

**ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL**



**Facultad de Ingeniería En Electricidad Y Computación**

**“Análisis del Rendimiento de Sistemas VoIP bajo Condiciones de Red  
Variable”**

**INFORME DE MATERIA DE GRADUACIÓN:**

**Previo a la obtención del título de:**

**LICENCIADO EN REDES Y SISTEMAS OPERATIVOS**

Presentado por:

Edward Gastón Sinche Cruz

Juan Carlos Enrique Ortega Acosta

GUAYAQUIL – ECUADOR

2011

## **AGRADECIMIENTO**

Agradezco a Dios por haberme dado la vida y por la oportunidad de tener una familia comprensiva y unida, que siempre me han brindado el apoyo necesario en todas las etapas de mi vida y me han motivado a seguir adelante en cada tropiezo que se me ha presentado a lo largo de mi carrera.

**Juan Carlos Enrique Ortega Acosta**

A Dios primeramente, tras guiarme por buen camino en las etapas de mi vida. A mis padres, por enseñarme el trabajo duro, la perseverancia y la Humildad. A la vida, por enseñarme que nada se consigue quedándonos sentados sino todo lo contrario con esfuerzo, sacrificio y dedicación.

**Edward Gastón Sinche Cruz**

## **DEDICATORIA**

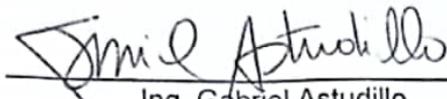
A mis padres por ser mis principales puntos de apoyo y comprensión en este camino. A mi hermana, por ser un angelito de Dios. A mi esposa Ninfa, por haberme dado su apoyo para alcanzar esta meta, A mi hija, por haberme complementado en esta vida. Al Ing. Gabriel Astudillo por haberme dado todo el apoyo necesario en la realización de este proyecto. Al Ing. Rayner Durango que a lo largo de la carrera impartió conocimientos y consejos que nos servirán de mucho en la formación de nuestro perfil profesional.

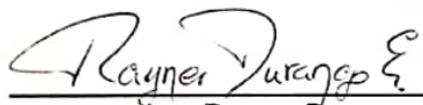
**Juan Carlos Enrique Ortega Acosta**

A mi padre por ser un ejemplo de superación. A mi madre por brindarme todo su apoyo en tiempos buenos y difíciles. A mi hermana por estar siempre pendiente de mis estudios. Al Ing. Gabriel Astudillo por encaminarme en el mundo de la telefonía IP. Al Ing. Rayner Durango por enseñarnos los valores y las normas necesarias para ser unos profesionales de éxito.

**Edward Gastón Sinche Cruz**

# TRIBUNAL DE SUSTENTACIÓN

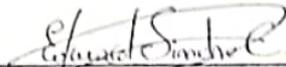
  
Ing. Gabriel Astudillo  
PROFESOR DE MATERIA DE GRADO

  
Ing. Rayner Durango  
PROFESOR DELEGADO POR EL DECANO

# DECLARACIÓN EXPRESA

"La responsabilidad del contenido de este trabajo de Grado, nos corresponde exclusivamente; y el patrimonio intelectual de la misma a la Escuela Superior Politécnica del Litoral"

(Reglamento de Graduación de la ESPOL)



---

**Edward Gastón Sinche Cruz**



---

**Juan Carlos Enrique Ortega Acosta**

# RESUMEN

Este proyecto de graduación consiste en la implementación del software Nagios, dirigido especialmente al monitoreo de una central telefónica Asterisk. Ya que Nagios es una herramienta de monitoreo de red muy potente y completa al momento de presentar informes y graficas de lo que ocurre en tiempo real en los dispositivos a monitorear.

Nagios es una herramienta muy flexible, por tal motivo es posible crear plugins para personalizar los monitoreo, enfocados específicamente en este proyecto a los canales de comunicación, para así poder evitar posibles errores y establecer mejoras y soluciones que hagan de Asterisk un servicio de mayor disponibilidad.

En cada capítulo de este proyecto se describe como se hizo para la implementación de un servidor Nagios, que esta personalizado para el monitoreo de una central Asterisk.

# INTRODUCCION

La telefonía puede separarse en dos grandes grupos según su tecnología de audio: ANALOGICA o telefonía convencional y DIGITAL refiriéndose a VoIP. La telefonía IP hoy en día es muy usada y requerida por muchas empresas debido a los bajos costos en cuanto a tarifas y mejores servicios en aplicaciones.

Sin embargo, estas aplicaciones de servicios solo estaban disponibles en caros sistemas propietarios, hoy en día Asterisk nos ofrece servicios como, la creación de extensiones, llamadas en conferencia, menús de voz interactivos y distribución automática de llamadas. Además se pueden crear nuevas funcionalidades mediante el propio lenguaje de Asterisk o módulos escritos en C o mediante scripts AGI escritos en Perl o en otros lenguajes.

Todas estas aplicaciones demandan un consumo de ancho de banda considerable, en la mayoría de los casos variable, haciendo que cada vez se creen sistemas de monitoreo para estas redes, para obtener valores, interpretar, y tomar la mejor decisión a los posibles problemas de tráfico que se provoquen, y poder solucionarlo de una manera rápida y eficiente.

# ÍNDICE

INTRODUCCION	
CAPITULO I: ANTECEDENTES Y JUSTIFICACION .....	1
<b>1.1 ANTECEDENTES</b> .....	1
<b>1.2 JUSTIFICACION</b> .....	4
<b>1.3 DESCRIPCION DEL PROYECTO</b> .....	6
1.3.1 OBJETIVO GENERAL .....	6
1.3.2 OBJETIVOS ESPECIFICOS .....	6
1.3.3 ALCANCE .....	10
<b>1.4 METODOLOGIA</b> .....	10
<b>1.5 PERFIL DE LA TESIS</b> .....	11
CAPITULO II: ASTERISK Y NAGIOS .....	12
<b>2.1 ASTERISK</b> .....	12
<b>2.2 PBX</b> .....	14
<b>2.3 FXS Y FXO</b> .....	16
2.3.1 FXS .....	16
2.3.2 FXO .....	16
2.3.3 FXO Y FXS SIN UNA CENTRALITA .....	17
2.3.4 FXO Y FXS CON UNA CENTRALITA .....	17
2.3.5 PROCEDIMIENTOS DEL FXS/FXO – COMO FUNCIONA TECNICAMENTE .....	18
<b>2.4 SIP</b> .....	19
2.4.1 FUNCIONAMIENTO DEL PROTOCOLO .....	20
2.4.1.1 AGENTES DE USUARIO .....	21
2.4.1.2 SERVIDORES DE REGISTRO .....	22
<b>2.5 IAX</b> .....	24

2.5.1	OBJETIVOS DEL IAX .....	25
2.5.2	PROPIEDADES BASICAS.....	25
<b>2.6</b>	<b>H.323</b> .....	<b>27</b>
2.6.1	LA TOPOLOGIA CLASICA DE UNA RED BASADA EN H.323 .....	28
<b>2.7</b>	<b>MGCP</b> .....	<b>28</b>
<b>2.8</b>	<b>UIT-T</b> .....	<b>30</b>
<b>2.9</b>	<b>GNU GPL</b> .....	<b>31</b>
<b>2.10</b>	<b>GNU/LINUX</b> .....	<b>32</b>
<b>2.11</b>	<b>IVR</b> .....	<b>33</b>
2.11.1	SERVICIOS .....	33
2.11.2	COMO FUNCIONA .....	35
<b>2.12</b>	<b>NAGIOS</b> .....	<b>35</b>
2.12.1	HISTORIA DE NAGIOS .....	35
2.12.2	QUE ES NAGIOS .....	36
2.12.3	NAGIOS A NIVEL DE EMPRESA .....	37
2.12.4	CARACTERISTICAS .....	38
<b>2.13</b>	<b>FUNCIONAMIENTO DE NAGIOS</b> .....	<b>41</b>
2.13.1	MONITOREO.....	41
2.13.2	ALERTAS .....	41
2.13.3	RESPUESTA .....	42
2.13.4	PRESENTACION DE INFORMES .....	42
2.13.5	MANTENIMIENTO.....	42
2.13.6	PLANIFICACION .....	43
<b>2.14</b>	<b>IMPORTANCIA DE LA MONITORIZACION DEL SERVIDOR ASTERISK</b> .....	<b>44</b>
<b>CAPITULO III: IMPLEMENTACION Y CONFIGURACION</b> .....		<b>46</b>
<b>3.1</b>	<b>COMPONENTES DEL PROYECTO</b> .....	<b>46</b>
3.1.1	SERVIDOR NAGIOS .....	46
3.1.2	SERVIDOR ASTERISK.....	47
3.1.3	SELECCIÓN DE SOFTWARE DE MONITOREO.....	48

<b>3.2</b>	<b>INSTALACION DEL SISTEMA NAGIOS</b> .....	49
3.2.1	PROCEDIMIENTO DE INSTALACION .....	49
<b>3.3</b>	<b>INSTALACION DE NRPE EN LOS SERVIDORES</b> .....	50
3.3.1	INSTALACION DE NRPE EN EL SERVIDOR ASTERISK .....	50
3.3.2	INSTALACION DE NRPE EN EL SERVIDOR NAGIOS .....	51
<b>3.4</b>	<b>INSTALACION DE SNMP EN LOS SERVIDORES</b> .....	52
3.4.1	INSTALACION DE SNMP EN EL SERVIDOR ASTERISK .....	52
3.4.2	INSTALACION DE SNMP EN EL SERVIDOR NAGIOS .....	53
<b>3.5</b>	<b>INSTALACION DEL MOTOR GRAFICO PNP4NAGIOS</b> .....	54
<b>3.6</b>	<b>INSTALACION DEL SERVIDOR DE CORREO SEND MAIL</b> .....	55
CAPITULO IV: FUNCIONAMIENTO Y PRUEBAS DEL PROYECTO .....		57
<b>4.1</b>	<b>INTERFAZ WEB DE NAGIOS</b> .....	57
<b>4.2</b>	<b>DESCRIPCION DEL FUNCIONAMIENTO DE NAGIOS</b> .....	59
<b>4.3</b>	<b>PRUEBAS REALIZADAS</b> .....	62
4.3.1	MONITOREO DE LAS LLAMADAS ACTIVAS .....	62
4.3.2	MONITOREO DE LAS LLAMADAS PROCESADAS .....	63
4.3.3	MONITOREO DEL NUMERO DE CANALES ACTIVOS .....	64
4.3.4	MONITOREO DEL CANAL IAX Y SIP .....	65
4.3.5	FUNCIONAMIENTO DE NOTIFICACIONES POR E-MAIL .....	66
<b>4.4</b>	<b>ANÁLISIS DEL RENDIMIENTO DE ASTERISK</b> .....	67
4.4.1	ANÁLISIS DE LA CARGA DEL PROCESADOR .....	67
4.4.2	ANÁLISIS DE LA CARGA DE LA MEMORIA RAM .....	67

# ÍNDICE DE FIGURAS

FIGURA. 1.1 ESQUEMA A IMPLEMENTAR .....	8
FIGURA 2.1 FXS/FXO SIN UNA PBX .....	17
FIGURA 2.2 FXS/FXO CON UNA PBX .....	17
FIGURA 2.3 FUNCIONAMIENTO DE NAGIOS .....	43
FIGURA 3.1 ELECCION DE SOFTWARE NAGIOS .....	48
FIGURA 4.1 SERVICIOS DEL SERVIDOR NAGIOS .....	58
FIGURA 4.2 ESTADO DE FALLO DEL SERVIDOR ASTERISK .....	59
FIGURA 4.3 ESTADO DE RECUPERACIÓN DEL SERVIDOR ASTERISK .....	60
FIGURA 4.4 NOTIFICACION SERVICIO ASTERISK CAÍDO .....	61
FIGURA 4.5 NOTIFICACION SERVICIO ASTERISK RECUPERADO .....	61
FIGURA 4.6 MONITOREO DE LLAMADAS ACTIVAS .....	62
FIGURA 4.7 MONITOREO MODO GRAFICO LLAMADAS ACTIVAS .....	63
FIGURA 4.8 MONITOREO DE LAS LLAMADAS REALIZADAS.....	63
FIGURA 4.9 MONITOREO MODO GRAFICO DE LAS LLAMADAS REALIZADAS...	64
FIGURA 4.10 MONITOREO DE LAS LLAMADAS ACTIVAS .....	64
FIGURA 4.11 MONITOREO DE LOS CANALES IAX Y SIP .....	65
FIGURA 4.12 MONITOREO MODO GRAFICO DEL CANAL SIP.....	65
FIGURA 4.13 MONITOREO MODO GRAFICO DEL CANAL IAX.....	65
FIGURA 4.14 NOTIFICACIONES PARA EL ADMINISTRADOR .....	66
FIGURA 4.15 CARGA DEL PROCESADOR DE ASTERISK.....	67
FIGURA 4.16 CARGA INICIAL DEL PROCESADOR Y DE LA MEMORIA RAM.....	68
FIGURA 4.17 CARGA FINAL DEL PROCESADOR Y DE LA MEMORIA RAM .....	68

# CAPITULO 1

## 1. ANTECEDENTES Y JUSTIFICACIONES

### 1.1 ANTECEDENTES

Antiguamente la infraestructura clásica de la telefonía existían muchos problemas tales como el mantenimiento de las centrales y la infraestructura con un alto costo, además se dependían de un solo proveedor, poca escalabilidad, los modelos de más capacidad son mucho más caros, las operadoras requerían una capacitación especial para manejar cada modelo de las Centrales y la poca funcionalidad al hablar por teléfono y su re direccionamiento.

Hoy en día la comunicación de muchas empresas, emplean un papel muy importante ya que estas llevan a un gran éxito o incluso al fracaso de las mismas. Es por esto que es importante implementar y manejar un buen sistema de comunicación basados en los requerimientos necesarios para dicha empresa.

Mark Spencer de la empresa Digium en 1999 creó la aplicación Asterisk para controlar y gestionar comunicaciones analógica, digital y

a la vez fue donada a la comunidad de licencia libre tras la cual se han recibido muchas colaboraciones y mejoras por parte de muchos desarrolladores de software libre sin solicitar nada a cambio.

Esta aplicación poco a poco se ha convertido en la evolución de las tradicionales centralitas analógicas y digitales permitiendo la integración de la tecnología más actual que es la VoIP. Asterisk se convierte así en el más completo, avanzado y económico sistema de comunicación existente en la actualidad.

Asterisk nos brinda soluciones basadas en Voz-IP es decir las comunicaciones pasan por la infraestructura de red de la empresa, también nos permite una gran escalabilidad y precios muchos más económicos a la vez el mantenimiento es totalmente independiente del proveedor por lo que es un software libre y no requiere de ninguna licencia de ningún tipo.

Asterisk es una solución moderna, poderosa, escalable y económica para las necesidades de comunicación, sin embargo la calidad de servicio que ofrecen las redes de comunicación no nos brindan la suficiente seguridad a la entrega de los datos ya que muchas cosas le

ocurren a los paquetes desde su origen hasta el destino como por ejemplo la pérdida de paquetes, el retardo, el Jitter, los errores en la transmisión de los datos.

De tal manera vamos a evaluar los servicios que nos brinda nuestro servidor Asterisk, para así poder analizar qué tan recomendable es trabajar con esta tecnología hoy en día.

## 1.2 JUSTIFICACION

Hoy en día las empresas han crecido de tal manera que su infraestructura de red ha crecido a pasos acelerados junto a los diferentes departamentos de servicios.

La necesidad de tener una comunicación permanente a bajo costo, y tan buena o quizás mejor a los sistemas propietarios de VoIP, nos lleva a buscar herramientas de código abierto muy eficaces para realizar este trabajo de comunicación usando VoIP como Asterisk.

Trayendo consigo una variedad de beneficios ya mencionadas anteriormente en la introducción del proyecto, beneficios como: la creación de extensiones, envío de mensajes de voz a e-mail, llamadas en conferencia, menús de voz interactiva y distribución automática de llamadas. Además se pueden crear nuevas funcionalidades mediante el propio lenguaje de Asterisk o módulos escritos en C o mediante scripts AGI escritos en Perl o en otros lenguajes.

Desde el punto de vista práctico la investigación estará bien justificada en las herramientas que se pueden usar con Asterisk para las empresas, que mejoraran y agilizaran el servicio al momento de la comunicación, creando satisfacción y equilibrio interno.

Pero no solo basta con tener una buena configuración de Asterisk para dar un buen servicio en cuanto a la calidad de la llamada, para eso usaremos un software de monitoreo para detectar y estudiar los posibles problemas que de por sí, hay en las redes, VoIP y de datos.

Por otro lado, la utilización del software de monitoreo Nagios, fue elegido debido a sus múltiples herramientas de monitoreo de, Aplicaciones, servicios, sistemas operativos, protocolos de red, métrica de sistema y componente de infraestructura, los cuales sirven para poder concluir cual es el comportamiento de la red.

Por último, esta investigación pondrá en manifiesto los conocimientos adquiridos durante la carrera, esperando que sirvan de bases para otros estudios que partan de la misma problemática que aquí se especifica.

