UNIVERSIDAD MAYOR DE SAN ANDRÉS
FACULTAD DE CIENCIAS PURAS Y NATURALES
CARRERA DE INFORMÁTICA

TESIS DE GRADO

"GESTION DE DISPOSITIVOS DE RED"

PARA OPTAR AL TITULO DE LICENCIATURA EN INFORMÁTICA

MENCIÓN: INGENIERÍA DE SISTEMAS INFORMÁTICOS

POSTULANTE: EDWIN CHURATA APAZA

TUTOR: Lic. LUISA VELASQUEZ LOPEZ M Sc.

REVISOR: Lic. JAVIER HUGO REYES PACHECO

LA PAZ – BOLIVIA

2009

Dedicatoria

A mis padres Antonio Churata Mamani y Nievez Apaza Nina, quienes siempre guiaron mis pasos dándome ejemplo de respeto, voluntad , esfuerzo además por darme todo el cariño, apoyo y comprensión que me brindaron para concluir mis estudios. A mis queridos hermanos por todo el apoyo desprendido que siempre me brindaron.

Agradecimientos

A la M.Sc Luisa Velasquez Lopez, ya que gracias a su guía, apoyo, concejos y por su dedicada revisión, corrección a detalle y las indicaciones para dar un adecuado curso en la realización dela presente tesis.

Agradecer de manera mas sincera a mi revisor, Lic Javier Hugo Reyes Pacheco, que a pesar de su apretada agenda, tuvo la buena voluntad, paciencia y dedicación en guiar y revisar el presente trabajo.

RESUMEN

La creciente integración de computadoras y comunicación dentro de un sistema único, a llevado a una sociedad nueva y de rápido crecimiento.

Esta tendencia de crecer rápidamente es en realidad totalmente universal. Los adelantos tecnológicos permiten que las comunicaciones tengan lugar a través de grandes distancias cada ves con mayor facilidad.

Hoy es más la interrelación y la interdependencia de oficinas y lugares de trabajo dispersos. Los nuevos conceptos de administración exigen una disponibilidad de los datos que cumplan, la persona adecuada debe recibir, la información adecuada, en el momento adecuado.

Esto obliga a solucionar cualquier problema que ocurra inmediatamente en los dispositivos de red.

Para la realización de la presente tesis se hace uso del método científico, en sus etapas de análisis, síntesis, inducción, concreción.

Se plantea la construcción de una herramienta de gestión de red para monitorear los dispositivos de red, para la construcción de la herramienta se utilizo de un lenguaje de programación(java), además de el paquete de snmp para java, el protocolo de red SNMP.

CONTENIDO

1 Presentación

1.1 Introducción 1

1.2 Antecedentes 3

1.3 Situación Problemática 3

1.4 Formulación del Problema 4

1.5 Objeto de Estudio 4

1.6 Justificación y relevancia 4

1.7 Hipótesis 5

1.8 Objetivos 5

1.8.1 Objetivo General 5

1.8.2 Objetivos Específicos 5

1.9 Alcances y Límites 6

1.10 Aportes 6

1.10.1 Aporte Teórico 6

1.10.2 Aporte Práctico 6

1.11 Metodología 6

1.11.1 Método Científico 6

2 Marco Referencial

2.1 Red de Computadoras 8

2.1.1 Clasificación de Redes 8

2.1.2 Tipo de Redes 9

2.1.3 Componentes Básicos de las Redes de Ordenadores 11

2.1.4 Clasificación de las Redes de Ordenadores 13

2.2 Software 15

2.2.1 Herramienta de Software 15

2.3 Protocolo 15

2.3.1 Protocolos de Red 15

2.4 Java 17
2.4.1 Librerías Java 17
2.5 Protocolo Administración de Red (SNMP) 18
2.5.1 Componentes Básicos 19
2.5.2 Comandos Básicos 20
2.5.3 Base de información de Administración SNMP (MIB) 20
2.5.4 Mensajes SNMP 21
2.6 Software Libre 23
2.7 Criptografía 26
2.7.1 Algoritmo RSA 28
2.7.1.1 Generación de Claves 28
3 Proceso de Investigación
3.1 Descripción de Métodos 30
3.2 Método Científico 30
3.2.1 Planteamiento del Problema 31
3.2.2 Construcción del Modelo Teórico 31
3.2.3 Planteamiento de la hipótesis 31
3.2.4 Construcción de la herramienta 31
3.3 Métodos que Contempla el Método Científico 32
3.3.1 Análisis 32
3.3.2 Síntesis 32
3.3.3 Inducción 32
3.3.4 Concreción 32
3.4 Métodos Propuestos 32
3.5 Descripción Formal 33
3.6 Ambiente de desarrollo 34
3.7 Descripción del Sistema 34

3.7.1 Creación y Envío de un Mensaje 34
3.7.2 Recepción de una Respuesta a un Mensaje y su Decodificación 36
3.7.3 Tratamiento de una ResponsePdu Decodificada 36
3.7.4 Cómo Solicitar todos los Valores de la MIB 37
3.7.4.1 Solicitar Toda la Tabla MIB 37
3.7.4.2 Solicitar Tabla de la MIB 38
3.7.5 Escuchar Traps e Informs 39
3.7.6 Encriptación 40
3.7.6.1 Generación de Claves 41
3.7.6.2 Demostración 41
4 Discusión y Conclusiones
4.1 Presentación de la Herramienta 43
4.2 Conclusión 61
4.2.1 Recomendación 63
Bibliografía

LISTA DE GRAFICAS

Fig.2.3.1.1 Serie de Protocolos dentro del Modelo OSI 16
Fig.2.6.1 Pantalla de NetMGR 23
Fig.2.3.2 Pantalla de Zenoss 24
Fig.2.3.3 Pantalla de ZABBIX 25
Fig.2.3.4 Pantalla de Cactil 25
Fig. 3.1.1 Proceso de Investigación 30
Fig.3.6.1.1 Creación de Variables Bindings y Añadir a un Vector 35
Fig. 3.6.1.2 Crear una Request PDU e Inicializar sus Valores 35
Fig. 3.6.1.3 Crear un Mensaje Codificarle y Envio 35
Fig. 3.6.2.1 Como Recibir la Respuesta 36
Fig. 3.6.2.1 Como Decodificar un Mensaje y la pdu Contenida 36
Fig.3.6.3.1 Comprobar el RequestID y ErrorStatus 36
Fig. 3.6.3.2 Añadir Variables Bindings a una Lista de Variables 37
Fig. 3.6.3.3 Ejemplo de Lanzamiento de Excepciones en SnmpSesion 37
Fig. 3.6.3.4 Retornar las Variables Respuesta 37
Fig. 3.6.4.1.1 Raíz de la MIB 37
Fig. 3.6.4.1.2 Llamada a getBulkMIBEntry 38
Fig. 3.6.4.1.3 Tratamiento de las Respuestas Parciales 38
Fig. 3.6.4.1.4 Retornar las Variables Binding 38
Fig. 3.6.4.2.1 Prefijo de OID 39
Fig. 3.6.5.1 Recepción del Mensaje 39
Fig. 3.6.5.2 Decodificación del Mensaje y la PDU 39
Fig. 3.6.5.3 Tratamiento de las Notificaciones 40
Fig. 3.6.6.2.1 Declaracion de Variables 42
Fig. 3.6.6.2.2 Métodos de GenerPrimos, Generaclave 42
Fig.4.1.1 Opción Fichero 44
Fig.4.1.2 Opción Editar 45
Fig.4.1.3 Opción Ayuda 46
Fig. 4.1.1.1 Operación Get 46
Fig. 4.1.1.2 Formulario Operación Get 47
Fig. 4.1.1.3 Comprobación de datos operación Get 48
Fig. 4.1.1.4 Resultado Operación Get 48
Fig. 4.1.1.5 Operación GetNext 48

Fig. 4.1.1.6 Formulario Operación GetNext 49
Fig. 4.1.1.7 Comprobación de datos operación getNext 49
Fig. 4.1.1.8. Resultado Operación GetNext 50
Fig. 4.1.1.9 Operación Set 50
Fig. 4.1.1.10 Formulario Operación Set 51
Fig. 4.1.1.11 Comprobación de datos operación Set 51
Fig. 4.1.1.12 Operación GetBulk 52
Fig. 4.1.1.13 Formulario Operación GetBulk 52
Fig. 4.1.1.14 Comprobación de datos operación GetBulk 53
Fig. 4.1.1.15 Resultado Operación GetBulk 53
Fig. 4.1.1.16 Operación Inform 54
Fig. 4.1.1.17 Formulario Operación Inform 55
Fig. 4.1.1.18 Comprobación de datos operación Inform 55
Fig. 4.1.1.19 Resultado Operación Inform 56
Fig. 4.1.1.20 Escucha de Traps 56
Fig. 4.1.1.21 Recibiendo notificaciones 57
Fig. 4.1.1.22 Confirmación de llegada de traps 57
Fig. 4.1.1.23 Operación Extraer Toda la MIB 58
Fig. 4.1.1.24 Formulario Extraer Toda la MIB 58
Fig. 4.1.1.25 Resultado operación Extraer Toda la MIB 59
Fig. 4.1.1.26 Operación Obtener tabla de OIDs 59
Fig. 4.1.1.27 Formulario Operación Obtener tabla de OIDs 60
Fig. 4.1.1.28 Resultado operación Obtener tabla de OIDs 61

LISTA DE TABLAS

Tabla 1.3.1 Matriz Causa-Efecto 3
Tabla 1.3.1 Lista de Problemas 3
Tabla 2.5.1 Tabla de Atributos de SNMP 18
Tabla 2.5.4.1 Los puertos Comúnmente Utilizados para SNMP 21
Tabla 2.5.4.2 Formato de Consultas y Respuestas 21

1 PRESENTACION

1.1 INTRODUCCION

La administración de redes abarca un amplio número de dispositivos. En general, se suelen tratar con muchos datos estadísticos e información sobre el estado de distintas partes de la red, y se realizan las acciones necesarias para ocuparse de fallos y otros cambios.

La técnica más primitiva para la monitorización de una red es hacer "pinging" a los hosts críticos; el "pinging" se basa en un datagrama de "echo", que es un tipo de datagrama que produce una réplica inmediata cuando llega al destino. La mayoría de las implementaciones TCP/IP incluyen un programa (generalmente, llamado "ping") que envía un eco a un host en concreto. Si recibimos réplica, sabremos que host se encuentra activo, y que la red que los conecta funciona; en caso contrario, sabremos que hay algún error. Mediante "pinging" a un razonable número de ciertos hosts, podremos normalmente conocer qué ocurre en la red. Si los ping a todos los hosts de una red no dan respuesta, es lógico concluir que la conexión a dicha red, no funciona.

Si sólo uno de los hosts no da respuesta, pero los demás de la misma red responden, es razonable concluir que dicho host no funciona.

El ping es una manera rápida de verificar la conectividad entre host. Se hace desde el prompt de Windows (MS-DOS).

Técnicas más sofisticadas de monitorización necesitan conocer información estadística y el estado de varios dispositivos de la red.

Para ello necesitará llevar la cuenta de varias clases de datagramas, así como de errores de varios tipos.

Este tipo de información será más detallada en los gateway, puesto que el gatewa clasifica los datagramas según protocolos e, incluso, él mismo responde a ciertos tipos de datagramas. Sin embargo, los bridges e incluso los repetidores con buffer contabilizan los datagramas reenviados, errores de interface.

Para llevar a cabo la monitorización. Usamos un protocolo SGMP y SNMP, ambos diseñados para permitirnos recoger información y cambiar los parámetros de la configuración y otras entidades de la red. Podemos ejecutar los correspondientes protocolos en cualquier host de nuestra red.

SGMP está disponible para varios gateway comerciales, así como para sistemas Unix que actúan como gatewa. Cualquier implementación SGMP necesita que se proporciones un conjunto de datos para que pueda empezar a funcionar, y tienen mecanismos para ir añadiendo informaciones que varían de un dispositivo a otro.

A finales de 1988 apareció una segunda generación del protocolo SGMP cual es SNMP, que es ligeramente más sofisticado y necesita más información para trabajar y, para ello, usa el llamado MIB (Management Información Base).

Es así, que estos protocolos persiguen el mismo objetivo: permitirnos recoger información crítica de una forma estandarizada. Se ordena la emisión de datagramas UDP desde un programa de administración de redes que se encuentra ejecutando en alguno de los hosts de red. Generalmente, la interacción es bastante simple, con el intercambio de un par de datagramas: una orden y una respuesta.

La administración de red se lleva a cabo al nivel de IP, por lo que se pueden controlar dispositivos que estén conectados en cualquier red accesible desde la Internet, y no únicamente aquellos localizados en la propia red local. Evidentemente, si alguno de los dispositivos de encaminamiento con el dispositivo remoto a controlar no funciona correctamente, no será posible su monitorización ni reconfiguración.

Un Administrador de red está compuesto por dos elementos: el agente (agent), y el gestor (manager). Es una arquitectura cliente-servidor, en la cual el agente desempeña el papel de servidor y el gestor hace el de cliente.

1.2 ANTECEDENTES

Las redes informáticas se han convertido en un elemento indispensable para el funcionamiento de cualquier organización. Así que la utilización de protocolos de red es muy importante en el desarrollo de estas.

Entre trabajos realizados utilización de estas herramientas en la Universidad Mayor de San Andrés Facultad de Ciencias Puras y Naturales Carrera de Informática están:

Autor: José Andrés Villa Loza, Tema: "Usabilidad sobre el protocolo WAP y tecnología celular GSM y GPRS", Año: 2006, Objetivo: La creación de "UGWP-51-1" para facilitar la usabilidad en aplicaciones WAP.

Autor: Víctor Eduardo Silvila, Tema: "Sistema de Información de Gestión Bajo la Intranet en High Tech Center", Año: 2007, Objetivo: Desarrollar un sistema de información de gestión, el cual se desarrollara bajo el enfoque de intranet, que permita procesar información en forma rápida y eficiente.

Entre trabajos realizados utilización de estas herramientas en otras universidades esta:

Autor: Universidad el Bosque, Tema: Descripción del Protocolo SNMP para la Gestión de Redes de Datos y Diseño de un Software para su Implementación en Dispositivos heterogéneos, Año:2002, Dirección Web: https://www.unbosque.edu.co/

1.3 SITUACION PROBLEMATICA

Tabla 1.3.1 Lista de Problemas

Numero	Problemas
1	Falta de optimización en los dispositivos de una red (LAN)
2	Tiempo de respuesta inadecuada en la entrega de reportes sobre el funcionamiento de los dispositivos de red (LAN)
3	Falta de control en los dispositivos de una red (LAN)
4	Atraso en cambio o reparación de dispositivos defectuosos de una red (LAN)
5	No existe una administración entre dispositivos de red

Tabla 1.3.1 Matriz Causa-Efecto

Problemas	Causa	Efecto	Solución
1	Bajo nivel en el desempeño de trabajo de la red	Bajo funcionamiento de la red	Coadyuvar con la detección de dispositivos de una red con fallas
2	Toma de decisiones tardía	Perdida de fiabilidad en los reportes	Obtener reportes confiables y en tiempo real

3	Falla en el funcionamiento del sistema de red	Mala recepción de mensajes entre terminales de red	Monitorear cada dispositivo de red
4	Recursos no usados en su totalidad	Que otros dispositivos de red queden ociosos	Realizar un seguimiento de desempeño de cada dispositivo de red
5	Mayor tiempo y esfuerzo en la detección de fallas entre los dispositivos de un red	Bajo funcionamiento de la red	Implementar software de Adm. de dispositivos de red

Fuente [L.V.L., 2009]

1.4 FORMULACION DEL PROBLEMA

¿La herramienta para la gestión de los dispositivos de una red "LAN" permitirá una buena y ágil gestión en los dispositivos de una red?

1.5 OBJETO DE ESTUDIO

Con el desarrollo de la herramienta propuesta de gestión de los dispositivos de red, se proporcionara un instrumento útil que permita gestionar los dispositivos de red de manera eficiente.

1.6 JUSTIFICACIÓN Y RELEVANCIA

Es conveniente desarrollar esta herramienta "Gestión de los dispositivos de red" por ser necesaria el monitoreo y seguimiento de desempeño de cada dispositivo de red en cualquier organización.

Desde el punto de vista social la gestión de los dispositivos de red es indispensable para el funcionamiento de cualquier organización. También tomar en cuenta que sin una adecuada administración de los dispositivos de red que soporte muchas de las actividades rutinarias de gran cantidad de flujo de datos en una organización, estas no podrían funcionar adecuadamente.

Antes de 1988 el protocolo SGMP se desarrollo para la administración de red para sistemas Unix.

A finales de 1988 apareció una segunda generación del protocolo SNMP, que es ligeramente más sofisticado y necesita más información para trabajar y, para ello, usa el llamado MIB (Management Information Base). Y compuesto por dos elementos: el agente

(agent), y el gestor (manager). En una arquitectura cliente-servidor, en la cual el agente desempeña el papel de servidor y el gestor hace el de cliente.

