UNIVERSIDAD RICARDO PALMA

FACULTAD DE INGENIERÍA

ESCUELA ACADÉMICO PROFESIONAL

DE INGENIERÍA INFORMÁTICA

HERRAMIENTA INTEGRADA DE MONITOREO DE REDES PARA SOPORTAR ESTUDIOS DE DISPONIBILIDAD

PROYECTO DE TESIS

PARA OPTAR EL TÍTULO PROFESIONAL DE

INGENIERO INFORMÁTICO

PRESENTADO POR

Luis Alberto Del Pozo Guevara

Asesor:

Glen Rodríguez Rafael

LIMA-PERÚ

2007

ÍNDICE

INDICE DE FIGURAS

Dedicatoria

A mi familia que siempre me apoyó e impulsó a continuar en los momentos difíciles de la tesis; a Wendy por siempre estar ahí para apoyarme.

A las personas que dieron su apoyo en el proceso de realización de la investigación. Así como a los señores miembros del jurado por su tiempo y crítica constructiva de la tesis.

Resumen

Esta tesis aborda el problema disponibilidad de redes basada en un monitoreo que ayude a manejar más eficientemente el ancho de banda de la red interna de la organización evitando o minimizando el uso de aplicaciones no permitidas a través de esta. Dicho problema es más critico ya que con el desarrollo de la Internet, el mercado globalizado y crecimiento tecnológico de la empresas, más el gran volumen de información que fluye a través de estas, las organizaciones debes estar más preparadas para asegurar que la información que fluye a través de su red, así como sus aplicaciones, tengan una mayor disponibilidad y performance frente aplicaciones no deseadas que pueden estar circulando por la red.

Ante lo mencionado anteriormente, los encargados de IT deben estar constantemente monitoreando y observado cuales son las posibles fallas a presentarse en caso se detecten sobrecargas en la red por aplicaciones no deseadas.
Actualmente existen un sin numero de aplicaciones de tipo P2P, chat, juegos en línea, etc., que hacen uso de nuestra interna y si estos no son controlados debidamente pueden llegar a sobre saturar nuestra red causando problemas de de conectividad, performance y disponibilidad de servicios internos. A si mismo si tomamos en cuenta el tipo de información que puede fluir a través de estas aplicaciones como virus informáticos y troyanos, la magnitud del problema puede ser mayor.

Es por ello que a partir de este problema, se propone una solución que integres dos módulos de monitoreo, de suceso de eventos (aplicación y sistema); y de tráfico de red en tiempo real, con envío de alertas automático, apenas los umbrales de alerta estándares definidos en la aplicación son sobrepasados.

En resumen se propone una herramienta integrada de monitoreo de red que involucra el análisis de eventos y ocurrencias tanto a nivel de servidor como de tráfico de red. Permitiendo visualizar mediante reportes gráficos dichas incidencias en el tiempo y analizarlas para tomar las medidas correspondientes por parte de la gerencia del negocio y el equipo de TI.

CAPITULO I. Introducción

1.1 Consideraciones generales

El objetivo principal de toda red en las organizaciones es posibilitar el intercambio de información y acceso a recursos por parte del personal que labora en esta.

Es aquí donde los sistemas de monitoreo de redes, juegan un papel importante ayudando a saber que es lo que esta fluyendo a través de nuestra red, permitiéndose dar un servicio continuo o tomar las medidas respectivas en caso de anomalías, tratando de minimizar el impacto en las operaciones normales de la organización.

Actualmente se ha dado un gran interés en hacer que los grandes sistemas organizacionales sean más robustos, pero aparte de la fiabilidad del hardware; la robustez de un sistema esta dado también en el buen diseño de ambos, hardware y software. Es por ello que una rápida recuperación en caso de un evento de falla debe de ayudar, sobre todo si costos intangibles como en caso de perdida de energía en sistemas de alta performance se dan.

Al contar las grandes empresas con sucursales que se interconectan a través de una red en diferentes partes del mundo, la disponibilidad de esa red y

el tráfico que fluye a través de ella es tanto de vital importancia como tener un sistema de tolerancia a fallas.

El monitoreo de la red y los sistemas tolerancia a fallas son dependientes uno del otro ya que toda la información fluye a través de una red y si no hay disponibilidad de red entonces la información que se transmita no alcanzara su destino y no se podrá acceder a ella.

Por ello al realizar el debido monitoreo de tráfico de red se estaría posibilitando una mejora en la disponibilidad de los servicios que ofrece la organización, ya que se podría limitar el tráfico a través de esta a solo los servicios para los cuales esta destinado y evitar posibles usos indebidos por parte del personal administrativo que labora en la empresa u organización.

Por consiguiente la organización podría maximizar el uso de los servicios que ofrece, pronosticando hasta cierta medida si será necesario aumentar hardware o restringir a ciertas aplicaciones el tráfico a través de la red interna o externa.

Al tener esta información relevante sobre que información fluye en la red podemos darle a la gerencia del negocio las razones por las cuales se debería incrementar, y cuanto más, el ancho de banda de la red de la organización, ya que se podría evaluar dicha información mediante un análisis costo beneficio de disponibilidad.

Finalmente también debemos realizar un monitoreo no solo durante los horarios regulares de trabajo del personal que labora en la empresa, sino fuera de horario de trabajo también, ya que hay procesos que solo pueden ser ejecutados en dicho horario, debido a la carga que tiene el sistema y su repercusión en la red de la organización. Así mismo podemos ayudar a aumentar la productividad del personal trabajador, ya que se sabría cuanto tiempo estuvo en Internet y cuanto es su consumo de red.

1.1 Formulación del problema

La presente tesis de investigación “Herramienta integrada de monitoreo de redes para soportar estudios de disponibilidad”, esta formulada basándose en las siguiente hipótesis:

- No se realiza un monitoreo exhaustivo del tráfico que fluye en la red, lo cual puede generar problemas de disponibilidad de los servicios que se ofrecen en la empresa.
- No se cuenta con reportes de tendencias de que tipo de aplicaciones circulan por la red, permitiéndonos establecer cual es la tendencia de uso de dichos programas en un día dado o a través del tiempo, tratando así de minimizar su impacto en la red de la empresa.
- No existe un registro de base de datos de conocimiento del tipo, origen y suceso del evento de error que se dio en el sistema y cual fue la solución que se dio a dicho problema.
- Evitar posibles denegaciones de servicios, caídas o saturación de enlaces de red.
- Demanda que supera la capacidad de transmisión de enlaces.

Con las respuestas a dichas hipótesis se pretende ayudar a que el personal encargado del área de sistemas, TI y gerencia del negocio; use dicha información para calcular los costos de tiempo de respuesta en el caso promedio y el peor de los casos, en ausencia de fallas; o también en el caso promedio y el peor de los casos, en presencia de fallas; determinando así como afecta a la organización y clientes de esta.

Así vez se puede evaluar el costo beneficio de disponibilidad, si es que se plantea un incremento en el ancho de banda de la red. Teniendo información como para sustentar dicho incremento y su beneficio a la gerencia del negocio, su impacto en la performance y acceso a las aplicaciones criticas del negocio.

La productividad del trabajador también se puede incrementar ya que se podría controlar los accesos y usos de la red, por parte del personal.

Cabe resaltar que frecuentemente las hipótesis no son absolutamente verdaderas, ya que dependen de una serie de factores que de no darse podrían dificultar seriamente su resultado.

Así mismo los datos resultantes de dicha hipótesis deben ser cuidadosamente manejados a fin de lograr una mayor productividad y disponibilidad.

1.3 Objetivo General

Se tiene como objetivo general realizar la implementación y construcción de una “Herramienta integrada de monitoreo de redes para soportar estudios de disponibilidad”.

Dicha herramienta integrará dos módulos que se encargaran del análisis del monitoreo de red y visor de eventos de sucesos, en tiempo real, que puedan ocurrir en los servidor que se desee monitorear, de tal forma se pueda obtener la información necesaria para tomar las medidas correctivas del caso, e identificar que tipo de tráfico fluye a través de la red en un momento especifico o en un periodo de tiempo, mitigando el tráfico no deseado y maximizándolo para las aplicaciones y sistemas propios de la organización.

1.4 Objetivos Específicos

Al contar con una “Herramienta integrada de monitoreo de redes para soportar estudios de disponibilidad”. Se pretende lograr lo siguiente:

- Determinar el tráfico que fluye a través de la red en tiempo real.
- Emitir alertas, advertencias e información, de cuando un proceso o sistema esta consumiendo más tráfico de red que el normal, definiendo umbrales.
- Determinar tendencias de consumo de tráfico de red de las aplicaciones a monitorear y que fluyen en la red, para tomar las medidas correctivas del caso.
- Mostrar visualmente en reportes la información de tráfico de red, ya sean en un momento dado de monitoreo tráfico o en un periodo de tiempo.
- Análisis de las bitácoras de eventos de sistema y aplicación de nuestros servidores, de un día en particular o periodo de tiempo.
- Emitir alertas cuando los eventos medidos sobrepasen los umbrales ya definidos.
- Mostrar visualmente en reportes de tendencia de los procesos que nuestras aplicaciones están corriendo en nuestros servidores para identificar posibles problemas de los mismos.

1.5 Justificación de la investigación:

Desde sus orígenes el mundo de la computación informática ha cambiado considerablemente y esta penetrando casi todas las áreas que están predispuestas a tener redes. La clave para un mercado exitoso en el mañana es hacer que la información este disponible en todo momento, en todo lugar, y a lo largo de diferentes canales.

En las universidades, donde se cuentan con una serie de servicios tanto a nivel administrativo como educacional, es de vital importancia controlar que estos sistemas solo fallen un determinado número de veces en un periodo de tiempo ya establecido. Las aplicaciones con diferentes propósitos como por ejemplo: ***El Proceso de Matrícula Universitario***, deben tener una operación continua desde el inicio hasta el fin del

proceso, es decir, cero interrupciones, colocando de esta manera a las computadoras dentro un nivel adecuado de fiabilidad y disponibilidad necesarios para evitar perdidas que se traducirían en costos para la organización.

En los **Bancos**, cuyo servicio de cajeros debe estar disponible las 24 horas del día, y cuyo margen de tolerancia a fallas debe ser 0.9999% aproximado en todo el año.

Cualquiera que sea el método de monitoreo de red a emplear se debe considerar lo siguiente:

- Que es lo que se desea monitorear, ¿que aplicaciones?,
- El desempeño de la performance de la red, en presencia de aplicaciones restringidas.
- El desempeño de la performance de la red, en ausencia de aplicaciones prohibidas.
- El desempeño de performance de la red, en presencia y ausencia de aplicaciones que usa la organización.
- Bajo que protocolos y puertos TCP/IP permitidos deben correr las aplicaciones.

De esta forma trataremos de establecer los medios a tener en cuenta para hacer que un sistema visto desde un punto de vista no ideal lo sea, mediante pautas de monitoreo de tráfico que fluye a través de la red, intentando de esta forma maximizar su uso y evitar que tráfico no deseado fluya a través de nuestra red.

1.6 Alcance

En la presente tesis tiene pensado realizar la construcción e implementación de una herramienta que integre el monitoreo de tráfico de red en tiempo real y de análisis de bitácoras de incidencias de servidores, con el fin de determinar las posibles aplicaciones que más

tráfico están causando en nuestra red y cuales son los posibles procesos o aplicativos que están teniendo un funcionamiento inadecuado, y así tomar las acciones preventivas, correctivas del caso. Evitando una inoperatibilidad de la red y sistemas de la organización.

El desarrollo de dicha herramienta esta conformada por dos módulos integrados en un solo software. El primer módulo del software esta destinado a permitir el análisis de las bitácoras de eventos tanto de aplicación como sistema de un servidor que correo solo bajo cualquier versión de Windows Server. Permitiendo a su vez visualizar dichos eventos en forma de reportes, ya sea para un día dado o para observar su comportamiento a través del tiempo.

También se esta considerado el envío mensajes de alertas en caso dichos sucesos de eventos superen los umbrales definidos por el software. El segundo módulo, se contempla el monitoreo de tráfico de red en tiempo real en base a aplicaciones y protocolos definidos en el software, una ves capturada esa información en tiempo real, esta es almacenada en una base de datos, la cual procesa la información y la muestra en tres tipos de reportes definidos en el sistema.

Se contempla así mismo la opción de visualizar dichos reporte por fecha o rango de fechas para determinar tendencias de tráfico de las aplicaciones.

En resumen se implementará una “Herramienta integrada de monitoreo de redes para soportar estudios de disponibilidad”, que monitoreará y permitirá hacer análisis de la información que esta proporciona para determinar tendencia de comportamiento de las aplicaciones que fluyen en la red de la organización.

CAPITULO II. Monitoreo de redes

En este capitulo será discutido a los problemas de monitoreo de redes y campos de aplicación. Así como los protocolos objetos de estudio de la presente tesis y que son considerados como los más usados por la comunidad de internautas.

2.1 Monitoreo de redes de computadoras

Las empresas de hoy en día utilizan como uno de sus principales recursos de operación a las computadoras, especialmente si están conectadas en red. Debido a ello, la confiabilidad y velocidad de la red de la empresa es crucial para que un negocio sea exitoso.

Los administradores de red tienen que asegurarse que la red este funcionando, sea confiable y veloz, así como que sea utilizada eficientemente. Para lograr todo esto es necesario monitorear la red.

Existen diferentes intereses que se pueden tener al monitorear lo que sucede en una red:

- Monitoreo de paquetes individuales.

- Monitoreo estadístico del uso de la red.
- Monitoreo sensible a eventos específicos que alertan o toman acciones.
- Monitoreo de accesos no permitidos.

2.1.1 Monitoreo de red

Actualmente es muy importante conocer en todo momento lo que está sucediendo en la red, sin embargo es muy común que algunos administradores de red se preocupen por otras cosas y dejen el monitoreo de la red para más adelante porque es considerado un lujo u ocio para quienes desean gozar con el morbo de saber que hacen los otros cuando navegan por Internet (7).

Si pensamos así no podremos estar más equivocados. El monitoreo de la red es vital hasta en la red más pequeña, incluso en algunos casos es conveniente monitorear el uso de la red usando una computadora personal ya así podremos saber cosas como lo siguiente:

- Detectar cuellos de botella en la red.
- Detectar que aplicaciones o servidores hacen más uso del ancho de banda.
- Entregar mejor calidad de servicios a los usuarios anticipándote a los problemas.
- Reducir costos de ancho de banda al optimizar su uso.
- Monitorear el uso de memoria y disco duro de servidores de red.
- Acceder a toda la información estadística en tiempo real desde tu navegador de Internet.

Sin este tipo de software se puede presentar un problema de bajo rendimiento en la red y estaremos prácticamente ciegos al intentar saber que esta sucediendo.

Existe mucho software y hardware para realizar monitoreos en la red desde hardware con costo de más de 1000 dlls hasta software Libre pasando por el freeware.

¿Qué monitorear?

Los recursos presentes en cada sistema son poder de CPU, ancho de banda, memoria y almacenamiento. En el primer vistazo, parecería que la supervisión sólo consistiría de examinar estas cuatro cosas. Lamentablemente, no es tan simple. Por ejemplo, considere una unidad de disco. ¿Qué cosas podría querer saber sobre su utilización?

- ¿Cuánto espacio libre está disponible?
- ¿Cuántas operaciones de E/S realiza en promedio por segundo?
- ¿Cuánto tiempo en promedio toma en completarse cada operación de E/S?
- ¿Cuántas de esas operaciones de E/S son lecturas? ¿Cuántas son escrituras?
- ¿Cuál es la cantidad promedio de datos leídos/escritos con cada E/S?

Hay más formas de estudiar el rendimiento de una unidad de disco; estos puntos solamente tocan la superficie. El concepto principal a tener en mente es que hay muchos tipos diferentes de datos para cada recurso. Las siguientes secciones exploran los tipos de

información de utilización que serían útiles para cada uno de los principales tipos de recursos.

¿Por qué Monitorear?

Monitorear el rendimiento del sistema

El monitorear el rendimiento del sistema se hace normalmente en respuesta a problemas de rendimiento. Bien sea que el sistema está corriendo muy lentamente, o los programas (y algunas veces hasta el sistema completo) fallan en ejecutarse. En cualquiera de estos casos, la supervisión del rendimiento del sistema se realiza normalmente como el primer y el último paso de un proceso de tres pasos:

1. Monitorear para identificar la naturaleza y ámbito de la escasez de recursos que están causando los problemas de rendimiento
2. Se analizan los datos producidos a partir de la supervisión y se toma un curso de acción (normalmente optimización del rendimiento o la adquisición de hardware adicional)
3. Monitorear para asegurarse de que se ha solucionado el problema de rendimiento

Monitorear la capacidad del sistema

La supervisión de la capacidad del sistema se hace como parte de un programa continuo de planificación. La planificación de capacidad utiliza el monitoreo a largo plazo de los recursos del sistema para determinar las tasas de cambio en la utilización de los recursos del sistema. Una vez que se conocen estas tasas de cambio, se hace posible conducir una planificación a largo plazo más exacta con respecto a la adquisición de recursos adicionales.

Monitorear para propósitos de planificación de capacidad es diferente de monitorear el rendimiento en dos formas:

- Se monitorea más o menos de forma continua
- Usualmente el monitoreo no es tan detallado

La razón de estas diferencias se generan de los objetivos del programa de planificación de capacidades. La planificación de capacidades requiere un punto de vista de "cuadro completo"; un punto de vista a corto plazo o el uso incorrecto de recursos es de poco interés. En vez de esto, se recopilan los datos sobre un período de tiempo, haciendo posible categorizar la utilización de recursos en términos de los cambios en la carga de trabajo.

En ambientes definidos de forma más limitada (donde solamente corre una aplicación, por ejemplo) es posible modelar el impacto de la aplicación en los recursos del sistema. Esto se puede hacer con suficiente exactitud para determinar, por ejemplo, el impacto de cinco representantes de servicio al cliente ejecutando la aplicación de servicio al cliente durante la hora pico del día.

Monitoreo de infraestructura

Monitoreo de servidores

Los servidores son el corazón y el espíritu de la infraestructura de la informática hoy en día. Corriendo aplicaciones críticas así como servicios TI tal como correo electrónico, archivo, servicios de impresora y base de datos, disponibilidad y rendimiento de tus servidores.

¿Qué son los registros de eventos de Windows?

Los registros de eventos contienen la información más importante para diagnosticar aplicaciones y fallas del sistema operativo, determinando la vitalidad y estado de tu sistema y verificar que el sistema y las aplicaciones estén operando correctamente.

El sistema de Windows almacena todos los registros en archivos binario .Existen tres registros de eventos básicos: Aplicación (AppEvent.Evt), Sistema (SysEvent.Evt) y Seguridad (SecEvent.Evt). Muchos servidores Windows 2000 (o posteriores) incluyen registros de eventos adicionales: Servidor DNS (DNSEvent.Evt), Servicio de replica de archivos (NtFrs.Evt) y de Servicio de directorio (NTDS.Evt).

El registro del sistema rastrea diversos eventos del sistema como el encendido, el apagado y eventos como controladores y hardware. El registro de aplicaciones es una fuente importante para la información del estado de aplicaciones. Cuando se integra correctamente con el sistema operativo de Windows, las aplicaciones pueden reportar sus errores al registro de eventos grabando una entrada de evento dentro del registro de aplicaciones. Los registros de seguridad rastrean eventos tales como logon, logoff, cambios de los derechos de acceso y apagados y encendidos del sistema.

Proteger tu red de ataques internos

Un estudio reciente por Gartner señala que "las penetraciones más dañinas a un sistema de seguridad de una compañía a menudo provienen de adentro". El estudio manifiesta que el 70% de los incidentes de seguridad que en realidad causan perdidas a la empresa involucra miembros de la empresa. Teniendo Firewalls y antivirus puedes protegerte de hackers del mundo exterior pero no te ayudará contra ataques generados dentro de la empresa. La única manera de proteger tu sistema de tales ataques es monitoreando los registros de servidores Windows 2000/XP/2003 y auto generar alertas en tiempo real.

Monitoreo de la utilización de la CPU, memoria y disco

El espacio de la CPU, memoria y disco son los recursos más críticos para cualquier servidor y si se sobrecarga puede causar que se caiga el servidor o la aplicación. Así pues el monitoreo de la utilización de la CPU, memoria y disco es esencial para garantizar la vitalidad y rendimiento de sus servidores críticos.

Los administradores pueden obtener informes automáticos para identificar los servidores sobrecargados y ocupados en términos de utilización de la CPU y la memoria. Mediante informes de uso del disco a nivel de partición en los servidores los administradores identifican las particiones que más se utilizan y las menos utilizadas de su red.

Monitoreo de aplicaciones

Las aplicaciones son los componentes más críticos de tu infraestructura. Las fallas de aplicación son usualmente los problemas más comunes que ocurren en una infraestructura IT. Monitoreando aplicaciones criticas con medidas preventivas, ayuda a prevenir fallas de aplicación e identificar oportunamente las degradaciones. Entre algunas de las aplicaciones que se monitorean se encuentran: Micrsoft Exchange, MySQL, Lotus Notes, MSSQL y Oracle.

Entre algunos tipos de monitoreo de aplicaciones se encuentran:

Monitoreo de servicios de Windows

Muchas aplicaciones de software que corren en Windows NT, 2000 o más viejos corren como servicios de Windows, corriendo como procesos en segundo plano sin interfaces de usuarios directos y sin usuarios logged-on (e.g. como un servidor). Ellos son automáticamente arrancados y por consecuencia permanecen corriendo. Si alguno de estos importantes servicios falla, muchos usuarios y servicios externos son afectados inmediatamente y de esta manera asegurar que estos servicios estén continuamente corriendo es un requerimiento crucial para la mayor parte de los administradores de sistema.

Monitoreo de servicios

Es posible monitorear la disponibilidad y el tiempo de respuesta de los servicios que se ejecutan en sus servidores. La funcionalidad de monitoreo de servicios le proporciona gráficos e informes detallados acerca de la disponibilidad de los servicios que está monitoreando. Entre algunos de los servicios que se pueden monitorear se encuentran: DNS, IMAP, SMTP, Echo, LDAP, Telnet, FTP, NNTP, Web, POP, HTTPS, etc.

Monitoreo de URLs

El monitoreo de URLs le ayuda a monitorear la disponibilidad de su sitio Web (o sitios Web, si tiene más de uno) o páginas intranet y a verificar si están sirviendo páginas en tiempo real. Algunas tipos de monitoreo se encuentran: Monitoreo de URLs, directorios virtuales, Intranet, coincidencia de contenido, servidores Web y aplicaciones Web.

¿Cómo monitorear?

Las herramientas de monitoreo son software de monitoreo de redes que ofrecen una combinación de monitoreo de WAN, Servidores y Aplicaciones con integración de mesa de ayuda, control de activos y análisis de la funcionalidad del trafico en la WAN. Estos automatizan varias tareas de monitoreo y eliminan la complejidad asociada con el control de la red.

Algunas de las herramientas de monitoreo cuentan con diversos tipos de monitoreo, los cuales son:

Monitoreo de la red: El cual detecta problemas de rendimiento de la red antes de que supongan costosos tiempos de inactividad

Monitoreo de servidores: Mejora la disponibilidad y el rendimiento de su infraestructura de servidores.

Monitoreo de aplicaciones: Identifica problemas de rendimiento de las aplicaciones antes de que tengan impacto en los usuarios finales.

2.2 Diseño y análisis de sistemas de tolerancia a fallas

Un sistema tolerante a fallas es aquel que puede continuar la correcta performance de sus tareas específicas aún en la presencia de fallas de hardware / software. Es la capacidad que tiene un sistema de alcanzar su objetivo de operación tolerante a fallas. Finalmente el término informática tolerante a fallas es usado para describir el proceso de realizar cálculos, como aquellos que son realizados por computadoras, pero en una manera de tolerancia a fallas (4).

2.2.1 Terminología fundamental

Los tres términos fundamentales en el diseño tolerante a fallas son: falla, error y avería. Entre los cuales hay una relación causa-efecto, más específicamente las fallas, son la causa del error, y los errores son la causa de las averías del sistema (6).

¿Que es una falla? Es un defecto físico, debilidad, imperfección, limitación que puede ocurrir en algún componente en particular de hardware o software, en otras palabras, es un error causado quizás por un problema de diseño, construcción, programación, un daño físico, uso, condiciones ambientales adversas o un error humano.

La falla de un componente del sistema no conduce directamente a la falla del sistema, pero puede ser el comienzo de una serie de fallas que quizás si terminen con la falla del sistema.

Un *error* es una manifestación de una falla, una desviación de la precisión o exactitud. Esencialmente una *avería* es la no realización de alguna acción que es debida o esperada.

El concepto de falla, error o avería, puede ser representado mejor usando el modelo de tres universos que es una adaptación del modelo de cuatro universos de Algirdas Avizienis (1982).

El primer universo es el físico en el cual las fallas ocurren, esta contenido por dispositivos semiconductores, impresoras y otras entidades que componen el sistema. El segundo universo es el informacional, donde los errores ocurren, afectando la información convirtiéndola en incorrecta.

El último universo es externo o universo del usuario, donde el usuario del sistema puede ver el efecto de la falla y error, aquí es donde la avería tiene lugar. La avería es cualquier desviación que ocurre de un comportamiento esperado o deseado del sistema.

Esta relación causa-efecto en el modelo de universo nos lleva a la definición de dos parámetros importantes: latencia de falla y latencia de error. *Latencia de falla* es la longitud de tiempo entre la ocurrencia de una falla y la aparición del error que se origino debido a la falla. *Latencia de error* es la longitud de tiempo entre la ocurrencia de un error y la aparición de la avería resultante. De esta forma el tiempo total es la suma de la latencia de falla y latencia de error.

Las fallas pueden ocurrir como resultado de una variedad de cosas que ocurren con componentes electrónicos o durante procesos del sistema. Los *errores de especificaciones*, incluyen algoritmos incorrectos, arquitecturas, o especificaciones de diseño de hardware y software.

Errores de implementación es otra causa; cuando en el momento de transformar las especificaciones de hardware y software a hardware y software físico, se hace un pobre diseño, pobre selección de componentes, pobre construcción y errores de codificación del software.

Otra causa de fallas son los *defectos de componentes*, imperfecciones en la manufactura, defectos aleatorios de dispositivos. Una última causa es la distorsión externa, por ejemplo radiación, interferencia electromagnética, errores de operación.

Para tener una descripción adecuada de fallas, hay que emplear cuatro atributos críticos: la naturaleza, duración, magnitud, y estimación. La *naturaleza* especifica el tipo de falla, fallas de hardware, software, circuitos analógicos o digitales. La *duración*, que es la longitud de tiempo en que la falla esta activa, aquí tenemos *falla permanente* si es que no se toman acciones correctivas y *fallas intermitentes* que aparecen y desaparecen en periodos cortos de tiempo.

Magnitud si es que la falla esta localizada en un módulo hardware o software dado, o si afecta globalmente a uno u otro o ambos. Finalmente la *estimación* que puede ser determinado o indeterminado.

Una *falla determinada* es aquella cuyo estatus permanece incambiable a lo largo del tiempo a menos que sea afectada externamente. *Falla indeterminada* es aquella cuyo estatus en un tiempo T, puede ser diferente de su estatus con un incremento de tiempo mayor o menor que T.

Hay tres técnicas primarias que se usan para mejorar o mantener la performance de un sistema normal en un ambiente donde las fallas es una preocupación: anulación de la falla, enmascaramiento de la falla y tolerancia de fallas. *Anulación de la falla* es una técnica que es usada para intentar prevenir la ocurrencia de una falla, puede incluir cosas como testeos, revisiones de diseño y otros métodos de control de calidad.

Enmascaramiento de la falla es cualquier proceso que prevenga que fallas se introduzcan en el sistema produciendo errores en la

estructura informacional de ese sistema. Y *tolerancia a fallas* es la habilidad del sistema de continuar realizando sus tareas después de la ocurrencia de una falla.

2.2.2 Objetivos de la tolerancia a fallas

La tolerancia a fallas es un atributo de diseño en un sistema que debe cumplir con varias metas funcionales y de performance, así como otros requerimientos como adicionales como son confiabilidad, fiabilidad, disponibilidad, seguridad, performance, mantenimiento y prueba. En suma, el objetivo al construir un sistema tolerante a fallas consiste en garantizar que continúe funcionando de manera correcta como un todo, incluso ante la presencia de fallas. La idea es que el sistema pueda seguir adelante (sobrevivir) a las fallas de componentes, en lugar de que estas sean poco probables (5).

Confiabilidad, es simplemente la calidad de servicio provisto por un sistema particular. Fiabilidad, disponibilidad, seguridad, performance, mantenimiento y prueba son ejemplos de medidas usadas para cuantificar la dependencia de un sistema.

Fiabilidad, es una función del tiempo, *R(t)* definida como la condición de probabilidad de que el sistema funcione correctamente a lo largo de un intervalo de tiempo, dado que el sistema estuviera funcionando correctamente en un tiempo t. La *no confiabilidad* su opuesto y esta expresado en la misma forma que la confiabilidad, es también referida como la probabilidad de falla.

Disponibilidad, es una función del tiempo *A(t)*, definida como la probabilidad de que el sistema este operando correctamente y esta disponible para realizar sus funciones en un instante del tiempo *t*. La disponibilidad difiere de la fiabilidad en que la fiabilidad involucra un intervalo de tiempo mientras que la disponibilidad es tomada de un instante de tiempo.

Seguridad, es la probabilidad *S(t)* de que el sistema realizara ya sea sus funciones correctamente o discontinuará sus funciones en una forma segura que no rompa las operaciones de otros sistemas o comprometa la seguridad de cualquier persona asociada al sistema.

Performance, es una función de tiempo *P(t)*, definida como la probabilidad de que la performance del sistema será o por encima de algún nivel *L*, en un instante de tiempo, *t*. *Degradación natural* es una característica importante que esta estrechamente relacionada con la performance. Es simplemente la habilidad del sistema de automáticamente disminuir su nivel de performance para compensar las fallas de hardware y software.

Mantenimiento, es probabilidad de tiempo *M(t)* de que el sistema fallido será restaurado a un estado operacional en un periodo de tiempo *t*. El proceso de restauración incluye la localización del problema, reparación física del sistema y traída de vuelta del sistema a su condición operativa.

Prueba, habilidad de probar ciertos atributos en sistemas. Medidas de prueba permiten a uno evaluar la facilidad con la cual ciertas pruebas pueden ser realizadas. Las pruebas están claramente relacionadas con el mantenimiento debido a la

importancia de minimizar el tiempo requerido para identificar y localizar el problema específico.

