Creación de software de análisis estadístico del tráfico de Internet aplicable a una red de área local

Ortega Galo, Velasco Saulo, Escuela Politécnica Nacional (EPN), Quito - Ecuador

***Resumen* – Este proyecto de titulación está orientado a desarrollar un software con una interfaz visual de escritorio que permita obtener datos estadísticos y gráficos del tráfico de red, diferenciando hosts, puertos y protocolos, que pueda ser ejecutada en sistemas operativos como Windows y Linux(CentOS y Ubuntu). Se usa como herramienta fundamental a WinPcap (Windows) y a libpcap(Linux) para la captura de paquetes del tráfico de Internet de una red LAN Ethernet y a jNetPcap para su decodificación desde una aplicación Java. Se usa NetBeans 6.5 como entorno de desarrollo, siguiendo los lineamientos del documento de requerimientos de software. Se describe brevemente el método de programación extrema y el diseño final alcanzado con las continuas iteraciones incrementales de desarrollo. En la etapa de diseño se utiliza UML(Unified Modeling Language) con sus diagramas de casos de uso, clases y de actividad para describir las relaciones y procesos principales de la aplicación final. En este documento se mostrarán los gráficos generados a partir de los datos capturados en el proceso de prueba y los escenarios de detección de anomalías.**

I. INTRODUCCIÓN

El servicio de Internet es uno de los puntos más críticos en un entorno corporativo, donde el uso óptimo de los recursos constituye el pilar fundamental para el crecimiento y estabilidad de una empresa.

Los encargados de regular el uso óptimo de este servicio son los administradores de red, siguiendo las normas de las políticas internas de la empresa.

El administrador de red debe hacer uso de herramientas de software para monitorear el ancho de banda de la conexión a Internet y también de la captura de paquetes. El problema surge al momento de la interpretación de los datos para la obtención de parámetros cuantitativos y cualitativos, en los cuales basar sus decisiones para establecer medidas restrictivas y correctivas.

El presente proyecto de titulación aborda este problema y se implementa como una solución la creación de un software con las siguientes directrices principales:

- Adquisición de datos basado en captura de paquetes.
- Almacenamiento de los valores capturados en una base de datos.
- Diferenciación de tráfico.
- Análisis de Estadística Descriptiva.
- Generación de gráficos y resúmenes de datos.

Estas características engloban los requerimientos que un software de este tipo debe poseer para tener una visión global del uso del servicio de Internet en una red local.

II. DESCRIPCIÓN DE LAS HERRAMIENTAS Y BIBLIOTECAS CLAVE

A. NetBeans

El Netbeans IDE es un ambiente de desarrollo integrado para la creación aplicaciones web, empresariales, escritorio y móviles para distintas plataformas como Java, PHP, JavasScript y Ajax, Groovy y Grails y C/C++. Es un producto libre y gratuito sin restricciones de uso.

Es una herramienta pensada para escribir, compilar, depurar y ejecutar programas.

B. WinPcap y libpcap

Libpcap de Unix provee de una interfaz de alto nivel para los sistemas de captura de paquetes y su análisis. Winpcap es el equivalente de libpcap para la plataforma Win32.

La potencialidad de estas bibliotecas puede ser aprovechada en varios tipos de herramientas de red que pueden ser de análisis, resolución de problemas, seguridad y monitorización. Más específicamente podrían ser:

- Analizadores de red y protocolos.
- Loggers de tráfico.
- Generadores de tráfico.
- Bridges y routers de nivel de usuario.
- Sistemas de detección de intrusos (NIDS).
- Escáneres de red.
- Herramientas de seguridad.

Sin embargo tienen limitaciones, es decir, no poseen la capacidad de bloquear, filtrar o manipular el tráfico generado por otros programas en la misma máquina. Por esta razón no puede ser usado en aplicaciones como limitadores de tráfico, planificadores de QoS y firewalls personales.

C. jNetPcap

jNetPcap es un API de desarrollo de software para el (SDK[1]) de Java, cuya función básica es la proveer de un "*envoltorio java*"[2] para la biblioteca *libpcap*[3] .

[1] Software Development Kit
[2] El término original en inglés es *java wrapper*

El objetivo de jNetPcap es el de proveer una mayor facilidad para el desarrollo de aplicaciones típicas de la biblioteca libpcap y WinPcap.

jNetPcap está conformada por una implementación java más una implementación nativa dependiente de plataforma. En el caso de sistemas Win32 la biblioteca es un .dll y en sistemas Unix es un fichero .so.

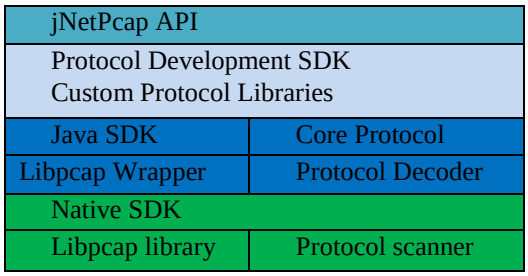

Fig. 1. Estructura de jNetPcap

D. Java

Java es un lenguaje de programación orientado a objetos desarrollado por Sun Microsystems pero actualmente es mantenido por Oracle Corporation. Este lenguaje contiene sintaxis de C y C++, pero que gestiona elementos de bajo nivel para evitar errores asociados a aquello.

Java se ha construido con extensas capacidades de conexión como TCP/IP, por lo tanto, contiene librerías de rutinas para acceder e interactuar con protocolos como ftp y http. Esto permite al programador acceder información a través de la red con tanta facilidad como ficheros locales.

Java posee las características de ser orientado a objetos, robusto, seguro, portable, multithreaded[4], dinámico y posee una arquitectura neutral.

III. Diseño e Implementación del Sistema

A. Programación extrema

Es un método de desarrollo ágil basado en procesos iterativos e incrementales y la participación activa del cliente.

A continuación un detalle del ciclo de entrega de la programación extrema.

• Selección de historias de usuario (requerimientos) y fragmentación en tareas más pequeñas y planificación de entrega: Para el desarrollo incremental en la planificación del proyecto se sintetiza las necesidades del cliente detalladas en un relato de sus necesidades. Luego de ello se realiza un planeación para ordenar todas las ideas del cliente y establecer tiempos. Posteriormente se realiza varias iteraciones para obtener una diversidad de versiones y analizar con el cliente hasta que se adapte a su necesidad.

[3] Biblioteca propia de sistemas Unix utilizada para captura de paquetes de red
[4] Significa que varios threads se ejecutan simultáneamente en un espacio de direcciones compartido.

• Desarrollar/integrar/probar el sofware: Se realiza la implementación, integración con requisitos o módulos anteriormente realizados y se efectúan pruebas para observar el comportamiento de las funcionalidades agregadas. Estas pruebas buscan verificar que los cambios no contengan errores lógicos o que no cumplan con los requisitos a fin de obtener una entrega que satisfaga las necesidades del cliente.

