



# UNIVERSIDAD DE GRANADA

3ºA - A2

GRADO EN INGENIERÍA INFORMÁTICA

---

## Ingeniería de Servidores: Práctica 4.

---

*Autor:*

Antonio David Villegas Yeguas

*Asignatura:*

*Ingeniería de Servidores*

15 de diciembre de 2019

<https://github.com/advy99/ISE/>

Esta obra está bajo una licencia Creative Commons “Reconocimiento-NoCommercial-CompartirIgual 4.0 Internacional”.



# Índice

<b>1. Phoronix Test Suite</b>	<b>2</b>
1.1. Instalación de Phoronix Test Suite . . . . .	2
1.1.1. Instalación en el anfitrión (ArchLinux) . . . . .	2
1.1.2. Instalación en Ubuntu Server . . . . .	2
1.2. Instalación en CentOS . . . . .	3
1.3. Instalación con Docker . . . . .	4
1.4. Ejecución de pruebas . . . . .	4
1.5. Configuración de Phoromatic . . . . .	4
1.6. Uso de Phoromatic . . . . .	5
1.6.1. Añadir y conectar equipos a Phoromatic . . . . .	6
1.6.2. Crear Tests Suites . . . . .	7
1.6.3. Ejecutar Suites . . . . .	8
1.6.4. Mostrar resultados . . . . .	10
<b>2. Apache AB</b>	<b>12</b>
2.1. Instalación de Apache AB . . . . .	12
2.2. Uso de AB . . . . .	12
2.3. Interpretar la salida de AB . . . . .	14
2.4. Comparación CentOS y Ubuntu Server . . . . .	16
<b>3. JMeter</b>	<b>17</b>
3.1. Instalación de JMeter . . . . .	17
3.2. Instalando el microservicio iseP4JMeter . . . . .	17
3.2.1. Instalación de Docker y Docker-compose en Ubuntu Server . . . . .	17
3.2.2. Instalación de Docker y Docker-compose en CentOS . . . . .	18
3.2.3. Instalación y ejecución de iseP4JMeter . . . . .	18
3.3. Prueba básica de conexión: Primer contacto con JMeter . . . . .	19
3.4. Configurando un test de carga para iseP4JMeter . . . . .	21
3.4.1. Configuración general . . . . .	21
3.4.2. Configuración para administradores . . . . .	22
3.4.3. Configuración para estudiantes . . . . .	23
3.4.4. Resultados . . . . .	24
<b>Bibliografía</b>	<b>25</b>

## 1. Phoronix Test Suite

Phoronix Test Suite<sup>[1]</sup> es un software que nos permite ejecutar un conjunto de benchmarks, ya sean benchmarks propios o usando la plataforma OpenBenchmarking<sup>[2]</sup>.

En esta práctica vamos a ejecutar distintos benchmarks sobre el anfitrión, la máquina virtual de Ubuntu Server y la máquina virtual de CentOS.

Antes de realizar los test, debido a la configuración de los distintos sistemas sabemos que el que mejor rendimiento obtendrá será el anfitrión, ya que este puede acceder a la totalidad de los recursos (en mi caso 8 núcleos de un Intel i5-8250U y 8 GB de memoria RAM DDR4) mientras que las máquinas virtuales solo a parte de esta (1 núcleo del Intel i5-8250U y 1 GB de RAM DDR4).

### 1.1. Instalación de Phoronix Test Suite

Para la instalación de Phoronix Test Suite el guión de prácticas nos recomendaba instalarlo desde el gestor de paquetes de los distintos sistemas operativos usados, sin embargo, debido a que las versiones están bastante desactualizadas y para usar Phoromatic necesitaremos una versión superior a la 9.0.0, lo instalará de forma manual.

#### 1.1.1. Instalación en el anfitrión (ArchLinux)

En mi caso, el anfitrión tiene instalado como sistema operativo ArchLinux. A pesar de que la instalación de forma manual es muy sencilla, este sistema operativo cuenta con un script creado por la comunidad para instalarlo. Nos bastara con instalar el paquete comunitario phoronix-test-suite desde el AUR.

En mi caso, al usar YAY<sup>[3]</sup> como AUR Helper basta con la siguiente orden en bash:

```
1 \$ yay -S phoronix-test-suite
```

#### 1.1.2. Instalación en Ubuntu Server

Para instalar Phoronix Test Suite basta con descargar el paquete disponible para cualquier distribución en la página de descargas de Phoronix<sup>[4]</sup>, extraer el paquete y ejecutar el script de instalación:

```
1 \$ wget  
→ https://phoronix-test-suite.com/releases/phoronix-test-suite-9.2.0.tar.gz  
2 \$ tar -xvf phoronix-test-suite-9.2.0.tar.gz  
3 \$ cd phoronix-test-suite/  
4 \$ sudo ./install-sh
```

Además tendremos que instalar algunos paquetes de PHP:

```
1 \$ sudo apt install php-xml php-gd
```

Y con esto tendremos instalado Phoronix Test Suite en nuestro Ubuntu Server.

```
Ubuntu-ISE (P4 - descargado iseP4JMeter - falta configurar y probar con jmeter) [Corriendo] - Oracle VM ... - ×
Archivo Máquina Ver Entrada Dispositivos Ayuda
antoniovy@ubuntu:"$ wget https://phoronix-test-suite.com/releases/phoronix-test-suite-9.2.0.tar.gz
--2019-12-09 15:55:13-- https://phoronix-test-suite.com/releases/phoronix-test-suite-9.2.0.tar.gz
Resolviendo phoronix-test-suite.com (phoronix-test-suite.com)... 162.252.83.89
Conectando con phoronix-test-suite.com (phoronix-test-suite.com)|162.252.83.89|:443... conectado.
Petición HTTP enviada, esperando respuesta... 200 OK
Longitud: 1368776 (1,3M) [application/x-gzip]
Grabando a: "phoronix-test-suite-9.2.0.tar.gz"

phoronix-test-suite-9.2. 100%[=====] 1,30M 1,65MB/s   in 0,8s
2019-12-09 15:55:14 (1,65 MB/s) - "phoronix-test-suite-9.2.0.tar.gz" guardado [1368776/1368776]

antoniovy@ubuntu:"$ ls
git phoronix-test-suite-9.2.0.tar.gz
antoniovy@ubuntu:"$
```

```
Ubuntu-ISE (P4 - descargado iseP4JMeter - falta configurar y probar con jmeter) [Corriendo] - Oracle VM ... - ×
Archivo Máquina Ver Entrada Dispositivos Ayuda
phoronix-test-suite/documentation/stubs/59_component_testing.html
phoronix-test-suite/documentation/stubs/68_offline_additions.html
phoronix-test-suite/documentation/man-pages/
phoronix-test-suite/documentation/man-pages/phoronix-test-suite.1
phoronix-test-suite/documentation/phoromatic.md
phoronix-test-suite/documentation/phoronix-test-suite.md
phoronix-test-suite/documentation/includes/
phoronix-test-suite/documentation/includes/pts-documentation.png
phoronix-test-suite/documentation/includes/pts-documentation.css
phoronix-test-suite/documentation/phoromatic.html
phoronix-test-suite/documentation/phoronix-test-suite-windows.md
phoronix-test-suite/documentation/TODO
phoronix-test-suite/documentation/phoronix-test-suite-windows.html
phoronix-test-suite/install-sh
phoronix-test-suite/AUTHORS
phoronix-test-suite/phoronix-test-suite.bat
phoronix-test-suite/COPYING
phoronix-test-suite/README.md
phoronix-test-suite/ChangeLog
phoronix-test-suite/release-highlights.md
antoniovy@ubuntu:"$ ls
git phoronix-test-suite phoronix-test-suite-9.2.0.tar.gz
antoniovy@ubuntu:"$ ls phoronix-test-suite
AUTHORS deploy install-sh phoronix-test-suite.bat release-highlights.md
ChangeLog documentation ob-cache pts-core
COPYING install.bat phoronix-test-suite README.md
antoniovy@ubuntu:"$ cd phoronix-test-suite/
antoniovy@ubuntu:"$ ./phoronix-test-suite$ sudo ./install-sh
[sudo] password for antoniovy:
-e
Phoronix Test Suite Installation Completed
Executable File: /usr/bin/phoronix-test-suite
Documentation: /usr/share/doc/phoronix-test-suite/
Phoronix Test Suite Files: /usr/share/phoronix-test-suite/
antoniovy@ubuntu:"$ ./phoronix-test-suite$
```

## 1.2. Instalación en CentOS

Al igual que con Ubuntu, descargamos el paquete y ejecutamos el script de instalación:

- 1 `\$ wget`  
→ `https://phoronix-test-suite.com/releases/phoronix-test-suite-9.2.0.tar.gz`
- 2 `\$ tar -xvf phoronix-test-suite-9.2.0.tar.gz`
- 3 `\$ cd phoronix-test-suite/`
- 4 `\$ sudo ./install-sh`

### 1.3. Instalación con Docker

Otra opción es instalar Phoronix Test Suite a través de Docker[5]

```
1 \$ docker pull phoronix/pts
```

Y ejecutarlo con:

```
1 \$ docker run -it phoronix/pts
```

### 1.4. Ejecución de pruebas

Una vez instalado, podremos realizar pruebas con la siguiente orden:

```
1 \$ phoronix-test-suite benchmark <nombre_prueba>
```

Algunas de las pruebas que he realizado y veremos más adelante son apache (benchmark que utiliza AB), php y smallpt (pequeña prueba para el procesador)

### 1.5. Configuración de Phoromatic

En mi caso ejecutaré el servidor de Phoromatic desde mi anfitrión con la ayuda del manual de phoronix-test-suite[6].

