UNIVERSIDAD CATÓLICA ANDRÉS BELLO

FACULTAD DE INGENIERÍA

ESCUELA DE INGENIERÍA DE TELECOMUNICACIONES

SISTEMA DE GESTIÓN Y MONITORIZACIÓN DE FALLAS PARA CLIENTES DE SANNET SOLUCIONES C.A

Trabajo Especial de Grado

Presentado ante la

UNIVERSIDAD CATÓLICA ANDRÉS BELLO

Como parte de los requisitos para optar al título de

INGENIERO EN TELECOMUNICACIONES

REALIZADO POR	**ARIADNE DOCTORS**
	ROGER VECCHIOTTI
PROFESOR GUÍA	**PABLO TERÁN**
FECHA	Caracas, Julio 2012

AGRADECIMIENTOS

A mi familia por su apoyo incondicional en todo momento, especialmente mis padres, Alejandro y Claudia, quienes han sido los pilares que me sostienen firme ante cualquier adversidad.

A Pablo Terán y José Bejarano de la empresa Sannet Soluciones por prestarnos el apoyo necesario y permitirnos ser parte de este proyecto.

Al personal de la empresa Servimeta por permitirnos utilizar su red como piloto de este proyecto.

A todas las personas que directa o indirectamente han tenido un impacto en mi vida a lo largo de la carrera,

Y finalmente, a los profesores de UCAB por colaborar con mi formación personal y profesional durante estos años.

Ariadne Doctors

A mis padres, Enzo y Mary, por haberme dado la oportunidad de cursar esta carrera, y haberme ayudado y apoyado siempre que lo he necesitado.

A mi hermana, Caro, por estar siempre ahí.

A mi novia, Mafe, por su apoyo incondicional y sus ánimos durante todo este proyecto, y por hacerme sentir tan querido.

A Pablo Terán y José Bejarano, por encargarse de la supervisión del trabajo realizado.

Y a Google, por hacerme la vida más sencilla.

Roger Vecchiotti

RESUMEN

En vista de que la empresa Sannet Soluciones C.A no ofrecía servicios de monitorización de redes a sus clientes, se planteó el proyecto de desarrollar un sistema de monitorización de redes que pudiese ser adaptado a las necesidades particulares de la red de cada cliente, de manera que pudiese ser ofrecido como un servicio adicional a los mismos. Teniendo esta situación como punto de partida se plantearon como objetivos el desarrollo de un sistema de monitorización que permitiese escalar los problemas para darles la atención necesaria, la detección temprana de situaciones no deseadas, el manejo de estadísticas para dar atención especial en los casos en los que se considere necesario, y finalmente gestionar las fallas en las redes de los clientes de la manera más cómoda posible.

Los objetivos planteados fueron alcanzados utilizando una metodología estructural y sistemática, se inició con una investigación teórica necesaria, que permitió tener los conocimientos necesarios para dar solución al problema, adaptando una herramienta de software libre al caso particular de uno de los clientes, para esto se tomó como punto de partida una lista de requerimientos planteados por la empresa. Una vez cubiertas las necesidades planteadas, se realizaron una serie de pruebas de funcionamiento a fin de cerciorar que la configuración realizada anteriormente fuese correcta. Los resultados obtenidos fueron exitosos, se logró implementar el sistema y dejarlo en funcionamiento, acorde con lo deseado inicialmente.

A partir de la realización de este trabajo especial de grado se obtuvo como principal conclusión que con las herramientas adecuadas y la dedicación necesaria es perfectamente posible lograr los objetivos planteados, y que Zabbix es una excelente herramienta de monitorización de redes si se utiliza correctamente.

ÍNDICE DE CONTENIDOS

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

INTRODUCCIÓN

A lo largo de la historia han existido las comunicaciones, pero en la medida que ha evolucionado la raza humana, han evolucionado igualmente los medios y métodos de comunicación. En los últimos años, con la aparición del internet, las redes se han vuelto una parte extremadamente importante de esa comunicación, estas permiten intercambio de cualquier tipo de información a corta y larga distancia, para que esto sea posible, se utilizan una serie de protocolos y especificaciones estándar, de manera que exista compatibilidad entre las redes a lo largo y ancho del globo.

Una parte muy importante del funcionamiento de las redes es su monitorización, ya que es de esta manera que se detecta casi cualquier problema o inconveniente, para que posteriormente sea solucionado.

Dos de los protocolos popularmente utilizados en esta área son ICMP y SNMP, ambos con gran importancia, ya que permiten la obtención de información referente a los estados de la conexión, recursos, entre otros.

Utilizando como base estos dos protocolos, se ha logrado realizar exitosamente el trabajo especial de grado presentado en las siguientes páginas, en conjunto con la implementación y adaptación de una herramienta de monitorización basada en software libre llamada Zabbix.

“Fundada en 2004, Sannet Soluciones C.A es una empresa venezolana de servicios profesionales orientada a las tecnologías de almacenamiento de datos, telecomunicaciones y sistemas operativos.” (Sannet Soluciones, 2012)

Entre sus clientes en Venezuela se cuenta la empresa Servimeta C.A, a la cual Sannet Soluciones C.A. presta servicio en el área de redes; es en la red de esta empresa, conformada por la conexión de equipos ubicados en distintas localidades en el territorio nacional, donde ocurre el desarrollo del trabajo presentado a continuación, a fin de mejorar el servicio actual, mediante la implementación de un

sistema de monitorización, el cual permitirá obtener información acerca de las actividades en la red permanentemente.

Al avanzar en el desarrollo del trabajo la meta principal fue satisfacer plenamente las necesidades del cliente, el mayor reto, lograrlo en ventanas de tiempo limitadas.

En la medida en que ocurrieron avances se presentaron algunas eventualidades que fueron solventadas, de manera que al finalizar el sistema quedó completamente operativo y en manos de los administradores de la red.

A continuación se presenta de una manera estructurada el proceso llevado a cabo para la realización de este trabajo, iniciando con el planteamiento de un problema puntual y los objetivos a cumplir, realizando una investigación teórica para tener las bases necesarias para seguir adelante, el planteamiento de una metodología compuesta por fases, el desarrollo de dichas fases y finalmente, los resultados obtenidos a partir del desarrollo y las conclusiones finales que quedaron como el más importante aprendizaje.

CAPITULO I

Planteamiento del proyecto

Las sociedades hoy en día hacen un continuo uso de las plataformas de comunicaciones, más que una opción se ha vuelto una necesidad, debido a la inmensa variedad de servicios que se pueden obtener en la adquisición y aplicación de redes LAN, así como en redes globales como la internet o las redes WAN; en un mundo cada día más globalizado, gracias a que en los últimos años todos los individuos de dichas sociedades han experimentado las consecuencias de una revolución tecnológica, el mundo tecnológico los ha involucrado en una sociedad red. Castells (2006) señala:

"La sociedad red es una estructura social hecha de redes de información, propulsada por las tecnologías de la información características del paradigma informacionalista. Por estructura social entiendo las disposiciones organizativas de los seres humanos en las relaciones de producción, consumo, experiencia y poder, tal como se expresan en la interacción significativa enmarcada por la cultura. Una red es un conjunto de nodos interconectados. Un nodo es el punto en el cual la curva se corta a sí misma. Las redes sociales son tan antiguas como la propia humanidad, pero han cobrado nueva vida bajo el informacionalismo porque las nuevas tecnologías realzan la flexibilidad inherente a las redes, al tiempo que solucionan los problemas de coordinación y gobierno que, a lo largo de la historia, lastraban a las redes en su competencia con las organizaciones jerárquicas."

Sabiendo esto, naturalmente surge la interrogante de cuan seguras sean las redes que se utilizan día a día, dada esta necesidad de que las comunicaciones en las redes sean seguras tanto en la privacidad, lo cual es un derecho común de todos los que participan en una comunicación privada, así como en la importancia de que no existan pérdidas de datos al ocurrir el intercambio de información entre los diferentes usuarios.

A través de los años las sociedades intelectuales en el área de comunicaciones, precisamente en IP, han ideado protocolos de administración de redes. Hoy en día existen diversas soluciones para estos problemas, como lo son una variedad de protocolos y programas que brindan potentes herramientas a los administradores de estas redes para evaluar y solucionar los problemas en las mismas.

1.1 Planteamiento del problema

Sannet Soluciones C.A. es una empresa de servicios profesionales que brinda soporte de Primer, Segundo y Tercer Nivel a terceros sobre la plataforma de comunicaciones y transporte de datos. Se encarga del despliegue y mantenimiento de redes, y cuenta con personal disponible para responder en caso de reportes de falla.

Debido a que Sannet Soluciones C.A. carecía de un sistema de gestión y escalamiento de casos que permitiese brindar soporte a sus clientes en caso de fallas de manera proactiva y realizar los niveles de escalamiento de acuerdo a los contratos, se desarrolló el mismo, mediante la adaptación de herramientas existentes a las necesidades del cliente, de manera que quedasen satisfechas las necesidades de todas las partes involucradas.

1.2 Objetivos

1.2.1 Objetivo general

- Desarrollar un sistema de gestión, monitoreo y escalamiento que permita el manejo de las fallas y alarmas de los diferentes clientes de Sannet Soluciones C.A.

1.2.2 Objetivos específicos

- Desarrollar un sistema de gestión que permita el manejo de las fallas de los clientes.

- Desarrollar un sistema de escalamiento de caso que permita gestionar las fallas de los clientes.

- Desarrollar un sistema de monitoreo para el manejo de estadísticas y niveles de disponibilidad de los diferentes clientes.

- Definir los parámetros y diferentes configuraciones para uno de los clientes de Sannet Soluciones C.A.

1.3 Alcance y limitaciones

- Se estableció un demo en la sede de Sannet Soluciones C.A en el cual se realizaron pruebas de funcionalidad.

- El trabajo se desarrolló en las oficinas de Sannet Soluciones C.A., por lo cual el horario estuvo limitado.

- Se presentaron limitantes en cuanto a disponibilidad de los equipos.

1.4 Justificación

Debido a que las personas y empresas necesitan estar constantemente comunicadas, es necesario crear estrategias en el ámbito de la seguridad y salud de las redes que comunican a los individuos dependientes de los distintos servicios prestados por empresas de telecomunicaciones, a su vez, estas empresas dependen de otras que les facilitan el trabajo en el área, tanto de proyecto como prestación de servicio técnico, por este motivo, Sannet Soluciones requería de una herramienta eficaz en la administración de redes, motivo por el cual se propuso el presente trabajo de grado. Hoy en día, disponer de sistemas adecuados para verificar la salud de los servicios prestados a terceros es vital tanto en el aspecto técnico como en el económico, ya que esto incluiría a su carpeta de presentación soporte de los diversos dispositivos necesarios para una red.

Es por ello que surgió la necesidad de realizar las investigaciones necesarias para obtener una buena base teórica y realizar el trabajo práctico con la mayor precisión y simplicidad posible, ya que todo sistema simple y con un buen rendimiento acarrea menores costos de implementación y mantenimiento, beneficiando así a las partes involucradas en los contratos de servicio.

La realización de este proyecto resulta beneficiosa para la empresa, ya que les permite ampliar su plataforma de servicios, lo cual se espera, tenga como consecuencia la ampliación de su cartera de clientes.

Desde el plano de la ingeniería de telecomunicaciones se obtiene un beneficio igualmente importante, ya que la posibilidad de conocer el estado de los equipos y enlaces que conforman una red es vital para ofrecer la máxima calidad de servicio al cliente, independientemente del área en que se desempeñe el mismo.

CAPITULO II

Marco Referencial

Este capítulo comprende los enfoques teórico, estudios y antecedentes sobre protocolos, tipos de mensajes y arquitecturas de red para su debida monitorización.

2.1 SNMP (*Simple Network Management Protocol* – Protocolo Simple de Gestión de Red)

Al tener una red es necesario poder monitorizar lo que ocurre con ella, desde el punto de vista de la administración. Uno de los factores más importantes que se deben considerar es el control y corrección de errores para poder ofrecer fidelidad. SNMP es un protocolo que permite de una manera muy simple hacer esto. A pesar de que ha presentado brechas en cuanto a seguridad, es uno de los más populares desde sus inicios, hasta la actualidad, de hecho, es el protocolo de facto para monitorización en internet. (Saperia, 2002)

Este es un protocolo de capa de aplicación (modelo OSI), está diseñado para facilitar el intercambio de información entre los equipos de la red y su administrador, a fin de que éste pueda supervisar su funcionamiento y sea posible tomar acciones en caso de fallas o de eventos indeseados. Como todos los protocolos, está estandarizado y tiene un lenguaje común, se conoce ampliamente que esto se hace para que los equipos puedan comunicarse entre sí, siempre y cuando sean compatibles, sin importar quienes sean el fabricante y el usuario.

Es uno de los protocolos de la familia TCP/IP y funciona sobre UDP, tiene un costo reducido, comparado con otros protocolos de monitorización, y los comandos son fáciles de entender y utilizar. La forma en que trabaja SNMP es a través del sondeo, el cual consiste en preguntar como gestor a un agente, esto significa enviar una solicitud pidiendo información sobre su estado físico, o bien pedir una actualización de su estado de trabajo, tras una solicitud, el gestor espera una respuesta

del agente, lo cual puede ser tanto una respuesta como una confirmación de cambio de estado. SNMP hace uso de los mensajes de interrupción, comúnmente llamados *trap*, en este tipo de mensaje los agentes pueden enviar información o alarmas a un gestor o nodo administrativo ante una eventual anomalía en la red.

2.1.2 Versiones

De manera global, hasta la actualidad, existen 3 versiones. Se empezó a desarrollar el protocolo en la década de 1980, en vista de la necesidad de una herramienta que permitiera la administración y gestión de las redes. Así nacieron HEMP y HEMS, así como otros protocolos que se publicaron antes de que la IETF decidiese que era necesario estandarizar, por lo que publicó un RFC, donde se especificó cómo debía desarrollarse el estándar, además de resaltar la importancia del mismo. En sus inicios, se definió que SNMP debía implementarse sobre SGMP, actualmente no es así. A continuación se describen las versiones que posteriormente se desarrollaron e implementaron. Los cambios relevantes en la evolución del protocolo han sido en el área de seguridad. (SNMPv2, 2011)

2.1.2.1. SNMPv1

Fue la primera versión, descrita en el RFC 1067, que posteriormente fue reemplazado por los RFC 1098 y 1157, respectivamente. La seguridad de esta versión se basó en comunidades con contraseñas simples sobre texto plano, lo cual significa que mientras se conozca la clave, se puede utilizar el equipo.

Luego se creó una versión SNMPsec, con la intención de incrementar la seguridad en SNMPv1, “en esta versión se incluyeron algunos mecanismos de criptografía y se proponía el uso de *parties*, que son entidades lógicas que pueden iniciar o recibir una comunicación SNMP.” (Arazo, 2011).

Los comandos (PDUs) utilizados son los siguientes:

Transmitidos por el NMS y recibidos por el agente:

- ***Get Request***, para solicitar atributos a un objeto.
- ***Get Next Request***, solicita el siguiente atributo del objeto.
- ***Set Request***, para actualizar uno o más atributos de algún objeto.
- ***Set Next Request***, para actualizar el siguiente atributo del mismo objeto.

Transmitidos por los agentes y recibidos por los NMS:

- ***Get Response***, devuelve los atributos solicitados con los comandos *GET* transmitidos por el NMS.
- ***Trap***, información de fallas en el agente.

2.1.2.2 SNMPv2

Se tomó como punto de partida SNMPsec, fue publicada por primera vez en 1992.

En esta versión se añadieron 3 comandos:

- ***Get Bulk Request,*** solo en SNMPv2, solicita un conjunto de valores en lugar de solicitarlos uno a uno.
- ***Inform Request***, descripción de la base local de información de gestión para intercambiar información de nodos administradores entre ellos.
- ***Report,*** para intercambio de información de control.

Además, se incluyeron algunos tipos de datos adicionales. “Sobre su arquitectura pueden construirse aplicaciones de gestión como alarmas y monitores de desempeño que hasta ahora quedaban fuera del estándar.” (Arazo, 2011).

Las versiones SNMPv2c, SNMPv2u y SNMPv2* aparecieron como mejoras de SNMPv2. SNMPv2c se utilizó mucho y su seguridad estaba basada en comunidades, donde se puede asociar un nombre a un perfil de la base de datos SNMP, junto con los derechos de acceso a dicho perfil. SNMPv2u tenía un modelo de acceso con usuarios y contraseñas. SNMPv2* se desarrolló como un protocolo distinto. No es compatible con el resto.

2.1.2.3 SNMPv3

Es la versión actual, en esta existen distintos módulos que tienen distintas tareas, independizando así los mecanismos de control de acceso, seguridad y gestión.

Se introdujeron el USM y el VACM, lo cual incrementó de manera notable la seguridad del protocolo, ya que permiten definiciones mucho mas especificas de los objetos accesibles.

Además se incluyeron mecanismos de autenticación y privacidad, así como una estructura sobre la cual se pueden configurar nombres de usuario, derechos de acceso y claves asociadas a estos, para mayor seguridad. (Valles, 2012)

2.2 Autenticación

Según el diccionario de la Real Academia Española, autenticar es “autorizar o legalizar algo”, “dar fe de la verdad de un hecho o documento con autoridad legal”. La autenticación en este caso es precisamente eso, aplicado a la rama de seguridad informática, es un proceso mediante el cual se pueden verificar las fuentes de la información, así como también se pueden verificar los usuarios, de manera confiable. Aplicándolo específicamente al tema de SNMP, los agentes y gestores necesitan poder autenticar las solicitudes y las respuestas de los otros equipos de la red para evitar amenazas, o para poder evadirlas en caso de que se introdujesen en la red.

Las amenazas más importantes que se pueden mencionar son, por ejemplo, la suplantación, que es simplemente cuando se reemplaza un equipo autorizado por un equipo intruso haciéndose pasar por éste. También se debe tener cuidado especial con la posibilidad de que exista modificación de información, esto puede ocurrir si un equipo ajeno a la red interviene en la comunicación entre dos equipos de la red y modifica alguna solicitud o alguna respuesta, puede suceder sin que exista suplantación.

2.2.1 Mecanismos de autenticación

2.2.1.1 HMAC (*Hash-based Message Authentication Code* – Código de Autenticación de Mensajes basado en Hash)

Es un mecanismo que, mediante el uso de algoritmos, calcula un MAC incluyendo una función *hash* y una clave secreta asociada.

MAC es el nombre que típicamente se le da a los mecanismos que sirven para verificar la integridad de la información transmitida en un enlace o red, o de la información almacenada en un medio poco confiable. Se usan entre dos entidades lógicas que comparten una clave que sirve para validar la información que transmiten entre ellas.

Una función *hash* es una función criptográfica que indexa datos grandes en datos pequeños. Se conocen en español como funciones resumen, ya que lo que hacen es resumir la información original. (Krawczyk, Bellane & Canetti, 1997)

El cálculo del HMAC ocurre de la siguiente manera:

> "El emisor calcula un resumen del mensaje basado en el contenido de la solicitud de respuesta y lo incluye en la cabecera de éste. El receptor calcula el resumen del mensaje recibido y verifica que concuerda con el enviado por el emisor. La clave solo es conocida

por emisor y receptor, por lo que cualquier modificación puede ser identificada.” (Arazo, 2011)

Las funciones hash que se pueden utilizar son MD5 y SHA-1.

