

INGENIERÍA DE SERVIDORES (2016-2017)  
GRADO EN INGENIERÍA INFORMÁTICA  
UNIVERSIDAD DE GRANADA

---

## Memoria Práctica 4

---

Elena María Gómez Ríos

23 de diciembre de 2016

## Índice

<b>1. Cuestión 1:</b>	<b>5</b>
1.1. a) ¿Seleccione, instale y ejecute uno, comente los resultados. Atención: no es lo mismo un benchmark que una suite, instale un benchmark. . . . .	5
<b>2. Cuestión 2:</b>	<b>13</b>
2.1. De los parámetros que le podemos pasar al comando ¿Qué significa -c 5? ¿y -n 100? Monitorice la ejecución de ab contra alguna máquina (cualquiera) ¿cuántas “tarefas” crea ab en el cliente? . . . . .	13
<b>3. Cuestión 3:</b>	<b>16</b>
3.1. Ejecute ab contra a las tres máquinas virtuales (desde el SO anfitrión a las máquina virtuales de la red local) una a una (arrancadas por separado).¿Cuál es la que proporciona mejores resultados? Muestre y coméntelos. (Use como máquina de referencia Ubuntu Server para la comparativa). . .	16
<b>4. Cuestión opcional 1:</b>	<b>20</b>
4.1. ¿Qué es Scala? Instale Gatling y pruebe los escenarios por defecto. . . . .	20
<b>5. Cuestión 4:</b>	<b>25</b>
5.1. Instale y siga el tutorial en <a href="http://jmeter.apache.org/usermanual/build-web-test-plan.html">http://jmeter.apache.org/usermanual/build-web-test-plan.html</a> [1] realizando capturas de pantalla y comentándolas. En vez de usar la web de jmeter, haga el experimento usando sus máquinas virtuales ¿coincide con los resultados de ab? . . . . .	25
<b>6. Cuestión 5:</b>	<b>32</b>
6.1. Programe un benchmark usando el lenguaje que desee. El benchmark debe incluir: . . . . .	32
6.1.1. Objetivo del benchmark. . . . .	32
6.1.2. Métricas (unidades, variables, puntuaciones, etc.). . . . .	33
6.1.3. Instrucciones para su uso. . . . .	33
6.1.4. Ejemplo de uso analizando los resultados. . . . .	33

## Índice de figuras

1.1. CentOS, instalación de Phoronix Suite. . . . .	6
1.2. CentOS, listado de benchmarks. . . . .	6
1.3. CentOS, instalación del benchmark Blender. . . . .	7
1.4. CentOS, Phoronix Suite GUI. . . . .	7
1.5. CentOS, Phoronix Suite GUI web. . . . .	8
1.6. CentOS, Phoronix Suite, listado de benchmarks. . . . .	8
1.7. CentOS, Phoronix Suite, listado de benchmarks instalados. . . . .	9
1.8. CentOS, Phoronix Suite, ejecución de benchmarks. . . . .	9

1.9. CentOS, Phoronix Suite, ejecución de benchmarks. . . . .	10
1.10. CentOS, Phoronix Suite, problemas con benchmark Blender. . . . .	10
1.11. CentOS, Phoronix Suite, ejecución de benchmark GLmark2. . . . .	11
1.12. CentOS, Phoronix Suite, ejecución de benchmark GLmark2. . . . .	11
1.13. CentOS, Phoronix Suite, benchmark GLmark2 finalizado. . . . .	12
1.14. CentOS, Phoronix Suite, resultado de benchmark GLmark2. . . . .	12
1.15. CentOS, Phoronix Suite, resultado de benchmark GLmark2. . . . .	13
2.1. CentOS, ab contra Windows Server. . . . .	14
2.2. CentOS, ab contra Windows Server. . . . .	14
2.3. CentOS, número de tareas de ab. . . . .	15
2.4. Windows, Adminitrador de tareas durante ejecución ab desde CentOS. . .	15
3.1. CentOS, resultado ab contra Windows. . . . .	17
3.2. CentOS, resultado ab contra Windows. . . . .	17
3.3. CentOS, desactivación del firewall. . . . .	18
3.4. CentOS, resultado ab contra CentOS. . . . .	18
3.5. CentOS, resultado ab contra CentOS. . . . .	19
3.6. CentOS, resultado ab contra Ubuntu Server. . . . .	19
3.7. CentOS, resultado ab contra Ubuntu Server. . . . .	20
4.1. CentOS, instalación de Gatling. . . . .	21
4.2. CentOS, ejecución de Gatling. . . . .	21
4.3. CentOS, resultados de Gatling, escenario 0. . . . .	22
4.4. CentOS, resultados de Gatling, escenario 0. . . . .	22
4.5. CentOS, resultados de Gatling, escenario 1. . . . .	23
4.6. CentOS, resultados de Gatling, escenario 1 . . . . .	23
4.7. CentOS, resultados de Gatling, escenario 5. . . . .	24
4.8. CentOS, resultados de Gatling, escenario 5 . . . . .	24
5.1. CentOS, JMeter - crear grupo de hilos. . . . .	26
5.2. CentOS, JMeter - grupo de hilos por defecto. . . . .	26
5.3. CentOS, JMeter - grupo de hilos. . . . .	27
5.4. CentOS, JMeter - añadir valores de las peticiones HTTP por defecto. . . .	27
5.5. CentOS, JMeter - valores de las peticiones HTTP por defecto. . . . .	28
5.6. CentOS, JMeter - gestor de Cookies de HTTP. . . . .	28
5.7. CentOS, JMeter - añadir peticiones HTTP. . . . .	29
5.8. CentOS, JMeter - peticiones HTTP. . . . .	29
5.9. CentOS, JMeter - añadir gráfico. . . . .	30
5.10. CentOS, JMeter - gráfico de resultados. . . . .	30
5.11. CentOS, JMeter - gráfico de resultados hacia Ubuntu Server. . . . .	31
5.12. CentOS, JMeter - gráfico de resultados hacia Windows. . . . .	31
5.13. CentOS, JMeter - gráfico de resultados hacia CentOS. . . . .	32
6.1. Ubuntu anfitrión, resultados del benchmark. . . . .	33
6.2. CentOS, resultados del benchmark. . . . .	33

## Índice de tablas

3.1. Resultados de ab. . . . .	16
6.1. Tiempos de la ejecución del benchmark. . . . .	34

## 1. Cuestión 1:

- 1.1. a) ¿Seleccione, instale y ejecute uno, comente los resultados.  
**Atención: no es lo mismo un benchmark que una suite, instale un benchmark.**

Para instalar Phoronix Suite en CentOS [3] se utiliza el comando `yum install phoronix-test-suite` (figura 1.1). Con el comando `phoronix-test-suite list-tests` se obtiene una lista de los benchmarks disponibles (figura 1.2). Para instalar un benchmark de la lista se utiliza el comando `phoronix-test-suite install <test>` (figura 1.3). La documentación utilizada para los comandos de phoronix ha sido [2], aunque podemos obtener la misma información con `man phoronix-test-suite`.

Podemos iniciar el cliente web con el comando `phoronix-test-suite gui` (figura 1.4), el cual abrirá una nueva ventana como se muestra en la figura 1.5 en la que aparece la monitorización de nuestro sistema. Desde aquí también podemos listar los test disponibles (figura 1.6) e instalarlos. En la pestaña de “Installed Tests” podemos ver los benchmarks ya instalados, en mi caso, como se muestra en la figura 1.7, aparece el “Blender” que es el benchmark que he instalado anteriormente.

Ejecutar el benchmark desde la GUI es muy sencillo, simplemente debemos pulsar sobre el benchmark que queremos ejecutar y nos saldrá una pantalla como la de la figura 1.8, si el benchmark no está instalado se instalará previamente. Mientras que se ejecuta el benchmark se muestra la siguiente pantalla de información, figura 1.9, donde informa del tiempo aproximado que va a tardar en completarse.

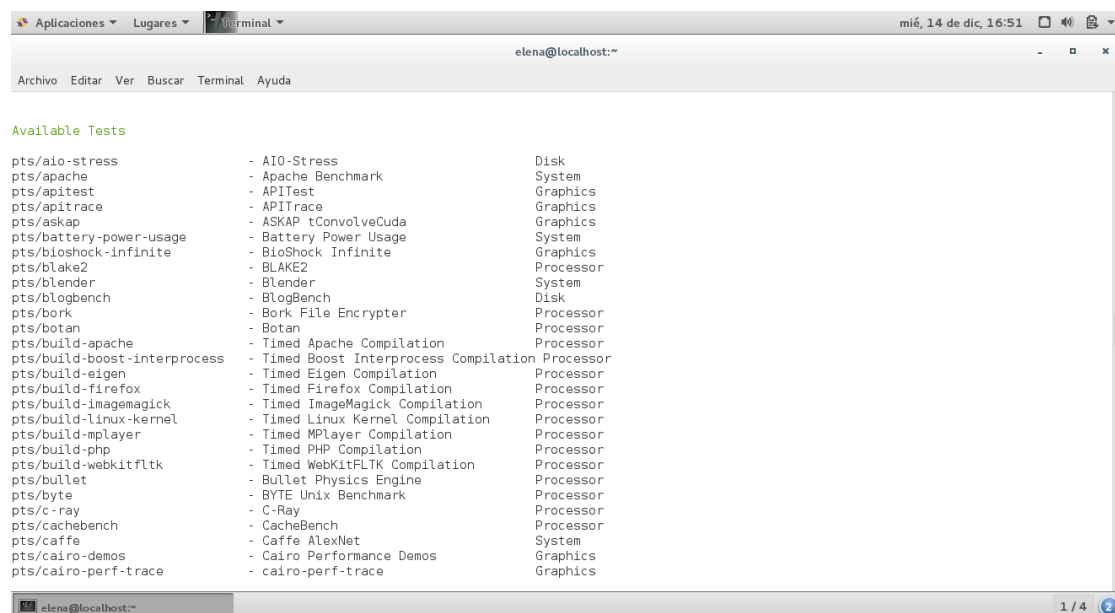
Debido a que mi máquina virtual no es capaz de finalizar el benchmark de Blender, he decidido cancelar la ejecución tras unos 46 minutos (figura 1.10) y probar con otro que requiera menos requisitos y tarde menos en finalizar la ejecución, como por ejemplo “GLmark2” (figura 1.11 y 1.12).

Una vez finalizado el benchmark de “GLmark2”, como se muestra en la figura 1.13, accedemos a los resultados donde podemos ver un listado de las especificaciones de nuestro sistema y una puntuación del test realizado (figuras 1.14 y 1.15).



```
elena@localhost:~$ sudo yum install phoronix-test-suite
[sudo] password for elena:
Complementos cargados:fastestmirror, langpacks
Loading mirror speeds from cached hostfile
* base: ftp.cica.es
* epel: mirror.airenetworks.es
* extras: ftp.cica.es
* updates: ftp.cica.es
Resolviendo dependencias
--> Ejecutando prueba de transacción
--> Paquete phoronix-test-suite.noarch 0:6.6.0-1.el7 debe ser instalado
--> Procesando dependencias: /usr/bin/ruby para el paquete: phoronix-test-suite-6.6.0-1.el7.noarch
--> Ejecutando prueba de transacción
--> Paquete ruby.x86_64 0:2.0.0.648-29.el7 debe ser instalado
--> Procesando dependencias: ruby-libs(x86_64) = 2.0.0.648-29.el7 para el paquete: ruby-2.0.0.648-29.el7.x86_64
--> Procesando dependencias: rubygem(bigdecimal) >= 1.2.0 para el paquete: ruby-2.0.0.648-29.el7.x86_64
--> Procesando dependencias: ruby(rubygems) >= 2.0.14.1 para el paquete: ruby-2.0.0.648-29.el7.x86_64
--> Procesando dependencias: libruby.so.2.0()(64bit) para el paquete: ruby-2.0.0.648-29.el7.x86_64
--> Ejecutando prueba de transacción
--> Paquete ruby-libs.x86_64 0:2.0.0.648-29.el7 debe ser instalado
--> Paquete rubygem-bigdecimal.x86_64 0:1.2.0-29.el7 debe ser instalado
--> Paquete rubygems.noarch 0:2.0.14.1-29.el7 debe ser instalado
--> Procesando dependencias: rubygem(rdoc) >= 4.0.0 para el paquete: rubygems-2.0.14.1-29.el7.noarch
--> Procesando dependencias: rubygem(psych) >= 2.0.0 para el paquete: rubygems-2.0.14.1-29.el7.noarch
--> Procesando dependencias: rubygem(io-console) >= 0.4.2 para el paquete: rubygems-2.0.14.1-29.el7.noarch
--> Ejecutando prueba de transacción
--> Paquete rubygem-io-console.x86_64 0:0.4.2-29.el7 debe ser instalado
--> Paquete rubygem-psych.x86_64 0:2.0.0-29.el7 debe ser instalado
--> Procesando dependencias: libyaml-0.so.2()(64bit) para el paquete: rubygem-psych-2.0.0-29.el7.x86_64
--> Paquete rubygem-rdoc.noarch 0:4.0.0-29.el7 debe ser instalado
--> Procesando dependencias: rubygem(irb) = 2.0.0.648 para el paquete: rubygem-rdoc-4.0.0-29.el7.noarch
--> Procesando dependencias: rubygem(json) >= 1.7.7 para el paquete: rubygem-rdoc-4.0.0-29.el7.noarch
```

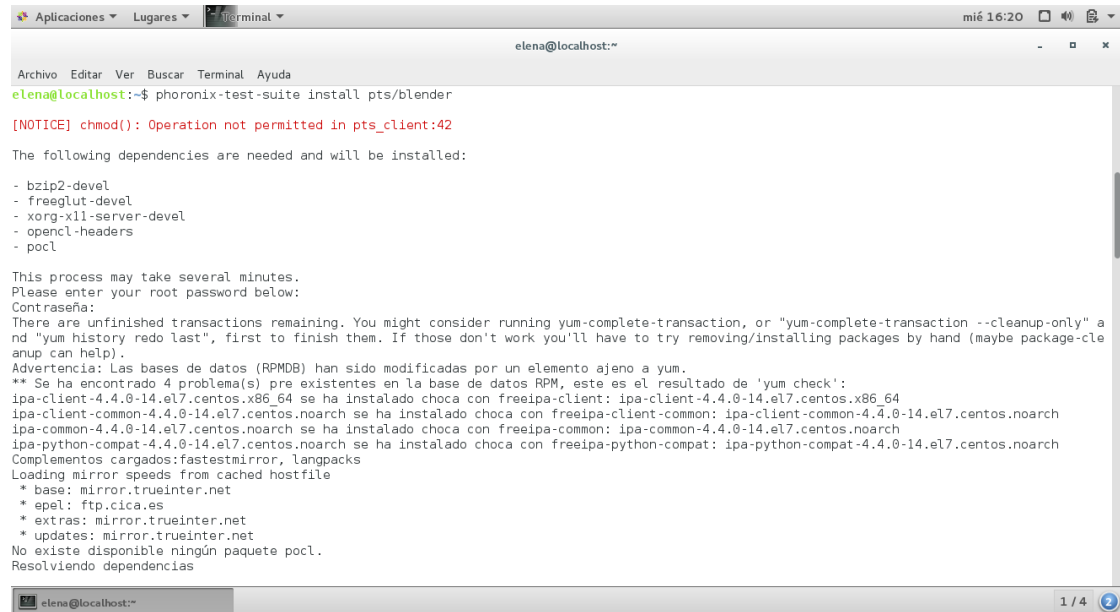
Figura 1.1: CentOS, instalación de Phoronix Suite.



```
Available Tests

pts/aio-stress          - AIO-Stress          Disk
pts/apache             - Apache Benchmark    System
pts/apitest            - APITest             Graphics
pts/apitrace           - APITrace            Graphics
pts/askap              - ASKAP tConvolveCuda Graphics
pts/battery-power-usage - Battery Power Usage System
pts/bioshock-infinite  - BioShock Infinite  Graphics
pts/blake2             - BLAKE2              Processor
pts/blender            - Blender             System
pts/blogbench          - BlogBench           Disk
pts/bork               - Bork File Encrypter Processor
pts/botan              - Botan              Processor
pts/build-apache       - Timed Apache Compilation Processor
pts/build-boost-interprocess - Timed Boost Interprocess Compilation Processor
pts/build-eigen        - Timed Eigen Compilation Processor
pts/build-firefox      - Timed Firefox Compilation Processor
pts/build-imagemagick  - Timed ImageMagick Compilation Processor
pts/build-linux-kernel - Timed Linux Kernel Compilation Processor
pts/build-mplayer      - Timed MPlayer Compilation Processor
pts/build-php          - Timed PHP Compilation Processor
pts/build-webkitgtk    - Timed WebKitGTK Compilation Processor
pts/bullet             - Bullet Physics Engine Processor
pts/byte               - BYTE Unix Benchmark Processor
pts/c-ray              - C-Ray              Processor
pts/cachebench         - CacheBench         Processor
pts/caffe              - Caffe AlexNet      System
pts/cairo-demos        - Cairo Performance Demos Graphics
pts/cairo-perf-trace   - cairo-perf-trace   Graphics
```

