

Ingeniería de Servidores (2015-2016)
GRADO EN INGENIERÍA INFORMÁTICA
UNIVERSIDAD DE GRANADA

Memoria Práctica 3

Pablo Martínez Ruano

1 de diciembre de 2015

Índice

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

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 (puede usar el comando date).

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.

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

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.

6.-Instale alguno de los monitores comentados arriba en su máquina y pruebe a ejecutarlos (tenga en cuenta que si lo hace en la máquina virtual, los resultados pueden no ser realistas). Alternativamente, busque otros monitores para hardware comerciales o de código abierto para Windows y Linux.

7.-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.

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

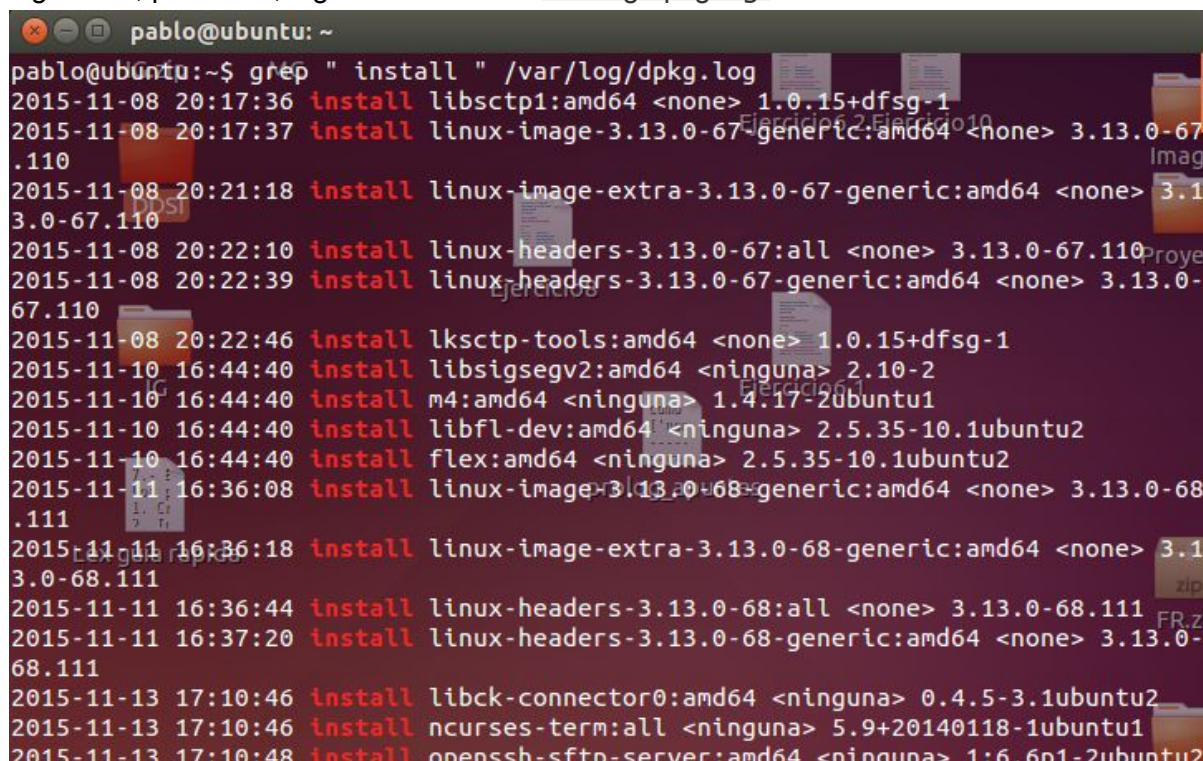
9.- Acceda a la consola mysql (o a través de phpMyAdmin) y muestre el resultado de mostrar el “profile” de una consulta (la creación de la BD y la consulta la puede hacer libremente).

Cuestión opcional 9.- Escriba un script en python y analice su comportamiento usando el profiler presentado.

Cuestión opcional 11: Al igual que ha realizado el “profiling” con MySQL, realice lo mismo con MongoDB y compare los resultados (use la misma información y la misma consulta, hay traductores de consultas SQL a Mongo).

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

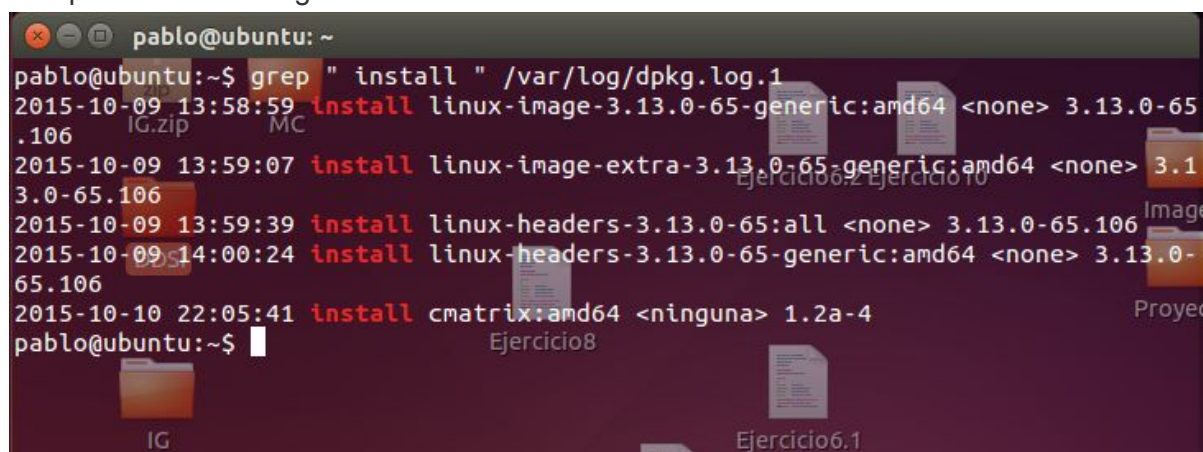
Para poder ver los programas que se han instalado, debemos averiguar el log donde esté registrado, para este, log esta situado en `/var/log/dpkg.log`¹ .



```
pablo@ubuntu: ~  
pablo@ubuntu:~$ grep " install " /var/log/dpkg.log  
2015-11-08 20:17:36 install libscpt1:amd64 <none> 1.0.15+dfsg-1  
2015-11-08 20:17:37 install linux-image-3.13.0-67-generic:amd64 <none> 3.13.0-67.110  
2015-11-08 20:21:18 install linux-image-extra-3.13.0-67-generic:amd64 <none> 3.13.0-67.110  
2015-11-08 20:22:10 install linux-headers-3.13.0-67:all <none> 3.13.0-67.110  
2015-11-08 20:22:39 install linux-headers-3.13.0-67-generic:amd64 <none> 3.13.0-67.110  
2015-11-08 20:22:46 install lksctp-tools:amd64 <none> 1.0.15+dfsg-1  
2015-11-10 16:44:40 install libsigsegv2:amd64 <ninguna> 2.10-2  
2015-11-10 16:44:40 install m4:amd64 <ninguna> 1.4.17-2ubuntu1  
2015-11-10 16:44:40 install libfl-dev:amd64 <ninguna> 2.5.35-10.1ubuntu2  
2015-11-10 16:44:40 install flex:amd64 <ninguna> 2.5.35-10.1ubuntu2  
2015-11-11 16:36:08 install linux-image-3.13.0-68-generic:amd64 <none> 3.13.0-68.111  
2015-11-11 16:36:18 install linux-image-extra-3.13.0-68-generic:amd64 <none> 3.13.0-68.111  
2015-11-11 16:36:44 install linux-headers-3.13.0-68:all <none> 3.13.0-68.111  
2015-11-11 16:37:20 install linux-headers-3.13.0-68-generic:amd64 <none> 3.13.0-68.111  
2015-11-13 17:10:46 install libck-connector0:amd64 <ninguna> 0.4.5-3.1ubuntu2  
2015-11-13 17:10:46 install ncurses-term:all <ninguna> 5.9+20140118-1ubuntu1  
2015-11-13 17:10:48 install openssh-sftp-server:amd64 <ninguna> 1:6.6p1-2ubuntu2
```

Figura 1.1: Ejecución del comando `grep "install" /var/log/dpkg.log`

La terminación .1.gz² no aparece en la documentación, pero si aparece la terminación ".1", esta permite ver los registros anteriores.



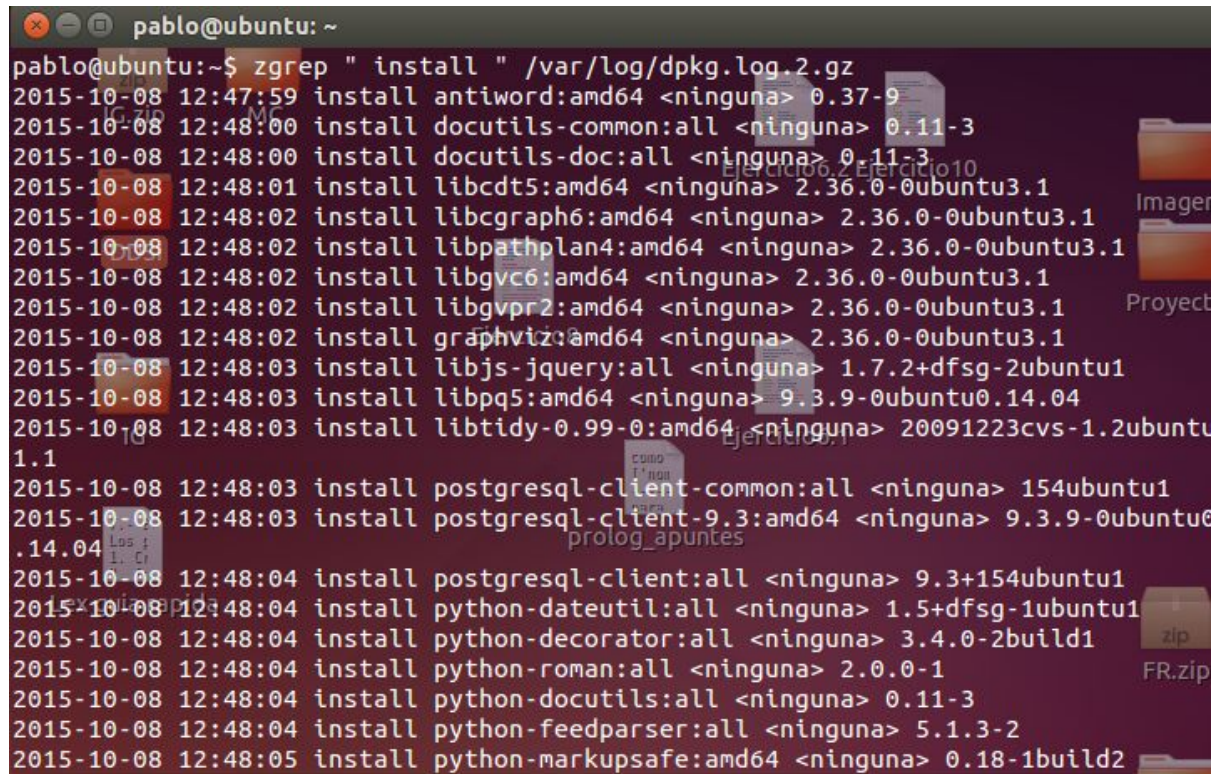
```
pablo@ubuntu: ~  
pablo@ubuntu:~$ grep " install " /var/log/dpkg.log.1  
2015-10-09 13:58:59 install linux-image-3.13.0-65-generic:amd64 <none> 3.13.0-65.106  
2015-10-09 13:59:07 install linux-image-extra-3.13.0-65-generic:amd64 <none> 3.13.0-65.106  
2015-10-09 13:59:39 install linux-headers-3.13.0-65:all <none> 3.13.0-65.106  
2015-10-09 14:00:24 install linux-headers-3.13.0-65-generic:amd64 <none> 3.13.0-65.106  
2015-10-10 22:05:41 install cmatrix:amd64 <ninguna> 1.2a-4  
pablo@ubuntu:~$
```