Gracias a la ingeniería del software el desarrollo de esta herramienta será sinergetico, dinámico y libre de entropías,

1.7 HIPÓTESIS

Hi: La herramienta para la gestión de los dispositivos de una red **ayudara** a mejorar la administración de los dispositivos de una red LAN

1.8 OBJETIVOS

1.8.1 OBJETIVO GENERAL

Desarrollar una herramienta de software que posibilite la administración de los dispositivos de red.

1.8.2 OBJETIVOS ESPECÍFICOS

- Investigar y analizar sobre los protocolos Administración de Red
- Analizar y seleccionar sobre técnicas de envió y recepción de mensajes protocolo Administración de Red
- Realizar un estudio de lenguajes de programación
- Seleccionar un método de criptografía.
- Desarrollo de la herramienta de gestión de dispositivos de red

1.9 ALCANCES Y LÍMITES

El alcance de este proyecto está limitado dentro del área de las redes de telecomunicaciones correspondientes a los dispositivos de una red de área loca si es posible, pero se la llevara a cavo a nivel de computadora.

Puesto que alcanzar los objetivos planteados permitirá establecer una Herramienta que ayude a administrar los dispositivos en una red de área local que lograran: Coadyuvar con la detección de dispositivos de una red con fallas, realizar un seguimiento de desempeño de

cada dispositivo de red a demás de realizar un escaneo de los componentes de una computadora, en lo posible de implementar la herramienta (software) en una red de área local o realizar pruebas de escaneo en una computadora.

1.10 APORTES

Los aportes de investigación del presente trabajo de tesis, se resumen en los siguientes puntos.

1.10.1 APORTE TEORICO

Para la ejecución del la presente tesis se aplicara conocimientos teóricos de redes de computadoras, paquete de programación, protocolos de red, algoritmos de encriptación y la ingeniería de software para la planificación, organización, y ejecución del prototipo.

1.10.1 APORTE PRÁCTICO

El desarrollo del prototipo se pone en practica la parte teorica, en la utilización del lenguaje de programación java, utilizando el paquete de snmp de java, además de el algoritmo de encriptación RSA.

1.11 METODOLOGIA

Para el desarrollo de la presente tesis se toma en cuenta el método científico, que sirve de guía para la organización del proceso de investigación, mediante los mismos se cumplirán con los requisitos necesarios para la conclusión de los objetivos planteados

1.11.1 METODO CIENTIFICO

Se tiene las siguientes definiciones que nos explican el concepto de lo que es el método científico.

1 El método científico es un conjunto de procedimientos lógicos que sigue la investigación científica para describir la relación interna y externa de los procesos de la realidad natural y social.

2 Llamamos método científico a la serie ordenada de que se hace uso en la investigación científica para obtener la extensión de nuestros conocimientos.

3 Se entiende por método científico al conjunto de procesos que el hombre debe emplear en la investigación y demostración de la verdad

El método científico esta apoyado en una variedad de métodos generales, como ser el análisis, la deducción, la inducción. A continuación se describen los principales métodos utilizados.

El Análisis: es la descomposición de algo en sus elementos.

El método analítico consiste en la separación de las partes de un todo para estudiarlas en forma individual.

La síntesis: es la reconstrucción de todo lo descompuesto por el análisis.

El método deductivo es aquel que parte de datos generales aceptados como validos para llegar a una conclusión de tipo particular.

El método inductivo es aquel que parte de los datos particulares para llegar a conclusiones generales.

La abstracción: consiste en un examen critico y cuidadoso de los fenómenos, notado y analizado los diferentes factores y circunstancias que aparecen influenciarlos.

La concreción: se proponen explicaciones tentativas o hipótesis, que deben ser probadas mediante experimentos. Si la experiencia repetida no las contradice pasan a ser teorías. Las teorías mismas sirven como guías para nuevos experimentos i constante mente están siendo sometidas a pruebas. En la teoría se aplica razonamiento lógico y deductivo al modelo. [Sampieri R., 2003]

2 MARCO REFERENCIAL

2.1 RED DE COMPUTADORAS

Una red de computadoras (también llamada red de ordenadores o red informática) es un conjunto de equipos (computadoras) conectados por medio de cables, señales, ondas o cualquier otro método de transporte de datos, que comparten información (archivos), recursos (CD-ROM, impresoras, otros.) y servicios (acceso a internet, e-mail, chat, juegos).

Para simplificar la comunicación entre programas (aplicaciones) de distintos equipos, se definió el Modelo OSI por la ISO, el cual especifica 7 distintas capas de abstracción.

2.1.1 Clasificación de redes

- **Por alcance:**
  - Red de área personal (PAN)
  - Red de área local (LAN)
  - Red de área de campus (CAN)
  - Red de área metropolitana (MAN)
  - Red de área amplia (WAN)

- **Por método de la conexión**:
  - Medios guiados: cable coaxial, cable de par trenzado, fibra óptica y otros

- Medios no guiados: radio, infrarrojos, microondas, láser y otras redes inalámbricas.

❖ **Por relación funcional:**
- Cliente-servidor
- Igual-a-Igual (p2p)

Arquitecturas de red

❖ **Por Topología de red:**

- Red de bus
- Red de estrella
- Red de anillo (o doble anillo)
- Red en malla (o totalmente conexa)
- Red en árbol
- Red Mixta (cualquier combinación de las anteriores)

❖ **Por la direccionalidad de los datos (tipos de transmisión):**

- Simplex (unidireccionales), un Equipo Terminal de Datos transmite y otro recibe. (p. ej. streaming)
- Half-Duplex (bidireccionales), sólo un equipo transmite a la vez. También se llama Semi-Duplex (p. ej. una comunicación por equipos de radio, si los equipos no son full dúplex, uno no podría transmitir (hablar) si la otra persona está también transmitiendo (hablando) porque su equipo estaría recibiendo (escuchando) en ese momento).
- Full-Duplex (bidireccionales) , ambos pueden transmitir y recibir a la vez una misma información. (p. ej. videoconferencia).

2.1.2 TIPO DE REDES

❖ **Red pública:** una red publica se define como una red que puede usar cualquier persona y no como las redes que están configuradas con clave de acceso personal. Es una red de computadoras interconectados, capaz de compartir información y que permite comunicar a usuarios sin importar su ubicación geográfica.

❖ **Red privada:** una red privada se definiría como una red que puede usarla solo algunas personas y que están configuradas con clave de acceso personal.

- **Red de área Personal (PAN):** (Personal Area Network) es una red de ordenadores usada para la comunicación entre los dispositivos de la computadora (teléfonos incluyendo las ayudantes digitales personales) cerca de una persona. Los dispositivos pueden o no pueden pertenecer a la persona en cuestión. El alcance de una PAN es típicamente algunos metros.

  Las PAN se pueden utilizar para la comunicación entre los dispositivos personales de ellos mismos (comunicación del intrapersonal), o para conectar con una red de alto nivel y el Internet (un up link).

  Las redes personales del área se pueden conectar con cables con los buses de la computadora tales como USB y FireWire. Una red personal sin hilos del área (WPAN) se puede también hacer posible con tecnologías de red tales como IrDA y Bluetooth.

- **Red de área local (LAN):** una red que se limita a un área especial relativamente pequeña tal como un cuarto, un solo edificio, una nave, o un avión. Las redes de área local a veces se llaman una sola red de la localización. Nota: Para los propósitos administrativos, LANs grande se divide generalmente en segmentos lógicos más pequeños llamados los Workgroups. Un Workgroups es un grupo de las computadoras que comparten un sistema común de recursos dentro de un LAN.

- **Red del área del campus (CAN):** Se deriva a una red que conecta dos o más LANs los cuales deben estar conectados en un área geográfica específica tal como un campus de universidad, un complejo industrial o una base militar.

- **Red de área metropolitana (MAN):** una red que conecta las redes de un área dos o más locales juntos pero no extiende más allá de los límites de la ciudad inmediata, o del área metropolitana. Las rebajadoras múltiples, los interruptores y los cubos están conectados para crear a una MAN.

- **Red de área amplia (WAN):** es una red de comunicaciones de datos que cubre un área geográfica relativamente amplia y que utiliza a menudo las instalaciones de transmisión proporcionadas por los portadores comunes, tales como compañías del teléfono.

Las tecnologías WAN funcionan generalmente en las tres capas más bajas del Modelo de referencia OSI: la capa física, la capa de transmisión de datos, y la capa de red.

2.1.3 COMPONENTES BÁSICOS DE LAS REDES DE ORDENADORES

❖ **Computador**

La mayoría de los componentes de una red media son las comoutadoras individuales, generalmente son sitios de trabajo (incluyendo ordenadores personales) o servidores.

❖ **Tipos de sitios de trabajo**

Hay muchos tipos de sitios de trabajo que se pueden incorporar en una red particular, algo de la cual tiene exhibiciones high-end, sistemas con varios CPU, las cantidades grandes de RAM, las grandes cantidades de espacio de almacenamiento en disco duro, u otros componentes requeridos para las tareas de proceso de datos especiales, los gráficos, u otros usos intensivos del recurso.

❖ **Tipos de servidores**

En las siguientes listas, hay algunos tipos comunes de servidores y de su propósito.

- **Servidor de archivo:** almacena varios tipos de archivos y los distribuye a otros clientes en la red.

- **Servidor de impresiones:** controla una o más impresoras y acepta trabajos de impresión de otros clientes de la red, poniendo en cola los trabajos de impresión (aunque también puede cambiar la prioridad de las diferentes impresiones), y realizando la mayoría o todas las otras funciones que en un sitio de trabajo se realizaría para lograr una tarea de impresión si la impresora fuera conectada directamente con el puerto de impresora del sitio de trabajo.

- **Servidor de correo:** almacena, envía, recibe, enruta y realiza otras operaciones relacionadas con email para los clientes de la red.

- **Servidor de fax:** almacena, envía, recibe, enruta y realiza otras funciones necesarias para la transmisión, la recepción y la distribución apropiadas de los fax.

- **Servidor de la telefonía:** realiza funciones relacionadas con la telefonía, como es la de contestador automático, realizando las funciones de un sistema interactivo para la respuesta de la voz, almacenando los mensajes de voz, encaminando las llamadas y controlando también la red o el Internet, p. ej., la entrada excesiva del IP de la voz (VoIP), etc.

- **Servidor proxy:** realiza un cierto tipo de funciones a nombre de otros clientes en la red para aumentar el funcionamiento de ciertas operaciones (p. ej., prefetching y depositar documentos u otros datos que se soliciten muy frecuentemente), también sirve seguridad, esto es, tiene un Firewall. Permite administrar el acceso a internet en una Red de computadoras permitiendo o negando el acceso a diferentes sitios Web.

- **Servidor del acceso remoto(RAS):** controla las líneas de módem de los monitores u otros canales de comunicación de la red para que las peticiones conecten con la red de una posición remota, responden llamadas telefónicas entrantes o reconocen la petición de la red y realizan los chequeos necesarios de seguridad y otros procedimientos necesarios para registrar a un usuario en la red.

- **Servidor de uso:** realiza la parte lógica de la informática o del negocio de un uso del cliente, aceptando las instrucciones para que se realicen las operaciones de un sitio de trabajo y sirviendo los resultados a su vez al sitio de trabajo, mientras que el sitio de trabajo realiza el interfaz operador o la porción del GUI del proceso (es decir, la lógica de la presentación) que se requiere para trabajar correctamente.

- **Servidor web:** almacena documentos HTML, imágenes, archivos de texto, escrituras, y demás material Web compuesto por datos (conocidos colectivamente como contenido), y distribuye este contenido a clientes que la piden en la red.

- **Servidor de reserva:** tiene el software de reserva de la red instalado y tiene cantidades grandes de almacenamiento de la red en discos duros u otras formas del almacenamiento (cinta, etc.) disponibles para que se utilice con el fin de asegurarse de que la pérdida de un servidor principal no afecte a la red. Esta técnica también es denominada clustering.

- **Impresoras:** muchas impresoras son capaces de actuar como parte de una red de ordenadores sin ningún otro dispositivo, tal como un "print server", a actuar como intermediario entre la impresora y el dispositivo que está solicitando un trabajo de impresión de ser terminado.

- **Terminal tonto:** muchas redes utilizan este tipo de equipo en lugar de puestos de trabajo para la entrada de datos. En estos sólo se exhiben datos o se introducen. Este tipo de terminales, trabajan contra un servidor, que es quien realmente procesa los datos y envía pantallas de datos a los terminales.

- **Otros dispositivos:** hay muchos otros tipos de dispositivos que se puedan utilizar para construir una red, muchos de los cuales requieren una comprensión de conceptos más avanzados del establecimiento de una red de la computadora antes de que puedan ser entendidos fácilmente (e.g., los cubos, las rebajadoras, los puentes, los interruptores, los cortafuegos del hardware, etc.). En las redes caseras y móviles, que conecta la electrónica de consumidor los dispositivos tales como consolas vídeo del juego está llegando a ser cada vez más comunes.

2.1.4 CLASIFICACIÓN DE LAS REDES DE ORDENADORES

- **Por capa de red**

Las redes de ordenadores se pueden clasificar según la capa de red en la cual funcionan según algunos modelos de la referencia básica que se consideren ser estándares en la industria tal como el modelo OSI de siete capas y el modelo del TCP/IP de cinco capas.

- **Por la escala**

Las redes de ordenadores se pueden clasificar según la escala o el grado del alcance de la red, por ejemplo como red personal del área (PAN), la red de área local (LAN), red del área del campus (CAN), red de área metropolitana (MAN), o la red de área amplia (WAN).

- **Por método de la conexión**

Las redes de ordenadores se pueden clasificar según la tecnología que se utiliza para conectar los dispositivos individuales en la red tal como HomePNA, línea comunicación, Ethernet, o LAN sin hilos de energía.

- **Por la relación funcional**

Las redes de computadores se pueden clasificar según las relaciones funcionales que existen entre los elementos de la red, servidor activo por ejemplo del establecimiento de una red, de cliente y arquitecturas del Par-a-par (workgroup). También, las redes de ordenadores son utilizadas para enviar datos a partir del uno a otro por el hardrive.

- **Por topología de la red**

Define como están conectadas computadoras, impresoras, dispositivos de red y otros dispositivos. En otras palabras, una topología de red describe la disposición de los cables y los dispositivos, así como las rutas utilizadas para las transmisiones de datos. La topología influye enormemente en el funcionamiento de la red.

Las topologías son las siguientes: bus, anillo o doble anillo, estrella, estrella extendida, jerárquica y malla.

- **Por los servicios proporcionados**

Las redes de ordenadores se pueden clasificar según los servicios que proporcionan, por ejemplo redes del almacén, granjas del servidor, redes del control de proceso, red de valor añadido, red sin hilos de la comunidad, etc.

- **Por protocolo**

Las redes de ordenadores se pueden clasificar según el protocolo de comunicaciones que se está utilizando en la red. Ver los artículos sobre la lista de los apilados del protocolo de red y la lista de los protocolos de red para más información.

[Douglas E., 2002]

2.2 SOFTWARE

Software. Programas de computadora, estructuras de datos y su documentación que sirven para hacer efectivo el método lógico, procedimiento o control requerido

2.2.1 HERRAMIENA DE SOFTWARE

Herramientas de software de sistema. son tecnología que ayudan a desarrollar el software. Por tanto, la herramienta case deberá adaptarse a un software de sistema,

[Pressman., 2002]

2.3 PROTOCOLO

Descripción de formal de formato de mensajes y reglas que dos o mas maquinas(computadoras) deben seguir para intercambiar mensajes. Los protocolos pueden describir detalles de bajo nivel de las interfaces de maquina a maquina(por ejemplo, el orden en que los bits de un octeto se envían a través de un cable) o el intercambio de programas de aplicación(por ejemplo, la forma en que un programa transfiere un archivo a través de una red de redes) La mayor parte de los protocolos incluye descripciones intuitivas de las interacciones esperadas así como especificaciones mas formales, utilizando modelos de maquina de estado finito. [Douglas E., 2002]

2.3.1 PROTOCOLOS DE RED

Los conjuntos de protocolos de red más utilizados hoy en día y cotejar las capas que los integran con las del modelo OSI. De esta forma, lograremos una visión clara y sencilla del modo en que operan estas pilas de protocolos reales y de la forma en que transportan los datos por la red.

También veremos qué protocolos de un determinado conjunto participan en la capa de red del modelo OSI. Estos protocolos son de suma importancia ya que contribuyen a encaminar los paquetes en una conexión entre redes.

- **Protocolo Función**

FTP El File Transfer Protocol o Protocolo de Transferencia de Archivos proporciona una interfaz y servicios para la transferencia de archivos en la red.