Figura 1.2: CentOS, listado de benchmarks.



```
elena@localhost:~$ phoronix-test-suite install pts/blender

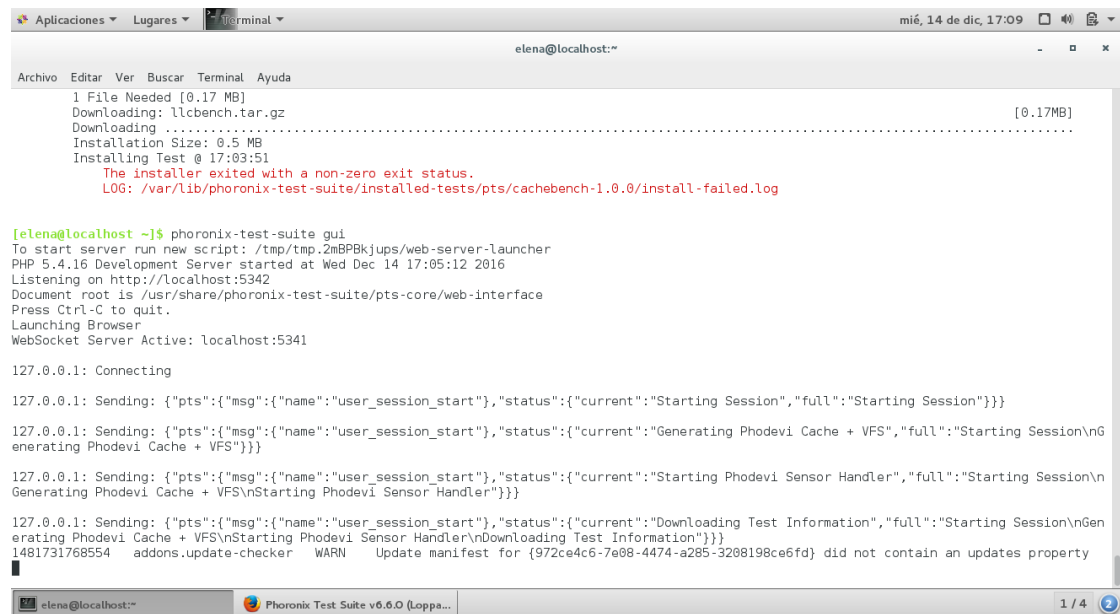
[NOTICE] chmod(): Operation not permitted in pts_client:42

The following dependencies are needed and will be installed:

- bzip2-devel
- freeglut-devel
- xorg-x11-server-devel
- openc1-headers
- poc1

This process may take several minutes.
Please enter your root password below:
Contraseña:
There are unfinished transactions remaining. You might consider running yum-complete-transaction, or "yum-complete-transaction --cleanup-only" and "yum history redo last", first to finish them. If those don't work you'll have to try removing/installing packages by hand (maybe package-clean up can help).
Advertencia: Las bases de datos (RPMDB) han sido modificadas por un elemento ajeno a yum.
** Se ha encontrado 4 problema(s) pre existentes en la base de datos RPM, este es el resultado de 'yum check':
ipa-client-4.4.0-14.el7.centos.x86_64 se ha instalado choca con freeipa-client: ipa-client-4.4.0-14.el7.centos.x86_64
ipa-client-common-4.4.0-14.el7.centos.noarch se ha instalado choca con freeipa-client-common: ipa-client-common-4.4.0-14.el7.centos.noarch
ipa-common-4.4.0-14.el7.centos.noarch se ha instalado choca con freeipa-common: ipa-common-4.4.0-14.el7.centos.noarch
ipa-python-compat-4.4.0-14.el7.centos.noarch se ha instalado choca con freeipa-python-compat: ipa-python-compat-4.4.0-14.el7.centos.noarch
Complementos cargados:fastestmirror, langpacks
Loading mirror speeds from cached hostfile
* base: mirror.trueinter.net
* epel: ftp.cica.es
* extras: mirror.trueinter.net
* updates: mirror.trueinter.net
No existe disponible ningún paquete poc1.
Resolviendo dependencias
```

Figura 1.3: CentOS, instalación del benchmark Blender.



```
1 File Needed [0.17 MB]
Downloading: llcbench.tar.gz [0.17MB]
Downloaded: .....
Installation Size: 0.5 MB
Installing Test @ 17:03:51
The installer exited with a non-zero exit status.
LOG: /var/lib/phoronix-test-suite/installed-tests/pts/cachebench-1.0.0/install-failed.log

[elena@localhost ~]$ phoronix-test-suite gui
To start server run new script: /tmp/tmp.2mBPBkups/web-server-launcher
PHP 5.4.16 Development Server started at Wed Dec 14 17:05:12 2016
Listening on http://localhost:5342
Document root is /usr/share/phoronix-test-suite/pts-core/web-interface
Press Ctrl-C to quit.
Launching Browser
WebSocket Server Active: localhost:5341

127.0.0.1: Connecting

127.0.0.1: Sending: {"pts":{"msg":{"name":"user_session_start"},"status":{"current":"Starting Session","full":"Starting Session"}}}

127.0.0.1: Sending: {"pts":{"msg":{"name":"user_session_start"},"status":{"current":"Generating Phodevi Cache + VFS","full":"Starting Session\nGenerating Phodevi Cache + VFS"}}}

127.0.0.1: Sending: {"pts":{"msg":{"name":"user_session_start"},"status":{"current":"Starting Phodevi Sensor Handler","full":"Starting Session\nGenerating Phodevi Cache + VFS\nStarting Phodevi Sensor Handler"}}}

127.0.0.1: Sending: {"pts":{"msg":{"name":"user_session_start"},"status":{"current":"Downloading Test Information","full":"Starting Session\nGenerating Phodevi Cache + VFS\nStarting Phodevi Sensor Handler\nDownloading Test Information"}}}
1481731768554 addons.update-checker WARN Update manifest for {972ce4c6-7e08-4474-a285-3208198ce6fd} did not contain an updates property
```

Figura 1.4: CentOS, Phoronix Suite GUI.

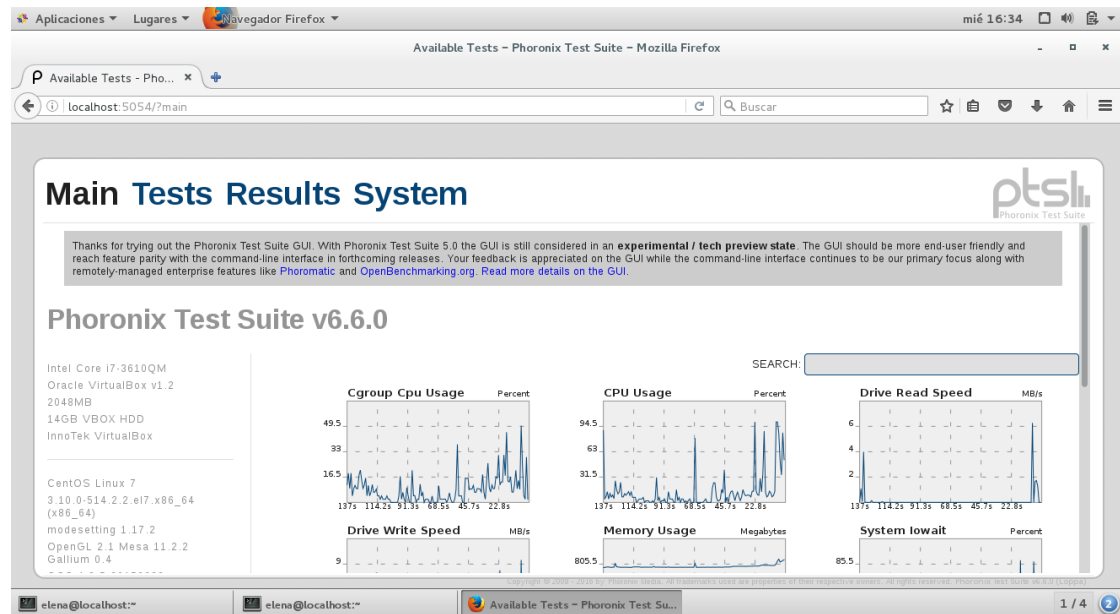


Figura 1.5: CentOS, Phoronix Suite GUI web.

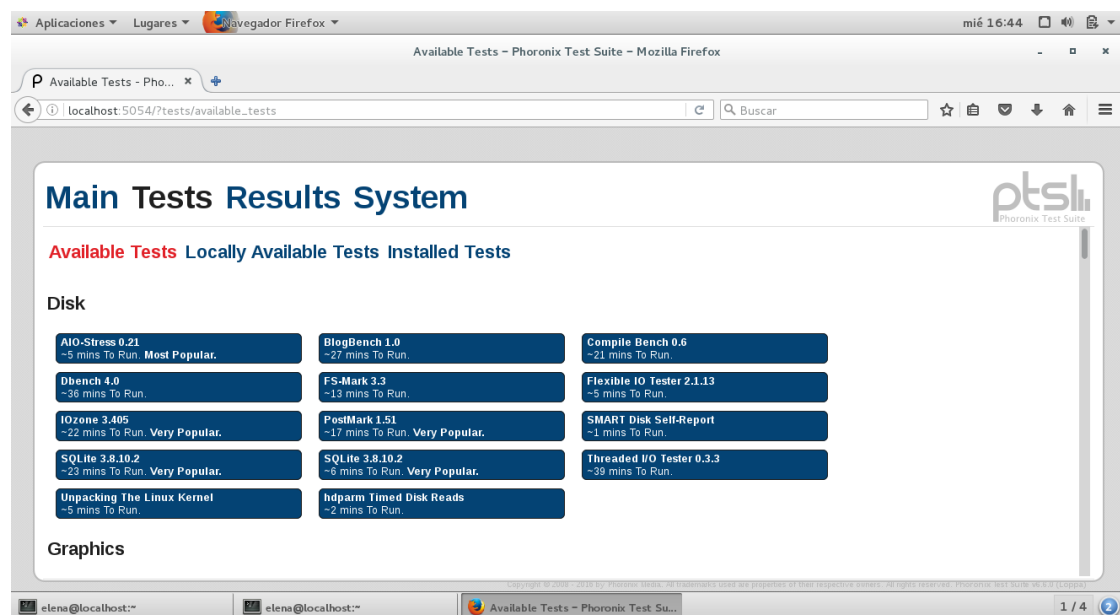


Figura 1.6: CentOS, Phoronix Suite, listado de benchmarks.



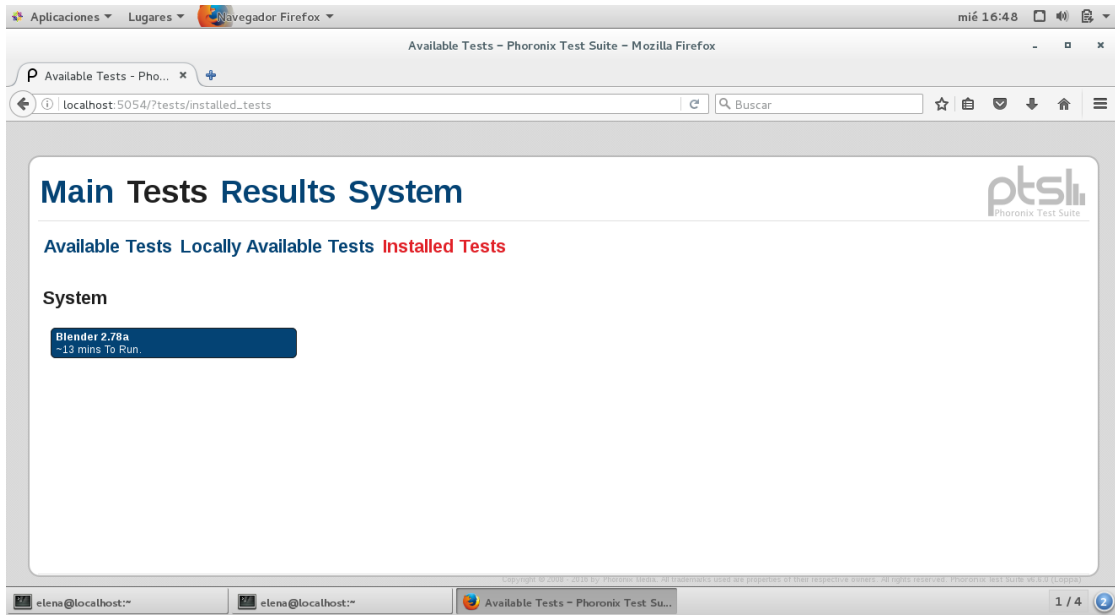


Figura 1.7: CentOS, Phoronix Suite, listado de benchmarks instalados.

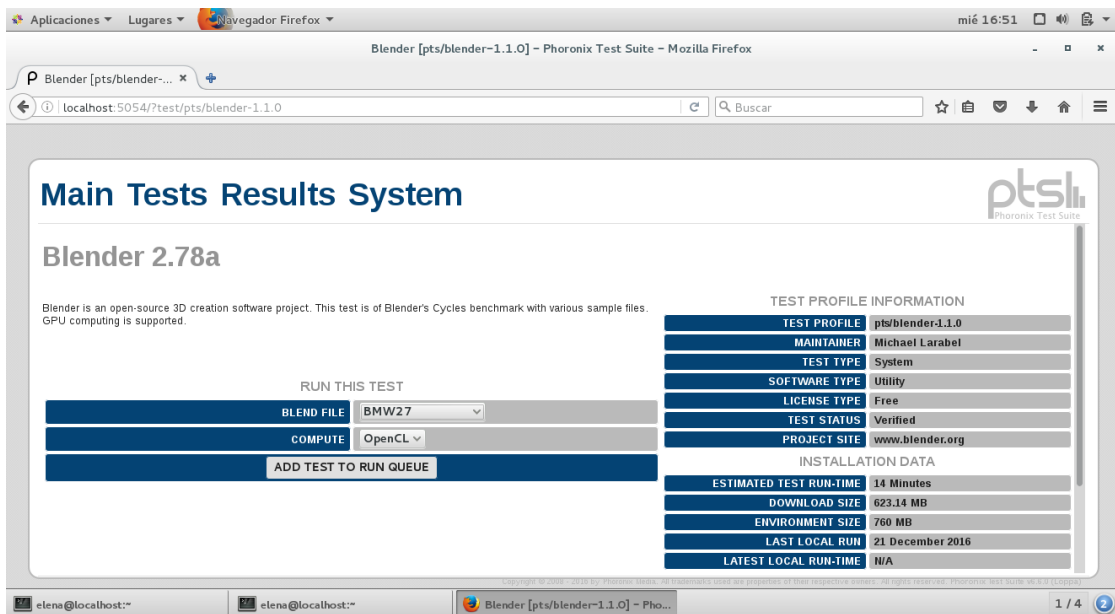


Figura 1.8: CentOS, Phoronix Suite, ejecución de benchmarks.

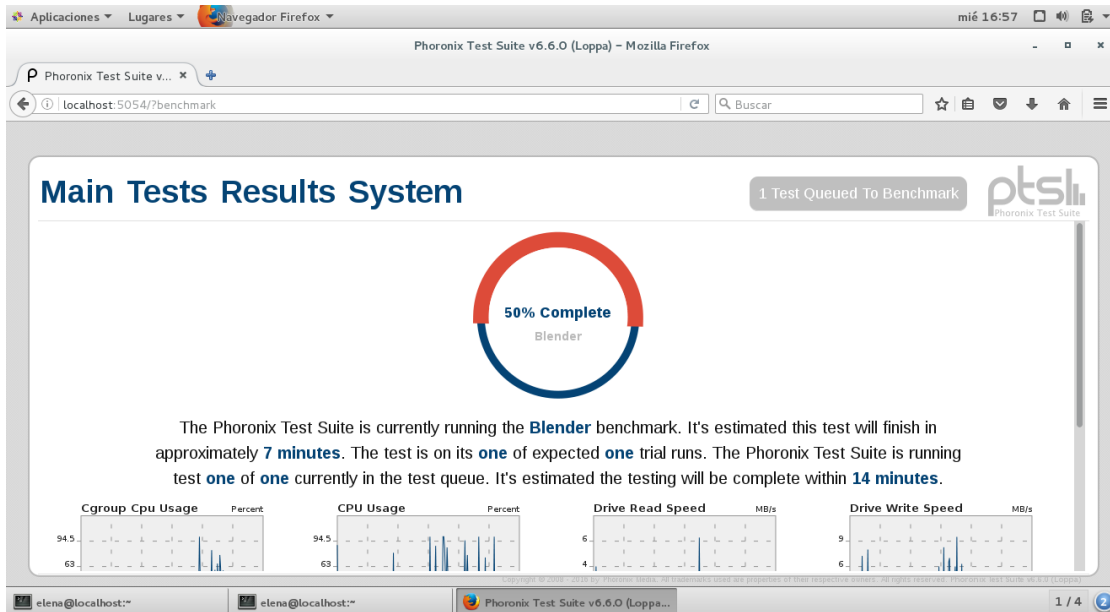


Figura 1.9: CentOS, Phoronix Suite, ejecución de benchmarks.