• Entregar y Evaluar el Sistema: Para finalizar la iteración se hace la prueba final la entrega de todo el sistema con el cliente para verificar su funcionalidad. Este ciclo continúa hasta que se hayan integrado todos los requisitos del cliente.

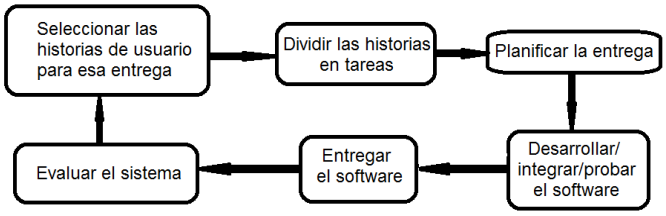

Fig. 1. Ciclo de entrega de la programación extrema [2]

Este método incluye otro principio básico, el cual es la programación en parejas, cuyo objetivo es la minimización de errores y aumento la productividad al escribir código de gran calidad, por la continua verificación del trabajo realizado entre ambos.

B. Descripción de requerimientos

Requerimientos del usuario resumidos en forma de diagrama UML.

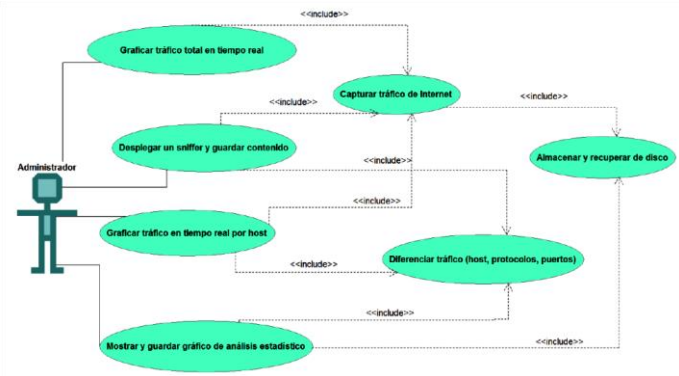

Fig. 2. Diagramas de casos de uso de los requerimientos del usuario

C. Descripción general del sistema

La solución fue dividida en dos módulos:

• TrafficStatistics: módulo que implementa todos los casos de uso que requieren captura de tráfico de Internet.
• QueryStatistics: módulo que implementa el análisis estadístico y resúmenes informativos de los datos almacenados por el módulo anterior.

Estos módulos fueron implementados en Java con la finalidad de proveer la característica multiplataforma.

Ambos módulos utilizan una base de datos implementada con MySQL para el almacenamiento y posterior análisis de los mismos como se muestra la Fig. 3.

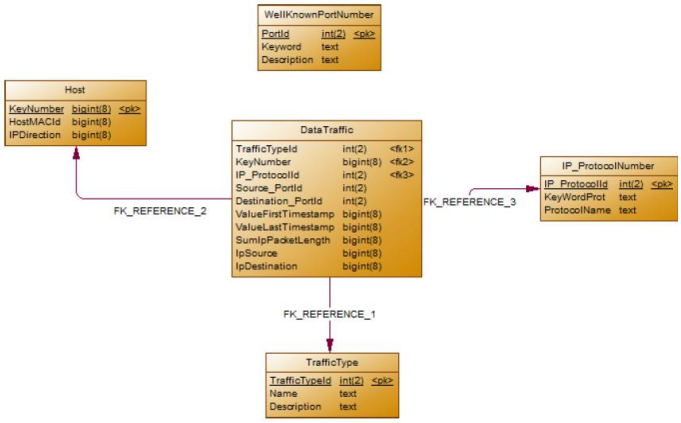

Fig. 3. Diagramas de casos de uso de los requerimientos del usuario

Tabla Host: Asigna un identificador entero autoincremental a cada estación de trabajo que va siendo ingresada en la tabla. El host es diferenciado por su MAC y dirección IP almacenados en un valor entero equivalente a su representación en octetos.

Tabla TrafficType: Identifica los tipos de tráfico existentes en una conexión a Internet, el tráfico entrante y el tráfico saliente. El tráfico entrante es identificado con el entero 0 y el saliente con el 1. Los otros campos de la tabla describen brevemente cada tipo de tráfico.

Tabla IP_ProtocolNumber: Lista cada uno de todos los posibles valores del campo protocolo dentro la cabecera del paquete IP que identifican los protocolos del siguiente nivel, asignados de acuerdo al RFC 1700, se incluyen los campos de sus siglas y nombre completo.

Tabla WellKnownPortNumber: Lista todos los puertos en el rango de 1 a 1024 denominados puertos bien conocidos, con sus siglas y descripción de acuerdo al RFC 1700. Estos valores son usados por los protocolos de transporte TCP y UDP para identificar una conexión.

Tabla DataTraffic: Tabla principal donde se añadirán los registros con la información recopilada en la captura de tráfico de Internet. Cada registro indica una conexión de una estación de trabajo de la red monitoreada con el exterior. Dada la importancia de esta tabla se describirán cada uno de sus campos:

TRAFFICTYPEID: clave externa para identificar el tipo de tráfico (entrante o saliente).

KEYNUMBER: clave externa que identifica la estación de trabajo a la que pertenece el registro.

IP_PROTOCOLID: clave externa para identificar el protocolo de capa superior en la cabecera del paquete IP.

SOURCE_PORTID: entero que indica el puerto origen en caso de que este exista. Este campo puede ser nulo en comunicaciones donde no se utilicen protocolos de capa transporte como ICMP.

DESTINATION_PORTID: entero que indica el puerto destino en caso de que este exista. Este campo puede ser nulo en comunicaciones donde no se utilicen protocolos de capa transporte como ICMP.

VALUEFIRSTTIMESTAMP: entero que almacena el valor del timestamp del primer paquete capturado de esta conexión.

VALUELASTTIMESTAMP: entero que almacena el valor del timestamp del momento inmediatamente anterior al ingreso del registro.

SUMIPPACKETLENGTH: entero que indica la cantidad total en bytes de los paquetes IP capturados incluidos las cabeceras, para esta conexión, en el lapso de tiempo comprendido desde el primer timestamp hasta el último.

IPSOURCE: dirección IP origen representada en su valor entero correspondiente. Puede ser privada o pública si el tráfico es saliente o entrante respectivamente.

IPDESTINATION: dirección IP destino representada en su valor entero correspondiente. Puede ser pública o privada si el tráfico es saliente o entrante respectivamente.

Nota: El valor del timestamp corresponde a la diferencia, medida en milisegundos, entre la hora actual y la medianoche del 1ero de Enero de 1970.

