

**INGENIERÍA DE SERVIDORES (2016-2017)**  
DOBLE GRADO EN INGENIERÍA INFORMÁTICA Y MATEMÁTICAS  
UNIVERSIDAD DE GRANADA

---

## Memoria Práctica 5

---

Miguel Ángel Torres López

26 de mayo de 2017

## Índice

1	Al modificar los valores del kernel de este modo, no logramos que persistan después de reiniciar la máquina. ¿Qué archivo hay que editar para que los cambios sean permanentes?	4
2	¿Con qué opción se muestran todos los parámetros modificables en tiempo de ejecución?	4
2.1	Elija dos parámetros y explique, en dos líneas, qué función tienen. . . . .	4
3	Realice una copia de seguridad del registro, modifique el valor de una clave y restaure la copia de seguridad hecha. Compruebe que la clave tiene el valor original.	5
4	Enumere qué elementos se pueden configurar en Apache y en IIS para que Moodle funcione mejor.	8
5	Ajuste la compresión en el servidor y analice su comportamiento usando varios valores para el tamaño del archivo a partir del cual comprimir. para comprobar que está comprimiendo puede usar el navegador o comandos como curl o lynx. Muestre capturas de pantalla de todo el proceso.	9
6	Elija un servicio y modifique un parámetro para mejorar su comportamiento. Monitoree el servicio antes y después de la modificación del parámetro aplicando cargas al sistema (antes y después) mostrando los resultados de la monitorización.	12

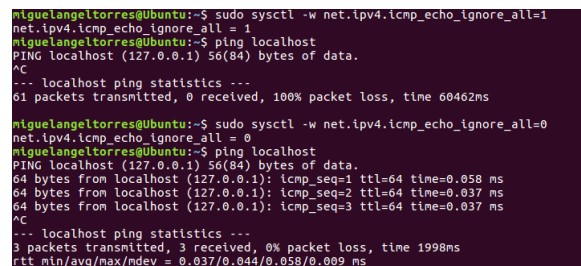
## Índice de figuras

1.1. Usando sysctl para cambiar características del sistema. . . . .	4
2.1. Valor del parámetro threads-max . . . . .	5
2.2. Valor del parámetro dmesg_restrict . . . . .	5
3.1. Abrir registro para edición. . . . .	6
3.2. Crear una nueva clave en el registro. . . . .	6
3.3. Exportar registro a un lugar seguro. . . . .	7
3.4. Registro exportado a un lugar seguro. . . . .	7
3.5. Cambio de valor de la clave creada. . . . .	8
3.6. Importar registro y recuperación del valor original de la clave. . . . .	8
5.1. Administrador de IIS . . . . .	10
5.2. Valores por defecto de la compresión . . . . .	10
5.3. Servidor web con contenido añadido. . . . .	11
5.4. Información sobre la petición al servidor web. . . . .	11
5.5. Cambiando el valor de las preferencias de compresión . . . . .	12
5.6. Información sobre la petición al servidor con mínimo para compresión alterado	12
6.1. Rendimiento base del servidor web. . . . .	13
6.2. Cambio de KeepAliveTimeout en el archivo de configuración. . . . .	13
6.3. Repercusión en el servidor web tras cambiar KeepAliveTimeout. . . . .	13
6.4. Cambio de KeepAlive en el archivo de configuración. . . . .	14
6.5. Repercusión en el servidor web tras desactivar KeepAlive. . . . .	14

## Índice de tablas

## 1. Al modificar los valores del kernel de este modo, no logramos que persistan después de reiniciar la máquina. ¿Qué archivo hay que editar para que los cambios sean permanentes?