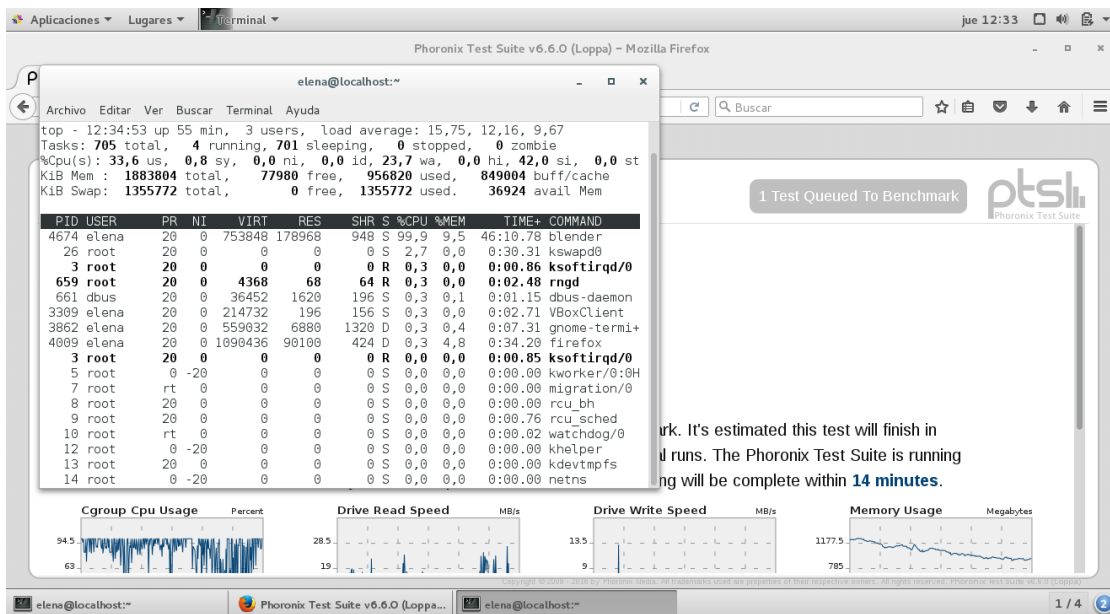


Figura 1.10: CentOS, Phoronix Suite, problemas con benchmark Blender.

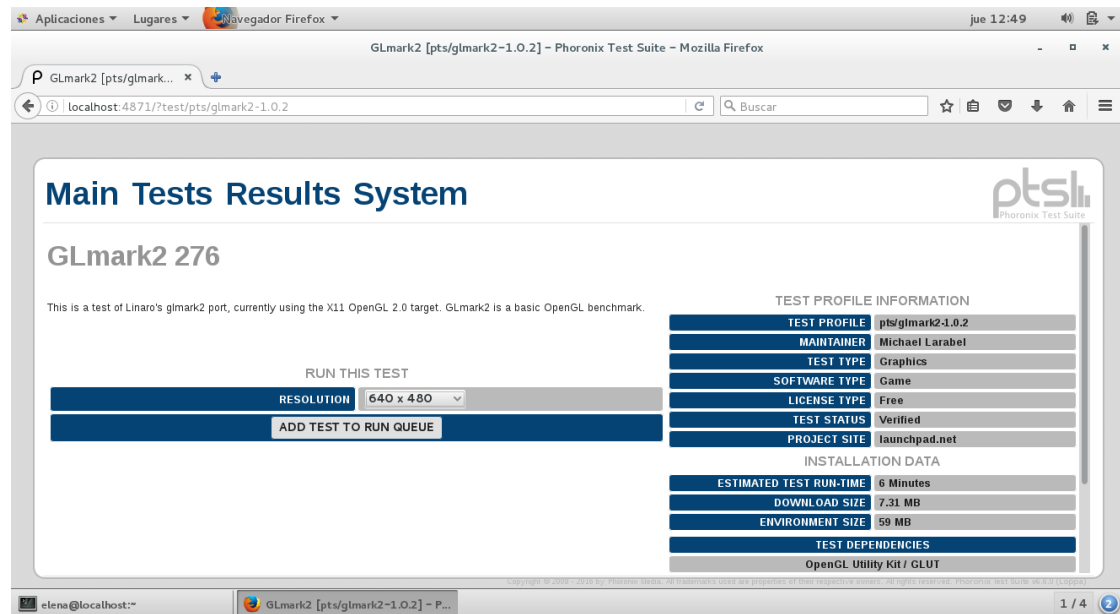


Figura 1.11: CentOS, Phoronix Suite, ejecución de benchmark GLmark2.

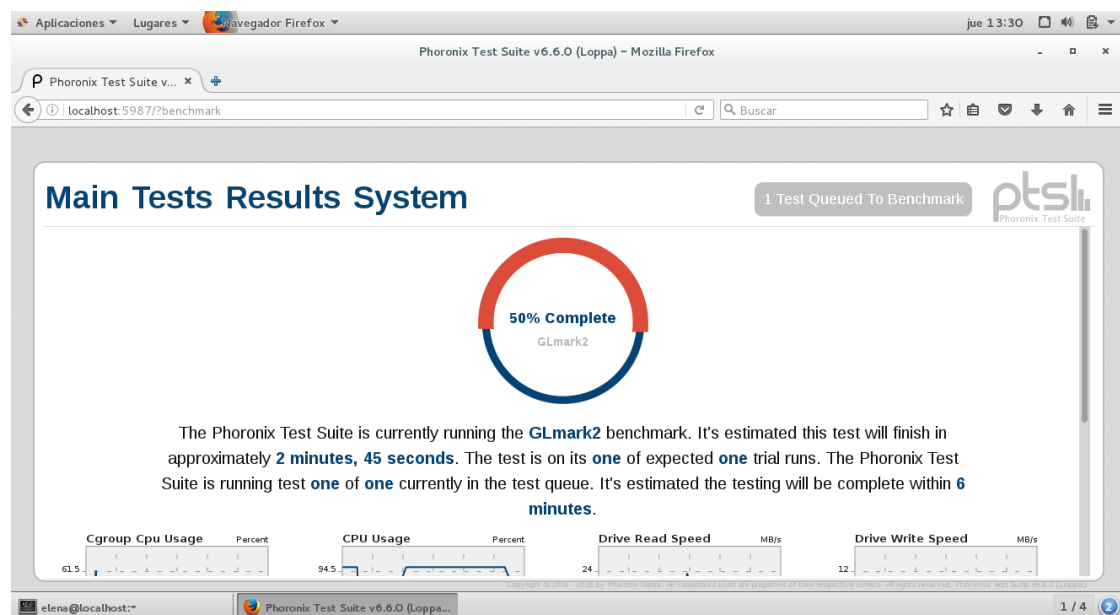


Figura 1.12: CentOS, Phoronix Suite, ejecución de benchmark GLmark2.

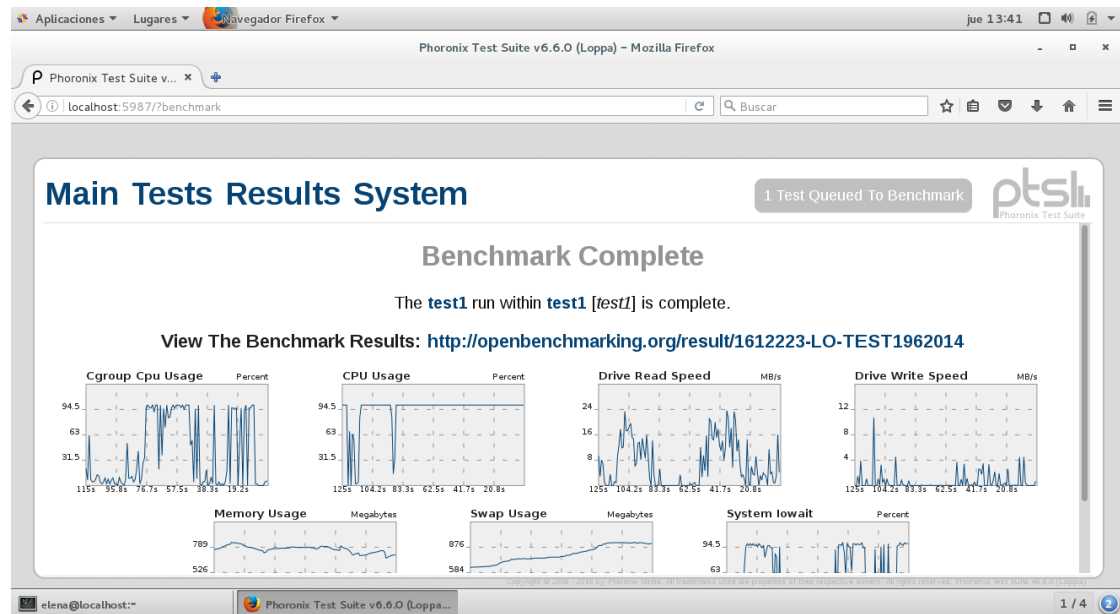


Figura 1.13: CentOS, Phoronix Suite, benchmark GLmark2 finalizado.

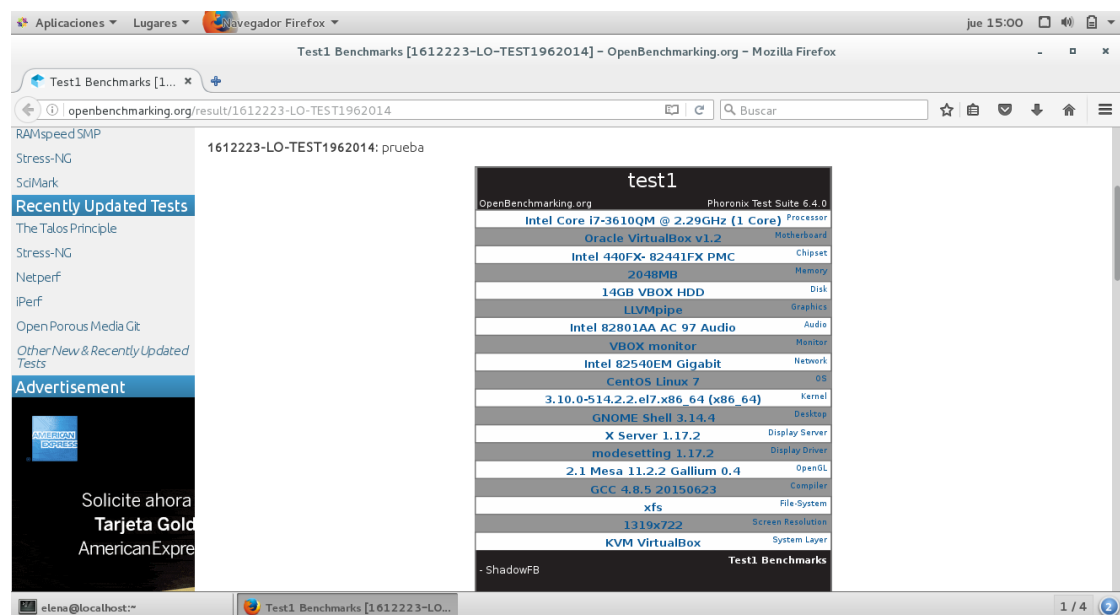


Figura 1.14: CentOS, Phoronix Suite, resultado de benchmark GLmark2.

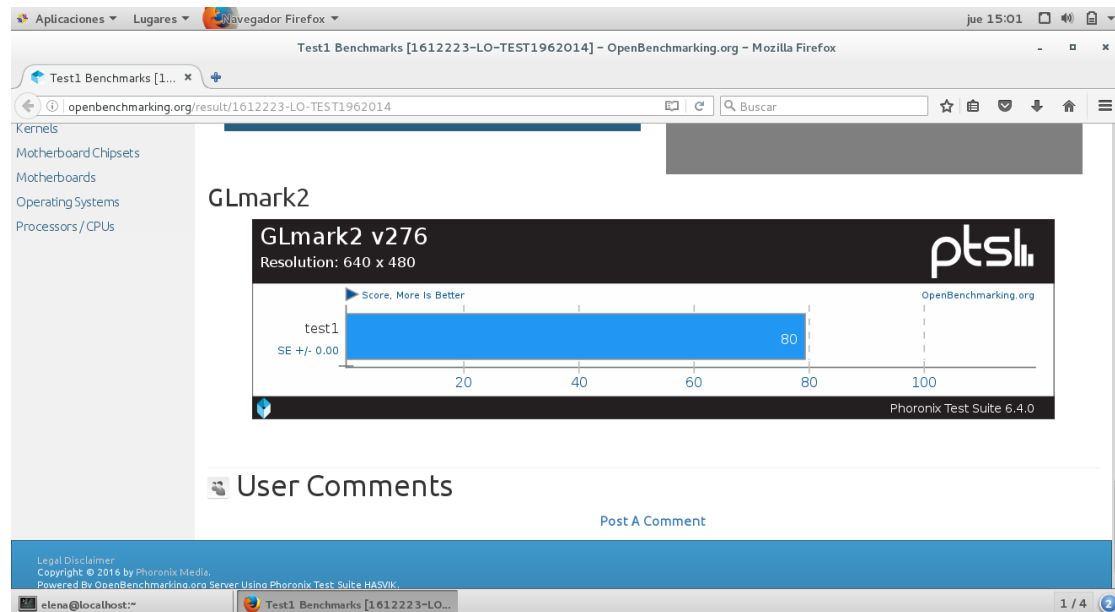


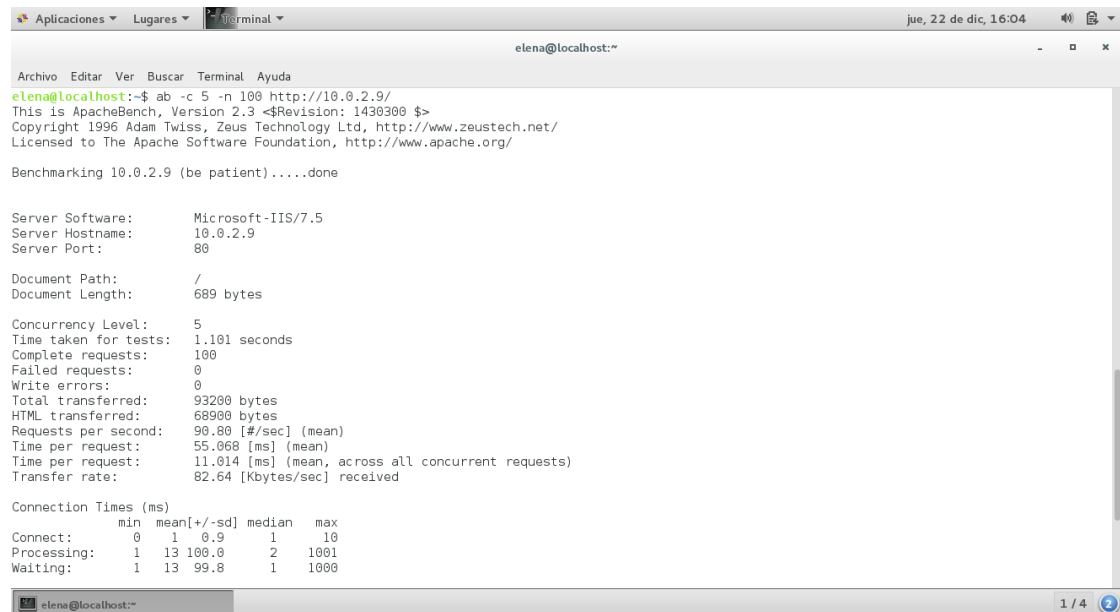
Figura 1.15: CentOS, Phoronix Suite, resultado de benchmark GLmark2.

## 2. Cuestión 2:

### 2.1. De los parámetros que le podemos pasar al comando ¿Qué significa -c 5? ¿y -n 100? Monitoree la ejecución de ab contra alguna máquina (cualquiera) ¿cuántas “tareas” crea ab en el cliente?

Tal y como se indica en el man de `ab` la opción `-c 5` indica que se podrán ejecutar concurrentemente 5 solicitudes y la opción `-n 100` significa que se harán 100 peticiones en el benchmarking actual.

He monitorizado la ejecución de `ab` desde una máquina virtual con CentOS hacia otra con Windows Server, el resultado de la ejecución se muestra en las figuras 2.1 y 2.2. El comando `ab` crea 1 tarea en el cliente como se muestra en la figura 2.3. En el servidor se puede ver un claro aumento del uso de la CPU durante la ejecución de `ab` (figura 2.4).



