

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

Memoria Práctica 5

Marcos Avilés Luque

20 de enero de 2017

Índice

1 Cuestión 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?	5
2 Cuestión 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.	6
3 Cuestión 3.	8
3.1 Realice una copia de seguridad del registro y restáurela, ilustre el proceso con capturas.	8
3.2 Abra una ventana mostrando el editor del registro.	12
4 Cuestión 4. Enumere qué elementos se pueden configurar en Apache y en IIS para que Moodle funcione mejor.	12
5 Cuestión 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.	13
6 Cuestión 6.	19
6.1 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.	19
6.2 Monitorice el servicio antes y después de la modificación del parámetro aplicando cargas al sistema (antes y después) mostrando los resultados de la monitorización.	39

Índice de figuras

1.1. Algunas variables disponibles para modificar que nos muestra la orden 'sysctl -a'	5
1.2. Ejemplos de uso con la orden 'sysctl'	6
1.3. Archivo de configuración 'sysctl.conf'	6
2.1. Parámetro del tamaño del buffer de recepción de socket.	7
2.2. Parámetro 'net.ipv4.conf.default.forwarding' de nuestro kernel.	7
3.1. Editor del Registro de Windows Server.	8

3.2.	Realizando copia de Seguridad del archivo de registro de Windows Server.	9
3.3.	Realizando copia de Seguridad del archivo de registro de Windows Server.	9
3.4.	Copia de seguridad realizada de nuestro archivo de configuración.	10
3.5.	Restaurando archivo de configuración de nuestro Windows Server.	10
3.6.	Restaurando archivo de configuración de nuestro Windows Server.	11
3.7.	Restaurando archivo de configuración de nuestro Windows Server.	11
5.1.	Panel de Administrador del servidor de Windows Server.	14
5.2.	Panel de Administrador del servidor IIS.	15
5.3.	Característica de compresión de nuestro servidor IIS.	15
5.4.	Modificando las características de compresión de nuestro servidor IIS.	16
5.5.	Directorio de nuestro servidor IIS para servir nuestras páginas webs.	17
5.6.	Petición de nuestra página web deseada a través del Windows Server.	18
5.7.	Realizando una petición mediante el comando 'wget'	19
6.1.	Seleccionando idioma para la instalación de Ubuntu Server.	20
6.2.	Seleccionando instalar Ubuntu Server.	20
6.3.	Seleccionando idioma para la instalación del sistema Ubuntu Server.	21
6.4.	Seleccionando nuestra ubicación de nuestro Ubuntu Server.	21
6.5.	Seleccionando la opción 'No' para detectar el teclado para la instalación de Ubuntu Server.	22
6.6.	Seleccionando origen de nuestro teclado.	22
6.7.	Seleccionando la distribución de nuestro teclado para la instalación de Ubuntu Server.	23
6.8.	Definiendo el nombre de nuestra máquina para Ubuntu Server.	23
6.9.	Definiendo el nombre del propietario para nuestro sistema Ubuntu Server.	24
6.10.	Definiendo el nombre del usuario de la nueva cuenta.	24
6.11.	Definiendo la contraseña de nuestro usuario para la nueva cuenta.	25
6.12.	Confirmando la contraseña de nuestro usuario.	25
6.13.	Omitiendo el proceso de cifrar nuestro Sistema Ubuntu Server.	26
6.14.	Asignando el particionado del disco automáticamente	27
6.15.	Seleccionando el disco duro donde deseamos instalar nuestro sistema.	27
6.16.	Aceptando las características del particionado automático.	28
6.17.	Añadir la información del proxy HTTP si es necesario para nuestro sistema.	29
6.18.	Instalación de actualizaciones de seguridad automáticamente.	29
6.19.	No seleccionando ningún servicio extra para nuestro servidor.	30
6.20.	Incluyendo el sistema de arranque a nuestra instalación de Ubuntu Server.	31
6.21.	Finalización de la instalación de Ubuntu Server.	31
6.22.	Primer inicio de nuestro sistema Ubuntu Server.	32
6.23.	Modificando nuestro PROMPT de nuestro servidor.	32
6.24.	Añadiendo una nueva tarjeta de red en nuestra máquina.	33
6.25.	Configuraciones de red y tarjetas disponibles en nuestro sistema.	33
6.26.	Configuraciones de red y tarjetas disponibles en nuestro sistema.	34
6.27.	Modificando el archivo de configuración de interfaces de nuestro sistema.	34
6.28.	Comprobando la configuración de nuestra nueva tarjeta de red.	35

6.29. Comprobando el correcto funcionamiento de nuestra tarjeta de red del servidor.	35
6.30. Instalando el servicio Apache en nuestro servidor Ubuntu.	36
6.31. Comprobando el estado del servicio Apache en nuestro servidor Ubuntu.	36
6.32. Petición de la página de prueba a nuestro servidor Ubuntu Server.	37
6.33. Instalación del servicio SSH en nuestro servidor.	38
6.34. Incluyendo la clave ssh de nuestro sistema anfitrión en nuestro servidor.	38
6.35. Transfiriendo el nuevo archivo html a nuestro servidor Ubuntu.	38
6.36. Solicitud de la nueva página html incluida en nuestro servidor.	39
6.37. Ejecutando el comando ab en nuestro sistema operativo anfitrión y monitorizando con el comando 'top' en el servidor.	40
6.38. Ejecutando el comando ab en nuestro sistema operativo anfitrión y monitorizando con el comando 'top' en el servidor.	40
6.39. Modificando el archivo de configuración 'apache2.conf'.	41
6.40. Reiniciando el servicio Apache2.	41
6.41. Monitorizando la carga del servidor con el comando 'top' mientras se ejecuta la carga de peticiones con el comando 'ab'.	42
6.42. Resultados obtenidos en el test de 'ab'.	42

1. Cuestión 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?.

Podemos consultar el man page del comando 'sysctl' [9], y vemos que este comando nos permite modificar o configurar los parámetros de kernel en tiempo de ejecución. En la referencia anterior o como también nos indica propio guión, y en [2], el comando 'sysctl' los valores de los parámetros que modifiquemos se almacenan en el directorio '/proc/sys', por lo que después de un reinicio del sistema estos valores que hemos modificado no persisten, ya que en cada reinicio se vuelven a cargar los valores definidos en los archivos de configuración de nuestro sistema.

El archivo de configuración de dichos parámetros del kernel se denomina 'sysctl.conf' que lo encontramos en el directorio '/etc/sysctl.conf', modificando los valores de las variables de este archivo, ya si serán valores permanentes en nuestro sistema.

Ya que hemos consultado el man de este comando vamos a ver un poco como podemos utilizar este comando, por ejemplo podemos listar todas las variables disponibles para poder modificar con el comando 'sysctl -a' como vemos en la Figura 1.1.

```
vn.memory_failure_recovery = 1
vn.nin_free_kbytes = 2736
vn.nin_slab_ratio = 5
vn.nin_unmapped_ratio = 1
vn.nmap_min_addr = 65536
vn.nr_hugepages = 0
vn.nr_hugepages_policy = 0
vn.nr_overcommit_hugepages = 0
vn.nr_overcommit_threads = 0
vn.nums_zonelist_order = default
vn.oom_dump_tasks = 1
vn.oom_kill_localhost_task = 0
vn.overcommit_kbytes = 0
vn.overcommit_memory = 0
vn.overcommit_ratio = 50
vn.page-cluster = 3
vn.panic_on_oom = 0
vn.percpu_pagelist_fraction = 0
vn.slab_reclaim_early = 1
vn.swappiness = 60
vn.user_reserve_kbytes = 14524
vn.ufs_cache_pressure = 100
vn.zone_reclaim_node = 0
root@ubuntu:~# sysctl -a
```

Figura 1.1: Algunas variables disponibles para modificar que nos muestra la orden 'sysctl -a'.

También encontramos algunos ejemplos de uso en el propio manual, y ya aprovechamos para comentar que por ejemplo, si queremos modificar el valor de un parámetro debemos de utilizar la opción '-w', que nos permite la modificación de un parámetro concreto. Pero no olvidemos como el objetivo de esta cuestión, que después de un reinicio del sistema, los valores que modifiquemos con esta orden no persistirán. Vemos algunos ejemplos en la Figura 1.2.

```

EXAMPLES
    $sbin/sysctl -a
    $sbin/sysctl -n kernel.hostname
    $sbin/sysctl -u kernel.domainname="example.com"
    $sbin/sysctl -p /etc/sysctl.conf
    $sbin/sysctl -a --pattern forward
    $sbin/sysctl -a --pattern forward$ 
    $sbin/sysctl -a --pattern 'net.ipv4.conf.(eth1|wlan0).arp'
    $sbin/sysctl --system --pattern 'net.ipv6'

DEPRECATED PARAMETERS
    The base_reachable_time and retrans_time are deprecated. The sysctl command does not
    allow changing values of these parameters. Users who insist to use deprecated kernel
    interfaces should values to /proc file system by other means. For example:
        echo 256 > /proc/sys/net/ipv6/neigh/eth0/base_reachable_time

FILES
    /proc/sys
    /etc/sysctl.conf

SEE ALSO
    sysctl.conf(5) regex(7)
root@ubuntu:~$ vi /etc/sysctl.conf

```

Figura 1.2: Ejemplos de uso con la orden 'sysctl'.

Hablaremos más detalladamente del archivo de configuración 'sysctl.conf' a lo largo de este guión, pero veamos un ejemplo del dicho archivo en la Figura 1.3.

```

# /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
# Uncomment the following to stop low-level messages on console
#kernel.printk = 3 4 1 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
# based on Router advertisements for this host
net.ipv6.conf.all.forwarding=1
#####

root@ubuntu:~$ vi /etc/sysctl.conf

```

Figura 1.3: Archivo de configuración 'sysctl.conf'.

2. Cuestión 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.

Como vimos en la cuestión anterior, podemos visualizar todos los parámetros disponibles para modificar con la orden 'sysctl -a', y podíamos ver un ejemplo en la Figura 1.1.

Ahora vamos a elegir dos parámetros y vamos a detallarlos:

- 'net.core.rmem_default' como vemos en [3], es el parámetro correspondiente al tamaño (en bytes) del socket para la recepción de paquetes, es decir, vamos a poner el

ejemplo para aclararnos un poco, la visualización de un vídeo en streaming, cuando solicitamos el servicio de un vídeo, vamos obteniendo los paquetes de dicho vídeo mientras que al mismo tiempo se reproduce dicho vídeo

Como sabemos normalmente los paquetes son obtenidos más rápidamente que la visualización del mismo, por lo que los paquetes se van almacenando en un buffer (una especie de cola"). Y este parámetro configura el tamaño de nuestro buffer.

Por defecto, estos buffer de recepción y envío se ajustan de forma dinámica, aunque nos podemos encontrar que el tamaño de esta cola esté configurada por defecto con un tamaño muy pequeño, y el usuario pueda modificar el tamaño que desee. Como vemos en la Figura 2.1 mi buffer tiene un tamaño definido de 212992 bytes, pero podríamos modificar dicho valor con cualquier otro valor que deseemos. Sólo deberíamos tener en cuenta, como vemos en la referencia, que no podríamos modificar el valor de este buffer superior al valor que contenga 'rmem_max', dando lugar a que si deseamos un valor mayor, deberíamos modificar tambien el valor de 'rmem_max' .

```
root@ubuntu:~# grep net.core.rmem_default
net.core.rmem_default = 212992
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>
root@ubuntu:~# grep net.core.rmem_max
net.core.rmem_max = 212992
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>
root@ubuntu:~#
```

Figura 2.1: Parámetro del tamaño del buffer de recepción de socket.

- Otro parámetro sería 'net.ipv4.conf.default.forwarding', como nos dice [1]. Para situarnos, en primer lugar sabemos que la NAT (Network Address Translation) es la encargada de traducir direcciones de red para cuando se realizan intercambio de paquetes entre varias redes, y la puerta de enlace se encarga de conectar nuestra red local con la red externa (Internet). Por esto queremos decir que podemos convertir nuestro sistema Ubuntu Server en una puerta de enlace gracias a este parámetro, ya que por defecto contendrá el valor '0' y lo podemos activar modificando el valor a '1'. Vemos el ejemplo 10.1 en la referencia y la visualización del parámetro en la Figura 2.2.

```
root@ubuntu:~# grep net.ipv4.conf.default.forwarding
net.ipv4.conf.default.forwarding = 0
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>
root@ubuntu:~#
```

Figura 2.2: Parámetro 'net.ipv4.conf.default.forwarding' de nuestro kernel.

3. Cuestión 3.

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

En [5] encontramos cómo se realiza una copia de seguridad del archivo de configuración o de registro, para nuestro servidor Windows. En este caso viene definido para todos los Sistemas Operativos Windows pero no aparece el sistema Windows Server, pero para nuestro caso es el mismo procedimiento como vemos en esta cuestión.

Comenzamos abriendo una terminal o 'PowerShell' en nuestro sistema y ejecutamos la orden 'regedit.exe' como vemos en la Figura 3.1.

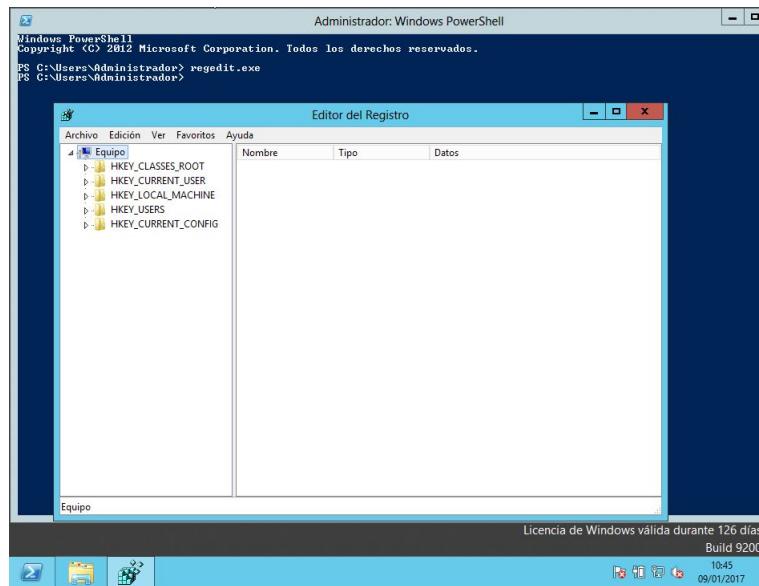


Figura 3.1: Editor del Registro de Windows Server.

Como hemos visto se nos abrirá el Editor del Registro y ahora hacemos click sobre Archivo -> Exportar como vemos en la Figura 3.2.

