UNIVERSIDAD MAYOR DE SAN ANDRES

FACULTAD DE INGENIERIA

CARRERA DE INGENIERIA ELECTRONICA

PROYECTO DE GRADO

PROTOTIPO DE UN SISTEMA DE GESTION DE RED, PARA LA SUBRED DEL LABORATORIO DE MULTIMEDIA, DE LA CARRERA DE INGENIERIA ELECTRONICA

POSTULANTE: **RAMIRO GUMERCINDO ALBERTO BELTRAN**

TUTOR: **ING. FELIX JAVIER SANABRIA G.**

LA PAZ – BOLIVIA

2022

UNIVERSIDAD MAYOR DE SAN ANDRÉS
FACULTAD DE INGENIERIA

LA FACULTAD DE INGENIERIA DE LA UNIVERSIDAD MAYOR DE SAN ANDRÉS AUTORIZA EL USO DE LA INFORMACIÓN CONTENIDA EN ESTE DOCUMENTO SI LOS PROPÓSITOS SON ESTRICTAMENTE ACADÉMICOS.

LICENCIA DE USO

El usuario está autorizado a:

a) Visualizar el documento mediante el uso de un ordenador o dispositivo móvil.
b) Copiar, almacenar o imprimir si ha de ser de uso exclusivamente personal y privado.
c) Copiar textualmente parte(s) de su contenido mencionando la fuente y/o haciendo la cita o referencia correspondiente en apego a las normas de redacción e investigación.

El usuario no puede publicar, distribuir o realizar emisión o exhibición alguna de este material, sin la autorización correspondiente.

”Un agradecimiento fraterno a todos mis docentes que forjaron mi espíritu de fuerza en la búsqueda del conocimiento más allá de mis límites, en especial al Ingeniero Javier Sanabria G., por brindarme la orientación, por ser mi guía para encontrar mi camino”

INDICE

INDICE DE FIGURAS

INDICE DE TABLAS

RESUMEN

Muchas instituciones empresariales poseen un conjunto de políticas y normas que gracias al uso de diferentes recursos, intervienen en la planificación, control y gestión de su institución. Un recurso de mucha importancia usado en nuestros días, es el uso de herramientas informáticas que establecen una comunicación instantánea, es así que la Carrera de Ingeniería Electrónica de la U.M.S.A., cuenta con una herramienta informática llamada CDE (Comunicación Docente Estudiante), para el uso en la metodología de la educación superior. El sistema es ejecutado en el internet, y funciona en servidores propios de la universidad, bajo una red de datos. Todas las dificultades que podría presentar la red de datos durante las 24 horas de su funcionamiento puede ser controlada y administrada por un Sistema de Administración de Red, de las cuales existen muchas en el mercado informático, bajo licencias costosas por parte de grandes proveedores de software, muchos de los cuales llegan a ser robustos y complejos en su comprensión y operatividad, tanto que algunas proveedores certifican el uso de su producto, bajo un curso de preparación y actualización constante acerca de su manejo y funcionamiento.

El presente proyecto realiza la Ingeniería a la Inversa sobre un producto de Administración de Red perteneciente a IBM, y a partir de esta, diseña e implementa un software propio llamado Sistema de Gestión de Red, bajo un entorno amigable y de fácil comprensión para el usuario. Este es un prototipo que permite al administrador de red, acceder al Sistema Gestor, bajo un ID de usuario y contraseña, para crear, modificar y actualizar los datos de los formularios relativos a los Usuarios, así como las Estaciones de trabajo que ellos usan y otros dispositivos.

También realiza el despliegue grafico estadístico del tráfico de datos (recursos de red) diario, semanal y mensual en los dispositivos de red. Muestra el uso de los recursos en los dispositivos de red más importantes. Además permite realizar diagnósticos de estatus de conectividad de los dispositivos de red, en tiempo real, usando los comandos de diagnóstico de red, de los protocolos ICMP y SNMP.

La posibilidad de agregar más funcionalidades al prototipo está abierta, ya que la administración de la información MIB y el uso del protocolo SNMP permite realizar una investigación sobre los datos recolectados, de la red de datos, dando lugar a 2 funciones importantes como son;

La captura de eventos en la red que luego genere alarmas

Crear mapas virtuales de la topología del entorno de red

Completando así la funcionalidad total que poseen todos los Sistemas de administración en el mercado informático

GENERALIDADES

1. GENERALIDADES

1.1 Introducción

En una Red abierta de datos[1], la administración de esta implica muchas tareas distintas de manejo, como la de realizar una documentación, un mantenimiento de la red, etc. Desde el momento que esta comienza a funcionar, aumentan más las tareas de responsabilidad, cuando se trata de obtener un mejor rendimiento, una mejor confiabilidad en la transmisión de datos, etc., dando lugar al origen del área de Control de red.

Una de las formas más básicas del control de la conexión de dispositivos hacia una red de datos, se tiene lugar todos los días, cuando un usuario común que realiza el proceso de inicio de sesión en su estación de trabajo, verifica que las conexiones hacia la red de datos, están funcionando correctamente, pero posteriormente podría ser, que el departamento de red reciba una llamada, indicando que uno de los usuarios no puede conectarse a la red, a lo que el encargado o administrador del departamento de red, acudirá al llamado para solucionar el problema de conexión. Sin embrago, este no es el método ideal para realizar el control de conexión hacia una red de datos. Otro aspecto de este tipo surge cuando cae un enlace de comunicaciones[2], dejando a toda la red local, aislada del mundo externo.

[1] Grupo de computadoras, impresoras y otros dispositivos que se comunican a través de un medio de transmisión.

[2] Canal de comunicaciones de red, que consiste en una ruta de comunicación entre un emisor y un receptor.

Un control de red no sería eficiente, si no se toma en cuenta un componente importante, como es el recurso de red[3], además los privilegios que goza el usuario sobre este recurso. El simple hecho de que el usuario tenga una cuenta de acceso a la red, no significa que todos los recursos de red estén a completa disposición de este usuario. Los derechos de usuario determinan, la autorización que ellos poseen para usar ciertos recursos. Estos derechos son establecidos por el administrador del departamento de red, para permitir o denegar el acceso a un determinado recurso de red. Por tal razón es importante realizar, un agrupamiento lógico de usuarios según la jerarquía de responsabilidad que ellos tengan sobre la red de datos. También es necesario realizar una documentación de todos los recursos que tiene la red.

Además realizar un control del flujo de Tráfico de datos[4] en los dispositivos de red, no solo ayudaría a predecir las necesidades de expansión del entorno físico de la red, sino también a manejar y respaldar las estadísticas del nivel de tráfico de datos, que el servicio de enlace de comunicaciones a estado prestando a la red durante los últimos tiempos.

Para alcanzar todos los componentes de solución mencionados, y para un efectivo control de la red, será necesario, usar un programa de software de Administración de Red, o crear una herramienta como un Sistema de Gestión de Red, con aplicaciones personalizadas, acorde a las necesidades que requiera la red de datos.

[3] Conjunto de servicios que presta una red de datos como; servicio de Internet, de impresora.

[4] Cantidad y tipo de paquete IP, que se transmite o se recibe en un dispositivo de red.

1.2 Antecedentes

Muchas instituciones empresariales poseen un conjunto de políticas y normas que gracias al uso de diferentes recursos, intervienen en la planificación, control y gestión de su institución. Un recurso de mucha importancia usado en nuestros días, es el uso de herramientas informáticas para establecer una comunicación instantánea.

Es así que la Carrera de Ingeniería Electrónica de la U.M.S.A., posee una herramienta informática llamada CDE (Comunicación Docente Estudiante), para el uso en la metodología de la educación superior. El sistema es ejecutado en el internet, y funciona en servidores propios de la universidad.

Diferentes dispositivos físicos conforman su red de datos, entre las cuales podemos mencionar los más importantes, equipos instalados en el nodo central, de la subred del Laboratorio de Multimedia de la Carrera de Ingeniería Electrónica.

- Servidor *FireWall*[5], para la seguridad y encriptación de los datos
- Servidor *Wamp*, para desarrollar aplicaciones *web*
- Servidor *MySQL*, para la gestión y administración de Bases de Datos
- Servidor para administración y soporte

1.3 Descripción del problema

Los diferentes servidores o dispositivos de red, intervienen en el funcionamiento de la herramienta CDE, el hecho de que deje de funcionar alguno de estos, ya sea a razón de que sobrepasen la cantidad de usuarios, que deseen conectarse al servidor o por la caída de algún enlace al internet, da origen a plantear estas preguntas: ¿Es responsabilidad del

[5] Es un dispositivo de red que garantiza el control para el acceso a una red privada.

usuario, diagnosticar los problemas del servidor, o que un problema de comunicación de enlace de internet, tenga que ver con el usuario? Responder estas dudas es importante, ya que afectan al trabajo de los usuarios y al coste de los servicios de red.

Cuando las responsabilidades de controlar un entorno de red, estén divididas, la red podrá funcionar más eficientemente. Es así que estas responsabilidades tienen que ser definidas, y exigidas sobre un área de administración. La acción de crear un área de administración exclusiva, para el funcionamiento de la herramienta CDE, puede limitar los recursos económicos del departamento, pero la acción de no crearlo, puede ocasionar que sea complicado solucionar con eficacia, los problemas de la red. Previendo que la red de datos, en el futuro prestará su servicio a más usuarios, también aumentaran los costes de mantenerla, actualizarla y controlarla a causa de:

- El aumento de más dispositivos de la red
- Las reparaciones a los dispositivos de red
- El despliegue de software
- La formación técnica a los administradores de la red

El funcionamiento continuo de los servidores y dispositivos de red debe estar garantizada, durante las 24 horas del día, bajo las circunstancias más imprevisibles. La caída de alguno de los servidores o dispositivos de red, creará un congestionamiento de tráfico, que afectara a los demás dispositivos de red, finalmente podría caer toda la red.

Los proveedores de servicio de Internet, proporcionan un recurso de red (ancho de banda) muy importante. Es importante conocer el valor nominal y real de este recurso, que el proveedor presta. A la hora de explotar la capacidad máxima de este recurso, resulta que

el valor nominal ofrecido por el proveedor, no es igual al valor real recibido. Otras deficiencias, como la caída del enlace a internet, perjudican al funcionamiento óptimo del sistema de red de datos local.

1.4 Presentación de la solución

En una red en crecimiento, será de gran ayuda darle un respaldo al área de informática, con un Sistema de Gestión de Red, que ayude a administrar, monitorear y documentar los recursos de red, además que servirá como base fundamental para crear un área de administración exclusiva para el CDE , y permitirá la expansión de la red en el futuro. Explotar cada recurso de red, para aumentar el rendimiento permitirá disminuir la necesidad de adquirir nuevos equipos en la red, lo que implica poseer un conocimiento acerca de sus características, funcionalidades etc. de cada dispositivo de red, por lo tanto será necesario levantar un informe de auditoría, de todos los equipos en la red.

Usando un Sistema de Gestión de Red para capturar y guardar el historial de tráfico, de los últimos eventos ocurridos, y luego analizarlos, ayudara a determinar las posibles causas que permiten los eventos de caída, congestionamiento de tráfico, etc., en la red. Un monitoreo y recolección estadística del flujo de tráfico de datos, ayudará a determinar el número de paquetes descartados en una Interfaz[6]. El descartar paquetes, ocurre cuando hay falta de espacio de búfer, sobre la interfaz que indica la congestión de tráfico. Identificar los equipos que no permiten el flujo de tráfico es ideal para detectar problemas de rendimiento.

[6] Conexión entre dos sistemas o dispositivos.

Utilizando un Sistema de Gestión de Red sobre un recurso de red (ancho de banda), para obtener un historial del comportamiento del tráfico del enlace que presta el proveedor de servicios de enlace, ayudará a realizar revisiones, observaciones del servicio real que se está utilizando, posteriormente este historial se podría usar para justificar las deficiencias de servicio que el proveedor haya prestado.

Aparte de la comodidad que brindara, el Sistema de Gestor para realizar las tareas de consulta mencionadas, la interfaz gráfica del SGR también está orientado a prestar una ventana al administrador, para que ayude a llevar el control de la red más eficientemente.

El uso de los recursos en una red ayudara a saber qué cantidad de su capacidad está siendo usada, sobre un recurso de red (Ej. Un servidor, una impresora, etc.), que usuarios están conectados, que tipo de sesión ellos realizan. Todo esto ayudara, a prevenir el colapso de los recursos, a realizar actualizaciones de equipo o software o finalmente cambiar, adicionar un dispositivo. La información sobre cada dispositivo de red, creará la documentación de la red local, también proporcionara, un punto de partida para que el administrador pueda volver en caso de que una actualización termine mal. Todo esto permitirá registrar, actualizaciones y cambios de configuración que se realice en la red.

Observar gráficamente el tráfico, en sus diferentes capas del modelo TCP/IP (Protocolo de Control de Transporte/ Protocolo de Internet), ayuda a realizar un análisis del comportamiento de los recursos de red, que los dispositivos de red usan sobre sus interfaces. Poseer esta información gráfica ya sea en tiempo real, diario, semanal, mensual, o en determinados horarios, permite al administrador observar el comportamiento de los equipos y recursos de red, para realizar actualizaciones o incorporaciones de nuevos recursos y equipos en la red local.

1.5 Justificación técnica

Existen muchos productos o sistemas, orientados a la Administración de Redes de Datos, creados por las empresas de desarrollo de hardware y software, muchos de los cuales llegan a ser robustos y complejos en su comprensión y operatividad, tanto que algunas empresas certifican el uso de su producto, bajo un curso de preparación y actualización constante acerca del manejo y funcionamiento del producto. Sin embrago en redes emergentes y en crecimiento, se requiere administrar la red lo antes posible, por lo que una herramienta que reúna con las siguientes características más esenciales como:

- Capacidad para realizar tareas básicas de administración y solución de problemas de red, usando comandos de monitoreo y estatus relacionadas a la administración de red.
- Una aplicación que ofrezca al usuario una interfaz amigable, de fácil comprensión y manejo de su funcionamiento.
- El costo económico para desarrollar el Sistema, debe ser moderadamente baja en comparación a los sistemas de administración de red que existen en el mercado.

1.6 Objetivo general y sus especificaciones

Diseñar e implementar el "Sistema de Gestión de Red" (SGR), que permita en los dispositivos de red; mostrar su estadística de tráfico de datos, levantar su auditoria requerida y monitorear su estado de conexión.

Diseñar una Interfaz Gráfica, (GUI), usando una página web, para que realice las siguientes tareas;

- El administrador de la red, bajo un ID de usuario y contraseña pueda acceder al Sistema Gestor, para crear, modificar y actualizar los datos de los formularios relativos a los Usuarios, así como las Estaciones de trabajo que ellos usan y otros dispositivos. Lo cual incluirá; datos personales del usuario, y detalles de configuración, listados de software, registros de mantenimiento, seguridad, etc. de las estaciones de trabajo y otros dispositivos de red.
- Realizar diagnósticos de conectividad de los dispositivos de red, en tiempo real, usando los comandos de diagnóstico de red, de los protocolos ICMP y SNMP.
- Permita el despliegue grafico estadístico del tráfico de datos (recursos de red) diario, semanal y mensual en los dispositivos de red, que muestre el uso de los recursos en los dispositivos de red más importantes.

Para conseguir los objetivos anteriores se tendrá que realizar una serie de tareas, tanto en el ámbito de la investigación y experimentación como las siguientes actividades;

- Elegir una aplicación de software, que realice tareas de administración de la Base de Información para la Gestión (MIB), que usa el Protocolo Simple de Administración de Red (SNMP), y realizar su revisión bibliográfica, acerca de su funcionalidad, configuración y capacidades avanzadas que este presta
- Investigar acerca de las capacidades de gestión y manejo, que ofrecen los diferentes servidores de Base de Datos Relacional (RDB), para elegir uno, que este acorde a las necesidades del SGR
- Diseñar un método de conexión e interacción, entre una Base de Datos Relacional y una página web
- Investigar acerca de aplicaciones visuales, que permitan leer datos de la RDB

FUNDAMENTO TEORICO

2 FUNDAMENTO TEORICO

2.1 Sistema de administración de red

Los sistemas que monitorean el rendimiento de una red de datos, son ideales para empresas que desean disponer de una gestión integral, sobre sus aplicaciones críticas de negocio.

Estos sistemas se basan en el uso de dos protocolos importantes, SNMP e ICMP, y la administración de la información MIB.

2.2 Protocolos de administración de red

- **Protocolo de mensajes de control en Internet (ICMP).** Es un protocolo de Internet de la capa de red, que indica errores y proporciona información relevante para el procesamiento de paquetes IP.
- **Protocolo Simple de Administración de Redes (SNMP).** Es un protocolo de la arquitectura de administración de red, que tiene dos componentes:
- **Estación de administración.** Es la interfaz del administrador de red (donde reside el sistema que administra el protocolo SNMP), que controla la red, además administra los MIB (Base de Información de Administración), que se extrae de los dispositivos sometidos a esta administración.
- **Agente de administración.** Es el componente que está en los dispositivos a ser administrados.

SNMP usa los siguientes comandos para comunicarse con los agentes:

- **GET:** Solicita de datos del agente, y este responde con los datos solicitados.

- **SET:** Configura nuevos datos a una o más variables del dispositivo manejado por el agente.
- **TRAP:** permite la posibilidad de interceptar al agente, y luego este notifique a la consola de administración sobre los eventos importantes.

2.3 Base de información de administración (MIB)

Es la información que describe la característica física y lógica de objetos de red, reside sobre un sistema agente. Esta información puede ser accedida y cambiada por el agente, en razón a las peticiones hechas por el administrador SNMP. Existen dos tipos de MIB, el estándar y el específico del fabricante. Algunas de las definiciones para los MIB son:

- ibm-mib
- rfc1213-mib-ii, rfc1231-802.5

2.3.1 MIB estándar rfc1213-mib-ii

Es el conjunto de definiciones de objeto estándar, del cual un MIB_II[7] está compuesto. Son registradas principalmente para manejar y ser usados en redes basadas en TCP/IP, a través del protocolo SNMP.

2.3.2 MIB especifico de fabricante

SNMP permite al fabricante definir extensiones de MIB, específicamente para controlar sus productos. Este MIB específico debe seguir ciertas definiciones y estándares

[7] Es la base de datos común para la gestión de equipos en internet, sobre SNMPv2 y SNMPv3.

como los otros MIB, para asegurar que la información que ellos contengan, pueda ser accedida por los agentes.

2.3.3 Convenciones de nombrado de MIB

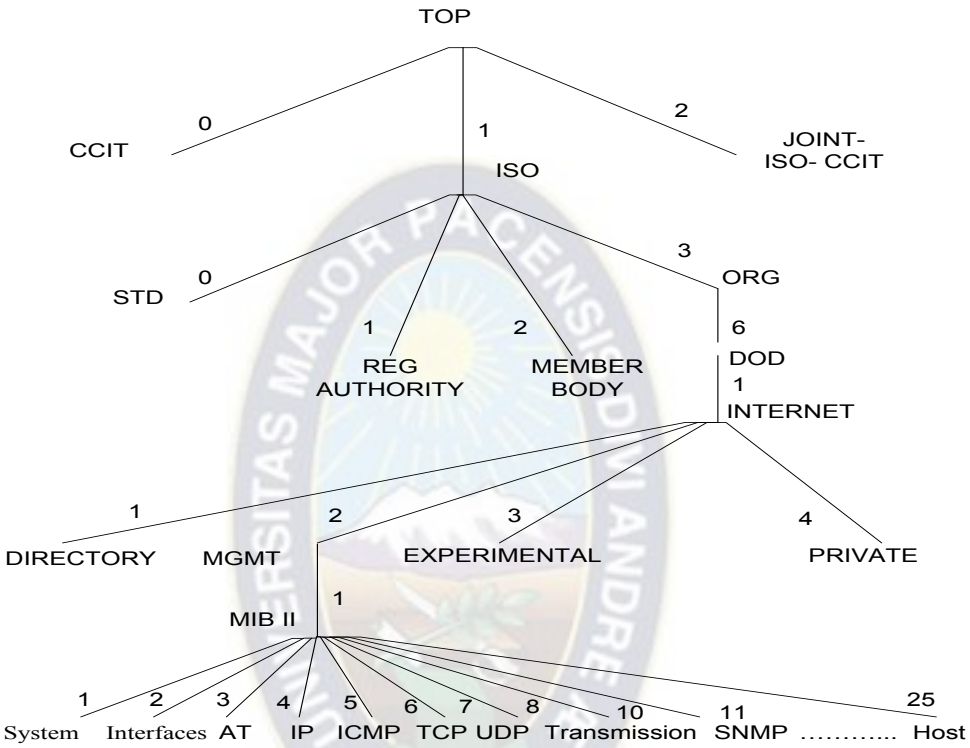


Fig. 1, Estructura del árbol MIB

Fuente: Guía de usuario de *NetView for Windows*

Los objetos MIB son organizados en una jerarquía lógica, llamada "estructura de árbol". El nombre llamado objeto ID, es creado por la ruta, que se toma desde la cúspide de la estructura del árbol, hasta la punta de la ramificación del objeto mismo. Cada lugar donde la ramificación termina se llama nodo hoja de rama. Un nodo rama es el actual objeto MIB. Solamente los nodos objetos retornan valores MIB desde el agente.

2.4 Algunos sistemas de administración conocidos

En el mercado existen muchos productos orientados a la administración de redes de datos. Algunos de estos son:

2.4.1 Monitor de red de Microsoft.

Es una herramienta de diagnóstico de red, que supervisa las redes de área local, y muestra los siguientes tipos de información:

- La dirección de origen del equipo que envía una trama[8] a la red
- La dirección de destino del equipo que recibió la trama
- Los protocolos utilizados para enviar la trama
- Los datos o una parte del mensaje que se envía

Después de capturar la información, diseña un filtro de presentación para mostrar la información, en la ventana del visor de tramas de su Monitor de red.

2.4.2 Sistema *Fluke Networks.*

Es un sistema integral de monitorización del rendimiento de servicios de red. Usa las fuentes de información, disponibles en la infraestructura de red como SNMP y Cisco IP SLA[9]. Analiza los servicios y las aplicaciones en red. Permite conocer dónde se produce un cuello de botella, y gestiona el rendimiento.

[8] Agrupación lógica de información enviada, como una unidad de la capa de enlace del modelo OSI.

[9] Herramienta de Cisco de monitoreo continuo activo de tráfico en la red.

2.4.3 Sistema *Application Performance Analyzer.*

Es idóneo para aplicaciones TCP (Protocolo para el Control de Transmisión) y UDP (Protocolo de Datagrama de Usuario), combina la potencia del análisis microscópico de un analizador de protocolos, con la facilidad de uso de un sistema de control estadístico.

2.4.4 Sistema de administración de red *NetView*

Es una aplicación de IBM para el control de tráfico de datos, basado y respaldado bajo el protocolo ICMP y SNMP. Posee un programa llamado *netmon* que informa sobre los datos obtenidos de los dispositivos, y genera peticiones especiales que el administrador los requiera.

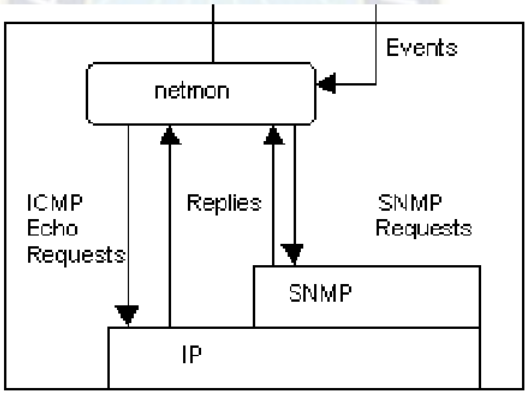


Fig. 2, Interacciones de comunicación

Fuente: Guía de usuario de *NetView for Windows*

Los objetos o dispositivos de red que han sido configurados como agentes, devuelven la información MIB, también respuestas de eco ICMP hacia el sistema administrador. La aplicación del sistema *NetView* usa ICMP para sus consultas y respuestas en aquellos dispositivos que no usen SNMP.

2.4.4.1 Servidor de Datos *NetViewDB*.

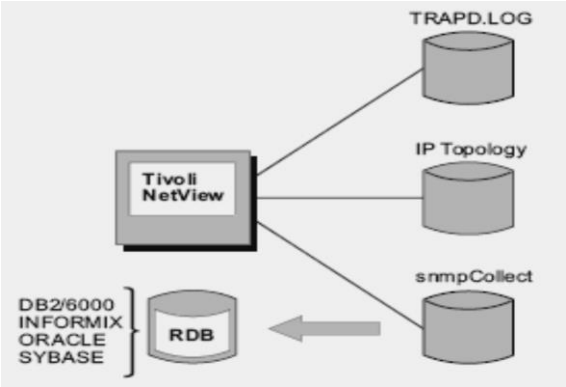


Fig. 3, Base de Datos relacional

Fuente: Guía de usuario de *NetViewfor Windows*

El servidor de datos del sistema *NetView* tanto en el sistema operativo Windows y la familia Unix[10], almacena tres tipos de datos, los cuales son:

- Datos de la topología IP.
- Datos tipo *Trap*.
- Datos *snmpCollect.*

El sistema permite exportar estos datos, a una Base de Datos Relacional externa. El sistema *NetView* para Windows, permite crear la base de datos *NetViewDB*, sobre un servidor *SQL* previamente instalado. El cual presenta 3 tipos de tablas, llamadas *schemas.*

- Tabla para los eventos.
- Tabla para snmpCollect
- Tabla para la topología de red.

[10] Conjunto de productos implementados bajo la compañía Unix.

2.5 Lenguajes de programación de páginas Web

Para que un sitio Web sea dinámico e interactivo con el cliente, es necesario que un programa, recoja las peticiones del usuario y genere una página Web personalizada. Estos programas se denominan CGI's (*Common Gateway Interface*), además estos se relacionan con bases de datos, en donde los programas depositan y recogen la información que necesitan.

2.5.1 Lenguaje de programa *Java Server Pages* (JSP).

Es una tecnología JAVA que permite generar dinámicamente las páginas web, en forma de documentos HTML (Lenguaje de Marcado de Hipertexto). El rendimiento de una página JSP es el mismo que se realiza sobre el servidor de consulta. Esto hace que JSP sea más eficiente que otras tecnologías web, que ejecutan el código de una manera puramente interpretada.

2.5.2 Procesador de Hipertexto PHP.

Es un lenguaje de programación de código abierto, pensado para funcionar como CGI, y funciona dentro del código HTML. Con una sintaxis fácil de comprender, permite hacer cambios rápidamente, puede ser programado con funciones orientadas a objetos. La ventaja importante del lenguaje, es la de implementar funciones para la consulta y modificación de datos en una Base de Datos

2.5.3 Lenguaje de programa *Active Server Pages* (ASP).

Es una tecnología de Microsoft del tipo "lado del servidor", para páginas web generadas dinámicamente, que ha sido comercializada como un anexo a *Internet Information Services* (IIS).

2.6 Sistemas de Gestión de Base de Datos Relacional o RDBMS

La mayoría de los sitios web, trabajan con una RDBMS. Podemos citar alguno de ellos.

- **PostgreSQL**. Es un servidor, orientado a objetos de software libre. Como muchos otros proyectos de código libre, el desarrollo de PostgreSQL es dirigido por una comunidad de desarrolladores denominada (PostgreSQL Global DevelopmentGroup).
- **MySQL.** Es un RDBMS de software libre. Es muy utilizado en plataformas Linux/Windows-Apache-MySQL-PHP[11]. Su popularidad está muy ligada a aplicaciones web en PHP. Permite una lectura rápida de sus datos, pero puede provocar problemas de integridad en entornos de alta concurrencia de usuarios.
- **Oracle.** Fabricado por Corporación Oracle. Considerado como uno de los sistemas de bases de datos más completos, destacando su soporte en Transacciones, Estabilidad, Escalabilidad y Soporte multiplataforma. Aunque su dominio en el mercado de servidores empresariales, fue casi total, recientemente sufre la competencia de SQL Server de Microsoft, y de la oferta de otros RDBMS con licencia libre como; PostgreSQL, MySql.

[11] Es un servidor web HTTP de código abierto, para plataformas Unix, Microsoft Windows.

