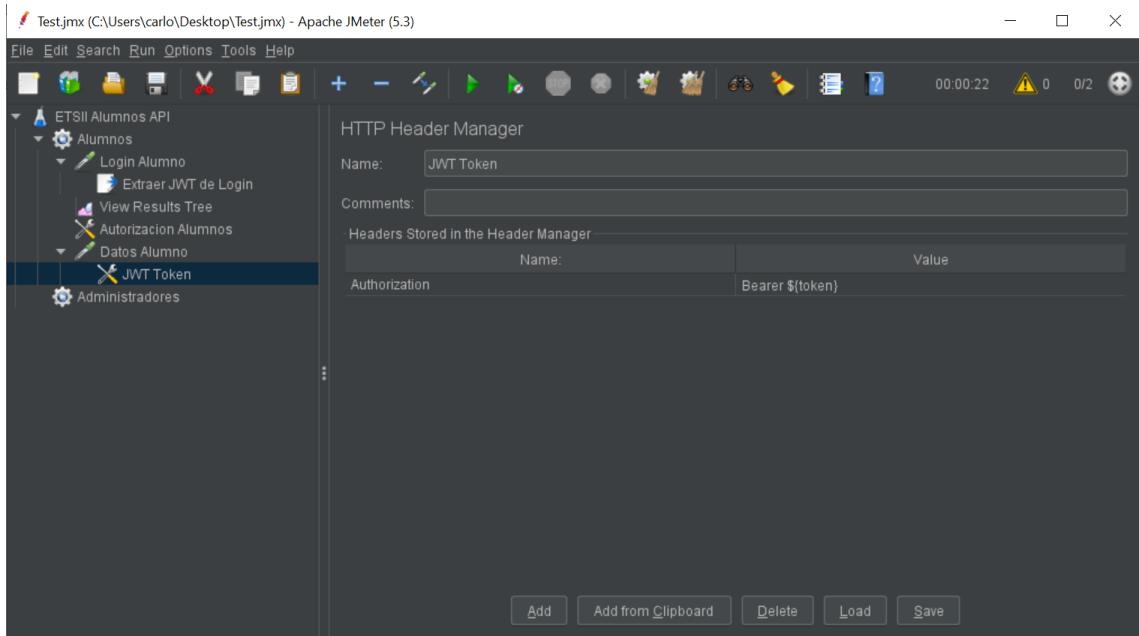
Y lo configuramos tal que así



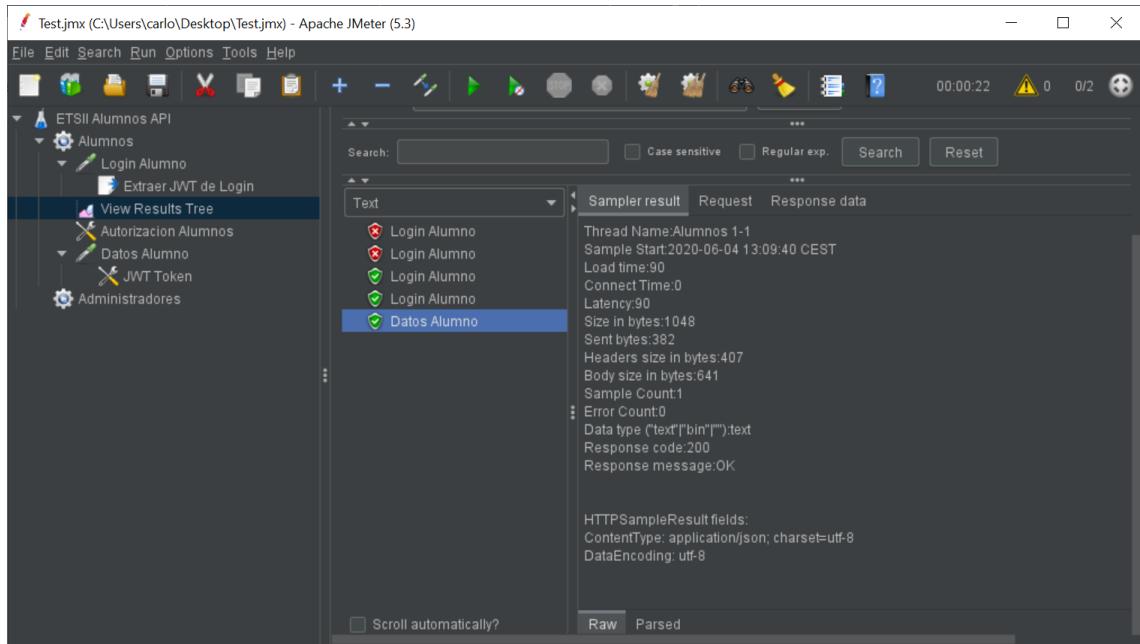
Una vez hecho esto se lo tenemos que pasar a Datos Alumno para ello hacemos

"Datos Alumnos/Add/"/Config Element"/"/HTTP Header Manager"

Le ponemos el nombre que queramos, le damos a Add, ponemos un nombre y en Value ponemos `Bearer ${token}` El token es porque asi se llama mi variable y el Bearer es por la estructura de la cabecera.



Vamos a Comprobar que todo esta funcionando. Volvemos a View Results Tree y pinchamos en el triángulo verde para ejecutar.

