



*ugr* | Universidad  
de **Granada**

**Grado en Ingeniería Informática.**

## Práctica 3.

---

**Nombre de la asignatura:**  
Ingeniería de Servidores.

**Realizado por:**  
Néstor Rodríguez Vico



**ESCUELA TÉCNICA SUPERIOR DE INGENIERÍAS  
INFORMÁTICA Y DE TELECOMUNICACIÓN.**

---

Granada, 5 de diciembre de 2016.

# Índice

|  |           |
|--|-----------|
| <b>1. Cuestión 1:</b>  | <b>7</b>  |
| 1.1. a) ¿Qué archivo le permite ver qué programas se han instalado con el gestor de paquetes? . . . . .  | 7         |
| 1.2. b) ¿Qué significan las terminaciones .1.gz o .2.gz de los archivos en ese directorio? . . . . .   | 7         |
| <b>2. Cuestión 2: ¿qué archivo ha de modificar para programar una tarea? Escriba la línea necesaria para ejecutar una vez al día una copia del directorio /codigo a /seguridad/\$fecha donde \$fecha es la fecha actual (puede usar el comando date).</b>  | <b>8</b>  |
| <b>3. Cuestión 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.</b>  | <b>10</b> |
| <b>4. Cuestión 4: Ejecute el monitor de “System Performance” y muestre el resultado. Incluya capturas de pantalla comentando la información que aparece.</b>   | <b>11</b> |
| <b>5. Cuestión 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.</b> | <b>14</b> |
| <b>6. Cuestión 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.</b>  | <b>20</b> |
| <b>7. Cuestión 7: Escriba un breve resumen sobre alguno de los artículos donde se muestra el uso de strace o busque otro y coméntelo.</b>  | <b>24</b> |
| <b>8. Cuestión 8: Escriba un script en Python o PHP y analice su comportamiento usando el profiler presentado.</b>   | <b>25</b> |
| <b>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).</b>  | <b>27</b> |
| <b>10. Cuestión opcional 1: Indique qué comandos ha utilizado para realizarlo así como capturas de pantalla del proceso de reconstrucción del RAID.</b>  | <b>31</b> |

|  |           |
|--|-----------|
| <b>11. Cuestión opcional 2: Instale Nagios en su sistema (el que prefiera) documentando el proceso y muestre el resultado de la monitorización de su sistema comentando qué aparece.</b>   | <b>33</b> |
| <b>12. Cuestión opcional 4: Pruebe a instalar este monitor en alguno de sus tres sistemas. Realice capturas de pantalla del proceso de instalación y comente capturas de pantalla del programa en ejecución.</b>                     | <b>38</b> |
| <b>13. Cuestión opcional 5: Pruebe a instalar este monitor en alguno de sus tres sistemas. Realice capturas de pantalla del proceso de instalación y comente capturas de pantalla del programa en ejecución.</b>                     | <b>46</b> |
| <b>14. Cuestión opcional 6: Instale el monitor. Muestre y comente algunas capturas de pantalla.</b>  | <b>53</b> |
| <b>15. Cuestión opcional 10: Escriba un script en PowerShell y analice su comportamiento usando el profiler presentado.</b>  | <b>56</b> |
| <b>16. 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).</b> | <b>59</b> |

# Índice de figuras

|       |   |    |
|-------|---|----|
| 1.1.  | A la izquierda el contenido de <code>/var/log/apt/history.log</code> en Ubuntu Server.<br>A la derecha el contenido de <code>/var/log/yum.log</code> en CentOS. . . . . | 7  |
| 1.2.  | Contenido del directorio <code>/var/log/</code> en mi máquina anfitrona. . . . .  | 8  |
| 2.1.  | Ejecución del comando <code>crontab -e</code> ; a la izquierda en Ubuntu Server y a la derecha en CentOS. . . . .   | 8  |
| 2.2.  | Creación del script; a la izquierda en Ubuntu Server y a la derecha en CentOS. . . . .  | 9  |
| 2.3.  | Fichero <code>crontab</code> modificado; a la izquierda en Ubuntu Server y a la derecha en CentOS. . . . .  | 9  |
| 3.1.  | <code>dmesg   tail</code> antes de conectar el USB. . . . .   | 10 |
| 3.2.  | <code>dmesg   tail</code> después de conectar el USB. . . . .   | 10 |
| 4.1.  | Resumen del resultado del análisis. . . . .   | 11 |
| 4.2.  | Categorías de información del análisis. . . . .   | 12 |
| 4.3.  | Información acerca de los procesos. . . . .   | 13 |
| 4.4.  | Información acerca del uso de la memoria. . . . .   | 13 |
| 5.1.  | Comenzamos la creación del recopilador de datos. . . . .  | 15 |
| 5.2.  | Elegimos el nombre de nuestro recopilador de datos. . . . .   | 15 |
| 5.3.  | Elegimos que tipo de datos queremos recopilar. . . . .  | 16 |
| 5.4.  | Elegimos que contadores de rendimiento queremos registrar. . . . .  | 16 |
| 5.5.  | Datos seleccionados. . . . .  | 17 |
| 5.6.  | Elegimos la frecuencia de muestro, 15 segundos en nuestro caso. . . . .   | 17 |
| 5.7.  | Dejamos los proveedores de seguimiento de eventos en blanco. . . . .  | 18 |
| 5.8.  | Guardamos los <code>logs</code> en la carpeta <code>logs</code> dentro del <i>Escritorio</i> . . . . .  | 18 |
| 5.9.  | Elegimos iniciar el recopilador y finalizamos el proceso. . . . .   | 19 |
| 5.10. | Resultado obtenido del análisis. . . . .  | 20 |
| 6.1.  | Parte de los parámetros disponibles para monitorizar. . . . .   | 22 |
| 6.2.  | Entradas y salidas de disco. . . . .  | 22 |
| 6.3.  | Número de hilos. . . . .  | 23 |
| 6.4.  | Interrupciones y cambios de contexto. . . . .   | 23 |
| 6.5.  | Uso de la memoria. . . . .  | 24 |
| 8.1.  | Script que imprime los números impares entre 0 y 100 y los suma. . . . .  | 26 |
| 8.2.  | Resultado del profiler. . . . .   | 26 |
| 9.1.  | Conexión con phpMyAdmin (máquinas conectadas en modo <i>host-only</i> ). . . . .  | 27 |
| 9.2.  | Preparativos para el profiling. . . . .   | 28 |
| 9.3.  | Sentencias ejecutadas. . . . .  | 28 |
| 9.4.  | Inserción de tuplas. . . . .  | 29 |
| 9.5.  | Recuperación de todas las tuplas de la tabla. . . . .   | 29 |
| 9.6.  | Datos recogidos por el profiler tras la consulta. . . . .   | 30 |
| 10.1. | Comprobamos que tenemos dos RAID. . . . .   | 31 |
| 10.2. | Provocamos un fallo software en el RAID. . . . .  | 32 |
| 10.3. | Quitamos en caliente y añadimos de nuevo el RAID. . . . .   | 32 |

|  |    |
|--|----|
| 10.4. Observamos el proceso de recuperación con <i>watch</i> . . . . .   | 32 |
| 10.5. RAID completamente restaurado. . . . .   | 32 |
| 11.1. Proceso de descarga de Nagios y preparación para la instalación. . . . .                                       | 34 |
| 11.2. Proceso de instalación de Nagios. . . . .  | 34 |
| 11.3. Conexión con Nagios (máquina conectadas en modo <i>host-only</i> ). . . . .                                    | 35 |
| 11.4. Paso 1 de la configuración de la interfaz web (máquina conectadas en modo <i>host-only</i> ). . . . .          | 35 |
| 11.5. Paso 2 de la configuración de la interfaz web (máquina conectadas en modo <i>host-only</i> ). . . . .          | 36 |
| 11.6. Conexión con Nagios (máquina conectadas en modo <i>host-only</i> ). . . . .                                    | 36 |
| 11.7. Monitorización del rendimiento con Nagios (máquina conectadas en modo <i>host-only</i> ). . . . .              | 37 |
| 11.8. Hypermap de mi servidor (máquina conectadas en modo <i>host-only</i> ). . . . .                                | 37 |
| 12.1. Archivo <i>/etc/apt/sources.list</i> . . . . .   | 38 |
| 12.2. Instalación de <i>Zabbix</i> . . . . .   | 39 |
| 12.3. Configuración del servidor de <i>Zabbix</i> . . . . .  | 39 |
| 12.4. Extracción de ficheros SQL. . . . .  | 40 |
| 12.5. Configuración de MySQL. . . . .  | 40 |
| 12.6. Importación de archivos necesarios para <i>Zabbix</i> . . . . .  | 41 |
| 12.7. Parámetros del archivo <i>/etc/php5/apache2/php.ini</i> modificados. . . . .                                   | 41 |
| 12.8. Parámetros del archivo <i>/etc/zabbix/zabbix.conf.php</i> modificados. . . . .                                 | 42 |
| 12.9. Configuración archivo apache de <i>Zabbix</i> . . . . .  | 42 |
| 12.10 Inicio de <i>Zabbix</i> . . . . .  | 43 |
| 12.11 Instalación del <i>Agente de Zabbix</i> . . . . .  | 43 |
| 12.12 Dirección IP del servidor con <i>Zabbix</i> instalado (máquinas conectadas en modo <i>host-only</i> ). . . . . | 43 |
| 12.13 Parámetro <i>hostname</i> . . . . .  | 44 |
| 12.14 <i>Zabbix</i> en funcionamiento. . . . .   | 44 |
| 12.15 Host disponibles. . . . .  | 44 |
| 12.16 Cambio del estado del servidor. . . . .  | 45 |
| 12.17 Parámetros monitorizados. . . . .  | 45 |
| 12.18 Valores procesados por el servidor <i>Zabbix</i> por segundo. . . . .  | 46 |
| 13.1. Instalación de <i>Cacti</i> . . . . .  | 46 |
| 13.2. Iniciamos los servicios. . . . .   | 47 |
| 13.3. Creación base de datos para <i>Cacti</i> . . . . .   | 48 |
| 13.4. Importación de las tablas para <i>Cacti</i> . . . . .  | 48 |
| 13.5. Parámetros modificados. . . . .  | 49 |
| 13.6. Permisos modificados. . . . .  | 49 |
| 13.7. Archivo de configuración de <i>Apache</i> y archivo <i>crontab</i> modificados. . . . .                        | 50 |
| 13.8. <i>Cacti</i> en funcionamiento. . . . .  | 50 |
| 13.9. Instalación nueva. . . . .   | 51 |
| 13.10 Finalización de la instalación. . . . .  | 51 |
| 13.11 Instalación de <i>Cacti</i> completada. . . . .  | 51 |
| 13.12 Cambio forzado de contraseña. . . . .  | 52 |

|  |    |
|--|----|
| 13.13Añadimos el gráfico al árbol. . . . .                   | 52 |
| 13.14Datos recogidos. . . . .                                | 53 |
| 14.1. Configuración de <i>AWStats</i> . . . . .              | 54 |
| 14.2. Configuración de <i>Apache</i> . . . . .               | 54 |
| 14.3. Fichero de <i>crontab</i> modificado. . . . .          | 55 |
| 14.4. Información proporcionada por <i>AWStats</i> . . . . . | 56 |
| 15.1. Resultado del script 15. . . . .                       | 57 |
| 15.2. Error al ejecutar el profiler. . . . .                 | 58 |
| 15.3. Ejemplo de ejecución de PoshProfiler [1]. . . . .      | 58 |
| 16.1. Creación de la colección y de los documentos. . . . .  | 60 |
| 16.2. Inserción y consulta de los documentos. . . . .        | 60 |
| 16.3. Entrada más reciente. . . . .                          | 61 |
| 16.4. 8 entradas más recientes. . . . .                      | 61 |

## 1. Cuestión 1:

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

Como podemos ver en la página de Ubuntu [2] el archivo que contiene dicha información en Ubuntu Server es `/var/log/apt/history.log`. En CentOS se encuentra en la misma ruta. El archivo es `/var/log/yum.log`. El contenido de dichos archivos se puede ver en la figura 1.1.

Figura 1.1: A la izquierda el contenido de `/var/log/apt/history.log` en Ubuntu Server. A la derecha el contenido de `/var/log/yum.log` en CentOS.

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

En mi caso existen dichos archivos en el directorio `/var/log`, como podemos ver en la imagen 1.2. Dichos archivos son creados con el comando `logrotate` [3] de forma periódica, para comprimir los *logs* y que el fichero *log* no crezca demasiado.

```
nrv/2016-11-18:~$ ls /var/log/
alternatives.log auth.log.2.gz dmesg.3.gz kern.log.2.gz syslog.4.gz
apport.log boot.log dmesg.4.gz lastlog lightdm syslog.5.gz
apport.log.1 bootstrap.log dpkg.log faillog pm-powersave.log syslog.6.gz
apport.log.2.gz btmp fontconfig.log pm-suspend.log syslog.7.gz
apport.log.3.gz ConsoleKit gpu-manager.log speech-dispatcher udev
apport.log.4.gz cups fsck samba unattended-upgrades
apport.log.5.gz dist-upgrade gpu-manager.log syslog upstart
apport.log.6.gz dmesg hp syslog.1 wtmp
apt dmesg.0 installer syslog.1 Xorg.0.log
auth.log dmesg.1.gz kern.log syslog.2.gz Xorg.0.log.old
auth.log.1 dmesg.2.gz kern.log.1 syslog.3.gz
nrv/2016-11-18:~$ |
```

Figura 1.2: Contenido del directorio `/var/log/` en mi máquina anfitrona.

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

Cómo podemos ver en el manual de `crontab` [4] si queremos editar el archivo `crontab` actual debemos ejecutar el comando `crontab -e`, como podemos ver en la figura 2.1. En el caso de Ubuntu Server el fichero es `/tmp/crontab.kWQ61Y/crontab`. En el caso de CentOS el fichero es `/tmp/crontab.yxrpPw`.

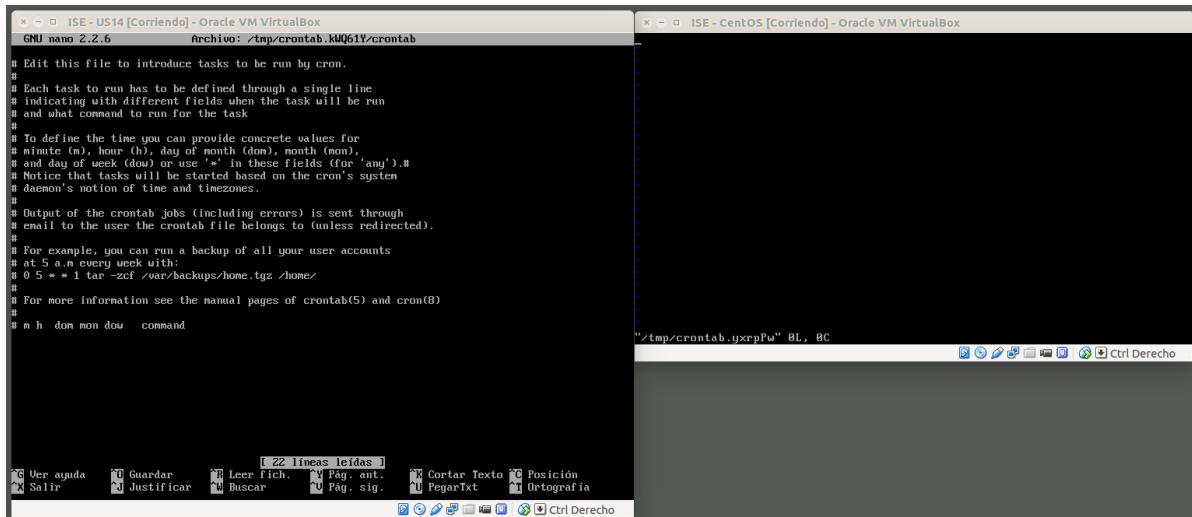


Figura 2.1: Ejecución del comando `crontab -e`; a la izquierda en Ubuntu Server y a la derecha en CentOS.

Lo primero que he hecho ha sido crear el script 2 y ubicarlo en el directorio `~`, como

se puede ver en la figura 2.2. Una vez tenemos todo listo, editamos el fichero *crontab* añadiendo la línea *00 11 \* \* \* ~/script.sh*, como se puede ver en el script <sup>1</sup> 2.

Script 2:

```
#!/bin/bash

fecha=$( date +"%m_%d_%Y" )
mkdir -p ~/seguridad
mkdir -p ~/seguridad/$fecha
cp -r ~/codigo ~/seguridad/$fecha
```

```
nrv/2016-11-18:~$ pwd
/home/nestor
nrv/2016-11-18:~$ ls -l
total 1
-rw-rw-r-- 1 nestor nestor 122 nov 18 16:40 script.sh
nrv/2016-11-18:~$ cat script.sh
#!/bin/bash
fecha=$(date +"%m_%d_%Y")
mkdir -p ~/seguridad/
mkdir -p ~/seguridad/$fecha
cp -r ~/codigo ~/seguridad/$fecha
nrv/2016-11-18:~$ _

nrv/2016-11-18:~$ pwd
/home/nrv
nrv/2016-11-18:~$ ls -l
total 4
-rw-rw-r--. 1 nrv nrv 121 nov 18 16:38 script.sh
nrv/2016-11-18:~$ cat script.sh
#!/bin/bash
fecha=$(date +"%m_%d_%Y")
mkdir -p ~/seguridad
mkdir -p ~/seguridad/$fecha
cp -r ~/codigo ~/seguridad/$fecha
nrv/2016-11-18:~$ _
```

Figura 2.2: Creación del script; a la izquierda en Ubuntu Server y a la derecha en CentOS.

```
GNU nano 2.2.6          Archivo: /tmp/crontab.mneb7K/crontab      Modificado
# Edit this file to introduce tasks to be run by cron.
#
# Each task 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 timezone.
#
# Output of the cron 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
00 11 * * * ~/script.sh
```

Figura 2.3: Fichero *crontab* modificado; a la izquierda en Ubuntu Server y a la derecha en CentOS.

<sup>1</sup>El script “2-scriptcrontab.sh” se encuentra dentro de la carpeta *Archivos auxiliares*.

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

El resultado de ejecutar `dmesg | tail` antes de conectar un USB es el que podemos ver en la figura 3.1.

```
nrv/2016-11-17:~$ dmesg | tail
[ 2799.301819] sd 9:0:0:0: [sdb] 8241152 512-byte logical blocks: (4.22 GB/3.93 GiB)
[ 2799.301940] sd 9:0:0:0: [sdb] Write Protect is off
[ 2799.301944] sd 9:0:0:0: [sdb] Mode Sense: 0b 00 00 08
[ 2799.302061] sd 9:0:0:0: [sdb] No Caching mode page found
[ 2799.302064] sd 9:0:0:0: [sdb] Assuming drive cache: write through
[ 2799.305084] sdb: sdb1
[ 2799.306763] sd 9:0:0:0: [sdb] Attached SCSI removable disk
[ 2799.459155] FAT-fs (sdb1): Volume was not properly unmounted. Some data may be corrupt. Please run fsck.
[ 2848.816034] sdb: detected capacity change from 4219469824 to 0
[ 2857.031803] usb 3-2: USB disconnect, device number 12
```

Figura 3.1: `dmesg | tail` antes de conectar el USB.

El resultado de ejecutar `dmesg | tail` después de conectar un USB es el que podemos ver en la figura 3.2.

```
nrv/2016-11-17:~$ dmesg | tail
[ 2886.722992] scsi host10: usb-storage 3-2:1.0
[ 2887.721025] scsi 10:0:0:0: Direct-Access      JetFlash Transcend 16GB    8.07 PQ: 0 ANSI: 4
[ 2887.722424] sd 10:0:0:0: [sdb] 30523392 512-byte logical blocks: (15.6 GB/14.6 GiB)
[ 2887.723108] sd 10:0:0:0: [sdb] Write Protect is off
[ 2887.723111] sd 10:0:0:0: [sdb] Mode Sense: 23 00 00 00
[ 2887.723333] sd 10:0:0:0: Attached scsi generic sg2 type 0
[ 2887.724922] sd 10:0:0:0: [sdb] Write cache: disabled, read cache: enabled, doesn't support DPO or FUA
[ 2887.728313] sdb:
[ 2887.731186] sd 10:0:0:0: [sdb] Attached SCSI removable disk
[ 2887.897591] FAT-fs (sdb): Volume was not properly unmounted. Some data may be corrupt. Please run fsck.
nrv/2016-11-17:~$ |
```

Figura 3.2: `dmesg | tail` después de conectar el USB.

En la salida tras conectar el almacenamiento USB podemos ver la siguiente información:

- Línea 1: Se detecta el almacenamiento USB.
- Línea 2: Podemos ver el nombre del USB y su capacidad, 16GB.
- Línea 3: Vemos que hay 30523392 bloques lógicos de 512 byte cada uno. De los 15.6 GiB disponibles hay 14.6 GiB libres.

- Línea 4: La protección de escritura está desactivada, esto nos indica que se pueden escribir datos en el almacenamiento USB.
- Línea 7: La escritura en caché está desactivada y la lectura en caché activada.
- Línea 9: Es un almacenamiento USB que se puede remover.
- Línea 10: Mensaje que indica que no se desmontó correctamente y que algunos datos podrían estar corruptos.

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

Una vez terminado el análisis, nos aparece el informe con la información recogida. Podemos ver información de todo tipo, desde información sobre la CPU, la red, el disco o la memoria, como podemos ver en la figura 4.1.

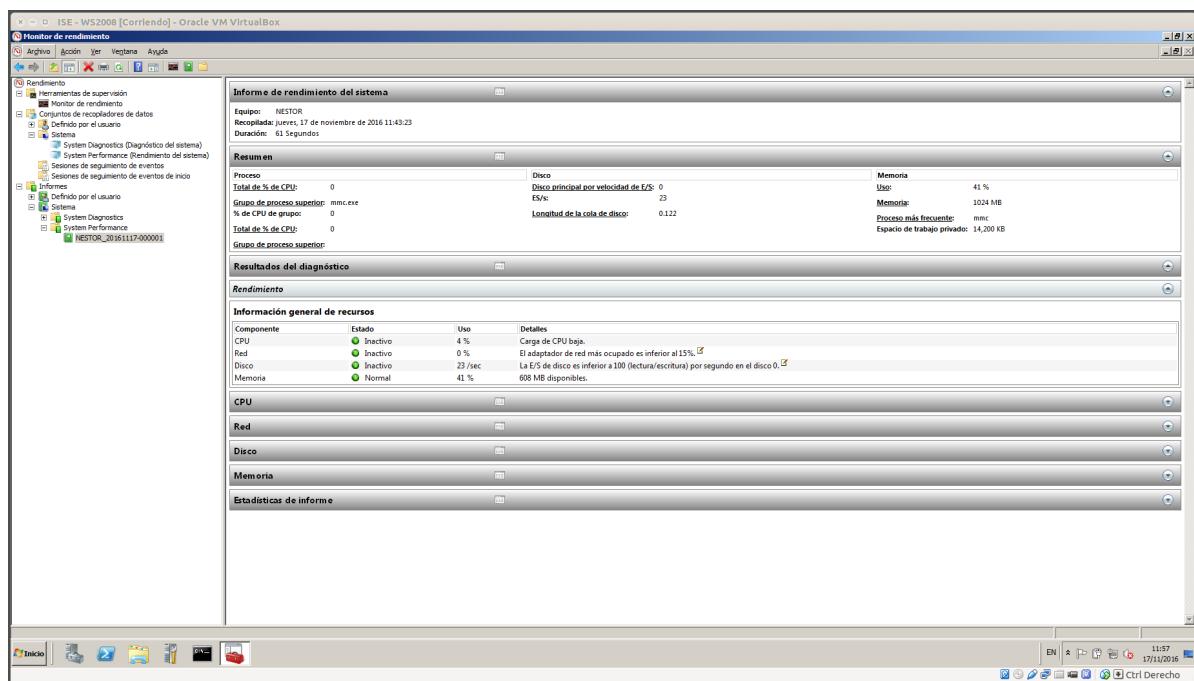


Figura 4.1: Resumen del resultado del análisis.

El análisis nos aporta información de los campos que podemos ver en la figura 4.2:

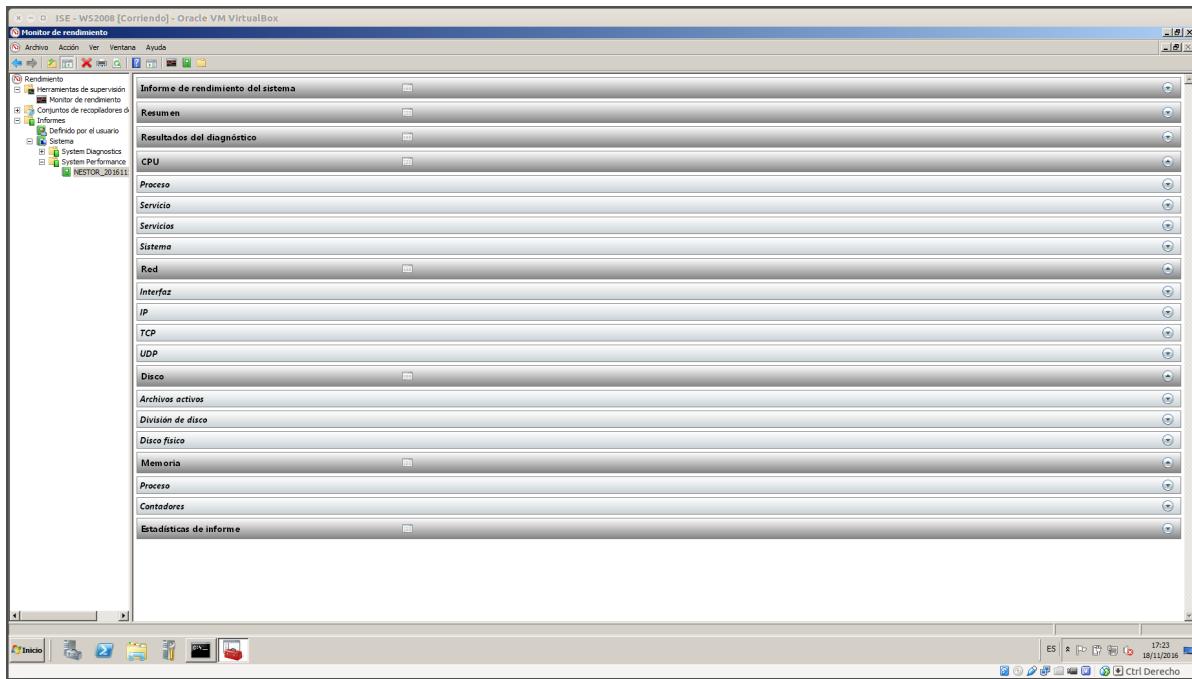


Figura 4.2: Categorías de información del análisis.

