

A dark blue vertical bar on the left side of the page. A blue arrow points to the right from the bar, containing the date.

1-12-2016

# Memoria Práctica 3

Ingeniería de Servidores

Several thin, curved lines in dark blue and light grey originate from the bottom left and sweep upwards and to the right.

Andrés Molina López  
UNIVERSIDAD DE GRANADA

## **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 de ese directorio?.....	4
2. ¿Qué archivo ha de modificar para programar una tarea? Escriba la línea necesaria para ejecutar una vez al día una copia del directorio ~/código a ~/seguridad/\$fecha donde \$fecha es la fecha actual.....	5
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.....	6
4. Ejecute el monitor de “System Performance” y muestre el resultado. Incluya capturas de pantalla comentando la información que aparece.....	8
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 de 15 segundos, almacene el resultado en el directorio Escritorio/logs. Incluya capturas de pantalla de cada paso.....	12
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. Monitorice varios parámetros y haga capturas de pantalla de lo que está mostrando comentando qué observa.....	17
7. Escriba un breve resumen sobre alguno de los artículos donde se muestra el uso de strace o busque otro y coméntelo.....	21
8. Escriba un script en Python o PHP y analice su comportamiento usando el profiler presentado.....	21
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).....	24

## **Índice de figuras:**

1ª Ilustración.....	4
2ª Ilustración.....	4

3ª Ilustración.....	5
4ª Ilustración.....	5
5ª Ilustración.....	6
6ª Ilustración.....	7
7ª Ilustración.....	7
8ª Ilustración.....	7
9ª Ilustración.....	8
10ª Ilustración.....	9
11ª Ilustración.....	9
12ª Ilustración.....	10
13ª Ilustración.....	10
14ª Ilustración.....	11
15ª Ilustración.....	11
16ª Ilustración.....	12
17ª Ilustración.....	13
18ª Ilustración.....	13
19ª Ilustración.....	14
20ª Ilustración.....	14
21ª Ilustración.....	15
22ª Ilustración.....	15
23ª Ilustración.....	16
24ª Ilustración.....	16
25ª Ilustración.....	17
26ª Ilustración.....	17
27ª Ilustración.....	18
28ª Ilustración.....	18

29ª Ilustración.....	19
30ª Ilustración.....	19
31ª Ilustración.....	20
32ª Ilustración.....	22
33ª Ilustración.....	22
34ª Ilustración.....	25
35ª Ilustración.....	25
36ª Ilustración.....	26
37ª Ilustración.....	27
38ª Ilustración.....	27

## 1ª Cuestión:

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

En Ubuntu ese archivo es *history.log* y lo encontramos en el directorio */var/log/apt*. En la *Ilustración 1* muestro el contenido del directorio y de este archivo. En mi caso se ve que solo hay actualizaciones de paquetes, ya que todo lo anterior se ha guardado en *history.log.1.gz*.

```
AndMolLop 01/12/16:~$ ls /var/log/apt/
history.log  history.log.1.gz  term.log  term.log.1.gz
AndMolLop 01/12/16:~$ cat /var/log/apt/history.log

Start-Date: 2016-12-01 19:13:14
Commandline: apt-get upgrade
Requested-By: andres (1000)
Upgrade: libdns-export162:amd64 (1:9.10.3.dfsg.P4-8ubuntu1.2, 1:9.10.3.dfsg.P4-8ubuntu1.3), libiscf
g140:amd64 (1:9.10.3.dfsg.P4-8ubuntu1.2, 1:9.10.3.dfsg.P4-8ubuntu1.3), bind9-host:amd64 (1:9.10.3.df
sg.P4-8ubuntu1.2, 1:9.10.3.dfsg.P4-8ubuntu1.3), dnstools:amd64 (1:9.10.3.dfsg.P4-8ubuntu1.2, 1:9.10
.3.dfsg.P4-8ubuntu1.3), libisc160:amd64 (1:9.10.3.dfsg.P4-8ubuntu1.2, 1:9.10.3.dfsg.P4-8ubuntu1.3), g
rub-legacy-ec2:amd64 (0.7.8-1-g3705bb5-0ubuntu1~16.04.3, 0.7.8-49-g9e904bb-0ubuntu1~16.04.1), libisc
-export160:amd64 (1:9.10.3.dfsg.P4-8ubuntu1.2, 1:9.10.3.dfsg.P4-8ubuntu1.3), im-config:amd64 (0.29-1
ubuntu12.2, 0.29-1ubuntu12.3), liblures141:amd64 (1:9.10.3.dfsg.P4-8ubuntu1.2, 1:9.10.3.dfsg.P4-8ubu
ntu1.3), libdns162:amd64 (1:9.10.3.dfsg.P4-8ubuntu1.2, 1:9.10.3.dfsg.P4-8ubuntu1.3), libisc140:amd
64 (1:9.10.3.dfsg.P4-8ubuntu1.2, 1:9.10.3.dfsg.P4-8ubuntu1.3), libbind9-140:amd64 (1:9.10.3.dfsg.P4-
8ubuntu1.2, 1:9.10.3.dfsg.P4-8ubuntu1.3)
Error: Sub-process /usr/bin/dpkg returned an error code (1)
End-Date: 2016-12-01 19:14:48
AndMolLop 01/12/16:~$ _
```

Ilustración 1. Contenido del directorio */var/log/apt* y del archivo *history.log* situado en este

Siguiendo de ejemplo la localización de este fichero en Ubuntu, buscamos en CentOS, y vemos que el archivo se encuentra localizado en */var/log* y que se llama *yum.log*. En la *Ilustración 2* muestro la localización del archivo y las últimas 20 líneas de su contenido.

```
[AndMolLop 01/12/16 ~]# ls -l /var/log/ | grep yum
-rw----- 1 root root 68906 nov 25 15:45 yum.log
[AndMolLop 01/12/16 ~]# tail -n 20 /var/log/yum.log
Nov 25 01:20:00 Installed: perl-IO-Compress-2.061-2.el7.noarch
Nov 25 01:20:01 Installed: perl-PIRPC-0.2020-14.el7.noarch
Nov 25 01:20:02 Installed: perl-DBI-1.627-4.el7.x86_64
Nov 25 01:20:04 Installed: perl-DBD-MySQL-4.023-5.el7.x86_64
Nov 25 01:20:14 Installed: 1:mariadb-server-5.5.50-1.el7_2.x86_64
Nov 25 01:31:17 Installed: libzip-0.10.1-8.el7.x86_64
Nov 25 01:31:18 Installed: php-common-5.4.16-36.3.el7_2.x86_64
Nov 25 01:31:19 Installed: php-cli-5.4.16-36.3.el7_2.x86_64
Nov 25 01:31:20 Installed: php-pdo-5.4.16-36.3.el7_2.x86_64
Nov 25 01:31:20 Installed: php-mysql-5.4.16-36.3.el7_2.x86_64
Nov 25 01:31:20 Installed: php-5.4.16-36.3.el7_2.x86_64
Nov 25 14:52:45 Installed: libmpc-1.0.1-3.el7.x86_64
Nov 25 14:52:47 Installed: cpp-4.8.5-4.el7.x86_64
Nov 25 14:52:49 Installed: kernel-headers-3.10.0-327.36.3.el7.x86_64
Nov 25 14:52:50 Installed: glibc-headers-2.17-106.el7_2.8.x86_64
Nov 25 14:52:50 Installed: glibc-devel-2.17-106.el7_2.8.x86_64
Nov 25 14:52:55 Installed: gcc-4.8.5-4.el7.x86_64
Nov 25 15:11:47 Installed: patch-2.7.1-8.el7.x86_64
Nov 25 15:44:41 Installed: perl-Net-SSLeay-1.55-3.el7.x86_64
Nov 25 15:45:11 Installed: webmin-1.820-1.noarch
[AndMolLop 01/12/16 ~]# _
```

Ilustración 2. Localización y contenido del archivo *yum.log*

[1] ("Ubuntu Logs [Adminbuntu]", 2016)

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

Los archivos log cuando alcanzan un determinado tamaño, o cada cierto tiempo se “reinician” para que no ocupen tanto espacio, pero la información que tenían no se pierde, sino que la herramienta logrotate, que es la encargada de “reiniciar” estos archivos, comprime su contenido en un nuevo archivo terminado en un número y luego vacía el archivo log. El número que se le pone al comprimido (.gz) depende de si hay otros hechos antes o no, de manera que cuanto más antiguo sea, mayor será el número que tendrá, ya que al crear uno nuevo a ese se le asigna el 1 por ser el más reciente, y a los anteriores se le cambia el nombre, simplemente sumándole uno al número que tuviesen.

[2] ("Rotación de ficheros logs. \*", 2016)

**2º Cuestión: ¿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.**

Lo primero es crear los directorios codigo y seguridad en el home del usuario en cuestión, además, al directorio código le he añadido un archivo de texto para que al hacerse la copia se vea que se copia algo. Muestro los directorios el contenido de codigo en la *Ilustración 3*.

```
AndMolLop 03/12/16:~$ ls
allowrootssh.sh  codigo  Descargas  Escritorio  Imágenes  numeros.txt  Plantillas  seguridad
changeport.py   copia.sh Documentos  fabfile.pyc Música      PDFs        Público     Vídeos
AndMolLop 03/12/16:~$ ls codigo/
codigos.txt
AndMolLop 03/12/16:~$ cat codigo/codigos.txt
Primero codigo: 1234
Segundo codigo: 4567
Tercer codigo: 7891
AndMolLop 03/12/16:~$ _
```

*Ilustración 3. Visualización de las carpetas necesarias y el contenido de codigo*

Lo siguiente es crear un script que automatice el proceso de copiar el contenido de codigo en un nuevo directorio en seguridad que tenga por nombre la fecha del día en que se ejecuta. Para esto yo he creado el script que muestro en la *Ilustración 4*.

```
GNU nano 2.5.3          Archivo: ./copia.sh

#!/bin/bash
#Autor: Andres Molina Lopez
#Fecha: 03/12/16
#Descripción: copia el contenido de la carpeta ~/codigo a la carpeta ~/seguridad/$fecha
FECHA=$(date +%d_%m_%y)
directorio=$HOME/seguridad/$FECHA
mkdir $directorio
cp -R $HOME/codigo $directorio
```

*Ilustración 4. Script para automatizar la copia de codigo a seguridad/fecha*

Por último, tenemos que añadir el script a *crontab* para que cron pueda ejecutarlo periódicamente. Para ello ejecutamos *crontab -e* para editar el archivo y poder añadir la tarea a cron. La añadimos escribiendo la siguiente línea *0 0 \* \* \* \$HOME/copia.sh*, lo que significa que ejecutará el script de copia a las 00:00 de cada día de cada mes todos los días de la semana. Esto también se puede poner como *@daily \$HOME/copia.sh*. En la *Ilustración 5* muestro como he añadido la ejecución del script a cron.

```
# Edit this file to introduce tasks to be run by cron.
#
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h dom mon dow  command
0 0 * * * $HOME/copia.sh

crontab: installing new crontab
AndMolLop 03/12/16:~$ ls seguridad/
AndMolLop 03/12/16:~$
```

*Ilustración 5. Adición de una nueva tarea a cron*

[3] (Pérez Esteso, 2016)

