

## Práctica 3. Monitorización de servicios.

---

Manuel Jiménez Molina

3 de febrero de 2017

## Índice

1. 1.a) ¿Qué archivo le permite ver qué programas se han instalado con el gestor de paquetes? 1.b) ¿Qué significan las terminaciones .1.gz o .2.gz de los archivos en ese directorio?	5
1.1. a) ¿Qué archivo le permite ver qué programas se han instalado con el gestor de paquetes? . . . . .	5
1.2. b) ¿Qué significan las terminaciones .1.gz o .2.gz de los archivos en ese directorio? . . . . .	6
2. ¿qué archivo ha de modificar para programar una tarea?Escriba la línea necesaria para ejecutar una vez al día una copia del directorio /codigo a /seguridad/\$fecha donde \$fecha es la fecha actual(puede usar el comando date).	7
3. Pruebe a ejecutar el comando, conectar un dispositivo USB y vuelva a ejecutar el comando. Copie y pegue la salida del comando.(considere usar dmesg   tail). Comente qué observa en la información mostrada.	8
4. Ejecute el monitor de “System Performance” y muestre el resultado. Incluya capturas de pantalla comentando la información que aparece.	9
5. Cree un recopilador de datos definido por el usuario (modo avanzado) que incluya tanto el contador de rendimiento como los datos de seguimiento:Todos los referentes al procesador, al proceso y al servicio web. Intervalo de muestra 15 segundos. Almacene el resultado en el directorio Escritorio /logs. Incluya las capturas de pantalla de cada paso.	15
6. Visite la web del proyecto y acceda a la demo que proporcionan ( <a href="http://demo.munin-monitoring.org/">http://demo.munin-monitoring.org/</a> ) donde se muestra cómo monitorizan un servidor. Monitoree varios parámetros y haga capturas de pantalla de lo que está mostrando comentando qué observa.	28
7. Escriba un breve resumen sobre alguno de los artículos donde se muestra el uso de strace o busque otro y coméntelo.	32
8. Escriba un script en Python o PHP y analice su comportamiento usando el profiler presentado.	33
9. Acceda a la consola mysql (o a través de phpMyAdmin) y muestre el resultado de mostrar el "profile" de una consulta (la creación de la BD y la consulta la puede hacer libremente).	34
10.Opcional 1: Indique qué comandos ha utilizado para realizarlo así como capturas de pantalla del proceso de reconstrucción del RAID.	37

11. Opcional 2: Instale Nagios en su sistema (el que prefiera) documentando el proceso y muestre el resultado de la monitorización de su sistema comentando qué aparece.	37
12. Opcional 3: Pruebe a instalar este monitor en alguno de sus tres sistemas. Realice capturas de pantalla del proceso de instalación y comente capturas de pantalla del programa en ejecución.	37
13. Opcional 4: Pruebe a instalar este monitor en alguno de sus tres sistemas. Realice capturas de pantalla del proceso de instalación y comente capturas de pantalla del programa en ejecución.	37
14. Opcional 5: Instale el monitor. Muestre y comente algunas capturas de pantalla.	45

## Índice de figuras

1.1. Archivo history.log . . . . .	6
1.2. Comparando antigüedad archivos de log apt . . . . .	7
2.1. Script para crontab . . . . .	7
2.2. Modificando crontab . . . . .	8
3.1. Desconectando ratón inalámbrico por usb (Logitech) . . . . .	8
3.2. Conectando ratón inalámbrico por usb (Logitech) . . . . .	9
4.1. Abrir Perfmon con PowerShell . . . . .	10
4.2. Monitor System Performance . . . . .	11
4.3. Perfmon. Resumen de disco físico . . . . .	11
4.4. Perfmon. Resumen de información del procesador . . . . .	12
4.5. Perfmon. Resumen de interfaz de red y memoria . . . . .	12
4.6. Perfmon. Monitor de rendimiento . . . . .	13
4.7. Perfmon. Recopilador de diagnóstico del sistema . . . . .	14
4.8. Perfmon. Recopilador de rendimiento del sistema . . . . .	15
5.1. Paso 1. Crear nuevo recopilador de datos . . . . .	16
5.2. Paso 2. Establecer nombre y comenzar configuración manual del recopilador de datos . . . . .	17
5.3. Paso 3. Incluir registros de datos en el recopilador . . . . .	18
5.4. Paso 4. Registrar contadores de rendimiento . . . . .	19
5.5. Paso 5. Agregar contadores de rendimiento . . . . .	20
5.6. Paso 6. Finalizar agregación de contadores de rendimiento . . . . .	21
5.7. Paso 7. Habilitar proveedores de seguimiento . . . . .	22
5.8. Paso 8. Ruta para guardar los datos de recopilador . . . . .	23
5.9. Paso 9. Finalizar creación del recopilador de datos . . . . .	24
5.10. Comprobar la creación del recopilador de datos . . . . .	25
5.11. Gráfico generado por el recopilador de datos . . . . .	26
5.12. Gráfico generado solo para servicio web . . . . .	27

5.13. Gráfico generado solo para E/S de procesos . . . . .	28
6.1. Gráfico de estado de los procesos durante un día . . . . .	29
6.2. Uso de disco del servidor durante un mes . . . . .	30
6.3. Uso de memoria del servidor durante un mes . . . . .	31
6.4. Información de la tabla de memoria . . . . .	32
7.1. Uso de strace para ls . . . . .	33
8.1. Script en Python para profiler . . . . .	33
8.2. Ejecutando profiler en Python . . . . .	34
9.1. Iniciar mysql . . . . .	34
9.2. Crear base de datos en mysql . . . . .	34
9.3. Usar una base de datos existente en mysql . . . . .	35
9.4. Ver tablas de una base de datos en mysql . . . . .	35
9.5. Crear tabla en una base de datos en mysql . . . . .	35
9.6. Insertar una fila en una tabla en mysql . . . . .	35
9.7. Insertar otra fila en una tabla en mysql . . . . .	35
9.8. Ver tabla mascota en mysql . . . . .	35
9.9. Ver profiles en mysql . . . . .	36
13.1. Configurar la base de datos para cacti . . . . .	38
13.2. Crear contraseña de administración para BD . . . . .	39
13.3. contraseña de MySQL para Cacti . . . . .	40
13.4. Confirmación de contraseña de MySQL . . . . .	41
13.5. Paso 1 guía instalación Cacti . . . . .	42
13.6. Paso 2 guía instalación Cacti . . . . .	42
13.7. Paso 3 guía instalación Cacti . . . . .	43
13.8. Interfaz de Cacti . . . . .	44
13.9. Gráfica de carga media de Cacti . . . . .	44
13.10gráfica de procesos de Cacti . . . . .	45
13.11Gráfica de usuarios conectados de Cacti . . . . .	45

## Índice de tablas

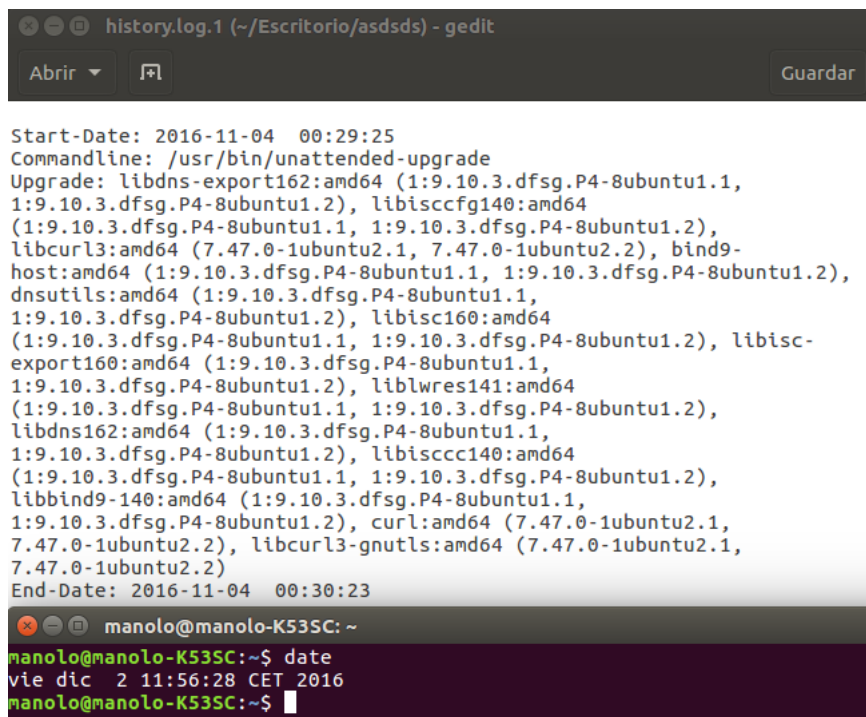
**1. 1.a) ¿Qué archivo le permite ver qué programas se han instalado con el gestor de paquetes? 1.b) ¿Qué significan las terminaciones .1.gz o .2.gz de los archivos en ese directorio?**

**1.1. a) ¿Qué archivo le permite ver qué programas se han instalado con el gestor de paquetes?**

El directorio `/proc`[9, 11, 5] contiene un sistema de archivos virtual. No contiene archivos reales, sino información del sistema en tiempo de ejecución, sobretodo de procesos.

Por otra parte, el directorio `/var`[10, 12] contiene datos como de archivos de registro del sistema, correo e impresión o por ejemplo archivos temporales y pasajeros. Contiene un directorio con archivos de log:

- `/var/log`: Contiene archivos de log de diferentes programas. Debido a esto pueden crecer indefinidamente. Este directorio contiene archivos como:
  - `/var/log/wtmp`: Contiene los archivos de login, los cuales registran todos los inicios y cierres de sesión en el sistema.
  - `/var/log/messages`: Contiene archivos syslog, los cuales almacenan mensajes del núcleo y programas del sistema.
  - `/var/log/apt/term.log`[8]: Contiene log de la actividad de apt.
  - `/var/log/apt/history.log`[1, 2]: Contiene el log de los paquetes instalados con el gestor de paquete apt. Este es el archivo que buscábamos.



The image shows a gedit window titled 'history.log.1 (~/Escritorio/asdsds) - gedit'. The window contains the following text:

```
Start-Date: 2016-11-04 00:29:25
Commandline: /usr/bin/unattended-upgrade
Upgrade: libdns-export162:amd64 (1:9.10.3.dfsg.P4-8ubuntu1.1,
1:9.10.3.dfsg.P4-8ubuntu1.2), libiscfg140:amd64
(1:9.10.3.dfsg.P4-8ubuntu1.1, 1:9.10.3.dfsg.P4-8ubuntu1.2),
libcurl3:amd64 (7.47.0-1ubuntu2.1, 7.47.0-1ubuntu2.2), bind9-
host:amd64 (1:9.10.3.dfsg.P4-8ubuntu1.1, 1:9.10.3.dfsg.P4-8ubuntu1.2),
dnsutils:amd64 (1:9.10.3.dfsg.P4-8ubuntu1.1,
1:9.10.3.dfsg.P4-8ubuntu1.2), libisc160:amd64
(1:9.10.3.dfsg.P4-8ubuntu1.1, 1:9.10.3.dfsg.P4-8ubuntu1.2), libisc-
export160:amd64 (1:9.10.3.dfsg.P4-8ubuntu1.1,
1:9.10.3.dfsg.P4-8ubuntu1.2), liblwres141:amd64
(1:9.10.3.dfsg.P4-8ubuntu1.1, 1:9.10.3.dfsg.P4-8ubuntu1.2),
libdns162:amd64 (1:9.10.3.dfsg.P4-8ubuntu1.1,
1:9.10.3.dfsg.P4-8ubuntu1.2), libisc140:amd64
(1:9.10.3.dfsg.P4-8ubuntu1.1, 1:9.10.3.dfsg.P4-8ubuntu1.2),
libbind9-140:amd64 (1:9.10.3.dfsg.P4-8ubuntu1.1,
1:9.10.3.dfsg.P4-8ubuntu1.2), curl:amd64 (7.47.0-1ubuntu2.1,
7.47.0-1ubuntu2.2), libcurl3-gnutls:amd64 (7.47.0-1ubuntu2.1,
7.47.0-1ubuntu2.2)
End-Date: 2016-11-04 00:30:23
```

Below the gedit window is a terminal window titled 'manolo@manolo-K53SC: ~'. The terminal shows the following commands and output:

```
manolo@manolo-K53SC:~$ date
vie dic  2 11:56:28 CET 2016
manolo@manolo-K53SC:~$
```

Figura 1.1: Archivo history.log

## 1.2. b) ¿Qué significan las terminaciones .1.gz o .2.gz de los archivos en ese directorio?

Según las referencias del apartado a), los archivos de log son mantenidos debido a pueden ocupar bastante espacio, sobre todo si fuese un servidor. Crecerían mucho y a la hora de buscar información en un archivo de log sería complicado. Existe un proceso en Linux que rota los archivos de log, renombrando y escribiendo en un nuevo archivo de log. Esto se realiza gracias a una tarea de cron[3] (ejecuta tareas cada cierto tiempo), que utiliza logrotate[7] (rota y comprime registros del sistema). De este modo, los archivos son rotados y comprimidos según su antigüedad, evitando que un archivo de log sea muy grande. Así por ejemplo .1.gz tendría información más actualizada que .2.gz.

Comprobamos el contenido de 2 archivos de log y vemos que aquel que tiene un número con extensión menor es el más actualizado.

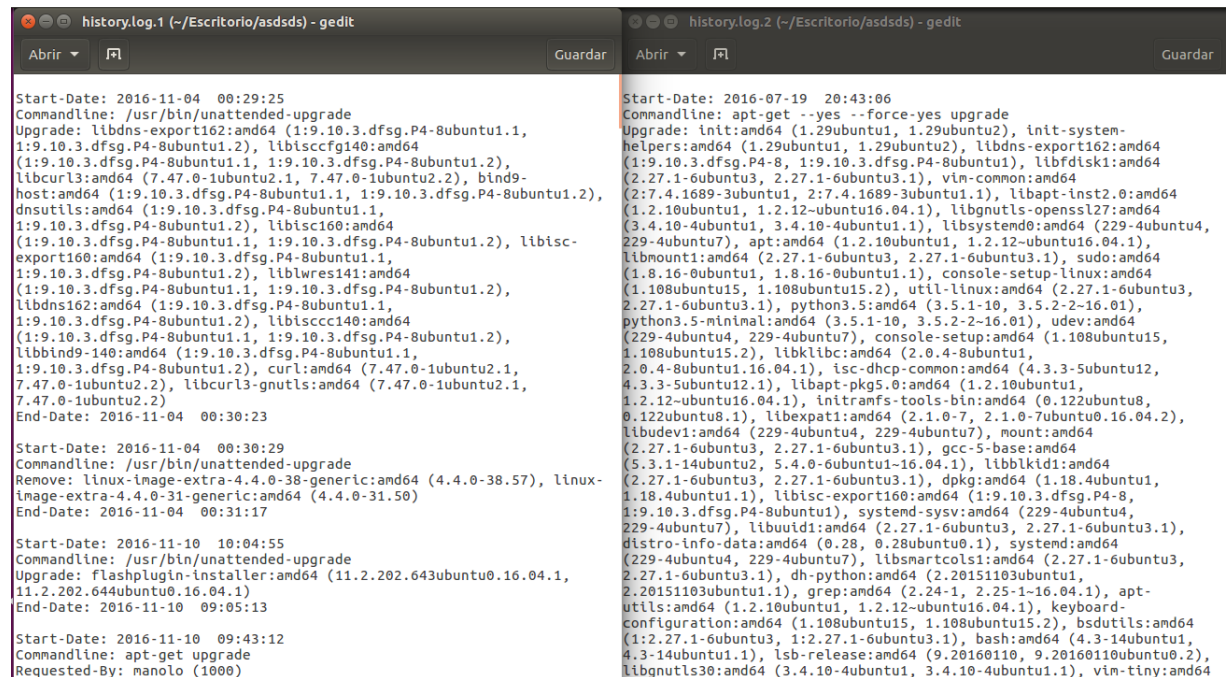


Figura 1.2: Comparando antigüedad archivos de log apt

## 2. ¿qué archivo ha de modificar para programar una tarea? Escriba la línea necesaria para ejecutar una vez al día una copia del directorio /codigo a /seguridad/\$fecha donde \$fecha es la fecha actual(puede usar el comando date).

