

**Ingeniería de Servidores (2016-2017)**  
GRADO EN INGENIERÍA INFORMÁTICA  
UNIVERSIDAD DE GRANADA

---

## Práctica 5. Ajuste del sistema.

---

Manuel Jiménez Molina

20 de enero 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? Elija dos parámetros y explique, en dos líneas, qué función tienen.	7
3. a) Realice una copia de seguridad del registro y restáurela, ilustre el proceso con capturas. b) Abra una ventana mostrando el editor del registro.	9
3.1. a) Realice una copia de seguridad del registro y restáurela, ilustre el proceso con capturas. . . . .	9
3.2. b) Abra una ventana mostrando el editor del registro. . . . .	16
4. Enumere qué elementos se pueden configurar en Apache y en IIS para que Moodle funcione mejor.	17
4.1. Elementos a configurar en Apache para que Moodle funcione mejor . . . . .	18
4.2. Elementos a configurar en IIS para que Moodle funcione mejor . . . . .	20
5. Ajuste la compresión en el servidor y analice su comportamiento usando varios valores para el tamaño de archivo a 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.	21
6. 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. 6.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.	25
7. Opcional 1: Realice lo mismo que en la cuestión 6 pero para otro servicio.	32

## Índice de figuras

1.1. Valor de variable kernel shmmni desde /proc/sys . . . . .	4
1.2. Valor de variable kernel shmmni con comando sysctl . . . . .	4
1.3. Cambiar parámetro shmmni con sysctl . . . . .	4
1.4. Comprobar parámetro shmmni con sysctl . . . . .	5
1.5. Comprobar parámetro shmmni con sysctl tras reiniciar . . . . .	5
1.6. Cambiar parámetro shmmni en /etc/sysctl.conf . . . . .	6
1.7. Cargar configuración por defecto con sysctl . . . . .	6

1.8.	Comprobar parámetro shmmni con sysctl . . . . .	7
1.9.	Comprobar parámetro shmmni con sysctl tras reiniciar . . . . .	7
2.1.	Orden sysctl -a hacia archivo parametros.data . . . . .	7
2.2.	Miramos dentro de parametros.data . . . . .	8
3.1.	Buscar en inicio regedit . . . . .	9
3.2.	Abrir registro en Windows . . . . .	10
3.3.	Exportar registro . . . . .	11
3.4.	Elegir directorio donde exportar registro y terminar . . . . .	12
3.5.	Comprobar la copia del registro en el directorio con PowerShell . . . . .	13
3.6.	Importar registro . . . . .	14
3.7.	Seleccionar archivo para importar el registro . . . . .	15
3.8.	Finalizando restauración del registro . . . . .	16
3.9.	Ventana de editor del registro . . . . .	17
4.1.	Código para cargar páginas en Apache más rápido usando caché . . . . .	20
4.2.	Código para reducir tamaño respuesta HTTP . . . . .	20
5.1.	Abrir administrador de IIS . . . . .	22
5.2.	Buscar compresión de la página principal de IIS . . . . .	23
5.3.	Configurar compresión estática y tamaño de compresión . . . . .	24
5.4.	Comprobación de compresión en cabecera http . . . . .	25
6.1.	Comprobar funcionamiento Nginx . . . . .	26
6.2.	Test inicial (foto 1). Nginx . . . . .	27
6.3.	Test inicial (foto 1). Nginx . . . . .	27
6.4.	Test inicial 1. Nginx . . . . .	28
6.5.	Cambiando parámetros configuración Nginx . . . . .	29
6.6.	Ab.Rendimiento Nginx tras cambiar parámetros (foto 1) . . . . .	30
6.7.	Ab.Rendimiento Nginx tras cambiar parámetros (foto 2) . . . . .	30
6.8.	Top. Rendimiento Nginx tras cambiar parámetros . . . . .	31

## 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?

El comando `sysctl` configura los parámetros del kernel en tiempo de ejecución. Con la opción `-p` carga en `sysctl` configuración de un archivo especificado o por defecto `/etc/sysctl.conf`[8].

Para que los cambios sean persistentes en el kernel tenemos que modificar el archivo `/etc/sysctl.conf` para que los cambios se carguen en cada reinicio[9]. Este archivo contiene valores de `sysctl` (valores del kernel) que pueden ser leídos por `sysctl`, el cual los configura para el kernel[6].

Cambiaremos la variable "shmmni" la cual establece el número máximo de segmentos de memoria compartida del sistema[7]. El valor actual de esta variable es:

- Vista desde el sistema de archivos `/proc/sys/kernel`.

```
[manolo@ubuntu ~] 2016-12-26 14:21:54
$cat /proc/sys/kernel/shmmni
4096
```

Figura 1.1: Valor de variable kernel shmmni desde `/proc/sys`

- Vista usando la orden `sysctl`.

```
sysctl: no se puede crear un stat sobre /proc/
[manolo@ubuntu ~] 2016-12-26 14:23:06
$sudo sysctl kernel.shmmni
kernel.shmmni = 4096
```

Figura 1.2: Valor de variable kernel shmmni con comando `sysctl`

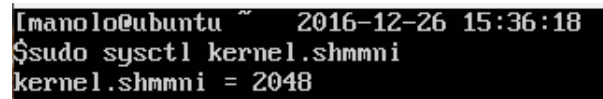
Vamos a comprobar cada caso para ver si los cambios serán persistentes o no:

- Los cambios usando solo `sysctl` no son persistentes:
  - Cambiamos shmmni con `sysctl`.

```
[manolo@ubuntu ~] 2016-12-26 15:35:44
$sudo sysctl -w kernel.shmmni=2048
[sudo] password for manolo:
kernel.shmmni = 2048
```

Figura 1.3: Cambiar parámetro shmmni con `sysctl`

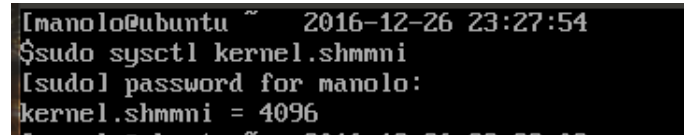
- Comprobamos que se ha cambiado correctamente.



```
[manolo@ubuntu ~ 2016-12-26 15:36:18]
$ sudo sysctl kernel.shmni
kernel.shmni = 2048
```

Figura 1.4: Comprobar parámetro shmni con sysctl

- Reiniciamos y comprobamos su valor.



```
[manolo@ubuntu ~ 2016-12-26 23:27:54]
$ sudo sysctl kernel.shmni
[sudo] password for manolo:
kernel.shmni = 4096
```

Figura 1.5: Comprobar parámetro shmni con sysctl tras reiniciar

- Los cambios modificando el archivo `/etc/sysctl.conf` son persistentes tras un reinicio:
  - Cambiamos parámetro shmni modificando el archivo `/etc/sysctl.conf`.

```
GNU nano 2.2.6 Archivo: /etc/sysctl.conf

#
# /etc/sysctl.conf - Configuration file for setting system variables
# See /etc/sysctl.d/ for additional system variables.
# See sysctl.conf (5) for information.
#

kernel.domainname = example.com
kernel.shmni=2048

# Uncomment the following to stop low-level messages on console
#kernel.printk = 3 4 1 3

#####3
# Functions previously found in netbase
#

# Uncomment the next two lines to enable Spoof protection (reverse-path filter)
# Turn on Source Address Verification in all interfaces to
# prevent some spoofing attacks
#net.ipv4.conf.default.rp_filter=1
#net.ipv4.conf.all.rp_filter=1

# Uncomment the next line to enable TCP/IP SYN cookies
# See http://lwn.net/Articles/277146/
# Note: This may impact IPv6 TCP sessions too
#net.ipv4.tcp_syncookies=1

# Uncomment the next line to enable packet forwarding for IPv4
#net.ipv4.ip_forward=1

# Uncomment the next line to enable packet forwarding for IPv6
# Enabling this option disables Stateless Address Autoconfiguration
```

Figura 1.6: Cambiar parámetro shmni en /etc/sysctl.conf

- Usamos `sysctl -p` para que los cambios tenga efecto y cargue el archivo correctamente.

```
[manolo@ubuntu ~] 2016-12-26 23:30:56
$ sudo sysctl -p
kernel.shmni = 2048
```

Figura 1.7: Cargar configuración por defecto con `sysctl`

- Comprobamos que el valor está cambiado antes del reinicio.

```
[manolo@ubuntu ~ 2016-12-26 23:31:09
$ sudo sysctl kernel.shmni
kernel.shmni = 2048
```

Figura 1.8: Comprobar parámetro shmni con sysctl

- Reiniciamos y comprobamos que el valor ahora sí sigue cambiado:

```
[manolo@ubuntu ~ 2016-12-26 23:35:26
$ sudo sysctl kernel.shmni
[sudo] password for manolo:
kernel.shmni = 2048
```

Figura 1.9: Comprobar parámetro shmni con sysctl tras reiniciar

## 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.

Para ver todos los parámetros modificables actuales en tiempo de ejecución usamos la orden `sysctl -a` [8]. Como hay muchos parámetros los meteremos en un fichero llamado `parametros.data`.

