Instituto Politécnico Nacional

Centro de Investigación e Innovación Tecnológica

“Desarrollo de un tablero de control para análisis de tráfico en una red local.”

T E S I S

QUE PARA OBTENER EL GRADO EN
MAESTRÍA EN TECNOLOGÍA AVANZADA

PRESENTA

ING. JONATAN JUÁREZ HINOJOSA

DIRECTOR DE TESIS

DR. GUSTAVO MARTÍNEZ ROMERO

Junio de 2011

SIP-14

INSTITUTO POLITÉCNICO NACIONAL

SECRETARÍA DE INVESTIGACIÓN Y POSGRADO

ACTA DE REVISIÓN DE TESIS

En la Ciudad de México siendo las 12:00 horas del día 26 del mes de Julio del 2011 se reunieron los miembros de la Comisión Revisora de la Tesis, designada por el Colegio de Profesores de Estudios de Posgrado e Investigación de

para examinar la tesis titulada: Desarrollo de un tablero de control para análisis de tráfico en una red local

Presentada por el alumno:

Juárez	Hinojosa	Jonatan
Apellido paterno	Apellido materno	Nombre(s)

Con registro: B 0 9 1 5 8 6

aspirante de:

Máestria en tecnología Avanzada

Después de intercambiar opiniones, los miembros de la Comisión manifestaron ***APROBAR LA TESIS***, en virtud de que satisface los requisitos señalados por las disposiciones reglamentarias vigentes.

LA COMISIÓN REVISORA

Director(a) de tesis

Dr.Gustavo Martínez Romero

Dr.Fernando Martínez Piñon

Dr.Jose Alfredo Alvarez Chávez

Dr.Gilberto Lorenzo Martínez Luna

M. en C. Agustín Cruz Contreras

M. en C. Vicente Mayagoitia Barragán

PRESIDENTE DEL COLEGIO DE PROFESORES

M. en C. Vicente Mayagoitia Barragán

INSTITUTO POLITÉCNICO NACIONAL
CENTRO DE INVESTIGACIÓN E
INNOVACIÓN TECNOLÓGICA
DIRECCIÓN

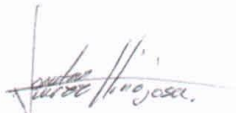
INSTITUTO POLITÉCNICO NACIONAL

SECRETARÍA DE INVESTIGACIÓN Y POSGRADO

CARTA CESIÓN DE DERECHOS

En la Ciudad de México el día 25 del mes Julio del año 2011, el (la) que suscribe Jonatan Juárez Hinojosa alumno (a) del Programa de Maestría en Tecnologia Avanzada con número de registro B091586, adscrito a Centro de Investigación e Innovación Tecnológica, manifiesta que es autor (a) intelectual del presente trabajo de Tesis bajo la dirección de Dr. Gustavo Martínez Romero y cede los derechos del trabajo intitulado Desarrollo de un tablero de control para el análisis de red local, al Instituto Politécnico Nacional para su difusión, con fines académicos y de investigación.

Los usuarios de la información no deben reproducir el contenido textual, gráficas o datos del trabajo sin el permiso expreso del autor y/o director del trabajo. Este puede ser obtenido escribiendo a la siguiente dirección cool_2004_7@hotmail.com. Si el permiso se otorga, el usuario deberá dar el agradecimiento correspondiente y citar la fuente del mismo.



Jonatan Juárez Hinojosa

Nombre y firma

Dedicatoria:

Este trabajo esta dedicado a Fernando Juárez Esparza, María Patricia Hinojosa Zúñiga, Omar Fernando Juárez Hinojosa por su apoyo incondicional y una especial dedicatoria a mi abuela Antonia Esparza Nava, trabajadora manual, que educo al hijo profesionista, trabajador del STC Metro, quién crio al hijo investigador, sin el esfuerzo de ellos no hubiera podido lograrlo.

Agradecimiento al M en C. Mauricio Olguín que gracias a su orientación y consejos se pudo concluir satisfactoriamente este trabajo de tesis.

Abstract.

The administration of a local area network (LAN) is a hard work which is hardened by the number of active devices, media, protocols, different topologies, mix of signals, and so on.

Dashboards allow to obtain information graphically about events that saturate the bandwidth of the transmission medium of a LAN. The objective of this work is the development of an application to detect network traffic, using the protocols IP, TCP, UDP and DNS implementation to detect the consumption of bandwidth of the transmission medium in order to generate alerts if the permitted limits are exceeded.

LAN traffic analysis is performed based on the information obtained by the packet sniffer. The “sniffing” method proposed is *man in the middle* (MITM), where the capture rate is the same of the transmission medium. The traffic analysis information is stored in a database, which will be displayed by the dashboard.

The dashboard architecture is composed by three main modules: 1) a data acquisition module that is responsible for collecting, analyzing and processing the packets over transmission medium, without affecting the process of sending and receiving data using a packet sniffer, 2) a database that stores information produced by the data acquisition module, and 3) a graphical interface that shows the behavior of the consumption of bandwidth, the number of packets input, output, errors, the active devices on the network. Also allows the edition of devices on the network and the setup of alarms and report generation.

Resumen.

La administración de una red de área local (LAN) es una ardua labor que se dificulta por el número de equipos activos, medios de comunicación, protocolos, diferentes topologías, mezcla de señales, etc.

Los tableros de control permiten obtener gráficamente, información de los eventos que saturan el ancho de banda del medio de transmisión de una LAN, el objetivo de este trabajo es desarrollar una aplicación para detectar el tráfico de red utilizando los protocolos IP, TCP, UDP y la aplicación DNS, para la detección del consumo del ancho de banda del medio de transmisión generando alertas si los límites permitidos son rebasados.

El análisis de tráfico de LAN, se realiza a partir de la información obtenida por el husmeador de paquetes. El husmeador propuesto es del tipo hombre en el medio (MITM), donde la velocidad de captura es la misma del medio de transmisión. La información del análisis de tráfico es almacenada en una base de datos, que será mostrada por el tablero de control.

La arquitectura del tablero de control está compuesta por tres elementos principales: 1) un módulo de adquisición de datos que se encarga de obtener, analizar y procesar los paquetes enviados en el medio de transmisión, sin afectar el proceso de envío y recepción de datos utilizando un husmeador paquetes, 2) una base de datos que almacena la información producida por el módulo de adquisición de datos, y 3) una interfaz gráfica que muestran el comportamiento del consumo de ancho de banda, la cantidad de paquetes de entrada, salida, errores, los equipos activos en la red. Permite además generar reportes, configuración parámetros de segmento, número de equipos, configurar alarmas, agregar, modificar, buscar y borrar nodos de red.

Contenido:

CAPÍTULO 1. INTRODUCCIÓN

1.1 REDES DE DATOS. 1

1.2 ADMINISTRACIÓN DE LAN. 2

1.3 TABLERO DE CONTROL..... 2

1.4 PLANTEAMIENTO DEL PROBLEMA. 3

1.5 OBJETIVO GENERAL 4

1.6 ALCANCES DEL PROYECTO..... 4

1.7 ESTRUCTURA GENERAL DEL DOCUMENTO. 6

1.8 RESUMEN. 8

CAPÍTULO 2. ANÁLISIS DE TRÁFICO

2.1 ANTECEDENTES DE LA COMUNICACIÓN ENTRE COMPUTADORAS. 9

2.2 LA ADMINISTRACIÓN DE RED. 10

2.3 HERRAMIENTAS DE ADMINISTRACIÓN DE RED. 11

2.3.1 Tablero de control. *11*

2.3.1.a Medidores de transmisión y recepción de datos. 13

2.3.1.b Detalles de interfaz de red. 13

2.3.1.c Capacidad de supervisión por día. 15

2.3.1.d Gráficas de comportamiento de red. 15

2.3.1.e Ventajas y desventajas de un tablero de control. 17

2.3.2 Husmeador de paquetes. *18*

2.3.2.a Captura por Ethernet compartido. 19

2.3.2.b Captura sobre una máquina. 20

2.3.2.c Captura usando un concentrador Ethernet (HUB Ethernet). 21

2.3.2.e Captura usando la máquina en el medio. 23

2.3.2.f Captura utilizando una llave de red (Switch + Tap). 24

2.3.2.g Captura con hombre en el medio o Intermediario (*MITM, Man-in-the-Middle*) 25

2.3.2.h Ventajas y desventajas del husmeador de paquetes de red. 26

2.3.3 Sistemas de detección de intrusiones. *27*

2.3.3.a Tipos de alarmas 28
2.3.3.b Ventajas y desventajas de los sistemas de detección de intrusiones (IDS). 30
2.4 COMPARACIÓN ENTRE LAS HERRAMIENTAS DE ADMINISTRACIÓN Y EL PRESENTE DESARROLLO. 31
2.5 RESUMEN. 35

CAPÍTULO 3. DESARROLLO DE LOS MÓDULOS DEL TABLERO DE CONTROL

3.1 ARQUITECTURA DEL GENERAL DEL SISTEMA 36
3.1.1 Análisis de tráfico. *39*
3.1.2 Base de datos *41*
3.1.3 Interfaz gráfica *42*
3.2 DISEÑO DEL SISTEMA 46
3.2.1 Análisis de tráfico *46*
3.2.1.a El husmeador de paquetes. 46
3.2.1.a.i Captura de paquetes 50
3.2.1.a.ii Análisis de paquetes. 52
3.2.1.a.ii.1 Análisis de encabezado de IP 52
3.2.1.a.ii.2 Análisis de encabezado de TCP 59
3.2.1.a.ii.3 Análisis de encabezado de UDP. 65
3.2.1.a.ii.4 Análisis de encabezado de DNS 70
3.2.1.b Detección de paquetes de entrada / salida. 75
3.2.1.c Comprobación de conexión (PING). 79
3.2.1.d Detección de interfaces de red. 83
3.2.2 Diseño de la bases de datos *87*
3.2.2.a Diseño lógico 87
3.2.2.a.i Diagramas entidad relación. 87
3.2.2.a.i.1 Consultas del algebra relacional para la base de datos del tablero de control 92
3.2.2.a.i.2 Consultas SQL de la base de datos del tablero de control 94
3.2.2.b Diseño físico 96
3.2.3 Interfaz gráfica *105*
3.2.3.a Controles de análisis de red. 105

3.2.3.a.i Husmeador de paquetes. 105
3.2.3.a.ii Contador de paquetes..... 107
3.2.3.a.iii Estadísticas. 107
3.2.3.a.iv Comprobación de conexión (PING). 108
3.2.3.a.v Interfaces de red. 109
3.2.3.b Medidores. 110
3.2.3.b.i Medidor del consumo de ancho de banda. 110
3.2.3.a.i.1 Alarmas. 114
3.2.3.a.i.2 Velocidad de transmisión 115
3.2.3.a.ii Medidor de equipos activos en la red. 120
3.2.3.c Controles de usuario. 121
3.2.3.c.i Barra de menú. 121
3.2.3.c.ii Botón de inicio / alto. 122
3.2.3.c.iii Seleccionar interfaz. 123
3.2.3.c.iv Editor de nodo de red. 123
3.2.3.c.iv.1 Botones de edición de nodo. 124
3.2.3.c.iv.2 Información y edición de nodo. 124
3.2.3.c.iv.3 Filtro. 126
3.3 RESUMEN. 127

CAPÍTULO 4. PRUEBAS Y RESULTADOS

4.1 PRUEBAS DEL MÓDULO ADQUISICIÓN DE DATOS DE RED. 128
4.1.1 Prueba del módulo husmeador de paquetes (Sniffer). 128
4.1.2 Prueba del módulo estadísticas. 134
4.1.3 Prueba del módulo comprobación de conexión de equipo (PING). 137
4.1.4 Prueba del módulo adaptadores de red. 142
4.2 PRUEBAS DE LA BASE DE DATOS. 146
4.2.1 Pruebas de desempeño en la carga de datos. 146
4.3.2Pruebas de desempeño en consultas de datos. 157
4.3 PRUEBAS DEL TABLERO DE CONTROL. 164
4.3.1 Prueba de la Barra de menú. 164
4.3.2 Prueba de la pestaña monitoreo. 167
4.3.3 Pruebas a la pestaña reporte. 170
4.3.4 Pruebas a la pestaña administración. 174

CAPÍTULO 5. CONCLUCIONES Y TRABAJO FUTURO

5.1 CONCLUSIONES. 187
5.2 TRABAJO A FUTURO. 188

BIBLIOGRAFÍA. 190

GLOSARIO. 193

APÉNDICE "A". 194
EL MODELO OSI. 194
APÉNDICE "B". 200
ADAPTADOR DE RED. 200
APÉNDICE "C". 204
BASES DE DATOS RELACIONALES. 204
APÉNDICE "D" 209
CÓDIGO HUSMEADOR DE PAQUETES O SNIFFER 209
APÉNDICE "E" 228
CÓDIGO DETECCIÓN DE PAQUETES DE ENTRADA / SALIDA. .. 228
APÉNDICE "F" 230
CÓDIGO DE COMPROBACIÓN DE CONEXIÓN (PING) 230
APÉNDICE "G" 232
CÓDIGO DE ADAPTADORES DE RED. 232
APÉNDICE "H" 234
CÓDIGO GENERA DATOS. 234
APÉNDICE I 235
MANUAL DE USUARIO. 235
APÉNDICE "J". 248
ESQUEMAS DE LA FAMILIA DE DIRECCIONES. 248

Índice de imágenes:

Pág

Capítulo 2. Análisis de tráfico.

Figura 2.1 - Tablero de control OpManager…… 12
Figura 2.2 - Graficas de transmisión y recepción…… 13
Figura 2.3 - Detalles de interfaz…… 14
Figura 2.4 - Graficas de viabilidad o cobertura por día…… 15
Figura 2.5 - Tráfico del día…… 16
Figura 2.6 - Errores descartados por día…… 16
Figura 2.7 - Total de bytes transferidos por día…… 16
Figura 2.8 - Paquetes transferidos por segundo en el día…….. 17
Figura 2.9 - Información de una captura típica de un sniffer…… 18
Figura 2.10 – Captura de paquetes de datos en una red con un concentrador… 19
Figura 2.11 – Paquetes de datos no capturados por la intervención del switch 20
Figura 2.12 – Captura solo en el equipo de interés…… 21
Figura 2.13 – Conexión del HUB entre el switch y la computadora…… 22
Figura 2.14 – Dispositivo de captura conectado a un puerto del switch…… 23
Figura 2.15 – Captura maquina en el medio…….. 24
Figura 2.16 – Captura utilizando una llave de red……. 25
Figura 2.17 – Conexión para realizar captura en hombre en el medio……. 26
Figura 2.18 – Puntos de instalación de IDS…….. 29

Capítulo 3. Desarrollo de los módulos del tablero de control

Figura 3.1 – Arquitectura del tablero de control…… 38
Figura 3.2 – Arquitectura de la captura de tráfico de red…….. 39
Figura 3.3 - Niveles de abstracción dentro de la base de datos……. 41
Figura 3.4 – Arquitectura de la interfaz grafica del tablero de control……. 44
Figura 3.5 – Diagrama EPC del olfateador de paquetes…… 48
Figura 3.6 – Paquete TCP encapsulado en una trama de datos de IP…… 52
Figura 3.7 – Formato de encabezado de IP…… 52
Figura 3.8 - Trama de 8 bits de servicio ..…….. 53
Figura 3.9 – Encabezado de los 3 bits de bandera…… 54
Figura 3.10 – Diagrama EPC función IPHeader…… 56
Figura 3.11 – Diagrama EPC análisis de encabezado de IP…….. 57
Figura 3.12 – Formato de encabezado TCP…….. 59
Figura 3.13 – Pseudo encabezado de TCP…… 60
Figura 3.14 – Diagrama EPC función TCPHeader…… 62
Figura 3.15 – Diagrama EPC de análisis de encabezado de TCP…… 63
Figura 3.16 – Formato de encabezado UDP……. 65
Figura 3.17 – Pseudo encabezado de UDP…… 65
Figura 3.18 – Diagrama EPC de la función UDPHeader……. 67
Figura 3.19 – Diagrama EPC del análisis de encabezado de UDP…… 68
Figura 3.20 – Diagrama EPC de la función DNSHeader……. 72

Figura 3.21 – Diagrama EPC del análisis de encabezado de DNS……... 73
Figura 3.22 – Diagrama EPC de la recepción de paquetes de entrada/salida..… 77
Figura 3.23 – Diagrama EPC comprobación de conexión (PING)……..……. 81
Figura 3.24 – Diagrama EPC de la obtención de información de los adaptadores de red……. 85
Figura 3.25 – Diagrama entidad "encabezado" con sus atributos IP_origen, TCP, UDP o desconocido …… 88
Figura 3.26 – Diagrama entidad "paquetes" con sus atributos Id_paquetes, entidad, salida y errores…… 89
Figura 3.27 – Diagrama entidad "equipo" ……..…… 90
Figura 3.28 – Diagrama entidad "área" con sus atributos Id_area, Id_seg_red, departamento……. 90
Figura 3.29 – Diagrama E-R del tablero de control…..…… 91
Figura 3.30 – Proceso de la expresión del algebra relacional…… 93
Figura 3.31 – Pantalla de inicio de MySQL Workbench……... 96
Figura 3.32 – Creación de un nuevo modelo EER…… 97
Figura 3.33 – Creación del nuevo esquema "vb_mysql"…….. 97
Figura 3.34 – Editor de tablas de MySQL Workbench…….. 98
Figura 3.35 – Edición de tabla encabezado…… 98
Figura 3.36 – Activación del administrador de conexiones…….. 99
Figura 3.37 – Cuadro de dialogo administrador de conexiones…… 99
Figura 3.38 – Activación de la función avance de ingeniería…… 100
Figura 3.39 – Vista de la sentencias de SQL para la creación del esquema y la tabla de encabezado…….. 101
Figura 3.40 – Edición de la tabla paquetes, equipo y área…… 102
Figura 3.41 – Creación de las tablas paquetes, equipo y área dentro del SABD 103
Figura 3.42 – Botón agregar diagrama…….. 103
Figura 3.43 – Diagrama E-R creado en el SABD……. 104
Figura 3.44 – Diagrama a bloque de los controles de análisis de red……. 105
Figura 3.45 – Diagrama a bloques de la organización de los medidores del tablero de control…….. 110
Figura 3.46 – Barra de progreso utilizada para medir la capacidad del canal … 114
Figura 3.47 – Barra de progreso que muestra el porcentaje de equipos activos en la red ……. 120
Figura 3.48- Diagrama a bloques de los controles de usuario…….. 121

Capítulo 4. Pruebas y Resultados

Figura 4.1 - Interfaz gráfica del husmeador de paquetes…….. 129
Figura 4.2 - Resultado de la prueba de detección de dirección IP origen y destino…… 129
Figura 4.3 – Resultado de la prueba de detección de protocolos……. 130
Figura 4.4 – Resultado de la prueba análisis de encabezado de IP…… 131
Figura 4.5 – Resultados de la prueba análisis de encabezado de TCP…….. 132
Figura 4.6 – Resultados de la prueba análisis de encabezado de UDP……. 133
Figura 4.7 – Ventana de conexión a la red del CIITEC…… 135
Figura 4.8 – Ventana de conexión a la red inalámbrica del CIITEC…… 135
Figura 4.9 – Resultado de las pruebas de detección de paquetes de entrada y salida por un medio de transmisión cableado…… 136
Figura 4.10 – Resultados de la prueba de detección de paquetes de entrada y

salida con conexión inalámbrica…… 136
Figura 4.11 – Resultado de la prueba de comprobación de conexión con el servidor. 138
Figura 4.12 – Resultado de la prueba de comprobación de conexión con cambios en los valores de eco, tamaño del buffer, tiempo de vida…… 139
Figura 4.13 – Resultados de la prueba de respuesta inaccesible a la computadora destino…….. 140
Figura 4.14 – Resultado de la prueba de término del tiempo de respuesta…… 141
Figura 4.15 – Resultado de la obtención de información de los adaptadores de red… 143
Figura 4.16 – Resultado de la obtención de los adaptadores de red…… 144
Figura 4.17 – Vista de la aplicación generador de datos…… 146
Figura 4.18 – Ventana de ubicación de archivo con los registros…… 147
Figura 4.19 – Función de la aplicación Generador de datos…… 147
Figura 4.20 – Mensaje que indica que los archivos ya fueron generados en el archivo de datos…… 148
Figura 4.21 – Edición de tabla "test_area00" sin llave primaria…… 148
Figura 4.22 – Edición de tabla "test_area00" sin índice…… 148
Figura 4.23 – Sintaxis de SQL para crear la tabla "Test_area00" sin llave primaria y sin índice en la base de datos en el SABD…… 149
Figura 4.24 – Mensaje de creación exitosa en el SABD de la tabla "Test_area00"……. 149
Figura 4.25 – Edición de tabla "test_area01" con llave primaria y autoincremento……... 150
Figura 4.26 – Sintaxis de SQL para crear la tabla "Test_area01" con llave primaria en la base de datos en el SABD…… 150
Figura 4.27 – Mensaje de creación exitosa en el SABD de la tabla "Test_area01"…… 150
Figura 4.28 – Edición de tabla "test_area02" con llave primaria y auto incremento……. 151
Figura 4.29 – Edición del indicie de la tabla "test_area02" …….. 151
Figura 4.30 – Sintaxis de SQL para crear la tabla "Test_area02" con llave primaria y con índice en el SABD…… 152
Figura 4.31 – Mensaje de creación exitosa en el SABD de la tabla "Test_area02"……. 152
Figura 4.32 – Mensaje de error del SABD al intentar cargar mas de 125mil registros…… 155
Figura 4.33 – Vista completa del tablero de control para análisis de red en una red local…….. 164
Figura 4.34 - Resultado de la activación del menú archivo…… 165
Figura 4.35 - Resultado de la activación del menú herramientas…… 165
Figura 4.36 - Resultado de la activación de la aplicación adaptadores de red… 165
Figura 4.37 - Resultado de la activación de la aplicación HacerPING…… 166
Figura 4.38 - Resultado de la activación de la aplicación Estadísticas…… 166
Figura 4.39 - Resultados de la prueba de la sección análisis de tráfico de red. … 167
Figura 4.40 - Resultado de la prueba de la sección contadores de paquetes… 168
Figura 4.41 - Resultado de la prueba de la sección cantidad de equipos activos en la red…… 168
Figura 4.42 - Resultado de la prueba de la sección consumo de ancho de banda 169
Figura 4.43 - Resultado de los medidores de paquetes de entrada, salida y

errores……. 169
Figura 4.44 - Resultado de detección de equipos por departamento…… 170
Figura 4.45 - Pestaña reporte del tablero de control…… 171
Figura 4.46 - Resultado de la prueba de detección del comportamiento de red… 172
Figura 4.47 - Botón generar reporte del la pestaña reporte…… 173
Figura 4.48 - Resultado de la prueba generar reporte…… 173
Figura 4.49 - Reporte del comportamiento de red…… 174
Figura 4.50 - Pestaña de administración…… 175
Figura 4.51 - Sección de acceso del administrador…… 175
Figura 4.52 - Pestaña de administración activada…… 176
Figura 4.53 - Botones de selección para el despliegue de información…… 177
Figura 4.54 - Resultado de la consulta a la tabla posgrado…… 177
Figura 4.55- Ventana Agregar nodo…… 178
Figura 4.56 - Resultado de la inserción de un nuevo registro en la tabla posgrado…… 178
Figura 4.57 - Ventana modificar nodo…… 179
Figura 4.58 – Mensaje de modificación satisfactoria…… 179
Figura 4.59 – Resultado de la modificación del registro de la tabla posgrado… 180
Figura 4.60 – Resultado de la búsqueda de un registro dentro de la base de datos……... 180
Figura 4.61 – Resultado de la prueba de borrado de un registro dentro de la base de datos…… 181
Figura 4.62 – Resultado de la consulta a la tabla posgrado…… 182
Figura 4.63 Sección de configuración de parámetros de red…… 182
Figura 4.64 – Sección medidor de ancho de banda…… 183
Figura 4.65 – Resultado de la edición de alarma en el medidor de consumo de ancho de banda del medio de transmisión…… 183
Figura 4.66 – Resultado de la detección del evento de consumo de ancho de banda..... 184
Figura 4.67 – Sección medidor de paquetes de entrada…… 184
Figura 4.68 – Resultado de la edición de alarma del medidor paquetes de entrada…… 185
Figura 4.69 – Resultado de la detección del evento de paquetes de entrada…… 185

Índice de tablas:

Pág

Capitulo 2. Análisis de tráfico

Tabla.2.1 Resumen de la comparación entre los parámetros de herramientas de administración de red…… 32
Tabla 2.2Tabla comparativa entre HIDS, NIDS, pasivos y activos contra ListeningWire…… 33

Capitulo 3. Desarrollo de los módulos del tablero de control

Tabla.3.1 – Función Interfaz gráfica…… 45
Tabla.3.2 – Función husmeador de paquetes…… 47
Tabla.3.3 – Descripción de los espacios de nombres System en Visual Basic.NET…… 50
Tabla.3.4 – Captura de paquetes…… 51
Tabla.3.5 – Descripción por bits del tipo de servicio…… 53
Tabla.3.6 – Descripción de los bits de bandera…… 54
Tabla.3.7 – Función IPHeader…… 55
Tabla.3.8 – Función TCPHeader…… 61
Tabla.3.9 - Función UDPHeader…… 66
Tabla.3.10 – Función DNSHeader…… 71
Tabla3.11 - Descripción del contenido del espacio de nombres System.Net.NetworkInformation…… 75
Tabla.3.12 – Función Estadísticas…… 76
Tabla.3.13. – Función HacerPing…… 80
Tabla.3.14 – Descripción de la clase NetworkInterfaces…… 83
Tabla.3.15 – Función adaptadores de red…… 84
Tabla.3.16 Símbolos usados en la notación E-R…… 88
Tabla.3.17 Relación Equipo…… 93
Tabla.3.18 Relación de la consulta a la tabla Equipo…… 94
Tabla.3.19 – Función Husmeador de paquetes del entorno gráfico…… 106
Tabla.3.20 – Función contador de paquetes del tablero de control…… 107
Tabla 3.21 – Función Estadísticas del entorno gráfico…… 108
Tabla 3.22 - Función Hacer PING del entorno gráfico…… 108
Tabla 3.23 - Función adaptadores de red del entorno gráfico…… 109
Tabla 3.24 - Relación del valor porcentual de la barra de progreso con el tamaño del paquete de datos…… 112
Tabla 3.25Función medidores…… 114
Tabla 3.26- Función alarmas…… 115
Tabla 3.27 – Tabla de relación entre velocidad de transmisión con el tamaño de paquete y el valor porcentual de la barra de progreso…… 119
Tabla 3.28 - Función velocidad de transmisión…… 119
Tabla 3.29 – Función medidor de equipos activos en la red…… 120
Tabla 3.30 – Función barra de menú…… 122
Tabla 3.31 – Función inicio / alto…… 122

Tabla 3.32 – Función seleccionar interfaz…… 123
Tabla.3.33 – Función de los botones de edición de nodo…… 124
Tabla.3.34 – Función información y edición de nodo…… 125
Tabla 3.35 – Función filtrado de la función editor de nodo de red…… 126

Capitulo 4. Pruebas y Resultados

Tabla 4.1–Resumen de resultados de las pruebas del modulo husmeador de paquetes……... 134
Tabla 4.2 Resumen de resultados del modulo estadísticas…… 137
Tabla 4.3 – Tabla de resultados de las pruebas del modulo PING…… 142
Tabla 4.4 – Tabla de resultados del modulo adaptadores de red…… 145
Tabla 4.5 – Resultados de la prueba de inserción con la mayor cantidad posible de registros…… 154
Tabla 4.6 - Resultados de la prueba de inserción con una expresión propia del SADB para la mayor carga de registros…… 156
Tabla 4.7 – Resultados de la prueba de consulta con la mayor cantidad posible de registros…… 159
Tabla 4.8 - Resultados de prueba de consulta con una expresión propia del SADB para la mayor carga de registros…… 160
Tabla 4.9 - Nombre y distribución de tablas con respecto al numero de registros…… 161
Tabla 4.10 – Resultados de la prueba de consulta de dos tablas relacionadas con un campo…… 163
Tabla 4.11 – Resultados de las pruebas de los módulos del tablero de control…….. 186

Índice de algoritmos:

	Pág
Algoritmo 3.1 – Algoritmo del husmeador de paquetes (sniffer)……..	49
Algoritmo 3.2 - Algoritmo para el análisis del encabezado de IP ……	58
Algoritmo 3.3 – Algoritmo para el análisis del encabezado de TCP…….	64
Algoritmo 3.4 Algoritmo para el análisis del encabezado UDP……...	69
Algoritmo 3.5 – Algoritmo para el análisis el encabezado de DNS……..	74
Algoritmo 3.6 – Algoritmo para capturar paquetes de entrada / salida……..	78
Algoritmo 3.7 – Algoritmo para la comprobación de conexión…….	82
Algoritmo 3.8 – Algoritmo que adquiere las características del adaptador de red.	86

Índice de gráficas:

	Pág
Gráfica 4.1 – Comportamiento de la inserción con la mayor carga de registros con llave primaria……	154
Gráfica 4.2 – Resultado del comportamiento de la inserción de registros utilizando una expreción propia del SABD……	157
Gráfica 4.3 – Comportamiento de la consulta con la mayor carga de registros y con llave primaria……	159
Gráfica 4.4 - Comportamiento en la consulta de registros utilizando una expreción propia del SABD……	160
Gráfica 4.5 – Comportamiento de las consultas con dos tablas relacionadas con un campo, tabla 1 sin llave primaria y tabla 2 con llave primaria……	163

Capítulo 1

Introducción

1.1 Redes de datos.

Las redes de datos comparten información, servicios y recursos, estas se clasifican según su área de cobertura como las redes de área local (LAN por sus siglas en inglés *Local Area Network*) que se ubican en lugares relativamente pequeños como hogares, edificios, etc, las redes inalámbricas (WLAN por sus siglas en inglés *Wireless Local Area Network*) que tienen la misma cobertura que una LAN con la diferencia de proporciona al usuario movilidad sin perder la conectividad, las redes de área metropolitana (MAN por sus siglas en inglés *Metropolitan Area Network*) que abarcan áreas geográficas extensas y las redes de área amplia (WAN por sus siglas en ingles *Wide Area Network*) que tienen un área de cobertura mayor proveyendo de servicio a un país o continente. Cada una de estas redes tiene la capacidad de integrar múltiples servicios mediante la transmisión de datos, voz y video, sobre diferentes medios de transmisión como cable par trenzado, fibras ópticas, canal de difusión[1], etc, empleando dispositivos de conexión como interruptores (switches), ruteadores, moduladores – demoduladores, filtros, etc.

[1] Medio de comunicación por donde viaja la forma de onda de la señal (portadora de información) del transmisor al receptor.

1.2 Administración de LAN.

La administración de red de área local es una tarea compleja pues requiere de un conocimiento detallado de la estructura física y lógica de la misma, de los equipos que la conforman, de los protocolos de comunicación, etc. Por esta razón los administradores de red o especialista en Tecnología de la Información (TI) utilizan herramientas de administración que les ayudan a mantener la red en operación continua, evitando el paro de operación. Las herramientas de administración de red son sistemas complejos que presentan los eventos que ocurren en el consumo del ancho de banda por el envío y recepción de datos en el medio de transmisión.

1.3 Tablero de control.

Una nueva tecnología que ayuda a la administración de red son los tableros de control o *Dashboard* que es una herramienta sofisticada que mediante un entorno gráfico compuesto de instrumentos muestra la información de forma clara e intuitiva permitiendo al usuario asociar imágenes y objetos con funciones que permiten obtener información del comportamiento de red, la finalidad de esta herramienta consiste en proporcionar un entorno visual sencillo para permitir la interacción del usuario con el tablero de control, sin la necesidad de introducir comandos gracias a que el tablero de control se encarga de adquirir, procesar, analizar y presentar la información para que el usuario pueda observar fácilmente el comportamiento de red.

El objetivo de este trabajo es crear una aplicación de administración de red que permita al especialista en tecnología de la información tener un mejor control de la

misma mediante una aplicación que muestre de forma gráfica el comportamiento de los datos que circulan en ella. El obtener información de las condiciones de transmisión y recepción de datos con una sola herramienta de administración hace posible disminuir los tiempos de respuesta y evitando la degradación de la red.

1.4 Planteamiento del problema.

Como se menciono en la introducción existen diferentes tipos de red que comparten servicios y recursos esto implica que se tiene que utilizar el ancho de banda del medio de transmisión para la transferencia de datos, cuando un equipo necesita enviar información a otro, se añade un encabezado por cada capa del modelo OSI[2] hasta que llega a la capa de enlace de datos, donde son encapsulados los encabezados junto con los datos, a esto se le llama paquete de datos, antes de enviarlo se almacena temporalmente en un espacio de memoria (buffer de salida) del adaptador de red[3] hasta que el medio de transmisión se encuentre desocupado para enviar el paquete. Sí el medio de transmisión se encuentra con demasiado flujo de información (tráfico) el adaptador de red acomoda los paquetes de datos dentro del buffer de salida, si este medio no se desocupa se comienza a llenar el espacio de memoria del adaptador de red provocando que se eliminen algunos paquetes de datos para generar espacio en el buffer de salida, provocando la perdida de información por la saturación en el medio de transmisión, por esta razón la detección oportuna del consumo excesivo del ancho de banda del medio de transmisión permite prevenir la saturación y posible degradación de la red.

[2] Para mayor información remitirse al apéndice “A”
[3] Para mayor información remitirse al apéndice “B”

1.5 Objetivo General

Desarrollar una aplicación gráfica, de uso sencillo para el análisis de tráfico de red que permita tener un monitoreo continuo del consumo del ancho de banda en una Red de Área Local (LAN), generando alertas al ser rebasado el nivel predeterminado del consumo del ancho de banda u otras condiciones predefinidas.

Los objetivos particulares son los siguientes:

1) Utilizar el protocolo TCP/IP, UDP y DNS.
2) Crear una base de datos para almacenar el comportamiento de red.
3) Detectar tráfico en una Red de Área Local.
4) Detectar el consumo de ancho de banda.
5) Generar alertas al rebasar el nivel predeterminado del consumo del ancho de banda.
6) Generar reportes de auditoría.

1.6 Alcances del proyecto.

Los alcances del presente desarrollo se dividen en tres partes descritos en los siguientes puntos:

1) Alcances indispensables: Son los objetivos esenciales con los cuales la aplicación debe cumplir:
   - Detectar el tráfico de red utilizando la pila de protocolos TCP/IP que es un estándar de comunicación el cual permite obtener información de la red.

- Almacenar el comportamiento de red en una base de datos con la información necesaria para localizar una falla en la misma.
- Detectar el consumo del ancho de banda del medio de transmisión en la red para evitar su degradación y el paro de operación.
- Generación de alertas al sobrepasar el consumo predeterminado del ancho de banda.
- Generación de reportes de auditoría para certificar niveles de operación óptimos y en un momento dado, justificar la inversión de nuevo equipo.

2) Alcances deseables: Son aspectos que se pueden incluir para aumentar las prestaciones en la aplicación:
   - Ejecución en tiempo real de la aplicación para mejorar el rendimiento de las alarmas para que sea lo más oportuna posible.
   - Detección de accesos no autorizados para el control de envío y recepción de información en los correos electrónicos, servidores y páginas de web.
   - Generar estadísticas de las situaciones anómalas que se presentan en la red, con la finalidad de predecir comportamientos que puedan causar su degradación.
   - Consultar la aplicación vía Internet para que la supervisión, consultas y la corrección de los problemas en la red se resuelvan vía remota.
   - Utilizar el protocolo SMTP[4] para el envío de alarmas y reportes por correo electrónico, IGMP[5] para el envío de mensajes de error vía Internet.

[4] Protocolo simple de transferencia de correo (*Simple Mail Transfer Protocol SMTP*), para saber la definición consulte el glosario.
[5] Protocolo de Mensajes de Control de Internet (*Internet Control Message Protocol*), para saber la definición consulte el glosario.

3) Alcances opcionales: Estos alcances no son necesarios para el funcionamiento del desarrollo, sin embargo incrementan el número de herramientas y mejoran la apariencia de la aplicación los cuales se describen a continuación:
   - Instalación de agentes en los equipos de cómputo para tener un mayor control de los equipos.
   - Detectar maquinas virtuales para el monitoreo de VLAN.
   - Conexión de escritorio remoto para solucionar problemas a larga distancia en equipos instalados en la red.
   - Crear un instalador para que la configuración de la aplicación sea lo menos compleja posible.

1.7 Estructura General del documento.

El capítulo dos se presenta el análisis de tráfico, se describen algunas herramientas de uso común y se muestran las ventajas y desventajas de cada una de ellas, finalizando con una comparación entre las aplicaciones utilizadas para administración y el presente desarrollo.

El capítulo tres explica el desarrollo de los distintos módulos que componen el tablero de control para el análisis de una red de área local, la implementación de la captura de datos, las aplicaciones de captura de paquetes de datos y prueba de conexión de equipos de red, continuando con el desarrollo de la base de datos y finaliza con el diseño e implementación del entorno gráfico, resultando así un software unificado con las herramientas necesarias para la administración de red.

El capitulo cuatro presenta los resultados de las pruebas de cada uno de los módulos; adquisición de datos de red, base de datos e interfaz gráfica, realizadas en la red del Centro de Investigación e Innovación Tecnológica (CIITEC) del Instituto Politécnico Nacional.

El capitulo cinco presentan las conclusiones a las que se llegaron después de haber desarrollado y probado el tablero de control, los objetivos que se cumplieron, lo aprendido de la maestría en tecnología avanzada y el trabajo que se tiene hacia futuro, seguido de las referencias consultadas durante el inicio, desenlace y final del este trabajo.

1.8 Resumen.

El contar con una herramienta sofisticada como son los tableros de control permite al especialista de tecnologías de la información (TI) obtener gráficamente la información de los eventos que saturan el ancho de banda del medio de transmisión de una LAN, por esta razón el objetivo de este trabajo es desarrollar una aplicación que detecte el tráfico de red utilizando los protocolos IP, TCP, UDP y la aplicación DNS para detectar el consumo de ancho de banda del medio de transmisión generando alertas al rebasar los limites permitidos, almacenando toda esta información en una base de datos para generar reportes del comportamiento de red.

Capítulo 2

Análisis de tráfico

Éste capítulo explica el análisis de tráfico entre computadoras así como las herramientas utilizadas para controlar el flujo de información entre ellas, la sección 2.1 contiene los antecedentes de la comunicación entre computadoras, la sección 2.2 presenta la administración de red y la sección 2.3 contiene las herramientas de análisis de tráfico, la sección 2.4 presenta una comparación entre las herramientas de análisis de tráfico y el presente desarrollo. La sección 2.5 contiene un resumen del capítulo.

2.1 Antecedentes de la comunicación entre computadoras.

La comunicación entre computadoras surge por la necesidad de intercambiar información entre equipos de cómputo ya que es costoso y lento el traslado de medios físicos de almacenamiento que contienen información. Además del problema de la distancia, existe el problema del manejo de diferentes protocolos de comunicación y diferentes sistemas operativos, ya que cada fabricante define su propio protocolo de comunicación, situación que complica el manejo de la información, y se entiende la necesidad de crear un estándar de comunicación y es así que en el año de 1974[1] nace el Protocolo de Control de Transmisión / Protocolo de Internet (TCP/IP por sus siglas en ingles, *Transmission Control Protocol / Internet Protocol*) el cual proporciona la comunicación entre equipos de cómputo.

Con el uso inicial del protocolo TCP/IP la red de computo se limitó para el uso académico y no se prestó mucha atención en la administración, seguridad de transmisión, recepción y almacenamiento de los datos, hasta que la cantidad de maquinas y usuarios conectados a la red creció, generando problemas por la saturación

del medio de transmisión provocando la perdida de información y la degradación de la información.

2.2 La administración de red.

Los administradores son personas capacitadas para resolver problemas dentro de la red, investigando y conociendo a fondo los procesos de comunicación entre equipos de cómputo tanto de forma física como lógica (Hardware y Software), y son los encargados de mantenerla en optimas condiciones, evitando que se llegue a niveles inaceptables en la operación como un bajo rendimiento en los procesos de carga y descarga de información, procesos de envío de información inconclusos, etc. Lo que llevaría a degradar la red.

El administrador de red requiere de un conocimiento sólido en la infraestructura de red (ruteadores, interruptores (switch), concentradores (HUB's), moduladores, demoduladores, filtros, etc.), protocolos de comunicación (modelo OSI[6], TCP/IP, IPX, etc.), topologías de conexión de red (anillo, bus, estrella, combinadas), medios de transmisión (Cableado estructurado, tipos de cable, fibras ópticas de comunicación, modos de propagación de las señales o puntos de acceso inalámbrico (WAP o AP por sus siglas en ingles *Network Access Point*), etc.), tipo de señales (Voz por IP (VoIP), audio, video, etc.), para resolver problemas en la red, además tiene que estar al tanto de los ataques[7] que puede sufrir la red, por software mal intencionado, por virus informáticos, accesos no autorizados, etc.

Se comprende que el esfuerzo humano para tener una red bien administrada no es suficiente y por lo tanto se debe de contar con una herramienta que ayude al administrador con su trabajo, mostrando información oportuna del comportamiento de red y que la información presentada sea clara al describir su comportamiento, para evitar saturar al administrador con información redundante que puede impedir que visualice las alertas para tomar acciones preventivas, entonces el trabajar con distintas herramientas de administración como tableros de control (*Dashboard*[8]), husmeadores de

[6] Para mayor información véase el apéndice "A."
[7] Es la violación a la seguridad, esto se realiza con fines de beneficio personal o para hacer daño.
[8] Dashboard.- es el nombre en inglés de tablero de instrumentos

paquetes de datos (*sniffer*[9]), sistemas de detección de intrusiones[10] (SDI, o IDS [29] por sus siglas en ingles, *Intrusion Detection System*), software de administración, etc. obstaculiza la visión del administrador de los problemas que presenta la red, evitando la ejecución de una acción preventiva para prevenir el bajo desempeño de la red. En la siguiente sección se describen algunas herramientas de administración.

2.3 Herramientas de administración de red.

En este apartado se analizarán las diversas herramientas que generalmente utiliza un administrador de red, mencionando ventajas y desventajas, la sección 2.3.1 presenta el tablero de control, la sección 2.3.2 muestra el husmeador de paquetes y la sección 2.3.3 presentan los sistemas de detección de intrusiones.

2.3.1 Tablero de control.

El tablero de control es una interfaz gráfica de usuario que se conoce como *Dashboard*, es un programa que utiliza un conjunto de imágenes y objetos gráficos para presentar información en una pantalla de forma visual e intuitiva, su objetivo principal es minimizar errores, mejorar el uso de la infraestructura de red y hacer más eficiente la administración de red.

En la figura.2.1 se muestra como se ve un tablero de control completo con cada una de sus secciones para mostrar la información al administrador de red, la forma en como se encuentra acomodada la información puede variar con respecto a cada fabricante, para este caso en particular se tomo la pantalla del tablero de control del fabricante ManageEngine Power IT ahead de su aplicación OpManager.

[9] Sniffer.- Olfateador (sniff: Aspiración)[26]
[10] Es cualquier persona que viola la seguridad de un sistema informático, esto lo realiza con fines de beneficio personal o para hacer daño.

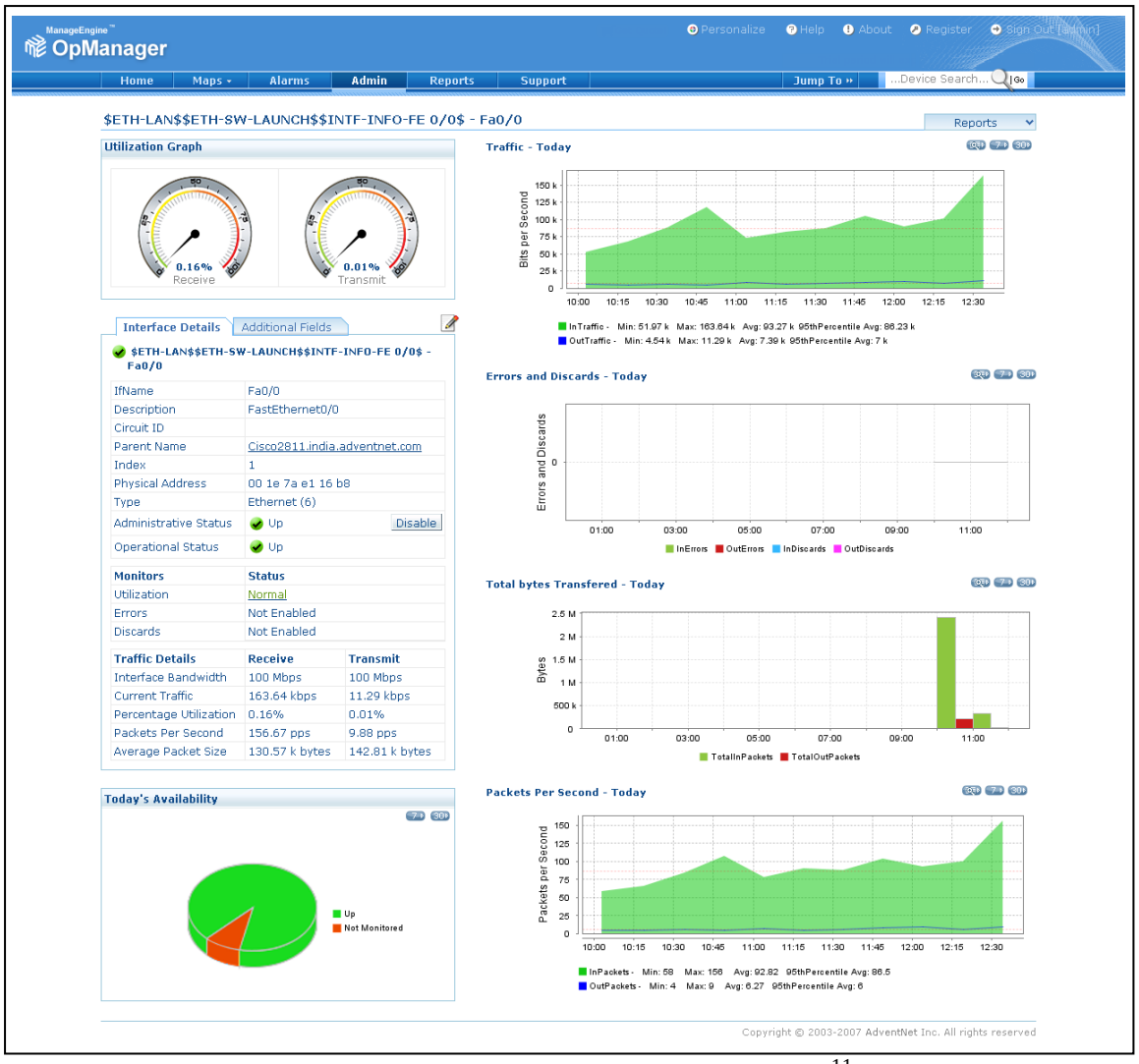


Figura.2.1 - Tablero de control OpManager[11]

Los tableros de control están constituidos por diferentes secciones que describen un determinado aspecto en la red[12] como se describe en los siguientes puntos:

- Medidores de transmisión y recepción de paquetes de datos.
- Detalles de interfaces de red.
- Cobertura de supervisión por día.
- Gráficas de comportamiento de red.
    - Tráfico por día.
    - Errores descartados.
    - Total de bytes transferidos por día.
    - Paquetes por segundo.

[11] OpManager es un software de administración de red que esta diseñado para administrar dispositivos de red, desarrollado por MenageEngine Powering IT ahead.
[12] Los nombres e imágenes son tomados del software OpManager y solo son utilizados como referencia.

Visto el tablero de control de forma general, a hora se describirá cada una de las secciones que lo componen para tener una idea más precisa, la sección 2.3.1.a contiene los medidores de transmisión y recepción de datos, la sección 2.3.1.b presenta los detalles de interfaces de red, la sección 2.3.1.c contiene la capacidad de supervisión por día, la sección 2.3.1.d describe las gráficas de comportamiento de red y la sección 2.3.1.e presenta las ventajas y desventajas de un tablero de control.

2.3.1.a Medidores de transmisión y recepción de datos.

Muestra los porcentajes de la taza de transferencia (MBytes) (Figura.2.2) de paquetes que entran y salen en la red, muestra al administrador una vista inmediata de los niveles de velocidad indicando si los parámetros son normales o si tiene una sobre carga de trabajo por alguna anomalía como puede ser la saturación en los medios de transmisión, el envío constante de paquetes de datos de una computadora en la red, etc.

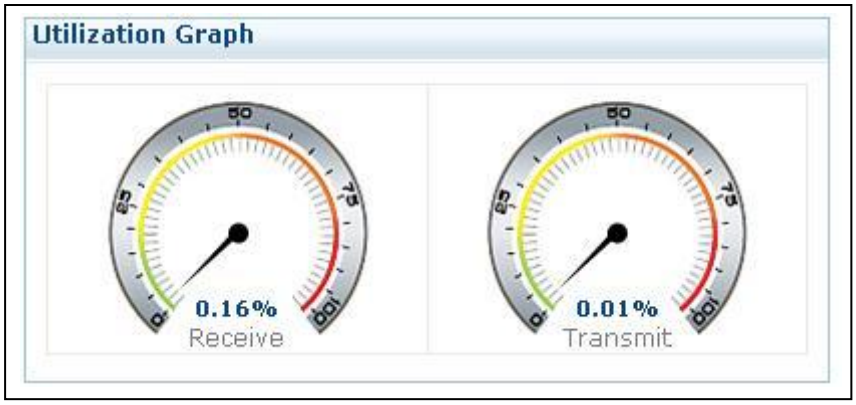


Figura.2.2 - Gráficas de transmisión y recepción.

2.3.1.b Detalles de interfaz de red.

Esta sección muestra las características de las interfaces de red instaladas en la computadora o equipos instalados en la red presentando una descripción de sus parámetros como:

- Nombre del parámetro o equipo.- Este es el nombre o la dirección IP con el cual se reconoce al dispositivo de red.

- Dirección física.- Esta es una dirección única del dispositivo (dirección MAC[13]).
- Tipo.- Se refiere al tipo de dispositivo, puede ser un switch, enrutador, concentrador, etc.
- Estado.- Muestra el estado del dispositivo ya sea encendido o apagado
- Detalles de tráfico.- Muestra el porcentaje en mega bits por segundo (Mbps) de transmisión y recepción de datos en el dispositivo.

Esto indica al administrador el estado y comportamiento de los equipos (fig. 2.3) para llevar el control tanto de su actividad y en algunos casos como inventario.

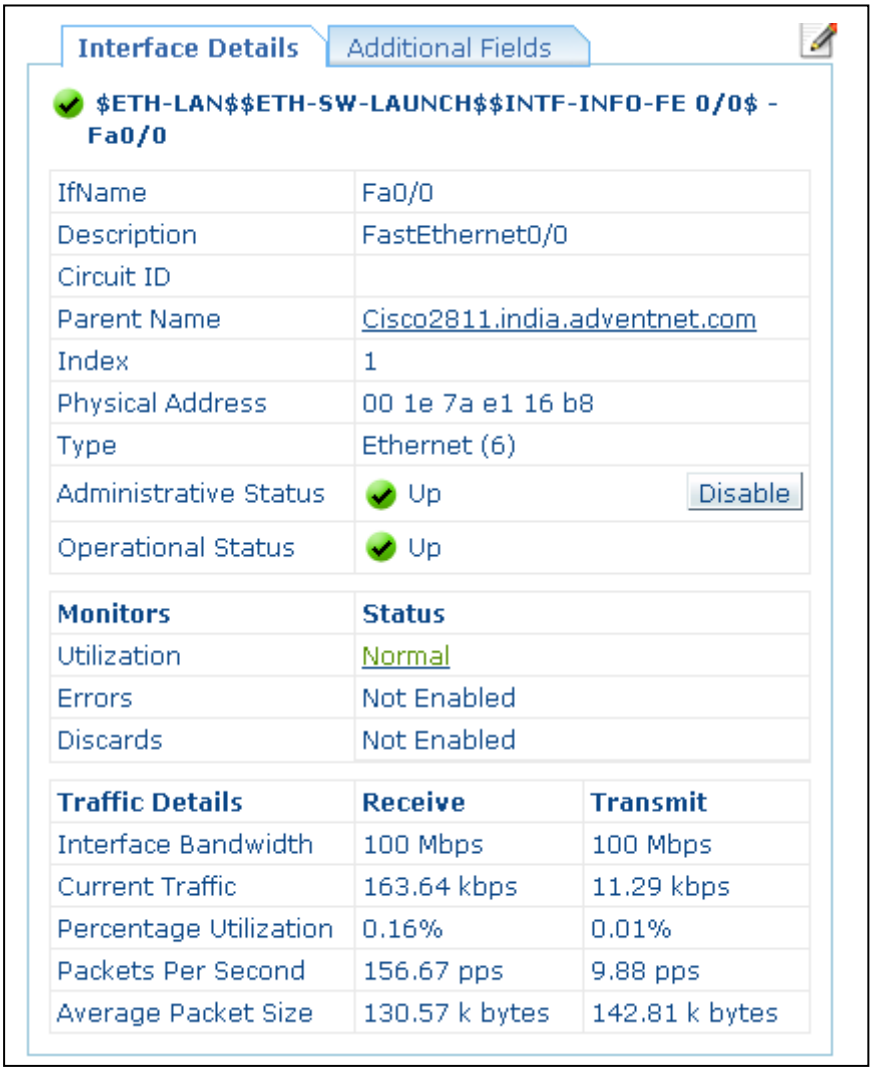


Figura.2.3 - Detalle de interfaz.