2.2.1.2 MD5 (*Message-Digest algorithm 5* – Algoritmo de Digestión de Mensajes 5)

Ronald Rivest (MIT) lo desarrolló, basándose en MD2 y MD4. No tiene mejor rendimiento que MD4 en cuanto a velocidad, pero es mucho más seguro. Produce un número de 128 bits (32 dígitos hexadecimales) partiendo de un texto de cualquier longitud, y lo hace siguiendo una serie de pasos que se enumeran a continuación, obtenidos de Arazo, (2011):

i. **Adición de bits.** El algoritmo inicia añadiendo un relleno al mensaje que sea congruente con 448 mod 512. El relleno está formado por 1 bit “1” seguido por la cantidad necesaria de bits “0”. Se puede añadir entre 1 y 512 bits. El relleno se añade aunque el mensaje original ya sea congruente con 448 mod 512.
ii. **Longitud del mensaje.** Se almacena la longitud original del mensaje en los últimos 64 bits del relleno. Si el tamaño fuese mayor a 2^{64} solo se utilizan los 64 bits menos significativos.
iii. **Inicialización del búfer MD (*Message Digest*).** 4 registros forman el búfer, estos se denominan A, B, C y D y se inicializan siempre con los siguientes valores hexadecimales:
   A = 67 45 23 01
   B = EF CD AB 89
   C = 98 BA DC FE
   D = 10 32 54 76
iv. **Procesado de mensaje en bloques de 16 palabras.** Cada bloque de 512 bits se divide en 16 palabras de 32 bits. El algoritmo opera en cada una de éstas por turnos, haciendo 64 operaciones que consisten de 4

etapas llamadas rondas, cada ronda tiene una función asociada y en cada ronda se utiliza una distinta. Las funciones son las siguientes:

$$F(X,Y,Z) = (X \wedge Y) \vee (\neg X \wedge Z)$$

$$G(X,Y,Z) = (X \wedge Z) \vee (Y \wedge \neg Z)$$

$$H(X,Y,Z) = X \oplus Y \oplus Z$$

$$I(X,Y,Z) = Y \oplus (X \vee \neg Z)$$

Los símbolos $\oplus, \wedge, \vee, \neg$ denotan las funciones lógicas: XOR, AND, OR y NOT, respectivamente.

v. **Salida.** El resumen del mensaje es el valor que contienen los registros A, B, C y D después de las operaciones. Empieza en el bit menos significativo de A y termina en el más significativo de D.

2.2.1.3 SHA-1 (*Secure Hash Algorithm-1* – Algoritmo Seguro de Hash-1)

Es un algoritmo de la familia de los SHA, que es una familia de funciones *hash* creadas por la Agencia de Seguridad de EEUU. Es el algoritmo más usado de la familia a pesar de que se han encontrado fallas de seguridad, y a pesar de que no se han encontrado ataques exitosos a ninguna de las SHA-2, está en desarrollo una variante SHA-3 que se espera sea publicada este año. (Arazo, 2011)

Esta función produce un resumen de 160 bits, su funcionamiento está basado en MD5. Tienen el mismo número de pasos y los 2 primeros son idénticos, las modificaciones se muestran a continuación:

i. **Inicialización del búfer SHA.** En lugar de los 4 registros que existen en MD5, hay 5 registros A, B, C, D y E. A, B, C y D se inicializan igual que en MD5, el registro E se inicializa con el siguiente valor hexadecimal:

E = C3 D2 E1 F0

ii. **Procesado del mensaje de bloques de 16 palabras.** También se utilizan 4 funciones no lineales, una para cada etapa. Cada una de las etapas está compuesta por 20 operaciones que incluyen una de las 4 funciones (la segunda y la última etapa usan la misma función). Se define también una tabla de constantes compuesta por 4 valores diferentes que se mantienen a lo largo de las operaciones. Dichas operaciones modifican los valores de los registros.

El mensaje se divide en bloques de 16 palabras de 32 bits cada una, a partir de estas palabras se hace una extensión, obteniendo 80 palabras.

iii. **Salida.** El resumen del mensaje es el valor que contienen los registros A, B, C, D y E.

2.3 Privacidad

La privacidad se puede definir como “Ámbito de la vida privada que se tiene derecho a proteger de cualquier intromisión.” (Real Academia Española). En este caso se refiere a la habilidad de mantener información en secreto o poder seleccionar a quien es revelada, la que se maneja entre el gestor y el resto de los equipos de la red. Para lograr esto se utilizan herramientas de encriptación para que la información no viaje en texto plano y no sea fácil acceder a ella sin tener autorización. Tener autorización significa también conocer el mecanismo de privacidad utilizado para poder hacer la operación necesaria para ver el mensaje completo. Para lograr el proceso de encriptación, los algoritmos que se utilizan son AES, DES y 3-DES. A continuación se da una breve explicación de los mismos.

2.3.1 Mecanismos de Privacidad

2.3.1.1 DES (*Data Encryption Standard* – Estándar de Cifrado de Datos)

Es un sistema de cifrado por bloques, es simétrico, los bloques son de 64 bits. La clave que se utiliza es de la misma longitud, pero como un byte se utiliza como control de paridad, solo son efectivos 56 bits de dicha clave.

2.3.1.2 3-DES (*Triple Data Encryption Standard* – Estándar de Cifrado de datos tripe)

Es simplemente el algoritmo que resulta de aplicar 3 veces consecutivas el algoritmo DES.

2.3.1.3 AES (Advanced *Encryption Standard* – Estándar de Cifrado Avanzado)

Al igual que los 2 anteriores, es un algoritmo de cifrado por bloques, en este caso son bloques de 128 bits. (Arazo, 2011)

2.4 Elementos de la Arquitectura

Ya que el protocolo SNMP es parte de un modelo de gestión de redes tipo TCP/IP se deben conocer con anticipación varios elementos de dicho modelo para una mejor comprensión de los componentes básicos en una estructura de administración de redes.

2.4.1 NMS (*Network Management Station* – Estación de Gestión de Red)

Son las estaciones de Gestión (hardware), son las interfaces entre el operador humano (administrador de red) y el sistema de gestión de la red, cuenta con una base de datos que contiene toda la información necesaria para la gestión y que se obtiene de todas las bases de datos de las entidades a gestionar.

Un NMS “Suele ser un equipo potente con un CPU veloz, mucha memoria y gran espacio de disco. Generalmente, un solo equipo se encarga de monitorizar todos los elementos de la red.” (Arazo, 2011)

La función de los NMS es ejecutar las aplicaciones que controlan y supervisan los dispositivos que se administran en la red. En cualquier red debe existir por lo menos un equipo gestor. La supervisión y el control se logran utilizando comandos básicos de lectura, escritura, notificación y las llamadas operaciones transversales, estas son utilizadas para determinar qué variables puede o no soportar un equipo, así como para recolectar información de cualquiera de las tablas variables.

2.4.2 Agente

Es un componente de software que reside en los nodos administrados (elementos de la red, *routers*, *switches*, módems, computadores, impresoras…). Su trabajo consiste en responder las solicitudes que recibe, dependiendo del tipo de solicitud, debe devolver un valor o modificarlo (en el dispositivo sobre el que actúa).

2.4.3 MIB (*Management Information Base* – Base de Información de Gestión)

Es una base de datos en la que están contenidos todos los objetos que se van a administrar en la red junto con sus características. La información se almacena en forma de árbol, de manera jerárquica.

Tiene 8 niveles de registro:

Grupo	Variable	Significado
*System (**sys**)*	*sysUpTime*	Tiempo desde el ultimo arranque
*Interfaces (**intf**)*	*ifNumber*	Número de Interfaces.
*Interfaces (**intf**)*	*ifInErrors*	Número de paquetes entrantes en los que el agente ha encontrado error.
*Address Traslation (**add trs**)*		
*Internet Protocol (**ip**)*	*ipInReceives*	Numero de paquetes recibidos
*Internet Control Message (**icmp**)*	*icmpInEchos*	Número de solicitudes ICMP recibidas.
*Transmision Control Protocol (**tcp**)*	*tcpInSegs*	Número de paquetes TCP recibidos.
*User Datagram Protocol (**udp**)*	*udpInDatagrams*	Número de datagramas UDP recibidos.

Tabla 1. Niveles de registro del MIB.
Fuente: Arazo, 2001.

“Para que un dispositivo pueda ser manejado a través de SNMP debe incorporar una MIB que incluya todos los objetos que van a ser accesibles desde un *manager* instalado en un NMS. La MIB contiene la forma en cómo pueden ser accedidos esos objetos, definiendo la relación existente entre ellos, su estructura y el tipo de datos al que pertenecen.” (Arazo, 2011)

Existen MIBs estándar que tienen una carga predeterminada de objetos que están en todas las redes comúnmente, pueden extenderse añadiendo los objetos que se consideren necesarios. Las MIBs se deben guardar en un lugar determinado, de manera que puedan ser encontradas por el gestor y los agentes cuando arrancan.

También existen MIBs con interfaces graficas, lo cual hace mucho más sencilla su utilización, ya que se puede acceder a los objetos como en una estructura común de directorios. Cuando no tienen interfaz grafica, deben accederse con

comandos. Si se tiene la MIB de un objeto, no es necesario solicitar los objetos con los comandos *get* ya que se pueden acceder y visualizar de forma sencilla, esto también disminuye el tráfico y mejora el rendimiento del sistema.

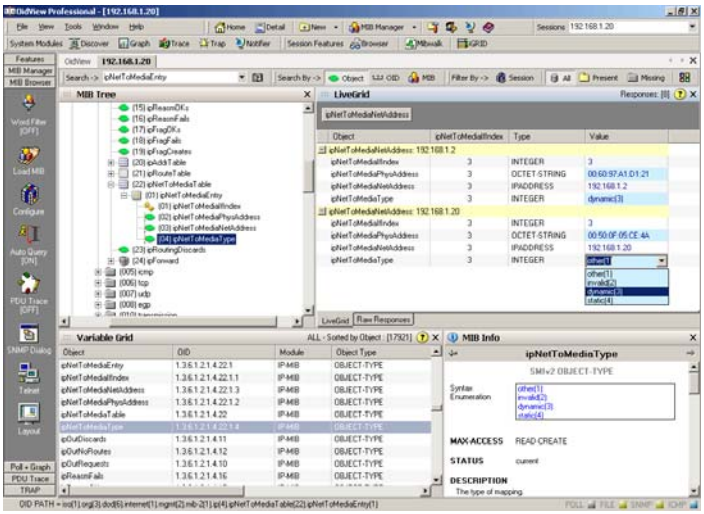


Figura 1. MIB con interfaz gráfica.
Fuente: http://www.oidview.com/images/mibbrowser-full.gif

2.4.3.1 Árbol MIB

La estructura de las MIBs es un árbol, una estructura en cascada, donde los nodos, que contienen distintos objetos, están ordenados por niveles, según la tarea que deben desempeñar. Se conoce como “el árbol MIB”.

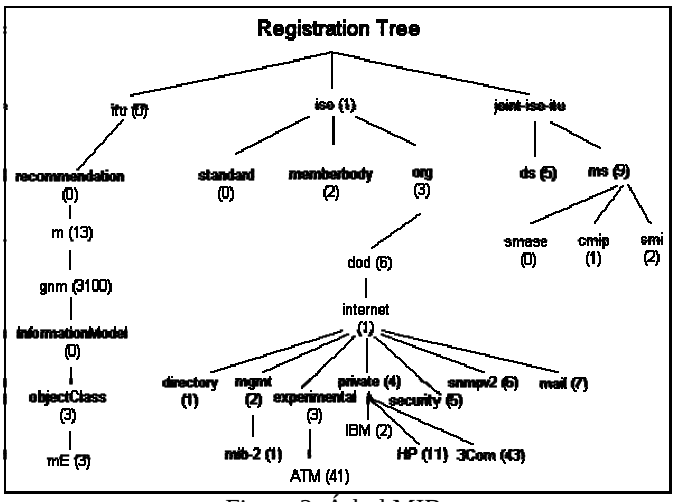


Figura 2. Árbol MIB.
Fuente: http://www.arcesio.net/osinm/osinminformacion.html

En la figura se puede observar claramente la división en nodos. La zona superior está reservada para las organizaciones estándares. La rama principal del árbol es el nodo ISO, dentro de ésta hay una rama que incluye el resto de las organizaciones (org). El resto de las ramas dentro del nodo ISO son para su propio uso; la misma lógica aplica para los demás nodos y sus ramas.

Ya que el internet es una herramienta extremadamente importante en la actualidad para todos, se describe a continuación éste nodo.

- ***directory***, reservada para memorias futuras que discutan sobre la organización de la estructura OSI usada en internet.
- ***mgmt,*** identifica los objetos definidos en documentos aprobados por IAB. Aquí se almacenan las MIBs estándar.
- ***experimental,*** almacena las MIBs que están en fase de pruebas.
- ***prívate,*** almacena las MIBs definidas de forma unilateral. Dentro de éste existe el nodo *enterprises* donde hay un espacio reservado para que cada fabricante conocido almacene sus MIBs.
- ***security,*** almacena objetos relacionados con seguridad del protocolo.
- ***snmpv2,*** para los nuevos objetos de SNMPv2.
- ***mail,*** para los objetos de correo electrónico.

Los tres últimos fueron añadidos a posteriori, originalmente solo se definieron las primeras 4 ramas. (Arazo, 2011)

Los niveles definidos como superiores van desde el *root*, el cual es la raíz de todos los demás, hasta las ramas *mib-2* y *enterprises*.

En la rama *mib-2* se describen los grupos de objetos.

2.4.4 ASN.1 (*Abstract Syntax Notation.1* – Notación de Sintaxis Abstracta.1)

Notación de sintaxis abstracta 1, es un estándar de notación que se desarrolló como parte del modelo OSI, como todos los estándares, describe normas, en este caso, son normas para representar, codificar, decodificar y transmitir datos, también describe tipos de datos que se pueden utilizar para ello. Es similar a un lenguaje de programación de alto nivel, lo cual lo hace sencillo de usar. Es comúnmente utilizado para describir protocolos, en este caso se menciona su existencia debido a que la sintaxis utilizada en SNMP es un subconjunto de ésta. (Arazo, 2011)

2.4.5 SMI (*Structure of Management Information* – Estructura de Información de Gestión)

Definida dentro del ASN.1, es donde se especifican todas las normas sintácticas para describir los objetos administrables dentro de la MIB, así como los tipos de datos y notaciones permitidos.

Garantiza que los distintos objetos que se van a administrar comparten un lenguaje común y se pueden comunicar exitosamente. Está diseñada para:

> "Definir las características generales asociadas a los objetos de una MIB, cómo deben ser descritos. Definir los tipos de datos que se pueden utilizar al crear los objetos en la MIB. Describir la estructura jerárquica para identificar los objetos. Definir la información de administración asociada a cada objeto de la MIB." (Arazo, 2011)

Símbolo	Significado
: : =	Asignación
|	Alternativa
{ }	Inicio y final de una lista
[ ]	Inicio y final de una etiqueta
( )	Inicio y final de un subtipo
-	Numero con signo
- -	Comentario
. .	Rango

Tabla 2. Símbolos ASN.1

Fuente: Arazo, 2011

Elemento	Convención
Tipo	Inicial en mayúscula
Valor	Inicial en minúscula
Macro	Todas mayúsculas
Módulo	Inicial en mayúsculas
Palabra clave	Todas mayúsculas

Tabla 3. . Convenciones ASN.1 para escritura de nombres de los tipos de elementos en una MIB.
Fuente: Arazo, 2011

2.5 Mensajes

Los mensajes SNMP están conformados por tres campos:

1. **Versión.** Donde se indica la versión del protocolo que se está utilizando.
2. **Comunidad.** Indica la comunidad que se utiliza.
3. **PDU.** Contiene la información específica del comando que generó el mensaje. Este campo puede tener configuraciones distintas según el comando enviado.

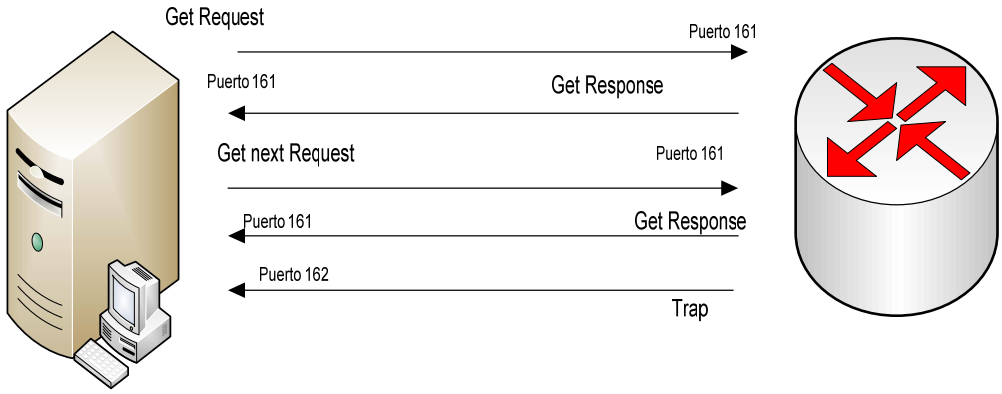


Figura 3. Intercambio de Mensajes.
Fuente: Elaboración propia

2.6 Redes de información

Una red se puede definir como una tela de araña, en este caso, los hilos son cables y los puntos de agarre computadoras o dispositivos, teniendo esto en cuenta, una red es la interconexión de múltiples equipos (computadores, *router*, *switches*…), debido a la necesidad de un intercambio de información constante, ya que hoy en día

dependemos de ello como nunca antes en la historia. Como ya se ha mencionado, en internet existen diversos protocolos, los cuales tienen una o varias tareas especificas en distintas áreas, para que se lleve a cabo un buen traslado de información (efectivo y seguro) a través de los distintos medios físicos a utilizar, lógicamente podemos imaginar que existen redes de distintos tamaños, lo cual es cierto y cada una tiene características y funciones distintas, éstas se definirán a continuación.

2.6.1 Redes de área local

Comúnmente llamadas redes LAN (*Local Area Network),* son redes de datos de alta velocidad y, como su nombre lo indica, son locales, lo cual quiere decir que están geográficamente limitadas, frecuentemente se implementan en oficinas o edificios. Al igual que cualquier otra red, su función es interconectar los equipos que se desea que la conformen, bien sea para que intercambien información o para poder acceder unos a otros de manera remota y utilizar los servicios que ofrece cada uno sin necesidad de que se implementen varios equipos iguales, un ejemplo de esto es una impresora de uso común en una oficina, todos los agentes pueden imprimir los documentos que generan en sus equipos, sin necesidad de cada uno tenga una impresora en su puesto de trabajo ya que la impresora está conectada a la red y se puede utilizar de manera remota. Este tipo de redes tienen protocolos y tecnología de transmisión diferentes a las demás, así mismo, pueden utilizar distintas topologías, dependiendo de las necesidades que se deseen cubrir.

“Las LANs están restringidas por tamaño, es decir, el tiempo de transmisión en el peor de los casos es limitado y conocido de antemano. El hecho de conocer este límite permite utilizar ciertos tipos de diseño, lo cual no sería posible de otra manera. Esto también simplifica la administración de la red.” (Tanenbaum, 2003)

2.6.1.2 Ethernet

Se puede considerar este como el estándar más usado en las redes de área local para la conexión de sus equipos, está definido por el estándar IEEE 802.3

publicado en 1983, es compatible con las capas 1 y 2 del modelo referencial OSI, este estándar funciona con modulaciones banda base o banda ancha, existen diversas tecnologías que han evolucionado a partir de esta, las cuales solo presentan ligeras modificaciones, ya sea en cuanto a velocidad o medio físico que utilizan.

