

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

---

## Memoria Práctica 5

---

Pedro Torres Barrilado

13 de enero de 2016

## 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? Elija dos parámetros y explique, en dos líneas, qué función tienen.	4
3. Realice una copia de seguridad del registro y restaurela, ilustre el proceso con capturas.	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.	8
5. Las cadenas de caracteres y valores numéricos tienen distintos tipos. Busque en la documentación de Microsoft y liste todos los tipos de valores.	9
6. Enumere qué elementos se pueden configurar en Apache y en IIS para que Moodle funcione mejor.	10
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.	11
8. Elija un servicio (el que usted quiera) 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.	15

## Índice de figuras

2.1. Resultado de sysctl. . . . .	4
3.1. Ventana de regedit. . . . .	5
3.2. Exportar en regedit. . . . .	6
3.3. Exportar en regedit. . . . .	6
3.4. Configuración al inicio de windows server. . . . .	7
4.1. Ejecutar cmd windows server. . . . .	8
4.2. Comando reg windows server. . . . .	8
7.1. Funcionamiento de ISS. . . . .	11
7.2. Administración de ISS. . . . .	12
7.3. Administración de compresión de ISS. . . . .	12
7.4. Administración de compresión de ISS. . . . .	13
7.5. Resultado de la orden curl a ISS. . . . .	13

7.6.	Resultado de la orden curl a ISS. . . . .	14
7.7.	Resultado de la orden curl a ISS. . . . .	14
8.1.	Funcionamiento de apache2 en la máquina anfitriona. . . . .	15
8.2.	Resultado de ab hacia apache2. . . . .	16
8.3.	Resultado de ab hacia apache2. . . . .	17

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?**

Como dice el la página de manual de la orden `sysctl`<sup>1</sup>, ésta modifica aspectos del kernel en tiempo de ejecución, por lo tanto para poder guardar los cambios para cada vez que se inicie el sistema es necesario modificar el archivo `sysctl` que se encuentra en `/etc/sysctl.conf`. Una vez modificado, para que se puedan aplicar lo cambios ejecutamos la orden `sysctl -p` y ya habría sido modificado.

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.**

Con la orden “`sysctl -a`” podemos ver todos los parámetros modificables que se encuentran en ese momento:

```
kernel.watchdog = 1
kernel.watchdog_thresh = 10
kernel.yama.ptrace_scope = 1
net.core.dev_weight = 64
net.core.message_burst = 10
net.core.message_cost = 5
net.core.netdev_budget = 300
net.core.netdev_max_backlog = 1000
net.core.netdev_tstamp_prequeue = 1
net.core.optmem_max = 10240
net.core.rmem_default = 163840
net.core.rmem_max = 163840
net.core.rps_sock_flow_entries = 0
net.core.somaxconn = 128
net.core.warnings = 1
net.core.wmem_default = 163840
net.core.wmem_max = 163840
net.core.xfrm_acq_expires = 30
net.core.xfrm_aevent_etime = 10
net.core.xfrm_aevent_rseqth = 2
net.core.xfrm_larval_drop = 1
net.ipv4.cipso_cache_bucket_size = 10
net.ipv4.cipso_cache_enable = 1
net.ipv4.cipso_rbm_optfmt = 0
```

Figura 2.1: Resultado de `sysctl`.

---

<sup>1</sup><http://linux.die.net/man/8/sysctl>

He elegido estos dos parámetros:

`net.core.dev_weight`: número máximo de paquetes que puede soportar el kernel en una interrupción NAPI<sup>2</sup>.

`net.core.netdev_max_backlog`: número máximo de paquetes que se pueden guardar en la cola de entrada cuando la interfaz recoge más paquetes de los que el kernel puede procesar.

Documentación de net.<sup>3</sup>

### 3. Realice una copia de seguridad del registro y restaurela, ilustre el proceso con capturas.

Para realizar una copia de seguridad del registro <sup>4</sup>, la podemos realizar con el mismo programa regedit de windows.

