

**Ingeniería de Servidores (2014-2015)**  
GRADO EN INGENIERÍA INFORMÁTICA  
UNIVERSIDAD DE GRANADA

---

## Memoria Práctica 5

---

Jesús Ángel González Novez

13 de enero de 2015

## 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? En Performance Tuning for Linux Servers puede encontrar una extensa lista de parámetros que pueden ser modificados y la descripción de éstos. También puede obtener una lista detallada mediante el comando sysctl.	4
2. ¿Con qué opción se muestran todos los parámetros modificables en tiempo de ejecución? Elija dos parámetros y explique, en dos líneas, qué función tienen.	5
3. Realice una copia de seguridad del registro y restaurela, ilustre el proceso con capturas. Para manipular los valores del registro tendremos que utilizar un programa específico para dicho fin, que puede ser invocado desde la línea de comandos.	5
4. ¿Cómo se abre una consola en Windows? ¿Qué comando hay que ejecutar para editar el registro? Muestre su ejecución con capturas de pantalla.	7
5. Las cadenas de caracteres y valores numéricos tienen distintos tipos. Busque en la documentación de Microsoft y liste todos los tipos de valores.	7
6. Enumere qué elementos se pueden configurar en Apache y en IIS para que Moodle funcione mejor.	7
6.1. Mejora de Moodle con Apache . . . . .	7
6.2. Mejora de Moodle con IIS . . . . .	8
7. Ajuste la compresión en el servidor y analice su comportamiento usando varios valores para el tamaño a de archivo partir del cual comprimir. Para comprobar que está comprimiendo puede usar el navegador o comandos como curl (see url) o lynx. Muestre capturas de pantalla de todo el proceso.	8
8. Usted parte de un SO con ciertos parámetros definidos en la instalación (Práctica 1), ya sabe instalar servicios (Práctica 2) y cómo monitorizarlos (Práctica 3) cuando los somete a cargas (Práctica 4). Al igual que ha visto cómo se puede mejorar un servidor web (Práctica 5 Sección 3.1), elija un servicio (el que usted quiera) y modifique un parámetro para mejorar su comportamiento. (9.b) Monitorice el servicio antes y después de la modificación del parámetro aplicando cargas al sistema (antes y después) mostrando los resultados de la monitorización.	11

<b>9. Cuestiones Opcionales</b>	<b>13</b>
9.1. Cuestión opcional 1: Realice lo mismo que en la cuestión 8 pero para otro servicio. . . . .	13

## Índice de figuras

3.1. Regedit en ejecución. . . . .	5
3.2. Archivo ->Exportar ->Guardar . . . . .	6
3.3. Archivo ->Importar ->Aceptar . . . . .	6
7.1. Inicio ->Ejecutar ->'intermgr' ->Aceptar . . . . .	9
7.2. Conectar con Localhost ->Click en compresión . . . . .	9
7.3. Ajustar valor mínimo ->Aplicar . . . . .	10
7.4. Ejecutando curl,1 . . . . .	10
7.5. Ejecutando curl,2 . . . . .	10
8.1. ab -n 20000 -c 20 http://192.168.1.34/ . . . . .	11
8.2. ab -n 20000 -c 20 http://192.168.1.34/ . . . . .	12
9.1. Vista de los motores MySQL configurados. . . . .	13
9.2. Resultado del test usando MyISAM. . . . .	14
9.3. Resultado del test usando INNODB. . . . .	14

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? En Performance Tuning for Linux Servers puede encontrar una extensa lista de parámetros que pueden ser modificados y la descripción de éstos. También puede obtener una lista detallada mediante el comando sysctl.**