2.2.3 Aplicaciones de la computación de tolerancia a fallas

Se pueden categorizar en cuatro áreas primarias: aplicaciones de larga vida, cómputo crítico, mantenimiento pospuesto, y alta disponibilidad. Cada una presenta diferentes requerimientos de diseño y retos.

a. Aplicaciones de larga vida. Para satélites y vuelos al espacio. El requerimiento típico de estas aplicaciones de larga vida es que tenga un 0.95, o mayor de probabilidad de continuar operando al final de un periodo de diez años. Sin embargo la vida del sistema se puede extender si eventualmente es hecho operacional el funcionamiento de estos sistemas una vez más.

b. Aplicaciones de cómputo crítico. Son aquellas aplicaciones tolerantes a fallas de cómputo donde los cálculos son críticos para la seguridad humana, limpieza ambiental, o equipos de protección. Por ejemplo sistemas de control de vuelos, sistemas militares, y ciertos tipos de controladores industriales.

En esta área la incorrecta performance del sistema del sistema podría producir un resultado devastador. El típico requerimiento de estas aplicaciones es del 0.9999999 (0.9_7) al final de tres horas de un periodo de tiempo, pudiendo cambiar variar de acuerdo al tipo de ambiente. Por ejemplo reacciones químicas que deben ser

precisamente controladas para prevenir explosiones u otros efectos no deseados.

c. Aplicaciones de mantenimiento pospuesto. Cuando mantener operaciones son extremadamente costosa, inconvenientes, o difíciles de realizar. En el espacio, el mantenimiento es imposible de realizar, mientras que en lugares remotos el costo de mantenimiento inesperado puede ser prohibitivo.

La meta principal es el uso de tolerancia a falla para permitir que el mantenimiento pueda ser pospuesto a momentos costo-efectivo más conveniente, evitando mantenimiento no programado.

d. Aplicaciones de alta disponibilidad. Parámetro clave en muchas aplicaciones bancarias y otros sistemas de tiempo compartido. Los usuarios de estos sistemas desean tener una alta disponibilidad de recibir servicio cuando este sea requerido.

2.3 Servidores y Bitácoras de eventos

La disponibilidad de las redes, sistemas de información y servidores son las principales preocupaciones de la mayoría de las organizaciones. Sistemas operativos para servidores como Windows cuentan con bitácoras o archivos de logs de eventos, en los cuales se almacenan las ocurrencias y eventos que pueden suceder en el sistema operativo tanto a nivel de aplicación, seguridad y sistema.

2.3.1 Visor de eventos de Windows

El visor de eventos de Windows es una herramienta que puede ayudar a testear y diagnosticar problemas de configuración. Mediante el uso del visor de eventos, los administradores pueden ver y establecer opciones de logeo para eventos, con la finalidad de reunir información sobre el hardware, software y problemas del sistema.

Por defecto computadoras que corren en sistemas operativos Windows Server 2003, 2000 y NT graban registros de eventos de tres formas:

- *Log de aplicación*, contiene eventos por aplicación o programas. Por ejemplo: un programa de base de datos puede registrar un error de archivo en el log de aplicación. Los desarrolladores de aplicación deciden que eventos registrar.
- *Log de seguridad*, se registran eventos como intentos validos o inválidos de logeo, así como eventos relacionados al uso de recursos como la creación, ingreso, o eliminación de archivos y otros objetos. Por ejemplo: si la auditoria de logeo esta habilitada, los intentos de logeo en el sistema son grabados en el log de seguridad.
- *Log de sistema*, contiene eventos registrados por los componentes del sistema Windows. Por ejemplo: la falla de un driver o de otro componente al cargar durante el inicio es registrado en el log del sistema.

Otros tipos de eventos y eventos de log pueden estar disponibles en una computadora dependiendo de que servicios estén

instalados en ella. Como por ejemplo: si se tienen una computadora con sistema operativo Windows 2003 y es controlador de dominio, entonces también contendrá dos logs adicionales:

- Log de servicio de directorio, este contiene los logs de eventos registrados por el servicio de directorio del directorio activo de Windows. Por ejemplo: problemas de conexión entre el servidor y el catalogo global son registrados en el log de servicio de directorio.
- El log de servicio de replicación de archivo, contiene eventos registrados por el log de servicio de replicación de archivo. Por ejemplo: las fallas de replicación de archivo y eventos que ocurren mientras que los controladores de domino están siendo actualizados con la información sobre cambios del volumen del sistema son almacenados en el log de replicación de archivo.

Cada registro de log esta clasificado por tipo, contiene información e cabecera, y una descripción del evento. El como entender y interpretar los detalles de los logs de eventos se explica a continuación:

Cabeceara de eventos, la cabecera del evento contiene la siguiente información sobre el evento:

- Fecha, en que el evento ocurrió.
- Hora, en que el evento ocurrió.
- Usuario, el nombre del usuario que inició sesión cuando el evento ocurrió.
- Computadora, donde el evento ocurrió.

- ID del evento, un número de evento que identifica el número de evento. El ID del evento puede ser usado por los representantes de soporte para ayudar a entender que ocurrió en el sistema.
- Fuente, fuente del evento. Esto puede ser el nombre del programa, un componente del sistema, o un componente individual del un programa grande.
- Tipo, tipo del evento. Este puede ser uno de los siguiente cinco tipos: error, información, advertencia, auditoria exitosa o auditoria fallida.
- Categoría, una clasificación del evento por el tipo de fuente del evento. Este el primariamente usado en el log de seguridad.

Tipos de eventos, la descripción de cada evento que es logeado depende del tipo de evento. Cada evento en el log puede ser clasificado dentro de los siguientes tipos de eventos:

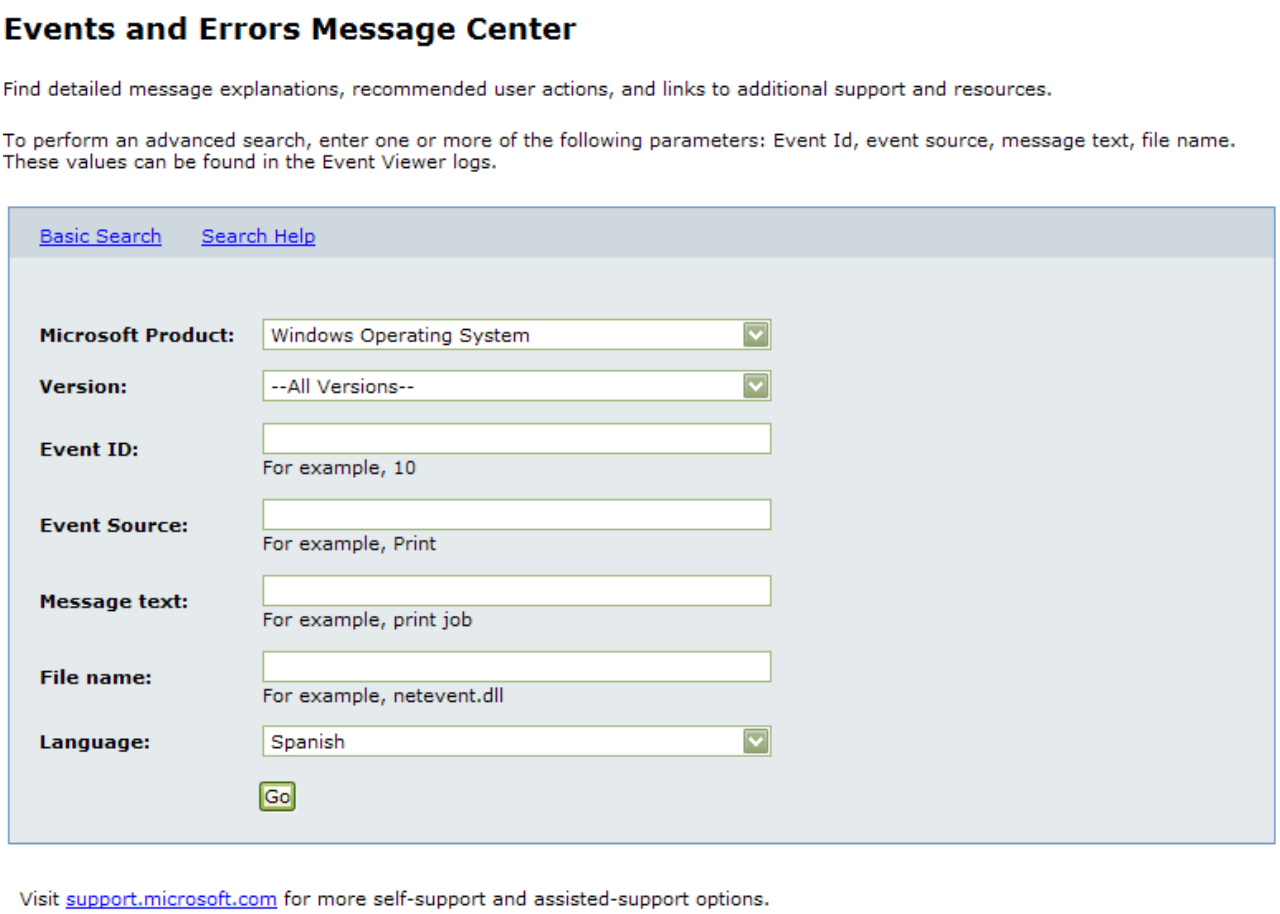
- Información, un evento que describe la realización de una tarea exitosa. Como es una aplicación, driver o servicio. Por ejemplo un evento de información es registrado cuando el driver de la tarjeta de red carga exitosamente.
- Advertencia, un evento que no es necesariamente significativo, sin embargo, puede indicar la posible ocurrencia del un problema futuro. Por ejemplo, un mensaje de advertencia es registrado cuando el espacio del disco empieza a acabarse.
- Error, un evento que describe un problema significativo, como es la falla de una tarea crítica. Los eventos de errores, pueden involucrar la pérdida de data o pérdida de funcionalidad. Por ejemplo, un evento de error es registrado, si un servicio falla al cargar durante el inicio.

- Auditoria exitosa (log de seguridad), un evento que describe realización exitosa de u evento de seguridad. Por ejemplo, un evento de auditoria exitosa es registrado cuando un usuario inicia sesión en su computadora,
- Auditoria fallida (log de seguridad), un evento que describe un evento de seguridad auditado que no se completo exitosamente. Por ejemplo, una auditoria fallida, puede ser registrada cuando un usuario no puede acceder a un driver de red.

Hay que resaltar que en la página Web de Microsoft: http://www.microsoft.com/technet/support/ee/ee_advanced.aspx, hay un centro de mensajes de error y eventos, esquematizado de la de la siguiente manera:

- Producto Microsoft, que se desea evaluar por ejemplo sistema operativo, SQL Server, etc.
- Versión, si es Windows 2003, SQL 2000, etc.
- ID del evento, por ejemplo 10.
- Fuente del evento, por ejemplo impresora.
- Mensaje de texto, por ejemplo trabajo de impresión.
- Nombre del archivo, por ejemplo netevent.dll.
- Idioma, como puede ser español, etc.

En la siguiente figura se ilustra el esquema de mensajes de errores y eventos.



Figuro N° 1: Centro de eventos y mensajes de error de Microsoft.

2.4 TCP / IP

Se considera el protocolo más importante en internet. La función principal de este protocolo es un procedimiento de comunicación general que garantiza la transmisión de datos entre los equipos.

2.5 UDP

Considerado como un protocolo no orientado a la conexión, que en lugar de entregar paquetes entrega datagramas; y en donde la diferencia entre estos dos es que en el paquete hay un control de flujo de errores y en el datagrama no.

2.6 Protocolo HTTP

Protocolo de transferencia de hipertexto, en donde el hipertexto es toda la información y contenido que se manejan en las páginas Web.

Su funcionamiento consiste sencillas operaciones de "solicitud/respuesta". Se basa en el modelo cliente – servidor y que articula los intercambios de información que existen entre los clientes web y los servidores HTTP.

2.7 Protocolo DNS

Aunque no es un protocolo en si, tiene su propia especificación en un paquete de datos, esta basado en la estructura cliente – servidor, el cual basa su mayor actividad en el servidor o bien llamado "resolvedor", la función principal es del servidor es proporcionar la información sobre la relación de direcciones IP – dominio, este protocolo se basa en un sistema jerárquico de dominios y subdominios.

Para finalidad de la presente tesis también se han tomado en consideración más protocolos de monitoreo como son SMTP, POP3, EMULE, Messenger, entre otros.

En resumen se ha hablado sobre el tema de monitoreo de red, dando las pautas de que es monitoreo de red, que monitorear, el como, porque y hacia que. Así mismo se hablo brevemente sobre los sistemas de tolerancia a fallas y como esta ligado al monitoreo de red. Finalmente se menciona los protocolos TCP/IP usados como son HTTP, DNS, SMTP, POP3, entre otros.

En el capitulo siguiente **Tecnología SNMP,** se hace mención al protocolo de monitoreo SNMP, su estructura básica, componentes que lo conforman y su funcionamiento.

CAPITULO III. TECNOLOGÍA SNMP

En este capitulo haremos una presentación sobre la estructura básica del protocolo de monitoreo SNMP. El objetivo de este protocolo es poder realizar un monitoreo del estado de la red y dispositivos que están conectados a esta. En el contexto de la presente tesis será usado como un mecanismo parte del sistema, para realizar el monitoreo de la red. Actualmente muchas aplicaciones de software de monitoreo de red usan este protocolo como base para obtener y presentar la información del estado de la red y el trafico que fluye a través de esta.

En la presente disertación la tecnología SNMP será usada, para construir un prototipo de sistema de monitoreo de tráfico de red en tiempo real y visor de sucesos de eventos con envío de alertas en línea.

3.1 SNMP

El protocolo simple de gestión de redes (SNMP) es un protocolo de la capa de aplicación que facilita el intercambio de administración de la información entre dispositivos de red. Este es parte del protocolo de control de transmisión / protocolo de Internet (TCP/IP). SNMP posibilita a

un administrador de red manejar la performance de red, encontrar y manejar problemas de red, y planear el crecimiento de la red.

Otra definición del protocolo SNMP es la dada por J. Case, M Hedor, M. Schoffstall y J Davin "...la arquitectura del modelo SNMP es una colección de estaciones de administración y elementos de red, donde las estaciones de administración de red ejecutan aplicaciones de administración, las cuales monitorean y controlan elementos de red, estos elementos de red son host, gateways, servidores terminales y parecidos, que tienen agentes responsables de la realización de funciones de administración de red, pedidas por las estaciones de administración de red."

Dos versiones de SNMP existen: SNMP versión 1 (SNMPv1) y SNMP versión 2 (SNMPv2). Ambas versiones tienen un número de características en común, pero SNMPv2 ofrece mejoras, como son operaciones de protocolo adicionales. Estandarizaciones de otras versiones de SNMP, SNMP versión 3 (SNMPv3) todavía esta pendiente. La siguiente figura ilustra una red básica administrada por SNMP.

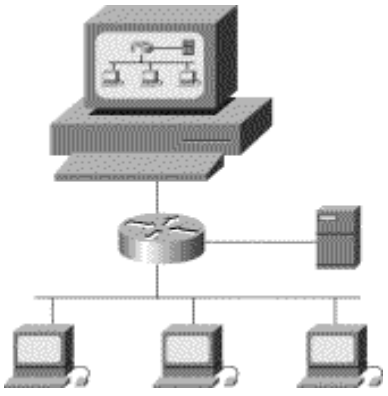


Figura N° 2:- SNMP facilita el intercambio de información de redes entre dispositivos.

3.1.1 Componentes básicos SNMP

Una red administrada por SNMP consiste de tres componentes claves: dispositivos administrados, agentes, y redes administradas por sistemas (NMSs).

Un *dispositivo administrado*, es un nodo de red (cualquier pieza de equipo que esta en la red de datos) que contiene un SNMP agente y que reside en una red administrada. Dispositivos administrados colectan y almacenan información administrada y hacen esta información disponible para NMSs usando SNMP. Los dispositivos administrados son también llamados elementos de red, y pueden ser routers y servidores de acceso, switches y bridges, hubs, computadoras host, o impresoras.

Un *agente*, es un software de administración de red que reside en un dispositivo administrado. Un agente tiene conocimiento local de la información administrada y traduce esa información en una forma compatible con SNMP.

Un *NMS*, es un conglomerado de herramientas de administración de red que ejecuta aplicaciones que monitorean y controlan dispositivos. NMSs provee el tamaño de procesamiento y recursos de memoria requeridos para la administración de la red. Uno o más NMSs deben existir en cualquier red administrada. La siguiente figura muestra la relación de estas tres computadoras.

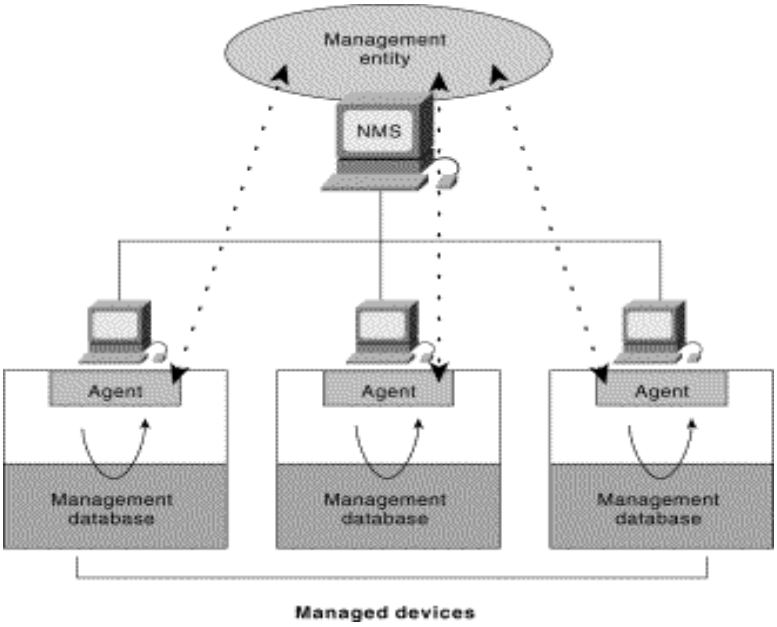


Figura N° 3: Red SNMP administrada que consiste de dispositivos administrados, agentes, y NMSs.

3.1.2 Comandos básicos SNMP

Los dispositivos administrados son monitoreados y controlados usando cuatro comandos básicos de SNMP: read, write, trap y operaciones transversales.

El comando *read*, es usado por un NMS para monitorear dispositivos administrados. El NMS examina diferentes variables que son mantenidas por dispositivos administrados.

El comando *write*, es usado por un NMS para controlar dispositivos administrados. El NMS cambia el valor de las variables almacenadas en los dispositivos administrados.

El comando *trap*, es usado por dispositivos administrados para reportar eventos asíncronos al NMS. Cuando ciertos de eventos ocurren, un dispositivo administrado envía una captura al NMS.

Operaciones transversales son usadas por el NMS para determinar que variables son dispositivos administrados de soporte y para subsecuentemente reunir información en tablas de variables, como son las tablas de ruteo.

3.1.3 Base de gestión de la información SNMP

La base de gestión de la información SNMP (MIB) es una colección de información que esta organizada jerárquicamente. Las MIBs son accedidas usando un protocolo de administración de redes como son SNMP. Ellos constan de objetos administrados y son identificados por objetos identificadores.

Un objeto administrado (algunas veces llamado objeto MIB, objeto, o MIB) es uno de cualquier número de características especificas de un dispositivo administrado. Objetos administrados constan de uno o más instancias de objetos, los cuales son esencialmente variables.

Dos tipos de objetos administrados existen: escalar y tabular. Objetos escalares define una sola instancia de un objeto. Objetos tabulares definen instancias de objetos múltiples relacionados que son agrupados en tablas MIB.

Un ejemplo de un objeto manejado es el atInput, que es un objeto escalar que contiene un solo objeto de la instancia, el valor del

número entero que indica el número total de los paquetes de Appletalk de la entrada en un interfaz de la router.

Un objeto identificador (o objeto ID) identifica únicamente un objeto manejado en la jerarquía MIB. La jerarquía MIB puede ser ilustrada como un árbol con raíz sin nombre, los niveles que son asignados por organizaciones diferentes. La figura muestra el árbol MIB.

El nivel alto de los objetos IDs del árbol MIB pertenecen a diferentes estándares de organizaciones, cuyo nivel bajo de objetos IDs están asociados por organizaciones asociadas.

Los vendedores pueden definir ramas privadas que incluyen objetos administrados para sus propios productos. MIBs que no han sido estandarizadas típicamente son posicionadas en ramas experimentales.

Los objetos administrados atInput pueden ser únicamente identificados ya sea por el nombre del objeto, iso.identified-organization.dod.internet.private.enterprise.cisco.temporary variables.AppleTalk.atInput, o por su objeto descriptor equivalente, 1.3.6.1.4.1.9.3.3.1.

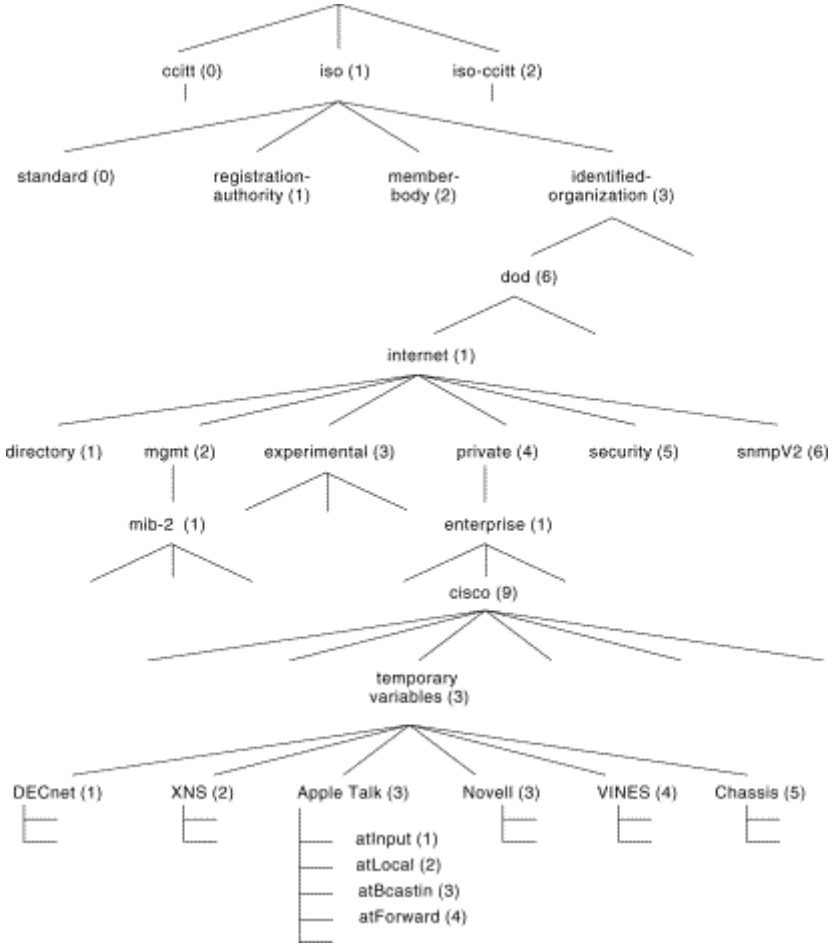


Figura N° 4: El árbol MIB muestra varias jerarquías asignadas por organizaciones diferentes.

3.1.4 SNMP versión 1

SNMP versión 1 (SNMPv1) es la implementación inicial del protocolo SNMP. Este esta descrito en el Request For Comments (RFC) 1157 y funciona con las especificaciones de la Estructura de Administración de la Información (SMI). SNMPv1 opera sobre un protocolo como User Datagram Protocol (UDP), protocolo de Internet (IP), OSI Connectionless Network Service (CLNS), AppleTalk Datagram-Delivery Protocol (DDP), y Novell Internet Packet Exchange (IPX). SNMPv1 es ampliamente usada y es el protocolo de administración de redes de facto en la comunidad de Internet.

a. SNMP y Estructura de administración de la información

La estructura de administración de la información (SMI) define las reglas para la administración de la información, usando Notación de Sintaxis Abstracta 1 (ASN.1). El SNMPv1 SMI esta definida en el RFC 1155. El SMI marca tres especificaciones claves: ASN.1 tipos de dato, SMI – tipos específicos de datos, y tablas SNMP MIB.

a.1 SNMPv1 y tipo de datos ASN.1

El SNMPv1 SMI especifica que todos los objetos administrados tienen un cierto subconjunto de ASN.1 tipos de datos asociados a el. Tres tipos de datos ASN.1 son requeridos: nombre, sintaxis y codificación. Los nombres de los servidores como los objetos identificadores (objeto ID). La sintaxis define los tipos de datos del objeto (por ejemplo entero o cadena). El SMI usa un subconjunto de definiciones de sintaxis ANS.1. La codificación de la data describe como la información asociada con el objeto administrado es ajustado al formato como una serie de datos para la trasmisión sobre la red.

a.2 SNMP Tablas MIB

SNMPv1 SMI define una alta estructura de tablas que son usadas para agrupar las instancias de objetos tabulares (eso es, un objeto que contiene múltiples variables). Las tablas están compuestas de

ceros y muchas filas, las cuales son indexadas en forma que permiten al SNMP recuperar o alterar una completa fila con un solo get, get-next o set comandos.

b. Protocolo de operaciones SNMPv1

SNMP es un simple protocolo de petición/respuesta. El sistema de administración de red envía una petición, y los dispositivos administrados regresan una respuesta. Este comportamiento es implementado usando uno de cuatro operaciones de protocolos: Get, Getnext, Set, y Trap. La operación Get es usada por el NMS para recuperar el valor de uno o más instancias de objetos del agente. Si el agente que responde a la operación de Get no puede proveer valores para todas las instancias de objetos en la lista, entonces no provee ningún valor. La operación Getnext es usada por el NMS para proveer el valor de la siguiente instancia del objeto en la tabla o lista que esta en el agente. La operación Set es usada por el NMS para establecer valores de instancias de objeto en el agente. La operación Trap es usada por el agente para asincrónicamente informar al NMS de un evento significativo.

3.1.5 SNMPv2

El SNMP versión 2 (SNMPv2) es una evolución de la versión inicial SNMPv1. Originalmente SNMP fue publicado como un conjunto de estándares de Internet en 1993. Como con SNMPv1, SNMPv2 funciona con las especificaciones del SMI. En teoría SNMPv2 ofrece un número de mejoras con respecto al SNMPv1, incluyendo operaciones de protocolo adicionales.

a. SNMPv2 y la estructura de administración de la información

El SMI define las reglas para describir la administración de la información, usando ASN.1.

El SNMPv2 SMI hace ciertas adiciones y mejoras al tipo de dato específico del SNMPv1 SMI, como son la inclusión de una cadena de bit, direcciones de red, y contadores. La cadena de bits esta definida solo en SNMPv2 y abarca ceros y nombres de bits que especifican un valor. Las direcciones de red representan una dirección de una familia de protocolos en particular. SNMPv1 soporta so9lo 32 bits de dirección IP, pero SNMPv2 puede soportar otros tipos de direcciones también. Los contadores son enteros no negativos que se incrementan hasta que alcanzan un valor máximo y luego retornan a cero. En SNMPv1, un contador de 32 bit de tamaño es especificado. En SNMPv2, un contador de 32 bit y 64 bit son definidos.

b. Módulos de información SMI

El SNMPv2 SMI también especifica módulos de información, el cual especifica un grupo de definiciones relacionadas. Tres tipos de módulos de información existen: módulos MIB, declaraciones de conformidad y declaraciones de capacidad. Los módulos de MIB contienen definiciones de objetos administrados interrelacionados. Las declaraciones de conformidad proveen una forma sistemática para describir un grupo administrado de objetos que deben ser implementados de conformidad al estándar. Las declaraciones de capacidad

son usadas para indicar el nivel preciso de soporte que un agente demanda con respecto a un grupo de MIB.

c. Protocolo de Operaciones SNMPv2

El Get, Getnext, y Set usadas en SNMPv1 son las mismas operaciones usadas en SNMPv2. Sin embargo, SNMPv2 agrega y mejora algunas operaciones de protocolo. Por ejemplo en SNMPv2 la operación Trap sirve se usa de la misma forma que en SNMPv1, pero usa un formato de mensaje diferente y esta diseñado para reemplazar al SNMPv1 Trap.

También se definen 2 nuevas operaciones de protocolo: GetBulk y inform. El primero es usado por el NMS para proveer eficientemente grandes bloques de datos. El inform permite a un NMS enviar un información Trap a otro NMS y luego recibir una respuesta. En SNMPv2 si el GetBulk no puede proveer un valor para todas las variables en la lista, da una resultados parciales.

3.2 Administración SNMP

SNMP es un protocolo de administración distribuida. Un sistema puede operar exclusivamente como ya sea un NMS o un agente, o ambas funciones. Cuando se esta operando como ambos NMS y agente, otro NMS puede ser requerido para que consulte el sistema de administración de dispositivos y provea un resumen de la información aprendida.

3.3 Interoperabilidad del SNMP

SNMPv2 es incompatible con SNMPv1 en dos áreas fundamentales: el formato del mensaje y el protocolo de operaciones. el mensaje SNMPv2 usa diferentes cabeceras y formatos de protocolo de unidad de datos (PDU) que los mensajes SNMPv1. SNMPv2 también usa dos protocolos de operaciones que no están especificados en SNMPv1. Más aun, en el RFC 1908 se definen dos posibles coexistencias entre el SNMPv1/v2: agentes Proxy sistemas bilingües de administración de redes.

3.3.1 Agentes Proxy

Un agente SNMPv2 puede actuar como un agente Proxy en representación del administrador de dispositivos SNMPv1, como se muestra:

- Un SNMPv2 NMS emite un comando previsto para un agente SNMPv1.
- El NMS envía un mensaje SNMP al agente Proxy SNMPv2.
- El agente Proxy envía un mensaje Get, Getnext, Set al agente SNMPv1.
- El mensaje GetBulk es convertido por el agente Proxy a un mensaje Getnext y luego es enviado al agente SNMPv1

El agente Proxy mapea el mensaje Trap SNMPv1 al mensaje Trap SNMPv2 y luego los envía al NMS.

3.3.2 Sistema de administración de redes bilingüe

Sistema de administración de redes bilingüe SNMPv2 soporta ambos SNMPv1 y SNMPv2. Para soportar este doble ambiente de administración, una aplicación de administración bilingüe en el

NMS debe contactar al agente. El NMS entonces examina la información almacenada en la base de datos local para determinar si el agente soporta ambos SNMPv1 o SNMPv2. Basado en la información en la base de datos, el NMS se comunica con el agente usando la versión apropiada del SNMP.

En el presente capitulo se trataron el tema del protocolo de monitoreo de red SNMP, componentes que lo conforman, las versiones de SNMP que existen y cuales son las diferencias y ventajas entre uno y otro.