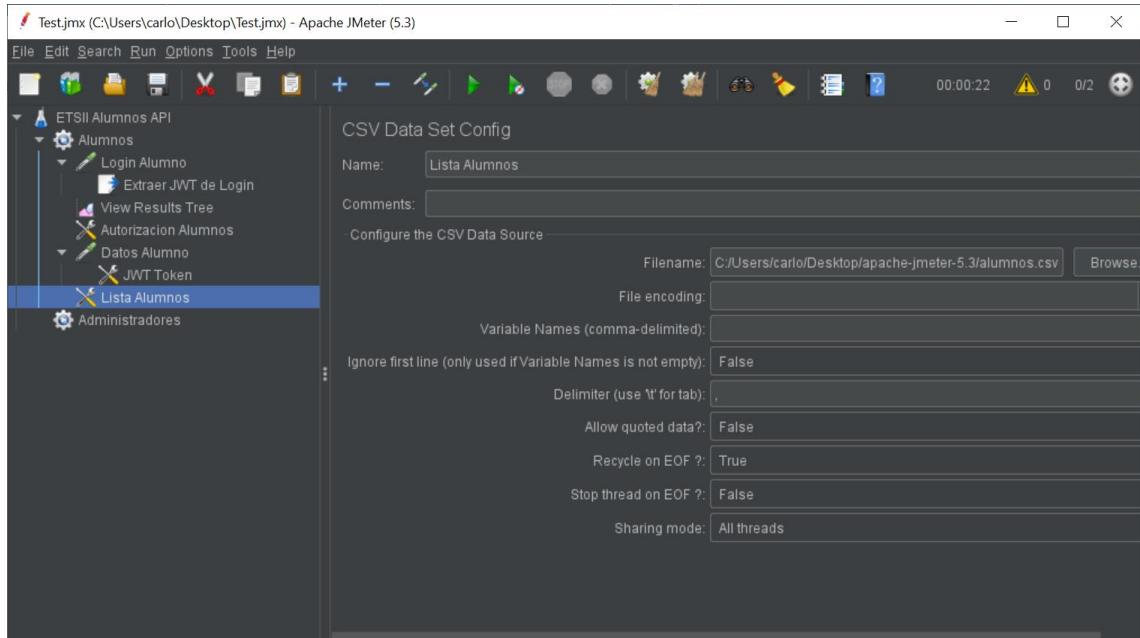


Vemos que ha funcionado.

Ahora vamos a añadir todos los alumnos de la lista, para ello vamos a

Alumnos/Add/”Config Element”/”Data Set Config”

Le pasamos el archivo alumnos.csv



Ahora vamos a cambiar en Login Alumno donde hemos puesto el usuario de ejemplo por \${login} en login y por \${password} en password.

The screenshot shows the Apache JMeter interface with the 'Test.jmx' file open. The left sidebar displays a tree structure of test plans and samplers. The 'Datos Alumno' sampler under the 'Alumnos' group is currently selected. The right panel is titled 'HTTP Request' and contains the configuration for this sampler. The 'Basic' tab is selected, showing the 'Web Server' settings (Protocol: http, Server Name or IP: 192.168.82.105, Port Number: 3000) and the 'HTTP Request' settings (Method: POST, Path: /api/v1/auth/login). Below these are sections for 'Parameters', 'Body Data', and 'Files Upload'. In the 'Parameters' section, there are two entries: 'login' with value '\${login}' and 'password' with value '\${password}'. Both parameters have 'URL Encode?' checked and 'Content-Type' set to 'text/plain'. The 'Include Equals?' checkbox is also checked. At the bottom of the panel are buttons for 'Detail', 'Add', 'Add from Clipboard', 'Delete', 'Up', and 'Down'.

Ahora en Datos Alumno cambiamos el usuario por \${__urlencode(\${login})} para asegurarnos de que pone lo que ponemos en el formato correcto (por ejemplo cambiando el @ por %40) y pinchamos en Add y añadimos \${login} en value

This screenshot shows the same Apache JMeter interface as the previous one, but with the 'Datos Alumno' sampler selected. The 'HTTP Request' panel now shows a GET request to the path /api/v1/alumnos/alumno/\${__urlencode(\${login})}. In the 'Parameters' section, there is a single entry for 'login' with the value \${login}. The 'URL Encode?' checkbox is checked, and the 'Content-Type' is set to 'text/plain'. The 'Include Equals?' checkbox is also checked. The other tabs ('Body Data', 'Files Upload') are visible at the bottom of the panel.

Vamos a volver a probarlo yendo a View Results Tree y clicando en el triángulo verde.

The screenshot shows the Apache JMeter interface with a test plan named 'Test.jmx'. The left pane displays a tree structure of requests under 'Alumnos': 'Login Alumno', 'Extraer JWT de Login', 'View Results Tree', 'Autorizacion Alumnos', 'Datos Alumno', 'JWT Token', and 'Lista Alumnos'. The 'Administradores' node is also visible. The right pane shows the 'Sampler result' tab for the selected 'Login Alumno' request. The results table lists several samples, with the last one being the selected 'Login Alumno' sample. The details panel shows the following information for this sample:

- Thread Name: Alumnos-1-1
- Sample Start: 2020-06-04 14:49:02 CEST
- Load time: 14
- Connect Time: 2
- Latency: 14
- Size in bytes: 582
- Sent bytes: 346
- Headers size in bytes: 399
- Body size in bytes: 183
- Sample Count: 1
- Error Count: 0
- Data type ("text/html"): text
- Response code: 200
- Response message: OK

Below the details, it says 'HTTPSampleResult fields:' with 'Content-Type: text/html; charset=utf-8' and 'DataEncoding: utf-8'.