El archivo a modificar es en /etc/crontab o ejecutando crontab -e[4].  
Escribimos un script llamado ejercicio2.sh para lo que nos piden:

```
GNU nano 2.2.6 Archivo: ejercicio2.sh
#!/bin/bash
fecha='date +%m-%d-%y'
directorio_origen="/codigo/*"
directorio_destino="/seguridad/$fecha"
mkdir $directorio_destino
cp $directorio_origen $directorio_destino
```

Figura 2.1: Script para crontab  
Modificamos crontab para que ejecute el script:

```
GNU nano 2.2.6 Archivo: /etc/crontab Modifi
# /etc/crontab: system-wide crontab
# Unlike any other crontab you don't have to run the `crontab`
# command to install the new version when you edit this file
# and files in /etc/cron.d. These files also have username fields,
# that none of the other crontabs do.

SHELL=/bin/sh
PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin

# m h dom mon dow user  command
17 * * * * root    cd / && run-parts --report /etc/cron.hourly
25 6 * * * root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.da
47 6 * * 7 root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.we
52 6 1 * * root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.mo
0 0 * * * /home/manolo/ejercicio2.sh
-
#
```

Figura 2.2: Modificando crontab

3. Pruebe a ejecutar el comando, conectar un dispositivo USB y vuelva a ejecutar el comando. Copie y pegue la salida del comando.(considere usar `dmesg | tail`). Comente qué observa en la información mostrada.

El comando `dmesg[6]` muestra información sobre el buffer que está conectado con el kernel. Usando el comando `tail` vemos los últimos acontecimientos que pasan en el kernel:

```
manolo@manolo-K53SC: ~
manolo@manolo-K53SC:~$ dmesg | tail
[ 4831.663511] usb 2-1.2: new full-speed USB device number 4 using ehci-pci
[ 4831.759452] usb 2-1.2: New USB device found, idVendor=046d, idProduct=c52f
[ 4831.759462] usb 2-1.2: New USB device strings: Mfr=1, Product=2, SerialNumber
=0
[ 4831.759467] usb 2-1.2: Product: USB Receiver
[ 4831.759471] usb 2-1.2: Manufacturer: Logitech
[ 4831.762620] input: Logitech USB Receiver as /devices/pci0000:00/0000:00:1d.0/
usb2/2-1/2-1.2/2-1.2:1.0/0003:046D:C52F.0003/input/input21
[ 4831.763211] hid-generic 0003:046D:C52F.0003: input,hidraw0: USB HID v1.11 Mou
se [Logitech USB Receiver] on usb-0000:00:1d.0-1.2/input0
[ 4831.766158] input: Logitech USB Receiver as /devices/pci0000:00/0000:00:1d.0/
usb2/2-1/2-1.2/2-1.2:1.1/0003:046D:C52F.0004/input/input22
[ 4831.872377] hid-generic 0003:046D:C52F.0004: input,hiddev0,hidraw1: USB HID v
1.11 Device [Logitech USB Receiver] on usb-0000:00:1d.0-1.2/input1
[ 6571.865665] usb 2-1.2: USB disconnect, device number 4
manolo@manolo-K53SC:~$

manolo@manolo-K53SC: ~
manolo@manolo-K53SC:~$ date
jue dic  8 16:57:07 CET 2016
```

Figura 3.1: Desconectando ratón inalámbrico por usb (Logitech)

Vemos como nos dice que el dispositivo USB 2-1-2 está desconectado. Ahora probamos a conectarlo:



```
manolo@manolo-K53SC: ~  
manolo@manolo-K53SC:~$ dmesg | tail  
[ 6571.865665] usb 2-1.2: USB disconnect, device number 4  
[ 6683.143054] usb 2-1.2: new full-speed USB device number 5 using ehci-pci  
[ 6683.238921] usb 2-1.2: New USB device found, idVendor=046d, idProduct=c52f  
[ 6683.238934] usb 2-1.2: New USB device strings: Mfr=1, Product=2, SerialNumber  
=0  
[ 6683.239048] usb 2-1.2: Product: USB Receiver  
[ 6683.239053] usb 2-1.2: Manufacturer: Logitech  
[ 6683.244823] input: Logitech USB Receiver as /devices/pci0000:00/0000:00:1d.0/  
usb2/2-1/2-1.2/2-1.2:1.0/0003:046D:C52F.0005/input/input23  
[ 6683.245257] hid-generic 0003:046D:C52F.0005: input,hidraw0: USB HID v1.11 Mou  
se [Logitech USB Receiver] on usb-0000:00:1d.0-1.2/input0  
[ 6683.247994] input: Logitech USB Receiver as /devices/pci0000:00/0000:00:1d.0/  
usb2/2-1/2-1.2/2-1.2:1.1/0003:046D:C52F.0006/input/input24  
[ 6683.355991] hid-generic 0003:046D:C52F.0006: input,hiddev0,hidraw1: USB HID v  
1.11 Device [Logitech USB Receiver] on usb-0000:00:1d.0-1.2/input1  
manolo@manolo-K53SC:~$  
  
manolo@manolo-K53SC: ~  
manolo@manolo-K53SC:~$ date  
jue dic  8 16:58:19 CET 2016  
manolo@manolo-K53SC:~$
```

Figura 3.2: Conectando ratón inalámbrico por usb (Logitech)

Tras conectarlo, no nos muestra ningún mensaje de que esté desconectado, por lo que todo funciona correctamente.

#### 4. Ejecute el monitor de “System Performance” y muestre el resultado. Incluya capturas de pantalla comentando la información que aparece.

Iniciamos Perfmon desde PowerShell. Para ello escribimos en la consola la orden perfmon.

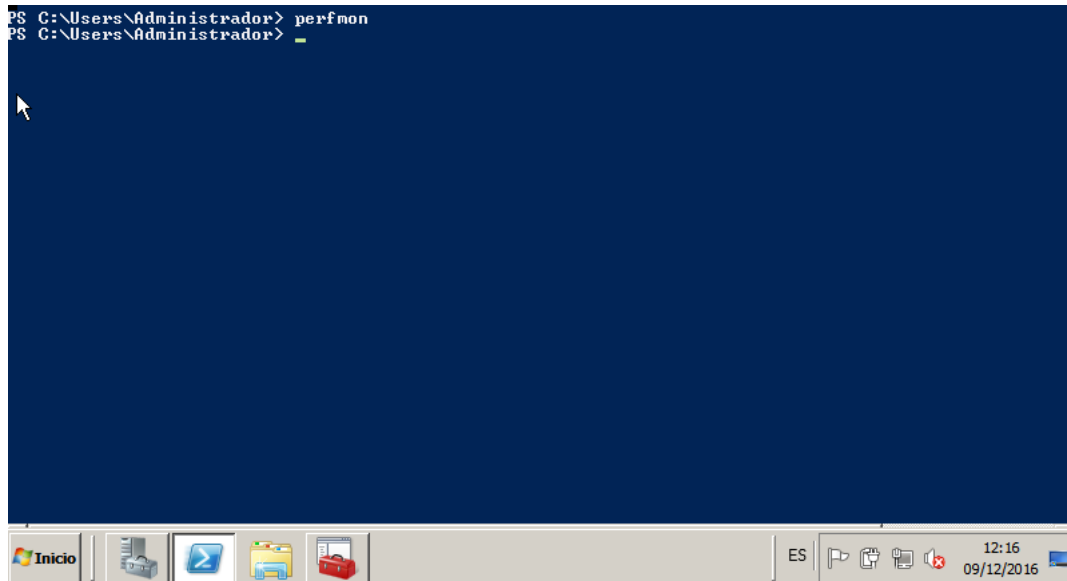


Figura 4.1: Abrir Perfmon con PowerShell

Se abrirá el monitor "System Performance" de Windows Server y podemos ver cierta información.

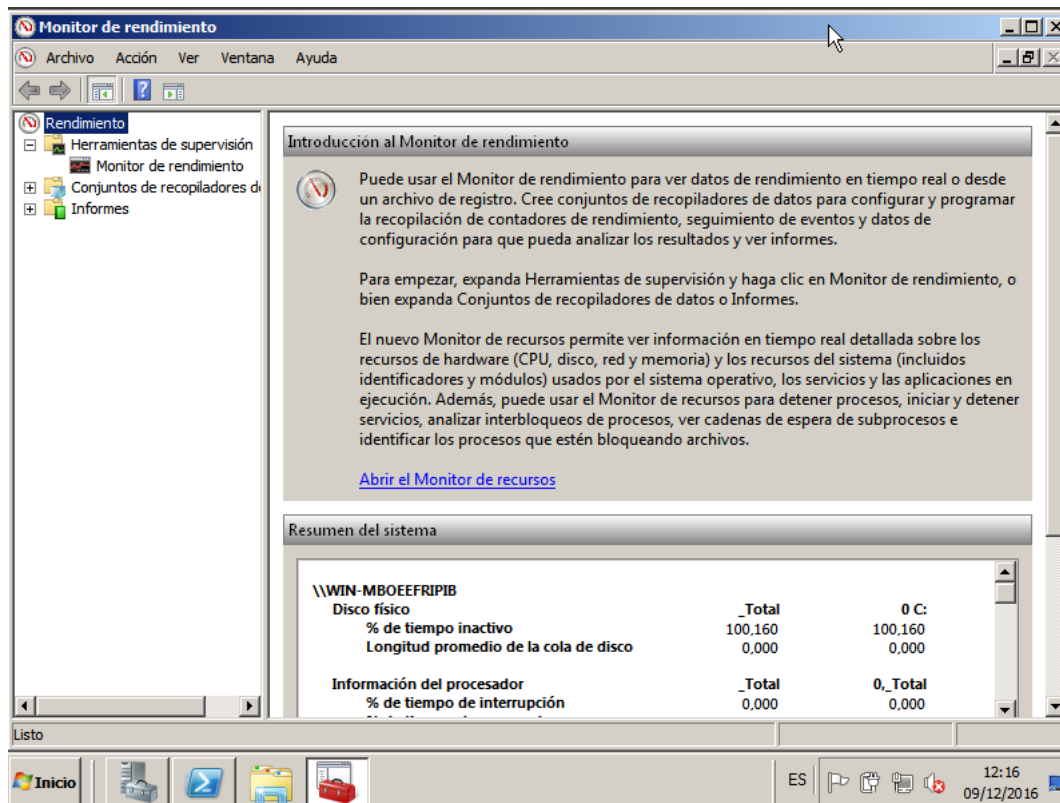


Figura 4.2: Monitor System Performance

Tenemos un resumen del sistema donde nos muestra información sobre:

- Disco físico. Muestra por ejemplo el porcentaje de tiempo inactivo.

**Resumen del sistema**

\\WIN-MBOEEFRIPIB		
<b>Disco físico</b>		
% de tiempo inactivo	_Total	0 C:
100,180	100,180	
Longitud promedio de la cola de disco	0,000	0,000
<b>Información del procesador</b>		
% de tiempo de interrupción	_Total	0, _Total
0,000	0,000	0,000
% de tiempo de procesador	0,000	0,000

Figura 4.3: Perfmon. Resumen de disco físico

- Información del procesador. En este caso la foto muestra porcentajes de tiempos de interrupción y de procesador.

Información del procesador	_Total	0._Total
% de tiempo de interrupción	0,000	0,000
% de tiempo de procesador	0,000	0,000
Estado de detención	0,000	0,000
Interfaz de red	Adaptador de escritorio Intel[R] PRO_1000 MT	
Total de bytes/s		0,000

Figura 4.4: Perfmon. Resumen de información del procesador

- Interfaz de red y memoria. Para la red se muestran los bytes totales usados, pero son 0 porque no estamos haciendo uso de la red. Por otro lado, la memoria por ejemplo nos indica el porcentaje de bytes en uso, que en este caso son 50,662.

Resumen del sistema		
Estado de detención	0,000	0,000
Interfaz de red	Adaptador de escritorio Intel[R] PRO_1000 MT	
Total de bytes/s		0,000
Memoria		
% de bytes confirmados en uso	30,662	
Errores de caché/s	0,000	
Mbytes disponibles	93,000	

Figura 4.5: Perfmon. Resumen de interfaz de red y memoria

Perfmon tiene un monitor de rendimiento que nos muestra de forma gráfica la información total del procesador/tiempo de procesador. Vemos como en el eje X nos muestra la hora actual y en el eje Y nos muestra el uso del procesador. Por ejemplo si abrimos el navegador o realizamos alguna acción, el cambio en la gráfica será notable.

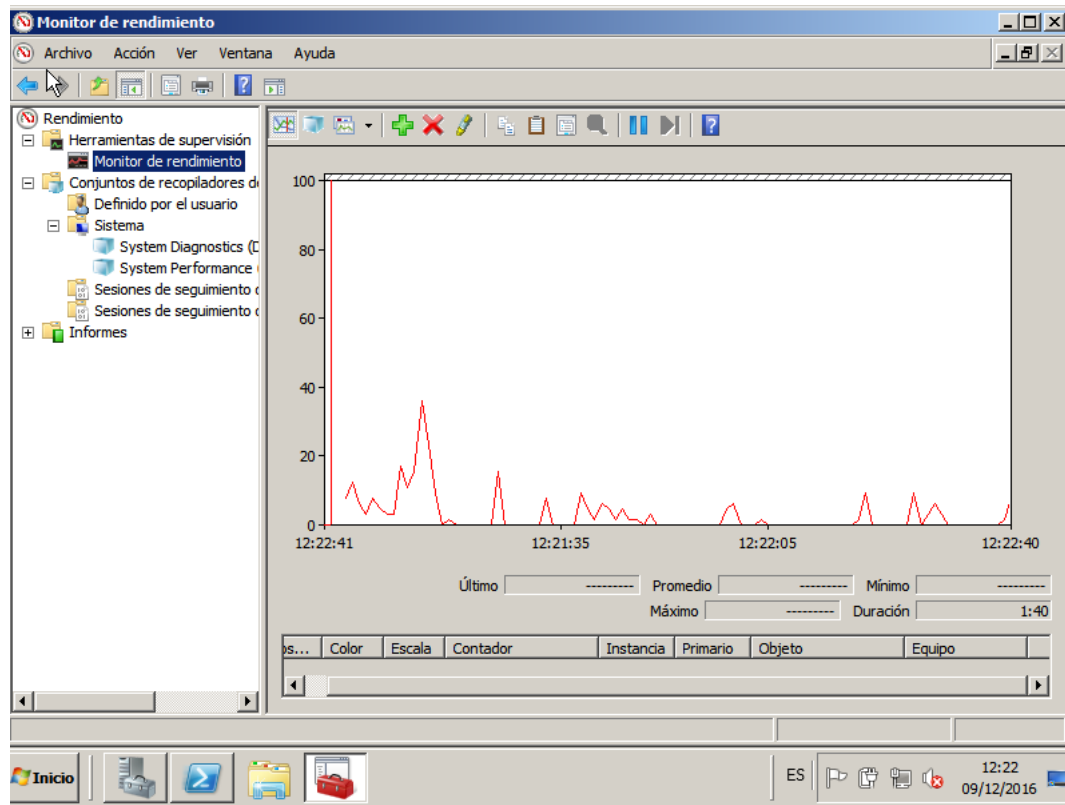


Figura 4.6: Perfmon. Monitor de rendimiento

También tenemos recopiladores de datos del sistema, que son:

- Diagnóstico del sistema.