Como dice el enunciado el comando `sysctl` nos permite modificar parámetros para el kernel en tiempo de ejecución. Si queremos ver la lista de parámetros actuales podemos usar `sysctl -a`, a continuación muestro una salida acortada de este comando:

```
kernel.sched_tunable_scaling = 1
kernel.sched_migration_cost = 500000
kernel.sched_nr_migrate = 32
kernel.sched_time_avg = 1000
kernel.sched_shares_window = 10000000
kernel.timer_migration = 1
...
net.ipv6.route.gc_timeout = 60
net.ipv6.route.gc_interval = 30
net.ipv6.route.gc_elasticity = 9
net.ipv6.route.mtu_expires = 600
net.ipv6.route.min_adv_mss = 1220
net.ipv6.route.gc_min_interval_ms = 500
net.ipv6.icmp.ratelimit = 1000
net.ipv6.bindv6only = 0
net.ipv6.ip6frag_secret_interval = 600
net.ipv6.mld_max_msf = 64
net.unix.max_dgram_qlen = 10
abi.vsyscall32 = 1
```

Si queremos realizar cambios permanentes deberemos modificar el archivo ubicado en `/etc/sysctl.conf`, una forma sería por ejemplo:

```
sudo nano /etc/sysctl.conf
```

2. ¿Con qué opción se muestran todos los parámetros modificables en tiempo de ejecución? Elija dos parámetros y explique, en dos líneas, qué función tienen.

Usaremos el siguiente comando: `sysctl -a`<sup>1</sup>

He elegido los dos siguientes parámetros:

**net.ipv4.ip\_forward** = 1, nos permite activar el NAT para poder poner reglas iptables, es muy común usarlo en ataques "man in the middle" para redirigir paquetes (actuar de router nosotros mismos).

**kernel.threads-max** = 1024, esta línea indica el número de procesos máximos que puede ejecutar concurrentemente el kernel.

3. Realice una copia de seguridad del registro y restaurela, ilustre el proceso con capturas. Para manipular los valores del registro tendremos que utilizar un programa específico para dicho fin, que puede ser invocado desde la línea de comandos.

Podemos usar **regedit**, propio de Windows. Adicionalmente podemos usar software de terceros como por ejemplo CCleaner<sup>2</sup>.

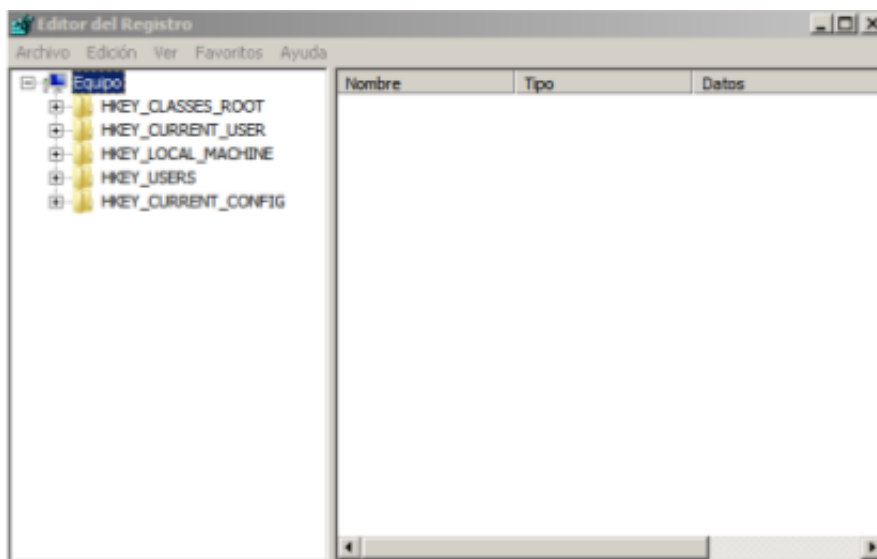


Figura 3.1: Regedit en ejecución.