**3ª Cuestión: 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.**

Lo primero es ejecutar *dmesg* | *tail* para que no muestre tanta información, ya que ocupa mucho más de lo que podemos visualizar. Al ejecutar este comando obtengo el resultado que muestro en la *Ilustración 6*.

```

AndMolLop 03/12/16:~$ dmesg | tail
[ 52.210775] audit: type=1400 audit(1480767665.772:6): apparmor="STATUS" operation="profile_load"
profile="unconfined" name="/usr/lib/NetworkManager/nm-dhcp-helper" pid=1110 comm="apparmor_parser"
[ 52.210778] audit: type=1400 audit(1480767665.772:7): apparmor="STATUS" operation="profile_load"
profile="unconfined" name="/usr/lib/NetworkManager/nm-dhcp-client.action" pid=1111 comm="apparmor_parser"
[ 52.238744] audit: type=1400 audit(1480767665.800:8): apparmor="STATUS" operation="profile_load"
profile="unconfined" name="/usr/lib/NetworkManager/nm-dhcp-client.action" pid=1111 comm="apparmor_parser"
[ 52.238750] audit: type=1400 audit(1480767665.800:9): apparmor="STATUS" operation="profile_load"
profile="unconfined" name="/usr/lib/NetworkManager/nm-dhcp-client.action" pid=1111 comm="apparmor_parser"
[ 52.238754] audit: type=1400 audit(1480767665.800:10): apparmor="STATUS" operation="profile_load"
profile="unconfined" name="/usr/lib/NetworkManager/nm-dhcp-helper" pid=1111 comm="apparmor_parser"
[ 52.238757] audit: type=1400 audit(1480767665.800:11): apparmor="STATUS" operation="profile_load"
profile="unconfined" name="/usr/lib/NetworkManager/nm-dhcp-helper" pid=1111 comm="apparmor_parser"
[ 53.914396] IPv6: ADDRCONF(NETDEV_UP): enp0s3: link is not ready
[ 53.915601] e1000: enp0s3 NIC Link is Up 1000 Mbps Full Duplex, Flow Control: RX
[ 53.915853] IPv6: ADDRCONF(NETDEV_CHANGE): enp0s3: link becomes ready
[ 55.797356] cgroup: new mount options do not match the existing superblock, will be ignored
AndMolLop 03/12/16:~$

```

Ilustración 6. Salida del comando `dmesg | tail`

Tras esto conectamos un dispositivo USB y volvemos a ejecutar el comando, para ver si el dispositivo se ha reconocido. Muestro el resultado de que se ha reconocido el pendrive en la *Ilustración 7*.

```

AndMolLop 03/12/16:~$ dmesg | tail
[ 361.774519] usbcore: registered new interface driver usb-storage
[ 361.806817] usbcore: registered new interface driver uas
[ 362.772374] scsi 4:0:0:0: Direct-Access PNY USB 3.0 FD PMAP PQ: 0 ANSI: 6
[ 362.774090] sd 4:0:0:0: Attached scsi generic sg3 type 0
[ 362.775219] sd 4:0:0:0: [sdc] 60555264 512-byte logical blocks: (31.0 GB/28.9 GiB)
[ 362.775584] sd 4:0:0:0: [sdc] Write Protect is off
[ 362.775587] sd 4:0:0:0: [sdc] Mode Sense: 45 00 00 00
[ 362.775930] sd 4:0:0:0: [sdc] Write cache: disabled, read cache: enabled, doesn't support DPO or FUA
[ 362.949844] sdc: sdc1
[ 362.952266] sd 4:0:0:0: [sdc] Attached SCSI removable disk
AndMolLop 03/12/16:~$ _

```

Ilustración 7. Visualización del reconocimiento del USB en `dmesg`

En la *Ilustración 7* vemos que lo primero que nos dice es que se ha registrado una nueva interfaz que es un USB de almacenamiento, luego, más abajo nos dice el nombre del dispositivo y su capacidad de almacenamiento, y, por último, nos dice el nombre de partición asignado.

En la *Ilustración 8* podemos ver que ejecutando `lsblk`, se muestra efectivamente el disco `sdc1` con el tamaño anteriormente indicado en la *Ilustración 7*.

```

AndMolLop 03/12/16:~$ lsblk
NAME        MAJ:MIN RM SIZE RO TYPE MOUNTPOINT
sda         8:0    0  15G  0 disk
├─sda1      8:1    0  15G  0 part
└─┬─nd0     9:0    0  15G  0 raid1
   │├─Hds-arranq 252:0  0  140M  0 lvm /boot
   │├─Hds-raiz   252:1  0  11,2G  0 lvm
   │├─┬─Hds-raiz_crypt 252:4  0  11,2G  0 crypt /
   ││├─Hds-hogar 252:2  0  952M  0 lvm
   ││├─┬─Hds-hogar_crypt 252:6  0  950M  0 crypt /home
   ││├─Hds-swap 252:3  0  1,9G  0 lvm
   ││└─┬─Hds-swap_crypt 252:5  0  1,9G  0 crypt [SWAP]
   └─sdb     8:16   0  15G  0 disk
      └─sdb1  8:17   0  15G  0 part
         └─┬─nd0     9:0    0  15G  0 raid1
            │├─Hds-arranq 252:0  0  140M  0 lvm /boot
            │├─Hds-raiz   252:1  0  11,2G  0 lvm
            │├─┬─Hds-raiz_crypt 252:4  0  11,2G  0 crypt /
            ││├─Hds-hogar 252:2  0  952M  0 lvm
            ││├─┬─Hds-hogar_crypt 252:6  0  950M  0 crypt /home
            ││├─Hds-swap 252:3  0  1,9G  0 lvm
            ││└─┬─Hds-swap_crypt 252:5  0  1,9G  0 crypt [SWAP]
            └─sdc     8:32   1  28,9G  0 disk
               └─sdc1  8:33   1  28,9G  0 part
sr0         11:0   1 1024M  0 rom
AndMolLop 03/12/16:~$ _

```

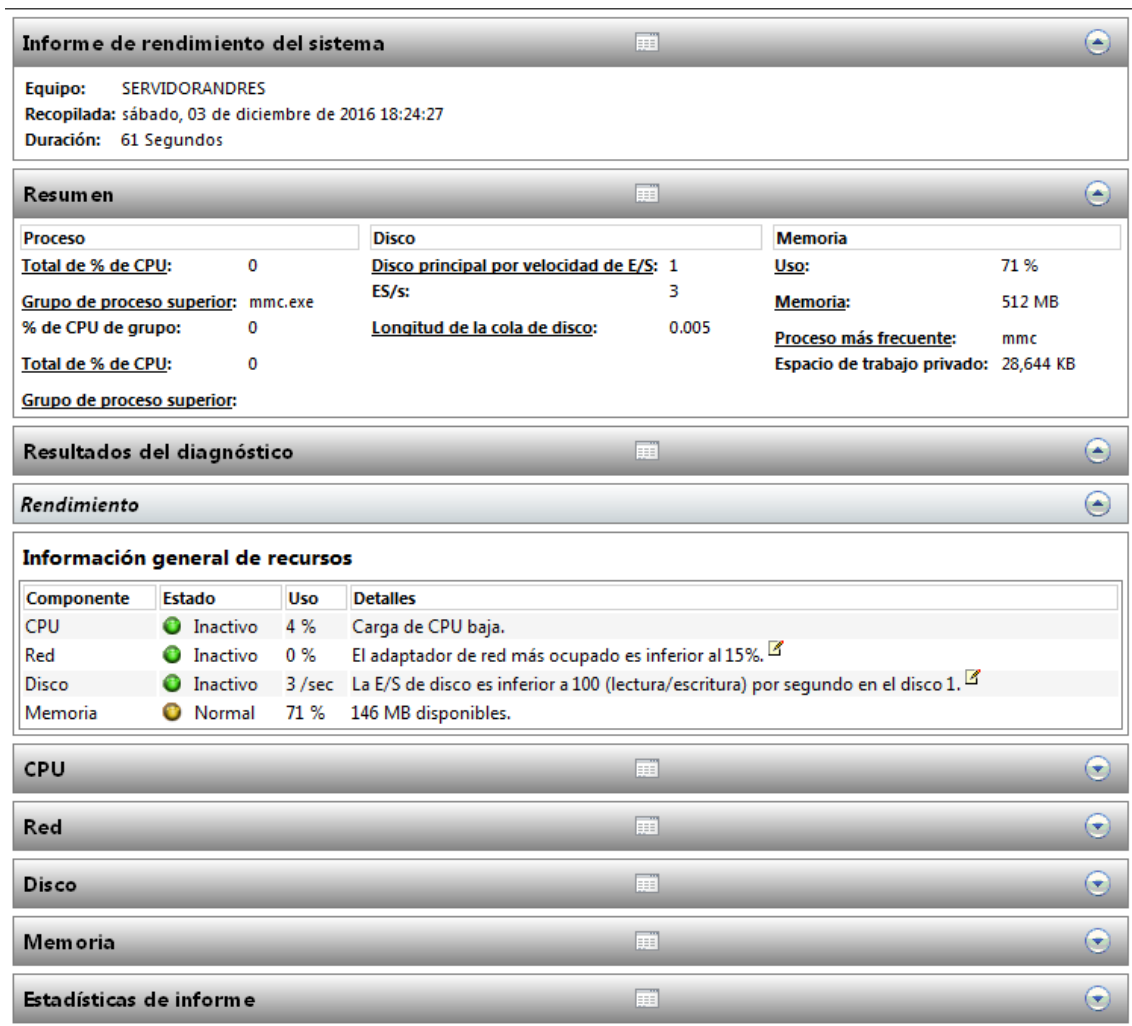
Ilustración 8. Salida de `lsblk` con el dispositivo conectado



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

Al ejecutar el Monitor de rendimiento e iniciar el Rendimiento del sistema, situado en Conjunto de recopiladores de datos, se nos genera un informe que podemos ver más abajo en Informes > Sistema > System Performance.

Al abrir el informe veremos la pantalla con el resultado del análisis, yo muestro el resultado del mío en la *Ilustración 9*.



*Ilustración 9. Informe de rendimiento del sistema*

Podemos ver que el informe se divide en 4 bloques principales (CPU, Red, Disco y Memoria), de los cuales se nos muestra una información general del uso que han tenido durante el análisis. Además de verlo así también podemos mostrarlo más detallado, entrando dentro de cada bloque, expandiendo su información o mostrando el resultado como un gráfico. Muestro la gráfica del análisis en la *Ilustración 10*, pero al hacer mediciones de tantas cosas es tan caótico que a primera vista es bastante incomprensible.

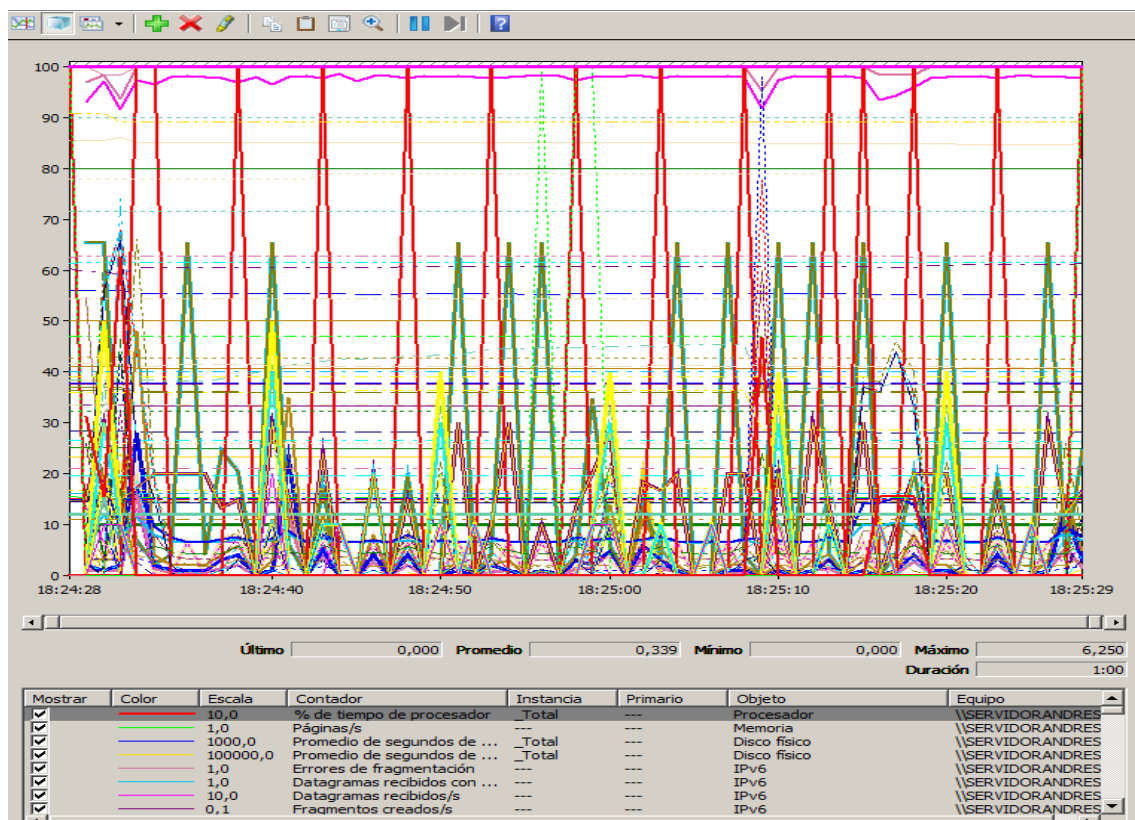


Ilustración 10. Gráfica del análisis de rendimiento

De manera que viendo las propiedades podemos cambiarlo para que se muestre como un informe, el cual es mucho más comprensible. En las *Ilustración 11 a Ilustración 16* muestro el informe de mi análisis, y en el ya si podemos ver las distintas variables que se han medido para calcular el porcentaje de uso de los distintos bloques, y además, al ser más detallado, también podemos ver el resultado de cada una de esas variables por separado.

