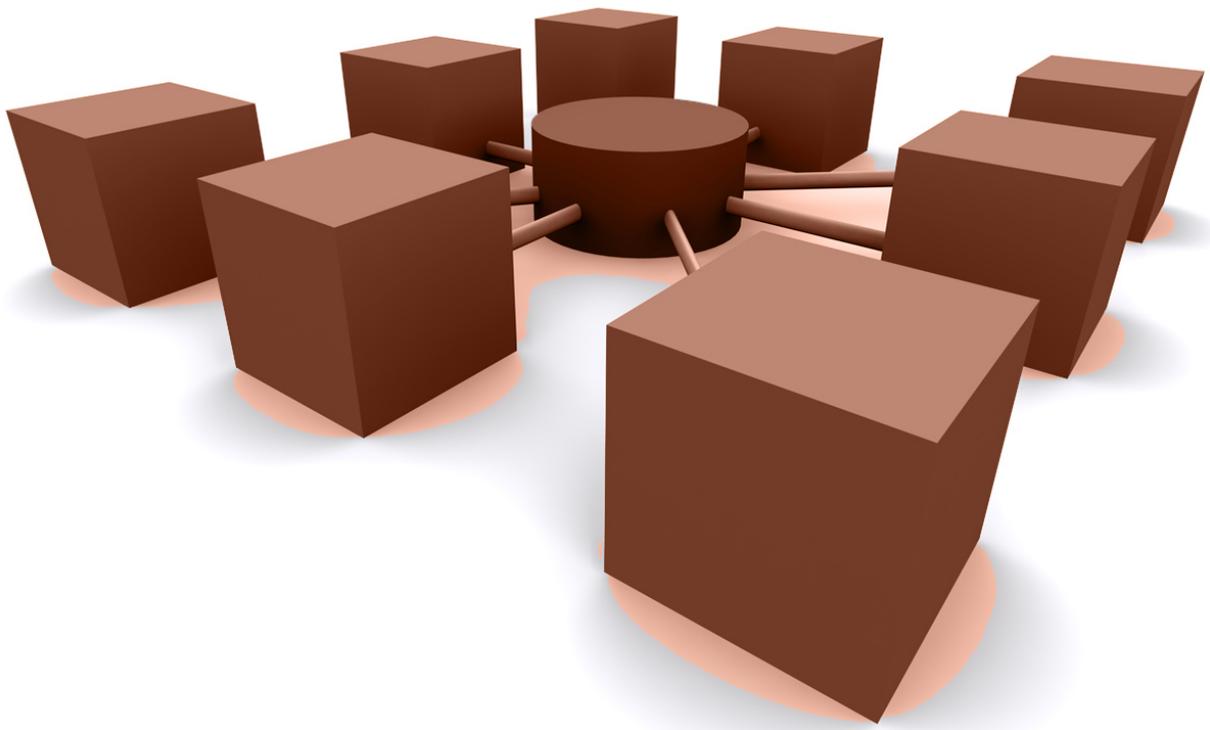


# EVALUACIÓN DE NAGIOS 1.0 PARA LINUX

PLANIFICACIÓN, ESPECIFICACIÓN,  
DISEÑO Y EVALUACIÓN DE REDES



MANUEL DOMÍNGUEZ-DORADO  
JOSÉ ANTONIO ZARANDIETA MORÁN

ISBN: 978-84-615-1184-6



TÍTULO:  
**EVALUACIÓN DE NAGIOS 1.0 PARA LINUX**  
PLANIFICACIÓN, ESPECIFICACIÓN,  
DISEÑO Y EVALUACIÓN DE REDES

AUTORES:  
MANUEL DOMÍNGUEZ-DORADO  
JOSÉ ANTONIO ZARANDIETA MORÁN

EDITA:  
MDD

PRIMERA EDICIÓN  
MAYO DE 2003

ISBN: 978-84-615-1184-6

SI LO DESEA PUEDE PONERSE EN CONTACTO CON LOS AUTORES  
DE LA OBRA EN LAS SIGUIENTES DIRECCIONES:

MANUEL DOMÍNGUEZ-DORADO  
INGENIERO@MANOLODOMINGUEZ.COM

JOSÉ ANTONIO ZARANDIETA MORÁN  
JAZ@UNEX.ES



# Índice

<b>Estructura del estudio</b> .....	6
<b>Descripción introductoria</b> .....	8
Introducción.....	8
Estructura del sistema.....	8
<b>Proceso de instalación</b> .....	10
Requisitos previos.....	10
Creación de usuarios.....	10
Instalación del núcleo de Nagios .....	10
Instalación de las extensiones oficiales .....	12
Activación y configuración de la interfaz web.....	13
<b>Configuración de Nagios</b> .....	15
Servicios .....	15
Equipos.....	16
Grupos de equipos.....	16
Contactos.....	17
Grupos de contactos .....	17
Comandos.....	18
Periodos de tiempo .....	18
Ampliación de los servicios.....	19
Dependencias de los servicios .....	20
Ampliación de los equipos .....	20
Dependencias de los equipos.....	21
Ampliación de los grupos de equipos.....	21
Opciones generales de Nagios.....	21
Configuración de los CGI .....	22
Configuración del acceso al SGBD.....	22
Comprobando la configuración del sistema .....	22
<b>Configuración avanzada de Nagios</b> .....	24
Autenticación para el acceso.....	24
Ejecución de comandos externos .....	26
<b>Extensiones de terceros</b> .....	28
Advanced Performance Addon for Nagios (APAN).....	28
Nagios Administration Tool (NAGAT) .....	28
Nagios Java System (NSJS).....	28
Nagios Remote Plugin Executor (NRPE).....	28
Nagios Service Check Acceptor (NSCA) .....	29
Nagios Statd.....	29

Nagios Watch.....	29
Nmap2Nagios.....	29
NTray.....	29
POM Sender.....	30
Remote Execution Layer.....	30
Remote_ctl.....	30
Repairer.....	30
SNMP Proxy daemon.....	30
Configuración de las extensiones de terceros.....	30
<b>Iniciando y parando Nagios.....</b>	<b>32</b>
<b>Comandos incorporados en Nagios.....</b>	<b>33</b>
check_by_ssh.....	33
check_dig.....	33
check_disk.....	33
check_disk_smb.....	33
check_dns.....	34
check_dummy.....	34
check_flexlm.....	34
check_ftp.....	34
check_hpjd.....	34
check_http.....	34
check_ifoperstatus.....	35
check_ifstatus.....	35
check_imap.....	35
check_ircd.....	35
check_ldap.....	35
check_load.....	35
check_log.....	36
check_mailq.....	36
check_mrtg.....	36
check_mrtgraf.....	36
check_nagios.....	36
negate.....	37
check_nntp.....	37
check_nt.....	37
check_ntp.....	37
check_nwstat.....	37
check_oracle.....	38
check_overcr.....	38
check_pop.....	38
check_procs.....	38
check_real.....	38
check_rpc.....	38
check_sensors.....	39
check_smtp.....	39
check_snmp.....	39
check_ssh.....	39

check_swap.....	39
check_tcp.....	39
check_time.....	40
check_udp.....	40
check_ups.....	40
check_users.....	40
check_vsz.....	40
urlize.....	41
<b>Estudio de las funcionalidades.....</b>	<b>42</b>
<b>Resumen de la interfaz web.....</b>	<b>45</b>
Visión general.....	45
Detalle de los servicios.....	45
Detalles de los equipos.....	46
Estado detallado de un equipo.....	46
Información sobre un equipo.....	47
Información de estado por grupo de equipos.....	47
Problemas con los equipos.....	48
Problemas con los servicios.....	49
Creación de comentarios para equipos.....	49
Administración web de Nagios.....	50
Cola de planificación.....	50
Configuración de informes.....	51
Informe de disponibilidad.....	51
Histograma.....	52
Histórico de eventos.....	52
Contactos.....	53
<b>Conclusiones.....</b>	<b>54</b>
<b>Caso práctico.....</b>	<b>55</b>
Descripción.....	55
Configuración de los CGI.....	55
Configuración de los contactos.....	56
Configuración de los grupos de contactos.....	56
Configuración de equipos.....	56
Configuración de grupos de equipos.....	57
Configuración de los servicios.....	58
Configuración de las dependencias.....	58
Configuración avanzada.....	58
<b>Autores.....</b>	<b>60</b>
<b>Bibliografía y fuentes consultadas.....</b>	<b>61</b>

## Estructura del estudio

El objetivo de este trabajo es ofrecer una visión rápida y general de qué puede ofrecer a un administrador de redes la herramienta libre Nagios para Linux. No intentaremos con este trabajo realizar un manual detallado de funcionamiento de Nagios aunque en algunos aspectos nos centremos a nivel muy detallado.

Así, el estudio tendrá estos apartados importantes:

- **Descripción introductoria:** donde haremos una brevísima y esquemática introducción a la estructura de la aplicación, las partes que la componen y cómo se relacionan entre sí.
- **Proceso de Instalación:** donde haremos una pequeña guía rápida de instalación de los componentes básicos de un sistema Nagios en su versión 1.0. No entraremos en todos los detalles aunque en principio debería servir para poner en marcha el sistema de monitorización desde cero sin muchos problemas.
- **Configuración de Nagios 1.0:** donde se explicará de forma breve qué cosas hay que configurar en Nagios, por qué hay que configurarlas y en qué ficheros se realiza la configuración. No será una guía de configuración sino que debería valer al usuario para comprender de forma sencilla de qué habla la áspera documentación de Nagios cuando explica la configuración de los objetos.
- **Configuración avanzada de Nagios:** donde explicamos un par de opciones avanzadas de configuración, que sin ser demasiado difíciles de configurar, ofrecen bastante funcionalidad nueva al sistema.
- **Extensiones de terceros:** donde mostraremos algunas de las extensiones o módulos que amplían la funcionalidad del sistema Nagios. En gran parte, Nagios adquiere su mayor funcionalidad gracias a estos módulos y la posibilidad de crear fácilmente muchos más.

- **Estudio de las funcionalidades:** aquí es donde analizaremos que funcionalidades permite Nagios, si es estable, si es sencillo, posibles problemas aparecen en su uso, carencias que encontramos, etcétera. Nos centraremos en algunos aspectos más que en otros, según sea necesario.
- **Estudio de la interfaz web:** donde explicamos de una forma somera las opciones que nos ofrece la interfaz web y que nos permite acceder de forma sencilla a todas y cada una de las partes configuradas.
- **Conclusiones:** haremos un balance de las características positivas y negativas que hemos encontrado en Nagios y haremos un pequeño resumen de las mismas. Terminaremos haciendo alguna recomendación con respecto al uso de Nagios.
- **Caso práctico:** crearemos un escenario de monitorización de algunas de las redes de la Escuela Politécnica de Cáceres para mostrar el funcionamiento práctico de la herramienta.

# Descripción introductoria

## Introducción

Nagios es un sistema de monitorización de equipos y de servicios de red, creado para ayudar a los administradores a tener siempre el control de qué está pasando en la red que administran y conocer los problemas que ocurren en la infraestructura que administran antes de que los usuarios de la misma los perciban. Es un sistema complejo y completo en cuanto a sus características que además hace uso en algunos casos de diversos sistemas como por ejemplo sistemas gestores de bases de datos, servidores web, etcétera. Es relativamente complejo de instalar y configurar.

Nagios es un software usado en todo el mundo, que debe correr en sistemas Linux (en la mayoría de los UNIX funciona sin problemas) y que permite extender su funcionalidad con la utilización/creación de extensiones. Está liberado bajo licencia GPL de la Free Software Foundation.

## Estructura del sistema

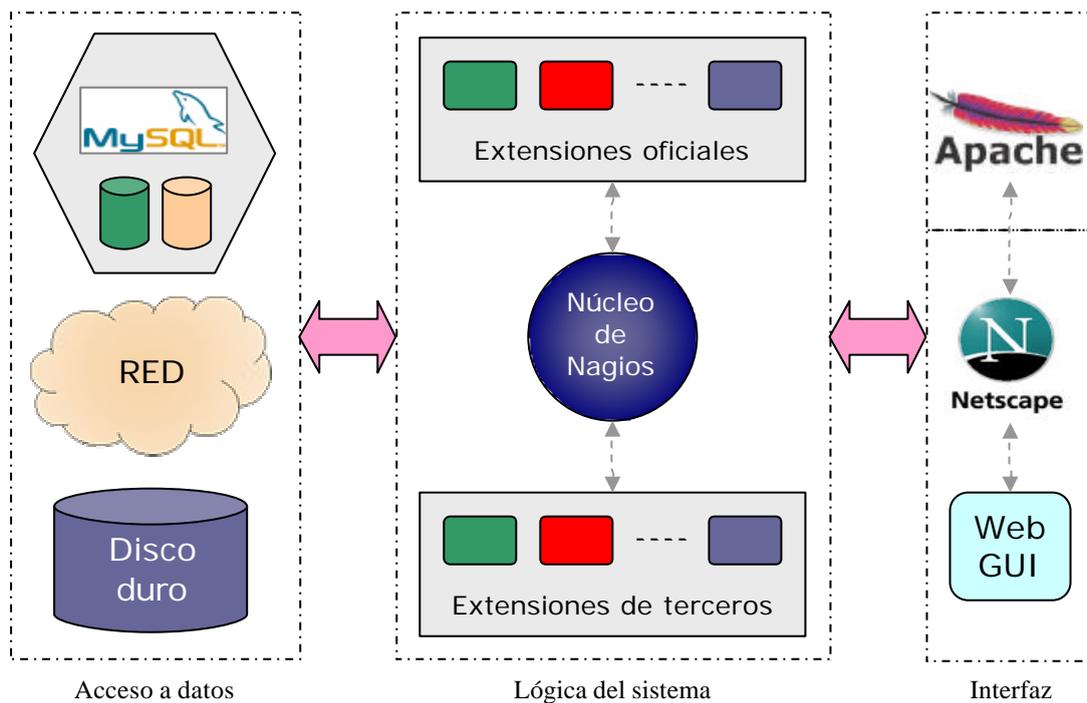
Nagios sigue grosso modo la siguiente estructura:

Un núcleo de la aplicación que forma la lógica de control de negocio de la aplicación contiene el software necesario para realizar la monitorización de los servicios y máquinas de la red para la que está preparado. Hace uso de diversos componentes que vienen con la aplicación, y puede hacer uso de otros componentes realizados por terceras personas.

