



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

**Implementación de un Servidor
Nagios para el control y monitoreo
de la Red del GDF**

TESINA

Que para obtener el título de
Ingeniero en Computación

P R E S E N T A

MARTÍNEZ RAMÍREZ SERGIO

Asesor Académico:

Ing. Cruz Sergio Aguilar Díaz



Ciudad Universitaria, Cd. Mx., 2016

Índice

INTRODUCCIÓN.....	5
1. PANORAMA GENERAL.....	6
1.1 Objetivo	6
1.2 Alcance.....	7
1.3 Requerimientos.....	7
2. DESCRIPCIÓN DE LA EMPRESA.....	8
2.1 Datos Generales de la Institución	8
2.2 Características del área.....	9
2.3 Experiencia Laboral	9
2.4 Cursos de Capacitación	10
2.5 Conocimientos	10
3. DESARROLLO DE PROYECTO SERVIDOR NAGIOS.....	11
3.1 Antecedentes	11
3.2 Metodología utilizada	11
3.3 Requerimientos básicos	12
3.4 Implementación y Configuración del Servidor Nagios.....	13
4. RESULTADOS.....	35
4.1 Estados del monitoreo	35
4.2 Notificaciones de problemas.....	38
5. CONCLUSIONES.....	40
REFERENCIAS	41
GLOSARIO.....	45
APENDICE A. CONFIGURACIÓN DE DOS INTERFACES DE RED EN CENTOS	48
APENDICE B. REPORTES POR AVAILABILITY DE NAGIOS	51
APENDICE C. METODOLOGIAS PARA EL DESARROLLO DE PROYECTOS.....	53
APÉNDICE D. INSTALACIÓN DE MRTG EN CENTOS.....	57
APÉNDICE E. INSTALACIÓN DE NRPE	59

ÍNDICE DE FIGURAS

Figura.1.1 Recursos Necesarios.	7
Figura.2.1 Organigrama de la Secretaría de Finanzas.	8
Figura.2.2 Conocimientos.	10
Figura.3.1 Ciclo de vida en prototipo.	11
Figura.3.2 Características recomendadas para Servidor Nagios Core.	12
Figura.3.3 Deshabilitar LINUX.	13
Figura.3.4 Instalación correcta de Nagios.	15
Figura.3.5 Página principal de Nagios.	16
Figura.3.6 Habilitar estadísticas desde el principio.	16
Figura.3.7 Definición del localhost.	17
Figura.3.8 Servicios asignados al localhost.	17
Figura.3.9 Monitoreo del localhost.	18
Figura.3.10 Configuración de cuenta de correo.	19
Figura.3.11 Envío de correo electrónico.	20
Figura.3.12 Comprobación de correo recibido.	20
Figura.3.13 Definiciones para monitorear archivos individuales.	21
Figura.3.14 Definiciones para monitorear directorios.	21
Figura.3.15 Comando para la revisión de puertos TCP.	22
Figura.3.16 Comandos de notificación por correo.	22
Figura.3.17 Comando para determinar el estado de un equipo.	23
Figura.3.18 Definición de un periodo de tiempo laboral.	24
Figura.3.19 Definición de un periodo de tiempo no laboral.	25
Figura.3.20 Definición de un contacto.	27
Figura.3.21 Definición de un grupo de contactos.	27
Figura.3.22 Definición de host.	29
Figura.3.23 Definición de grupo de hosts.	30
Figura.3.24 Definición de servicio.	31
Figura.3.25 Definición de grupo de servicio.	32
Figura.3.26 Uso de la directiva use.	33
Figura.3.27 Uso de SNMP.	33
Figura.3.28 Configuración del tráfico.	33
Figura.3.29 Definición con check_nrpe.	34
Figura.4.1 Estados totales de hosts y servicios.	35
Figura.4.2 Detalle de los hosts y servicios.	36
Figura.4.3 Información detallada de un servicio.	37
Figura.4.4 Reporte por Availability.	38
Figura.A.1 Agregar un adaptador de red.	48
Figura.A.2 Asignación de las conexiones.	49

Servidor Nagios para el control y monitoreo de red del GDF

Figura.A.3 Configuración de conexiones de red.....	49
Figura.A.4 Configuración estática del adaptador.....	50
Figura.A.5 Configuración del proxy en el navegador.....	50
Figura.B.1 Opciones de reportes Nagios.....	51
Figura.B.2 Selección del tipo de reporte.....	51
Figura.B.3 Selección del servicio.....	52
Figura.B.4 Selección del periodo de tiempo.....	52
Figura.B.5 Reporte de un servicio.....	52
Figura.C.1 Modelo por prototipo.....	53
Figura.C.2 Modelo en cascada.....	55
Figura.C.3 Modelo incremental.....	56
Figura.D.1 Gráfica MRTG.....	57
Figura.E.1 Comandos Linux del archivo nrpe.cfg.....	60

INTRODUCCIÓN

En este trabajo se presenta la implementación de un sistema de monitoreo que será implementado dentro de la Secretaría de Finanzas de la Ciudad de México para poder monitorear sus equipos, y que ayude a detectar de forma inmediata el momento en el que ocurre un problema antes de que los usuarios finales noten las fallas.

En el primer capítulo se describe la importancia de implementar un sistema de monitoreo en cualquier organización pública o privada, se menciona el objetivo de realizar este proyecto y lo que se necesita para poder implementarlo correctamente.

En el segundo capítulo se habla un poco de la dependencia donde presté mi servicio social, en este caso la Secretaría de Finanzas, así como una descripción del área a la que pertencí en ese tiempo, además se habla sobre la experiencia profesional que adquirí al participar en la implementación de este proyecto, así como mis conocimientos aplicados.

En el tercer capítulo se describe como fue implementado el sistema de monitoreo, instalaciones y configuraciones necesarias para poner en marcha el sistema, los requerimientos y la metodología utilizada.

En el cuarto capítulo se menciona sobre los resultados que se obtienen después de que se implementó el sistema, se describe la información que muestra el sistema así como las acciones que se toman para notificar que existe un problema.

Finalmente se incluyen algunos manuales de instalación y configuración que son importantes para potencializar el sistema de monitoreo que fue implementado.

1. PANORAMA GENERAL

En la actualidad con el crecimiento de las redes de datos, es fundamental en cualquier tipo de empresa u organización independientemente la actividad a la cual se dedique. Es muy importante contar con una buena infraestructura de red, así como los equipos adecuados para llevar a cabo las actividades de dicha organización.

Para que las organizaciones tengan éxito es necesario que se tenga en buenas condiciones todos los equipos de redes, en muchas ocasiones esto no se toma en cuenta por los involucrados o se cree que tiene una importancia menor. Sin embargo es fundamental detectar cuando un equipo está fallando y es vital saber dónde o qué es lo que está provocando el problema para corregirlo en el menor tiempo posible. En estos casos es necesario destinar recursos para implementar medidas que nos sirvan para detectar cuando la red tiene un problema, para ello se utilizan los sistemas de monitoreo.

Hay que tomar en cuenta que son muchos los dispositivos que se involucran en la transmisión de datos dentro de cualquier red. Se debe tomar en cuenta la capacidad de cada dispositivo para soportar más carga de trabajo, si en algún momento un dispositivo llega a saturarse, es muy probable que se produzcan colapsos de servicios.

Los sistemas de monitoreo nos ayudan a detectar y prevenir problemas en el funcionamiento de la red así como de sus equipos tales como switches, routers, accesspoint, servidores, tráfico, ups, entre otros.

1.1 Objetivo

Implementar un sistema de monitoreo de red que permita detectar los problemas que ocurran en la red y sus equipos con el fin de tener la más alta disponibilidad de los servicios de la organización.

Enviar notificaciones en tiempo real en el momento en que ocurra un incidente crítico o bien cuando se resuelva un problema.

Servidor Nagios para el control y monitoreo de red del GDF

1.2 Alcance

Proporcionar la información del estado de los equipos en tiempo y forma a las áreas correspondientes sobre las Administraciones Tributarias y Centros de Servicio adjuntas a la Secretaría con la finalidad de tener la mayor disponibilidad posible de la red, equipos y servicios.

1.3 Requerimientos

Para el desarrollo de este sistema de monitoreo se utilizará un Servidor Nagios Core, que es la versión libre de Nagios.

Nagios Core está diseñado para ejecutarse en GNU Linux o alguna variante de Unix que cuente con acceso a la red, estas distribuciones deben tener un Kernel 2.4 o superior.

Los requisitos de hardware para la instalación únicamente del Servidor Nagios Core es un procesador de 2GHz+, 1Gb de RAM y 2GB de disco duro, esto nos servirá para monitorear unos cuantos equipos y servicios.

Se debe tener un compilador de C instalado, un Servidor Web (de preferencia Apache), se necesita la librería GD versión 1.6.3 o superior, PHP y SNMP.

Como en la mayoría de los casos, en las organizaciones se cuenta con cientos de equipos a monitorear, por lo que debemos tener un servidor con una mayor cantidad de recursos, a continuación se muestra una figura que lo ejemplifica:

Host*	CoreCPU	CPU Speed	RAM	DiscoDuro
< 100	-	>2GHz	4Gb	40Gb
100- 200	Dual-Core	>3GHz	4Gb	80Gb
200 >	Quad-Core	>3GHz	>4Gb	100Gb

Figura.1.1 Recursos Necesarios.

2. DESCRIPCIÓN DE LA EMPRESA

2.1 Datos Generales de la Institución

La Secretaría de Finanzas tiene como función el desarrollo de las políticas de ingresos y administración tributaria, la programación, recaudación, presupuestación y evaluación del gasto público de la Ciudad de México.

Tiene como objetivo proponer, dirigir la política económica del Gobierno de la Ciudad de México en materia financiera, fiscal, de gasto, de ingreso y deuda pública, con el propósito de fortalecer a la Ciudad de México con crecimiento económico de calidad, equitativo, incluyente y sostenido.

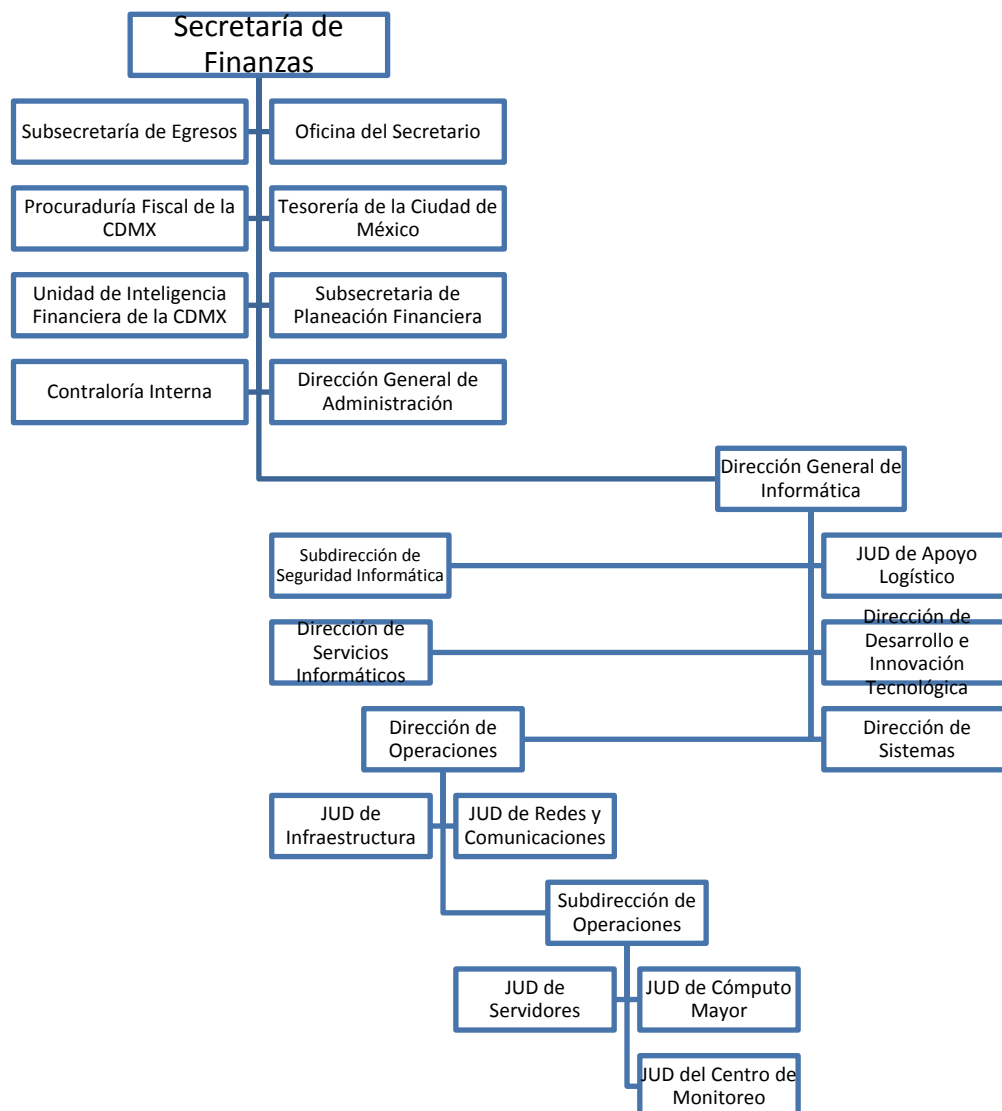


Figura.2.1 Organigrama de la Secretaría de Finanzas.

2.2 Características del área

La Jefatura de Unidad Departamental (JUD) del Centro de Monitoreo tiene la finalidad de monitorear los equipos de red de las administraciones tributarias de la Secretaría de Finanzas para reaccionar a ciertos problemas que ocurran con los dispositivos notificando al área o responsable correspondiente en donde se determinará la acción a tomar a dicho problema.

