

Práctica 4: Benchmarking y Ajuste del Sistema

David Gil Bautista

Mail: davidbautista@correo.ugr.es

DNI: 45925324M

Instalación

Para la instalación hemos seguido la documentación oficial de docker y hemos alojado el software en una máquina virtual corriendo Ubuntu Server con 1 GB de RAM y una CPU Intel Core I5 6300HQ con 2 hilos.

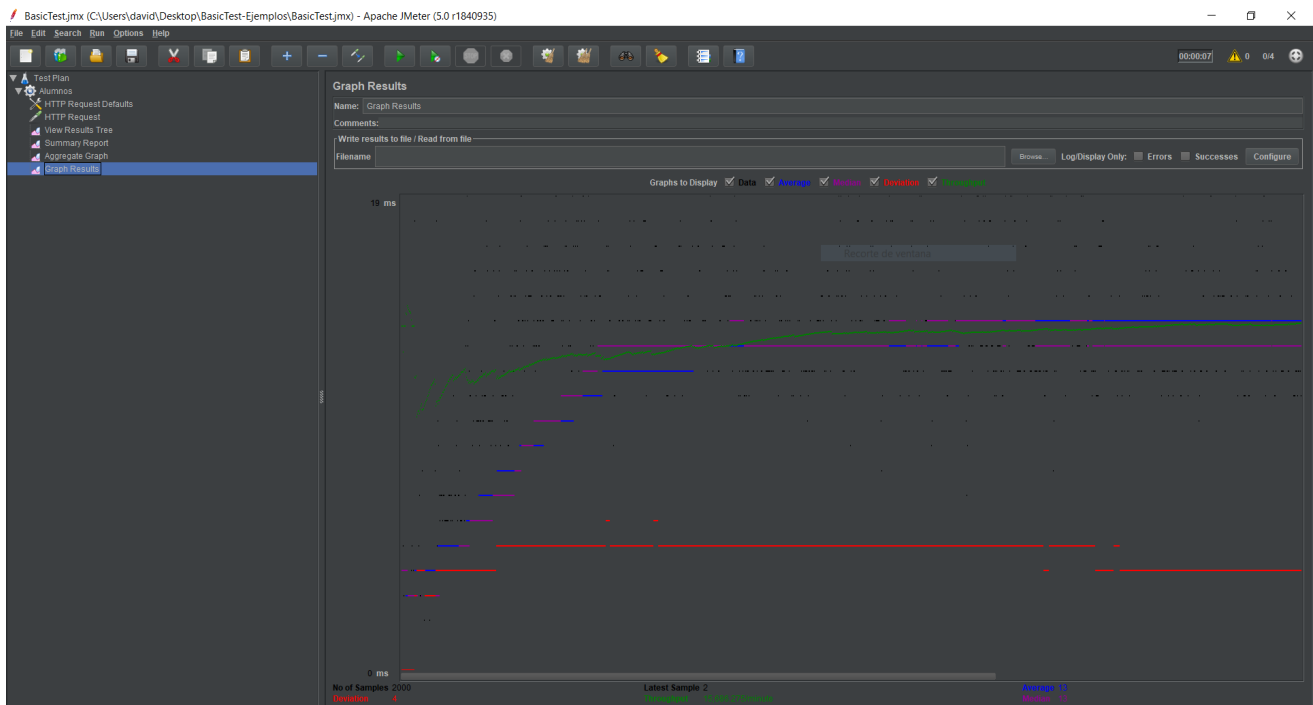
Tras instalar docker descargamos el repositorio de github y ejecutamos *docker-compose up*, con esto construiremos el contenedor. Una vez ya esté construido podremos ejecutarlo sin necesidad de construirlo de nuevo.

Test básico

Para este test usaremos el ejemplo que se nos proporciona en SWAD, tan sólo modificaremos la IP del host para establecer la misma que tenemos configurada en nuestra máquina virtual.

Para este test contamos con los siguientes componentes:

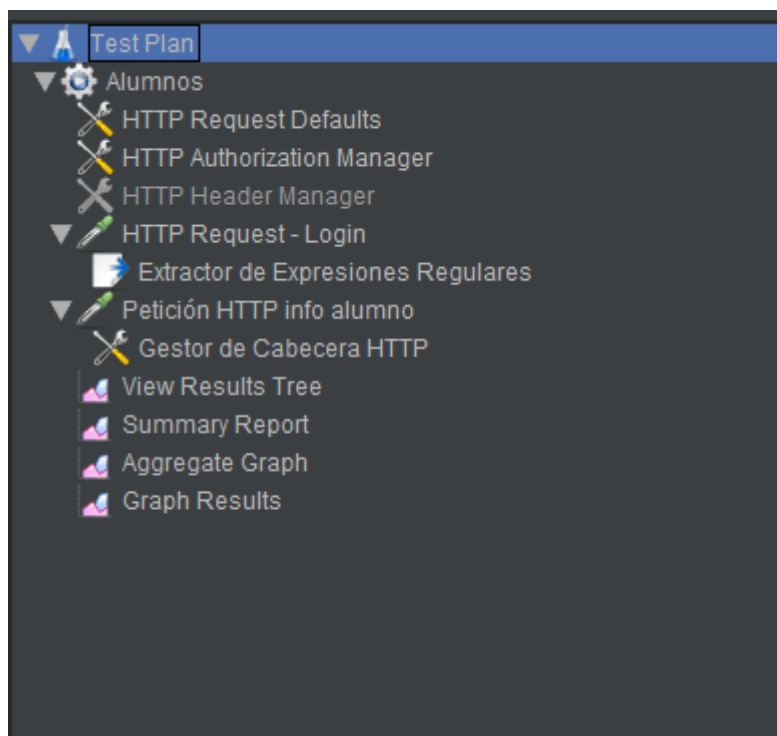
En este test básico tendremos un grupo de 4 hebras que se ejecutarán 500 veces en total. En cada ejecución establecemos los datos de la petición con **HTTP Request Defaults** y en **HTTP Request** pondremos que tendremos una petición de tipo *GET*. Después podemos ver que tenemos 4 formas de medir los resultados y a continuación mostramos la que nos otorga los datos en forma de gráfica:



Como podemos ver con el paso del tiempo aumenta el flujo de datos que maneja nuestro servidor y también la media de tiempo que tarda en procesar las solicitudes.

Test básico Etsii2

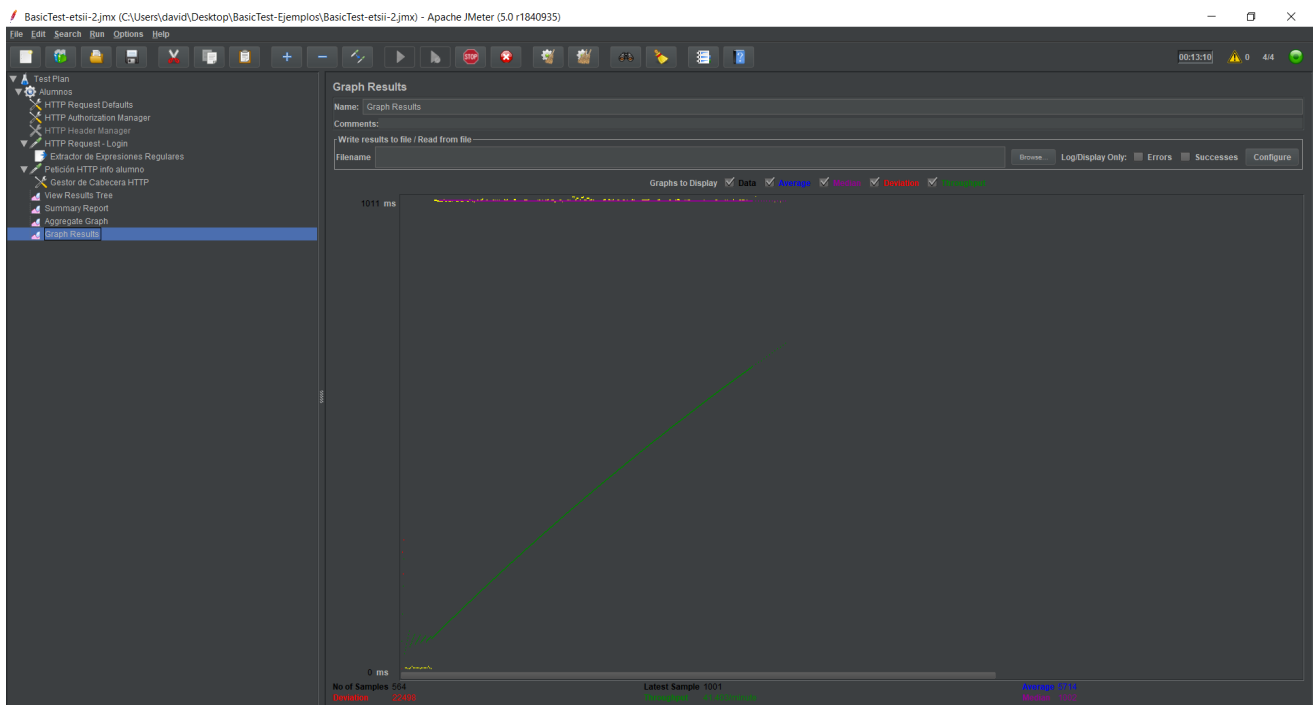
Para este test podemos observar que tenemos los siguientes elementos:



En este ejemplo podemos ver que ahora tenemos un **HTTP Authorization Manager** lo que nos permitirá establecer la conexión con la api del servidor, en el siguiente **HTTP Header Manager** mandamos la autorización. Una vez establecida la conexión mandamos los datos del alumno con haciendo un **POST** con el **HTTP Request - Login**. Tras hacer el Login procedemos a pedir la información del usuario haciendo un **GET**

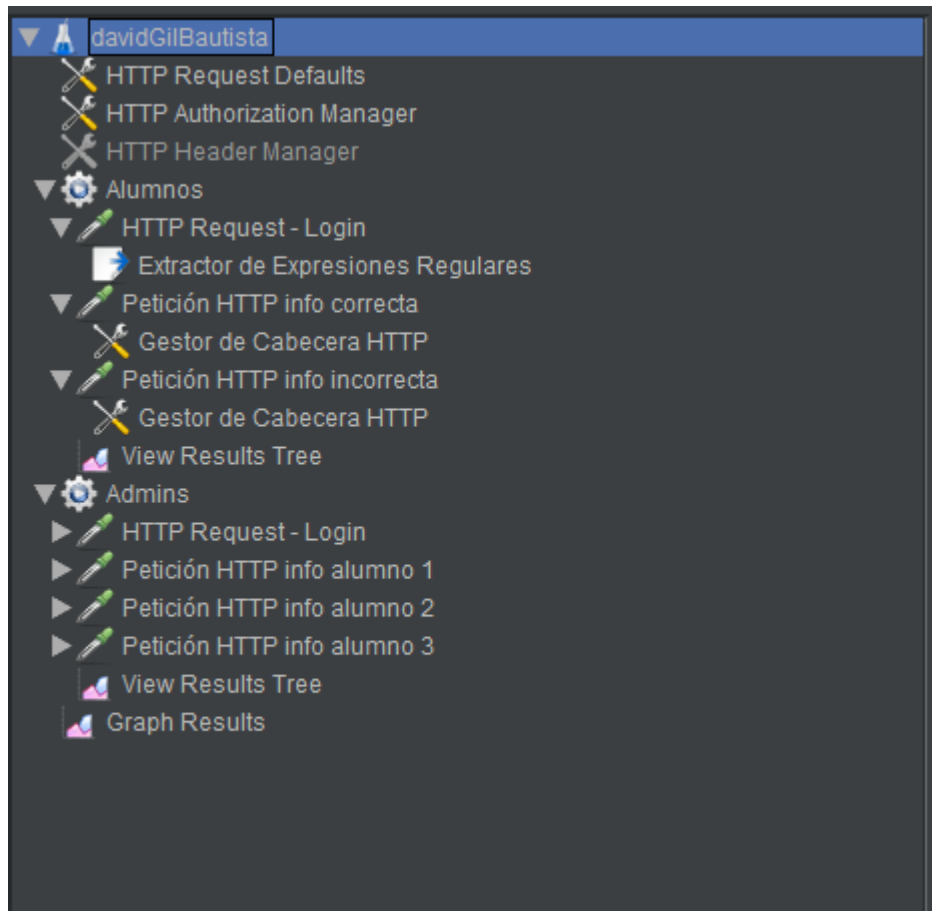
con un **HTTP Request**.

Podemos ver el resultado a continuación:



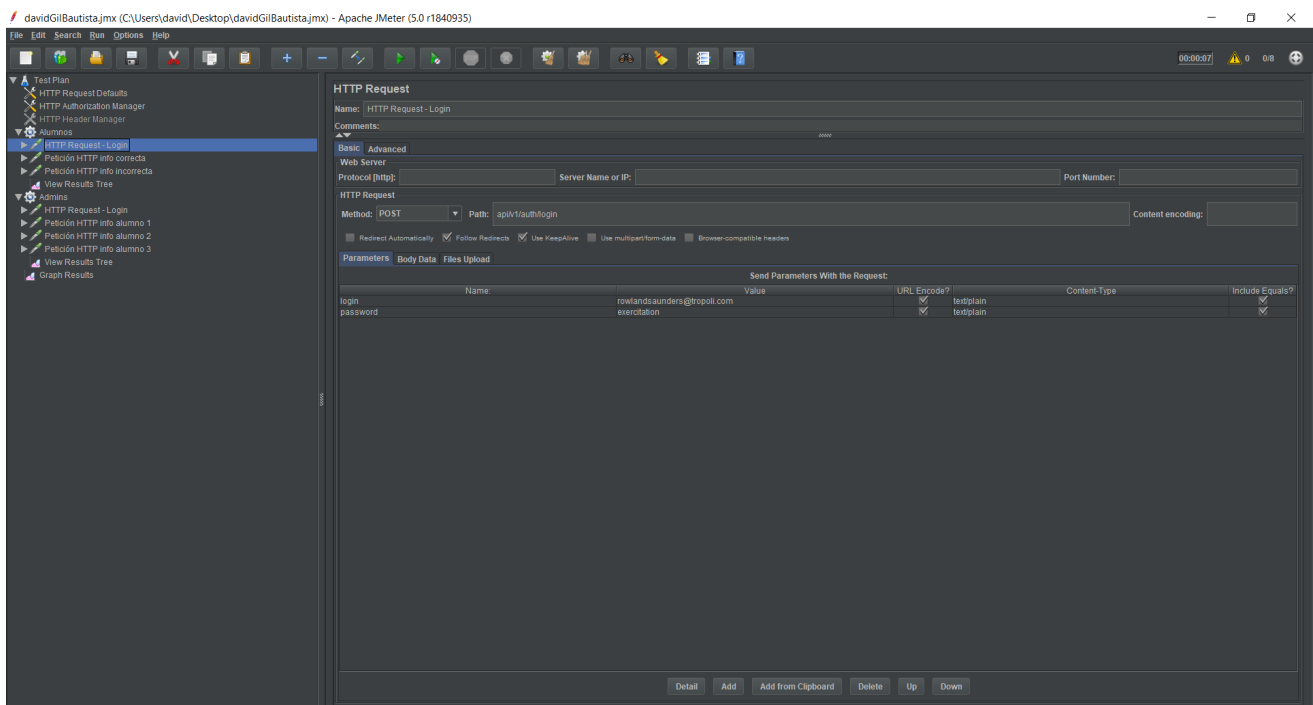
Mi test

Para la realización de mi test personalizado he usado los siguientes componentes:

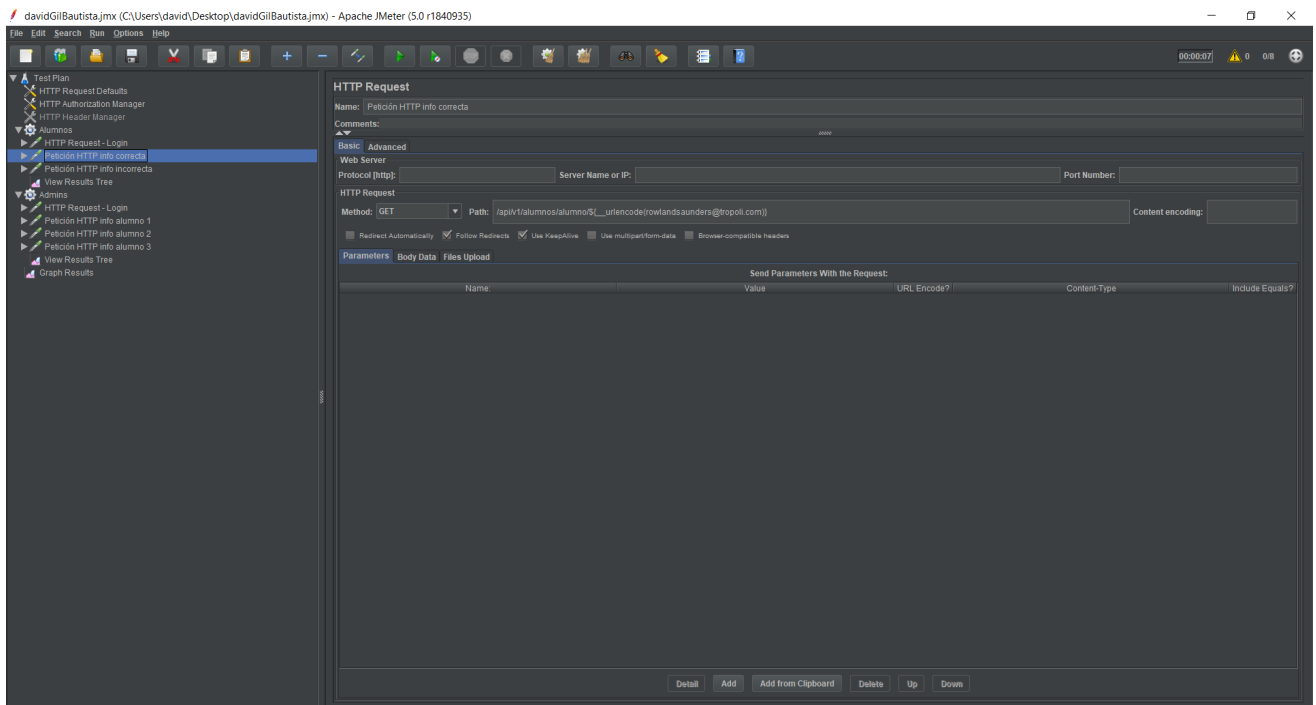


Al igual que los ejemplos anteriores creamos la configuración por defecto para establecer la conexión HTTP y crearemos dos grupos de hebras en los que nos loguearemos y pediremos la información del usuario que se ha logueado y la de otro usuario.