Vemos que sigue funcionando así que podemos continuar.

Ahora vamos a añadir un tiempo de espera Random para simular que el usuario está haciendo cosas. Para ello vamos a Alumnos/add/timer/"Gaussian Random Timer" y ponemos una desviación de 500 y un Delay de 3000.

Ahora vamos a añadir un par de auditores de resultados, le damos a

Alumnos/add/listener/Summary Report

Y

Alumnos/add/listener/Aggregate Report

The screenshot shows the Apache JMeter interface with the same test plan 'Test.jmx'. The left pane shows the tree structure with 'Espera Random' selected. The right pane displays the 'Gaussian Random Timer' configuration dialog. The 'Name:' field is set to 'Espera Random'. Under 'Thread Delay Properties', the 'Deviation (in milliseconds):' field is set to 500 and the 'Constant Delay Offset (in milliseconds):' field is set to 3000.

Hacemos una prueba mas y desactivamos el View Results Tree ya que es un test pesado.

Test.jmx (C:\Users\carlo\Desktop\Test.jmx) - Apache JMeter (5.3)

File Edit Search Run Options Tools Help

ETSII Alumnos API

- Alumnos
 - Login Alumno
 - Extrar JWT de Login
 - View Results Tree
 - Autorizacion Alumnos
 - Datos Alumno
 - JWT Token
 - Lista Alumnos
 - Espera Random
 - Summary Report
 - Aggregate Report
- Administradores

Summary Report

Name: Summary Report

Comments:

Write results to file / Read from file

Filename: Browse... Log/Display Only: Errors Successes

Label	# Sampl...	Average	Min	Max	Std. Dev.	Error %	Through...	Receive...	Sent KB/...	Avg. Bytes
Login Al...	1	24	24	24	0.00	0.00%	41.7/sec	23.68	14.08	582.0
Datos A...	1	33	33	33	0.00	0.00%	30.3/sec	31.01	12.31	1048.0
TOTAL	2	28	24	33	4.50	0.00%	36.1/min	0.48	0.22	815.0

Include group name in label? Save Table Header

Test.jmx (C:\Users\carlo\Desktop\Test.jmx) - Apache JMeter (5.3)

File Edit Search Run Options Tools Help

ETSII Alumnos API

- Alumnos
 - Login Alumno
 - Extrar JWT de Login
 - View Results Tree
 - Autorizacion Alumnos
 - Datos Alumno
 - JWT Token
 - Lista Alumnos
 - Espera Random
 - Summary Report
 - Aggregate Report
- Administradores

Aggregate Report

Name: Aggregate Report

Comments:

Write results to file / Read from file

Filename: Browse... Log/Display Only: Errors Successes

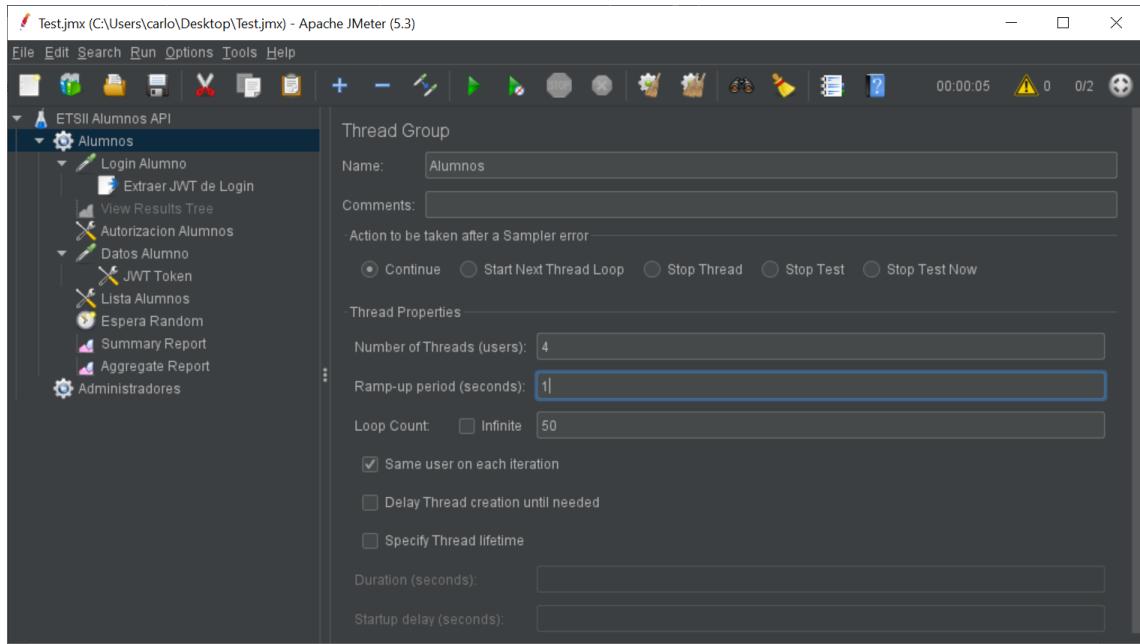
Label	# Sampl...	Average	Median	90% L...	95% L...	99% L...	Min	Maxim...	Error %	Throu...	Receiv...	Sent K...
Login ...	1	24	24	24	24	24	24	24	0.00%	41.7/s...	23.68	14.08
Datos...	1	33	33	33	33	33	33	33	0.00%	30.3/s...	31.01	12.31
TOTAL	2	28	24	33	33	33	24	33	0.00%	36.1/...	0.48	0.22