[13] MAC (control de acceso al medio, *Media Access Control*) es un identificador de 48 bits único de una tarjeta de red.

2.3.1.c Capacidad de supervisión por día.

La supervisión o monitoreo de la red (fig.2.4) se refiere al alcance que tiene el tablero de control, mostrando con colores (en este caso con color verde) todos los equipos que son monitoreados y los que no se pueden monitorear (en este caso color naranja) que son equipos con los que no se a logrado tener comunicación debido a diversas razones como lo indican los siguientes puntos:

1) Equipo apagado.
2) El equipo esta desconectado de la red.
3) Desactivación del agente de protocolo simple de administración de red (SNMP[14] por sus siglas en ingles, *simple network management protocol*).
4) Que el equipo éste infectado con un virus y no permita al tablero de control monitorearlo, etc.

Son diversas las causas por las cuales algunos equipo no pueden ser vistos por el tablero de control y no es la finalidad de esta sección describir cada una de sus causas, sino mas bien mencionar sólo algunas casos.

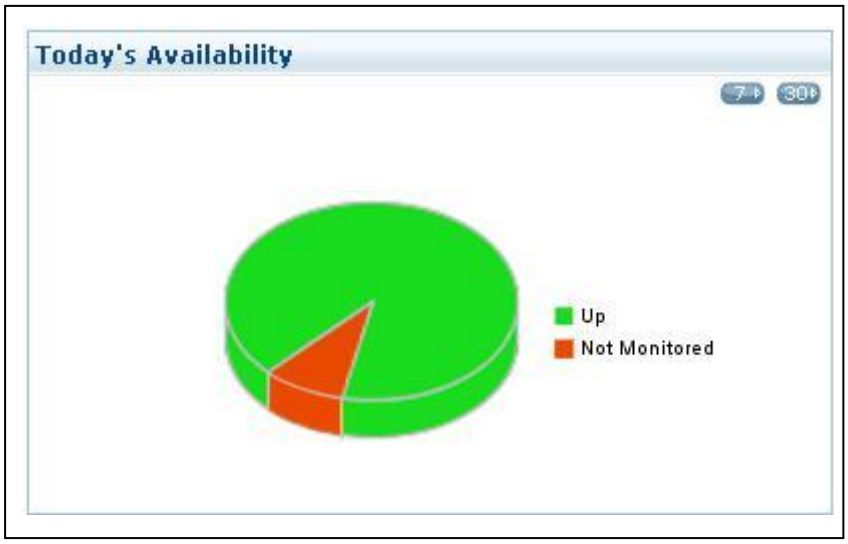


Figura.2.4 - Grafica de viabilidad o cobertura por día.

2.3.1.d Gráficas de comportamiento de red.

En esta sección se tienen varias gráficas que muestran el tráfico generado en la red por día (fig.2.5) donde se presenta información del tráfico y la utilización de los puertos, identificando a los dispositivos que utilizan más recursos en la red.

[14] Mas adelante se abundara mas en el tema.

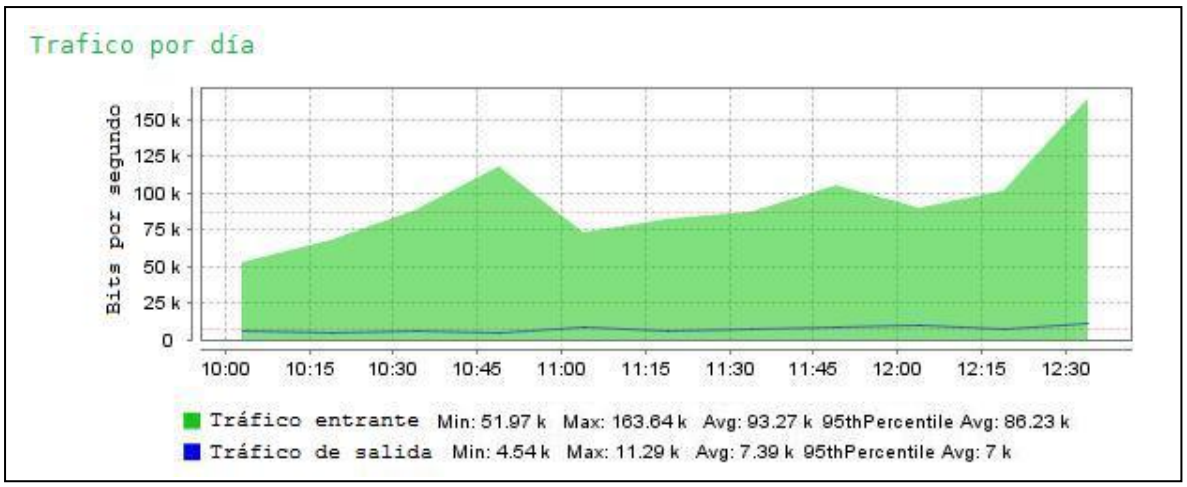


Figura. 2.5 - Tráfico del día.

La gráfica de errores descartados por día (fig.2.6) muestra la cantidad de errores en el sistema por fallas en las conexiones o por interferencia en los medios de transmisión.

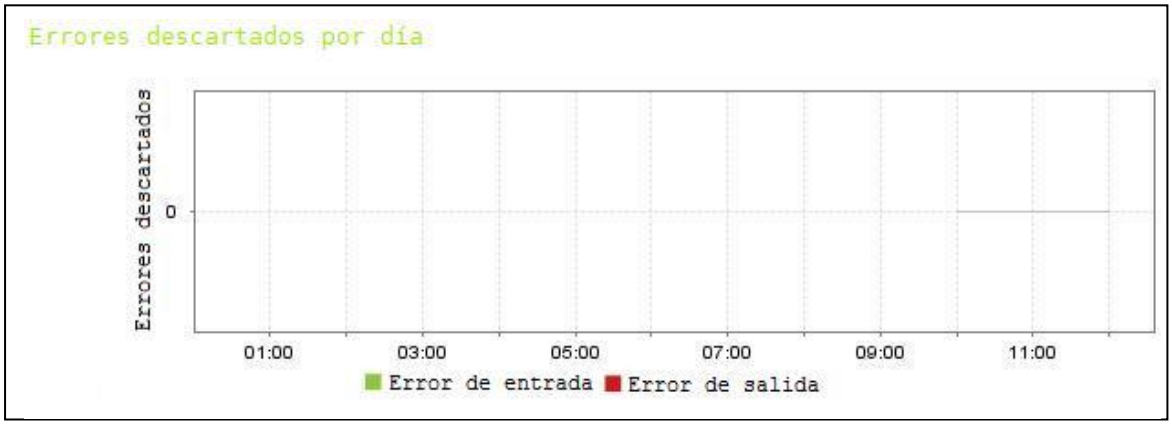


Figura.2.6 - Errores descartados por día.

El total de bytes transferidos (fig.2.7) muestra el tiempo y la cantidad de bytes que se tiene en los medios de transmisión para el envío y recepción de datos que circulan por la red.

Figura.2.7 - Total de bytes transferidos por día.

La gráfica de paquetes por segundo del día (fig.2.8), presentan una estadística de forma gráfica de los máximos, mínimos y el promedio del comportamiento de los paquetes de entrada y salida de la red.

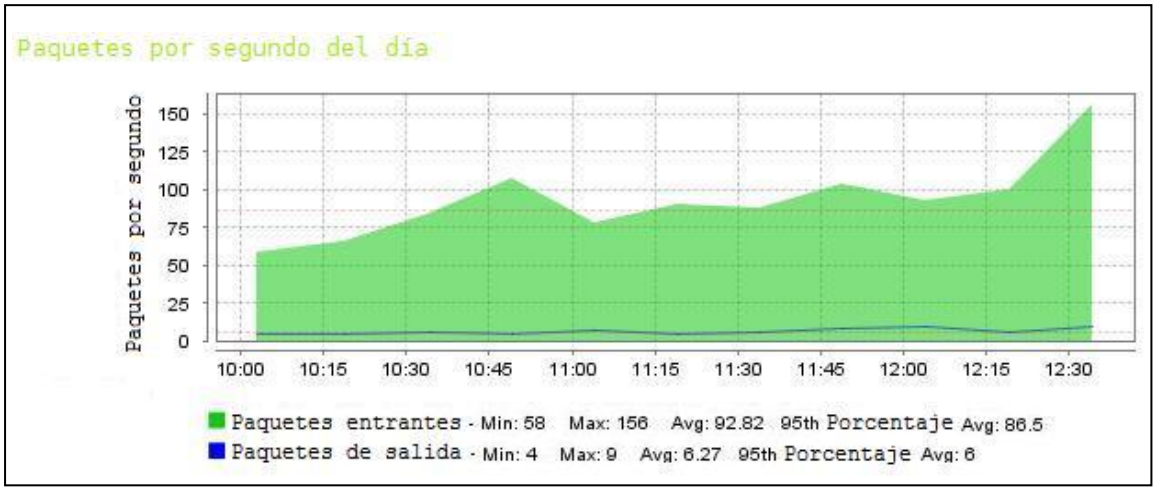


Figura.2.8 - Paquetes transferidos por segundo en el día.

La figura 2.8 se muestra la grafica de paquetes transferidos por segundo en el día, permiten al administrador de red tener una visión en tiempo real de la transmisión y recepción de los datos que circulan por la red.

2.3.1.e Ventajas y desventajas de un tablero de control.

Ventajas:

- Cuenta con un ambiente gráfico intuitivo y amigable hacia el usuario.
- Describe el comportamiento de la red por medio de estadísticas y gráficas.
- Es fácil de manejar.
- La información se presenta en una sola ventana sin saturar el monitor.

Desventajas:

- No cuenta con ningún tipo de alerta para el administrador de red.
- Alguna de la información que presenta es redundante.
- No cuenta con una base de datos.
- Se debe tener una constante supervisión del tablero para poder prevenir la degradación de la red.

2.3.2 Husmeador de paquetes.

El husmeador de paquetes o *sniffer* es un programa que captura los paquetes de datos que circulan por la red sin interferir con los procesos de envío y recepción de paquetes de datos, obtiene información como IP origen, IP destino, tiempo de conexión, tipo de protocolo utilizado, información transmitida de una computadora a otra, etc. (fig.2.9). La obtención de esta información es valiosa para el administrador de red por que así tiene el control del flujo de información que cruza por los medios de transmisión, esta información se obtiene mediante las conexiones que realiza el husmeador entre las capas de protocolos.

No.	Tiempo	Fuente	Destino	Protocolo	Información
1	0.000000	192.168.2.3	mail.packet0.com	TCP	1061 › http [SYN] Seq=0 Ack=0 Win=16384 Len=0 MSS=1460
2	0.063590	mail.packet0.com	192.168.2.3	TCP	http › 1061 [SYN, ACK] Seq=0 Ack=1 Win=5840 Len=0 MSS=1380
3	0.063665	192.168.2.3	mail.packet0.com	TCP	1061 › http [ACK] Seq=1 Ack=1 Win=16560 Len=0
4	0.064056	192.168.2.3	mail.packet0.com	HTTP	GET / HTTP/1.1
5	0.163470	mail.packet0.com	192.168.2.3	TCP	http › 1061 [ACK] Seq=1 Ack=399 Win=6432 Len=0

Figura.2.9 - Información de una captura típica de un husmeador de paquetes.

Para que el husmeador capture toda esta información es necesario que se coloque la tarjeta de red[15] en modo promiscuo, esto es, permitir a la tarjeta de red que capte todos los paquetes de datos aun que no sean dirigidos a ella, de esta manera la computadora será capaz de ver todos los datos transmitidos, permitiendo al husmeador comenzar a hacer la lectura.

Algunos de sus usos se presentan en los siguientes puntos:

- Captura de contraseñas y nombres de usuario en la red.
- Análisis de fallos en las comunicaciones de red.
- Mediciones del tráfico de red para detectar posibles cuellos de botella.
- Para desarrolladores en aplicaciones cliente – servidor, permitiendo analizar la información real que se transmite en la red.

Uno de los husmeadores mas utilizados por los administradores de red es wireshark que realiza la captura de paquetes de datos mediante winpcap[16] que es una herramienta de conexión entre capas, y esta herramienta incluye aspectos como son filtros de

[15] Para mayor información del adaptador de red remítase al apéndice “C”.
[16] Winpcap funciona en sistemas operativos como Unix y Windows.

paquetes de datos, un motor de generación de estadísticas de red y soporte de captura remoto.

Un husmeador se pude conectar de distinta maneras en una red para así capturar las tramas de datos en diferentes puntos y tipos de redes, a continuación se presentan las diversas formas en que se puede conectar un equipo con la herramienta de captura de datos instalada, la sección 2.3.2.a presenta la captura por Ethernet compartido, la sección 2.3.2.b contiene la captura sobre una maquina, la sección 2.3.2.c presenta la captura utilizando un concentrador Ethernet, la sección 2.3.2.d contiene el modo monitor de captura en el switch, la sección 2.3.2.e presenta la captura utilizando la maquina en el medio, la sección 2.3.2.f contiene la captura utilizando una llave de red, la sección 2.3.2.g presenta la captura con hombre en el medio o intermediario y la sección 2.3.2.h contiene la comparación de ventajas y desventajas del husmeador de paquetes.

2.3.2.a Captura por Ethernet compartido.

Las redes Ethernet utilizan dispositivos para compartir la información como los concentradores (HUB's) para conectar todos los nodos juntos, esto significa que todos los paquetes pueden recibirse por todos los nodos de red. Por consiguiente si una computadora tiene instalado la herramienta de captura de datos y el adaptador de red esta configurado en modo promiscuo, todos los paquetes que se encuentren circulando por esa red pueden ser vistos por ese adaptador (fig.2.10).

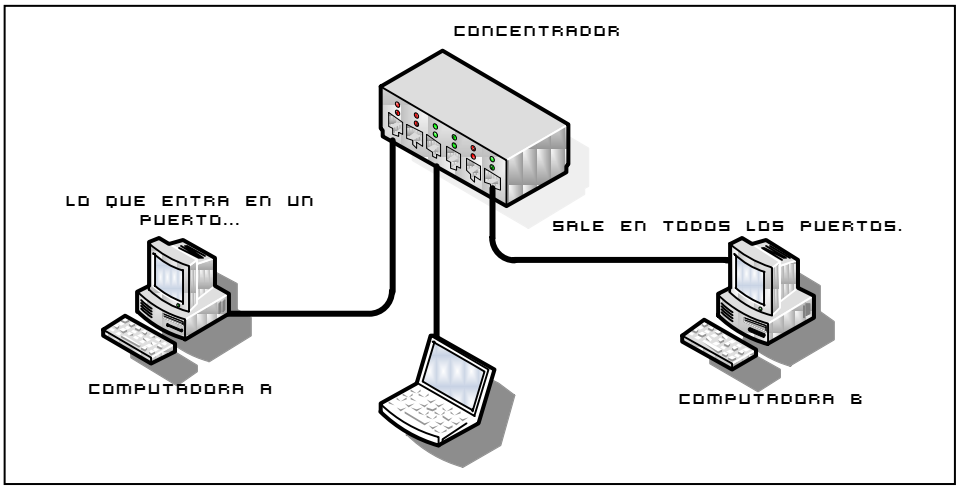


Figura.2.10 - Captura de paquetes de datos en una red con un concentrador.

En la actualidad estos dispositivos ya no se usan frecuentemente por que sus velocidades de operación ya no son compatibles con las velocidades que se manejan en la redes actuales (1GB a 10GB) por lo que se evita su uso y se emplean en su lugar los interruptores (switches) para conectar los nodos juntos. Esto optimiza la red, pero complica la captura de las tramas de datos (los interruptores pueden interferir el tráfico visto en uno de sus puertos) por que el switch puede dirigir la dirección o direcciones a un puerto específico. Como los paquetes no son enviados a todos los equipos conectados en el switch no se puede obtener todos los paquetes de datos.

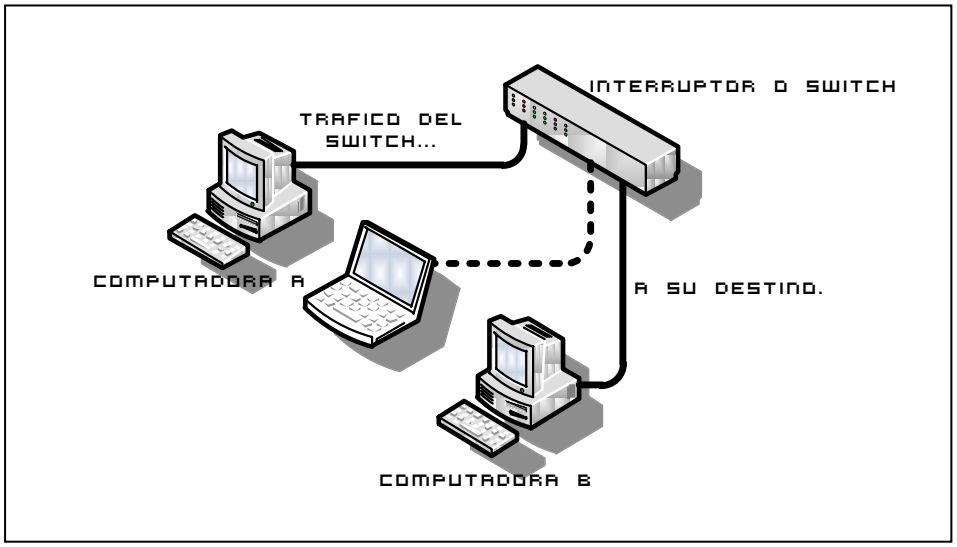


Figura.2.11 - Paquetes de datos no capturados por la intervención del switch.

En la figura 2.11 se observan tres equipos dos conectados al switch y uno con una conexión limitada, las computadoras A y B pueden transferir datos pero la computadora personal no, la razón es que el switch solo dirige los datos a las direcciones de las computadoras que tienen configuradas en su tabla de direcciones.

2.3.2.b Captura sobre una máquina.

Si solo se necesita capturar los paquetes de datos en una computadora, lo único que se tiene que hacer es instalar la herramienta de captura de paquetes en una maquina que éste directamente conectada al switch como lo muestra la figura.2.12. La ventaja de utilizar esta configuración es que es fácil de usar por que la captura datos se lleva acabo en un equipo, pero con la desventaja de que no es posible capturar los demás paquetes

de datos que circulan por la red perdiendo los demás paquetes de datos que circulan por la red.

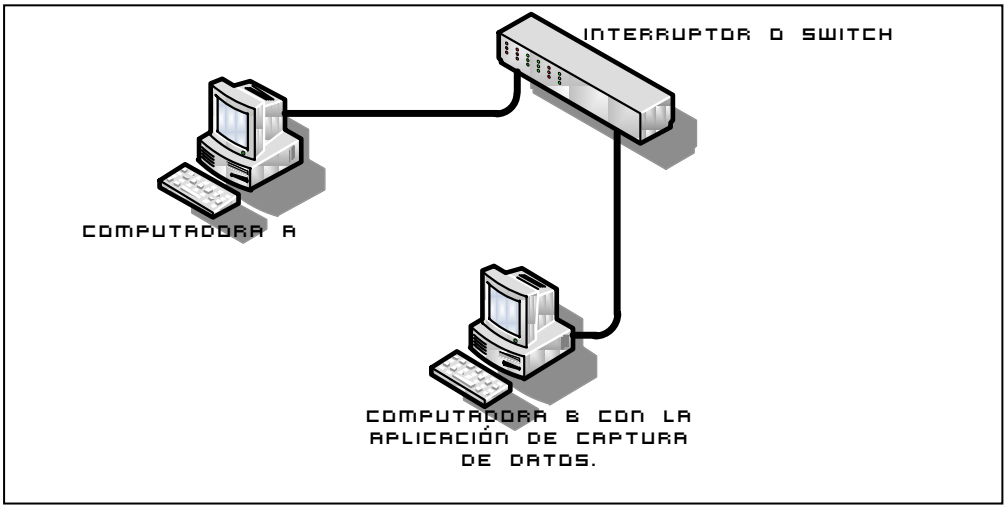


Figura.2.12 - Captura solo en el equipo de interés.

En la figura 2.12 se observa la conexión de los dos equipos en el switch, y solo son capturados los paquetes de datos un equipo limitando al husmeador de paquetes a captar la información en un solo punto en la red.

2.3.2.c Captura usando un concentrador Ethernet (HUB Ethernet).

Se coloca un HUB en la línea Ethernet donde se desea capturar la información de la red (fig.2.13), esto es colocar el HUB entre el switch y un segmento de red o computadora, entonces el HUB proporciona los paquetes de datos de esa subred en el cual si se puede obtener la información que circula por la subred. La ventaja de utilizar este método es que se captura todo el tráfico que circula por toda la subred, pero la desventaja es que afectan el tráfico de la red formando cuellos de botella como se menciono anteriormente por la diferencia de velocidades de transmisión.

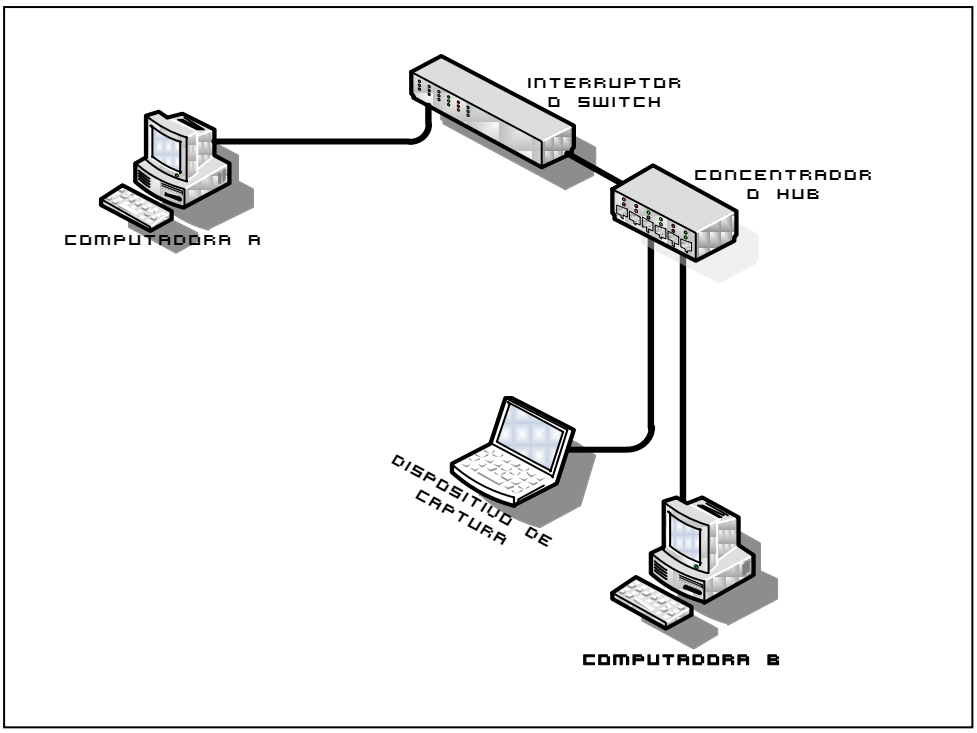


Figura.2.13 - Conexión del HUB entre el switch y la computadora.

2.3.2.d Modo monitor de captura en el switch.

En éste modo de monitoreo se tiene un puerto dedicado en el switch para el dispositivo de captura (fig.2.14), usando la dirección que proporciona el switch se puede seleccionar un puerto a supervisar, donde se realiza la captura de los paquetes de datos, se debe de tener precaución al utilizar éste método ya que si no se cuenta con un equipo por lo menos tan rápido como el puerto monitoreado es posible que se pierdan algunos paquetes de datos, la mejora que presenta éste modo de monitoreo es que ya se tiene conectado el equipo de captura de datos directamente al switch, la desventaja es que de igual forma que los métodos anteriores solo se obtiene una pequeña parte de la información que circula por la red, en éste caso los del puerto al que se esta conectado.

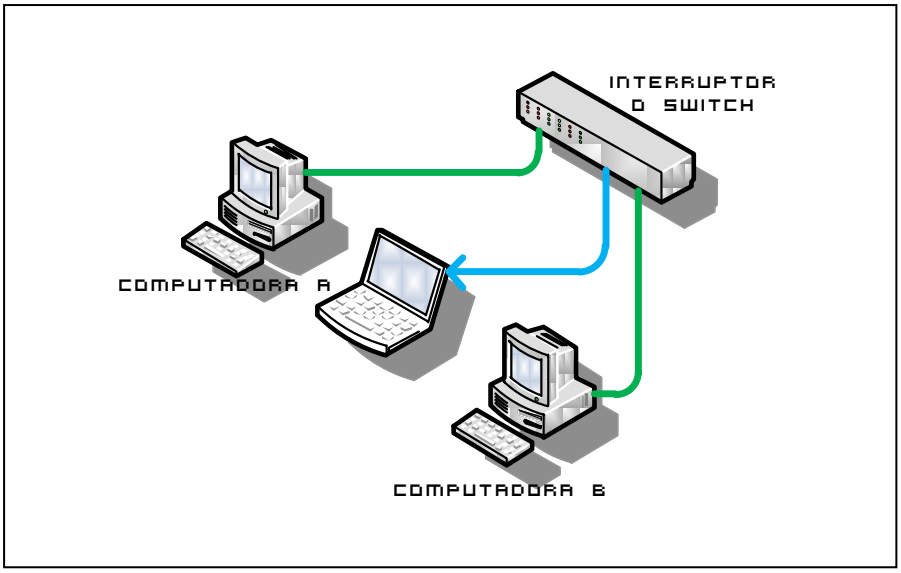


Figura.2.14 - Dispositivo de captura conectado a un puerto del switch.

En la figura 2.14 se observa que se encuentran tres equipos conectados en el interruptor y que ahora la computadora con la herramienta de captura de datos instalada se encuentra conectada para poder realizar la captura de datos en un puerto del switch.

2.3.2.e Captura usando la máquina en el medio.

Para utilizar éste método de captura se necesita una computadora con dos adaptadores de red, para utilizarlas como un puente transparente, capturando todo el tráfico de una sola computadora (fig.2.15) de un segmento de red. El puente es transparente en la línea de IP y protocolos similares, y es casi transparente a nivel de Ethernet creando un pequeño retraso en la retransmisión de los paquetes de datos, mejorando la velocidad de captura en el dispositivo y evitando la perdida de paquetes de datos, la ventaja es que se maneja la misma velocidad que utiliza el switch, la desventaja es que sólo se obtiene la información en un sólo extremo de la red.

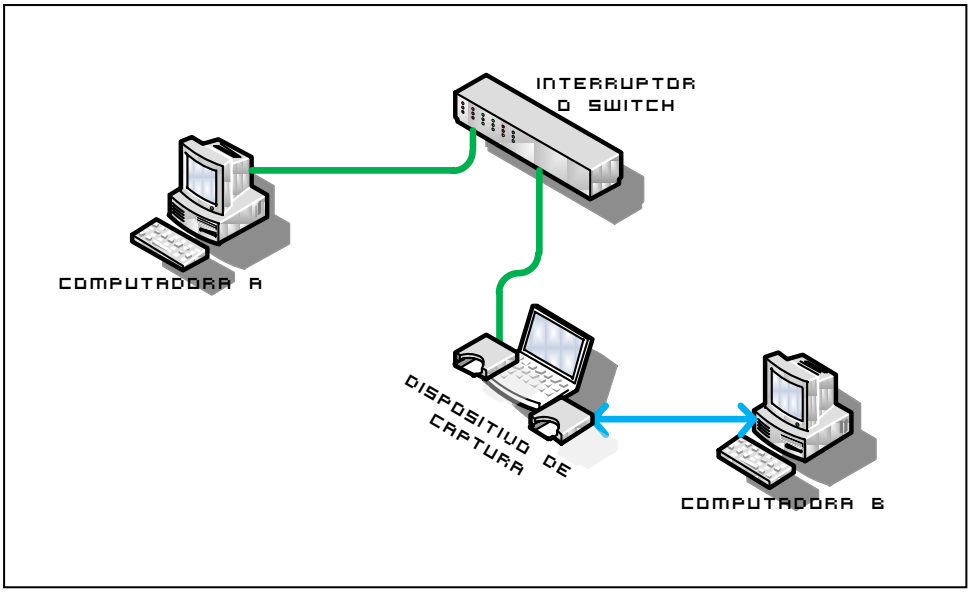


Figura.2.15 - Captura maquina en el medio.

En la figura 2.15 se observan tres computadoras y solo dos están conectadas al switch (la computadora “A” y el dispositivo de captura), el dispositivo de captura recibe los datos transmitidos hacia la computadora “B”, captura la información y la retransmite generando una interferencia y un retraso en el envió de información de la computadora “A” a la computadora “B” y la captura solo se limita a un extremo de la red.

2.3.2.f Captura utilizando una llave de red (Switch + Tap).

Para usar las llaves de red se tiene que capturar ambos rendimientos. El dispositivo de captura debe de contar con dos interfaces de red y también se tiene que configurar la herramienta de captura para que combine estas dos capturas (fig.2.16).

En la mayoría de los sistemas Unix, incluyendo Red Hat, puede unirse dos puertos de Ethernet, y esto previene el tener que utilizar dos interfaces para unir los paquetes de datos, la ventaja de utilizar este método es la captura de tráfico de red en todo un segmento de red a la misma velocidad que el switch, independientemente de la velocidad del adaptador de red del equipo utilizado para la captura, pero tiene la desventaja del costo por que se tiene que adquirir dos tarjetas junto con la llave de red, adicionándole la incomodidad de estar uniendo los paquetes en el dispositivo de captura para evitar la perdida de paquetes de datos.

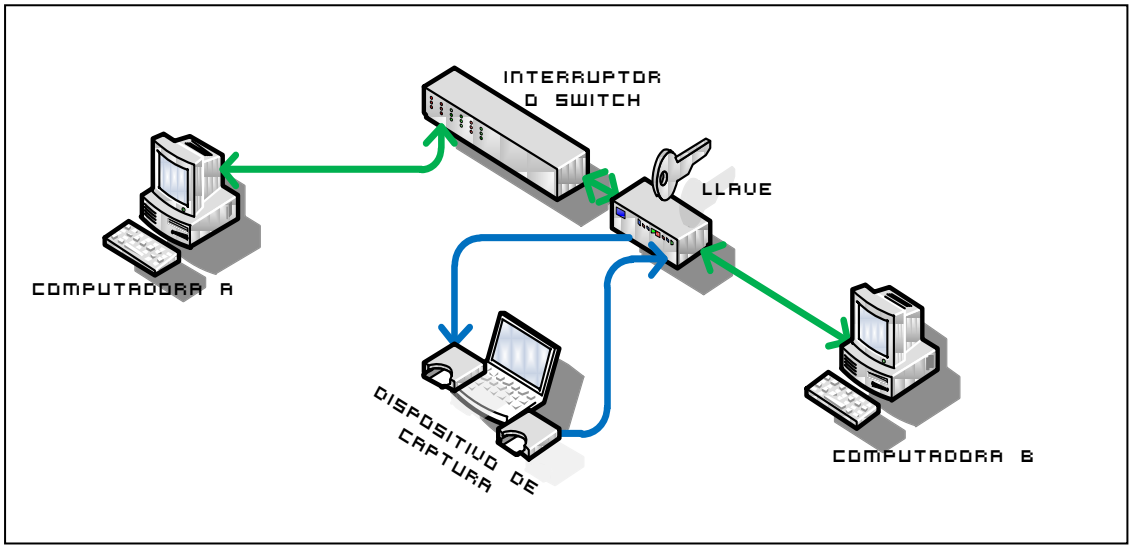


Figura.2.16 - Captura utilizando una llave de red.

2.3.2.g Captura con hombre en el medio o Intermediario (*MITM*[17], *Man-in-the-Middle*)

Este método de captura de paquetes MITM consiste en engañar al switch, se hace pensar al switch que la información que esta enviando a una dirección MAC es la MAC a la que se esta dirigiendo esa máquina. Esto hace que el switch dirija el tráfico a la computadora con la dirección MAC con el dispositivo de captura, donde se reciben y después se envía la trama de datos como si nada hubiera pasado (fig.2.17). Este tipo de ataques puede causar estragos en el switch de la LAN (Red de Área Local, *Local Area Network*). La ventaja de utilizar este método es que es barato y la velocidad de captura es la mima que emplea el switch, pero la desventaja es que existe la perdida de paquetes por que se aumenta el tráfico en la red por la retransmisión de los paquetes de datos a la computadora destino.

[17] Un ataque de MITM es cuando el enemigo adquiere la capacidad de leer, insertar y modificar los paquetes entre dos computadoras sin que estas se den cuenta. El atacante es capas de interceptar información entre las dos computadoras.

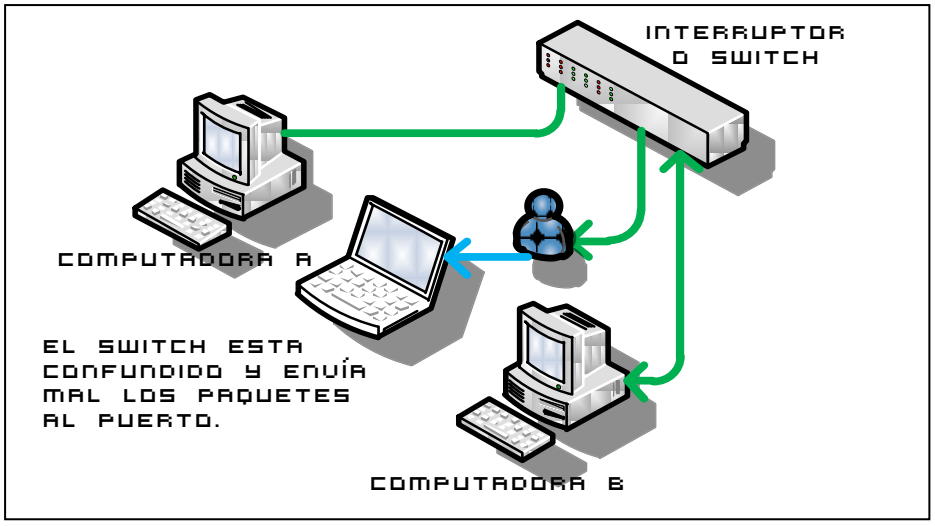


Figura.2.17 - Conexión para realizar la captura en hombre en el medio.

En la figura 2.17 se observa la conexión de hombre en el medio, donde el switch envía la información de la red a la computadora con le herramienta de captura de datos, que después reenvía la información a las computadoras destino, provocando que se incremente el tráfico en la red.

2.3.2.h Ventajas y desventajas del husmeador de paquetes de red.

Ventajas:

- Obtención de la información que circula en la red
- Obtención de las interfaces de red activas en la computadora de captura
- Obtención del tipo de protocolo, dirección IP de origen y destino de los datos
- Control del flujo de datos que circulan por la red
- Descripción de los puertos de comunicación utilizados
- Las capturas se pueden imprimir en archivos de texto para posterior almacenamiento

Desventajas:

- El entorno de la aplicación es poco gráfico.
- La presentación de la información es muy rápida y se tiene que estar supervisando constantemente.
- No hace ningún tipo de alarma al presentarse alguna anomalía en la red
- No cuenta con una base de datos para guardar el historial(log)
- Se tiene que comprar controladores especiales para relazar la captura de datos para redes inalámbricas

2.3.3 Sistemas de detección de intrusiones.

Los Sistemas de Detección de Intrusiones (SDI o IDS[29] por sus siglas en ingles, *Intrusion Detection System*), son un sistemas de monitoreo de tráfico de red que contiene una base de datos y un husmeador de paquetes incluido en su arquitectura, a demás no interfiere con el rendimiento de la red y sirve para detectar accesos no autorizados (Intrusiones[18]), estos a su vez se dividen en dos tipos, los que son instalados en un servidor o una computadora (HostIDS[29]) o en una red (NetworkIDS[29]), en ambos casos se tiene una base de datos en donde se encuentra almacenada la información del comportamiento de una computadora o de una red según sea el caso, y la detección se realiza mediante comparaciones numéricas entre lo almacenado en la base de datos contra lo obtenido de la red, si estos parámetros no coinciden se manda una alerta.

Como se menciono anteriormente existen dos tipos de IDS, los HostIDS (HIDS) que protegen a una computadora, supervisando los eventos locales y analizando información del sistema guardándola en un fichero o registro oficial de eventos durante un rango de tiempo llamado historial (log) y los NetwortIDS (NIDS) que capturan y analizan los paquetes que son transmitidos en una red, almacenando la información en algún dispositivo conectado en la red, al conjunto de NIDS séles conoce como DIDS (IDS Distribuido o *Distributed IDS*) que se basan en una arquitectura cliente servidor realizando un sensado centralizando la información, a su vez estos se encuentran divididos en dos grupos que son pasivos y activos.

Los IDS pasivos [29] detectan una posible violación de seguridad dentro de la red, la registra y reaccionan produciendo un evento informativo de alerta, pero sin tomar ninguna acción y los IDS activos [29] producen una acción sobre la fuente del ataque para así neutralizarla, como puede ser interactuar con el muro de fuego para cerrar la conexión de un posible ataque.

Algunas aplicaciones de IDS pasivos se describen en los siguientes puntos:

[18] Es cualquier persona que viola la seguridad de un sistema informático, esto lo realiza con fines de beneficio personal o para hacer daño.

- *GFI LANguard[31]*.- Es una aplicación que escanea la red en busca de puertos abiertos, accesos no autorizados y debilidades con una administración minima.
- *Tripwire[32]*.- Es una solución que tiene como prioridad la seguridad, generado un reporte de los eventos diarios de la infraestructura de la red.

Algunas aplicaciones de IDS activos se describen en los siguientes puntos:

- *Bro[33]*.- Es un NIDS pasivo que monitorea el tráfico de red y observa actividades sospechosas y detecta intrusiones mediante análisis de tráfico en la red que conllevan a una comparación de eventos de diversas actividades. El análisis incluye detección de ataques y actividades inusuales (como fallas de conexión).
- *Firestorm[34]*.- Es un NIDS de alto desempeño que incluye un sensor y una plataforma de análisis en tiempo real, además de crear reportes, tener acceso remoto mediante consola.

2.3.3.a Tipos de alarmas

Los IDS pasivos detectan un posible ataque y lo registran en la base de dato como un evento en los registros de auditoría, después envía la alerta mediante correo electrónico y también permite enviar alarmas por medio de mensajes de texto SMS (por sus siglas en ingles, *Short Message Service*, Servicio de mensajes cortos) a teléfonos móviles.

En los IDS activos se tienen que configurar las respuestas automáticas a los posibles ataques, como por ejemplo bloquear un puerto por donde éste surgiendo algún ataque, en éste tipos de configuración es preciso seguir muy de cerca el funcionamiento del IDS ya que se pueden provocar falsos positivos[19] los que disminuyen el rendimiento de la red por alguna acción equivocada del IDS.

[19] Errores que provocan los IDS's activos por alguna acción preventiva tomada que provoca errores en algunas partes de la red.

Es recomendable que se éste siempre al tanto del sistema por que envía alarmas utilizando distintos métodos de comunicación. Éste tipo de medidas se deben de tomar para detectar alarmas en casos críticos donde la red puede dejar de funcionar.

Un IDS se puede colocar en distintos puntos de una red para que muestre información concreta de lo que esta pasando en una sección de ella, como se muestra en la figura 2.18.

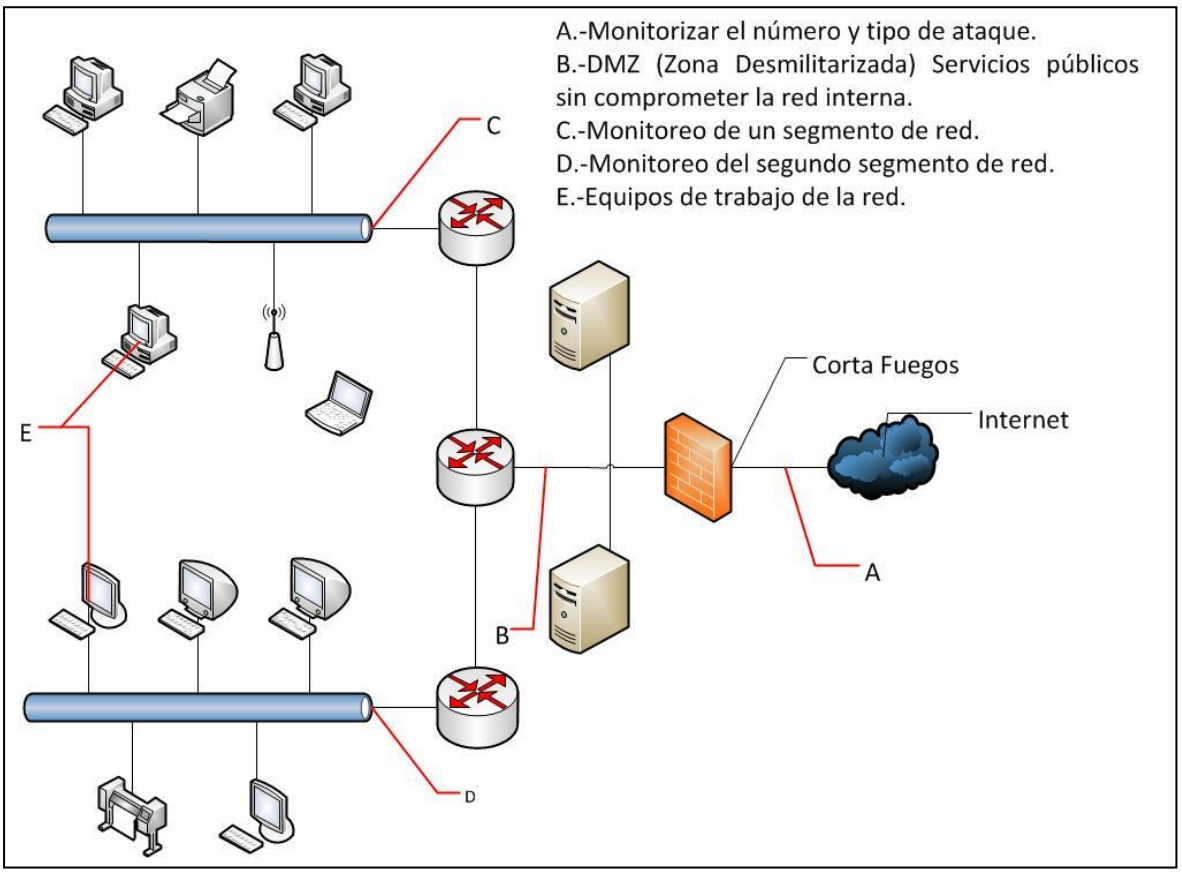


Figura.2.18 - Puntos de instalación de un IDS.

En la figura 2.18 se observan los distintos puntos en donde se puede conecta el IDS y así obtener información del comportamiento de red, a continuación se describe la información que se obtiene en cada uno de los puntos:

A) Colocar un IDS antes del corta fuegos (Fireware) sive para detectar ataques cuyo objetivo es el cortafuegos principal, vigila el número y tipo de ataque dirigidos contra la red, pero algunos ataques que utilicen en sus comunicaciones algún algoritmo de encriptación no puede ser detectado.

B) El tener un IDS en la Zona Desmilitarizada (DMZ[20]) que es la zona entre Internet y la red interna permite detecta ataque contra los servidores que ofrecen servicios públicos a los usuarios, también se detectan consecuencias como conexiones salientes.

C) El tener un IDS en una zona con mayor actividad facilita la detección de ataques dentro de la red, como los realizados por personal interno.

D) Igual que el inciso C.

E) Al colocar un IDS en las propias computadoras proporciona la ventaja de evitar la encriptación en las comunicaciones, pero la detección es limitada a sólo un equipo, con esto se quiere decir que si el equipo se encuentra conectado a un switch el encaminamiento de los paquetes de datos es solo a un puerto por lo cual la información obtenida es limitada.

2.3.3.b Ventajas y desventajas de los sistemas de detección de intrusiones (IDS).

Ventajas:

- Detección de accesos no autorizados
- Proporciona diferentes tipos de seguridad, dependiendo de su ubicación.
- Cuenta con una base de datos para almacenar el comportamiento de la red.
- Proporciona alertas por diferentes medios como correo electrónico, mensajes SMS a teléfonos móviles, etc.
- Dependiendo del tipo de IDS puede o no tomar acción sobre algún tipo de ataque.

Desventajas:

- No detecta accesos que utilicen algún tipo de algoritmo de encriptación.
- Si el IDS es activo se tiene que estar constantemente al tanto de las configuraciones.
- Debe de tener una constante supervisión por la aparición de los falsos positivos.
- Los falsos positivos generan más errores en la red.
- Se necesita tener un conocimiento profundo de redes junto con una capacitación contante del IDS que se utilice.

[20] Proporciona servicios públicos sin tener que permitir accesos a la red privada de la organización.

2.4 Comparación entre las herramientas de administración y el presente desarrollo.

En las secciones anteriores se han descrito las herramientas que normalmente utiliza y ayudan al administrador de red a mantener en condiciones de operación una red, se puede ver que las características comunes que ayudan al administrador de red en las herramientas anteriormente mencionadas son los siguientes:

- Notificaciones visuales de alertas presentando de forma rápida, en cuadros de diálogo los errores que se generan en la red.
- Gráficas del comportamiento de red, igual que el punto anterior, con la diferencia que estas gráficas muestran información de los segmentos de red y el estado de los equipos instalados en la red.
- Almacenamiento del comportamiento normal y anormal de la red, con la finalidad de corroborar mediante comparaciones numéricas en la base de datos si la red no tiene algún problema.
- Generación de reportes, la finalidad es respaldar actualizaciones o procedimientos que se tenga que hacer sobre la red, también puede ser empleado para auditorías.
- Generación de alertas (mail, SMS), por la practicidad de estar siempre informado del comportamiento de red.

Estas características se unifican en una sola aplicación ListeningWire (El presente desarrollo) que además agrega otro beneficio como es la medición del consumo del ancho de banda, lo que genera un plus al momento de elegir entre un *dashboard*, *snort*[21] y un IDS. En la siguiente Tabla (Tabla.2.1) se muestra una comparación entre sus parámetros.

[21] (Snort.- Resoplido)Es un sniffer junto con un detector de intruso[30]

Parámetros a cubrir.	***Snort***	***IDS***	***ListeningWire.***
Verificación del consumo de ancho de banda.			X
Generación de alertas (mail y SMS).	X	X	
Generación de reportes.		X	X
Generación de un formato de reportes.			X
Envío de trama SNMP		X	
Detección de intrusos.	X	X	
Almacenamiento del comportamiento normal y anormal de la red.	X	X	X
Reconfiguración de dispositivos externos.		X	
Notificaciones visuales de las alertas.		X	X
Graficas del comportamiento de red.			X
Responder ante anomalías de la red.	X	X	
Software gratuito.	X		X
Plataforma Unix y Linux	X	X	X
Plataforma Windows	X	X	X

Nota: La X representa una característica importante

Tabla.2.1 Resumen de la comparación entre los parámetros de herramientas de administración de red

La Tabla.2.1 representa el resumen de características que son importantes para la administración de red y se puede notar que existen algunas similitudes en las características de un IDS y de ListeningWire por lo que es conveniente hacer una comparación mas detalla de los diferentes tipos de IDS con ListeningWire, el presente desarrollo aporta un mayor beneficio a la administración como lo muestra la tabla. 2.2.

Características	**HostIDS**		**NetwortIDS**		**Proyecto**
Herramientas IDS.	*GFI LANguard*	*Tripwire*	*Bro*	*Firestorm*	*Listening Wire.*
Detección de Accesos no deseados	No	-	-	-	No
Detección del Consumo de Ancho de Banda	No	No	No	No	Sí
Ejecución en tiempo real	No	Sí	No	Sí	Sí
Detección de situaciones anómalas En la red.	Sí	-	Sí	No	Sí
Estadísticas de situaciones anómalas	No	-	Sí	No	No
Generación de alertas	Sí	Sí	Sí	Sí, por logs	Sí
Grafica de alerta de las anomalías	Sí	Sí	-	-	No
Generación de reportes para auditoria.	Sí	Sí	-	-	Sí
Almacenamiento del comportamiento de red en una base de datos.	No	Sí	Sí	No	Sí
IDS pasivo	No	Sí	Sí	-	Sí
IDS activo	Sí	No	-	Sí	No
Agente en host	Sí	Sí	No	No	No
Facilidad de instalación	No	No	No	Sí	No
Facilidad de administración	Sí	No	No	-	Sí
Número de falsos-positivos	No	-	-	-	No
Nivel de saturación de la red	No	No	No	No	Sí
Tamaño de la red	-	-	-	-	Amplia
Detección de Maquinas virtuales	Sí	Sí	No	-	No
Conexión de escritorio remoto	Sí	Sí	No	-	No
TCP	Sí	Sí	Sí	Sí	Sí
UDP	Sí	Sí	Sí	Sí	Sí
ARP	No	No	Sí	Sí	No
IPv4	No	No	No	Sí	Sí
ICMP	Sí	No	No	Sí	Sí
NDS	-	-	-	-	Sí
SMTP	Sí	No	No	-	No
IGMP	No	No	No	Sí	No
802.1 (VLAN)	No	-	-	Sí	No
IrDA	No	-	-	Sí	No
IPX	No	No	No	Sí	No
Sistema operativo Unix	No	Sí	Sí	Sí	Sí
Sistema operativo Linux	No	Sí	No	Sí	Sí
Sistema operativo Windows	Sí	Sí	No	Sí	Sí
Requiere personal especializado	-	-	Sí	-	No

Sí.- Es una ventaja y denota que si cumple la condición

No.- Es una desventaja y denota que no cumple la condición

(-).- Parámetro no especificado

Tabla 2.2Tabla comparativa entre HIDS, NIDS, pasivos y activos contra ListeningWire.

En la Tabla.2.2 se muestra con detalle la comparativa entre características en los tipos de IDS's contra ListeningWire que no es un IDS pero que contiene elementos

indispensables como la utilización de los protocolos estándar TCP/IP que es el conjunto de protocolos de comunicación, direccionamiento, transferencia de correo entre otros (las definiciones de los protocolos las puede encontrar en el glosario) también se muestra otras características como el almacenamiento del comportamiento de red, la generación de reportes, graficas visuales de alerta, y se agrega otra característica que es la detección del consumo de ancho de banda con la cual no cuentan los IDS's y que esta remarca en la Tabla.2.2.

2.5 Resumen.

El aumento del número de equipos y usuarios en un LAN dificulta la administración por la pérdida de paquetes de datos por el incremento en el consumo del ancho de banda del medio de transmisión, por esta razón el especialista en tecnologías de la información emplea herramientas como los tableros de control que mediante una interfaz gráfica muestra la información del comportamiento de red, un husmeador de paquetes que es un programa que captura los paquetes de datos que no interferir con el proceso de envío y recepción de información y los sistemas de detección de intrusiones que son sistemas de monitoreo de tráfico que detectar accesos no autorizados, estas son distintas aplicaciones que el presente desarrollo pretende unificar utilizando un husmeador de paquetes, la presentación de alarmas y almacenamiento del comportamiento de red de los IDS's para finalmente presentar la información utilizando la sofisticada tecnología de los tableros de control para mantener los niveles de operación en condiciones óptimas evitando la degradación de la red.

Capítulo 3

Desarrollo de los módulos del tablero de control

En éste se capítulo se presenta el diseño y construcción de los módulos que integran el sistema del tablero de control. En la sección 3.1 se presenta la arquitectura general del sistema, la sección 3.2 explica el desarrollo del sistema y la sección 3.3 presenta un resumen del capítulo.