Las notificaciones que hace la JUD del Centro de Monitoreo se realizan tomando en cuenta cierta tolerancia dependiendo el equipo que presente el problema, esto debido a que en ocasiones se puede tratar de una intermitencia que se encuentran debidamente identificadas.

Cuando ocurre un incidente y se cumple la tolerancia de ese equipo existen diferentes formas hacer la notificación, por llamada telefónica, por correo electrónico o personalmente, se da seguimiento a la incidencia y se realiza un reporte con la descripción que proporciona el área responsable.

Como en toda organización, hay ocasiones en las que se le realizan mantenimiento a los equipos, por lo que es necesario que el área correspondiente le notifique al Centro de Monitoreo los servicios que pudieran verse afectados para no realizar ninguna acción cuando este problema sea reflejado por el sistema de monitoreo.

2.3 Experiencia Laboral

Durante mi estancia en el área del Centro de Monitoreo realicé actividades tales como la configuración de sistemas operativos, principalmente el GNU Linux, utilizando máquinas virtuales para efectuar las pruebas correspondientes antes de hacer la implementación.

Adicionalmente a esto, use algunos protocolos de red y que son aplicados a sistemas de monitoreo, en este caso Nagios. Gran parte de estas actividades las aprendí en un principio dentro de la Facultad, por esta razón creo que el haber realizado mi Servicio Social dentro de esta área puede ser considerado como una experiencia laboral y que es muy importante para mí para mis inicios dentro del ámbito laboral.

2.4 Cursos de Capacitación

- CCNA Routing and Switching.

En este curso de CISCO se abarcaron varios puntos para el uso y configuración de dispositivos de comunicación como lo son Switch, Router, Router inalámbrico, así como el manejo de protocolos y los diferentes tipos de redes de datos, escalamiento de las mismas y sacar el mayor provecho de los dispositivos CISCO y la aplicación correcta de los protocolos.

2.5 Conocimientos

Los conocimientos aprendidos o desarrollados ya sea en el área de redes y que son necesarios para el Centro de Monitoreo son los siguientes:

- Paqueterías Informáticas tales como Word, Excel, PowerPoint, LibreOffice, entre otras.
- Servidores Virtuales en VMware, VirtualBox utilizando diferentes versiones de sistemas operativos.
- Herramientas de desarrollo, MySQL, PHP, HTML, HTML5, CSS.
- PuTTY, WinSCP, UltraISO.
- Servidores Web IIS, Apache.



Figura.2.2 Conocimientos.

3. DESARROLLO DE PROYECTO SERVIDOR NAGIOS

Para el desarrollo de este proyecto fue necesario identificar la problemática que se tenía en la red de la Secretaría de Finanzas, ya que la red se hacía demasiado lenta y por ello los procesos que corrían eran muy lentos. Generalmente una de las causas es el tráfico en la red.

3.1 Antecedentes

La Secretaría de Finanzas cuenta con administraciones tributarias en diferentes lugares de la Ciudad de México, y estas cuentan con equipos que son indispensables para tener comunicación por lo que se requiere que tengan la mayor disponibilidad posible.

Además, también existen servidores para diferentes propósitos y que son muy importantes para el funcionamiento de la Secretaría, por lo que de la misma forma que la red, se necesita que estos servidores tengan una gran disponibilidad.

Cuando ocurre un problema y se cuenta con muchos equipos es muy complicado saber qué es lo que está pasando, esto implica pérdida de tiempo y a la vez, al ser una organización que recauda dinero, se tienen pérdidas económicas fuertes si no se soluciona el problema en el menor tiempo posible.

3.2 Metodología utilizada

La metodología que se aplicó en el desarrollo de este proyecto fue una metodología en prototipo, ya que con el listado rápido de los requisitos se procede a instalar y el siguiente proceso aplicar prueba y error según las necesidades que vayan surgiendo a lo largo del tiempo.

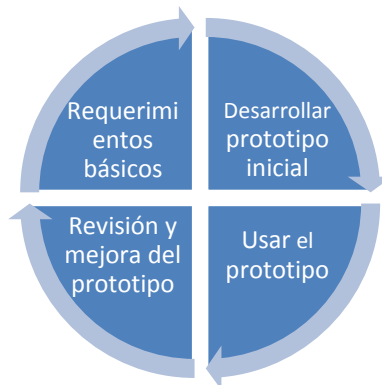


Figura.3.1 Ciclo de vida en prototipo.

3.3 Requerimientos básicos

La Secretaría de Finanzas necesita tener una alta disponibilidad de la gran mayoría de sus equipos, algunos de ellos pueden tener cierta tolerancia a fallos, pero otros deben cumplir con el 99% de funcionamiento, por esto, es necesario que se cuente con algún sistema que nos permita detectar si algún equipo está fallando, si existe un excedente de tráfico en algún enlace, entre otro tipo de problemas.

Existen alrededor de 400 equipos tales como switches, routers, ups, accesspoint, servidores, entre otros más. Adicionalmente se requiere que a estos equipos se les monitoree cierto servicio en particular y que no necesariamente es el mismo para todos ellos, por poner algunos ejemplos:

- La disponibilidad. Este caso aplica para el 100% de los equipos, en el que se verifica que exista comunicación con el equipo.
- Memoria. Se verifica la cantidad de memoria del equipo usada o libre según sea el caso.
- Disco Duro. Se verifica la cantidad de espacio en disco del equipo, en algunos casos existen equipos con más de un disco duro.
- Cantidad de Tráfico. Se verifica la cantidad de tráfico que pasa por cierto enlace en un Switch o Router.
- Verificar Puertos. Se verifica que un puerto específico esté abierto en un servidor.

El monitoreo de equipos y servicios se hace a petición de las áreas interesadas, y se consideró la posibilidad de que incremente la cantidad de equipos a monitorear.

Nagios



192.168.100.1

- **Sistema Operativo CentOS 6 ó sup.**
- **Intel Core Quad 2.66 Ghz.**
- **8GB RAM**
- **500GB de disco duro**

Figura.3.2 Características recomendadas para Servidor Nagios Core.

Servidor Nagios para el control y monitoreo de red del GDF

Para poder realizar el monitoreo de los equipos necesitamos principalmente el sistema Nagios, que se instala en un servidor Linux. Además, se utilizarán algunas otras herramientas que en conjunto con Nagios servirán para tener un mayor alcance del monitoreo.

3.4 Implementación y Configuración del Servidor Nagios

Para la implementación y configuración de Servidor de Monitoreo realizaremos los siguientes pasos:

Paso 1. Instalación del Sistema Operativo

El servidor que fue proporcionado por el área correspondiente cuenta con el Sistema Operativo GNU Linux CentOS 6.7 previamente instalado y configurado, por lo cual el trabajo realizado, es únicamente el Monitoreo de los equipos y red.

Paso 2. Instalación de Nagios

Para la instalación de Nagios es necesario conocer la versión del SO, los requerimientos necesarios como un compilador de C, tener instalado un Servidor Apache, librería GD versión 1.6.3 o superior, PHP y SNMP. Antes que se inicie con la instalación de estas paqueterías se tiene que deshabilitar SELINUX. Para ello se realizó lo siguiente:

Del archivo `/etc/sysconfig/selinux` se cambió la opción de “SELINUX=enforcing” a “SELINUX=disabled”,

```
# This file controls the state of SELinux on the system.
# SELINUX= can take one of these three values:
#   enforcing - SELinux security policy is enforced.
#   permissive - SELinux prints warnings instead of enforcing.
#   disabled - SELinux is fully disabled.
SELINUX=disabled
# SELINUXTYPE= type of policy in use. Possible values are:
#   targeted - Only targeted network daemons are protected.
#   strict - Full SELinux protection.
SELINUXTYPE=targeted
```

Figura.3.3 Deshabilitar LINUX.

Reiniciamos y ahora procedemos a instalar lo necesario, para ello se instaló lo siguiente:

Servidor Nagios para el control y monitoreo de red del GDF

```
# yum install -y httpd php gcc glibc glibc-common gd gd-devel make net-  
snmp net-snmp-utils
```

El siguiente paso es descargar Nagios Core y los plugins desde su página oficial, esto lo podemos realizar con lo siguiente:

```
# wget http://prdownloads.sourceforge.net/sourceforge/nagios/nagios-  
3.5.0.tar.gz
```

```
# wget https://www.nagios-plugins.org/download/nagios-plugins-1.5.tar.gz
```

Se deben crear los usuarios necesarios:

```
# useradd nagios  
# groupadd nagcmd  
# usermod -a -G nagcmd nagios  
# usermod -a -G nagcmd apache
```

Extraemos los paquetes descargados e ingresamos al directorio “nagios” que se creó, una vez dentro del directorio debemos compilar e instalar Nagios.

```
# tar -zxvf nagios-3.5.0.tar.gz  
# tar -zxvf nagios-plugins-1.5.tar.gz  
  
# ./configure --with-command-group=nagcmd  
# make all  
# make install  
# make install-init  
# make install-config  
# make install-commandmode  
# make install-webconf  
# cp -r contrib/eventhandlers/ /usr/local/nagios/libexec  
# /usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg
```

Si todo fue instalado de forma correcta, se nos indicará no hubo ningún error.

```
Running pre-flight check on configuration data...

Checking objects...
  Checked 42 services.
  Checked 12 hosts.
  Checked 3 host groups.
  Checked 0 service groups.
  Checked 1 contacts.
  Checked 1 contact groups.
  Checked 25 commands.
  Checked 5 time periods.
  Checked 0 host escalations.
  Checked 0 service escalations.
Checking for circular paths...
  Checked 12 hosts
  Checked 0 service dependencies
  Checked 0 host dependencies
  Checked 5 timeperiods
Checking global event handlers...
Checking obsessive compulsive processor commands...
Checking misc settings...

Total Warnings: 0
Total Errors: 0
```

Figura.3.4 Instalación correcta de Nagios.

Iniciamos los procesos Apache, Nagios, SNMP y creamos el usuario con el que se ingresará a la interfaz gráfica.

```
# service httpd start
# service nagios start
# service snmpd start
# htpasswd -c /usr/local/nagios/etc/htpasswd.users nagiosadmin
```

Salimos del directorio “nagios” e ingresamos al directorio “nagios-plugins-1.5” y procedemos a instalar los plugins.

```
# ./configure --with-nagios-user=nagios --with-nagios-group=nagios
# make
# make install
```

Realizamos las configuraciones finales y podemos ingresar al modo gráfico de Nagios usando un explorador, para ello debemos usar la IP del servidor donde fue instalado, es necesario deshabilitar el firewall del servidor para poder ingresar:

Servidor Nagios para el control y monitoreo de red del GDF

```
# chkconfig --add nagios
# chkconfig nagios on
# chkconfig --add httpd
# chkconfig httpd on
# chkconfig snmpd on
```

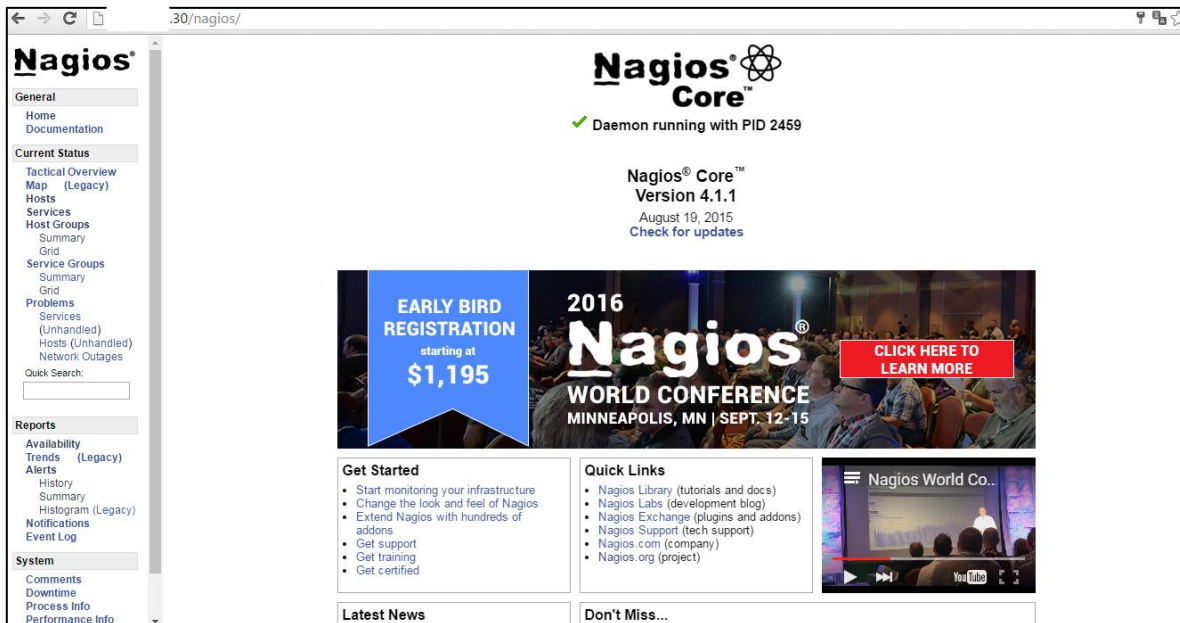


Figura.3.5 Página principal de Nagios.

Una vez instalado Nagios, es necesario tener estadísticas de todos los equipos y servicios que se van monitorear desde el primer momento en el que se ingresan al sistema, por lo que se debe asignar la opción “*log_initial_states*” con valor de uno (1). Esta opción se encuentra en el archivo “*/usr/local/nagios/etc/nagios.cfg*”

```
# INITIAL STATES LOGGING OPTION
# If you want Nagios to log all initial host and service states to
# the main log file (the first time the service or host is checked)
# you can enable this option by setting this value to 1. If you
# are not using an external application that does long term state
# statistics reporting, you do not need to enable this option. In
# this case, set the value to 0.

log_initial_states=1
```

Figura.3.6 Habilitar estadísticas desde el principio.

