



## MÓDULO: PROYECTO

	<b>ALUMNO/A: LARA SANTIUSTE GÓMEZ</b>
	<b>CURSO: 2021/2022</b>
	<b>CICLO: GRADO SUPERIOR</b>
	<b>FAMILIA PROFESIONAL: INFORMÁTICA Y COMUNICACIONES</b>
<b>RESPONSABLE/S DEL SEGUIMIENTO</b>	<b>JOSÉ LUIS CAMPOS REINA</b>
<b>TÍTULO DEL PROYECTO</b>	<b>MONITORIZACIÓN CON NAGIOS</b>
<b>DESCRIPCIÓN DEL PROYECTO</b>	<b>IMPLEMENTACIÓN DE UN SERVIDOR NAGIOS CAPAZ DE MONITORIZAR CPU Y ESPACIO DE DISCO DE SERVIDORES O EQUIPOS ADEMÁS DE MONITORIZAR EL TRÁFICO Y LA CONECTIVIDAD DE SWITCHES Y ROUTERS.</b>

# ÍNDICE

1. [INTRODUCCIÓN](#)
2. [ORIGEN Y CONTEXTUALIZACIÓN DEL PROYECTO](#)
3. [OBJETIVO GENERAL DEL PROYECTO](#)
4. [OBJETIVOS ESPECÍFICOS](#)
5. [TAREAS](#)
  - 5.1 [Tarea 1: Instalación Nagios](#)
  - 5.2 [Tarea 2: Programar Notificaciones](#)
  - 5.3 [Tarea 3: Monitorización de Servidor y Equipo](#)
  - 5.4 [Tarea 4: Monitorización de Switch y Access Point](#)
  - 5.5 [Tarea 5: Implementación de herramientas de gráficos estadísticos](#)
  - 5.6 [Tarea 6: Investigación y Documentación](#)
6. [RECURSOS MATERIALES](#)
7. [RECURSOS HUMANOS](#)
8. [CRONOGRAMA](#)
9. [BIBLIOGRAFÍA](#)
10. [ANEXO I](#)
11. [ANEXO II](#)

## **INTRODUCCIÓN**

Nagios es un sistema de monitorización de redes ampliamente utilizado, tanto para monitorizar equipos (hardware) y servicios (software) alertando cuando el comportamiento de los mismos no es el deseado.

Al ser unos sistemas de monitorización más antiguos con código abierto, usuarios como sus desarrolladores han tenido dos décadas para desarrollar plugins y herramientas a medida que surgen nuevas necesidades, por lo tanto dispone de más funcionalidades que otros sistemas de monitorización.

Entre sus características principales figuran la monitorización de servicios de red (como SMTP, HTTP, SNMP), monitorización de recurso del sistema (carga del procesador, uso de los discos, memoria, estado de los puertos...) independientemente del sistema operativo, posibilidad de monitorización por túneles de SSH y SSL cifrados y la posibilidad de programar plugins específicos.

Se trata de un software que otorga mucha versatilidad, se puede consultar casi cualquier parámetro de interés y genera alertas

## **ORIGEN Y CONTEXTUALIZACIÓN DEL PROYECTO**

La idea de implementar Nagios surgió debido al crecimiento de la empresa y en consecuencia el aumento de los dispositivos de red, equipos y servidores.

Debido a esto se ha creado la necesidad de tener más control además de centralizarlo con el fin de tener fácil acceso a la información y aumentar la disponibilidad.

La empresa cuenta con tres switches, un access point y una máquina virtual que debe monitorizarse. Todos los dispositivos de red trabajan en la red 10.90.90 en cambio la máquina a monitorizar y el futuro servidor Nagios trabajan en la red 10.1.1 por lo que se ha creado una interfaz de red para conectar el servidor a los dispositivos de red.

## **OBJETIVO GENERAL DEL PROYECTO**

Aplicar los conocimientos obtenidos con los módulos de administración de sistemas operativos y servicios de red e Internet para conocer Nagios, sus plugins de monitorización principales, como funciona la creación de alertas en distintos dispositivos y ver diferentes herramientas estadísticas.

## **OBJETIVOS ESPECÍFICOS**

Acceso a Nagios desde web, donde se deberán ver los distintos dispositivos( tres switches con sus respectivos puertos en uso, un access point con servicios de control de tráfico de ethernet y wifi, una máquina virtual con monitorización de cpu y espacio de disco)

Acceso por web a PnP4Nagios, Granafa (con dashboards personalizados de los servicios que tenemos) y Nagvis (con mapas de red de los servicios)

## **TAREAS**

Tarea 1: Instalación Nagios

### **Objetivo y metodología**

La finalidad de la primera tarea es la implementación de Nagios en bruto, solo con su paquete de plugins para monitorizar la propia máquina donde se instale. Para esto previamente crearemos una máquina virtual con 2 GB de RAM y 20GB de espacio de almacenamiento e instalaremos el sistema operativo Debian 11 Bullseye.

#### **Subtarea 1.1: Instalación de Debian 11**

Descargar una iso Debian11 desde la página oficial e instalarla en una máquina virtual en Proxmox.

## Subtarea 1.2: Instalación Nagios

Para realizar las configuraciones previas necesitaremos ejecutar comandos con privilegios de root, crear usuario para Nagios, actualizar el sistema operativo, habilitar puertos 22, 80 y 443. Instalar prerequisites( wget, gettext, unzip, gcc, g++ automake autoconf, make, perl, libperl-dev, libnet-snmp-perl, curl, net-tools, snmp, snmptrapd, snmpd, snmptt, apache 2, libapache2-mod-php, libssl-dev, openssl).