SMTP El Simple Mail Transport Protocol o Protocolo Simple de Transferencia de Correo proporciona servicios de correo electrónico en las redes Internet e IP.

TCP El Transport Control Protocol o Protocolo de Control de Transporte es un protocolo de transporte orientado a la conexión. TCP gestiona la conexión entre las computadoras emisora y receptora de forma parecida al desarrollo de las llamadas telefónicas.

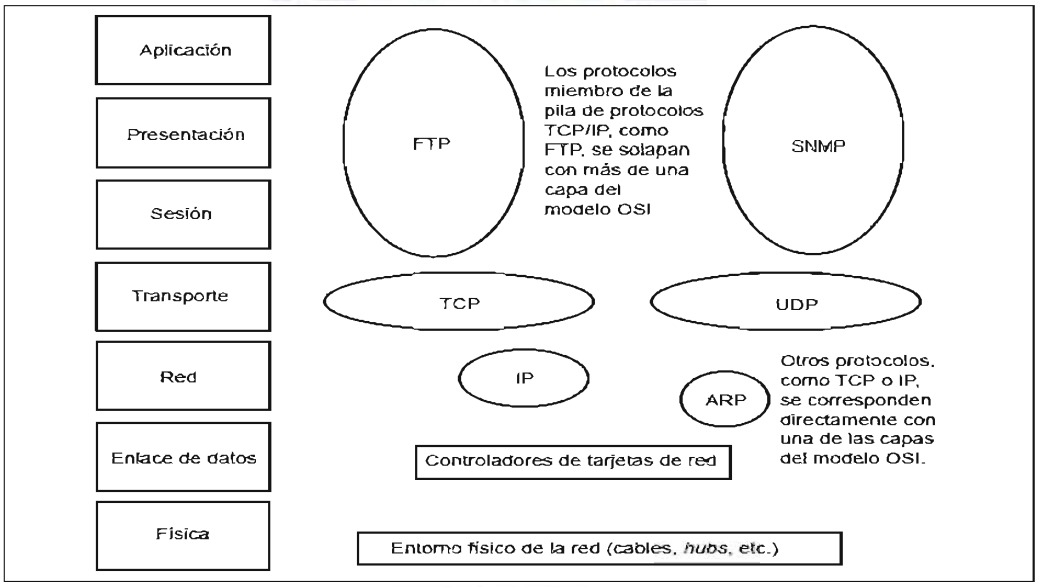
UDP El User Datagram Protocol o Protocolo de Datagrama de Usuario es un protocolo de transporte sin conexión que proporciona servicios en colaboración con TCP.

IP El Internet Protocol o Protocolo Internet es la base para todo el direccionamiento que se produce en las redes TCP/IP y proporciona un protocolo orientado a la capa de red sin conexión. Funciona de forma semejante a una carta con remite echada al buzón y después entregada a su destinatario.

ARP El Address Resolution Protocol o Protocolo de Resolución de Direcciones hace corresponder las direcciones IP con las direcciones MAC de hardware.

SNMP El Protocolo Simple de Administración de Red o SNMP es un protocolo de la capa de aplicación que facilita el intercambio de información de administración entre dispositivos de red.

Fig.2.3.1.1 serie de protocolos dentro del modelo osl



2.4 JAVA

La plataforma Java es el nombre de un entorno o plataforma de computación originaria de Sun Microsystems, capaz de ejecutar aplicaciones desarrolladas usando el Lenguaje de programación Java u otros lenguajes que compilen a bytecode y un conjunto de herramientas de desarrollo. En este caso, la plataforma no es un hardware específico o un sistema operativo, sino más bien una máquina virtual encargada de la ejecución de aplicaciones, y un conjunto de librerías estándar que ofrecen funcionalidad común.

La plataforma es así llamada la Plataforma Java (antes conocida como Plataforma Java e incluye:

- Plataforma Java, Edición Estándar (Java Platform, Standard Edition), o Java SE (antes J2SE)
- Plataforma Java, Edición Empresa (Java Platform, Enterprise Edition), o Java EE (antes J2EE)
- Plataforma Java, Edición Micro (Java Platform, Micro Edition), o Java ME (antes J2ME)

Desde 2006, la versión actual de la Plataforma Java Standard Edition se le conoce como Java SE 6 como versión externa, y 1.6 como versión interna. Sin embargo, se prefiere el término versión 6. Una visión general de la multitud de tecnologías que componen la Plataforma Java.

2.4.1 LIBRERIAS JAVA

En la mayoría de los sistemas operativos actuales, se ofrece una cantidad de código para simplificar la tarea de programación. Este código toma la forma, normalmente, de un conjunto de librerías dinámicas que las aplicaciones pueden llamar cuando lo necesiten. Pero la Plataforma Java está pensada para ser independiente del sistema operativo subyacente, por lo que las aplicaciones no pueden apoyarse en funciones dependientes de cada sistema en concreto. Lo que hace la Plataforma Java, es ofrecer un conjunto de libererías estándar, que contiene mucha de las funciones reutilizables disponibles en los sistemas operativos actuales.

Las librerías de Java tienen tres propósitos dentro de la Plataforma Java. Al igual que otras librerías estándar, ofrecen al programador un conjunto bien definido de funciones para realizar tareas comunes, como manejar listas de elementos u operar de forma sofisticada sobre cadenas de caracteres. Además, las librerías proporcionan una interfaz abstracta para tareas que son altamente dependientes del hardware de la plataforma destino y de su sistema operativo. Tareas tales como manejo de las funciones de red o acceso a ficheros, suelen depender fuertemente de la funcionalidad nativa de la plataforma destino. En el caso concreto anterior, las librerías java.net y java.io implementan el código nativo internamente, y ofrecen una interfaz estándar para que aplicaciones Java puedan ejecutar tales funciones. Finalmente, no todas las plataformas soportan todas las funciones que una aplicación Java

espera. En estos casos, las librerías bien pueden emular esas funciones usando lo que esté disponible, o bien ofrecer un mecanismo para comprobar si una funcionalidad concreta.

[Sun Microsystem., 2009]

2.5 PROTOCOLO ADMINISTRACIÓN DE RED (SNMP)

El Protocolo Simple de Administración de Red o SNMP es un protocolo de la capa de aplicación que facilita el intercambio de información de administración entre dispositivos de red. Es parte de la familia de protocolos TCP/IP. SNMP permite a los administradores supervisar el desempeño de la red, buscar y resolver sus problemas, y planear su crecimiento.

Las versiones de SNMP más utilizadas son dos: SNMP versión 1 (SNMPv1) y SNMP versión 2 (SNMPv2). Ambas versiones tienen un número de características en común, pero SNMPv2 ofrece mejoras, como por ejemplo, operaciones adicionales.

SNMP en su última versión (SNMPv2) posee cambios significativos con relación a sus predecesor ver Tabla 2.5.1 atributos de SNMP, sobre todo en aspectos de seguridad, sin embargo no ha sido mayoritariamente aceptado en la industria

Tabla 2.5.1 Tabla de atributos de SNMP

Simple Network Management Protocol (SNMP)	
Familia:	Familia de protocolos de Internet
Función:	facilita el intercambio de información de administración entre dispositivos de red
Última versión:	SNMP
Puertos:	161, 162

Ubicación en la pila de protocolos

Aplicación	SNMP

Transporte	UDP
Red	IP (IPv4 y IPv6)

2.5.1 COMPONENTES BÁSICOS

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

- Dispositivos administrados;
- Agentes;
- Sistemas administradores de red (NMS's).

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

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

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

2.5.2 COMANDOS BÁSICOS

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

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

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

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

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

2.5.3 BASE DE INFORMACIÓN DE ADMINISTRACIÓN SNMP (MIB)

Una Base de Información de Administración (MIB) es una colección de información que está organizada jerárquicamente. Las MIB's son accedidas usando un protocolo de administración de red, como por ejemplo, SNMP.

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

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

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

2.5.4 MENSAJES SNMP

Para realizar las operaciones básicas de administración anteriormente nombradas, el protocolo SNMP utiliza un servicio no orientado a la conexión para enviar un pequeño grupo

de mensajes entre los administradores y agentes. La utilización de un mecanismo de este tipo asegura que las tareas de administración de red no afectarán al rendimiento global de la misma, ya que se evita la utilización de mecanismos de control y recuperación como los de un servicio orientado a la conexión,

Tabla 2.5.4.1 Los puertos comúnmente utilizados para SNMP

Número	Descripción
161	SNMP
162	SNMP-trap

Los paquetes utilizados para enviar consultas y respuestas SNMP poseen el siguiente formato:

Tabla 2.5.4.2 Formato de consultas y respuestas.

Versión	Comunidad	SNMP

- Versión: Número de versión de protocolo que se está utilizando (por ejemplo 1 para SNMP);
- Comunidad: Nombre o palabra clave que se usa para la autenticación. Generalmente existe una comunidad de lectura llamada "public" y una comunidad de escritura llamada "private";
- SNMP Contenido de la unidad de datos del protocolo, el que depende de la operación que se ejecute.

Get

A través de este mensaje el NMS solicita al agente retornar el valor de un objeto de interés mediante su nombre. En respuesta el agente envía una respuesta indicando el éxito o fracaso de la petición. Si la petición fue correcta, el mensaje resultante también contendrá el valor del objeto solicitado. Este mensaje puede ser usado para recoger un valor de un objeto, o varios valores de varios objetos, a través del uso de listas.

GetNext

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

Set

Este tipo de mensaje es utilizado por el NMS para solicitar a un agente modificar valores de objetos. Para realizar esta operación el NMS envía al agente una lista de nombres de objetos con sus correspondientes valores.

Trap

Una trap es generado por el agente para reportar ciertas condiciones y cambios de estado a un proceso de administración. Una trap es un mensaje espontáneo enviado por el Agente al Administrador, al detectar una condición predeterminada, como es la conexión/desconexión de una estación o una alarma. El formato de la PDU es diferente:

GetBulk

Este mensaje es usado por un NMS que utiliza la versión 2 del protocolo SNMP típicamente cuando es requerida una larga transmisión de datos, tal como la recuperación de largas tablas. En este sentido es similar al mensaje GetNextusado en la versión 1 del protocolo, sin embargo, GetBulkes un mensaje que implica un método mucho más rápido y eficiente, ya que a través de un solo mensaje es posible solicitar la totalidad de la tabla.

Inform

Un NMS que utiliza la versión 2 del protocolo SNMP transmite un mensaje de este tipo a otro NMS con las mismas características, para notificar información sobre objetos administrados.

[Douglas E., 2000]

2.6 SOFTWARE LIBRE

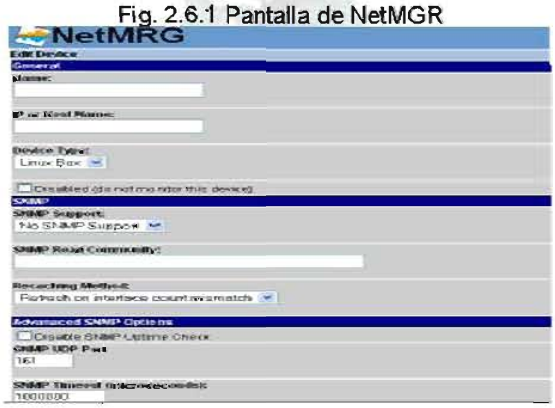
❖ NETMRG

NetMRG es una herramienta para la monitorización, y recogida de datos de forma gráfica Esta herramienta está basada en RRDTOOL (Round Robin Database) es un sistema que guarda y muestra series de datos por ejemplos el ancho de banda, la temperatura de una habitación de hosting, la carga del servidor, ver Fig. 2.6.1 Pantalla de NetMGR.
Por lo que NetMRG es capaz de crear cualquier tipo de gráfico a partir de cualquier dato sobre la red.

❖ Componentes

- **Gatherer**
  El Gatherer es un programa multihilo echo con C++ que interactúa entre la base de datos y el RRDTOOL.
- **Web Interface**
  La interfaz web consiste en una serie de scripts PHP y paginas HTML. Estas interactúan con la base de datos para guardar la configuración y para seleccionar que gráfica se tiene que mostrar.
- **Database**
  Es una base de datos relacional que funciona sobre MySQL y es utilizada por la interfaz web para mostrar los datos.
- **RRDtool**
  RRDtool es un sistema gráfico basado en round-robin databases (RRDs). Este se utiliza para guardar y crear los gráficos de la base de datos.

Fig. 2.6.1 Pantalla de NetMGR



Fuente [netmgr.com, 2009]

❖ **NAGIOS**

Nagios es un sistema de monitorización de equipos y de servicios de red, escrito en php , el lenguage con el cual esta desarrollado, ver interfaza dela aplicación en la Fig. 2.6.2 Pantalla de Zenoss.

❖ **Componentes**

- **Web Interface**
  La interfaz web consiste en una serie de scripts PHP y paginas HTML. Interfaz con que interactúa con la aplicación
- **Database**
  Es una base de datos relacional que funciona sobre MySQL y es utilizada por la interfaz web para mostrar los datos.
- **Monitorización**
  Monitorización de servicios de red (SNMP).

❖ **ZENOSS**

Este es un sistema de monitoreo multiplataforma pero en nuestro caso utilizaremos para la administracion remota de equipos, servidores y dispositivos de red.

❖ **Componentes**

- SNMP
- python
- MySql.

Fig. 2.6.2 Pantalla de Zenoss

Fuente [zenoss.com,2009]

❖ **ZABBIX**

Zabbix es un programa para monitorear los recursos de un equipo en forma remota que consume pocos recursos, permite centralizar la información en un servidor que monitoreo de multiples hosts, ver interfaz de la apicaion en Fig. 2.6.3 Pantalla de ZABBIX.

❖ **Componentes**

Los requisitos utilizar zabbix son:

• apache 1.3.12 o superior (trabaja con 2.x)
• mysql 3.22 en adelante o PostgreSQL 7 o mayor
• PHP4 o superior (necesita modulo GD para generar las gráficas) las librerías NETSNMP son necesarias para el server para instalarlo a partir del código fuente es necesario GCC

Fig. 2.6.3 Pantalla de ZABBIX

Fuente [zambbix.com,2009]

❖ **CACTIL**

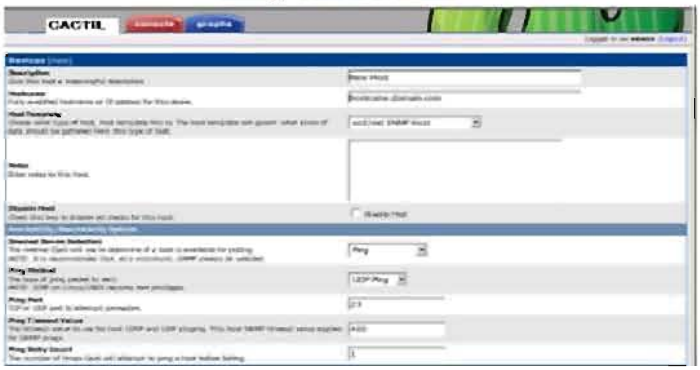
Programa para monitorear de servicios y eastado de los sistemas se realiza básicamente de dos formas.

Monitorizacion remota atravez de la red, atraves de la SNMP, ver interfaz de la aplicación en Fig. 2.6.4 Pantalla de Cactil

❖ **Componentes:**

• httpd
• php
• php-mysql
• php-snmp
• mysql
• mysql-server
• net-snmp

Fig. 2.6.4 Pantalla de Cactil



Fuente [cactil.com,2009]

2.7 CRIPTOGRAFIA

La Criptología

La Criptología (del griego **criptos=oculto** y **logos=tratado, ciencia**).

La Criptografía se ocupa del diseño de procedimientos para cifrar, es decir, para enmascarar una determinada información de carácter confidencial. El Criptoanálisis, por su parte, se ocupa de romper esos procedimientos de cifrado para así recuperar la información original. Ambas disciplinas siempre se han desarrollado de forma paralela, pues cualquier método de cifrado lleva siempre emparejado su Criptoanálisis correspondiente.

La Criptografía como medio para proteger la información personal es un arte tan antiguo como la propia escritura. Como tal, permaneció durante siglos vinculada muy estrechamente a los círculos militares y diplomáticos, puesto que eran los únicos que en principio tenían auténtica necesidad de ella. En la actualidad la situación ha cambiado drásticamente: el desarrollo de las comunicaciones electrónicas, unido al uso masivo y generalizado de los computadores, hace posible la transmisión y almacenamiento de grandes flujos de información confidencial que es necesario proteger.

Es entonces cuando la Criptografía pasa de ser una exigencia de minorías a convertirse en una necesidad real del hombre de la calle, que ve esta falta de protección de sus datos privados una amenaza para su propia intimidad.

A y B son, respectivamente, el emisor y receptor de un determinado mensaje. A transforma el mensaje original (texto claro o texto fuente), mediante un determinado procedimiento de cifrado controlado por una clave, en un mensaje cifrado (criptograma) que se envía por un canal público. En recepción, B con conocimiento de la clave transforma ese criptograma en el texto fuente, recuperando así la información original.