Tal y como se menciona en la documentación de RedHat[9] podemos usar el archivo *sysctl.conf*. Este archivo perdurará entre reinicios y será leído por el kernel en cada inicio de la máquina, cargando así las preferencias establecidas en él. Por ejemplo, vamos a modificar un parámetro para que nuestra máquina no responda a un *ping*. Por motivos de rapidez y no tener que reiniciar la máquina, voy a usar el comando *sysctl*. Como puede verse en la Figura 1.1 los *ping* dejan de contestarse.



```
miguelangel@ubuntu:~$ sudo sysctl -w net.ipv4.icmp_echo_ignore_all=1
net.ipv4.icmp_echo_ignore_all = 1
miguelangel@ubuntu:~$ ping localhost
PING localhost (127.0.0.1) 56(84) bytes of data.
^C
--- localhost ping statistics ---
61 packets transmitted, 0 received, 100% packet loss, time 60462ms

miguelangel@ubuntu:~$ sudo sysctl -w net.ipv4.icmp_echo_ignore_all=0
net.ipv4.icmp_echo_ignore_all = 0
miguelangel@ubuntu:~$ ping localhost
PING localhost (127.0.0.1) 56(84) bytes of data:
64 bytes from localhost (127.0.0.1): icmp_seq=1 ttl=64 time=0.058 ms
64 bytes from localhost (127.0.0.1): icmp_seq=2 ttl=64 time=0.037 ms
64 bytes from localhost (127.0.0.1): icmp_seq=3 ttl=64 time=0.037 ms
^C
--- localhost ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 1998ms
rtt min/avg/max/ndev = 0.037/0.044/0.058/0.009 ms
```

Figura 1.1: Usando *sysctl* para cambiar características del sistema.

Nótese que al cambiar el archivo los cambios no tendrán efecto a no se que se obligue al kernel a cargar el archivo de nuevo. Esto se hace con opción *-p* del comand *sysctl*[4].

## 2. ¿Con qué opción se muestran todos los parámetros modificables en tiempo de ejecución?

Como puede verse en el manual en línea[4], la opción *-a* lista todos los valores disponibles para modificar en tiempo de ejecución. Se puede usar el comando *grep* visto en prácticas anteriores para filtrar el resultado.

### 2.1. Elija dos parámetros y explique, en dos líneas, qué función tienen.

En la referencia [5], tenemos una gran cantidad de parámetros especificados con una pequeña descripción y su rango de valores. Por ejemplo:

- **threads-max.** Determina el número máximo de hebras que pueden crearse derivados de la utilidad para dividir tareas *fork*. El valor mínimo permitido es 20, es decir, como mínimo se tiene que dejar derivarse 20 threads.

- **dmesg\_restrict.** Cuando está activado, valor 1, se obliga a los usuarios a tener privilegios para poder ejecutar la orden dmesg. Si está desactivado, valor 0, todos los usuarios pueden utilizar dmesg para ver las llamadas al kernel.

Ambos parámetros pueden verse con el comando `sysctl -a | grep kernel` con sus valores pre-determinados. En las Figuras 2.1 y 2.2 pueden verse los de mi máquina.

```
kernel.stack_tracer_enabled = 0
kernel.sysctl_writes_strict = 0
kernel.sysrq = 176
kernel.tainted = 0
kernel.threads-max = 62339
kernel.timer_migration = 1
kernel.traceoff_on_warning = 0
kernel.tracepoint_printk = 0
kernel.unknown_nmi_panic = 0
```

Figura 2.1: Valor del parámetro threads-max

```
kernel.core_pattern = /usr/share/apport/apport %p %s %c %P
kernel.core_pipe_limit = 0
kernel.core_uses_pid = 0
kernel.ctrl-alt-del = 0
kernel.dmesg_restrict = 0
kernel.domainname = (none)
kernel.ftrace_dump_on_oops = 0
kernel.ftrace_enabled = 1
```

Figura 2.2: Valor del parámetro dmesg\_restrict

### 3. Realice una copia de seguridad del registro, modifique el valor de una clave y restaure la copia de seguridad hecha. Compruebe que la clave tiene el valor original.