2.6.1 Microsoft SQL Server

Es un RDBMS basado en el lenguaje *Transact-SQL*[12], capaz de poner a disposición de muchos usuarios, grandes cantidades de datos de manera simultánea. Sus características son:

- Escalabilidad, estabilidad y seguridad
- Permite trabajar en modo cliente-servidor bajo un entorno gráfico
- Permite administrar información de otros servidores de datos
- Disparadores (*triggers*). Es la ejecución de un procedimiento almacenado basado en una determinada acción, sobre una tabla específica. SQL permite crear una amplia funcionalidad a través de su sistema de activación de disparadores. Estos se los define por las siguientes características; El momento en que el disparador debe arrancar, el evento que lo activa para llamar una función específica.

[12] Es una extensión al SQL de Microsoft y Sybase. SQL, para realizar búsquedas, alterar y definir bases de datos relacionales

2.7 Ingeniería de software

La Ingeniería de Software podría definirse como; “El establecimiento y el uso de principios sólidos de la ingeniería para obtener económicamente un software confiable y que funcione de modo eficiente en máquinas reales”. Esta definición no menciona los aspectos técnicos de la calidad de software, ni la necesidad de satisfacer al cliente o el tiempo de entrega de un producto. El IEEE (Instituto de Ingenieros Electrónicos y Eléctricos) define como la aplicación de un enfoque sistemático, disciplinado y cuantificable al desarrollo, operación y mantenimiento del software; es decir, la aplicación de la Ingeniería de Software.

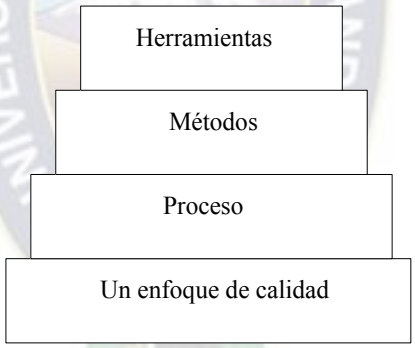


Fig. 4, Estratos de la ingeniería de software.

Fuente; elaboración propia

Como se muestra en la figura 2.4, la Ingeniería de Software es una tecnología estratificada y la base que la soporta está orientada al Enfoque de la calidad. El Proceso de Desarrollo de Software forma la base, para el control de la gestión de los proyectos de software, y establece el contexto en el cual se aplican los métodos técnicos, se generan productos de trabajo (modelos, documentos, datos, etc.), se establecen los fundamentos, se

asegura la calidad y cualquier cambio se maneja de manera apropiada, todo esto permite que el software que se desarrolla, se lo realice de manera racional y que se lo entregue a tiempo al cliente.

Los Métodos de la Ingeniería del Software proporcionan los "como técnicos" para construir software. Se basan en un conjunto de principios básicos, que gobiernan el área de la tecnología actual, incluye actividades de modelado del diseño, la construcción del software, la realización de pruebas y el soporte. Finalmente la Herramienta de la Ingeniería de Software proporciona el soporte automatizado para el desarrollo del software.

2.7.1 Modelos de Proceso para desarrollar software

Las siguientes actividades se usan como base, para la descripción de modelos de Proceso

- Comunicación. Esta actividad del marco de trabajo, implica una intensa colaboración con los clientes, además abarca la investigación de requisitos.
- Planeación. Abarca la creación de modelos, que permiten al desarrollador y cliente entender mejor, los requisitos del software y el diseño que lograra satisfácelos.
- Construcción. Esta actividad combina la generación del código, (ya sea manual o automatizado) y la realización de pruebas para descubrir errores en el código.
- Despliegue. El software (como una entidad completa) se entrega al cliente, quien evalúa el producto recibido y proporciona información basada en su evaluación.

2.7.1.1 Desarrollo Iterativo

Es un método de construcción de productos, cuyo ciclo de vida está compuesto por un conjunto de iteraciones, las cuales tienen como objetivo entregar versiones del

software. Cada iteración se considera un subproyecto que genera productos de software, permitiendo al usuario tener puntos de verificación, control y pruebas para finalmente integrarlos todos.

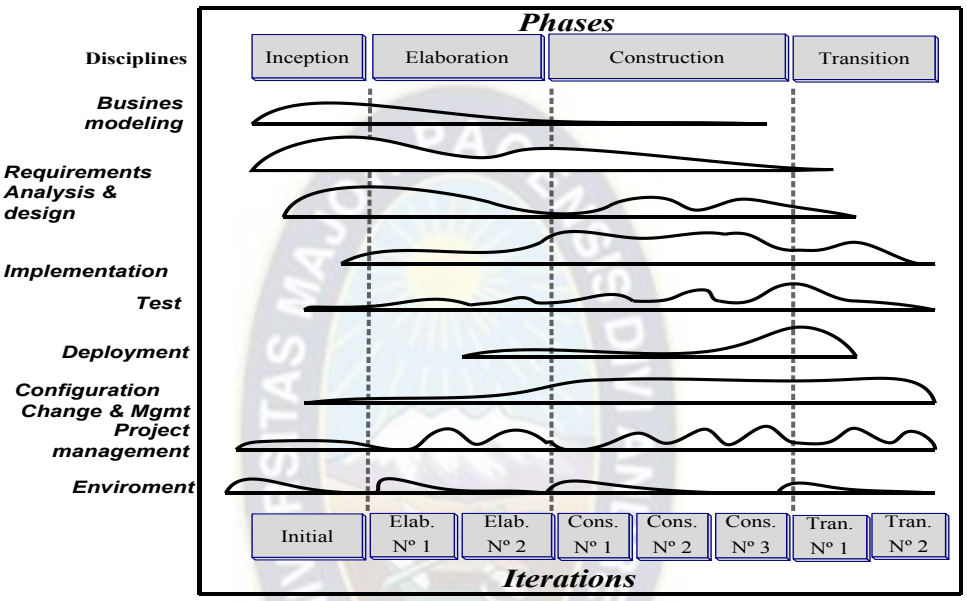


Fig. 5, 5 Disciplinas a través de las iteraciones.

Fuente: IBM RUP *Rational Unified Process*

Versión 2005 05 00 *Rational Software Corporation*

Las iteraciones posee un conjunto de disciplinas o actividades, estas son las especificaciones de requerimientos, el análisis, diseño, las pruebas, la administración de la configuración y el proceso de gerencia de proyectos. En la Figura 2.5 se muestra la relación entre las iteraciones y disciplinas o actividades.

2.7.1.2 Orientación al Manejo del Riesgo

Cada proyecto tiene asociado intrínsecamente un conjunto de riesgos, que requiere un plan de manejo bien establecido, documentado y una implementación eficaz, para evitar

retrasos en los tiempos de entrega, problemas de calidad del producto, o en el peor de los casos, que puedan afectar la culminación del proyecto.

En las etapas iníciales se implementan las funcionalidades con mayor exposición al riesgo, y complejidad, mejorando la posibilidad de éxito del proyecto.

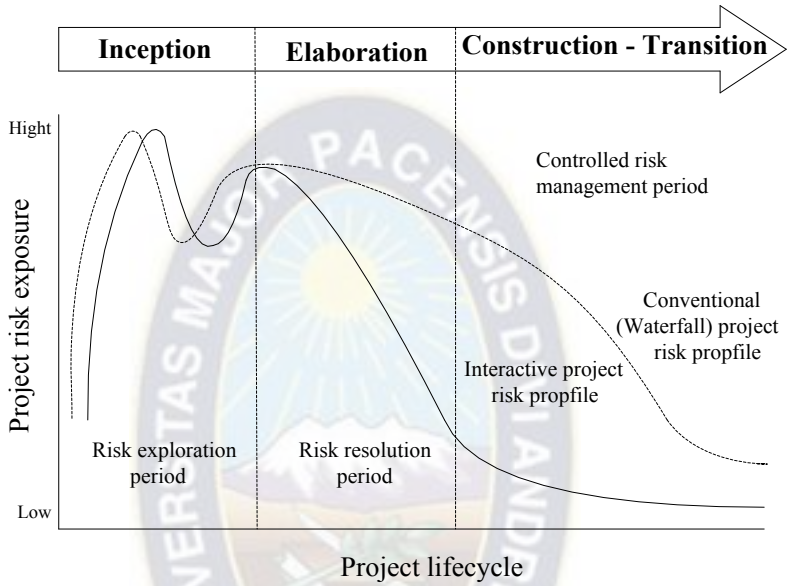


Fig. 6, Perfiles de riesgo

Fuente: *Principles of Managing Iterative Development v.2.0 Rational Software Corporation.*

2.7.1.3 Orientación al cliente

El cliente es quien conoce el valor, que aportará el producto, y define las prioridades desde la perspectiva organizacional, por lo tanto es necesario contar con su participación, en la planificación de las fases. Posteriormente se requiere su participación en cada iteración, para proveer retroalimentación temprana al equipo de desarrolladores, garantizando el cumplimiento de las expectativas que tiene, además de ofrecerle una visibilidad permanente del estado del proyecto, asegurando su compromiso para terminarlo exitosamente.

2.7.1.4 Desarrollo evolutivo

Este proceso inicia con un concepto poco claro del producto a construir, y sólo se tiene claridad en la medida que se vaya desarrollando y verificando el producto con el cliente, Este tipo de proyectos se asemejan al patrón, que siguen los proyectos de investigación y desarrollo de nuevos productos.

2.7.1.5 Modelo en Espiral

El Modelo en Espiral se centra en algunas prácticas ya citadas, tales como; la Orientación al Manejo de Riesgos, la Orientación al Cliente y el Desarrollo Iterativo (Ver Figura 7). El modelo se organiza en un conjunto de iteraciones, que pueden considerarse como pequeños proyectos, que siguen el ciclo de vida completo.

Fig. 7, Mapa conceptual del "modelo en espiral".

Fuente: *Principles of Managing Iterative Development v.2.0 Rational Software Corporation*

Las primeras iteraciones tienen como objetivo definir el alcance del proyecto, e identificar los riesgos del proyecto, para determinar su viabilidad y en caso de seguir adelante, definir un plan de manejo para mitigarlos o eliminarlos. También se realiza el

modelo de casos de uso inicial, para determinar qué casos de uso definirán la arquitectura del sistema. Además el usuario participa activamente en la priorización de los casos de uso a desarrollar, y en el proceso de pruebas, con lo cual se logra obtener una funcionalidad estable y operativa.

El modelo en espiral tiene muchas ventajas respecto a otros, por su orientación a la resolución temprana de riesgos, por la definición de la arquitectura del sistema en sus fases iníciales, y por su proceso continuo de verificación de la calidad (Ver Figura 8).

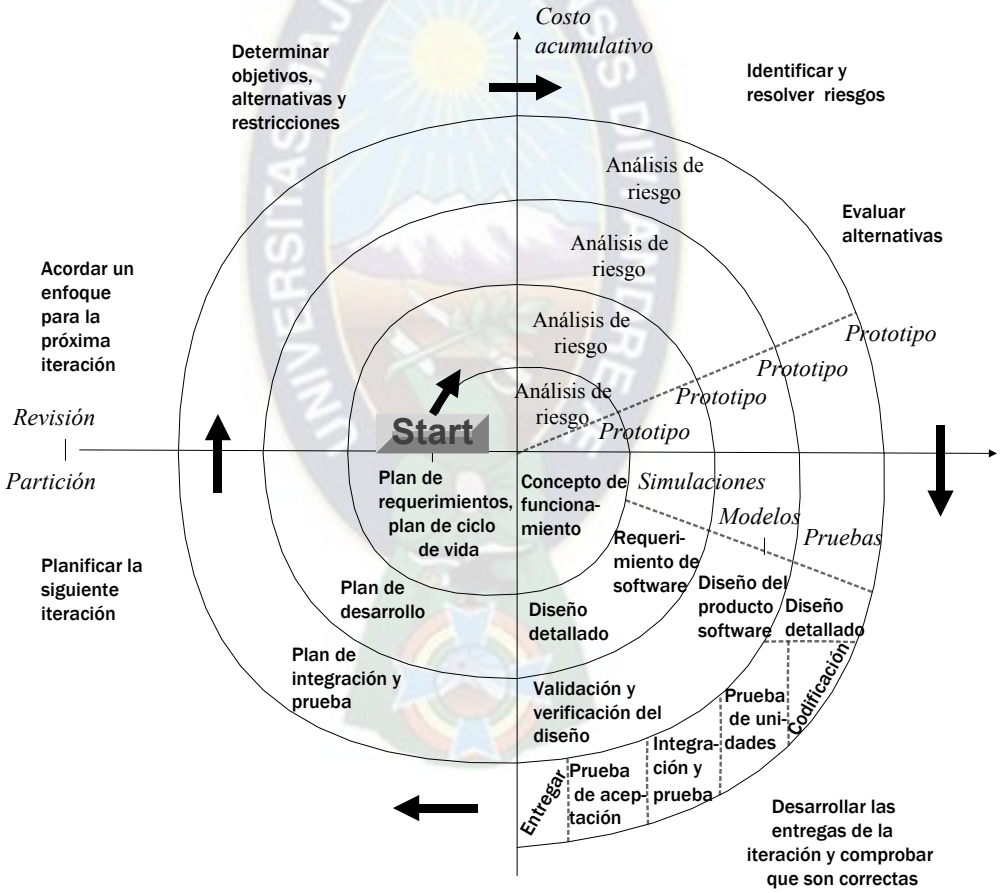


Fig. 8, Modelo en espiral.

Fuente: *Desarrollo y Gestión de Proyectos Informáticos. Steve McConnell, McGraw Hill*

2.7.2 Metodología para el proceso de desarrollo de software

La concepción de un sistema de software va mucho más allá de levantar los requerimientos, elaborar un conjunto de modelos y comenzar a programar. Es así que surge el concepto de la "Arquitectura de software", que se convierte en el eje orientador, que permitirá controlar el desarrollo iterativo e incremental del sistema, a través de su ciclo de vida. Esta arquitectura se define en la fase inicial del proyecto, y se refina a través de todo el proyecto.

Para definir la Arquitectura de software, se han formalizado métodos como; *Evo, Microsoft Solution Framework,* y UP[13] (Proceso Unificado), bajo el modelo en espiral

2.7.2.1 RUP (Proceso Unificado de Desarrollo de *Rational*)

Rational elaboró un método de referencia para el desarrollo de software, basado en el modelo en espiral.

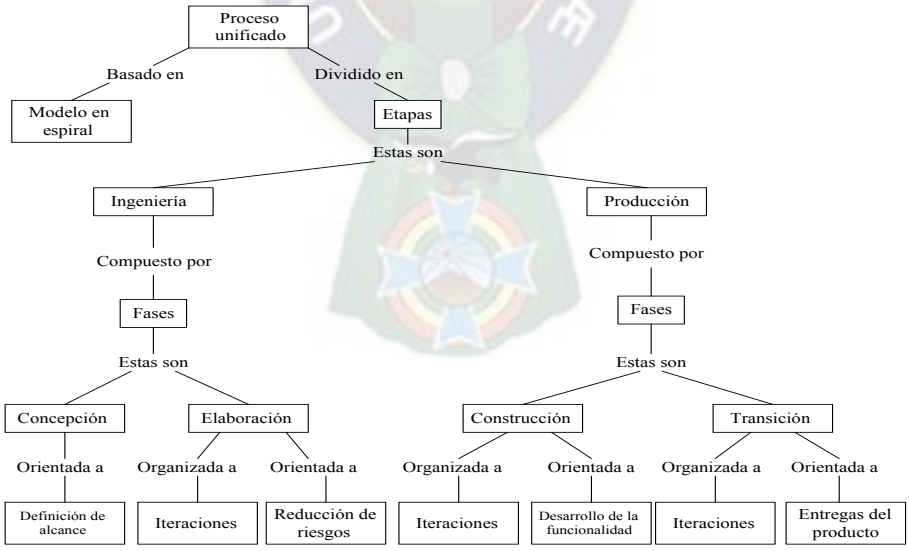


Fig. 9, Mapa conceptual del Proceso Unificado de Racional

Fuente: *Principles of Managing Iterative Development v.2.0 Rational Software Corporation*

[13] Es un Proceso de desarrollo de software.

RUP es fundamental en seis prácticas del modelo en espiral, están orientadas al modelo y con ellas se pretende solucionar muchos de los problemas asociados al desarrollo de software.

- El desarrollo iterativo
- La administración de requerimientos
- La arquitectura basada en componentes
- En el modelamiento visual
- En la verificación continua de la calidad
- La administración del cambio

Para una mejor organización, RUP agrupa las iteraciones en etapas y fases que facilitan la administración del proyecto (Ver Figura 9).

a. Etapa de ingeniería

Esta etapa agrupa la fase de conceptualización y elaboración del diseño inicial del sistema. Se inicia el proceso de administración de los requerimientos con la especificación de casos de usos, así como el proceso de aseguramiento de la calidad a través de los casos de prueba.

La fase de concepción tiene como propósito, definir y acordar el alcance del proyecto con los patrocinadores, proponer una visión general de la arquitectura de software, identificar los riesgos asociados al proyecto. En la fase de elaboración se define la arquitectura del sistema, la selección y especificación de los casos de uso, y el primer análisis del dominio del problema. Se diseña la solución preliminar del problema, y comienza la ejecución del plan de manejo de riesgos.

En la planeación de las fases y de las iteraciones y a partir del modelo de casos de uso y de la lista de riesgos, se determina qué casos de uso deben implementarse primero, para atacar los riesgos de mayor exposición y un plan de trabajo para la siguiente fase e iteración. Al final de la fase se determina la viabilidad de continuar el proyecto, si se decide proseguir, dado que la mayor parte de los riesgos han sido mitigados, se escriben los planes de trabajo de las etapas de construcción y transición, y se detalla el plan de trabajo de la primera iteración de la fase de construcción.

b. Etapa de producción

En esta etapa se realiza un proceso de refinamiento, de la estimación de tiempo y recurso, para la fase de construcción y transición, se implementa los casos de uso pendientes y se entrega el producto final, garantizando la capacitación y el soporte adecuado. La fase de construcción. Tiene como propósito, completar la funcionalidad del sistema, para ello se deben clarificar los requerimientos pendientes, administrar algún cambio en los artefactos construidos. En la fase de transición se asegura, que el software esté disponible para los usuarios finales, ajustar los errores y defectos encontrados, capacitar a los usuarios y proveer el soporte técnico necesario.

Las metodologías y estándares utilizados en el desarrollo de software, dan las guías para recorrer el camino desde antes de empezar la implementación, con lo cual se asegura la calidad del producto final, así como también la entrega del mismo en un tiempo estipulado. Para llevar al éxito la elaboración del software, la metodología RUP usa la herramienta UML, que es la base del modelamiento visual de RUP.

2.7.3 Lenguaje unificado de modelado (UML)

UML surge como respuesta al problema de no contar con un lenguaje estándar para escribir planos de software. Es una notación estándar promovida por el consorcio OMG (*Object Management Group*) para el modelado de sistemas software, ya sea para el Modelado Orientado a Objetos, Modelado de Datos, Modelado de Componentes, Modelado de Flujos de Trabajo. Al implementar un lenguaje de modelado común, se crea una documentación también común, que todo desarrollador que conozca UML será capaz de entender, independientemente del lenguaje de programación que vaya a utilizar para el desarrollo final.

UML divide cada subproyecto en diagramas que representan las diferentes perspectivas del proyecto, cada diagrama usa una notación pertinente, sin embargo se puede agrupar en dos tipos; unos que dan una perspectiva estática y otra dinámica. La combinación de estos tipos crean lo que se llama "Vistas del proyecto", gracias a estas vistas , se detecta los problemas de propagación de errores o las partes que necesitan ser sincronizadas en la estructura, así como del estado de cada una de las instancias en cada momento. Estos diagramas juntos son los que representan la arquitectura del proyecto.

Las vistas más conocidas en UML son:

- Vista casos de uso; Se forma con los diagramas de casos de uso, colaboración, estados y actividades.
- Vista de diseño; Con los diagramas de clases, objetos, colaboración, estados y actividades.

- Vista de despliegue; Con los diagramas de despliegue, estados y actividades.
- Vista de procesos; Se forma con los diagramas de la vista de diseño, recalcando las clases y objetos referentes a procesos.
- Vista de implementación; Se forma con los diagramas de componentes, colaboración, estados y actividades.

Como la complejidad del desarrollo del sistema lo requiera, pueden existir más vistas y por lo tanto más diagramas, que especifican la complejidad del proyecto. Algunos de los diagramas del tipo dinámico más usados son:

2.7.3.1 Diagrama de secuencia y colaboración

Muestran a los diferentes objetos y las relaciones que pueden tener entre ellos, los mensajes que se envían entre ellos. Son dos diagramas diferentes, que se puede pasar de uno a otro sin pérdida de información, pero que nos dan puntos de vista diferentes del sistema.

2.7.3.2 Diagrama de estados

Muestra los estados, eventos, transiciones y actividades de los diferentes objetos. Son útiles en sistemas que reaccionen a eventos.

2.7.3.3 Diagrama de actividades

Es un caso especial del diagrama de estados. Muestra el flujo entre los objetos. Se utilizan para modelar el funcionamiento del sistema y el flujo de control entre objetos.

Los diagramas del tipo estático más conocidos se mencionan con más detalle seguidamente.

2.7.3.4 Diagrama de caso de uso

El diagrama de casos de uso define, el comportamiento de una parte del sistema, solo especifica cómo deben comportarse y no como están implementadas las partes que lo definen. Un caso de uso especifica un requerimiento funcional, es decir indica "esta parte debe hacer esto, cuando pase esto". El diagrama posee diferentes figuras que pueden mantener diversas relaciones entre ellas:

- **Casos de uso:** Cada caso de uso contiene un nombre, que indica su funcionalidad. Está representado por una elipse y sus relaciones con otros casos de uso son:
  - ***Include:*** Representado por una flecha.
  - ***Extends:*** Una relación de una caso de "Uso A", hacia un caso de "Uso B" indica que el caso de "Uso B" implementa la funcionalidad del caso de "Uso A".
  - ***Generalization:*** Es la típica relación de herencia.
- **Actores:** se representan por un muñeco. Su relación es:
  - ***Communicates:*** Comunica un actor con un caso de uso, o con otro actor.
- **Parte del sistema**: Representado por un cuadro, identifica las diferentes partes del sistema y contiene los casos de uso que la forman.

Se emplea el diagrama de dos formas, para modelar el contexto y requisito del sistema.

2.7.3.5 Modelado de contexto

Se debe modelar la relación del sistema con los elementos externos, ya que son estos elementos los que forman el contexto del sistema. Los pasos a seguir son:

Identificar y organizar los actores que interactúan con el sistema

Especificar sus vías de comunicación con el sistema

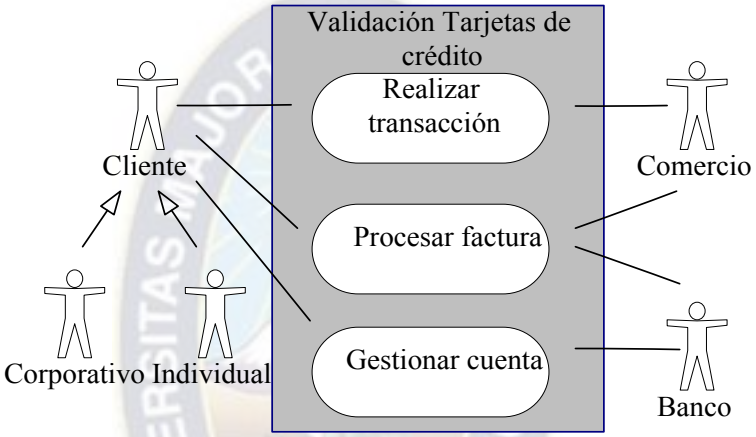


Fig. 10, Casos de uso con elementos externos.

Fuente; Elaboración propia

2.7.3.6 Modelado de requisitos

Los requisitos establecen un contrato entre el sistema y su exterior, definen lo que se espera que realice el sistema, sin definir su funcionamiento interno. No indica relaciones entre actores, tan solo indica cuales deben ser las funcionalidades o requisitos del sistema.

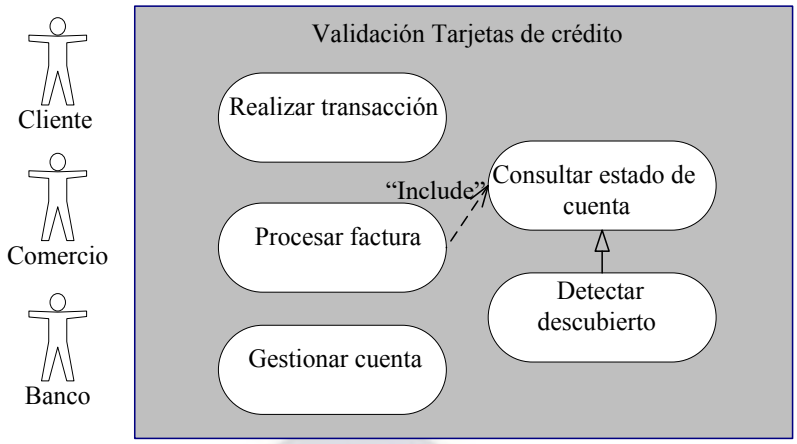


Fig. 11, Casos de uso con modelado de requisitos

Fuente; Aprendiendo UML, Joseph Schmuller *Prentice Hall*

Para modelar los requisitos es recomendable;

- Establecer su contexto, para lo que se puede usar un diagrama de casos de uso.
- Identificar las necesidades de los elementos del contexto (Actores).
- Nombrar esas necesidades, y darles forma de caso de uso.
- Identificar qué casos de uso pueden ser especializaciones de otros, o buscar especializaciones comunes para los casos de uso ya encontrados.

2.7.4 Diagrama de clases

Es donde se da libertad a los conocimientos de diseño orientado a objetos, forma parte de la vista estática del sistema. En el diagrama de clases es donde se define las características de las clases, interfaces, colaboraciones y relaciones de dependencia y generalización.

2.7.4.1 Clase

Una clase es la descripción de un concepto del dominio de la aplicación. Está representada por un rectángulo que dispone tres apartados, el primero indica el nombre

único que la diferencia de otras, el segundo es el atributo que representa alguna propiedad de la clase, mostrando su nombre y su tipo, e incluso su valor por defecto y una tercera parte llamada método u operación, es la implementación de un servicio de la clase, que muestra un comportamiento común a todos los objetos, es decir es una función que le indica a las instancias de la clase que hagan algo.

UML permite modificar las propiedades de un diagrama, mediante estereotipos y restricciones. Un estereotipo permite indicar especificaciones de lenguaje, al que se refiere el diagrama, y una restricción permite forzar el comportamiento que debe tener el objeto al que se le aplica.

2.7.4.2 Relaciones entre clases

En una relación de dos clases, la clase de origen es desde la que se realiza la acción de relacionar, es decir desde la que parte la flecha, el destino es el que recibe la flecha. Las relaciones se pueden modificar con estereotipos o con restricciones. Existen tres relaciones principales entre las clases:

2.7.4.3 Dependencias

Es una relación donde una clase usa a otra que la necesita para su cometido. Se representa con una flecha discontinua, va desde la clase utilizadora a la clase utilizada. Con la dependencia se muestra que un cambio en la clase utilizada puede afectar al funcionamiento de la clase utilizadora, pero no al contrario.

UML permite dar más significado a las dependencias, usando estereotipos. Los estereotipos de relación Clase-objeto son:

- ***Derive:*** Se utiliza para indicar relaciones entre dos atributos, indica que el valor de un atributo depende directamente del valor de otro. Es decir el atributo "Edad" depende directamente del atributo "Fecha nacimiento".
- ***Bind:*** La clase utilizada es una plantilla, y necesita de parámetros para ser utilizada.
- ***Friend:*** Especifica una visibilidad a las interioridades de la clase destino.
- ***Refine:*** Se utiliza para indicar que una clase es la misma que otra, pero más refinada, es decir dos vistas de la misma clase, la destino con mayor detalle.