Para configurar Phoromatic debemos incluir algunas extensiones de PHP, editando el archivo /etc/php/php.ini

```
1 \$ sudo vim /etc/php/php.ini
```

Y descomentamos las extensiones sockets, sqlite3 y zip.

Con esto podemos pasar a ejecutar el servidor de Phoromatic con la siguiente orden:

```
1 \$ phoronix-test-suite start-phoromatic-server
```

```
phoronix-test-suite start-phoromatic-server
◆ phoronix-test-suite start-phoromatic-server

Updated OpenBenchmarking.org Repository Index
pts: 366 Distinct Tests, 1348 Test Versions, 53 Suites
New Test Available: pts/deepspeech
Updated OpenBenchmarking.org Repository Index
system: 26 Distinct Tests, 75 Test Versions, 1 Suite

Updated OpenBenchmarking.org Repository Index
git: 6 Distinct Tests, 6 Test Versions, 0 Suites

Phoronix Test Suite v9.0.1
An outdated version of the Phoronix Test Suite is installed.
The version in use is 9.0.1 (9010), but the latest is pts-core 9200.
Visit https://www.phoronix-test-suite.com/ to update this software.

Port 8193 chosen as random port for this instance. Change the default port via the Phoronix Test Suite user configuration file.
Phoronix Test Suite v9.0.1 (Asker) starting Phoromatic Server
Phoronix Test Suite User-Data Directory Path: /home/antonio/.phoronix-test-suite/
Phoronix Test Suite Configuration File: /home/antonio/.phoronix-test-suite/user-config.xml
Phoromatic Server Log File: /home/antonio/.phoronix-test-suite/phoromatic.log

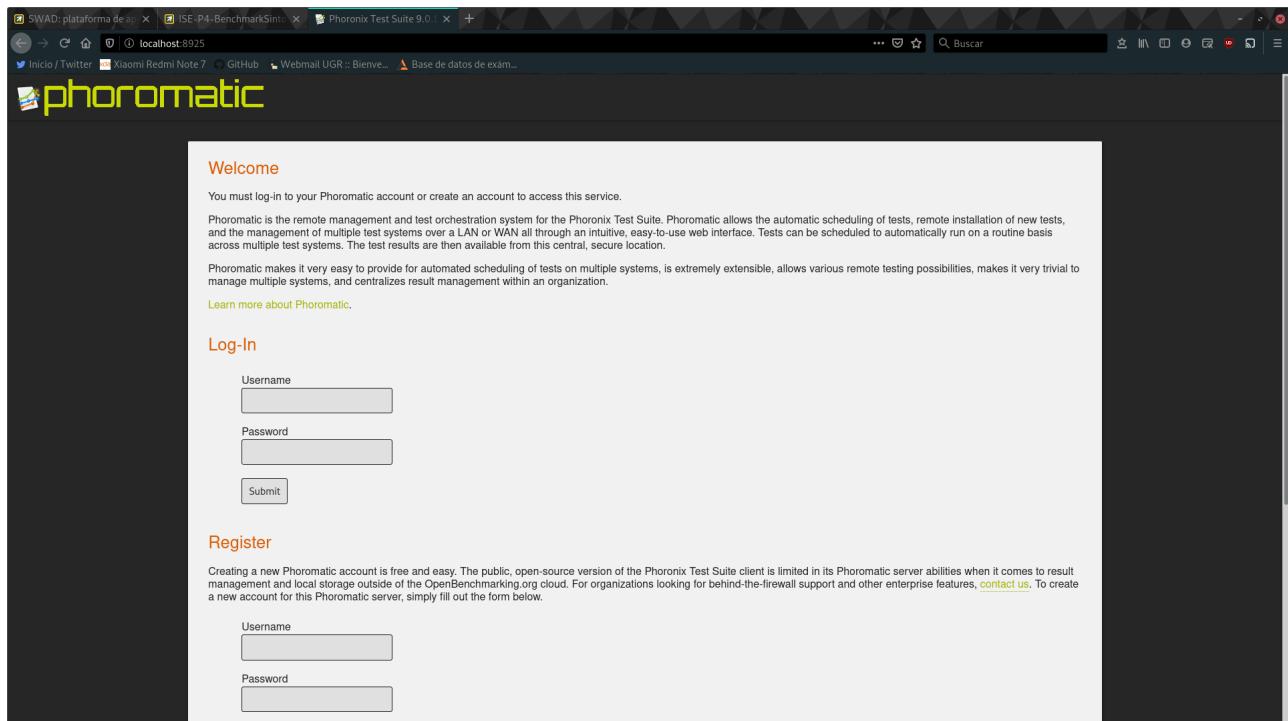
Launching with PHP built-in web server.

Could not open input file: pts-core/phoronix-test-suite.php
Could not open input file: pts-core/phoronix-test-suite.php
The Phoromatic Web Interface Is Accessible At: http://localhost:8193
Press [ENTER] to kill server...
```

## 1.6. Uso de Phoromatic

Como podemos ver en la imagen anterior, podemos acceder a Phoromatic a través de la ruta `http://localhost:8193`. Cada vez que iniciemos el servidor Phoromatic nos asignará un puerto, a no ser que especifiquemos un puerto en concreto en el archivo de configuración ubicado en `~/.phoronix-test-suite/`

Al entrar en dicha URL nos pedirá un login:



Tras registrarnos, encontraremos la siguiente página:

The screenshot shows a Linux desktop environment with several open windows. The main window is the Phoromatic web interface, version 9.0.1, running at localhost:8925. The interface has a dark theme with a navigation bar at the top labeled 'MAIN', 'SYSTEMS', 'TESTS', 'TESTING', and 'RESULTS'. Below the navigation bar, there's a search bar and a message about an outdated version of the software. The main content area displays system statistics: 0 Systems Running Tests, 3 Systems Idling, 0 Systems Shutdown, and 0 Systems Needing Attention. On the right side, it shows 0 Schedules Active Today, 0 Schedules In Total, 10 Active Benchmark Tickets, and 0 Test Results Today / 10 Benchmark Results Total. A section for 'Today's Scheduled Tests' is also present. Other windows visible in the background include 'SWAD: plataforma de...', 'ISE: P4-BenchmarkSuite...', and 'Phoronix Test Suite 9.0...'. The taskbar at the bottom shows icons for Inicio, Twitter, Xiaomi Redmi Note 7, GitHub, Webmail UGR, and Base de datos de exámenes.

En mi caso aparecen tres máquinas, ya que tengo añadidos los tres sistemas.

### 1.6.1. Añadir y conectar equipos a Phoromatic

Como vemos en la última imagen, añadir o conectar un sistema es tan sencillo como ejecutar el comando que nos indica el servidor:

1 \\$ phoronix-test-suite phonomatic.connect 192.168.1.74:8925/B2JGUJ

La ruta puede variar cada vez que iniciemos el servidor (a excepción de fijar una en los archivos de configuración), mientras que la última parte de la URL es un identificador único asociado al usuario de Phoromatic.

La primera vez que añadimos un equipo nos preguntará que nombre queremos asignarle dentro del servidor de Phoromatic. En mi caso los equipos se llaman CentOS-ISE, Ubuntu-ISE y antonio, siendo los dos primeros las máquinas virtuales de la asignatura y el último mi anfitrión.

En la pestaña de sistemas de Phoromatic podemos ver los sistemas que tenemos conectados, y cuando fue su ultima conexión. (La tasa de refresco es de aproximadamente uno a dos minutos.)