-Primeramente abrimos regedit:

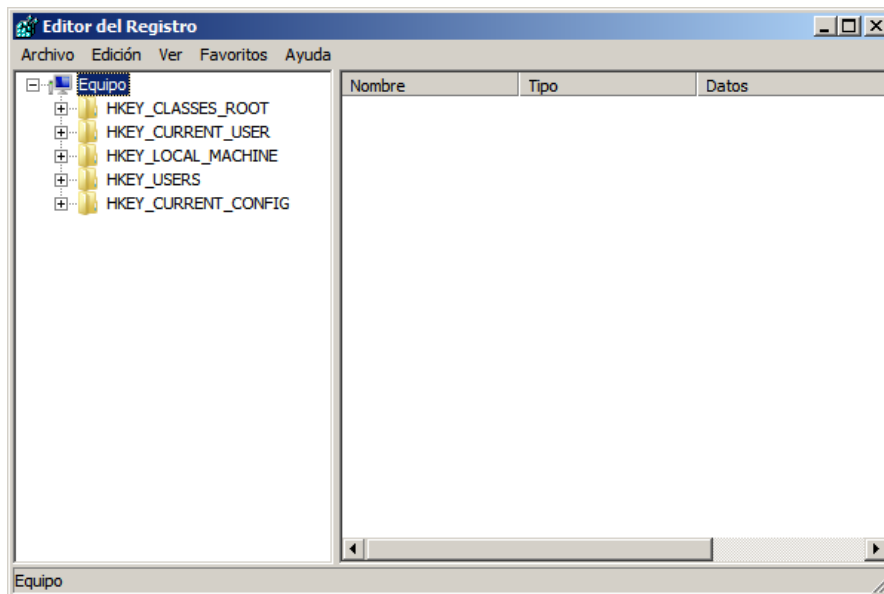


Figura 3.1: Ventana de regedit.

<sup>2</sup>[https://en.wikipedia.org/wiki/New\\_API](https://en.wikipedia.org/wiki/New_API)

<sup>3</sup><https://www.kernel.org/doc/Documentation/sysctl/net.txt>

<sup>4</sup><http://windows.microsoft.com/es-es/windows/back-up-registry#1TC=windows-7>

-Tras abrirlo seleccionamos lo que queremos copiar y con click derecho, exportar e indicamos en la carpeta donde queramos la copia de nuestro registro. (También se puede realizar usando la pestaña archivo):

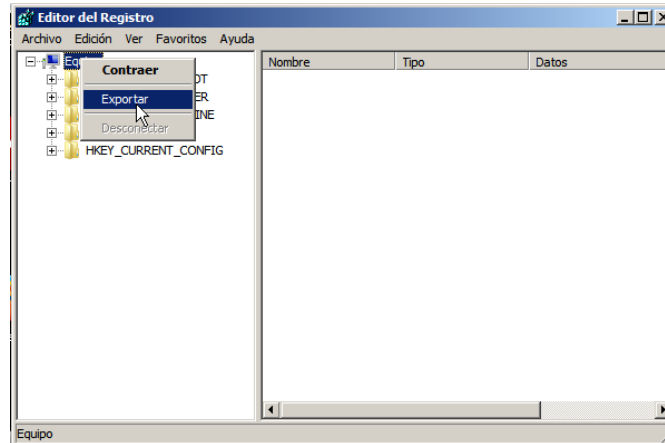


Figura 3.2: Exportar en regedit.

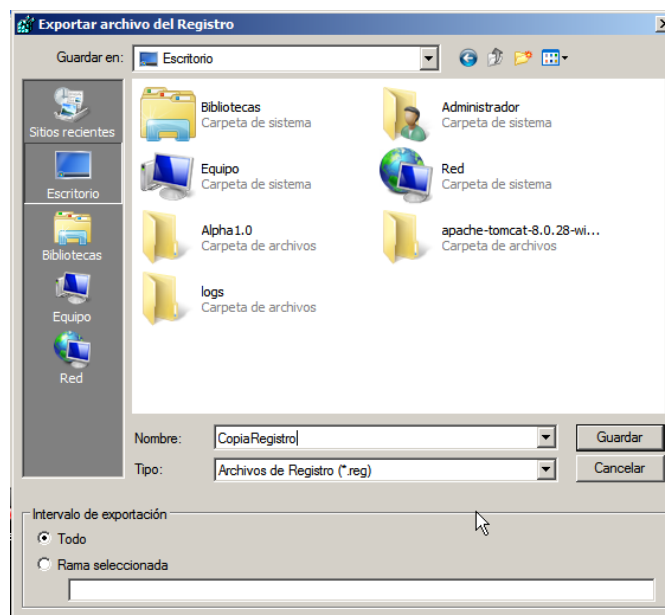


Figura 3.3: Exportar en regedit.

