



**Nagios®**

# Sistema de monitorización de servicios en Linux

Memoria del trabajo de  
Servicios en red

realizada por  
**Sergio Arranz Sobrino**

Centro de formación profesional  
González Cañadas

21 de Febrero de 2015

## **Resumen**

A día de hoy es de gran interés por parte del sector tecnológico el uso de herramientas informáticas que permitan manejar y controlar los equipos, de modo que se puedan solucionar todas las incidencias posibles y como consecuencia ahorrar en tiempo y costes.

No solamente esto, es muy importante porque su buen uso además puede mejorar la calidad del servicio que se ofrece tanto a usuarios y clientes como a los técnicos responsables de los sistemas.

Este sistema permite ver desde una Interfaz Web el estado de los servicios y hosts asociados, log de incidencias y eventos que se producen y entre otras muchas cosas el manejo de herramientas que evitan a los administradores en muchos casos trabajar incómodamente.

Otro aspecto a tener en cuenta y muy favorecedor es el tema de las notificaciones, podemos elegir desde las más clásicas como el correo electrónico hasta algunas de las más novedosas como aplicaciones para todo tipo de smartphones que permiten monitorizar y notificar de modo muy completo y profesional.

## **1. Introducción**

### **1.1 Presentación**

### **1.2 Objetivos**

### **1.3 Estructura de la memoria**

### **1.4 Motivaciones personales**

## **2. Instalación y configuración**

### **2.1 Requisitos previos**

### **2.2 Instalación**

#### 2.2.1 Instalación de Nagios

#### 2.2.2 Configuración de Interfaz Web

### **2.3 Configuración de equipos**

#### 2.3.1 Equipos Windows

#### 2.3.2 Equipos Linux

### **2.4 Sistema de notificaciones Nagios**

#### 2.4.1 Notificaciones vía Email

#### 2.4.2 Notificaciones vía SMS

#### 2.4.3 Notificaciones vía Android

## **3. Conclusiones**

## **4. Bibliografía**

## **5. Apéndice**

## **6. Glosario**

## 1. Introducción

En esta parte inicial del trabajo se describe la presentación, objetivos, estructura de la memoria y un pequeño apartado extra en el que hablaré personalmente sobre mis motivaciones en la elección de este trabajo.

### 1.1 Presentación

*Nagios* es una herramienta *Open Source* (de código libre) orientada a la supervisión y monitorización de equipos y servicios en red cuyo fin es básicamente avisar de cualquier comportamiento imprevisto que se produzca en estos.

Sus características principales son la gestión de servicios en red (*SMTP, POP3, HTTP, SSH, SQL, etc*), gestión de recursos hardware de los sistemas (carga de procesador, memoria, uso de discos, etc), posibilidad de monitorización remota por *túneles SSL cifrados o SSH* y posibilidad de programar scripts nuevos.

### 1.2 Objetivos

Objetivo principal: llevar a cabo la implementación de un servidor *Nagios* en una máquina virtual *Debian* (Linux).

Destacan los siguientes aspectos específicos:

- Instalación y configuración de plugins
- Configuración de Interfaz Web
- Configuración y monitorización de hosts Windows y Linux
- Configuración, modificación y uso de archivos *Nagios*
- Instalación y configuración de notificaciones

## 1.3 Estructura de la memoria

- Introducción: se explica la teoría y todo lo básico para entender el funcionamiento y desarrollo del trabajo.
- Implementación: se detallan las fases de instalación y configuración de todo lo referente a *Nagios*.
- Pruebas: se comprueba el correcto funcionamiento del servicio y sus componentes a lo largo de todas las fases.
- Conclusiones: se describen las conclusiones finales y las desviaciones y problemas surgidos a lo largo del trabajo.
- Bibliografía: se citan las fuentes consultadas y sus autores.
- Apéndice: se expone un índice de imágenes, códigos implementados en los ficheros de configuración y un glosario con la aclaración de términos y abreviaturas.

## 1.4 Motivaciones personales

La elección del trabajo fue en base a la mención del mismo como propuesta (entre otras muchas) por parte del profesor de la asignatura, finalmente me acabó convenciendo un documento sobre una guía de instalación básica de lo mismo que era para una plataforma similar facilitada por un compañero.

Mi motivación principal ha sido los diversos usos que se le puede dar a esta herramienta y a otras similares y conocer cómo funciona me ha ayudado en gran parte a comprender como operan los técnicos y desarrolladores ante la resolución de cualquier imprevisto que pueda surgir.

Además, tras ver tanta documentación y proyectos tan interesantes, tengo pensado seguir investigando en el futuro y probando sistemas de monitorización para otras plataformas.

El próximo que miraré a fondo será *NagiosPi* para Raspberry Pi.

## 2. Instalación y configuración

Antes de llevar a cabo este apartado, se definirán brevemente los requisitos previos en los cuales se indicará las plataformas sobre las que se va a trabajar y su explicación.

### 2.1 Requisitos previos



[Virtualbox v4.3.36](#): programa con una versión estable que usaremos para llevar a cabo todas las tareas en un entorno virtualizado.

Sistema Anfitrión: Windows 8.1, probado también en Windows 7

Servidor: [Debian 7.8 LXDE GUI](#) IP: 192.168.1.20/24

Cliente Linux: [Debian 7.8 LXDE GUI](#) IP: 192.168.1.50/24

Cliente Windows: [W7 Ultimate x64](#) IP: 192.168.1.51/24

Es importante que las direcciones IP sean fijas y estén todas en el mismo rango de red. Además, hay que cambiar el adaptador de red de las tres máquinas en Virtualbox a Modo puente:

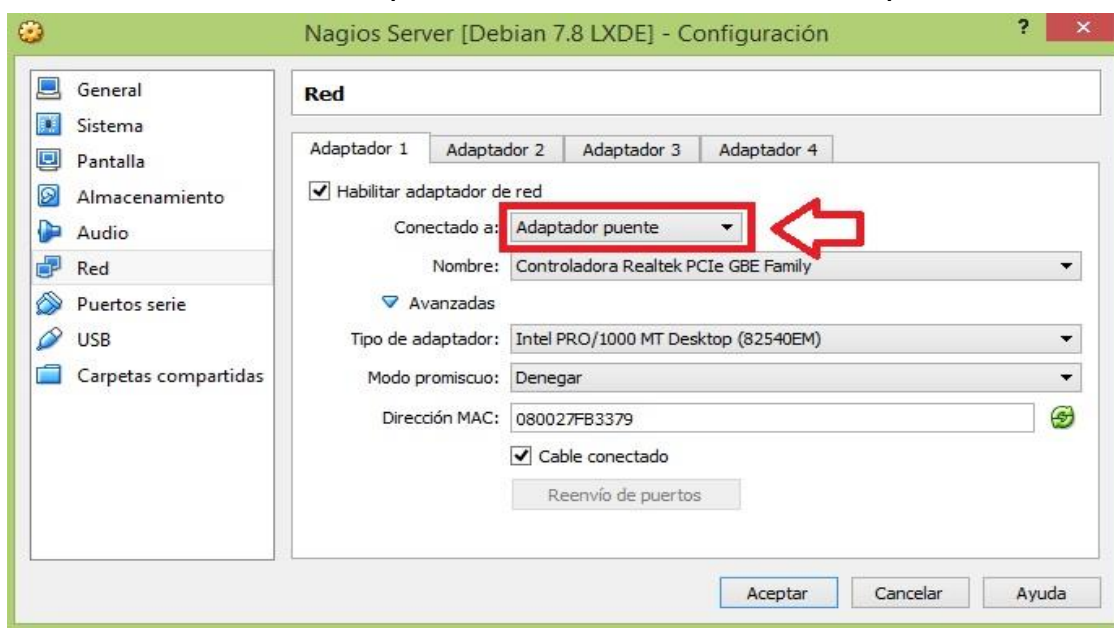


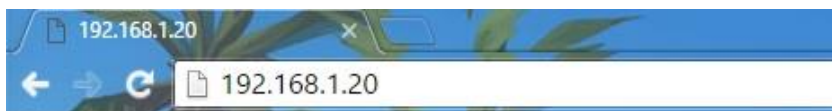
Ilustración 1. Configuración de Adaptadores de red en VirtualBox.

## 2.2 Instalación

Comenzamos instalando los paquetes *Apache 2*, *PHP*, *GD development libraries* y *GC compiler and development libraries*.

```
# apt-get install apache2
```

Vemos si funciona accediendo por IP: "<http://192.168.1.20>"



**It works!**

This is the default web page for this server.

The web server software is running but no content has been added, yet.

Ilustración 2. Ejemplo de instalación correcta de Apache Web server

```
# apt-get install php5
```

\*Instalar también "libapache2-mod-php5" en caso de no tenerlo (suele venir por defecto al instalar PHP 5).

```
# apt-get install make gcc g++ sudo libgd2-xpm libgd2-xpm-dev  
libpng12-dev libgd-tools libpng3-dev
```

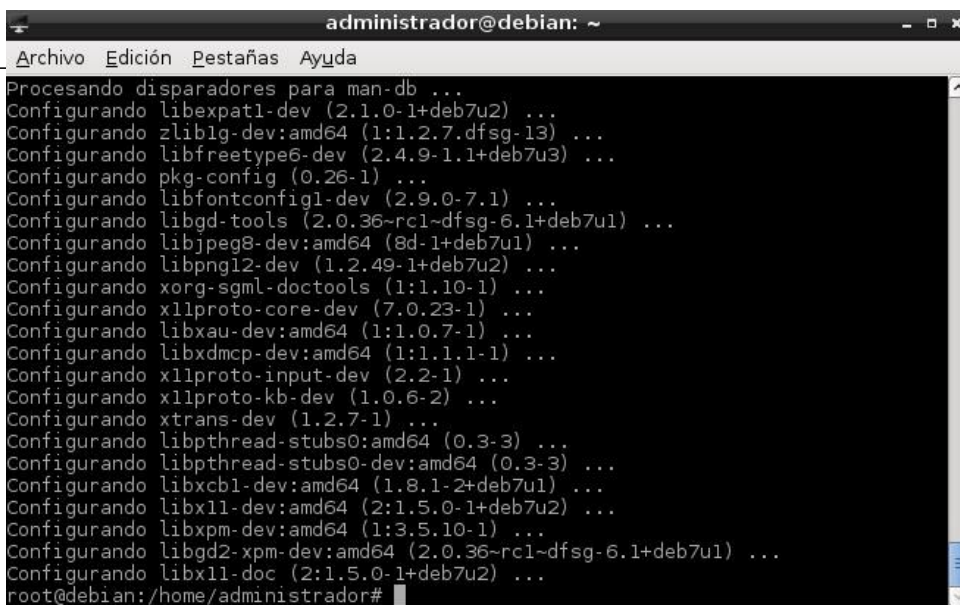


Ilustración 3. Ejemplo de correcta instalación de librerías

## 2.2.1 Instalación de Nagios

Por motivos de seguridad es mejor crear un usuario cada servicio, de modo que crearemos uno para *Nagios*:

```
# adduser nagios
```