Para la descarga de nagios 4.4.6 y sus plugins se puede realizar desde la página web oficial de Nagios.

Descomprimirlos con tar e instalarlos con ./configure y make.

Adicionalmente también se puede crear alias en .bashrc para facilitar el reinicio y ver el estado de Nagios.

```
alias nagioscheck='/usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg'
alias nagiosreload='sudo systemctl restart nagios'
alias apeth0='./check_iftraffic64.pl -H 10.90.90.74 -i eth0'
```

	Duración (horas)	Fecha inicio	Fecha fin
Tarea 1		Lunes 9 de 4 Mayo	Lunes 9 de 4 Mayo
Subtarea 1.1	1		
Subtarea 1.2	2		

## Tarea 2: Programar Notificaciones

### Objetivo y metodología

Conocer cómo programar alertas y las distintas formas de recibirlas.

Empezaré con la forma básica que es por correo electrónico, luego con la aplicación oficial y por último crearé un bot de Telegram enlazado con Nagios para recibir las alertas por mensaje directo al móvil.

#### Subtarea 2.1: Notificaciones por correo electrónico

El primer paso es instalar las dependencias para las notificaciones por correo que son: php, xml, composer, smtp.

Se creará un archivo .json donde añadiremos los datos de la cuenta de correo que enviará las notificaciones. SMTP tiene sus propios correos electrónicos, por ejemplo: smtp.gmail.com y smtp-mail.outlook.com.

Se tendrá que modificar el fichero commands.cfg para que los comandos apunten al cliente smtp y definir contactos para que usen los comandos en contacts.cfg.

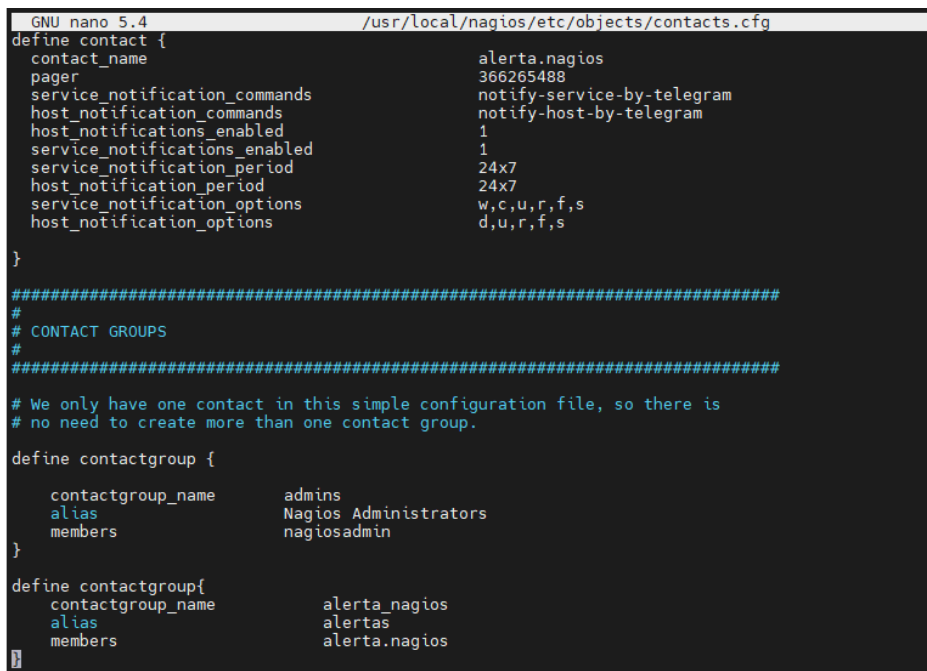
#### SMTP

Un servidor SMTP es un ordenador encargado de llevar a cabo un servicio que permite el transporte del correo electrónico por Internet. El servidor se encarga de confirmar que el remitente y destinatario son adecuados para permitir el envío. Unas de las ventajas de utilizar SMTP es ahorrar la gestión de un propio servidor email.

## Composer

Composer es un gestor de dependencias de php esto quiere decir que es un software que se encarga de evaluar la compatibilidad, de descargar, instalar, actualizar y desinstalar paquetes de software dentro de un proyecto, para ello tenemos que crear un archivo .json donde podemos declarar los parámetros de nuestra aplicación. Con esto conseguimos que composer descargue las librerías necesarias para ejecutar el proyecto.

```
{
  "host": "smtp.gmail.com", {
    "username": "the-senders-email@gmail.com",
    "password": "the-password-of-the-account",
    "secure": "tls",
    "port": 587
  }
}
```



```
GNU nano 5.4 /usr/local/nagios/etc/objects/contacts.cfg
define contact {
    contact_name          alerta.nagios
    pager                 366265488
    service_notification_commands  notify-service-by-telegram
    host_notification_commands  notify-host-by-telegram
    host_notifications_enabled    1
    service_notifications_enabled 1
    service_notification_period   24x7
    host_notification_period      24x7
    service_notification_options  w,c,u,r,f,s
    host_notification_options     d,u,r,f,s
}

#####
#
# CONTACT GROUPS
#
#####

# We only have one contact in this simple configuration file, so there is
# no need to create more than one contact group.

define contactgroup {
    contactgroup_name      admins
    alias                  Nagios Administrators
    members                 nagiosadmin
}

define contactgroup{
    contactgroup_name      alerta_nagios
    alias                  alertas
    members                 alerta.nagios
}
```