En el proceso de transmisión, el criptograma puede ser interceptado por un enemigo criptoanalista que lleva a cabo una labor de desencriptado; es decir, intenta, a partir del criptograma y sin conocimiento de la clave, recuperar el mensaje original. Un buen sistema criptográfico será, por tanto, aquel que ofrezca un descifrado sencillo pero un desencriptado imposible o, en su defecto, muy difícil.

La finalidad de la Criptografía es múltiple: primeramente, mantener la confidencialidad del mensaje; es decir, que la información allí contenida permanezca secreta; a continuación, garantizar la autenticidad tanto del criptograma (integridad) como del par remitente/destinatario. En efecto, el criptograma recibido da de ser realmente el enviado (evitando así manipulaciones o alteraciones en el proceso de transmisión), a la vez que el remitente y destinatario han de ser realmente quienes dicen ser, y no remitentes y/o destinatarios fraudulentos.

La Criptografía clásica se ocupaba únicamente del primer aspecto, mientras que la Criptografía de hoy en día, basada en el concepto de comunicaciones seguras, ha de garantizar conjuntamente todas ellas.

El tipo particular de transmisión aplicada al texto claro o las características de las claves utilizadas marcan la diferencia entre los diversos métodos criptográficos. Una primera clasificación en base a las claves utilizadas puede desglosarse tal y como sigue:

- **Métodos simétricos:** son aquellos en los que la clave de cifrado coincide con la de descifrado. Lógicamente, dicha clave tiene que permanecer secreta, lo que presupone que emisor y receptor se han puesto de acuerdo previamente en la determinación de la misma, o bien que existe un centro de distribución de claves que se la ha hecho llegar a ambos por un canal seguro.

- **Métodos asimétricos:** son aquellos en los que la clave de cifrado es diferente a la de descifrado. En general, la clave de cifrado es conocida libremente por el público, mientras que la de descifrado es conocida únicamente por el usuario.

Los métodos simétricos son propios de la Criptografía clásica o Criptografía de clave secreta, mientras que los métodos asimétricos corresponden a la Criptografía de clave pública, introducida por Difie y Hellman en 1976.

Una de las diferencias fundamentales entre la Criptografía clásica y la Criptografía de hoy en día radica en el concepto de seguridad. Antes, los procedimientos de cifrado tenían una seguridad probable; hoy, los procedimientos de cifrado han de tener una seguridad matemáticamente demostrable.

Puesto que las comunicaciones electrónicas se usan hoy en día para casi todas las actividades de interés social, están expuestas a todos los trucos y manipulaciones, consecuencia de las flaquezas humanas. Si en el mundo existieran honradez y confianza mutua, no habría necesidad de la Criptografía pero, a falta de aquéllas, la Criptografía trata de suplirlas con protocolos y algoritmos matemáticos de seguridad demostrable.

2.7.1 ALGORITMO RSA

Este sistema de clave pública fué diseñado en 1977 por los profesores del MIT (Massachusetts Institute of Technology) Ronald R. Rivest, Adi Shamir y Leonard M. Adleman, de ahí las siglas con las que es conocido. Desde entonces, este algoritmo de cifrado se ha convertido en el prototipo de los de clave pública.

La seguridad de RSA radica en la dificultad de la factorización de números grandes: es fácil saber si un número es primo, pero es extremadamente difícil obtener la factorización en números primos de un entero elevado, debido no a la dificultad de los algoritmos existentes, sino al consumo de recursos físicos (memoria, necesidades hardware...incluso tiempo de ejecución) de tales algoritmos.

Se ha demostrado que si n es el número de dígitos binarios de la entrada de cualquier algoritmo de factorización, el coste del algoritmo es con un tiempo de ejecución perteneciente a la categoría de los llamados problemas intratables.

2.7.1.1 GENERACIÓN DE CLAVES

Supongamos que **Alicia** desea permitir que **Benito** le mande un mensaje cifrado sobre un canal inseguro. Lo primero que ha de hacer es generar los pares de clave pública (n,e) y privada (n,d).

1. Elegimos dos números muy grandes p y q que sean diferentes y totalmente independientes el uno del otro. Calculamos $n = p * q$.

2. Calculamos la función de totient de n: $totient(n) = (p-1)*(q-1)$

3. Elegimos un entero e, tal que $1 < e < totient(n)$.

4. Calculamos un d, tal que $d * e = 1 \bmod totient(n)$.

5. La clave pública es (n,e) y la privada (n,d).

Alicia le transmite su clave pública a **Benito** y conserva la clave privada en secreto. Los valores p y q son muy sensibles, ya que son la descomposición en factores primos de n y los que dieron lugar a e y d. Generalmente destruidos, aunque pueden conservarse en secreto, junto con la clave privada.

- **Encriptación**

Ahora supongamos que **Benito** desea mandarle un mensaje a **Alicia**. Lo único que tendrá que hacer es consultar la clave pública de **Alicia**, dividir el mensaje que quiere enviarle, asignarle un alfabeto numérico a cada trozo y calcular para cada división: $c = ne \bmod n$.

- **Desencriptación**

Con el mensaje que le ha llegado a **Alicia**, lo que tiene que hacer es dividirlo y usar su clave privada para calcular: $n = cd \bmod n$.

[Horat-Cañizales,2006]

3 PROCESO DE INVESTIGACIÓN

3.1 DESCRIPCIÓN DE MÉTODOS

Para el fin del presente trabajo de tesis se hace uso de diferentes metodologías que proporcionan los pasos necesarios para concretar el proceso de investigación, siguiendo una lógica determinada y justificada en el desarrollo de cada uno de los puntos, que se llevara a cabo en el proceso de investigación ver Fig. 3.1.1.

Posteriormente se denota los métodos utilizados para la realización del presente trabajo de tesis.

Fig. 3.1.1 Proceso de investigación

3.2 MÉTODO CIENTÍFICO

El método científico es el procedimiento planteado que se sigue en la investigación para descubrir las propiedades del objeto de estudio, el proceso de razonamiento que intenta llegar a demostrarlos con racionalidad, para comprobarlos en el experimento y con las técnicas de su aplicación.

3.2.1 PLANTEAMIENTO DEL PROBLEMA

Para el planteamiento del problema se partió de la identificación de los una lista de problemas tabla 1.3.1 detalladas en el Capitulo 1, por medio de la matriz causa efecto Tabla 1.3.2 en el Capitulo 1, y se logro identificar y formular el siguiente problema principal:

¿La herramienta de gestión de dispositivos de red permitirá una buena y ágil gestión en los dispositivos de una red?

3.2.2 CONSTRUCCIÓN DEL MODELO TEÓRICO

El modelo teórico comprende las siguientes pasos

Selección de factores pertinentes al problema: Este paso constituye el marco teórico de del trabajo, donde se selecciona definiciones y la teoría necesarios para solucionar el problema planteado.

3.2.3 PLANTEAMIENTO DE LA HIPÓTESIS

"Hi: La herramienta para le gestión de dispositivos de una red ayudara a mejorar la administración de los dispositivos de una red LAN red"

La hipótesis muestra la interrelación de dos variables:

$$y = f(x)$$

Y= variable Dependiente= herramienta para le gestión de dispositivos

X=variable independiente = dispositivos de una red

Termino relacional =" Ayudara" que articula las variables

3.2.4 CONSTRUCCIÓN DE LA HERRAMIENTA

Se plantea la construcción de una herramienta capas de gestionar los dispositivos de una red.

La prueba del correcto funcionamiento de la herramienta de gestión de dispositivos de red se lo realizara mediante un prototipo (software).

Se redactan las conclusiones y recomendaciones finales de la investigación que incluye:

Comprobar las conclusiones contra las predicciones: Se realiza un informe acerca lo que se consigui con la tesis, si la hipótesis planteada a sido demostrada o no.

Sugerencia acerca de trabajos posteriores: Se menciona los problemas identificados y que pueden ser resueltos a futuro.

3.3 MÉTODOS QUE CONTEMPLA EL MÉTODO CIENTÍFICO

3.3.1 ANÁLISIS:

Es la descomposición de algo en su elementos. En este trabajo se descompone en temas de estudio en partes es decir que se estudian, investigan por separado, redes de computadoras, protocolos de red, el protocolo SNMP, métodos de criptografía.

3.3.2 SÍNTESIS:

Es la reconstrucción de todo lo descompuesto por el análisis. En este trabajo la síntesis se realiza cuando se construye el prototipo donde se unen todo los elementos analizados por separado.

3.3.3 INDUCCIÓN:

Es donde se parte de los datos particulares para llegar a la conclusión general. Se utiliza el método inductivo para la construcción del prototipo aplicando el método en espiral de la ingeniería de software.

3.3.4 CONCRECIÓN:

Se propone demostrar la hipótesis mediante experimentos. Se hace uso de la concreción cuando se llega a demostrar la hipótesis por medio de la experimentación.

3.4 METODOS PROPUESTOS

Para poder dar solución a los problemas encontrados, alcanzar los objetivos establecidos y demostrar la veracidad de la hipótesis el presente trabajo utilizara los siguientes métodos en la construcción del prototipo.

Métodos de ingeniería de software para la construcción del prototipo como es el modelo en Espiral, que acompaña la naturaleza interactiva de construcción de prototipos con los aspectos controlados y sistemáticos del modelo lineal secuencial.

Este modelo proporciona el potencial para el desarrollo rápido de versiones incrementales es asi que durante las ultimas iteraciones, se producen versiones cada ves mas completas.

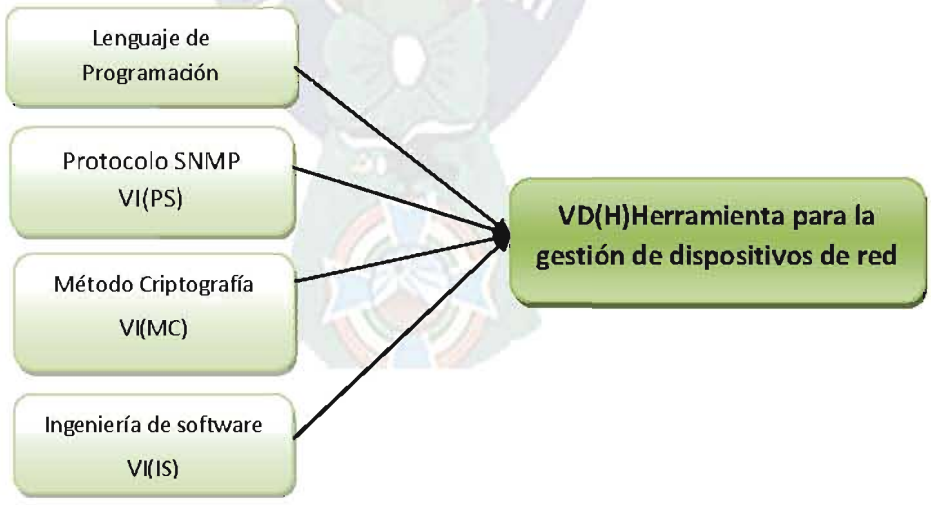
Métodos de Criptologia para el diseño de procedimientos para cifrar, es decir, para enmascarar una determinada información de carácter confidencial.

Específicamente la utilización del método asimétrico. El Método asimétrico en los que la clave de cifrado es diferente a la de descifrado. En general, la clave de cifrado es conocida libremente por el público, mientras que la de descifrado es conocida únicamente por el usuario.

3.5 DESCRIPCION FORMAL

La descripción formal intenta realizar una descripción delas principales variables que forman parte del del prototipo a ser realizado.

La herramienta de gestión de dispositivos de red estará compuesta principalmente de las siguientes **variables indepentientes :** la primera variables es el lenguaje de programación (JAVA)en que va a ser desarrollado la herraminta, la segunda variable el protocolo de Simple network managemend Protocol(SNMP),la tercera avariable método de criptologia y la cuarta variable es la ingeniería de Software que constituye la base para el desarrollo de la herramienta.



Es así donde se obtiene:

- VI(LP)= Variable Independiente LP
- VI(PS)= Variable Independiente PS
- VI(MC)= Variable Independiente MC
- VI(IS)= Variable Independiente IS
- VD(H)= Variable Dependiente VD

3.6 AMBIENTE DE DESARROLLO

A continuación se describe el hardware y software empleados para le desarrollo del sistema.

Hardware: se emplea una computadora personal integrada, Pentium IV, con un microprocesador intel de 1.8Ghz, de 512MB de memoria RAM.

Software: Se emplea el sistema operativo Windows XP de Microsoft, jdk1.6.0_05 de la sum microsystem además del IDE NetBeans 5.0

Es importante destacar, que el prototipo creado bajo el lenguaje de programación Java, también es capas de ejecutarse en plataforma Linux , debido a que el lenguaje de programación Java trabaja sin inconvenientes en la mencionada plataforma.

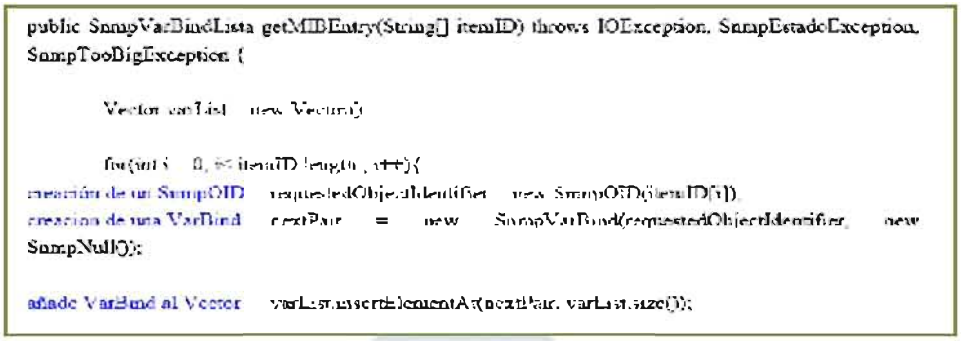
3.7 DESCRIPCIÓN DEL SISTEMA

3.7.1 CREACIÓN Y ENVÍO DE UN MENSAJE

Inicialmente se necesita tener los oids de las variables que se van a incluir en el mensaje. Si se trata de una operación set se necesitará también el nuevo valor con el tipo.

Con estos datos se van creando variables bindings y añadiéndolas a un contenedor, en este caso un Vector. Veámoslo con líneas de código.

Fig.3.6.1.1 Creación de variables bindings y añadir a un Vector



Una vez creadas las *variables bindings* en el vector, se debe crear la snmp correspondiente con el tipo de operación.

Se declara la pdu y se inicializan sus atributos con los valores adecuados.

Fig. 3.6.1.2 Crear una request PDU e inicializar sus valores

```
SnmpPDURequest pdu = new SnmpPDURequest();
pdu.tipoPDU =SnmpCtesComandos.pduGetRequestPdu;
pdu.requestID = requestID;
pdu.listaVarBind = new SnmpVarBind[varList.size()];
varList.copyInto(pdu.listaVarBind);
pdu.version = agente.getParametros().getVersionProtocolo();
pdu.comunidad = agente.getParametros().getReadComunidad().getBytes();
pdu.puerto = 161;
        pdu.version = agente.getParametros().getVersionProtocolo();
        pdu.comunidad = agente.getParametros().getReadComunidad().getBytes();
        pdu.puerto = 161;
```

El siguiente paso será crear el mensaje. Para esto se debe codificar la pdu (BER), codificar el mensaje, crear un *datagramPacket* con el mensaje codificado (BER) y por último, enviar dicho *datagramPacket* por el *dSocket* inicializado al iniciar la *SnmpSesion*.

Fig. 3.6.1.3 Crear un mensaje codificarle y envio

```
SnmpMensaje mensaje = new SnmpMensaje();
mensaje.codificaPDU(pdu, 1024); codificamos la pdu y queda incluida en el mensaje
byte[] messageEncoding = new byte[2000];
int longCodificada = mensaje.codificarMensaje(messageEncoding);
DatagramPacket outPacket = new DatagramPacket (messageEncoding,
messageEncoding.length, agente.getdirInternet(), agente.getPuerto());
dSocket.send(outPacket);
```

3.7.2 RECEPCIÓN DE UNA RESPUESTA A UN MENSAJE Y SU DECODIFICACIÓN

Para recibir el mensaje se debe preparar un *datagramPacket* para albergar el paquete respuesta. Una vez creado éste, se debe invocar al método *received* del *datagramSocket* instanciado en *SnmpSesion,* indicando que se debe recibir en el *datagramPacket* creado anteriormente.

Fig. 3.6.2.1 Como recibir la respuesta.

```
DatagramPacket inPacket = new DatagramPacket(new byte[receiveBufferSize], receiveBufferSize);

dSocket.receive(inPacket);
```

Una vez recibida la respuesta, debemos conseguir los datos y decodificarlos (llegan codificados según las reglas BER) hasta llegar a obtener la PDU decodificada.