Aunque permite la captura de paquetes SNMP Trap para notificar sucesos, no es un sistema de monitorización y gestión basado en SNMP sino que realiza su labor basándose en una gran cantidad de pequeños módulos software que realizan chequeos de parte de la red.

Muestra los resultados de la monitorización y del uso de los diversos componentes en una interfaz web a través de un conjunto de CGI's y de un conjunto de páginas HTML que vienen incorporadas de serie. Y que permiten al administrador una completa visión de qué ocurre, dónde y en algunos casos, por qué.

Por último, si se compila para ello, Nagios guardará los históricos en una base de datos para que al detener y reanudar el servicio de monitorización, todos los datos sigan como iban, sin cambios.



Así que en las siguientes páginas, antes de profundizar en el uso de Nagios, vamos a explicar cómo se realiza correctamente la instalación de este sistema, que no es nada trivial; al menos no es algo rápido.

## Proceso de instalación

### Requisitos previos

Es evidente, a tenor de lo visto en el esquema anterior, que es necesario tener una máquina con Linux, el SGBD MySQL (sólo en algunos casos) y el servidor web Apache correctamente instalado, configurado y activo. No es objeto de este estudio el ser un manual de instalación de ninguno de estos softwares por lo que emplazamos al lector a leer la correspondiente documentación si fuera necesario.

A partir de este momento supondremos que se cuenta con un equipo relativamente moderno con todas estas aplicaciones corriendo sobre Linux Red Hat 8.0. En concreto, según los creadores de Nagios, debería tratarse de una máquina con al menos 256 Mb de RAM y 20 Gb de disco duro.

### Creación de usuarios

Nagios utiliza para el proceso de instalación y para otras funciones al usuario *nagios* perteneciente al grupo *nagios* por lo que el primer paso debe ser la creación de dichos usuarios en el Linux. Con la orden siguiente se realizan ambas cosas:

```
adduser nagios
```

### Instalación del núcleo de Nagios

Lo primero que hay que hacer es bajarse el paquete con los fuentes de Nagios de la página web oficial <http://www.nagios.org>. En nuestro caso hemos bajado el paquete *nagios-1.0.tar.gz*, por tanto, la versión 1.0.

Desempaquetamos y descomprimimos el paquete:

```
tar xzvf nagios-1.0.tar.gz
```

el siguiente paso será crear como root el directorio donde se instalará el software una vez compilado. Por defecto es `‘/usr/local/nagios’` así que si no queremos otro lugar, creamos ese, como usuario root:

```
mkdir /usr/local/nagios
```

Ahora procederemos a la configuración de las fuentes para una compilación acorde a nuestro sistema. Tomaremos todas las opciones por defecto por b que procederemos con el siguiente comando desde el directorio donde se ha descomprimido y desempquetado el paquete Nagios.

```
./configure
```

Sólo si hubiésemos creado otro directorio de instalación distinto de `‘/usr/local/nagios’` tendríamos que invocar al guión de configuración de la siguiente forma.

```
./configure --prefix=directorio_elegido
```

Donde `‘directorio_elegido’` es el directorio en el que queremos instalar el software. Tras cualquiera de las dos formas, los fuentes quedarán configuradas para ser compiladas correctamente. Procedemos entonces a la compilación de las mismas. Para ello haremos uso de la herramienta make.

```
make all
```

Que compilará Nagios 1.0 (esa versión en nuestro caso). Para que los ejecutables recién compilados queden instalados en el directorio elegido debemos indicar que se proceda a la instalación.

```
make install
```

Que instalará los ficheros binarios y los ficheros HTML que serán necesarios para el correcto funcionamiento del sistema de monitorización.

Nagios ofrece guiones para ser iniciado durante el arranque del sistema. Para que dichos guiones se instalen, se debe utilizar el siguiente comando.

```
make install-init
```

Con esto quedará instalado el núcleo del sistema. Sin embargo, actualmente este núcleo hace labores de coordinación y gestión entre un número de extensiones que realmente son las que realizan las labores de monitorización.

## Instalación de las extensiones oficiales

Lo siguiente que hay que hacer es instalar todas las extensiones oficiales (plugins) que permitirán la monitorización. Para ello lo primero que hay que hacer es bajarse, de nuevo, el paquete correspondiente de la página oficial <http://www.nagios.org> que en nuestro caso ha sido el paquete llamado *nagios-plugins-1.3.0.tar.gz*, pero no debería variar la instalación para cualquier versión posterior.

Estos plugins son programas compilados o shell scripts que realizan la monitorización de ciertos servicios o máquinas y devuelven unos valores que son los que el núcleo de Nagios utiliza. Para instalarlos, lo primero que hay que hacer es desempaquetar y descomprimir el paquete como se especifica a continuación.

```
tar xzfv nagios-plugins-1.3.0.tar.gz
```

De nuevo hay que configurar las fuentes. El parámetro más importante que hay que especificar en este caso es el lugar donde se encuentra instalado Nagios, puesto que las extensiones se instalan en una rama de dicho árbol de directorios. Si hemos instalado Nagios en su lugar por defecto `/usr/local/nagios` entonces simplemente escribiremos en el directorio donde se ha descomprimido el paquete de plugins el siguiente comando.

```
./configure
```

Sin embargo, si se ha utilizado otro directorio de instalación distinto para Nagios, habrá que invocar a esta orden de la siguiente forma.

```
./configure --prefix=directorio_instalacion_nagios
```

Donde *directorio\_instalacion\_nagios* es el directorio que elegimos en su momento para instalar el sistema de monitorización. De cualquiera de las dos formas, las fuentes de las extensiones quedarán bien configuradas para compilar sin problemas en nuestro sistema. Para proceder a la compilación, ejecutaremos la siguiente orden.

```
make all
```

Tras lo cual los binarios habrán sido correctamente generados si no ha habido ningún problema. Ahora hay que proceder a la instalación de dichos binarios a partir del directorio que le indicamos durante la configuración. Lo hacemos como sigue.

```
make install
```

Con lo cual tendremos instaladas correctamente las extensiones oficiales. En este punto tenemos instalado el sistema primario de Nagios (ver esquema al principio de este documento).

## Activación y configuración de la interfaz web

Es aconsejable (prácticamente obligatorio) activar y configurar la interfaz web de Nagios que nos permitirá interactuar con el sistema de monitorización a través de cualquier navegador. A partir de este momento supondremos que se ha instalado el sistema de monitorización bajo *'/usr/local/nagios'* y que el servidor web Apache está configurado y funcionando correctamente.

Si se han seguido los pasos de la instalación por defecto, sin elegir los parámetros de instalación, los binarios de Nagios debe ser accesible en <http://localhost/nagios/cgi-bin/> pero para ello hay que añadir algunas líneas en el fichero de configuración de apache *http.conf*.

Establecemos antes de nada un alias para poder acceder a través del servidor http a los binarios de nuestra distribución de Nagios. Para ello añadimos lo siguiente al fichero de configuración de apache:

```
ScriptAlias /nagios/cgi-bin/ /usr/local/nagios/sbin/  
<Directory "/usr/local/nagios/sbin/">  
    AllowOverride AuthConfig  
    Options ExecCGI  
    Order allow,deny  
    Allow from all  
</Directory>
```

Del mismo modo, deberemos crear un alias para poder acceder a los ficheros html de nuestra distribución Nagios vía el servidor Apache. Incluimos en el mismo fichero las siguientes líneas sabiendo que **deben estar colocadas después de las que hemos escrito en el paso anterior.**

```
Alias /nagios/ /usr/local/nagios/share/  
<Directory "/usr/local/nagios/share">  
    Options None  
    AllowOverride AuthConfig  
    Order allow,deny  
    Allow from all  
</Directory>
```

Lo que nos permitirá poder acceder a la interfaz web del sistema de monitorización que estamos instalando en <http://localhost/nagios/> que es lo que pretendemos. Para probarlo, deberemos reiniciar el servidor http para que lea la nueva configuración. Esto lo hacemos con la orden siguiente.

```
/etc/rc.d/init.d/httpd restart
```

Y probamos el correcto funcionamiento usando cualquier navegador y probando la dirección <http://localhost/nagios/>.

Según la configuración del servidor web, puede ocurrir que la página no se muestre si no se incluye la barra “/” al final de la URL. En cualquier caso, así se comprobará si funciona correctamente o no la interfaz web, pero nada más funcionará porque no se ha configurado los CGI’s para ello y no se ha levantado el servicio “nagios”.

## Configuración de Nagios

Nagios tiene una configuración bastante larga y meticulosa por lo que en este apartado haremos mención únicamente a alguno de los ficheros que hay que modificar y qué es lo que se configura en cada uno de ellos, instando al lector a que consulte la documentación correspondiente en cada momento para saber más.

Si se han seguido todos los pasos anteriores, en el directorio `/usr/local/nagios/etc` se encontrarán los ficheros de configuración de ejemplos del sistema de monitorización. Bastará con quitarles a todos la parte del nombre en que pone `sample` para tener todos los ficheros de configuración necesarios que habrá que retocar pero que están auto documentados en gran parte.

Nagios puede configurarse exclusivamente en un solo fichero `nagios.cfg`, pero por comodidad y facilidad se suele dividir el fichero de configuración en varios cuyo nombre puede ser elegido por el administrador. A continuación explicamos para qué vale cada uno de ellos.

### Servicios

Se consideran servicios a los procesos tradicionales como STPM, FTP, WWW, etcétera que corren en alguna máquina de la red y también a cualquier tipo de valores como por ejemplo el tiempo de retorno de los PING o cualquier otra métrica parecida que nos interese aplicar. Junto con un servicio se define también el equipo (ver más abajo) en el que debe estar disponible dicho servicio. Todos estos datos se suelen definir en el fichero `services.fcg`.

Algunas de las características más importantes que podemos configurar en este fichero con respecto a los servicios (a cada uno de ellos) son:

- Descripción del servicio: `service_description`.
- Equipo en el que estará funcionando: `host_name`.
- Periodo de tiempo por el cuál se registrará: `check_period`.
- Comando usado para comprobar el servicio: `check_command`.
- Grupo de contacto para notificaciones: `contact_groups`.
- Opciones de las notificaciones: `notification_options`.

## Equipos

Se considera un equipo a cualquier dispositivo que pertenezca a la red como por ejemplo servidores, routers, switches, impresoras de red... Se definen generalmente en el fichero `hosts.conf`.

Algunas de las características más importantes que podemos configurar en este fichero con respecto a los equipos (cada uno de ellos) son:

- Nombre del equipo: `host_name`.
- Alias para el equipo: `alias`.
- Dirección IP del equipo: `address`.
- Comando usado para comprobar el equipo: `check_command`.

## Grupos de equipos

Un grupo de equipos es un conjunto de equipos previamente definidos que tienen algo en común y que por razones de facilidad en la administración se agrupan. Por ejemplo un grupo 'impresoras' podría agrupar a todos los equipos que sean impresoras, el grupo 'Sala Linux' podría contener a todos los equipos de la sala llamada 'Sala Linux', etcétera. Un equipo SIEMPRE tiene que pertenecer a un grupo. Asimismo puede pertenecer a

más de un grupo. Los grupos de equipos se configuran generalmente en `hostgroups.cfg`.

Algunas de las características más importantes que podemos configurar en este fichero con respecto a los grupos de equipos (cada uno de ellos) son:

- Nombre del grupo de equipos: `hostgroup_name`.
- Alias para el grupo: `alias`.
- Contactos para el grupo de equipos: `contact_groups`.
- Miembros del grupo: `members`.