## **Directivas de notificación**

Al definir los contactos y servicios podemos añadir las directivas de notificaciones para que nos notifique cuando queramos. Los atributos de directivas son:

Atributos de directiva service\_notification\_options

- w: Notificación si hay warning en el estado de servicio
- u: Notificación si hay unknown en el estado de servicio
- c: Notificación si hay critical en el estado de servicio
- r: Notificación cuando está recuperándose el servicio (ok states)
- f: Notificar cuando el servicio empieza o acaba de estar en flapping
- n (none): Ningún tipo de notificación

Atributos de directiva host\_notification\_options

- d: Notificar si el host está caído
- u: Notificar si el host está inalcanzable
- r: Notificación cuando está recuperándose el host(up states)
- f: Notificar cuando el host empieza o acaba de estar en flapping
- s: Notifica cuando el host o servicio programado para apagarse empieza o acaba
- n (none): Ningún tipo de notificación

## **Subtarea 2.2: Notificaciones por aplicación oficial**

Para utilizar la aplicación oficial primero en Nagios debemos copiar nath\_status.php y ServerAlarmNotify.php desde /usr/local/nagios/share para moverlos a /usr/local/nagios/libexec además de darle permisos de ejecución a la carpeta.

Habilitamos el módulo SSL de Apache y modificar el fichero /usr/local/nagios/etc/objects/commands.cfg para crear los comandos sm-host-push-notify y sm-service-push-notify. En las líneas de ambos comandos aparecerá una clave que será distinta para cada aplicación móvil



así que una vez instalada y vinculada la aplicación será necesario cambiar la clave editando el fichero.

Descargamos e instalamos en el teléfono móvil la aplicación ServerAlarms – Nagios Client en Android e iOS.

Agregamos el servidor Nagios completando los campos que se indican. En el campo PHP SERVER hay que añadir la línea: [https://IP\\_del\\_servidor/nagios/](https://IP_del_servidor/nagios/)

Hacer click en TEST AND UPDATE.

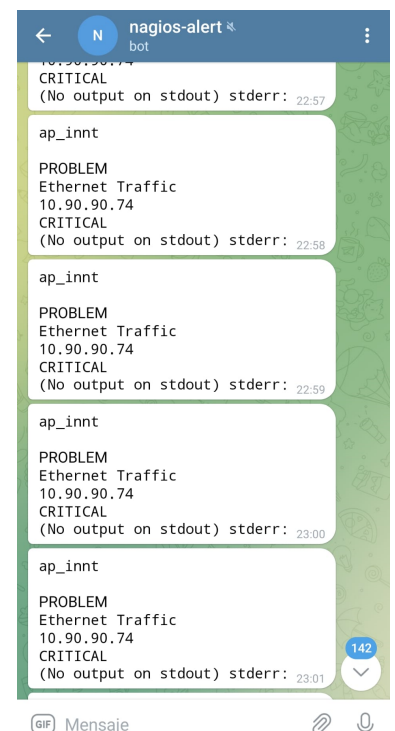
Si el servidor se añadió correctamente, ir a Menú -> Settings y pulsar el botón *SHOW KEY*. Se obtendrá la clave que hay que sobrescribir en el fichero `commands.cfg`, mencionada anteriormente.

### Subtarea 2.3: Notificaciones por Telegram

En Telegram buscamos BotFather, escribimos en un chat `/start` y seguimos los pasos para crear un bot, en mi caso lo llamé `nagios-alert`. Una vez hecho esto guardamos su token.

En Nagios instalamos `python`, `twx.botapi` y el script `nagios_telegram.py`. Editamos el archivo `commands.cfg` para definir dos comandos, uno de `host` y otro de `servicio` para Telegram. Por último editamos el archivo `contacts.cfg` añadiendo los comandos que hemos definido.

Para rellenar el dato `pager` tenemos que saber nuestra `id` de telegram, una forma de saberla es utilizar el bot `userinfobot`.



	Duración (horas)	Fecha inicio	Fecha fin
Tarea 2			
Subtarea 2.1	2	Martes 10 de Mayo	Martes 10 de Mayo
Subtarea 2.2	2	Miércoles 11 de Mayo	Miércoles 11 de Mayo
Subtarea 2.3	2 + 3 extras	Jueves 12 de Mayo	Jueves 13 de Mayo

### Tarea 3: Monitorización de Servidor y Equipo

#### Objetivo y metodología

Hacer que Nagios monitorice el CPU y el espacio de disco de un servidor. Configuraremos previamente Nagios instalando NRPE, posteriormente instalaremos los plugins de Nagios y NRPE en el servidor, definiremos los comandos y los enlazaremos con Nagios.

#### Subtarea 3.1: Instalación NRPE y configuración en Nagios

Descargamos nrpe 4.0.5 desde su página oficial, lo instalamos con `./configure` y `make`. Editamos `commands.cfg` y definimos un comando para `check_nrpe`. Creamos un archivo en `/server` donde definimos el host y los servicios utilizando los comandos que definiremos en los equipos a monitorizar.

```
GNU nano 5.4 /usr/local/nagios/etc/servers/testserver.cfg
define host {
    use                linux-server,host-pnp
    host_name          Proxmox_server
    alias              <An alias for your server, can contain spaces>
    address            10.1.1.21
    max_check_attempts 5
    check_period       24x7
    notification_interval 30
    notification_period 24x7
}
define service {
    use                generic-service,srv-pnp
    host_name          Proxmox_server
    service_description CPU load
    check_command       check_nrpe!check_load
}
define service {
    use                generic-service,srv-pnp
    host_name          Proxmox_server
    service_description /dev/sda2 free space
    check_command       check_nrpe!check_sda2
}
define service {
    use                generic-service,srv-pnp
    host_name          Proxmox_server
    service_description /dev/sda3 free space
    check_command       check_nrpe!check_sda3
}
define service {
    use                generic-service,srv-pnp
    host_name          Proxmox_server
    service_description /dev/sda4 free space
    check_command       check_nrpe!check_sda4
}
```

```
define command{  
    command_name check_nrpe  
    command_line $USER1$/check_nrpe -H $HOSTADDRESS$ -c $ARG1$  
}
```