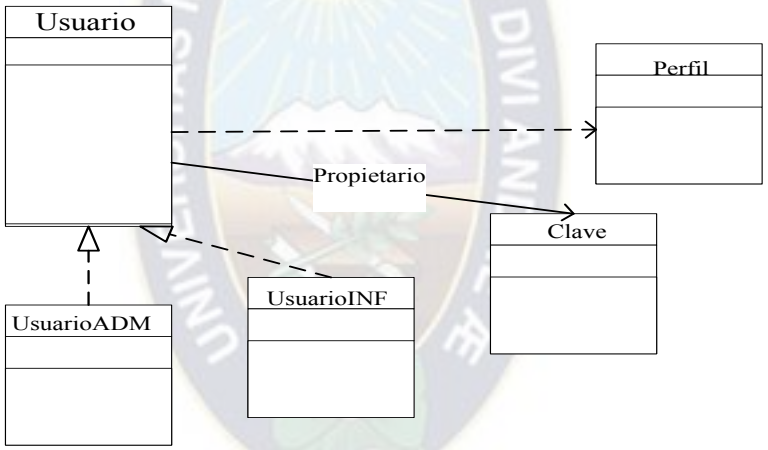


Fig. 12, Diagrama de clases.

Fuente; Aprendiendo UML, Joseph Schmuller *Prentice Hall*

2.7.4.4 Generalización.

Es la herencia, donde se tiene una o varias clases padre o superclase y una clase hijo o subclase. UML permite modificar la relación de Generalización con un estereotipo y dos restricciones. El estereotipo de generalización es:

- ***Implementation:*** El hijo hereda la implementación del padre, sin publicar ni soportar sus interfaces.

Las restricciones de generalización son:

- ***Complete***: La generalización ya no permite más hijos.
- ***Incomplete:*** Se incorpora más hijos a la generalización.
- **Asociación.** Especifica la relación entre los objetos de una clase, con los elementos de otra clase. Se representa mediante una línea continua, que une las dos clases y se puede indicar el nombre, multiplicidad en los extremos, su rol y agregación.

2.7.5 Diagrama de objetos

Forma parte de la vista estática del sistema. En este diagrama se modelan las instancias de las clases del diagrama de clases, se muestra a los objetos y sus relaciones, pero en un momento concreto del sistema. En estos diagramas también se pueden incorporar clases, para mostrar la clase de la que es un objeto representado.

2.7.6 Diagrama de componentes

Se utilizan para modelar la vista estática de un sistema. En él se sitúan librerías, tablas archivos, ejecutables y documentos que formen parte del sistema. Uno de los usos principales, es para ver que componentes pueden compartirse entre diferentes partes de un sistema.

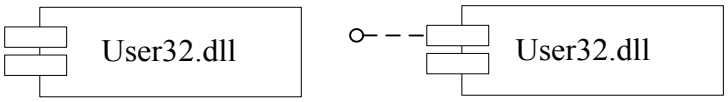


Fig. 13, Componentes.

Fuente; Aprendiendo UML, Joseph Schmuller *Prentice Hall*

En la 1ra. Fig. 13 se tiene un componente del sistema de Windows, en la otra figura se tiene el mismo componente, pero además dispone de una interface que da acceso a su contenido. Se puede modelar diferentes partes del sistema y modelar diferentes entidades que no tiene nada que ver entre ellas, como por ejemplo:

- Ejecutables y bibliotecas
- Tablas de base de datos
- API [14]
- Código fuente, Hojas HTML
- Código fuente

Se utiliza para documentar los diferentes ficheros del código fuente. Al combinar estos ficheros, mediante la relación de dependencia, se obtendrá una visión para la creación del ejecutable o librería. Al tener documentadas las relaciones, se pueden realizar cambios en el código de un archivo, teniendo en cuenta, donde se utiliza y que otros ficheros pueden verse afectados por su modificación. En la fig.2.14 se tiene la relación entre los diferentes ficheros de un sistema.

[14]Application Programming Interface, conjunto de funciones y procedimientos para ser usado por otro software

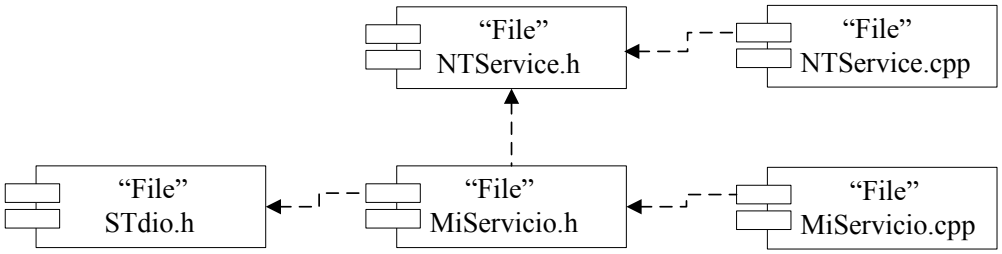


Fig. 14, Diagrama de código fuente

Fuente; El Lenguaje unificado de modelado, Grady Buch

.

2.7.7 Diagrama de despliegue

En el diagrama de despliegue se indica, la situación física de los componentes lógicos desarrollados, es decir se sitúa el software en el hardware que lo contiene. Cada hardware se representa como un nodo. Un nodo se representa como un cubo, es un elemento donde se ejecutan los componentes y representa el despliegue físico de estos componentes.

Como los componentes pueden residir en más de un nodo, se puede situar el componente de forma independiente, sin que pertenezca a ningún nodo y relacionarlo con los nodos en los que se sitúa.

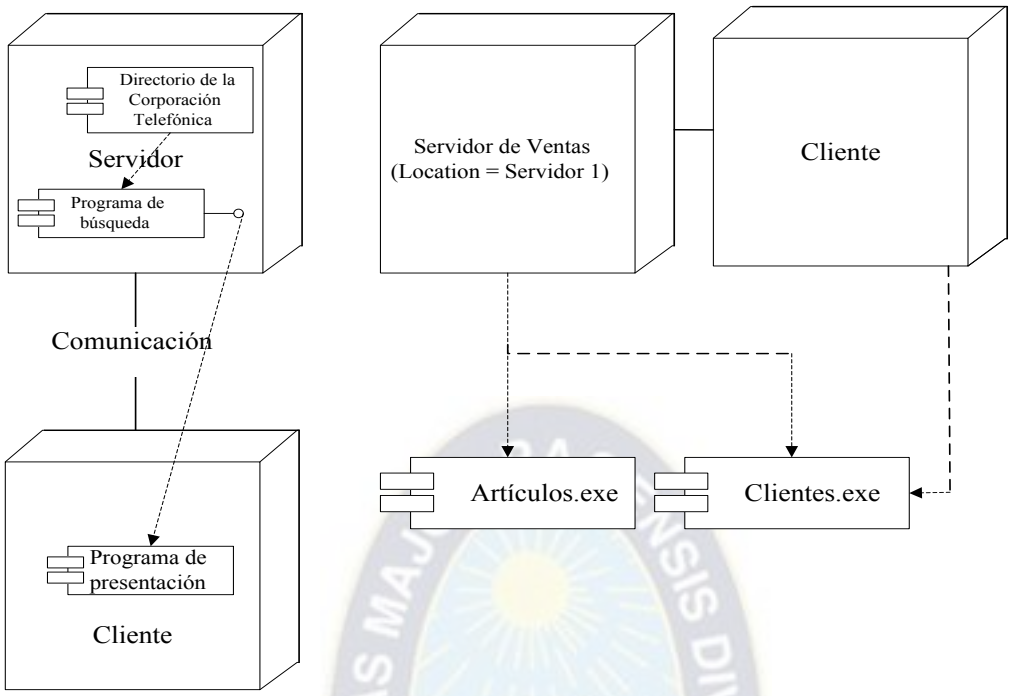


Fig. 15, Diagrama de despliegue.

Fuente; Aprendiendo UML, Joseph Schmuller *Prentice Hall*

INGENIERIA DE DISEÑO

3 INGENIERIA DE DISEÑO

3.1 Especificaciones de diseño

Tabla 1, Objetivos para el diseño

OBJETIVOS	ACCIONES	FUNDAMENTO TEORICO	INSTRUMENTO
Elegir una aplicación de software, que realice tareas del Protocolo Simple de Administración de Red (SNMP), y realizar su revisión bibliográfica, acerca de su funcionalidad, configuración y capacidades	Investigar y realizar estudios de prueba experimental, con el Sistema *NetView,* para determinar las posibles dificultades que esté presente, o influya en el rendimiento de la red. Interpretar los datos que recolecta de la red	Revisar en la guía de usuario del Sistema *NetView,* todas las secciones relacionadas a la administración, configuración, etc.	Realizar pruebas experimentales, con la aplicación de software *NetView,* en una red de datos
Investigar acerca de las capacidades de gestión y manejo, que ofrecen los diferentes servidores de Base de Datos Relacional (RDB), para elegir una, que este acorde a las necesidades del proyecto	Elegir una RDB que permita la obtención de registros, de una base de datos externa, además que permita al administrador realizar tareas de gestión y manejo, desde un sitio Web	Revisar todas las secciones de manejo, administración, etc., acerca del servidor de datos SQL	Administrar una RBD (servidor SQL), usando sus aplicaciones como; procedimientos almacenados, desencadenadores, etc. Crear una instancia en la RDB, para el vaciado de datos desde *NetView*

OBJETIVOS	ACCIONES	FUNDAMENTO TEORICO	INSTRUMENTO
Diseñar un método de conexión e interacción, entre una Base de Datos Relacional y una página web	Elegir un lenguaje de programación, apto para este tipo de requerimiento, y un gestor de bases de datos	Lenguaje de programación PHP, que genera un sitio Web dinámico e interactivo con el usuario.	Notación UML Programación orientada a objetos Crear base de datos
Diseñar una Interfaz Gráfica, (GUI), usando una página web, para estas tareas; El administrador de la red, bajo un ID de usuario y contraseña acceda al Sistema Gestor, para crear, modificar y actualizar los datos de los formularios de los Usuarios, y sus Estaciones de trabajo que incluye; datos de usuario, de configuración, listas de software etc	Elaborar un informe, acerca de la ubicación de los dispositivos de red, en el área de la red local Determinar los datos que reunirán los formularios, la información que permitirá realizar una auditoría de todos los recursos de red	Análisis preliminar a los requerimientos Bases de datos Análisis y diseño del sistema	Notación UML Programación orientada a objetos Crear base de datos

OBJETIVOS	ACCIONES	FUNDAMENTO TEORICO	INSTRUMENTO
Realizar diagnósticos de estatus de conectividad de los dispositivos de red, en tiempo real, usando los comandos de diagnóstico de red, de los protocolos ICMP y SNMP. Diseñar una Interfaz Gráfica de Usuario que permita el despliegue grafico estadístico del tráfico de datos (recursos de red) diario, semanal y mensual en los dispositivos de red, que muestre el uso de los recursos en los dispositivos de red más importantes	Determinar los comandos de diagnóstico de red más importantes, que se utilizan para monitorear la conectividad en una red de datos Hallar las características funcionales, que determinan el uso de los recursos de red (ancho de banda, velocidad de la interfaz, etc.), en un dispositivo de red.	Investigar cuales son las tareas, de poleo (pregunta del estatus de conectividad de un dispositivo de red), en una red de datos Revisar la sección de administración y configuración de red, en la guía de usuario del Sistema *NetView* PHP que permita el despliegue grafico desde una base de datos SQL Revisar en la guía de usuario del servidor *NetView*, la sección administración de una red y administración y configuración de los MIB	Ejecutar los comandos que serán utilizados, como monitores del estatus de la red local Notación UML Programación orientada a objetos Crear base de datos Tareas de configuración del sistema *Netview,* relacionadas al área del entorno de la red local Configurar la información MIB, en los dispositivos de red más importantes, que hayan sido elegido Crear base de datos

3.2 Requerimientos para el diseño del Gestor de red

Los diferentes tipos de tráfico, que se graficara en el sistema Gestor de Red, llamados MIB, son los que describen el comportamiento de los dispositivos de red.

a) Rfc1213-mib-ii

Es un estándar, que describe un conjunto de tipos de datos pertenecientes al protocolo TCP/IP. Los objetos MIB organizados lógicamente, tienen un nombre llamado "ID del objeto". Y estos serán usados por la interfaz gráfica, para realizar tareas de reportes gráficos. Las características que presentan estos MIB se muestran en la siguiente tabla.

Tabla 2, Características de los objetos Mib que pertenecen a Mib II

Nombre del objeto	ID de objeto	Tipo de dato	Descripción
.iso.org.dod.internet.mgmt.mib-2.system	.1.3.6.1.2.1.1	Otros	Características del Hardware y software.
.iso.org.dod.internet.mgmt.mib-2.interfaces	.1.3.6.1.2.1.2	Capa Física del modelo TCP/IP	Características en la interfaz física. Ancho de banda, dir. física,
.iso.org.dod.internet.mgmt.mib-2.at	.1.3.6.1.2.1.3	Otros	Las equivalencias entre la dirección física y lógica
.iso.org.dod.internet.mgmt.mib-2.ip	.1.3.6.1.2.1.4	Protocolo Internet de la capa de Red del modelo TCP/IP	Características en la interfaz lógica. Dirección IP, paquetes recibidos

Nombre del objeto	ID de objeto	Tipo de dato	Descripción
.iso.org.dod.internet.mgmt.mib-2.icmp	.1.3.6.1.2.1.5	Protocolo de Mensajes de Control Internet en la capa de Red del modelo TCP/IP	Características de tráfico de mensajes. Enviados, recibidos, perdidos, etc.
.iso.org.dod.internet.mgmt.mib-2.tcp	.1.3.6.1.2.1.6	Protocolo de Control de Transmisión en la capa de transporte de TCP/IP	Características de tráfico de segmentos. Enviados, recibidos, perdidos, etc.
.iso.org.dod.internet.mgmt.mib-2.udp	.1.3.6.1.2.1.7	Protocolo de Datagrama de Usuario del modelo TCP/IP	Características de tráfico de datagramas enviados, recibidos
.iso.org.dod.internet.mgmt.mib-2.egp	.1.3.6.1.2.1.8	Otros	Características del trafico EGP[15]
.iso.org.dod.internet.mgmt.mib-2.transmission	.1.3.6.1.2.1.10	Otros	Estatus de información y control de variables para una colección *ethernet-like*[16] interfaces

[15]Border Gateway Protocol, protocolo que intercambia información de ruteo entre sistemas autónomos.

[16]Define una parte de la MIB para su uso con protocolos de gestión de red en la comunidad de Internet, para la gestión de Ethernet-como interfaces.

Nombre del objeto	ID de objeto	Tipo de dato	Descripción
.iso.org.dod.internet.mgmt.mib-2.snmp	.1.3.6.1.2.1.11	Protocolo Simple de Administración de Red del modelo TCP/IP	Características del tráfico SNMP. Mensajes enviados, recibidos, perdidos, etc.

En la columna Tipo de Dato de la tabla 3, existen los objetos MIB, que están clasificados como protocolos conocidos en el modelo TCP/IP, y los demás están clasificados como otros, ya que estos últimos no pertenecen al conjunto de capas, que tiene el modelo TCP/IP. El conjunto de datos u objetos MIB, será clasificado en siete grupos. De esta forma el usuario podrá elegir, uno de estos grupos cuando así lo requiera.

Tabla 3, Grupos de tipos de datos en el modelo TCP/IP y el modelo OSI

Grupo de objetos Mib	**Tipo de dato en el Modelo TCP/IP**	**Capas en el modelo OSI**
Grupo de Objetos Mib 1	Interface	Capa 1 y 2
Grupo de Objetos Mib 2	ICMP	Capa 3
Grupo de Objetos Mib 3	IP	Capa 3
Grupo de Objetos Mib 4	TCP	Capa 4
Grupo de Objetos Mib 5	UDP	Capa 4
Grupo de Objetos Mib 6	SNMP	Capa 4
Grupo de Objetos Mib 7	Otros	Otros

b) Base de datos *NetViewDB*

Las tablas *snmpCollect* y *miblookup* de la base de datos *NetViewDB,* es donde se guarda los siete grupos de objetos MIB.

- **Tabla para *snmpCollect***

- Objeto Mib. Es el tipo de dato que se captura, con una consulta SNMP
- Instancia. Un Mib (tipo de dato) posee más de una instancia. Las instancias describen los diferentes puertos que posee una interfaz de red
- Dirección IP. Dirección lógica del dispositivo monitoreado
- Tiempo de partida. El instante cuando parte la consulta SNMP del Mib
- Tiempo de llegada. El instante cuando llega la respuesta SNMP del Mib
- El valor. Es el tipo de dato Mib capturado bajo las circunstancias anteriores

- **Tabla *Miblookup***

- ID del Mib. Valor que describe al Mib en la tabla *snmpCollect*
- Nombre del Mib. Nombre especifico del Mib

Para realizar un diagrama gráfico, es necesario proyectar en la interfaz gráfica, una función matemática que describa el comportamiento de los datos en el eje del tiempo, bajo ciertos requerimientos. Estos requerimientos serán definidos por las columnas de las tablas, más los siete grupos de Mib definidos anteriormente. Es decir las columnas que participaran y ayudaran a describir esta función son:

- Tipo de dato. Determinado por la columna Objeto Mib, y los siete grupos de tipos de datos que pertenecen al estándar Rfc1213-mib-ii.
- Dirección de red del dispositivo. Determinado por la columna Dirección IP.

- Puerto de la interfaz. Determinado por la columna Instancia.
- Primer dato. Determinado por la columna Tiempo de partida.
- Último dato. Determinado por la columna Tiempo de llegada.
- Tipo de gráfico. Será graficado por el lenguaje de programación PHP.
- Posición de los datos en el eje del tiempo. Será realizado por PHP.

3.3 Ingeniería de requerimientos

El objetivo del proceso de la ingeniería de requisitos es, darle a todas las partes una explicación escrita del problema.

3.3.1 Actores

Debido a que existen dos usuarios es decir el administrador del sistema, y el usuario común que manipulan el sistema, se considerara al actor usuario común, como secundario y al administrador como primario.

a). Actor primario. Es el encargado de otorgar permisos, cuentas de usuario al usuario secundario, además él puede ingresar y manipular los formularios de administración de auditoria y finalmente él tiene la función de velar la seguridad del sistema.

b). Actor secundario. Solo tiene la atribución de consultar los reportes gráficos y realizar consultas de conexión en la interfaz gráfica.

3.3.2 Casos de uso del sistema

Caso de uso: Reporte grafico

Actor secundario: Usuario común.

Meta en el contexto: Establecer el sistema para monitorear los dispositivos de red, en forma de reportes gráficos.

Condiciones previas: El sistema ha sido diseñado, para clasificar diferentes funciones de reportes de gráfico.

Escenario:

- Usuario común observa la interfaz gráfica.
- Usuario común introduce la contraseña.
- El sistema despliega solo las opciones habilitadas para el usuario.
- Usuario común selecciona el departamento, en donde se encuentra el dispositivo de red
- Usuario común selecciona el dispositivo de red al que desea monitorear.
- El sistema despliega toda la información concerniente a sus características del dispositivo seleccionado.
- El usuario común elige el icono de “graficar el tráfico actual”, o “graficar el tráfico histórico”.
- El sistema despliega una ventana de visualización, con diferentes opciones para graficar, es decir: dirección IP, instancia, función MIB, cantidad de datos, tipo de gráfico, posición de los datos en el eje del tiempo.
- El sistema despliega la salida de un gráfico, con las características que se seleccionó, en el anterior punto.

Excepciones:

- La contraseñas son incorrectas o no se reconocen; véase el caso de uso:" administrar usuarios".
- El usuario común selecciona "conexiones actuales"; véase el caso de uso:" consultas de conexión".

Prioridad: Prioridad esencial, debe implementarse.

Disponible en: Primer incremento.

Frecuencia de uso: Muchas veces.

Canal hacia el actor: A través de un explorador web (*browser*), sobre una computadora personal (PC) y conexión a la red local, al sitio Web de la interfaz gráfica.

Actores secundarios: Administrador de sistemas

Canales hacia los actores secundarios:

Administrador de sistemas: sistema basado en PC.

Caso de uso: Administrar usuarios

Actor Primario: Administrador de sistema.

Meta en el contexto: Establecer el sistema para registrar datos de; cuentas de usuario, lugar de trabajo, datos personales de los usuarios que usan un dispositivo de red.

Condiciones previas: El sistema ha sido diseñado para, editar, los datos personales, y de seguridad del usuario

Escenario:

- El administrador entra al sitio Web.
- El administrador introduce su ID y la contraseña.
- El sistema despliega todos las opciones de las funciones mas importantes
- El administrador selecciona el departamento, en donde se encuentra el usuario de la red local.
- El administrador selecciona el usuario de la red local al que desea controlar sus registros.
- El sistema despliega toda la información personal, y de seguridad del usuario seleccionado.
- El administrador elige la opción “datos personales” o “formulario de seguridad”.
- El administrador elige la elige el icono “nuevo usuario”, “modificar” o “borrar”.
- El sistema despliega una ventana de visualización, con diferentes opciones para modificar, o crear una cuenta nueva, es decir: nombres, dirección, función que desempeña lugar de trabajo, ID de usuario y contraseña.
- El sistema despliega la salida con la nueva cuenta modificada, o creada con las características, que se seleccionó en el anterior punto.

Excepciones:

- El administrador selecciona “estaciones” en las funciones más importantes, véase el caso de uso:” administrar estaciones”.
- El administrador elije “estaciones”, y después “graficar el trafico actual” o” graficar el trafico histórico”, véase el caso de uso:” reporte grafico”.

- El administrador selecciona “estaciones” y después “conexiones actuales”; véase el caso de uso:” consultas de conexión.

Prioridad: Prioridad esencial, debe implementarse.

Disponible en: Primer incremento.

Frecuencia de uso: Poco frecuente.

Canal hacia el actor: A través de un browser basado en PC y conexión a la red local, al sitio Web

Actores secundarios: Ninguno

Canales hacia los actores Secundarios: Ninguno

Caso de uso: Administrar estaciones

Actor Primario: Administrador de sistema.

Meta en el contexto: Establecer el sistema para registrar detalles relativos a la configuración del servidor, y de la estación de trabajo, entre estos como listados de software, registros de mantenimiento, registros de seguridad.

Condiciones previas: El sistema ha sido diseñado para, editar las características de hardware y software en un dispositivo de red.

Escenario:

- El administrador entra al sitio Web.
- El administrador introduce su ID y la contraseña.

- El sistema despliega todas las opciones, de las funciones más importantes
- El administrador selecciona el departamento, en donde se encuentra el usuario de la red
- El administrador selecciona la estación de la red local, a la que desea controlar sus registros.
- El sistema despliega toda la información, acerca de las características y la configuración de hardware y software, de la estación de trabajo o servidor que fue seleccionado.
- El administrador elige la opción “características de las estaciones”, o “formulario de configuración”.
- El administrador la elige el icono “nueva estación”, “modificar” o “borrar”.
- El sistema despliega una ventana, con diferentes opciones para modificar o guardar una cuenta nueva, es decir: dispositivos removibles, dispositivos fijos, tarjetas periféricas, tarjetas de internase de red, sistemas operativos.
- El sistema despliega la salida con la nueva configuración modificada, o creada con las características que se seleccionó en el anterior punto.

Excepciones:

- El administrador selecciona “usuarios” en las funciones más importantes; véase el caso de uso:” administrar usuarios”.
- El administrador elije “estaciones”, y después “graficar el trafico actual” o “graficar el trafico histórico”, véase el caso de uso; “reporte grafico”.
- El administrador selecciona “estaciones” y después “conexiones actuales”; véase el caso de uso; “consultas de conexión”.

Prioridad: Prioridad esencial, debe implementarse.

Disponible en: Primer incremento.

Frecuencia de uso: Poco frecuente.

Canal hacia el actor: A través de un browser basado en PC, y conexión a la red local, al sitio Web de la interfaz gráfica.

Actores secundarios: Ninguno

Canales hacia los actores Secundarios; Ninguno

Caso de uso: Consultas de conexión

Actor secundario: usuario común.

Meta en el contexto: Establecer el sistema, para monitorear los dispositivos de red en tiempo real, los reportes de conexión, de acuerdo a los comandos de diagnóstico de red, que se utilizan para monitorear la conectividad en una red de datos.

Condiciones previas: El sistema ha sido diseñado, para reconocer funciones relacionadas a comandos de diagnóstico de red, para monitorear la conectividad de un dispositivo de red.

Escenario:

- Usuario común observa la interfaz gráfica.
- Usuario común introduce la contraseña.
- El sistema despliega solo las opciones más importantes habilitadas, para el usuario.
- Usuario común selecciona el departamento donde se encuentra el dispositivo de red

- Usuario común selecciona el dispositivo de red al que desea monitorear.
- El sistema despliega toda la información concerniente a sus características del dispositivo seleccionado.
- El usuario común elige el icono de “conexiones actuales”
- El sistema despliega una ventana de visualización, en donde existe un lugar donde introducir la dirección IP a consultar, y con dos opciones para monitorear es decir: “*ping*”, “*snmpWalk*”
- El sistema despliega la salida con una respuesta con todas las características de conectividad, que se eligió en el anterior punto.

Excepciones:

¿La contraseña es incorrecta?; véase el caso de uso: “administrar usuarios”.

El administrador selecciona “estaciones”, luego “graficar el trafico actual” o “graficar el trafico histórico”, véase el caso de uso: “reporte grafico”.

Prioridad: Prioridad moderada, se implementara después de las funciones básicas.

Disponible en: segundo incremento.

Frecuencia de uso: Muchas veces.

Canal hacia el actor: A través de un buscador web basado en PC, y conexión a la red local, al sitio Web de la interfaz gráfica.

Actores secundarios: Administrador de sistemas.

Canales hacia los actores secundarios:

Administrador de sistemas: sistema basado en PC.

3.4 Ingeniería de diseño

3.4.1 Diagramas de casos de uso

La vista de casos de uso, modela la funcionalidad del sistema, según lo perciben los usuarios externos llamados actores. Un caso de uso es una unidad coherente de funcionalidad, expresada como transacción entre los actores y el sistema.

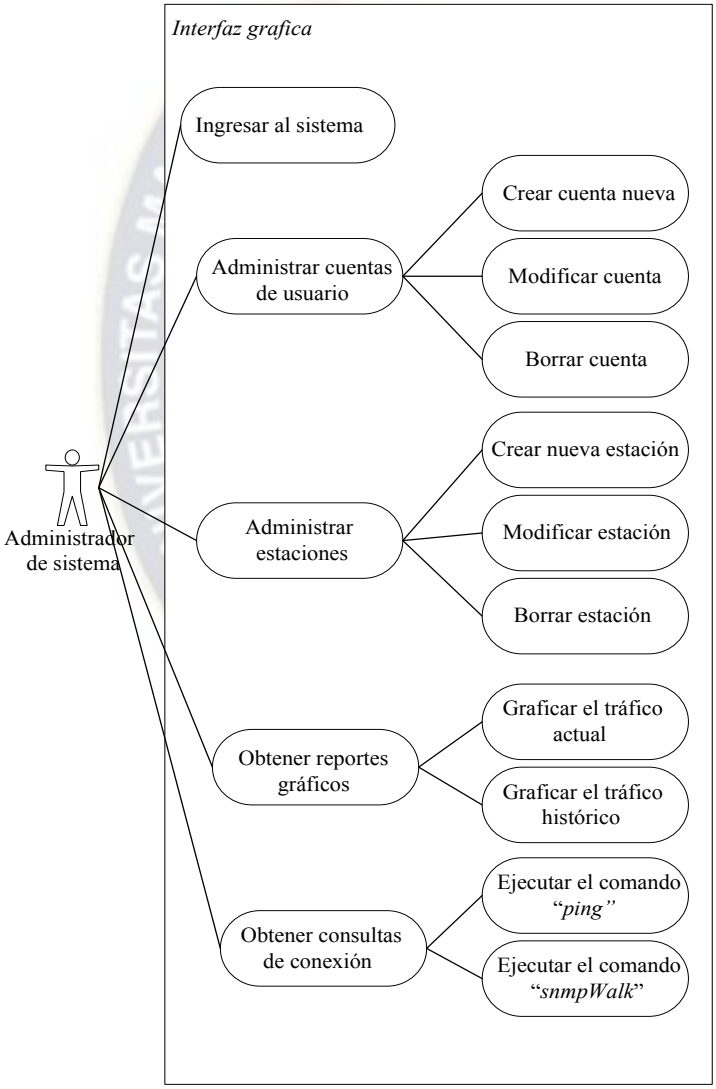


Fig. 16, Caso de uso del administrador de sistema.