Ya tendríamos una copia de nuestro registro en el sistema.

-Para restaurarlo<sup>5</sup>, apagamos el sistema, y tras arrancar la bios pulsamos F8, lo cual nos muestra una serie de opciones:



Figura 3.4: Configuración al inicio de windows server.

-Y seleccionamos la opción "La última configuración válida conocida(Avanzada)". Tras esto empezaría a ejecutarse nuestro sistema con una configuración correcta.

<sup>5</sup><https://technet.microsoft.com/en-us/library/cc772043.aspx>

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

Para abrir una consola en windows, pulsamos en Inicio, ejecutar y escribimos cmd":

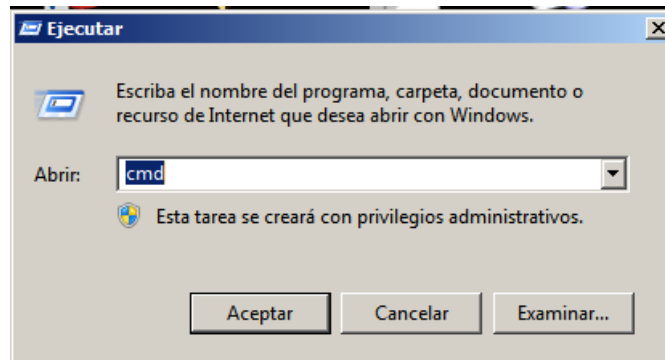


Figura 4.1: Ejecutar cmd windows server.

Si queremos modificar el registro desde la consola de windows, se puede realizar con el comando reg":

```
C:\Users\Administrador>REG /?
REG operación [lista de parámetros]

operación [ QUERY | ADD | DELETE | COPY | 
           SAVE | LOAD | UNLOAD | RESTORE | 
           COMPARE | EXPORT | IMPORT | FLAGS ]

Código devuelto: <excepto en REG COMPARE>
0 - correcto
1 - con error

Para obtener ayuda acerca de una operación, escriba:
REG operación /?

Ejemplos:
REG QUERY /?
REG ADD /?
REG DELETE /?
REG COPY /?
REG SAVE /?
REG RESTORE /?
REG LOAD /?
REG UNLOAD /?
REG COMPARE /?
REG EXPORT /?
REG IMPORT /?
REG FLAGS /?
```

Figura 4.2: Comando reg windows server.

Desde ahí ya se puede realizar la modificación de los registros según lo parámetros que introduzcamos.



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

Los tipos que podemos encontrar en windows son: <sup>6</sup>

**Valor binario: REG\_BINARY:** Datos binarios sin formato. La mayoría de la información sobre componentes de hardware se almacena en forma de datos binarios y se muestra en formato hexadecimal en el Editor del Registro.

**Valor: DWORD: REG\_DWORD:** Datos representados por un número de 4 bytes de longitud (un valor entero de 32 bits).

**Valor alfanumérico expandible: REG\_EXPAND\_SZ:** Cadena de datos de longitud variable. Este tipo de datos incluye variables que se resuelven cuando un programa o servicio utiliza los datos.

**Valor de cadena múltiple: REG\_MULTI\_SZ:** Cadena múltiple. Valores que contienen listas o valores múltiples; este es el formato cuya lectura resulta más sencilla. Las entradas aparecen separadas por espacios, comas u otros signos de puntuación.

**Valor de cadena: REG\_SZ:** Cadena de texto de longitud fija.

**Valor binario: REG\_RESOURCE\_LIST:** Serie de matrices anidadas diseñada para almacenar una lista de recursos utilizados por el controlador de un dispositivo de hardware o uno de los dispositivos físicos que controla.

**Valor binario : REG\_RESOURCE\_REQUIREMENTS\_LIST:** Serie de matrices anidadas diseñadas para almacenar una lista de controladores de dispositivo de posibles recursos de hardware que el controlador, o uno de los dispositivos físicos que controla, pueden utilizar.

**Valor binario : REG\_FULL\_RESOURCE\_DESCRIPTOR:** Serie de matrices anidadas diseñada para almacenar una lista de recursos utilizados por un dispositivo físico de hardware.