Creamos el grupo y añadimos a los users nagios y www-data:

```
# groupadd nagios
```

```
# usermod -G nagios nagios
```

```
# usermod -G www-data,nagios www-data
```

Seleccionamos la última versión de **Nagios Core** Free (4.1.1 actualmente) desde el [enlace oficial](#) copiando el que dice "Latest stable release":

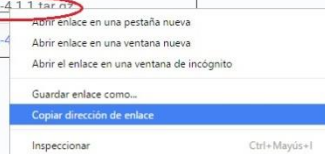
### Nagios Core

#### Latest Version 4 Releases

Version	Date	Notes	Type	Link
4.1.1	2015-08-19	Latest stable release	Source code	<a href="#">nagios-4.1.1.tar.gz</a>
4.0.8	2014-08-12	Previous stable release	Source code	<a href="#">nagios-4.0.8.tar.gz</a>

#### Additional Resources

- [Installation Documentation](#)
- [Nagios Core Documentation](#)
- [Changelog \(version history\)](#)
- [Git repository](#)



Volvemos al servidor y hacemos la descarga, descomprimos e instalamos *Nagios Core* con los siguientes parámetros:

```
# wget https://assets.nagios.com/downloads/nagioscore/releases/nagios-4.1.1.tar.gz
```

```
# tar xvzf nagios-4.1.1.tar.gz
# cd nagios-4.0.4/
# ./configure --prefix=/usr/local/nagios --with-cgiurl=/nagios/cgi-bin
--with-htmurl=/nagios/ --with-nagios-user=nagios --with-nagios-
group=nagios --with-command-group=nagios
# make all
# make install
# make install-init
# make install-commandmode
# make install-config
# make install-webconf
```



A continuación instalamos *Nagios Plugins*, un añadido que permite conectar con equipos remotos Windows para gestionar recursos hardware además de otros servicios o dispositivos.

Seleccionamos la última versión de Nagios Plugins desde el [enlace oficial](#) y seleccionamos la "Latest Stable release":

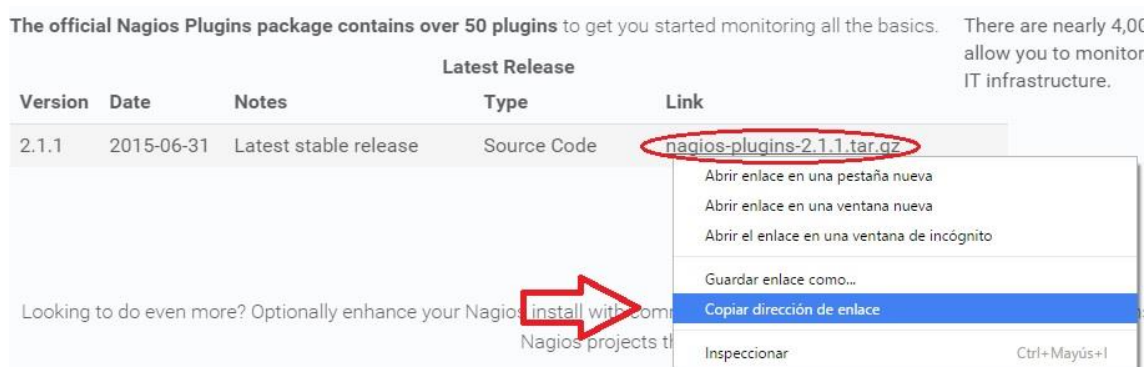


Ilustración 5. Descarga de *Nagios Plugins* 2.1.1 desde web oficial

Descargamos, descomprimos e instalamos en el servidor:

```
# wget http://www.nagios-plugins.org/download/nagios-plugins-2.1.1.tar.gz
# tar xvfz nagios-plugins-2.1.1.tar.gz
# cd nagios-plugins-2.1.1/
# ./configure
# make && make install
```

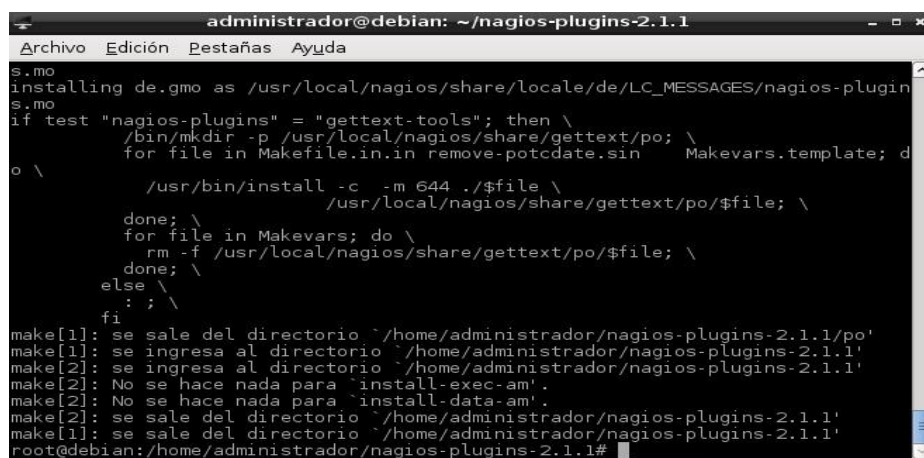


Ilustración 6. Ejemplo de correcta instalación de *Nagios Plugins*

Una vez instalado y compilado correctamente, comprobamos que todo está correcto listando el directorio "libexec":

```
# ls /usr/local/nagios/libexec/
```

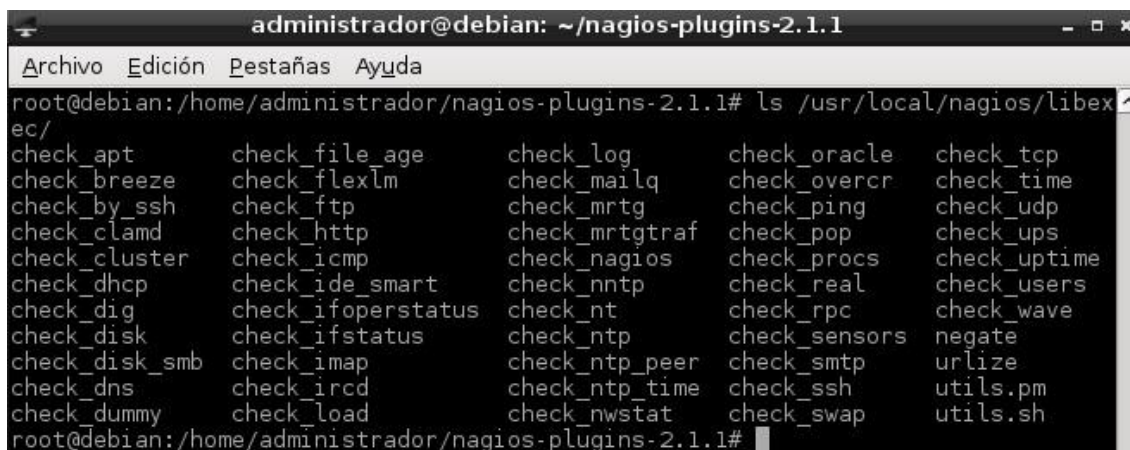


Ilustración 7. Ejemplo de listado de plugins de Nagios correcto

Permitimos que Nagios se inicie con el sistema activando el demonio con las opciones y runlevels que tiene nuestra máquina por defecto, además de crear un enlace simbólico.

```
# update-rc.d -f nagios defaults 99
# ln -s /etc/init.d/nagios /etc/rcS.d/S99nagios
```

## 2.2.2 Configuración de Interfaz Web

Creamos el fichero del sitio para Nagios en Apache:

```
# nano /etc/apache2/sites-available/nagios
```

Añadir este texto (usa las rutas de instalación de esta guía):

```
<VirtualHost *:80>
DocumentRoot /usr/local/nagios/share
ScriptAlias /nagios/cgi-bin /usr/local/nagios/sbin
<Directory "/usr/local/nagios/sbin">
Options ExecCGI
AllowOverride None
Order allow,deny
Allow from all
AuthName "Access"
AuthType Basic
AuthUserFile /usr/local/nagios/etc/htpasswd.users
Require valid-user
</Directory>
Alias /nagios /usr/local/nagios/share
<Directory "/usr/local/nagios/share">
```

```
Options None
AllowOverride None
Order allow,deny
Allow from all
AuthName "Access"
AuthType Basic
AuthUserFile /usr/local/nagios/etc/htpasswd.users
Require valid-user
</Directory>
</VirtualHost>
```

Activamos y recargamos el sitio con los siguientes comandos:

```
# a2ensite nagios
# service apache2 reload
```

\*Recomendable eliminar los sitios por defecto y desactivar el sitio "000-default":

```
# a2dissite 000-default
# rm d* /etc/apache2/sites-available
```

Creamos el archivo para guardar las contraseñas de los usuarios que tengan permiso para entrar a la Interfaz Web. Añadimos un usuario también.

```
# htpasswd -c /usr/local/nagios/etc/htpasswd.users nagiosadmin
```

\*Recomendable revisar si en /usr/local/nagios/etc/cgi.cfg aparece activada la opción "use\_authentication=1", viene por defecto.