Dentro de la categoría de los procesos podemos ver información sobre que procesos se han estado ejecutando durante el análisis e información sobre los mismos. Podemos ver que, por ejemplo, el proceso *rundll32.exe* tiene el identificador de proceso 1152, ha iniciado 7 subprocesos, ha usado 3 subprocesos y ha ocupado un 0.1 % de CPU. También podemos ver información acerca del número de subprocesos o incluso el número de interrupciones que ha habido durante la ejecución del análisis. Todo esto y mucha más información lo podemos ver en la figura 4.3.

Dentro de la categoría de los procesos podemos ver información sobre el uso de la memoria. Por ejemplo, podemos ver que durante el análisis se han producido de media 40 errores de páginas, ha habido de media 0.117 errores de caché y el número medio de páginas entradas libres en la tabla de páginas del sistema asciende a 33554694. Todo esto y mucha más información lo podemos ver en la figura 4.4.

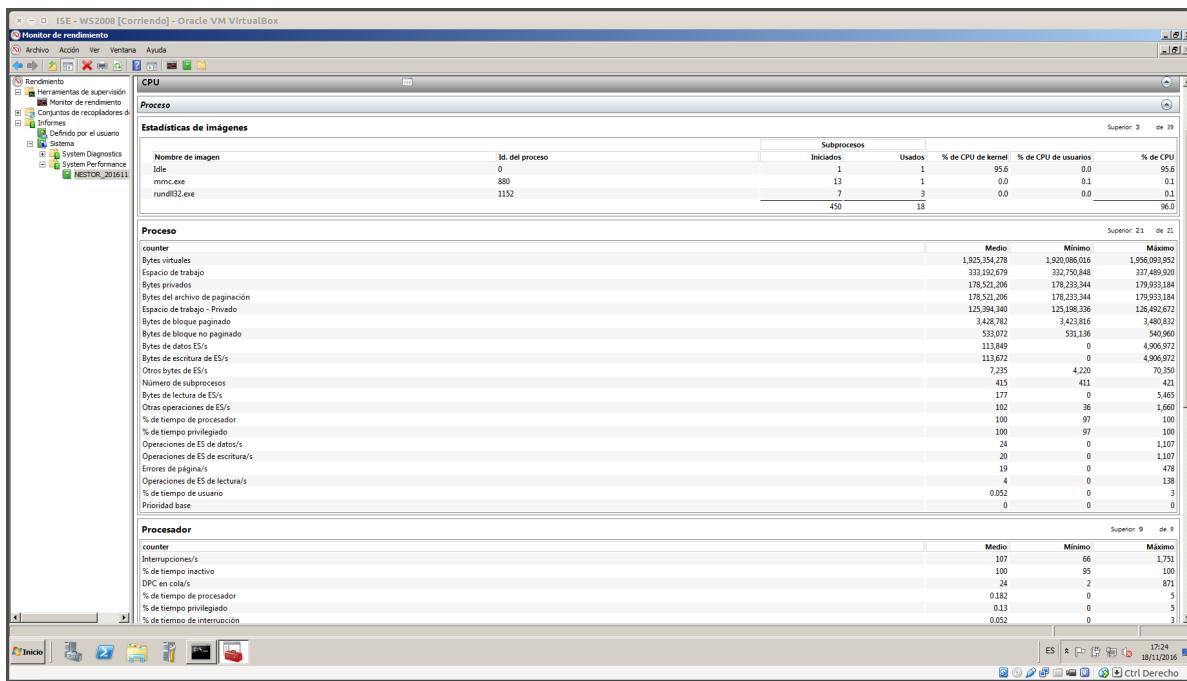


Figura 4.3: Información acerca de los procesos.

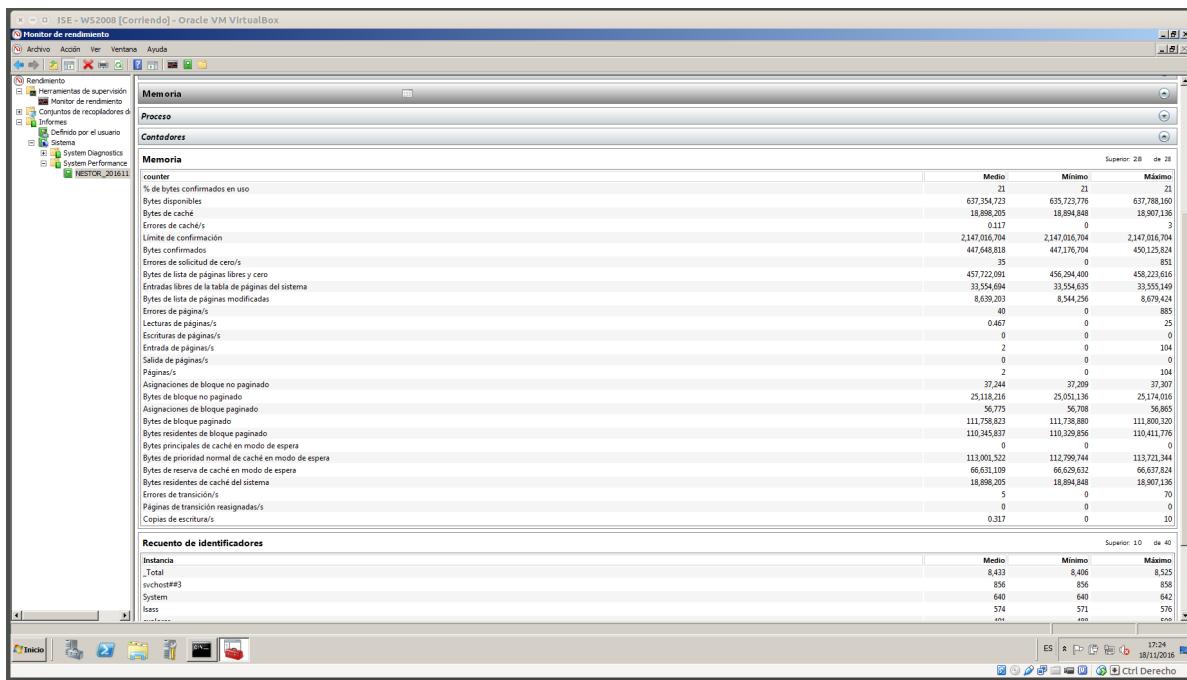


Figura 4.4: Información acerca del uso de la memoria.

**5. Cuestión 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.**

1. Nos posicionamos en la categoría de *Definidos por el usuario* dentro de *Conjuntos de recopiladores de datos* y le damos click derecho, a continuación a *Nuevo* y finalmente a *Conjunto de recopiladores de datos*, como podemos ver en la figura 5.1.
2. Elegimos el nombre del recopilador. En mi caso he elegido *RecopiladorISE*, como podemos ver en la figura 5.2.
3. Elegimos que tipos de datos queremos recolectar. En mi caso *Contador de rendimiento* y *Datos de seguimiento de eventos*, como podemos ver en la figura 5.3.
4. Ahora tenemos que seleccionar los datos que queremos recopilar. En un principio no hay ninguno seleccionado, como podemos ver en la figura 5.4.
5. Para seleccionar los datos, nos vamos a la categoría que queremos y le damos al botón de agregar. En nuestro caso debemos agregar la categoría entera de *procesador*, la de *proceso* y la de *servicio web*. El resultado final es el que podemos ver en la figura 5.5.
6. Una vez hemos escogido los datos, seleccionamos el intervalo de muestra, en nuestro caso elegimos 15 segundos, como podemos ver en la figura 5.6.
7. En los proveedores de seguimiento no seleccionamos nada, como podemos ver en la figura 5.7.
8. A continuación elegimos donde queremos que se guarde la *logs*, en nuestro caso queremos que se guarden en el la carpeta *logs* dentro del *Escritorio*, como podemos ver en la figura 5.8.
9. Finalmente, seleccionamos la opción de iniciar el recopilador y le damos a finalizar, como podemos ver en la figura 5.9.

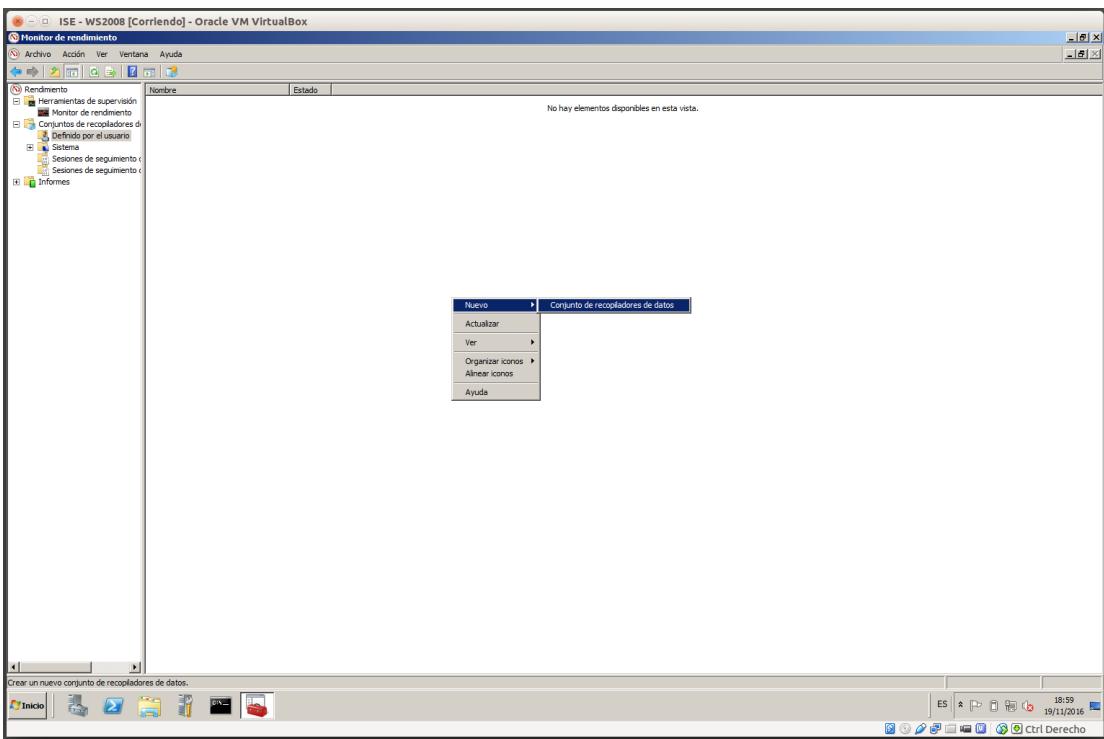


Figura 5.1: Comenzamos la creación del recopilador de datos.

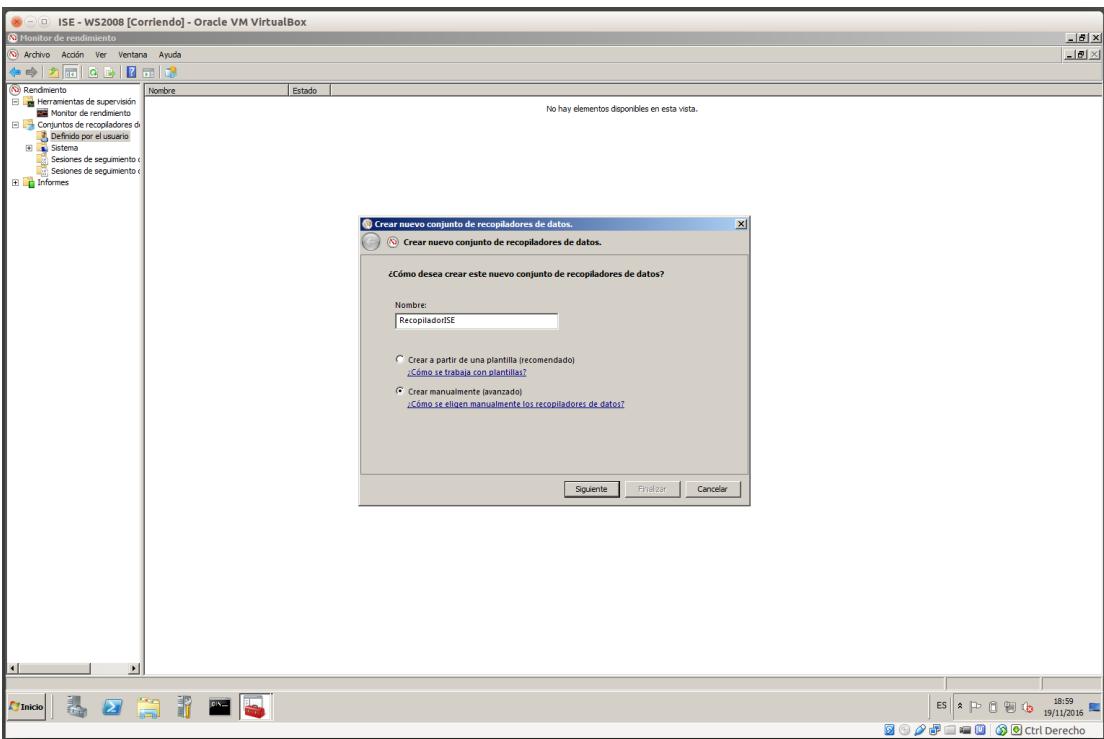


Figura 5.2: Elegimos el nombre de nuestro recopilador de datos.

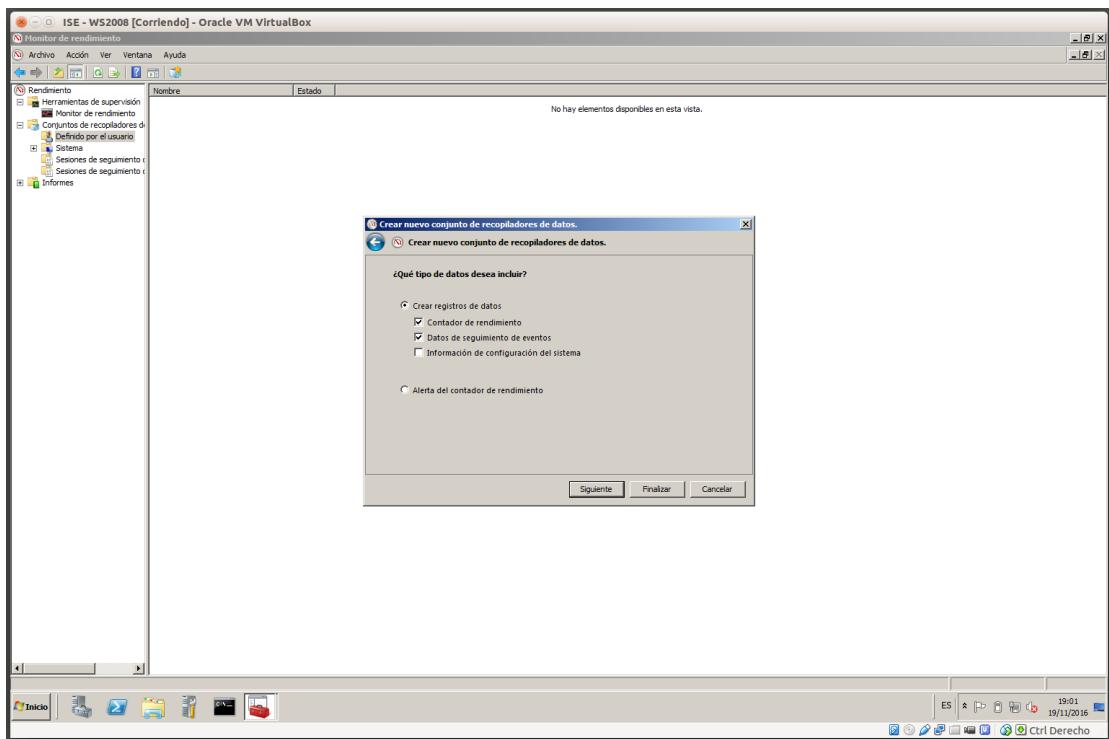


Figura 5.3: Elegimos que tipo de datos queremos recopilar.

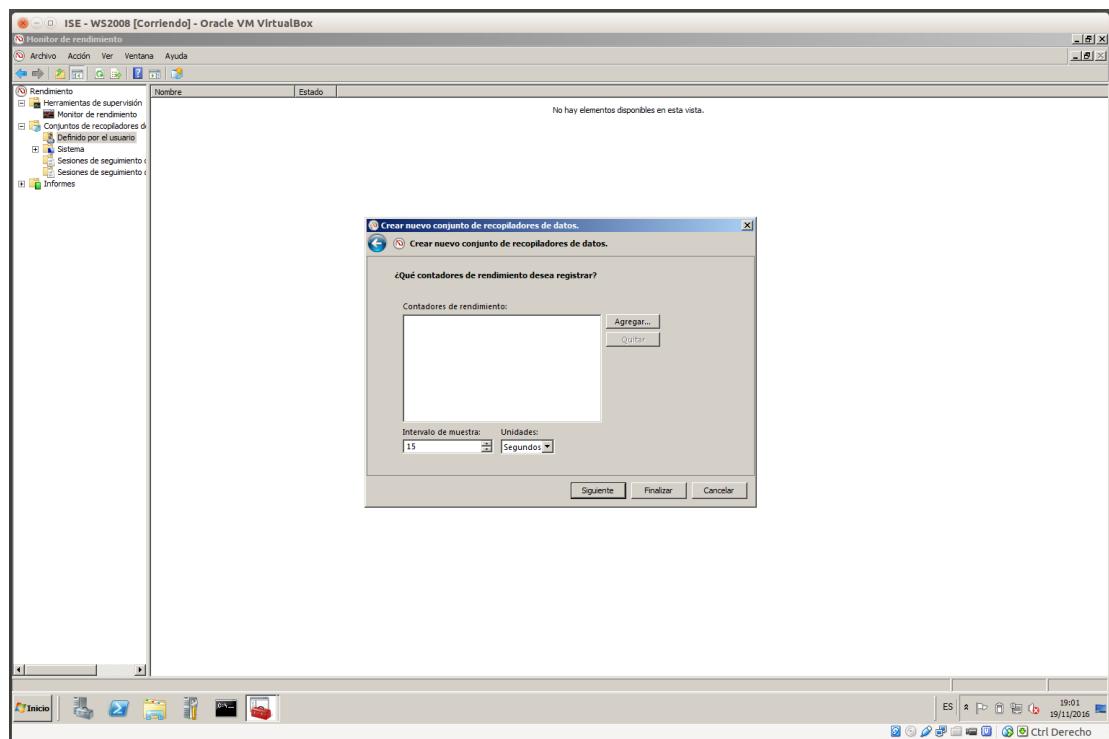


Figura 5.4: Elegimos que contadores de rendimiento queremos registrar.

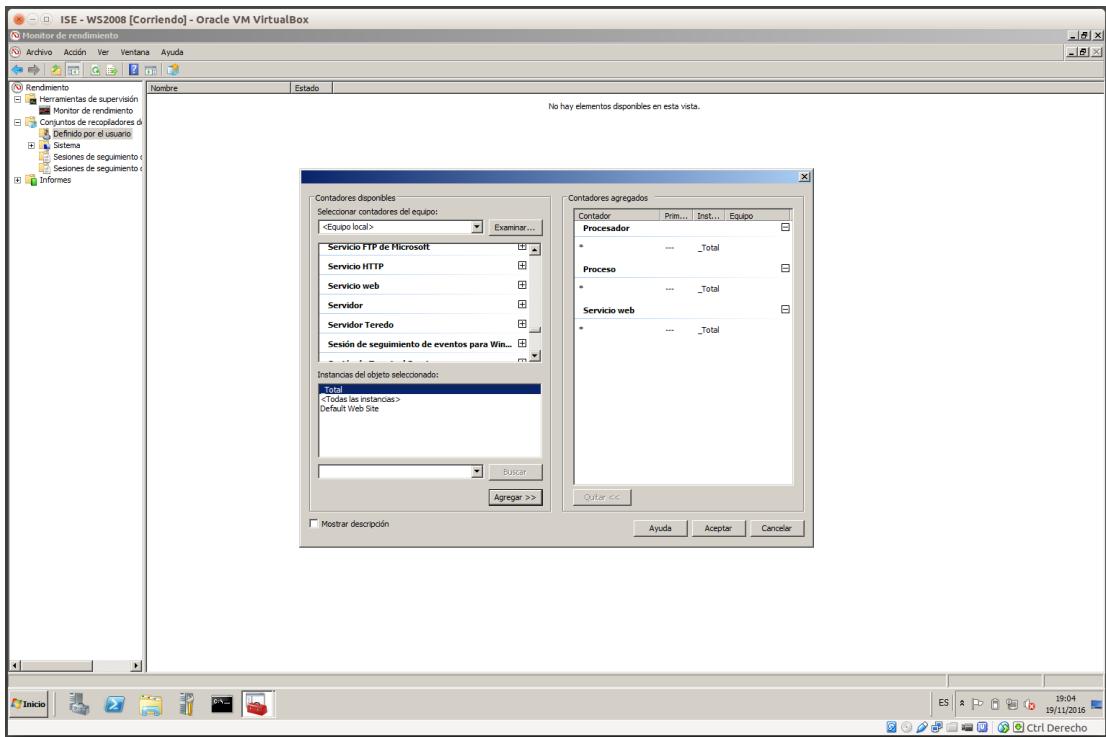


Figura 5.5: Datos seleccionados.

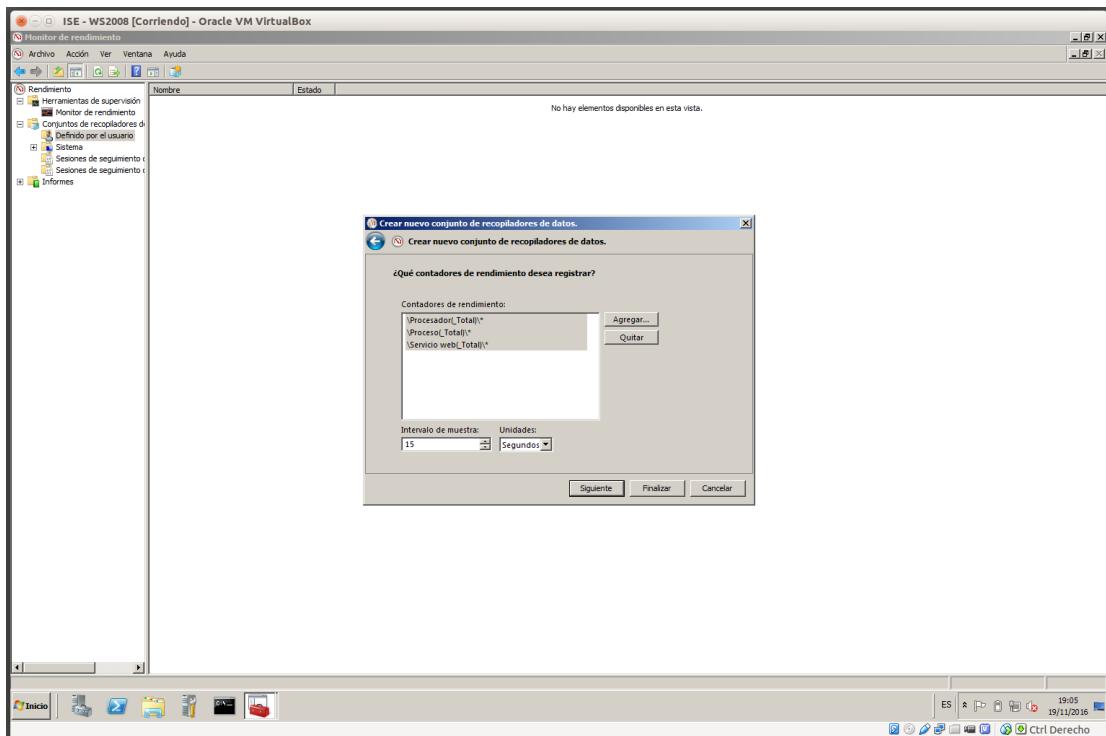


Figura 5.6: Elegimos la frecuencia de muestro, 15 segundos en nuestro caso.

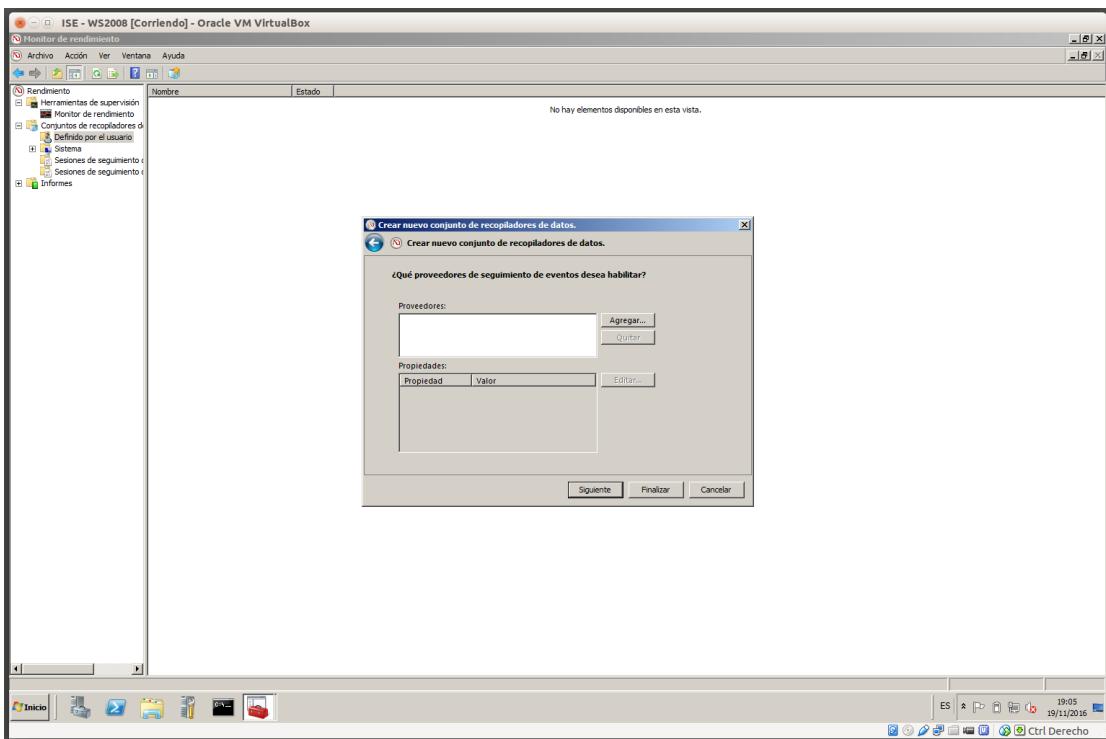


Figura 5.7: Dejamos los proveedores de seguimiento de eventos en blanco.