3.1 Arquitectura del general del sistema.

En la figura 3.1 se muestra la arquitectura del sistema que esta compuesta por tres niveles, el primer nivel presenta el tablero de control que se compone de 3 módulos principales ubicados en el segundo nivel que presentan la arquitectura de: análisis de tráfico, base de datos e interfaz gráfica. Cada módulo esta compuesto de varios submodulos que están en el tercer nivel de la jerarquía que se describen con mayor detalle en la sección de diseño (sección 3.2), a continuación se describe la función general cada uno de los módulos:

1. Análisis de de tráfico.- Este módulo se encarga de adquirir, procesar y analizar los datos que provienen de la red, el módulo contiene los siguientes submodulos: captura de tráfico de red, detección de paquetes de entrada salida, comprobación de conexión y detección de interfaces de red.

2. Base de datos.- Este módulo se encarga de almacenar los datos que provienen del módulo de adquisición de datos de red y de los datos que ingresa el usuario, el módulo contiene los siguientes submodulos: diagramas entidad relación, expresiones del algebra relacional y consultas en SQL (por sus siglas en ingles *International Query Language*, lenguaje estructurado de consulta [3]).

3. Interfaz gráfica.- Despliega la información contenida en la base de datos de forma gráfica, el módulo contiene los siguientes submodulos: medidor de tráfico de red, medidor de ancho de banda, alarmas, Agregar, eliminar, modificar y buscar nodo.

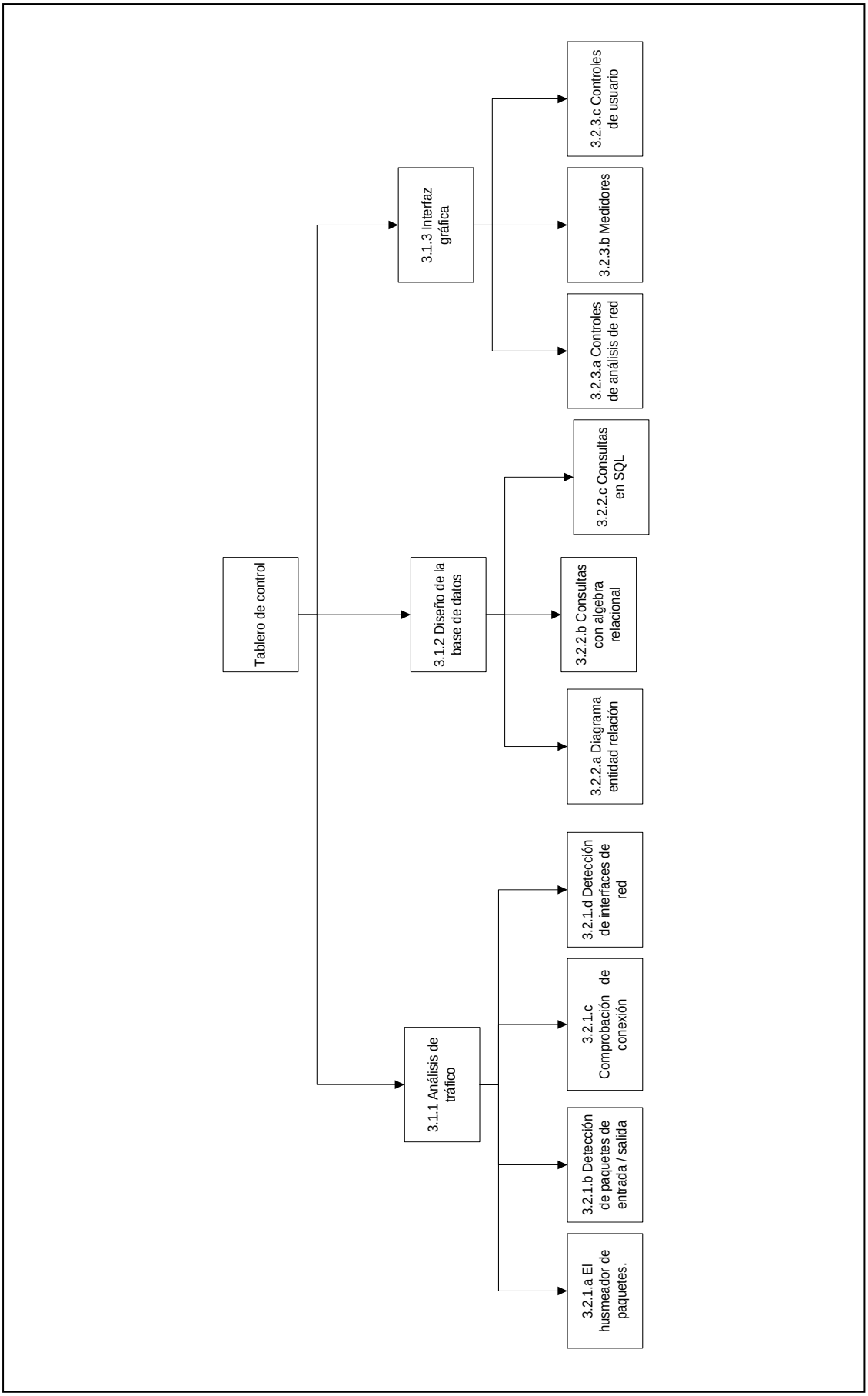


Figura.3.1 – Arquitectura del tablero de control (El número indica la sección del texto)

3.1.1 Análisis de tráfico.

Este módulo obtiene, procesar y analizar la información que cruza por el medio de transmisión. Su primer componente contiene un husmeador de paquetes con la arquitectura que se muestra en la figura.3.2.

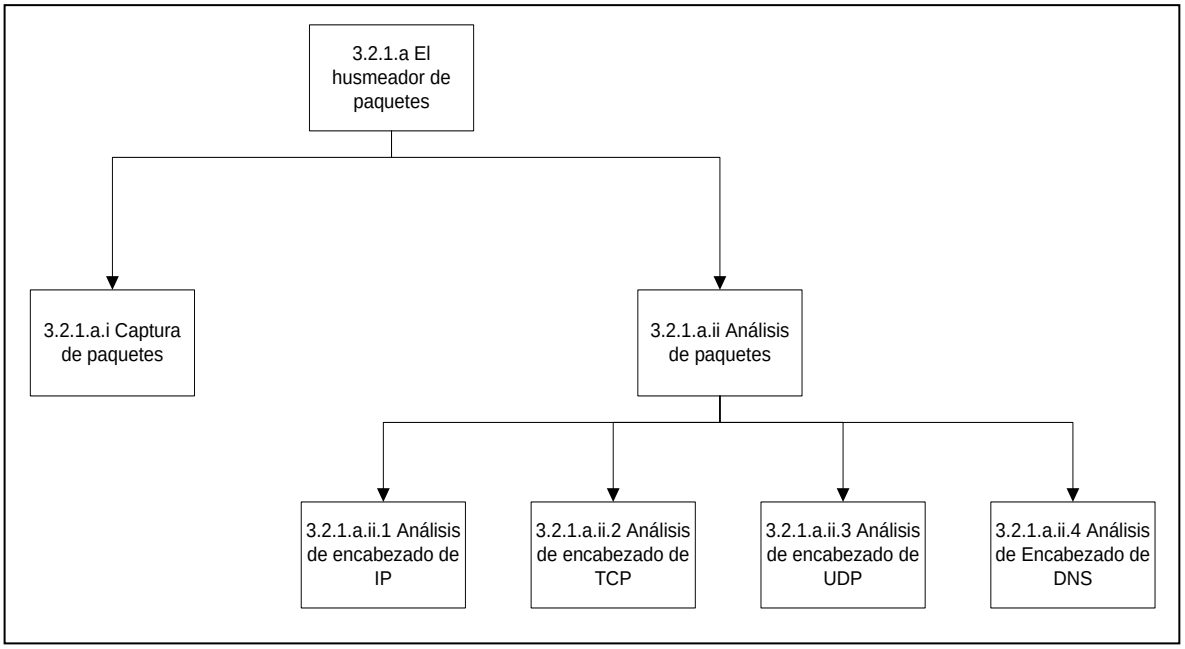


Figura.3.2 Arquitectura de la captura de tráfico de red.

Como se observa en la figura 3.2 el segundo nivel de la jerarquía esta compuesta por los siguientes componentes:

1) Captura de paquetes: la captura se realiza por medio de la apertura de un socket[22] y la activación de recepción de datos de la red.
2) Análisis de paquetes: este submodulo procesa y analiza los paquetes se compone de 4 funciones principales para obtener información de los encabezados que están contenidos dentro del paquete de datos descritos en los siguientes puntos:

    a) **Encabezado de IP**: obtiene la versión de IP, tipo de servicio, longitud total del encabezado, la identificación, las banderas, tiempo de vida del

[22] Un *sockets* es un objeto que representa un acceso de bajo nivel en la pila de IP para la entrega de paquetes de datos provenientes de la tarjeta de red a los procesos apropiados. Un *socket* queda definido por un par de direcciones IP local y remota, un protocolo de transporte y un par de números de puerto local y remoto [2].

paquete de datos, tipo de protocolo, la suma de comprobación, IP origen e IP destino.

b) **Encabezado de TCP**: obtiene el puerto de origen, el puerto destino, número de reconocimiento, banderas, datos de compensación, espacio de ventana, suma de comprobación y el punto de urgencia.

c) **Encabezado de UDP**: obtiene el puerto de origen, el puerto destino, la longitud de la trama de datos y la suma de comprobación.

d) **Encabezado de DNS**: obtiene el encabezado, el sistema de nombres de dominio, si alguno de los protocolos TCP ó UDP utilizan el puerto 53[23] y los registros de recursos encadenados (RRs por sus siglas en ingles *Concatenated Resource Records*).

El análisis del tráfico de red se explica en detalle en la sección 3.2.1

[23]Para mayor información revisar en línea: www.iana.org/assignments/port-numbers.txt

3.1.2 Base de datos

La base de datos del sistema almacena la información de tráfico obtenida por el husmeador de paquetes (*sniffer*[24]), junto con la cantidad de paquetes de entrada, salida, errores y los datos que el usuario ingresa. Las operaciones que se realizan sobre la base de datos del sistema se ejecutan en tres niveles de abstracción como lo muestra la figura 3.3.

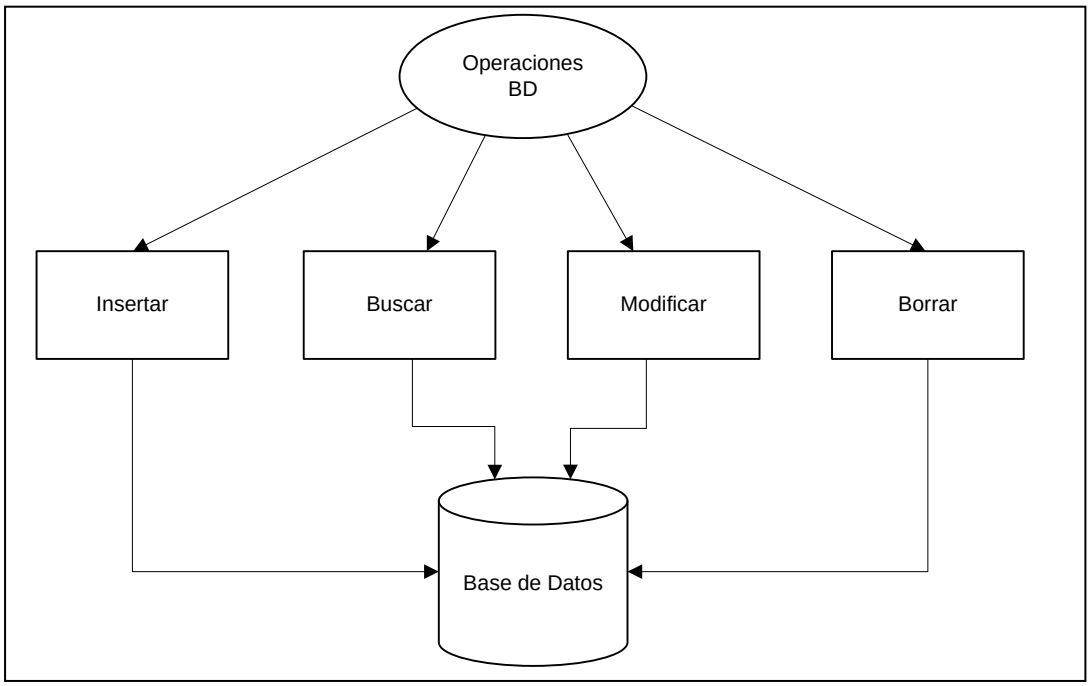


Figura 3.3 - Niveles de abstracción dentro de la base de datos.

En la figura 3.3 se observan los tres niveles de abstracción para simplificar la interacción del usuario con el sistema. El primer nivel de la jerarquía es la vista que tiene el usuario desde el entorno gráfico donde al seleccionar alguna de las opciones como insertar que agrega un nuevo registro sobre una tabla, buscar que realiza un búsqueda de un registro determinado dentro de una tabla, modificar que actualiza un registro y borrar que elimina un registro de una tabla dentro de la base de datos, se activa el segundo nivel de la jerarquía (nivel lógico) que describe que datos se almacenan en la base de datos para después pasar al tercer nivel de la jerarquía donde se describe como es almacenada la información en las estructura de datos. Estos tres niveles permiten al usuario manipular la información del sistema del tablero de control. El diseño de la base de datos se explica en detalle en la sección 3.2.2.

[24] Remitirse a husmeador de paquetes del marco teórico.

3.1.3 Interfaz gráfica

La interfaz gráfica presenta la información e interactúa con el usuario para facilitar la administración de red mediante la interacción del usuario con la computadora, en la figura 3.4 se presenta la arquitectura del entorno gráfico del tablero de control y en la sección 3.2.3 se describe el diseño de cada módulos de la interfaz gráfica.

La interfaz grafica esta compuesta por los siguientes componentes:

- Barra de menú: Permite al usuario la selección de dos opciones el menú archivo que contiene la función "Salir" y el menú herramientas que contiene las funciones "Interfaces de red, Hacer PING y Estadísticas".
- Pestaña monitoreo: Presenta el comportamiento de red mediante los siguientes componentes:
  - Análisis de tráfico de red: Le presenta al usuario información del análisis del comportamiento de red mediante el husmeador de paquetes.
  - Medidor del consumo de ancho de banda: Le indica al usuario el porcentaje del consumo del ancho de banda del canal en ese momento, así como la velocidad de transmisión de los paquetes de datos que viajan en la red.
  - Medidor de cantidad de equipos activos en la red: Le indica al usuario la cantidad de equipos conectados en la red en ese momento y la medición porcentual que se genera.
  - Medidores de paquetes: le permiten al usuario visualizar la cantidad de paquetes de entrada, salida y con errores que se obtienen dentro de la red.
  - Contador de paquetes: Le indica al usuario la cantidad de paquetes que entran de un protocolo como: TCP, UDP, Desconocido.
  - Consulta de equipos: Le permite al usuario consultar las características y el estado de los equipos instalados en un departamento determinado.
  - Botón de inicio / alto: Le permite al usuario iniciar o detener la ejecución del programa.
  - Selección de interfaz de red: Le permite al usuario seleccionar el nombre o dirección de un adaptado de red instalado en el equipo de cómputo.

- Pestaña reporte: Le presenta al usuario una tabla con las direcciones IP, tipo de protocolo (TCP, UDP, Desconocido) y el número de veces que se repite un protocolo, también le permite al usuario generar el reporte para imprimirlo o almacenarlo en un directorio dentro del disco de la computadora.
- Pestaña administración: Le permite al usuario la edición de información de los nodos por departamentos mediante los siguientes componentes:
  - Insertar nodo: Le permite al usuario ingresar datos a un nuevo registro de un equipo para darlo de alta dentro de la base de datos.
  - Modificar nodo: Le permite al usuario realizar modificaciones a un determinado registro dentro de la base de datos.
  - Borrar nodo: Le permite al usuario borrar un registro determinado dentro de la base de datos.
  - Buscar nodo: Le permite al usuario realizar una búsqueda dentro de la base de datos.
  - Configurar parámetros de red: Le permite al usuario configurar el segmento y el número de equipos con los que cuenta su red
  - Configurar alarmas: Le permite al usuario configurar los eventos para que el sistema genere una alarma.

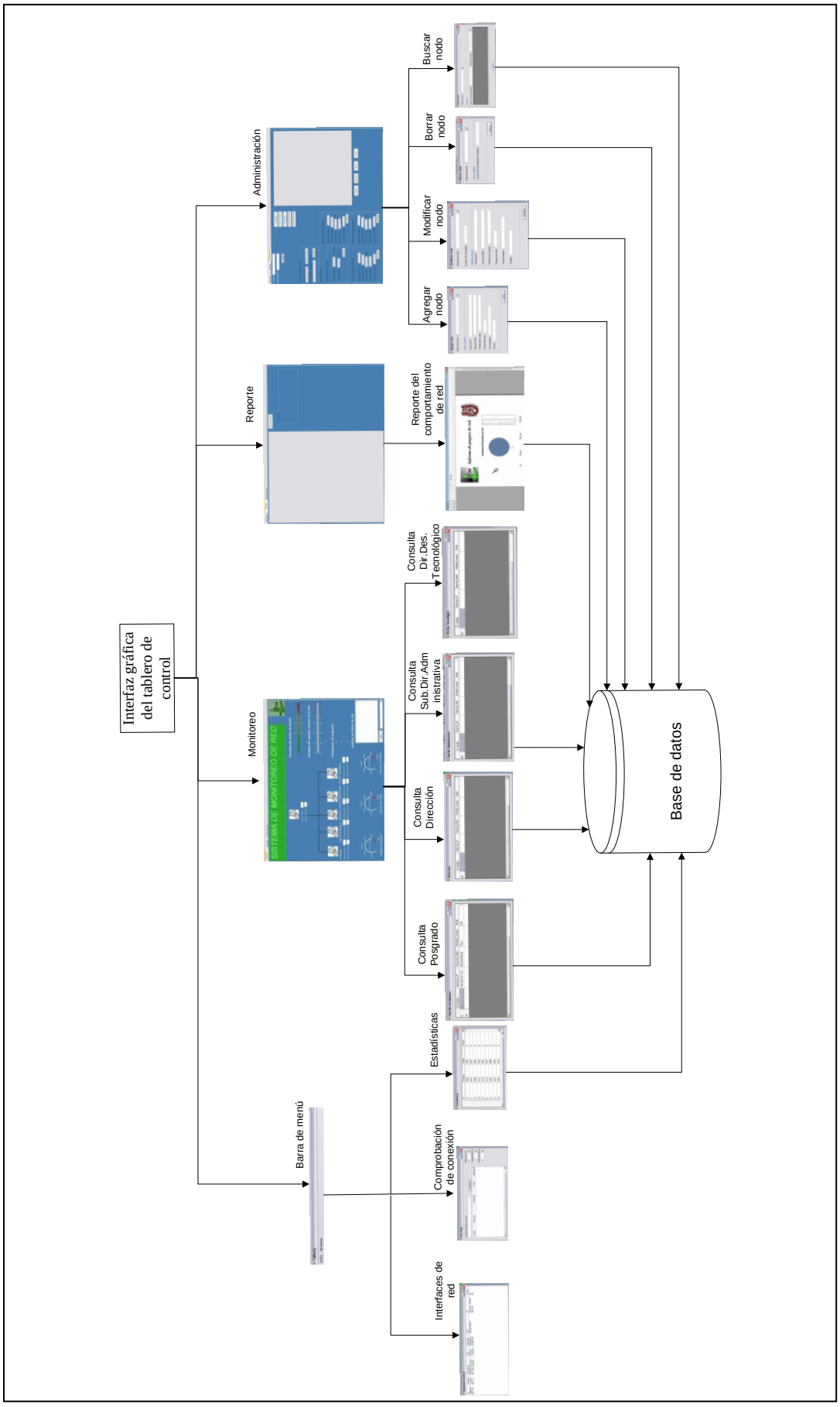


Figura 3.4 - Arquitectura de la interfaz gráfica del tablero de control

La tabla 3.1 contiene la descripción de la función de la interfaz gráfica

Función:	Interfaz gráfica
Entradas:	Selección de interfaz de red (ComboBox), Botón de inicio / alto (Botton1)
Salidas:	Información del tráfico de red, Información del consumo del ancho de banda del canal de transmisión, contador de números de paquetes de red, información del husmeador de paquetes, información de la base de datos.

Descripción:

La interfaz grafica es la encargada de presentar la información al usuario de forma fácil e intuitiva, para facilitar la administración de red, los siguientes puntos presentan los pasos para iniciar la aplicación.

1) Seleccionar una interfaz de red instala en la computadora.
2) Activar el botón inicio para comenzar la ejecución del programa, el botón inicio cambia su estado a alto.
3) El husmeador de paquetes comienza a realizar el análisis de datos, presentando la IP de origen y destino separas por un guion.
4) Se realiza la conexión a la base de datos.
5) Se almacena el comportamiento de red en la base de datos.
6) Se despliega la velocidad de transmisión con la que los datos viajan en la red.
7) Se despliega el porcentaje del consumo de ancho de banda del canal.
8) Se despliega el número de equipos activos en la red.
9) Se despliega el porcentaje de equipos activos en la red.
10) Se despliega la cantidad de protocolos TCP, UDP, Desconocidos que circulan por la red.
11) Se despliega la cantidad de paquetes de entrada, salida, y con errores.

Tabla.3.1 – Función Interfaz gráfica.

3.2 Diseño del sistema

En esta sección se explica en detalle el diseño de cada componente de la arquitectura del sistema, la sección 3.2.1 presenta el análisis de paquetes, la sección 3.2.2 describe la base de datos y la sección 3.2.3 explica la interfaz gráfica.

3.2.1 Análisis de tráfico

Para el análisis de tráfico la sección 3.2.1.a explica el husmeador de paquetes, la sección 3.2.1.b presenta la detección de paquetes de entrada / salida, la sección 3.2.1.c explica la comprobación de conexión (PING) y la sección 3.2.1.d presenta la detección de los adaptadores de red instalados en el equipo.

3.2.1.a El husmeador de paquetes.

La detección de paquetes es posible por medio del husmeador o *sniffer* que coloca a la tarjeta de red[25] en una configuración de modo promiscuo[26], así el adaptador de red comienza a escuchar la información que es transmitida por la red sin alterarla y sin generar retrasos entre el transmisor y receptor. La función de este modulo es capturar los paquetes de datos del medio de transmisión, analizar del encabezado de IP y los protocolos que están encapsulados en él como TCP, UDP, etc.

Para la construcción de este módulo se utilizando un *buffer* en donde se contienen los paquetes o tramas[27] de datos obtenidos de la red para después llamar al método `BeginReceive()` para continuar con la captura de los paquetes, en caso de haber algún problema con el tamaño de un paquete se genera un error de desbordamiento en la operación, de lo contrario se continuara con la recepción de paquetes que serán adicionadas en el buffer para ser vistas en un formulario de Windows por medio de un `TreeView`.

[25] Para mayor información acerca del adaptador de red remitirse al apéndice "C".
[26] Remitirse a husmeador de paquetes del marco teórico.
[27] Remitirse al apéndice "A" en el tema capa de enlace de datos.

La función que realiza este trabajo se llama husmeador de paquetes (Tabla.3.2) que junto con el diagrama EPC[28] (por sus siglas en ingles, *Event-driven Process Chains*, Cadenas de procesos condicionados por evento) (fig.3.5) y su algoritmo se encarga de capturar y analizar la información que proviene de la red.

Función:	Husmeador de paquetes de red[29].
Entrada:	Paquetes que circulan por la red.
Salida:	IP_origen, IP_destino, Tipo de protocolos IP, UDP, TCP, DNS y Desconocido.
Variables importantes:	bContinuaCap, Socket, TCP, UDP, DNS, Desconocido.
Funciones:	IPHeader, TCPHeader, DNSHeader

Descripción:

1) Seleccionar la interfaz de captura y activar el husmeador.
2) Verificar el estado de bContinuaCap, si esta activo:
    a) Abre el socket y activa la captura de datos.
    b) Comienza a recibir los paquetes
    c) Analiza el encabezado de IP (IPHeader).
    d) Analiza el protocolo encapsulado en la trama de IP:
        i. En caso de ser TCP, analiza el encabezado de TCP (TCPHeader), si el puerto origen o destino es 53 analiza el encabezado de DNS (DNSHeader), de lo contrario solo pasa la información de TCPHeader.
        ii. En caso de ser UDP, analiza el encabezado de UDP (UDPHeader), si el puerto origen o destino es 53 analiza el encabezado de DNS (DNSHeader), de lo contrario solo pasa la información de UDPHeader.
        iii. En caso de ser un protocolo desconocido, solo coloca Desconocido
3) De lo contrario bContinuaCap es falso, cierra el socket y finaliza el programa.

Tabla.3.2 – Función husmeador de paquetes

[28] Son notaciones gráficas semi – oficial que se usa principalmente para representar negocios. EPC se utiliza en las prácticas industriales y la práctica académica [http://office.microsoft.com/es-ar/visio-help/crear-un-diagrama-epc-cadena-de-procesos-condicionados-por-eventos-HP001057504.aspx].

[29] El codigo fuente lo puede localizar en el apendice “D”

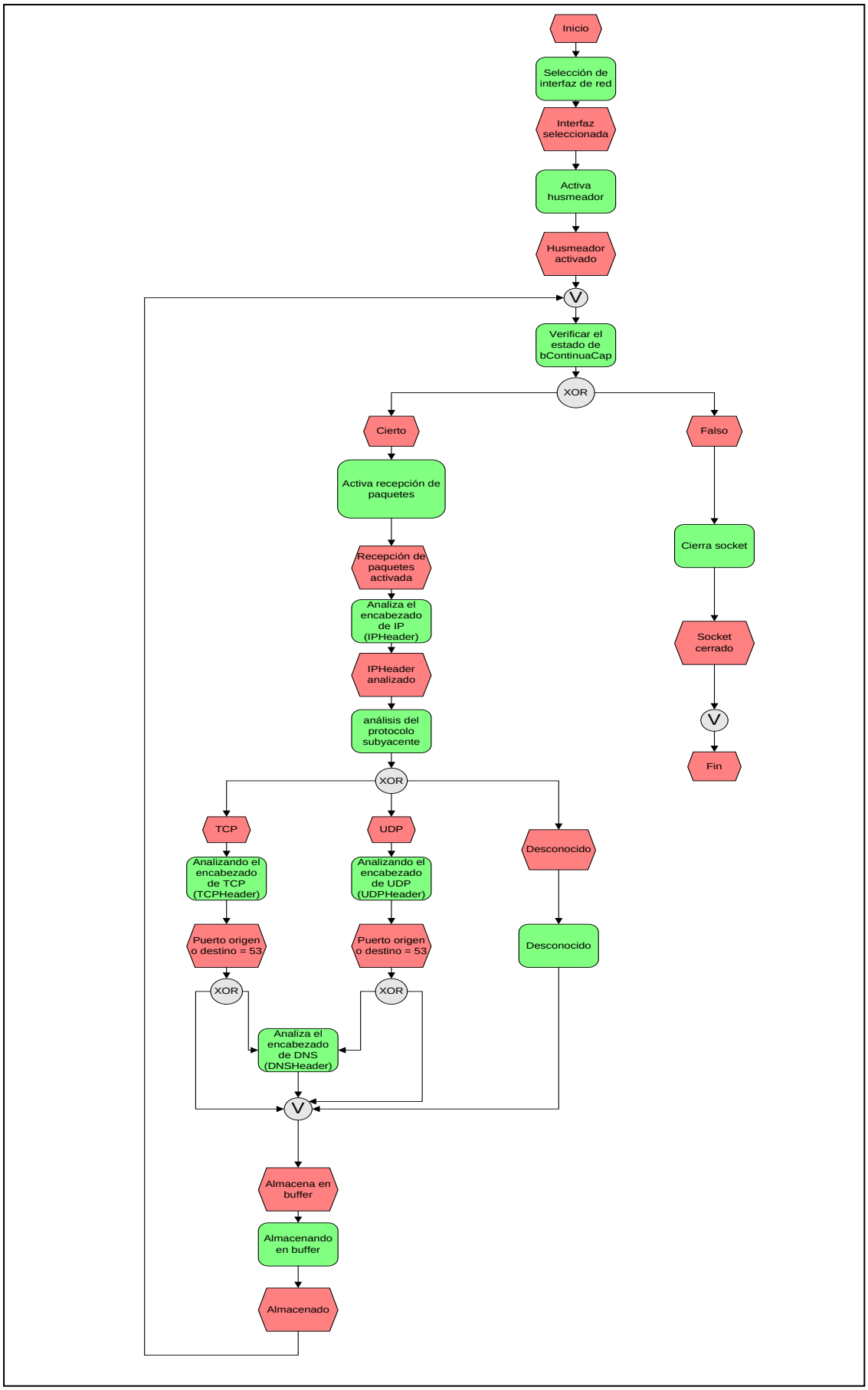


Figura.3.5. – Diagrama EPC del olfateo de paquetes

Nombre:	**Algoritmo de olfateo de paquetes (sniffer)**
Descripción:	Abre un socket para colocar el adaptador de red en modo promiscuo y permitir la captura de los datos.
Variables importantes:	bContinuaCap, Socket, TCP, UDP, DNS, Desconocido Nodoraiz, ipHeader, ipNodo tcpHeader, tcpNodos, udpHeader, udpNodo, dnsHeader, Node.
Funciones:	IPHeader, TCPHeader, DNSHeader

```
[Inicio]
    TCP◄——— 6
    UDP◄——— 17
    Desconocido◄——— -1
    ByteDato[4069]
    bContinuaCap◄——— Falso
    Abre socket (mainSocket)
      Si Socket = Abierto
            bContinuaCap◄——— Cierto
            ActivaRecepción (ar)
      De lo contrario
            Envía mensaje de error
            CierraSocket( )
   ActivaRecepción(ar)
      Si bContinuaCap = Cierto
            AnalizaProtocolo(ByteDato, nRecep)
            Redimenciona ByteDato
            Para i = 1 hasta nRecep
                  AddNodoArbol(Nodoraiz)
   AnalizaProtocolo(ByteDato, nReceived)
      ipHeader◄——— IPHeader(ByteData, nReceived)
      ipNodo ◄——— tcpHeader
      Nodoraiz.Nodo.Add(ipNodo)
      Para cada protocolo subyacente en ipHeader
            En caso de TCP
                  tcpHeader◄ TCPHeader(ipHeader.Dato, longitud)
                  tcpNodo◄——— tcpHeader
                  Nodoraiz.Nodo.Add (tcpNodo)
                  Si puerto origen ó destino  = 53
                        dnsNodo ◄——— TCPNodoArbol(tcpHeader)
            En caso de UDP
                  udpHeader◄——UDPHeader(ipHeader.Dato, longitud)
                  udpNodo◄——— udpHeader
                  Nodoraiz.Nodo.Add(udpNodo)
                  Si puerto origen o destino = 53
                        dnsNodo ◄——— UDPNodoArbol(udpHeader)
            En caso de Desconocido
                  Enviar Desconocido
    AddNodoArbol (Node)
      TreeView.Nodo.Add (Node)
    CirreaSocket()
      Socket◄——— Close()
[FIN]
```

Algoritmo 3.1 – Algoritmo del husmeador de paquetes (*sniffer*).

3.2.1.a.i Captura de paquetes

Para activar la captura de datos primero se tiene que cambiar la configuración del adaptador de red por medio de la apertura de un *socket* de conexión [40], de este modo el adaptador de red recibe y transmite los paquetes de datos que circulan por la red sin alterar la información.

Para la construcción de aplicaciones de red en Visual Studio .NET es importante entender como programar un socket. La plataforma .NET dispone de un esquema lógico de nombres que agrupa clases[30], funciones, tipos de datos, etcétera (espacios de nombres[31]) que se llama `System.Net` (Tabla 3.3) que contiene las clases para distintos protocolos de red, que a su vez contiene el espacio de nombres `System.Net.Socket()`[32] que contiene los servicios de acceso a la red para abrir y cerrar un socket.

Espacio de nombres	Descripción
System.Net	Proporciona una interfaz de programación para muchos de los protocolos que se utilizan actualmente en las redes.
System.Net.Socket	Ofrece una solución administrada de la interfaz Windows Socket (Winsock) para desarrollos que controlen el acceso a la red.

Tabla.3.3 – Descripción de los espacios de nombres System en Visual Basic.NET[33]

La función abrir captura de paquetes (Tabla.3.4) define los pasos para realiza la apertura del socket así como el inicio del proceso de escucha para que la tarjeta de red éste en configuración de modo promiscuo y así se pueda recibir tanto los paquetes que están dirigidos hacia la tarjeta de red, como los que están dirigidos a otras computadoras.

[30] Son los datos y métodos con los que dispone los objetos de ese tipo.
[31] En línea: http://lospasosdelrex.wordpress.com/2010/06/08/%C2%BFque-son-los-namespace-en-vb-net/
[32] Para mayor información visite http://msdn.microsoft.com/es-es/library/system.net.sockets%28v=vs.80%29.aspx
[33] Para mayor información de los espacios de nombres con los que se puede trabajar consultar: http://msdn.microsoft.com/es-es/library/gg145039.aspx

Función:	Captura de paquetes[34]
Entradas:	Familia de direcciones, Tipo de socket, Tipo de protocolo, Nombre o dirección o dirección del adaptador de red de origen.
Salidas:	Longitud del paquete de datos, Comportamiento de envió y recepción de paquetes del socket, Dirección IP, Número de puerto.
Variables Importantes:	mainSocket, ByteDato, bContinuaCap

Descripción:

1) Crear una variable del tipo *socket* (que puede utilizar cualquier puerto que no este dentro de los primeros 1024 números de puerto, ya que estos están reservados [2]) para almacenar los datos capturados.
2) Inicializar una instancia nueva de la clase *socket* con la lista de direcciones (`AddressFamily`[35] que obtiene la familia de direcciones del protocolo IP[36]) que especifica el parámetro de direcciones que utiliza el *socket*.
3) Especificar el tipo de conexión *Raw Socket*, la conexión *Raw* permite el acceso directo al protocolo IP para la generación de encabezados con segmentos TCP ó UDP [2].
4) Indicarle el tipo de protocolo a trabajar en este caso IP ya que la mayor parte del tráfico de red esta encapsulado en este protocolo (también puede ser TCP, UDP, etc).
5) Conecta el *socket* a la dirección IP seleccionada, utilizando el método `IPEndPoint()` que solicita el inicio de una conexión asíncrona con un equipo remoto.
6) Agrupar las opciones del *socket* que se encuentran en función del nivel de compatibilidad con los protocolos `SocketOptionLevel.IP` y las opciones establecidas del encabezado de IP `SocketOptionName.HeaderIncluded`.
7) Crear una pila de paquetes que incluye el conjunto de los encabezados tanto para los paquetes de salida como de entrada.
8) Modificar los parámetros de operación del *socket* usando el método `IOControl()` que proporciona acceso al socket.
9) Activar de la recepción asíncrona de paquetes con `BeginReceive()` que comienza a recibir los datos del *socket*.

Tabla.3.4. – Función abrir socket y activar recepción de datos.

[34] El código fuente lo puede encontrar en el apéndice "D".
[35] En el apéndice "B" puede encontrar los esquemas de familia de direcciones y los tipos de protocolos que puede soporta un socket
[36] Para mayor información visite http://msdn.microsoft.com/es-es/library/system.net.ipendpoint.addressfamily%28v=vs.80%29.aspx

3.2.1.a.ii Análisis de paquetes.

Para analizar el tráfico de red se comienza por el paquete de datos para analizar el encabezado de IP [21](sección 3.2.1.a.ii.1) ya que este encabezado encapsula protocolos como TCP [22] (sección 3.2.1.a.ii.2) o UDP [23](sección 3.2.1.a.ii.3), además de los protocolos que se encuentran por encima de ellos como DNS [24] (sección 3.2.1.a.ii.4), http, FTP, etcétera. En la figura 3.6 se muestra como es encapsulado un encabezado de TCP en una trama de IP.

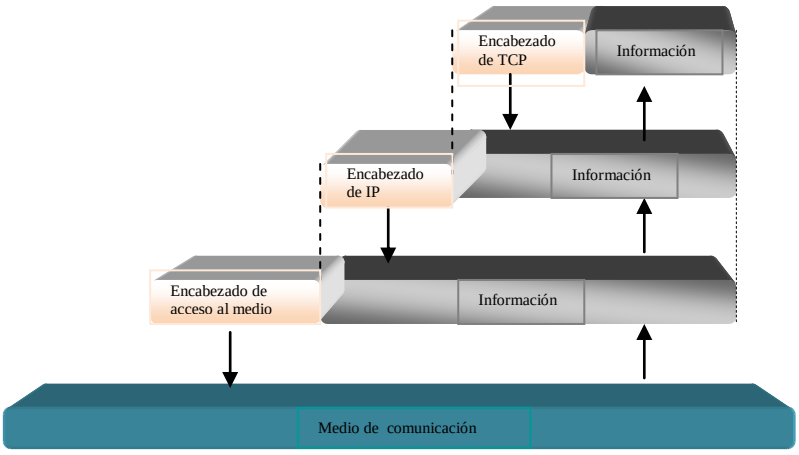


Figura.3.6 – Paquete de TCP encapsulado en una trama de datos de IP

3.2.1.a.ii.1 Análisis de encabezado de IP

El formato del encabezado del protocolo de internet es el siguiente (fig 3.7):

Versión	IHL	Tipo de Servicio	Longitud	
Identificación			Banderas	Fragmento de compensación
Tiempo de vida		Protocolo	Suma de comprobación del encabezado	
Dirección origen				
Dirección destino				
Opciones				Relleno

Figura.3.7 – Formato de encabezado de IP

Para analizar el encabezado de IP, se tiene que definir como esta compuesto el encabezado, que se describen en los siguientes puntos:

1) Los primeros 8 bits almacenan la versión que indica el formato del encabezado de Internet y la longitud del encabezado IP en palabras de 32 bits indicando el comienzo de los datos [21].
2) Los siguientes 8 bits identifican el tipo de servicio que resumen en tres los parámetros de la calidad del servicio que son: un bajo retraso, alta fiabilidad y alto rendimiento. Estos son utilizados para dirigir la selección de los parámetros reales transmitiendo una trama de datos por la red [21].

Posición	Descripción normal	Descripción de alto desempeño
Bits 0-2	Precedente	-
Bit 3	0 = Retraso normal	1 = Bajo retraso
Bits 4	0 = Rendimiento normal	1 = Alto rendimiento
Bits 5	0 = Fiabilidad normal	1 = Alta fiabilidad
Bit 6-7	Reservado para uso futuro	-

Tabla.3.5 – Descripción por bits del tipo de servicio.

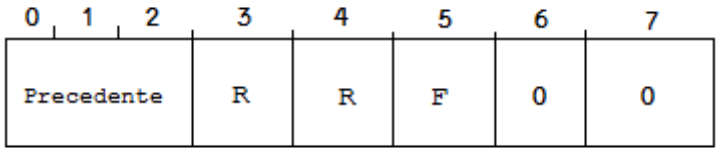


Figura.3.8 – Trama de los 8 bits del tipo de servicio.

3) Los siguientes 16 bits representan la longitud total de la trama de datos (encabezado + mensaje). Este campo permite una longitud de 65,535 Bytes. La computadora debe de estar lista para aceptar las tramas de datos de hasta 570 octetos (que llegan completos o en fragmentos). La recomendación es que se envíen tramas de datos grandes de 576 Bytes, así se asegura que el destino esta preparado para aceptar tramas de datos grandes [21].
4) Los siguientes 16 bits almacenan la identificación, que es un valor asignado para identificar al remitente ayudando a la colocación de los fragmentos de la trama de datos [21].
5) Los siguientes 16 bits contienen las banderas (Tabla.3.6 y Fig.3.9) y el fragmento de compensación, este campo indica de donde pertenece el fragmento

de la trama de datos, el fragmento de compensación se mide en 8 Bytes (64 bits). El primer fragmento ha compensar debe tener el valor de cero [21].

Posición	Descripción normal	Descripción de proceso
Bit 0	Reservado, debe ser cero	-
Bit 1	0 = Puede fragmentarse	1 = No se puede fragmentar
Bit 2	0 = Ultimo fragmento	1 = Más fragmentos

Tabla.3.6 – Descripción de los bits de bandera

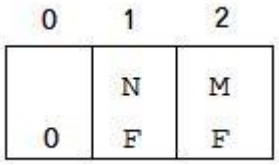


Figura.3.9 - Encabezado de los 3 bits de bandera.

6) Los siguientes 8 bits contienen el tiempo de vida (TTL por sus siglas en ingles *Time To Live*). Este campo indica el tiempo máximo permitido para que la trama de datos permanezca en el sistema de Internet. Si se tiene un valor cero, entonces la trama de datos debe ser destruida. El tiempo es medido en segundos y cada modulo que es procesado disminuye el TTL en uno [21].
7) Los siguientes 8 bits contienen el protocolo, este campo indica el siguiente nivel de protocolo usado en la trama de datos de Internet [21].
8) Los siguientes 16 bits contienen la suma de comprobación (Checksum), es la suma de los cambios en el campo del encabezado de IP, esto se realiza recalculando y verificando en cada punto que el encabezado de Internet es procesado [21].
9) Los siguientes 32 bits contienen la dirección IP origen, indica la dirección de la computadora origen [21].
10) Los últimos 32 contienen la dirección IP destino, indica hacia que computadora va dirigida la trama de datos [21]

Función:	IPHeader[37]
Entradas:	Datos del paquete (byBuffer), longitud total del paquete de IP (nReceived)
Salidas:	Tamaño del paquete, Longitud del encabezado, Contenido del encabezado de IP.
Variables importantes:	byVersionAndHeaderLength, byDifferenteatedServices, usTotalLength, usIdentification, usFlagsAndOffset, byTTL, byProtocol, sChecksum, uiSourceIPAddress, uiDestinationIPAddress, byHeaderLength, byIPDato.

Descripción:

1) Se obtienen los datos del paquete y la longitud del paquete de datos.
2) Se crea espacio en la memoria para almacenar temporalmente los datos obtenidos anteriormente.
3) Se leen los primeros ocho bits de los datos del paquete que contienen la versión de IP y se almacenan en byVersionAndHeaderLength.
4) Se leen los siguientes ocho bits que contienen el tipo de servicio y se almacenan en byDifferenteatedServices.
5) Se leen los siguientes ocho bits que contienen la longitud total del encabezado y se almacenan en usTotalLength.
6) Se leen los siguientes dieciséis bits que contienen el identificador y se almacenan en usIdentification.
7) Se leen los siguientes dieciséis bits que contienen los bits de compensación de fragmentación y las banderas y se almacenan en usFlagsAndOffset.
8) Se leen los siguientes ocho bits que contienen el tiempo de vida y se almacenan en byTTL.
9) Se leen los siguientes ocho bits que contienen el protocolo subyacente y se almacenan en byProtocol.
10) Se leen los siguientes dieciséis bits que contienen la suma de comprobación y se almacenan en sChecksum.
11) Se leen los siguientes treinta y dos bits que contienen la dirección IP origen y se almacenan en uiSourceIPAddress.
12) Se leen los últimos treinta y dos bits que contienen la dirección IP destino y se almacenan en uiDestinationIPAddress.
13) Se calcula el tamaño real del encabezado de de IP
14) Se copian los elementos del encabezado de IP

Tabla.3.7 – Función IPHeader.

[37] El codigo fuente lo puede encontrar en el apendice “D”

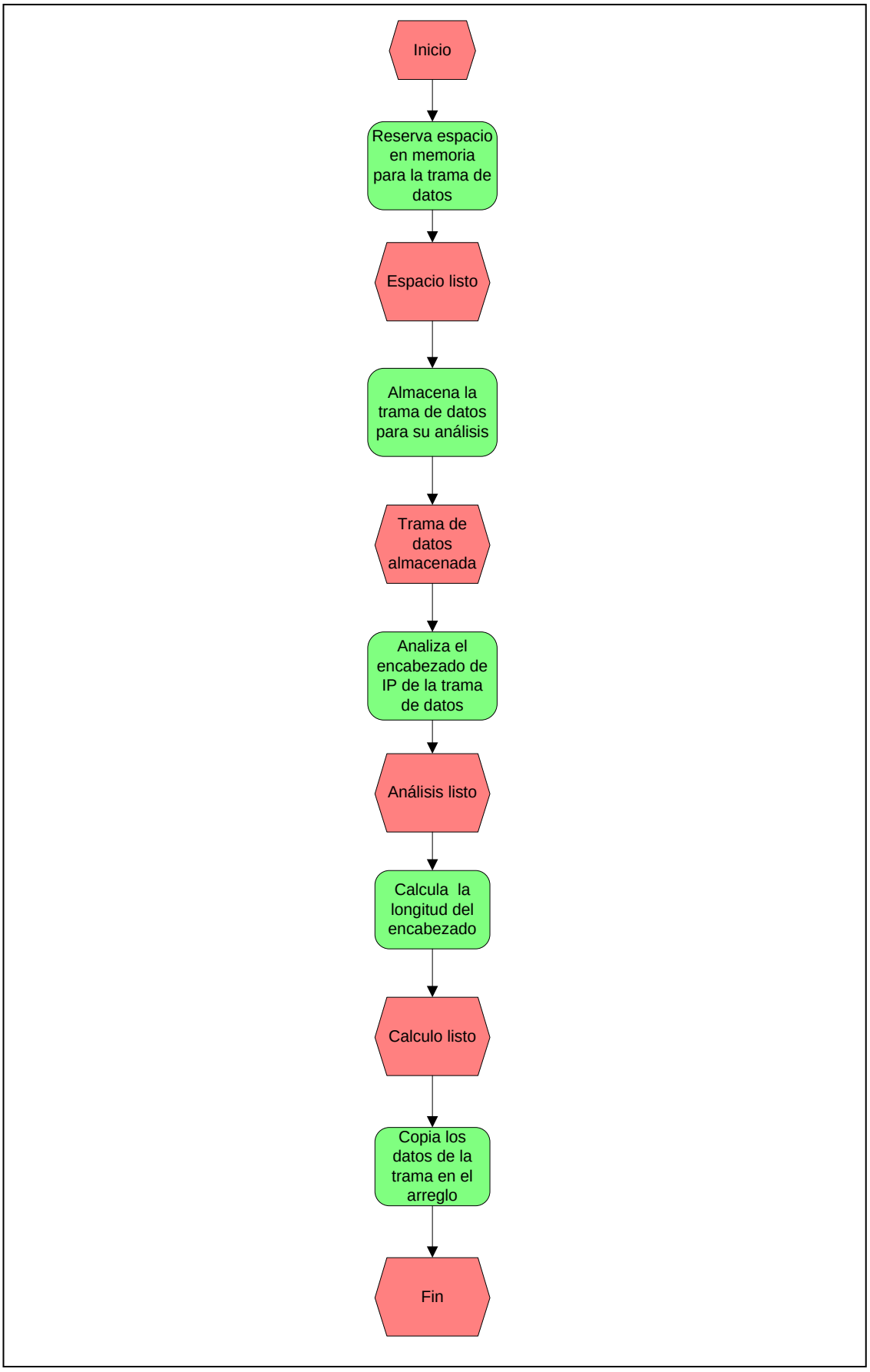


Figura.3.10. – Diagrama EPC función IPHeader

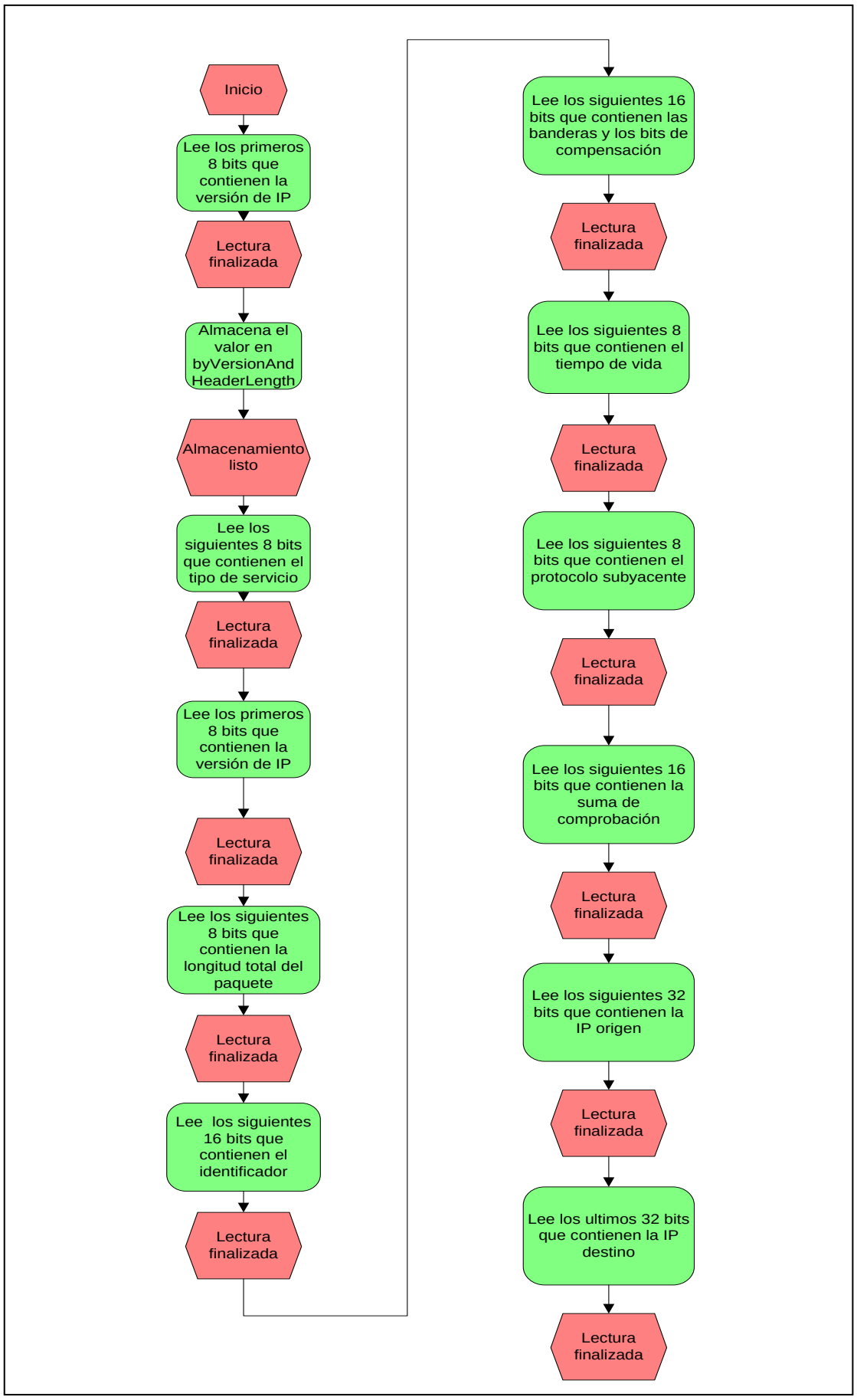


Figura.3.11. – Diagrama EPC análisis de encabezado de IP.

Nombre:	**Algoritmo de la función del encabezado de IP (IPHeader).**
Descripción:	Analiza y extrae la información del encabezado de IP
Variables importantes:	Espaciomem,BinaryReader, ByVersionAndHeaderLength, byDifferentiatedServices, usTotalLength, usIdentification, usflagsAndOffset, byTTL, byProtocol, sChacksum, uiSourceIPAddress, uiDestinationIPAddress, byHeaderLength,

```
IPHeader(byBuffer(),nReceived)
[Inicio]
   Espaciomem ◄──── MemoryStream
   BinaryReader ◄── Espaciomem

   byVersionAndHeaderLength[0,…,7] ◄──── BinaryReader [0,…,7]
   byDifferentiatedServices[0,…,7] ◄──── BinaryReader [7,…,15]
   usTotalLength[0,…,7] ◄──── BinaryReader [16,…,23]
   usIdentification[0,…,15] ◄──── BinaryReader [24,…,39]
   usflagsAndOffset[0,…,15] ◄──── BinaryReader [40,…,55]
   byTTL[0,…,7] ◄──── BinaryReader [56,…,63]
   byProtocol[0,…,7] ◄──── BinaryReader [64,…,71]
   sChecksum[0,…,15] ◄──── BinaryReader [72,…,87]
   uiSourceIPAddress[0,…,32] ◄── BinaryReader [88,…,119]
   uiDestinationIPAddress[0,…,32] ◄──── BinaryReader [120,…,151]
   byHeaderLength ◄──── byVersionAndHeaderLength[4,5,6,7]
   byHeaderLength ◄──── byHeaderLength*4
   Arry.Copy(byBuffer,byHeaderLength,byIPDato,0,_
             usTotalLength- byHeaderLength)
[FIN]
```

Algoritmo 3.2 - Algoritmo para el análisis del encabezado de IP

Esta información se almacena en una variable para después encontrar el tipo de protocolo que es encapsulado en la trama de datos de IP, el paso siguiente es analizar el encabezado de la trama de datos del protocolo encapsulado para identificar si es TCP, UDP o desconocido.