Fuente: Elaboración propia

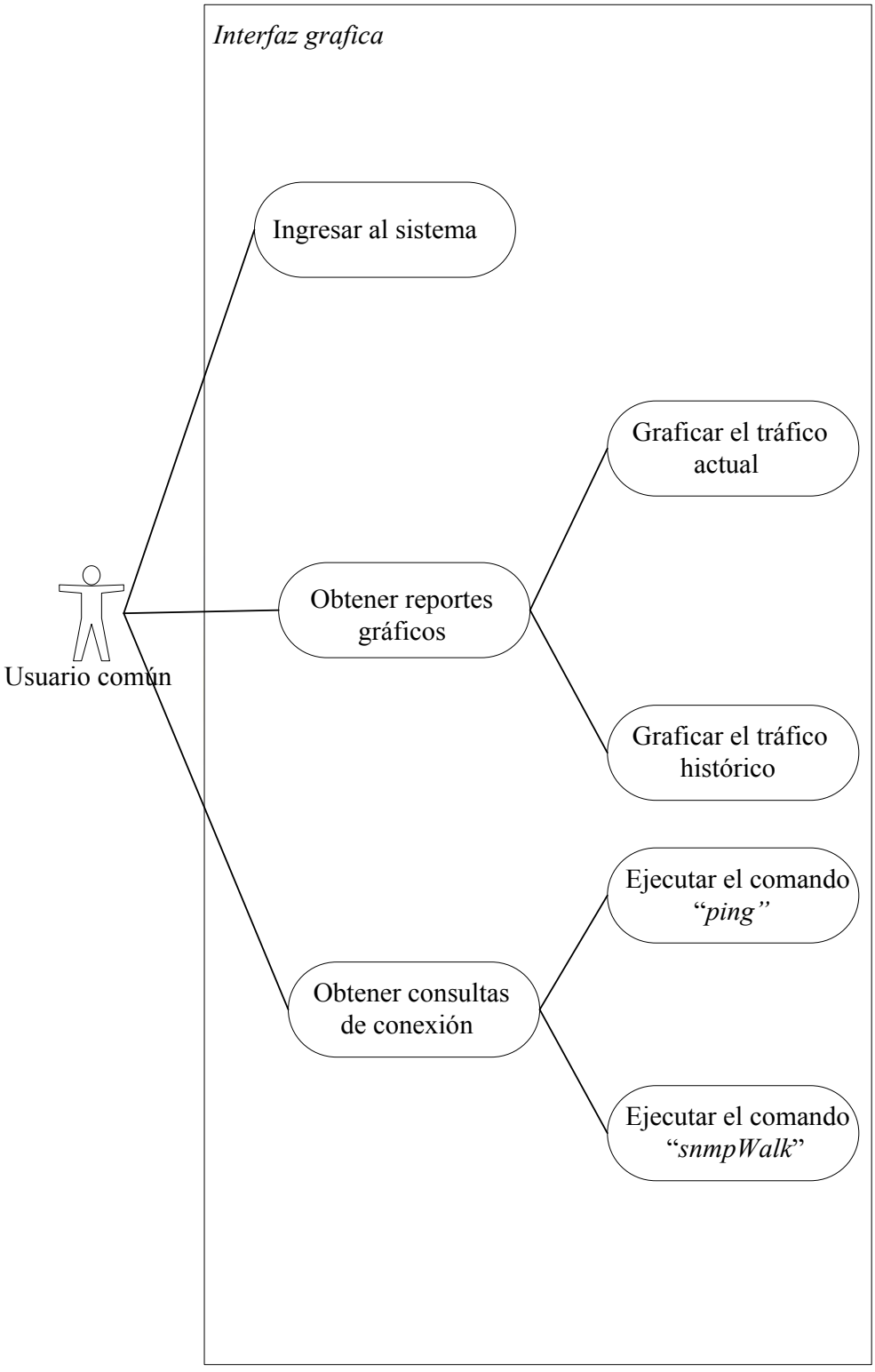


Fig. 17, Caso de uso del usuario común.

Fuente: Elaboración propia

3.4.2 Diagramas de colaboración

Un diagrama de colaboración muestra los roles en la interacción, en una disposición geométrica. Los mensajes se muestran como flechas, ligadas a las líneas de la relación, que conectan a los roles. La secuencia de mensajes, se indica con los números secuénciales que preceden a las descripciones del mensaje.

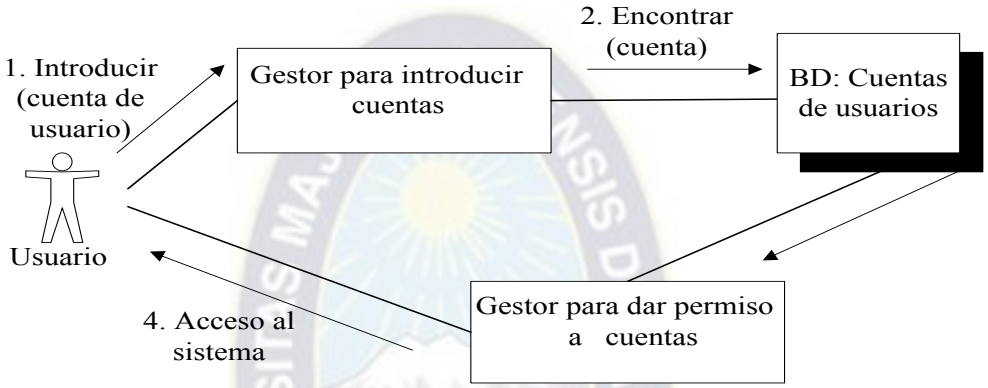


Fig. 18, Diagrama de colaboración "Ingresar al sistema".

Fuente: Elaboración propia

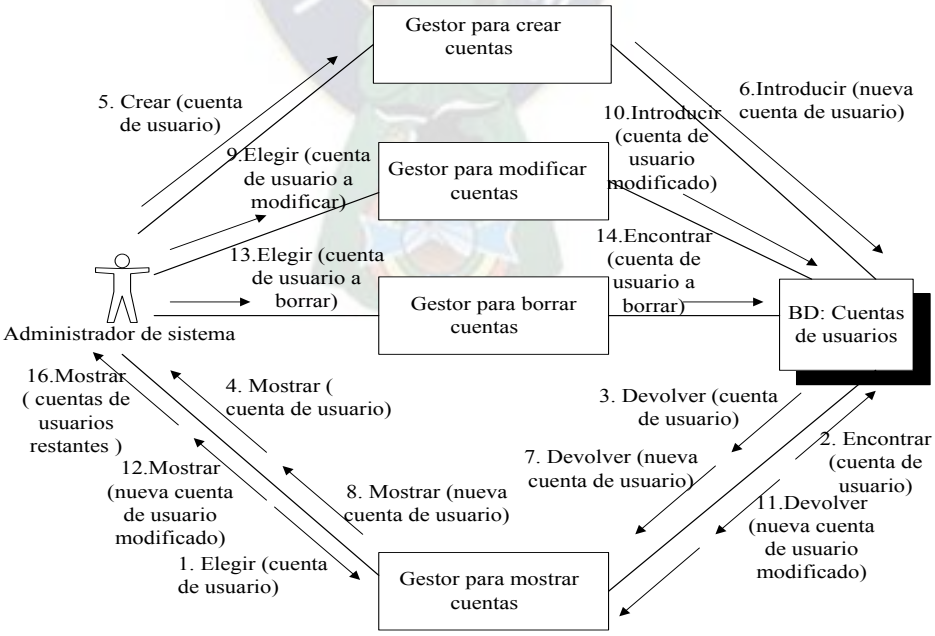


Fig. 19, Diagrama de colaboración "Administrar cuentas de usuario".

Fuente: Elaboración propia

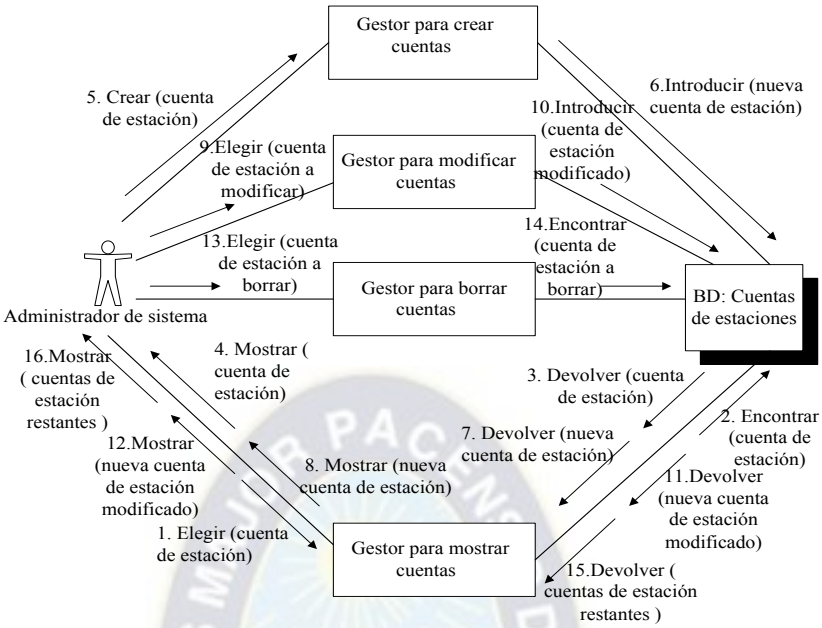


Fig. 20, Diagrama de colaboración “Administrar cuentas de estaciones”.

Fuente: Elaboración propia

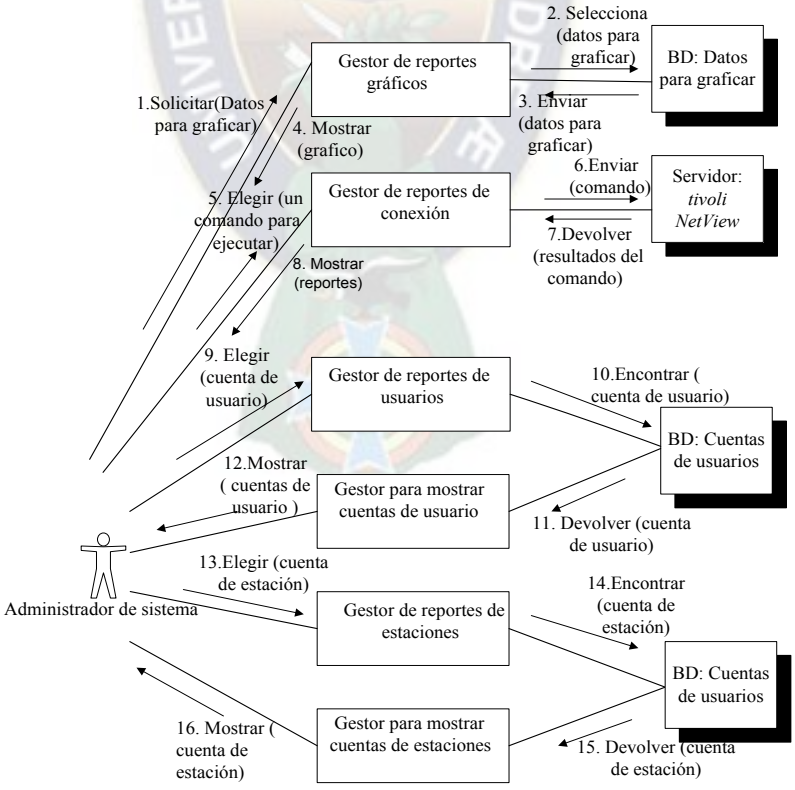


Fig. 21, Diagrama de colaboración “Mostrar reportes”.

Fuente: Elaboración propia

3.4.3 Clases de la interfaz grafica

Las clases son el centro alrededor del cual se organiza, la vista de clases. Las clases se dibujan como rectángulos. Las listas de atributos y de operaciones se muestran en compartimientos separados, las relaciones entre clases se dibujan como las líneas que conectan rectángulos de clases. El siguiente diagrama de clases, contiene el dominio: "Realizar operaciones sobre los usuarios y estaciones, y realizar reportes gráficos y de conexión".

3.4.3.1 Clases, sus atributos y operaciones

a) Clase Usuario.

- **Clase**. Hace referencia a todas las tareas, que se puede realizar sobre los registros de usuarios que operan la interfaz gráfica.
- **Atributos.** Los atributos para la clase Usuario son; nombres, apellidos, dirección, teléfonos, correo electrónico, función, lugar de trabajo, fecha de ingreso a la empresa, fecha de salida de la empresa, ID de usuario, contraseña de seguridad, fecha que expira la contraseña.
- **Operaciones**. Las operaciones que se realiza son; insertar nuevo usuario, modificar usuario y borrar usuario.

b) Clase Estación.

- **Clase.** Hace referencia a todas las tareas, que se puede realizar en las estaciones de trabajo y dispositivos de red que operan dentro de la red local.

- **Atributos.** Los atributos son; Nombre de la estación, localización física, DNI de usuario, dispositivos removibles, dispositivos fijos, tarjetas periféricas, tarjetas de interface de red, sistemas operativos.
- **Operaciones.** Las operaciones que se realiza son: insertar nueva estación, modificar estación y borrar estación.

c) Clase Administrador del sistema.

- **Clase.** Es el encargado de administrar el sistema, y tiene acceso a todo el sistema.
- **Atributos.** Los atributos son nombres, apellidos, dirección, teléfonos, etc.
- **Operaciones.** Existe una sola operación llamada; acceso a todo el sistema.

d) Clase Usuario común.

- **Clase**. Su acceso al sistema es limitado, ya que no goza de los privilegios como el administrador del sistema.
- **Atributos.** Los atributos son nombres, apellidos, dirección, teléfonos, etc.
- **Operaciones.** Existe una sola operación llamada; acceso limitado al sistema.

e) Clase Reporte grafico

- **Clase.** Hace referencia a los tipos de grafico estadístico, que se realiza con las diferentes funciones MIB, que el servidor RDB almacena.
- **Atributos.** Los atributos son: Dirección IP e instancias, Funciones MIB a graficar, cantidad de datos a graficar, tipo gráfico, posición de los datos en el eje del tiempo.
- **Operaciones**. Existe las siguientes operaciones, Graficar y Obtener los últimos datos actualizados.

f) Clase Reporte usuarios.

- **Clase.** Hace referencia a la exposición de los datos personales y de seguridad del usuario que permite la interactividad con los dispositivos de la red local.
- **Atributos.** Los atributos de la clase son: nombres, apellidos, dirección, teléfonos
- **Operaciones.** Las operaciones que se realiza son: Ubicación del lugar trabajo, usuarios que operan en un determinado lugar,

g) Clase Reporte estaciones.

- **Clase.** Hace referencia a la exposición de todas las características que involucra la configuración de hardware y software de las estaciones de trabajo en la red local.
- **Atributos.** Los atributos de la clase son: Nombre de la estación, localización geográfica, DNI de usuario, etc.
- **Operaciones.** Las operaciones que se realiza son: Ubicación geográfica del dispositivo en la red local, estaciones de trabajo o dispositivos de red que operan en un determinado lugar,

h) Clase Conexiones actuales.

- **Clase.** Hace referencia a la ejecución de comandos, dentro de una red de datos, para evaluar el estado actual del dispositivo.
- **Atributos.** Los atributos para la clase Conexiones actuales son: Dirección IP y número de peticiones para el comando *ping* y Dirección IP para el comando *snmpWalk.*
- **Operaciones.** Son: la ejecución del comando *ping* y *snmpWalk.*

3.4.3.2 Métodos de la interfaz grafica

Tabla 4, Clase Usuario.

TAREA	DATOS DE ENTRADA	DATOS DE SALIDA
Nuevo usuario	Nombres, apellidos, dirección, etc.	Se crea un nuevo registro de usuario.
Modificar usuario	nombres, apellidos, dirección	Se modifica el registro de usuario.
Borrar usuario	Nombres, apellidos, dirección	Se elimina todo el registro del usuario.
Cambiar de ventana	Cambiar	Sale del formulario

Tabla 5, Clase Estación.

TAREA	DATOS DE ENTRADA	DATOS DE SALIDA
Nueva estación	Nombre de la estación, localización física, etc.	Se crea un nuevo registro de estación.
Modificar estación	Nombre de la estación, localización física,	Se modifica el registro de estación.
Borrar estación	Nombre de la estación, localización física,	Se elimina todo el registro de la estación
Cambiar de ventana	Cambiar	Sale del formulario

Tabla 6, Clase Administrador del sistema.

TAREA	DATOS DE ENTRADA	DATOS DE SALIDA
Acceso a todo el sistema.	nombres, apellidos, dirección, etc.	Se tiene un acceso completo a todo el sistema
Cambiar de ventana	Cambiar	Sale de la ventana

Tabla 7, Clase Usuario común.

TAREA	DATOS DE ENTRADA	DATOS DE SALIDA
Acceso limitado al sistema.	nombres, apellidos, dirección, etc.	Se tiene un acceso limitado al sistema
Cambiar de ventana	Cambiar	Sale de la ventana

Tabla 8, Clase Reporte gráfico.

TAREA	DATOS DE ENTRADA	DATOS DE SALIDA
Obtener últimos datos actualizados	Dirección IP e instancias, Funciones MIB a graficar, etc.	Se actualiza los últimos datos que se almacena en el servidor RDB
Cambiar de ventana	Cambiar	Sale de la ventana

Tabla 9, Clase reporte Usuarios.

TAREA	DATOS DE ENTRADA	DATOS DE SALIDA
Ubicación del lugar trabajo	Nombre del departamento donde trabaja.	Se obtiene todos los lugares de trabajo clasificados por departamentos.
Todos los usuarios que operan en un determinado lugar	DNI del usuario.	Se obtiene todos los usuarios que operan en un área de trabajo.
Cambiar de ventana	Cambiar	Sale de la ventana

Tabla 10, Clase reporte Estaciones.

TAREA	DATOS DE ENTRADA	DATOS DE SALIDA
Ubicación del lugar donde trabaja la estación	Nombre del departamento donde trabaja la estación.	Se obtiene todos los lugares de trabajo clasificados por departamentos.
Todos las estaciones que operan en un determinado lugar	Nombre de la unidad.	Se obtiene todas las estaciones que operan en un determinado lugar de trabajo o departamento
Cambiar de ventana	Cambiar	Sale de la ventana

Tabla 11, Clase Conexiones actuales.

TAREA	DATOS DE ENTRADA	DATOS DE SALIDA
Ejecución del comando ping	Dirección IP y número de peticiones.	Se obtiene los resultados inmediatos del comando ejecutado.
Ejecución del comando *snmpWalk*	Dirección IP	Se obtiene los resultados inmediatos del comando ejecutado.
Cambiar de ventana	Cambiar	Sale de la ventana

3.4.3.3 Diagrama de clases de la Interfaz Gráfica de Usuario

Es la relación entre clases y como comparten sus métodos y operaciones.

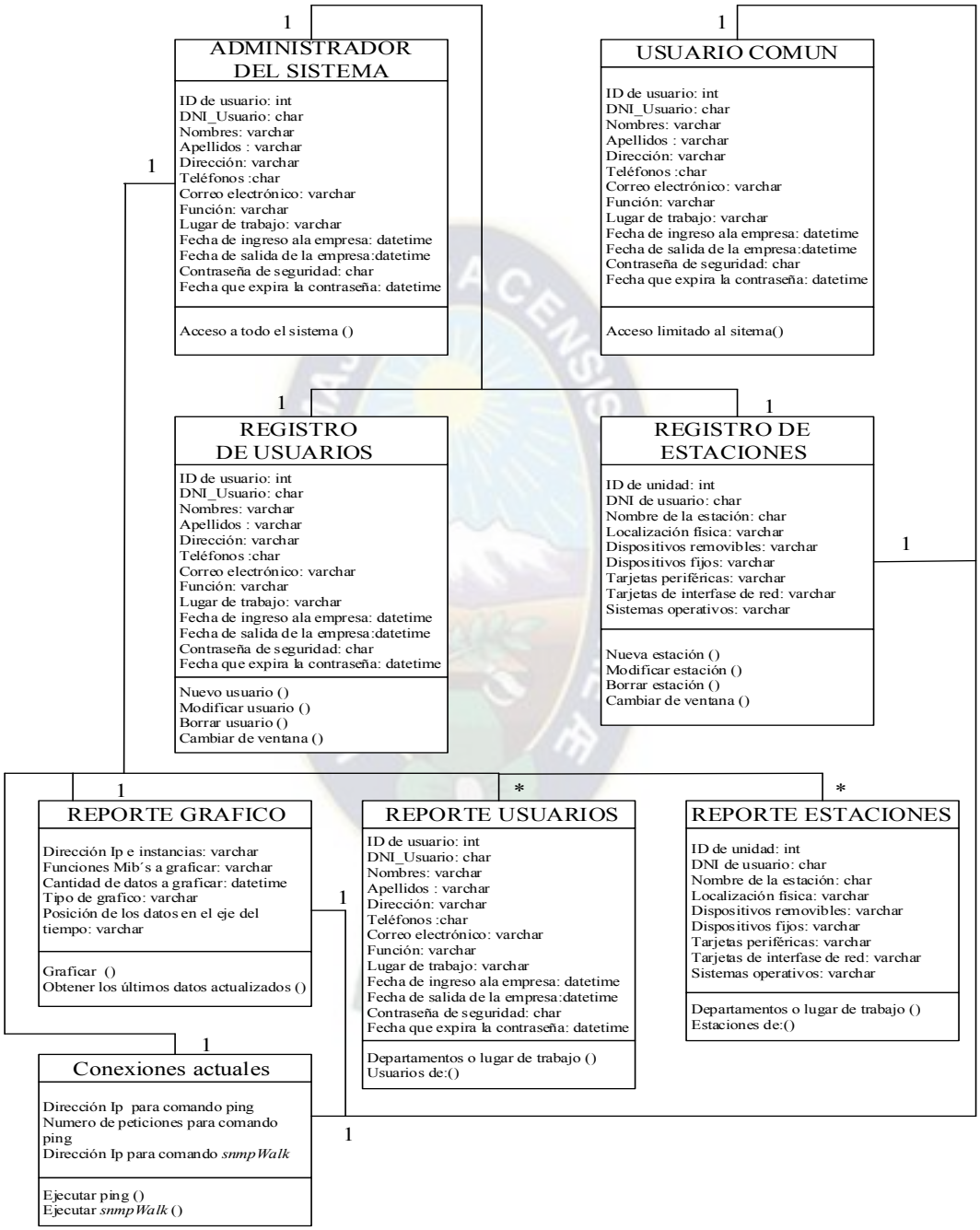


Fig. 22, Diagrama de clases

Fuente: Elaboración propia

3.4.3.4 Diagrama de clases para la base de datos

La base de datos del sistema posee dos partes, una exclusivamente para administrar cuentas de usuarios y estaciones, y la otra para almacenar datos de las funciones MIB, que posteriormente se graficara.

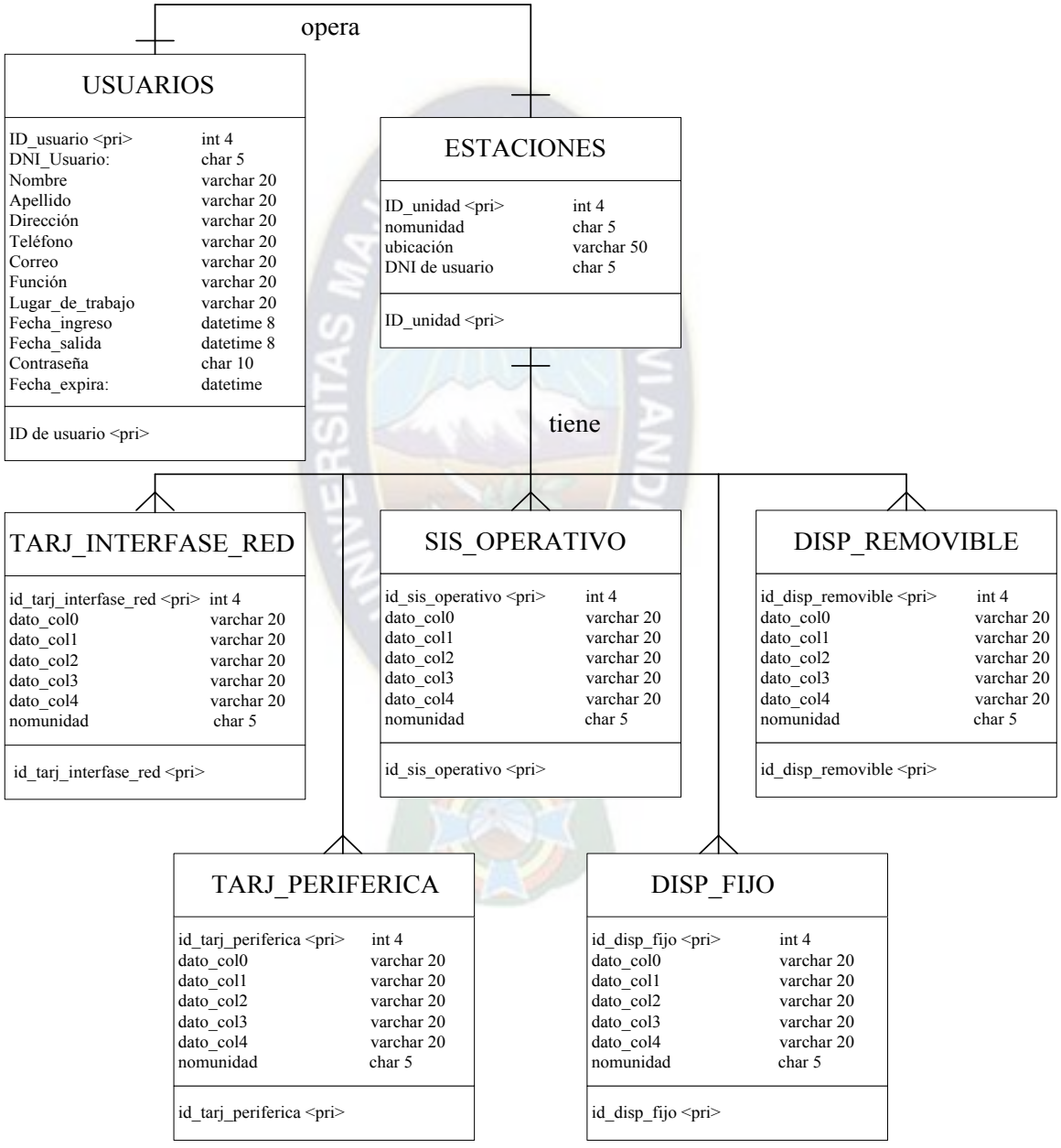


Fig. 23, Diagrama de base de datos usuarios y estaciones.