Cada vez que realizamos una nueva configuración es necesario reiniciar el proceso Nagios para que los cambios tengan efecto.

Servidor Nagios para el control y monitoreo de red del GDF

```
# service nagios restart
```

Por defecto Nagios crea el equipo “localhost” con algunos servicios a monitorearse, este archivo lo encontramos en “/usr/local/nagios/etc/objects/localhost.cfg”

```
# Define a host for the local machine
define host{
    use                linux-server

    host_name          localhost
    alias              localhost
    address            127.0.0.1
}
```

Figura.3.7 Definición del localhost.

Y además encontramos unos ejemplos como el Ping y la Capacidad del Disco que se monitorean y están asignados al localhost.

```
# Define a service to "ping" the local machine
define service{
    use                local-service
    host_name          localhost
    service_description PING
    check_command      check_ping!100.0,20%!500.0,60%
}

# Define a service to check the disk space of the root partition
# on the local machine. warning if < 20% free, critical if
# < 10% free space on partition.
define service{
    use                local-service
    host_name          localhost
    service_description Root Partition
    check_command      check_local_disk!20%!10%!/
}
```

Figura.3.8 Servicios asignados al localhost.

Es recomendable crear un archivo de configuración por cada equipo a monitorear, esto porque será más fácil de administrar y realizar modificaciones futuras. Cada archivo debe tener la extensión *.cfg*.

Cuando ya instaló lo anterior mencionado podemos encontrar por defecto que se está monitoreando el localhost, es decir, se está monitoreando el mismo servidor, y adicionalmente tiene algunos servicios como lo son:

Servidor Nagios para el control y monitoreo de red del GDF

- Current Load: Indica la carga media del procesador.
- Current Users: Indica la cantidad de usuarios conectados al servidor.
- HTTP: Indica el estado del servidor web.
- PING: Indica la cantidad de paquetes perdidos y el tiempo de respuesta.
- Root Partition: Indica la cantidad de espacio libre en disco.
- SSH: Indica el estado del SSH.
- Swap Usage: Indica la cantidad de Swap libre.
- Total Processes: Indica el total de procesos que se están ejecutando.

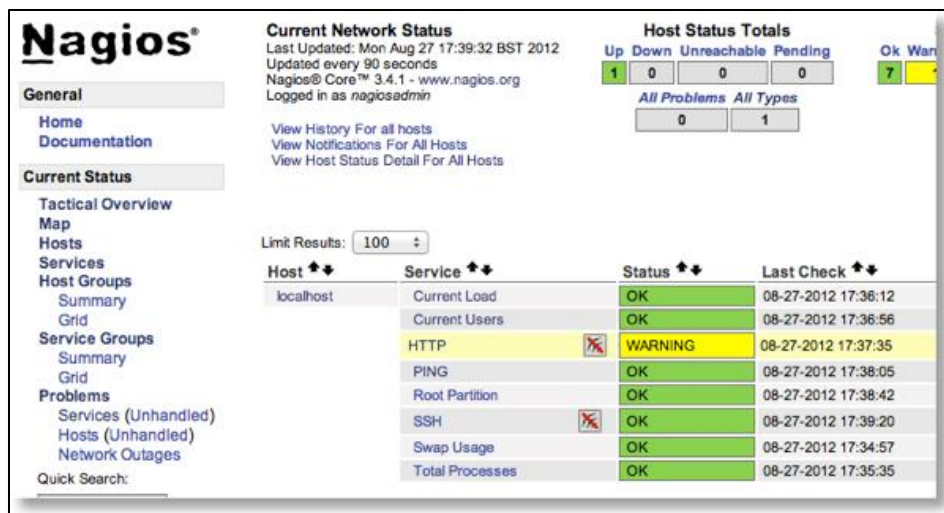


Figura.3.9 Monitoreo del localhost.

Lo que sigue es configurar cada cuando se realizará un chequeo del equipo o servicio y el número de intentos que se realizarán antes de enviar una notificación por correo electrónico, pero para ello se tiene que realizar la configuración del correo electrónico en el mismo servidor.

Paso 3. Instalación y configuración de Postfix

Para poder enviar correo necesitamos un servidor SMTP, en este caso se debe utilizar el servidor SMTP con el que trabaja la Secretaría.

El primer paso es instalar los paquetes necesarios para el funcionamiento del correo electrónico.

Servidor Nagios para el control y monitoreo de red del GDF

```
# yum install postfix mailx cyrus-sasl-plain
```

Es necesario tener una cuenta del servidor de correo; a continuación se debe utilizar la cuenta y contraseña para realizar la siguiente configuración:

Dentro del directorio `/etc/postfix/` hay que crear un archivo llamado `sasl_passwd`

Hay que abrir el archivo creado y escribir lo siguiente:

```
smtp.server.com cuentaSMTP:passwordSMTP
```

Dónde:

smtp.server.com: es la dirección SMTP del servidor de correo.

cuentaSMTP: es la cuenta de correo

passwordSMTP: es la contraseña de la cuenta de correo en texto claro



```
smtp.server.com e [redacted]@gr[redacted].com:30 [redacted]
```

Figura.3.10 Configuración de cuenta de correo.

Como no es seguro tener la cuenta de correo en texto claro se debe utilizar el siguiente comando:

```
# postmap hash:/etc/postfix/sasl_passwd
```

Una vez realizado lo anterior se creará un archivo en el mismo directorio llamado `sasl_passwd.db`, con esto, ya podemos eliminar el archivo que creamos `sasl_passwd`. Ahora dentro del archivo `/etc/postfix/main.cf` hay que escribir al final del mismo las siguientes instrucciones:

```
smtp_sasl_auth_enable = yes
smtp_sasl_password_maps = hash:/etc/postfix/sasl_passwd
smtp_sasl_security_options = noanonymous
smtp_tls_security_level = secure
smtp_tls_mandatory_protocols = TLSv1
smtp_tls_mandatory_ciphers = high
smtp_tls_secure_cert_match = nexthop
smtp_tls_CAfile = /etc/pki/tls/certs/ca-bundle.crt
```

Servidor Nagios para el control y monitoreo de red del GDF

```
relayhost = smtp.server.com:puerto
```

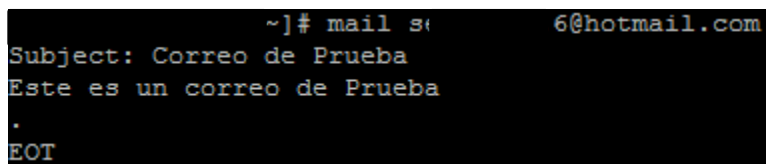
La última línea está compuesta por la dirección del servidor de correo y el puerto utilizado.

Se debe de reiniciar el proceso postfix.

```
# service postfix restart
```

Con esta configuración ya podemos enviar correo a cualquier cuenta existente, podemos realizar una prueba con la siguiente instrucción:

```
# mail cuenta@servidor.com
```



```
~]# mail s: 6@hotmail.com
Subject: Correo de Prueba
Este es un correo de Prueba
.
EOT
```

Figura.3.11 Envío de correo electrónico.

Podemos consultar que el correo se recibió correctamente.

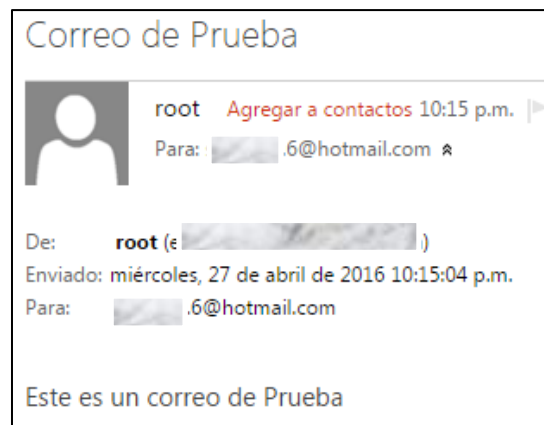


Figura.3.12 Comprobación de correo recibido.

Paso 4. Configuración de Nagios

Una vez instalado Nagios debemos realizar las configuraciones necesarias para poder monitorear equipos y servicios. Con la instalación de Nagios, se creó el directorio `/usr/local/nagios/etc/`, en este se encuentran varios archivos que son indispensables para configuración correcta de Nagios y otros que serán de ayuda.

Servidor Nagios para el control y monitoreo de red del GDF

- **Archivo de configuración principal**

El archivo `nagios.cfg`, ubicado en el directorio `/usr/local/nagios/etc/` es el archivo principal de Nagios y contiene la configuración que afecta directamente al funcionamiento de Nagios. Como configuración básica, en este archivo se debe definir la ubicación de todos los archivos que se vayan agregando para el monitoreo, una de las formas que podemos hacer es indicar la ubicación de un archivo en específico utilizando la directiva `cfg_file=ruta-del-archivo.cfg`.

```
cfg_file=/usr/local/nagios/etc/objects/switch_A.cfg
cfg_file=/usr/local/nagios/etc/objects/servidor_1.cfg
cfg_file=/usr/local/nagios/etc/objects/servidor_2.cfg
cfg_file=/usr/local/nagios/etc/objects/windows_A.cfg
```

Figura.3.13 Definiciones para monitorear archivos individuales.

Esta forma de especificar los archivos se vuelve compleja cuando el número de archivos incrementa, por lo que otra manera más sencilla es indicar un directorio en donde se ubican los archivos a monitorear, esto se realiza con la directiva `cfg_dir=ruta-del-directorio`.

```
cfg_dir=/usr/local/nagios/etc/servers
cfg_dir=/usr/local/nagios/etc/printers
cfg_dir=/usr/local/nagios/etc/switches
cfg_dir=/usr/local/nagios/etc/routers
```

Figura.3.14 Definiciones para monitorear directorios.

- **Archivo `cgi.cfg`**

En este archivo se definen las directivas necesarias para el funcionamiento del CGI. Un CGI es una tecnología de la World Wide Web que permite a un cliente, un navegador web, solicitar datos de un programa que se está ejecutando en un servidor web.

En este archivo se define la ubicación del archivo de configuración principal de Nagios, en este caso es `nagios.cfg`.

- **Definición de los comandos**

Los comandos son los que usa Nagios para realizar los chequeos o envío de notificaciones por medio de los plugins ubicados en el directorio `/usr/local/nagios/libexec/`. Un archivo de gran ayuda es `commands.cfg`, en este

Servidor Nagios para el control y monitoreo de red del GDF

archivo ubicado en `/usr/local/nagios/etc/objects/` se encuentran todas las definiciones de comandos necesarios que sirven como referencia para realizar los chequeos de cada equipo o servicio y así evitar escribir todo el comando completo cada que se quiera agregar un nuevo equipo o servicio a monitorear.

El formato de la definición de un comando sigue el siguiente patrón:

```
define command{
    command_name      nombre_comando
    command_line      comando
}
```

Dónde:

nombre_comando: es el nombre que se le asignará a la definición.

comando: es en sí, el comando completo que se utilizará para realizar un chequeo.

Ejemplo de una definición de comando:

```
# 'check_udp' command definition
define command{
    command_name      check_udp
    command_line      $USER1$/check_udp -H $HOSTADDRESS$ -p $ARG1$
}
```

Figura.3.15 Comando para la revisión de puertos TCP.

En este archivo también encontramos la referencia del comando que utilizará Nagios para enviar un correo cuando ocurra una incidencia de un equipo o servicio, estos comandos pueden ser modificados para justarlos a la información que se necesita ser enviada en caso de algún incidente. En la siguiente figura se muestra parte de los comandos utilizados para el correo.

```
'notify-host-by-email' command definition
define command{
    command_name      notify-host-by-email
    command_line      /usr/bin/printf "%b" "***** Nagios *****\n\nNotification Type:
}

'notify-service-by-email' command definition
define command{
    command_name      notify-service-by-email
    command_line      /usr/bin/printf "%b" "***** Nagios *****\n\nNotification Type:
}
```

Figura.3.16 Comandos de notificación por correo.

Servidor Nagios para el control y monitoreo de red del GDF

Otro comando importante es el que se utiliza cuando se ingresa un nuevo equipo a monitorear y que servirá para determinar si el equipo está en estado UP o en estado DOWN, normalmente se utiliza el ping para este tipo de casos y en el archivo `commands.cfg` se encuentra la definición utilizada como se muestra en la siguiente figura.

```
# 'check-host-alive' command definition
define command{
    command_name    check-host-alive
    command_line    $USER1$/check_ping -H $HOSTADDRESS$ -w 3000.0,80% -c 5000.0,100% -p 5
}
```

Figura.3.17 Comando para determinar el estado de un equipo.

En el ejemplo de la figura 3.17 se enviarán 5 ping al host y si el tiempo de respuesta es mayor a 5000ms o bien se pierden el 100% de los paquetes el equipo se colocará en estado DOWN, de lo contrario estará en modo UP.

Los siguientes comandos están definidos para el monitoreo de servicios, entre ellos podemos encontrar:

- `check_local_disk`: Revisa la cantidad de espacio en disco duro.
- `check_local_load`: Revisa la carga del CPU.
- `check_local_procs`: Revisa la cantidad de procesos que se están ejecutando.
- `check_local_users`: Revisa el número de usuarios conectados al servidor.
- `check_local_swap`: Revisa la cantidad de SWAP
- `check_http`: Revisa el servicio apache
- `check_ssh`: Revisa el protocolo SSH
- `check_tcp`: Revisa un puerto TCP
- `check_udp`: Revisa un puerto UDP

Existen más comandos dentro de este archivo y se pueden agregar más dependiendo de las necesidades. Todos estos comandos tienen la principal característica que mandan llamar un plugin, y para poder ejecutarse correctamente necesitan ciertos parámetros que dependen de cada plugin y que determinarán el estado que tomará cada servicio, en la gran mayoría de los plugins podemos obtener información sobre su uso correcto utilizando el parámetro `-help` ó `-h` desde la línea de comandos, por ejemplo, `/usr/local/nagios/libexec/check_snmp -help`

- **Definición de los periodos de tiempo**

Los periodos de tiempo sirven definir cuándo es válido realizar chequeos o bien enviar notificaciones. Se puede hacer uso del archivo `timeperiods.cfg` ubicado en el directorio `/usr/local/nagios/etc/objects/` contiene la definición de unos periodos de tiempo que servirán para las configuraciones.