## Contactos

Un contacto es la definición de una persona (generalmente responsable de la red) que debe ser contactado cuando ocurre algún suceso en la red. Junto con la definición del contacto se especifican también las circunstancias bajo las cuales se tienen que producir avisos de alarma: caída del sistema, equipos inaccesibles... Los contactos se configuran generalmente en el fichero `contacts.cfg`.

Algunas de las características más importantes que podemos configurar en este fichero con respecto a los contactos (cada uno de ellos) son:

- Nombre del contacto: `contact_name`.
- Alias para el contacto: `alias`.
- Periodos de notificación de errores de equipos: `host_notification_period`.
- Periodos de notif. de errores servicios: `service_notification_period`.
- Comando para notif. de err. servicios: `service_notification_commands`.
- Comando para notif. de err. equipos: `host_notification_commands`.

## Grupos de contactos

Al igual que los grupos de equipos, los grupos de contactos son un conjunto, en este caso de personas (contactos, en realidad) que se agrupan para permitir una mayor flexibilidad a la hora de que Nagios realice notificaciones. Por ejemplo, el grupo 'Administradores de la sala Linux' podría agrupar a todos los administradores de esa sala y cuando hubiese algún problema que notificar, todos ellos serían notificados a la vez. Se configuran generalmente en `contactgroups.cfg`.

Algunas de las características más importantes que podemos configurar en este fichero con respecto a los grupos de contactos (cada uno de ellos) son:

- Nombre del grupo de contactos: `contactgroup_name`.
- Alias para el grupo: `alias`.
- Miembros del grupo: `members`.

## Comandos

Un comando es una tarea específica que se declara como por ejemplo hacer un PING, hacer un telnet o cualquier otra. Se especifica la línea de comandos y a partir de ese momento ese comando que se ha definido puede ser usado por Nagios. Esto permite mucha flexibilidad ya que podemos añadir nuevas funcionalidades a Nagios sin más que crearnos nuevos comandos. Los comandos se declaran generalmente en `misccommands.cfg` y `checkcommands.cfg`.

Algunas de las características más importantes que podemos configurar en estos ficheros con respecto a los comandos (cada uno de ellos) son:

- Nombre del comando: `command_name`.
- Comando a ejecutar: `command_line`.

## Periodos de tiempo

Un periodo de tiempo es un rango horario que se asigna para cada día de la semana (debe ser siempre semanal) de tal forma que luego ese periodo de tiempo que se ha

creado, se pueda asignar a una tarea concreta y formar así un calendario o agenda. Por ejemplo se puede establecer para cada día de la semana el horario laboral de una empresa y luego llamarle 'Horario de oficina'. Así se podrá decir a cualquier tarea que funcione exclusivamente ajustándose al periodo de tiempo 'Horario de oficina'. Se configuran generalmente en `timeperiods.cfg`.

Algunas de las características más importantes que podemos configurar en este fichero con respecto a los periodos de tiempo (cada uno de ellos) son:

- Nombre del periodo de tiempo: `timeperiod_name`.
- Alias del periodo de tiempo: `alias`.
- Horario para el lunes: `monday`.
- Horario para el martes: `tuesday`.
- Horario para el miércoles: `wednesday`.
- Horario para el jueves: `thursday`.
- Horario para el viernes: `friday`.
- Horario para el sábado: `saturday`.
- Horario para el domingo: `sunday`.

## Ampliación de los servicios

Se usan para intensificar las notificaciones con respecto a un servicio de un equipo. Es completamente opcional. La ampliación de los servicios se configura generalmente en el fichero `escalations.cfg`.

Algunas de las características más importantes que podemos configurar en este fichero con respecto a la ampliación de los servicios (cada uno de ellas) son:

- Nombre del equipo donde está el servicio: `host_name`.
- Descripción del servicio: `service_description`.
- Primera notificación: `first_notification`.
- Última notificación: `last_notification`.

## Dependencias de los servicios

Una dependencia de servicio es una característica avanzada de Nagios que permite que en una jerarquía donde unos servicios dependen de otros, al fallar uno del que dependen varios, se supriman las notificaciones ocurridas por el mal funcionamiento de estos últimos ya que se deben a que ha fallado aquel del que dependen, y no ellos. También se puede especificar que los servicios dependientes no sean comprobados inútilmente en esta situación. Está especialmente indicado para expertos que tienen que monitorizar instalaciones excesivamente complejas. Se especifican generalmente en el fichero `dependencies.cfg`.

Algunas de las características más importantes que podemos configurar en este fichero con respecto a la dependencia de los servicios (cada uno de ellas) son:

- Descripción del servicio: `service_description`.
- Equipo donde está el servicio: `host_name`.
- Descripción del servicio dependiente: `dependent_service_description`.
- Equipo del servicio dependiente: `dependent_host_name`.

## Ampliación de los equipos

Se usan para intensificar las notificaciones con respecto a un equipo. Es completamente opcional. La ampliación de los equipos se configura generalmente en el fichero `escalations.cfg`.

Algunas de las características más importantes que podemos configurar en este fichero con respecto a la ampliación de los equipos (cada uno de ellas) son:

- Nombre del equipo: `host_name`.
- Primera notificación: `first_notification`.
- Última notificación: `last_notification`.

## Dependencias de los equipos

Una dependencia de equipo es una característica avanzada de Nagios que permite que cuando un equipo del que dependen varios falla, no se notifique el fallo de los equipos dependientes ni se intente comprobar su estado, porque no han fallado realmente sino que ha fallado el equipo del cual dependen. Está especialmente indicado para expertos que tienen que monitorizar instalaciones excesivamente complejas. Se especifican generalmente en el fichero `dependencies.cfg`.

Algunas de las características más importantes que podemos configurar en este fichero con respecto a la dependencia de los equipos (cada uno de ellas) son:

- Equipo donde está el servicio: `host_name`.
- Equipo del servicio dependiente: `dependent_host_name`.

## Ampliación de los grupos de equipos

Se usan para intensificar las notificaciones con respecto a todos los equipo de un grupo de equipos. Es completamente opcional. La ampliación de los grupos de equipos se configura en el fichero `escalations.cfg`.

Algunas de las características más importantes que podemos configurar en este fichero con respecto a la ampliación de los grupos de equipos (cada uno de ellas) son:

- Nombre del grupo de equipos: `hostgroup_name`.
- Primera notificación: `first_notification`.
- Última notificación: `last_notification`.

## Opciones generales de Nagios

Aparte de lo anteriormente visto, el fichero principal de configuración de Nagios es obligatoriamente `nagios.cfg` que no debería contener muchas cosas si hemos configurado cada parte en un fichero independiente, como se ha visto en las líneas

anteriores. Este fichero contiene información sobre cuáles son los ficheros de configuración (los de los párrafos anteriores) que hemos usado para configurar cada parte, qué grupo y/o usuario hacen funcionar Nagios, fichero de sucesos, etcétera.

## Configuración de los CGI

Los CGI que Nagios utiliza para su funcionamiento y para la presentación web de los datos referentes a la monitorización se configuran en el fichero `cgi.cfg`. Generalmente este fichero contiene información relativa a la localización de los HTML que se muestran, a los CGI, a la configuración del acceso a los servicios, a los equipos y a la propia configuración de Nagios tanto para obtener información como para actuar sobre el sistema.

## Configuración del acceso al SGBD

## Comprobando la configuración del sistema

Una vez que se han introducido todos los datos correctos en los distintos ficheros de configuración de Nagios, no es necesario pero si recomendable verificar que no existen errores de configuración. Para ello, usaremos una opción del ejecutable de nagios que nos permitirá conocer si hay errores o no y, caso de haberlos, nos indicará el lugar concreto dentro del fichero de configuración donde puede estar el error.

```
nagios -v /usr/local/nagios/etc/nagios.cfg
```

Nagios comprobará que se han especificado todos los objetos correctamente. Además comprobará que se han declarado en el orden correspondiente y necesario para el buen funcionamiento de todo el sistema, el siguiente:

1. Verificar que todos los contactos son grupos al menos de un grupo de contacto.
2. Verificar que todos los miembros de un grupo de contacto son contactos válidos.

3. Verificar que todos los equipos son miembros de al menos un grupo de equipos.
4. Verificar que todos los equipos especificados en un grupo de equipos son equipos válidos.
5. Verificar que todos los equipos tienen al menos un servicio asociado a ellos.
6. Verificar que todos los comandos usados en los servicios y los equipos, son válidos.
7. Verificar que todos los comandos usados en los manejadores de eventos de servicios y equipos, son válidos.
8. Verificar que todos los comandos usados para notificaciones de contactos, equipos y servicios, son válidos.
9. Verificar que todos los periodos de tiempos usados para servicios, equipos y contactos, son válidos.
10. Verificar que todos los periodos de tiempos para comprobación de servicios, son válidos.

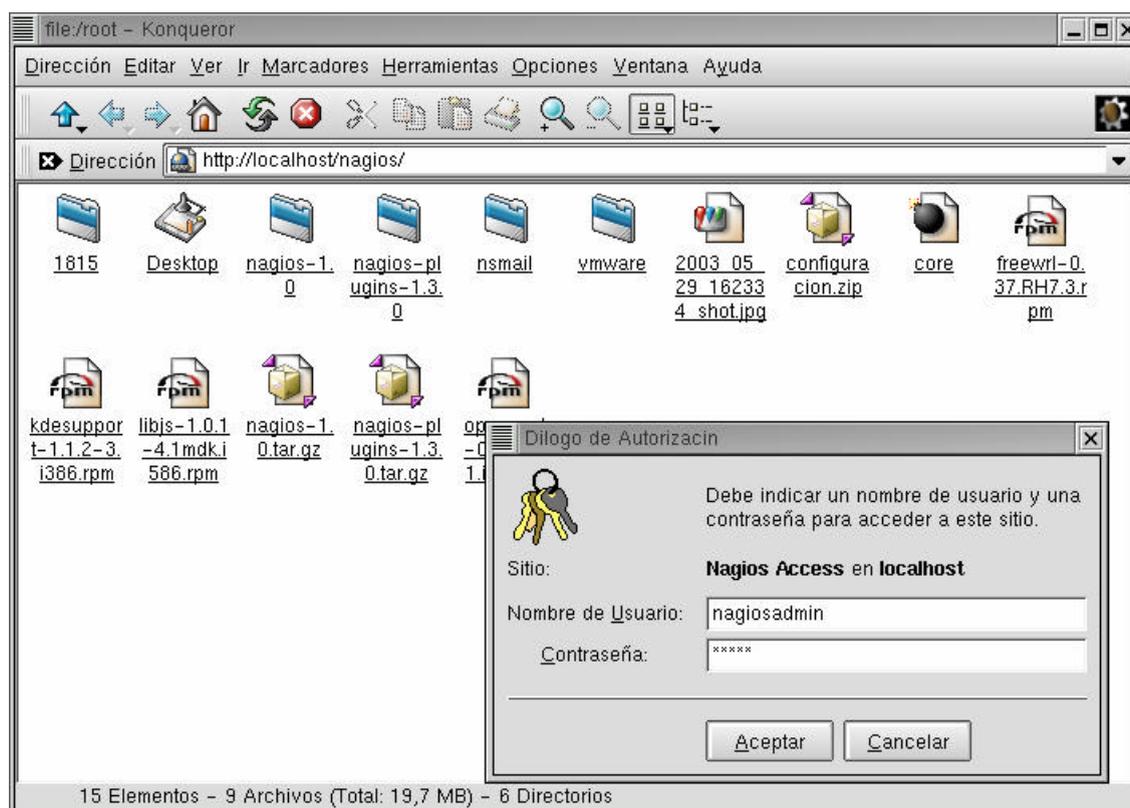
Si el comando anterior no devuelve ningún error, significará que nuestros ficheros de configuración son correctos sintácticamente hablando y habremos terminado, pudiendo hacer funcionar el sistema en cualquier momento.

## Configuración avanzada de Nagios

Nagios es una herramienta extremadamente compleja y tiene muchísimas opciones. Dentro de la parte avanzada del sistema, hay posibilidades para montar sistemas de monitorización distribuido, planificar caídas del sistema, realizar chequeos pasivos de servicios y equipos, etcétera. A continuación vamos a comentar sólo dos aspectos que nos parecen interesantes y que añaden funcionalidades importantes al sistema.

### Autenticación para el acceso

Esta opción nos permite montar Nagios como un sistema de monitorización al que se puede acceder de forma remota sin peligro a que cualquier persona acceda y modifique, observe, etcétera.



Se puede especificar autenticación tanto para acceder al sistema como para acceder al uso de los CGI's. Este proceso se realiza como normalmente se hace en un servidor HTTP cualquiera. Con las líneas que configuramos en un principio en el fichero

`httpd.conf` de Apache el servidor estará ya preparado. El siguiente paso es crear los siguientes ficheros:

```
/usr/local/nagios/sbin/.htaccess
/usr/local/nagios/share/.htaccess
```

Ambos con el siguiente contenido:

```
AuthName "Acceso a Nagios"
AuthType Basic
AuthUserFile /usr/local/nagios/etc/htpasswd.users
require valid-user
```

Tras esto, tenemos que crear al menos un usuario que será el administrado del sistema Nagios y que tendrá los permisos necesarios para ejecutar CGI's, ver el sistema y otras opciones. Hacemos lo siguiente:

```
htpasswd -c /usr/local/nagios/etc/htpasswd.users <username>
```

Donde `<username>` es el nombre de usuario que queremos crear. Tras esto se nos pedirá que introduzcamos la clave para ese usuario. Lo hacemos y ya tenemos creado nuestro administrador del sistema que tendrá privilegios.

