UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

MONITORIZACIÓN DE SERVICIOS DE RED Y SERVIDORES

T E S I S

QUE PARA OBTENER EL GRADO ACADÉMICO DE

I N G E N I E R O E N C O M P U T A C I Ó N

P R E S E N T A N

Delgadillo Rivera José Luis
García Ronquillo Leonardo Daniel

DIRECTOR DE TESIS:
Ing. Noé Cruz Marín

MÉXICO, D. F. Junio, 2010

MONITORIZACIÓN DE SERVICIOS DE RED Y SERVIDORES

Agradecimientos y Dedicatorias

Agradezco a mis padres, María Francisca y José Encarnación, por el amor, el esfuerzo y el apoyo incondicional, con los cuales he logrado terminar mis estudios profesionales que constituyen la herencia más grande que pudiera recibir.

Gracias a Heidi Elizabeth por todo el cariño y gran apoyo, por siempre creer en mí y por ponerle sabor a la vida.

Gracias a todos mis amigos sin excepción, por brindarme tan buenos momentos, y por que han estado conmigo en los instantes en que los he necesitado.

Agradezco a la Universidad Nacional Autónoma de México por haberme brindado el espacio y la oportunidad de formarme profesionalmente, haciendo de mí una persona preparada para enfrentarme a la vida y contribuyendo de manera significativa en mi desarrollo personal.

Gracias a la Unidad de Servicios de Cómputo Académico (UNICA) por las todas las oportunidades de crecimiento que me ha brindado. A sus becarios por brindarme su amistad y confianza. A la generación de becarios número 52 en especial a Darío Eduardo, Leonardo Daniel, José Guadalupe, Jaime Romo y Santa Rosa porque aprendí mucho de cada uno de ustedes y por todos los momentos de diversión y entretenimiento que pasamos juntos.

Gracias al Ing. Noé Cruz por haberme brindado su confianza y transmitido tantos conocimientos como profesor, jefe y director de tesis.

José Luis Delgadillo Rivera

MONITORIZACIÓN DE SERVICIOS DE RED Y SERVIDORES

Agradecimientos y Dedicatorias

Antes que nada les agradezco a mis padres Estela y Leonardo todo el apoyo que me han dado en la vida para poder llegar a ser quien soy. Siempre estuvieron conmigo alentándome a seguir en la lucha y nunca renunciar a mis metas. Gracias a ustedes aprendí el significado de la responsabilidad.

Agradezco a la Universidad Nacional Autónoma de México y en especial a la Facultad de Ingeniería, que me han dado todo para formarme como persona profesional y siempre llevaré en alto el orgullo de ser puma.

Gracias a mis abuelos por su cariño y que siempre confiaron en mí. Muchos ya no están conmigo pero siempre los tendré en mi corazón.

Gracias a toda mi familia, a mis primos y a mis tíos. Gracias a mis hermanitas, Imelda y Marisol, por soportarme todos estos años. Por todo su apoyo y su cariño.

Gracias a mis hermanos de la vida Diana, Magaly, Paola, Oscar y Rodolfo por encontrarlos en mi camino, me siento afortunado de tenerlos a mi lado. Siempre estuvieron conmigo en las buenas y en las malas, siempre obtuve un buen consejo de ustedes. Con nadie he vivido tantas experiencias y con nadie he reído tanto hasta llorar. Siempre serán parte importante de mi vida. Gracias a ustedes aprendí que la amistad también significa familia.

Gracias a UNICA, por todas las oportunidades que me dio para seguir aprendiendo de mi carrera. Gracias a todos sus becarios por ese gran compañerismo.

Gracias a todos los amigos que conocí en la facultad, Gustavo, Fabiola, José Luis Delgadillo, José Guadalupe, Jaime, Darío, mi tocayo Leonardo, Germán, Santa, Armando, José Luis Carrillo porque cada día aprendí de ustedes algo nuevo.

Gracias al Ing. Noé Cruz por el apoyo y las enseñanzas como profesor y jefe.

Leonardo Daniel García Ronquillo

ÍNDICE GENERAL

ÍNDICE DE FIGURAS

OBJETIVOS

- Crear un sistema para la monitorización del desempeño de la red y los servidores utilizados en la Unidad de Servicios de Cómputo Académico de la Facultad de Ingeniería de la UNAM.

- Contabilizar y llevar un registro de los datos relevantes de servidores y switches.

- Implementar diferentes funciones para mandar alertas así como reportes.

INTRODUCCIÓN

Hoy en día es importante conocer el estado de los sistemas que se manejan dentro de una organización para garantizar el correcto funcionamiento de los mismos y no perder tiempo y dinero, ya que generalmente estos sistemas o servidores son trascendentales porque manejan información crítica. Es necesario que estos sistemas se mantengan prestando el servicio el mayor tiempo posible, si existiera una anomalía o percance se tendría que avisar oportunamente a los administradores de servidores para que tomen las medidas necesarias para evitar una denegación de servicios o pérdida de información.

Es importante que estos avisos sean confiables ya que si se reportaran falsas alarmas el sistema perdería credibilidad. Cabe resaltar que el modo de envío de alertas debe tener más opciones de envío para asegurar que el mensaje de falla en un sistema llegue realmente a la persona adecuada y no se quede simplemente en su bandeja de entrada.

Para brindar un buen servicio a los usuarios es de vital importancia contar con un esquema de monitorización, el cual sea capaz de notificar fallas de red y mostrar su comportamiento mediante el análisis de tráfico. Se deben tomar en cuenta elementos importantes para ser monitorizados, así como herramientas especiales que se utilizan en esta tarea.

La carga de información que viaja a través de la red normalmente es baja. En una red bien diseñada, la mayor parte del tiempo se trabaja a niveles óptimos, donde los paquetes viajan sin mayor problema. También existen casos en donde el comportamiento de la red es inusual o simplemente deja de prestar el servicio y ahí es cuando la monitorización entra en juego como una herramienta sumamente importante que nos ayudará a detectar problemas y dar atención oportuna a sucesos de mayor relevancia. Al monitorizar la red podemos darnos cuenta en dónde se encuentra el problema de una manera más específica o al menos nos dirá en dónde no está el problema.

Existen dos formas de poder revisar el estado actual de los servicios y de los servidores de una organización, se puede realizar de forma manual, o de forma automatizada con herramientas diseñadas especialmente para esta tarea, el método manual para recabar datos de algún sistema o monitorizarlo, dependerá de las habilidades y experiencia del administrador del equipo o de la red, en algunos casos, esta forma de obtener los datos suele ser tediosa y es susceptible a fallas humanas al momento de extraer los datos.

Para el método automatizado se utilizan scripts de lectura de bitácoras o herramientas especializadas en la adquisición de los datos relevantes de los servidores, esto suele ser más práctico para el administrador, y es una forma más rápida, clara y confiable de conocer el estado actual de los servicios de red o de un servidor específico.

La correcta interpretación de los datos que se logran obtener con el proceso de monitorización es sumamente importante ya que ayuda a los administradores de los servicios a tomar decisiones oportunas agilizando de esta manera la solución a los problemas que se presenten. No sólo se puede actuar de forma reactiva, el poder analizar oportunamente el estado de los equipos con que se cuenta, los servicios que se tienen y el estado de la red nos permite actuar de forma preventiva al percatarse de algún posible fallo en el sistema, evitando así de manera considerable la interrupción de los servicios ya sean de red o de los servidores.

El contar con uno o varios servidores dentro de una organización hace que la monitorización de los mismos sea un asunto importante. En un sistema automatizado se debe discriminar la información recopilada de éstos para que sólo se dé aviso de los eventos que necesiten atención.

El esquema de monitorización propuesto en esta tesis pretende informar del correcto comportamiento de la red, los servidores y servicios de una organización, en el caso específico de la Unidad de Servicios de Computo Académico (UNICA), el Departamento de Redes y Administración de Servidores (DROS) se encarga tanto de la administración de los servidores como de la red.

En el primer capítulo de esta tesis se definen los conceptos básicos de una red de datos, así como los tipos de servidores que se encuentran dentro de una red. Se describen de forma breve los modelos de referencia OSI ("*Open System Interconnection*") y TCP/IP ("*Transmission Control Protocol/ Internet Protocol*"). También se describen aspectos importantes dentro de la monitorización de dispositivos y cómo el protocolo SNMP ("*Simple Network Management Protocol*") ayuda a efectuar esta tarea. Al final se definen características de los sistemas Linux, que son la base de operación de algunos servidores.

En el capítulo dos se evalúan herramientas de monitorización que existen actualmente en el mercado. Algunas de ellas son software libre[1] y algunas otras son software propietario[2] y cada una de ellas está orientada a objetivos específicos de monitorización y se expone un comparativo entre ellas.

El tercer capítulo aborda las diferentes metodologías de ingeniería de software como guía para la realización de un sistema automatizado.

El cuarto capítulo es de vital importancia para esta tesis ya que muestra la implementación del esquema de monitorización propuesto, basado en diferentes metodologías de desarrollo y de toma de decisiones. Con este capítulo se pretende

[1] Software libre se refiere a la libertad de los usuarios para ejecutar, copiar, distribuir, estudiar, cambiar y mejorar el software.

[2] Software propietario es aquel código fuente que no está disponible o el acceso a éste se encuentra restringido por un acuerdo de licencia.

abordar el caso específico de la Unidad de Servicios de Cómputo Académico analizando el estado actual de UNICA, tomando en cuenta los recursos que se tienen, y los que se puedan adicionar para poder evaluar el buen desempeño de los servicios que se brindan.

Capítulo 1 **Conceptos básicos para la implementación de un Sistema de Monitorización**

- 1.1 Conceptos Básicos
- 1.2 Monitorización
- 1.3 SNMP
- 1.4 El sistema operativo Linux
- 1.5 Otros sistemas Operativos

1 CONCEPTOS BÁSICOS PARA LA IMPLEMENTACIÓN DE UN SISTEMA DE MONITORIZACIÓN

1.1 Conceptos Básicos

Se comenzará definiendo y explicando algunos conceptos básicos que a lo largo de esta tesis se manejarán.

1.1.1 Red de Datos.

- **Definición**

La palabra **datos** se refiere a hechos, conceptos e instrucciones presentados en cualquier formato. En el contexto de los sistemas de información basados en computadoras, los datos se representan con unidades de información binaria, o bits, producidos y consumidos en forma de ceros y unos.

Una red es un conjunto de dispositivos llamados nodos, conectados por enlaces de un medio físico. Un nodo puede ser una computadora, una impresora o cualquier otro dispositivo capaz de enviar y/o recibir datos generados por otros nodos de la red.

Una red de datos es un conjunto de computadoras y dispositivos conectados entre sí mediante una o más vías de transmisión (Forouzan, 2002). La red existe para cumplir un determinado objetivo que es la transmisión de datos donde se realiza el intercambio de datos entre dos dispositivos a través de algún medio de transmisión. Esta transmisión se considera local si los dispositivos se encuentran en un área geográfica delimitada y se considera remota si los dispositivos están separados por una distancia considerable.

Este intercambio de datos es la base de muchos servicios, basados en redes de computadoras y se han convertido en una parte indispensable de los negocios, la industria y el entretenimiento. Existen muchos ejemplos donde las redes de datos son usadas por ejemplo en una aplicación de venta donde se capturen los pedidos ya sean por internet o vía telefónica y éstos se conecten a una red de procesamiento de pedidos. Un ejemplo más es la transferencia de dinero sin tener que ir a un banco, un cajero automático es un ejemplo de transferencia electrónica de fondos. La mayoría de los servicios que se ofrecen actualmente están ligados de una u otra forma a una red de datos.

1.1.2 Topologías

El término topología se define como la forma en que está diseñada la red ya sea de manera física o lógica. La topología de una red es la representación geométrica de la

relación entre todos los enlaces y los dispositivos que los enlazan entre sí (Forouzan, 2002). Existen topologías básicas como son la de malla, estrella, árbol, bus y anillo.

- **Topología en malla**

En este tipo de configuración cada dispositivo está conectado a uno o más de los otros dispositivos (también llamados nodos) y así, es posible llevar los datos de un nodo al otro por diferentes caminos como se muestra en la figura 1.1. Una diferencia importante con las otras topologías es que no se requiere tener un concentrador o servidor central. Entre las ventajas que se tienen con esta topología son: el sistema completo no se inhabilita si un enlace falla, la seguridad de saber que cuando un mensaje viaja a través de un enlace, solamente lo ve el receptor adecuado, también es más fácil detectar y aislar fallos. Entre las desventajas, están la cantidad de cable necesario y el número de puertos de entrada/salida. El costo para implementar esta configuración es excesivamente cara. Por estas razones, la topología en malla se utiliza junto con otras topologías para formar una topología híbrida.

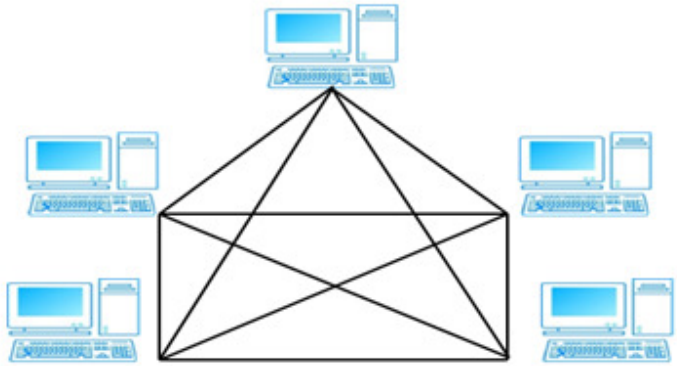


Figura 1.1 Topología en malla

- **Topología en Estrella**

En esta topología cada dispositivo solamente tiene un enlace directo con el concentrador central. Los dispositivos no están directamente enlazados entre sí. La topología en estrella no permite el tráfico directo entre dispositivos. El controlador actúa como un intercambiador, es decir, si un dispositivo quiere enviar un dato a otro, se envían los datos al controlador para que éste retransmita ese dato al dispositivo final (figura 1.2). La ventaja es que es más barata y fácil de instalar. Si falla un enlace, solamente ese enlace se verá afectado. Todos los demás enlaces permanecerán activos. Esta cualidad permite identificar fallas de una manera más sencilla. La desventaja es que cada nodo debe estar enlazado con el nodo central, por ello, en la estrella se requiere más cable que en otras topologías de red, excepto en la de malla.

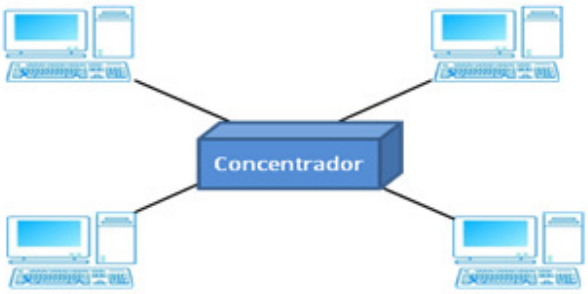


Figura 1.2 Topología en estrella

- **Topología en Árbol**

La topología en árbol es una variante de la de estrella. Los nodos del árbol están conectados a un concentrador central que controla el tráfico de la red. Sin embargo, no todos los dispositivos se conectan directamente al concentrador central. La mayoría de los dispositivos se conectan a un concentrador secundario que, a su vez, se conecta al concentrador central que generalmente es un hub[3] o switch (figura 1.3). Las ventajas y desventajas son, casi las mismas que las de una estrella. Sin embargo, el incluir concentradores secundarios puede incrementar la distancia a la que puede viajar la señal entre dos dispositivos y permite a la red aislar y priorizar las comunicaciones de distintas computadoras.

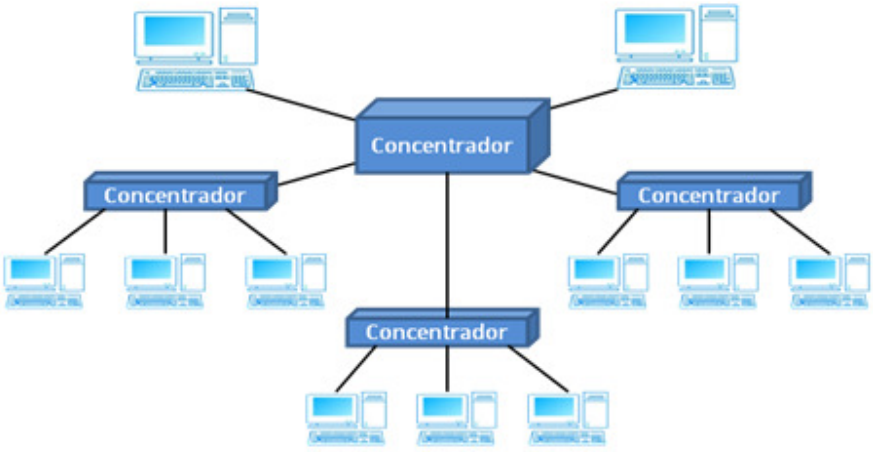


Figura 1.3 Topología de árbol

[3] Dispositivo que permite centralizar el cableado de una red y poder ampliarla

- **Topología de Bus**

Una topología en bus es multipunto, es decir, un cable largo actúa como una red troncal que conecta a todos los dispositivos de red como lo ilustra la figura 1.4. Existen muchas limitantes para esta configuración. Cuando las señales viajan a través de la red troncal, parte de la energía se transforma en calor, lo que se traduce en debilitamiento de la señal a medida que viaja por el cable. Por esta razón, hay un límite en el número de conexiones que un bus puede soportar y en la distancia entre estas conexiones. Como ventaja de esta topología está la sencillez de instalación, menor uso de cable que en otras topologías. Entre sus desventajas está la dificultad para agregar nuevos dispositivos y si el cable de bus llega a fallar, interrumpe todas las comunicaciones.

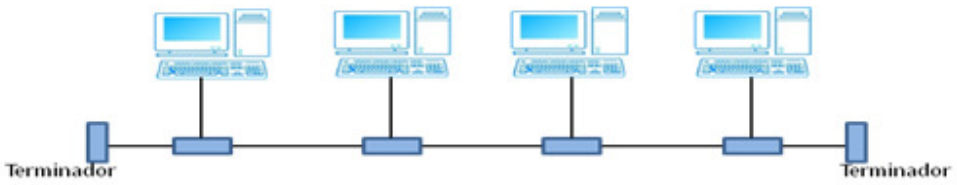


Figura 1.4 Topología en bus

- **Topología en Anillo**

Cada dispositivo tiene una línea de conexión directa solamente con los dos dispositivos que están a sus lados. La señal pasa a lo largo del anillo en una dirección hasta que alcanza su destino. Un anillo es relativamente fácil de instalar y reconfigurar. Cada dispositivo está enlazado solamente a sus vecinos inmediatos (figura 1.5). Las únicas restricciones están relacionadas con aspectos del medio físico y el tráfico. Sin embargo, el tráfico unidireccional puede ser una desventaja, una rotura del anillo puede inhabilitar toda la red.

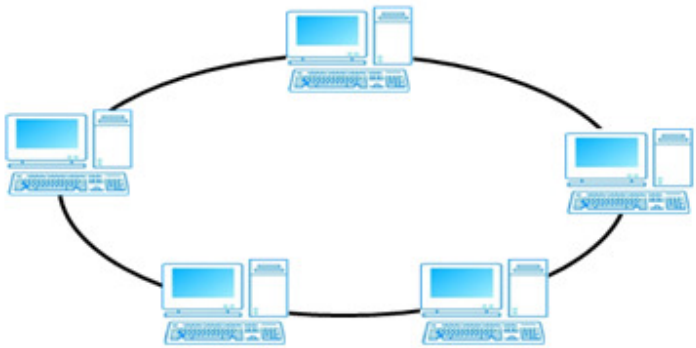


Figura 1.5 Topología en Anillo

1.1.3 Servidor

- **Definición**

Un servidor es un programa que se ejecuta en una máquina remota y que ofrece un servicio a los clientes. Cuando arranca, abre una puerta para la llegada de las peticiones de los clientes, pero nunca termina hasta que no se le solicite expresamente que lo haga. Un programa servidor es un programa infinito. Una vez iniciado se ejecuta de manera constante. Espera la llegada de peticiones de los clientes. Cuando una petición llega, responde a la misma.

- **Tipos de servidores**

En la actualidad existen diversos tipos de servidores, orientados a dar determinados servicios, a continuación se explicarán los servidores más comunes.

- **Servidor de aplicaciones**

Es un tipo de servidor que permite el procesamiento de datos de una aplicación de cliente. Un servidor de aplicaciones generalmente administra la mayor parte de las funciones lógicas del negocio y de acceso a los datos de la aplicación. Los principales beneficios de la aplicación de la tecnología de servidores de aplicación son la centralización y la disminución de la complejidad en el desarrollo de aplicaciones.

- **Servidor de bases de datos**

Un servidor de base de datos es un programa que provee servicios de base de datos a otros programas u otras computadoras

- **Servidor FTP**

Uno de los servicios más antiguos de Internet, File Transfer Protocol permite mover uno o más archivos.

- **Servidor de correo**

Los servidores de correo mueven y almacenan el correo electrónico a través de las redes corporativas y a través de Internet.

- **Servidor web**

Un servidor web es un programa que implementa el protocolo HTTP (hypertext transfer protocol) y está diseñado para transferir lo que llamamos hipertextos, páginas web o páginas HTML.

- **Servidor Proxy**

Se sitúan entre un programa del cliente y un servidor para filtrar peticiones, mejorar el funcionamiento y compartir conexiones.

- **Servidor DNS**

Un servidor DNS ("*Domain Name System*") se utiliza para proveer a las computadoras clientes un nombre equivalente a las direcciones IP. El uso de este servidor es transparente para los usuarios cuando su configuración es correcta y adecuada.

- **Concepto cliente-servidor**

Un programa de aplicación, llamado cliente, se ejecuta en una máquina local, solicita un servicio a otro programa de aplicación, llamado servidor, que se ejecuta en una máquina remota. Muchos clientes pueden utilizar los servicios de un servidor. El servidor proporciona un puerto de comunicaciones donde deben conectarse todos los clientes que deseen obtener dicho servicio. Lo que hace el cliente es abrir el canal de comunicaciones para conectarse a la dirección de red atendida por el servidor. Después envía al servidor un mensaje de petición de servicio y espera hasta recibir respuesta. Finalmente cierra el canal de comunicación y termina la ejecución del proceso. El proceso del servidor es abrir el canal de comunicación e informar a la red tanto de la dirección a la que le responderá como de su disposición para aceptar peticiones de servicio. Después, espera a que el cliente realice una petición de servicio en la dirección que él tiene declarada. Cuando recibe una petición de servicio, atenderá al cliente y finalmente la conexión es cerrada.

1.1.4 Modelo OSI

Un sistema abierto es un modelo que permite que dos sistemas diferentes se puedan comunicar independientemente de la arquitectura que tengan. Así, el modelo de referencia de interconexión de sistemas abiertos, conocido como modelo OSI ("*Open System Interconnection*"), creado por la ISO ("*Organización Internacional de Normalización*"), tiene como fin poner orden entre todos los sistemas y componentes requeridos en la transmisión de datos, además de simplificar la interrelación entre fabricantes. Así, todo dispositivo de cómputo y telecomunicaciones podrá ser referenciado al modelo y por ende concebido como parte de un sistema interdependiente

con características muy precisas en cada nivel. El modelo OSI está compuesto por siete capas (ver figura 1.6). La capa física en el nivel 1, la capa de enlace de datos en el nivel 2, la capa de red en el nivel 3, la capa de transporte en el nivel 4, la capa de sesión en el nivel 5, la de capa de presentación en el nivel 6 y finalmente la capa de aplicación en el nivel 7. (Telecomunicaciones, 2002)

Figura 1.6 Modelo OSI

- **Nivel Físico**

El nivel físico coordina las funciones necesarias para transmitir el flujo de datos a través de un medio físico. Trata con las especificaciones eléctricas y mecánicas de la interfaz y del medio de transmisión. También define los procedimientos y las funciones que los dispositivos físicos y las interfaces tienen que llevar a cabo para que sea posible la transmisión.

- **Nivel de enlace de datos**

El nivel de enlace de datos transforma el nivel físico, un simple medio de transmisión, en un enlace fiable y es responsable de la entrega nodo a nodo. Hace que el nivel físico aparezca ante el nivel de red como un medio libre de errores.

- **Nivel de red**

El nivel de red es responsable de la entrega de un paquete desde el origen al destino a través de múltiples redes. Mientras que el nivel de enlace de datos supervisa la entrega del paquete entre dos sistemas de la misma red, el nivel de red asegura que cada paquete vaya del origen al destino.

- **Nivel de transporte**

El nivel de transporte es responsable de la entrega origen a destino de todo el mensaje. En el nivel de red se supervisa la entrega de paquetes individuales sin reconocer ninguna relación entre estos paquetes. El nivel de transporte asegura que todo el mensaje llegue intacto y en orden, supervisando tanto el control de errores como el control de flujo a nivel origen a destino.

- **Nivel de sesión**

El nivel de sesión es el controlador de diálogo de la red. Establece, mantiene y sincroniza la interacción entre sistemas de comunicación.

- **Nivel de presentación**

El nivel de presentación está relacionado con la sintaxis y la semántica de la información intercambiada entre dos sistemas.

- **Nivel de aplicación**

El nivel de aplicación permite al usuario acceder a la red. Proporciona las interfaces de usuario y el soporte para servicios como el correo electrónico, el acceso y la transferencia de archivos remotos, la gestión de datos compartidos y otros tipos de servicios para información distribuida.

1.1.5 Modelo TCP/IP

La familia de protocolos TCP/IP se desarrolló antes que el modelo OSI. Por lo tanto, los niveles de Protocolo de Control de Transmisión/Protocolo de Red (TCP/IP) no coinciden exactamente con los del modelo OSI. La figura 1.7 ilustra la familia de protocolos TCP/IP. Está compuesta por cuatro niveles: acceso a red, internet, transporte y aplicación.

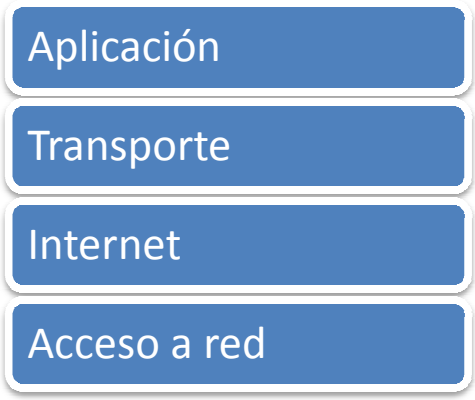


Figura 1.7 Modelo TCP/IP

- **Acceso a red**

Especifica información detallada de cómo se envían físicamente los datos a través de la red, que incluye cómo se realiza la señalización eléctrica de los bits mediante los dispositivos de hardware que conectan directamente con un medio de red, como un cable coaxial, un cable de fibra óptica o un cable de cobre de par trenzado. Los protocolos usados en este nivel son Ethernet, Token Ring, FDDI, X.25, Frame Relay, RS-232, v.35.

- **Internet**

Empaqueta los datos en datagramas IP, que contienen información de las direcciones de origen y destino utilizada para reenviar los datagramas entre hosts y a través de redes. Realiza el enrutamiento de los datagramas IP. Se usan los protocolos IP, ICMP, ARP, RARP.

- **Transporte**

Permite administrar las sesiones de comunicación entre equipos host. Define el nivel de servicio y el estado de la conexión utilizada al transportar datos. Los protocolos que se manejan en este nivel son TCP, UDP.

- **Aplicación**

Define los protocolos de aplicación TCP/IP y cómo se conectan los programas de host a los servicios del nivel de transporte para utilizar la red. Se usan los protocolos HTTP, Telnet, FTP, TFTP, SNMP, DNS, SMTP, X Windows y otros protocolos de aplicación. (Microsoft, El modelo TCP/IP, 2003)

1.1.6 Puertos

Son canales que utiliza el subsistema de red para redireccionar la información al programa apropiado. Son una numeración lógica que se asigna a las conexiones tanto en el origen como en el destino. No tiene ninguna significación física. Están representados por un número de 16 bits usado para identificar los puntos finales de la conexión, en las cabeceras TCP o UDP. Los números de puerto oscilan entre 0 y 65,535. Se tienen tres categorías para su clasificación:

- Puertos bien conocidos o reservados

Son puertos del 0 al 1023. Se utilizan para servicios de red bien conocidos como son FTP[4], HTTP[5], SMTP[6], DNS[7] entre otros. TCP[8] y UDP[9] utilizan estos puertos para determinar el servicio correcto ya que son puertos predeterminados para cada aplicación y están controlados por la IANA (Internet Asigned Numbers Authority) o Agencia de Asignación de Números de Internet.

- Puertos registrados

Oscilan entre 1024 y 49151. Pueden ser usados de manera temporal por los clientes, pero también pueden representar servicios registrados por un tercero.

- Puertos dinámicos o privados

Oscilan entre 49152 y 65535. Pueden también ser usados por el cliente, pero se utilizan con menos frecuencia.

A continuación se presenta una lista de los puertos más utilizados:

Puerto	Servicio o aplicación
21	FTP (File Transfer Protocol)
22	SSH (Secure Shell)
23	Telnet
25	SMTP (Simple Mail Transfer Protocol)
53	DNS (Domain Name System)
80	HTTP (Hypertext Transfer Protocol)

[4] File Transfer Protocol - Protocolo de transferencia de archivos.
[5] Hypertext Transfer Protocol - Protocolo de transferencia de hipertexto.
[6] Simple Mail Transfer Protocol - Protocolo simple de transferencia de correo.
[7] Domain Name System - Sistema de nombre de dominio.
[8] Transmission Control Protocol - Protocolo de control de transmisión.
[9] User Datagram Protocol- Protocolo de datagrama de usuario

110	POP3 (Post Office Protocol)
161	SNMP (Simple Network Management Protocol)
443	HTTPS (Hypertext Transfer Protocol Secure)
3306	MySQL

Tabla 1.1 Puertos más conocidos. La lista completa está en la página de la IANA

1.1.7 Protocolos

- **Protocolo de datagramas de usuario (UDP)**

El protocolo de datagramas de usuario (User Datagram Protocol, por sus siglas en inglés), es un protocolo a nivel de transporte que va de extremo a extremo y que añade sólo direcciones de puertos, control de errores mediante sumas de comprobación y la información de longitud de datos del nivel superior. El paquete producido por el protocolo UDP se denomina datagrama de usuario (figura 1.8).

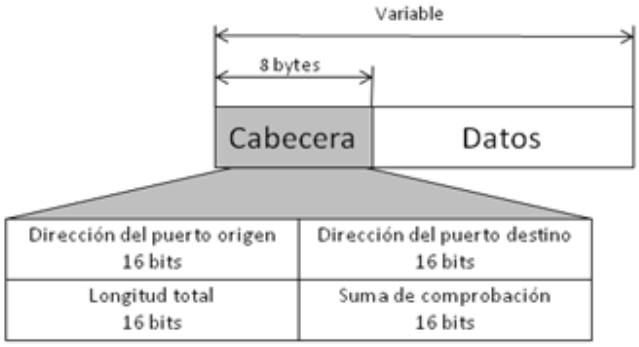


Figura 1.8 Formato de un datagrama UDP

- Dirección del puerto origen. Es la dirección del programa de aplicación que ha creado el mensaje.

- Dirección del puerto destino. Es la dirección del programa de aplicación que recibirá el mensaje.
- Longitud total. Este campo define la longitud total del datagrama de usuario en bytes.

- Suma de comprobación. Esta suma de comprobación es un campo de 16 bits utilizado para la detección de errores.

UDP proporciona sólo las funciones básicas necesarias para la entrega extremo a extremo de una transmisión. No ofrece funciones de secuenciación ni de reordenación y no puede especificar el paquete dañado cuando se informa de un error.

- **Protocolo de control de transmisión (TCP)**

El protocolo de control de transmisión (Transmission Control Protocol por sus siglas en inglés.) Este protocolo proporciona servicios completos de transporte a las aplicaciones. TCP es un protocolo de transporte puerto a puerto que ofrece un flujo fiable, es decir, que está orientado a conexión: se debe de establecer una conexión entre ambos extremos de la transmisión antes de poder enviar datos. TCP comienza cada transmisión informando al receptor de que hay datagramas en camino y finaliza cada transmisión con una terminación de conexión. De esta forma, el receptor conoce la transmisión entera en lugar de un único paquete. TCP es responsable de la entrega fiable del flujo entero de bits contenido en el mensaje inicialmente generado por la aplicación emisora. La fiabilidad se asegura mediante la detección de errores y la retransmisión de las tramas con errores; todos los segmentos deben ser recibidos y confirmados antes de que la transmisión se considere completa.

En el extremo emisor de cada transmisor, TCP divide las transmisiones largas en unidades de datos más pequeñas y empaqueta cada una de ellas en una trama denominada segmento. Cada segmento incluye un número de secuencia para la posterior reordenación de los segmentos en el receptor, junto con un número identificador de confirmación y un campo que indica el tamaño de la ventana deslizante utilizada en las confirmaciones. Los segmentos se transportan por la red dentro de datagramas IP. En el extremo receptor, TCP captura cada datagrama y reordena la transmisión de acuerdo a los números de secuencia.

Debido a los servicios ofrecidos por TCP requiere que la cabecera del segmento sea amplia (figura 1.9). Debido a que UDP utiliza un tamaño de trama más pequeño es mucho más rápido que TCP, pero menos fiable.

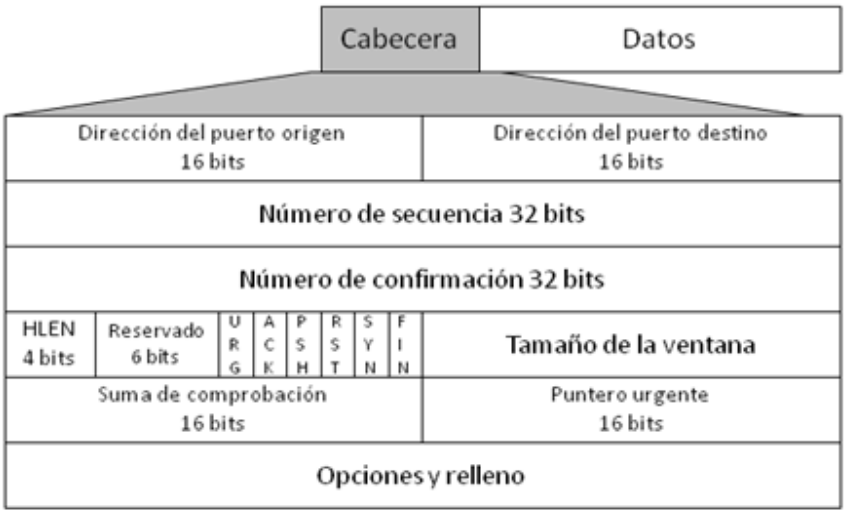


Figura 1.9 Formato del segmento de TCP

- Dirección del puerto origen. Esta dirección define el programa de aplicación de la computadora origen.

- Dirección del puerto destino. Este campo define el programa de aplicación de la computadora destino.