Fig. 3.6.2.1 Como decodificar un mensaje y la pdu contenida

```
byte[] encodedMessage = inPacket.getData();// los datos se pasan a un array de bytes

SnmpMensaje receivedMessage = new SnmpMensaje();
receivedMessage.decodificarMensaje(encodedMessage); //decodificar el mensaje

SnmpPDURequest receivedPDU = new SnmpPDURequest();
receivedPDU = (SnmpPDURequest)receivedMessage.decodificarPDU(); //decodificar pdu
```

3.7.3 TRATAMIENTO DE UNA RESPONSEPDU DECODIFICADA

Una vez decodificada la pdu, falta comprobar que es la respuesta a la petición, mediante el *requestID.* Si es la respuesta adecuada, comprobar que no se ha producido ningún error mediante el *errorStatus*. Y por último actuar consecuentemente tanto si se ha producido algún error como si todo es correcto.

Fig.3.6.3.1 Comprobar el RequestID y errorStatus

```
if (receivedPDU.requestID == requestID) { //Comprueba el requestID

        if (receivedPDU.getErrorStatus() == 0){ //Comprueba el errorStatus
        .....
        }
        else{
        .....
        }
}
```

Si todo ha sido correcto, se añade en una *SnmpVarBindLista* las variables *bindings* incluidas en la lista de variables de la respuesta.

Fig. 3.6.3.2 Añadir variables bindings a una lista de variables

```
if (receivedPDU.requestID == requestID) {

        if (receivedPDU.getErrorStatus() == 0){
        for(int i = 0 ; i<receivedPDU.listaVarBind.length; i++){
                if (receivedPDU.listaVarBind[i].getOid().toString().equals(itemID[i])){
                        retrievedVars.añadirVarBind(receivedPDU.listaVarBind[i]);
        }//añadir variables a la lista que se va a retornar a la aplicación
}
```

Por ultimo se retorna la lista de variables bindings ó se lanza una excepción indicando el error producido. Estos errores podrán ser diferentes según el tipo de operación solicitada.

Fig. 3.6.3.3 Ejemplo de lanzamiento de excepciones en SnmpSesion.

```
                        if(receivedPDU.getErrorStatus() == 1)
        throw new SnmpEstadoException("TooBigException: ResponsePDU demasiado
                                        grande");
else
        throw new SnmpEstadoException ("genError: Error en la variable" +
                receivedPDU.getErrorIndex());if (receivedPDU.requestID == requestID)
```

Fig. 3.6.3.4 Retornar las variables respuesta

```
requestID++;
return retrievedVars;
```

3.7.4 CÓMO SOLICITAR TODOS LOS VALORES DE LA MIB

3.7.4.1 SOLICITAR TODA LA TABLA MIB

Para solicitar todo el contenido de una tabla MIB se deben realizan las siguientes acciones:

Crear un array de String con el valor 1.0, que representa la raíz del árbol de la MIB.

Fig. 3.6.4.1.1 Raíz de la MIB

```
String[] oids= {"1.0"};
```

Mientras el valor de flag sea True (mientras no se alcance el final de la MIB), se llama al método getBulkMIBEntry con un valor de 0 en la variable Non-repeaters y un valor de 150 en

la de Max-repetitions. De esta forma se irán obteniendo en cada mensaje recibido 150 variables binding.

Fig. 3.6.4.1.2 Llamada a getBulkMIBEntry

```
while(flag){
    try{
        SnmpVarBindLista respuestaParcial = getBulkMIBEntry(0,150, oids);
        for(int i = 0; i<respuestaParcial.size(); i++){
            if(respuestaParcial.getVarBindAt(i).getValor().esEndOfMibView()){
              flag = false;
              break;
            }
```

Conforme van llegando respuestas del agente, se introducen las variables binding en una variable SnmpVarBindLista. Además, si no se ha alcanzado el final de la MIB, se introduce en el array de String creado en la figura 10, el OID de la última variable binding recibida. Con ello se consigue que la próxima respuesta contenga las 150 variables siguientes.

Fig. 3.6.4.1.3 Tratamiento de las respuestas parciales

```
            varBindRespuesta.añadirVarBind(respuestaParcial.getVarBindAt(i));
        }
        if(flag)
            oids[0]= respuestaParcial.getVarBindAt(respuestaParcial.size()-1).getOid().toString();
    }
```

Por último se retorna la SnmpVarBindLista a la clase linkIDElibv2 con todas las variables y se imprime de la misma forma que en el punto anterior.

Fig. 3.6.4.1.4 Retornar las variables binding

```
return varBindRespuesta;
```

3.7.4.2 SOLICITAR TABLA DE LA MIB

Para solicitar aquellas variables cuyo prefijo de OID coincida con el OID especificado por el usuario, se realizan los mismos pasos que para visualizar toda la MIB. Únicamente existen dos diferencias que se explican a continuación:

Al crear el array de String, se introduce la base de OID que especifique el usuario.

Fig. 3.6.4.2.1 Prefijo de OID

```
String[] oids= {baseID};
```

Cuando se llama al método getBulkMIBEntry, el valor de la variable Maxrepetitions se reduce a 50, debido a que el número de variables binding solicitadas por el usuario son menores que al solicitar toda la MIB.

3.7.5 ESCUCHAR TRAPS E INFORMS

Para poder escuchar los traps e informs que otros dispositivos nos envían, se ha implementado una clase escuchaTraps, Antes de comenzar a recibir notificaciones, hay que crear un hilo por el cual poder escucharlas. Una vez creado el hilo, la clase escuchaTraps permanece a la escucha de estas notificaciones mediante el método run. Al recibir una de ellas se siguen los siguientes pasos:

Se crea un DatagramPacket y se introduce el mensaje en él mediante el método receive de la clase dSocket.

Fig. 3.6.5.1 Recepción del mensaje

```
DatagramPacket paqueteEntrada = new DatagramPacket(new byte[tamanioBufferRecepcion],
                                tamanioBufferRecepcion);
dSocket.receive(paqueteEntrada);
```

Se decodifican tanto el mensaje como la PDU contenida en él, y se comprueba si la PDU es de tipo trapV1, trapV2 ó inform. En caso negativo se lanza una excepción.

Fig. 3.6.5.2 Decodificación del mensaje y la PDU

```
byte[] mensajeCodificado = paqueteEntrada.getData();
SnmpMensaje mensajeRecibido = new SnmpMensaje();
mensajeRecibido.decodificarMensaje(mensajeCodificado);

SnmpPDU pduRecibida = new SnmpPDU();
pduRecibida = (SnmpPDU)mensajeRecibido.decodificarPDU();
if ((pduRecibida.tipoPDU != pduRecibida.pduV1TrapPdu && pduRecibida.tipoPDU
 != pduRecibida.pduV2TrapPdu && pduRecibida.tipoPDU != pduRecibida.pduInformRequestPdu)) {

        throw new SnmpEstadoException("No es un trap o inform de v1 ni v2. El mensaje es de
        tipo: " + pduRecibida.getClass().toString());
}
```

Si la PDU es de alguno de los tres tipos vistos en el punto anterior, se trata, sacando la información que contenga. Después se crea un archivo .txt en el disco duro para

almacenarla, y si es de tipo inform, se vuelve a enviar el mensaje, pero esta vez, mediante una responsePDU.

Fig. 3.6.5.3 Tratamiento de las notificaciones

```
else{
        switch(pduRecibida.tipoPDU){
                case 164:   pduTrap =(SnmpPDUTrap)pduRecibida;
                            dirAgente = pduTrap.agentAddr;
                            oid = pduTrap.enterprise;
                            trapGenerico = pduTrap.genericTrap;
                            archivo=new FileOutputStream(".\\logs\\traps.txt", true);
                            break;

                case 167:   pduTrapv2 =(SnmpPDURequest)pduRecibida;
                            dirAgentev2 = pduTrapv2.direccion;
                            oid = pduTrapv2.listaVarBind[1].getOid();
                            archivo=new FileOutputStream(".\\logs\\traps.txt", true);
                            break;

                default:pduTrapv2 =(SnmpPDURequest)pduRecibida;
                            dirAgentev2 = pduTrapv2.direccion;
                            oid = pduTrapv2.listaVarBind[1].getOid();
                            paqueteEntrada.setAddress(dirOriRecibido);
                            dSocket.send(paqueteEntrada);
                            archivo=new FileOutputStream(".\\logs\\inform.txt", true);
                            break;
    }
```

Una vez decodificada ya podemos extraer la información necesaria o que queramos mostrar.

3.7.6 ECRIPTACION

El sistema criptográfico con clave pública RSA es un algoritmo asimétrico que utiliza una clave pública, la cual se distribuye, y otra privada, la cual es guardada en secreto por su propietario.

Los mensajes enviados usando el algoritmo RSA se representan mediante números y el funcionamiento se basa en el producto de dos números primos grandes (mayores que 10^{100}) elegidos al azar para conformar la clave de descifrado.

Emplea expresiones exponenciales en aritmética modular.

La seguridad de este algoritmo radica en que no hay maneras rápidas conocidas de factorizar un número grande en sus factores primos utilizando computadoras tradicionales.

3.7.6.1 GENERACIÓN DE CLAVES

1. Elegimos dos números muy grandes p y q que sean diferentes y totalmente independientes el uno del otro. Calculamos n = p * q.
2. Calculamos la función de totient de n: totient(n) = (p-1)*(q-1)
3. Elegimos un entero e, tal que 1 < e < totient(n) y que ademas sea coprimo con totient(n).
4. Calculamos un d, tal que d * e = 1 mod totient(n).
5. La clave pública es (n,e) y la privada (n,d).

3.7.6.2 DEMOSTRACION

p=61	1º nº primo Privado
q=53	2º nº primo Privado
n=pq=3233	producto p*q
e=17	exponente Público
d=2753	exponente Privado

La clave pública (e, n). La clave privada es d. La función de cifrado es:

$$encrypt(m) = m^e (mod\ n) = m^{17} (mod\ 3233)$$

Donde m es el texto sin cifrar La función de descifrado es:

$$decrypt(c) = c^d (mod\ n) = c^{2753} (mod\ 3233)$$

Donde c es el texto cifrado. Para cifrar el valor del texto sin cifrar 123, nosotros calculamos:

$$encrypt(123) = 123^{17} (mod\ 3233) = 855$$

Para descifrar el valor del texto cifrado, nosotros calculamos

$$decrypt(855) = 855^{2753} (mod\ 3233) = 123$$

Fig. 3.6.6.2.1 Declaracion de variables

```
public class RSA
{
  int tamPrimo;        // Tamaño de cada primo
  BigInteger n, p, q ;     // n = p * q
  BigInteger totient;        // totient = (p-1)*(q-1)
  BigInteger e, d;     // e * d = 1 mod n
```

Fig. 3.6.6.2.2 métodos de generPrimos, generaclave

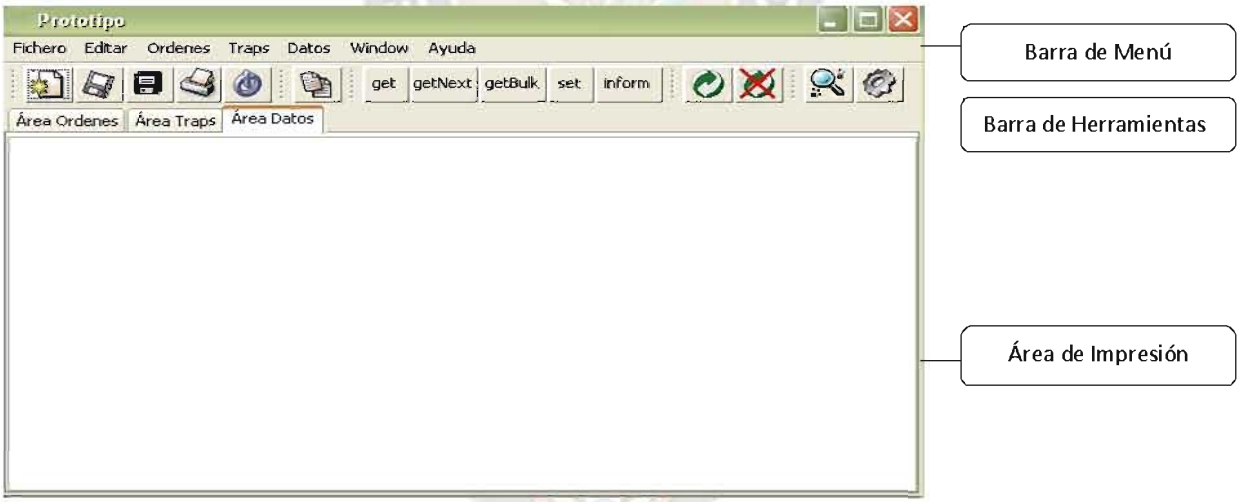
```
public void generaPrimos() {
  p = new BigInteger(tamPrimo, 10, new Random());
  do q = new BigInteger(tamPrimo, 10, new Random()) ;
    while(q.compareTo(p) == 0);
}

public void generaClaves() {
  // n = p * q
  n = p.multiply(q);
  // totient = (p-1)*(q-1)
  totient = p.subtract(BigInteger.valueOf(1));
  totient = totient.multiply(q.subtract(BigInteger.valueOf(1)));
  // Elegimos un e coprimo de y menor que n
  do e = new BigInteger(2 * tamPrimo, new Random());
    while((e.compareTo(totient) != -1 ) ||
(e.gcd(totient).compareTo(BigInteger.valueOf(1)) != 0));
  // d = e^1 mod totient
  d = e.modInverse(totient);}
```

4 DISCUSIÓN Y CONCLUCIONES

4.1 PRESENTACIÓN DELA HERRAMIENTA

Se dará una visión general del prototipo

INTERFAS INICIAL



BARRA DE MENÚ

En esta barra se encuentran todas las funciones del prototipo. Dispone de siete títulos desplegables:

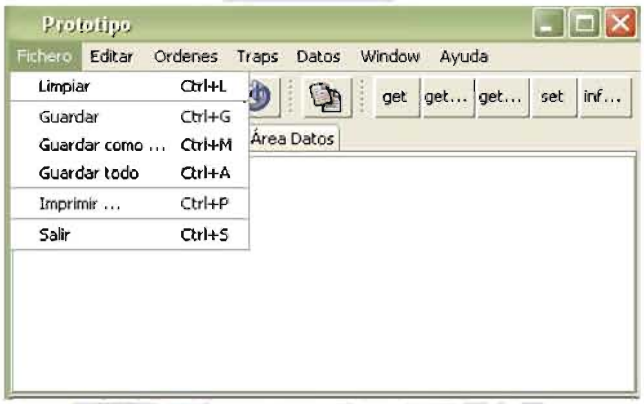
- Fichero
- Editar
- Órdenes
- Traps

• Datos
• Window
• Ayuda.

A continuación se pasará a explicar cada una de las distintas funcionalidades que aparecen dentro de estos menús y entre paréntesis las teclas de acceso rápido creadas para cada una de ellas ver Figura.4.1.1.

FICHERO:

Fig.4.1.1 opción Fichero



Limpiar: Si se pulsa esta opción, se borrará el contenido del área de texto que se tenga activa en ese momento. Si se ha producido alguna modificación, antes de borrar, notificará al usuario, que si desea continuar se perderán los datos no almacenados.(Ctrl + L)

Guardar: Permite guardar los datos del área de texto activa es ese momento. Si se pulsa y no se ha guardado anteriormente aparecerá una ventana donde se podrá elegir la ubicación del nuevo fichero. En caso contrario lo guardará en la ruta especificada con anterioridad. Señalar que si no ha habido ninguna modificación en el área de texto activa, al pulsar esta opción no sucederá nada.(Ctrl + G)

Guardar como…: Esta opción es semejante a la anterior, con la salvedad de que siempre preguntará un nombre y una ubicación para el nuevo documento. Si se escoge un nombre ya utilizado, informará de este hecho y dará la posibilidad de escoger otro ó continuar.(Ctrl + M)

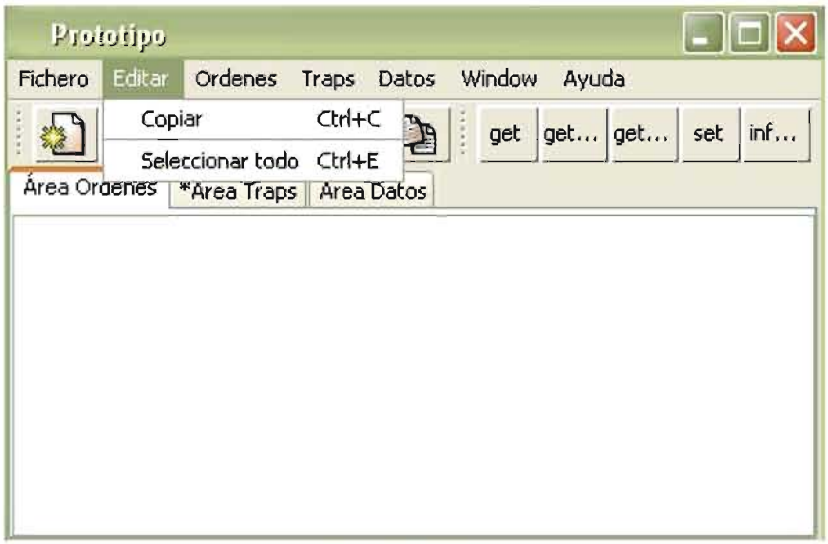
Guardar todo: Permite guardar el contenido de las áreas de texto que hayan sufrido alguna modificación. De este modo si únicamente el área de órdenes y la de datos han cambiado, solo
preguntará por ellas y no por el área de traps.(Ctrl + A)

Imprimir: Esta opción sirve para imprimir el área de texto activa en ese momento. Si no hay conectada ninguna impresora, aparecerá un mensaje de error informando de ello.(Ctrl + P).