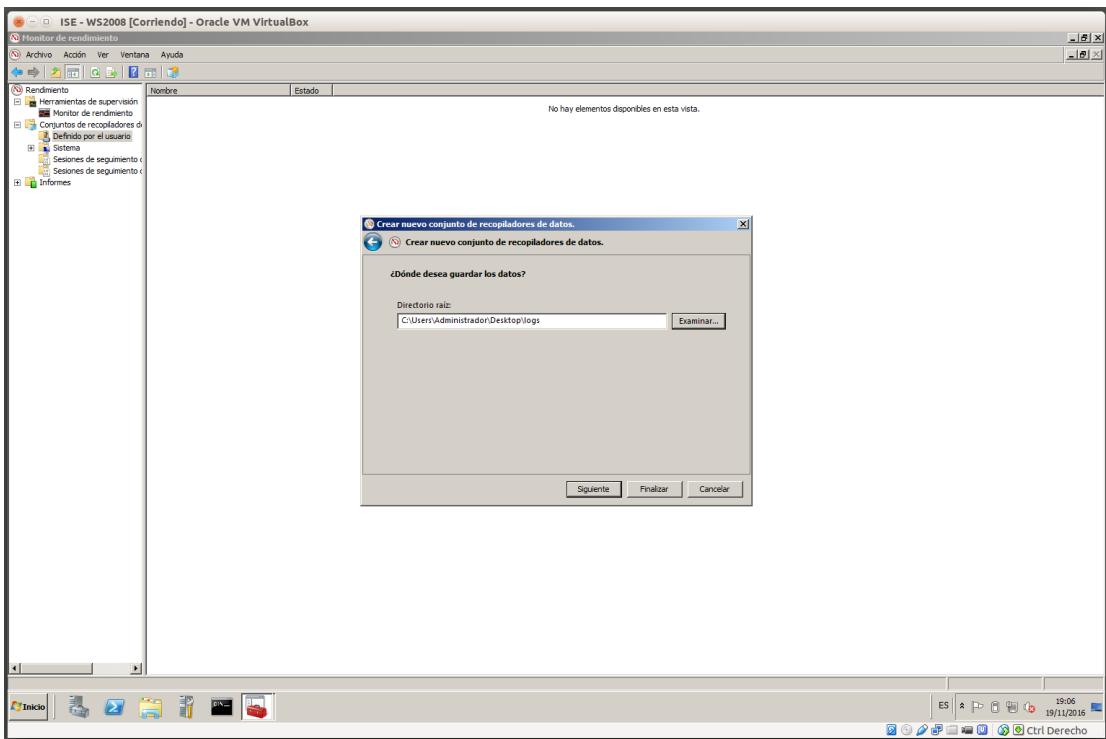


Figura 5.8: Guardamos los *logs* en la carpeta *logs* dentro del *Escritorio*.

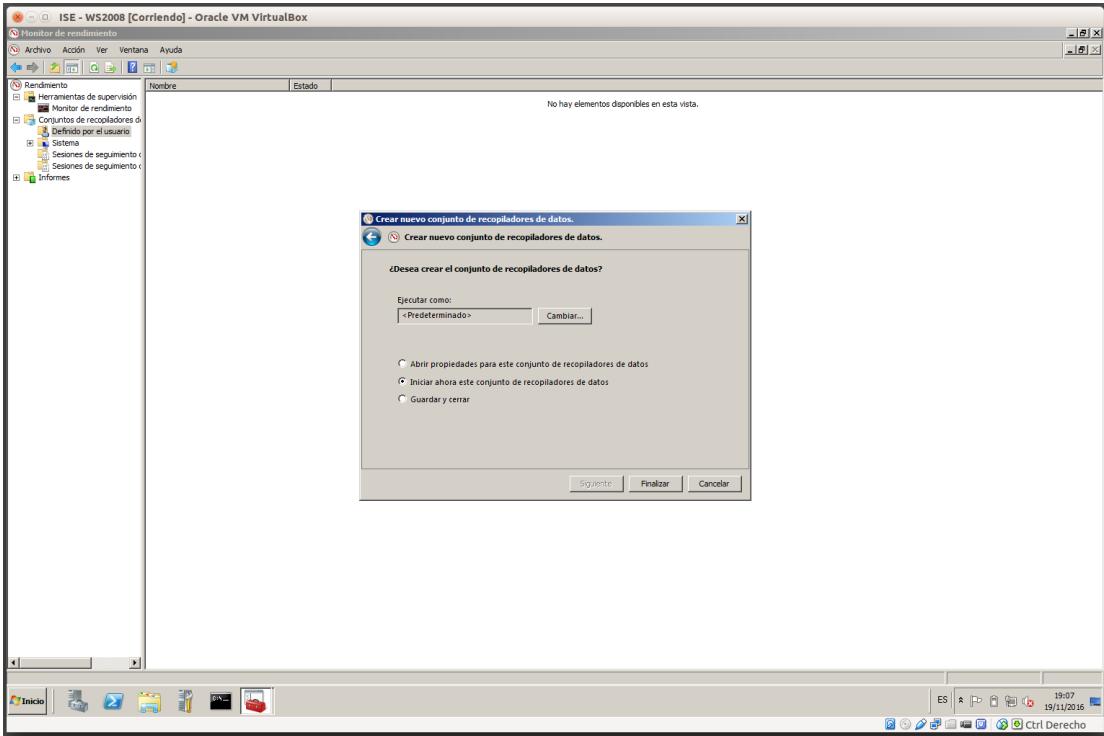


Figura 5.9: Elegimos iniciar el recopilador y finalizamos el proceso.

Una vez hemos creado el recopilador de datos, he generado un análisis para ver que todo funciona correctamente. Para ello he parado el recopilador y lo he puesto de nuevo en funcionamiento. Tras un minuto de análisis, el resultado obtenido lo podemos ver en la figura 5.10. Los primeros 5 segundos de análisis no he hecho nada, por eso no podemos ver nada destacable en la gráfica. Tras esto cinco segundos, he creado abierto un archivo de texto, lo he modificado y lo he guardado. Por eso podemos ver que en el instante de tiempo *20:18:46* empieza a haber datos sobre el número de bytes de lectura y escritura, que son las líneas de color rosa y verde fosforito (las dos seleccionadas en la figura 5.10). El resto de he modificado el archivo pero cambiando menos contenido y lo he ido guardando en distintos momentos del análisis, por eso podemos ver que siguen apareciendo datos sobre bytes de escritura y lectura. Finalmente, he cerrado el archivo y lo he abierto de nuevo, por eso comienza de nuevo a subir el número de bytes leídos (línea de color verde fosforito) en el instante de tiempo *20:19:46*.

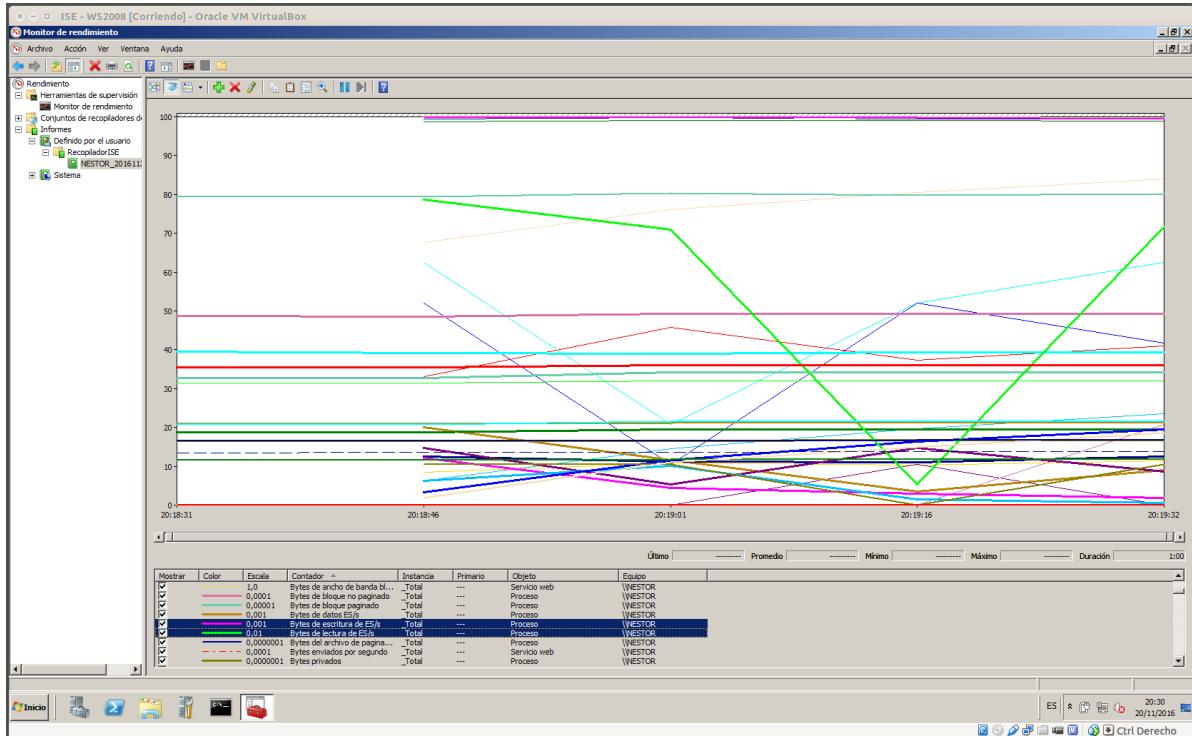


Figura 5.10: Resultado obtenido del análisis.

## 6. Cuestión 6: 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 vemos al acceder a la página de la demo de Munin [5] es que podemos monitorizar muchísimos parámetros distintos, como podemos ver en la figura 6.1.

Yo me he centrado en cuatro parámetros distintos: *entradas y salidas de disco*, *número de hilos*, *interrupciones y cambios de contexto* y *uso de la memoria*. En los datos sobre *entradas y salidas de disco* podemos ver la información recopilada en dos gráficas; en la primera de ellas se recoge los datos del parámetro monitorizado en el día actual y en la segunda los datos de la semana actual. En los tres parámetros restantes podemos ver la información recopilada en cuatro gráficas; en la primera de ellas se recoge los datos del parámetro monitorizado en el día actual, en la segunda los datos de la semana actual,

en la tercera los datos del mes actual y en la cuarta los datos del año actual. Pasemos a comentar cada uno de estos parámetros con mas detenimiento.

- **Entradas y salidas de disco:** El eje  $X$  de la gráfica representa el instante de tiempo en el que se han recogido los datos. El eje  $Y$  de la gráfica representa el número de lecturas de disco si es un valor negativo y el número de escrituras si es un valor positivo. Viendo la gráfica de la figura 6.2 podemos observar que el número de lecturas es notablemente mayor que el número de escrituras.
- **Número de hilos:** El eje  $X$  de la gráfica representa el instante de tiempo en el que se han recogido los datos. El eje  $Y$  de la gráfica representa el número de hilos activos en el servidor. Viendo la gráfica de la figura 6.3 podemos observar que el número de hilos, exceptuando los primero días del año, oscila entre 66 y 74 hilos activos.
- **Interrupciones y cambios de contexto:** El eje  $X$  de la gráfica representa el instante de tiempo en el que se han recogido los datos. El eje  $Y$  de la gráfica representa el número de interrupciones y cambios de contexto producidos por segundo. En color verde están representados los cambios de contexto y en azul las interrupciones. Viendo la gráfica de la figura 6.4 podemos observar que el número de eventos en este mes se mantiene en el mismo intervalo, entre 100 y 160 eventos por segundo. Pero si observamos la gráfica que nos aporta los datos recopilados en el año actual podemos observar que el número de eventos es de orden creciente, ya que en Enero los mínimos eran de 80 interrupciones, en Febrero el número mínimo de cambios de contexto es de 100 y en Noviembre ambos han crecido hasta 120 y 140 respectivamente.
- **Uso de la memoria:** El eje  $X$  de la gráfica representa el instante de tiempo en el que se han recogido los datos. El eje  $Y$  de la gráfica representa el número1 de bytes usados en las operaciones referentes a memoria. Como podemos ver en la figura 6.5 se ha representado con un color cada operación, para poder diferenciarlas visualmente con facilidad. También podemos observar que la mayoría del uso de la memoria se debe a la *memoria virtual comprometida* (119.22M en el momento actual). Lo que menos repercute en el uso de la memoria es la cantidad total de memoria virtual usada, *vmalloc\_used* (508K en el momento actual).

The screenshot shows the Munin configuration interface. On the left, there's a sidebar with 'PROBLEMS' (Critical: 0, Warning: 0, Unknown: 0), 'GROUPS' (munin-monitoring.org, vm, vpn), and 'CATEGORIES' (1sec:network, 1sec:system, active\_inactive, cache, candidates, direct\_map, disk, experimental, free, github, hugepages, kernel, logins, memory, munin, network:services). The main content area shows a tree structure under 'munin-monitoring.org': build.b.munin-monitoring.org (disk, other, processes, system). Under 'disk', there are metrics like Disk IOs per device, Disk latency per device, Disk usage in percent, Inode usage in percent, Throughput per device, and Utilization per device. Under 'system', there are metrics like Available entropy, CPU usage, File table usage, Fork rate, Individual interrupts, Inode table usage, Interrupts and context switches, Load average, Logged in users, Memory usage, Swap in/out, and System uptime.

Figura 6.1: Parte de los parámetros disponibles para monitorizar.

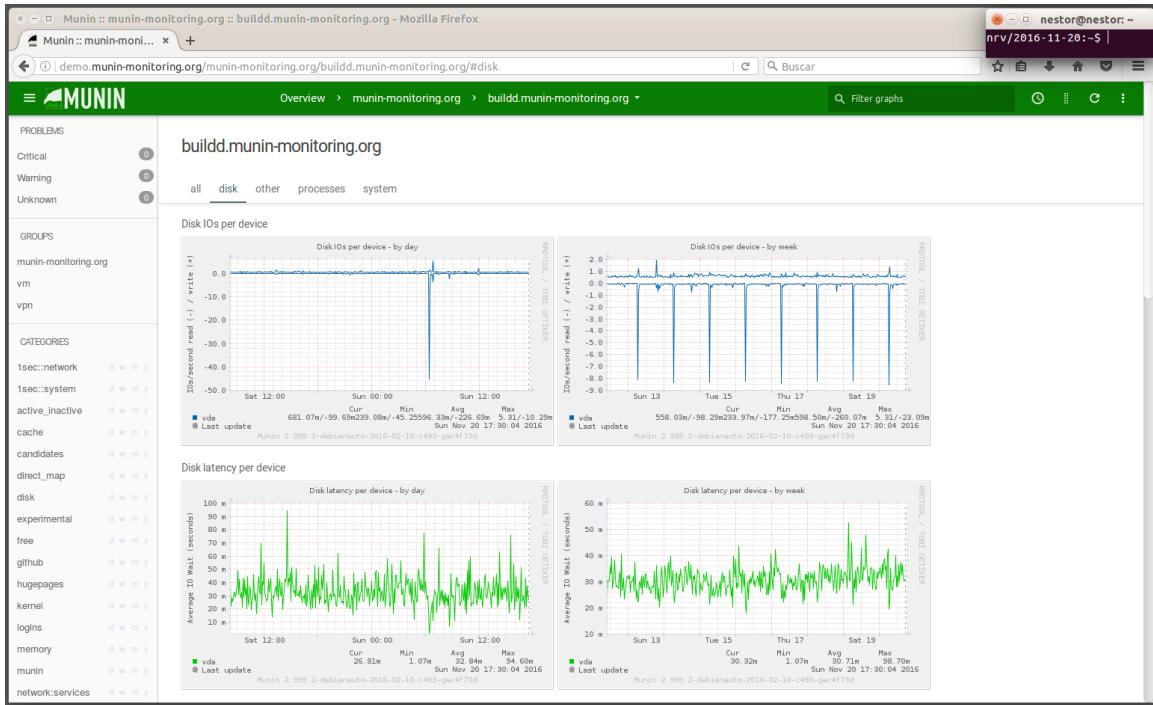


Figura 6.2: Entradas y salidas de disco.

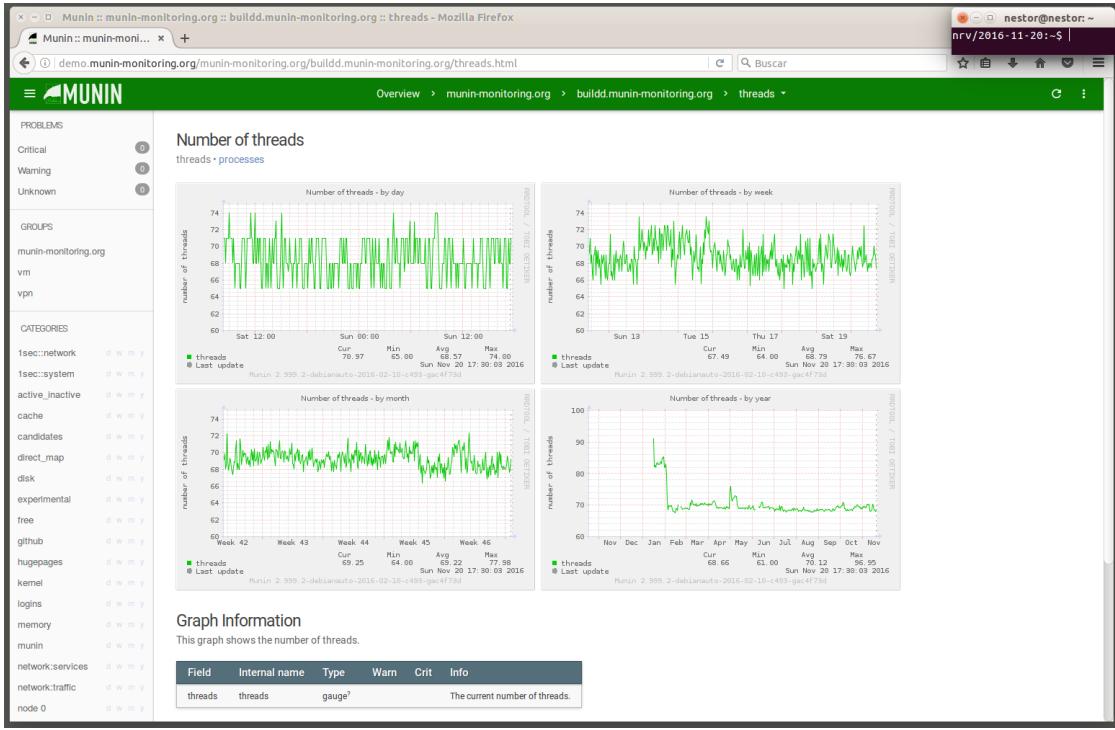


Figura 6.3: Número de hilos.

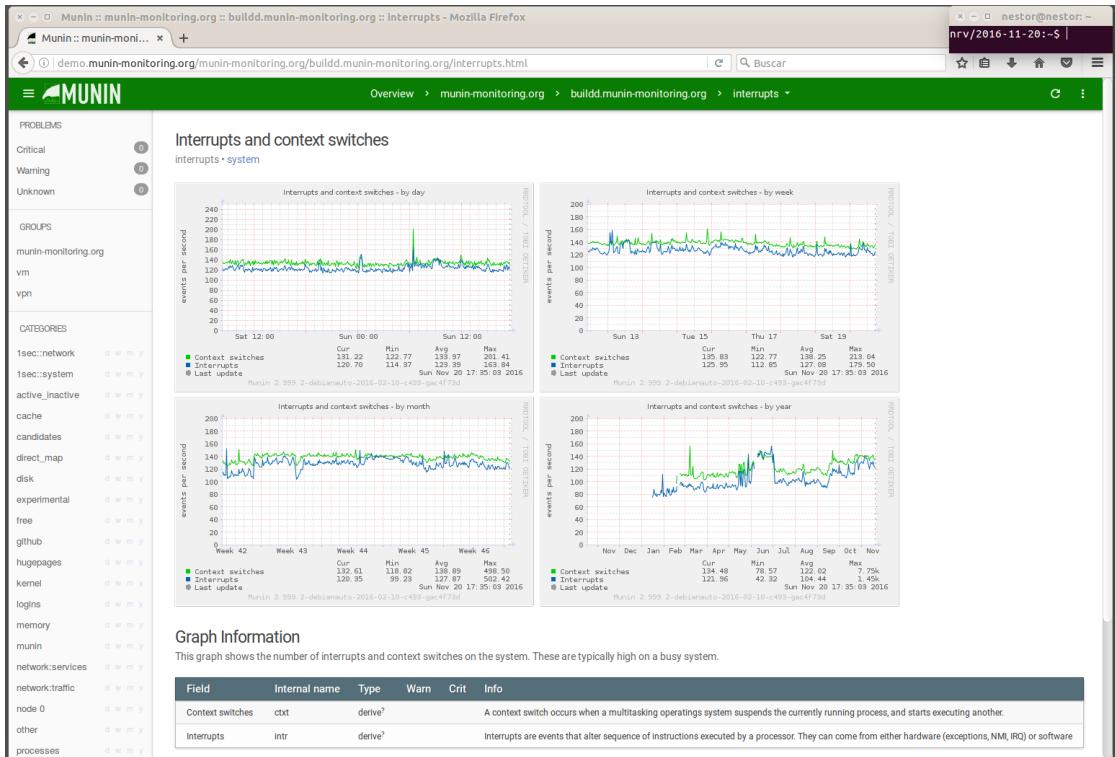


Figura 6.4: Interrupciones y cambios de contexto.

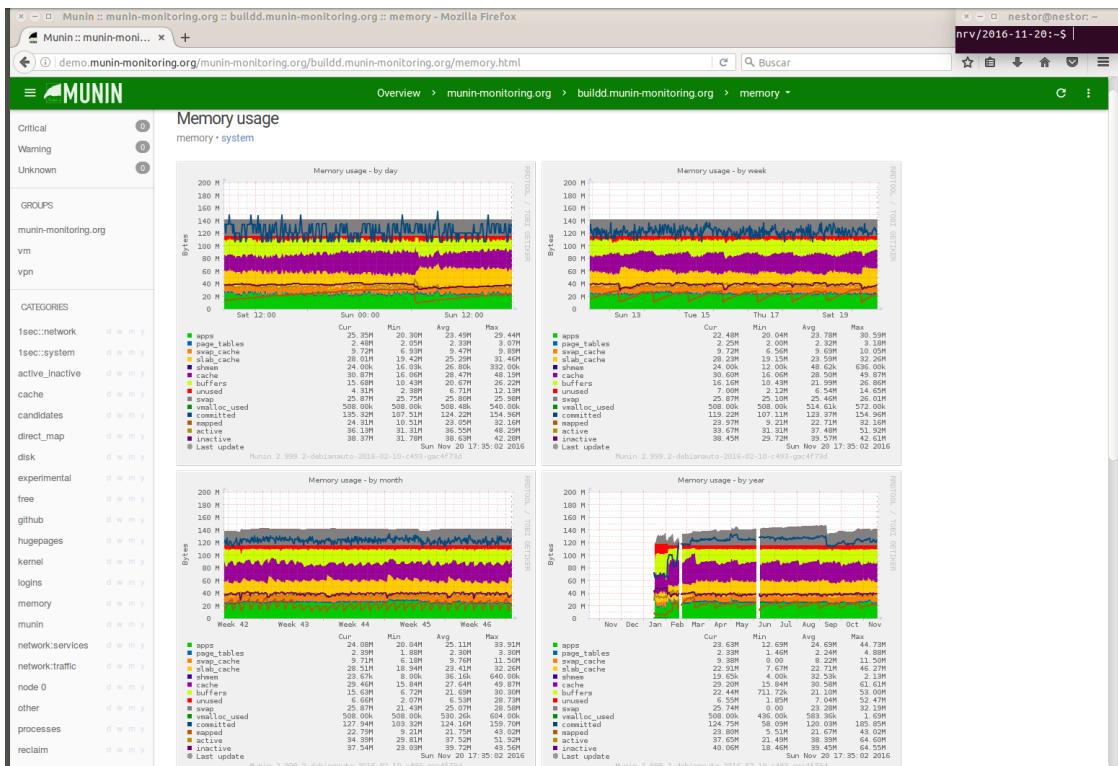


Figura 6.5: Uso de la memoria.

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

El artículo que voy a comentar es uno de los dos que se nos ha dado en el guió de prácticas. El que he elegido es el del blog *Softlayer* [6].

Lo primero a tratar es que es *strace*, como bien aclara el artículo, no es un depurador, es una herramienta para programadores. Se usa para monitorizar las llamadas del sistema y las señales producidas en la ejecución de un programa. El uso de *strace* es bastante sencillo, lo único que hay que hacer es “adjuntar” *strace* a la ejecución de nuestro programa. Por ejemplo, si queremos ver que ocurre cuando se ejecuta *ls* sobre el directorio */home* sólo debemos ejecutar *strace ls /home*. Esto nos mostrará por pantalla todo lo sucedido internamente. Esta aplicación de *strace* puede ser un poco básica, aunque bien sirve como ejemplo didáctico. Lee, escritor del artículo de *Softlayer* [6] nos comenta algunos ejemplos que podría tener *strace* en la vida real. De los dos ejemplos que explica, el que más me ha gustado ha sido el segundo, así que voy a comentarlo con más detenimiento.