- Número de secuencia. Un flujo de datos del programa de aplicación se puede dividir en dos o más segmentos TCP. El campo con el número de secuencia indica la posición de los datos en el flujo de datos original.

- Número de confirmación. Se utiliza para confirmar la recepción de datos desde otro dispositivo que participa en la comunicación.

- Longitud de la cabecera. Este campo de cuatro bits indica el número de palabras de 32 bits de la cabecera TCP. Los cuatro bits pueden definir hasta 15 que al multiplicarlo por cuatro se obtiene el número total de bytes de la cabecera.

- Reservado. Este campo de seis bits está reservado para uso futuro.

- Control. Cada bit del campo de control de seis bits funciona de forma individual e independiente. Un bit puede definir el uso de un segmento o servir como una comprobación de la validez de otros campos. El bit urgente valida el campo de

puntero urgente. El bit ACK valida el campo con el número de confirmación. El bit PSH se utiliza para informar al emisor de que se necesita un mayor ancho de banda. El bit RST se utiliza para reiniciar la conexión cuando hay confusión en los números de secuencia en tres tipos de segmentos: petición de conexión, confirmación de conexión y la recepción de confirmación. El bit FIN se utiliza en la terminación de la conexión.

- Tamaño de la ventana. Este campo de 16 bits define el tamaño de la ventana deslizante.

- Suma de comprobación. Este campo de 16 bits se utiliza para la detección de errores.

- Puntero urgente. Este es el último campo requerido en la cabecera. Su valor es válido sólo si el bit URG se encuentra activado. En este caso, el emisor está informando al receptor que hay datos urgentes en la porción de datos del segmento.

- Opciones y relleno. El resto de la cabecera TCP define campos opcionales. Se utilizan para evitar información adicional al receptor o para alineamiento.

1.1.8 Servicio de Red

Un servicio es un conjunto de actividades que buscan satisfacer una necesidad. Se puede definir como un bien no material. En redes de datos la definición varía ya que la finalidad de una red es que los usuarios de los sistemas informáticos dentro de una organización, puedan hacer uso de los mismos, mejorando el rendimiento de la organización. Así se pueden obtener ciertas ventajas en el entorno de trabajo como son, mayor facilidad de comunicación, mejora de la competitividad, mejora de la dinámica de grupo, reducción del presupuesto para proceso de datos, reducción de los costos de proceso por usuario, mejoras en la administración de los programas, mejoras en la integridad de los datos, mejora en los tiempos de respuesta, flexibilidad en el proceso de datos, mayor variedad de programas, mayor facilidad de uso y mejor seguridad.

Para que todo esto sea posible, la red debe prestar una serie de servicios a sus usuarios, como son:

- Acceso. Los servicios de acceso a la red comprenden tanto la verificación de la identidad del usuario para determinar cuáles son los recursos de la misma que

puede utilizar, como servicios para permitir la conexión de usuarios de la red desde lugares remotos. Para el control de acceso, el usuario debe identificarse conectándose con un servidor en el cual se autentifica por medio de un nombre de usuario y una clave de acceso. Si ambos son correctos, el usuario puede conectarse a la red.

- Archivos. El servicio de archivos consiste en ofrecer a la red grandes capacidades de almacenamiento para descargar o eliminar los discos de las estaciones. Esto permite almacenar tanto aplicaciones como datos en el servidor, reduciendo los requerimientos de las estaciones. Los archivos deben ser cargados en las estaciones para su uso.

- Correo. El correo electrónico es la aplicación de red de las más utilizadas. Permite claras mejoras en la comunicación frente a otros sistemas. Además, tiene un costo mucho menor para transmitir iguales cantidades de información. Frente al correo convencional tiene la clara ventaja de la rapidez.

- Otros. Las redes más modernas, con grandes capacidades de transmisión, permiten transferir contenidos diferentes de los datos, como pueden ser imágenes o sonidos. Esto permite aplicaciones como: estaciones integradas (voz y datos), telefonía integrada, videoconferencias, etc.

1.2 Monitorización

1.2.1 ¿Qué es?

Por definición, la monitorización es la acción y efecto de monitorizar, que según el Diccionario de la Lengua Española en su vigésima segunda edición, significa observar, mediante aparatos especiales, el curso de uno o varios parámetros fisiológicos o de otra naturaleza, para detectar posibles anomalías. Por lo que, la monitorización de red, es la acción que nos permite verificar sistemáticamente el desempeño y la disponibilidad de los dispositivos críticos dentro de la red, a través de la identificación y detección de posibles problemas.

1.2.2 Tipos de monitorización

Existen varias formas de monitorización. Las dos más comunes son la monitorización pasiva y la activa. Ambas tienen ventajas y desventajas.

- **Monitorización pasiva**

La monitorización pasiva solo está al tanto de lo que pasa sin modificar ningún parámetro u objeto. Usa dispositivos para ver el tráfico que está circulando a través de la red, estos dispositivos pueden ser *sniffers*[10] o en su defecto sistemas incluidos en los *switches* y *ruteadores*. Ejemplos de estos sistemas son la monitorización remota (RMON) y el protocolo simple de administración de red. Una de las características de la monitorización pasiva es que no incrementa el tráfico en la red para poder realizar las lecturas. Se puede capturar el tráfico teniendo un puerto espejo (*mirror*) en un dispositivo de red o un dispositivo intermedio que esté capturando el tráfico.

- **Monitorización activa**

La monitorización activa tiene la capacidad de inyectar paquetes de prueba dentro de la red o enviar paquetes a servidores con determinadas aplicaciones, siguiéndolos para medir los tiempos de respuesta. Por lo tanto, genera tráfico extra lo suficiente para recabar datos precisos. Este tipo de monitorización permite el control explícito en la generación de paquetes para realizar las mediciones, como el control en la naturaleza de generar tráfico, las técnicas de muestreo, frecuencia, tamaño de los paquetes entre otras.

- **Alarmas**

Una alarma es un aviso o señal de cualquier tipo, que advierte de la proximidad de un peligro. Son consideradas como eventos fuera de lo común y por lo tanto, necesitan atención inmediata para mitigar la posible falla detectada. Las alarmas son activadas cuando un parámetro ha alcanzado cierto nivel o se tiene un comportamiento fuera de lo normal.

1.2.3 ¿Qué es lo que se debe monitorizar?

Existen personas que se refieren a la "salud de la red" cuando hablan del rendimiento de la red y su capacidad para dar servicio. Existen funciones críticas dentro de los dispositivos de red que necesitan ser monitorizados constantemente y las alarmas que se lleguen a presentar deben ser atendidas tan pronto como sea posible cuando un evento ocurra. Algunos de los parámetros más comunes son la utilización de ancho de banda, el consumo de CPU, consumo de memoria, el estado físico de las conexiones, el tipo de tráfico que manejan los *ruteadores, switches, hubs,*, *firewalls* y los servicios de web, correo, base de datos entre otros. También se debe estar pendientes de las bitácoras de conexión al sistema y configuración de los sistemas, ya que aquí se presenta

[10] Programa para monitorizar y analizar el trafico en una red de computadoras, detectando los cuellos de botellas y problemas que existan en ella.

información valiosa cuando un evento sucede. Estos parámetros deben de ser monitorizados muy de cerca para detectar posibles cambios y tendencias que afecten al usuario final.

1.3 Desempeño óptimo del sistema

Para que un sistema se considere óptimo tiene que cumplir con todas las especificaciones del usuario. En caso de que el sistema sea un servidor se puede decir que el sistema es óptimo cuando está bien administrado, seguro, con respuestas rápidas a los clientes. Para ello se tienen que medir de alguna forma, parámetros clave del sistema, para que no alcancen niveles que generen un mal comportamiento.

1.3.1 Funcionamiento Óptimo para servidores Windows

Dentro de la página de soporte de Microsoft se conjuntan ciertos parámetros clave que se deben de tener en cuenta para que los servidores Windows se encuentren en óptimas condiciones, entre los más relevantes se tienen:

Usar hardware de 64 bits, memoria suficiente y tarjetas de red rápidas

- El uso del procesador de un servidor, debe mantener una carga del 60 por ciento aproximadamente durante las horas de máxima actividad. Este porcentaje admite períodos de carga muy elevada. Si el uso del procesador está por encima del 75 por ciento de manera continuada, el rendimiento del procesador se considera un cuello de botella.

- Asignar al menos 2 gigabytes (GB) de RAM por cada procesador para los servidores cliente web y los servidores de aplicaciones.

- Asignar al menos 4 GB de RAM por cada procesador para los servidores de base de datos.
- Usar tarjetas de red de gigabit para todas las funciones de servidor.

- Para los servidores cliente web y los servidores de aplicaciones, utilizar tarjetas NIC duales en los entornos de producción: una para los usuarios y otra para la comunicación de SQL Server.

Mantener las bases de datos limpias y en buen estado.
Anticipar el crecimiento de todas las bases de datos y registros si puede. Asegurarse de supervisar los tamaños para que no se quede sin espacio en disco.

- No sobrecargar los servidores de base de datos con demasiada información.
- No almacenar más de 50 bases de datos en una única instancia física de SQL Server.
- Limitar las bases de datos de contenido a 100 GB.
- Memoria usada: menos del 70%
- Espacio libre en disco: más del 25% (Microsoft, Procedimientos recomendados para un funcionamiento óptimo, 2008)

1.3.2 Funcionamiento Óptimo para servidores Linux Fedora 12

La cantidad mínima de memoria swap disponible nunca debe de ser menor que la cantidad real de memoria física mas 256 Kb.

- RAM mínimo para modo texto: 256 MB
- Mínimo de RAM para gráficos: 384 MB
- RAM recomendado para gráficos: 512 MB

El tamaño en disco duro dependerá de los paquetes instalados. Se necesita espacio adicional para los datos del usuario, y se debe reservar al menos un 5% de espacio libre para el funcionamiento adecuado del sistema, ya que como sabemos estos sistemas colapsan cuando alguna partición llega al 100%. (RedHat, Requerimientos de hardware, 2009)

1.3.3 Ancho de banda en una red Ethernet

Se puede encontrar en muchas bibliografías que la tecnología Ethernet se satura al llegar al 37% de su utilización. Sin embargo, esto no es correcto. De acuerdo con Charles Spurgeon, en su libro Ethernet The Definitive Guide, el 37% fue reportado por primera vez por Bob Metcalfe y David Boggs en 1976. En un documento escrito por ellos, describen el desarrollo y operación de la primera Ethernet, la cual operaba a 3 Mbps. En su modelo asumían una transmisión constante de 250 computadoras. En su modelo, el sistema alcanzó una saturación de cerca del 36.8 por ciento en la utilización del canal. Los autores advirtieron que éste era un modelo simplificado y que no tenía ninguna relación con el funcionamiento normal de las redes. Este mito del desempeño de Ethernet persistió por muchos años, debido quizás a que nadie entendía que fue una prueba de la peor carga de tráfico de red que pudieran imaginar.

Las nuevas pruebas realizadas por estos mismos autores muestran lo contrario. Realizaron pruebas con 24 computadoras mandando un flujo constante de datos a 10 Mbps en un canal Ethernet usando diferentes tamaños de frames. Para frames enviados entre pocas computadoras, la utilización del canal fue mayor a 9.5 Mbps y para frames más grandes resultó cercano al 10 Mbps, es decir casi del 100 por ciento. No se registra en ningún caso una saturación en un punto arbitrario de 37%. Los experimentos demostraron que la tecnología Ethernet puede transportar grandes cantidades de tráfico entre un conjunto de equipos manteniéndose estable y sin mayores problemas. (Figura 1.10)

Estos autores recomiendan no tener muchas computadoras en un mismo dominio de colisión, es preferible usar switches y routers para segmentar la red en múltiples dominios de colisión. También recomiendan evitar combinar el uso de aplicaciones en tiempo real con aplicaciones para la transferencia de archivos ya que provocan retardos en estas aplicaciones.

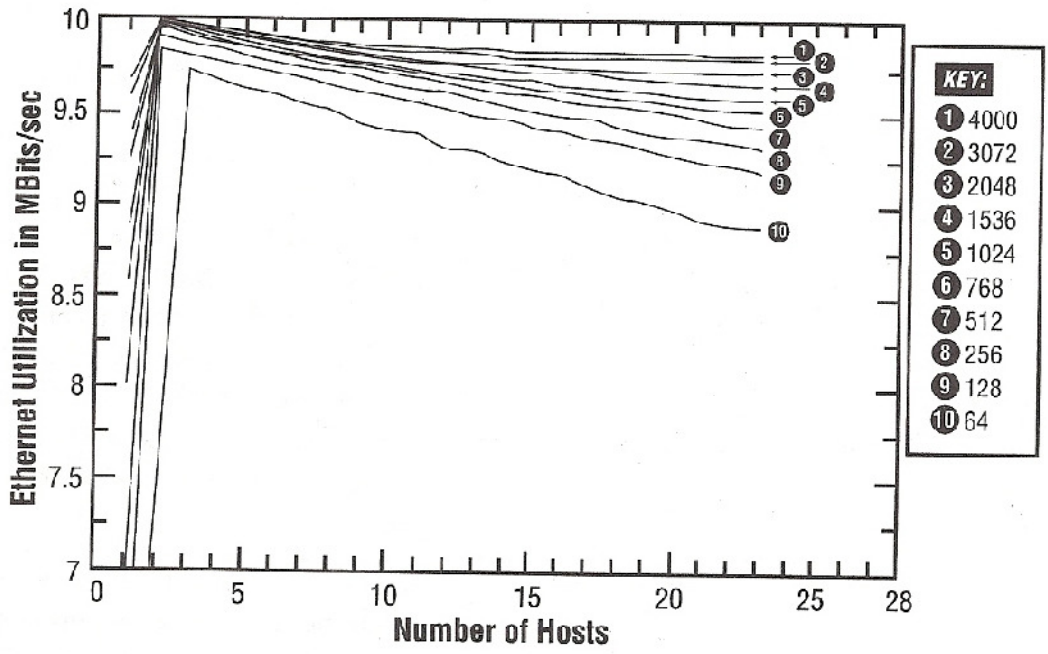


Figura 1.10 Experimento de utilización de Ethernet

1.4 SNMP ("*Simple Network Management Protocol*")

1.4.1 ¿Qué es SNMP?

El Protocolo Simple de Administración de Red (Simple Network Management Protocol) es un protocolo de internet para el manejo de dispositivos dentro de redes IP y pertenece a la capa de aplicación (MAURO, 2001). Existen muchos tipos de dispositivos que soportan SNMP, como por ejemplo están los *ruteadores, switches, servidores*,

estaciones de trabajo, impresoras, así como UPS (*"Uninterruptible Power Supply: Suministro de Energía Ininterrumpible'*).

Se puede usar SNMP para monitorizar el estado de los *ruteadores*, servidores y otros componentes de red pero también se usa para controlar dispositivos de red o tomar acciones de forma automática en caso de que se presente un problema. Se puede monitorizar información, ésta puede ser simple, como la cantidad de tráfico que entra o sale en una interface, o puede ser algo más complejo como la temperatura del aire dentro de un ruteador. También puede verificar la velocidad a la cual opera una interfaz de red.

1.4.2 Funcionamiento

Una red administrada a través de SNMP consiste de tres componentes:

- Dispositivos administrados
- Agentes
- Sistemas administradores de red NMS's (por sus siglas en inglés Network Management Stations).

Un dispositivo administrado, es un nodo de red que contiene un agente SNMP y reside en una red administrada. Estos recogen y almacenan información de administración, la cual ponen a disposición de los NMS's usando SNMP. Los dispositivos administrados pueden ser ruteadores, servidores de acceso, switches, bridges, hubs, computadoras o impresoras.

Un agente es un módulo de software de administración de red que reside en un dispositivo administrado. Un agente posee un conocimiento local de información de administración (memoria libre, número de paquetes IP recibidos, rutas, etcétera), la cual es traducida a un formato compatible con SNMP y organizada en jerarquías.

Un NMS ejecuta aplicaciones que supervisan y controlan a los dispositivos administrados. Los NMS's proporcionan el volumen de recursos de procesamiento y memoria requeridos para la administración de la red. Uno o más NMS's deben existir en cualquier red administrada (MAURO, 2001).

1.4.3 Comunidades

El servicio SNMP ofrece una forma rudimentaria de seguridad mediante la utilización de nombres de comunidad y capturas de autenticación. Es posible restringir las

comunicaciones SNMP del agente y permitirle que se comunique únicamente con una lista definida de sistemas de administración SNMP. Los nombres de comunidad permiten autenticar los mensajes SNMP y, por lo tanto, proporcionan un esquema de seguridad rudimentario para el servicio SNMP. No existe ninguna relación entre los nombres de comunidad y los nombres de dominio o de grupo de trabajo. Un nombre de comunidad puede considerarse una contraseña compartida por las consolas de administración de SNMP y los equipos administrados. Éstos pueden ser configurados con el acceso de sólo lectura o lectura y escritura.

1.4.4 SNMP y UDP

El SNMP usa el protocolo de datagramas de usuario como protocolo de transporte para intercambiar datos entre los sistemas administradores y los agentes. UDP fue escogido sobre el protocolo TCP debido a que puede enviar mensajes sin establecer una conexión con el receptor. Esta característica de UDP lo hace poco confiable ya que no se sabe si hay pérdida de datagramas durante el envío de los mismos. Depende de la aplicación SNMP determinar si los datagramas están perdidos y volver a transmitirlos si así se requiere. Esto es normalmente acompañado con un tiempo máximo. El NMS envía una petición UDP a un agente y espera por la respuesta. La cantidad de tiempo que el NMS espera, depende de cómo esté configurado. Si se alcanza el tiempo máximo de espera y el NMS no obtiene información del agente, se asume que el paquete se perdió y retransmite nuevamente la petición.

1.4.5 Comandos básicos

Los dispositivos administrados son supervisados y controlados usando cuatro comandos SNMP básicos: lectura, escritura, notificación y operaciones transversales.

El comando de lectura es usado por un NMS para supervisar elementos de red. El NMS examina diferentes variables que son mantenidas por los dispositivos administrados.

El comando de escritura es usado por un NMS para controlar elementos de red. El NMS cambia los valores de las variables almacenadas dentro de los dispositivos administrados.

El comando de notificación es usado por los dispositivos administrados para reportar eventos en forma asíncrona a un NMS. Cuando cierto tipo de evento ocurre, un dispositivo administrado envía una notificación al NMS.

Las operaciones transversales son usadas por el NMS para determinar qué variables soporta un dispositivo administrado y para recoger secuencialmente información en tablas de variables, como por ejemplo, una tabla de rutas.

- **Operaciones SNMP**

El protocolo de unidad de datos (PDU por sus siglas en inglés, Protocol Data Unit) es el formato de mensaje que los NMS's y los agentes usan para enviar y recibir información. Hay un formato estándar para cada una de las operaciones:

- **Operación get**

La petición get es iniciada por el NMS, el cual envía la petición al agente. El agente recibe la petición y la procesa. Algunos dispositivos sometidos a mucha carga, como lo son los ruteadores, pueden no ser capaces de responder a la petición y tendrán que rechazarla. Si el agente es capaz de recolectar la información solicitada, éste envía de vuelta un get-response al NMS, donde es procesada (figura 1.11).

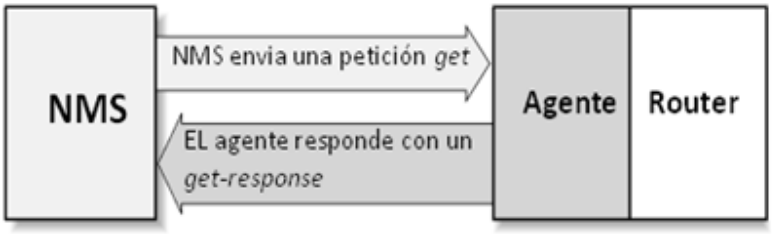


Figura 1.11 Secuencia de petición get

- **Operación get-next**

Este mensaje es usado para recorrer una tabla de objetos. Una vez que se ha usado un mensaje get para recoger el valor de un objeto, puede ser utilizado el mensaje get-next para repetir la operación con el siguiente objeto de la tabla. Siempre el resultado de la operación anterior será utilizado para la nueva consulta. De esta forma un NMS puede recorrer una tabla de longitud variable hasta que haya extraído toda la información para cada fila existente.

- **Operación get-bulk**

Este mensaje es usado por un NMS, que utiliza la versión 2 del protocolo SNMP normalmente cuando es requerida una larga transmisión de datos, tal como la recuperación de largas tablas. En este sentido es similar a la operación get-next usado en la versión 1 del protocolo, sin embargo, get-bulk es un mensaje que implica un método

mucho más rápido y eficiente, ya que a través de un solo mensaje es posible solicitar la totalidad de la tabla.

- **Operación set**

El comando set es usado para cambiar el valor de un objeto administrado o para crear una nueva fila en una tabla. Objetos que están definidos en la MIB como lectura y escritura o sólo escritura pueden ser alterados o creados usando este comando. (Figura 1.12) (Case, Fedor, Schoffstall, & Davin, 1990)

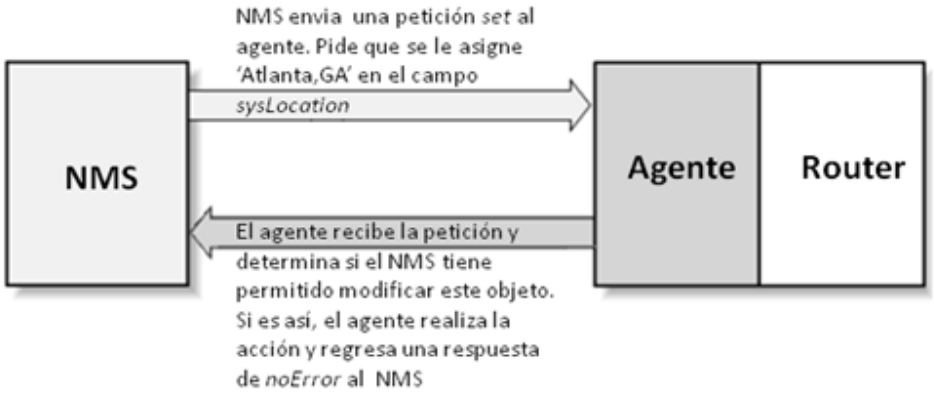


Figura 1.12 Secuencia de petición set

- **Operación trap**

Un trap es la forma de comunicase el agente con el NMS para decirle que algo malo ha sucedido. El trap se origina desde el agente y es enviado al destino, que normalmente es la dirección IP del NMS (figura 1.13). Algunas situaciones que un trap podría reportar son:
Una interfaz de red en un dispositivo, donde el agente está activo, quedó inactivo.
Una interfaz de red en un dispositivo, donde el agente esta activo, se activó nuevamente.
El ventilador de un router o switch se apagó.

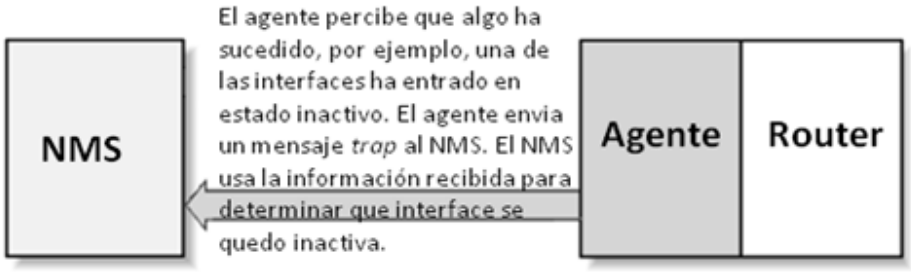


Figura 1.13 Generación de la operación trap

1.4.6 MIB

Una MIB (Management Information Base) es una base de información de administración, lo cual quiere decir que es una colección de información que está organizada jerárquicamente (figura 1.14). Las MIB's son accedidas usando un protocolo de administración de red, como por ejemplo, SNMP.

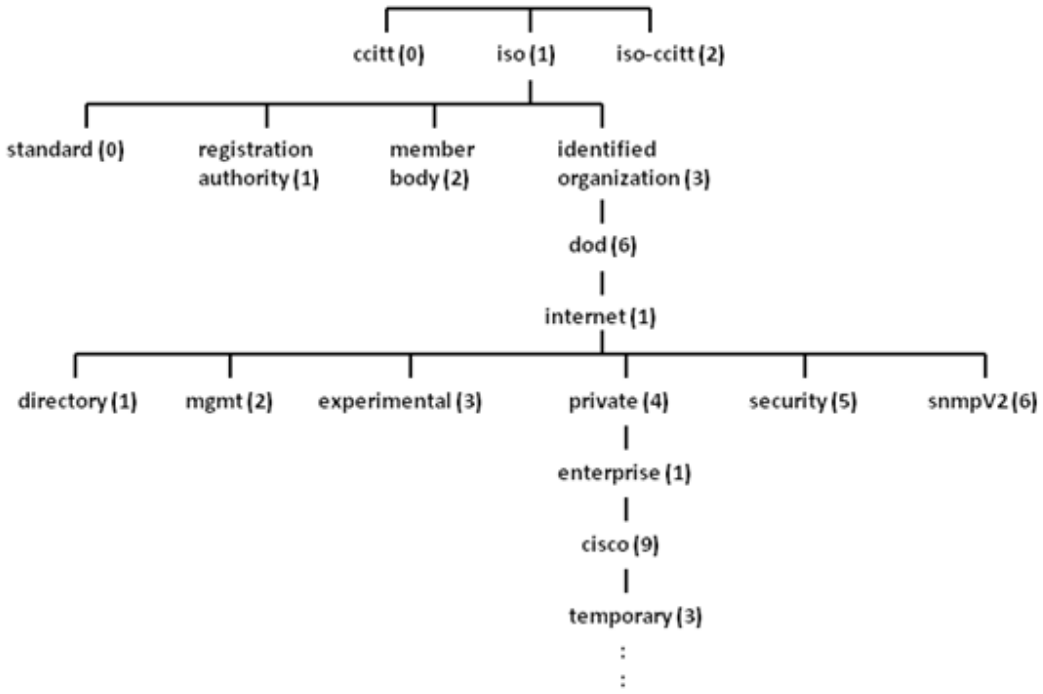


Figura 1.14 Árbol jerárquico MIB

Un objeto administrado (algunas veces llamado objeto MIB, objeto, o MIB) un número de características específicas de un dispositivo administrado. Los objetos administrados están compuestos de una o más instancias de objeto, que son esencialmente variables (Rose & McCloghrie, 1991).

Existen dos tipos de objetos administrados: escalares y tabulares. Los objetos escalares definen una simple instancia de objeto. Los objetos tabulares definen múltiples instancias de objeto relacionadas que están agrupadas conjuntamente en tablas MIB.

Un ejemplo de un objeto administrado es *atInput*, que es un objeto escalar que contiene una simple instancia de objeto, el valor entero que indica el número total de paquetes AppleTalk de entrada sobre una interfaz de un router.

Un identificador de objeto (object ID) únicamente identifica un objeto administrado en la jerarquía MIB. La jerarquía MIB puede ser representada como un árbol con una raíz anónima y los niveles, que son asignados por diferentes organizaciones.

El corazón del árbol MIB se encuentra compuesto de varios grupos de objetos, los cuales en su conjunto son llamados mib-2. Los grupos son los siguientes:

- System: de este nodo cuelgan objetos que proporcionan información genérica del sistema gestionado.

- Interfaces: En este grupo está la información de las interfaces de red presentes en el sistema. Incorpora estadísticas de los eventos ocurridos en el mismo.

- At (address translation o traducción de direcciones): este nodo es obsoleto, pero se mantiene para preservar la compatibilidad con la MIB-I. En él se almacenan las direcciones de nivel de enlace correspondientes a una dirección IP.

- Ip: En este grupo se almacena la información relativa a la capa IP, tanto de configuración como de estadísticas.

- Icmp: En este nodo se almacenan contadores de los paquetes ICMP entrantes y salientes.

- Tcp: En este grupo está la información relativa a la configuración, estadísticas y estado actual del protocolo TCP.

- Udp: En este nodo está la información relativa a la configuración, estadísticas del protocolo UDP.

- Egp: Aquí está agrupada la información relativa a la configuración y operación del protocolo EGP.

- Transmission: De este nodo cuelgan grupos referidos a las distintas tecnologías del nivel de enlace implementadas en las interfaces de red del sistema gestionado.

- **Mensajes SNMP**

Para realizar las operaciones básicas de administración anteriormente nombradas, el protocolo SNMP utiliza un servicio no orientado a la conexión (UDP) para enviar un pequeño grupo de mensajes (PDUs) entre los administradores y agentes. La utilización de un mecanismo de este tipo asegura que las tareas de administración de red no afectarán

al rendimiento global de la misma, ya que se evita la utilización de mecanismos de control y recuperación como los de un servicio orientado a la conexión, por ejemplo TCP.

Los puertos comúnmente utilizados para SNMP son los siguientes:
161-snmp
162-snmp-trap

1.4.7 Versiones existentes

- SNMP Versión 1 (SNMPv1). Es la versión estándar del protocolo SNMP. Como ya se había mencionado anteriormente, la seguridad de SNMPv1 está basada en comunidades, que no son más que contraseñas: cadenas en texto plano que permiten a cualquier aplicación basada en SNMP tener acceso a la información de esos dispositivos con tan sólo poseer la cadena.

- SNMP Versión 2 (SNMPv2).Tiene características en común con la versión 1 pero ofrece mejoras, como por ejemplo, operaciones adicionales. Utiliza el mismo modelo administrativo que la primera versión del protocolo SNMP, y como tal no incluye mecanismos de seguridad. Las únicas mejoras introducidas en la nueva versión consisten en una mayor flexibilidad de los mecanismos de control de acceso, ya que se permite la definición de políticas de acceso consistentes en asociar un nombre de comunidad con un perfil de comunidad formado por una vista MIB y unos derechos de acceso a dicha vista (read-only o read-write).

- SNMP Versión 3 (SNMPv3). Éste agrega soporte para una autenticación fuerte y comunicación privada entre entidades administradas. Es un protocolo de manejo de red interoperable, que proporciona seguridad de acceso a los dispositivos por medio de una combinación de autenticación y cifrado de paquetes que trafican por la red.

1.4.8 Ventajas y Desventajas

Ventajas

- Protocolo estándar
- Diseño simple
- Es muy usado
- Flexible, se puede adaptar a las necesidades de gestión de cualquier elemento

- Es la única manera conocida de gestionar una red grande heterogénea (como la propia Internet)
- Se tiene experiencia usando SNMP
- Con la versión 3 se resuelven problemas graves de seguridad
- Nuevos dispositivos saldrán al mercado con soporte para SNMP.

Desventajas

- Bajo nivel de seguridad
- Difícil de implementar
- El número de administradores y programadores que realmente entienden cómo funciona SNMP es muy limitado.

Al final, el administrador es el que decide si usa o no el protocolo SNMP para administrar sus dispositivos de red y qué versión deberá elegir, dependiendo del nivel de seguridad que le sea exigido.

1.5 El sistema operativo Linux

1.5.1 Historia

Linux, es un sistema operativo. Es una implementación de libre distribución UNIX para computadoras personales (PC), servidores y estaciones de trabajo.

Es la denominación de un sistema operativo tipo-Unix y el nombre de un núcleo.
Es uno de los paradigmas más prominentes del software libre y del desarrollo del código abierto, cuyo código fuente está disponible públicamente, para que cualquier persona pueda libremente usarlo, estudiarlo, redistribuirlo y, con los conocimientos informáticos adecuados, modificarlo.

Linux es usado como sistema operativo en una amplia variedad de plataformas de hardware y computadores, incluyendo las computadoras de escritorio (PCs x86 y x86-64, y Macintosh y PowerPC), servidores, supercomputadoras, mainframes, y dispositivos empotrados así como teléfonos celulares.
En 1983 Richard Stallman fundó el proyecto GNU, con el fin de crear sistemas operativos parecidos a UNIX y compatibles con POSIX. Dos años más tarde creó la "Fundación del Software Libre" y escribió la GNU General Public License para posibilitar el software libre en el sistema de copyright (Von Hagen & Jones, 2006).

El software GNU se extendía muy de prisa y en poco tiempo una multitud de programas fueron escritos, de manera que ya a principios de 1990 había bastantes software GNU como para hacer un sistema operativo propio, pero faltaba el Kernel[11].

A principios de los años 1990, no había un sistema operativo libre completo. A pesar de que el proyecto GNU era desarrollado constantemente, no disponía sin embargo de ningún buen Kernel basado en UNIX, por el contrario, era un número de proyectos de software libres que podían ser traducidos en las variantes UNIX mediante el compilador de GNU.

El 5 de octubre de 1991, Linus anunció la primera versión "Oficial" de Linux, - versión 0.02.

Con esta versión Linus pudo ejecutar Bash (GNU Bourne Again Shell) y gcc (Compilador GNU de C) pero no hubo una mejoría en su funcionamiento. En este estado de desarrollo ni se pensaba en los términos soporte, documentación, distribución. Después de la versión 0.03, Linus saltó en la numeración hasta la 0.10, más programadores a lo largo y ancho del internet empezaron a trabajar en el proyecto y después de revisiones, Linus incrementó el número de versión hasta la 0.95 (marzo 1992). En Diciembre de 1993 el núcleo del sistema estaba en la versión 0.99 y la versión 1.0, llegó el 14 de marzo de 1994.

1.5.2 Características generales

- Multitarea. Linux es un sistema operativo que permite ejecutar varios procesos de manera simultánea. Como punto de referencia, MS-DOS es un sistema monotarea, y todos los sistemas Windows son multitarea.

- Multiusuario. Linux permite tener varios usuarios trabajando en la misma máquina y al mismo tiempo. En la actualidad el acceso se hace mediante otras máquinas. Por ejemplo, cuando una persona revisa su correo en un servidor, al mismo tiempo que otras, estamos siendo usuarios conectados en el servidor a la vez, cada quien viendo sus archivos y realizando sus tareas.

- Multiplataforma. Linux funciona en varias arquitecturas de procesadores, como Intel, Sparc, PowerPC, a diferencia de otros sistemas operativos, que sólo funcionan en una arquitectura determinada.
- Soporte de tecnologías venideras (multiprocesador). Linux en su versión de kernel más reciente soporta hasta 16 procesadores. Ya existen las versiones de 64 bits para Linux, manejo de grandes volúmenes de memoria RAM, etc.

[11] Es un software que actúa de sistema operativo.

- Memoria virtual. Usando paginación a disco, a una partición o un archivo en el sistema de archivos, o ambos, con la posibilidad de añadir más áreas de intercambio sobre la marcha. Un total de 16 zonas de intercambio de 128 Mb de tamaño máximo pueden ser usadas en un momento dado con un límite teórico de 2 Gb para intercambio. Este límite se puede aumentar fácilmente con el cambio y compilación de unas cuantas líneas en el código.

- Shells programables. El shell es un programa que se encuentra en el sistema que interpreta los comandos que son la interfaz con el sistema operativo y facilitan su control, el shell es el encargado de comunicaciones con el kernel.