Posteriormente, se debe modificar una serie de directivas del fichero de configuración de los CGI de Nagios, `/usr/local/nagios/etc/cgi.cfg`. En concreto, se deben de colocar las siguientes líneas (si existen ya con otro valor, comentarlas con #).

```
use_authentication=1
authorized_for_system_information=<username>
authorized_for_system_commands=<username>
authorized_for_configuration_information=<username>
authorized_for_all_hosts=<username>
authorized_for_all_host_commands=<username>
authorized_for_all_services=<username>
```

```
authorized_for_all_service_commands=<username>
```

En todos los casos *<username>* es el nombre de usuario del usuario que hemos creado unos pasos arriba. Se podrían crear distintos usuarios con distintos privilegios. En este caso hemos configurado un solo administrador con todos los privilegios.

Después de haber seguido todos estos pasos, al intentar acceder a Nagios no aparecerá la ventana de autenticación donde deberemos introducir el nombre de usuario y la clave del administrador. ¡Hay que reiniciar el servicio Nagios para ver los cambios!

## Ejecución de comandos externos

Otra característica avanzada de Nagios es que permite vía web la ejecución de ciertas tareas más allá del propio conjunto de CGI's que vienen de serie, como por ejemplo la caída o el reinicio del propio Nagios, etcétera.

Para poder ejecutar este tipo de comandos es necesario también configurar el sistema de una forma un tanto especial. No hay que olvidar que al configurar Nagios de este modo se está permitiendo desde la web activar o desactivar opciones que en principio sólo estaban disponibles desde la consola del sistema.



Para configurar Nagios de esta forma, hay que editar el fichero principal `/usr/local/nagios/etc/nagios.cfg` y añadir (o modificar si ya existen) las siguientes líneas:

```
check_external_commands=1
command_check_interval=-1
command_file=/usr/local/nagios/var/rw/nagios.cmd
```

Lo que hará que Nagios active el chequeo para buscar comandos externos, con tanta frecuencia como sea posible por el sistema y buscará los comandos en el fichero

`/usr/local/nagios/var/rw/nagios.cmd` que hará de cola. El fichero *nagios.cmd* no tiene que existir, se crea solo. Sin embargo el directorio donde reside, `/usr/local/nagios/var/rw` debe tener permisos de lectura y escritura para el usuario *nagios* y el grupo *nagios*, que deben ser los dueños del mismo. Si no existe dicho directorio lo creamos y si existe pero no tiene los permisos, cambiamos los mismos.

## Extensiones de terceros

Nagios tiene un gran número de colaboradores debido a que es software libre además de ser un producto muy completo y funcional (y barato). Así hay extensiones creadas por estos colaboradores y que se pueden bajar de la misma página del producto, para casi todas las cosas que se pueda imaginar. GUI's escritos en C++ para la monitorización del sistema de monitorización (paradójico), extensiones para poder medir la temperatura de los equipos y las salas donde se encuentran desde el monitor, etcétera.

Comentaremos a continuación el nombre y la descripción de las extensiones de terceros que se pueden bajar de la página web oficial de Nagios, para que el lector se haga una idea de las funcionalidades añadidas con las que podrá contar para ampliar o mejorar las capacidades del sistema de monitorización. Son, entre otras, las siguientes:

### Advanced Performance Addon for Nagios (APAN)

Es una extensión que facilita la integración de la herramienta DRR (ver más abajo) con el sistema Nagios. Su propósito es recoger de forma sencilla estadísticas de los servicios monitorizados por nagios y presentarlos vía web de una forma gráfica y amena.

### Nagios Administration Tool (NAGAT)

Es una herramienta creada en PHP que permite vía web la configuración de todos los objetos que hemos estado viendo; equipos, contactos, servicios...

### Nagios Java System (NSJS)

Es una herramienta compuesta por un cliente y un servidor, ambos escritos en Java, cuyo cometido es permitir a los usuarios ejecutar software de monitorización en sus equipos y enviar los resultados al sistema de monitorización central.

### Nagios Remote Plugin Executor (NRPE)

Colocado en las máquinas de la red, permite al sistema de monitorización ejecutar de forma remota comprobaciones de los sistemas como por ejemplo procesos, discos, etcétera que de otra forma no se podría hacer.

### Nagios Service Check Acceptor (NSCA)

Esta utilidad acepta los resultados de cualquier programa instalado en las máquinas de la red que sepa comunicarse con ella. Generalmente se usa en conjunción con NRPE para la monitorización de discos, memoria, etcétera, pero en la parte del servidor. Como característica importante hay que decir que permite la transmisión de dichos datos de forma segura usando multitud de algoritmos de cifrado.

### Nagios Statd

Esta extensión es un *daemon* escrito en Python y un cliente que permiten consultar remotamente información relativa a la máquina que los ejecute, por ejemplo usuarios, carga del sistema, sistema de ficheros, etcétera.

### Nagios Watch

Esta extensión es una interfaz gráfica escrita en Perl y las librerías GTK que tiene como utilidad mostrar el estado de Nagios.

### Nmap2Nagios

Es una extensión para generar ficheros de configuración de los objetos de Nagios basados en plantillas a partir de la salida XML de nmap.

### NTray

Esta extensión es una aplicación para Windows que queda residente en la barra del escritorio y muestra mediante colores el estado de los servicios que están siendo monitorizados por Nagios. Permite comunicación vía SSL.

## POM Sender

Esta extensión permite la generación y envío de eventos hacia un servidor BMC Patrol Operations Manager.

## Remote Execution Layer

Otra extensión que permite el envío de los datos resultantes de las comprobaciones en los equipos remotos hacia el sistema de monitorización Nagios. Realiza la transmisión de los datos usando el mail para ello por lo que puede fácilmente pasar cortafuegos que con otras extensiones no se pueden evitar.

## Remote\_ctl

Consiste en un CGI escrito en Perl que permite la activación y desactivación de las comprobaciones de servicios a través de un servidor web (una página web, clientes MIDP, etcétera).

## Repairer

Esta extensión añade un manejador de eventos para Nagios además de un agente SNMP que están diseñados para reparar servicios de forma automática en Red Hat Linux.

## SNMP Proxy daemon

Esta extensión es un servidor proxy para SNMP y un agente SNMP que permite la consulta del estado de las máquinas remotas.

## Configuración de las extensiones de terceros

Las extensiones de terceros son programas y utilidades que deben ser bajadas, compiladas e instaladas de forma separada a Nagios y por tanto con cada una de ellas ha de bajarse las instrucciones de compilación, instalación y configuración, que sobrepasan las aspiraciones de este documento.

## Iniciando y parando Nagios

Si se ha instalado Nagios como se ha ido explicando en este documento, se deberá haber instalado un script de inicio en `/etc/rc.d/init.d` llamado 'nagios'. La forma de iniciar el sistema de monitorización es como tradicionalmente se hace para los servicios Linux:

```
/etc/rc.d/init.d/nagios start
```

Tras lo cual ya podremos comenzar vía web a ver qué está ocurriendo, si todo funciona, hacer uso de los CGI, etcétera. Del mismo modo, cuando queramos desactivar el sistema lo haremos como:

```
/etc/rc.d/init.d/nagios stop
```

Lo cual terminará el proceso de monitorización. Por último, si se desea reiniciar el sistema Nagios para leer una nueva configuración, lo haremos como sigue:

```
/etc/rc.d/init.d/nagios reload
```

## Comandos incorporados en Nagios

Aunque ya hemos comentado que Nagios es fácilmente ampliable al poder diseñar comandos específicos para testar y monitorizar equipos y servicios, la verdad es que el propio sistema incorpora un gran número de comandos ya diseñados y funcionales que se pueden utilizar. Algunos de ellos hacen uso de librerías y paquetes que deben estar instalados en el sistema. Remitimos al lector a que lea el fichero REQUERIMENTS que se encuentra en el directorio de estas utilidades y donde se especifica qué debe estar instalado y de donde se puede bajar para usar según qué comando. En las siguientes líneas se comentan someramente cuales son los comandos que vienen de serie y que hacen:

`check_by_ssh`

Este comando es muy interesante interesante. Permite ejecutar comandos en ordenadores remotos vía SSH (de forma segura, por tanto). El resultado de ese comando será tomado por Nagios.

`check_dig`

Este comando sirve para comprobar el funcionamiento del servicio de DNS en un equipo remoto. Utiliza *dig* para este menester.

`check_disk`

Este comando sirve para comprobar el espacio libre de un volumen montado en el sistema de ficheros donde se esté ejecutando Nagios. Permite especificar dos umbrales y generar disparadores advertencias cuando se supera el menor, y errores críticos cuando se supera el segundo.

`check_disk_smb`

Este comando funciona exactamente igual que *check\_disk* pero realiza la comprobación utilizando *samba* para realizar la comprobación de volúmenes compartidos en quipos remotos, en redes Windows.

### check\_dns

Este comando permite hacer una consulta DNS para averiguar la dirección IP de un equipo dado el nombre o viceversa. Utiliza *nslookup* para ello; permite especificar el servidor DNS a usar o si no usa el especificado en */etc/resolv.conf*.

### check\_dummy

Este comando permite realizar una consulta a un dispositivo ficticio (devuelve el mismo parámetro que se le pasa). Puede ser utilizado para comprobaciones y depuraciones.

### check\_flexlm

Este comando comprueba el funcionamiento de un sistema FlexLM. Este sistema es un servidor de licencias en red usado para obtener permisos de uso de software en red. Devuelve distintos errores dependiendo del estado de estos servidores de licencias.

### check\_ftp

Este comando realiza comprobaciones de conexión a un servidor FTP remoto. Permite conocer el estado de este servicio.

### check\_hpjd

Este comando comprueba el estado de una impresora HP que incorpore una tarjeta JetDirect. Realiza este cometido usando para ello el protocolo SNMP.

### check\_http

Este comando comprueba servicios HTTP y HTTPS en equipos remotos. Permite además realizar el seguimiento de redirecciones, tiempos de conexión, la expiración de los certificados para SSL, etcétera. Es especialmente útil para servidores web que sirvan de base para aplicaciones de comercio electrónico.

check\_ifoperstatus

Este comando comprueba el estado de operación de interfaces de red remotas.

check\_ifstatus

Este comando comprueba el estado general de interfaces de red remotas.

check\_imap

Este comando realiza conexiones contra un servidor IMAP para comprobar su estado de funcionamiento. Permite generar advertencias y errores críticos.

check\_ircd

Este comando comprueba el funcionamiento de un servidor de IRC remoto. Realiza conexiones para ello.

check\_ldap

Este comando realiza conexiones y búsquedas LDAP contra un servidor remoto y comprueba así su estado de funcionamiento y si responde dentro del tiempo esperado o no.

check\_load

Este comando trabaja en local en la máquina que está ejecutando el sistema Nagios. Comprueba la carga del sistema en función de unos umbrales que tiene preestablecidos y permite generar advertencias o errores severos según sea esta carga.

### check\_log

Este comando es muy interesante para administradores del sistema. Funciona en local y permite buscar coincidencia de patrones en ficheros de suceso. Cuando el patrón que se busca es encontrado, Nagios recoge esta incidencia.

### check\_mailq

Este comando funciona en local en la máquina que corre Nagios. Permite comprobar el número de mensajes que hay en espera en las colas de *Sendmail*. Se puede establecer un límite para que se genere una notificación en ese caso.

### check\_mrtg

Este comando también trabaja en local en la máquina que está ejecutando Nagios y permite monitorizar los ficheros de sucesos de MRTG; en concreto permite monitorizar cualquiera de los parámetros que se vuelcan sobre dichos ficheros como por ejemplo conexiones, carga del procesador, entrada, salida, etcétera. Permite establecer umbrales que si se superan generan notificaciones.

### check\_mrtgtraf

Este comando permite comprobar el servicio UPS en un equipo remoto y establecer umbrales para, según el valor devuelto, disparar una advertencia, un error severo o nada.