The screenshot shows the Phoromatic Test Systems interface. At the top, there are tabs for MAIN, SYSTEMS, TESTS, TESTING, and RESULTS. Below the tabs, there is a search bar and a 'Test Systems' section with a 'Add A System' button. The 'ACTIVE SYSTEMS' table lists three systems: 'ubuntu-ISE', 'CentOS-ISE 10.0.2.15', and 'antonio 192.168.1.74'. Each row shows the system name, status ('IDLING, WAITING FOR TASK'), and last communication time ('LAST COMMUNICATION: 9 DECEMBER 2019 17:33'). Below this is a 'System Groups' section with a 'New Group Name:' input field and a 'Select System(s) To Add To Group:' dropdown containing 'CentOS-ISE' and 'antonio'.

## 1.6.2. Crear Tests Suites

En la pestaña Tests podemos crear perfiles de test, así como nuevas suites personalizadas, en nuestro caso, vamos a crear una suite que tenga como test SmallPT:

The screenshot shows the Phoromatic Build Suite configuration page. The 'Title:' field is set to 'SmallPT'. The 'Suite Version:' field is set to '1.0.0'. The 'Description:' field contains the text 'Prueba básica para el procesador'. In the 'Tests in Schedule:' section, there is a list of tests: 'Smallpt [pts/smallpt-1.2.1]'. Below this, there is a note about 'Smallpt' being a C++ global illumination renderer. The 'Create Suite' button is visible at the bottom right.

Como vemos, podemos hacer que una suite ejecute un conjunto de tests, por si queremos, por

ejemplo, aplicar distintos tests de CPU a un equipo, sin tener que ejecutarlos uno a uno manualmente.

### 1.6.3. Ejecutar Suites

En la sección Testing tenemos disponible la opción Run a benchmark, donde podemos escoger que suite ejecutar y en qué sistemas ejecutarlos. En mi caso, al estar virtualizando dos de los sistemas en el anfitrión solo mandaré las pruebas a un único host simultáneamente, sin embargo, si fueran servidores distintos podríamos mandar simultáneamente suites a ejecutar.

The screenshot shows the Phoronix Test Suite 9.0 interface on a Linux desktop. The main window title is "ISE-P4-BenchmarkSuite". The left sidebar lists various benchmarks: PHP-CentOS, PHP-Arch, PHP-Ubuntu, Apache-Arch, SmallPT-CentOS, SmallPT-Ubuntu, SmallPT-Arch, and Prueba FFmpeg. Below the sidebar, a status bar displays "17:45 CET - 9 DECEMBER 2019", "3 SYSTEMS", "0 SUITES", "10 RESULTS", and "0 ACTIVITY EVENTS TODAY". The main configuration area includes fields for "Title" (set to "Procesador-Ubuntu"), "Test Run Identifier" (set to "Procesador-Ubuntu1"), "Test Suite To Run" (set to "Smallpt - smallpt-1.0.0"), and a "Description" section with a large text input field. Under "System Targets", there is a list of systems: "CentOS-ISE" (unchecked), "antonio" (unchecked), and "ubuntu-ISE" (checked). A "Stress Testing" section at the bottom contains a note about testing stability/reliability rather than performance. The overall interface is dark-themed with light-colored text and buttons.

Vemos como también nos permite establecer el numero de concurrencia, así como el tiempo mínimo que estará ejecutando el test.

SWAD: plataforma de ap... x ISE-P4-BenchmarkSuite x Phoronix Test Suite 9.0.0 +

localhost:8925/benchmark

Inicio / Twitter Xiaomi Redmi Note 7 GitHub Webmail UGR - Bienven... Base de datos de exámenes

Test Suite: 10 runs.

**Build a suite** to add/select more tests to run or [view local suites](#) for more information on a particular suite. A test suite is a set of test profiles to run in a pre-defined manner.

Smallpt - smallpt-1.0.0

**Description:**

The description is an optional way to add more details about the intent or objective of this test run.

[Empty text area]

---

**System Targets:**

Select the systems that should be benchmarked at their next earliest convenience.

Systems:  CentOS-ISE  antonio  ubuntu-ISE

---

**Environment Options**

**Stress Testing**

If you wish to test systems for stability/reliability rather than performance, use this option and specify the number of tests to run concurrently (two or more) and (optionally) for the total period of time to continue looping the benchmarks. These options are intended to just stress the system and will not record any benchmark results. From the command-line this testing mode can be used via the *phoronix-test-suite stress-run* sub-command.

Concurrent Number Of Test Processes: Disabled

Force Loop Time: Disabled

---

**Run Benchmark**

17:45 CET - 10 December 2018  
Copyright © 2008 - 2018 by Phoronix Media. All rights reserved.  
All trademarks used are properties of their respective owners.  
Phoronix Test Suite v9.0.1 (Akamai) - Core Version 9010 - PHP 7.4.0

OpenBenchmarking.org ptsli

Una vez mandamos el benchmark, phoronix-test-suite se encargará de instalarlo, ejecutarlo, y devolver los resultados a Phoromatic.

VirtualBox Machine | localhost:8925/ | Inicio / Twitter | Xiaomi Redmi Note 7 | GitHub | Webmail UGR :: Bienven... | Base de datos de exam... | ... | Buscar

# phoromatic

MAIN SYSTEMS TESTS TESTING RESULTS

SEARCH

An outdated version of the Phoronix Test Suite / Phoromatic is currently installed. The version in use is v9.0.1 (v9010), but the latest is pts-core v9200. Visit [Phoromatic-Test-Suite.com](#) to update this software.

## Phoromatic

To get started with your new account, the basic steps to get started include:

Connect/sync the Phoronix Test Suite client systems (the systems to be benchmarked) to this account. In the simplest form you just need to run the following command on the test systems: `phoronix-test-suite phoromatic connect 192.168.1.74:8925/B2JGUJ`. For more information view the instructions on the [systems page](#).

Configure your account settings.

Create a test schedule. A schedule is for running test(s) on selected system(s) on a routine, timed basis or whenever a new Git commit is applied to a code-base, or other events occurred. You can also enrich the potential by adding a benchmark ticket for one-time testing on one or more systems.

View the automatically generated [test results](#).

If you are interested in Phoromatic and the Phoronix Test Suite for enterprise testing, please [contact us](#) for commercial support and sponsorship that we can continue to develop this leading open-source Linux distribution. If you would like to contribute patches, you can do so via our [GitHub project](#).

Ubuntu-ISE (P4 - descargado iSE4JMeter - falta configurar y probar con jMeter) [Corriendo] - Oracle VM VirtualBox

Archivo Máquina Ver Entrada Dispositivos Ayuda

[17:36:49] Idling, waiting for task assignment...

[17:37:50] Idling, waiting for task assignment...

[17:38:51] Idling, waiting for task assignment...

[17:39:51] Idling, waiting for task assignment...

[17:40:52] Idling, waiting for task assignment...

[17:41:52] Idling, waiting for task assignment...

[17:42:53] Idling, waiting for task assignment...

[17:43:54] Idling, waiting for task assignment...

[17:44:54] Idling, waiting for task assignment...

[17:45:55] Idling, waiting for task assignment...

[17:46:55] Idling, waiting for task assignment...

[17:47:56] Idling, waiting for task assignment...

[17:48:57] Idling, waiting for task assignment...

[17:49:57] Idling, waiting for task assignment...

Ubuntu-ISE Repository Index  
pts: 168 Distinct Tests, 360 Test Versions, 28 Suites

Ubuntu-ISE Repository Index  
system: 13 Distinct Tests, 40 Test Versions, 9 Suites

Installed: pts/smallpt-1.2.1  
pts/smallpt-1.2.1: Test Started

CtrlDerecho

### 1.6.4. Mostrar resultados

En la opción de Results tenemos disponibles los distintos resultados:

Test Name	Date	Viewed
Apache-CentOS CENTOS-ISE	26 NOVEMBER 17:26	4 TIMES VIEWED
Apache-Ubuntu UBUNTU-ISE	26 NOVEMBER 16:35	5 TIMES VIEWED
PHP-CentOS CENTOS-ISE	26 NOVEMBER 15:19	1 TIMES VIEWED
PHP-Arch ANTONIO	26 NOVEMBER 15:17	1 TIMES VIEWED
PHP-Ubuntu UBUNTU-ISE	26 NOVEMBER 15:15	1 TIMES VIEWED
Apache-Arch ANTONIO	26 NOVEMBER 13:04	5 TIMES VIEWED
SmallPT-CentOS CENTOS-ISE	26 NOVEMBER 12:50	1 TIMES VIEWED
SmallPT-Ubuntu UBUNTU-ISE	26 NOVEMBER 12:38	2 TIMES VIEWED
Prueba FFmpeg ANTONIO	26 NOVEMBER 12:05	0 TIMES VIEWED
SmallPT-Arch ANTONIO	26 NOVEMBER 11:59	2 TIMES VIEWED

Podemos escoger varios (o solo uno) y comparar los resultados, pinchando en la nueva opción del menú: Compare