Metemos los parámetros en el archivo `parametros.data`.

```
[manolo@ubuntu ~ 2016-12-27 10:51:49
$ sudo sysctl -a > parametros.data
sysctl: leyendo clave «net.ipv6.conf.all.stable_secret»
sysctl: leyendo clave «net.ipv6.conf.default.stable_secret»
sysctl: leyendo clave «net.ipv6.conf.eth0.stable_secret»
sysctl: leyendo clave «net.ipv6.conf.eth1.stable_secret»
sysctl: leyendo clave «net.ipv6.conf.lo.stable_secret»
```

Figura 2.1: Orden `sysctl -a` hacia archivo `parametros.data`

Miramos el contenido del archivo `parametros.data`.

```
GNU nano 2.2.6 Archivo: parametros.data

fs.inotify.max_user_instances = 128
fs.inotify.max_user_watches = 8192
fs.lease-break-time = 45
fs.leases-enable = 1
fs.mqueue.msg_default = 10
fs.mqueue.msg_max = 10
fs.mqueue.msgsize_default = 8192
fs.mqueue.msgsize_max = 8192
fs.mqueue.queues_max = 256
fs.nr_open = 1048576
fs.overflowgid = 65534
fs.overflowuid = 65534
fs.pipe-max-size = 1048576
fs.pipe-user-pages-hard = 0
fs.pipe-user-pages-soft = 16384
fs.protected_hardlinks = 1
fs.protected_symlinks = 1
fs.quota.allocated_dquots = 0
fs.quota.cache_hits = 0
fs.quota.drops = 0
fs.quota.free_dquots = 0
fs.quota.lookups = 0
fs.quota.reads = 0
fs.quota.syncs = 0
fs.quota.writes = 0
fs.suid_dumpable = 2
kernel.acct = 4 2 30
kernel.acpi_video_flags = 0
kernel.auto_msgmni = 0
kernel.bootloader_type = 114
kernel.bootloader_version = 2
kernel.cad_pid = 1

^G Ver ayuda  ^O Guardar  ^R Leer fich.  ^Y Pág. ant.  ^K Cortar Texto
^X Salir      ^J Justificar ^W Buscar     ^U Pág. sig.  ^U PegarTxt
```

Figura 2.2: Miramos dentro de parametros.data

Por ejemplo vamos a elegir dos parámetros [12]:

- **kernel.dmesg\_restrict**: Indica si los usuarios no privilegiados no pueden usar dmesg para ver los mensajes del buffer de registro del kernel. Cuando se establece a 0 no hay restricciones. Cuando se establece a 1, los usuarios deben tener CAP\_SYSLOG para usar dmesg.
- **kernel.hung\_task\_panic**: Controla el comportamiento del kernel cuando se detecta una tarea que se queda colgada (o bloqueada para siempre). Este archivo



aparece si CONFIG\_DETECT\_HUNG\_TASK está habilitado.  
Con el valor 0, la operación continuaría (comportamiento predeterminado).  
Con valor 1, entraría en pánico de inmediato.

**3. a) Realice una copia de seguridad del registro y restáurela, ilustre el proceso con capturas. b) Abra una ventana mostrando el editor del registro.**

**3.1. a) Realice una copia de seguridad del registro y restáurela, ilustre el proceso con capturas.**

Podemos encontrar documentación oficial de Microsoft para realizar una copia de seguridad del registro y restaurarla.

Primero crearemos una copia de seguridad del registro para posteriormente restaurarla.

- Pulsamos inicio y escribimos regedit.

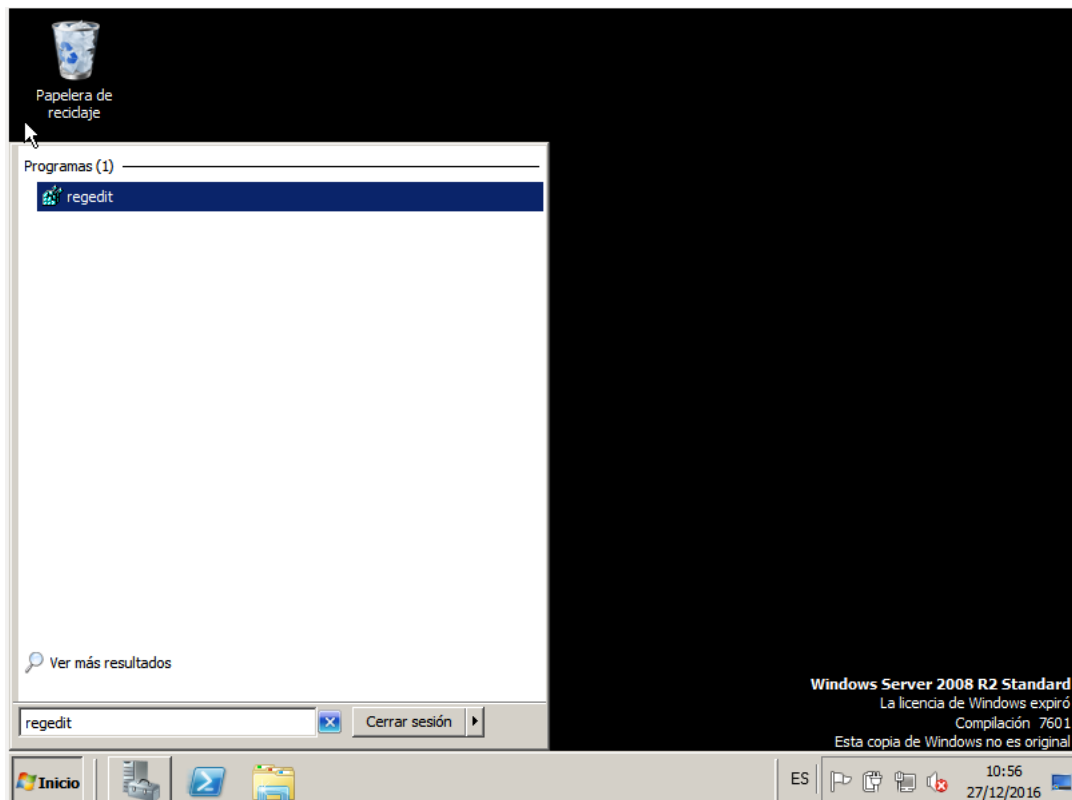


Figura 3.1: Buscar en inicio regedit

- Abrimos el registro pulsando en regedit. Nos aparecerán una lista de directorios que contendrán valores de configuración del sistema.

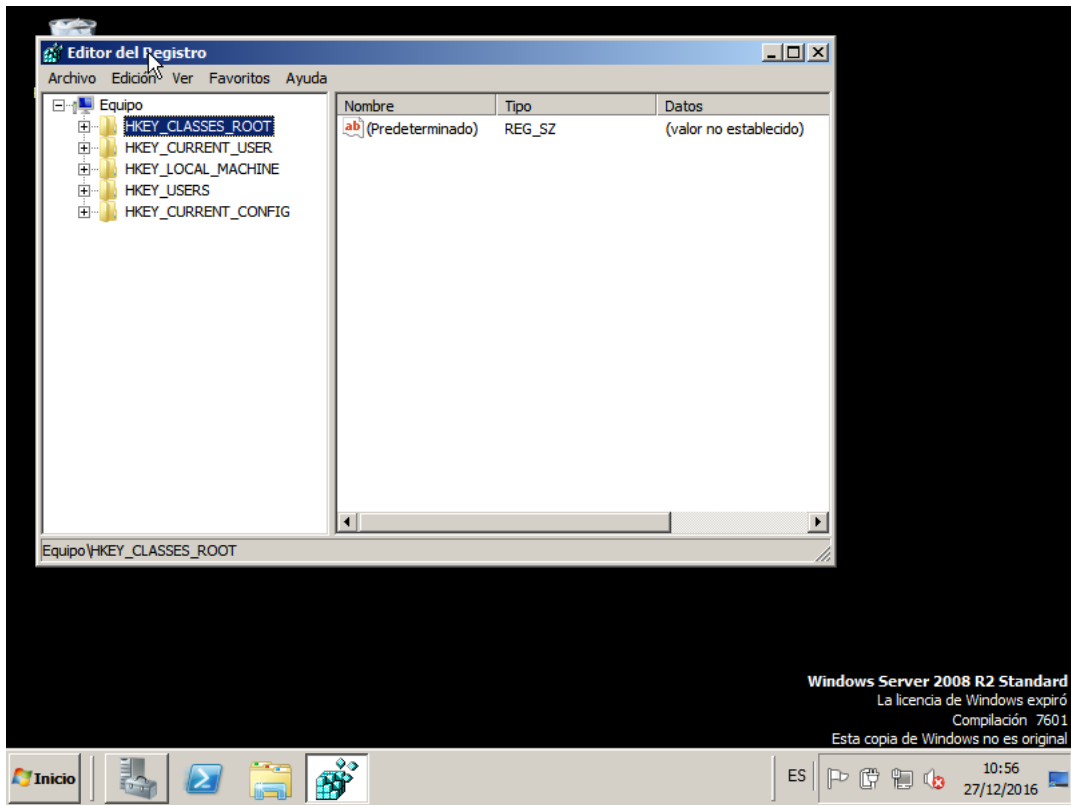


Figura 3.2: Abrir registro en Windows

- Hacemos click derecho sobre `.Equipo` le damos a exportar.