El formato de definición de un periodo de tiempo sigue el siguiente patrón:

```
define timeperiod{
    timeperiod_name  nombre_perodo
    alias            alias_perodo
    [weekday]       rango_tiempo
    exclude         nombre_perodo1,...nombre_perodoN
}
```

Dónde:

nombre_perodo: es el nombre que se le asignará al periodo de tiempo. Este campo es obligatorio.

alias: Es una breve descripción del periodo de tiempo que haga referencia a los días que serán asignados. Este campo es obligatorio.

[weekday]: Aquí en donde se introducen todos los días que serán tomados en cuenta por el periodo de tiempo, se puede colocar desde domingo hasta sábado (debe ser en inglés: `sunday...saturday`), y además se puede colocar el horario válido para cada uno de esos días.

exclude: Se puede colocar el nombre de un periodo de tiempo que será excluido de esa definición.

Ejemplo de definición de un periodo de tiempo:

```
define timeperiod{
    timeperiod_name workhours
    alias            Horario de Trabajo
    monday          09:00-17:00
    tuesday         09:00-17:00
    wednesday       09:00-17:00
    thursday        09:00-17:00
    friday          09:00-17:00
}
```

Figura.3.18 Definición de un periodo de tiempo laboral.

Servidor Nagios para el control y monitoreo de red del GDF

No solo podemos asignar días de la semana, también es posible colocar días particulares del año, como por ejemplo días no laborales o algún tipo de celebración.

```
define timeperiod{
    timeperiod_name mex-celebraciones
    alias Celebraciones en Mexico
    january 1 00:00-24:00 ; Año nuevo
    february 5 00:00-24:00 ; Constitucion1917
    march 21 00:00-24:00 ; BenitoJuarez
    may 5 00:00-24:00 ; DiaDelTrabajo
    september 16 00:00-24:00 ; Independencia
    november 2 00:00-24:00 ; DiaMuertos
    november 20 00:00-24:00 ; Revolucion
    december 25 00:00-24:00 ; Navidad
}
```

Figura.3.19 Definición de un periodo de tiempo no laboral.

Con la definición de la Figura 3.18 y excluyendo la definición de la Figura 3.19 se restringen los chequeos y notificaciones únicamente a los días laborales quitando los días festivos. Se pueden realizar muchas formas, ajustándose a las necesidades de cada equipo y servicio.

- **Definición de contactos**

Un contacto nos ayuda a identificar a una persona, el archivo contacts.cfg ubicado en el directorio /usr/local/nagios/etc/objects/ se describen las definiciones de los contactos que servirán para identificar a quién se debe contactar en caso de que exista algún problema.

El formato de definición de un contacto sigue el siguiente patrón:

```
define contact{
    contact_name nombre_contacto
    alias alias_periodo
    host_notifications_enabled [0/1]
    service_notifications_enabled [0/1]
    host_notification_period nombre_timeperiod
    service_notification_period nombre_timeperiod
    host_notification_options [d,u,r,f,n]
    service_notification_options [w,u,c,r,f,n]
    host_notification_commands nombre_comando
    service_notification_commands nombre_comando
    email direccion_email
}
```

Servidor Nagios para el control y monitoreo de red del GDF

Dónde:

contact_name: Es el nombre que se le asignará al contacto. Este campo es obligatorio.

alias: Es una breve descripción para identificar la persona, normalmente se suele escribir el nombre completo de la persona.

host_notifications_enabled: En esta directiva se define si este contacto recibirá notificaciones de cualquier host, si se coloca cero (0) el contacto no recibirá notificaciones de los host, si se coloca en uno (1) el contacto recibirá las notificaciones de los host. Este campo es obligatorio.

service_notifications_enabled: En esta directiva se define si este contacto recibirá notificaciones de cualquier servicio, si se coloca cero (0) el contacto no recibirá notificaciones de los servicios, si se coloca en uno (1) el contacto recibirá las notificaciones de los servicios. Este campo es obligatorio.

host_notification_period: Se asigna el periodo de tiempo en el que se enviarán notificaciones de los host a este contacto, estos periodos de tiempo se definen en el archivo timeperiods.cfg. Este campo es obligatorio.

service_notification_period: Se asigna el periodo de tiempo en el que se enviarán notificaciones de los servicios a este contacto, estos periodos de tiempo se definen en el archivo timeperiods.cfg. Este campo es obligatorio.

host_notification_options: Aquí se definen el tipo de notificaciones de los host que se enviarán al contacto; **d=** notifica host en DOWN, **u=** notifica host en estado UNREACHABLE, **r=** notifica cuando un host está en UP, **f=** notifica cuando un host inicia o termina un “aleteo”, si no se enviará ninguna notificación se debe usar **n**. Este campo es obligatorio.

service_notification_options: Aquí se definen el tipo de notificaciones de los servicios que se enviarán al contacto; **w=** notifica servicios en WARNING, **u=** notifica servicios en UNKNOWN, **c=** notifica servicios en CRITICAL, **r=** notifica cuando un servicio se recupera (estado en OK), **f=** notifica cuando un servicio inicia o termina un “aleteo”, si no se enviará ninguna notificación se debe usar **n**. Este campo es obligatorio.

host_notification_commands: Aquí se asigna el comando que se utilizará para enviar las notificaciones de los host, que puede ser uno que se definió en el archivo commands.cfg. Este campo es obligatorio.

service_notification_commands: Aquí se asigna el comando que se utilizará para enviar las notificaciones de los servicios, que puede ser uno que se definió en el archivo commands.cfg. Este campo es obligatorio.

email: Aquí se escribe la dirección de correo del contacto, que será la que se utilice para enviar las notificaciones.

Ejemplo de la definición de un contacto:

```
define contact{
    contact_name          lbravo
    alias                 Luxx Xxxvo Xxxrez
    host_notifications_enabled 1
    service_notifications_enabled 1
    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-by-email
    host_notification_commands host-notify-by-email
    email                lbravo@server.com
}
```

Figura.3.20 Definición de un contacto.

En ocasiones, existe la necesidad de agrupar varios contactos en un mismo grupo, por lo que también podemos realizar esta función por medio de la definición de un contactgroup.

El formato de definición de un grupo de contacto sigue el siguiente patrón:

```
define contactgroup{
    contactgroup_name    nombre_grupo
    alias                alias_grupo
    members              miembro1,miembro2,...miembroN
    contactgroup_members grupos_contactos
}
```

Dónde:

contactgroup_name: Es el nombre que se le asignará al grupo de contacto. Este campo es obligatorio.

alias: Es una descripción que ayude a identificar el grupo de contacto. Este campo es obligatorio.

members: Es la lista de los contactos que estarán asociados a este grupo.

contactgroup_members: Es la lista de grupos de contactos que estarán asociados a este grupo.

```
define contactgroup{
    contactgroup_name    admins
    alias                Administradores
    members              lbravo,xmart,lisuae
}
```

Figura.3.21 Definición de un grupo de contactos.

- **Definición de los hosts**

Conociendo las funciones básicas de las definiciones anteriormente mencionadas, como lo son los comandos, los periodos de tiempo, los contactos y los grupos de contactos, podemos comenzar a definir los host que deben ser monitoreados, para ello se usa el siguiente patrón, en el cual solo se describen las directivas necesarias para el proyecto:

```
define host{
    host_name           nombre_del_host
    alias               alias_host
    address              direccionIP
    check_command        comando
    max_checks_attempts #intentos
    check_interval      #minutos_estado_normal
    retry_interval      #minutos_incidente
    check_period         periodo_chequeo
    contact_group        grupo_contacto
    notification_interval #minutos_notificaciones
    notification_period  periodo_tiempo_notificaciones
    notification_options notificaciones_validas
    notifications_enabled [0/1]
}
```

Dónde:

host_name: Es el nombre que se le asigna al host. Este campo es obligatorio.

alias: Es una descripción que ayude a identificar el host.

address: Dirección IP del host.

check_command: Es el comando que se utilizará para realizar los chequeos del host.

max_checks_attempts: Es el número de chequeos que se realizarán en un incidente antes de enviar una notificación. Este campo es obligatorio.

check_interval: Es el intervalo de tiempo en el que se realizarán chequeos en un estado normal o bien cuando se ha superado el número de intentos y se mantiene el incidente, el valor asignado será tomado como minutos.

retry_interval: Es el intervalo de tiempo en el que se realizarán chequeos en el momento en que el host pasa de un estado normal a un incidente, cuando se superé el número de intentos los chequeos se harán nuevamente por el tiempo definido en *check_interval*.

check_period: Se define el periodo de tiempo válido para realizar chequeos. Este campo es obligatorio.

Servidor Nagios para el control y monitoreo de red del GDF

contact_group: Se define el grupo de contactos a los que se enviarán las notificaciones. Este campo es obligatorio.

notification_interval: Define el intervalo de tiempo en el que se enviarán notificaciones cuando el host se encuentre en un incidente, el valor asignado será tomando como minutos. Este campo es obligatorio.

notification_period: Define el periodo de tiempo válido para enviar notificaciones. Este campo es obligatorio.

notification_options: Se definen el tipo de notificaciones que serán enviadas, **d**= envía notificaciones de estado DOWN, **r**= envía notificaciones de una recuperación (estado UP), **u**= envía notificaciones UNREACHABLE, **f**= envía notificaciones cuando inicia o termina un aleteo, **n**= no envía ningún tipo de notificación.

notifications_enabled: Se indica si se habilitan o no las notificaciones, uno (1) para habilitar, cero (0) para no habilitarlas.

```
define host{
    host_name          Server_A2
    alias              Servidor web
    address            192.168.1.30
    check_command      check-host-alive
    check_interval     5
    retry_interval     1
    max_check_attempts 6
    check_period       24x7
    contact_groups     admins-servers
    notification_interval 180
    notification_period 24x7
    notification_options d,r
}
```

Figura.3.22 Definición de host.

Es muy común que existan equipos que pueden ser agrupados, como por ejemplo servidores, switches o routers, por ello es necesario definir grupos de host para tener una mejor organización, esto lo realizamos con un hostgroup y queda definido como:

```
define hostgroup{
    hostgroup_name     nombre_grupo
    alias              alias
    members            hosts
}
```

Dónde:

hostgroup_name: Es el nombre que se le asigna al grupo de hosts. Este campo es obligatorio.

Servidor Nagios para el control y monitoreo de red del GDF

alias: Es una descripción que ayude a identificar el grupo de hosts. Este campo es obligatorio.

members: Es la lista de los hosts que pertenecen al grupo, se deben ir separados por comas.

```
define hostgroup{
    hostgroup_name    servers_Ax
    alias             Servidores web
    members           server_A1,server_A2,server_A3
}
```

Figura.3.23 Definición de grupo de hosts.

- **Definición de los servicios.**

Los servicios están asociados a los hosts, y entre ellos podemos tener la carga del CPU, cantidad de procesos, espacio en disco, etc. Los servicios se pueden definir con el siguiente patrón:

```
define service{
    host_name          nombre_del_host
    service_description descripcion
    check_command      comando
    max_checks_attempts #intentos
    check_interval     #minutos_estado_normal
    retry_interval     #minutos_incidente
    check_period       periodo_chequeo
    contact_group      grupo_contacto
    notification_interval #minutos_notificaciones
    notification_period periodo_tiempo_notificaciones
    notification_options notificaciones_validas
    notifications_enabled [0/1]
}
```

Dónde:

host_name: Es el nombre del host al que estará asociado el servicio. Este campo es obligatorio.

service_descripcion: Es el nombre que se le asignará al servicio. Este campo es obligatorio.

check_command: Es el comando que se utilizará para realizar los chequeos del servicio.

max_checks_attempts: Es el número de chequeos que se realizarán en un incidente antes de enviar una notificación. Este campo es obligatorio.

Servidor Nagios para el control y monitoreo de red del GDF

check_interval: Es el intervalo de tiempo en el que se realizarán chequeos en un estado normal o bien cuando se ha superado el número de intentos y se mantiene el incidente, el valor asignado será tomado como minutos. Este campo es obligatorio.

retry_interval: Es el intervalo de tiempo en el que se realizarán chequeos en el momento en que el servicio pasa de un estado normal a un incidente, cuando se superé el número de intentos los chequeos se harán nuevamente por el tiempo definido en *check_interval*. Este campo es obligatorio.

check_period: Se define el periodo de tiempo válido para realizar chequeos. Este campo es obligatorio.

contact_group: Se define el grupo de contactos a los que se enviarán las notificaciones. Este campo es obligatorio.

notification_interval: Define el intervalo de tiempo en el que se enviarán notificaciones cuando el servicio se encuentre en un incidente, el valor asignado será tomando como minutos. Este campo es obligatorio.

notification_period: Define el periodo de tiempo válido para enviar notificaciones. Este campo es obligatorio.

notification_options: Se definen el tipo de notificaciones que serán enviadas, **w**= envía notificaciones de estado WARNING, **u**= envía notificaciones de estado UNKNOWN, **c**= envía notificaciones en estado CRITICAL, **r**= envía recuperaciones de un servicio (estado OK), **f**= envía notificaciones cuando inicia o termina un aleteo, **n**= no envía ningún tipo de notificación.

notifications_enabled: Se indica si se habilitan o no las notificaciones, uno (1) para habilitar, cero (0) para no habilitarlas.

```
define service{
    host_name             server_A3
    service_description   Revisar Disco SDA1
    check_command         check-disk!/dev/sda1
    max_check_attempts   5
    check_interval        6
    retry_interval        2
    check_period          24x7
    notification_interval 180
    notification_period   24x7
    notification_options  c,r
    contact_groups        admins_Servers
}
```

Figura.3.24 Definición de servicio.