### **1.3 DESCRIPCION DEL PROYECTO**

El análisis del rendimiento de la calidad de servicio de la implementación de una central telefónica basada en Asterisk pretenda alcanzar los siguientes objetivos:

#### **1.3.1 OBJETIVO GENERAL**

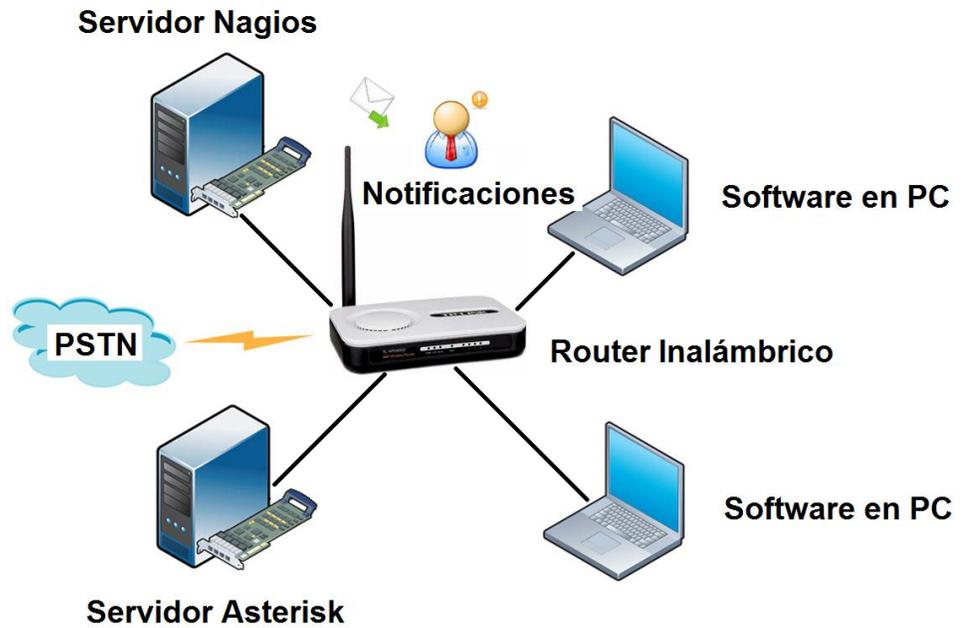
Evaluar la rentabilidad de un sistema VoIP en cuanto a la disponibilidad de servicio de una red basada en Servidor Asterisk utilizando el sistema de monitoreo Nagios.

#### **1.3.2 OBJETIVOS ESPECÍFICOS**

- Implementar una central telefónica IP (IPBX) basado en Asterisk para poder realizar las pruebas necesarias para evaluar el rendimiento del sistema VoIP.
- Personalizar Nagios para monitorear el Servidor Asterisk, puesto que nativamente no viene configurado con estos servicios.

- Realizar pruebas que nos permitan visualizar el comportamiento de los servicios de Asterisk a monitorear, mediante las herramientas de presentación de informes gráficos de Nagios.

El proyecto que se va a implementar y documentar consiste en la instalación de una centralita telefónica basada en un servidor Asterisk que va a estar conectada a dos computadoras que van a tener previamente instalado un SoftPhones cada uno, el flujo de llamadas desde la telefonía IP conectada al servidor Asterisk va hacer monitoreada por la aplicación Nagios para así poder hacer pruebas de monitoreo de los canales SIP, canales IAX, llamadas activas, llamadas procesadas, etc., que nos permitan visualizar el estado de los servicios del servidor Asterisk. El esquema a utilizar en este proyecto se define en la figura 1.1.



**Figura. 1.1 Esquema a Implementar**

Para la red física utilizaremos un router inalámbrico TP-Link TL-WR340G el cual nos permite utilizar tanto la red inalámbrica como la red cableada ya que este dispositivo cuenta con 4 puertos RJ45 para interconectar tanto los usuarios como el Servidor Asterisk y Nagios, otra característica que brinda este dispositivo es el direccionamiento dinámico mediante la opción de DHCP con que cuenta este dispositivo.

El sistema de monitoreo Nagios se instalara en una computadora con sistema operativo Centos5.4, en el cual instalaremos los paquetes necesarios tales como: Apache 2.0, PHP, GCC: Librería de desarrollo y compilación, GD: Librerías de desarrollo, Nagios 3.2.0 Core, Nagios Plugins 1.4.13, Nagios addons, NRPE 2.12 y Net-SNMP para la comunicación entre el Servidor Nagios con el Servidor Asterisk.

Por último instalaremos Asterisk en una computadora con un sistema operativo Centos5.4 con sus respectivos paquetes tales como: Asterisk-1.6.2.14, Asterisk-addons-1.6.2.2, libpri-1.4.10.2.

### **1.3.3 ALCANCE**

En este proyecto tal como se indica en los objetivos, se pretende:

- Realizar una tabla comparativa para la elección del mejor programa en cuanto a sus herramientas y aplicativos para el monitoreo de Asterisk.
- Desarrollar un escenario que refleje las características del monitoreo de un sistema VoIP.
- Analizar los servicios que vamos a monitorear basándonos en la herramienta grafica PNP4Nagios.

### **1.4 METODOLOGÍA**

Para alcanzar los objetivos antes mencionados, realizaremos una instalación de un Servidor Asterisk como aplicación raíz la que permite convertir a una computadora en una Central Telefónica basada en VoIP y un Servidor Nagios con el que podemos monitorear el Servidor Asterisk, sobre dos plataformas bases de código abierto Linux cada uno con la distribución de CentOS 5.4.

## **1.5 PERFIL DE LA TESIS**

Nuestra tesis tiene como objetivo principal monitorear los servicios de una Central VoIP basada en Servidor Asterisk para así poder verificar la disponibilidad de la centralita.

En el capítulo 2, se revisa los fundamentos teóricos, de los principales conceptos relacionados con VoIP, Asterisk y Nagios. Además de la importancia del monitoreo con la herramienta Nagios, de un servidor Asterisk.

En el capítulo 3, mostramos los requerimientos de software y hardware para la instalación de ambos servidores, Nagios y Asterisk.

En el capítulo 4, se realizan las pruebas necesarias y análisis para mostrar el funcionamiento del servidor Nagios frente a los servicios de Asterisk.

Por último se elaboran las conclusiones y las recomendaciones a seguir para este proyecto.

# CAPITULO 2

## 2. ASTERISK Y NAGIOS

### 2.1 ASTERISK

Es un programa de software libre que proporciona funcionalidades de una central telefónica PBX. Como cualquier PBX, se puede conectar un número determinado de teléfonos para hacer llamadas entre sí e incluso conectar a un proveedor de VoIP o bien a una Red Digital de Servicios Integrados tanto básicos como primarios.

Mark Spencer, de Digium, inicialmente creó Asterisk y actualmente es su principal desarrollador, junto con otros programadores que han contribuido a corregir errores y añadir novedades y funcionalidades. Originalmente desarrollado para el sistema operativo GNU/Linux, Asterisk actualmente también se distribuye en versiones para los sistemas operativos BSD, MacOSX, Solaris y Microsoft Windows, aunque la plataforma nativa (GNU/Linux) es la que cuenta con mejor soporte de todas.

Asterisk incluye muchas características anteriormente sólo disponibles en costosos sistemas propietarios PBX como buzón de voz, conferencias, IVR, distribución automática de llamadas, y otras muchas más. Los usuarios pueden crear nuevas funcionalidades escribiendo un plan de marcado en el lenguaje de script de Asterisk o añadiendo módulos escritos en lenguaje C o en cualquier otro lenguaje de programación reconocido por Linux.

Para conectar teléfonos estándar analógicos son necesarias tarjetas electrónicas telefónicas FXS o FXO fabricadas por Digium u otros proveedores, ya que para conectar el servidor a una línea externa no basta con un simple módem.

Quizá lo más interesante de Asterisk es que reconoce muchos protocolos VoIP como pueden ser SIP, H.323, IAX y MGCP. Asterisk puede interoperable con terminales IP actuando como un registrador y como Gateway entre ambos. [1]

## **2.2PBX**

Un PBX o PABX siglas en inglés de Private Branch Exchange y Private Automatic Branch Exchange para PABX, cuya traducción al español sería Central secundaria privada automática, es cualquier central telefónica conectada directamente a la red pública de teléfono por medio de líneas troncales para gestionar, además de las llamadas internas, las entrantes y salientes con autonomía sobre cualquier otra central telefónica. Este dispositivo generalmente pertenece a la empresa que lo tiene instalado y no a la compañía telefónica, de aquí el adjetivo privado a su denominación.

Un PBX se refiere al dispositivo que actúa como una ramificación de la red primaria pública de teléfono, por lo que los usuarios no se comunican al exterior mediante líneas telefónicas convencionales, sino que al estar el PBX directamente conectado a la RTPC (red telefónica pública conmutada), será esta misma la que encaminara la llamada hasta su destino final mediante enlaces unificados de transporte de voz llamados líneas troncales. En otras palabras, los usuarios de una PBX no tienen asociada ninguna central de

teléfono pública, ya que es el mismo PBX que actúa como tal, análogo a una central pública que da cobertura a todo un sector mientras que un PBX lo ofrece a las instalaciones de una compañía generalmente.

Erróneamente se le llama PBX a cualquier central telefónica aunque no gestione las llamadas externas, bastando solo con que conmute líneas exteriores pertenecientes a otra central que sí estaría conectada a la RTPC. Estas serían centrales híbridas: gestionan llamadas y enlazan líneas internas o extensiones pero al momento de comunicarse a un destino exterior, tan solo interconectaría el terminal con una línea convencional de la compañía de teléfono, mientras que un PBX se encargaría de procesar directamente el número marcado hacia el procesador central de la ciudad.[2]

## **2.3 FXS y FXO**

### **2.3.1 FXS**

La interfaz de abonado externo es el puerto que efectivamente envía la línea analógica al abonado. En otras palabras, es el puerto de la pared que envía tono de marcado, corriente para la batería y tensión de llamada.

### **2.3.2 FXO**

Interfaz de central externa es el puerto que recibe la línea analógica. Es el puerto del teléfono o aparato de fax, o el puerto de su centralita telefónica analógica. Envía una indicación de colgado o descolgado, cierre de bucle. Como el puerto FXO está adjunto a un dispositivo, tal como un fax o teléfono, el dispositivo a menudo se denomina dispositivo FXO.

### 2.3.3 FXO Y FXS SIN UNA CENTRALITA

El teléfono se conecta directamente al puerto FXS que brinda la empresa telefónica.

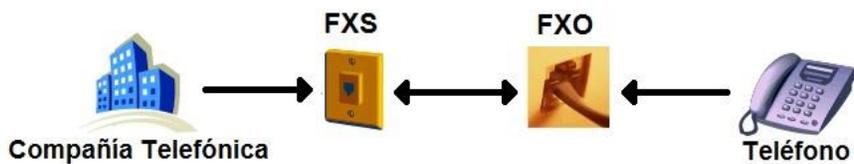


Figura 2.1.- FXS/FXO sin una PBX

### 2.3.4 FXO Y FXS CON UNA CENTRALITA

Debe conectar las líneas que suministra la empresa telefónica a la centralita y luego los teléfonos a la centralita. Por lo tanto, la centralita debe tener puertos FXO, para conectarse a los puertos FXS que suministra la empresa telefónica y puertos FXS para conectar los dispositivos de teléfono o fax.

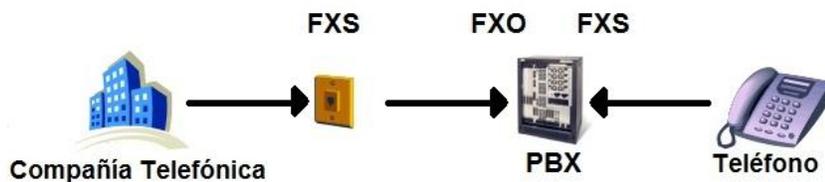


Figura 2.2.- FXS/FXO con una PBX

### **2.3.5 PROCEDIMIENTOS DEL FXS / FXO – CÓMO FUNCIONA TÉCNICAMENTE**

Secuencia:

Cuando se realiza una llamada:

1. Tome el teléfono, el dispositivo FXO. El puerto FXS detecta que ha descolgado el teléfono.
2. Marque el número de teléfono, que pasa como dígitos de Tono Dual Multi Frecuencia (DTMF) al puerto FXS.

Llamada entrante: El puerto FXS recibe una llamada y luego envía un voltaje de llamada al dispositivo FXO adjunto.

1. El teléfono suena
2. En cuanto levante el teléfono, podrá responder la llamada.
3. Finalización de la llamada – normalmente el puerto FXS depende de alguno de los dispositivos FXO conectados para finalizar la llamada.

Nota: La línea telefónica analógica pasa un voltaje DC de aproximadamente 50 voltios al puerto FXS. Es por ello que

recibe una “descarga” eléctrica cuando toca una línea telefónica conectada. Esto permite realizar una llamada cuando se produce un corte de energía. [3]

## **2.4 SIP**

Session Initiation Protocol o Protocolo de Inicio de Sesiones, es uno de los protocolos de señalización para voz sobre IP.

El protocolo SIP fue diseñado por el IETF con el concepto de "caja de herramientas", es decir, el protocolo SIP se vale de las funciones aportadas por otros protocolos, que da por hechas y no vuelve a desarrollar. Debido a este concepto, SIP funciona en colaboración con otros muchos protocolos. El protocolo SIP se concentra en el establecimiento, modificación y terminación de las sesiones, y se complementa entre otros con el SDP, que describe el contenido multimedia de la sesión, por ejemplo qué direcciones IP, puertos y códec se usarán durante la comunicación. También se complementa con el RTP (Real-time Transport Protocol). RTP es el verdadero portador para el contenido de voz y vídeo que intercambian los participantes en una sesión establecida por SIP.

Otro concepto importante en su diseño es el de extensibilidad. Esto significa que las funciones básicas del protocolo, definidas en la RFC 3261, pueden ser extendidas mediante otras RFC (Requests for Comments) dotando al protocolo de funciones más potentes.

Las funciones básicas del protocolo incluyen:

- Determinar la ubicación de los usuarios, aportando movilidad.
- Establecer, modificar y terminar sesiones entre usuarios.

#### **2.4.1 FUNCIONAMIENTO DEL PROTOCOLO**

El protocolo SIP permite el establecimiento de sesiones multimedia entre dos o más usuarios. Para hacerlo se vale del intercambio de mensajes entre las partes que quieren comunicarse.

### **2.4.1.1 AGENTES DE USUARIO**

Los usuarios, que pueden ser seres humanos o aplicaciones de software, utilizan para establecer sesiones lo que el protocolo SIP denomina Agentes de usuario. Estos no son más que los puntos extremos del protocolo, es decir son los que emiten y consumen los mensajes del protocolo SIP. Un videoteléfono, un teléfono, un cliente de software (SoftPhone) y cualquier otro dispositivo similar es para el protocolo SIP un agente de usuario. El protocolo SIP no se ocupa de la interfaz de estos dispositivos con el usuario final, sólo se interesa por los mensajes que estos generan y cómo se comportan al recibir determinados mensajes.

Los agentes de usuario se comportan como clientes (UAC: User Agent Clients) y como servidores (UAS: User Agent Servers). Son UAC cuando realizan una petición y son UAS cuando la reciben. Por esto los agentes de usuario deben implementar un UAC y un UAS.

Además de los agentes de usuario existen otras entidades que intervienen en el protocolo, estos son los Servidores de Registro, los Proxy y los Redirectores. A continuación se describe su finalidad.

#### **2.4.1.2 SERVIDORES DE REGISTRO**

El protocolo SIP permite establecer la ubicación física de un usuario determinado, esto es, en qué punto de la red está conectado. Para ello se vale del mecanismo de registro. Este mecanismo funciona como sigue:

Cada usuario tiene una dirección lógica que es invariable respecto de la ubicación física del usuario. Una dirección lógica del protocolo SIP es de la forma usuario@dominio es decir tiene la misma forma que una dirección de correo electrónico. La dirección física denominada "dirección de contacto" es dependiente del lugar en donde el usuario está conectado de su dirección IP. Cuando un usuario inicializa su terminal, por ejemplo conectando su teléfono o abriendo su software de

telefonía SIP, el agente de usuario SIP que reside en dicho terminal envía una petición con el método REGISTER a un Servidor de Registro, informando a qué dirección física debe asociarse la dirección lógica del usuario. El servidor de registro realiza entonces dicha asociación (denominada binding). Esta asociación tiene un período de vigencia y si no es renovada, caduca. También puede terminarse mediante una cancelación del registro. La forma en que dicha asociación es almacenada en la red no es determinada por el protocolo SIP, pero es vital que los elementos de la red SIP accedan a dicha información. [4]

## 2.5 IAX2

Inter-Asterisk eXchange protocol es uno de los protocolos utilizado por Asterisk, un servidor, PBX central telefónica de código abierto patrocinado por Digium. Es utilizado para manejar conexiones VoIP entre servidores Asterisk, y entre servidores y clientes que también utilizan protocolo IAX.

El protocolo IAX ahora se refiere generalmente al IAX2, la segunda versión del protocolo IAX. El protocolo original ha quedado obsoleto en favor de IAX2.