3.2.1.a.ii.2 Análisis de encabezado de TCP

El encabezado de TCP tiene el siguiente formato (fig.3.12):

<table>
<tr><td colspan="8">Puerto origen</td><td colspan="2">Puerto destino</td></tr>
<tr><td colspan="10">Número de secuencia</td></tr>
<tr><td colspan="10">Número de acknowledgment</td></tr>
<tr><td>Dato de compensación</td><td>Reservado</td><td>URG</td><td>ACK</td><td>PSH</td><td>RST</td><td>SYN</td><td>FIN</td><td colspan="2">Ventana</td></tr>
<tr><td colspan="8">Suma de comprobación</td><td colspan="2">Punto de urgencia</td></tr>
<tr><td colspan="9">Opciones</td><td>Base</td></tr>
<tr><td colspan="10">Dato</td></tr>
</table>

Figura.3.12 – Formato de encabezado de TCP

1) Los primeros 16 bits contienen el número del puerto de la computadora origen [22].
2) Los siguientes 16 bits contienen el número de puerto de la computadora destino [22].
3) Los siguientes 32 bits almacenan el número de secuencia, que es el número de secuencia del primer byte de datos del segmento [22].
4) Los siguientes 32 bits almacenan el número de reconocimiento (Acknowledgment), si el bit de control ACK contiene en su conjunto de archivos el valor de la siguiente secuencia numérica, el remitente del segmento espera a recibir información, una vez que la conexión es establecida estos valores son enviados [22].
5) Los siguientes 16 bits contienen las banderas y los datos de compensación, los primeros 4 bits indican el número de palabras de 32 bits en el encabezado de TCP. El encabezado de TCP (incluyendo opciones) es un número de 32 bits de longitud, los siguientes 6 bits se reservan para un uso futuro y siempre deben de permanecer en cero, por último los siguientes 6 bits de izquierda a derecha son los siguientes[22]:

   URG: Campo de punto urgente significativo (*Urgent Point field significant*)

   ACK: Campo de reconocimiento significativo (*Acknowledgment field significant*)

   PSH: Función de empuje (*Push Function*)

RST: Reinicio de conexión (*Reset the connection*)

SYN: Sincronización de números de secuencia (*Synchronize sequence numbers*)

FIN: No más datos de remitente (*No more data from sender*)

6) Los siguientes 16 bits almacenan el espacio de ventana, que es el número de octetos de datos que comienzan en el indicador del campo de reconocimiento que el remitente del segmento esta dispuesto a aceptar [22].
7) Los siguientes 16 bits contienen el Checksum, que es el complemento de 16 bits del complemento de suma, en complemento "a uno" de todas las palabras de 16 bits del encabezado y el texto [22].

La suma de comprobación también cubre un encabezado de 96 bits en un pseudo encabezado (fig.3.13) del prefijo del encabezado de TCP. Este pseudo encabezado contiene la dirección origen, la dirección destino, el protocolo, y la longitud de TCP. Esta información es llevada en el protocolo de Internet y es transferida a través de la interfaz de red en los argumentos de los resultados de las llamadas por el TCP sobre el IP.

<table>
<tr><td colspan="3">Dirección origen</td></tr>
<tr><td colspan="3">Dirección destino</td></tr>
<tr><td>Cero</td><td>PTCL</td><td>Longitud de TCP</td></tr>
</table>

Figura.3.13 – Pseudo encabezado de TCP

8) Los últimos 16 bits contienen el punto de urgencia, que es el campo que comunica el valor de compensación positiva del número de secuencia en este segmento. El punto de urgencia indica el número de secuencia del Byte después de los datos urgentes. Este campo sólo puede ser interpretado en segmentos de bit de control URG [22].

Función:	TCPHeader[38].
Entradas:	Datos del paquete TCP, Longitud total del paquete de TCP.
Salidas:	Tamaño del paquete, Longitud del encabezado, contenido del encabezado de TCP, Tamaño real del encabezado de TCP.
Variables importantes:	usSourcePort, usDestinationPort, uiSecuenceNumber, usAcknoledgementNumber, usDataOffsetAndFlags, usWindows, sChecksum, usUrgentPointer, byHeaderLength, usMessageLength, byTCPDato.

Descripción:

1) Se obtienen los datos del paquete y la longitud del paquete de datos.
2) Se crea espacio en la memoria para almacenar temporalmente los datos obtenidos anteriormente.
3) Se leen los primeros dieciséis bits que contienen el puerto de origen y se almacenan en usSourcePort.
4) Se leen los siguientes dieciséis bits que contienen el puerto destino y se almacenan en usDestinationPort.
5) Se leen los siguientes treinta y dos bits que contienen el número de secuencia y se almacenan en uiSecuenceNumber.
6) Se leen los siguientes treinta y dos bits que contienen el número de reconocimiento y se almacenan en usAcknoledgementNumber.
7) Se leen los siguientes dieciséis bits que contienen las banderas de compensación de datos y se almacenan en usDataOffsetAndFlangs.
8) Se leen los siguientes dieciséis bits que contienen el espacio de ventana y se almacenan en usWindows.
9) Se leen los siguientes dieciséis bits que contienen las banderas de compensación de datos y se almacenan en usDataOffsetAndFlangs.
10) Se leen los siguientes dieciséis bits que contienen la suma de comprobación y se almacenan en sChecksum.
11) Se leen los siguientes dieciséis bits que contienen el punto de urgencia y se almacenan en usUrgentPointer.
12) Calculo del tamaño de la longitud del encabezado de TCP.
13) Se realiza el cálculo de la longitud del mensaje restándole a la longitud total del paquete de TCP la longitud del encabezado.
14) Se copian los elementos del encabezado de TCP

Tabla.3.8– Función TCPHeader.

[38] El código fuente lo puede encontrar en el apéndice "D"

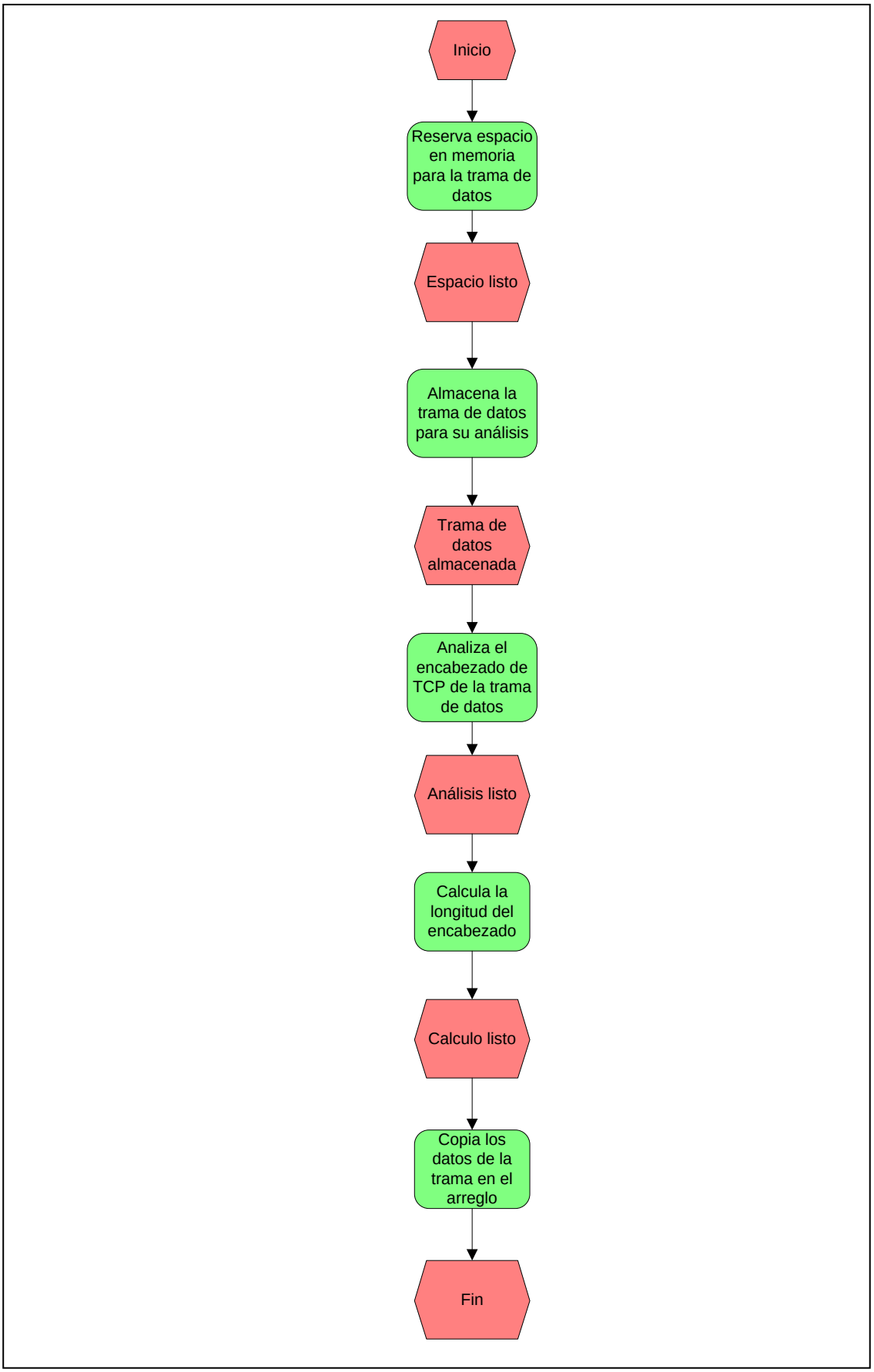


Figura.3.14. – Diagrama ECP función TCPHeader

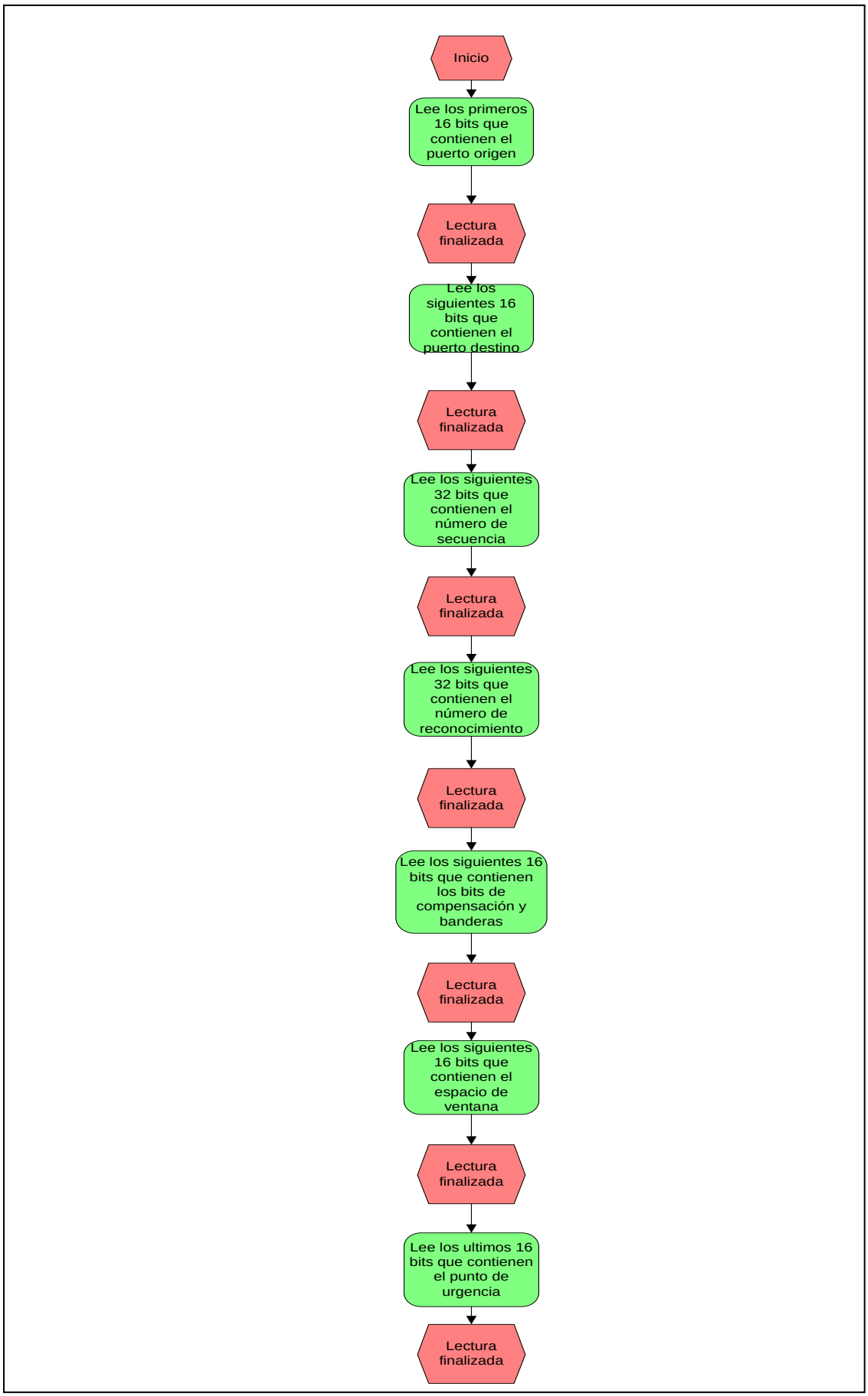


Figura.3.15 – Diagrama EPC de análisis de encabezado de TCP

Nombre:	**Algoritmo de la función TCPHeader**
Descripción:	Analiza y extrae la información del encabezado de TCP
Variables importantes:	Espaciomem, BinaryReader, usSourcePort, usDestinationPort, uiSequenceNumber, uiAcknowledgementNumber, usDataOffsetEndFlangs, usWindows, sChechksum, usUrgentPointer, byHeaderlength, usMessagLengtn.

```
IPHeader(byBuffer(),nReceived)
[Inicio]
   Espaciomem ←———— MemoryStream
   BinaryReader ←—— Espaciomem

   usSourcePort[0,…,15] ←— BinaryReader[0,…,15]

   usDestinationPort[0,…,15] ←— BinaryReader[16,…,31]

   uiSequenceNumber[0,…,32] ←—— BinaryReader[32,…,63]

   uiAcknowledgementNumber[0,…,32] ←— BinaryReader[64,…,95]

   usDataOffsetEndFlags[0,…,15] ←———— BinaryReader[96,…,111]

   usWindows[0,…,15] ←———— BinaryReader[112,…,127]

   sChecksum[0,…,15] ←———— BinaryReader[128,…,143]

   usUrgentPoniter[0,…,15] ←———— BinaryReader[144,…,159]

   byHeaderlength ←———— usDataOffsetAndFlags[96,97,98,99]

   byHeaderlength ←——— byHeaderlength*4

   usMessagLength ←—— nReceived - byHeaderlength

   Array.Copy(byBuffer,byHeaderLength,byTCPDato,0,_
              nReceived - byHeaderLength)
[FIN]
```

Algoritmo 3.3 – Algoritmo para el análisis del encabezado de TCP

3.2.1.a.ii.3 Análisis de encabezado de UDP.

Para el caso de UDP se tiene el siguiente formato (fig 3.16):

Puerto de origen	Puerto destino
Longitud	Suma de control
Octetos de datos…	

Figura3.16 – Formato del encabezado de UDP

1) Los primeros 16 bits almacenan el puerto de origen, indicando el puerto del proceso de emisión que es también el puerto al cual la respuesta se dirige en ausencia de otra información (si no es utilizada su valor debe ser cero) [23].
2) Los siguientes 16 bits almacenan el puerto de destino, tiene significado dentro del contexto de una dirección de destino en un entorno de Internet [23].
3) Los siguientes 16 bits representan la longitud en Bytes de la trama de datos de usuario, incluyendo el encabezado y los datos [23].
4) Los últimos 16 bits contienen el Checksum, que es el complemento "a uno" de los 16 bits de la suma de complementos "a uno" de las palabras de la combinación de un pseudo encabezado (fig.3.17) construido con información del encabezado de IP, el encabezado de UDP y los datos, rellenada con bytes de valor cero en la parte final hasta tener un múltiplo de dos bytes [23].

Dirección de origen		
Dirección de destino		
cero	Protocolo	Longitud UDP

Figura.3.17 – Pseudo encabezado de UDP

El pseudo encabezado antecede al encabezado de UDP que contiene la dirección origen, destino, el protocolo y la longitud de UDP. Esto proporciona protección de las tramas de datos mal encaminadas.

Función:	UDPHeader[39]
Entradas:	Datos del paquete de UDP, Longitud del paquete de UDP
Salidas:	Tamaño del paquete, Longitud del encabezado, Contenido del encabezado de UDP, Tamaño real del encabezado de UDP
Variables importantes:	usSourcePort, usDestinationPort, usLongth, sChecksum, byUDPDato

Descripción:

1) Se obtienen los datos del paquete y la longitud del paquete de datos.
2) Se crea espacio en la memoria para almacenar temporalmente el los datos obtenidos anteriormente.
3) Se leen los primeros dieciséis bits que contienen el puerto de origen y se almacenan en usSourcePort.
4) Se leen los siguientes dieciséis bits que contienen el puerto destino y se almacenan en usDestinationPort.
5) Se leen los siguientes dieciséis bits que contienen la longitud del paquete de UDP y se almacenan en usLongth.
6) Se leen los siguientes dieciséis bits que contienen la suma de comprobación y se almacenan en sChecksum.
7) Se copian los elementos del encabezado de UDP

Tabla.3.9 - Función UDPHeader

[39] El codigo fuente lo puede encontat en el apéndice "D"

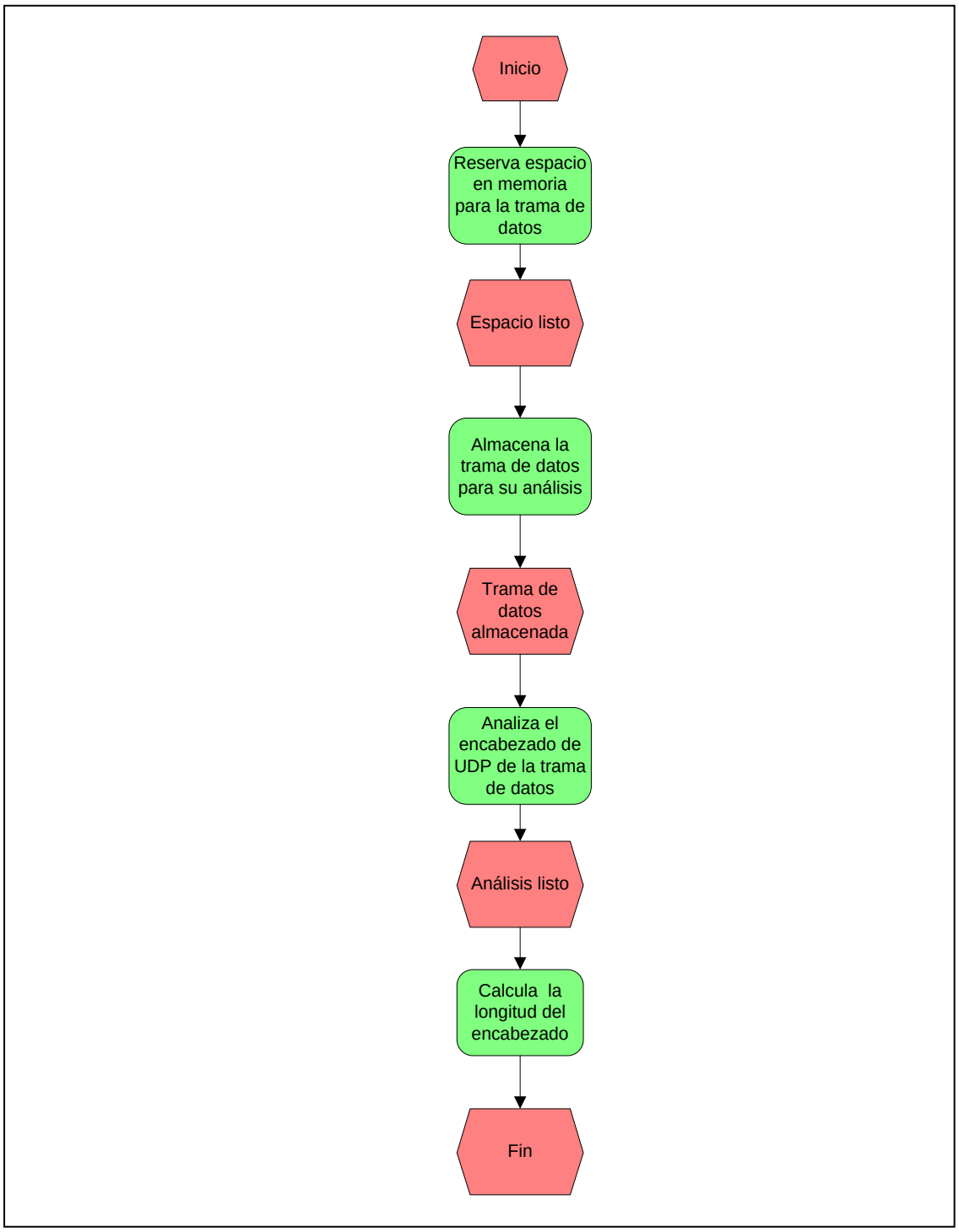


Figura.3.18 Diagrama EPC de la función UDPHeader.

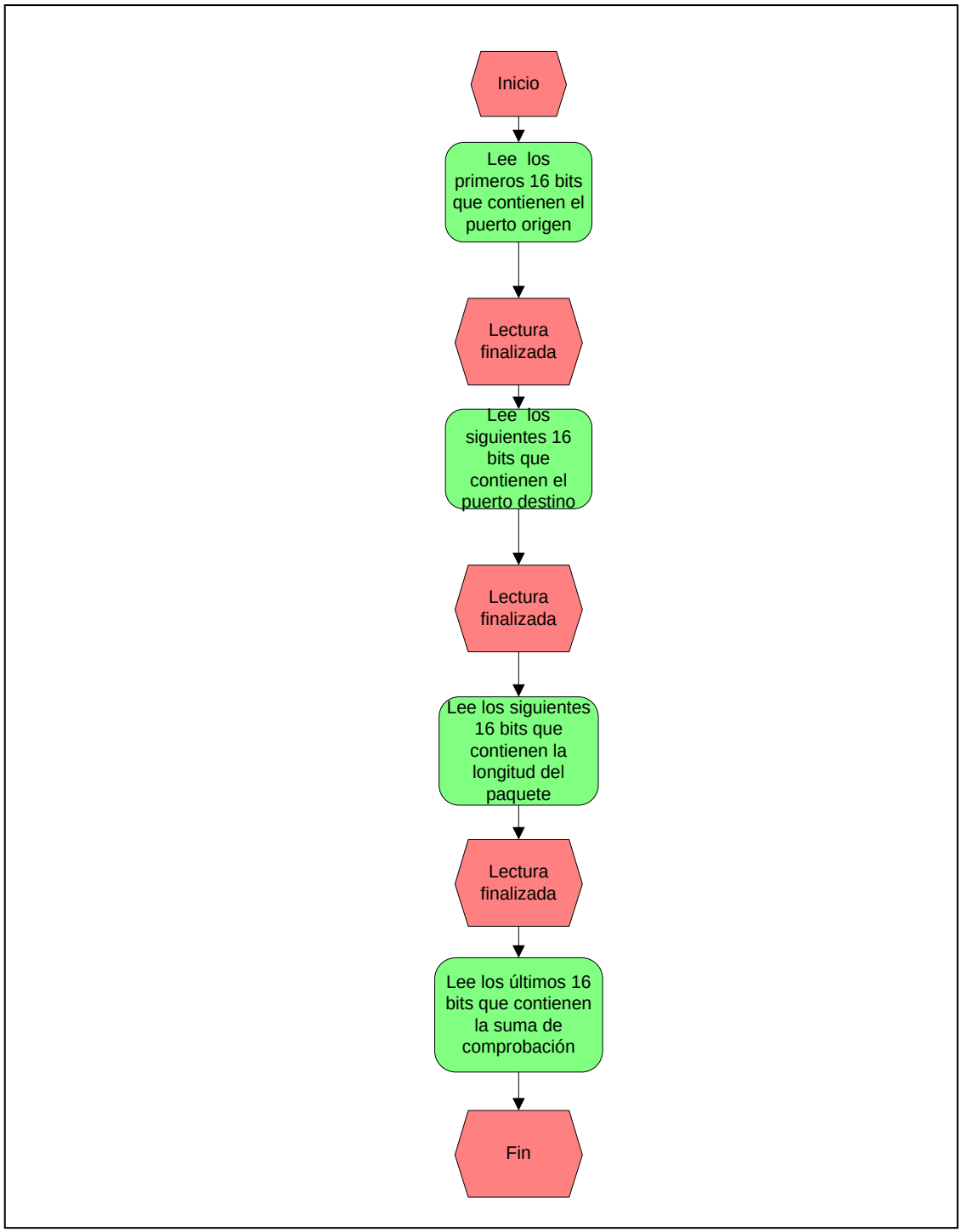


Figura.3.19 Diagrama EPC del análisis del encabezado de UDP

Nombre:	**Algoritmo de la función UDPHeader.**
Descripción:	Analiza y extrae los datos del encabezado de UDP.
Variables importantes:	Espaciomem, BinaryReader, usSourcePort, usDestinationPort, usLength, sChecksum.

```
UDPHeader(byBuffer(),nReceived)

[Inicio]
   Espaciomem ←——— MemoryStream
   BinaryReader ←——— Espaciomem

   usSourcePort[0,…,15] ←—— BinaryReader[0,…,15]

   usDestinationPort[0,…,15]←—— BinaryReader[16,…,31]

   usLength[0,…,15]←——— BinaryReader[32,…,47]

   sChecksum[0,…,15]←——— BinaryReader[48,…,63]

   Array.Copy(byBuffer,8,byUDPDato,0,_
              nReceived - 8)

[FIN]
```

Algoritmo 3.4 Algoritmo para el análisis del encabezado UDP.

Tanto para TCP como UDP se utiliza el puerto 53 para el uso del sistema de nombres de dominio (DNS por sus siglas en ingles, *Domain Names System*) que es descrito en las solicitudes de comentarios 1010 (RFC[40] por sus siglas en inglés *Requests for Comments*), RFC 1304, RFC 1035 y RFC 1183. DNS fue desarrollado para proporcionar un sistema que convierte direcciones IP en nombres de dominio fácilmente reconocibles [2] y en el caso de que el protocolo no se conozca solo se señala el protocolo como desconocido.

[40] Las notas de la serie de solicitudes de los comentarios del documento (RFC) contienen notas técnicas y organizativas de Internet. Cubren muchos aspectos de las redes de computo, incluidos los protocolos, procedimientos, programas y conceptos, disponible en línea en: [http://www.rfc-editor.org/ RFCoverview.html]

3.2.1.a.ii.4 Análisis de encabezado de DNS

Dentro del protocolo de sistemas de nombre de dominio los datos son llevados en un formato llamado "mensaje". El nivel mas alto del formato de mensaje es divido en cinco secciones (algunas de las cuales están vacías) mostradas a continuación [24]:

1) Encabezado
2) Pregunta.- Pregunta por el nombre del servidor.
3) Respuesta.- RRs[41] contestación de la pregunta
4) Autoridad.- RRs señalando hacia una autoridad
5) Adicional.- RRs propiedad de información adicional

El encabezado incluye los campos que especifican cuál de las cinco secciones están presentes, y también especifica si el mensaje es una pregunta o una respuesta, también dentro del encabezado se encuentra el identificador de 16 bits asignado por el programa que genera cualquier clase de pregunta. Este identificador copia la respuesta correspondiente y puede ser usado por el solicitante para que coincida la respuesta con la pregunta que queden pendientes [24].

Después del encabezado se deriva la sección de preguntas que contiene los campos que describen la pregunta del nombre del servidor. Las tres últimas secciones tienen una lista posiblemente vacían de registros de recurso encadenados (RRs). La sección de respuesta contiene RRS que contesta la pregunta; la sección de autoridad contiene RRS que señala hacia un servidor de nombre con autoridad; la sección de registros adicional contiene RRS que se relacionan con la pregunta [24].

[41] Registros de recursos encadenados (RRs por sus siglas en ingles *Concatenated Resource Records*)

Función:	DNS Header[42]
Entrada:	Datos del paquete de DNS, Longitud del paquete de DNS
Salida:	Identificación, banderas, total de preguntas, total de respuestas, autoridad RRs y propiedad de información adicional.
Variables importantes:	usIdentification, usFlags, usTotalAnswerRRs, usTotalQuestions, usTotalAuthorityRRs, usTotalAdditionalRRs.

Descripción:

1) Se obtienen los datos del paquete y la longitud del paquete de datos.
2) Se crea espacio en la memoria para almacenar temporalmente los datos obtenidos anteriormente.
3) Se leen los primeros dieciséis bits que contienen la identificación y se almacena en usIdentification.
4) Se leen los siguientes dieciséis bits que contienen las banderas y se almacena en usFlags.
5) Se leen los siguientes dieciséis bits que contienen el número de consultas totales de la lista de consuntas y se almacena en usTotalQuestions.
6) Se leen los siguientes dieciséis bits que contienen el número total de respuestas de la lista de respuestas y se almacena en usTotalAnswerRRs.
7) Se leen los siguientes dieciséis bits que contienen la lista de autorizaciones y se almacenan en usTotalAuthorityRRs.
8) Se leen los últimos dieciséis bits que contienen el número de entradas de la fuente adicional de la lista y se almacena en usTotalAdditionalRRs.

Tabla.3.10 – Función DNSHeader.

[42] El código fuente lo puede encontrar en el apéndice "D"

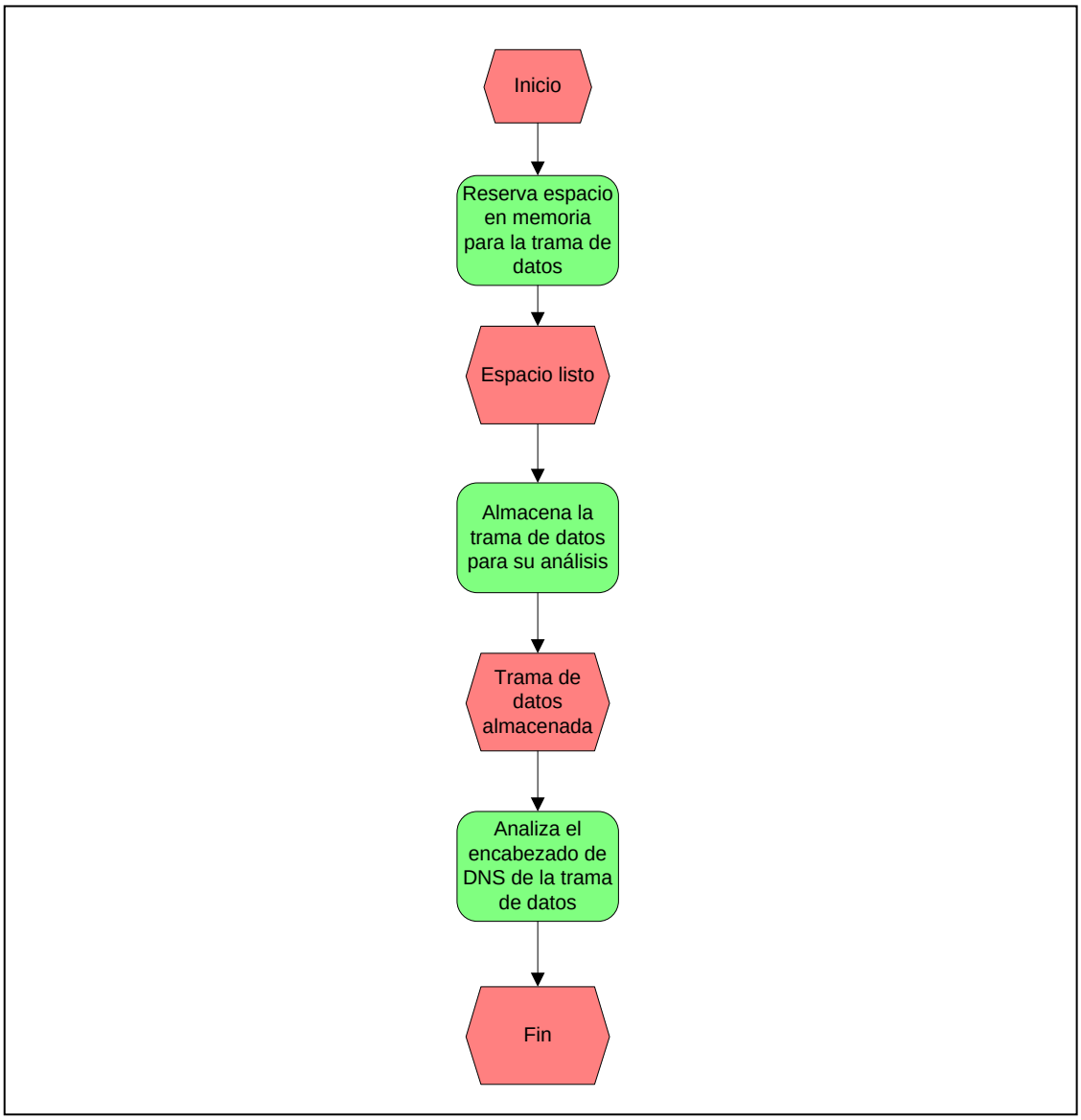


Figura.3.20 – Diagrama EPC de la función DNSHeader.

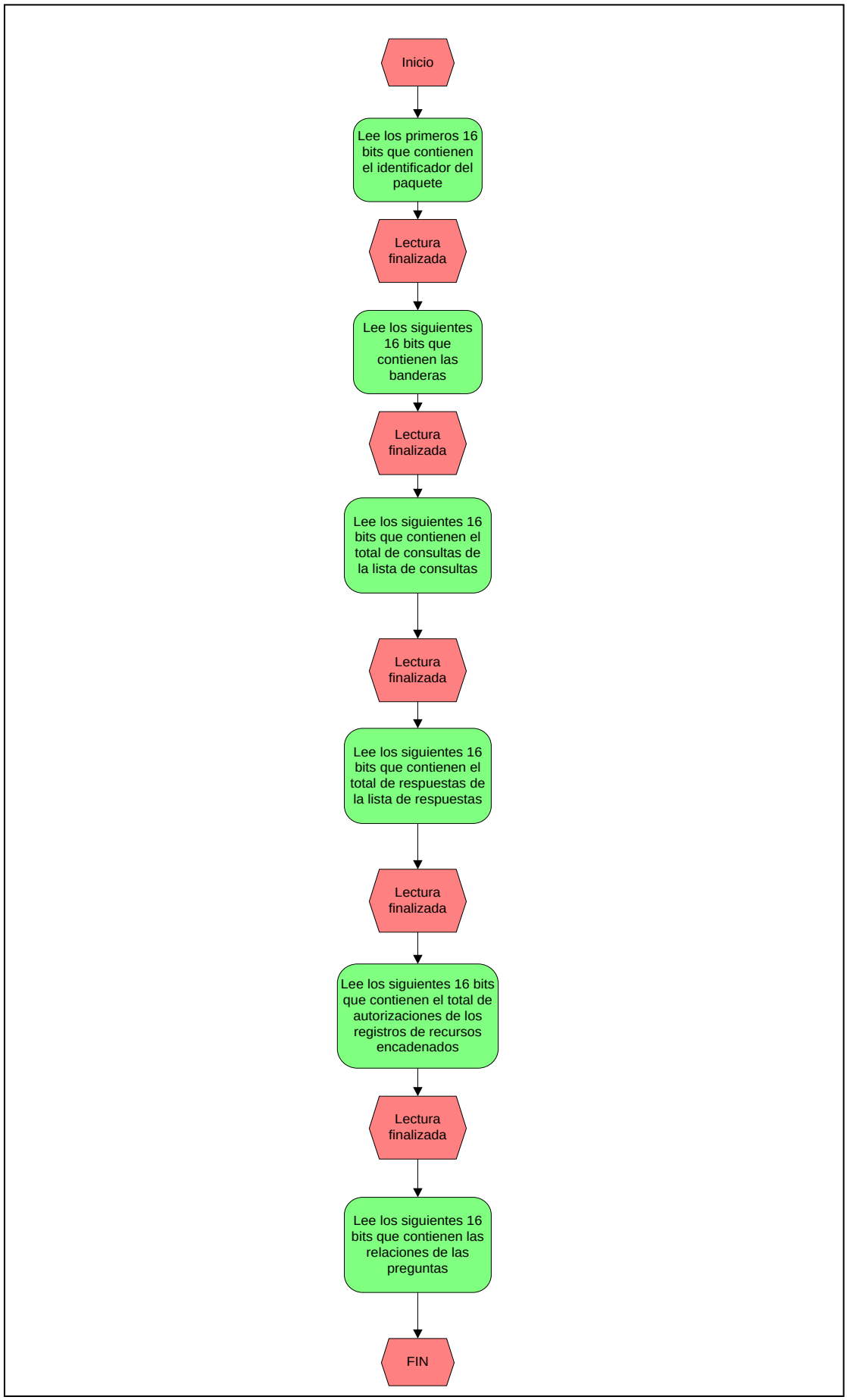


Figura.3.21 – Diagrama EPC del análisis del encabezado de DNS.

Nombre:	**Algoritmo de la función DNSHeader.**
Descripción:	Análisa y extrae la información del encabezado de DNS
Variables importantes:	usIdentification, usFlags, usTotalAnswerRRs, usTotalQuestions, usTotalAuthorityRRs, usTotalAdditionalRRs.

```
DNSHeader(byBuffer(),nReceived)
[Inicio]
   Espaciomem ◄──────── MemoryStream
   BinaryReader◄──── Espaciomem
   usIdentification[0,…,15]◄──── BinaryReader[0,…,15]
   usFlags[0,…,15]◄──────── BinaryReader[16,…,31]
   usTotalQuestions[0,…,15]◄──── BinaryReader[32,…,47]
   usTotalAnswerRRS[0,…,15]◄──── BinaryReader[48,…,63]
   usTotalAuthorityRRS[0,…,15] ◄──────── BinaryReader[64,…,79]
   usTotalAdditionalRRS[0,…,15]◄──────── BinaryReader[80,…,95]
[FIN]
```

Algoritmo 3.5 – Algoritmo para el análisis el encabezado de DNS.

3.2.1.b Detección de paquetes de entrada / salida.

Este módulo obtiene los parámetros de información estadística: número paquetes de datos que se reciben y envían de la red.

Para la construcción de este modulo se utiliza el espacio de nombres `System.Net.NetworkInformation()` que permite obtener información de los adaptadores de red que se encuentran instalados en el equipo mediante la clase `IPGlobalProperties` (Tabla 3.11) que permite el acceso a esta información a través del método `GetIPGlobalProperties()`, este es un método que devuelve las propiedades de IP con los datos estadísticos del sistema[43] [5].

Espacio de nombres	Descripción
System.Net.NetworkInformation	Proporciona acceso a los datos del tráfico de red y a los cambios de direcciones para el equipo local.
Clases	**Descripción**
IPGlobalProperties	Proporciona información sobre la conectividad de red del equipo local.
Método	**Descripción**
GetIPGlobalProperties	Obtiene un objeto que proporciona información sobre estadísticas de tráfico y conectividad de red del equipo local.

Tabla3.11 - Descripción del contenido del espacio de nombres System.Net.NetworkInformation[44].

Con la función de estadísticas (Tabla.3.12), el diagrama EPC de estadísticas (fig.3.22) y su algoritmo se obtiene la información relacionada a la transmisión y recepción de paquetes del adaptador de red.

[43] Si se requiere obtener información estadística adicional de un adaptador en especifico, se puede utilizar el método GetIPv4Statistics() de la clase NetworkInterface.
[44] Para mayor información de este espacio de nombres remitirse a: http://msdn.microsoft.com/es-es/library/system.net.networkinformation%28v=vs.80%29.aspx

Función:	Estadísticas[45]
Entrada:	Paquetes recibidos, Paquetes enviados
Salida:	Paquetes de entrada, Paquetes de salida
Variables Importantes:	Datos, Cadena

Descripción:

1) Se crea una matriz dinámica[46] para el almacenamiento de los datos.
2) Inicia una solicitud de estadísticas de tráfico y conectividad de red del equipo local.
3) Inicia una solicitud de estadísticas del protocolo de internet del equipo local.
4) El valor de las estadísticas obtenidas se asigna a la matriz de Datos.
5) Los datos son colocados en una cadena para ser mostrados.

Tabla.3.12 – Función Estadísticas

[45] El código fuente lo puede encontrar en el apéndice "E"
[46] Se denomina matriz dinámica a las matrices que tienen un número variable de elementos.

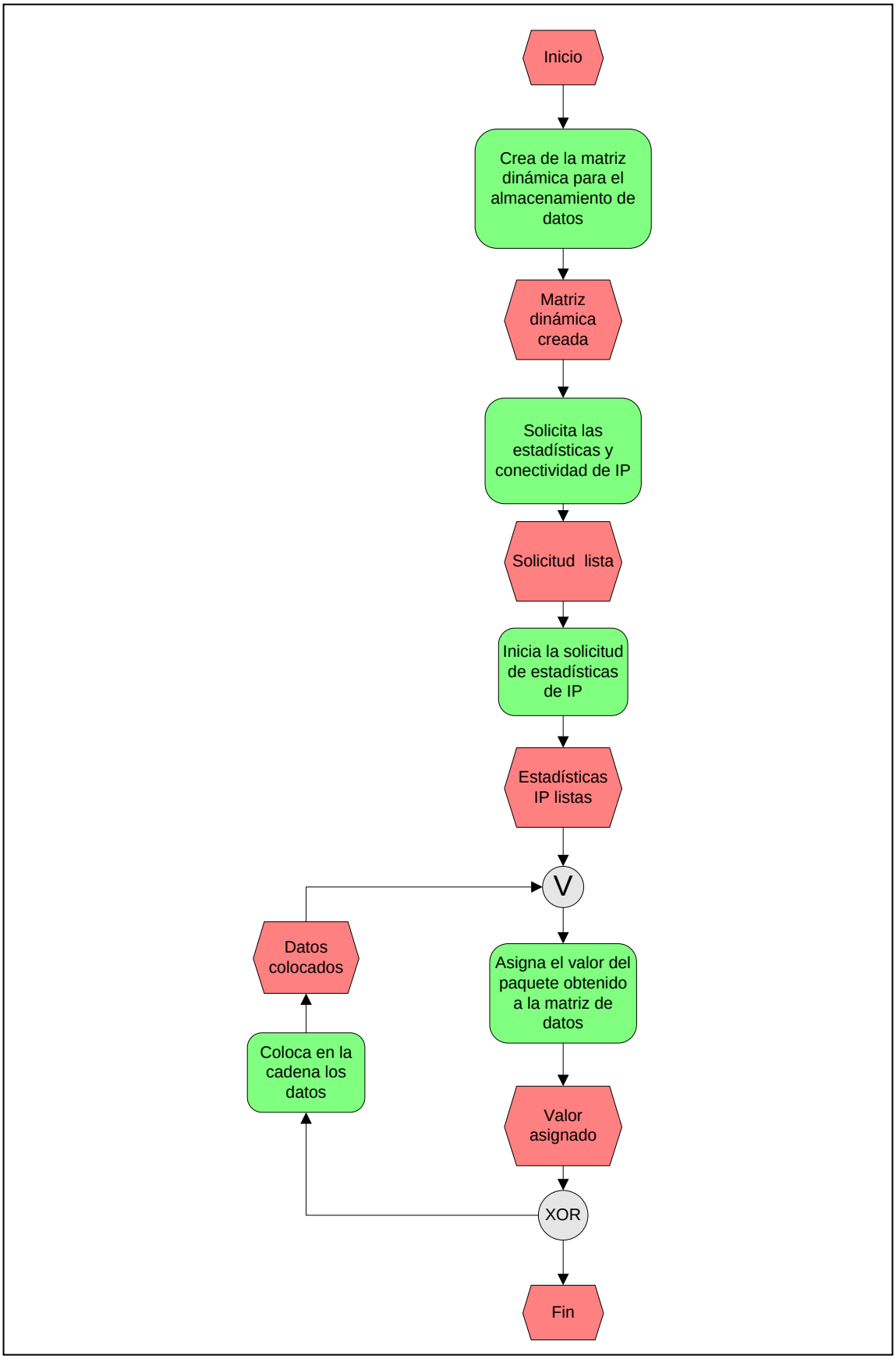


Figura.3.22 – Diagrama EPC para la recepción de paquetes de entrada / salida.

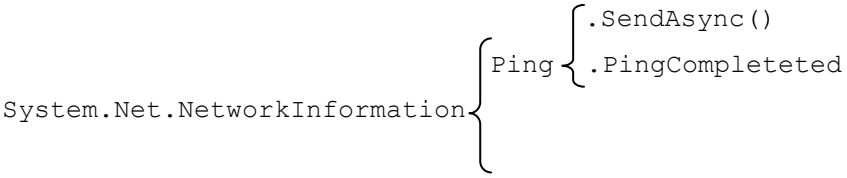
Nombre:	**Algoritmo paquetes de entrada / salida.**
Descripción:	Adquiere el número de paquetes que entran y salen del adaptador de red.
Variables importantes:	Datos, Cadena.

```
[Inicio]
  Datos() de tipo entero
  Solicitud de las propiedades de IP
  Obtención de estadísticas de las propiedades de IP
             Datos← Obteniendo los paquetes ( _
                          "Paquetes de Entrada"←── Paquetes recibidos
                          "Paquetes de Salida"←──Salida de paquetes)
       For Cadena=1,2,… hasta Datos
             Cadena(i)←── Datos
             Listbox ←── Cadena(i)
[Fin]
```

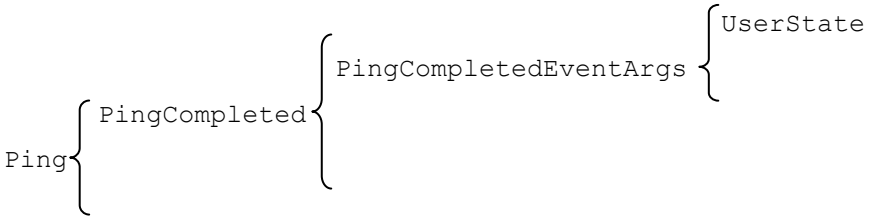
Algoritmo 3.6 – Algoritmo para capturar paquetes de entrada / salida.

3.2.1.c Comprobación de conexión (PING).

El siguiente módulo permite comprobar de forma rápida si un equipo se encuentra activo o no. En el espacio de nombres `System.Net.NetworkInformation` se encuentra la clase `Ping`, que tiene el mismo funcionamiento que la herramienta *ping*[47] (por sus siglas en ingles *Paket Internet Groper*, Rastreador de paquetes de internet) de los sistemas operativos, en donde se sabe si es posible la conexión con otro equipo mediante el envió de paquetes ICMP[48], si la conexión es exitosa se obtiene la respuesta del equipo indicando que el equipo esta activo, de lo contrario se enviara un mensaje de fallo en la conexión.



La clase `Ping` utiliza el método `SendAsync()` para efectuar una comprobación de manera asíncrona y así evitar que el programa se quede en espera de una respuesta generando errores en tiempo de ejecución, por lo cual se utilizan también los procesos de captura de errores `Try…Catch` que procesan los errores en tiempo de ejecución, para que ejecute otras instrucciones en caso de producirse un error, después con el objeto `PingCompletedEventArgs` (de su propiedad `UserState`) entrega un argumento a `SendAsync()` que lo utiliza como un elemento para agrega la dirección de conexión, nombre o URL[49] del equipo al que se envían los paquetes, el resultado, la longitud de los datos recibidos y tiempo empleado en la transición.



[47] Para mas información remítase al glosario
[48] Para mas información remitirse al glosario
[49] El localizador uniforme de recursos o URL (por sus siglas en ingles, *Uniform Resource Locator*), es una secuencia de caracteres bajo una forma estándar para darle nombre a determinados recursos en la red.

Los datos que entrega el modulo de comprobación de conexión de equipo se encuentran en la función HacerPing (Tabla.3.13), el diagrama ECP (fig.3.23) y su algoritmo.

Función:	HacerPing[50].
Entrada:	Nombre o dirección IP, Dirección IP, Estado de conexión, tamaño del paquete, tiempo de vida.
Salida:	Dirección IP, Número de Eco, Tiempo de vida, Tamaño del paquete.
Variables Importantes:	L.- Tamaño del paquete de datos. w.- Tiempo de vida I.- Índice
Descripción: 1) Crear el objeto ping. 2) Realiza un intento de conexión con la dirección IP del equipo al que se desea la conexión. 3) El *número de eco* o intentos de conexión puede ser satisfactoria o no. 4) El *tiempo de vida* es el tiempo que se emplea en la comunicación y se mide en milisegundos. 5) El *tamaño del paquete o buffer* es la matriz de paquetes de datos recibidos como respuesta y son de tipo byte. Este programa también cuenta con la ventaja de poder ampliar el numero de *eco*[51], el *tamaño del buffer*[52] (Bytes) y el *tiempo de vida*[53] (TTL), obtenido así un modulo muy completo con el cual el administrador de red puede hacer pruebas de conexión personalizadas.	

Tabla.3.13 – Función HacerPing

[50] El codigo fuente lo puede encontrar en el apéndice "F"
[51] Para mas información remítase al glosario
[52] Para mas información remítase al glosario
[53] Para mas información remítase al glosario

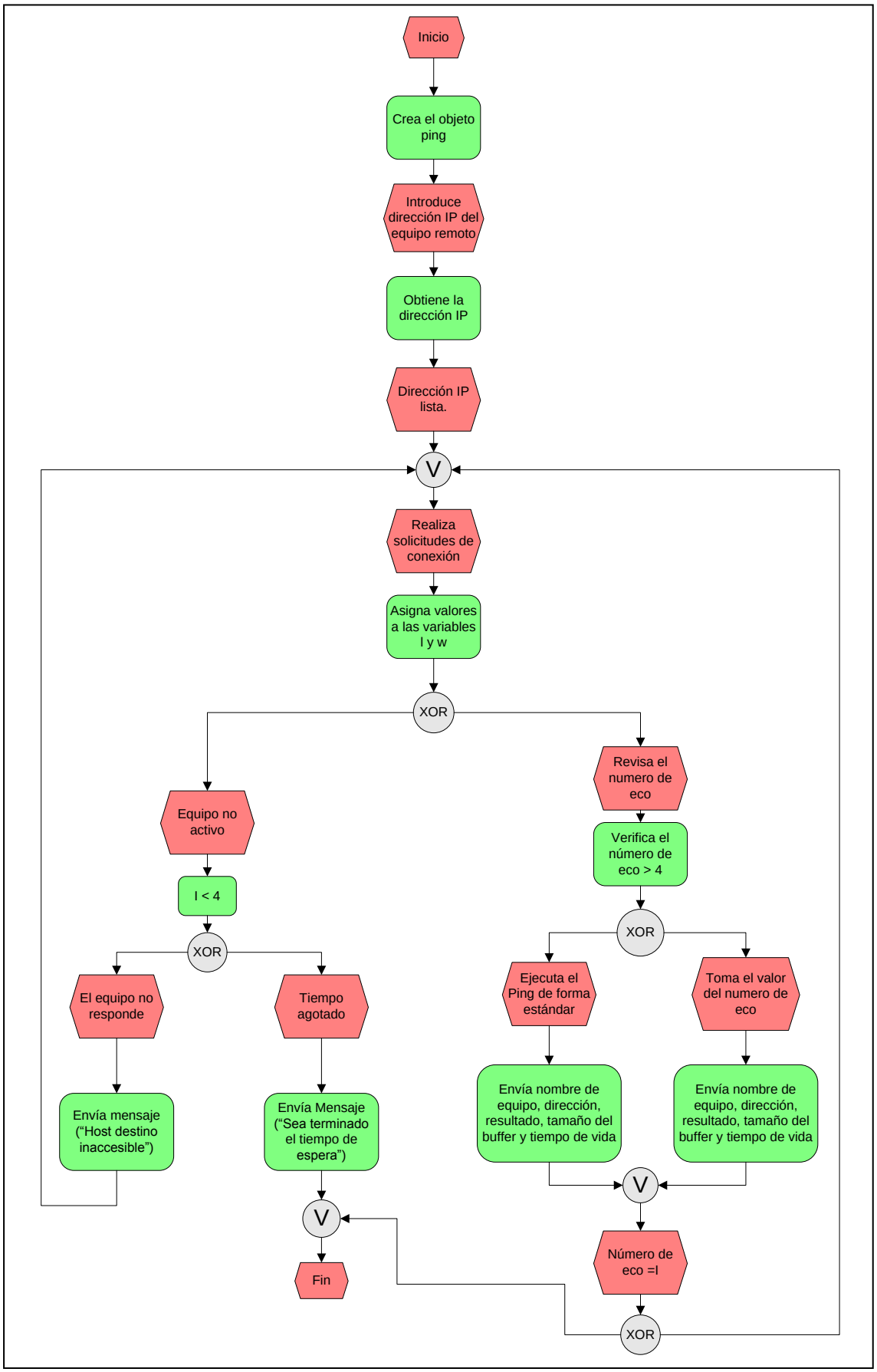


Figura.3.23 – Diagrama EPC comprobación de conexión (PING).