Test Name	Date	Viewed
Apache-CentOS CENTOS-ISE	26 NOVEMBER 17:26	4 TIMES VIEWED
Apache-Ubuntu UBUNTU-ISE	26 NOVEMBER 16:35	5 TIMES VIEWED
PHP-CentOS CENTOS-ISE	26 NOVEMBER 15:19	1 TIMES VIEWED
PHP-Arch ANTONIO	26 NOVEMBER 15:17	1 TIMES VIEWED
PHP-Ubuntu UBUNTU-ISE	26 NOVEMBER 15:15	1 TIMES VIEWED
Apache-Arch ANTONIO	26 NOVEMBER 13:04	5 TIMES VIEWED
SmallPT-CentOS CENTOS-ISE	26 NOVEMBER 12:50	1 TIMES VIEWED
SmallPT-Ubuntu UBUNTU-ISE	26 NOVEMBER 12:38	2 TIMES VIEWED
Prueba FFmpeg ANTONIO	26 NOVEMBER 12:05	0 TIMES VIEWED
SmallPT-Arch ANTONIO	26 NOVEMBER 11:59	2 TIMES VIEWED

Nos aparecerá esta pestaña, donde tenemos información de cada sistema, así como los resultados:

Phoronix Test Suite 9.0.1

SmallPT-Ubuntu

	SPT-UBU	SPT-ARCH	SPT-COS
image: Global Illumination Renderer; 128 Samples	191.76	41.10	191.31
Difference	1.00%	1.79%	1.05%
Standard Error	4.96	0.60	1.63
Standard Deviation	8.59%	0.62%	1.46%

Smallpt v1.0  
Global Illumination Renderer; 128 Samples

Method	Time (Seconds)	Standard Error (SE)	N
SPT-UBU	191.76	±4.96	11
SPT-ARCH	41.10	±0.60	15
SPT-COS	191.31	±1.63	3

1. (CXX) g++ options: -fopenmp -O3

17:55 CET 9 December 2019  
Copyright © 2008 - 2019 by Phoronix Media. All rights reserved.

OpenBenchmarking.org

## 2. Apache AB

Apache AB (Apache Benchmark) es un software diseñado y distribuido por Apache para comprobar y medir el rendimiento del servicio Apache, es decir, medir cuantas peticiones es capaz de servir nuestro servidor con Apache instalado.

### 2.1. Instalación de Apache AB

AB es instalado junto a Apache, esto quiere decir que en los equipos usados, al instalar apache2 en el caso de Ubuntu Server o httpd en el caso de CentOS el software AB ya es instalado automáticamente.

### 2.2. Uso de AB

Como vemos en la documentación oficial de AB[7] tenemos distintas opciones para usar AB, desde opciones para hacer login en caso de que el servidor este protegido, protocolo de seguridad a usar, manejo de cookies, carga de datos desde archivos csv entre muchas otras opciones.

Nosotros nos centraremos en las opciones -n y -c:

- **Opción -n:** Con este parámetro podemos establecer el número de peticiones que se van a realizar al servidor. Su valor por defecto es 1, por lo que no suele ser representativo y no es suficiente para obtener un benchmark realista.
- **Opción -c:** Número de peticiones concurrentes, es decir, el número de peticiones que se realizarán a la vez. Su valor por defecto es 1.

También cabe mencionar que aunque exista la opción -c no existe un paralelismo real, es decir, nuestra máquina solo ejecutará una instancia de AB en la que realizará muchas peticiones por segundo, pero el asignar un número N de concurrencia no significa ejecutar N instancias de AB. Sin embargo, para el servidor si serán peticiones distintas, por lo que tendrá que generar una instancia de apache por cada petición.

The screenshot shows a terminal window with two main sections. On the left, a command-line interface displays system status and a grep search. On the right, a graphical interface shows ApacheBench benchmarking results and a system monitor.

**Left Terminal (CLI):**

```
+ ps -a
  PID TTY      TIME CMD
 1148 ttys2    00:07:07 Xorg
 1274 ttys2    00:00:03 ls
20472 pts/2    00:00:00 ssh
21036 pts/1    00:00:02 ab
21155 pts/0    00:00:00 ps

+
+ ab -n 50000 -c 300 http://192.168.56.105/
This is ApacheBench, Version 2.3 <Revision: 1843412 >
Copyright 1996 Adam Twiss, Zeus Technology Ltd, http://www.zeustech.net/
Licensed to The Apache Software Foundation, http://www.apache.org/
Benchmarking 192.168.56.105 (be patient)
Completed 5000 requests
Completed 10000 requests
Completed 15000 requests
Completed 20000 requests
Completed 25000 requests
Completed 30000 requests
Completed 35000 requests
Completed 40000 requests
[...]
```

**Right Terminal (Grep and System Monitor):**

```
Archivo Editor Ver Terminal Pestañas Ayuda
www-data 3223 0.1 0.4 377216 10528 ?
www-data 3227 0.1 0.4 377216 10528 ?
www-data 3228 0.1 0.4 377216 10528 ?
www-data 3229 0.1 0.4 377216 10528 ?
www-data 3230 0.1 0.4 377228 10528 ?
www-data 3231 0.1 0.4 377228 10528 ?
www-data 3232 0.1 0.4 377228 10528 ?
www-data 3233 0.1 0.4 377216 10528 ?
www-data 3234 0.1 0.4 377228 10528 ?
www-data 3235 0.1 0.4 377228 10528 ?
www-data 3236 0.1 0.4 377216 10528 ?
www-data 3237 0.1 0.4 377228 10528 ?
www-data 3238 0.1 0.4 377228 10528 ?
www-data 3239 0.1 0.4 377216 10528 ?
www-data 3240 0.1 0.4 377216 10528 ?
www-data 3241 0.1 0.4 377216 10528 ?
www-data 3242 0.1 0.4 377228 10528 ?
www-data 3243 0.1 0.4 377216 10528 ?
www-data 3244 0.1 0.4 377228 10528 ?
www-data 3245 0.1 0.4 377216 10528 ?
www-data 3246 0.1 0.4 377228 10528 ?
www-data 3247 0.1 0.4 377228 10528 ?
www-data 3250 0.1 0.4 377216 10528 ?
www-data 3251 0.1 0.4 377228 10528 ?
www-data 3258 0.0 0.4 377216 10516 ?
www-data 3259 0.0 0.4 377216 10516 ?
www-data 3260 0.0 0.4 377228 10520 ?
antonio+ 3262 0.0 0.0 16756 1032 pts/0 S+ 10:58
antoniovyg@ubuntu:~ 1 2 3 4 5 6
```

antonio@antonio: ~ 16.60 GB ~ 70.28 GB █ 42.63% ⚡ 17% [50-59]:54 🔍 48% 0:45 ◀ 25% 70% O Thu 12/12 10:58:41

## 2.3. Interpretar la salida de AB

La ejecución de AB nos dará la siguiente salida:

```
Benchmarking 192.168.56.105 (be patient)
Completed 5000 requests
Completed 10000 requests
Completed 15000 requests
Completed 20000 requests
Completed 25000 requests
Completed 30000 requests
Completed 35000 requests
Completed 40000 requests
Completed 45000 requests
Completed 50000 requests
Finished 50000 requests

Server Software: Apache/2.4.18
Server Hostname: 192.168.56.105
Server Port: 80

Document Path: /
Document Length: 11315 bytes

Concurrency Level: 300
Time taken for tests: 27.473 seconds
Complete requests: 50000
Failed requests: 0
Total transferred: 579450000 bytes
HTML transferred: 565750000 bytes
Requests per second: 1819.97 [#/sec] (mean)
Time per request: 164.837 [ms] (mean)
Time per request: 0.549 [ms] (mean, across all concurrent requests)
Transfer rate: 20597.35 [Kbytes/sec] received

Connection Times (ms)
              min  mean[+/-sd] median   max
Connect:        0   34 217.6     0   7102
Processing:     3   97 624.5    52  27440
Waiting:        2   78 618.2    51  27437
Total:          3  131 670.0    52  27451

Percentage of the requests served within a certain time (ms)
  50%   52
  66%   57
  75%   61
  80%   66
  90%  123
  95%  456
  98% 1074
  99% 1284
100% 27451 (longest request)

~ took 28s
```

Como vemos en la documentación de AB[8] la salida de AB se compone de:

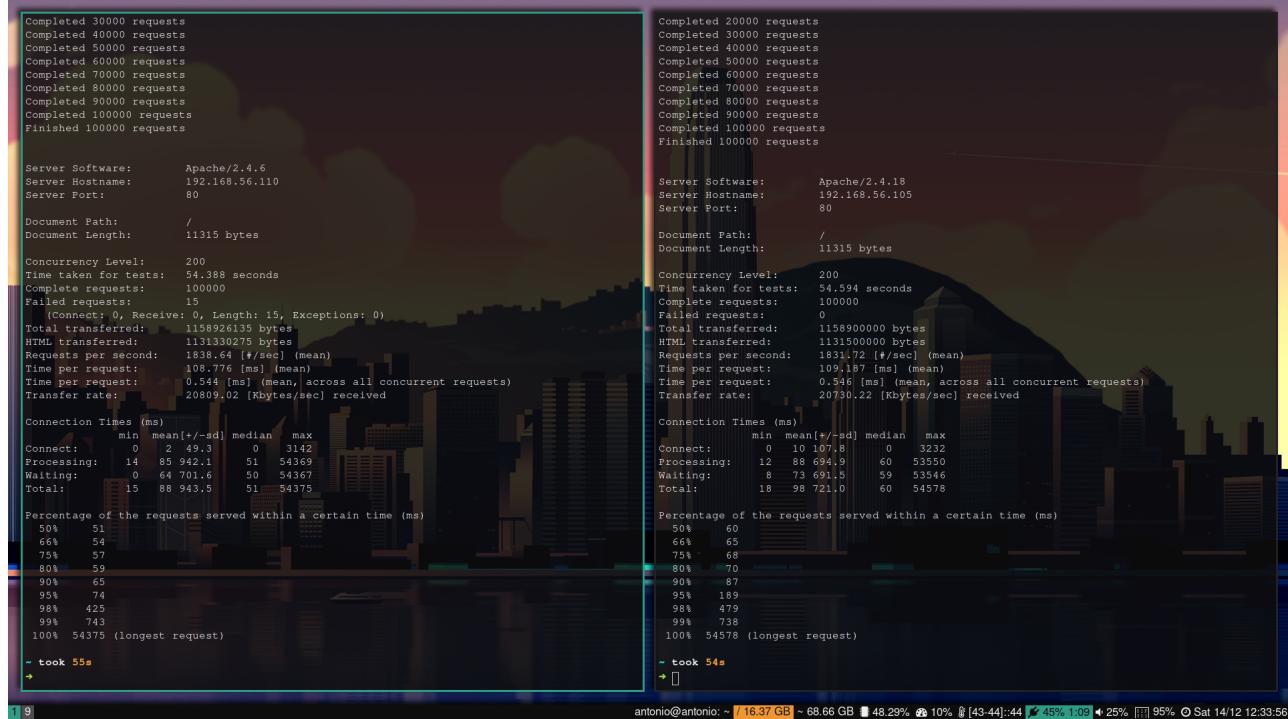
1. Información sobre el servidor (Server Software, Server Hostname y Server Port), en nuestro caso el servidor ejecuta una versión de Apache 2.4.18, el hostname es 192.168.56.105 y estamos accediendo a través del puerto 80.
2. Información sobre el documento solicitado(Document Path y Document Length). El documento solicitado se encuentra en la ruta / (raíz) del servidor Apache, y tiene una longitud de

11315 bytes.

3. Nivel de concurrencia. Corresponde al parámetro -c pasado en la ejecución de AB. Por defecto vale 1.
4. Tiempo que ha necesitado el test. En nuestro 27.473 segundos.
5. Peticiones completadas. En nuestro ejemplo 50.000, es decir, todas.
6. Peticiones fallidas. En nuestro ejemplo 0.
7. Total de datos transferidos. En nuestro caso 579.450.000 bytes.
8. HTML transferido. En nuestro caso 565.750.000 bytes, que corresponden a la longitud del documento por el número de peticiones.
9. Media de peticiones por segundo. En nuestro Apache ha servido una media de 1819.97 peticiones por segundo.
10. Tiempo medio por petición (secuencial). En nuestro ejemplo 164.837 ms.
11. Tiempo medio por petición (concurrente). En nuestro ejemplo 0.549 ms.
12. Velocidad de transmisión. En nuestro ejemplo 20597 KBytes/segundo.
13. Tiempos de conexión. Nos muestra los tiempos medios de conexión, procesado, espera y el total.
14. Porcentaje de peticiones servidas a los T milisegundos. Normalmente esta medición la hacemos con el 99 % de las peticiones, es decir, la mayoría de estas.

## 2.4. Comparación CentOS y Ubuntu Server

Para realizar esta prueba he clonado el archivo `index.html` por defecto de Apache en Ubuntu Server en CentOS, asegurando que el tamaño de bytes a transferir es el mismo, para poder realizar una prueba fiable.



Vemos como el número de bytes transferidos es el mismo, tanto en el total como en HTML.

De esta comparación podemos obtener que CentOS es ligeramente más rápido al debido al tiempo de servicio de las peticiones, sin embargo esta prueba es demasiado pequeña ya que únicamente transferimos un archivo HTML básico.

## 3. JMeter

JMeter[9] es un proyecto desarrollado por Apache con el objetivo de desarrollar una herramienta de prueba de carga para analizar y medir el desempeño tanto de un servidor como de los servicios que este pueda ofrecer.

### 3.1. Instalación de JMeter

Para instalar JMeter en mi anfitrión (ArchLinux) podemos instalarlo desde el AUR (Arch User Repository)

```
1 \$ yay -S jmeter
```

En CentOS y Ubuntu no será necesario instalar JMeter, ya que las peticiones las haremos desde nuestro anfitrión.

### 3.2. Instalando el microservicio iseP4JMeter

Para realizar la práctica se nos pide usar un microservicio dado por los profesores de la asignatura[10].

Para usar este microservicio debemos instalar primero docker y docker-compose.

#### 3.2.1. Instalación de Docker y Docker-compose en Ubuntu Server

Siguiendo los pasos del guión de prácticas:

Primero debemos añadir la llave GPG de Docker a APT:

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

Añadimos el repositorio de Docker:

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

Actualizamos la lista de repositorios e instalamos Docker y Docker-compose:

```
1 \$ sudo apt update  
2 \$ sudo apt install docker-ce docker-compose
```

Añadimos a nuestro usuario al grupo de docker:

```
1 \$ sudo usermod -aG docker \$USER
```

Tras esto ya tenemos todo lo necesario instalado, aunque para que esta última instrucción tenga efecto debemos cerrar y volver a abrir la sesión.

### 3.2.2. Instalación de Docker y Docker-compose en CentOS

En el guión de prácticas no encontraremos como instalar docker en CentOS, pero con una búsqueda rápida encontramos la documentación de Docker[11].

Instalamos el paquete yum-utils para gestionar los repositorios de yum:

```
1 \$ sudo yum install -y yum-utils device-mapper-persistent-data lvm2
```

Añadimos el repositorio de Docker:

```
1 \$ sudo yum-config-manager --add-repo  
→ https://download.docker.com/linux/centos/docker-ce.repo
```

Instalamos Docker y Docker-compose:

```
1 \$ sudo yum install docker-ce docker-compose
```

Añadimos a nuestro usuario al grupo de docker:

```
1 \$ sudo usermod -aG docker \$USER
```

Activamos y ejecutamos el servicio de Docker:

```
1 \$ sudo systemctl start docker  
2 \$ sudo systemctl enable docker
```

Tras esto ya tenemos Docker y Docker-compose instalado.

### 3.2.3. Instalación y ejecución de iseP4JMeter

Para ejecutar el microservicio necesario para realizar la práctica basta con clonar el repositorio del software[10] y ejecutar docker-compose dentro de el.

```
1 \$ git clone https://github.com/davidPalomar-ugr/iseP4JMeter.git  
2 \$ cd iseP4JMeter  
3 \$ docker-compose up -d
```

Esto hará que se instale todo lo necesario y comience la ejecución del microservicio.

En el archivo docker-compose.yml encontraremos toda la configuración para docker-compose, como por ejemplo que ejecutará docker-compose al iniciarse y la configuración de cada servicio como el puerto de nodejs (3000 por defecto), por el que accederemos para realizar la práctica o el puerto de la base de datos en mongodb.

Dentro de cada servicio (nodejs, mongodb) encontraremos su respectivo archivo Dockerfile, que contiene la información necesaria de docker para ser ejecutado.

Para parar el servicio simplemente ejecutamos:

```
1 \$ docker-compose down
```

Cabe añadir que el microservicio usará el puerto 3000, en un principio deberíamos añadir este puerto al firewall ya sea con `ufw` o `firewall-cmd`, sin embargo el propio docker se encargará de añadir las reglas necesarias a `iptables` para que estas rutas sean accesibles desde cualquier sitio.