CAPITULO IV. HERRAMIENTAS DE CÓDIGO ABIERTO

En la actualidad la mayoría de las grandes empresas cuentan con un sin número de servicios que ofrecen a sus trabajadores y clientes. Muchos de estos servicios están conectados a través de una red para uso interno y externo.

El mundo globalizado de hoy hace que la mayoría de las empresas tomen medidas para proteger dichos servicios de potenciales atacantes. Es por ello que es de vital importancia maximizar el uso del ancho de banda de la red de la empresa, de tal forma que a través de ellas solo transiten los servicios para los cuales esta disponible.

En muestro país la mayoría de la empresas no realizan un debido monitoreo de sus redes, ya sea por falta de recursos o por desconocimiento de cómo hacerlo.

Entre los ejemplos de nuestra realidad tenemos la oficina de informática y computo de la Universidad Ricardo Palma (OFICIC), al tratarse de una institución universitaria que cuenta con muchos servicios disponibles

para alumnos, profesores y personal administrativos. No realiza un monitoreo del tráfico de red exhaustivo y no cuentan con un historial de eventos de sucesos de problemas de red y servidores que funcionan en esta. Al no contar con este tipo de información no se puede realizar un estudio minucioso de cuales serian las posibles causas y fallas de los servicios; y si se hay problemas se busca únicamente una solución inmediata para corregir el problema pero no se busca determinar cual fue el verdadero origen de la falla.

Estos estudios de monitoreo de red no solo sirven para determinar que tipo de tráfico pasa por la red sino también para determinar un posible crecimiento en el ancho de banda de la red, ya sea, por crecimiento de la empresa, crecimiento de usuarios de esta o un mayor cantidad de servicios que ofrece.

Así mismo posibilitamos a que el negocio de la organización pueda tener una mejor disponibilidad de servicios, rápidas respuestas y procesos internos que se estén ejecutando durante la jornada laboral, contando con la información necesaria en el momento que se requiere, sin retraso alguno.

Otro ejemplo seria el departamento de cómputo del Ministerio del Interior, el cual al tratarse de una institución de gran envergadura y cobertura a nivel nacional, no cuenta con estadísticas de fallas de servicios de red ni con información de ningún tipo que pueda dar una idea del estado actual de la red, situación que podría ser critica si es que su información se ve comprometida de alguna forma y no esta disponible (en algunos casos 24/7).

4.1 Determinación de herramientas faltantes

Aún cuando en el mercado existen herramientas que hacen un trabajo similar de monitoreo de tráfico de red, dichas herramientas no filtran de manera precisa la información que un administrador de red podría estar buscando en particular.

Al encontrarse dicha información de manera amplia el administrador muchas veces se ve en la necesidad de realizar un trabajo minucioso para descartar aquella información que no le es de interés. Ante dicha situación el administrador de red opta por no realizar dicha labor ya que le resulta tediosa y abarca demasiado tiempo que podría usar para otras actividades.

Otro inconveniente de estas herramientas es la complejidad de uso ya que al contener múltiples opciones muchas de estas no son usadas debido a su falta de conocimiento, ocasionando que estos productos sean dejados de lado su uso.

4.2 Estudio de alternativas

En el mercado hay herramientas disponibles que pueden ayudar a los administradores de red a realizar un monitoreo más adecuado del tráfico que fluye en su red, dándoles la posibilidad de tomar las acciones preventivas y correctivas del caso.

4.2.1 Definición de alternativas

Podemos decir que las empresas o organizaciones cuentan con las siguientes opciones entres las cuales pueden elegir bajo

factores que ellos mismo deben determinar. Entre estas opciones están: software disponible en el mercado; que muchas veces es de un costo elevado y que ofrece amplia gama de opciones de monitoreo de red. Software hecho a medida, donde la empresa solicita a un outsourcing la elaboración de un software hecho para realizar labores específicas, y software gratuito.

4.2.2 Código abierto

Entre algunas herramientas de código abierto podemos mencionar:

- **MRTG – Multi router traffic grapher**, este es un programa de código abierto que se puede implementar sobre sistemas operativos UNIX (más común) y Windows para realizar muestreos gráfico del tráfico con un tiempo de muestreo de mínimo 5 minutos en el modo normal usando diferentes fuentes de datos, pero generalmente el utiliza el SNMP.

  Dado que cada ciclo de muestreo, se genera un gráfico de cada monitoreo, esto trae como desventajas una carga al sistema que en la práctica hace limitada su implementación a gran escala, y además dado que su gráfico es un muestreo de 5 minutos, es difícil detectar alguna falla durante los lapsos entre cada muestreo.

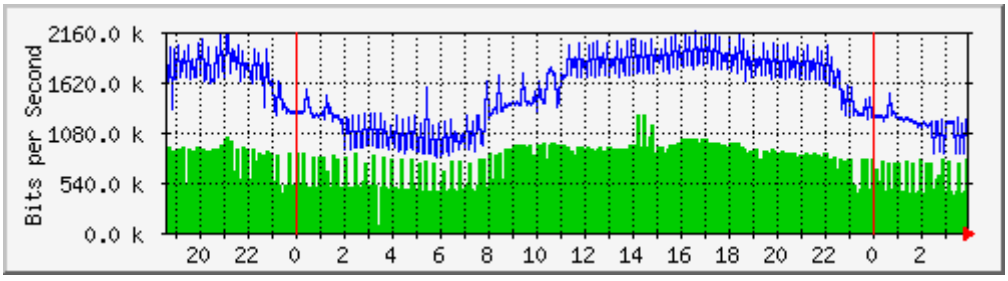


Figura Nº 5: Gráfica de monitoreo de tráfico de red con el MTRG.

- **Herramientas RRD – Round robin database tools**, este como tal no es un programa de monitoreo de tráfico, sino más bien un programa de código abierto que en conjunto con el programa MRTG, permite tener mayor capacidad de monitoreo al descargar la tarea de generar los gráficos cada 5 minutos y además almacenar los muestreos dentro de una base de datos (exportable a XML) lo cual lo hace más eficiente, y permite que el muestreo se pueda llevar a cabo con un mínimo de un minuto.

  Adicionalmente trae módulos para tomar esa información y generar los gráficos. Generalmente se utiliza un programa interfaz en conjunto con el RRD para generar los gráficos de los cuales hay varios

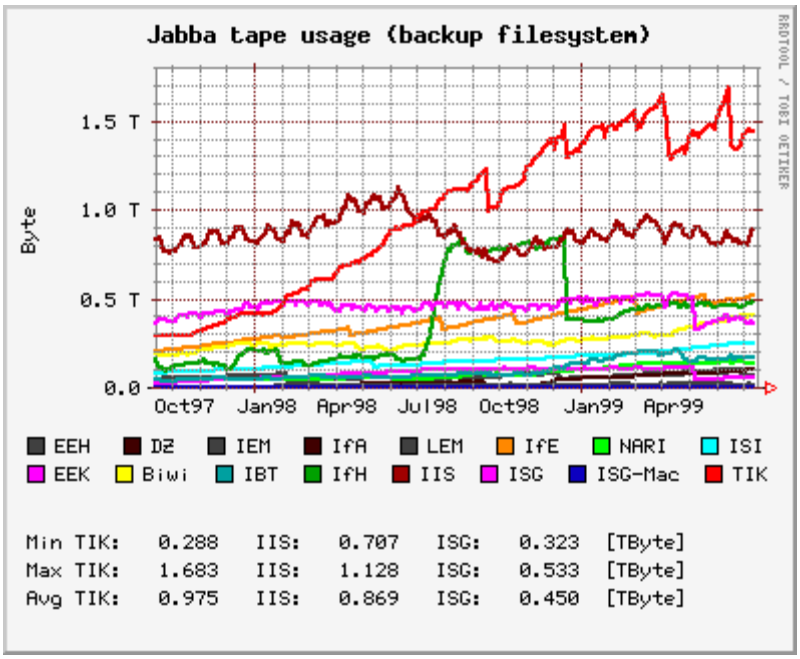


Figura Nº 6: Muestreo de tráfico con el RRD.

- **Router2.cgi**, Este es un programa de Código Abierto que funciona como un CGI usando PERL y que utiliza los datos generados del RRD Tools y la configuración de MRTG para

generar los gráficos de monitoreo cuando se necesite visualizarlo y de manera organizada y personalizada.

- **Ethereal ,** es una herramienta multi-plataforma para resolver problemas de análisis, desarrollo de protocolos de software y educación. Es de licencia de código abierto que permite a los expertos en la comunidad de redes añadir mejoras. Correo bajo todas las plataformas, Linux, Unix y Windows.

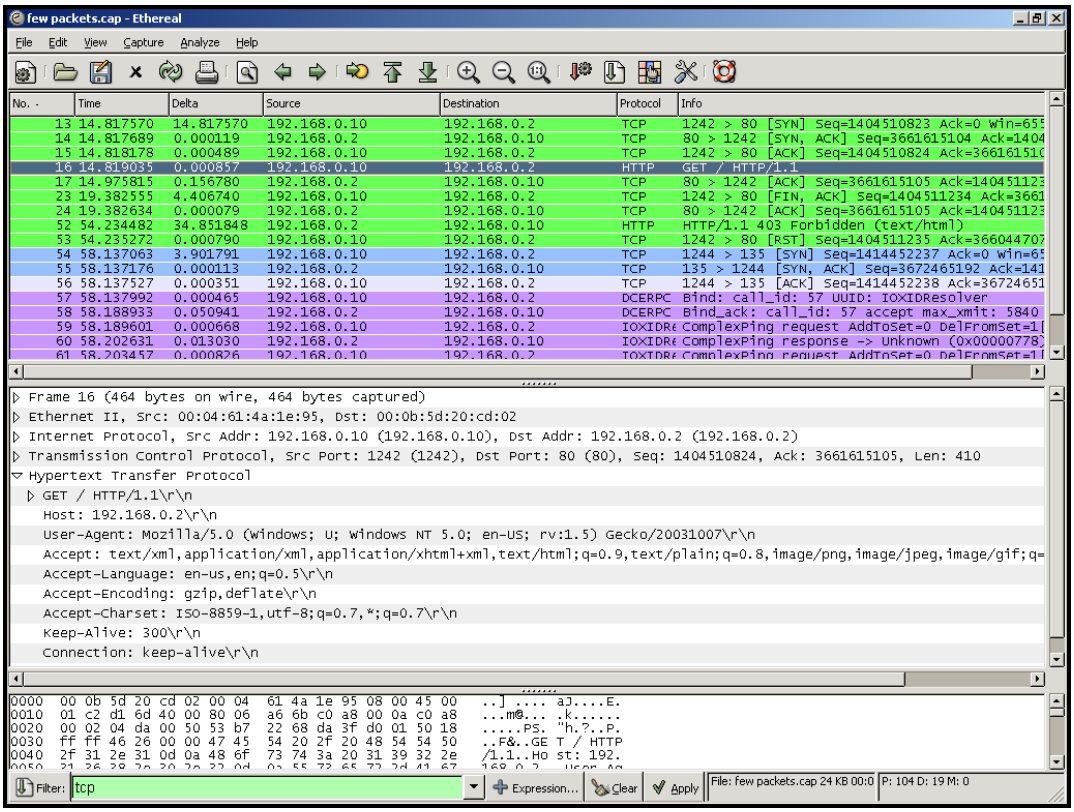


Figura Nº 7: Muestra grafica de los datos capturados con Ethereal.

- **STG - SNMP Traffic Grapher**, Es programa gratuito que corre desde Windows 98, y fue desarrollado como un complemento al MRTG. Realiza gráficos de dos colores (generalmente interpretado como entrada/salida) a través de contadores en agentes SNMP en tiempo real con un mínimo de tiempo de muestreo desde 100 milisegundos, aunque en la práctica, es

útil con un muestreo entre 500 ms a 5000 milisegundos (5 segundo), además de poder guardar las muestras en un archivo de CSV y procesarlo con otro programa, como también permite guardar la configuración del monitor como un archivo y poder utilizarlo después.

Este es un programa básico que debe tener siempre ya que ayuda a detectar fallas rápidamente sin esperar por los gráficos de MRTG. Se puede descargar del sitio http://leonidvm.chat.ru/.

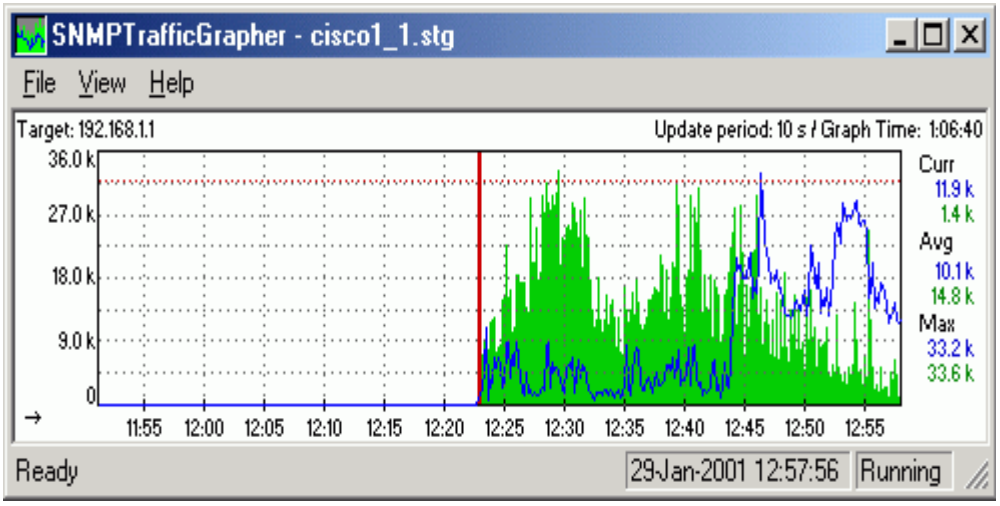


Figura Nº 8: Muestreo grafico de red con STG.

4.2.3. Solución comercial

- **PRTG Traffic Grapher,** Es programa comercial (con una versión gratuita limitada) que sirve como complemento al STG pero con mayores facilidades y opciones tales como un servidor Web para mostrar los gráficos a través de un navegador, gráficos históricos, etc. (ver ejemplo). Se puede descargar del sitio http://www.paessler.com/prtg/

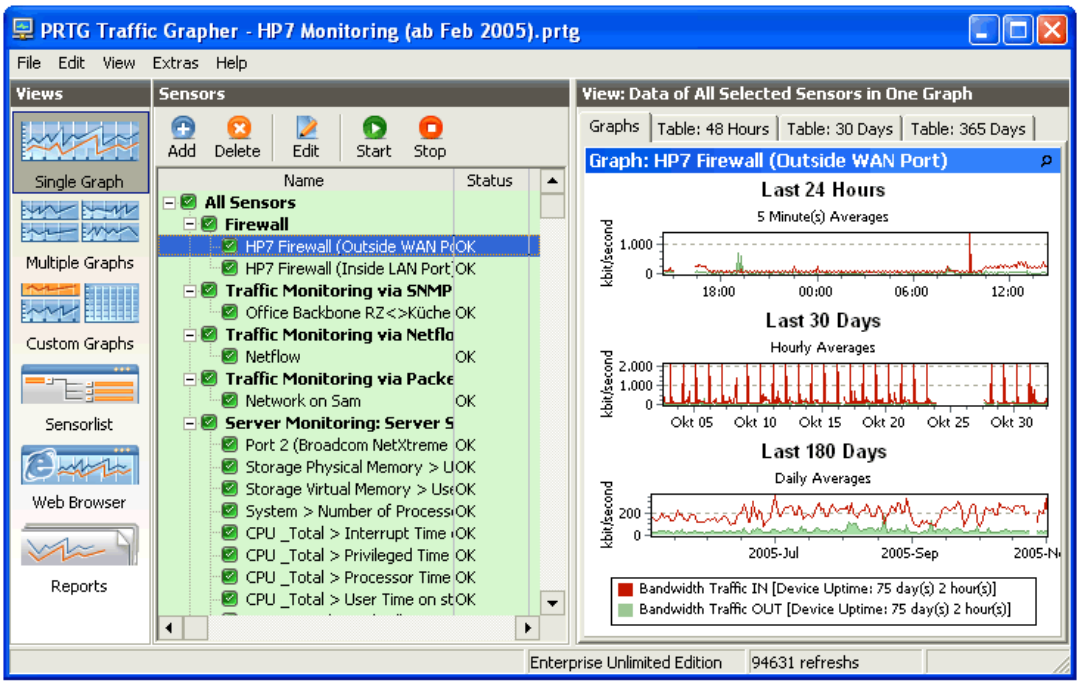


Figura Nº 9: Interfaz grafica del PRTG.

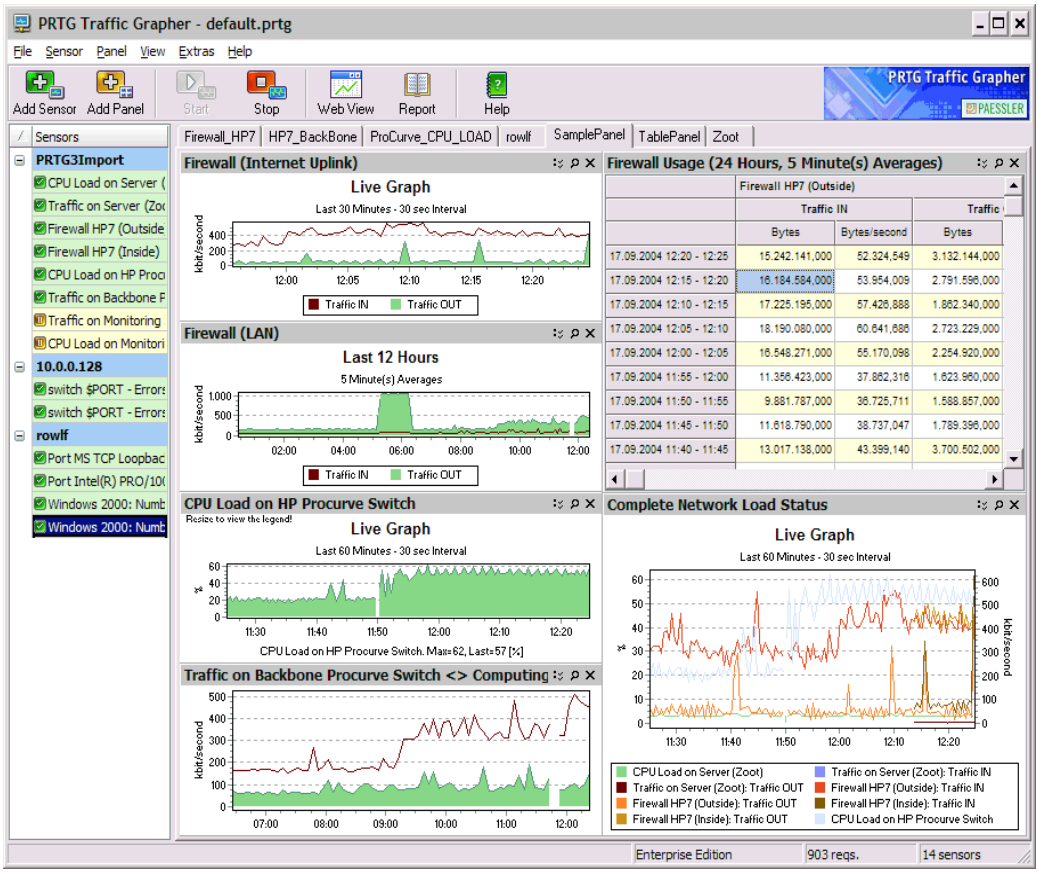


Figura Nº 10: Interfaz grafica del PRTG.

- **Sniffer network analyzer,** línea de productos que pertenece a network associates. Herramienta para el monitoreo, resolución de problemas de red, visualización de aplicaciones y picos de tráfico de red. Cuenta con una gama de productos para el monitoreo de tráfico de red, entre los cuales tenemos: sniffer para ambientes distribuidos, para ambientes movibles, gíreles y para protección de red.

En el presente capitulo hemos explorado algunas de las herramientas comerciales más conocidas y de código abierto que existen, también hemos visto sus ventajas y desventajas de funcionalidad.

En el siguiente capitulo V implementación de una herramienta integrada de red, vamos a ver la razón de ser esta tesis, su sistema prototipo implementado y la problemática que se pretende solucionar con su implementación.

CAPITULO V.

IMPLEMENTACIÓN DE UNA HERRAMIENTA INTEGRADA DE RED

En el presente capitulo se presenta una aplicación que aborda una herramienta de monitoreo de redes para soportar estudios de disponibilidad. La propuesta se desea plantear la realización de un prototipo de software que integrara dos funciones que en la mayoría de los sistemas comerciales y de software libre tienen por separado, y que en muchos casos su alto costo hace difícil su adquisición.

El sistema realizará dos funciones: Primero la aplicación tomará las bitácoras de eventos de los servidores Windows y realizará una estadística de fallas y tipos de errores que se están dando en el sistema y pueden ocasionar problemas en el funcionamiento del servidor; con esta información se puede saber que tan factible es que ocurra un problema en el servicios que ofrecen las organizaciones.

Cabe resaltar que este proceso es automático con solo seleccionar un par de opciones en el sistema, por lo cual hace se muy sencillo, intuitivo

y amigable su uso. En caso que el sistema tenga un gran número de eventos registrados, que dificulte su visualización en la opción del reporte del sistema, se tiene la posibilidad de exportar dicha información a un Excel para así poder visualizar el gráfico del tipo que se desee, es decir, ya sea por barras, pie, etc.

En la segunda parte del sistema, la aplicación deberá así mismo realizar un monitoreo en tiempo real del tráfico que fluye a través de la red o hacia un servidor (cualquiera que sea el caso a monitorear), con la finalidad de contar con información oportuna para conocer el estado de la red, poseer informaciones de transferencia de total que nos permita medir el uso de los servicios así como la carga de la red.

Todo lo mencionado se visualizará en reportes de performance de la red y alertas en línea en caso de saturación. Posibles casos de cuello de botella en la red y servidores pueden ser prevenidos mediante su uso.

Toda la información se contendrá en una base de datos por lo cual se tendrá un registro histórico de la información que podrá ser accedida en cualquier momento.

La facilidad de uso de este módulo del sistema es un punto importante a resaltar, así como los protocolos ya definidos en el sistema que son de mayor uso por los usuarios de internet, lo que mejora y profundiza el determinar cual es la cantidad de tráfico que fluye de su uso.

Al no ser un sistema que abarca un gran sin número de protocolos hacemos que el administrador tengo en forma más rápida y sencilla información especifica, sin tener que hacer un análisis y búsqueda compleja como es el casos de las demás soluciones mencionadas anteriormente en la tesis, que brindan información en bruto y no esta

filtrada, haciendo más arduo, lento y en muchos casos dejado de lado, por parte del administrador.

Así mismo sus reportes ayudan a dar análisis de tendencia de uso de protocolo y aplicaciones, sus horas picos y consumo de ancho de banda en el tiempo.

5.1 Motivación

En la actualidad como se menciona en capítulos previo, las organizaciones cuentan con una serie de sistemas internos y servicios que ofrecen a sus consumidores, a mismo estas organizaciones hacen alianzas estratégicas que les ayuden a competir mas fuertemente en el mercado globalizado de hoy en día. Dichas organizaciones se concentran más en los servicios que ofrecen y en el aumento de su productividad que a veces dejan de lado una parte importante de toda organización, es decir, su red de comunicación. Es por ello que un problema que tiene toda organización es el control de que es lo que esta fluyendo a través de su red, “el monitoreo de la red”.

En capítulos previos, se explica que es lo que significa monitorear una red; el que monitorear, el porque monitorear y hacia que hacer un monitoreo. En este sentido usaremos la tecnología del protocolo SNMP para hacer un sistema personalizado de monitoreo de red y análisis de sucesos de eventos de servidores en tiempo real. Con lo cual conseguiremos integrar dos funcionalidades que a pesar de existir actualmente, se muestran por separado o su costo es relativamente alto.

5.2 Arquitectura conceptual de herramienta integra de monitoreo para soportar estudios de disponibilidad

A continuación se muestra la arquitectura conceptual propuesta para la utilización de la estrategia de herramienta integra de monitoreo para soportar estudios de disponibilidad (Figura 18).

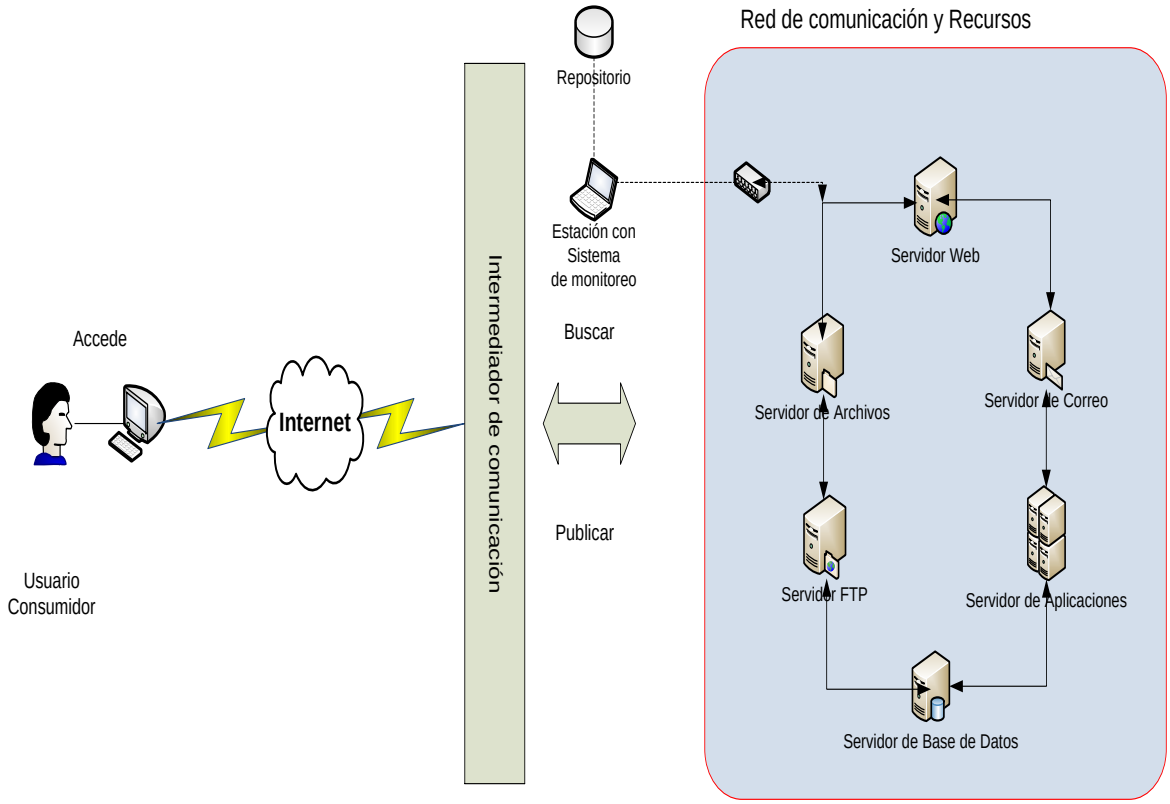


Figura 11: Arquitectura conceptual del herramienta integrada de monitoreo de redes para soportar estudios de disponibilidad.

5.3 Caracterización del Problema

El problema radica en que no existe en el mercado una herramienta integrada de monitoreo de red y análisis de sucesos de eventos de servidores. Este sistema involucra funcionalidades que pueden ser usadas en cualquier ambiente de topología de red. La aplicación esta desarrollada en dos lenguajes de programación totalmente compatibles como son Visual Basic 6.0 y Visual Basic .Net 2003, teniendo como base de datos Microsoft Access 2003, pudiéndose usar SQL server 2000 si así se desease. Es un sistema totalmente extensible que puede ir nutriéndose de más características según sea solicitado., permitiéndose el desarrollo de un sistema más completo y robusto.

5.4 Descripción de la solución

La solución al problema descrito anteriormente, requiere la utilización de la tecnología SNMP para la realización del monitoreo y análisis del trafico de red. También se usará tecnología de visor de suceso de eventos de los servidores bajo la plataforma Windows para analizar los tipos de eventos, su origen y posibles fallas.

5.5 Creación del software

El presente programa esta escrito en el lenguaje de programación Visual Basic 6.0 y Visual Basic .NET, tendiendo como base de datos para el almacenamiento de información Microsoft Access 2003 (es posible usar una base de datos de mayor capacidad de almacenamiento, confiabilidad y performance como Microsoft SQL Server 2000 Enterprise). El programa esta compuesto de los siguientes módulos:

- **Módulo de control visor de sucesos**, representa la primera parte del sistema. Aquí el sistema muestra una pantalla mediante la cual se debe de seleccionar si se desea cargar automáticamente los eventos de aplicación o de sistema del servidor.

  Una vez realizada la carga de las bitácoras de eventos, el sistema lo que hace es generar dos tipos de reportes: Por origen de suceso, y Por tipo de suceso. *Por Origen de suceso*, se muestra gráficamente cual fue la aplicación que tuvo mayor incidencia en el servidor y la cantidad de veces que esta se origino. *Por tipo de suceso*, se visualiza de qué clase es la incidencia que se origino, ya sea de tipo advertencia, de error o información.

  Este ultimo tipo de reporte también muestra la cantidad de tipos de sucesos que se origino. Estos análisis de incidencias para que tengan mayor relevancia deben ser tomados en un periodo de tiempo, que debe ser establecido por el administrador de red o encargado, con la finalidad de poder predecir si es hay algún tipo de aplicación que pueda generar un problema mayor en un largo plazo si es que no se toman medidas correctivas pertinentes.

  Así mismo dichos análisis de log de eventos pueden ser impreso, exportado a formatos: PDF, Excel, DOC y RTF.

  Esta parte del sistema también contempla la posibilidad de hacer un refresco del tipo de evento seleccionado (aplicación o sistema), con los cual se actualizan las ocurrencias que puedan darse en el log seleccionado.

El sistema al usar una base de datos como medio almacenamiento permite la opción de tomar varias muestras en diversos días y verlas para hacer un análisis cuando se desee, mediante la selección de la fecha del log que se desee visualizar. Ver **Anexo 1**.

El código de implementación de este módulo del sistema esta en el **Anexo 6**.

- **Módulo de simulación,** representa la segunda parte del sistema. Aquí se realiza el monitoreo de tráfico de red en tiempo real, el cual debe ser previamente configurado para que monitoree un periodo de tiempo especifico con intervalos de medición a ser seleccionados, así como cantidad de nodos que se desea.