*strace* puede ser usado para encontrar errores. Cuando una llamada a una función falla, *strace* devuelve una línea con “ = -1” en la salida. Por ejemplo, intentamos iniciar *Apache* pero este no se inicia y no sabemos porque. Para encontrar el problema, podemos usar *strace* de la siguiente manera;: *strace -Ff -o output.txt -e open /etc/init.d/httpd start*<sup>2</sup>. *Apache* intentara iniciarse y *strace* hará su trabajo. Una vez acabado el intento de inicio, sólo debemos aplicar *grep* ‘= -1’ sobre el fichero al que hemos redirigido la salida de *strace* y ver que ha sucedido. En el blog *Softlayer* tenemos el contenido de dicho fichero. Si nos fijamos en la última línea de ese fragmento, podemos ver lo siguiente: [pid 13748] *open(“/etc/httpd/logs/error\_log”, O\_WRONLY|O\_CREAT|O\_APPEND|O\_LARGEFILE, 0666) = -1 EACCES (Permission denied)*, es decir, no se tiene permiso para abrir el archivo */etc/httpd/logs/error\_log*. Si ahora inspeccionamos los permisos de dicho fichero, podemos ver que tiene el atributo de inmutable, y eso es lo que causa el error.

Como dice Lee, gracias a *strace* podemos encontrar errores que de otro modo no encontrariámos o tardaríamos mucho tiempo en darnos cuenta de a que se debe el error.

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

El script que voy a usar es el script <sup>3</sup> 8.

```
#!/usr/bin/python

suma = 0

for i in range(100):
    if i%2 != 0:
        print i,
        suma += i
    print 'La suma es: %i' % suma
```

La prueba la voy a hacer en Ubuntu Server. El script lo he ubicado en */home/nestor*, como se puede ver en la figura 8.1.

---

<sup>2</sup>El argumento *-Ff* es para seguir todos los subprocessos hijo. El argumento *-o* es para redirigir la salida a un fichero. EL argumento *-e* es para mostrar sólo las llamadas a funciones que se indica tras el argumento, *open* en este caso.

<sup>3</sup>El script “8-profiler.py” se encuentra dentro de la carpeta *Archivos auxiliares*.

```

ISE - US14 [Corriendo] - Oracle VM VirtualBox
nrw/2016-11-21:$ pwd
/home/nestor
nrw/2016-11-21:$ ls
8-profiler.py
nrw/2016-11-21:$ cat 8-profiler.py
#!/usr/bin/python

suma = 0

for i in range(100):
    if i%2 != 0:
        print i,
        suma += i
print 'La suma es: %i' % suma
nrw/2016-11-21:$ -

```

Figura 8.1: Script que imprime los números impares entre 0 y 100 y los suma.

Como bien nos recomienda el ejercicio, el profiler que voy a usar es *CProfile* [7]. Para ejecutar el programa usando el profiler, debemos ejecutar *python -m cProfile 8-profiler.py*. El resultado lo podemos ver en la figura 8.2.

```

nrw/2016-11-21:$ python -m cProfile 8-profiler.py
1 3 5 7 9 11 13 15 17 19 21 23 25 27 29 31 33 35 37 39 41 43 45 47 49 51 53 55 57 59 61 63 65 67 69
71 73 75 77 79 81 83 85 87 89 91 93 95 97 99 La suma es: 2500
    3 function calls in 0.000 seconds

Ordered by: standard name

   ncalls  tottime  percall  cumtime  percall filename:lineno(function)
         1    0.000    0.000    0.000    0.000 8-profiler.py:3(<module>)
         1    0.000    0.000    0.000    0.000 {method 'disable' of '_lsprof.Profiler' objects}
         1    0.000    0.000    0.000    0.000 {range}

nrw/2016-11-21:$

```

Figura 8.2: Resultado del profiler.

La primera información que podemos ver es que se han producido 3 llamadas a funciones y que el programa ha tardado 0.000 segundos. A continuación podemos ver que criterio se ha usado para ordenar las funciones, en este caso se han ordenado por nombre estándar. Finalmente, podemos ver una tabla que tiene 6 columnas y 3 filas (una fila por cada función que ha sido llamada). El significado de las columnas es [7]:

- **ncalls:** Número de veces que se llama a la función. En mi caso todas las funciones han sido llamadas una única vez.
- **tottime:** Tiempo total de la función excluyendo llamadas a subfunciones.
- **percall:** Cociente de *tottime* entre *ncalls*.
- **cumtime:** Tiempo acumulado en la función y todas las subfunciones.
- **percall:** Cociente de *cumtime* entre las llamadas primitivas.

- **filename:lineno(function)**: Información sobre la función, como el archivo donde se encuentra, la línea donde está y el nombre de la función.

## 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).

Como bien pone en el guión, para aprender a usar el profiler de MySQL me he fijado en las dos secciones que hablan acerca de ello en la página de MySQL [8] [9]. Yo lo voy a hacer con phpMyAdmin, así que lo primero que hago es conectarme a mi servidor y entrar a phpMyAdmin. Para ello escribimos en el buscador de mi máquina local la dirección IP de mi servidor seguido de `/phpmyadmin`. En mi caso debemos escribir `192.168.56.103/phpmyadmin`, como podemos ver en la figura 9.1.

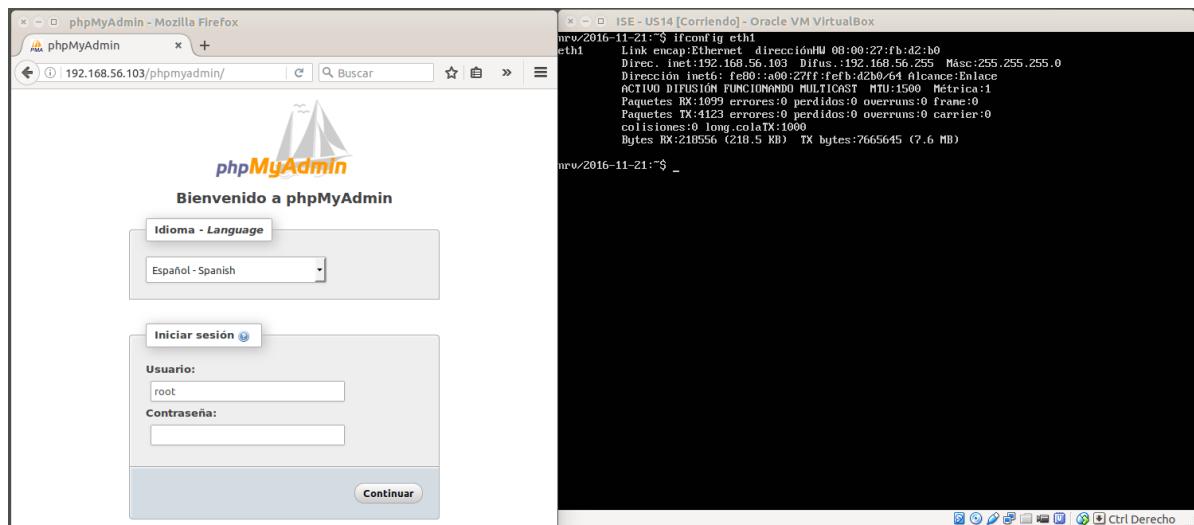


Figura 9.1: Conexión con phpMyAdmin (máquinas conectadas en modo *host-only*).

Una vez nos conectamos, abrimos una consola de SQL y ejecutamos las siguientes sentencias, como se puede ver en la figura 9.2:

- CREATE DATABASE EJ9;
- USE EJ9;
- CREATE TABLE tabla (dato1 INT, dato2 INT);
- SET profiling=1;

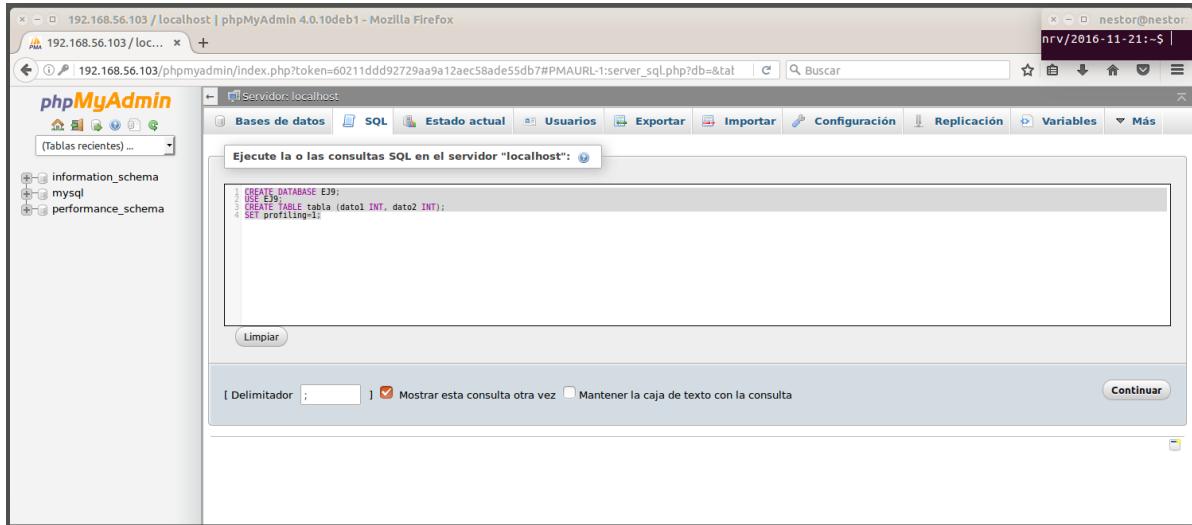


Figura 9.2: Preparativos para el profiling.

Como podemos ver en la figura 9.3, la primera sentencia que ejecutamos ha creado la base de datos *EJ9*, la segunda ha indicado que vamos a usar la base de datos *EJ9*, la tercera sentencia sirve para crear una tabla llamada *tabla* con dos columnas, *dato1* que es un entero y *dato2* que es un entero. La última sentencia es para habilitar el proceso de *profiling*.

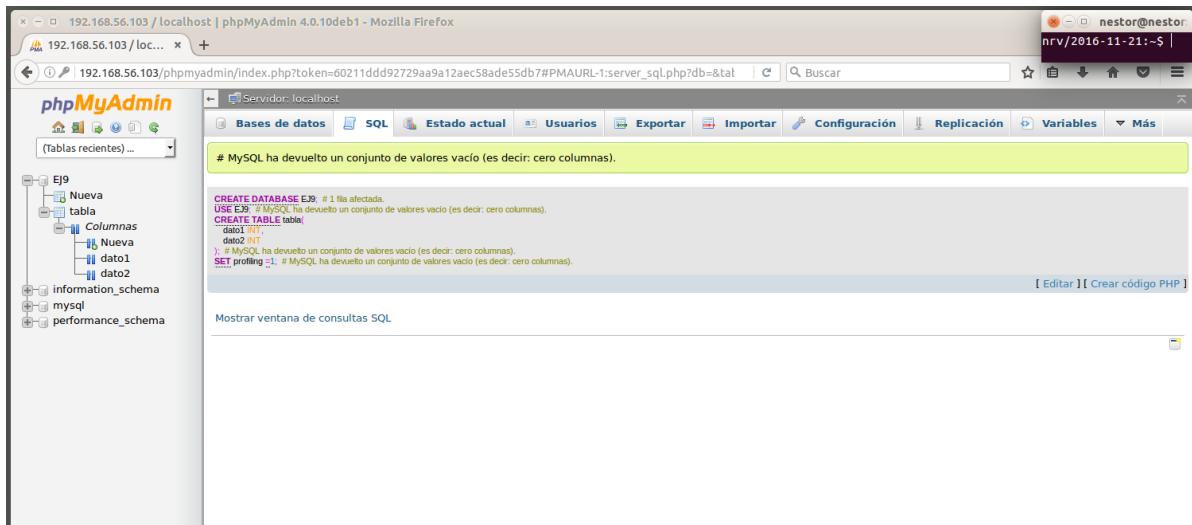


Figura 9.3: Sentencias ejecutadas.

Finalmente, insertamos algunas tuplas en la tabla, como podemos ver en la figura 9.4.

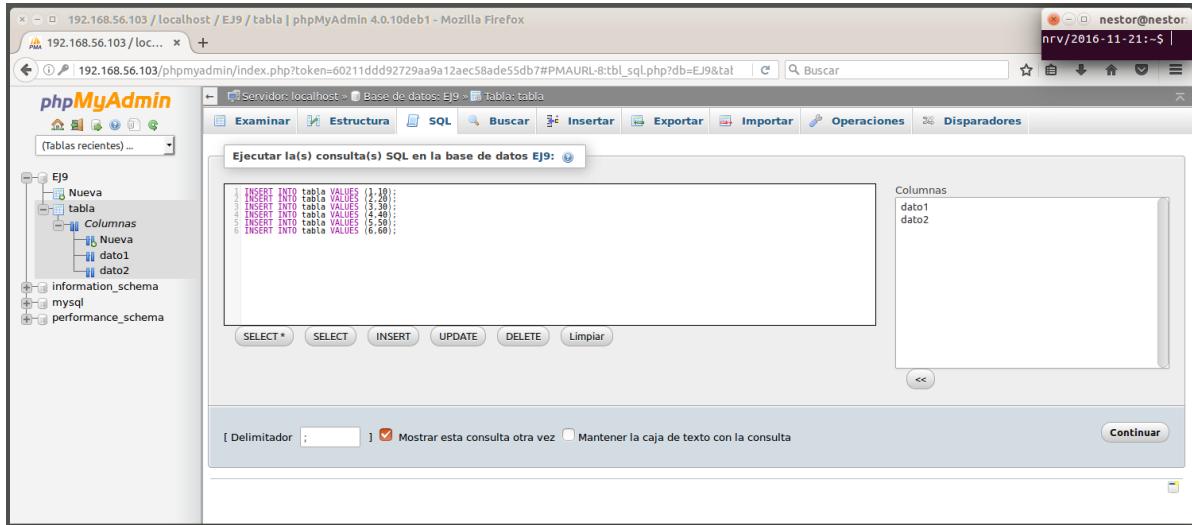


Figura 9.4: Inserción de tuplas.

Una vez tenemos las tuplas insertadas, realizamos una consulta. En mi caso, he recuperado todas las tuplas de la tabla ejecutando la sentencia *SELECT \* FROM tabla LIMIT 0,30*. El resultado lo podemos ver en la figura 9.5.

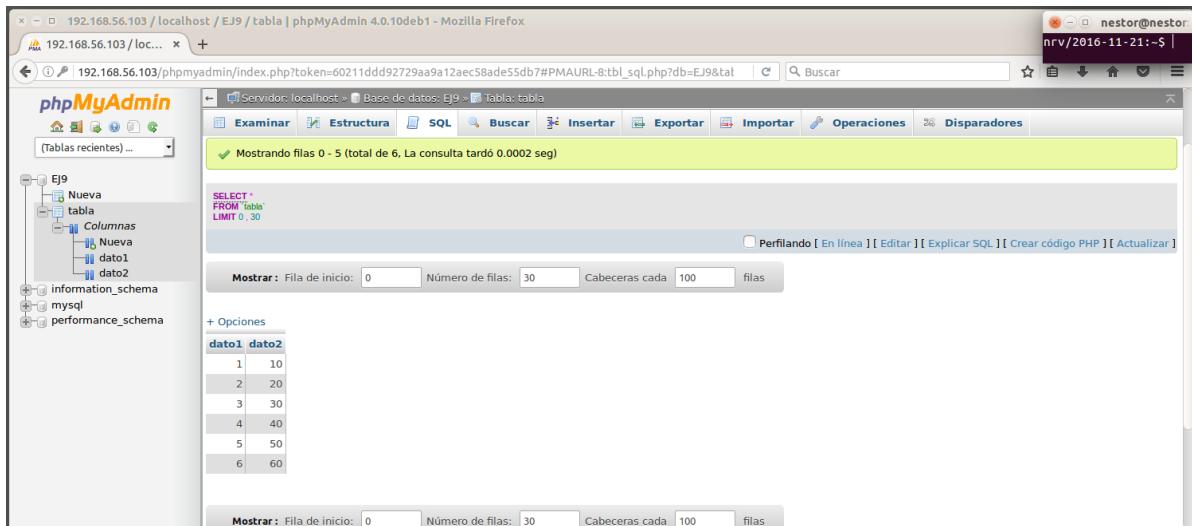


Figura 9.5: Recuperación de todas las tuplas de la tabla.

Tras realizar la consulta, marcamos la opción *Perfilando* y volvemos a ejecutar la consulta. En este caso, el resultado que obtenemos es el que podemos ver en la figura 9.6.

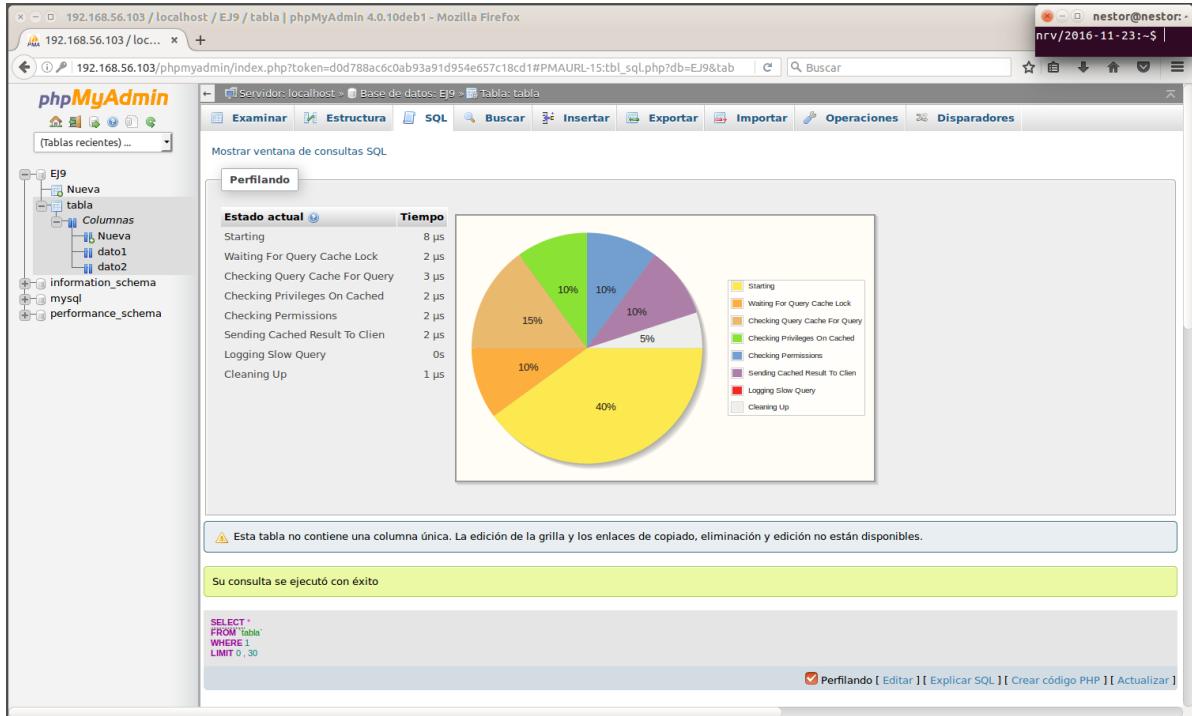


Figura 9.6: Datos recogidos por el profiler tras la consulta.

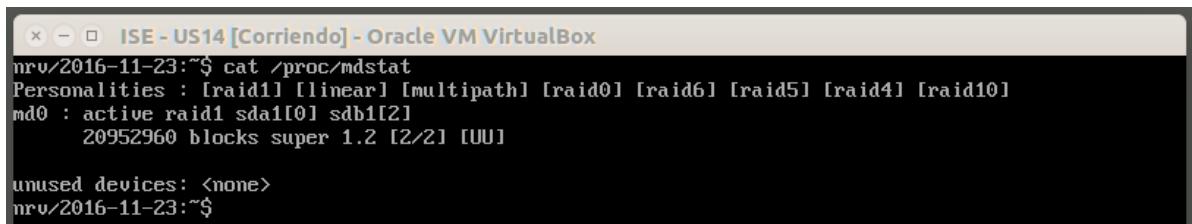
El significado de las secciones que aparecen en los datos del profiler son [10]:

- **Starting:** Tiempo empleado en comenzar.
- **Waiting For Query Cache Lock:** Tiempo esperando a conseguir un cerrojo sobre la cache de la consulta.
- **Checking Query Cache For Query:** Tiempo que tarda el servidor en comprobar si la consulta actual está presente en la caché de consultas.
- **Checking Privileges On Cached Query:** Tiempo que tarda el servidor en comprobar si el usuario tiene privilegios para acceder al resultado de una consulta ubicada en caché.
- **Checking Permissions:** Tiempo empleado en comprobar que los permisos son correctos.
- **Sending Cached Result To Client:** Tiempo que emplea el servidor en coger el resultado de una consulta en caché y enviarlo al cliente.
- **Logging Slow Query:** Tiempo empleado en loguearse en la cola lenta.
- **Cleaning up:** Tiempo empleado en limpiar los datos tras la consulta.

## 10. Cuestión opcional 1: Indique qué comandos ha utilizado para realizarlo así como capturas de pantalla del proceso de reconstrucción del RAID.

Lo que he hecho ha sido seguir la misma idea que en la cuestión opcional 1 de la práctica 1 de esta asignatura. Al igual que en la práctica 1, me he basado en la información que podemos obtener de la Wiki de Linux [11]. Los pasos que he seguido son los siguientes:

1. Primero comprobamos que efectivamente tenemos dos RAID. Para ello ejecutamos el comando `cat /proc/mdstat` como se puede ver en la figura 10.1. También se podría haber hecho con `watch -n2 cat /proc/mdstat` pero como no se producen cambios en el fichero, no le sacamos partido a la utilidad `watch`.
2. A continuación, producimos un fallo por software en el RAID ejecutando el comando `sudo mdadm --manage --set-faulty /dev/md0 /dev/sdb1` como podemos ver en la figura 10.2.
3. Retiramos el RAID en caliente ejecutando `sudo mdadm /dev/md0 -r /dev/sdb1` y lo añadimos ejecutando `sudo mdadm /dev/md0 -a /dev/sdb1`, como podemos ver en la figura 10.3.
4. Ahora si podemos sacarle partido a `watch`. Para comprobar el proceso de recuperación del RAID ejecutamos `watch -n2 cat /proc/mdstat`. De este modo, cada dos segundos se ejecutará `cat /proc/mdstat`, proporcionándonos así una idea de como va el progreso, como se puede ver en la figura 10.4.
5. Tras un tiempo de recuperación, el RAID se ha recuperado correctamente y vuelve a estar en funcionamiento, como podemos ver en la figura 10.5.



```
ISE - US14 [Corriendo] - Oracle VM VirtualBox
nrv/2016-11-23:~$ cat /proc/mdstat
Personalities : [raid1] [linear] [multipath] [raid0] [raid6] [raid5] [raid4] [raid10]
md0 : active raid1 sda1[0] sdb1[2]
      20952960 blocks super 1.2 [2/2] [UU]

unused devices: <none>
nrv/2016-11-23:~$
```

Figura 10.1: Comprobamos que tenemos dos RAID.

```

nrw/2016-11-23:~$ sudo mdadm --manage --set-faulty /dev/md0 /dev/sdb1
[sudo] password for nestor:
[ 1093.933066] md/raid1:md0: Disk failure on sdb1, disabling device.
[ 1093.933066] md/raid1:md0: Operation continuing on 1 devices.
mdadm: set /dev/sdb1 faulty in /dev/md0
nrw/2016-11-23:~$ cat /proc/mdstat
Personalities : [raid1] [linear] [multipath] [raid0] [raid6] [raid5] [raid4] [raid10]
md0 : active raid1 sda1[0] sdb1[2](F)
      20952960 blocks super 1.2 [2/1] [U_]

unused devices: <none>
nrw/2016-11-23:~$ _

```

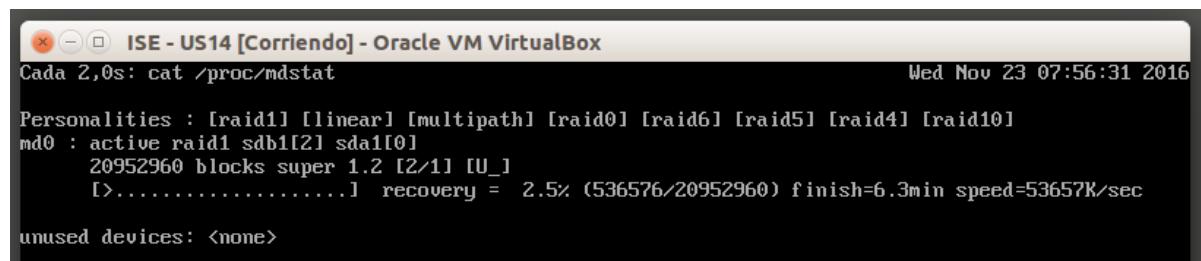
Figura 10.2: Provocamos un fallo software en el RAID.

```

nrw/2016-11-23:~$ 
nrw/2016-11-23:~$ sudo mdadm /dev/md0 -r /dev/sdb1
mdadm: hot removed /dev/sdb1 from /dev/md0
nrw/2016-11-23:~$ sudo mdadm /dev/md0 -a /dev/sdb1
mdadm: added /dev/sdb1
nrw/2016-11-23:~$ 

```

Figura 10.3: Quitamos en caliente y añadimos de nuevo el RAID.



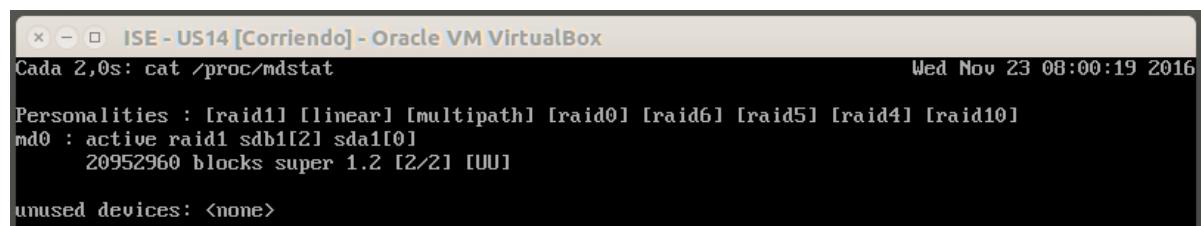
```

ISE - US14 [Corriendo] - Oracle VM VirtualBox
Cada 2,0s: cat /proc/mdstat                               Wed Nov 23 07:56:31 2016
Personalities : [raid1] [linear] [multipath] [raid0] [raid6] [raid5] [raid4] [raid10]
md0 : active raid1 sdb1[2] sda1[0]
      20952960 blocks super 1.2 [2/1] [U_]
      [>.....] recovery =  2.5% (536576/20952960) finish=6.3min speed=53657K/sec

unused devices: <none>

```

Figura 10.4: Observamos el proceso de recuperación con *watch*.



```

ISE - US14 [Corriendo] - Oracle VM VirtualBox
Cada 2,0s: cat /proc/mdstat                               Wed Nov 23 08:00:19 2016
Personalities : [raid1] [linear] [multipath] [raid0] [raid6] [raid5] [raid4] [raid10]
md0 : active raid1 sdb1[2] sda1[0]
      20952960 blocks super 1.2 [2/2] [UU]

unused devices: <none>

```

Figura 10.5: RAID completamente restaurado.