### check\_nagios

Este comando se ejecuta en la máquina que está ejecutando Nagios y permite comprobar que el archivo de sucesos del sistema de monitorización no sea más antiguo de lo que se especifique.

negate

Este comando sirve para, en combinación con cualquiera de los otros plugins, negar su valor. Por ejemplo, el uso normal del comando *check\_ftp* es que devuelve OK cuando el servicio esté funcionando y CRITICAL cuando no. Con este comando se invierten los valores. Es útil para cuando se desea tener notificación explícita de que algo está funcionando bien en lugar de cuando falla.

check\_nntp

Este comando establece conexiones NNTP contra un servidor remoto especificado para comprobar que el servicio de NEWS esté activo.

check\_nt

Este comando realiza peticiones a un equipo Windows NT/2000/XP remoto que esté ejecutando el servicio NSClient para comprobar parámetros locales a dicho equipo como por ejemplo uso de la CPU, de la memoria, del disco, etcétera.

check\_ntp

Este comando ejecuta *ntpdate* para comprobar que el *timestamp* de la máquina local que ejecuta Nagios no difiere en más de lo especificado del *timestamp* de una máquina remota dado.

check\_nwstat

Este comando realiza peticiones a un equipo Novell remoto que esté ejecutando el servicio MRTGEXT NLM para comprobar parámetros locales a dicho equipo como por ejemplo uso de la CPU, de la memoria, del disco, etcétera.

check\_oracle

Este comando permite comprobar el estado de un SGBD Oracle en un ordenador remoto así como el estado de los *tablespaces*, de bases de datos, de las caché, etcétera, de dicho servidor.

check\_overcr

Este comando permite comprobar el estado de un servicio Over-CR ejecutándose en un sistema UNIX remoto. Realiza peticiones a este servicio para comprobar su estado.

check\_pop

Este comando comprueba si el servicio POP de un equipo remoto está funcionando correctamente. Realiza peticiones para ello.

check\_procs

Este comando funciona en la máquina donde se está ejecutando Nagios. Comprueba el número de procesos que se están ejecutando en la máquina y genera advertencias cuando este número sobrepasa el umbral especificado.

check\_real

Este comando comprueba si el servicio REAL de un equipo remoto está funcionando correctamente. Realiza peticiones para ello.

check\_rpc

Este comando comprueba si un servicio RPC remoto está registrado y funcionando correctamente. Utiliza para ello llamadas a *rpcinfo*.

check\_sensors

Este comando funciona en la máquina local donde se ejecuta Nagios; necesita de paquetes adicionales instalados en el sistema de monitorización y su función es comprobar el estado del hardware de la máquina.

check\_smtp

Este comando permite conocer el estado de un servicio SNMP de una máquina remota. Realiza conexiones a este servicio para averiguar la información necesaria.

check\_snmp

Este comando permite conocer el estado de una máquina remota mediante la consulta a su agente SNMP. Utiliza para ello el protocolo SNMP en cualquiera de sus versiones 1, 2 ó 3.

check\_ssh

Este comando permite controlar si el servicio SSH de una máquina remota está activo o no. Realiza peticiones a este servicio para obtener la información necesaria.

check\_swap

Este comando funciona en local, en la máquina donde está instalado Nagios. Permite monitorizar el tamaño de la memoria de intercambio utilizada y generar advertencias o errores cuando este valor sobrepasa los umbrales establecidos.

check\_tcp

Este comando permite realizar peticiones arbitrarias a conexiones (*sockets*) TCP contra sistemas remotos. Por tanto permite monitorizar cualquier servicio que utilice *sockets* TCP para recibir peticiones.

check\_time

Este comando permite comprobar si el servicio de hora (TIME) está funcionando en una máquina remota. Realiza conexiones a este servicio para obtener la información.

check\_udp

Este comando permite realizar peticiones arbitrarias a conexiones (*sockets*) UDP contra sistemas remotos. Por tanto permite monitorizar cualquier servicio que utilice *sockets* UDP para recibir peticiones.

check\_ups

Este comando permite monitorizar el estado del servicio UPS en máquinas remotas; para ello hace peticiones a este servicio. Necesita paquetes adicionales instalados en el sistema de monitorización.

check\_users

Este comando permite conocer el número de usuarios conectados actualmente en el sistema local, en el que se está ejecutando Nagios. Genera advertencias y errores cuando el número supera el umbral fijado.

check\_vsz

Este comando permite comprobar que el tamaño en memoria de un programa determinado no sea mayor de un límite fijado. Cuando se produzca el caso contrario se generarán advertencias y/o errores.

urlize

Este comando permite, usando con otro comando, que la salida de este último se pueda mostrar en la pantalla de un navegador en formato HTML como un enlace hipertexto navegable.

Estos son los comandos que acompañan a Nagios y que ser invocados cada uno con sus respectivos parámetros y su forma de ejecución. Para saber más acerca de estos datos, necesarios para el uso de los comandos, se pueden invocar en línea de comandos con el parámetro `-h` lo que mostrará en la pantalla una ventana de descripción del comando, los parámetros que usa y cómo se invoca.

Por ejemplo, de la forma:

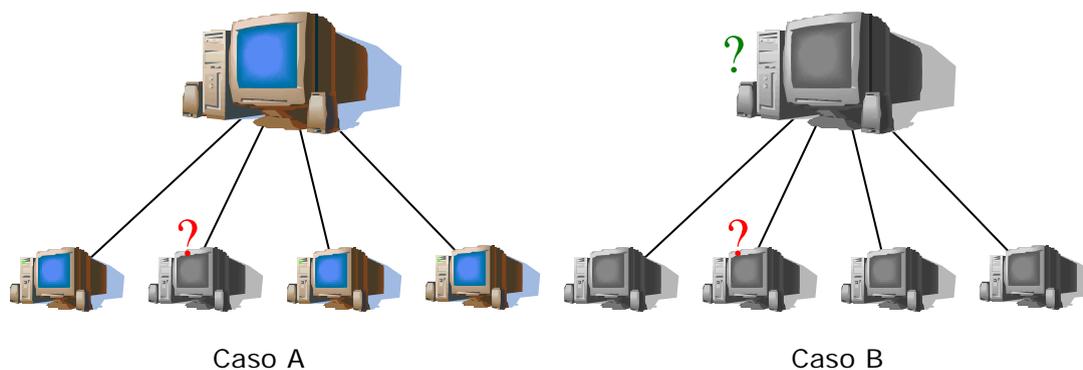
```
./comando_que_sea -h
```

## Estudio de las funcionalidades

Ya hemos comentado qué es Nagios: básicamente es un sistema que testa servicios y otros parámetros de una red de muy diversas formas y notifica todas las incidencias rápidamente a los administradores. Es por tanto un sistema de alerta temprana.

Muestra la información en una interfaz web desde la que el propio administrador puede establecer algunos parámetros, lo que lo hace interesante pues permite observar esta interfaz de forma remota vía un cliente HTTP. Incluso desde dicha interfaz web, previa autenticación HTTP, permite también programar en el tiempo los chequeos a máquinas o servicios previamente configurados, las notificaciones, etcétera.

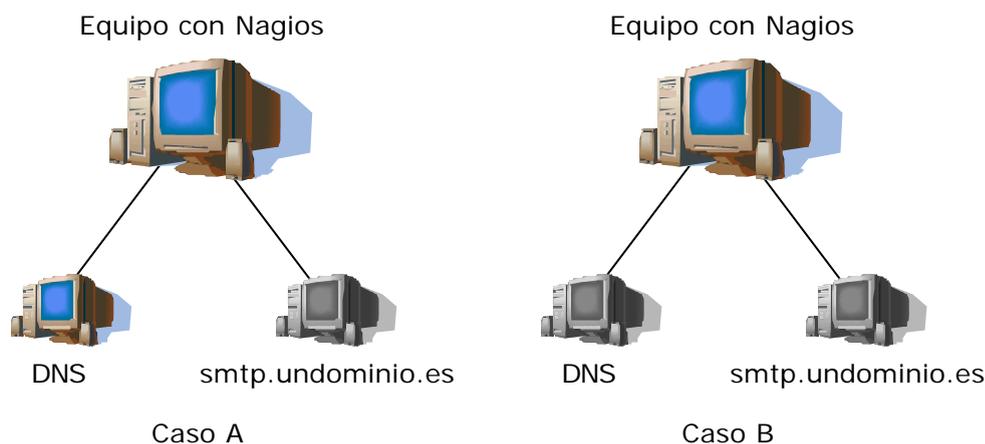
Incorpora características muy interesantes como por ejemplo las dependencias de servicios o de equipos que permiten establecer jerarquías de servicios o de máquinas. De esta forma Nagios puede detectar si un servicio está inactivo o inaccesible; en el primer caso el equipo o servicio estaría *down*, mientras que en el segundo caso, el estado del servicio o equipo no se sabría porque la caída de uno superior impide su monitorización. En la siguiente figura se puede observar este caso:



En el caso A, el equipo marcado con un punto rojo no es accesible porque está caído. Sin embargo, en el caso B el mismo equipo no es accesible porque el servidor del que depende está caído, pero el puede estar perfectamente, aunque no se pueda comprobar. Esto es difícil detectar si el sistema de monitorización no permite establecer jerarquías de equipos. Nagios no tiene problemas en este aspecto. En el caso A notificaría al

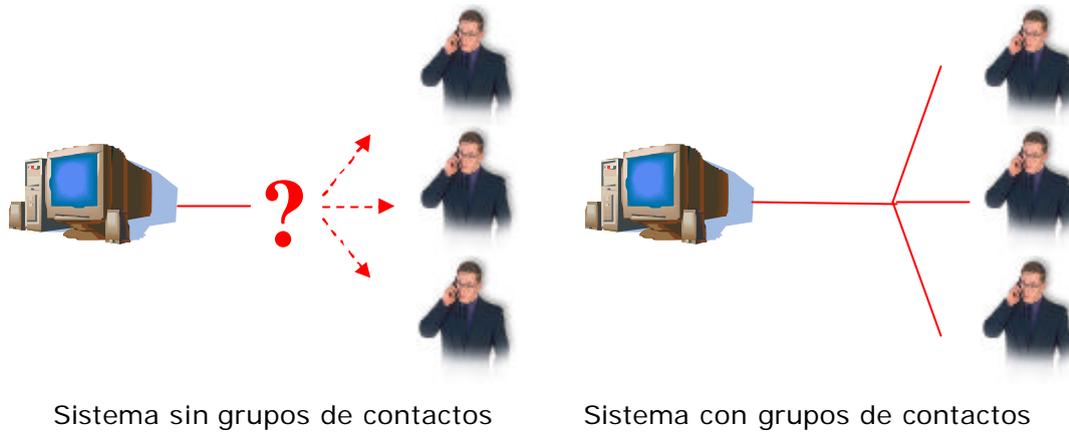
administrador que el equipo marcado en rojo está caído. En el caso sólo B indicaría al administrador, que el equipo marcado con verde ha caído.

El mismo caso que ocurre con la monitorización de equipos ocurre con la monitorización de servicios; veamos el siguiente caso:



En el caso A, un equipo de la red que quiera conectar con el servidor de correo saliente no podrá porque está caído. Nagios detecta el problema y notifica al administrador. En el caso B el servidor de correo podría estar o no activo, pero está inaccesible para los equipos de la red porque el servidor DNS está caído y es él el encargado de traducir el nombre `smtp.undominio.es` a su correspondiente IP. Si se ha establecido una dependencia entre el servicio de correo saliente y el servicio de DNS, Nagios notificará al administrador que el servidor DNS está caído y no mencionará que el servidor SMTP está inaccesible a no ser que, cuando vuelva a funcionar el DNS, `smtp.undominio.es` siga sin poder accederse.

Otra característica que ofrece, bastante importante es la de poder agrupar diversos contactos (personas a quién notificar) en grupos de tal forma que cuando una notificación se produzca para equipos o servicios supervisados por esas personas, dicha notificación llegue a todos y cada uno de ellos y no exclusivamente a una persona. Esto proporciona flexibilidad si por ejemplo la administración de la red se realiza en jornadas divididas por turnos. En ese caso, si el sistema no permite esta característica... ¿a quien se notifica? Se corre el riesgo de notificar a una persona que en ese momento no se encuentra en su jornada laboral.



En un sistema con grupos de contactos la notificación llegará a todos ellos. Sólo responderá el que deba hacerlo, pero así se asegura que alguno de los responsables acudirá a solucionar el problema. En sistemas sin esta característica se corre el riesgo, al enviar una sola notificación, de que se le envía a una persona que no esté disponible en ese momento y el fallo perdure.

Nagios también permite la creación sencilla de nuevos comandos (llamados *plugins*) para añadir nuevas funcionalidades al sistema, o bien combinar varios de los que se encuentran activos. En cierto modo Nagios puede ser tan flexible como se desee tanto en cuanto es software libre y por tanto el código fuente es abierto y modificable por cualquiera.