Nombre:	**Algoritmo del rastreador de paquetes de Internet.**
Descripción:	Envía una solicitud de conexión a un equipo remoto para verificar si se encuentra activo o no.
Variables importantes:	L.- Tamaño del paquete de datos. w.- Tiempo de vida I.- Índice

```
[Inicio]
   Crear el objeto Ping dentro del método asociado al botón "compara"
      l,w,I  tipo  Integer
      l ←—— tamaño del paquete de datos
      w ←—— tiempo de vida
      Para I = 1,2… hasta el valor de NumericUpDown1
            if (Numero de eco > 4)
                        Línea(I) ←—— Nombre ó dirección IP
                        Línea(I) ←—— Dirección IP
                        Línea(I) ←—— Conexión exitosa o fallida
                        Línea(I) ←—— Tamaño de Buffer(l)
                        Línea(I) ←—— Tiempo de conexión(w)
            else
                        Línea(I) ←—— Nombre ó dirección IP
                        Línea(I) ←—— Dirección IP
                        Línea(I) ←—— Conexión exitosa o fallida
                        Línea(I) ←—— Tamaño de Buffer
                        Línea(I) ←—— Tiempo de conexión
            if (I < 4)
                  Mensaje("Host de destino inaccesible")
            else
                  Mensaje("Sea terminado el tiempo de espera del host")
[Fin]
```

Algoritmo 3.7 – Algoritmo para la comprobación de conexión.

3.2.1.d Detección de interfaces de red.

Este módulo sirve para obtener información como el estado actual del adaptador, el tipo de adaptador, su dirección, etc, de las tarjetas de red instaladas en un equipo.

Para construir este módulo se utiliza el espacio de nombres `System.Net.NetworkInformation` que permite obtener las configuraciones de los adaptadores de red que se encuentran instalados en un equipo de computo, el espacio de nombres `System.Net.NetworkInformation` contiene la clase `NetworkInterfaces` (tabla 3.14) que trabaja con propiedades y métodos que aportan información que existe en cada uno de los adaptadores de red, esta información se obtiene mediante el método `GetAllNetworkInterfaces()` que enumera las interfaces de red.

Clase	Descripción
NetworkInterfaces	Proporciona información estadística y de configuración para una interfaz de red.
Método	**Descripción**
GetAllNetworkInterfaces	Devuelve objetos que describen las interfaces de red del equipo local.

Tabla.3.14 – Descripción de la clase NetworkInterfaces

Después que el método `GetAllNetworkInterfaces()` enumera las interfaces se realiza la solicitud de la información de los adaptadores de red instalados en el equipo como el identificador (Id), nombre, descripción, tipo, velocidad, MAC, (por sus siglas en ingles *Media Acces Control*, Control de acceso al medio) dirección IP y el estado del adaptador, dentro de esta información también se encuentra la información del *loopback*[54], cuyo índice podemos obtener mediante la propiedad `LoopbackInterfaceIndex` que contiene el índice de la interfaz de bucle invertido.

[54] Interface de red virtual.

La función adaptadores de red (Tabla.3.15), el diagrama EPC (fig.3.24) y su algoritmo obtienen la información de las interfaces de red instaladas en una computadora.

Función:	Adaptadores de red[55].
Entradas:	Id, Name, Description, Type, Speed, MAC, IP, Status.
Salidas:	Identificador, Nombre, Descripción, Tipo de interface, Dirección física, Dirección IP, Estado.
Variables importantes:	Adaptador, Direcciones, Dirección

Descripción:

1) Detecta y enumera los adaptadores de red instalados en el equipo
2) Obtén de cada uno de los adaptadores de red el ID, nombre de la interface de red y el tipo.
3) Obtén el tipo de interfaz de red con la propiedad `NetworkInterfaceType()` en donde se obtiene el tipo de adaptador de red ya sea: Ethernet, Token Ring, Wireless, loopback, etc.
4) Obtén las direcciones asociadas a cada adaptador mediante el método `GetPhysicalAddress()` que devuelve una matriz de tipo Byte con la dirección física o MAC y `GetIPProperties()` que obtiene las propiedades de configuración del protocolo de internet que afectan directamente al DNS (de sus siglas en ingles *Domain Name System*, sistema de nombres de dominio) y las direcciones asignadas al adaptador de red.
5) Por ultimo obtén el estado de la tarjeta de red mediante la propiedad `OperationalStatus()` que indica el estado actual del adaptador que puede ser activo, caído, en espera, etc.

Tabla.3.15 – Función adaptadores de red.

[55] El codigo fuente lo puede encontrar en el apéndice “G”.

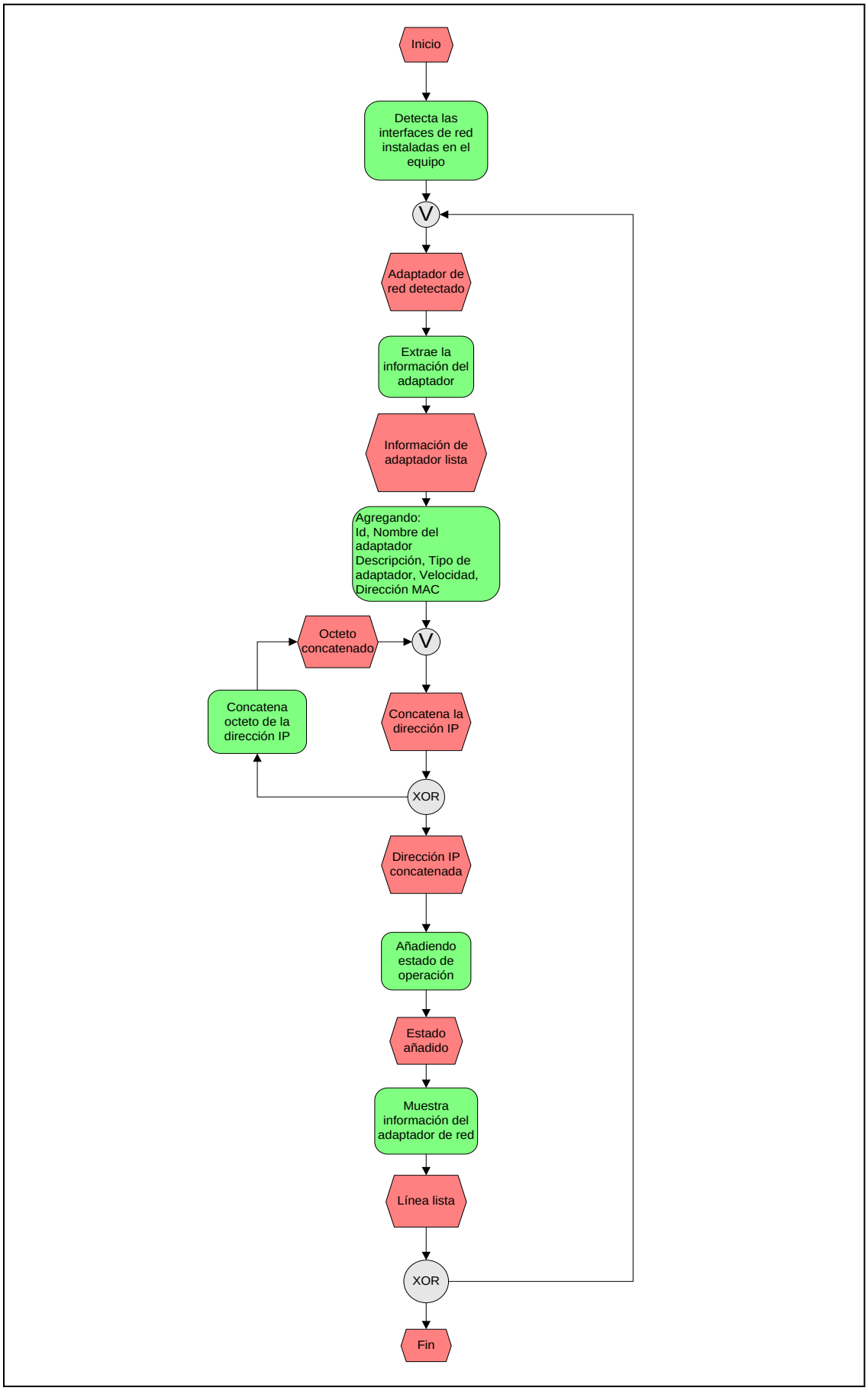


Figura.3.24 – Diagrama EPC para la obtención de información de los adaptadores de red.

Nombre:	**Algoritmo de adaptadores de red.**
Descripción:	Detecta y adquiere las características de los adaptadores de red instalados en la computadora.
Variables importantes:	Adaptador, Direcciones, Dirección

```
[Inicio]

   Detección de tarjetas de red
   Extraer información del adaptador de red
   Para i= 1,2… hasta el total de adaptadores de red
              Línea(i)←Id
              Línea(i) ←Nombre
              Línea(i) ←Description
              Línea(i) ←Tipo de interface de red
              Línea(i) ←Velocidad de la tarjeta en Kbits/s
              Direcciones← " "
              Para d = 1,2… hasta el ultimo  octeto de la dirección IP del
              adaptador de red
                     Direcciones←dirección(d)
              Línea(i) ←Direcciones
              Línea(i) ←Estado Operacional

[Fin]
```

Algoritmo 3.8 – Algoritmo que adquiere las características del adaptador de red.

3.2.2 Diseño de la bases de datos

El diseño de la base de datos se divide en dos secciones, la sección 3.2.2.a se presenta el diseño lógico de la base de datos y la sección 3.2.2.b explica el diseño físico dentro del manejador de base de datos.

3.2.2.a Diseño lógico

La sección 3.2.2.a.i describe las entidades y los diagramas entidad relación, la sección 3.2.2.a.ii presenta las expresiones del algebra relacional para las consultas de la base de datos y la sección 3.2.2.a.iii muestra las consultas de SQL del algebra relacional.

3.2.2.a.i Diagramas entidad relación.

Para el diseño de bases de datos se comienza con los diagrama entidad[56] relación[57] (E-R) [3] por que expresan gráficamente la estructura general de la base de datos y facilitan el diseño, en los diagramas E-R se utilizan tres conceptos básicos que se describen en los siguientes puntos:

1) Conjunto entidad: Es un conjunto de entidades del mismo tipo que tienen en común las mismas propiedades y atributos[58].
2) Conjunto relación: Es un conjunto de relaciones del mismo tipo.
3) Los atributos: Representan los valores permitidos de una entidad.

Los diagramas E-R utilizados para el diseño de la base de datos del tablero de control están compuestos principalmente por los siguientes componentes:

- Rectángulo: Representa un conjunto entidad.
- Elipse: Representa un atributo.
- Rombo: Representa un conjunto relación.

[56] Una entidad es una cosa u objeto del mundo real que es distinguible de todos los demás objetos [3].
[57] Una relación es una asociación entre varias entidades [3].
[58] Los atributos son propiedades descriptivas que posee cada miembro de un conjunto de entidades [3].

- Línea: Es la unión entre los atributos con la entidad y de la entidad con la relación.
- Elipse discontinua: Determina un atributo derivado[59].

En la tabla 3.16 se presenta los componentes y su símbolo usado en la notación[60] de los diagramas E-R.

Componente	*Símbolo*
Entidad	
Atributo	
Relación	
Unión	
Atributo derivado.	

Tabla.3.16 Símbolos usados en la notación E-R.

Con los componentes de la tabla 3.16 y sus símbolos se crean las estructuras de las entidades que componen la base de datos del tablero de control, los diagramas se describen en los siguientes puntos:

- Entidad encabezado.- Esta entidad almacena el identificador de encabezado, la IP origen junto con los protocolos TCP, UDP y Desconocido que obtiene el sniffer (fig3.25).

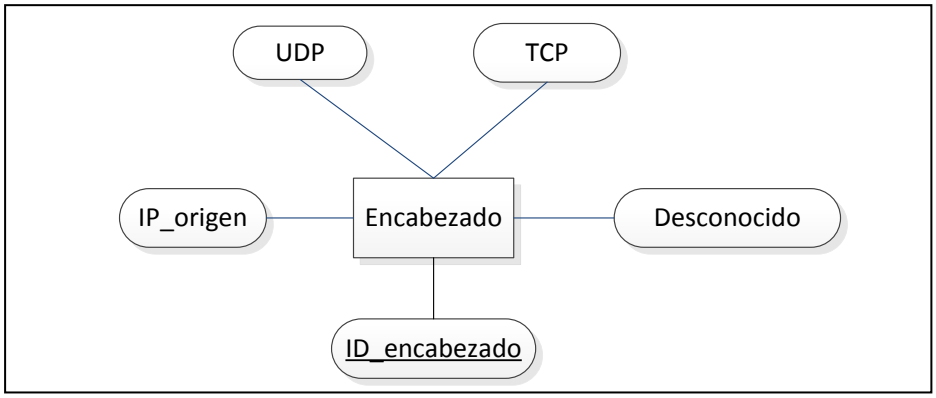


Figura.3.25 – Diagrama entidad "encabezado" con sus atributos IP_origen, TCP, UDP, Desconocido.

[59] El valor de este atributo se puede obtener a partir del valor de otro atributo [3].
[60] Sistema de símbolos que se adoptan para expresar los conceptos de los diagramas E-R.

Como se puede ver en la figura 3.25 la entidad encabezado tiene el atributo "ID_encabezado" subrayado lo que denota la clave[61] primaria de la entidad que es un valor de identificación único que permite distinguir un valor de otro [3], esto permite eliminar la redundancia[62] de información en la base de datos.

- Entidad paquetes.- Esta entidad contiene los paquetes de entrada y salida del adaptador de red, junto con los paquetes que tengan errores en el encabezado IP (fig.3.26).

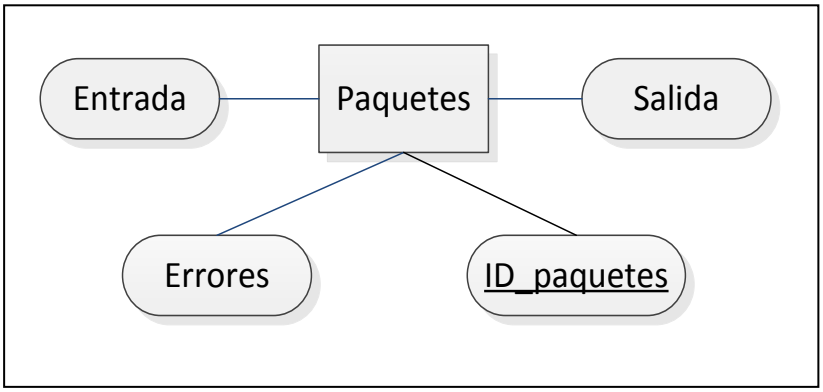


Figura.3.26 – Diagrama entidad "paquetes" con sus atributos ID_paquetes, entrada, salida y errores.

En la figura 3.26 se observa que el identificador de paquetes es la clave primaria porque se distingue de los valores que toman los otros atributos, eliminando la redundancia de información en la base de datos.

- Entidad equipo.- Esta entidad almacena la información de los equipos de cómputo que el usuario puede editar como el id_equipo, el número de Nodo, Tipo (PC, MAC, AP, Servidor, etc) y el identificador de segmento de red (fig.3.27).

[61] Las claves permiten identificar un conjunto de atributos que resulta suficiente para distinguir las entidades entre si [3].
[62] Se refiere a un mal diseño de la base de datos y presenta repeticiones en la información [3].

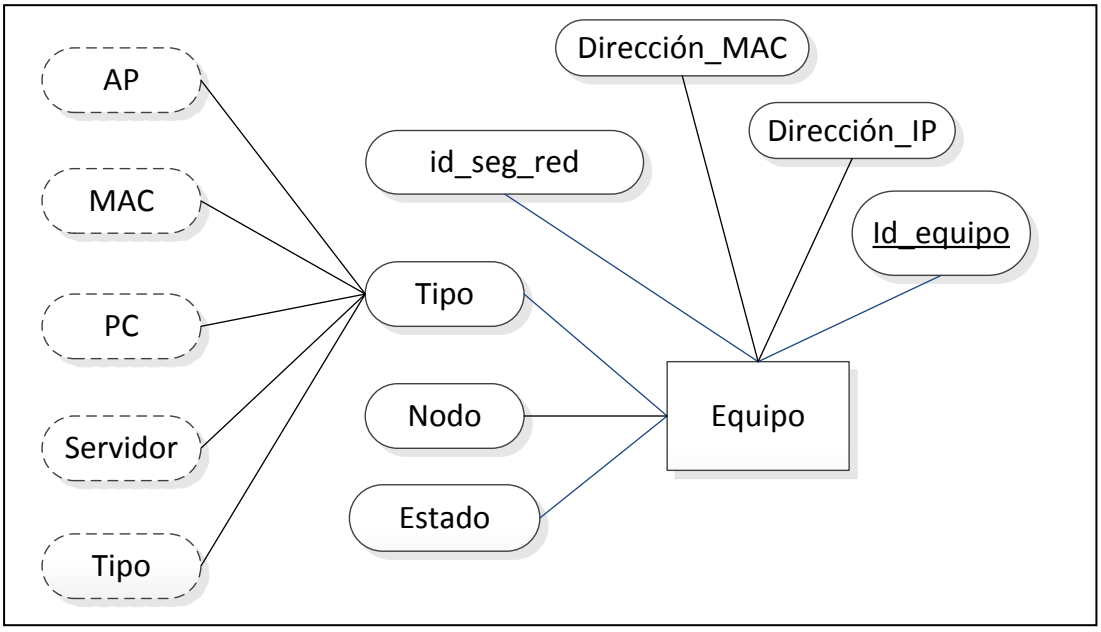


Figura.3.27 – Diagrama entidad “equipo”.

En la figura 3.27 se observan los atributos de la entidad equipo como el identificador de equipo, dirección IP, dirección MAC, identificador de segmento de red, tipo en donde se pueden ver sus atributos derivados denotados por elipses discontinuas, nodo y estado.

- Entidad área.- Esta entidad contiene información que el usuario puede editar como el identificador de segmento de red y el departamento (fig.3.28).

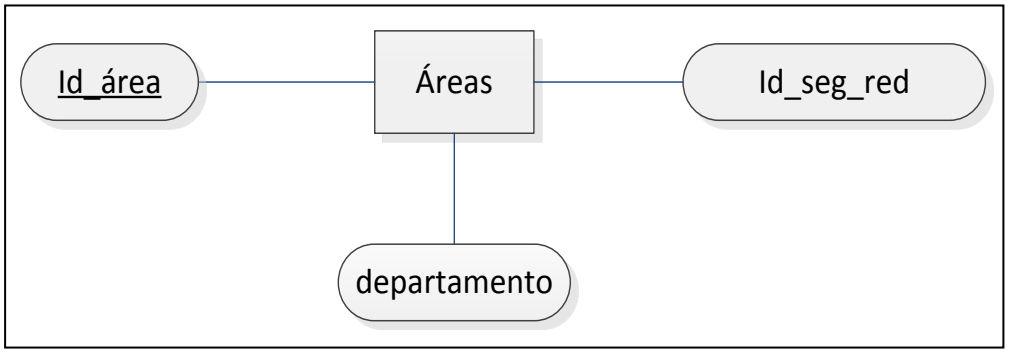


Figura.3.28 – Diagrama entidad “área” con sus atributos Id_área, Id_seg_red, departemto.

En la figura 3.28 el atributo identificador de área (Id_área) no puede ser editado por el usuario, para facilitar el diseño de la base de datos, el valor que toma este atributo se incrementa cada vez que se ingresa un nuevo

departamento e identificador de segmento de red, así se pueden distinguir fácilmente las aéreas por medio de un valor definido en vez del nombre completo de un área que complicaría el manejo de la información.

Con la descripción de cada uno de los diagramas de entidades se crea el diagrama E-R con límites de cardinalidad[63] el cual contiene tres relaciones que une a las entidades (fig.3.29), las relaciones son descritas en los siguientes puntos:

- La relación ***se encuentra,*** indica la pertenencia de un objeto del mundo real (en este caso en particular una computadora, un servidor, etc) en una ubicación física determinada (en este caso un área de trabajo).
- La relación ***envía y recibe,*** indica la acción de la tarjeta de red que transmite y recibe paquetes de datos de la red.
- La relación ***contiene,*** indica que dentro del paquete de datos se encuentran contenidos los encabezados.

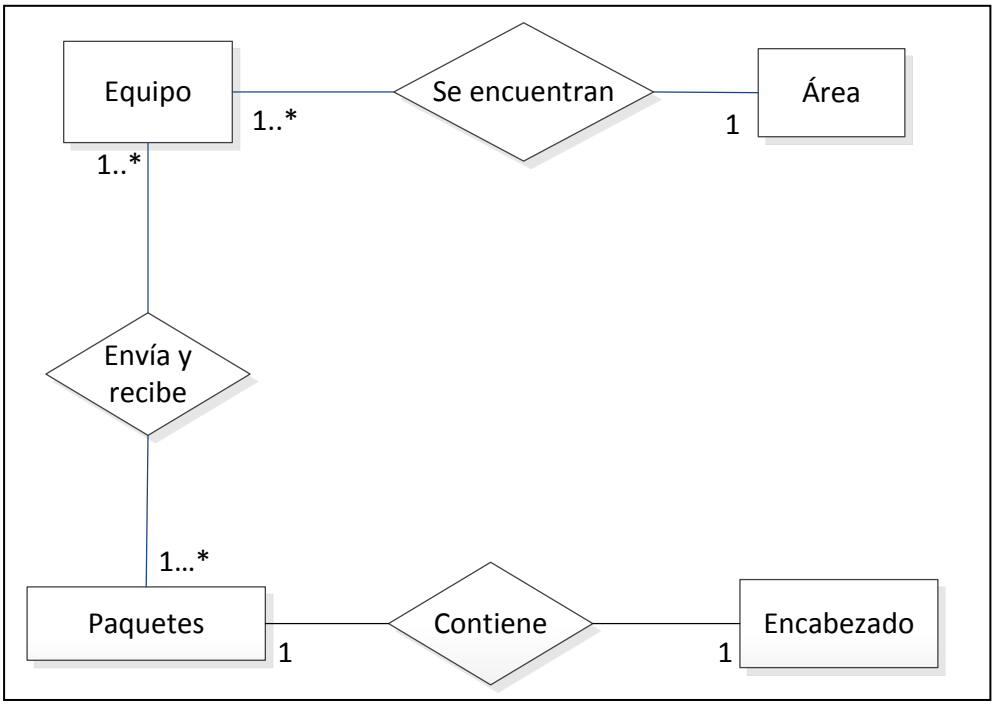


Figura.3.29 – Diagrama E-R del tablero de control.

El diagrama E-R de la figura 3.29 indica que en un área determinada se encuentra uno o mas equipos instalados, teniendo así una cardinalidad varios a uno[64], otra

[63] Es el número de participación entre las entidades asociadas en el conjunto de relaciones [3].

relación es entre las entidades equipo y paquetes en donde un equipo envía y recibe paquetes de datos de la red teniendo así una cardinalidad varios a varios[65] y por ultimo la relación en donde cada paquete contiene un encabezados y un encabezado es contenido en un paquete de datos, teniendo así una cardinalidad uno a uno[66].

Finalmente los esquemas derivados de las entidades se muestran en los siguientes puntos donde la clave primaria de cada esquema de relaciones se denota mediante un subrayado:

- Área = (Id_área, Id_seg_red, Departamento)
- Equipo = (Id_equipo, Diraccion_IP, Direccion_MAC, Id_seg_red, Nodo, Tipo, Estado)
- Paquete = (ID_paquete, Entrada, Salida, Error)
- Encabezado = (ID_encabezado, IP_origen, TCP, UDP, Desconocido)

3.2.2.a.i.1 Consultas del algebra relacional para la base de datos del tablero de control

Las consultas del algebra relacional[67] para solicitar información de la base de datos en el tablero de control son las siguientes:

1) Mostrar de los equipos que se encuentran activos en la red su identificador de equipo, dirección IP, dirección MAC, nombre de usuario, el número de nodo, el tipo y estado del equipo.

$$\prod_{ID\_equipo, Dirección\_IP, Dirección\_MAC, Nombre\_ususario, Nodo, Estado} \left(\sigma_{Estado=Activo}\left(Equipo\right)\right)$$

Ejemplos:

Supóngase que la relación *Equipo* es la que se muestra en la tabla.3.17, donde se observan los campos de ID_equipo, Direccion_IP, Direccion_MAC,Nombre_ususario, Nodo, Tipo, Estado.

---

[64] Cada entidad de A se asocia con una entidad de B [3].
[65] Cada entidad de A se asocia con cualquier número de entidades de B, y cada entidad de B se asocia con cualquier número de entidades de A [3].
[66] Cada entidad de A se asocia con uno de B, y cada entidad de B se asocia con una de A [3].
[67] Para mayor información remitirse al apéndice "C", subtitulo C.1.

ID_equipo	Direccion_IP	Direccion_MAC	Nombre_usuario	Nodo	Tipo	Estado
1	148.204.181.174	00-E1-E2-21	Jonatan Juárez	66	PC	Activo
2	148.204.181.171	01-E2-E1-22	Patricia Lopez	64	PC	Inactivo
3	148.204.181.170	02-PE-D1-23	Fernando Piñon	63	PC	Activo
4	148.204.181.169	03-CH-AM-15	Gustavo Martínez	60	PC	Inactivo
5	148.204.181.168	04-PI-PU-03	Agustín Cruz	58	PC	Activo

Tabla.3.17 - Relación *Equipo*

El proceso de los pasos de la ejecución de la consulta se observa en la figura 3.30.

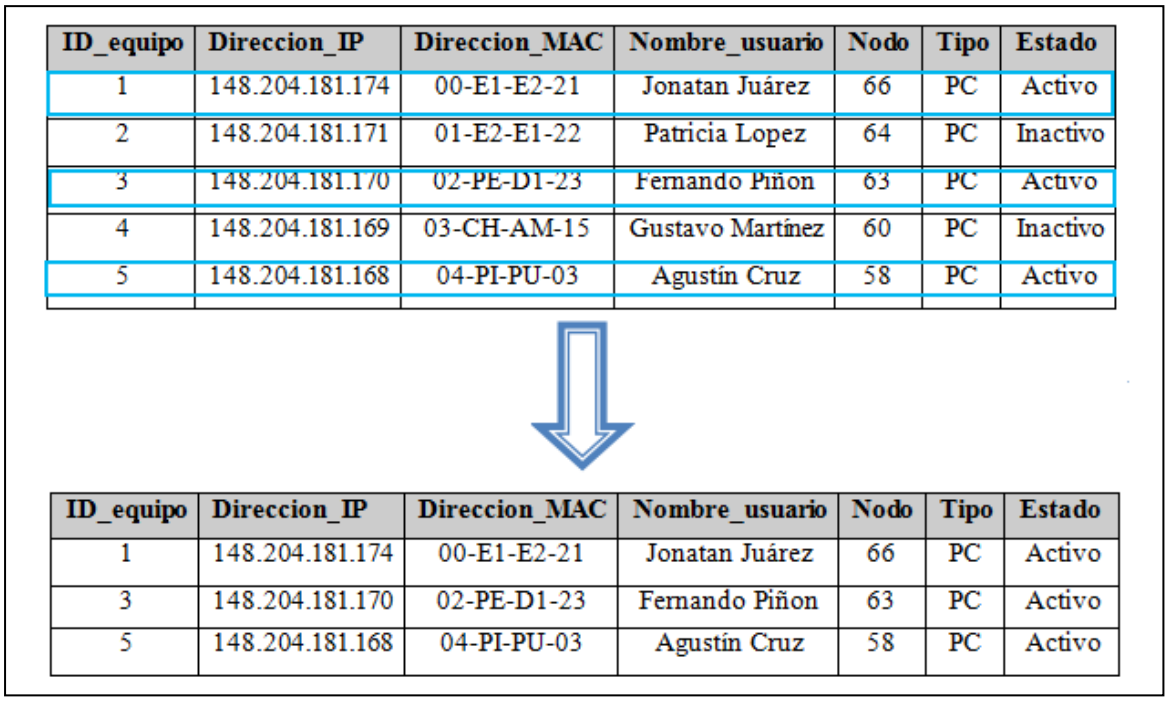


Figura 3.30 - Proceso de la expresión del algebra relacional.

En la figura 3.30 se puede observar los paso que se realizan en el algebra relacional, primero se realiza la operación selección y después la operación proyección para terminar con el resultado de la tabla 3.18.

ID_equipo	Direccion_IP	Direccion_MAC	Nombre_usuario	Nodo	Tipo	Estado
1	148.204.181.174	00-E1-E2-21	Jonatan Juárez	66	PC	Activo
3	148.204.181.170	02-PE-D1-23	Fernando Piñon	63	PC	Activo
5	148.204.181.168	04-PI-PU-03	Agustín Cruz	58	PC	Activo

Tabla3.18 – Relación de la consulta a la tabla *Equipo*.

2) Mostrar el identificador de paquetes, paquetes de entrada, salida y con errores que obtiene el adaptador de red.

$$\prod_{ID\_paquete, Entrada, Salida, Errores}(Paquetes)$$

3) Insertar dentro un nuevo nodo que contenga la dirección IP, la dirección MAC, el nombre del usuario, Nodo de conexión, tipo y estado del equipo.

$$Equipo \leftarrow Equipo \bigcup \left\{ \begin{pmatrix} Direccion\_IP, Direccion\_MAC, \\ Nombre\_usuario, Nodo, Tipo, Estado \end{pmatrix} \right\}$$

4) Borrar un registro de la tabla posgrado donde el identificador de equipo sea el número 2.

$$Posgrado \leftarrow Posgrado - \sigma_{ID\_equipo=2}(Posgrado)$$

5) Actualizar todos los campos de un registro de un equipo de posgrado donde el nombre de usuario es Jonatan Juárez Hinojosa.

$$Posgrado \leftarrow Posgrado \bigcup \prod_{\substack{ID\_equipo, Direccion\_IP, Direccíon\_MAC \\ Nodo, Tipo, Estado}} (Posgrado)$$

6) Encontrar el equipo con la dirección IP 148.204.181.168 y mostrar su identificador de equipo, dirección MAC, nombre de usuario, nodo de conexión, tipo y estado del equipo.

$$\prod_{\substack{ID\_equipo, Direccion\_MAC, Nombre\_usuario \\ Nodo, Tipo, Estado}} \left(\sigma_{Direccion\_IP=148204181168}(Equipo) \right)$$

3.2.2.a.i.2 Consultas SQL de la base de datos del tablero de control

Las consultas de la sección 3.2.2.a.i.1 del algebra relacional escritas en el lenguaje SQL[68] son las siguientes:

[68] Para mayor información remitirse al apéndice "C", subtitulo C.2.

1) Mostrar el identificador de equipo, dirección IP, dirección MAC, nombre de usuario, el número de nodo, el tipo y estado del equipo, de las computadoras activas en la red.

Select *

From *Equipo*

Where *Estado = Activo;*

2) Mostrar el identificador de paquetes, paquetes de entrada, salida y con errores que obtiene el adaptador de red.

Select *

From *Paquetes;*

3) Insertar dentro un nuevo nodo que contenga la dirección IP, la dirección MAC, el nombre del usuario, Nodo de conexión, tipo y estado del equipo.

Insert into *Euipo*

Value *(Direccion_IP,Direccion_MAC,Nombre_usuario,*

Nodo, Tipo, Estado);

4) Borrar un registro de la tabla posgrado donde el identificador de equipo sea el número 2.

Delete from *Equipo*

Where *ID_equipo = 2;*

5) Actualizar todos los campos de un equipo de posgrado donde el nombre de usuario es Jonatan Juárez Hinojosa.

Update *Posgrado*

SET *(ID_equipo, Direccion_IP,Direccion_MAC, Nodo, Tipo, Tipo, Estado)*

Where *Nombre_usuario = Jonatan Juárez Hinojosa*

6) Encontrar el equipo con la dirección IP 148.204.181.168 y mostrar su identificador de equipo, dirección MAC, nombre de usuario, nodo de conexión, tipo y estado del equipo.

Select *ID_equipo, Direccion_MAC,Nombre_usuario, Nodo, Tipo, Estado*

From *Equipo*

Where *Direccion_IP = 148.204.181.168*

3.2.2.b Diseño físico

En esta sección se muestra el diseño de las tablas dentro del manejador de base de datos (SABD); el SABD que se utiliza es MySQL Workbench por su facilidad uso, fiabilidad de almacenamiento y recuperación de datos además de poderse conseguir de forma gratuita en el sitio web del fabricante.

Para comenzar con el diseño de las tablas dentro del SABD se tiene que crear primero un esquema, esto se hace activando la opción nuevo modelo EER como se muestra en la figura 3.31

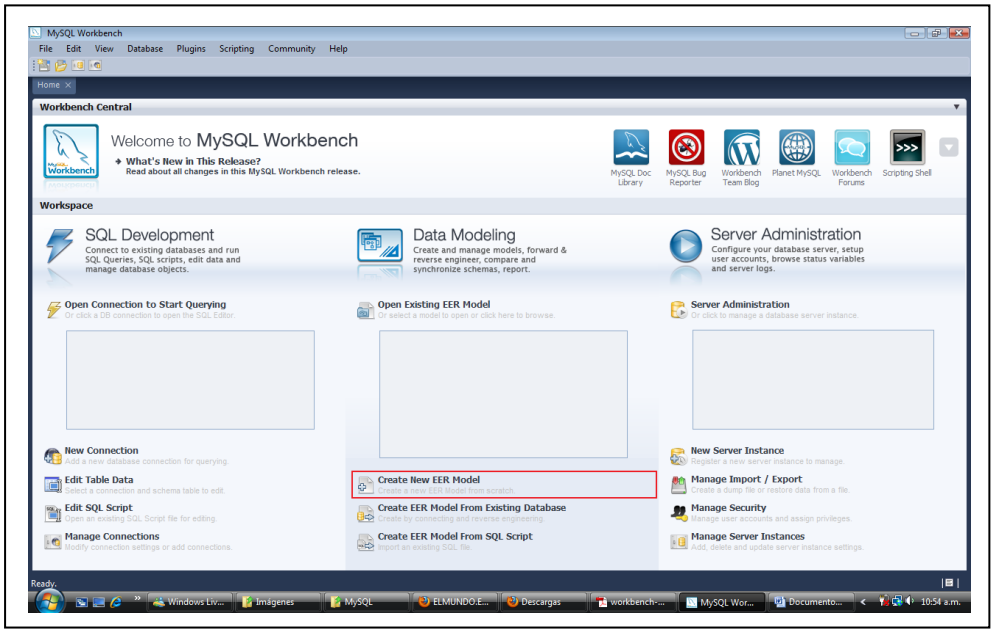


Figura 3.31 - Pantalla de inicio MySQL Workbench.

En la figura 3.31 se puede observar en un recuadro de color rojo la opción crear un nuevo modelo EER, cuando se crea un nuevo modelo este tiene un nombre predeterminado que es “mydb” como se muestra en la figura 3.32.

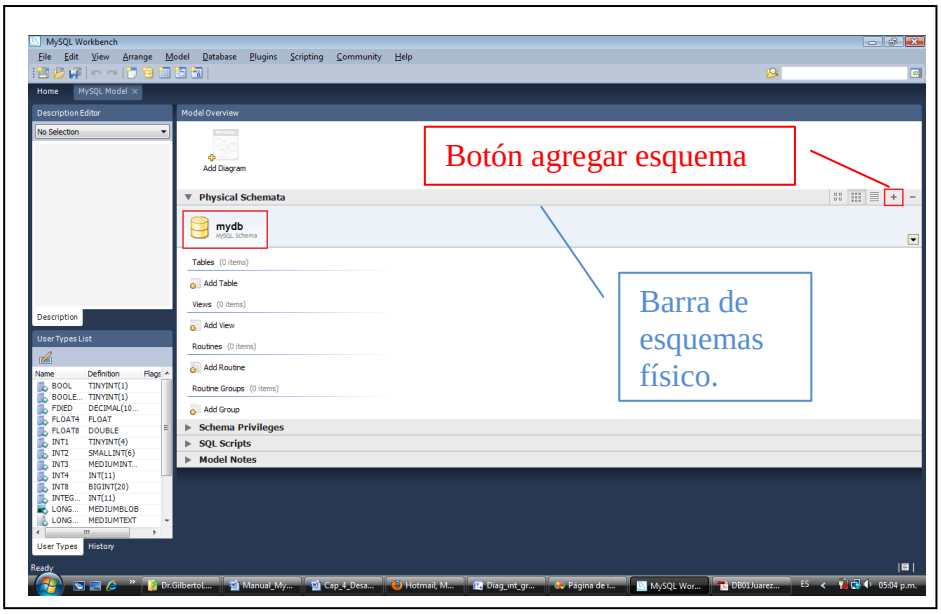


Figura 3.32 – Creación del nuevo modelo de EER

En la figura 3.32 se puede observar la barra de tareas de esquema físico en donde se activa el botón con el símbolo [+] para añadir un nuevo esquema, en donde se crea la base de datos de nombre "vb_msql" con la cual se conectan las aplicaciones del tablero de control (fig 3.33) para realizar inserciones, consultas y actualizaciones.

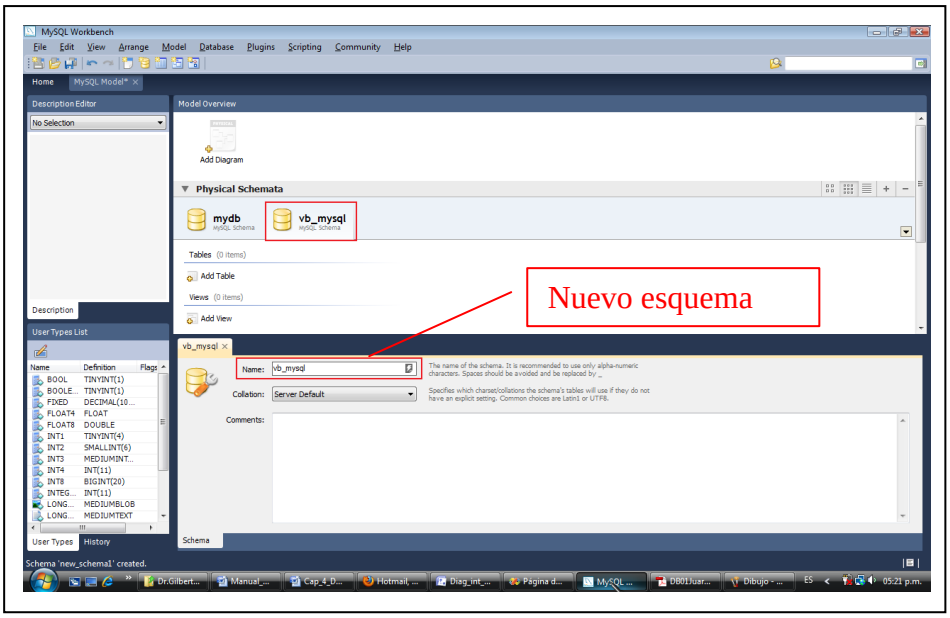


Figura 3.33 – Creación del nuevo esquema "vb_mysql".

En la figura 3.33 se observa la creación de un nuevo esquema, se puede ver como en el tabsheet (encerrado en un rectángulo de color rojo) se cambia el nombre y aparece el del esquema que se acaba de crear,

Después para realizar la edición de las tablas se activa el botón adicionar tabla como lo muestra la figura 3.34.

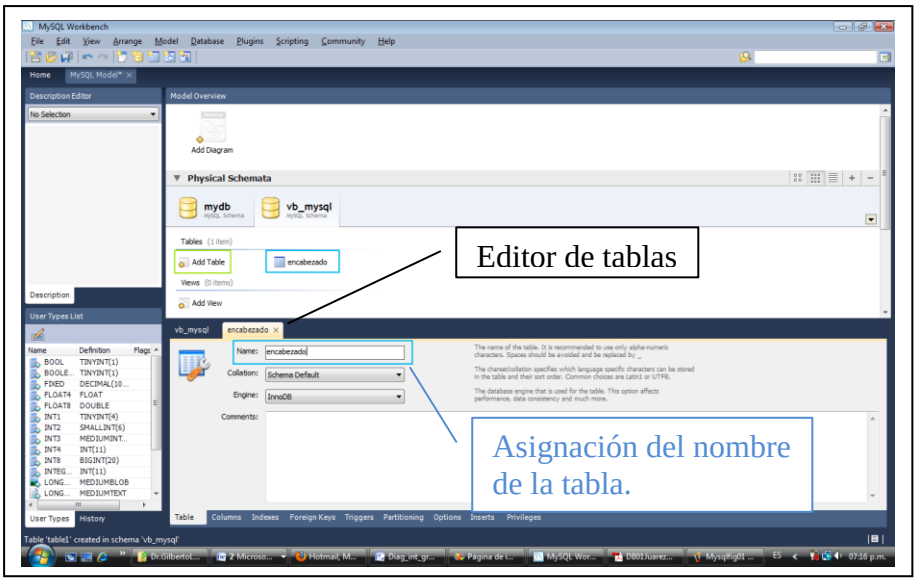


Figura 3.34 – Editor de tablas de MySQL Workbench.

En la figura 3.34 se puede observar dentro de un recuadro de color verde el botón adicionar tabla, cuando se activa este botón se ingresa al editor de tabla en donde colocamos un nombre de tabla (recuadro de color azul); en la pestaña columnas se edita cada uno de los campos o columnas que contendrá la tabla, en este caso los campos que componen a la tabla encabezado son: ID_encabezado, IP_origen, TCP, UDP y Desconocido como se muestra en la figura 3.35.

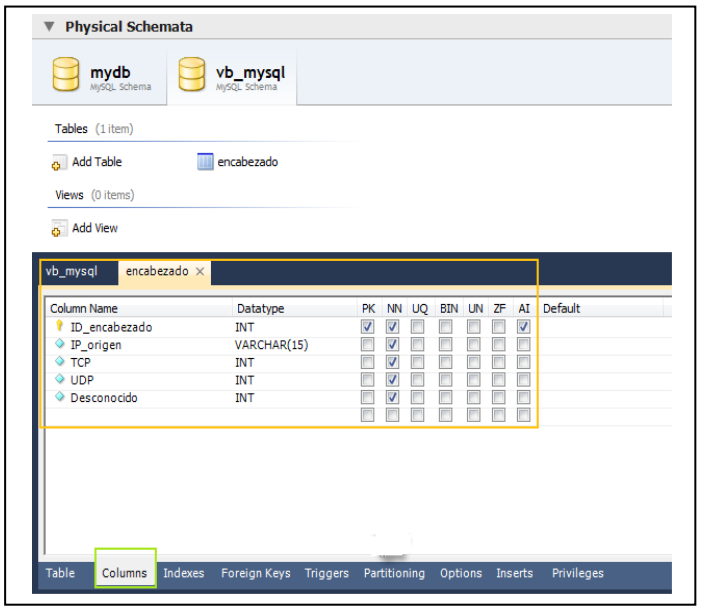


Figura 3.35 – Edición de tabla encabezado.

En la figura 3.35 se observa la pestaña columnas enmarcada en un recuadro de color verde en donde se realiza la edición de las columnas de la tabla de encabezado, en un recuadro de color amarillo se encuentran los nombres de las columnas junto con sus propiedades, una vez terminada la edición se activa el botón guardar de la barra de herramientas del SABD para guardar la base de datos.

Después es necesario crear una conexión a un servidor (en este caso local) porque Visual Basic requiere de este parámetro para conectarse con MySQL Workbench y así insertar, actualizar y consultar de datos dentro de la base de datos. La conexión al servidor se realiza activando la función administrador de conexiones del menú base de datos como lo muestra la figura 3.36.

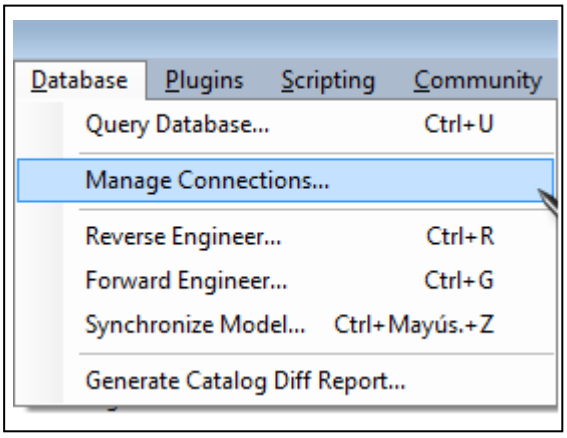


Figura 3.36 – Activación del administrador de conexiones.

En la figura 3.36 se puede ver la ubicación de la función administrador de conexiones, cuando se activa se observa un cuadro de dialogo igual al de la figura 3.37.

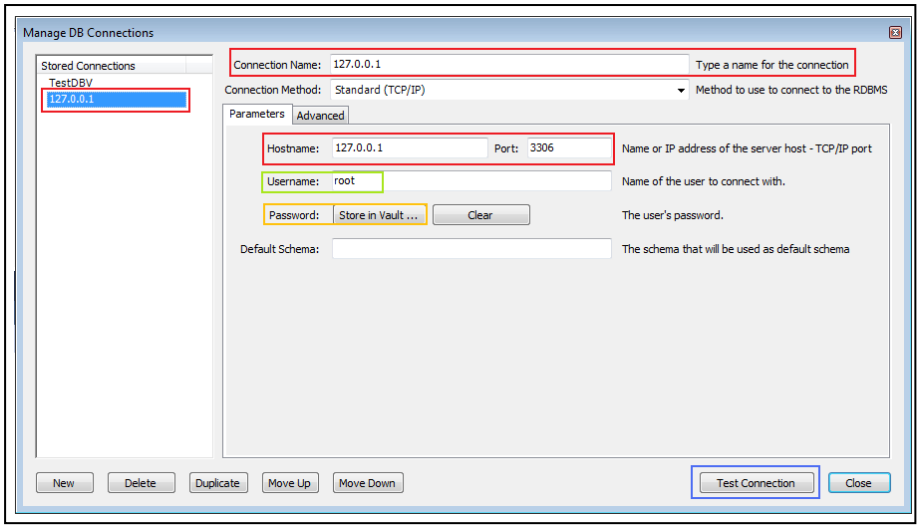


Figura 3.37 – Cuadro de dialogo administrador de conexiones.

En la figura 3.37 se puede observar encerrados en recuadros de color rojo la dirección[69] y el puerto por donde se conecta el servidor, en un recuadro de color verde se encuentra el nombre de usuario que en este caso es root[70], encerrado en un recuadro de color amarillo se encuentra la contraseña la cual se tiene que colocar para tener acceso a la base de datos y finalmente encerrado en un botón de color azul se encuentra el botón prueba de conexión que al activarse envía un mensaje de conexión satisfactoria o fallo en la conexión.

Una vez que se a establecido satisfactoriamente la conexión de la base de datos al servidor se activa la función avance de ingeniería del menú base de datos como lo muestra la figura 3.38.

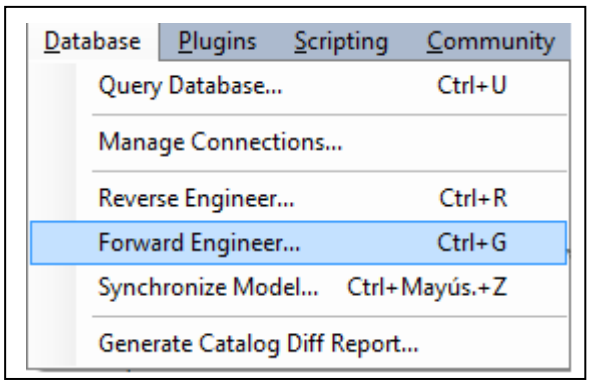


Figura 3.38 – Activación de la función avance de ingeniería.

En la figura 3.38 se observa la ubicación de la función avance de ingeniería, cuando se activa esta función se muestra una primera página del asistente que contiene el conjunto de opciones para la creación de la base de datos, no se selecciona ninguna opción porque se requiere activar ninguna opción, se activa el botón siguiente para continuar.

En la siguiente página se selecciona el objeto que se desea exportar al servidor activo. En este caso sólo tenemos la tabla encabezado, entonces ningún otro objeto tiene que ser seleccionado, se activa el botón siguiente para continuar.

[69] MySQL Workbench toma la dirección del equipo local que es 127.0.0.1 ya que la base de datos se encuentra almacenada dentro del disco duro local de la computadora.
[70] En MySQL Workbench root es el usuario con totalidad de privilegios dentro del SABD.

En la siguiente pantalla, se revisa la sintaxis de SQL que muestra la escritura sobre el servidor activo para crear el esquema. Se examina que la sentencia de SQL correspondan a las configuraciones que se han realizado (fig 3.39) y se activa el botón el botón siguiente para continuar.

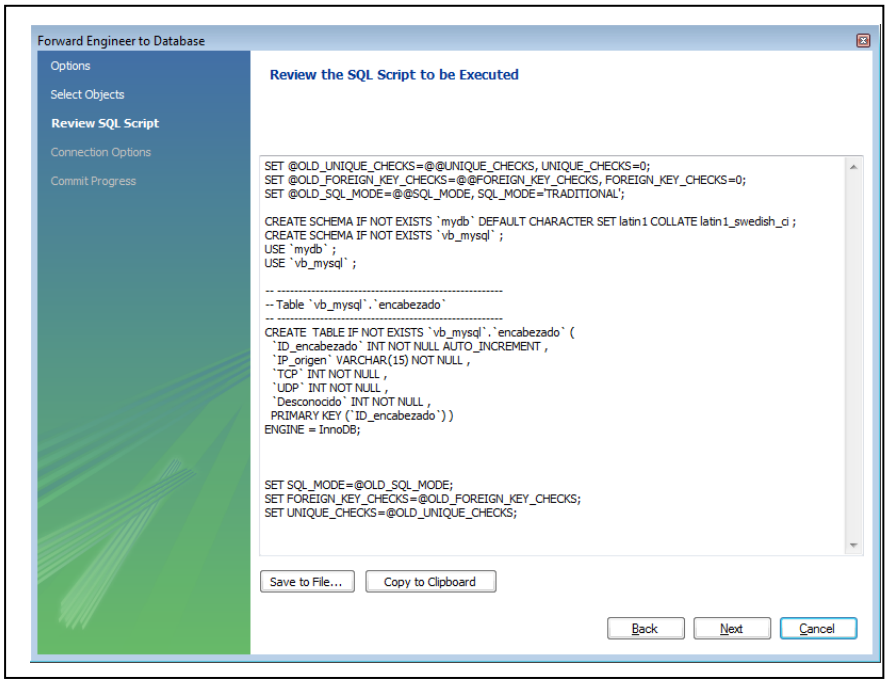


Figura 3.39 Vista de las sentencias de SQL para la creación del esquema y la tabla encabezado dentro de la base de datos.

En la figura 3.39 se observa las sentencias de SQL propias del manejador de la base de datos para la creación del esquema y la tabla dentro de la base de datos, se activa el botón siguiente.

En la siguiente ventana se selecciona la conexión al servidor creado anteriormente (en este caso es el 172.0.0.1) y se activa el botón ejecutar para ejecutar las instrucciones de SQL, una vez terminada la ejecución se muestra la ventana de ejecución exitosa y se activa el botón cerrar, con esto ya se tiene creada la base de datos para el tablero de control.

Para crear mas tablas dentro del mismo esquema se tiene que activar el botón adicionar tabla para volver a acceder al editor de tabla la figura 3.40 muestra la edición de las tablas que componen a la base de datos junto con la definición de sus características.