Include group name in label? Save Table Header

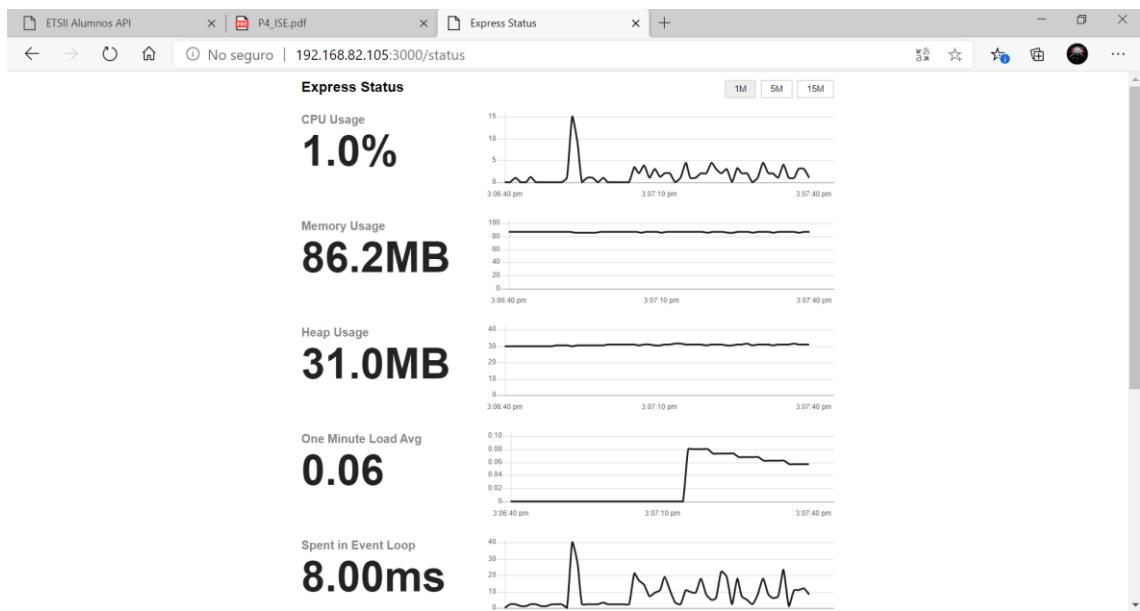
Vemos que va funcionando todo.

Ahora vamos a hacer una prueba mas “real” para ver que todo funciona.

Nos vamos a Alumnos y lo configuraremos así

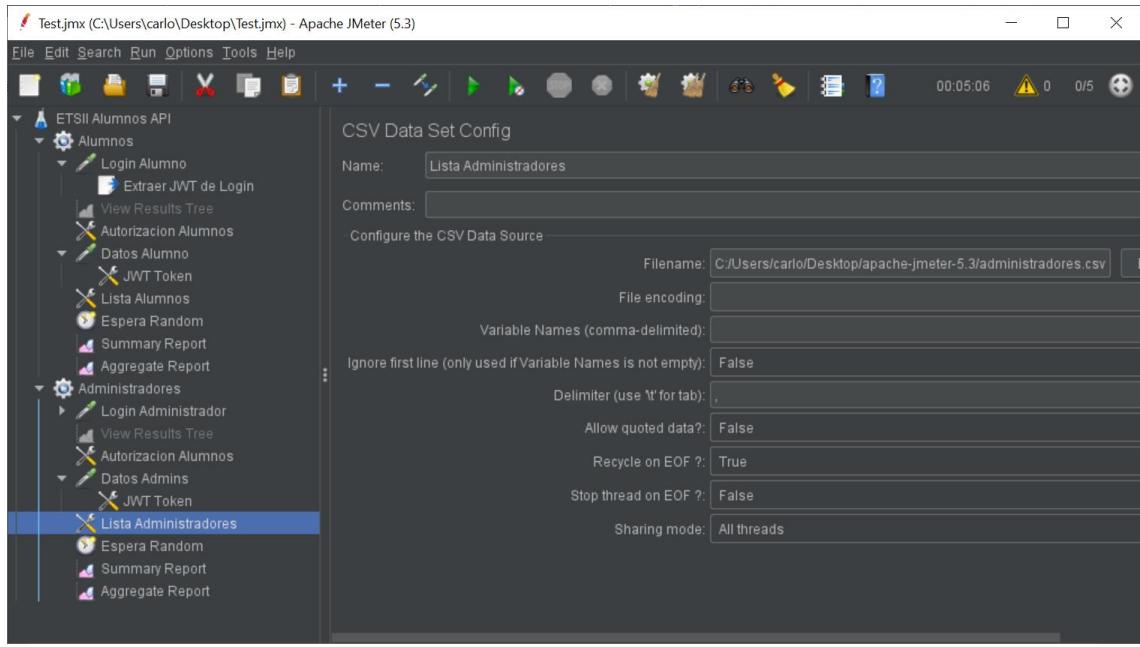


Ahora nos vamos a la nuestra pagina del servidor en nuestro caso 192.168.82.105:3000/status para ver las gráficas y ejecutamos el test en el JMeter.