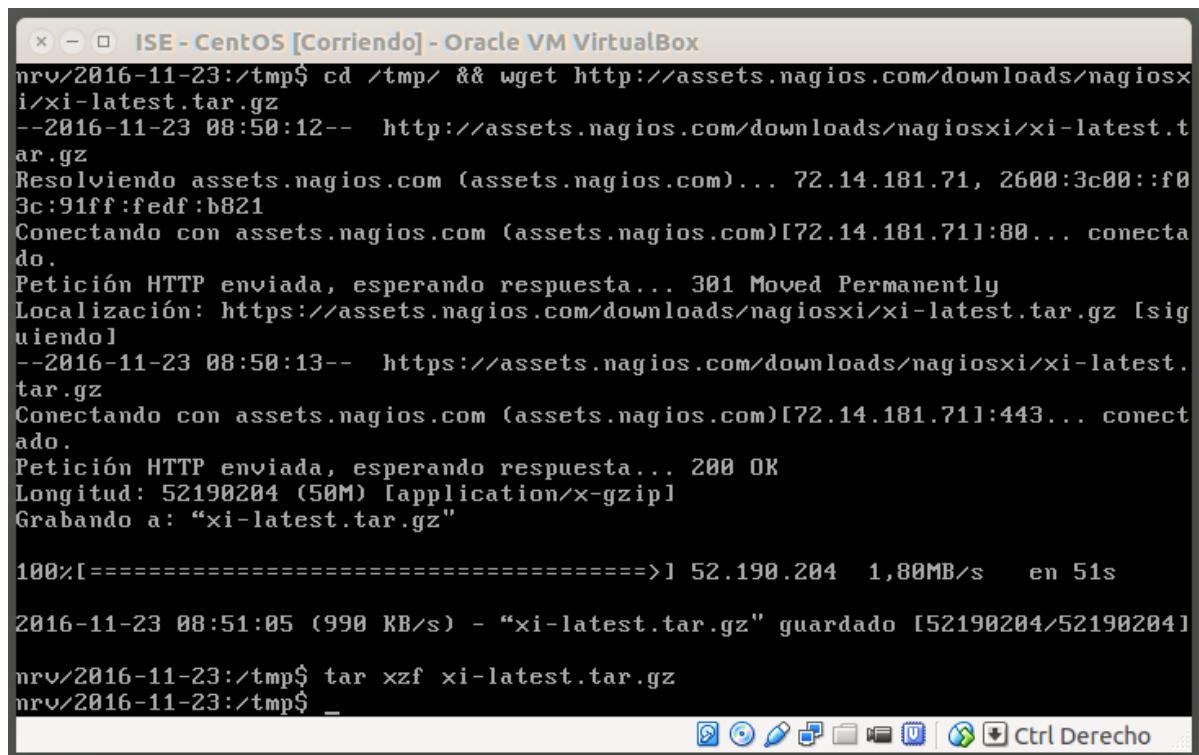
## **11. Cuestión opcional 2: Instale Nagios en su sistema (el que prefiera) documentando el proceso y muestre el resultado de la monitorización de su sistema comentando qué aparece.**

El proceso de instalación lo podemos ver en la página oficial de Nagios [12]. Yo lo voy a hacer en CentOS. Los pasos a seguir son:

1. El proceso de instalacion se hace desde el directorio `/tmp`, así que ejecutamos `cd /tmp` para irnos a dicha carpeta, como podemos ver en la figura 11.1.
2. Descargamos la última versión de *Nagios*. Para ello ejecutamos el comando `wget http://assets.nagios.com/downloads/nagiosxi/xi-latest.tar.gz`, como podemos ver en la figura 11.1.
3. Descomprimimos Nagios ejecutando `tar xzf xi-latest.tar.gz`, como podemos ver en la figura 11.1.
4. Nos vamos a la carpeta que se ha creado ejecutando `cd /tmp/nagiosxi`, como podemos ver en la figura 11.2.
5. Ejecutamos el script de instalación con el comando `sudo ./fullinstalation4`, como podemos ver en la figura 11.2. Nos preguntara si queremos continuar, le decimos que sí pulsando la tecla *y*.

---

<sup>4</sup>Cuidado: este script cambia la contraseña del usuario root de MySQL a *nagiosxi*



```

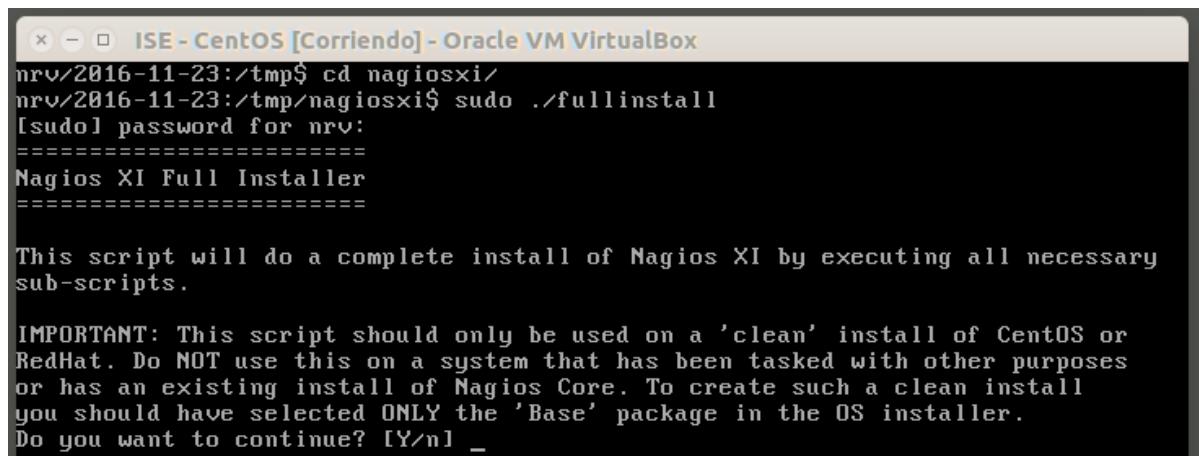
ISE - CentOS [Corriendo] - Oracle VM VirtualBox
nrv/2016-11-23:/tmp$ cd /tmp/ && wget http://assets.nagios.com/downloads/nagiosxi/.../xi-latest.tar.gz
--2016-11-23 08:50:12-- http://assets.nagios.com/downloads/nagiosxi/.../xi-latest.tar.gz
Resolviendo assets.nagios.com (assets.nagios.com)... 72.14.181.71, 2600:3c00::f0
3c:91ff:fedf:b821
Conectando con assets.nagios.com (assets.nagios.com)[72.14.181.71]:80...
Petición HTTP enviada, esperando respuesta... 301 Moved Permanently
Localización: https://assets.nagios.com/downloads/nagiosxi/.../xi-latest.tar.gz [siguiendo]
--2016-11-23 08:50:13-- https://assets.nagios.com/downloads/nagiosxi/.../xi-latest.tar.gz
Conectando con assets.nagios.com (assets.nagios.com)[72.14.181.71]:443...
Petición HTTP enviada, esperando respuesta... 200 OK
Longitud: 52190204 (50M) [application/x-gzip]
Grabando a: "xi-latest.tar.gz"

100%[=====] 52.190.204 1,80MB/s en 51s
2016-11-23 08:51:05 (990 KB/s) - "xi-latest.tar.gz" guardado [52190204/52190204]

nrv/2016-11-23:/tmp$ tar xzf xi-latest.tar.gz
nrv/2016-11-23:/tmp$ _

```

Figura 11.1: Proceso de descarga de Nagios y preparación para la instalación.



```

ISE - CentOS [Corriendo] - Oracle VM VirtualBox
nrv/2016-11-23:/tmp$ cd nagiosxi/
nrv/2016-11-23:/tmp/nagiosxi$ sudo ./fullinstall
[sudo] password for nrv:
=====
Nagios XI Full Installer
=====

This script will do a complete install of Nagios XI by executing all necessary
sub-scripts.

IMPORTANT: This script should only be used on a 'clean' install of CentOS or
RedHat. Do NOT use this on a system that has been tasked with other purposes
or has an existing install of Nagios Core. To create such a clean install
you should have selected ONLY the 'Base' package in the OS installer.
Do you want to continue? [Y/n] _

```

Figura 11.2: Proceso de instalación de Nagios.

Una vez hemos instalado Nagios, en el manual de instalación [12] podemos ver que para acceder a Nagios desde mi máquina anfitriona debemos poner en el buscador *http://IP/nagios*, donde *IP* es la dirección del servidor al que nos queremos conectar. En mi caso, la dirección IP es *192.168.56.101*, como podemos ver en la figura 11.3. Tras introducir la dirección en el navegador, podemos ver que nos conectamos correctamente a Nagios, como se puede ver en la figura 11.3.

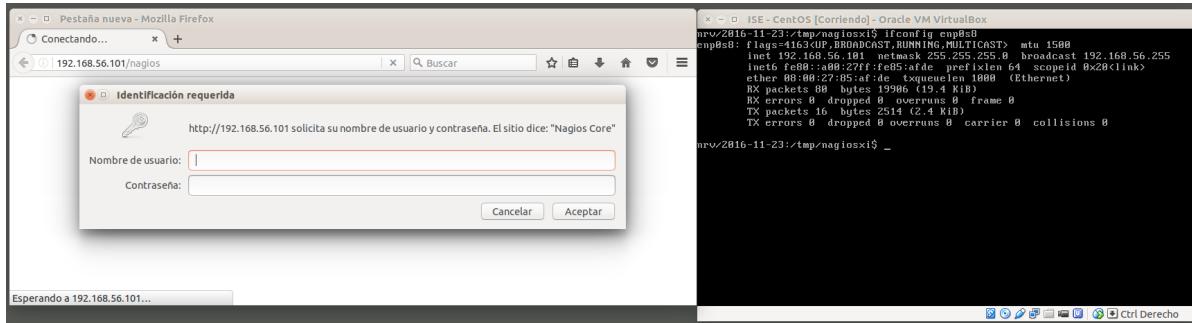


Figura 11.3: Conexión con Nagios (máquina conectadas en modo *host-only*).

Vemos que nos pide una contraseña. Para poder acceder debemos seguir los pasos que nos indica Nagios en su guía para configurar la interfaz web [13]. Debemos acceder a la dirección de nuestro servidor, como podemos ver en la figura 11.4. Una vez en dicha página, le damos a acceder y ahí configuraremos nuestros datos, como podemos ver en la figura 11.5. A continuación, pulsamos *install*.

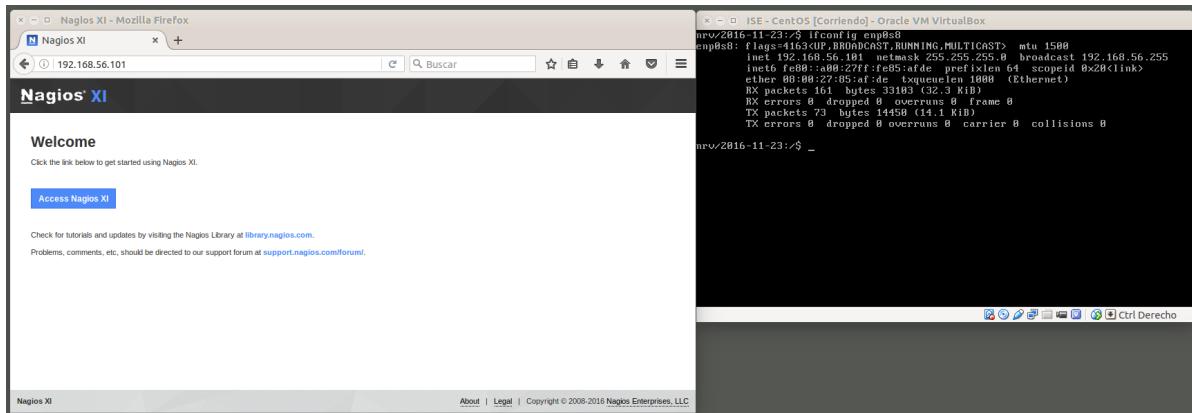


Figura 11.4: Paso 1 de la configuración de la interfaz web (máquina conectadas en modo *host-only*).

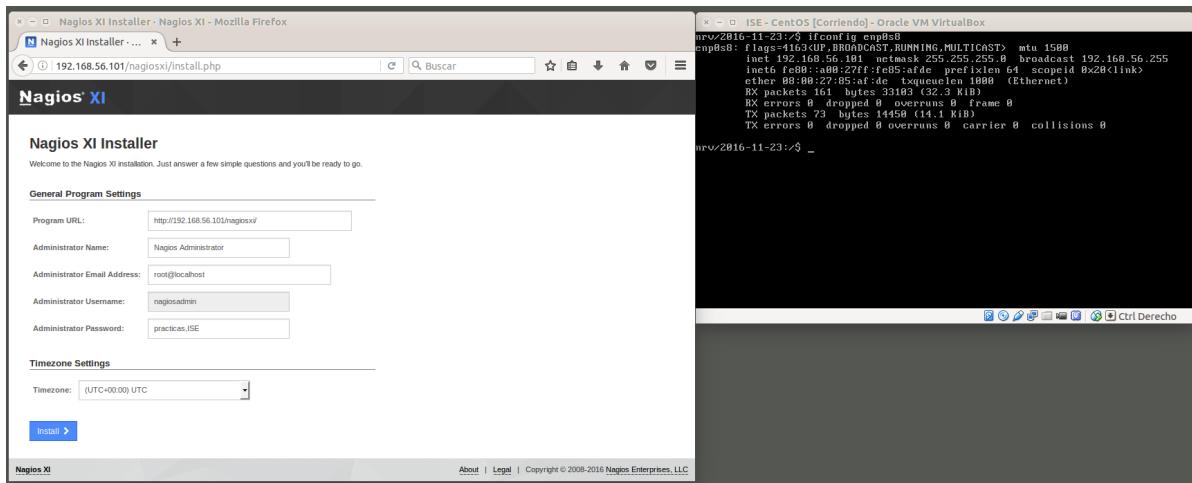


Figura 11.5: Paso 2 de la configuración de la interfaz web (máquina conectadas en modo *host-only*).

Ahora si podemos acceder usando los datos que hemos introducido en el paso anterior (ver figura 11.5), como podemos ver en la figura 11.6.

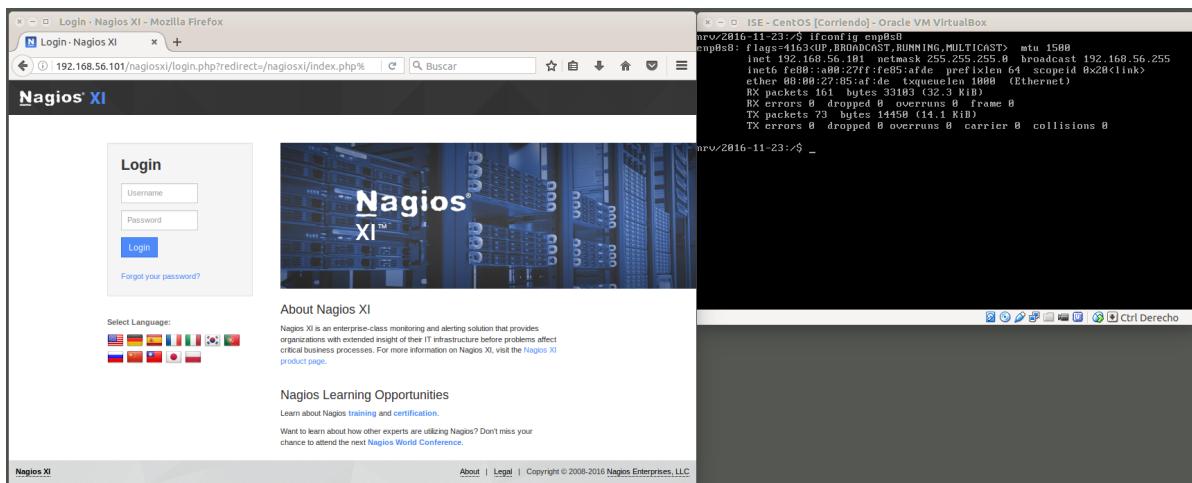


Figura 11.6: Conexión con Nagios (máquina conectadas en modo *host-only*).

Una vez hemos accedido a Nagios podemos ver que se pueden monitorizar diferentes parámetros. Como la mayoría de monitores, nos permite monitorizar el rendimiento de nuestro servidor, como podemos ver en la figura 11.7. Una de las características que más me ha llamado la atención es la llamada *Hypermap*. Esta opción nos muestra un mapa con el estado actual de los dispositivos de red (hosts), como podemos ver en la figura 11.8.

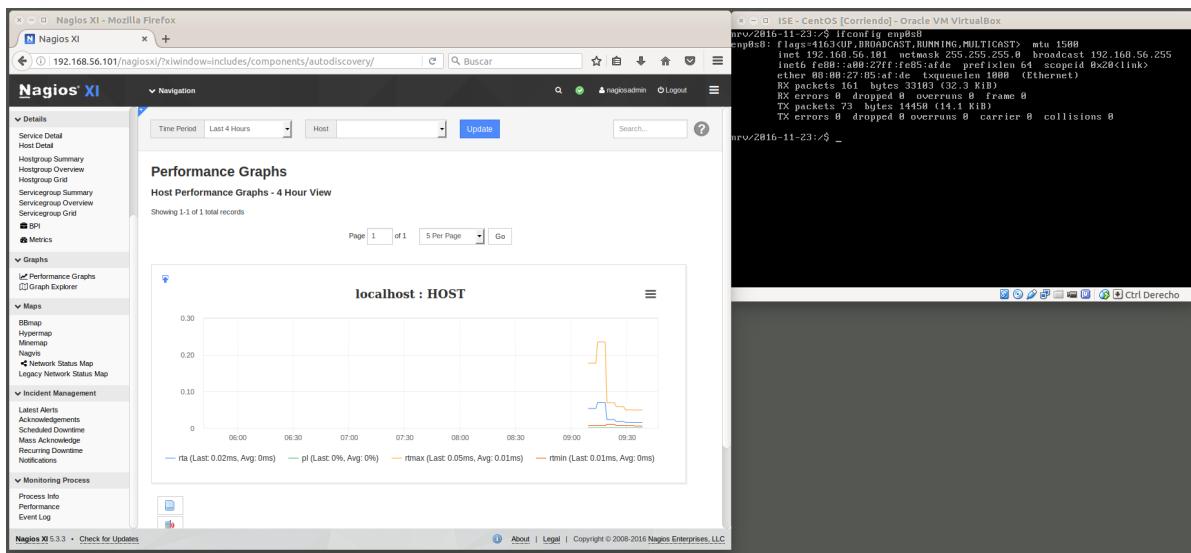


Figura 11.7: Monitorización del rendimiento con Nagios (máquina conectadas en modo *host-only*).

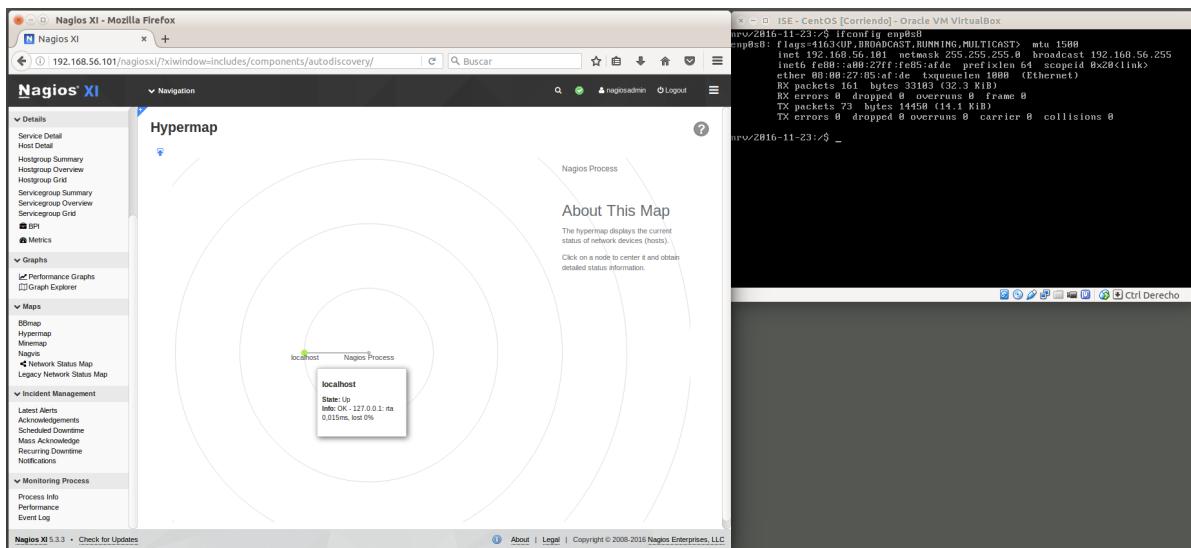


Figura 11.8: Hypermap de mi servidor (máquina conectadas en modo *host-only*).

## **12. Cuestión opcional 4: Pruebe a instalar este monitor en alguno de sus tres sistemas. Realice capturas de pantalla del proceso de instalación y comente capturas de pantalla del programa en ejecución.**

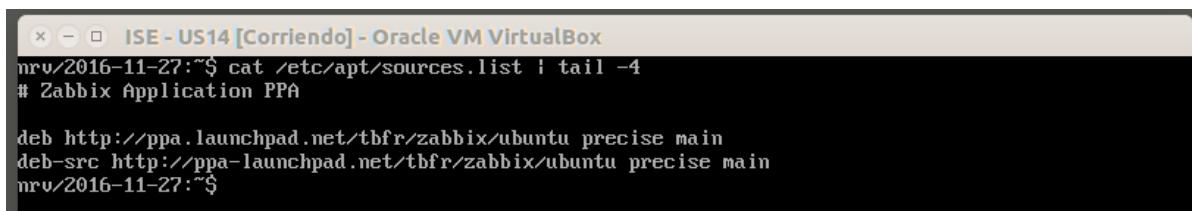
Voy a instalar *Zabbix* en Ubuntu Server siguiendo los pasos que nos indica Digital Ocean [14]. Los pasos a seguir para la instalación son:

1. *Zabbix* se encuentra en los repositorios de Ubuntu, pero está desactualizado, por ello vamos a instalarlo desde los repositorios del programa. Para ello, editamos el fichero */etc/apt/sources.list* y añadimos las líneas:

```
# Zabbix Application PPA
deb http://ppa.launchpad.net/tbfr/zabbix/ubuntu precise main
deb-src http://ppa.launchpad.net/tbfr/zabbix/ubuntu precise main
```

El archivo quedaría como podemos ver en la figura 12.1.

2. A continuación añadimos la clave del PPA para que *apt* confíe en la fuente. Para ello ejecutamos: *sudo apt-key adv --keyserver keyserver.ubuntu.com --recv-keys C407E17D5F76A32B*, como podemos ver en la figura 12.2.
3. Una vez hecho esto, actualizamos los repositorios con *sudo apt-get update* e instalamos *Zabbix* ejecutando *zsudo apt-get install zabbix-server-mysql php5-mysql zabbix-frontend-php*, como podemos ver en la figura 12.2.



```
ISE - US14 [Corriendo] - Oracle VM VirtualBox
mrw/2016-11-27:~$ cat /etc/apt/sources.list | tail -4
# Zabbix Application PPA

deb http://ppa.launchpad.net/tbfr/zabbix/ubuntu precise main
deb-src http://ppa.launchpad.net/tbfr/zabbix/ubuntu precise main
mrw/2016-11-27:~$
```

Figura 12.1: Archivo */etc/apt/sources.list*.

```

ISE - US14 [Corriendo] - Oracle VM VirtualBox
mrw/2016-11-27:~$ sudo apt-key adv --keyserver keyserver.ubuntu.com --recv-keys C407E17D5F76A32B
Executing: gpg --ignore-time-conflict --no-options --no-default-keyring --homedir /tmp/tmp.IhEAj3gDR
R --no-auto-check-trustdb --trust-model always --keyring /etc/apt/trusted.gpg --primary-keyring /etc
/apt/trusted.gpg --keyserver keyserver.ubuntu.com --recv-keys C407E17D5F76A32B
gpg: solicitando clave 5F76A32B de hkp servidor keyserver.ubuntu.com
gpg: clave pública "Launchpad PPA for Tobias Frederick" importada
gpg: Cantidad total procesada: 1
gpg: importadas: 1 (RSA: 1)
mrw/2016-11-27:~$ sudo apt-get update && sudo apt-get install zabbix-server-mysql php5-mysql zabbix-
frontend-php

```

Figura 12.2: Instalación de *Zabbix*.

Los siguiente que debemos hacer es configurar el server de *Zabbix*, para ello editamos el archivo */etc/zabbix/zabbix\_server.conf*, y cambiamos el valor de *DBName*, *DBUser* y *DBPassword*. Dichos parámetros han quedado como podemos ver en la figura 12.3.

```

ISE - US14 [Corriendo] - Oracle VM VirtualBox
mrw/2016-11-27:~$ sudo cat /etc/zabbix/zabbix_server.conf | grep DBName=zabbix
DBName=zabbix
mrw/2016-11-27:~$ sudo cat /etc/zabbix/zabbix_server.conf | grep DBUser=zabbix
DBUser=zabbix
mrw/2016-11-27:~$ sudo cat /etc/zabbix/zabbix_server.conf | grep DBPassword=pra
DBPassword=practicas,ISE
mrw/2016-11-27:~$ _

```

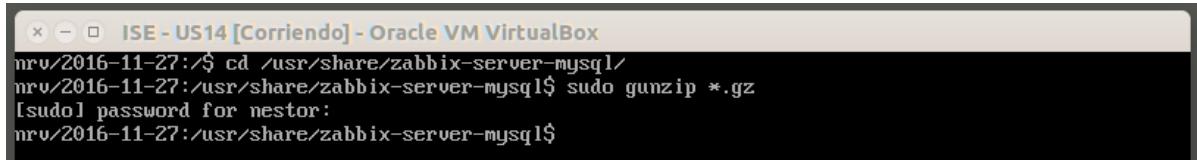
Figura 12.3: Configuración del servidor *Zabbix*.