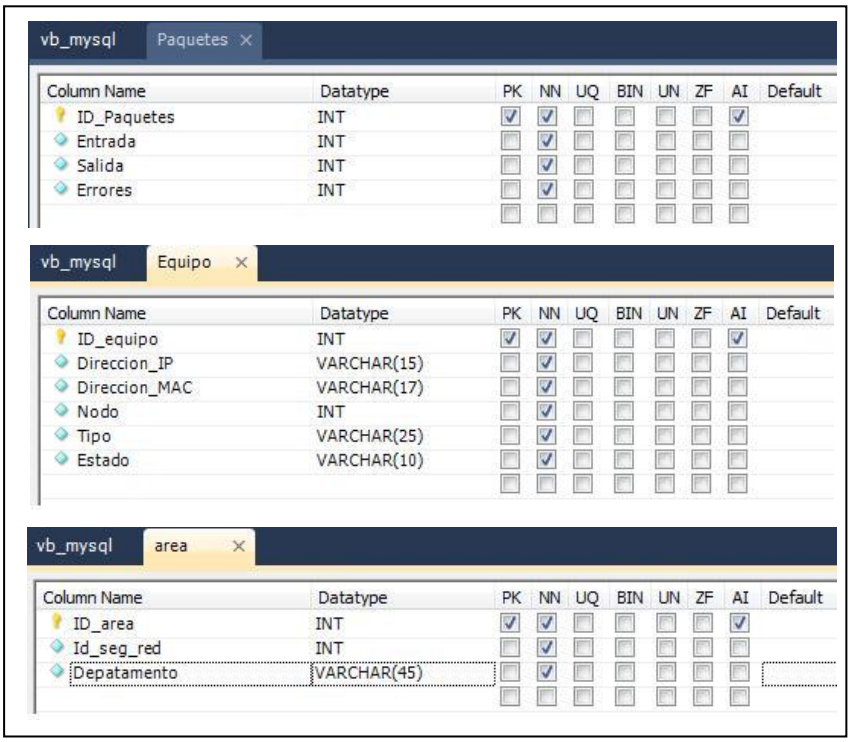


Figura 3.40 Edición de las tablas paquetes, equipo y área.

En la figura 3.40 se muestra la edición de las tablas paquete, equipo y área, junto con sus propiedades, las cuales contienen la estructura planteada en los diagramas entidad relación de la sección 3.2.2.a.i, una vez que se termina de editar cada una de las tablas se activa el comando de adelanto de ingeniería del menú base de datos para que se ejecuten las sentencias de SQL y se creen las tablas dentro del esquemas en la base de datos como lo muestra la figura 3.41.

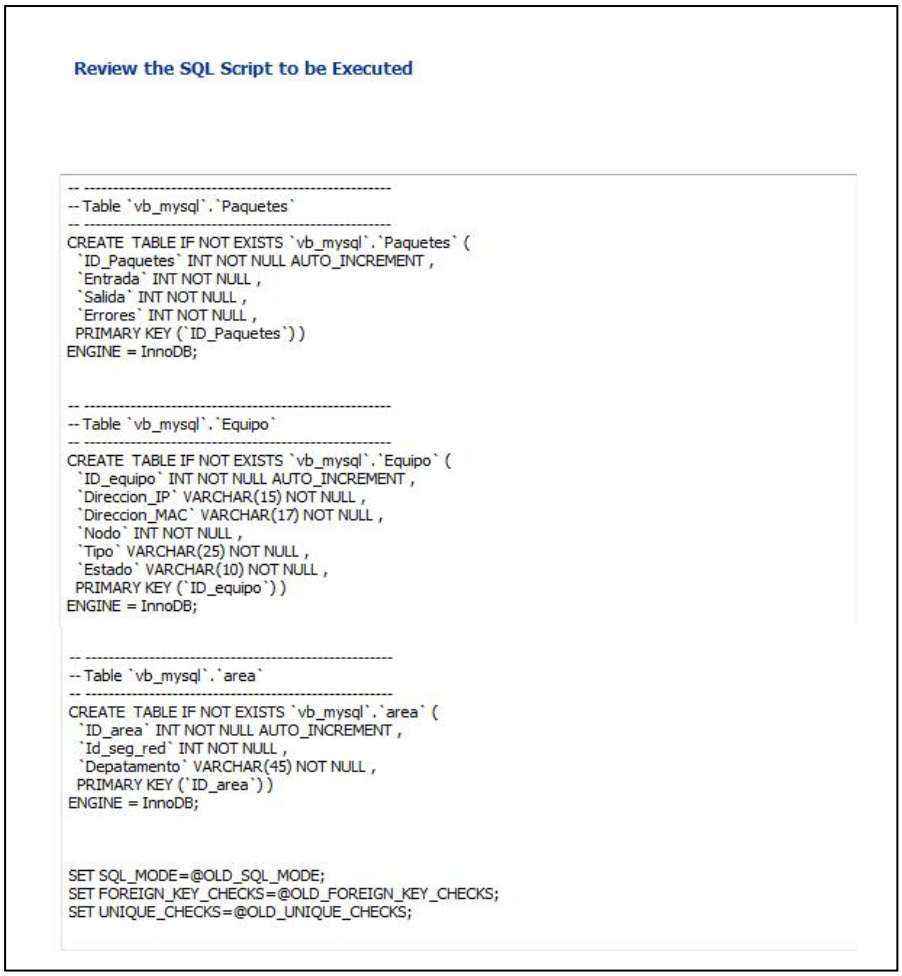


Figura 3.41 – Creación de las tablas paquetes, equipo y área dentro del SABD.

En la figura 3.41 se muestra la creación de las tablas paquetes, equipos y área dentro del sistema de administración de base de datos de los diagramas entidad relación de la sección 3.2.2.a.i, después de la creación de las tablas se genera el diagrama entidad relación dentro del SABD activando el botón adicionar diagrama (fig 3.42) permitiendo crear el diagrama E-R del tablero de control como lo muestra la figura 3.43

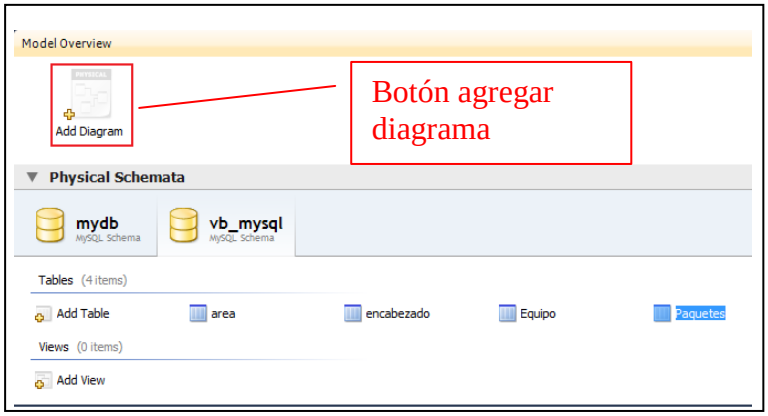


Figura 3.42 – Botón agregar diagrama.

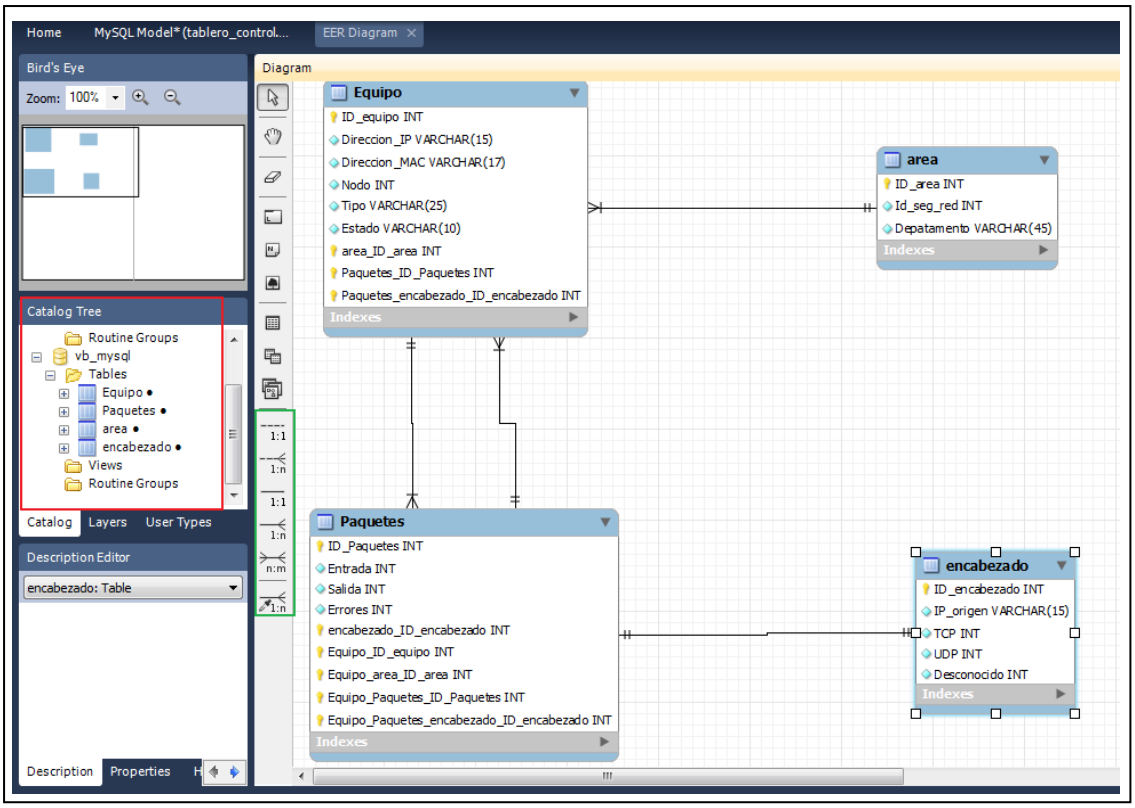


Figura 3.43 – Diagrama E-R creado en el SABD.

En la figura 3.43 se encuentran marcados con un recuadro de color rojo las tablas contenidas dentro del esquema de la base de datos “vb_mysql”, para generar el diagrama solo se tiene que enviar la tabla deseada al escritorio, cuando se tienen las tablas necesarias se utilizan los conectores (encerrados en un recuadro de color verde) para relacionar las tablas y obtener el diagrama entidad relación de la sección 3.2.2.a.i.

3.2.3 Interfaz gráfica

La interfaz grafica se divide en tres módulos como lo indica la arquitectura de la sección 3.1, un módulo de controles de análisis de red (sección 3.2.3.1), un módulo de medidores (sección 3.2.3.2) y un módulo de controles de usuario.

3.2.3.a Controles de análisis de red.

Los controles de análisis de red presentan información concreta de los paquetes de datos que viajan por la red, la sección 3.2.3.a.i presenta la descripción del husmeador de paquetes, la sección 3.2.3.a.ii explica el funcionamiento del contador de paquetes, la sección 3.2.3.a.iii presenta las estadísticas, la sección 3.2.3.a.iv explica la comprobación de conexión y la sección 3.2.3.a.v presenta las interfaces de red. La figura 3.44 presenta el diagrama a bloques de esta sección.

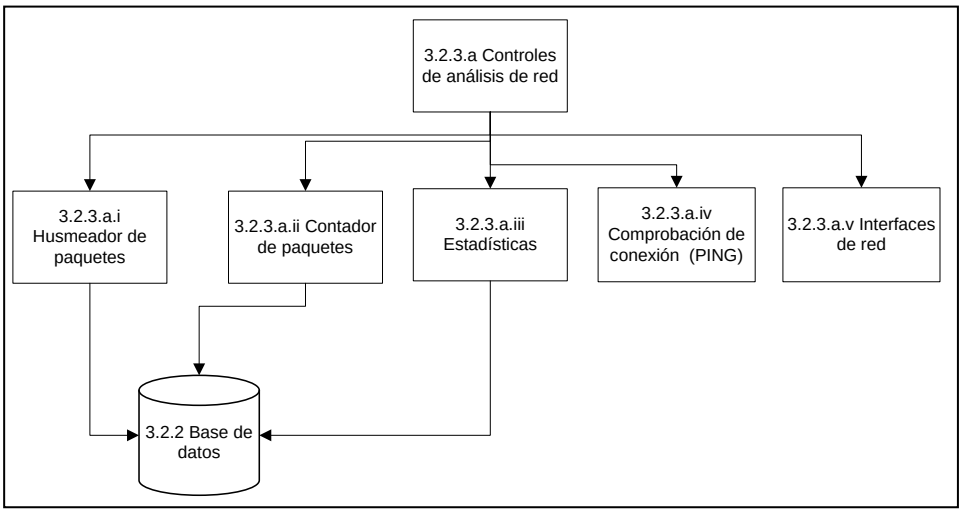


Figura.3.44 – Diagrama a bloques de los controles de análisis de red.

3.2.3.a.i Husmeador de paquetes.

El husmeador de paquetes o *sniffer* presenta la información que circula por la red, sin interferir en el proceso de envió y recepción de datos entre el transmisor y receptor, la tabla 3.19 presenta la función despliegue de información del husmeador de paquetes.

Función:	Despliegue de información del husmeador de paquetes.
Entrada:	Paquetes de datos que circulan por la red.
Salida:	Dirección IP origen, dirección IP destino, información del encabezado de IP e información de los protocolos contenidos en el como TCP, UDP o desconocido.

Descripción:

Al seleccionar un adaptador de red y activar el botón inicio en la interfaz gráfica, el husmeador de paquetes realiza los siguientes puntos:

1) Coloca a la tarjeta de red en modo promiscuo para la recepción de paquetes de datos de la red.
2) Comienza la recepción de los datos.
3) Obtiene la dirección origen, destino y protocolos del paquete de IP, junto con sus características mencionadas en los siguientes puntos:
   a) Versión de IP, Longitud de encabezado, Diferencia de servicios., Longitud total del encabezado, Identificador, Fragmentación de compensación, Banderas, Tiempo de vida del paquete, Tipo de protocolo que contiene el paquete de IP como TCP,UDP o desconocido, las características del protocolo TCP son las siguientes:
      i. Puerto origen, Puerto destino.
      ii. Número de secuencia, Longitud del encabezado.
      iii. Banderas, Suma de comprobación.

      Las características del protocolo UDP son las siguientes:
      i. Puesto origen.
      ii. Puerto destino.
      iii. Longitud, Suma de comprobación.

      Si alguno de los protocolos TCP, UDP utiliza el puerto 53 se obtiene el DNS con las siguientes características:
      i. Identificador, Banderas, Consultas.
      ii. Respuestas RRS, Autoridad RRS, Adicional RRS.
   b) IP origen,IP destino.
4) Analiza y despliega la información[71] de la información que circula por la red en forma de árbol en el entorno gráfico, el tipo de protocolo contenido por el paquete de IP
5) Conexión a la base de datos.
6) Se envía la IP origen y el tipo de protocolo encapsulado en el encabezado de IP a la base de datos.
7) Al activar el botón alto se detiene la captura de datos de red, permaneciendo los valores capturados en pantalla y almacenados en la base de datos.

Tabla.3.19 – Función Husmeador de paquetes del entorno gráfico.

[71] La información que se muestra al usuario se describió en la sección 3.2.1 Captura de tráfico de red.

3.2.3.a.ii Contador de paquetes.

Esta función presenta el número de paquetes que utilizan protocolos TCP, UDP o desconocidos encapsulados en los paquetes de datos de IP, la tabla 3.20 contiene la descripción de la función despliegue de información de los contadores de paquetes.

Función:	Despliegue de información de los contadores de paquetes.
Entrada:	Paquetes de datos de IP.
Salida:	El conteo de los paquetes que utilicen protocolo TCP, UDP o desconocido
Descripción: Al seleccionar un adaptador de red y activar el botón inicio en la interfaz gráfica, el contador de paquetes realiza los siguientes puntos: 1) Si el paquete de datos de IP encapsula un protocolo TCP, UDP o desconocido se incrementa en una unidad el contador de dicho protocolo. 2) El contador de paquetes toma los valores de los contadores y los actualiza en la base de datos. 3) El contador de paquetes toma los valores de los contadores y actualiza la interfaz gráfica. 4) Mientras no se active el botón de alto el contador actualiza los valores en la base de datos y en la interfaz gráfica, cuando se activa el botón de alto los datos ya se encuentran almacenados en la base de datos y los últimos valores actualizados se mantienen activos en la interfaz gráfica.	

Tabla.3.20 – Función contador de paquetes del tablero de control.

3.2.3.a.iii Estadísticas.

Las estadísticas son los números de paquetes de entrada, salida y errores que obtiene el adaptador de red, en la tabla 3.21 se describe esta función.

Función:	Despliegue de información de estadísticas.
Entrada:	Paquetes de datos de la red.
Salida:	Número de paquetes de datos de entrada, salida y con errores.

Descripción:

1) Se activa la función del menú herramientas.
2) Se presenta la aplicación indicando el número de paquetes de entrada, salida y con errores que se están obteniendo de la red en ese momento.
3) Conexión a la base de datos.
4) Envió de información estadística a la base de datos para su almacenamiento.
5) Despliega el número de paquetes de entrada, salida y con errores en los medidores en el entorno gráfico.

Tabla 3.21 – Función Estadísticas del entorno gráfico.

3.2.3.a.iv Comprobación de conexión (PING).

La aplicación "hacer ping" sirve para comprobar la conexión con algún equipo o para verificar que el equipo tiene conexión a internet, esta función aparece en pantalla cuando es activada desde el menú herramientas, la descripción de esta función se encuentra en la tabla 3.22.

Función:	Despliegue de información de Hacer PING
Entrada:	Nombre o dirección IP del equipo remoto, tiempo de vida, numero de eco, tamaño del buffer.
Salida:	Despliegue de nombre equipo remoto, dirección IP, estado de la solicitud de conexión, tiempo de vida y tamaño del buffer.

Descripción:

1) Activar la aplicación desde el menú herramientas
2) Escribir el nombre o dirección del equipo remoto donde se desea comprobar la conexión en la sección "Introduzca nombre de equipo",
3) Si se desea realizar modificaciones en los siguientes controles:
   a) Numero de eco.
   b) Tamaño del buffer.
   c) Tiempo de vida.

   Active los botones con flechas para incrementar o disminuir el valor del parámetro deseado.
4) Active el botón comprobar
5) La respuesta del equipo remoto aparecerá en pantalla como lo indican los siguientes puntos:
   a) La respuesta del equipo es exitosa.
   b) No hay repuesta del equipo remoto.
   c) Se termino el tiempo de espera del equipo remoto.

Tabla 3.22 - Función Hacer PING del entorno gráfico.

3.2.3.a.v Interfaces de red.

La aplicación interfaces de red se presenta en pantalla una vez que es activada desde el menú herramientas, la descripción de de esta función se encuentra en la tabla 3.23.

Función:	Despliegue de información de interfaces de red.
Entrada:	Características del adaptador de red.
Salida:	Despliegue de las características en la aplicación.
Descripción: Esta aplicación solo presenta las características de los adaptadores de red instalados en el equipo de computo, el usuario no puede modificar ninguna parámetro por que no es el propósito de esta aplicación alterar la configuración de fabrica del adaptador de red, las características se enumeran en los siguientes puntos: 1) Identificador (id): Muestra el número de identificación del adaptador de red. 2) Nombre: Contiene el nombre del adaptador de red. 3) Descripción: Contiene la información definida por el fabricante del adaptador de red. 4) Tipo: Indica el tipo de adaptador de red (Ethernet, Wi-fi, etc.) 5) Dirección física (MAC): Contiene la dirección física del adaptador de red. 6) IP: contiene la dirección IP del adaptador de red. 7) Estado: Detecta el estado activo o inactivo del adaptador de red.	

Tabla 3.23 – Función adaptadores de red del entorno gráfico.

3.2.3.b Medidores.

Los medidores presentan de forma gráfica la información de la sección 3.2.3.b.i que contiene el medidor del consumo de ancho de banda y de la sección 3.2.3.b.ii que presenta el medidor de tráfico generado por los equipos activos en la red. La figura 3.45 presenta el diagrama a bloques de la organización de este modulo.

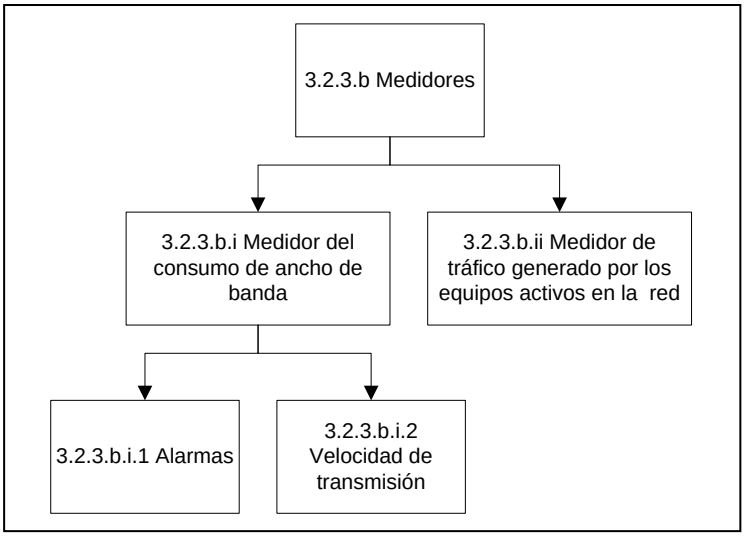


Figura 3.45 – Diagrama a bloques de la organización de los medidores del tablero de control.

3.2.3.b.i Medidor del consumo de ancho de banda.

El medidor del consumo de ancho de banda indica gráficamente el consumo porcentual de la capacidad del canal del medio de transmisión mediante una barra de progreso (fig 3.46), los pasos para determinar los valores de la barra de progreso son los siguientes:

1) Para el calculo de la capacidad del canal del medio de transmisión se utiliza la ecuación de Claude Sannon que expresa la relación entre el ancho de banda (medido en Hz) con la velocidad de transmisión (medida en bps[72]) su ecuación es la siguiente:

[72] bps.- Bits por segundo

$$C = \omega \log_2 (1 + SNR)$$

Donde:

SNR: es la relación señal a ruido con un valor de 10,000 lo que equivale a 40dB.

ω: es la frecuencia de transmisión.

$$\log_2 = \frac{\log_{10} X}{\log_2 2} = \frac{\log_{10} X}{0.3010}$$

Para un cable UTP categoría 5e el estándar de cableado para telecomunicaciones en edificios comerciales (EIA/TIA 568-B.2[73]) designa una frecuencia de transmisión de 100Mhz, por lo tanto el cálculo de la capacidad del canal del medio de transmisión es el siguiente:

$$C = 100MHz \log_2 (1 + 10000)$$
$$C = 100MHz \log_2 (10001)$$
$$C = \frac{100 \cdot 10^6 Hz \log_{10}(10001)}{0.3010}$$
$$C = 1.3Gbps$$

2) El calculo de la velocidad de transmisión máxima es el número máximo de paquetes por segundo que cruzan por la red, para obtener la cantidad porcentual del consumo del ancho de banda del canal de transmisión se utilizan cinco variables que son las siguientes:

   a. N: Es el número de computadoras [49].
   b. R: Es la taza de transmisión con un valor contante de 10Mbps [49].
   c. d: Es la distancia del BUS [49].
   d. V: Es la velocidad de propagación del medio con un valor de $2 \cdot 10^8 \, {}^{m}\!/_{s}$ [49].
   e. L: Es el tamaño del paquete de datos [49].

[73] Establece el estándar para los componentes del cableado estructurado de par trensado.

Para calcular la velocidad de transmisión se toman los valores de los diferentes tamaños de paquetes de datos y seles asocia un valor porcentual como se muestra en la tabla 3.24.

Porcentaje (%)	Tamaño del paquete (Bytes)
0	0
20	64
40	128
60	215
80	512
100	1024

Tabla 3.24 - Relación del valor porcentual de la barra de progreso con el tamaño del paquete de datos.

La formula matemática empleada para definir la transmisión con respecto a un ciclo que consta de un intervalo de transmisión y uno de contención matemáticamente se expresa con la siguiente ecuación [49]:

$$U = \frac{I_T}{I_T + I_C} = \frac{1/2a}{1/2a + (1-A)/A} = \frac{1/2a}{1 + 2a(1-A)/A}$$

Donde:

$$A = \left(1 - \frac{1}{N}\right)^{N-1}$$

$$a = \frac{R \cdot d}{V \cdot L}$$

Es importante mencionar que al tamaño de paquete de datos se le suman 24 Bytes que representan 8 Bytes de preámbulo, 4 Bytes de chequeo de errores y 12 Bytes que representan el espacio entre paquetes para calcular la cantidad máxima de paquetes que se pueden enviar por segundo [49].

Ejemplo:

Determinar la capacidad del canal de transmisión de una red que cuenta con las siguientes características:

- 16 equipos de computo
- Distancia del BUS de 200m
- Un tamaño de paquete de 128 Bytes

Solución:

Datos:

$N = 16$

$R = 10Mbps$

$d = 200m$

$L = 128Bytes$

Calculo:

$$\frac{(128+24)(8bits)}{10000000\ {}^{bits}/_{s}} = 121.6\mu s$$

$$\text{Número de paquetes por segundo} = \frac{1paquete}{121.6\mu s} = 8223.68\ {}^{paquetes}/_{seg}$$

$$a = \frac{R \cdot d}{V \cdot L} = \frac{(10000000\ {}^{bits}/_{s})(200m)}{(2 \cdot 10^{8}\ {}^{m}/_{s})[(128)(8bits)]} = 9.765623 \cdot 10^{-3}$$

$$A = \left(1 - \frac{1}{16}\right)^{15} = 0.379812$$

$$U = \frac{1}{1 + \frac{2a(1-A)}{A}} = \frac{1}{1 + \frac{2(9.765623 \cdot 10^{-3})(1-0.379812)}{0.379812}} = \frac{1}{1.0318922}$$

$$U = 0.969094$$

$$(8223.6842\ {}^{paquetes}/_{seg})(0.969094) = 7969.52\ {}^{paquetes}/_{seg}$$

En la sección 3.2.3.a.i.2 se realizan los cálculos para determinar las diferentes velocidades de transmisión, tomando la relación del valor porcentual con el tamaño del paquete de datos de la tabla 3.24, en la tabla 3.25 se describe la función medidores.

Función:	Medidores
Entrada:	Longitud total del paquete de datos de los equipos activos en la red
Salida:	Incremento o decremento en el valor de las barras de progreso, dependiendo del tamaño del paquete de datos y del numero de equipos activos.

Descripción:

Los medidores indican de forma grafica la cantidad porcentual del uso del ancho de banda, también muestran el tráfico generado por los equipos activos en ese momento en la red, debajo de cada uno de los medidores se despliega la velocidad de transmisión y la cantidad de equipos conectados, la figura 3.46 muestra la relación entre el porcentaje del consumo del ancho de banda con el tamaño del paquete de datos.

Porcentaje (%)	**0**	**20**	**40**	**60**	**80**	**100**
Paquetes (Bytes)	0	64	128	256	512	1024

Figura 3.46 Barra de progreso utilizada para medir la capacidad del canal.

Tabla 3.25Función medidores.

3.2.3.a.i.1 Alarmas.

Las alarmas son eventos con comportamiento inusual como un bajo rendimiento en los procesos de carga y descarga de información, procesos de envío de información inconclusos, etc. Las alarmas más comunes que se reportan son cuando el estado operacional de un dispositivo o servicio cambia.

Existen tipos de alarmas basados en patrones previamente definidos por el usuario con valores máximos y mínimos, cuando estos patrones no se encuentran dentro de los parámetros definidos se considera como un comportamiento fuera de los valores normales de operación, la función de este módulo se describe en la tabla 3.26.

Función:	Alarmas
Entrada:	Parámetros definidos por el usuario
Salida:	Sonido de alerta, indicación gráfica de alerta.
Descripción: El envió de una alarma es a causa de un evento inusual en la red por el incremento o disminución en los valores definidos por el usuario, cuando se definen los valores máximos y mínimos permitidos se emite un sonido de prevención para indicar que se han sobrepasado estos valores, además el color de la barra de progreso del medidor del uso del ancho de banda cambia para mostrar visualmente que existe actividad inusual en la red.	

Tabla 3.26 - Función alarmas.

3.2.3.a.i.2 Velocidad de transmisión

La velocidad de transmisión es el tiempo que tarda en llegar un paquete de datos del equipo origen al destino, en la sección 3.2.3.b.i se planteo la expresión matemática para calcular la velocidad de transmisión, en esta sección se realizan los cálculos de velocidad utilizando el valor del tamaño de paquete de datos (tabla 3.24), los cálculos se presentan en los siguientes puntos:

1) El cálculo de velocidad de transmisión para un tamaño de paquete de datos de 64 bytes con un número aproximado de equipos de 40 y una distancia de BUS aproximada de 30 metros[74] el cálculo es el siguiente:

$N = 40$

$R = 10Mbps$

$d = 30m$

$V = 2 \cdot 10^8 \frac{m}{s}$

$L = 64bytes$

[74] Las aproximaciones del número de equipos de computo como la distancia del BUS se obtuvieron de mediciones físicas en el edificio de posgrado del CIITEC.

$$\frac{(64+24)(8bits)}{10000000\,{}^{bits}/_{s}} = 70.4\mu s$$

El número de paquetes por segundo $= \frac{1paquete}{70.4\mu s} = 14204.54\ \frac{paquetes}{seg}$

$$a = \frac{R \cdot d}{V \cdot L} = \frac{(10000000bps)(30m)}{(2 \cdot 10^{8}\,{}^{m}/_{s})[(64)(8bits)]} = 1.953125 \cdot 10^{-2}$$

$$A = \left(1 - \frac{1}{40}\right)^{40-1} = 0.37254609$$

$$U = \frac{1}{1+\frac{2a(1-A)}{A}} = \frac{1}{1+\frac{[2(1.953125 \cdot 10^{-2})][(1-0.37254609)]}{0.37254609}} = \frac{1}{1.1315805} = 0.883719$$

$$(14204.54\ \frac{paquetes}{seg})(0.883719) = 12552.82\ \frac{paquetes}{seg}$$

2) El cálculo de velocidad de transmisión para un tamaño de paquete de datos de 128 bytes con un número aproximado de equipos de 40 y una distancia de BUS aproximada de 30 metros el cálculo es el siguiente:

$N = 40$

$R = 10Mbps$

$d = 30m$

$V = 2 \cdot 10^{8}\ \frac{m}{s}$

$L = 128bytes$

$$\frac{(128+24)(8bits)}{10000000\,{}^{bits}/_{s}} = 121.6\mu s$$

El número de paquetes por segundo $= \frac{1paquete}{121.6\mu s} = 8223.6\ \frac{paquetes}{seg}$

$$a = \frac{R \cdot d}{V \cdot L} = \frac{(10000000bps)(30m)}{(2 \cdot 10^{8}\,{}^{m}/_{s})[(128)(8bits)]} = 1.46484375 \cdot 10^{-3}$$

$$A = \left(1 - \frac{1}{40}\right)^{40-1} = 0.37254609$$

$$U = \frac{1}{1+\frac{2a(1-A)}{A}} = \frac{1}{1+\frac{[2(1.46484375 \cdot 10^{-3})][(1-0.37254609)]}{0.37254609}} = \frac{1}{1.1315805} = 0.99508995$$

$$(8223.6\ \frac{paquetes}{seg})(0.99508995) = 12552.82\ \frac{paquetes}{seg}$$

3) El cálculo de velocidad de transmisión para un tamaño de paquete de datos de 256 bytes con un número aproximado de equipos de 40 y una distancia de BUS aproximada de 30 metros el cálculo es el siguiente:

$N = 40$

$R = 10Mbps$

$d = 30m$

$V = 2 \cdot 10^8 \frac{m}{s}$

$L = 256bytes$

$$\frac{(256+24)(8bits)}{10000000 \, bits/s} = 224\mu s$$

$$\text{El número de paquetes por segundo} = \frac{1paquete}{224\mu s} = 4464.28 \frac{paquetes}{seg}$$

$$a = \frac{R \cdot d}{V \cdot L} = \frac{(10000000bps)(30m)}{(2 \cdot 10^8 \, m/s)[(256)(8bits)]} = 7.32421875 \cdot 10^{-3}$$

$$A = \left(1 - \frac{1}{40}\right)^{40-1} = 0.37254609$$

$$U = \frac{1}{1 + \frac{2a(1-A)}{A}} = \frac{1}{1 + \frac{[2(7.32421875 \cdot 10^{-3})][(1-0.37254609)]}{0.37254609}} = \frac{1}{1.0786395} = 0.92709380$$

$$\left(4464.28 \frac{paquetes}{seg}\right)(0.92709380) = 4138.80 \frac{paquetes}{seg}$$

4) El cálculo de velocidad de transmisión para un tamaño de paquete de datos de 512 bytes con un número aproximado de equipos de 40 y una distancia de BUS aproximada de 30 metros el cálculo es el siguiente:

$N = 40$

$R = 10Mbps$

$d = 30m$

$V = 2 \cdot 10^8 \frac{m}{s}$

$L = 512bytes$

$$\frac{(512+24)(8bits)}{10000000 \, bits/s} = 428.8\mu s$$

$$\text{El número de paquetes por segundo} = \frac{1 paquete}{428.8 \mu s} = 2332.08 \frac{paquetes}{seg}$$

$$a = \frac{R \cdot d}{V \cdot L} = \frac{(10000000 bps)(30m)}{(2 \cdot 10^8 \, m/s)[(512)(8 bits)]} = 3.66 \cdot 10^{-4}$$

$$A = \left(1 - \frac{1}{40}\right)^{40-1} = 0.37254609$$

$$U = \frac{1}{1 + \frac{2a(1-A)}{A}} = \frac{1}{1 + \frac{[2(3.66 \cdot 10^{-4})][(1-0.37254609)]}{0.37254609}} = \frac{1}{1.0012328} = 0.99876871$$

$$\left(2332.08 \frac{paquetes}{seg}\right)(0.99876871) = 2329.20 \frac{paquetes}{seg}$$

5) El cálculo de velocidad de transmisión para un tamaño de paquete de datos de 1024 bytes con un número aproximado de equipos de 40 y una distancia de BUS aproximada de 30 metros el cálculo es el siguiente:

$$N = 40$$

$$R = 10 Mbps$$

$$d = 30m$$

$$V = 2 \cdot 10^8 \frac{m}{s}$$

$$L = 1024 bytes$$

$$\frac{(1024 + 24)(8 bits)}{10000000 \, bits/s} = 838.4 \mu s$$

$$\text{El número de paquetes por segundo} = \frac{1 paquete}{838.4 \mu s} = 1192.74 \frac{paquetes}{seg}$$

$$a = \frac{R \cdot d}{V \cdot L} = \frac{(10000000 bps)(30m)}{(2 \cdot 10^8 \, m/s)[(1024)(8 bits)]} = 1.8310546 \cdot 10^{-4}$$

$$A = \left(1 - \frac{1}{40}\right)^{40-1} = 0.37254609$$

$$U = \frac{1}{1 + \frac{2a(1-A)}{A}} = \frac{1}{1 + \frac{[2(1.8310546 \cdot 10^{-4})][(1-0.37254609)]}{0.37254609}} = \frac{1}{1.000616} = 0.9993843$$

$$\left(1192.74 \frac{paquetes}{seg}\right)(0.9993843) = 1192.005 \frac{paquetes}{seg}$$

En la tabla 3.28 se describe la función de la velocidad de transmisión en donde se muestra la relación del valor del porcentaje con el tamaño del paquete de datos y la velocidad de transmisión.

Función:	Velocidad de transmisión.
Entrada:	El tamaño del paquete de datos.
Salida:	Número de paquetes por segundo que circulan por la red.

Descripción:

El medidor de velocidad indica el numero de paquetes por segundo que circulan por la red, dependiendo del valor porcentual que tenga en ese momento la barra de progreso será el valor de la velocidad como se observa en la tabla 3.27.

Porcentaje de la barra de progreso (%)	**Paquetes (Bytes)**	**Velocidad de transmisión** $(Paquetes/seg)$
0	0	0
20	64	12552.82
40	128	8183.22
60	256	4138.80
80	512	2329.20
100	1024	1192.005

Tabla 3.27 – Tabla de relación entre velocidad de transmisión con el tamaño de paquete y el valor porcentual de la barra de progreso.

En la tabla 3.27 se observa la relación que existe entre el valor porcentual que tiene la barra de progreso con el tamaño del paquete de datos y la velocidad de transmisión, de esta manera es posible tener la información del consumo del ancho de banda junto con la cantidad de paquetes por segundo que cruzan por la red.

Tabla 3.28 - Función velocidad de transmisión

3.2.3.a.ii Medidor de equipos activos en la red.

El medidor de tráfico de red indica gráficamente el porcentaje y el número de equipos que se encuentran activos en la red, la función del medidor es descrita en la tabla3.29.

Función:	Medidor de tráfico de red.
Entrada:	Dirección IP del equipo que se encuentra activo en la red.
Salida:	Número de equipos conectados en red y cantidad de tráfico de los equipos conectados.
Descripción: El medidor de trafico de red toma las direcciones IP de los equipos activos que detecta el *sniffer* y las compara con un arreglo interno que contiene todas las direcciones IP del segmento, si la dirección IP ya se encuentra dentro del arreglo no se toma en cuenta para incrementar tanto el contador de equipos como el medidor de trafico de red (fig 3.47), de lo contrario si una nueva dirección IP no se encuentra dentro del arreglo se entiende que es un equipo que se acaba de activar y se incrementa el contador de equipos y el medidor de trafico de red generado por los equipos activos.  0% 20% 40% 60% 80% 100% Figura 3.47 - Barra de progreso que muestra el porcentaje de equipos activos en la red. En la figura 3.47 se muestra la barra de progreso con el porcentaje de los equipos activos de la red, este valor se obtiene mediante la multiplicación del número de equipos activos por el 100%, entre la cantidad total de los equipos que componen a la red. Ejemplo: Suponiendo que se tenga una red con 40 computadoras y que solo estén activas 10 computadoras se tendrá una medición en la barra de progreso del 25%.	

Tabla 3.29 – Función medidor de equipos activos en la red.

3.2.3.c Controles de usuario.

Los controles de usuario permiten la activación, selección y paro de algunas funciones, la sección 3.2.3.c.i presenta la barra de menú, la sección 3.2.3.c.ii explica el comportamiento del botón de inicio / alto y la sección 3.2.3.c.iii contiene la selección de interfaz y la sección 3.2.3.c.iv explica el editor de nodo de red. La figura 3.48 presenta el diagrama a bloques de los controles de usuario.

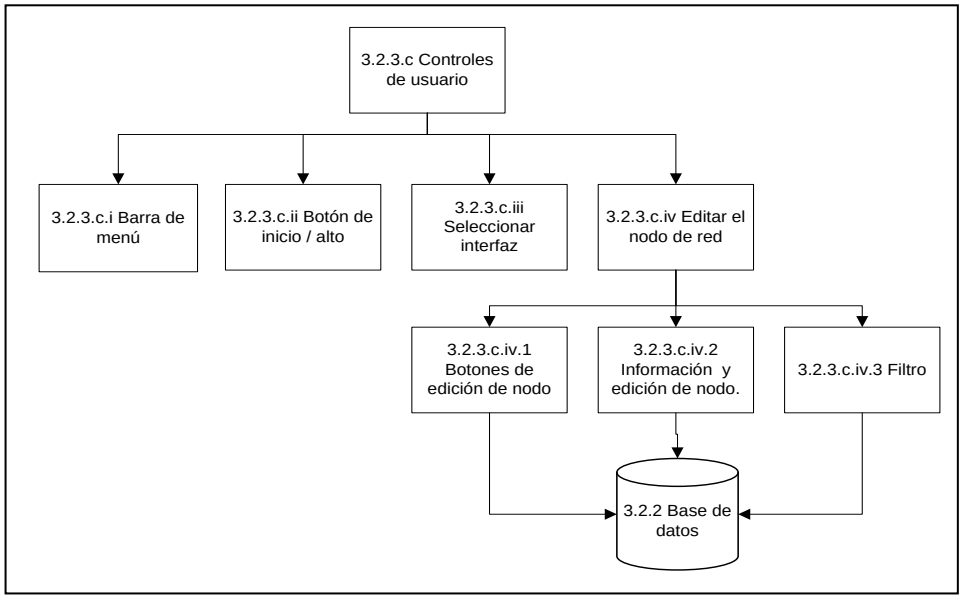


Figura 3.48 – Diagrama a bloques de los controles de usuario

3.2.3.c.i Barra de menú.

La barra de menú es la línea en la parte superior de la ventana, justo debajo del titulo de la aplicación, esta barra contiene funciones que se llaman submenús que contienen aplicaciones, esta barra facilita el acceso a aplicaciones que el usuario puede utilizar con solo activarlas, en la tabla 3.30 se describe la función de la barra de menú.

Función:	Barra de menú.
Entrada:	Activación de la aplicación por parte del usuario.
Salida:	Aplicación seleccionada por el usuario.
Descripción: La barra de menú contiene dos menús que son los siguientes: 1) Archivo: Contiene la función salir que permite al usuario salir de la aplicación con solo activarla. 2) Herramientas: Contiene las aplicaciones adaptadores de red, hacer PING y estadísticas, las cuales se pueden activar para visualizarlas en pantalla y utilizarlas. También se puede acceder a las aplicaciones por medio de teclas de acceso rápido, para acceder a cada submenú debe pulsar la combinación de teclas Alt+"tecla de la letra subrayada en el menú".	

Tabla 3.30 – Función barra de menú

3.2.3.c.ii Botón de inicio / alto.

La función de inicio y alto es la encargada de poner en marcha el tablero de control y depende de la función seleccionar interfaz, la descripción de esta función se encuentra en la tabla 3.31.

Función:	Inicio / Alto
Entrada:	Activación por parte del usuario.
Salida:	Activación o paro de las funciones del tablero de control.
Descripción: Este botón tiene dos estados que se describen en los siguientes puntos: 1) Inicio: Activa a las funciones de captura de datos, medidores de comportamiento de red, contadores de paquetes, conexiones a la base de datos, husmeador de paquetes, etc. 2) Alto: Detiene a las funciones mencionadas en el punto 1. Esta función esta ligada al adaptador de red instalado en el equipo para que las funciones de análisis de red obtengan la información y la muestren al usuario.	

Tabla 3.31 – Función inicio / alto

3.2.3.c.iii Seleccionar interfaz.

La función seleccionar interfaz detecta el nombre o la dirección IP asociada al adaptador de red instalado en el equipo de computo, en donde se obtiene toda la información de la red, la tabla 3.32 contiene la descripción de esta función.

Funciones:	Seleccionar interfaz
Entrada:	Nombre o dirección del adaptador de red (`comboBox`)
Salida:	Información de la red.
Descripción: En el `comboBox` de la interfaz de red se obtiene el nombre o dirección de los adaptadores de red instalados en la computadora, se selecciona un adaptador sobre el cual se desea iniciar la captura de datos de red y después se activa el botón de inicio /alto descrito en la tabla 3.39 de la sección 3.4.3.2 botón de inicio / alto, para comenzar con la captura de datos de la red y que las demás funciones del tablero de control obtengan información.	

Tabla 3.32 – Función seleccionar interfaz

3.2.3.c.iv Editor de nodo de red.

El editor de nodo de red se encarga de almacenar en la base de datos la información que el usuario puede borrar, insertar, modificar y buscar en la interfaz gráfica, para facilitar la edición de los nodos se contienen los siguientes puntos:

- Botón insertar.
- Botón modificar.
- Botón borrar.
- Botón buscar.

3.2.3.c.iv.1 Botones de edición de nodo.

Los botones de herramientas de edición de nodo facilita al usuario insertar nodos, buscar nodos para moverse entre los registros de la base de datos, borrar nodos y realizar modificaciones, la tabla 3.33 describe la función de la barra de herramientas.

Función:	Botones de herramientas de edición de nodo.
Entrada:	Borrar registro, insertar registro, modificar registro y buscar registro
Salida:	Los registros actualizados
Descripción: Los botones son independiente del botón de inicio / alto y sirve para realizar modificaciones en la base de datos, su funcionamiento se describe en los siguientes puntos: • Botón Borrar: Elimina el registro seleccionado de una tabla determinada en la base de datos. • Botón Insertar: Inserta un registro en una tabla determinada en la base de datos. • Botón modificar: Actualiza alguna modificación de un registro de una tabla determinada en la base de datos. • Botón Buscar: Realiza una consulta a la base de datos para encontrar un registro de una tabla determinada en la base de datos.	

Tabla.3.33 – Función de los botones de edición de nodo.

3.2.3.c.iv.2 Información y edición de nodo.

La información que contiene el editor de nodo se puede editar en la interfaz gráfica o directamente en la base de datos, los cambios se verán reflejados cuando se actualicen las ventanas respectivamente, la información que contiene esta función se encuentra en la tabla 3.34.

Función:	Información y edición de nodo.
Entrada:	Identificador de equipo, identificador de segmento de red, número de nodo, Tipo de equipo, departamento donde se encuentra ubicado el equipo, el estado del equipo.
Salida:	Información contenida en la base de datos

Descripción:

El usuario puede realizar modificaciones a los registros de la base de datos desde el entorno grafico, en el caso de borrar un registro se tienen los siguientes puntos:

1) Active el botón borrar.
2) Seleccione una tabla.
3) Seleccione el registro que se desea borrar.
4) Active el botón eliminar.

Estos cambios se verán reflejados directamente en la base de datos, en al caso de agregar un registro se tienen los siguientes puntos:

1) Active el botón Agregar.
2) Seleccione la tabla donde desea agregar un nuevo registro.
3) Escriba la información del nuevo registro
4) Active el botón insertar.

En el caso de actualizar la información de un registro se tienen los siguientes pasos:

1) Active el botón modificar.
2) Seleccione la tabla donde desea modificar el registro
3) Seleccione el registro que desea actualizar
4) Active el botón modificar

En el caso de buscar un registro en una tabla, se tienen los siguientes pasos:

1) Active el botón buscar.
2) Seleccione la tabla donde desea buscar el registro.
3) Seleccione un valor y un campo[75].
4) Active el botón buscar

Tabla.3.34 – Función información y edición de nodo.

[75] Los campos son las columnas de la tabla de la base de datos.

3.2.3.c.iv.3 Filtro.

Esta función permite realizar un filtrado de información en el editor de nodo de red, en caso de que el usuario requiera información definida de un o más registros, en la tabla 3.35 se describe la función filtro.

Función:	Filtro.
Entrada:	Campo (comboBox) y parámetro[76] de interés del usuario (imputBox)
Salida:	Despliegue de información filtrada.
Descripción: Para el filtrado de la información se tiene los siguientes pasos: 1) Seleccione el campo de interés. 2) Active el botón buscar. 3) Escriba en el cuadro de dialogo la palabra de interés[77]. 4) Active el botón aceptar. 5) La información aparecerá en el entorno gráfico. Para volver a ver toda la información solo borre la información en el comboBox o si se requiere hacer otro filtrado seleccione otro campo y realice los puntos anteriores.	

Tabla 3.35 – Función filtrado de la función editor de nodo de red.

[76] Los parámetros es una palabra que contiene una celda.
[77] Esta palabra debe de estar contenida en un campo de la tabla.

3.3 Resumen.

La arquitectura del tablero de control esta compuesta por tres elementos principales, 1) un módulo de adquisición de datos que se encarga de obtener, analizar y procesar los datos enviados por el medio de transmisión, sin afectar el proceso de envío y recepción de datos utilizando un husmeador paquetes, 2) una base de datos que almacena la información del módulo adquisición de datos y 3) una interfaz gráfica que despliega la información utilizando medidores que muestran el comportamiento del consumo de ancho de ancho de banda, la cantidad de paquetes de entrada, salida y errores, junto los equipos activos en la red; además permite generar reportes, agregar, modificar, buscar y borrar nodos dentro de la base de datos, junto con la configuración de segmento de red, equipos instalados y alarmas.

Capítulo 4

Pruebas y resultados

Éste capítulo comienza con la sección 4.1 que contiene la prueba del módulo adquisición de datos de red, la sección 4.2 presenta las pruebas de la base de datos y la sección 4.3 contiene las pruebas de la interfaz gráfica. Las pruebas de los módulos se realizan utilizando la infraestructura de red instalada en el edificio de posgrado del Centro de Investigación e Innovación Tecnológica (CIITEC) del Instituto Politécnico Nacional.

4.1 Pruebas del módulo adquisición de datos de red.

En esta sección se encuentran las pruebas de los submodulos, la sección 4.2.1 contiene la prueba del módulo husmeador de paquetes (*sniffer*), la sección 4.2.2 presenta la prueba del módulo estadísticas, la sección 4.2.3 contiene las pruebas del módulo comprobación de conexión (PING) y la sección 4.2.4 presenta la prueba del módulo adaptadores de red.

4.1.1 Prueba del módulo husmeador de paquetes (Sniffer).

La prueba de este módulo consiste en detectar el tráfico de red, conectando un cable UTP categoría 5e en el puerto RJ45 del adaptador de red instalado en la computadora.

El procedimiento para realizar la prueba se describe en los siguientes puntos:

1) Seleccione el nombre o dirección IP asociada a la interfaz de red (`ComboBox`) instalada en la computadora (fig.4.1).
2) Active el botón “Iniciar” para dar comienzo con la captura de datos de la red, el estado del botón cambia a “Alto”.

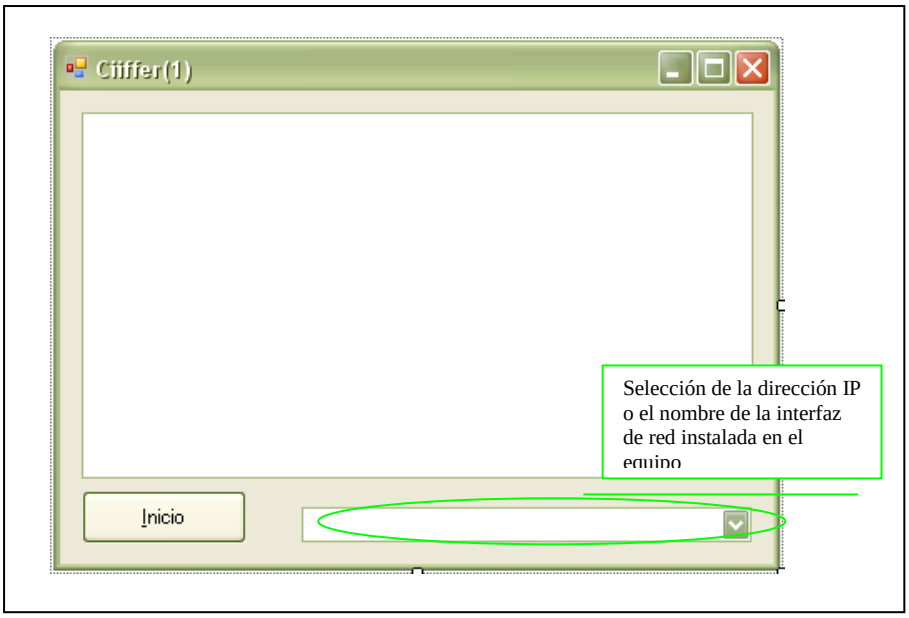


Figura 4.1 Interfaz gráfica del husmeador de paquetes.

3) Prueba de detección de direcciones IP origen y destino (en su versión cuatro en el `TreeView`) separadas por un guion, el resultado de esta prueba se muestra en la figura 4.2.

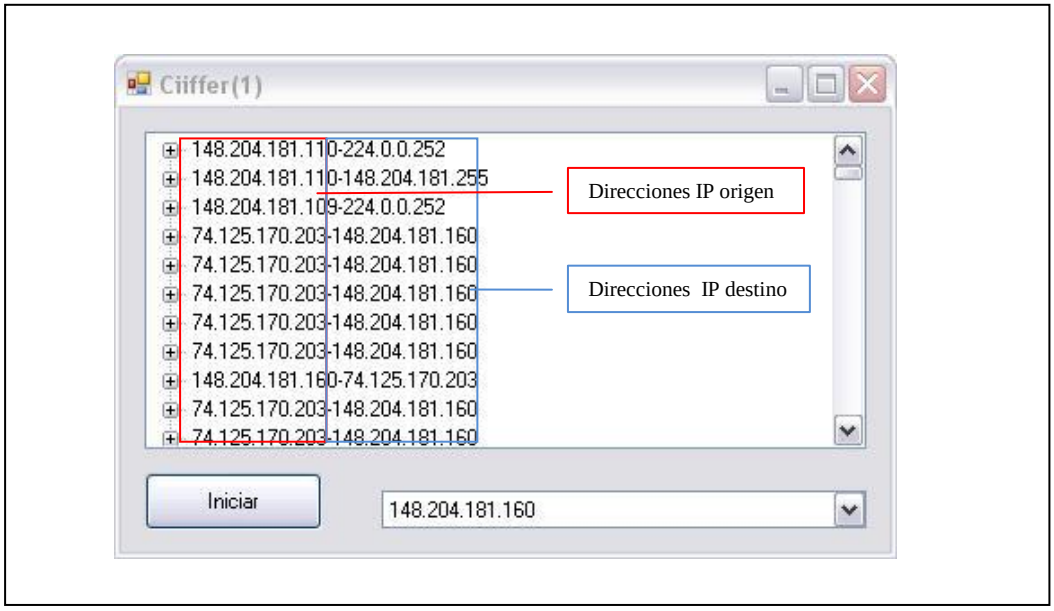


Figura 4.2 – Resultado de la prueba de detección de dirección IP origen y destino

En la figura 4.2 se observa el resultado de la prueba de detección de las IP's origen encerradas en un recuadro en color rojo y las IP's destino encerradas

en un recuadro azul, lo que valida que el husmeador de paquetes detecta información de la red.