### 3.3. Prueba básica de conexión: Primer contacto con JMeter

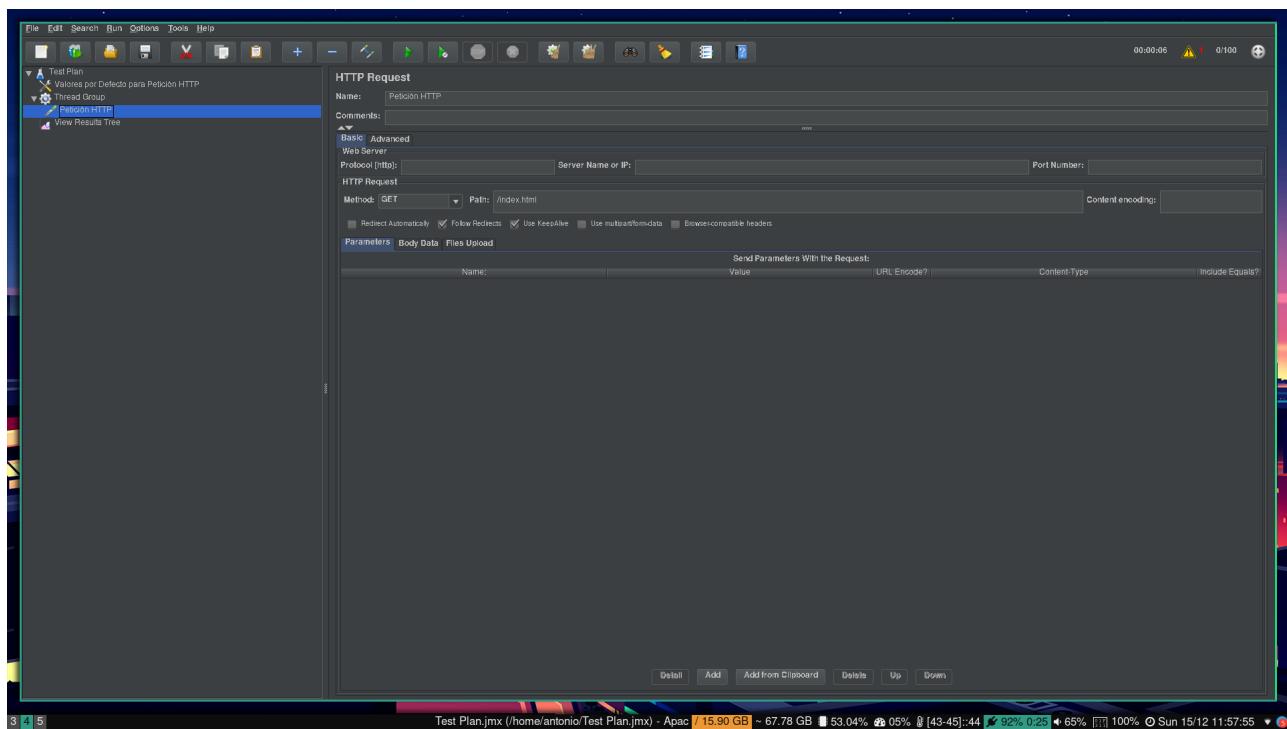
Para empezar a trabajar con JMeter realizaremos un test básico de conexión con nuestro servidor.

Para esta prueba básica añadiremos al Test Plan un elemento de configuración "HTTP Request default" donde estableceremos la IP del servidor a 192.168.56.105, y el puerto lo dejaremos en blanco para que use el puerto por defecto (80). Más adelante veremos como parametrizar estos valores.

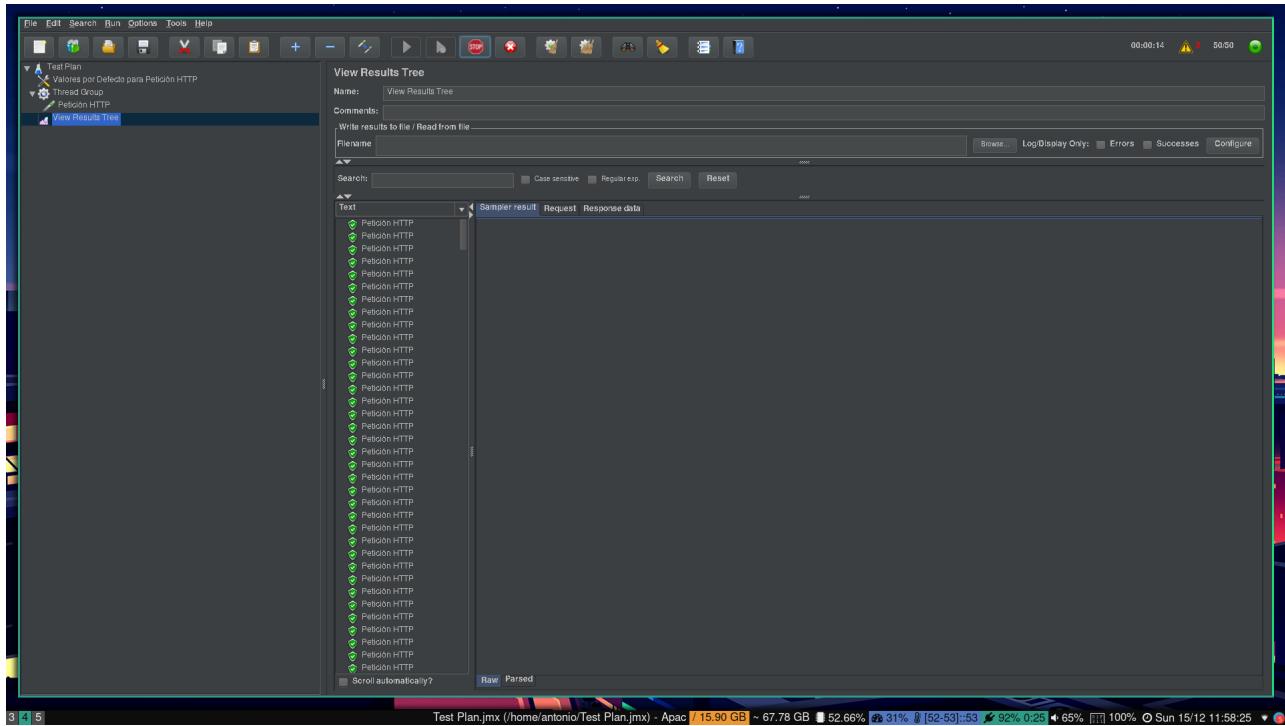
Añadiremos un Thread Group, es decir, un grupo de usuario que se encargarán de realizar las peticiones. En esta sección podemos configurar el número de peticiones a realizar, el número de usuarios o incluso si queremos que este en un bucle infinito pidiendo peticiones.

Dentro del Thread Group añadiremos una petición HTTP, haciendo click derecho sobre el Thread Group ->Add ->Sampler ->HTTP Request.

Configuramos el HTTP Request para que obtenga la ruta `/index.html`



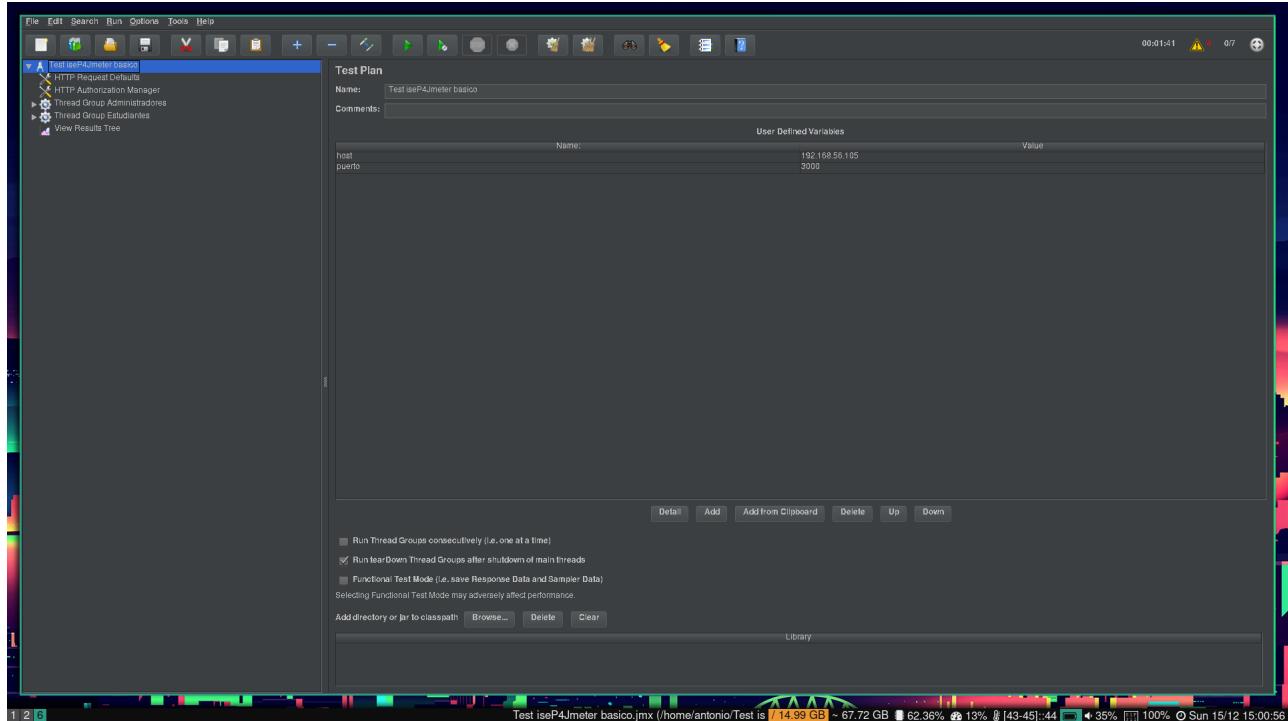
Al ejecutar vemos como tenemos conexión con el servidor.