<sup>1</sup>man sysctl, línea 30

<sup>2</sup><https://www.piriform.com/ccleaner/download>

Exportando:

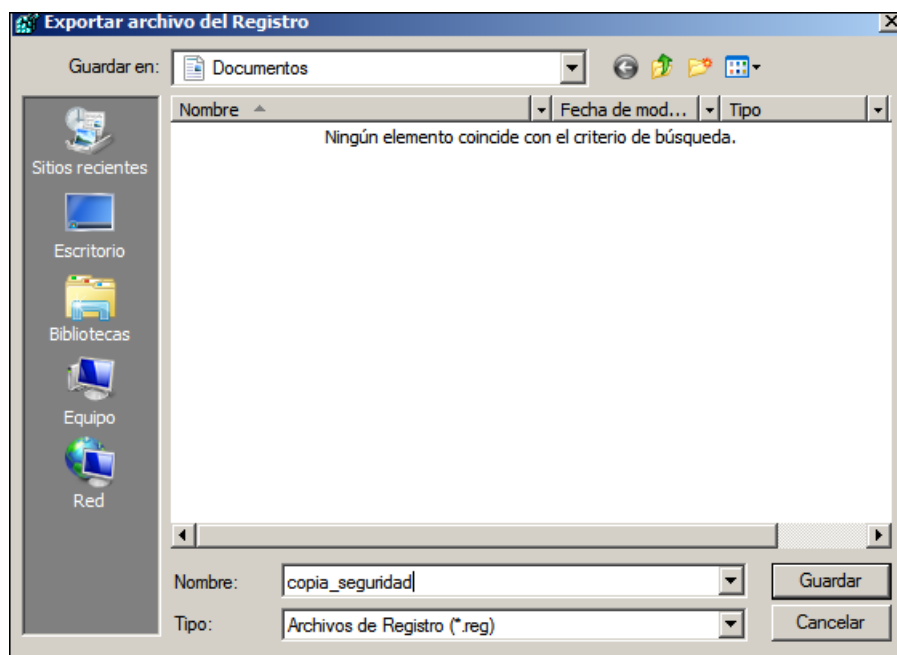


Figura 3.2: Archivo ->Exportar -> Guardar

Importando:

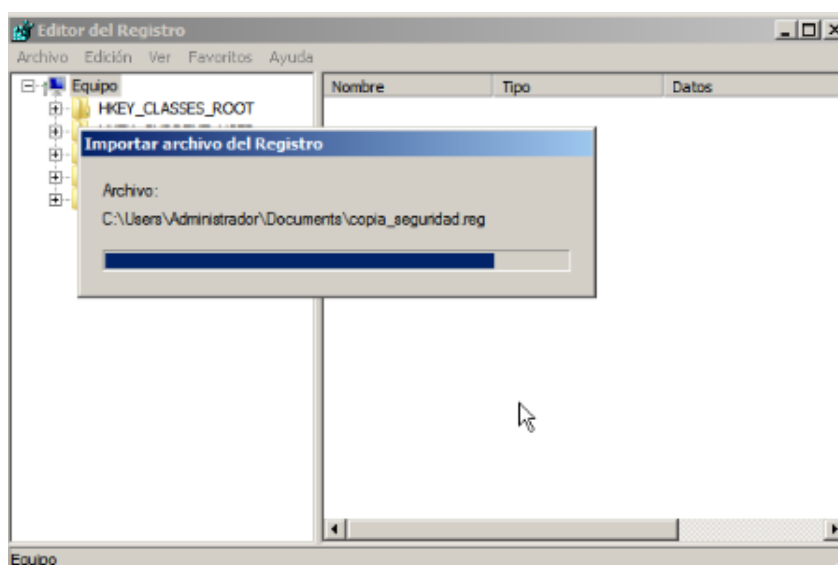


Figura 3.3: Archivo ->Importar -> Aceptar

4. **¿Cómo se abre una consola en Windows? ¿Qué comando hay que ejecutar para editar el registro? Muestre su ejecución con capturas de pantalla.**

Tenemos dos opciones:

1. Pulsar la combinación de teclas: **Tecla de Windows + R**
  2. Abrir el menú **inicio**, click en **ejecutar**, escribir **cmd** y pulsar **enter** o dar **ok**.
- Una vez en la terminal escribimos **regedit** y pulsamos **enter**.