Tras realizar estos ajustes reiniciamos Nagios y Apache2:

```
# service apache2 restart
# service nagios restart
```

Abrimos un navegador y escribimos <http://192.168.1.20>:

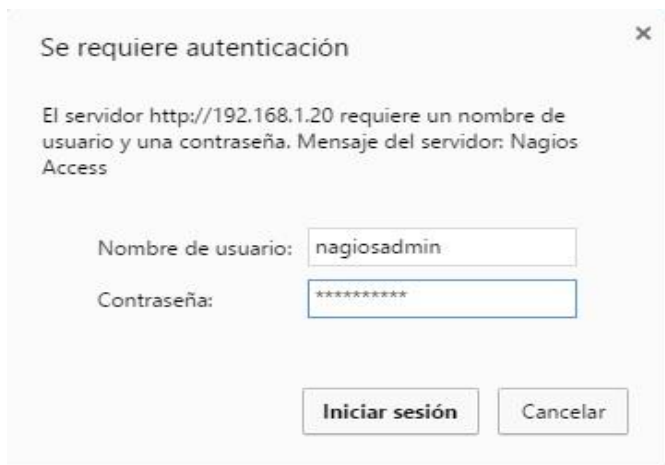


Ilustración 8. Ejemplo de autenticación en Interfaz Web Nagios

Una vez logueados correctamente veremos esta ventana:

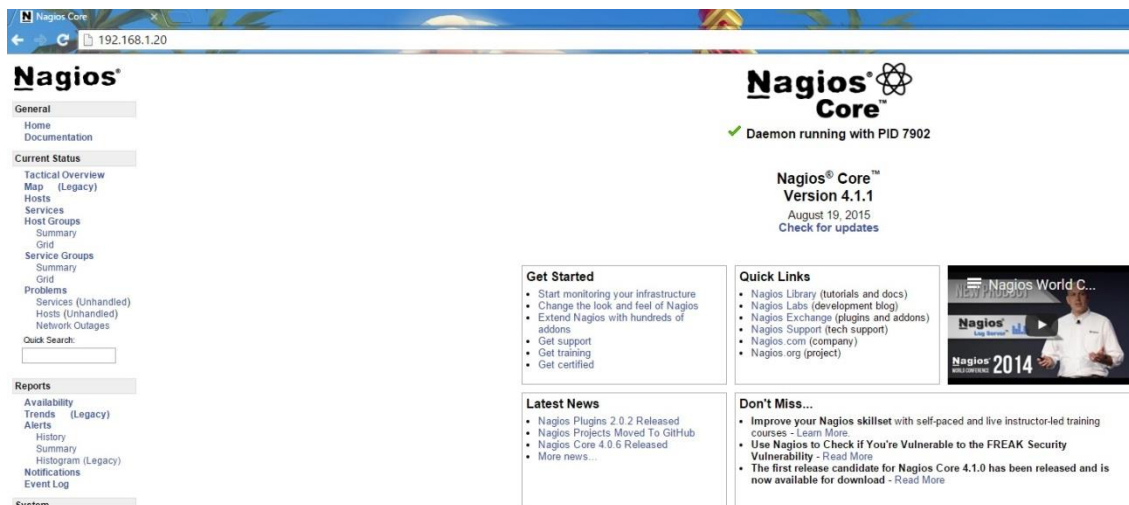


Ilustración 9. Ejemplo de index correcta de Interfaz Web Nagios

## 2.3 Configuración de equipos

En este punto se mostrará un ejemplo de monitorización remota de un equipo Windows y otro Linux con sus configuraciones.

### 2.3.1 Equipos Windows

Descargamos la última versión de NSClient++ del [enlace oficial](#) para nuestro Windows y comenzamos con la instalación seleccionando "Typical" como modo de instalación.

En la siguiente ventana debemos poner en Allowed Hosts la IP de nuestro servidor Nagios (192.168.1.20) y la contraseña que queremos (nos da una aleatoria por defecto).

Además dejaremos marcadas las tres primeras casillas para habilitar el uso de dichos plugins que utilizaremos más adelante.

En el modo que utilizará el servidor NRPE dejaremos por defecto marcado el radio botón "Safe Mode".

Una vez realizados todos estos cambios procedemos a hacer la instalación y finalizarla correctamente.

\*Abajo se cita una ilustración de la pestaña de configuración

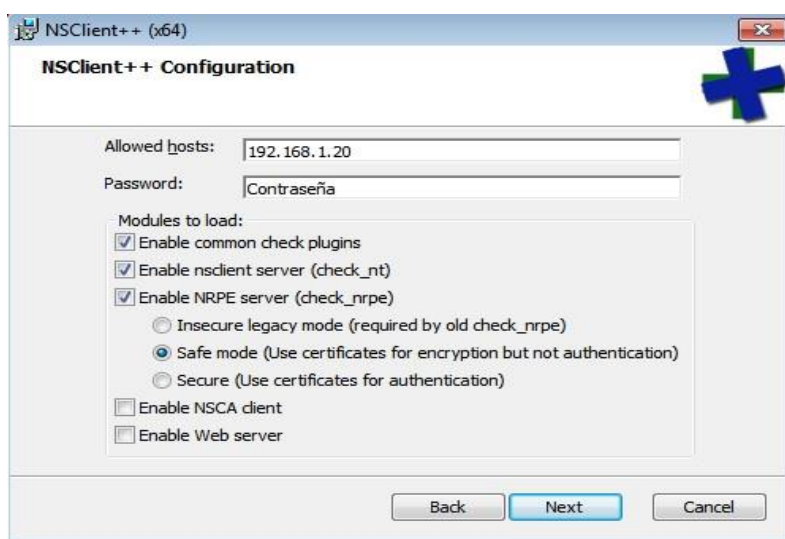


Ilustración 10. Ejemplo válido de configuración de NSClient++

Volvemos al servidor y a partir de ahora usaremos el usuario de Nagios, creamos una copia de seguridad de "windows.cfg":

```
# su nagios
# cd /usr/local/nagios/etc/objects/
# cp windows.cfg windows_org.cfg
```

Accedemos al archivo "windows.cfg" con nano y definimos el nuestro host y hostgroup del siguiente modo:

```
# /usr/local/nagios/etc/objects/windows.cfg

define host{
use windows-server ;    Plantilla de la que hereda la config.
host_name Windows 50 ;  Nombre corto para identificar al host.
alias Equipo 50 ;        Nombre/Desc para dicho host.
address 192.168.1.50 ;   Dirección IP del equipo a monitorizar
hostgroups equipos ;     Nombre de hostgroup del cual pertenece.
}

define hostgroup{
hostgroup_name equipos ;
alias Equipos de Windows ;
}
```

Definimos todos los servicios que vienen por defecto editando solo la directiva "host\_name" como en este ejemplo:

```
define service{
use generic-service ; Nombre de la plantilla de la que hereda config.
host_name Windows 50 ; Nombre de nuestro host Windows definido
service_description Memory Usage ; Descripción del servicio
check_command check_nt!MEMUSE!-w 80 -c 90 ; Comando con parámetros
}
```

Para monitorizar los recursos HW nos interesa el comando check\_nt, por ello lo editaremos en el fichero commands.cfg dejándolo con nuestra contraseña del siguiente modo:

```
# /usr/local/nagios/etc/objects/commands.cfg

# 'check_nt' command definition
define command{
command_name check_nt
command_line $USER1$/check_nt -H $HOSTADDRESS$ -p 12489 -s
Contraseña -v $ARG1$ $ARG2$
}
```

Habilitamos el uso de windows.cfg en el fichero de config. principal de Nagios descomentando la siguiente linea:

```
# /usr/local/nagios/etc/nagios.cfg

#cfg_file=/usr/local/nagios/etc/objects/windows.cfg
```

Verificamos que la config. es correcta con este comando:

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

Si no ha dado ningún error procedemos a reiniciar el servicio:

```
# sudo service nagios restart
```

Hemos acabado con la configuración, pasamos a enseñar los resultados en la interfaz gráfica y un par de utilidades.

Podemos reiniciar el servicio Nagios desde Interfaz gráfica haciendo clic en la barra lateral sobre "Process Info", pinchando sobre la opción "Restart the Nagios process" y confirmando con el botón "Commit", tal y como se muestra en la imagen:

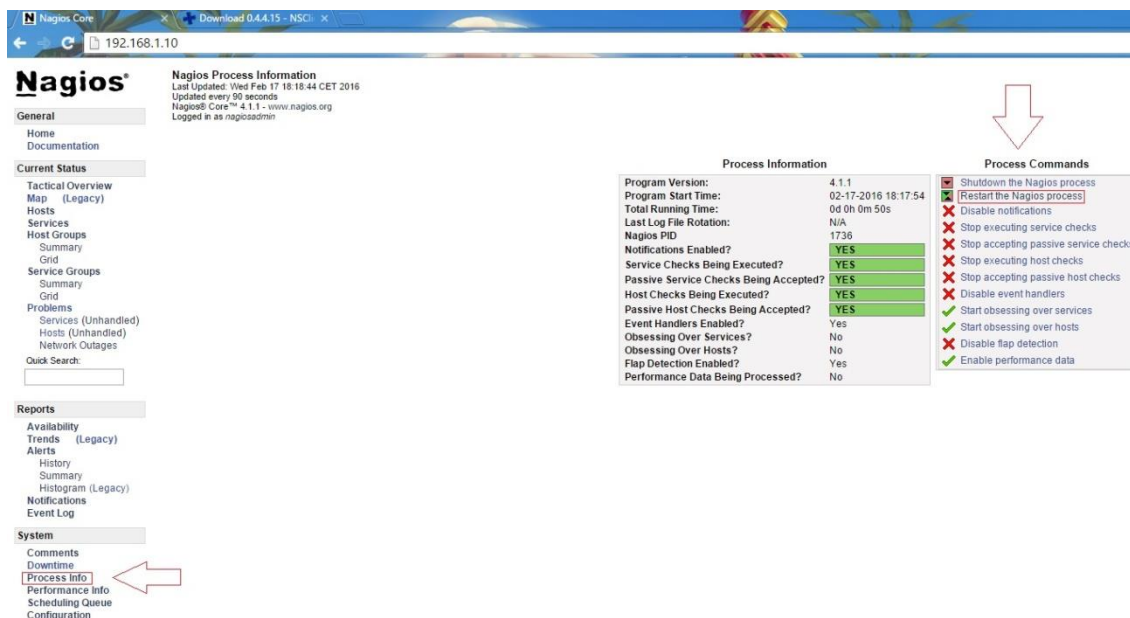


Ilustración 11. Pasos para reiniciar servicio Nagios desde Interfaz Web

A continuación visualizamos el estado de nuestro Host Windows haciendo clic en "Grid", debería aparecernos todas las casillas de nuestro host Windows en gris:

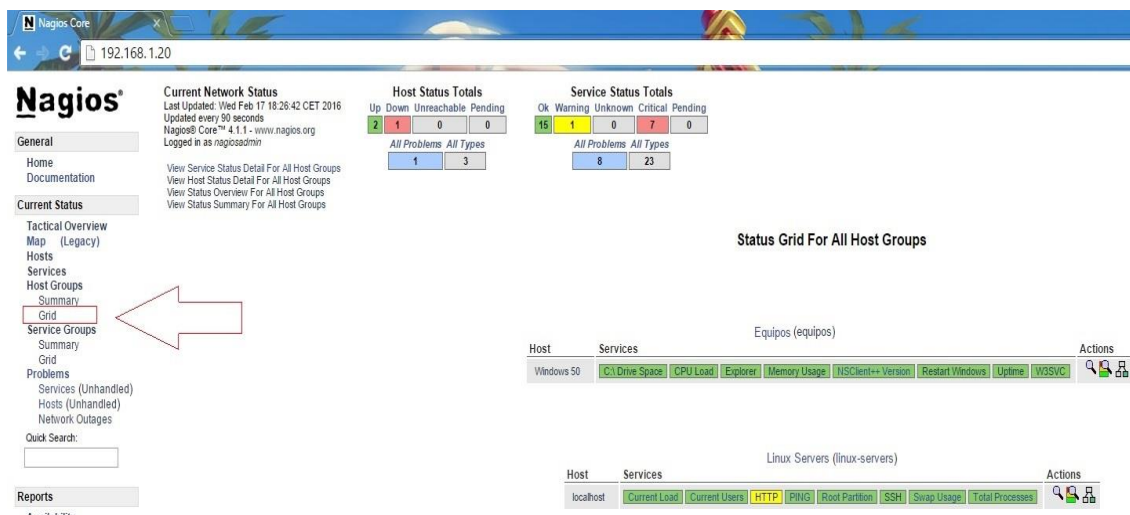


Ilustración 12. Visualización de estado de nuestros hosts



Es normal que aparezcan todos en gris puesto que no se han comprobado todavía, para forzar todos hacemos clic en nuestro host "Windows 50" y a la derecha hacemos clic sobre la opción "Schedule a check of all services on this host".

A continuación seleccionamos la hora futura en la que queremos forzarlo teniendo en cuenta la fecha y hora del servidor que aparece arriba a la izquierda.

Tras pulsar "Commit" debería forzarse el check unos segundos más tarde de la hora futura que hemos especificado y entonces podremos ver el estado de los servicios.



Ilustración 13. Host cmds

También es posible ejecutar comandos para monitorizar los servicios desde la terminal, simplemente tenemos que poner el comando con sus correspondiente parámetros. Ejemplo:

```
# cd /usr/local/nagios/libexec/  
#./check_nt -H 192.168.1.50 -v UPTIME -p 12489 -s password
```

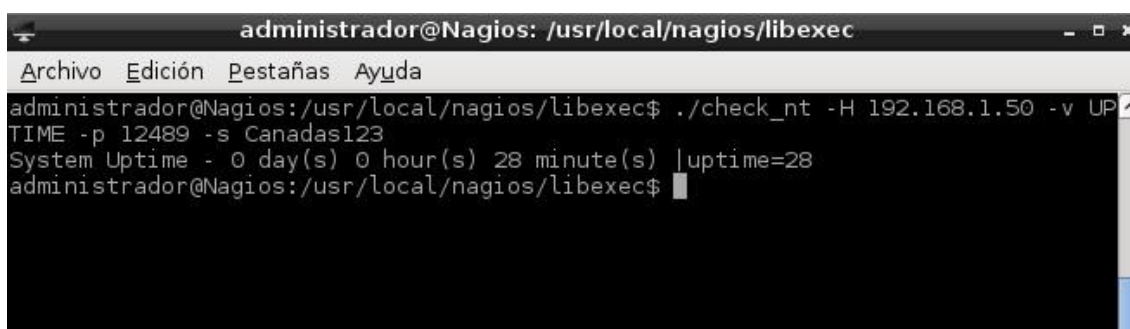


Ilustración 14. Prueba de monitorización a través de la terminal



## Información adicional

El comando `check_nt!MEMUSE` muestra la suma de la memoria física + virtual y en el caso del disco duro muestra la capacidad por porcentaje, esto para mi no es muy razonable y por tanto si queremos cambiarlo para que muestre solo la memoria física y la capacidad de disco en unidades deberíamos hacer uso de SNMP y NRPE.

**Plantillas** > `/usr/local/nagios/etc/objects/templates.cfg`

En este archivo se encuentran cinco plantillas preconfiguradas:

- **Generic-host**: Plantilla genérica que usan el resto de plantillas.
- **Linux-server**: Plantilla para servidores Linux. Es la que se usa para el servidor Nagios (localhost).
- **Windows-server**: Plantilla para servidores Windows.
- **Generic-printer**: Plantilla para monitorizar impresoras.
- **generic-switch**: Plantilla para monitorizar switches.

Hay infinidad de directivas que podemos editar y añadir en estas plantillas para modificar los hosts a nuestro gusto.

**Tiempos** > `/usr/local/nagios/etc/objects/timeperiods.cfg`

En este fichero se pueden establecer ciertos periodos de tiempo para definir cuando queremos que nuestros hosts sean monitorizados y que se envíen notificaciones.

Es muy útil porque se pueden especificar intervalos de tiempo muy concretos para todo tipo de situaciones que nos puedan surgir.

## 2.3.2 Equipos Linux

Nos dirigimos a nuestro cliente Linux y comenzamos instalando los paquetes Build-Essential y Xinetd:

```
# apt-get install build-essential xinetd
```

Volvemos a instalar Nagios Plugins desde el [enlace oficial](http://www.nagios-plugins.org/download/nagios-plugins-2.1.1.tar.gz) seleccionando nuevamente "Latest Stable release" y descargamos, descomprimos e instalamos en el servidor:

```
# wget http://www.nagios-plugins.org/download/nagios-plugins-2.1.1.tar.gz
# tar xvzf nagios-plugins-2.1.1.tar.gz
# cd nagios-plugins-2.1.1/
# ./configure
# make && make install
```

Creamos un usuario para Nagios y le hacemos con el comando Chown propietario de las siguientes carpetas:

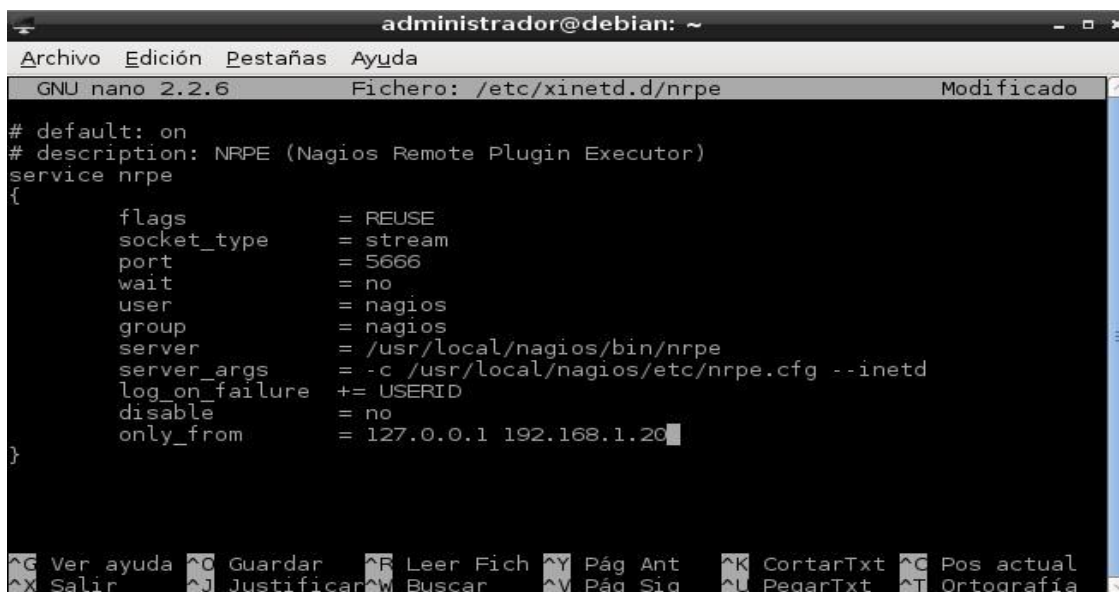
```
# adduser nagios
# chown nagios.nagios /usr/local/nagios/
# chown -R nagios.nagios /usr/local/nagios/libexec/
```

Ahora [descargamos](#), compilamos e instalamos NRPE:

```
# cd
# apt-get install libssl-dev
# wget https://sourceforge.net/projects/nagios/files/nrpe-2.x/nrpe-2.15/nrpe-2.15.tar.gz
# tar xvzf nrpe-2.15.tar.gz
# cd nrpe-2.15
# ./configure --enable-command-args --with-ssl=/usr/bin/openssl
--with-ssl-lib=/usr/lib/x86_64-linux-gnu
# make all
# make install-plugin
# make install-daemon
# make install-daemon-config
# make install-xinetd
```

Añadimos la IP del server al archivo "nrpe" de "xinetd.d" en la directiva que pone "only\_from":

```
# nano /etc/xinetd.d/nrpe
```



```
administrador@debian: ~
GNU nano 2.2.6  Archivo  Edición  Pestañas  Ayuda
Fichero: /etc/xinetd.d/nrpe  Modificado

# default: on
# description: NRPE (Nagios Remote Plugin Executor)
service nrpe
{
    flags                = REUSE
    socket_type          = stream
    port                 = 5666
    wait                 = no
    user                  = nagios
    group                 = nagios
    server                = /usr/local/nagios/bin/nrpe
    server_args           = -c /usr/local/nagios/etc/nrpe.cfg --inetd
    log_on_failure        += USERID
    disable               = no
    only_from             = 127.0.0.1 192.168.1.20
}

^G Ver ayuda  ^C Guardar  ^R Leer Fich  ^Y Pág Ant  ^K CortarTxt  ^O Pos actual
^X Salir      ^J Justificar  ^W Buscar    ^V Pág Sig  ^L PegarTxt   ^T Ortografía
```

Ilustración 15. Añadiendo dirección IP del servidor en NRPE del cliente.