Como en el caso de los hostgroups, también podemos hacer grupos de servicios para facilitar la administración, esta configuración la podemos realizar por medio del siguiente patrón:

Servidor Nagios para el control y monitoreo de red del GDF

```
define servicegroup{
    servicegroup_name    nombre_grupo
    alias                 alias
    members               servicios
}
```

Dónde:

servicegroup_name: Es el nombre que se le asigna al grupo de servicios. Este campo es obligatorio.

alias: Es una descripción que ayude a identificar el grupo de servicios. Este campo es obligatorio.

members: Es la lista de los servicios que pertenecen al grupo, el formato que se utiliza es <host1>,<servicio1>,<host2>,<servicio2>,...,<hostN>,<servicioN>

```
define servicegroup{
    servicegroup_name    Servicio_SSH
    alias                 Servicios SSH
    members               server_A2,RevisaSSH,server_A5,RevisaSSH
}
```

Figura.3.25 Definición de grupo de servicio.

• Archivos de configuración

Dentro del directorio `/usr/local/nagios/etc/objects/` se encuentran algunos archivos que muestran ejemplos de cómo configurar los hosts y los servicios, estos archivos son:

- Localhost.cfg
- Switch.cfg
- Printer.cfg

En estos archivos se observa el uso de una directiva llamada “use”, esta nos sirve para mandar llamar una configuración creada por defecto y que se encuentra en el archivo `templates.cfg`, esto sirve para evitar escribir toda una configuración por cada hosts, ahora solo mandamos llamar la configuración previa y solo hay que preocuparse por los datos necesarios como lo son el nombre del host, la dirección IP y el comando a utilizar.


```
define host{
    use                generic-switch
    host_name          Switch
    alias              Switch
    address            192.168.1.25
    hostgroups         switches
}
```

Figura.3.26 Uso de la directiva use.

- **Uso de SNMP**

Un protocolo que se utilizó en el monitoreo es SNMP (Simple Network Management Protocol), y que es fundamental para poder monitorear routers, switches y UPS. Para poder realizar el monitoreo es importante que el área correspondiente configure correctamente SNMP en sus dispositivos, estas actividades no forman parte del área del Centro de Monitoreo.

Para el monitoreo de estos equipos se usa el plugin *check_snmp* y necesita como parámetros la comunidad y el OID que identifica la variable que se desea ser monitoreada.

```
define service{
    use                generic-service
    host_name          switch
    service_description Uptime
    check_command      check_snmp!-C public -o sysuptime.0
}
```

Figura.3.27 Uso de SNMP.

- **Monitoreo de tráfico**

En algunos equipos es necesario monitorear el tráfico de los puertos, por lo que es necesario otra herramienta llamada MRTG que se encargue de obtener estos datos.

MRTG guarda la información del tráfico en el directorio */var/www/mrtg/* en archivos con extensión *.log*, Nagios por su lado, utiliza el plugin *check_mrtgtraf* para realizar el chequeo de estos archivos que genera MRTG, se necesita especificar el archivo, si se revisará el tráfico promedio o máximo, los umbrales y los minutos de tolerancia que se tendrá para el archivo.

```
define service{
    use                local-service
    host_name          Switch_C5
    service_description Trafico Puerto 0/1
    check_command      check_local_mrtgtraf!/var/www/mrtg/switch_0-1.log!AVG!1000000,1000000!5000000,5000000!5
}
```

Figura.3.28 Configuración del tráfico.

- **Nagios Remote Plugin Executor**

Otra herramienta que es muy útil para el monitoreo es NRPE, esta nos va a permitir instalar los plugins de Nagios en los servidores remotos, haciendo esto, en el servidor Nagios únicamente se debe decir qué plugin se desea ejecutar en el servidor remoto y obtener la información. Esta herramienta es importante porque es posible instalarla en servidores Windows y Linux. Es necesario instalar NRPE tanto en el servidor Nagios como en el servidor remoto.

Cuando se desea monitorear por NRPE es necesario mandar llamar en el servidor Nagios el plugin *check_nrpe* y pasarle como parámetro la definición del comando que se aloja en el servidor remoto. Este proceso no tiene ninguna diferencia si es un servidor Linux o un servidor Windows.

```
define service{
    use                local-service
    host_name          Servidor_windows
    service_description Disco Duro D:
    check_command      check_nrpe!nt_check_disk_d
}
```

Figura.3.29 Definición con *check_nrpe*.

4. RESULTADOS

4.1 Estados del monitoreo

Nagios utiliza diferentes estados a cada uno de los host y servicios, y precisamente nos indica cómo se encuentra ese host o servicio según sea el caso. Para el caso de los host tenemos los siguientes estados:

- Estado UP: Nos indica que el equipo está en conexión con la red y nos podemos comunicar con él.
- Estado DOWN: Nos indica que no está el equipo, no existe comunicación con él o se encuentra apagado.
- Estado UNREACHABLE: Nos indica que el host no puede ser alcanzado por el servidor de monitoreo.
- Estado PENDING: Este estado es mostrado cuando recién damos de alta un equipo y apenas se está realizando el primer chequeo.

En el caso de los servicios, se muestran los siguientes estados:

- Estado OK: Nos indica que no existe ningún tipo de problema en ese servicio.
- Estado WARNING: Nos indica que hay un problema menor que no tiene un impacto importante en el funcionamiento del equipo.
- Estado UNKNOWN: Nos indica que existen un problema al tratar de obtener la información del equipo, normalmente esto ocurre porque no está bien configurado el servicio.
- Estado CRITICAL: Nos indica que existe un problema que podría afectar el funcionamiento del equipo si no se soluciona inmediatamente.
- Estado PENDING: Este estado es mostrado cuando recién damos de alta un servicio y apenas se está realizando el primer chequeo.

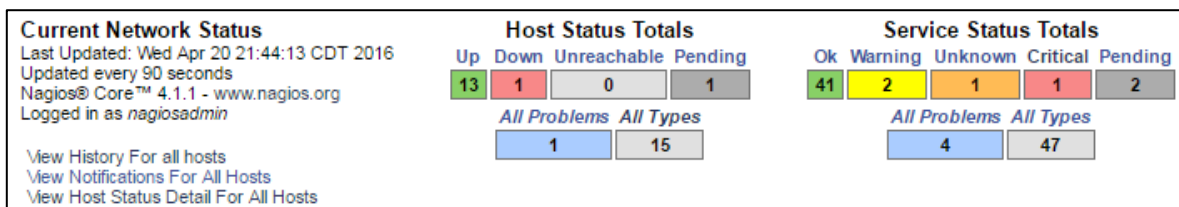


Figura.4.1 Estados totales de hosts y servicios.

Servidor Nagios para el control y monitoreo de red del GDF

En la Figura 4.2, se muestra como están agrupados los servicios que se están monitoreando:

- en la primera columna se coloca el nombre del Host (Equipo) y también podemos saber su IP
- la segunda columna (Service) se encuentra el servicio que se está monitoreando y está asociado al equipo, el servicio puede tener cualquier nombre pero es importante que nos indique lo que se está monitoreando.
- La tercera columna muestra el estado en el que se encuentra el servicio y este tiene un color asociado a cada estado.
 - OK: verde
 - WARNING: amarillo
 - UNKNOWN: naranja
 - CRITICAL: rojo
 - PENDING: gris
- La cuarta columna indica la última vez que se realizó un chequeo, por defecto, Nagios los realiza cada 5 minutos.
- La quinta columna indica el tiempo que se ha mantenido el servicio en ese mismo estado de forma consecutiva.
- La sexta columna indica el número de intento que se realiza cuando no es un estado OK. Cuando se llega al número máximo de intentos se envía una notificación.
- La séptima columna nos muestra información adicional sobre el estado de ese servicio.

Current Network Status		Host Status Totals			Service Status Totals					
Last Updated: Thu Apr 21 13:35:54 CDT 2016 Updated every 30 seconds Nagios® Core™ 4.1.1 - www.nagios.org Logged in as nagiosadmin		Up	Down	Unreachable	Pending	Ok	Warning	Unknown	Critical	Pending
		4	0	0	0	21	1	0	0	0
		All Problems All Types			All Problems All Types					
		0			4					
					1				22	
Service Status Details For Host Group 'Servidores'										
Line Results:	100									
Host	Service	Status	Last Check	Duration	Attempt	Status Info				
Servidor_Nagios	Disponibilidad del Servidor	OK	21-04-2016 13:33:35	3d 19h 16m 21s	1/4	PING OK - Packet loss = 0%, RTA = 0.65 ms				
	Usuarios Conectados	OK	21-04-2016 13:34:58	3d 19h 15m 28s	1/4	USERS OK - 0 users currently logged in				
	Verificar Cantidad de Procesos	OK	21-04-2016 13:34:46	3d 19h 14m 35s	1/4	PROCS OK: 95 processes with STATE = RSZDT				
	Verificar Carga del CPU	OK	21-04-2016 13:32:18	3d 19h 13m 42s	1/4	OK - load average: 0.03, 0.19, 0.15				
	Verificar Disco Duro	OK	21-04-2016 13:33:41	3d 19h 16m 24s	1/4	DISK OK - free space: / 4267 MB (64% inode=81%):				
	Verificar SSH	OK	21-04-2016 13:35:04	3d 19h 16m 56s	1/4	SSH OK - OpenSSH_5.3 (protocol 2.0)				
	Verificar SWAP	OK	21-04-2016 13:35:23	3d 19h 16m 3s	1/4	SWAP OK - 100% free (815 MB out of 815 MB)				
Servidor_Web	Verificar Servidor Apache	OK	21-04-2016 13:32:25	3d 19h 15m 10s	1/4	HTTP OK: HTTP/1.1 200 OK - 285 bytes in 0.015 second response time				
	Disponibilidad del Servidor	OK	21-04-2016 13:33:47	3d 18h 4m 56s	1/4	PING OK - Packet loss = 0%, RTA = 0.76 ms				
Servidor_Windows	Verificar cantidad de procesos	OK	21-04-2016 13:35:10	0d 15h 1m 6s	1/4	PROCS OK: 139 processes				
	Disponibilidad del Servidor	OK	21-04-2016 13:34:33	3d 18h 18m 18s	1/4	PING OK - Packet loss = 0%, RTA = 2.14 ms				
	Verificar Carga del CPU	OK	21-04-2016 13:32:31	0d 14h 19m 28s	1/4	NOW: Mean:2.625000% Variance: 4.134375% CUMULATIVE: Mean:2.625000% Variance: 4.134375%				
	Verificar Disco Duro C:	OK	21-04-2016 13:33:54	3d 18h 20m 36s	1/4	Used: 218661 MB (54%) Free: 181138 MB (45%)				
	Verificar Disco Duro D:	OK	21-04-2016 13:35:17	3d 16h 30m 5s	1/4	Used: 2062 MB (53%) Free: 1759 MB (46%)				
	Verificar Disco Duro E:	OK	21-04-2016 13:32:14	3d 16h 29m 3s	1/4	Used: 50 MB (1%) Free: 3800 MB (98%)				
WindowsServer	Verificar Memoria RAM	WARNING	21-04-2016 13:32:37	0d 17h 15m 58s	4/4	Mem: 1306 MB (79%) / 1642 MB (20%) Paged Mem: 1472 MB (35%) / 4095 MB (64%)				
	Verificar Puerto 5666	OK	21-04-2016 13:34:00	3d 18h 1m 53s	1/4	TCP OK - 0.002 second response time on 192.168.1.124 port 5666				
	Disponibilidad del Servidor	OK	21-04-2016 13:35:03	2d 20h 40m 6s	1/4	PING OK - Packet loss = 0%, RTA = 2.35 ms				

Figura.4.2 Detalle de los hosts y servicios.

Servidor Nagios para el control y monitoreo de red del GDF

Dentro del servidor de monitoreo de Nagios, podemos obtener más detalles de cada servicio ingresando a cada uno de ellos, en la Figura 4.3, se muestran las características del servicio y que nos proporciona más información al respecto.

En este caso se muestra la información de un Disco Duro de un Servidor, entre lo más destacado de podemos encontrar es la cantidad usada y cantidad libre de ese disco (Status Information), también tenemos la última vez que se realizó un chequeo (Last Check Time), la próxima vez que se realizará un chequeo (Next Scheduled Time) y la última vez que hubo un cambio de estado (Last State Change), cabe recordar que se pueden cambiar los tiempos de chequeo, ya sea para hacerlos más largos o más cortos.

Service State Information	
Current Status:	OK (for 3d 21h 6m 6s)
Status Information:	Used: 2062 MB (53%) Free: 1759 MB (46%)
Performance Data:	
Current Attempt:	1/4 (HARD state)
Last Check Time:	21-04-2016 18:10:15
Check Type:	ACTIVE
Check Latency / Duration:	0,000 / 0,000 seconds
Next Scheduled Check:	21-04-2016 18:15:15
Last State Change:	17-04-2016 21:05:49
Last Notification:	N/A (notification 0)
Is This Service Flapping?	NO (0,00% state change)
In Scheduled Downtime?	NO
Last Update:	21-04-2016 18:11:53 (0d 0h 0m 2s ago)

Figura.4.3 Información detallada de un servicio.

Otra cosa que es muy importante es saber el rendimiento que ha tenido un equipo o servicio en cierto periodo de tiempo, por lo que Nagios también nos permite obtener esa información, esa opción es *Availability* (Disponibilidad).