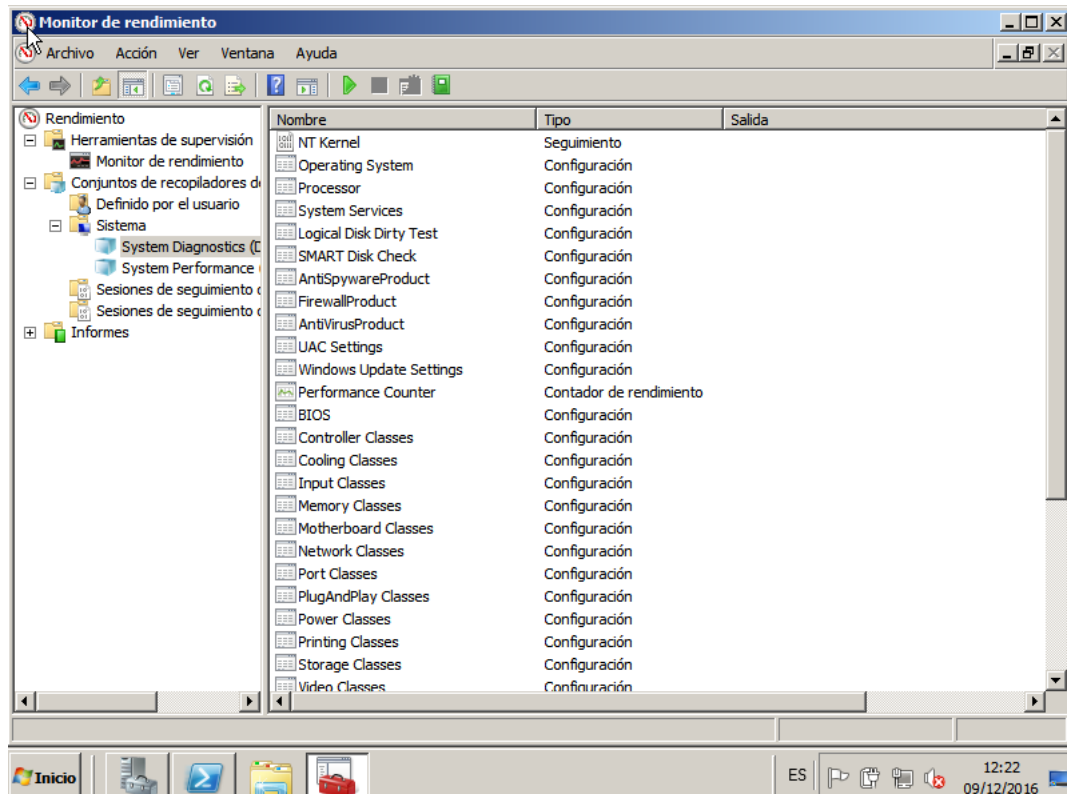


Figura 4.7: Perfmon. Recopilador de diagnóstico del sistema

- Rendimiento del sistema.

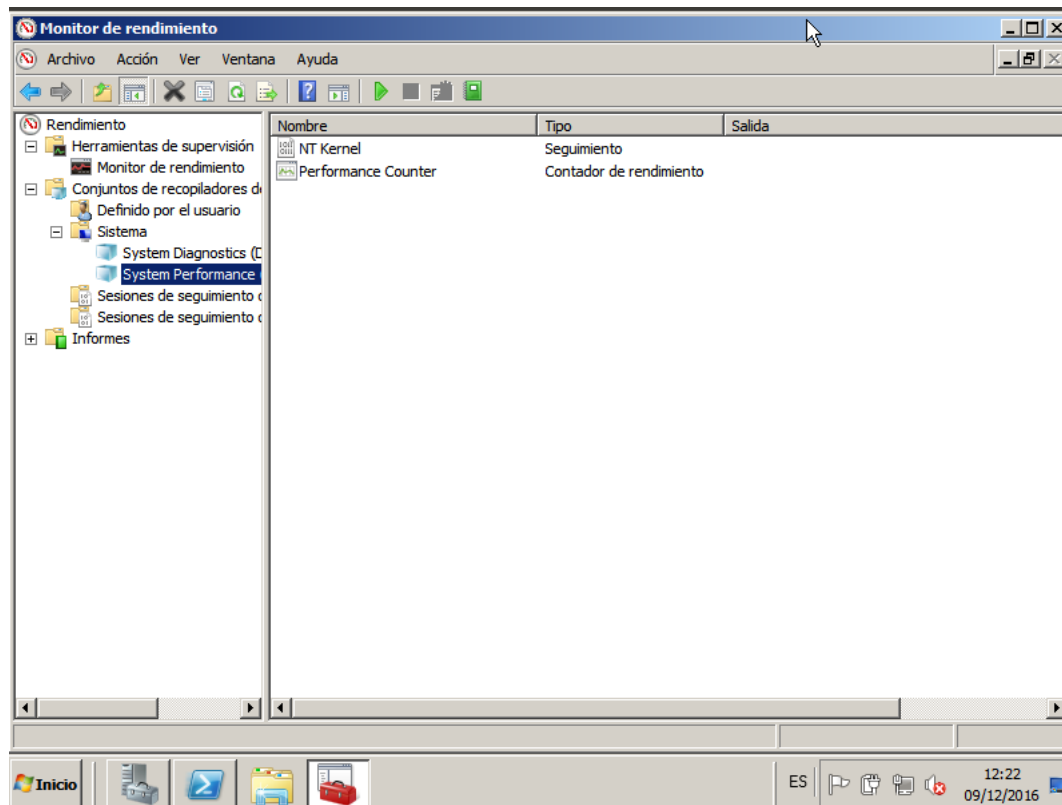


Figura 4.8: Perfmon. Recopilador de rendimiento del sistema

5. Cree un recopilador de datos definido por el usuario (modo avanzado) que incluya tanto el contador de rendimiento como los datos de seguimiento: Todos los referentes al procesador, al proceso y al servicio web. Intervalo de muestra 15 segundos. Almacene el resultado en el directorio Escritorio /logs. Incluya las capturas de pantalla de cada paso.

Vamos a ir paso por paso explicando como hacer un recopilador de datos:

- **Paso 1:** Hacemos click derecho sobre conjuntos de recopiladores del sistema > definidos por el usuario. Nos aparecerá la opción para nuevo > conjunto de recopiladores de datos. Pulsamos y empezamos a configurar el recopilador.

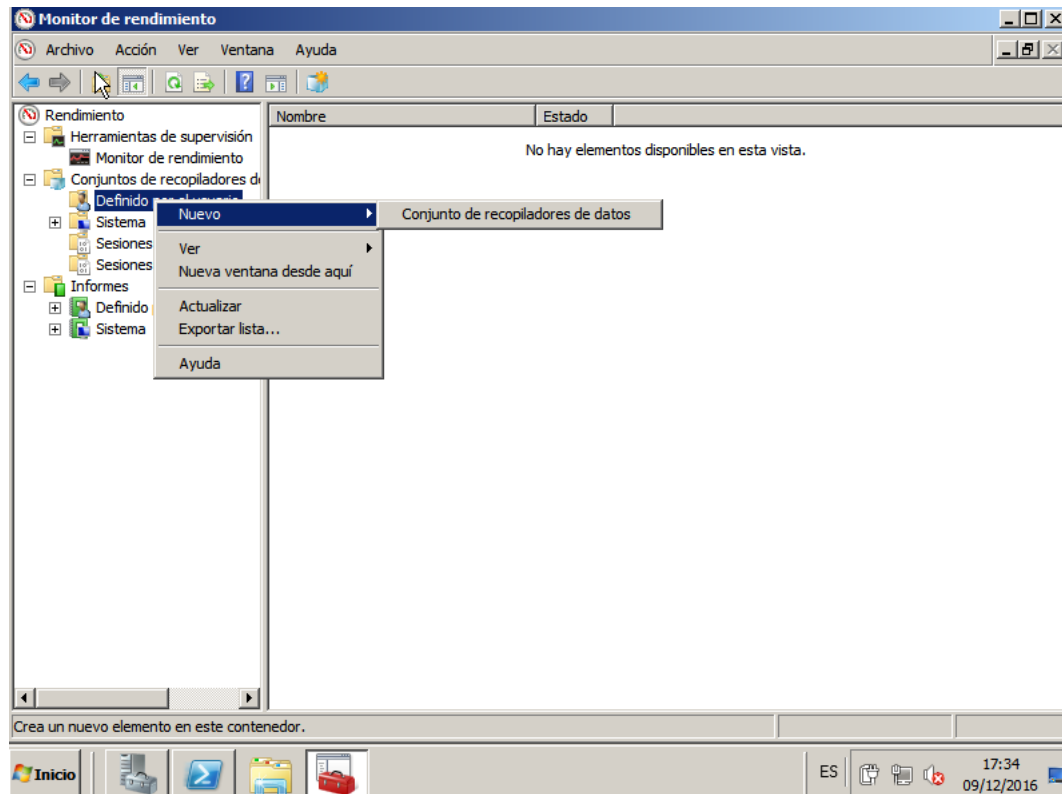


Figura 5.1: Paso 1. Crear nuevo recopilador de datos

- **Paso 2:** Elegimos un nombre para el recopilador de datos, por ejemplo Copilador de datos ISEz seleccionamos la opción de Crear manualmente(avanzado)".



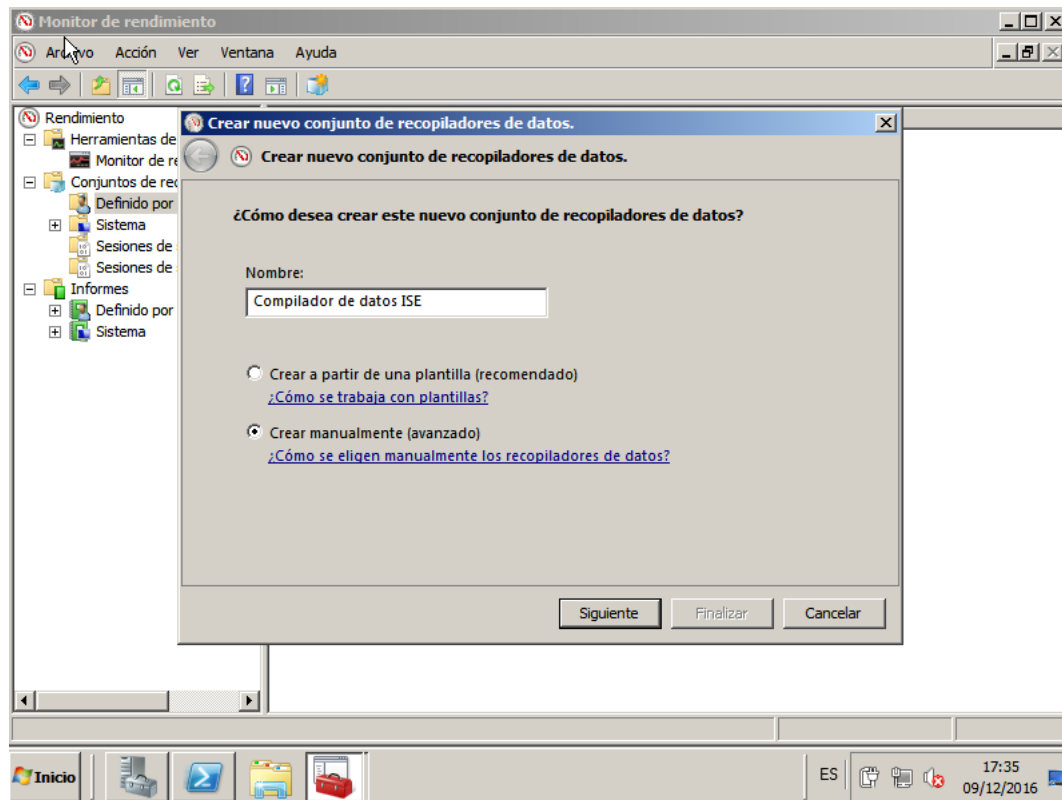


Figura 5.2: Paso 2. Establecer nombre y comenzar configuración manual del recopilador de datos

- **Paso 3:** Incluimos los datos que queremos para el recopilador. En este caso, el ejercicio nos pide que elegir Contador de rendimiento "Datos de seguimiento de eventos", por lo que elegimos estos.

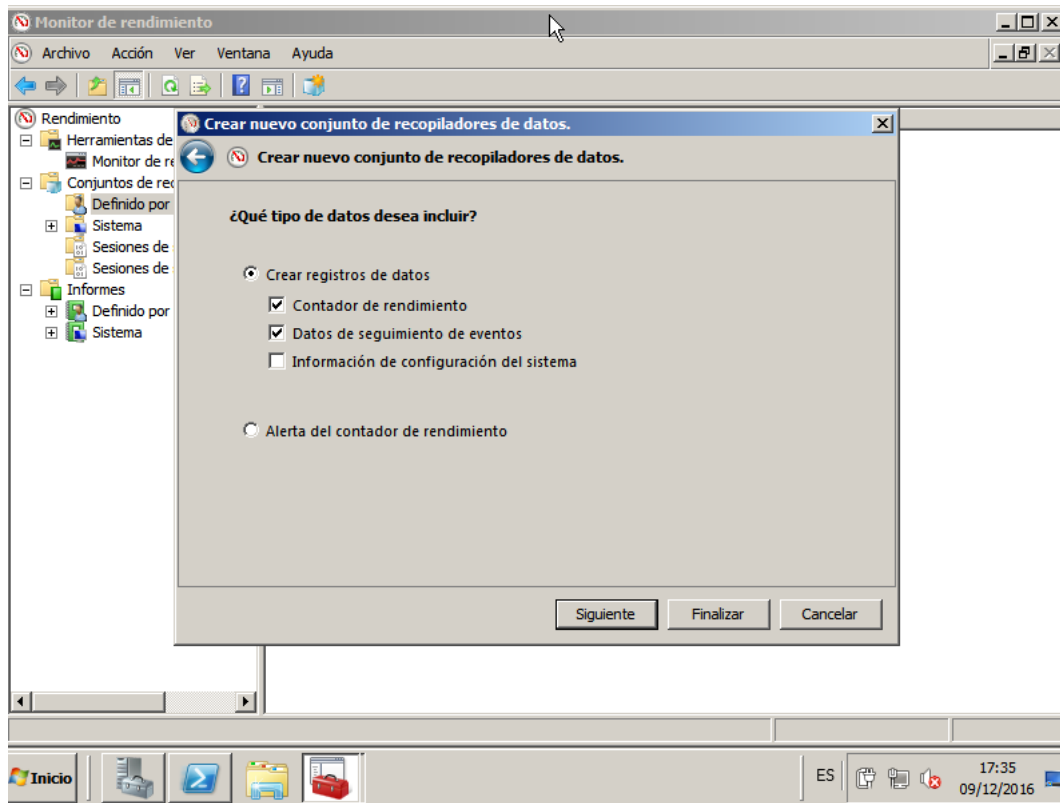


Figura 5.3: Paso 3. Incluir registros de datos en el recopilador

- **Paso 4:** Nos aparece una nueva pantalla para incluir contadores de rendimiento al recopilador de datos. Pulsamos en agregar para realizarlo.