A continuación configuraremos MySQL. Para ello seguimos los siguientes pasos:

1. Extramos los ficheros SQL del directorio *cd /usr/share/zabbix-server-mysql/* ejecutando *sudo gunzip \*.gz*, como podemos ver en la figura 12.4.
2. Entramos a MySQL como usuarios root ejecutando *mysql -u root -p*, como podemos ver en la figura 12.5.
3. Creamos un usuario para *Zabbix* que coincida con los datos que escribimos en el fichero */etc/zabbix/zabbix\_server.conf* ejecutando *create user 'zabbix'@'localhost' identified by 'practicas,ISE';*, como podemos ver en la figura 12.5.
4. Creamos una base de datos para *Zabbix* ejecutando *create database zabbix;*, como podemos ver en la figura 12.5.
5. Le damos permisos sobre dicha base de datos al usuario que hemos creado con anterioridad ejecutando *grant all privileges on zabbix.\* to 'zabbix'@'localhost';*, como podemos ver en la figura 12.5.
6. Actualizamos los permisos ejecutando *flush privileges;*, como podemos ver en la figura 12.5.
7. Salimos de MySQL ejecutando *exit;*, como podemos ver en la figura 12.5.

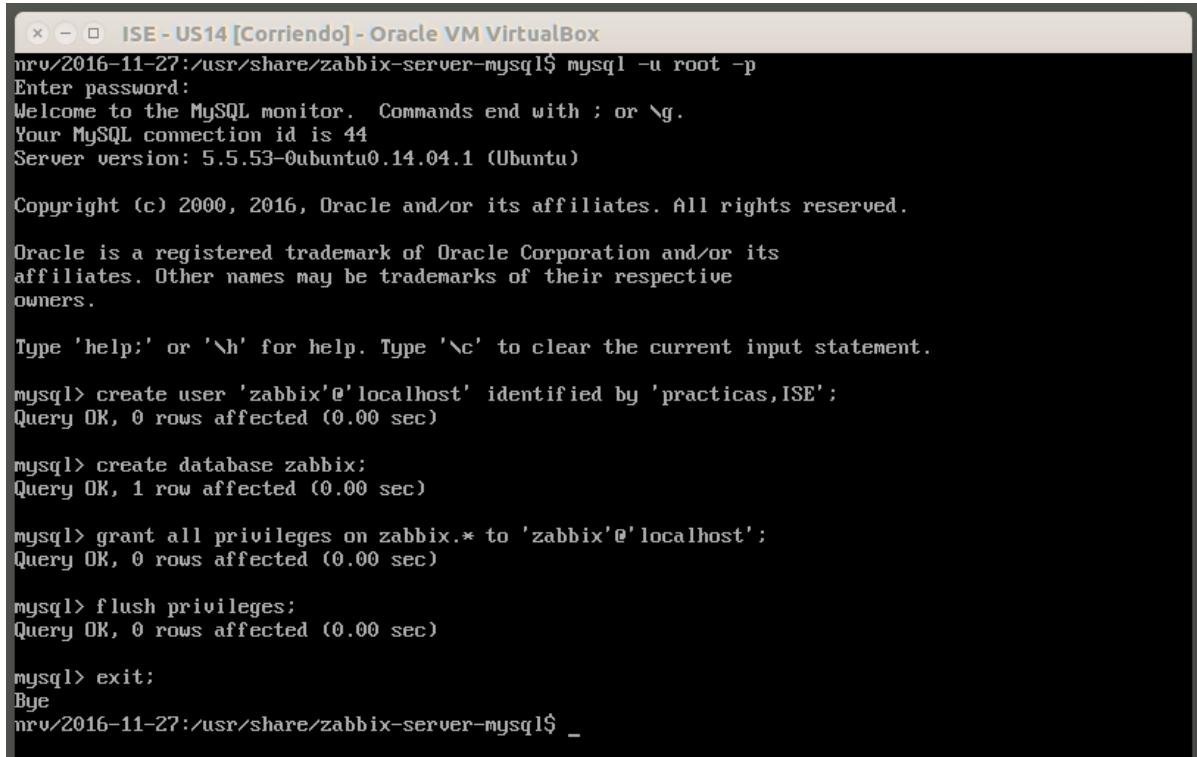
8. Importamos los archivos que *Zabbix* necesita para funcionar, para ello y como podemos ver en la figura 12.6, ejecutamos:

- `mysql -u zabbix -p zabbix < schema.sql`
- `mysql -u zabbix -p zabbix < images.sql`
- `mysql -u zabbix -p zabbix < data.sql`



```
mrw/2016-11-27:/usr/share/zabbix-server-mysql$ sudo gunzip *.gz
[sudo] password for nestor:
mrw/2016-11-27:/usr/share/zabbix-server-mysql$
```

Figura 12.4: Extracción de ficheros SQL.



```
mrw/2016-11-27:/usr/share/zabbix-server-mysql$ mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 44
Server version: 5.5.53-0ubuntu0.14.04.1 (Ubuntu)

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

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

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

mysql> create user 'zabbix'@'localhost' identified by 'practicas,ISE';
Query OK, 0 rows affected (0.00 sec)

mysql> create database zabbix;
Query OK, 1 row affected (0.00 sec)

mysql> grant all privileges on zabbix.* to 'zabbix'@'localhost';
Query OK, 0 rows affected (0.00 sec)

mysql> flush privileges;
Query OK, 0 rows affected (0.00 sec)

mysql> exit;
Bye
mrw/2016-11-27:/usr/share/zabbix-server-mysql$ _
```

Figura 12.5: Configuración de MySQL.

```

nrv/2016-11-27:/usr/share/zabbix-server-mysql$ mysql -u zabbix -p zabbix < schema.sql
Enter password:
nrv/2016-11-27:/usr/share/zabbix-server-mysql$ mysql -u zabbix -p zabbix < images.sql
Enter password:
nrv/2016-11-27:/usr/share/zabbix-server-mysql$ mysql -u zabbix -p zabbix < data.sql
Enter password:
nrv/2016-11-27:/usr/share/zabbix-server-mysql$ _

```

Figura 12.6: Importación de archivos necesarios para *Zabbix*.

Una vez hemos configurado MySQL, pasamos a la configuración de PHP. Para ello editamos el fichero */etc/php5/apache2/php.ini*. Buscamos y modificamos los siguientes parámetros para que queden como a continuación:

- `post_max_size = 16M`
- `max_execution_time = 300`
- `max_input_time = 300`
- `date.timezone = UTC`

El resultado lo podemos ver en la figura 12.7.

```

nrv/2016-11-27:$ sudo cat /etc/php5/apache2/php.ini | grep post_max_size
post_max_size = 16M
nrv/2016-11-27:$ sudo cat /etc/php5/apache2/php.ini | grep max_execution_time
max_execution_time = 300
nrv/2016-11-27:$ sudo cat /etc/php5/apache2/php.ini | grep 'max_input_time = 300'
max_input_time = 300
nrv/2016-11-27:$ sudo cat /etc/php5/apache2/php.ini | grep 'date.timezone = UTC'
date.timezone = UTC
nrv/2016-11-27:$

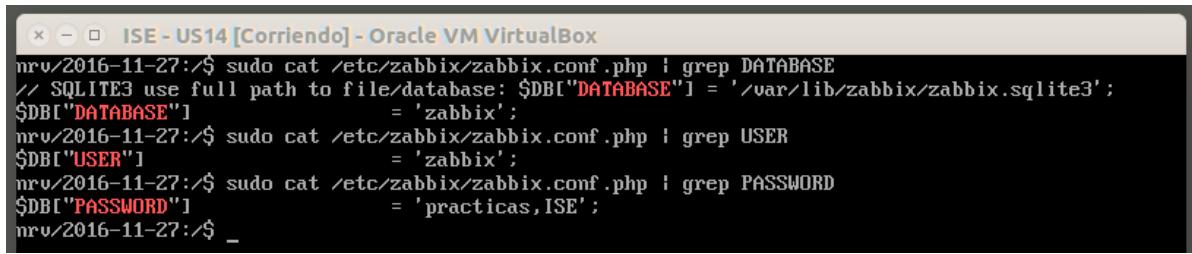
```

Figura 12.7: Parámetros del archivo */etc/php5/apache2/php.ini* modificados.

Copiamos el archivo de configuración de php específico de *Zabbix* ejecutando *sudo cp /usr/share/doc/zabbix-frontend-php/examples/zabbix.conf.php.example /etc/zabbix/zabbix.conf.php* para poder ejecutarlo y cambiar los parámetros que podemos ver a continuación para que queden de la siguiente manera:

- `$DB[“DATABASE”] = ‘zabbix’;`
- `$DB[“USER”] = ‘zabbix’;`
- `$DB[“PASSWORD”] = ‘practicas,ISE’`

El resultado lo podemos ver en la figura 12.8.



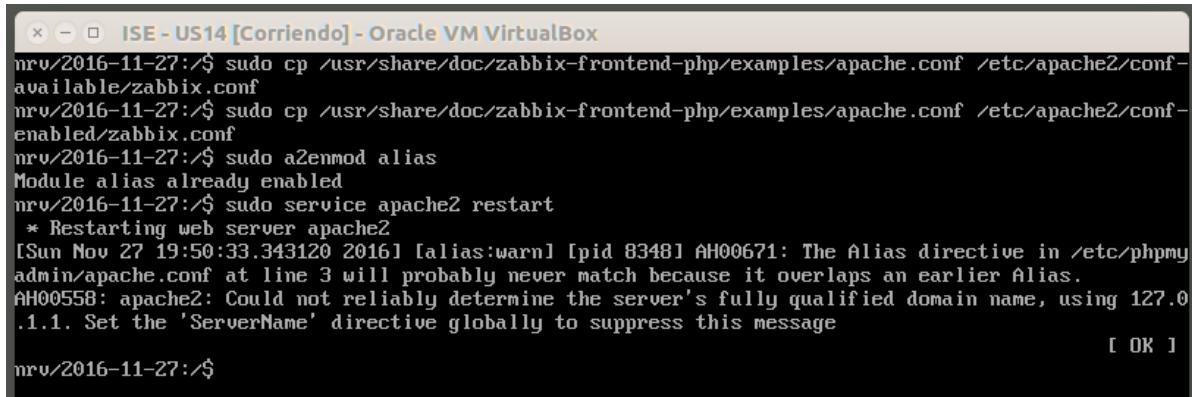
```
ISE - US14 [Corriendo] - Oracle VM VirtualBox
mrw/2016-11-27:$ sudo cat /etc/zabbix/zabbix.conf.php | grep DATABASE
// SQLITE3 use full path to file/database: $DBI["DATABASE"] = '/var/lib/zabbix/zabbix.sqlite3';
$DBI["DATABASE"] = 'zabbix';
mrw/2016-11-27:$ sudo cat /etc/zabbix/zabbix.conf.php | grep USER
$DBI["USER"] = 'zabbix';
mrw/2016-11-27:$ sudo cat /etc/zabbix/zabbix.conf.php | grep PASSWORD
$DBI["PASSWORD"] = 'practicas,ISE';
mrw/2016-11-27:$ _
```

Figura 12.8: Parámetros del archivo */etc/zabbix/zabbix.conf.php* modificados.

A continuación copiamos el archivo apache de *Zabbix* en los archivos de configuración de apache ejecutando. Para ello ejecutamos:

- *sudo cp /usr/share/doc/zabbix-frontend-php/examples/apache.conf /etc/apache2/conf-available/zabbix.conf*
- *sudo cp /usr/share/doc/zabbix-frontend-php/examples/apache.conf /etc/apache2/conf-enable/zabbix.conf*

Nos aseguramos de que los alias están habilitados dentro de Apache ejecutando *sudo a2enmod alias* y reiniciamos *Apache* ejecutando *sudo service apache2 restart*. Todo este proceso lo podemos ver en la figura 12.9. Lo siguiente que tenemos que hacer es editar el fichero */etc/default/zabbix-server* y cambiar el valor de *START* a “yes” e iniciamos *Zabbix* ejecutando *sudo service zabbix-server start* como podemos ver en la figura 12.10.



```
ISE - US14 [Corriendo] - Oracle VM VirtualBox
mrw/2016-11-27:$ sudo cp /usr/share/doc/zabbix-frontend-php/examples/apache.conf /etc/apache2/conf-available/zabbix.conf
mrw/2016-11-27:$ sudo cp /usr/share/doc/zabbix-frontend-php/examples/apache.conf /etc/apache2/conf-enabled/zabbix.conf
mrw/2016-11-27:$ sudo a2enmod alias
Module alias already enabled
mrw/2016-11-27:$ sudo service apache2 restart
* Restarting web server apache2
[Sun Nov 27 19:50:33.343120 2016] [alias:warn] [pid 8348] AH00671: The Alias directive in /etc/phpmyadmin/admin/apache.conf at line 3 will probably never match because it overlaps an earlier Alias.
AH00558: apache2: Could not reliably determine the server's fully qualified domain name, using 127.0.1.1. Set the 'ServerName' directive globally to suppress this message
[ OK ]
```

Figura 12.9: Configuración archivo apache de *Zabbix*.

```
nrv/2016-11-27:$ sudo cat /etc/default/zabbix-server | grep START
START=yes
nrv/2016-11-27:$ sudo service zabbix-server start
zabbix-server start/running, process 7444
nrv/2016-11-27:$
```

Figura 12.10: Inicio de Zabbix.

Ya está todo listo en el servidor, ahora debemos configurar el cliente. El cliente en mi caso va a ser mi máquina anfitriona. Debemos instalar el *Agente de Zabbix*, para ello seguimos los 2 primeros pasos que realizamos en el servidor 12, como podemos ver en la figura 12.11. A continuación instalamos el *Agente de Zabbix* ejecutando *sudo apt-get install zabbix-agent*.

```
# Zabbix Application PPA
deb http://ppa.launchpad.net/tbfr/zabbix/ubuntu precise main
deb-src https://ppa.launchpad.net/tbfr/zabbix/ubuntu precise main
nrv/2016-11-27:$ sudo apt-key adv --keyserver keyserver.ubuntu.com --recv-keys C407E1705F76A32B
Executing: gpg --ignore-time-conflict -o no-options -o no-default-keyring --homedir /tmp/tmp.tllkgkfFu --no-auto-check-trustdb --trust-model always --keyring /etc/apt/trusted.gpg --primary --keyring /etc/apt/trusted.gpg --keyring /etc/apt/trusted.gpg.d/sun-java-community-team-sun-jav6.gpg --keyring /etc/apt/trusted.gpg.d/webupd8team-java.gpg --keyserver keyserver.ubuntu.com --recv-keys C407E1705F76A32B
gpg: soliciting clave 5F76A32B de hkp servidor keyserver.ubuntu.com
gpg: clave 5F76A32B: clave pública "Launchpad PPA for Tobias Frederick" importada
gpg: Cantidad total procesada: 1
gpg:           Importadas: 1 (RSA: 1)
```

Figura 12.11: Instalación del Agente de Zabbix.

Una vez instalado, pasamos a modificar los archivos de configuración. El primero de ellos es */etc/zabbix/zabbix\_agentd.conf*. Como podemos ver en la figura 12.12 la dirección IP de mi servidor es *192.168.56.103* así que en dicho archivo, en el parámetro *Server* debemos poner dicha dirección, como se ve en la figura 12.12 y reiniciamos el servicio ejecutando *sudo service zabbix-agent restart*.

```
nestor@nestor:~$ sudo cat /etc/zabbix/zabbix_agentd.conf | grep 192.168.56.103
Server=192.168.56.103
nrv/2016-11-27:$ sudo service zabbix-agent restart
zabbix-agent stop/waiting
zabbix-agent start/running, process 16874
nrv/2016-11-27:$
```

| Interface | Dirección IP   | Estado | Colisiones | Paquetes RX | Paquetes TX |
|-----------|----------------|--------|------------|-------------|-------------|
| eth0      | 192.168.56.103 | Up     | 0          | 12963       | 1060        |
| eth1      | 192.168.56.102 | Up     | 0          | 3172        | 1060        |
| lo        | 127.0.0.1      | Up     | 0          | 99          | 1060        |

Figura 12.12: Dirección IP del servidor con Zabbix instalado (máquinas conectadas en modo *host-only*).

Dentro del mismo fichero debemos editar el valor de *Hostname* y poner el *hostname* de la máquina cliente, *nestor* en mi caso, y reiniciamos el servicio, como podemos ver en la figura 12.13

```

nestor@nestor: ~
nrv/2016-11-27:~$ sudo cat /etc/zabbix/zabbix_agentd.conf | grep nestor
Hostname=nestor
nrv/2016-11-27:~$ sudo service zabbix-agent restart
zabbix-agent stop/waiting
zabbix-agent start/running, process 17524
nrv/2016-11-27:~$ 

```

Figura 12.13: Parámetro *hostname*.

Finalmente, para acceder al servicio debemos escribir en la barra del navegador la dirección IP de nuestro servidor seguido de */zabbix*, como podemos ver en la figura 12.14. Por defecto, el usuario y contraseña para entrar son *admin* y *zabbix* respectivamente.

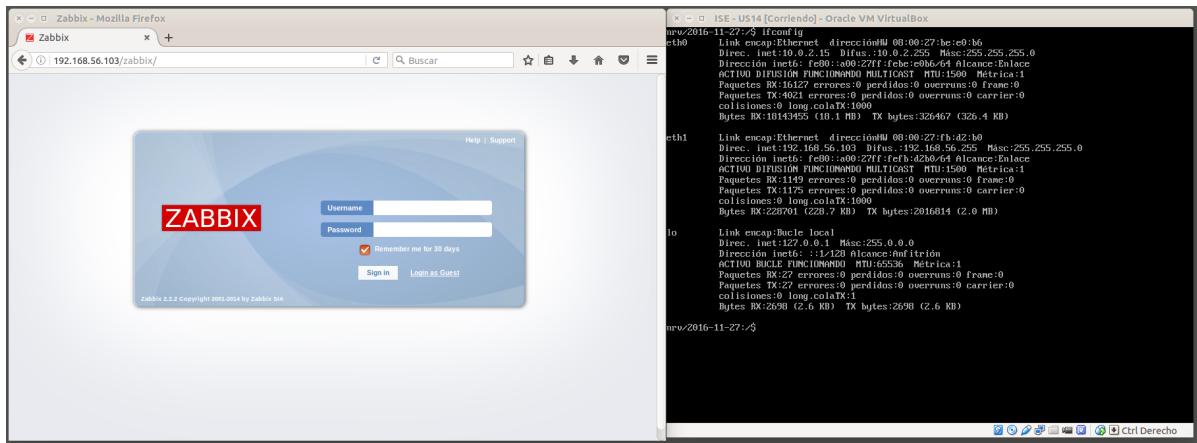


Figura 12.14: *Zabbix* en funcionamiento

Una vez hemos entrado, nos vamos a la pestaña *Configuration* y dentro de dicha pestaña elegimos la de *Hosts*, como podemos ver la figura 12.15. Una vez aquí, pinchamos sobre *Zabbix Server*.

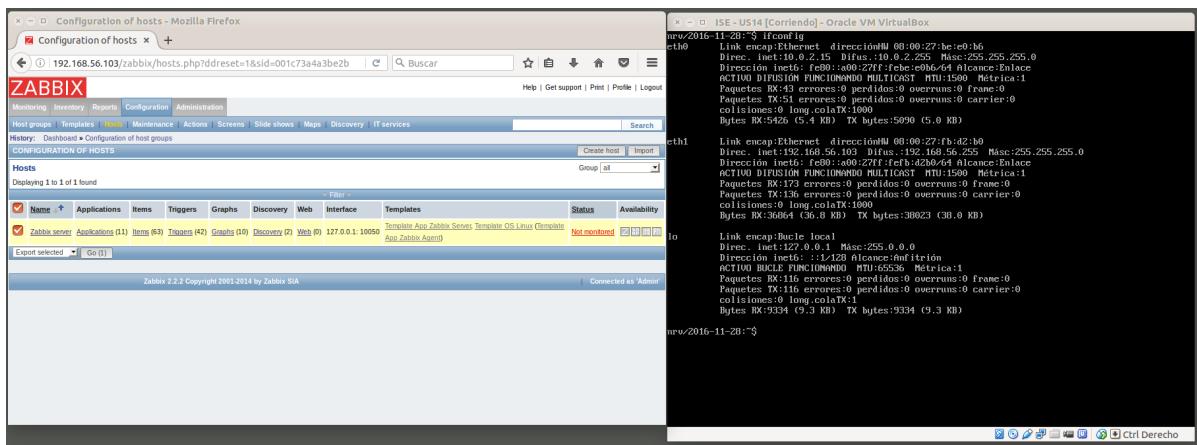


Figura 12.15: Host disponibles.

Dentro del servidor de Zabbix, al final de todo, en la sección de *Status* elegimos *Monitored*, como podemos ver en la figura 12.16.

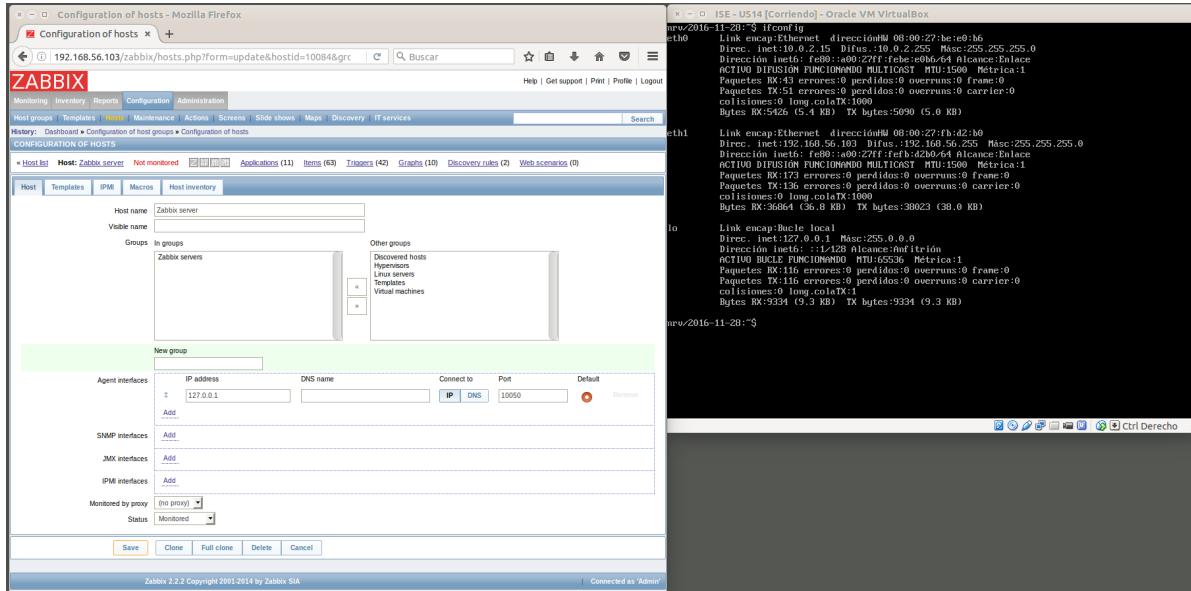


Figura 12.16: Cambio del estado del servidor.

Tras unos instantes, podemos ver que en la pestaña *Last data* dentro de *Monitoring* los distintos hosts que se pueden modificar. Dentro de nuestro servidor Zabbix, podemos ver una gran cantidad de parámetros monitorizados. Parte de ellos los podemos ver en la figura 12.17.

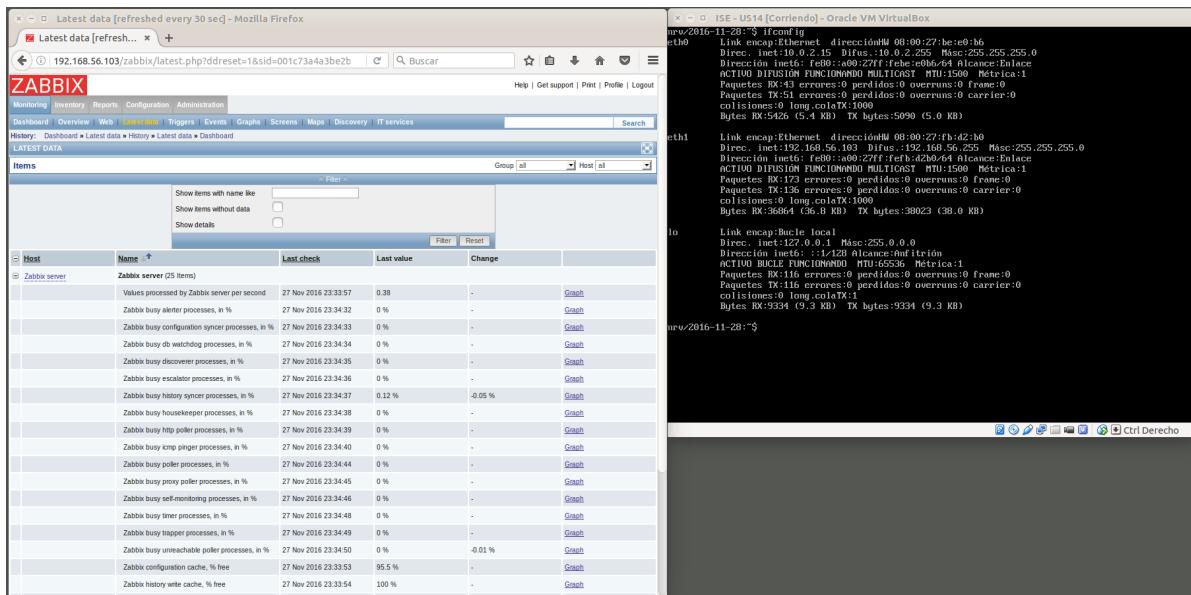


Figura 12.17: Parámetros monitorizados.

Uno de los parámetros que podemos ver es el número de valores procesados por el servidor *Zabbix* por segundo. Podemos ver una gráfica sobre la información recogida de dicho parámetro en la figura 12.18. En dicha gráfica podemos ver que no hay mucha actividad en el servidor. El servidor se ha puesto en marcha a las 23:14 y la gráfica llega hasta las 23:36. En este periodo el valor se mantiene “constante” en un valor algo superior a 0.38. Podemos ver que hay dos picos a las 23:18 y 23:28 con un valor cercano a 0.42 valores procesados por el servidor *Zabbix* por segundo.