¹ "ListInstalledPackagesByDate - Community Help Wiki." 2012.
<<https://help.ubuntu.com/community/ListInstalledPackagesByDate>>

² "ListInstalledPackagesByDate - Community Help Wiki." 2012.
<<https://help.ubuntu.com/community/ListInstalledPackagesByDate>>

Figura 1.2: Ejecución del comando grep "install" /var/log/dpkg.log.1

La terminación .gz³ se encuentran los registros archivados.



```
pablo@ubuntu: ~  
pablo@ubuntu:~$ zgrep " install " /var/log/dpkg.log.2.gz  
2015-10-08 12:47:59 install antiword:amd64 <ninguna> 0.37-9  
2015-10-08 12:48:00 install docutils-common:all <ninguna> 0.11-3  
2015-10-08 12:48:00 install docutils-doc:all <ninguna> 0.11-3  
2015-10-08 12:48:01 install libc6:amd64 <ninguna> 2.36.0-0ubuntu3.1  
2015-10-08 12:48:02 install libcgraph6:amd64 <ninguna> 2.36.0-0ubuntu3.1  
2015-10-08 12:48:02 install libpathplan4:amd64 <ninguna> 2.36.0-0ubuntu3.1  
2015-10-08 12:48:02 install libgvc6:amd64 <ninguna> 2.36.0-0ubuntu3.1  
2015-10-08 12:48:02 install libgvpr2:amd64 <ninguna> 2.36.0-0ubuntu3.1  
2015-10-08 12:48:02 install graphviz:amd64 <ninguna> 2.36.0-0ubuntu3.1  
2015-10-08 12:48:03 install libjs-jquery:all <ninguna> 1.7.2+dfsg-2ubuntu1  
2015-10-08 12:48:03 install libpq5:amd64 <ninguna> 9.3.9-0ubuntu0.14.04  
2015-10-08 12:48:03 install libtidy-0.99-0:amd64 <ninguna> 20091223cvs-1.2ubuntu1.1  
2015-10-08 12:48:03 install postgresql-client-common:all <ninguna> 154ubuntu1  
2015-10-08 12:48:03 install postgresql-client-9.3:amd64 <ninguna> 9.3.9-0ubuntu0.14.04  
2015-10-08 12:48:04 install postgresql-client:all <ninguna> 9.3+154ubuntu1  
2015-10-08 12:48:04 install python-dateutil:all <ninguna> 1.5+dfsg-1ubuntu1  
2015-10-08 12:48:04 install python-decorator:all <ninguna> 3.4.0-2build1  
2015-10-08 12:48:04 install python-roman:all <ninguna> 2.0.0-1  
2015-10-08 12:48:04 install python-docutils:all <ninguna> 0.11-3  
2015-10-08 12:48:04 install python-feedparser:all <ninguna> 5.1.3-2  
2015-10-08 12:48:05 install python-markupsafe:amd64 <ninguna> 0.18-1build2
```

Figura 1.3: Ejecución del comando zgrep "install" /var/log/dpkg.log.2.gz

El encargado de asignar estos números es logrotate⁴, su importancia es vital ya que para evitar que la partición de log se llene poco a poco los va comprimiendo y les va asignando número para ahorrar espacio.

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 (puede usar el comando date).

Para programar una tarea hay que modificar el fichero /etc/crontab⁵

Para el ejercicio hemos preparado un pequeño script (llamado script.sh):

```
#!/bin/bash  
date
```

³ "ListInstalledPackagesByDate - Community Help Wiki." 2012.

<<https://help.ubuntu.com/community/ListInstalledPackagesByDate>>

⁴ "logrotate - LinuxCommand.org." 2006.<http://www.linuxcommand.org/man_pages/logrotate8.html>

⁵ "CronHowto - Community Help Wiki - Official Ubuntu ..." 2006. 30 Nov. 2015

<<https://help.ubuntu.com/community/CronHowto>>

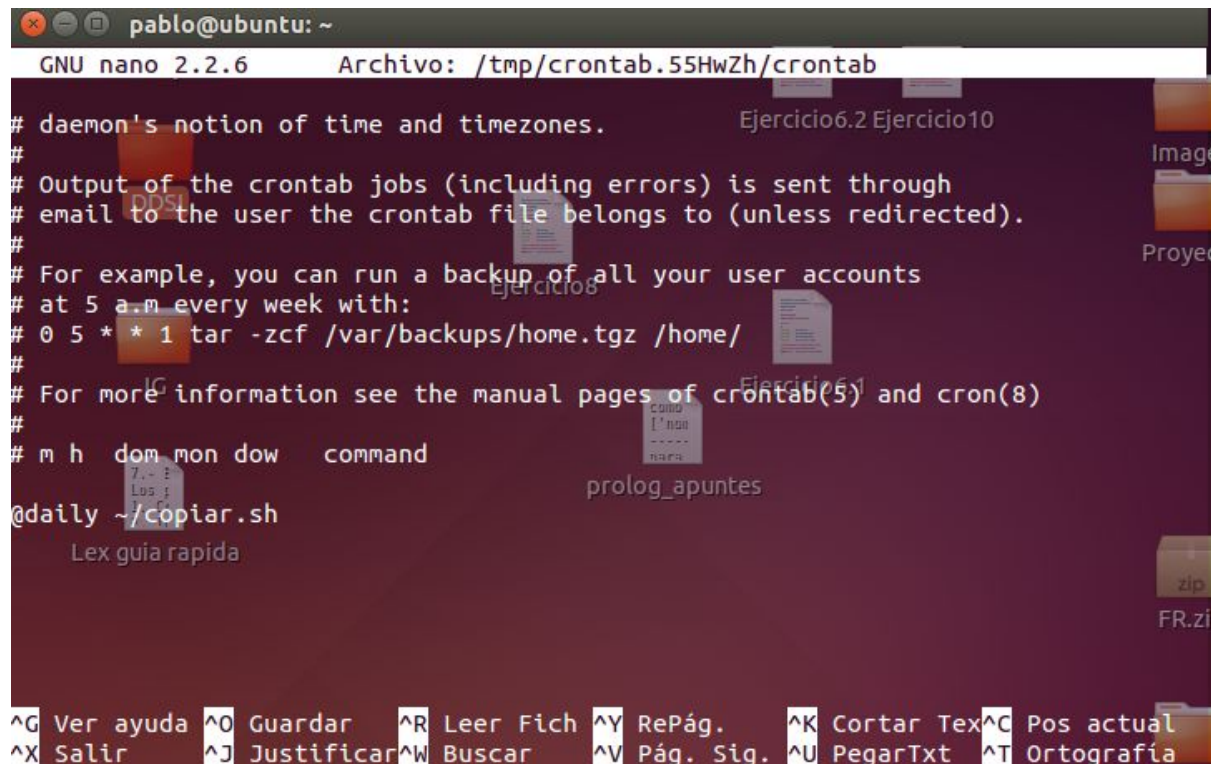
fecha = \$?⁶

mkdir ~/seguridad/\$fecha

cp ~/codigo ~/seguridad/\$fecha

Una vez realizado esto tenemos que darle permiso de ejecución `chmod a+x copiar.sh`

Creamos un archivo cron con el comando `crontab -e`



```
GNU nano 2.2.6 Archivo: /tmp/crontab.55HwZh/crontab
# 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
@daily ~/copiar.sh
```

Figura 2.1: Creando un cron del script.

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.

He conectado un pendrive al usb y ejecutamos `dmesg | tail -27` y nos muestra:

⁶ Este script está inspirado de las prácticas de la asignatura FS, lo menciono porque dijo que un compañero tenía algo parecido, aunque lo comentamos en clase posiblemente se le olvide y me valore negativamente la práctica.


```
pablo@ubuntu: ~  
[ 404.584621] usb 1-1: new high-speed USB device number 3 using ehci-pci  
[ 404.798244] usb 1-1: New USB device found, idVendor=058f, idProduct=6387  
[ 404.798250] usb 1-1: New USB device strings: Mfr=1, Product=2, SerialNumber=3  
[ 404.798253] usb 1-1: Product: Mass Storage  
[ 404.798255] usb 1-1: Manufacturer: Generic  
[ 404.798257] usb 1-1: SerialNumber: 1DEDB2CE  
[ 404.799877] usb-storage 1-1:1.0: USB Mass Storage device detected  
[ 404.799963] scsi34 : usb-storage 1-1:1.0  
[ 405.800849] scsi 34:0:0:0: Direct-Access    USB2.0   Flash Disk           8.07 PB  
Q: 0 ANSI: 2  
[ 405.801694] sd 34:0:0:0: Attached scsi generic sg3 type 0  
[ 405.812174] sd 34:0:0:0: [sdb] 3891200 512-byte logical blocks: (1.99 GB/1.85  
GiB)  
[ 405.816954] sd 34:0:0:0: [sdb] Write Protect is off  
[ 405.816963] sd 34:0:0:0: [sdb] Mode Sense: 03 00 00 00  
[ 405.821014] sd 34:0:0:0: [sdb] No Caching mode page found  
[ 405.821026] sd 34:0:0:0: [sdb] Assuming drive cache: write through  
[ 405.837936] sd 34:0:0:0: [sdb] No Caching mode page found  
[ 405.837947] sd 34:0:0:0: [sdb] Assuming drive cache: write through  
[ 405.935853]   sdb: sdb1  
[ 405.952283] sd 34:0:0:0: [sdb] No Caching mode page found  
[ 405.952294] sd 34:0:0:0: [sdb] Assuming drive cache: write through  
[ 405.952303] sd 34:0:0:0: [sdb] Attached SCSI removable disk  
pablo@ubuntu:~$
```

Figura 3.1: Ejecución de `dmesg | tail -27` al insertar usb

Nos da información básica de nuestro pendrive, como se puede ver nos dice que es de almacenamiento, nos dice el id del producto, nos indica que se ha conectado en un puerto USB 2.0 nos dice su capacidad total y la capacidad usada.

Ahora quitamos el USB:

```
pablo@ubuntu: ~  
[ 404.798250] usb 1-1: New USB device strings: Mfr=1, Product=2, SerialNumber=3  
[ 404.798253] usb 1-1: Product: Mass Storage  
[ 404.798255] usb 1-1: Manufacturer: Generic  
[ 404.798257] usb 1-1: SerialNumber: 1DEDB2CE  
[ 404.799877] usb-storage 1-1:1.0: USB Mass Storage device detected  
[ 404.799963] scsi34 : usb-storage 1-1:1.0  
[ 405.800849] scsi 34:0:0:0: Direct-Access    USB2.0   Flash Disk           8.07 P  
Q: 0 ANSI: 2  
[ 405.801694] sd 34:0:0:0: Attached scsi generic sg3 type 0  
[ 405.812174] sd 34:0:0:0: [sdb] 3891200 512-byte logical blocks: (1.99 GB/1.85  
GiB)  
[ 405.816954] sd 34:0:0:0: [sdb] Write Protect is off  
[ 405.816963] sd 34:0:0:0: [sdb] Mode Sense: 03 00 00 00  
[ 405.821014] sd 34:0:0:0: [sdb] No Caching mode page found  
[ 405.821026] sd 34:0:0:0: [sdb] Assuming drive cache: write through  
[ 405.837936] sd 34:0:0:0: [sdb] No Caching mode page found  
[ 405.837947] sd 34:0:0:0: [sdb] Assuming drive cache: write through  
[ 405.935853]   sdb: sdb1  
[ 405.952283] sd 34:0:0:0: [sdb] No Caching mode page found  
[ 405.952294] sd 34:0:0:0: [sdb] Assuming drive cache: write through  
[ 405.952303] sd 34:0:0:0: [sdb] Attached SCSI removable disk  
[ 2909.358403] [sched_delayed] sched: RT throttling activated  
[ 3721.951617] usb 1-1: USB disconnect, device number 3  
pablo@ubuntu:~$
```

Figura 3.2: Ejecución de `dmesg | tail -27` al sacar usb.

Ahora solo nos indica que el USB ha sido desconectado.