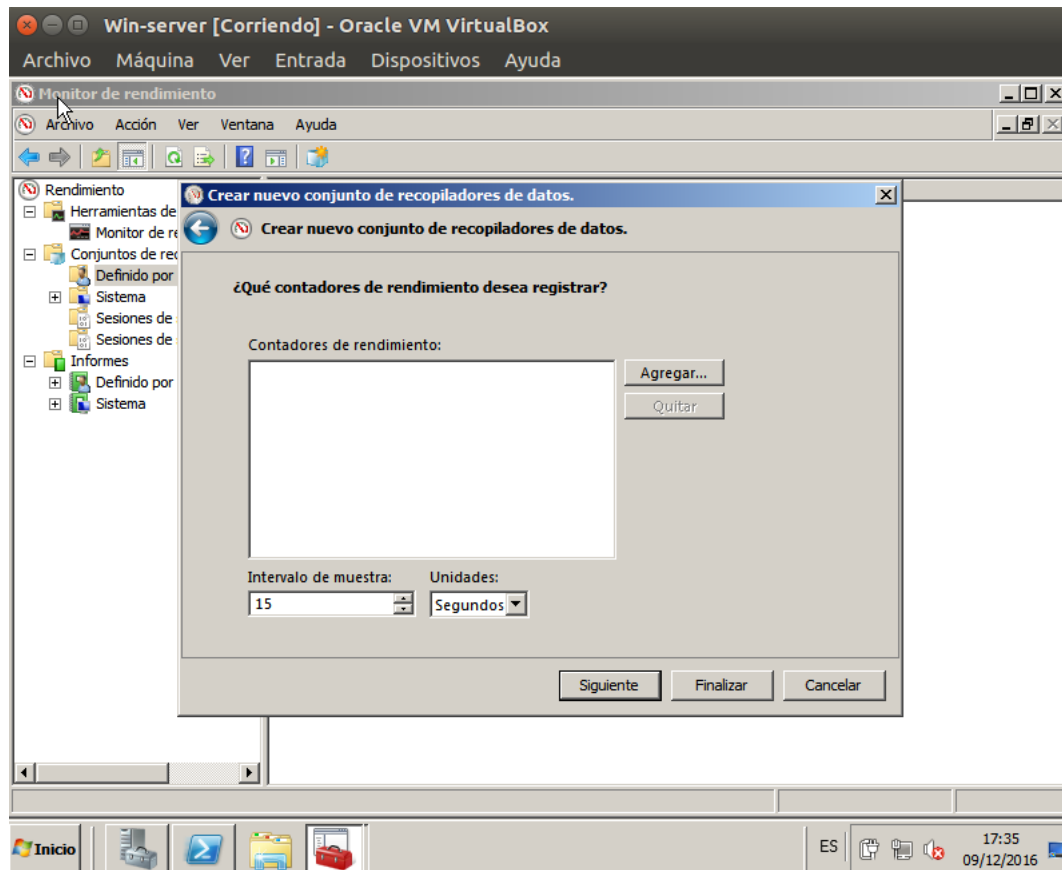


Figura 5.4: Paso 4. Registrar contadores de rendimiento

- **Paso 5:** Agregamos los contadores de procesador, proceso y servicio web. Aceptamos para terminar.

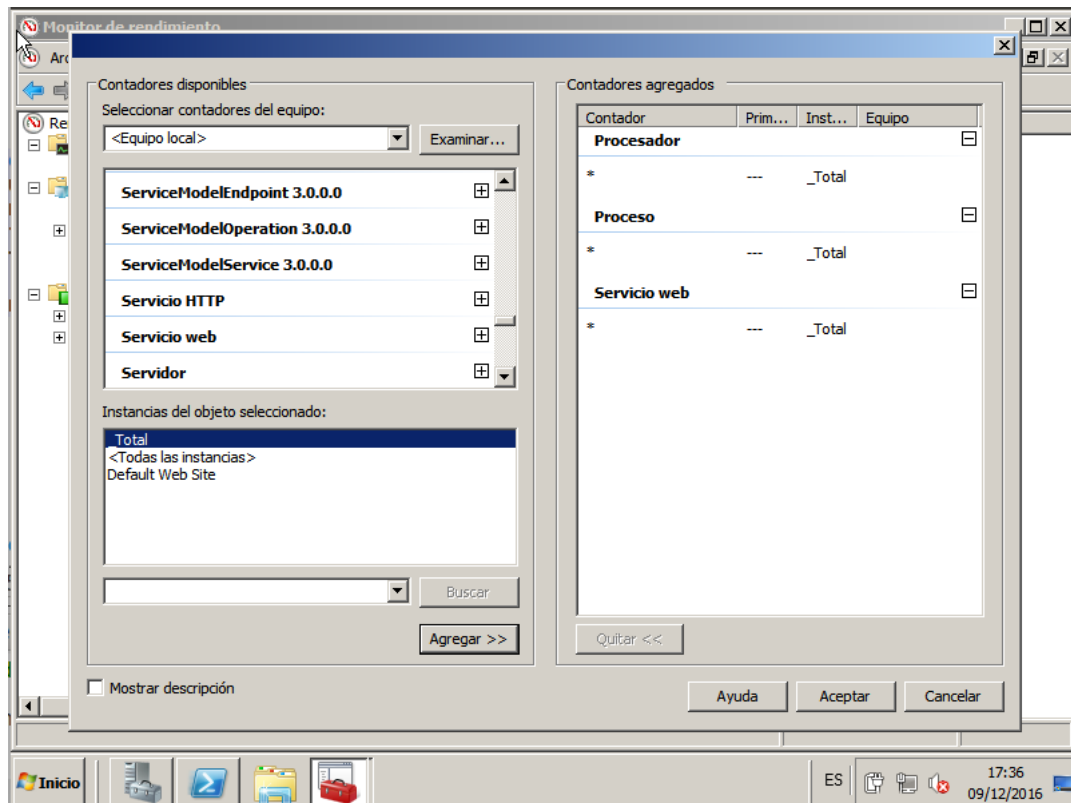


Figura 5.5: Paso 5. Agregar contadores de rendimiento

- **Paso 6:** Comprobamos que los contadores que agregamos están presentes. Establecemos el intervalo de muestra a 15s como pide en el guión de prácticas y continuamos y pulsamos siguiente.

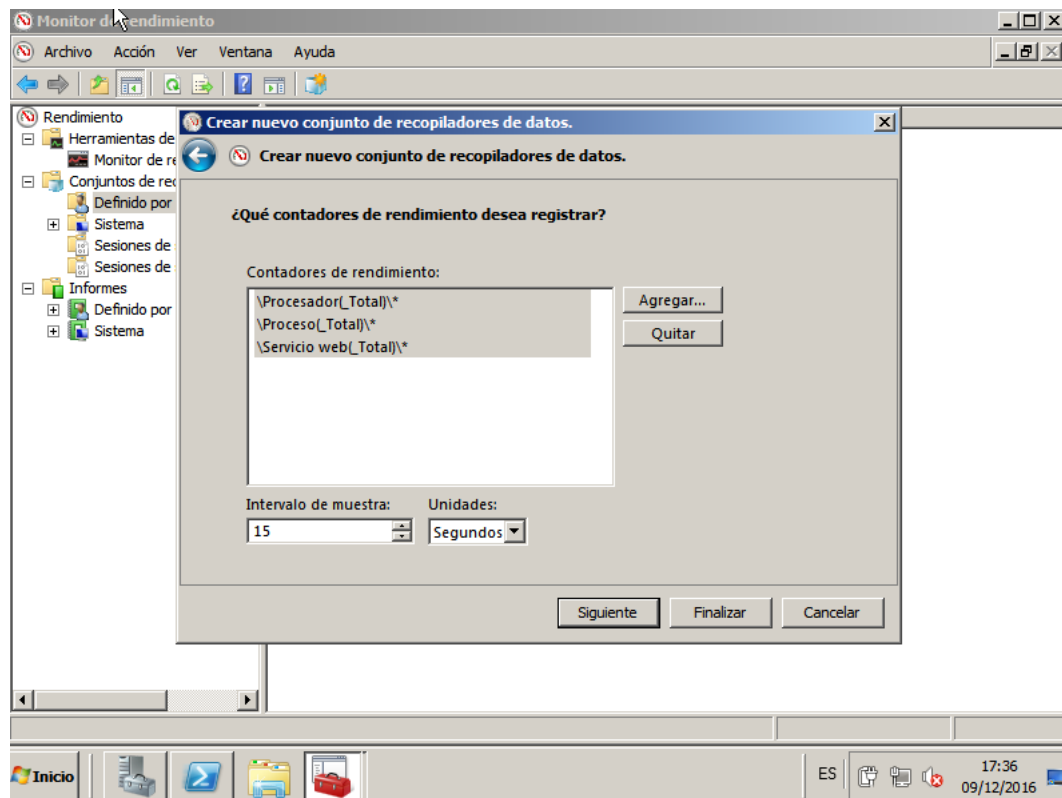


Figura 5.6: Paso 6. Finalizar agregación de contadores de rendimiento

- **Paso 7:** Nos aparece una pantalla para habilitar proveedores de seguimiento. No tenemos que agregar ninguno, por lo que le damos a siguiente.

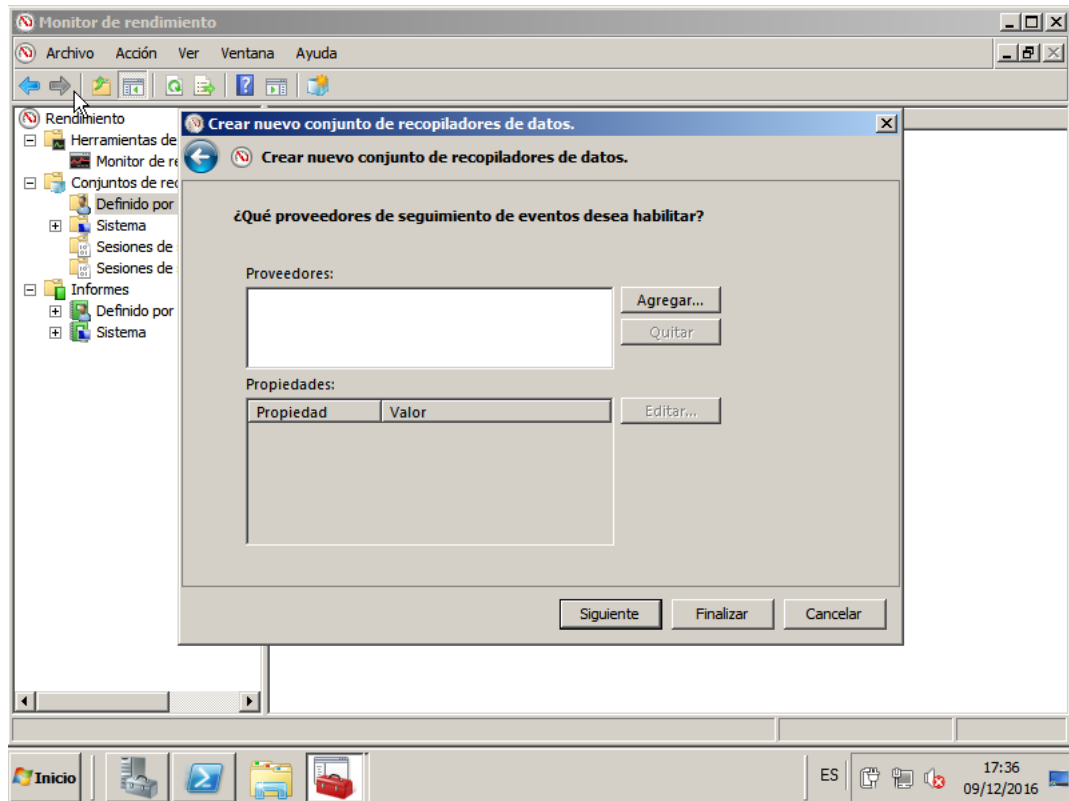


Figura 5.7: Paso 7. Habilitar proveedores de seguimiento

- **Paso 8:** Ahora nos pide una ruta donde guardar los datos del recopilador. Para ello vamos a crear una carpeta en Escritorio que se llame "logs"tal y como nos pide en el guión. Tras crearla le indicamos que sea esa carpeta el directorio raíz.

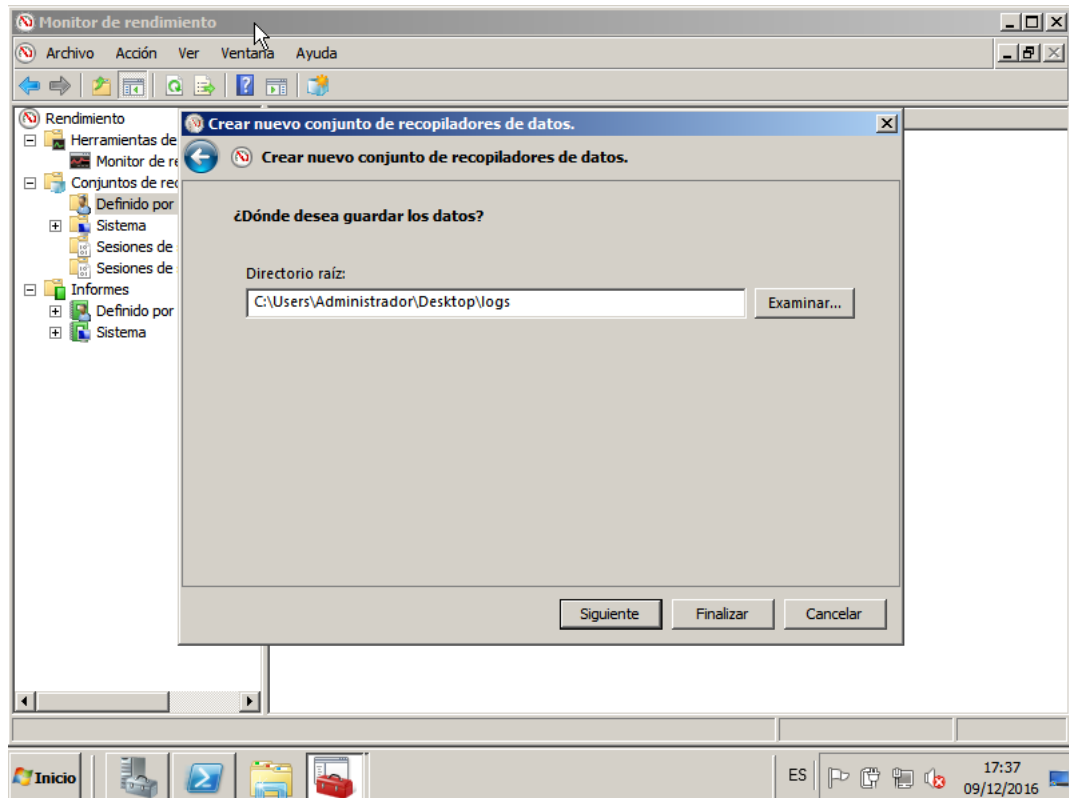


Figura 5.8: Paso 8. Ruta para guardar los datos de recopilador

- **Paso 9:** Finalizar la creación del conjunto de recopiladores de datos. Pulsamos finalizar y se creará.