4) Prueba de detección de información asociada a las direcciones IP origen y destino, activando el botón con el símbolo mas (+) de cada dirección detectada, en donde se puede ver el tipo de protocolo TCP o UDP (según sea el caso) que esta siendo transportado por el encabezado de IP, el resultado se muestra en la figura 4.3.

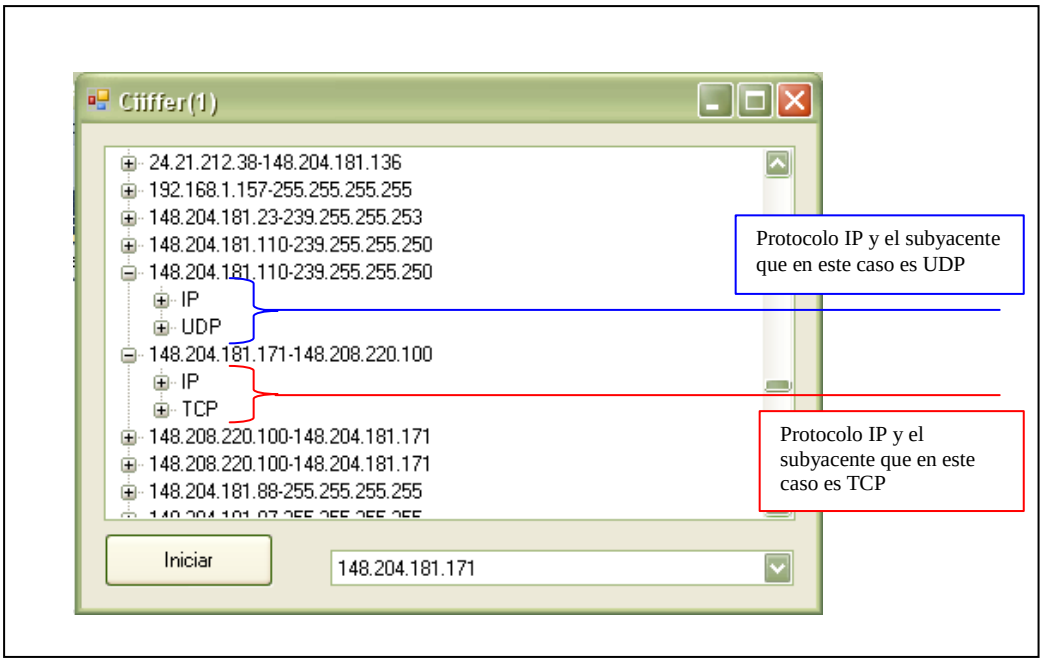


Figura. 4.3 – Resultado de la prueba de detección de protocolos.

En la figura 4.3 se observa el resultado de la prueba de detección de los protocolos utilizados para la transmisión de información activando el botón con el símbolo (+), los cuales se enumeran en los siguientes puntos:

1) IP
2) TCP
3) UDP

5) Prueba de detección de información del encabezado de IP, el resultado de esta prueba se muestra en la figura 4.4.

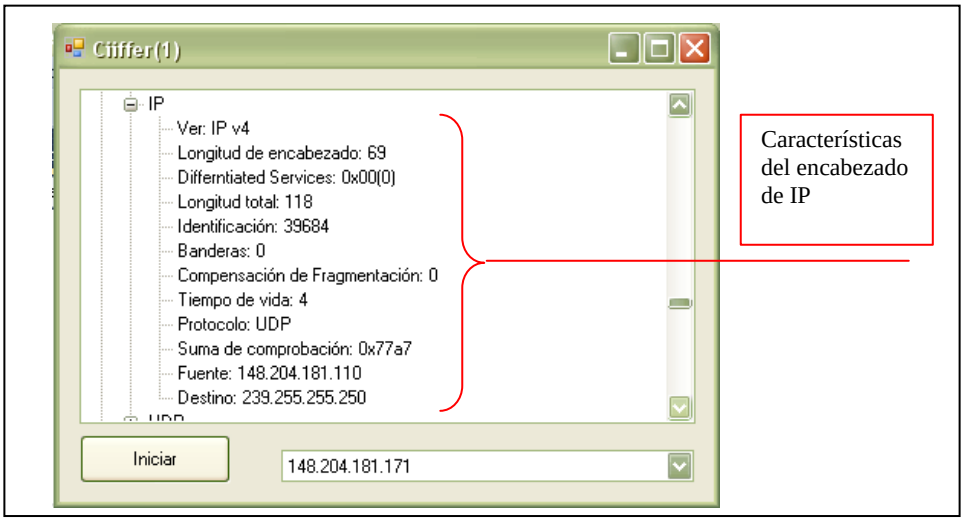


Figura. 4.4 – Resultado de la prueba de análisis del encabezado de IP

En la figura 4.4 se observar el resultado de la detección de las características del encabezado de IP activando el botón con el símbolo (+), estas son las descritas en el capítulo 3 en la sección análisis de encabezado de IP, las cuales se enumeran en los siguientes puntos:

1) Versión de IP
2) Servicios
3) Longitud total de la trama de datos
4) Indicadores
5) Fragmentación
6) Banderas
7) Tiempo de vida
8) Protocolo encapsulado por la trama de datos
9) Dirección IP origen
10) Dirección IP destino

6) Prueba de detección de información asociada al protocolo UDP, el propósito de esta prueba es comprobar la detección de las características del encabezado de UDP, el resultado de esta prueba se muestra en la figura 4.5.

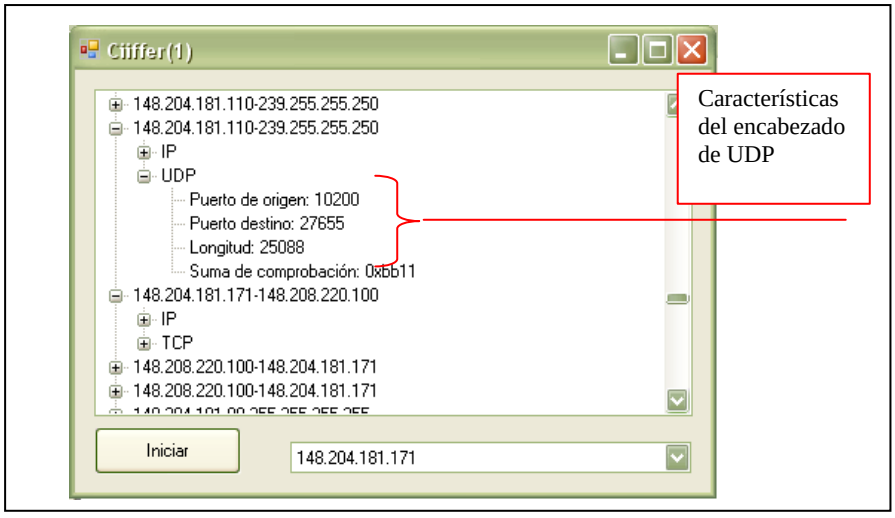


Figura.4.5 – Resultado de la prueba de análisis del encabezado de UDP

En la figura 4.5 se observa el resultado de la prueba de detección de las características del encabezado de UDP activando el botón con el símbolo (+), descritas en el capítulo 3 en la sección análisis de encabezado de UDP, las cuales se enumeran en los siguientes puntos:

1) Puerto de origen.

2) Puerto destino.

3) Longitud del paquete de UDP.

4) Suma de comprobación.

7) Detección de información asociada al protocolo TCP, el propósito de esta prueba es observar la detección del protocolo TCP así como de sus características, el resultado de esta prueba se muestra en la figura 4.6.

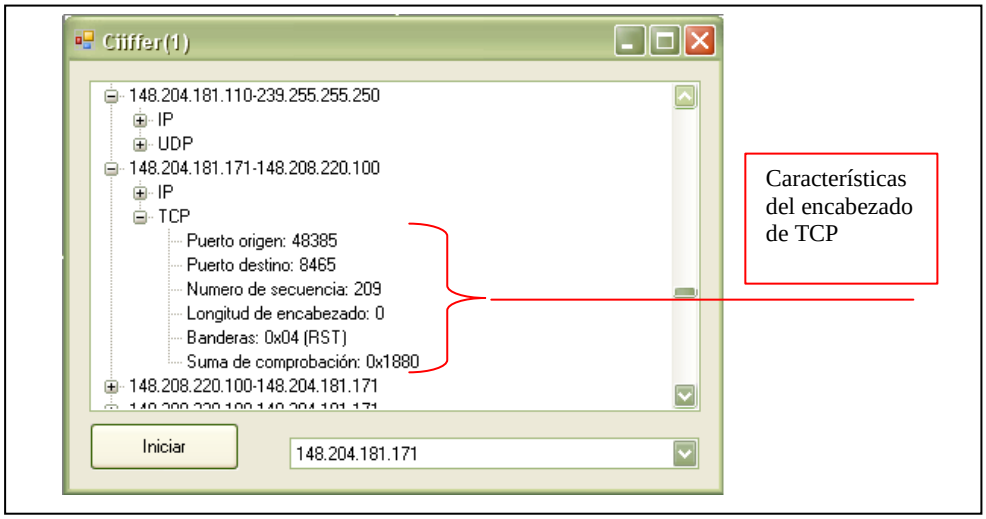


Figura. 4.6 – Resultado de la prueba de análisis del encabezado de TCP.

La figura 4.6 contiene el resultado de la prueba de detección de las características del encabezado de TCP activando el botón con el símbolo (+), descritas en el capítulo 3 en la sección análisis de encabezado de TCP, las cuales se enumeran en los siguientes puntos:

1) Puerto de origen
2) Puerto destino
3) Número de secuencia
4) Longitud del encabezado
5) Bandera de compensación de datos
6) Suma de comprobación.

Con los resultados satisfactorios de las pruebas de este módulo ya se cuenta con información suficiente para que el tablero de control funcione con los datos obtenidos directamente de la red por medio del husmeador de paquetes, una de las ventajas es que no ocupa herramientas prediseñadas como WinPcap para obtener la información de la red como lo hacen otras aplicaciones comerciales, otra ventaja que tiene el módulo husmeador de paquetes que funciona para cualquier tipo de interfaz de red, la tabla 4.1 contiene el resumen de los resultado de la pruebas de este módulo.

Resultado de las pruebas del modulo husmeador de paquetes.		
Parámetro	Cumple	No Cumple
Detección de dirección IP origen	✓	
Detección de dirección IP destino	✓	
Análisis del encabezado de IP	✓	
Descripción del encabezado de IP	✓	
Análisis del encabezado TCP	✓	
Descripción del encabezado de TCP	✓	
Análisis del encabezado UDP	✓	
Descripción del encabezado de UDP	✓	
Análisis del encabezado DNS	✓	
Descripción del encabezado de DNS	✓	
Maneja protocolo ARP		✗
Maneja protocolo SNMP		✗
No incrementa el trafico de red	✓	
No interfiere en el envió y recepción de paquetes	✓	

Tabla 4.1 – Resumen de resultados de las pruebas del modulo husmeador de paquetes

4.1.2 Prueba del módulo estadísticas.

La prueba de este módulo se realiza de dos formas que están descritas en los siguientes puntos:

1) La primera prueba se realiza conectando un cable de red UTP categoría 5e en el puerto RJ45 del adaptador de red instalado en la computadora, para demostrar la recepción de paquetes por un medio de transmisión cableada (fig.4.7)

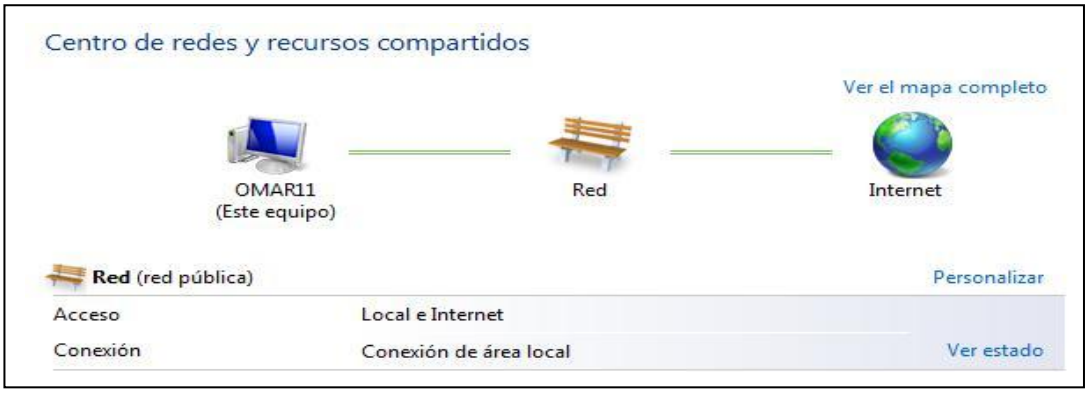


Figura 4.7 – Ventana de conexión a la red del CIITEC

2) La segunda prueba es deshabilitando la interfaz de red y desconectando el cable UTP categoría 5e, después se activa la conexión de red inalámbrica (Fig. 4.8) para demostrar que funciona la recepción de paquetes por medio de un canal de difusión[78] (conexión inalámbrica).

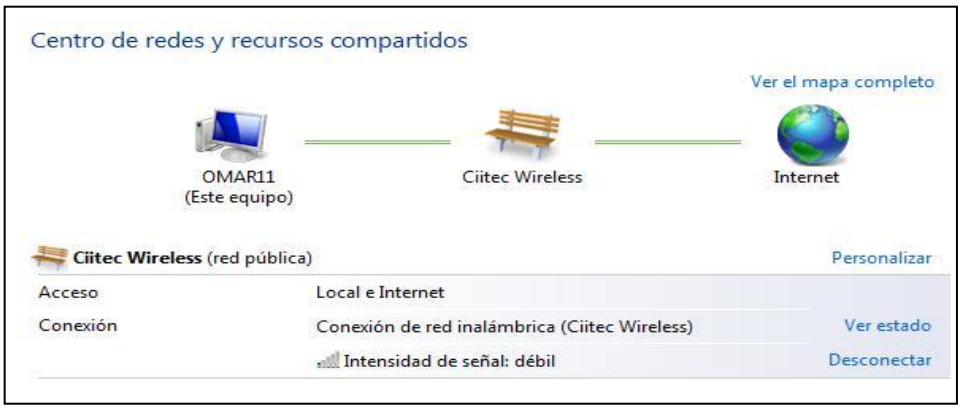


Figura. 4.8 Ventana de conexión a la red inalámbrica del CIITEC

Las pruebas que se realizan a este módulo se describen en los siguientes puntos:

1) Para la primer prueba se activa el adaptador de red Ethernet y después se activa el módulo de estadísticas, después aparece la ventana con las estadísticas de paquetes de entrada y salida del adaptador de red por medio del cable UTP categoría 5e (fig.4.9), después se activa el botón cerrar de la venta para parar la ejecución del programa.

[78] Medio de comunicación por donde viaja la forma de onda de la señal (portadora de la información) del transmisor al receptor.

Figura. 4.9 – Resultado de la prueba de detección de paquetes de entrada y salida por un medio de transmisión cableado.

La figura 4.9 contiene el resultado de la prueba de detección de paquete de entrada salida por medio de un cable UTP categoría 5e, demostrando que el modulo de estadísticas funciona por un medio de transmisión.

2) Para realizar la segunda prueba se desconectan el cable UTP del adaptador de red Ethernet y se activa el adaptador de red inalámbrico, después se activa el modulo de estadísticas e inmediatamente aparece la ventana con la información de los paquetes de entrada y salida que adquiere el adaptador de red inalámbrico del medio de difusión (fig.4.10), después se activa el botón cancelar de la ventana para parar la ejecución del programa.

Figura. 4.10 – Resultado de la prueba de detección de paquetes de entrada y salida con conexión inalámbrica.

La figura 4.10 contiene la cantidad de paquetes de entrada y salida que adquiere el adaptador de red al estar conectado a la red inalámbrica del CIITEC, es importante destacar este hecho por que muchas de las aplicaciones necesitan de

un software especial para que el adaptador de red pueda adquirir información por medio de una conexión inalámbrica.

La tabla 4.2 contiene los resultados de las pruebas experimentales que se realizaron a este modulo de estadísticas.

Resultados del modulo estadísticas		
Parámetro	Cumple	No cumple
Detecta paquetes de entrada	✓	
Detecta paquetes de salida	✓	
Funciona de forma inalámbrica	✓	
Funciona de forma cableada	✓	

Tabla 4.2 Resumen de resultados del modulo estadísticas.

4.1.3 Prueba del módulo comprobación de conexión de equipo (PING).

Las pruebas para el módulo de comprobación de conexión de equipo también se realizan de dos formas, cableada e inalámbrica (como en la sección 4.2.2), bajo estos dos tipos de conexiones se realizan tres pruebas descritas en los siguientes puntos:

1) Comprobar la conexión con el servidor www.google.com, sin realizar modificaciones en el número de eco, tamaño del buffer y tiempo de vida,

   1. Activar la aplicación Hacer PING
   2. Escribir www.google.com en el espacio debajo del letrero “Introduzca nombre del equipo”, encerrado en un recuadro de color rojo (`TextBox`) (fig.4.11).
   3. Activar el botón **Comprobar.**
   4. El resultado de la solicitud de conexión se despliega cuatro veces en pantalla (`ListView`) (fig.4.11), por que es la forma estándar en como lo realiza el comando *ping* del sistema de Windows.

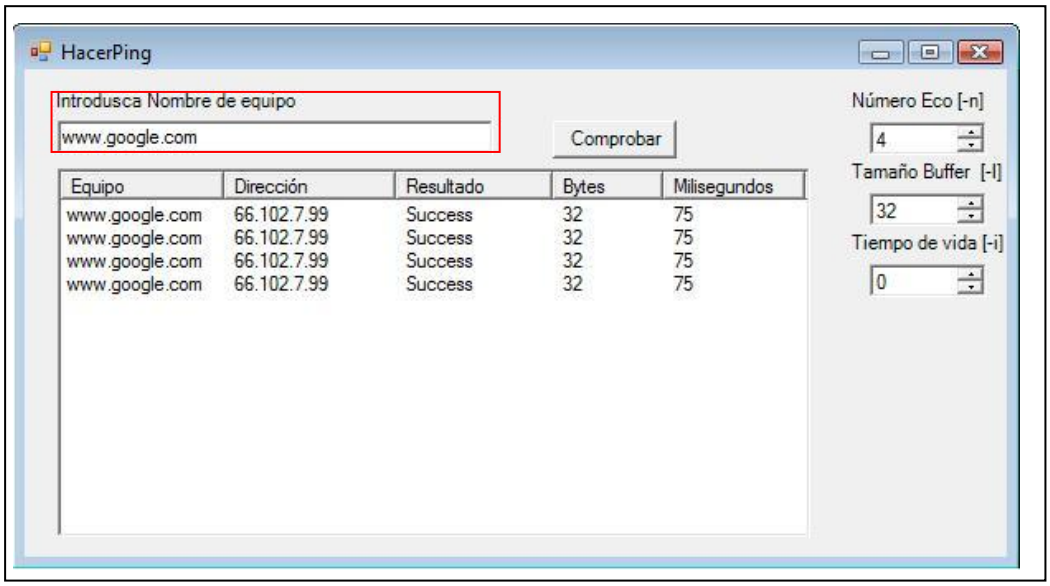


Figura. 4.11 – Resultado de la prueba de comprobación de conexión con el servidor www.google.com.

En la figura 4.11 se observa el resultado de la respuesta de la solicitud hecha al servidor www.google.com por medio del uso del protocolo ICMP con el cual se obtiene la información enumerada en los siguientes puntos y que se explicó en el capítulo 3 en la sección comprobación de conexión (PING):

1) Resuelve el nombre o dirección del equipo remoto.

2) Comprueba conexión regresando la IP del equipo remoto.

3) Envió de mensaje de conexión con el equipo remoto.

4) Tamaño de buffer del paquete de datos.

5) Tiempo de vida del paquete.

2) Comprobar la conexión con el servidor www.google.com, modificando los parámetros número de eco, tamaño del buffer y tiempo de vida, los pasos para realizar esta prueba se describen en los siguientes puntos:

1. Colocar el número de eco igual al valor de 10 activando el botón con la flecha hacia arriba del "Número Eco", marcado con un recuadro en color rojo (`NumericUpDow1`) (fig.4.12).

2. Colocar el tamaño del buffer igual con 45 activando el botón con la flecha hacia arriba del "Tamaño Buffer", marcado con un recuadro de color verde (`NumericUpDow2`) (fig.4.12).

3. Colocar el tiempo de vida igual con 5 activando el botón con la flecha hacia arriba del "Tiempo de vida", marcado con un recuadro de color azul (`NumericUpDow3`) (fig.4.12).

4. Escriba www.google.com en la sección "Introduzca nombre de equipo" (`TextBox`).

5. Activar el botón **Comprobar**.

6. El resultado de la solicitud de respuesta de la conexión se despliega en pantalla (`ListView`) (fig.4.12), donde los valores de número de eco, tamaño de buffer y tiempo de vida se pueden observar claramente.

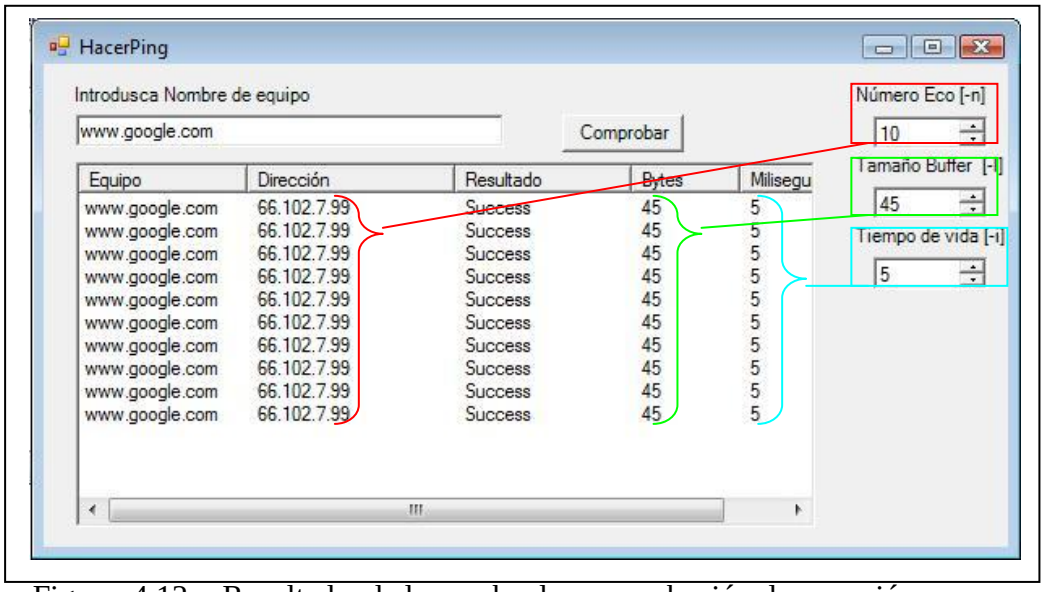


Figura. 4.12 - Resultado de la prueba de comprobación de conexión con cambios en los valores de eco, tamaño de buffer y tiempo de vida.

En la figura 4.12 se observa el resultado de la prueba comprobación de conexión, modificando los siguientes parámetros:

1) Variación del número de eco.

2) Variación del tamaño del buffer.

3) Variación del tiempo de vida.

3) Prueba de comprobación de conexión con un equipo que no responda a la solicitud de conexión y visualizar el mensaje de "Host de destino inaccesible" (fig.4.13) por tres ocasiones en que se intenta realizar la conexión y después de realizar los tres intentos sin existo de respuesta por parte del equipo remoto se visualiza el mensaje "Se a terminado el tiempo de espera del host" (fig.4.14), los pasos para realizar esta prueba se describen en los siguientes puntos:

1. Escriba algún nombre o dirección IP de un equipo en la sección "Introduzca nombre de equipo" (en este caso se escribió el nombre jhjonatan).

2. Activar el botón **Comprobar**.

3. Se obtienen las respuestas del host de destino inaccesible por tres ocasiones mediante un cuadro de mensaje (fig.4.13).

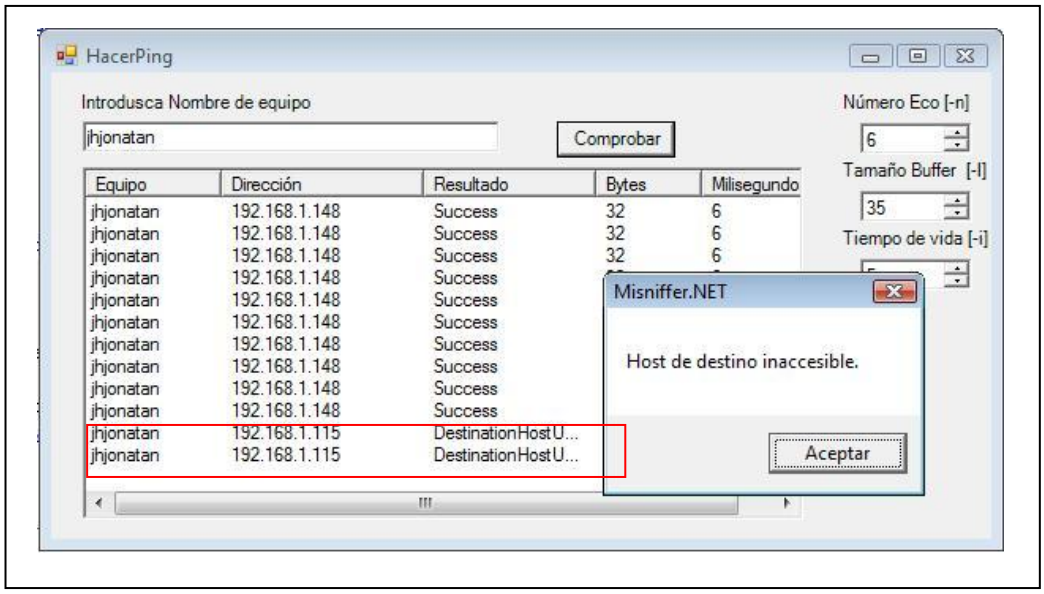


Figura. 4.13 – Resultado de la prueba de respuesta inaccesible a la computadora destino

En la figura 4.13 se observar el resultado encerrado en un rectángulo de color rojo con dos intentos de conexión sin repuesta del equipo remoto, también se observa el cuadro de dialogo con el mensaje "Host de destino inaccesible" por tercera ocasión.

4. Prueba del despliegue de la respuesta del término de tiempo de espera, que se presenta por medio de un cuadro de mensaje después de tres intentos de conexión (fig.4.14), mediante el mensaje "Sea terminado el tiempo de espera del host".

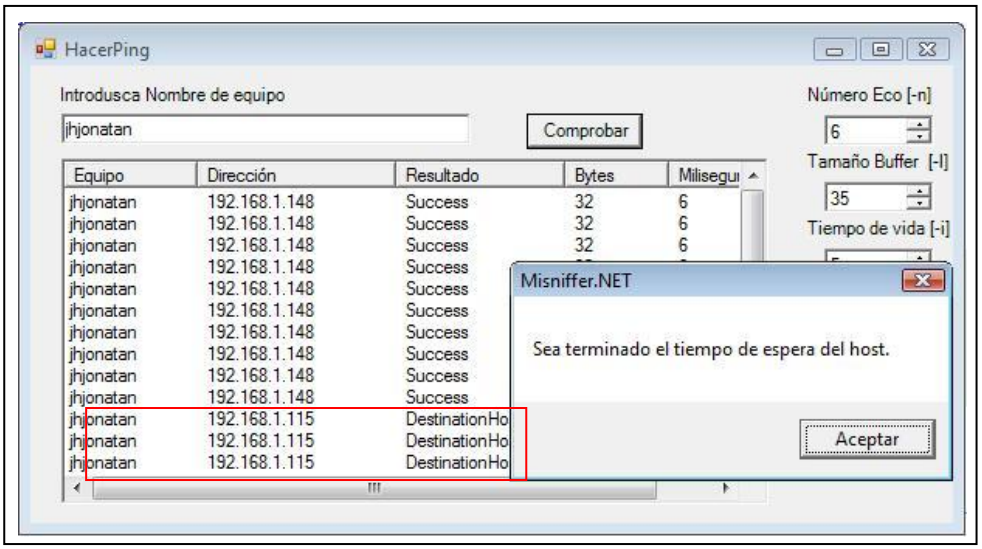


Figura. 4.14 – Resultado de la prueba de término del tiempo de respuesta

La Figura 4.14 contiene el resultado del intento de conexión a un equipo que no responde a la solicitud de conexión por tercera ocasión, presentando un cuadro de dialogo indicando que el tiempo para realizar la conexión sea agotado.

5. Para cerrar el modulo se activa el botón aceptar del cuadro de dialogo del termino de tiempo de espera y luego se activa el botón cerrar de la aplicación hacer PING para detener el programa, el resumen de los resultados de las pruebas se encuentran en la tabla 4.3.

Resultados de pruebas de la función PING		
Parámetro	Cumple	No cumple
Manejo de protocolo ICMP	✓	
Comprueba conexión con el equipo remoto	✓	
Resuelve el nombre o dirección del host.	✓	
Envió de mensaje de conexión al host especificado.	✓	
Variación del valor de número de eco.	✓	
Variación del valor de tamaño de buffer.	✓	
Variación de tiempo de vida.	✓	
Tipo de servicio.		✗
Envió de mansaje de equipo no disponible	✓	
Envió de mensaje de tiempo terminado de espera	✓	
Marcador no fragmentar en paquetes.		✗

Tabla 4.3 – Tabla de resultados de las pruebas del modulo PING.

4.1.4 Prueba del módulo adaptadores de red.

La prueba del módulo detección de interfaces de red también se divide en dos partes las cuales se describen en los siguientes puntos:

1) La primera prueba por medio de la conexión de un cable UTP categoría 5e conectado al puerto RJ45 del adaptador de red Ethernet instalado en la computadora (fig. 4.15); se activa el módulo adaptadores de red para mostrar la información del adaptador de red.

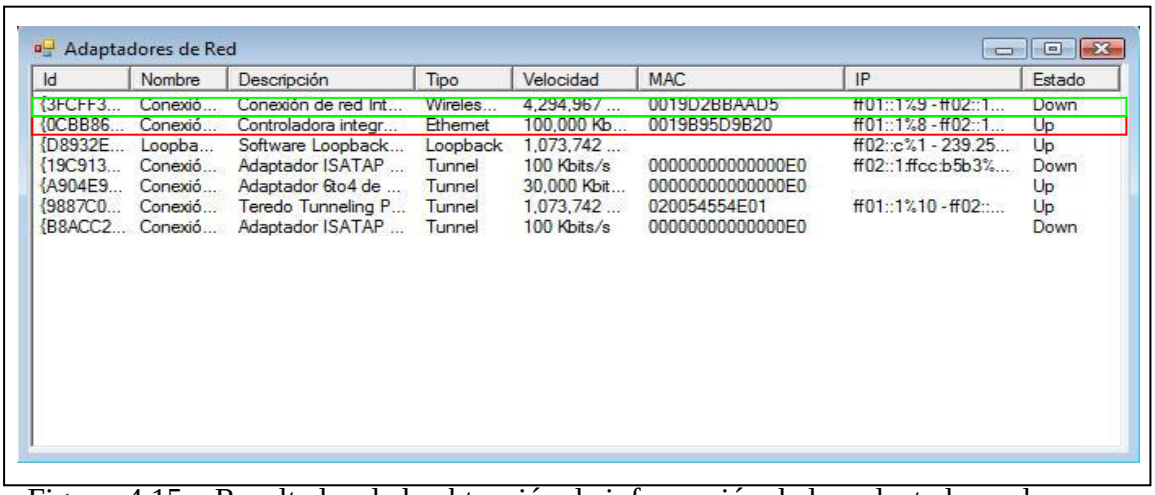


Figura. 4.15 – Resultados de la obtención de información de los adaptadores de red

En la figura 4.15 se observa el resultado del despliegue de la información de las interfaces de red instaladas en el equipo, la columna “Estado” indica que la interfaz de red inalámbrica se encuentra desactivada (enmarcada en un recuadro de color verde) y que el adaptador de red Ethernet esta activo (enmarcada en un recuadro de color rojo), esto demuestra también que el programa detecta correctamente los dispositivos activos e inactivos, además de las características descritas en el capítulo 3 en la sección de interfaces de red, los siguientes puntos se enumeran los parámetros que son detectados:

a. Identificador del adaptador de red,

b. Nombre

c. Descripción

d. Tipo

e. Velocidad

f. Dirección física (MAC)

g. Dirección IP

h. Estado, que en este caso tiene activo el adaptador de red Ethernet.

2) La segunda prueba se realiza desactivando el adaptador de red Ethernet y activando la interfaz de red inalámbrica la cual detecta la red inalámbrica del CIITEC y se conecta a ella, entonces se realiza la prueba de detección de interfaces de red, la cual ahora indica que la interface de red inalámbrica esta

activa (enmarcada en un recuadro de color verde) y la conexión Ethernet como inactiva (enmarcada en un recuadro de color rojo) (fig.4.16).

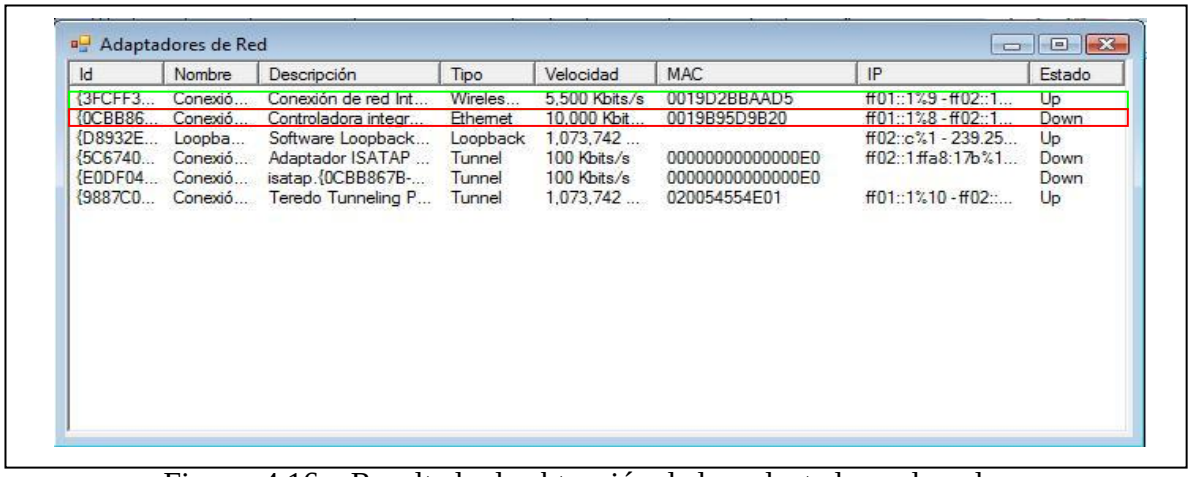


Figura. 4.16 – Resultado de obtención de los adaptadores de red

La figura 4.16 contiene el resultado de la detección de los parámetros del adaptador de red como el identificador, nombre del adaptador de red, su descripción, el tipo, la velocidad que maneja, su dirección física, la dirección IP y el estado del adaptador de red. En la tabla 4.4 se presenta el resumen de las pruebas realizadas al módulo adaptadores de red.

Resultados del modulo adaptadores de red		
Parámetro	Cumple	No cumple
Detecta las interfaces de red instaladas en el equipo	✓	
Obtiene el identificador del adaptador de red	✓	
Obtiene el nombre de fabrica del adaptador de red	✓	
Obtiene la descripción del adaptador de red	✓	
Indica el tipo de adaptador de red	✓	
Indica la velocidad del adaptador de red	✓	
Obtiene la dirección física del adaptador de red	✓	
Obtiene la dirección IP asociada al adaptador de red	✓	
Indica el estado del adaptador de red	✓	
Funciona de forma inalámbrica	✓	
Funciona de forma cableada	✓	
Identifica con algún color el equipo activo o inactivo		✘

Tabla 4.4 – Tabla de resultados del modulo adaptadores de red

4.2 Pruebas de la base de datos.

En esta sección se realizan pruebas para ver el desempeño del sistema de administración de base de datos (SABD) utilizado para almacenar la información del análisis de red del tablero de control, la sección 4.3.1 contiene las pruebas de desempeño en la carga de datos y la sección 4.3.2 presenta las pruebas de desempeño en la consulta de datos.

4.2.1 Pruebas de desempeño en la carga de datos.

Pruebas de desempeño del manejador de base de datos en la carga de datos a una sola tabla (M = Millones de registros)

1) Los rangos de los registros de carga son 1M, 5M, 10M, 20M

   Para desarrollar esta prueba se creo una aplicación en Visual Basic.NET con la cual se generan los millones de registros para hacer las pruebas con el manejador de la base de datos, la aplicación se llama "Genera datos[79]". Al ejecutar la aplicación aparece en pantalla una ventana como se muestra en la figura 4.17.

Figura.4.17 - Vista de la aplicación Generador de datos.

   En la figura 4.17 se puede observar que se tiene un indicador de registros y un botón para comensar la ejecución del programa. El número de registros se modifica directamente en el codigo fuente de la aplicación.

[79] El código fuente lo puede encontrar en el apéndice "D".

Cuando se activa el botón “Generar” apararece una ventana para indicar la ubicación y el nombre con el que se guarda el archivo con los registros como se puede ver en la figura.4.18.

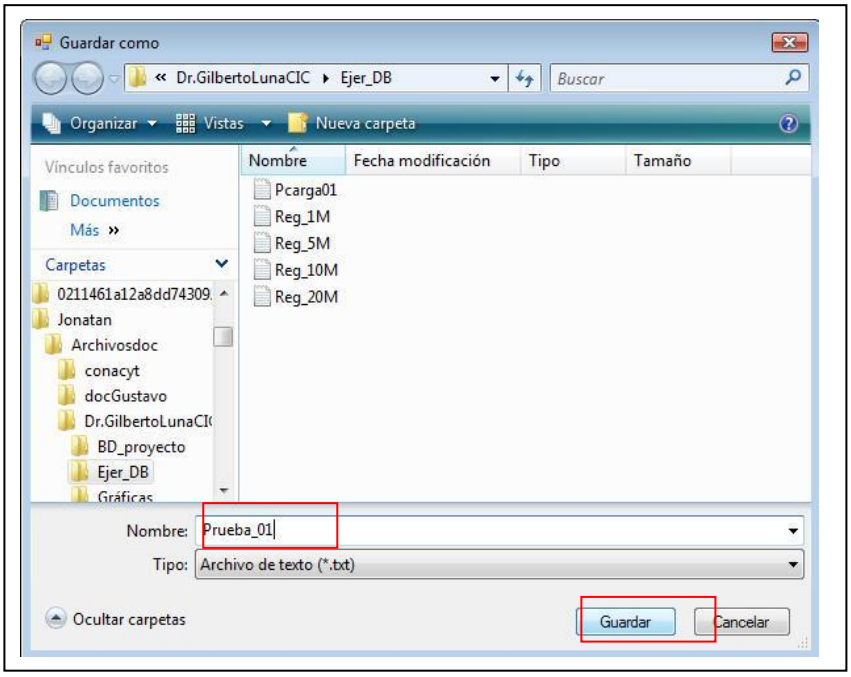


Figura.4.18 - Ventana de ubicación de archivos con los registros.

En la figura 4.18 se observa que se asigna el nombre del archivo y se activa el botón “Guardar”, una vez que se guarda el archivo el programa empieza a generar los registros en la ubicación indicada, mostrando el avance en la sección “Número de registros:”, cada vez que llegue a un intervalo determinado aparecerá un cuadro de dialogo indicando la cantidad de registros escritos en el archivo como se muestra en la figura.4.19.

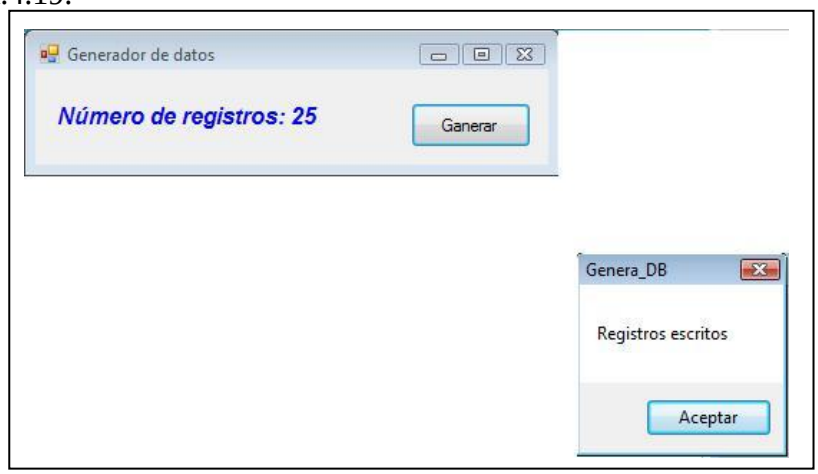


Figura.4.19 - Funcionamiento de la aplicación Generador de datos.

Una vez que se halla terminado de generar los datos en el archivo, aparecerá otro cuadro de dialogo mostrando el mensaje “Generación de datos finalizada”, indicando que el archivo con los registros se encuentra listo (Fig.4.20).

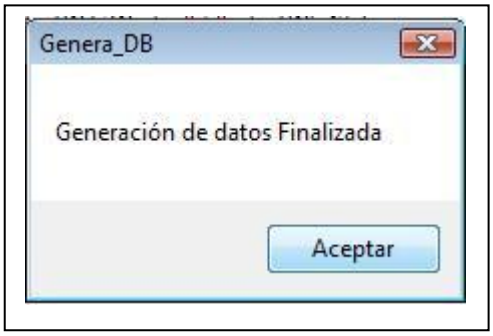


Figura.4.20 - Mensaje que indica que los archivos ya fueron generados en el archivo de datos.

2) Son tres los estados para realizar las pruebas de las tablas dentro de la base de datos los cuales se describen en los siguientes puntos:

a) Sin llave y sin índice.

Se edita una tabla en el SABD de MySQL, en este caso se crea una tabla con el nombre “test_area00” en la base de datos como lo muestra la figura 4.21.

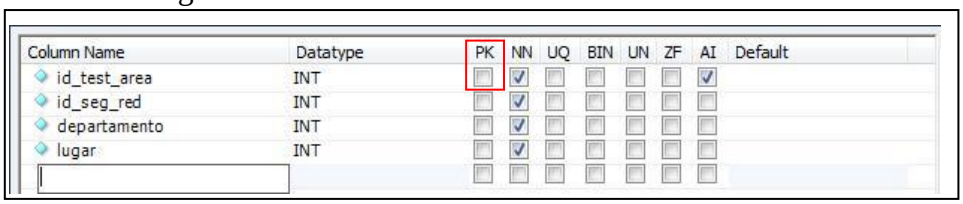


Figura.4.21 – Edición de la tabla “test_area00” sin llave primario

En la figura 4.21 se observar un rectángulo de color rojo que indica que no se activa la casilla de llave primaria (PK) en la tabla “test_area00” así como tampoco se crea un índice (fig 4.22).

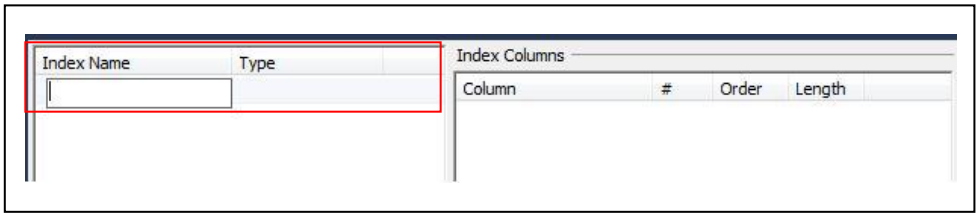


Figura.4.22 – Edición de la tabla “test_area00” sin índice

En la figura 4.22 se observa un rectángulo rojo que muestra que no se configura ningún tipo de índice en la tabla "test_area00", después de terminar la edición de la tabla en el sistema de administración de base de datos se utiliza la sintaxis de SQL de la figura 4.23 para crear la tabla.

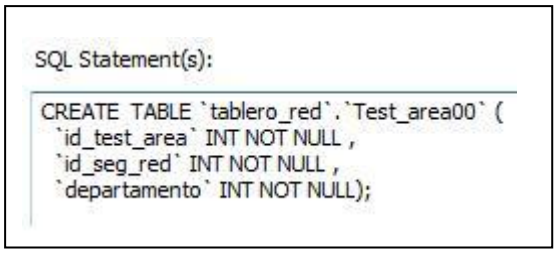


Figura.4.23 – Sintaxis de SQL para crear la tabla "test_area00" sin llave primaria y sin índice en la base de datos en el SABD

En la figura 4.23 se observa la sintaxis de SQL propias del SABD para la creación de la tabla "test_area00", sin llave primaria y sin índice, cuando se ejecuta esta sintaxis correctamente aparece el mensaje de ejecución exitosa como se puede ver en la figura 4.24.

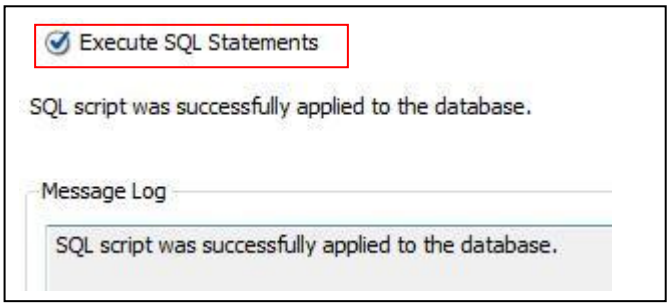


Figura.4.24 – Mensaje de creación exitosa en el SABD de la tabla "test_area00"

En la figura 4.24 se encierra en un rectángulo de color rojo el mensaje de ejecución satisfactoria, lo que quiere decir que la tabla sea creado correctamente y esta lista para almacenar información.

b) Con llave.

Para el desarrollo de este punto se crea una tabla nueva similar a la del punto anterior con la diferencia de que tiene un campo identificado como llave primaria (PK) y que se auto incrementa (AI).

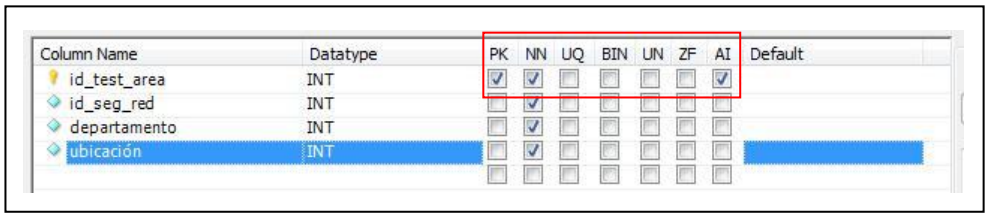


Figura.4.25 - Edición de la tabla “test_area01” con llave primaria y auto incremento.

En la figura 4.25 se observar un rectángulo de color rojo que indica que se activa la casilla de llave primaria y auto incremento en la tabla “test_area01”, después de activar las propiedades de la tabla el SABD utiliza la sintaxis de SQL para crear la tabla (fig 4.26).

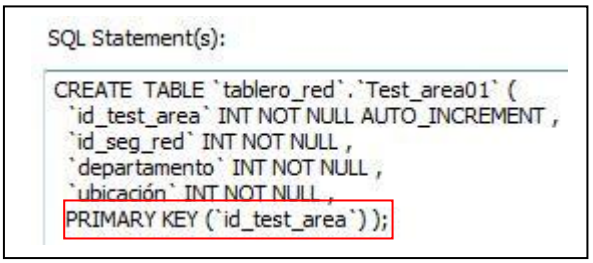


Figura.4.26 - Sintaxis para crear la tabla “test_area01” con llave primaria y sin índice en el SABD

En la figura 4.26 se observa la sintaxis de SQL propias del SABD para la creación de la tabla “test_area01”, con la llave primaria encerrada en un recuadro en color rojo, cuando se ejecuta esta sintaxis correctamente aparece el mensaje de ejecución exitosa como se puede ver en la figura 4.27.

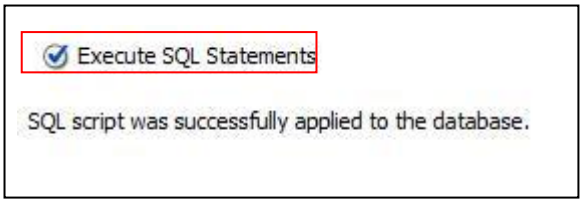


Figura.4.27 - Mensaje de creación exitosa en el SABD de la tabla “test_area01”

En la figura 4.27 se encierra en un rectángulo de color rojo el mensaje de ejecución satisfactoria, lo que quiere decir que la tabla se ha creado correctamente con la llave primaria y esta lista para almacenar información.

c) Con llave y un índice.

Para el desarrollo de este punto se crea una tabla nueva similar a la del punto anterior (fig 4.28) con la diferencia de la generación de un índice.

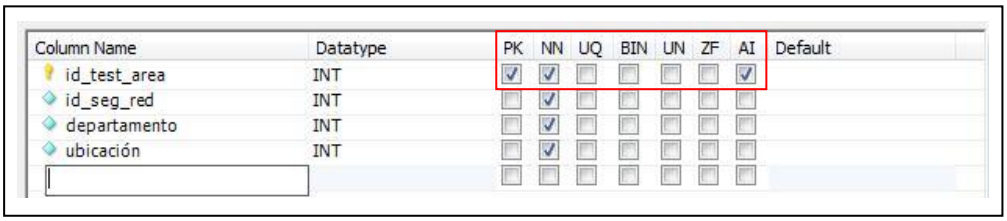


Figura.4.28 - Edición de la tabla "test_area02" con llave primaria y auto incremento

En la figura 4.28 se observar un rectángulo de color rojo que indica que sea activa la casilla de llave primaria y auto incremento en la tabla "test_area02", después se crea el índice como lo muestra la figura 4.29.

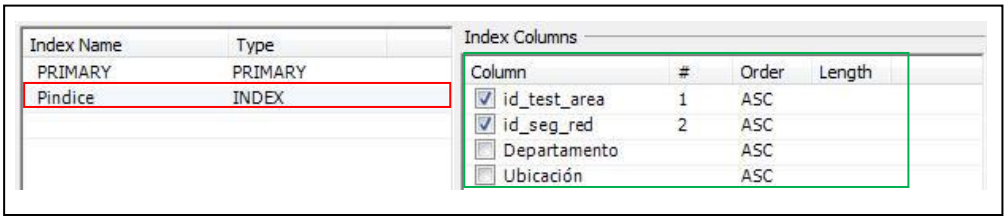


Figura.4.29 - Edición del índice de la tabla "test_area02"

En la figura 4.29 se muestra la ventana para la creación del índice de la tabla "test_area02" encerrado en un recuadro de color rojo y en otro recuadro de color verde se muestra la sección para seleccionar las columnas que contendrá el índice, después se utiliza la sintaxis de la figura 4.30 para crear la tabla con llave primaria e índice.

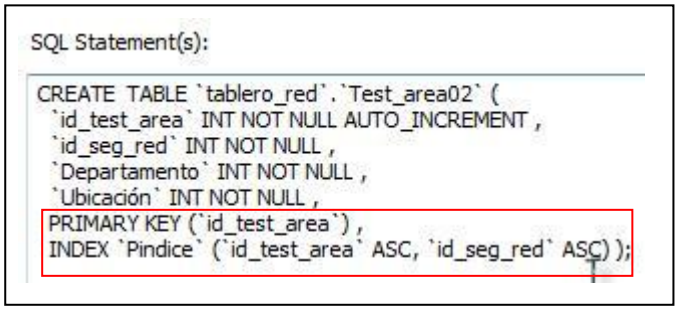


Figura.4.30 - Sintaxis para crear la tabla "test_area02" con llave primaria y con índice en el SABD

En la figura 4.30 se observa la sintaxis de SQL propias del SABD para la creación de la tabla "test_area02", con llave primaria y el índice encerrado en un recuadro en color rojo, cuando la ejecución de la sintaxis de SQL es exitosa se obtiene el resultado de la figura 4.31.

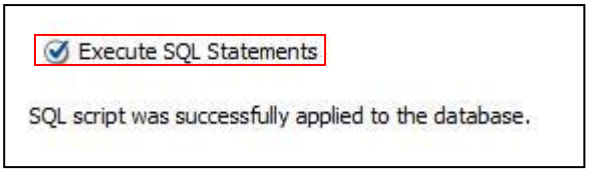


Figura.4.31 - Mensaje de creación exitosa en el SABD de la tabla "test_area02"

En la figura 4.31 se encierra en un rectángulo de color rojo el mensaje de ejecución satisfactoria, lo que quiere decir que la tabla "test_area02"sea creado correctamente con la llave primaria e índice y esta lista para almacenar información.