A terminal window titled 'Aplicaciones Lugares terminal' with a status bar showing 'jue, 22 de dic, 16:04'. The prompt is 'elena@localhost:~'. The user has run the command 'ab -c 5 -n 100 http://10.0.2.9/'. The output shows ApacheBench version 2.3 results for a Microsoft-IIS/7.5 server. The benchmarking process is complete, showing a concurrency level of 5 and 100 requests. The total transferred is 93200 bytes, with a requests per second of 90.80. The time per request is 11.014 ms, and the transfer rate is 82.64 Kbytes/sec. Connection times are also displayed in a table format.

```
Archivo Editar Ver Buscar Terminal Ayuda
elena@localhost:~$ ab -c 5 -n 100 http://10.0.2.9/
This is ApacheBench, Version 2.3 <$Revision: 1430300 $>
Copyright 1996 Adam Twiss, Zeus Technology Ltd, http://www.zeustech.net/
Licensed to The Apache Software Foundation, http://www.apache.org/

Benchmarking 10.0.2.9 (be patient).....done

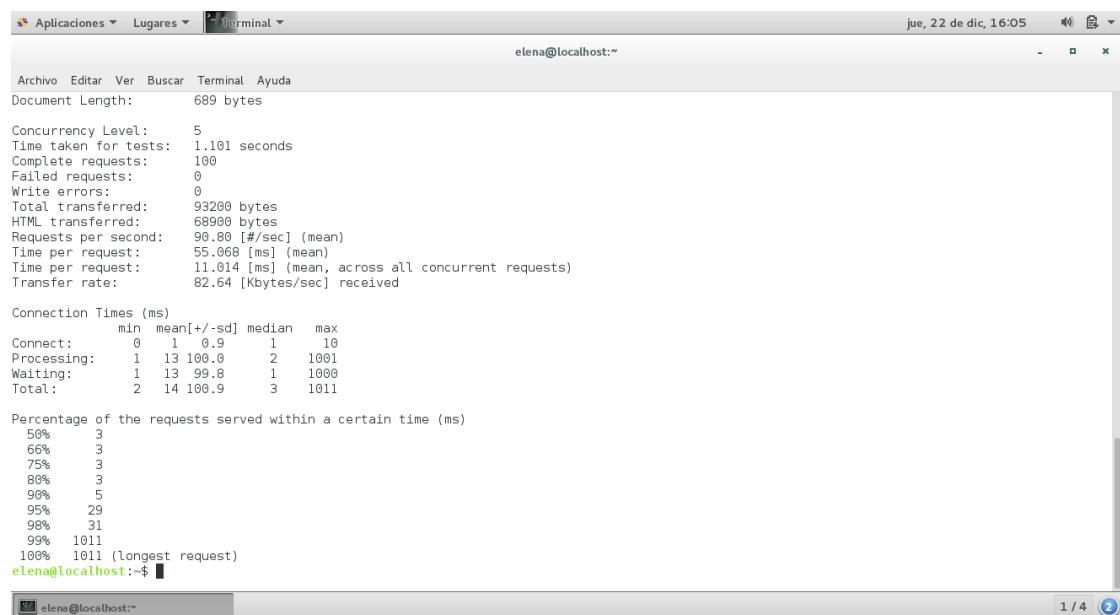

Server Software:      Microsoft-IIS/7.5
Server Hostname:      10.0.2.9
Server Port:          80

Document Path:        /
Document Length:      689 bytes

Concurrency Level:     5
Time taken for tests:  1.101 seconds
Complete requests:     100
Failed requests:        0
Write errors:           0
Total transferred:     93200 bytes
HTML transferred:      68900 bytes
Requests per second:   90.80 [#/sec] (mean)
Time per request:      55.068 [ms] (mean)
Time per request:      11.014 [ms] (mean, across all concurrent requests)
Transfer rate:         82.64 [Kbytes/sec] received

Connection Times (ms)
              min      mean[+/-sd] median   max
Connect:        0        1   0.9      1      10
Processing:      1       13 100.0      2     1001
Waiting:         1       13  99.8      1     1000
Total:           2       13 100.8      3     1011
```

Figura 2.1: CentOS, ab contra Windows Server.



A terminal window titled 'Aplicaciones Lugares terminal' with a status bar showing 'jue, 22 de dic, 16:05'. The prompt is 'elena@localhost:~'. The user has run the command 'ab -c 5 -n 100 http://10.0.2.9/'. The output shows ApacheBench version 2.3 results for a Microsoft-IIS/7.5 server. The benchmarking process is complete, showing a concurrency level of 5 and 100 requests. The total transferred is 93200 bytes, with a requests per second of 90.80. The time per request is 11.014 ms, and the transfer rate is 82.64 Kbytes/sec. Connection times are also displayed in a table format. Additionally, the percentage of requests served within a certain time is shown.

```
Archivo Editar Ver Buscar Terminal Ayuda
elena@localhost:~$ ab -c 5 -n 100 http://10.0.2.9/
This is ApacheBench, Version 2.3 <$Revision: 1430300 $>
Copyright 1996 Adam Twiss, Zeus Technology Ltd, http://www.zeustech.net/
Licensed to The Apache Software Foundation, http://www.apache.org/

Benchmarking 10.0.2.9 (be patient).....done


Server Software:      Microsoft-IIS/7.5
Server Hostname:      10.0.2.9
Server Port:          80

Document Length:      689 bytes

Concurrency Level:     5
Time taken for tests:  1.101 seconds
Complete requests:     100
Failed requests:        0
Write errors:           0
Total transferred:     93200 bytes
HTML transferred:      68900 bytes
Requests per second:   90.80 [#/sec] (mean)
Time per request:      55.068 [ms] (mean)
Time per request:      11.014 [ms] (mean, across all concurrent requests)
Transfer rate:         82.64 [Kbytes/sec] received

Connection Times (ms)
              min      mean[+/-sd] median   max
Connect:        0        1   0.9      1      10
Processing:      1       13 100.0      2     1001
Waiting:         1       13  99.8      1     1000
Total:           2       14 100.9      3     1011

Percentage of the requests served within a certain time (ms)
 50%    3
 66%    3
 75%    3
 80%    3
 90%    5
 95%   29
 98%   31
 99%  1011
100%  1011 (longest request)
elena@localhost:~$
```

Figura 2.2: CentOS, ab contra Windows Server.

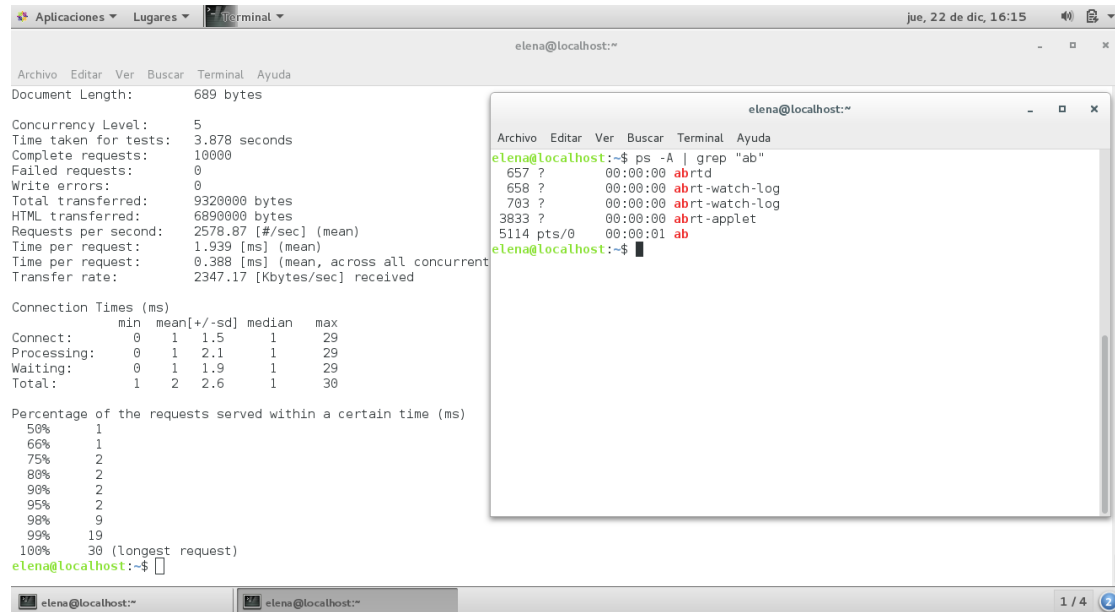


Figura 2.3: CentOS, número de tareas de ab.

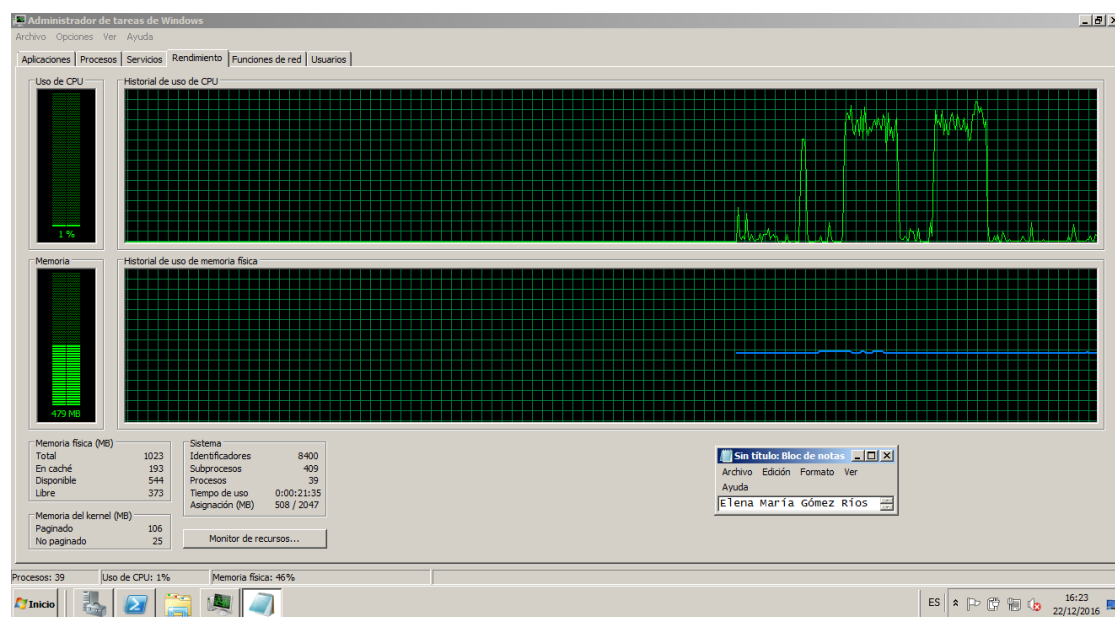


Figura 2.4: Windows, Adminitrador de tareas durante ejecución ab desde CentOS.

### 3. Cuestión 3:

#### 3.1. Ejecute **ab** contra a las tres máquinas virtuales (desde el SO anfitrión a las máquinas virtuales de la red local) una a una (arrancadas por separado). ¿Cuál es la que proporciona mejores resultados? Muestre y coméntelos. (Use como máquina de referencia Ubuntu Server para la comparativa).

En primer lugar comentar que este ejercicio lo voy a realizar desde una máquina virtual CentOS en vez de utilizar el SO anfitrión ya que tengo un problema con los puertos, como ya dije en prácticas anteriores, y me es imposible realizar la redirección de puertos de VirtualBox.

En primer lugar voy a ejecutar **ab** contra Windows, los resultados se muestran en las figuras 3.1 y 3.2. Para hacerlo contra CentOS he tenido que desactivar el firewall como se muestra en la figura 3.3 ya que por defecto el firewall bloquea todo el tráfico impidiendo ejecutar **ab**. Los resultados de **ab** contra CentOS se muestran en las figuras 3.4 y 3.5. Por último he realizado el test **ab** contra Ubuntu Server como se muestra en las figuras 3.6 y 3.7. Igualmente en Ubuntu he tenido que deshabilitar el firewall con el comando `sudo ufw disable`.

Para una fácil comparación entre los resultados de **ab** contra las diferentes máquinas virtuales voy a mostrar los resultados en la tabla 3.1.

Tabla 3.1: Resultados de **ab**.

SO	Tamaño web	Tiempo (s)	Datos Transferidos	Respuestas/s	Ratio Transferencia
CentOS	4897 bytes	64.347	493.90 MB	1554.08	7859.94 KB/s
Ubuntu	11510 bytes	54.763	1123.71 MB	1826.06	21012.13 KB/s
Windows	689 bytes	31.895	88.882 MB	3135.33	2853.64 KB/s

Como se puede observar mirando la columna correspondiente al ratio de transferencia, la máquina virtual que proporciona mejores resultados es Ubuntu, seguida de CentOS y por último Windows. Ubuntu tiene un ratio de transferencia 7.3632 veces mejor que Windows, y 2.67 veces mejor que CentOS.



```
Aplicaciones Lugares terminal jue, 22 de dic, 16:32
elena@localhost:~$ ab -c 5 -n 100000 http://10.0.2.9/
This is ApacheBench, Version 2.3 <$Revision: 1430300 $>
Copyright 1996 Adam Twiss, Zeus Technology Ltd, http://www.zeustech.net/
Licensed to The Apache Software Foundation, http://www.apache.org/

Benchmarking 10.0.2.9 (be patient)
Completed 10000 requests
Completed 20000 requests
Completed 30000 requests
Completed 40000 requests
Completed 50000 requests
Completed 60000 requests
Completed 70000 requests
Completed 80000 requests
Completed 90000 requests
Completed 100000 requests
Finished 100000 requests

Server Software:      Microsoft-IIS/7.5
Server Hostname:      10.0.2.9
Server Port:          80

Document Path:        /
Document Length:       689 bytes

Concurrency Level:     5
Time taken for tests:  31.895 seconds
Complete requests:     100000
Failed requests:        0
Write errors:           0
Total transferred:     93200000 bytes
```

Figura 3.1: CentOS, resultado ab contra Windows.

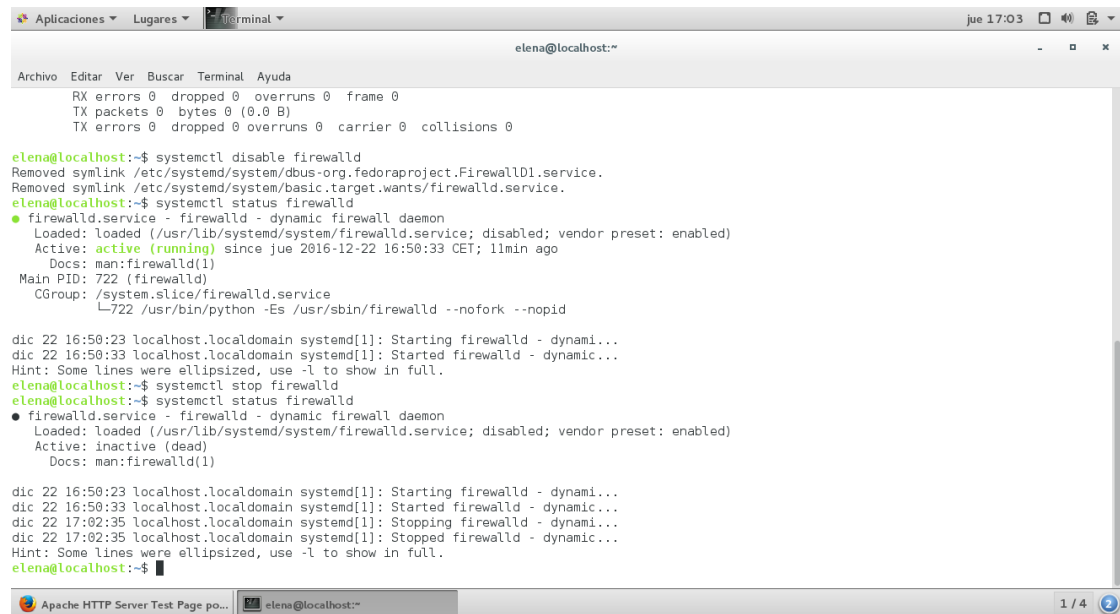
```
Aplicaciones Lugares terminal jue, 22 de dic, 16:32
elena@localhost:~$ ab -c 5 -n 100000 http://10.0.2.9/
Document Length:       689 bytes

Concurrency Level:     5
Time taken for tests:  31.895 seconds
Complete requests:     100000
Failed requests:        0
Write errors:           0
Total transferred:     93200000 bytes
HTML transferred:      68900000 bytes
Requests per second:   3135.33 [#/sec] (mean)
Time per request:      1.595 [ms] (mean)
Time per request:      0.319 [ms] (mean, across all concurrent requests)
Transfer rate:         2853.64 [Kbytes/sec] received

Connection Times (ms)
  min  mean[+/-sd] median  max
Connect:    0    1  0.7      1    33
Processing:  0    1  0.9      1    29
Waiting:    0    1  0.9      1    29
Total:      1    1  1.2      1    47

Percentage of the requests served within a certain time (ms)
 50%    1
 66%    1
 75%    1
 80%    2
 90%    2
 95%    2
 98%    2
 99%    2
100%   47 (longest request)
elena@localhost:~$
```

Figura 3.2: CentOS, resultado ab contra Windows.