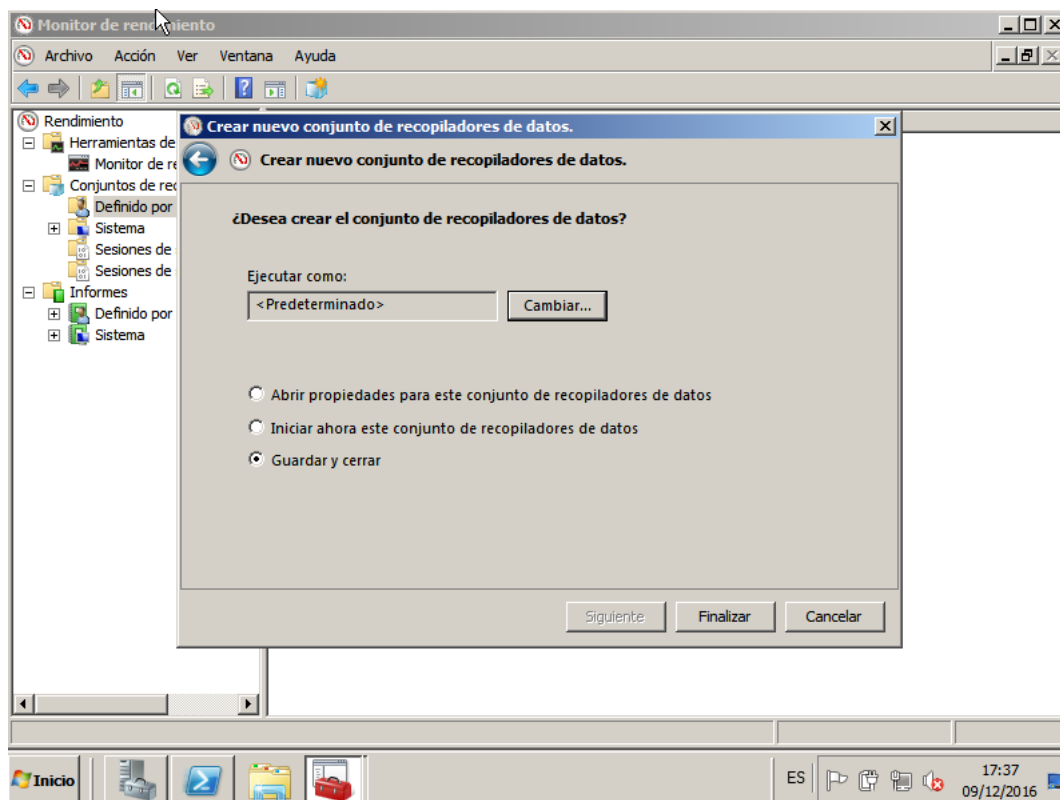


Figura 5.9: Paso 9. Finalizar creación del recopilador de datos

Comprobamos que el recopilador ha sido creado. Se encontrará dentro de Conjuntos de recopiladores de datos > Definido por el usuario.



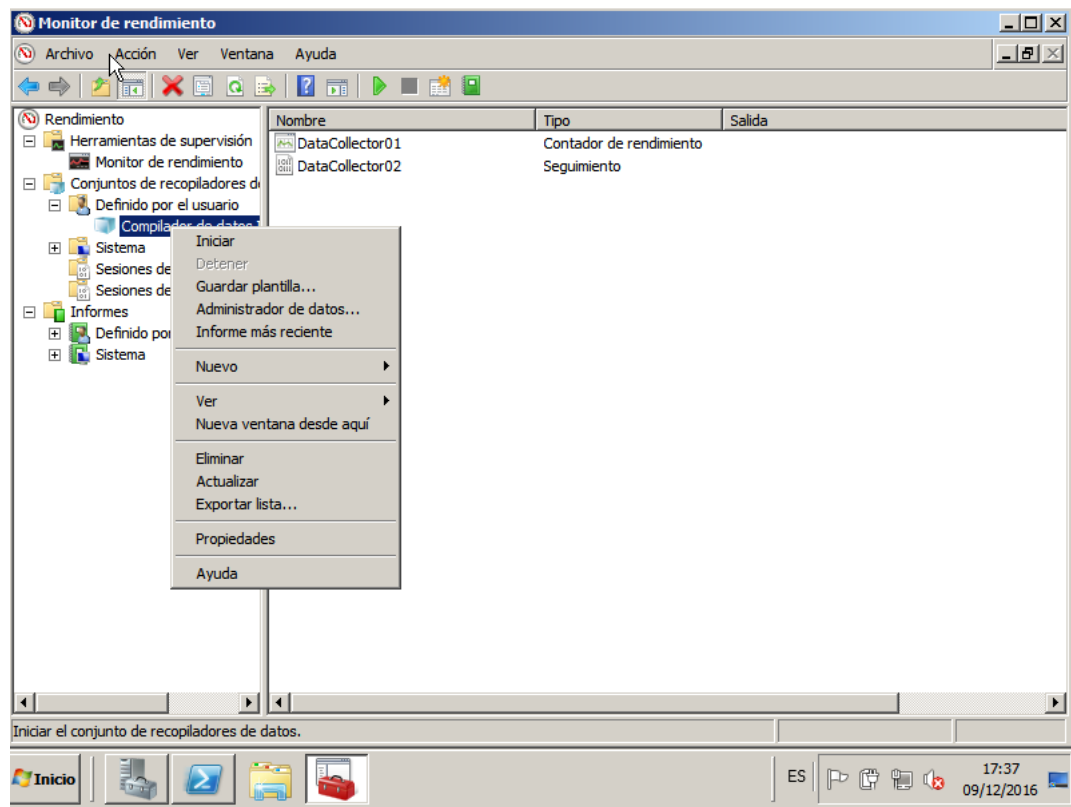


Figura 5.10: Comprobar la creación del recopilador de datos

Hacemos click derecho sobre él y nos permitirá la opción de iniciar. De este modo empezar a recopilar datos. Esperaremos unos minutos para que lo haga y lo pararemos dándole a "Detener".

Vamos a ver que ha generado. Para ello nos vamos a Informes > Definido por el usuario > Copilador de datos ISE. Nos aparecerá un informe que al pulsa sobre él nos mostrará la siguiente gráfica:

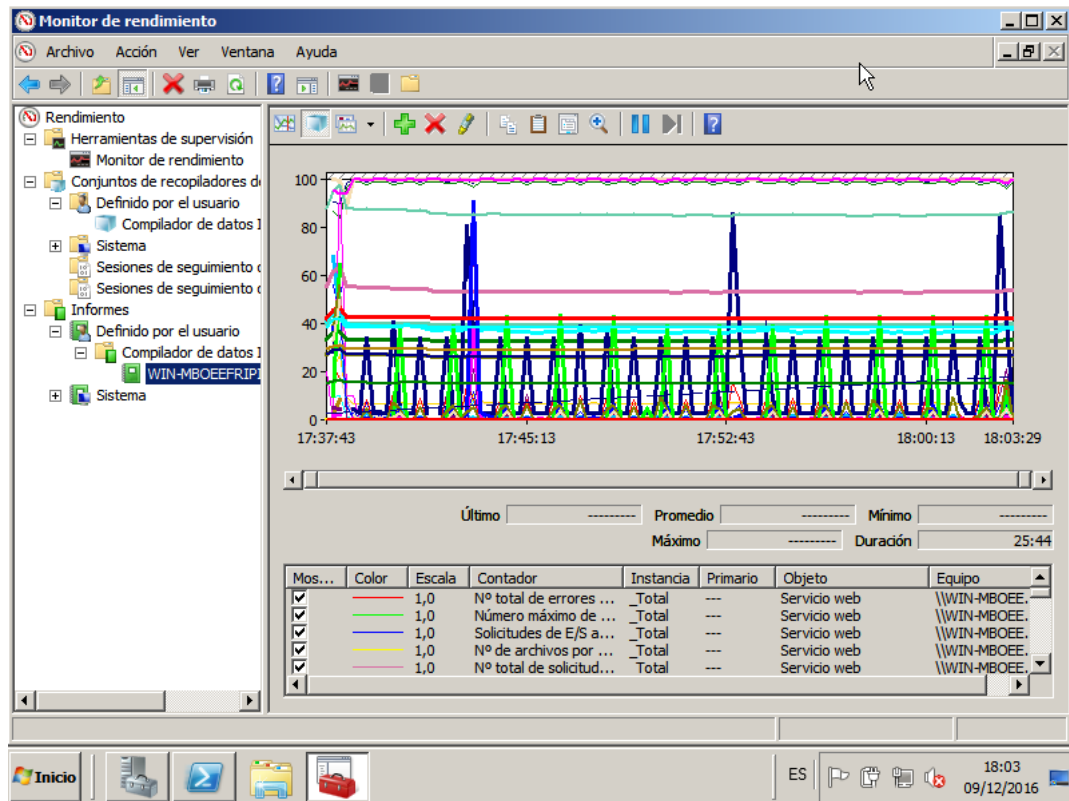


Figura 5.11: Gráfico generado por el recopilador de datos

Por ejemplo, vamos a comprobar los datos recopilados del servicio web. Vemos como tenemos abajo del gráfico una lista con todo lo que nos muestra, de modo que dejamos únicamente los objetos de servicio web. Vemos como no ha habido actividad en nada del servicio web.

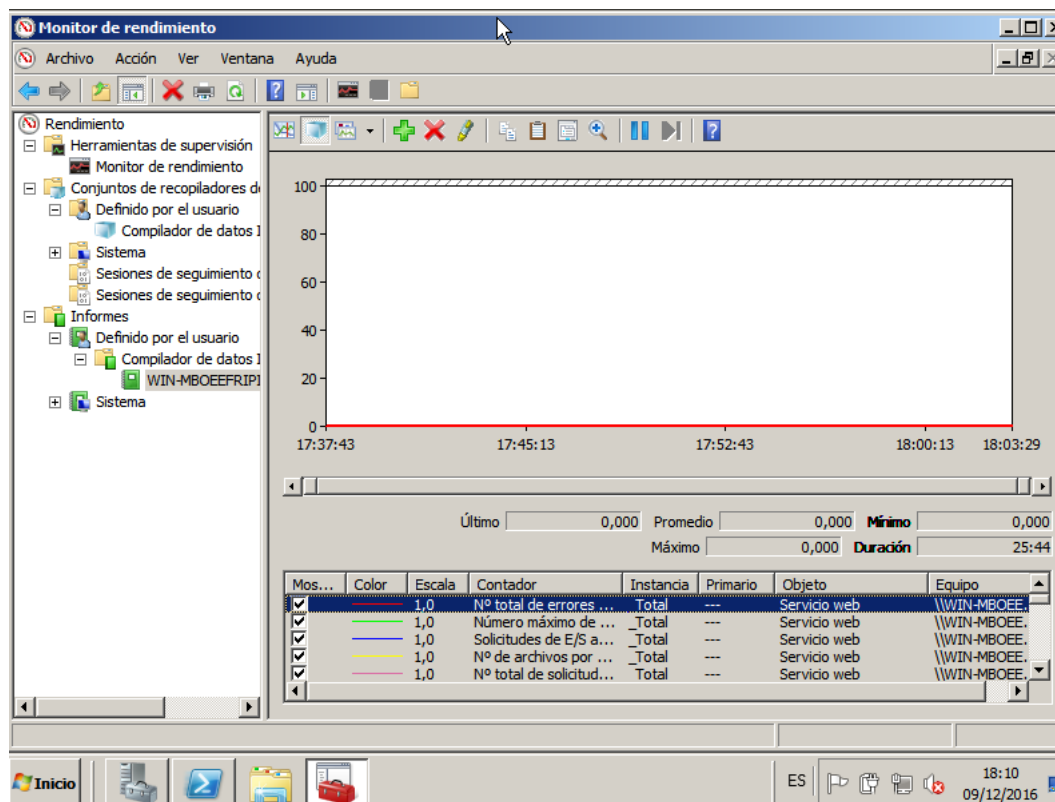


Figura 5.12: Gráfico generado solo para servicio web

Vamos a ver en los procesos las operaciones de E/S para datos, escritura y lectura. En color azul datos, en color carne lectura y en violeta escritura. Vemos como las E/S de datos han sido mayores que las de lectura (que son las segundas) y de escritura (las últimas). A las 17:43 la E/S de datos tiene un máximo de 90,920 ya que abrí el navegador Internet Explorer.<sup>en</sup> ese momento. Ese ha sido su único periodo de más actividad, ya que luego presenta picos de procesador mas inferiores entre 5 y 6 regularmente. Por otra parte, la E/S de lectura tiene picos muy pequeños durante todo el periodo de recopilación de datos, teniendo valores de procesador entorno a 3 y 4. Finalmente la E/S de escritura es la más inferior, ya que en el periodo de medición la única escritura realizada era en los archivos del propio recopilador. Igual para la E/S de lectura, y para la de datos, por eso sus gráficas son similares excepto en un momento puntual (máximo de E/S de datos).