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

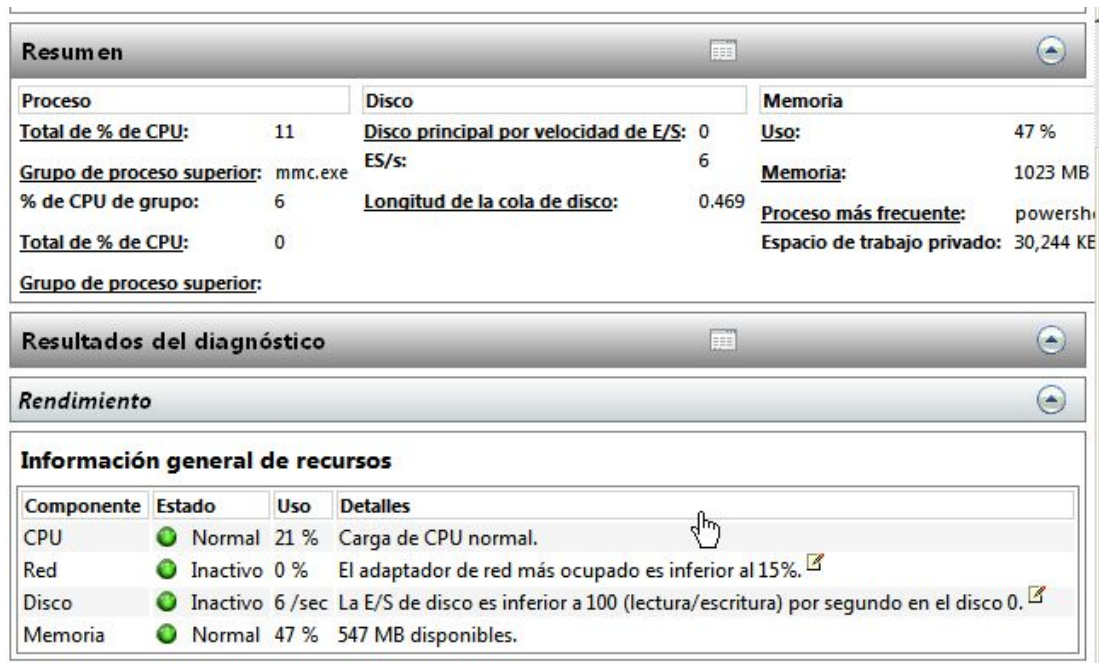


Figura 4.1: Resumen del “System performance”

Podemos observar que la carga de la CPU es normal, solo consume lo propio del sistema operativo y varias aplicaciones en segundo plano. La red está inactiva debido a que no estamos haciendo uso de ella nosotros ni ningún software, el disco duro está igual y la memoria pese a que aparentemente consumiendo 47% esto es solo la memoria del propio SO y alguna aplicación suave en segundo plano.

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.

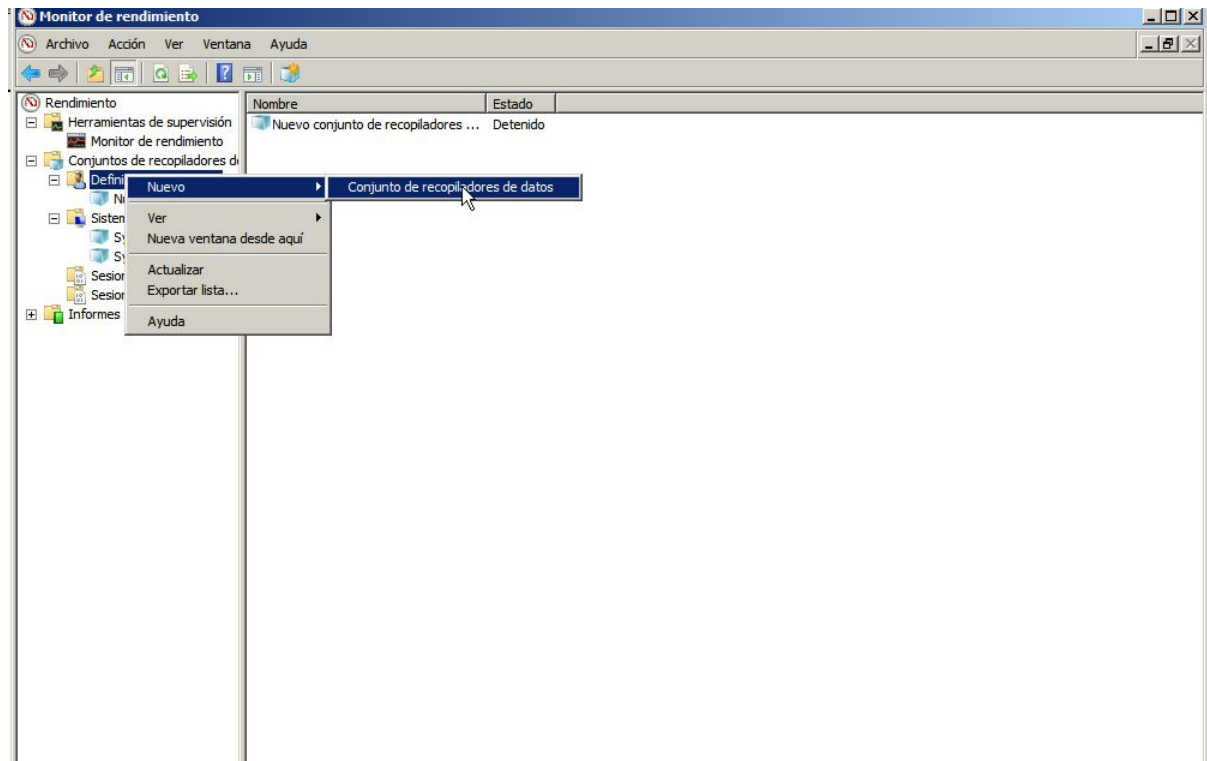


Figura 5.1: Empezamos a crear un conjunto de recopiladores de datos

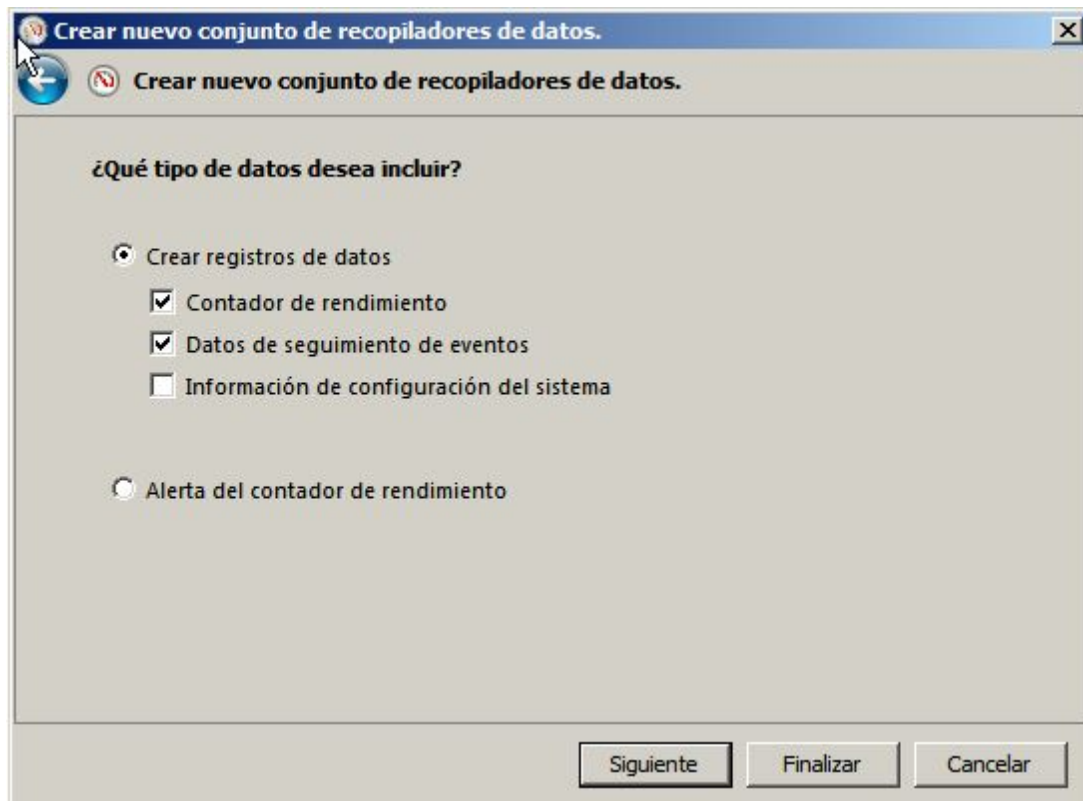


Figura 5.2: Marcamos contador de rendimiento y Datos de seguimiento de eventos.