Comprobamos que en el archivo `/etc/services` hay una línea descomentada con "nrpe 5666/tcp" con este comando:

```
# grep nrpe /etc/services
```

Accedemos al archivo de config. principal de Nagios y abajo del todo, descomentamos las líneas de los cuatro últimos comandos (se pueden añadir nuevos):

```
command[check_users]=/usr/local/nagios/libexec/check_users -w
$ARG1$ -c $ARG2$
command[check_load]=/usr/local/nagios/libexec/check_load -w
$ARG1$ -c $ARG2$
command[check_disk]=/usr/local/nagios/libexec/check_disk -w
$ARG1$ -c $ARG2$ -p $ARG3$
command[check_procs]=/usr/local/nagios/libexec/check_procs -w
$ARG1$ -c $ARG2$ -s $ARG3$
```

Además buscaremos la variable `dont_blame_nrpe=0` y la activaremos cambiando su valor a 1:

```
dont_blame_nrpe=1
```

Al finalizar los cambios reiniciamos Xinetd:

```
# service xinetd restart
```

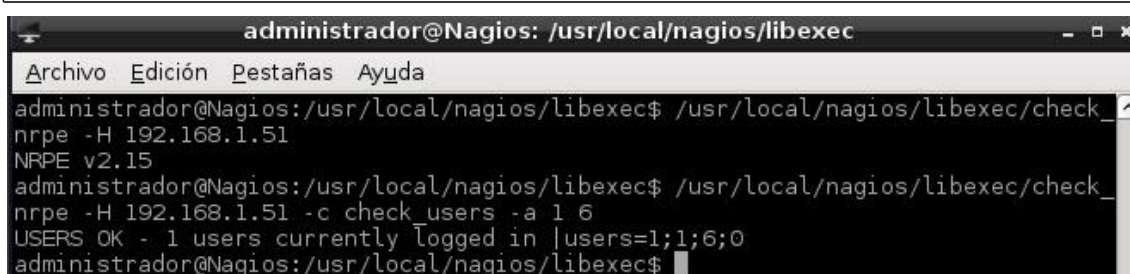
## Comprobaciones desde el servidor

Comprobar funcionamiento de NRPE:

```
# /usr/local/nagios/libexec/check_nrpe -H 192.168.1.51
```

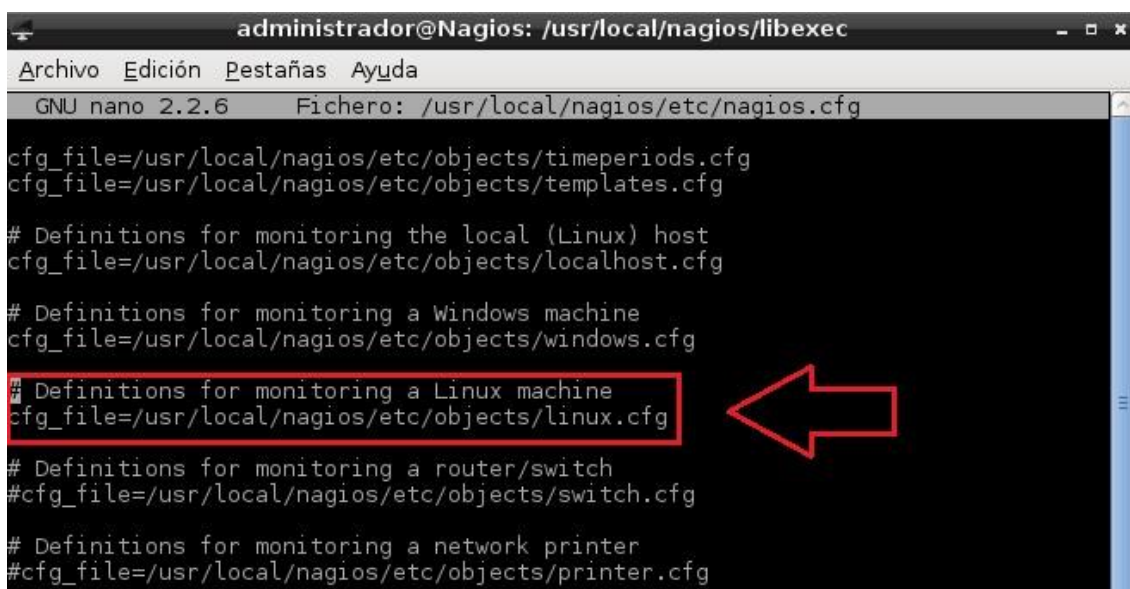
Comprobar funcionamiento pasando un ej. con argumentos:

```
# /usr/local/nagios/libexec/check_nrpe -H 192.168.1.51 -c  
check_users -a 1 6
```



```
administrador@Nagios: /usr/local/nagios/libexec
Archivo Edición Pestañas Ayuda
administrador@Nagios:/usr/local/nagios/libexec$ /usr/local/nagios/libexec/check_
nrpe -H 192.168.1.51
NRPE v2.15
administrador@Nagios:/usr/local/nagios/libexec$ /usr/local/nagios/libexec/check_
nrpe -H 192.168.1.51 -c check_users -a 1 6
USERS OK - 1 users currently logged in |users=1;1;6;0
administrador@Nagios:/usr/local/nagios/libexec$
```

A continuación, para que la monitorización sea automática editaremos el archivo principal en el servidor añadiendo la siguiente línea bajo la sección "OBJECT CONFIGURATION FILES (S)":



```
administrador@Nagios: /usr/local/nagios/libexec
Archivo Edición Pestañas Ayuda
GNU nano 2.2.6 Archivo: /usr/local/nagios/etc/nagios.cfg
cfg_file=/usr/local/nagios/etc/objects/timeperiods.cfg
cfg_file=/usr/local/nagios/etc/objects/templates.cfg
# Definitions for monitoring the local (Linux) host
cfg_file=/usr/local/nagios/etc/objects/localhost.cfg
# Definitions for monitoring a Windows machine
cfg_file=/usr/local/nagios/etc/objects/windows.cfg
# Definitions for monitoring a Linux machine
cfg_file=/usr/local/nagios/etc/objects/linux.cfg
# Definitions for monitoring a router/switch
#cfg_file=/usr/local/nagios/etc/objects/switch.cfg
# Definitions for monitoring a network printer
#cfg_file=/usr/local/nagios/etc/objects/printer.cfg
```

Ilustración 17. Definición de nuestro nuevo Host Linux

A continuación crearemos nuestro nuevo archivo "linux.cfg" junto con su definición de host y servicios:

```
# sudo nano /usr/local/nagios/etc/objects/linux.cfg
```

Definimos el host, grupo perteneciente y seis servicios, cuatro con parámetros que vienen en el NRPE del host linux y otros dos para mostrar que no es necesario usar NRPE en linux para monitorizar algo (PING y SSH).

```
# /usr/local/nagios/etc/objects/linux.cfg
```

```
define host{
    use          linux-server      ; Inherit default values from a template
    host_name    Linux 51          ; The name we're giving to this host
    alias        Equipo 51         ; A longer name associated with the host
    address      192.168.1.51      ; IP address of the host
    hostgroups   linux-clients
}
```

```
define hostgroup{
    hostgroup_name linux-clients  ; The name of the hostg$
    alias          Clientes Linux ; Long name of the group
}
```

```
define service{
    use          generic-service
    host_name    Linux 51
    service_description Users Check
    check_command check_nrpe!check_users!-a 1 5
}
```

```
define service{
    use          generic-service
    host_name    Linux 51
    service_description Load
    check_command check_nrpe!check_load!-a 512 768
}
```

```
define service{
    use                generic-service
    host_name          Linux 51
    service_description Root Partition
    check_command       check_nrpe!check_disk!-a 10 5 / GB
}

define service{
    use                generic-service
    host_name          Linux 51
    service_description Check Process
    check_command       check_nrpe!check_procs!-a 10 50 STATUSFLAGS
}

define service{
    use                generic-service      ; Name of servi$
    host_name          Linux 51
    service_description SSH
    check_command       check_ssh
    notifications_enabled 0
}

define service{
    use                generic-service      ; Name of servi$
    host_name          Linux 51
    service_description PING
    check_command       check_ping!100.0,20%!500.0,60%
}
```

Por último antes de realizar las pruebas, añadiremos al final de la lista el comando `check_nrpe` en "commands.cfg":

```
# /usr/local/nagios/etc/objects/commands.cfg
# 'check_nrpe' command definition
define command{
    command_name check_nrpe
    command_line $USER1$/check_nrpe -H $HOSTADDRESS$ -c $ARG1$
    $ARG2$
}
```

Aquí como podemos comprobar, nuestro host Linux se ha cargado correctamente con todos los servicios iniciados con sus correspondientes estados:



Ilustración 18. Funcionamiento correcto del host Linux desde Interfaz

A la derecha se muestra el ejemplo de un servicio de nuestro host Linux que se ha cargado correctamente.

Como se puede observar, su estado es OK porque nosotros le habíamos pasado en los argumentos que su estado fuese Warning a partir de 2 usuarios.