Fuente: Elaboración propia

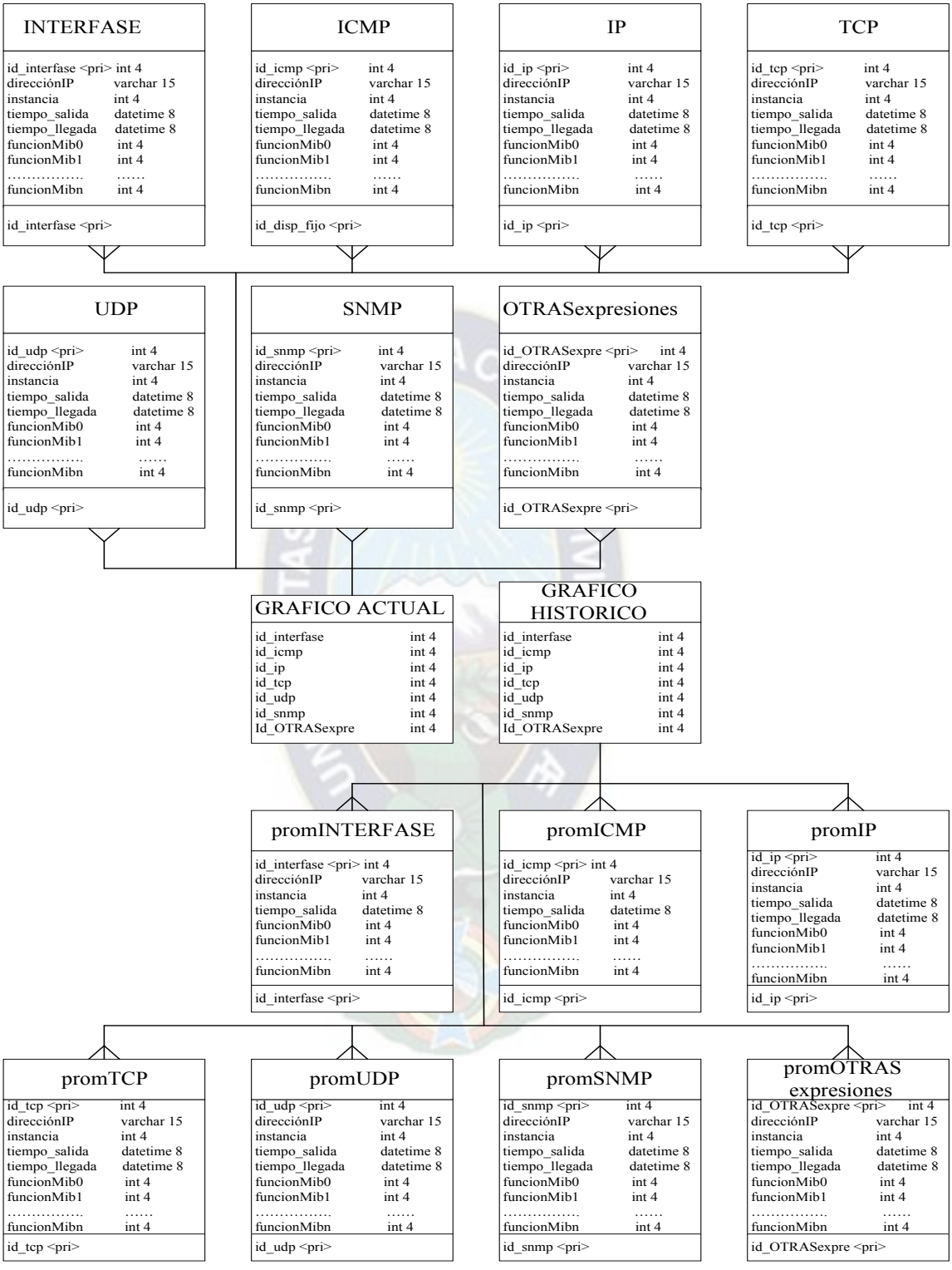


Fig. 24, Diagrama de base de datos gráficos.

Fuente: Elaboración propia

3.4.4 Diagrama de paquetes

La descripción completa del Sistema Gestor, se puede ver representada mediante un modelo, que abarca un conjunto de elementos tales como clases y casos de usos. EL contenido del modelo se representa mediante paquetes, además estos paquetes (llamado paquete raíz), contienen en su interior otros paquetes, que representan subsistemas que son una porción del sistema principal, con una interfaz que puede ser implementado como un componente destino, tal como se muestra en la figura 3.11.

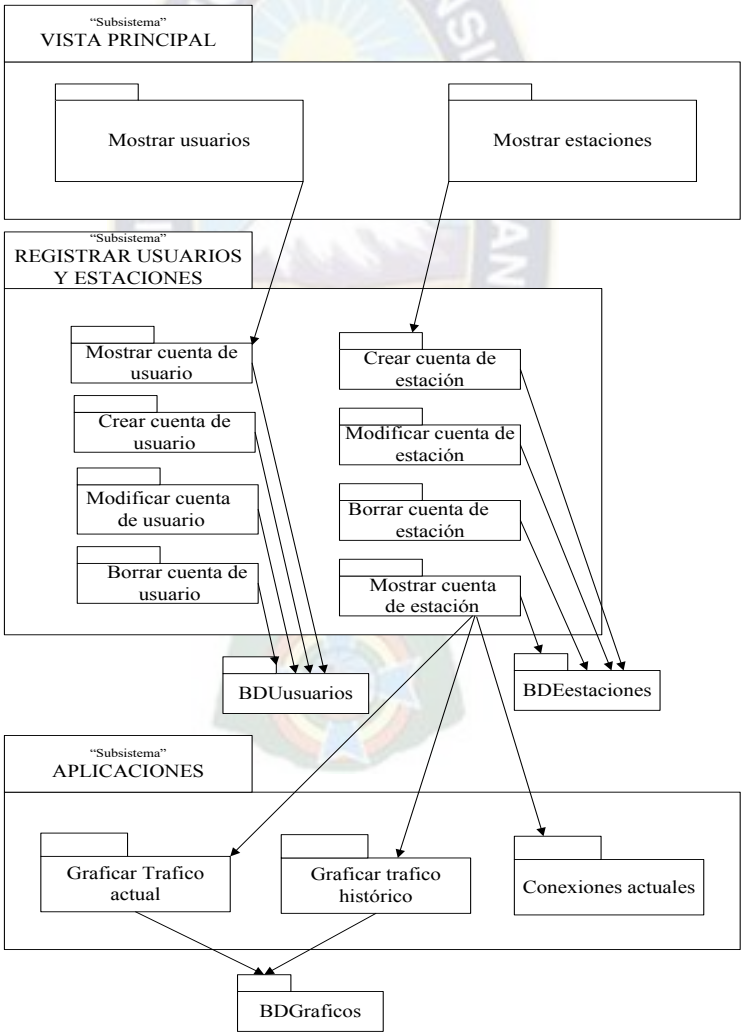


Fig. 25, Diagrama de paquetes.

Fuente: Elaboración propia

3.4.5 Diagrama de sistemas distribuidos

La capa de software de aplicación y la capa de software del sistema, forman la base sobre la cual se sostiene todo el funcionamiento del Sistema.

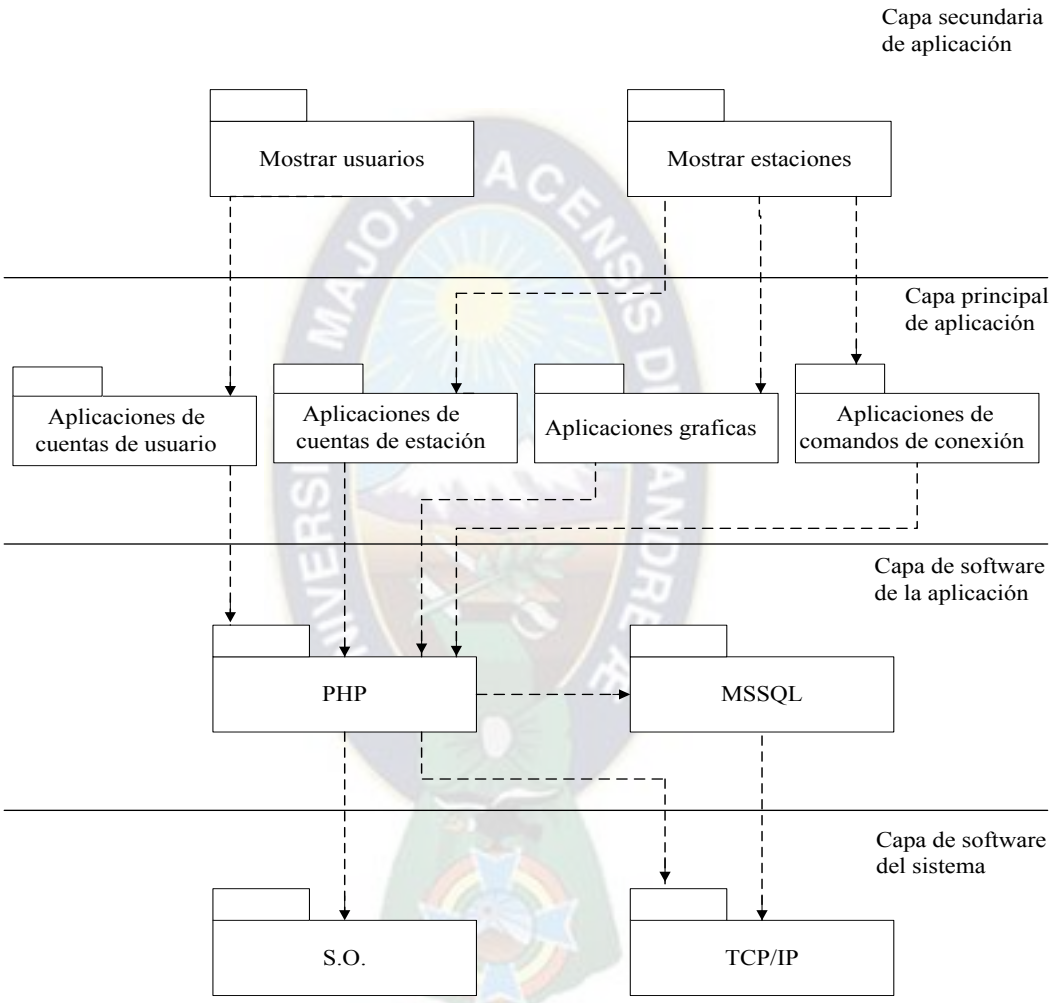


Fig. 26, Diagrama de sistemas distribuidos.

Fuente: Elaboración propia

3.4.6 Diseño de la interfaz de usuario

3.4.6.1 Diagrama de componentes y de interfaz

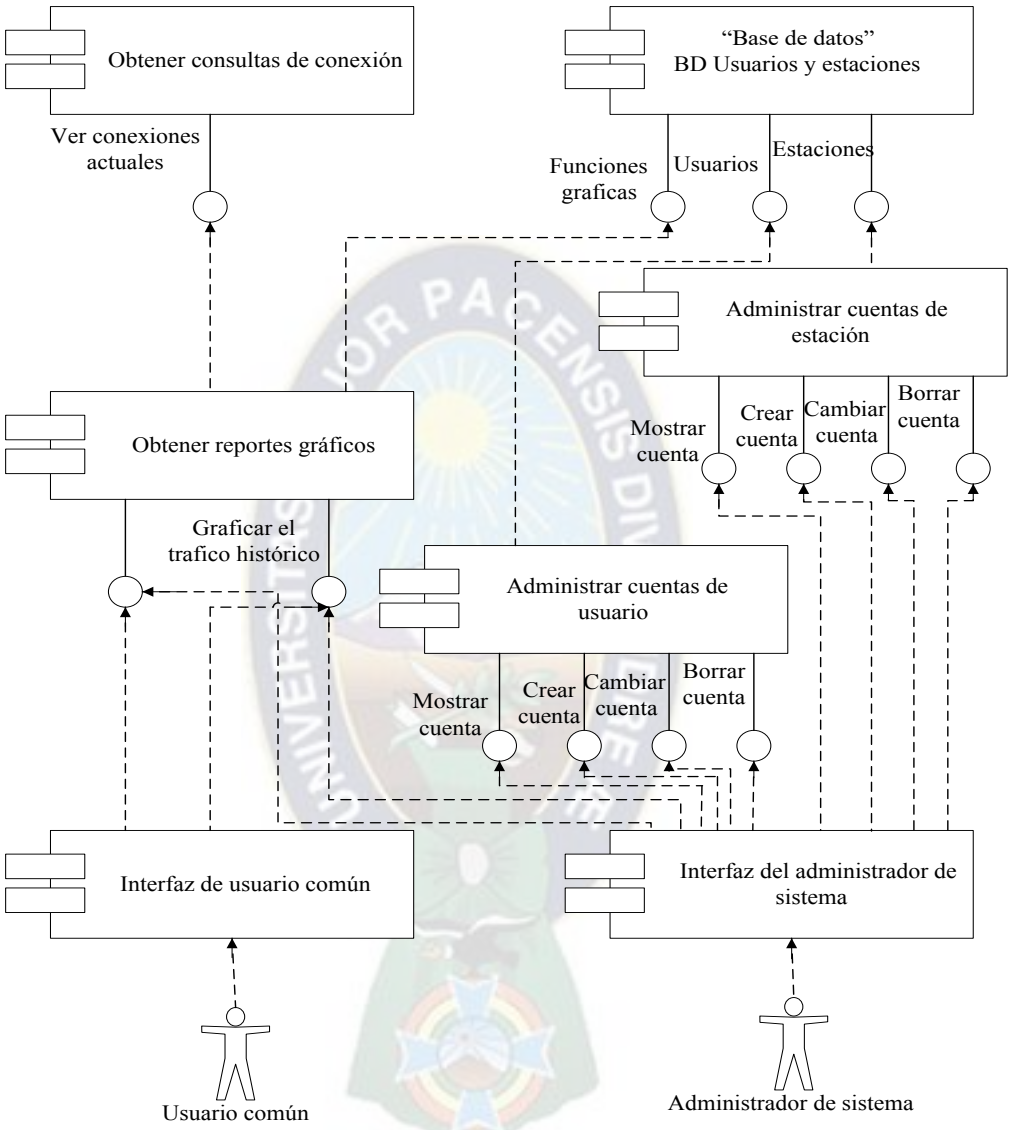


Fig. 27, Diagrama de interfaz de usuario.

Fuente: Elaboración propia

Un componente es una unidad de implementación, con interfaces definidas, pensada para ser utilizada como parte reemplazable del sistema. El circulo pequeño indica la interfaz que conecta la Base de Datos Relacional y la página web (interfaz gráfica de

usuario común o administrador del sistema) y la línea sólida que va desde un componente a un interfaz, indica que el componente proporciona los servicios de la interfaz, Una flecha de guiones indica, que el componente requiere los servicios proporcionados por la interfaz.

3.4.7 Diagrama de despliegue

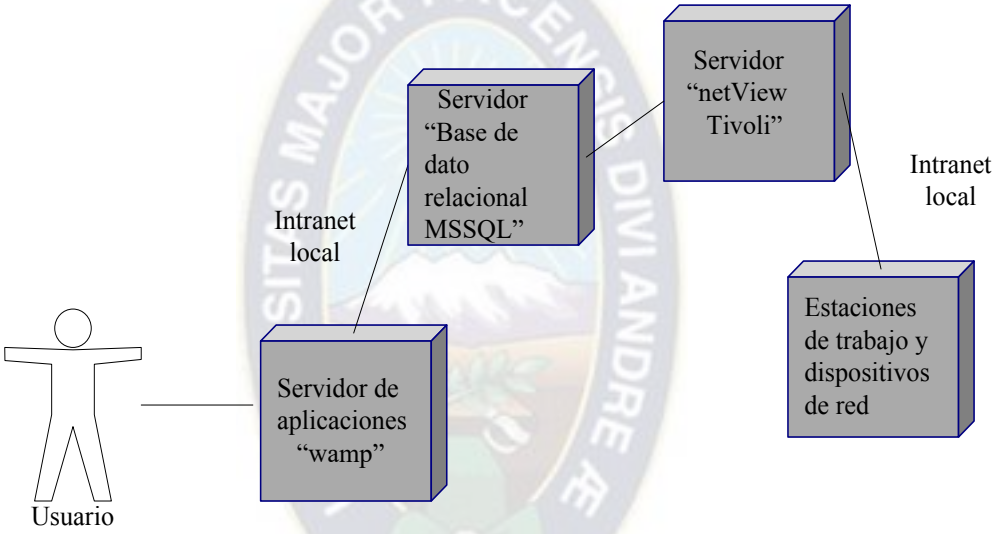


Fig. 28, Diagrama de despliegue del sistema.

Fuente: Elaboración propia

El Sistema Gestor, usa tres servidores; un servidor *wamp* para la aplicación PHP, otro de base de datos SQL, para almacenar los datos de usuarios, estaciones y funciones a graficar, y un último servidor *netViewDB,* encargado de recolectar datos o funciones Mib.

COSTOS DE IMPLEMENTACION

4 COSTOS DE IMPLEMENTACION

4.1 Requerimientos

La implementación del Sistema Gestión de Red no exige realizar sobre él, un análisis exigente de viabilidad, ya que esta es una herramienta de software, donde la inversión económica, no es tan significante, y no influye en el desarrollo ni la culminación del proyecto. Sin embargo se puede realizar un análisis, desde el punto de vista de la implementación física del SGR, al incorporar un servidor más en la Red local.

4.2 Requerimientos de Hardware

Las características requeridas y de implementación para el Sistema de Gestión, se muestran en la tabla 12. Estas son expuestas, de acuerdo a las características de hardware, que requiere el Sistema SNMP de *NetView* para su funcionamiento, en su guía de usuario.

Tabla 12, Características de requerimiento del ordenador donde funciona el SGR

Detalle	Características de requerimiento	Características de implementación
Pentium	IV	IV
Procesador	4.00 GHz	2.00 GHz
Disco Duro	1 TB	100 GB
Memoria RAM	2 GB	1 GB
Tarjeta de Red	*Fast Ethernet* PCI	*Fast Ethernet* PCI
Unidad de DVD-ROM	12X	8X

4.3 Requerimientos de Software.

De acuerdo al software que requiere el sistema *NetView*, en la tabla 4.2, se muestra sus características.

Tabla 13, Características de Software que se requiere para el servidor *NetView*

Detalle	Características
Sistema Operativo	Microsoft Windows 7
Protocolo instalado y configurado	TCP/IP
Servicio instalado y Configurado	SNMP Protocolo Simple de Administración de Red
Origen de Base de datos	SQL Server 2012
Internet Explorer	V8.0 o versiones superiores

4.4 Gasto económico

4.4.1 Costos de implementación.

El análisis de costo en la implementación, está relacionado a los costos de inversión, en el mantenimiento del *software* y *hardware*, este último para dos casos diferentes; uno para la implementación de hardware adquiriendo un nuevo equipo, y el otro implementado en un equipo libre de la red local.

Tabla 14, Costo de mantenimiento anual del sistema

Costos iníciales	Costo final y Tasa de cambios
Duración: 12 meses/año	TC: 6,96 Bs/$us
Costo por 24[horas]: 7,25 $us	31 dia/mes * 12 mes/año * 7,25 $us/dia =
	Costo mantenimiento anual: 2697,00 $us

Tabla 15, Características del servidor para la implementación del Sistema

Detalle	Características
Nombre del servidor	Bundle HP Proliant ML350 G5 Xeon Quad Core E5420 *Adicional 3 Discos Duros HP
Tarjeta madre	SKU# 458242-001/375861-B21FT
Procesador	Intel Xeon QuadCore E54202.5 GHz
3 Discos Duros	HP 72GB
Memoria RAM	2 GB de mem. DDR2 SDRAM, ampliable
Tarjeta de Red	10/100 Mbps Ethernet PCI Adapter
Unidad de CD- ROM	16x Speed DVD-ROM
Precio Total 15910,00 $us	

Tabla 16, Costo del software requerido para la implementación del Sistema

Detalle	Características	Precio en Dólares ($us)
Sistema Operativo	Windows 7	199.99
Gestor de Base de Datos	MSSQL Server 2012	899.99
Servidor Internet Explorer	WAMP Server 2.5	libre
Lenguaje de programación	PHP 7.0	libre
Administrador de Red	NetView 7.1.2	0.00
Total		1099.98 $us

Tabla 17, Costos de la inversión del Sistema

Detalle	Costos anual en bolivianos (Bs.)	Costos en Dólares ($us.)
Costo del desarrollo del sistema realizado por un analista de sistema	7000,00([Bs]/mes)*12 mes = 84000,00	1011,56([$us]/mes)*12 mes = 12069,00
Material de escritorio (Papel, tinta impresora)	600,00	86,20
Costo de investigación (Internet, fotocopias etc.)	160,00*12 mes=1920	256.46
Total Anual	86520,00 bs.	12411,46 $us

Tabla 18, Costos Finales

Detalle	Costos en Bolivianos (Bs.)	Costos en Dólares ($us.)
Software y hardware	118559.56 bs	(1099.98 + 15910,00)$us = 17009.98 $us
Mantenimiento	18771,12 bs anual	2697,00 $us anual
Inversión en el sistema	86520,00 bs anual	12411,46 $us

4.4.2 Beneficios de implementación

El presente proyecto requerirá una inversión de 86520,00 bs. Aunque no se obtendrá los beneficios en resultados económicos,

Los beneficios se pueden calificar, de acuerdo a los resultados obtenidos, al revisar los objetivos vencidos en el desarrollo del proyecto. Solo se podrá obtener resultados favorables si se enfoca los resultados desde el punto de vista del objetivo general.

CONCLUSIONES
Y
OBSERVACIONES

5 CONCLUSIONES Y OBSERVACIONES

5.1 Estudios de prueba experimental sobre el software *NetView*.

Después de realizar la primera prueba, sobre 80 dispositivos de red, en la sala de red de un café internet, se observó la siguiente cantidad de mensajes ICMP en la interfaz del administrador *NetView,* un total de 3.2 mensajes recuperados y 0.2 mensajes emitidos a la red, dando una relación de:

3.2/0.2=16 …….relación del tráfico de entrada y salida en la interfaz del sistema *NetView*.

Una tecnología de Ethernet como 10Base5 soporta una velocidad de 10 Mbps. Un paquete de datos IP, en su capacidad máxima podría contener 64 kbits y un paquete de ICMP corresponde al 0.2 de cantidad de un paquete IP. Si en su máxima cantidad de recepción se pudo observar 3.2 de mensajes ICMP´s, por lo tanto realizando la siguiente operacion:

(3,2*0,2)[paquetes]=0,64[paquetes]; 0,64[paquetes]*64[kbits]/1[paquetes] =40,96[kbits];

100%*40,96[kbits]/64[kbits]=64%, lo que indica que se está usando, el 64% de la capacidad, que puede enviar un paquete IP, sin considerar la verdadera capacidad de la red, que es de 10Mbps.

Por otra parte se configura el administrador *NetView,* para realizar consultas ICMP, cada 2 minutos, que determinaran el estatus de cada host. Para realizar consultas SNMP (GET), cada 2 horas que descubrirán nuevos nodos y consultas MIB´s cada 1 día.

Debido a que se realizó la operación *unmanage* (tarea de configuración, que desconoce un monitoreo constante hacia los nodos), Se pudo observar que el flujo de tráfico no se vio afectada.

Para reducir aún más el tráfico, se debe configurar la funcionalidad de Agente del protocolo SNMP, solamente sobre aquellos dispositivos considerados importantes, donde fluye la información de todos los usuarios, como es el enrutador (*Router)*.

5.2 Eventos y la Topología de red

Acerca de la funcionalidad, configuración y capacidades avanzadas que presta el sistema *NetView,* El sistema ofrece un acceso abierto y libre a su base de datos, lo que permite construir aplicaciones personalizadas que no posee.

La cantidad de paquetes de entrada descartados (*ifInDiscards*), paquetes de salida descartadas (*ifOutDiscards*), en relación al volumen total de tráfico de entrada y salida correspondiente, pueden ser utilizados para identificar problemas en la red. Para analizar el rendimiento de la red, se debe recolectar los datos de las variables *ifInDiscards* y *ifOutDiscards,* sobre aquellas interfaces que tengan la cantidad más alta de paquetes descartados (tráfico de congestión), y hacer que la función *snmpCollect* genere un evento (*Trap*), donde el número de estos paquetes descartados excedan un umbral dado.

Este *Trap,* puede ser utilizado para realizar una investigación sobre los datos recolectados, en las diferentes tablas de la Base de Datos del Sistema Gestor, por ejemplo para realizar una búsqueda, de aquellos dispositivos de red, que devuelva la siguiente información:

- El host que ha generado este tipo de *Trap,* desde la Tabla Sistema Evento.
- Identificar la hora y la fecha en el cual se generó este evento, desde de la Tabla *snmpCollect*, para saber los horarios de más alto tráfico.

- La identificación de cada uno de estos hosts, que genera este tipo de tráfico desde la Tabla Topología de red.

5.3 Acerca de la funcionalidad del Sistema Gestor

En la estructura de árbol de MIB-II, que representa el estándar para redes TCP/IP, se tiene una diversidad de variables relacionados al trágico de datos, entre las más principales tenemos, Interfaz, IP, ICMP, TCP, UDP, SNMP. Además SNMP tiene la capacidad de manejar estándares de información sobre dispositivos de red de diferentes fabricantes.

Existen diferentes tareas que el Sistema Gestor tendrá que implementar para ser independiente de un programa administrador que obtiene datos MIB. Sus características son:

- Recolectar más de una sola variable MIB
- Especificarse intervalos de poleo
- Colectar datos para una sola variable de MIB
- Especificar la recolección para una o varias interfaces
- Almacenar y revisar los umbrales de datos establecidos
- Generar eventos
- Mostrar los reportes en formato tabla
- Salvar las salidas de los reportes en un archivo por defecto, además de la que ya existe en el servidor SQL

5.4 Acerca de la Base de Datos Relacional SQL Server

Para diferentes instancias de gestión y manejo los datos que obtiene el servidor SQL*Server*, gracias a la ejecución del comando *snmpcollect,* es necesario relacionarlos entre las tablas *MibLookUp* y *SnmpCollectData*, ya que cada variable MIB ubicada en la tabla *MibLookUp*, está relacionada con in ID que lo identifica en la tabla *SnmpCollectData*. El manejo de una base de datos como SQLServer, resultó ser flexible para ser usada por estas aplicaciones, lo que facilita la administración y personalización propia de estos reportes de datos.

5.5 Respecto al diseño de la Interfaz Gráfica de Usuario (GUI)

Usando una página web, que permita el despliegue grafico estadístico del tráfico de datos diario, semanal y mensual en los dispositivos de red, se debe recalcar la confiabilidad que la herramienta PHP ofrece, ya que puede autentificar al usuario, mediante funciones que posee PHP, para páginas que se autodestruyen cuando se cierra la ventana. Además el Sistema de Gestor de Red está protegido ante posibles modificaciones, de las funciones de base de datos (Procedimientos almacenados SQL), ya que estos se ejecuta en un servidor externo, y no así en el host donde se presta el servicio Web, y precisamente esta es una de las ventajas principales de PHP.

5.6 Respecto a las capacidades de gestión y manejo, de la Base de Datos Relacional

Se debe mencionar que esta etapa dificulto el avance programado del proyecto, al tratar de realizar la tarea copiar o vaciar datos de la base de datos SNMP de *NetView*, ya que implicó anular los desencadenadores de origen, es decir desde la base de datos

NetView. Sin embargo se logró vencer el objetivo, aunque aún falta realizar pruebas de vaciado para grandes cantidades de datos.

El uso del diseñador de DTS´s (Servicio de Transformación de Datos), la combinación de tareas que ejecuta el asistente para la exportación e importación de datos, y la programación en la ejecución de paquetes DTS ayudo finalmente a la ejecución del objetivo de este trabajo.

Inicialmente, para desarrollar la página Web, usando el programa PHP, se usó la Base de Datos Relacional *mySQL,* pero debido a su baja seguridad se tuvo que emigrar al uso del Servidor de datos *SQL server*. Además gracias a la utilidad que posee PHP, con las funciones PEAR TEST, se facilita las conexiones desde la página Web, hacia la Base de Datos y así se logra, el acceso a las tablas almacenadas en el servidor SQL de Microsoft.

5.7 Diseño final de la interfaz gráfica

Para la documentación y actualización de todos los recursos de la red en una base de datos, el Sistema de Gestion se sujeta bajo tres subsistemas; *Netview*, RDB-SQLServer y *Html-Php*, Es en el subsistema RDB-SQLServer, donde se desarrolló la mayor parte de la ingeniería de diseño. En este último subsistema se realiza la administración de la base de datos, gracias a procedimientos almacenados bajo el estándar UML, para finalmente realizar y ejecutar consultas, desde el subsistema *Html-Php.*

La aplicación "Conexiones de red" bajo el comando "*snmpWalk*", tiene la ventaja de ser ejecutada, desde cualquier dispositivo en la red local, gracias a que la aplicación se ejecuta en el servidor y no en el navegador. Recoge las peticiones del usuario y genera una página Web personalizada.

La velocidad de respuesta, a las tareas que ofrece el Sistema Gestor, está limitada a la cantidad de datos, que el sistema analiza y evalúa desde RDB-SQL y la cantidad de vínculos, que tiene la página Web. Aunque sea inevitable la disminución de la velocidad de respuesta, después de la ejecución de tareas o cuando la cantidad de datos de la base de datos haya aumentado, el Sistema Gestor reduce al mínimo otros factores que influyan en el retardo, como es el caso de la cantidad de vínculos de páginas web que se abren en un instante, de tal manera que cuando se realiza una conexión de vínculo, esta conexión no pueda reproducirse en otra página, es así que en la aplicación de la interfaz gráfica, se puede ver conexiones de vínculos hacia una sola página.

6 BIBLIOGRAFIA

[USER´S GUIDE, 2006] USER´S GUIDE, NetView for Windows, Tivoli Systems an IBM Company version 6.0, 2006.