Salir: Con esta opción se sale de la aplicación. Antes de ello, si existe algún área de texto sin guardar, informa al usuario de este hecho y le permite guardar la información, no guardarla, o bien cancelar la acción de salir de la aplicación.(Ctrl + S)

EDITAR:
En este menú se encuentran dos opciones para poder editar la información del área de texto ver Figura.4.1.2

Fig.4.1.2 Opción Editar



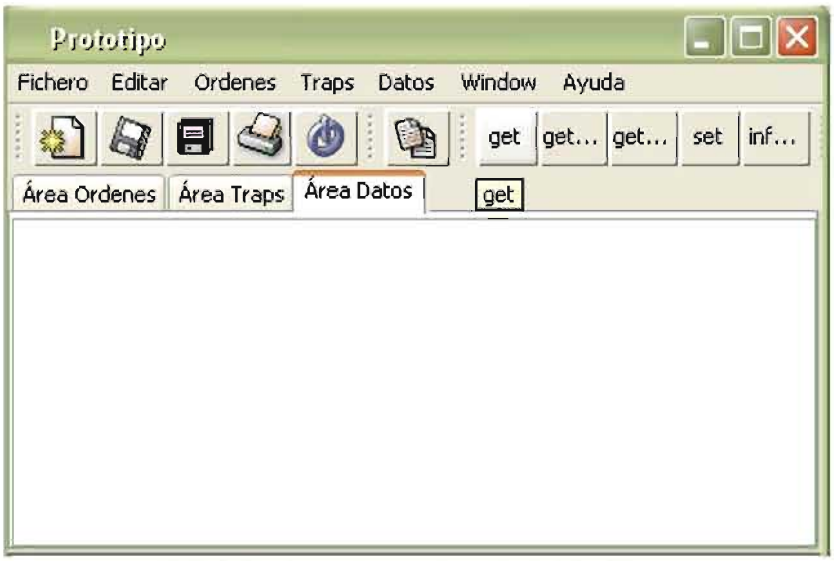
Copiar: Esta opción permite copiar la información que se deseé de cualquier área de texto para luego pegarlo en algún otro documento.(Ctrl + C)

Seleccionar todo: Selecciona todo el contenido del área de texto activa en ese momento, para poder pegarlo donde el usuario deseé.(Ctrl + E)

ÓRDENES:
traps y datos: Estos tres menús se explicarán más adelante en el apartado "Operaciones de gestión".

WINDOW:

Este menú permite moverse entre las tres áreas de texto disponibles en la aplicación



Dentro de este menú se pueden encontrar tres opciones:

- Área de Órdenes (Ctrl + O)
- Área de Traps (Ctrl + T)
- Área de Datos (Ctrl + D)

Cada una de ellas activa el área de texto que lleva su nombre. La explicación de cada una de estas áreas aparece más adelante bajo el título "Área de impresión".

AYUDA:

Contiene la opción de visionar el "Manual de Usuario" y el logotipo de la aplicación ver Figura.4.1.3.

Fig.4.1.3 Opción Ayuda

Contenido: Con es

About: Permite visionar el logotipo de la aplicación.

4.1.1 OPERACIONES DE GESTIÓN DE RED

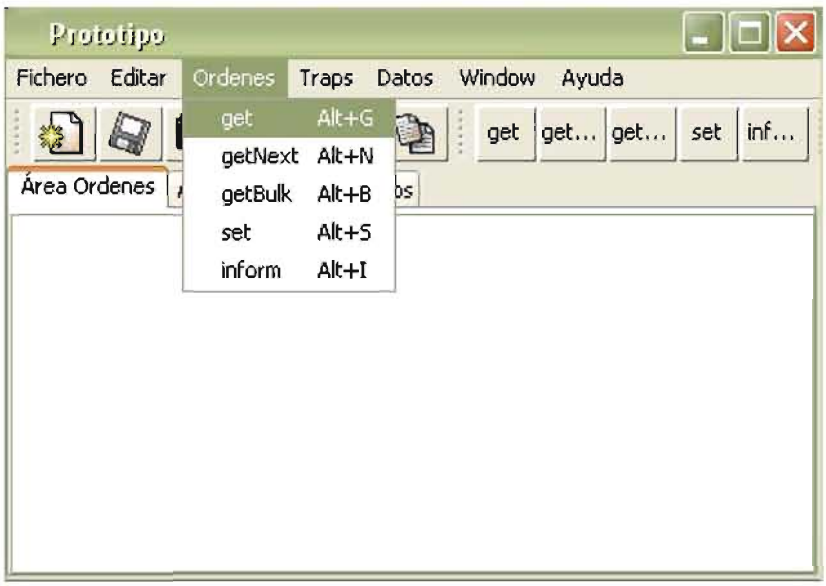
En este punto se van a explicar cómo realizar las operaciones de gestión de red.

OPERACIÓN GET

En esta operación el dispositivo gestor, solicita al dispositivo agente, el valor de una ó varias variables de la MIB. Más información en Operaciones Snmp→Get de la Memoria. Para ello se deben seguir los siguientes pasos:

Primero se debe lanzar el formulario de la operación get. Esto se puede hacer mediante el menú Órdenes→get, con las teclas de acceso rápido Alt + G ó bien con el icono get de la Barra de Herramientas.

Fig. 4.1.1.1 Operación Get

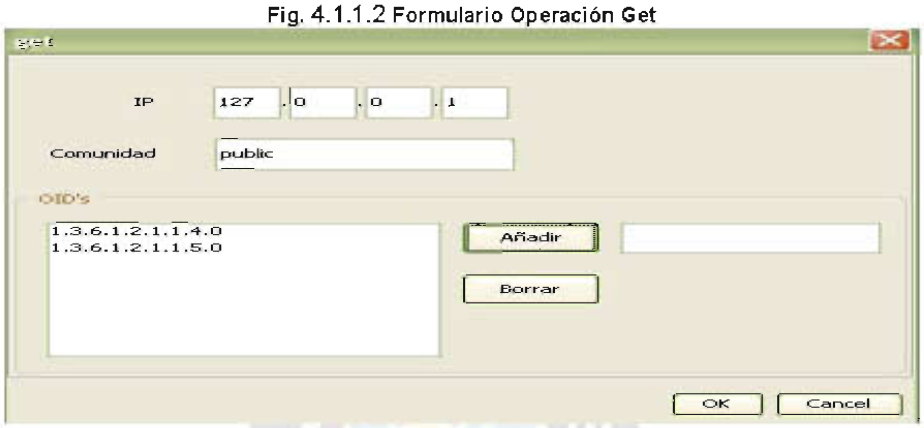


Aparecerá el formulario de esta operación en el cual se deben rellenar los campos IP con el dispositivo sobre el que se quiere actuar, la Comunidad que será public y el OID de la/las variable/s que se desean monitorizar.

Señalar que si en cualquiera de las celdas del campo IP, se introduce una secuencia de dígitos que no se corresponde con una dirección de red, la aplicación no permitirá al usuario seguir a la siguiente celda. Este proceso de control se realiza en todos los formularios.

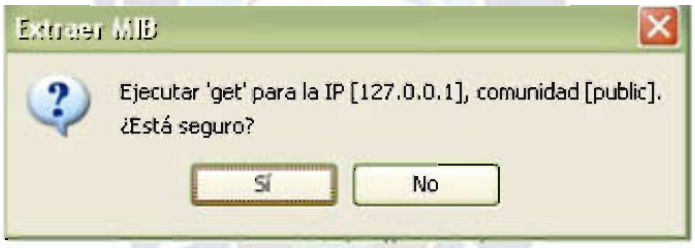
Este ejemplo se realizará sobre la IP 127.0.0.1 y con las variables 1.3.6.1.2.1.1.4.0 y 1.3.6.1.2.1.1.5.0

Fig. 4.1.1.2 Formulario Operación Get



Una vez rellenados los campos, se pulsa la tecla OK y aparecerá un mensaje con los datos introducidos preguntando si se desea continuar.

Fig. 4.1.1.3 Comprobación de datos operación Get



En caso afirmativo se ejecutará la operación y la respuesta aparecerá reflejada en el área de texto "Área Órdenes". Si se pulsa NO, se volverá al formulario para poder realizar los cambios requeridos por el usuario.

Fig. 4.1.1.4 Resultado Operación Get

OPERACIÓN GETNEXT

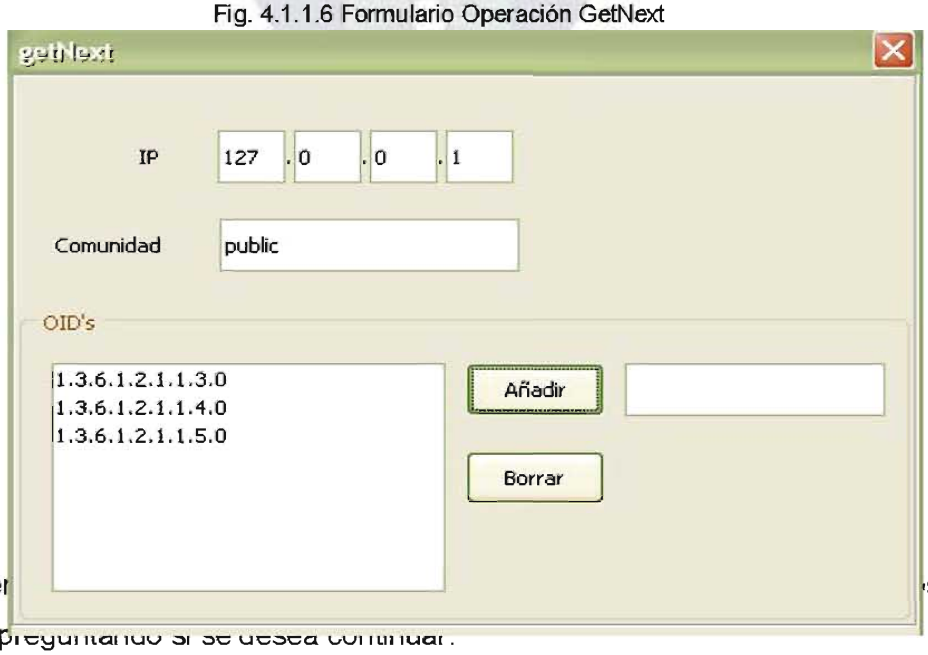
En esta operación el dispositivo gestor, solicita al dispositivo agente, el valor de las siguientes variables MIB a las especificadas. Más información en Operaciones Snmp→GetNext de la Memoria.Los pasos a seguir son los siguientes:

Primero se debe lanzar el formulario de la operación geNext. Esto se puede hacer mediante el menú Órdenes→geNext, con las teclas de acceso rápido Alt + N ó bien con el icono getNext de la "Barra de Herramientas".

Fig. 4.1.1.5 Operación GetNext

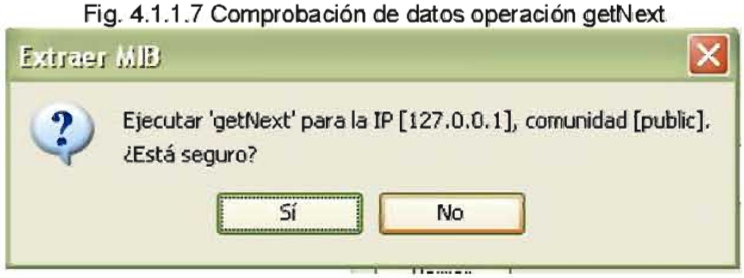
Aparecerá el formulario de esta operación en el cual se deben rellenar los campos IP con el dispositivo sobre el que se quiere actuar, la Comunidad que será public y el OID de la/las variable/s que se desean monitorizar. Este ejemplo se realizará sobre la IP 127.0.0.1 y con las variables 1.3.6.1.2.1.1.3.0 1.3.6.1.2.1.1.4.0 y 1.3.6.1.2.1.1.5.0

Fig. 4.1.1.6 Formulario Operación GetNext



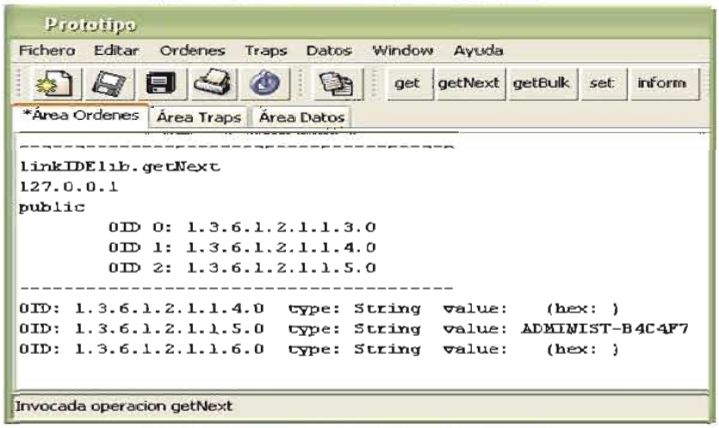
Una vez reller s datos introducidos preguntando si se desea continuar.

Fig. 4.1.1.7 Comprobación de datos operación getNext



En caso afirmativo se ejecutará la operación y la respuesta aparecerá reflejada en el área de texto "Área Órdenes". Si se pulsa NO, se volverá al formulario para poder realizar los cambios requeridos por el usuario.

Fig. 4.1.1.8.Resultado Operación GetNext



OPERACIÓN SET

En esta operación el dispositivo gestor intenta cambiar uno o varios valores de las variables MIB del dispositivo agente. A diferencia de las operaciones anteriores, se debe una comunidad con permisos de lectura/escritura. Para este ejemplo se ha creado comunidad private. Más información en Operaciones Snmp→Set de la Memoria. pasos a seguir son los siguientes:

Primero se debe lanzar el formulario de la operación set. Esto se puede hacer mediante el menú Órdenes→Set, con las teclas de acceso rápido Alt + S ó bien con el icono set de la "Barra de Herramientas".

Fig. 4.1.1.9 Operación Set

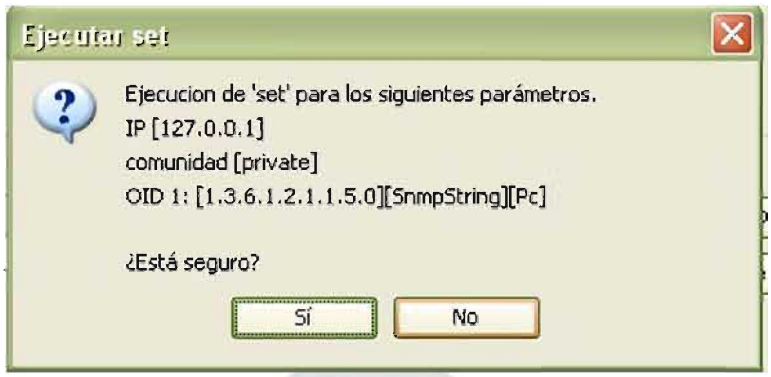
Aparecerá el formulario de esta operación en el cual se deben rellenar los campos IP con el dispositivo sobre el que se quiere actuar, la Comunidad que será private y al menos una de las filas de la tabla OIDs. En esta tabla aparecen los campos OID de las variables a las que se quiere cambiar el valor, tipo de la variable y valor nuevo que se desea introducir. Para este ejemplo se realizará sobre el dispositivo 127.0.0.1 con la variable de oid 1.3.6.1.2.1.1.5.0 que es de tipo String y con nuevo valor PC.

Fig. 4.1.1.10 Formulario Operación Set

En caso de que el usuario llene la tabla, puede agregar nueva filas mediante el botón Nueva OID ó también eliminar alguna de ellas con el botón Borrar selección

Una vez rellenados los campos, se pulsa la tecla OK y aparecerá un mensaje con los datos introducidos preguntando si se desea continuar.