Las velocidades de transmisión en Ethernet son las siguientes:

- **Ethernet:** 10Mbps.
- **FastEthernet:** 100Mbps.
- **GigabitEthernet:** 1000Mbps.
- **10GigabitEthernet:** 10000Mbps.

Los medios físicos de transmisión usados en este protocolo son los siguientes:

- Cable coaxial grueso o delgado.
- Cable UTP categoría de la 3 a la 6.
- Fibra óptica, multimodo o monomodo.

En la siguiente tabla se describe la relación entre las velocidades de transmisión y los medios físicos a usar.

Velocidades Mbps	Modulación	Medio	Descripción
10	BASE	2 5 T	Coaxial delgado Coaxial grueso Cable UTP
100	BASE	T F	Cable UTP Fibra óptica
1000	BASE	T S,L	Cable UTP Fibra óptica

Tabla 4. Tecnologías Ethernet.
Fuente: http://www.ie.itcr.ac.cr/faustino/Redes/Clase8/4.2Ethernet.pdf

2.6.2 Redes de Área Metropolitana

Son las denominadas redes MAN (*Metropolitan Area Network)*, son una versión extendida a nivel de geográfico de las redes LAN, lo que se quiso hacer con este tipo de red fue usar una arquitectura muy similar a las redes de área local pero con conexiones privadas de largas distancias, este tipo de red fue la base de la evolución de lo que actualmente se conoce como tecnologías 4G como WiMAX y LTE, las cuales tienen como característica principal el alcance de altas velocidades a largas distancias.

2.6.3 Redes de Área Amplia

Son aquellas redes que operan más allá de las distancias geográficas en las que puede operar una red LAN, este tipo de red es necesario para conectar distintas redes privadas, comúnmente el acceso a éstas se hace a través de un *router*, pueden tener una extensión que abarque una ciudad o ciudades, incluso continentes, por lo general, las empresas no pueden crear enlaces directos con otras sucursales en su misma ciudad o alrededor del mundo, por lo que se ven en la necesidad de pagar a un proveedor de servicios de redes de área local, para poder conectarse a otras sucursales o otros *hosts* alrededor del planeta.

"Las tecnologías LAN proporcionan velocidad y rentabilidad para la transmisión de datos dentro de organizaciones, a través de áreas geográficas relativamente pequeñas. Sin embargo, hay otras necesidades empresariales que requieren la comunicación entre sitios remotos, incluidas las siguientes:

- Los empleados de las oficinas regionales o las sucursales de una organización necesitan comunicarse y compartir datos con la sede central.

- Con frecuencia, las organizaciones desean compartir información con otras organizaciones que se encuentran a grandes distancias. Por ejemplo, los fabricantes de software comunican periódicamente información sobre

productos y promociones a los distribuidores que venden sus productos a los usuarios finales.

- Con frecuencia, los empleados que viajan por temas relacionados con la empresa necesitan acceder a la información que se encuentra en las redes corporativas.” (Cisco Systems, Inc., 2007)

A continuación se presenta una representación gráfica de una red WAN.

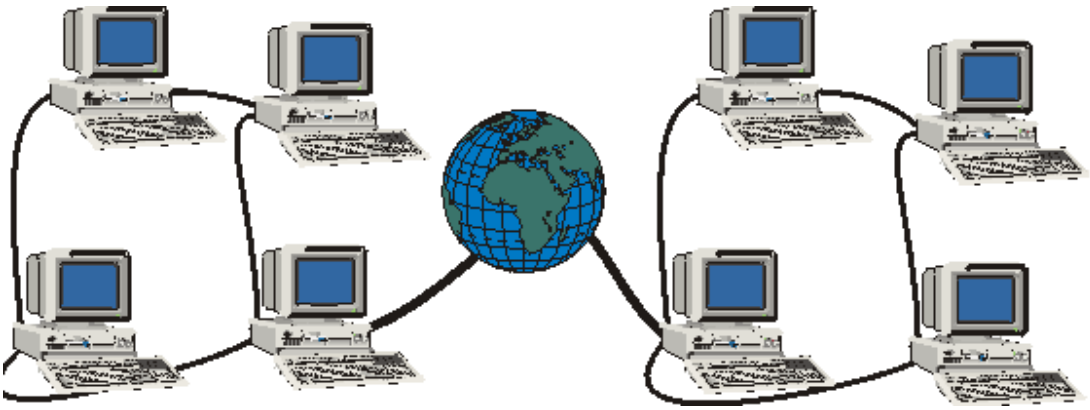


Figura 4. Red de área amplia.
Fuente: http://profecarolinaquinodoz.com/principal/wp-content/uploads/2009/04/wan1.gif

2.6.4 Redes virtuales privadas

En los últimos años, el costo de implementación de un enlace físico dedicado ha incrementado considerablemente, por lo que en muchos casos se han visto afectadas las empresas que los requieren, por ejemplo, empresas con distintas sedes o sucursales en el territorio nacional. Como solución a este problema se han implementado redes privadas virtuales, éstas establecen conexiones punto a punto sin necesidad de la arquitectura que requiere un enlace dedicado, ésta conexión se puede lograr a través de una red WAN. (Gutierrez, Sancho & Casas)

Concretamente, una VPN (*Virtual Private Network)* es una red virtual que se puede implementar sobre una red física, que bien puede ser internet o WAN, independientemente del protocolo de enlace de datos que esté utilizando, por ello,

representa un bajo costo para las empresas comparado con un enlace dedicado de un proveedor de servicios.

2.6.4.1 Funcionamiento general de una red virtual privada

Su funcionamiento general es simple, dado que éstas crean una especie de túnel privado, donde ambas partes negocian el tipo de autenticación y encriptación de datos a utilizar ya que es una conexión privada usando un medio de transmisión público, como el internet.

Por lo general, las empresas necesitan implementar servidores VPN en su red para aplicar este servicio, el mismo podría tener la operatividad como *gateway* de la LAN que se desea conectar con su par a distancia, o puede ser simplemente un computador más de la red. También se debe instalar un cliente VPN, software que sirve para comunicar los equipos de la red con el servidor VPN y, una vez efectuado éste proceso, poder acceder al túnel privado entre los dos puntos de interés.

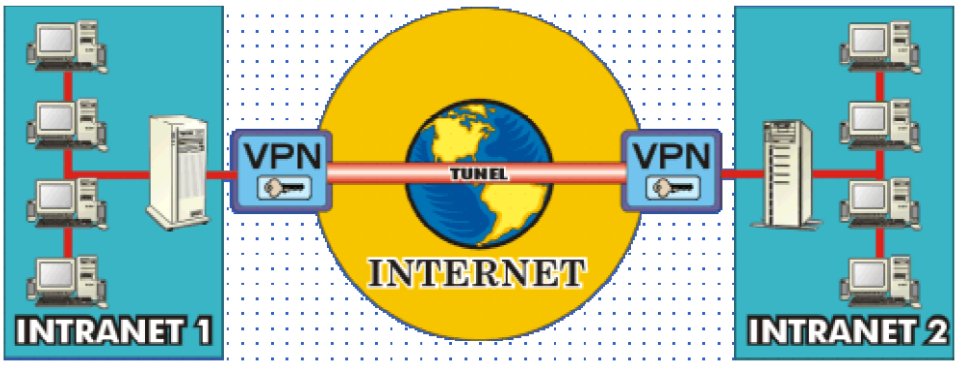


Figura 5. Esquema de una VPN.
Fuente: http://www.redes-linux.com/manuales/vpn/Estudio_VPN.pdf

2.7 Elementos de Red

2.7.1 *Routers*

Los *routers* son equipos que se utilizan para "enrutar" la información, se encargan de marcar o determinar la vía por la cual debe viajar dicha información, mediante una serie de reglas y estándares, y tras una serie de configuraciones; lo hacen de forma automatizada, haciendo el trabajo del administrador de la red increíblemente más sencillo. Mediante el uso de estos equipos se pueden interconectar redes, bien sea LANs, WANs, o cualquier tipo de red, con otro tipo, por ejemplo, la red local de una casa a la internet. Debido a las características antes mencionadas (interconectar y enrutar), los *routers* son los responsables de que los paquetes sean entregados a través de las redes en las que estos trabajan de manera inequívoca. Para lograr esto, son capaces de crear rutas alternas en caso de que falle la ruta principal, de esta manera garantizan el servicio. Así mismo, proveen servicios integrados de datos, voz y video, y dan prioridad a los paquetes, por lo que se pueden utilizar para transmisiones en tiempo real sin ningún problema, además, ya que tienen la capacidad de permitir o denegar el acceso de paquetes a la red colaboran con la seguridad de la misma. (Cisco Systems, Inc., 2007)

Estos equipos son extremadamente similares a los computadores, tienen CPU (*Central Processing Unit* – Unidad de Procesamiento Central*)*, memorias ROM (*Read-Only Memory* – Memoria de Solo lectura) y RAM (*Random Access Memory* – Memoria de Acceso Aleatorio), y cuentan con un sistema operativo, mediante el cual se pueden programar y configurar.

Los *routers*, en general, son invisibles para el usuario. Entre el computador que una persona opera y el servidor al que ingresa puede haber cualquier cantidad de *routers* que se encargan de que la conexión pueda realizarse. Debido a que conectan muchas redes, tienen varias interfaces, cada una de las cuales está identificada con

una dirección IP correspondiente con la red en la que se encuentre. Es común que tengan interfaces para WAN e interfaces para redes LAN.

Para lograr hacer su trabajo y determinar las rutas para los paquetes, la herramienta que utilizan es la tabla de enrutamiento, en esta tabla están contenidos una serie de detalles que permiten al *router*, utilizando distintos algoritmos, decidir cuál es la ruta más conveniente en cada caso, también incluye la dirección de la interfaz por la que saldrá el paquete. Dependiendo de los protocolos que utilicen los enlaces a los que se conectan estas interfaces, los paquetes pueden tener formatos diferentes (Ethernet, PPP…) cada uno de los protocolos tiene sus propios estándares de encapsulación de paquetes, por lo que el *router* puede hacer el proceso de des-encapsulado y re-encapsulado dependiendo de los medios a los cuales está conectado.

Para aprender sobre las redes remotas y poder hacer el direccionamiento, los *routers* utilizan rutas estáticas y enrutamiento dinámico, las rutas estáticas suelen utilizarse cuando se trabaja con una red de pocos equipos, cuando la red tiene un único punto de salida a internet o en situaciones que no requieran de enrutamiento dinámico; por otra parte, el enrutamiento dinámico es una herramienta extremadamente útil cuando se desea acceder a redes remotas, éste tipo de enrutamiento también actualiza las tablas de enrutamiento cada cierto tiempo, y, de haber cambios en la topología, los refleja modificando la misma.

Los *routers* cuentan con una serie de interfaces físicas mediante las cuales se pueden administrar, se les conoce como puertos de administración, éstos puertos no se utilizan para el envío de paquetes. Entre éstos el más popular es el llamado puerto de consola, allí se puede conectar un computador donde se ejecute un software que emule el terminal que se necesita para configurar el *router* sin necesidad de acceder a la red, o bien se puede conectar un terminal. La razón por la que es necesario un terminal es que los *routers* tienen un sistema operativo que administra sus recursos (asignación de memoria, procesos, seguridad, sistemas de archivos y cualquier otro recurso de software además de los recursos de hardware). Además de las interfaces de

administración previamente mencionadas, los enrutadores tienen interfaces de conexión, a través de éstas se conectan a las redes, pueden ser de distintos tipos, para conectare a distintos tipos de red, todas las interfaces que se mencionan en este párrafo son físicas y se encuentran en la parte externa del *router*. Al igual que un sinfín de equipos electrónicos, los *routers* cuentan con indicadores LED destinados a cumplir distintas funciones, generalmente se cuenta con un LED que indica si el equipo esté encendido o no, uno por cada interfaz física que indica si existe una conexión y con uno por cada interfaz física que indica que se está enviando/recibiendo información, esto no es una condición *sine qua non*, dependiendo del tipo de *router* (marca, modelo) éstas características pueden variar.

Entre las interfaces para conexión de red comúnmente se observan LAN y WAN, las cuales se utilizan, como es de suponerse, para conectar el *router* a una LAN o a una WAN, respectivamente.

2.8 ICMP (*Internet Control Message Protocol* – Protocolo de Mensajes de Control de Internet)

El protocolo ICMP es uno de los protocolos del grupo de protocolos de internet, del cual forman parte todos los protocolos usados para internet y redes similares, envía mensajes con características similares a las de un datagrama UDP, pero con un formato más simple, ya que en vez de contener los datos enviados con información de usuario, se basa en controlar si el paquete no puede alcanzar un destino o si su vida a expirado. Se puede decir que es un protocolo orientado al manejo de errores, es una herramienta de gran valor para los administradores de red, ya que de esta forma se puede confirmar o no la conexión entre varios *hosts* y al mismo tiempo muestra información del problema.

Existen distintos tipos de mensajes ICMP, entre los cuales destacan los mensajes echo request (8) y echo reply (0), los cuales son utilizados para hacer el comúnmente conocido “ping”, bien sea para comprobar el estado de un enlace local,

un enlace remoto o el estado de la tarjeta de red del propio dispositivo. (El Protocolo ICMP, 2012)

2.9 Zabbix

Zabbix es un software de fuente abierta, esto quiere decir que está basado en software libre. Esta herramienta permite controlar numerosos parámetros de red, así como la salud e integridad de los distintos servicios o dispositivos. Zabbix utiliza distintos mecanismos de notificaciones, como SMS y correos electrónicos. (Zabbix Manual, 2012)

Zabbix es libre de costo, esto significa que su código fuente se distribuye gratuitamente y está disponible para el público en general. En la página oficial de Zabbix se puede encontrar que ofrece este producto como se muestra a continuación:

Zabbix ofrece

- La detección automática de servidores y dispositivos de red
- Monitorización con administración centralizada WEB
- Soporte para los mecanismos de captura
- Software de servidor para Linux, Solaris, HP-UX, AIX, BSD libres, BSD Open OS X
- Agentes de alto rendimiento (software de cliente para Linux, Solaris, HP-UX, AIX, BSD libres, BSD Open, OS X, Tru64/OSF1, Windows NT 4.0,Windows 2000, Windows 2003, Windows XP, Windows Vista)
- Autenticación de usuario segura
- Permisos de usuario flexibles
- Interfaz basada en web
- Notificación flexible de correo electrónico de eventos predefinidos
- Alto nivel (de negocios) vista de los recursos controlados

- Registro de auditoría

2.10 Cacti

Cacti es una herramienta RRD, almacena toda la información necesaria para crear gráficos y llenarlos con datos en una base de datos MySQL. Maneja el acopio de información y soporta SNMP.

Funciona recopilando la información de los *paths* o caminos que se le indican. También se pueden escribir *scripts* que devuelvan información y utilizarla para realizar los gráficos. Además permite crear usuarios para que cada uno maneje los gráficos con configuraciones particulares. (*What is Cacti?, 2012)*

2.11 Nagios

Nagios es un sistema de monitoreo que permite a organizaciones identificar y resolver problemas de infraestructura IT antes de que afecten procesos críticos. (*About Nagios*, 2012)

Ofrece programas de entrenamiento para instalar, configurar y manejar un despliegue de Nagios. Los programas funcionan en línea y no son gratuitos, prometen una capacitación completa. Existen las opciones de realizar el entrenamiento de manera interactiva (*Live Nagios Training* – Entrenamiento Nagios En Vivo) o ver las lecciones cuando al estudiante le convenga (*Self-Paced Training* – Entrenamiento al Propio Ritmo). (*Nagios Training*, 2012)

Nagios *Enterprises*, quienes patrocinan oficialmente Nagios, ofrece servicios de consultoría profesional para organizaciones, antes, durante y después del despliegue de Nagios incluyendo:

- Problemas con desempeño de *troubleshooting*.
- Arquitectura de soluciones avanzadas de monitoreo.

- Integración de Nagios a otras soluciones IT.
- Personalización de Nagios para satisfacer necesidades específicas.
- Desarrollo de mejores prácticas para diferentes escenarios de supervisión. (*Consulting and Implementation Services*, 2012)

Este software es capaz de monitorear una infraestructura completa para asegurar que los procesos funcionan de manera apropiada. En caso de fallas, Nagios puede alertar al personal el problema, permitiendo la rápida resolución de las mismas.

CAPITULO III

Marco Metodológico

A continuación se presenta la metodología utilizada para el desarrollo de este proyecto, esta consistió en cinco fases: Investigación, ensayos, implementación del software, configuración de parámetros y pruebas de funcionamiento.

3.1 Fase 1: Investigación

La fase de investigación consistió en obtener los conocimientos e información necesarios para que pudiese ser puesta en marcha la implementación de manera exitosa. En este proceso se investigó detalladamente acerca de los protocolos que se utilizarían como herramientas de monitorización. También se dedicó tiempo a recolectar información acerca de los equipos que formarían parte de la red a monitorizar.

Se inició con una investigación exhaustiva del protocolo simple de gestión de red (SNMP), ya que sería este en el que se basaría el sistema a desarrollar. Allí se incluyó una descripción detallada del protocolo, versiones, su funcionamiento y los comandos más utilizados; además se describen características importantes como los métodos de autenticación y mecanismos de privacidad disponibles. Se realiza una descripción de la arquitectura del protocolo, elementos, sintaxis, estructura y mensajes.

Una vez cubierta la información referente al protocolo SNMP se procedió a realizar una descripción de las redes, que son y qué tipos existen actualmente (locales, metropolitanas, amplias). Se incluyen en la investigación algunos de los estándares utilizados al implementar dichas redes (Ethernet, *Frame-Relay*…). Se introdujo el concepto de redes privadas virtuales.

Una vez completada la fase previamente descrita se procedió a investigar un poco acerca del hardware, los elementos que conforman la red. Se hizo una

especificación de los equipos (*routers*, firewall, *switches*), su funcionamiento y el papel que representan en la red.

3.2 Fase 2: Ensayos

Al culminar la fase de investigación teórica se realizó la selección del software a utilizar. Acto seguido se procedió con la instalación del mismo en las máquinas personales a fin de conocer su funcionamiento y familiarizarse con sus capacidades. De igual manera, se hicieron prácticas con la sintaxis hasta lograr finalmente la configuración correcta de los parámetros a monitorizar.

Todo lo anterior se realizó utilizando una máquina virtual con sistema operativo Ubuntu, debido a que en la sede del cliente se utilizaría una máquina con sistema operativo Linux.

3.3 Fase 3: Implementación del software en la red

Para la instalación en la red del cliente se utilizó la máquina virtual con la cual se llevó cabo la fase de ensayos, a fin de simplificar la instalación.

Se realizaron modificaciones correspondientes con los nombres de usuario y las claves de acceso al sistema, las cuales fueron elegidas por el administrador de la red.

3.4 Fase 4: Configuración de parámetros a monitorizar

En una reunión con el administrador de la red del cliente, se determinaron todos los elementos que se desearían monitorizar en la red y los parámetros específicos de éstos sobre los cuales se requería la obtención de datos.

3.5 Fase 5: Pruebas de funcionamiento

Una vez realizadas todas las configuraciones necesarias para la monitorización de los equipos y sus parámetros correspondientes, así como la configuración de las alarmas y notificaciones de falla requeridas por el cliente, se procedió a realizar una serie de pruebas a fin de asegurarse que la configuración estuviese hecha correctamente y el sistema estuviese funcionando correctamente y así, de ser necesario, realizar las correcciones correspondientes.