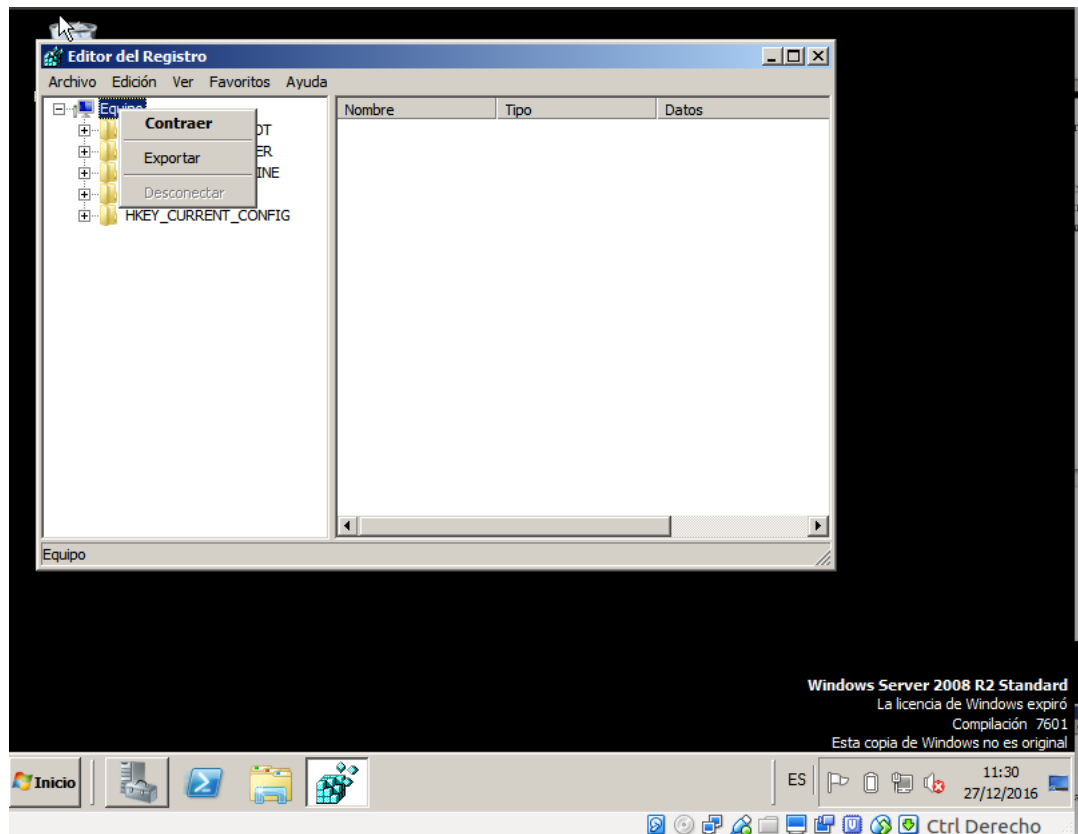


Figura 3.3: Exportar registro

- Elegimos directorio donde guardar registro y terminamos dándole a "Guardar".

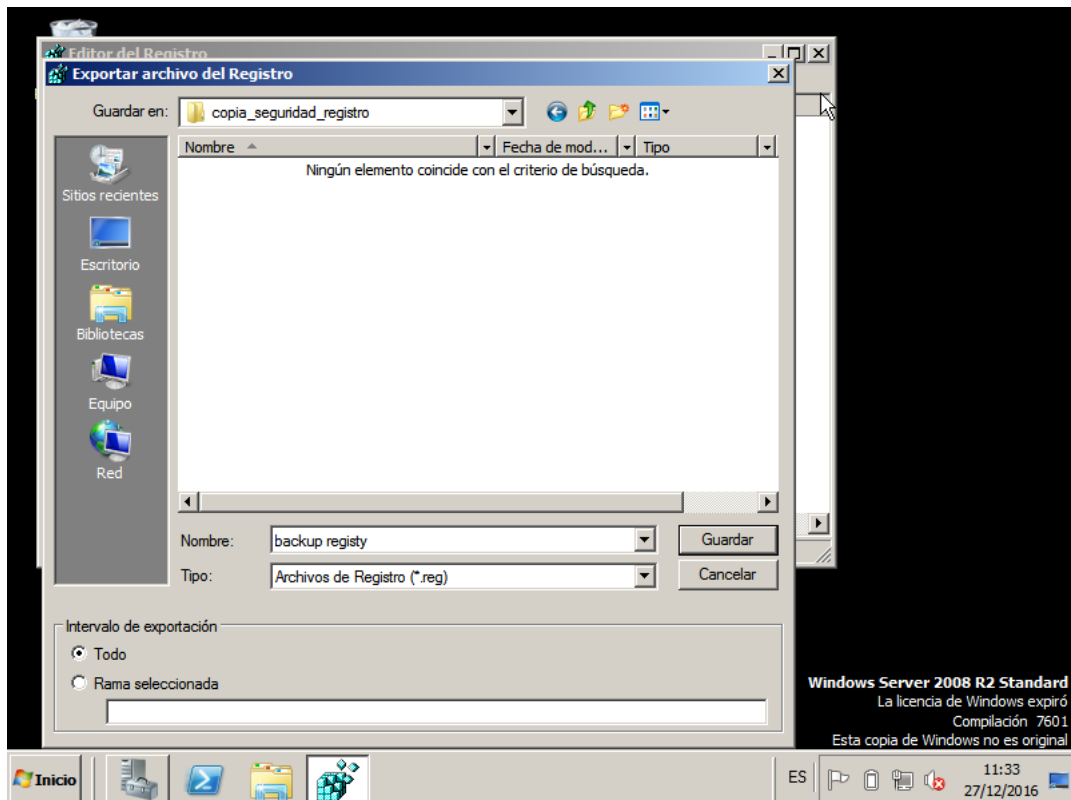


Figura 3.4: Elegir directorio donde exportar registro y terminar

- Comprobamos que ha sido guardada correctamente mirando el directorio con PowerShell.

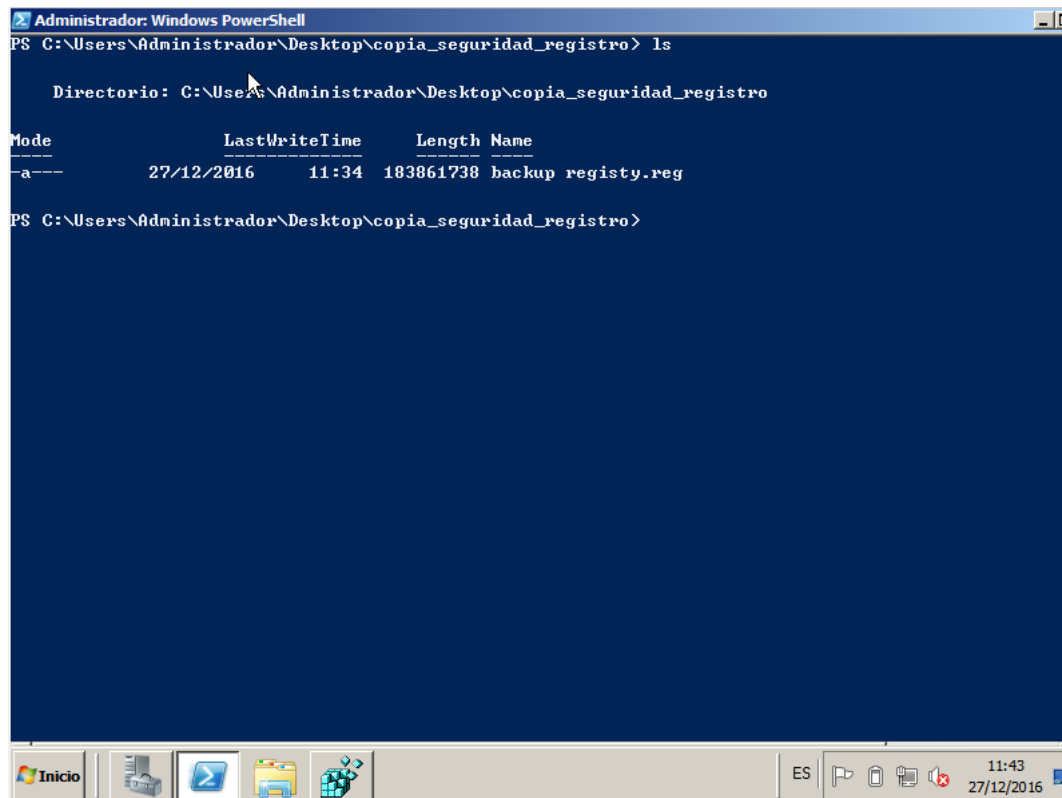


Figura 3.5: Comprobar la copia del registro en el directorio con PowerShell

Ya tenemos la copia de seguridad del registro guardada en una carpeta llamada copia\_seguridad\_registro.

Procedemos a restaurar la copia que hemos realizado. Para ello se hará lo siguiente:

- Volvemos a escribir regedit en inicio para abrir el registro.
- Una vez abierto el registro, nos vamos a su menú desplegable y elegimos "Archivo» Importar".