### Subtarea 3.2: Configuración NRPE en los equipos

Ahora desde el equipo que queremos monitorizar actualizamos e instalamos prerequisites. Descargamos plugins de nagios y nrpe, descomprimoslos e instalamos. Todo esto es una repetición de los pasos anteriores. Vamos al archivo nrpe.cfg y añadimos la ip del servidor de Nagios en la parte de allowed\_hosts y server\_address lo ponemos a 0.0.0.0.

```
allowed_hosts=127.0.0.1,::1,<Nagios server IP>  
server_address=0.0.0.0
```

Por último definimos comandos con los plugins para chequeo de disco y CPU.

```
command[check_sda1]= /usr/local/nagios/libexec/check_disk -w 20% -c 10% -p  
/dev/sda1
```

### NRPE

Nagios Remote Plugin Executor es un servicio o demonio que se instala en las máquinas a monitorizar escuchando para que desde Nagios, con el comando 'check\_nrpe', le podamos hacer peticiones

	Duración (horas)	Fecha inicio	Fecha fin
Tarea 3			
Subtarea 3.1	2 + 6 extras	Viernes 13 de Mayo	Martes 17 de Mayo
Subtarea 3.2	1 + 3 extras	Lunes 16 de Mayo	Miércoles 18 de Mayo

#### Tarea 4: Monitorización de Switch y Access Point

##### Objetivo y metodología

Monitorizar dispositivos de red como el tráfico(Ethernet y Wifi) de un acces point y el tráfico de los puertos de tres switches.

##### Subtarea 4.1: Instalación SNMP

Instalar librerías de NET-SNMP-PERL, instalar traductor de traps snmptt, descargar el código fuente de snmptt y descomprimirlo.

Copiar archivos de snmptt a otro directorio y editar los ficheros snmptt., snmptt.ini, snmptt.conf y logrotate.conf. Crear directorios y ficheros de snmptt con sus respectivos servicios, habilitar puerto 162 y servicios snmptrapd.

Crear un directorio (eventhandlers) y un documento llamado submit\_check\_result y lo editamos. Le añadimos los permisos correspondientes. Después generamos el fichero con información del trap a partir del .mib, el .mib puede ser uno genérico o específico del dispositivo de red, si necesitáramos un específico podríamos descargarlo de la página web del fabricante. Con este comando aparte de un archivo trap generamos un archivo de configuración de snmptt que le tendremos que añadir su ruta en el archivo snmptt.ini.

En templates.cfg definimos el host y el servicio del dispositivo. El servicio que hemos creado tendrá que estar asociado al servicio en el .cfg del dispositivo que previamente tendremos que haber creado. Por último añadimos la ruta del cfg del dispositivo en nagios.cfg

### **Monitoreo con check\_iftraffic64.pl**

Desde la página oficial de Nagios descargamos este plugin en la ruta /usr/local/nagios/libexec, otorgamos los correspondientes permisos y definimos su comando en commands.cfg.

### **SNMP**

Es uno de los protocolos que se usan para administrar y monitorizar dispositivos de red, la mayoría de ellos tienen un agente SNMP instalado aunque en muchos casos hay que activarlo.

### **MIBS**

Los ficheros mibs sirven tanto para SNMP en general como para los traps, lo que hacen es traducir el OID del árbol snmp del fabricante o producto a una cadena de texto más inteligible para así no tener que escribir códigos numéricos cuando se hace una consulta como por ejemplo con snmpget.

Los .mibs tienen siempre una parte común o genérica y una parte del fabricante que cuelga de la rama private. Esa parte es distinta a cada dispositivo y generalmente se puede descargar desde la página web del fabricante del dispositivo. Con los mibs genéricos se pueden hacer cosas como obtener información general de los dispositivos y sus interfaces. Los mibs genéricos se pueden descargar desde la página web de snmp.

```

root@srvNagios:/# ls /usr/share/snmp/mibs/
AP                               NET-SNMP-AGENT-MIB.txt        NET-SNMP-VACM-MIB.txt        wget-log
'attachment.php?link_id=4019'   NET-SNMP-EXAMPLES-MIB.txt    UCD-DEMO-MIB.txt            wget-log.1
'attachment.php?link_id=4019.1' NET-SNMP-EXTEND-MIB.txt      UCD-DISKIO-MIB.txt          wget-log.2
'attachment.php?link_id=4019.2' NET-SNMP-MIB.txt             UCD-DLMOD-MIB.txt
Genericos                       NET-SNMP-PASS-MIB.txt        UCD-IPFWACC-MIB.txt
LM-SENSORS-MIB.txt             NET-SNMP-TC.txt             UCD-SNMP-MIB.txt
root@srvNagios:/# ls /usr/share/snmp/mibs/AP
UBNT-MIB.txt  UBNT-UniFi-MIB.txt
root@srvNagios:/# ls /usr/share/snmp/mibs/Genericos
P-BRIDGE-MIB.mib  Q-BRIDGE-MIB.mib  RMON2-MIB.mib  TOKEN-RING-RMON-MIB.mib
root@srvNagios:/#

```

## TRAPS

Los traps son las alertas que envía snmp cuando los servicios no funcionan como se pretende. En los mibs genéricos vienen definidos algunos traps aunque la mayoría de los traps son propios de cada dispositivo.