El protocolo IAX2 fue creado por Mark Spencer para la señalización de VoIP en Asterisk. El protocolo crea sesiones internas y dichas sesiones pueden utilizar cualquier códec que pueda transmitir voz o vídeo. El IAX esencialmente provee control y transmisión de flujos de datos multimedia sobre redes IP. IAX es extremadamente flexible y puede ser utilizado con cualquier tipo de dato incluido vídeo.

El diseño de IAX se basó en muchos estándares de transmisión de datos, incluidos SIP, el cual es el más común actualmente.

### **2.5.1 OBJETIVOS DEL IAX**

El principal objetivo de IAX ha sido minimizar el ancho de banda utilizado en la transmisión de voz y vídeo a través de la red IP, con particular atención al control y a las llamadas de voz y proveyendo un soporte nativo para ser transparente a NAT. La estructura básica de IAX se fundamenta en la multiplexación de la señalización y del flujo de datos sobre un simple puerto UDP entre dos sistemas.

IAX es un protocolo binario y está diseñado y organizado de manera que reduce la carga en flujos de datos de voz. El ancho de banda para algunas aplicaciones se sacrifica en favor del ancho de banda para VoIP.

### **2.5.2 PROPIEDADES BÁSICAS**

IAX2 es robusto, lleno de novedades y muy simple en comparación con otros protocolos. Permite manejar una gran cantidad de códec y un gran número de streams, lo que significa que puede ser utilizado para transportar virtualmente

cualquier tipo de dato. Esta capacidad lo hace muy útil para realizar video conferencias o realizar presentaciones remotas.

IAX2 utiliza un único puerto UDP, generalmente el 4569, para comunicaciones entre terminales VoIP, para señalización y datos. El tráfico de voz es transmitido in-band, lo que hace a IAX2 un protocolo casi transparente a los cortafuegos y realmente eficaz para trabajar dentro de redes internas. En esto se diferencia de SIP, que utiliza una cadena RTPC out-of-band para entregar la información.

IAX2 soporta Trunking red, donde un simple enlace permite enviar datos y señalización por múltiples canales. Cuando se realiza Trunking, los datos de múltiples llamadas son manejados en un único conjunto de paquetes, lo que significa que un datagrama IP puede entregar información para más llamadas sin crear latencia adicional. Esto es una gran ventaja para los usuarios de VoIP, donde las cabeceras IP son un gran porcentaje del ancho de banda utilizado. [5]

## 2.6 H.323

H.323 es una recomendación del ITU-T International Telecommunication Union, que define los protocolos para proveer sesiones de comunicación audiovisual sobre paquetes de red. A partir del año 2000 se encuentra implementada por varias aplicaciones de Internet que funcionan en tiempo real como Microsoft Netmeeting y Ekiga. Anteriormente conocido como Gnome Meeting, el cual utiliza la implementación OpenH323. Es una parte de la serie de protocolos H.32x, los cuales también dirigen las comunicaciones sobre RDSI, RTC o SS7.

H.323 es utilizado comúnmente para Voz sobre IP, Telefonía de Internet o Telefonía IP y para videoconferencia basada en IP. Es un conjunto de normas ITU para comunicaciones multimedia que hacen referencia a los terminales, equipos y servicios estableciendo una señalización en redes IP. No garantiza una calidad de servicio, y en el transporte de datos puede, o no, ser fiable; en el caso de voz o vídeo, nunca es fiable. Además, es independiente de la topología de la red y admite rutas, permitiendo usar más de un canal de cada tipo, voz, vídeo y datos al mismo tiempo.

### **2.6.1 LA TOPOLOGÍA CLÁSICA DE UNA RED BASADA EN H-323**

- Portero: realiza el control de llamada en una zona. Es opcional pero su uso está recomendado, de modo que si existe, su uso será obligatorio. Traduce direcciones, ofrece servicio de directorio, control de admisión de terminales, control de consumo de recursos y procesa la autorización de llamadas, así como también puede encaminar la señalización.
- Pasarela: es el acceso a otras redes, de modo que realiza funciones de codificación y traducción de señalización.
- MCU: soporte multi-conferencia. Se encarga de la negociación de capacidades.[6]

## **2.7 MGCP**

MGCP es un protocolo de control de dispositivos, donde un Gateway esclavo (MG, Media Gateway) es controlado por un maestro (MGC, Media Gateway Controller, también llamado Call Agent).

MGCP, Media Gateway Control Protocol, es un protocolo interno de VoIP cuya arquitectura se diferencia del resto de los protocolos VoIP por ser del tipo cliente – servidor. MGCP está definido informalmente en la RFC 3435, y aunque no ostenta el rango de estándar, su sucesor, Megaco está aceptado y definido como una recomendación en la RFC 3015.

Está compuesto por:

- Un MGC, Media Gateway Controller
- Uno o más MG, Media Gateway
- Uno o más SG, Signaling Gateway.

Un Gateway tradicional, cumple con la función de ofrecer conectividad y traducción entre dos redes diferentes e incompatibles como lo son las de Conmutación de Paquetes y las de Conmutación de Circuitos. En esta función, el Gateway realiza la conversión del flujo de datos, y además realiza también la conversión de la señalización, bidireccionalmente.

MGCP separa conceptualmente estas funciones en los tres elementos previamente señalados. Así, la conversión del contenido multimedia es realizada por el MG, el control de la señalización del lado IP es realizada por el MGC, y el control de la señalización del lado de la red de Conmutación de Circuitos es realizada por el SG.

MGCP introduce esta división en los roles con la intención de aliviar a la entidad encargada de transformar el audio para ambos lados, de las tareas de señalización, concentrando en el MGC el procesamiento de la señalización. [7]

## **2.8 UIT-T**

Define la Red Digital de Servicios Integrados, RDSI o ISDN en inglés, como una red que procede por evolución de la Red Digital Integrada RDI y que facilita conexiones digitales extremo a extremo para proporcionar una amplia gama de servicios, tanto de voz como de otros tipos, y a la que los usuarios acceden a través de un conjunto de interfaces normalizados.

RDSI es una red que procede por evolución de la red telefónica existente, que al ofrecer conexiones digitales de extremo a extremo permite la integración de multitud de servicios en un único acceso, independientemente de la naturaleza de la información a transmitir y del equipo terminal que la genere.

En el estudio de la RDSI se han definido unos llamados puntos de referencia que sirven para delimitar cada elemento de la red. Estos son llamados R, S, T, U y V, siendo el U el correspondiente al par de hilos de cobre del bucle telefónico entre la central y el domicilio del usuario. [8]

## **2.9 GNU GPL**

La Licencia Pública General de GNU o más conocida por su nombre en inglés GNU GPL, es una licencia creada por la Fundación de Software Libre en 1989, y está orientada principalmente a proteger la libre distribución, modificación y uso de software. Su propósito es declarar que el software cubierto por esta licencia es software libre y protegerlo de intentos de apropiación que restrinjan esas libertades a los usuarios. [9]

## 2.10 GNU/LINUX

El nombre GNU, GNU's Not Unix (GNU no es Unix), viene de las herramientas básicas de sistema operativo creadas por el proyecto GNU, iniciado por Richard Stallman en 1983 y mantenido por la FSF. El nombre Linux viene del núcleo Linux, inicialmente escrito por Linus Torvalds en 1991.

La contribución de GNU es la razón por la que existe controversia a la hora de utilizar Linux o GNU/Linux para referirse al sistema operativo formado por las herramientas de GNU y el núcleo Linux en su conjunto.

Es uno de los términos empleados para referirse a la combinación del núcleo o Kernel libre similar a Unix denominado Linux, que es usado con herramientas de sistema GNU. Su desarrollo es uno de los ejemplos más prominentes de software libre; todo su código fuente puede ser utilizado, modificado y redistribuido libremente por cualquiera bajo los términos de la GPL Licencia Pública General de GNU y otra serie de licencias libres. [10]

## **2.11 IVR**

Son las siglas de Interactive Voice Response, que se traduce del inglés como Respuesta de Voz Interactiva. También se utiliza el término VRU que significa Voice Response Unit.

Consiste en un sistema telefónico que es capaz de recibir una llamada e interactuar con el humano a través de grabaciones de voz y el reconocimiento de respuestas simples, como "sí", "no" u otras. Es un sistema automatizado de respuesta interactiva, orientado a entregar y/o capturar información a través del teléfono, permitiendo el acceso a servicios de información u otras operaciones.

### **2.11.1 SERVICIOS**

El IVR Interactive Voice Response es comúnmente implementado en empresas o entidades que reciben grandes cantidades de llamadas, a fin de reducir la necesidad de personal y los costos que el servicio ofrecido representen para dicha entidad. Entre otras, podemos mencionar a las bancas telefónicas.

Las empresas suelen usar la tecnología de IVR para encaminar una llamada entrante hacia un departamento u otro, sin la necesidad de intervención humana, así reduciendo el tiempo de espera de sus clientes.

En los centros de atención telefónico al cliente, se usan los IVR's para guiar el llamante hacia los agentes con mayor conocimiento de una materia específica, reduciendo así el tiempo de la llamada y evitando la necesidad de hacer transferencias entre agentes.

Se los está implementando también en empresas de taxis, el cual mediante la identificación del número que llama, se puede saber dónde se encuentra el pasajero y generar el viaje en forma rápida sin la intervención de un telefonista humano. Puede combinarse con SMS para prestar cualquier clase de servicio: tele votación, encuestas, sorteos, acceso a bases de datos, servicios informativos, etc.

### **2.11.2 ¿CÓMO FUNCIONA?**

El usuario realiza una llamada a un número de teléfono, el sistema de audio respuesta contesta la llamada y le presenta al usuario una serie de acciones a realizar, esto se hace mediante mensajes, menús de opciones previamente grabados en archivos de audio, por ejemplo "Presione uno para ventas, dos para administración". El usuario elige la opción a realizar introduciendo un número en el teclado del teléfono y navega por los diferentes menús hasta encontrar la información solicitada o que el sistema encamine la llamada al destinatario elegido. [11]

## **2.12 NAGIOS**

### **2.12.1 HISTORIA DE NAGIOS**

Nagios fue llamado originalmente Netsaint, nombre que se debió cambiar por coincidencia con otra marca comercial, fue creado y es actualmente mantenido por Ethan Galstad, junto con un grupo de desarrolladores.

Nagios es un acrónimo recursivo que significa “Nagios Ain’t Gonna Insist On Sainthood” en otras palabras “Nagios no insistirá con la santidad” Esto hace referencia al antiguo Software bajo el de “Netsaint”, la palabra Nagios se divide en dos palabras, Net y Hagios también deletreado agios, que significa Santo en antiguo y moderno Griego.

Nagios fue originalmente diseñado para ser ejecutado en GNU/Linux, pero también se ejecuta bien en variantes de Unix.

### **2.12.2 ¿QUE ES NAGIOS?**

Nagios es un sistema de monitorización de redes de código abierto, que vigila tanto hardware como software, alertando cuando el comportamiento de los mismos no sea el deseado.

Se trata de un software que proporciona una gran versatilidad para consultar prácticamente cualquier parámetro de interés de un sistema, y genera alertas, que pueden ser recibidas por los Administradores

correspondientes mediante correo electrónico o mensajes SMS.

Nagios está licenciado bajo la GNU General Public License Versión 2 publicada por la Fundación de Software Libre.

### **2.12.3 NAGIOS A NIVEL DE EMPRESA**

Nagios es un sistema de vigilancia de gran alcance que ayuda a las organizaciones, a identificar y resolver los problemas de infraestructura de IT antes de que afecten los procesos de negocio críticos.

Diseñado con la escalabilidad y la flexibilidad en mente, Nagios le da la tranquilidad de saber que sus procesos de negocio de su organización no se verá afectada por los cortes de desconocidos.

Nagios es una potente herramienta que le proporciona la conciencia inmediata de la misión de su organización, la infraestructura de IT críticos. Capaz de detectar, reparar y

mitigar los problemas de futuro antes de que afecten a los usuarios finales y clientes.

#### **2.12.4 CARACTERÍSTICAS**

A continuación se detallan las principales características de Nagios.

- Monitoreo de servicios de red SMTP, POP3, HTTP, NTTP, ICMP, SNMP.
- Monitoreo de los recursos de un host, carga del procesador, uso de los discos, logs del sistema en varios sistemas operativos, incluso Microsoft Windows con el plugin NRPE\_NT.
- Monitoreo remoto, a través de túneles SSL cifrados o SSH.
- Se puede programar plugins específicos para monitorizar diversos sistemas, tal es el caso de Asterisk.

- Diseño simple de plugins, que permiten a los usuarios desarrollar sus propios chequeos de servicios dependiendo de sus necesidades, usando sus herramientas preferidas Bash, C++, Perl, Ruby, Python, PHP, C#, etc.
- Chequeo de servicios paralizados.
- Posibilidad de definir la jerarquía de la red, permitiendo distinguir entre host caídos y host inaccesibles.
- Notificaciones a los contactos cuando ocurren problemas en servicios o hosts, así como cuando son resueltos, vía email, SMS, o cualquier método definido por el usuario junto con su correspondiente plugin).
- Posibilidad de definir manejadores de eventos que ejecuten al ocurrir un evento de un servicio o host para resoluciones de problemas proactivas.

- Soporte para implementar host de monitores redundantes.
- Interfaz web opcional, para observar el estado de la red actual, notificaciones, historial de problemas, archivos de logs, etc.

Nagios provee también una muy buena e intuitiva interfaz web la cual presenta gráficamente la dependencia de los host de la red, de donde a donde están conectados y hasta su ubicación física, esta interfaz también se puede combinar con un módulo VRML en las que se puede visualizar a los host de manera 3D.

Permite una vista del estado general de todos los hosts y el estado detallado de cada uno de los servicio. Gracias a su sistema de Plugins aumenta la posibilidad de monitorear o incluso tomar medidas pro activas como reiniciar servicios o reiniciar las maquinas una vez se detecta y establece que hay un problema. [12]

## **2.13 FUNCIONAMIENTO DE NAGIOS**

### **2.13.1 MONITOREO**

Los administradores de red pueden configurar Nagios para supervisar los componentes de infraestructura críticos, incluidos los parámetros de sistema, protocolos de red, aplicaciones, servicios, servidores y la infraestructura de red.

### **2.13.2 ALERTAS**

Nagios envía alertas cuando los componentes de la infraestructura fallan y cuando se recuperan, enviando a los administradores notificaciones las cuales pueden ser entregadas por correo electrónico, SMS, o script personalizado.

### **2.13.3 RESPUESTA**

Una vez que se reciben las notificaciones los administradores de red pueden comenzar a resolver los problemas de seguridad inmediatamente.

### **2.13.4 PRESENTACIÓN DE INFORMES**

Los informes proporcionan un registro histórico de los fallos, eventos, notificaciones, y la respuesta de alerta para su posterior revisión. Los informes ayudan a la resolución de posibles fallos y para dar mayor disponibilidad a los servicios.

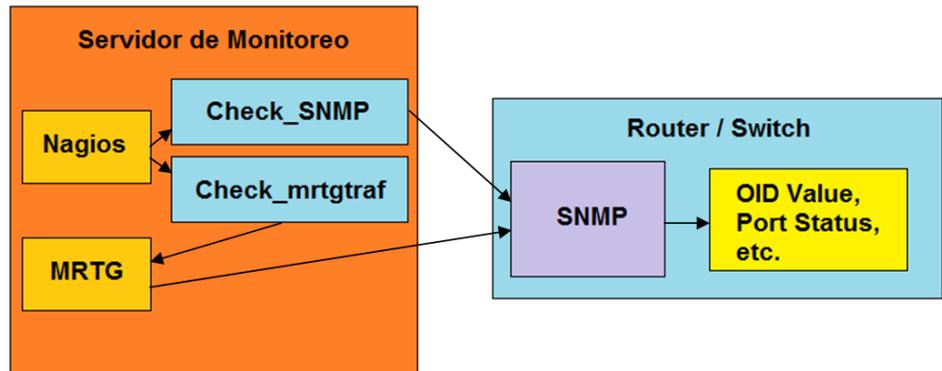
### **2.13.5 MANTENIMIENTO**

El tiempo de inactividad programado impide alertas de mantenimiento programado y ventanas durante la actualización.

### 2.13.6 PLANIFICACIÓN

Tendencias de la capacidad de planificación y los gráficos e informes le permiten determinar las mejoras de la infraestructura necesaria antes de que ocurra algún fallo. [13]

Un ejemplo de cómo funciona Nagios para monitorear un Router o un Switch se ve en la figura 2.3.



**Figura. 2.3 Funcionamiento de Nagios**

Nagios ejecuta el comando `check_snmp`, al cual se le pasan los parámetros definidos en la directiva “define host”. El comando a su vez lanza una petición SNMP para obtener los valores que nos interesan. El router o switch envía la información de vuelta para que Nagios la procese y actúe conforme se tiene definido. En este

mismo ejemplo, no sólo se hace la consulta al Switch, sino que además se consulta al MRTG que permite a los usuarios ver la carga de una red de manera gráfica para procesar los datos históricos del ancho de banda de las interfaces. [14]

## **2.14 IMPORTANCIA DE LA MONITORIZACIÓN DEL SERVIDOR**

### **ASTERISK**

Al realizar una implementación estable de un IP-PBX basado en Asterisk no termina cuando realizamos la primera llamada, a estos equipos se les debe de dar servicio constantemente para evitar problemas futuros, este servicio consiste en la instalación de actualizaciones, mantenimiento en óptimas condiciones del Hardware, limpieza del equipo y monitoreo del mismo.