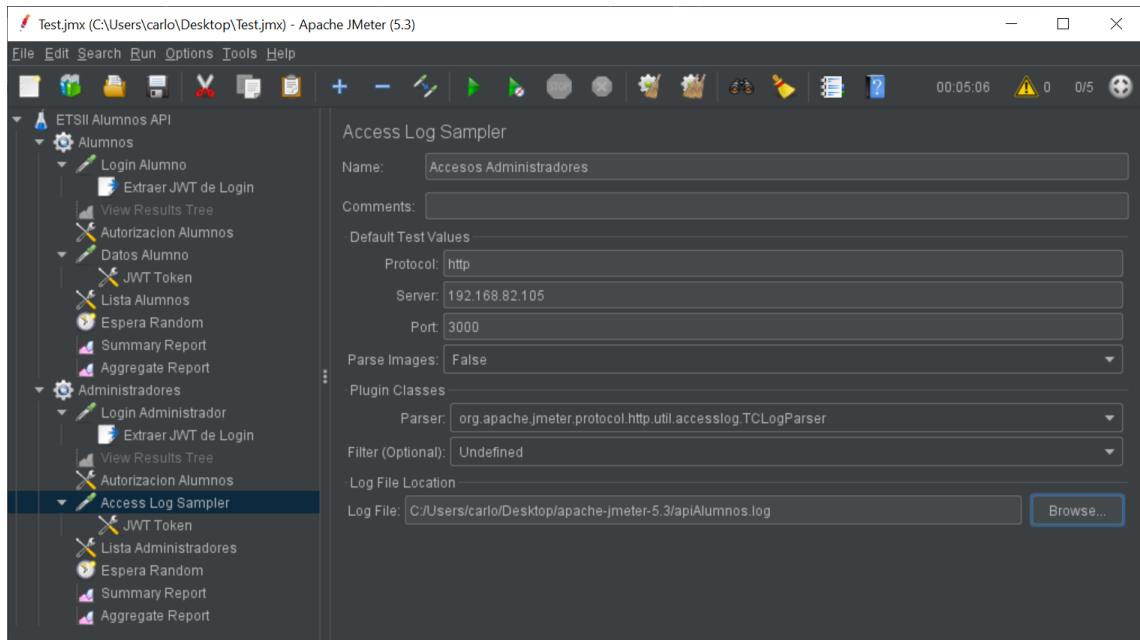


Ya tenemos totalmente configurados los alumnos así que ahora vamos con los admins. Para ahorrar tiempo vamos a copiar lo que hemos hecho en alumnos y a modificar los datos.

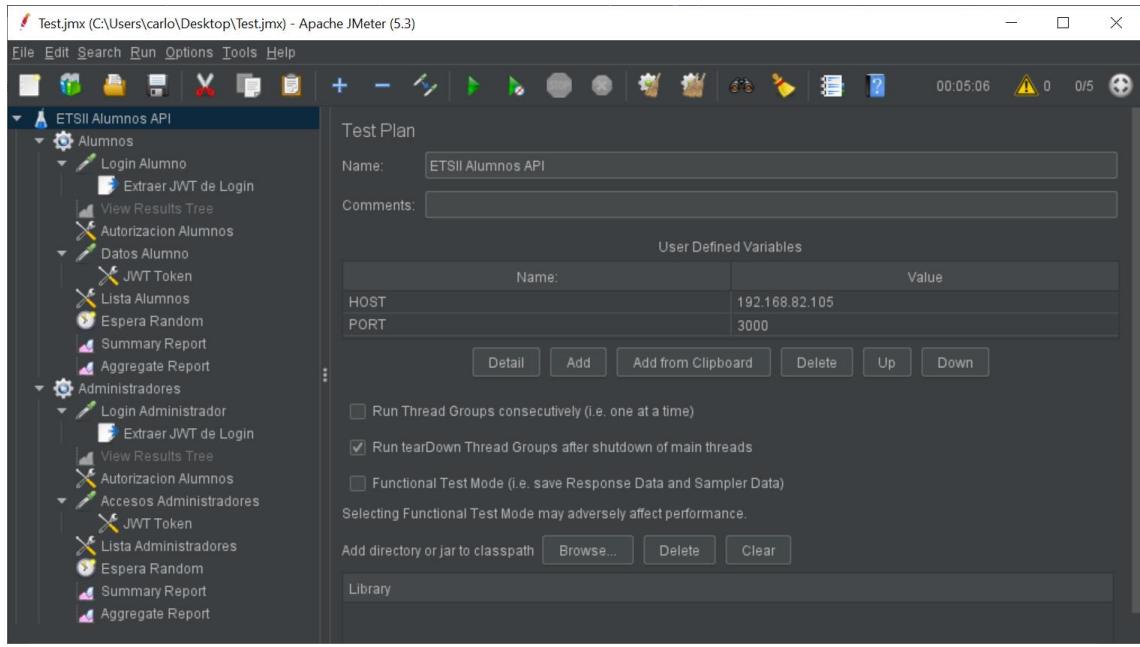
Vamos a cambiar el archivo de alumnos.csv por administradores.csv



Y añadimos Aministradores/Add/"Sampler Access Log" donde añadimos el archivo .log del repositorio, arastramos el JWT Token desde Datos Alumnos borramos Datos Alumno y dejamos el Access Log Sampler que acabamos de añadir tal que así.