En la documentación de Windows[7] vienen algunas directrices para hacerlo. Puesto que lo hacemos con interfaz, el cambio y la creación de claves es bastante intuitiva. El proceso a seguir para comprobar que la restauración del registro es efectiva es el siguiente:  
Primero abrimos el registro desde el buscador del inicio.

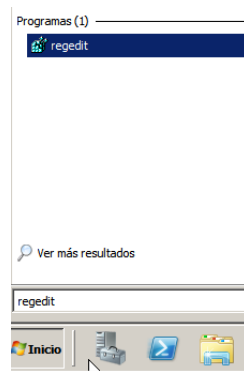


Figura 3.1: Abrir registro para edición.

Creamos una nueva clave de ejemplo y le damos un valor cualquiera.

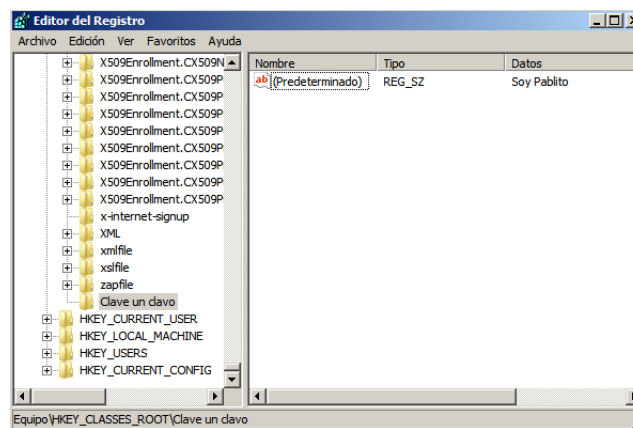


Figura 3.2: Crear una nueva clave en el registro.

Desde el menú vemos las opciones de importar y exportar, la primera para cargar un registro y la segunda para exportar el registro actual a un archivo. Pinchamos en exportar y seleccionamos un lugar seguro donde guardar el archivo con la configuración actual del registro.

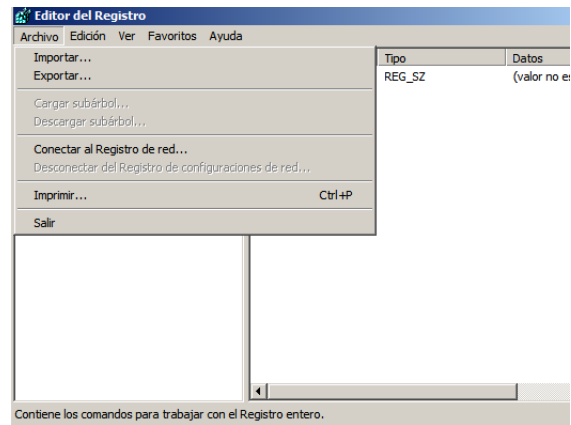


Figura 3.3: Exportar registro a un lugar seguro.

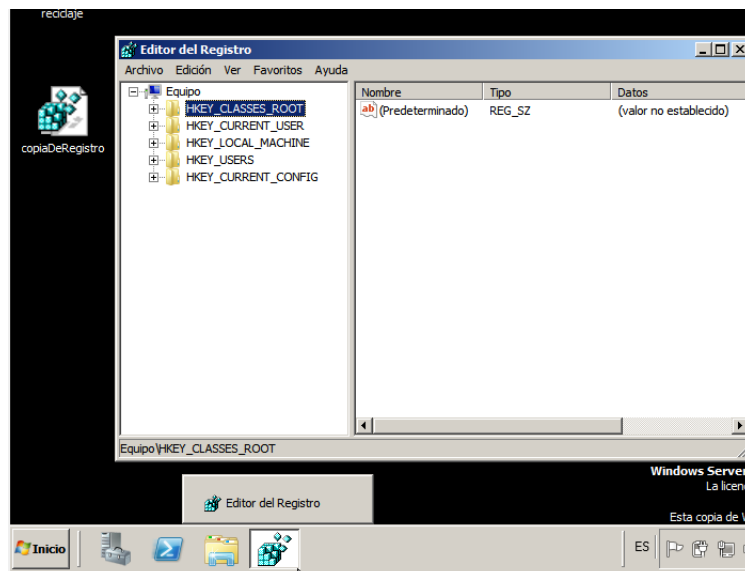


Figura 3.4: Registro exportado a un lugar seguro.