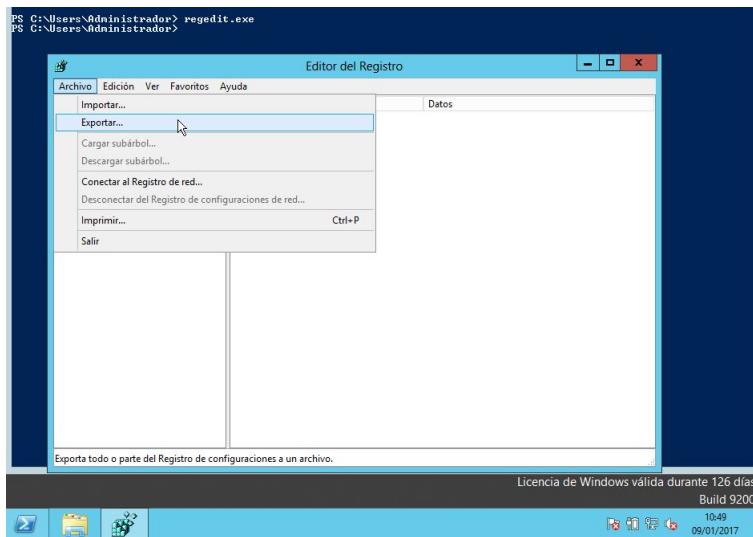


Figura 3.2: Realizando copia de Seguridad del archivo de registro de Windows Server.

Una vez ya aquí, ahora tan solo nos falta elegir el destino donde queremos que se almacena nuestra copia de seguridad del archivo de configuración y le definimos el nombre, podemos fijarnos en la 3.3, donde en mi caso, he elegido la carpeta 'Copia de seguridad' que se encuentra en el 'Escritorio' y he definido el nombre de Copia1.

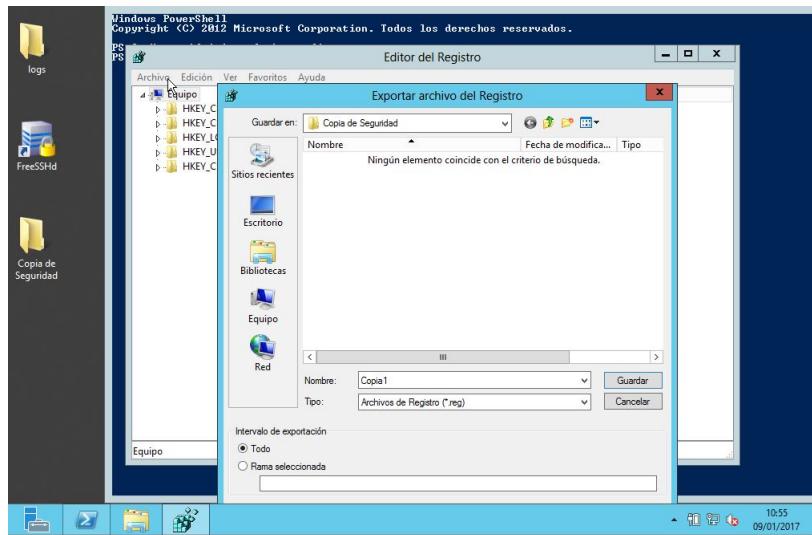


Figura 3.3: Realizando copia de Seguridad del archivo de registro de Windows Server.

También tenemos la opción de guardar solo una trama del archivo de configuración o todo el contenido de dicho archivo. Nosotros para mayor seguridad seleccionamos 'Todo' que viene marcado por defecto, y guardamos. Compramos que se ha realizado el proceso

Figura 3.4, y efectivamente ya tenemos realizada nuestra copia de seguridad de nuestro archivo de configuración.

```
Administrator:~> PS C:\Users\Administrador> cd ..\Desktop
Administrator:~> PS C:\Users\Administrador\Desktop> dir

  Directorio: C:\Users\Administrador\Desktop

Mode LastWriteTime Length Name
d--- 09/01/2017 10:58 Copia de Seguridad
d--- 09/12/2016 13:11 Logs
-a--- 20/12/2016 17:42 938 FreeSSHd.lnk

Administrator:~> PS C:\Users\Administrador\Desktop> cd '..\Copia de Seguridad'
Administrator:~> PS C:\Users\Administrador\Desktop\Copia de Seguridad> dir

  Directorio: C:\Users\Administrador\Desktop\Copia de Seguridad

Mode LastWriteTime Length Name
-a--- 09/01/2017 10:58 158864932 Copial.reg

Administrator:~>
```

Figura 3.4: Copia de seguridad realizada de nuestro archivo de configuración.

Ahora pasamos al proceso contrario, es decir, a la restauración de dicho archivo, para ello, una vez nos situamos en el Editor de Registro como conseguimos anteriormente en la Figura 3.2, y en esta caso hacemos click sobre Archivo -> Importar como vemos en la Figura 3.5.

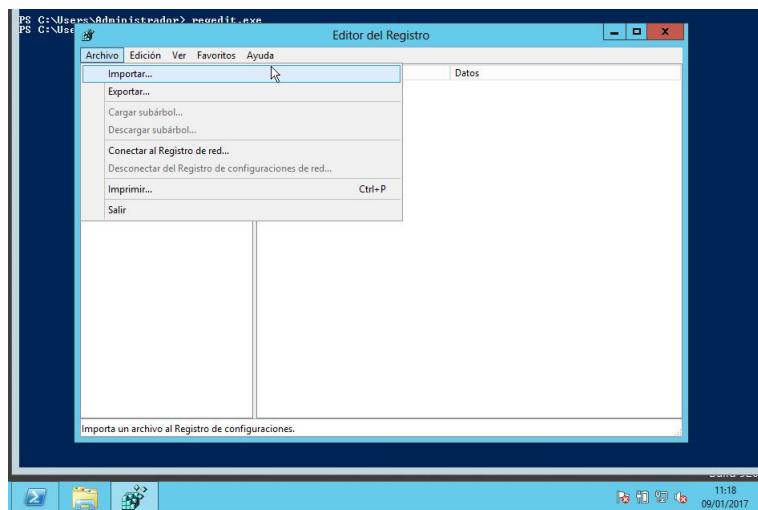


Figura 3.5: Restaurando archivo de configuración de nuestro Windows Server.

Ahora seleccionamos el origen donde guardemos nuestra copia de seguridad de nuestro archivo, la seleccionamos y pulsamos Abrir, Figura 3.6.

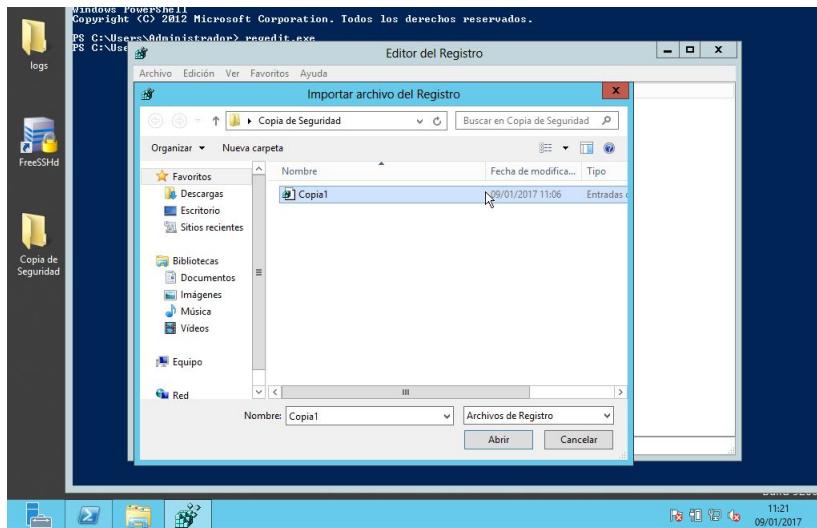


Figura 3.6: Restaurando archivo de configuración de nuestro Windows Server.

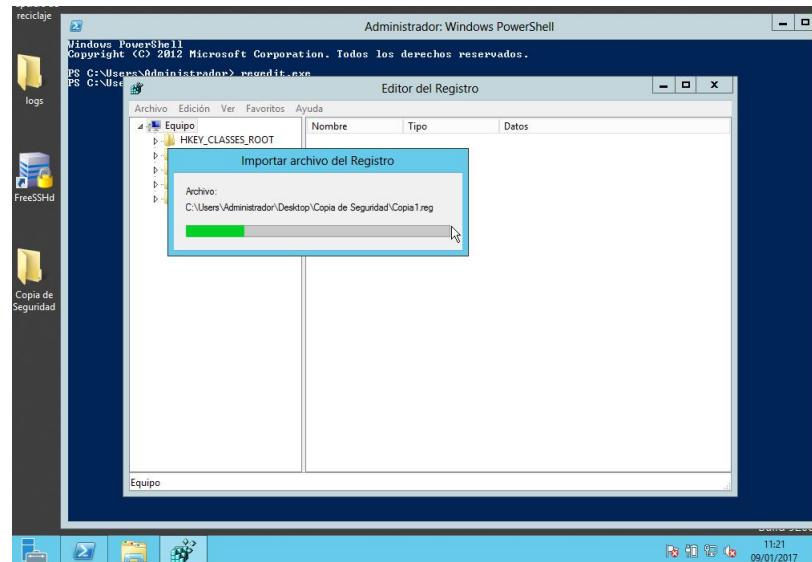


Figura 3.7: Restaurando archivo de configuración de nuestro Windows Server.

Después de este proceso ya obtendremos nuestra restauración del archivo de configuración. Figura 3.7.

Probablemente obtengamos una mensaje de error diciendo que no ha sido posible la escritura de todos los datos, debido a que algún registro está siendo usado o está abierto. Para ello debemos de encontrar o buscar el proceso y eliminar dicho proceso.

3.2. Abra una ventana mostrando el editor del registro.

Con hemos realizado en el apartado anterior, para obtener la ventana con el Editor del registro, nos vamos con ejecutar la orden 'regedit.exe' en la consola de comandos o PowerShell, podemos ver un ejemplo en la Figura 3.1.

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

Para el caso de Apache en el capítulo 5.3 Desempeño de Apache de [4] encontramos varias recomendaciones para obtener mejoras en el rendimiento del servicio Apache como por ejemplo:

- Podemos definir la directiva 'MaxClients' con el valor apropiado de la memoria de nuestro servidor Apache mediante la fórmula: Total de memoria disponible * 80 % / Uso máximo de memoria por los procesos de Apache. Definiendo así un mejor rendimiento para el servidor Moodle. Nota: si el valor de MaxClients es superior a ServerLimit debemos modificar también esta directiva, y debemos tener en cuenta que no deberemos sobrepasar el valor de nuestra memoria disponible, ya que sino nuestro servidor consumirá más RAM de la disponible y deberá hacer uso de la memoria compartida swap y no obtendremos mejoras en el rendimiento.
- Minimizar lo más posible el número de módulos de Apache en el archivo httpd.conf para mejorar el uso de memoria consumida.
- Utilizar la última versión de Apache, en la documentación menciona Apache2.
- Intentar disminuir el valor de 'MaxRequestsPerChild' en el archivo httpd.conf (entre 20-30).
- Para servidores sobrecargados podemos ajustar la directiva 'KeepAlive a Off' y disminuir el valor de 'KeepAliveTimeout' que es la encargada de definir el tiempo de espera para servir por ejemplo una página a un cliente, por defecto contiene el valor 15 (segundos), es decir, por cada usuario le permite la espera de 15 segundos para descarga la página solicitada, si somos capaces de disminuir esta espera manteniendo el tiempo necesario para que el cliente obtenga la página sin problemas, daremos un respiro a nuestro servidor, y podrá responder más peticiones.
- Podemos ajustar la directiva 'Reverse Proxy server' para que nuestro servidor almacene las páginas en caché, y el servicio de dichas páginas será más rápido.
- Si no usamos un archivo .htaccess, podemos ajustar la varialbe 'AllowOverride' a None para desactivar la búsqueda de dicho archivo anterior.
- Ajustar la directiva 'DirectoryIndex' para configurar los archivos índices de nuestro servicio web, evitando así la negociación de contenido.

- Configurar la variable 'ExtendedStatus' a Off.
- Modificar la directiva 'HostnameLookups' a Off para reducir la latencia del servicio de asignación de IPs.
- Reducir el valor de 'Timeout' entre el rango de 30 y 60 (segundos).
- Podemos usar la orden 'Options -Indexes FollowSymLinks' para reducir las peticiones de E/S.
- Cache. Debemos de evitar el almacenamiento en cache una semana antes de una actualización, ya que si almacenamos todo el contenido en caché y realizamos una actualización, sabemos que después de un reinicio del sistema la caché es volátil, y se perderá toda la información.
- La compresión de los archivos de respuesta que suministran las páginas de nuestro servidor, disminuirá los tiempos de respuesta y mejorará el rendimiento del servidor.

Para el caso de IIS la configuración que podemos realizar se encuentra en el registro: HKLM\SYSTEM\CurrentControlSet\Services\Inetinfo\Parameters\ y podemos realizar:

- Modificar el valor de la directiva 'ListenBackLog' equivalente al tiempo de respuesta para cada usuario.
- 'MemCacheSize' encargada de ajustar la cantidad de memoria de caché que se utilizará para los archivos que se almacenen en ella.
- Modificar el valor de 'MaxCachedFileSize' que en este caso se encarga de definir el tamaño máximo de un archivo que se puede almacenar en la caché.
- Se puede cambiar el tiempo que permanecerán los archivos almacenados en la memoria caché a través de 'ObjectCacheTTL', la métrica es milisegundos.

5. Cuestión 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.

Entre las mejoras de rendimiento de los servidores Web, se encuentra la compresión de los archivos html a la hora de servir una página web, mejorando así el tiempo de respuesta de nuestro servidor.

Como nos dice [6], en nuestro servidor IIS de windows server no permite el almacenamiento de respuestas dinámicas en caché, pero sí respuestas estáticas, por lo que podremos llevar a cabo esta mejora de rendimiento para páginas html estáticas.

Para ello necesitamos la configuración de IIS. Seguimos el procedimiento que nos indica [8]. Una vez nos encontramos en el panel de Administrador del servidor, hacemos click sobre Herramientas ->Administrador de Internet Information Services (IIS), que se encuentra en el margen superior derecho como vemos en la Figura 5.1.