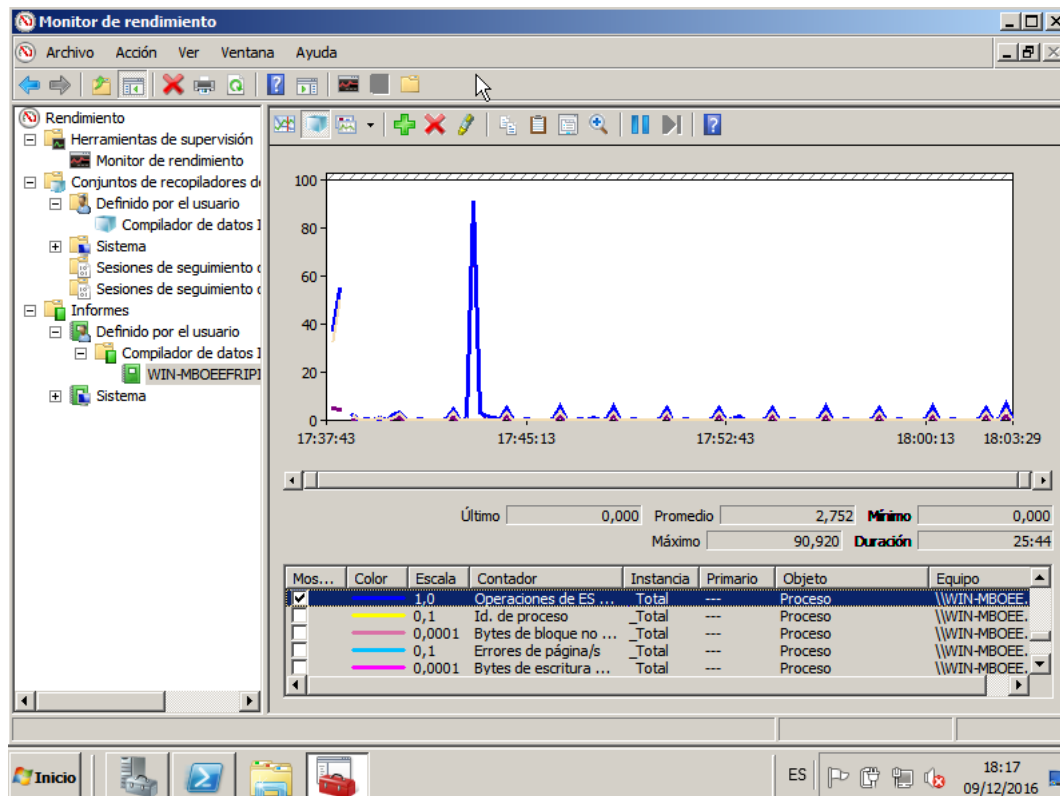


Figura 5.13: Gráfico generado solo para E/S de procesos

- Visite la web del proyecto y acceda a la demo que proporcionan (<http://demo.munin-monitoring.org/>) donde se muestra cómo monitorizan un servidor. Monitoree varios parámetros y haga capturas de pantalla de lo que está mostrando comentando qué observa.

Munin[14] permite monitorizar parámetros relacionados con disco, procesos o el sistema, recopilar sus datos durante diferentes periodos (días, semanas, meses o años) y realizar un seguimiento gráfico de dichos.

Por ejemplo vamos a ver un gráfico sobre el estado de los procesos de un servidor en un día.

Vemos como en el eje X se representan las horas del día y en el eje Y el número de procesos. Esta gráfica nos muestra que casi la totalidad de procesos del servidor están durmiendo (actualmente 71) teniendo una media de 72.22 procesos durmiendo durante

un día concreto. Podemos ver que tiene solo 1 proceso ejecutándose actualmente y que la media de procesos ejecutándose es solo 1. Los demás estados de los procesos no han ocurrido, por lo que no se ven gráficamente.

La conclusión de esta gráfica es que la CPU del servidor no está consumiendo muchos recursos ya que solo tiene un proceso activo.

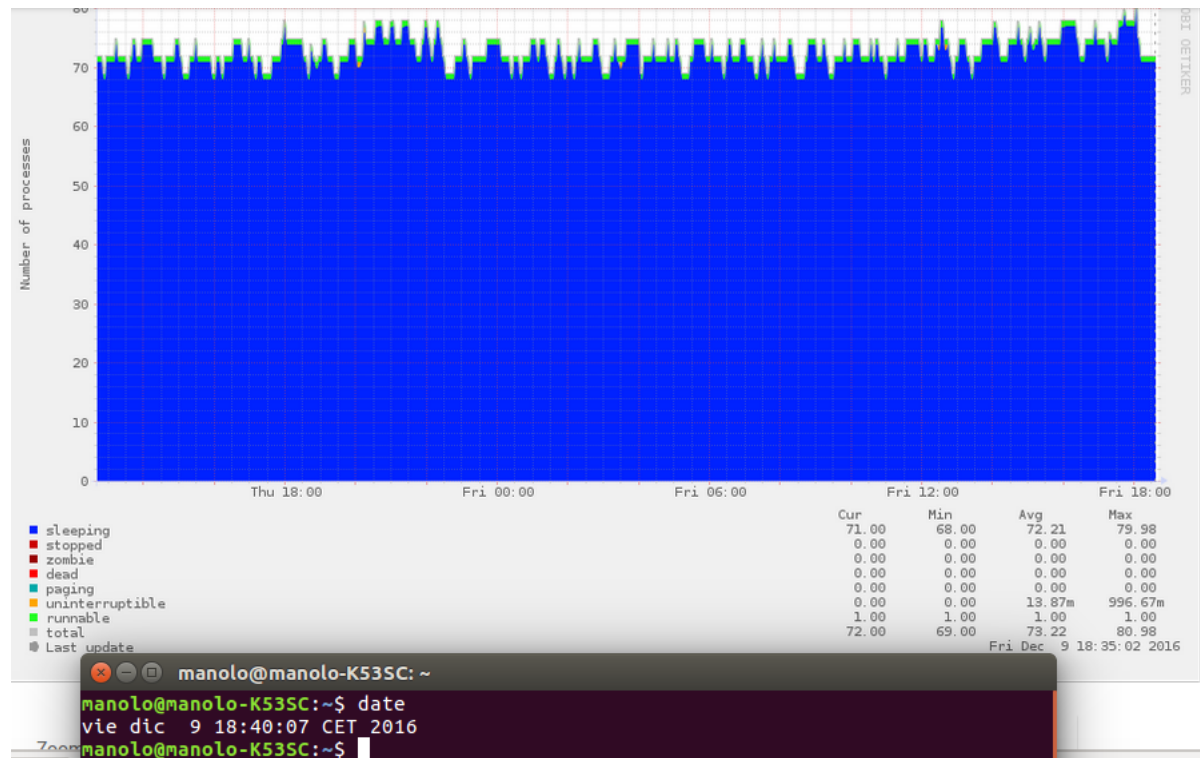


Figura 6.1: Gráfico de estado de los procesos durante un día

Vamos a ver ahora una gráfica del sistema referente al uso de disco en un periodo de un mes. Vemos que el disco ha sido utilizado solo por el root con una media de uso del 31.57%.



Figura 6.2: Uso de disco del servidor durante un mes

Observemos otra gráfica sobre el uso de memoria durante un mes. Vemos como quienes más usan la memoria son committed (memoria asignada a programas), inactive (memoria no usada), active (memoria usada recientemente) y apps (memoria usada por aplicaciones en espacio de usuario). Quienes menos la usan son shemem (memoria compartida), page tables (memoria para mapear entre direcciones físicas y lógicas) y unused (memoria perdida, que no es usada para nada).

Como conclusión, el servidor tiene mucha memoria dedica a los programas y aplicaciones de usuario. Tiene una cantidad importante de memoria no usada, por lo que la memoria no está completamente llena y otra parte de memoria que es usada con frecuencia, lo que hace destacar que el servidor tiene actividad.

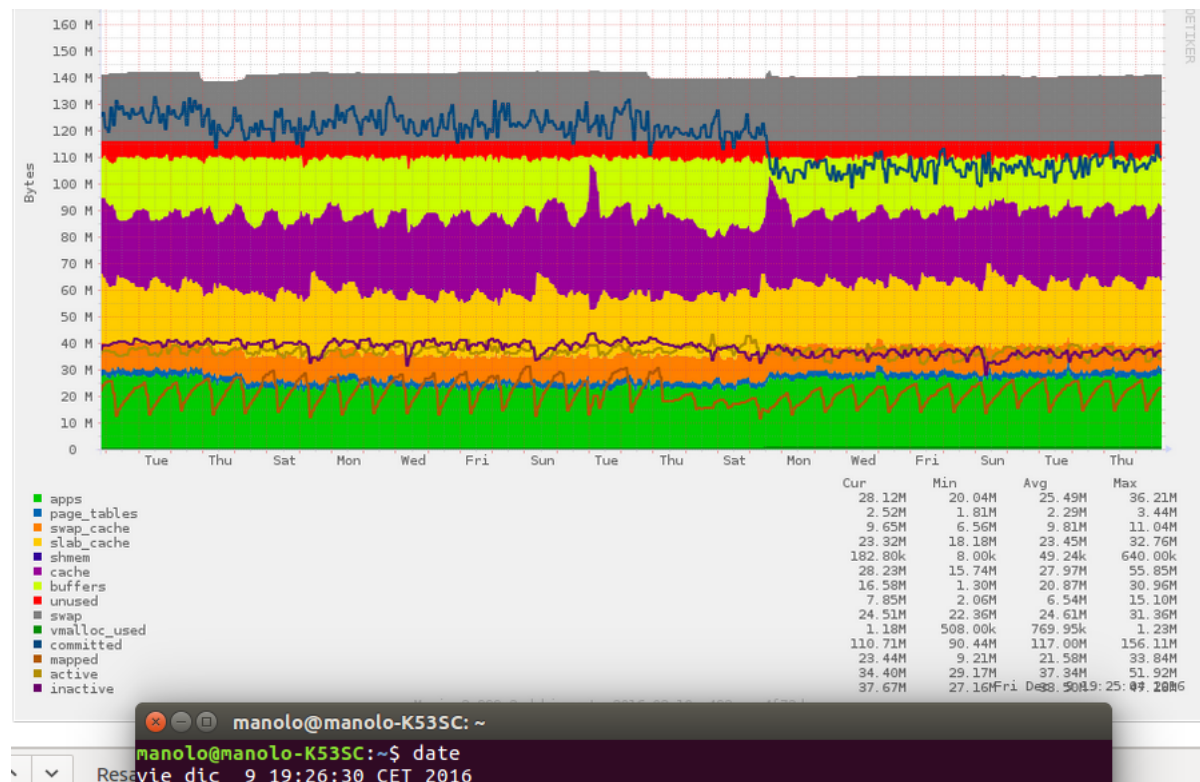


Figura 6.3: Uso de memoria del servidor durante un mes

Field	Internal name	Type	Warn	Crit	Info
buffers	buffers	gauge?			Block device (e.g. harddisk) cache. Also where "dirty" blocks are stored until written.
committed	committed	gauge?			The amount of memory allocated to programs. Overcommitting is normal, but may indicate memory leaks.
unused	free	gauge?			Wasted memory. Memory that is not used for anything at all.
shmem	shmem	gauge?			Shared Memory (SYSV SHM segments, tmpfs).
page_tables	page_tables	gauge?			Memory used to map between virtual and physical memory addresses.
active	active	gauge?			Memory recently used. Not reclaimed unless absolutely necessary.
vmalloc_used	vmalloc_used	gauge?			'VMalloc' (kernel) memory used
apps	apps	gauge?			Memory used by user-space applications.
inactive	inactive	gauge?			Memory not currently used.
slab_cache	slab	gauge?			Memory used by the kernel (major users are caches like inode, dentry, etc).
swap_cache	swap_cache	gauge?			A piece of memory that keeps track of pages that have been fetched from swap but not yet been modified.
mapped	mapped	gauge?			All mmap()ed pages.
swap	swap	gauge?			Swap space used.
cache	cached	gauge?			Parked file data (file content) cache.

Figura 6.4: Información de la tabla de memoria

## 7. Escriba un breve resumen sobre alguno de los artículos donde se muestra el uso de strace o busque otro y coméntelo.

Strace nos sirve para hacer un seguimiento de las llamadas al sistema realizadas por algún procedimiento (seguir un flujo de llamadas), permitiendo así comprobar si hay errores durante dichas llamadas, permitiendo de este modo resolver problemas relacionados con dicho procedimiento.

Este comando podría usarse para un comando o un archivo ejecutable. Para ver como realiza el seguimiento voy a hacer un ejemplo con ls[17]:

```
$ strace ls
```



```

brk(NULL) = 0x21ee000
brk(0x220f000) = 0x220f000
open("/proc/filesystems", O_RDONLY) = 3
fstat(3, {st_mode=S_IFREG|0444, st_size=0, ...}) = 0
read(3, "nodev\tsysfs\nnodev\trootfs\nnodev\ttr", 1024) = 344
read(3, "", 1024) = 0
close(3) = 0
open("/usr/lib/locale/locale-archive", O_RDONLY|O_CLOEXEC) = 3
fstat(3, {st_mode=S_IFREG|0644, st_size=5557808, ...}) = 0
mmap(NULL, 5557808, PROT_READ, MAP_PRIVATE, 3, 0) = 0x7faab7cf1000
close(3) = 0
ioctl(1, TCGETS, {B38400 opost isig icanon echo ...}) = 0
ioctl(1, TIOCGWINSZ, {ws_row=43, ws_col=143, ws_xpixel=0, ws_ypixel=0}) = 0
open(".", O_RDONLY|O_NONBLOCK|O_DIRECTORY|O_CLOEXEC) = 3
fstat(3, {st_mode=S_IFDIR|0755, st_size=4096, ...}) = 0
getdents(3, /* 69 entries */, 32768) = 2352
getdents(3, /* 0 entries */, 32768) = 0
close(3) = 0
open("/usr/lib/x86_64-linux-gnu/gconv/gconv-modules.cache", O_RDONLY) = 3
fstat(3, {st_mode=S_IFREG|0644, st_size=26258, ...}) = 0
mmap(NULL, 26258, PROT_READ, MAP_SHARED, 3, 0) = 0x7faab90d6000
close(3) = 0
futext(0x7faab8c938c8, FUTEX_WAKE_PRIVATE, 2147483647) = 0
fstat(1, {st_mode=S_IFCHR|0620, st_rdev=makedev(136, 4), ...}) = 0
write(1, "Android\t\t Descargas\t Drop", 88) = 88
ntillas
) = 88
write(1, "AndroidStudioProjects Documento", 95) = 95
tch
) = 95
write(1, "archivoprueba\t documento.t", 99) = 99
bando-patch-mejora Videos
) = 99
write(1, "DBA_practica2\t documento.t", 108) = 108
bando-patch.rej VirtualBox VMS
) = 108
close(1) = 0
close(2) = 0
exit_group(0) = ?
+++ exited with 0 +++
manolo@manolo-K53SC:~$ date
vie dic 9 19:50:57 CET 2016

```

Figura 7.1: Uso de strace para ls

Vemos como ls realiza llamadas al sistema como `open()`, `close()`, `fstat()` para tener información del estado de los archivos, `mmap()` para asignar o quitar archivos o dispositivos a la memoria con el fin de obtener como resultado lo que se espera de él, obtener la lista de archivos que hay en un directorio.