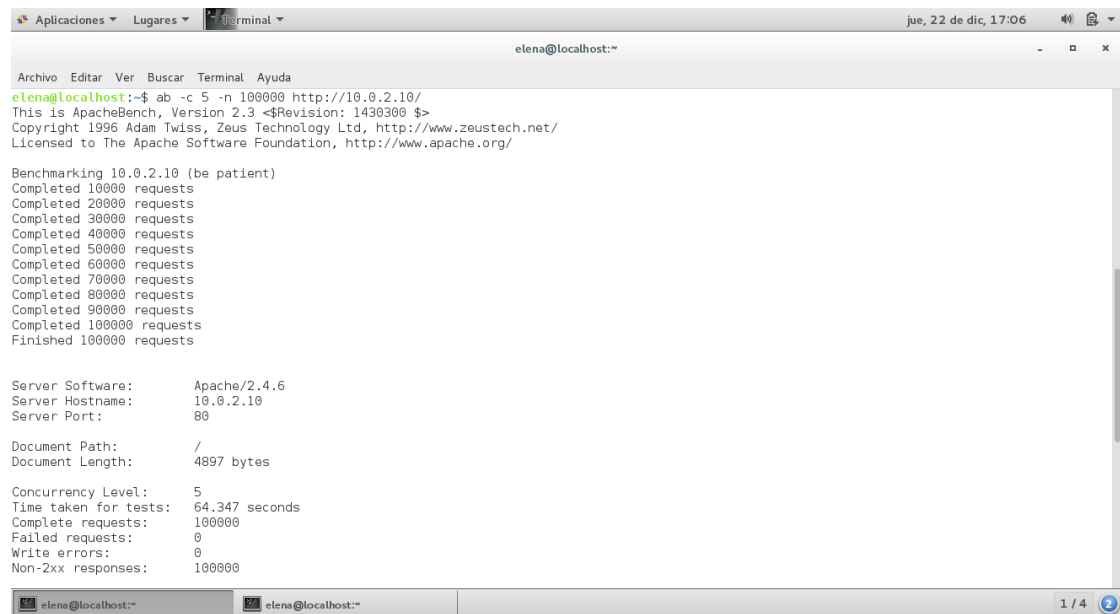


```
elena@localhost:~$ systemctl disable firewalld
Removed symlink /etc/systemd/system/dbus-org.fedoraproject.FirewallD1.service.
Removed symlink /etc/systemd/system/basic.target.wants/firewalld.service.
elena@localhost:~$ systemctl status firewalld
● firewalld.service - firewalld - dynamic firewall daemon
   Loaded: loaded (/usr/lib/systemd/system/firewalld.service; disabled; vendor preset: enabled)
   Active: active (running) since jue 2016-12-22 16:50:33 CET; 11min ago
     Docs: man:firewalld(1)
   Main PID: 722 (firewalld)
   CGroup: /system.slice/firewalld.service
           └─722 /usr/bin/python -Es /usr/sbin/firewalld --nofork --nopid

dic 22 16:50:23 localhost.localdomain systemd[1]: Starting firewalld - dynami...
dic 22 16:50:33 localhost.localdomain systemd[1]: Started firewalld - dynamic...
Hint: Some lines were ellipsized, use -l to show in full.
elena@localhost:~$ systemctl stop firewalld
elena@localhost:~$ systemctl status firewalld
● firewalld.service - firewalld - dynamic firewall daemon
   Loaded: loaded (/usr/lib/systemd/system/firewalld.service; disabled; vendor preset: enabled)
   Active: inactive (dead)
     Docs: man:firewalld(1)

dic 22 16:50:23 localhost.localdomain systemd[1]: Starting firewalld - dynami...
dic 22 16:50:33 localhost.localdomain systemd[1]: Started firewalld - dynamic...
dic 22 17:02:35 localhost.localdomain systemd[1]: Stopping firewalld - dynami...
dic 22 17:02:35 localhost.localdomain systemd[1]: Stopped firewalld - dynamic...
Hint: Some lines were ellipsized, use -l to show in full.
elena@localhost:~$
```

Figura 3.3: CentOS, desactivación del firewall.



```
elena@localhost:~$ ab -c 5 -n 100000 http://10.0.2.10/
This is ApacheBench, Version 2.3 <$Revision: 1430300 $>
Copyright 1996 Adam Twiss, Zeus Technology Ltd, http://www.zeustech.net/
Licensed to The Apache Software Foundation, http://www.apache.org/

Benchmarking 10.0.2.10 (be patient)
Completed 10000 requests
Completed 20000 requests
Completed 30000 requests
Completed 40000 requests
Completed 50000 requests
Completed 60000 requests
Completed 70000 requests
Completed 80000 requests
Completed 90000 requests
Completed 100000 requests
Finished 100000 requests


Server Software:      Apache/2.4.6
Server Hostname:      10.0.2.10
Server Port:          80

Document Path:        /
Document Length:      4897 bytes

Concurrency Level:    5
Time taken for tests:  64.347 seconds
Complete requests:    100000
Failed requests:       0
Write errors:         0
Non-2xx responses:    100000
```

Figura 3.4: CentOS, resultado ab contra CentOS.

The screenshot shows a terminal window titled 'terminal' with the user 'elena@localhost'. The terminal displays the output of an ApacheBench test. The results show a concurrency level of 5, a total time of 64.347 seconds, and 100,000 complete requests. The transfer rate is 7859.94 Kbytes/sec. Connection times are also listed, with a total of 1.3 ms. The percentage of requests served within a certain time is shown as 100% for 474 ms (longest request).

```
Concurency Level:      5
Time taken for tests:  64.347 seconds
Complete requests:    100000
Failed requests:      0
Write errors:         0
Non-2xx responses:    100000
Total transferred:    517900000 bytes
HTML transferred:     489700000 bytes
Requests per second:  1554.08 [#/sec] (mean)
Time per request:     3.217 [ms] (mean)
Time per request:     0.643 [ms] (mean, across all concurrent requests)
Transfer rate:        7859.94 [Kbytes/sec] received

Connection Times (ms)
      min      mean[+/-sd] median    max
Connect:    0      0   0.3      0      29
Processing:  1      3   1.9      3     474
Waiting:    0      3   1.8      3     474
Total:      1      3   2.0      3     474

Percentage of the requests served within a certain time (ms)
 50%      3
 66%      3
 75%      3
 80%      3
 90%      3
 95%      3
 98%      4
 99%      8
100%     474 (longest request)
elena@localhost:~$
```

Figura 3.5: CentOS, resultado ab contra CentOS.

The screenshot shows a terminal window titled 'terminal' with the user 'elena@localhost'. The terminal displays the output of an ApacheBench test. The results show a concurrency level of 5, a total time of 54.763 seconds, and 100,000 complete requests. The transfer rate is 1178300000 bytes. The server software is Apache/2.4.7, and the server hostname is 10.0.2.11. The document path is / and the document length is 11510 bytes.

```
Benchmarking 10.0.2.11 (be patient)
Completed 10000 requests
Completed 20000 requests
Completed 30000 requests
Completed 40000 requests
Completed 50000 requests
Completed 60000 requests
Completed 70000 requests
Completed 80000 requests
Completed 90000 requests
Completed 100000 requests
Finished 100000 requests

Server Software:      Apache/2.4.7
Server Hostname:      10.0.2.11
Server Port:         80

Document Path:        /
Document Length:      11510 bytes

Concurency Level:      5
Time taken for tests:  54.763 seconds
Complete requests:    100000
Failed requests:      0
Write errors:         0
Total transferred:    1178300000 bytes
```

Figura 3.6: CentOS, resultado ab contra Ubuntu Server.

```
elena@localhost:~$
Document Length: 11510 bytes

Concurrency Level: 5
Time taken for tests: 54.763 seconds
Complete requests: 100000
Failed requests: 0
Write errors: 0
Total transferred: 1178300000 bytes
HTML transferred: 1151000000 bytes
Requests per second: 1826.06 [#/sec] (mean)
Time per request: 2.738 [ms] (mean)
Time per request: 0.548 [ms] (mean, across all concurrent requests)
Transfer rate: 21012.13 [Kbytes/sec] received

Connection Times (ms)
      min      mean[+/-sd] median    max
Connect:    0       0   0.5      0      25
Processing:  0       2   3.9      1     188
Waiting:    0       1   1.8      1      49
Total:      1       3   4.0      2     188

Percentage of the requests served within a certain time (ms)
 50%    2
 66%    2
 75%    2
 80%    2
 90%    3
 95%    7
 98%   14
 99%   19
100%  188 (longest request)
elena@localhost:~$
```

Figura 3.7: CentOS, resultado ab contra Ubuntu Server.

## 4. Cuestión opcional 1:

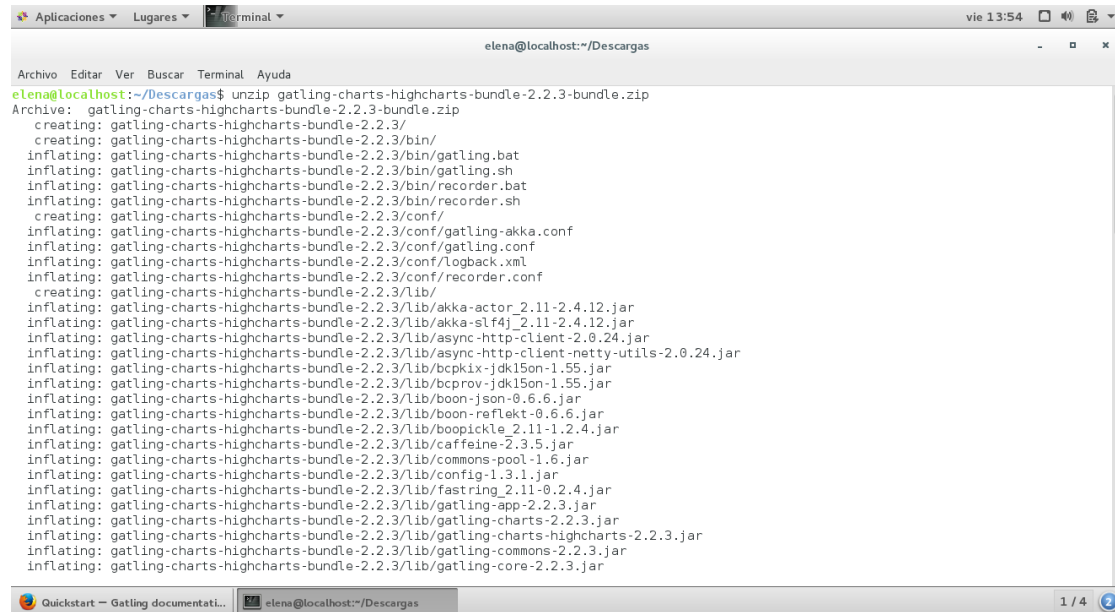
### 4.1. ¿Qué es Scala? Instale Gatling y pruebe los escenarios por defecto.

Tal y como se dice en la página oficial de Scala [5], Scala (“Scalable Language”) es un lenguaje de programación orientado a objetos y cuenta con todas sus características principales. Utiliza una sintaxis concisa, elegante e incluso permite agregar nuevas operaciones a las clases existentes, no importa si vienen de Scala o Java.

Para instalar Gatling he seguido el tutorial de su página oficial [4]. Primero debemos descargarnos el paquete de Gatling que proporciona la página web y lo descomprimos (figura 4.1). Para ejecutar Gatling debemos ejecutar el archivo, `gatling.sh`, que se encuentra en la carpeta `/bin` de este paquete (figura 4.2).

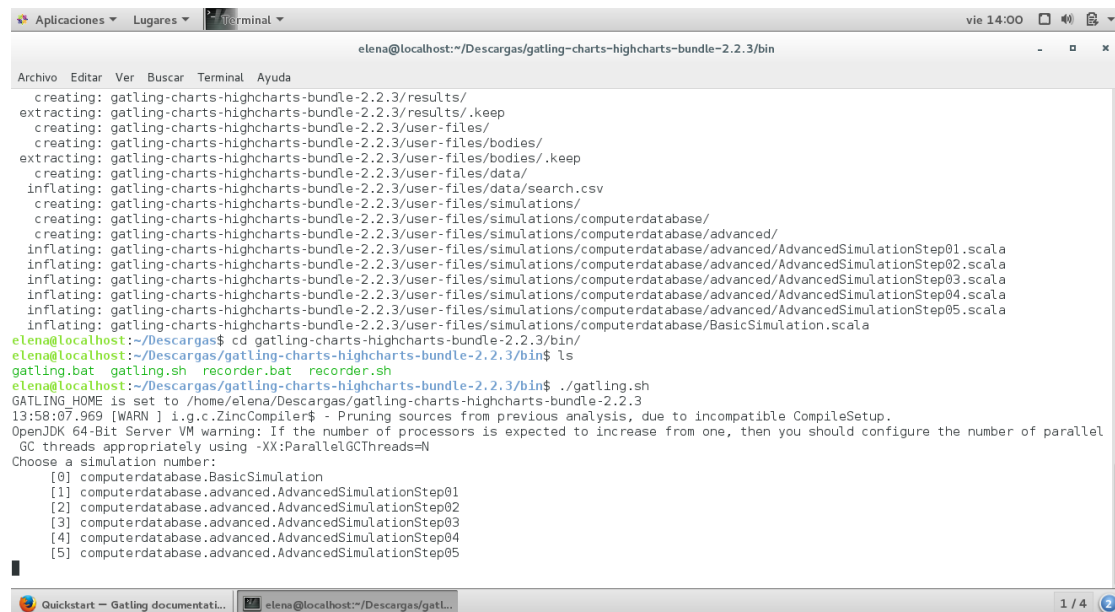
Nos da la opción de elegir entre 6 simulaciones, por ejemplo seleccionamos la opción 0 “computerdatabase.BasicSimulation”, la cual nos muestra la información que aparece en la figura 4.3, y podemos verlos más cómodamente en la web de resultados que crea Gatling, como se muestra en la figura 4.4.

También podemos seleccionar alguno de los otros escenarios por defecto como “computerdatabase.advanced.AdvancedSimulationStep01”, el cual nos devuelve los datos de las figuras 4.5 y 4.6. O el escenario “computerdatabase.advanced.AdvancedSimulationStep05”, el cual nos devuelve los datos de las figuras 4.7 y 4.8.



```
elena@localhost:~/Descargas
Archivo Editar Ver Buscar Terminal Ayuda
elena@localhost:~/Descargas$ unzip gatling-charts-highcharts-bundle-2.2.3-bundle.zip
Archive:  gatling-charts-highcharts-bundle-2.2.3-bundle.zip
  creating:  gatling-charts-highcharts-bundle-2.2.3/
  creating:  gatling-charts-highcharts-bundle-2.2.3/bin/
  inflating:  gatling-charts-highcharts-bundle-2.2.3/bin/gatling.bat
  inflating:  gatling-charts-highcharts-bundle-2.2.3/bin/gatling.sh
  inflating:  gatling-charts-highcharts-bundle-2.2.3/bin/recorder.bat
  inflating:  gatling-charts-highcharts-bundle-2.2.3/bin/recorder.sh
  creating:  gatling-charts-highcharts-bundle-2.2.3/conf/
  inflating:  gatling-charts-highcharts-bundle-2.2.3/conf/gatling-akka.conf
  inflating:  gatling-charts-highcharts-bundle-2.2.3/conf/gatling.conf
  inflating:  gatling-charts-highcharts-bundle-2.2.3/conf/logback.xml
  inflating:  gatling-charts-highcharts-bundle-2.2.3/conf/recorder.conf
  creating:  gatling-charts-highcharts-bundle-2.2.3/lib/
  inflating:  gatling-charts-highcharts-bundle-2.2.3/lib/akka-actor_2.11-2.4.12.jar
  inflating:  gatling-charts-highcharts-bundle-2.2.3/lib/akka-slf4j_2.11-2.4.12.jar
  inflating:  gatling-charts-highcharts-bundle-2.2.3/lib/async-http-client-2.0.24.jar
  inflating:  gatling-charts-highcharts-bundle-2.2.3/lib/async-http-client-netty-utils-2.0.24.jar
  inflating:  gatling-charts-highcharts-bundle-2.2.3/lib/bcpkix-jdk15on-1.55.jar
  inflating:  gatling-charts-highcharts-bundle-2.2.3/lib/bcprov-jdk15on-1.55.jar
  inflating:  gatling-charts-highcharts-bundle-2.2.3/lib/boon-json-0.6.6.jar
  inflating:  gatling-charts-highcharts-bundle-2.2.3/lib/boon-reflekt-0.6.6.jar
  inflating:  gatling-charts-highcharts-bundle-2.2.3/lib/boopickle_2.11-1.2.4.jar
  inflating:  gatling-charts-highcharts-bundle-2.2.3/lib/caffeine-2.3.5.jar
  inflating:  gatling-charts-highcharts-bundle-2.2.3/lib/commons-pool-1.6.jar
  inflating:  gatling-charts-highcharts-bundle-2.2.3/lib/config-1.3.1.jar
  inflating:  gatling-charts-highcharts-bundle-2.2.3/lib/fastring_2.11-0.2.4.jar
  inflating:  gatling-charts-highcharts-bundle-2.2.3/lib/gatling-app-2.2.3.jar
  inflating:  gatling-charts-highcharts-bundle-2.2.3/lib/gatling-charts-2.2.3.jar
  inflating:  gatling-charts-highcharts-bundle-2.2.3/lib/gatling-charts-highcharts-2.2.3.jar
  inflating:  gatling-charts-highcharts-bundle-2.2.3/lib/gatling-commons-2.2.3.jar
  inflating:  gatling-charts-highcharts-bundle-2.2.3/lib/gatling-core-2.2.3.jar
```