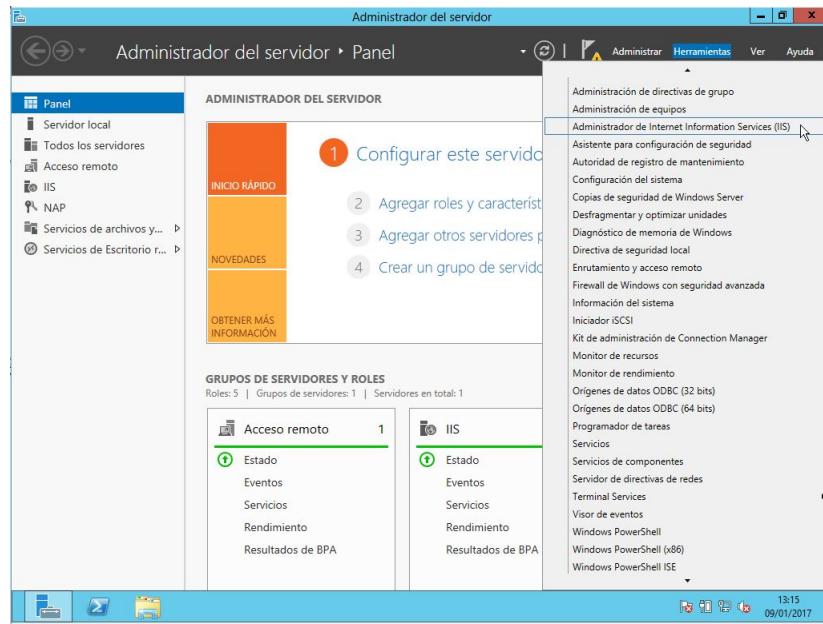


Figura 5.1: Panel de Administrador del servidor de Windows Server.

Una vez nos encontramos en el Administrador, hacemos click en el margen superior izquierdo donde se encuentras nuestro servidor IIS 'WIN-NTVUS0B26FI...' y nos aparecerá el menú de características como vemos en la Figura 5.2.

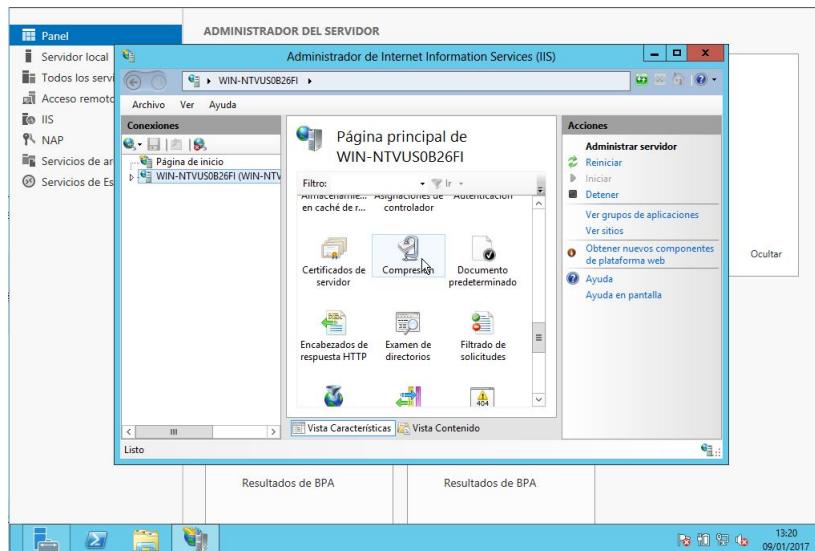


Figura 5.2: Panel de Administrador del servidor IIS.

Buscamos la característica Compresión y hacemos doble click sobre ella como podemos comprobar también en la Figura anterior. Y nos encontramos en la Figura 5.3.

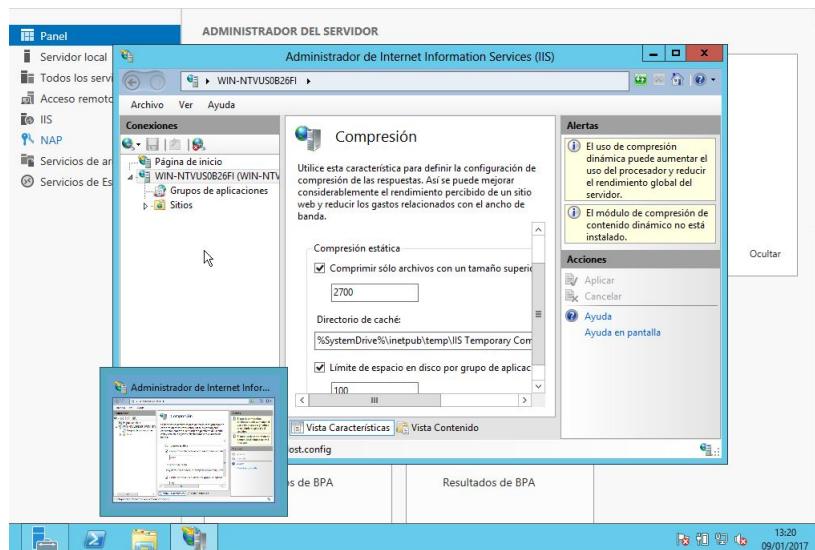


Figura 5.3: Característica de compresión de nuestro servidor IIS.

Como hemos mencionado antes esta configuración sólo tendrá efecto sobre la compresión estática, es decir, sobre nuestro servicio de página html estáticas. Como vemos en la Figura anterior nos encontramos que nuestro servidor IIS utiliza la compresión para archivos superiores a 2700 bytes, y debemos de minimizar este valor para que utilice la

compresión con archivos de menor tamaño.

Para mi caso voy a utilizar el valor 1, para que que utilice la compresión para todos los archivos que tengan un peso mayor a 1 byte. Podemos ver este paso en la Figura 5.4 y pinchamos en aplicar.

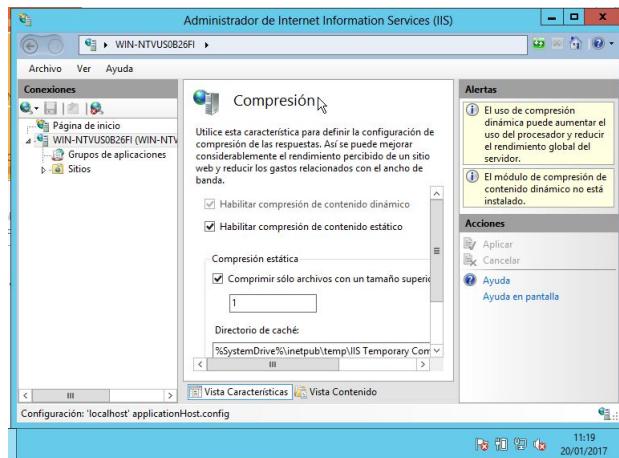


Figura 5.4: Modificando la características de compresión de nuestro servidor IIS.

Ya nuestro servidor IIS esta configurado para utilizar la compresión de nuestras páginas html estáticas a la hora de ser enviadas al cliente, vamos a verificar que es cierto y funciona.

Para llevar a cabo este procedimiento he incluido una página en mi servidor IIS con texto plano, que tiene un tamaño de 1596 KB, como podemos ver en la Figura 5.5 respecto al archivo 'prueba' .

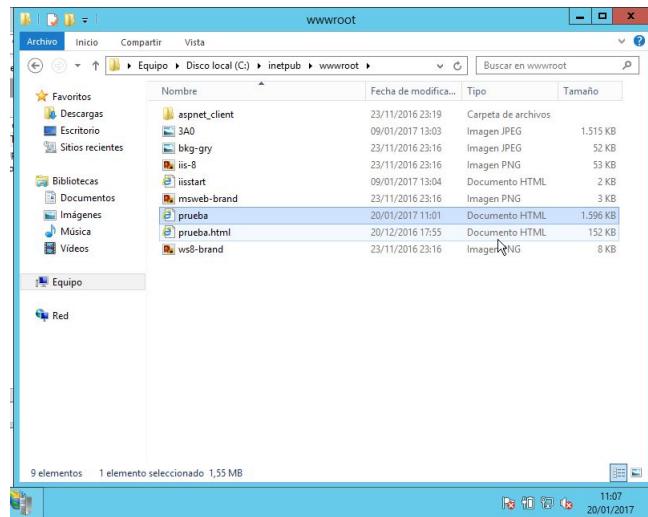


Figura 5.5: Directorio de nuestro servidor IIS para servir nuestras páginas webs.

Ahora realizamos una petición de la página de inicio desde nuestra máquina anfitriona que actuará como cliente. Voy a utilizar el navegador 'firefox mozilla', antes de realizar la petición debemos de realizar los siguientes pasos:

- Nos situamos en inspeccionar elemento que nos ofrece 'mozilla' haciendo click con el botón derecho del ratón.
- Una vez abierta esta herramienta buscamos en el panel superior la pestaña 'Red'.

Ya una vez nos encontramos en este punto, ya podemos realizar la petición a nuestro servidor Windows, facilitando la dirección IP y la página sobre la que deseamos realizar la petición, como podemos ver en la Figura 5.6.

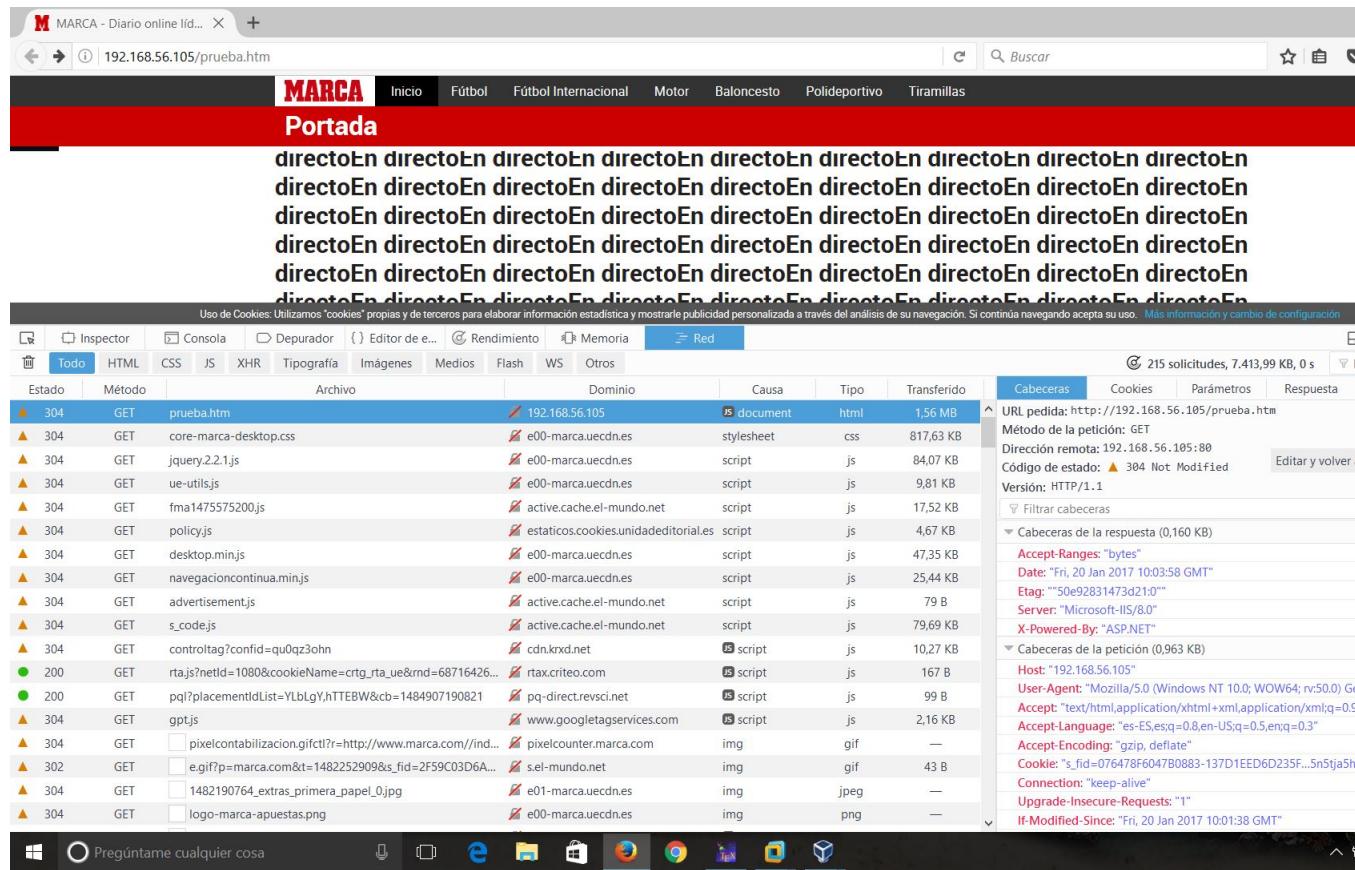


Figura 5.6: Petición de nuestra página web deseada a través del Windows Server.

Mientras se realiza el trabajo el navegador realizando la petición y cargando el contenido recibido mediante la petición, debemos de encontrar en el panel izquierdo la petición (GET) de la página que hemos solicitado. Nos podemos fijar en la Figura anterior para este proceso.

Una vez la hemos encontrado, en el panel derecho, encontramos una pestaña denominada "cabeceras", en la que obtendremos toda la información sobre como se ha realizado y obtenido la página solicitada.

Comentando esta información un poco, podemos ver la información correspondiente a la fecha, el servidor que ha sido responsable, dirección de red del servidor... y mucha más información que podemos encontrar.

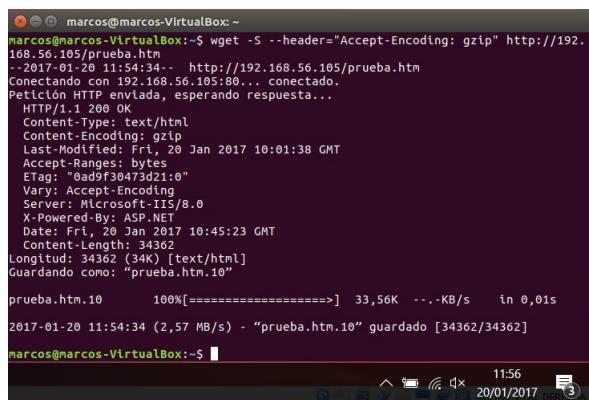
Entorno al objetivo de esta cuestión nos centramos en la directiva 'Accept-Encoding' que es la encargada de mostrarnos como ha sido enviada nuestra petición, es decir, por ejemplo en este caso, le hemos solicitado la solicitud al servidor ofreciéndole que nos puede devolver la página comprimida mediante 'gzip o deflate', para que el servidor decida según

su configuración enviarnos la página web solicitada comprimida.

Ahora particularmente para saber que la página la hemos recibido comprimida, debemos de visualizar la directiva 'Content-Encoding', que en mi caso no he conseguido visualizar. He intentado cambiar la configuración de IIS, cambiando la compresión en varios tamaños, pero por otro motivo externo, no llegué a conseguir que mi servidor IIS enviará la página html comprimida.