CAPITULO IV

Desarrollo

En el siguiente capítulo se describe el proceso que se siguió para la realización del trabajo especial de grado, las fases en las que se dividió el mismo y todos los detalles correspondientes con la implementación del software y su puesta en marcha, así como las pruebas de funcionamiento realizadas al culminarlo. Se observará más adelante que las cinco fases planteadas en la metodología se redujeron a 4 fases al momento de proceder con la realización del trabajo.

4.1 Fase 1: Investigación

El trabajo se inició con una fase de investigación exhaustiva acerca de los temas que se tratarían y los protocolos que se utilizarían. La finalidad principal de esta fase fue obtener los conocimientos teóricos necesarios para poder proceder con las fases de implementación del proyecto; igualmente resulta importante al ser el capítulo inicial del trabajo ya que permite al lector el mismo beneficio de conocer la teoría necesaria para el desarrollo del mismo.

Se investigaron tópicos como el protocolo simple de gestión de red (SNMP), siendo este el más importante ya que se utilizó como protocolo de monitorización, en este entorno se realizaron descripciones de las versiones existentes del protocolo, los comandos que se utilizan comúnmente en cada una de ellas, las características importantes de seguridad y privacidad disponibles al utilizarlo, entre otros. Se describió de manera detallada la arquitectura del protocolo, deteniéndose en cada uno de los elementos que la componen, y la lógica de funcionamiento del mismo.

Adicionalmente se investigó acerca del protocolo de mensajes de control de internet (ICMP), ya que se utilizaría junto con SNMP para realizar las tareas de monitorización.

De manera complementaria se dedicó parte del capítulo a la descripción de las redes de información, los tipos de redes más comunes según su alcance, los elementos de una red y algunos de los estándares utilizados en ellas; se mencionan igualmente las redes privadas virtuales.

Finalmente, en esta fase se realizó la selección del software Zabbix, la razón por la cual se tomó la decisión de utilizarlo es que, además de ser totalmente gratuito y coincidir con los requerimientos del cliente, es de configuración sencilla y es capaz de obtener datos, manejar estadísticas y crear gráficos, sin necesidad de utilizar software adicional.

4.2 Fase 2: Instalación de software y ensayos

4.2.1 Instalación de máquina virtual

Se procedió a instalar el sistema operativo Ubuntu 10.10 en el software de virtualización Virtualbox, versión 4.1.14, esto se debió a que el personal encargado por parte de la empresa cliente (Servimeta) decidió tener el servicio de monitorización en un equipo virtual con un sistema operativo de software libre Linux.

Se eligió el sistema operativo Ubuntu 10.10 debido la experiencia personal con el mismo y además, dado que el software de monitorización trabaja sobre software libre Linux en sus distintas versiones.

El proceso de instalación se llevó de manera normal sin ningún error técnico.

Figura 6. Proceso de instalación de Ubuntu 10.10 sobre Virtualbox 4.1.14.
Fuente: Elaboración propia.

4.2.2 Instalación de Zabbix

Para realizar el proceso de instalación de Zabbix es necesario antes, tener en cuenta que se deben instalar y configurar una serie de paquetes a fin de satisfacer los requisitos mínimos necesarios para la instalación correcta del software de monitorización.

Estos requisitos de software se pueden observar en la siguiente tabla:

Software	**Versión**	**Comentarios**
Apache	1.3.12 o posterior	
PHP	5.0 o posterior	
Módulos PHP: php-gd	GD 2.0 o posterior	Debe soportar imágenes .png
Soporte PHP *TrueType*		Con -ttf
Soporte PHP bc		*php-bcmath, --enable-bcmath*
Soporte PH XML		php-xml o php5-dom
Soporte PHP *session*		*php-session*
Soporte PHP *socket*		*php-net-socket, --enable-sockets.*
Soporte PHP *multibyte*		*php-mbsgring, --enable-mbstring*
MySQL php-mysql	3.22 o posterior	Si se utilizará MySQL como base de datos para Zabbix.

Tabla 5. Requisitos de software para la instalación de Zabbix.
Fuente: www.zabbix.com.

Por otra parte, los requerimientos de hardware para un óptimo desempeño del software dependen de la cantidad de *hosts* a monitorear, ya que en este caso se trabajó con una red de menos de 500 hosts, se utilizó la siguiente configuración, que supera la recomendada por la documentación de Zabbix, proporcionada en la página web del mismo:

- Plataforma : Ubuntu Linux 32bits
- CPU/MEMORIA: Core2duo 2.33Ghz / 2Gb de RAM

4.2.2.1 Instalación de requisitos de software

Como se mencionó anteriormente, Zabbix requiere de la instalación previa de un software de base de datos y de php para su funcionamiento, para la instalación del mismo se utilizó el comando que se muestra a continuación:

- apt-get install apache2 php5 libapache2-mod-php5 php5-gd php5-mysql mysql-server libmysql++-dev libsnmp-dev libcurl4-openssl-dev libiksemel-dev openssh-server libssh2-1-dev build-essential fping

Se puede observar en la siguiente imagen el proceso de descarga que tuvo lugar al implementar el comando anterior.

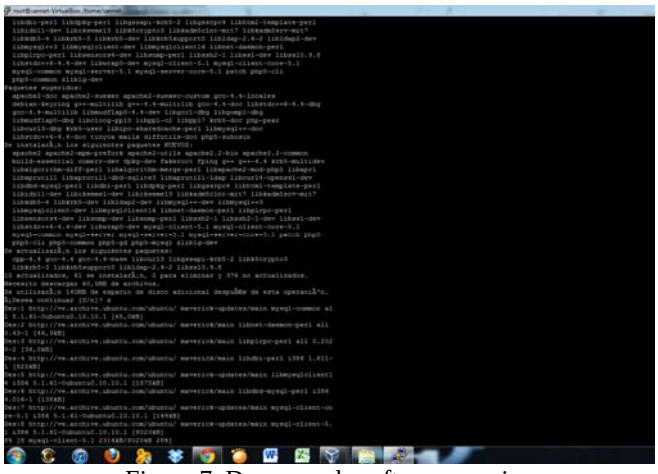


Figura 7. Descarga de software previo.
Fuente: Elaboración propia

En el proceso de configuración de paquetes se abrió la ventana de configuración e instalación de MySQL, la cual solicita una contraseña de usuario "*root*" de MySQL, para la administración de la base de datos.

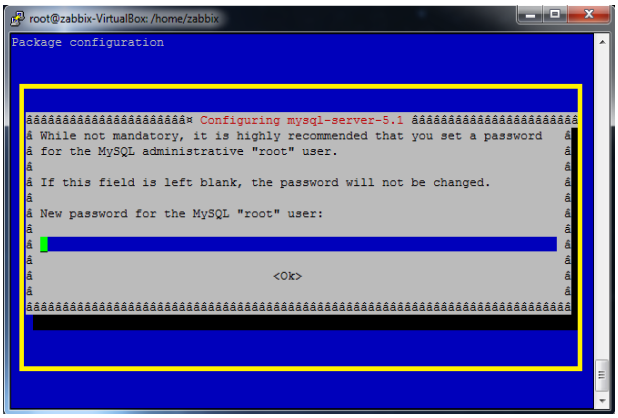


Figura 8. Configuración de contraseña del administrador de la base de datos.
Fuente: Elaboración propia

Una vez finalizado este proceso se contó con los elementos necesarios para el correcto funcionamiento del software de monitorización Zabbix.

4.2.2.2 Creación de usuario Zabbix

Para la gestión y configuración de los parámetros internos de la red se utilizaron contraseñas y nombres de usuario simples ya que sería utilizada para realizar ensayos en una red casera; y para la creación del nuevo gestor se creó un usuario homónimo al software, para lo cual se utilizó el siguiente comando:

- adduser --disabled-password --disabled-login zabbix

4.2.2.3 Descarga de fuente Zabbix

Anteriormente se habían descargado los paquetes necesarios para que el software funcionase correctamente, por lo cual el siguiente paso fue descargar el software de monitorización Zabbix, para la descarga se utilizó la página principal www.zabbix.com, en la cual se consiguieron todas las versiones existentes, así como los manuales de funcionamiento.

Para la descarga se utilizó el comando de consola de Linux **wget** seguido del link obtenido de la versión escogida en la página principal: http://sourceforge.net/projects/zabbix/files/ZABBIX%20Latest%20Stable/1.8.11/zabbix-1.8.13.tar.gz/download ?use_mirror=voxel

Al completar la descarga fue necesario descomprimir el archivo, para lo cual se utilizó el comando **tar xzf** seguido del nombre del archivo comprimido:

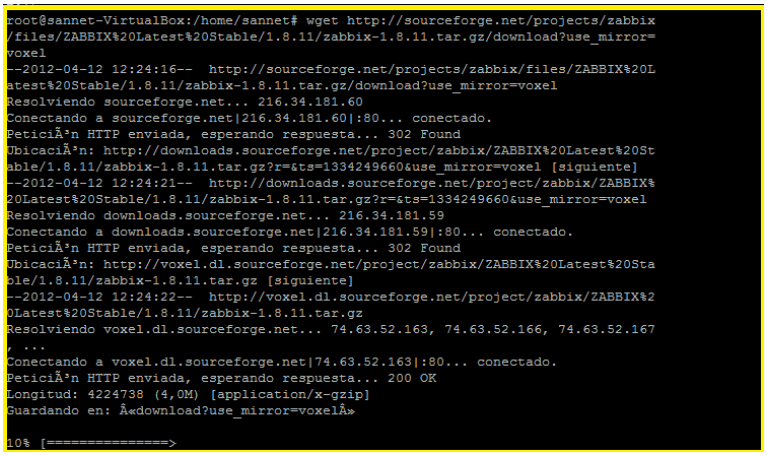


Figura 9. Instalación de la fuente de Zabbix.
Fuente: Elaboración propia

4.2.2.4 Configuración de base de datos

Se escogió utilizar como software de base de datos MySQL, debido a su popularidad para este tipo de aplicaciones, aunque este no es el único software de base de datos que se puede utilizar con Zabbix. Una vez descargados los paquetes correspondientes con el software de base de datos MySQL, resultó necesario configurar el mismo de manera que pudiese gestionar los usuarios configurados en Zabbix junto con los datos correspondientes a cada uno; para la configuración, la solución utilizada fue la creación de una nueva base de datos para el usuario nuevo, al llevar a cabo este proceso se otorgaron permisos al usuario de manera que los mismos se ajustasen a las labores que este debe cumplir dentro de la red.

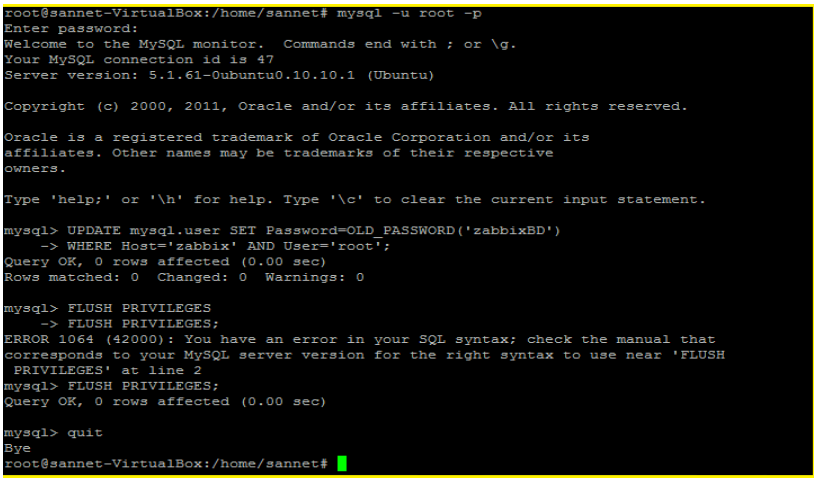


Figura 10. Creación de base de datos y otorgación de permisos.
Fuente: Elaboración propia.

4.2.2.5 Configuración de puertos

Debido a que Zabbix funciona como un servicio de red, se configuraron dos puertos para el envío de paquetes SNMP mediante el uso de los siguientes comandos:

- cd /etc/*services:*
- *Zabbix_agent* 10050/tcp
- *Zabbix_trap* 10051/tcp

4.2.2.6 Configuración de zabbix_*server*.conf

En la fase de configuración del servidor se modificó el archivo zabbix_*server*.conf, de manera que reconociese el usuario y contraseña correspondientes al administrador de la red, este archivo se ubica en la carpeta **/etc/zabbix/zabbix_*server*.conf**. Como es de esperarse, si no se hubiese realizado esta modificación, el servidor no tendría manera de acceder a la base de datos.

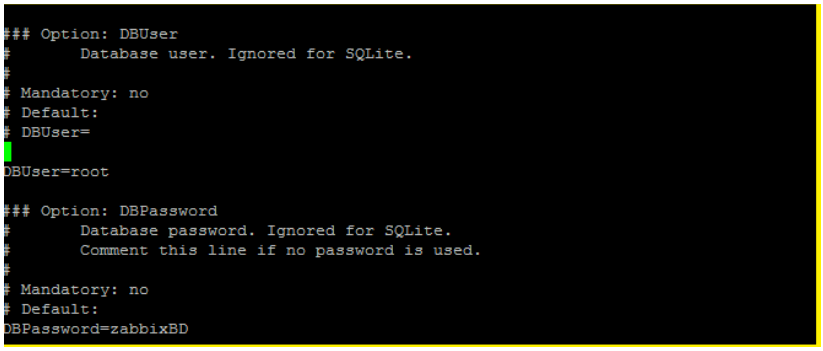


Figura 11. Configuración de zabbix_server.conf.
Fuente: Elaboración propia.

4.2.2.7 Configuración de *scripts* de inicio

Debido a que el servidor ni el agente levantan el servicio por defecto al prender el equipo, se modificaron las carpetas **/etc/init.d/zabbix-*server*** y **/etc/init.d/zabbix-*agent*,** en las cuales se agregaron los siguientes códigos:

- cd etc/init.d/zabbix-server
- PATH=/bin:/usr/bin:/usr/local/bin:/sbin:/usr/sbin:/usr/local/sbin
- DAEMON=/usr/local/sbin/${NAME}
- cd /etc/init.d/zabbix-agent
- PATH=/bin:/usr/bin:/usr/local/bin:/sbin:/usr/sbin:/usr/local/sbin
- DAEMON=/usr/local/sbin/${NAME}

Una vez hecho esto, se actualizó tanto el agente, como el servidor zabbix, el proceso se muestra en la siguiente imagen:

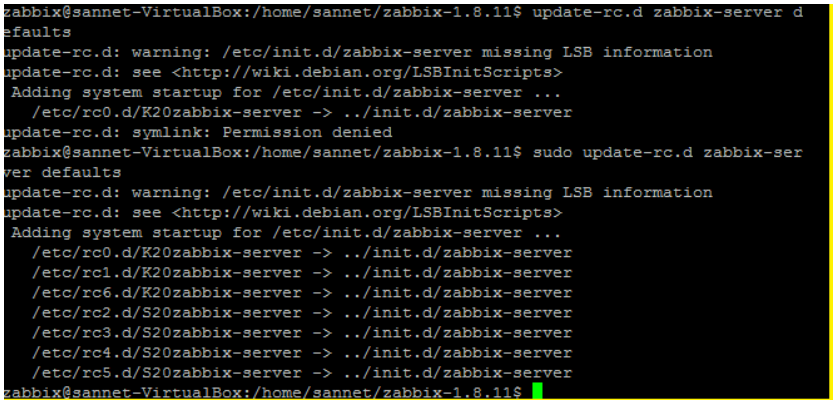


Figura 12. Actualización de zabbix-server.
Fuente: Elaboración propia.

4.2.2.8 Instalación de interfaz web

Zabbix trabaja por medio de una interfaz web para más comodidad al momento de hacer configuraciones desde cualquier sistema operativo en la red con acceso al servidor y con un explorador de internet.

Para esto se procedió a instalar un *front-end* basado en lenguaje PHP, el cual corre sobre el servidor apache con los siguientes comandos:

- cd /var/www
- rm *index*.html
- cp -R /*home*/zabbix/zabbix-1.8.2/*frontends*/php/*

Una vez completada la instalación se llevó a cabo la modificación de los parámetros PHP, esto se debió a que la configuración por defecto de PHP no es eficiente para el funcionamiento de Zabbix.

Se modificaron los siguientes parámetros:

- *max_execution_time:* tiempo máximo de ejecución.
- *max_input_time:* tiempo máximo de entrada.

- *post_max_size:* tamaño máximo de publicación.

- *date.timezone:* fecha, zona horaria.

```
;;;;;;;;;;;;;;;;;;;
; Resource Limits ;
;;;;;;;;;;;;;;;;;;;

; Maximum execution time of each script, in seconds
; http://php.net/max-execution-time
; Note: This directive is hardcoded to 0 for the CLI SAPI
max_execution_time = 300

; Maximum amount of time each script may spend parsing request
; idea to limit this time on productions servers in order to el
; long running scripts.
; Note: This directive is hardcoded to -1 for the CLI SAPI
; Default Value: -1 (Unlimited)
; Development Value: 60 (60 seconds)
; Production Value: 60 (60 seconds)
; http://php.net/max-input-time
max_input_time = 600
```

Figura 13. Configuración de PHP.
Fuente: Elaboración propia.

4.2.2.9 Comprobación y últimos pasos de instalación

Una vez realizados todos los pasos anteriores se introdujo la dirección IP del servidor Zabbix en el explorador web, la dirección por defecto de Zabbix es "dirección_ip/zabbix", esto fue modificado dentro del servidor apache para poder acceder de forma directa al colocar únicamente la dirección IP del servidor. Acto seguido se verificó que la configuración se hubiese realizado correctamente.

	Current value	Required	Recommended	
PHP version	5.3.3-1ubuntu9.10	5.0	5.3.0	Ok
PHP memory limit	256M	128M	256M	Ok
PHP post max size	32M	16M	32M	Ok
PHP upload max filesize	2M	2M	16M	Ok
PHP max execution time	300	300	600	Ok
PHP max input time	600	300	600	Ok
PHP timezone	America/Caracas			Ok
PHP databases support	MySQL			Ok
PHP BC math	yes			Ok
PHP MB string	yes			Ok
PHP Sockets	yes			Ok
PHP Session	yes			Ok
PHP GD	2.0	2.0	2.0.34	Ok
GD PNG Support	yes			Ok
libxml module	2.7.7	2.6.15	2.7.6	Ok
ctype module	yes			Ok

Ok

Figura 14. Comprobación PHP.
Fuente: Elaboración propia

Luego se verificó la existencia de la conexión con la base de datos, esto se muestra la siguiente imagen:

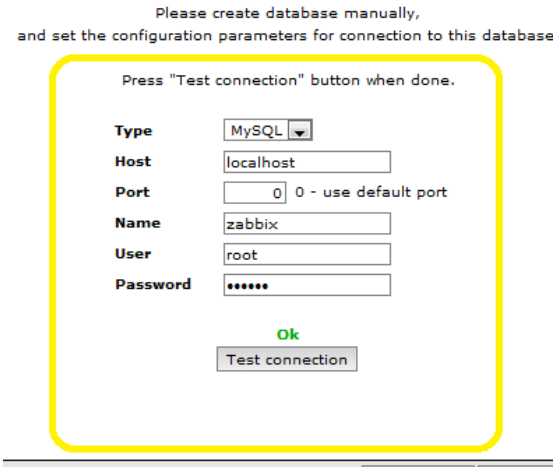


Figura 15. Verificación de conexión con MySQL.
Fuente: Elaboración propia.