Figura 4.1: CentOS, instalación de Gatling.



```
elena@localhost:~/Descargas/gatling-charts-highcharts-bundle-2.2.3/bin
Archivo Editar Ver Buscar Terminal Ayuda
  creating:  gatling-charts-highcharts-bundle-2.2.3/results/
  extracting:  gatling-charts-highcharts-bundle-2.2.3/results/.keep
  creating:  gatling-charts-highcharts-bundle-2.2.3/user-files/
  creating:  gatling-charts-highcharts-bundle-2.2.3/user-files/bodies/
  extracting:  gatling-charts-highcharts-bundle-2.2.3/user-files/bodies/.keep
  creating:  gatling-charts-highcharts-bundle-2.2.3/user-files/data/
  inflating:  gatling-charts-highcharts-bundle-2.2.3/user-files/data/search.csv
  creating:  gatling-charts-highcharts-bundle-2.2.3/user-files/simulations/
  creating:  gatling-charts-highcharts-bundle-2.2.3/user-files/simulations/computerdatabase/
  creating:  gatling-charts-highcharts-bundle-2.2.3/user-files/simulations/computerdatabase/advanced/
  inflating:  gatling-charts-highcharts-bundle-2.2.3/user-files/simulations/computerdatabase/advanced/AdvancedSimulationStep01.scala
  inflating:  gatling-charts-highcharts-bundle-2.2.3/user-files/simulations/computerdatabase/advanced/AdvancedSimulationStep02.scala
  inflating:  gatling-charts-highcharts-bundle-2.2.3/user-files/simulations/computerdatabase/advanced/AdvancedSimulationStep03.scala
  inflating:  gatling-charts-highcharts-bundle-2.2.3/user-files/simulations/computerdatabase/advanced/AdvancedSimulationStep04.scala
  inflating:  gatling-charts-highcharts-bundle-2.2.3/user-files/simulations/computerdatabase/advanced/AdvancedSimulationStep05.scala
  inflating:  gatling-charts-highcharts-bundle-2.2.3/user-files/simulations/computerdatabase/BasicSimulation.scala
elena@localhost:~/Descargas$ cd gatling-charts-highcharts-bundle-2.2.3/bin/
elena@localhost:~/Descargas/gatling-charts-highcharts-bundle-2.2.3/bin$ ls
gatling.bat  gatling.sh  recorder.bat  recorder.sh
elena@localhost:~/Descargas/gatling-charts-highcharts-bundle-2.2.3/bin$ ./gatling.sh
GATLING HOME is set to /home/elena/Descargas/gatling-charts-highcharts-bundle-2.2.3
13:58:07.969 [WARN ] i.g.c.ZincCompiler$ - Pruning sources from previous analysis, due to incompatible CompileSetup.
OpenJDK 64-Bit Server VM warning: If the number of processors is expected to increase from one, then you should configure the number of parallel
GC threads appropriately using -XX:ParallelGCThreads=N
Choose a simulation number:
[0] computerdatabase.BasicSimulation
[1] computerdatabase.advanced.AdvancedSimulationStep01
[2] computerdatabase.advanced.AdvancedSimulationStep02
[3] computerdatabase.advanced.AdvancedSimulationStep03
[4] computerdatabase.advanced.AdvancedSimulationStep04
[5] computerdatabase.advanced.AdvancedSimulationStep05
```

Figura 4.2: CentOS, ejecución de Gatling.

```
elena@localhost:~/Descargas/gatling-charts-highcharts-bundle-2.2.3/bin

---- Scenario Name -----
[#####]100%
waiting: 0 / active: 0 / done:1

Simulation computerdatabase.BasicSimulation completed in 23 seconds
Parsing log file(s)...
Parsing log file(s) done
Generating reports...

=====
---- Global Information -----
> request count                13 (OK=13 KO=0 )
> min response time            65 (OK=65 KO=- )
> max response time            178 (OK=178 KO=- )
> mean response time           96 (OK=96 KO=- )
> std deviation                 35 (OK=35 KO=- )
> response time 50th percentile 79 (OK=79 KO=- )
> response time 75th percentile 112 (OK=112 KO=- )
> response time 95th percentile 162 (OK=162 KO=- )
> response time 99th percentile 175 (OK=175 KO=- )
> mean requests/sec            0.542 (OK=0.542 KO=- )
---- Response Time Distribution -----
> t < 800 ms                   13 (100%)
> 800 ms < t < 1200 ms         0 ( 0%)
> t > 1200 ms                   0 ( 0%)
> failed                        0 ( 0%)
=====

Reports generated in 0s.
Please open the following file: /home/elena/Descargas/gatling-charts-highcharts-bundle-2.2.3/results/basicsimulation-1482498233075/index.html
elena@localhost:~/Descargas/gatling-charts-highcharts-bundle-2.2.3/bin$
```

Figura 4.3: CentOS, resultados de Gatling, escenario 0.

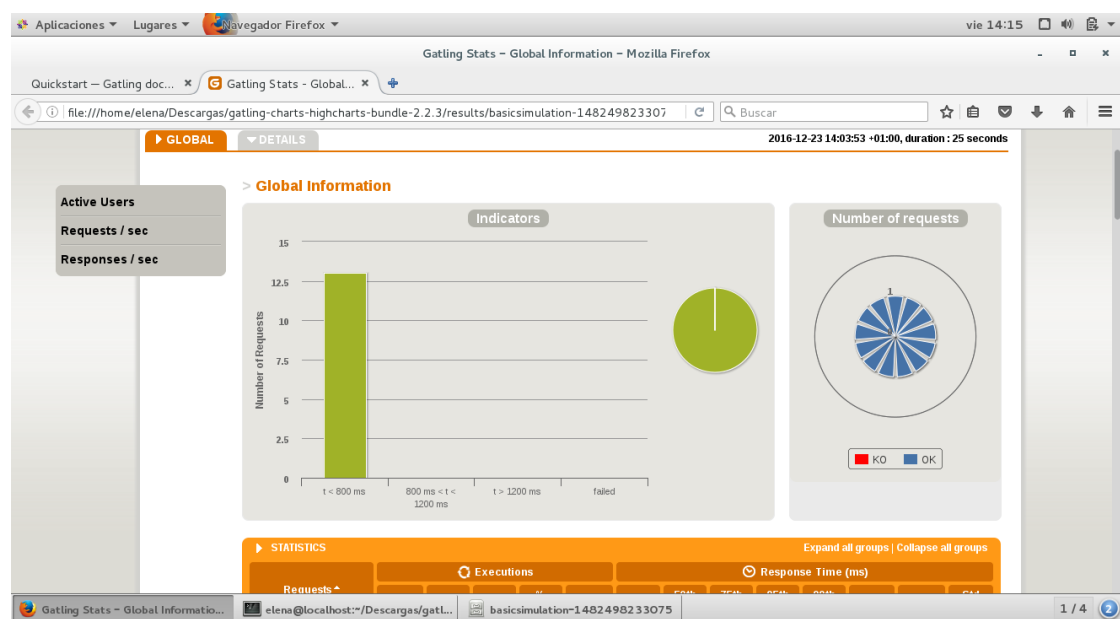


Figura 4.4: CentOS, resultados de Gatling, escenario 0.

```

elena@localhost:~/Descargas/gatling-charts-highcharts-bundle-2.2.3/bin

Archivo Editar Ver Buscar Terminal Ayuda
[#####]100%
waiting: 0 / active: 0 / done:1

Simulation computerdatabase.advanced.AdvancedSimulationStep01 completed in 16 seconds
Parsing log file(s)...
Parsing log file(s) done
Generating reports...

----- Global Information -----
> request count                13 (OK=13 K0=0 )
> min response time           70 (OK=70 K0=- )
> max response time          1585 (OK=1585 K0=- )
> mean response time          210 (OK=210 K0=- )
> std deviation                399 (OK=399 K0=- )
> response time 50th percentile 79 (OK=79 K0=- )
> response time 75th percentile 87 (OK=87 K0=- )
> response time 95th percentile 760 (OK=760 K0=- )
> response time 99th percentile 1420 (OK=1420 K0=- )
> mean requests/sec           0.812 (OK=0.812 K0=- )
----- Response Time Distribution -----
> t < 800 ms                   12 ( 92%)
> 800 ms < t < 1200 ms        0 ( 0%)
> t > 1200 ms                  1 ( 8%)
> failed                       0 ( 0%)

Reports generated in 0s.
Please open the following file: /home/elena/Descargas/gatling-charts-highcharts-bundle-2.2.3/results/advancedsimulationstep01-1482498975719/index.html
elena@localhost:~/Descargas/gatling-charts-highcharts-bundle-2.2.3/bin$

```

Figura 4.5: CentOS, resultados de Gatling, escenario 1.

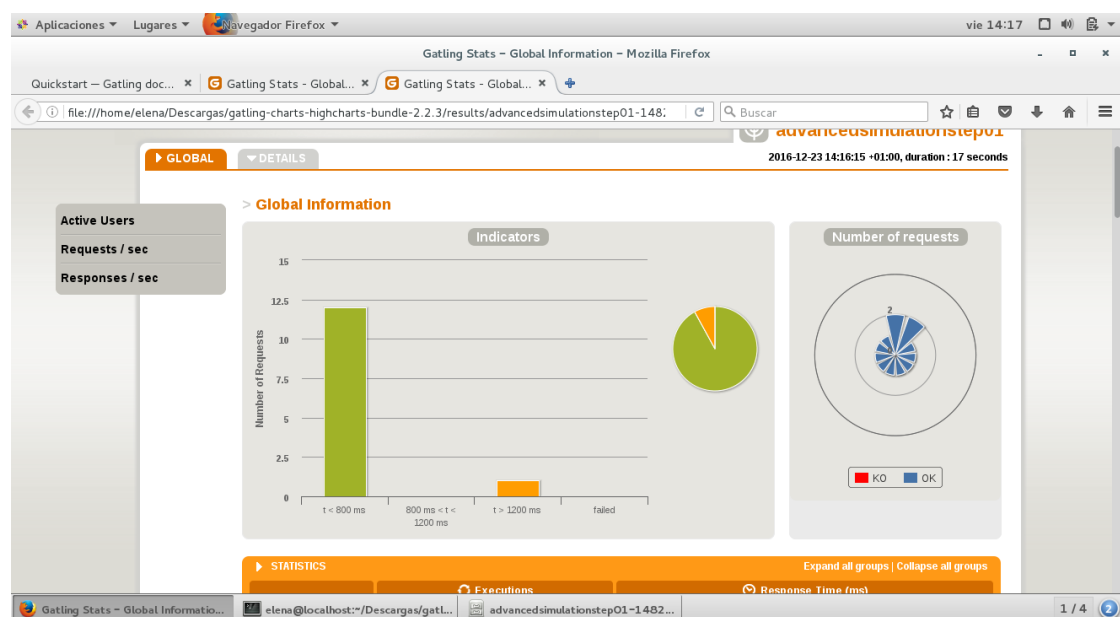


Figura 4.6: CentOS, resultados de Gatling, escenario 1

```

elena@localhost:~/Descargas/gatling-charts-highcharts-bundle-2.2.3/bin

Simulation computerdatabase.advanced.AdvancedSimulationStep05 completed in 16 seconds
Parsing log file(s) ...
Parsing log file(s) done
Generating reports ...

----- Global Information -----
> request count                105 (OK=103 KO=2 )
> min response time            59 (OK=59 KO=66 )
> max response time            458 (OK=458 KO=68 )
> mean response time           87 (OK=88 KO=67 )
> std deviation                 47 (OK=48 KO=1 )
> response time 50th percentile 74 (OK=74 KO=67 )
> response time 75th percentile 82 (OK=83 KO=68 )
> response time 95th percentile 152 (OK=153 KO=68 )
> response time 99th percentile 247 (OK=248 KO=68 )
> mean requests/sec            6.176 (OK=6.059 KO=0.118 )
----- Response Time Distribution -----
> t < 800 ms                   103 ( 98%)
> 800 ms < t < 1200 ms        0 ( 0%)
> t > 1200 ms                  0 ( 0%)
> failed                       2 ( 2%)
----- Errors -----
> status.find.is(201), but actually found 200                2 (100,0%)

Reports generated in 0s.
Please open the following file: /home/elena/Descargas/gatling-charts-highcharts-bundle-2.2.3/results/advancedsimulationstep05-1482499266017/index.html
elena@localhost:~/Descargas/gatling-charts-highcharts-bundle-2.2.3/bin$

```

Figura 4.7: CentOS, resultados de Gatling, escenario 5.

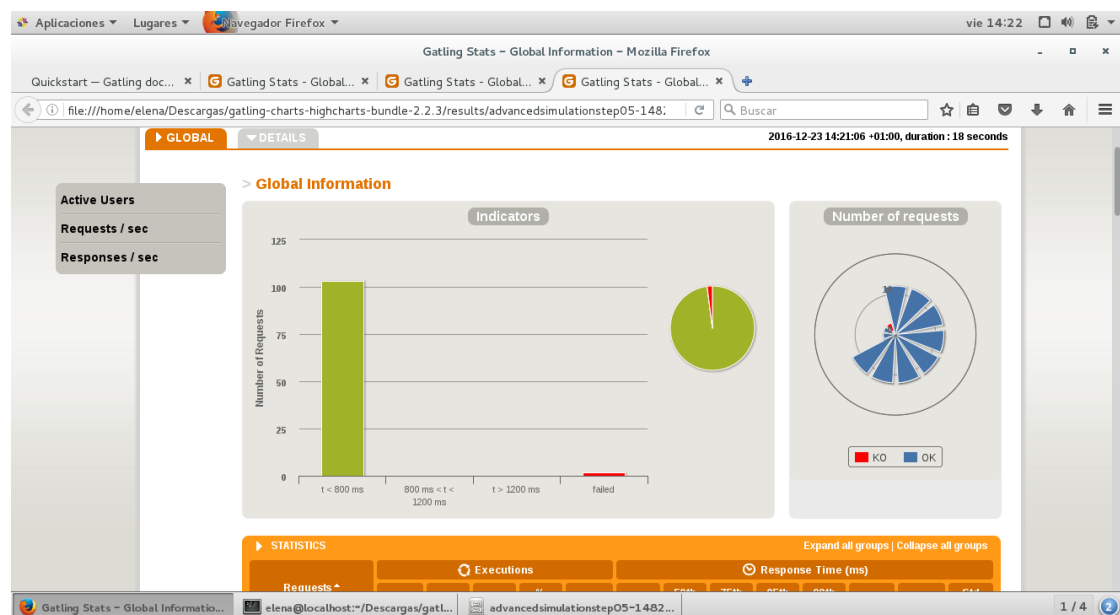


Figura 4.8: CentOS, resultados de Gatling, escenario 5



## 5. Cuestión 4:

- 5.1. Instale y siga el tutorial en <http://jmeter.apache.org/usermanual/build-web-test-plan.html> [1] realizando capturas de pantalla y comentándolas. En vez de usar la web de jmeter, haga el experimento usando sus máquinas virtuales ¿coincide con los resultados de ab?

Para instalar JMeter nos descargamos el binario .zip de la página oficial, lo descomprimos y ejecutamos el script `apache-jmeter-3.1/bin/jmeter`.

Lo primero que tenemos que hacer es crear un grupo de hilos, como se muestra en la figura 5.1 y nos saldrá algo parecido a la figura 5.2 con los valores por defecto. Cambiamos los valores del número de hilos a 5, y contador del bucle a 2, tal y como se muestra en la figura 5.3.

Ahora añadimos los valores de las peticiones HTTP por defecto, para ello debemos entrar tal y como se muestra en la figura 5.4, una vez añadidos se mostrará algo similar a la figura 5.5 teniendo en cuenta que hemos puesto la ip de nuestra máquina virtual de Ubuntu Server. También podemos añadir un gestor de Cookies de HTTP, como se muestra en la figura 5.6.

A continuación añadimos las peticiones HTTP para nuestro test, como se muestra en la figura 5.7 con los valores que se muestran en la figura 5.8.

Finalmente añadimos una gráfica para ver los resultados de los test creados tal y como se muestra en la figura 5.9 y seleccionamos el fichero de salida de JMeter (figura 5.10). Ejecutando el test de JMeter varias veces hacia Ubuntu Server se obtiene la siguiente gráfica (5.11). Hacemos lo mismo para Windows (figura 5.12) y CentOS (figura 5.13), cambiando la ip en los valores por defecto de las peticiones de HTTP. Para que las gráficas fuesen significativas he cambiado el número de bucles de 2 a 200.

Como se puede observar en las gráficas el rendimiento en Ubuntu es 33.5/minuto, en Windows 35,84/minuto y en CentOS 41.039/minuto.

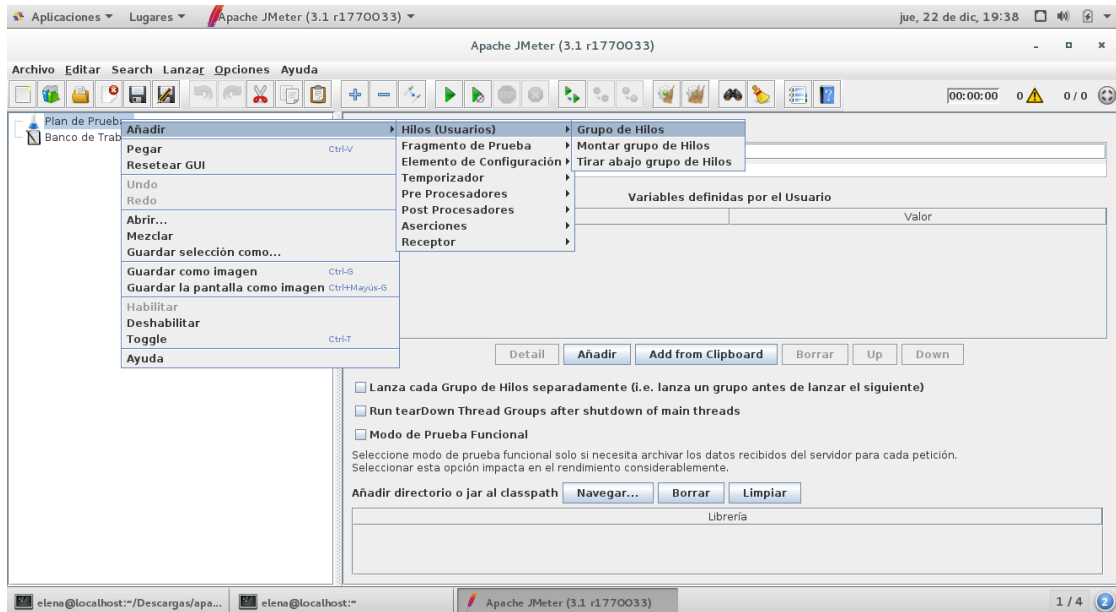


Figura 5.1: CentOS, JMeter - crear grupo de hilos.