También pensé que podía ser debido al navegador, aunque por defecto no debería ser, y utilicé el comando 'wget' como podemos consultar en [10].

Utilicé la orden 'wget -S --header="Accept-Encoding: gzip" http://192.168.56.105/prueba.htm' como podemos ver en la Figura 5.7.



```
marcos@marcos-VirtualBox:~$ wget -S --header="Accept-Encoding: gzip" http://192.168.56.105/prueba.htm
192.168.56.105/prueba.htm
--2017-01-20 11:54:34-- http://192.168.56.105/prueba.htm
Conectando con 192.168.56.105:80... conectado.
Petición HTTP enviada, esperando respuesta...
HTTP/1.1 200 OK
Content-Type: text/html
Content-Encoding: gzip
Last-Modified: Fri, 20 Jan 2017 10:01:38 GMT
Accept-Ranges: bytes
ETag: "8ad9f30473d21:0"
Vary: Accept-Encoding
Server: Microsoft-IIS/8.0
X-Powered-By: ASP.NET
Date: Fri, 20 Jan 2017 10:45:23 GMT
Content-Length: 34362
Longitud: 34362 (34K) [text/html]
Guardando como: "prueba.htm.10"
prueba.htm.10          100%[=====] 33,56K  ---KB/s  in 0,01s
2017-01-20 11:54:34 (2,57 MB/s) - "prueba.htm.10" guardado [34362/34362]
marcos@marcos-VirtualBox:~$
```

Figura 5.7: Realizando una petición mediante el comando 'wget'.

Y efectivamente, podemos ver la directiva 'Content-Encoding: gzip', que nos indica que el contenido de nuestra página ha sido comprimido. Incluso podemos ver como el tamaño del archivo 'html' ha cambiado de 1,56 MB aprox. que tenía anteriormente, a un tamaño de 34 K.

6. Cuestión 6.

- 6.1. **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.**

Esta cuestión partimos desde cero, por lo tanto vamos a comenzar por la instalación de nuestro servidor, en mi caso he elegido Ubuntu Server 16.04 LTS donde podemos encontrar la descarga en [7].

Voy a demostrar paso a paso por lo que este proceso se puede hacer un poco extenso pero es una instalación muy sencilla.

Una vez arrancada la ".ISO." en nuestro equipo los pasos son:

Seleccionamos el idioma Figura 6.1.

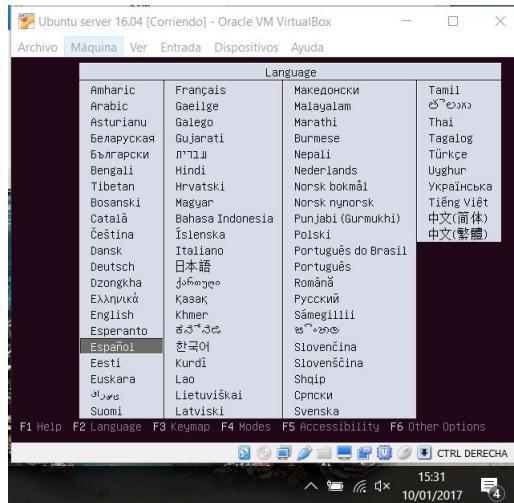


Figura 6.1: Seleccionando idioma para la instalación de Ubuntu Server.

Pulsamos enter sobre Instalar Ubuntu Server como vemos en la Figura 6.2.

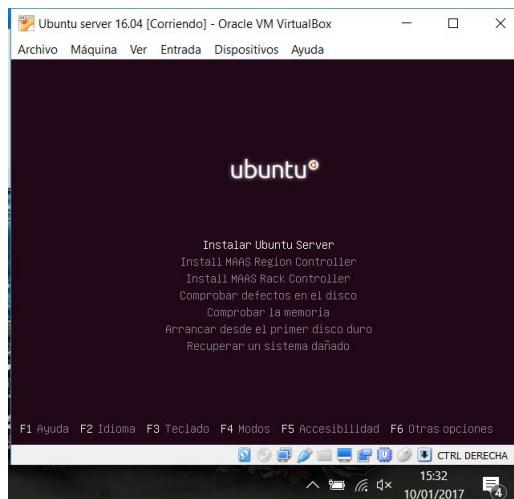


Figura 6.2: Seleccionando instalar Ubuntu Server.

Volvemos a seleccionar el idioma para nuestro sistema Ubuntu Server, Figura 6.3.

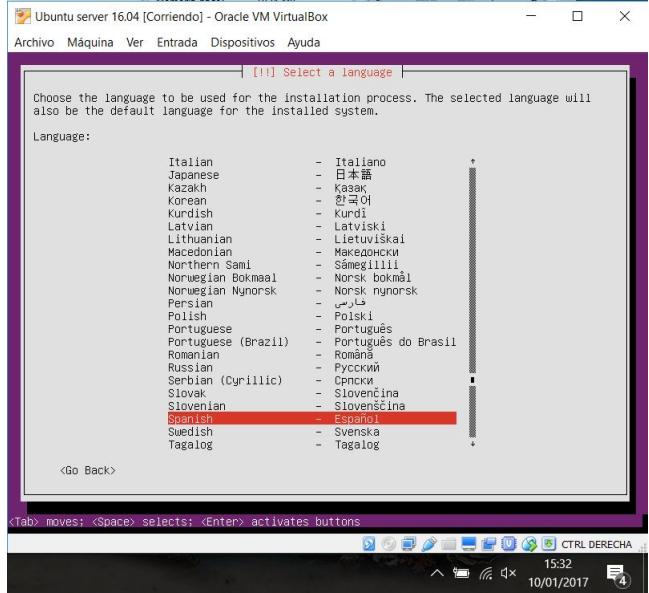


Figura 6.3: Seleccionando idioma para la instalación del sistema Ubuntu Server.

Ahora toca seleccionar la ubicación, Figura 6.4.

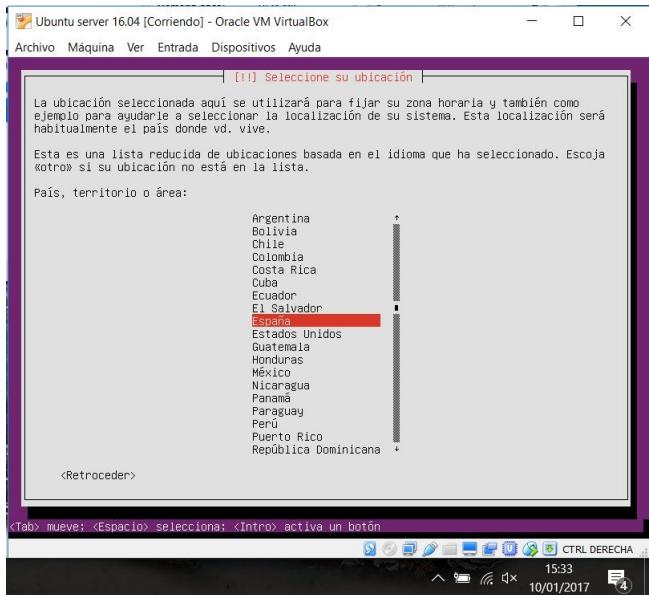


Figura 6.4: Seleccionando nuestra ubicación de nuestro Ubuntu Server.

Continuamos y nos pide detectar el teclado, pero seleccionamos la opción 'No' para seleccionarlo desde una lista Figura 6.5.

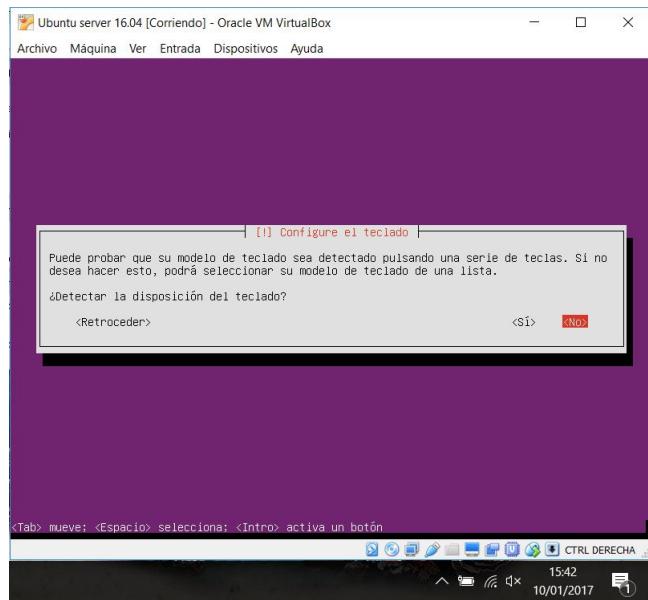


Figura 6.5: Seleccionando la opción 'No' para detectar el teclado para la instalación de Ubuntu Server.

Y ahora en este momento seleccionamos el origen de nuestro teclado, en nuestro caso 'Spanish' como vemos en la Figura 6.6.

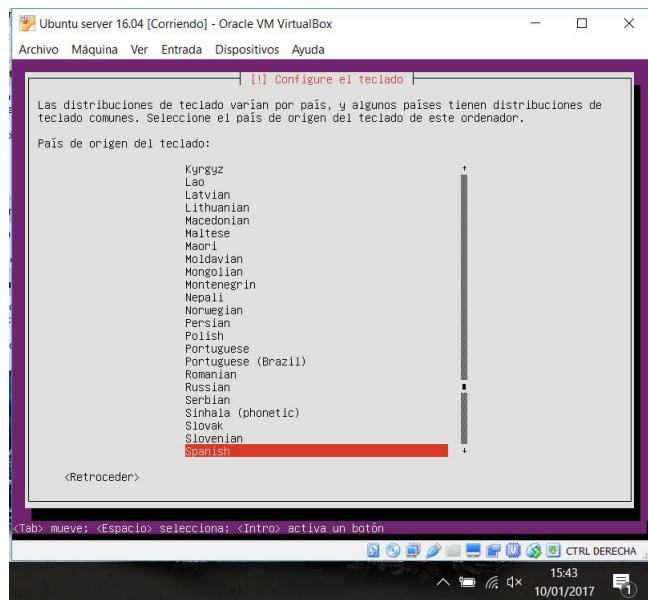


Figura 6.6: Seleccionando origen de nuestro teclado.

Y ahora la distribución del teclado, que seleccionamos de nuevo 'Spanish', Figura 6.7.

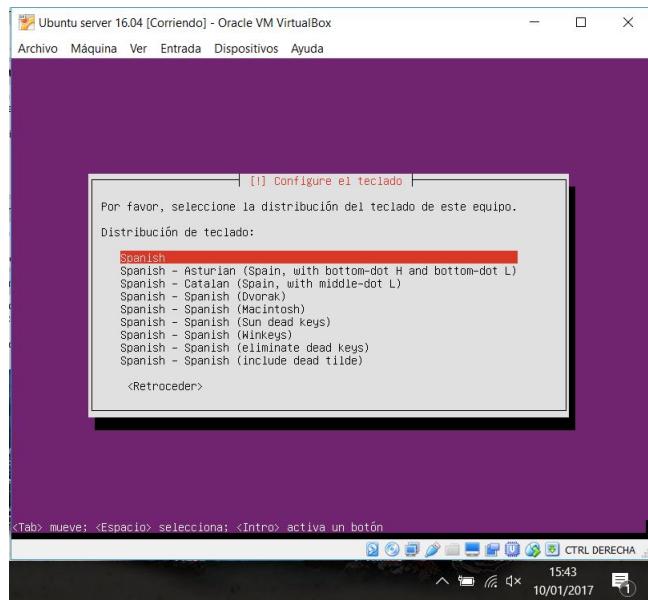


Figura 6.7: Seleccionando la distribución de nuestro teclado para la instalación de Ubuntu Server.

A continuación introducimos el nombre que deseamos definir a nuestra máquina, Figura 6.8, en mi caso 'marcos'.

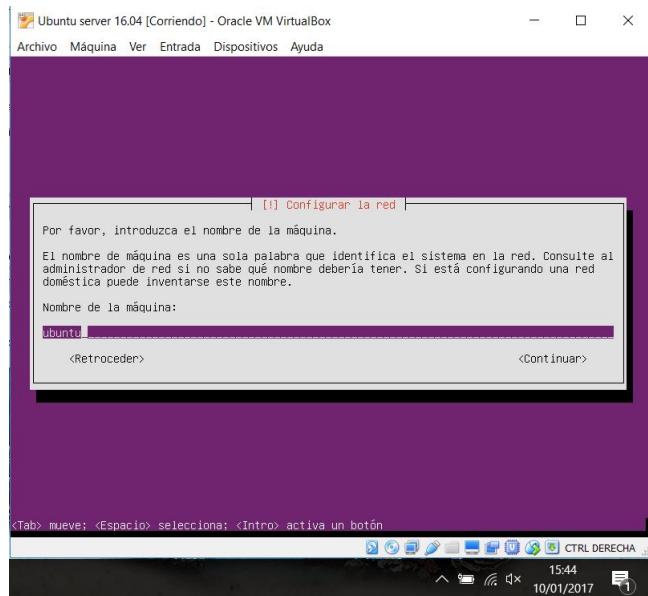


Figura 6.8: Definiendo el nombre de nuestra máquina para Ubuntu Server.

El siguiente paso es definir el nombre para el propietario de nuestro sistema Figura 6.9,

y posteriormente una cuenta, con nombre y contraseña para nuestro usuario. Primero el nombre del usuario, Figura 6.10 y posteriormente la contraseña, que hay que confirmarla, Figura 6.11 y 6.12.

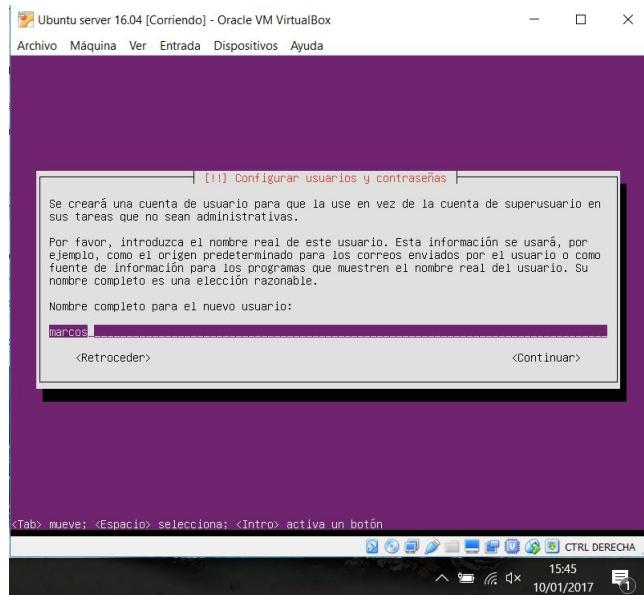


Figura 6.9: Definiendo el nombre del propietario para nuestro sistema Ubuntu Server.