  El sistema lo que haces es detectar y seleccionar automáticamente la tarjeta de red del servidor o si se desea ser puede colocar en punto de la red donde se sabe que pasa todo el tráfico de red ya sea para un servidor en particular o toda la red.

  Una ves empezado el monitoreo el sistema registra automáticamente todos el tráfico concerniente a los protocolos definidos a ser monitoreados por el sistema, entre los cuales tenemos los siguientes protocolos: HTTP, HTTPS, FTP, SMTP, DNS, ICMP, POP3, ICQ, IRC, MSN Messenger, mostrado una gráfica de vida en tiempo real que variará conforma haya mayor o menor tráfico que circule sobre esos protocolos.

Una vez finalizado el monitoreo el sistema cuenta con tres tipos de reportes: Tiempo total de tráfico, tiempo total de tráfico por protocolo y tiempo total de tráfico por IP.

El reporte de *Tiempo Total de Tráfico*, muestra el resultado global del tráfico en el tiempo e intervalo establecido de monitoreo.

El reporte de *Tiempo Total de Tráfico por Protocolo*, se muestra cuales son protocolos que causaron mas tráfico, y

Tiempo total de Tráfico por IPs, se muestra las IPs que tuvieron mayor comunicación en el periodo de monitoreo.

Al ser toda la información automáticamente almacenada en la base de datos, esta puede ser usada para analizar un periodo de tiempo de días, pudiendo determinar así una tendencia de tráfico en el tiempo, lo que permitiría al administrador tomar las medidas correspondientes del caso, ya sea para mitigar el uso de recursos que no son indispensables para el funcionamiento de la organización, y maximizar su uso para los servicios que esta ofrece. Ver **Anexo 2**.

El código de implementación de este módulo esta en el **Anexo 6**.

5.7 Arquitectura física de herramienta integrada de monitoreo de red para soportar estudios de disponibilidad

A continuación mostramos la arquitectura propuesta para una herramienta integrada de monitoreo de red para soportar estudios de disponibilidad (Figura 19).

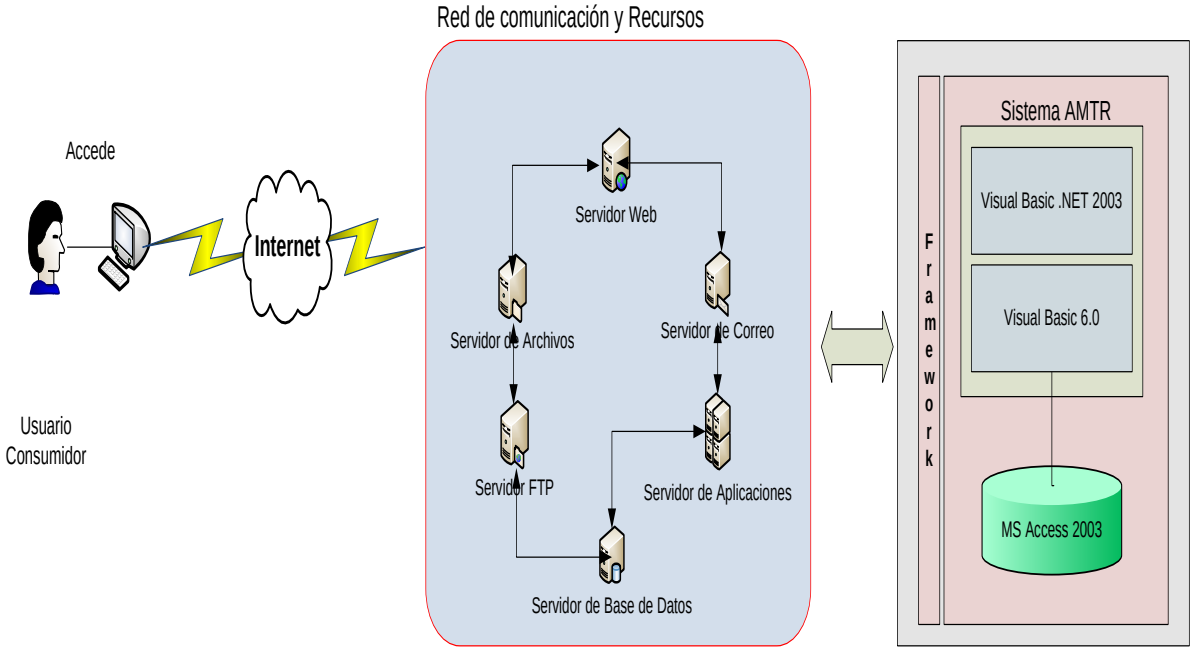


Figura 12: Arquitectura física de Herramienta integrada de monitoreo de red para soportar estudios de disponibilidad.

La arquitectura física esta compuesta de la siguiente manera:

(1) sistema AMTR, sistema con las funcionalidades para el monitoreo de red y análisis de eventos en servidores Windows. Desarrollado bajo las plataformas Microsoft Visual Basic 6.0 y Visual Basic .NET 2003, contando con un repositorio de base de datos Microsoft Access 2003. (2) los componentes WinSock, System.dll y EventLog, cuyas funcionalidades son usadas en el monitoreo de red y captura de eventos de servidores. (3) El uso de la tecnología Microsoft .NET Framework en

conjunto con la plataforma Visual Basic .NET para la funcionalidad de registro de eventos de sucesos de los servidores Windows.

5.8 Vista de Arquitectura

La Vista de Arquitectura presenta la vista lógica de los componentes del sistema.

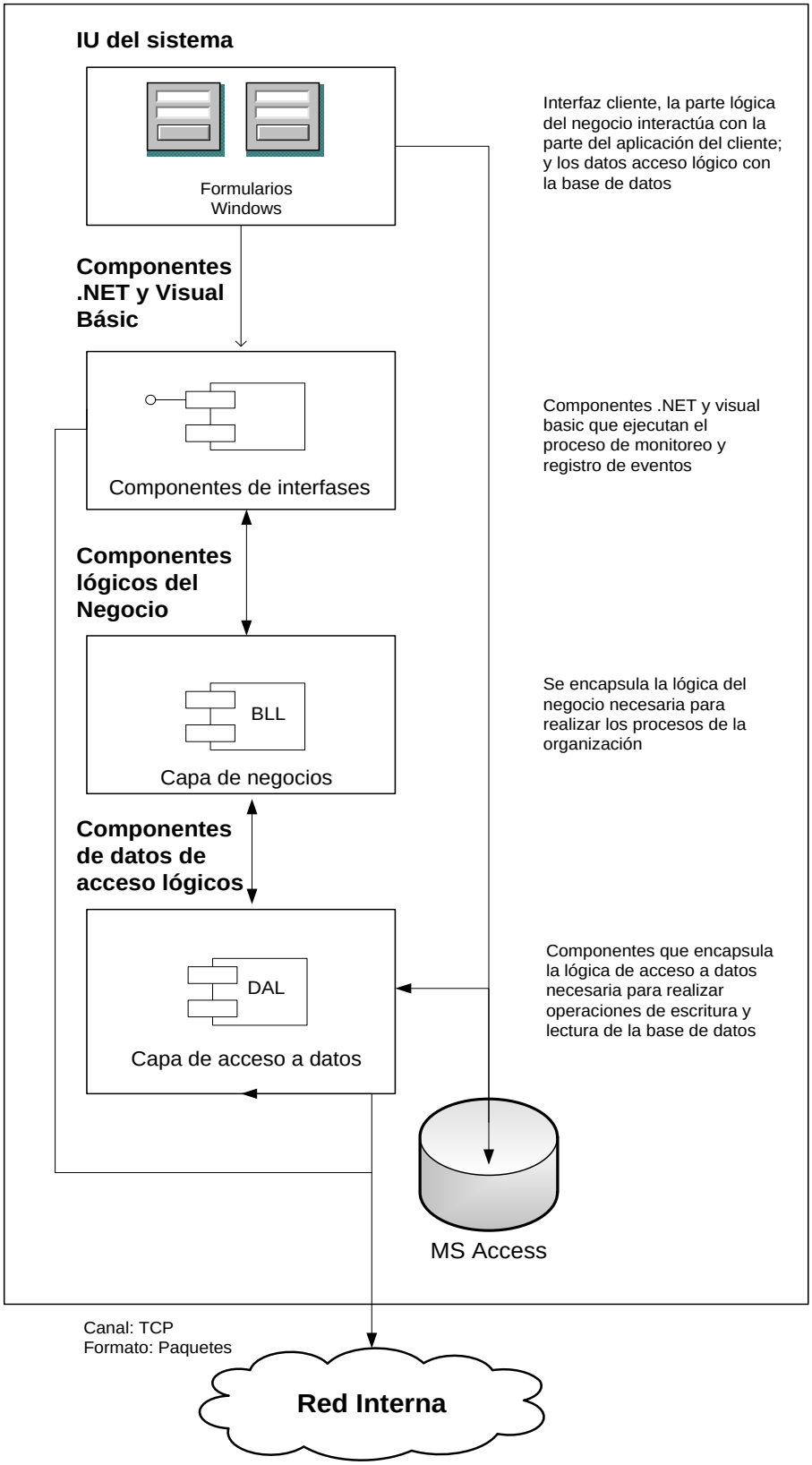


Figura 13: Vista de la arquitectura del sistema.

5.9 Vista de Despliegue

La Vista de Despliegue presenta los nodos físicos y la configuración utilizada por el sistema.

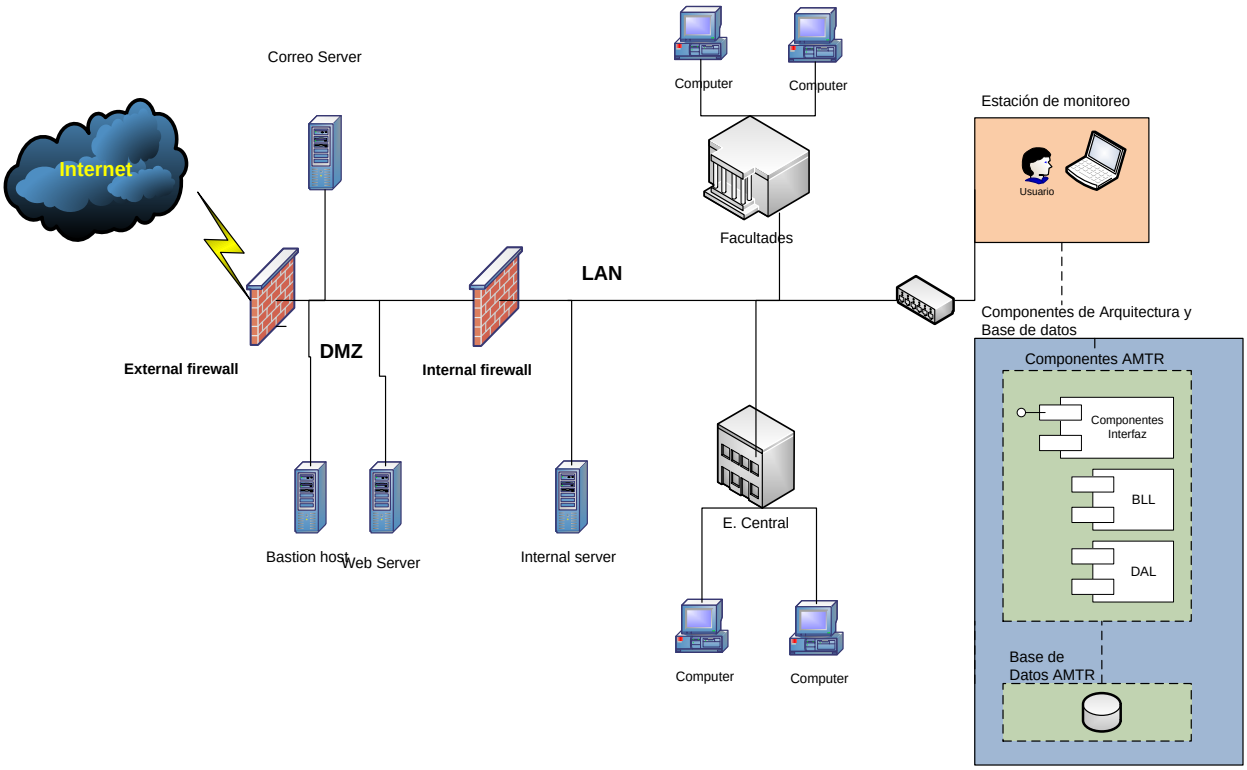


Figura 14: Vista de despliegue del sistema.

Escenario del despliegue

El presente sistema AMTR, no esta pensado para un escenario en particular, ya que es un sistema extensible y adaptable a cualquier ambiente de red de organización sobre el cual se desee usar. Inicialmente por motivos de accesos, será usado en la red de la Universidad Ricardo Palma, en un segmento de red en particular.

El esquema de despliegue seria una PC Workstation o Laptop, sobre el cual el sistema es cargado para funcionar. Mediante una configuración especial llamada Port Mirroring en un Switch de red, haremos que la PC Workstation o Laptop, pueda leer todo el tráfico que circula por la red o

segmento de red a monitorear, como resultado la aplicación registrara los eventos de tráfico resultantes del monitoreo de red realizado.

5.10 Vista de Implementación

La Vista de Implementación describe los componentes que implementan las clases y lógica del modelo de diseño físicamente, definiéndolos en capas y jerarquías, ilustra, además las dependencias entre éstos. El sistema se implementará en tres capas, las cuales se describen en el siguiente gráfico:

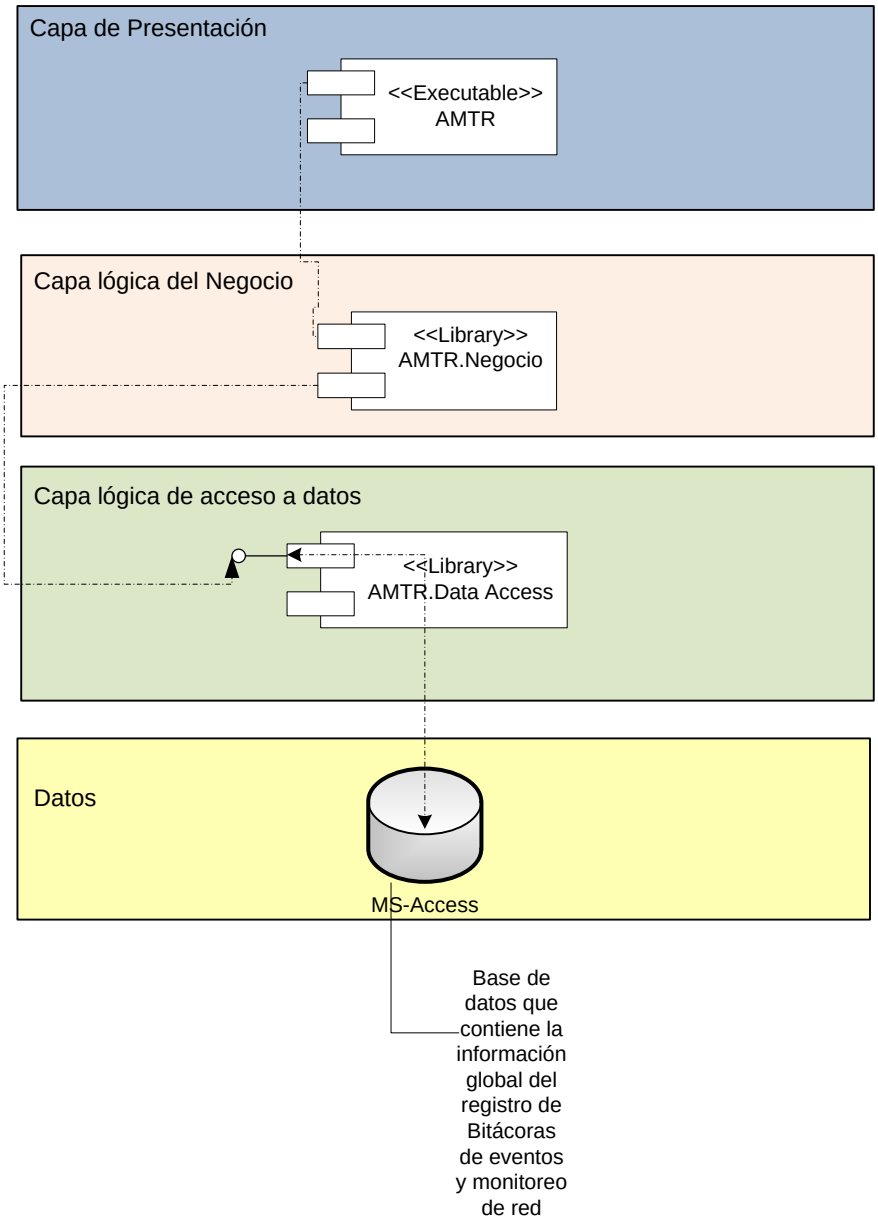


Figura 15: Vista de implementación del sistema.

Capa de presentación

- **AMTR** aplicación Windows, interfaz de usuarios que usan el usuario de la aplicación del sistema.

Capa lógica del negocio

- **AMTR.Negocio**, Componente DLL que contiene las clases e interfaces que encapsulan la lógica del negocio de la aplicación.

Capa de lógica de acceso a datos

- **AMTR.DataAccess,** Componente DLL que contiene las clases e interfaces que encapsulan la lógica de acceso a datos de nuestra aplicación.

Datos

- **Access DB,** Base de datos que contiene la información global del registro de Bitácoras de eventos y monitoreo de red

En resumen en este capítulo hemos visto la problemática por la cual estamos desarrollando la herramienta de monitoreo de red para soportar estudios de disponibilidad, hemos mostrado la arquitectura conceptual y física del sistema, así como también escenario de despliegue dentro de la organización y como es que puede ser usado.

En el siguiente capítulo VI se describen las conclusiones, recomendaciones y trabajos futuros.

CAPITULO VI. CONCLUSIONES, RECOMENDACIONES Y TRABAJOS FUTUROS

6.1 Conclusiones

La realización del presente proyecto ha dado como resultado la creación de una herramienta integrada para el monitoreo de tráfico en tiempo real y control de visor de bitácoras, cuyos objetivos y ventajas son las de proporcionar la información del tipo de tráfico que circula por la red en base a los parámetros de protocolos predeterminados, análisis de bitácoras de los aplicativos que están funcionado en un servidor, y envió de mensajes de alerta en tiempo real. El presente resultado es el siguiente:

1. El módulo de visor de control de sucesos, que tenia como objetivo recolectar automáticamente las bitácoras de eventos de aplicación y sistema de Windows, ha sido realizado exitosamente:

- Se cumple con el objetivo de proporcionar una interfaz más amigable a los administradores de red y a la gerencia de TI para observar las incidencias de tipo error, advertencia y información ya sea del día en que se desea realizar el análisis o de un periodo de días. Esto ayuda al administrador a realizar un seguimiento de cuales podrían ser los problemas en caso de no atender o corregir procesos; o aplicativos que estén funcionando incorrectamente, marcando así una tendencia de funcionamiento en el tiempo.
- Otra de las funcionalidades que se tenia planteada era la exportación de los resultados del análisis de los logs a formatos pdf, xls, doc o rtf; a su como la opción de impresión.
- La visualización mediante dos tipos de reporte: por origen del suceso y por tipo del suceso, ya sea del día o periodo de días analizados.
- Envío de alertas en tiempo real, en caso se supere los umbrales definidos; funcionan adecuadamente alertando al administrador de posibles problemas. Dichos umbrales se pueden editar de acuerdo a las necesidades y realidades de la organización.

2. El módulo de Monitoreo de tráfico en tiempo real, cuyo objetivo es el de monitorear el tráfico en tiempo real y emitir reportes:
    - Se logro capturar el tráfico de 10 por protocolo de aplicaciones que están entre las más comúnmente usadas, entre ellas tenemos por ejemplo: http, ftp, pop3, smtp, msn Messenger, emule, entre otros más. Estos monitoreos son programables de acuerdo a la necesidad y tiempo que el administrador desee.
    - Visualización del tráfico capturado en tiempo real, mediante tres tipos de reportes: reporte de tiempo total

de tráfico, de tiempo total de tráfico por protocolo y tiempo total de tráfico por IP.

- Visualización de los reportes por día o rango de días, ayudando al administrador de red para determinar tendencias, ver el comportamiento y consumo de ancho de banda de red por dichos programas. Pudiendo así tomar las acciones del caso y mitigar su impacto en caso de ser aplicaciones no deseadas o maximizar su uso para aplicaciones propietarias de la empresa.

En conclusión puedo decir que la Herramienta integrada de monitoreo de redes para soportar estudios de disponibilidad, cumple con su función en base a los alcances propuestos de realizar un monitoreo de tráfico en tiempo real con emisión de reportes del trafico monitoreado de la fecha o rango de fechas, y envío de mensajes de alerta en caso se sobre pase los umbrales de consumo de tráfico establecidos; a su ves se realiza un análisis de eventos de los sucesos que ocurren en el servidor monitoreado.

Así mismo podemos decir que la información que proporciona el sistema, ayudara a tomar decisiones y a evaluar estudios costo beneficios como por ejemplo, en aumentos en el ancho de banda de la red, mejor eficiencia en la productividad del trabajador al tener un mayo control de su consumo y uso de internet, mayor performance y respuesta los procesos internos que ejecuta la organización, ya que la red estará mas disponible al tener un mejor control y restricciones de la red. Toda esta información recolectada por el sistema ayudara a que la gerencia del negocio y de TI puedan tomar las decisiones que mejor vayan acorde con las necesidades del negocio.

También cabe resaltar que una aplicación de este tipo pude ser adaptada de acuerdo a las necesidades del negocio y cuyo costo de

realización es significativamente menor a las soluciones que hay en el mercado.

6.2 Recomendaciones y trabajos futuros

- La información resultante del monitoreo y análisis de servidores, permite facilitar a la gerencia del negocio la tomar las decisiones sobre temas como el aumento de ancho de banda de la red.
- Usar la presente tesis para realizar mayores avances en el método de monitoreo de redes haciéndolo más efectivo y preciso en la detección del tipo de tráfico de red.
- Hacer que la herramienta en cuestión pueda ser accedida remotamente a través de una interface web.
- Ampliar el campo de monitoreo de red para que pueda soportar esquemas de redes distribuidas y Wireless.
- Interfaz de Administración remota a través de la web, para la realización del monitoreo de red.

ANEXO 01

MANUAL DE DISEÑO DEL SISTEMA

1. Introducción

El objetivo del presente manual es el de describir el sistema tanto en su parte de datos (modelos entidad/relación, diccionario de datos), como en la parte de análisis y diseño, usando para esto la notación del UML, con los diagramas de Casos de Uso y Secuencia, así como haciendo un listado y breve descripción de programación en Visual Basic que componen el aplicativo.

2. Alcances

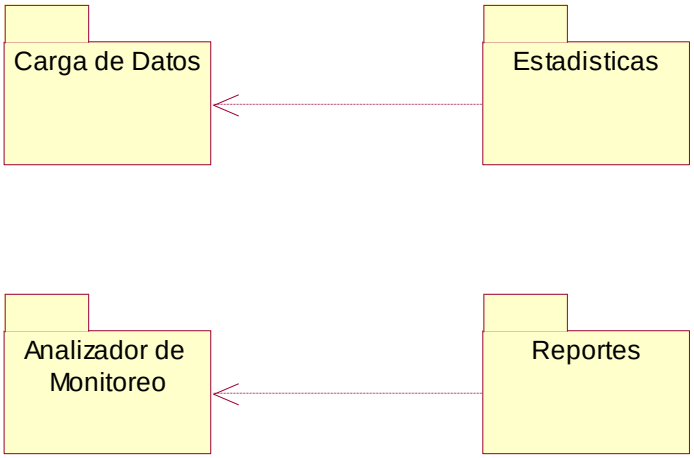
Este manual esta dirigido al personal del área de Soporte técnico y administradores de infraestructura de red, cuyas actividades se encuentren vinculadas a la mejor performance de servidores y servicios de red.

3. Modelamiento del Sistema: Diagramas UML

Los diagramas realizados para el modelamiento del sistema son los que se muestran a continuación en el siguiente orden:

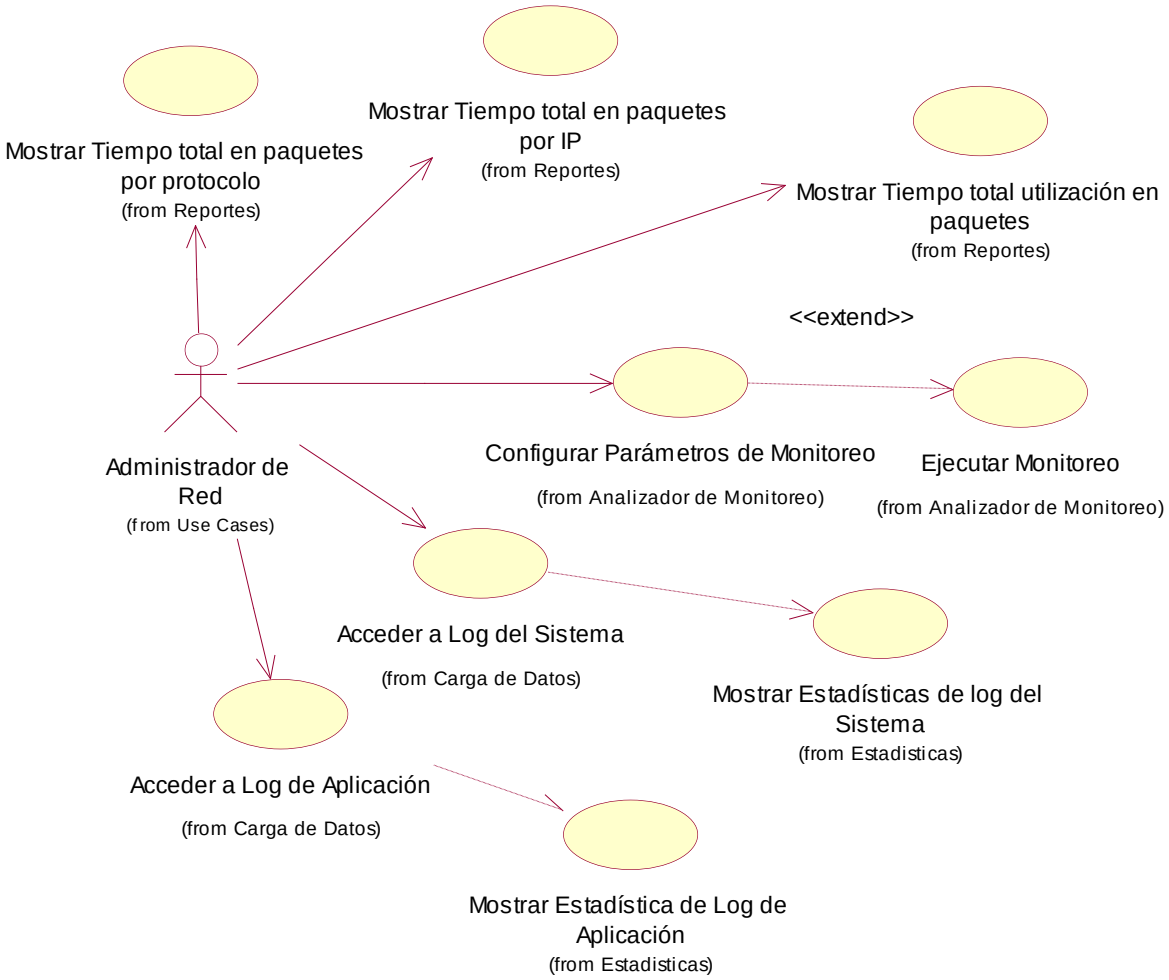
3.1 Diagrama de Paquetes

Diagrama de Paquetes



3.2 Diagramas de Casos de Uso

DIAGRAMA DE CASOS DE USO GENERAL



3.2.1 Caso de Uso: Control del Visor de Eventos del Sistema / Aplicación

1.-Caso de Uso del Sistema		Control del Visor de Eventos del Sistema / Aplicación
2.- Descripción		
El administrador de red accederá al módulo control de Visor de sucesos, donde elegirá la carga del archivo que contiene los datos del log de eventos del sistema o aplicación del servidor a ser evaluado.		
3.- Actor(es)		
Administrador de Red.		
4.- Precondiciones		
Sistema debe ser instalado en servidor a evaluar.		
5.- Poscondiciones		
Reporte de Estadísticas de Log de Eventos del Sistema / Aplicación.		
6.- Flujo de Eventos		
Nro	**Acción del Actor**	**Respuesta del Sistema**
1	El administrador de red ingresa al módulo logs y elige visualizar ya sea los logs del sistema o aplicación.	El Sistema hará un escaneo y mostrará todos los eventos que del día o según la fecha del evento que se desee visualizar.
7.- Requerimiento asociado		
No existe		
4.- Prototipo de interfaz de usuario		

3.2.2 Caso de Uso: Reportes de Log del Sistema / Aplicación

1.-Caso de Uso del Sistema	Reportes de Log del Sistema / Aplicación	
2.- Descripción		
El administrador de red visualizara los reporte previamente cargados de acuerdo a una fecha de calendario en que fue realizada dicha carga.		
3.- Actor(es)		
Administrador de Red.		
4.- Precondiciones		
Haber hecho la carga previamente una carga de eventos de sistema o aplicación.		
5.- Poscondiciones		
Reportes de Estadísticas de Log de Eventos de Aplicación.		
6.- Flujo de Eventos		
Nro	**Acción del Actor**	**Respuesta del Sistema**
1	El administrador de red elegirá una fecha de calendario que contenga archivos que han sido cargados previamente.	El Sistema mostrara un reporte grafico del resultado de dicha carga, pudiendo ser visualizada por código de error o mensaje de error.
7.- Requerimiento asociado		
Haber realizado una carga de log de eventos en el sistema.		
4.- Prototipo de interfaz de usuario		

3.2.3 Caso de Uso: Monitoreo de Tráfico en Tiempo Real

1.-Caso de Uso del Sistema	Monitoreo de tráfico en Tiempo Real
2.- Descripción	
El administrador de red eligira los parámetros de monitoreo de tráfico en tiempo real.	
3.- Actor(es)	
Administrador de Red.	
4.- Precondiciones	
Ninguna.	
5.- Poscondiciones	
Reporte de tiempo total en paquetes Reporte de tiempo total en paquetes por protocolo Reporte de tiempo total en paquetes por IP	
6.- Flujo de Eventos	

Nro	**Acción del Actor**	**Respuesta del Sistema**
1	El administrador de red eligira los parámetros de simulación: duración de la simulación, puntos de medición y Nº de nodos participantes.	El Sistema realizara el monitoreo en base a los parámetros especificados.