### Subtarea 4.2: Configuración para Switch

En /usr/local/nagios/etc/objects/ crearemos un .cfg para el switch, en él habrá que definir el host y el servicio, utilizando el comando que hemos definido y los servicios previamente añadido a templates configuraremos los datos del switch.

Para monitorizar el tráfico del switch habrá que hacerlo puerto por puerto de la siguiente forma:

```

GNU nano 5.4 /usr/local/nagios/etc/objects/sw_innt.cfg
define service {
    use generic-service, srv-pnp
    host_name sw2_innt
    service_description Puerto 01
    check_command check_iftraffic!1!40!60
    contact_groups alerta_nagios
}

define service {
    use generic-service, srv-pnp
    host_name sw2_innt
    service_description Puerto 17
    check_command check_iftraffic!17!40!60
    contact_groups alerta_nagios
}

#Puertos ocupados 10.90.90.85 (Puertos 1,3,6,7,9,10,11,13,15,17,18,20,24)

define service {
    use generic-service, srv-pnp
    host_name sw1_innt
    service_description Puerto 01
    check_command check_iftraffic!1!40!60
    contact_groups alerta_nagios
}

define service {
    use generic-service, srv-pnp
    host_name sw1_innt
    service_description Puerto 03

```

### Subtarea 4.3: Configuración para Access Point

Para configurar el Access Point lo haremos de la misma manera creando un archivo .cfg específico. Para monitorizarlo con iftraffic tendremos que añadir las interfaces de red al comando de la siguiente forma.

```
GNU nano 5.4 /usr/local/nagios/etc/objects/ap_innt.cfg
#definición de host
define host {
    use                generic-ap,host-pnp
    host_name          ap_innt
    alias              ap_inntelia interior
    address            10.90.90.74
}

# Monitor traffic with check_iftraffic64

#define service {
#use                trap-service
#host_name          ap_innt
#service_description Ping
#check_command      check_ping!200.0,20%!600.0,60%
#check_interval     5
#retry_interval     1
#}

define service {
    use                generic-service,srv-pnp
    host_name          ap_innt
    service_description WiFi0
    check_command      check_iftraffic_v1!wifi0!80!95
    contact_groups     alerta_nagios
}

define service {
    use                generic-service,srv-pnp
    host_name          ap_innt
    service_description WiFi1
    check_command      check_iftraffic_v1!wifi1!80!95
}
```

	Duración (horas)	Fecha inicio	Fecha fin
Tarea 4			
Subtarea 4.1	2	Miércoles 18 de Mayo	Jueves 19 de Mayo
Subtarea 4.2	3 + 7 extras	Viernes 20 de Mayo	Martes 24 de Mayo
Subtarea 4.3	2 + 3 extras	Lunes 23 de Mayo	Miércoles 25 de Mayo

## Tarea 5: Implementación de herramientas de gráficos estadísticos

### Objetivo y metodología

#### Subtarea 5.1: Instalación y configuración PNP4Nagios

Descargamos desde su página oficial, descomprimir e instalar. Editamos nagios.cfg para añadir distintas rutas. También editamos el archivo commands.cfg para definir los comandos de pnp4nagios. En templates.cfg definimos un host y un servicio para permitir que nagios haga gráficas de los hosts/servicios que queramos.

Me salieron diferentes errores. El primer error decía que la función set\_magic\_quotes\_runtime está obsoleta. Se solucionaba fácilmente añadiendo en el archivo input.php function\_exists

El segundo error decía que el parámetro sizeof debía estar en la array de un objeto, se soluciona editando el archivo data.php de la siguiente forma.

También salió error de librerías por la falta de php-xml. Se puede solucionar instalándolo con un apt install normal.

```
public function getFirstPage(){
    $pages = $this->getPages();
    if(sizeof($pages) > 0){
        return urldecode($pages[0]);
    }else{
        return FALSE;
    }
}
```

Y cambiarla de esta forma:

```
public function getFirstPage(){
    $pages = $this->getPages();
    if (is_array($pages) && sizeof($pages) > 0){
        return urldecode($pages[0]);
    }else{
        return FALSE;
    }
}
```

#### Subtarea 5.2: Instalación y configuración Grafana

Descargamos Grafana desde su página oficial, instalarla con dpkg y habilitar el servicio. Habilitar puerto 3000 que es el que Grafana utiliza, instalar los paquetes de fuentes necesarios y los plugins de Grafana.



En el directorio pnp4nagios descargamos la api. Reiniciar servicio y con eso ya está disponible.

Para añadir datasource lo hacemos desde la interfaz de Grafana añadiendo el link de pnp4nagios.

Una vez añadido podemos crear dashboards con los dispositivos y servicios que elijamos y así personalizar las gráficas.



### Subtarea 5.3: Instalación y configuración NagVis

Para instalar Nagvis y mk-livestatus debemos descargar la última versión desde su página oficial. Una vez descargado se descomprime e instala. Lo siguiente es modificar el archivo nagios.cfg y añadir la ruta adecuada de livestatus.