\\SERVIDORANDRES				
Disco físico				
% de tiempo de disco	_Total	0 R:	1 C:	2 R:
% de tiempo de escritura en disco	0,014	---	---	---
% de tiempo de lectura de disco	0,014	---	---	---
% de tiempo inactivo	100,339	---	---	---
Bytes de disco/s	37,577,439	---	---	---
Bytes de escritura en disco/s	37,577,439	---	---	---
Bytes de lectura de disco/s	0,000	---	---	---
ES divididas/s	0,033	---	---	---
Escripciones en disco/s	2,384	---	---	---
Lecturas de disco/s	0,000	---	---	---
Longitud actual de la cola de disco	0,000	---	---	---
Longitud promedio de cola de escritura de disco	0,000	---	---	---
Longitud promedio de cola de lectura de disco	0,000	---	---	---
Longitud promedio de la cola de disco	0,000	---	---	---
Promedio de bytes de disco/escritura	19,985,258	---	---	---
Promedio de bytes de disco/lectura	0,000	---	---	---
Promedio de bytes de disco/transerencia	19,985,258	---	---	---
Promedio de segundos de disco/escritura	0,000	0,000	0,000	0,000
Promedio de segundos de disco/lectura	0,000	0,000	0,000	0,000
Promedio de segundos de disco/transerencia	0,000	---	---	---
Transferencias de disco/s	2,384	---	---	---
Interfaz de red				
Adaptador de escritorio Intel[R] PRO_1000 MT				
Ancho de banda actual	1.000.000,000	isatap.telefonica.net	100.000,000	
Bytes enviados/s	11,767		0,000	
Bytes recibidos/s	41,008		0,000	
Conexiones descargadas	0,000		0,000	
Longitud de la cola de salida	0,000		0,000	
Paquetes de no unidifusión enviados/s	0,117		0,000	
Paquetes de no unidifusión recibidos/s	0,483		0,000	
Paquetes de salida con errores	0,000		0,000	
Paquetes de salida descartados	0,000		0,000	
Paquetes de unidifusión enviados/s	0,000		0,000	
Paquetes de unidifusión recibidos/s	0,000		0,000	
Paquetes enviados/s	0,117		0,000	
Paquetes recibidos con errores	0,000		0,000	
Paquetes recibidos descartados	0,000		0,000	
Paquetes recibidos desconocidos	0,000		0,000	
Paquetes recibidos/s	0,483		0,000	
Paquetes/s	0,600		0,000	
Total de bytes/s	52,775		0,000	

Ilustración 11. Informe del análisis de rendimiento parte 1

<b>IPv4</b>		
Datagramas de salida descartados		0,000
Datagramas de salida sin enrutamiento		9,000
Datagramas enviados/s		0,017
Datagramas fragmentados/s		0,000
Datagramas recibidos con errores de dirección		0,000
Datagramas recibidos con errores de encabezado		0,000
Datagramas recibidos con protocolo desconocido		0,000
Datagramas recibidos descartados	79,016	
Datagramas recibidos entregados/s	0,200	
Datagramas recibidos/s	0,217	
Datagramas reenviados/s	0,000	
Datagramas/s	0,233	
Errores de fragmentación	0,000	
Errores de reensamblado de fragmentos	0,000	
Fragmentos creados/s	0,000	
Fragmentos recibidos/s	0,000	
Fragmentos reensamblados/s	0,000	
<b>IPv6</b>		
Datagramas de salida descartados	0,000	
Datagramas de salida sin enrutamiento	5,000	
Datagramas enviados/s	0,100	
Datagramas fragmentados/s	0,000	
Datagramas recibidos con errores de dirección	0,000	
Datagramas recibidos con errores de encabezado	0,000	
Datagramas recibidos con protocolo desconocido	0,000	
Datagramas recibidos descartados	8,000	
Datagramas recibidos entregados/s	0,233	
Datagramas recibidos/s	0,233	
Datagramas reenviados/s	0,000	
Datagramas/s	0,333	
Errores de fragmentación	0,000	
Errores de reensamblado de fragmentos	0,000	
Fragmentos creados/s	0,000	
Fragmentos recibidos/s	0,000	
Fragmentos reensamblados/s	0,000	

Ilustración 12. Informe del análisis de rendimiento parte 2

<b>Memoria</b>		
% de bytes confirmados en uso	26,497	
Asignaciones de bloque no paginado	32.279,951	
Asignaciones de bloque paginado	39.077,230	
Bytes confirmados	426.638.151,344	
Bytes de bloque no paginado	19.671.677,902	
Bytes de bloque paginado	62.694.114,623	
Bytes de caché	15.145.396,459	
Bytes de lista de páginas libres y cero	60.660.954,230	
Bytes de lista de páginas modificadas	4.074.512,787	
Bytes de prioridad normal de caché en modo de espera	89.218.199,082	
Bytes de reserva de caché en modo de espera	3.372.753,836	
Bytes disponibles	153.251.907,148	
Bytes principales de caché en modo de espera	0,000	
Bytes residentes de bloque paginado	61.475.386,754	
Bytes residentes de caché del sistema	15.145.396,459	
Bytes residentes de código del sistema	2.594.648,131	
Bytes residentes de controladores del sistema	5.447.680,000	
Copias de escritura/s	0,316	
Entrada de páginas/s	0,000	
Entradas libres de la tabla de páginas del sistema	33.555.166,000	
Errores de caché/s	1,283	
Errores de página/s	85,144	
Errores de solicitud de cero/s	66,817	
Errores de transición/s	17,661	
Escrituras de páginas/s	0,000	
KB disponibles	149.660,066	
Lecturas de páginas/s	0,000	
Límite de confirmación	1.610.145,792	
Mbytes disponibles	145,820	
Páginas de transición reasignadas/s	0,000	
Páginas/s	0,000	
Salida de páginas/s	0,000	
Total de bytes de código del sistema	7.159.808,000	
Total de bytes de controladores del sistema	4.694.016,000	
Uso máximo de los bytes de caché	21.143.552,000	
<b>Procesador</b>		
	<b>_Total</b>	<b>0</b>
% de tiempo de DPC	0,000	---
% de tiempo de interrupción	0,000	---
% de tiempo de procesador	0,339	0,339
% de tiempo de usuario	0,052	---
% de tiempo en C1	97,409	---
% de tiempo en C2	0,000	---
% de tiempo en C3	0,000	---
% de tiempo inactivo	99,661	---
% de tiempo privilegiado	0,286	---
DPC en cola/s	9,586	---
Interrupciones/s	79,617	---
Transiciones a C1/s	71,714	---
Transiciones a C2/s	0,000	---
Transiciones a C3/s	0,000	---
Velocidad de DPC	0,246	---

Ilustración 13. Informe del análisis de rendimiento parte 3

Proceso	Total
% de tiempo de procesador	100,000
% de tiempo de usuario	0,052
% de tiempo privilegiado	99,948
Bytes de bloque no paginado	553.719,541
Bytes de bloque paginado	3.771.087,213
Bytes de datos ES/s	30.466,862
Bytes de escritura de ES/s	30.158,590
Bytes de lectura de ES/s	308,272
Bytes del archivo de paginación	232.978.801,311
Bytes privados	232.978.801,311
Bytes virtuales	2.815.200,676
Errores de página/s	55,141
Espacio de trabajo	361.206.834,361
Espacio de trabajo - Privado	143.616.906,492
Id. de proceso	0,000
Id. de proceso de creación	0,000
Número de identificadores	8.499,525
Número de subprocesos	412,262
Operaciones de ES de datos/s	7,502
Operaciones de ES de escritura/s	1,550
Operaciones de ES de lectura/s	5,951
Otras operaciones de ES/s	61,873
Otros bytes de ES/s	5.411,222
Prioridad base	0,000
Tiempo transcurrido	0,000
Uso máximo de bytes virtuales	3.326.712,916
Uso máximo de los bytes del archivo de paginación	250.321.265,311
Uso máximo del espacio de trabajo	406.459.643,803

Ilustración 14. Informe del análisis de rendimiento parte 4

<b>Servidor</b>	
Archivos abiertos	0,000
Bloques de contexto en cola/s	0,000
Búsquedas de directorio de archivos	0,000
Bytes de bloque no paginado	119.639,000
Bytes de bloque paginado	37.712,000
Bytes hash enviados de BranchCache para SMB	0,000
Bytes recibidos/s	0,000
Bytes transmitidos/s	0,000
Errores de acceso concedido	0,000
Errores de bloque no paginado	0,000
Errores de bloque paginado	0,000
Errores de inicio de sesión	0,000
Errores de permisos de acceso	0,000
Errores del sistema	0,000
Falta de elementos de trabajo	0,000
Identificadores duraderos reconectados	0,000
Identificadores resistentes reconectados	0,000
Inicios de sesión/s	0,000
Respuestas hash enviadas de BranchCache para SMB	0,000
Sesiones cerradas	0,000
Sesiones cerradas forzosamente	0,000
Sesiones cerradas por un error	0,000
Sesiones cuyo tiempo de espera terminó	0,000
Sesiones del servidor	1,000
Solicitudes de bloqueo rechazadas	0,000
Solicitudes de encabezado de hash de BranchCache para SMB	0,000
Solicitudes de generación de hash de BranchCache para SMB	0,000
Solicitudes hash recibidas de BranchCache para SMB	0,000
Total de archivos abiertos	0,000
Total de bytes/s	0,000
Total de identificadores duraderos	0,000
Total de identificadores resistentes	0,000
Total de inicios de sesión	1,000
Uso máximo del bloque no paginado	119.913,000
Uso máximo del bloque paginado	37.792,000
<b>Sistema</b>	
% de cuota de Registro en uso	1,958
Bytes de control de archivo/s	5.419,696
Bytes de escritura de archivo/s	30.158,590
Bytes de lectura de archivo/s	308,272
Cambios de contexto/s	307,481
Correcciones de alineación/s	0,000
Emulaciones de punto flotante/s	0,000
Envíos de excepciones/s	12,984
Llamadas al sistema/s	3.271,091
Longitud de la cola del procesador	0,082
Operaciones con datos de archivo/s	7,502
Operaciones de control de archivo/s	61,526
Operaciones de escritura de archivo/s	1,550
Operaciones de lectura de archivo/s	5,951
Procesos	36,066
Subprocesos	412,262
Tiempo de actividad del sistema	1.707,031

Ilustración 15. Informe del análisis de rendimiento parte 5

<b>TCPv4</b>	
Conexiones activas	5,000
Conexiones establecidas	0,000
Conexiones pasivas	4,000
Conexiones reinicializadas	5,000
Errores de conexión	0,000
Segmentos enviados/s	0,000
Segmentos recibidos/s	0,000
Segmentos retransmitidos/s	0,000
Segmentos/s	0,000
<b>TCPv6</b>	
Conexiones activas	0,000
Conexiones establecidas	0,000
Conexiones pasivas	0,000
Conexiones reinicializadas	0,000
Errores de conexión	0,000
Segmentos enviados/s	0,000
Segmentos recibidos/s	0,000
Segmentos retransmitidos/s	0,000
Segmentos/s	0,000
<b>UDPv4</b>	
Datagramas enviados/s	0,000
Datagramas recibidos con errores	11,000
Datagramas recibidos/s	0,000
Datagramas sin puerto/s	0,033
Datagramas/s	0,000
<b>UDPv6</b>	
Datagramas enviados/s	0,000
Datagramas recibidos con errores	8,000
Datagramas recibidos/s	0,000
Datagramas sin puerto/s	0,000
Datagramas/s	0,000

*Ilustración 16. Informe del análisis de rendimiento parte 6*

Como vemos en estas 6 ilustraciones, hay muchísimas variables de las cuales se han hecho mediciones. Estas variables se agrupan en distintos bloques más pequeños, que al juntarlos nos dan los resultados de los bloques principales, así el bloque principal de CPU está formado por los bloques de procesador, proceso, servidor y sistema, el bloque principal de Red está formado por los bloques interfaz de red, IPv4, IPv6, TCPv4, TCPv6, UDPv4 y UDPv6, el bloque principal de Disco está formado por el bloque disco físico y el bloque principal de Memoria está formado por el bloque de memoria.