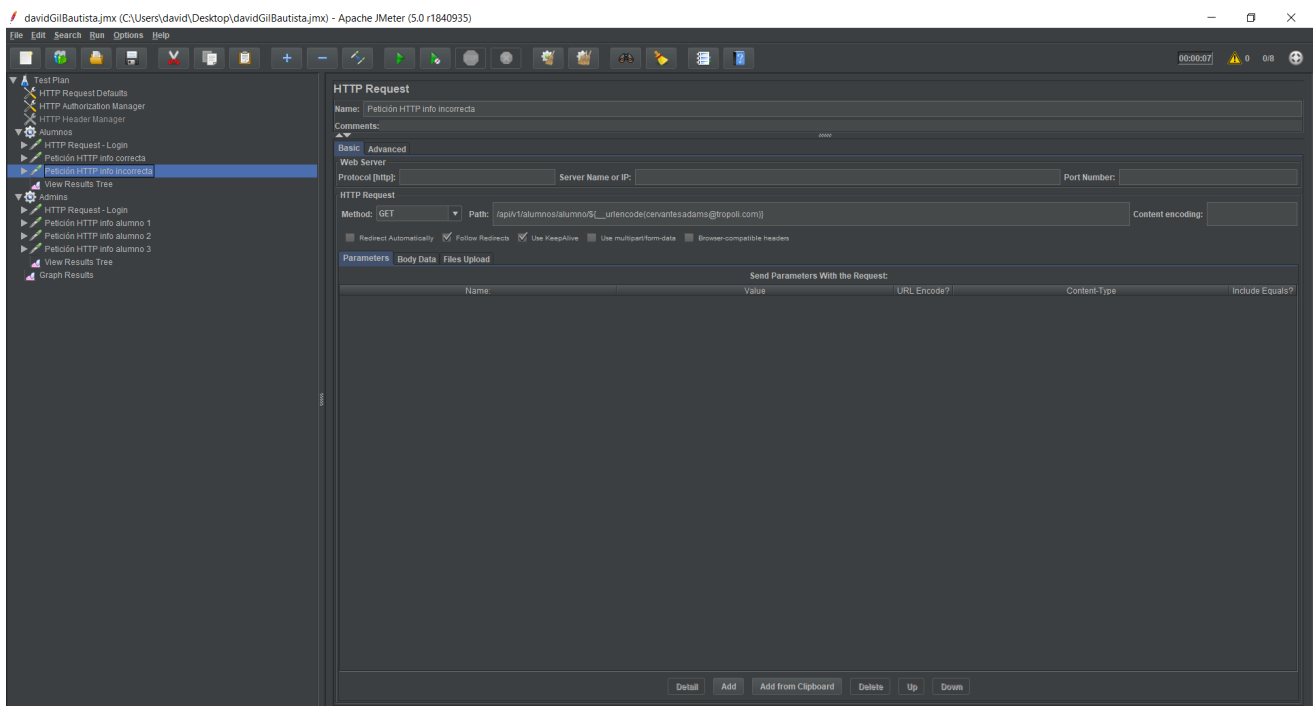
En el primer grupo de hebras nos logueamos como un alumno.