5. **Las cadenas de caracteres y valores numéricos tienen distintos tipos. Busque en la documentación de Microsoft y liste todos los tipos de valores.**

Valor binario	REG_BINARY
Valor DWORD	REG_DWORD
Valor alfanumérico expandible	REG_EXPAND_SZ
Valor de cadena múltiple	REG_MULTI_SZ
Valor de cadena	REG_SZ
Valor binario	REG_RESOURCE_LIST
Valor binario	REG_RESOURCE_REQUIREMENTS_LIST
Valor binario	REG_FULL_RESOURCE_DESCRIPTOR
Ninguna	REG_NONE
Vínculo	REG_LINK
Valor QWORD	REG_QWORD

Información extraída de la página de soporte oficial de Microsoft<sup>3</sup>

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

6.1. **Mejora de Moodle con Apache**

- Modificar el parámetro **MaxClients**:

**MaxClients** = Total available memory \* 80% / Max memory usage of apache process

---

<sup>3</sup><http://support.microsoft.com/kb/256986>

- Reducir el número de módulos que Apache carga en el archivo httpd.conf al mínimo necesario.
- Usar por supuesto la última versión de Apache.
- Reducir el parámetro **MaxRequestPerChild** como mucho a 20-30 (si no perdemos eficiencia)
- Si nuestro Moodle no contiene links a recursos o imágenes podemos establecer **KeepAlive Off**
- Podemos también usar un **Reverse Proxy Server** para hacer caché de los archivos HTML con imágenes.
- Si no usamos el fichero .htaccess, establecer **AllowOverride** a **None**
- Configurar la directiva **DirectoryIndex**  
index.php index.html index.htm - Reducir el valor de **TimeOut** a 30-60 segundos.

## 6.2. Mejora de Moodle con IIS

Para realizar los cambios debemos modificar:

HKLM\SYSTEM\CurrentControlSet\Services\Inetinfo\Parameters

- Cambiar el valor de **ListenBackLog** a un valor entre 2-5.
- Cambiar el valor de **MemCacheSize** ya que IIS usa para la caché el 50 % de la memoria disponible por defecto.
- Cambiar el valor de **MaxCachedFileSize**, el valor por defecto es de 256KB.
- Crear un nuevo valor DWORD llamado ObjectCacheTTL para cambiar la cantidad de tiempo (en milisegundos) que los objetos en la memoria caché se mantienen en la memoria. El valor predeterminado es 30.000ms(30s).

Existen más parámetros modificables, podemos verlos todos en su página oficial<sup>4</sup>.

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

El procedimiento es como sigue a continuación:

---

<sup>4</sup>[https://docs.moodle.org/23/en/Performance\\_recommendations](https://docs.moodle.org/23/en/Performance_recommendations)