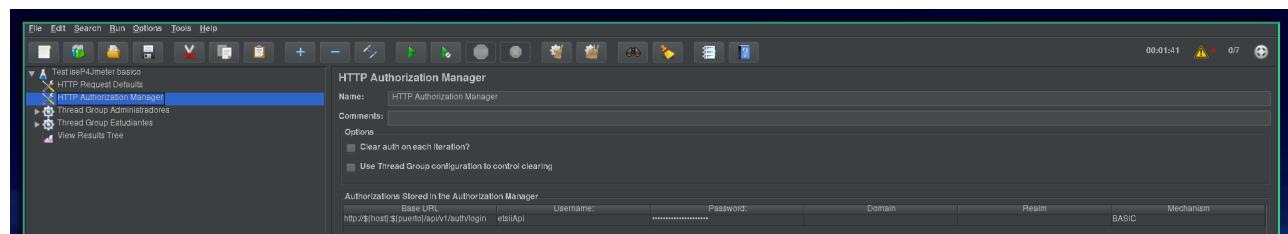
## 3.4. Configurando un test de carga para iseP4JMeter

### 3.4.1. Configuración general

Al igual que con la prueba básica, creamos un elemento de configuración de HTTP, sin embargo, esta vez configuraremos la IP con el valor \${host} y el puerto con \${puerto}. Esto hace referencia a variables, las cuales crearemos en la pestaña del Test, en User Defined Variables. En mi caso estableceré host a 192.168.56.105 ya que uso Ubuntu para ejecutar el microservicio y como puerto el 3000 porque no he modificado el que venia por defecto.



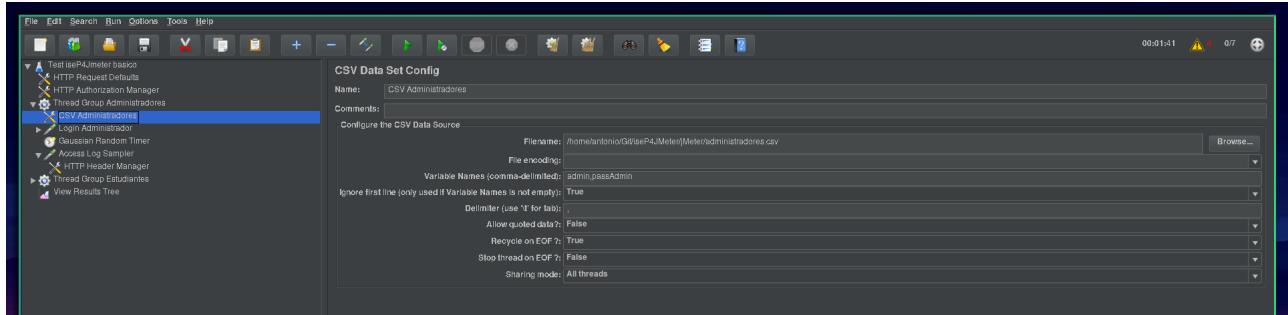
Como vemos en la imagen también añadimos un elemento de configuración "HTTP Authorization Manager" para gestionar el acceso de BasicAuth, donde añadimos el usuario y clave dados por la aplicación.



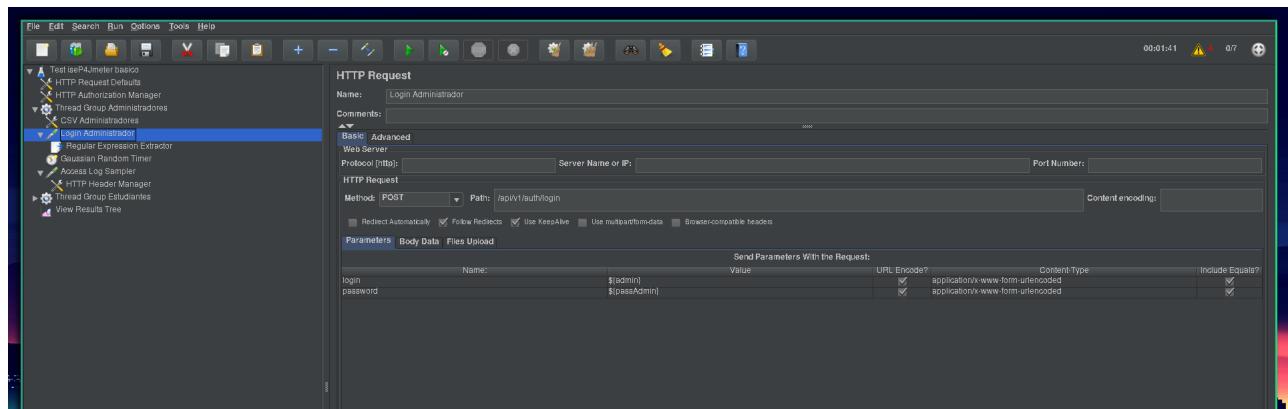
También añadimos dos Thread Group, uno para estudiantes y otro para administradores, así como un View Results Tree para ver los resultados.

### 3.4.2. Configuración para administradores

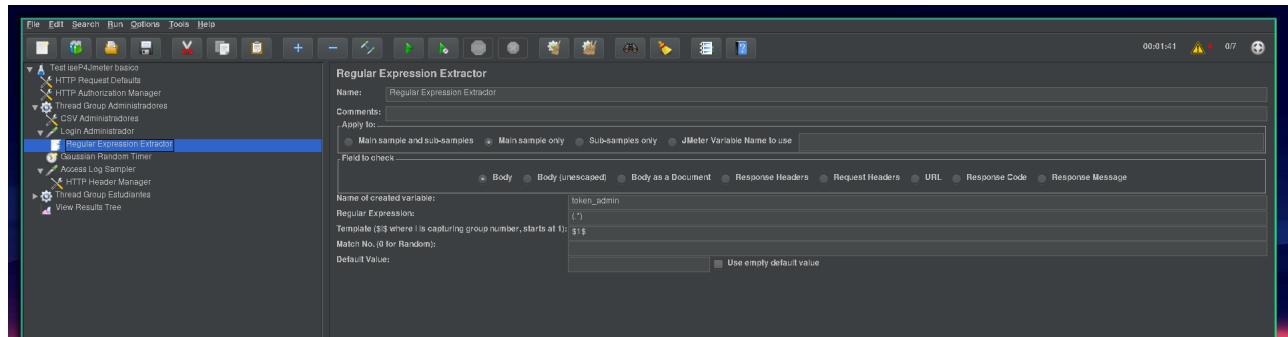
Dentro del Thread Group para administradores tenemos que añadir un CSV Data Set Config para que los datos de inicio los tome de un archivo. Dicho archivo tendrá dos columnas, una para el usuario y otra para la contraseña, así que configuraremos dos variables en JMeter. También tenemos que seleccionar que ignore la primera fila ya que esta fila es usada para marcar que columna es el usuario y la contraseña.