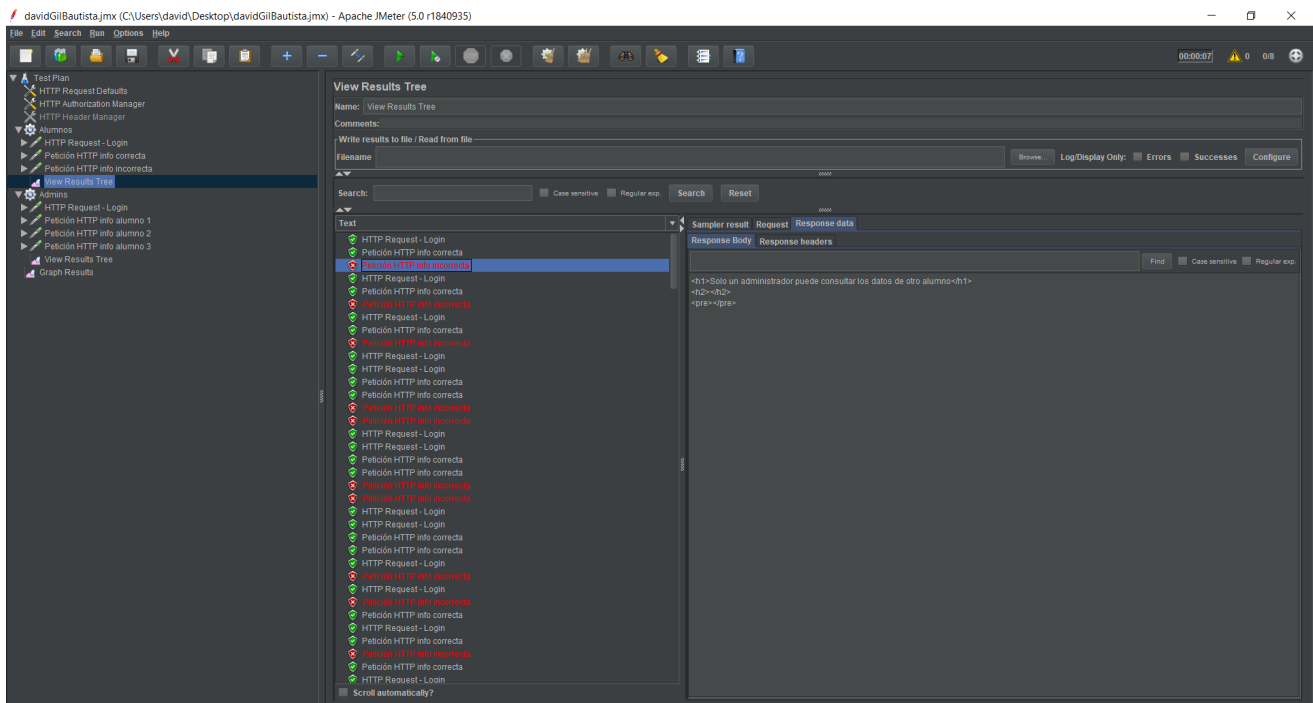
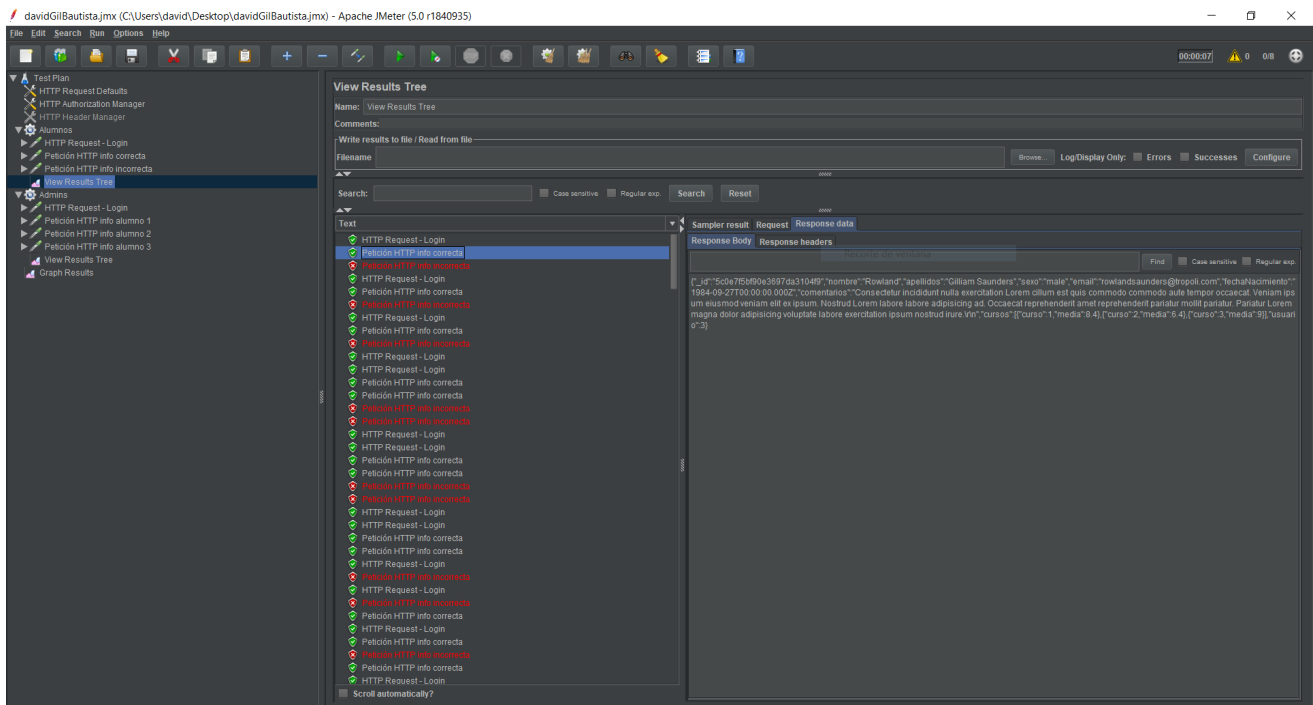
Ahora pediremos la información del alumno que se ha logueado.