El monitorizar nuestros servidores nos permite adelantarnos a futuras fallas y estar atentos a niveles críticos que estos puedan alcanzar, tales como gran consumo de RAM, poco espacio en disco duro, desconexión generalizada de canales SIP, estado de canales, uso de partición SWAP

y muchas cosas más que podrían ocurrir si no tenemos un buen sistema de monitoreo.

Si bien Nagios no está diseñado nativamente para el monitoreo de Asterisk, existen muchas maneras para poder hacerlo, una de ellas y la que vamos a usar es NET-SNMP que se encuentran disponible en la plataforma Linux para su respectiva comunicación entre los dos servicios, haciendo de este sistema una herramienta más interesante ya que no solo nos permitirá medir niveles de disco duro y RAM, sino también monitorizar nuestras extensiones SIP, IAX, o cualquier otro tipo de canal o servicio que nos ayudara a tener nuestro IP-PBX bajo control.

# CAPITULO 3

## 3. IMPLEMENTACIÓN Y CONFIGURACIÓN

### 1.1 COMPONENTES DEL PROYECTO

Para la implementación del proyecto utilizaremos un servidor el cual tendrá instalado Asterisk, y otro el cual tendrá instalado el sistema de monitoreo Nagios. Y varios softphones para realizar las llamadas por los canales SIP e IAX.

#### 1.1.1 SERVIDOR NAGIOS

➤ Hardware

- Intel Core 2 duo
- Memoria Ram 2 Gb
- Disco duro de 120 Gb
- Adaptador de red Fast Ethernet Netlink (TM)

➤ Software

- Centos 5.4

## 1.1.2 SERVIDOR ASTERISK

### ➤ Hardware

- Intel Celeron M
- Memoria Ram 512 Mb
- Disco duro de 120 Gb
- Adaptador de red Broadcom

### ➤ Software

- Centos 5.4

### 1.1.3 SELECCIÓN DE SOFTWARE DE MONITOREO

Para la elección de Nagios como nuestro software de monitoreo, se tuvo que hacer un análisis según como lo indica la figura 3.1, concluyendo que Nagios sea el que cumple con las mejores características para este proyecto y en general.

DESCRIPCIONES	VQMANAGER	NAGIOS	HOBBIT	MUNIN	MONIT
Interfaz Web	X	X	X	X	X
Alertas y Notificaciones		X	X		X
Basta información en la red		X			X
Flexible Plugins	X	X	X	X	
Escalable y Robusto		X	X		
Complejidad en instalación y configuración		X			
Graficas estadísticas	X	X	X	X	X
Reportes	X	X			
Autenticación de usuarios		X			
Usado para redes locales	X	X	X	X	X
Usados para redes empresariales		X	X		
Licencia Libre		X	X	X	X
Versatilidad		X	X		
Potencia		X			
Fácil de usar	X	X	X	X	X
Orientado a VOIP	X				X

**Figura 3.1.- Elección de software Nagios**

## **1.2 INSTALACIÓN DEL SISTEMA NAGIOS**

El procedimiento de instalación de Nagios fue diseñado paso a paso durante el desarrollo de este proyecto.

### **1.2.1 PROCEDIMIENTO DE INSTALACIÓN**

A continuación se define el procedimiento de instalación del software Nagios.

- I.** Creamos el usuario nagios.
- II.** Generamos el grupo nagcmd.
- III.** A los usuarios nagios y apache los agregamos al grupo nagcmd.
- IV.** Generamos una carpeta llamada nagios.
- V.** Dentro de la carpeta generada anteriormente descargamos los paquetes NagiosCore 3.2.0, NagiosPlugins 1.4.14, y NagiosNrpe 2.12.
- VI.** Instalar y a compilar Nagios.
- VII.** Instalar y compilar Nagios Plugins.
- VIII.** Configuración inicial de Nagios.

Para información más detallada de cada uno de estos pasos ver el anexo 1.

### **1.3 INSTALACIÓN DE NRPE EN LOS SERVIDORES**

Este pequeño demonio te permite ejecutar plugins locales en equipos remotos de equipos Linux, como es esto, el plugincheck\_nrpe es llamado por Nagios y hace las peticiones de los plugins hacia el equipo remoto, requiere que NRPE esté siendo ejecutado en ese equipo como un stand alone daemon o como un servicio en xinetd, también soporta encriptación nativa de SSL/TLS.

#### **1.3.1 INSTALACIÓN DE NRPE EN EL SERVIDOR ASTERISK**

A continuación definimos el procedimiento de instalación de NRPE en el Servidor Asterisk.

- I.** Instalamos Openssl.
- II.** Generamos el usuario nagios con su credencial.
- III.** Descargamos y descomprimos Nagios Plugins.
- IV.** Compilamos e instalamos Nagios Plugins.
- V.** Cambiamos permisos.
- VI.** Instalamos Xinetd.

- VII. Descargamos y descomprimos el demonio NRPE.
- VIII. Compilamos e instalamos NRPE.
- IX. Instalamos el demonio NRPE como un servicio en Xinetd.
- X. Editamos el archivo `/etc/xinetd.d/nrpe`.
- XI. Agregamos esta línea `"nrpe 5666/tcp #NRPE"` al archivo `/etc/services`.
- XII. Reiniciamos Xinetd.
- XIII. Probamos el demonio NRPE localmente.

### **1.3.2 INSTALACIÓN DE NRPE EN EL SERVIDOR NAGIOS**

A continuación se describe el procedimiento de instalación de NRPE en el Servidor Nagios.

- I. Descargamos y descomprimos el demonio NRPE.
- II. Configuramos y compilamos en archivo NRPE.
- III. Probamos la comunicación con el demonio NRPE de nuestro Servidor Asterisk.

Para información más detallada de cada uno de estos pasos ver el anexo 2.

## **1.4 INSTALACIÓN DE SNMP EN LOS SERVIDORES**

Este servicio facilita el intercambio de información para administración de dispositivos en la red. Es parte de la familia de protocolos TCP/IP. SNMP permite a los administradores supervisar el funcionamiento de la red, buscar y resolver sus problemas, y planear su crecimiento.

### **1.4.1 INSTALACIÓN DE SNMP EN EL SERVIDOR ASTERISK**

A continuación se define el procedimiento de instalación de SNMP en el servidor Asterisk.

- I.** Instalamos los paquetes que se necesitan para la instalación del módulo `res_snmp`.
- II.** Nos ubicamos en la carpeta de las fuentes de Asterisk y volvemos a compilar y configurar Asterisk asociado con `snmp`.
- III.** Verificamos si Asterisk cuenta con el módulo `res_snmp`.
- IV.** Ahora configuramos SNMP de modo que pueda interactuar con Asterisk editando el archivo `snmpd.conf`.
- V.** Cambiamos los permisos del `agentX`.
- VI.** Configuramos el archivo `res_snmp.conf`.

- VII.** Copiamos los OID de Asterisk en la carpeta snmp.
- VIII.** Reiniciamos SNMP.
- IX.** Exportamos los archivos MIB del Asterisk al SNMP.
- X.** Reiniciamos Asterisk.
- XI.** Probamos la instalación con el comando `snmpwalk -OT -c 821514821514 -v 2c localhost .1.3.6.1.4.1.22736`.

Los IOD son usados para agregarlos a los servicios de Asterisk que van a ser monitoreados respectivamente.

#### **1.4.2 INSTALACIÓN DE SNMP EN EL SERVIDOR NAGIOS**

A continuación se define el procedimiento de instalación de SNMP en el servidor Nagios.

- I.** Editar el archivo  
`/usr/local/nagios/etc/objects/commands.cfg`.
- II.** En Asterisk editamos el archivo  
`/etc/asterisk/manager.conf`.
- III.** Actualizamos la configuración de Asterisk.
- IV.** Modificamos el archivo `localhost.cfg` del Servidor Nagios para agregar los servicio con los parámetros OID.

**V.** Reiniciamos Nagios.

Para información más detallada de cada uno de estos pasos ver el anexo 3.

## **1.5 INSTALACIÓN DEL MOTOR GRÁFICO PNP4NAGIOS**

A continuación se define el procedimiento de instalación de PNP4NAGIOS en el Servidor Nagios.

- I.** Instalar PHP 5.2.
- II.** Descargamos los paquetes rrdtool, php-gd y pnp4nagios.
- III.** Compilamos e instalamos pnp4nagios-0.6.11.
- IV.** Configuramos pnp4nagios-0.6.11.
- V.** Modificamos el archivo localhost.cfg para agregar los servicios gráficos de pnp4nagios.
- VI.** Activar el proceso NPCD el cual nos permite crear las gráficas automáticamente.
- VII.** Modificamos el fichero templates.cfg agregando unas sentencias, para poder pre visualizar los gráficos en forma de miniaturas.

Para información más detallada de cada uno de estos pasos ver el anexo 4.

## 1.6 INSTALACIÓN DEL SERVIDOR DE CORREO SEND MAIL

Nagios utiliza un servidor de correo para poder enviar sus notificaciones a cualquier cuenta de correo por parte de la administración, utilizando los más comunes como Sendmail, Postfix entre otros.

Para poder instalar Sendmail utilizamos los siguientes comandos:

- `yum -y install sendmail sendmail-cf dovecot m4 make`

Para poder instalar Postfix utilizamos los siguientes comandos:

- `yum -y install postfix postfix-pflogsumm dovecot cyrus-sasl cyrus-sasl-devel cyrus-sasl-gssapi cyrus-sasl-md5 cyrus-sasl-ntlm cyrus-sasl-plain cyrus-sasl-sql`

Cabe recalcar que estos servidores no pueden estar trabajando al mismo tiempo por lo que provoca conflicto entre las notificaciones de Nagios,

por lo tanto se debe eliminar uno de ellos, porque ambos vienen instalados por defecto:

- `yum remove sendmail`
- `yum remove postfix`

# CAPITULO 4

## 2. FUNCIONAMIENTOS Y PRUEBAS DEL PROYECTO

### 4.1 INTERFAZ WEB DE NAGIOS

Nagios cuenta una interface muy amigable para el usuario del sistema ya que permite visualizar información del monitoreo desde diferentes perspectivas a través de sus diferentes tipos de reporte, en entre las principales características que se pueden visualizar hemos destacado las siguientes:

- Reporte resumido del estado de un grupo de host
- Esquema o mapa de todos los host que se están monitoreando.
- Reporte específico de los servicios de un host.
- Reportes Gráficos de los servicios de un host.
- Presentación de alertas a través del cambio de colores en los servicios verde que quiere decir “OK o activo” rojo cuando esta “inactivo” y naranja cuando no se reconoce.
- Notificaciones.

Los servicios que se visualizarán en la interface web se muestran en la figura 4.1:

- Llamadas activas.
- Llamadas procesadas.
- Canales en uso.
- Canales IAX.
- Canales SIP.
- Servicio de SSH.
- Número total de procesos.
- Carga del procesador.

Host ↑	Service ↑	Status ↑	Last Check ↑	Duration ↑	Attempt ↑	Status Information
asterisk	<a href="#">Asterisk Active Calls</a>	OK	03-03-2011 16:16:23	0d 0h 49m 18s	1/4	Current Active Calls OK - 0
	<a href="#">Asterisk Calls Processed</a>	OK	03-03-2011 16:16:23	0d 0h 49m 18s	1/4	Current Calls Processed OK - 5
	<a href="#">Asterisk Channels In Use</a>	OK	03-03-2011 16:16:23	0d 0h 49m 18s	1/4	Current Channels Used OK - 0
	<a href="#">IAX2</a>	OK	03-03-2011 16:16:23	0d 0h 49m 18s	1/4	Canales IAX2 Activos OK - 0
	<a href="#">SIP</a>	OK	03-03-2011 16:16:23	0d 0h 49m 18s	1/4	Canales SIP Activos OK - 0
	<a href="#">SSH</a>	OK	03-03-2011 16:16:23	0d 0h 49m 14s	1/4	SSH OK - OpenSSH_4.3 (protocol 2.0)
	<a href="#">Total Processes</a>	OK	03-03-2011 16:16:23	33d 15h 20m 56s	1/4	PROCS OK: 59 processes with STATE = RSZDT
nagios	<a href="#">Current Load</a>	OK	03-03-2011 16:16:23	0d 0h 31m 32s	1/4	OK - load average: 0.58, 0.63, 0.86
	<a href="#">SSH</a>	OK	03-03-2011 16:16:23	18d 18h 53m 21s	1/4	SSH OK - OpenSSH_4.3 (protocol 2.0)
	<a href="#">Total Processes</a>	OK	03-03-2011 16:16:23	18d 18h 53m 21s	1/4	PROCS OK: 56 processes with STATE = RSZDT

**Figura 4.1 Servicios del Servidor Nagios**

## 4.2 DESCRIPCIÓN DEL FUNCIONAMIENTO DE NAGIOS.

Una vez que ya se hayan implementado todos los pasos para el funcionamiento del sistema Nagios con todas las características y servicios requeridos, procedemos a entrar a la interfaz web de Nagios.

Veremos la interfaz de Nagios con todos los servicios requeridos para este proyecto, si el servidor Asterisk no ha sido aún levantado los servicios saldrá en un estado de crítico como se muestran en la figura 4.2.

Host	Service	Status	Last Check	Duration	Attempt	Status Information
asterisk	Asterisk Active Calls	CRITICAL	03-23-2011 22:32:37	20d 5h 46m 5s	1/4	(Service Check Timed Out)
	Asterisk Calls Processed	CRITICAL	03-23-2011 22:32:36	20d 5h 46m 5s	1/4	(Service Check Timed Out)
	Asterisk Channels In Use	CRITICAL	03-23-2011 22:32:37	20d 5h 46m 5s	1/4	(Service Check Timed Out)
	IAX2	CRITICAL	03-23-2011 22:32:37	20d 5h 46m 5s	1/4	(Service Check Timed Out)
	SIP	CRITICAL	03-23-2011 22:32:37	20d 5h 46m 5s	1/4	(Service Check Timed Out)
	SSH	CRITICAL	03-23-2011 22:32:37	20d 5h 46m 5s	1/4	(Service Check Timed Out)
	Total Processes	CRITICAL	03-23-2011 22:32:40	53d 21h 37m 14s	1/4	(Service Check Timed Out)
nagios	Current Load	OK	03-23-2011 22:32:40	20d 6h 47m 50s	1/4	OK - load average: 0.51, 0.29, 0.20
	SSH	OK	03-23-2011 22:32:40	33d 1h 9m 39s	1/4	SSH OK - OpenSSH_4.3 (protocol 2.0)
	Total Processes	OK	03-23-2011 22:32:40	33d 1h 9m 39s	1/4	PROCS OK: 74 processes with STATE = RSZDT

**Figura 4.2 Estado de fallo del Servidor Asterisk**

Cuando ya el servidor Asterisk sea levantado los servicios que están siendo monitoreados pasaran a un estado de recuperación, como se muestra en la figura 4.3.

Host	Service	Status	Last Check	Duration	Attempt	Status Information
asterisk	Asterisk Active Calls	OK	03-03-2011 16:16:23	0d 0h 49m 18s	1/4	Current Active Calls OK - 0
	Asterisk Calls Processed	OK	03-03-2011 16:16:23	0d 0h 49m 18s	1/4	Current Calls Processed OK - 5
	Asterisk Channels In Use	OK	03-03-2011 16:16:23	0d 0h 49m 18s	1/4	Current Channels Used OK - 0
	IAX2	OK	03-03-2011 16:16:23	0d 0h 49m 18s	1/4	Canales IAX2 Activos OK - 0
	SIP	OK	03-03-2011 16:16:23	0d 0h 49m 18s	1/4	Canales SIP Activos OK - 0
	SSH	OK	03-03-2011 16:16:23	0d 0h 49m 14s	1/4	SSH OK - OpenSSH_4.3 (protocol 2.0)
	Total Processes	OK	03-03-2011 16:16:23	33d 15h 20m 56s	1/4	PROCS OK: 59 processes with STATE = RSZDT
nagios	Current Load	OK	03-03-2011 16:16:23	0d 0h 31m 32s	1/4	OK - load average: 0.58, 0.63, 0.86
	SSH	OK	03-03-2011 16:16:23	18d 18h 53m 21s	1/4	SSH OK - OpenSSH_4.3 (protocol 2.0)
	Total Processes	OK	03-03-2011 16:16:23	18d 18h 53m 21s	1/4	PROCS OK: 56 processes with STATE = RSZDT

**Figura 4.3 Estado de recuperación del Servidor Asterisk**

Si algún servicio del servidor Asterisk que está siendo monitoreado se cae, automáticamente enviarán notificaciones al mail del administrador. A continuación se describe el contenido de las notificaciones que se muestran en la figura 4.4 y figura 4.5.