- Consolas virtuales múltiples. Varias sesiones a través de la consola entre las que se puede cambiar con las combinaciones adecuadas de teclas (independiente del hardware de video). Se crean dinámicamente y es posible hasta tener 64.

1.5.3 Grupos y usuarios

Los usuarios pueden ser gente real, es decir, cuentas ligadas a un usuario físico en particular o cuentas que existen para ser usadas por aplicaciones específicas.

Los grupos son siempre expresiones lógicas de organización, reuniendo usuarios para un propósito común. Los usuarios dentro de un mismo grupo pueden leer, escribir o ejecutar archivos que pertenecen al grupo.

Cada usuario y grupo tiene un número de identificación único llamado userid (UID) y un groupid (GID) respectivamente. Cuando se crea un archivo se asigna a un usuario y a un grupo. De la misma manera se asignan los permisos de lectura, escritura y ejecución para el propietario del archivo, para el grupo y para cualquier otro usuario en un host. El usuario y el grupo de un archivo particular, así como los permisos en ese archivo, pueden ser cambiados por root o, en la mayoría de los casos, por el creador del archivo.

1.5.4 Sistema de archivos

Todos los archivos del sistema de archivos de Linux se organizan de dos distintos tipos: jerárquica y arborescente.
Se dice que es con forma de árbol, ya que dentro de un directorio pueden existir varios directorios y archivos, de manera recursiva, por lo que el sistema adquiere la forma de un árbol invertido.

Es jerárquico debido a que existen diferentes niveles dentro de él. En el nivel más alto se encuentra un directorio llamado "root", el cual está representado por una / (diagonal). Los demás archivos son “descendientes” de root.

El número de niveles es largamente arbitrario, sin embargo muchos sistemas Linux tienen organizaciones similares.

1.5.5 Análisis de bitácoras

Casi todas las actividades realizadas en un sistema Unix o Linux son susceptibles de ser, en mayor o menor medida, monitorizadas, como por ejemplo las siguientes actividades:

- Horas de acceso de cada usuario al sistema hasta las páginas web más frecuentemente visitadas.
- Intentos fallidos de conexión.
- Los programas ejecutados.
- El tiempo de CPU que cada usuario consume.

Esta facilidad para recoger información tiene ventajas inmediatas para la seguridad: es posible detectar un intento de ataque, así como también detectar usos indebidos de los recursos o actividades "sospechosas".
Sin embargo, existen también desventajas, ya que la gran cantidad de información que potencialmente se registra puede ser aprovechada para crear negaciones de servicio o, más habitualmente, esa cantidad de información puede hacer difícil detectar problemas por el volumen de datos a analizar. Los archivos de registro en Linux la mayoría son simples archivos de texto.

1.5.6 Distribuciones

Una distribución es un sistema operativo basado en Linux configurado por alguien o alguna organización, para un sector específico de usuarios, que incluyen alguna versión del kernel de Linux, las utilidades particulares de la distribución, como puede ser OpenOffice, Gnome, KDE, etc., los shells, y un instalador del sistema operativo, así como un cargador de arranque que facilita tener varios sistemas operativos en un solo sistema.

Algunas distribuciones son:

- **Red Hat Enterprise**

Esta es una distribución que tiene muy buena calidad, contenidos y soporte a los usuarios por parte de la empresa que la distribuye. Es famoso en todo el mundo por los diferentes esfuerzos orientados a apoyar el movimiento del software libre. Es necesario el pago de una licencia de soporte y está enfocada a empresas.

- **Fedora**

Esta es una distribución patrocinada por RedHat y soportada por la comunidad. Fácil de instalar, estable y seguro. Su distribución es gratuita. Gracias a la comunidad de desarrolladores a nivel mundial Fedora busca nuevas tecnologías.

- **Debian**

Otra distribución con muy buena calidad. El proceso de instalación es quizás un poco más complicado, pero sin mayores problemas. Posee gran estabilidad aunque no cuente con las últimas versiones de los programas.

- **Ubuntu**

Una de las distribuciones más usadas por su facilidad de manejo, rápida instalación y lanzamientos regulares. Enfocado principalmente a computadoras personales. Patrocinado por Canonical Ltd. empresa de origen sudafricana.

1.6 Otros sistemas Operativos

1.6.1 Windows Server 2003

Diseñado para medianas y grandes empresas, Windows Server 2003 es el sistema operativo recomendado para los servidores que ejecuten aplicaciones como sistemas de red, mensajería, inventario, bases de datos, sitios web y servidores de archivos e impresión. Proporciona alta confiabilidad, rendimiento y un gran valor empresarial. Fue lanzado el 24 de abril de 2003 y fue una notable actualización al Windows 2000 Server. Está basado en tecnología NT.

- **Características**

Windows Server 2003 permite aumentar el rendimiento y la capacidad de un servidor mediante la adición de procesadores y memoria. Este enfoque para aumentar la capacidad del servidor es conocido como escalado vertical.

Es uno de los sistemas operativos más seguros que Windows haya lanzado al mercado. Protege las redes ante un posible código poco elaborado o malintencionado. Tiene avances en la funcionalidad de seguridad, incluyendo una seguridad mejorada para Servicios de Internet Information Server (IIS), la infraestructura de claves públicas (PKI) y Kerberos, y también la compatibilidad con las tarjetas inteligentes y la biométrica.

Es fácil de usar y administrar. Contiene herramientas que facilitan la configuración de funciones específicas de servidor y de las tareas habituales de administración de servidores. Permite una mayor productividad para administradores de tecnologías de la información y usuarios, a través de funciones avanzadas de administración de sistemas y almacenamiento.

Se proporciona una mayor escalabilidad, con la posibilidad de escalar desde un único procesador hasta sistemas de 32 direcciones. Para tener una mayor disponibilidad, el servicio Microsoft Cluster admite ahora clústeres de hasta ocho nodos y nodos separados geográficamente.

Se tiene una amplia confiabilidad en la infraestructura de red con conexiones seguras compartiendo recursos desde cualquier lugar y a cualquier dispositivo. Incluye mejoras a la versión 6 del protocolo de Internet (IPv6).

Se cuentan con cuatro versiones de este sistema operativo:

- **Standar Edition**

Satisface las necesidades de empresas de todos los tamaños, proporciona solución óptima para compartir archivos e impresoras, conectividad segura a Internet, implementación centralizada de aplicaciones y un entorno de trabajo que conecta eficazmente a empleados, socios y clientes.

- **Enterprise Edition**

La plataforma para las medianas y grandes empresas para implementar aplicaciones de forma segura, así como servicios Web.

- **Web Edition**

Optimizado específicamente para albergar y servir páginas web, manteniendo las funcionalidades esenciales que garantizan la fiabilidad, seguridad y facilidad de gestión características de Windows Server. Es la edición adecuada para implementar servidores web dedicados a bajo costo.

- **Datacenter Edition**

Es el servidor orientado a aplicaciones críticas de negocio que exigen los más altos niveles de escalabilidad y fiabilidad.

1.6.2 Windows Server 2008

Windows Server 2008 es el nombre del sistema operativo para servidores de Microsoft. Es el sucesor de Windows Server 2003, distribuido al público casi cinco años antes. Al igual que Windows Vista, Windows Server 2008 se basa en el núcleo Windows NT 6.0. Una segunda versión, denominada Windows Server 2008 R2, está actualmente en desarrollo.

- **Características**

Hay algunas diferencias con respecto a la arquitectura del nuevo Windows Server 2008, que pueden cambiar drásticamente la manera en que se usa este sistema operativo. Estos cambios afectan a la manera en que se gestiona el sistema hasta el punto de que se puede llegar a controlar el hardware de forma más efectiva, se puede controlar mucho mejor de forma remota y cambiar de forma radical la política de seguridad. Entre las mejoras que se incluyen, están:

- Nuevo proceso de reparación de sistemas NTFS: proceso en segundo plano que repara los archivos dañados.
- Creación de sesiones de usuario en paralelo: reduce tiempos de espera en los Terminal Services y en la creación de sesiones de usuario a gran escala.
- Cierre limpio de Servicios.
- Sistema de archivos SMB2: de 30 a 40 veces más rápido el acceso a los servidores multimedia.
- Address Space Load Randomization (ASLR): protección contra malware en la carga de controladores en memoria.
- Windows Hardware Error Architecture (WHEA): protocolo mejorado y estandarizado de reporte de errores.
- Virtualización de Windows Server: mejoras en el rendimiento de la virtualización.
- PowerShell: inclusión de una consola mejorada con soporte GUI para administración.
- Server Core: el núcleo del sistema se ha renovado con muchas y nuevas mejoras.

- **Ediciones**

La mayoría de las ediciones de Windows Server 2008 están disponibles en x86-64 (64 bits) y x86 (32 bits). Windows Server 2008 para sistemas basados en Itanium soporta procesadores IA-64. La versión IA-64 se ha optimizado para escenarios con altas cargas de trabajo como servidores de bases de datos y aplicaciones de línea de negocios (LOB). Por ende no está optimizado para su uso como servidor de archivos o servidor de medios. Microsoft ha anunciado que Windows Server 2008 será el último sistema operativo para servidores disponible en 32 bits. Windows Server 2008 está disponible en las ediciones que figuran a continuación, similar a Windows Server 2003.

- Windows Server 2008 Standard Edition (x86 y x86-64)
- Windows Server 2008 R2 Todas las Ediciones (Solo 64Bit)
- Windows Server 2008 Enterprise Edition (x86 y x86-64)
- Windows Server 2008 Datacenter Edition (x86 y x86-64)
- Windows HPC Server 2008 (remplaza Windows Compute Cluster Server 2003)
- Windows Web Server 2008 (x86 y x86-64)
- Windows Storage Server 2008 (x86 y x86-64)
- Windows Small Business Server 2008 (Nombre clave "Cougar") (x86-64) para pequeñas empresas
- Windows Essential Business Server 2008 (Nombre clave "Centro") (x86-64) para empresas de tamaño medio
- Windows Server 2008 para sistemas basados en Itanium
- Windows Server 2008 Foundation Server

Server Core está disponible en las ediciones Web, Standard, Enterprise y Datacenter, aunque no es posible usarla en la edición Itanium. Server Core es simplemente una opción de instalación alterna soportada y en sí no es una edición propiamente dicha. Cada arquitectura dispone de un DVD de instalación independiente. Windows Server 2008 Standard Edition está disponible gratuitamente para estudiantes a través del programa Microsoft DreamSpark.

1.6.3 Unix

Los orígenes del sistema UNIX se remontan al desarrollo de un proyecto iniciado en 1968. Este proyecto fue realizado por General Electric los laboratorios Bell de AT&T, llevaron a cabo el desarrollo de un sistema operativo con nuevos conceptos como la multitarea, la gestión de archivos o la interacción con el usuario. El resultado de estas investigaciones se bautizó como MULTICS. El proyecto resultó ser demasiado ambicioso, por lo que no llegó a buen fin y terminó abandonándose.

Posteriormente la idea de este proyecto se vuelve a retomar y conduce al gran desarrollo en 1969 del sistema operativo UNIX. Entre los investigadores destacaban Ken Thompson y Dennis Ritchie. En principio, este sistema operativo recibió el nombre de UNICS, aunque un año después pasa a llamarse UNIX, como se conoce hoy en día.

El código de UNIX estaba inicialmente escrito en lenguaje ensamblador, pero en 1973, Dennis Ritchie llevó a cabo un proyecto para reescribir el código de UNIX en lenguaje C. UNIX se convirtió así en el primer sistema operativo escrito en lenguaje de alto nivel. Con este nuevo enfoque fue posible portar al sistema operativo a otras máquinas sin muchos cambios.

- **Características**

Entre las características de este sistema operativo se encuentran:
Multiusuario, con capacidad de simular multiprocesamiento y procesamiento no interactivo. Está escrito en un lenguaje de alto nivel como lo es el lenguaje C. Dispone de un lenguaje de control programable llamado shell. Ofrece facilidades para la creación de programas y sistemas. Además proporciona el ambiente adecuado para las tareas de diseños de software. Emplea manejo dinámico de memoria por intercambio o paginación. Tiene capacidad de interconexión de procesos. Permite comunicación entre procesos. Emplea un sistema jerárquico de archivos, con facilidades de protección de archivos, cuentas y procesos. Tiene facilidad para redireccionamiento de Entradas/Salidas. Garantiza un alto grado de portabilidad.

El sistema de archivos de Unix está basado en un modelo arborescente y recursivo, en el cual los nodos pueden ser tanto archivos como directorios, y éstos últimos pueden contener a su vez directorios o subdirectorios. Debido a esta filosofía, se maneja al sistema con muy pocas órdenes, que permiten una gran gama de posibilidades. Todo archivo de Unix está controlado por múltiples niveles de protección, que especifican los permisos de acceso al mismo. La diferencia que existe entre un archivo de datos, un programa, un manejador de entrada/salida o una instrucción ejecutable se refleja en estos parámetros, de modo que el sistema operativo adquiere características de coherencia y elegancia que lo distinguen.

Hay que recordar que los sistemas GNU/Linux están basados en los sistemas UNIX, por ello su gran similitud en sus características, sin embargo no son iguales. Una diferencia sustancial es que se requieren licencias para el uso de los sistemas Unix.

- **Implementaciones más importantes**

Solaris de Sun Microsystems. Uno de los sistemas operativos Unix más difundido en el entorno empresarial y conocido por su gran estabilidad. Parte del código fuente de Solaris se ha liberado con licencia de fuentes abiertas (OpenSolaris).

AIX de IBM. El UNIX "propietario" de IBM ha cumplido 20 años de vida en el 2006 y continúa en pleno desarrollo, con una perceptible herencia del mainframe en campos como la virtualización o la RAS de los servidores, heredada de sus "hermanos mayores".
HP-UX de Hewlett-Packard. Este sistema operativo también nació ligado a las computadoras departamentales de este fabricante. También es un sistema operativo estable que continúa en desarrollo.

Mac OS X. Curiosamente sus propios usuarios lo desconocen, se trata de un UNIX completo. Su diferencia marcada es que posee una interfaz gráfica propietaria llamada Aqua, y es principalmente desarrollada en Objective-C en lugar de C o C++.

1.6.4 Mac OS

Es el sistema operativo de Macintosh, desarrollado, comercializado y vendido por Apple Inc. para la línea de computadoras Macintosh. El Mac OS fue el primer sistema operativo con una interfaz gráfica de usuario en tener éxito. Desde 1984 Apple se ha esforzado por desarrollar una serie de sistemas operativos que sean lo más amigable posible, dependiendo de los recursos de hardware que se cuenten. Estos esfuerzos han convergido en el Mac OS. La versión 8 de este sistema salió en 1997 con la característica de controlar múltiples aplicaciones al mismo tiempo con un mejor desempeño de la computadora con el procesador. Su apariencia fue cambiada para tener un aspecto 3D. En 1999 liberan la versión 9 y con esta versión se podían tener varios usuarios con sus propias configuraciones. En 2001 se llegó a la versión 10, que es el sistema operativo que está virtualmente en todas las Mac’s, y fue trabajado por una década para poder obtener la versatilidad y popularidad que tiene ahora.

- **Versiones recientes**

Mac OS X 10.0, conocido como “Cheetah”, fue lanzado el 24 de marzo del 2001. Contenía todas las características de un sistema operativo moderno, protegía la memoria, y así los programas no podían utilizar la información de otros programas, de esta forma el

procesador no se bloqueaba, los drivers de los dispositivos podían ser cargados o descargados si era necesario.

Mac OS X 10.1, o "Puma", fue lanzado al siguiente año de haber salido Mac OS X 10.0 e incorporaba mejor un desempeño. Se corrigieron algunos de los errores y se añadieron actualizaciones. Fueron incluidos drivers adicionales para poder soportar más dispositivos. Mac OS X 10.2 "Jaguar" fue lanzado el 25 de agosto de 2002 y contaba con un nuevo incremento en su rendimiento y más de 150 mejoras, entre éstas estaba el mayor soporte para redes de Microsoft Windows.

Mac OS X 10.3 "Panther" se lanzó el 24 de octubre de 2003. Además de tener un rendimiento mucho mayor, incorporó la mayor actualización en la interfaz de usuario, y muchas mejoras que Jaguar el año anterior. El Finder fue actualizado e incorpora una interfaz metálica y búsqueda rápida.

Mac OS X 10.4 "Tiger" fue lanzado el 29 de abril de 2005 y fue la versión disponible más reciente, contenía más de 200 nuevas mejoras, pero como sucedió con el lanzamiento de Panther, algunas máquinas antiguas no podían soportarlo, en particular, cualquier equipo Apple que no contara con conexión FireWire no podía ser soportado en Tiger.

Mac OS X 10.5 "Leopard" se lanzó el 26 de octubre de 2007. Entre las nuevas características están la posibilidad de poder volver en el tiempo a una versión específica del contenido de una carpeta, del disco duro completo o de un sólo archivo. El nuevo Finder permite abrir varios archivos a la vez con diferentes extensiones y no hay necesidad de abrir el programa, incluso los usuarios pueden hacer búsquedas en otras Mac conectadas en red.

Mac OS X v10.6 "Snow Leopard" es la séptima y más reciente versión del sistema operativo de Apple Mac OS X, fue lanzado públicamente el 28 de agosto de 2009.Es más rápido que la versión 10.5 Leopard y está disponible en el Apple Store y en distribuidores autorizados Apple (Falla, 2007).

Las mejoras incluidas en esta versión son:

- Ajuste automático de la zona horaria.
- Mejoras en la selección de texto en PDF.
- Tiempos de instalación menores y menor uso de espacio de disco duro
- Soporte para 16 TB[12] teóricos de RAM a través de mayor desarrollo de tecnologías de núcleos de 64 bits.

[12] Un terabyte es una unidad de medida de almacenamiento de datos cuyo símbolo es TB y equivale a 1024 GB.

- Grand Central: una tecnología de programación paralela que permite usar las ventajas de los procesadores multinúcleo y optimizar la ejecución de aplicaciones de Mac OS X.
- OpenCL (Open Computing Language): que permitirá a los desarrolladores programar aplicaciones que utilicen la unidad de procesamiento gráfico (GPU) para usos no gráficos.

Capítulo 2 **Evaluación de herramientas de monitorización**

- 2.1 Herramientas de monitorización
- 2.2 Tabla comparativa de herramientas

2 EVALUACIÓN DE HERRAMIENTAS DE MONITORIZACIÓN

2.1 Herramientas de monitorización

En la actualidad se han creado múltiples programas y aplicaciones referentes a la monitorización de redes, algunos son desarrollos sencillos mientras que otros trabajan de forma más compleja. La forma de trabajar de estas herramientas puede variar, esto dependerá de los protocolos que usen para obtener la información requerida y la forma en que se despliega esta información también es distinta, dependiendo de la complejidad de la herramienta. En este capítulo se evalúan algunas de las herramientas que se encuentran en el mercado, algunas cuentan con licencia de software libre y otras son propietarias.

2.1.1 Cacti

Cacti es una aplicación muy vistosa que almacena toda la información necesaria para crear gráficas, estos datos provienen de una base de datos en MySQL. Esta aplicación es manejada por PHP y tiene soporte para SNMP que se complemente con RRDtool donde obtiene la información de los dispositivos mediante un recolector de datos que se ejecuta cada 5 minutos para posteriormente crear gráficas de todo tipo, desde el tráfico en la red hasta el uso de memoria de una computadora y sus particiones. Maneja usuarios, los cuales tienen distintos privilegios, como por ejemplo cambiar parámetros a las gráficas mientras que otros sólo pueden verlas (Cacti, About Cacti, 2004).

- **¿Qué es RRDtool?**

RRDtool es un sistema para almacenar y mostrar datos a través del tiempo. Tiene la característica que los datos se almacenan de manera compacta, ésta es la función de Round Robin. Algo de gran importancia es que la base de datos no crece con el tiempo.
Con RRDtool se puede mostrar los datos fácilmente en forma de gráficos para distintos periodos de tiempo. De manera interna se recogen los datos a graficar con una frecuencia determinada y se almacenan en diferentes archivos, según su resolución.

En esta base de datos RRD (Round Robin Database) hay tres archivos RRA (Round-Robin Archive) para diferentes periodos de tiempo. Uno de los archivos recolecta la información generada durante los últimos 5 minutos. El archivo que contiene los valores para una hora se calculan a partir de 12 valores del anterior archivo RRA. Finalmente el archivo para un día, tiene un valor que resume la medida para un día. Este se obtiene a partir de 24 valores del RRA de 1 hora. Cada uno de los archivos tiene un tamaño fijo, por lo que no crece en el tiempo. Existe un apuntador al último dato recogido. Los diferentes archivos se actualizan en paralelo a partir de las actualizaciones periódicas.

- **Elementos de configuración.**

Cacti cuenta con una interfaz gráfica y una serie de menús amigables, la forma de configurarlo también es sencilla para el usuario, cuenta con distintos menús donde se puede configurar desde los usuarios y sus privilegios, agregar nuevos host, hasta agregar nuevos plugins y gráficas.

Básicamente Cacti cuenta con dos pestañas, la primera de ellas llamada "console" donde es posible administrar la herramienta y cambiar la configuración de ésta, y de "graphs" que es un menú en el cual podemos tener accesos a las gráficas de los host que ya se han configurado (figura 2.1).

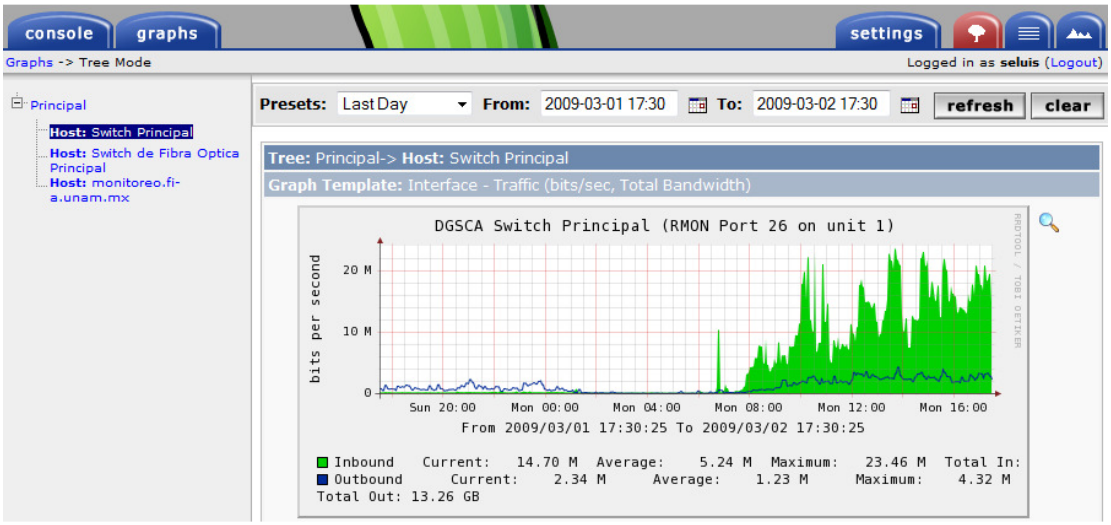


Figura 2.1 Gráfica del ancho de banda en un puerto

Cacti no sólo puede graficar los datos que nos proporcione el SNMP también puede graficar datos de diferentes scripts dándonos así una gran flexibilidad.

Los datos que nos proporciona Cacti son gráficas que tienen la característica de mostrar a detalle el dato que queremos con fecha, la hora exacta del evento, como los datos son proporcionados por un histórico, también nos puede almacenar los datos de todo un año, dándonos el pico máximo, el mínimo, el valor actual y el valor promedio. A todo esto se le puede agregar que Cacti nos proporciona la opción de acercarnos al valor que se requiere, para ello utiliza un método de interpolación de datos muy exacto, que nos da una idea de cómo estaba el valor en ese momento.

- **Ventajas**

La principal ventaja que maneja Cacti, es su capacidad para desplegar la información de los dispositivos que se monitorizan de forma gráfica. Se puede ver el comportamiento, la disponibilidad, el almacenamiento, temperatura y otras características más que se manejan en esta aplicación.

Gracias al protocolo SNMP es posible que cualquier dispositivo sea monitorizado por Cacti. Se puede hacer uso de las tres versiones de este protocolo (SNMPv3, SNMPv2 y SNMPv3). Soporta el manejo de un gran número de dispositivos para ser monitorizados al mismo tiempo, más de 10,000. Tiene la capacidad de manejar usuarios para acceder al sistema y otorgar diferentes privilegios a los mismos, desde el usuario que sólo puede ver las gráficas hasta el que tiene el privilegio de administrador.

Guarda un historial de cada parámetro de hasta un año para hacer un comparativo anual.

- **Desventajas**

Se mostró que Cacti tiene diversas cualidades y ventajas pero también posee debilidades y carencias. La información desplegada en las gráficas no siempre llega a ser clara y se necesita de un procesamiento extra por parte del usuario o administrador de Cacti para interpretar los datos. Es cierto, las gráficas son muy bonitas gracias a las herramientas que se utilizan para ser concebidas, pero no sirven de mucho si no se sabe qué es realmente lo que nos muestran.

2.1.2 Paessler Router Traffic Grapher (PRTG)

PRTG Traffic Grapher es una aplicación de Windows fácil de utilizar en la monitorización y clasificación del uso del ancho de banda.

Provee a los administradores con lecturas de tendencia en vivo y de largo plazo de sus dispositivos de red.

PRTG es principalmente utilizado para la monitorización del uso del ancho de banda (figura 2.2), pero además, se puede emplear para visualizar muchos otros aspectos de una red, tales como utilización de memoria y CPU. Con PRTG Traffic Grapher el usuario recibe datos detallados y entendibles referentes al uso del ancho de banda y de red que le ayuda a optimizar la eficiencia de la misma.

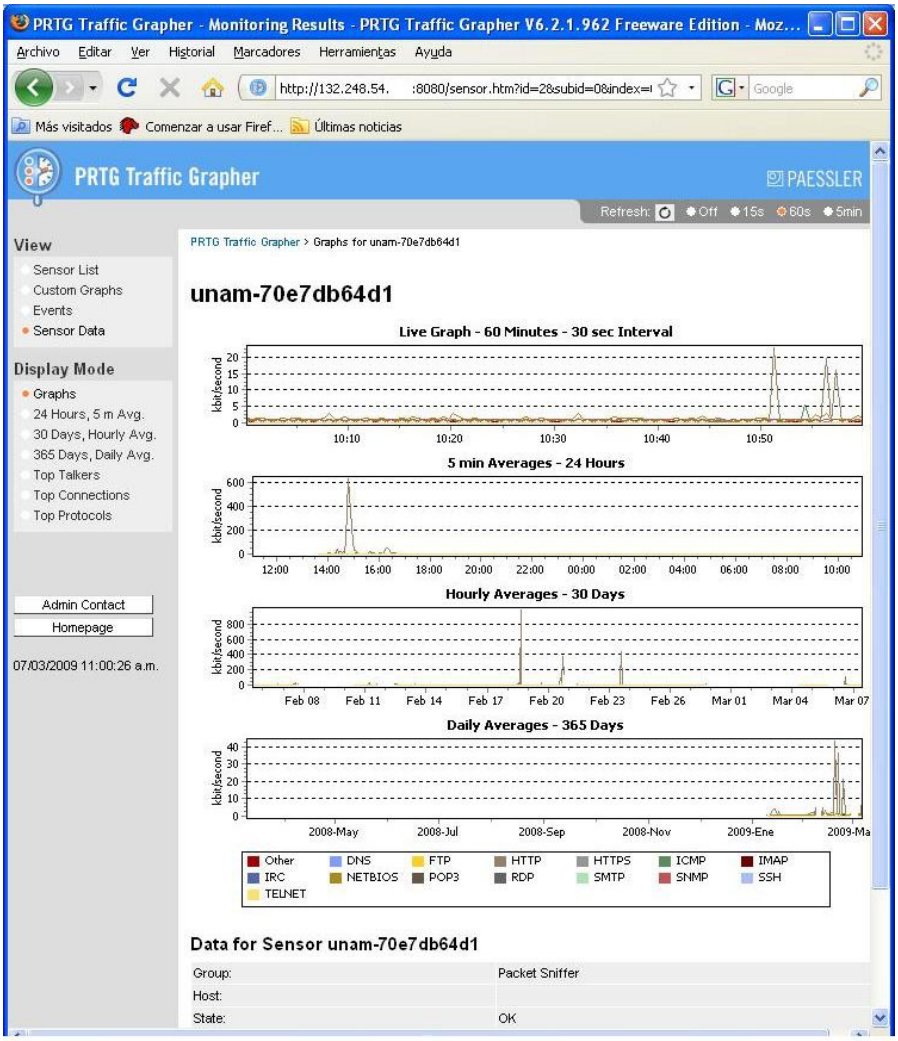
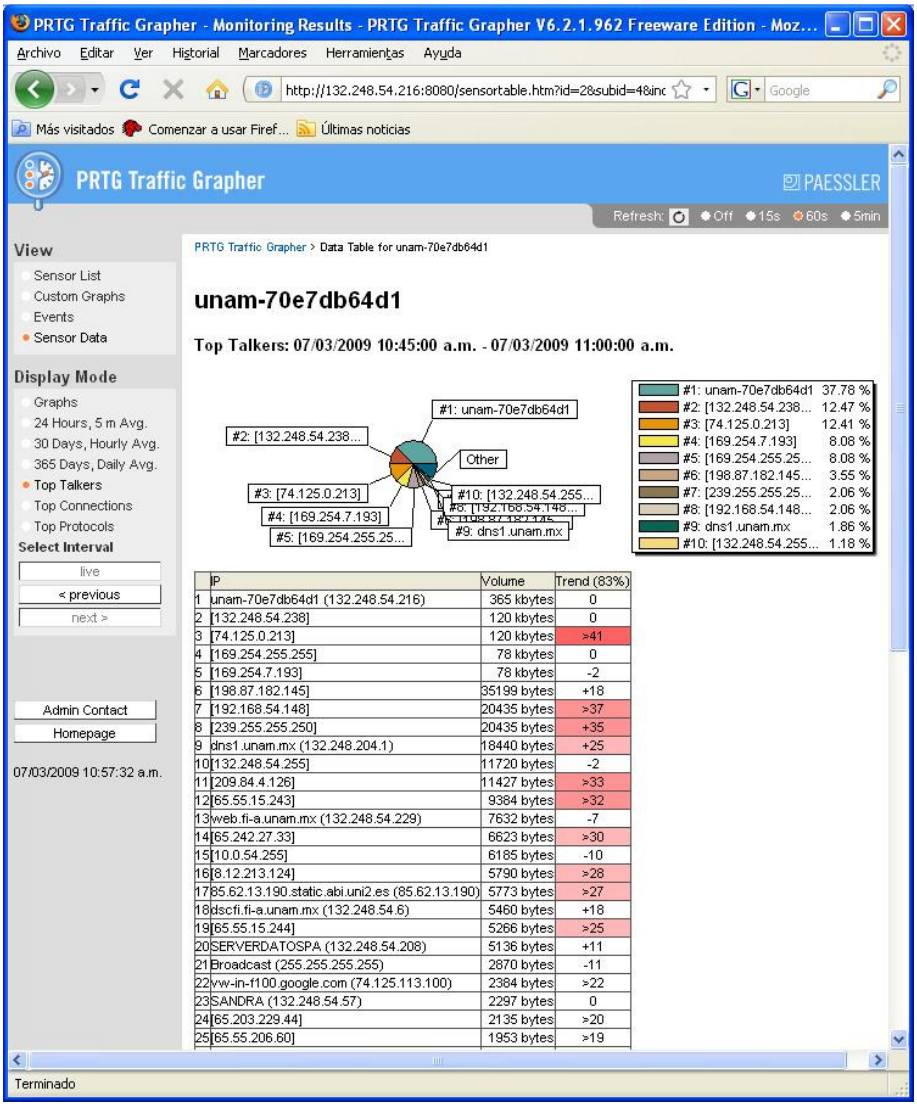


Figura 2.2 Gráficas del consumo de ancho de banda.

El entendimiento del tema de consumo de ancho de banda y recursos, es la clave para una mejor administración de red, para poder evitar los cuellos de botella en lo referente al ancho de banda y desempeño del servidor, averiguar qué aplicaciones, servidores o equipos consumen el ancho de banda (figura 2.3), entregar la mejor calidad de servicio a sus usuarios al ser proactivo, reducir costos al comprar hardware y ancho de banda acorde a las necesidades actuales.

La edición freeware de PRTG Traffic Grapher es completamente gratuita para uso comercial y personal y puede ser descargada sin costo, con el inconveniente que solo puede obtener información de únicamente 10 sensores. PRTG Traffic Grapher está diseñada para ser ejecutada sobre la red en una máquina Windows durante las 24 horas del día y registra constantemente los parámetros de uso de red. Los datos registrados son almacenados en una base de datos interna para posterior análisis.

Usando una interface fácil de usar, se puede configurar los parámetros para la monitorización, así como también crear reportes de uso. Para acceso remoto PRTG Traffic Grapher viene con un servidor web incorporado para proveer acceso a gráficos y tablas. Todos los métodos para capturar información sobre el uso de la red son soportados por SNMP (Simple Network Management Protocol) que es la forma básica de reunir información sobre el uso de la red y del ancho de banda. Se puede usar para monitorizar routers y switches puerto a puerto, así como para obtener lecturas de la memoria, carga de CPU, etc.



	IP	Volume	Trend (83%)
1	unam-70e7db64d1 (132.248.54.216)	365 kbytes	0
2	[132.248.54.238]	120 kbytes	0
3	[74.125.0.213]	120 kbytes	>41
4	[169.254.255.255]	78 kbytes	0
5	[169.254.7.193]	78 kbytes	-2
6	[198.87.182.145]	35199 bytes	+18
7	[192.168.54.148]	20435 bytes	>37
8	[239.255.255.250]	20435 bytes	+35
9	dns1.unam.mx (132.248.204.1)	18440 bytes	+25
10	[132.248.54.255]	11720 bytes	-2
11	[209.84.4.126]	11427 bytes	>33
12	[65.55.15.243]	9384 bytes	>32
13	web.fi-a.unam.mx (132.248.54.229)	7632 bytes	-7
14	[65.242.27.33]	6623 bytes	>30
15	[10.0.54.255]	6185 bytes	-10
16	[8.12.213.124]	5790 bytes	>28
17	85.62.13.190.static.abi.uni2.es (85.62.13.190)	5773 bytes	>27
18	dscfi.fi-a.unam.mx (132.248.54.6)	5460 bytes	+18
19	[65.55.15.244]	5266 bytes	>25
20	SERVERDATOSPA (132.248.54.208)	5136 bytes	+11
21	Broadcast (255.255.255.255)	2870 bytes	-11
22	vw-in-f100.google.com (74.125.113.100)	2384 bytes	>22
23	SANDRA (132.248.54.57)	2297 bytes	0
24	[65.203.229.44]	2135 bytes	>20
25	[65.55.206.60]	1953 bytes	>19

Figura 2.3 Lista de equipos que consumen mayor ancho de banda.