Service State Information	
Current Status:	<b>OK</b> (for 0d 0h 39m 59s)
Status Information:	USERS OK - 1 users currently logged in
Performance Data:	users=1;1;5;0
Current Attempt:	1/3 (HARD state)
Last Check Time:	02-17-2016 20:35:14
Check Type:	ACTIVE
Check Latency / Duration:	0,000 / 0,000 seconds
Next Scheduled Check:	02-17-2016 20:45:14
Last State Change:	02-17-2016 20:05:14
Last Notification:	N/A (notification 0)
Is This Service Flapping?	<b>NO</b> (5,00% state change)
In Scheduled Downtime?	<b>NO</b>
Last Update:	02-17-2016 20:45:03 ( 0d 0h 0m 10s ago)
Active Checks:	<b>ENABLED</b>
Passive Checks:	<b>ENABLED</b>
Obsessing:	<b>ENABLED</b>
Notifications:	<b>ENABLED</b>
Event Handler:	<b>ENABLED</b>
Flap Detection:	<b>ENABLED</b>

Ilustración 19. Ejemplo de un servicio de host Linux funcionando.

Una vez concluida la parte de instalación y configuración de equipos Windows y Linux es el turno de empezar a manejar el sistema de notificaciones en tres plataformas distintas.

## 2.4 Sistema de notificaciones Nagios



He probado el sistema de notificaciones monitorizando hosts de Nagios con cuatro sistemas distintos, estos han sido los resultados finales:

Email	✓	Postfix	
Android	✓	aNag	
SMS	✓	DescomSMS	
Whatsapp	⚠	Yowsup	

Todos han funcionado a la perfección a excepción de Whatsapp, el cual me ha dado problemas a la hora de compilar el programa (Yowsup) y al dar de alta en Whatsapp algún número que no esté registrado actualmente.

### 2.4.1 Notificaciones vía Email

Se realizará la implementación con el servidor SMTP externo de Gmail y con Postfix puesto que es el agente de correo linux que hemos visto en clase y que por tanto he investigado a fondo para tratar de implementarlo.

Instalación de Postfix y paquetes necesarios:

```
# apt-get install postfix mailutils libsasl2-2 ca-certificates  
libsasl2-modules
```



Accedemos con nano al fichero de configuración de postfix ubicado en `/etc/postfix/main.cf`, comentamos la línea `"relayhost"` con `#` y la añadimos al final del fichero con el resto de líneas nuevas quedando así:

```
#/etc/postfix/main.cf
relayhost =[smtp.gmail.com]:587
smtp_sasl_auth_enable = yes
smtp_sasl_password_maps = hash:/etc/postfix/sasl_passwd
smtp_sasl_security_options = noanonymous
smtp_tls_CAfile = /etc/postfix/cacert.pem
smtp_use_tls = yes
```

Creamos con nano el fichero `/etc/postfix/sasl_passwd` y dentro añadimos lo siguiente:

```
#/etc/postfix/sasl_passwd
[smtp.gmail.com]:587 TUCORREO@gmail.com:TUCONTRASEÑA
```

Establecemos permisos y actualizamos config. de Postfix:

```
#chmod 600 /etc/postfix/sasl_passwd
#postmap /etc/postfix/sasl_passwd
#chmod 600 /etc/postfix/sasl_passwd.db
```

Verificamos nuestro certificado con el siguiente comando:

```
# cat /etc/ssl/certs/Thawte_Premium_Server_CA.pem | tee -a
/etc/postfix/cacert.pem
```

En caso de generar algún error verificamos que tenemos las librerías SSL que instalamos los paquetes en `/etc/ssl/certs/`.

El resultado debería ser parecido al que se muestra:



Ilustración 21. Activación de Aplicaciones menos seguras en Gmail

```
# echo "Probando Postfix + nagios" | mail -s "mi primera
notificacion nagios" sergioas1996@gmail.com
```

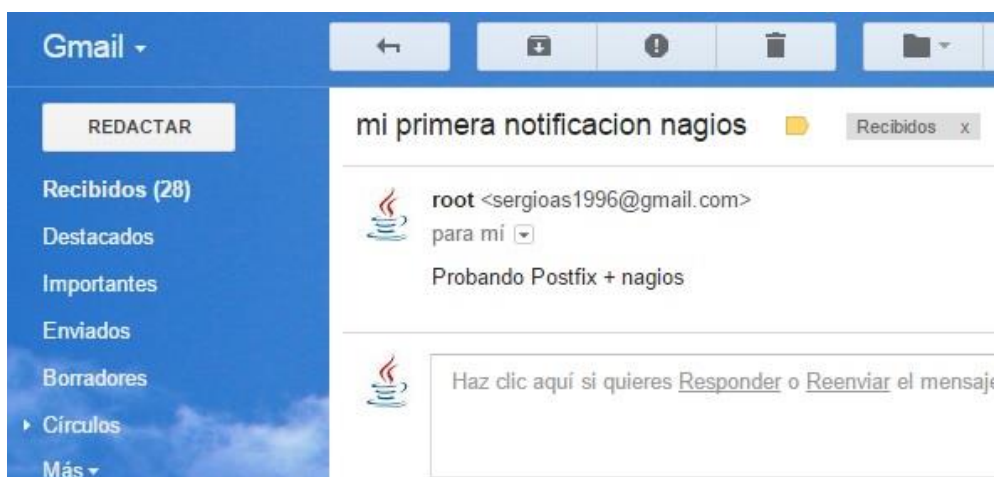


Ilustración 22. Ejemplo de correo generado desde un terminal

A continuación configuramos las notificaciones en el servidor en `"/usr/local/nagios/etc/objects/contacts.cfg"` sustituyendo `"nagios@localhost"` por el destinatarios a notificar:

**#usr/local/nagios/objects/contacts.cfg**

```
define contact{
    contact_name      nagiosadmin      ;
    use               generic-contact  ;
    alias             Nagios Admin    ;
    email             sergioas1996@gmail.com ;
}
```

\*Si son varios destinatarios deben ir separados por comas.

Accedemos al fichero `templates.cfg` y en la plantilla `"windows-server"` modificamos simplemente los siguientes valores:

**#usr/local/nagios/objects/templates.cfg**

```
notification_interval 30 ; Reenviar not. cada 30 minutos
notification_options   d,r ; Mandarlas solo en dichos estados
```

En la parte de servicios, modificamos la plantilla `"generic-service"` para que notifique estado crítico cada 30 min:

**#usr/local/nagios/objects/templates.cfg**

```
notification_options   c      ;
notification_interval   30     ;
```

Verificamos que la configuración es correcta con el comando:

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

Reiniciamos el servidor Nagios:

```
# service nagios restart
```

Ahora solo tenemos que esperar a recibir notificaciones o podemos enviar notificaciones personalizadas al instante entrando en nuestro host y pulsando sobre "Send custom host notification":

The screenshot displays the Nagios web interface for a host named 'Equipos 50 (Windows 50)' with IP 192.168.1.50. The 'Host State Information' section shows the host is DOWN (CRITICAL) for 0d 3h 32m 40s. The 'Host Commands' section on the right has a red arrow pointing to the 'Send custom host notification' option. Below the main information, there are sections for 'Host Comments' and 'Host Information'.

**Host Information**  
Last Updated: Wed Feb 17 22:57:06 CET 2016  
Updated every 30 seconds  
Nagios Core™ 4.1.1 - www.nagios.org  
Logged in as nagiosadmin

View Status Detail For This Host  
View Alert History For This Host  
View Trends For This Host  
View Alert Histogram For This Host  
View Availability Report For This Host  
View Notifications For This Host

Host: Equipos 50 (Windows 50)  
Member of: equipos  
192.168.1.50

**Host State Information**  
Host Status: DOWN (for 0d 3h 32m 40s)  
Status Information: CRITICAL - Host Unreachable (192.168.1.50)  
Performance Data: 10/10 (HARD state)  
Current Attempt: 02-17-2016 22:55:28  
Last Check Time: 02-17-2016 22:55:28  
Check Type: ACTIVE  
Check Latency / Duration: 0.000 / 3.000 seconds  
Next Scheduled Active Check: 02-17-2016 23:01:31  
Last State Change: 02-17-2016 19:24:26  
Last Notification: 02-17-2016 22:55:04 (notification 8)  
Is This Host Flapping? NO (0.00% state change)  
In Scheduled Downtime? NO  
Last Update: 02-17-2016 22:57:03 ( 0d 0h 0m 3s ago)

Active Checks: ENABLED  
Passive Checks: ENABLED  
Obsessing: ENABLED  
Notifications: ENABLED  
Event Handler: ENABLED  
Flap Detection: ENABLED

**Host Commands**  
Locate host on map  
Disable active checks of this host  
Re-schedule the next check of this host  
Submit passive check result for this host  
Stop accepting passive checks for this host  
Stop obsessing over this host  
Acknowledge this host problem  
Disable notifications for this host  
Send custom host notification  
Delay next host notification  
Schedule downtime for this host  
Schedule downtime for all services on this host  
Disable notifications for all services on this host  
Enable notifications for all services on this host  
Schedule a check of all services on this host  
Disable checks of all services on this host  
Enable checks of all services on this host  
Disable event handler for this host  
Disable flap detection for this host

**Host Comments**  
Add a new comment Delete all comments  
Entry Time Author Comment Comment ID Persistent Type Expires Actions  
This host has no comments associated with it

Ilustración 23. Envío de notificación personalizada

Un ejemplo de resultado final sería el siguiente:

The screenshot shows a Gmail interface with a notification email from 'sergioas1996@gmail.com'. The email subject is '\*\* PROBLEM Host Alert: Windows 50 is DOWN \*\*'. The body of the email contains the following information:

\*\*\*\*\* Nagios \*\*\*\*\*  
Notification Type: PROBLEM  
Host: Windows 50  
State: DOWN  
Address: 192.168.1.50  
Info: CRITICAL - Host Unreachable (192.168.1.50)  
Date/Time: Wed Feb 17 22:24:26 CET 2016

Ilustración 24. Ejemplo de notificación final de correo

## 2.4.2 Notificaciones vía SMS

A lo largo del desarrollo del trabajo tuve problemas con este tipo de notificaciones porque al parecer era necesario integrar un Modem con GPRS y la documentación estaba muy desactualizada, por tanto opté por Descom SMS, una empresa española profesional que ha lanzado una solución eficaz de integración de notificaciones Nagios vía SMS.