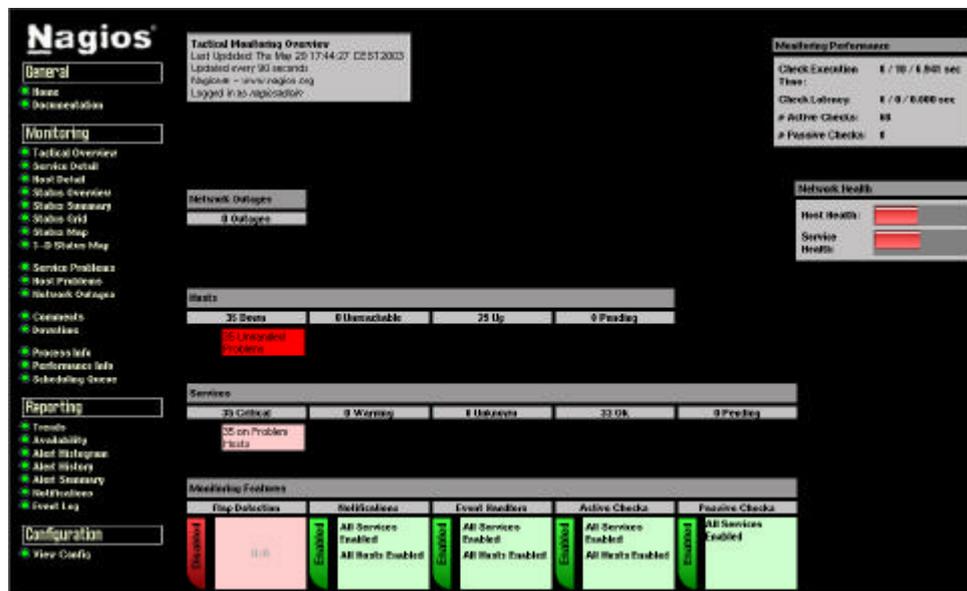
## Resumen de la interfaz web

La web de administración de Nagios es complicada y sencilla a la vez. Complicada porque tiene infinitas posibilidades y sencilla porque todas ellas se hacen de la misma forma con lo cual saber utilizar la interfaz web es cuestión de minutos.

Nagios tiene principalmente servicios y máquinas y en la interfaz web muestra distinta información sobre ellos. Además cada vez que aparece un servicio o equipo en la web, lo hace en forma de enlace por lo que pinchando sobre dicho enlace se conoce más sobre ese servicio o máquina (según el caso). A la izquierda está el menú de opciones que aparecen a la derecha. Es sencillo, rápido e intuitivo.

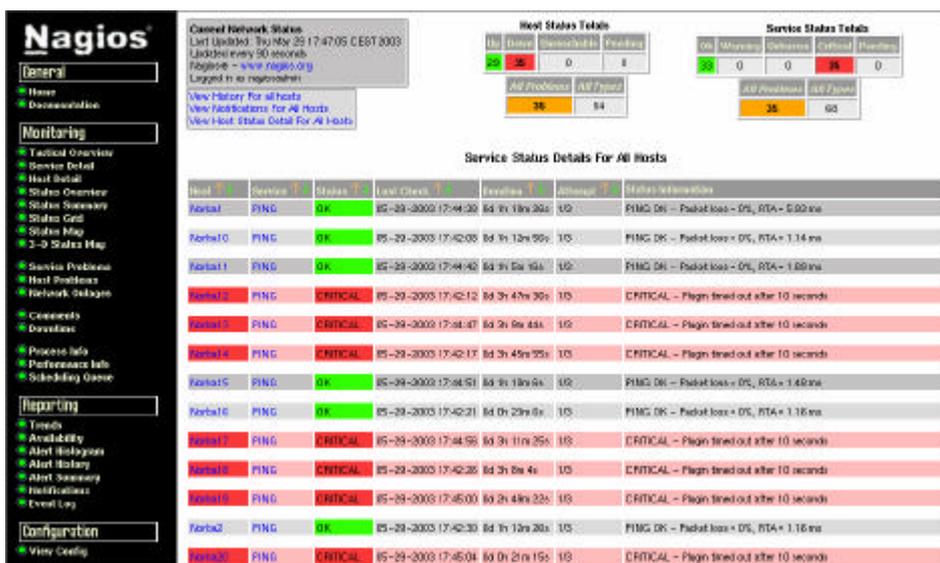
A continuación mostramos capturas de las partes más significativas de la misma, además de una pequeña explicación sobre cada una de ellas.

### Visión general



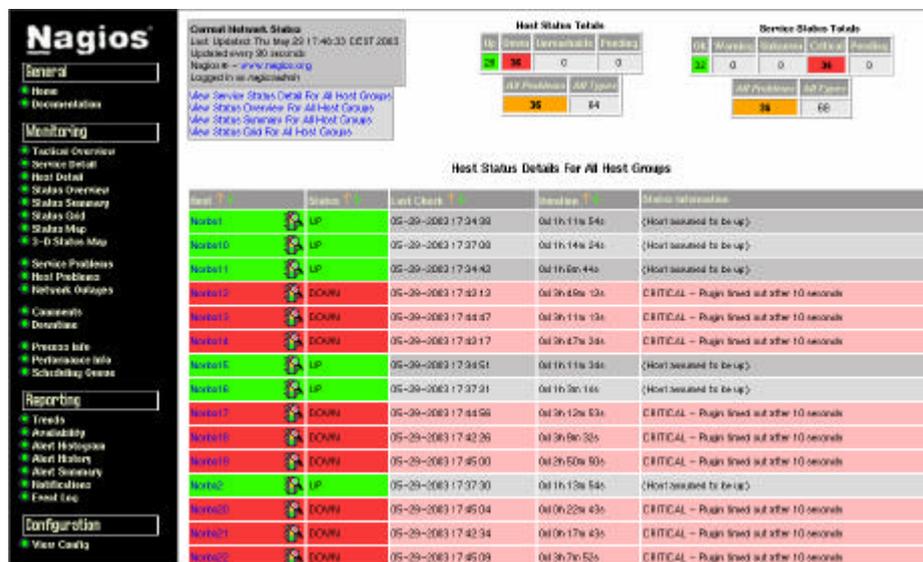
Muestra de forma rápida un resumen de todo el sistema que permita tomar decisiones rápidas apoyadas en una base real del estado del sistema.

### Detalle de los servicios



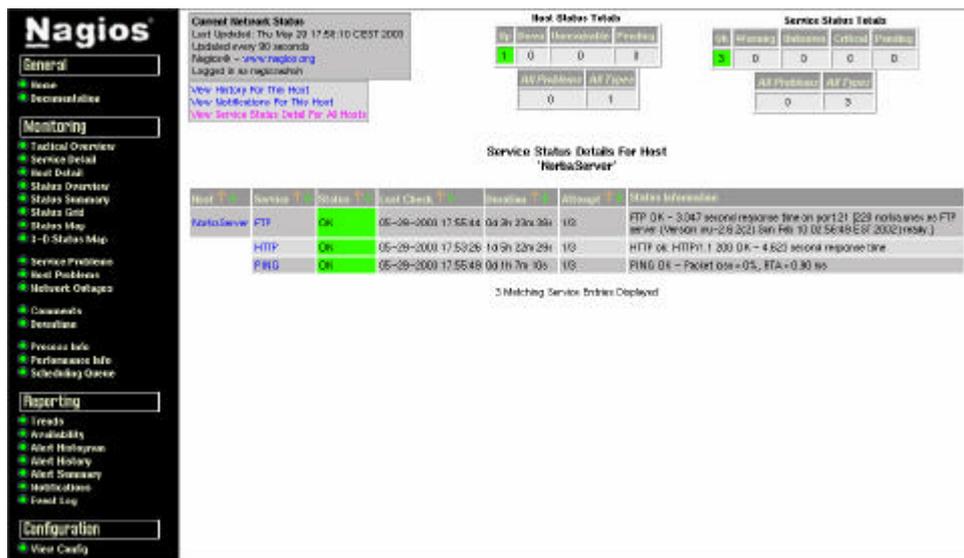
Muestra el estado de los servicios que se están monitorizando así como una descripción textual de si ha habido problemas, si no se tienen datos suficientes, etcétera.

### Detalles de los equipos



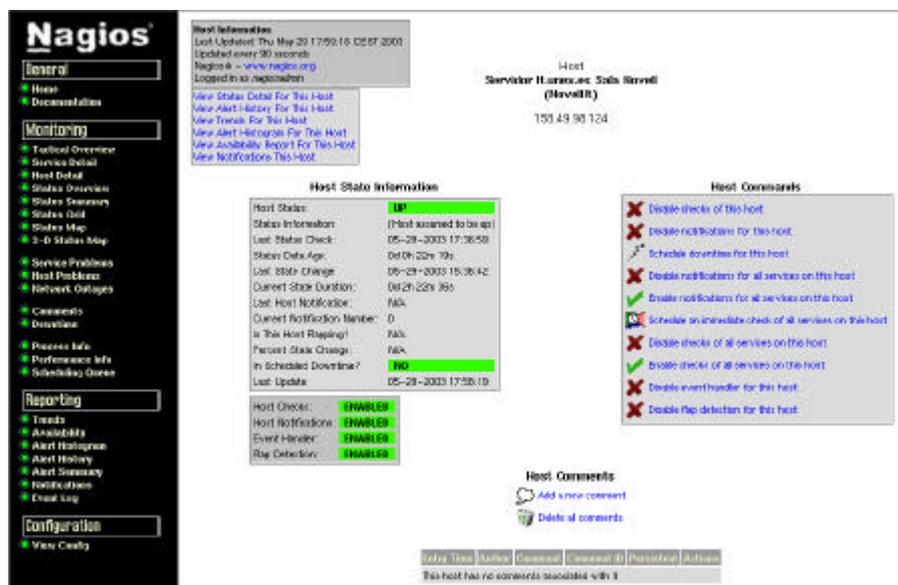
Muestra si los equipos que están siendo monitorizados se encuentran activos, si se encuentran caídos o si el acceso a los mismos está dificultado por alguna cuestión.

### Estado detallado de un equipo



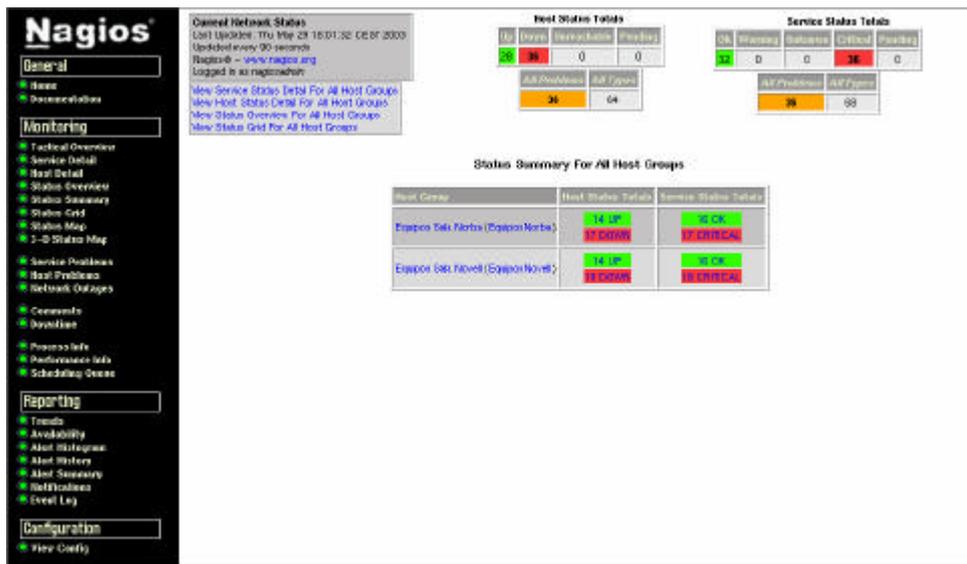
Muestra para cada equipo monitorizado, su estado, el estado de los servicios que tiene asociados y algunos datos extra.

Información sobre un equipo



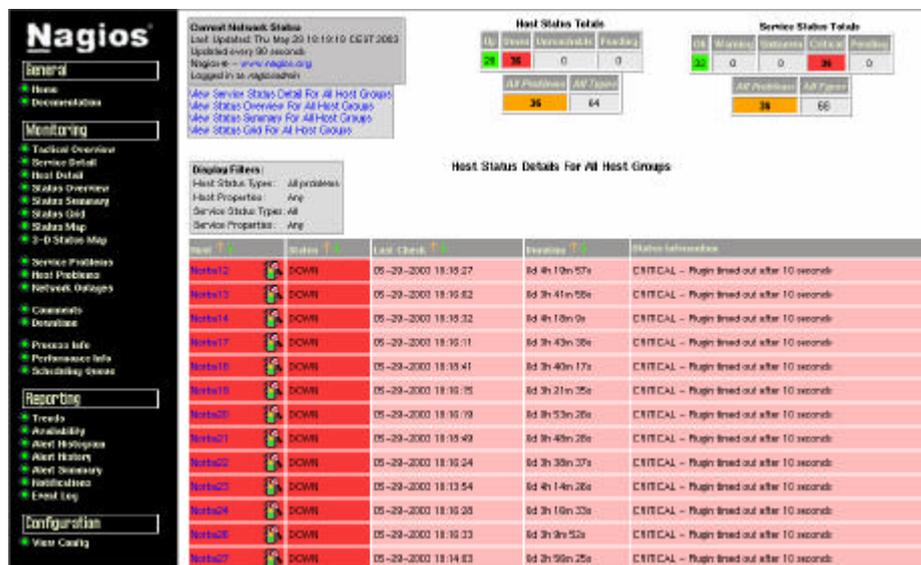
Muestra datos muy detallados sobre un equipo concreto y permite además la ejecución de algunos comandos que afectan a dicho equipo.