Ahora cambiamos el valor de nuestra clave creada.

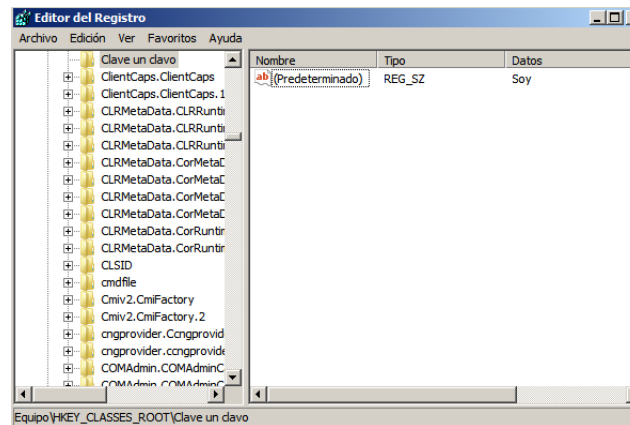


Figura 3.5: Cambio de valor de la clave creada.

En el mismo panel de antes, se nos da la opción de importar un registro. Desde el buscador que aparece elegimos el archivo que contiene los datos del registros anterior. Nos aparecerá un mensaje avisando de la sobrescritura de los valores de las claves del registro.

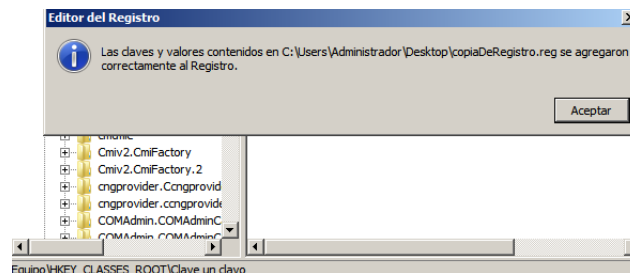


Figura 3.6: Importar registro y recuperación del valor original de la clave.

#### 4. Enumere qué elementos se pueden configurar en Apache y en IIS para que Moodle funcione mejor.

En la documentación[8] de Moodle vienen varios parámetros recomendados para mejorar el rendimiento de la plataforma. Entre estos hay una sección específica para Apache y otra para IIS.

##### ■ Apache Server

- Establecer *MaxRequestWorkers* como el 80% de la memoria total dividido por el máximo de memoria permitido por proceso de Apache.
- Usar Apache 2 en lugar de Apache.



- Establecer *KeepAliveTimeout* a 2 o 5 segundos. Para servidores muy saturados, considerar desactivar *KeepAlive* solo si la página no contiene recursos dinámicos.
- Desactivar *ExtendedStatus*.
- Desactivar *HostnameLookups* para reducir la latencia de DNS.
- Reducir el tiempo de *TimeOut*.
- Considerar el uso de *mod\_deflate* para comprimir las respuestas del servidor.

■ IIS

- El equivalente a *KeepAliveTimeout* es *ListenBackLog*.
- Se puede cambiar *MemCacheSize* para probar otros tamaños para la caché de IIS.
- Crear una DWORD llamada *ObjectCacheTTL* para cambiar el tiempo que permanece un objeto en caché, por defecto a 30 segundos.