Se completó el número de puerto sobre el cual trabaja el servidor Zabbix, como se muestra la siguiente imagen:

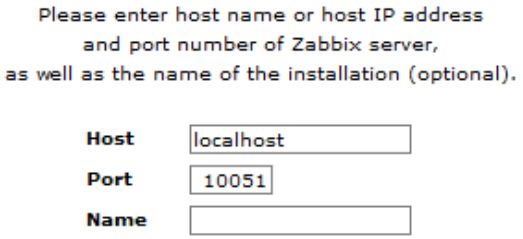


Figura 16. Configuración de puerto.
Fuente: Elaboración propia.

Por último se copió el archivo de configuración zabbix.conf.php a la carpeta conf ubicada en /var/www/conf, como se muestra en la siguiente imagen:

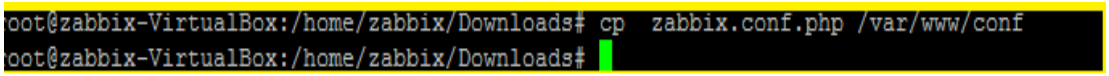


Figura 17. Copiar zabbix.conf.php a apache.
Fuente: Elaboración propia.

Los pasos finales de esta fase consistieron en la familiarización con el software, se realizaron pruebas con la ayuda de los manuales. Una vez se hubo comprendido la sintaxis de configuración y la lógica que utiliza el software se pudo continuar con el proyecto.

4.2.3 Pruebas y configuraciones locales

Para confirmar el funcionamiento del servidor, se llevaron a cabo una serie de pruebas, las cuales fueron realizadas en un entorno de red de área local con salida hacia internet de CANTV, las pruebas llevadas a cabo fueron las siguientes:

1. Confirmar si el agente Zabbix del mismo servidor estuviese en funcionamiento y recolectando los datos del sistema.

2. Configuración y confirmación de entregas de paquetes ICMP para comprobar conexión con los dispositivos en la red de prueba.

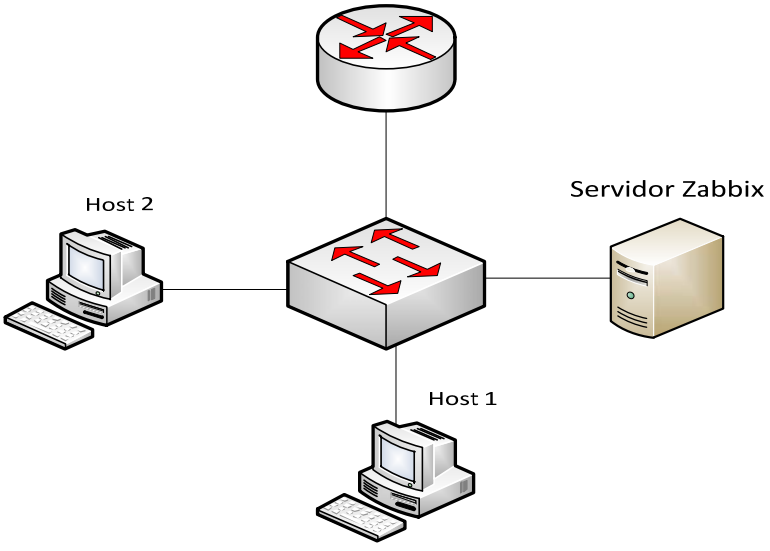


Figura 18. Topología de red de pruebas.
Fuente: Elaboración propia

4.2.3.1 Agente Zabbix

Para confirmar que el agente Zabbix funcionase correctamente, se verificó su estado dentro del servidor y se confirmó la obtención de valores. Al entrar en la página principal del servidor, específicamente en la pestaña *dashboard*, se puede apreciar el estado del sistema:

Status of Zabbix

Parameter	Value	Details
Zabbix server is running	Yes	localhost:10051
Number of hosts (monitored/not monitored/templates)	48	2 / 0 / 46
Number of items (monitored/disabled/not supported)	105	100 / 0 / 5
Number of triggers (enabled/disabled)[problem/unknown/ok]	4	3 / 1 [1 / 0 / 2]
Number of users (online)	2	1
Required server performance, new values per second	3.82	-

Updated: 21:03:54

Figura 19. Estado de Zabbix.
Fuente: Elaboración propia

Una vez confirmado el correcto funcionamiento del servidor, se procedió a la verificación de los datos obtenidos cada 30 segundos, se decidió que el tiempo de actualización por defecto resultaba conveniente; a continuación se pueden observar algunos de los ítems monitorizados por Zabbix.

Host	Description	Last check	Last value	Change	History
Zabbix server	- other - (98 Items)				
	Buffers memory	20 May 2012 21:13:52	146.8 MB	+156 KB	Graph
	Cached memory	20 May 2012 21:13:53	394.22 MB	+48 KB	Graph
	Checksum of /etc/passwd	20 May 2012 21:08:04	99755926	-	Graph
	Checksum of /etc/services	20 May 2012 21:08:05	3718822210	-	Graph
	Checksum of /usr/bin/ssh	20 May 2012 21:08:06	1447671618	-	Graph
	Checksum of /usr/sbin/sshd	20 May 2012 21:08:07	3118612449	-	Graph
	CPU idle time (avg1)	20 May 2012 21:14:10	87.66	+0.93	Graph

Figura 20. Valores de los ítems monitorizados en Zabbix servidor.
Fuente: Elaboración propia

4.2.3.2 Configuración ICMP en Zabbix

Para el envío de paquetes ICMP con Zabbix, fue necesario instalar en el sistema operativo Linux la herramienta **fping,** debido a que es de gran utilidad al momento de validar si un *host* está activo o no. Para que el servidor pueda usar la

herramienta en Zabbix se le debe indicar la ruta para llegar a la misma. Para obtener dicha ruta se puede utilizar el comando ***locate* fping,** una vez hecho esto se debe modificar en el archivo **zabbix_*server*.conf**, la línea ***FpingLocation*=**, el cambio que debe hacerse es el de la ruta predeterminada por la ruta obtenida al localizar **fping.**

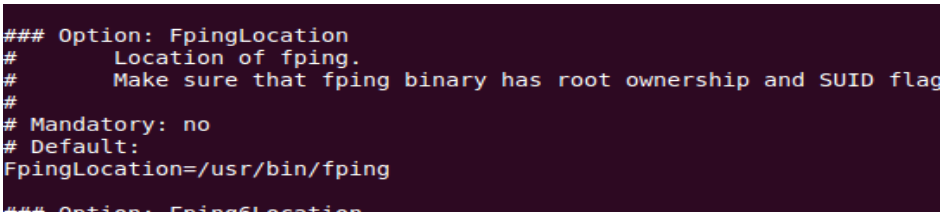


Figura 21. Configuración de fping dentro de Zabbix.
Fuente: elaboración propia.

Al finalizar la modificación del archivo de configuración, se realizó la asignación de permisos utilizando los siguientes comandos:

- chown root:zabbix /usr/sbin/fping
- chmod 4710 /usr/bin/fping

Al hacer esto se otorgaron a Zabbix los permisos necesarios para la utilización del paquete, para validar la asignación de permisos se introduce el comando **ls -l fping**, dentro de la carpeta **/usr/bin,** como se muestra en la siguiente imagen:

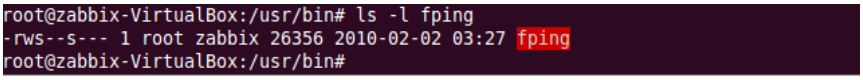


Figura 22. Validación fping.
Fuente: Elaboración Propia.

Donde:

- **r** significa permiso para leer
- **w** significa permiso para escribir

Finalizada la configuración de la herramienta dentro de Zabbix se procedió a la configuración del ítem para la comprobación de conexión con un *host* de la red en la interfaz virtual.

4.2.3.3 Configuración de ítems ICMP.

Para comprobar la conexión entre servidor-*host*, en primer lugar se debe agregar el *host* para ser monitorizado por Zabbix, esto se configuró haciendo clic en la pestaña *configuration*, luego se hizo clic en host de algunos de los grupos predeterminados y en la siguiente ventana se seleccionó *create host* para su configuración.

Dentro de la ventana de configuración de *host* se debe configurar el puerto de escucha, en este caso el 161, que es el puerto que usa el protocolo SNMP, y en la dirección de IP se colocó la IP del *host* a monitorizar.

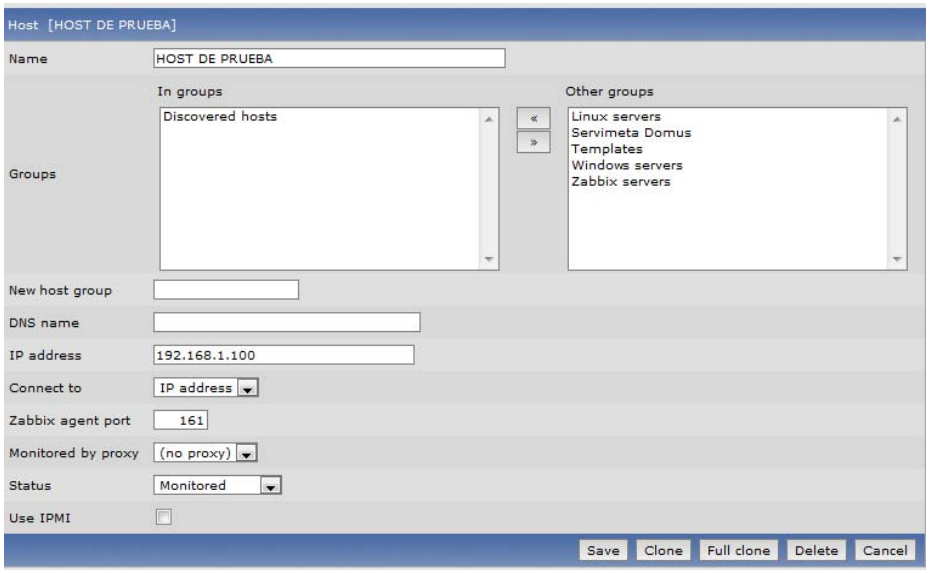


Figura 23. Configuración de host.
Fuente: Elaboración propia.

Al finalizar la creación del *host*, se realizó la configuración de un ítem para comprobar la conexión entre el servidor y un *host*. Esto se configuró en la ventana "ítem", dentro del menú *host*, que a su vez se encuentra dentro de la ventana *configuration*. En la siguiente imagen se muestra la configuración de un ítem.

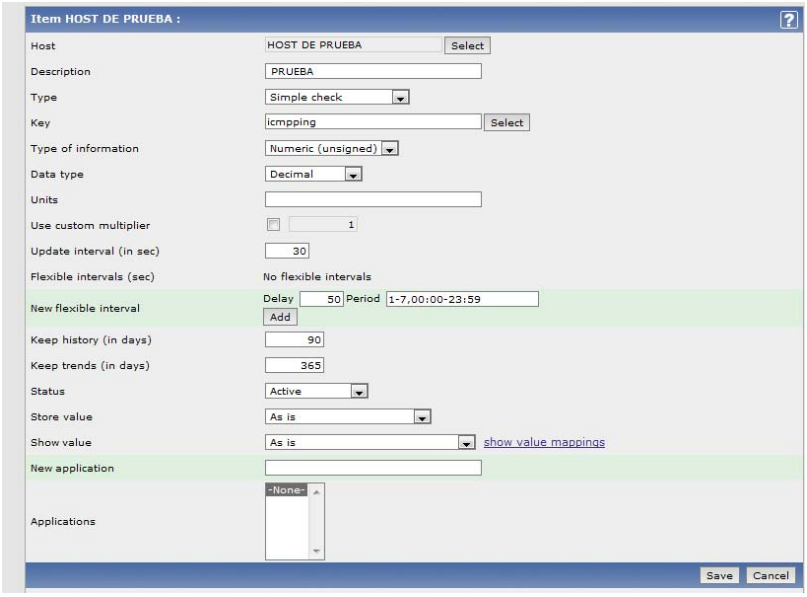


Figura 24. Configuración de ítem ICMP.
Fuente: Elaboración propia.

Una vez configurados los ítems, los resultados obtenidos se pudieron observar en la ventana *latest data* ubicada en *monitoring*, en el caso de pruebas de conexión utilizando protocolo ICMP, los valores obtenidos fueron: 1 (uno) para indicar que la conexión esta activa o 0 (cero) para indicar que la conexión esta inactiva.

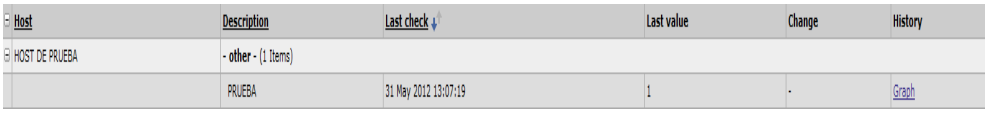


Figura 25. Prueba ICMP a host de la red.
Fuente: Elaboración propia

4.3 Fase 3: Implementación del software en la red del cliente y configuración de parámetros a monitorizar

4.3.1 Instalación del servidor

A fin de no realizar una vez mas todo el proceso de instalación descrito anteriormente, debido a que requiere de mucho tiempo, y se contaban con espacios de tiempo limitados en las visitas realizadas a la empresa, se instaló una copia de la máquina virtual utilizada en la fase anterior en la red del cliente. Al realizar la

instalación de la copia fue necesario cambiar los nombres de usuario y contraseñas de acceso configurados anteriormente, de manera que fuese el administrador de la red el que los escogiese esta vez; sin embargo, *grosso modo*, no hubo grandes cambios, debido a que la configuración utilizada inicialmente se realizó en miras de satisfacer las necesidades del cliente de Sannet Soluciones C.A.

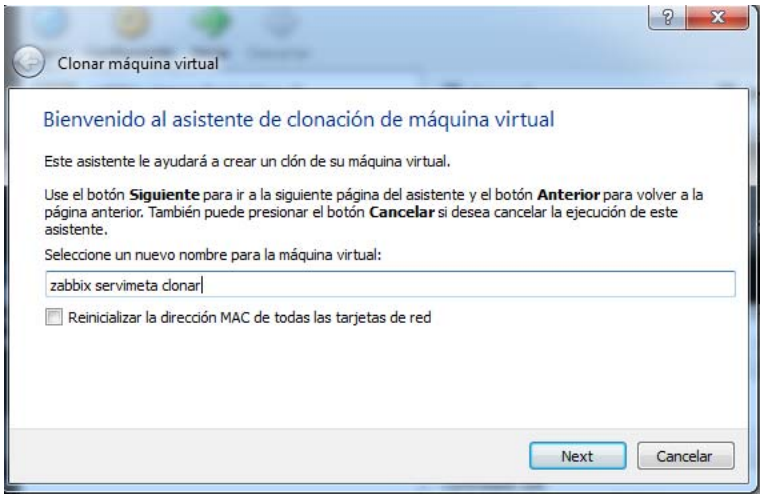


Figura 26. Clonado de máquina virtual.
Fuente: Elaboración propia

4.3.2 Configuración de parámetros

Una vez instalada la máquina virtual y realizadas las modificaciones pertinentes se llevó a cabo una reunión con el administrador de la red, a fin de determinar los parámetros y equipos a monitorizar, el resultado de dicha reunión fue la siguiente lista de requerimientos:

Sede Domus:

Router:

- Cisco Serie 1800: monitorizar CPU, memoria, tráfico y Ping constante

Sede Menegrande:

Routers:

- Cisco Serie 2800: monitorear CPU, memoria, tráfico y Ping constante.
- Cisco Serie 2800: monitorizar CPU, memoria, tráfico y Ping constante

Sede Propatria:

Router:

- Cisco Serie 1800: monitorizar CPU, memoria, tráfico y Ping constante.

Sede Urdaneta:

Router:

- Cisco Serie1800: monitorizar CPU, memoria y ping constante.

Sede El Recreo:

Router:

- Cisco Serie 1800: monitorizar CPU, memoria y ping constante.

Sede Dorado:

Router:

- Cisco Serie 800: monitorear CPU, memoria y ping constante.

Servidor:

- Servidor Dell: Ping constante

Sede Samat:

Router:

- Cisco Serie 800: monitorizar CPU, memoria, tráfico y Ping constante

Servidor:

- Servidor Dell: Ping constante.

Junto con la lista de requerimientos se recibió el diagrama de la red de la empresa, en el cual se encontraban reflejadas todas las direcciones IP necesarias, así como las interfaces y los túneles virtuales, así mismo, en el diagrama se reflejan los tipos de conexiones utilizadas en cada caso. Una modificación de dicho diagrama, adaptada a las políticas de privacidad de la empresa, se presenta en la siguiente figura:

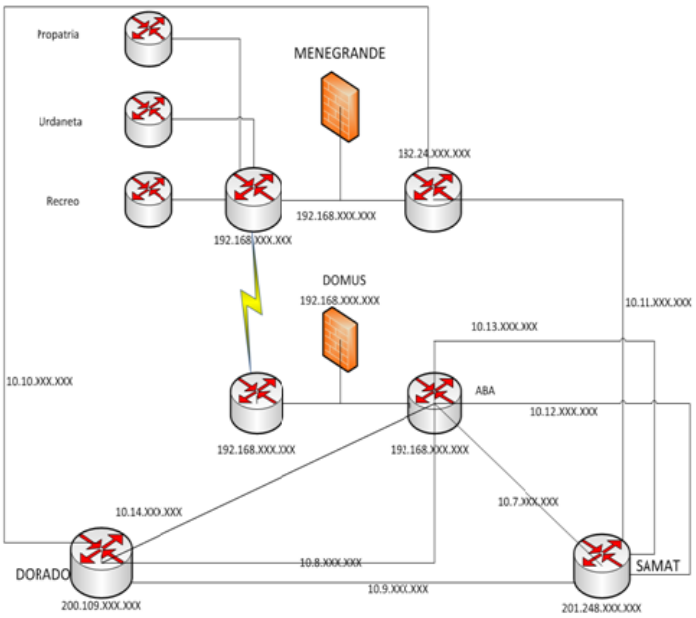


Figura 27. . Diagrama de red Servimeta C.A.
Fuente: Servimeta C.A

4.3.3 Configuración de comunidades

Dentro del archivo de configuración **snmp.conf**, ubicado en la carpeta **/etc/snmp**, se utilizó el siguiente *script* para la configuración de las comunidades.

- com2sec local 127.0.0.1/32 comunidad
- com2sec miredlocal 192.168.0.0/24 comunidad

- group MyRWGroup v1 local
- group MyRWGroup v2c local
- group MyRWGroup usm local
- #Se asigna ACL al grupo de solo lectura
- group MyROGroup v1 miredlocal
- group MyROGroup v2c miredlocal
- group MyROGroup usm miredlocal
- # Ramas MIB que se permiten ver
- # name incl/excl subtree mask(optional)
- view all included .1 80
- #Permisos
- # group context sec.model sec.level prefix read write notife
- access MyRWGroup any noauth exact all all all
- access MyROGroup any noauth exact all none all

4.3.4 Obtención de OIDs

Para poder configurar todos los parámetros a monitorizar con Zabbix fue necesario conocer el OID correspondiente en cada caso, así como la comunidad a la que pertenece cada uno de los enrutadores, a fin de que la comunicación entre el servidor de monitorización y los elementos de la red funcionase de manera correcta.

Por razones de seguridad y privacidad, las comunidades fueron configuradas por el administrador de la red.