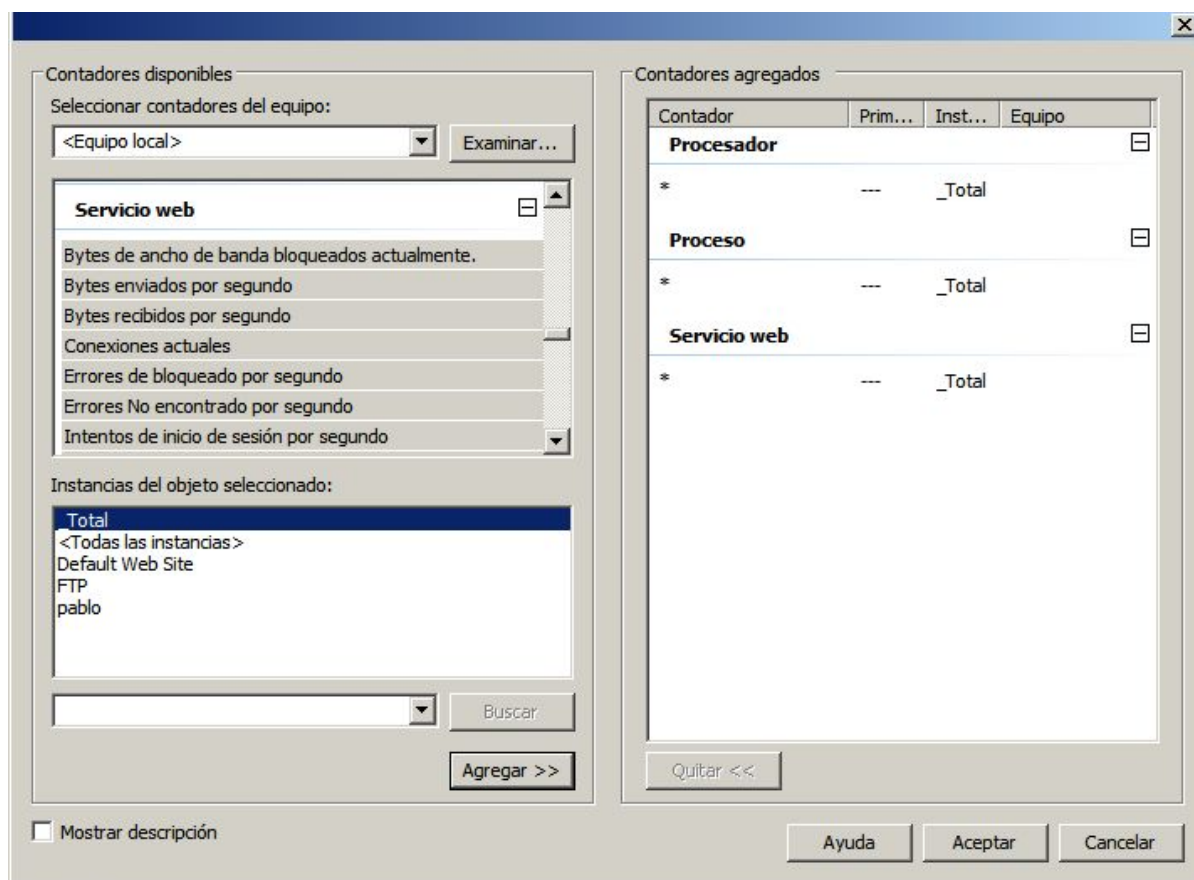


Figura 5.3: Marcamos todo lo relacionado con Procesador, Proceso y Servidor web

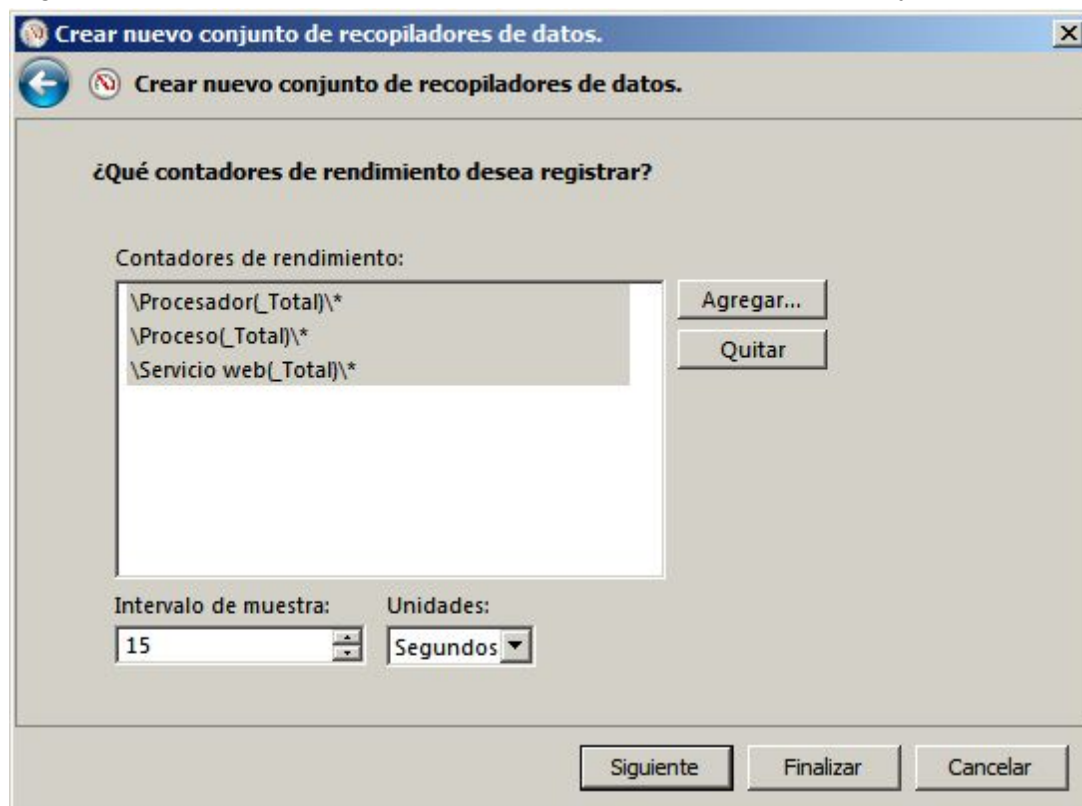


Figura 5.4: Marcamos un intervalo de muestra de 15 segundos.

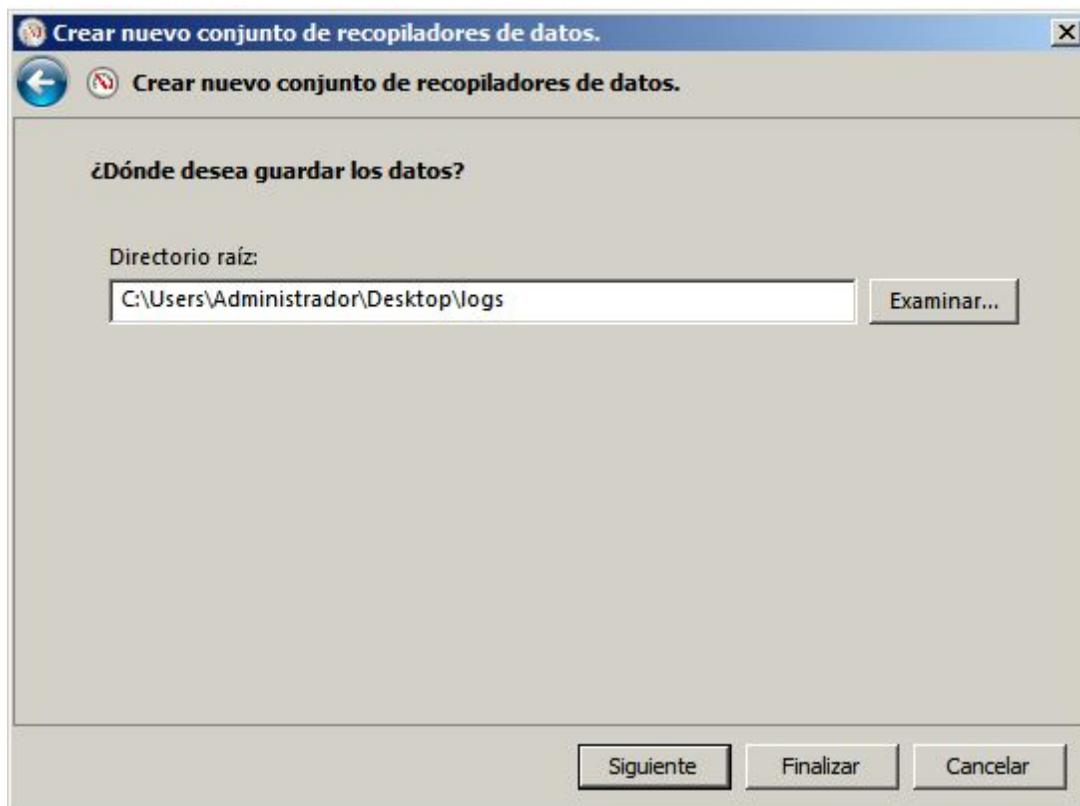


Figura 5.5: Guardamos los datos es Escritorio/logs.

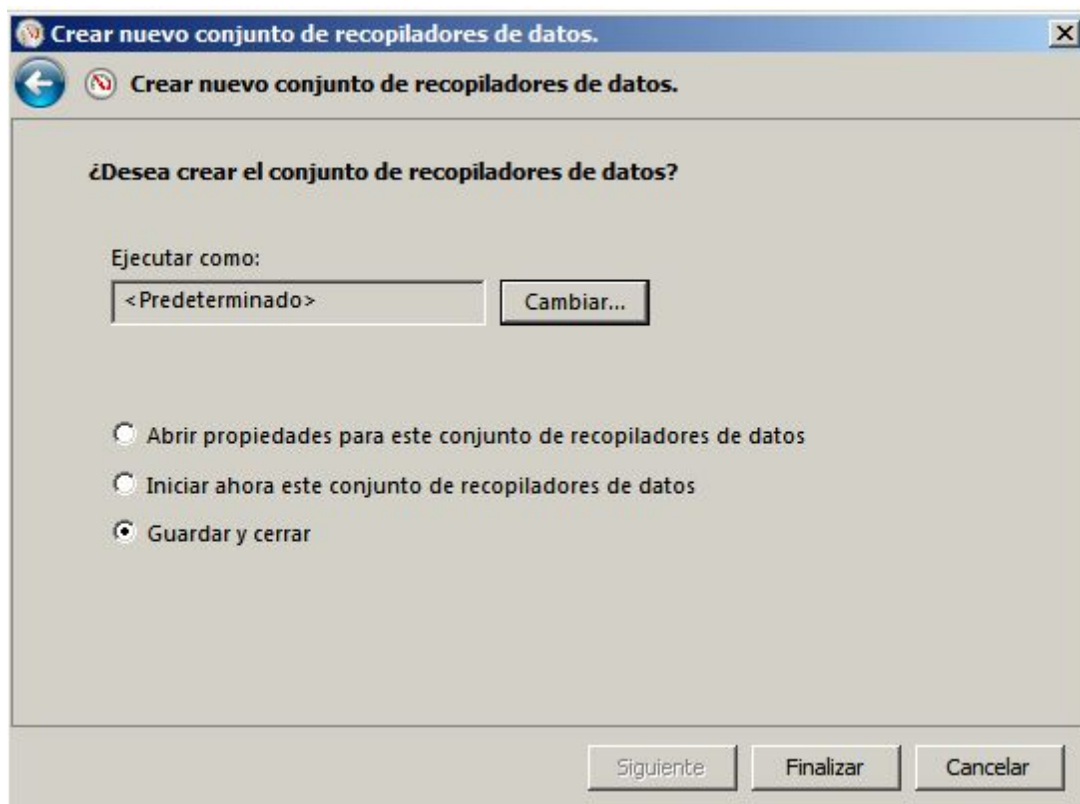


Figura 5.6: Marcamos guardar y cerrar y finalizamos.



Figura 5.7: Observamos los resultados finales.

Viendo el resultado podemos observar la mayoría de los contadores son constantes debido que no le estamos metiendo carga al sistema, sin embargo algunas gráficas hacen cosas “raras”, esto es debido a que estamos en una máquina virtual y los resultados tienden a ser no fiables cuando vemos este tipos de anomalías. Entre estas destaca N° de solicitudes por segundo, que se dispara al igual que el N° de solicitudes Head por segundo.

6.-Instale alguno de los monitores comentados arriba en su máquina y pruebe a ejecutarlos (tenga en cuenta que si lo hace en la máquina virtual, los resultados pueden no ser realistas). Alternativamente, busque otros monitores para hardware comerciales o de código abierto para Windows y Linux.

Hemos instalado xsensors⁷ y los hemos ejecutado, sin embargo el resultado parece ser no realista ya que en nuestro server nos indica que la temperatura de los cores es nula.

⁷ "XSensors — Ubuntu Apps Directory." 2012. 30 Nov. 2015
<<https://apps.ubuntu.com/cat/applications/precise/xsensors/>>

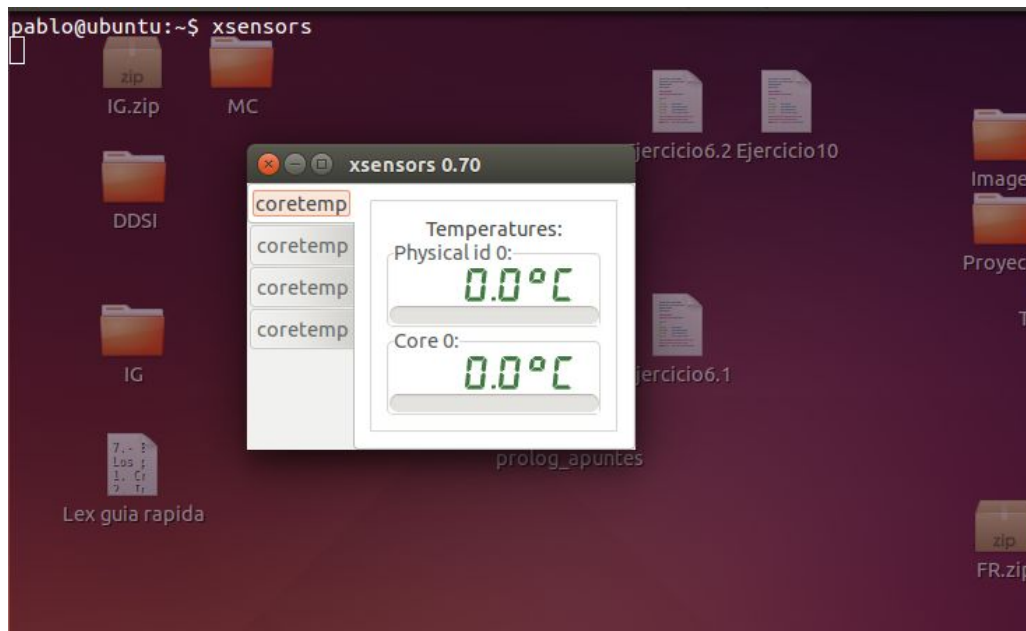


Figura 6.1: Resultado de xsensors aparentemente no muy realista.