Fig. 4.1.1.11 Comprobación de datos operación Set



En caso afirmativo se ejecutará la operación y la respuesta aparecerá reflejada en el área de texto "Área Órdenes". Si se pulsa NO, se volverá al formulario para poder realizar los cambios requeridos por el usuario.

OPERACIÓN GETBULK

Esta operación no puede ser realizada sobre un dispositivo agente que únicamente posea la versión 1 del protocolo. Si se produce esta situación, aparecerá un mensaje de error en el Área de Órdenes. Se utiliza para volcados masivos de la MIB. Para más información ver el apartado Operaciones Snmp→GetBulk de la Memoria. Los pasos a seguir son:

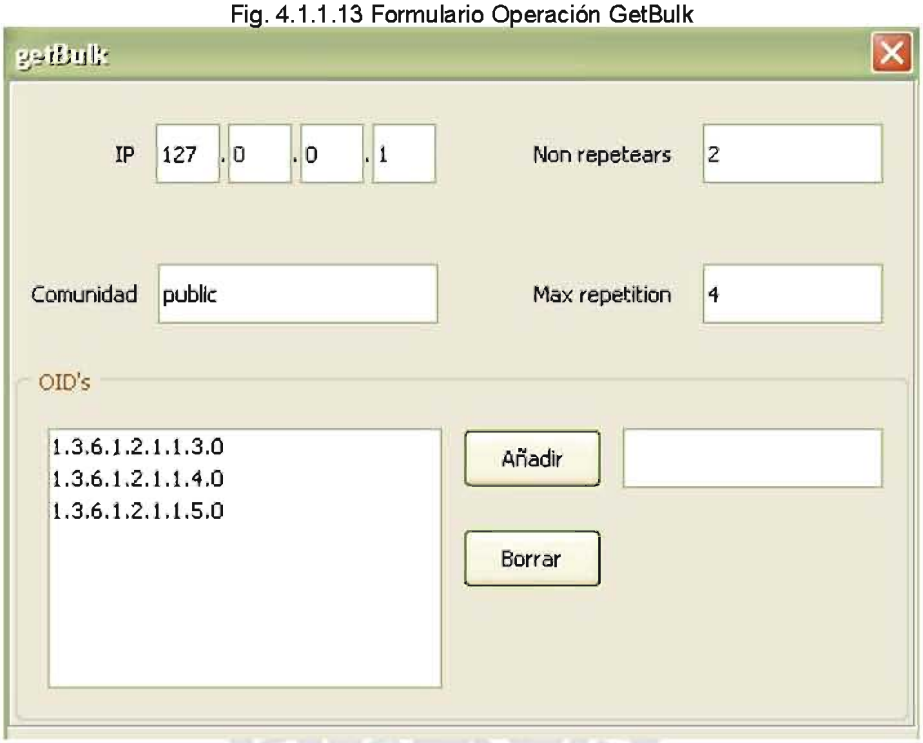
Lanzar el formulario de la operación getBulk. Esto se puede hacer mediante el menú Órdenes→getBulk, con las teclas de acceso rápido Alt + B ó bien con el icono getBulk de la "Barra de Herramientas".

Fig. 4.1.1.12 Operación GetBulk

Aparecerá el formulario de esta operación, en el cual se deben rellenar los campos IP con el dispositivo sobre el que se quiere actuar, la Comunidad que será public, los campos Non-

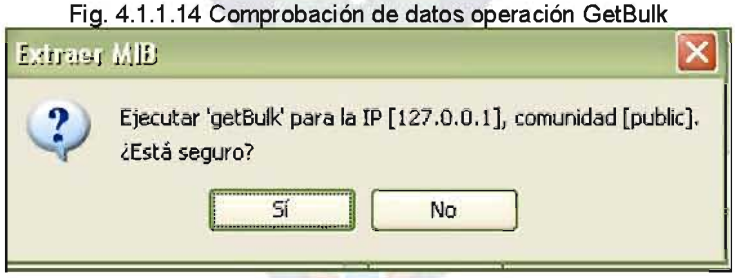
repeaters y Max-repetitions y el OID de la/las variable/s que se desean monitorizar. Este ejemplo se realizará sobre la IP 127.0.0.1 con las variables 1.3.6.1.2.1.1.3.0 1.3.6.1.2.1.1.4.0 y 1.3.6.1.2.1.1.5.0 y con los campos Non-repeaters a 2 y Maxrepetitions a 4.

Fig. 4.1.1.13 Formulario Operación GetBulk



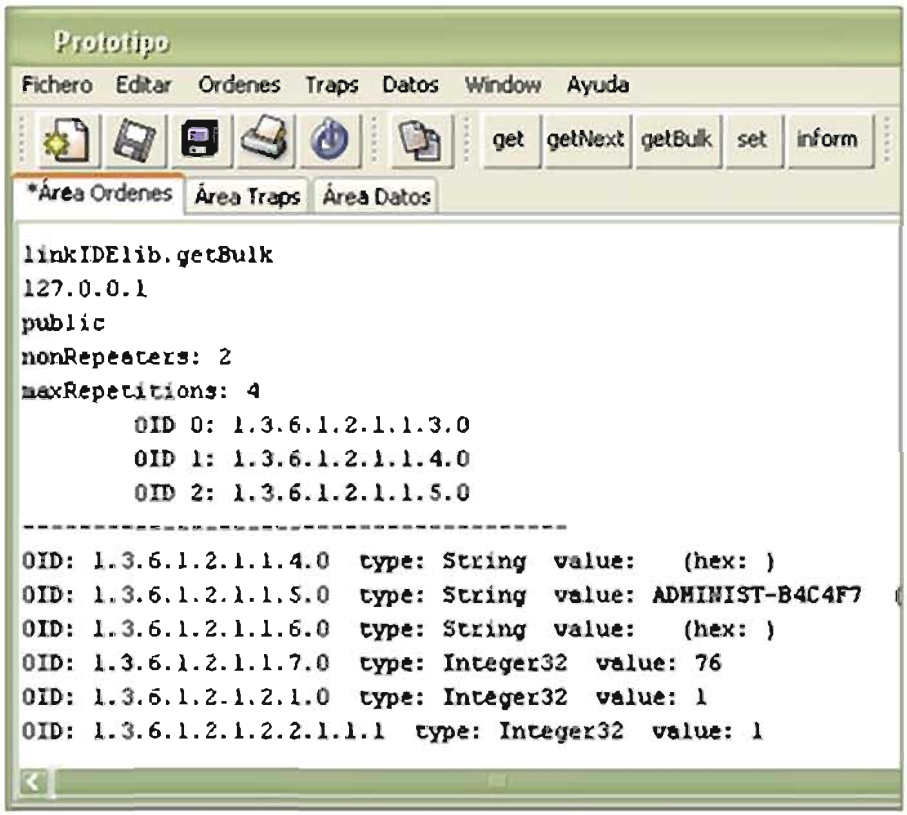
Una vez rellenados los campos, se pulsa la tecla OK y aparecerá un mensaje con los datos introducidos preguntando si se desea continuar.

Fig. 4.1.1.14 Comprobación de datos operación GetBulk



En caso afirmativo se ejecutará la operación y la respuesta aparecerá reflejada en el área de texto "Área Órdenes". Si se pulsa NO, se volverá al formulario para poder realizar los cambios requeridos por el usuario.

Fig. 4.1.1.15 Resultado Operación GetBulk



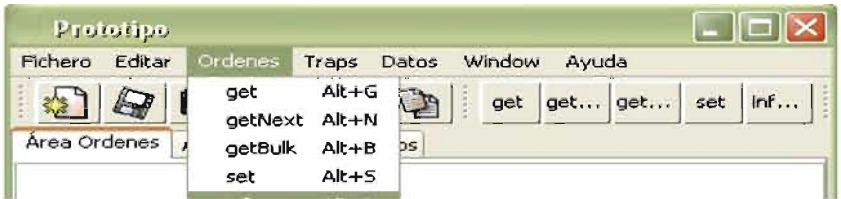
OPERACIÓN INFORM

Esta operación la ejecuta un dispositivo gestor hacia otro gestor para comunicarle alguna notificación. A diferencia de los traps, sí tienen confirmación de llegada, ya que el gestor que ha recibido el mensaje lo devuelve al dispositivo que se lo ha mandado. Esta confirmación de llegada aparece en el Área de Órdenes. Al igual que ocurría con getBulk, esta operación no puede ser ejecutada con la versión 1 del protocolo, apareciendo un mensaje de error si se produjera esta situación. Más información en Operaciones Snmp→Inform de la Memoria

Los pasos para poder mandar un inform son:

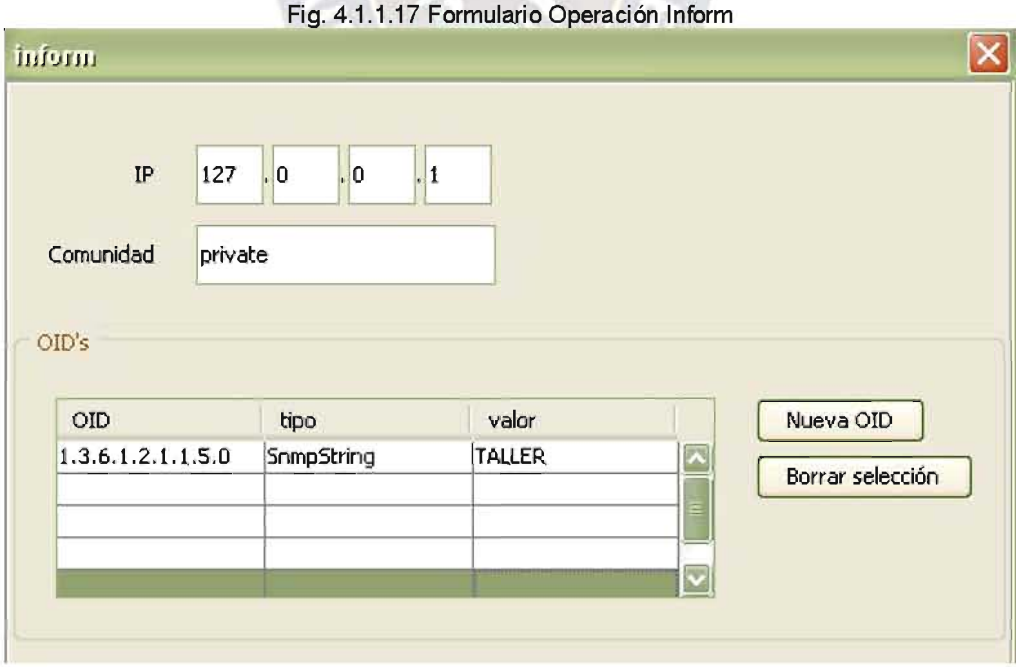
Lanzar el formulario de la operación inform. Esto se puede hacer mediante el menú Órdenes→Inform, con las teclas de acceso

Fig. 4.1.1.16 Operación Inform



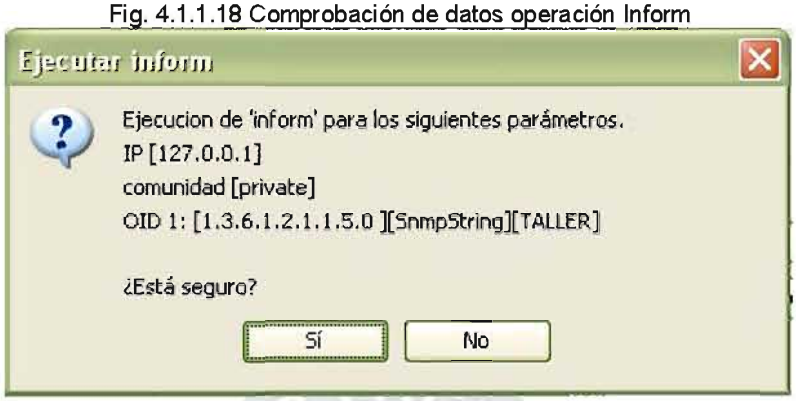
Aparecerá el formulario (es igual al de la operación set) de esta operación en el cual se deben rellenar los campos IP con el dispositivo sobre el que se quiere actuar, la Comunidad que será private y al menos una de las filas de la tabla OIDs. En esta tabla aparecen los campos OID de las variables a las que se quiere informar al gestor, tipo y valor de la variable. Para este ejemplo se realizará sobre el dispositivo 127.0.1 con la variable de oid 1.3.6.1.2.1.1.5.0 que es de tipo String y con su valor en la MIB que es TALLER.

Fig. 4.1.1.17 Formulario Operación Inform



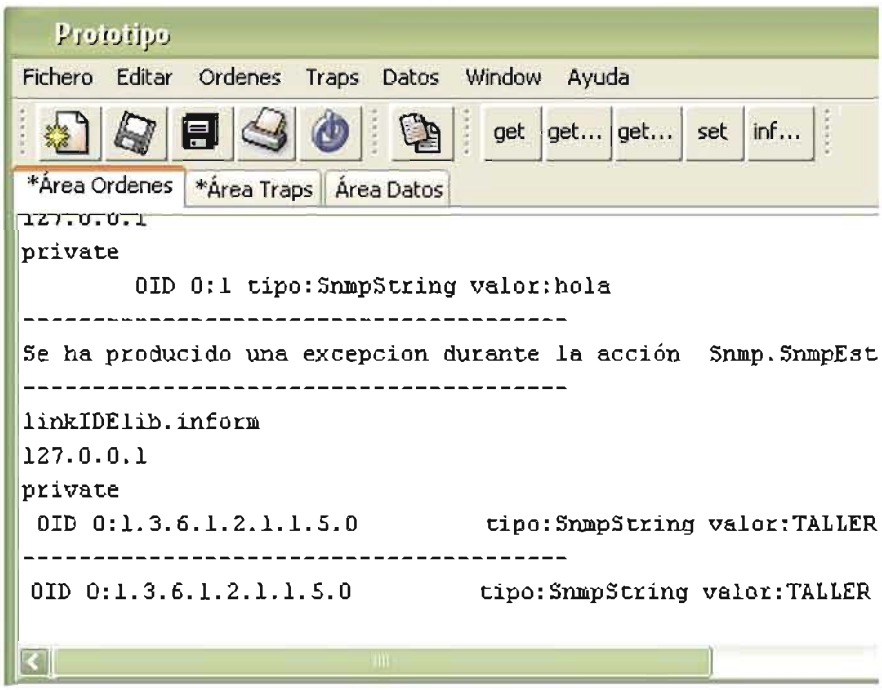
Una vez rellenados los campos, se pulsa la tecla OK y aparecerá un mensaje con los datos introducidos preguntando si se desea continuar

Fig. 4.1.1.18 Comprobación de datos operación Inform



En caso afirmativo se ejecutará la operación y la respuesta aparecerá reflejada en el área de texto "Área Órdenes". Si se pulsa NO, se volverá al formulario para poder realizar los cambios requeridos por el usuario.

Fig. 4.1.1.19 Resultado Operación Inform

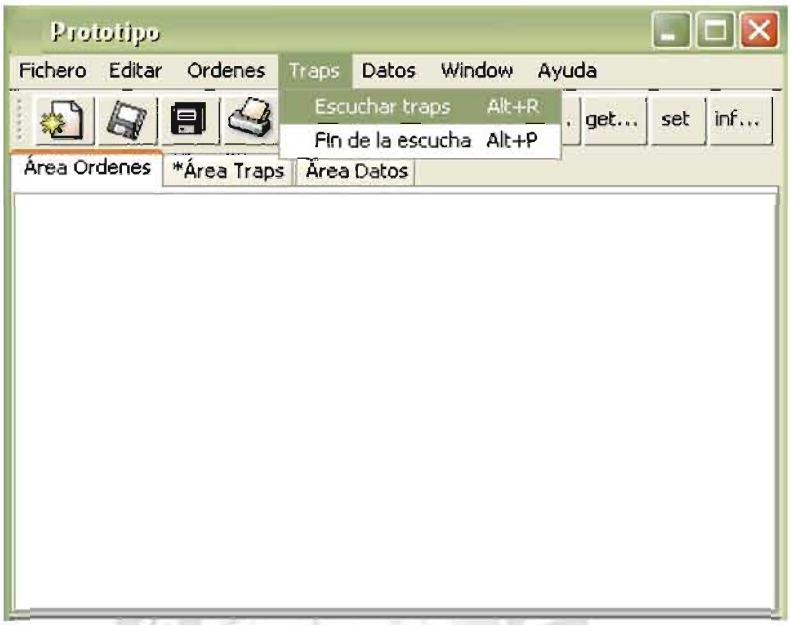


OPERACIÓN ESCUCHAR TRAPS

Mediante la ejecución de esta función, la aplicación queda habilitada para la escucha de notificaciones por parte de los agentes (traps) ó por parte de otros gestores (informs). Más información en Operaciones Snmp→Traps de la Memoria. Los pasos a seguir son los siguientes:

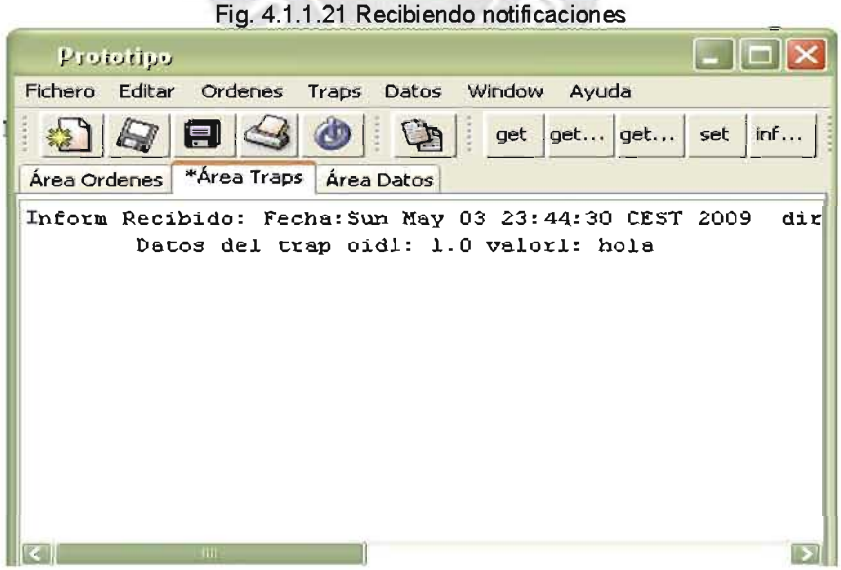
Habilitar la escucha de traps mediante el menú Traps→Escuchar Traps, con las teclas de acceso rápido Alt + E ó con el icono de la "Barra de Herramientas".