[PRESSMAN, 2012] PH. D. ROGER S. PRESSMAN Ingeniería del software, Mc Graw-Hill, México, D.F. Sexta edición 2006.

[RUMBAUGH, 2000] JAMES RUMBAUGH, IVAR JACOBSON, GRADY BOOCH, El Lenguaje unificado de modelado Manual de referencia, Addison Wesley de Pearson Educación, S.A., Madrid, 2000.

[DATE, 2008] C. J. DATE, Sistemas de Bases de Datos, Addison Wesley Iberoamericana S. A., México, D. F. Quinta edición 2008.

[ULLMAN, 2010] LARRY ULLMAN, Guía de aprendizaje PHP, Pearson Educación, S.A., Madrid, 2001.

[GUÍA DEL PRIMER AÑO, 2016] GUÍA DEL PRIMER AÑO, Academia de Networking de Cisco Systems, Pearson Educación, S.A., Madrid, Segunda edición 2016.

7 ACRONIMOS

API Interfaz de Programación de aplicaciones

CDE Comunicación Docente Estudiante

CGI *Common Gateway Interface*

GUI Interfaz Gráfica,

HTML Lenguaje de Marcado de Hipertexto

ICMP Protocolo de mensajes de control en Internet

IIS *Internet Information Services*

JSP .J*ava Server Pages*

MIB Base de Información de Administración

OMG *Object Management Group*

PHP Procesador de Hipertexto

RDBMS Sistemas de Gestión de Base de Datos Relacional

RUP Proceso Unificado de Desarrollo de *Rational*

SGR Sistema de Gestión de Red

SQL Lenguaje de Consulta Estructurada

SNMP Protocolo Simple de Administración de Redes

TCP Protocolo para el Control de Transmisión

UP Proceso Unificado

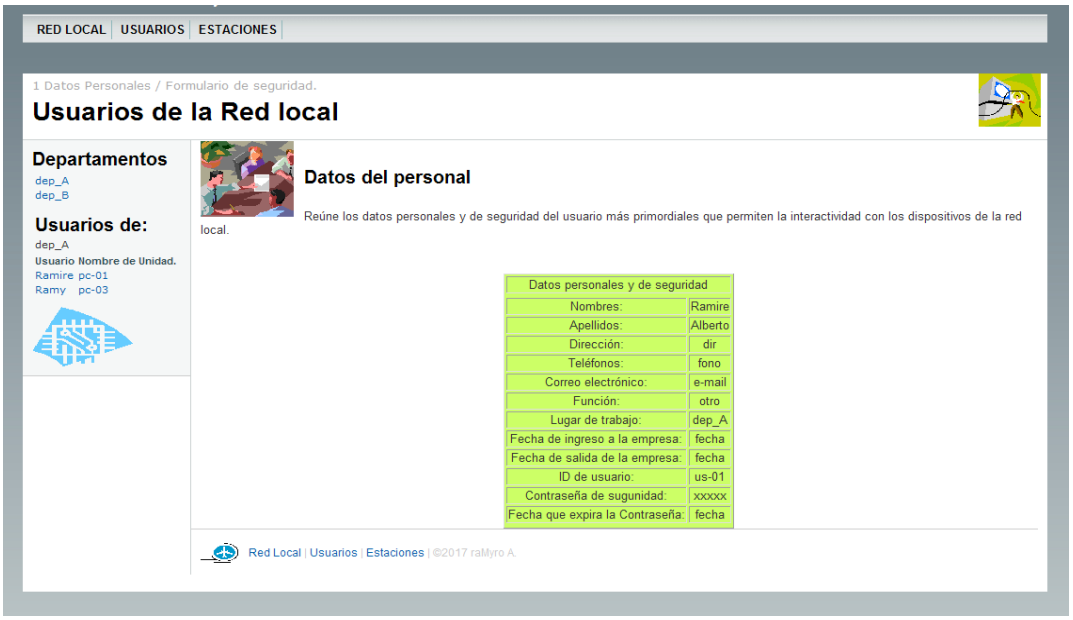
ANEXOS

8 ANEXOS

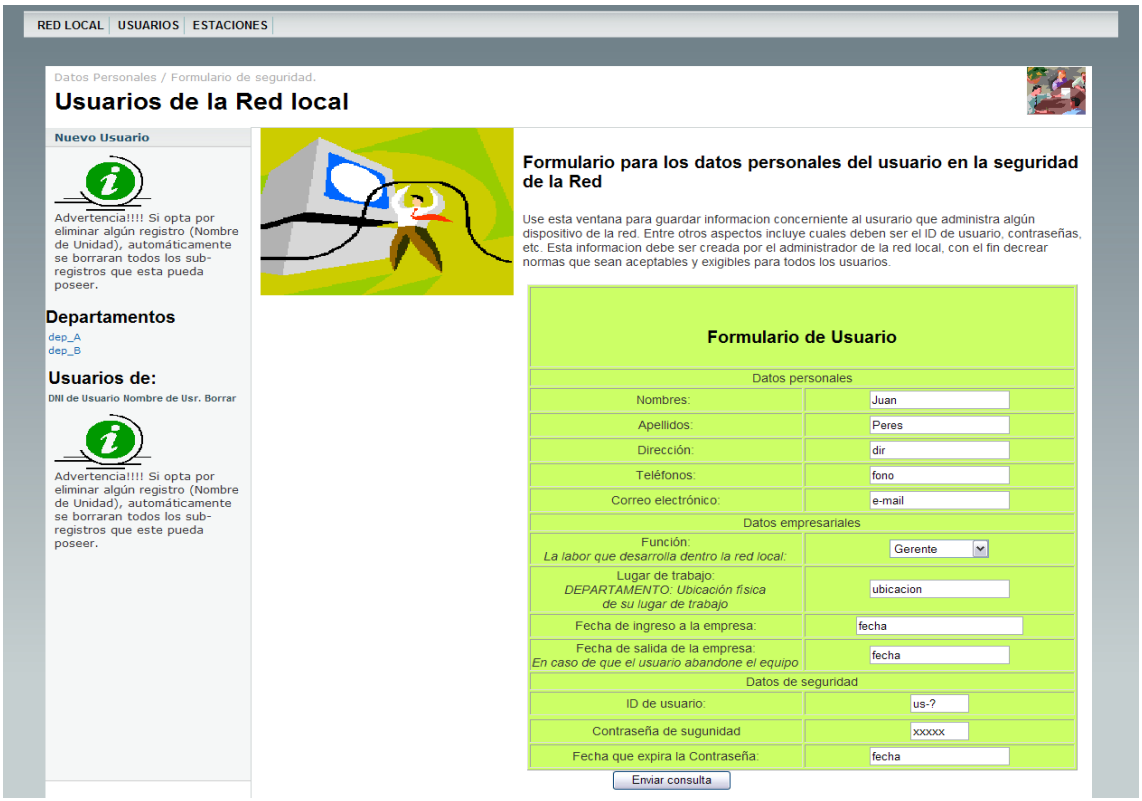
8.1 Pantalla de la ventana “Inicio de sesión”

8.2 Pantalla de la ventana “Red Local”

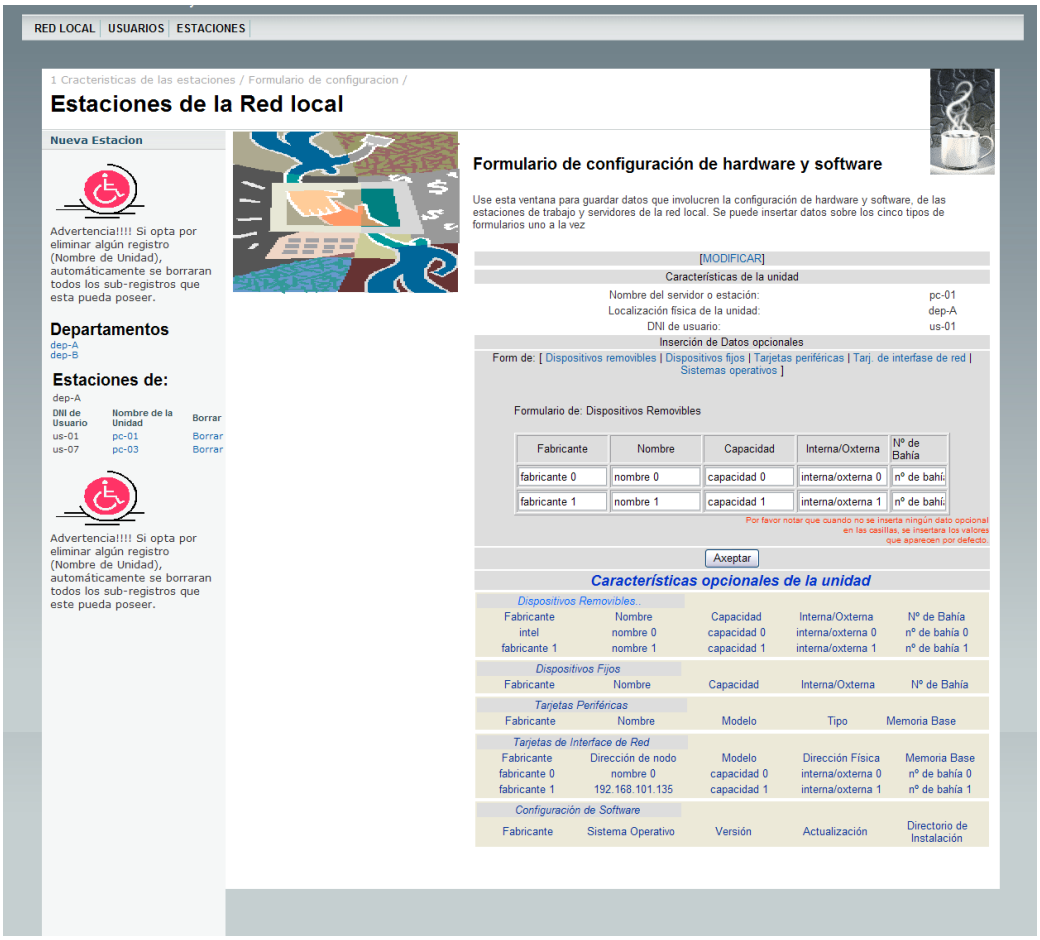
8.3 Pantalla de la ventana “Usuarios Datos personales”



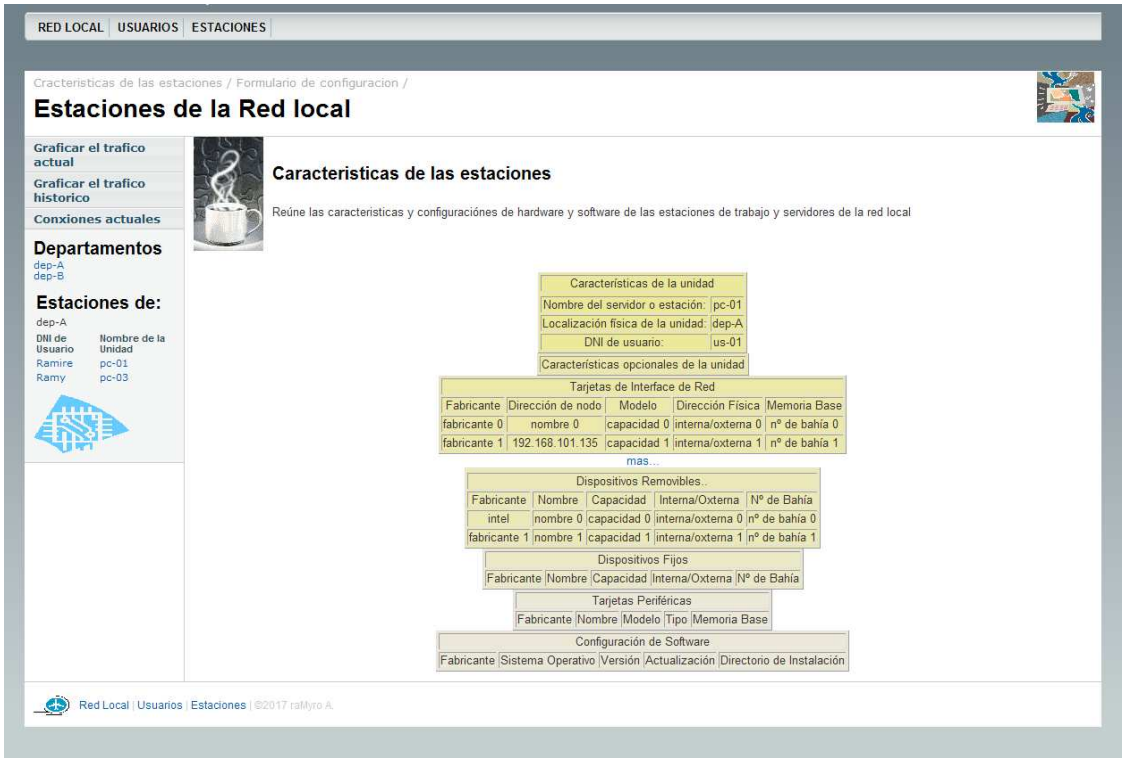
8.4 Pantalla de la ventana Usuarios “Formulario de seguridad” “Nuevo Usuario”



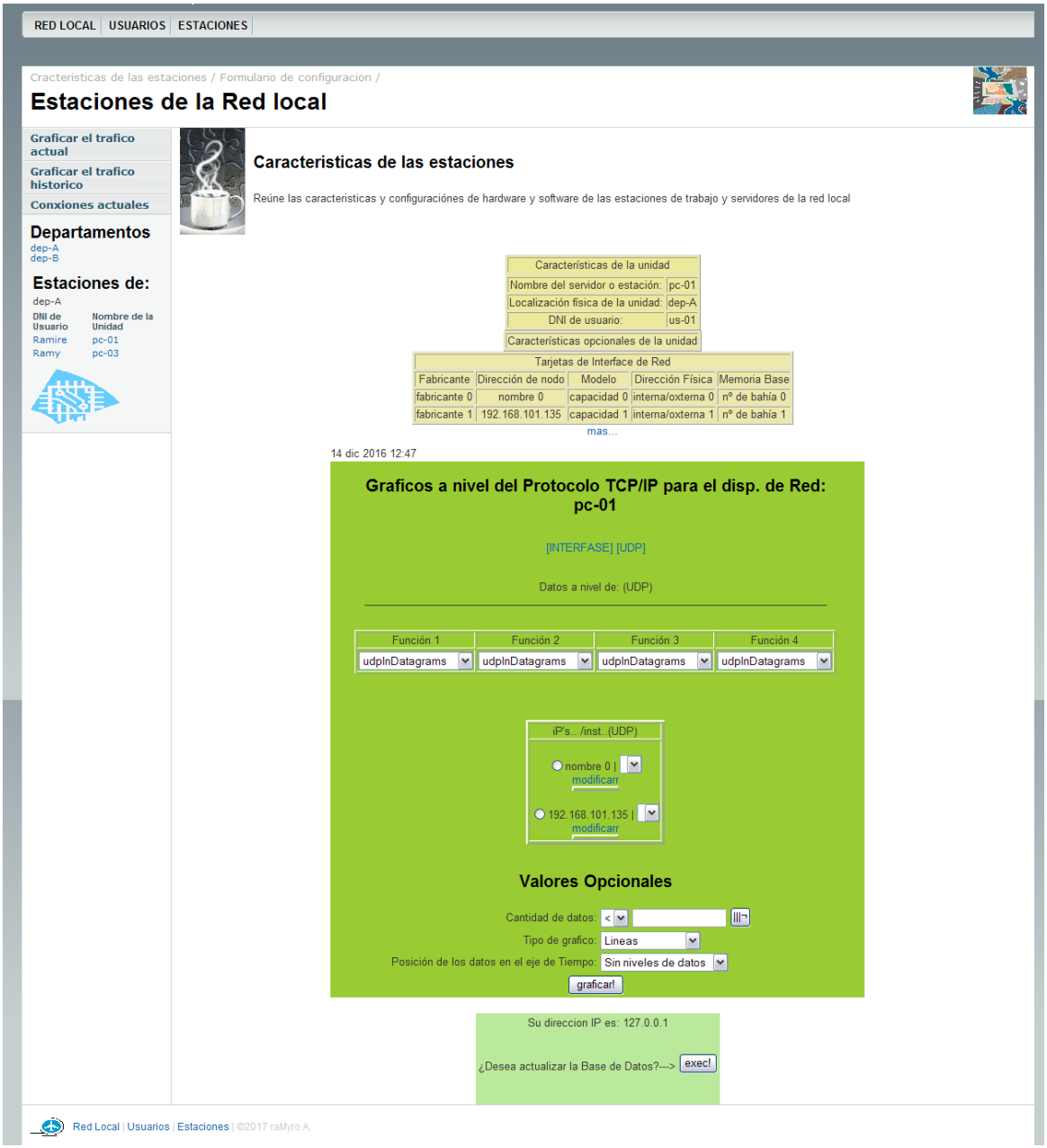
8.5 Pantalla de la ventana Estaciones “Formulario de seguridad” “Nueva estación”, “Modificar” y “Borrar”



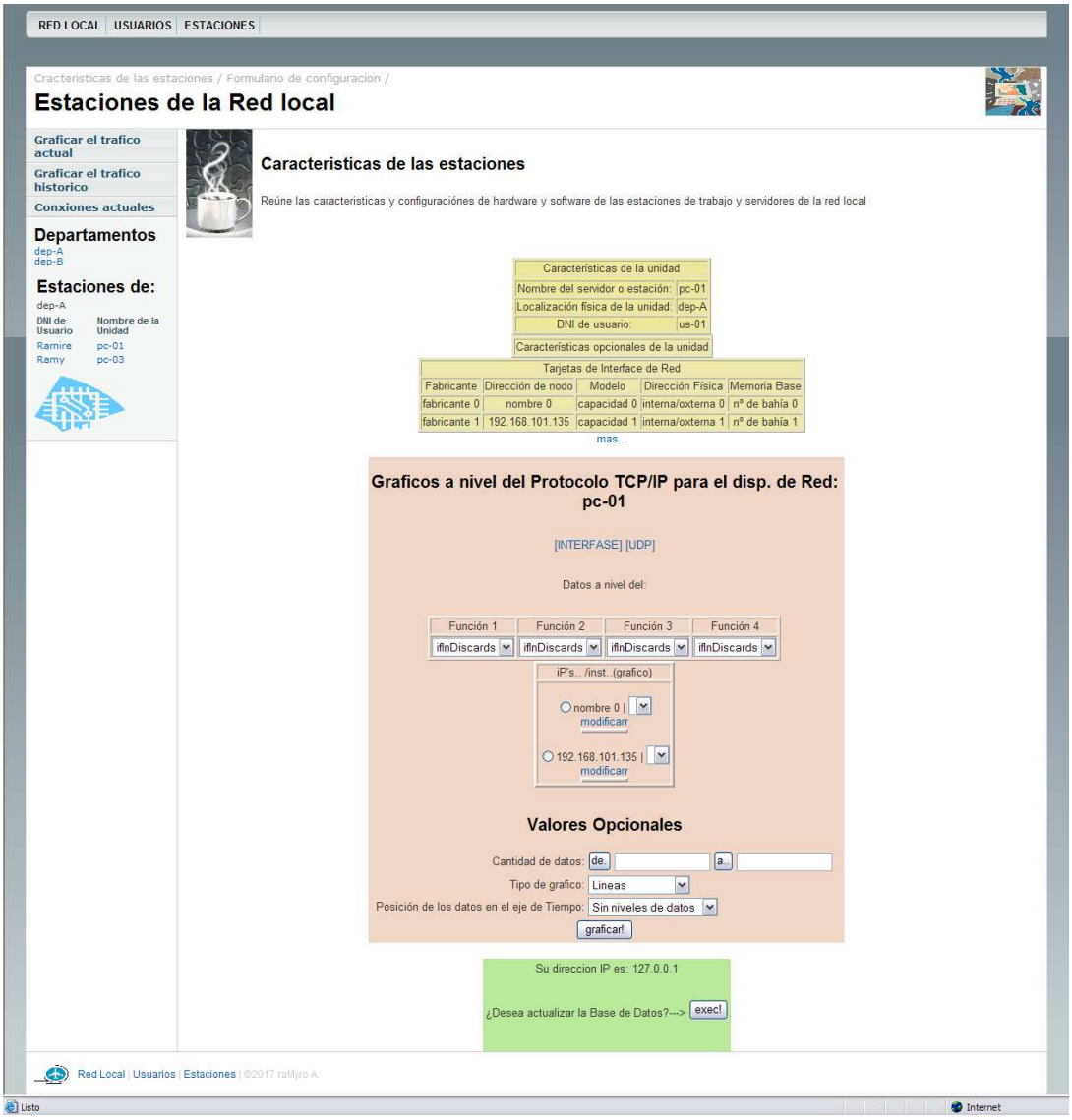
8.6 Pantalla de la ventana Estaciones “Características de las estaciones”