Las plataformas Linux tiene diferentes monitores útiles que vienen por defecto y otros hay que instalarlos. Uno de los que viene por defecto es top.

```
top - 20:40:14 up 1:45, 2 users, load average: 0,04, 0,04, 0,05
Tareas: 341 total, 1 ejecutar, 340 hibernar, 0 detener, 0 zombie
%Cpu(s): 0,1 usuario, 0,1 sist, 0,0 adecuado, 99,8 inact, 0,0 en espera, 0,0
KiB Mem: 3075320 total, 1233472 used, 1841848 free, 81704 buffers
KiB Swap: 1045500 total, 0 used, 1045500 free. 527568 cached Mem
```

PID	USUARIO	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	HORA+	ORDEN
1654	root	20	0	266580	35884	11564	S	1,3	1,2	0:19.21	Xorg
2783	pablo	20	0	577744	21100	13964	S	0,7	0,7	0:04.38	gnome-termi+
187	root	20	0	0	0	0	S	0,3	0,0	0:04.61	kworker/0:1
1370	root	20	0	165500	4612	3700	S	0,3	0,1	0:19.33	vmtoolsd
2524	pablo	20	0	331372	22516	17080	S	0,3	0,7	0:23.45	vmtoolsd
4444	pablo	20	0	29376	1944	1252	R	0,3	0,1	0:00.07	top
1	root	20	0	33904	3220	1476	S	0,0	0,1	0:02.53	init
2	root	20	0	0	0	0	S	0,0	0,0	0:00.02	kthreadd
3	root	20	0	0	0	0	S	0,0	0,0	0:00.07	ksoftirqd/0
4	root	20	0	0	0	0	S	0,0	0,0	0:00.00	kworker/0:0
5	root	0	-20	0	0	0	S	0,0	0,0	0:00.00	kworker/0:0H
7	root	20	0	0	0	0	S	0,0	0,0	0:04.05	rcu_sched
8	root	20	0	0	0	0	S	0,0	0,0	0:00.79	rcuos/0
9	root	20	0	0	0	0	S	0,0	0,0	0:00.61	rcuos/1
10	root	20	0	0	0	0	S	0,0	0,0	0:00.50	rcuos/2
11	root	20	0	0	0	0	S	0,0	0,0	0:00.44	rcuos/3
12	root	20	0	0	0	0	S	0,0	0,0	0:00.00	rcuos/4
13	root	20	0	0	0	0	S	0,0	0,0	0:00.00	rcuos/5
14	root	20	0	0	0	0	S	0,0	0,0	0:00.00	rcuos/6
15	root	20	0	0	0	0	S	0,0	0,0	0:00.00	rcuos/7
16	root	20	0	0	0	0	S	0,0	0,0	0:00.00	rcuos/8
17	root	20	0	0	0	0	S	0,0	0,0	0:00.00	rcuos/9
18	root	20	0	0	0	0	S	0,0	0,0	0:00.00	rcuos/10
19	root	20	0	0	0	0	S	0,0	0,0	0:00.00	rcuos/11

Figura 6.2: Resultado del comando top

Ahora instalamos htop⁸, y lo ejecutamos.

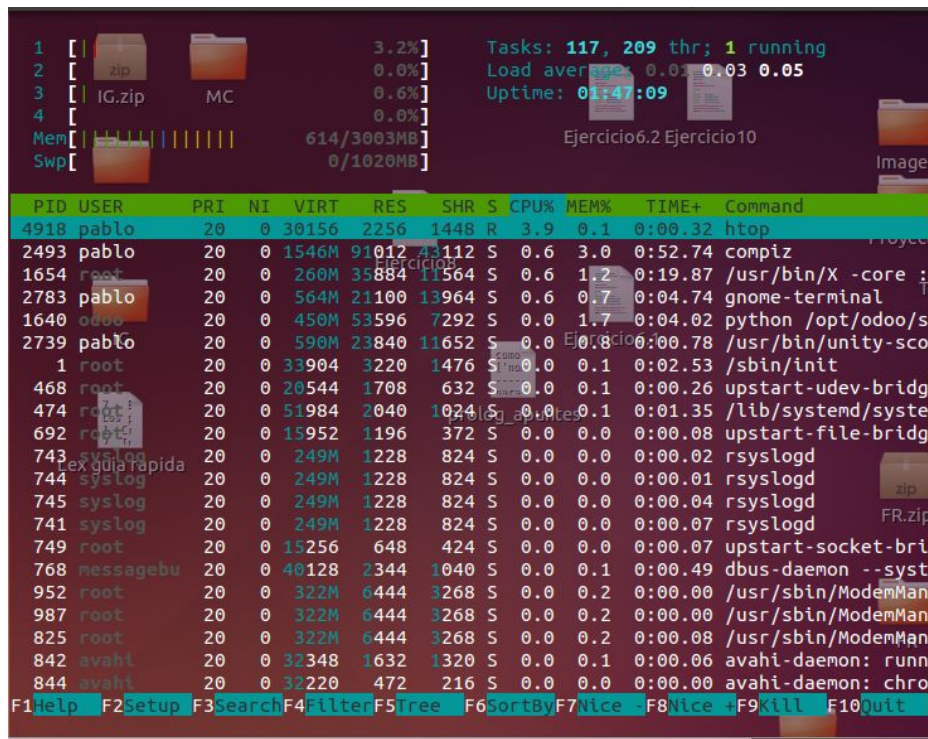


Figura 6.3: Resultado del comando htop

Ahora probamos nmon⁹

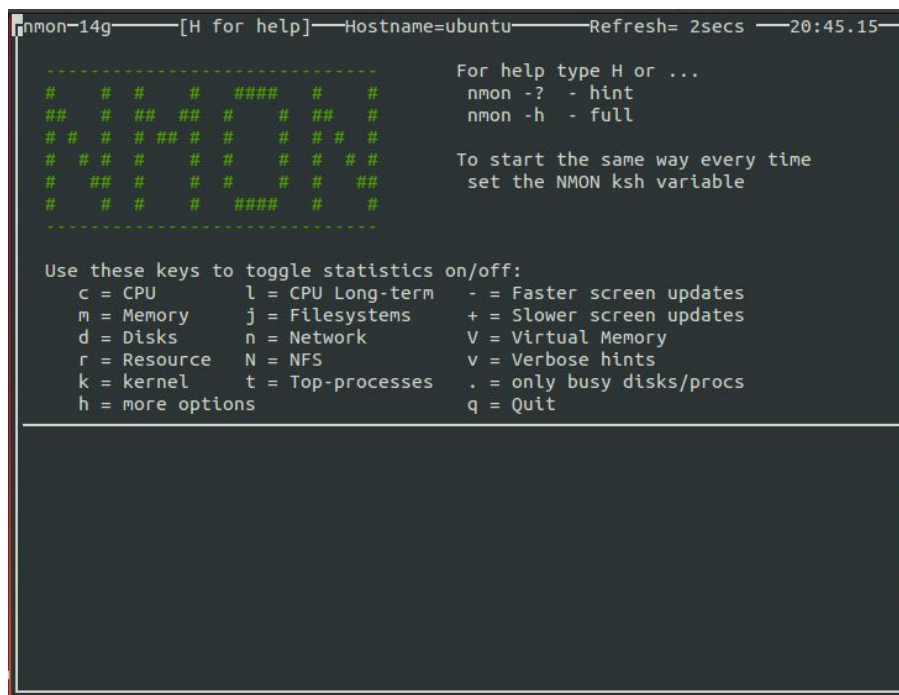


Figura 6.4: Resultado del comando nmon

⁸ "Ubuntu Manpage: htop - interactive process viewer." 2011.

<<http://manpages.ubuntu.com/manpages/precise/man1/htop.1.html>>

⁹ "Not AIX : nmon Manual - IBM." 2013.

<<https://www.ibm.com/developerworks/community/wikis/home/wiki/Not+AIX/page/nmon+Manual>>


```

nmon-14g-----Hostname=ubuntu-----Refresh= 2secs -----20:45.33-----
Memory Stats
Total MB      RAM      High      Low      Swap      Page Size=4 KB
3003.2        3003.2    -0.0      -0.0     1021.0
Free MB       1755.7    -0.0      -0.0     1021.0
Free Percent  58.5%     100.0%    100.0%    100.0%

MB                                     MB                                     MB
      Cached= 516.3      Active= 736.6
Buffers= 80.8 Swapcached= 0.0 Inactive = 336.0
Dirty  = 0.0 Writeback = 0.0 Mapped  = 125.3
Slab   = 84.1 Commit_AS = 3048.1 PageTables= 25.2

```

Figura 6.5: Resultado del comando nmon

Ahora para Windows Server 2008R2 hemos instalado el speedfan 4.51¹⁰ ahora mostramos un ejemplo.

¹⁰ "SpeedFan - Access temperature sensor in your ... - Almico." <<http://www.almico.com/speedfan.php>>

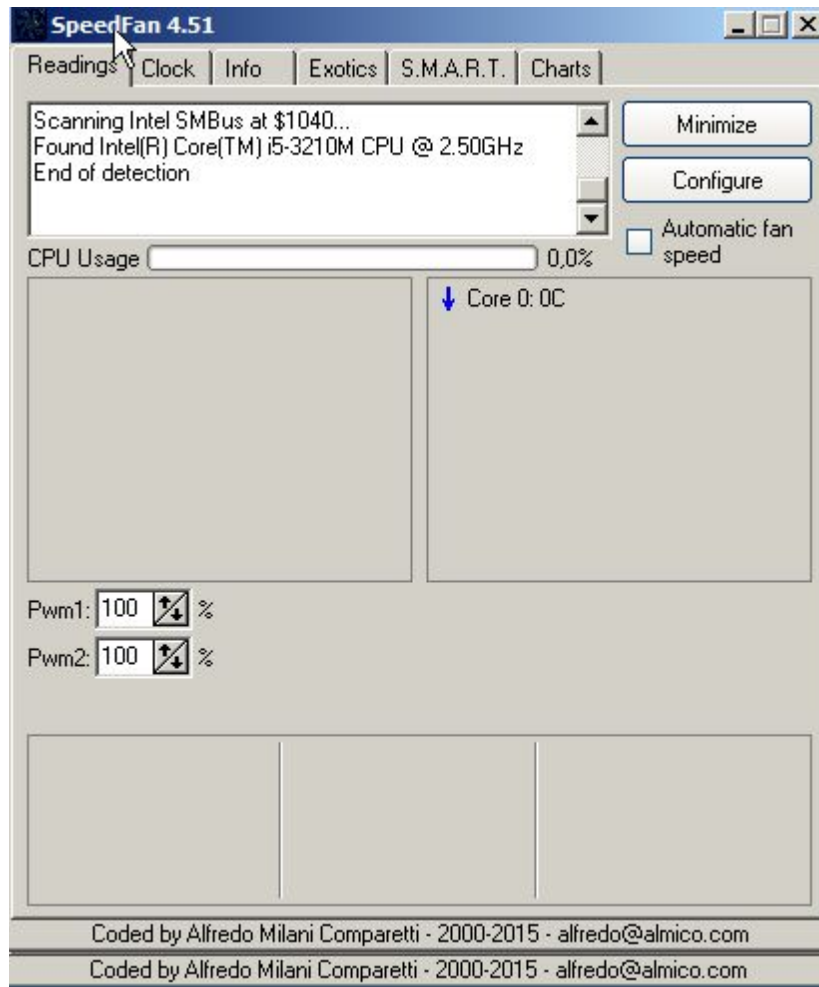


Figura 6.6: Resultado de ejecución del Speedfan.

Al estar en una máquina virtual no permite detectar la temperatura, sin embargo marca el CPU Usage, al momento de hacer captura como salgo del servidor se pone al 0,0%, normalmente ronda entre el 5% o 7%.

7.-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.