IV. Escenarios de Captura

El software puede ser instalado en los ambientes de red ilustrados por los gráficos que se muestran a continuación:

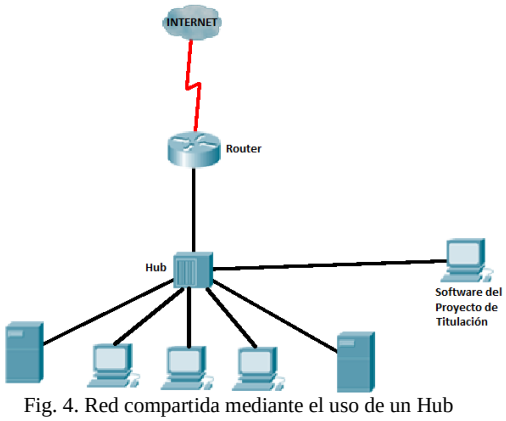

Fig. 4. Red compartida mediante el uso de un Hub

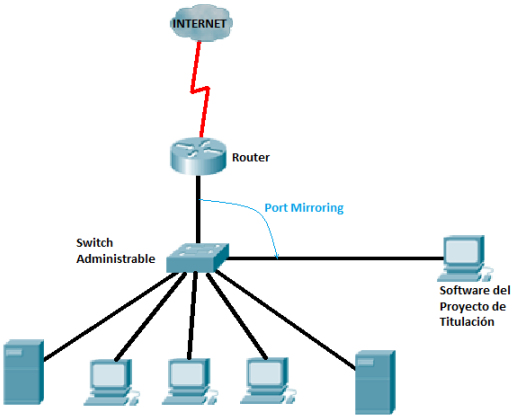

Fig. 5. Red conmutada mediante un Switch Administrable

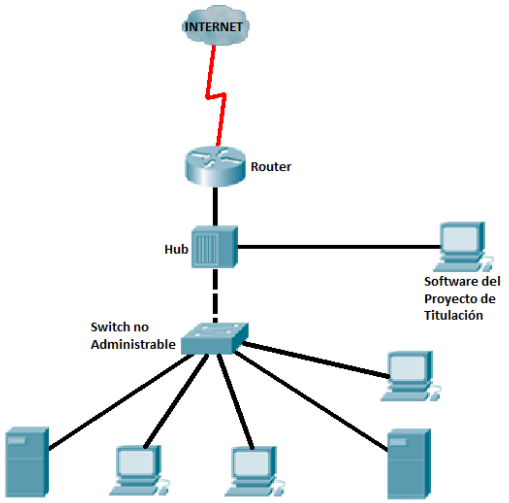

Fig. 6. Red conmutada mediante un Switch no Administrable

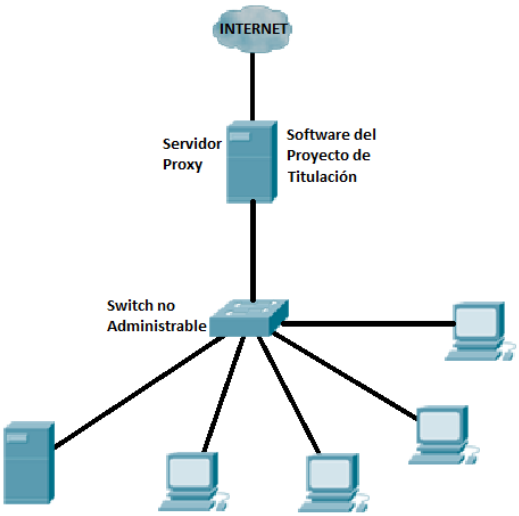

Fig. 7. Red conmutada mediante un Switch no Administrable y que usa un servidor Proxy

V. Pruebas

Para la realización de las pruebas se siguió el esquema mostrado en la Fig. 8.

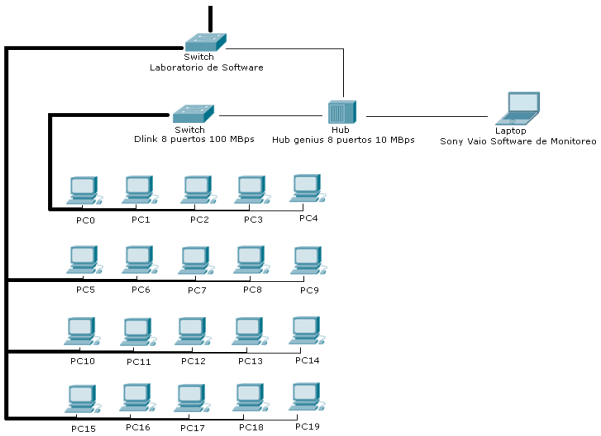

Fig. 8. Diagrama de conexión de red para monitoreo de datos.

La computadora portátil se conecta directamente al hub para monitorear todo el tráfico que genera las 5 estaciones de trabajo.

VI. Generación de gráficos y resúmenes estadísticos para los datos capturados

El software está en la capacidad de generar los siguientes gráficos y resúmenes estadísticos:

En tiempo real.

• Gráfico bitrate vs tiempo incoming y outgoing (Fig. 9).- Este es generado desde el inicio de la captura de datos y muestra en tiempo real el flujo entrante y saliente total de la red. Incluye un gradiente de color que fluctúa de verde (valores bajos) a rojo (valores altos) de acuerdo al límite fijado por el usuario. Este límite también es utilizado para fijar el valor de la tasa de transferencia a partir del cual se activará una alarma sonora.

• Monitoreo de paquetes en Modo Sniffer (Fig. 10).- Una vez seleccionados los respectivos filtros y eligiendo las estaciones de trabajo deseadas puede iniciarse el despliegue en formato de texto de los paquetes decodificados, sin interferir en el proceso de almacenamiento en la base de datos.

Datos almacenados en la base de datos.

• Tasa de transferencia promedio de uso de Internet (Fig. 11).- Este tipo de diagrama permite visualizar el promedio de tasa de transferencia en las horas señaladas por el usuario, representando un rango de tiempo determinado. Este promedio se lo realiza diferenciando estaciones de trabajo, protocolos y puertos.

• Porcentajes de uso de tráfico de Internet 2D y 3D (Fig. 12).- Este diagrama permite representar la cantidad total de datos entrantes o salientes para los protocolos TCP, UDP, ICMP o para un puerto específico en un pastel de porcentajes.

• Histograma y distribución de frecuencias acumuladas de protocolos de tráfico de Internet (Fig. 13).- Genera un histograma de frecuencias y su distribución de frecuencias acumuladas con los datos obtenidos de la base, estableciendo de manera automática rangos uniformes para los valores de ancho de banda de tráfico de Internet monitoreados.