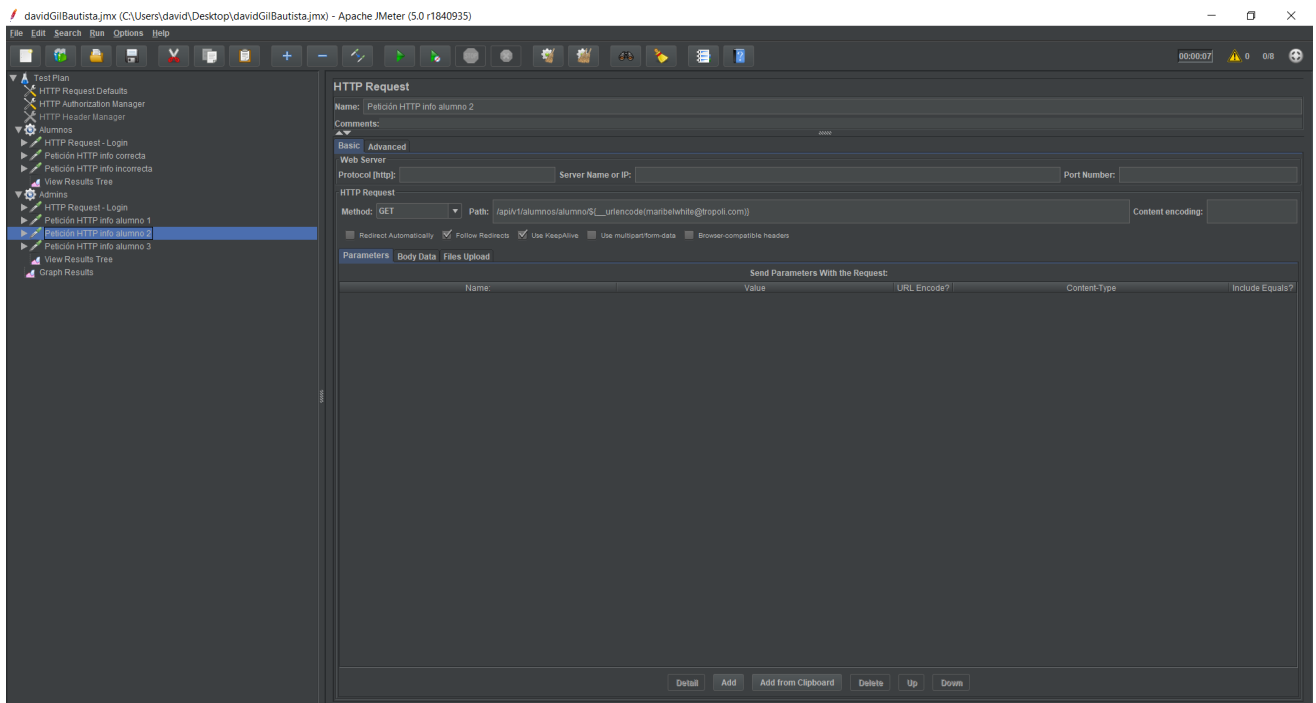
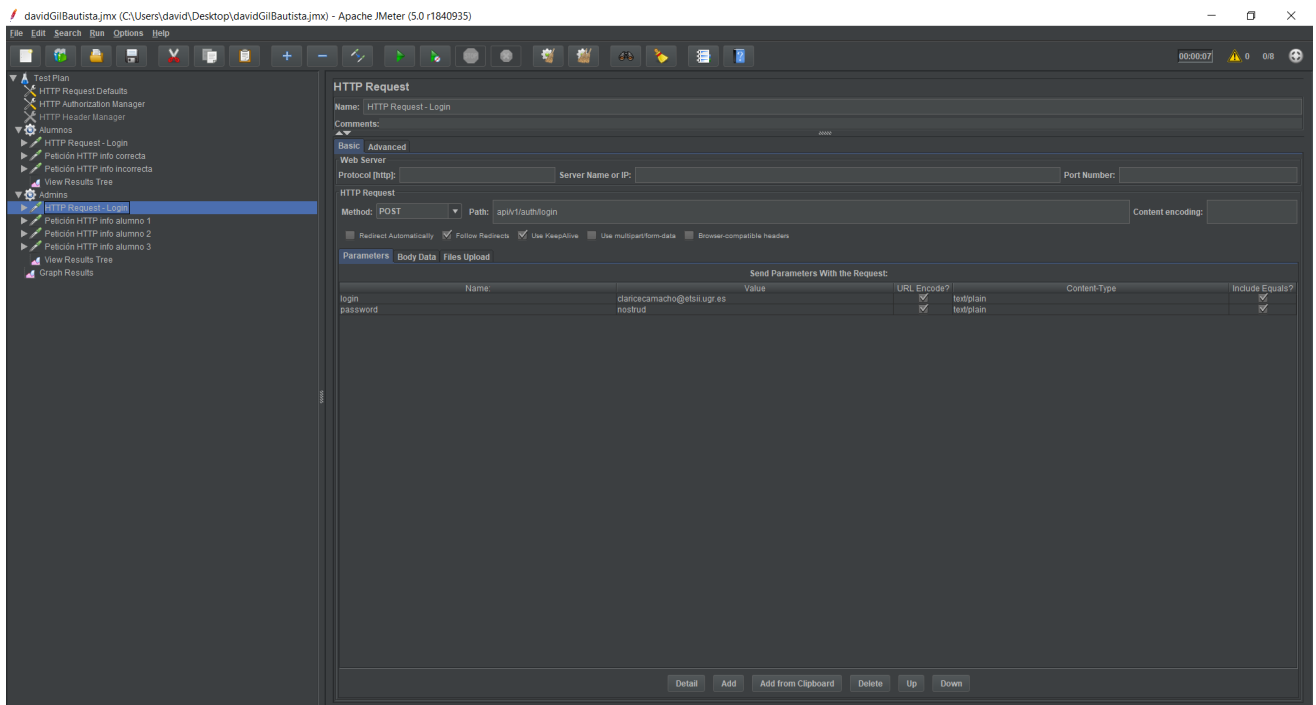
Y la de otro usuario distinto.