Monitorización de Paquetes: Con la monitorización de Paquetes que tiene incorporado, PRTG puede inspeccionar todos los paquetes de datos, pasando por la tarjeta de red para calcular el uso de ancho de banda.

- **Ventajas**

PRTG Traffic Grapher es una herramienta fácil de instalar y manejar, la forma en que puede configurarse es muy sencilla ya que cuenta con interfaces amigables para el usuario.

La forma en que nos muestra los datos es fácilmente interpretable, ya que lo hace de manera gráfica y nos realiza una lista los datos de manera legible.

Cuenta con su servidor web en el cual se pueden consultar los datos fuera de la máquina donde se encuentra la herramienta.

- **Desventajas**

Aunque los gráficos son fácilmente interpretables las gráficas no cuentan con un buen acercamiento llegando a ser un poco inexactas. Es una buena herramienta para graficar anchos de banda y la monitorización de paquetes, pero a comparación de otras herramientas solo se limita a estas dos funciones y no puede graficar otros datos distintos a los que se puedan obtener con el protocolo SNMP o con el recolector de paquetes que tiene integrado.

2.1.1 PRTG Network Monitor

PRTG Network Monitor es una potente herramienta de monitorización de Paessler AG. Asegura la disponibilidad de componentes de la red y mide el tráfico y el uso ella. Ahorra costos ayudando a evitar fallos, optimizar conexiones, economizando tiempo de implementación.

El programa opera 24 horas, 7 días a la semana en una máquina basada en Windows, los datos de monitorización son guardados en una base de datos para poder generar reportes históricos.

Las funciones del programa incluyen: monitorización del tiempo de actividad/inactividad, tráfico y uso, revisión de paquetes, análisis profundo y proporcionar reportes concisos (figura 2.11). Una interfaz amigable basada en web permite a los usuarios a configurar rápidamente los dispositivos y sensores de red que deseen monitorizar. Además, puede generar reportes de uso y proveer a colegas y clientes con acceso a gráficos y tablas de datos.

Todos los métodos de adquisición de datos de uso de red vienen integrados, así mismo los métodos comunes de la adquisición de datos en la red son soportados: SNMP (Simple

Network Management Protocol) y WMI (Windows Management Instrumentation) son usados para adquirir datos acerca del uso y el rendimiento de todos los sistemas que componen su red, incluyendo el uso de puertos individuales de switches y routers.

PRTG Network Monitor incluye más de 30 tipos de sensores para los servicios de red comunes (por ejemplo, PING, HTTP, SMTP, POP3, FTP, etc.), permitiendo a los usuarios a monitorizar redes por velocidad y fallas. Tan pronto como ocurra una caída, las alertas son enviadas por correo electrónico, mensajes SMS, radio localizador y otros medios. Los tiempos de requisición y caídas son grabadas en una base de datos interna, haciendo fácil compilar reportes de desempeño y tiempo de caída.

Mediante sus capacidades de sniffing de paquetes, PRTG puede inspeccionar todos los paquetes de datos analizando la red LAN o WLAN para calcular el uso de ancho de banda. Cuenta con el protocolo NetFlow que es usado por muchos enrutadores Cisco para medir el uso de ancho de banda. Es el método de monitorización más potente, adecuado para redes con mucho tráfico.

La Edición Gratuita de PRTG Network Monitor es completamente libre para uso personal y comercial. La Edición Comercial es usada para monitorear más de diez sensores.

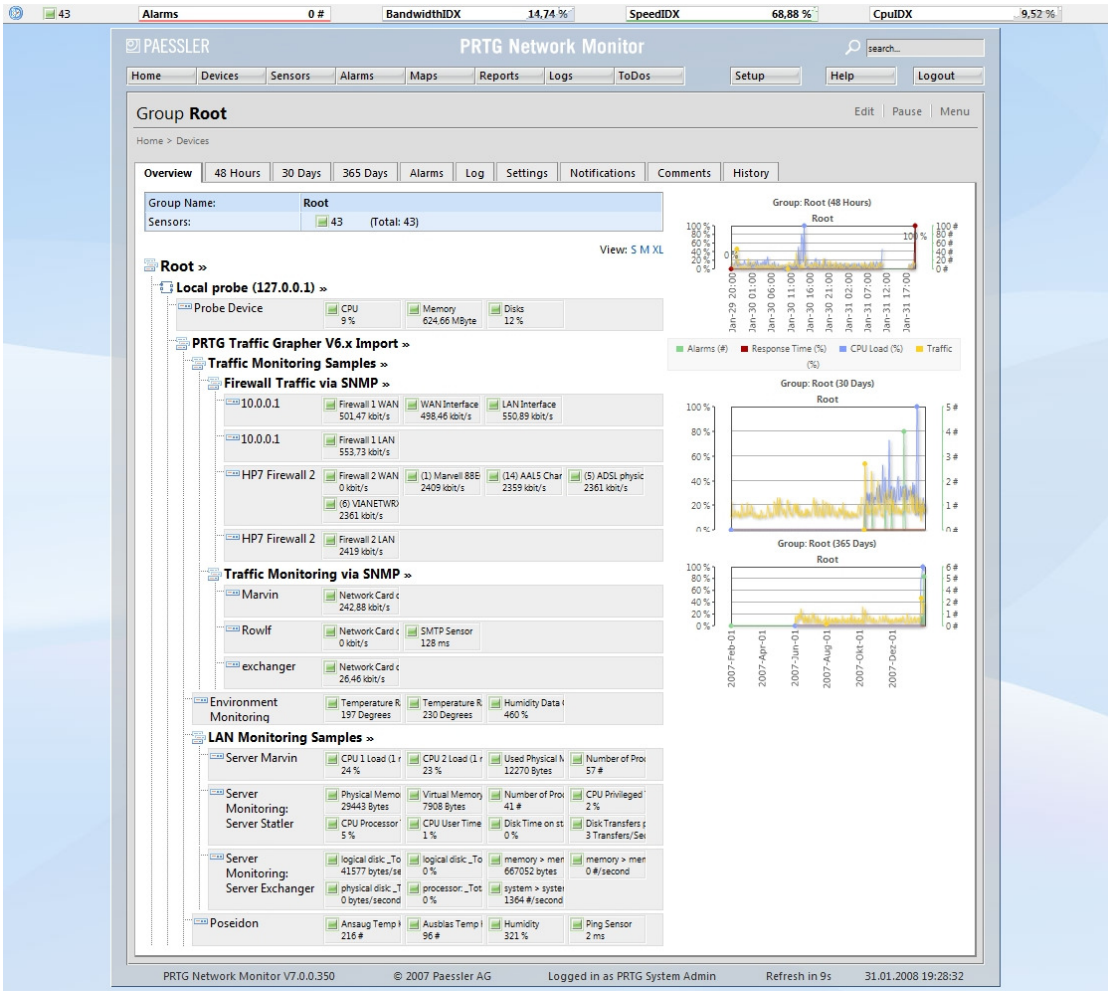


Figura 2.4 Reportes de sensores.

- **Ventajas**

Cuenta con un sistema de alarmas que nos permite darnos cuenta rápidamente de las fallas en algún sistema monitorizado.

Tiene una administración amigable vía web que le permite al usuario modificar cualquier parámetro, agregar o quitar sensores, además de que por este medio es posible consultar la información.

- **Desventajas**

La desventaja más importante en la versión gratuita, es su limitación en el número de dispositivos a monitorizar. La versión comercial puede variar en su precio, dependiendo

del número de sensores que necesite la empresa, los precios pueden variar de US$295.00 para 100 sensores, hasta US$ 6,525.00 para un número ilimitado de sensores.

2.1.2 Nagios

Es un sistema de código abierto para la monitorización de redes y se ha convertido en una herramienta muy popular entre los administradores de red. Monitoriza servidores y servicios que le sean especificados notificando los cambios que se hayan producido en los dispositivos. Comenzó con el nombre de Netsaint y fue creado por Ethan Galstad y hasta la fecha él y otro grupo de desarrolladores mantiene este proyecto activo. Fue diseñado originalmente para sistemas Linux pero también es usado en sistemas operativos Unix. Tiene licencia GNU publicada por la Free Software Fundation (Nagios, 2009).

Algunas de las características que monitoriza Nagios son los servicios de red, como los protocolos SMTP, POP3[13], HTTP, ICMP[14], etc. y los recursos de los servidores (carga del procesador, porcentaje de uso de los discos duros, etc.).

Tiene la facilidad de poder crear simples "plugins" (pequeño programa que interactúa con alguna aplicación del servidor para obtener una función o información específica), que permiten a los usuarios crear sus propios parámetros para ser verificados.

Así también tiene la posibilidad de definir la jerarquía de la red, permitiendo distinguir entre host caídos y host inaccesibles. Notifica vía correo electrónico, mensajes de texto SMS, o cualquier otro método definido por el usuario a un grupo de contactos cuando ocurre un problema con un servicio o en un host.

Contiene una interfaz web opcional, para observar el estado de la red actual, notificaciones, historial de problemas, archivos logs, etc.

Los únicos requerimientos que necesita Nagios para su funcionamiento son una computadora con sistema operativo Linux (Unix), un compilador de C y opcionalmente un servidor Web.

- **Funcionamiento**

Como se mencionó, Nagios funciona a través de plugins. Para monitorizar un sistema Linux o Unix remoto, se utiliza el plugin NRPE (Nagios Remote Plugin Executor). Éste funciona para recopilar los datos de los recursos locales de la máquina que se desea monitorizar (figura 2.4).

[13] Post Office Protocol - Protocolo de coreo electrónico.
[14] Protocolo de Mensajes de Control de Internet - Internet Control Message Protocol.

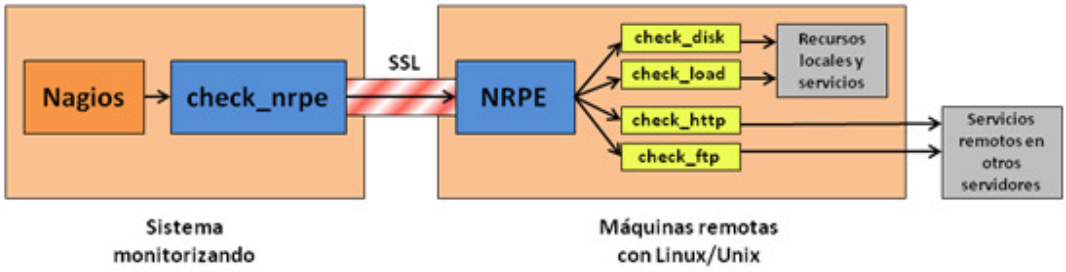


Figura 2.5 Funcionamiento de Nagios en máquinas Linux/Unix

Cuando Nagios necesita monitorizar uno de los recursos del sistema remoto ejecuta el modulo check_nrpe e indica cuál servicio necesita ser monitorizado. El check_nrpe contacta al demonio NRPE instalado en la máquina remota a través de una conexión SSL protegida. El demonio ejecuta el plugin apropiado para verificar el servicio o recurso solicitado. El resultado se pasa al demonio NRPE y de regreso al módulo check_nrpe que a su vez, entrega el resultado a Nagios.

Lo mismo sucede si se requiere monitorizar sistemas Windows, sólo que se necesita un módulo diferente, en este caso check_nt (figura 2.5). Dentro de la máquina remota Windows se necesita un plugin para interactuar con el servidor Nagios, como puede ser NSClient, NC_Net o algun otro.

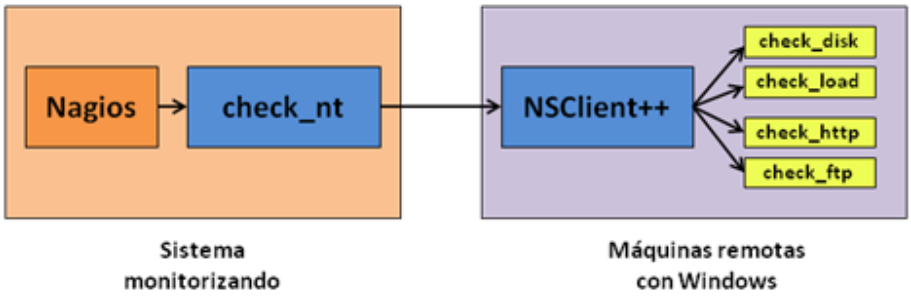


Figura 2.6 Funcionamiento de Nagios en máquinas Windows.

La instalación y configuración de Nagios se realiza a través scripts en modo consola.

En la parte gráfica, Nagios ofrece un menú en la parte izquierda de la pantalla, donde se encuentran las diferentes características que ofrece Nagios, como el estado de los hosts, reportes con histogramas y las alertas. En la parte central de la venta, se muestra a detalle cada parámetro monitorizado, así como también un pequeño cuadro resumen, con el estado en el que se encuentran todos los dispositivos (figura 2.6). (Galstad, 2007).

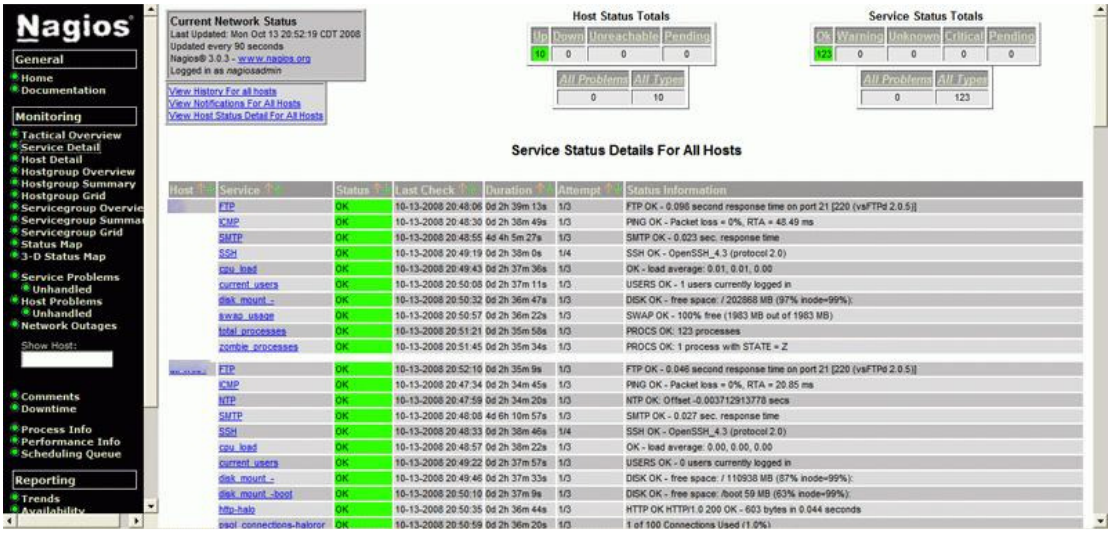


Figura 2.7 Vista general de los equipos

Nagios también cuenta con un esquema gráfico donde se pueden ver los equipos respecto a su distancia y grupo al que pertenecen (figura 2.7). Así mismo posee un historial donde se ven los errores, alertas y eventos no fallidos.

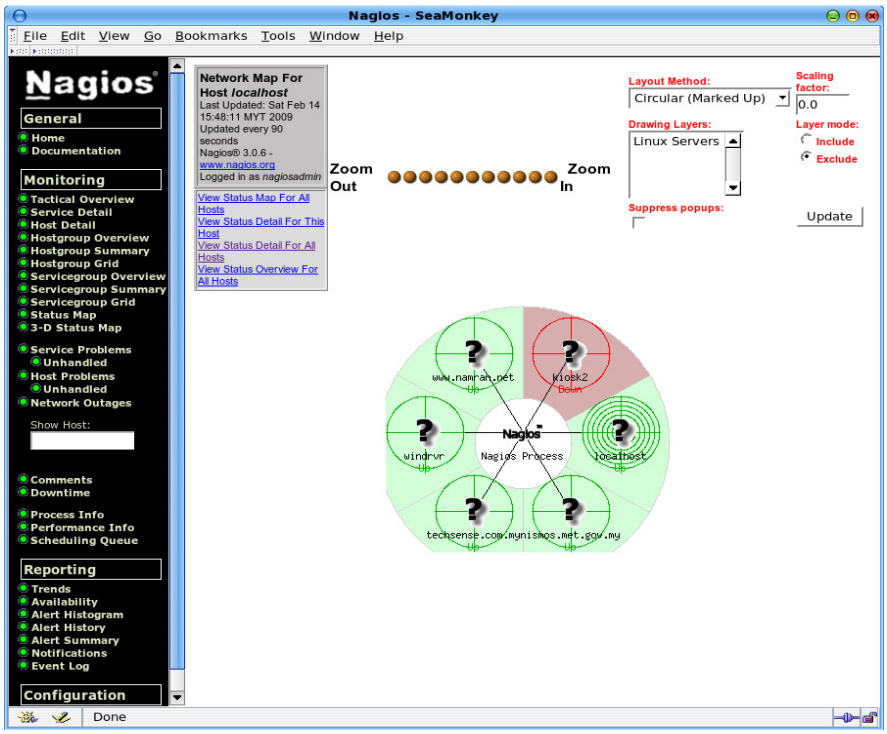


Figura 2.8 Esquema gráfico

- **Ventajas**

Nagios es una herramienta de monitorización completa. Una de sus ventajas claras, es el uso de plugins prediseñados para revisar el estado de los servicios y la capacidad para crear los propios para monitorizar aspectos específicos de acuerdo a las necesidades del administrador. Tiene soporte para monitorizar miles de dispositivos y equipos. Envía alertas vía correo electrónico y mensajes SMS en caso de que algún equipo falle. Tiene una interfaz gráfica web nada compleja.

- **Desventajas**

La configuración de los equipos no es nada fácil, ya que todo se realiza vía consola mediante archivos de configuración donde se agregan, modifican, y dan de baja los equipos, siendo ésta su peor desventaja ya que no se pueden hacer estas modificaciones desde el entorno Web.

2.1.3 Snort

Desde un inicio, la misma documentación de Snort indica que no es realmente difícil de usar la aplicación, sin embargo, puede llegar a serlo debido a que su uso es a través de línea de comandos y cuenta con múltiples opciones que no siempre llegan a ser obvias o fáciles.

Snort es un sniffer y un detector de intrusos basado en red. Implementa un motor de detección de ataques y barrido de puertos que permite registrar, alertar y responder ante cualquier anomalía previamente definida. Es muy similar al tcpdump pero con características diferentes y funciones mejoradas. Se encuentra bajo la licencia pública general de GNU, lo cual lo hace gratuito y funciona en los sistemas operativos Windows y Unix (What is Snort?, 2008).

Snort puede llegar a ser configurado de tres formas para su ejecución:

- Modo Sniffer. Simplemente lee los paquetes salientes de la red y los muestra de manera continua en la pantalla de la consola.

- Modo registro de paquetes. Permite guardar en un archivo la información de los paquetes para un análisis posterior.

- Modo Sistema de Detección de Intrusos en Red (NIDS por sus siglas en inglés). Permite el análisis del tráfico de red, para buscar paquetes que coincidan con algún patrón establecido en las reglas de configuración y después, efectuar acciones, dependiendo de estas reglas.

Se evaluó la segunda opción ya que un IDS no se encuentra dentro del objetivo.

La forma de instalar, es un tanto complicada, ya que primeramente se necesitan descargar varios programas alternos por que Snort se apoya de otros programas para poder recopilar la información que necesita para su funcionamiento, en el caso de instalar Snort en ambiente Windows se utilizan:

- El Microsoft Baseline Security Analyzer (MBSA) que sirve para detectar comunes desconfiguraciones de seguridad.
- Winpcap, que permite acceder a conexiones entre capas de red en ambientes Windows.
- Oinkmaster, es un simple script en Perl que ayuda a actualizar las reglas de Snort.
- ActivePerl, que es la distribución estándar de Perl, necesaria para que Oinkmaster funcione.
- Entre otros programas más.

- **Ventajas**

La eficiencia que se maneja es enorme, ya que emplea una gran cantidad de procesos automatizados para eliminar el error humano y minimizar los tiempos involucrados en el manejo y configuración de Snort. El sistema informa a una lista de usuarios vía email cuando ocurre un cambio o existen errores en el sistema.

- **Desventajas**

El sistema central administra todas las configuraciones de Snort, lo hace susceptible a que un error de configuración sea distribuido y no sea identificado a tiempo. Por esta razón, se recomienda que sólo usuarios experimentados de Snort manejen el entorno de configuración.

2.1.4 Ntop

Este proyecto comenzó en 1997, como una aplicación para monitorizar dentro de la Universidad de Pisa. Para el siguiente año se lanzó una primera versión comercial con licencia GPL. Desde entonces se han indo incluyendo nuevas características y agregando

soportes para diferentes sistemas operativos. Para 2008 se alcanza la versión 3, junto con otras aplicaciones que se complementan con ntop.

ntop es una herramienta simple de monitorización de software libre con licencia GPL. Una de sus grandes características es que está diseñado para varias plataformas, ya sea Unix, Windows o MacOS y se necesitan requerimientos mínimos para su instalación.

Clasifica los paquetes de acuerdo a los protocolos empleados, guarda un historial de las sesiones TCP para generar gráficas, aunque éstas no sean su fuerte. Identifica ruteadores, DNS, proxy y servidores de internet (figura 2.8) e informa acerca del estado de los paquetes y los servidores (figura 2.9). También tiene soporte para tecnologías propietarias de Cisco para la recolección de estadísticas de IP.

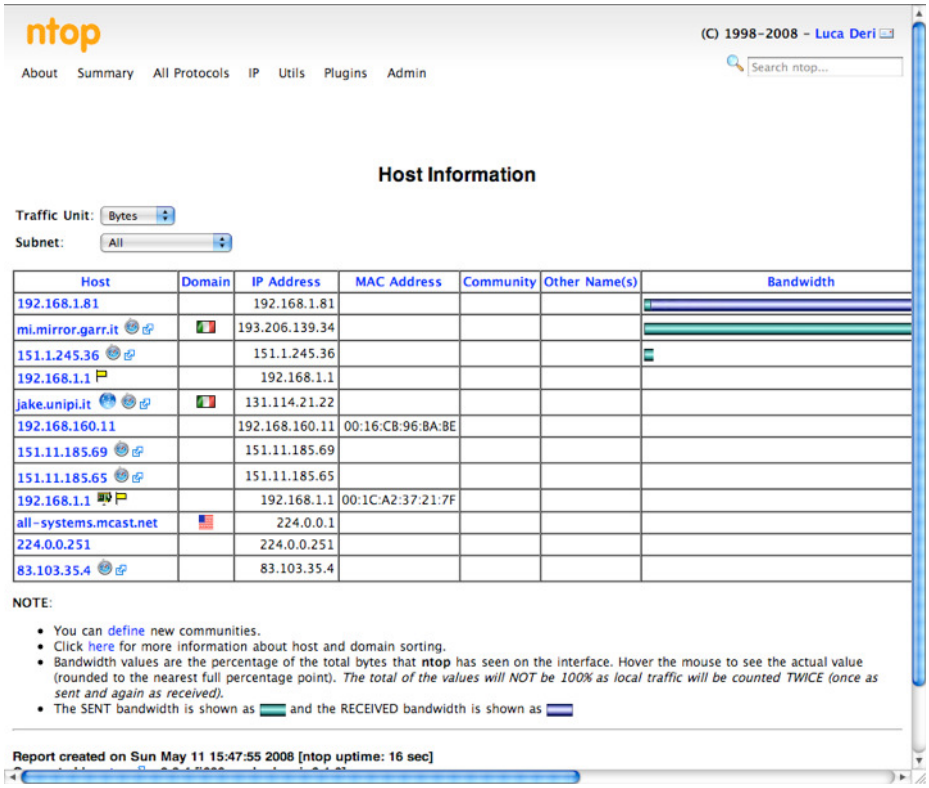


Figura 2.9 Información de lo hosts con ntop

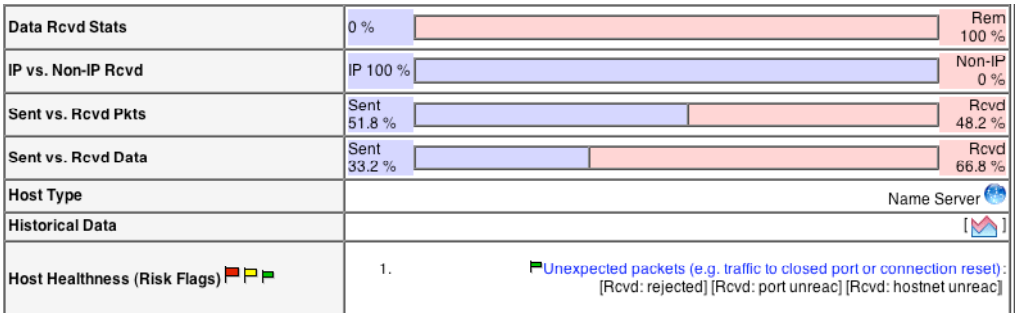


Figura 2.10 Información de los paquetes

- **Ventajas**

Algunas ventajas que tiene ntop es que el tráfico de red puede ser desplegado de diferentes formas, como puede ser el destino, la fuente, protocolo, dirección MAC, etc. así como también las estadísticas pueden ser agrupadas por protocolo y por número de puerto. Tiene la habilidad de reconocer el sistema operativo de algunos dispositivos.

En este ámbito, ntop puede ser ejecutado desde casi cualquier plataforma y su instalación no es muy compleja. Cuenta con una interfaz web como también el manejo de historiales.

- **Desventajas**

La principal desventaja de ntop es que no despliega información de manera instantánea, solamente los totales de largos periodos de actividad y muestra promedios. Esto puede presentar dificultades a la hora de diagnosticar problemas que hayan comenzado repentinamente. La parte web no muestra sesiones, es decir, cualquier persona que sepa la ruta de acceso puede entrar al sistema, lo cual no es seguro. Las gráficas que se manejan no muestran suficiente información, más bien, la información se representa en tablas que pueden llegar a ser muy extensas y dificultan la lectura de valores.

2.1.5 OSSIM (Open Source Security Information Management)

OSSIM, manejador de información segura de código abierto. El objetivo de esta herramienta es conjuntar una serie de herramientas, las cuales trabajando juntas, otorgan al administrador de red o de seguridad, una vista detallada de cada aspecto de sus servicios y dispositivos. Entre las herramientas que maneja están:

- Arpwatch, es usado para detectar anomalías en sistemas operativos tipo Mac.
- P0f, usado para la detección de sistemas operativos.
- Pads, usado para la detección de anomalías en el servicio.

- Nessus, se encarga de la detección de vulnerabilidades.
- Snort, la parte de IDS, usado en conjunto con Nessus.
- Ntop, crea una base de información para detectar un comportamiento anormal.
- Nagios, monitoriza la disponibilidad de la información.
- Entre otros.

Su instalación puede ser muy complicada o muy fácil. Se pueden instalar una a una de forma manual todas las herramientas que utiliza esta aplicación, volviéndose tediosa y hasta a veces complicada o simplemente instalar una máquina desde cero con un disco instalador que contiene el sistema operativo Debian y todas las herramientas necesarias para que, en cuestión de minutos se tenga lista esta aplicación.

Gracias al manejo de sesiones, se puede conectar vía web a la aplicación OSSIM donde muestra una barra de opciones con los diferentes parámetros a revisar. Cuenta con una configuración vía web, lo cual otorga una mayor facilidad para agregar o modificar un elemento en el esquema de monitorización.

La mayor característica de esta aplicación, es que conjunta una gran cantidad de herramientas de monitorización de software libre y desde una misma pantalla (figura 2.10) se puede acceder a los diferentes parámetros que se especializan dichas herramientas y así tener un mejor manejo de ellas.

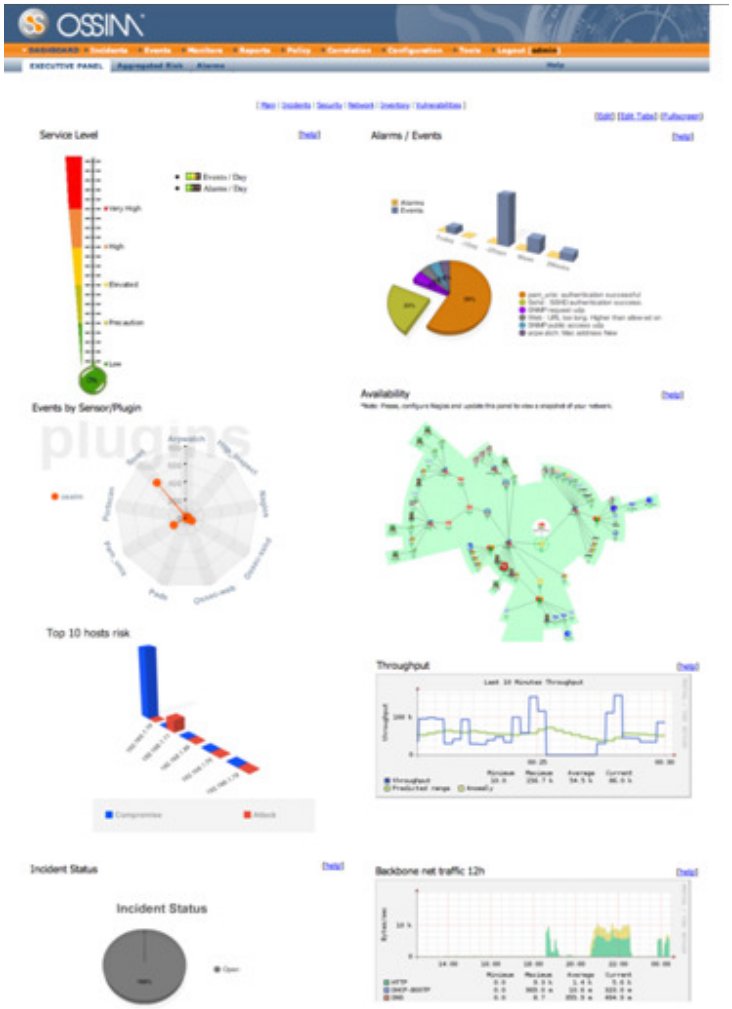


Figura 2.11 Pantalla principal de OSSIM

- **Ventajas**

La capacidad de contener una gran cantidad de herramientas ya definidas en el mercado y poder usarlas de manera práctica desde una misma consola con interfaz gráfica. El conjunto de estas herramientas hacen que la monitorización sea más extensa para evaluar muchísimas características de una red.

- **Desventajas**

No incorpora algo nuevo al área de monitorización. Simplemente junta herramientas que ya han sido desarrolladas por otros. Además, al tratar de ejecutar muchas herramientas al mismo tiempo, puede haber pérdida de paquetes e incluso un mal rendimiento del sistema.

2.1.6 Intelligent Management Center (IMC)de 3Com®

IMC es una herramienta desarrollada por la empresa 3Com® para administrar todos los dispositivos de red, no sólo los que están bajo la marca de 3Com®, sino también equipos de otros fabricantes (3Com&H3C, 2009). Está desarrollado bajo una arquitectura orientada a servicios que permite integrar herramientas tradicionalmente separadas para poder administrar recursos, servicios y usuarios. Cuenta con aplicaciones que permiten llevar un mejor control sobre la red y así poder encontrar errores o fallas antes de que se vuelvan más grandes. Para esto, tiene un sistema de alertas capaz de enviar correo electrónico o mensajes SMS. La pantalla inicial del sistema (figura 2.12) nos muestra a primera vista con qué equipos cuenta nuestra red así como el estado en el que se encuentra.

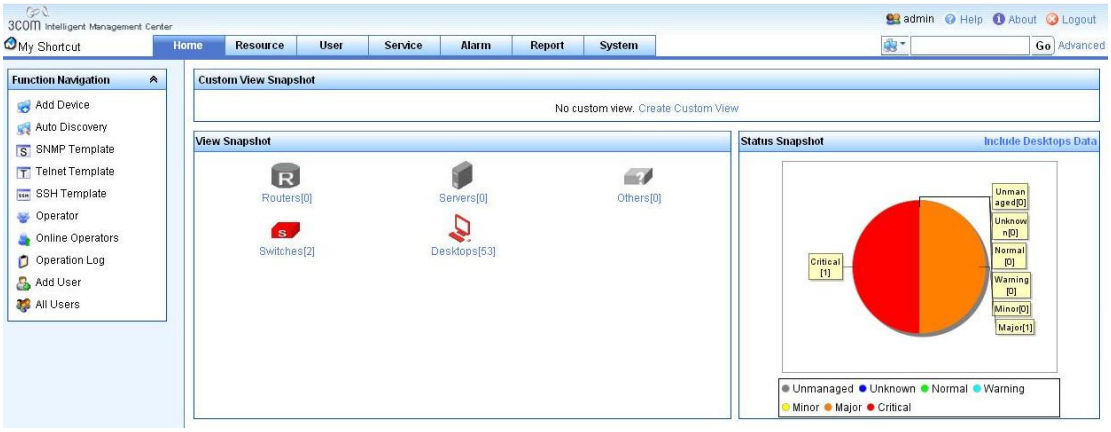


Figura 2.12 Inicio de IMC

- **Características**

IMC cuenta con un control de acceso a la herramienta, donde diferentes usuarios tienen acceso a diferentes dispositivos, ya sea para monitorizar o realizar un cambio. Hay diferentes formas de presentar los resultados, dependiendo de la configuración que se tenga para cada usuario.

Se pueden crear gráficas que representan la topología de la red, para poder darnos una idea de cómo están interconectados los dispositivos y tener una idea más clara a la hora de resolver un problema.

IMC detecta dispositivos mediante Secure Shell, Telnet o ping[15]. También tienen soporte para el protocolo SNMP en versión 1 y 2. De esta forma puede acceder a todos los equipos activos sin importar el nombre del fabricante. Si un equipo no es accesible o un parámetro se encuentra por encima de lo normal, se activa una alerta que a su vez envía un correo electrónico a cierto destinatario o incluso un mensaje vía SMS. Hay que aclarar que para ello se necesita hardware adicional. Se pueden crear reportes del comportamiento de los dispositivos en formato .pdf o .xls para dotar a los administradores de red mayores recursos a la hora de tomar decisiones.

Se puede analizar el tráfico de la red para deducir qué tipo de información viaja a través de la misma, para así saber qué protocolos son los más utilizados por los usuarios y saber quién está haciendo mal uso de la red.

Existen diferentes versiones de IMC que admiten mayor o menor número de nodos, es decir, dispositivos conectados en la red para poder ser administrados. El Intelligent Management Center puede ser utilizado sobre plataforma Windows o Solaris. Se necesitan al menos de 2 GB a 4 GB de memoria RAM y de 10 GB a 190 GB de disco duro, dependiendo del número de nodos y el sistema operativo donde se encuentre instalado. Necesita una base de datos, en este caso hace uso del Microsoft SQL Server 2005 u Oracle 10g. La figura 2.13 muestra los parámetros de configuración de la herramienta.

[15] Comando para diagnosticar los errores en redes o enrutadores IP a través del envío de paquetes.