**5ª Cuestión: Cree un recopilador de datos definido por el usuario (modo avanzada) que incluya tanto el contador de rendimiento como los datos de seguimiento:**

**Todos los referentes al procesador, proceso y al servicio web.**

**Intervalo de muestra de 15 segundos.**

**Almacene el resultado en el directorio Escritorio/logs.**

**Incluya las capturas de pantalla de cada paso.**

Estando en el Monitor de rendimiento nos vamos a la pestaña de Conjunto de recopiladores de datos > Definidos por el usuario, y le damos click derecho y a Nuevo > Conjunto de recopiladores de datos, con lo que nos abrirá la ventana que muestro en la *Ilustración 17*.

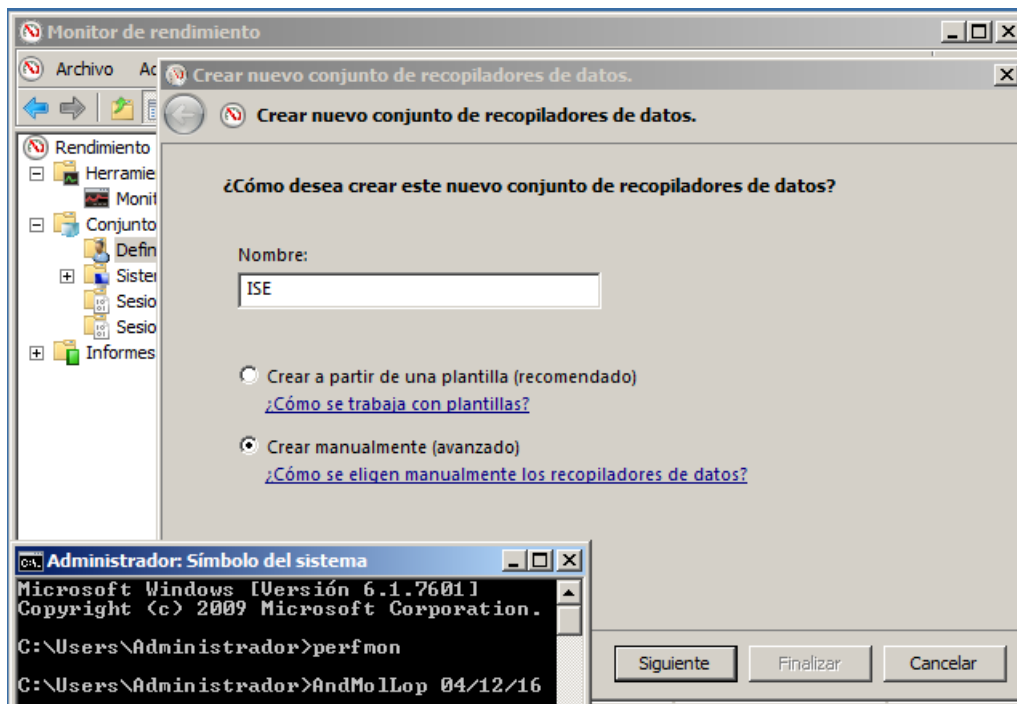


Ilustración 17. Creación de conjunto de recopiladores de datos parte 1

En esta ventana de la *Ilustración 17* simplemente le ponemos un nombre, marcamos crear manualmente y le damos a siguiente, con lo que nos llevará a la ventana de la *Ilustración 18*, en la cual marcamos Crear registros de datos, y en él, Contador de rendimiento y Datos de seguimiento de eventos, y le damos a siguiente.

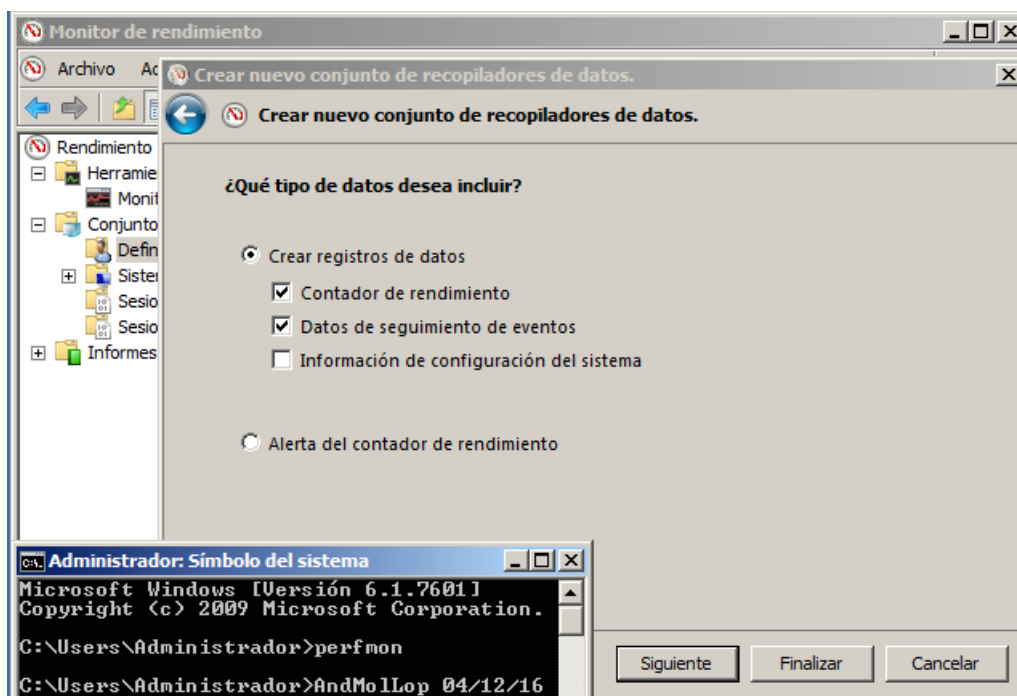
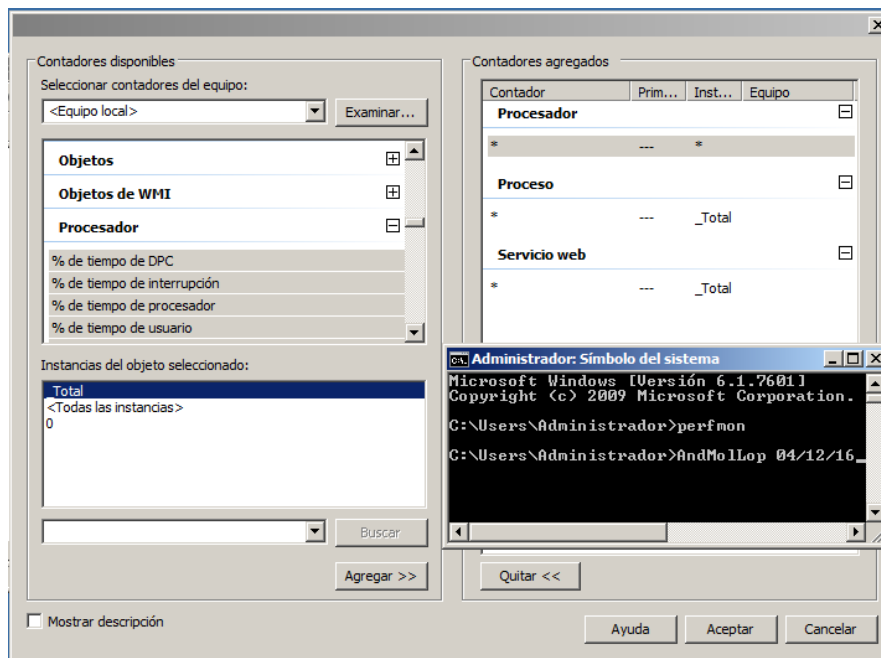


Ilustración 18. Creación de conjunto de recopiladores de datos parte 2

En la siguiente ventana que se nos abre, dejamos el intervalo de muestra en 15 segundos ya que es lo que queremos, y le damos al botón de agregar. Esto nos llevará a

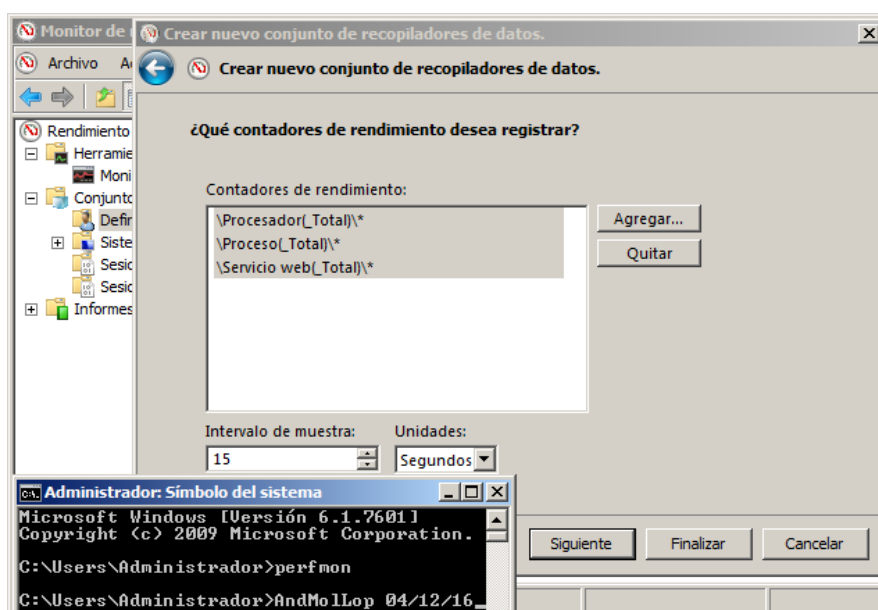


otra ventana en la que tenemos que buscar en una lista procesador, proceso y servicio web. Según los vamos encontrando, le damos doble click encima para abrirlo y marcarlo, y le damos al botón de agregar, así pasaremos a verlo en el lado izquierdo de la ventana donde pone Contadores agregados (el asterisco debajo del nombre del contador quiere decir que se ha agregado todos sus subcontadores. Muestro esta ventana ya configurada en la *Ilustración 19*.



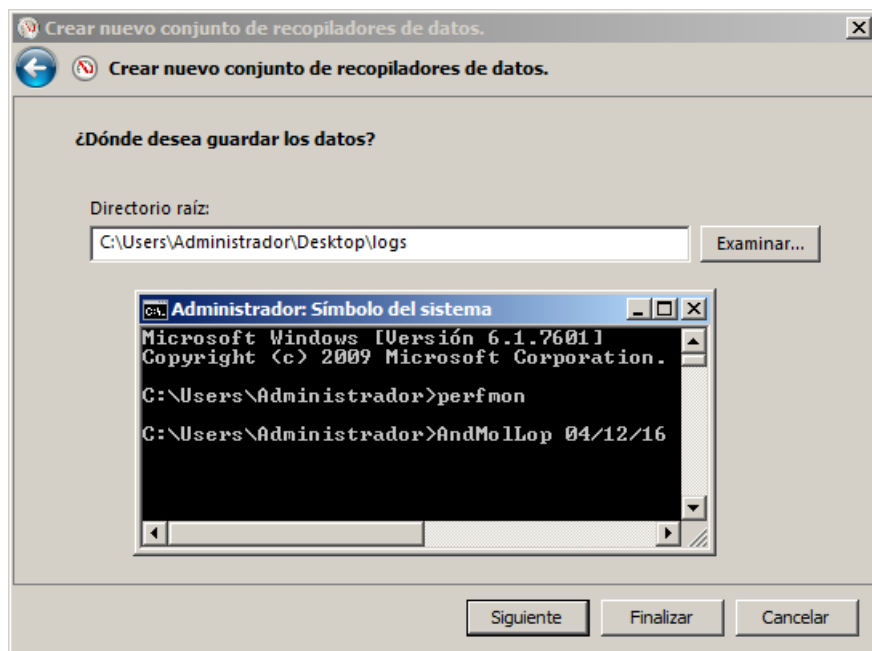
*Ilustración 19. Creación de conjunto de recopiladores de datos parte 3*

Una vez le damos a aceptar en la ventana de la *Ilustración 19*, nos devuelve a la ventana anterior, en la que ya si estará rellena la lista de Contadores de rendimiento. Aquí comprobamos que todo esté como queremos y le damos a siguiente. Muestro esta ventana de confirmación en la *Ilustración 20*.