- Tipo de notificación: Problema o Recuperación.
- Servicio: Servicios que se monitorean con Nagios.
- Host: Nombre del servidor.
- Dirección IP
- Estado del servicio o host y la Fecha/Hora del incidente.

```
** PROBLEM Service Alert: servidor asterisk/SIP is CRITICAL **
☆ nagios@servnagios.servnagios.com para usuario
***** Nagios *****

Notification Type: PROBLEM

Service: SIP
Host: servidor asterisk
Address: 200.126.12.119
State: CRITICAL

Date/Time: Thu Mar 3 16:47:05 ECT 2011

Additional Info:
(Service Check Timed Out)
```

**Figura 4.4.- Notificación servicio Asterisk caído**

```
** RECOVERY Service Alert: servidor asterisk/SIP is OK **
☆ nagios@servnagios.servnagios.com para usuario
***** Nagios *****

Notification Type: RECOVERY

Service: SIP
Host: servidor asterisk
Address: 200.126.12.119
State: OK

Date/Time: Thu Mar 3 16:28:20 ECT 2011

Additional Info:
Canales SIP Activos OK - 0
```

**Figura 4.5.- Notificación servicio Asterisk recuperado**

### 4.3 PRUEBAS REALIZADAS

Con el proyecto ya finalizado con todos sus requerimientos activos y funcionando se procedió a realizar algunas pruebas para verificar su correcto funcionamiento.

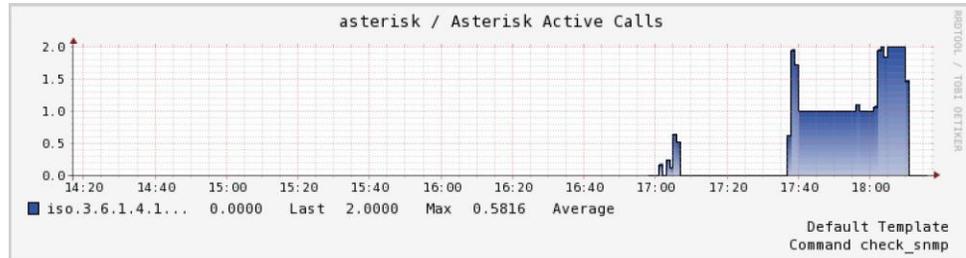
#### 4.3.1 MONITOREO DE LAS LLAMADAS ACTIVAS

Para esta prueba tuvimos que hacer llamadas desde los softphones para usar los canales IAX y SIP, y notamos que cuando se hacían las llamadas, el monitoreo se iniciaba en tiempo real como se muestra en la figura 4.6.

Host	Service	Status	Last Check	Duration	Attempt	Status Information
asterisk	Asterisk Active Calls	OK	03-03-2011 16:00:23	0d 0h 33m 19s	1/4	Current Active Calls OK - 1
	Asterisk Calls Processed	OK	03-03-2011 16:00:23	0d 0h 33m 19s	1/4	Current Calls Processed OK - 1
	Asterisk Channels In Use	OK	03-03-2011 16:00:23	0d 0h 33m 19s	1/4	Current Channels Used OK - 2
	IAX2	OK	03-03-2011 16:00:23	0d 0h 33m 19s	1/4	Canales IAX2 Activos OK - 0
	SIP	OK	03-03-2011 16:00:23	0d 0h 33m 19s	1/4	Canales SIP Activos OK - 2
	SSH	OK	03-03-2011 16:00:23	0d 0h 33m 15s	1/4	SSH OK - OpenSSH_4.3 (protocol 2.0)
	Total Processes	OK	03-03-2011 16:00:23	33d 15h 4m 57s	1/4	PROCS OK: 57 processes with STATE = RSZDT
revoce	Current Load	OK	03-03-2011 16:00:23	0d 0h 15m 33s	1/4	OK - load average: 0.44, 0.60, 1.15
	SSH	OK	03-03-2011 16:00:23	18d 18h 37m 22s	1/4	SSH OK - OpenSSH_4.3 (protocol 2.0)
	Total Processes	OK	03-03-2011 16:00:23	18d 18h 37m 22s	1/4	PROCS OK: 36 processes with STATE = RSZDT

Figura 4.6 Monitoreo de llamadas activas

Además podemos observar el monitoreo en forma gráfica del tiempo total de las llamadas activas cómo se muestra en la figura 4.7.



**Figura 4.7 Monitoreo modo grafico de las llamadas activas**

### 4.3.2 MONITOREO DE LAS LLAMADAS PROCESADAS

Para esta prueba se hizo algunas llamadas desde los softphones en diferentes intervalos de tiempos para contar el número de llamadas hechas como se muestra en la figura 4.8.

Host	Service	Status	Last Check	Duration	Attempt	Status Information
asterisk	Asterisk Active Calls	OK	03-03-2011 16:01:13	0d 0h 34m 10s	1/4	Current Active Calls OK - 1
	Asterisk Calls Processed	OK	03-03-2011 16:01:13	0d 0h 34m 10s	1/4	Current Calls Processed OK - 1
	Asterisk Channels In Use	OK	03-03-2011 16:01:13	0d 0h 34m 10s	1/4	Current Channels Used OK - 2
	IAX2	OK	03-03-2011 16:01:13	0d 0h 34m 10s	1/4	Canales IAX2 Activos OK - 1
	SIP	OK	03-03-2011 16:01:13	0d 0h 34m 10s	1/4	Canales SIP Activos OK - 1
	SSH	OK	03-03-2011 16:01:13	0d 0h 34m 6s	1/4	SSH OK - OpenSSH_4.3 (protocol 2.0)
	Total Processes	OK	03-03-2011 16:01:13	33d 15h 5m 48s	1/4	PROCS OK: 33 processes with STATE = RSZDT
nmap	Current Load	OK	03-03-2011 16:01:13	0d 0h :6m 24s	1/4	OK - load ave age: 0.5/, 0.62, 1.13
	SSH	OK	03-03-2011 16:01:13	18d 18h 38m 13s	1/4	SSH OK - OpenSSH_4.3 (protocol 2.0)
	Total Processes	OK	03-03-2011 16:01:13	18d 18h 38m 13s	1/4	PROCS OK: 30 processes with STATE = RSZDT

**Figura 4.8 Monitoreo de las llamadas realizadas**

Además podemos observar el monitoreo en forma gráfica del total de las llamadas realizadas cómo se muestra en la figura 4.9.

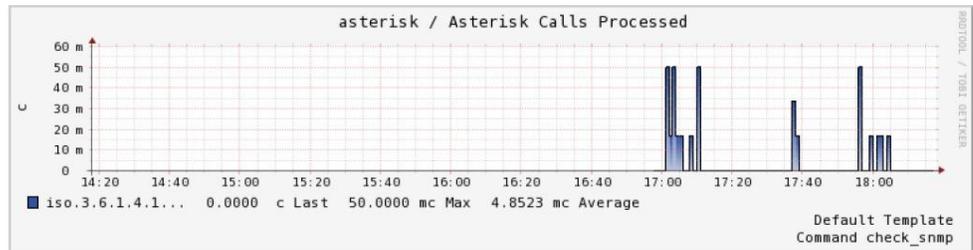


Figura 4.9 Monitoreo modo grafico de las llamadas realizadas

### 4.3.3 MONITOREO DEL NÚMERO DE CANALES ACTIVOS

Para esta prueba tuvimos que hacer llamadas desde los softphones para usar los canales IAX y SIP para que nuestro servidor nagios empiece a monitorear el total de los canales activos como se muestra en la figura 4.10.

Host ↑	Service ↑	Status ↑	Last Check ↑	Duration ↑	Attempt ↑	Status Information
asterisk	Asterisk Active Calls	OK	03-03-2011 16:01:13	0d 0h 34m 10s	1/4	Current Active Calls OK - 1
	Asterisk Calls Processed	OK	03-03-2011 16:01:13	0d 0h 34m 10s	1/4	Current Calls Processed OK - 4
	Asterisk Channels In Use	OK	03-03-2011 16:01:13	0d 0h 34m 10s	1/4	Current Channels Used OK - 2
	IAX2	OK	03-03-2011 16:01:13	0d 0h 34m 10s	1/4	Canales IAX2 Activos OK - 1
	SIP	OK	03-03-2011 16:01:13	0d 0h 34m 10s	1/4	Canales SIP Activos OK - 1
	SSH	OK	03-03-2011 16:01:13	0d 0h 34m 6s	1/4	SSH OK - OpenSSH_4.3 (protocol 2.0)
	Total Processes	OK	03-03-2011 16:01:13	33d 19h 5m 48s	1/4	PROCS OK: 53 processes with STATE = RSZDT
nagios	Current Load	OK	03-03-2011 16:01:13	0d 0h : 0m 24s	1/4	OK - load average: 0.57, 0.62, 1.13
	SSH	OK	03-03-2011 16:01:13	18d 18h 38m 13s	1/4	SSH OK - OpenSSH_4.3 (protocol 2.0)
	Total Processes	OK	03-03-2011 16:01:13	18d 18h 38m 13s	1/4	PROCS OK: 90 processes with STATE = RSZDT

Figura 4.10 Monitoreo de las llamadas activas

### 4.3.4 MONITOREO DE LOS CANAL IAX Y SIP

Para esta prueba tuvimos que hacer llamadas desde los softphones para usar los canales IAX y SIP, monitoreando en tiempo real ambos canales como se muestra en la figura 4.11.

Host	Service	Status	Last Check	Duration	Attempt	Status Information
asterisk	Asterisk Active Calls	OK	03-03-2011 16:01:13	0d 0h 34m 10s	1/4	Current Active Calls OK - 1
asterisk	Asterisk Calls Processed	OK	03-03-2011 16:01:13	0d 0h 34m 10s	1/4	Current Calls Processed OK - 4
asterisk	Asterisk Channels In Use	OK	03-03-2011 16:01:13	0d 0h 34m 10s	1/4	Current Channels Used OK - 2
asterisk	IAX2	OK	03-03-2011 16:01:13	0d 0h 34m 10s	1/4	Canales IAX2 Activos OK - 1
asterisk	SIP	OK	03-03-2011 16:01:13	0d 0h 34m 10s	1/4	Canales SIP Activos OK - 1
asterisk	SSH	OK	03-03-2011 16:01:13	0d 0h 34m 6s	1/4	SSH OK - OpenSSH_4.3 (protocol 2.0)
asterisk	Total Processes	OK	03-03-2011 16:01:13	33d 19h 5m 48s	1/4	PROCS OK: 33 processes with STATE = RSZDT
monite	Current Load	OK	03-03-2011 16:01:13	0d 0h : 6m 24s	1/4	OK - load ave age: 0.5r / 0.6z / 1.13
monite	SSH	OK	03-03-2011 16:01:13	18d 18h 38m 13s	1/4	SSH OK - OpenSSH_4.3 (protocol 2.0)
monite	Total Processes	OK	03-03-2011 16:01:13	18d 18h 38m 13s	1/4	PROCS OK: 30 processes with STATE = RSZDT

Figura 4.11 Monitoreo de los canales IAX y SIP

Además podemos observar el monitoreo en forma gráfica de los canales IAX y SIP como se muestra en la figura 4.12 y 4.13.

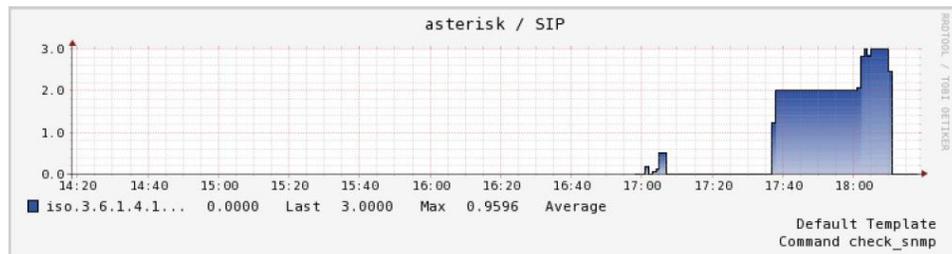


Figura 4.12 Monitoreo modo grafico del canal SIP

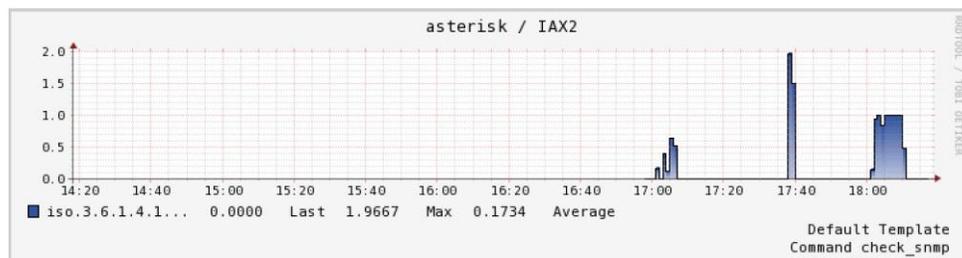


Figura 4.13 Monitoreo modo grafico del canal IAX

#### 4.3.5 FUNCIONAMIENTO DE LAS NOTIFICACIONES POR E-MAIL

Para esta prueba tuvimos que simular la caída del servidor Asterisk deteniendo el servicio de Asterisk a través del comando *service asterisk stop*, se verifico como los servicios a monitorear se cambiaron a un estado de falla. Por lo consiguiente el servidor Nagios envió las notificaciones al correo del administrador. Luego se procedió a reconectar el servicio de Asterisk a través del comando *service asterisk start* y el servidor Nagios envió las notificaciones de una exitosa recuperación de servidor Asterisk. En la figura 4.14 se muestra las notificaciones que se envían al correo del administrador.

Host	Service	Type	Time	Contact	Notification Command	Information
asterisk	<a href="#">Asterisk Channels In Use</a>	OK	03-03-2011 16:28:21	<a href="#">nagiosadmin</a>	<a href="#">notify-service-by-email</a>	Current Channels Used OK - 0
asterisk	<a href="#">Asterisk Active Calls</a>	OK	03-03-2011 16:28:20	<a href="#">nagiosadmin</a>	<a href="#">notify-service-by-email</a>	Current Active Calls OK - 0
asterisk	<a href="#">SIP</a>	OK	03-03-2011 16:28:20	<a href="#">nagiosadmin</a>	<a href="#">notify-service-by-email</a>	Canales SIP Activos OK - 0
asterisk	<a href="#">IAX2</a>	OK	03-03-2011 16:28:20	<a href="#">nagiosadmin</a>	<a href="#">notify-service-by-email</a>	Canales IAX2 Activos OK - 0
asterisk	<a href="#">Asterisk Calls Processed</a>	OK	03-03-2011 16:28:20	<a href="#">nagiosadmin</a>	<a href="#">notify-service-by-email</a>	Current Calls Processed OK - 0
asterisk	<a href="#">Asterisk Channels In Use</a>	UNKNOWN	03-03-2011 16:27:11	<a href="#">nagiosadmin</a>	<a href="#">notify-service-by-email</a>	No valid data returned
asterisk	<a href="#">SIP</a>	UNKNOWN	03-03-2011 16:27:11	<a href="#">nagiosadmin</a>	<a href="#">notify-service-by-email</a>	No valid data returned
asterisk	<a href="#">Asterisk Active Calls</a>	UNKNOWN	03-03-2011 16:27:10	<a href="#">nagiosadmin</a>	<a href="#">notify-service-by-email</a>	No valid data returned
asterisk	<a href="#">Asterisk Calls Processed</a>	UNKNOWN	03-03-2011 16:27:10	<a href="#">nagiosadmin</a>	<a href="#">notify-service-by-email</a>	No valid data returned
asterisk	<a href="#">IAX2</a>	UNKNOWN	03-03-2011 16:27:10	<a href="#">nagiosadmin</a>	<a href="#">notify-service-by-email</a>	No valid data returned

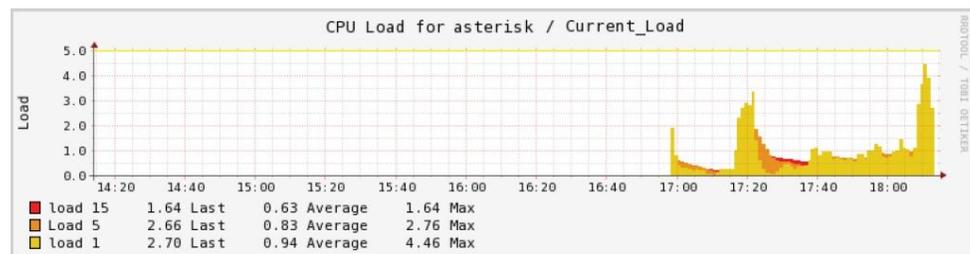
Figura 4.14 Notificaciones para el administrador.

## 4.4 ANÁLISIS DE RENDIMIENTO DE ASTERISK

Este análisis está basado en el redimiendo del hardware que utilizamos para la instalación del servidor Asterisk.

### 4.4.1 ANÁLISIS DE LA CARGA DEL PROCESADOR

Para esta prueba se realizaron varias llamadas las cuales fueron monitoreadas por Nagios, y se pudo observar que cuando se activan las llamadas, aumenta la carga del procesador, como se muestra en la figura 4.15.