Descargar script de DescomSMS y agregar permisos:

```
# mkdir -p /usr/local/bin  
# wget -O /usr/local/bin/dcsms.pl http://www.descomsms.com/developer/dcsms.pl  
# chmod +x /usr/local/bin/dcsms.pl
```

El siguiente paso es instalar las librerías de perl XML::Parser, Crypt::SSLeay y LWP::UserAgent para que pueda conectarse a su plataforma segura:

```
# apt-get update; sudo apt-get install libcrypt-ssleay-perl  
liblwp-useragent-determined-perl libxml-parser-perl
```

Comando de prueba de envío de SMS:

```
# /usr/local/bin/dcsms.pl -i "tucodigodecliente" -u "usuario" -p  
"pass" -m "destinatario" -t "Prueba de notificación por SMS  
Nagios" -s "DescomSMS"
```

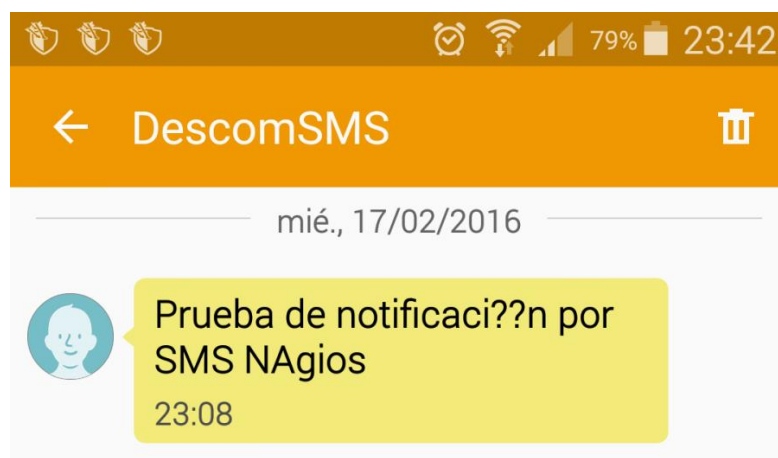


Ilustración 25. Ejemplo de envío de SMS

Para automatizar el envío de notificaciones en Nagios será necesario definir dos nuevos comandos:

**#usr/local/nagios/etc/objects/commands.cfg**

```
# 'notify-host-by-dcsms' command definition
define command{
    command_name notify-host-by-dcsms
    command_line /usr/bin/perl /usr/local/bin/dcsms.pl -i codcli -u
user -p pass -m "$CONTACTPAGER$" -t "$NOTIFICATIONTYPE$: Host
$HOSTALIAS$ is $HOSTSTATE$" -s "Nagios"
}
# 'notify-service-by-dcsms' command definition
define command{
    command_name notify-service-by-dcsms
    command_line /usr/bin/perl /usr/local/bin/dcsms.pl -i codcli -u
user -p pass -m "$CONTACTPAGER$" -t "$NOTIFICATIONTYPE$:
$HOSTALIAS$/ $SERVICEDESC$ is $SERVICESTATE$ ($SERVICEOUTPUT$)" -
s "Nagios"
}
```

Por último es necesario definir un contacto con el tipo de notificaciones a enviar y nº de destinatario:

**#usr/local/nagios/etc/objects/contacts.cfg**

```
define contact{
    contact_name descom
    alias Descom Alarm
    service_notification_period 24x7
    host_notification_period 24x7
    service_notification_options w,u,c,r
    host_notification_options d,u,r
    service_notification_commands notify-service-by-
email,notify-service-by-dcsms
    host_notification_commands notify-host-by-
email,notify-host-by-dcsms
    email soporte@descom.es
    pager 003461111111
}
```

En dicho ejemplo se enviarían notificaciones warning, unknown, critical y recovery a los servicios y down, unknown y recovery a los hosts al número de destinatario 611111111.

## 2.4.3 Notificaciones vía Android

He optado por integrar notificaciones con aplicaciones para smartphones porque en mi opinion es la opción más completa y personalizable con diferencia.

Ventajas como poder monitorizar ciertas partes del servidor a través de la app o notificar a través de vibración o iluminación de "X" LED con "X" color según el estado me parece una auténtica novedad.

Hay mucho soporte y las hay para todo tipo de sistemas, concretamente la más famosa para iOS es "Touchmon" y la más famosa y poderosa para Android es "aNag", la cual veremos por encima en esta guía.

Empezamos descargando la app desde el Play Store de nuestro smartphone, aquí teneis su [enlace oficial](#) por si acaso.

A continuación, en la aplicación hacemos clic sobre la ruedecita de Ajustes del panel superior, Monitoring instances - Add instance y este sería un ejemplo con mi servidor:

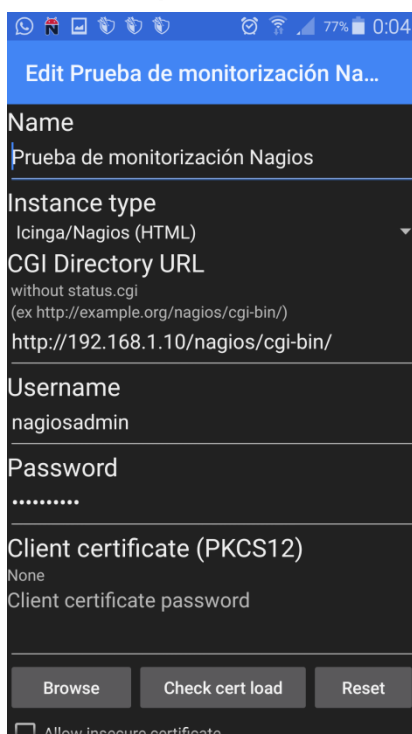


Ilustración 26. Conexión a app Android notificaciones

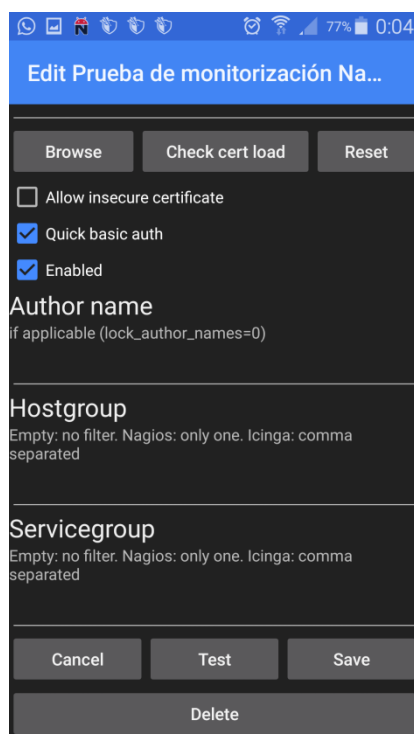


Ilustración 27. Conexión a app Android notificaciones



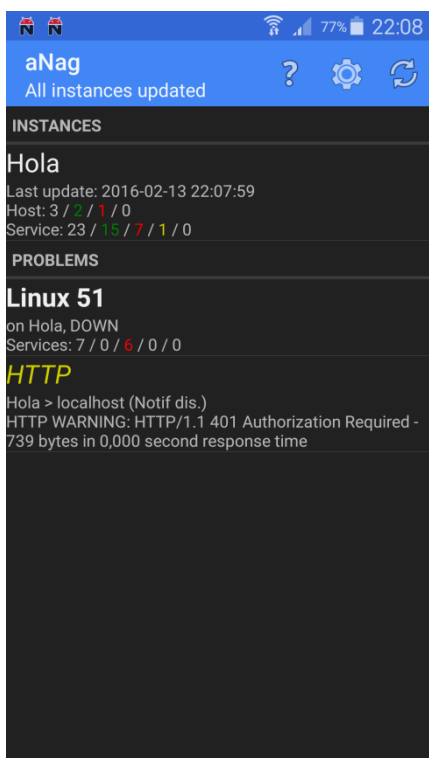


Ilustración 28. Control de hosts y servicios con app

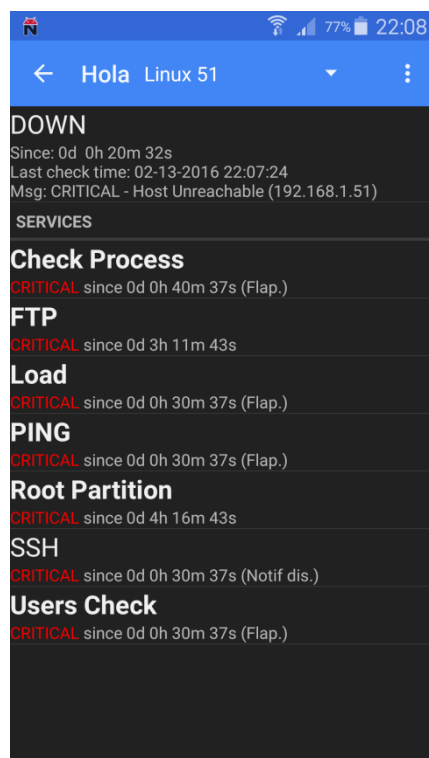


Ilustración 29. Control de un host en estado DOWN.