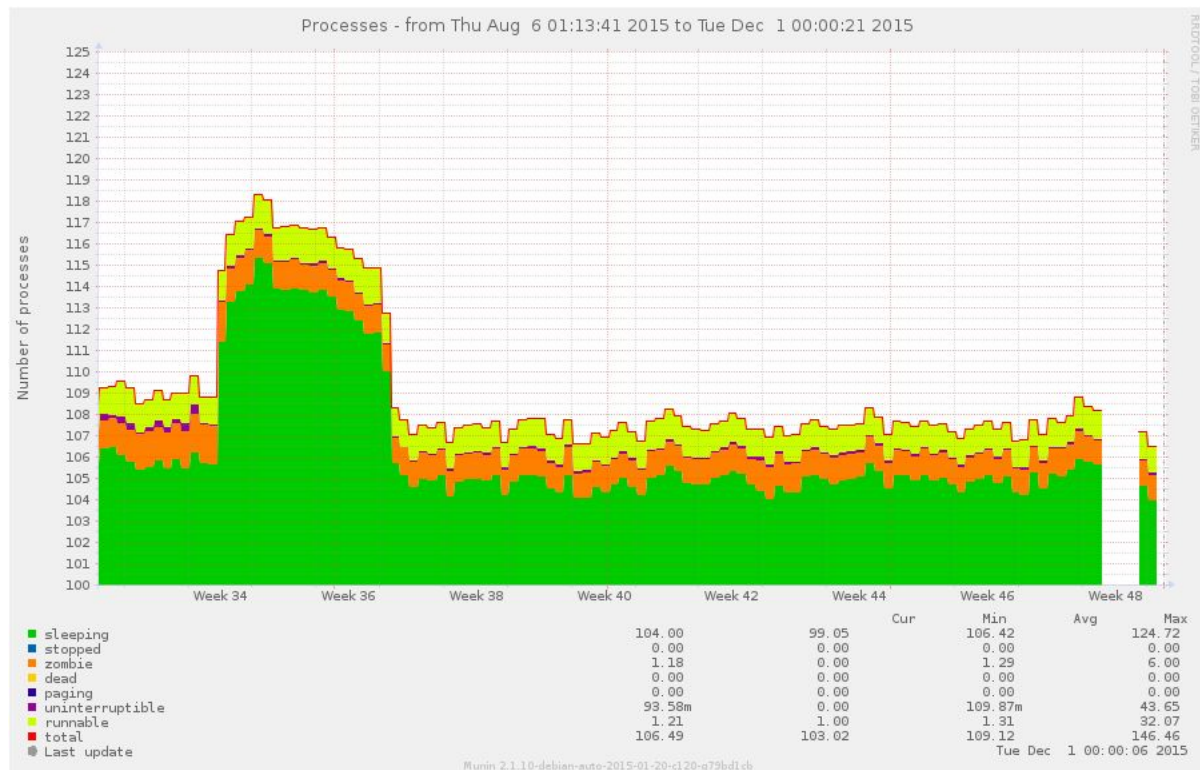


Figura 7.1: Monitorización de los procesos.

En esta gráfica se puede ver el número de procesos realizados por el procesador desde la semana 34 hasta la semana 48, también se puede ver el estado de los procesos. De la gráfica lo que más destaca es que de la semana 34 hasta 36, se puede ver en esta que es cuando más procesos se han ejecutado, por lo que podremos deducir que en esas semanas es cuando el servidor ha sufrido más carga. Por otra parte en la semana 48 la gráfica tiene algún fallo y dejó de marcar, porque luego empieza a marcar como si fuese la continuación y no como si se estuviera reiniciando el servicio.

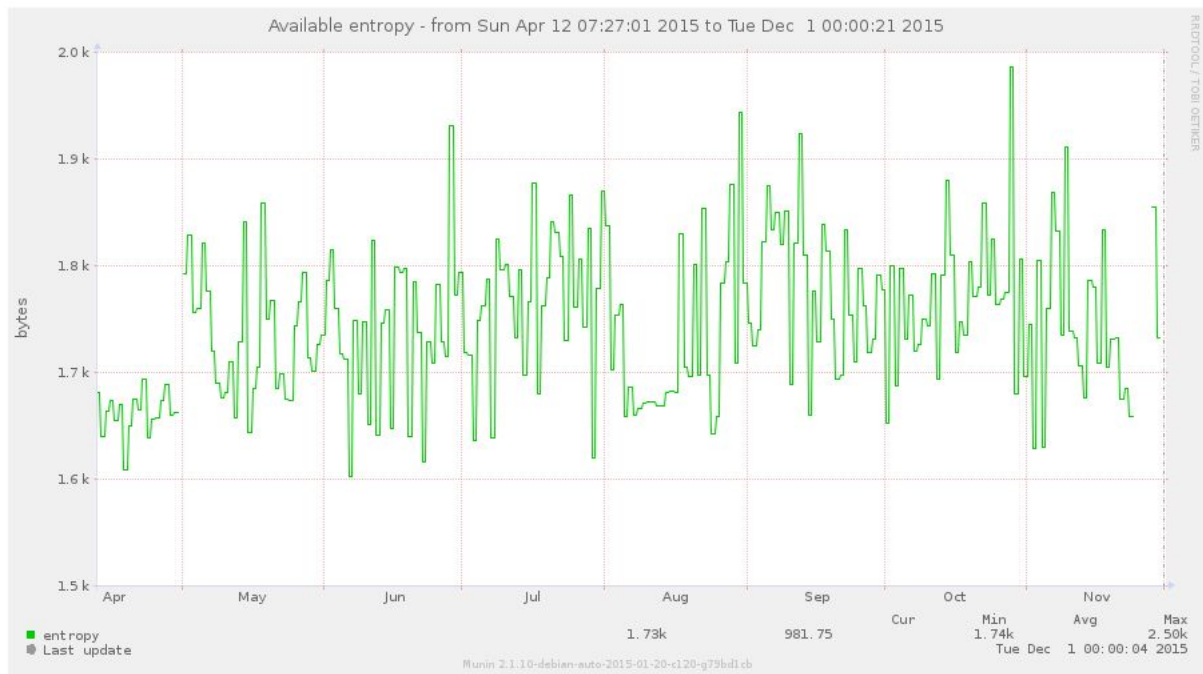


Figura 7.2: Monitorización del nivel de entropía

Aquí se puede ver el nivel de entropía que se produce en el sistema, se puede ver que resulta aceptable por lo que las operaciones de las conexiones SSL/TLS no van a tardar mucho tiempo en completarse. Por otra parte en Noviembre la gráfica tiene algún fallo y dejó de marcar, porque luego empieza a marcar como si fuese la continuación y no como si se estuviera reiniciando el servicio.

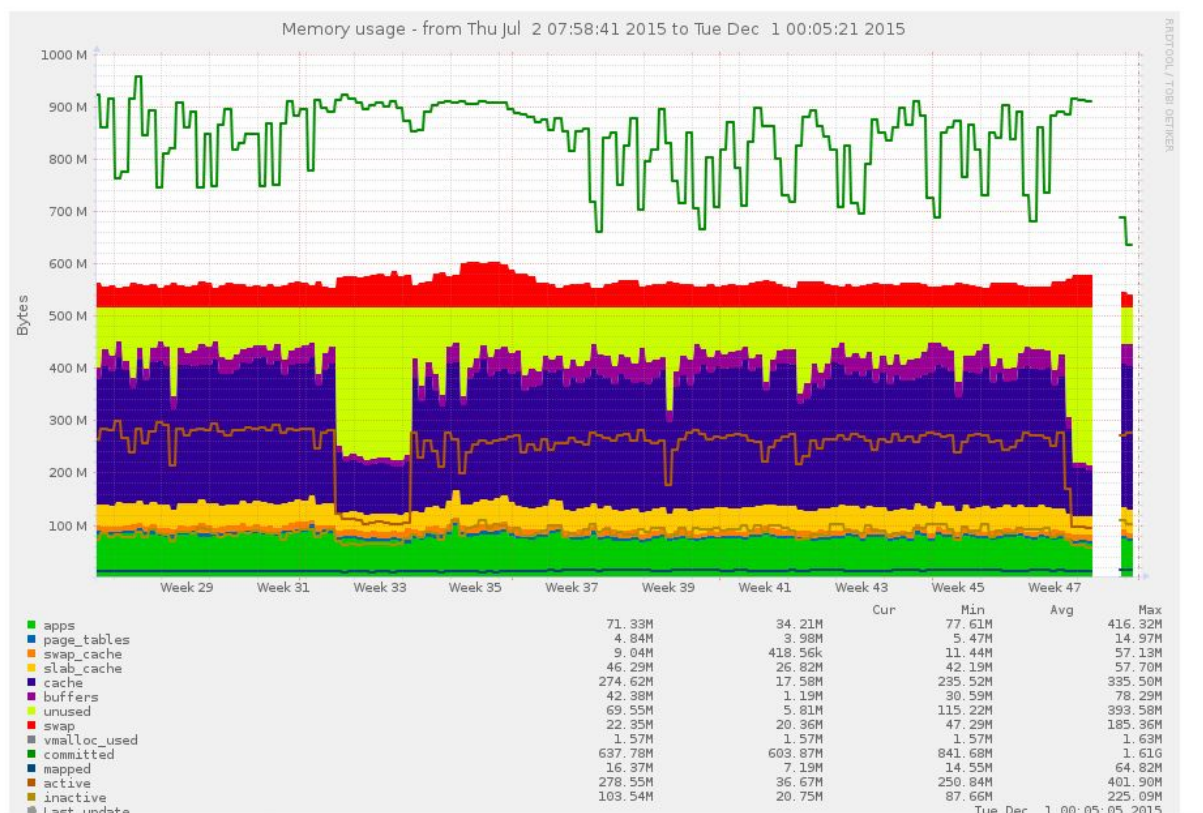


Figura 7.3: Monitorización de la memoria del servidor.

Aquí se puede ver el uso de la memoria que el servidor, aparece detalladamente qué aplicaciones o procesos hacen uso de la memoria. Se puede ver que durante todo el año gasta una cantidad regular de memoria, excepto en la semana 33 que viendo las otras gráficas no hay manera de contrastar que está sucediendo. Por otra parte en la semana 48 la gráfica tiene algún fallo y dejó de marcar, porque luego empieza a marcar como si fuese la continuación y no como si se estuviera reiniciando el servicio.

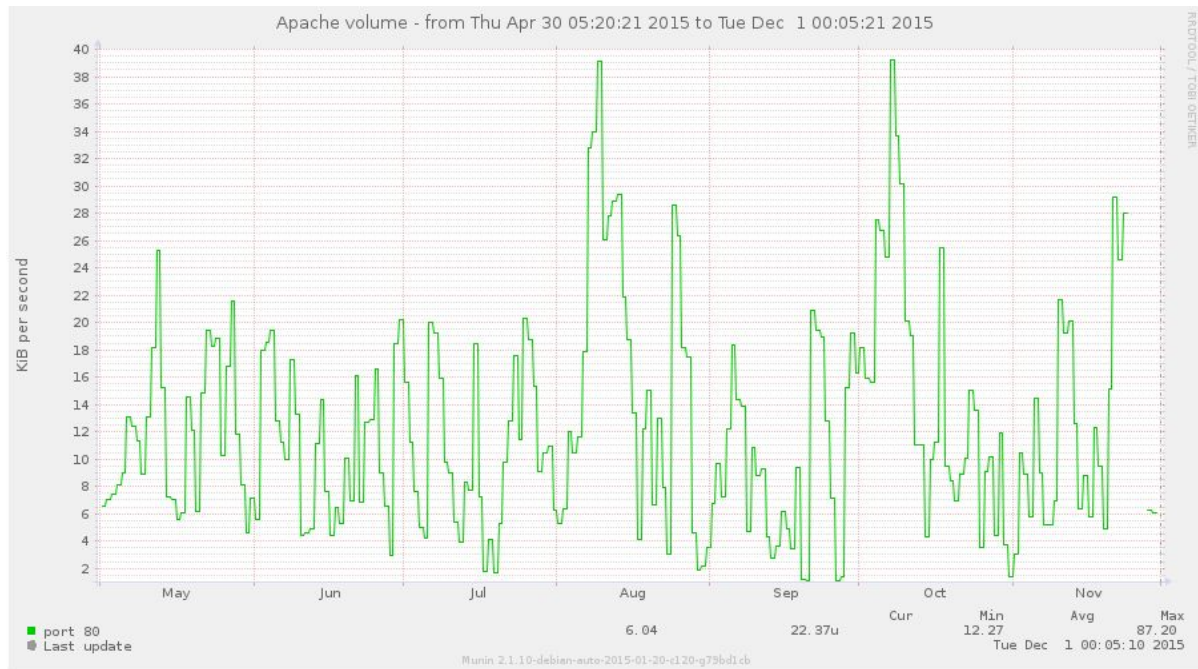


Figura 7.4: Monitorización del Apache

Aquí se puede ver el volumen de datos que se produce en el Apache, en esta gráfica se ve que durante el mes de julio y septiembre ha sido cuando más carga ha tenido el servidor y que a finales de años se vuelve a producir ese vacío del cual hemos hablado en las figuras anteriores.