**Ninguna: REG\_NONE:** Datos sin ningún tipo en particular. El sistema o una aplicación escribe estos datos en el Registro y los muestra en el Editor del Registro en formato hexadecimal como un valor binario.

**Vínculo: REG\_LINK:** Cadena Unicode que da nombre a un vínculo simbólico.

---

<sup>6</sup><https://support.microsoft.com/es-es/kb/256986>

**Valor: QWORD REG\_QWORD:** Datos representados por un número entero de 64 bytes. Estos datos se muestran en el Editor del Registro como un valor binario.

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

### Mejoras de Apache para moodle:

- Modificar el parámetro MaxClients usando esta fórmula:  $\text{MaxClients} = \text{memoria total disponible} * 80 \%$ . Intentando no poner demasiados clientes debido a que llenaría la RAM.
- Reducir el número de módulos necesarios en httpd.conf para minimizar el uso de memoria.
- Usar la última versión de apache, en este caso Apache2 debido a que está bien optimizado.
- Modificar el parámetro MaxRequestsPerChild mínimo entre 20 y 30.
- Si el servidor tiene mucha carga lo mejor es mantener esta variable KeepAlive en Off(en caso de que moodle contenga imágenes y enlaces) y el parámetro KeepAliveTimeout con un valor entre 2 y 5.
- Si no se quiere modificar KeepAlive, es posible usar un Reverse Proxy server enfrente de el servidor HTML para hacer caché de los archivos con imágenes.
- Si no usamos el archivo .htaccess cambiamos el parámetro AllowOverride a None.
- Configurar correctamente DirectoryIndex, para evitar negociación de contenido entre directorios.
- Dejar HostnameLookups a Off para reducir la latencia de DNS.
- Reducir el valor de la variable TimeOut entre 30 y 60 segundos.
- Caching:(moodle 1.9) a el navegador se le indica que elementos del html los saque de la memoria y no de otra petición al servidor.

### Mejoras de ISS para moodle:

Modificamos los parámetros en este registro:

HKLM\SYSTEM\CurrentControlSet\Services\Inetinfo\Parameters\
---

- Modificamos el valor de ListenBackLog para que se encuentre entre 2 y 5.
- Cambiamos MemCacheSize para ajustar la caché que va a usar ISS para la caché de sus archivos.
- Cambiamos MaxCachedFileSize para ajustar el tamaño máximo de la caché.
- Crear una nueva DWORD llamada ObjectCacheTTL para indicar el tiempo en milisegundos en el que un objeto en la caché se mantiene en memoria.

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.**

Según el manual de windows<sup>7</sup> para cambiar la configuración de la compresión tenemos que:<sup>8</sup>

1-Comprobamos que ISS funciona en nuestra máquina anfitriona probando en el navegador `http://(dirección ip de windows server)`



Figura 7.1: Funcionamiento de ISS.

<sup>7</sup>[https://technet.microsoft.com/en-us/library/cc730629\(v=ws.10\).aspx](https://technet.microsoft.com/en-us/library/cc730629(v=ws.10).aspx)

<sup>8</sup><https://www.iis.net/configreference/system.webserver/urlcompression>

2-Accedemos a la configuración de nuestro servidor: (Pulsando en el icono de la barra de tareas de Administración de servidor, roles y la configuración de ISS)

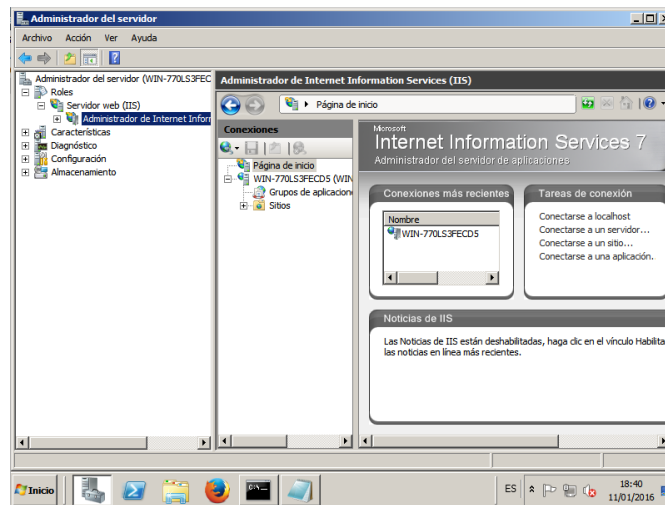


Figura 7.2: Administración de ISS.