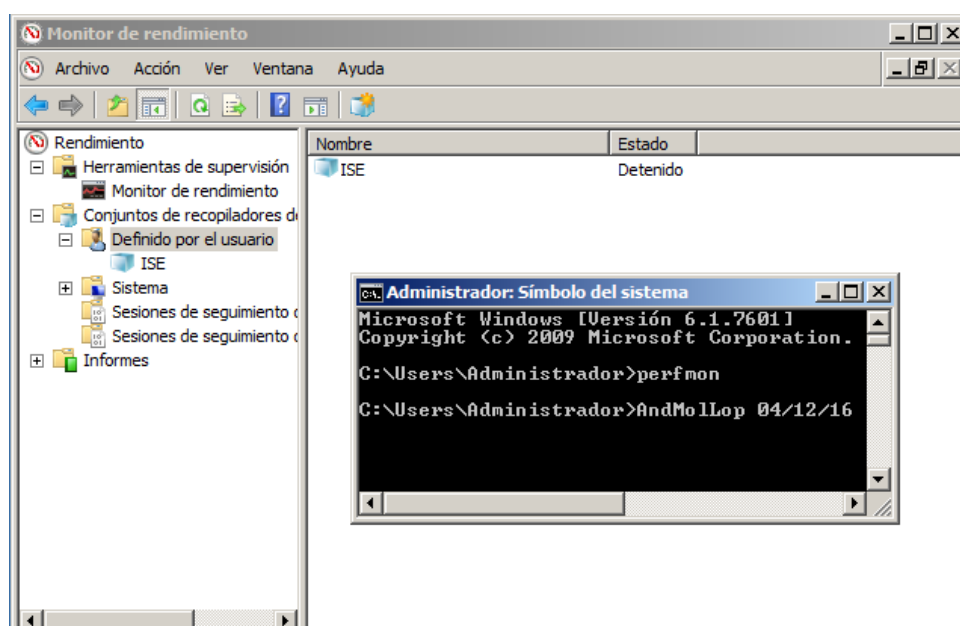
*Ilustración 20. Creación de conjunto de recopiladores de datos parte 4*

En la siguiente venta que nos saldrán proveedores de seguimiento simplemente le damos a siguiente sin agregar ninguno. Esto nos llevará a una ventana en la cual tenemos que elegir el directorio donde queremos que se guarden los informes, en mi caso le doy a examinar, y busco Escritorio/logs, como muestro en la *Ilustración 21*.



*Ilustración 21. Creación de conjunto de recopiladores de datos parte 5*

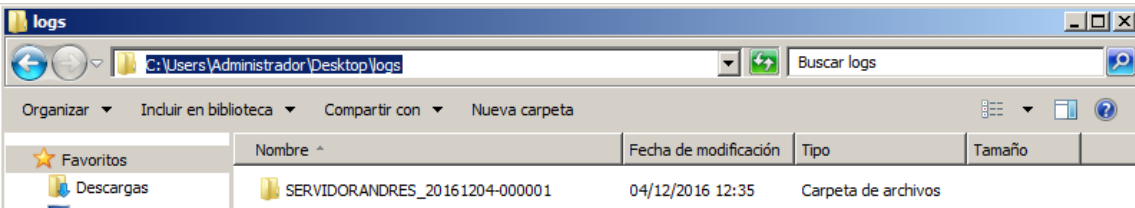
Al darle a siguiente en esta ventana (*Ilustración 21*), no llevará a la última, la cual la dejamos como está con Guardar y cerrar marcado y le damos a finalizar. Tras esto, en el Monitor de rendimiento, en la parte de Conjuntos de recopiladores de datos > Definido por el usuario, veremos que hay uno que se llama como nosotros hemos indicado al principio. Lo muestro en la *Ilustración 22*.



*Ilustración 22. Conjunto de recopilación de datos creado*

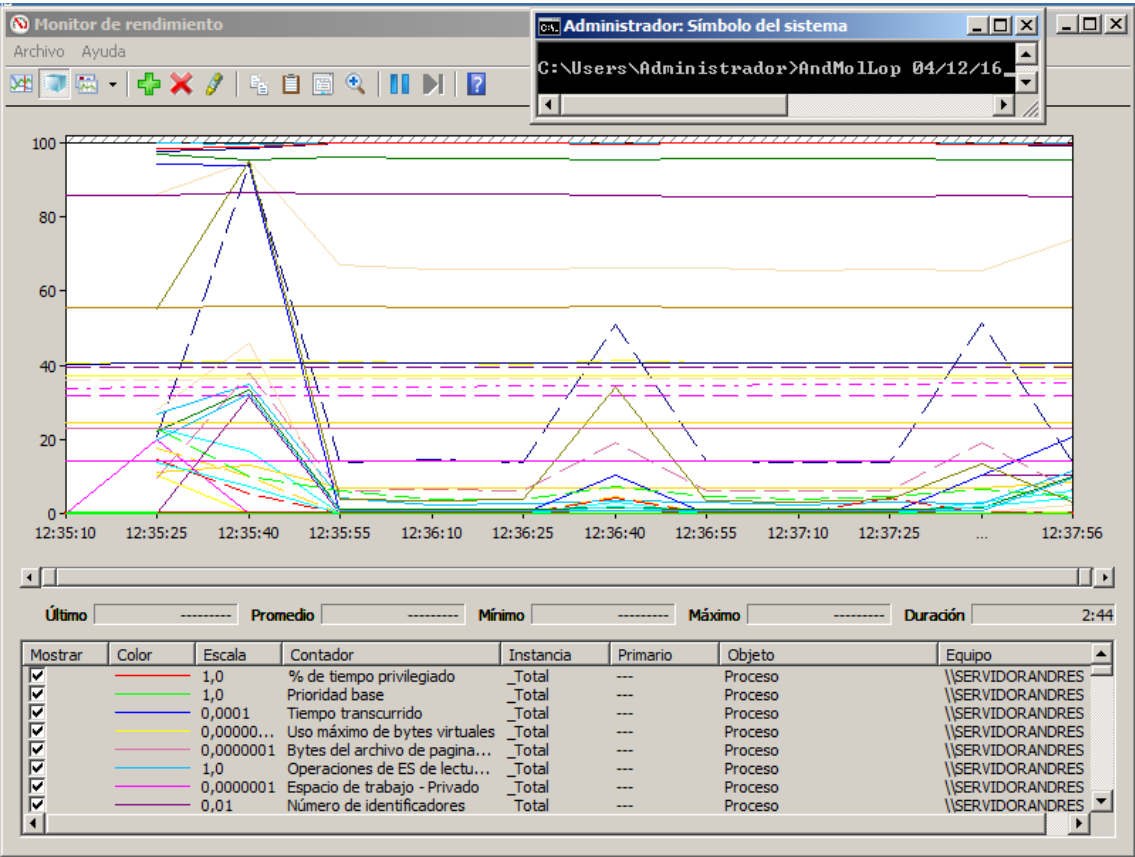


Ahora si le damos a click derecho en él y le damos a iniciar, se nos generará un informe en la carpeta indicada. Como vemos en la *Ilustración 23*, a mí se me ha generado correctamente.



*Ilustración 23. Informe generado correctamente por el conjunto de recopiladores personalizado*

Y dentro de esas sub carpeta están los archivos creados, en los cuales podemos ver el gráfico de análisis. Muestro el obtenido en la *Ilustración 24*.



*Ilustración 24. Gráfico del análisis personalizado*

[4] ("Crear un conjunto de recopiladores de datos desde el Monitor de rendimiento", 2016)

**6ª Cuestión: Visite la web del proyecto y acceda a la demo que proporcionan (<http://demo.munin-monitoring.org/>) donde se muestra cómo monitorizan un servidor. Monitorice varios parámetros y haga capturas de pantalla de lo que está mostrando comentando qué observa.**

Lo primero que hacemos es instalar Munin (yo en mi caso lo he hecho en CentOS 7 porque tengo GUI), para instalarlo ejecutamos `yum install epel-release` para instalar el repositorio EPEL, después ejecutamos `yum install munin munin-node` para instalar munin propiamente. Una vez instalado, ejecutamos `systemctl enable munin-node`, `systemctl start munin-node` para habilitarlo y que comience a funcionar el servicio. Si tras esto ejecutamos `systemctl status munin-node` deberemos ver una información como la que muestro en la *Ilustración 25* en la que se nos dice que el servicio está activo y funcionando.

```
[AndMolLop 05/12/16 ~]#systemctl status munin-node
● munin-node.service - Munin Node Server.
   Loaded: loaded (/usr/lib/systemd/system/munin-node.service; enabled; vendor p
  reset: disabled)
   Active: active (running) since lun 2016-12-05 19:02:06 CET; 14s ago
     Docs: man:munin-node
   Process: 4201 ExecStart=/usr/sbin/munin-node (code=exited, status=0/SUCCESS)
    Main PID: 4202 (munin-node)
      CGroup: /system.slice/munin-node.service
              └─4202 /usr/bin/perl -wT /usr/sbin/munin-node

dic 05 19:02:05 localhost.localdomain systemd[1]: Starting Munin Node Server....
dic 05 19:02:06 localhost.localdomain systemd[1]: Started Munin Node Server..
Hint: Some lines were ellipsized, use -l to show in full.
[AndMolLop 05/12/16 ~]#
```

*Ilustración 25. Status de munin-node*

Lo siguiente es configurar el servidor con Munin, para ello vamos a `/etc/httpd/conf.d/munin.conf`. Y añadimos las siguientes líneas para decir quien tiene acceso a Munin desde el servidor.

```
GNU nano 2.3.1      Fichero: /etc/httpd/conf.d/munin.conf      Modifi
# This file can be used as a .htaccess file, or a part of your apache
# config file.
#
# For the .htaccess file option to work the munin www directory
# (/var/www/html/munin) must have "AllowOverride all" or something close
# to that set.
#
# As a config file enclose it in <directory> like so:
<directory /var/www/html/munin>

AuthUserFile /etc/munin/munin-htpasswd
AuthName "Munin"
AuthType Basic
require valid-user
Order Deny,Allow
Deny from all
Allow from 127.0.0.1 10.0.2.15
```

*Ilustración 26. Modificación de la configuración de Munin para poner desde donde se puede acceder*

Tras esto reiniciamos apache2 con `systemctl restart httpd`. Lo siguiente es ir al archivo `/etc/munin/munin.conf` y descomentar las 4 líneas que mostramos en la *Ilustración 27*.

```
GNU nano 2.3.1      Fichero: /etc/munin/munin.conf

# Example configuration file for Munin, generated by 'make build'

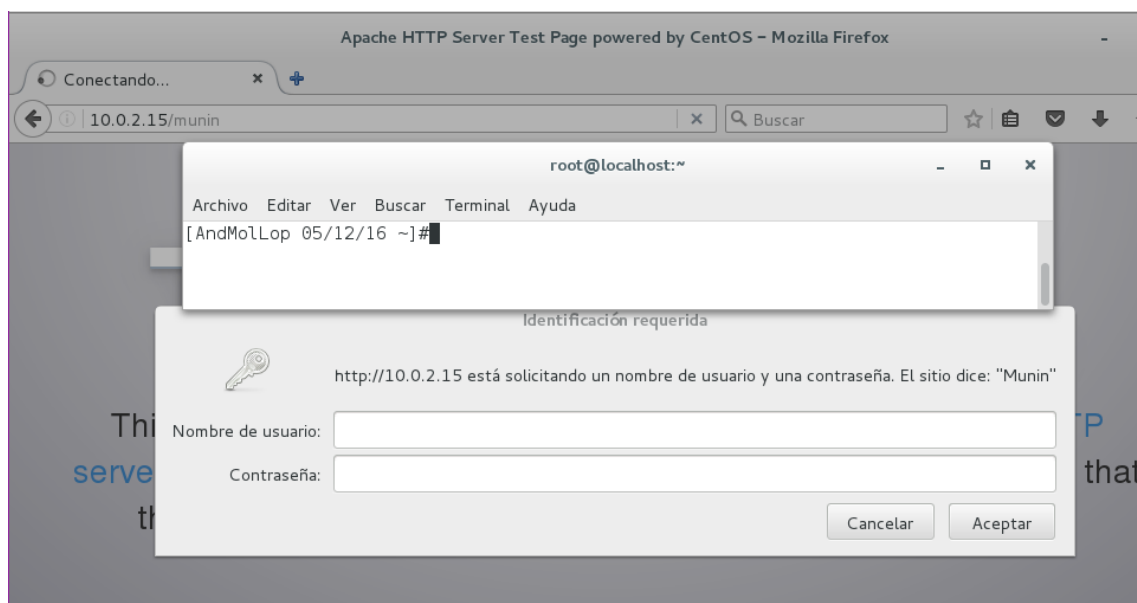
# The next three variables specifies where the location of the RRD
# databases, the HTML output, logs and the lock/pid files. They all
# must be writable by the user running munin-cron. They are all
# defaulted to the values you see here.
#
dbdir    /var/lib/munin
htmldir  /var/www/html/munin
logdir   /var/log/munin
rundir   /var/run/munin
```

*Ilustración 27. Modificación de la configuración de Munin*

Proseguimos ejecutando `chown -R munin:munin /var/www/html/munin/` para modificar la propiedad de los archivos.