Ahora editamos el archivo nagvis.conf y añadimos

```
GNU nano 5.4 /etc/apache2/conf-available/nagvis.conf
# NagVis Apache2 sample configuration file
#
# #####
Alias /nagvis "/usr/share/nagvis/share/"
<Directory "/usr/share/nagvis/share/">
  Options FollowSymLinks
  AllowOverride None
</Directory>
<IfModule mod_authz_core.c>
  # Apache >= 2.4
  Require all granted
</IfModule>
<IfModule !mod_authz_core.c>
  # Apache < 2.4
  Order allow,deny
  Allow from all
</IfModule>

# To enable Nagios basic auth on NagVis use the following options
# Just uncomment it. Maybe you need to adjust the path to the
# Auth user file.
#
# If you use the NagVis internal auth mechanism based on the web
# for you won't need this.
#
AuthName "NagVis Access"
AuthType Basic
AuthUserFile /usr/local/nagios/etc/htpasswd.users
Require valid-user
```

Lo siguiente es editar el archivo `nagvis.ini.php` y descomentar las siguientes líneas

```
GNU nano 5.4 /usr/local/nagvis/etc/nagvis.ini.php
; create users in the database when provided users does not exist. The option
; logonenvcreatorole tells the module to assign the new user to a specific role
; set to empty string to disable that behaviour.

; LogonMultisite: This module uses the authentication provided by auth.* cookies
; which have been generated by Check_MK multisite when using the cookie based
; authentication. Since 1.2.1i2 Check_MK uses a new cookie format. To be able
; to use this, you need to define a new option called logon_multisite_serials
; which points to the auth.serial file generated by Check_MK.
; Special options for this module:

; logon_multisite_htpasswd="/path/to/htpasswd"
; logon_multisite_serials="/path/to/auth.serials"
; logon_multisite_secret="/path/to/auth.secret"
; logon_multisite_createuser="1"
; logon_multisite_creatorole="Guests"

; logonmodule="LogonMixed"
; logonenvvar="REMOTE_USER"
; logonenvcreateuser="1"
; logonenvcreatorole="Guests"

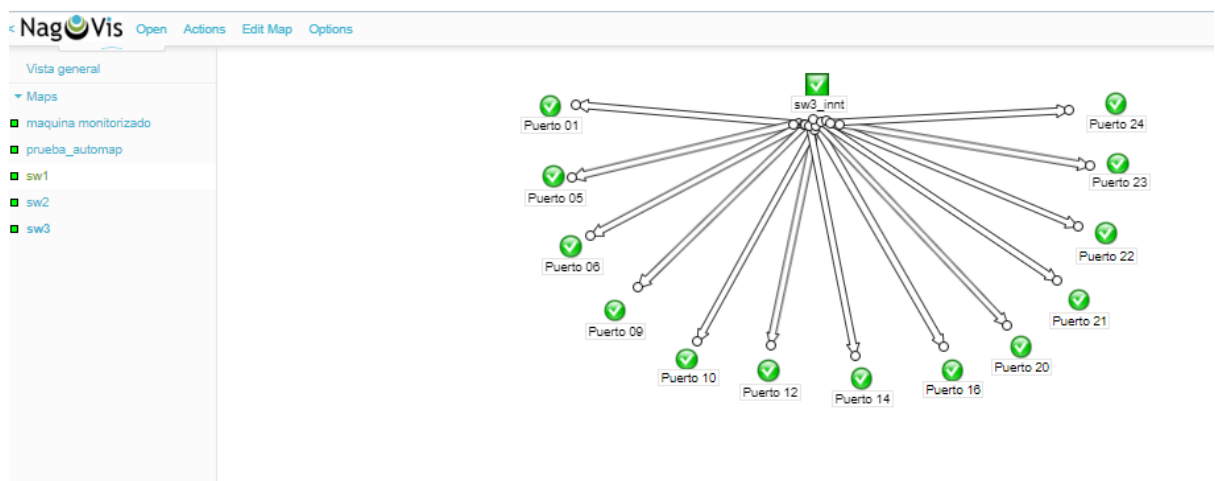
; Enable/Disable access to permitted monitoring objects only. This is useful if you want
; to restrict the access to status information by NagVis AND by the authorization mechanism
; of the monitoring software (e.g. Nagios).
```

Si nos aparece un error por no encontrar la ruta `/var/lib/nagios3/rw/live`

Editar el archivo `nano /etc/nagvis/nagvis.ini.php`

Buscar `socket:unix` y sustituir la ruta por la ruta del live, en mi caso  
`/usr/local/nagios/var/rw/live`

Una vez implementado podemos utilizar la opción de `automap` para que nagvis se encargue de crear un mapa automático con los dispositivos creados o añadir los iconos de servicios y equipos que prefiramos para así hacer un mapa personalizado.



	Duración (horas)	Fecha inicio	Fecha fin
Tarea 5			
Subtarea 5.1	2 + 6 extras	Miércoles 25 de Mayo	Sábado 28 de Mayo
Subtarea 5.2	1	Viernes 27 de Mayo	Viernes 27 de Mayo
Subtarea 5.3	2 +3 extras	Sábado 28 de Mayo	Martes 31 de Mayo

## Tarea 6: Investigación y Documentación

### Objetivo y metodología

Búsqueda en internet y documentación antes de comenzar una tarea y si surge algún problema buscar posibles soluciones además de probarlas y anotarlas.

	Duración (horas)	Fecha inicio	Fecha fin
Tarea 6	15 + 1 extra	Lunes 9 de Mayo	Martes 31 de Mayo

## RECURSOS MATERIALES

Para realizar el proyecto necesitamos crear un servidor Nagios en una máquina con los requisitos 2GB de RAM, 20GB de almacenamiento de disco y 2 núcleos. En esta máquina instalaremos el sistema operativo gratuito Debian11.