3-Pinchamos en nuestra máquina y en la zona central "Página principal", en filtro ponemos compresión y hacemos doble click en lo que nos aparece:

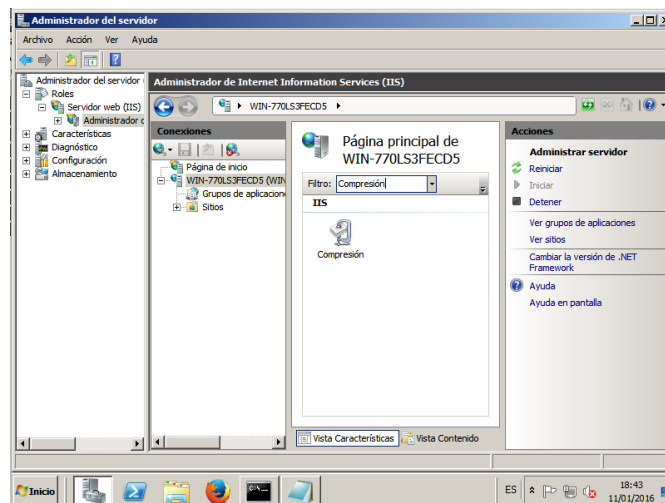


Figura 7.3: Administración de compresión de ISS.

4-Una vez dentro, para habilitar la compresión tenemos que tener activado el tick de "habilitar la compresión de contenido estático", y ya podemos modificar las opciones, en este caso lo que modificaremos será la parte de “Comprimir archivos con un tamaño superior a ” y el número indicado.

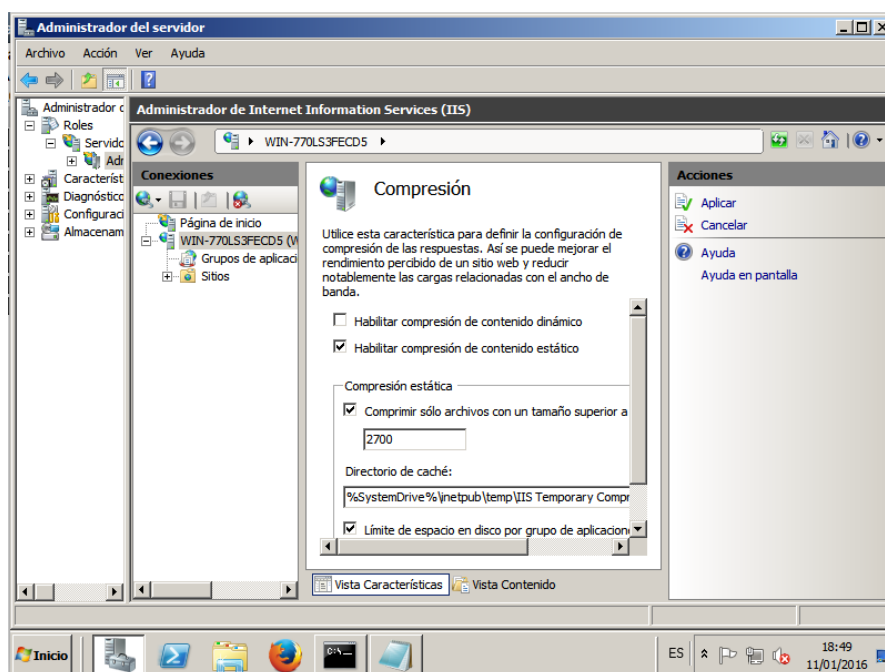


Figura 7.4: Administración de compresión de ISS.

Ahora que ya sabemos modificar los parámetros y activar la compresión, vamos a probar con la orden curl<sup>9</sup> si se realiza la compresión:

-Primeramente desactivamos los ticks de compresión para comprobar el tamaño que nos da la orden curl:

```
MacBook-Pro-de-Pedro:~ pedro$ curl -H "Accept-Encoding: gzip" -I http://192.168.250.129
HTTP/1.1 200 OK
Content-Length: 689
Content-Type: text/html
Last-Modified: Mon, 16 Nov 2015 16:11:35 GMT
Accept-Ranges: bytes
ETag: "7743d778920d11:0"
Server: Microsoft-IIS/7.5
X-Powered-By: ASP.NET
Date: Mon, 11 Jan 2016 17:28:45 GMT
```

Figura 7.5: Resultado de la orden curl a ISS.