8.7 Pantalla de la ventana “Graficar el Tráfico actual”

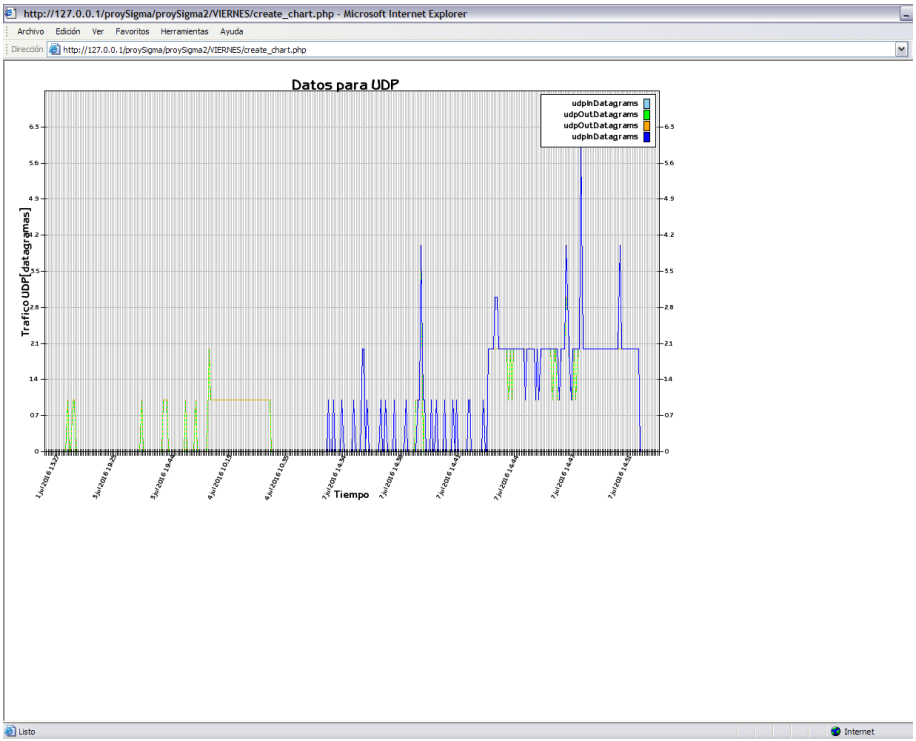


8.8 Pantalla de la ventana “Graficar el Tráfico histórico”



8.9 Pantalla de la ventana “Evaluar las conexiones actuales Ping y *SnmpWallk*”

8.10 Pantalla de la ventana “Grafico del tráfico de algunas funciones”



9 CODIGO FUENTE

Lenguaje de código en SQL

Nombre: Procedimiento almacenado OBTENER PROMEDIO
Nº de Pág.: 1

```
CREATE PROCEDURE obtenerPromedio AS

Declare
@tabla VARCHAR(80),
@Char_tabla varchar(800),
@NombreMib varchar(80),
@IdMib integer,
@IdMib8 varchar(80),
@identificador1 varchar (800),
@identificador2 varchar (1),

@CONTAR INTEGER,
@contador integer,
@char_NombreMib varchar(15),--dos char´s d NombreMib

@Char_NomMib varchar(800),@Char_NomMibq varchar(800),
@Char_NomMibIF varchar(800),@Char_NomMibIC varchar(800),@Char_NomMibIP varchar(800),
@Char_NomMibTC varchar(800),@Char_NomMibUD varchar(800),@Char_NomMibSN varchar(800),
@Char_NomMibOTRO varchar(800),

@longitudIF int,@longitudIC int,@longitudIP int,@longitudTC int,
@longitudUD int,@longitudSN int,@longitudOTRO int

set @tabla =''
set @Char_tabla=''
set @NombreMib=''
set @IdMib=''
set @Char_NomMib=''
set @Char_NomMibIF=''
set @Char_NomMibIC=''
set @Char_NomMibIP=''
set @Char_NomMibTC=''
set @Char_NomMibUD=''
set @Char_NomMibSN=''
set @Char_NomMibOTRO=''
```

```
set @longitudIF=0
set @longitudIC=0
set @longitudIP=0
set @longitudTC=0
set @longitudUD=0
set @longitudSN=0
set @longitudOTRO=0

exec CREAR_tablaYcolumna
--exec COPIAR_SnmpCollectData

--//////////////////////////////////////////(-a0-)//////////////////////////////////////////
/*                    Selecciona y guarda las cadenas unicas para cada tabla ? que involucra:
Char_NomMib?--->todas las funciones(Nombre de MIB) que forman parte del grupo de la tabla ?,
Char_tabla----->Determina la cadena de todos los nombres de tabla a ser ordenadas en -b0-
*/
Declare Obtener_NombreMib cursor for select IdMib,NombreMib from MibLevantado order by IdMib
Open Obtener_NombreMib
        fetch NEXT FROM Obtener_NombreMib into @IdMib,@NombreMib
        while @@fetch_status=0 begin
                set @IdMib8=@IdMib
                set @identificador1=''
                set @identificador2=''
                set @char_NombreMib=''
                set @contador=1

                while @contador!=3 begin
                        set @identificador1=left(@NombreMib,@contador)
                        set @identificador2=right(@identificador1,1)
                        set @char_NombreMib=@char_NombreMib+@identificador2
                        set @contador=@contador+1
                end
```

```
if (@char_NombreMib='if')begin   if(@Char_NomMibIF='')set @Char_tabla=@Char_tabla+'INTERFASE'+'.'
                                 set @Char_NomMibIF=@Char_NomMibIF+@NombreMib+','
end
else begin
        if (@char_NombreMib='ic')begin  if(@Char_NomMibIC='')set @Char_tabla=@Char_tabla+'ICMP'+'.'
                                        set @Char_NomMibIC=@Char_NomMibIC+@NombreMib+','
        end
        else begin
                if (@char_NombreMib='ip')begin  if(@Char_NomMibIP='')set @Char_tabla=@Char_tabla+'IP'+'.'
                                                set @Char_NomMibIP=@Char_NomMibIP+@NombreMib+','
                end
                else begin
                        if (@char_NombreMib='tc')begin   if(@Char_NomMibTC='')set @Char_tabla=@Char_tabla+'TCP'+'.'
                                                         set @Char_NomMibTC=@Char_NomMibTC+@NombreMib+','
                        end
                        else begin

                                if (@char_NombreMib='ud')begin  if(@Char_NomMibUD='')set @Char_tabla = @Char_tabla + 'UDP' + '.'
                                                                set @Char_NomMibUD=@Char_NomMibUD+@NombreMib+','
                                end
                                else begin
                                        if (@char_NombreMib='sn')begin if(@Char_NomMibSN='')set @Char_tabla = @Char_tabla + 'SNMP' + '.'
                                                                       set @Char_NomMibSN = @Char_NomMibSN + @NombreMib + ','
                                        end
                                        else begin if(@Char_NomMibOTRO='')set @Char_tabla = @Char_tabla + 'OTRASexpresiones' + '.'
                                                set @Char_NomMibOTRO=@Char_NomMibOTRO+@NombreMib+','
                                        end
                                end
                        end
                end
        end
fetch NEXT FROM Obtener_NombreMib into @IdMib,@NombreMib
end
```

```
set @longitudIF=abs(len(@Char_NomMibIF)-1)
        set @longitudIC=abs(len(@Char_NomMibIC)-1)
        set @longitudIP=abs(len(@Char_NomMibIP)-1)
        set @longitudTC=abs(len(@Char_NomMibTC)-1)
        set @longitudUD=abs(len(@Char_NomMibUD)-1)
        set @longitudSN=abs(len(@Char_NomMibSN)-1)
        set @longitudOTRO=abs(len(@Char_NomMibOTRO)-1)
        set @Char_NomMibIF=left(@Char_NomMibIF,@longitudIF)
        set @Char_NomMibIC=left(@Char_NomMibIC,@longitudIC)
        set @Char_NomMibIP=left(@Char_NomMibIP,@longitudIP)
        set @Char_NomMibTC=left(@Char_NomMibTC,@longitudTC)
        set @Char_NomMibUD=left(@Char_NomMibUD,@longitudUD)
        set @Char_NomMibSN=left(@Char_NomMibSN,@longitudSN)
        set @Char_NomMibOTRO=left(@Char_NomMibOTRO,@longitudOTRO)

Close Obtener_NombreMib
Deallocate Obtener_NombreMib
--//////////////////////////////////////////(-aF-)//////////////////////////////////////

--//////////////////////////////////////(-b0-)/////////////////////////////////////////
/*              MIENTRAS EXISTA AL MENOS UNA TABLA "promTABLA" DISPONIBLE,ESTE BLOQUE
                ACTUALIZARA ESTAS TABLAS, DESPUES DE EVALUAR
                CADA UNA DE LAS TAREAS(Tarea A,Tarea B,...)

                DETERMINA Y ELIGE:
Tabla           EL TIPO DE TABLA A EVALUARCE (interfase,icmp,ip,tcp,udp,snmp,otros) PARA
Char_NomMib     CADA UNA DE LAS CADENAS ESPECIFICAS A SU RESPECTIVA TABLA ?
*/
```

```
set @NombreMib=''
set @identificador1=''
set @identificador2=''
set @CONTAR=1

while (@identificador1!=@Char_tabla) BEGIN
        set @identificador1=left(@Char_tabla,@CONTAR)
        set @identificador2=right(@identificador1,1)

        IF (@identificador2!='.')set @tabla=@tabla+@identificador2
  /*B*/ELSE BEGIN --SELECT @TABLA
                if(@tabla='INTERFASE')begin     set @Char_NomMib=@Char_NomMibIF

                                        end
                else begin if(@tabla='ICMP')begin set @Char_NomMib=@Char_NomMibIC

                                                end
                        else begin if(@tabla='IP')begin    set @Char_NomMib=@Char_NomMibIP

                                                        end
                                else begin if(@tabla='TCP')begin        set @Char_NomMib=@Char_NomMibTC

                                                                end
                                        else begin if(@tabla='UDP')begin set @Char_NomMib=@Char_NomMibUD

                                                                        end
                                                else begin if(@tabla='SNMP')begin       set @Char_NomMib=@Char_NomMibSN

                                                                                end
                                                        else begin     set @Char_NomMib=@Char_NomMibOTRO

                                                                        end
                                                end
                                        end
                                end
```

```
            end
end
SET @Char_NomMibq=QUOTENAME(@Char_NomMib,'')

EXEC('
DECLARE
@tiempo_salida DATETIME,
@tiempoPromTabla DATETIME,
@tiempoMemoria DATETIME,
@primerTiempo DATETIME,
@segundoTiempo DATETIME

set @tiempo_salida =0
set @tiempoPromTabla =0
set @tiempoMemoria =0
set @primerTiempo=0
set @segundoTiempo=0

--//////////////////////////////////////(-b10-)////////////////////////////////
/*                  DETERMINA EL PERIODO DE TIEMPO(Horas) PARA LOS DATOS A SER PROCESADOS DESDE EL ULTIMO DATO QUE SE
                    ALMACENO EN LAS TABLAS "promTABLA" HASTA EL ULTIMO DATO DE UN PERIODO DE HORA QUE SE    REGISTRO
                    EN LAS TABLAS "TABLA"
tiempoActual
primerTiempo
segundoTiempo
*/
SET @tiempoMemoria=(select memo_'+@tabla+' from [MEMORIA]where id_memoria=0)

declare scrol_cursor SCROLL CURSOR FOR SELECT tiempo_salida from prom'+@tabla+'
order by tiempo_salida asc
open scrol_cursor
          FETCH last from scrol_cursor into  @tiempo_salida
          SET @tiempoPromTabla=@tiempo_salida
close scrol_cursor
deallocate scrol_cursor
```

```
set @primerTiempo=@tiempoPromTabla
set @primerTiempo=DATEadd(minute,((-1)*(DATEPART(minute,@primerTiempo))),@primerTiempo)
set @segundoTiempo=DATEADD(hour, 1, @tiempoPromTabla)
set @segundoTiempo=DATEadd(minute,((-1)*(DATEPART(minute,@segundoTiempo))),@segundoTiempo)

--SELECT DATEDIFF ( hour , @tiempoPromTabla, @tiempoMemoria ) as diferencia
--SELECT @tiempoPromTabla,@tiempoMemoria
--//////////////////////////////////////(-b1F-)////////////////////////////////

--//////////////////////////////////////(-b20-)////////////////////////////////
/*      REALIZA TODAS LAS TAREAS POR VENIR DENTRO DEL PERIODO DETERMINADO EN (-b1-)
        Y CORRESPONDIENTES A LA TABLA ACTUAL "TABLA". ALMENOS ESTE DEBE CONTENER
        UN PERIODO DE UNA HORA
        PROCESA TODAS LAS FUNCIONES(Char_NomMib) CORRESPONDIENTES
        A LA TABLA ACTUAL "TABLA" PARA OBTENER SU PROMEDIO EN EL PERIODO
        DETERMINADO EN (-b1-). ALMENOS ESTE DEBE CONTENER UN PERIODO
        DE UNA HORA
*/
WHILE (DATEDIFF ( hour , @tiempoPromTabla, @tiempoMemoria )>=2) BEGIN

        --SELECT @tiempoPromTabla,@tiempoMemoria,@primerTiempo,@segundoTiempo

        DECLARE

        @cantidad as  integer,
        @Char_NomMibq as varchar (800),
        @memoDireccionIP varchar(15),
        @memoTiempoInicio datetime,
        @memoInstancia integer,

        @longitudIF varchar(80)--??????????????????????????

        set @Char_NomMibq='+@Char_NomMibq+'
        set @Char_NomMibq=@Char_NomMibq+","
```

```
--//////////////////////////////////////((-b30-))//////////////////////////////////
/**/ /*    CREA UNA TABLA TEMPORAL("tempoTABLA")PARA LOS DATOS DE LA                  /**/
/**/       TABLA ACTUAL e.d."TABLA" EN DONDE SE DETERMINARA Y ALMACENARA              /**/
/**/       TODOS LOS NODOS E INSTANCIAS A SER PROCESADOS DENTRO DEL                   /**/

/**/       PERIODO OBTENIDO EN ((-b1-)).                                              /**/
/**/ */                                                                                    /**/
/**/ CREATE TABLE #tempoTABLA (      id_TABLA int IDENTITY (0, 1) NOT NULL ,        /**/
/**/                                memoDireccionIp varchar (15) NULL,                 /**/
/**/                                memoInstancia  integer,                            /**/
/**/                                memoTiempoInicio datetime,                         /**/
/**/                                memoTiempoFinal datetime,                                 /**/
/**/                                memoAux VARCHAR (800) NULL                                /**/
/**/                                )                                                         /**/
/**/ INSERT INTO #tempoTABLA (memoAux)  VALUES ("")                                    /**/
/**/                                                                                      /**/
/**/ INSERT #tempoTABLA (memoDireccionIP,memoInstancia)                                /**/
/**/        SELECT DISTINCT DireccionIP,Instancia from '+@tabla+'                      /**/
/**/        where tiempo_salida > @primerTiempo and tiempo_salida < @segundoTiempo      /**/
/**/                                                                                          /**/
/**/        --select * from #tempoTABLA                                                /**/
--//////////////////////////////////////((-b3F-))//////////////////////////////////

--//////////////////////////////////////((-b40-))////////////////////////////////////////////////////////////////////////////////////////////
/*          APUNTA A UN NODO ESPECIFICO Y UNA INSTANCIA ESPECIFICA DE LA TABLA
            "tempoTABLA".
*/


DECLARE punteroTempoTABLA cursor for select memoDireccionIP,memoInstancia
      from #tempoTABLA WHERE id_TABLA > 0 ORDER BY memoTiempoInicio ASC
                  open punteroTempoTABLA
                        FETCH NEXT FROM punteroTempoTABLA into @memoDireccionIP,@memoInstancia
```

```
if(@memoDireccionIP is NULL)   BEGIN
        set @longitudIF=0
END
else       BEGIN

        while @@fetch_status=0  BeGIN

                DECLARE
                        @NombreMib varchar(80),
                        @identificador3 varchar (800),
                        @identificador4 varchar (1),
                        @contador integer,
                        @contadorMib2do integer
                set @NombreMib=""
                set @identificador3=""
                set @identificador4=""
                set @contador=1
                set @contadorMib2do=0

                while (@identificador3!=@Char_NomMibq) BEGIn
                        set @identificador3=left(@Char_NomMibq,@contador)
                        set @identificador4=right(@identificador3,1)
                        if (@identificador4!=",")set @NombreMib = @NombreMib + @identificador4
                        else begiN
                                set @contadorMib2do=@contadorMib2do+1
                                --select @contadorMib2do as contadorMib2do, @NombreMib, @memoDireccionIP,
                                @memoInstancia
                                /*select @NombreMib as NombreMibNombre,@NombreMib as
                                MibNombreMibNombreMib,
                                @NombreMib as MibNombreMibNombreMibNombreMib*/
                                --////////////////////////////////////((-b50-))////////////////////////////////////

                                EVALUA EL PROMEDIO TOTAL DE UNA DTERMINADA FUNCION MIB
```

Lenguaje de código en SQL

Nombre: Procedimiento almacenado OBTENER PROMEDIO
Nº de Pág.:10

```
EXEC(" SE OBTIENE EL PROMEDIO(AVG)PARA UNA DETERMINADA
DIRECCION IP(@memoDireccionIP)
Y UNA DETERMINADA INSTANCIA(@memoInstancia) CORRESPONDIENTE
A UNA DETERMINADA FUNCION MIB(@NombreMib) EN UN INTERVALO
        DE TIEMPO(@primerTiempo y @segundoTiempo)

        DECLARE
                @PROMEDIO as  integer,
                @promedio8 as varchar(800),
                @promedio88 as varchar(800)


        set @PROMEDIO=0


        set @PROMEDIO=( select avg(ALL "+ @NombreMib + ") from '+@tabla+
        where
        direccionIp="""+@memoDireccionIP+""" and instancia =" +
        @memoInstancia+"
        and tiempo_salida > """+@primerTiempo+"""
        and tiempo_salida < """+@segundoTiempo+""" )

        if (@PROMEDIO is not null) begin
                set @promedio8=@promedio
                --SELECT @promedio8 as promedioCHAR8

        end
        else begin
                set @promedio8=-1
        end
        CONCATENAR LA CADENA DE PROMEDIOS COMPUESTA POR
        CADA FUNCION MIB

        SE LEE LA CADENA DE PROMEDIOS Y SE CONCATENA A ESTE EL
        NUEVO VALOR
```

```
                PROMEDIO(AVG) QUE PERTENECE A UNA NUEVA FUNCION
                MIB@NombreMib EVALUADA
                set @promedio88=( select memoAux from #tempoTABLA where
                id_TABLA=0)
                --SELECT @promedio88 as promedio88
                set @promedio8=@promedio88+@promedio8+"","

                SET @promedio8=QUOTENAME(@promedio8,"""""""")
                --select @promedio8 as promedio8TOTAL
                LA NUEVA CADENA CONCATENADA SE LA GUARDA EN UNA
                MEMORIA AUXILIAR(memoAux) PARA SER POSTERIOMENTE
                LEIDA Y AL MACENADA EN LA TABLA PRMEDIOS QUE LE
                CORRESPONDE


                EXEC(""
                        UPDATE #tempoTABLA  SET memoAux = "" + @promedio8 + ""
                WHERE id_TABLA = 0
                "")
                --select * from     #tempoTABLA where id_tabla=0
        ")

        set @NombreMib=""
    enD

    set @contador=@contador+1

ENd
Declare
@col_direccionIp VARCHAR(80),
@col_instancia VARCHAR(80),
@col_tiempoHora VARCHAR(80),
@col_valores VARCHAR(800),
@col_Mibs VARCHAR(800),
```

```
                                                @promediosRecu varchar(800)

                                                set @promediosRecu=""

                                                set @promediosRecu=( select memoAux from #tempoTABLA
                                                where id_TABLA = 0 )
                                                set @longitudIF=abs(len(@promediosRecu)-1)
                                                set @promediosRecu=left(@promediosRecu,@longitudIF)
                                                SET @col_direccionIP=QUOTENAME(@memoDireccionIP,"""")
                                                SET @col_instancia=QUOTENAME(@memoInstancia,"""")
                                                SET @col_tiempoHora = QUOTENAME (convert(varchar(80), @segundoTiempo, 113), """")
                                                --select @promediosRecu
                                                SET @col_valores=QUOTENAME(@promediosRecu,"""")
                                                set @longitudIF=abs(len(@Char_NomMibq)-1)
                                                set @col_Mibs=left(@Char_NomMibq,@longitudIF)

                                                --select @col_valores
                                                --select @col_Mibs
                                                EXEC("
                                                        INSERT prom'+@tabla+' ( direccionIP, instancia, tiempo_salida, "+@col_Mibs+" ) VALUES
("+@col_direccionIP+","+@col_instancia+","+ @col_tiempoHora+","+@promediosRecu+")
                                                ")
                                                --select @promediosRecu as PROMEDIOopaRAGUARDAR
                                                UPDATE #tempoTABLA  SET memoAux ="" WHERE id_TABLA = 0

                                                FETCH NEXT FROM punteroTempoTABLA into @memoDireccionIP, @memoInstancia--,
@memoTiempoInicio
                                                        EnD
                                                END
                                        close punteroTempoTABLA
                        deallocate punteroTempoTABLA
                        --////////////////////////////////////(-b4F)////////////////////////////////
                        set @tiempoPromTabla=DATEADD(hour, 1, @tiempoPromTabla)
                        set @primerTiempo=DATEADD(hour, 1, @primerTiempo)
```

```
                set @segundoTiempo=DATEADD(hour, 1, @segundoTiempo)

                DROP TABLE #tempoTABLA
            END
            --////////////////////////////////////(-b2F-)////////////////////////////////
            ')
            set @tabla=''
            set @Char_NomMib=''


  /*B*/EnD

        set @CONTAR=@CONTAR+1
END
--DELETE [SnmpCollectData]
--SELECT @TABLA
--select @tabla,@Char_NomMib
GO
```

Lenguaje de código en SQL

Nombre: Procedimiento almacenado CONEXIONES
Nº de Pág.: 1

```
CREATE PROCEDURE CONEXIONES AS
-------------------------------------------------
DECLARE @TABLA TABLE (  id_TABLA int IDENTITY (0, 1) NOT NULL ,
                        ipAddressORIGEN varchar (15) NULL,
                        puertoORIGEN  varchar (15) NULL,
                        ipAddressDESTINO varchar (15) NULL,
                        puertoDESTINO varchar (15) NULL
                        )

DECLARE
@ipAddressORIGEN varchar (15),
@ipAddressDESTINO varchar (15),
@puertoORIGEN varchar (15),
@puertoDESTINO varchar (15),

@INSTANCE varchar (440),-- Entrada
@Conexion varchar (440),
@identificador1 varchar (440),
@identificador2 varchar (1),
@char_recu varchar(150),
@byte_recu bigint,
@ByteContador int,
@CharContador int
-------------------------------------------------
set @ipAddressORIGEN =''
set @ipAddressDESTINO=''
set @puertoORIGEN=''
set @puertoDESTINO=''

SET @Conexion=''
SET @INSTANCE=''
set @identificador1=''
set @identificador2=''
set @char_recu=''
set @byte_recu=0
set @CharContador=0
```

```
set @ByteContador=0

BULK INSERT MIN_HACIENDA_NETVIEW.dbo.CONEXIONES_TcpUdp   FROM 'e:\contents.txt'

DECLARE CONEXION cursor for select Conexion FROM CONEXIONES_TcpUdp
OPEN CONEXION
fetch next from CONEXION INTO @Conexion
WHILE @@FETCH_STATUS=0 BEGIN
        set @ipAddressORIGEN =''
        set @ipAddressDESTINO=''
        set @puertoORIGEN=''
        set @puertoDESTINO=''

        SET @INSTANCE=''
        set @identificador1=''
        set @identificador2=''
        set @char_recu=''
        set @byte_recu=0
        set @CharContador=0
        set @ByteContador=0

        SET @INSTANCE=@Conexion
        while @identificador2!=':' begin
                set @identificador1=left(@INSTANCE,@CharContador)
                set @identificador2=right(@identificador1,1)
                if (@CharContador>=44)begin     if(@identificador2=':')goto alla
                                                set @char_recu=@char_recu+@identificador2
                                        end
                set @CharContador=@CharContador+1
        end
        alla:
        set @INSTANCE=@char_recu+'.'
        set @identificador1=''
        set @identificador2=''
        set @char_recu=''
```

```
set @CharContador=0

while @identificador1!=@INSTANCE
begin
set @identificador1=left(@INSTANCE,@CharContador)--Identifica cada caracter de
set @identificador2=right(@identificador1,1)              --izquierda a aderecha en
        if (@identificador2='.')                          --@INSTANCE
        begin
                set @byte_recu=@char_recu

                if (@ByteContador=0 /*or @ByteContador=5*/ ) begin  if (@byte_recu<128)begin goto alli
                                                                    end
                end
                IF (@ByteContador BETWEEN 0 AND 3)BEGIN
                   if(@ByteContador=3)begin set @ipAddressORIGEN=@ipAddressORIGEN+@char_recu
                   end
                   else begin set @ipAddressORIGEN=@ipAddressORIGEN+@char_recu+'.'
                   end
                END

                IF (@ByteContador=4)BEGIN SET @puertoORIGEN=@char_recu
                END
                IF (@ByteContador BETWEEN 5 AND 8)BEGIN
                   if(@ByteContador=8)begin set @ipAddressDESTINO=@ipAddressDESTINO+@char_recu
                   end
                   else begin set @ipAddressDESTINO=@ipAddressDESTINO+@char_recu+'.'
                   end
                END
                IF (@ByteContador=9)BEGIN SET @puertoDESTINO=@char_recu
                END

                set @char_recu=''
                set @ByteContador=@ByteContador+1  --Contador para cada byte(tres caracteres)
        end                                                --ó cada punto(.)de @ipAddressINVvarchar
        else
```

```
                begin
                        set @char_recu=@char_recu+@identificador2
                end
                set @CharContador=@CharContador+1       --Contador para cada caracter de
                                                                --@INSTANCE
        end

        INSERT @tabla VALUES (@ipAddressORIGEN,@puertoORIGEN,@ipAddressDESTINO,@puertoDESTINO)
        alli:
        fetch next from CONEXION INTO @Conexion
END
CLOSE CONEXION
DEALLOCATE CONEXION


select ipAddressORIGEN as IPorigen,puertoORIGEN as PtoORIGEN,ipAddressDESTINO as IPdestino,puertoDESTINO AS PtoDESTINO from @tabla
delete CONEXIONES_TcpUdp
GO
```

Lenguaje de código en SQL

Nombre: Procedimiento almacenado CREAR_tablaYcolumna
Nº de Pág.: 1

```
CREATE PROCEDURE CREAR_tablaYcolumna AS
Declare
@tabla1  varchar(80),
@mibName varchar(80),
@NombreMib varchar(80),
@mibID int,
@mibName1 varchar(80),
@NombreMib1 varchar(80),
@char_mibName1 varchar(15),
@char_NombreMib1 varchar(15),
@identificador1 varchar (15),
@identificador2 varchar (1),
@char_mibName varchar(15),--2 chars d mibName
@identificador3 varchar (15),
@identificador4 varchar (1),
@char_NombreMib varchar(15),
@contador integer,
@condicionX int,
@condicion int,
@condicion0 int
set @mibName=''
set @NombreMib=''
set @mibName1=''
set @NombreMib1=''
set @char_mibName1=''
set @char_NombreMib1=''
set @identificador1=''
set @identificador2=''
SET @tabla1=''
set @condicionX=0
Declare Obtener_mibLookup cursor for select mibID,mibName from netviewdb.dbo.mibLookup
order by mibID
Open Obtener_mibLookup
        fetch NEXT FROM Obtener_mibLookup into @mibID,@mibName
        while @@fetch_status=0  begin
                set @condicion=0
```

```
set @condicion0=0
Declare Obtener_NombreMib cursor for select NombreMib from MibLevantado
order by IdMib
Open Obtener_NombreMib
        fetch NEXT FROM Obtener_NombreMib into @NombreMib
        while @@fetch_status=0 begin
                set @condicionX=@condicionX+1
                if @mibName=@NombreMib begin
                        set @condicion0=@condicion0+1
                        break
                end
                else begin
                        set @identificador1=''
                        set @identificador2=''
                        set @char_mibName=''
                        set @identificador3=''
                        set @identificador4=''
                        set @char_NombreMib=''
                        set @contador=1
                        while @contador!=3 begin
                                set @identificador1=left(@mibName,@contador)
                                set @identificador2=right(@identificador1,1)
                                set @char_mibName=@char_mibName+@identificador2
                                set @identificador3=left(@NombreMib,@contador)
                                set @identificador4=right(@identificador3,1)
                                set @char_NombreMib=@char_NombreMib+@identificador4
                                set @contador=@contador+1
                        end
                        if(@char_mibName=@char_NombreMib)begin

                                set @mibName1=@mibName
                                set @char_mibName1=@char_mibName
                                set @char_NombreMib1=@char_NombreMib
                                set @condicion=@condicion+1
                        end
```

```
                        end
                fetch NEXT FROM Obtener_NombreMib into @NombreMib
                end
        Close Obtener_NombreMib
        Deallocate Obtener_NombreMib
        if(@condicionX=0)begin
                set @condicion0=0
                set @condicion=0
                set @char_NombreMib='zz'
                set @char_mibName=''
                set @identificador1=''
                set @identificador2=''
                set @contador=1
                while @contador!=3 begin
                        set @identificador1=left(@mibName,@contador)
                        set @identificador2=right(@identificador1,1)
                        set @char_mibName=@char_mibName+@identificador2
                        set @contador=@contador+1
                end
        end
        if(@condicion0=0)begin  if(@condicion=0)begin
                                        set @mibName1=@mibName
                                        set @char_mibName1=@char_mibName
                                        set @char_NombreMib1=@char_NombreMib
                                end
                Exec  CREAR_TABLAoCOLUMNA @mibName1,@NombreMib1,@char_mibName1,@char_NombreMib1,@tabla1 OUTPUT
                INSERT [MibLevantado] VALUES (@mibID,@mibName1,@tabla1)
        end
        fetch NEXT FROM Obtener_mibLookup into @mibID,@mibName
    end
Close Obtener_mibLookup
Deallocate Obtener_mibLookup

--select @mibName as mibName,@NombreMib as NombreMib,@char_mibName as char_mibName,@char_NombreMib as char_NombreMib
GO
```

Lenguaje de código en SQL

Nombre: Procedimiento almacenado TABLAoCOLUMNA
Nº de Pág.: 1

```
CREATE PROCEDURE CREAR_TABLAoCOLUMNA
@mibName varchar(80),
@NombreMib varchar(80),
@char_mibName varchar(2),
@char_NombreMib varchar(2),
@tabla1 varchar(80)OUTPUT
AS
Declare
@tabla varchar(80),
@ID_tabla varchar(80),
@columna varchar(80)
---------0-----------
/*set @mibName='snnooOutcastPkts'
set @NombreMib='snInUcastPkts'
set @char_mibName='sn'
set @char_NombreMib='sn'*/
---------0-----------

if (@char_mibName='if')begin      set @tabla='INTERFASE'
                                  set @columna=@mibName
end
else begin
        if (@char_mibName='ic')begin      set @tabla='ICMP'
                                          set @columna=@mibName
        end
        else begin
                if (@char_mibName='ip')  begin    set @tabla='IP'
                                          set @columna=@mibName
                end
                else begin

                        if (@char_mibName='tc')begin      set @tabla='TCP'
                                                          set @columna=@mibName
                        end
                        else begin
```

```
                        if (@char_mibName='ud')begin     set @tabla='UDP'
                        set @columna=@mibName
                        end
                        else begin
                                if (@char_mibName='sn')  begin   set @tabla='SNMP'
                                                                 set @columna=@mibName
                                end
                                else begin          set @tabla='OTRASexpresiones'
                                                    set @columna=@mibName
                                                    if exists (select * from dbo.sysobjects where id = object_id( N'[dbo].[OTRASexpresiones] ') and
OBJECTPROPERTY(id, N'IsUserTable') = 1)
                                                    set @char_mibName=@char_NombreMib

                                end
                        end
                    end
                end
            end
        end

if(@char_mibName!=@char_NombreMib)begin

        set @ID_tabla='Id_'+@tabla+''

        EXEC ('if exists (select * from dbo.sysobjects where
                id = object_id(N''[dbo].['+@tabla+']'') and
                OBJECTPROPERTY(id, N''IsUserTable'') = 1)drop table [dbo].['+@tabla+']
        CREATE TABLE [dbo].['+@tabla+'] (
                ['+@ID_tabla+'] [int] IDENTITY (0, 1) PRIMARY KEY NOT NULL ,
                [direccionIP] [varchar] (15) COLLATE Modern_Spanish_CI_AS NULL ,
                [instancia] [int] NOT NULL ,
                [tiempo_salida] [datetime] NOT NULL ,
                [tiempo_llegada] [datetime] NOT NULL ,
                ['+@columna+'] [int] NULL
```

```
		) ON [PRIMARY]'
		)
		EXEC ('ALTER TABLE [MEMORIA] ADD [memo_'+@tabla+'] [datetime] NULL')
		EXEC (
			'declare @fecha_actual datetime
			set @fecha_actual=GETDATE()
			update [MEMORIA] set [memo_'+@tabla+'] =@fecha_actual where id_MEMORIA=0'
		)
		SET @tabla1=@tabla
		EXEC ('if exists (select * from dbo.sysobjects where
			id = object_id(N''[dbo].[prom'+@tabla+']'') and
			OBJECTPROPERTY(id, N''IsUserTable'') = 1)drop table [dbo].[prom'+@tabla+']
		CREATE TABLE [dbo].[prom'+@tabla+'] (
			['+@ID_tabla+'] [int] IDENTITY (0, 1) PRIMARY KEY NOT NULL ,
			[direccionIP] [varchar] (15) COLLATE Modern_Spanish_CI_AS NULL ,
			[instancia] [int] NULL ,
			[tiempo_salida] [datetime]  NULL ,
			['+@columna+'] [int] NULL
		) ON [PRIMARY]'
		)
		EXEC (
			'declare @fecha_actual datetime
			set @fecha_actual=GETDATE()
			insert [prom'+@tabla+'] (tiempo_salida) VALUES (@fecha_actual)

		')
end
else begin
		EXEC ('ALTER TABLE ['+@tabla+'] ADD ['+@columna+'] [int] NULL')
		EXEC ('ALTER TABLE [prom'+@tabla+'] ADD ['+@columna+'] [int] NULL')
		SET @tabla1=@tabla

		--EXEC ('ALTER TABLE [MEMORIA] ADD [memo_'+@tabla+'] [int] NULL')
end
GO
```