También se incluye una tabla de resumen de estadística descriptiva, la cual contiene información sobre el número de valores de bitrate, la media, el error estándar de la media, la media recortada, la desviación estándar, además muestra el valor mínimo, el máximo e indicadores como el primer cuartil, la mediana (segundo cuartil) y el tercer cuartil.

• Reconstrucción de historial de la base de datos con líneas (Fig. 14).- Por medio de un diagrama de líneas se reconstruyen los valores de la base de datos del tráfico de Internet entrante o saliente, para las estaciones de trabajo, protocolos y rango de tiempo seleccionados.

• Reconstrucción de historial de la base de datos en pasos (Fig. 15).- Realiza una representación gráfica en el tiempo de los valores de bitrate recuperados de la base de datos, representados en pasos. Estos valores son filtrados por protocolos, puertos y estaciones de trabajo de acuerdo al rango de tiempo elegido por el usuario.

• Series de tiempo (Fig. 16).- Grafica series de tiempo para los distintos protocolos y estaciones de trabajo, asignando el intervalo de tiempo más adecuado para la representación de los datos monitoreados de manera automática, de acuerdo al rango de tiempo elegido por el usuario.

• Series de tiempo en pasos (Fig. 17).- El mismo principio que el anterior pero usando pasos para graficar cada intervalo regular.

•DNS reverso y ranking para IPs más utilizadas (Fig. 18).- Utilizando un diagrama de barras se mostrará la cantidad de bytes transferidos para las distintas direcciones IP registradas en la base de datos por cada host seleccionado. Incluye un mecanismo que realiza el proceso de DNS reverso para la resolución de nombres de dominio de las direcciones IP.

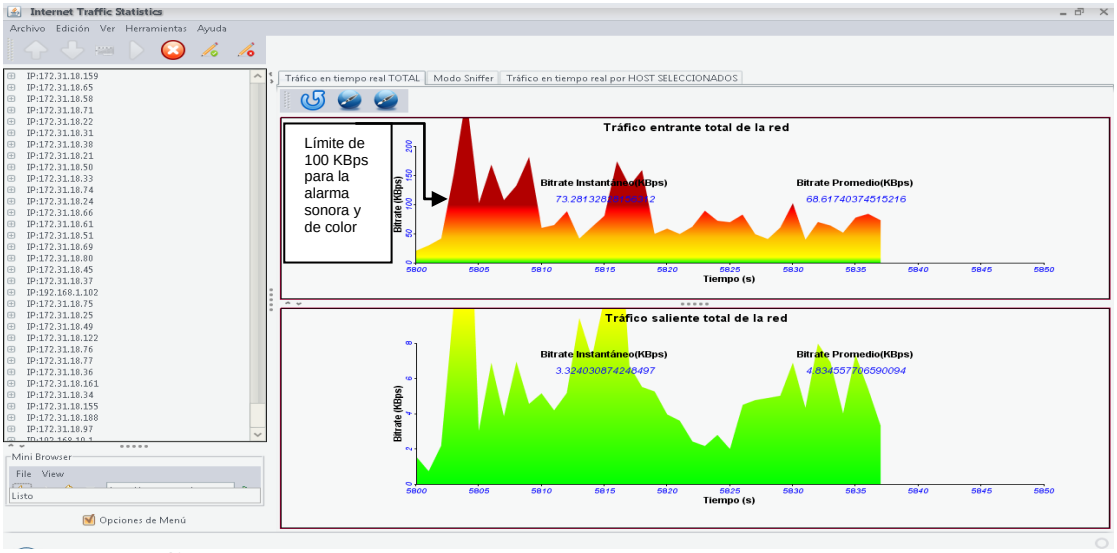

Fig. 9. Gráfico bitrate vs tiempo incoming y outgoing de las pruebas de monitoreo sobrepasando los 100KBps.

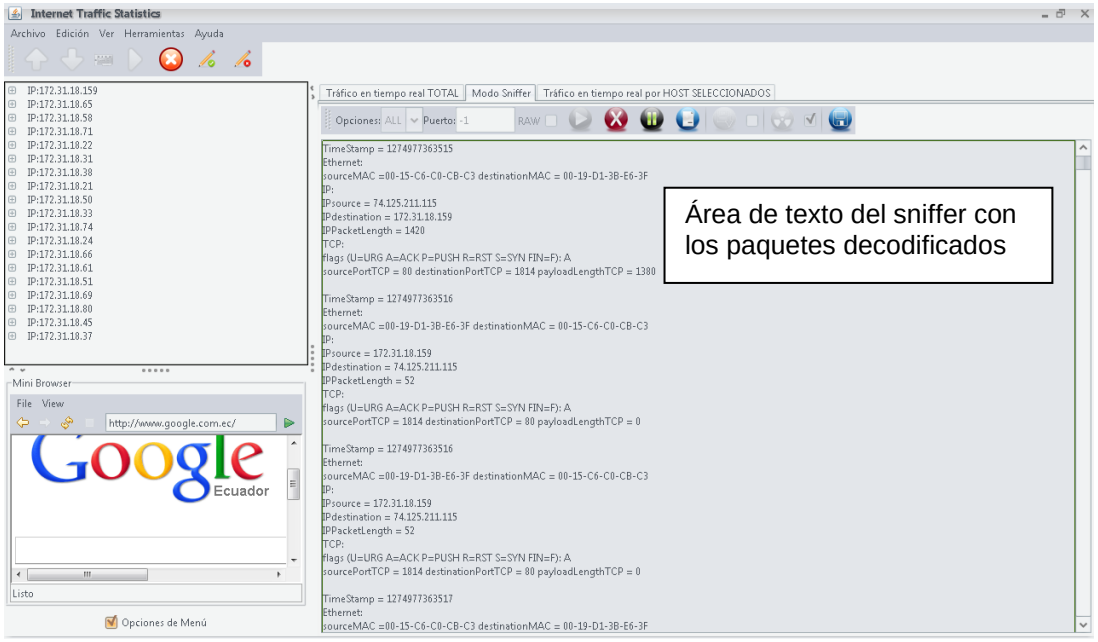

Fig. 10. Monitoreo de paquetes en Modo Sniffer.