Una vez realizada la configuración de comunidades se procedió a la obtención de los OIDs de cada enrutador, esto se logró mediante el uso de la herramienta *snmpwalk* en la consola de comandos de Linux, este comando devuelve una lista que contiene todos los OIDs dentro del rango indicado. A continuación se muestra una captura del proceso de obtención de los OIDs para uno de los *routers* del cliente.

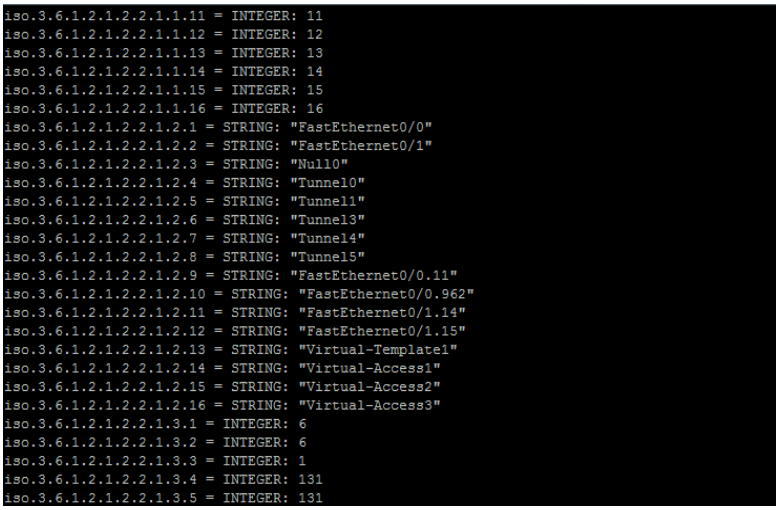


Figura 28. Obtención de los OIDs del *Router* principal.
Fuente: Elaboración propia

El proceso de obtención de OIDs se llevó a cabo todos los enrutadores que se monitorizarían, ya que cada uno cuenta con identificadores de objeto distintos. Para la monitorización de estado de los servidores no fue necesario conocer los identificadores de objetos ya que se utilizó un protocolo distinto (ICMP).

4.3.5 Configuración de monitorización del tráfico

Para monitorizar el tráfico de entrada y salida en los elementos de la red, se crearon grupos, ya que cada elemento que se desee monitorizar debe ser registrado como un *host* dentro de un grupo en el software Zabbix, los grupos se crearon en correspondencia con la ubicación geográfica de los equipos. Posterior a la creación de

los grupos se crearon *hosts*, al igual que en la fase de ensayos. Seguidamente se crearon los ítems en cada uno de los casos, se considera un ítem cada parámetro a monitorizar en un equipo.

En el caso de la lectura del tráfico de entrada y salida en los enlaces de la red se utilizó como apoyo la información proporcionada por Cisco Systems, Inc en su página web respecto al OID utilizado para realizar esta lectura en sus equipos de enrutamiento.

Specific Object Information	
Object	ifInOctets
OID	1.3.6.1.2.1.2.2.1.10
Type	Counter32
Permission	read-only
Status	current
MIB	IF-MIB ; - View Supporting Images
Description	"The total number of octets received on the interface, including framing characters. Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of ifCounterDiscontinuityTime."

Figura 29. Información OID de tráfico de entrada en *Routers* de Cisco System.
Fuente:http://tools.cisco.com/Support/SNMP/do/BrowseOID.do?objectInput=1.3.6.1.2.1.2.2.1.10&translate=Translate&submitValue=SUBMIT&submitClicked=true

En la ventana de creación de ítem se rellenaron los siguientes campos:

- *Description:* descripción del ítem.
- *Type:* versión SNMP.
- *SNMP OID:* OID correspondiente
- *SNMP community:* comunidad a la que pertenece el dispositivo, se debe tomar en cuenta que el servidor debe pertenecer a la misma comunidad para poder obtener información del equipo.
- *Type of information:* tipo de dato, en este caso *Numeric*(*float*).
- *Units:* unidad de medida de datos, en este caso b/s (bits por segundo).

- *Use custom multiplier:* multiplicador, en caso de desearlo. En este caso se colocó el número 8, debido a que se deseaba obtener las medidas en Bytes por segundo.
- *Store value: Delta(speed per second),* ya que el dato que se obtiene es una velocidad.

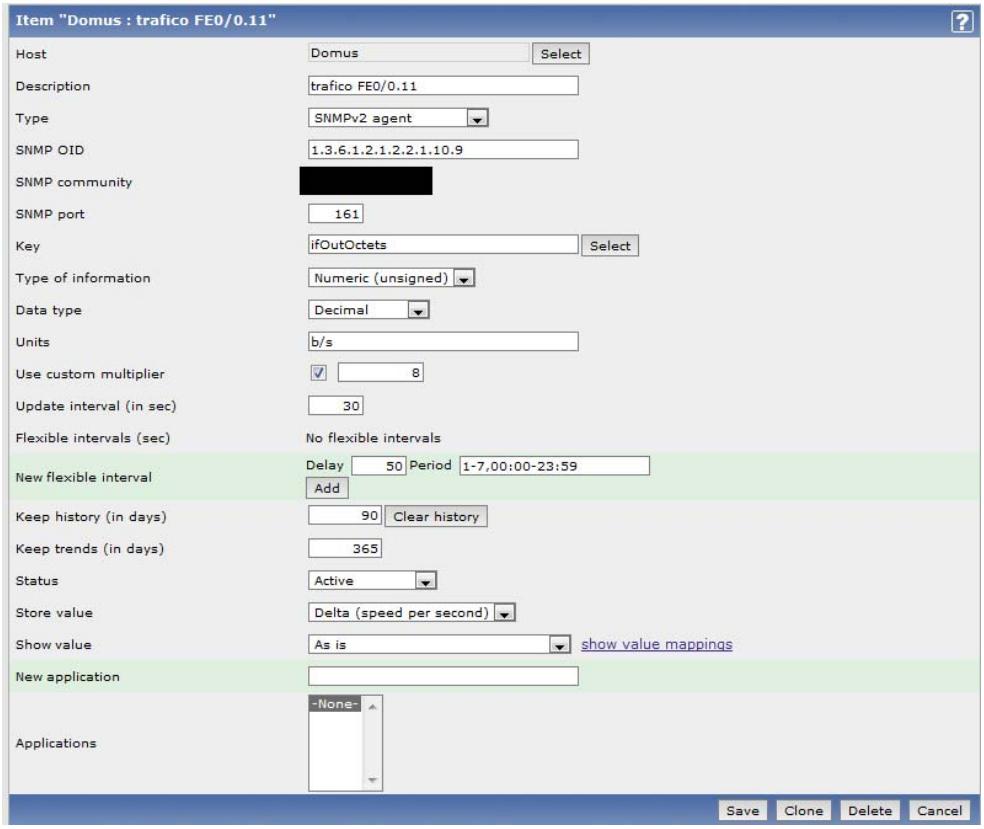


Figura 30. Configuración de ítem de tráfico.
Fuente: Elaboración propia.

Este proceso se repitió para cada una de las interfaces de los enrutadores.

4.3.6 Monitorización de la CPU

Para la monitorización del uso de CPU de los enrutadores se realizó el mismo proceso de creación de ítems, muy similar al descrito anteriormente, en este caso resulta un poco más sencillo, ya que se modifican menos parámetros que en el caso del tráfico, los campos modificados fueron los siguientes:

- *Description:* descripción del ítem.
- *Type:* se escogió como versión SNMP la versión 2.
- *SNMP OID:* Se colocó la cadena OID correspondiente
- *Units:* se colocó el símbolo %, dado el valor obtenido es un porcentaje.

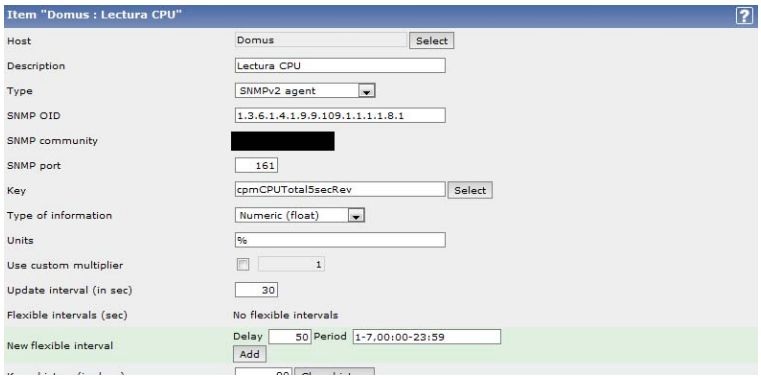


Figura 31. Configuración de ítem CPU.
Fuente: Elaboración propia.

Version	Cisco IOS Software releases 12.2(3.5) or later	Cisco IOS Software releases later to 12.0(3)T and prior to 12.2(3.5)	Cisco IOS Software releases prior to 12.0(3)T
MIB	CISCO-PROCESS-MIB	CISCO-PROCESS-MIB	OLD-CISCO-CPU-MIB
Objects	cpmCPUTotal5minRev (.1.3.6.1.4.1.9.9.109.1.1.1.1.8)	cpmCPUTotal5min (.1.3.6.1.4.1.9.9.109.1.1.1.1.5)	avgBusy5 (.1.3.6.1.4.1.9.2.1.58)
	cpmCPUTotal1minRev (.1.3.6.1.4.1.9.9.109.1.1.1.1.7)	cpmCPUTotal1min (.1.3.6.1.4.1.9.9.109.1.1.1.1.4)	avgBusy1 (.1.3.6.1.4.1.9.2.1.57)
	cpmCPUTotal5secRev (.1.3.6.1.4.1.9.9.109.1.1.1.1.6)	cpmCPUTotal5sec (.1.3.6.1.4.1.9.9.109.1.1.1.1.3)	busyPer (.1.3.6.1.4.1.9.2.1.56)

Figura 32. OIDs para monitorización de CPU en *Routers* Cisco.
Fuente: Elaboración propia.

El mismo proceso fue llevado a cabo con cada uno de los enrutadores monitorizados en la red.

4.3.7 Configuración de ítems ICMP

Este proceso fue muy simple dado que se llevó a cabo siguiendo los mismos pasos que en los ensayos realizados en la fase 2. Se crearon ítems de envío de paquetes ICMP y se configuraron de la manera descrita anteriormente. Este proceso

se realizó una vez por cada interfaz incluida en el diagrama facilitado por el administrador de la red. Una vez configurados todos los ítems se realizó a la validación de datos, y acto seguido a la configuración de alarmas.

La razón para utilizar protocolo ICMP en este caso en lugar de SNMP como para el resto de los parámetros es que fue solicitado explícitamente por el administrador de la red que se hiciese de esta manera.

4.3.8 Configuración de alarmas

En una segunda reunión con el administrador de la red, se determinaron los ítems que contarían con una alarma, así como el tipo de alarma que se utilizaría (correo electrónico). La decisión final fue que las alertas vía correo electrónico solo ocurrirían en caso de falla en los mensajes ICMP, y que las recibirían: el administrador de la red de Servimeta C.A y los encargados por parte de Sannet Soluciones C.A de prestarle soporte técnico.

Una vez determinados los casos de alerta se procedió a la configuración de las alarmas. Para realizar la configuración se debe ingresar a la ventana *triggers* dentro de la configuración de *host*, una vez allí, se procede a modificar los parámetros necesarios como se muestra a continuación:

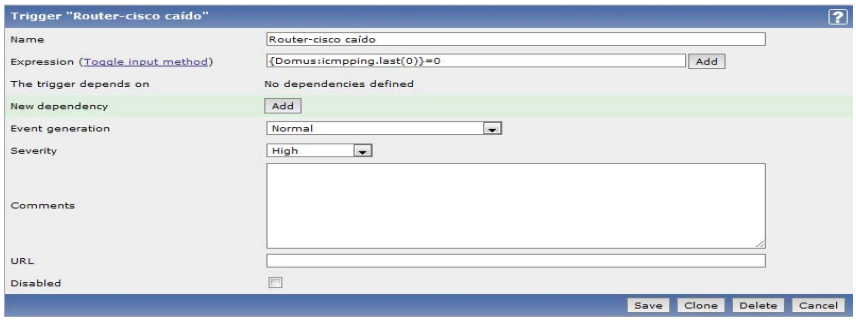


Figura 33. Configuración de alarma.
Fuente: Elaboración propia.

El campo *expression* se completa de manera automática al rellenar los campos de la ventana que se genera al presionar el botón *add*. Estos campos se pueden observar en la siguiente imagen:

Figura 34. Generación de expresión de alarma ICMP.
Fuente: Elaboración propia.

Se observa que el valor insertado es 0, debido a la lógica utilizada para la obtención de respuestas de los mensajes ICMP.

4.3.8.1 Configuración de envío de alarmas

Para realizar el envío de alarmas, se debe realizar la configuración correspondiente al protocolo SMTP, ya que es mediante este que Zabbix realiza el envío de correos electrónicos. Para esto simplemente se accedió a la opción *media types*, donde aparece una ventana de configuración como la que se observa en la imagen. En los campos censurados se ingresó la información correspondiente con el servidor SMTP y la dirección de correo desde la cual serían enviadas las alertas.

Figura 35. Configuración de e-mail.
Fuente: Elaboración propia.

Una vez configurado el servidor SMTP, para envío de correos electrónicos, se otorgaron los permisos correspondientes y se agregaron las direcciones de correo los destinatarios; esto se realizó ingresando en la ventana *users* ubicada en *administration*, en el menú principal.

Para que el envío del correo electrónico ocurriese automáticamente al ocurrir un error se configuraron acciones. Para hacer esta configuración se ingresó en el menú *actions* dentro de *configuration;* allí se procedió a configurar el tipo de acción a tomar, en este caso: enviar un correo electrónico, por ende se seleccionó *e-mail* en la opción *send only to,* y se configuró finalmente el tipo de alarma a ser enviada, así como las características de la falla.

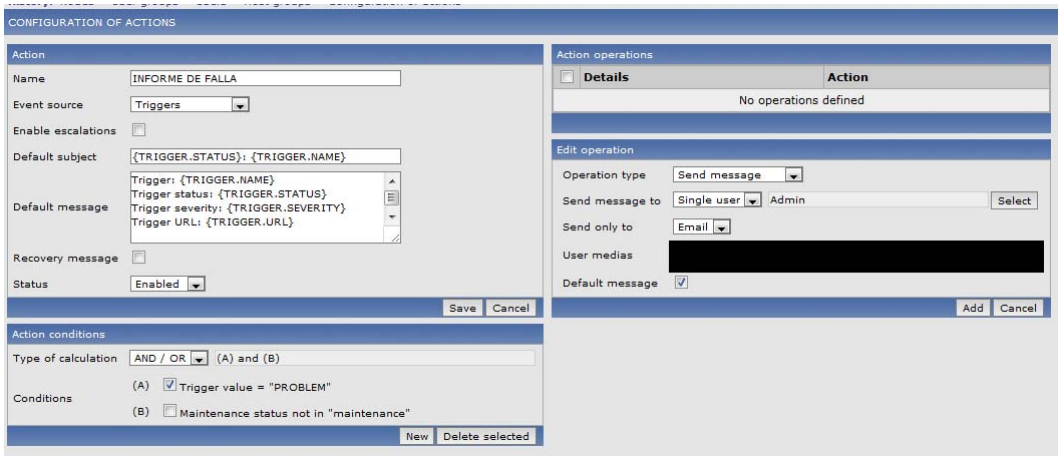


Figura 36. Configuración de una acción.
Fuente: Elaboración propia.

4.3.9 Creación de gráficos

El paso final en el proceso de implementación consistió en la creación de gráficos de comparación entre el tráfico de entrada y salida de cada una de las interfaces monitoreadas.

Para la creación de dichos gráficos se siguió la ruta *configuration-hosts-graphs*; al estar dentro del área de configuración de gráficos se hizo clic en el botón *create graph*, y en la ventana que aparece a continuación se añadieron los ítems que formarían parte del gráfico, en este caso estos fueron tráfico de entrada y tráfico de salida. También se seleccionó el tipo de línea a utilizar en los gráficos así como los colores para cada uno de los ítems.

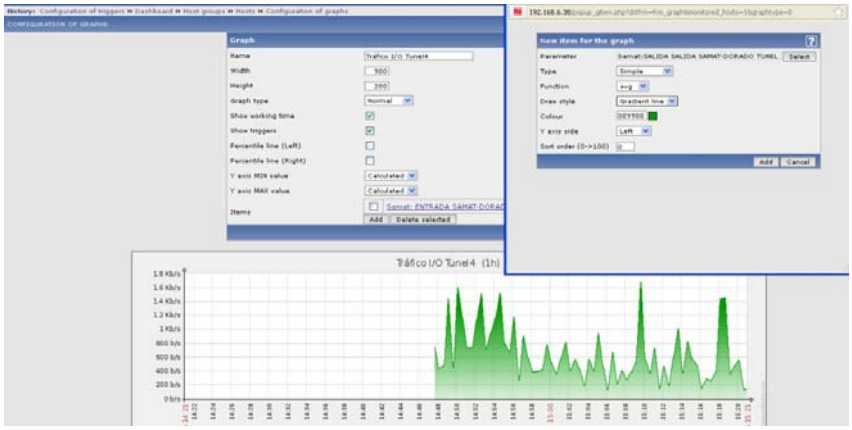


Figura 37. Creación de gráfico de tráfico.
Fuente: Elaboración propia.

Este proceso se repitió para cada una de las interfaces en las cuales se configuró monitorización de tráfico. También se creó un gráfico de uso de CPU por cada enrutador, se siguió el mismo proceso.

4.3.9.1 Creación de *screens*

Se crearon pantallas, en cada una de las cuales estaban contenidos todos los gráficos correspondientes a cada uno de los grupos. Para esto se ingresó en el menú *configuration-screens-create screen*, allí se agregó el nombre de las pantallas así como el número de columnas y filas correspondientes con el numero de gráficos a insertar y la distribución deseada de los mismos.

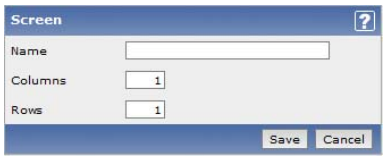


Figura 38. Creación de screens.
Fuente: Elaboración propia.

A continuación se hizo clic en *save* para guardar esta configuración, una vez creadas las pantallas para cada grupo se accedió a cada una de ellas para agregar los gráficos que se deseaban observar.

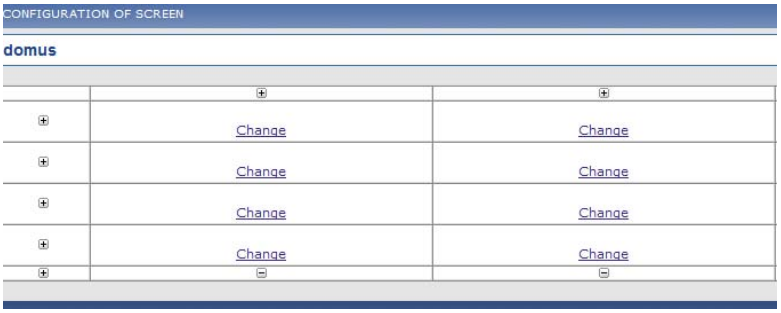


Figura 39. Adición de gráficos en las pantallas.
Fuente: Elaboración propia.

Para añadir los gráficos se hizo clic en *change* y se seleccionó en gráfico que se deseaba colocar en cada una de las divisiones de la matriz.

4.3.9.2 Creación de *Slideshows*

Una vez configurados todos los gráficos se crearon presentaciones a fin de que hubiese una rotación automática de los mismos.