Luego hacemos que el apartado de Munin en el servidor requiera de un usuario y contraseña, por lo que ejecutamos `htpasswd -c /etc/munin/munin-htpasswd username` para crearlo.

Y con esto ya está listo, y podemos acceder a él a través de `http://nuestra_IP/munin`, lo que nos abrirá una ventana para identificarnos (con el usuario creado anteriormente) y tras esto ya podemos usar munin sin problemas. Muestro en la *Ilustración 28* la solicitud de identificación para el acceso a munin, y la *Ilustración 29* la ventana que vemos una vez hemos accedido a él.



*Ilustración 28. Solicitud de identificación para acceso a Munin*

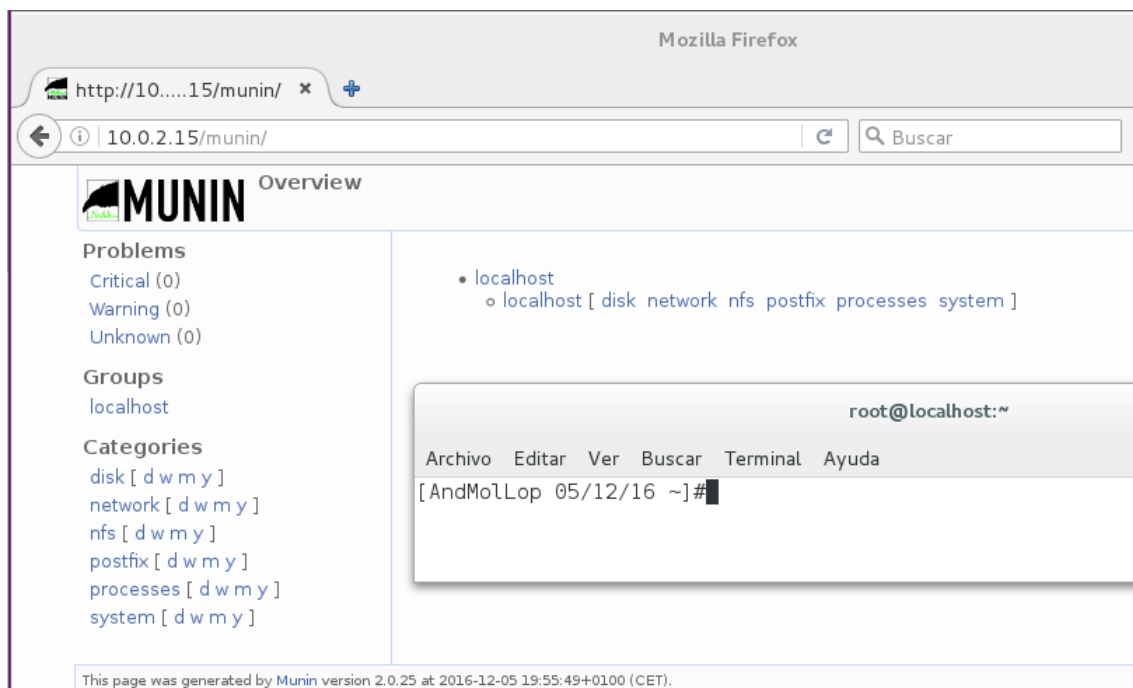


Ilustración 29. Ventana inicial de Munin

Y ahora ya podemos hacer mediciones sobre distintos parámetros del servidor solo con pulsar en la categoría en el lado izquierdo. Por ejemplo, en la *Ilustración 30* muestro la latencia media de la partición /root por día (la gráfica es chica, porque lleva poco tiempo funcionando Munin).

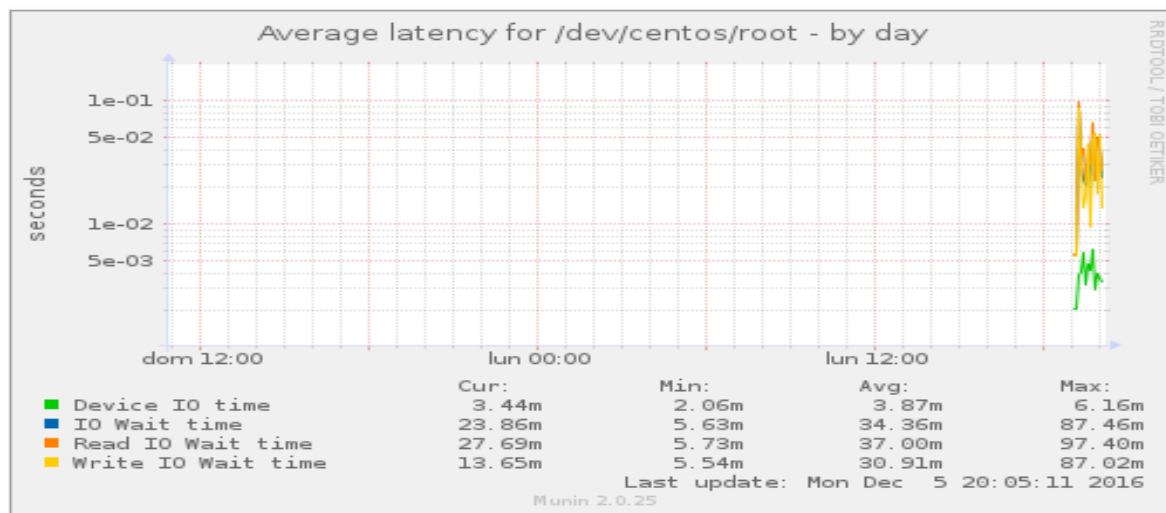


Ilustración 30. Latencia media de la partición /root

La siguiente descripción de las variables que se muestran en la gráfica es una traducción literal de la información que ofrece Munin en su página web ([http://demo.munin-monitoring.org/munin-monitoring.org/demo.munin-monitoring.org/diskstats\\_latency/sda.html](http://demo.munin-monitoring.org/munin-monitoring.org/demo.munin-monitoring.org/diskstats_latency/sda.html)):

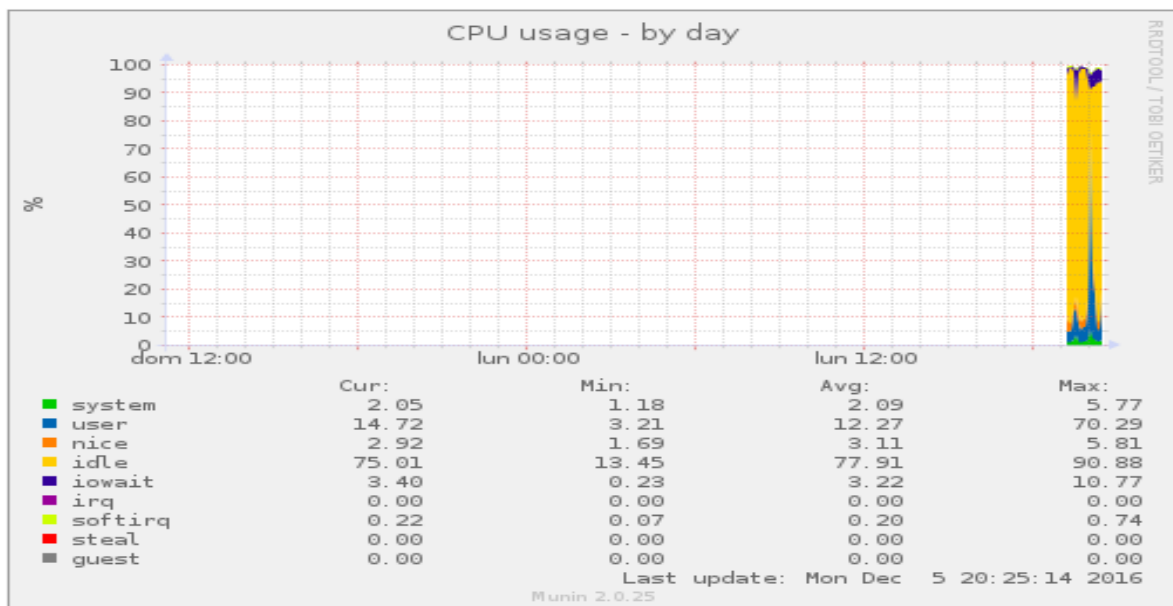
“Device IO time: tiempo medio que toma una E/S en el bloque del dispositivo sin incluir los tiempos de cola, solo el tiempo de ida y vuelta para la solicitud de disco.

IO Wait time: tiempo medio de espera para una E/S desde el inicio de la solicitud (incluye tiempos de cola y otros).

Read IO Wait time: tiempo medio de espera para una E/S de lectura desde el inicio de la solicitud hasta el final (incluye tiempos de cola y otros).

Write IO Wait time: tiempo medio de espera para una E/S de escritura desde el inicio de solicitud hasta el final (incluye tiempos de cola y otros).”

En la *Ilustración 31* muestro otro ejemplo de medición, en este caso de porcentaje de CPU usado al día.



*Ilustración 31. Medición del uso de CPU por día*

Nuevamente la descripción de las variables vuelve a ser una traducción literal de la descripción que ofrece Munin en su página web (<http://demo.munin-monitoring.org/munin-monitoring.org/demo.munin-monitoring.org/cpu.html>):

“System: tiempo de CPU usado por el kernel en actividades del sistema.

User: tiempo de CPU usado por programas normales y demonios.

Nice: tiempo de CPU usado por programas de prioridad 1.

Idle: tiempo de CPU inactiva.

Iowait: tiempo que la CPU ha estado esperando para que acaban operación de E/S cuando no había otra cosa que hacer.

Irq: tiempo de CPU dedicado al manejo de interrupciones.

Softirq: tiempo de CPU dedicado al manejo de interrupciones “agrupadas”.

Steal: tiempo que una CPU virtual tenía tareas ejecutables pero la CPU no estaba funcionando.

Guest: tiempo dedicado a ejecutar una CPU virtual para sistemas operativos invitados.”

Esto son tan solo dos ejemplos de las muchas mediciones del servidor que se pueden hacer con Munin.

[5] (*Server monitoring with Munin in centos 7*, 2016)

[6] (Munin, 2016)

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

*Strace* es una de las herramientas más potentes que ofrecen las distribuciones Linux a la hora de administrar un sistema, ya que nos muestra que está pasando en la ejecución de un servicio o programa, enseñándonos las llamadas del sistema que este hace, de manera que se pueden detectar los fallos de estas de manera rápida y fácil.

[7] ("Sysadmin Tips and Tricks - Using strace to Monitor System Calls | SoftLayer Blog", 2016)

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

He elegido como lenguaje Python y el programa que he escrito ha sido la ordenación por burbuja, la cual tiene una eficiencia bastante mala, del orden de  $O(n^2)$ , y, además, le he añadido al programa que genere un vector de 10000 elementos de manera aleatoria, el cual será el vector a ordenar.