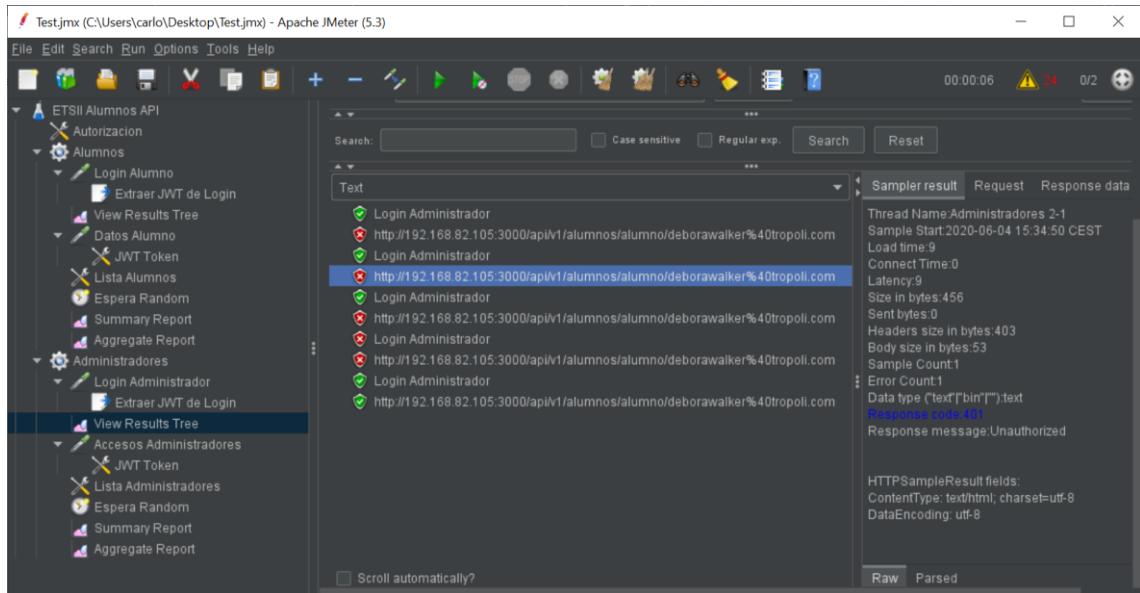


Antes de seguir vamos a parametrizarlo todo para ello nos vamos a ETSII Alumnos Api y definimos las variables HOST y PORT dejándolo así.

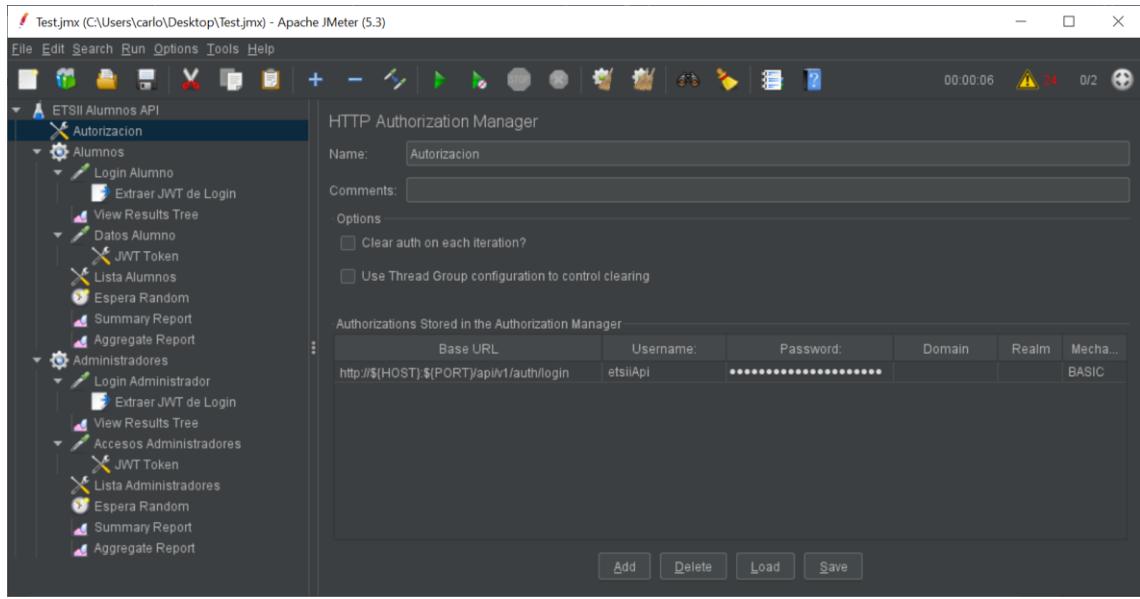


Y ahora vamos a cambiar en todos sitios donde hayamos puesto nuestra ip por \${HOST} y el puerto por \${PORT}