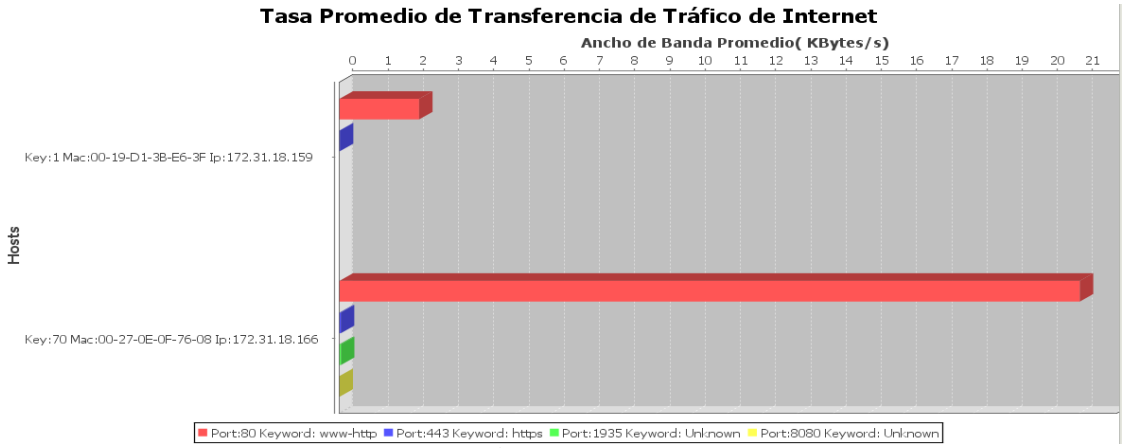

Fig. 11. Tasa de transferencia promedio de uso de Internet.

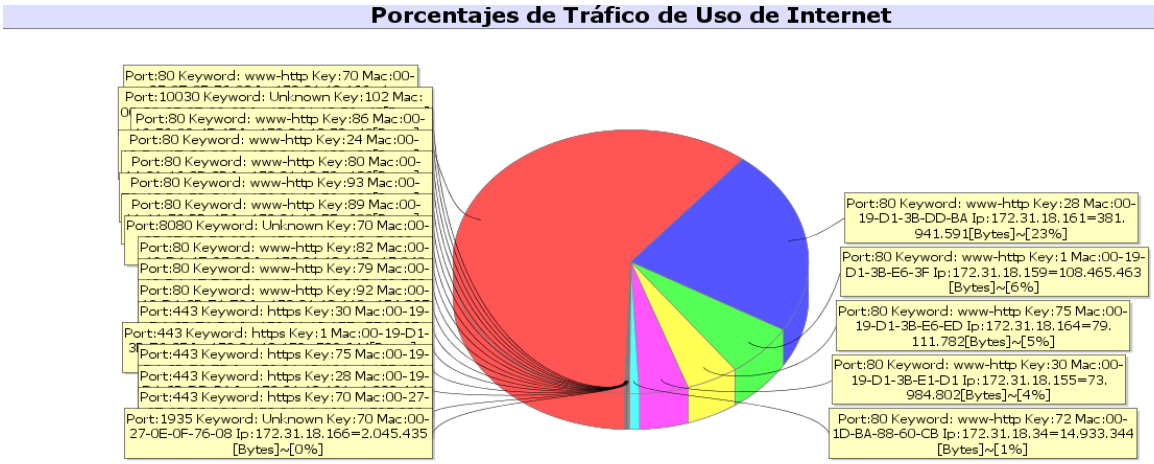

Fig. 12. Porcentajes de uso de tráfico de Internet 3D

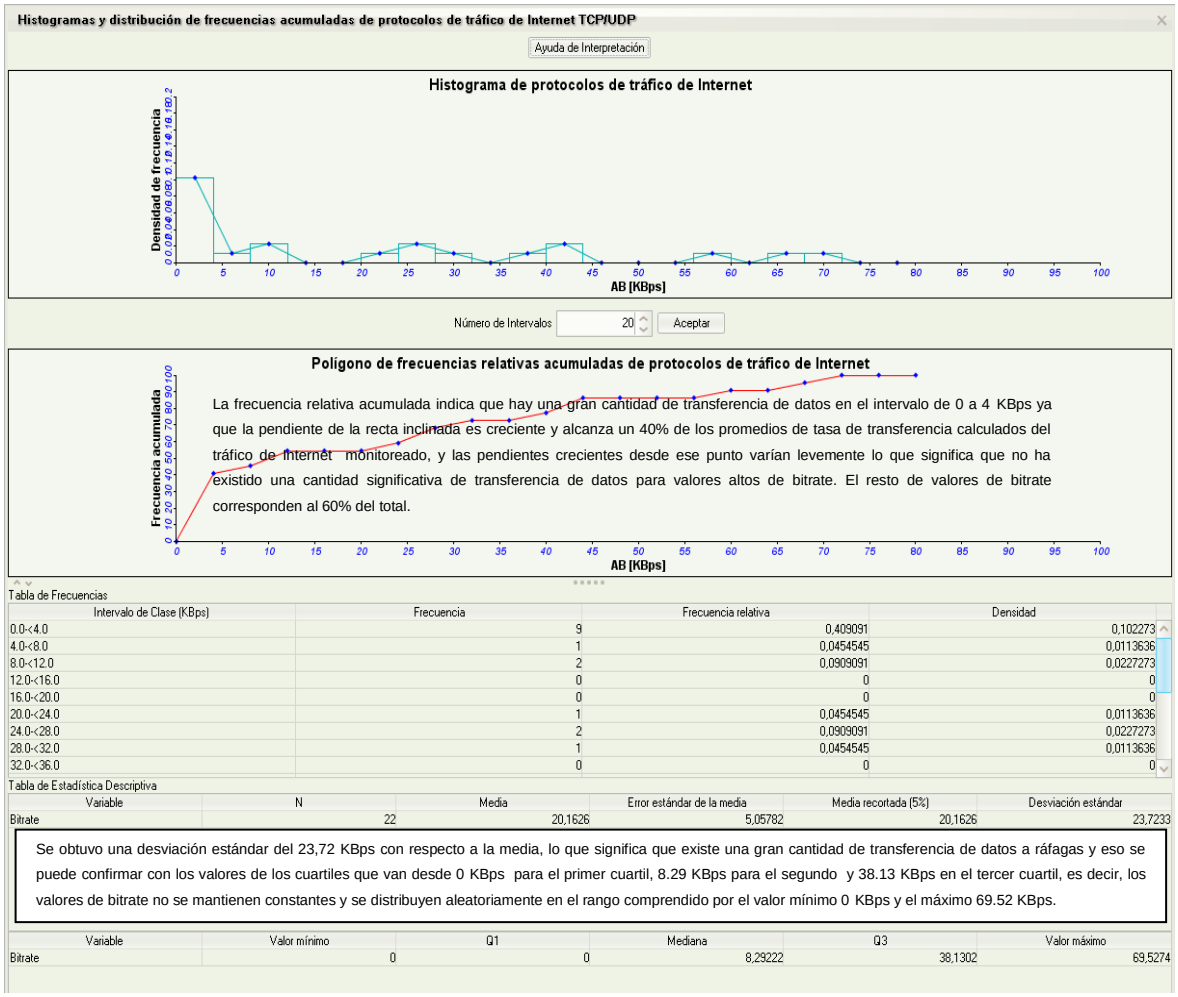

Tabla de Frecuencias

Intervalo de Clase (KBps)	Frecuencia	Frecuencia relativa	Densidad
0.0-<4.0	9	0,409091	0,102273
4.0-<8.0	1	0,0454545	0,0113636
8.0-<12.0	2	0,0909091	0,0227273
12.0-<16.0	0	0	0
16.0-<20.0	0	0	0
20.0-<24.0	1	0,0454545	0,0113636
24.0-<28.0	2	0,0909091	0,0227273
28.0-<32.0	1	0,0454545	0,0113636
32.0-<36.0	0	0	0