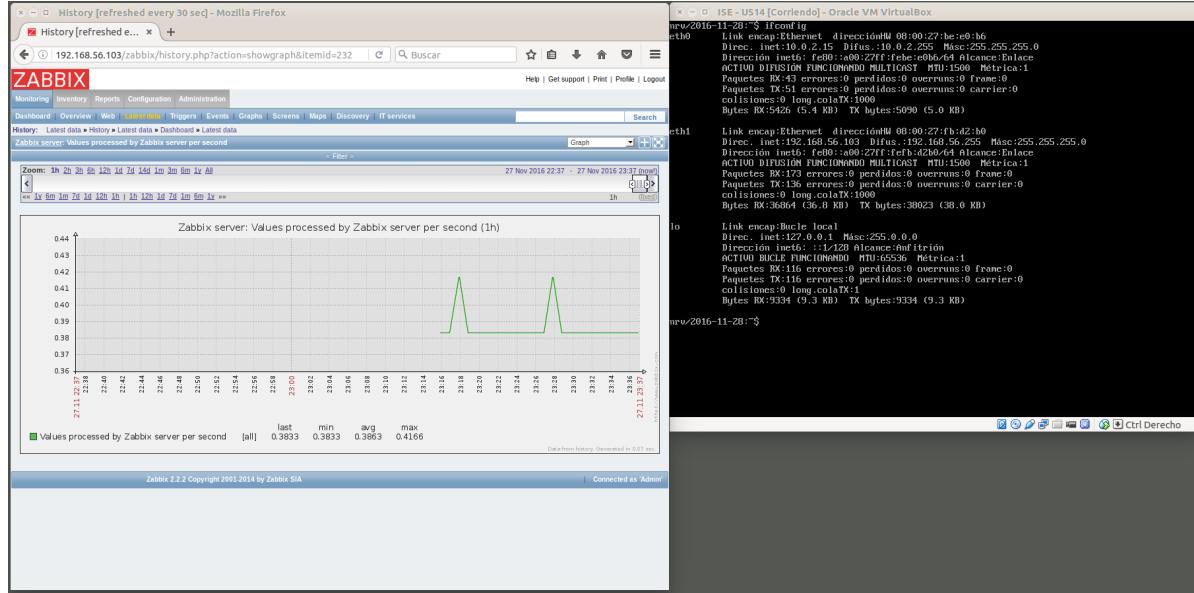


Figura 12.18: Valores procesados por el servidor *Zabbix* por segundo.

### 13. Cuestión opcional 5: Pruebe a instalar este monitor en alguno de sus tres sistemas. Realice capturas de pantalla del proceso de instalación y comente capturas de pantalla del programa en ejecución.

La instalación voy a hacerla en CentOS. Siguiendo la documentación de la página oficial de Cacti [15], debemos ejecutar el comando *sudo yum install cacti*, como podemos ver en la figura 13.1.

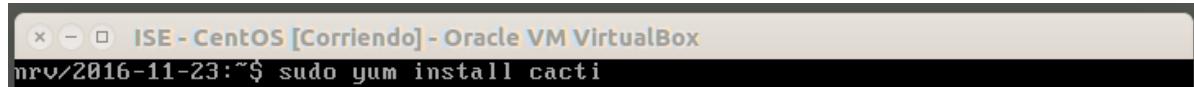


Figura 13.1: Instalación de Cacti.

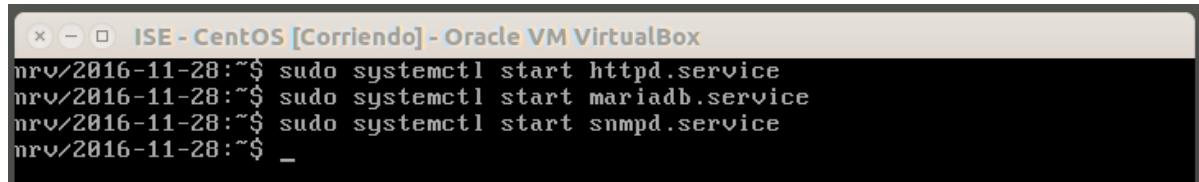
Cómo podemos ver en la documentación de Cacti [16], para su correcto funcionamiento

necesitamos que los siguientes paquetes esté instalados:

- httpd
- php
- php-mysql
- php-snmp
- mysql
- mysql-server<sup>5</sup>
- net-snmp

La mayoría de estos paquetes los hemos ido instalando a lo largo de las prácticas y los que no, ya vienen instalados por defecto en CentOS. Una vez está todo instalado, iniciamos los servicios. Como podemos ver en la figura 13.2, ejecutamos:

- systemctl start httpd.service
- systemctl start mariadb.service
- systemctl start snmpd.service



```
ISE - CentOS [Corriendo] - Oracle VM VirtualBox
nrv/2016-11-28:~$ sudo systemctl start httpd.service
nrv/2016-11-28:~$ sudo systemctl start mariadb.service
nrv/2016-11-28:~$ sudo systemctl start snmpd.service
nrv/2016-11-28:~$ _
```

Figura 13.2: Iniciamos los servicios.

A continuación, debemos configurar las bases de datos de MySQL (MariaDB en mi caso) tal y como nos indican en la documentación de Cacti [17]. Lo primero que hacemos es entrar a MariaDB ejecutando *mysql -u root -p*. Una vez dentro, como podemos ver en la figura 13.3, realizamos los siguientes pasos:

1. Creamos la base de datos para *Cacti* ejecutando *create database cacti;*
2. Concedemos permisos sobre esa base de datos y creamos un usuario ejecutando *grant all on cacti.\* to 'cacti'@'localhost' identified by 'practicas,ISE';*
3. Actualizamos los permisos ejecutando *flush privileges;*
4. Salimos de MariaDB ejecutando *exit;*

<sup>5</sup>Dado que lo estoy haciendo en CentOS, yo uso *mariadb*, como ya explique en prácticas anteriores.

```

nrv/2016-11-28:~$ mysql -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 713
Server version: 5.5.50-MariaDB MariaDB Server

Copyright (c) 2000, 2016, Oracle, MariaDB Corporation Ab and others.

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

MariaDB [(none)]> create database cacti;
Query OK, 1 row affected (0.00 sec)

MariaDB [(none)]> grant all on cacti.* to 'cacti'@'localhost' identified by 'practicas,ISE';
Query OK, 0 rows affected (0.00 sec)

MariaDB [(none)]> flush privileges;
Query OK, 0 rows affected (0.00 sec)

MariaDB [(none)]> exit;
Bye
nrv/2016-11-28:~$ 

```

Figura 13.3: Creación base de datos para *Cacti*.

Importamos las tablas de *Cacti* ejecutando `mysql -u cacti -p cacti < /usr/share/doc/cacti-0.8.8b/cacti.sql`, como podemos ver en la figura 13.4.

```

nrv/2016-11-28:~$ mysql -u cacti -p cacti < /usr/share/doc/cacti-0.8.8b/cacti.sql
Enter password:
nrv/2016-11-28:~$ 

```

Figura 13.4: Importación de las tablas para *Cacti*.

A continuación, debemos editar el archivo `/etc/cacti/db.php` y cambiar los parámetros que podemos ver a continuación para que queden de la siguiente manera:

- `$database_type = "mysql";`
- `$database_default = "cacti";`
- `$database_hostname = "localhost";`
- `$database_username = "cacti";`
- `$database_password = "practicas,ISE";`

El resultado de modificar dicho archivo lo podemos ver en la figura 13.5.

```

nrv/2016-11-28:~$ sudo cat /etc/cacti/db.php | grep database_
$database_type = "mysql";
$database_default = "cacti";
$database_hostname = "localhost";
$database_username = "cacti";
$database_password = "practicas, ISE";
$database_port = "3306";
$database_ssl = false;
nrv/2016-11-28:~$ _

```

Figura 13.5: Parámetros modificados.

Asignamos los permisos a `/usr/share/cacti/rra` y a `/usr/share/cacti/log` correctamente. Para ello, como podemos ver en la figura 13.6 ejecutamos:

- `chown -R cacti /usr/share/cacti/rra/`
- `chown -R cacti /usr/share/cacti/log/`

```

nrv/2016-11-28:~$ sudo chown -R cacti /usr/share/cacti/rra
nrv/2016-11-28:~$ sudo chown -R cacti /usr/share/cacti/log
nrv/2016-11-28:~$ _

```

Figura 13.6: Permisos modificados.

A continuación debemos editar el fichero de configuración de *Apache* en función de la versión que tenemos. En CentOS, tengo la versión 2.4.6<sup>6</sup>, por lo tanto debemos modificar la parte correspondiente a versiones 2.4 de *Apache*. Debemos permitir conexiones desde el exterior de nuestro servidor, por eso cambiamos la línea que hay debajo de la línea `# httpd 2.4` por *Requiere all granted*. Una vez hecho el cambio, reiniciamos *Apache* ejecutando `sudo systemctl restart httpd.service`. Por último editamos nuestro fichero *crontab* ejecutando `*/5 * * * * cacti php /usr/share/cacti/poller.php > /dev/null 2>&1`. Este proceso lo podemos ver en la figura 13.7.

---

<sup>6</sup>Para ver la versión que tenemos instalada debemos ejecutar `httpd -v`

```

ISE - CentOS [Corriendo] - Oracle VM VirtualBox
nrv/2016-12-01:~$ sudo cat /etc/httpd/conf.d/cacti.conf | head -20 | tail -10
Alias /cacti      /usr/share/cacti

<Directory /usr/share/cacti/>
    <IfModule mod_authz_core.c>
        # httpd 2.4
        Require all granted
    </IfModule>
    <IfModule !mod_authz_core.c>
        # httpd 2.2
nrv/2016-12-01:~$ sudo systemctl restart httpd.service
nrv/2016-12-01:~$ */5 * * * * cacti php /usr/share/cacti/poller.php > /dev/null
Z>&1
nrv/2016-12-01:~$ _

```

Figura 13.7: Archivo de configuración de *Apache* y archivo *crontab* modificados.

Finalmente, para acceder a *cacti* desde nuestra máquina anfitriona, debemos introducir la dirección IP de nuestro servidor seguido de */cacti*. En mi caso, la dirección IP de mi servidor es *192.168.56.101* (máquinas conectadas en modo *host-only*), así que debemos escribir *192.168.56.101/cacti*, como podemos ver en la figura 13.8.

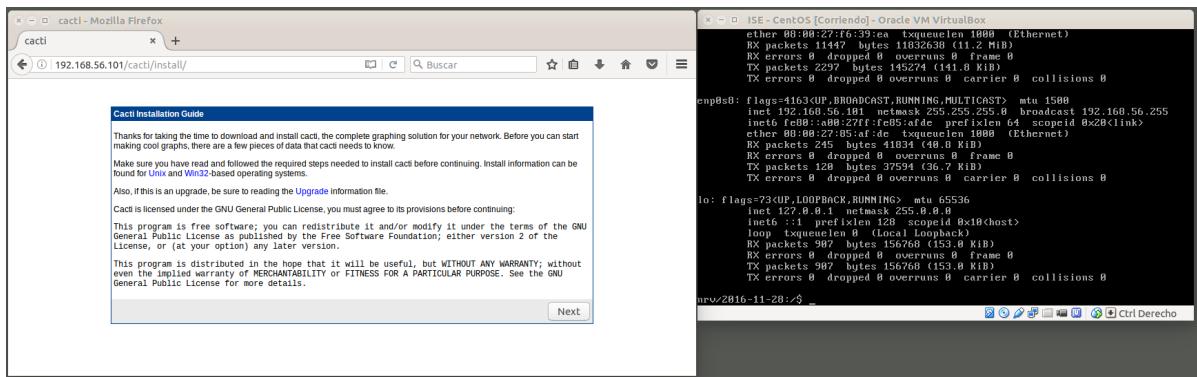


Figura 13.8: *Cacti* en funcionamiento.

Desde el navegador, debemos instalar cacti. Para ello, pulsamos el botón de *Next*. En la pantalla que podemos ver en la figura 13.9, elegimos *New Install* y le damos a *Next*. En la siguiente ventana, que podemos ver en la figura 13.10, le damos a *Finish*. Finalmente, podemos ver la pantalla de inicio de sesión, como se ve en la figura 13.11.

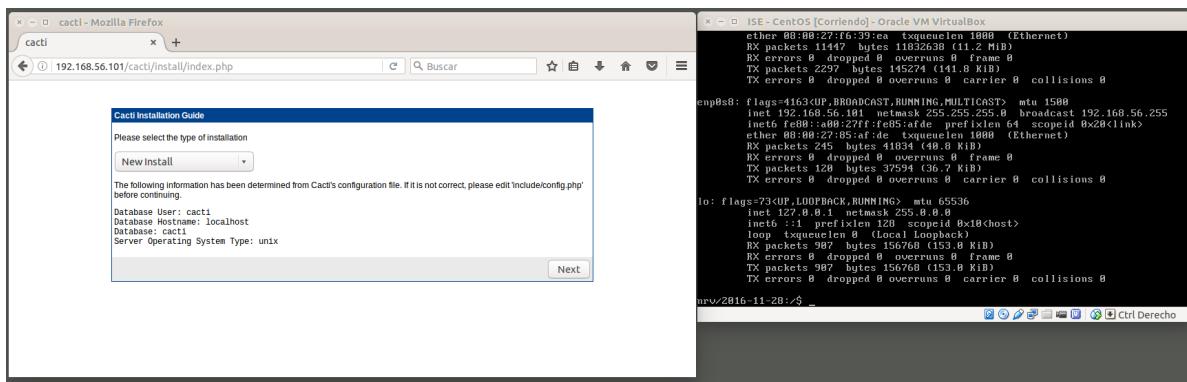


Figura 13.9: Instalación nueva.

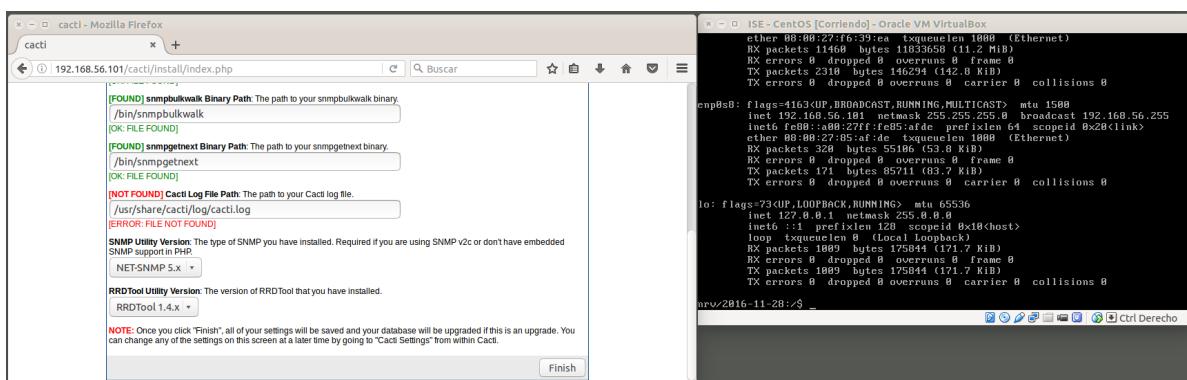


Figura 13.10: Finalización de la instalación.

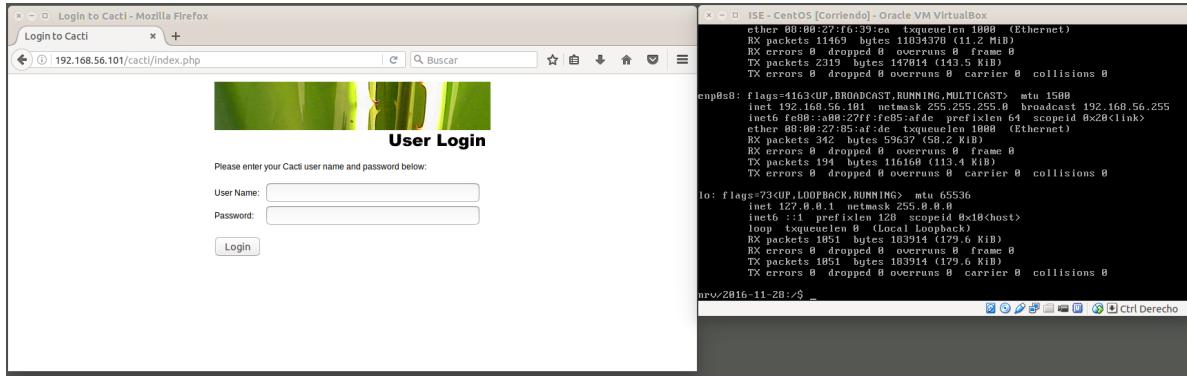


Figura 13.11: Instalación de *Cacti* completada.

En la página que vemos en la figura 13.11 accedemos poniendo como usuario y contraseña *admin* y *admin* correspondientemente. Una vez introducida los datos, se nos obliga a cambiar la contraseña, como podemos ver en la figura 13.12. En mi caso, he introducido la de siempre: *practicas,ISE*.

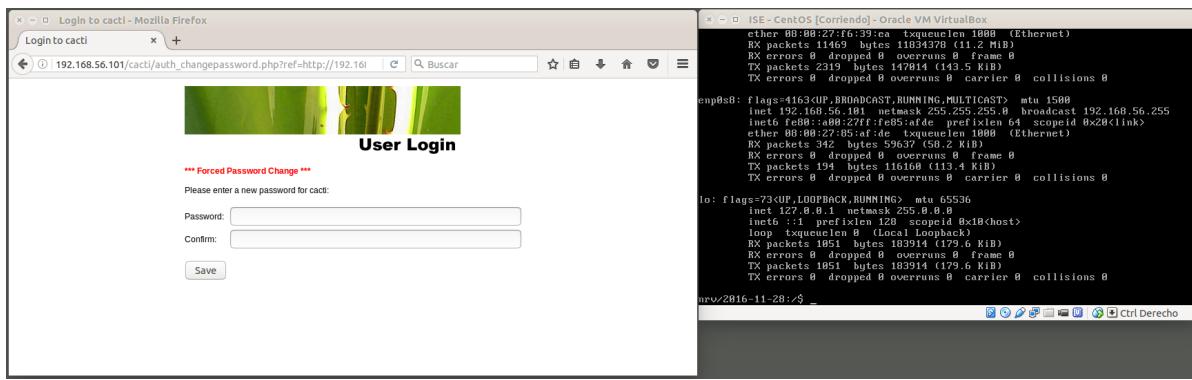


Figura 13.12: Cambio forzado de contraseña.

Una vez instalado hemos instalado *Cacti*, seleccionamos el gráfico que queremos añadir al árbol de gráfico y lo añadimos, tal y como podemos ver en la figura 13.13. Para ver el gráfico, pinchamos sobre la opción de *graphs* que podemos ver en la parte superior de *cacti*. Podemos ver los datos en la figura 13.14.

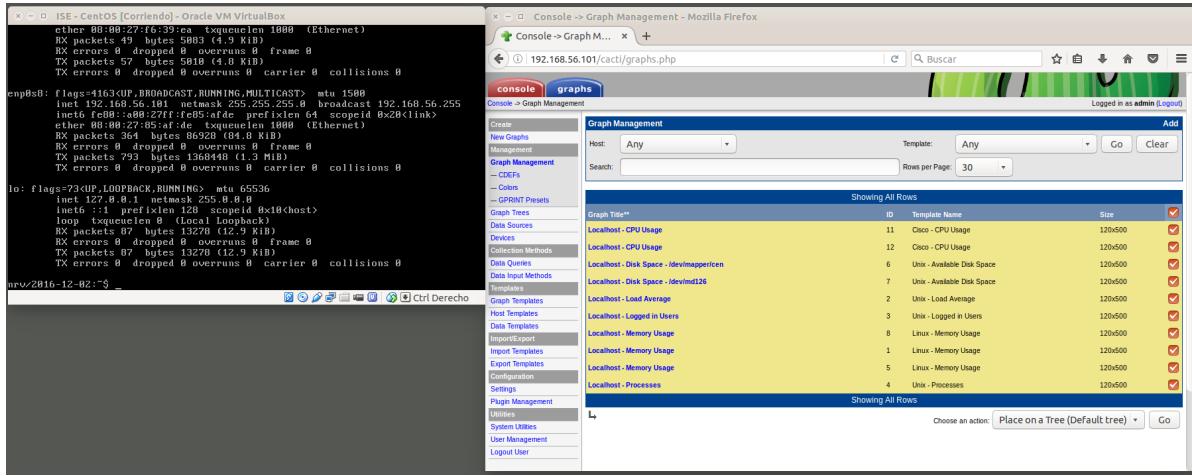


Figura 13.13: Añadimos el gráfico al árbol.

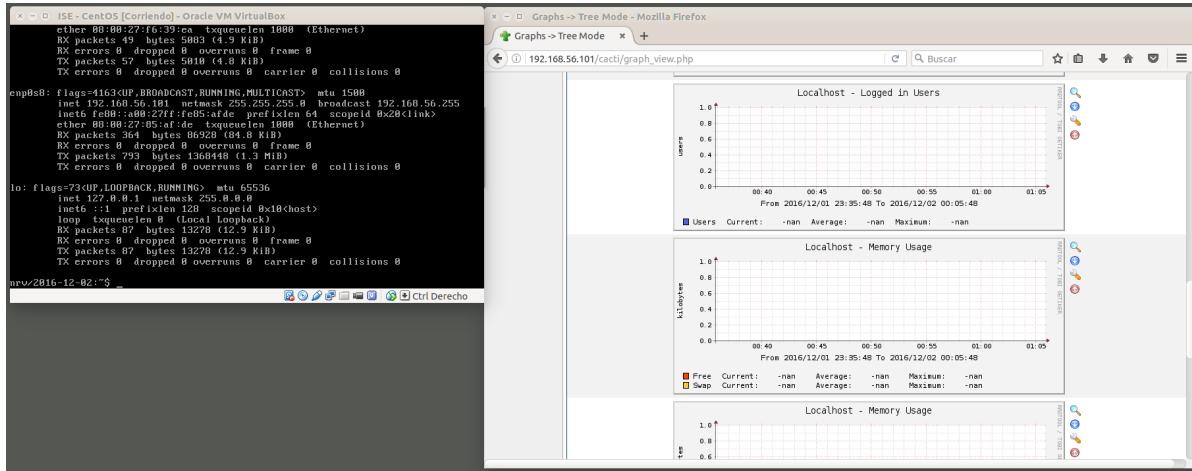


Figura 13.14: Datos recogidos.

Los gráficos que podemos ver en la figura 13.14 representan la carga media del servidor y número de usuarios identificados en el sistema. En ambos casos parece que no hay datos ya que no se ha producido nada en el servidor en el intervalo de tiempo seleccionado.

## 14. Cuestión opcional 6: Instale el monitor. Muestre y comente algunas capturas de pantalla.

Para instalar *AWStats* ejecutamos `sudo apt-get install awstats`. Una vez lo hemos instalado, como podemos ver en la *Community Help Wiki* de Ubuntu [18], si quisiésemos monitorizar varios dominios, debemos crear un archivo de configuración para cada uno. En mi caso sólo voy a monitorizar el localhost, así que con modificar el archivo que tenemos es suficiente. Debemos editar el fichero cambiando los parámetros que vemos a continuación para que reflejen la información correcta. En mi caso, el valor de los parámetros es el siguiente:

- LogFile=“/var/log/apache2/access.log”
- SiteDomain=“localhost”
- HostAliases=“localhost 127.0.0.1”

El resultado de la modificación de dicho archivo lo podemos ver en la figura 14.1. Una vez editado el fichero, creamos las estadísticas iniciales para *AWStats*. Para ello ejecutamos `sudo /usr/lib/cgi-bin/awstats.pl -config=localhost -update`, tal y como se puede ver en la figura 14.1.

```

nrv/2016-12-01:$ cat /etc/awstats/awstats.conf | grep LogFile=""
LogFile="/var/log/apache2/access.log"
nrv/2016-12-01:$ cat /etc/awstats/awstats.conf | grep SiteDomain=""
SiteDomain="localhost"
nrv/2016-12-01:$ cat /etc/awstats/awstats.conf | grep HostAliases=""
HostAliases="localhost 127.0.0.1"
nrv/2016-12-01:$ sudo /usr/lib/cgi-bin/awstats.pl -config=localhost -update
[sudo] password for nestor:
Create/Update database for config "/etc/awstats/awstats.conf" by AWStats version 7.2 (build 1.992)
From data in log file "/var/log/apache2/access.log"...
Phase 1 : First bypass old records, searching new record...
Direct access after last parsed record (after line 1396)
Jumped lines in file: 1396
Found 1396 already parsed records.
Parsed lines in file: 18
Found 0 dropped records,
Found 0 comments,
Found 0 blank records,
Found 0 corrupted records,
Found 0 old records,
Found 18 new qualified records.
nrv/2016-12-01:$ _

```

Figura 14.1: Configuración de *AWStats*.