## 8. Escriba un script en Python o PHP y analice su comportamiento usando el profiler presentado.

Voy a realizar un script en Python[16] que realiza un bucle con 90000 iteraciones. El contenido del script es el siguiente:

```

GNU nano 2.2.6          Archivo: profiling.py

import cProfile
class monitorizar:
    def bucle(self):
        contador=1
        print("Inicio variable contador:"+str(contador))
        for i in range(90000):
            contador=contador+1
        print("Final variable contador:"+str(contador))

ejecuta=monitorizar()
cProfile.run('ejecuta.bucle()')

```

Figura 8.1: Script en Python para profiler

Ejecutamos el profiler para ver que información nos muestra:

```
manolo@ubuntu ~ 2016-12-09 21:01:13
$python profiling.py
Inicio variable contador:1
Final variable contador:90001
4 function calls in 0.013 seconds

Ordered by: standard name

ncalls  tottime  percall  cuntime  percall  filename:lineno(function)
1      0.000    0.000    0.013    0.013  <string>:1(<module>)
1      0.010    0.010    0.013    0.013  profiling.py:4(bucle)
1      0.000    0.000    0.000    0.000  {method 'disable' of '_lsprof.Profiler' objects}
1      0.003    0.003    0.003    0.003  {range}
```

Figura 8.2: Ejecutando profiler en Python

Vemos como ejecuta el script y muestra la información de salida que tenía y además el profiler nos da datos sobre las llamadas que se realizan o por ejemplo el tiempo que tardó por llamada o el tiempo total.

## 9. Acceda a la consola mysql (o a través de phpMyAdmin) y muestre el resultado de mostrar el "profile" de una consulta (la creación de la BD y la consulta la puede hacer libremente).

Vamos a crear unas tablas y a realizar un par de operaciones en mysql para despues mostrar el "profile"[15].

- sudo mysql -p . Para acceder a mysql.

```
manolo@ubuntu ~ 2016-12-09 20:28:44
$sudo mysql -p
[sudo] password for manolo:
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 49
Server version: 5.5.53-0ubuntu0.14.04.1 (Ubuntu)

Copyright (c) 2000, 2016, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>
```

Figura 9.1: Iniciar mysql

- mysql>SET profiling = 1; Para poder mostrar profiles.
- mysql>CREATE DATABASE ise;

```
mysql> CREATE DATABASE ise;
Query OK, 1 row affected (0.00 sec)
```

Figura 9.2: Crear base de datos en mysql

- `mysql>use ise;`

```
mysql> use ise;
Database changed
```

Figura 9.3: Usar una base de datos existente en mysql

- `mysql>show tables;` nos dice que no tenemos

```
mysql> show tables;
Empty set (0.00 sec)
```

Figura 9.4: Ver tablas de una base de datos en mysql

- `mysql>create table mascota(nombre VARCHAR(20), propietario VARCHAR(20), especie VARCHAR(20));`

```
mysql> create table mascota(nombre VARCHAR(20), propietario VARCHAR(20), especie VARCHAR(20));
Query OK, 0 rows affected (0.50 sec)
```

Figura 9.5: Crear tabla en una base de datos en mysql

- `mysql>insert into mascota values('lucera','manolo','perro');`

```
mysql> insert into mascota values('lucera','manolo','perro');
Query OK, 1 row affected (0.30 sec)
```

Figura 9.6: Insertar una fila en una tabla en mysql

- `mysql>insert into mascota values('luna','pepe','gato');`

```
mysql> insert into mascota values('luna','pepe','gato');
Query OK, 1 row affected (0.35 sec)
```

Figura 9.7: Insertar otra fila en una tabla en mysql

- `mysql>select * from mascota;`

```
mysql> select * from mascota;
+-----+-----+-----+
| nombre | propietario | especie |
+-----+-----+-----+
| lucera | manolo      | perro   |
| luna   | pepe       | gato    |
+-----+-----+-----+
2 rows in set (0.00 sec)
```

Figura 9.8: Ver tabla mascota en mysql

- `mysql>show profiles;` (con esto vemos el tiempo de cada operación)

```

+----+-----+-----+
| Query_ID | Duration | Query |
+----+-----+-----+
| 1 | 0.33051150 | insert into mascota values('pico', 'manolo', 'perro') |
| 2 | 0.0007025 | drop ise |
| 3 | 0.00011875 | drop ise |
| 4 | 0.25871375 | drop database ise |
| 5 | 0.00030625 | SELECT DATABASE() |
| 6 | 0.00065700 | CREATE DATABASE ise |
| 7 | 0.00018925 | SELECT DATABASE() |
| 8 | 0.00071275 | show databases |
| 9 | 0.20105075 | show tables |
| 10 | 0.00038500 | show tables |
| 11 | 0.52265450 | create table mascota(nombre VARCHAR(20), propietario VARCHAR(20), especie |
| 12 | 0.40040050 | insert into mascota values('lucera', 'manolo', 'perro') |
| 13 | 0.34364975 | insert into mascota values('luna', 'pepe', 'gato') |
| 14 | 0.00042350 | select * from mascota |
+----+-----+-----+
14 rows in set (0.00 sec)

mysql> _

```

Figura 9.9: Ver profiles en mysql

Vemos como el profile nos muestra el tiempo que ha tardado en realizarse cada operación que hemos realizado en mysql. Vemos que las operaciones más costosas han sido las de creación de la tabla y la de inserción de valores en ella, ya que estamos realizando operaciones de escritura. Vemos como las operaciones de solo lectura como mostrar lo que hay en las tablas son más rápidas.

10. **Opcional 1: Indique qué comandos ha utilizado para realizarlo así como capturas de pantalla del proceso de reconstrucción del RAID.**
11. **Opcional 2: Instale Nagios en su sistema (el que prefiera) documentando el proceso y muestre el resultado de la monitorización de su sistema comentando qué aparece.**
12. **Opcional 3: Pruebe a instalar este monitor en alguno de sus tres sistemas. Realice capturas de pantalla del proceso de instalación y comente capturas de pantalla del programa en ejecución.**
13. **Opcional 4: Pruebe a instalar este monitor en alguno de sus tres sistemas. Realice capturas de pantalla del proceso de instalación y comente capturas de pantalla del programa en ejecución.**

Vamos a instalar Cacti[13] paso por paso:

- Instalamos Cacti: `sudo apt-get install cacti` . Dentro del proceso de instalación nos instalará:
  - Nos dará a elegir entre varios servidores, elegiremos Apache2.
  - Instalará php.
  - Instalará MySQL.
  - Otros servicios necesarios para Cacti, como rddtool y snmp.
  - Configuración de Cacti.
    - Configurar la base de datos para Cacti.

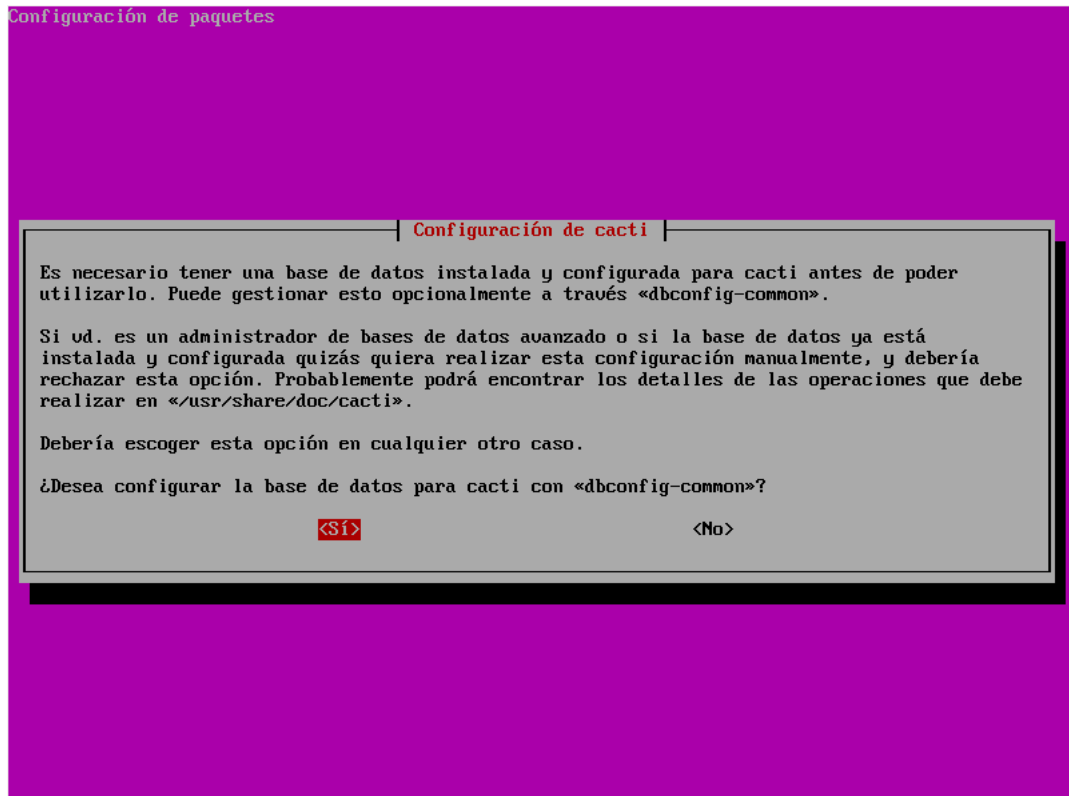


Figura 13.1: Configurar la base de datos para cacti

- o Crear contraseña para usuario de administración de la base de datos.

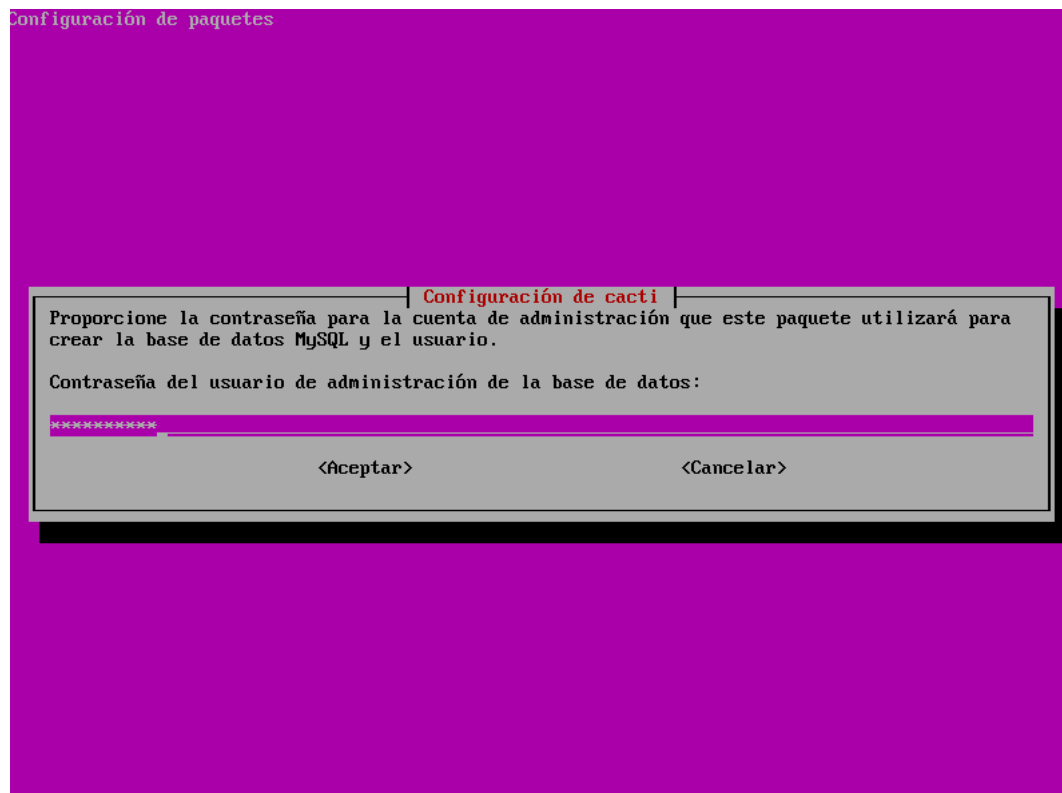


Figura 13.2: Crear contraseña de administración para BD

- Crear contraseña de MySQL para Cacti.

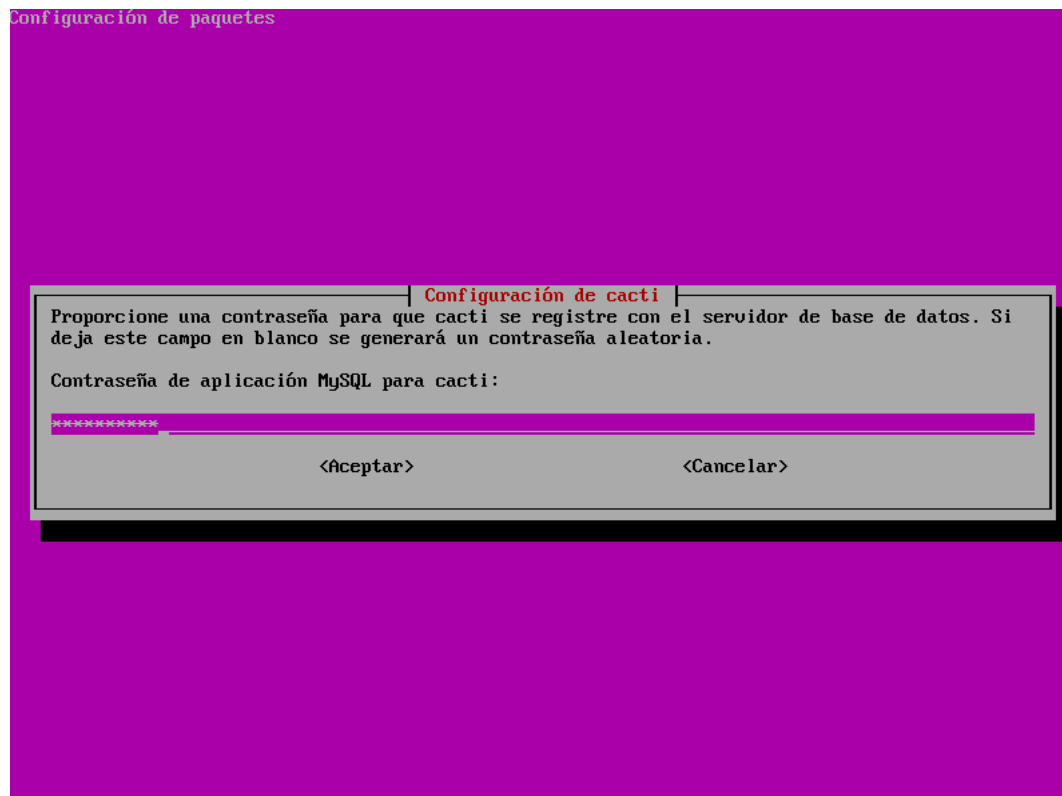


Figura 13.3: contraseña de MySQL para Cacti

- Confirmación de contraseña de MySQL.