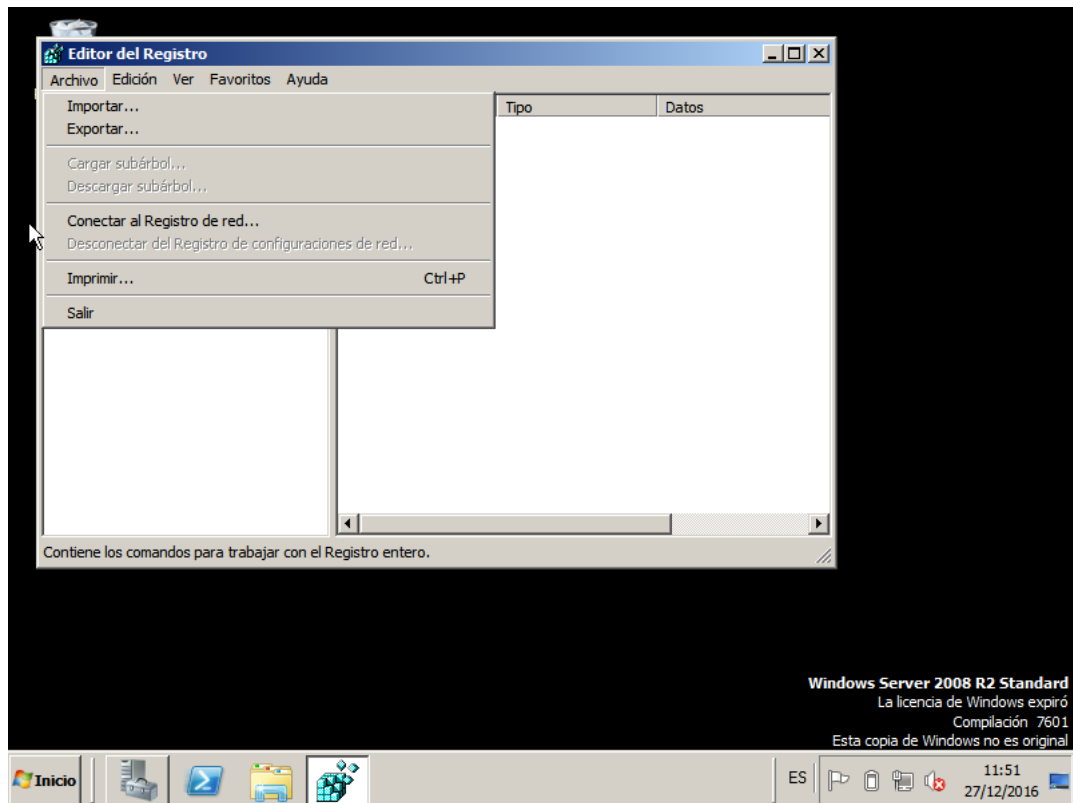


Figura 3.6: Importar registro

- Seleccionamos la carpeta que contenía la copia de seguridad del registro.

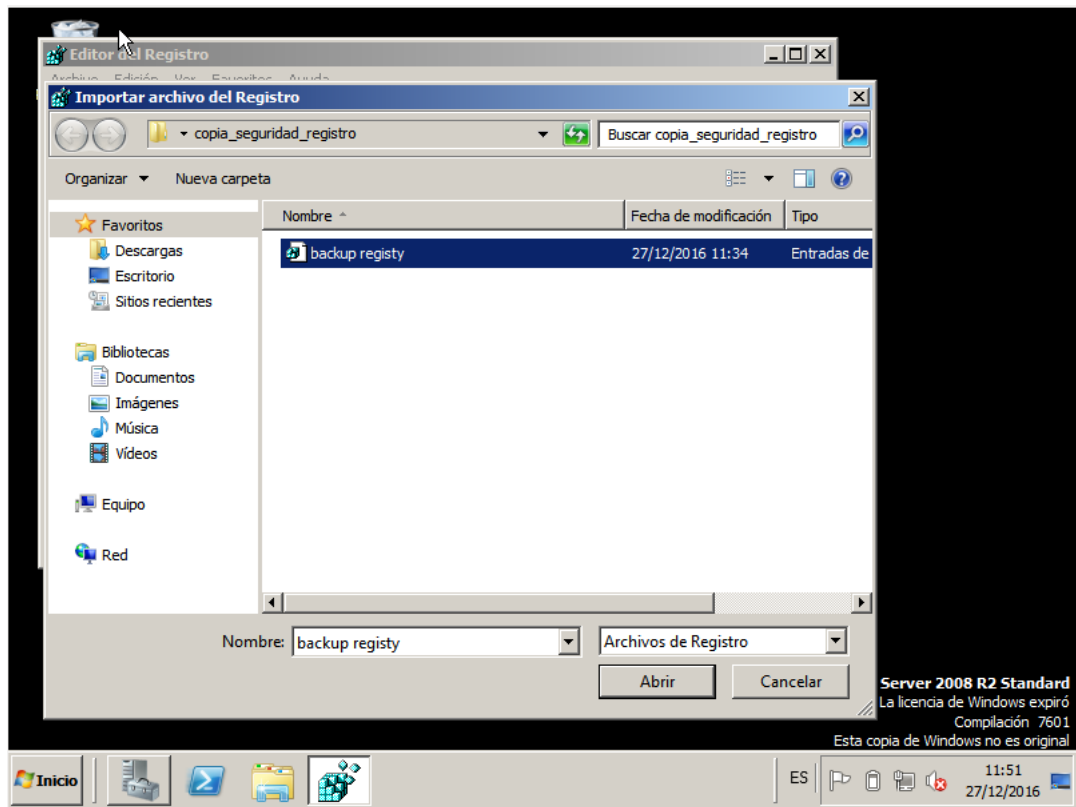


Figura 3.7: Seleccionar archivo para importar el registro

- Acaba el proceso de restauración del registro.

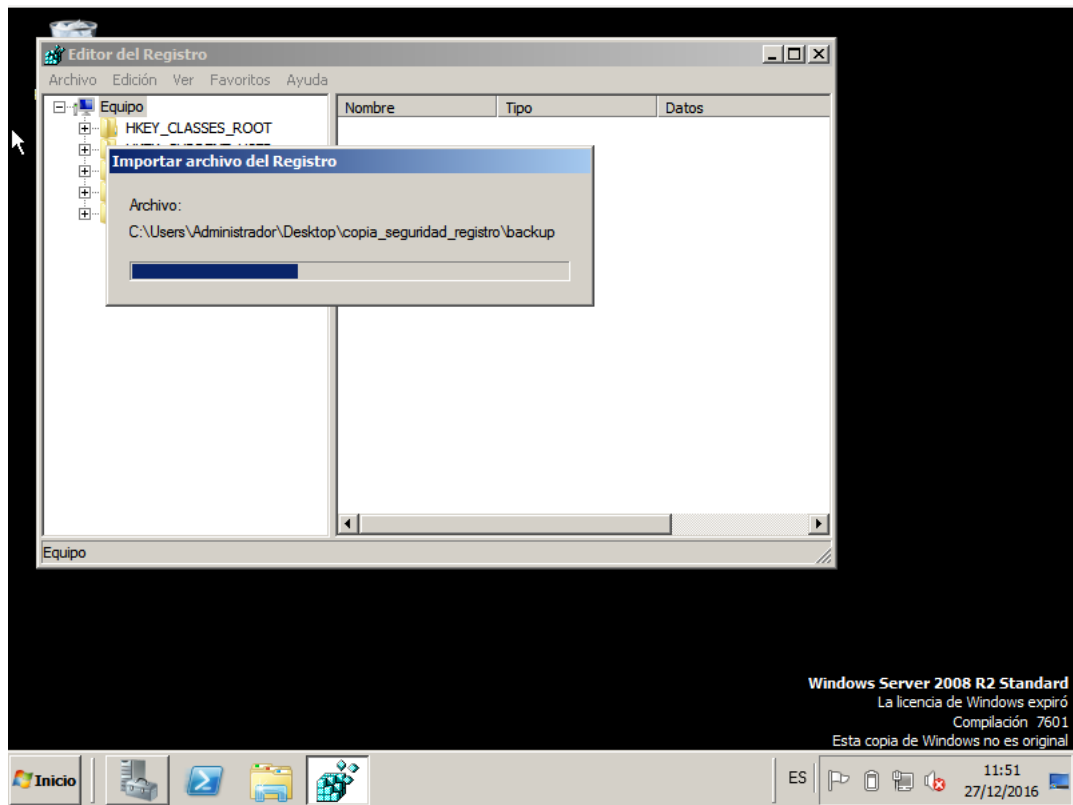


Figura 3.8: Finalizando restauración del registro

### 3.2. b) Abra una ventana mostrando el editor del registro.

Encontramos una referencia de Windows de información de registro para usuarios avanzados[11]. Podemos abrir el editor del registro abriendo una consola del sistema (cmd) y escribiendo regedit.