**Importante:** Tras cada cambio debemos pulsar aplicar y reiniciar el servidor.

---

<sup>9</sup>Páginademanualdecurl

-Ahora activamos la compresión, pero con el valor que nos venía por defecto, en este caso 2700, por lo que no debería realizar la compresión:

```
MacBook-Pro-de-Pedro:~ pedro$ curl -H "Accept-Encoding: gzip" -I http://192.168.250.129
HTTP/1.1 200 OK
Content-Length: 689
Content-Type: text/html
Last-Modified: Mon, 16 Nov 2015 16:11:35 GMT
Accept-Ranges: bytes
ETag: "7743d778920d11:0"
Server: Microsoft-IIS/7.5
X-Powered-By: ASP.NET
Date: Mon, 11 Jan 2016 17:30:15 GMT
```

Figura 7.6: Resultado de la orden curl a ISS.

Como podemos comprobar el tamaño es de 689 como cuando no teníamos la compresión realizada, por lo cual lo está realizando correctamente.

-Ahora vamos a cambiar el valor de la compresión a 100, por lo que debería comprimirlo ya que el tamaño es de 689.

```
MacBook-Pro-de-Pedro:~ pedro$ curl -H "Accept-Encoding: gzip" -I http://192.168.250.129
HTTP/1.1 200 OK
Content-Length: 457
Content-Type: text/html
Content-Encoding: gzip
Last-Modified: Mon, 16 Nov 2015 16:11:35 GMT
Accept-Ranges: bytes
ETag: "80d56778920d11:0"
Vary: Accept-Encoding
Server: Microsoft-IIS/7.5
X-Powered-By: ASP.NET
Date: Mon, 11 Jan 2016 17:29:11 GMT
```

Figura 7.7: Resultado de la orden curl a ISS.

Se puede observar claramente que el tamaño a cambiado de 689 a 457, por lo cuál ya vemos que está realizando la compresión, pero si queremos asegurarnos, en la línea Vary se ve que pone "Accept-encoding" y en la línea Content encoding "gzip", que son los indicadores de que acepta compresión y cómo lo comprime.

8. Elija un servicio (el que usted quiera) y modifique un parámetro para mejorar su comportamiento. Monitorice el servicio antes y después de la modificación del parámetro aplicando cargas al sistema (antes y después) mostrando los resultados de la monitorización.

El servicio elegido es Apache2 que se encuentra corriendo en una máquina con el sistema operativo instalado Ubuntu Server. Vamos a realizar mejoras accediendo al fichero de configuración de apache y cambiando algunos parámetros. Tras esto probaremos si se ha realizado mejora o no con el apache benchmark debido a que es muy sencillo de utilizar y ofrece la información adecuada para comprobar si se realiza o no mejora. Comprobamos que nuestro servicio funciona en nuestra máquina anfitriona:

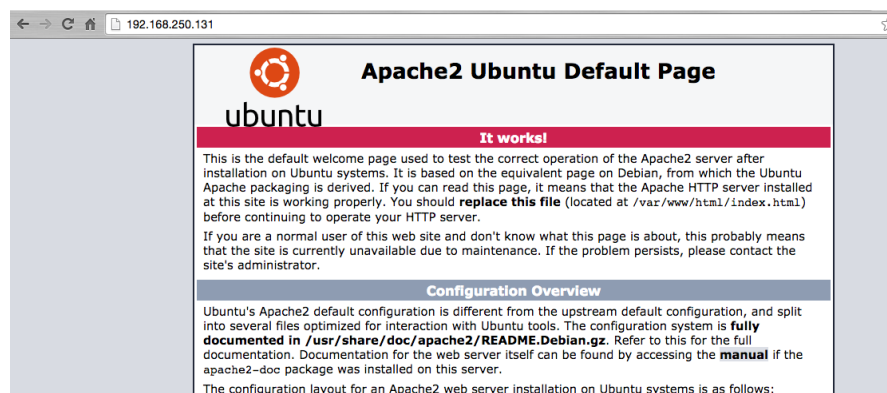


Figura 8.1: Funcionamiento de apache2 en la máquina anfitriona.