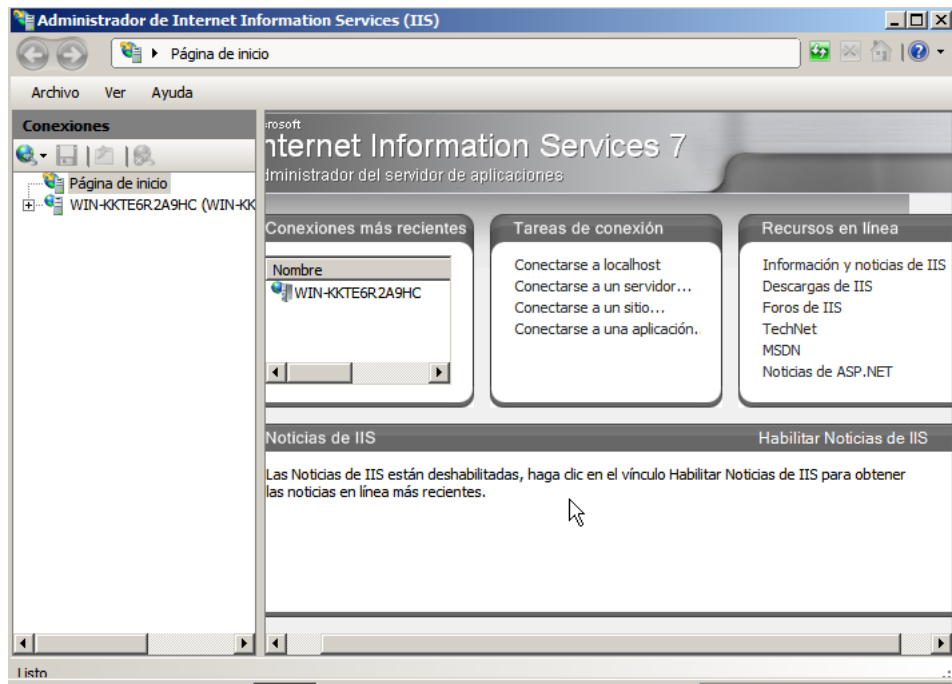


Figura 7.1: Inicio -> Ejecutar -> 'intermgr' -> Aceptar

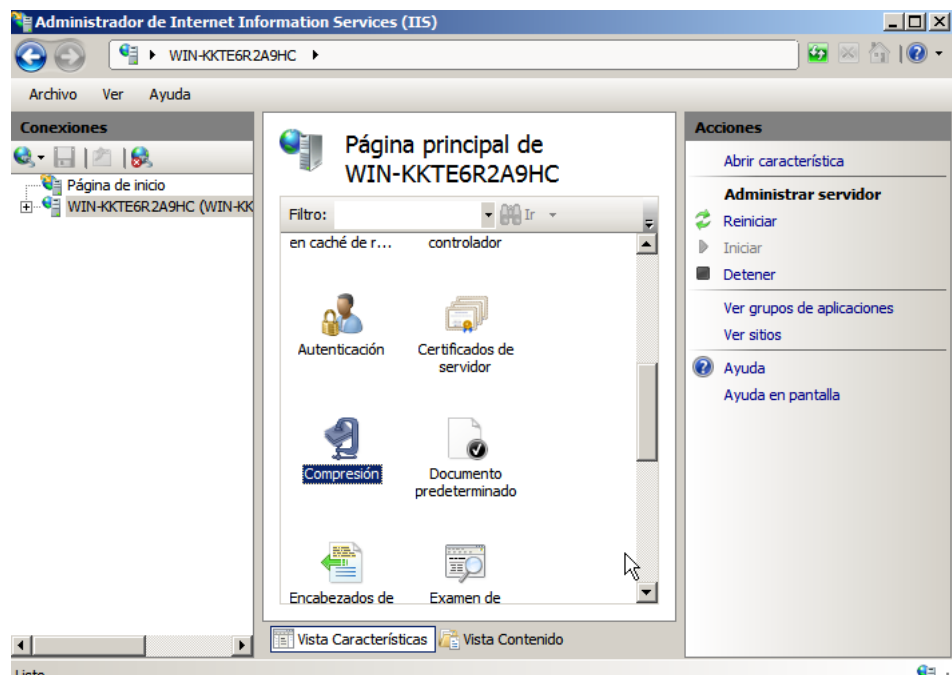


Figura 7.2: Conectar con Localhost -> Click en compresión

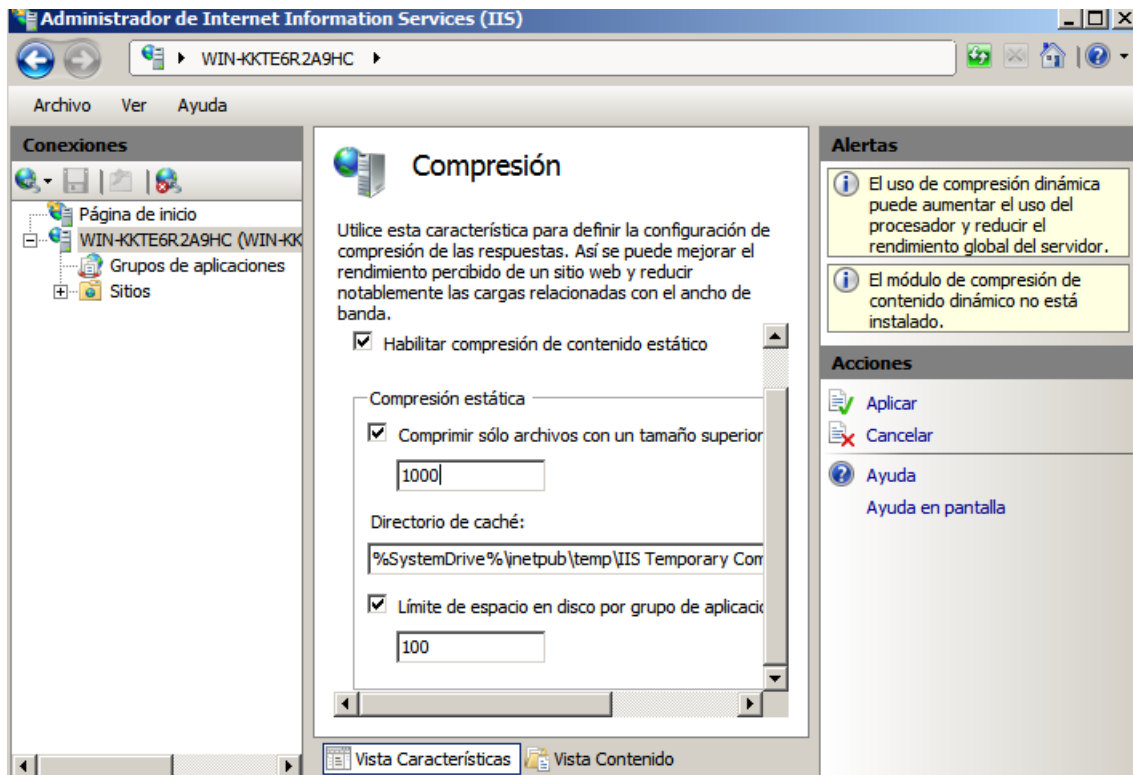


Figura 7.3: Ajustar valor mínimo -> Aplicar

Ahora ya desde el anfitrión(Linux) podemos usar el comando **curl**<sup>5</sup>

```
jesus@jesuspc:~$ curl http://192.168.1.34 --silent -H "Accept-Enconding: gzip,d
eflate" --Write-out "Tam=%{size_download}\n" --output /dev/null
Tam=723
```

Figura 7.4: Ejecutando curl,1

Si bajamos el valor de compresión del IIS a por ejemplo 600B,y volvemos a pasar el Curl:

```
ate" --Write-out "Tamaño de descarga=%{size_download}\n" --output /dev/null
amaño de descarga=457
```

Figura 7.5: Ejecutando curl,2

Como vemos se ha visto reducido el tamaño, la compresión funciona como se esperaba.

<sup>5</sup>man curl

8. Usted parte de un SO con ciertos parámetros definidos en la instalación (Práctica 1), ya sabe instalar servicios (Práctica 2) y cómo monitorizarlos (Práctica 3) cuando los somete a cargas (Práctica 4). Al igual que ha visto cómo se puede mejorar un servidor web (Práctica 5 Sección 3.1), elija un servicio (el que usted quiera) y modifique un parámetro para mejorar su comportamiento. (9.b) Monitorice el servicio antes y después de la modificación del parámetro aplicando cargas al sistema (antes y después) mostrando los resultados de la monitorización.

He elegido modificar algo el servicio de Apache partiendo de lo aprendido en la cuestión 6. Para hacerlo debemos ejecutar como root algún editor de texto y editar el fichero de configuración de Apache. En mi caso:

```
sudo gedit /etc/apache2/apache2.conf
```

Los parámetros modificados son los siguientes:

```
KeepAlive On -> KeepAlive Off
MaxKeepAliveRequests 100 -> MaxKeepAliveRequests 200
KeepAliveTimeout 5 -> KeepAliveTimeout 2
```