Cuando se generan estos reportes, nos arroja una tabla como la mostrada en la Figura 4.4. En esa tabla tenemos el desglose de los cuatro estados posibles, el tiempo en el que se mantuvo en ese estado, el porcentaje correspondiente y el porcentaje de tiempo conocido, esto porque en ocasiones se generan reportes en un periodo de tiempo muy grande y puede darse el caso en el que el servicio aún no se monitoreaba, por lo que no se le puede asignar a un estado en particular.

Servidor Nagios para el control y monitoreo de red del GDF

State	Type / Reason	Time	% Total Time	% Known Time
OK	Unscheduled	0d 23h 44m 3s	98,892%	98,892%
	Scheduled	0d 0h 0m 0s	0,000%	0,000%
	Total	0d 23h 44m 3s	98,892%	98,892%
WARNING	Unscheduled	0d 0h 7m 56s	0,551%	0,551%
	Scheduled	0d 0h 0m 0s	0,000%	0,000%
	Total	0d 0h 7m 56s	0,551%	0,551%
UNKNOWN	Unscheduled	0d 0h 0m 0s	0,000%	0,000%
	Scheduled	0d 0h 0m 0s	0,000%	0,000%
	Total	0d 0h 0m 0s	0,000%	0,000%
CRITICAL	Unscheduled	0d 0h 8m 1s	0,557%	0,557%
	Scheduled	0d 0h 0m 0s	0,000%	0,000%
	Total	0d 0h 8m 1s	0,557%	0,557%
Undetermined	Nagios Not Running	0d 0h 0m 0s	0,000%	
	Insufficient Data	0d 0h 0m 0s	0,000%	
	Total	0d 0h 0m 0s	0,000%	
All	Total	1d 0h 0m 0s	100,000%	100,000%

Figura.4.4 Reporte por Availability.

4.2 Notificaciones de problemas

El Centro de Monitoreo está funcionando las 24 horas del día los 365 días del año, cuando ocurre un incidente se debe notificar al área o responsable correspondiente.

No todos los incidentes son notificados, únicamente cuando un equipo o servicio entra en estado DOWN o CRITICAL se les da una tolerancia dependiendo del equipo:

- Si se trata de un equipo de red, como lo son switches, routers, access point se tiene una tolerancia de 5 minutos, si se cumple la tolerancia se notifica inmediatamente vía telefónica al responsable para indicarle el problema que surgió y para que le dé una solución.
- Si se trata de relojes biométricos se tiene una tolerancia de 60 minutos, si se cumple la tolerancia y es día hábil se notifica vía telefónica al responsable para que le dé solución al problema.
- Si se trata de un UPS se tiene una tolerancia de 5 minutos, si se cumple esta tolerancia se le notifica vía telefónica al responsable para que le dé una solución.

Servidor Nagios para el control y monitoreo de red del GDF

- En el caso de los servidores de cualquier tipo, no existe tolerancia, en cuanto se registra un equipo o servicio en CRITICAL se notifica inmediatamente al responsable para informarle el problema y se le dé una solución lo más pronto posible.

Como la Secretaría cuenta con Administraciones Tributarias en distintos puntos de la Ciudad de México, existen ocasiones en que hay fallas eléctricas, en estos casos lo podemos intuir porque los equipos que se encuentran en esa Administración Tributaria entran en estado DOWN, en estos casos se confirma con una llamada telefónica la falta de energía eléctrica. En este caso solo se notifica a los responsables la falta de energía, en caso contrario se notifica que hay problema con los equipos.

Con el sistema de monitoreo instalado se logró agilizar la detección de problemas que surgen con los equipos, esto nos reduce tiempo invertido en ubicar el problema, ahora cada responsable debe enfocarse únicamente en resolver el problema que se presente.

Esta reducción de tiempo invertido en buscar que es lo que está sucediendo, a la vez se reducen costos, ya que si deja de funcionar un equipo de muy alta importancia podría representar pérdidas económicas fuertes.

5. CONCLUSIONES

Con todas las actividades realizadas en mi servicio social puedo concluir que aprendí el funcionamiento y la importancia de un sistema de monitoreo, que en este caso fue Nagios, pero que pueden ser aplicadas para cualquier tipo de sistema.

La implementación de este sistema de monitoreo me ayudó a aplicar gran parte de mis conocimientos en un proyecto real que es muy utilizado en la mayoría de las organizaciones, es importante mencionar que a pesar de contar con las bases, al implementar el sistema Nagios, se necesita buscar día con día nueva información al respecto para poder potencializar más y más el sistema en un futuro.

Todo este sistema me sirvió también para abrir más mis conocimientos, me ayudó a la parte de investigar y saber dónde buscar esta información que me ayude a mí y al Centro de Monitoreo a tener en buen funcionamiento día con día el sistema Nagios.

Adicionalmente estas actividades me adentraron al mundo laboral, conocer cuáles son mis responsabilidades y hasta donde llegan estas, todo esto sirvió para mi formación.

REFERENCIAS

- ❖ **Tema:** Sistema de Alimentación Ininterrumpida
URL:https://es.wikipedia.org/wiki/Sistema_de_alimentaci%C3%B3n_ininterrumpida,
Fecha: 10 de abril del 2016
- ❖ **Tema:** What is Nagios?
URL: <https://www.nagios.org/about/>
Fecha: 10 de abril del 2016
- ❖ **Tema:** About Nagios Core,
URL:<https://assets.nagios.com/downloads/nagioscore/docs/nagioscore/4/en/about.html>
Fecha: 10 de abril del 2016
- ❖ **Tema:** Biblioteca de Gráficos GD,
URL:https://es.wikipedia.org/wiki/Biblioteca_de_gr%C3%A1ficos_GD
Fecha: 10 de abril del 2016
- ❖ **Tema:** System Requirements,
URL:<https://wiki.icinga.org/display/howtos/System+Requirements>
Fecha: 10 de abril del 2016
- ❖ **Tema:** Nagios XI - Hardware Requirements,
URL:<https://assets.nagios.com/downloads/nagiosxi/docs/Nagios-XI-Hardware-Requirements.pdf>
Fecha: 10 de abril del 2016
- ❖ **Tema:** Secretaría de Finanzas,
URL:http://www.finanzas.df.gob.mx/transparencia/docs/Atribuciones_SF.pdf
Fecha: 11 de abril del 2016
- ❖ **Tema:** Nagios and Nagios,
URL:<http://sites.box293.com/nagios/guides/configurations-and-definitions/hard-and-soft-states>
Fecha: 18 de abril del 2016

- ❖ **Tema:** La importancia del monitoreo de red
URL: <http://mundocontact.com/la-importancia-del-monitoreo-de-red-en-la-implementacion-de-soluciones-de-negocios/>
Fecha: 19 de abril del 2016

- ❖ **Tema:** Configuring Postfix as a Gmail Relay on CentOS,
URL: <https://charlesauer.net/tutorials/centos/postfix-as-gmail-relay-centos.php>
Fecha: 20 de abril del 2016

- ❖ **Tema:** ¿Qué es GNU/Linux?
URL: <https://www.debian.org/releases/stable/mips/ch01s02.html.es>
Fecha: 24 de abril del 2016

- ❖ **Tema:** Servidor Apache HTTP
URL: https://es.wikipedia.org/wiki/Servidor_HTTP_Apache
Fecha: 24 de abril del 2016

- ❖ **Tema:** PHP
URL: <https://secure.php.net/>
Fecha: 25 de abril del 2016

- ❖ **Tema:** VirtualBox
URL: <https://es.wikipedia.org/wiki/VirtualBox>
Fecha: 25 de abril del 2016

- ❖ **Tema:** HTML
URL: <https://developer.mozilla.org/es/docs/Web/HTML>
Fecha: 25 de abril del 2016

- ❖ **Tema:** HTML5
URL: <https://developer.mozilla.org/es/docs/HTML/HTML5>
Fecha: 25 de abril del 2016

- ❖ **Tema:** CSS
URL: <https://developer.mozilla.org/es/docs/Web/CSS>
Fecha: 25 de abril del 2016

Servidor Nagios para el control y monitoreo de red del GDF

- ❖ **Tema:** Acceso remoto SSH
URL: <http://www.aemilius.net/ayuda/articulos/acceso-ssh-ssl-secure-shell-telnet-putty.html>
Fecha: 26 de abril del 2016

- ❖ **Tema:** WinSCP Introducción
URL: <https://winscp.net/eng/docs/lang:es>
Fecha: 26 de abril del 2016

- ❖ **Tema:** Modelo de prototipo
URL: <http://es.slideshare.net/yanezcabrera/modelo-de-prototipo>
Fecha: 27 de abril del 2016

- ❖ **Tema:** Modelo en Cascada
URL: http://librosweb.es/libro/tdd/capitulo_1/modelo_en_cascada.html
Fecha: 27 de abril del 2016

- ❖ **Tema:** Nagios Core, Objects definitions
URL: <https://assets.nagios.com/downloads/nagioscore/docs/nagioscore/4/en/objectdefinitions.html>
Fecha: 27 de abril del 2016

- ❖ **Tema:** Modelo Incremental
URL: <https://procesosoftware.wikispaces.com/Modelo+Incremental?responseToken=25bd2e3b0654a13f2d7c499d5a54f0c0>
Fecha: 27 de abril del 2016

- ❖ **Tema:** Interfaz de Entrada Común.
URL: https://es.wikipedia.org/wiki/Interfaz_de_entrada_com%C3%BAn
Fecha: 01 de mayo del 2016

- ❖ **Tema:** Centos Install and Configure MRTG
URL: <http://www.cyberciti.biz/faq/centos-fedora-linux-multi-router-traffic-grapher-tutorial/>
Fecha: 03 de mayo del 2016

- ❖ **Tema:** The Multi Router Traffic Grapher
URL: <http://oss.oetiker.ch/mrtg/>
Fecha: 03 de mayo del 2016

Servidor Nagios para el control y monitoreo de red del GDF

- ❖ **Tema:** MRTG
URL: <https://es.wikipedia.org/wiki/MRTG>
Fecha: 03 de mayo del 2016

- ❖ **Tema:** Espacio de Intercambio.
URL: https://es.wikipedia.org/wiki/Espacio_de_intercambio
Fecha: 03 de mayo del 2016

GLOSARIO

Access Point: Es un dispositivo de red que interconecta dispositivos inalámbricos para dar lugar a una red inalámbrica.

Aleteo: En Nagios se llama aleteo (flapping) cuando en un periodo de tiempo muy corto el equipo está cambiando de un estado a otro.

Apache: Es un servidor web HTTP de código abierto, para plataformas Unix, Microsoft y otras que implementen el protocolo HTTP/1.1

Comunidad: Es una palabra usada en SNMP que se usa para la autenticación.

CSS: Hojas de Estilo en Cascada, es el lenguaje utilizado para describir la presentación de documentos HTML o XML, esto incluye varios lenguajes basados en XML como lo son XHTML o SVG.

GNU/Linux: En un sistema GNU/Linux, Linux es el núcleo. Dado que el núcleo Linux no forma en sí un sistema operativo funcional, se prefiere utilizar el término GNU/Linux.

HTML: Lenguaje de Marcado de Hipertextos, es el elemento de construcción más básico de una página web y se usa para crear y representar visualmente una página web.

HTML5: Conjunto de tecnologías que permite a los sitios web y a las aplicaciones ser más diversas y de gran alcance.

Kernel: También conocido como Núcleo se define el corazón del sistema operativo y es la parte fundamental del mismo

LibreOffice: Es una paquetería de oficina y de código libre, entre el software que cuenta tiene un procesador de texto, hojas de cálculo, presentaciones, bases de datos, entre otros.

Librería GD: Es una librería de gráficos para manipular imágenes.

MRTG: Multi Router Traffic Grapher, es una herramienta para obtener el tráfico de datos de una interfaz de red.

Servidor Nagios para el control y monitoreo de red del GDF

MySQL: Sistema gestor de Base de Datos de código abierto desarrollado actualmente por Oracle.

Nagios: Es un sistema de monitoreo de gran alcance que permite a las organizaciones identificar y resolver los problemas de infraestructura antes de que afecten los procesos críticos del negocio.

Nagios Core: Es un sistema de monitoreo de código abierto. Monitorea equipos y servicios que se especifiquen, alertando cuando algo está fallando o algún servicio mejora.

OID: Es un identificador de objeto utilizado en SNMP que identifica a una variable del equipo.

PHP: Es un lenguaje de programación de propósito general que es adecuado para el desarrollo web.

Ping: Comprueba el estado de la comunicación entre dos o más equipos.

Plugin: Es una serie de comandos que se ejecutan para comprobar el estado de un equipo o un servicio.

Protocolo de comunicación: Es un conjunto de reglas que permiten que varias entidades se puedan comunicar entre ellas para transmitir información.

PuTTY: Es un cliente de red que soporta protocolos SSH, Telnet y sirve para iniciar una sesión remota con otra máquina o servidor.

Router: Dispositivo de red que opera en la capa 3 (Capa de Red) del modelo OSI y que sirve para el enrutamiento de paquetes y separar los dominios de broadcast.

Servidor: Es un dispositivo que proporciona información y/o servicios que clientes le soliciten.

SNMP: Protocolo Simple de Administración de Red, es un protocolo de la capa de aplicación que facilita el intercambio de información de administración entre dos dispositivos de red.

SWAP: Es el espacio de intercambio que se usa para guardar las imágenes de los procesos que no han de mantenerse en memoria física.

Servidor Nagios para el control y monitoreo de red del GDF

Switch: Es un dispositivo de red que opera en la capa 2 (Enlace de datos) del modelo OSI.

Tráfico: Es la cantidad de datos que fluye por cierto enlace de red.

UltraISO: Herramienta que sirve para trabajar con imágenes .iso

Unix: Comprende el núcleo del sistema operativo multiusuario y multitarea, desarrollado en los laboratorios Bell de AT&T.

UPS: Uninterruptible Power Supply (Sistema de Alimentación Ininterrumpida) es un dispositivo que proporciona energía eléctrica por cierto tiempo durante un apagón eléctrico a todos los que equipos que se encuentren conectados.