Una vez comprobado que funciona, realizamos un test ab con mil conexiones totales y 100 conexiones simultáneas cada vez hacia la máquina corriendo el servicio apache:

```
MacBook-Pro-de-Pedro:~ pedro$ ab -n 1000 -c 100 http://192.168.250.131/
This is ApacheBench, Version 2.3 <$Revision: 1663405 $>
Copyright 1996 Adam Twiss, Zeus Technology Ltd, http://www.zeustech.net/
Licensed to The Apache Software Foundation, http://www.apache.org/

Benchmarking 192.168.250.131 (be patient)
Completed 100 requests
Completed 200 requests
Completed 300 requests
Completed 400 requests
Completed 500 requests
Completed 600 requests
Completed 700 requests
Completed 800 requests
Completed 900 requests
Completed 1000 requests
Finished 1000 requests

Server Software:      Apache/2.4.7
Server Hostname:      192.168.250.131
Server Port:          80

Document Path:        /
Document Length:      11510 bytes

Concurrency Level:    100
Time taken for tests:  1.073 seconds
Complete requests:    1000
Failed requests:       0
Total transferred:    11783000 bytes
HTML transferred:     11510000 bytes
Requests per second:  931.83 [#/sec] (mean)
Time per request:     107.316 [ms] (mean)
Time per request:     1.073 [ms] (mean, across all concurrent requests)
Transfer rate:        10722.44 [Kbytes/sec] received

Connection Times (ms)
      min    mean[+/-sd] median    max
Connect:    0       1   0.9      0      5
Processing:  7      74  47.4     63    530
Waiting:    6      72  38.6     63    225
Total:     12      75  47.2     64    531
```

Figura 8.2: Resultado de ab hacia apache2.

Como vemos en la captura:

Tiempo total del test = 1.073s

Peticiones por segundo = 1746.92

Tiempo por petición = 1.073ms

Ahora realizaremos mejoras en apache cambiando estos parámetros en /etc/apache2/apache2.conf:

Timeout 20

KeepAlive On

KeepAliveTimeout 3

HostNameLookups Off

MaxKeepAliveRequest 200



Una vez realizados los cambios volvemos a realizar el apache benchmark a nuestro servicio:

```
MacBook-Pro-de-Pedro:~ pedro$ ab -n 1000 -c 100 http://192.168.250.131/
This is ApacheBench, Version 2.3 <$Revision: 1663405 $>
Copyright 1996 Adam Twiss, Zeus Technology Ltd, http://www.zeustech.net/
Licensed to The Apache Software Foundation, http://www.apache.org/

Benchmarking 192.168.250.131 (be patient)
Completed 100 requests
Completed 200 requests
Completed 300 requests
Completed 400 requests
Completed 500 requests
Completed 600 requests
Completed 700 requests
Completed 800 requests
Completed 900 requests
Completed 1000 requests
Finished 1000 requests


Server Software:      Apache/2.4.7
Server Hostname:      192.168.250.131
Server Port:          80

Document Path:        /
Document Length:       11510 bytes

Concurrency Level:     100
Time taken for tests:   0.769 seconds
Complete requests:      1000
Failed requests:         0
Total transferred:      11783000 bytes
HTML transferred:       11510000 bytes
Requests per second:    1300.56 [#/sec] (mean)
Time per request:       76.890 [ms] (mean)
Time per request:       0.769 [ms] (mean, across all concurrent requests)
Transfer rate:          14965.32 [Kbytes/sec] received


Connection Times (ms)
  min   mean[+/-sd] median   max
Connect:    0       1   1.0         0       5
Processing:   9      59  25.2        54     265
Waiting:     8      52   9.5         53     153
Total:      14      60  25.1        54     265
```

Figura 8.3: Resultado de ab hacia apache2.

Y se pueden ver la variación de tiempo en las ejecuciones:

#### **Sin mejora:**

Tiempo total del test = 1.073s  
Petición por segundo = 931.83  
Tiempo por petición = 1.073ms

#### **Con mejora:**

Tiempo total del test = 0.769s  
Petición por segundo = 1300.56  
Tiempo por petición = 0.769ms

Vemos que hay una reducción de casi 0.300 segundos entre ejecuciones y la cantidad de peticiones por segundo que se realizan es mucho mayor que en el caso que no hay ninguna mejora.