Información de estado por grupo de equipos



Muestra un resumen de los equipos y servicios activos y caídos según los grupos a los que pertenece cada grupo y de una forma amena, sencilla y muy rápida.

### Problemas con los equipos



Esta opción muestra exclusivamente los equipos que están teniendo problemas así como una descripción de los mismos. Es especialmente útil para un administrador de red saber inmediatamente qué equipos están fallando.

## Problemas con los servicios

**General Network Status**  
Last Updated: Thu May 29 18:18:02 CEST 2003  
Updates every 30 seconds  
Nagios® - www.nagios.org  
Logged in as nagiosadmin

**Host Status Totals**

Up	Down	Unreachable	Pending
26	0	0	0
26 Problems	0	0	0

**Service Status Totals**

Up	Down	Critical	Warning
0	0	36	0
0 Problems	0	36	0

**Display Filters:**  
Host Status Type: All  
Host Problem: Any  
Service Status Type: All Problems  
Service Problem: Any

**Service Status Details For All Hosts**

Host	Service	Status	Last Check	Duration	Attempt	Status Information
host01	FRG	CRITICAL	05-29-2003 18:13:27	0d 4h 16m 33s	1/0	CRITICAL - Plugin timed out after 30 seconds
host02	FRG	CRITICAL	05-29-2003 18:13:02	0d 3h 40m 41s	1/0	CRITICAL - Plugin timed out after 30 seconds
host03	FRG	CRITICAL	05-29-2003 18:13:32	0d 4h 16m 52s	1/0	CRITICAL - Plugin timed out after 30 seconds
host04	FRG	CRITICAL	05-29-2003 18:13:11	0d 3h 42m 22s	1/0	CRITICAL - Plugin timed out after 30 seconds
host05	FRG	CRITICAL	05-29-2003 18:13:41	0d 3h 39m 1s	1/0	CRITICAL - Plugin timed out after 30 seconds
host06	FRG	CRITICAL	05-29-2003 18:13:15	0d 3h 20m 19s	1/0	CRITICAL - Plugin timed out after 30 seconds
host07	FRG	CRITICAL	05-29-2003 18:13:10	0d 0h 52m 12s	1/0	CRITICAL - Plugin timed out after 30 seconds
host08	FRG	CRITICAL	05-29-2003 18:13:48	0d 0h 47m 12s	1/0	CRITICAL - Plugin timed out after 30 seconds
host09	FRG	CRITICAL	05-29-2003 18:13:24	0d 3h 37m 21s	1/0	CRITICAL - Plugin timed out after 30 seconds
host10	FRG	CRITICAL	05-29-2003 18:13:54	0d 4h 13m 3s	1/0	CRITICAL - Plugin timed out after 30 seconds
host11	FRG	CRITICAL	05-29-2003 18:13:26	0d 3h 15m 17s	1/0	CRITICAL - Plugin timed out after 30 seconds
host12	FRG	CRITICAL	05-29-2003 18:13:33	0d 3h 8m 35s	1/0	CRITICAL - Plugin timed out after 30 seconds

Esta opción muestra exclusivamente los servicios que están teniendo problemas así como una descripción de dichos problemas. Es especialmente útil para un administrador de red saber inmediatamente qué servicios están dejando de funcionar.

## Creación de comentarios para equipos

**Nagios®**

**Commanded Interface**  
Last Updated: Thu May 29 18:20:10 CEST 2003  
Nagios® - www.nagios.org  
Logged in as nagiosadmin

**You are requesting to add a host comment.**

**Commanded Options**

Host Name:

Persistent:

Author (User Name):

Comment:

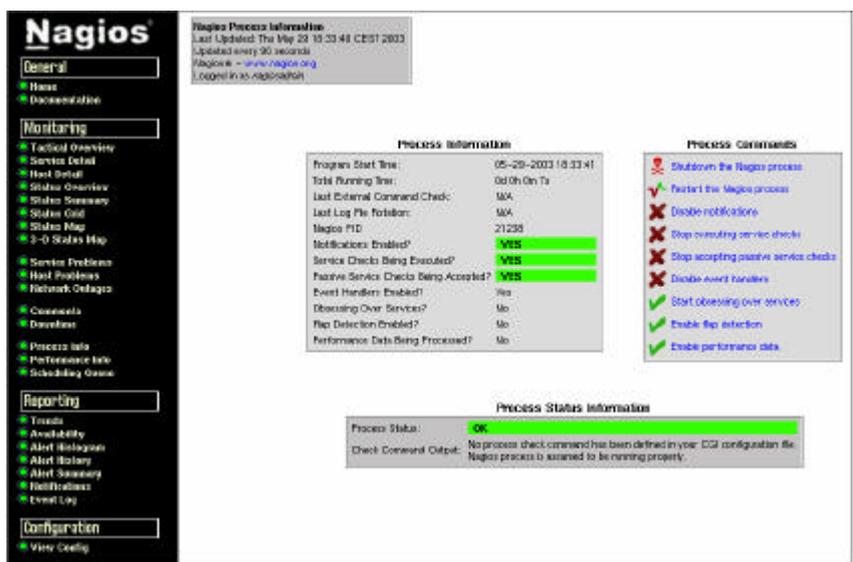
**Commanded Description**

This command is used to add a comment for the specified host. If you work with other administrators, you may find it useful to share information about a host that is having problems if more than one of you may be working on it. If you do not check the 'persistent' option, the comment will be automatically be deleted the next time Nagios is restarted.

Please enter all required information before submitting the command.  
Required fields are marked in red.  
Failure to supply all required values will result in an error.

Permite asociar un comentario a un equipo. Es especialmente útil si varios administradores por turnos administran las máquinas. Uno puede dejar notas sobre ciertos equipos para que otro las vea cuando llegue su jornada laboral.

### Administración web de Nagios



El propio sistema Nagios puede ser administrado vía web mediante la ejecución de comandos. Además se puede ver su estado, las incidencias que ha tenido, etcétera.

### Cola de planificación



Esta opción muestra y permite cambiar la fecha y hora para la cual están planificadas la ejecución de los chequeos a servicios y equipos.

## Configuración de informes

Común para casi cualquier informe. Permite elegir el rango de tiempo, la forma de presentación, el orden, etcétera, de los datos que aparecerán en el informe.

## Informe de disponibilidad

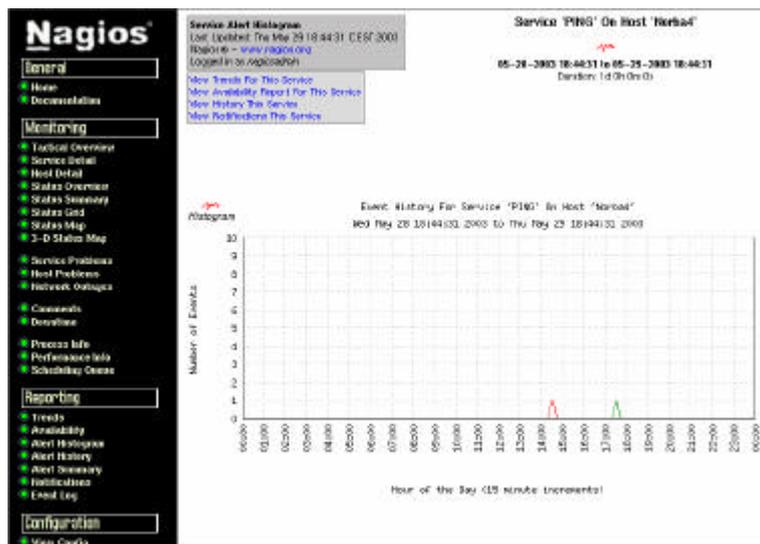
Host	Uptime	Downtime	Time Unreachable	Time Inaccessible
NovaDevs	0.000% (0.000s)	0.000% (0.000s)	0.000% (0.000s)	100.000%
NovaDB	0.000% (0.000s)	0.000% (0.000s)	0.000% (0.000s)	100.000%
NovaW	0.000% (0.000s)	0.000% (0.000s)	0.000% (0.000s)	100.000%
NovaS	0.000% (0.000s)	0.000% (0.000s)	0.000% (0.000s)	100.000%
NovaServer	0.000% (0.000s)	0.000% (0.000s)	0.000% (0.000s)	100.000%

Host	Uptime	Downtime	Time Unreachable	Time Inaccessible
Model	0.000% (0.000s)	0.000% (0.000s)	0.000% (0.000s)	100.000%
Model 0	0.000% (0.000s)	0.000% (0.000s)	0.000% (0.000s)	100.000%
Model 1	0.000% (0.000s)	0.000% (0.000s)	0.000% (0.000s)	100.000%
Model 2	0.000% (0.000s)	0.000% (0.000s)	0.000% (0.000s)	100.000%
Model 3	0.000% (0.000s)	0.000% (0.000s)	0.000% (0.000s)	100.000%
Model 4	0.000% (0.000s)	0.000% (0.000s)	0.000% (0.000s)	100.000%
Model 5	0.000% (0.000s)	0.000% (0.000s)	0.000% (0.000s)	100.000%
Model 6	0.000% (0.000s)	0.000% (0.000s)	0.000% (0.000s)	100.000%
Model 7	0.000% (0.000s)	0.000% (0.000s)	0.000% (0.000s)	100.000%
Model 8	0.000% (0.000s)	0.000% (0.000s)	0.000% (0.000s)	100.000%
Model 9	0.000% (0.000s)	0.000% (0.000s)	0.000% (0.000s)	100.000%

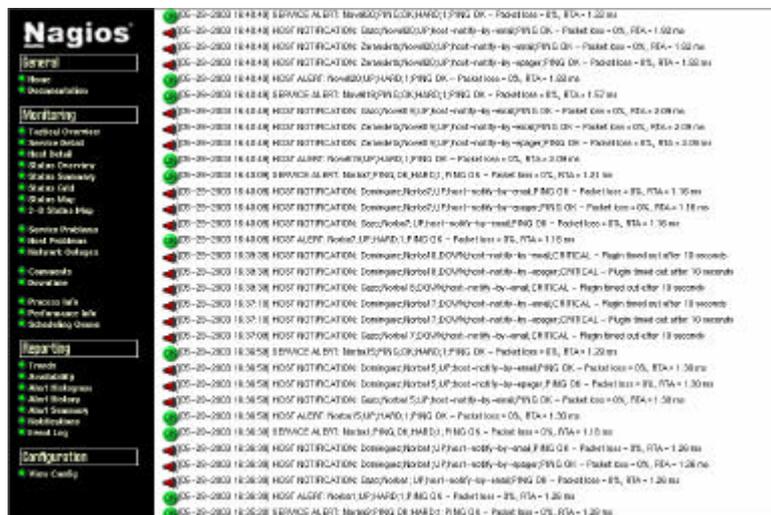
Esta opción presenta en la ventana web un listado con todos los equipos y los porcentajes de tiempo en los que cada uno ha estado activo e inactivo. Esto permite obtener unas estadísticas para ver si una máquina falla con frecuencia y poner medidas al respecto.

### Histograma



Como cualquiera de los demás tipos de informe, el histograma de forma gráfica muestra distintos parámetros, a elegir, sobre los servicios y equipos monitorizados.

### Histórico de eventos



Al más puro estilo UNIX/Linux, esta opción muestra el total de sucesos que han ocurrido en el sistema, desde que un equipo haya caído hasta que a cierto contacto se le ha enviado una notificación vía correo electrónico.

## Contactos

The screenshot shows the Nagios web interface with the 'Contacts' configuration page selected. The page title is 'Contacts' and it includes a 'Refresh' button. The table below lists the configured contacts:

Contact Name	Alias	Email Address	Pager Address	Name	Email Address	Priority	Next Notification Period	Service and Hosts	Next Notification Command
Domingo	Ricardo Domingo	ricardo@redextremadura.net	ricardo@redextremadura.net	Unknown	Warning, Critical, Recovery	24x7	24x7	notify-by-email, notify-by-email	host-notify-by-email, host-notify-by-email
Gato	Alfonso Gato	agato@redextremadura.net		Critical, Recovery	Down, Recovery	workhours	workhours	notify-by-email	host-notify-by-email
Zarandieta	José Zarandieta	zarandieta@redextremadura.net	zarandieta@redextremadura.net	Unknown	Warning, Critical, Recovery	24x7	24x7	notify-by-email, notify-by-email	host-notify-by-email, host-notify-by-email

Esta opción permite ver los datos de configuración de los contactos, esto es, horas de contacto, métodos para notificaciones, dirección de correo, datos personales, etcétera. Lo mismo aparece para otros parámetros que no sean los contactos, pero sólo mostramos esta opción como ejemplo.