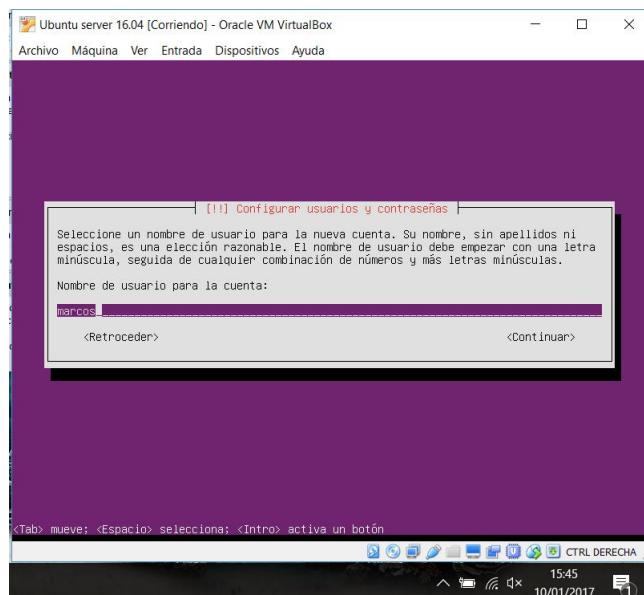


Figura 6.10: Definiendo el nombre del usuario de la nueva cuenta.

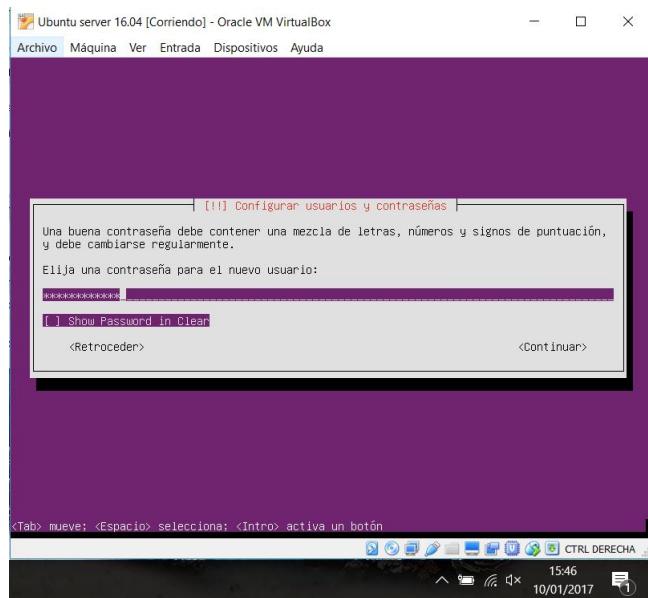


Figura 6.11: Definiendo la contraseña de nuestro usuario para la nueva cuenta.

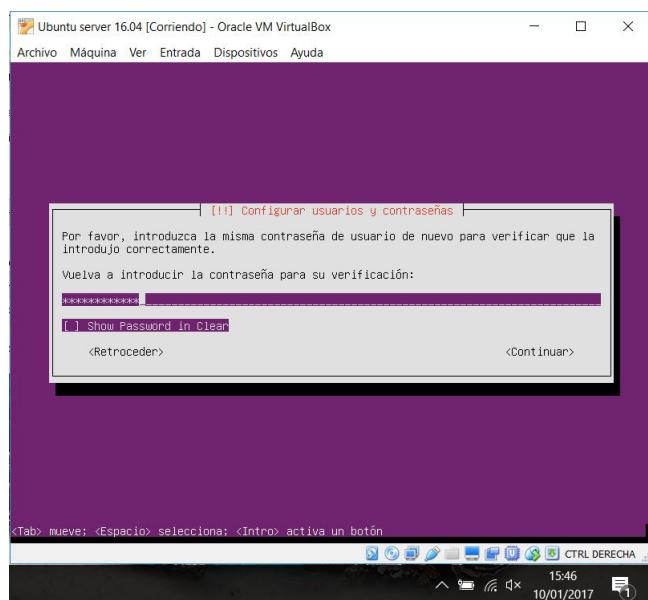


Figura 6.12: Confirmando la contraseña de nuestro usuario.

Continuando con el proceso, ahora nos toca elegir si deseamos realizar un cifrado de nuestra carpeta personal, podemos volver a la práctica 1 de esta asignatura y seguir el proceso de cifrar las distintas particiones de nuestro sistema. Para no extender este proceso de instalación elegimos la opción 'No', como vemos en la Figura 6.13.

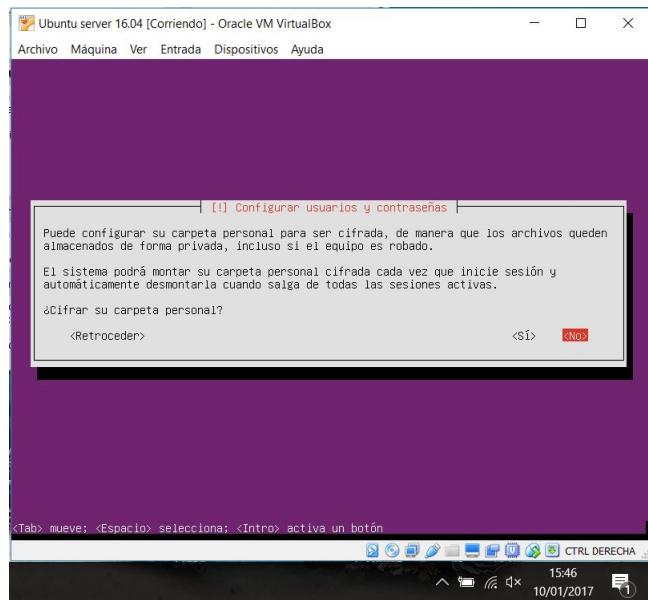


Figura 6.13: Omitiendo el proceso de cifrar nuestro Sistema Ubuntu Server.

En este paso, como realicemos también la práctica 1, podemos crear las distintas particiones para cada directorio básico de nuestro servidor, para un uso más eficiente de los directorios del disco, asignándole manualmente tanto el tamaño, como el tipo de archivos... Pero para este caso vamos a utilizar la opción 'guiado-utilizar todo el disco', que asignará automáticamente todo el espacio del disco y abreviaremos el proceso de la práctica 1, Figura 6.14.

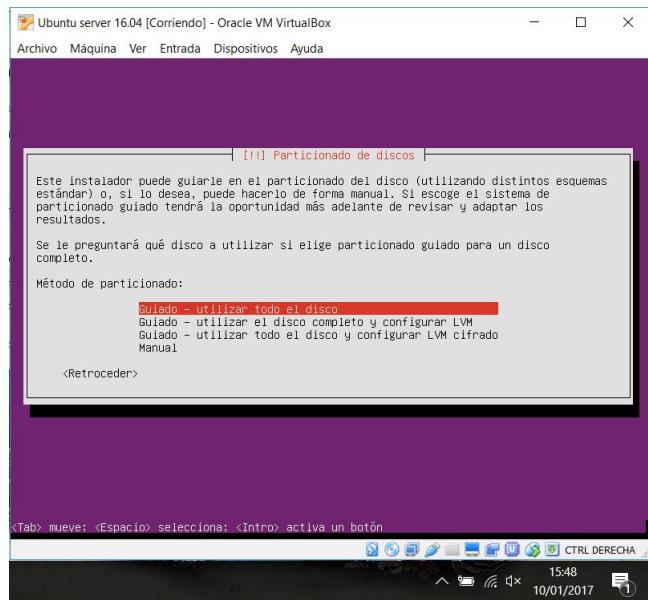


Figura 6.14: Asignando el particionado del disco automáticamente

Seleccionamos el disco donde deseamos instalar nuestro sistema operativo en el caso de disponer de varios discos duros, en mi caso solo dispongo de uno, como vemos en la Figura 6.15.

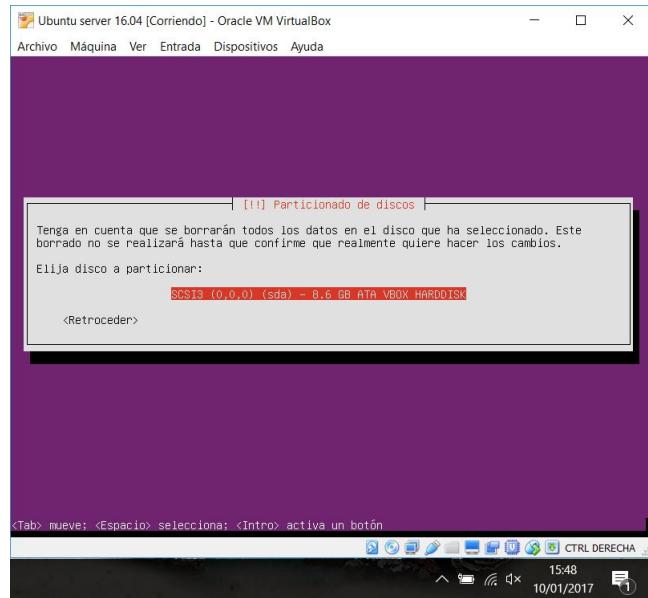


Figura 6.15: Seleccionando el disco duro donde deseamos instalar nuestro sistema.

Como elegimos el particionado automático y ya hemos elegido el disco duro donde deseamos instalar nuestro sistema.

mos realizar la instalación, el asistente selecciona el tamaño de las particiones y el tipo de archivo automáticamente, seleccionando la opción 'Sí' continuamos con el proceso, Figura 6.16.

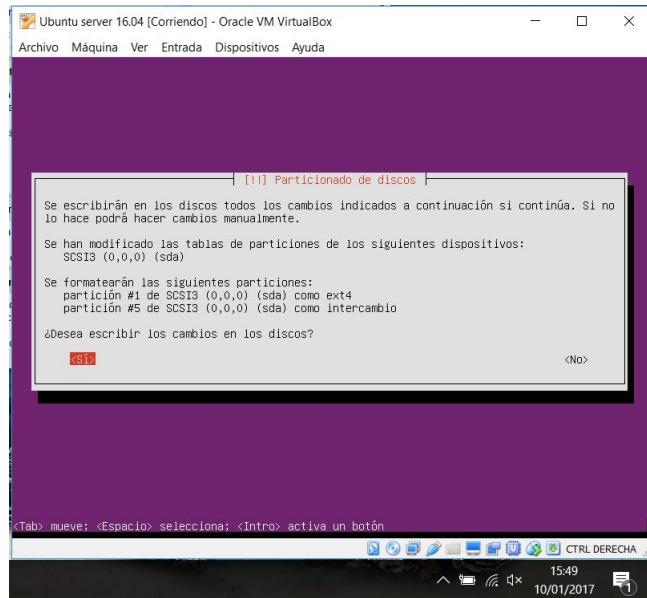


Figura 6.16: Aceptando las características del particionado automático.

Ahora nos toca indicar la información necesaria del proxy HTTP, en el caso de que necesitemos o tengamos un proxy configurado para tener acceso a la red. En el caso de que lo necesitemos hay que indicar el usuario, contraseña, dirección del servidor y el puerto configurado como nos indica en la Figura 6.17, en mi caso no necesito proxy por lo que dejo en blanco esta opción.

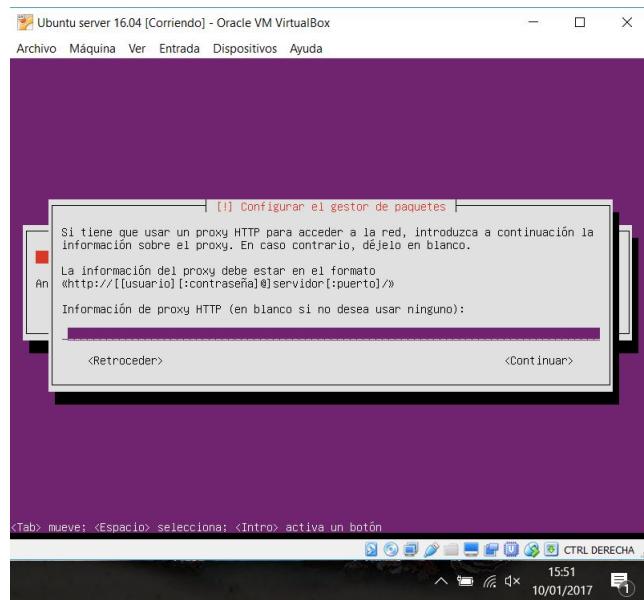


Figura 6.17: Añadir la información del proxy HTTP si es necesario para nuestro sistema.

Normalmente las actualizaciones las incluimos manualmente para mayor seguridad, pero en este proceso le vamos a indicar al asistente que incluya las actualizaciones automáticas para que haga este proceso por nosotros como vemos en la Figura 6.18.

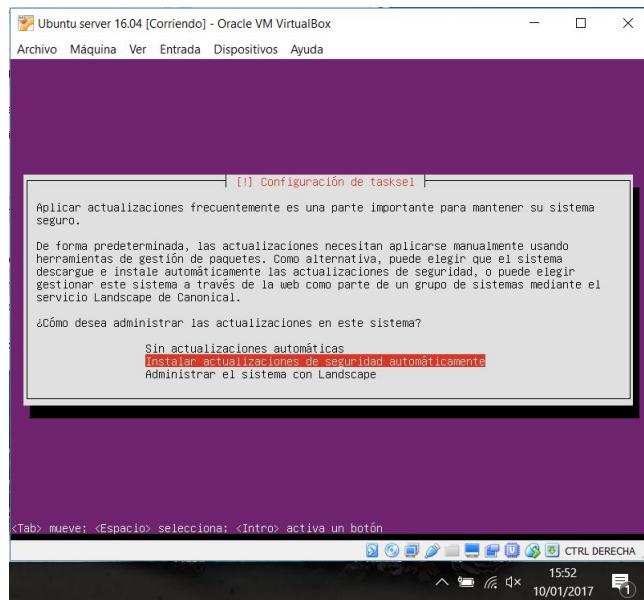


Figura 6.18: Instalación de actualizaciones de seguridad automáticamente.

El asistente nos facilita la instalación de servicios automáticos para nuestro servidor,

pero en este caso si nos vamos a preocupar personalmente de instalar y configurar los servicios de nuestro servidor manualmente. Por lo que seleccionamos continuar sin más, Figura 6.19.