**Figura 4.15 Carga del procesador de Asterisk**

### 4.4.2 ANÁLISIS DE LA CARGA DE LA MEMORIA RAM

Para esta prueba se realizaron varias llamadas para comparar el rendimiento de la memoria RAM con respecto al procesador. Y se pudo observar que la carga de la RAM no aumenta durante las llamadas realizadas, no obstante la carga del procesador si

aumenta, en la figura 4.16 se muestra en la parte inicial la carga del procesador con respecto a la memoria RAM cuando aún no se producían las llamadas.



**Figura 4.16 Carga inicial del procesador y de la memoria RAM**

Una vez que se realizaron las llamadas se pudo observar un considerable aumento en la carga del procesador, caso contrario en la carga de la memoria RAM que no se produjo ningún aumento, como se muestra en la figura 4.17.



**Figura 4.17 Carga final del procesador y de la memoria RAM**

# CONCLUSIONES Y RECOMENDACIONES

## CONCLUSIONES

1. Como hemos podido ver durante la realización de este proyecto de graduación, Nagios es un sistema de monitorización muy completo con grandes posibilidades de ampliación de sus características de monitoreo. Además es un software gratuito y libre lo que es conveniente para las empresas que deseen implementar un sistema de monitoreo potente a bajo costo.
2. Sin embargo es algo complejo al momento de configurarlo para que funciones correctamente pero vale el esfuerzo hacerlo ya que Nagios tiene las mejores características para el monitoreo de un servidor Asterisk.
3. También se han analizado los factores que influyen en el rendimiento de Asterisk, observando que el rendimiento se encuentra afectado por la capacidad del procesador. Este factor es importante al momento de instalar un servidor Asterisk, ya que al mayor número de llamadas simultáneas, mayor será la utilización del procesador.

## RECOMENDACIONES

1. Si se va a monitorear los servicios tales como, llamadas activas, llamadas procesadas y número de canales en uso. Se recomienda trabajar con el versión 1.6 de Asterisk ya que las versiones anteriores no cuentan con los parámetros necesarios para el monitoreo de estos servicios.
2. No confiarse de la información que se encuentra en internet, ya que en muchos de los manuales y foros hay información errónea y muy reducida acerca de la instalación de Nagios.
3. Actualizar el sistema operativo de lo contrario pueden surgir problemas al instalar Asterisk 1.6.
4. De acuerdo a nuestras conclusiones y experiencias adquiridas con el desarrollo de este proyecto de graduación recomendamos como trabajo futuro:
  1. Configurar Nagios para que las alerta de notificaciones sean enviadas a teléfonos celulares por medio de mensajes escritos.
  2. Instalar la nueva versión de Asterisk 1.8 para poder monitorear los nuevos servicios incluidos tales como canales disponibles al momento, llamadas en cola dando así prioridad a comunicaciones más importantes, entre otros.

# **ANEXOS**

# ANEXO 1

## PROCEDIMIENTO DETALLADO INSTALACIÓN DEL SISTEMA NAGIOS

### PROCEDIMIENTO DETALLADO DE LA INSTALACIÓN DEL SISTEMA NAGIOS

Tener actualizado el sistema operativo previamente con:

```
yum update
```

Tener instalado en su servidor la versión más reciente de apache, y los siguientes paquetes necesarios para que nuestro servidor nagios funcione correctamente, instalándolo de la siguiente manera:

```
yum install -y httpd gccglib cglibc-common gdgd-devel openssl-devel
```

Creamos el usuario nagios:

```
useradd -m nagios
```

Generamos el grupo nagcmd para permitir el envío de comandos desde la consola web y agregamos los usuarios nagios y apache a grupo creado anteriormente:

```
groupadd nagcmd
```

```
usermod -a -G nagcmd nagios
```

```
usermod -a -G nagcmd apache
```

Generamos la carpeta donde descargaremos nagios:

```
mkdir /opt/nagios
```

Descargamos dentro de nuestra nueva carpeta:

Nagios core 3.2.0

```
wget http://prdownloads.sourceforge.net/sourceforge/nagios/nagios-3.2.0.tar.gz
```

Nagios Plugins 1.4.14

```
wget http://prdownloads.sourceforge.net/sourceforge/nagiosplug/nagios-plugins-1.4.14.tar.gz
```

Nagiosnrpe 2.12 (Para monitorizar nuestro servidor Asterisk).

```
wget http://prdownloads.sourceforge.net/sourceforge/nagios/nrpe-2.12.tar.gz
```

Para buscar la versión estable más reciente visita:

<http://www.nagios.org/download/download.php>

### **Instalación de Nagios:**

Descomprimos la carpeta de nagios e ingresamos:

```
cd /opt/nagios
```

```
tarxzf nagios-3.2.0.tar.gz
```

```
cd nagios-3.2.0
```

Configuramos y Compilamos los archivos de configuración:

```
./configure --with-command-group=nagcmd
```

Compilamos binarios, init script y archivos de configuración:

```
make all
```

```
make install
```

```
make install-init
```

```
make install-config
```

```
make install-command mode
```

Con esto nagios queda instalado en **/usr/local/nagios**

Instalación de interfaz Web y Creación de Usuario Admin y su credencial:

```
cd /opt/nagios/nagios-3.2.0
```

```
make install-webconf
```

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

```
service httpd restart
```

No olvidar la contraseña que especifico ya que la requerirá para ingresar a la interfaz web.

## **INSTALACIÓN Y COMPILACIÓN DE NAGIOS PLUGINS.**

Descomprimos la carpeta de nagios-plugins e ingresamos:

```
cd /opt/nagios
```

```
tar xzf nagios-plugins-1.4.14.tar.gz
```

```
cd nagios-plugins-1.4.14
```

Configuramos y Compilamos los archivos de configuración:

```
./configure --with-nagios-user=nagios --with-nagios-group=nagios
```

```
make
```

```
make install
```

Permitimos que Nagios y Apache inicien al iniciar nuestro sistema:

```
chkconfig --add nagios
```

```
chkconfig nagios on
```

```
chkconfig httpd on
```

Verificamos los archivos de configuración:

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

Si se muestra a continuación lo siguiente, vamos bien con la instalación de nagios.

Total Warnings: 0

Total Errors: 0

## **INICIAMOS NAGIOS**

Iniciamos el servicio de nagios con el comando:

```
service nagios start
```

- La interfaz web quedó instalada en **/usr/local/nagios/share/**
- El archivo http.conf se añadió a **/etc/httpd/conf.d/nagios.conf**
- Para Ingresar al portal del Nagios utilizar **http://dirección-del-servidor/nagios**, utilizar el usuario **nagiosadmin** y la contraseña que le asignamos en pasos anteriores.

## ARCHIVOS DE CONFIGURACIÓN LOCALHOSTS.CFG

Editamos el archivo localhosts.cfg para agregar el servidor Asterisk con sus respectivos servicios básicos y modificar el servidor Nagios para su correcto monitoreo:

```
gedit /usr/local/nagios/etc/objects/localhosts.cfg
```

### Servidores a Monitorear:

```
# Define a host for the local machine

define host{
    use                linux-server,host-pnp
    host_name          nagios
    alias              servidor nagios
    address            200.126.12.117
    icon_image         NAGIOS.jpg
    statusmap_image    NAGIOS.jpg
}

define host{
    use                linux-server,host-pnp
    host_name          asterisk
    alias              servidor asterisk
    address            200.126.12.119
    icon_image         ASTERISK.jpg
    statusmap_image    ASTERISK.jpg
}
```

Se agregan los servidores a un Grupo:

```
# Define an optional hostgroup for Linux machines

define hostgroup{
    hostgroup_name linux-servers
    alias          servidor nagios
    members       nagios, asterisk
}

```

Servicios Básicos a Monitorear

Define para el monitoreo de procesos activos, warning si es >250 procesos y critical si es >400 procesos.

```
# Server Nagios

define service{
    use                local-service, srv-pnp
    host_name          nagios
    service_description Total Processes
    check_command      check_local_procs!250!400!RSZDT
}

```

Define para monitorear la carga.

```
define service{
    use                local-service, srv-pnp
    host_name          nagios
    service_description Current Load
    check_command      check_local_load!5.0,4.0,3.0!
10.0,6.0,4.0
}

```

Define para monitorear el servicio de ssh.

```
define service{
    use                local-service, srv-pnp
    host_name          nagios
    service_description SSH
    check_command       check_ssh
    notifications_enabled 1
}
```

Define para el monitoreo de procesos activos, warning si es >250 procesos y critical si es >400 procesos.

```
# Server Asterisk

define service{
    use                local-service, srv-pnp
    host_name          asterisk
    service_description Total Processes
    check_command       check_local_procs!250!400!RSZDT
}
```

Define para monitorear la carga.

```
define service{
    use                local-service, srv-pnp
    host_name          asterisk
    service_description Current Load
    check_command       check_local_load!5.0,4.0,3.0!
10.0,6.0,4.0
}
```

Define para monitorear el servicio de ssh.

```
define service{
    use                local-service, srv-pnp
    host_name          asterisk
    service_description SSH
    check_command       check_ssh
    notifications_enabled 1
}
```

# **ANEXO 2**

**PROCEDIMIENTO DETALLADO INSTALACIÓN DE NRPE**

**EN LOS SERVIDORES**

## **INSTALACIÓN Y PUESTA EN MARCHA DE NRPE EN LOS SERVIDORES**

### **Instalación de nrpe en el Servidor Asterisk**

Instalamos Openssl:

```
yum install openssl-devel
```

Generamos el usuario nagios con su credencial (debemos ser usuario root):

```
/usr/sbin/useradd nagios
```

```
Passwd nagios
```

Descargamos y Descomprimos NagiosPlugins:

```
mkdir /opt/nagios
```

```
cd /opt/nagios
```

```
wget http://prdownloads.sourceforge.net/sourceforge/nagiosplug/nagios-plugins-1.4.14.tar.gz
```

```
tarxzf nagios-plugins-1.4.14.tar.gz
```

Compilamos e Instalamos Nagios Plugins:

```
cd nagios-plugins-1.4.14
```

```
./configure
```

```
make
```

```
make install
```

Cambiamos Permisos:

```
chown nagios.nagios /usr/local/nagios
```

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

Instalamos Xinetd:

```
yum install xinetd
```

Descargamos y Descomprimos el Demonio NRPE:

```
cd /opt/nagios
```

```
wget http://prdownloads.sourceforge.net/sourceforge/nagios/nrpe-2.12.tar.gz
```

```
tar xzf nrpe-2.12.tar.gz
```

```
cd nrpe-2.12
```

Compilamos e Instalamos NRPE:

```
./configure
```

```
make all
```

```
make install-plugin
```

```
make install-daemon
```

```
make install-daemon-config
```

Instalamos el Demonio NRPE Como un Servicio en Xinetd:

```
make install -xinetd
```

Editamos la siguiente línea del archivo **/etc/xinetd.d/nrpe** para agregar la ip de nuestro servidor Nagios.

```
gedit /etc/xinetd.d/nrpe
```

```
only_from = 127.0.0.1, 200.126.12.117
```

- Donde **200.126.12.117** es la dirección de nuestro servidor Nagios.

- Aquí podemos ingresar la IP Pública de nuestro servidor Nagios en caso de que no se encuentre dentro de la misma red.
- De ser así será necesario redirigir el puerto **5666 TCP** al servidor que queremos monitorizar.

Agregamos la siguiente línea para nuestro demonio NRPE en el archivo **/etc/services**

```
gedit /etc/services
```

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

Reiniciamos Xinetd:

```
service xinetd restart
```

Probamos el demonio NRPE localmente:

```
netstat -at | grep nrpe
```

Debemos obtener la siguiente salida:

```
Tcp      0      0 *:nrpe      *.*      LISTEN
```

En caso de no obtener dicha salida debemos revisar lo siguiente:

- Se agregó correctamente la entrada en el archivo **/etc/services**
- La línea `only_from` del archivo **/etc/xinetd.d/nrpe** contiene **127.0.0.1**
- Xinetd no está correctamente instalado o corriendo.

Verificamos que el demonio NRPE está correctamente instalado, para ello utilizaremos el plugin `check_nrpe` que instalamos para pruebas.

```
/usr/local/nagios/libexec/check_nrpe -H localhost
```

Debemos de obtener la siguiente salida:

NRPE v2.12

- Los comandos que se utilizaran para medir los recursos locales se encuentran configurados dentro del archivo **/usr/local/nagios/etc/nrpe.cfg**
- Para probar los comandos configurados por defecto podemos correr cualquiera de los siguientes comandos:

```
/usr/local/nagios/libexec/check_nrpe -H localhost -c check_users
```

```
/usr/local/nagios/libexec/check_nrpe -H localhost -c check_load
```

```
/usr/local/nagios/libexec/check_nrpe -H localhost -c check_hda1
```

```
/usr/local/nagios/libexec/check_nrpe -H localhost -c check_total_procs
```

```
/usr/local/nagios/libexec/check_nrpe -H localhost -c check_zombie_procs
```

## **Instalación de nrpe en el Servidor Nagios**

Descargamos y Descomprimos el Demonio NRPE:

```
cd /opt/nagios
```

```
wget http://prdownloads.sourceforge.net/sourceforge/nagios/nrpe-2.12.tar.gz
```

```
tar xzf nrpe-2.12.tar.gz
```

```
cd nrpe-2.12
```

Configuramos y compilamos en archivo NRPE:

```
./configure
```

```
make all
```

```
make install-plugin
```

Probamos la comunicación con el demonio NRPE de nuestro Servidor Asterisk:

```
/usr/local/nagios/libexec/check_nrpe -H <ip-servidor-Asterisk>
```

```
/usr/local/nagios/libexec/check_nrpe -H 200.126.12.119
```

Debemos de obtener:

NRPE v2.12

En caso de no obtener dicha salida debemos revisar lo siguiente:

- Verificar que ningún firewall este bloqueando la comunicación entre ambos servidores.
- Verificar que el demonio NRPE esté funcionando correctamente en el servidor a monitorizar.
- Asegúrese que el servidor a monitorizar no tenga tablas IP que bloquen el tráfico de entrada o salida.

### **Archivos de configuración commands.cfg**

Editamos el archivo de commands.cfg agregando la siguiente definición para poder usar el plugin NRPE:

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

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

Hasta el momento ya tenemos nuestro servidor de monitoreo instalado con Nagios y check\_nrpe, también hemos configurado exitosamente el demonio nrpe en nuestro servidor Asterisk que nos permite tener control sobre el consumo de recursos.

# **ANEXO 3**

**PROCEDIMIENTO DETALLADO INSTALACIÓN DE SNMP**

**EN LOS SERVIDORES**

## INSTALACIÓN DE SNMP EN LOS SERVIDORES

### SNMP Servidor Asterisk

Para empezar debemos de tener instalado y configurado Asterisk, luego buscamos si tenemos el módulo `res_snmp` instalado y cargado en Asterisk:

Ingresamos a la consola de Asterisk:

```
asterisk -r
```

```
CLI> module show like snmp
```

Si nos muestra un mensaje similar a la figura, significa que no lo tenemos instalado.

```
asterisk*CLI> module show like snmp
Module                Description                Use Count
0 modules loaded
snmp*CLI> quit
Executing last minute cleanups
```

Salimos de Asterisk:

```
CLI>quit
```

Paramos Asterisk:

```
/etc/init.d/asterisk stop
```

Instalamos los paquetes que se necesitan para la instalación del módulo res\_snmp:

```
yum install net-snmp net-snmp-devel net-snmp-libs net-snmp-perl net-snmp-utils
```

Entramos en la carpeta de las fuentes de Asterisk y volvemos a compilar:

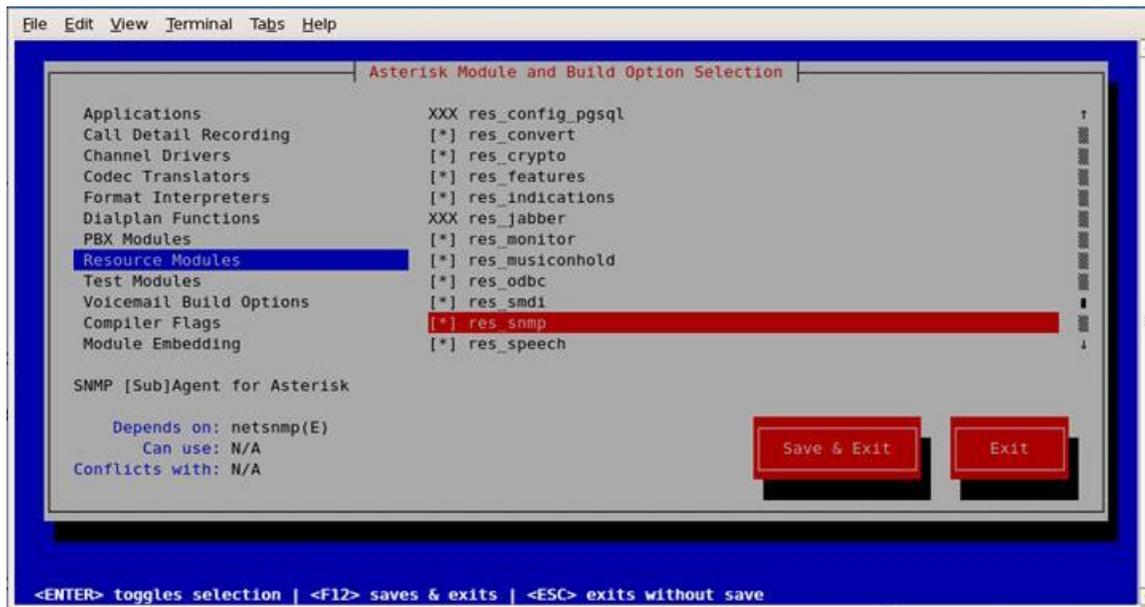
```
cd /usr/src/asterisk-1.6.2.14
```

```
make distclean
```

```
./configure --with-snmp
```

```
make menu select
```

En la ventana que aparece nos aseguramos que el módulo res\_snmp esté activado:



Configuramos nuevamente Asterisk:

make

make install

Volvemos a arrancar Asterisk y averiguamos si ahora el modulo está cargado:

/etc/init.d/asterisk start

asterisk -r

CLI> module show like snmp

Si nos muestra esto significa que ya lo tenemos instalado:

```
asterisk*CLI> module show like snmp
Module                Description                Use Count
res_snmp.so          SNMP [Sub]Agent for Asterisk    0
1 modules loaded
asterisk*CLI> █
```

Salimos de la consola de Asterisk:

CLI>quit

Ahora configuramos SNMP de modo que pueda interactuar con Asterisk editando el archivo snmpd.conf:

gedit /etc/snmp/snmpd.conf

```
master agentx

agentXPerms 0660 0660 asterisk asterisk

com2sec local 127.0.0.1/32 821514821514
com2sec remote 200.126.12.0/24 821514821514

group asterisk v1 local
group asterisk v2c local
group NetWork v1 remote
group NetWork v2c remote

view all included .1

access asterisk "" any noauth exact all none none
access NetWork "" any noauth exact all none none
```

Primero definimos el agente y los permisos de acceso. Luego definimos dos comunidades. Una tiene acceso local y otra remoto, útil si queremos hacer consultas desde otro servidor. Hay que sustituir XXX.XXX.XXX.XXX con la ip del servidor remoto. Luego definimos dos grupos y qué tipo de consultas pueden hacer (en este caso SNMP versión 1 y versión 2c. Definimos el tipo de objetos que pueden consultar (en estos casos todos) y los permisos que cada grupo tiene.