## Conclusiones

Como hemos podido ver en este estudio, Nagios es un sistema de monitorización muy completo, con grandes posibilidades de ampliación que realiza perfectamente su labor. Además es un software gratuito y libre, lo que lo convierte en un candidato ideal para cualquier organización que desee implantar un sistema de gestión de red. Así lo demuestra el amplio número de organizaciones entre empresas y universidades y organismos gubernamentales que lo usan. Sin embargo es un sistema complejo y tedioso de configurar e instalar correctamente, por lo que quizás no merezca la pena el esfuerzo de configuración e implantación (y formación) para usarlo con redes pequeñas.

Además de lo anterior, se echa de menos que el sistema trabaje de forma nativa con protocolos de gestión de red estándares como SNMP en lugar de realizar la gestión y monitorización de la red con herramientas propias, *plugins*, etcétera que podrían tener problemas para traspasar cortafuegos, ser dependientes de la plataforma, etcétera.

En cualquier caso, Nagios es una herramienta que cualquier administrador de servicios de red y/o de sistemas debería conocer pues supone una alternativa a productos comerciales caros, es un producto probado, escalable y puede suponer un ahorro importante en licencias software para las organizaciones que lo usen (y por tanto un punto a favor del administrador).

Nagios es un sistema que podríamos haber estudiado y explotado mucho más, pero su puesta en marcha ha implicado los siguientes aspectos que nos han consumido tiempo:

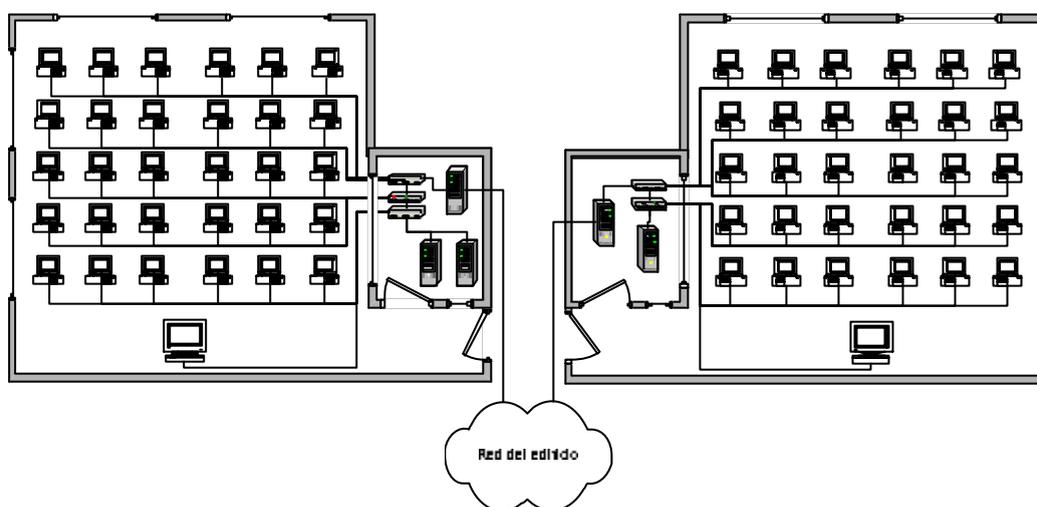
- Conocimientos a nivel usuario y administrador de GNU/Linux.
- Instalación y configuración avanzada de Apache Web Server.
- Conocimientos avanzados de inglés técnico.

Y lo hemos tenido que realizar en 4 máquinas distintas. No siendo comparable, en este sentido, a otros programas cuya instalación se reduce a un simple clic de ratón.

## Caso práctico

### Descripción

Para usar un ejemplo que resulte familiar y creíble, vamos a monitorizar equipos y servicios de varias salas de la Escuela Politécnica de Cáceres, en concreto los equipos de la sala Norba y de la sala Novell. De este modo tendremos que crear dependencias de servicios, de máquinas



En la figura anterior se observa la disposición de las dos salas, su organización, su modo de conexión, los equipos importantes, el modo de conexión a la red del edificio, etcétera.

### Configuración de los CGI

En este fichero hemos configurado que el acceso al sistema Nagios y a los CGI se autentica y hemos dado todos los privilegios al usuario `nagiosadmin` con clave `peder`, para que solamente este usuario pueda realizar tareas de administrador. Nos debatimos entre esto y crear más usuarios con diferentes niveles de acceso pero para mostrar el funcionamiento del sistema no lo encontramos necesario. Hemos añadido las siguientes líneas al fichero `cgi.cfg`.

```
use_authentication=1
authorized_for_system_information=nagiosadmin
authorized_for_system_commands=nagiosadmin
authorized_for_configuration_information=nagiosadmin
authorized_for_all_hosts=nagiosadmin
authorized_for_all_host_commands=nagiosadmin
authorized_for_all_services=nagiosadmin
authorized_for_all_service_commands=nagiosadmin
```

## Configuración de los contactos

Hemos modificado totalmente el fichero `contacts.cfg` para definir tres administradores de las dos salas en estudio. Los administradores definidos son los siguientes.

Nombre	Horario
Domínguez	24 horas 7 días a la semana
Zarandieta	24 horas 7 días a la semana
Gazo	Horario laboral

Existen más datos referentes a cada uno de ellos pero se ha optado por mantener una configuración estándar (notificaciones por email, pager...).

## Configuración de los grupos de contactos

Hemos definido en el fichero `contactgroups.cfg` dos grupos, uno para cada sala administrada: `ContactosNovell` y `ContactosNorba`. El primero de ellos está formado por los administradores de la sala Novel, que en este caso hemos decidido que sean `Zarandieta` y `Gazo`. Igualmente para la segunda, hemos pensado que lo formen `Gazo` y `Domínguez`.

## Configuración de equipos

La configuración de los equipos la hemos definido en el fichero `hosts.cfg` basándonos en la plantilla genérica que venía en los ficheros de configuración de ejemplo en Nagios. No obstante, es posible distinguir aspectos como habilitación de notificaciones, eventos, registro de los mismos, etcétera, utilizando diferentes plantillas y asignándolas, posteriormente, a diferentes definiciones de equipos.

Cada uno de los 65 equipos definidos puede llevar su especificación particular, esto es, intervalos de notificaciones, periodo de notificación, etcétera. Los campos específicos de los equipos que estamos tratando son el nombre, el alias para reconocerlo en la interfaz web y su dirección IP.

Hemos definido los siguientes equipos:

`NovellP`: equipo del profesor de la sala Novell.

`NovellServer`: `carvajal.unex.es`

`NovellIT`: `it.unex.es`

`Novell1-Novell30`: equipos de usuario de la sala Novell.

`NorbaServer`: `norba.unex.es`

`Norba1-Norba30`: equipos de usuario de la sala Norba

## Configuración de grupos de equipos

Hemos realizado cambios en el fichero `hostgroups.cfg` para crear tres grupos de equipos:

`EquiposNovell`: Incluye todos los equipos que se encuentran físicamente en la sala Novell de la Politécnica. Las notificaciones se enviarán, lógicamente, al grupo `ContactosNovell`.

`EquiposNorba`: Incluye todos los equipos que se encuentran físicamente en la sala Norba de la Politécnica. Las notificaciones se enviarán, lógicamente, al grupo `ContactosNorba`.

**EquiposCríticos:** se trata de los equipos cuyo funcionamiento se considera crítico para el sistema, compuesto por servidores web, ftp, dhcp... Están los servidores de ambas salas y los equipos de los profesores. Las Notificaciones se envían tanto a **ContactosNorba**, como a **ContactosNovell**.

## Configuración de los servicios

Hemos definido los servicios en el fichero `services.cfg`. En concreto hemos configurado el sistema para que gestione los siguientes servicios (cada configuración de cada equipo se realiza individualmente):

**Todos los equipos:** comprobación del Round Trip Time mediante un PING y comprobación de que están activos mediante un `check_alive`.

**NorbaServer y NovellIT:** comprobación de los servicios HTTP y FTP, además de los descritos en el punto anterior.

## Configuración de las dependencias

En el fichero `dependencias.cfg` hemos definido las dependencias entre equipos, lo que llamamos *host dependency*. Teniendo en cuenta la distribución de la sala y suponiendo que el sistema de monitorización está situado fuera de estas salas, ningún equipo podría ser monitorizado si fallara su respectivo router. Así que hemos decidido crear dos dependencias, una por sala, en la que todos los equipos de la sala dependen del router: **NorbaServer** o **NovellServer**, según el caso.

## Configuración avanzada

Hemos activado la posibilidad de que mediante la interfaz web, el administrador pueda modificar parámetros de ejecución del sistema Nagios. Para ello hemos modificado el fichero principal `nagios.cfg`, como se explica en el apartado de configuración avanzada de este manual.

Existen otra serie de ficheros de configuración que no hemos citado aquí bien porque los hemos reutilizado tal y como ocurre con `timeperiods.cfg` y `checkcommands.cfg`; o bien porque su utilización es opcional en cuyo caso hemos eliminado su contenido (`misccommands.cfg` o `escalations.cfg`)

## Autores



**Manuel Domínguez Dorado** es (en 2.003) estudiante de 5º curso de Ingeniería Informática en la Escuela Politécnica de Cáceres, Universidad de Extremadura (ESPAÑA). Sus áreas de interés son las redes LAN, MAN o WAN, las redes corporativas, protocolos de comunicaciones, la calidad de servicio en transmisiones y los sistemas distribuidos, ingeniería web y sistemas intranet/extranet/Internet. Ha escrito diversos artículos para revistas del sector informático y actualmente desarrolla su proyecto final de carrera en relación con la tecnología MPLS.



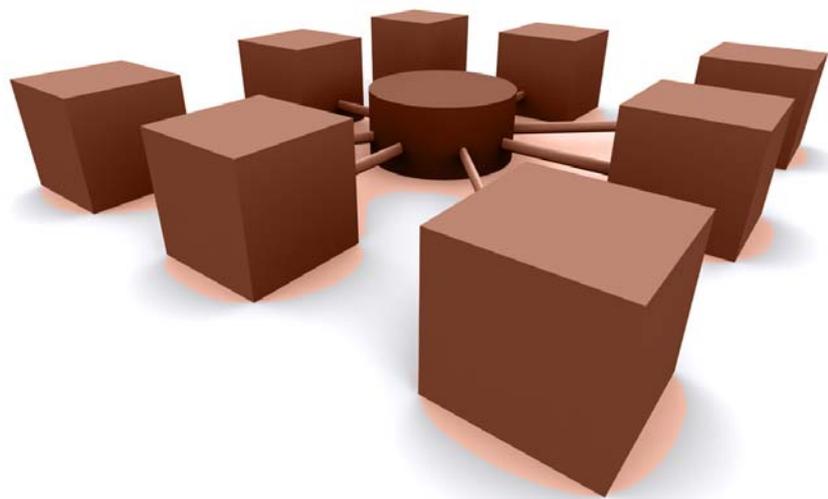
**Jose Antonio Zarandieta** es Diplomado en Informática por la UNEX, Cisco Certified Network Associate (CCNA) e Instructor de CCNA (CCAI) y actualmente (Mayo 2003) estudia 5º Curso de Ing. Informática en la Universidad de Extremadura (España). Ha escrito diferentes artículos y libros en el sector de la informática. Actualmente realiza como proyecto final de carrera un Simulador de Redes de Área Local.

## Bibliografía y fuentes consultadas

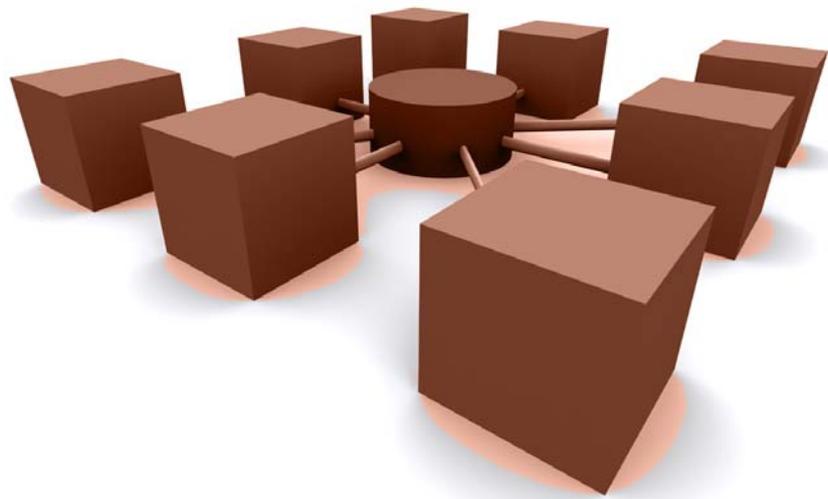
Para la realización de este estudio hemos tenido que consultar las siguientes fuentes y bibliografía:

- ***“Nagios versión 1.0 documentation”***.  
Ethan Galstad, 2.002
- ***“Procedure for the installation of the Nagios network monitoring program”***.  
Ahk, 2.002
- ***“Manual del servidor HTTP Apache”***.  
Apache Foundation, 2.003
- ***“Documentación de los plugins de Nagio”***.





ISBN: PENDIENTE



ISBN: 978-84-615-1184-6