VirtualBox: Software de Virtualización para arquitecturas x86/amd64 que actualmente desarrolla Oracle

VMware: Software de Virtualización para arquitecturas x86 que actualmente desarrolla EMC Corporation

WinSCP: Es una aplicación de software libre, es un cliente SFTP gráfico para Windows que emplea SSH.

APENDICE A. CONFIGURACIÓN DE DOS INTERFACES DE RED EN CENTOS

Es común que en las organizaciones exista un Proxy, y además no podamos disponer de una IP para asignarla a nuestra máquina virtual, por lo que es necesario tener dos interfaces de Red, una para tener salida a Internet y otra para tener nuestra propia red privada.

Para configurar nuestros adaptadores de red es necesario realizar lo siguiente: nos dirigimos a *Player > Manage > Virtual Machine Settings...* Una vez ahí agregamos un adaptador de red.

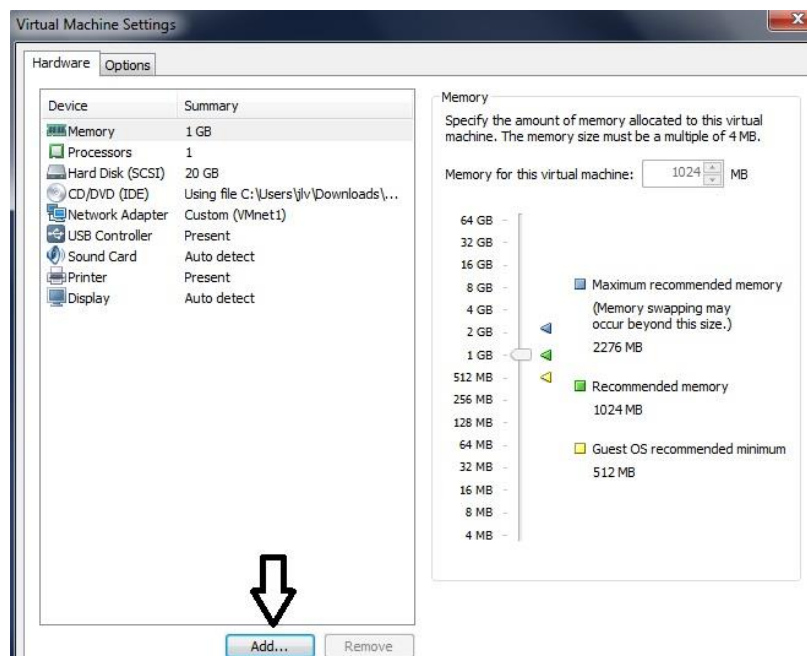


Figura.A.1 Agregar un adaptador de red.

Ahora que ya tenemos dos adaptadores de red los colocaremos en modo Custom...El primer adaptador se asignará como VMnet1(NAT), y el segundo adaptador se asignará como VMnet8(Host-Only).

Servidor Nagios para el control y monitoreo de red del GDF

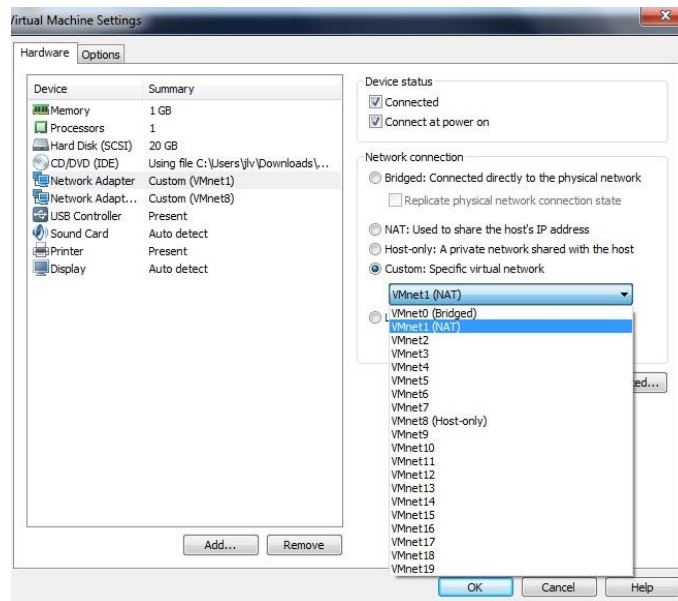


Figura.A.2 Asignación de las conexiones.

Dentro del sistema operativo debemos de configurar las direcciones IP. Para ello debemos ingresar en “Editar las conexiones...”

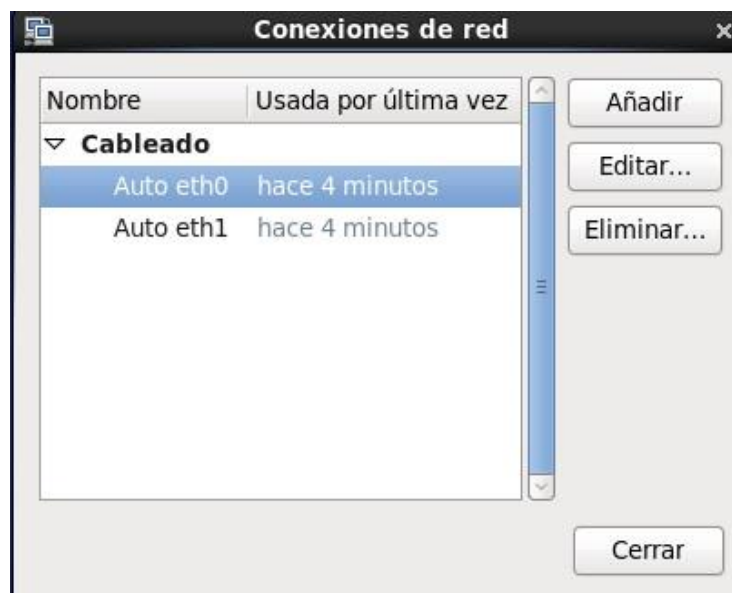


Figura.A.3 Configuración de conexiones de red.

El primer adaptador (eth0) debe ser asignado como DHCP. El otro adaptador (eth1) será asignado de forma estática.

Servidor Nagios para el control y monitoreo de red del GDF

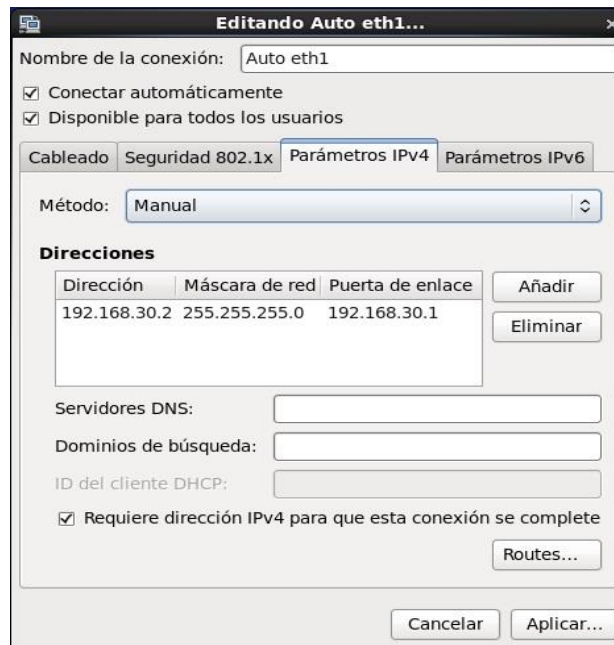


Figura.A.4 Configuración estática del adaptador.

Ahora es necesario configurar el Proxy, dentro del navegador de CentOS entramos a las *Preferencias > Avanzado > Red > Configuración...*

Ahí es donde asignaremos la IP del proxy y el puerto utilizado.

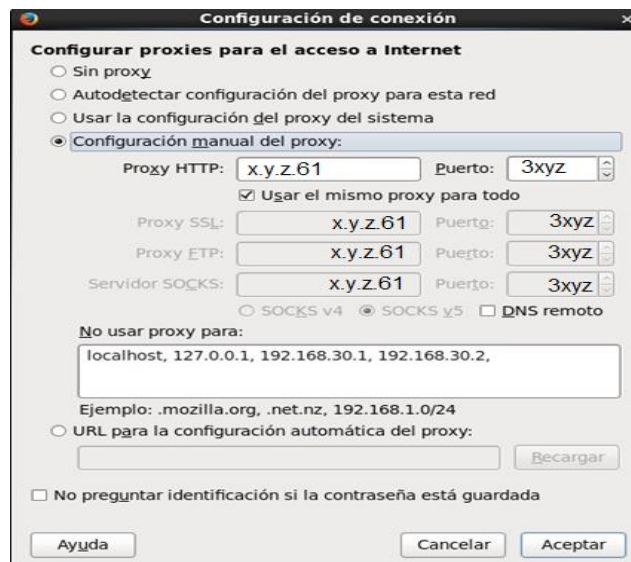


Figura.A.5 Configuración del proxy en el navegador.

Guardamos, aplicamos los cambios y nuestra configuración de red estará lista

APENDICE B. REPORTES POR AVAILABILITY DE NAGIOS

Para obtener reportes estadísticos de Nagios tenemos la opción Availability en el panel izquierdo del sistema Nagios.

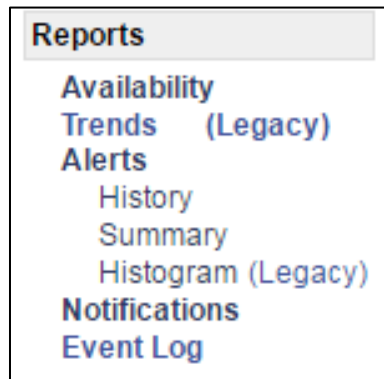


Figura.B.1 Opciones de reportes Nagios.

El primer paso es seleccionar el tipo de reporte que se realizará, ya sea un host, un servicio o un grupo de estos. Como ejemplo se seleccionará un Servicio:

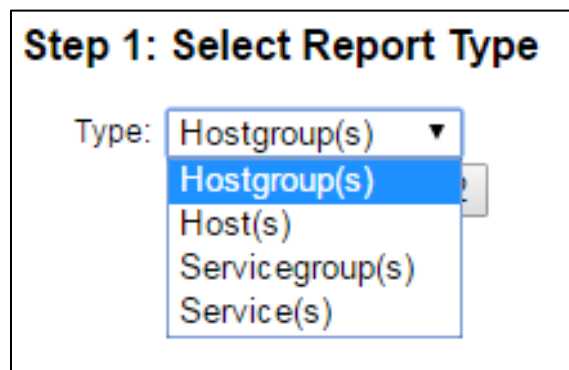


Figura.B.2 Selección del tipo de reporte.

El segundo paso es seleccionar el servicio al que le generaremos un reporte.

Servidor Nagios para el control y monitoreo de red del GDF

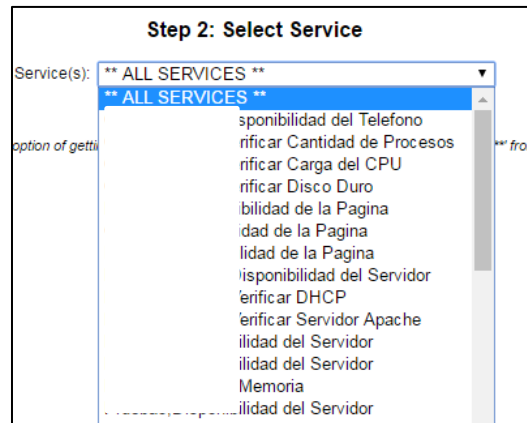


Figura.B.3 Selección del servicio.

El tercer y último paso es seleccionar el periodo de tiempo del cual se generará el reporte.

The image shows a web interface titled "Step 3: Select Report Options". It includes a "Report Period:" dropdown menu set to "Last 7 Days". Below this, there are fields for "Start Date (Inclusive):" and "End Date (Inclusive):", both set to "February 1, 2016" and "February 21, 2016" respectively.

Figura.B.4 Selección del periodo de tiempo.

Con esto ya obtenemos nuestro reporte de ese servicio por el periodo del tiempo definido en el paso 3.

State	Type / Reason	Time	% Total Time	% Known Time
OK	Unscheduled	1d 6h 52m 48s	64,333%	64,333%
	Scheduled	0d 0h 0m 0s	0,000%	0,000%
	Total	1d 6h 52m 48s	64,333%	64,333%
WARNING	Unscheduled	0d 0h 9m 4s	0,315%	0,315%
	Scheduled	0d 0h 0m 0s	0,000%	0,000%
	Total	0d 0h 9m 4s	0,315%	0,315%
UNKNOWN	Unscheduled	0d 0h 0m 0s	0,000%	0,000%
	Scheduled	0d 0h 0m 0s	0,000%	0,000%
	Total	0d 0h 0m 0s	0,000%	0,000%
CRITICAL	Unscheduled	0d 16h 58m 8s	35,352%	35,352%
	Scheduled	0d 0h 0m 0s	0,000%	0,000%
	Total	0d 16h 58m 8s	35,352%	35,352%
Undetermined	Nagios Not Running	0d 0h 0m 0s	0,000%	
	Insufficient Data	0d 0h 0m 0s	0,000%	
	Total	0d 0h 0m 0s	0,000%	
All	Total	2d 0h 0m 0s	100,000%	100,000%

Figura.B.5 Reporte de un servicio.

APENDICE C. METODOLOGIAS PARA EL DESARROLLO DE PROYECTOS

El objetivo de la Ingeniería de Software es optimizar la producción de software proporcionando las bases para el desarrollo de software de forma eficaz, para ello es fundamental aplicar metodologías que nos ayudarán a seguir cierto proceso para facilitar el trabajo y tener un software de alta calidad. A continuación se muestran algunas metodologías que se usan en el desarrollo de software.