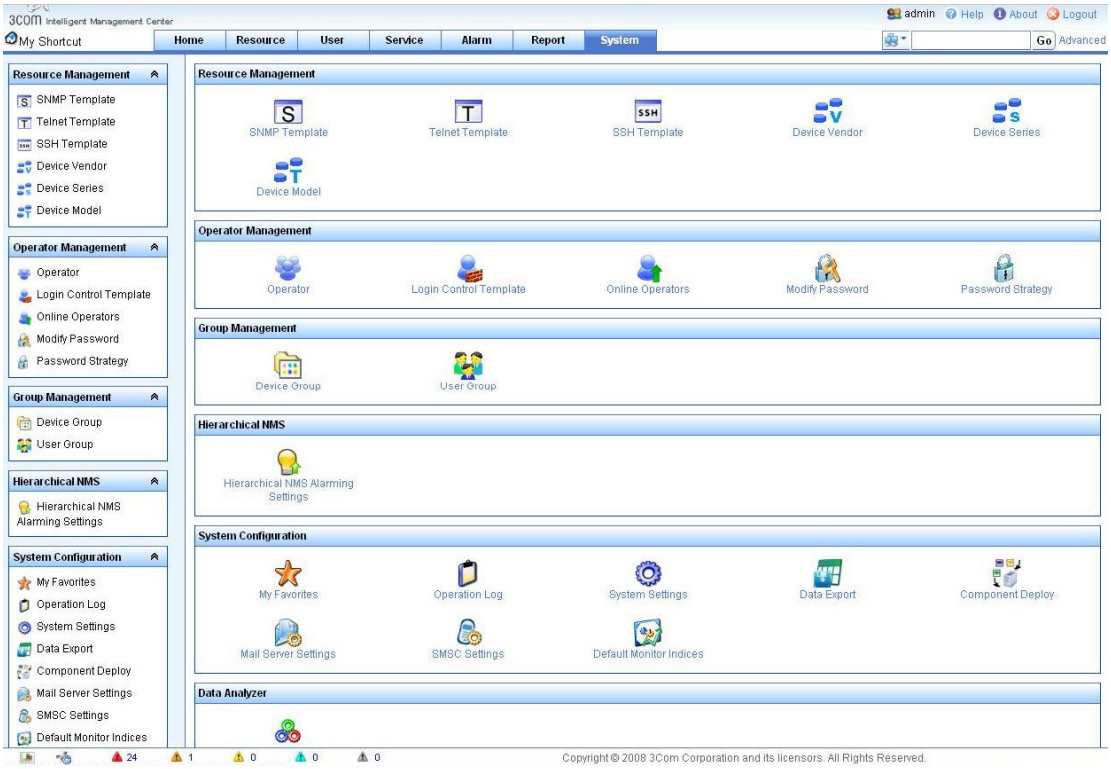


Figura 2.13 Configuración de IMC

- **Ventajas**

Cuenta con una interfaz gráfica simple que es muy entendible y fácil de usar. Tiene en conjunto múltiples aplicaciones que son útiles para manejar y optimizar la red. Esas aplicaciones hacen de esta herramienta de monitorización y administración de equipos una de las más completas con la que se cuenta hoy en día ya que se pueden encontrar programas y desarrollos que sólo cubren cierta parte de las múltiples tareas que necesita llevar a cabo un administrador de red.

- **Desventajas**

La principal desventaja que tiene IMC es que es un software que se encuentra bajo una licencia propiedad de una empresa privada, por lo que su costo es elevado. No todas las aplicaciones vienen instaladas, ya que dependen de la versión que se haya adquirido. Agregado a esto, se necesita una licencia de Windows SQL Server para poder manejar los datos. Una vez que se cuente con una base de datos más grande ya que inicialmente sólo puede almacenar hasta 4 GB. Lo que lleva a utilizar gran cantidad de recursos de nuestro sistema como es la memoria RAM, el procesador y gran cantidad de espacio en el disco duro para funcionar correctamente.

2.2 Tabla comparativa de herramientas

Características		CACTI	PRTG	NAGIOS	SNORT	IMC	NTOP	OSSIM	PRTG Network Monitor
Recolección de datos		SNMP, RRDTool	SNMP, WMI, Packet Sniffing	SNMP, SSL, SSH	Packet Sniffing	SNMP, SSH, Telnet	SNMP, RMON	SNMP, SSL, SSH, Packet Sniffing, RMON	SNMP, WMI, NetFlow
Alarmas	**Correo**	✓	✓	✓	X	✓	✓	✓	✓
	SMS	*	*	*	X	✓	*	*	*
Sistema Operativo		Linux Unix Windows	Windows	Linux Unix	Linux Unix Windows	Windows	Linux Unix Windows	Linux Unix	Windows
Tipo de Software		Libre	Licencia (Freeware)	Libre	Libre	Licencia	Libre	Libre	Licencia (Freeware)
Permisos a usuarios		✓	✓	✓	✓	✓	X	✓	✓
Generación de Reportes		*	✓	*	*	✓	X	✓	✓

Tabla 2.1 Comparativa de herramientas utilizadas para la monitorización

* Se necesitan plugins adicionales.

En este capítulo se analizaron sólo una muestra del total de herramientas que existen actualmente, ya que sería imposible analizar todas las existentes a detalle, y no es el propósito de este trabajo. Estas herramientas funcionan de manera similar con sus propias características, algunas son más elaboradas que otras, unas de ellas cuentan con licencia comercial y otras tienen licencia de software libre, pero la mayoría contiene funciones similares que nos ayudan a cuidar la integridad y disponibilidad de la red.

Hay que tener en cuenta que es decisión del administrador de red escoger qué herramienta le acomoda mejor dependiendo de sus necesidades.

Estas herramientas nos proporcionan aspectos importantes como su funcionamiento, los parámetros que se ocupan para monitorizar, los protocolos usados, los sensores y alarmas para poder implementar un sistema de monitorización que cumpla los objetivos planteados.

Capítulo 3 **Metodologías de Ingeniería de Software**

- 3.1 Proceso de diseño
- 3.2 Modelos del proceso del software
- 3.3 Iteración de procesos

3 METODOLOGÍAS DE INGENIERÍA DE SOFTWARE

3.1 Proceso de diseño

La resolución de problemas siempre ha estado sumamente ligado al trabajo que desempeña un ingeniero, cuya función principal es la de realizar diseños o desarrollar soluciones a necesidades sociales, industriales o económicas.

Los ingenieros, utilizan el conocimiento de la ciencia y la matemática y la experiencia apropiada para encontrar las mejores soluciones a problemas concretos, creando modelos matemáticos apropiados a los problemas que les permiten analizarlos rigurosamente y probar las soluciones potenciales.

Para poder desarrollar una solución apropiada para los distintos problemas que se presentan, los ingenieros encargados del proyecto deben incorporar las estrategias adecuadas de desarrollo y utilizar distintas herramientas de ingeniería del software, para poder dar respuesta a estos problemas, estas estrategias se denominan paradigmas de ingeniería del software o modelado de procesos.

Distintos autores, vinculados con el modelado de procesos han contribuido con distintos planteamientos sobre la forma en que se debe evaluar los problemas para su solución, uno de estos atures es Edward V. Krick, que nos proporciona cinco fases para dar respuesta a los problemas de manera ingenieril (ver Figura 3.1).

- Formulación del problema: el problema de que se trate se define en forma amplia y sin detalles.

- Análisis del problema: en esta etapa se le define con todo detalle.

- Búsqueda de soluciones: las soluciones alternativas se reúnen mediante indagación, investigación, invención, etcétera.

- Decisión: todas las alternativas se evalúan, comparan y se seleccionan hasta que se obtiene la solución óptima.

- Especificación: la solución elegida se expone por escrito detalladamente.

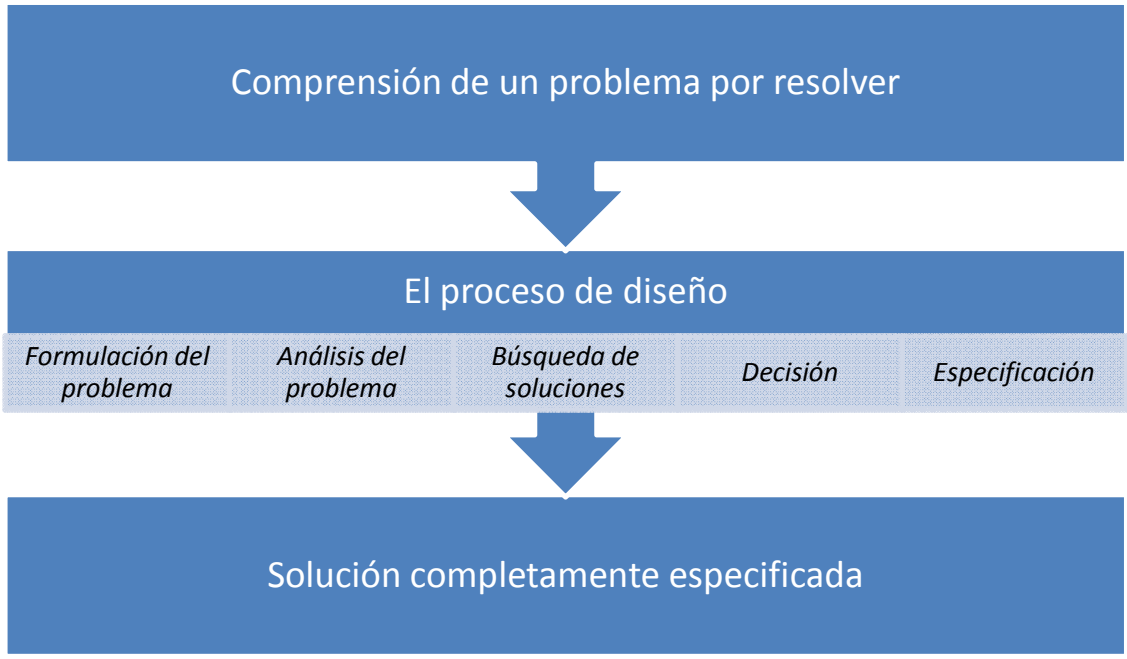


Figura 3.1 Proceso de diseño

- **Formulación del problema**

Lo primero que nos interesa es conocer, saber, lo que será investigado: ¿Por qué?, ¿Para qué?, ¿Cuál es el valor o la importancia del hecho o fenómeno a investigar?

Además de esto tenemos que valorar los alcances del problema, se sugiere que se planteen de forma generalizada con un amplio panorama en un principio para que posteriormente cuando se entre en detalles, no se pierdan de vista los objetivos principales, ya que una vez que se ven las partes del problema por separado es casi imposible ver el problema de forma general.

La formulación del problema debe de expresarse en términos generales del problema particular, ignorando los detalles momentáneamente y concentrándose en la identificación de los estados de entrada y salida, (A y B), para tener el conocimiento de la situación actual y a donde se quiere llegar.

- **Análisis del problema**

En el análisis del problema es necesario saber más al respecto de la entrada y la salida del problema analizado, es durante esta etapa del proceso de diseño que se determinan las características cualitativas y cuantitativas de los estados A y B.

Muy pocas características de los estados A y B son constantes, la mayor parte de éstas sufren alguna variación o modificación, estas características dinámicas son conocidas como variables de entrada y variables de salida.

Generalmente hay límites para el grado en que pueden fluctuar estas variables, estos límites establecidos son necesarios para poder resolver satisfactoriamente el problema antes planteado y usualmente son estimaciones de confianza que se le asignan a los estados de entrada y salida.

Algo que también se debe tomar muy en cuenta en esta etapa, son las restricciones, que es una característica de una solución que se fija previamente por una decisión, por la naturaleza del problema, por requisitos legales o por cualquier otra disposición que sea necesario cumplir.

No obstante existen restricciones absurdas que provienen de decisiones sin fundamento o sin el correcto análisis o por mera suposición, que el ingeniero deberá detectar y exponer con el fin de ampliar el número de soluciones posibles.

Es en esta etapa donde los criterios que se utilizan para seleccionar el mejor diseño deben identificarse. Realmente, los criterios cambian muy poco de problema a problema; el costo de construcción o fabricación, la seguridad, la confiabilidad, la facilidad de mantenimiento, la funcionalidad, entre otros. Lo que cambia significativamente de problema a problema, es la importancia relativa de criterios en particular.

Un criterio especialmente importante afectará a los tipos de soluciones que se destacan en la búsqueda de alternativas, y este hecho debe ser conocido antes que inicie tal búsqueda.

Para que un ingeniero resuelva inteligentemente el problema, debe determinar primero la utilización o uso esperado, es decir, el grado en que se ha de emplear la solución propuesta.

- **Búsqueda de soluciones**

En esta fase del proceso de diseño, se buscan activamente las soluciones posibles, basándose en una búsqueda e investigación exhaustiva. Este proceso de búsqueda de soluciones es relativamente directo y consiste en explorar nuestra memoria, consultar libros, informes técnicos, pero hay otra fuente de soluciones, las propias ideas que son el producto del proceso mental llamado invención.

La invención de cada ingeniero es diferente en proporción del grado de conocimientos que se tenga al respecto del problema y las propias aptitudes, estos conocimientos darán pauta para una mejor solución del problema, ya que se podrá resolver de una manera más

óptima. Al tener mayor conocimiento del los estados de entrada y salida se tiene la posibilidad de que el problema sea resuelto de una manera más adecuada.

No obstante, el inventar soluciones es un método poco directo y controlable ya que a veces los problemas a los que nos enfrentamos ya han sido resueltos en otra ocasión, el buscar soluciones hechas es un método más directo y la inventiva es utilizada para adecuar esta solución a nuestro problema u optimizar la solución para un mejor desempeño.

La búsqueda de la solución óptima es un proceso que puede continuar de manera indefinida, pero claro está, que en la vida real siempre existen límites, como por ejemplo límites temporales, económicos o ambos, etcétera. Es difícil definir si la solución que se encontró, es la óptima con respecto a los límites de búsqueda. Lo que sí es posible establecer respecto a la elección de la solución, es que esta decisión debe tomarse antes de llegar a un valor de utilidad decreciente.

- **Decisión**

En la fase de búsqueda se amplía el número y la variedad de las soluciones posibles.
En la fase de decisión lo que se necesita es un procedimiento de eliminación que reduzca estas alterativas a la solución preferible. La eliminación comienza con despreciar las ideas deficientes o de calidad inferior, con frecuencia este proceso se hace de manera burda y relativamente rápida, a las ideas restantes se les añaden más detalles y serán evaluadas mediante métodos más refinados. Este método es continuo y se llevará a cabo hasta que se consiga la solución adecuada.

Para la toma de decisiones existe un procedimiento que consta de 4 pasos:

- Seleccionar los criterios y determinar su importancia relativa.
- Predecir el funcionamiento de las soluciones alternativas con respecto a tales criterios.
- Comparar las alternativas sobre la base de los funcionamientos predichos.
- Hacer la elección.

Existen criterios como la confiabilidad, la operatividad, la disponibilidad y el mantenimiento que son criterios generalizados para la toma de dediciones:

- La confiabilidad significa que la probabilidad de que el elemento o sistema en cuestión no falle durante un periodo de tiempo bajo condiciones prescritas.

- La operatividad se refiere a la facilidad con que un diseño determinado puede ser manejado u operado.

- La disponibilidad es la proporción de tiempo que una máquina está en condiciones de ser utilizada.

El mantenimiento define los costos y tiempos necesarios para que la solución se ajuste a modificaciones o fallos predecibles al exponerse condiciones extremas.

- **Especificación**

Los datos de entrada a esta fase son la solución elegida, parte de ella en forma de bosquejo, apuntes, cálculos, etcétera, y gran parte de ella todavía en la cabeza del ingeniero. Además de ser incompleto, este material está desorganizado y difícilmente en condiciones de poder ser presentado a los jefes o a los clientes.

Falta describir con los detalles suficientes los atributos físicos, lógicos y las características de funcionamiento de la solución propuesta, de manera que las personas que deben aprobarla, los encargados de su construcción y quienes la manejarán y conservarán, puedan desempeñar satisfactoriamente sus funciones. Es de vital importancia que la presentación de de la solución se dé a conocer de manera cuidadosa y detallada ya que generalmente alguien ajeno a nosotros operara o manipulara la obra realizada.

Los datos de salida de esta fase consisten usualmente de dibujos del proyecto, un informe escrito y, posiblemente, un modelo físico o iconográfico tridimensional. Los primeros de planos o presentaciones pueden ser simplemente dibujos de una solución cuidadosamente realizados; detallados y acotados.

El informe técnico, suele ser un documento bastante formal que describe la propuesta con palabras, diagramas, planos y referencias técnicas. Este informe también describe el funcionamiento de la solución y proporciona una evaluación cabal de ella.

El diagrama general del proceso de diseño lo podemos expresar como se muestra en la Figura 3.2

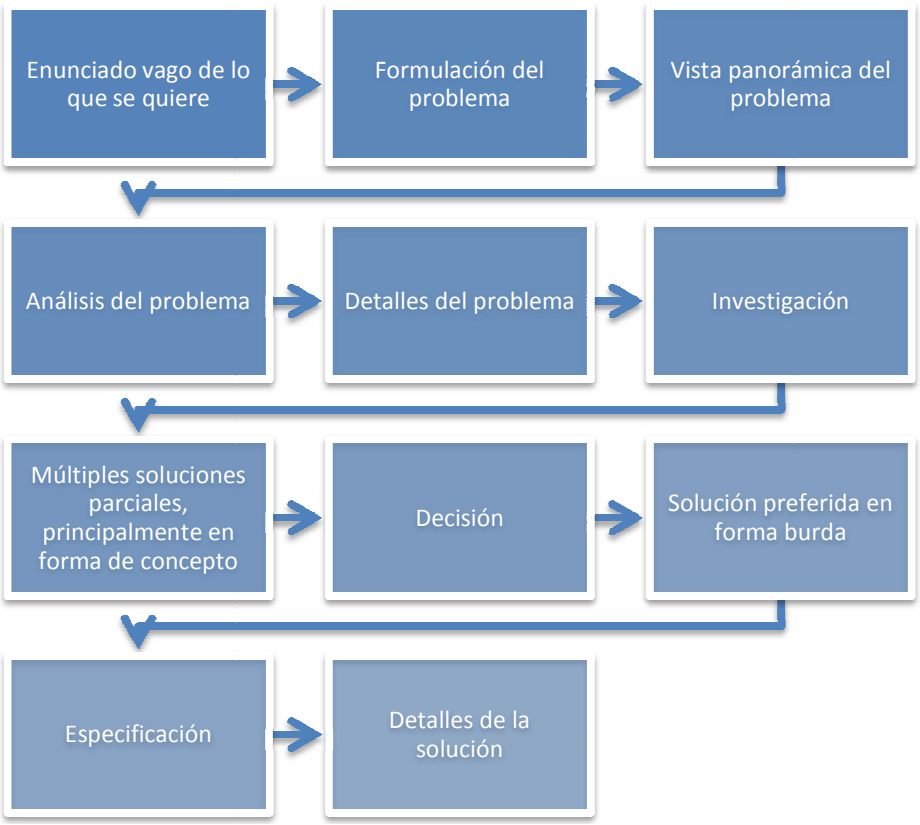


Figura 3.2 Diagrama general del proceso de diseño

3.2 Modelos del proceso del software

Un modelo del proceso del software es una representación abstracta de un proceso del software. Cada modelo de proceso representa un proceso desde una perspectiva particular y así proporciona sólo información parcial sobre ese proceso (Sommerville, 2005).

Los modelos de procesos generales son:

- El modelo en cascada. Considera las actividades fundamentales del proceso de especificación, desarrollo, validación y evolución, y los representa como fases separadas en el proceso, tales como la especificación de requerimientos, el diseño del software, la implementación, las pruebas, etc.

- Desarrollo evolutivo. Este enfoque entrelaza las actividades de especificación, desarrollo y validación. Un sistema inicial se desarrolla rápidamente a partir de especificaciones abstractas. Éste se refina basándose en las peticiones del cliente para producir un sistema que satisfaga sus necesidades.

- Ingeniería del software basada en componentes. Este enfoque se basa en la existencia de un número significativo de componentes reutilizables. El proceso de desarrollo del sistema se enfoca en integrar estos componentes en el sistema, más que en desarrollarlos desde cero.

Estos modelos generales no son descripciones definitivas de los procesos del software. Más bien, son abstracciones de los procesos que se pueden utilizar para explicar diferentes enfoques para el desarrollo de software. Estos modelos pueden ser extendidos y adaptados para crear procesos más específicos y adecuados para un proyecto en particular.

Estos tres modelos de procesos genéricos se utilizan ampliamente en la práctica actual de la ingeniería del software, no se excluyen mutuamente y a menudo se utilizan juntos, especialmente para el desarrollo de sistemas grandes. Cabe mencionar que los subsistemas, dentro de un sistema más grande, pueden ser desarrollados utilizando enfoques diferentes, por lo tanto, aunque es conveniente estudiar estos modelos separadamente, debe entenderse que en la práctica a menudo se combinan. Se han propuesto todo tipo de variantes de estos procesos genéricos y pueden ser usados en algunas organizaciones. La variante más importante es probablemente el desarrollo formal de sistemas, donde se crea un modelo formal matemático de un sistema. Este modelo se transforma entonces, usando transformaciones matemáticas que preservan su consistencia en código ejecutable.

3.2.1 El modelo en cascada

El primer modelo de proceso de desarrollo de software que se publicó se derivó de procesos de ingeniería de sistemas más generales. Debido a la cascada de una fase a otra, dicho modelo se conoce como modelo en cascada o como ciclo de vida del software. Las principales etapas de este modelo se transforman en actividades fundamentales de desarrollo (figura 3.3):

Análisis y definición de requerimientos. Los servicios, restricciones y metas del sistema se definen a partir de las consultas con los usuarios. Estos requerimientos se definen en detalle y sirven como base del sistema.

Diseño del sistema y del software. El proceso de diseño del sistema divide los requerimientos en sistemas hardware o software. Establece una arquitectura completa del sistema. El diseño del software identifica y describe las abstracciones fundamentales del sistema software y sus relaciones.

Implementación y prueba de unidades. Durante esta etapa, el diseño del software se lleva a cabo como un conjunto o unidades de programas. La prueba de unidades implica verificar que cada una cumpla su especificación.
Integración y prueba del sistema. Los programas o las unidades individuales de programas se integran y prueban como un sistema completo para asegurar que se cumplan los requerimientos del software.

Funcionamiento y mantenimiento. Ésta es la fase más larga del ciclo de vida. El sistema se instala y se pone en funcionamiento práctico. El mantenimiento implica corregir errores no descubiertos en las etapas anteriores del ciclo de vida, mejorar la implementación de las unidades del sistema y resaltar los servicios del sistema una vez que se descubren nuevos requerimientos.

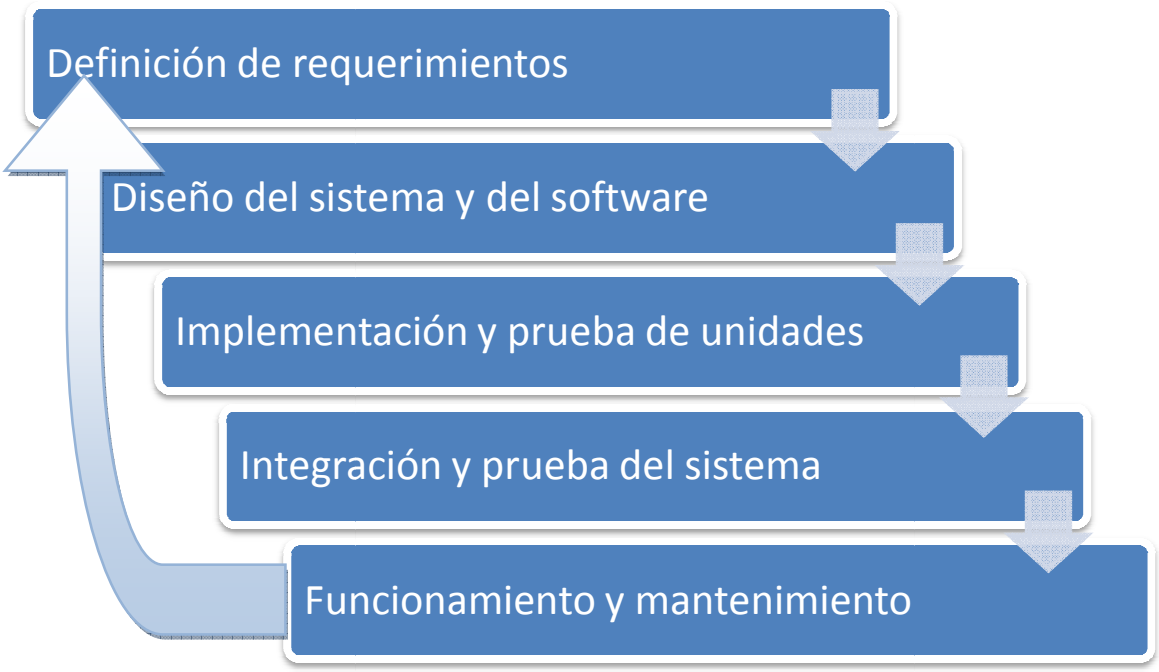


Figura 3.3 Modelo en cascada

Las ventajas del modelo en cascada son que la documentación se produce en cada fase y que éste cuadra con otros modelos del proceso de ingeniería. Su principal problema es su inflexibilidad al dividir el proyecto en distintas etapas. Se deben hacer compromisos en las etapas iniciales, lo que hace difícil responder a los cambios en los requerimientos del cliente. Por lo tanto, el modelo en cascada sólo se debe utilizar cuando los requerimientos se comprendan bien y sea improbable que cambien radicalmente durante el desarrollo del sistema.

3.2.2 Desarrollo evolutivo

El desarrollo evolutivo se basa en la idea de desarrollar una implementación inicial, exponiéndola a los comentarios del usuario y refinándola a través de las diferentes versiones hasta que se desarrolla un sistema adecuado (figura 3.4). Las actividades de especificación, desarrollo y validación se entrelazan en vez de separarse, con una rápida retroalimentación entre éstas.

Existen dos tipos de desarrollo evolutivo según Ian Sommerville dice:

- Desarrollo exploratorio, donde el objetivo del proceso es trabajar con el cliente para explorar sus requerimientos y entregar un sistema final. El desarrollo empieza con las partes del sistema que se comprenden mejor. El sistema evoluciona agregando nuevos atributos propuestos por el cliente.

- Prototipos desechables, donde el objetivo del proceso de desarrollo evolutivo es comprender los requerimientos del cliente y entonces desarrollar una definición mejorada de los requerimientos para el sistema. El prototipo se centra en experimentar todos los requerimientos del cliente que no se comprenden del todo.

En la producción de sistemas, un enfoque evolutivo para el desarrollo de software suele ser más efectivo que el enfoque en cascada, ya que satisface las necesidades inmediatas de los clientes. La ventaja de un proceso del software que se basa en un enfoque evolutivo es que la especificación se puede desarrollar de forma creciente. Tan pronto como los usuarios desarrollen un mejor entendimiento de su problema, éste se puede reflejar en el sistema software.

Pero desde una perspectiva de ingeniería y de gestión, el enfoque evolutivo tiene dos problemas:

- El proceso no es visible. Los administradores tienen que hacer entregas regulares para medir el progreso. Si los sistemas se desarrollan rápidamente, no es rentable producir documentos que reflejen cada versión del sistema.

- A menudo los sistemas tienen una estructura deficiente. Los cambios continuos tienden a corromper la estructura del software. Incorporar cambios en él se convierte cada vez más en una tarea difícil y costosa.

Los problemas del desarrollo evolutivo son más perjudiciales para sistemas grandes y complejos con un periodo de vida largo, donde diferentes equipos desarrollan distintas partes del sistema.

Para sistemas grandes, se recomienda un proceso mixto que incorpore las mejores características del modelo en cascada y del desarrollo evolutivo.

Las partes del sistema bien comprendidas se pueden especificar y desarrollar utilizando un proceso basado en el modelo en cascada.

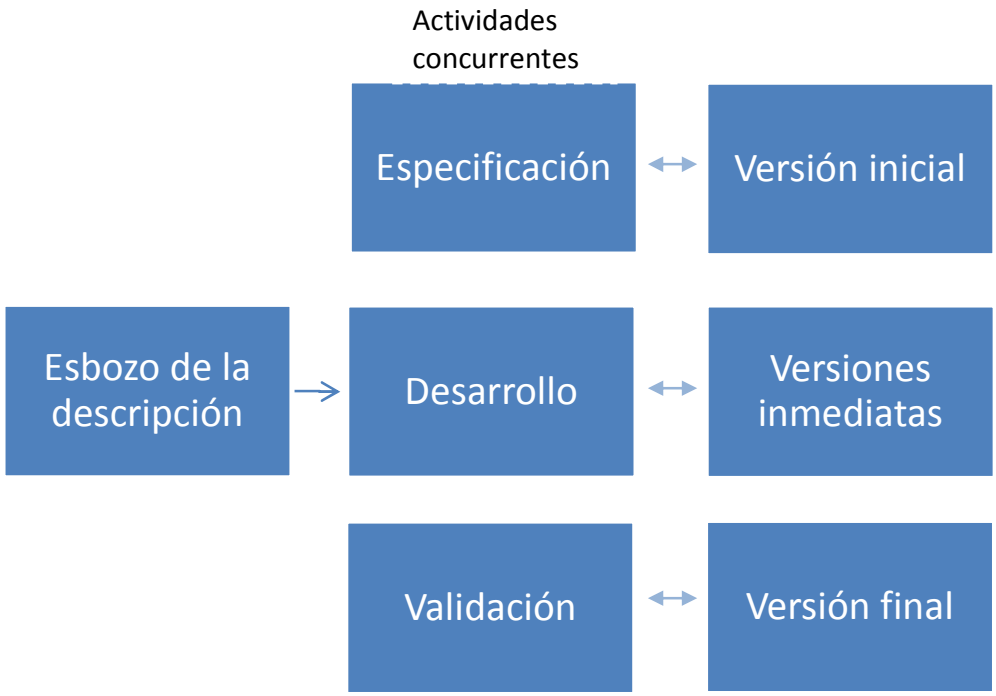


Figura 3.4 Desarrollo evolutivo

3.2.3 Ingeniería del software basada en componentes

En la mayoría de los proyectos de software existe algo de reutilización de software. Por lo general, esto sucede informalmente cuando las personas que trabajan en el proyecto conocen diseños o códigos similares al requerido. Los buscan, los modifican según lo creen necesario y los incorporan en el sistema.

Este enfoque basado en la reutilización, se compone de una gran base de componentes software reutilizables. Algunas veces estos componentes son sistemas por sí mismos que se pueden utilizar para proporcionar una funcionalidad específica, como dar formato al texto o efectuar cálculos numéricos. Aunque la etapa de especificación de requerimientos y la de validación son comparables con otros procesos, las etapas intermedias en el proceso orientado a la reutilización son diferentes.

Estas etapas son (figura 3.5):

- Análisis de componentes. Dada la especificación de requerimientos, se buscan los componentes para implementar esta especificación. Por lo general, no existe una concordancia exacta y los componentes que se utilizan sólo proporcionan parte de la funcionalidad requerida.

- Modificación de requerimientos. En esta etapa, los requerimientos se analizan utilizando información acerca de los componentes que se han descubierto. Estos componentes se modifican para reflejar los componentes disponibles. Si las modificaciones no son posibles, la actividad de análisis de componentes se puede llevar a cabo nuevamente para buscar soluciones alternativas.

- Diseño del sistema con reutilización. En esta fase se diseña o se reutiliza un marco de trabajo para el sistema. Los diseñadores tienen en cuenta los componentes que se reutilizan, y organizan el marco de trabajo para que los satisfaga.

- Desarrollo e integración. Para crear el sistema, el software que no se puede adquirir externamente se desarrolla, y los componentes y los sistemas comerciales se integran. En este modelo, la integración de sistemas es parte del proceso de desarrollo, más que una actividad separada.

La ingeniería del software basada en componentes tiene la ventaja de reducir la cantidad de software a desarrollarse y así reduce los costos y los riesgos. Por lo general, también permite una entrega más rápida del software. Sin embargo, los compromisos en los requerimientos son inevitables, y esto puede dar lugar a un sistema que no cumpla las necesidades reales de los usuarios. Más aún si las nuevas versiones de los componentes reutilizables no están bajo el control de la organización que los utiliza, se pierde parte del control sobre la evolución del sistema.

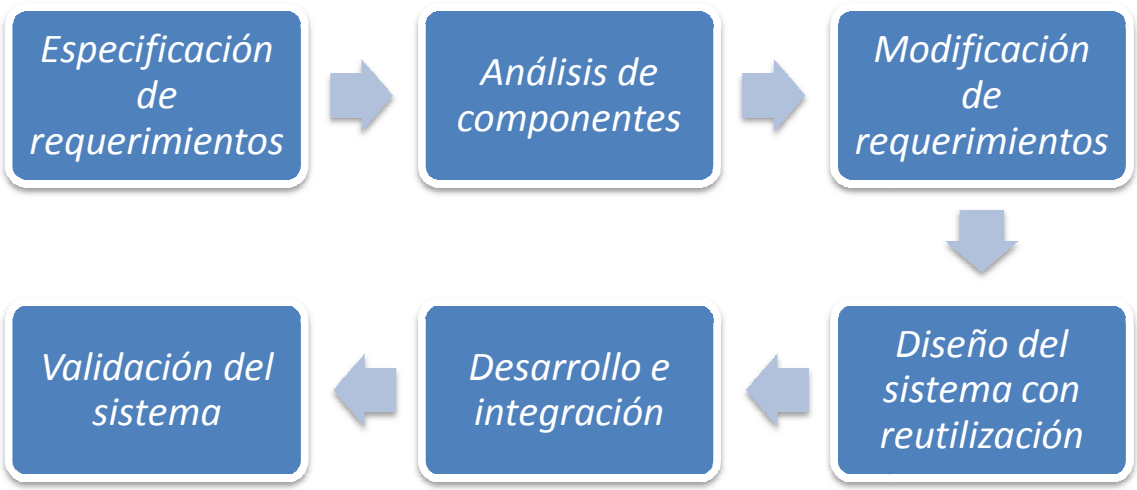


Figura 3.5 Etapas de la Ingeniería del software basada en componentes

3.3 Iteración de procesos

En todos los proyectos de software los cambios son inevitables. Los requerimientos del sistema cambian cuando se tienen que adaptar a las exigencias externas.

Cambian los diseños y la implementación cuando se dispone de nuevas tecnologías. El proceso del software no es un proceso único, puesto que las actividades del proceso se repiten regularmente conforme el sistema se rehace entendiendo a las peticiones de cambios.

Existen dos modelos de procesos que han sido diseñados para apoyar la iteración de procesos:

- Entrega incremental. La especificación, el diseño y la implementación del software se dividen en una serie de incrementos, los cuales se desarrollan por turnos.

- Desarrollo en espiral. El desarrollo del sistema gira en espiral hacia fuera, empezando con un esbozo inicial y terminando con el desarrollo final del mismo.