Guardamos los cambios y arrancamos SNMP:

```
service snmpd restart
```

Cambiamos los permisos del agentX:

```
# chmod 755 /var/agentx
```

Configuramos el archivo res\_snmp.conf:

```
gedit /etc/asterisk/res_snmp.conf
```

Le quitamos los “;” a las siguientes líneas quedando de la siguiente manera:

```

;
; Configuration file for res_snmp
;-----
;
; Res_snmp can run as a subagent or standalone SNMP agent. The
standalone snmp
; agent is based on net-snmp and will read a configuration file
called
; asterisk.conf in the net-snmp configuration file path, starting
with
; /etc/snmp on many systems.
;
; If you use the subagent model, you need to enable agentx in
snmpd.conf
; Note that you can only run one Asterisk on the system in this
case.
;
; Please read documentat in doc/snmp.txt to get more information
about
; snmp support in Asterisk

[general]
; We run as a subagent per default -- to run as a full agent
; we must run as root (to be able to bind to port 161)
subagent = yes
; SNMP must be explicitly enabled to be active
enabled = yes

```

Guardamos los cambios y copiamos los OID de Asterisk en la carpeta de snmp.

Un OID (identificador de objeto) es una cadena alfanumérica que se utiliza para identificar de forma única un objeto.

```
cp /usr/src/asterisk-1.6.2.14/doc/asterisk-mib.txt /usr/share/snmp/mibs
```

```
cp /usr/src/asterisk-1.6.2.14/doc/digium-mib.txt /usr/share/snmp/mibs
```

Reiniciamos el SNMP:

```
service snmpd restart
```

Exportamos los archivos MIB del asterisk al SNMP:

```
export MIBS=all
```

Reiniciamos Asterisk:

```
service asterisk restart
```

Ahora probaremos si la instalación y configuración ha sido exitosa en el servidor Asterisk utilizando dos comandos y seleccionaremos los parámetros OID correspondientes para agregar los servicios de Asterisk que van hacer monitoreados respectivamente:

```
snmpwalk -OT -c 821514821514 -v 2c localhost .1.3.6.1.4.1.22736
```

**ASTERISK-MIB::astConfigCallsActive.0 = Gauge32: 0**

**ASTERISK-MIB::astConfigCallsProcessed.0 = Counter32: 0**

ASTERISK-MIB::astNumModules.0 = INTEGER: 187

ASTERISK-MIB::astNumIndications.0 = INTEGER: 40

ASTERISK-MIB::astCurrentIndication.0 = STRING: us

ASTERISK-MIB::astChanTypeName.2 = STRING: Phone

ASTERISK-MIB::astChanTypeName.3 = STRING: DAHDI

ASTERISK-MIB::astChanTypeName.4 = STRING: OOH323

**ASTERISK-MIB::astChanTypeName.5 = STRING: IAX2**

ASTERISK-MIB::astChanTypeName.6 = STRING: Skinny

**ASTERISK-MIB::astChanTypeName.7 = STRING: SIP**

ASTERISK-MIB::astChanTypeName.8 = STRING: Bridge

ASTERISK-MIB::astChanTypeName.9 = STRING: Console

ASTERISK-MIB::astChanTypeName.10 = STRING: MGCP

ASTERISK-MIB::astChanTypeName.11 = STRING: Local

ASTERISK-MIB::astChanTypeName.12 = STRING: USTM

snmpwalk -On -c 821514821514 -v 2c localhost .1.3.6.1.4.1.22736

.1.3.6.1.4.1.22736.1.1.1.0 = STRING: 1.6.2.14

.1.3.6.1.4.1.22736.1.1.2.0 = Gauge32: 10602

.1.3.6.1.4.1.22736.1.2.1.0 = Timeticks: (154228) 0:25:42.28

.1.3.6.1.4.1.22736.1.2.2.0 = Timeticks: (154228) 0:25:42.28

.1.3.6.1.4.1.22736.1.2.3.0 = INTEGER: 2791

.1.3.6.1.4.1.22736.1.2.4.0 = STRING: /var/run/asterisk/asterisk.ctl

**.1.3.6.1.4.1.22736.1.2.5.0 = Gauge32: 0**

**.1.3.6.1.4.1.22736.1.2.6.0 = Counter32: 0**

.1.3.6.1.4.1.22736.1.5.4.1.7.1 = Gauge32: 0

.1.3.6.1.4.1.22736.1.5.4.1.7.2 = Gauge32: 0

.1.3.6.1.4.1.22736.1.5.4.1.7.3 = Gauge32: 0

.1.3.6.1.4.1.22736.1.5.4.1.7.4 = Gauge32: 0

**.1.3.6.1.4.1.22736.1.5.4.1.7.5 = Gauge32: 0**

.1.3.6.1.4.1.22736.1.5.4.1.7.6 = Gauge32: 0

**.1.3.6.1.4.1.22736.1.5.4.1.7.7 = Gauge32: 0**

.1.3.6.1.4.1.22736.1.5.4.1.7.8 = Gauge32: 0

.1.3.6.1.4.1.22736.1.5.4.1.7.9 = Gauge32: 0

.1.3.6.1.4.1.22736.1.5.4.1.7.10 = Gauge32: 0

.1.3.6.1.4.1.22736.1.5.4.1.7.11 = Gauge32: 0

.1.3.6.1.4.1.22736.1.5.4.1.7.12 = Gauge32: 0

**.1.3.6.1.4.1.22736.1.5.5.1.0 = Gauge32: 0**

## SNMP Servidor Nagios

Configuración de SNMP en el servidor Nagios.

Para añadir los dos comandos en Nagios modificaremos el archivo  
commands.cfg

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

Al final del archivo añadimos:

```
define command{
    command_name    check_snmp
    command_line    $USER1$/check_snmp -H $HOSTADDRESS$ -C
821514821514 -o $ARG1$ -P 2c -l $ARG2$ -w $ARG3$ -c $ARG4$
}
```

```
define command{
    command_name    check_asterisk
    command_line    $USER1$/contrib/check_asterisk.pl -h
$HOSTADDRESS$ -m mgr -u admin -p sesamo
}
```

Admin y Sesamo indican respectivamente el usuario y la contraseña para conectarse al AMI de Asterisk. Para definirlos tenemos que modificar el archivo manager.conf de Asterisk de la siguiente forma:

```
gedit /etc/asterisk/manager.conf
```

```
[general]
displayssystemname = yes
enabled = yes
webenabled = yes
port = 5038

[admin]
secret = sesamo
deny=0.0.0.0/0.0.0.0
permit=127.0.0.1/255.255.255.255
read = system,call,log,verbose,command,agent,user,config
write = system,call,log,verbose,command,agent,user,config
```

Para actualizar la configuración:

```
/etc/init.d/asterisk reload
```

Para terminar la configuración modificamos el archivo localhost.cfg del Servidor Nagios para agregar los servicios adicionales con los parámetros OID que en pasos anteriores seleccionamos de la información que nos mostró nuestro Servidor Asterisk:

```
gedit /usr/local/nagios/etc/objects/localhosts.cfg
```

Define para monitorear los canales SIP activos con su OID correspondiente

```
“.1.3.6.1.4.1.22736.1.5.4.1.7.7”:
```

```

define service{
    use                local-service, srv-pnp
    host_name          asterisk
    service_description SIP
    check_command
check_snmp!.1.3.6.1.4.1.22736.1.5.4.1.7.7!"Canales SIP
Activos"!8!16
    notifications_enabled 1
    icon_image           SIPIAX.jpg
}

```

Define para monitorear los canales IAX activos con su OID correspondiente

“.1.3.6.1.4.1.22736.1.5.4.1.7.5”:

```

define service{
    use                local-service, srv-pnp
    host_name          asterisk
    service_description IAX2
    check_command
check_snmp!.1.3.6.1.4.1.22736.1.5.4.1.7.5!"Canales IAX2
Activos"!4!8
    notifications_enabled 1
    icon_image           SIPIAX.jpg
}

```

Define para monitorear el servicio de llamadas activas con su OID

correspondiente “.1.3.6.1.4.1.22736.1.2.5.0”:

```

define service{
    use                local-service, srv-pnp
    host_name          asterisk
    service_description Asterisk Active Calls
    check_command
check_snmp!.1.3.6.1.4.1.22736.1.2.5.0!"Current Active Calls"!
8!16
    notifications_enabled 1
    icon_image           SIPIAX.jpg
}

```

Define para monitorear el servicio de procesos de llamadas con su OID correspondiente “.1.3.6.1.4.1.22736.1.2.6.0”:

```
define service{
    use                local-service, srv-pnp
    host_name          asterisk
    service_description Asterisk Calls Processed
    check_command
check_snmp!.1.3.6.1.4.1.22736.1.2.6.0!"Current Calls
Processed"!8!16
    notifications_enabled 1
    icon_image           SIPIAX.jpg
}
```

Define para monitorear el servicio de canales en uso con su OID correspondiente “.1.3.6.1.4.1.22736.1.5.1.0”:

```
define service{
    use                local-service, srv-pnp
    host_name          asterisk
    service_description Asterisk Channels In Use
    check_command
check_snmp!.1.3.6.1.4.1.22736.1.5.1.0!"Current Channels Used"!
8!16
    notifications_enabled 1
    icon_image           SIPIAX.jpg
}
```

Ahora estamos listos para arrancar Nagios. Antes de hacerlo hay una forma de controlar que todos los archivos de Nagios estén sin errores:

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

Si la respuesta termina con:

Total Warnings: 0

Total Errors: 0

Things look okay - No serious problems were detected during the pre-flight check

Significa que todo está bien. En caso contrario Nagios indicará el archivo y la línea donde se encuentra el error.

Ahora podremos ingresar a nuestro servidor de monitoreo ingresando a la dirección <http://200.126.12.117/nagios> e ingresando nuestro usuario de administrador con su respectiva contraseña.

Applications Places System 3:45 PM

Nagios - Mozilla Firefox

File Edit View History Bookmarks Tools Help

http://200.126.12.117/nagios/

Como configurar Po... En un momento: P... Como configurar D... Enviar mail autentic... Loading... instalacion y config... Configurar postfix a...

N Nagios Gmail - Recibidos (2) - juca...

# Nagios®

**Nagios® Core™  
Version 3.2.0**  
August 12, 2009  
[Check for updates](#)

[Read what's new in Nagios Core 3](#)

**A new version of Nagios is available!**  
Visit [nagios.org](http://nagios.org) to download Nagios 3.2.3.

Copyright © 2009 Nagios Core Development Team and Community Contributors.  
Copyright © 1999-2009 Ethan Galstad.  
See the THANKS file for more information on contributors.

Nagios Core is licensed under the GNU General Public License and is provided AS IS with NO WARRANTY OF ANY KIND, INCLUDING THE WARRANTY OF DESIGN, MERCHANTABILITY, AND FITNESS FOR A PARTICULAR PURPOSE. Nagios, Nagios Core and the Nagios logo are trademarks, service marks, registered trademarks or registered service marks owned by Nagios Enterprises, LLC. Usage of the Nagios marks are governed by our [trademark policy](#).

**Nagios®** PROBATIONER

http://200.126.12.117/nagios/docs/

**General**  
Home  
Documentation  
**Current Status**  
Tactical Overview  
Map  
Hosts  
Services  
Host Groups  
• Summary  
• Grid  
Service Groups  
• Summary  
• Grid  
Problems  
• Services (Unhandled)  
• Hosts (Unhandled)  
• Network Outages  
Quick Search:  
Reports  
Availability  
Trends  
Alerts  
• History  
• Summary  
• Histogram  
Notifications

# **ANEXO 4**

**PROCEDIMIENTO DETALLADO INSTALACIÓN DE  
MOTOR GRÁFICO PNP4NAGIOS**

## **INSTALACIÓN DEL MOTOR GRAFICO PNP4NAGIOS**

Un requisito fundamental para instalar PNP4NAGIOS es actualizar el php a la versión 5.2.

Primero verificar si contamos con la versión php-5.2:

```
rpm -qa |grep php
```

```
php-common-5.1.6-15.el5.i386
```

```
php-cli-5.1.6-15.el5.i386
```

```
php-5.1.6-15.el5.i386
```

```
php-pdo-5.1.6-15.el5.i386
```

```
php-bcmath-5.1.6-15.el5.i386
```

```
php-ldap-5.1.6-15.el5.i386
```

```
php-devel-5.1.6-15.el5.i386
```

```
php-gd-5.1.6-15.el5.i386
```

```
php-xml-5.1.6-15.el5.i386
```

php-mbstring-5.1.6-15.el5.i386

php-mysql-5.1.6-15.el5.i386

php-dba-5.1.6-15.el5.i386

Segundo creamos un archivo para poder añadir los repositorios de desarrollo que vamos a configurar para que solo los paquetes PHP se actualicen:

/etc/yum.repos.d/CentOS-Testing.repo

Agregamos la siguiente configuración en este archivo:

# CentOS-Testing:

# !!!! CAUTION !!!!

# This repository is a proving grounds for packages on their way to CentOSPlus and CentOS Extras.

# They may or may not replace core CentOS packages, and are not guaranteed to function properly.

# These packages build and install, but are waiting for feedback from testers as to

# functionality and stability. Packages in this repository will come and go during the

# development period, so it should not be left enabled or used on production systems without due

# consideration.

[c5-testing]

name=CentOS-5 Testing

baseurl=http://dev.centos.org/centos/\$releasever/testing/\$basearch/

enabled=1

gpgcheck=1

gpgkey=http://dev.centos.org/centos/RPM-GPG-KEY-CentOS-testing

includepkgs=php

Tercero actualizamos php con el siguiente comando:

yum update

Cuarto reiniciamos el servicio de http con el siguiente comando:

```
service httpd restart
```

Por último verificamos que todos los paquetes de php se han actualizado a la versión 5.2:

```
rpm -qa |grepphp
```

```
php-cli-5.2.6-2.el5s2
```

```
php-mbstring-5.2.6-2.el5s2
```

```
php-devel-5.2.6-2.el5s2
```

```
php-pdo-5.2.6-2.el5s2
```

```
php-gd-5.2.6-2.el5s2
```

```
php-dba-5.2.6-2.el5s2
```

```
php-common-5.2.6-2.el5s2
```

```
php-bcmath-5.2.6-2.el5s2
```

```
php-xml-5.2.6-2.el5s2
```

php-pear-1.5.1-2.el5s2

php-ldap-5.2.6-2.el5s2

php-5.2.6-2.el5s2

php-mysql-5.2.6-2.el5s2

php -v

PHP 5.2.6 (cli) (built: Sep 15 2008 20:42:05)

Copyright (c) 1997-2008 The PHP Group

Zend Engine v2.2.0, Copyright (c) 1998-2008 Zend Technologies

Ahora estamos listos para descargar la última versión de PNP4NAGIOS e instalarlo.