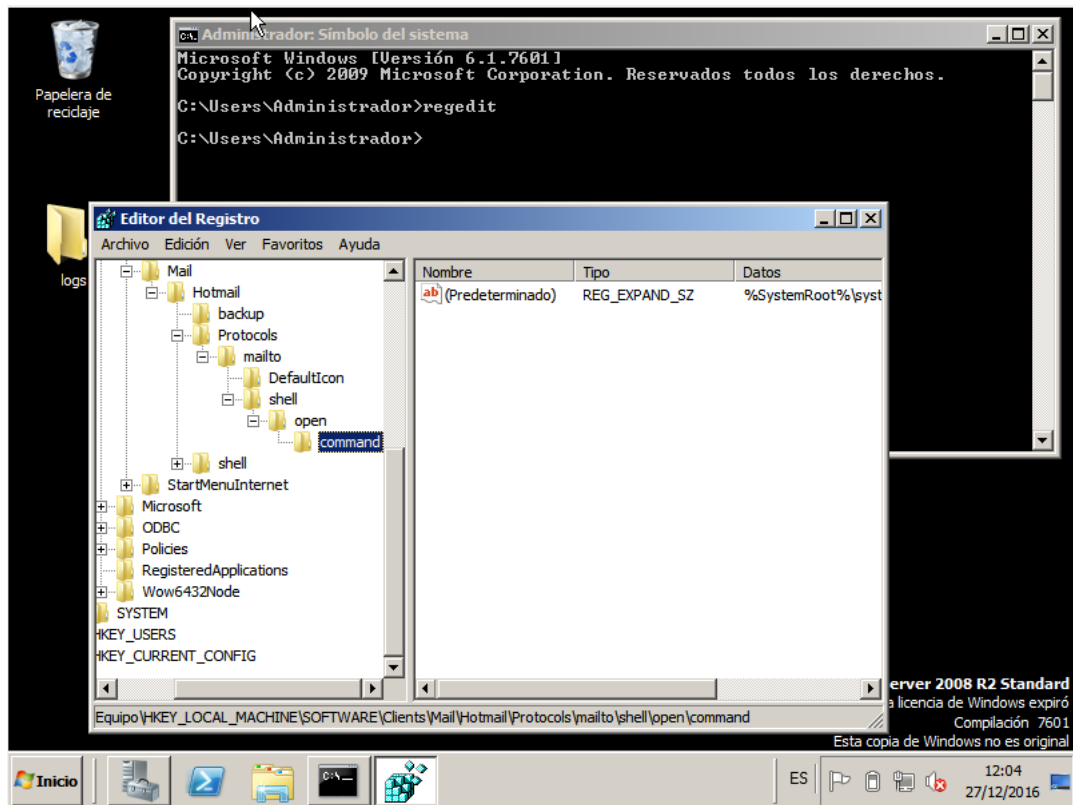


Figura 3.9: Ventana de editor del registro

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

Viendo la referencia de la práctica[10], tenemos una serie de elementos a configurar para mejorar cualquier servidor:

- Escalabilidad:
  - Si es un sitio web grande, separar la web de la base de datos.
  - Si es un sitio web pequeño, tener base de datos y web juntas.
  - Conseguir balanceo de carga, para compartir el trabajo entre varios servidores web. Se deberían usar más de un servidor web, que consultan a una misma base de datos y hacen referencia a un mismo área de archivos.
- Configuración hardware:
  - Aumentar RAM del servidor web.
  - Aumentar memoria principal (para reducir cambio a disco y permitir a tu servidor soportar y gestionar más usuarios).

- Mejorar el rendimiento obteniendo la mejor capacidad de procesador que puedas (dual o dual core por ejemplo).
  - Tener una BIOS moderna para permitir hyperthreading.
  - Usar discos duros SCSI mejor que SATA. Esto se debe a que SATA incrementa la utilización de CPU mientras que SCSI tiene procesadores propios integrados que son usados al tener múltiples dispositivos.
  - Comprar discos duros con un tiempo de búsqueda bajo para aumentar de acceso a informes de Moodle.
  - Establece un tamaño de archivos de intercambio correctamente. Recomendado ponerlo a 4 x RAM física.
  - Usar un sistema de disco RAID.
  - Usar gigabit ethernet para mejorar latencia y rendimiento.
  - Comprobar configuración de la tarjeta de red. Puede obtener mejor rendimiento aumentando el uso de buffers en los descriptors de transmisión y recepción.
- Sistema operativo:
    - Usar Linux (recomendado), Windows o Mac OS X para el sistema operativo del servidor. Los sistemas operativos Unix requieren menos memoria que Mac OS X o Windows Server para realizar las mismas tareas. Además Linux no necesita licencia, pero puede ser difícil de aprender.
    - Comprobar instrucciones específicas del fabricante para optimizar.
  - Rendimiento servidor web:
    - Instalar Firefox y la extensión firebug para saber el tiempo de carga de cada componente de la página.
  - Rendimiento PHP:
    - Usar PHP accelerator para facilitar la carga de CPU, como APC, PHPA, WinCache.
    - Mejoras en rendimiento de lecturas/escrituras alojando las páginas PHP almacenadas en caché en un sistema de archivos TMPFS.

#### 4.1. Elementos a configurar en Apache para que Moodle funcione mejor

Los elementos recomendados a cambiar en Apache para mejorar el funcionamiento de Moodle son los siguientes:

- Si usas Apache en servidor Windows, **utilizar compilación de Apache Lounge**, la cual informa de tener mejoras en rendimiento y estabilidad.
- Establecer directivas **MaxClients** correctamente. Debes usar esta fórmula:

$$MaxClients = \frac{Memoria\ total\ disponible * 80\ \%}{Uso\ de\ memoria\ mximo\ de\ proceso\ Apache}$$

El uso de memoria de Apache para un proceso es normalmente 10MB, pero Moodle podría usar fácilmente hasta 100 MB por proceso, por lo que se divide la memoria disponible en MB por 100 para obtener una configuración adecuada para MaxClients.

Si estableces MaxClients a más de 256, tienes que establecer la directiva ServerLimit.

- **Reducir el número de módulos** que carga Apache en el archivo httpd.conf con el fin de reducir la memoria necesaria.
- Usar la **última versión de Apache**.
- Para sistemas Unix/Linux, podría reducir **MaxRequestPerChild** en httpd.conf hasta 20-30.
- Para un servidor muy cargado, puede desactivar **KeepAlive** (si sus páginas de Moodle no contienen enlaces a recursos o imágenes subidas) o bajando **KeepAliveTimeout** entre 2 y 5. Cuanto más grande sea este valor, más procesos del servidor estarán esperando conexiones que seguramente inactivas. Para saber realmente el valor de KeepAliveTimeout tendríamos que observar el tiempo que tardan los usuarios en descargar una página.
- Como alternativa a KeepAlive desactivado, podemos **configurar un servidor proxy** para almacenar en caché archivos HTML con imágenes.
- Si no usas el archivo .htaccess, puedes establecer la variable **AllowOverride** a nada para evitar búsquedas de .htaccess.
- Evitar negociación de contenido estableciendo **DirectoryIndex** correctamente.
- Si no estás desarrollando en el servidor, pon **ExtendedStatus Off** y deshabilita mod\_info y mod\_status.
- Deja **HostnameLookups Off** para reducir latencia DNS.
- Considerar reducir el valor de **Timeout** entre 30-60 segundos.
- Configurar Apache para que cargue las páginas mucho más rápido **especificando que el navegador debe almacenar en caché elementos** como imágenes y reutilizarlos en la memoria local en vez de pedirlos cada vez que se solicita la página. Son 2 pasos:
  - Instalar y habilitar **mod\_expires**.

- Agregar este código al archivo de configuración del servidor virtual dentro de la sección <directorio> para el directorio root (o con el archivo .htaccess si AllowOverrides está On):

```
<IfModule mod_expires.c>
ExpiresActive On
ExpiresDefault "access plus 1 seconds"
ExpiresByType text/html "access plus 1 seconds"
ExpiresByType image/gif "access plus 1 week"
ExpiresByType image/jpeg "access plus 1 week"
ExpiresByType image/png "access plus 1 week"
ExpiresByType text/css "access plus 1 week"
ExpiresByType text/javascript "access plus 1 week"
ExpiresByType application/x-javascript "access plu
s 1 week"
ExpiresByType text/xml "access plus 1 seconds"
</IfModule>
```

Figura 4.1: Código para cargar páginas en Apache más rápido usando caché

Con esto todo permanece en caché nos HTML y XML, que cambian dinámicamente.

La **compresión** reduce los tiempos de respuesta reduciendo el tamaño de la respuesta HTTP. Para ello:

- Instalar y habilitar mod\_deflate.
- Agregar este código al archivo de configuración del servidor virtual dentro de la sección <directorio> para el directorio root (o con el archivo .htaccess si AllowOverrides está On):

```
<ifModule mod_deflate.c>
    AddOutputFilterByType DEFLATE text/html text/plain text/xml
</ifmodule>
```

Figura 4.2: Código para reducir tamaño respuesta HTTP

## 4.2. Elementos a configurar en IIS para que Moodle funcione mejor

Los elementos recomendados a cambiar en IIS para que Moodle funcione mejor son los siguientes:

- El equivalente a KeepAliveTimeout es **ListenBackLog** (localizado en HKLM/SYSTEM/Current Cont. Establecer entre 2 y 5.
- Ajustar **MemCacheSize** a la cantidad de memoria (Mb) que ISS usará para archivos en caché.
- Cambiar el valor de **MaxCachedFileSize** para establecer el tamaño máximo de un archivo que está almacenado en caché en la caché de archivos en bytes.

- Crear un nuevo DWORD llamado **ObjectCacheTTL** para cambiar la duración del tiempo (milisegundos) que los objetos de caché están en memoria.

## **5. Ajuste la compresión en el servidor y analice su comportamiento usando varios valores para el tamaño de archivo a 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.**

Los cuellos de botella suelen estar mayormente en la red. Tenemos que enviar 1MB a un cliente para que pueda ver el contenido de una web, pero si comprimimos ese contenido y se lo enviamos, dicho envío será menor.

Vamos a hacer que ISS comprima las páginas web. Miramos en el enlace proporcionado en el guión de prácticas para ver una guía de IIS de Microsoft[14]. En esta guía nos centraremos en la optimización de rendimiento de IIS[17].