Fig. 4.1.1.20 Escucha de Traps



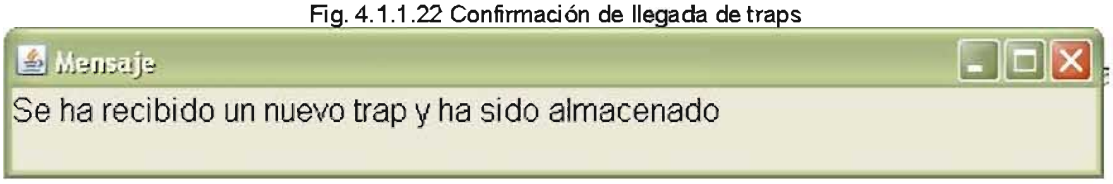
Una vez hecho esto, aparecerá un mensaje en el Área de Traps informando al usuario que la escucha ha comenzado. A partir de ese momento todos los traps e informs serán presentados en este área.

Fig. 4.1.1.21 Recibiendo notificaciones



Por cada notificación recibida aparecerá una ventana informando de este hecho y de que ha sido almacenado.

Fig. 4.1.1.22 Confirmación de llegada de traps



Todas las notificaciones que se escuchen quedarán guardadas en dos ficheros de forma automática. Si es un trap, se guardará en la ruta .\traps.txt y si es un inform en .\inform.text. Ambos archivos deben ser abiertos con la opción Aplicación MFC Wordpad y en caso de que no existan se crearán.

En cualquier momento el usuario puede dejar de escuchar notificaciones pulsando Traps→Fin de la Escucha, las teclas de acceso rápido Alt + F ó bien el icono de la Barra de Herramientas. Automáticamente aparecerá un mensaje en el Área de Traps informando del fin de la escucha.

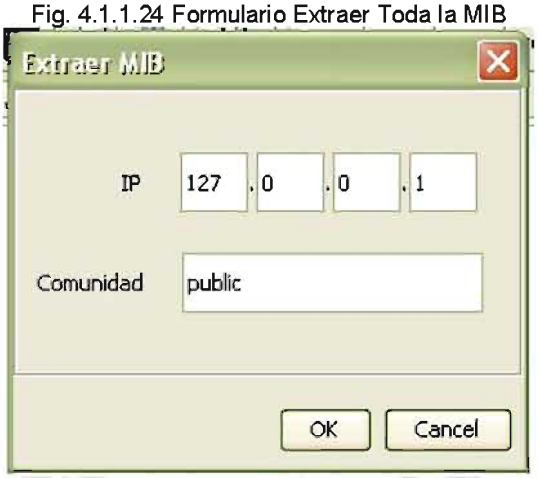
OPERACIÓN EXTRAER TODA LA MIB

Mediante esta operación el gestor puede visualizar todo el contenido de la MIB del dispositivo agente. La información aparece en el Área de Datos. Los pasos a seguir son:
Lanzar el formulario de la operación extraer toda la MIB. Esto se puede hacer mediante el menú Datos→Extraer toda la MIB, con las teclas de acceso rápido Alt + M ó bien con el icono de la Barra de Herramientas.

Fig. 4.1.1.23 Operación Extraer Toda la MIB

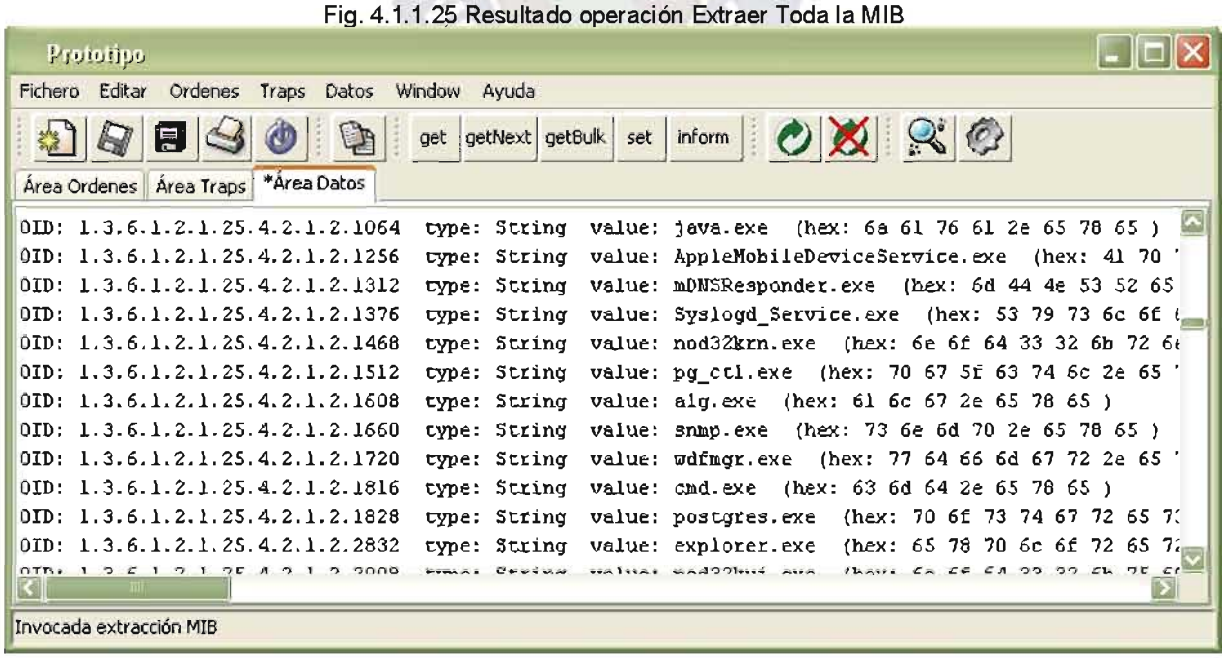
Aparecerá el formulario de esta operación en el cual se deben rellenar los campos IP con el dispositivo sobre el que se quiere actuar y la Comunidad que será public. Este ejemplo se realizará sobre la IP 127.0.0.1

Fig. 4.1.1.24 Formulario Extraer Toda la MIB



Una vez rellenados los campos, se pulsa la tecla OK y aparecerá un mensaje con los datos introducidos preguntando si se desea continuar. Si se elige la opción NO se vuelve al formulario para que el usuario pueda cambiar los parámetros que deseé, en caso afirmativo se ejecuta la operación.

Fig. 4.1.1.25 Resultado operación Extraer Toda la MIB



OPERACIÓN OBTENER TABLA DE OIDS

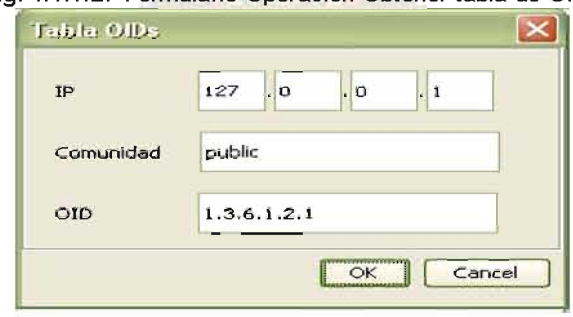
Esta operación se utiliza para descargas masivas de la MIB ya que permite visionar una gran cantidad de variables. Para ello se debe introducir el prefijo de algún OID, obteniendo como respuesta todas aquellas variables MIB cuyo prefijo coincida con el establecido. Su funcionamiento es el siguiente:

Lanzar el formulario de la operación Obtener tabla de OIDs. Esto se puede hacer mediante el menú Datos→Obtener tabla de OIDs, con las teclas de acceso rápido Alt + O ó bien con el icono de la Barra de Herrramientas.

Fig. 4.1.1.26 Operación Obtener tabla de OIDs

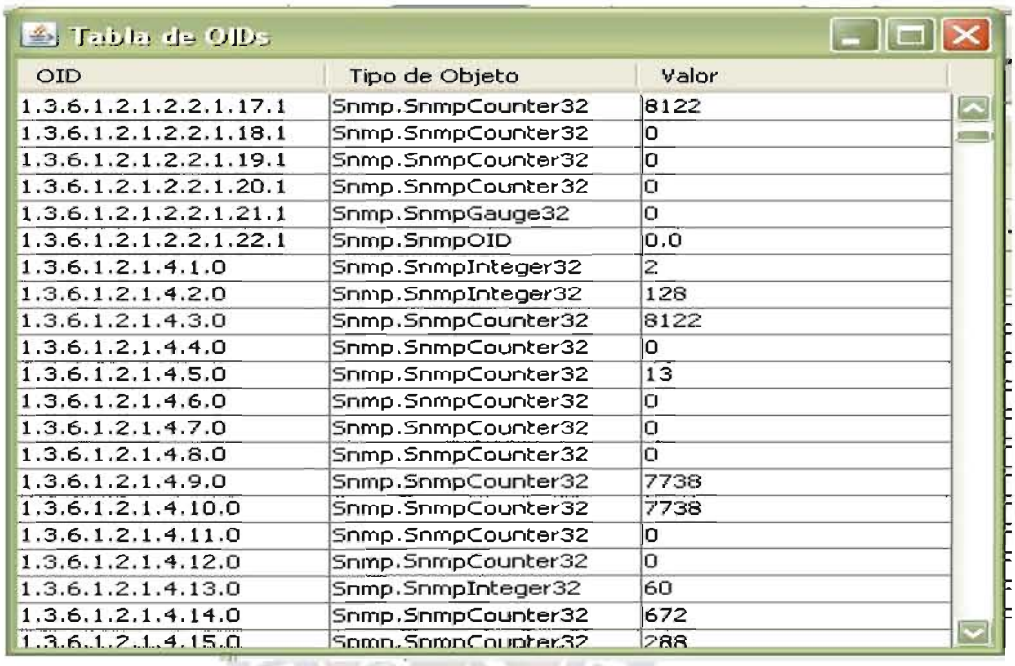
Aparecerá el formulario de esta operación en el cual se deben rellenar los campos IP con el dispositivo sobre el que se quiere actuar, la Comunidad que será public y el prefijo de los OID de las variables que se quieren visionar. En este ejemplo la IP será 127.0.0.1 y el prefijo de OID 1.3.6.1.2.1

Fig. 4.1.1.27 Formulario Operación Obtener tabla de OIDs



Al igual que en el resto de operaciones, una vez rellenados los campos, se pulsa la tecla OK y aparecerá un mensaje con los datos introducidos preguntando si se desea continuar. Si la respuesta es afirmativa aparecerá la información solicitada en el Área de Datos de la aplicación.

Fig. 4.1.1.28 Resultado operación Obtener tabla de OIDs



Además de poder ver la información en operación, también se pueden visualizar campos OID, Tipo de objeto y Valor.

4.2 CONCLUSION

Con la presente investigación se a demostrado principalmente la construcción de una herramienta para la gestión de red, en la conclusión de esta herramienta intervinieras muchos factores entre las mas importantes podemos indicar el lenguaje de programación (java), haciendo uso del protocolo snmp, los mensajes Get, GetNext, GetBulk, Set, Inform de snmp, el algoritmo de encriptación RSA .

- "Investigar y analizar sobre los protocolos Administración de Red" Se logro identificar el Protocolo Simple de Administración de Red o SNMP es un protocolo de la capa de aplicación que facilita el intercambio de información de administración entre dispositivos

de red ,permite a los administradores supervisar el desempeño de la red, buscar y resolver sus problemas. Desarrollo de la sección 2.5 "Protocolo Administración de Red(SNMP)"

- *"Analizar y seleccionar sobre técnicas de envió y recepción de mensajes protocolo Administración de Red" El protocolo SNMP utiliza un servicio no orientado a la conexión para enviar un pequeño grupo de mensajes entre los administradores y agentes. Los mensajes Get, GetNext, GetBulk, Set, Inform. Denotados en la sección 2.5.4 "Mensajes SNMP"*

- "Realizar un estudio de lenguajes de programación" El lenguaje de programación utilizada para realizar el prototipo fue JAVA , dando una introducción del lenguaje en la sección 2.4 "JAVA"

- "Desarrollo de la herramienta de gestión de dispositivos de red" la herramienta se concluyo, se da una descripción y funcionamiento del prototipo en la sección 3.6 Descripción del Sistema y 4.1 Presentación dela herramienta.

Se cumplió con el objetivo general de "Desarrollar una herramienta de software que posibilite la administración de los dispositivos de red" cumpliendo con los objetivos específicos de la sección 1.8.2 y con las contribución del Marco Referencial capitulo 2, Proceso de Investigación capitulo 3.

El problema señala ¿La herramienta para la gestión de los dispositivos de una red, permitirá una buena y ágil gestión en los dispositivos de una red?

Llevado a cabo los objetivos específicos y haber logrado el objetivo general se da por solucionado el problema además del desarrollo de la herramienta y sometida a prueba.

La hipotesis del presente trabajo sostiene que "***Hi***: La herramienta para la gestión de los dispositivos de una red **ayudara** a mejorar la administración de los dispositivos de una red LAN"

Con la construcción de la herramienta donde hace uso de los mensajes Get, GetNext, GetBulk, Set, Inform, a través de la Mib se podrá solicitar al agente sobre su estado. Se prueba experimentalmente en la sección 4.1 Presentación de la herramienta, por lo tanto la hipótesis planteada se a demostrado

4.2.1 RECOMENDACIÓN

Como consecuencia del presente trabajo surgen algunos típicos que pueden ser ampliados en futuros trabajos relacionados con este tema en este sentido se pretende incorporar las siguientes recomendaciones:

Trabajar con la versión de SNMP.v3 ya que es la versión actual del protocolo de gestión de red simple pues esta contempla una mayor variedad de mensajes, para el monitoreo de red.

Utilizar una base de datos para luego mostrar mediante distribuciones estadísticas el comportamiento del dispositivo de red monitorizado.

Profundizar el estudio en el protocolo de transporte de datos TCP con la integración del protocolo de simple administración de red SNMP.

BIBLIOGRAFIA

- Bunge Mario, LA INVESTIGACIÓN CIENTÍFICA, 2da Edición., Editorial Ariel, México 1983.
- Douglas E. Comer y David Stevens, INTERCONECTIVIDAD DE REDES, 3ra Edición, México 2000, Editorial Pearson Education.
- Douglas Mauro, Kevin Schmidt, ESSENTIAL SNMP, 2da Edition, Editorial O'Reilly United Estates 2005,
- Hernández R. Fernández C. Pilar L, METODOLOGIA DE LA INVESTIGACION, 5ta Edición, México 2006, Editorial Mc Graw Hill.
- Mischa Schwartz, REDES DE TELECOMUNICACION, 1ra Edición, Editorial Addison Wesley Iberoamericana.
- Nestor G. Sainz , COMUNIACON DE REDES Y PROCESAMIENTO DE DATOS, 1ra Edicion, Colombia 2002, Editorial Mc Graw Hill.
- Pressman R, INGENIERIA DEL SOFTWARE, 6ta Edición, México 2006, Editorial Mc Gran Hill. .
- SunSoft , PROGRAMMING REFERENCE, http://dlc.sun.com/pdf/802-1312/802-1312.pdf, Sun 1996.
- Sun Microsystems.Inc., SNMP Management Service Package, http://www.sun.com
- Seguridad Criptográfica ,http://www.inixa.com/seguridad_criptografia.php