Lenguaje de código en SQL

Nombre: Procedimiento almacenado COPIAR_SnmpCollectData
Nº de Pág.: 1

```
CREATE PROCEDURE COPIAR_SnmpCollectData
/*@p integer  OUTPUT,@u integer output*/
AS
declare
@memo_theKey as integer,
@memo_startTime as datetime,
@primer_datoKey as integer,
@primer_datoTime as datetime,
@ultimo_datoKey as integer,
@ultimo_datoTime as datetime,
@thekey as integer,
@endTime as datetime,
@startTime as datetime,/*condicional*************************************************/
@cantidad as integer
set @cantidad= 0
set @primer_datoKey=0
set @ultimo_datoKey=0
set @ultimo_datoTime=0
set @primer_datoTime=0
declare puntero_memoria cursor for select memo_theKey,memo_startTime from [MEMORIA] where id_memoria=0
open puntero_memoria
        fetch puntero_memoria into @memo_theKey,@memo_startTime
        set @primer_datoKey= @memo_theKey
        set @primer_datoTime= @memo_startTime
close puntero_memoria
deallocate puntero_memoria
/*DECLARE scrol_cursor SCROLL CURSOR FOR SELECT thekey,endTime FROM NetViewDB.dbo.Snmpcollectdata where endTime > @primer_datoTime
OPEN scrol_cursor
        FETCH ABSOLUTE 1 FROM scrol_cursor
        if(@primer_datoKey>@thekey)set @primer_datoKey=0
CLOSE scrol_cursor
DEALLOCATE scrol_cursor*/
--select @thekey,@endTime
declare nro_de_datos cursor for select thekey,startTime from NetViewDB.dbo.Snmpcollectdata
where thekey > @primer_datoKey order by thekey asc
--where startTime > @primer_datoTime order by startTime asc
```

```
open nro_de_datos
        fetch nro_de_datos into @thekey,@startTime
        while @@fetch_status=0  begin
                fetch nro_de_datos into @thekey,@startTime
                set @cantidad=@cantidad+1
        end

        set @ultimo_datoKey=@thekey
        set @ultimo_datoTime=@startTime
        if(@ultimo_datoKey<@primer_datoKey)set @primer_datoKey=0
close nro_de_datos
deallocate nro_de_datos
--select @primer_datoTime,@ultimo_datoTime
if @cantidad!=0 begin
        UPDATE [MEMORIA] set memo_theKey=@ultimo_datoKey,memo_startTime=@ultimo_datoTime
        INSERT [SnmpCollectData] SELECT * FROM netviewdb.dbo.snmpcollectdata where
        thekey between @primer_datoKey and @ultimo_datoKey order by thekey asc
end
GO
```

Lenguaje de código en SQL

Nombre: Procedimiento almacenado CONVERTIR_IP_CHAR_A_INT
Nº de Pág.: 1

```
CREATE PROCEDURE CONVERTIR_IP_CHAR_A_INT
@ipAddressINVvarchar varchar (15),
@ipAddressNVint bigint OUTPUT
 AS
--*****************************BLOQUE QUE PERMITE CONVERTIR DEL FORMATO DE ******************
--*************************LAS DIRECCIONES IP TARDICIONALES 255.255.255.255*****************
--*************************************AL FORMATO DE netviewDB ****************************
------------Variables propias de este bloque------
DECLARE                                                 --
--@ipAddressINVvarchar varchar (15),-- Entrada      --
--@ipAddressNVint bigint,              -- Salida          --
@identificador1 varchar (15),                       --
@identificador2 varchar (1),                        --
@char_recu varchar(15),                             --
@byte_recu bigint,                                  --
@contador int,                                      --
@constante1 bigint,                                 --
@constante2 bigint                                  --
-----------------------------------------------------------------
SET @ipAddressNVint=0
set @identificador1=''
set @identificador2=''
set @char_recu=''
set @byte_recu=0
set @contador=0
SET @constante1=255
SET @constante2=1
SET @ipAddressINVvarchar =@ipAddressINVvarchar+'.'
while @identificador1!=@ipAddressINVvarchar
begin
set @identificador1=left(@ipAddressINVvarchar,@contador)--Identifica cada caracter de izquierda
set @identificador2=right(@identificador1,1)                    --a aderecha en@ipAddressINVvarchar
        if (@identificador2!='.')
        begin
                set @char_recu=@char_recu+@identificador2
        end
```

```
        else
        begin
                set @byte_recu=@char_recu
                set @byte_recu=(@byte_recu&@constante1)*@constante2
                SET @ipAddressNVint=@ipAddressNVint|@byte_recu

                SET @constante2=256*@constante2
                set @char_recu=''
        end
        set @contador=@contador+1
end
--*************************************************************************************************
--*************************************************************************************************
GO
```

Lenguaje de código en SQL

Nombre: Procedimiento almacenado CONVERTIR_IP_INT_A_CHAR
Nº de Pág.: 1

```
CREATE PROCEDURE CONVERTIR_IP_INT_A_CHAR
@ipAddressNVint bigint,   --Entrada          --
@ipAddressINVvarchar varchar(15) OUTPUT --Salida        --
 AS
--*****************************BLOQUE QUE PERMITE CONVERTIR EL FORMATO DE ******************
--*****************************LAS DIRECCIONES IP ORIGINALES DE netviewDB A *****************
--*****************************EL FORMATO CLASICO 255.255.255.255*****************************
                        ---------Variables propias de este bloque---------
                        DECLARE
                        --@ipAddressNVint int,              --Entrada          --
                        --@ipAddressINVvarchar varchar(15), --Salida       --
                        @byte_recu bigint, @char_recu varchar(15) ,@contador int, @constante0 bigint, @constante1 bigint                                --
                        ---------------------------------------------------------------
                        set @ipAddressINVvarchar=''
                        SET @char_recu=''
                        SET @byte_recu=0
                        SET @contador=0
                        SET @constante0=255
                        SET @constante1=1
                        while @contador!=4                  --Permite la evaluacion para cuatro bytes(8 bits)
                        begin
                                SET @byte_recu = @ipAddressNVint&@constante0
                                set @char_recu = @byte_recu/@constante1
                                if(@contador=3) --No permite la adicion de un punto(.) al
                                        begin               --final de la direccion IP
                                        SET @ipAddressINVvarchar=@ipAddressINVvarchar+@char_recu
                                        end
                                else
                                        begin
                                    SET @ipAddressINVvarchar=@ipAddressINVvarchar+@char_recu+'.'
                                        end
                                        SET @contador=@contador+1
                                        SET @constante0=256*@constante0
                                        SET @constante1=256*@constante1
                        end
GO
```

Lenguaje de código en SQL

Nombre: Procedimiento almacenado ObtenerPuertos
Nº de Pág.: 1

```
CREATE PROCEDURE obtenerPuertos
@direccionIp varchar(15)
--,@nroPuerto varchar(15) output,@descPuerto varchar(255) output
AS
DECLARE @cmd sysname, @var sysname
SET @var = @direccionIp
--SET @var = '192.168.5.81'
SET @cmd = 'snmpwalk  ' + @var + ' .1.3.6.1.2.1.2.2.1.5 > E:\contents1.txt'
EXEC master..xp_cmdshell @cmd, NO_OUTPUT
BULK INSERT MIN_HACIENDA_NETVIEW.dbo.CONEXIONES_TcpUdp   FROM 'e:\contents1.txt'
--select * from CONEXIONES_TcpUdp
DECLARE @tablaPuertos TABLE (          id_tablaPuertos int IDENTITY (0, 1) NOT NULL ,
                                                nroPuerto integer NULL,
                                                descPuerto varchar (255) NULL
                                              )

DECLARE
@maximo integer,
@Conexion varchar (440),
@INSTANCE varchar (440),
@colNroPuerto varchar (440),
@colDescPuerto varchar (440),
@identificador1 varchar (440),
@identificador2 varchar (1),
@identificador3 varchar (440),
@identificador4 varchar (1),
@char_recuA varchar(150),
@char_recuB varchar(150),
@longChar int,
@longChar1 int,
@longChar2 int,
@CharContador int
----------------------------------------
set @maximo=0
SET @Conexion=''
SET @INSTANCE=''
set @colNroPuerto=''
```

Lenguaje de código en SQL

Nombre: Procedimiento almacenado ObtenerPuertos
Nº de Pág.: 2

```
set @colDescPuerto=''
set @identificador1=''
set @identificador2=''
set @identificador3=''
set @identificador4=''
set @char_recuA=''
set @char_recuB=''
set @longChar=0
set @longChar1=0
set @longChar2=0
set @CharContador=0

DECLARE CONEXION cursor for select Conexion FROM CONEXIONES_TcpUdp
OPEN CONEXION
fetch next from CONEXION INTO @Conexion
WHILE @@FETCH_STATUS=0 BEGIN
        set @INSTANCE=''
        set @colNroPuerto=''
        set @colDescPuerto=''
        set @identificador1=''
        set @identificador2=''
        set @identificador3=''
        set @identificador4=''
        set @char_recuA=''
        set @char_recuB=''
        set @CharContador=0
        SET @INSTANCE=@Conexion
        SET @longChar=len(@INSTANCE)
        SET @longChar1=@longChar-35
        set @colNroPuerto=RIGHT(@Conexion,@longChar1)
        SET @longChar2=@longChar-46--65
        set @colDescPuerto=RIGHT(@Conexion,@longChar2)
                while @identificador2!=':' begin
                set @identificador1=left(@colNroPuerto,@CharContador)
                set @identificador2=right(@identificador1,1)
```

```
                if @identificador2!=':'      set @char_recuA=@char_recuA+@identificador2
                set @CharContador=@CharContador+1
        end
        set @CharContador=0
        while @identificador4!='y' begin/*cero(0): indica que el puerto no esta activado*/
                set @identificador3=left(@colDescPuerto,@CharContador)
                set @identificador4=right(@identificador3,1)
                --select  @identificador4
                if @identificador4!=''      begin    if @identificador4!=0      begin    set @char_recuB=@char_recuB+@identificador4
                                                                                            set @identificador4='y'
                                                                                    end
                                                         else goto ALLI
                                                end
                set @CharContador=@CharContador+1
        end
        --select @char_recuA,@char_recuB,@colDescPuerto
        INSERT @tablaPuertos VALUES (@char_recuA,@char_recuB)
        ALLI:
        --select @char_recuA,@char_recuB,@colDescPuerto
        SET @longChar=0
        SET @longChar1=0
        SET @longChar2=0
        fetch next from CONEXION INTO @Conexion
END
CLOSE CONEXION
DEALLOCATE CONEXION
set @maximo=(select max (distinct nroPuerto) from @tablaPuertos)
delete CONEXIONES_TcpUdp
SET @cmd = 'snmpwalk  ' + @var + ' .1.3.6.1.2.1.2.2.1.2 > E:\contents1.txt'
EXEC master..xp_cmdshell @cmd, NO_OUTPUT
exec('BULK INSERT MIN_HACIENDA_NETVIEW.dbo.CONEXIONES_TcpUdp   FROM "e:\contents1.txt"
WITH   (          LASTROW  = '+@maximo+'        )

')
delete @tablaPuertos
```

Lenguaje de código en SQL

Nombre: Procedimiento almacenado ObtenerPuertos
Nº de Pág.: 4

```
DECLARE CONEXION cursor for select Conexion FROM CONEXIONES_TcpUdp
	OPEN CONEXION
	fetch next from CONEXION INTO @Conexion
	WHILE @@FETCH_STATUS=0 BEGIN
		SET @INSTANCE=''
		set @colNroPuerto=''
		set @colDescPuerto=''
		set @identificador1=''
		set @identificador2=''
		set @identificador3=''
		set @identificador4=''
		set @char_recuA=''
		set @CharContador=0
		SET @INSTANCE=@Conexion
		SET @longChar=len(@INSTANCE)
		SET @longChar1=@longChar-35
		set @colNroPuerto=RIGHT(@Conexion,@longChar1)
		SET @longChar2=@longChar-65
		set @colDescPuerto=RIGHT(@Conexion,@longChar2)
		while (@identificador2!=':' ) begin
			set @identificador1=left(@colNroPuerto,@CharContador)
			set @identificador2=right(@identificador1,1)
			if @identificador2!=':'	set @char_recuA=@char_recuA+@identificador2
			set @CharContador=@CharContador+1
		end
		INSERT @tablaPuertos VALUES (@char_recuA,@colDescPuerto)
		SET @longChar=0
		SET @longChar1=0
		SET @longChar2=0
		fetch next from CONEXION INTO @Conexion
	END
	CLOSE CONEXION
DEALLOCATE CONEXION
select   nroPuerto,descPuerto from @tablaPuertos ORDER BY nroPuerto asc
delete CONEXIONES_TcpUdp
GO
```

Lenguaje de código en SQL

Nombre: Procedimiento almacenado ObtenerPuertosPc
Nº de Pág.: 1

```
CREATE PROCEDURE obtenerPuertosPc
@direccionIp varchar(15)
--,@nroPuerto varchar(15) output,@descPuerto varchar(255) output
AS
DECLARE @cmd sysname, @var sysname
SET @var = @direccionIp
--SET @var = '127.0.0.1'
--SET @var = '192.168.5.81'
SET @cmd = 'snmpwalk  ' + @var + ' .1.3.6.1.2.1.2.2.1.2 > E:\contents1.txt'
EXEC master..xp_cmdshell @cmd, NO_OUTPUT
BULK INSERT MIN_HACIENDA_NETVIEW.dbo.CONEXIONES_TcpUdp   FROM 'e:\contents1.txt'
DECLARE @condicion integer
if (select top 1 dirIp from Puertos where dirIp=@direccionIp) is null    begin
        set @condicion=1
        --select @condicion
end
--select * from CONEXIONES_TcpUdp
DECLARE @tablaPuertos TABLE (        id_tablaPuertos int IDENTITY (0, 1) NOT NULL ,
                                     nroPuerto varchar (255) NULL,
                                     descPuerto varchar (255) NULL
                                    )
declare
@condicion2 integer,
@Conexion varchar (440),
@INSTANCE varchar (440),
@colNroPuerto varchar (440),
@colDescPuerto varchar (440),
@identificador1 varchar (440),
@identificador2 varchar (1),
@char_recu varchar(150),

@longChar int,
@longChar1 int,
@longChar2 int,
@CharContador int
------------------------------
```

Lenguaje de código en SQL

Nombre: Procedimiento almacenado ObtenerPuertosPc
Nº de Pág.: 2

```
set @condicion2=0
SET @Conexion=''
SET @INSTANCE=''
set @colNroPuerto=''
set @colDescPuerto=''
set @identificador1=''
set @identificador2=''
set @char_recu=''
set @longChar=0
set @longChar1=0
set @longChar2=0
set @CharContador=0
DECLARE CONEXION cursor for select Conexion FROM CONEXIONES_TcpUdp
OPEN CONEXION
fetch next from CONEXION INTO @Conexion
WHILE @@FETCH_STATUS=0 BEGIN
        SET @INSTANCE=''
        set @colNroPuerto=''
        set @colDescPuerto=''
        set @identificador1=''
        set @identificador2=''
        set @char_recu=''
        set @CharContador=0
        SET @INSTANCE=@Conexion

        SET @longChar=len(@INSTANCE)
        SET @longChar1=@longChar-35
        set @colNroPuerto=right(@Conexion,@longChar1)

        while @identificador2!=':' begin
                set @identificador1=left(@colNroPuerto,@CharContador)
                set @identificador2=right(@identificador1,1)
                if @identificador2!=':'      set @char_recu=@char_recu+@identificador2
                set @CharContador=@CharContador+1
        end
        SET @longChar2=@longChar-65
```

```
        SET @colDescPuerto=RIGHT(@Conexion,@longChar2)
        SET @longChar=0
        SET @longChar1=0
        SET @longChar2=0
        --select @char_recu,@colDescPuerto
        if @condicion is not null begin
                INSERT Puertos VALUES (@direccionIp,'',@char_recu,'','',@colDescPuerto)
        end
        --INSERT @tablaPuertos VALUES (@char_recu,@colDescPuerto)
        fetch next from CONEXION INTO @Conexion
END
CLOSE CONEXION
DEALLOCATE CONEXION
--select   nroPuerto,descPuerto from @tablaPuertos
select  @condicion2,idPuerto,nombreMib ,comment  from  Puertos where dirIp=@direccionIp order by idPuerto
delete CONEXIONES_TcpUdp
GO
```

Lenguaje de código en SQL

Nombre: Procedimiento almacenado ACTUALISACIONtotal
Nº de Pág.: 1

```
CREATE PROCEDURE ACTUALISACIONtotal000
@Ip varchar(15),@condicion2 integer output,@instancias varchar (800) output
AS
CREATE TABLE #tablaPuertos  ( dirIp varchar (255) NULL,
                              dirIpNeuf varchar (255) NULL,
                              idPuerto integer NULL,
                              idPuertoNeuf integer NULL,
                              tipoPuerto varchar (255) NULL,
                              mibName varchar (255) NULL,
                              comment varchar (255) NULL
                              )
/*CREATE TABLE tablaMemo  ( id_memo int IDENTITY (0, 1) NOT NULL ,
                              dirIp varchar (255) NULL,
                              descPuerto varchar (255) NULL,
                              idPuertoNeuf integer NULL,
                              idPuertoPasse integer NULL
                              )*/
DECLARE @cmd sysname, @var sysname,@direccionIp char(80),@message varchar(80)
--SET @var = 'dir /p'
SET @var =@Ip
set @direccionIp=@var
SET @cmd = 'snmpwalk  ' + @var + ' .1.3.6.1.2.1.2.2.1.2 > E:\contents1.txt'
EXEC master..xp_cmdshell @cmd, NO_OUTPUT
BULK INSERT MIN_HACIENDA_NETVIEW.dbo.CONEXIONES_TcpUdp   FROM 'e:\contents1.txt'

DECLARE @condicion integer
if (select top 1 dirIp from Puertos) is null     begin
        set @condicion=1
        --select @condicion
end
DECLARE @cont integer,@Conexion varchar (440)
set @cont=0
DECLARE CONT cursor for select Conexion FROM CONEXIONES_TcpUdp
        OPEN CONT
        fetch next from CONT INTO @Conexion
```

```
	WHILE @@FETCH_STATUS=0 BEGIN
		set @cont=@cont+1
		if(@cont>4) goto alli
		fetch next from CONT INTO @Conexion
	END
	alli:
	CLOSE CONT
DEALLOCATE CONT
if (@cont>4) BEGIN
	declaRE
	--@Conexion varchar (440),
	@INSTANCE varchar (440),
	@colNroPuerto varchar (440),
	@colDescPuerto varchar (440),
	@colTipoPuerto varchar (440),
	@colCommPuerto varchar (440),
	@identificador1 varchar (440),
	@identificador2 varchar (1),
	@char_recu varchar(150),
	@longChar int,
	@longChar1 int,
	@longChar2 int,
	@CharContador int,
	@Contador int
	-----------------------------
	SET @Conexion=''
	SET @INSTANCE=''
	set @colNroPuerto=''
	set @colDescPuerto=''
	set @colTipoPuerto=''
	set @colCommPuerto=''
	set @identificador1=''
	set @identificador2=''
	set @char_recu=''
	set @longChar=0
	set @longChar1=0
```

```
set @longChar2=0
set @CharContador=0
set @Contador=0
DECLARE CONEXION cursor for select Conexion FROM CONEXIONES_TcpUdp
OPEN CONEXION
fetch next from CONEXION INTO @Conexion
WHILE @@FETCH_STATUS=0 BEGIN
        SET @INSTANCE=''
        set @colNroPuerto=''
        set @colDescPuerto=''
        set @colCommPuerto=''
        set @colTipoPuerto=''
        set @identificador1=''
        set @identificador2=''
        set @char_recu=''
        set @CharContador=0
        SET @INSTANCE=@Conexion
        SET @longChar=len(@INSTANCE)
        SET @longChar1=@longChar-35
        set @char_recu=right(@Conexion,@longChar1)
        while @identificador2!=':' begin
                set @identificador1=left(@char_recu,@CharContador)
                set @identificador2=right(@identificador1,1)
                if @identificador2!=':'     set @colNroPuerto=@colNroPuerto+@identificador2
                set @CharContador=@CharContador+1
        end
        SET @longChar2=@longChar1-@CharContador-26
        set @char_recu=''
        set @CharContador=0
        set @identificador1=''
        set @identificador2=''
        set @char_recu=RIGHT(@Conexion,@longChar2)
        while @identificador2!=':' begin
                set @identificador1=left(@char_recu,@CharContador)
                set @identificador2=right(@identificador1,1)
                if @identificador2!=':'     set @colTipoPuerto=@colTipoPuerto+@identificador2
```

```
                        set @CharContador=@CharContador+1
                end
                set @longChar2=@longChar2-@CharContador
                set @char_recu=RIGHT(@Conexion,@longChar2)
        --select @char_recu
                set @Contador=0
                set @CharContador=1
                set @identificador1=''
                set @identificador2=''
                while @identificador1!=@char_recu begin
                        set @identificador1=left(@char_recu,@CharContador)
                        set @identificador2=right(@identificador1,1)
                        if @identificador2!=''      begin
                                set @colDescPuerto=@colDescPuerto+@identificador2
                                --select @char_recu as recu,@colDescPuerto as descrip,@CharContador as charcont,@Contador as cont
                                set @Contador=@Contador+1
                        end
                        else goto alla
                        set @CharContador=@CharContador+1
                end
        --select @char_recu as recu,@colDescPuerto as descrip,@CharContador as charcont,@Contador as cont
                alla:
                set @Contador=@longChar2-@Contador
                set @colCommPuerto=RIGHT(@Conexion,@Contador)
                SET @longChar=0
                SET @longChar1=0
                SET @longChar2=0
                if @condicion is not null begin
                        INSERT Puertos VALUES (@var, @var, @colNroPuerto, @colNroPuerto, @colTipoPuerto, @colDescPuerto, @colCommPuerto)
                end
                INSERT #tablaPuertos VALUES (@var,@var, @colNroPuerto, @colNroPuerto, @colTipoPuerto, @colDescPuerto, @colCommPuerto)
                fetch next from CONEXION INTO @Conexion
        END
        CLOSE CONEXION
        DEALLOCATE CONEXION
END
```

```
ELSE BEGIN
	--DECLARE @direccionIp varchar(15)
	--SET @direccionIp = '127.0.0.1'
	exec obtenerPuertosPc @direccionIp
END

--select * from #tablaPuertos ORDER BY idPuerto ASC
delete CONEXIONES_TcpUdp
if (select top 1 id_memo from tablaMemo where dirIp=@direccionIp) is null    begin
	INSERT tablaMemo select distinct dirIpNeuf,mibName,idPuertoNeuf,idPuertoPasse from
	#tablaPuertos cross join Puertos where #tablaPuertos.idPuerto=Puertos.idPuerto and
	#tablaPuertos.dirIp=Puertos.dirIp and idPuertoNeuf!=idPuertoPasse order by idPuertoNeuf asc
	--select @condicion
--SELECT * FROM tablaMemo
--DROP TABLE #tablaPuertos
--DROP TABLE tablaMemo
end
-------------------------------------------------------------------------------------------
declare @idPuertoPasse integer,@idPuertoPasse8 varchar(80)/*,@instancias varchar (800)*/,@longitud integer
--,@condicion2 integer ,@instancias varchar (800)
set @idPuertoPasse=0
set @idPuertoPasse8=''
set @instancias=''
set @longitud=0
set @condicion2=0
DECLARE INSTANCIAS cursor for select idPuertoPasse FROM tablaMemo where dirIp=@direccionIp
	OPEN INSTANCIAS
		fetch next from INSTANCIAS INTO @idPuertoPasse
		set @instancias=''
		set @condicion2=0
		WHILE @@FETCH_STATUS=0 BEGIN
			set @idPuertoPasse8=@idPuertoPasse
			set @instancias = @instancias + ' instancia ='+@idPuertoPasse8+' or '
			set @condicion2=@condicion2+1
			fetch next from INSTANCIAS INTO @idPuertoPasse
```

```
                END
                set @longitud=abs(len(@instancias)-3)
                set @instancias=left(@instancias,@longitud)
                --select @instancias,@longitud
                --if @condicion2 is not null begin
                        --EXEC ACTUALIZARinstancia @instancias,@direccionIp,@message OUTPUT-- sedebe anotar las
                        --variables de entrada y salida de acuedo al mismo orden en las q
                        --se resibira en el procedimiento  alamcenado
                        --select @message
                        --select @condicion2 as condicion2
                --end
        CLOSE INSTANCIAS
DEALLOCATE INSTANCIAS
if @condicion2=0 begin
        --select @condicion2
        select  @condicion2,idPuerto,/*idPuertoNeuf,tipoPuerto,*/nombreMib ,comment  from  Puertos where dirIp=@direccionIp order by idPuerto
        --select  @condicion2,/*idPuerto,*/idPuertoNeuf,/*tipoPuerto,*/mibName ,comment  from  #tablaPuertos where dirIp=@direccionIp order by idPuerto
end
else select @condicion2 , @instancias, descPuerto,idPuertoNeuf,idPuertoPasse FROM tablaMemo where dirIp=@direccionIp
DROP TABLE #tablaPuertos
--delete  tablaMemo
GO
```

Lenguaje de código en PHP-JAVA

Nombre: ventana principal RED_LOCAL
Nº de Pág.: 1

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
 "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<!-- DW6 -->
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<title>Untitled Document</title>
<link rel="stylesheet" href="proySigma2/RED_LOCAL/emx_nav_left.css" type="text/css" />
<script type="text/javascript">
<!--
var time = 3000;
var numofitems = 7;
//menu constructor
function menu(allitems,thisitem,startstate){
callname= "gl"+thisitem;
divname="subglobal"+thisitem;
this.numberofmenuitems = 7;
this.caller = document.getElementById(callname);
this.thediv = document.getElementById(divname);
this.thediv.style.visibility = startstate;
}
/menu methods
function ehandler(event,theobj){
  for (var i=1; i<= theobj.numberofmenuitems; i++){
    var shutdiv =eval( "menuitem"+i+".thediv");
    shutdiv.style.visibility="hidden";
}
   theobj.thediv.style.visibility="visible";
}
function closesubnav(event){
  if ((event.clientY <26)||(event.clientY > 85)){
    for (var i=1; i<= numofitems; i++){
      var shutdiv =eval('menuitem'+i+'.thediv');
      shutdiv.style.visibility='hidden';
    }
  }
```

```
}
// -->
</script>
</head>
<body onmousemove="closesubnav(event);">
<div id="masthead">
<h1 id="siteName">Ministerio de Hacienda</h1>

<div id="globalNav">
   <img alt="" src="SSSS/gblnav_left.gif" height="32" width="4" id="gnl" /> <img alt="" src="RED/glbnav_right.gif" height="32" width="4" id="gnr" />
<div id="globalLink">
            <a href="proySigma2/RED_LOCAL.php" id="gl3" class="glink" onmouseover="ehandler(event,menuitem3);">RED LOCAL</a>
            <a href="proySigma2/USUARIOs.php" id="gl1" class="glink" onmouseover ="ehandler(event,menuitem1);">USUARIOS</a>
            <a href="proySigma2/ESTACIONes.php" id="gl2" class="glink" onmouseover ="ehandler(event,menuitem2);" >ESTACIONES</a>
</div>
 <!--end globalLinks-->
</div>

<div id="pagecell1">
<!--pagecell1-->
<img alt="" src="proySigma2/RED/tl_curve_white.gif" height="6" width="6" id="tl" /> <img alt="" src="RED/tr_curve_white.gif"

height="6" width="6" id="tr" />
<div id="pageName">
   <h2>RED LOCAL</h2>
<img alt="small logo" src="proySigma2/RED_LOCAL/DIES.WMF" height="59" width="66"/>
</div>
<div id="col2">
<div class="feature">
<img src="proySigma2/RED_LOCAL/UNO.GIF" alt="" width="100" height="86" />
<h3>Diagrama de red </h3>
<p>El diagrama de red, le permite navegar sobre cada una de las maquinas de la red local
      y cada uno de los usuarios que la administra o la usa.
</p>
</div>
```

Lenguaje de código en PHP-JAVA

Nombre: ventana principal RED_LOCAL
Nº de Pág.: 4

```
<div class="story">
<!--<h3>RED LOCAL</h3>-->
<div id="advert">
<img src="proySigma2/RED_LOCAL/SIETE.WMF" alt="" width="107" height="66" />
</div>
</div>
<!--end col1 div -->
<div id="siteInfo">

<img src="proySigma2/RED_LOCAL/CINCO.WMF" width="44" height="22" /> <a href="proySigma2/RED_LOCAL.php">Red Local</a>
          | <a href="proySigma2/USUARIOS.php">Usuarios</a> | <a href="proySigma2/ESTACIONES.php">Estaciones</a> | &copy;2008
        raMyro A.
</div>
</div>
<!--end pagecell1-->
<br />
<script type="text/javascript">
<!--
    var menuitem1 = new menu(7,1,"hidden");
                        var menuitem2 = new menu(7,2,"hidden");
                        var menuitem3 = new menu(7,3,"hidden");
                        var menuitem4 = new menu(7,4,"hidden");
                        var menuitem5 = new menu(7,5,"hidden");
                        var menuitem6 = new menu(7,6,"hidden");
                        var menuitem7 = new menu(7,7,"hidden");
  // -->
</script>
</body>
</html
```