**5. Ajuste la compresión en el servidor y analice su comportamiento usando varios valores para el tamaño del archivo a partir del cual comprimir. para comprobar que está comprimiendo puede usar el navegador o comandos como curl o lynx. Muestre capturas de pantalla de todo el proceso.**

En la documentación de Microsoft[6] viene paso a paso explicado el proceso para activar y ajustar la compresión de IIS. Primero buscamos el administrador de IIS en el buscador de inicio. Al abrirlo buscamos en la vista de características la opción de compresión como aparece en la Figura 5.1

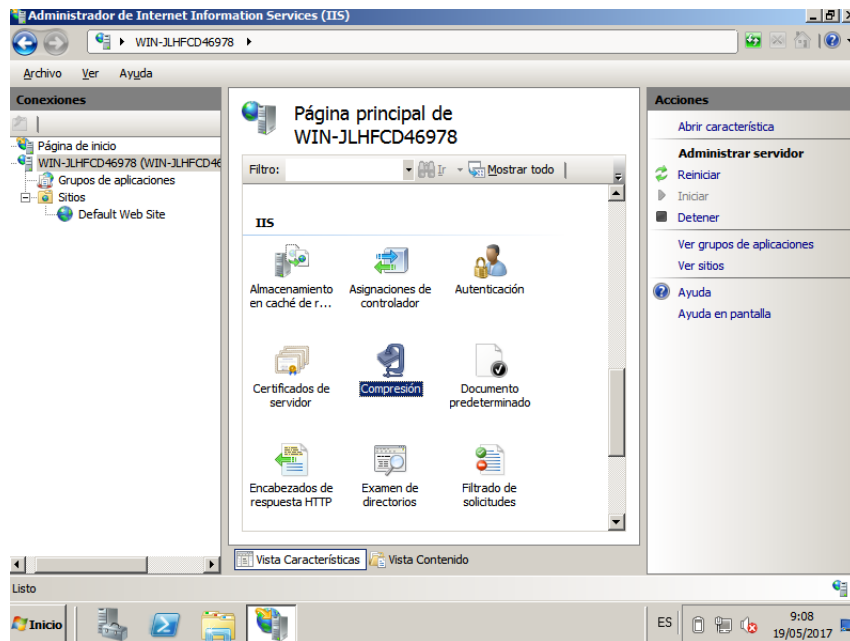


Figura 5.1: Administrador de IIS

Se nos abrirá un panel como el que aparece en la Figura 5.2. Por ahora, vamos a hacer una prueba tal cual están los datos por defecto, así que dejaremos todas las casillas como se muestran en la Figura.

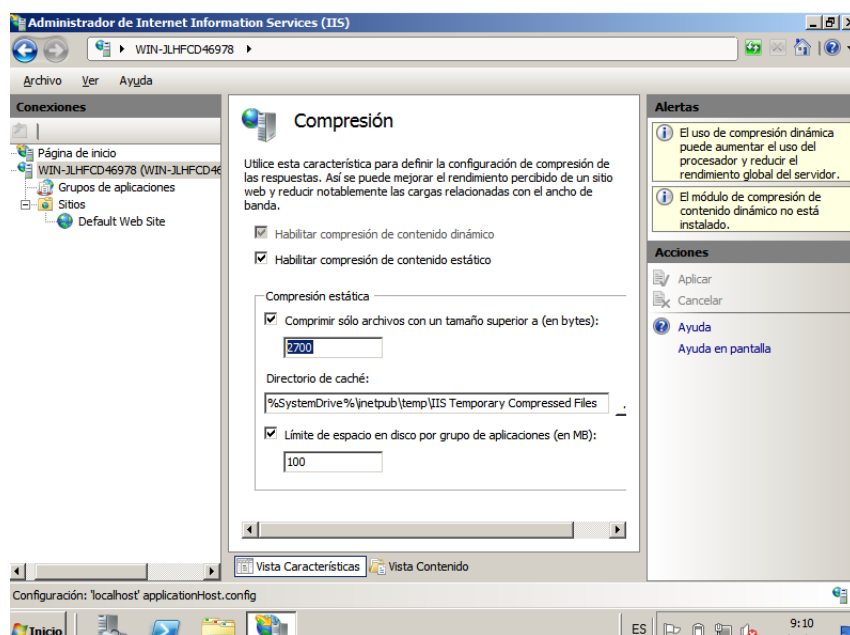


Figura 5.2: Valores por defecto de la compresión

Con motivo de comprobar si la página verdaderamente se comprime o no, modificaré su contenido para añadirle más peso. En la práctica dejé solo un mensaje saludando, ahora adjuntaré dos fotografías y buen conjunto de mensajes como en la Figura 5.3. Esto hará que nuestra página se comprima con los valores por defectos de compresión.