La esencia de los procesos iterativos es que la especificación se desarrolla junto con el software.

En el enfoque incremental, no existe una especificación completa del sistema hasta que el incremento final se especifica.

3.3.1 Desarrollo en espiral

El modelo en espiral (figura 3.6) fue propuesto originalmente por Boehm (Boehm, 1988). Más que representar el proceso del software como una secuencia de actividades retroalimentadas, se representa como una espiral. Cada ciclo en la espiral representa una fase del proceso del software. El ciclo interno se refiere a la viabilidad del sistema. El siguiente ciclo a la definición de requerimientos, el siguiente ciclo al diseño del sistema, y así sucesivamente.

Cada ciclo de la espiral se divide en cuatro sectores:

- Definición de objetivos. Para esta fase del proyecto se definen los objetivos específicos. Se identifican las restricciones del proceso, y se traza un plan detallado. Se identifican los riesgos del proyecto. Dependiendo de estos riesgos, se planean estrategias alternativas.

- Evaluación y reducción de riesgos. Se lleva a cabo un análisis detallado para cada uno de los riesgos del proyecto. Se definen los pasos para reducir dichos riesgo.

- Desarrollo y validación. Después de la evaluación de riesgos. se elige un modelo para el desarrollo del sistema.

- Planificación. El proyecto se revisa y se toma la decisión de si se debe continuar con un ciclo posterior de la espiral. Si se decide continuar, se desarrollan los planes para la siguiente fase del proyecto.

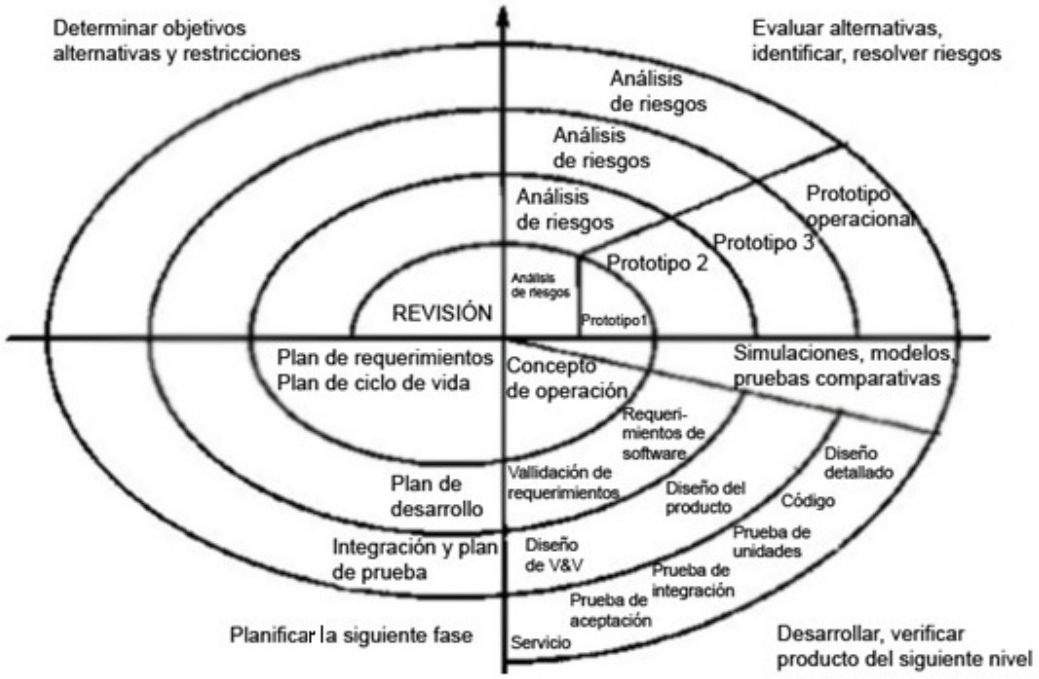


Figura 3.6 Desarrollo en espiral

La diferencia principal entre el modelo en espiral y los otros modelos del proceso del software es la consideración explícita del riesgo en el modelo en espiral. Informalmente, el riesgo significa sencillamente algo que puede ir mal. Los riesgos originan problemas en el proyecto, por lo tanto, la disminución de riesgos es una actividad muy importante en la gestión del proyecto.

Un ciclo de la espiral empieza con la elaboración de objetivos. Se enumeran formas alternativas de alcanzar estos objetivos y las restricciones impuestas en cada una de ellas. Cada alternativa se evalúa contra cada objetivo y se identifican las fuentes de riesgo del proyecto. El siguiente paso es resolver estos riesgos mediante actividades de recopilación de información como la de detallar más el análisis, la construcción de prototipos y la simulación. Una vez que se han evaluado los riesgos, se lleva a cabo cierto desarrollo seguido de una actividad de planificación para la siguiente fase del proceso.

3.3.2 Entrega incremental

La entrega incremental (figura 3.7) es un enfoque intermedio Entre el modelo en cascada y el modelo evolutivo, este modelo combina las ventajas de estos modelos. En un proceso de desarrollo incremental, los clientes identifican a grandes rasgos los servicios que proporcionará el sistema. Identifican qué servicios son más importantes y cuáles

menos. Se definen varios incrementos en donde cada uno proporciona un subconjunto de la funcionalidad del sistema. La asignación de servicios a los incrementos, depende de la prioridad del servicio con los servicios de prioridad más alta entregados primero.

Una vez que los incrementos del sistema se han identificado, los requerimientos para los servicios que se van a entregar en el primer incremento se definen en detalle, y éste se desarrolla.

Una vez que un incremento se completa y entrega, los clientes pueden ponerlo en servicio.

Esto es, que se entrega tempranamente la parte de la funcionalidad del sistema. Esto nos ayuda a clarificar sus requerimientos para los incrementos posteriores, pues se puede experimentar con el sistema. Tan pronto como se completan los nuevos incrementos, se integran en los existentes, de tal forma que la funcionalidad del sistema mejora con cada incremento entregado.

Las ventajas de este proceso de desarrollo incremental son:

- Los clientes no tienen que esperar hasta que el sistema completo se entregue para sacar provecho de él. El primer incremento satisface los requerimientos más críticos de tal forma que pueden utilizar el software inmediatamente.

- Los clientes pueden utilizar los incrementos iniciales como prototipos y obtener experiencia sobre los requerimientos de los incrementos posteriores del sistema.

- Existe un bajo riesgo de un fallo total del proyecto. Aunque se pueden encontrar problemas en algunos incrementos, lo normal es que el sistema se entregue de forma satisfactoria al cliente.

- Puesto que los servicios de más alta prioridad se entregan primero, y los incrementos posteriores se integran en ellos, es inevitable que los servicios más importantes del sistema sean a los que se les hagan más pruebas. Esto significa que es menos probable que los clientes encuentren fallos de funcionamiento del software en las partes más importantes del sistema.

No obstante existen algunos problemas en el desarrollo incremental. Los incrementos deben ser pequeños (no más de 20.000 líneas de código) y cada uno debe entregar alguna funcionalidad del sistema. Puede ser difícil adaptar los requerimientos del cliente a incrementos de tamaño apropiado. Es difícil identificar los recursos comunes que requieren todos los incrementos por que los requerimientos no se definen en detalle hasta que un incremento se implementa.

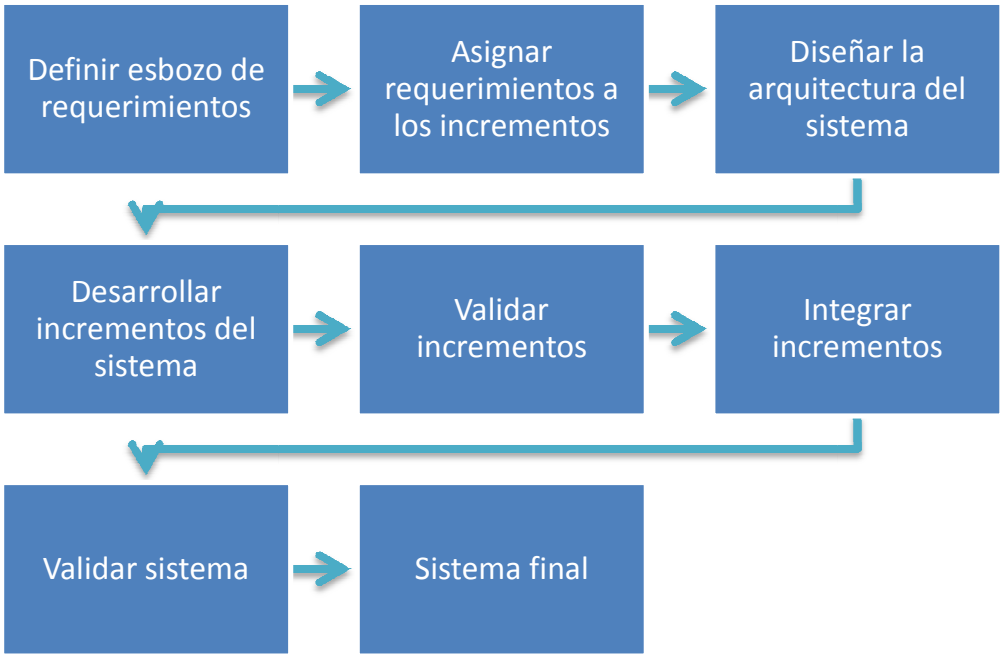


Figura 3.7 Entrega incremental

Capítulo 4 Caso: Unidad de Servicios de Cómputo Académico.

- 4.1 Formulación del problema
- 4.2 Análisis del problema
- 4.3 Búsqueda de soluciones
- 4.4 Desarrollo del Sistema de Monitorización

4 CASO: UNIDAD DE SERVICIOS DE CÓMPUTO ACADÉMICO.

4.1 Formulación del problema

Los sistemas computacionales que operan hoy en día tienen una gran capacidad de procesamiento y manejan grandes cantidades de información, más aún los servidores y equipos de red, por lo cual son de suma importancia dentro de cualquier organización. Es prioridad que estos equipos operen siempre de la mejor manera y estén disponibles en todo momento, para que los usuarios tengan la información que soliciten en cualquier instante.

Una de las necesidades relevantes dentro de la Unidad de Servicios de Cómputo Académico es la medición de diferentes parámetros de red, así como la medición del desempeño y disponibilidad de los dispositivos de red y servidores que se manejan dentro de este organismo. Actualmente se realizan estas tareas de medición con alcances limitados provocando intermitencia en el servicio, lo que a su vez ocasiona que los usuarios y el prestigio de la organización sean los más afectados.

Con esta tesis se tiene la oportunidad de mejorar el proceso de monitorización de los diferentes servicios prioritarios e institucionales que se otorgan. Por lo cual, se requiere mostrar los datos de los equipos y de la red a los administradores de manera eficaz para que ellos, si es el caso, sean alertados y tengan conocimiento en qué momento el equipo ha fallado, además conocer el comportamiento de los sistemas bajo ciertas condiciones, para que posteriormente se realice un análisis y una planeación a futuro basándose en el desempeño de los equipos. Aunado a esto, es necesario anticiparse a los problemas que puedan surgir para tener una reacción preventiva basada en la vigilancia constante.

4.2 Análisis del problema

Algunos de los aspectos que necesitan ser revisados periódicamente en los servidores son los siguientes:

- Los usuarios conectados al servidor
- El tiempo que ha estado activo
- Sus direcciones MAC
- Los puertos que tiene abiertos
- Los servicios que se tienen habilitados

- El porcentaje uso del procesador

- La cantidad de RAM, Disco Duro y Swap utilizada

También se necesita conocer el ancho de banda utilizado en cada puerto crítico de los switches con los que se cuenta. La mayor parte de estos equipos son marca 3Com.

La gran mayoría de los equipos que actúan como servidores en la Unidad de Servicios de Cómputo Académico operan con sistemas operativos Linux, en diversas distribuciones porque sus características principales se adecúan a las necesidades de esta unidad al ser multitarea, multiusuario y multiplataforma, además de ser un sistema operativo con gran estabilidad, buen manejo de sus recursos y robusto. También se cuenta con servidores Windows para aplicaciones especiales, mismas que son utilizadas por diferentes departamentos dentro de la Unidad.

Los usuarios de esta red son principalmente alumnos, académicos y administrativos de la Facultad, así como becarios de la Unidad, que son la base para el desarrollo de nuevas ideas y desarrollos en beneficio a la Facultad de Ingeniería.

Se pretende obtener información de los servidores para conocer el estado en el que se encuentran y tener una visión amplia a la hora de tomar decisiones, dependiendo del desempeño y las necesidades de éstos, es decir, si se necesita cambiar o mejorar el software o hardware del equipo.

Los administradores deben ser informados oportunamente del estado de los equipos para poder atender de forma rápida cualquier anomalía que pudiera afectar el desempeño o la disponibilidad de éstos.

Restricciones

- El manejo de la información debe ser fácil.

- Toda la información recopilada debe ser almacenada para tener estadísticas históricas del desempeño de los sistemas a monitorizar.

- Tener de forma organizada la información y disponible en cualquier momento y desde cualquier lugar.

- Por la forma en que está distribuida la carga de trabajo entre el personal y la forma en que está estructurada la administración de la red y los servidores, se requiere tener una monitorización centralizada.

- Se debe de monitorizar servidores con diversos sistemas operativos, ya sea basados en Windows Linux o Unix, pero sin comprometer la seguridad de la Unidad de Servicios de Cómputo Académico.

4.3 Búsqueda de soluciones

Como se mencionó, en el capítulo dos, hay una extensa lista de programas, utilerías y comandos que se pueden utilizar para la monitorización de la red y de servidores, pero no todas las expuestas en esta tesis serán de utilidad al momento de la monitorización de los recursos de la Unidad de Servicios de Cómputo Académico, ya que se deben de satisfacer las necesidades específicas de ésta.

Después de analizar las necesidades de la Unidad de Servicios de Cómputo Académico y hacer una evaluación de algunas de las mejores herramientas de monitorización que existen en el mercado decidimos utilizar sólo algunas de ellas ya que no todas estas aplicaciones cubren las necesidades de esta Unidad por completo. Por esta razón se tomó la decisión de diseñar un sistema que recopile información de los servidores de manera constante y a su vez integre el poder y las características de las herramientas existentes en el mercado, para así, tener un sistema completo que monitorice los servicios, tanto de switches como de servidores existentes en la Unidad de Servicios de Cómputo Académico.

Se eligieron sólo dos herramientas de la amplia grama existente, Cacti e Intelligent Management Center (IMC) de 3Com®.

Cacti nos permite manejar gráficas de interpretación simple sobre el ancho de banda consumido en la red, además nos proporciona una agradable interfaz gráfica, fácil de utilizar y configurar. Se puede utilizar para obtener información sobre los servidores y su estado, pero la forma en que nos despliega la información no es clara, por lo cual debe ser complementada con otras herramientas más poderosas.

Intelligent Management Center (IMC) de 3Com® nos ayudará a la monitorización de otros dispositivos y aspectos importantes de la red, esta herramienta tiene la capacidad de hacer un trazo detallado de la topología de la red que será de gran ayuda para encontrar rápidamente los dispositivos que presenten algún problema. Esta herramienta en específico es para la Unidad de Servicios de Cómputo Académico de gran utilidad ya que es compatible con todos los switches utilizados en la Facultad de Ingeniería que cuenta, en su mayor parte, con switches marca 3Com.

Ambas herramientas utilizan el protocolo SNMP en sus 3 versiones, en particular IMC usa otros protocolos como Secure Shell, Telnet o ping para detectar los dispositivos conectados a la red.

Aunque IMC es una herramienta que nos brinda mucha información detallada, es un sistema complejo de usar y configurar.

En la Unidad de Servicios de Cómputo Académico se necesita un dispositivo que pueda obtener la información que nos brindan estas herramientas y además proporcione información sobre los servidores, pero de una forma clara y fácil de entender para que los administradores de red y de los servidores tengan los datos necesarios para prevenir o atender cualquier incidente que se llegara a presentar.

Para esto se requiere diseñar un sistema de monitorización adecuado a las necesidades de la Unidad que recopile información de manera centralizada, además de integrar las herramientas de monitorización antes mencionadas en conjunto con un motor SNMP para obtener la información más relevante de los servicios de red y servidores que maneja la Unidad de Servicios de Cómputo Académico. Al anterior sistema lo hemos denominado SIMON.

Así mismo, es necesario que este sistema cuente con módulos para alertar a los usuarios cuando algo esté fallando en los dispositivos de red o en algún servidor, para que se tomen las medidas necesarias oportunamente. Este sistema necesita además un módulo de reportes que nos proporcione la información de un determinado evento para tener un mejor control sobre los dispositivos.

4.3.1 Diagramas UML

Una vez vistos los requerimientos y restricciones para el desarrollo del sistema de monitorización, se muestran los diagramas UML que sirven como base para la creación de dicho sistema. Estos diagramas ayudan a ejemplificar de una manera simple y clara las funciones que realiza SIMON.

El Lenguaje Unificado Modelado es una herramienta efectiva para que tanto el programador como el cliente estén hablando el mismo idioma para la creación de sistemas (Schmuller, 2001). Permite a los desarrolladores generar diseños que capturen sus ideas en una forma convencional y fácil de comprender para otras personas. UML está compuesto por diversos elementos gráficos que se combinan para conformar diagramas. Debido a que UML es un lenguaje, cuenta con reglas para combinar tales elementos. A continuación se muestra el diagrama de estados (figura 4.1) y los diagramas de casos de uso (figura 4.2) para SIMON.

Diagrama de Estados

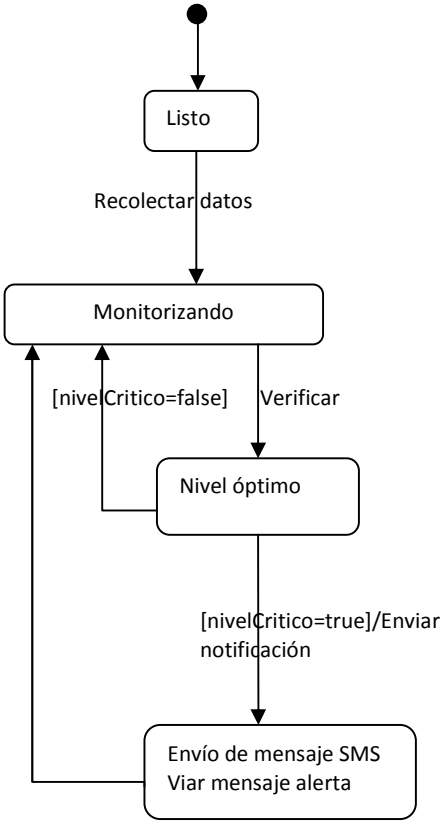


Figura 4.1 Diagrama de Estados de SIMON

Al entrar al sistema se tiene el estado “Listo”, es decir, el punto inicial del sistema. Se cuenta con un script que se ejecuta cada intervalo de tiempo para recolectar los datos de los servidores y switches. Al ser procesados estos datos, el sistema se encuentra en un estado “monitorizando” para, posteriormente verificar si los dispositivos se encuentran en un “nivel óptimo”. Si llegara a ser el caso, el sistema regresará al estado “Monitorizando”, pero si alcanza el nivel crítico de un dispositivo, inmediatamente se envía un mensaje SMS de alerta al administrador de dicho equipo. Al final el sistema vuelve a monitorizar los dispositivos.

Diagramas de caso de uso

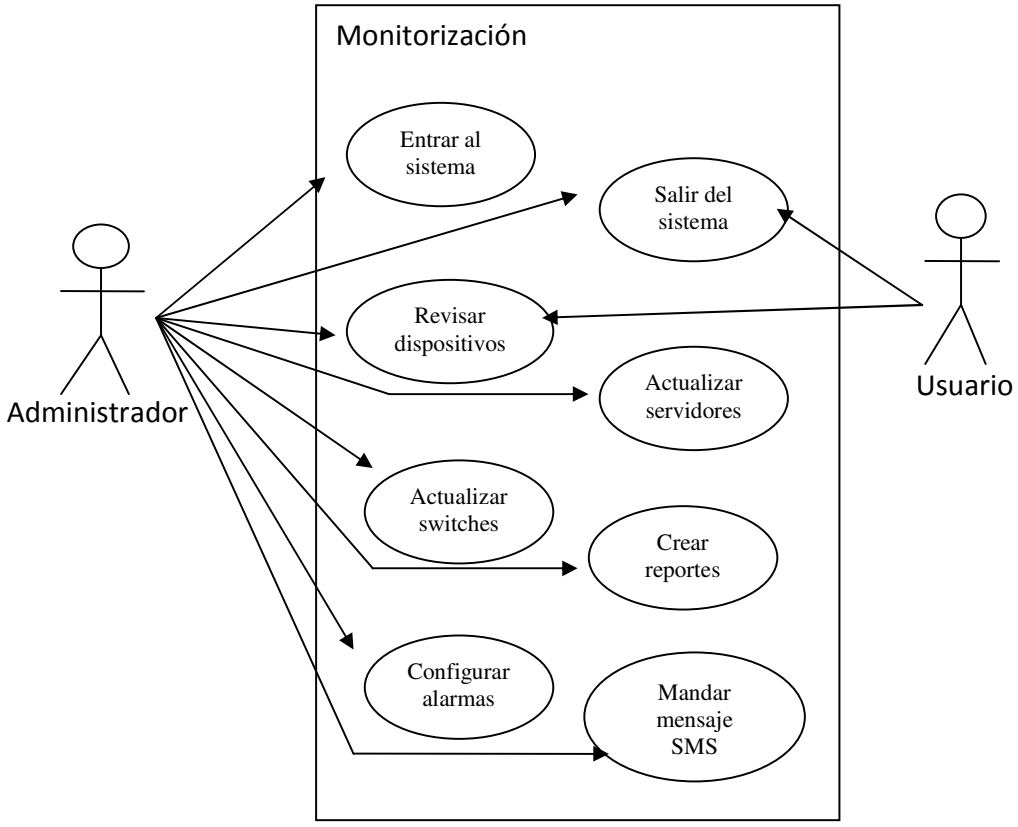
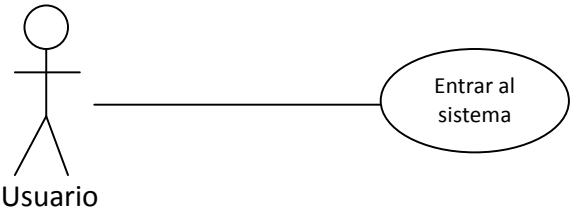


Figura 4.2 Diagrama de casos de uso de SIMON

Caso de uso: Entrar al sistema

Actor: Usuario



Descripción: El usuario debe teclear su clave de acceso para entrar al sistema

Precondiciones:

- El usuario entra a la página del sistema
- Necesita revisar los dispositivos o crear reportes

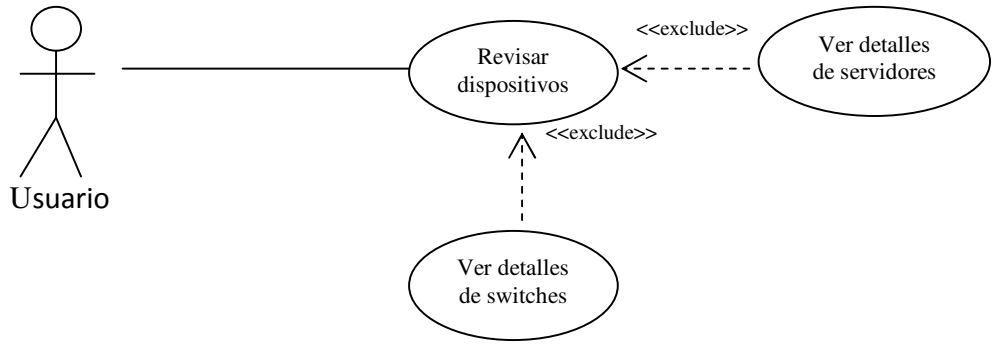
Flujo:

ACTOR		SISTEMA		
Paso	**Acción**	**Paso**	**Acción**	**Excepción**
1	Entra a la página del sistema	2	Despliega la interface para la introducción de la clave de acceso	
3	Teclea clave de acceso			
4	Envía datos	5	Valida la clave de acceso	E1
		6	Muestra la pantalla principal del sistema	

Excepciones:

ID	Nombre	Acción
E1	Clave de acceso incorrecto (no existe o nula)	El sistema informa el error en la clave y permite nuevamente su introducción

Caso de uso: Revisar dispositivos
Actor: Usuario



Descripción: El usuario puede revisar el estado de los dispositivos incluyendo ver las gráficas de los mismos.

Precondición:

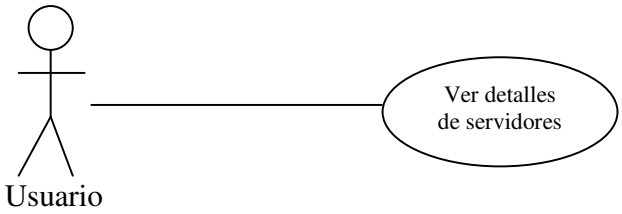
- El usuario desea presentar un informe de los dispositivos monitorizados mediante un reporte
- Necesita conocer el estado en el que se encuentran los dispositivos.

Flujo:

ACTOR		SISTEMA		
Paso	**Acción**	**Paso**	**Acción**	**Excepción**
1	El usuario elige la opción de “servidores” o “switches” del menú principal	2	El sistema despliega la interfaz correspondiente a la opción seleccionada	

Caso de uso: Ver detalles de servidores

Actor: Usuario



Descripción: El usuario desea saber el comportamiento de cada servidor.

Precondición: El usuario hace una verificación de rutina.

Flujo:

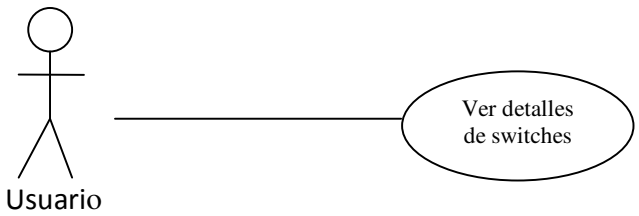
ACTOR		SISTEMA		
Paso	**Acción**	**Paso**	**Acción**	**Excepción**
1	Selecciona el servidor de la lista mostrada	2	Dependiendo el servidor seleccionado, se muestran sus datos.	E1

Excepciones:

ID	Nombre	Acción
E1	Botón Cancelar o cerrar	El sistema regresa a la pantalla del menú principal

Caso de uso: Ver detalles de switches

Actor: Usuario



Descripción: El usuario desea saber el comportamiento de cada switch.

Precondición: El usuario hace una verificación de rutina.

Flujo:

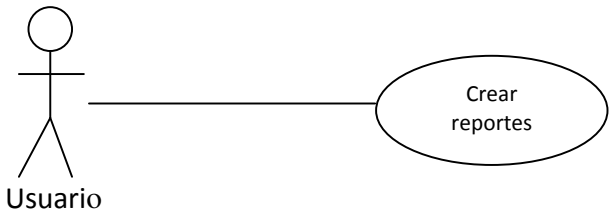
ACTOR		SISTEMA		
Paso	**Acción**	**Paso**	**Acción**	**Excepción**
1	Selecciona el switch de la lista mostrada	2	Dependiendo el switch seleccionado, se muestran sus datos.	E1

Excepciones:

ID	Nombre	Acción
E1	Botón Cancelar o cerrar	El sistema regresa a la pantalla del menú principal

Caso de uso: Crear reportes

Actor: Usuario



Descripción: El usuario tiene la necesidad de crear un reporte de cada dispositivo.

Precondición:

- Necesita entregar un reporte periódicamente de cada dispositivo

- El usuario necesita comprobar el desempeño de un dispositivo en particular.

Flujo:

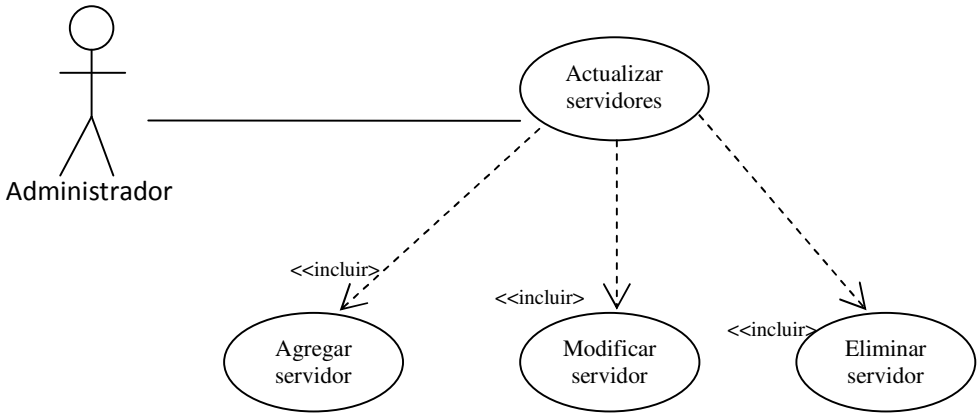
ACTOR		SISTEMA		
Paso	**Acción**	**Paso**	**Acción**	**Excepción**
1	El usuario selecciona el botón "generar reporte" del dispositivo presentado en pantalla	2	Genera la salida de un archivo con la información del dispositivo.	E1
3	Verifica la información mostrada en el archivo creado			
		4	Se regresa a la pantalla del dispositivo	

Excepciones:

ID	Nombre	Acción
E1	Botón Cancelar o cerrar	El sistema regresa a la pantalla del dispositivo

Caso de Uso: Actualizar servidores

Actor: Administrador



Descripción: El administrador tiene la necesidad de actualizar su catalogo de servidores, donde puede agregar un servidor, modificarlo o eliminarlo.

Precondición:

- El administrador desea monitorizar un servidor nuevo que aun no está dado de alta en el sistema
- El administrador necesita modificar algunos datos asociados a un servidor.

- Un servidor ya no es más útil y necesita darlo de baja.

Flujo:

ACTOR		SISTEMA		
Paso	**Acción**	**Paso**	**Acción**	**Excepción**
1	Selecciona alguna de las opciones: “alta de servidor”; “Actualización de servidor” o “Baja de servidor”	2	El sistema despliega la interfaz correspondiente a la opción seleccionada	

Caso de Uso: Agregar servidores

Actor: Administrador

Descripción: El administrador dará de alta un nuevo servidor para monitorizarlo.

Precondición:

- El administrador desea monitorizar un servidor nuevo que la Unidad ha adquirido recientemente y que no está dado de alta en el sistema

Flujo:

ACTOR		SISTEMA		
Paso	**Acción**	**Paso**	**Acción**	**Excepción**
		1	Se despliegan los campos que se deben de llenar	E1
2	Llena los campos correspondientes			E1
3	Guarda los datos	4	Almacena el nuevo servidor	
		5	Regresa a la pantalla de servidores	

Excepciones:

ID	Nombre	Acción
E1	Botón Cancelar o cerrar	El sistema regresa a la pantalla de servidores

Caso de Uso: Modificar servidores

Actor: Administrador

Descripción: El administrador modifica datos de un servidor.

Precondición:

- El administrador tiene la necesidad de cambiar datos de los servidores

Flujo:

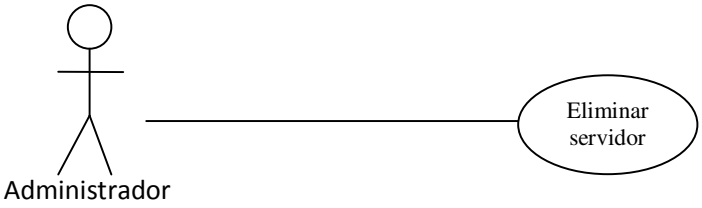
ACTOR		SISTEMA		
Paso	**Acción**	**Paso**	**Acción**	**Excepción**
1	Selecciona el servidor a modificar	2	Se despliegan los campos que se deben de llenar	E1
3	Llena los campos correspondientes			E1
4	Guarda los datos			E1
		5	Almacena los cambios	
		6	Regresa a la pantalla de servidores	

Excepciones:

ID	Nombre	Acción
E1	Botón Cancelar o cerrar	El sistema regresa a la pantalla de servidores

Caso de Uso: Eliminar servidores

Actor: Administrador



Descripción: El administrador elimina un servidor.

Precondición:

- El administrador ya no tiene la necesidad de monitorizar un servidor por obsoleto

Flujo:

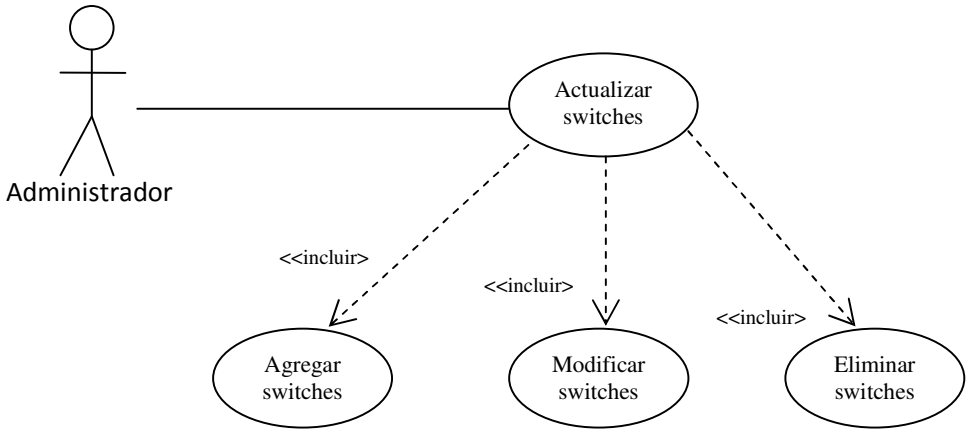
ACTOR		SISTEMA		
Paso	**Acción**	**Paso**	**Acción**	**Excepción**
1	Selecciona el servidor a eliminar	2	Pregunta si realmente se va a eliminar ese servidor	E1
3	Acepta la eliminación			E1
		4	Elimina el servidor	
		5	Notificación de servidor eliminado	
		6	Regresa a la pantalla de servidores	

Excepciones:

ID	Nombre	Acción
E1	Botón Cancelar o cerrar	El sistema regresa a la pantalla de servidores

Caso de Uso: Actualizar switches

Actor: Administrador



Descripción: El administrador tiene la necesidad de actualizar su catálogo de servidores, donde puede agregar un servidor, modificarlo o eliminarlo.

Precondición:

- El administrador desea monitorizar un switch nuevo que aun no está dado de alta en el sistema
- El administrador necesita modificar algunos datos asociados a un switch.
- Un switch ya no es más útil y necesita darlo de baja.

Flujo:

ACTOR		SISTEMA		
Paso	**Acción**	**Paso**	**Acción**	**Excepción**
1	Selecciona alguna de las opciones: “alta de switch”; “Actualización de switch” o “Baja de switch”	2	El sistema despliega la interfaz correspondiente a la opción seleccionada	

Caso de Uso: Agregar switches

Actor: Administrador

Descripción: El administrador dará de alta un nuevo switch para monitorizarlo.