Descargamos los siguientes paquetes rrdtool, php-gd y pnp4nagios:

```
wget http://dag.wieers.com/rpm/packages/rpmforge-release/rpmforge-release-0.3.6-1.el5.rf.i386.rpm
```

```
rpm -hiv rpmforge-release-0.3.6-1.el5.rf.i386.rpm
```

```
yum -y install rrdtoolphp-gd
```

```
wget http://softlayer.dl.sourceforge.net/project/pnp4nagios/PNP-0.6/pnp4nagios-0.6.11.tar.gz
```

```
tar zxvf pnp4nagios-0.6.11.tar.gz
```

```
cd pnp4nagios-0.6.11
```

```
./configure --with-nagios-user=nagios --with-nagios-group=nagiosadmin
```

```
make all
```

```
make install-webconf
```

```
make install-config
```

make install-init

Configuramos PNP4NAGIOS:

```
gedit /etc/httpd/conf.d/pnp4nagios.conf
```

También en el fichero /usr/local/pnp4nagios/etc/config.php editamos la siguiente opción:

*Link to NagiosCGIs*

```
$conf['nagios_base'] = "/nagios/cgi-bin";
```

Si queremos ponerlo en español modificamos la siguiente directiva:

```
$conf['lang'] = "es_ES";
```

Renombramos todos los ficheros que contengan `–sample` al final, eliminando la palabra `sample` en la siguiente ruta /usr/local/pnp4nagios/etc

Reiniciamos Apache para que la actualización a php-5.2 surta efecto:

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

Modificamos el fichero ubicado en /usr/local/nagios/etc/nagios.cfg para activar el performance data en modo síncrono:

*process\_performance\_data=1*

*process\_perf\_data 0*

*enable\_environment\_macros=1*

*host\_perfdata\_command=process-host-perfdata*

*service\_perfdata\_command=process-service-perfdata*

Nagios debe ser notificado sobre los comandos referenciados para ello  
modificamos el fichero commands.cfg:

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

Agregamos en el fichero commands.cfg:

```
define command {
    command_name    process-service-perfdata
    command_line    /usr/bin/perl
                   /usr/local/pnp4nagios/libexec/process_perfdata.pl
}

define command {
    command_name    process-host-perfdata
    command_line    /usr/bin/perl
                   /usr/local/pnp4nagios/libexec/process_perfdata.pl -d
                   HOSTPERFDATA
}
```

Utilizaremos dos plantillas para poder enlazar los servidores Nagios y Asterisk como también sus respectivos servicios, para poder visualizarlos gráficamente agregando en nuestro fichero localhost.cfg en la parte donde especificamos nuestros servidores agregamos “host-pnp” y en la parte donde indicamos los servicios de ambos servidores agregamos “srv-pnp” de tal manera quedando de esta forma:

### Servidores agregamos “**host-pnp**”

```
define host{
    use                linux-server,host-pnp
    host_name          nagios
    alias               servidor nagios
    address             200.126.12.117
    icon_image         NAGIOS.jpg
    statusmap_image    NAGIOS.jpg
}

define host{
    use                linux-server,host-pnp
    host_name          asterisk
    alias               servidor asterisk
    address             200.126.12.119
    icon_image         ASTERISK.jpg
    statusmap_image    ASTERISK.jpg
}
```

## Servicios agregamos “**srv-pnp**”

```
define service{
    use                local-service, srv-pnp
    host_name          nagios
    service_description Total Processes
    check_command      check_local_procs!250!400!RSZDT
}
```

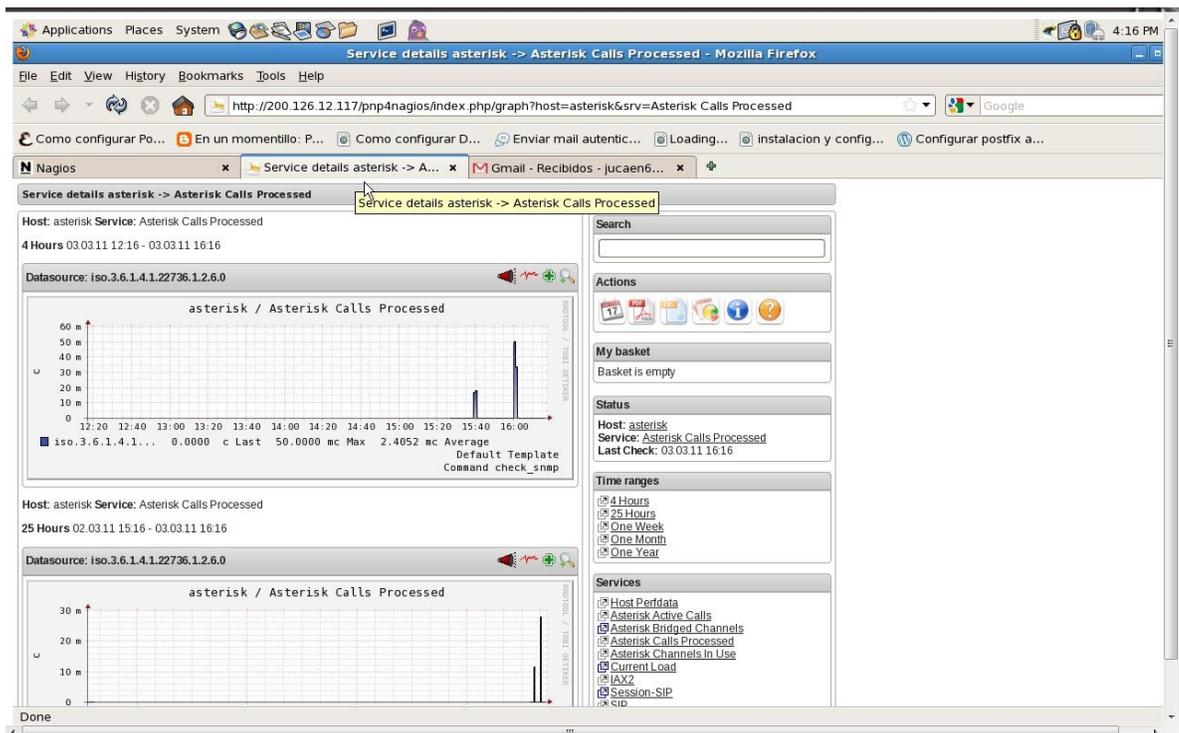
```
define service{
    use                local-service, srv-pnp
    host_name          asterisk
    service_description Total Processes
    check_command      check_local_procs!250!400!RSZDT
}
```

Una vez configurado todo, podemos arrancar y activar el proceso NPCD el cual nos permite crear las gráficas automáticamente:

```
/etc/init.d/npcd start
```

```
chkconfig npcd on
```

Para visualizar las gráficas abrimos un navegador e ingresamos en la dirección de nuestro motor gráfico <http://200.126.12.117/pnp4nagios>, comprobamos que todo está perfectamente, del no ser así eliminamos el fichero ubicado en `/usr/local/pnp4nagios/share/install.php`



También modificaremos el fichero `templates.cfg` agregando un par de sentencias para poder visualizar los gráficos sin tener que hacer click en ningún icono de la consola del servidor nagios, esto se puede hacer usando **CGI Includes** que incluye códigos de JavaScript en nuestro `status.cgi`:

```

define host {
    name          host-pnp
    action_url    /pnp4nagios/index.php/graph?host=$HOSTNAME
    $&srv=_HOST_' class='tips' rel='/pnp4nagios/index.php/popup?host=
$HOSTNAME$&srv=_HOST_
    register      0
}

define service {
    name          srv-pnp
    action_url    /pnp4nagios/index.php/graph?host=$HOSTNAME$&srv=
$SERVICEDESC$' class='tips' rel='/pnp4nagios/index.php/popup?
host=$HOSTNAME$&srv=$SERVICEDESC$
    register      0
}

```

The screenshot displays the Nagios web interface. At the top, there are navigation menus and a search bar. The main content area is divided into several sections:

- Current Network Status:** Last Updated: Thu Mar 3 16:16:28 ECT 2011. Updated every 3 seconds. Nagios® Core™ 3.2.0 - www.nagios.org. Logged in as nagiosadmin.
- Host Status Totals:** A summary table showing counts for Up, Down, Unreachable, and Pending hosts.
- Service Status Totals:** A summary table showing counts for Ok, Warning, Unknown, Critical, and Pending services.
- Service Status Details For All Hosts:** A table listing services for hosts 'asterisk' and 'nagios'.

Host	Service	Status	Last Check	Duration	Attempt	Status Information
asterisk	Asterisk Active Calls	OK	03-03-2011 16:16:23	0d 0h 46m 21s	1/4	Current Active Calls OK - 0
	Asterisk Calls Processed	OK	03-03-2011 16:16:23	0d 0h 46m 21s	1/4	Current Calls Processed OK - 5
	Asterisk Channels In Use	OK	03-03-2011 16:16:23	0d 0h 46m 21s	1/4	Current Channels Used OK - 0
	IAX2	OK	03-03-2011 16:16:23	0d 0h 46m 21s	1/4	Canales IAX2 Activos OK - 0
	SIP	OK	03-03-2011 16:16:23	0d 0h 46m 21s	1/4	Canales SIP Activos OK - 0
nagios	Total Processes	OK	03-03-2011 16:16:23	33d 15h 20m 59s	1/4	PROCS OK: 59 processes with STATE = RSZDT
	Current Load	OK	03-03-2011 16:16:23	0d 0h 31m 35s	1/4	OK - load average: 0.58, 0.63, 0.86
	SSH	OK	03-03-2011 16:16:23	18d 18h 53m 24s	1/4	SSH OK - OpenSSH_4.3 (protocol 2.0)
	Total Processes	OK	03-03-2011 16:16:23	18d 18h 53m 24s	1/4	PROCS OK: 56 processes with STATE = RSZDT

10 Matching Service Entries Displayed

# **ANEXO 5**

**ARCHIVOS DE CONFIGURACIÓN DEL SERVIDOR**

**ASTERISK**

## ARCHIVOS DE CONFIGURACIÓN DEL SERVIDOR ASTERISK

### SIP.CONF:

```
[general]
context=default
srvlookup=yes
```

```
[801]
type=friend
secret=801
qualify=yes
nat=no
host=dynamic
canreinvite=no
context=internal
```

```
[802]
type=friend
secret=802
qualify=yes
nat=no
host=dynamic
canreinvite=no
context=internal
```

## **IAX.CONF:**

```
[general]
jitterbuffer=yes
bindport=4569
calltokenoptional=0.0.0.0/0.0.0.0
requirecalltoken=auto
maxcallnumbers=512
```

```
[804]
qualify=yes
callerid=Edward Sinche
secret=804
type=friend
host=dynamic
context=internal
```

```
[805]
qualify=yes
callerid=JuanCarlos Ortega
secret=805
type=friend
host=dynamic
context=internal
```

## **EXTENSIONS.CONF:**

```
[general]
autofallthrough=yes
clearglobalvars=no
```

```
[globals]
EdwardGaston=SIP/801
JuanCarlos=SIP/802
EdwardSinche=IAX2/804
JuanCarlosOrtega=IAX2/805
```

```
[macro-extensiones]
```

```
exten=>s,1,Dial(${ARG1},5,r)
exten=>s,2,VoiceMail(${MACRO_EXTEN}@default,u)
exten=>s,3,Hangup()
exten=>s,102,VoiceMail(${MACRO_EXTEN}@default,b)
exten=>s,103,Hangup()
```

```
[internal]
```

```
exten=>801,1,Macro(extensiones,SIP/801)
exten=>802,1,Macro(extensiones,SIP/802)
exten=>804,1,Macro(extensiones,IAX2/804)
exten=>805,1,Macro(extensiones,IAX2/805)

exten=>800,1,VoiceMailMain()
```

## **GLOSARIO DE TÉRMINOS**

**Centos.5.5.-** “Community Enterprise Operating System” es una distribución de Linux.

**VOIP.-** Voz sobre IP, es la tecnología que permite la transmisión de la voz sobre el protocolo IP.

**Asterisk.-** Es una aplicación de código abierto para controlar y gestionar comunicaciones de cualquier tipo, ya sean analógicas, digitales o VoIP mediante todos los protocolos VoIP que implementa.

**SIP.-** “Session Initiation Protocol” SIP es un protocolo de control de capa de aplicación que permite establecer, modificar y finalizar sesiones multimedia (conferencias), tales como las llamadas de telefonía por Internet.

**IAX.-** “Inter-Asterisk eXchange protocol” es un protocolo que busca minimizar el ancho de banda utilizado en la transmisión de voz y vídeo a través de la red.

**Nagios.-** Es un sistema de vigilancia de gran alcance que permite a las organizaciones a identificar y resolver los problemas de infraestructura de IT antes de que afecten los procesos de negocio críticos. Proporciona control de todos los componentes de la infraestructura de misión crítica - incluidos las

aplicaciones, servicios, sistemas operativos, protocolos de red, parámetros del sistema, y la infraestructura de red.

**NRPE.-** Es un demonio que permite la comunicación de Nagios con una maquina Linux/Unix ya que hace posible la ejecución del plugin de Nagios en los host remotos Linux.

**SNMP.-** “Simple Network Management Protocol”, es un protocolo de capa de aplicación que facilita el intercambio de información de gestión entre dispositivos de red. Es parte del conjunto de protocolos de Protocolo de Control de Transmisión / Protocolo Internet (TCP / IP)

**Sendmail.-** Es un agente de transporte de correo, responsable de la mayoría de envíos del correo de internet.

**Postfix.-** Es un servidor de correo de software libre. Un programa informático para el enrutamiento y envío de correo electrónico, creado con la intención de que sea una alternativa más rápida, fácil de administrar y segura al ampliamente utilizado Sendmail.

**Plugins.-** Complemento o plug-in en ingles, es una aplicación que se relaciona con otra para aportarle una función nueva y generalmente muy específica.

**MIB.-** “Management Information Base” es un conjunto de datos que contiene información jerárquica, estructurada de todos los dispositivos gestionados en una red de comunicaciones

**OID.-** Identificadores de objeto para identificar las variables de la MIB

**GCI.-** “Common Gateway Interface” es un estándar para la interfaz de aplicaciones externas con los servidores de información, como el HTTP o servidores Web.

**Addons.-** Subprogramas opcionales que sólo funcionan anexados a otro y que sirven para incrementar o complementar sus funcionalidades.

**RTPC.-**Red Telefónica Pública Conmutada: Es el conjunto de elementos que hacen posible la transmisión conmutada de voz, con acceso generalizado al público.

# BIBLIOGRAFÍA

[1] Wikipedia, “Introducción a la Central VoIP Asterisk”,

<http://es.wikipedia.org/wiki/Asterisk>, 18 Marzo 2011.

[2] Wikipedia, “Introducción a PBX (Private Branch Exchange)”,

<http://es.wikipedia.org/wiki/PBX>, 10 Marzo 2011.

[3] 3CX Ltd, “¿Qué significan los términos FXS y FXO?”, [http://www.3cx.es/voip-](http://www.3cx.es/voip-sip/fxs-fxo.php)

[sip/fxs-fxo.php](http://www.3cx.es/voip-sip/fxs-fxo.php), 2011.

[4] Wikipedia, “Session Initiation Protocol”,

[http://es.wikipedia.org/wiki/Session\\_Initiation\\_Protocol](http://es.wikipedia.org/wiki/Session_Initiation_Protocol), 2 Marzo 2011

[5] Wikipedia, “Introducción al protocolo IAX2”, <http://es.wikipedia.org/wiki/IAX>,

25 Noviembre 2010

[6] Wikipedia, “Introducción al protocolo H.323”,

<http://es.wikipedia.org/wiki/H.323>, 18 Junio 2010

[7] Wikipedia, “Introducción al protocolo MGCP”,

<http://es.wikipedia.org/wiki/MGCP>, 8 Febrero 2010

[8] Wikipedia, "Introducción a la Red Digital de Servicios Integrados",

<http://es.wikipedia.org/wiki/RDSI>, 11 Febrero 2011

[9] Wikipedia, "GNU General Public License",

[http://es.wikipedia.org/wiki/Licencia\\_p%C3%BAblica\\_general\\_de\\_GNU](http://es.wikipedia.org/wiki/Licencia_p%C3%BAblica_general_de_GNU),

21 Marzo 2011

[10] Wikipedia, "Introducción a GNU/Linux",

<http://es.wikipedia.org/wiki/GNU/Linux>, 28 Marzo 2011

[11] Wikipedia, "Interactive Voice Response", <http://es.wikipedia.org/wiki/IVR>,

9 Diciembre 2010

[12] Wikipedia, "Introducción al servidor de monitoreo Nagios",

<http://es.wikipedia.org/wiki/Nagios>, 29 Enero 2011

[13] Centurytechnical, "Servicios que nos ofrece el servidor Nagios",

<http://centurytechnical.com/22500098ac0e77402/0000009cb50b77415/0000009>

[cb50b7a724/index.html](http://centurytechnical.com/22500098ac0e77402/0000009cb50b77415/0000009cb50b7a724/index.html), 2010

[14 ] Steven Wittensy Stefan Nagtegaal, "¿Cómo monitorear servidores y equipo

de comunicaciones?", <http://aspiranteageek.wordpress.com/tag/nagios/>,

4 mayo 2010