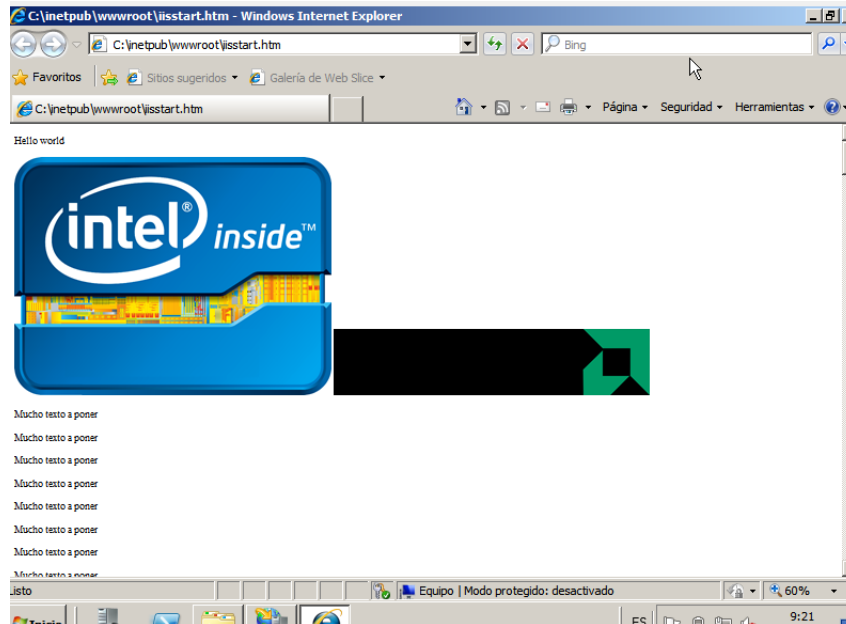


Figura 5.3: Servidor web con contenido añadido.

Podemos ver en la Figura 5.4 como usando la orden `curl[3]` sacamos la información de las peticiones al servidor web. Observamos que en la cabecera se nos especifica que se ha usado `gzip` para comprimir los datos transmitidos.

```
miguelangeltorres@Ubuntu:~$ curl 192.168.56.103 -I -H 'Accept-Encoding: gzip,deflate'
HTTP/1.1 200 OK
Date: Fri, 19 May 2017 11:17:16 GMT
Server: Microsoft-IIS/7.5
Last-Modified: Thu, 04 May 2017 09:17:30 GMT
ETag: "2c39-54eaf3e0dd05b-gzip"
Accept-Ranges: bytes
Vary: Accept-Encoding
Content-Encoding: gzip
Content-Length: 3186
Content-Type: text/html
```

Figura 5.4: Información sobre la petición al servidor web.

Ahora modificamos el valor mínimo que el archivo ha de tener para ser comprimido. Pondremos un número muy alto para asegurarnos de que nuestra página no sobrepasa el mínimo.