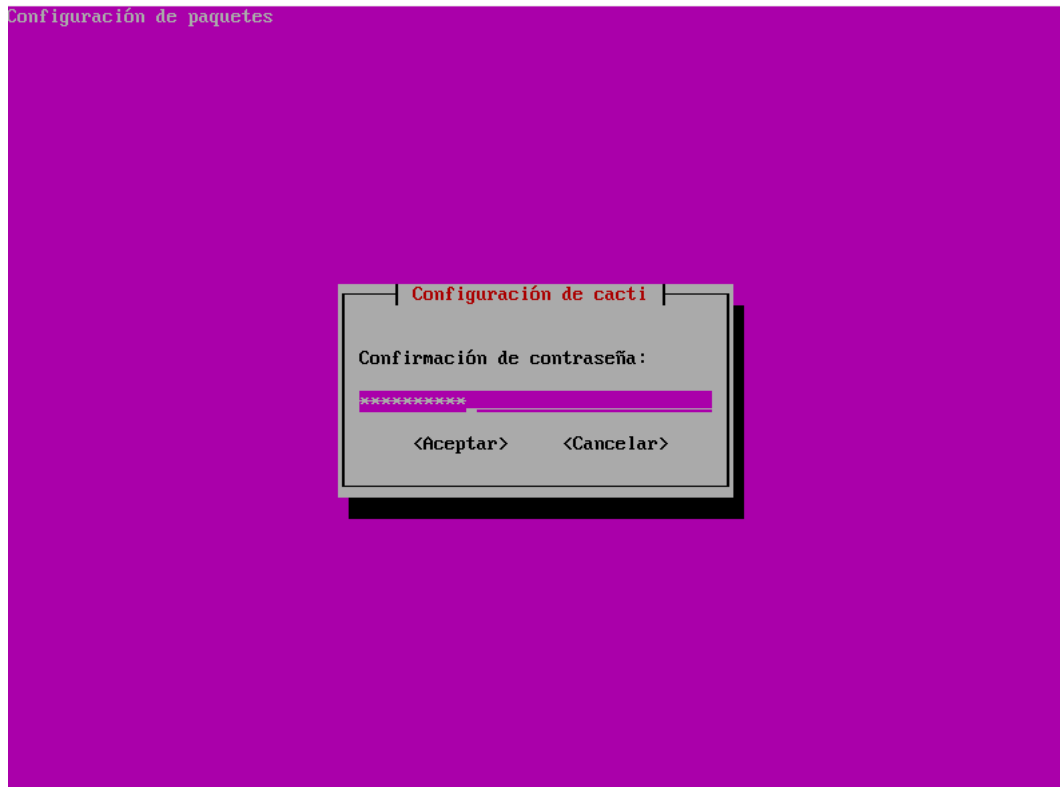


Figura 13.4: Confirmación de contraseña de MySQL

- Comprobamos que Cacti ha sido instalado. Para ello, desde la máquina anfitriona nos conectamos a la ip de nuestro servidor escribiendo en el navegador: 192.168.56.102/cacti. Nos pedirá usuario y contraseña (será ".admin", ".admin" por defecto). Nos pedirá actualizar la contraseña solo con iniciar sesión. Después de cambiar la contraseña, nos aparecerá una guía de instalación. Aceptamos para pasar a otro paso de instalación 'next' y acabamos 'Finish' para finalizarla:

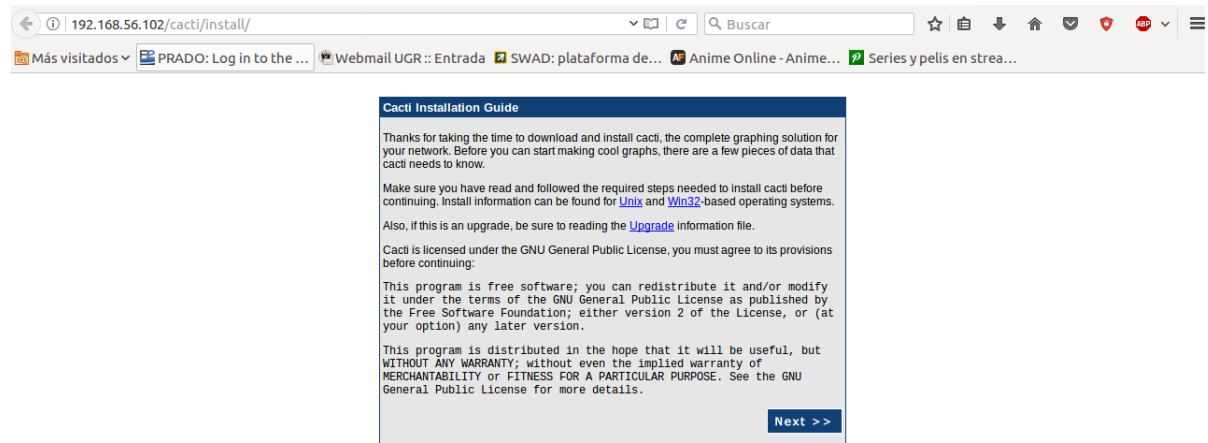


Figura 13.5: Paso 1 guía instalación Cacti

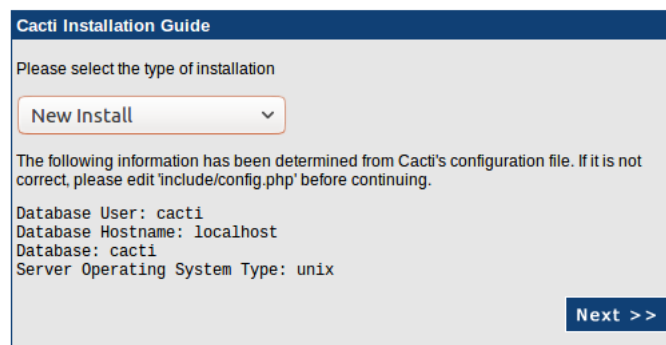


Figura 13.6: Paso 2 guía instalación Cacti

**Cacti Installation Guide**

Make sure all of these values are correct before continuing.

**[FOUND] RRDTool Binary Path:** The path to the rrdtool binary.  
  
[OK: FILE FOUND]

**[FOUND] PHP Binary Path:** The path to your PHP binary file (may require a php recompile to get this file).  
  
[OK: FILE FOUND]

**[FOUND] snmpwalk Binary Path:** The path to your snmpwalk binary.  
  
[OK: FILE FOUND]

**[FOUND] snmpget Binary Path:** The path to your snmpget binary.  
  
[OK: FILE FOUND]

**[FOUND] snmpbulkwalk Binary Path:** The path to your snmpbulkwalk binary.  
  
[OK: FILE FOUND]

**[FOUND] snmpgetnext Binary Path:** The path to your snmpgetnext binary.  
  
[OK: FILE FOUND]

**[FOUND] Cacti Log File Path:** The path to your Cacti log file.  
  
[OK: FILE FOUND]

**SNMP Utility Version:** The type of SNMP you have installed. Required if you are using SNMP v2c or don't have embedded SNMP support in PHP.

**RRDTool Utility Version:** The version of RRDTool that you have installed.

**NOTE:** Once you click "Finish", all of your settings will be saved and your database will be upgraded if this is an upgrade. You can change any of the settings on this screen at a later time by going to "Cacti Settings" from within Cacti.

**Finish**

Figura 13.7: Paso 3 guía instalación Cacti

- Vemos la interfaz de Cacti, donde nos muestra las diferentes opciones a realizar. Se pueden crear nuevos gráficos (de varios parámetros del sistema)

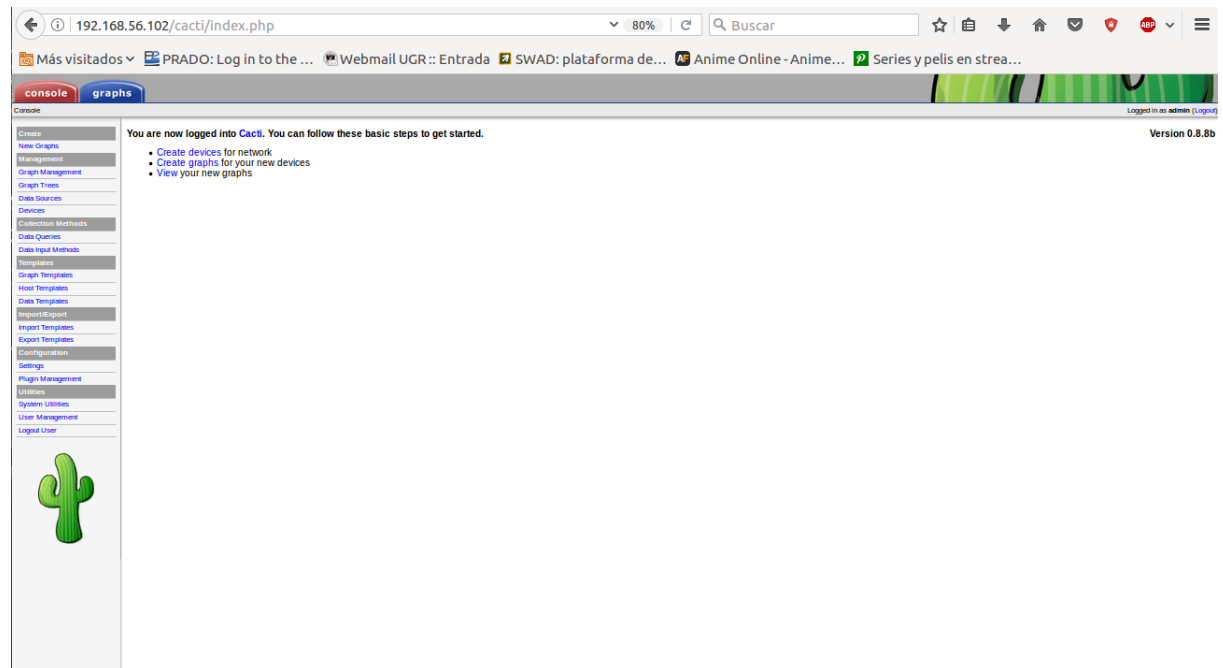


Figura 13.8: Interfaz de Cacti

- Pulsamos sobre las gráficas, mostrándose varias como vemos aquí:
  - Carga media del localhost. No hemos generado mucha carga, ya que no hemos realizado ninguna acción que lo requiera.

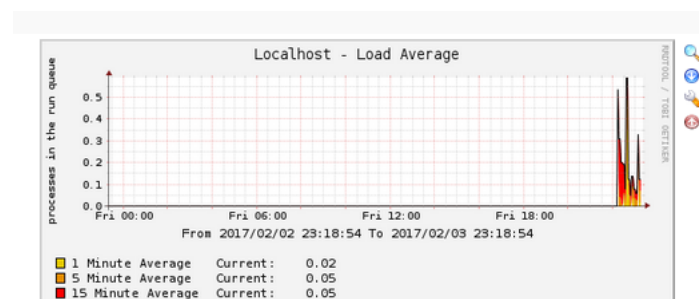


Figura 13.9: Gráfica de carga media de Cacti

- Procesos del localhost. La cantidad de procesos ejecutándose en nuestro Ubuntu Server.

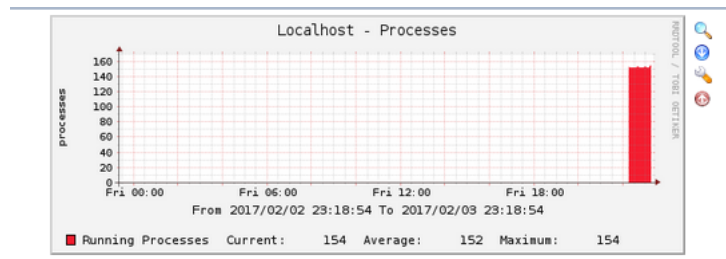


Figura 13.10: gráfica de procesos de Cacti

- Usuarios conectados en localhost. Como solo estoy yo conectado, por eso es solo uno.

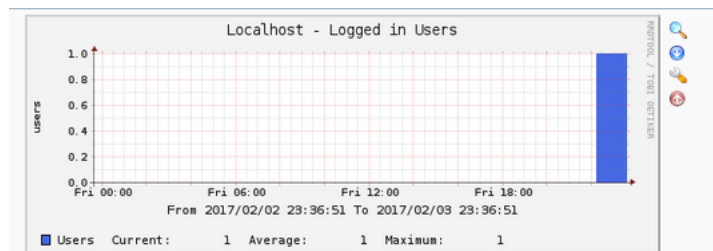


Figura 13.11: Gráfica de usuarios conectados de Cacti

## 14. Opcional 5: Instale el monitor. Muestre y comente algunas capturas de pantalla.

### Referencias

- [1] [https://books.google.es/books?id=a\\_OoCwAAQBAJ&printsec=frontcover&hl=es#v=onepage&q&f=false](https://books.google.es/books?id=a_OoCwAAQBAJ&printsec=frontcover&hl=es#v=onepage&q&f=false), consultado el 1 de Noviembre de 2016.
- [2] [https://books.google.es/books?id=I5HvCgAAQBAJ&pg=PT474&lpg=PT474&dq=apt+history.log&source=bl&ots=FVUhUwLA1a&sig=f1AbUjyeyTvfrWmp7zvBj1aTnpE&hl=es&sa=X&ved=0ahUKEwj\\_m6fmt9PQAhXMWBQKHevACeo4FBDoAQhSMAC#v=onepage&q=apt%20history.log&f=false](https://books.google.es/books?id=I5HvCgAAQBAJ&pg=PT474&lpg=PT474&dq=apt+history.log&source=bl&ots=FVUhUwLA1a&sig=f1AbUjyeyTvfrWmp7zvBj1aTnpE&hl=es&sa=X&ved=0ahUKEwj_m6fmt9PQAhXMWBQKHevACeo4FBDoAQhSMAC#v=onepage&q=apt%20history.log&f=false), consultado el 1 de Noviembre de 2016.
- [3] <https://linux.die.net/man/1/crontab>, consultado el 1 de Noviembre de 2016.
- [4] <https://linux.die.net/man/1/crontab>, consultado el 1 de Noviembre de 2016.
- [5] <https://linux.die.net/man/5/proc>, consultado el 1 de Noviembre de 2016.
- [6] <https://linux.die.net/man/8/dmesg>, consultado el 1 de Noviembre de 2016.

- [7] <https://linux.die.net/man/8/logrotate>, consultado el 1 de Noviembre de 2016.
- [8] [https://www.debian.org/doc/manuals/debian-reference/ch02.en.html#\\_package\\_activity\\_logs](https://www.debian.org/doc/manuals/debian-reference/ch02.en.html#_package_activity_logs), consultado el 1 de Noviembre de 2016.
- [9] <http://tldp.org/LDP/Linux-Filesystem-Hierarchy/html/proc.html>, consultado el 1 de Noviembre de 2016.
- [10] <http://www.tldp.org/LDP/Linux-Filesystem-Hierarchy/html/var.html>, consultado el 1 de Noviembre de 2016.
- [11] <http://www.tldp.org/LDP/sag/html/proc-fs.html>, consultado el 1 de Noviembre de 2016.
- [12] <http://www.tldp.org/pub/Linux/docs/ldp-archived/system-admin-guide/translations/es/html/ch04s06.html>, consultado el 1 de Noviembre de 2016.
- [13] <http://www.cacti.net/>, consultado el 3 de Febrero de 2017.
- [14] <http://demo.munin-monitoring.org/>, consultado el 9 de Noviembre de 2016.
- [15] <http://dev.mysql.com/doc/refman/5.7/en/show-profile.html>, consultado el 9 de Noviembre de 2016.
- [16] <https://docs.python.org/2/library/profile.html>, consultado el 9 de Noviembre de 2016.
- [17] <https://linux.die.net/man/1/strace>, consultado el 9 de Noviembre de 2016.