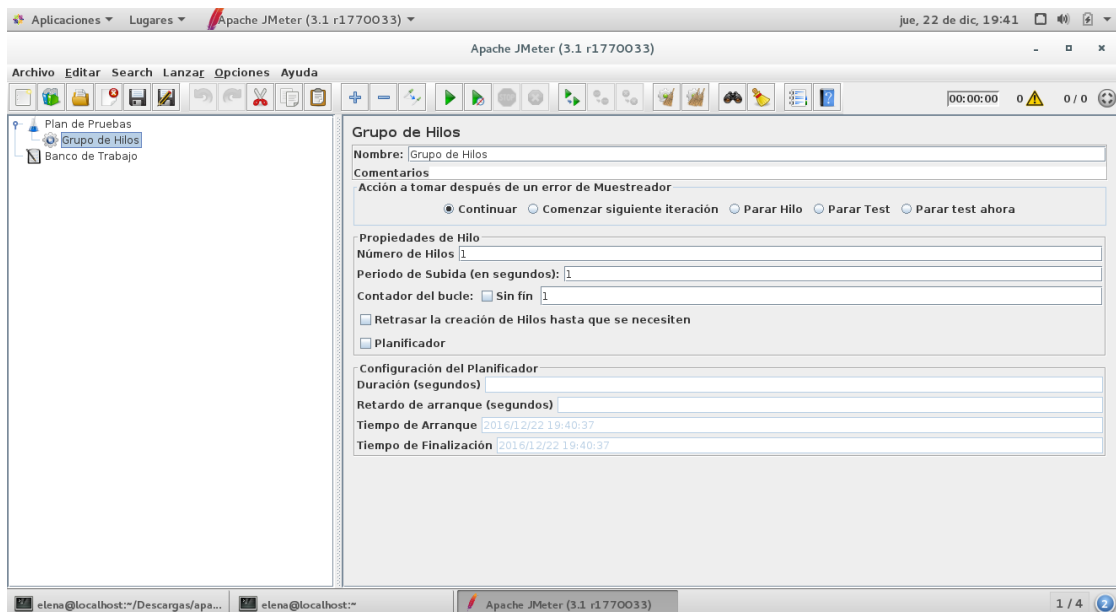


Figura 5.2: CentOS, JMeter - grupo de hilos por defecto.

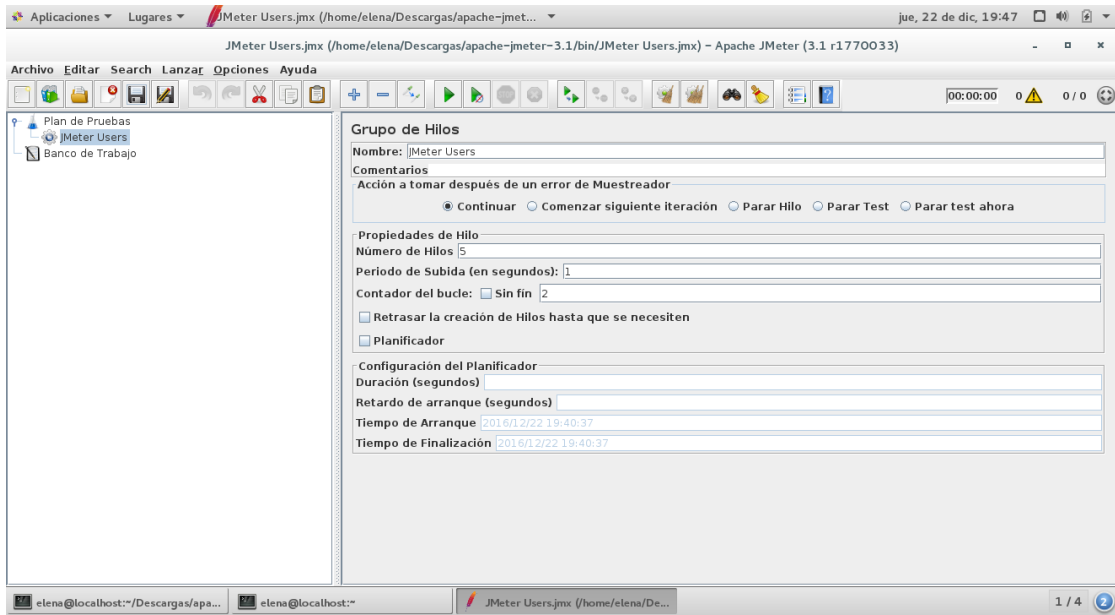


Figura 5.3: CentOS, JMeter - grupo de hilos.

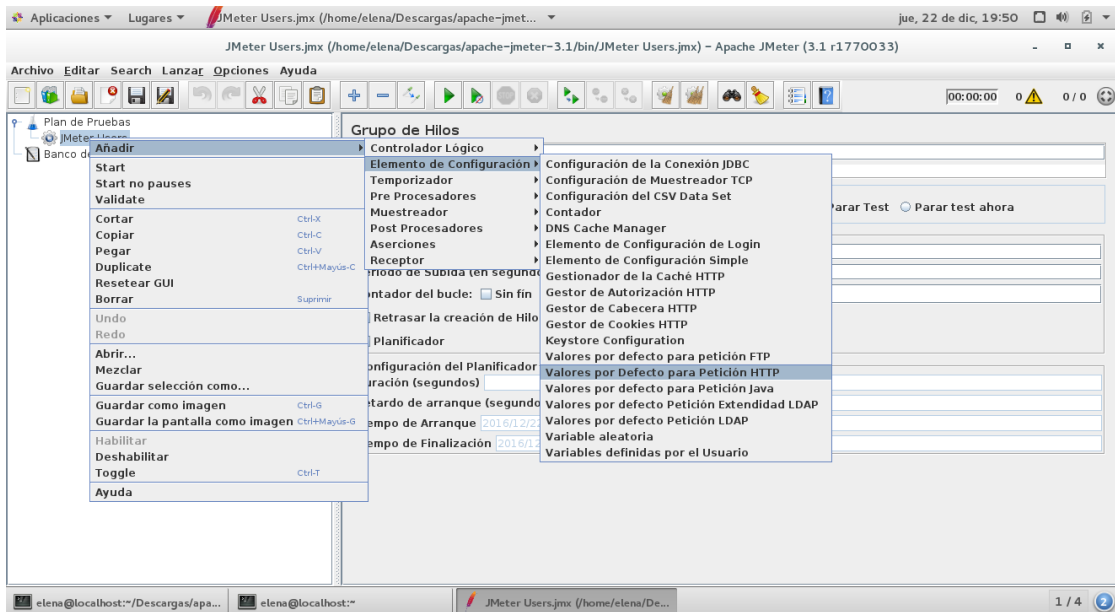


Figura 5.4: CentOS, JMeter - añadir valores de las peticiones HTTP por defecto.

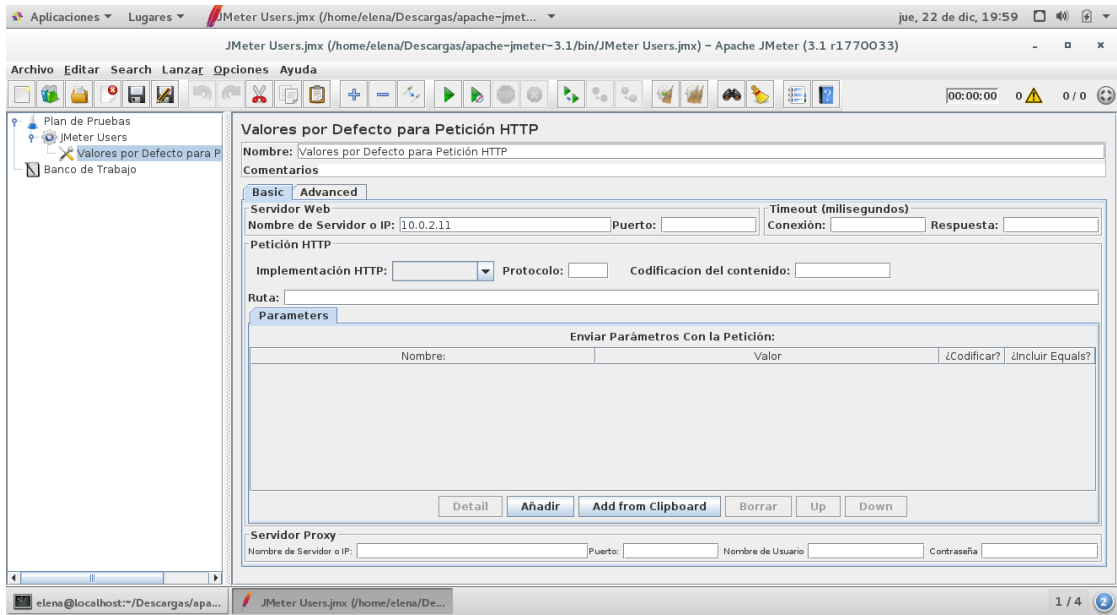


Figura 5.5: CentOS, JMeter - valores de las peticiones HTTP por defecto.

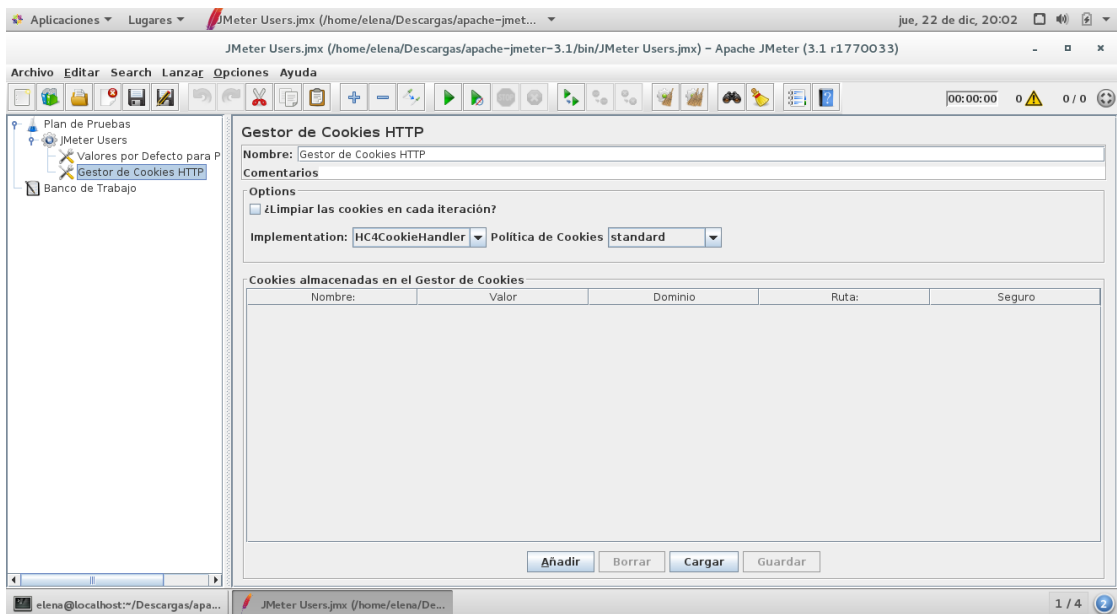


Figura 5.6: CentOS, JMeter - gestor de Cookies de HTTP.

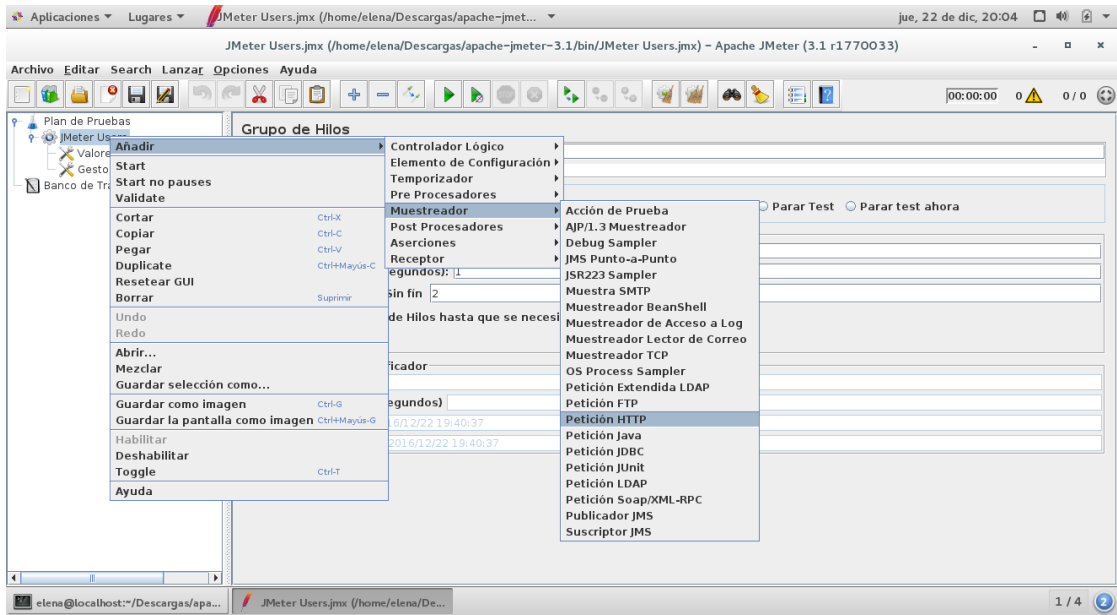


Figura 5.7: CentOS, JMeter - añadir peticiones HTTP.

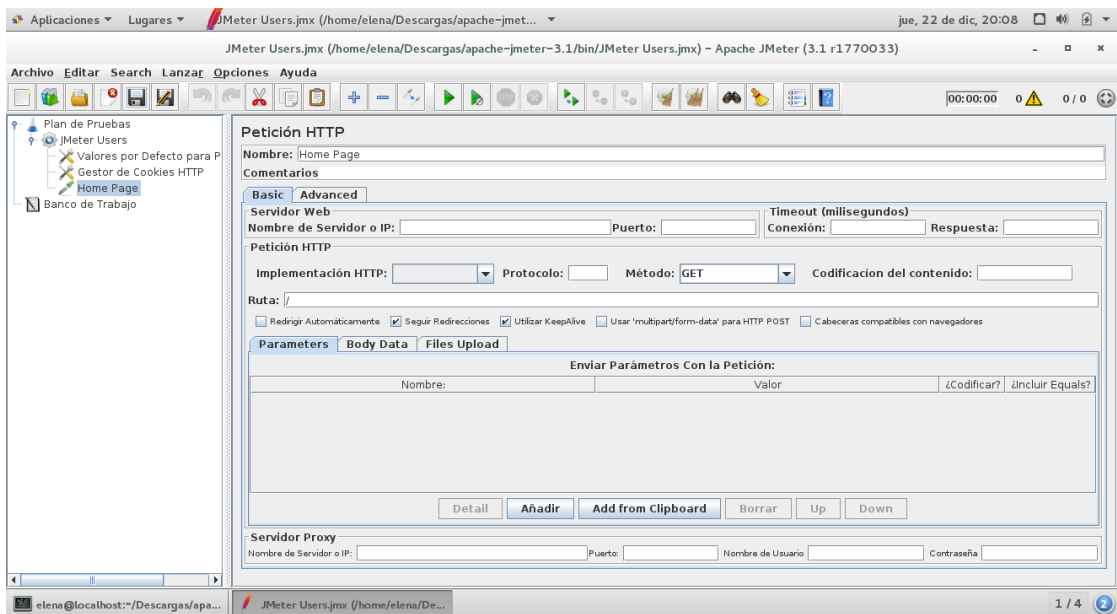


Figura 5.8: CentOS, JMeter - peticiones HTTP.