Para optimizar el rendimiento, Microsoft nos ofrece dos características:

- **Compresión:** Para comprimir HTTP para hacer más eficiente el uso de banda ancha y aumentar el rendimiento de sitios web y aplicaciones. Se puede configurar para contenido estático y dinámico.
- **Almacenamiento en caché de la salida:** Permite administrar las reglas de almacenamiento en caché de salida y controlar el almacenamiento de caché del contenido servido.

Nos centraremos en la compresión. Windows nos proporciona las siguientes opciones de compresión[18]:

- Solo archivos estáticos.
- Solo respuestas dinámicas de la aplicación.
- Tanto archivos estáticos como respuestas dinámicas de la aplicación.

La compresión dinámica puede afectar a los recursos de la CPU ya que IIS no almacena en caché las versiones comprimidas de la salida dinámica.

En cambio, las respuestas estáticas comprimidas se pueden almacenar en caché sin degradar los recursos de la CPU.

Los pasos para realizar la compresión son [13, 15, 16]:

- Configurar compresión:

- Abrimos el administrador de IIS.

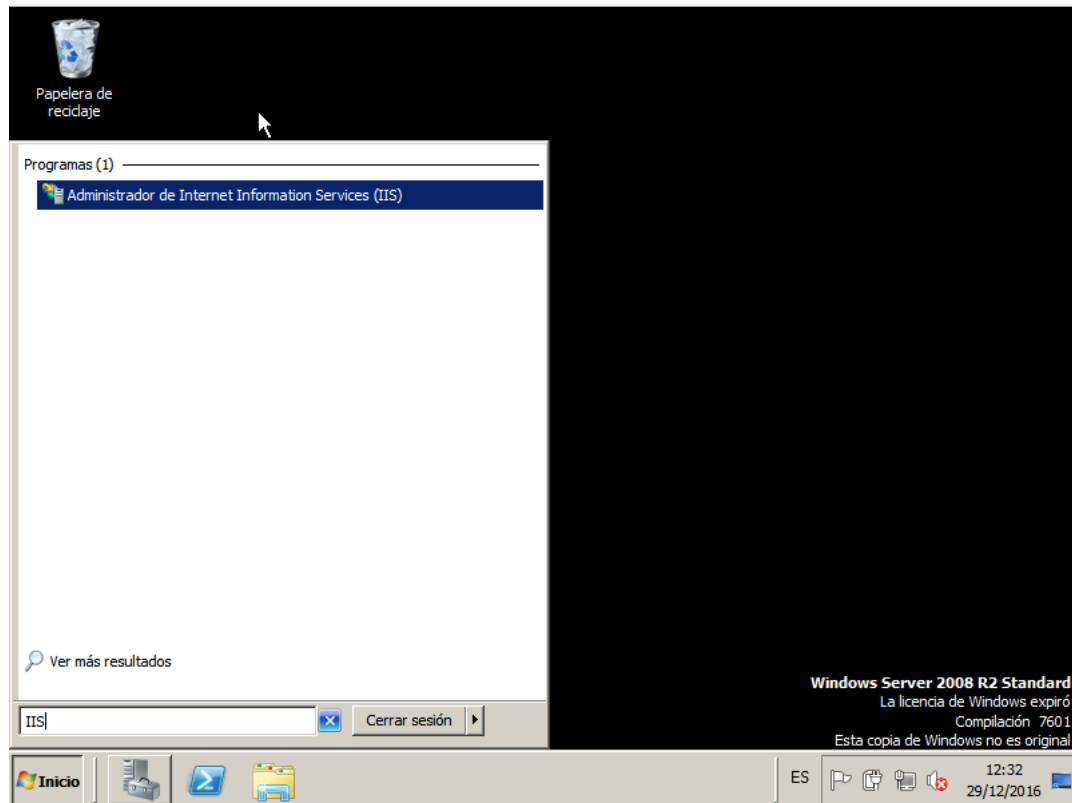


Figura 5.1: Abrir administrador de IIS

- Seleccionamos en el menú de la izquierda nuestro servidor y en su página principal (pantalla central) ponemos en el filtro de búsqueda la palabra "compresión".

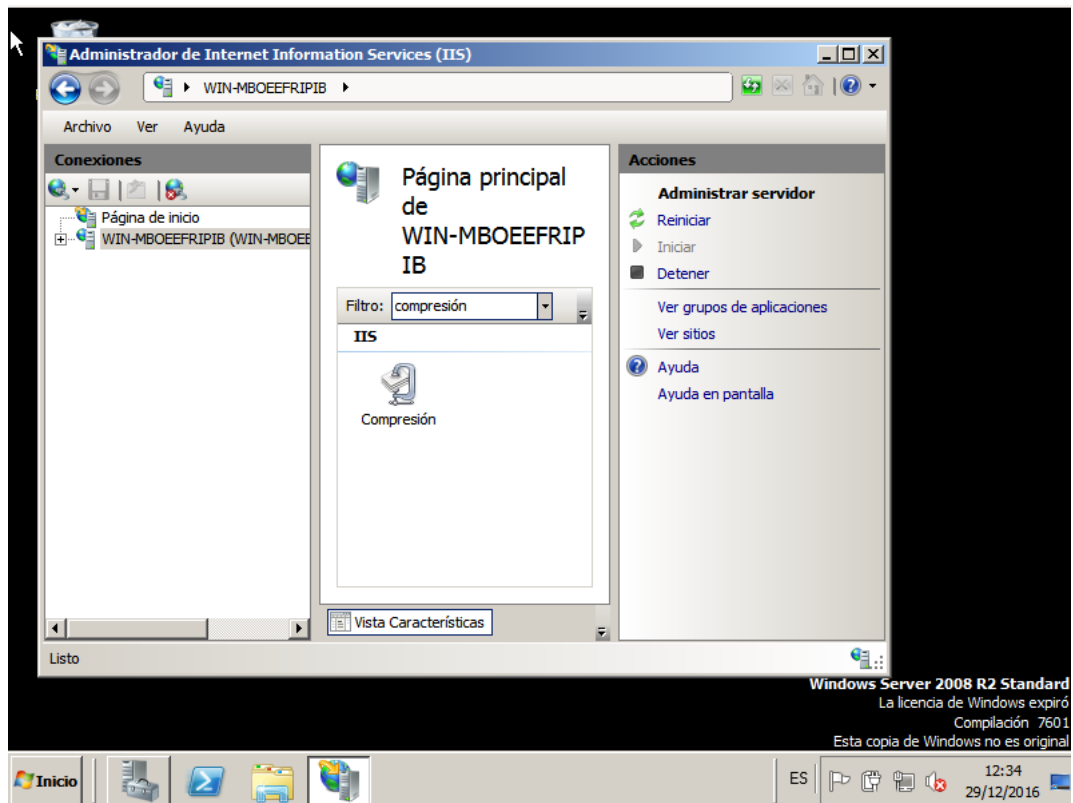


Figura 5.2: Buscar compresión de la página principal de IIS

- Seleccionamos compresión y comprobamos que esté activada la compresión estática y dejamos el tamaño de compresión a 2700 bytes.

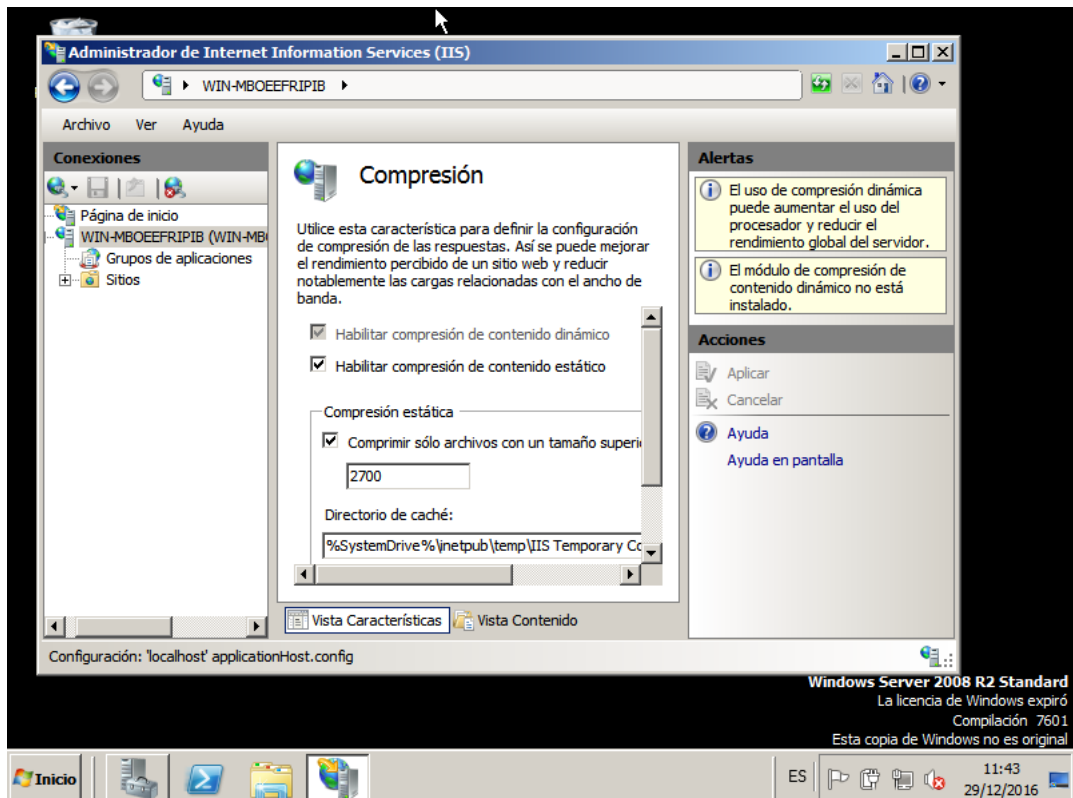


Figura 5.3: Configurar compresión estática y tamaño de compresión

- Comprobación de la compresión. Para ello abriremos en la máquina anfitriona el navegador y miraremos el contenido de la cabecera de respuesta HTTP. La cabecera nos mostrará que existe una variable llamada `.Accept-Encoding`: `"gzip,deflate"` indicando que el contenido fue comprimido.