7.- Requerimiento asociado
No existe.
4.- Prototipo de interfaz de usuario

3.2.4 Caso de Uso: Reportes de monitoreo de Tráfico en Tiempo Real

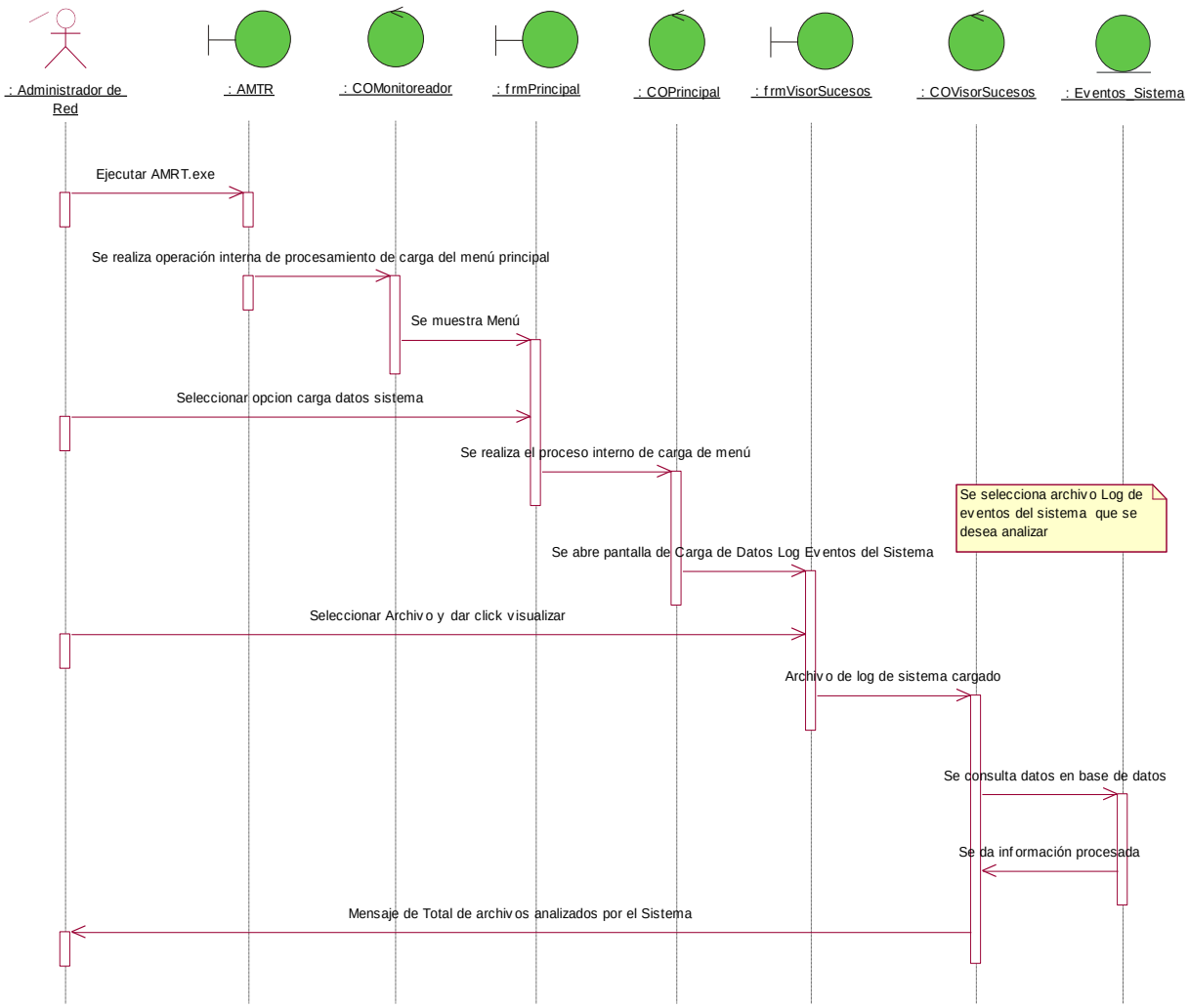
1.-Caso de Uso del Sistema	Reportes de monitoreo de Tráfico en Tiempo Real.
2.- Descripción	
El administrador de red podrá visualizar 4 tipos de reportes de acuerdo a sus requerimientos.	
3.- Actor(es)	
Administrador de red	
4.- Precondiciones	
Haber realizado un monitoreo de Tráfico en Tiempo Real.	
5.- Poscondiciones	
Ninguna	
6.- Flujo de Eventos	

Nro	**Acción del Actor**	**Respuesta del Sistema**
1	El administrador de red selecciona entre los tipos de reporte preestablecidos: tiempo total en paquetes, por IP y protocolo.	El Sistema mostrará el reporte elegido con los datos de monitoreo de tráfico.

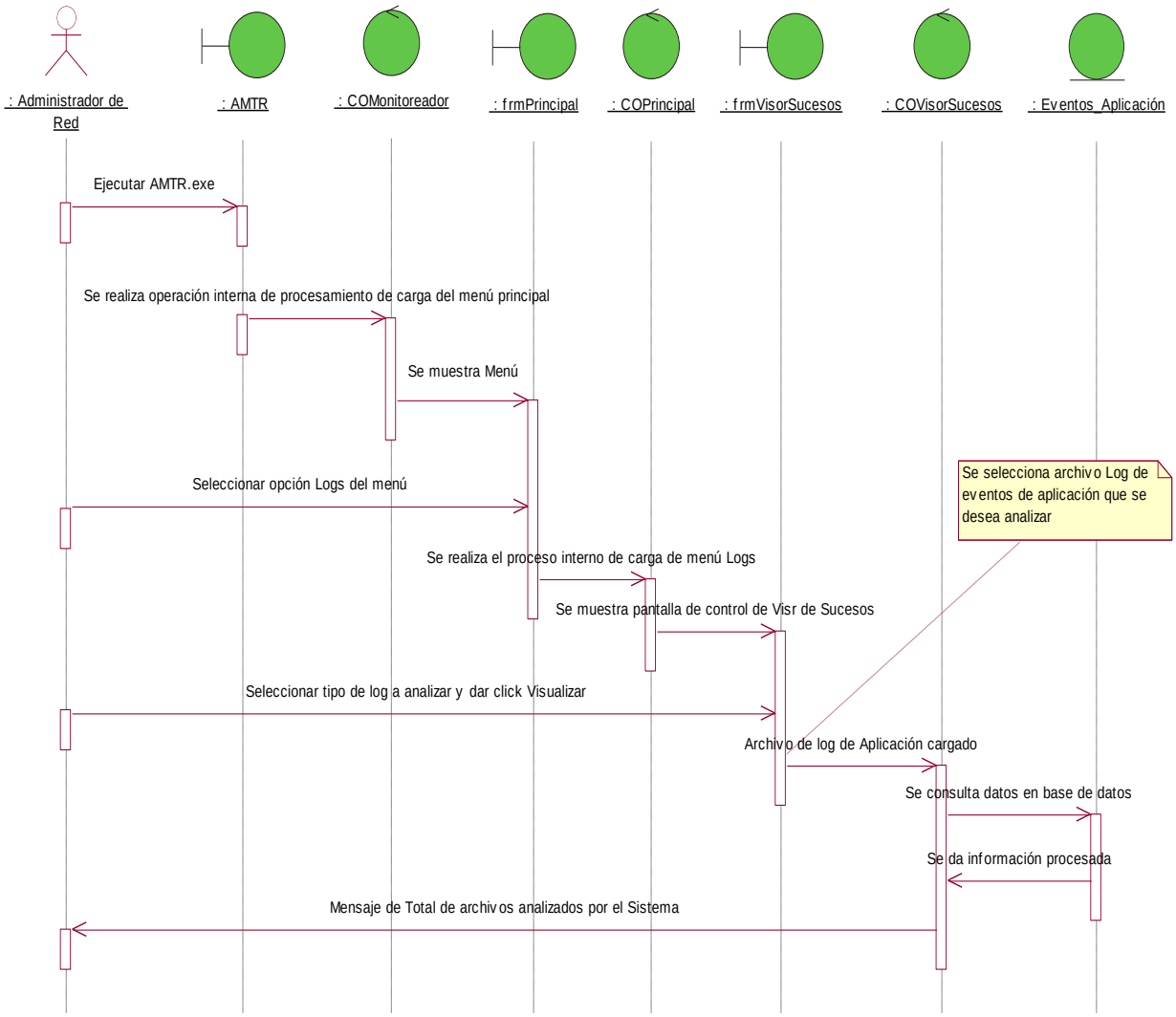
7.- Requerimiento asociado
Haber ejecutado un monitoreo de tráfico de red previamente.
4.- Prototipo de interfaz de usuario

3.3 Diagramas de sistema

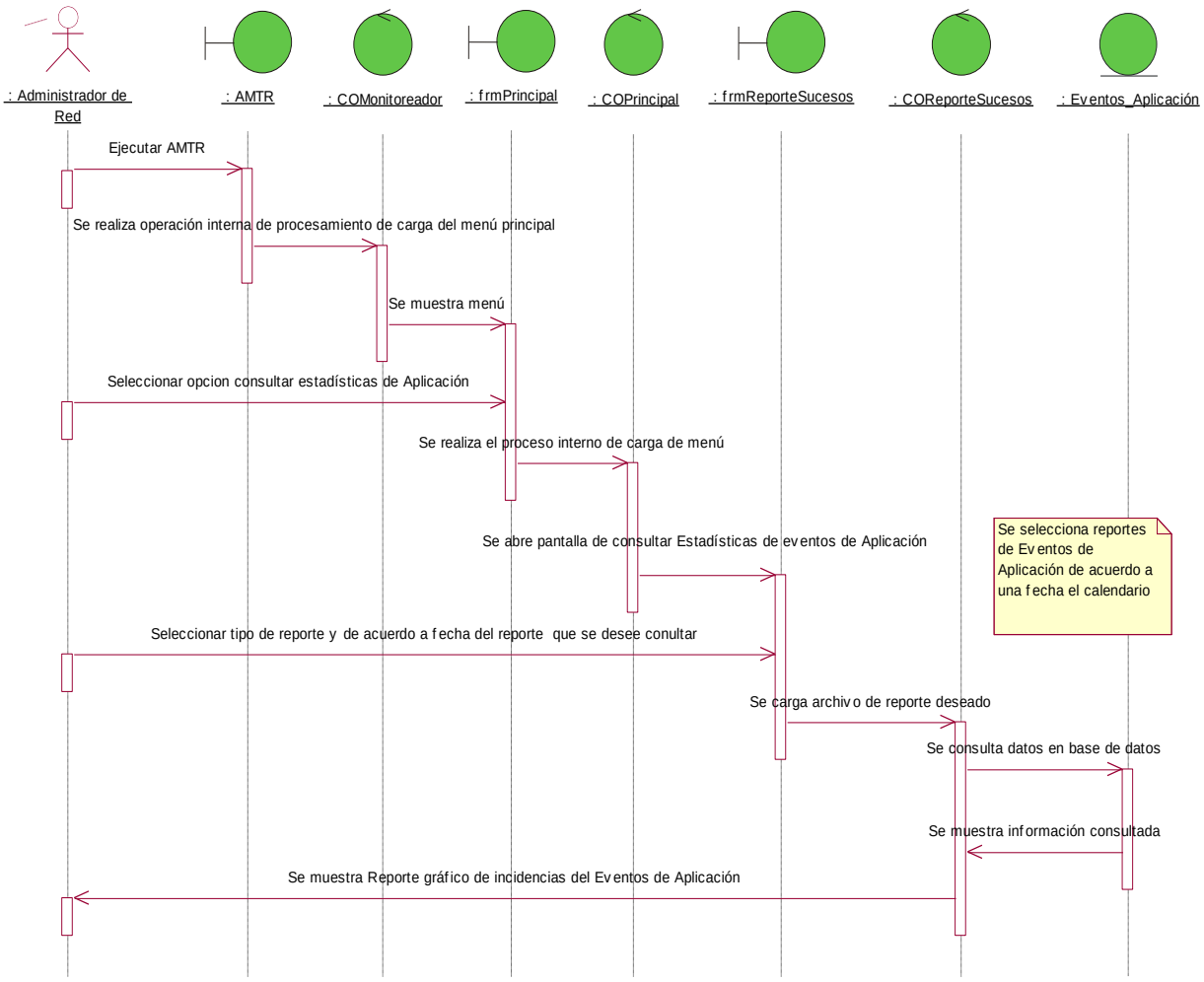
3.3.1 Diagrama de Secuencia: Control de Visor de Sucesos Sistema



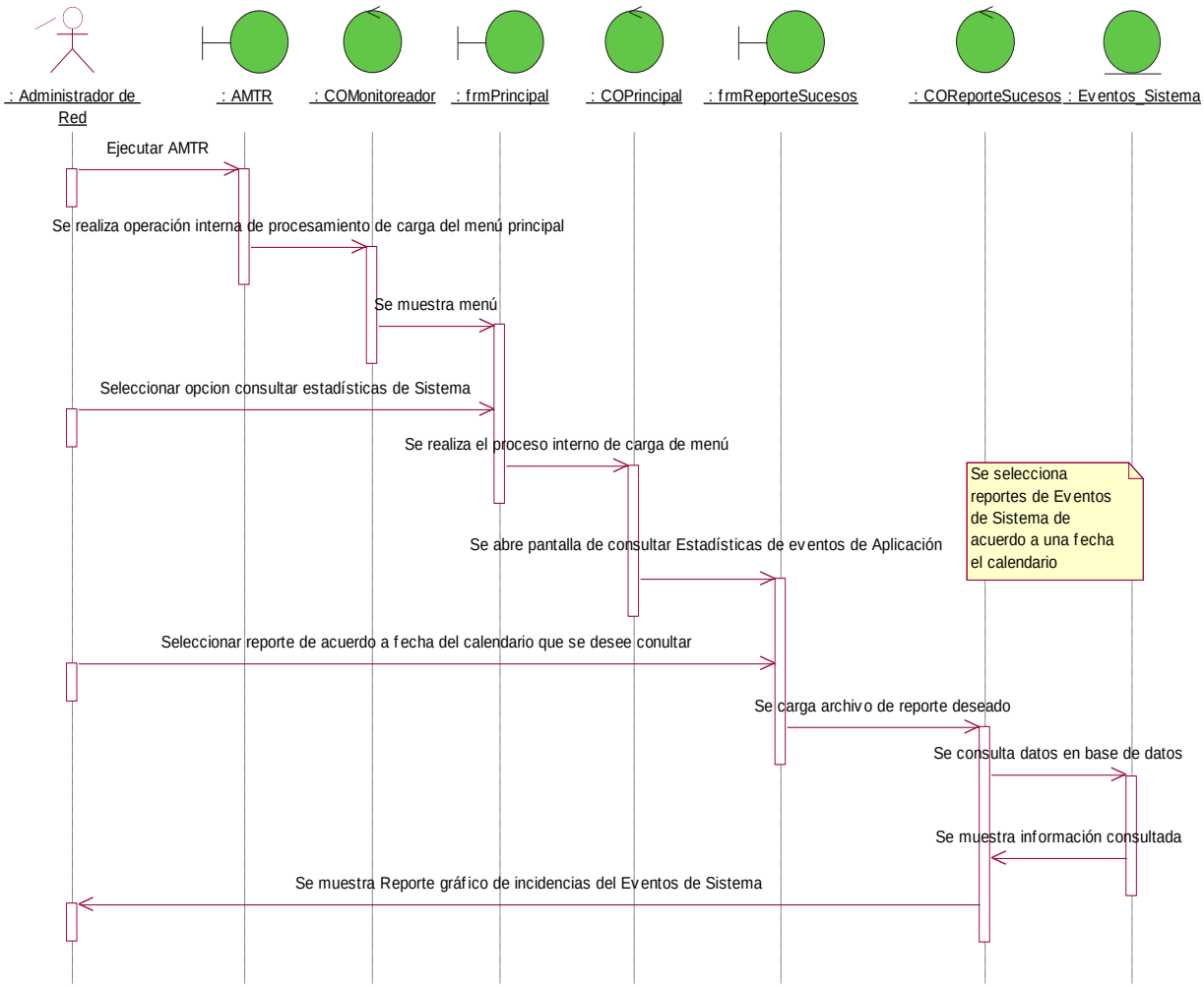
3.3.2 Diagrama de Secuencia: Control del Visor de Sucesos Aplicación



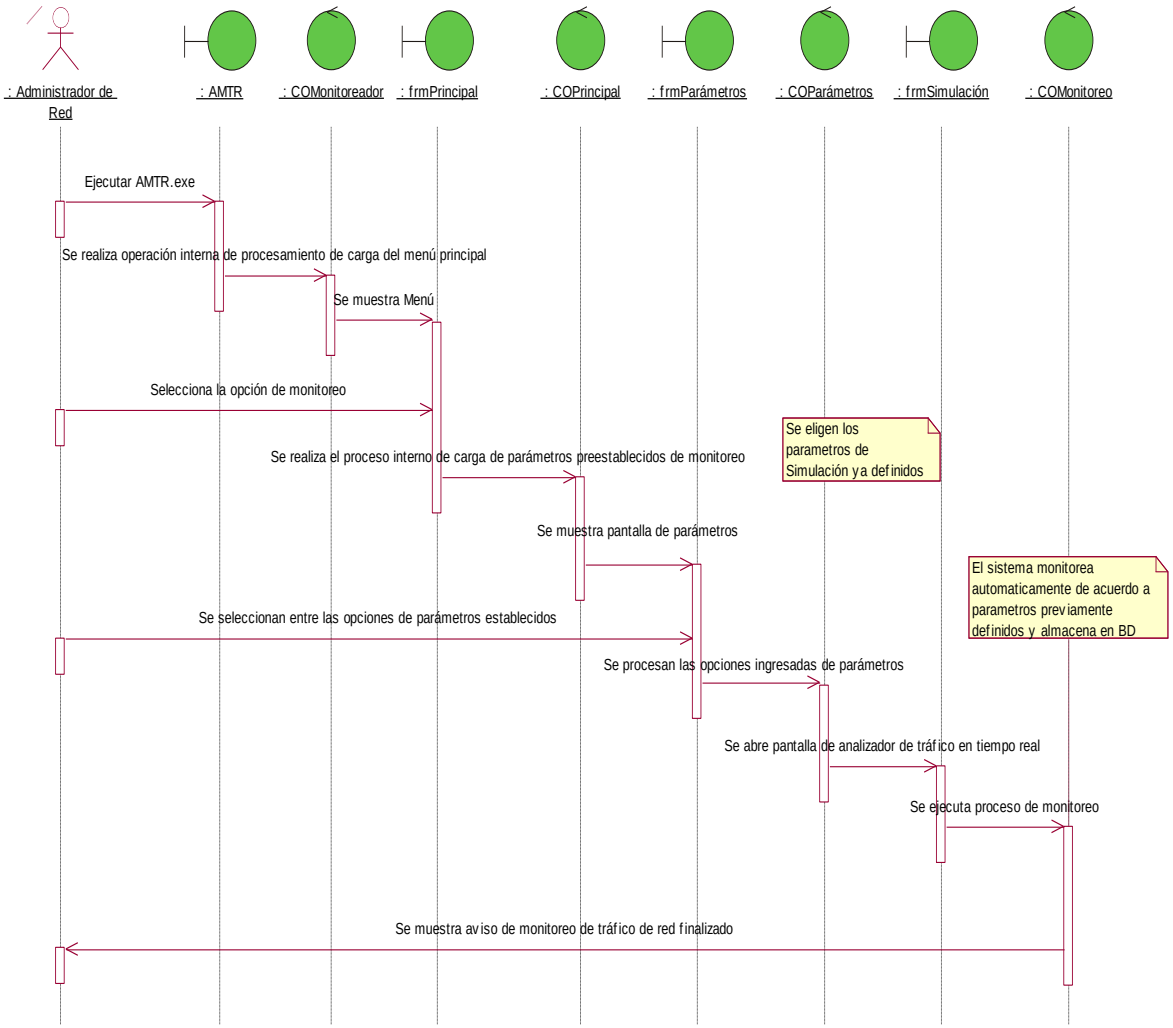
3.3.3 Diagrama de Secuencia: Consultar Reporte de Aplicación



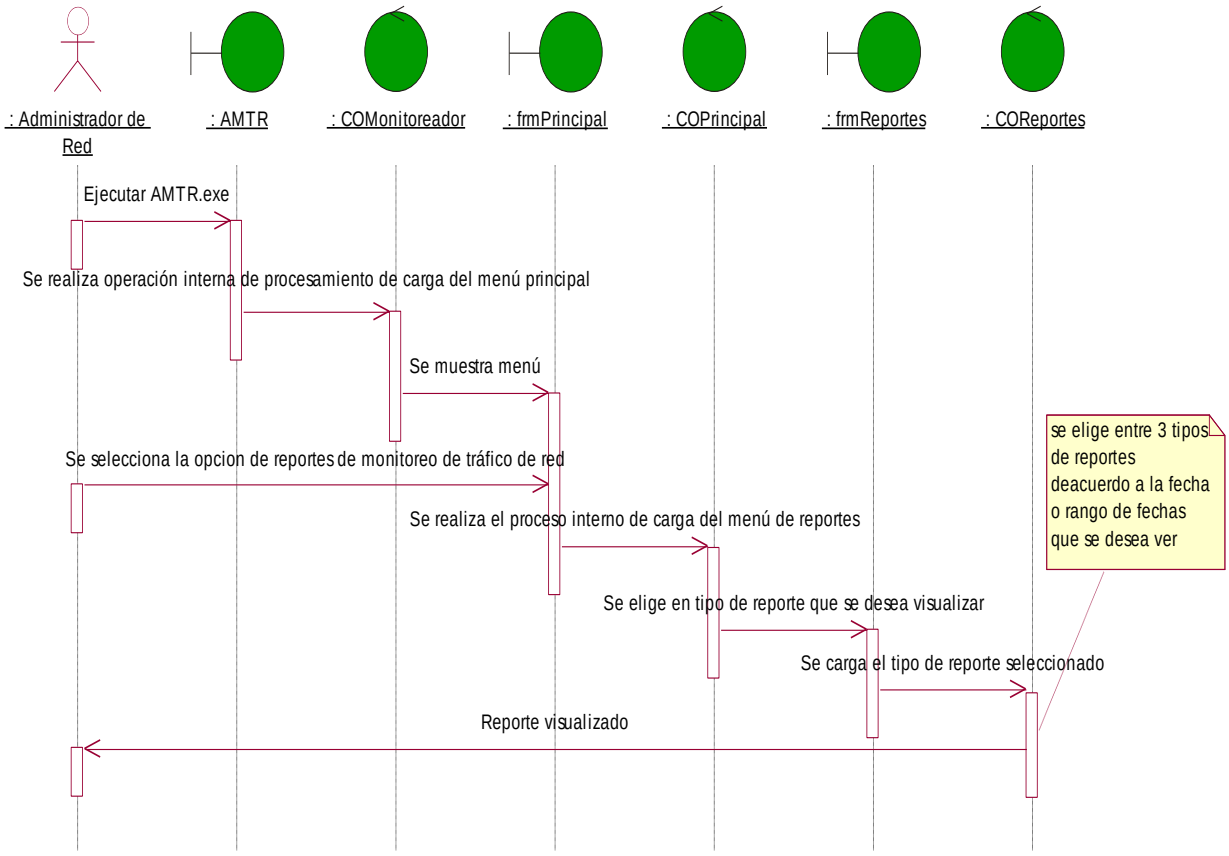
3.3.4 Diagrama de Secuencia: Consultar Reportes del Sistema



3.3.5 Diagrama de Secuencia: Analizador de Monitoreo de tráfico en tiempo real

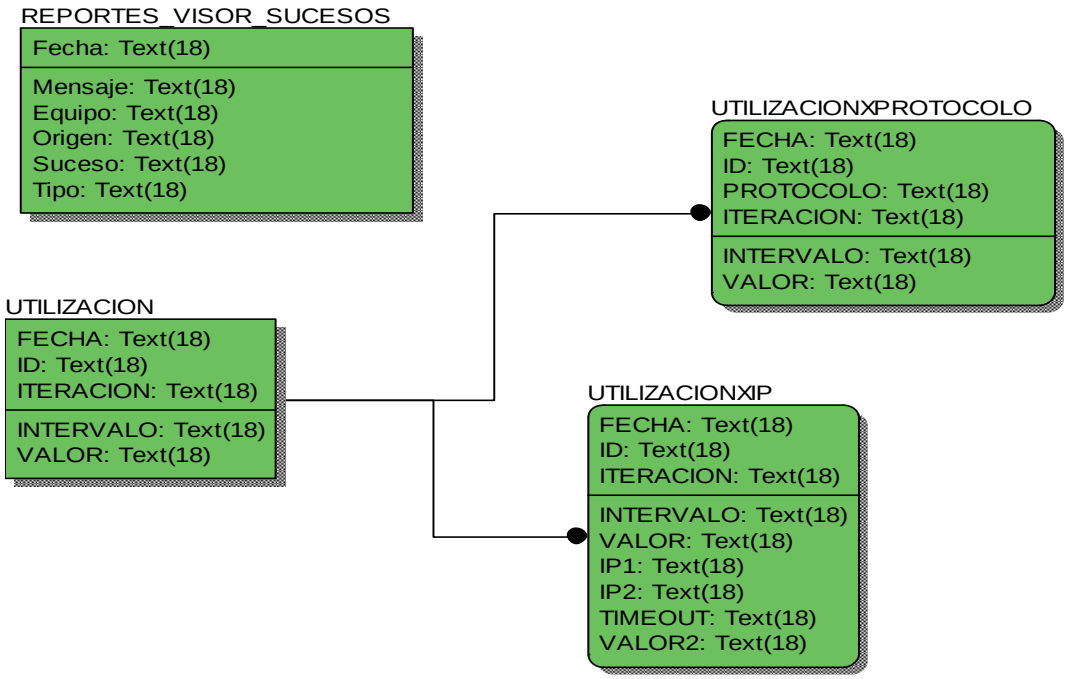


3.3.5 Diagrama de Secuencia: Consultar reportes de monitoreo de tráfico de red



4. Diccionario de Datos - Base de Datos

4.1 Modelo Físico de Datos



4.2 Lista de Tablas

Nombre	Descripción
EVENTOS_VISOR_SUCESOS	Información sobre eventos de aplicación registrados en servidores
UTILIZACION	Datos registrados de monitoreo de tráfico de red
UTILIZACIONXIP	Datos de utilización de tráfico por IP
UTILIZACIONXPROTOCOLO	Datos de utilización de tráfico por protocolo

4.3 Descripción de las Tablas del Sistema

4.3.1 Tabla_reportes_visor_sucesos

Nombre	Código	Tipo
Fecha	Fecha	Texto
Mensajes	Mensajes	Texto
Equipo	Equipo	Texto
Origen	Origen	Texto
Suceso	Suceso	Texto
Tipo	Tipo	Texto

4.3.2 Tabla_utilización

Relación de porcentaje de utilización total en paquetes.

Nombre	Código	Tipo
Fecha	Fecha	Texto
ID	ID	Texto
Iteración	Iteración	Numero
Intervalo	Intervalo	Numero
Valor	Valor	Texto

4.3.3 Tabla_utilizacionxip

Contenedores por Declaración

Nombre	Código	Tipo
Fecha	Fecha	Texto

Nombre	Código	Tipo
ID	ID	Texto
Iteración	Iteración	Numero
Intervalo	Intervalo	Número
Valor	Valor	Texto
Fecha	Fecha	Texto
IP1	IP1	Texto
IP2	IP2	Texto
Timeout	Número	Número
Valor2	Valor2	Texto

4.3.4 Tabla_utilizacionxprotocolo

Datos de porcentaje de uso por protocolo.

Nombre	Código	Tipo
Fecha	Fecha	Text
ID	ID	Text
Iteración	Iteración	Number
Intervalo	Intervalo	Number
Protocolo	Protocolo	Text

5. Descripción de los objetos del sistema

5.1 Estructuras de datos

5.1.1 Formularios

Nombre	Descripción
FrmCargaDatos	Formulario de búsqueda de archivo de Log de eventos de Sistema y posterior carga.
FrmCargaDatos2	Formulario de consulta de archivos de Log de eventos de Aplicación y posterior carga.
FrmEstadistica	Formulario de consulta por calendario del Log de eventos cargados en el sistema.
FrmEstadistica2	Formulario de consulta por calendario del Log de eventos cargados en el sistema.
FrmEstadisticaUtilizacion	Formulario de consulta de los eventos registrados del monitoreo de tráfico en tiempo real realizado.
FrmEstadisticaUtilizacionXIP	Formulario de consulta de uso por IP.
FrmEstadisticaUtilizacionXProtocolo	Formulario de consulta de consumo de tráfico por protocolo.
FrmParametros	Formulario donde ingresan los parámetros de monitoreo preestablecidos.
FrmPrincipal	Formulario donde se muestra el menú del sistema.
FrmSimulador	Formulario para el monitoreo de tráfico en tiempo real.