Antes de las modificaciones:

```
Server Software:      Apache/2.2.22
Server Hostname:      192.168.1.34
Server Port:          80

Document Path:        /
Document Length:      723 bytes

Concurrency Level:     20
Time taken for tests:  12.455 seconds
Complete requests:     20000
Failed requests:        0
Write errors:           0
Total transferred:     18260000 bytes
HTML transferred:      14460000 bytes
Requests per second:   1605.80 [#/sec] (mean)
Time per request:      12.455 [ms] (mean)
Time per request:      0.623 [ms] (mean, across all concurrent requests)
Transfer rate:         1431.73 [Kbytes/sec] received

Connection Times (ms)
  min  mean[+/-sd] median  max
Connect:    0    0  0.1    0    2
Processing:  1   12  3.5   12   51
Waiting:    1   12  3.0   12   38
Total:      1   12  3.5   12   51

Percentage of the requests served within a certain time (ms)
 50%    12
 66%    13
 75%    14
 80%    14
 90%    16
 95%    18
 98%    22
 99%    25
100%    51 (longest request)
```

Figura 8.1: `ab -n 20000 -c 20 http://192.168.1.34/`

Después de las modificaciones:

```
Server Software:      Apache/2.2.22
Server Hostname:      192.168.1.34
Server Port:          80

Document Path:        /
Document Length:       723 bytes

Concurrency Level:     20
Time taken for tests:   12.211 seconds
Complete requests:     20000
Failed requests:        0
Write errors:           0
Total transferred:     18260000 bytes
HTML transferred:      14460000 bytes
Requests per second:   1637.87 [#/sec] (mean)
Time per request:       12.211 [ms] (mean)
Time per request:       0.611 [ms] (mean, across all concurrent requests)
Transfer rate:          1460.33 [Kbytes/sec] received

Connection Times (ms)
              min    mean[+/-sd] median    max
Connect:        0      0   0.1      0      3
Processing:      1     12   3.3     12     62
Waiting:         1     11   2.9     11     49
Total:           2     12   3.3     12     62

Percentage of the requests served within a certain time (ms)
 50%      12
 66%      13
 75%      14
 80%      14
 90%      15
 95%      17
 98%      21
 99%      24
100%      62 (longest request)
```

Figura 8.2: `ab -n 20000 -c 20 http://192.168.1.34/`

Diferencias encontradas:

Time taken for tests: 12.455s ->12.211s

Requests per second: 1605.80 ->1637.87

Time per request: 0.623s ->0.611s (across all concurrent requests)

Transfer rate: 1431.73KB/s ->1460.33KB/s

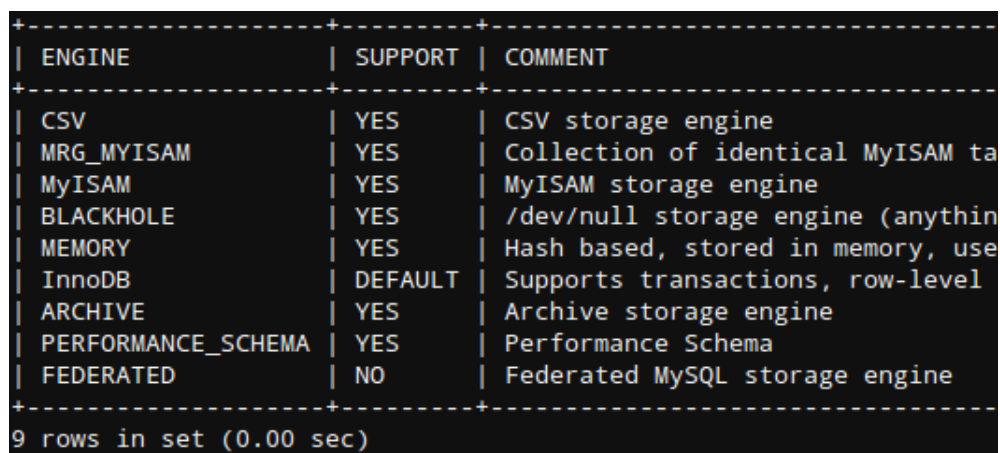
Como vemos ha mejorado la tasa de transferencia y las peticiones por segundo.