Figura 5.4: Comprobación de compresión en cabecera http

6. 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. 6.b) Monitoree el servicio antes y después de la modificación del parámetro aplicando cargas al sistema (antes y después) mostrando los resultados de la monitorización.

Vamos a instalar el servidor web Nginx. Este servidor es una alternativa a Apache o también una complementación a este. Podría usarse junto a Apache, usando a Nginx como frontend y Apache como backend.[4, 2]

Combinando ambos, tendremos una especie de dispositivo de descarga de red frente a los servidores Apache, consiguiendo que las conexiones lentas a Internet sean más rápidas y

confiables por el lado del servidor, descargando conexiones keepalive (mensaje entre dispositivos para comprobar que la conexión entre ambos está operativa) de los servidores Apache. Con esto conseguimos un rendimiento de Apache como si estuviera a un "nivel local".

Cabe destacar que Nginx protege a los servidores Apache de vulnerabilidades de picos en el tráfico o ataques de conexión lenta como por ejemplo Slowloris y slowquestprequest.

Vamos a proceder a la instalación de Nginx en Ubuntu Server, la configuración de sus parámetros para mejorar su comportamiento, comprobado las mejoras mediante monitorización del servicio:

- **Instalamos Nginx:**
  - `sudo apt-get install nginx`
- **Configurar firewall para puerto 80:**
  - Comprobamos que el firewall (ufw) permite la conexión con el puerto 80. Sino, tenemos que habilitarla.
- **Configurar red de Ubuntu Server:**
  - Debemos configurar la red para establecer conexión entre la máquina anfitriona y la máquina virtual que tiene Nginx instalado.
- **Comprobar que Nginx funciona correctamente:**
  - Nos aseguramos que funciona Nginx. Probamos a conectarnos a la ip de la máquina virtual para ver su página de inicio.



Figura 6.1: Comprobar funcionamiento Nginx

- **Realizar test inicial:**
  - Vamos a realizar un test de Nginx con Apache Benchmark (ab) desde la máquina anfitriona, monitorizando a la vez con top la máquina anfitriona.

Ejecución de ab:

```
manolo@manolo-K53SC:~$ ab -c 200 -n 25000 http://192.168.56.101/
This is ApacheBench, Version 2.3 <$Revision: 1706008 $>
Copyright 1996 Adam Twiss, Zeus Technology Ltd, http://www.zeustech.net/
Licensed to The Apache Software Foundation, http://www.apache.org/

Benchmarking 192.168.56.101 (be patient)
Completed 2500 requests
Completed 5000 requests
Completed 7500 requests
Completed 10000 requests
Completed 12500 requests
Completed 15000 requests
Completed 17500 requests
Completed 20000 requests
Completed 22500 requests
Completed 25000 requests
Finished 25000 requests


Server Software:      nginx/1.4.6
Server Hostname:      192.168.56.101
Server Port:          80

Document Path:        /
Document Length:      612 bytes

Concurrency Level:    200
Time taken for tests:  13.876 seconds
Complete requests:    25000
Failed requests:       0
Total transferred:    21325000 bytes
HTML transferred:     15300000 bytes
Requests per second:  1801.63 [#/sec] (mean)
Time per request:     111.011 [ms] (mean)
Time per request:     0.555 [ms] (mean, across all concurrent requests)
Transfer rate:        1500.77 [Kbytes/sec] received
```

Figura 6.2: Test inicial (foto 1). Nginx

```
Connection Times (ms)
      min  mean[+/-sd] median   max
Connect:    0    5 106.0      0   7011
Processing: 24    71 538.4     44  13856
Waiting:    24    71 538.4     44  13856
Total:      30    76 575.9     45  13869

Percentage of the requests served within a certain time (ms)
 50%    45
 66%    45
 75%    46
 80%    46
 90%    50
 95%    57
 98%    72
 99%   207
100%  13869 (longest request)
```

Figura 6.3: Test inicial (foto 1). Nginx

Comprobación con ps:

```
lmanolo@ubuntu ~ 2017-01-20 16:34:22
$top | grep nginx
1668 www-data 20 0 86232 4308 2764 R 39,2 0,4 0:09.30 nginx
1667 www-data 20 0 86232 4344 2800 R 22,9 0,4 0:09.26 nginx
1666 www-data 20 0 86232 4308 2764 R 6,3 0,4 0:09.47 nginx
1669 www-data 20 0 86232 4308 2764 R 6,3 0,4 0:08.80 nginx
1666 www-data 20 0 86232 4308 2764 R 26,0 0,4 0:10.25 nginx
1667 www-data 20 0 86232 4344 2800 S 26,0 0,4 0:10.04 nginx
1669 www-data 20 0 86232 4308 2764 R 21,6 0,4 0:09.45 nginx
1668 www-data 20 0 86232 4308 2764 S 20,6 0,4 0:09.92 nginx
1669 www-data 20 0 86232 4308 2764 R 37,5 0,4 0:10.58 nginx
1666 www-data 20 0 86232 4308 2764 R 27,2 0,4 0:11.07 nginx
1667 www-data 20 0 86232 4344 2800 R 17,3 0,4 0:10.56 nginx
1668 www-data 20 0 86232 4308 2764 R 16,6 0,4 0:10.42 nginx
1666 www-data 20 0 86232 4308 2764 S 4,7 0,4 0:11.21 nginx
1667 www-data 20 0 86232 4344 2800 S 4,7 0,4 0:10.70 nginx
1669 www-data 20 0 86232 4308 2764 S 4,7 0,4 0:10.72 nginx
1668 www-data 20 0 86232 4308 2764 S 4,0 0,4 0:10.54 nginx
1668 www-data 20 0 86232 4308 2764 R 0,7 0,4 0:10.56 nginx
1669 www-data 20 0 86232 4308 2764 S 0,3 0,4 0:10.73 nginx
1666 www-data 20 0 86232 4308 2764 S 0,3 0,4 0:11.22 nginx
1668 www-data 20 0 86232 4308 2764 S 0,3 0,4 0:10.57 nginx
```

Figura 6.4: Test inicial 1. Nginx

Por la respuesta de ab, el tamaño de la página de inicio de Nginx es de 612 bytes. Para un nivel de concurrencia 200, para 25000 peticiones se ha tardado un total de 13.076 segundos.

Por otra parte, viendo el comando top podemos ver como Nginx se llega a saturar un poco la cpu, alcanzando un pico de 39,2%, teniendo valores entre 4,7% y 39,2%.

#### ■ Cambiando elementos para mejor rendimiento en Nginx:

- Vamos a seguir unas recomendaciones de Nginx para mejorar el rendimiento de Nginx[5, 3, 1]. Dentro del archivo de configuración /etc/nginx/nginx.conf cambiaremos los siguientes parámetros:
  - **worker\_processes**: Es el número de procesos de trabajos de Nginx. Lo establecemos a 8.
  - **worker\_connections**: Número máximo de conexiones que cada proceso puede manejar simultáneamente. Lo establecemos dependiendo de la salida del comando ulimit -n (limitación del sistema). Será 1024.
  - **keepalive\_timeout**: Tiempo que una conexión permanece abierta. Estará a 63 por defecto, pero lo establecemos a 3.
  - **multi\_accept**: Se usa para que los 'workers' acepten todas las conexiones nuevas al mismo tiempo. Lo habilitamos.

```
GNU nano 2.2.6 Archivo: /etc/nginx/nginx.conf

user www-data;
worker_processes 8;
pid /run/nginx.pid;

events {
    worker_connections 1024;
    multi_accept on;
}

http {

    ##
    # Basic Settings
    ##

    sendfile on;
    tcp_nopush on;
    tcp_nodelay on;
    keepalive_timeout 3;
    types_hash_max_size 2048;
    # server_tokens off;

    # server_names_hash_bucket_size 64;
    # server_name_in_redirect off;

    include /etc/nginx/mime.types;
    default_type application/octet-stream;

    ##
    # Logging Settings
    ##

    [ 96 líneas leídas ]
    ^G Ver ayuda    ^O Guardar      ^R Leer fich.   ^Y Pág. ant.    ^K Cortar Texto
    ^X Salir        ^J Justificar   ^W Buscar      ^U Pág. sig.    ^U PegarTxt
```