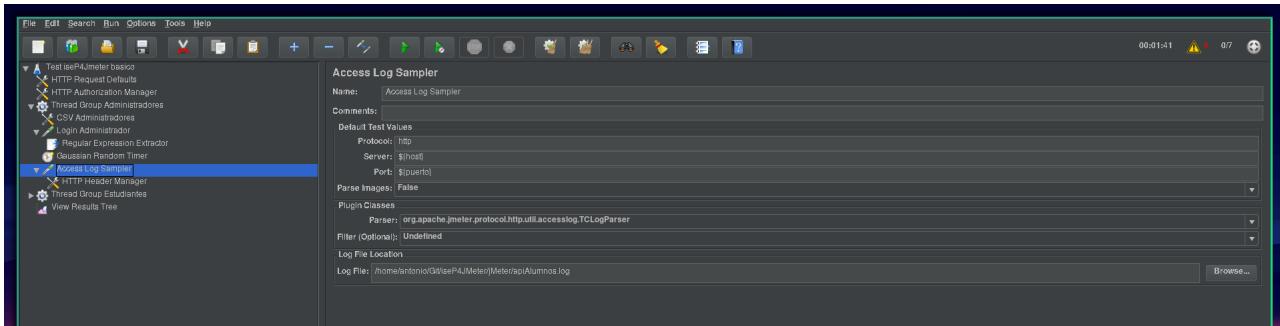
Crearemos un HTTP Request para que el administrador se inicie sesión. Esta petición será de tipo POST a la ruta /api/v1/auth/login, y le pasaremos dos parámetros, login y password, con los valores recogidos del fichero CSV. El tipo de contenido de estos parámetros será application/x-www-form-urlencoded.



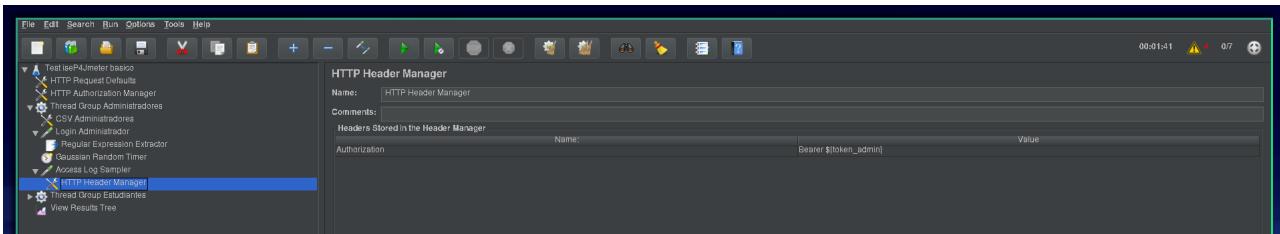
Una vez hecha la petición de login el servidor nos devolverá el JWT Token del administrador, el cuál usaremos para referirnos a este administrador en futuras peticiones. Almacenaremos este token con un extractor de expresiones regulares, y almacenaremos todo el cuerpo de la respuesta en una variable llamada token\_admin, que más adelante usaremos.



Como nos pide el ejercicio, añadimos una espera aleatoria y tras eso añadimos el Access Log Sampler, con el que simularemos el muestreo de acceso de los administradores.



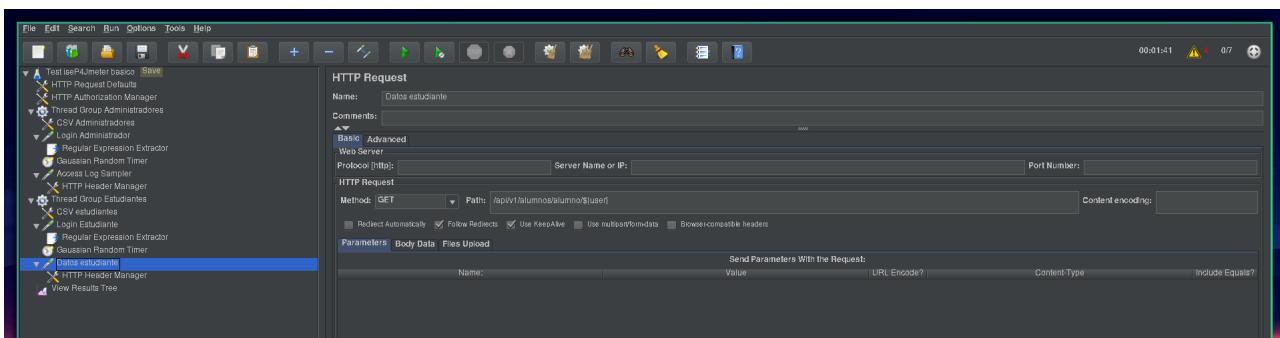
Para realizar esto, debemos añadir al Access Log Sampler una cabecera de HTTP que adjunte el token obtenido anteriormente.

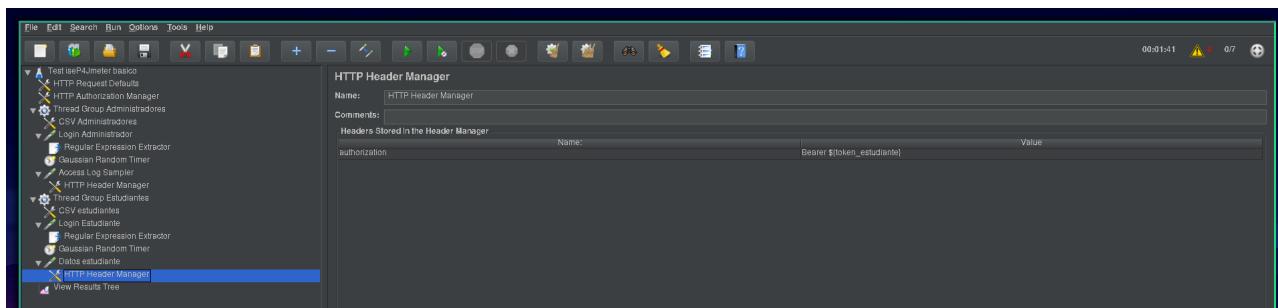


### 3.4.3. Configuración para estudiantes

Al igual que para los administradores, configuraremos el CSV Data Set Config y el login de los estudiantes de la misma forma que para los administradores, solo que usando los datos del archivo alumnos.csv

Además de las opciones de login, debemos añadir una segunda petición HTTP, que en este caso será de tipo GET, con el que obtendremos los datos del estudiante. Pediremos que nos de la ruta /api/v1/alumnos/alumno/\${user}, añadiendo un gestor de cabeceras HTTP que adjunte el token, como hicimos con el Acces Log Sampler del administrador.

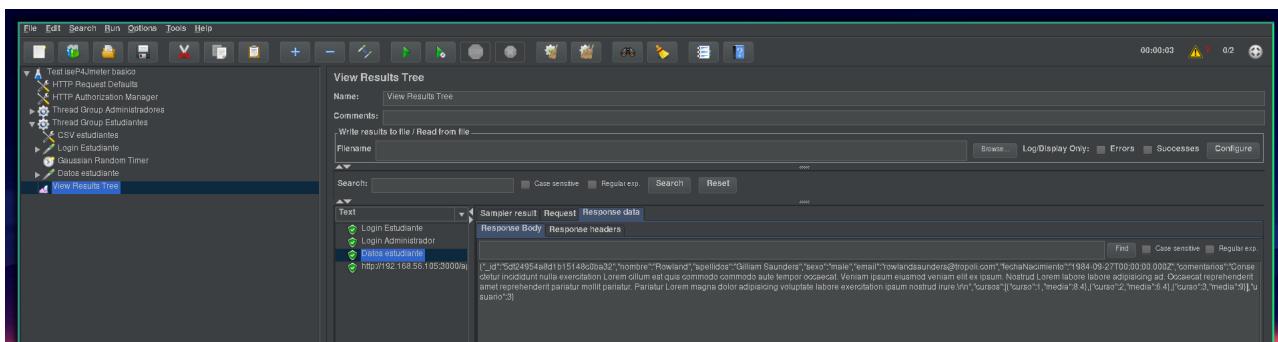




### 3.4.4. Resultados

Finalmente, si ejecutamos el test de JMeter veremos como todo funciona correctamente.

En mi caso he establecido una única ejecución en los Thread Group por legibilidad.



## Bibliografía

- [1] Phoronix Test Suite <https://www.phoronix-test-suite.com/>
- [2] OpenBenchmarking <https://www.openbenchmarking.org/>
- [3] Yet Another Yogurt (AUR Helper) <https://github.com/Jguer/yay>
- [4] Phoronix Test Suite - Descarga <https://www.phoronix-test-suite.com/?k=downloads>
- [5] Docker Phoronix Test Suite <https://hub.docker.com/r/phoronix/pts/>
- [6] Man phoronix-test-suite <https://linux.die.net/man/1/phoronix-test-suite>
- [7] Documentación de AB <https://httpd.apache.org/docs/2.4/programs/ab.html>
- [8] Salida de ejecución de AB <https://httpd.apache.org/docs/2.4/programs/ab.html#output>
- [9] Página principal de JMeter <https://jmeter.apache.org/>
- [10] Página de iseP4JMeter <https://github.com/davidPalomar-ugr/iseP4JMeter>
- [11] Instalación de Docker en CentOS <https://docs.docker.com/v17.09/engine/installation/linux/docker-ce/centos/#set-up-the-repository>