Modelos:

- Modelo por Prototipo.
- Modelo en Cascada.
- Modelo Incremental.

MODELO PROTOTIPO

Este tipo de metodología nos permite que un sistema o partes de él se desarrollen de forma rápida y así comprender más fácilmente los aspectos necesarios del sistema y, a la vez, estar de acuerdo con la solución que se proporciona para cubrir la necesidad del cliente.

Esta metodología se puede aplicar cuando se definen los requisitos de manera genérica y conforme se está desarrollando ir ajustándose a las necesidades del cliente. Se pueden ir realizando pruebas y correcciones hasta que se quede satisfecho con una solución.

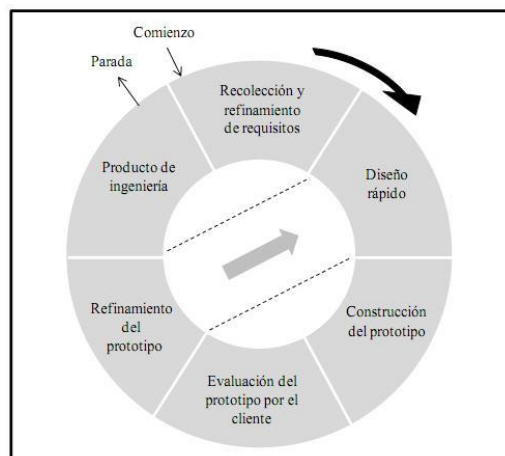


Figura.C.1 Modelo por prototipo.

Servidor Nagios para el control y monitoreo de red del GDF

. Existen diferentes etapas en esta metodología, y se describen a continuación:

- Identificar los requisitos básicos: En esta etapa se tienen de describir los requisitos de forma general, una vez que se tenga la información mínima necesaria, se procede a elaborar un diseño lógico del sistema de información.
- Desarrollar un prototipo inicial: Esta etapa consiste en, según las especificaciones, realizar la parte técnica del proyecto, ya sea la parte de programación o bien la parte de instalación de las necesidades del software.
- Usar el prototipo: Se realizan las primeras evaluaciones para comprobar el sistema y se analiza la forma en la que se implementará dentro de la organización.
- Revisión y mejora del prototipo: Se evalúa constantemente el sistema después de estar en funcionamiento, se realizan las pruebas correspondientes y de ser necesario realizar las actualizaciones para ver si el sistema cumple con los objetivos que se requieren

La ventaja de utilizar esta metodología es que no necesariamente se necesitan detallar los requisitos generales, sino que es suficiente con los objetivos generales para proceder al desarrollo del proyecto.

MODELO EN CASCADA

Esta metodología tiene una visión muy sencilla, el desarrollo del software se debe realizar siguiendo una secuencia de fases. Cada una de las fases tiene un conjunto de metas bien establecidas y las actividades de cada una contribuyen a la satisfacción de las metas. Este tipo de metodología abarca las siguientes fases:

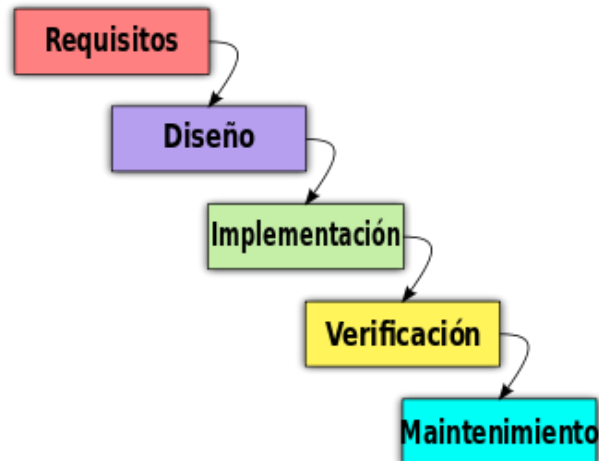


Figura.C.2 Modelo en cascada.

- **Análisis del sistema:** En esta primera fase se establecen los requisitos de todos los elementos del sistema, además se van asignando más detalladamente los requisitos de cada elemento que se va a ver involucrado en el sistema.
- **Diseño:** El diseño del software queda enfocado a los atributos distintos del proyecto como la estructura de los datos, la arquitectura del software, la característica de la interfaz, etc. En esta etapa se hace la representación del proyecto con las características requeridas antes de que inicie a implementación.
- **Implementación:** Con la fase del diseño concluida, se debe implementar el código fuente, realizando todo tipo de pruebas y corrección de errores.
- **Pruebas:** Una vez que se generó el código, se empiezan con las pruebas correspondientes. Se pruebas nos sirven para asegurar que cierta entrada produce los resultados que se requieren.
- **Mantenimiento:** Es común que el proyecto sufra cambios después de la entrega, esto debido a que posiblemente se han encontrado errores o bien a que el proyecto debe adaptarse a los cambios que se produzcan a lo largo del tiempo.

Servidor Nagios para el control y monitoreo de red del GDF

Este modelo es muy sencillo ya que las fases son muy intuitivas a la hora de estar trabajando en el proyecto, sin embargo no es muy común que se siga el flujo secuencial que propone la metodología, a veces es necesario realizar iteraciones en las fases para solucionar problemas que pudieran surgir en una fase avanzada del proyecto.

MODELO INCREMENTAL

Este modelo fue propuesto por Harlan Mills en el año 1980, sugirió el modelo incremental de desarrollo como una forma de reducir la repetición del trabajo en el proceso de desarrollo y así poder retrasar la toma de decisiones en los requisitos hasta adquirir todo el conocimiento necesario con el proyecto.

El modelo incremental aplica secuencias lineales de forma escalonada mientras progresa el tiempo. Cada secuencia lineal produce un incremento en el proyecto, generalmente el primer incremento es el producto esencial del proyecto.

De manera general, cada proceso se divide en cuatro partes:

- Análisis
- Diseño
- Código
- Prueba

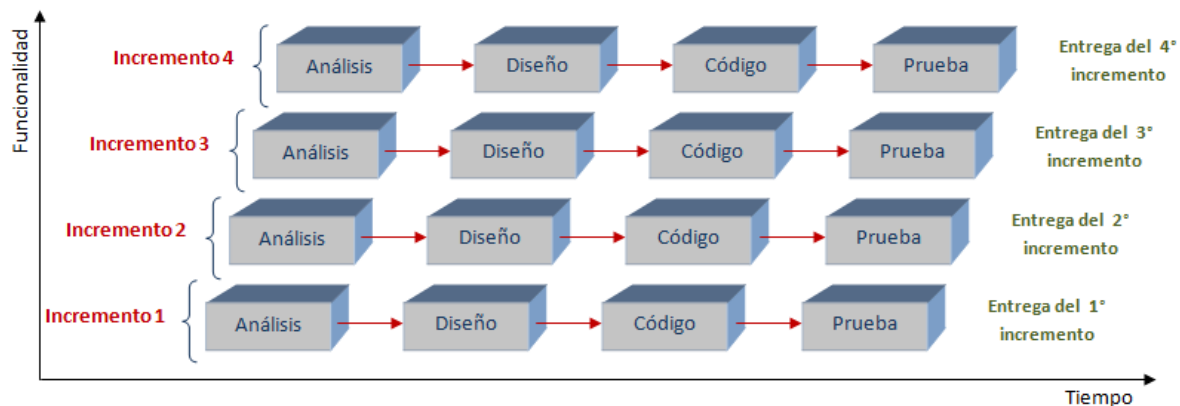


Figura.C.3 Modelo incremental.

Para la producción del software, se usa el principio de trabajo en cadena o Pipeline, con esto se realizan entregas constantes de lo obtenido en cada incremento. Con cada entrega, el cliente incluye o desecha los elementos a fin de que el software se adapte mejor a sus necesidades reales. Este proceso se repite hasta que se entregue el proyecto completo.

APÉNDICE D. INSTALACIÓN DE MRTG EN CENTOS

MRTG, o por sus siglas en inglés, Multi Router Traffic Grapher, es una herramienta de software libre que es utilizada para supervisar la carga de tráfico en interfaces de red por medio de SNMP, este protocolo proporciona la información de los bytes transmitidos por una interfaz de red distinguiendo entre la entrada y la salida.

MRTG genera gráficas que muestran la cantidad de tráfico que ha pasado por una interfaz de red.

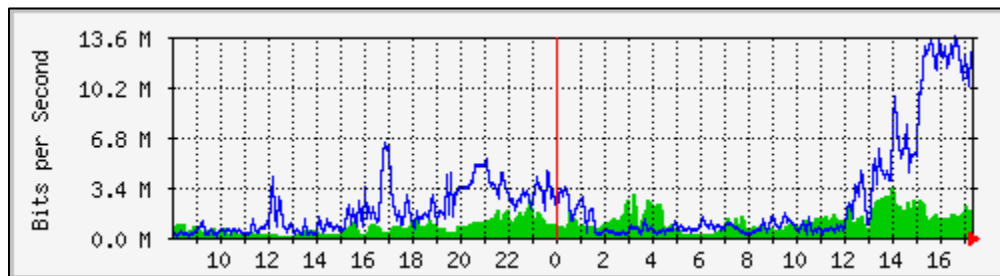


Figura.D.1 Gráfica MRTG.

Para instalar la herramienta MRTG se siguen los siguientes pasos:

Paso 1. Instalar SNMP y MRTG.

```
# yum install mrtg net-snmp net-snmp-utils
```

Paso 2. Configurar SNMP en el equipo.

Se debe configurar SNMP en el dispositivo que se va a monitorear, en este caso, y como ejemplo se configurará SNMP en el localhost. Se debe editar el archivo `/etc/snmp/snmpd.conf` incluyendo las siguientes líneas.

```
com2sec      local    localhost    public
group        MyRWGroup v1          local
group        MyRWGroup v2c         local
group        MyRWGroup usm           local
view         all    included    .1          80
access MyRWGroup "" any    noauth    exact all    all    none
```

Se guarda el archivo y se reinicia el servicio snmpd

Servidor Nagios para el control y monitoreo de red del GDF

```
# chkconfig      snmpd      on
# service        snmpd      restart
```

Paso 3. Configurar MRTG

Se debe usar el comando `cfgmaker` para crear el archivo `mrtg.cfg`, en este caso se usa la comunidad definida en la configuración SNMP y la dirección IP del dispositivo, que en este ejemplo es el `localhost`.

```
# cfgmaker --global 'WorkDir: /var/www/mrtg' --output /etc/mrtg/mrtg.cfg
public@localhost
```

Finalmente se debe crear la página web que mostrará el estado de las interfaces:

```
# indexmaker --output=/var/www/mrtg/index.html /etc/mrtg/mrtg.cfg
```

Ahora solo se debe esperar unos 5 minutos a que se generen los primeros archivos con la información de los equipos, estos estarán ubicados en el directorio `/var/www/mrtg/`.

APÉNDICE E. INSTALACIÓN DE NRPE

NRPE es una herramienta que sirve para ejecutar plugins en los servidores remotos, su instalación en un servidor Linux se realiza de la siguiente manera:

Paso 1. Descargar el paquete NRPE de la siguiente página web:

```
sourceforge.net/projects/nagios/files/nrpe-2.x/nrpe-2.15
```

Paso 2. Se instalan los paquetes necesarios

```
# yum install -y gcc glibc glibc-common gd gd-devel make net-snmp  
openssl-devel
```

Paso 3. Se descargan los plugins de Nagios

```
wget https://www.nagios-plugins.org/download/nagios-plugins-2.1.1.tar.gz
```

Paso 4. Realizar la instalación de los plugins

```
# tar -xzf nagios-plugins-2.1.1.tar.gz  
# cd nagios-plugins-2.1.1  
# ./configure  
# make  
# make install  
  
# chown nagios:nagios /usr/local/nagios/  
# chown nagios:nagios /usr/local/nagios/libexec/  
# chown nagios:nagios /usr/local/nagios/libexec/*
```

Paso 5. Instalar el paquete Xinetd

```
# yum install xinetd
```

Paso 6. Descomprimir e instalar el paquete NRPE que se descargó al inicio.

```
# tar -xzf nrpe-2.15.tar.gz  
# cd nrpe-2.15  
  
# ./configure  
# make all
```

Servidor Nagios para el control y monitoreo de red del GDF

```
# make install
# make install-plugin
# make install-daemon
# make install-daemon-config
# make install-xinetd
```

Paso 7. Dar permisos al servidor Nagios, esto se realiza en el archivo `/etc/xinetd.d/nrpe`, donde se ubica la directiva `only_from` colocar la IP del servidor Nagios.

Paso 8. En el archivo `/etc/services` incluir al final del archivo la siguiente línea.

```
nrpe      5666/tcp      # NRPE
```

Paso 9. Iniciar el servicio Xinetd

```
# service xinetd start
```

Se debe permitir el puerto 5666 en el firewall del servidor.

Una vez instalado, se generará un archivo en el directorio `/usr/local/nagios/etc/` llamado `nrpe.cfg`, en donde se deben colocar los comandos para ejecutar los plugins.

```
command[check_users]=/usr/local/nagios/libexec/check_users -w 5 -c 10
command[check_load]=/usr/local/nagios/libexec/check_load -w 15,10,5 -c 30,25,20
command[check_hda1]=/usr/local/nagios/libexec/check_disk -w 20% -c 10% -p /dev/hda1
command[check_zombie_procs]=/usr/local/nagios/libexec/check_procs -w 5 -c 10 -s Z
command[check_total_procs]=/usr/local/nagios/libexec/check_procs -w 150 -c 200
command[check_memoria]=/usr/local/nagios/libexec/check_proc2s -w 150 -c 200
```

Figura.E.1 Comandos Linux del archivo nrpe.cfg

Ahora solo podemos ejecutar el plugin `check_nrpe` en el servidor Nagios con la definición del comando en el servidor remoto.