Una vez hemos configurado *AWStats*, pasamos a configurar *Apache*. Lo primero que debemos hacer indicarle a *Apache* que use *mod\_cgi* ejecutando *sudo a2enmod cgi*. Dado que no tenemos ningún *Host Virtual*, creamos el archivo */etc/apache2/sites-available/default* y añadimos las líneas que podemos ver en la figura 14.2. También debemos añadir estas líneas al fichero */etc/apache2/sites-enabled/000-default.conf* y reiniciamos *Apache* ejecutando *sudo service apache2 restart*. Este proceso lo podemos ver en la figura 14.2

```

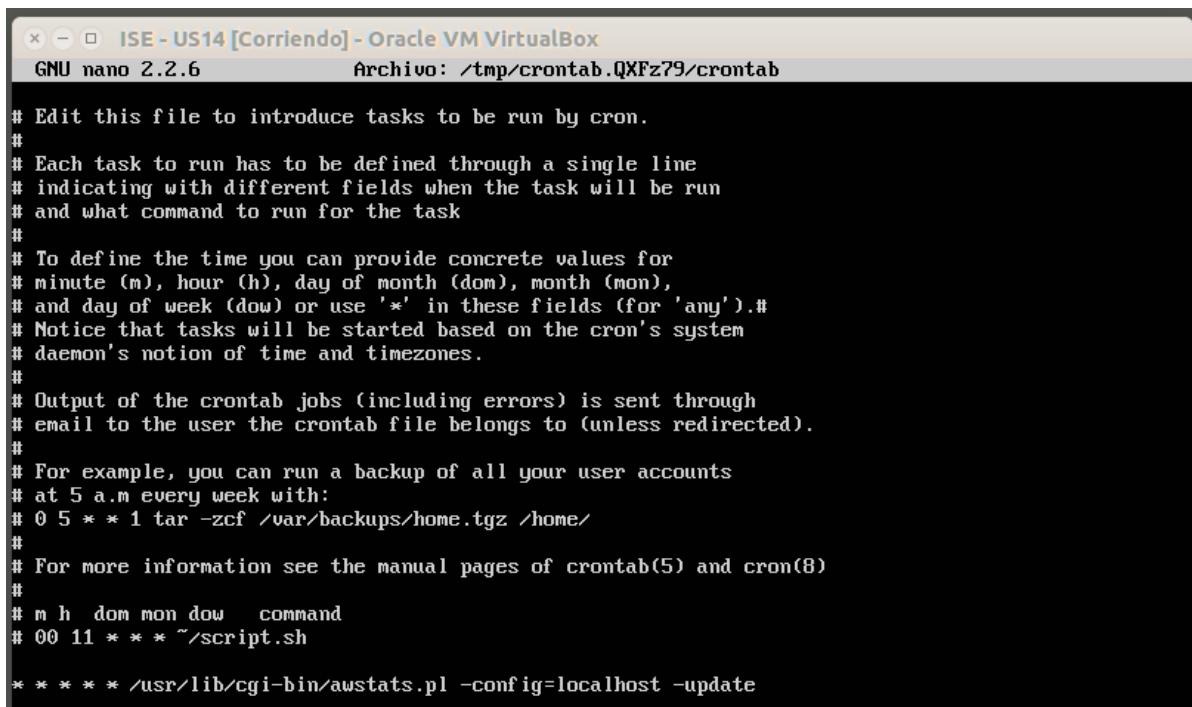
nrv/2016-12-01:$ sudo a2enmod cgi
[Thu Dec 01 23:35:52.495570 2016] [alias:warn] [pid 5974] AH00671: The Alias directive in /etc/phpmyadmin/apache.conf at line 3 will probably never match because it overlaps an earlier Alias.
AH00558: apache2: Could not reliably determine the server's fully qualified domain name, using 127.0.1.1. Set the 'ServerName' directive globally to suppress this message
Module cgi already enabled
nrv/2016-12-01:$ cat /etc/apache2/sites-available/default
Alias /awstatsclasses "/usr/share/awstats/lib/"
Alias /awstats-icon "/usr/share/awstats/icon/"
Alias /awstatscss "/usr/share/doc/awstats/examples/css"
ScriptAlias /awstats/ /usr/lib/cgi-bin/
Options +ExecCGI -MultiViews +SymLinksIfOwnerMatch
nrv/2016-12-01:$ cat /etc/apache2/sites-enabled/000-default.conf | tail -8 | head -5
    Alias /awstatsclasses "/usr/share/awstats/lib/"
    Alias /awstats-icon "/usr/share/awstats/icon/"
    Alias /awstatscss "/usr/share/doc/awstats/examples/css"
    ScriptAlias /awstats/ /usr/lib/cgi-bin/
    Options +ExecCGI -MultiViews +SymLinksIfOwnerMatch
nrv/2016-12-01:$ sudo service apache2 restart
 * Restarting web server apache2
[Thu Dec 01 23:36:08.757288 2016] [alias:warn] [pid 6019] AH00671: The Alias directive in /etc/phpmyadmin/apache.conf at line 3 will probably never match because it overlaps an earlier Alias.
AH00558: apache2: Could not reliably determine the server's fully qualified domain name, using 127.0.1.1. Set the 'ServerName' directive globally to suppress this message
[ OK ]
nrv/2016-12-01:$ _

```

Figura 14.2: Configuración de *Apache*

Por último, modificamos nuestro archivo crontab para actualizar *AWStats* cada minuto. Para ello ejecutamos `* * * * * /usr/lib/cgi-bin/awstats.pl -config=localhost -update`, como podemos ver en la figura 14.3.

Para acceder a *AWStats*, debemos introducir en el navegador de nuestra máquina anfitriona `http://<IP del servidor>/awstats/awstats.pl?`. En mi caso, como podemos ver en la figura 14.4, la dirección IP de mi servidor es `19.168.56.101` (máquinas conectadas en modo *host-only*) así que debemos introducir `http://192.168.56.101/awstats/awstats.pl?`. *AWStats* nos muestra información como puede ser el número de visitas o el ancho de banda. Toda esta información se puede ver representada en gráficas, existiendo una gráfica para cada mes. En mi caso, podemos ver que la actividad empieza en Octubre, que fue cuando se instaló el servidor web para las prácticas de la asignatura.



The screenshot shows a terminal window titled "ISE-US14 [Corriendo] - Oracle VM VirtualBox". The title bar also includes "GNU nano 2.2.6" and "Archivo: /tmp/crontab.QXFz79/crontab". The terminal displays the following content:

```
# 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
# 00 11 * * * ~/script.sh
* * * * * /usr/lib/cgi-bin/awstats.pl -config=localhost -update
```

Figura 14.3: Fichero de *crontab* modificado.

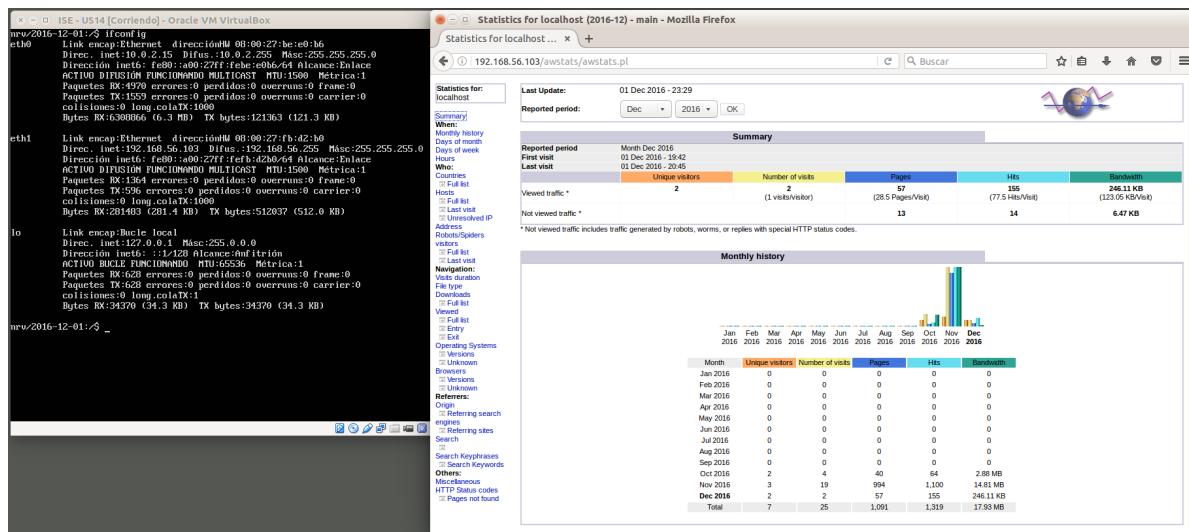


Figura 14.4: Información proporcionada por *AWStats*

## 15. Cuestión opcional 10: Escriba un script en PowerShell y analice su comportamiento usando el profiler presentado.

El script que voy a hacer en PowerShell es el script que hice para la cuestión 8 en Python. El script lo podemos ver en el script 8. El resultado lo podemos ver en el script <sup>7</sup> 15.

```
$suma=0

for ( $i=0; $i -le 100; $i++){
    If ( $i % 2 -ne 0) {
        $i
        $suma += $i
    }
}

“La suma es: ” + $suma
```

<sup>7</sup>El script “O10-script.ps1” se encuentra dentro de la carpeta *Archivos auxiliares*

The screenshot shows a Windows PowerShell window titled "Administrador: Windows PowerShell". The command run was ".\010-script.ps1". The output consists of a large number of lines starting from 1 and ending at 99, followed by the message "La suma es: 2500". The window is set against a desktop background with icons for Start, File Explorer, Task View, and others.

```
ISE - WS2008 [Corriendo] - Oracle VM VirtualBox
Administrator: Windows PowerShell
Windows PowerShell
Copyright (C) 2009 Microsoft Corporation. Reservados todos los derechos.

PS C:\Users\Administrador> cd ..\Desktop
PS C:\Users\Administrador\Desktop> .\010-script.ps1
1
3
5
7
9
11
13
15
17
19
21
23
25
27
29
31
33
35
37
39
41
43
45
47
49
51
53
55
57
59
61
63
65
67
69
71
73
75
77
79
81
83
85
87
89
91
93
95
97
99
La suma es: 2500
PS C:\Users\Administrador\Desktop>
```

Figura 15.1: Resultado del script 15.

Para usar el profiler, debemos descargarlo de la página de Microsoft [19]. Una vez descargado, para ejecutarlo he tenido que instalar *.NET Framework* [20]. Tras descargarlo, lo ejecutamos. Pero, tal y como hable con usted en tutoría el día 1 de Diciembre, da error. El error lo podemos ver en la figura 15.2.

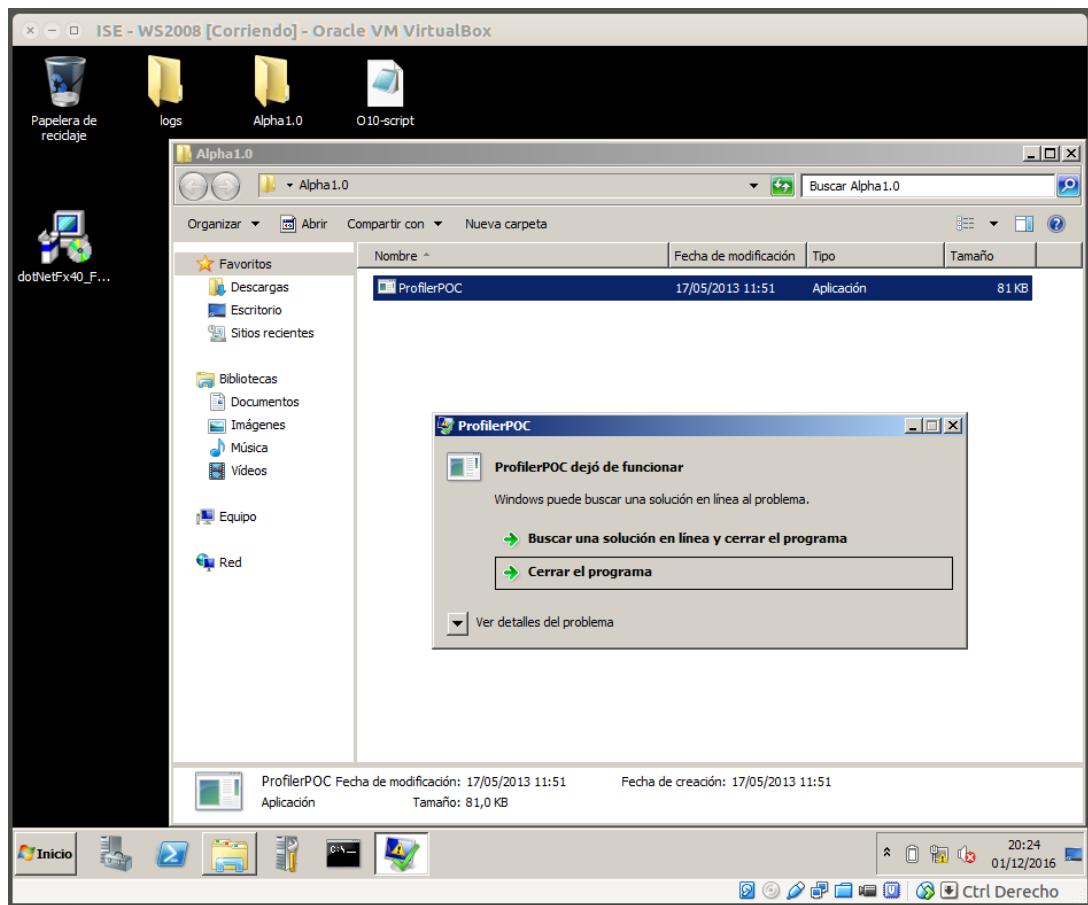


Figura 15.2: Error al ejecutar el profiler.

El resultado que veríamos es el siguiente:

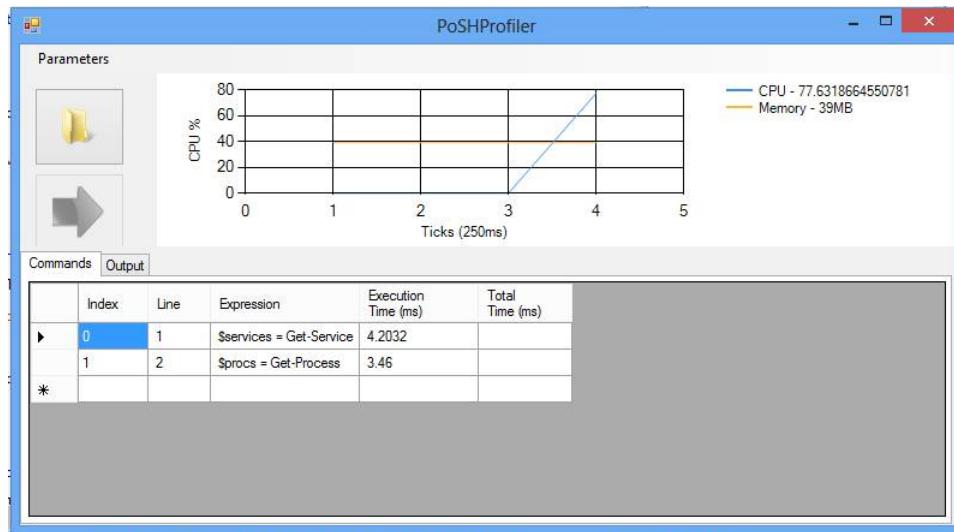


Figura 15.3: Ejemplo de ejecución de PoshProfiler [1].

## **16. 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).**

Para ver como usar el profiler de *MongoDB* he usado la página oficial de *MongoDB* [21]. Dentro de *MongoDB* ejecutamos `db.setProfilingLevel(2, 1)`, como podemos ver en la figura 16.1, para activar el profiler en el segundo nivel (para “profilear” todas las operaciones) y poner como umbral para determinar cuando una consulta se considera lenta en 1ms. Lo primero que voy a hacer es crear la colección con los documentos que voy a usar, para ello ejecutamos las siguientes instrucciones:

1. Creamos la colección que vamos a usar, *ProfilerMongoDB*, para ello ejecutamos `db.createCollection('ProfilerMongoDB')`, como podemos ver en la figura 16.1.
2. Creamos seis documentos para luego insertarlos en la colección que hemos creado en el paso anterior. Como podemos ver en la figura 16.1 ejecutamos:
  - a) `d1 = { ID: 1, valor: 10 }`
  - b) `d2 = { ID: 2, valor: 20 }`
  - c) `d3 = { ID: 3, valor: 30 }`
  - d) `d4 = { ID: 4, valor: 40 }`
  - e) `d5 = { ID: 5, valor: 50 }`
  - f) `d6 = { ID: 6, valor: 60 }`
3. Insertamos los documentos en la colección *ProfilerMongoDB*. Como podemos ver en la figura 16.2 ejecutamos:
  - a) `db.ProfilerMongoDB.insert(d1)`
  - b) `db.ProfilerMongoDB.insert(d2)`
  - c) `db.ProfilerMongoDB.insert(d3)`
  - d) `db.ProfilerMongoDB.insert(d4)`
  - e) `db.ProfilerMongoDB.insert(d5)`
  - f) `db.ProfilerMongoDB.insert(d6)`
4. Realizamos la consulta ejecutando `db.ProfilerMongoDB.find()`, como podemos ver en la figura 16.2.

```

> db.setProfilingLevel(2,1)
{
  "was" : 0,
  "slowms" : 100,
  "ok" : 1
}
> db.createCollection('ProfilerMongoDB')
{
  "ok" : 1
}
> d1 = {ID: 1, valor: 10}
{
  "ID" : 1,
  "valor" : 10
}
> d2 = {ID: 2, valor: 20}
{
  "ID" : 2,
  "valor" : 20
}
> d3 = {ID: 3, valor: 30}
{
  "ID" : 3,
  "valor" : 30
}
> d4 = {ID: 4, valor: 40}
{
  "ID" : 4,
  "valor" : 40
}
> d5 = {ID: 5, valor: 50}
{
  "ID" : 5,
  "valor" : 50
}
> d6 = {ID: 6, valor: 60}
{
  "ID" : 6,
  "valor" : 60
}
>

```

The terminal window title is "nestor@nesto" and the command line shows "nrv/2016-11-30:~\$ |". The bottom of the window has standard Linux terminal icons.

Figura 16.1: Creación de la colección y de los documentos.

```

> db.ProfilerMongoDB.insert(d1)
WriteResult({ "nInserted" : 1 })
> db.ProfilerMongoDB.insert(d2)
WriteResult({ "nInserted" : 1 })
> db.ProfilerMongoDB.insert(d3)
WriteResult({ "nInserted" : 1 })
> db.ProfilerMongoDB.insert(d4)
WriteResult({ "nInserted" : 1 })
> db.ProfilerMongoDB.insert(d5)
WriteResult({ "nInserted" : 1 })
> db.ProfilerMongoDB.insert(d6)
WriteResult({ "nInserted" : 1 })
> db.ProfilerMongoDB.find()
{
  "_id" : ObjectId("583eaa745357180bdab1531a"),
  "ID" : 1,
  "valor" : 10
}
{
  "_id" : ObjectId("583eaa765357180bdab1531b"),
  "ID" : 2,
  "valor" : 20
}
{
  "_id" : ObjectId("583eaa785357180bdab1531c"),
  "ID" : 3,
  "valor" : 30
}
{
  "_id" : ObjectId("583eaa795357180bdab1531d"),
  "ID" : 4,
  "valor" : 40
}
{
  "_id" : ObjectId("583eaa7b5357180bdab1531e"),
  "ID" : 5,
  "valor" : 50
}
{
  "_id" : ObjectId("583eaa7d5357180bdab1531f"),
  "ID" : 6,
  "valor" : 60
}
>

```

The terminal window title is "nestor@nesto" and the command line shows "nrv/2016-11-30:~\$ |". The bottom of the window has standard Linux terminal icons.

Figura 16.2: Inserción y consulta de los documentos.

Una vez hecha la consulta, vamos a ver los datos recogidos por el profiler, para ello ejecutamos `db.system.profile.find().limit(1)`. La consulta anterior nos devolvería la entrada más reciente realizada sobre la base de datos. Como podemos ver en la figura 16.3, la opción `command` nos indica que dicha entrada corresponde a la creación de la colección. También podemos ver más información. Para ello ejecutamos `db.system.profile.find().limit(8)`. La consulta anterior nos devolvería las 8 entradas más recientes realizadas sobre la base de datos. ¿Por qué 8? En el proceso visto anteriormente 16 se ejecutaban 8 comandos sobre la base de datos ya que los comandos para la creación de los documentos no cuentan porque no son sobre la base de datos; uno

para crear la colección, seis para insertar los documentos y uno para la consulta. Parte de los datos los podemos ver en la figura 16.4. En este caso podemos ver información sobre la inserción del último documento y sobre la recuperación de los documentos de la colección. En la información acerca del *insert* podemos ver el contenido del documento que se está insertando y en qué base se está insertando. En la información acerca del *find* podemos ver que el atributo *nReturned* es 6. Este valor indica el número de elementos de la colección que han sido devueltos.

```
> db.system.profile.find().limit(1)
{ "op" : "command", "ns" : "test.ProfilerMongoDB", "command" : { "create" : "ProfilerMongoDB" }, "keyUpdates" : 0, "writeConflicts" : 0, "numYield" : 0, "locks" : { "Global" : { "acquireCount" : { "r" : NumberLong(1), "w" : NumberLong(1) } }, "Database" : { "acquireCount" : { "W" : NumberLong(1) } } }, "responseLength" : 22, "protocol" : "op_command", "millis" : 61, "execStats" : { }, "ts" : ISODate("2016-11-30T10:29:36.369Z"), "client" : "127.0.0.1", "allUsers" : [ 1, "user" : "" ]
> _
```

Figura 16.3: Entrada más reciente.

```
) } } }, "responseLength" : 25, "protocol" : "op_command", "millis" : 0, "execStats" : { }, "ts" : ISODate("2016-11-30T10:31:23.667Z"), "client" : "127.0.0.1", "allUsers" : [ 1, "user" : "" ]
{ "op" : "insert", "ns" : "test.ProfilerMongoDB", "query" : { "insert" : "ProfilerMongoDB", "documents" : [ { "_id" : ObjectId("583eaa7d5357180bdab1531f"), "ID" : 6, "valor" : 60 } ], "ordered" : true }, "ninserted" : 1, "keyUpdates" : 0, "writeConflicts" : 0, "numYield" : 0, "locks" : { "Global" : { "acquireCount" : { "r" : NumberLong(1), "w" : NumberLong(1) } }, "Database" : { "acquireCount" : { "w" : NumberLong(1) } }, "Collection" : { "acquireCount" : { "w" : NumberLong(1) } } }, "responseLength" : 25, "protocol" : "op_command", "millis" : 0, "execStats" : { }, "ts" : ISODate("2016-11-30T10:31:25.410Z"), "client" : "127.0.0.1", "allUsers" : [ 1, "user" : "" ]
{ "op" : "query", "ns" : "test.ProfilerMongoDB", "query" : { "find" : "ProfilerMongoDB", "filter" : { }, "keysExamined" : 0, "docsExamined" : 6, "cursorExhausted" : true, "keyUpdates" : 0, "writeConflicts" : 0, "numYield" : 0, "locks" : { "Global" : { "acquireCount" : { "r" : NumberLong(2) } }, "Database" : { "acquireCount" : { "r" : NumberLong(1) } }, "Collection" : { "acquireCount" : { "r" : NumberLong(1) } } }, "nreturned" : 6, "responseLength" : 423, "protocol" : "op_command", "millis" : 0, "execStats" : { "stage" : "COLLSCAN", "filter" : { "$and" : [ ] }, "nReturned" : 6, "executionTimeMillisEstimate" : 0, "works" : 8, "advanced" : 6, "needTime" : 1, "needYield" : 0, "saveState" : 0, "restoreState" : 0, "isEOF" : 1, "invalidates" : 0, "direction" : "forward", "docsExamined" : 6 }, "ts" : ISODate("2016-11-30T10:31:37.153Z"), "client" : "127.0.0.1", "allUsers" : [ 1, "user" : "" ]
> db.system.profile.find().limit(8)_
```

Figura 16.4: 8 entradas más recientes.

## Referencias

- [1] <https://poshprofiler.codeplex.com/>. Fecha de acceso: 01/12/2016.
- [2] <https://help.ubuntu.com/community/ListInstalledPackagesByDate>. Año de publicación/última edición: 2014 - Fecha de acceso: 17/11/2016.
- [3] <https://linux.die.net/man/8/logrotate>. Fecha de acceso: 17/11/2016.
- [4] <https://linux.die.net/man/1/crontab>. Fecha de acceso: 17/11/2016.
- [5] <http://demo.munin-monitoring.org/munin-monitoring.org/index.html>. Fecha de acceso: 20/11/2016.
- [6] [http://blog.softlayer.com/2013/sysadmin-tips-and-tricks-using-strace-to-monitor-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-utm_source=twitter&utm_medium=social&utm_content=beyond-the-command-line-with-strace&utm_campaign=blog-development-tips-and-tricks). Fecha de acceso: 21/11/2016.
- [7] <http://docs.python.org/2/library/profile.html>. Fecha de acceso: 21/11/2016.
- [8] <http://dev.mysql.com/doc/refman/5.7/en/show-profiles.html>. Fecha de acceso: 21/11/2016.
- [9] <http://dev.mysql.com/doc/refman/5.7/en/show-profile.html>. Fecha de acceso: 21/11/2016.
- [10] <https://dev.mysql.com/doc/refman/5.7/en/query-cache-thread-states.html>. Fecha de acceso: 23/11/2016.
- [11] [https://raid.wiki.kernel.org/index.php/Detecting,\\_querying\\_and\\_testing](https://raid.wiki.kernel.org/index.php/Detecting,_querying_and_testing). Fecha de acceso: 23/11/2016.
- [12] [https://assets.nagios.com/downloads/nagiosxi/docs/XI\\_Manual\\_Installation\\_Instructions.pdf#\\_ga=1.242062663.413415278.1479885662](https://assets.nagios.com/downloads/nagiosxi/docs/XI_Manual_Installation_Instructions.pdf#_ga=1.242062663.413415278.1479885662). Fecha de acceso: 23/11/2016.
- [13] <https://library.nagios.com/library/products/nagios-xi/tutorials/nagios-xi-web-interface-setup-guide/>. Fecha de acceso: 23/11/2016.
- [14] <https://www.digitalocean.com/community/tutorials/how-to-install-zabbix-on-ubuntu-configure-it-to-monitor-multiple-vps-servers>. Fecha de acceso: 23/11/2016.
- [15] [http://www.cacti.net/download\\_cacti.php](http://www.cacti.net/download_cacti.php). Fecha de acceso: 23/11/2016.
- [16] [http://www.cacti.net/downloads/docs/html/install\\_unix.html](http://www.cacti.net/downloads/docs/html/install_unix.html). Fecha de acceso: 28/11/2016.

- [17] [http://www.cacti.net/downloads/docs/html/unix\\_configure\\_cacti.html](http://www.cacti.net/downloads/docs/html/unix_configure_cacti.html).  
Fecha de acceso: 28/11/2016.
- [18] <https://help.ubuntu.com/community/AWStats>. Año de publicación/última edición: 2015 - Fecha de acceso: 29/11/2016.
- [19] <https://gallery.technet.microsoft.com/Powershell-script-profiler-4382ffad>.  
Fecha de acceso: 29/11/2016.
- [20] <https://www.microsoft.com/es-es/download/details.aspx?id=17718>. Fecha de acceso: 29/11/2016.
- [21] <http://docs.mongodb.org/manual/tutorial/manage-the-database-profiler/>.  
Fecha de acceso: 30/11/2016.