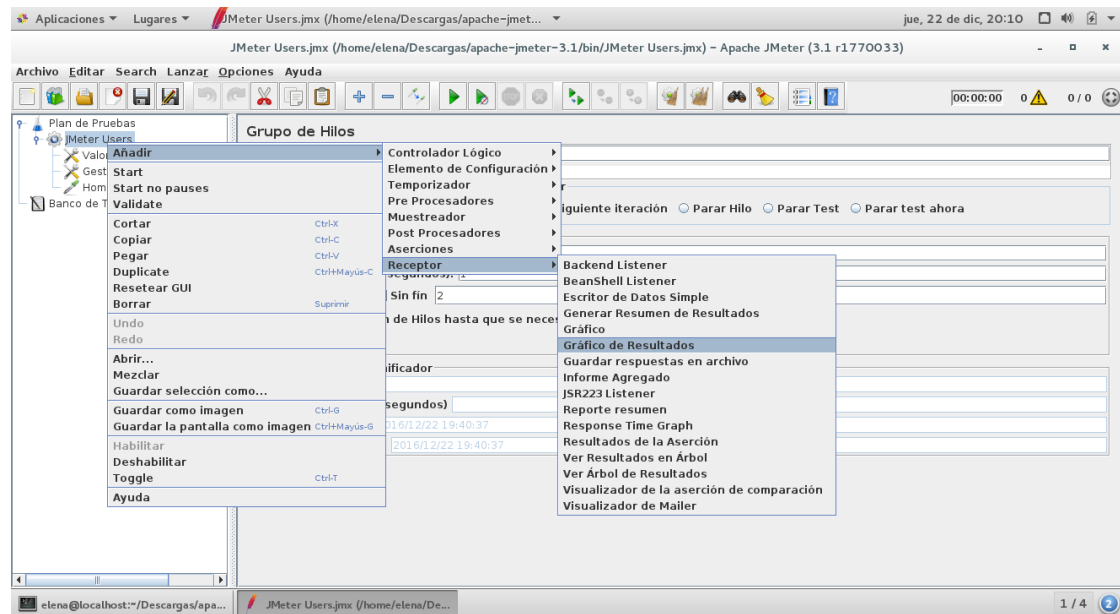


Figura 5.9: CentOS, JMeter - añadir gráfico.

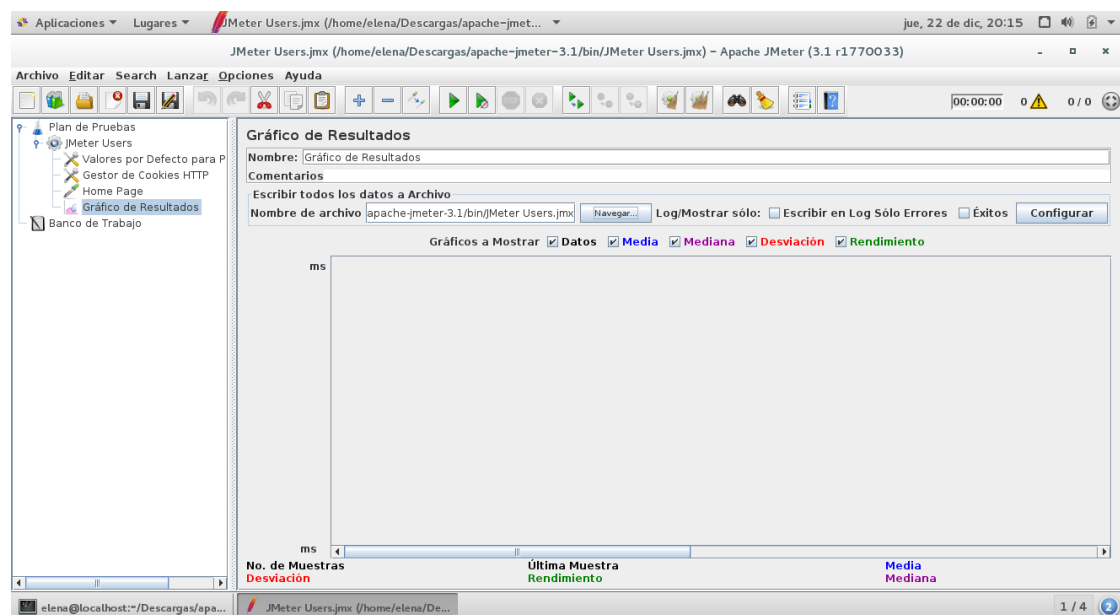


Figura 5.10: CentOS, JMeter - gráfico de resultados.

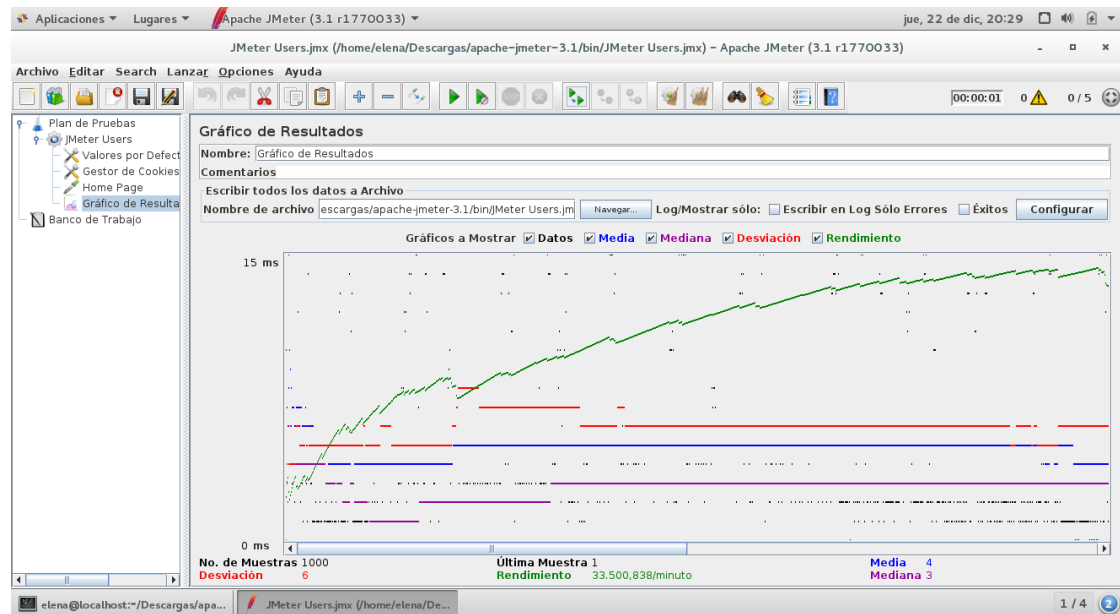


Figura 5.11: CentOS, JMeter - gráfico de resultados hacia Ubuntu Server.

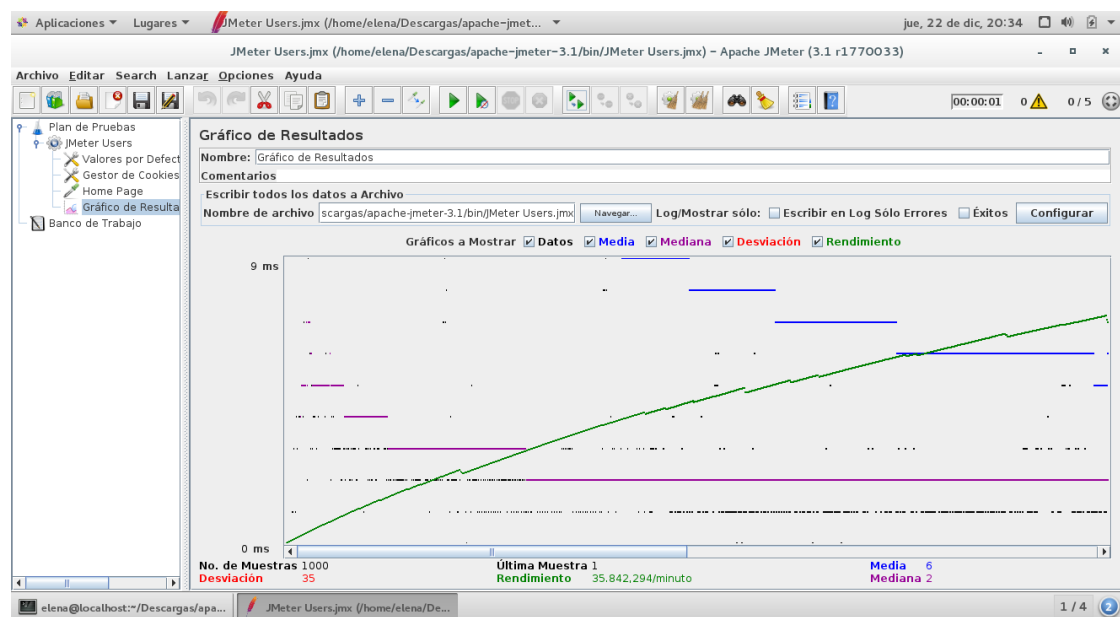


Figura 5.12: CentOS, JMeter - gráfico de resultados hacia Windows.

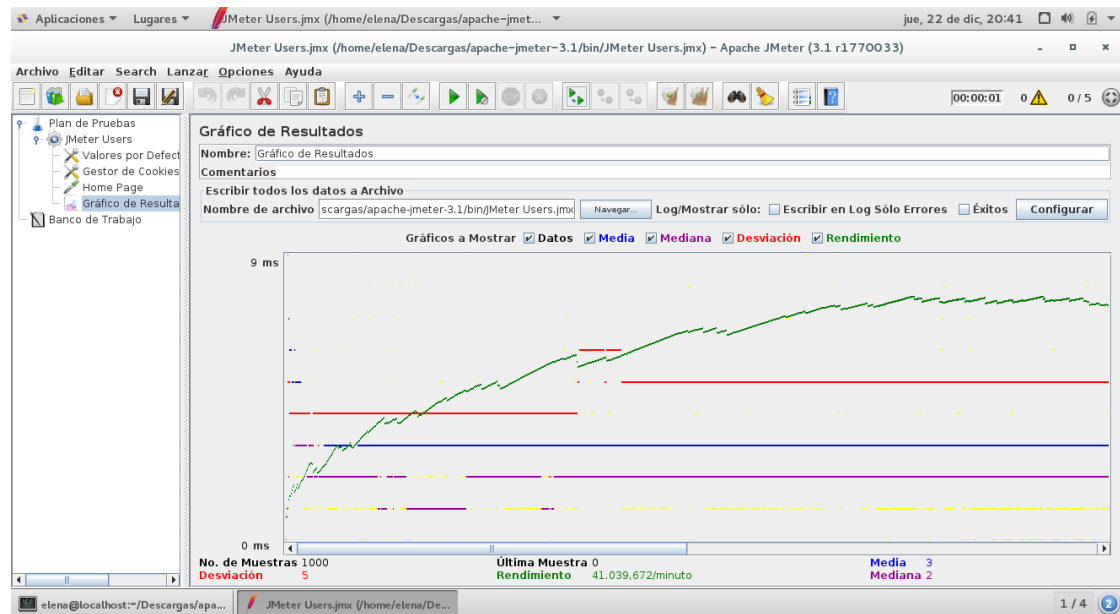


Figura 5.13: CentOS, JMeter - gráfico de resultados hacia CentOS.

## 6. Cuestión 5:

### 6.1. Programe un benchmark usando el lenguaje que desee. El benchmark debe incluir:

1. Objetivo del benchmark.
2. Métricas (unidades, variables, puntuaciones, etc.).
3. Instrucciones para su uso.
4. Ejemplo de uso analizando los resultados.

Tenga en cuenta que puede comparar varios gestores de BD, lenguajes de programación web (tiempos de ejecución, gestión de memoria, ...), duración de la batería, servidor DNS, etc. . Alternativamente, puede descargar alguno de algún repositorio en github y modificarlo según sus necesidades.

#### 6.1.1. Objetivo del benchmark.

Voy a utilizar como base un ejercicio que realizamos en la asignatura de Arquitectura de Computadores el cual calcula el tiempo de ejecutar n operaciones en coma flotante de forma secuencial. El benchmark está escrito en C y realiza un número de operaciones



de suma, multiplicación y división. El código del benchmark está incluido en el zip con nombre “benchmark\_float.c”

Por lo tanto el objetivo del benchmark es analizar el tiempo que tarda el procesador en realizar operaciones en coma flotante. Mostrará el tiempo de forma independiente de las sumas, multiplicaciones y divisiones.

### 6.1.2. Métricas (unidades, variables, puntuaciones, etc.).

Como ya hemos dicho anteriormente el benchmark realiza una medición del tiempo de ejecución, por cada una de las operaciones, la cual se muestra en segundos.

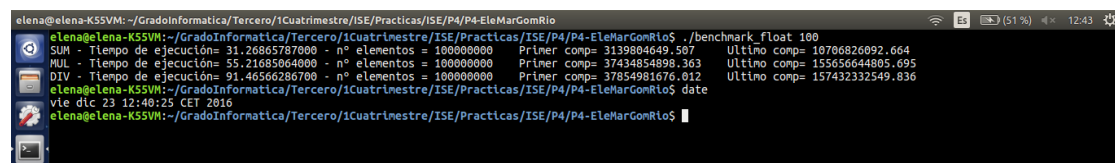
### 6.1.3. Instrucciones para su uso.

Para compilar el benchmark se usará la siguiente línea (lrt es una librería de tiempo real):  
`sudo gcc benchmark_float.c -o benchmark_float -lrt`

Para ejecutarlo se usa: `./benchmark_float <longitud>`, siendo “longitud” un valor pasado como parámetro por el que dependerá el número de operaciones realizadas, siendo su valor máximo 1000, por lo que si se introduce un valor superior a éste se limita a 1000.

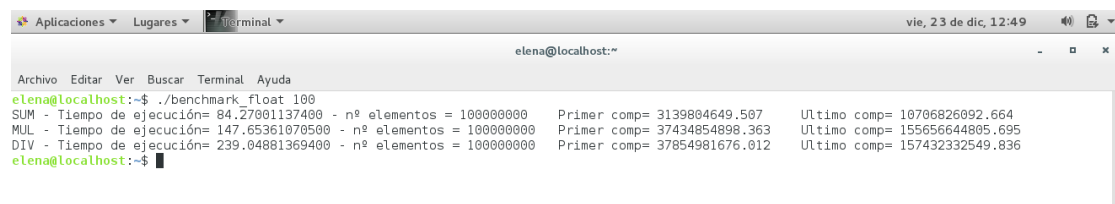
### 6.1.4. Ejemplo de uso analizando los resultados.

Los resultados de las ejecuciones del benchmark se pueden ver en las figuras: Ubuntu anfitrión 6.1, CentOS 6.2



```
elena@elena-K55VM: ~/GradoInformatica/Tercero/1Cuatrimestre/ISE/Practicas/ISE/P4/P4-EleMarGonRio$ ./benchmark_float 100
SUM - Tiempo de ejecución= 31.26865787000 - nº elementos = 1000000000 Primer comp= 3139804649.507 Ultimo comp= 10706826092.664
MUL - Tiempo de ejecución= 55.21685064900 - nº elementos = 1000000000 Primer comp= 37434854898.363 Ultimo comp= 155656644805.695
DIV - Tiempo de ejecución= 91.46566286700 - nº elementos = 1000000000 Primer comp= 37854981676.012 Ultimo comp= 157432332549.836
elena@elena-K55VM:~/GradoInformatica/Tercero/1Cuatrimestre/ISE/Practicas/ISE/P4/P4-EleMarGonRio$ date
vie dic 23 12:48:25 CET 2016
elena@elena-K55VM:~/GradoInformatica/Tercero/1Cuatrimestre/ISE/Practicas/ISE/P4/P4-EleMarGonRio$
```

Figura 6.1: Ubuntu anfitrión, resultados del benchmark.



```
elena@localhost:~$ ./benchmark_float 100
SUM - Tiempo de ejecución= 84.27001137400 - nº elementos = 1000000000 Primer comp= 3139804649.507 Ultimo comp= 10706826092.664
MUL - Tiempo de ejecución= 147.65361070500 - nº elementos = 1000000000 Primer comp= 37434854898.363 Ultimo comp= 155656644805.695
DIV - Tiempo de ejecución= 239.04881369400 - nº elementos = 1000000000 Primer comp= 37854981676.012 Ultimo comp= 157432332549.836
elena@localhost:~$
```

Figura 6.2: CentOS, resultados del benchmark.

Para poder comparar mejor los tiempos del benchmark extraigo los tiempos en la tabla 6.1 que se muestra a continuación.

Tabla 6.1: Tiempos de la ejecución del benchmark.

SO	Suma (s)	Multipliación (s)	División (s)
CentOS	84.27	147.65	239.05
Ubuntu	31.27	55.22	91.47

Se puede observar claramente que el anfitrión con Ubuntu es mucho mejor que CentOS en este tipo de operaciones.

## Referencias

- [1] <http://jmeter.apache.org/usermanual/build-web-test-plan.html>, consultado el 14 de Diciembre de 2016. Tutorial ejercicio 4.
- [2] <http://www.phoronix-test-suite.com/documentation/phoronix-test-suite.html>, consultado el 14 de Diciembre de 2016. Comandos de Phoronix.
- [3] [http://www.phoronix.com/scan.php?page=news\\_item&px=NzIwMQ](http://www.phoronix.com/scan.php?page=news_item&px=NzIwMQ), consultado el 14 de Diciembre de 2016. Phoronix.
- [4] <http://gatling.io/docs/2.2.3/quickstart.html>, consultado el 23 de Diciembre de 2016. Instalación de Gatling.
- [5] <https://www.scala-lang.org/what-is-scala.html>, consultado el 23 de Diciembre de 2016. Qué es Scala.