## 9. Cuestiones Opcionales

### 9.1. Cuestión opcional 1: Realice lo mismo que en la cuestión 8 pero para otro servicio.

En este caso vamos a suponer que tenemos un servidor humilde en memoria (como es el caso) y queremos optimizar el uso de RAM de todos los servicios posibles, en este caso en concreto vamos a intentar mejorar MySQL.

```
mysql> use information_schema ;
mysql> select * from ENGINES;
```



ENGINE	SUPPORT	COMMENT
CSV	YES	CSV storage engine
MRG_MYISAM	YES	Collection of identical MyISAM ta
MyISAM	YES	MyISAM storage engine
BLACKHOLE	YES	/dev/null storage engine (anythin
MEMORY	YES	Hash based, stored in memory, use
InnoDB	DEFAULT	Supports transactions, row-level
ARCHIVE	YES	Archive storage engine
PERFORMANCE_SCHEMA	YES	Performance Schema
FEDERATED	NO	Federated MySQL storage engine

9 rows in set (0.00 sec)

Figura 9.1: Vista de los motores MySQL configurados.

Voy a comparar el rendimiento usando un benchmark sencillo para los motores de almacenamiento INNODB y MyISAM. Para ello usaré el comando **mysqlslap**<sup>6</sup>:

```
mysqlslap --user=root --auto-generate-sql -vv --concurrency=100
--number-of-queries=10000 --engine=innodb -p
```

```
mysqlslap --user=root --auto-generate-sql -vv --concurrency=100
--number-of-queries=10000 --engine=myisam -p
```

Este test lo que hace es algo similar al que vimos con el comando **ab** para Apache, realiza peticiones concurrentemente al servidor MySQL. Para este estudio realizaremos 10000 peticiones con una concurrencia de 100 conexiones.

---

<sup>6</sup>man mysqlslap

```

jesus@jesuspc:~$ mysqlslap --user=root --auto-generate-sql -vv --concurrency=100 --number-of-queries=10000 --engine=myisam -p
Building Create Statements for Auto
Building Query Statements for Auto
Parsing engines to use.
Enter password:
Starting Concurrency Test
Loading Pre-data
Generating primary key list
Generating stats
Benchmark
    Running for engine myisam
    Average number of seconds to run all queries: 15.209 seconds
    Minimum number of seconds to run all queries: 15.209 seconds
    Maximum number of seconds to run all queries: 15.209 seconds
    Number of clients running queries: 100
    Average number of queries per client: 100

```

Figura 9.2: Resultado del test usando MyISAM.

```

jesus@jesuspc:~$ mysqlslap --user=root --auto-generate-sql -vv --concurrency=100 --number-of-queries=10000 --engine=innodb -p
Building Create Statements for Auto
Building Query Statements for Auto
Parsing engines to use.
Enter password:
Starting Concurrency Test
Loading Pre-data
Generating primary key list
Generating stats
Benchmark
    Running for engine innodb
    Average number of seconds to run all queries: 21.005 seconds
    Minimum number of seconds to run all queries: 21.005 seconds
    Maximum number of seconds to run all queries: 21.005 seconds
    Number of clients running queries: 100
    Average number of queries per client: 100

```

Figura 9.3: Resultado del test usando INNODB.

Como vemos la diferencia es de unos 4.8s, un tiempo considerable. Por tanto he decidido cambiar de motor de almacenamiento, ya que el que viene por defecto con MySQL es INNODB, voy a cambiarlo a MyISAM ya que mi servidor no requiere de las funcionalidades adicionales de INNODB y puede funcionar con MyISAM.

Para ello en el fichero que tenemos en `/etc/mysql/my.cnf` añadiremos las siguientes líneas:

```

default-storage-engine=myisam
skip_innodb

```

Es también destacable que el motor MyISAM consume bastante menos memoria RAM que el motor INNODB dado que tiene menos funcionalidades.