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

Por lo leído en el segundo texto, strace es una herramienta que nos permite ver que ocurre en situaciones en las que no encontramos errores de depuración aparente, o no tenemos ficheros log. Según el texto no debería tratarse como un debugger ni una herramienta para programadores. Solamente realiza llamadas al sistema para seguir la pista de lo que realiza nuestro programa detalladamente y de esa manera poder ver lo que hace realmente y no lo que nosotros pensamos o deseamos que hiciese.

También puede se puede ver el las llamadas al sistema al ejecutar un comando en el terminal.

Su funcionamiento resulta bastante simple.


```
strace ./miprograma
```

Como en el servidor no tenemos ningún programa para correr vamos a probar con el comando `strace clear` y mostramos varias capturas de lo que muestra.

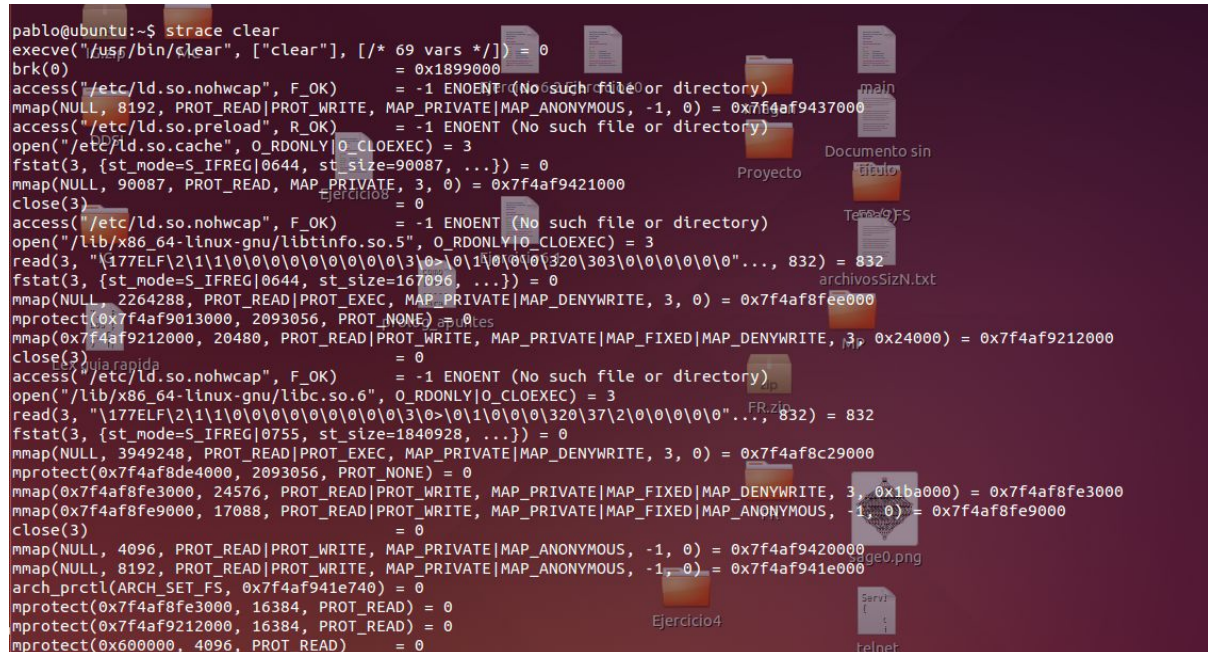


Figura 8.1: Resultado del comando `strace clear` (1)

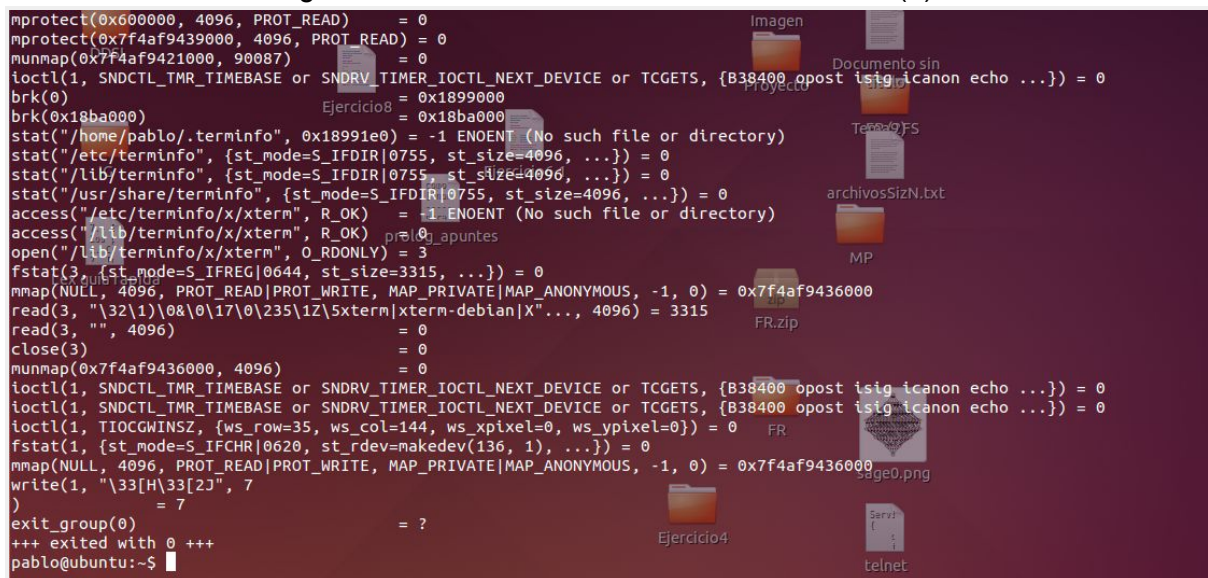
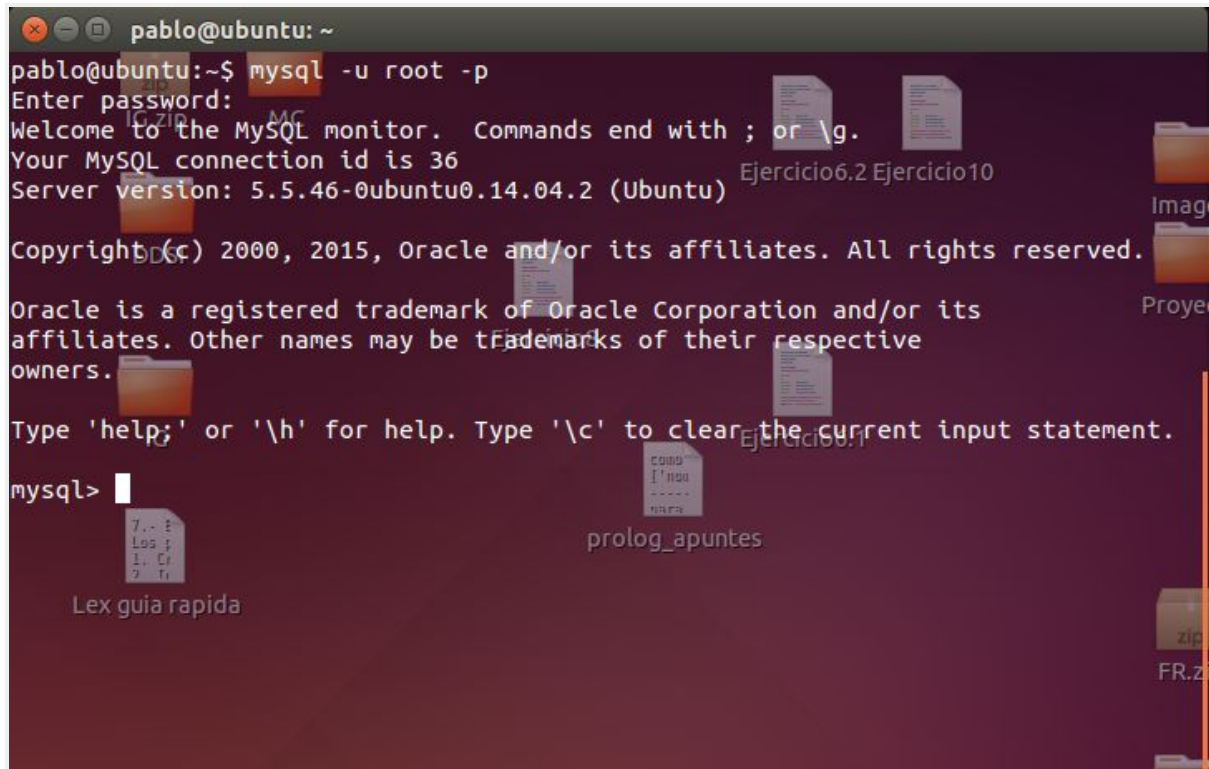


Figura 8.2: Resultado del comando `strace clear` (2)

Se puede apreciar como un comando tan simple como clear en el terminal, puede resultar más complejo de lo que es viendo tantas llamadas al sistema que genera y tantos procesos

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).

Primero accederemos a la base de datos con el comando `mysql -u root -p`



```
pablo@ubuntu: ~  
pablo@ubuntu:~$ mysql -u root -p  
Enter password:  
Welcome to the MySQL monitor.  Commands end with ; or \g.  
Your MySQL connection id is 36  
Server version: 5.5.46-0ubuntu0.14.04.2 (Ubuntu)  
Copyright (c) 2000, 2015, 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: Entramos en mysql previamente haciendo ssh al servidor. Escribimos use familia para acceder una base de datos previamente realizada.



```
mysql> use familia  
Reading table information for completion of table and column names  
You can turn off this feature to get a quicker startup with -A  
Database changed  
mysql> clear  
mysql> SHOW TABLES  
-> ;  
+-----+  
| Tables in familia |  
+-----+  
| familia           |  
+-----+  
1 row in set (0.00 sec)
```

Figura 9.2: use familia para cambiar de base datos y lo mostramos .

Ahora pondremos el 'profile' en activo.¹¹

set profiling = 1;

```
mysql> set profiling = 1;
Query OK, 0 rows affected (0.00 sec)
```

Figura 9.3: set profiling = 1 para activarlo.

Ahora mostramos las tablas completas con select * from familia;

```
mysql> select * from familia;
+-----+-----+-----+
| nombre | apellido | sexo |
+-----+-----+-----+
| Juan   | Ruano    | H    |
| Lucia  | Ruano    | M    |
| Marti  | Ferrer   | M    |
| Manuel | Hernandez| H    |
+-----+-----+-----+
4 rows in set (0.00 sec)
```

Figura 9.4: select * from familia para mostrar la tabla.

Finalmente consultamos el "profiler" para ver cuánto tardó cada operación.