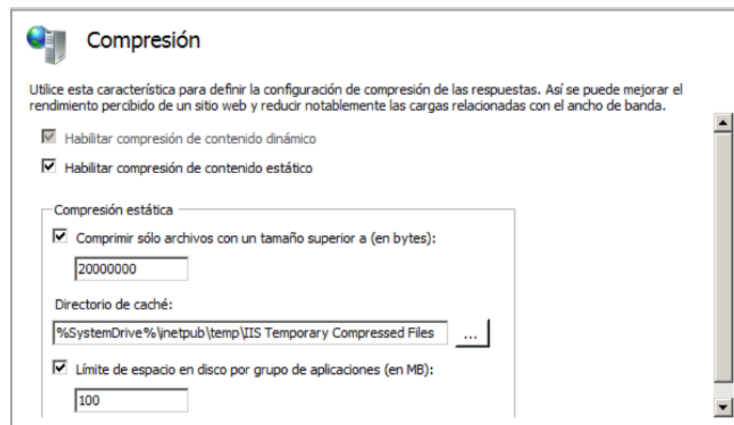


Figura 5.5: Cambiando el valor de las preferencias de compresión

En la Figura 5.6 tenemos el resultado de ejecutar de nuevo curl. Ahora el mínimo de peso para empezar a comprimir es más grande que el peso de nuestra página y, por lo tanto, la compresión ya no es realizada al enviar los datos. Podemos ver también que el tamaño de los datos enviados ha crecido. Esto se debe a que anteriormente se estaba comprimiendo al enviar la información.

```
miguelangel@ubuntu:~$ curl 192.168.56.103 -I -H 'Accept-Encoding: gzip,deflate'
HTTP/1.1 200 OK
Date: Fri, 19 May 2017 11:17:16 GMT
Server: Microsoft-IIS/7.5
Last-Modified: Thu, 04 May 2017 09:17:30 GMT
ETag: "2c39-54eaf3e0dd05b-gzip"
Accept-Ranges: bytes
Content-Length: 3846
Content-Type: text/html
```

Figura 5.6: Información sobre la petición al servidor con mínimo para compresión alterado

## 6. Elija un servicio y modifique un parámetro para mejorar su comportamiento. Monitoree el servicio antes y después de la modificación del parámetro aplicando cargas al sistema (antes y después) mostrando los resultados de la monitorización.

Dado que hemos visto en un ejercicio anterior como mejorar el rendimiento de un servidor apache modificando varios parámetros, ahora vamos a ponerlo en práctica para comprobar si realmente se produce una mejora. Primero haremos una prueba con la herramienta *ab*[1] vista en prácticas anteriores para ver el rendimiento base del servidor web.

```
Licensed to The Apache Software Foundation, http://www.apache.org/
Benchmarking 192.168.56.101 (be patient).....done

Server Software:      Apache/2.4.18
Server Hostname:      192.168.56.101
Server Port:          80

Document Path:        /
Document Length:      69 bytes

Concurrency Level:    5
Time taken for tests:  0.218 seconds
Complete requests:    100
Failed requests:       0
Keep-Alive requests:  100
Total transferred:    37405 bytes
HTML transferred:     6900 bytes
Requests per second:  458.75 [#/sec] (mean)
Time per request:     10.899 [ms] (mean)
Time per request:     2.180 [ms] (mean, across all concurrent requests)
Transfer rate:        167.57 [Kbytes/sec] received
```

Figura 6.1: Rendimiento base del servidor web.

Siguiendo las indicaciones de la cuestión 4, vamos a modificar el parámetro *KeepAliveTimeout* y lo vamos a reducir. Para ello vamos al archivo de configuración del servidor Apache, situado por defecto en */etc/apache2/apache2.conf*. Como se puede ver en la figura 6.2 el archivo de configuración contiene una pequeña explicación del parámetro, una explicación más detallada puede verse en la documentación de Apache[2].

```
#
# KeepAliveTimeout: Number of seconds to wait for the next request from the
# same client on the same connection.
#
KeepAliveTimeout 2
```

Figura 6.2: Cambio de KeepAliveTimeout en el archivo de configuración.