5.1.2 Módulos

Nombre	Descripción
mdlapi (mdlapi)	Clases que comprenden el registro de monitoreo de red
mdlwinsock (mdlwinsock)	
mdmain (principal.bas)	
modping (modping.bas)	
modTimers (modTimers.bas)	

5.1.3 Módulos de clase

Nombre	Descripción
APITimer	Clase con agujas de medición para monitoreo de tráfico.
clsRequest	
colRequest	
SpeedometerGauge	

5.2 Diagrama de componentes

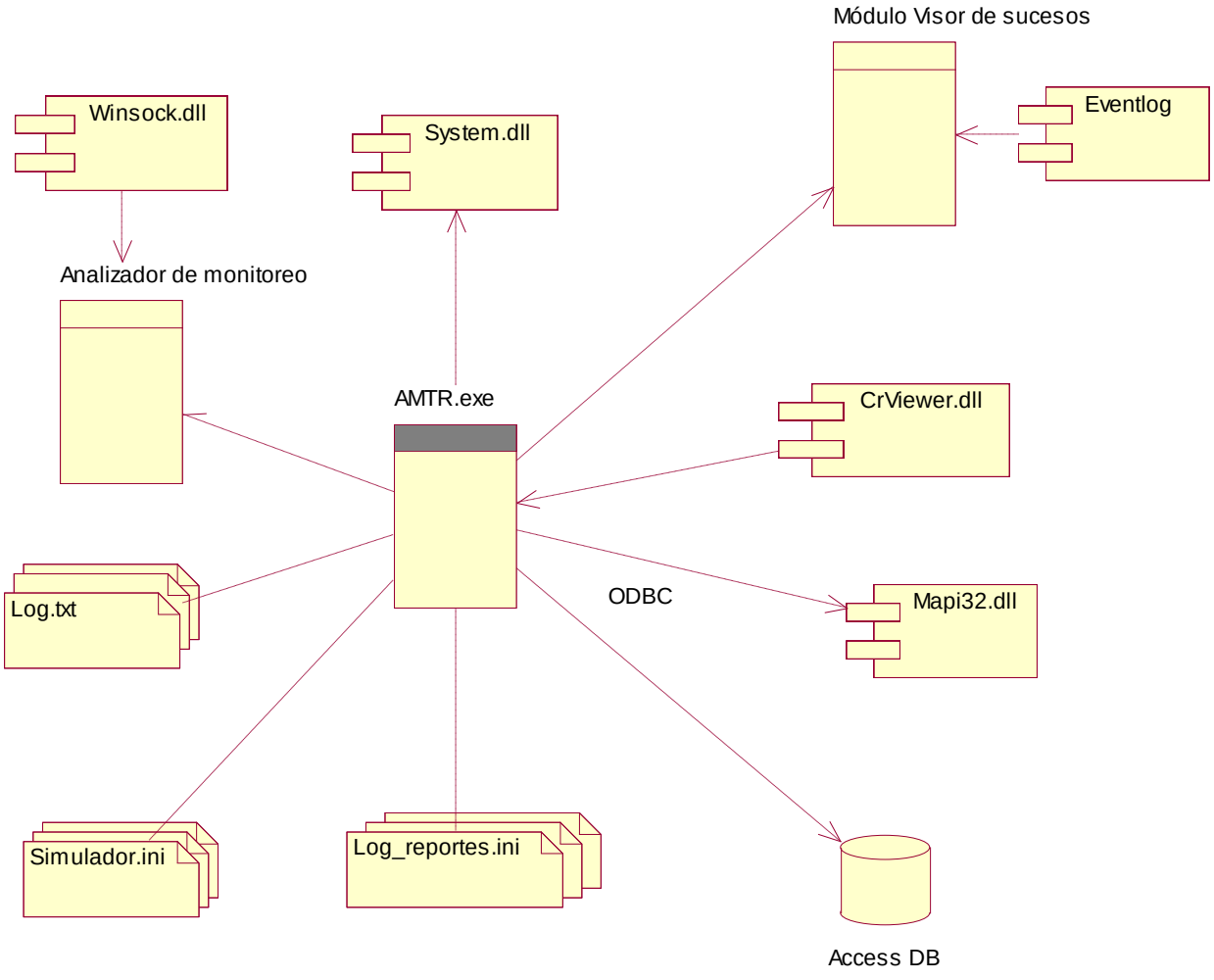
Entre los componentes usados en el sistema tenemos los siguientes:

- **System.dll**, componente que proporciona clases que permiten interactuar con los procesos del sistema, registros de eventos y contadores de rendimiento.
- Se uso el componente **EventLog** que proporciona la funcionalidad para escribir en registros de eventos, leer las entradas de los registros de eventos y crear y eliminar registros de eventos y orígenes de eventos en la red.
- **Winsock**, es el componente q permite escuchar en los distintos puertos (TCP, UDP, MSN, etc.) es una biblioteca de vínculos

dinámicos que proporciona unas interfaces de programación de aplicaciones comunes (API) para programadores de aplicaciones de red que utilizan la pila de Protocolo de control de transmisión/Protocolo de Internet.

- **Mapi32.dll**, es el componente para el envío de e-mail.
- **crviewer.dll**, componente de crystal reports para los reportes.

Diagrama de Componentes



5.3 Pruebas del sistema

Un punto muy importante después de haber creado el software es el pruebas del sistema, donde se someterá el producto a pruebas funcionamiento y carga de información para determinar si se cumplen con los requisitos mínimos con los que un software debe contar.

5.3.1 Requerimientos para prueba

Requerimientos funcionales (El requerimiento funcional mencionados debajo tienen relación directa con nuestro blanco de prueba).

El problema de	Tráfico excesivo y lentitud en la red
Afecta	A las aplicaciones de servidores y demás recursos compartidos que corren simultáneamente por la red.
El impacto es	Que el personal trabajador no podrá acceder con rapidez a los recursos y servicios que ofrece la organización.
Una solución exitosa sería	Desarrollar un módulo que permita al personal administrador de red ser informado mediante el monitoreo, alertas y reportes de cuales son los potenciales problemas que se podrían dar en caso de no ser atendidos.

5.3.2 Estrategia de pruebas

Exponemos a continuación las siguientes:

a. Caso de prueba de interfaz de usuario

Objetivos de la Prueba	Verificar, incluyendo pantallas, campos y uso de métodos de acceso (movimientos de Mouse, clic derecho y clic izquierdo)
Técnicas	1. Crear o modificar planes de prueba para cada pantalla verificando a través de la navegación pantalla a pantalla.
Criterio de Realización	Cada pantalla, campo y opción del sistema debe satisfactoriamente realizar la funciona para la cual esta predeterminada. Se combinara todas las formas posibles de opciones para determinar si hay posibles fallas.
Consideraciones Especiales	Se deberá cumplir con los estándares solicitados.

Ver documento **Anexo 3** Caso de Prueba del sistema.

b. Caso de Prueba de carga de Log de sistema / aplicación del servidor

Objetivos de la prueba	Verificar la conducta resultante al cargar los logs de eventos del Windows al sistema.
Técnica	Intentar forzar la carga archivos que no están en el formato establecido por el sistema.
Criterio de Realización	Multiplicar las cargas al sistema: completa satisfacción de la prueba sin alguna falla y con un tiempo aceptable.

Ver documento **Anexo 3,** figura 27 y 28 de caso de prueba del sistema.

c. Prueba de esfuerzo

Objetivo de la prueba	Medir el uso de procesamiento, espacio de disco y consumo de memoria de memoria del Servidor.
Técnica	• Uso de pruebas de desarrollo para el desempeño de archivos o prueba de carga. • Prueba del limites de los recursos, pruebas que pueden ser corridas en una sola maquina (la RAM puede ser limitado). • Permanente pruebas de esfuerzo.

d. Prueba de volumen

Objetivo de la prueba	Máximo tamaño de la base de datos y consistencia en los datos registrados.
Técnica :	Se realiza un monitoreo por un periodo largo de tiempo (mínimo una hora), verifica el tamaño de la base de datos y la integridad de la data almacenada en ella mediante la visualización de los reportes y / o haciendo consultas manualmente a la base de datos.

Criterio complementario	Todos los planes de prueba deben ser ejecutados y especificando los limites del sistema alcanzado o exceso sin daños en el software.
Consideraciones especiales :	Tiempo de duración es aceptable con altos volúmenes y no por debajo del promedio.

e. Prueba de recuperación y fallos

Objetivos de prueba:	En caso de interrupción por parte del cliente. En caso de Interrupción por parte del Servidor. En caso de Interrupción de comunicación en la red hacia el Servidor.
Técnica:	Realizando cortes imprevistos del sistema ya sea por falla de equipos, eléctrica o no respuesta entre el cliente y servidor.

5.4 Especificación de requerimientos de software

5.4.1 Requerimientos funcionales

- Sistema operativo Windows 2000 / XP / 2000 Server / 2003 Server.
- Base de datos MS Access 2003.
- La aplicación para el monitoreo de red en tiempo real será en Visual Basic 6.0.
- La aplicación para el análisis, registro y carga de eventos será en Visual Basic .NET 2003.

5.4.2 Requerimientos no funcionales

- **Funcionalidad**

  El sistema puede ser reutilizado para agregar nuevas funcionales al sistema.

- **Usabilidad**

  El sistema tiene la capacidad de ser usado y entendido fácilmente por el usuario, para las funciones especificas que este realiza.

- **Fiabilidad**

  El sistema tiene la capacidad de mantener un nivel de funcionamiento cuando sea requerido. Se hicieron las pruebas anteriormente mencionadas.

- **Performance**

  El sistema no consume muchos recursos y su impacto en el sistema que monitorea es mínimo, su repositorio de base de datos no consume mucha data debido a que solo registra información específica (el tamaño máximo de una BD MS Access es 2GB).

- **Soportabilidad**

  El sistema tiene la capacidad de trabajar en bajo cualquier ambiente de topología de red, organizacional y hardware. Manteniendo sus características.

  - **De facilidad de instalación y adaptación**, ya que trabaja bajo la familia de sistemas operativos Windows 2000 / XP / 2003. Y su aplicativo instalador, que como prerrequisito requiere la instalación del .NET Framework 1.1 y MCAD 2.8, así como la instalación de office Outlook como servicios disparador de los correos de alertas en tiempo real. El instalador completará el resto de la instalación fácilmente sin interacción del usuario.
  - **El requisito de Hardware**, Pentium IV 1.8 MHz, 512MB Disco de 40GB.

ANEXO 02

INTERFACES MÓDULO DE VISOR DE CONTROL DE SUCESOS

MÓDULO DE CONTROL DEL VISOR DE SUCESOS

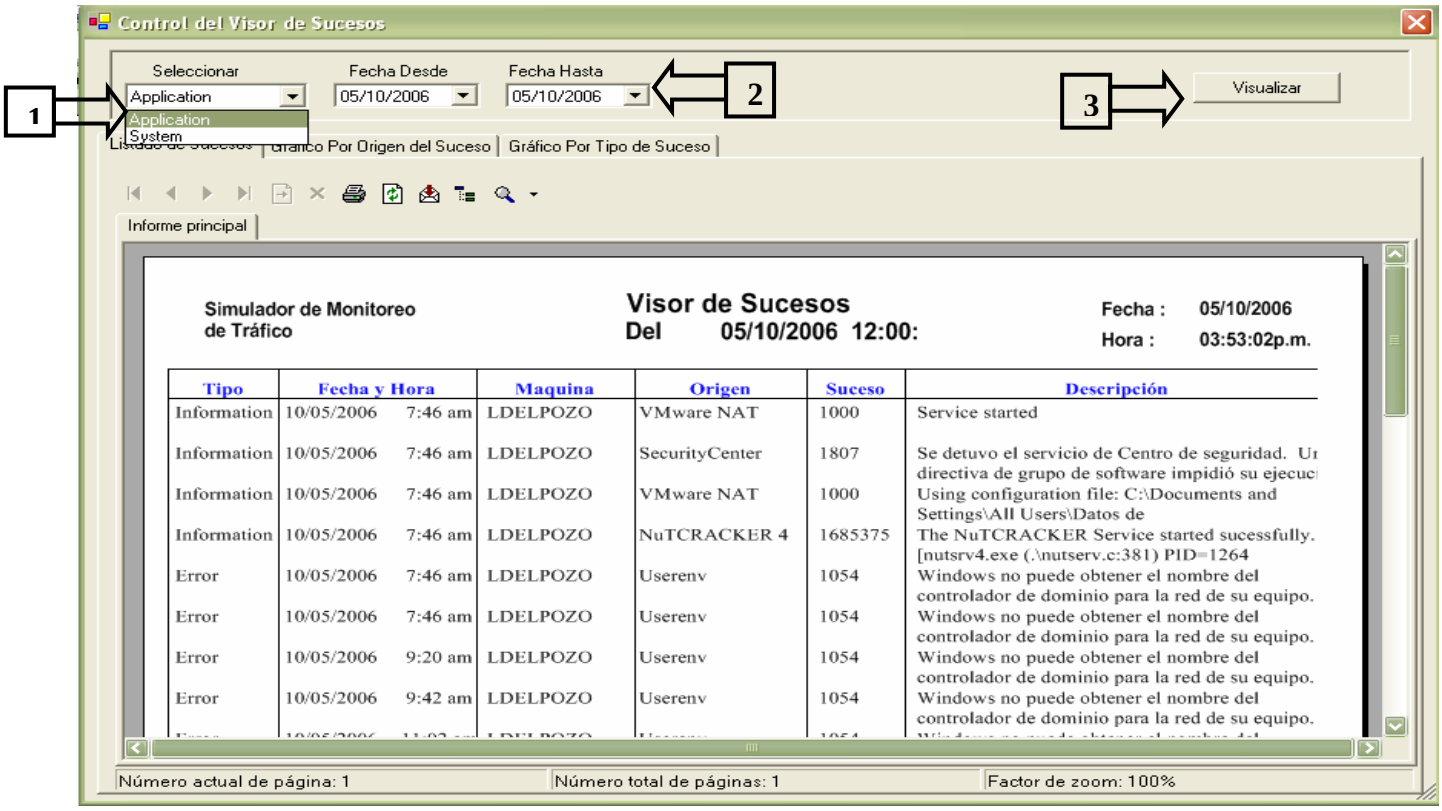


Figura Nº 25: Proceso de carga de datos del visor de sucesos.

Leyenda:

1. Se selecciona el tipo de eventos que se desea carga en el sistema: Aplicación o sistema,

2. Selección de la fecha o rangos de fecha que se desea visualizar al realizar la carga del tipo de evento seleccionado previamente,

3. Una ves realizados los dos pasos anteriores se da clic en el botón visualizar para realizar la carga automática de los eventos sucedidos en el servidor.

4. Proceso de carga de datos realizado.

El siguiente paso para la visualización de los reportes se explica a continuación.

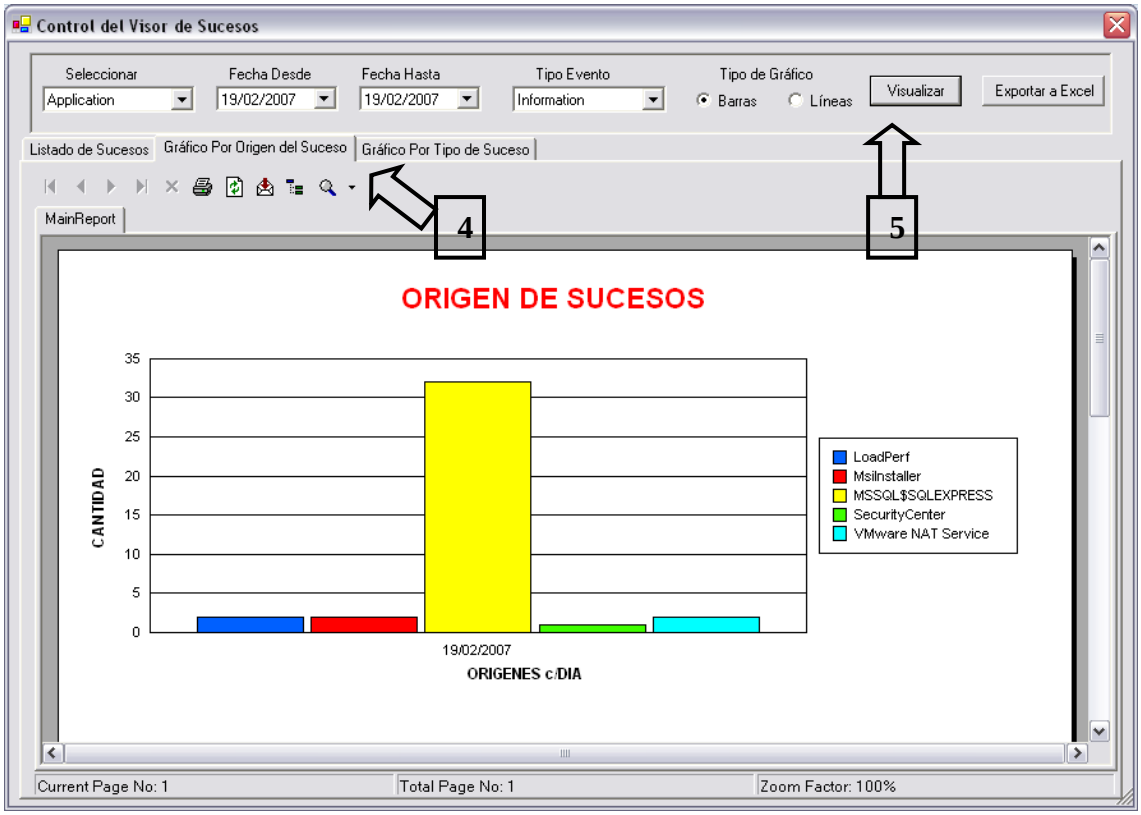


Figura Nº 26: Reporte por origen del suceso.

Leyenda:

4. Los reportes se pueden visualizar rápidamente cambiando de la pestaña listado de sucesos a la Grafico por Origen del Suceso,

5. Si se desea se puede seleccionar el tipo de grafico del reporte ya sea por barras o por líneas, el sistema automáticamente cambia el gráfico por reportes al seleccionado (si se tiene demasiados eventos registrados, para una mejor visualización se puede exportar dicho reporte a un Excel y ser personalizado).

El proceso de realización del reporte por tipo de suceso es realizado de la misma forma que en el explicado anteriormente, pasos 6 y 7.

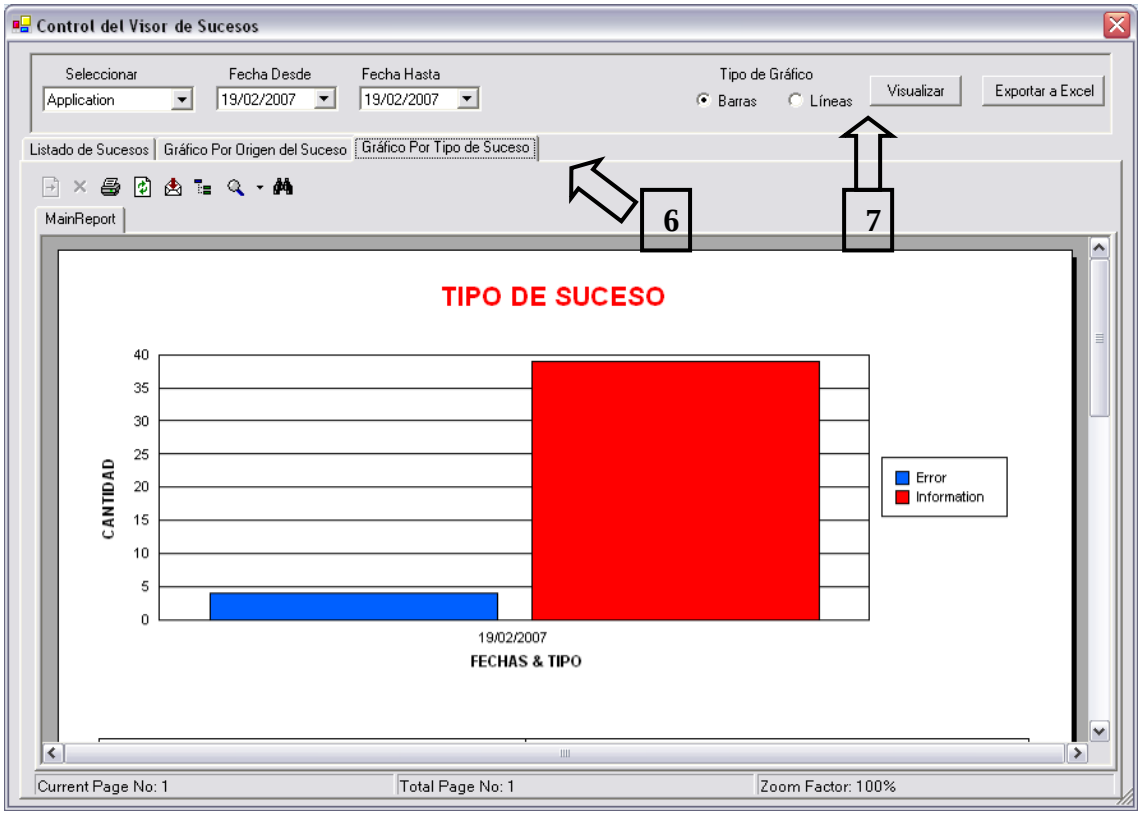


Figura Nº 27: Reporte por Tipo de sucesos

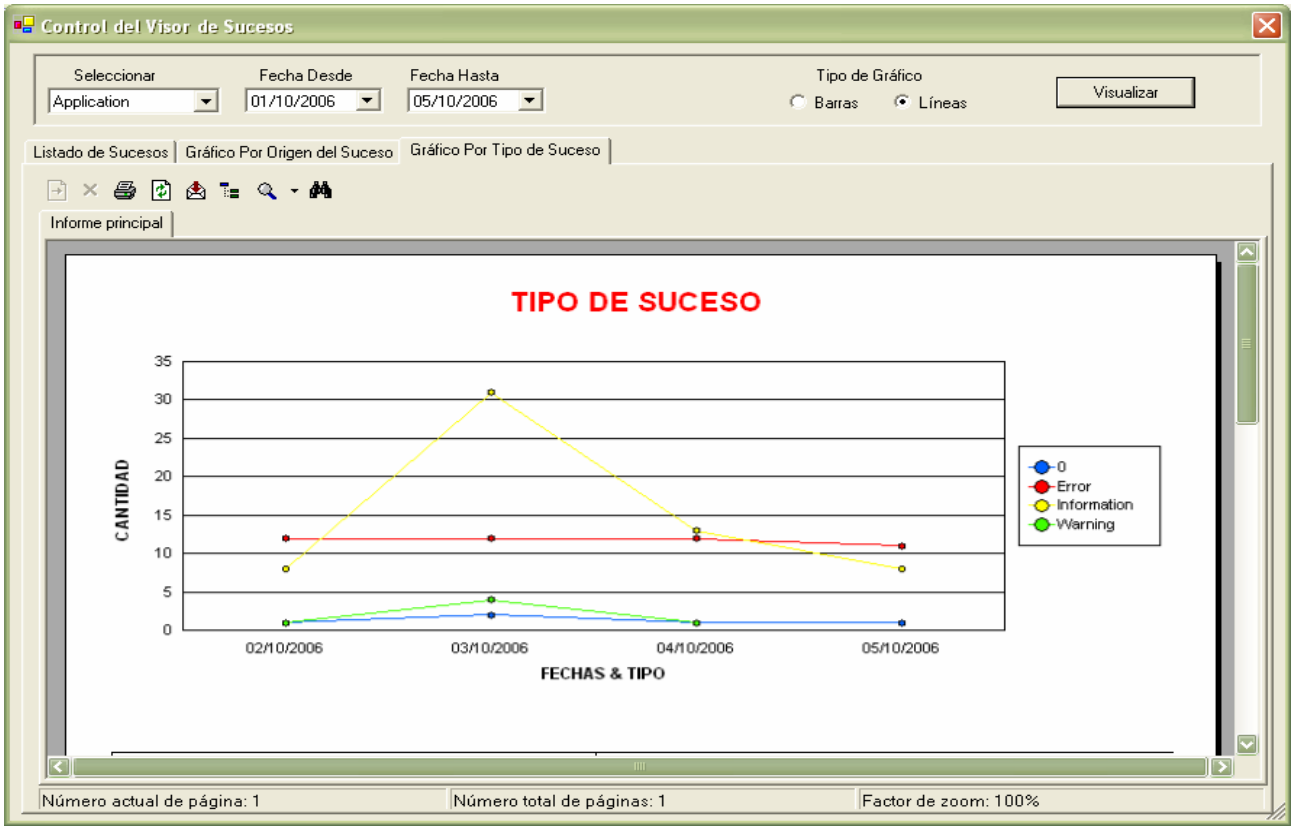


Figura Nº 28: Reporte por Tipo de sucesos, tipo de grafico lineal.

ANEXO 03

INTERFACES MÓDULO ANALIZADOR DE MONITOREO EN TIEMPO REAL

Módulo de Analizador de monitoreo de tráfico en tiempo real

Figura N° 29: Parâmetros de monitoreo de tráfico.

Leyenda:

1. Se selecciona los parámetro de duración del monitoreo,
2. Se selecciona los puntos o intervalos de medición,
3. Se seleccionan la cantidad de nodos participantes que se visualizaran en los reportes de monitoreo predefinidos por el sistema.

El proceso de monitoreo de tráfico en tiempo real es iniciado, y visualizado como se muestra a la siguiente figura.

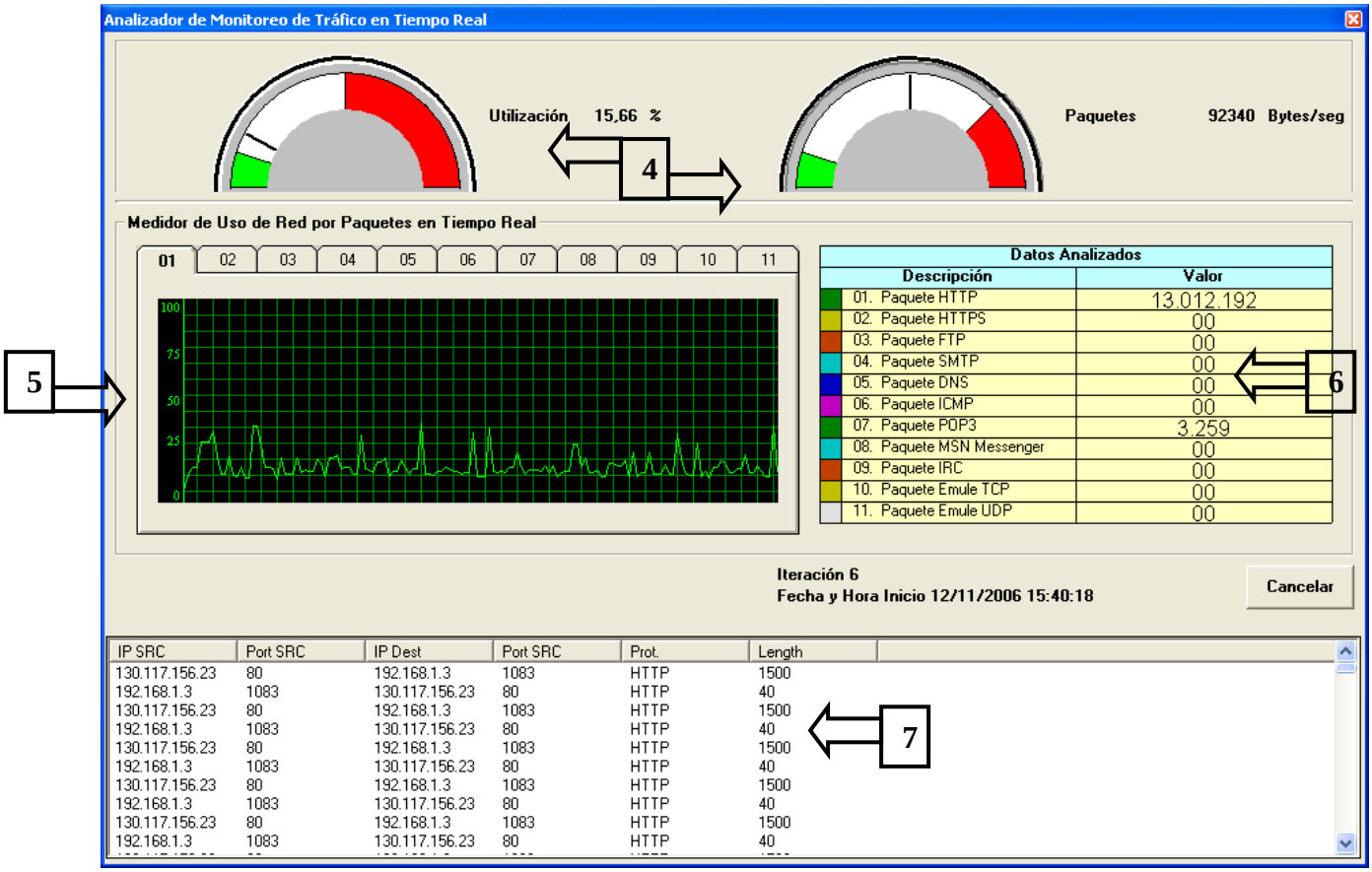


Figura N° 30: Interfase de monitoreo de tráfico.

Leyenda:

La interfaz se divide en cuatro divisiones de visualización:

4. Dos Tacómetros de monitoreo, uno que muestra el porcentaje de tráfico total que pasa en el momento por la tarjeta de red; y la cantidad de paquetes en bytes / segundo.

5. El medidor de vida de uso de red en paquetes en tiempo real por cada protocolo.

6. La sección de datos por protocolo analizados y su valor tráfico que esta registrando.

7. La sección de tráfico general que muestra la IP fuente, destino, el protocolo de comunicación y puerto, y la longitud del paquete.

El proceso de monitoreo continua hasta su culminación. El siguiente paso es la visualización de los reportes a continuación.

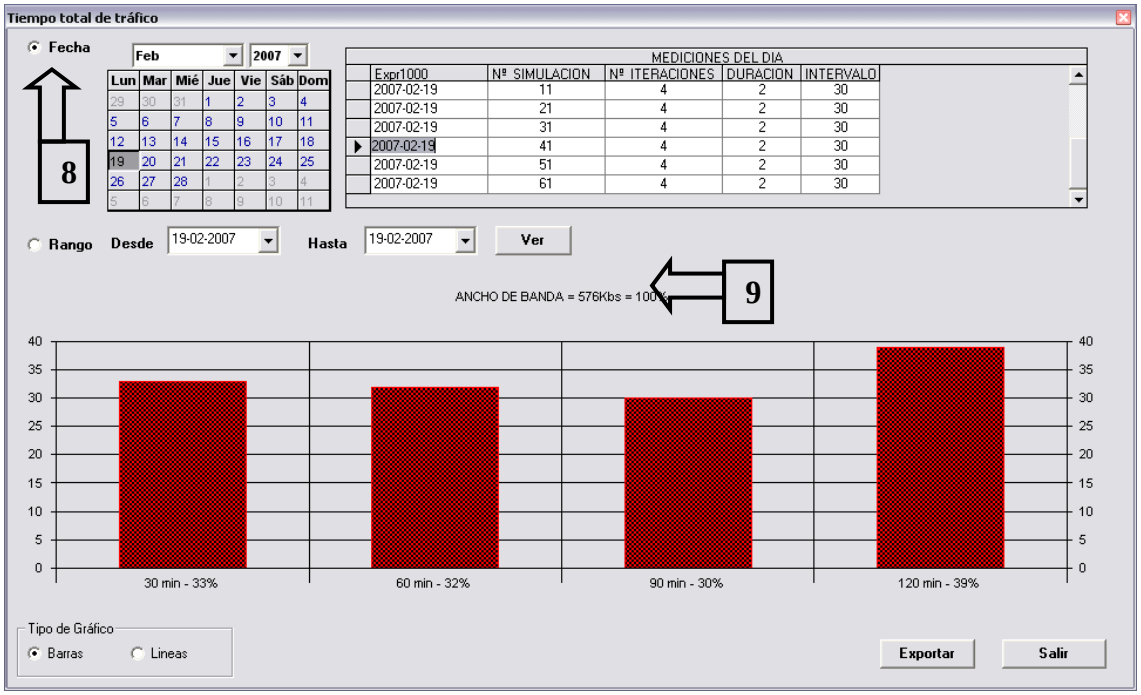


Figura N° 31: Reporte de tiempo total de tráfico.