```
mysql> show profiles;
+-----+-----+-----+
| Query_ID | Duration | Query |
+-----+-----+-----+
| 1        | 0.00020025 | select * from familia |
+-----+-----+-----+
1 row in set (0.00 sec)
```

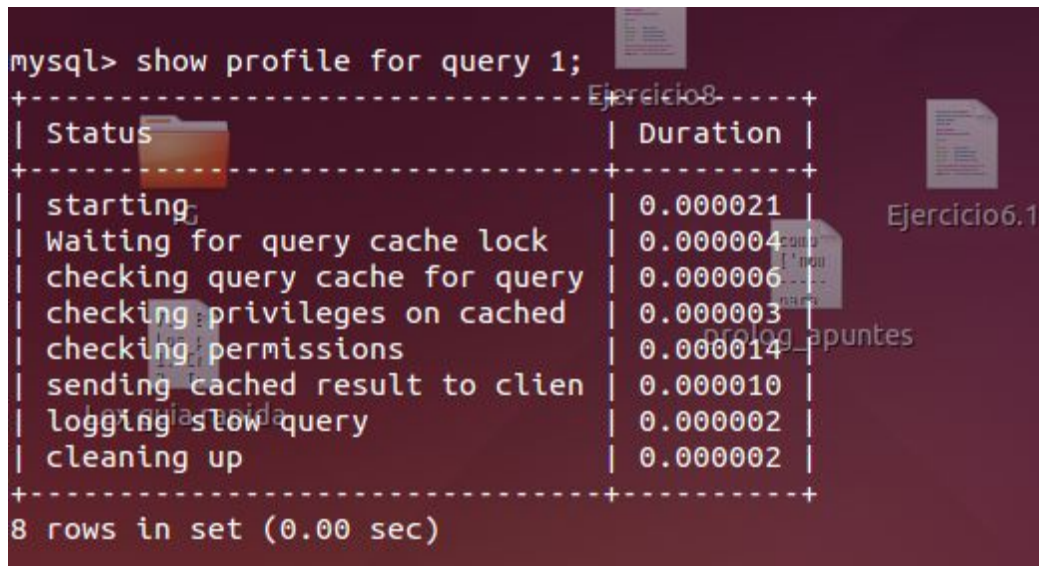
Figura 9.5: show profiles para ver tiempo que tarda la operación.

En este caso fueron operaciones muy veloces dada la magnitud de la base de datos. Como vemos "mysql" de por sí nos da un tiempo de hasta 8 decimales tras cada operación y vemos que para el select nos dijo que tardó 0.00020025s. Se ve que abajo redondea hasta 2 decimales.

También existe una opción más avanzada show profile for query 1;¹² que muestra información más detallada.

¹¹ "MySQL 5.5 Reference Manual :: 13.7.5.31 SHOW PROFILE ..." 2011.
<<https://dev.mysql.com/doc/refman/5.5/en/show-profile.html>>

¹² "MySQL 5.5 Reference Manual :: 13.7.5.31 SHOW PROFILE ..." 2011.
<<https://dev.mysql.com/doc/refman/5.5/en/show-profile.html>>



```
mysql> show profile for query 1;
```

Status	Duration
starting	0.000021
Waiting for query cache lock	0.000004
checking query cache for query	0.000006
checking privileges on cached	0.000003
checking permissions	0.000014
sending cached result to clien	0.000010
logging slow query	0.000002
cleaning up	0.000002

8 rows in set (0.00 sec)

Figura 9.6: show profile for query 1; para ver tiempo que tarda la operación, más detallada que el show profile.

Cuestión opcional 9.- Escriba un script en python y analice su comportamiento usando el profiler presentado.

En el siguiente guión para un uso correcto del profiler hemos investigado en la documentación¹³.

El guión es el siguiente:

```
ejercicio9.py x
1 import cProfile
2 class prueba:
3
4     def bucle(self):
5         j=1
6         print("Valor final de j:"+str(j))
7         for i in range(99000):
8             j = j + 1
9             print("Valor final de j:"+str(j))
10 p=prueba()
11 cProfile.run('p.bucle()')
12
```

Figura opciona 9.1: Script en python de ejemplo del profiler.

El resultado es el siguiente:

¹³ "26.4. The Python Profilers — Python 2.7.11rc1 documentation." 2014. 1 Dec. 2015
<<https://docs.python.org/2/library/profile.html>>


```

pablo@ubuntu: ~
pablo@ubuntu:~$ sudo python ejercicio9.py
Valor final de j:1
Valor final de i:99001
2 class function calls in 0.006 seconds
3
4 Ordered by: standardname
5
6 ncalls tottime ppcall valcuntine1 ppcall filename:lineno(function)
7 1 0.000 for 0:000 range(0:0000)0.006 <string>:1(<module>)
8 1 0.004 0.004 j = 0.006 0.006 ejercicio9.py:4(bucle)
9 1 0.000 print(Valor final de j:000 str(j))
ler p=objects}
11 cProfile.run(0.002.bucle0.002) 0.002 0.002 {range}
12
pablo@ubuntu:~$

```

Figura opcional 9.2: Resultado del script en python de ejemplo del profiler.

Cabe destacar en esta que los tiempos del profiler son realmente bajos, esto demuestra la rapidez de nuestro sistema para realizar este tipos de tareas con suma rapidez.

Cuestión opcional 11: Al igual que ha realizado el “profiling” con MySQL, realice lo mismo con MongoDB y compare los resultados (use la misma información y la misma consulta, hay traductores de consultas SQL a Mongo).

Bueno primero de todo crearemos una base datos llamada familia¹⁴ y una colección llamada familia y en ella insertamos los datos¹⁵.

```

> use familia
switched to db familia

```

Figura opcional 11.1: Crear base de datos familia.

```

> db.familia.save({nombre:'Juan',apellido:'Ruano',sexo:'H'})
2015-12-01T13:03:41.236+0100 E QUERY SyntaxError: Unexpected string
> db.familia.save({nombre:'Juan',apellido:'Ruano',sexo:'H'})
WriteResult({ "nInserted" : 1 })
> db.familia.save({nombre:'Lucia',apellido:'Ruano',sexo:'M'})
WriteResult({ "nInserted" : 1 })
> db.familia.save({nombre:'Mari',apellido:'Ferrer',sexo:'M'})
WriteResult({ "nInserted" : 1 })
> db.familia.save({nombre:'Manuel',apellido:'Hernandez',sexo:'H'})
WriteResult({ "nInserted" : 1 })

```

Figura opcional 11.2: Crear colección familia e insertar elementos.

¹⁴ "mongo Shell Quick Reference — MongoDB Manual 3.0." 2013.

<<https://docs.mongodb.org/manual/reference/mongo-shell/>>

¹⁵ "mongo Shell Quick Reference — MongoDB Manual 3.0." 2013.

<<https://docs.mongodb.org/manual/reference/mongo-shell/>>

En la documentación hemos visto cuál sería la conversión del select * from al Mongo¹⁶ sería así:

```
> db.familia.find();
{ "_id" : ObjectId("565d8cc37d65bc143823ceb8"), "nombre" : "Juan", "apellido" : "Ruano", "sexo" : "H" }
{ "_id" : ObjectId("565d8cea7d65bc143823ceb9"), "nombre" : "Lucia", "apellido" : "Ruano", "sexo" : "M" }
{ "_id" : ObjectId("565d8cfb7d65bc143823ceba"), "nombre" : "Mari", "apellido" : "Ferrer", "sexo" : "M" }
{ "_id" : ObjectId("565d8d0a7d65bc143823cebb"), "nombre" : "Manuel", "apellido" : "Hernandez", "sexo" : "H" }
```

Figura opcional 11.3: Conversión del select * from del MySql al Mongo.

En mongodb debemos indicar el nivel de profiling¹⁷ con valor 1 para consultas lentas y valor 2 para todas, como la nuestra será una consulta rápida usaremos el 2.¹⁸

```
> db.setProfilingLevel(2)
{ "was" : 0, "slowms" : 100, "ok" : 1 }
```

Figura opcional 11.4: Indicamos el nivel 2 porque vamos hacer una consulta rápida

Ahora realizaremos la consulta usando profiling.¹⁹

```
> db.system.profile.find();
{ "op" : "query", "ns" : "familia.familia.profile", "query" : { }, "ntoreturn" : 0, "ntoskip" : 0, "nscanned" : 0, "nscannedObjects" : 0, "keyUpdates" : 0, "writeConflicts" : 0, "numYield" : 0, "locks" : { "Global" : { "acquireCount" : { "r" : NumberLong(2) } }, "MMAPV1Journal" : { "acquireCount" : { "r" : NumberLong(1) } }, "Database" : { "acquireCount" : { "r" : NumberLong(1) } }, "Collection" : { "acquireCount" : { "R" : NumberLong(1) } } }, "nreturned" : 0, "responseLength" : 20, "millis" : 0, "execStats" : { "stage" : "EOF", "nReturned" : 0, "executionTimeMillisEstimate" : 0, "works" : 1, "advanced" : 0, "needTime" : 0, "needFetch" : 0, "saveState" : 0, "restoreState" : 0, "isEOF" : 1, "invalidates" : 0 }, "ts" : ISODate("2015-12-01T12:09:52.387Z"), "client" : "127.0.0.1", "allUsers" : [ ], "user" : "" }
{ "op" : "query", "ns" : "familia.system.profile", "query" : { }, "ntoreturn" : 0, "ntoskip" : 0, "nscanned" : 0, "nscannedObjects" : 1, "keyUpdates" : 0, "writeConflicts" : 0, "numYield" : 0, "locks" : { "Global" : { "acquireCount" : { "r" : NumberLong(2) } }, "MMAPV1Journal" : { "acquireCount" : { "r" : NumberLong(1) } }, "Database" : { "acquireCount" : { "r" : NumberLong(1) } }, "Collection" : { "acquireCount" : { "R" : NumberLong(1) } } }, "nreturned" : 1, "responseLength" : 694, "millis" : 0, "execStats" : { "stage" : "COLLSCAN", "filter" : { "Sand" : [ ] }, "nReturned" : 1, "executionTimeMillisEstimate" : 0, "works" : 3, "advanced" : 1, "needTime" : 1, "needFetch" : 0, "saveState" : 0, "restoreState" : 0, "isEOF" : 1, "invalidates" : 0, "direction" : "forward", "docsExamined" : 1 }, "ts" : ISODate("2015-12-01T12:10:07.538Z"), "client" : "127.0.0.1", "allUsers" : [ ], "user" : "" }
```

Figura opcional 11.5:Resultado de hacer profiling al Mongo

Como detalle final cabe destacar que el resultado de los find() en algunos casos son ilegibles y estoy se puede arreglar agregando al final ".pretty()"²⁰ después del find().

¹⁶ "mongo Shell Quick Reference — MongoDB Manual 3.0." 2013.

<<https://docs.mongodb.org/manual/reference/mongo-shell/>>

¹⁷ "Analyze Performance of Database Operations — MongoDB ..." 2013.

<<https://docs.mongodb.org/manual/tutorial/manage-the-database-profiler/>>

¹⁸ "Analyze Performance of Database Operations — MongoDB ..." 2013.

<<https://docs.mongodb.org/manual/tutorial/manage-the-database-profiler/>>

¹⁹ "Analyze Performance of Database Operations — MongoDB ..." 2013.

<<https://docs.mongodb.org/manual/tutorial/manage-the-database-profiler/>>

²⁰ "mongo Shell Quick Reference — MongoDB Manual 3.0." 2013.

<<https://docs.mongodb.org/manual/reference/mongo-shell/>>

```
> db.familia.find().pretty();
{
  "_id" : ObjectId("565d8cc37d65bc143823ceb8"),
  "nombre" : "Juan",
  "apellido" : "Ruano",
  "sexo" : "H"
}
{
  "_id" : ObjectId("565d8cea7d65bc143823ceb9"),
  "nombre" : "Lucia",
  "apellido" : "Ruano",
  "sexo" : "M"
}
{
  "_id" : ObjectId("565d8cfb7d65bc143823ceba"),
  "nombre" : "Mari",
  "apellido" : "Ferrer",
  "sexo" : "M"
}
{
  "_id" : ObjectId("565d8d0a7d65bc143823cebb"),
  "nombre" : "Manuel",
  "apellido" : "Hernandez",
  "sexo" : "H"
}
```

Figura opcional 11.6:Resultado de añadir pretty() a los find().