Para la creación de estas se accedió a la ruta *configuration-slides-create slide show* y allí se añadieron las pantallas que formarían parte las presentaciones de una en una haciendo clic en el botón *add.*

Figura 40. Creación de slideshows.
Fuente: Elaboración propia.

4.4 Fase 4: Pruebas de funcionamiento

Las pruebas de funcionamiento del sistema de gestión se llevaron a cabo bajo la supervisión del administrador de la red de Servimeta, y en conjunto con parte del equipo de apoyo de Sannet Soluciones, esto debido a que algunas de las pruebas

consistieron en validar los datos recolectados por Zabbix, para luego ser comparados con los valores mostrados por el sistema operativo de los enrutadores.

4.4.1 Validación de datos

Para validar los datos, el analista de sistemas contratado por Sannet Soluciones se encargó de la recolección de datos directamente en los enrutadores, estos se compararon con los datos recolectados por el software de gestión Zabbix.

Los ítems se configuraron de manera que los agentes envíen información cada 30 segundos, este parámetro se dejo por defecto debido a que es el tiempo de actualización recomendado por los creadores del software para evitar congestión en la red. La comparación de los datos obtenidos desde los enrutadores y los obtenidos desde el software de gestión se realizó más de una vez para cada uno de los ítems configurados, a fin de certificar que la información fuese correcta.

Por otro lado, para una nueva confirmación, se hicieron pruebas de validación usando el comando ***snmpget*** en Linux para obtener los valores en tiempo real de los enrutadores como se muestra en la siguiente figura para la prueba de uso de CPU de un enrutador:

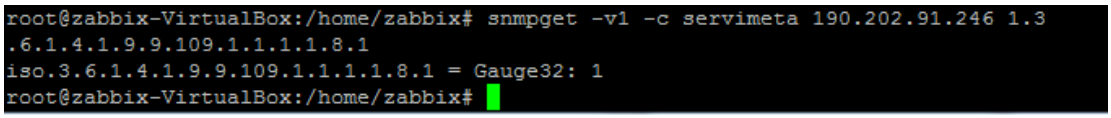


Figura 41. Prueba con el comando snmpget.
Fuente: Elaboración propia

4.4.2 Envío de alarmas

Una vez conformes con la validación de los datos obtenidos, se procedió a la confirmación de envió de alarmas, para comprobar esto se interrumpieron conexiones de manera controlada, a fin de comprobar que las alarmas locales respondiesen adecuadamente, incluidos los casos en los que se debía enviar un correo electrónico.

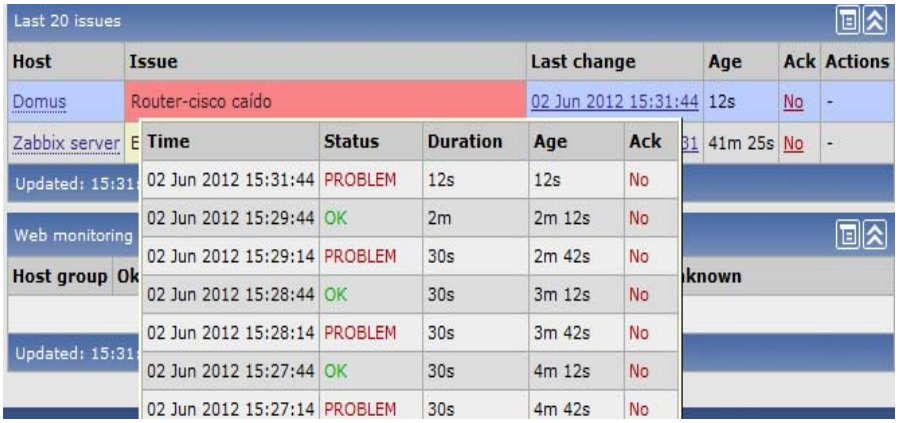


Figura 42. Activación de una alarma.
Fuente: Elaboración propia.

Al realizar la desconexión del enlace se observó que las alarmas correctas apareciesen en el registro de eventos del sistema. Acto seguido, se confirmó que la recepción de correos electrónicos ocurriese acorde a la configuración.

CAPÍTULO V
Resultados

Con el fin de mostrar la relación entre las labores llevadas a cabo a lo largo de la realización de este trabajo especial de grado y los objetivos planteados al principio del mismo, se hizo una recolección de resultados basados en cada una de las fases descritas en el marco metodológico. Dicha relación se presenta el este capítulo.

Ya que gran parte de la realización del trabajo consistió en la implementación del sistema de monitorización presentado en el capítulo cuatro, a continuación se presentarán los resultados, deseados o no, y en el caso de los últimos, el método utilizado para la resolución de problemas.

5.1 Instalación inicial

Utilizando como apoyo los manuales de usuario, disponibles de manera gratuita en la página web del software de gestión Zabbix, se logró la exitosa instalación, además se logró manejar un nivel de dominio del software suficientemente profundo para brindar un servicio de monitorización confiable. Parte importante del uso del soporte brindado por los creadores del software, fue la utilización de sus tablas de recomendaciones como guías al momento de determinar la mejor combinación de características de hardware que permitiese obtener los mejores resultados a partir de Zabbix, tomando en cuenta factores como el número de *hosts* en la red, y las capacidades de procesamiento de los equipos, entre otros.

Utilizando estos mismos manuales, junto con un poco de práctica, se aprendió la sintaxis y la lógica a ser utilizada para la correcta configuración de las características que se deseaba monitorizar, de manera que se obtuviesen resultados fiables, ya que esto no depende únicamente de que la instalación sea adecuada, sino de una combinación óptima entre recursos, instalación y configuración que se adapte a las necesidades de la red.

En la fase de instalación se presentaron problemas debido a que algunas de las contraseñas del software de base de datos, MySQL, no se configuraron correctamente en el primer momento. Dichos problemas se presentaron como errores al momento de intentar acceder al servidor y, en otros casos, cuando fue posible ingresar al sistema, al intentar obtener información a través del mismo, ya que la conexión entre el servidor Zabbix y la base de datos no funcionaba correctamente.

El método utilizado para ubicar el problema fue llevar a cabo un proceso de recapitulación de las acciones tomadas hasta el momento, lo que tuvo como reacción verificar que todas las configuraciones fuesen correctas y de este modo se observó que se había perdido congruencia entre las contraseñas, teniendo esto como consecuencia el error mencionado anteriormente.

La solución fue simplemente un cambio de contraseñas, lo cual se puede observar en la siguiente figura.

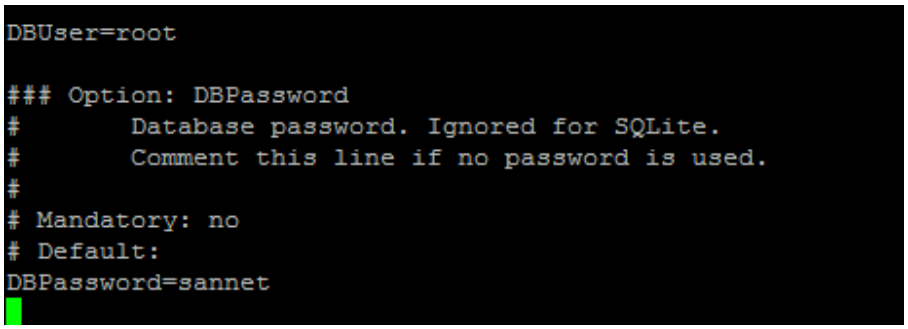


Figura 43. Resolución de error en la base de datos.
Fuente: Elaboración propia

5.2 Configuración de parámetros básicos

Una vez se logró la instalación de manera satisfactoria, se procedió a la configuración de una serie de parámetros básicos locales que se utilizarían para la realización de pruebas de funcionamiento iniciales. Para esto se configuró un agente local, que tuvo como objetivo emular un equipo conectado a la red del servidor. En este agente se configuró el envío de paquetes ICMP y la obtención de valores

correspondientes con el *performance* del hardware como uso de CPU y de memoria RAM.

5.3 Pruebas locales de funcionamiento

Al contar con el software en estado funcional y sin problemas aparentes se decidió seguir adelante con las pruebas de configuración. Como se mencionó anteriormente, se llevó a cabo un proceso de configuración de parámetros básicos a fin de comprender la forma correcta de ingresar los parámetros que componen cada ítem en el software y así obtener los resultados deseados.

El siguiente paso fue realizar pruebas básicas de funcionamiento con los parámetros configurados, esto con la intención de tener dominio suficiente de la lógica bajo la cual funciona el sistema y así lograr satisfacer los requerimientos del cliente al pasar a la fase de implementación en la red del mismo.

5.3.1 Pruebas del agente

Como parte de las pruebas realizadas con el software, previo a su instalación en la red del cliente, se logró la extracción de 103 datos, entre los cuales se pueden contar la carga de memoria RAM, la carga del Disco Duro e información referente al sistema operativo del servidor. La capacidad de obtener este tipo de datos se considera realmente importante, ya que el estado de los recursos de hardware será un factor determinante al momento de definir si se requiere una actualización a nivel físico en la red o no.

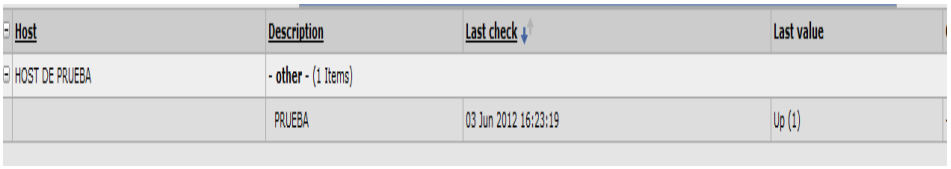
En último lugar, pero no por eso con menor importancia, se logró comprobar que es sistema funcionaba correctamente, para esto se extrajeron datos del *localhost*, estos fueron validados a través de una interfaz virtual del servidor sin mayores inconvenientes. Es importante destacar que el agente utilizado por el servidor es un agente Zabbix, el cual se configuró en la carpeta /etc/zabbix/zabbix_agentd.conf.

```
Server=127.0.0.1

### Option: Hostname
#       Unique, case sensitive hostname.
#       Required for active checks and must match hostname as configured on the server.
#       Value is acquired from HostnameItem if undefined.
#
# Mandatory: no
# Default:
# Hostname=

Hostname=Zabbix server
```

Figura 44. Configuración de agente Zabbix.
Fuente: Elaboración propia

5.3.2 Pruebas con paquetes ICMP

Al estar configurados los ítems necesarios para realizar pruebas con el protocolo de control ICMP, a fin de detectar el estado, activo o inactivo, de los *hosts* de la red de prueba, se accedió a la sección *latest data*, en la cual se encuentra toda la información reciente con la que cuenta el sistema, y allí se observaron los resultados correspondientes con las respuestas a los mensajes ICMP enviados por el sistema.



Host	Description	Last check	Last value
HOST DE PRUEBA	- other - (1 Items)		
	PRUEBA	03 Jun 2012 16:23:19	Up (1)

Figura 45. Host de prueba en estado activo.
Fuente: Elaboración propia

Como se puede observar en la figura anterior, los resultados obtenidos en este caso fueron correspondientes con los deseados, ya que el host de prueba se encontraba activo; posteriormente, a fin de obtener comprobación total del funcionamiento, se desactivó el *host*, lo cual arrojó un resultado de *host* inactivo, tal como era deseado.

HOST DE PRUEBA	- other - (1 Items)				
	PRUEBA	03 Jun 2012 19:03:37	Down (0)	-	Graph

Figura 46. Host de prueba en estado inactivo.
Fuente: Elaboración propia.

5.4 Instalación en la red del cliente

La razón por la cual se tomó la decisión de utilizar el software previamente instalado en lugar de realizar una nueva instalación en el equipo del administrador de la red, fue principalmente el tiempo que requería la instalación completa; y esta decisión fue posible debido a que, desde un principio se definió que el servidor funcionaria sobre una máquina virtual.

En lo referente a la instalación en la red del cliente fue un proceso realmente simple, ya que consistió en utilizar la máquina virtual que se había instalado previamente para realizar los ensayos, y hacerle pequeñas modificaciones en cuanto a las contraseñas y nombres de usuario; igualmente se requirió de la modificación de las comunidades a las que pertenecían los equipos para lograr compatibilidad entre la red y el nuevo sistema de monitorización.

Posterior a esto se realizaron las configuraciones correspondientes con todos los parámetros que se debían monitorizar, estas se llevaron a cabo sin eventualidades y de manera organizada y sistemática.

Debido a que para realizar la configuración de todos los ítems, excepto los paquetes ICMP, era necesario conocer los identificadores de objeto de los elementos, se llevó a cabo un proceso de recolección de los OIDs de todos estos. Este proceso resultó parcialmente exitoso, ya que entre los OID recuperados no se obtuvieron los correspondientes con las memorias RAM de los equipos.

Entre los OIDs recuperados se utilizaron los siguientes:

- 1.3.6.1.2.1.2.2.1.10. X para medir el tráfico de entrada a cada interfaz de los enrutadores.
- 1.3.6.1.2.1.2.2.1.16. X: para medir el tráfico de salida a cada interfaz de los enrutadores.

- 1.3.6.1.4.1.9.9.109.1.1.1.1.8.1: para medir en porcentaje el uso de CPU de los enrutadores.

La letra “X” en el caso de los OID utilizados para medir el tráfico en las interfaces corresponde con el último dígito del OID del elemento, ya que el OID para medición del tráfico en una interfaz está estrechamente relacionado con el identificador de la interfaz en la que se realizará dicha medición.

A continuación se presenta un ejemplo, en el cual se puede observar la relación entre los OIDs explicada anteriormente:

Interfaz virtual FastEthernet0/0.11 del enrutador principal de la empresa Servimeta C.A.

El OID obtenido para esta interfaz fue: **1.3.6.1.2.1.2.2.1.2.9,** por lo cual, para medición de su tráfico de salida, la letra “X” será reemplazada por el número nueve (9), que corresponde con el último dígito del identificador de la interfaz. Al insertar el OID resultante (**1.3.6.1.2.1.2.2.1.16.9)** en el espacio correspondiente en la ventana de configuración, se comenzaron a obtener los resultados esperados.

Anteriormente se mencionó que no se logró la recuperación de los OID correspondientes con la memoria RAM de los equipos, y por esta razón no se logró la medición del uso de la misma. Al afrontar esta situación se acudió al soporte técnico de los fabricantes, Cisco Systems, Inc., y se obtuvo como respuesta que el problema se presentaba debido a la falta de actualización del software de los enrutadores. Ya que la resolución de este problema requería de la actualización del software de los *routers*, no se tomaron acciones al respecto.

5.5 Pruebas en la red del cliente

Una vez realizada la instalación y las configuraciones pertinentes, tuvo lugar la fase de pruebas de funcionamiento descrita en el capítulo anterior, fue después de

realizar estas pruebas y verificar los resultados de las mismas, comparándolos con los resultados obtenidos accediendo directamente a los sistemas operativos de los equipos, que se comprobó que eran confiables, por lo que era perfectamente factible que el software quedase en funcionamiento permanente.

5.5.1 Pruebas de monitorización del tráfico y CPU

En las tablas presentadas a continuación se muestran en detalle los ítems configurados a lo largo de la implementación del software en la red del cliente.

Sede	***Router***	**N° Interfaces**	**Uso CPU**	**Tráfico entrada**	**Tráfico salida**	**Ping**	**Total ítems**
Domus	Cisco serie 1800	6	✓	✓	✓	✓	14
Menegrande	Cisco serie 2800	5	✓	✓	✓	✓	12
	Cisco serie 2800	5	✓	✓	✓	✓	12
Propatria	Cisco serie 1800	2	✓	✓	✓	✓	6
Urdaneta	Cisco serie 1800	2	✓	✓	✓	✓	6
Recreo	Cisco serie 1800	2	✓	✓	✓	✓	6
Dorado	Cisco serie 800	4	✓	✓	✓	✓	10
Samat	Cisco serie 800	5	✓	✓	✓	✓	12

Tabla 6. Ítems configurados en los *routers*.
Fuente: Elaboración propia

Sede	**Servidor**	**Ping (ICMP)**	**Total ítems**
Dorado	Dell	✓	1
Samat	Dell	✓	1

Tabla 7. Ítems configurados en los servidores.
Fuente: Elaboración propia

Al finalizar con la configuración de los ítems de monitorización de tráfico de entrada y salida en cada una de las interfaces, se procedió a configurar los ítems de porcentaje de uso de CPU de cada equipo. A petición del administrador de la red, y a fin de no sobrecargar de información los equipos, se configuró que la información que cumpliese 90 días en el sistema se eliminase de forma automática.

Como se explicó anteriormente, estos datos fueron validados con apoyo del equipo de Sannet Soluciones C.A. para el manejo de la información utilizada por los enrutadores y gracias al comando **snmpget**, los resultados obtenidos fueron idénticos a los recopilados por el servidor Zabbix, con lo cual se logró determinar que la obtención de datos utilizando el servidor era confiable. En la siguiente figura se muestra parte del proceso de validación de datos, en este ejemplo particular se puede observar la comparación entre los porcentajes de uso de CPU obtenidos, en la parte superior directamente del enrutador, y en la parte inferior el valor obtenido utilizando el software de monitorización Zabbix.

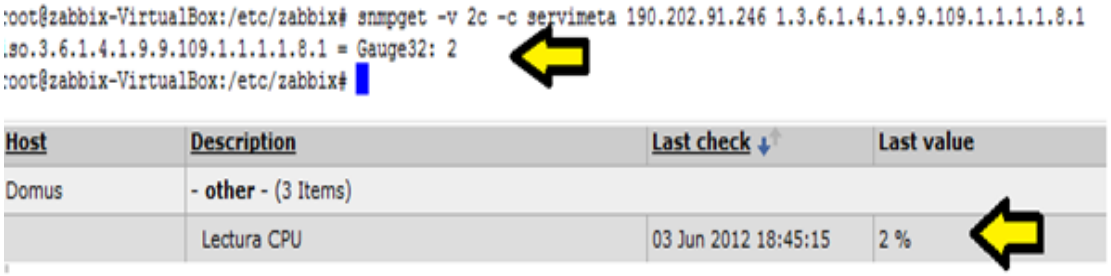


Figura 47. Validación de datos uso de CPU.
Fuente: Elaboración propia

Igualmente, se comprobó la recopilación de datos a lo largo del tiempo, utilizando como herramienta las gráficas, en estas se observó que presentaban variaciones correspondientes con la actividad que estuviese tomando lugar en la red. A continuación se puede observar la fluctuación de datos obtenidos en un período de 3 horas de medición de tráfico de entrada en la red WAN de la sede principal del cliente.

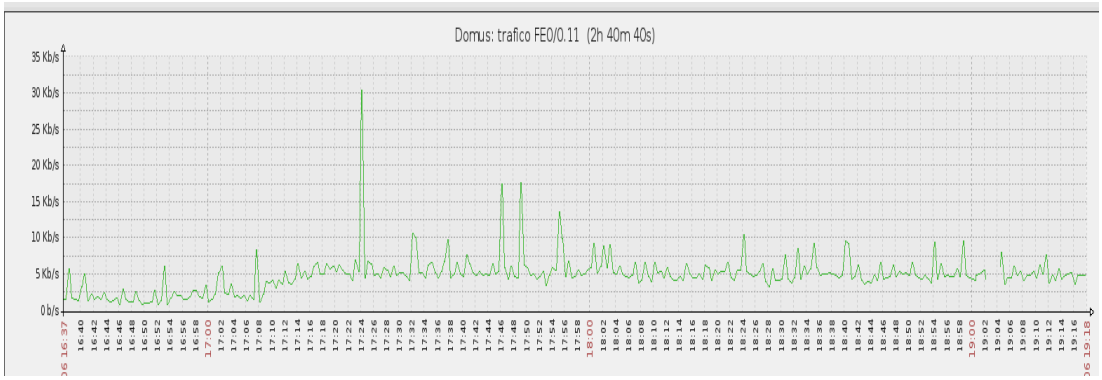


Figura 48. Tráfico de entrada de la WAN del enrutador ubicado en la Sede Domus.
Fuente: Elaboración propia.