Tabla de Estadística Descriptiva

Variable	N	Media	Error estándar de la media	Media recortada (5%)	Desviación estándar
Bitrate	22	20,1626	5,05782	20,1626	23,7233

Se obtuvo una desviación estándar del 23,72 KBps con respecto a la media, lo que significa que existe una gran cantidad de transferencia de datos a ráfagas y eso se puede confirmar con los valores de los cuartiles que van desde 0 KBps para el primer cuartil, 8.29 KBps para el segundo y 38.13 KBps en el tercer cuartil, es decir, los valores de bitrate no se mantienen constantes y se distribuyen aleatoriamente en el rango comprendido por el valor mínimo 0 KBps y el máximo 69.52 KBps.

Variable	Valor mínimo	Q1	Mediana	Q3	Valor máximo
Bitrate	0	0	8,29222	38,1302	69,5274

Fig. 13. Histograma de protocolos y Distribución de frecuencias relativas acumuladas de los datos monitoreados.

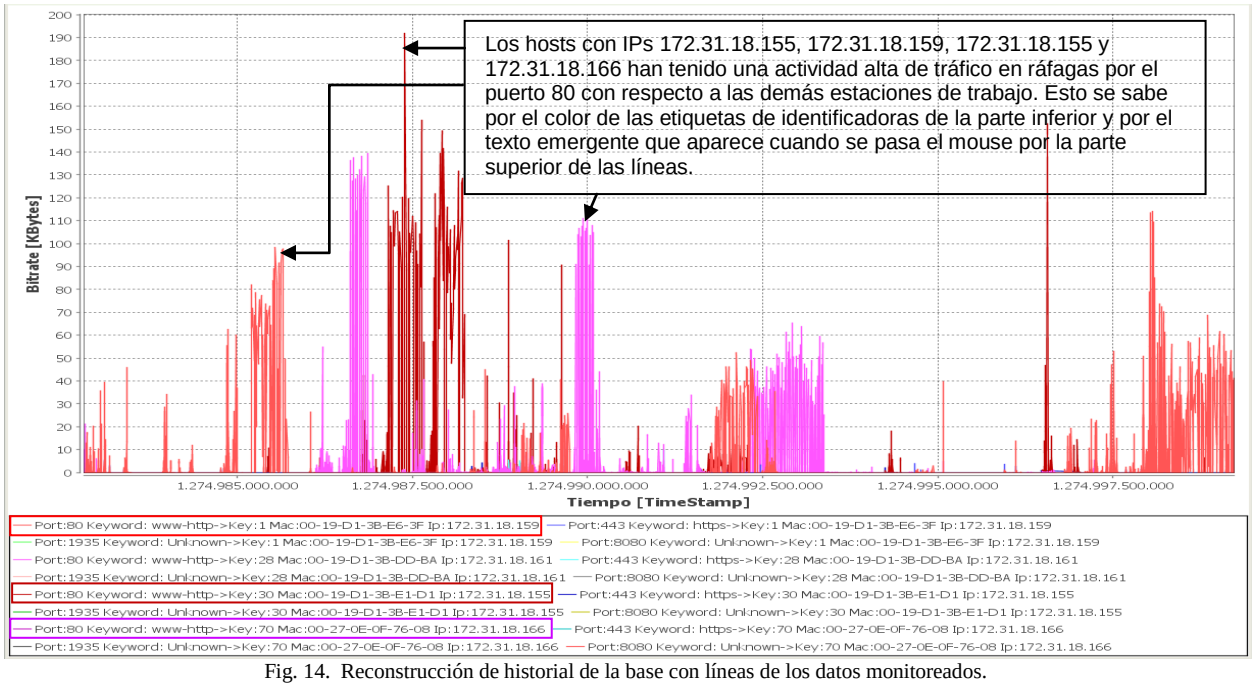

Fig. 14. Reconstrucción de historial de la base con líneas de los datos monitoreados.