El programa lo he dividido en una serie de funciones más chicas, para que en el profiler se pueda ver bien el tiempo que está dentro de cada función haciendo cada cosa.

En la *Ilustración 32* muestro el programa descrito anteriormente, y en la *Ilustración 33* muestro la salida del profiler de Python cProfile, el cual se ejecuta escribiendo en la terminal *Python -m cProfile programa.py* y nos muestra por pantalla tanto la salida de la ejecución del programa (no la muestro en la ilustración por ser demasiado larga) como el tiempo en cada llamada y el número de llamadas realizadas.

```

1 #Autor: Andres Molina Lopez
2 #Fecha: 06/18/16
3 #Descripción: Genera un vector de 10000 de manera aleatoria, y lo ordena de menor a mayor usando el algoritmo
  de ordenación por burbuja
4 import random
5 def aleatorio():
6     return random.randint(1,100000)
7
8 def incrementar(a):
9     return a+1
10
11 def mostrar(a):
12     print a
13
14 def comparar(a,b):
15     if(a>b):
16         return 1
17     else:
18         return 0
19
20 def intercambiar(a,b):
21     return b,a
22
23 elementos = []
24 i=0
25 while(i<10000):
26     elementos.append(aleatorio())
27     i=incrementar(i)
28
29 tam = len(elementos)
30 j=0
31 while(j<tam):
32     k=j
33     while(k<tam):
34         comparacion = comparar(elementos[j],elementos[k])
35         if(comparacion == 1):
36             elementos[j],elementos[k] = intercambiar(elementos[j],elementos[k])
37             k=incrementar(k)
38         j=incrementar(j)
39
40 for elemento in elementos:
41     mostrar(elemento)

```

Ilustración 32. Programa en Python de ordenación por burbuja

Archivo Editar Ver Buscar Terminal Ayuda

99904  
99905  
99905  
99913  
99940  
99960

124467835 function calls in 77.325 seconds

Ordered by: standard name

ncalls	tottime	percall	cumtime	percall	filename:lineno(function)
1	0.000	0.000	0.000	0.000	future_.py:48(<module>)
1	0.000	0.000	0.000	0.000	future_.py:74(Feature)
7	0.000	0.000	0.000	0.000	future_.py:75( <code>__init__</code> )
10000	0.127	0.000	0.127	0.000	burbuja.py:11(mostrar)
50005000	8.601	0.000	8.601	0.000	burbuja.py:14(comparar)
24377784	4.324	0.000	4.324	0.000	burbuja.py:20(intercambiar)
1	55.853	55.853	77.325	77.325	burbuja.py:4(<module>)
10000	0.003	0.000	0.017	0.000	burbuja.py:5(aleatorio)
50025000	8.398	0.000	8.398	0.000	burbuja.py:8(incrementar)
1	0.003	0.003	0.003	0.003	hashlib.py:73(<module>)
6	0.000	0.000	0.000	0.000	hashlib.py:89( <code>__get_openssl_constructor</code> )
1	0.000	0.000	0.000	0.000	random.py:100(seed)
10000	0.009	0.000	0.010	0.000	random.py:173(randrange)
10000	0.004	0.000	0.014	0.000	random.py:237(randint)
1	0.001	0.001	0.004	0.004	random.py:40(<module>)
1	0.000	0.000	0.000	0.000	random.py:649(WichmannHill)
1	0.000	0.000	0.000	0.000	random.py:72(Random)
1	0.000	0.000	0.000	0.000	random.py:799(SystemRandom)
1	0.000	0.000	0.000	0.000	random.py:91( <code>__init__</code> )
1	0.000	0.000	0.000	0.000	{hashlib.openssl_md5}
1	0.000	0.000	0.000	0.000	{hashlib.openssl_shal}
1	0.000	0.000	0.000	0.000	{hashlib.openssl_sha224}
1	0.000	0.000	0.000	0.000	{hashlib.openssl_sha256}
1	0.000	0.000	0.000	0.000	{hashlib.openssl_sha384}
1	0.000	0.000	0.000	0.000	{hashlib.openssl_sha512}
1	0.000	0.000	0.000	0.000	{binascii.hexlify}
1	0.000	0.000	0.000	0.000	{function seed at 0x7f2055f3df50}
6	0.000	0.000	0.000	0.000	{getattr}
6	0.000	0.000	0.000	0.000	{globals}
1	0.000	0.000	0.000	0.000	{len}
1	0.000	0.000	0.000	0.000	{math.exp}
2	0.000	0.000	0.000	0.000	{math.log}
1	0.000	0.000	0.000	0.000	{math.sqrt}
10000	0.001	0.000	0.001	0.000	{method 'append' of 'list' objects}
1	0.000	0.000	0.000	0.000	{method 'disable' of 'lsprof.Profiler' objects}
10000	0.001	0.000	0.001	0.000	{method 'random' of '_random.Random' objects}
1	0.000	0.000	0.000	0.000	{method 'union' of 'set' objects}
1	0.000	0.000	0.000	0.000	{posix.urandom}

[AndMolLop 06/12/16 Escritorio]#

Ilustración 33. Ejecución cProfiler sobre el programa de ordenación burbuja

Como vemos en la *Ilustración 33*, lo primero que nos dice cProfile es el número de llamadas a funciones que se han realizado en el tiempo de ejecución del programa (diciéndonos también cuanto ha sido el tiempo de ejecución en segundos).

Lo siguiente que nos muestra es una serie de columnas que nos dicen la siguiente información:

Ncalls: nos dice el número de veces que se ha llamado a la función.

Tottime: es el tiempo total empleado en la función, excluyendo el tiempo que se haya podido gastar en posibles subfunciones que llame internamente.

Percall: es tottime dividido ncalls.

Cumtime: es el tiempo acumulado empleado en la función, contando el tiempo empleando en las subfunciones que tenga.

Percall: es cumtime dividido ncalls.

Filename:lineno(function): nos dice el nombre del archivo, la línea y el nombre de la función que se está llamando.

Una vez sabemos que significa cada variable, podemos ver que el programa se ha pasado más tiempo en la función de comparación, seguido muy de cerca de la función de incrementar, aunque la función de comparación es llamada 20000 veces menos. También podemos observar que lo siguiente que más tiempo ha gastado ha sido la función de intercambiar valores, pero podemos ver que esta ha sido llamada menos de la mitad de las veces que la de comparación, por lo que podemos asumir que el vector generado no requería de mucha ordenación.

En la *Ilustración 33* he marcado la columna de cumtime, porque en principio me parece más interesante, ya que nos dice el tiempo que se está en la función desde que esta se invoca hasta que esta devuelve la llamada, lo que me parece más significativo, pero también veo importante la comparación entre tottime y cumtime, ya que nos puede decir si el problema de eficiencia en tiempo está en la función o en alguna de las subfunciones a las que esta pueda llamar. Aunque en mi programa tottime y cumtime son casi siempre iguales ya que no se hacen llamadas a subfunciones, salvo en la función de aleatorio, ya que esta utiliza funciones propias de la biblioteca random, y como podemos ver, en este caso, está más tiempo en las subfunciones que en la propia función. Las subfunciones también las podemos ver un poco más abajo en la *Ilustración 33*, las cuales son randrange y randint, que se llaman el mismo número de veces que aleatorio, ya que esta las invoca siempre en su ejecución.

[8] ("26.4. The Python Profilers — Python 2.7.13 documentation", 2016)



### **9ª Cuestión: Acceda a la consola de 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 se puede hacer libremente)**

Lo primero que vamos a hacer para ello es acceder a la consola de mysql ejecutando `mysql -u user -p`. Una vez dentro tenemos que crear una base de datos y unas tablas. Para no complicarnos tanto, he encontrado que el propio MySQL tiene unas tablas para hacer pruebas en [https://github.com/datacharmer/test\\_db](https://github.com/datacharmer/test_db), por lo que solo tendremos que descargárnoslas haciendo `git clone https://github.com/datacharmer/test_db.git`, y con esto se nos creará una carpeta llamada `test_db` que tendrá el contenido del repositorio.

Ahora para importarla a MySQL lo único que tenemos que hacer es irnos al directorio `test_db` y dentro de él ejecutar `mysql -u user -p < employees.sql`. Una vez la hemos instalado podemos comprobarlo ejecutando `mysql -p -t < test_employees_md5.sql` y esto nos debería mostrar la información con el nombre de las tablas, número de filas por tabla y crc y comprobar que coinciden.

Tras esto ya si podemos volver a acceder a mysql, y si ejecutamos `show databases;` veremos que hay una tabla que se llama `employees`, ahora ejecutamos `use employees;` para “movernos” a esa base de datos, y después podemos ejecutar `show tables;` para ver que tablas hay en la base de datos.

Una vez que sabemos las tablas que hay y como están formadas, haciendo `describe nombre_tabla;` podemos pensar que una consulta lógica sobre la base de datos puede ser, por ejemplo, mirar cuales son los nombres de los empleados del departamento de recursos humanos y desde que fecha están contratados en él. Antes de comenzar con la consulta activamos el profiler, para ello ejecutamos `set profiling = 1;` ya que por defecto viene desactivado.

Ahora vamos a proceder a hacer la consulta, la cual se puede realizar de dos maneras, una es usando el operador NATURAL JOIN, y la otra es usando su equivalente con `=`. Por lo tanto, vamos a realizar la misma consulta de las dos maneras, para ver con el profiler cual es mejor.

La primera manera de realizar la consulta la vamos a hacer de la siguiente forma:

```
Select first_name, last_name, dept_name, from_date from current_dept_emp
natural join (select dept_no, dept_name from departments where dept_name = "Human
Resources") d natural join (select emp_no, first_name, last_name from employees) e;
```

Esto nos mostrará el resultado deseado. Si ahora ejecutamos `show profiles;` nos dirá la ID que se le ha asociado a la consulta, el tiempo que ha tardado, y la consulta que se ha hecho. Esto lo muestro en la *Ilustración 34*.

```

| Howell      | Petereit    | Human Resources | 1989-08-10 |
| Mitchel    | Rosti       | Human Resources | 1992-10-01 |
| Nathan     | Chinia      | Human Resources | 1995-05-12 |
| Fumiyo     | Delgrande   | Human Resources | 1995-05-26 |
| Deniz      | Lundstrom   | Human Resources | 1999-10-20 |
| Subhash    | Barriga     | Human Resources | 1998-02-15 |
| Heeju      | Mandelberg  | Human Resources | 1999-12-31 |
| Greger     | Jahnichen   | Human Resources | 1989-03-28 |
| Kamman     | Varley      | Human Resources | 1997-11-01 |
| Shaunak    | Uural       | Human Resources | 2000-10-05 |
| Mitsuyuki  | Doering     | Human Resources | 1986-07-03 |
| Yolla      | Auria       | Human Resources | 1986-04-25 |
| Kasidit    | Krohm       | Human Resources | 1999-06-18 |
| Steen      | Keohane     | Human Resources | 1987-02-25 |
| Yunning    | Mitina      | Human Resources | 1991-03-07 |
+-----+-----+-----+-----+
16071 rows in set (1,06 sec)

mysql> show profiles;
+-----+-----+-----+
| Query_ID | Duration | Query |
+-----+-----+-----+
| 1 | 1.07209750 | select first_name, last_name, dept_name, from_date from current_dept_emp n
natural join (select dept_no, dept_name from departments where dept_name = "Human Resources") d natur
al join (select emp_no, first_name, last_name from employees) e |
+-----+-----+-----+
1 row in set, 1 warning (0,00 sec)

mysql>

```

Ilustración 34. Consulta con natural join y visualización de su ID en el profiler

Ahora para la segunda manera de realizar la consulta la realizamos ejecutando:

*Select first\_name, last\_name, dept\_name, from\_date from current\_dept\_emp c,  
(select dept\_no, dept\_name from departments where dept\_name = "Human Resources")  
d, (select emp\_no, first\_name, last\_name from employees) where d.dept\_no = c.dept\_no  
and e.emp\_no = c.emp\_no;*