Tendremos que añadir esta máquina a la misma red que los dispositivos que queramos monitorizar. Podemos monitorizar tanto el almacenamiento como el uso de CPU en equipos y servidores con todo tipo de sistemas operativos, solamente necesitaremos acceso como administrador para poder crear y ejecutar diferentes comandos.

También necesitaremos tener acceso tanto al Switch y Router que queramos monitorizar, ya que se tendrá que activar SNMP si no estuviera habilitado. Además necesitamos saber el modelo y fabricante de los dispositivos de red con el fin de descargar .mib específicos para su correcta monitorización. El fabricante del Switch a monitorizar es D'Link y el de el Router es Ubiquiti.

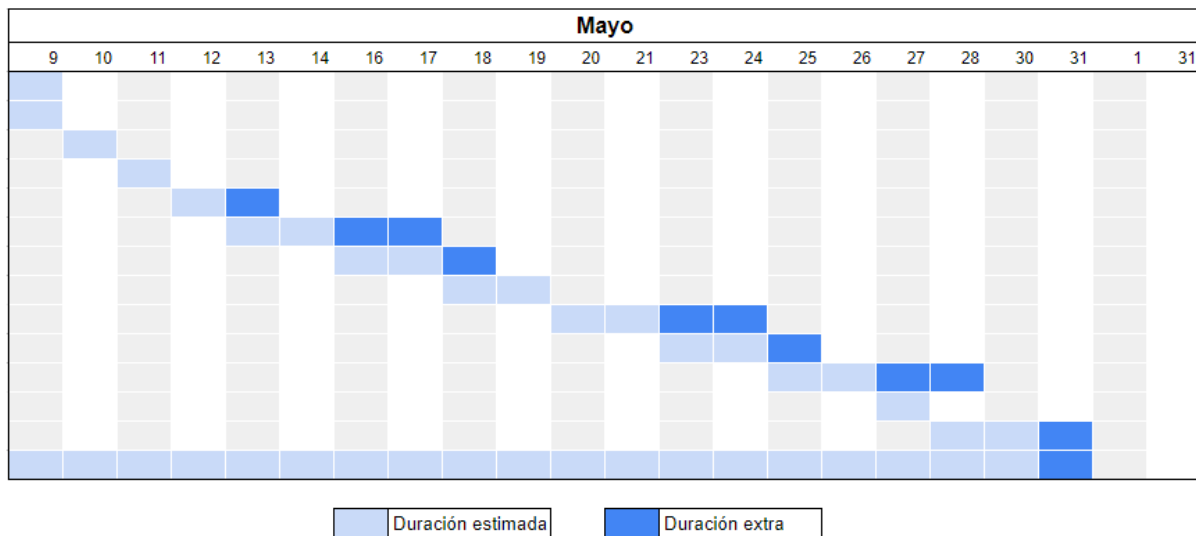
## RECURSOS HUMANOS

Se han necesitado en torno a 70 horas para la realización del proyecto. Debido a la escasez de conocimientos sobre Nagios y su configuración se ha necesitado hacer una previa investigación en cada subtarea del proyecto además de las incompatibilidades entre las diferentes versiones de las herramientas y plugins.

Actividad		Nombre	Estimación	Horas Reales
Tarea 1	Subtarea 1.1	Instalación Debian11	1	1
	Subtarea 1.2	Instalación Nagios	2	2
Tarea 2	Subtarea 2.1	Notificaciones correo electrónico	2	2
	Subtarea 2.2	Notificaciones aplicación oficial	2	2
	Subtarea 2.3	Notificaciones Telegram	2	5
Tarea 3	Subtarea 3.1	Instalación NRPE y configuración	2	8
	Subtarea 3.2	Configuración NRPE en equipos	2	3
Tarea 4	Subtarea 4.1	Instalación SNMP	2	2
	Subtarea 4.2	Configuración Switch	3	10
	Subtarea 4.3	Configuración Access Point	2	5
Tarea 5	Subtarea 5.1	Instalación y configuración PNP4Nagios	2	8
	Subtarea 5.2	Instalación y configuración Grafana	1	1
	Subtarea 5.3	Instalación y configuración Nagvis	2	5
Tarea 6		Investigación y Documentación	15	16
TOTAL:			40	70

## CRONOGRAMA

En un primer momento pensaba realizar todas las tareas programadas en 40h, pero con los diferentes problemas he necesitado más tiempo. La mayor parte del tiempo lo he usado en solucionar errores e investigar el funcionamiento de las distintas herramientas para implementarlas correctamente.