En el resultado podemos ver que obtenemos una respuesta favorable al solicitar la información del usuario que se ha logueado pero al pedir la de otro distinto nos dice que solo un administrador puede conseguir la información de otro alumno.



Para el caso de los administradores nos loguearemos como uno y pediremos la información de 3 alumnos distintos.



En este caso podemos ver que no obtenemos ningún fallo y podemos consultar la información.

Test Plan

HTTP Request Defaults

HTTP Authorization Manager

HTTP Header Manager

Alumnos

HTTP Request - Login

Peticón HTTP info correcta

Peticón HTTP info incorrecta

Admins

HTTP Request - Login

Peticón HTTP info alumno 1

Peticón HTTP info alumno 2

Peticón HTTP info alumno 3

View Results Tree

Graph Results

View Results Tree

Name: View Results Tree

Comments:

Write results to file / Read from file

Filename

Search

Reset

Case sensitive

Regular exp

Log/Display Only

Errors

Successes

Configure

Text

Peticón HTTP info alumno 3

HTTP Request - Login

HTTP Request - Login

HTTP Request - Login

Peticón HTTP info alumno 1

Peticón HTTP info alumno 1

Peticón HTTP info alumno 1

Peticón HTTP info alumno 2

Peticón HTTP info alumno 2

Peticón HTTP info alumno 2

Peticón HTTP info alumno 2

Peticón HTTP info alumno 3

Peticón HTTP info alumno 3

Peticón HTTP info alumno 3

HTTP Request - Login

HTTP Request - Login

HTTP Request - Login

Peticón HTTP info alumno 1

Peticón HTTP info alumno 1

Peticón HTTP info alumno 1

Peticón HTTP info alumno 2

Peticón HTTP info alumno 2

Peticón HTTP info alumno 2

Peticón HTTP info alumno 2

Peticón HTTP info alumno 3

Peticón HTTP info alumno 3

Peticón HTTP info alumno 3

Peticón HTTP info alumno 3

HTTP Request - Login

HTTP Request - Login

HTTP Request - Login

Sampler result

Request

Response data

Thread Name: Admins 2-4

Sample Start: 2019-12-18 22:56:18 CET

Load time: 22

Connect Time: 0

Latency: 22

Size in bytes: 1570

Sent bytes: 387

Headers size in bytes: 408

Body size in bytes: 1162

Sample Count: 1

Error Count: 0

Data type ("text/bin"): text

Response code: 200

Response message: OK

HTTPSampleResult fields:

ContentType: application/json; charset=utf-8

DataEncoding: utf-8

Raw

Parsed