Leyenda:

El reporte de tiempo total de tráfico por IP puede ser mostrado de dos formas:

8. Fecha del reporte, donde se ve la cantidad de monitoreos que pueden haber tenido a lugar en un día en particular, se pueden haber realizado uno o más monitoreos en un mismo día.

9. Rango de fechas, este reporte permite mostrar un análisis de tendencias de consumo de tráfico total de red. Ambos tipos de reporte pueden mostrarse en barras o líneas.

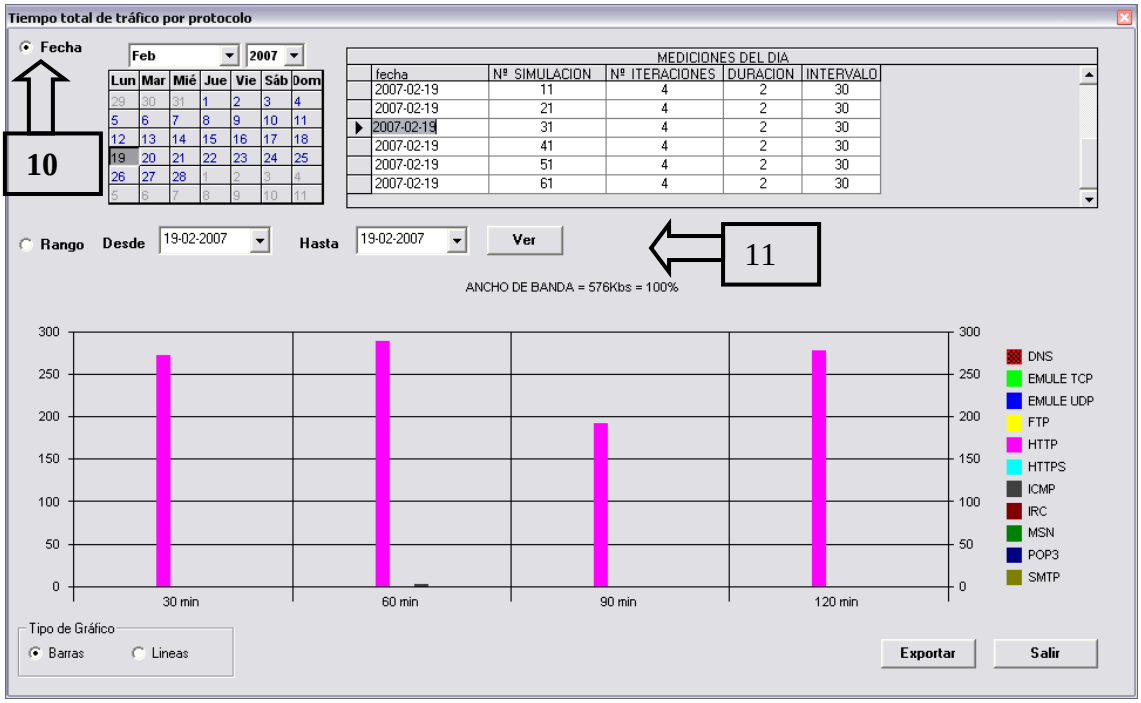


Figura N° 32: Reporte de tiempo total de tráfico por protocolo.

Leyenda:

El reporte de tiempo total de tráfico por protocolo puede ser mostrado de dos formas:

10. Fecha del reporte, muestra en consumo total de tráfico de red por protocolo, se pueden haber realizado uno o más monitoreos en un mismo día.

11. Rango de fechas, aquí se pueden ver tendencias consumo de tráfico por protocolo. Ambos tipos de reporte pueden mostrarse en barras o líneas.

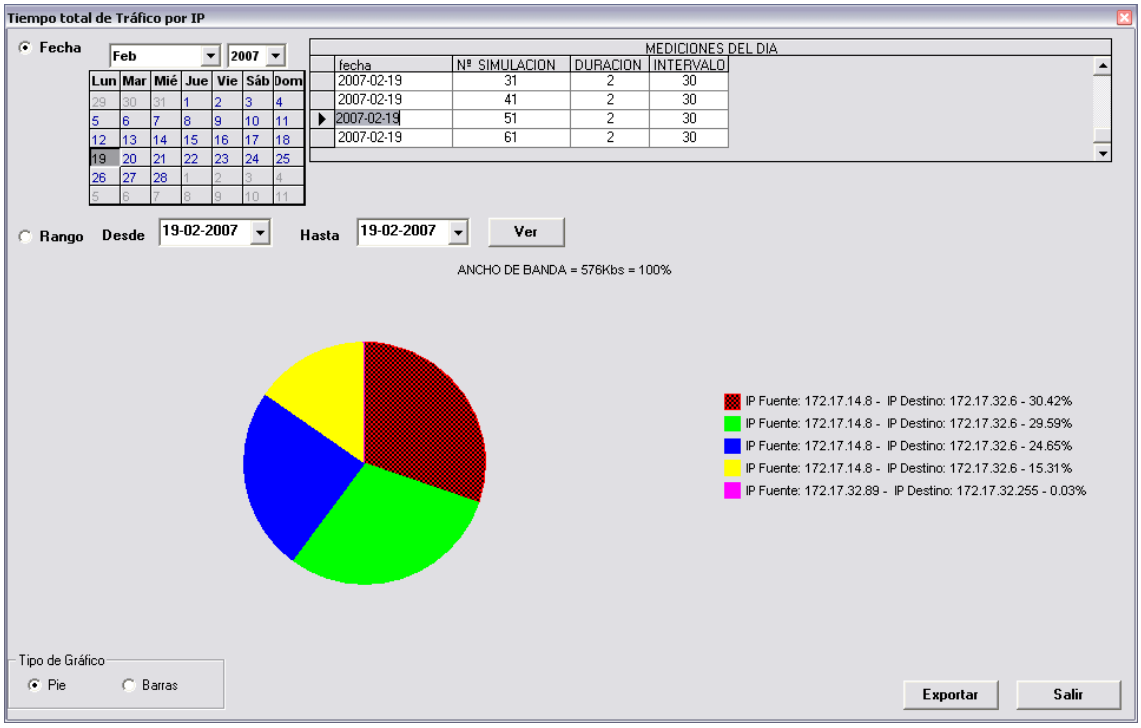


Figura N° 33: Reporte de tiempo total de tráfico por IP.

Leyenda:

El reporte de tiempo total de tráfico por IP puede ser mostrado de dos formas:

12. Fecha del reporte, muestra en consumo total de tráfico de red por IP fuente y destino, se pueden haber realizado uno o más monitoreos en un mismo día.

13. Rango de fechas, aquí se pueden ver tendencias consumo de tráfico por IP fuente y destino. Ambos tipos de reporte pueden mostrarse en barras o líneas.

ANEXO 04

CASOS DE PRUEBAS DEL SISTEMA

PRUEBAS DEL SISTEMA

1. Introducción

Documentar las pruebas realizadas al sistema analizador de monitoreo de tráfico en tiempo real, a fin de garantizar su buen desempeño para su uso como instrumento de apoyo a las funciones del área de de soporte técnico y administración de red.

2. Alcances

Este manual esta dirigido al personal del área de soporté técnico y administración de red, cuyas actividades se encuentren vinculadas al mejor funcionamiento y disponibilidad de los recursos y sistemas de red.

3. Módulo de Control de visor de sucesos de sistema o aplicación

Se verifican si los datos presentados en el sistema son correctos.

Carga de logs sin modificaciones ni incidencias

Se realizó la carga de los logs del sistema y / o aplicación sin ninguna incidencia y de manera automática. Como se muestra en la figura a continuación.

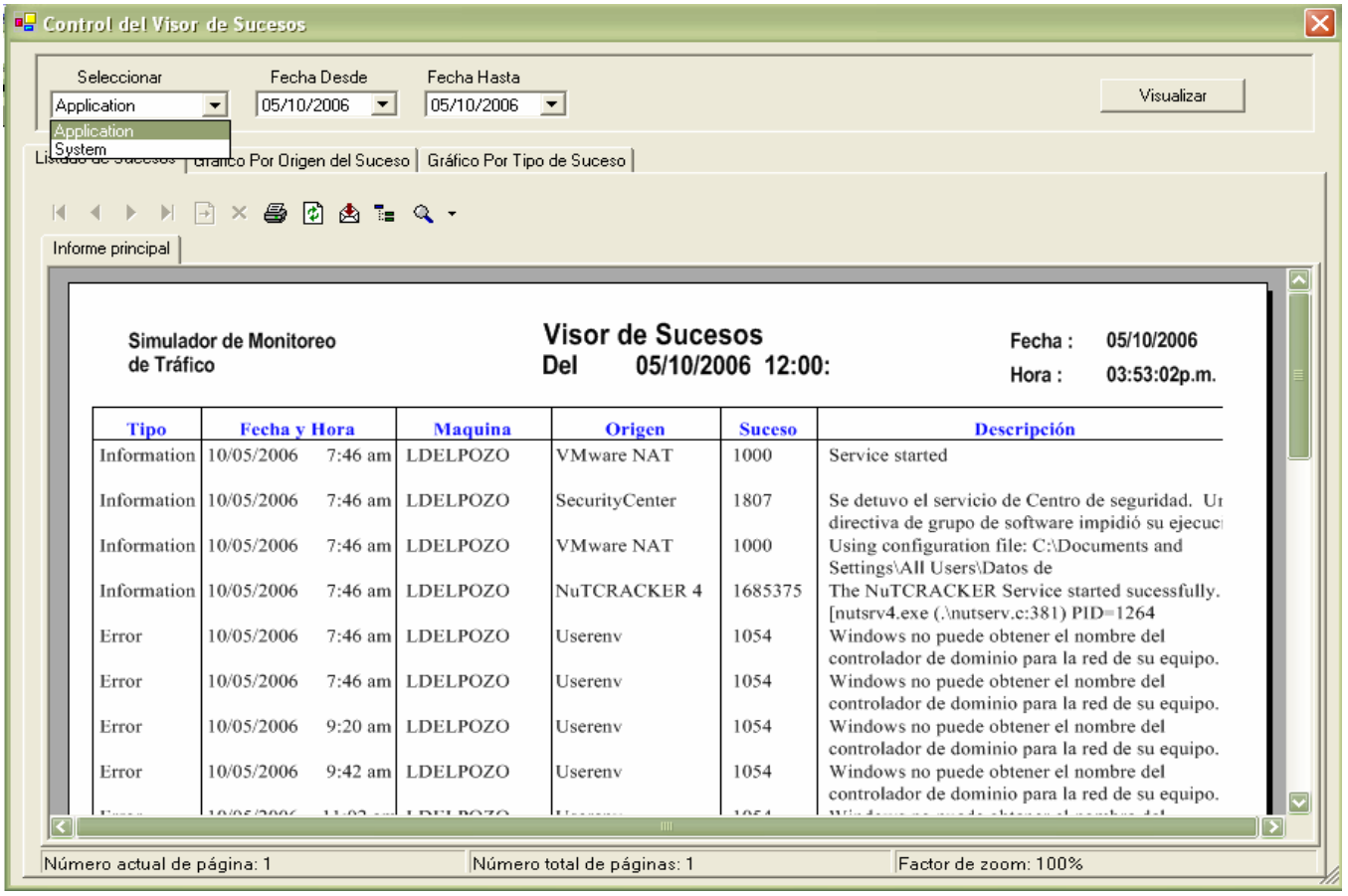


Figura N° 34: Carga de Log Sin incidencias.

4. Reportes de eventos sistema o aplicación

4.1 Reporte de eventos del sistema

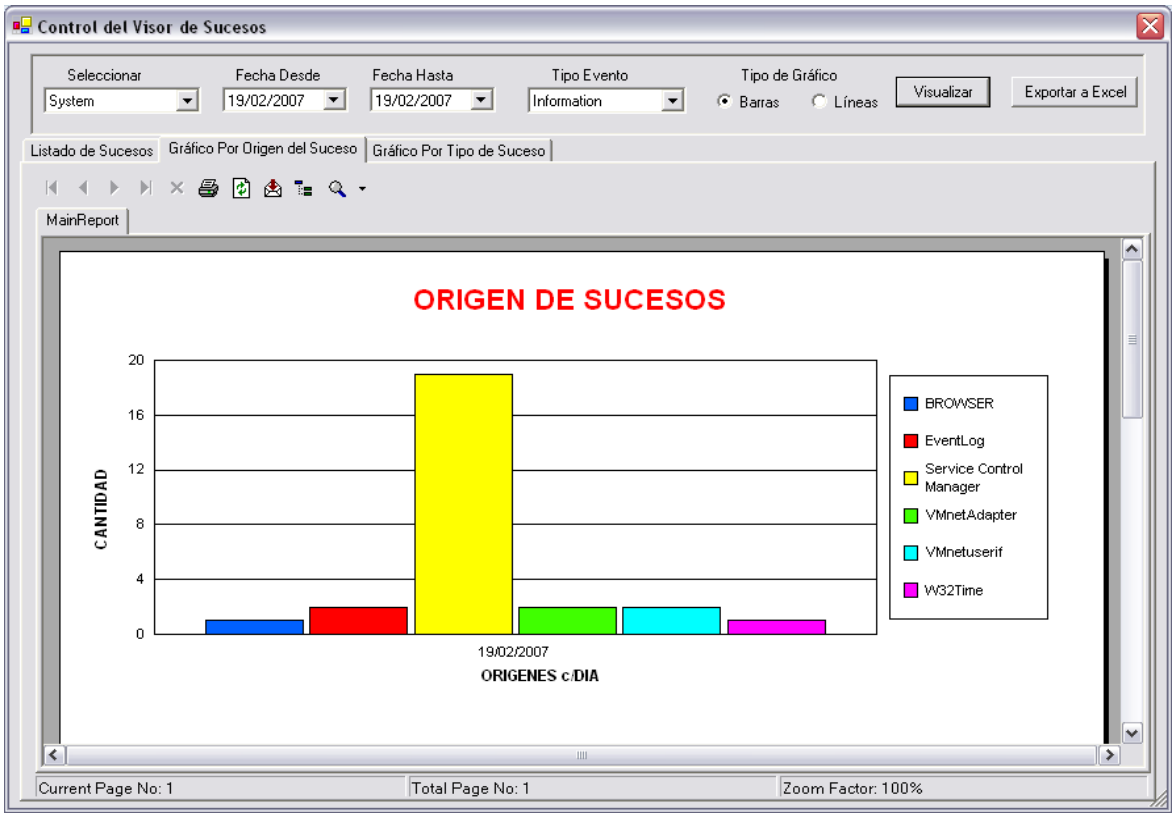


Figura N° 35: Carga de reporte sin incidencias.

4.2 Reporte de eventos de aplicación

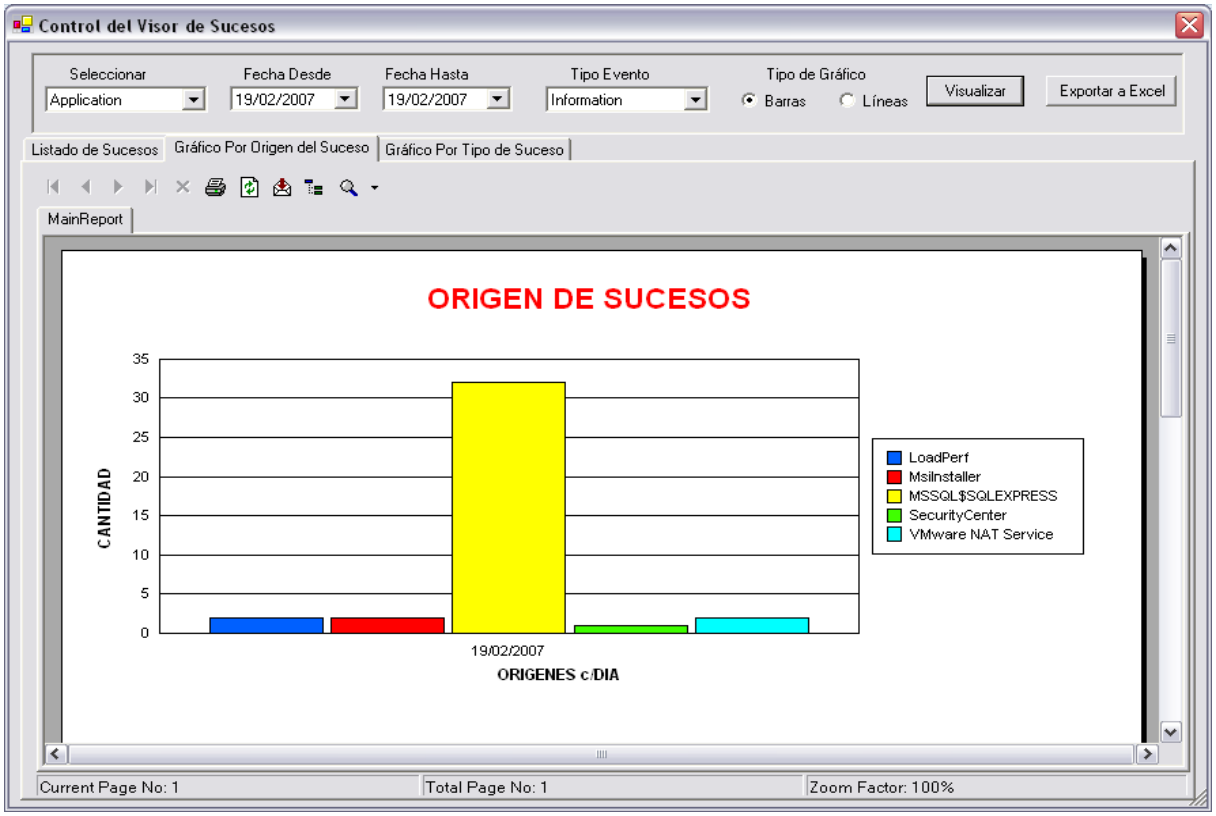


Figura N° 36: Carga de reporte sin incidencias.

5. Analizador de monitoreo y análisis de tráfico de red

Se verifica que los parámetros de monitoreo son ejecutados correctamente de inicio a fin; así como los datos mostrados por los reportes.

5.1 Selección de parametros de monitoreo

Se permite seleccionar solo los parámetros predefinidos.

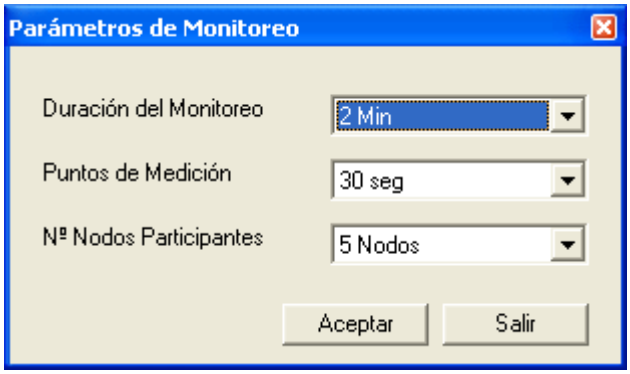


Figura N° 37: Selección de parámetros permitidos por el sistema.

5.2 Analizador de monitoreo de tráfico en tiempo real

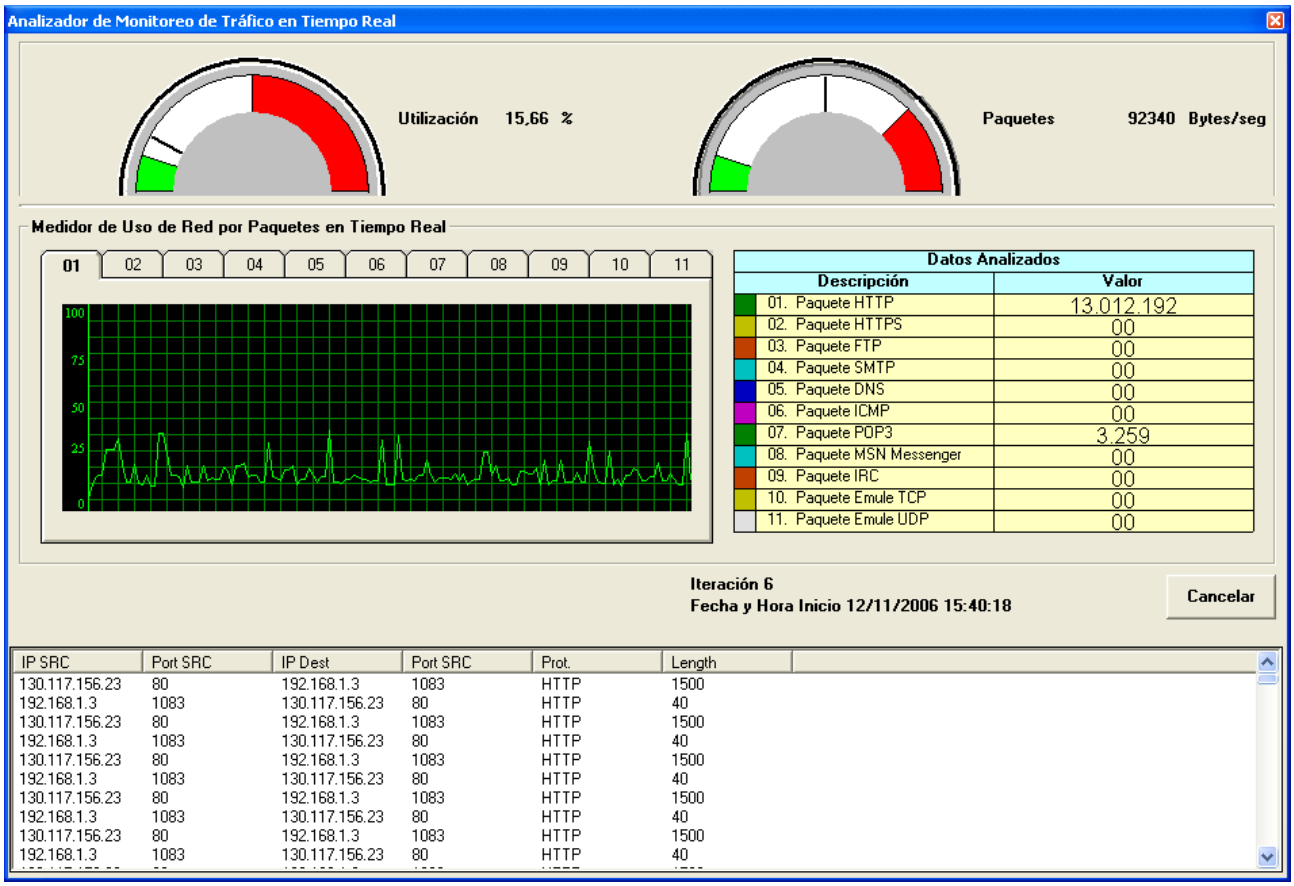


Figura N° 41: Monitoreo de tráfico de inicio a fin sin incidencias.

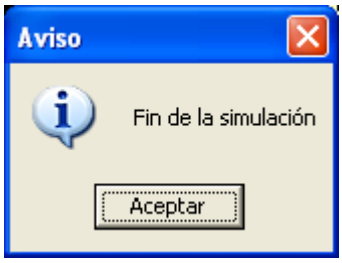


Figura N° 38: finalización del monitoreo sin incidencia.

6. Reportes de monitoreo y análisis de tráfico de red

6.1 Reporte de tráfico total por porcentaje de tiempo

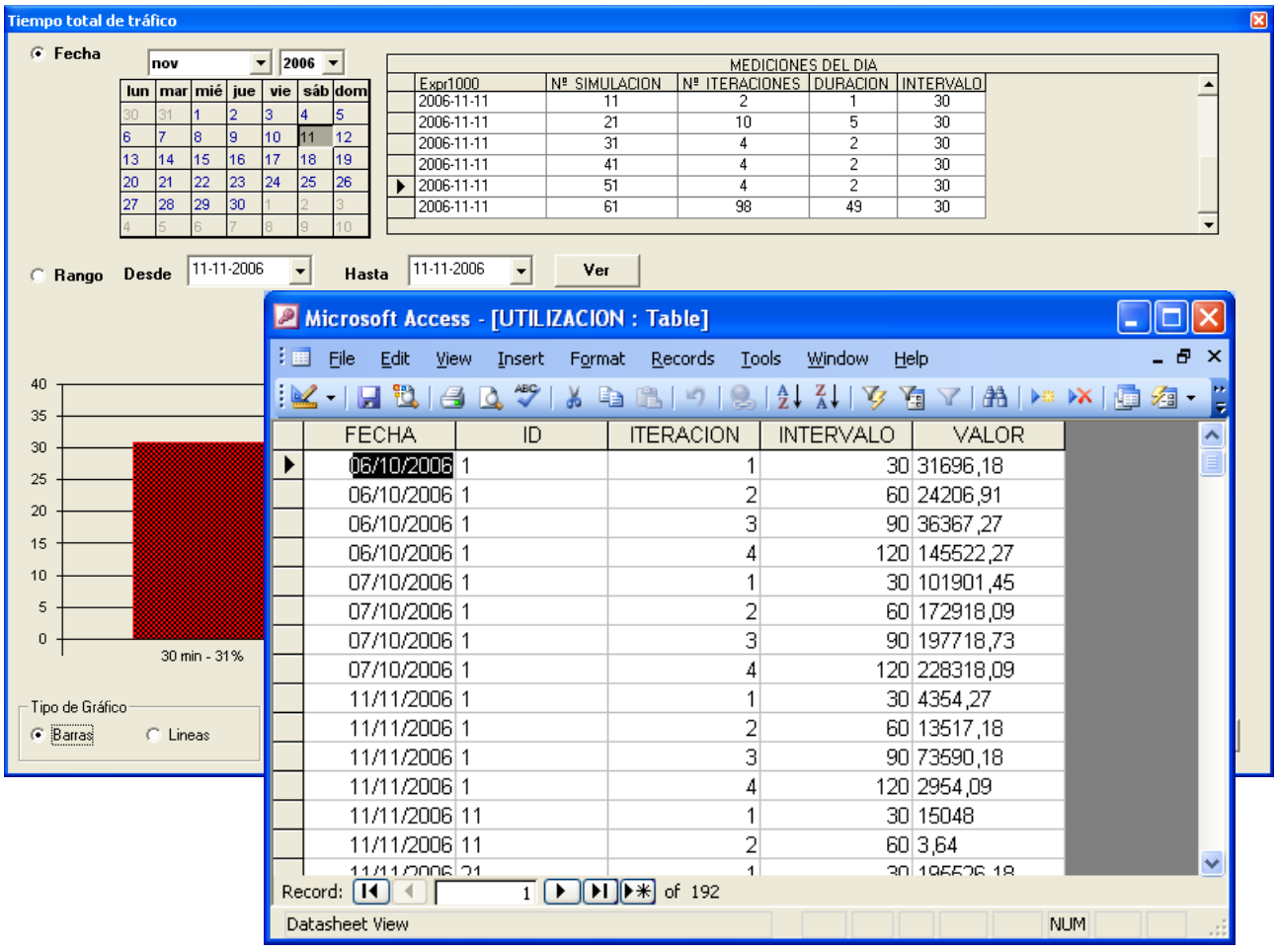


Figura N° 39: Consistencia en los datos emitidos por reporte con respecto a la data contenida en la base de datos.

5.2 Reporte de tráfico total por protocolo

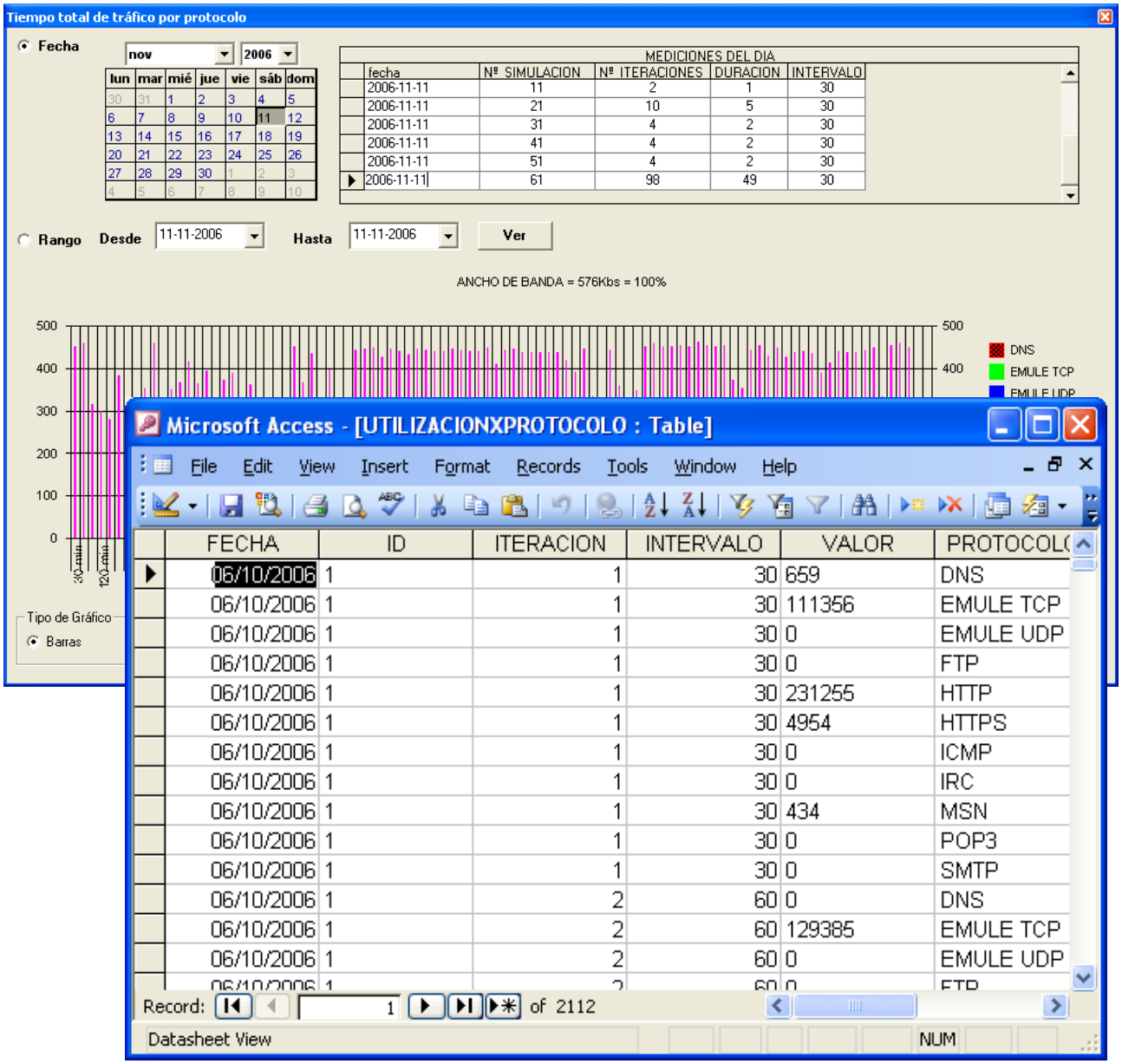


Figura N° 40: Consistencia en los datos emitidos por reporte con respecto a la data contenida en la base de datos.