5.5.2 Pruebas de funcionamiento de alertas

Una vez finalizada la configuración completa del sistema, en la cual se incluye la configuración de todos los ítems enumerados en las tablas del apartado anterior, así como el servidor SMTP dentro de Zabbix para envío de correos electrónicos, se realizaron pruebas para verificar que estos últimos fuesen recibidos según lo planificado. Tal como se describió en el capítulo 4 del presente trabajo, la metodología utilizada para la comprobación del funcionamiento de las alertas de correo electrónico consistió en la desconexión de algunos de los enlaces de la red del cliente; al realizar la desconexión, se pudo comprobar que se activasen las alarmas correspondientes con el enlace desconectado, entre las cuales debía estar la falla del envío y recepción de respuestas correspondientes con los paquetes **fping** (ICMP).

Después de un tiempo aproximado de una semana, en la cual el sistema permaneció activo, se recibieron quejas por parte del administrador de la red, referentes a la excesiva cantidad de *e-mails* recibidos en ese período, específicamente en las horas de alto tráfico en la red.

Al realizar la evaluación correspondiente, se logró determinar que los problemas de envío y recepción de paquetes ICMP, se debía a los retrasos ocasionados por la congestión característica de las horas “pico”, ya que en estas se realiza la mayor parte del intercambio de información entre los elementos de la red.

Como medida de resolución, se decidió aumentar el tiempo de vida de los paquetes ICMP por un tiempo que les permitiese llegar a su destino a pesar de cualquier retraso presente en la red, de manera que, en adelante, las alertas de correo electrónico recibidas si correspondieron realmente con problemas de conexión en los enlaces.

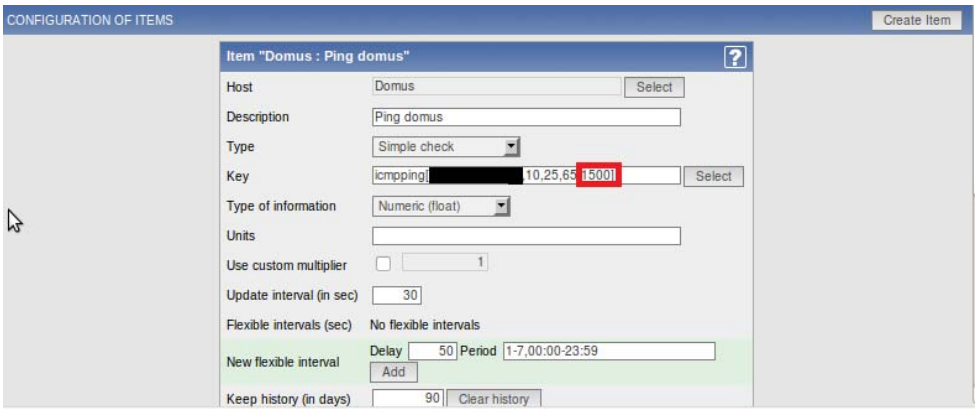


Figura 49. Modificación de tiempo de vida fping.
Fuente: Elaboración propia.

En la imagen anterior se puede apreciar cómo se cambió el tiempo de vida predeterminado de un paquete ICMP de 500 milisegundos a 1500 milisegundos, a fin de asegurar su llegada al destino.

En relación con los objetivos planteados al iniciar el presente trabajo de grado, se logró desarrollar un sistema de gestión para la red de Servimeta C.A., el cual permite detectar fallas casi instantáneamente, haciendo posible que las mismas sean escaladas al nivel de gestión requerido para su resolución. Adicionalmente, se podrán obtener relaciones estadísticas de los problemas presentados en cualquiera de las sedes, de manera que será posible, una vez que el software haya estado en funcionamiento un tiempo suficiente, determinar la propensión a fallas de cada uno de los equipos de la red, y esto a su vez permitirá al administrador de la red darles especial atención a los que así lo requieran y así ofrecer un servicio con garantías.

Sannet Soluciones presta servicio a varias empresas, que no se desempeñan en el área de telecomunicaciones pero, debido a las estrictas políticas de privacidad de

algunas de estas, solo fue posible realizar pruebas con el software en la red de Servimeta C.A.

El manejo de información respecto a las fallas y errores presentes en la red por parte de Sannet Soluciones, fue determinado por los administradores de la red de Servimeta C.A. ya que, debido a sus políticas de privacidad, se rehusaron a permitir que se tuviese acceso remoto a la misma.

		ENTRADA Metro-Metro Domus dorado tunel4	07 May 2012 14:47:27	128 b/s	-24 b/s	Graph
		ENTRADA Metro-Metro Domus Samat tunel0	07 May 2012 14:47:21	440 b/s	+8 b/s	Graph
		PING Domus Cisco 1800	07 May 2012 14:47:16	1	-	Graph
		SALIDA ABA METRO DOMUS SAMAT TUNEL 3	07 May 2012 14:47:30	128 b/s	-24 b/s	Graph
		SALIDA ABA-ABA DOMUS SAMAT TUNEL 5	07 May 2012 14:47:24	128 b/s	-24 b/s	Graph
		SALIDA DOMUS ABA METRO 0/0.11	07 May 2012 14:47:18	0 b/s	-	Graph
		SALIDA DOMUS CISCO-DOMUS 3COM 0/1.14	07 May 2012 14:47:26	447.1 Kb/s	-93.42 Kb/s	Graph
		SALIDA METR-METRO 0/0.962	07 May 2012 14:47:32	25.01 Kb/s	-5.5 Kb/s	Graph
		Salida Metro-Metro Domus dorado tunel1	07 May 2012 14:47:20	152 b/s	-32 b/s	Graph
		SALIDA Metro-Metro Domus dorado tunel4	07 May 2012 14:47:28	248 b/s	-16 b/s	Graph
		SALIDA Metro-Metro Domus Samat tunel0	07 May 2012 14:47:22	432 b/s	+8 b/s	Graph
⊟	Dorado	**- other -** (2 Items)				
		CPU DORADO	07 May 2012 14:47:38	1 %	+1 %	Graph
		PING DORADO 10.14.1.2	07 May 2012 14:47:37	1	-	Graph
⊟	Menegrande	**- other -** (12 Items)				
		CPU MENEGRANDE SEDES	07 May 2012 14:47:44	0 %	-	Graph
		ENTRADA MENEGRANDE-DOMUS 3COM 0/3/1.83	07 May 2012 14:47:39	7.03 Kb/s	+3.06 Kb/s	Graph
		ENTRADA MENEGRANDE-MENEGRANDE F0/0	07 May 2012 14:47:41	1.02 Kb/s	+72 b/s	Graph
		ENTRADA MENEGRANDE-PROPATRIA 0/3/1.82	07 May 2012 14:47:37	1.7 Kb/s	+168 b/s	Graph
		ENTRADA MENEGRANDE-RECREO 0/3/0.84	07 May 2012 14:47:33	13.7 Kb/s	-14.74 Kb/s	Graph
		ENTRADA MENEGRANDE-URDANETA 0/3/1.81	07 May 2012 14:47:35	2.14 Kb/s	-104 b/s	Graph
		PING MENEGRANDE	07 May 2012 14:47:43	1	-	Graph
		SALIDA MENEGRANDE-DOMUS 3COM 0/3/1.83	07 May 2012 14:47:40	8.2 Kb/s	+3.07 Kb/s	Graph
		SALIDA MENEGRANDE-MENEGRANDE F0/0	07 May 2012 14:47:42	960 b/s	+112 b/s	Graph
		SALIDA MENEGRANDE-PROPATRIA 0/3/1.82	07 May 2012 14:47:38	1.22 Kb/s	+48 b/s	Graph
		SALIDA MENEGRANDE-RECREO 0/3/0.84	07 May 2012 14:47:34	20.99 Kb/s	+8.7 Kb/s	Graph
		SALIDA MENEGRANDE-URDANETA 0/3/1.81	07 May 2012 14:47:36	1.73 Kb/s	-472 b/s	Graph
⊟	Menegrande Movistar	**- other -** (10 Items)				
		CPU MENEMOVISTAR	07 May 2012 14:47:24	1 %	-1 %	Graph
		ENTRADA MENEMOVISTAR DORADO TUNEL 2	07 May 2012 14:47:19	128 b/s	-	Graph
		ENTRADA MENEMOVISTAR F 0/1	07 May 2012 14:47:47	18.94 Kb/s	-23.56 Kb/s	Graph
		ENTRADA MENEMOVISTAR MENESEDES F0/0	07 May 2012 14:47:45	1.02 Kb/s	+40 b/s	Graph
		ENTRADA MENEMOVISTAR SAMAT TUNEL 3	07 May 2012 14:47:21	152 b/s	+24 b/s	Graph

Figura 51. Monitorización de varias sedes
Fuente: Elaboración propia.

CAPÍTULO VI

Conclusiones y Recomendaciones

En el siguiente capítulo se presentan las conclusiones obtenidas al finalizar el trabajo especial de grado, y las recomendaciones que se consideran necesarias para la implementación de este software a futuro en un entorno similar.

6.1 Conclusiones

En general, se considera que los objetivos fueron cumplidos a cabalidad, ya que se logró la implementación exitosa de un sistema de monitorización en la red, que además cumplió con todas las exigencias del cliente, cubriéndose así las necesidades del mismo.

El instrumento Zabbix cumplió con todas las expectativas y demostró ser una herramienta eficaz al momento de detectar fallas en los distintos dispositivos de red. Los logros obtenidos al utilizar esta herramienta de software libre, demuestran que, correctamente implementadas, las soluciones de software libre son tan eficientes como el resto de las soluciones existentes.

El éxito en la implementación del sistema se debe principalmente a que se trabajó bajo un esquema en el cual se definieron de manera clara las necesidades presentes, y toda la labor se realizó en función de satisfacerlas.

Si en algún momento se desease, o resultase necesario, se podría implementar el mismo software en otras redes, adaptando la configuración a las necesidades presentes en la misma, al igual que se hizo previo a la implementación llevada a cabo para la realización de este trabajo.

Las herramientas de monitorización son vitales para prevención y anticipación de problemas en una red, hoy en día, a medida que las redes se expanden, los costos

de mantenimiento aumentan, por ello es necesario hallar las fallas en el menor tiempo posible.

6.2 Recomendaciones

Se recomienda que para que el software de monitorización funcione de manera óptima todo el tiempo se le realicen procesos de mantenimiento periódicamente; igualmente se recomienda que se actualice con la mayor frecuencia posible, ya que esto permitirá obtener el mejor rendimiento del mismo.

Se considera que sería una excelente iniciativa la oferta de cursos y certificaciones para el manejo e implementación de este software, ya que actualmente no se cuenta con los mismos en Venezuela.

REFERENCIAS BIBLIOGRÁFICAS

About Nagios. Recuperado el 16 de enero de 2012 de, http://www.nagios.org/about

Arazo, M. (2011). *Diseño e implementación de una sistema de monitorización y control para un módem satélite a través del protocolo SNMP.* Proyecto Final de Carrera, Escola Tècnica Superior d'Enginyeria de Telecomunicació de Barcelona, Universitat Politècnica de Catalunya, Barcelona, España.

Arquitectura de administración OSI. Recuperado el 19 de enero de 2012, de http://www.arcesio.net/osinm/osinminformacion.html

Case, Fedor, Schoffstall, & Davin. (05/1990). *RFC1157*. Recuperado el 22 de diciembre de 2011, de http://www.ietf.org/rfc/rfc1157.txt

Castells, M. (2006). *La sociedad red: una visión global.* Madrid, España. Alianza Editorial.

Cisco Systems, Inc. (2007). *CCNA Exploration 4.0 Aspectos básicos de networking*.

Cisco Systems, Inc. (2007). *CCNA Exploration 4.0 Conceptos y protocolos de enrutamiento*.

Comandos. Recuperado el 10 de abril de 2012 de, http://www.guia-ubuntu.org/index.php?title=Comandos

Configuració d' una màquina UNÍX com a un router. Recuperado el 20 de enero de 2012, de http://docencia.ac.upc.edu/FIB/STD/lab/IP_forwarding.pdf

Consulting and Implementation Services. Recuperado el 16 de enero de 2012 de, http://www.nagios.com/services/consulting/

Diccionario de la Real Academia Española.

El protocolo ICMP. Recuperado el 17 de junio de 2012 de, http://es.kioskea.net/contents/internet/icmp.php3

El protocolo ICMP. Recuperado el 16 de junio de 2012 de, http://neo.lcc.uma.es/evirtual/cdd/tutorial/red/icmp.html

Features. Recuperado el 16 de enero de 2012 de, www.cacti.net/features.php

Gutiérrez, Sancho & Casas. *Estudio sobre las VPN (Redes Privadas Virtuales)*. Recuperado el 17 de enero de 2012, de http://www.redes-linux.com/manuales/vpn/Estudio_VPN.pdf

Interiano, E. *Redes de Computadoras Ethernet*. Recuperado el 15 de enero de 2012, de http://www.ie.itcr.ac.cr/faustino/Redes/Clase8/4.2Ethernet.pdf

Krawczyk, Bellare & Canetti. (02/1997). *RFC 2104*. Recuperado el 29 de diciembre de 2011, de http://www.faqs.org/rfcs/rfc2104.html

La Empresa. Recuperada el 14 de junio de 2012, de http://www.sannet.com.ve/laempresa.html

Martínez, Narváez. (2010). *Implementación de Zabbix como herramienta de monitorización de infraestructura informática de la compañía Santini System Group Ltda.* Proyecto final de carrera, Facultad de Ingeniería Electrónica, Universidad Santo Tomás, Bogotá, Colombia.

MIB Browser. Recuperada el 29 de diciembre de 2011, de http://www.oidview.com/images/mibbrowser-full.gif

Monitor everything. Recuperado el 17 de junio de 2012 de, http://www.zabbix.com/es/monitor_everything.php

MySQL. Recuperado el 10 de abril de 2012 de, http://www.guia-ubuntu.org/index.php?title=MySQL

Nagios Training. Recuperado el 16 de enero de 2012 de, http://www.nagios.com/services/training/

Red de Área Extensa (WAN). Recuperada el 14 de enero de 2012, de http://profecarolinaquinodoz.com/principal/wp-content/uploads/2009/04/wan1.gif

Saperia, J. (2002). *SNMP at the edge, Building effective management systems.* USA. Editorial McGraw-Hill Professional.

SNMPv2. Recuperado el 03 de diciembre de 2011, de http://www4.ujaen.es/~mdmolina/grr/Tema%203.pdf

Tanenbaum, A. (2003). *Redes de Computadoras.* Mexico. Editorial McGraw-Hill.

Valles, K. *SNMPv3*. Recuperado el 03 de enero de 2012, de http://neutron.ing.ucv.ve/revista-e/No6/Valles%20Kirssy/SNMPV3/Snmpv3.htm

What is Cacti?. Recuperado el 16 de enero de 2012 de, http://www.cacti.net/what_is_cacti.php

Zabbix manual. Recuperado el 15 de enero de 2012 de, http://www.zabbix.com/documentation/1.8/manual

APÉNDICES

APÉNDICE A

Abreviaturas y Acrónimos

AES *Advanced Encryption Standard* – Estándar Avanzado de Cifrado

ASN.1 *Abstract Syntax Notation 1* – Notación Abstracta de Sintaxis 1

ATM *Asynchronous Transfer Mode* – Modo de Transferencia Asincrona

CPU *Central Processing Unit* – Unidad Central de Procesamiento

DES *Data Encryption Standard* – Estándar de Cifrado de Datos

HEMP *High-level Entity Management Protocol* – Protocolo de Gestión de Entidades de Alto Nivel

HEMS *High-level Entity Management System* – Sistema de Gestión de Entidades de Alto Nivel

HMAC *Hash-based Message Authentication Code* – Código de Autenticación de Mensajes basado en Hash

HTTP *Hypertext Trasfer Protocol* – Protocolo de Transferencia de Hipertexto

IAB *Internet Architecture Board* – Junta de Arquitectura de Internet

IANA *Internet Assigned Numbers Authority* – Autoridad de Asignación de Números de Internet

ICMP *Internet Control Message Protocol* – Protocolo de Mensajes de Control de Internet

IEEE *Institute of Electrical and Electronics Engineers* – Instituto de Ingenieros Electricos y Electrónicos

IETF *Internet Engineering Task Force* – Fuerza de Tareas de Ingeniería de Internet

IP *Internet Protocol* – Protocolo de Internet

IT *Information Technology* – Tencnología de Información

ISO *International Organization for Standardization* – Organización Internacional para la Estandarización

ITU *International Telecommunications Union* – Unión Internacional de Telecomunicaciones

LAN *Local Area Network* – Red de Área Local

LED *Light Emitting Diode* – Diodo Emisor de Luz

MAC *Media Access Control* – Control de Acceso al Medio

MAN *Metropolitan Area Network* – Red de Área Metropolitana

MIT *Massachusets Institute of Technology* – Instituto Tecnológico de Massachusets

MD2 *Message-Digest algorithm 5* – Algoritmo de Digestión de Mensajes 2

MD4 *Message-Digest algorithm 5* – Algoritmo de Digestión de Mensajes 4

MD5 *Message-Digest algorithm 5* – Algoritmo de Digestión de Mensajes 5

MIB *Management Information Base* – Base de Gestión de Información

NMS *Network Management Station* – Estación de Gestión de Red

OID *Object Identifier* – Identificador de Objeto

OSI *Open System Interconnection* – Interconexión de Sistema Abierto

PDU *Protocol Data Unit* – Unidad de Datos de Protocolo

PHP *PHP Hypertext Pre-processor* – Procesador de hipertexto PHP

PPP *Point-to-Point Protocol* – Protocolo Punto a Punto

PSTN *Public Switching Telephone Network* – Red de Teléfono de Conmutación Publica

RAM *Random Access Memory* – Memoria de Acceso Aleatorio

RFC *Request For Comments* – Solicitud de Comentarios

ROM *Read Only Memory* – Memoria de Solo Lectura

RRD *Round-Robin Database* – Base de Datos Round Robin

SGMP *Simple Gateway Monitoring Protocol* – Protocolo Simple de Monitorización de Puertas de Enlace

SHA-1 *Secure Hash Algorithm-1* – Algoritmo Seguro de Hash-1

SMI *Structure of Management Information* – Estructura de Información de Gestión

SMTP *Simple Mail Transfer Protocol* – Protocolo Simple de Transferencia de Correo

SNMP *Simple Network Management Protocol* – Protocolo Simple de Gestión de Red

TCP *Transmission Control Protocol* – Protocolo de Control de Transmisión

UDP *User Datagram Protocol* – Modelo de Seguridad basado en Usuario

UTP *Unshielded Twisted Pair* – Par Trenzado Desprotegido

USM *User-based Security Model* – Modelo de Seguridad Basado en Usuario

VACM *View-based Access Control Model* – Modelo de Control de Acceso basado en Vistas

VPN *Virtual Private Network* – Red Virtual Privada

WAN *Wide Area Network* – Red de Área Extensa

XML *eXtensible Markup Language* – Lenguaje de Marcas Extensibles