Precondición:

- El administrador desea monitorizar un switch nuevo que la Unidad ha adquirido recientemente y que no está dado de alta en el sistema

Flujo:

ACTOR		SISTEMA		
Paso	**Acción**	**Paso**	**Acción**	**Excepción**
		1	Se despliegan los campos que se deben de llenar	E1
2	Llena los campos correspondientes			E1
3	Guarda los datos	4	Almacena el nuevo switch	
		5	Regresa a la pantalla de switches	

Excepciones:

ID	Nombre	Acción
E1	Botón Cancelar o cerrar	El sistema regresa a la pantalla de switches

Caso de Uso: Modificar switch

Actor: Administrador

Descripción: El administrador modifica datos de un switch.

Precondición:

- El administrador tiene la necesidad de cambiar datos de los switches

Flujo:

ACTOR		SISTEMA		
Paso	**Acción**	**Paso**	**Acción**	**Excepción**
1	Selecciona el switch a modificar	2	Se despliegan los campos que se deben de llenar	E1
3	Llena los campos correspondientes			E1
4	Guarda los datos			E1
		5	Almacena los cambios	
		6	Regresa a la pantalla de switches	

Excepciones:

ID	Nombre	Acción
E1	Botón Cancelar o cerrar	El sistema regresa a la pantalla de switches

Caso de Uso: Eliminar switch

Actor: Administrador

Descripción: El administrador elimina un switch.

Precondición:

- El administrador ya no tiene la necesidad de monitorizar un switch por obsoleto

Flujo:

ACTOR		SISTEMA		
Paso	**Acción**	**Paso**	**Acción**	**Excepción**
1	Selecciona el swich a eliminar	2	Pregunta si realmente se va a eliminar ese switch	E1
3	Acepta la eliminación			E1
		4	Elimina el switch	
		5	Notificación deswitch eliminado	
		6	Regresa a la pantalla de switches	

Excepciones:

ID	Nombre	Acción
E1	Botón Cancelar o cerrar	El sistema regresa a la pantalla de switches

Caso de Uso: Configurar alarmas

Actor: Administrador

Descripción: El administrador configura los parámetros para generar las alarmas.

Precondición:

- El administrador necesita modificar los niveles a los cuales se lanzará la alarma.

Flujo:

ACTOR		SISTEMA		
Paso	**Acción**	**Paso**	**Acción**	**Excepción**
1	Selecciona el dispositivo configurar	2	Se despliegan los campos que se deben de llenar	E1
3	Llena los campos correspondientes			E1
4	Guarda los datos			
		5	Almacena los cambios	
		6	Regresa a la pantalla inicial	

Excepciones:

ID	Nombre	Acción
E1	Botón Cancelar o cerrar	El sistema regresa a la pantalla inicial

Caso de Uso: Mandar mensajes SMS

Actor: Administrador

Descripción: El administrador envía mensajes SMS a los administradores de los equipos.

Precondición:

- El administrador necesita avisar de un evento o suceso a los administradores de forma rápida.

Flujo:

ACTOR		SISTEMA		
Paso	**Acción**	**Paso**	**Acción**	**Excepción**
1	Selecciona el grupo de administradores	2	Se despliegan los campos que se deben de llenar	E1
3	Teclea el contenido del mensaje			E1
4	Presiona botón enviar			
		5	Envía el o los mensajes	
		6	Regresa a la pantalla inicial	

Excepciones:

ID	Nombre	Acción
E1	Botón Cancelar o cerrar	El sistema regresa a la pantalla inicial

4.4 Desarrollo del software de monitorización

4.4.1 Software necesario

- **Sistema operativo**

El sistema operativo que se eligió para desarrollar el sistema de monitorización fue Linux, ya que por sus características descritas en el capítulo 1 hacen que sea la plataforma ideal para el correcto desarrollo de SIMON, debido a que es posible instalar aplicaciones web, bases de datos y protocolos necesarios para el sistema, además de tener características similares al del resto de los servidores con los que cuenta la unidad, de esta forma el personal de UNICA puede familiarizarse con la administración del servidor de monitorización.

- **Manejador de bases de datos**

La cantidad de información que el motor SNMP recolecta crece a medida que se van agregando más equipos, por lo que se necesita un buen manejador de bases de datos capaz de llevar este ritmo de trabajo. Según la página oficial de MySQL, se puede operar con más de 50 millones de registros, proporciona rapidez en las consultas, seguridad, soporte en grandes bases de datos y conectividad con otras aplicaciones web como PHP, por lo que fue elegido para trabajar en el sistema SIMON.

- **Lenguaje de programación**

PHP es un lenguaje de programación, diseñado originalmente para la creación de páginas web dinámicas y puede combinarse con el lenguaje HTML[16]. Tiene la ventaja de que es interpretado por el servidor web y convertido a HTML para que los navegadores lo desplieguen. Además, cuenta con acceso a información almacenada en una base de datos. Se puede conectar con la mayoría de los manejadores de bases de datos, destacando MySQL y PostgreSQL, convirtiéndolo en una buena opción para desarrollar los scripts para recuperar la información de los dispositivos como servidores y switches.

- **Servidor web**

SIMON necesita una interfaz web para que los usuarios, a los cuales está dirigido el sistema, puedan visualizar los datos de los dispositivos y para cubrir esta necesidad se

[16] HyperText Markup Language - Lenguaje de Marcado de Hipertexto.

requiere de un servidor web. Apache, es un servidor web que implementa el protocolo HTTP para las plataformas Linux, Unix, MacOS y Windows. Apache, es usado para muchas otras tareas donde el contenido necesita ser puesto a disposición en una forma segura y confiable. Presenta módulos para conexión con PHP, lenguaje que se utiliza en SIMON, por lo que se decidió a usar este servidor web.

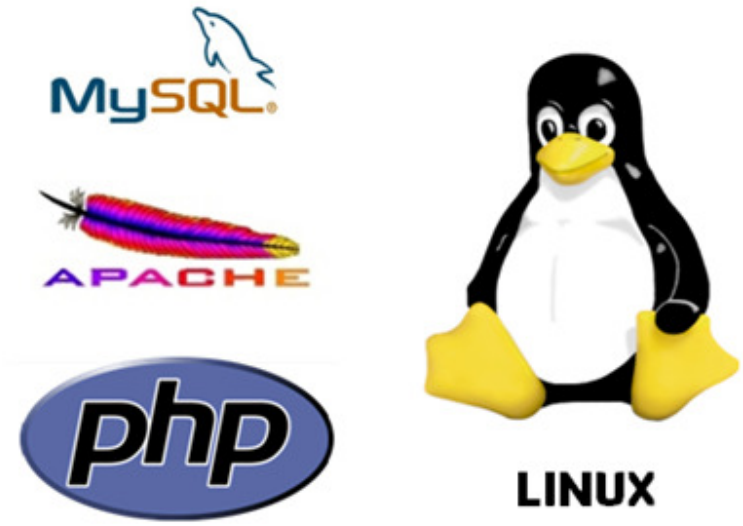


Figura 4.3 Software usado en SIMON

El equipo donde está alojado SIMON está diseñado para ser un servidor web dinámico de bajo costo. La combinación de Linux, Apache, MySQL y PHP definen la base de este servidor (figura 4.3). Sin embargo, SIMON se apoya en otros programas y aplicaciones que a continuación se describen.

4.4.2 Utilerías

- **GNOKII**

Gnokii, es un conjunto de herramientas de código abierto, con una serie de controladores para usar teléfonos celulares con distintos sistemas operativos, como Linux, Solaris, sistemas basados en BSD y Windows. La meta principal de Gnokii fue trabajar con equipos Nokia, pero actualmente hay más dispositivos de otras marcas que funcionan con esta aplicación. Incluso la versión 0.6.1 ofrece soporte para teléfonos con el sistema operativo Symbian. La mayoría de los celulares Nokia, desde la serie 3110/3180, 5110/6110, 7110/6210 y 6510/6310 son soportados. En las primeras etapas de desarrollo de Gnokii se especulaba que la empresa Nokia ofrecería soporte al proyecto, desafortunadamente no hubo un acuerdo en cuanto el uso de binarios y el uso de código

abierto por lo que la aplicación continuó con o sin el apoyo de Nokia. El proyecto original se creó debido a algunas discusiones entre Francois Dessart, Hugh Blemings y otros. Su principal meta era desarrollar un remplazo para el software de datos de Nokia, el cual debía ejecutarse bajo Linux. El proyecto comenzó en octubre de 1998 y produjo un código preliminar funcional en Linux. Un proyecto similar comenzó en manos de Staffan Ulfberg para desarrollar software para el Nokia 6110 y modelos similares. En febrero de 1999, los dos proyectos combinados formaron el actual proyecto Gnokii (Gnokii, 2009).

Gnokii puede ser usado desde la línea de comandos para el envío de mensajes SMS mediante el siguiente comando:

Shell> echo "el contenido de mensaje " | gnokii –sendsms 1234567890

Se utilizó la aplicación Gnokii junto con un teléfono celular marca Nokia modelo 6131 (figura 4.4) para el módulo de envío de alarmas por medio de mensajes de texto, cubriendo así uno de los requerimientos primordiales de este sistema, alertar a los administradores de fallos críticos en los dispositivos de manera inmediata.

Figura 4.4 Conexión del servidor de monitorización con celular Nokia 6131

- **XOOPS**

XOOPS es una herramienta utilizada en el SIMON porque es un sistema de administración de contenido poderoso, flexible y fácil de usar, que está hecho en el lenguaje PHP. Este sistema permite a los administradores manejar sitios web dinámicos, construir comunidades en línea, gestionar usuarios, modificar la estructura del sitio y proveer de contenido a través de una interfaz sencilla. XOOPS maneja toda la parte tediosa, para que los desarrolladores sólo se concentren en el contenido del sitio. XOOPS son las siglas en inglés de eXtensible Object Oriented Portal System (sistema de

portal extensible orientado a objetos). Comenzó como un sistema de portal; sin embargo, XOOPS se ha convertido en un sistema de gestión de contenido para la creación fácil de sitios web y se encuentra bajo los términos de la licencia pública general (GPL) y cualquier persona es libre para utilizarlo, modificarlo y redistribuirlo bajo los mismos términos de la GPL. Esta herramienta cuenta con diversos módulos como son noticias, foros, enlaces, descargas, documentos, etc. y pueden ser instalados, desinstalados, activados o desactivados con un simple click usando el sistema de administración de módulos de XOOPS (Xoops, Soporte oficial en español, 2009). Es un sistema de gran alcance y de uso fácil para asignar permisos que permitan fijar a los administradores permisos a ciertos grupos de usuarios. SIMON utiliza a XOOPS como el armazón para mostrar la información y administrar de forma segura las sesiones de los usuarios que pueden tener acceso al sistema. De esta manera se ahorran pasos en la creación del proyecto, haciendo que los recursos tanto humanos como de tiempo se enfoquen directamente al desarrollo del sistema SIMON.

4.4.3 Funcionamiento de SIMON

Las infraestructuras de comunicaciones y los servidores de red como firewalls, servidores web, de correo y otros elementos existentes en la red, son piezas clave para el correcto funcionamiento de la organización. Es fundamental garantizar su funcionamiento en condiciones óptimas.

SIMON ofrece una solución integral de monitorización de servicios usando herramientas de software libre (Cacti) y software comercial (IMC) en ambiente Linux. La plataforma permite detectar rápidamente cuando se produce una incidencia en el servicio y notificar a la persona o personas apropiadas para su resolución, a través de la consola del sistema, mediante correo electrónico o directamente a un teléfono móvil mediante SMS.

La plataforma de monitorización se ejecuta en un sistema Linux y los sistemas monitorizados pueden ser Linux, Windows y otros sistemas UNIX, incluyendo equipo activo como lo son los switches. La monitorización de SIMON garantiza la máxima disponibilidad de los servidores y de los servicios de red en entornos donde se requiere una disponibilidad de 24x7.

El sistema SIMON se basa en la recolección de datos de los equipos a monitorizar, mediante una serie de scripts realizados en su mayoría en el lenguaje PHP. Estos scripts obtienen los datos necesarios de los equipos mediante el protocolo SNMP, para esto, PHP cuenta con funciones especiales para el manejo de este protocolo. Los scripts se ejecutan cada cierto tiempo para recolectar periódicamente la información, cada 5 minutos. Una vez obtenidos los datos, éstos se almacenan en la base de datos de MySQL, para después ser utilizados por los diferentes módulos con los que se cuentan y desplegarlos en una interfaz gráfica vía web.

Como se mencionó anteriormente, al sistema se incorporan CACTI e IMC de 3com®, para lo cual se muestran dentro de la interfaz gráfica. La figura 4.5 expone el funcionamiento general de SIMON.

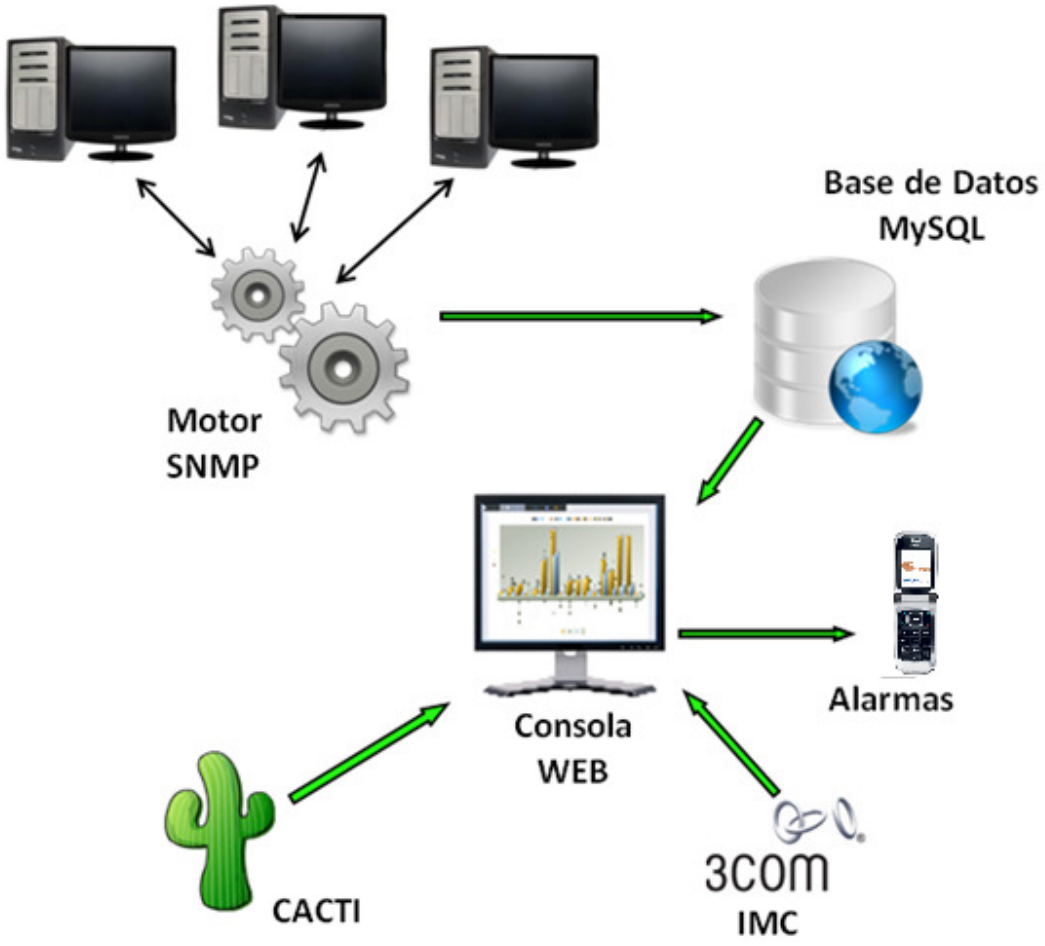


Figura 4.5 Funcionamiento general de SIMON

- **Prototipo 1**

Para desarrollar el sistema SIMON, fue necesario crear paso a paso diferentes módulos para así, cubrir las necesidades y objetivos planteados desde un inicio, por lo que el primer prototipo creado fue el motor que recolecta los datos de los servidores y switches para después, hacer una conexión a la base de datos y almacenarlos (figura 4.6)

Este motor está basado en una serie de scripts desarrollados en PHP. Su función es obtener los datos que se consideran esenciales para conocer el estado del equipo y así tomar acciones correspondientes. Entre los datos que se recopilan están: el tiempo activo del servidor, la dirección MAC, el ancho de banda de la tarjeta de red, los puertos TCP y UDP abiertos, los servicios que se están ejecutando, el número de usuarios conectados; así como el porcentaje de uso del CPU por parte de los usuarios y del sistema.
En el caso de los switches se obtienen el ancho de banda de entrada y salida de datos de los puertos en el switch.

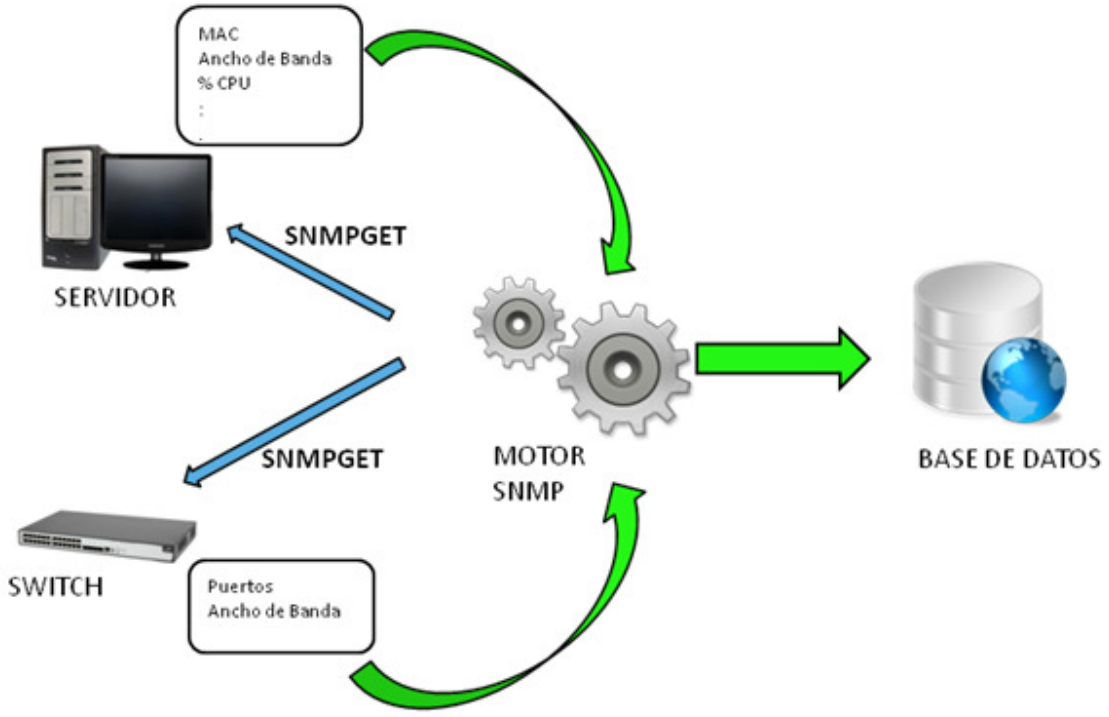


Figura 4.6 Funcionamiento del prototipo 1

La función PHP que se utiliza para obtener estos datos, tanto de los servidores como de los switches es:

string **snmpget**(string *hostname*, string *comunidad*, string *objetoId*);

donde:

hostname: es la dirección IP del host que cuenta con el agente snmp

comunidad: la comunidad snmp con permisos de lectura

objetoId: es el objeto OID del snmp.

Esta función es usada para leer el valor de un objeto snmp especificado por *objetoId,* la comunidad es necesaria ya que es un parámetro del protocolo snmp para poder leer el dato que se pide.

En este primer prototipo todos los datos fueron recopilados sin mayor problema. Sin embargo, en algunos casos hubo que hacer ciertas conversiones y adaptaciones, ya que snmp devuelve los datos en unidades poco convenientes para la aplicación. De esta manera, los datos ya procesados fueron almacenados en la base de datos.

La base de datos que se utiliza es una base relacional, y se planteo el Diagrama Entidad Relación para el almacenamiento de los datos de la aplicación (figura 4.7)

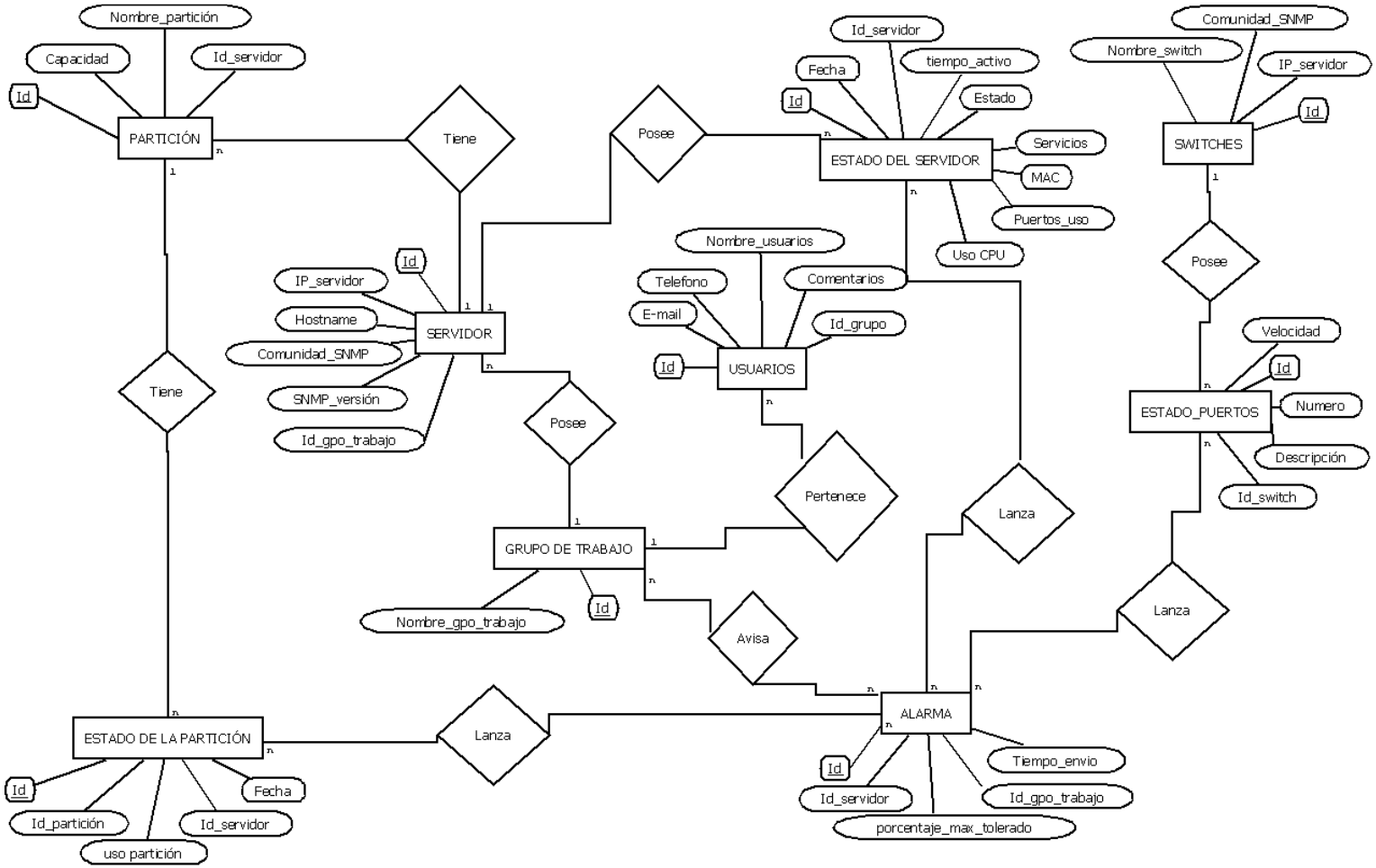


Figura 4.7 Diagrama Entidad Relación

La base de datos está conformada como lo muestra la figura 4.8

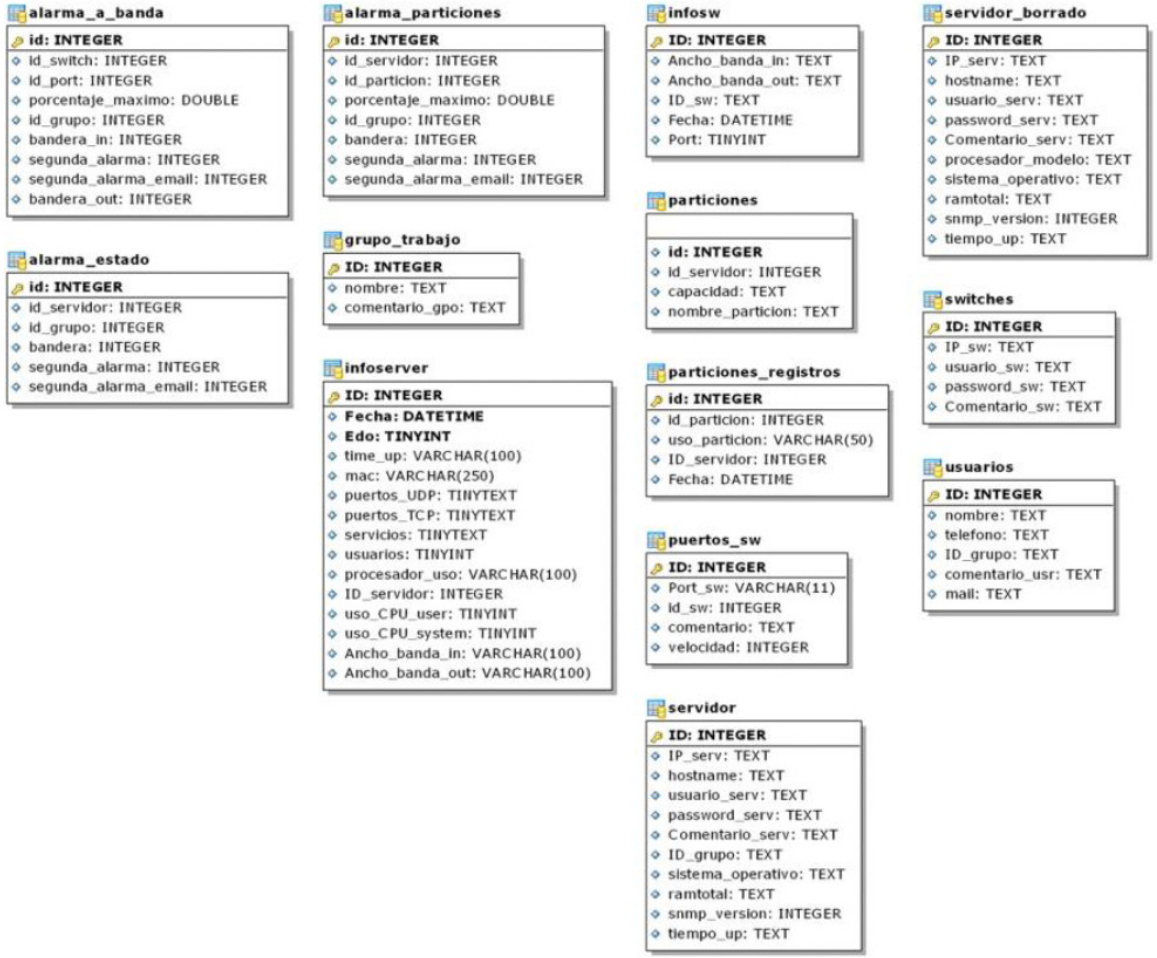


Figura 4.8 Tablas de la Base de Datos

- **Prototipo 2**

Con la primera parte se tienen los datos ya almacenados en la base de datos, pero éstos no son de utilidad si no se les da un buen manejo para mostrarlos de una manera clara y sencilla para el administrador.

En el prototipo 2 se diseña y desarrolla la interfaz gráfica con la ayuda de la aplicación XOOPS. La administración y configuración de usuarios para ingresar al sistema la proporciona XOOPS. La figura 4.9 muestra la pantalla de la aplicación en su estado original, tal y como es una vez instalada la aplicación. La figura 4.10 muestra la misma aplicación después de haber hecho el diseño para SIMON

Figura 4.9 Pantalla inicial de XOOPS

Figura 4.10 Pantalla inicial del sistema SIMON

La autenticación para el ingreso al sistema se realiza con la ayuda de un módulo de seguridad precargado que se encarga de proteger y prevenir ataques. Se pueden declarar IP's que se sabe son confiables y no representan un peligro, por ejemplo, un rango de direcciones que se encuentran dentro de nuestra red y protegidas por un firewall. Se puede fijar la cantidad de veces que un usuario intenta darse de alta en 10 minutos, si no lo consigue en esa cantidad de ocasiones, su IP se bloquea durante cierto tiempo para prevenir ataques de fuerza bruta[17]. Para prevenir ataques de denegación de servicios se ajusta un límite de recargas de la página y el tiempo que se vigila para recargas frecuentes y así ser considerado como un ataque malicioso. También cuenta con protección contra travesías de directorio, ya que elimina la declaración **".."** (subir nivel) de todas las peticiones que intenten moverse entre las carpetas. Cuenta con métodos para evitar la inyección de código SQL y combatir el spam.

Se diseñó una página principal donde se muestran los datos del perfil del usuario, el tipo de permiso que tiene y además la opción para cambiar los datos personales de la cuenta con la que se está teniendo acceso al sistema. Se cuenta con 2 tipos de privilegios para los usuarios, el usuario *monitor* que sólo puede observar los equipos y sus gráficas, así como los históricos de los mismos y acceder a las otras herramientas de monitorización, también puede enviar mensajes SMS y crear reportes, y el usuario *administrador* que puede hacer cambios en los dispositivos, como editar, borrar y agregar servidores o switches para tenerlos monitorizando con SIMON. También puede enviar mensajes SMS directamente a grupos de administradores en caso de que lo requiera. Puede insertar alarmas o modificar los límites de uso de particiones en servidores o anchos de banda en switches y administrar a grupos de usuarios para la recepción de las alarmas.

En la parte superior se tiene un menú de pestañas que muestran datos de los servidores, datos de los switches, un historial de los dispositivos, así como ligas para ingresar a los otros sistemas de monitorización y un botón para salir del sistema (figura 4.11) .

Figura 4.11 Menú superior

[17] La forma de recuperar una clave probando todas las combinaciones posibles hasta encontrar aquella que permite el acceso.

Igualmente se añade un menú en la parte derecha donde contiene diferentes opciones, dependiendo de los privilegios del usuario; por ejemplo, al usuario administrador se muestran las opciones de administrar servidores, administrar switches, administrar grupos, administrar usuarios y las mismas opciones que en el menú superior (figura 4.12) así como también un menú de usuario para cambiar la configuración de su cuenta. Al usuario monitor no se le muestran las opciones de administración. De esta manera, los usuarios tienen otra forma de acceder a las opciones del sistema.

Figura 4.12 Menú lateral

En la pestaña “Servidores”, se muestra un listado de todos los nombres de los servidores registrados en la base de datos, (figura 4.13) y cada uno es una liga a otra página con los datos de ese servidor, con pestañas extras que muestran la gráfica de las particiones del disco duro (figura 4.14).

Figura 4.13 Listado de servidores

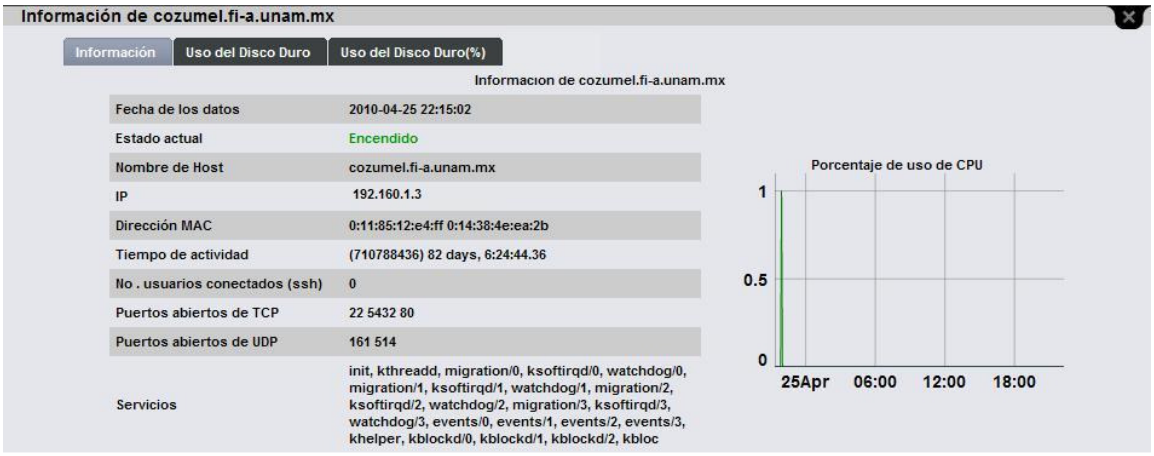


Figura 4.14 Información de los servidores

La pestaña “Switches” lista todos los equipos monitorizados que se encuentran registrados en el sistema (figura 4.15), al hacer clic en ellos se despliegan las gráficas correspondientes a los anchos de banda de cada puerto monitorizado (figura 4.16).

Figura 4.15 Listado de switches

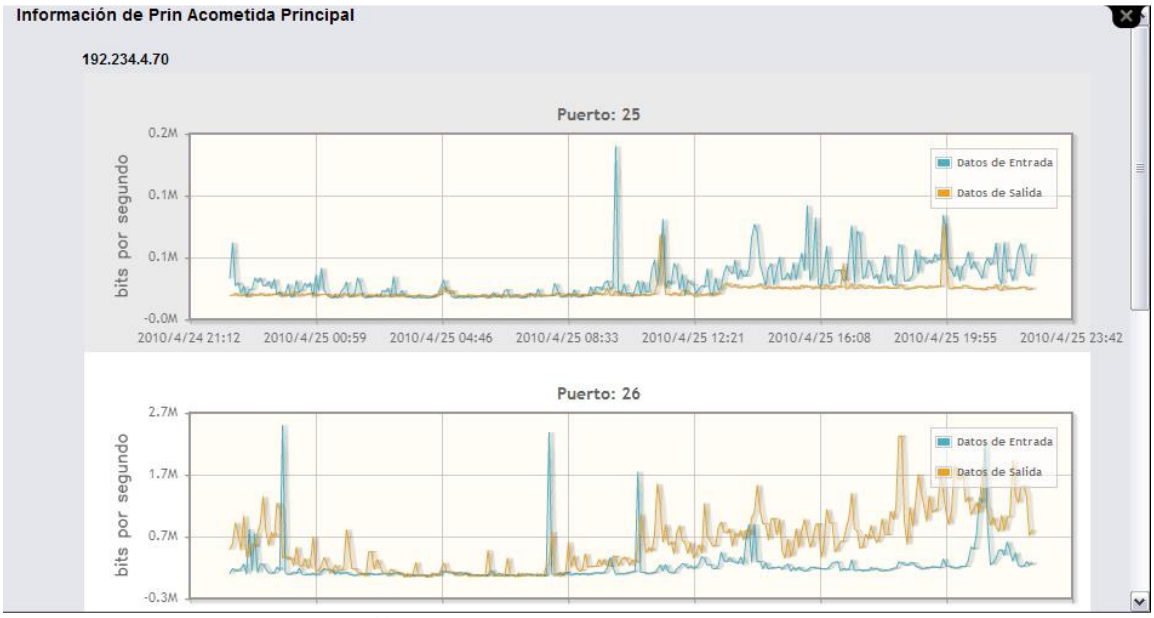


Figura 4.16 Gráficas de los anchos de banda en cada puerto del switch

Dentro de la pestaña “Gráficas” se pueden ver los historiales de cada servidor y de cada switch con rangos de tiempo predeterminados, o bien, el usuario puede escoger sus propios intervalos con días y horas en específico (figura 4.17 y 4.18).