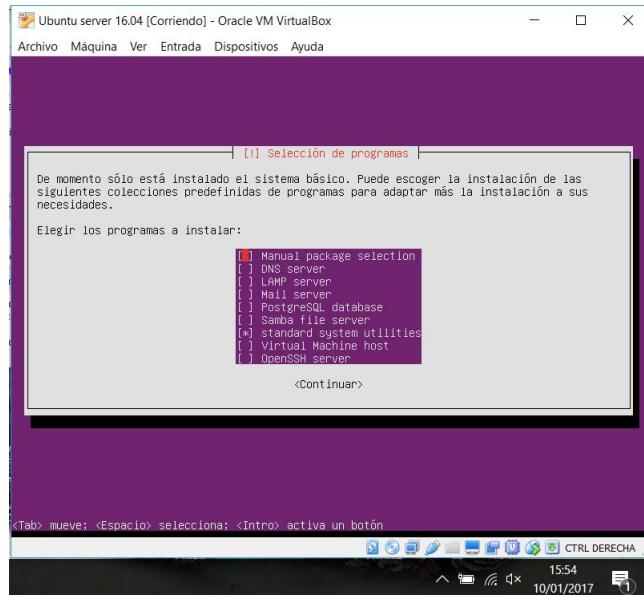


Figura 6.19: No seleccionando ningún servicio extra para nuestro servidor.

Ya ultimando la configuración básica de nuestro servidor, nos falta incluir el sistema de arranque (GRUB) en nuestra instalación, seleccionamos 'Si' y continuamos, Figura 6.20.

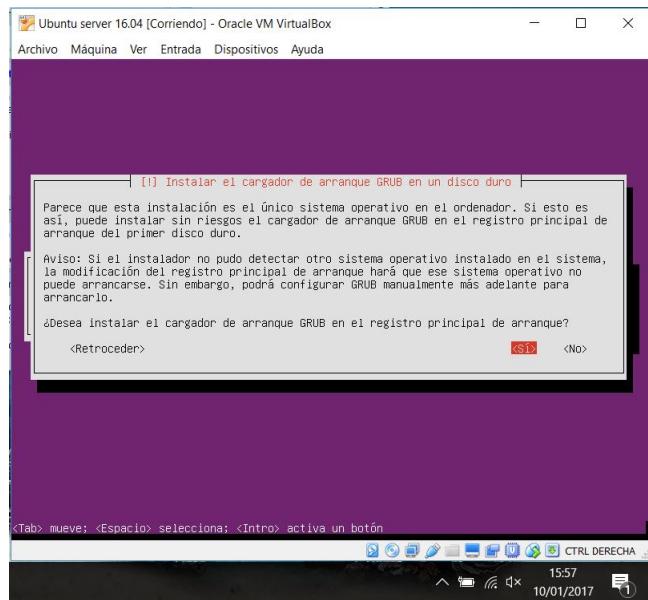


Figura 6.20: Incluyendo el sistema de arranque a nuestra instalación de Ubuntu Server.

Después de esperar para completar la instalación, ya tenemos nuestro sistema completamente instalado. Continuamos y automáticamente se reinicia nuestro sistema como vemos en la Figura 6.21.

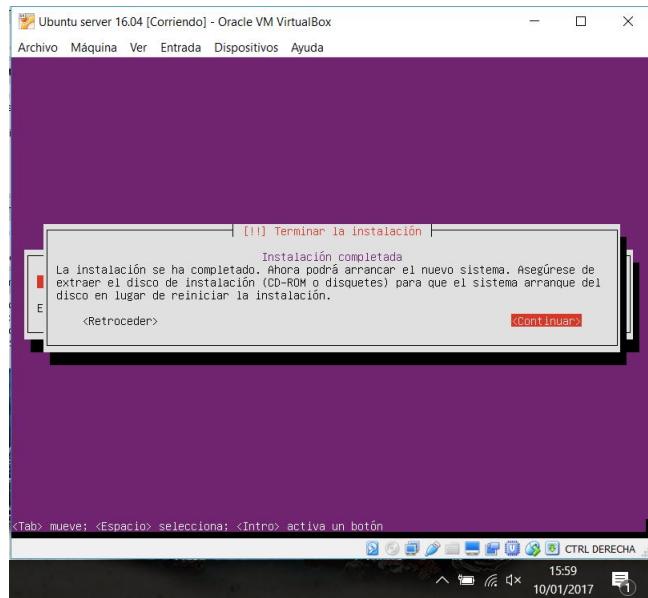


Figura 6.21: Finalización de la instalación de Ubuntu Server.

Una vez se reinicia nuestra máquina podemos iniciar sesión con nuestro usuario y con-

traseña que hemos definido en este proceso, Figura 6.22.

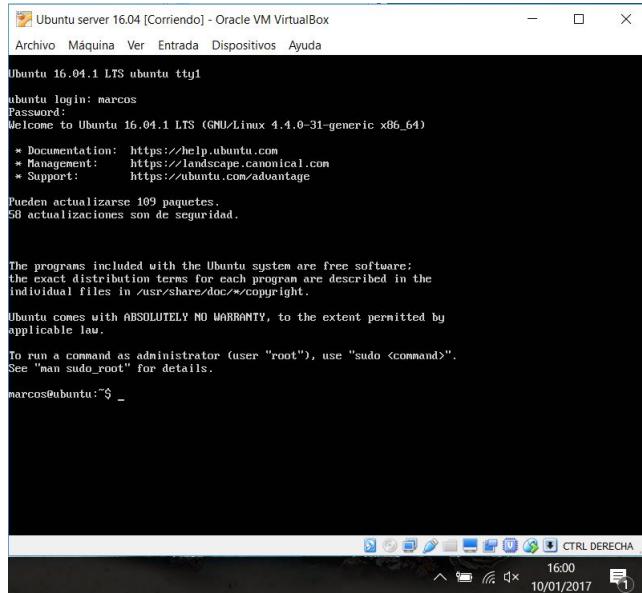


Figura 6.22: Primer inicio de nuestro sistema Ubuntu Server.

Ya tenemos instalado, listo e iniciado nuestro servidor Ubuntu, ahora ¿que sigue?, en primer lugar cumplimos con un objetivo mínimo cambiando el PROMPT de nuestro servidor para evitar así 'anticopias' de la realización de las prácticas. Para ello necesitamos modificar la variable 'PS1' del archivo 'bashrc' como vemos en la Figura 6.23.

```
Ubuntu server 16.04 [Corriendo] - Oracle VM VirtualBox
Archivo Máquina Ver Entrada Dispositivos Ayuda
color_prompt=
fi
if [ "$color_prompt" = yes ]; then
    PS1='${debian_chroot:+($debian_chroot)}\[\033[01;32m\]\u@\h \d \t \n\033[00m\]\n$ '
else
    PS1='${debian_chroot:+($debian_chroot)}\u@\h \d \t \n$ '
fi
unset color_prompt force_color_prompt
# If this is an xterm set the title to user@host:dir
case $TERM in
xterm*|rxvt*)
    PS1="\[\e[0;0;32m\]$debian_chroot:\e[0;0m\]\u@\h \d \t \n\$ "
(*)
;;
esac
# enable color support of ls and also add handy aliases
if [ -x /usr/bin/dircolors ]; then
    test -r .dircolors && eval "$(dircolors -b) > .dircolors" || eval "$(dircolors -b)"
    alias ls='ls --color=auto'
    alias dir='dir --color=auto'
    alias dircolors='dircolors --color=auto'
    alias grep='grep --color=auto'
    alias fgrep='fgrep --color=auto'
    alias egrep='egrep --color=auto'
fi
# colored GCC warnings and errors
export GCC_COLORS='error=01:31:warning=01:35:note=01:36:caret=01:quote=01'
.bashrc" 117L, 3796C escritos
marcos@ubuntu:~$ vi .bashrc
```

Figura 6.23: Modificando nuestro PROMPT de nuestro servidor.

Ya realizado lo anterior continuamos con el objetivo de esta cuestión, ¿que necesitamos ahora?, por ejemplo podemos añadir una nueva tarjeta de red, ya que en mi caso utilizo una máquina virtual de VirtualBox, si nuestro caso es una máquina física necesitamos configurar las tarjetas de red disponibles para que tengamos acceso a nuestro servidor externamente.

Añadimos la tarjeta de red en VirtualBox como en la Figura 6.24.

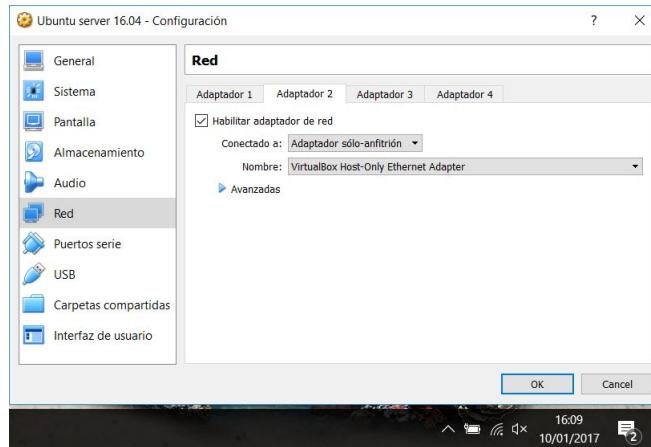


Figura 6.24: Añadiendo una nueva tarjeta de red en nuestra máquina.

¿Ya esta todo listo? Como he de esperar no todo es tan fácil, hacemos un pequeño 'ifconfig' para ver que configuraciones de red disponemos y que tarjetas de red se encuentran activadas, nos fijamos en la Figura 6.25.

```
marcos@ubuntu mar ene 10 2016:~$ ifconfig
enp0s3    Link encap:Ethernet direcciónHW 08:00:27:56:65:a1
          Direc. inet:10.0.2.15  Búf. :10.0.2.255 Másc:255.255.255.0
          Dirección inet6: fe80::a00:27ff:fe56:65a1/64 Alcance:Enlace
          ACTIVO BUCLE FUNCIONANDO MULTICAST MTU:1500 Métrica:1
          Paquetes RX:9 errores:0 perdidos:0 ove runns:0 frame:0
          Paquetes TX:9 errores:0 perdidos:0 ove runns:0 carrier:0
          colisiones:0 long.colasTX:1000
          Bytes RX:590 (590.0 B) TX bytes:990 (990.0 B)

lo        Link encap:Bucle local
          Direc. inet:127.0.0.1  Másc:255.0.0.0
          Dirección inet6: ::1/128 Alcance:Anfitrión
          ACTIVO BUCLE FUNCIONANDO MTU:65536 Métrica:1
          Paquetes RX:166 errores:0 perdidos:0 ove runns:0 frame:0
          Paquetes TX:166 errores:0 perdidos:0 ove runns:0 carrier:0
          colisiones:0 long.colasTX:1
          Bytes RX:11040 (11.8 KB) TX bytes:11040 (11.8 KB)

marcos@ubuntu mar ene 10 2016:~$
```

Figura 6.25: Configuraciones de red y tarjetas disponibles en nuestro sistema.

Ya vemos que falta algo, nuestra nueva tarjeta de red no se encuentra, con la ayuda de la orden 'ifconfig -a' vemos en la Figura 6.26 que si se encuentra disponible nuestra tarjeta añadida, en mi caso la tarjeta identificada por 'enp0s8', pero no está funcionando correctamente.

```

marcos@ubuntu:~$ ifconfig -a
enp0s3 Link encap:Ethernet direcciónHW 00:00:27:56:65:a1
      Direcc. inet:10.0.2.15 Difus.:10.0.2.255 Masc:255.255.255.0
          Dirección inet6: fe80::a00:27ff:fe56:65a1/64 Alcance:Bucle
          ACTIVO DIFUSIÓN FUNCIONANDO MULTICAST MTU:1500 Métrica:1
          Paquetes RX:8 errores:0 perdidos:0 oerruns:0 frame:0
          Paquetes TX:16 errores:0 perdidos:0 oerruns:0 carrier:0
          colisiones:0 long.colatTX:1000
          Bytes RX:1540 (1.5 KB) TX bytes:1520 (1.5 KB)

enp0s8 Link encap:Ethernet direcciónHW 00:00:27:24:5c:ca
      DIFUSIÓN MULTICAST MTU:1500 Métrica:1
          Paquetes RX:0 errores:0 perdidos:0 oerruns:0 frame:0
          Paquetes TX:0 errores:0 perdidos:0 oerruns:0 carrier:0
          colisiones:0 long.colatTX:1000
          Bytes RX:0 (0.0 B) TX bytes:0 (0.0 B)

lo Link encap:Bucle local
      Direcc. inet:127.0.0.1 Masc:255.0.0.0
          Dirección inet6: ::1/128 Alcance:localhost
          PCTIM BUCLE FUNCIONANDO MTU:65536 Métrica:1
          Paquetes RX:160 errores:0 perdidos:0 oerruns:0 frame:0
          Paquetes TX:160 errores:0 perdidos:0 oerruns:0 carrier:0
          colisiones:0 long.colatTX:1
          Bytes RX:11840 (11.8 KB) TX bytes:11840 (11.8 KB)

```

Figura 6.26: Configuraciones de red y tarjetas disponibles en nuestro sistema.

Procedemos a configurar la tarjeta de red para que pueda activarse, para ello, necesitamos modificar el archivo de configuración de interfaces que lo encontramos en el directorio '/etc/network/interfaces' y necesitamos añadir la siguiente información como vemos en la Figura 6.27.

```

# and how to activate them. For more information, see interfaces(5).
source /etc/network/interfaces.d/*
# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
auto enp0s3
iface enp0s3 inet dhcp

# The second network interface
auto enp0s8
iface enp0s8 inet dhcp

```

Figura 6.27: Modificando el archivo de configuración de interfaces de nuestro sistema.