## BIBLIOGRAFÍA

- <https://www.north-networks.com/que-es-nagios/>
- <https://es.wikipedia.org/wiki/Nagios>
- <https://es.joecomp.com/how-install-configure-nagios-ubuntu-18>
- [https://www.ionos.es/digitalguide/servidores/herramientas/plugins-para-nagios-\\_instalacion-de-extensiones/](https://www.ionos.es/digitalguide/servidores/herramientas/plugins-para-nagios-_instalacion-de-extensiones/)
- <https://www.manageengine.com/es/network-monitoring/what-is-snmp.html>
- <https://cloud-infra.engineer/nagios-notifications-via-telegram/>
- <https://www.booleanworld.com/guide-monitoring-servers-nagios/>
- [https://exchange.nagios.org/directory/Plugins/Network-Connections%2C-Stats-and-Bandwidth/check\\_iftraffic64/details](https://exchange.nagios.org/directory/Plugins/Network-Connections%2C-Stats-and-Bandwidth/check_iftraffic64/details)
- <https://exchange.nagios.org/directory/Addons/Notifications/Telegram-Notifications/details>
- <https://www.bujarra.com/nagios-monitorizando-nrpe/>
- <https://assets.nagios.com/downloads/nagioscore/docs/nagioscore/4/en/monitoring-routers.html>
- <https://www.ionos.es/digitalguide/servidores/know-how/snmp/#:~:text=SNMP%20trap%3A%20las%20SNMP%20traps.pueden%20distinguirse%20de%20dos%20maneras.>
- <https://elfreneticoinformatico.com/tarabajando-con-mibs-introduccion-snmp-interpretacion-de-oids/>
- <https://es.mailjet.com/blog/news/servidor-smtp/>
- <https://www.sysadminsdecuba.com/2017/10/instalacion-de-nagvis-y-su-integracion-al-sistema-de-monitoreo-centreon-2-8-en-debian-8/#:~:text=apt-get%20install%20php5-sqlite,Descargamos%20Nagvis.&text=Reiniciamos%20el%20servicio%20de%20apache.&text=Movemos%20la%20configuración%20del%20nagvis,apache2%2Fconf-available%2F.&text=Teniendo%20en%20cuenta%20la%20configuración.reiniciamos%20el%20servicio%20de%20apache.>
- <https://howtoforge.es/como-instalar-la-herramienta-de-monitorizacion-grafana-en-ubuntu-18-04-lts/>
- <https://nksistemas.com/instalacion-de-pnp4nagios-en-nagios-con-centos/>

## ANEXO I

Comandos más usados:

Añadir usuarios a grupos

```
sudo usermod -a -G “grupo” “usuario”
```

Instalar repositorios y paquetes

```
apt install “paquete a descargar”
```

Descargar archivos desde web

```
wget “ruta”
```

Descomprimir archivos

```
tar -xzf “nombre archivo comprimido”
```

Habilitar puertos

```
sudo ufw allow “nombre puerto”
```

Compilar los datos

```
make all
```

Instalación

```
make install
```

Instalación de Nagios

```
Make install-init
```

```
Make install-commandmode
```

```
Make install-config
```

```
make install-webconf
```

Habilitar servicios

```
sudo systemctl enable “servicio”
```

Iniciar servicios

**sudo systemctl start “servicio”**

Restaurar servicios

**sudo systemctl restart “servicio”**

Habilitar el inicio de Nagios automático al arrancar el sistema

**sudo systemctl enable nagios, apache2**

Instalar módulos Perl

**cpan “nombre módulo”**

Crear archivos

**sudo nano**

**sudo touch**

Dar permisos de ejecución a archivos

**sudo chmod a+x “nombre archivo”**

Cambiar propietario archivos

**sudo chown “usuario”.”grupo”**



## ANEXO II

Ruta y contenido de los archivos de distintos archivos de configuración:

### **/opt/smtp-cli-master/config.json**

```
{
  "host": "host por defecto de snmp"
  "username": "correo que recibe la notificación"
  "password": "contraseña de la cuenta"
  "secure": "tls",
  "port": 587
}
```

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

```
define contact {
contact_name           "nombre del contacto"
pager                  "id del usuario de telegram"
email                  "correo que recibe la notificación"
service_notification_commands  notify-service-by-telegram,
                                notify-service-by-email
host_notification_commands    notify-host-by-telegram,
                                notify-host-by-email
host_notifications_enables    "si queremos habilitarlas(1) si no (0)"
service_notifications_enabled  "si queremos habilitarlas(1) si no (0)"
service_notification_options  "a elección (w,c,u,r,f,s)"
host_notification_options     "a elección (d,u,r,f,s,n)"
host_notification_period      "constantemente (24x7)"
service_notification_period    "constantemente (24x7)"
```

```

}
define contactgroup {
    contactgroup_name    "nombre del grupo de contacto"
    alias                "alias"
    members              "miembros del grupo"
}

```

### **/usr/local/nagios/etc/object/archivo\_conf\_switch**

```

define host {
    use    generic-switch, host pnp (generico y pnp4nagios)
    host_name    "nombre host"
    address    "dirección ip switch"
}

define service {
    use                generic-service, srv-pnp (genérico y pnp4nagios)
    host_name    "nombre host"
    service_description    "descripción que saldrá en nagios"
    check_command    check_iftraffic!1!40!60 "comando del plugin 1
                                                indica el puerto"
    contact_group    "grupo de contacto"
}

```

### **/usr/local/nagios/etc/object/archivo\_conf\_accesspoint**

Mismo formato que el archivo anterior pero modificando la ip. En vez de monitorizar por puerto monitorizamos interfaces (eth1, eth0 y wifi0) por ejemplo.

```
check_iftraffic!eth1!40!60
```

### **/usr/local/nagios/etc/servers/testserver.cfg**

```

define host {
    use                linux-server

```

```
host_name      "nombre del servidor remoto"
address        "dirección máquina a monitorizar"
max_check_attempts 5
check_period   24x7
notification_interval 30
notification_period 24x7
}

define service {
    use                generic-service
    host_name          "nombre servidor remoto"
    service_description CPU load
    check_command       check_nrpe!check_load

o
    check_command       check_nrpe!check_sda1
}
```

Archivo de configuración de máquina a monitorizar:

**/usr/local/nagios/etc/nrpe.cfg**

```
command[check_sda1]= /usr/local/nagios/libexec/check_disk -w 20% -c 10% -p
/dev/sda1
```

Manda alerta warning cuando el espacio del disco sda1 sea menor a un 20% y alerta crítica cuando sea menor de un 10%