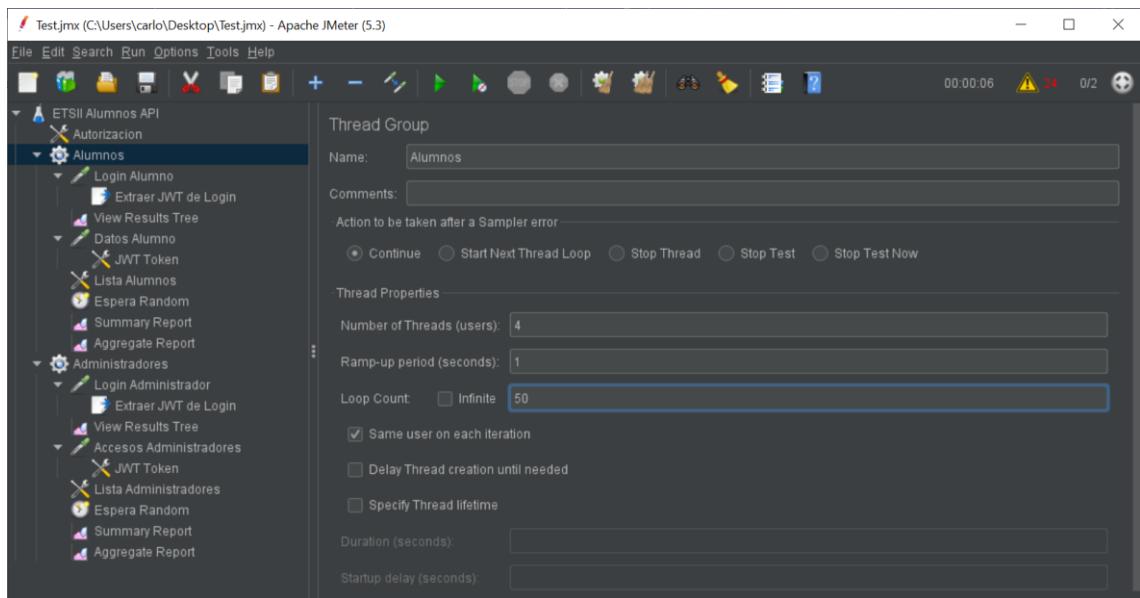
Ahora sigamos por donde íbamos, vamos a hacer una prueba simple, activamos el view Results Tree de administradores, tanto en Alumnos como en Administradores ponemos todo a 1 y ejecutamos.



Tuvimos un error que se soluciona añadiendo /api/v1/auth/login en Autorizacion (Autorizacion es autorización alumnos sacado de Alumnos y Administradores ya que es lo mismo en los 2)



Ahora vamos a pasar a hacer la prueba grande, para ello desactivamos los View Result Tree de ambos lados y cambiamos los valores tanto de Alumnos como de Administradores a 4, 1 50 y ejecutamos.



Test.jmx (C:\Users\carlo\Desktop\Test.jmx) - Apache JMeter (5.3)

File Edit Search Run Options Tools Help

Thread Group

Name: Administradores

Comments:

Action to be taken after a Sampler error:

- Continue
- Start Next Thread Loop
- Stop Thread
- Stop Test
- Stop Test Now

Thread Properties

Number of Threads (users): 4

Ramp-up period (seconds): 1

Loop Count: Infinite 50

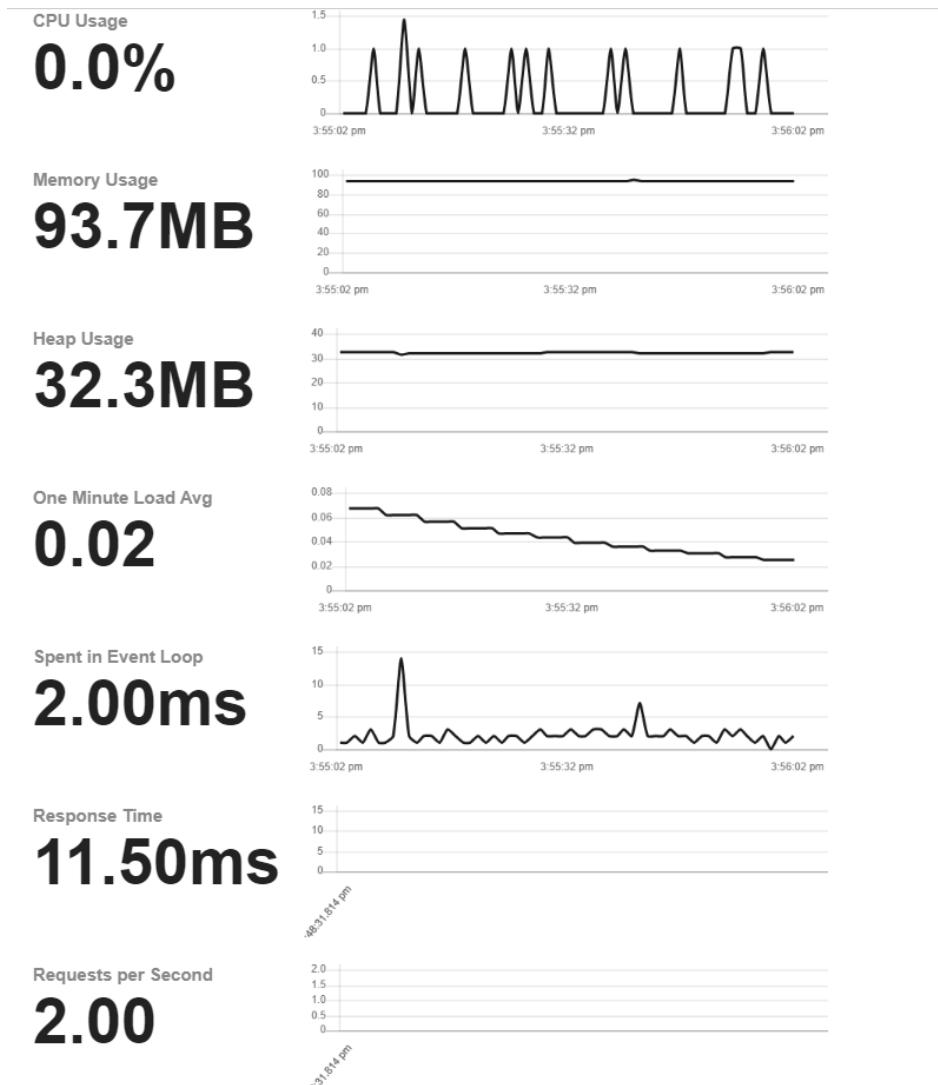
Same user on each iteration

Delay Thread creation until needed

Specify Thread lifetime

Duration (seconds):

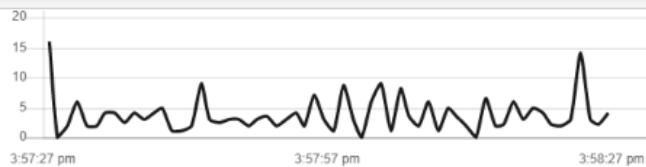
Startup delay (seconds):



Estas son las graficas del servidor en reposo

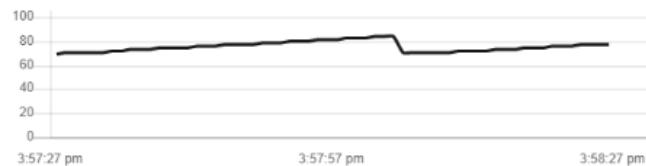
CPU Usage

4.0%



Memory Usage

77.3MB



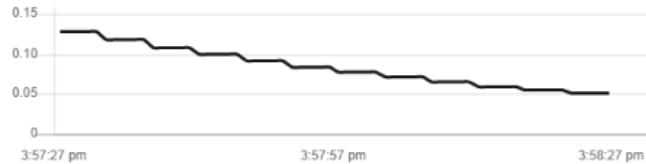
Heap Usage

26.7MB



One Minute Load Avg

0.05



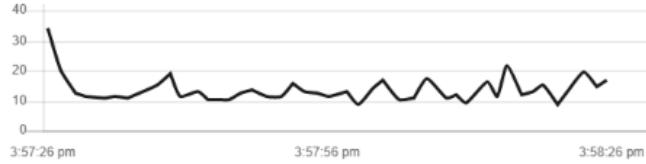
Spent in Event Loop

25.00ms



Response Time

16.73ms



Requests per Second

2.98



Estas son las gráficas ejecutándose el test.

Una vez hecho esto vamos a crear un archivo de resultados desde el terminal, Para ello nos vamos a la carpeta donde tenemos JMeter desde el terminal usamos el comando

`jmeter -n -t test.jmx -l results.jtl -JHOST=192.168.82.105 -JPORT=3000` para que nos cree un archivo results.jtl en la carpeta bin del jmeter, test.jmx es como se llama mi archivo de jmeter.

```
Selezionar Símbolo del sistema - jmeter -n -t test.jmx -l results.jtl -JHOST=192.168.82.105 -JPORT=3000
Microsoft Windows [Versión 10.0.18363.836]
(c) 2019 Microsoft Corporation. Todos los derechos reservados.

C:\Users\carlo>cd Desktop

C:\Users\carlo\Desktop>cd apache-jmeter-5.3

C:\Users\carlo\Desktop\apache-jmeter-5.3>cd bin

C:\Users\carlo\Desktop\apache-jmeter-5.3\bin>jmeter -n -t test.jmx -l results.jtl -JHOST=192.168.82.105 -JPORT=3000
Creating summariser <summary>
Created the tree successfully using test.jmx
Starting standalone test @ Thu Jun 04 16:19:20 CEST 2020 (1591280360027)
Waiting for possible Shutdown/StopTestNow/HeapDump/ThreadDump message on port 4445
summary +    19 in 00:00:09 =    2.0/s Avg:    26 Min:    12 Max:    91 Err:    0 (0.00%) Active: 8 Started: 8 Finished: 0
summary +    78 in 00:00:30 =   2.6/s Avg:    18 Min:     9 Max:    49 Err:    0 (0.00%) Active: 8 Started: 8 Finished: 0
summary =   97 in 00:00:39 =   2.5/s Avg:    20 Min:     9 Max:    91 Err:    0 (0.00%)
summary +    81 in 00:00:30 =   2.7/s Avg:    19 Min:    12 Max:    39 Err:    0 (0.00%) Active: 8 Started: 8 Finished: 0
summary =  178 in 00:01:09 =   2.6/s Avg:    19 Min:     9 Max:    91 Err:    0 (0.00%)
```