Después de modificar y guardar los cambios del archivo anterior, necesitamos reiniciar el servicio que activa las interfaces, hacemos uso de la orden '/etc/init.d/networking restart', y nuestra tarjeta de red cobrará vida si todo va correctamente, como vemos en la Figura 6.28. Para comprobarlo realizamos de nuevo la ejecución de la orden 'ifconfig' y efectivamente vemos que nuestra tarjeta añadida 'enp0s8', ya esta configurada y obtiene la dirección IP '192.168.56.101' que pertenece a la dirección IP de nuestro servidor. Podemos ver este proceso en la Figura 6.28.

```

root@ubuntu:~home/narcos$ /etc/init.d/networking restart
[ ok ] Restarting networking (via systemctl): networking.service.
root@ubuntu:~home/narcos$ ifconfig
enp0s3    Link encap:Ethernet direcciónHW 00:00:27:56:65:a1
          Direc. inet: 10.0.2.15  Difus.: 10.0.2.255 Másc: 255.255.255.0
          Dirección inet6: fe80::a00:27ff:fe56:65a1/64 Alcance:Enlace
          ACTIVO DIFUSIÓN FUNCIONANDO MULTICAST MTU:1500 Métrica:1
          Paquetes RX:12 errores:0 perdidos:0 oerruns:0 frame:0
          Paquetes TX:28 errores:0 perdidos:0 oerruns:0 carrier:0
          colisiones:0 long.colat:1000
          Bytes RX:2078 (2.0 KB) TX bytes:3282 (3.2 KB)

enp0s8    Link encap:Ethernet direcciónHW 00:00:27:24:5c:ca
          Direc. inet: 192.168.56.101  Difus.: 192.168.56.255 Másc: 255.255.255.0
          Dirección inet6: fe80::a00:27ff:fe24:5c:ca/64 Alcance:Enlace
          ACTIVO DIFUSIÓN FUNCIONANDO MULTICAST MTU:1500 Métrica:1
          Paquetes RX:43 errores:0 perdidos:0 oerruns:0 frame:0
          Paquetes TX:8 errores:0 perdidos:0 oerruns:0 carrier:0
          colisiones:0 long.colat:1000
          Bytes RX:4952 (4.9 KB) TX bytes:1192 (1.1 KB)

lo        Link encap:Bucle local
          Direc. inet: 127.0.0.1 Másc: 255.0.0.0
          Dirección inet6: ::1/128 Alcance:Anfitrión
          ACTIVO BUCLE FUNCIONANDO MTU:65536 Métrica:1
          Paquetes RX:160 errores:0 perdidos:0 oerruns:0 frame:0
          Paquetes TX:160 errores:0 perdidos:0 oerruns:0 carrier:0
          colisiones:0 long.colat:1
          Bytes RX:11040 (11.0 KB) TX bytes:11040 (11.0 KB)

root@ubuntu:~home/narcos$ exit
exit
narcos@ubuntu mar ene 10 2016:"$ "

```

Figura 6.28: Comprobando la configuración de nuestra nueva tarjeta de red.

¿Ya es todo? Vamos a comprobar que realmente funciona, para ello podemos hacer un simple 'ping' desde por ejemplo nuestra máquina anfitriona a la dirección IP que ha obtenido nuestra tarjeta de red anterior. Vemos tal efecto en la Figura 6.29. Efectivamente hasta ahora todo funciona correctamente.

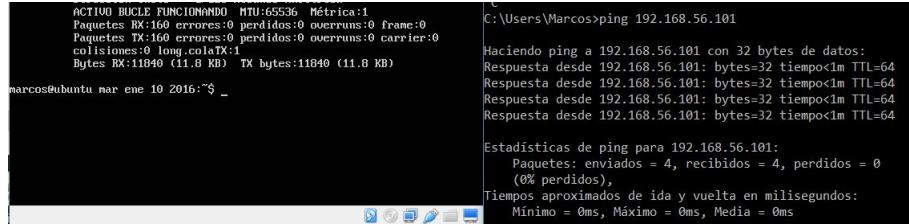


Figura 6.29: Comprobando el correcto funcionamiento de nuestra tarjeta de red del servidor.

Ya tenemos nuestro servidor Ubuntu corriendo, nuestra tarjeta de red configurada y ¿ahora?. Seguimos con el siguiente paso que pertenece a la instalación de un servicio concreto para nuestro servidor Ubuntu. En mi caso voy a elegir como llevamos realizando en las prácticas anteriores un servicio HTTP, para esto es necesario la instalación del servicio Apache.

Para llevar acabo este proceso, realizamos la ejecución de la orden 'apt install apache2', recordando que debemos tener privilegios de administrador ('sudo su'). Figura 6.30.

```
Configurando liblulu5.1-0 and64 (5.1.5-Subuntu1) ...
Configurando apache2-bin (2.4.18-2ubuntu3.1) ...
Configurando apache2-utils (2.4.18-2ubuntu3.1) ...
Configurando apache2-data (2.4.18-2ubuntu3.1) ...
Configurando apache2 (2.4.18-2ubuntu3.1) ...
Enabling module ppm_event.
Enabling module authz_core.
Enabling module authz_host.
Enabling module authn_core.
Enabling module auth_basic.
Enabling module access_compat.
Enabling module authn_file.
Enabling module authz_user.
Enabling module alias.
Enabling module dir.
Enabling module autoindex.
Enabling module env.
Enabling module mime.
Enabling module negotiation.
Enabling module setenvif.
Enabling module filter.
Enabling module deflate.
Enabling module status.
Enabling conf charset.
Enabling conf localized-error-pages.
Enabling conf other-ghosts-access-log.
Enabling conf security.
Enabling conf security-dash-bin.
Enabling site 000-default.
Configurando ssl-cert (1.0.37) ...
Procesando disparadores para libc-bin (2.23-0ubuntu3) ...
Procesando disparadores para systemd (229-Subuntu7) ...
Procesando disparadores para ureadahead (0.100.0-19) ...
Procesando disparadores para ufw (0.35-0ubuntu2) ...
root@ubuntu:/home/marcos$ exit
exit
marcos@ubuntu:~$ apt-get install apache2
```

Figura 6.30: Instalando el servicio Apache en nuestro servidor Ubuntu.

¿Listo? Vamos a comprobar si el servicio ha sido instalado correctamente y esta activo, hacemos uso de la orden '/etc/init.d/apache2 status' como vemos en la Figura 6.31, y parece que todo esta funcionando correctamente.

```
marcostubuntu mar ece 10 2016: 5 /etc/init.d/apache2 status
 * apache2.service - LSB: Apache2 web server
   Loaded: loaded (/etc/init.d/apache2; bad; vendor preset: enabled)
   Drop-In: /lib/systemd/system/apache2.service.d
     └─apache2-systemd.conf
 Active: active (running) since mar 2017-01-10 18:06:16 CET; 7min ago
   Docs: man:systemd-sysv-generator(8)
 CGroup: /system.slice/apache2.service
         ├─3725 /usr/sbin/apache2 -k start
         ├─3726 /usr/sbin/apache2 -k start
         ├─3727 /usr/sbin/apache2 -k start
         └─3728 /usr/sbin/apache2 -k start

ene 10 18:06:15 ubuntu systemd[1]: Starting LSB: Apache2 web server...
ene 10 18:06:15 ubuntu apache2[37200]: * Starting Apache httpd web server apache2
ene 10 18:06:15 ubuntu apache2[37200]: AH00558: apache2: Could not reliably determine the serv...sage
ene 10 18:06:16 ubuntu apache2[37200]: *
ene 10 18:06:16 ubuntu systemd[1]: Started LSB: Apache2 web server.
hint: Some lines were ellipsized, use -l to show in full.

marcostubuntu mar ece 10 2016: $
```

Figura 6.31: Comprobando el estado del servicio Apache en nuestro servidor Ubuntu.

Sabemos que podemos realizar algunas modificaciones básicas como realicemos en la práctica 2, como cambiar el puerto por defecto, y para ello consecuencia de habilitar el puerto que elijamos, pero para no extender más esta cuestión, dejamos los parámetros por defecto. Vamos a realizar una pequeña prueba, para verificar que nuestro servidor nos da servicio sobre una página solicitada. Vamos hacer uso de la página por defecto que contiene apache2, y realizamos una petición desde nuestra máquina anfitriona a dicha página por defecto, utilizando la dirección IP de nuestro servidor (192.168.56.101). Figura 6.32.

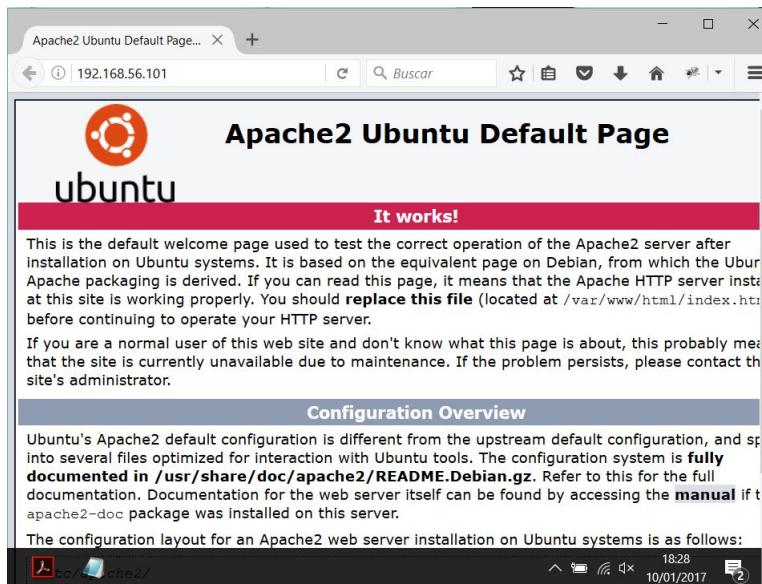


Figura 6.32: Petición de la página de prueba a nuestro servidor Ubuntu Server.

Tengo que recordar que ahora mismo en este punto, nuestro servidor Apache nos da el servicio de páginas web estáticas, si desearamos dar el servicio de páginas dinámicas necesitamos instalar un intérprete por ejemplo, PHP.

Yo me voy a quedar en este punto de ofrecer el servicio de páginas webs estáticas. Pero como el siguiente apartado vamos a realizar la monitorización y mejora de nuestro servidor voy a realizar un pequeño proceso, que consiste en incluir una página web con más contenido que la de prueba de Apache.

En primer lugar necesitamos incluir el servicio 'ssh' en nuestro servidor para poder realizar la transferencia de una nueva página web estática a nuestro servidor, y que pueda dar el servicio de esta nueva página.

Hacemos uso de la orden 'apt-get install openssh-server' como vemos en la Figura 6.33,

```

Selezionando el paquete python3-requests previamente no seleccionado.
Preparando para desempaquetar .../python3-requests_2.9.1-3_all.deb ...
Desempaquetando python3-requests (2.9.1-3) ...
Selezionando el paquete tcpd previamente no seleccionado.
Preparando para desempaquetar .../tcpd_7.6.q-25_amd64.deb ...
Desempaquetando tcpd (7.6.q-25) ...
Selezionando el paquete ssh-import-id previamente no seleccionado.
Preparando para desempaquetar .../ssh-import-id_5.5-0ubuntu1_all.deb ...
Desempaquetando ssh-import-id (5.5-0ubuntu1) ...
Procesando disparadores para systemd (2.23-ubuntu1) ...
Procesando disparadores para udev (0.35-0ubuntu2) ...
Procesando disparadores para ufw (0.35-0ubuntu2) ...
Configurando libupnp:amd64 (7.6.q-25) ...
Configurando openssh-client (1:7.2p2-4ubuntu2.1) ...
Configurando ncurses-term (6.0+20160213-ubuntu1) ...
Configurando openssh-sftp-server (1:7.2p2-4ubuntu2.1) ...
Configurando openssh-server (1:7.2p2-4ubuntu2.1) ...
Creating SSH RSA key; this may take some time ...
Z04HgZ256:efGauq3MfU1FzD6Gr1rE1LxJUNy1bHL2gEBBT2sU root@ubuntu (RSA)
Creating SSH DSA key; this may take some time ...
1024 SH0256:6hT0Zf1Zg5al1GvNMeR4V4+F3Dnhhuvr:z100s4 root@ubuntu (DSA)
Creating SSH ECDSA key; this may take some time ...
256 SH0256:ckN8puXusmHf8Q000nDQKG6G6VV1cdwYrEXX55LMSc root@ubuntu (ECDSA)
Creating SSH ED25519 key; this may take some time ...
256 SH0256:UcQmqtOoPuyt74P6cxnf7ReLNSh0dpmu17ThbUjA root@ubuntu (ED25519)
Configurando python3-urllib3 (1.13.1-2ubuntu0.16.04.1) ...
Configurando python3-requests (1.13.1-3) ...
Configurando tcpd (7.6.q-25) ...
Configurando ssh-import-id (5.5-0ubuntu1) ...
Procesando disparadores para libc-bin (2.23-ubuntu3) ...
Procesando disparadores para systemd (2.23-ubuntu1) ...
Procesando disparadores para udev (0.35-0ubuntu2) ...
Procesando disparadores para ufw (0.35-0ubuntu2) ...
root@ubuntu:/var/www/html# exit
exit
marcos@ubuntu mar ene 10 2016:/var/www/html$ apt-get install openssh-server_

```

Figura 6.33: Instalación del servicio SSH en nuestro servidor.

Recordar que necesitaremos tener privilegios de administrador para la instalación de este servicio, y volviendo a repetir, dejaremos los parámetros por defecto, ya que para el servicio SSH, podíamos cambiar de nuevo el puerto por defecto, inhabilitar la conexión para usuario root... como realicemos en la práctica 2.

Una vez instalado, deberíamos volver a comprobar si ha sido correcta la instalación y el estado correcto del servicio, pero por resumir un poco esta cuestión ya en este punto, después de la instalación todo está correcto, y procedemos a realizar la transferencia de una nueva página web con más contenido.

En la Figura 6.34, recordamos que para realizar la conexión ssh entre nuestra máquina anfitriona y nuestro servidor, necesitamos incluir la clave de nuestro sistema anfitrión en nuestro servidor, para ello hacemos uso de la orden 'ssh-keygen -f /home/marcos/.ssh/known_hosts -R 192.168.56.101' como vemos en la Figura 6.34.

```

marcos@marcos-VirtualBox:~$ ssh-keygen -f "/home/marcos/.ssh/known_hosts" -R 192.168.56.101
# Host 192.168.56.101 found: line 1
/home/marcos/.ssh/known_hosts updated.
Original contents retained in /home/marcos/.ssh/known_hosts.old
Procesando disparadores para ufw (0.35-0ubuntu2) ...
root@ubuntu:/var/www/html# exit
exit
marcos@ubuntu mar ene 10 2016:/var/www/html$ apt-get install openssh-server

```

Figura 6.34: Incluyendo la clave ssh de nuestro sistema anfitrión en nuestro servidor.

Ahora con la ayuda de la orden 'scp ruta_del_archivo usuario@IP_servidor:' transferimos el nuevo fichero html al servidor, y lo debemos de ubicar en el directorio del servidor: '/var/www/html/' como vemos en la Figura 6.35.

```

marcos@marcos-VirtualBox:~/Escritorio$ scp ./prueba.html marcos@192.168.56.101:
marcos@192.168.56.101's password:
prueba.html                                         100% 162KB   8.2MB/s  00:00
marcos@marcos-VirtualBox:~/Escritorio$ 

```

Figura 6.35: Transfiriendo el nuevo archivo html a nuestro servidor Ubuntu.