5.3 Reporte de tráfico total en paquetes por IP

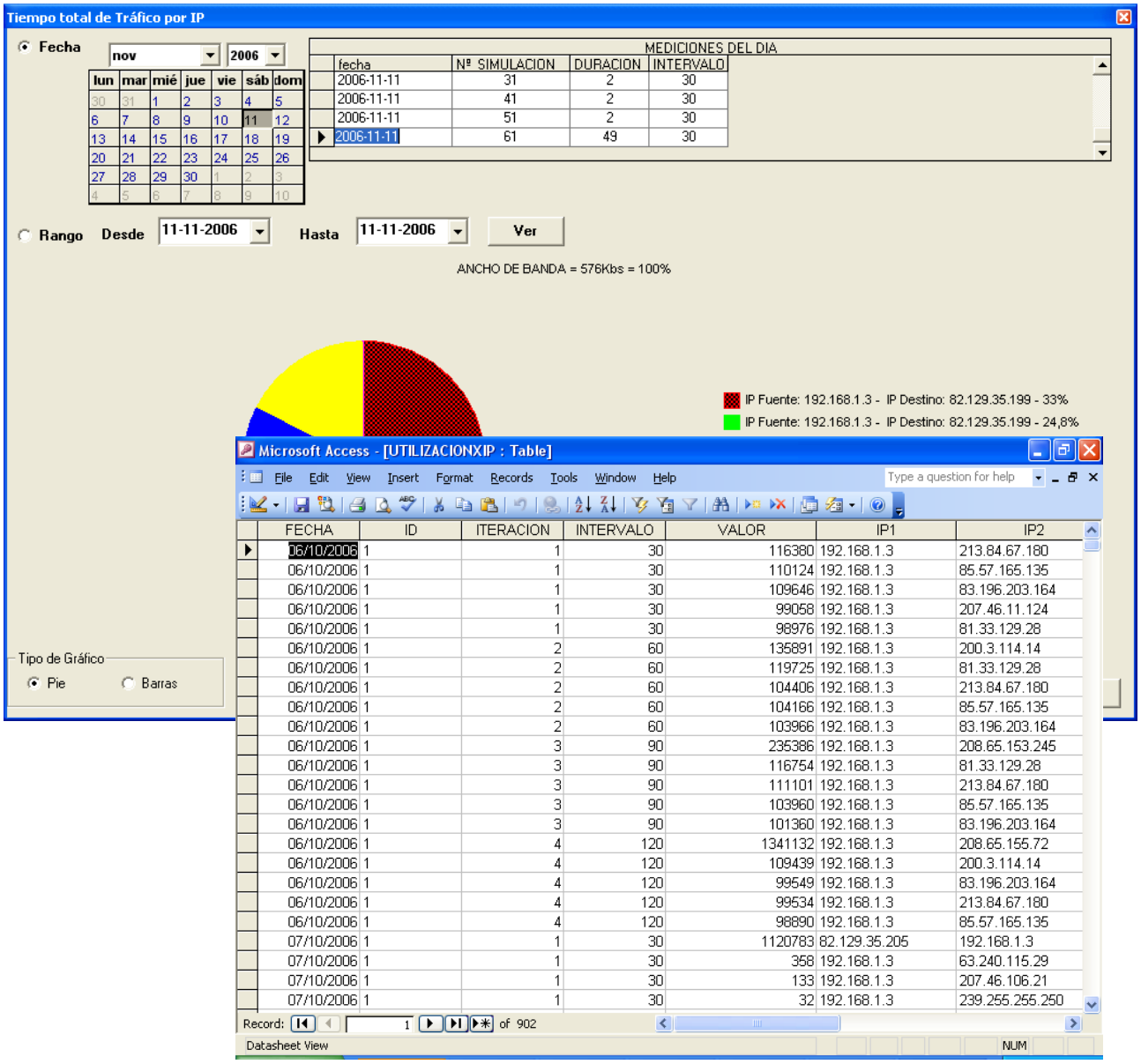


Figura N° 41: Consistencia en los datos emitidos por reporte con respecto a la data contenida en la base de datos.

ANEXO 05

COSTO DE DESARROLLO DEL SISTEMA

ANEXO 06

CÓDIGO DE LAS INTERFACES DEL SISTEMA

Código del Servicio WinSock

A continuación se muestra el código para el registro del monitoreo de red, que se encarga del monitoreo de trafico de red en tiempo real y que puede ser reutilizado para ampliar su campo de monitoreo a mas protocolos y puede ser usado para otros tipos de implementaciones de monitoreo de red.

```
Option Explicit
Public cnt As Long
Public Const WM_USER = &H400
Public Const WINSOCKMSG = WM_USER + 1
Public Const SIO_RCVALL = &H98000001
Public Const SO_RCVTIMEO = &H1006
Public Const AF_INET = 2
Public Const INVALID_SOCKET = -1
Public Const SOCKET_ERROR = -1
Public Const FD_READ = &H1&
Public Const FD_WRITE = &H2&
Public Const FD_CONNECT = &H10&
Public Const FD_CLOSE = &H20&
Public Const PF_INET = 2
Public Const SOCK_STREAM = 1
Public Const SOCK_RAW = 3
Public Const IPPROTO_TCP = 6
Public Const IPPROTO_IP = 0
Public Const GWL_WNDPROC = (-4)
Public Const WSA_DESCRIPTIONLEN = 256
Public Const WSA_DescriptionSize = WSA_DESCRIPTIONLEN + 1
Public Const WSA_SYS_STATUS_LEN = 128
Public Const WSA_SysStatusSize = WSA_SYS_STATUS_LEN + 1
Public Const INADDR_NONE = &HFFFF
Public Const SOL_SOCKET = &HFFFF&
Public Const SO_LINGER = &H80&
Public Const hostent_size = 16
Public Const sockaddr_size = 16
Type WSADataType
   wVersion As Integer
   wHighVersion As Integer
   szDescription As String * WSA_DescriptionSize
   szSystemStatus As String * WSA_SysStatusSize
   iMaxSockets As Integer
   iMaxUdpDg As Integer
   lpVendorInfo As Long
```

```
End Type
Type HostEnt
   h_name As Long
   h_aliases As Long
   h_addrtype As Integer
   h_length As Integer
   h_addr_list As Long
End Type
Type sockaddr
   sin_family As Integer
   sin_port As Integer
   sin_addr As Long
   sin_zero As String * 8
End Type
Type LingerType
   l_onoff As Integer
   l_linger As Integer
End Type

Type ipheader
   ip_verlen As Byte
   ip_tos As Byte
   ip_totallength As Integer
   ip_id As Integer
   ip_offset As Integer
   ip_ttl As Byte
   ip_protocol As Byte
   ip_checksum As Integer
   ip_srcaddr As Long
   ip_destaddr As Long
End Type

Type tcpheader
   src_portno As Integer
   dst_portno As Integer
   Sequenceno As Long
   Acknowledgeno As Long
   DataOffset As Byte
   flag As Byte
   Windows As Integer
   checksum As Integer
   UrgentPointer As Integer
End Type


Type udpheader
   src_portno As Integer
   dst_portno As Integer
   udp_length As Integer
   udp_checksum As Integer
```

```
End Type

Private Const SIO_GET_INTERFACE_LIST = &H4004747F

  Type sockaddr_gen
    AddressIn As sockaddr
    filler(0 To 7) As Byte
  End Type

  Type INTERFACE_INFO
iiFlags As Long     ' Interfaces flags
iiAddress As sockaddr_gen     ' Direccion de interfaces
iiBroadcastAddress As sockaddr_gen     ' Direcciones broadcast
iiNetmask As sockaddr_gen     ' Mascara de red
  End Type


  Type aINTERFACE_INFO
   interfaceinfo(0 To 7) As INTERFACE_INFO
  End Type

Public Declare Function bind Lib "wsock32.dll" (ByVal s As Integer, addr
As sockaddr, ByVal namelen As Integer) As Integer
Public Declare Function setsockopt Lib "wsock32.dll" (ByVal s As Long,
ByVal Level As Long, ByVal optname As Long, optval As Any, ByVal
optlen As Long) As Long
Public Declare Function getsockopt Lib "wsock32.dll" (ByVal s As Long,
ByVal Level As Long, ByVal optname As Long, optval As Any, optlen As
Long) As Long
Public Declare Function WSAGetLastError Lib "wsock32.dll" () As Long
Public Declare Function WSAIsBlocking Lib "wsock32.dll" () As Long
Public Declare Function WSACleanup Lib "wsock32.dll" () As Long
Public Declare Function Send Lib "wsock32.dll" Alias "send" (ByVal s As
Long, buf As Any, ByVal buflen As Long, ByVal flags As Long) As Long
Public Declare Function recv Lib "wsock32.dll" (ByVal s As Long, buf As
Any, ByVal buflen As Long, ByVal flags As Long) As Long
Public Declare Function WSAStartup Lib "wsock32.dll" (ByVal wVR As
Long, lpWSAD As WSADataType) As Long
Public Declare Function htons Lib "wsock32.dll" (ByVal hostshort As
Long) As Integer
Public Declare Function ntohs Lib "wsock32.dll" (ByVal netshort As Long)
As Integer
Public Declare Function socket Lib "wsock32.dll" (ByVal af As Long,
ByVal s_type As Long, ByVal protocol As Long) As Long
Public Declare Function closesocket Lib "wsock32.dll" (ByVal s As Long)
As Long
Public Declare Function Connect Lib "wsock32.dll" Alias "connect"
(ByVal s As Long, addr As sockaddr, ByVal namelen As Long) As Long
```

```
Public Declare Function WSAAsyncSelect Lib "wsock32.dll" (ByVal s As
Long, ByVal hWnd As Long, ByVal wMsg As Long, ByVal lEvent As
Long) As Long
Public Declare Function inet_addr Lib "wsock32.dll" (ByVal cp As String)
As Long
Public Declare Function gethostbyname Lib "wsock32.dll" (ByVal
host_name As String) As Long
Public Declare Function inet_ntoa Lib "wsock32.dll" (ByVal inn As Long)
As Long
Public Declare Function WSACancelBlockingCall Lib "wsock32.dll" () As
Long
Public Declare Function WSAIoctl Lib "ws2_32.dll" (ByVal s As Long, _
ByVal dwIoControlCode As Long, _
lpvInBuffer As Any, _
ByVal cbInBuffer As Long, _
lpvOutBuffer As Any, _
ByVal cbOutBuffer As Long, _
lpcbBytesReturned As Long, _
lpOverlapped As Long, _
lpCompletionRoutine As Long) As Long

Declare Function SetWindowLong Lib "user32" Alias "SetWindowLongA"
(ByVal hWnd As Long, ByVal nIndex As Long, ByVal dwNewLong As
Long) As Long
Declare Function CallWindowProc Lib "user32" Alias "CallWindowProcA"
(ByVal lpPrevWndFunc As Long, ByVal hWnd As Long, ByVal Msg As
Long, ByVal wParam As Long, ByVal lParam As Long) As Long
Public Declare Function lstrlen Lib "kernel32" Alias "lstrlenA" (ByVal
lpString As Any) As Long
Public Declare Sub MemCopy Lib "kernel32" Alias "RtlMoveMemory"
(Dest As Any, Src As Any, ByVal cb&)


Public saZero As sockaddr
Public WSAStartedUp As Boolean, Obj As TextBox
Public PrevProc As Long, lSocket As Long

'Protocolos
'
'Enum SockProtocols
'  IPPROTO_IP = 0                 'dummy for IP
'  IPPROTO_ICMP = 1                  'control message protocol
'  IPPROTO_IPIP = 4
'  IPPROTO_GGP = 2                   ' gateway^2 (deprecated)
'  IPPROTO_TCP = 6                  ' tcp
'  IPPROTO_EGP = 8
'  IPPROTO_PUP = 12                   ' pup
'  IPPROTO_UDP = 17                   ' user datagram protocol
'  IPPROTO_IDP = 22                  ' xns idp
'  IPPROTO_ND = 77                   ' UNOFFICIAL net disk proto
```

```
'  NSPROTO_IPX = 1000
'  NSPROTO_SPX = 1256
'  NSPROTO_SPXII = 1257
'End Enum
'
'
'subclassing function
Public Sub HookForm(F As Form)
   PrevProc = SetWindowLong(F.hWnd, GWL_WNDPROC, AddressOf
WindowProc)
End Sub
Public Sub UnHookForm(F As Form)
   If PrevProc <> 0 Then
      SetWindowLong F.hWnd, GWL_WNDPROC, PrevProc
      PrevProc = 0
   End If
End Sub
Public Function WindowProc(ByVal hWnd As Long, ByVal uMsg As
Long, ByVal wParam As Long, ByVal lParam As Long) As Long
   Debug.Print uMsg
   If uMsg = WINSOCKMSG Then
      ProcessMessage wParam, lParam
   Else
   'If cGetInputState() <> 0 Then

      WindowProc = CallWindowProc(PrevProc, hWnd, uMsg, wParam,
lParam)

   'End If
   End If
End Function
Sub display1(readbuffer() As Byte)
'texbox -> scrollbars=3 & multiline=true
On Error GoTo errhand
Dim ip_header As ipheader
Dim tcp_header As tcpheader
Dim udp_header As udpheader
Dim lvitem
CopyMemory_any ip_header, readbuffer(0), Len(ip_header)
                cnt = cnt + 1
                 Obj.Parent.Label2 = cnt

                 Obj.Text = Obj.Text & ntohs(ip_header.ip_totallength) & "
bytes" & vbCrLf
                 Obj.Text = Obj.Text & getascip(ip_header.ip_srcaddr) & "->"
& getascip(ip_header.ip_destaddr) & vbCrLf

                 'icmp
                 If ip_header.ip_protocol = 1 Then
                    Obj.Text = Obj.Text & "ICMP" & vbCrLf
```

```
                End If
                'tcp
                If ip_header.ip_protocol = 6 Then
                    Obj.Text = Obj.Text & "TCP" & vbCrLf
                    CopyMemory_any tcp_header, readbuffer(0 + 20),
Len(tcp_header)
                    Obj.Text = Obj.Text & "(src port) " &
ntohs(tcp_header.src_portno) & vbCrLf
                    Obj.Text = Obj.Text & "(dst port) " &
ntohs(tcp_header.dst_portno) & vbCrLf
                End If
                'udp
                If ip_header.ip_protocol = 17 Then
                    Obj.Text = Obj.Text & "UDP" & vbCrLf
                    CopyMemory_any udp_header, readbuffer(0 + 20),
Len(udp_header)
                    Obj.Text = Obj.Text & "(src port) " &
ntohs(udp_header.src_portno) & vbCrLf
                    Obj.Text = Obj.Text & "(dst port) " &
ntohs(udp_header.dst_portno) & vbCrLf
                End If
                Obj.Text = Obj.Text & String(34, "*") & vbCrLf
Exit Sub
errhand:
MsgBox err.Description & vbCrLf & err.Number & vbCrLf &
err.LastDllError
AddLog err.Description & vbCrLf & err.Number & vbCrLf &
err.LastDllError
End Sub

'our Winsock-message handler
Public Sub ProcessMessage(ByVal lFromSocket As Long, ByVal lParam
As Long)
    On Error GoTo errhand
    Dim X As Long, strCommand As String
    Dim readbuffer(0 To 1499) As Byte
    Dim ip_header As ipheader

    Select Case lParam
        Case FD_CONNECT 'estamos conectados
            Debug.Print "FD_CONNECT"
        Case FD_WRITE 'escritura sobre nuestra conexion
            Debug.Print "FD_WRITE"
        Case FD_READ 'lectura sobre nuestra conexion
            'inicio de lectura de data
            'Debug.Print "FD_READ"

            Do
                X = recv(lFromSocket, readbuffer(0), 1500, 0)
                If X > 0 Then
```

```
                frmSimulador.display2 readbuffer()
                End If
                If X <> 1500 Then Exit Do


            Loop
        Case FD_CLOSE 'conexion cerrada
            Debug.Print "FD_CLOSE"
    End Select
    Exit Sub
errhand:
    MsgBox err.Description & vbCrLf & err.Number & vbCrLf &
err.LastDllError
    AddLog err.Description & vbCrLf & err.Number & vbCrLf &
err.LastDllError
End Sub
'Funciones standares de la funcion WinSock
'del wsksock.bas-file
Public Function StartWinsock(sDescription As String) As Boolean
    Dim StartupData As WSADataType
    If Not WSAStartedUp Then
        If Not WSAStartup(&H101, StartupData) Then
            WSAStartedUp = True
            sDescription = StartupData.szDescription
        Else
            WSAStartedUp = False
        End If
    End If
    StartWinsock = WSAStartedUp
End Function
Sub EndWinsock()
    Dim ret&
    If WSAIsBlocking() Then
        ret = WSACancelBlockingCall()
    End If
    ret = WSACleanup()
    WSAStartedUp = False
End Sub
'the nice part
Function ConnectSock(ByVal Host$, ByVal Port&, ByVal HWndToMsg&,
ByVal Async%) As Long
    Dim s&, SelectOps&, Dummy&
    Dim RCVTIMEO As Long
    Dim sockin As sockaddr
    Dim ret As Long

    sockin = saZero
    sockin.sin_family = AF_INET
    sockin.sin_port = htons(Port)
```

```
    If sockin.sin_port = INVALID_SOCKET Then
       ConnectSock = INVALID_SOCKET
       MsgBox "INVALID_SOCKET"
       Exit Function
    End If

    sockin.sin_addr = GetHostByNameAlias(Host$)

    If sockin.sin_addr = INADDR_NONE Then
       ConnectSock = INVALID_SOCKET
       MsgBox "INVALID_SOCKET"
       Exit Function
    End If


    s = socket(AF_INET, SOCK_RAW, IPPROTO_IP)
    If s < 0 Then
       ConnectSock = INVALID_SOCKET
       MsgBox "INVALID_SOCKET"
       Exit Function
    End If


RCVTIMEO = 5000
ret = setsockopt(s, SOL_SOCKET, SO_RCVTIMEO, (RCVTIMEO), 4)
If ret <> 0 Then
   MsgBox "setsockopt failed"
   If s > 0 Then Dummy = closesocket(s)
   Exit Function
End If

'revisin del si el setsockopt esta ok...
'Dim v As Long
'ret = getsockopt(s, SOL_SOCKET, &H1006, v, 4)
'Debug.Print v

ret = bind(s, sockin, Len(sockin))
If ret <> 0 Then
    If s > 0 Then Dummy = closesocket(s)
    MsgBox "bind failed"
    Exit Function
End If


Dim lngInBuffer As Long
Dim lngBytesReturned As Long
Dim lngOutBuffer As Long

lngInBuffer = 1
ret = WSAIoctl(s, SIO_RCVALL, lngInBuffer, Len(lngInBuffer), _
```

```
lngOutBuffer, Len(lngOutBuffer), lngBytesReturned, ByVal 0, ByVal 0)
If ret <> 0 Then
   If s > 0 Then Dummy = closesocket(s)
   MsgBox "WSAIoctl failed"
   Exit Function
End If

SelectOps = FD_READ 'Or FD_WRITE Or FD_CONNECT Or
FD_CLOSE
ret = WSAAsyncSelect(s, HWndToMsg, WINSOCKMSG, ByVal
SelectOps)
If ret <> 0 Then
   If s > 0 Then Dummy = closesocket(s)
   ConnectSock = INVALID_SOCKET
   MsgBox "INVALID_SOCKET"
   Exit Function
End If

ConnectSock = s
Exit Function
errhand:
MsgBox err.Description & vbCrLf & err.Number & vbCrLf &
err.LastDllError
AddLog err.Description & vbCrLf & err.Number & vbCrLf &
err.LastDllError
End Function
Function GetHostByNameAlias(ByVal hostname$) As Long
   On Error Resume Next
   Dim phe&
   Dim heDestHost As HostEnt
   Dim addrList&
   Dim retIP&
   retIP = inet_addr(hostname)
   If retIP = INADDR_NONE Then
      phe = gethostbyname(hostname)
      If phe <> 0 Then
         MemCopy heDestHost, ByVal phe, hostent_size
         MemCopy addrList, ByVal heDestHost.h_addr_list, 4
         MemCopy retIP, ByVal addrList, heDestHost.h_length
      Else
         retIP = INADDR_NONE
      End If
   End If
   GetHostByNameAlias = retIP
   If err Then GetHostByNameAlias = INADDR_NONE
End Function
Function getascip(ByVal inn As Long) As String
   On Error Resume Next
   Dim lpStr&
   Dim nStr&
```

```
    Dim retString$
    retString = String(32, 0)
    lpStr = inet_ntoa(inn)
    If lpStr = 0 Then
       getascip = "255.255.255.255"
       Exit Function
    End If
    nStr = lstrlen(lpStr)
    If nStr > 32 Then nStr = 32
    MemCopy ByVal retString, ByVal lpStr, nStr
    'retString = Left(retString, nStr)
    getascip = Mid(retString, 1, nStr)
    If err Then getascip = "255.255.255.255"
End Function



Public Function wsck_enum_interfaces(ByRef str() As String) As Long
Dim lngSocketDescriptor   As Long
Dim lngInBuffer           As Long
Dim lngBytesReturned      As Long
Dim lngWin32apiResultCode As Long
Dim mudtWSAData As WSADataType
Dim desc As String

Call StartWinsock(desc)

lngSocketDescriptor = socket(AF_INET, SOCK_STREAM, 0)

If lngSocketDescriptor Then
    If lngWin32apiResultCode Then
    wsck_enum_interfaces = err.LastDllError
    Exit Function
End If

End If

Dim buffer As aINTERFACE_INFO

lngWin32apiResultCode = _
WSAIoctl(lngSocketDescriptor, SIO_GET_INTERFACE_LIST, _
ByVal 0, ByVal 0, _
buffer, 1024, lngBytesReturned, ByVal 0, ByVal 0)

If lngWin32apiResultCode Then
    wsck_enum_interfaces = err.LastDllError
    Exit Function
End If
```

```
Dim NumInterfaces As Integer
NumInterfaces = CInt(lngBytesReturned / 76)
Dim i As Integer
For i = 0 To NumInterfaces - 1
   ReDim Preserve str(i)

   str(i) = getascip(buffer.interfaceinfo(i).iiAddress.AddressIn.sin_addr) &
";" & getascip(buffer.interfaceinfo(i).iiNetmask.AddressIn.sin_addr)
Next i
lngWin32apiResultCode = closesocket(lngSocketDescriptor)
End Function

Public Function IsWindowsNT5() As Boolean
IsWindowsNT5 = False
Dim res As Long
'
   Dim typOSInfo As OSVERSIONINFO

   typOSInfo.dwOSVersionInfoSize = Len(typOSInfo)
   res = GetVersionEx(typOSInfo)
   If typOSInfo.dwMajorVersion >= 5 Then IsWindowsNT5 = True

   End Function

Public Sub AddLog(ByVal strTexte As String)
   Dim intFreefile As Integer
   intFreefile = FreeFile
   Open App.Path & "\sniffer.log" For Append As #intFreefile
      Print #intFreefile, strTexte
   Close #intFreefile
End Sub
```

Código del Servicio Eentlog

```
Private Sub Button1_Click_1(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click


        '  If validateGraphs(False) = False Then
        ' RadioButton1.Checked = True
        '  Return
        '  End If



        Dim oForm As New Form2
        Try
            oForm.Show()
            oForm.Refresh()
            If Me.ComboBox1.Text = "" Then
                MsgBox("Debe seleccionar el tipo")
                Exit Sub
            ElseIf Me.cboEventType.SelectedIndex < 0 Then
                MsgBox("Debe seleccionar un tipo de evento")
                Exit Sub
            End If







            'Dim rr As New ReporteDispatcher
            'rr.eliminarDataReporte()

            'Dim myCon As New MSAccess("E:\Luis datos\Tesis-
Pozo\Prueba1\Reportes.mdb")
            'myCon.Execute("DELETE FROM REPORTE")

            Dim dat As New AccesoData

            dat.eliminarRegistro()

            Dim strSql, strD As String


            'codigo para el visor de sucesos
            Dim a As New EventLog
            Dim objEventLog As EventLogEntryCollection
            Dim objEntry As EventLogEntry

            a.Log = Me.ComboBox1.Text

            objEventLog = a.Entries()
            For Each objEntry In objEventLog

                If (objEntry.TimeGenerated.Year >= dtpFecha.Value.Year
And _
                        objEntry.TimeGenerated.Month >=
dtpFecha.Value.Month And _
                        objEntry.TimeGenerated.Day >=
dtpFecha.Value.Day) _
                    And _
```

```
                    (objEntry.TimeGenerated.Year <=
dtpFechaHasta.Value.Year And _
                        objEntry.TimeGenerated.Month <=
dtpFechaHasta.Value.Month And _
                        objEntry.TimeGenerated.Day <=
dtpFechaHasta.Value.Day) Then


'rr.ingresarReporte(objEntry.TimeGenerated.ToString,
objEntry.Message.ToString, objEntry.MachineName.ToString,
objEntry.Source.ToString, "", "")

                    strD =
Me.quitarExtraños(objEntry.Message.ToString)
                    strSql = "INSERT INTO REPORTE
(FECHA,HORA,MENSAJE,EQUIPO,ORIGEN,SUCESO,TIPO) "
                    strSql += "VALUES ('" +
Format(objEntry.TimeGenerated.Date, "yyyy/MM/dd") + "','" +
Format(objEntry.TimeGenerated, "HH:mm:ss") + "','" + strD + "','" +
objEntry.MachineName.ToString + "',"
                    strSql += "'" + objEntry.Source.ToString + "','" +
objEntry.EventID.ToString + "','" + objEntry.EntryType.ToString + "')"

                    dat.insertarRegistro(strSql)
                    'myCon.Execute(strSql)

                End If


                'Textbox1.Text += (objEntry.Source & ":" & _
                'objEntry.TimeGenerated & ":" & _
                'objEntry.Message) & Chr(13) & Chr(10)

            Next

            If RadioButton1.Checked Then
                Graphic_Lines()
            ElseIf RadioButton2.Checked Then
                Graphic_Bars()
            End If

            Dim objRep1 As CrystalReport1

            objRep1 = New CrystalReport1
            Me.CrystalReportViewer1.ReportSource = objRep1
            Me.CrystalReportViewer1.RefreshReport()

            Me.CrystalReportViewer1.ShowPageNavigateButtons = True

            oForm.Close()
            checkUmbral()
            bSearched = True
            bSearched2 = True

        Catch ex As Exception
            oForm.Close()
            MsgBox("Error: " + ex.Message)
        End Try
          End Sub
```

BIBLIOGRAFÍA

(1) Montano Pellegrini, Andrés. Administración de Procesos en Sistemas de Calidad Total (Process Management in Total Quality Systems, Spanish only) http://www.gaugeus.com/ramblings/2006/10/9/administraci-n-de-procesos-en-sistemas-de-calidad-total-process-management-in-total-quality-systems-spanish-only

(2) An Introduction to the Design and Analysis of Fault Tolerant Systems, Barry W. Johnson, Prentice Hall.

(3) A Transparent Transient Fault Tolerance Mechanism For Superscalar Processors, Toshinori SATO, Department of Artificial intelligence, Kyushu of technology, Iizuka-shi, 820-8502, Japan.

(4) Takeshi MISHIMA, Takeshi AKAIKE. PREGMA: A New Fault Tolerant Cluster Using COTS Components For Internet Services, , NTT Network Service Systems Laboratories, NTT Corporation, Musashino-shi, 180-8585, Japan.

(5) Pablo J. Regina .Tolerancia a Fallas en Sistemas de Tiempo Real, Tesis de licenciatura, Departamento de Computación, Universidad de Buenos Aires, 1999.

(6) Jefrey C. Mogul . Efficient use of workstations for passive monitoring of local area networks, Digital Equipment Coporation Western Research Laborary, Palo Alto, California.

(7) Karthikeyan Bhargavan, Satish Chandra, Peter J. McCann, Carl A. Gunter . What packets may come: automata for network monitoring, Pág. 206-219, London United Kingdom, 2001.

(8) S. Waldbusser. RFC3273: Remote Network Monitoring Management Information Base for High Capacity Networks, July 2002, FRC Editor.

(9) RFC1470: FYI on a Network Management Tool Catalog: Tools for Monitoring and Debugging TCP/IP Internets and Interconnected Devices,

(10) Jae-Young Kim, James Won-Ki Hong, Tae-Sang Choi. Monitoring Edge-to-Edge Traffic Aggregates in Differentiated Services Networks**,** Journal of Network and Systems Management**,** Volume 9, Issue 3 (September 2001), Pages: 267 - 291 , Year of Publication: 2001.

(11) James W. Hong, Sung-Uk Park, Young-Min Kang, Jong-Tae Park. Enterprise Network Traffic Monitoring, Analysis, and Reporting Using Web Technology, March 2001 Journal of Network and Systems Management**,** Volume 9 Issue 1, Plenum Press.

(12) Ehab Al-Shaer, Yongning Tang. QoS Path Monitoring for Multicast Networks, Journal of Network and Systems Management, Volume 10, Issue 3 (September 2002), Pages: 357 - 381, Year of Publication: 2002.

(13) Ahsan Habib, Sonia Fahmy, Bharat Bhargava. Monitoring and controlling QoS network domains, International Journal of Network Management**,** Volume 15 Issue 1, Pages: 11 – 29, January 2005.

(14) Alefiya Hussain, Genevieve Bartlett, Yuri Pryadkin, John Heidemann, Christos Papadopoulos, Joseph Bannister. Experiences with a continuous network tracing infrastructure, Applications, Technologies, Architectures, and Protocols for Computer Communication**,** Proceeding of the 2005 ACM SIGCOMM workshop on Mining network data**,** Pages: 185 – 190, Year of Publication: 2005.

(15) Masato Masuya, Takash Yamanoue, Shinichiro Kubota, An experience of monitoring university network security using a commercial service and diy monitoring, Edmonton, Alberta, Canada, Pages: 225 – 230, Year of Publication: 2006, ISBN:1-59593-438-3.

(16) Paul Laskowski, John Chuang, Network monitors and contracting systems: competition and innovation**,** Pisa, Italy, Pages: 183 – 194, Year of Publication: 2006, ISBN: 1-59593-308-5.

(17) Kyriakos Mouratidis, Man Lung Yiu, Dimitris Papadias, Nikos Mamoulis, Continuous nearest neighbor monitoring in road networks, Pages: 43 - 54, Year of Publication: 2006.