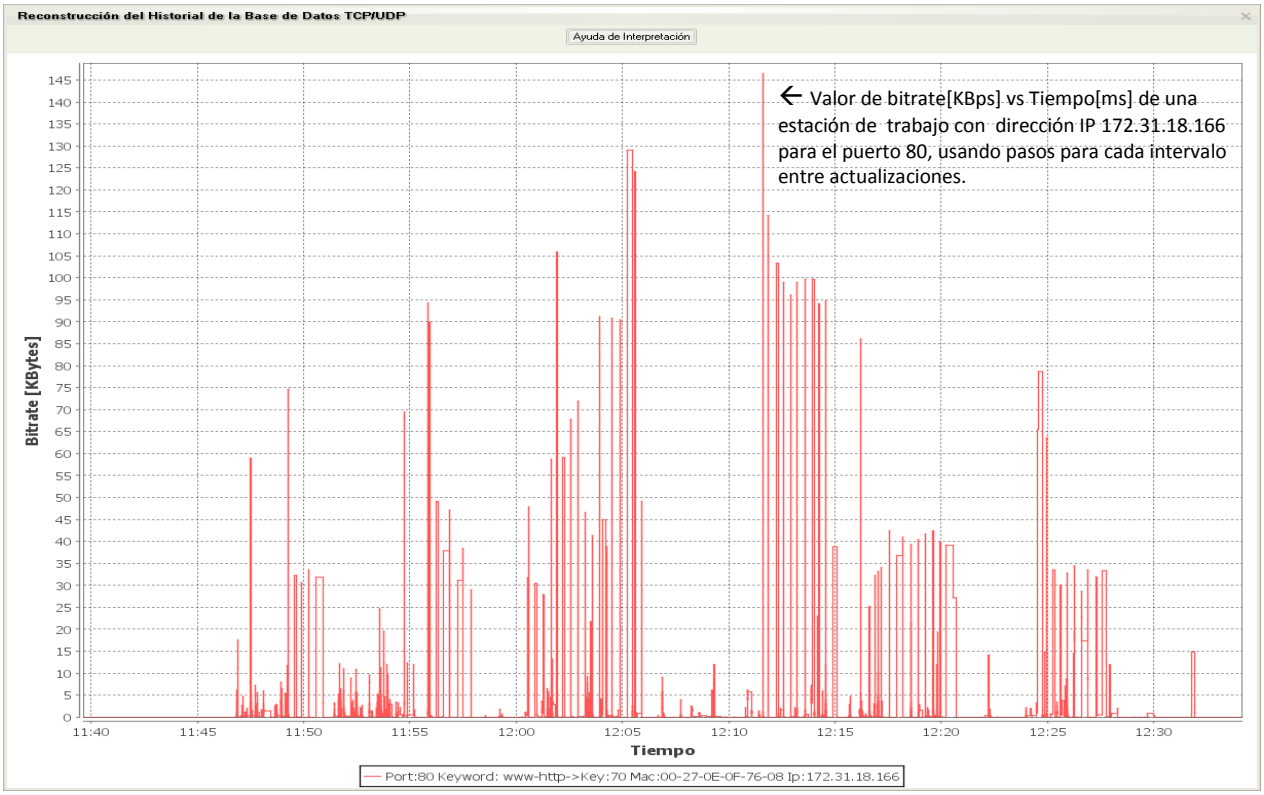

Fig. 15. Reconstrucción de historial de la base de datos en pasos.

Fig. 16. Series de tiempo de los datos monitoreados.

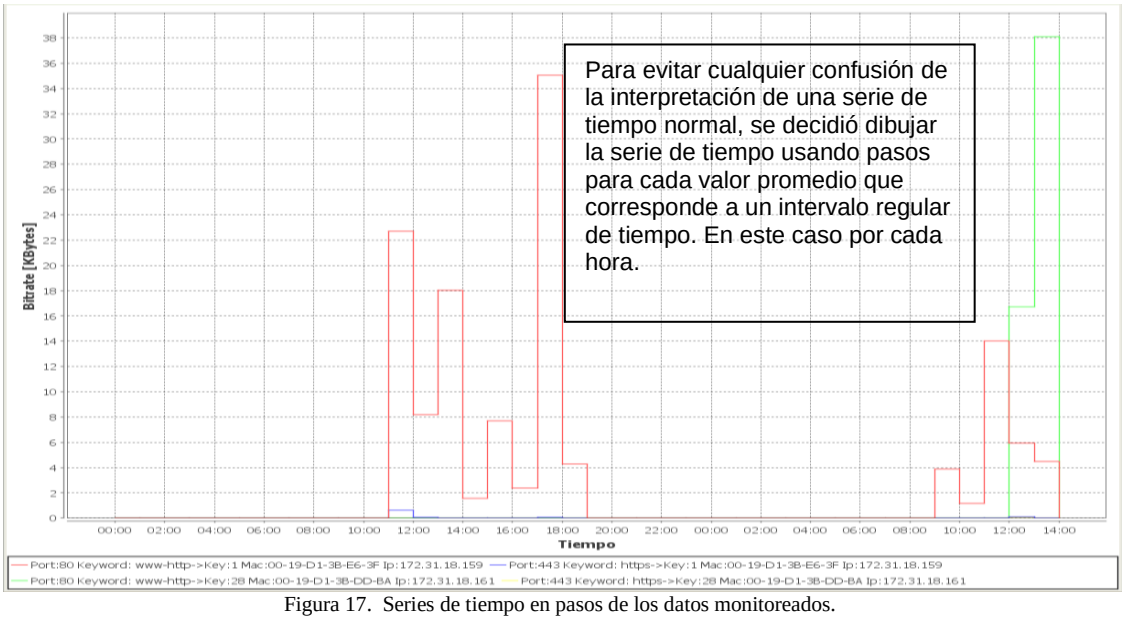

Figura 17. Series de tiempo en pasos de los datos monitoreados.

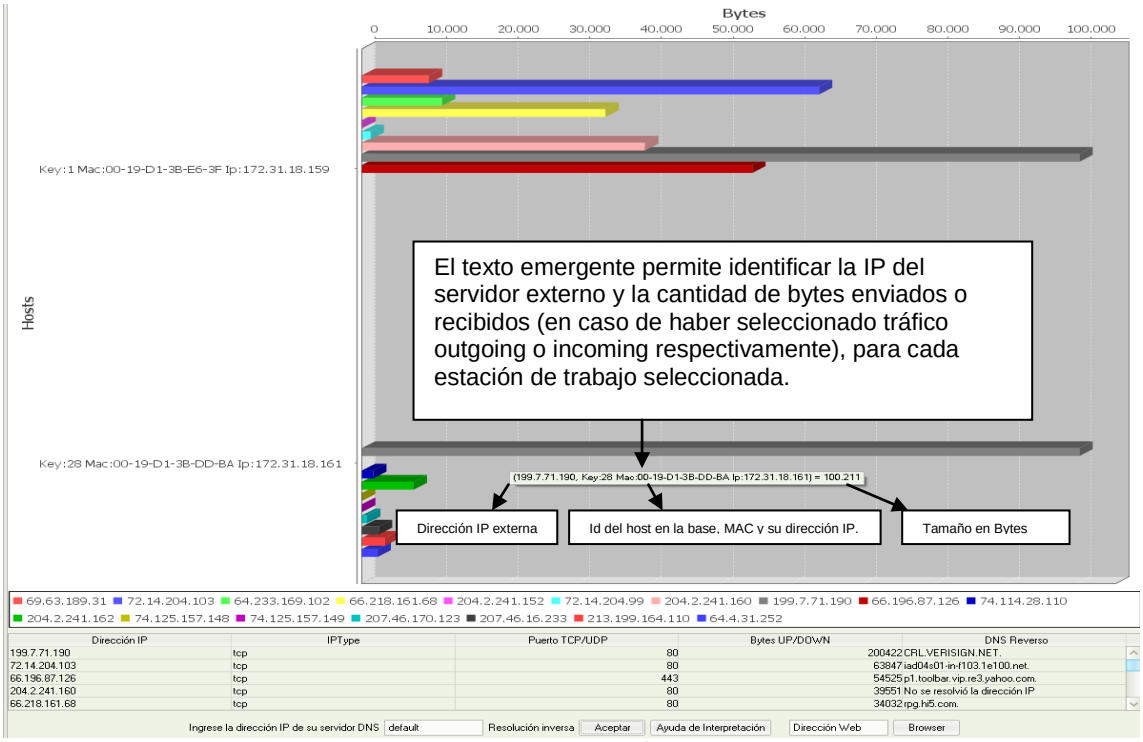

Fig. 18. DNS reverso y ranking para IPs más utilizadas de los datos monitoreados.