Por último comprobamos que todo ha tenido éxito, y solicitamos la nueva página al

servidor para verificar que el servidor nos ofrece dicha página, nos fijamos en la Figura 6.36.

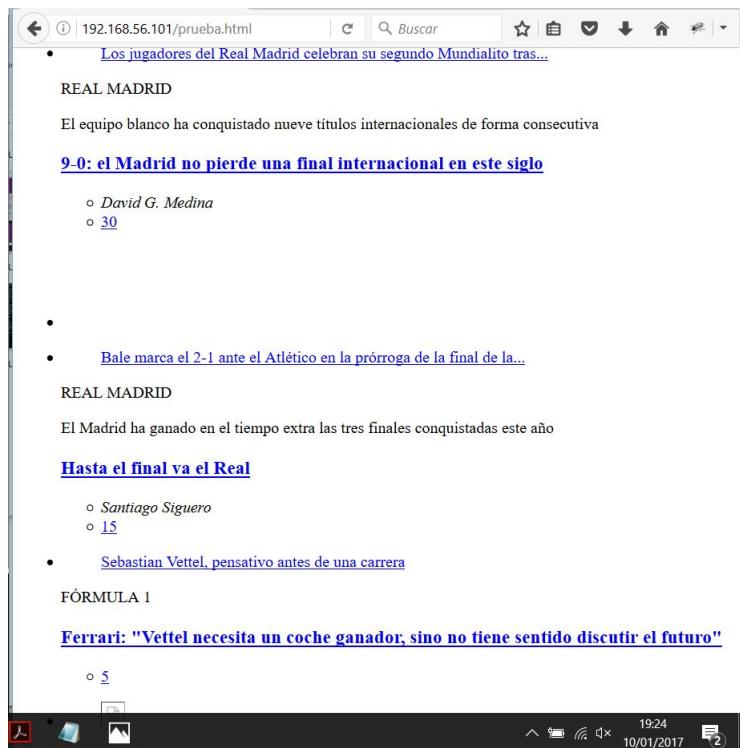


Figura 6.36: Solicitud de la nueva página html incluida en nuestro servidor.

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

Pasamos ahora a la mejora del servicio, para ello en primer lugar vamos a monitorizar el servicio con una carga de trabajo, con la ayuda del comando 'ab', para posteriormente después de mejorar el rendimiento del servidor para nuestro servidor, podamos volver a monitorizar y poder evaluar los resultados obtenidos.

Voy a fijar los valores de 10000 solicitudes con una concurrencia de 100 peticiones cada solicitud para el comando 'ab'.

Al mismo tiempo en el servidor, voy a lanzar el comando 'top' para realizar un pequeño seguimiento de la monitorización del trabajo de nuestro servidor, para poder ver que comportamiento obtiene cuando se le solicita el servicio HTTP. Figura 6.37.

The screenshot shows a terminal window titled 'marcos@marcos-VirtualBox: ~'. It displays the output of the 'ab' command, which is benchmarking a server at 'http://192.168.56.101/prueba.html'. The 'ab' command parameters are '-c 100 -n 10000'. The output shows the progress of the test, from 1000 to 9000 requests completed.

```

Entrada Dispositivos Ayuda
marcos@marcos-VirtualBox: ~
66% 80
75% 95
80% 110
90% 147
95% 187
98% 280
99% 312
100% 487 (longest request)
marcos@marcos-VirtualBox:~$ ab -c 100 -n 10000 http://192.168.56.101/prueba.html
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 1000 requests
Completed 2000 requests
Completed 3000 requests
Completed 4000 requests
Completed 5000 requests
Completed 6000 requests
Completed 7000 requests
Completed 8000 requests
Completed 9000 requests

```

Simultaneously, the 'top' command is running in the background, showing the system's CPU usage. The top output includes columns for PID, USUARIO, PR, NI, UIRT, RES, SHR, S, %CPU, %MEM, HORA+, and ORDEN. The CPU usage for the apache2 process (PID 7555) is visible, along with other system processes like ksoftirqd, rcu_bh, and various kernel threads.

Figura 6.37: Ejecutando el comando ab en nuestro sistema operativo anfitrión y monitoreando con el comando 'top' en el servidor.

Y en la Figura 6.38 encontramos los resultados que hemos obtenido con 'ab'.

The screenshot shows the detailed results of the 'ab' test. It provides statistics such as the number of requests, transfer times, and connection times.

```

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

Document Path:        /prueba.html
Document Length:     165461 bytes

Concurrency Level:   100
Time taken for tests: 13.213 seconds
Complete requests:   10000
Failed requests:    0
Total transferred:  1657370000 bytes
HTML transferred:   1654610000 bytes
Requests per second: 756.85 [#/sec] (mean)
Time per request:   132.126 [ms] (mean)
Time per request:   1.321 [ms] (mean, across all concurrent requests)
Transfer rate:       122498.25 [kbytes/sec] received

Connection Times (ms)
              min  mean[+/-sd] median   max
Connect:        0    6   10.5    2    98
Processing:   21   126   95.5   90   661
Waiting:       1    72   54.1    66   324
Total:        25   131  100.5   94   759

```

Figura 6.38: Ejecutando el comando ab en nuestro sistema operativo anfitrión y monitoreando con el comando 'top' en el servidor.

Como podemos ver en estas dos figuras anteriores, destacando los valores más importantes:

- Por parte del servidor, vemos que se crean varios procesos de Apache2, y aproximadamente llega hasta el 31 % de uso de la CPU.
- Por parte de los resultados de 'ab', podemos ver que ha tenido un tiempo total de la ejecución completa de 13 segundos aproximadamente.
- Un tiempo medio de 132 milisegundos aprox. por petición, y un valor medio aproximado de 756 peticiones resueltas por segundo.

Nos ayudamos de la cuestión 4, y vamos a intentar mejorar el servicio HTTP de nuestro servidor. Para ello he obtenido por realizar varios cambios:

- Modificando el archivo apache2.conf que lo encontramos en el directorio '/etc/apache2/apache2.conf'. Modificamos la variable KeepAlive a 'Off', encargada de deshabilitar las conexiones persistentes, es decir, más de una solicitud por conexión. Y la variable KeepAliveTimeout cambiando su valor a '2' para disminuir los tiempos de espera entre una conexión y otra. Como vemos en la Figura 6.39.

```

# This needs to be set in /etc/apache2/envvars
#
PidFile ${APACHE_PID_FILE}
#
# Timeout: The number of seconds before receives and sends time out.
# Timeout 300
#
# KeepAlive: Whether or not to allow persistent connections (more than
# one request per connection). Set to "Off" to deactivate.
# We recommend you leave this number high, for maximum performance.
# KeepAlive Off
#
# MaxKeepAliveRequests: The maximum number of requests to allow
# during a persistent connection. Set to 0 to allow an unlimited amount.
# We recommend you leave this number high, for maximum performance.
# MaxKeepAliveRequests 100
#
# KeepAliveTimeout: Number of seconds to wait for the next request from the
# same client on the same connection.
# KeepAliveTimeout 2
#
# These need to be set in /etc/apache2/envvars
User ${APACHE_RUN_USER}
Group ${APACHE_RUN_GROUP}
"/etc/apache2/apache2.conf" 221L, 7116C escritos
root@ubuntu:/home/marcos$ exit
exit
marcos@ubuntu lun ene 16 2016:"$ vi /etc/apache2/apache2.conf"

```

Figura 6.39: Modificando el archivo de configuración 'apache2.conf'.

Una vez que tengamos todas las modificaciones de mejoras realizadas, debemos de recordar siempre reiniciar los servicios pertenecientes a las modificaciones realizadas para que tengan efecto, en mi caso como he modificado el archivo de configuración 'apache2.conf' voy a reiniciar el servicio como vemos en la Figura 6.40.

```

marcos@ubuntu lun ene 16 2016:"$ sudo su
[sudo] password for marcos:
Lo sentimos, vuelva a intentarlo.
[sudo] password for marcos:
root@ubuntu:/home/marcos$ /etc/init.d/apache2 restart
bash: /etc/init.d/apache2: No existe el archivo o el directorio
root@ubuntu:/home/marcos$ /etc/init.d/apache2 restart
[ ok ] Restarting apache2 (via systemctl): apache2.service.
root@ubuntu:/home/marcos$ "

```

Figura 6.40: Reiniciando el servicio Apache2.

Ahora ya una vez realizadas las mejoras y reiniciado el servicio, vamos a realizar la misma monitorización anterior con la misma carga del comando 'ab', y analizamos los resultados. Podemos ver los resultados en la Figura 6.41 y Figura 6.42.

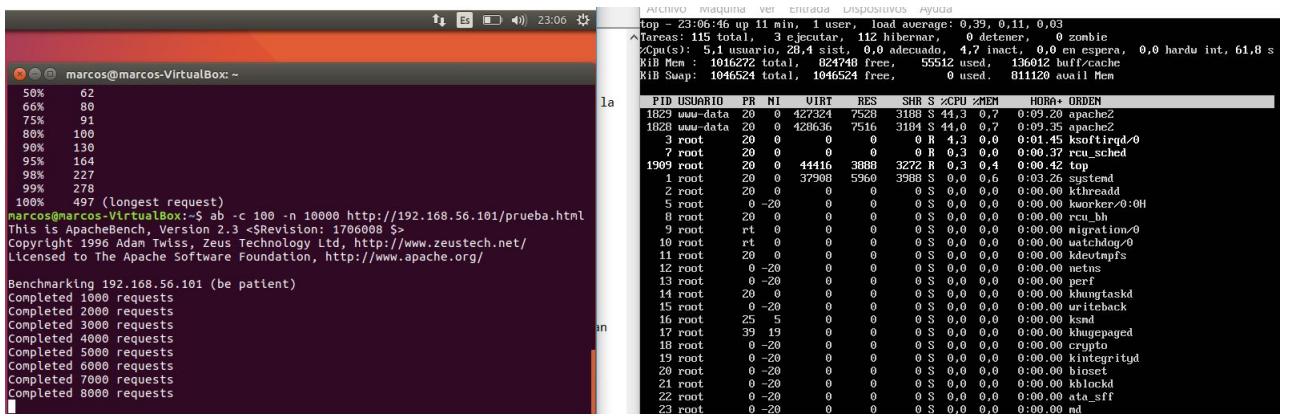


Figura 6.41: Monitorizando la carga del servidor con el comando 'top' mientras se ejecuta la carga de peticiones con el comando 'ab'.

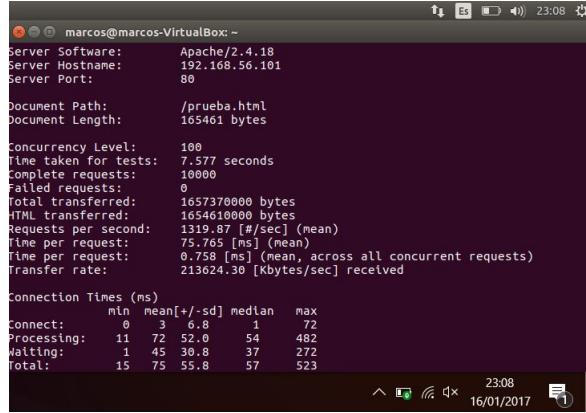


Figura 6.42: Resultados obtenidos en el test de 'ab'.

Comparando los resultados obtenidos antes de la mejora y después de la mejora podemos destacar:

- Referente a la carga de CPU del servidor, hemos obtenido después de la mejora hasta un 44 % aproximadamente frente al 30 % aprox. antes de la mejora del uso de CPU. Podemos decir que después de la mejora hemos sobrecargado un poco más la CPU del servidor, y el uso de la misma.
- El tiempo total que se ha utilizado para la realización del test 'ab' hemos obtenido 7.6 segundos después de la mejora ante los 13.2 segundos antes de la mejora, es decir, hemos obtenido una mejora de aproximadamente 6 segundos. Aunque realmente esta mejora es menos importante para nosotros.

- Aquí llegamos a los resultados más relevantes, donde podemos destacar que antes de la mejora obtuvimos un valor medio de 757 peticiones aproximadamente por segundo, frente a las 1320 peticiones aproximadamente de media por segundo. Es muy considerable esta mejora, ya que podemos hablar de 563 peticiones más de media que atiende nuestro servidor después de la mejora. Dicho de otra manera, nuestro servidor es aproximadamente un 74,39 % mejor que antes.
- El tiempo de respuesta también es significativo, ya que de los 132,17 milisegundos de media por petición antes de la mejora, se convierte en 75,76 milisegundos de media por petición, es decir, nuestro servidor tarda 56,41 milisegundos menos de media por petición después de la mejora. Dicho también de otra manera, nuestro servidor es un 57,32 % aproximadamente más rápido en atender una petición después de la mejora.

Referencias

- [1] Debian 10.1. Puerta de enlace. El manual del Administrador de Debian. <https://debian-handbook.info/browse/es-ES/stable/network-infrastructure.html>. consultado el 9 de Enero de 2017.
- [2] Inc. 3.4. Using the sysctl Command. Red Hat®. https://www.centos.org/docs/5/html/5.1/Deployment_Guide/s1-proc-sysctl.html. consultado el 8 de Enero de 2017.
- [3] Inc. 8.2. Parámetros de redes optimizadas. Tamaño de búfer de recepción de socket. Red Hat®. https://access.redhat.com/documentation/es-ES/Red_Hat_Enterprise_Linux/6/html/Performance_Tuning_Guide/s-network-dont-adjust-defaults.html. consultado el 8 de Enero de 2017.
- [4] Desempeño de Apache. Documentación Moodle. https://docs.moodle.org/all/es/Recomendaciones_sobre_desempe%C3%B3n. consultado el 9 de Enero de 2017.
- [5] Copia de seguridad del registro. © 2017 Microsoft. <https://support.microsoft.com/es-es/kb/322756>. consultado el 9 de Enero de 2017.
- [6] Configuring HTTP Compression in IIS 7. © 2017 Microsoft. <https://technet.microsoft.com/en-us/library/30e64245-ec6b-4c4e-891b-b7249e36c0a0>. consultado el 9 de Enero de 2017.
- [7] Download Ubuntu Server 16.04 LTS. © 2017 Canonical Ltd. <https://www.ubuntu.com/download/server/thank-you?country=ES&version=16.04.1&architecture=amd64>. consultado el 10 de Enero de 2017.
- [8] Configure Compression (IIS 7). © 2017 Microsoft. <https://technet.microsoft.com/en-us/library/cc730629>. consultado el 9 de Enero de 2017.

- [9] sysctl(8) Linux man page. <https://linux.die.net/man/8/sysctl>. consultado el 8 de Enero de 2017.
- [10] wget(1) Linux man page. <https://linux.die.net/man/1/wget>. consultado el 20 de Enero de 2017.