Al igual que antes esto nos mostrará el resultado deseado (el mismo que antes), y si volvemos a ejecutar *show profiles;* veremos que a esta consulta se le ha asignado también una ID y nos dice su tiempo de ejecución. Esto lo muestro en la *Ilustración 35*.

```

| Fumiyo     | Delgrande   | Human Resources | 1995-05-26 |
| Deniz      | Lundstrom   | Human Resources | 1999-10-20 |
| Subhash    | Barriga     | Human Resources | 1998-02-15 |
| Heeju      | Mandelberg  | Human Resources | 1999-12-31 |
| Greger     | Jahnichen   | Human Resources | 1989-03-28 |
| Kamman     | Varley      | Human Resources | 1997-11-01 |
| Shaunak    | Uural       | Human Resources | 2000-10-05 |
| Mitsuyuki  | Doering     | Human Resources | 1986-07-03 |
| Yolla      | Auria       | Human Resources | 1986-04-25 |
| Kasidit    | Krohm       | Human Resources | 1999-06-18 |
| Steen      | Keohane     | Human Resources | 1987-02-25 |
| Yunning    | Mitina      | Human Resources | 1991-03-07 |
+-----+-----+-----+
16071 rows in set (1,07 sec)

mysql> show profiles;
+-----+-----+-----+
| Query_ID | Duration | Query |
+-----+-----+-----+
| 1 | 1.07209750 | select first_name, last_name, dept_name, from_date from current_dept_emp n
natural join (select dept_no, dept_name from departments where dept_name = "Human Resources") d natur
al join (select emp_no, first_name, last_name from employees) e |
| 2 | 1.07347100 | select first_name, last_name, dept_name, from_date from current_dept_emp c
, (select dept_no, dept_name from departments where dept_name = "Human Resources") d, (select emp_no
, first_name, last_name from employees) e where d.dept_no = c.dept_no and e.emp_no = c.emp_no |
+-----+-----+-----+
2 rows in set, 1 warning (0,00 sec)

mysql> _

```

Ilustración 35. Consulta equivalente sin natural join y visualización de su ID

De momento en la *Ilustración 35* si comparamos los tiempos de las dos consultas podemos que ver que la diferencia es casi nula. De modo que vamos a ver, por ejemplo, el uso de CPU de ambas para ver si hay mayor diferencia.

Por cuestiones de espacio en la pantalla, voy a hacer que la salida de cada visualización del profiler se guarde en un archivo distinto. Para ello ejecutamos `\T /home/nombre_usuario/profiler_natural_join.out`, y una vez que hemos dicho el archivo en el que queremos que se guarde, ejecutamos *show profile cpu for query 1*;

Ahora para la segunda consulta ejecutamos lo mismo, pero cambiando el archivo `\T /home/nombre_usuario/profiler_equivalente.out`, y una vez cambiado ejecutamos *show profile cpu for query 2*;

Con esto las salidas del profiler se habrán almacenado en los archivos indicado, como muestro en la *Ilustración 36*, y ahora podremos revisarlos más detenidamente.

```
AndMolLop 07/12/16:~$ ls
allowrootssh.sh  copia.sh  Escritorio  Música  Plantillas  Público  Videos
changeport.py   Descargas fabfile.pyc numeros.txt profiler_equivalente.out  seguridad
codigo          Documentos Imágenes   PDFs    profiler_natural_join.out test_db
AndMolLop 07/12/16:~$ cat profiler_natural_join.out | head -n 10
mysql> show profile cpu for query 1;
+-----+-----+-----+-----+
| Status          | Duration | CPU_user | CPU_system |
+-----+-----+-----+-----+
| starting        | 0.000102 | 0.000000 | 0.000000   |
| checking permissions | 0.000014 | 0.000000 | 0.000000   |
| checking permissions | 0.000001 | 0.000000 | 0.000000   |
| checking permissions | 0.000003 | 0.000000 | 0.000000   |
| Opening tables  | 0.000040 | 0.004000 | 0.000000   |
| checking permissions | 0.000002 | 0.000000 | 0.000000   |
AndMolLop 07/12/16:~$ cat profiler_equivalente.out | head -n 10
mysql> show profile cpu for query 2;
+-----+-----+-----+-----+
| Status          | Duration | CPU_user | CPU_system |
+-----+-----+-----+-----+
| starting        | 0.000116 | 0.000000 | 0.000000   |
| checking permissions | 0.000005 | 0.000000 | 0.000000   |
| checking permissions | 0.000001 | 0.000000 | 0.000000   |
| checking permissions | 0.000003 | 0.000000 | 0.000000   |
| Opening tables  | 0.000042 | 0.000000 | 0.000000   |
| checking permissions | 0.000002 | 0.000000 | 0.000000   |
AndMolLop 07/12/16:~$
```

*Ilustración 36. Ficheros de salida y visualización parcial de su contenido*

Vemos que los archivos de salida se han creado correctamente, de modo que ahora podemos analizar su contenido más detalladamente. En la *Ilustración 37* muestro la tabla generada por el profiler para la consulta con natural join. Y en la *Ilustración 38* muestro la tabla de la consulta equivalente sin natural join.

Y, por último, comparando ambas tablas, vemos que la diferencia de uso de CPU entre una consulta y la otra también es nula, por lo que podemos deducir, que en este caso da igual hacer la consulta de una manera o de otra, ya que la carga en el sistema va a ser la misma.

[9] ("MySQL :: MySQL 5.7 Reference Manual", 2016)

Status	Duration	CPU_user	CPU_system
starting	0.000102	0.000000	0.000000
checking permissions	0.000014	0.000000	0.000000
checking permissions	0.000001	0.000000	0.000000
checking permissions	0.000003	0.000000	0.000000
Opening tables	0.000040	0.004000	0.000000
checking permissions	0.000002	0.000000	0.000000
checking permissions	0.000002	0.000000	0.000000
checking permissions	0.000032	0.000000	0.000000
checking permissions	0.000002	0.000000	0.000000
checking permissions	0.000008	0.000000	0.000000
init	0.000031	0.000000	0.000000
checking permissions	0.000002	0.000000	0.000000
checking permissions	0.000021	0.000000	0.000000
checking permissions	0.000002	0.000000	0.000000
checking permissions	0.000019	0.000000	0.000000
System lock	0.000006	0.000000	0.000000
optimizing	0.000002	0.000000	0.000000
optimizing	0.000002	0.000000	0.000000
statistics	0.000012	0.000000	0.000000
preparing	0.000009	0.000000	0.000000
Sorting result	0.000009	0.000000	0.000000
statistics	0.000104	0.000000	0.000000
preparing	0.000009	0.000000	0.000000
executing	0.000006	0.000000	0.000000
Sending data	0.000033	0.000000	0.000000
executing	0.000002	0.000000	0.000000
Sending data	0.487151	0.484000	0.000000
converting HEAP to ondisk	0.431399	0.432000	0.000000
Sending data	0.147627	0.148000	0.000000
end	0.000012	0.000000	0.000000
query end	0.000009	0.000000	0.000000
closing tables	0.000002	0.000000	0.000000
removing tmp table	0.000395	0.000000	0.000000
closing tables	0.000009	0.000000	0.000000
freeing items	0.004992	0.000000	0.000000
cleaning up	0.000033	0.000000	0.000000

Ilustración 37. Profiler de la consulta con natural join

Status	Duration	CPU_user	CPU_system
starting	0.000116	0.000000	0.000000
checking permissions	0.000005	0.000000	0.000000
checking permissions	0.000001	0.000000	0.000000
checking permissions	0.000003	0.000000	0.000000
Opening tables	0.000042	0.000000	0.000000
checking permissions	0.000002	0.000000	0.000000
checking permissions	0.000001	0.000000	0.000000
checking permissions	0.000044	0.000000	0.000000
checking permissions	0.000002	0.000000	0.000000
checking permissions	0.000012	0.000000	0.000000
init	0.000044	0.000000	0.000000
checking permissions	0.000002	0.000000	0.000000
checking permissions	0.000027	0.000000	0.000000
checking permissions	0.000002	0.000000	0.000000
checking permissions	0.000033	0.000000	0.000000
System lock	0.000007	0.000000	0.000000
optimizing	0.000003	0.000000	0.000000
optimizing	0.000003	0.000000	0.000000
statistics	0.000027	0.000000	0.000000
preparing	0.000013	0.000000	0.000000
Sorting result	0.000018	0.000000	0.000000
statistics	0.000251	0.000000	0.000000
preparing	0.000024	0.000000	0.000000
executing	0.000006	0.000000	0.000000
Sending data	0.000040	0.000000	0.000000
executing	0.000001	0.000000	0.000000
Sending data	0.483945	0.484000	0.000000
converting HEAP to ondisk	0.435921	0.436000	0.000000
Sending data	0.145869	0.144000	0.000000
end	0.000011	0.000000	0.000000
query end	0.000009	0.000000	0.000000
closing tables	0.000002	0.000000	0.000000
removing tmp table	0.000355	0.000000	0.000000
closing tables	0.000008	0.000000	0.000000
freeing items	0.006597	0.000000	0.000000
cleaning up	0.000029	0.000000	0.000000

Ilustración 38. Profiler de la consulta equivalente sin natural join

## **Bibliografía:**

- Cuestión 1:

[1] Tipos de archivos log en Ubuntu Server y descripción de su contenido. *Ubuntu Logs [Adminbuntu]*. (2016). *Adminbuntu.com*. Retrieved 1 December 2016, from [http://www.adminbuntu.com/ubuntu\\_logs](http://www.adminbuntu.com/ubuntu_logs)

[2] Funcionamiento de logrotate. *Rotación de ficheros logs.* \*. (2016). *Obasoft.es*. Retrieved 1 December 2016, from [http://www.obasoft.es/CF/SIINF/SIINF\\_05\\_Contenidos/rotacin\\_de\\_ficheros\\_logs\\_.html](http://www.obasoft.es/CF/SIINF/SIINF_05_Contenidos/rotacin_de_ficheros_logs_.html)

- Cuestión 2:

[3] Introducir nuevas tareas a cron. Pérez Esteso, M. (2016). *PROGRAMAR TAREAS EN LINUX USANDO CRONTAB*. *Geekytheory.com*. Retrieved 3 December 2016, from <https://geekytheory.com/programar-tareas-en-linux-usando-crontab/>

- Cuestión 5:

[4] Crear un conjunto de recopiladores de datos desde el Monitor de rendimiento. (2016). *Technet.microsoft.com*. Retrieved 4 December 2016, from [https://technet.microsoft.com/es-es/library/cc722148\(v=ws.11\).aspx](https://technet.microsoft.com/es-es/library/cc722148(v=ws.11).aspx)

- Cuestión 6:

[5] Mediciones de Munin y que significan sus variables. Munin,. (2016). *Munin :: munin-monitoring.org :: demo.munin-monitoring.org*. *Demo.munin-monitoring.org*. Retrieved 5 December 2016, from <http://demo.munin-monitoring.org/munin-monitoring.org/demo.munin-monitoring.org/>

[6] Instalación y configuración de Munin en CentOS 7 *Server monitoring with Munin in centos 7*. (2016). Retrieved from <https://www.youtube.com/watch?v=oHJFZz5h1Y>

- Cuestión 7:

[7] Qué es y cómo usar Strace. *Sysadmin Tips and Tricks - Using strace to Monitor System Calls | SoftLayer Blog*. (2016). *Blog.softlayer.com*. Retrieved 5 December 2016, from [http://blog.softlayer.com/2013/sysadmin-tips-and-tricks-using-strace-to-monitor-system-calls#utm\\_source=twitter&utm\\_medium=social&utm\\_content=beyond-the-command-line-with-strace&utm\\_campaign=blog\\_development-tips-and-tricks](http://blog.softlayer.com/2013/sysadmin-tips-and-tricks-using-strace-to-monitor-system-calls#utm_source=twitter&utm_medium=social&utm_content=beyond-the-command-line-with-strace&utm_campaign=blog_development-tips-and-tricks)

- Cuestión 8:

[8] Los profilers de Python, cómo usarlos y cómo entenderlos. 26.4. *The Python Profilers — Python 2.7.13 documentation*. (2016). *Docs.python.org*. Retrieved 6 December 2016, from <https://docs.python.org/2/library/profile.html>

- Cuestión 9:

[9] Manual de MySQL para ver sus comandos y como usar su profiler MySQL  
:: *MySQL 5.7 Reference Manual*. (2016). *Dev.mysql.com*. Retrieved 7 December  
2016, from <http://dev.mysql.com/doc/refman/5.7/en/>