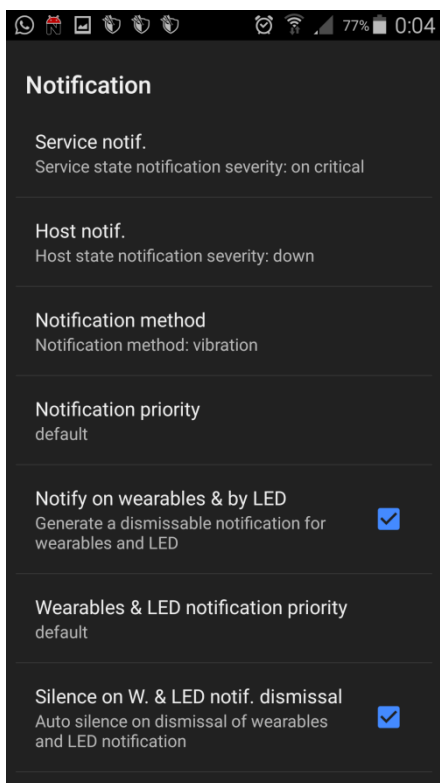


Ilustración 30. Opciones avanzadas de notificación muy completas

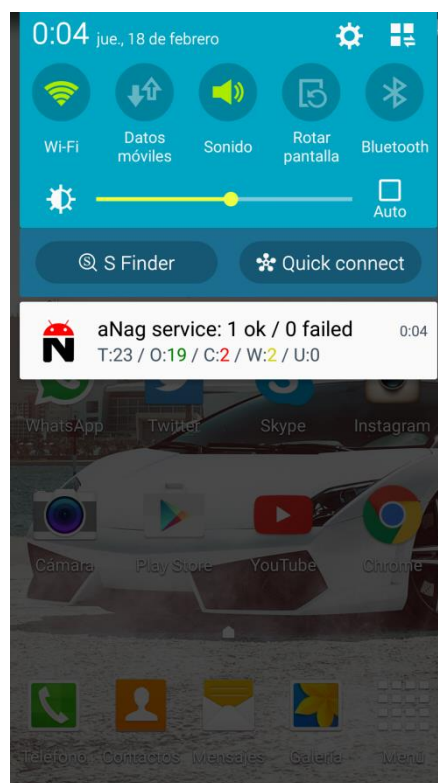


Ilustración 31. Notificaciones y información avanzada



## 3. Conclusiones

La principal conclusión que he sacado de este proyecto ha sido comprender el funcionamiento de los sistemas de monitorización tras documentarme e investigar a fondo y comprender la enorme importancia que supone para los Servicios en red.

Estoy muy contento con la estructura que he seguido a la hora de realizar el trabajo documentándome y basándome en la estructura de algunos proyectos de fin de carrera, en gran parte me ha ayudado a mejorar muchísimo y a organizarme todos los puntos de manera mucho más ordenada y cómoda.

He de añadir que la gente que he contactado por correo, foros y la empresa (Descom SMS) que me ha regalado 20 SMS gratis para realizar pruebas con notificaciones de Nagios tras exponerles mi situación, se han portado todos genial conmigo en todo momento.

Las desviaciones que me han surgido a la hora de realizar el trabajo han sido las siguientes:

- A pesar de investigar mucho sabiendo que era un tema complicado, no he sido capaz de implementar a la perfección un event\_handler/manejador de eventos, mi idea era realizar uno que reiniciase un servicio automáticamente pero no fuí capaz de implementarlo totalmente, abrí [este hilo](#) en los foros oficiales en el que se puede ver preguntas y respuestas.
- No he sido capaz de monitorizar como parte "extra" alguno de los servicios que hemos visto en clase, intenté realizarlo con FTP siguiendo [este PDF oficial](#) de Nagios pero tuve problemas con las "iptables routes" a la hora de añadir los puertos 20 y 21 a las excepciones y no fuí capaz de monitorizarlo como me hubiese gustado.

## 4. Bibliografía

Principales fuentes externas consultadas:

- [Documentación Nagios Core](#)
- [Youtube](#) (un ejemplo de los muchos videos que he mirado)
- [Trabajo de fin de carrera de monitorización de servicios Linux, Universidad de Barcelona](#). De gran utilidad para ayudarme a estructurar mi trabajo.
- [Foros Nagios Core](#)
- [Empresa Descom SMS](#)
- [Notificaciones Email Postfix](#)
- [Notificaciones Whatsapp](#)
- [Proyecto Nagios en servidores Debian 2010-2011](#)
- [Gmail](#) (permitirme usar su servidor SMTP externo)

Además de todas las fuentes consultadas durante el trabajo, quiero agradecer el soporte recibido durante casi 1 semana por parte del técnico en sistemas Xavier Carretero cuyo canal de Youtube es [este](#).

Quiero agradecer a la empresa Descom SMS por regalarme 20 SMS al presentarles mi propuesta y pedirles el favor para implementar la solución de notificación por SMS sin coste adicional.

Además quiero agradecer a mi profesor de la asignatura del trabajo (Mario Dorrego Martín) por dar la propuesta del trabajo entre otras muchísimas y a un compañero de clase (Ramsés Diez Galván) por pasarme documentación en Ubuntu Server sobre Nagios para ver de que iba el tema e interesarme.

## 5. Apéndice

<b>Ilustración 1. Configuración de Adaptadores de red en VirtualBox.</b>	.....	iError! Marcador no definido.
<b>Ilustración 2. Ejemplo de instalación correcta de Apache Web server</b>	.....	iError! Marcador no definido.
<b>Ilustración 3. Ejemplo de correcta instalación correcta de librerías</b>	.....	iError! Marcador no definido.
<b>Ilustración 5. Descarga de <i>Nagios Plugins</i> 2.1.1 desde web oficial</b>	.....	iError! Marcador no definido.
<b>Ilustración 6. Ejemplo de correcta instalación de <i>Nagios Plugins</i></b>	.....	iError! Marcador no definido.
<b>Ilustración 7. Ejemplo de listado de plugins de Nagios correcto</b>	.....	iError! Marcador no definido.
<b>Ilustración 8. Ejemplo de autenticación en Interfaz Web Nagios</b>	.....	iError! Marcador no definido.
<b>Ilustración 9. Ejemplo de index correcta de Interfaz Web Nagios</b>	.....	iError! Marcador no definido.
<b>Ilustración 10. Ejemplo válido de configuración de NSClient++</b>	.....	iError! Marcador no definido.
<b>Ilustración 11. Pasos para reiniciar servicio Nagios desde Interfaz Web</b>	.....	iError! Marcador no definido.
<b>Ilustración 12. Visualización de estado de nuestros hosts .....</b>		iError! Marcador no definido.
<b>podremos ver el estado de los servicios. Ilustración 13. Host cmds</b>	.....	iError! Marcador no definido.
<b>Ilustración 14. Prueba de monitorización a través de la terminal</b>	.....	iError! Marcador no definido.
<b>Ilustración 15. Añadiendo dirección IP del servidor en NRPE del cliente.</b>	.....	iError! Marcador no definido.
<b>Ilustración 17. Definición de nuestro nuevo Host Linux .....</b>		iError! Marcador no definido.

<b>Ilustración 18. Funcionamiento correcto del host Linux desde Interfaz.....</b>	<b>iError! Marcador no definido.</b>
<b>Ilustración 19. Ejemplo de un servicio de host Linux funcionando. ....</b>	<b>iError! Marcador no definido.</b>
<b>Ilustración 20. Ejemplo de certificado generado correctamente ....</b>	<b>iError! Marcador no definido.</b>
<b>Ilustración 21. Activación de Aplicaciones menos seguras en Gmail ....</b>	<b>iError! Marcador no definido.</b>
<b>Ilustración 22. Ejemplo de correo generado desde un terminal ....</b>	<b>iError! Marcador no definido.</b>
<b>Ilustración 23. Envío de notificación personalizada .</b>	<b>iError! Marcador no definido.</b>
<b>Ilustración 24. Ejemplo de notificación final de correo .....</b>	<b>iError! Marcador no definido.</b>
<b>Ilustración 25. Ejemplo de envío de SMS</b>	<b>iError! Marcador no definido.</b>
<b>Ilustración 26. Conexión a app.....</b>	<b>Ilustración 27. Conexión a app..... iError! Marcador no definido.</b>
<b>Ilustración 28. Control de .....</b>	<b>Ilustración 29. Control de un .....</b>
<b>hosts y servicios con app.....</b>	<b>host en estado DOWN. iError! Marcador no definido.</b>
<b>Ilustración 30. Opciones avanzadas</b>	<b>Ilustración 31. Notificaciones y ..... iError! Marcador no definido.</b>
<b>de notificación muy completas</b>	<b>información avanzada iError! Marcador no definido.</b>

## 6. Glosario

- check\_nt: plugin para monitorizar recursos hardware de equipos Windows facilmente.
- Critical: estado que se produce cuando hay un problema grave.
- Event Handler: es el manejador de eventos, indica un comando a ejecutar cuando se produce un cambio de estado.
- Flapping: es un estado especial que tienen los host y los servicios, ocurre cuando su estado cambia muchas veces en poco tiempo.

- Generic-service: plantilla predefinida para servicios.
- Generic-host: Plantilla genérica que usan el resto de plantillas.
- HARD: Estado que se produce cuando el problema o recuperación es real.
- Local-service: Hereda de generic-service y es usada para los servicios de Nagios (localhost).
- Linux-server: Plantilla para servidores Linux. Es la que se usa para Nagios (localhost).
- Make: herramienta de gestión de dependencias
- Notificaciones: mensajes de alertas enviados a un contacto cuando sucede algo.
- NSClient: agente, el cual actúa como un proxy entre el plugin que hace la monitorización y la máquina.
- NRPE: es un pequeño daemon que permite ejecutar plugins locales en equipos remotos de equipos Linux
- Objeto nagios: nombre que reciben las definiciones de hosts, contactos, servicios, comandos etc...
- OK: estado que dice que funciona correctamente.
- Runlevels: modo de operación en los sistemas operativos que implementan el estilo de sistema de arranque de iniciación tipo UNIX System V.
- Servicios: son procesos informáticos a los que puede referirse tanto para hosts (POP,SMTP, HTTP, ...) como para cualquier otro tipo de métrica asociada a otro (Ping, Usuarios logueados, Espacio en disco, Memoria, ...).
- SNMP: protocolo de la capa de aplicación que facilita el intercambio de información de administración entre dispositivos de red.
- SOFT: Estado que se produce en un host cuando hay un problema o se produce la recuperación de uno pero todavía no se confirma.
- SMTP: protocolo de red utilizado para el intercambio de mensajes de correo electrónico entre computadoras u otros dispositivos.

- Time periods: se usan para definir periodos de tiempo concretos en los que se monitoriza un equipo.
- Unknown: estado que dice que hay algún problema desconocido.
- Warning: estado que se produce cuando hay un problema leve o se está cerca de uno grave.
- Windows-server: Plantilla para servidores Windows. Es la que venimos usando.
- Xinetd: es un servicio o demonio que usan gran parte de los sistemas Unix dedicado a administrar la conectividad basada en Internet.