VII. ESCENARIOS DE DETECCIÓN DE ANOMALÍAS

A. Primer Escenario – Detección de descargas no autorizadas

Para este propósito pueden utilizarse de manera combinada los gráficos en tiempo real y los generados a partir de los datos capturados en la base de datos. Los primeros permiten reconocer al instante que se está utilizando de manera no autorizada la conexión a Internet, si se sobrepasa el límite fijado por el usuario del programa. En el segundo caso el usuario puede utilizar los distintos tipos de gráficos para establecer el volumen de datos y el bitrate promedio, para de esta manera detectar a la estación de trabajo que ha realizado esta actividad.

B. Segundo Escenario –Detección de posibles envíos de spam o replicación de gusanos informáticos

Con la finalidad de detectar este tipo de anomalía el usuario del programa debe tener presente sobre todo al tráfico saliente de las estaciones de trabajo, el uso de cada una de ellas permite analizar este tráfico desde distintas perspectivas, tales como bitrate promedio, cantidad de bytes y reconstrucción temporal.

C. Segundo Escenario –Detección de posibles envíos de spam o replicación de gusanos informáticos

A partir de los datos obtenidos del monitoreo de tráfico de Internet se requiere saber si existe algún comportamiento inusual de una o de varias estaciones de trabajo. Siendo un comportamiento sospecho iniciar una conexión repetitiva a una dirección IP. Este tipo de anomalía describiría un posible ataque, ya que estaría conectándose con la estación de trabajo remotamente infectada para así causar daño en la red local. Para este tipo de análisis se requiere utilizar la herramienta "DNS reverso y ranking para IPs más utilizadas".

D. Cuarto Escenario – Detección de patrones de comportamiento

Con los datos obtenidos del monitoreo de tráfico de Internet se requiere saber si existe algún comportamiento inusual con alguna estación de trabajo específica, donde las descargas de grandes archivos no están permitidas. El objetivo es verificar si existe un patrón de conexiones repetitivas en horas y periodos regulares del día. Para este tipo de análisis se requiere utilizar la herramienta "Series de tiempo en pasos". Este tipo de gráfico permite observar con gran facilidad en que momentos del día se registran los picos en los valores de descargas y si estos se repiten a lo lago de los días.

El administrador de red debe aplicar el reglamento interno de la institución o empresa para tomar las respectivas acciones correctivas.

VIII. CONCLUSIONES

- Este documento muestra la forma en que un software generador de gráficos estadísticos del tráfico de red ayuda al administrador de red de manera significativa en la toma de decisiones para la optimización y control del uso del ancho de banda de la conexión a Internet.

- Al ser la adquisición de datos basada en la captura de paquetes de tráfico de Internet de una red de área local disminuye la complejidad de instalación pues no requiere realizar cambios significativos a nivel de hardware y se evita el tener que levantar servicios de gestión de red como es el caso de SNMP, cuya implementación puede requerir de mucho esfuerzo en el caso de una red con muchas estaciones de trabajo con sus respectivas limitaciones.

- El uso de la metodología de desarrollo de programación extrema facilitó el desarrollo del software ya que se atendió directamente a las necesidades del usuario y con sus continuas iteraciones con el cliente, permitió llegar a un producto innovador y efectivo. De esta forma se ha comprobado que la presente metodología es valedera y eficiente al momento de realizar un software de manera ágil con resultados eficientes.

- Esta herramienta que se ha desarrollado con el fin de monitorear el tráfico de Internet se puede también utilizar para realizar estudios sobre los diferentes comportamientos y tendencias sociales de las personas que usan el Internet como forma de acceso al mundo virtual.

- Finalmente el costo de licencias para el desarrollo es nulo por la utilización de herramientas gratuitas y de APIs opensource, como es el caso de Java, MySQL y NetBeans, lo cual representa una gran ventaja frente a otras herramientas similares.

IX. REFERENCIAS

[1] Stevens, Perdita; Pooley, Rob; "*Utilización de UML en Ingeniería del Software con Objetos y Componentes*," Traducción de la Primera Edición. Editorial Addison Wesley. España. 2002

[2] Sommerville, Ian; "*Ingeniería de software*," Séptima edición. Editorial Addison Wesley. España. 2005

[3] Deitel, Harvey M.; Deitel, Paul J.; *"Cómo programar Java,"* Quinta Edición. Editorial Pearson

[4] Delap, Scott; *"Desktop Java Live,"* Primera Edición. Editorial SourceBeat. Colorado. 2005

[5] Navidi, William; *"Estadística para Ingenieros,"* Traducción de la Primera Edición. Editorial McGraw-Hill. México. 2006

[6] Spiegel, Murray R.; Stephens, Larry J.; "*Estadística*," Tercera Edición. Editorial McGraw-Hill. México. 2002

[7] Stallings, William; *"Comunicaciones y Redes de Computadoras,"* Sexta Edición. Prentice Hall

X. BIOGRAFÍAS

Saulo Velasco, nació en la ciudad de Ibarra el 3 de noviembre de 1985. Realizó sus estudios de primaria en el Instituto Inocencio Jácome de San Antonio de Ibarra. Obtuvo su título de Bachiller en Ciencias Físico-Matemáticas en el Colegio Nacional Teodoro Gómez de la Torre de la misma ciudad. Cursó sus estudios universitarios en la Escuela Politécnica Nacional en la carrera Ingeniería en Electrónica y Redes de Información. Actualmente labora en el campo del desarrollo de software.

Áreas de interés: informática y redes, animación digital, generative art.

(saulo_velasco@hotmail.com)

Galo Efrén Ortega Álvarez, nació en la ciudad de Quito el 25 de marzo de 1983. Realizó sus estudios de primaria en el Instituto Francisco Febres Cordero "La Salle" de Quito. Obtuvo su título en Bachiller en Ciencias Físico-Matemáticas en el Colegio San Luis Gonzaga de la misma ciudad. Cursó sus estudios universitarios en la Escuela Politécnica Nacional en la carrera de Ingeniería en Electrónica y Redes de Información.

Actualmente labora en el campo del desarrollo de software con herramientas libres.

(galoefren@hotmail.com)

Xavier Alexander Calderón Hinojosa, nació el 16 de Agosto de 1972 en Quito-Ecuador, se graduó en el Colegio La Salle, especialidad Físico-Matemático en el año 1990. En el 2011 obtiene el título de Ingeniero en Electrónica y Telecomunicaciones en la Escuela Politécnica Nacional y en el 2002 se gradúa como Máster en Tecnologías de la Información en Fabricación en la Universidad Politécnica de Madrid. Actualmente trabaja en la Escuela Politécnica Nacional donde es profesor principal a tiempo completo, Miembro del Consejo de Departamento (Departamento de Electrónica y Redes de Información). Jefe del Laboratorio de Informática de la Facultad de Ingeniería Eléctrica y Electrónica y Director Proyecto Semilla.

(xavier.calderon@epn.edu.ec)