Veamos ahora como ha mejorado el rendimiento con respecto a la anterior ejecución. Se ha reducido el tiempo de respuesta de cada petición y se ha aumentado ligeramente la velocidad de transferencia de datos como puede verse en la Figura 6.3

```
Licensed to The Apache Software Foundation, http://www.apache.org/
Benchmarking 192.168.56.101 (be patient).....done

Server Software:      Apache/2.4.18
Server Hostname:      192.168.56.101
Server Port:          80

Document Path:        /
Document Length:      69 bytes

Concurrency Level:    5
Time taken for tests:  0.198 seconds
Complete requests:    100
Failed requests:       0
Keep-Alive requests:  100
Total transferred:    37405 bytes
HTML transferred:     6900 bytes
Requests per second:  495.50 [#/sec] (mean)
Time per request:     10.719 [ms] (mean)
Time per request:     2.124 [ms] (mean, across all concurrent requests)
Transfer rate:        182.41 [Kbytes/sec] received
```

Figura 6.3: Repercusión en el servidor web tras cambiar KeepAliveTimeout.

Ahora vamos a dar un paso más, vamos a desactivar el *KeepAlive*. Para ello volvemos a modificar el archivo de configuración como puede verse en la Figura 6.4.

```
#  
# KeepAlive: Whether or not to allow persistent connections (more than  
# one request per connection). Set to "Off" to deactivate.  
#  
KeepAlive On Off
```

Figura 6.4: Cambio de KeepAlive en el archivo de configuración.

El nuevo test de ab ejecutado contra el servidor web nos da la salida que aparece en la Figura 6.5. Volvemos a ver una tímida mejoría en las características medidas. La mejora es bastante pequeña a causa del tamaño de la web, que hace que los datos transferidos sean muy pocos y haga menos significativo el cambio de los parámetros.

```
Licensed to The Apache Software Foundation, http://www.apache.org/  
Benchmarking 192.168.56.101 (be patient).....done  
  
Server Software:      Apache/2.4.18  
Server Hostname:      192.168.56.101  
Server Port:          80  
  
Document Path:        /  
Document Length:      69 bytes  
  
Concurrency Level:    5  
Time taken for tests:  0.187 seconds  
Complete requests:    100  
Failed requests:       0  
Total transferred:     37405 bytes  
HTML transferred:     6900 bytes  
Requests per second:  488.50 [#/sec] (mean)  
Time per request:      10.697 [ms] (mean)  
Time per request:      2.119 [ms] (mean, across all concurrent requests)  
Transfer rate:         185.04 [Kbytes/sec] received
```

Figura 6.5: Repercusión en el servidor web tras desactivar KeepAlive.

## Referencias

- [1] Apache.org. <http://httpd.apache.org/docs/current/programs/ab.html>.
- [2] Apache.org. <https://httpd.apache.org/docs/2.4/mod/core.html>.
- [3] Linux Die. <https://linux.die.net/man/1/curl>.
- [4] Linux Die. <https://linux.die.net/man/8/sysctl>.
- [5] Kernel.org. <https://www.kernel.org/doc/Documentation/sysctl/kernel.txt>.
- [6] Microsoft. [https://technet.microsoft.com/en-us/library/cc730629\(v=ws.10\).aspx](https://technet.microsoft.com/en-us/library/cc730629(v=ws.10).aspx).
- [7] Microsoft. [https://www.microsoft.com/resources/documentation/windows/xp/all/proddocs/en-us/regedit\\_o.mspx?mfr=true](https://www.microsoft.com/resources/documentation/windows/xp/all/proddocs/en-us/regedit_o.mspx?mfr=true).
- [8] Moodle. [https://docs.moodle.org/33/en/Performance\\_recommendations#Apache\\_performance](https://docs.moodle.org/33/en/Performance_recommendations#Apache_performance).
- [9] RedHat. [https://access.redhat.com/documentation/en-US/Red\\_Hat\\_Enterprise\\_MRG/1.3/html/Realtime\\_Tuning\\_Guide/sect-Realtime\\_Tuning\\_Guide-General\\_System\\_Tuning-Setting\\_persistent\\_tuning\\_parameters.html](https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_MRG/1.3/html/Realtime_Tuning_Guide/sect-Realtime_Tuning_Guide-General_System_Tuning-Setting_persistent_tuning_parameters.html).