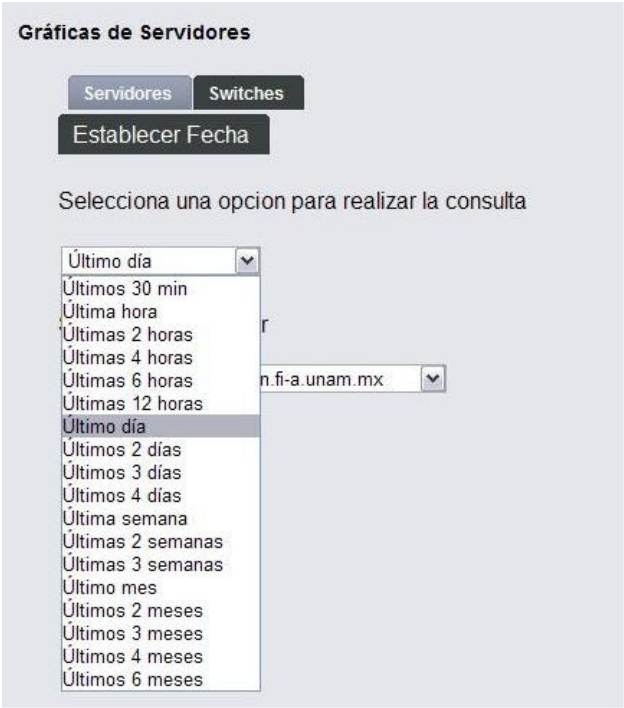


Figura 4.17 Submenú de datos históricos con intervalo predeterminado

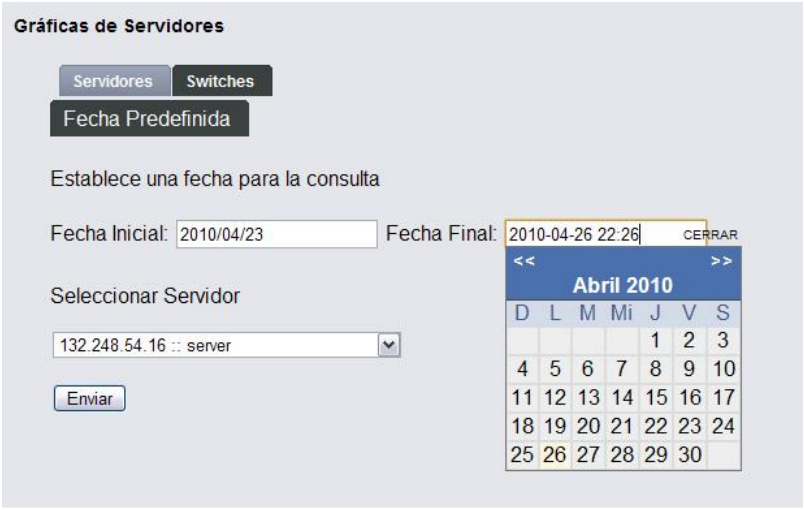
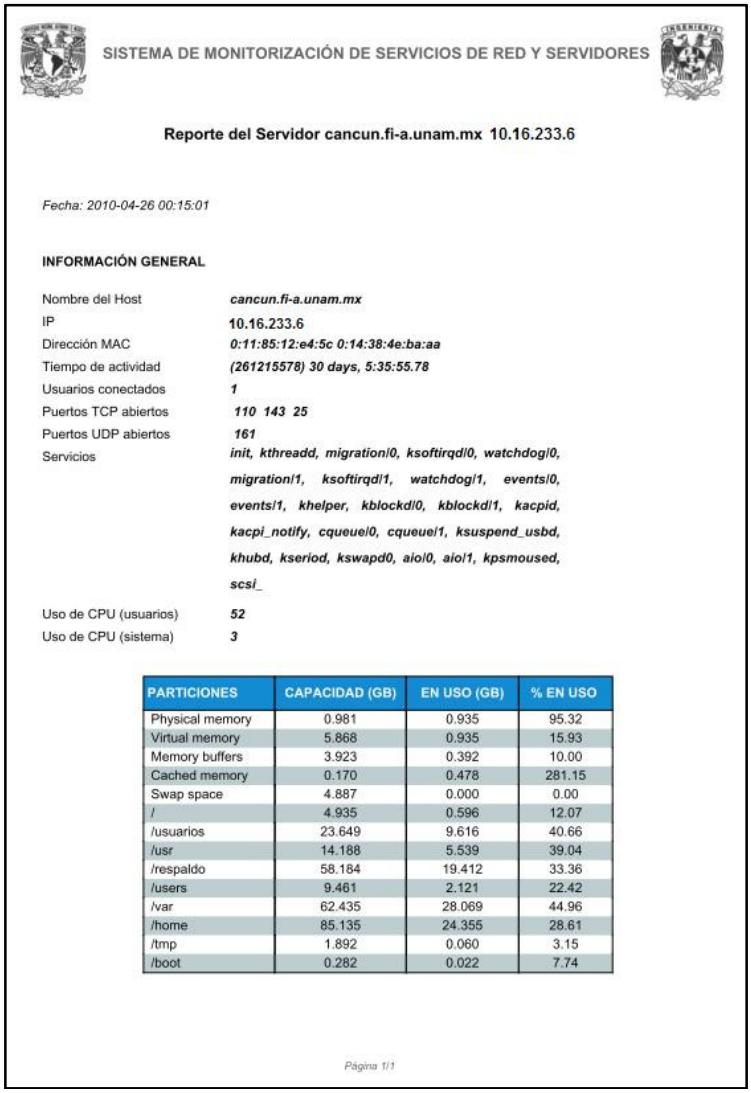


Figura 4.18 Submenú de datos históricos con intervalo definido por el usuario

Las pestañas “Cacti” e “IMC” son ligas a las páginas principales de esos sistemas, desplegando dichas páginas dentro de la consola web de SIMON.

Hasta este punto ya se cuenta con un sistema vía web capaz de mostrar visualmente la información de los equipos y así saber el estado que guardan, comparando resultados con

diferentes herramientas desde un mismo lugar. Sin embargo, un requisito dentro de los objetivos del proyecto es que se pueden hacer reportes de los diferentes equipos. Por lo que se desarrolló un módulo para generar reportes en formato de archivo PDF para cada servidor, así el usuario puede crear un reporte cada vez que lo necesite (figura 4.19). La librería usada fue FPDF, que es una clase escrita en PHP que permite generar documentos PDF directamente desde PHP, es decir, sin usar la biblioteca PDFlib. La F de FPDF significa Free, gratis y libre de usar o modificar.



SISTEMA DE MONITORIZACIÓN DE SERVICIOS DE RED Y SERVIDORES

Reporte del Servidor cancun.fi-a.unam.mx 10.16.233.6

Fecha: 2010-04-26 00:15:01

INFORMACIÓN GENERAL

Nombre del Host	*cancun.fi-a.unam.mx*
IP	10.16.233.6
Dirección MAC	*0:11:85:12:e4:5c 0:14:38:4e:ba:aa*
Tiempo de actividad	*(261215578) 30 days, 5:35:55.78*
Usuarios conectados	*1*
Puertos TCP abiertos	*110 143 25*
Puertos UDP abiertos	*161*
Servicios	*init, kthreadd, migration/0, ksoftirqd/0, watchdog/0, migration/1, ksoftirqd/1, watchdog/1, events/0, events/1, khelper, kblockd/0, kblockd/1, kacpid, kacpi_notify, cqueue/0, cqueue/1, ksuspend_usbd, khubd, kseriod, kswapd0, aio/0, aio/1, kpsmoused, scsi_*
Uso de CPU (usuarios)	*52*
Uso de CPU (sistema)	*3*

PARTICIONES	CAPACIDAD (GB)	EN USO (GB)	% EN USO
Physical memory	0.981	0.935	95.32
Virtual memory	5.868	0.935	15.93
Memory buffers	3.923	0.392	10.00
Cached memory	0.170	0.478	281.15
Swap space	4.887	0.000	0.00
/	4.935	0.596	12.07
/usuarios	23.649	9.616	40.66
/usr	14.188	5.539	39.04
/respaldo	58.184	19.412	33.36
/users	9.461	2.121	22.42
/var	62.435	28.069	44.96
/home	85.135	24.355	28.61
/tmp	1.892	0.060	3.15
/boot	0.282	0.022	7.74

Página 1/1

Figura 4.19 Vista de un reporte en formato PDF

En el prototipo 2 se realizaron todas las funciones necesarias para que la presentación de la información almacenada en la base de datos fuera de ayuda a los usuarios que

consultan este sistema, tratando de tener una interfaz lo más simple pero en balance con la funcionalidad y la estética.

- **Prototipo 3**

Hasta el prototipo 2 se tiene ya un sistema funcional, que permite saber el estado de todos los dispositivos monitorizados, pero todavía no se cuenta con la respuesta oportuna del administrador del equipo en caso de una anomalía. Es por eso que en este prototipo se crea el módulo de alarmas.

Es necesario estar informado oportunamente si ocurre algún incidente en cualquier dispositivo. El sistema SIMON cuenta con un módulo especial de alarmas, las cuales son activadas cuando un equipo pasa por alguno de estos casos:

- Deja de existir comunicación entre SIMON y el sistema monitorizado. Para este caso el sistema realiza la comprobación de los servicios para asegurar que su estado está encendido y permite saber de forma oportuna cuándo está fallando un servicio.

- El ancho de banda de un switch ha excedido el nivel delimitado por el administrador.

- El volumen de información ocupado en alguna partición del disco duro de algún servidor monitorizado ha llegado al límite máximo permitido por el administrador. De esta manera se pueden tomar decisiones tales como dimensionar correctamente el hardware antes que éste sea insuficiente para dar el servicio requerido o identificar la utilización excesiva de recursos que puede indicar procesos mal definidos o que están fallando, o una utilización no apropiada de los recursos por parte de los usuarios.

Una vez que SIMON detecta alguno de estos eventos, emite automáticamente una alarma, vía correo electrónico y mensaje SMS (figura 4.20). Los correos se envían cada 3 horas y los mensajes SMS son enviados cada 8 horas hasta que el problema sea solucionado. Sólo son alertados los administradores asociados al equipo en cuestión. Esto es con la finalidad de no llenar las bandejas de entrada de los administradores de los equipos y de no mal gastar los recursos económicos mandando constantemente varios mensajes. Hay que recordar que el envío de alarmas por medio de mensajes de texto, funciona con un celular común activado con una línea telefónica de una compañía celular. El intervalo de tiempo con que se envían dichas alertas es modificable, pero los intervalos de tiempo actuales fueron definidos por el jefe del Departamento de Redes y Operación de Servidores.

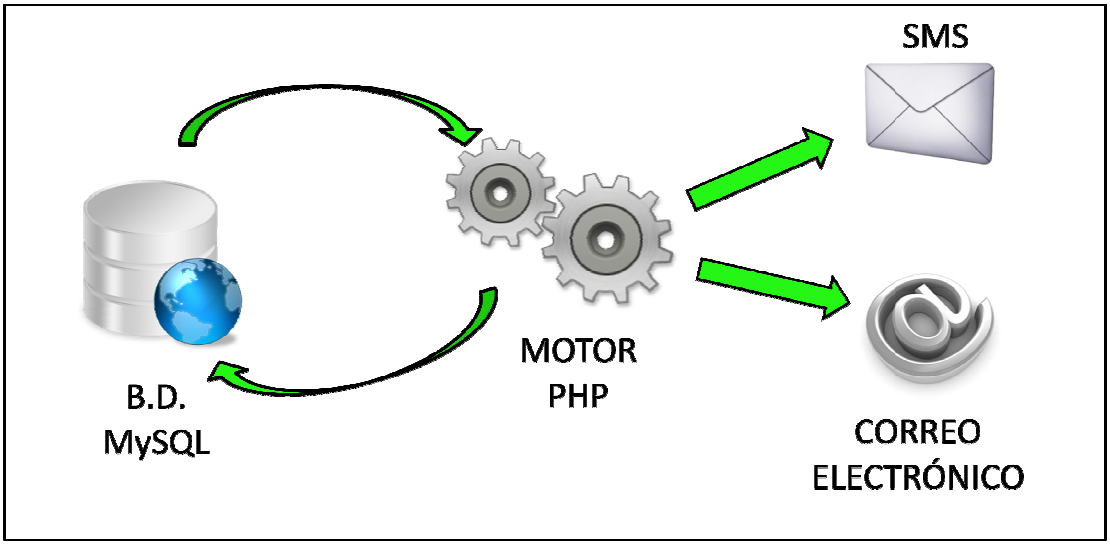


Figura 4.20 Envío de alarmas

Con el fin de llevar un mejor control de los eventos ocurridos, todas las alarmas emitidas son almacenadas, no importando si la alarma fue mandada manualmente por algún usuario del sistema o por el mismo sistema. Esta sección del sistema SIMON es accesible desde el menú principal con el nombre de "Registro de Alarmas", y todos los usuarios tienen acceso a ella (figura 4.21).

Figura 4.21 Registro de Alarmas

En la consola web se puede apreciar el apartado de “Admin alarma partición”, donde la primera sección nos indica las particiones que ya han sido agregadas a las alarmas y por lo mismo son editables. En la figura 4.22 se muestra que la partición de /home está dentro de las alarmas, siendo el 95% el volumen máximo permitido de datos en esa partición y a su vez se puede modificar ese valor dependiendo del criterio de cada administrador.
Se muestra también el grupo de trabajo al cual pertenece el administrador y se le informa de la situación del servidor enviando la alarma correspondiente. Se puede modificar la frecuencia de envío de alarmas SMS, por defecto se encuentra cada 8 horas. Así mismo, la repetición de la alarma por medio de correo electrónico también es modificable. Cada alarma de partición tiene un porcentaje máximo permitido independiente de las demás, ya que cada partición es diferente y el volumen de crecimiento varía entre cada una, es decir, la partición /tmp temporal puede crecer y decrecer de un momento a otro mientras que si ocurriese lo mismo en la partición /home se estaría hablando de un problema serio. Es por eso que una partición puede tener un máximo de 95% mientras que otra partición puede tener un máximo de 80%, esto lo decide el administrador del equipo. La segunda sección muestra las particiones de los servidores que se pueden agregar a las alarmas.

En la liga “Admin alarmas servidor” se muestran los servidores que están siendo monitorizados para saber si se encuentran en línea o se ha perdido la comunicación con estos equipos. Igualmente, se pueden agregar los servidores que se encuentran en la parte inferior para habilitar este tipo de alarmas en ellos.

La liga “Admin alarmas switch” permite modificar o agregar la alarma para el ancho de banda en un puerto de un switch, con las mismas características que las anteriores, modificando el porcentaje máximo permitido en el puerto, alertar al grupo de usuarios encargado del switch y configurar la repetición de las alarmas SMS y de correo electrónico.

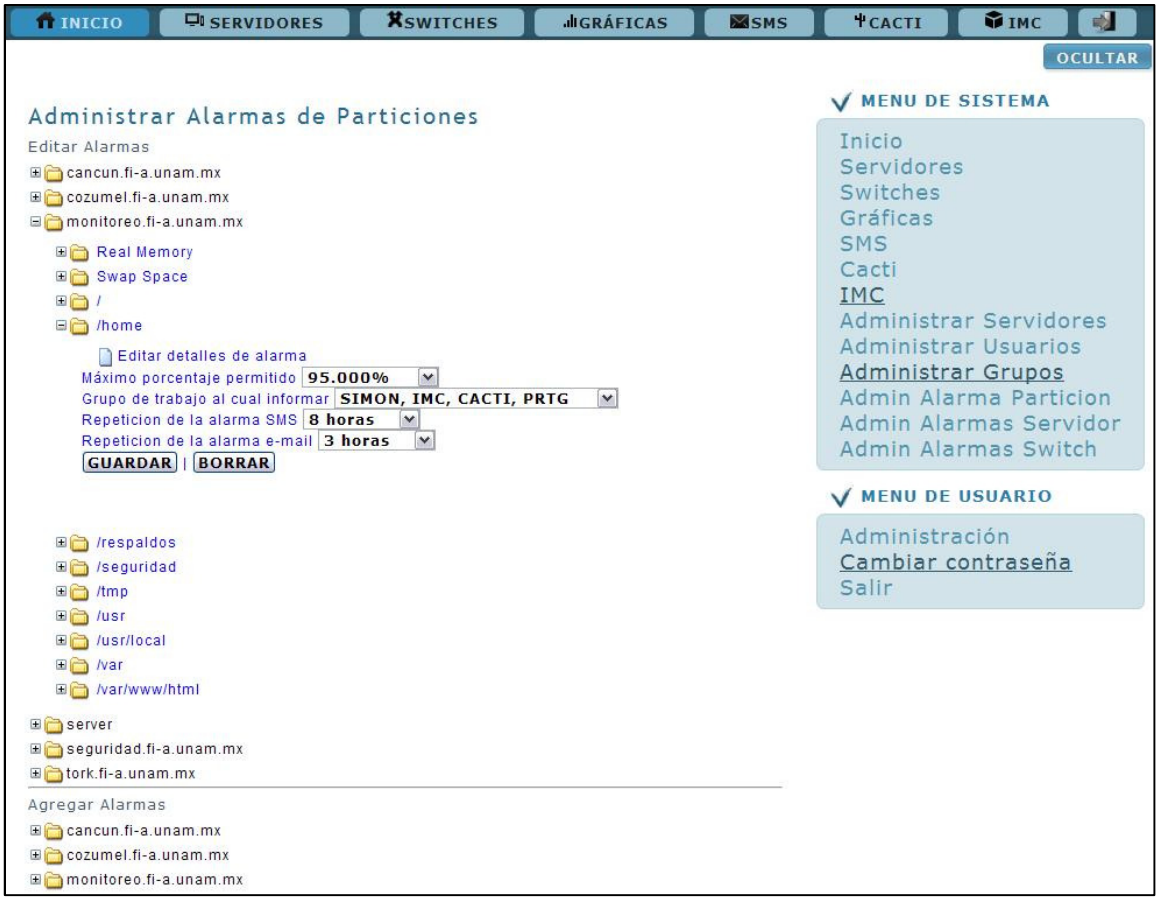


Figura 4.22 Administración de alarmas de particiones

Adicionalmente si algún usuario del sistema necesita enviar un mensaje SMS al resto de sus compañeros para informar acerca de alguna emergencia o simplemente notificar alguna noticia importante o urgente, lo puede hacer desde la pestaña “SMS”, donde se encuentra una lista de usuarios separados conforme al grupo al que pertenecen, donde se puede seleccionar al personal interesado para recibir el mensaje (figura 4.23).

Figura 4.23 Pestaña SMS

- **Logros Obtenidos**

Los resultados que se obtuvieron con este proyecto, fueron de gran utilidad para aquellas personas encargadas de la administración de equipos de red dentro de la Unidad de Servicios de Cómputo Académico, al poder comparar con varias herramientas distintas el estado de su equipo, siendo que estas herramientas tienen funciones en común, pero a la vez, cada una cuenta con opciones diferentes para complementarse entre sí y así tener más información a la hora de tomar decisiones importantes.

Después de haber desarrollado e implementado las herramientas de monitorización de red se pueden enlistar los logros que se obtuvieron con cada herramienta a lo largo de todo este proceso, entre los que destacan:

	LOGROS OBTENIDOS
SIMON	✓ Instalación **desarrollo** y configuración. ✓ Apreciar el **comportamiento de los servidores y de los switches** de una manera clara y resumida. ✓ Informar oportunamente de problemas que ocurren con los dispositivos al personal encargado de la administración de los mismos, mediante un **sistema de alarmas vía correo electrónico y mensajes de texto SMS.** ✓ Ayuda a mantener los servicios de red que proporciona UNICA de manera casi ininterrumpida. ✓ Generar **reportes automatizados** del funcionamiento de los equipos. ✓ Permite observar el comportamiento de los dispositivos durante cierto tiempo gracias al **manejo de datos históricos**. ✓ **Conjunta otras herramientas** para enriquecer las funciones de monitorización. ✓ Sienta las bases para **generar una cultura de monitorización** de equipos en la Unidad.

CACTI	✓ Instalación y configuración. ✓ Se obtuvo información visual sobre el comportamiento de los switches y de los servidores. ✓ Ayudó para aclarar dudas sobre el funcionamiento de dispositivos. ✓ Se observa el desempeño de los equipos de red durante cierto tiempo gracias a los datos históricos que maneja. ✓ Es posible contar con bitácoras de eventos
IMC de 3com	✓ Instalación y configuración. ✓ Ver en forma de mapa la topología de la red ✓ Recopilar estadísticas de los equipos ✓ Detectar equipos de capa 2 y capa 3 de forma automática. ✓ Contar con la opción de administrar equipos
PRTG	✓ Instalación y configuración. ✓ Monitorizar servicios de red ✓ Detectar el desempeño de la red a nivel de protocolos ✓ Contar con reportes históricos ✓ Inspeccionar los paquetes que viajan a través de la red ✓ Se tiene un sniffer capaz de identificar las IP's que generan más tráfico en la red.

Tabla 4.1 Logros Obtenidos

- **Uso de la herramienta en casos reales**

Existe un caso donde la herramienta SIMON fue probada en el mundo real, aún cuando todavía no se encontraba 100% desarrollada. Antes de desarrollar el módulo de alarmas, SIMON ya se encontraba recopilando datos de los dispositivos que se encontraban dentro de la red de la Facultad de Ingeniería. Se detectó que la partición */var* de cierto servidor de bases de datos se encontraba a su máxima capacidad desde varios días antes, abriendo la posibilidad a la pérdida de datos o un posible fallo del sistema de archivos. En este caso

se le informó de manera verbal al administrador y éste a su vez corrigió el problema de inmediato, lográndose ver reflejada su acción en la gráfica.

La figura 4.24 muestra la gráfica de la partición */var* generada por SIMON. Este tipo de documentación fue posible debido a que el sistema de monitoreo cuenta con un histórico de datos.

Después de haber finalizado por completo el sistema con las alarmas, los eventos de este tipo son combatidos de manera oportuna.

Figura 4.24 Uso de la partición /var en un servidor de bases de datos

- **Costos estimados de la aplicación.**

Para obtener el costo de este proyecto se debe tomar en cuenta el perfil del desarrollador, que tiene que cumplir con las características de manejo del lenguaje PHP, bases de datos MySQL y aplicaciones web, conocimientos de diversos sistemas operativos, protocolos de red, manejo y administración del sistema operativo Linux, entre otros. Debido a la complejidad del proyecto se planteó la necesidad de que el equipo de trabajo fuera de dos personas.

Duración del proyecto: 5 meses
Horas invertidas por día: 8
Días laborados por semana: 5
Semanas laboradas por mes: 4
Desarrolladores: 2

Se trabajaron alrededor de 20 días por mes. Por lo que el total de horas trabajadas en este proyecto fue de:

$$\frac{20\ días}{[mes]} * \frac{8\ [hrs]}{[día]} * 5\ [meses] * 2\ [personas] = \ 1600\ [horas]$$

Con un costo de $75 pesos cada hora, el costo de la mano de obra de este sistema sería de $120,000. Hay que agregar el costo del equipo de cómputo que opera como servidor, el equipo celular para el envío de los mensajes de texto sumando entre estos dos $135,000 + (IVA) nos da un costo total aproximado de $156,600 pesos M.N. en mayo de 2010.

La siguiente tabla compara los precios totales al implementar cada herramienta de monitorización.

Aplicación	Software	Licencia Sistema Operativo	Licencia Base de Datos	Licencia Nodos/Sensores	Equipo Computo	Otros	TOTAL + IVA
SIMON	$ 120,000	Linux $ 0.00	MySQL $ 0.00	-----	$ 13,500	Teléfono Celular $ 1,500	$ 156,600
PRTG	$ 0.00	Windows 7 $ 4,600	-----	Ilimitada $ 136,519	$ 13,500	Software Envío SMS $ 9,195 Teléfono Celular $ 1,500	$ 191,764
CACTI	$ 0.00	Linux $ 0.00	MySQL $ 0.00	----	$ 13,500	Teléfono Celular $ 1,500	$ 17,400
IMC	$ 0.00	Windows Server 2008 $ 13,326	SQL Server 2005 $ 77,706	Ilimitada $ 344,900	$ 16,259	Teléfono Celular $ 1,500	$ 526,282

Tabla 4.2 Tabla comparativa de costos estimados.

CONCLUSIONES Y COMENTARIOS FINALES

Diseñamos, desarrollamos y se pusimos en funcionamiento, un sistema de monitorización de fácil uso con una interfaz web amigable capaz de recopilar datos valiosos de servidores y de switches, para ayudar a mantener un óptimo servicio de red y contar con información suficiente para tomar acciones oportunas para corregir posibles fallos.

El sistema (SIMON), es capaz de guardar registros de hasta un año de antigüedad, de esta forma se puede llevar un registro exacto de los datos relevantes de los equipos, ya que el sistema puede generar gráficos históricos con información de switches y servidores, e integra otras herramientas que complementan la información de la monitorización.

Una de las características importantes del sistema, es la rapidez con la que se envían las alertas a los administradores responsables en caso de fallos en los servicios o equipos. Los administradores responsables son avisados oportunamente, vía mensajes de texto a su celular y por correo electrónico, de los eventos anormales en los equipos que se encuentran bajo su responsabilidad.

Incorporamos herramientas adicionales de monitorización de equipos y redes para complementar la información que se obtiene, ayudando a los administradores en la toma de decisiones.

Como elemento significativo en este proyecto, incluimos el envío de mensajes SMS de forma manual, controlados directamente por los usuarios del sistema SIMON, para que, en caso de ser necesario, informen acerca de sucesos importantes al resto de sus compañeros. De esta manera, el personal cuenta con otro medio para mantenerse comunicado.

Comprendimos mejor el funcionamiento del protocolo SNMP. Usamos las comunidades de sólo lectura para garantizar la integridad de la información ya que este protocolo es la base del sistema SIMON.

Al realizar este proyecto de tesis, pusimos a prueba nuestra habilidad para resolver problemas, tanto en la obtención de los datos como en el desarrollo mismo de la aplicación. Al final llegamos a tener un sistema funcional y de gran ayuda para el personal de UNICA.

Se sabe que todo proyecto de cómputo debe de estar en continua evolución con el propósito de mejorar los servicios que ofrece, es por ello que este sistema no ha llegado a su culminación. La implementación, los mecanismos y módulos que lo definen son la base para el desarrollo de nuevas y más complejas herramientas para la monitorización de servicios de red dentro de la Facultad de Ingeniería.

Al concluir este proyecto de tesis cumplimos con los objetivos iniciales y cubrimos otros puntos ya mencionados, que aunque no estaban contemplados desde un principio, se desarrollaron igualmente. Este proyecto lo llevamos a cabo pensando en cubrir las necesidades de la Unidad de Servicios de Cómputo Académico.

BIBLIOGRAFÍA Y MESOGRAFÍA

/proc/meminfo explicado . (3 de Junio de 2009). Recuperado el 29 de Abril de 2010, de Linux AV: http://www.linuxav.net/index.php/2009/06/procmeminfo-explicado/

3com. (2009). Detalles del producto. Recuperado el 10 de Septiembre de 2009, de http://www.3com.com/prod/es_LA_AMER/detail.jsp?tab=features&sku=3CR15800

3com. (9 de Febrero de 2009). Software and Documentation. Recuperado el 2009 de Octubre de 2009, de http://www.3com.com/products/en_US/searchbyfile.jsp?fileid=2&fname=&sort=prodnum&prodcat=65&pgnum=91>

3Com&H3C. (Abril de 2009). iMC Intelligent Management Center v3.2 SP1. Recuperado el 22 de Noviembre de 2009, de http://support.3com.com/documents/netmgr/imc/readme_plat_3.20-R2602P08.html

Alvestrand, H. (19 de Junio de 1995). 1.3.6.1.2.1 - SNMP MIB-2. Recuperado el 1 de Abril de 2009, de http://www.alvestrand.no/objectid/1.3.6.1.2.1.html

ByteSphe. (2009). HOST-RESOURCES-V2-MIB. Recuperado el 30 de Marzo de 2009, de http://www.oidview.com/mibs/0/HOST-RESOURCES-V2-MIB.html

Cacti. (2004). About Cacti. Recuperado el 15 de Agosto de 2008, de http://www.cacti.net/

Cacti. (1 de Junio de 2008). Data Query Templates. Recuperado el 6 de Marzo de 2009, de http://docs.cacti.net/howto:data_query_templates

Case, J., Fedor, M., Schoffstall, M., & Davin, J. (1990). Simple Network Management Protocol (SNMP). RFC 1157.

Chable, H. (2007). Herramientas de monitoreo y detección de intrusos en servidores Linux. México: Tesis de Maestría. Centro de investigación y de estudios avanzados del Instituto Politécnico Nacional.

Drysdale, D. (3 de Julio de 2002). 3com switches. Recuperado el 15 de Febrero de 2009, de HP forum: http://forums11.itrc.hp.com/service/forums/questionanswer.do?admit=109447626+1258665714056+28353475&threadId=3232

Falla, S. (26 de Octubre de 2007). Historia y Evolución del Sistema Operativo Mac OS. Recuperado el 12 de Septiembre de 2008, de

http://www.maestrosdelweb.com/editorial/historia-y-evolucion-del-sistema-operativo-mac-os

Forouzan, B. A. (2002). Transmisión de datos y redes de comunicaciones. España: McGraw-Hill.

Galstad, E. (1 de Mayo de 2007). NRPE Documentation. Recuperado el 12 de Septiembre de 2008, de Nagios: http://nagios.sourceforge.net/docs/nrpe/NRPE.pdf

Gnokii. (2009). Welcome to gnokii.org. Recuperado el 20 de Febrero de 2010, de Gnokii Project: http://www.gnokii.org/

Hewlett-Packard. (30 de Marzo de 2009). SNMP - State of Port, Operational Mode, Port tagging. Recuperado el 25 de Mayo de 2009, de HP forum: http://forums11.itrc.hp.com/service/forums/questionanswer.do?admit=109447626+1253656340655+28353475&threadId=1328065

Kastenholz, F. (1991). SNMP Communications Services. RFC 1270.

Krick, E. (2008). Introducción a la Ingeniería y al Diseño en la Ingeniería. México, D.F.: Limusa.

MAURO, D. (Julio de 2001). Essential SNMP. Recuperado el 8 de Septiembre de 2008, de O'Reilly Media: http://www.intranetwerx.com/bookshelf2/networking_2ndEd/snmp/ch02_06.htm

Microsoft. (2003). El modelo TCP/IP. Recuperado el 4 de septiembre de 2008, de http://technet2.microsoft.com/windowsserver/es/library/d1e53415-9a93-4407-87d2-3967d62182dc3082.mspx?mfr=true

Microsoft. (18 de Noviembre de 2002). Introducción a Windows Server 2003, Enterprise Edition. Recuperado el 8 de Septiembre de 2008, de http://www.microsoft.com/latam/windowsserver2003/evaluation/overview/enterprise.mspx

Microsoft. (28 de Agosto de 2008). Procedimientos recomendados para un funcionamiento óptimo. Recuperado el 4 de Diciembre de 2009, de http://technet.microsoft.com/es-es/library/cc850692.aspx

Nagios. (2009). Recuperado el 17 de Noviembre de 2008, de http://www.nagios.org/

Port Numbers. (s.f.). Recuperado el 5 de Septiembre de 2008, de Internet Assigned Numbers Authority: http://www.iana.org/assignments/port-numbers

Ramirez, C. (2000). Sistema de monitoreo y administración de la red del Instituto de Ingeniería. México: Tesis Licenciatura. Facultad de Ingeniería Universidad Nacional Autónoma de México.

RedHat. (s.f.). /proc/meminfo. Recuperado el 21 de Enero de 2010, de http://www.redhat.com/docs/manuals/enterprise/RHEL-4-Manual/en-US/Reference_Guide/s2-proc-meminfo.html

RedHat. (2009). Requerimientos de hardware. Recuperado el 4 de Diciembre de 2009, de http://docs.fedoraproject.org/release-notes/f12/es-ES/html/index.html#sect-Release_Notes-Hardware_Requirements

RIGBY, C. (s.f.). Apache::WebSNMP. Recuperado el 8 de Marzo de 2009, de CPAN: http://search.cpan.org/~rigbyc/WebSNMP-0.10/lib/Apache/WebSNMP.pm

Rodríguez, D. (2008). Referencias. Recuperado el 25 de Mayo de 2009, de http://cipactli.fi-a.unam.mx/ ~rp_dario/curso/archivos//PHP_prebes_2008_2/Referencias/guia_php_total.htm

Rose, M., & McCloghrie, K. (1991). Concise MIB Definitions. RFC 1212.

Schmuller, J. (2001). Aprendiendo UML en 24 horas. Prentice Hall.

Snmptrap example. (2002). Recuperado el 6 de Marzo de 2009, de iReasoning Networks: http://tl1.ireasoning.com/javadocs/examples/snmp/snmptrap.java.html

Sommerville, I. (2005). Ingeniería de Software. México: Prentice Hall.

Spurgeon, C. (2000). Ethernet: the definitive guide. Estados Unidos: O'Reilly.

Stanford. (11 de marzo de 2001). Passive vs Active Monitoring. Recuperado el 4 de septiembre de 2008, de Stanford Linear Accelerator Center: http://www.slac.stanford.edu/comp/net/wan-mon/passive-vs-active.html

Telecomunicaciones, E. d. (19 de mayo de 2002). EL modelo OSI . Recuperado el 28 de agosto de 2008, de http://elsitiodetelecomunicaciones.iespana.es/modelo_osi.htm

Von Hagen, B., & Jones, B. (2006). Linux Server Los mejores trucos. España: Anaya Multimedia/O'Reilly.

What is Snort? (2008). Recuperado el 15 de septiembre de 2008, de Snort: http://www.snort.org/

Xoops. (2009). Soporte oficial en español. Recuperado el 11 de Enero de 2010, de http://www.esxoops.com/

Xoops. (8 de Septiembre de 2007). Xoops Home. Recuperado el 14 de enero de 2010, de Xoops Project: http://www.xoops.org>