Figura 6.5: Cambiando parámetros configuración Nginx

- Restauramos el servicio Nginx para guardar los cambios:
  - `sudo service nginx restart`
- Comprobamos las mejoras de rendimiento:
  - Respecto a ab:

```

manolo@manolo-K53SC:~$ ab -c 200 -n 25000 http://192.168.56.101/
This is ApacheBench, Version 2.3 <$Revision: 1706008 $>
Copyright 1996 Adam Twiss, Zeus Technology Ltd, http://www.zeustech.net/
Licensed to The Apache Software Foundation, http://www.apache.org/

Benchmarking 192.168.56.101 (be patient)
Completed 2500 requests
Completed 5000 requests
Completed 7500 requests
Completed 10000 requests
Completed 12500 requests
Completed 15000 requests
Completed 17500 requests
Completed 20000 requests
Completed 22500 requests
Completed 25000 requests
Finished 25000 requests


Server Software:      nginx/1.4.6
Server Hostname:      192.168.56.101
Server Port:          80

Document Path:        /
Document Length:      612 bytes

Concurrency Level:    200
Time taken for tests:  6.505 seconds
Complete requests:    25000
Failed requests:       0
Total transferred:    21325000 bytes
HTML transferred:     15300000 bytes
Requests per second:  3843.48 [#/sec] (mean)
Time per request:     52.036 [ms] (mean)
Time per request:     0.260 [ms] (mean, across all concurrent requests)
Transfer rate:        3201.65 [Kbytes/sec] received

```

Figura 6.6: Ab.Rendimiento Nginx tras cambiar parámetros (foto 1)

```

Connection Times (ms)
      min  mean[+/-sd] median   max
Connect:    0      0  15.5      0   998
Processing:  0     51  48.1     37   398
Waiting:    0     51  48.1     36   398
Total:      0     52  50.4     37  1032

Percentage of the requests served within a certain time (ms)
 50%    37
 66%    65
 75%    71
 80%    81
 90%   110
 95%   140
 98%   177
 99%   227
100%  1032 (longest request)

```

Figura 6.7: Ab.Rendimiento Nginx tras cambiar parámetros (foto 2)

- Respecto a top:

```

lmanolo@ubuntu ~ 2017-01-20 20:07:46
$top | grep nginx

```

2413	www-data	20	0	86332	4260	2704 R	4,0	0,4	0:00.74	nginx
2414	www-data	20	0	86332	4348	2704 R	3,7	0,4	0:01.27	nginx
2418	www-data	20	0	86332	4348	2704 R	3,7	0,4	0:00.90	nginx
2419	www-data	20	0	86332	4256	2704 R	3,3	0,4	0:00.90	nginx
2419	www-data	20	0	86332	4348	2704 R	14,3	0,4	0:01.33	nginx
2412	www-data	20	0	86332	4292	2704 R	14,0	0,4	0:01.11	nginx
2413	www-data	20	0	86332	4348	2704 R	14,0	0,4	0:01.16	nginx
2411	www-data	20	0	86332	4348	2704 R	13,6	0,4	0:01.20	nginx
2416	www-data	20	0	86332	4348	2704 R	11,7	0,4	0:01.14	nginx
2418	www-data	20	0	86332	4348	2704 R	11,3	0,4	0:01.24	nginx
2415	www-data	20	0	86332	4264	2704 R	10,7	0,4	0:01.02	nginx
2414	www-data	20	0	86332	4348	2704 R	9,0	0,4	0:01.54	nginx
2411	www-data	20	0	86332	4348	2704 R	12,0	0,4	0:01.56	nginx
2413	www-data	20	0	86332	4348	2704 R	12,0	0,4	0:01.52	nginx
2416	www-data	20	0	86332	4348	2704 R	12,0	0,4	0:01.50	nginx
2418	www-data	20	0	86332	4348	2704 R	12,0	0,4	0:01.60	nginx
2419	www-data	20	0	86332	4348	2704 R	12,0	0,4	0:01.69	nginx
2412	www-data	20	0	86332	4292	2704 R	11,7	0,4	0:01.46	nginx
2414	www-data	20	0	86332	4348	2704 R	11,7	0,4	0:01.89	nginx
2415	www-data	20	0	86332	4264	2704 R	11,7	0,4	0:01.37	nginx
2412	www-data	20	0	86332	4348	2704 S	0,3	0,4	0:01.47	nginx
2413	www-data	20	0	86332	4348	2704 S	0,3	0,4	0:01.53	nginx
2414	www-data	20	0	86332	4348	2704 S	0,3	0,4	0:01.90	nginx
2415	www-data	20	0	86332	4264	2704 S	0,3	0,4	0:01.38	nginx
2419	www-data	20	0	86332	4348	2704 S	0,3	0,4	0:01.70	nginx
2411	www-data	20	0	86332	4348	2704 S	0,3	0,4	0:01.57	nginx
2415	www-data	20	0	86332	4264	2704 S	0,3	0,4	0:01.39	nginx

Figura 6.8: Top. Rendimiento Nginx tras cambiar parámetros

- **Analizando nuevos resultados:** Hemos de decir que Nginx ya venía con valores por defecto que optimizan el servidor, como por ejemplo la compresión.
  - **Mejor tiempo de respuesta.** Con las mejoras realizadas, conseguimos un tiempo de 6.505 segundos frente a los 13.876 segundos sin las mejoras. Serían 7.371 segundos mejorados.
  - El tiempo por petición ha pasado de 0.555 ms a 0.260 ms. Esto supone una mejora de 0.295 ms por petición.
  - **Existe una menor sobrecarga de CPU en el servidor.** Vemos como ahora no se llega si quiera al 20 %, mientras que antes se llegó incluso a 39 %.
  - **Gracias a habilitar la variable multi\_accept, el rendimiento ha sido espectacular** (comprobé muchas veces y tras cambiarlo disminuyó hasta 6 segundos).

Como conclusión decir que la mejora de rendimiento no me la esperaba, pero ha sido en mayor parte gracias a la variable multi\_accept. Las demás se notaban medio segundo o incluso menos. Se consigue tener un servidor menos sobrecargado y capaz de soportar más clientes y de responder de forma más eficiente.

## 7. Opcional 1: Realice lo mismo que en la cuestión 6 pero para otro servicio.

### Referencias

- [1] <https://www.linode.com/docs/websites/nginx/configure-nginx-for-optimized-performance/>, consultado el 20 de Enero de 2017.
- [2] <https://www.nginx.com/>, consultado el 20 de Enero de 2017.
- [3] <https://www.nginx.com/blog/10-tips-for-10x-application-performance/>, consultado el 20 de Enero de 2017.
- [4] <https://www.nginx.com/blog/nginx-vs-apache-our-view/>, consultado el 20 de Enero de 2017.
- [5] <https://www.nginx.com/blog/tuning-nginx/>, consultado el 20 de Enero de 2017.
- [6] <http://man7.org/linux/man-pages/man5/sysctl.conf.5.html>, consultado el 26 de Diciembre de 2016.
- [7] [https://access.redhat.com/documentation/en-US/Red\\_Hat\\_Enterprise\\_Linux/5/html/Tuning\\_and\\_Optimizing\\_Red\\_Hat\\_Enterprise\\_Linux\\_for\\_Oracle\\_9i\\_and\\_10g\\_Databases/sect-Oracle\\_9i\\_and\\_10g\\_Tuning\\_Guide-Setting\\_Shared\\_Memory-Setting\\_SHMMNI\\_Parameter.html](https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Linux/5/html/Tuning_and_Optimizing_Red_Hat_Enterprise_Linux_for_Oracle_9i_and_10g_Databases/sect-Oracle_9i_and_10g_Tuning_Guide-Setting_Shared_Memory-Setting_SHMMNI_Parameter.html), consultado el 26 de Diciembre de 2016.
- [8] <https://linux.die.net/man/8/sysctl>, consultado el 26 de Diciembre de 2016.
- [9] <http://tldp.org/HOWTO/TCP-Keepalive-HOWTO/usingkeepalive.html>, consultado el 26 de Diciembre de 2016.
- [10] [https://docs.moodle.org/23/en/Performance\\_recommendations](https://docs.moodle.org/23/en/Performance_recommendations), consultado el 27 de Diciembre de 2016.
- [11] <https://support.microsoft.com/en-us/kb/256986>, consultado el 27 de Diciembre de 2016.
- [12] <https://www.kernel.org/doc/Documentation/sysctl/kernel.txt>, consultado el 27 de Diciembre de 2016.
- [13] [https://technet.microsoft.com/en-us/library/cc730629\(v=ws.10\).aspx](https://technet.microsoft.com/en-us/library/cc730629(v=ws.10).aspx), consultado el 29 de Diciembre de 2016.
- [14] [https://technet.microsoft.com/en-us/library/cc732976\(v=ws.10\).aspx](https://technet.microsoft.com/en-us/library/cc732976(v=ws.10).aspx), consultado el 29 de Diciembre de 2016.
- [15] [https://technet.microsoft.com/en-us/library/cc753681\(v=ws.10\).aspx](https://technet.microsoft.com/en-us/library/cc753681(v=ws.10).aspx), consultado el 29 de Diciembre de 2016.



- [16] [https://technet.microsoft.com/en-us/library/cc754668\(v=ws.10\).aspx](https://technet.microsoft.com/en-us/library/cc754668(v=ws.10).aspx), consultado el 29 de Diciembre de 2016.
- [17] [https://technet.microsoft.com/en-us/library/cc770381\(v=ws.10\).aspx](https://technet.microsoft.com/en-us/library/cc770381(v=ws.10).aspx), consultado el 29 de Diciembre de 2016.
- [18] [https://technet.microsoft.com/en-us/library/cc771003\(v=ws.10\).aspx](https://technet.microsoft.com/en-us/library/cc771003(v=ws.10).aspx), consultado el 29 de Diciembre de 2016.