3) Son cuatro los tipos de carga de datos :

a) Un *insert* por cada registro.

Para el desarrolló de este punto se utiliza la siguiente sintaxis de SQL:

Insert into *Nombre_tabla (nombre_columna,…)*

Values *(expresión,…);*

Esta sintaxis permite insertar un nuevo registro en la tabla [55], lo que quiere decir que para tener una tabla con 1 Millon de registros se tiene que repetir la sintaxis **insert** 1 Millon de veces. El propósito de esta prueba es ver cuanto tiempo tarda el manejador de base de datos (SABD) en hacer esta carga[80].

[80] Esta prueba también se encuentra limitada por las características de la computadora.

Para esta prueba la sintaxis es la siguiente:

Insert into *test_area00(id_area00,id_seg_red,departamento)*
Values *(1,1,1);*

Esta sintaxis se repite un millón de veces y después se ejecuta la inserción en la tabla *test_area00* dentro de la base de datos, dando como resultado un tiempo de respuesta en la carga de los datos de 9 horas.

b) Un *insert* con la mayor carga posible de registros.

Para el desarrolló de este punto se utiliza el comando **insert** pero con una sintaxis diferente, esto permite insertar varios registros al mismo tiempo mejorando el tiempo en la carga de datos, la sintaxis es la siguiente:

Insert into *Nombre_tabla (nombre_columna,…)*
Values *(expresión,…), (expresión,…),…, (expresión,…);*

Esta sintaxis puede insertar varios registros pero tiene un limite, el propósito de esta prueba es ver cual es el limite máximo que soporta el SABD utilizando el comando **insert** y el tiempo que tarda en cargar los datos, con esto se mide la eficiencia del manejador de base de datos.

Para esta prueba la sintaxis es la siguiente:

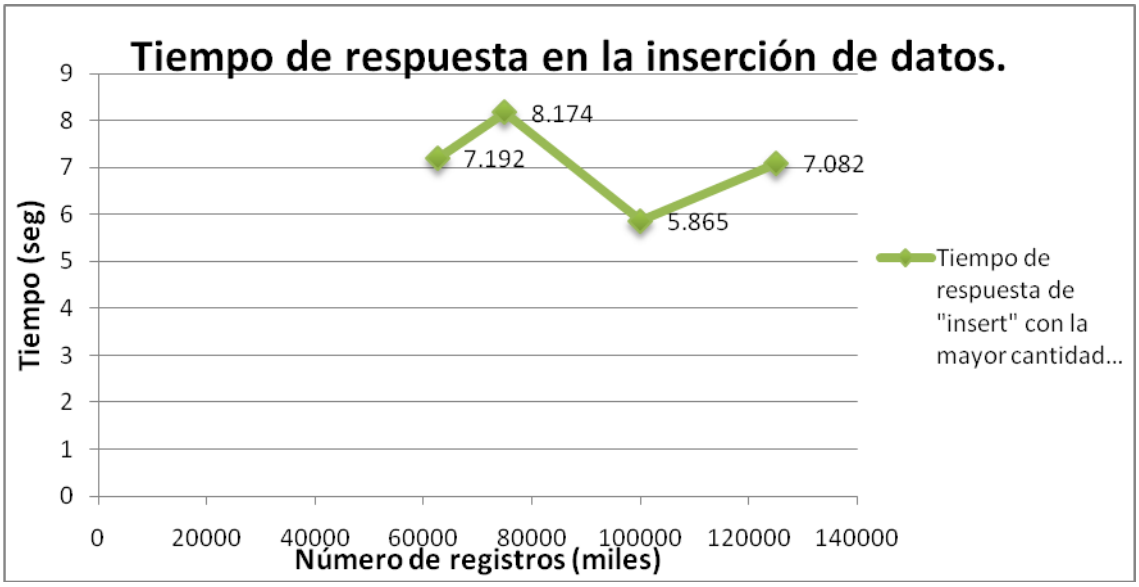
Insert into *test_area01(id_area01,id_seg_red,departamento)*
Values *(1,1,1), (1,1,1), (1,1,1), (1,1,1), (1,1,1), (1,1,1), (1,1,1), (1,1,1), (1,1,1) ,… (1,1,1);*

Como se puede observar en la sintaxis anterior el comando **values** puede contener un número finito de registros para ser insertados en la base de datos, después de ejecutar la sintaxis se obtienen los resultados contenidos en la tabla 4.5.

Nombre tabla	Número de registros (mil)	Tiempo de respuesta (seg)
Test_area01_1	**62,500**	7.192
Test_area01_2	**75,000**	8.174
Test_area01_3	**100,000**	5.865
Test_area01_4	**125,000**	7.082

Tabla 4.5 – Resultados de la prueba de inserción con la mayor cantidad posible de registros.

En la tabla 4.5 se puede observar diferentes tablas con distintas cantidades de registros insertados y cada uno tiene un tiempo de respuesta, esto es por que se pretende encontrar la cantidad máxima de registros que se pueden insertar en una tabla, el resultado es que en la tabla "Test_area01_4" se pudo insertar una cantidad máxima de 125 mil registros, con un tiempo de respuesta de 7.082 segundos, la gráfica 4.1 muestra el comportamiento de esta prueba.



Gráfica 4.1 – Comportamiento de la inserción con la mayor carga de registros con llave primaria.

En la gráfica 4.1 se observa el comportamiento de la inserción de registros con la mayor carga posible en relación con el tiempo en que

se tardo en insertar los registros. Al incrementar el número de registros el SABD envía un mensaje de error indicando que no se pueden insertar más registros como lo indica la figura 4.32.

Figura.4.32 – Mensaje de error del SABD al intentar cargar mas 125 mil registros.

c) Utilizando una expresión SQL propia del sistema de administración de base de datos (SABD) para cargar la mayor cantidad de registros.

La expresión de SQL que se utiliza para resolver este punto es el comando **Load** cuya su sintaxis es la siguiente:

Load data local infile *'dirección y nombre_archivo.txt'*
Into table *nombre_tabla*;

El comando **Load data local infile** lee los registros desde un archivo de texto a una tabla a alta velocidad [55] y el comando **into table** los coloca dentro de la tabla especificada en la base de datos.

Para la prueba de insertar 1 millón de registros la sintaxis de SQL que se utiliza es la siguiente:

Load data local infile
'F:/Jonatan/Archivosdoc/Dr.GilbertoLunaCIC/Ejer_DB/Reg_1M.txt'
Into table *area02_1m;*
Para 5 millones de registros la sintaxis de QSL es la siguiente:
Load data local infile
'F:/Jonatan/Archivosdoc/Dr.GilbertoLunaCIC/Ejer_DB/Reg_5M.txt'
Into table *area02_5m;*
Para 10 millones de registros la sintaxis de SQL es la siguiente:

Load data local infile

'F:/Jonatan/Archivosdoc/Dr.GilbertoLunaCIC/Ejer_DB/Reg_10M.txt
'

Into table *area02_10m;*

Para 20 millones de registros la sintaxis de SQL es la siguiente:

Load data local infile

'F:/Jonatan/Archivosdoc/Dr.GilbertoLunaCIC/Ejer_DB/Reg_20M.txt

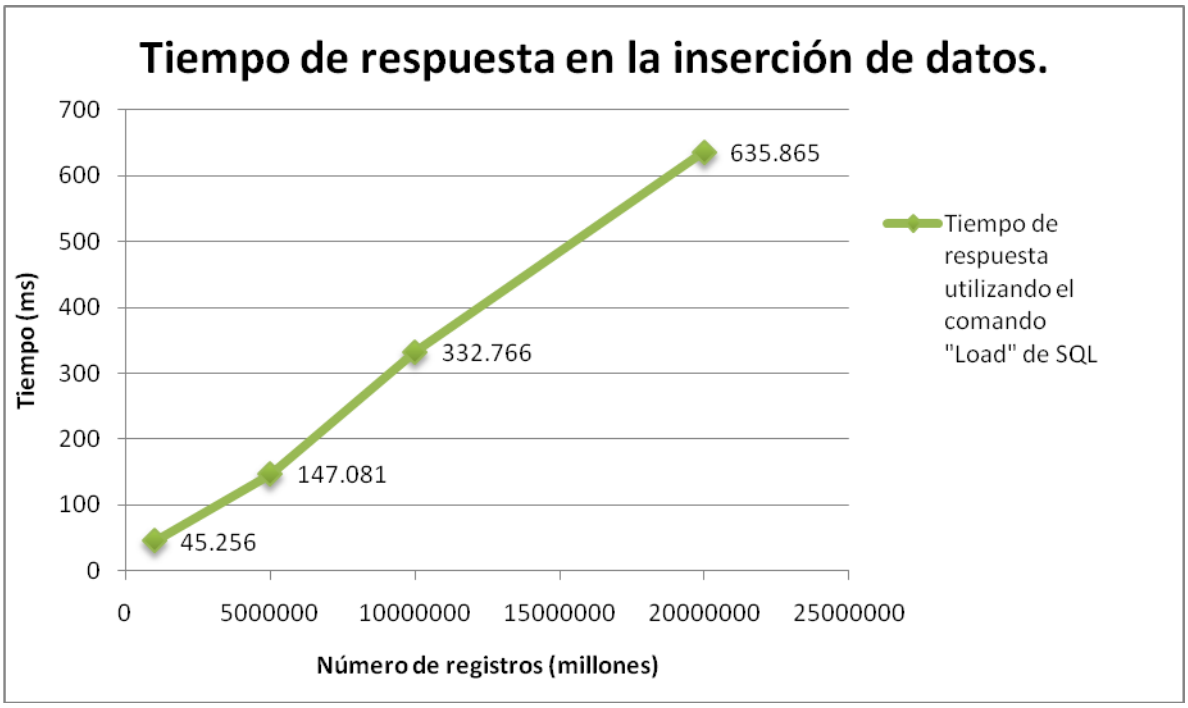
Into table *area02_20m;*

Los resultados de las pruebas para insertar la mayor cantidad de registros utilizando una expresión propia del SABD se observan en la tabla 4.6.

Nombre de la tabla	Número de registros (millones)	Tiempo de respuesta (seg)
Area_02_1M	**1**	45.256
Area_02_5M	**5**	174.081
Area_02_10M	**10**	332.766
Area_02_20M	**20**	653.863

Tabla 4.6 - Resultados de la prueba de inserción con una expresión propia del SADB para la mayor carga de registros

En la tabla 4.6 se puede observar el tiempo que tarada el SADB para insertar registros en la base de datos utilizando una expresión de SQL que es mas rápida al insertar datos que utilizando los dos métodos anteriores con el comando **insert**, la gráfica 4.2 contiene el comportamiento de la prueba.



Gráfica 4.2 – Resultado del comportamiento de la inserción de registros utilizando una expreción propia del SABD.

4.3.2Pruebas de desempeño en consultas de datos.

En esta sección se realizan dos pruebas, la primera prueba es medir el tiempo de respuesta de la consulta a la base de datos creada en la sección 4.3.1 en su pregunta tres, inciso a,b y c, la segunda prueba es medir el tiempo de respuesta con dos tablas que están relacionadas con un campo, estas pruebas tienen la finalidad de medir el tiempo que tarda el SABD en hacer una petición de información a la base de datos para después mostrarla en la pantalla, las pruebas son las siguientes:

1. Pruebas de desempeño en consulta con una sola tabla
    a. Son tres estados de tabla.
        i. Sin llaves y sin índices

        Para resolver este punto se realiza una consulta a la tabla “test_area00” de la sección 4.3.1 con un millón de registros, utilizando la siguiente sintaxis de SQL:

Select *

From *nombre_tabla;*

La sintaxis que se utiliza para solicitar información de la tabla "test_area01[81]", es la siguiente:

Select *

From *test_area01;*

El tiempo de respuesta de la consulta a la tabla "test_area00" con un millón de registros es de 7.702 segundos.

ii. Con una llave

Para el desarrolló de este punto, se realiza una solicitud de información a la base de datos de cada una de las tablas de la sección 4.3.1 inciso b, utilizando la siguiente sintaxis de SQL:

Select *

From *nombre_tabla;*

La sintaxis que se utiliza para solicitar información de la tabla "test_area01_1", es la siguiente:

Select *

From *test_area01_1;*

Para la tabla "test_area01_2", es la siguiente:

Select *

From *test_area01_2;*

Para la tabla "test_area01_3", es la siguiente:

Select *

From *test_area01_3;*

Para la tabla "test_area01_4", es la siguiente:

Select *

From *test_area01_4;*

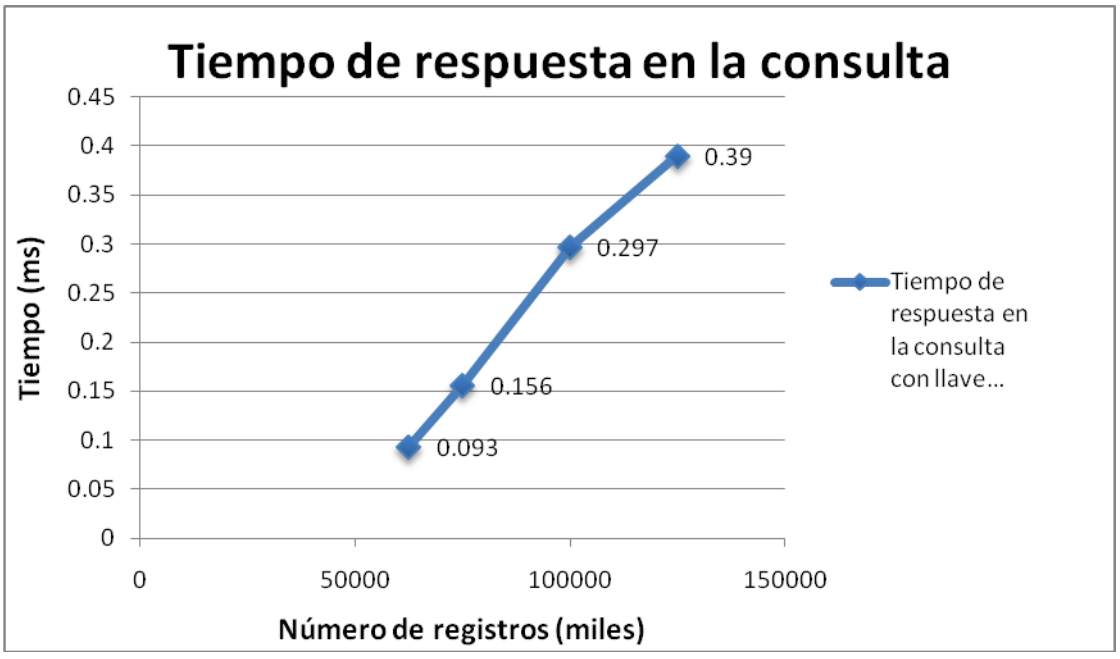
Los resultados de las pruebas de consulta se encuentran en la tabla 4.7

[81] Sección 4.3.1, pregunta tres, inciso a.

Nombre tabla	Número de registros (mil)	Tiempo de respuesta (ms)
Test_area01_1	**62,500**	93
Test_area01_2	**75,000**	156
Test_area01_3	**100,000**	297
Test_area01_5	**125,000**	390

Tabla 4.7 – Resultados de la prueba de consulta con la mayor cantidad posible de registros.

En la tabla 4.7 se puede observar que el tiempo de respuesta en la consulta a cada una de las tablas varia con respecto al número de registros que contiene cada tabla, el comportamiento de esta prueba se presenta en la gráfica 4.3.



Gráfica 4.3 – Comportamiento de la consulta con la mayor carga de registros y con llave primaria.

iii. Con una llave y un índice

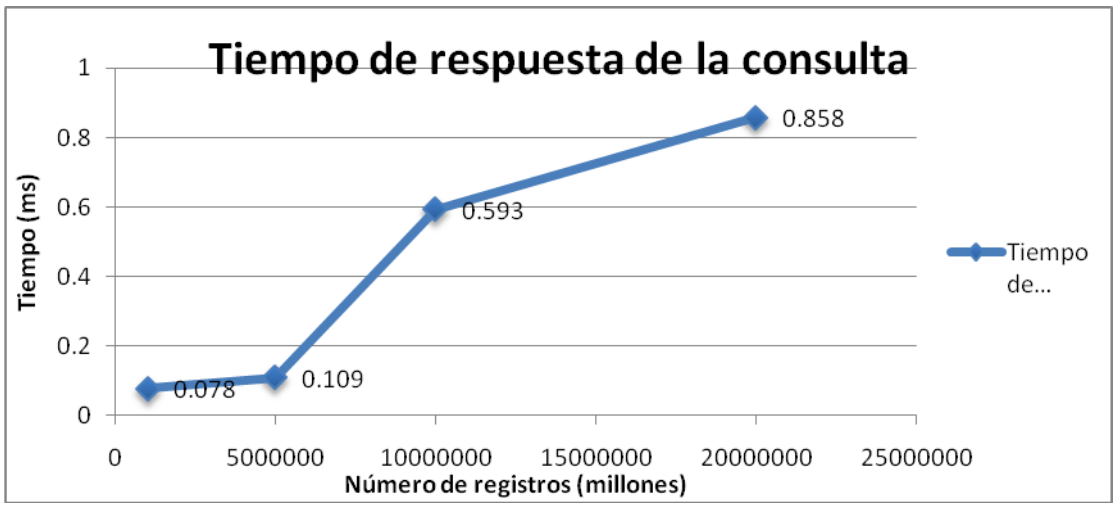
En esta prueba se realiza la consulta con la misma sintaxis de SQL utilizada en el punto anterior, y de igual forma se realiza una consulta

a cada una de las tablas contenidas en la base de datos como lo muestra la tabla 4.8.

Nombre de la tabla	Número de registros (millones)	Tiempo de respuesta (ms)
Area_02_1M	**1**	78
Area_02_5M	**5**	109
Area_02_10M	**10**	593
Area_02_20M	**20**	858

Tabla 4.8 - Resultados de prueba de consulta con una expresión propia del SADB para la mayor carga de registros.

En la tabla 4.8 se puede observar el nombre de la tabla, el número de registros que contiene cada una de las tabla y el tiempo que tardo el SADB para solicitar la información y presentarla en pantalla, el comportamiento de la consultas se encuentra en la gráfica 4.4.



Gráfica 4.4 - Comportamiento en la consulta de registros utilizando una expreción propia del SABD.

En la gráfica 4.4 se puede observar que los tiempos de respuesta en la consulta son pequeños considerando el número de registros que contiene cada una de las tablas, esto quiere decir que la base de datos del tablero de control para el análisis de tráfico de red de área local puede almacenar la información del comportamiento de red y de los equipos que se encuentran instalados en la infraestructura de red, ademas que el tiempo en se presente esta información al administrador de red es minima.

2. Pruebas de desempeño en consultas con dos tablas que están relacionadas con al menos un campo (K= mil registros)

   a. El número de registros de la tabla 1 (sin llave primaria) es de 1K y los de la tabla 2 (con llave primaria) son 10K, 20K, 50K y 100K.

      Para desarrollar este punto se crearan distintas tablas de la misma forma que en la sección 4.3.1 pregunta 1, por que el número de registros es distinto al de las pruebas anteriores, la distribución de las tablas se realiza como lo indica la tabla 4.9.

	Nombre de tabla	Cantidad de registros (k = mil)
Tabla1	Área	1K
Tabla2	Equipo_10K	10K
	Equipo_20K	20K
	Equipo_50K	50K
	Equipo_100K	100K

Tabla 4.9 - Nombre y distribución de tablas con respecto al numero de registros

En la tabla 4.9 se observa que la tabla área (tabla 1) contiene mil registros y la tabla equipo (tabla 2) contiene distintas cantidades de registros que van desde 10 mil hasta 100 mil, ambas tablas tienen un campo en común (id_seg_red), con la creación de estas tablas se cumple con los requisitos para realizar las pruebas de consulta.

b. La tabla 1 sin llave primaria y la tabla 2 con llave primaria, realizar la consulta de junta natural.

Para el desarrolló de este punto se utiliza el comando **Join** que realiza un producto cartesiano entre dos tablas que están relacionadas con al menos un campo eliminando la repetición de registros (redundancia de información) en la tabla resultante, la sintaxis de SQL es la siguiente:

Select *columna1,clumna2…*
From *nombre_tabla1* **Join** *nombre_tabla2*
Where *Condición.*

La sintaxis que se utiliza para solicitar información de la tabla1 con 1K y la tabla 2 con 10K es la siguiente:

Select *id_area, id_equipo, departamento, tipo, estado*
From *area* **Join** *equipo_10K*
Where *area.id_seg_red = equipo_10K.id_seg_red;*

Para la tabla 1 con 1K y la tabla 2 con 20K la sintaxis es la siguiente:

Select *id_area, id_equipo, departamento, tipo, estado*
From *area* **Join** *equipo_20K*
Where *area.id_seg_red = equipo_20K.id_seg_red;*

Para la tabla 1con 1K y la tabla 2 con 50K la sintaxis es la siguiente:

Select *id_area, id_equipo, departamento, tipo, estado*
From *area* **Join** *equipo_50K*
Where *area.id_seg_red = equipo_50K.id_seg_red;*

Para la tabla 1 con 1K y la tabla 2 con 100K la sintaxis es la siguiente:

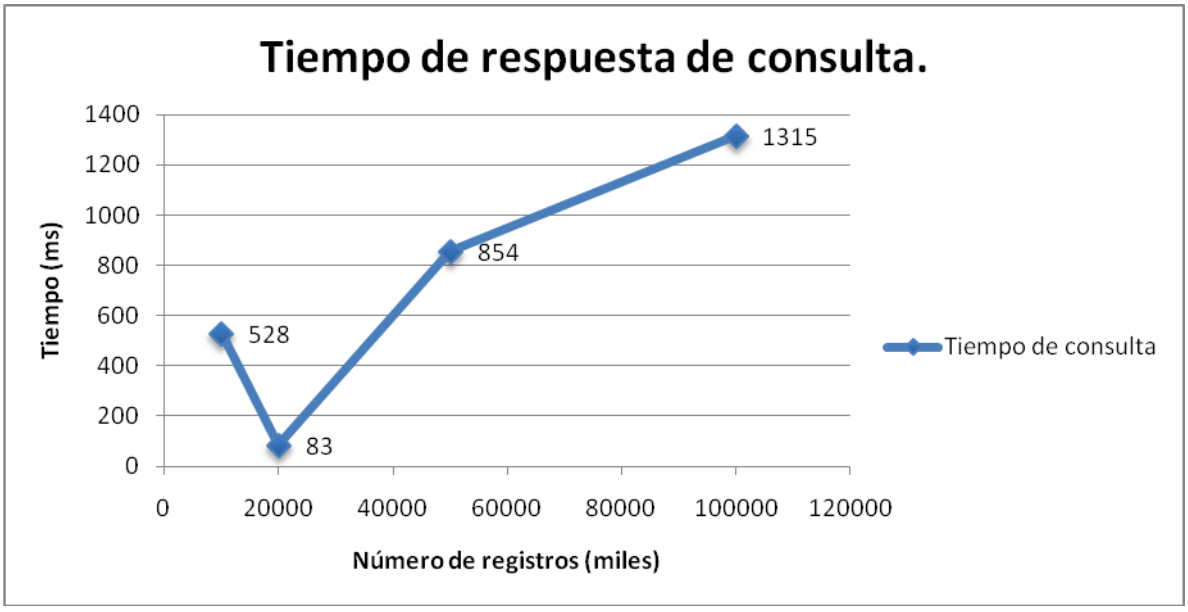
Select *id_area, id_equipo, departamento, tipo, estado*
From *area* **Join** *equipo_100K*
Where *area.id_seg_red = equipo_100K.id_seg_red;*

El resultado de los tiempos de respuesta de cada una de las consultas se encuentra en la tabla 4.10.

Número de registros (K = mil)	Tiempo de respuesta (ms)
1K	528
20K	83
50K	854
100K	1315

Tabla 4.10 Resultados de la prueba de consulta de dos tablas relacionadas con un campo.

En la tabla 4.10 se observa la relación entre la cantidad de registros y el tiempo en que se realiza la petición y la presentación de información en la consulta de dos tablas que están relacionadas con un campo; el comportamiento de estas consultas se presenta en la gráfica 4.5.



Gráfica 4.5 – Comportamiento de las consultas con dos tablas relacionadas con un campo, tabla 1 sin llave primaria y tabla 2 con llave primaria.

En la gráfica 4.5 se observa el comportamiento de las consultas utilizando el comando **Join** con dos tablas que tienen un campo en común, se observan los tiempos que consume cada una de las consultas realizadas a las tablas contenidas en la base de datos y se observa que el tiempo que tarda en presentar la información en el SABD es pequeño considerando el número de registros.

4.3 Pruebas del tablero de control.

En esta sección se agrega cada uno de los módulos probados en las secciones anteriores, además de otros módulos como los medidores de paquetes de datos, medidor de consumo de ancho de banda, medidor de equipos activos en la red que se incluyen en el tablero del control para análisis de red en una área local (fig 4.33).

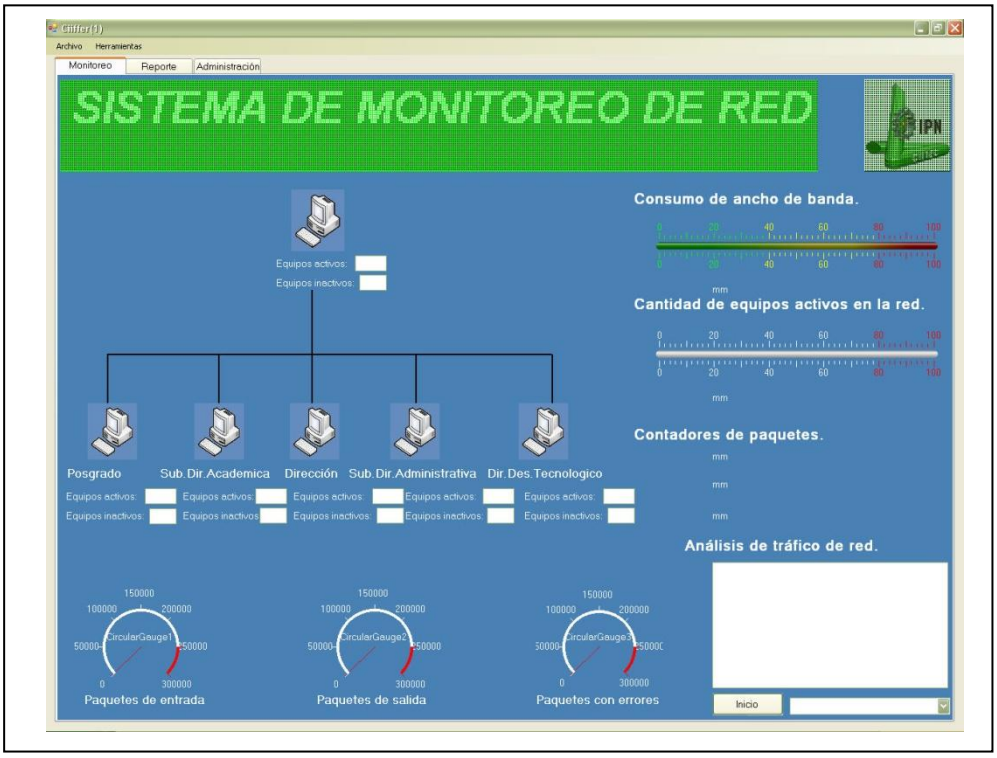


Figura 4.33 – Vista completa del tablero de control para análisis de red en una red local.

En la sección 4.4.1 se presentan las pruebas a la barra de menú, la sección 4.4.2 explica las pruebas de la pestaña monitoreo, la sección 4.4.3 presenta las pruebas a la pestaña reporte y la sección 4.4.4 muestra las pruebas a la pestaña administración.

4.3.1 Prueba de la Barra de menú.

Para esta prueba no es necesario que se active el botón de inicio / alto del tablero de control, los casos para realizar esta prueba se describen en los siguientes puntos:

1) Active el tablero de control (fig 4.33)

2) Posicione el cursor encima del menú archivo, active el menú para que se despliegue la opción salir (fig 4.34).

Figura 4.34 – Resultado de la activación del menú archivo.

En la figura 4.34 se puede observar el resultado de la activación del menú archivo, si se activa el comando salir dentro de este menú el tablero de control se cerrara.

3) Posicione el cursor encima del menú herramientas, active el menú para que se desplieguen sus opciones (fig 4.35)

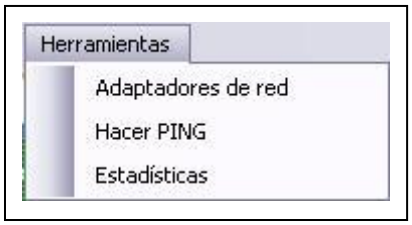


Figura 4.35 – Resultado de la activación del menú herramientas.

En la figura 4.35 se observa como resultado el despliegue de las funciones adaptadores de red, HacerPING y Estadísticas cuando se activa el menú herramienta, en los siguientes puntos se realiza las pruebas para verificar que las aplicaciones funcionen al ser activadas dentro del menú herramientas:

a. Active la función adaptadores de red para abrir la aplicación (fig 4.36)

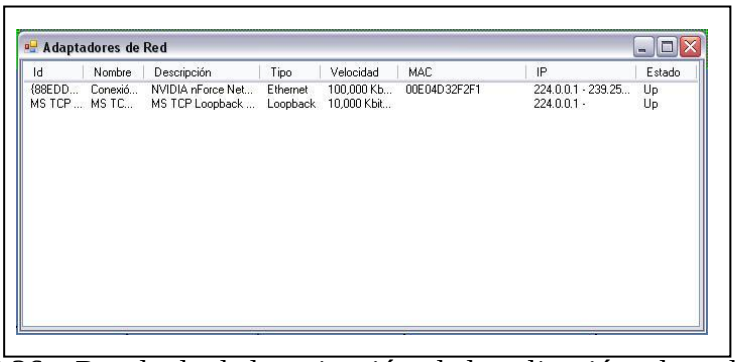


Figura 4.36 – Resultado de la activación de la aplicación adaptadores de red.

La figura 4.36 presenta el resultado al activar la aplicación adaptadores de red, dando como resultado la aparición de la ventana con la detección del los adaptadores de red instalados en el equipo de computo.

b. Active la función HacerPING para activar la aplicación (fig 4.37).

Figura 4.37 – Resultado de la activación de la aplicación HacerPING

La figura 4.37 muestra el resultado al activar la aplicación HacerPING, obteniendo el despliegue de la aplicación en pantalla.

c. Active la función Estadísticas para abrir la aplicación (fig 4.38).

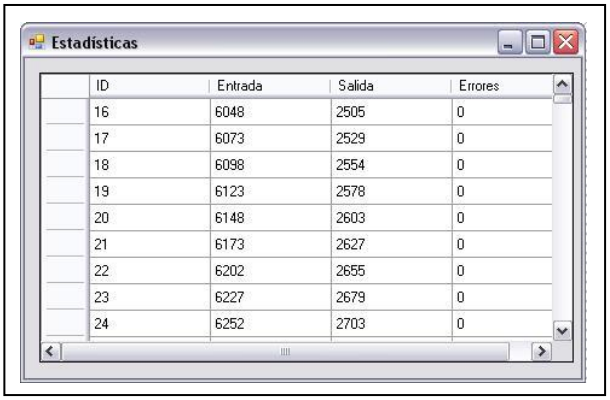


Figura 4.38 – Resultado de la activación de la aplicación Estadísticas

En la figura 4.38 se observa el resultado cuando se activa la función Estadísticas, dando como resultado el llamado a la aplicación para que se pueda desplegar en pantalla, arrojando la información de los paquetes de entrada, salida y con errores que se obtienen de la red.

4.3.2 Prueba de la pestaña monitoreo.

Para realizar esta prueba es necesario tener un cable UTP en el puerto RJ45 del adaptador de red, el proceso de estas pruebas se describen en los siguientes puntos:

1) Seleccione el nombre o la dirección IP asociada a su adaptador de red, dentro de la sección adaptadores.

2) Active el botón inicio/alto, cuando se active el botón su estado cambiara a alto.

3) Prueba obtención de información en la sección análisis de tráfico de red (fig 4.39).

Figura 4.39 – Resultados de la prueba de la sección análisis de tráfico de red.

En la figura 4.39 se puede observar el resultado de la prueba de la sección de análisis de red en donde se detecta la dirección IP origen y destino, también se observa la detección del tipo de protocolo utilizado en la comunicación que en este caso es TCP/IP.

4) Prueba detección del número de protocolos TCP, UDP y desconocidos en la sección de contadores de paquetes (fig 4.40).

Figura 4.40 – Resultado de la prueba de la sección contadores de paquetes

La figura 4.40 muestra el resultado de la obtención del número de paquetes de TCP, UDP y desconocidos que se obtiene del paquete de datos, cada vez que se detecta un protocolo se incrementa el contador correspondiente.

5) Prueba obtención del porcentaje y el número de equipos activos en la red (fig 4.41).

Figura 4.41 – Resultado de la prueba de la sección cantidad de equipos activos en la red.

En la figura 4.41 se presenta el resultado de la obtención del porcentaje y el número de equipos activos en la red, se observa que el numero de equipos activos en el momento son 9, estos son equipos pertenecientes al segmento de red que se configura en la pestaña de administración, la barra de progreso despliega el porcentaje que representa el numero de equipos activos en el momento.

6) Prueba detección del consumo de ancho de banda y la velocidad de transmisión (fig 4.42).

Figura 4.42 – Resultado de la prueba de la sección consumo de ancho de banda.

En la figura 4.42 de presenta el resultado de la sección consumo de ancho de banda, en donde se puede observar que se tiene un 20% del consumo de ancho de banda en el momento de la prueba, también se muestra la velocidad que es de 12552.82 bits/s con la que se están viajando los paquetes de datos en el medio de transmisión.

7) Prueba Detección de paquetes de entrada, salida y con errores (fig 4.43)

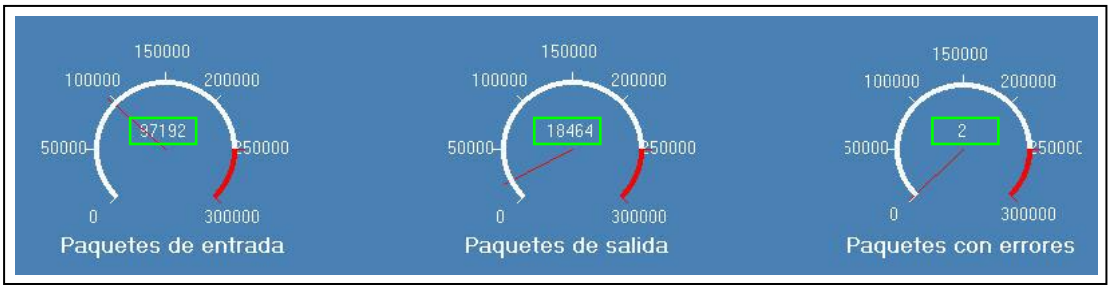


Figura 4.43 - Resultado de los medidores de paquetes de entrada, salida y errores

La figura 4.43 muestra el resultado de la detección de la cantidad de paquetes de entrada, salida y con errores que se obtiene de la red en ese momento, en donde se observa tanto el número de paquetes (encerrado en recuadro de verde), la escala (de color blanco) y el nivel de alerta (color rojo) al tener una cantidad grande de paquetes.

8) Detección de los equipos activos por departamento (fig 4.44)

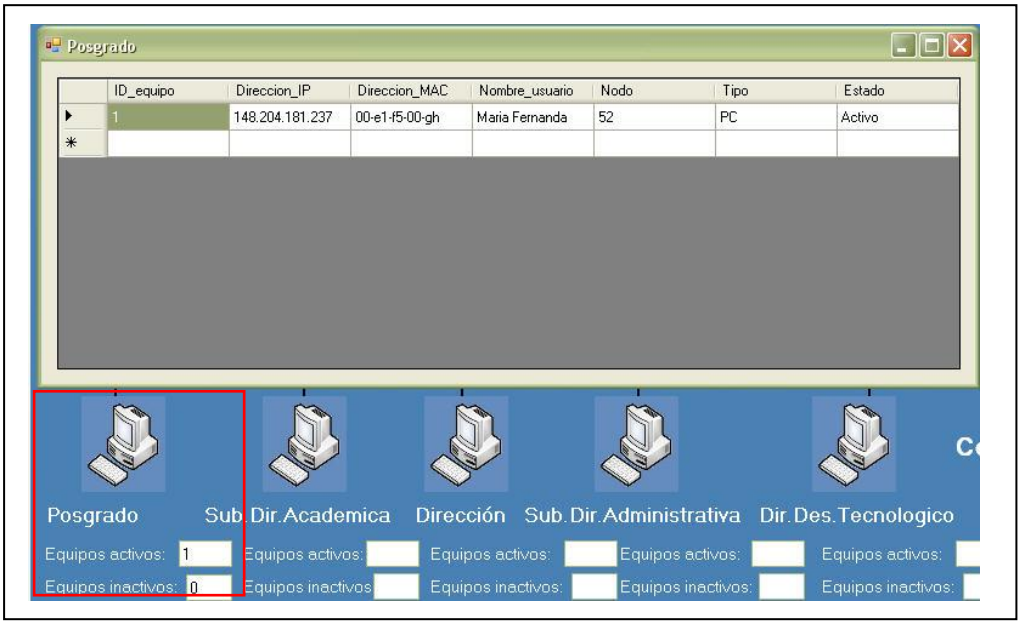


Figura 4.44 - Resultado de detección de equipos por departamento.

La figura 4.44 muestra la detección de un equipo en el departamento de posgrado, se observa en la sección de equipos activos el numero "1" y en la sección de equipos inactivos el numero "cero" (enmarcado en un recuadro de color rojo), también se observa en la figura 4.44 una ventana que realiza una consulta a la base de datos y despliega características como ID_equipo, Dirección_IP, Dirección_MAC, Nombre_usuario, Nodo,Tipo y Estado del equipo detectado, cabe mencionar que los equipos son dados de alta en la pestaña administración y en esta sección solo se determina el estado activo o inactivo.

4.3.3 Pruebas a la pestaña reporte.

De igual forma que en la sección 4.2.2 se tiene que tener un cable UTP conectado al puerto RJ45 del adaptador de red para la obtener información como dirección IP y tipo de protocolo (TCP,UDP y Desconocido), también debe estar activado el botón inicio/alto por que de lo contrario esta pantalla permanece bloqueada (Fig 4.45), las pruebas que se realizan a esta sección del tablero de control se describen en los siguientes puntos:

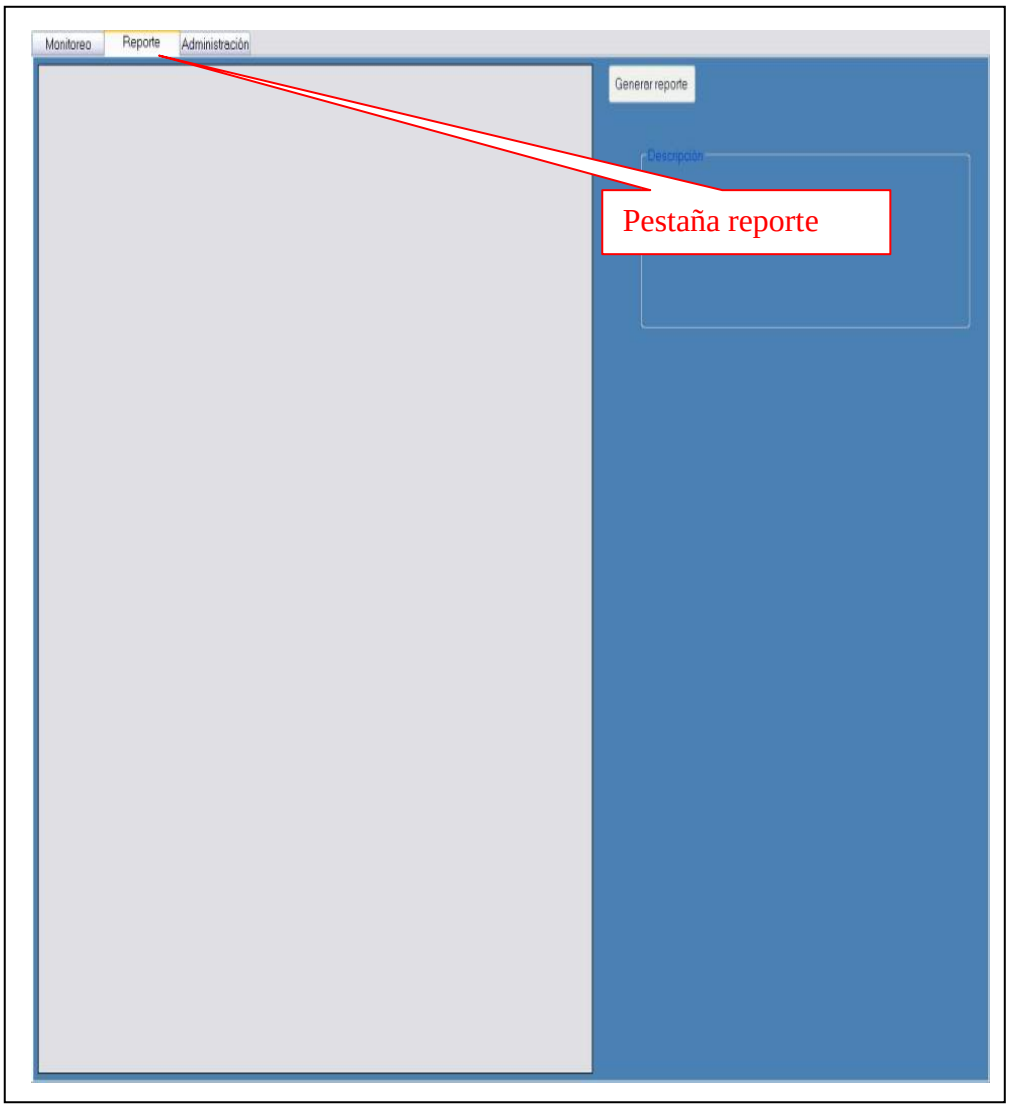
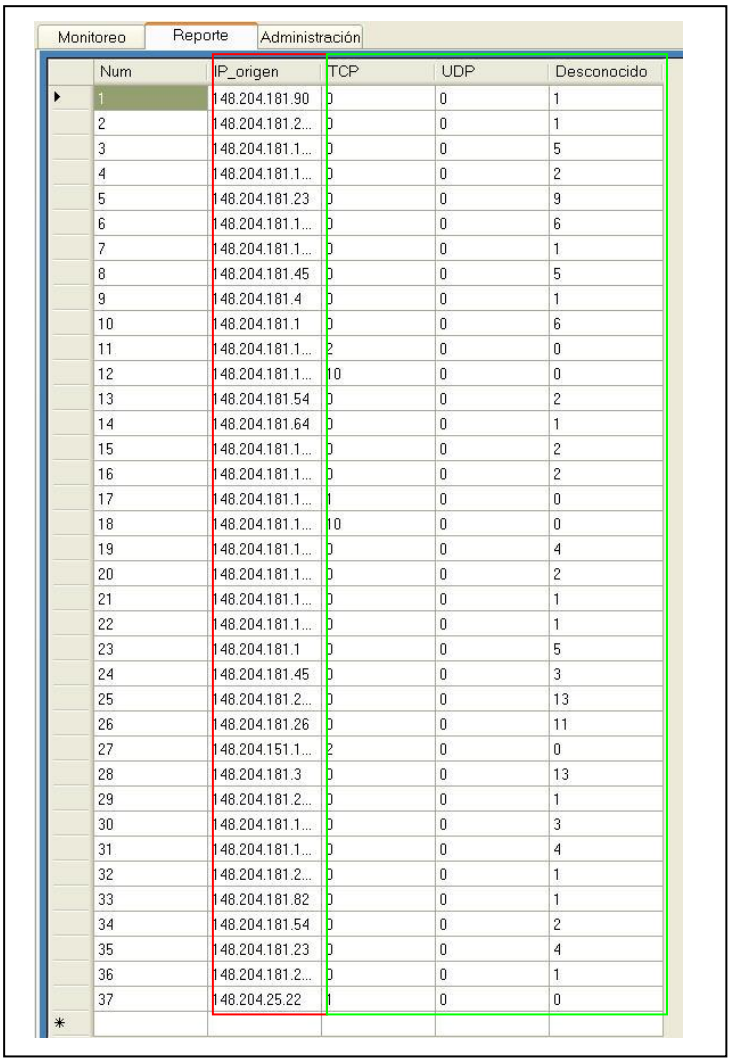


Figura 4.45 - Pestaña reporte del tablero de control.

La figura 4.45 presenta la pestaña de reporte, la cual se encuentra bloqueada si el botón de inicio/alto de la pestaña de monitoreo no esta activo, se diseño de esta manera para que el usuario no piense que el programa falla si no se despliega información en esta ventana cuando el botón inicio/alto esta inactivo.

1) Prueba de detección de direcciones IP origen, protocolos TCP, UDP y desconocidos (fig 4.46), para mostrar el reporte de comportamiento de red.



Num	IP_origen	TCP	UDP	Desconocido
1	148.204.181.90	0	0	1
2	148.204.181.2...	0	0	1
3	148.204.181.1...	0	0	5
4	148.204.181.1...	0	0	2
5	148.204.181.23	0	0	9
6	148.204.181.1...	0	0	6
7	148.204.181.1...	0	0	1
8	148.204.181.45	0	0	5
9	148.204.181.4	0	0	1
10	148.204.181.1	0	0	6
11	148.204.181.1...	2	0	0
12	148.204.181.1...	10	0	0
13	148.204.181.54	0	0	2
14	148.204.181.64	0	0	1
15	148.204.181.1...	0	0	2
16	148.204.181.1...	0	0	2
17	148.204.181.1...	1	0	0
18	148.204.181.1...	10	0	0
19	148.204.181.1...	0	0	4
20	148.204.181.1...	0	0	2
21	148.204.181.1...	0	0	1
22	148.204.181.1...	0	0	1
23	148.204.181.1	0	0	5
24	148.204.181.45	0	0	3
25	148.204.181.2...	0	0	13
26	148.204.181.26	0	0	11
27	148.204.151.1...	2	0	0
28	148.204.181.3	0	0	13
29	148.204.181.2...	0	0	1
30	148.204.181.1...	0	0	3
31	148.204.181.1...	0	0	4
32	148.204.181.2...	0	0	1
33	148.204.181.82	0	0	1
34	148.204.181.54	0	0	2
35	148.204.181.23	0	0	4
36	148.204.181.2...	0	0	1
37	148.204.25.22	1	0	0

Figura 4.46 – Resultado de la prueba de detección del comportamiento de red.

En la figura 4.46 se observa la detección de las direcciones IP origen (enmarcadas en un recuadro de color rojo) junto con el tipo de protocolo (TCP,UDP o desconocido) y el número de veces que el paquete de datos utiliza ese protocolo para el envió y recepción de datos (encerrados en un recuadro de color verde).

2) Prueba para generar el reporte de comportamiento de red, para esta prueba se tienen que seguir los siguientes pasos:

   a. Activar el botón generar reporte (4.47).

Figura 4.47 – Botón generar reporte del la pestaña reporte.

b. Prueba de activación de la ventana de reporte que contenga la información del comportamiento de red (4.48).

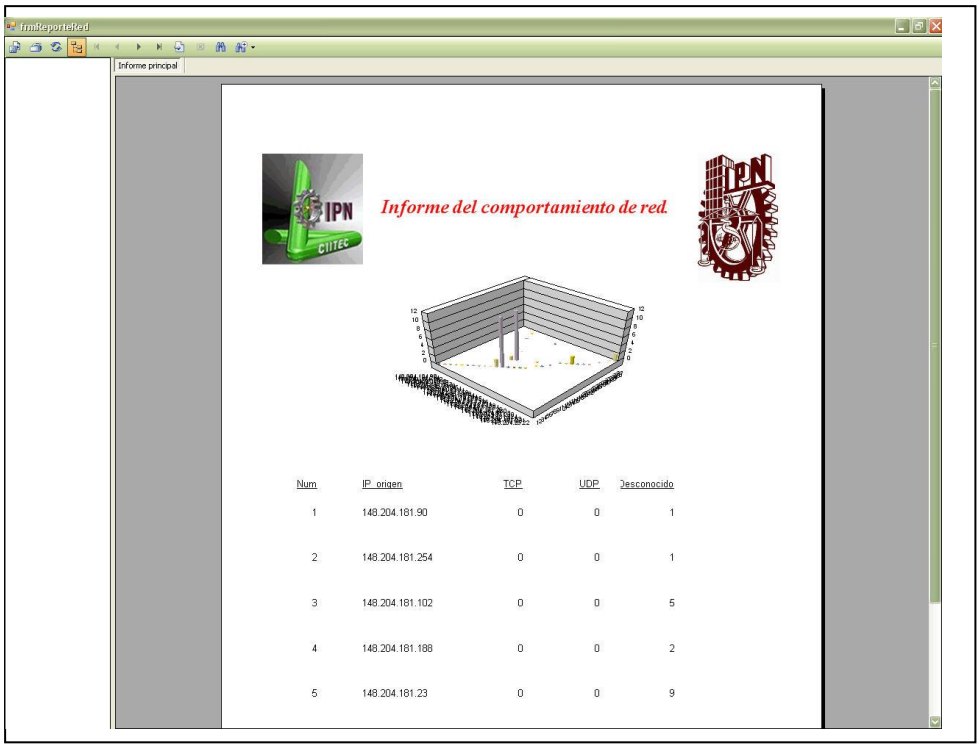
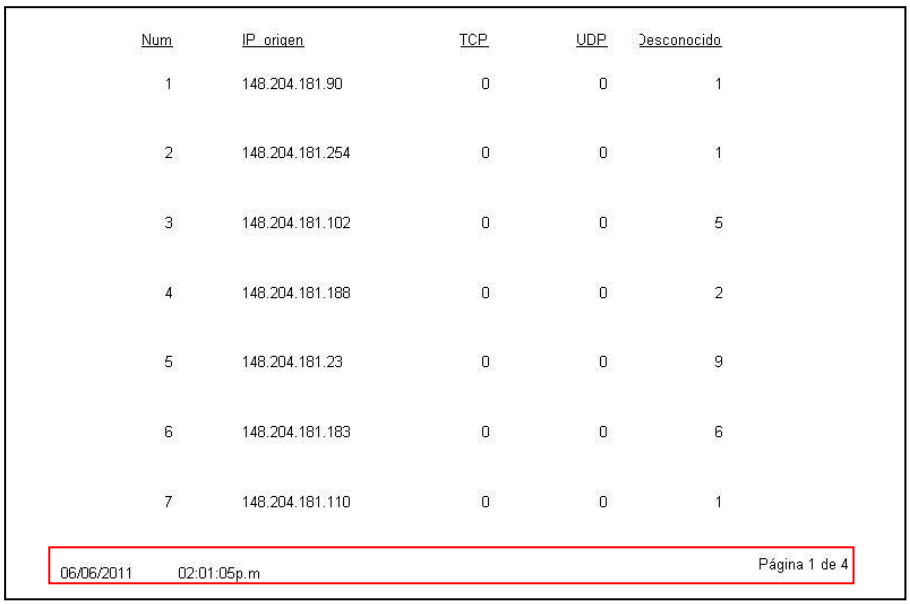


Figura 4.48 - Resultado de la prueba generar reporte.

La figura 4.48 muestra la ventana de reporte en donde se observa los datos del comportamiento de red, en este reporte se inserta la hora, fecha y número de paginas que contiene el reporte (fig 4.49).



Num	IP_origen	TCP	UDP	Desconocido
1	148.204.181.90	0	0	1
2	148.204.181.254	0	0	1
3	148.204.181.102	0	0	5
4	148.204.181.188	0	0	2
5	148.204.181.23	0	0	9
6	148.204.181.183	0	0	6
7	148.204.181.110	0	0	1

06/06/2011 02:01:05p.m Página 1 de 4

Figura 4.49 – Reporte del comportamiento de red

En la figura 4.49 se observa encerrado en recuadro de color rojo la inserción de la fecha, hora y número de páginas que contiene el reporte de red, se diseño de esta manera para evitar la manipulación del reporte por algún usuario no autorizado.

4.3.4 Pruebas a la pestaña administración.

Para esta sección no es necesario que se encuentre activado el botón inicio/alto de la pestaña de monitoreo porque en esta sección es donde el usuario puede realizar configuración de red, alarmas y edición de los nodos de red; las pruebas para esta sección se describen en los siguientes puntos:

1) Comprobación de acceso a usuario con privilegios de administrador, antes de ingresar el nombre de usuario y la contraseña todas las funciones dentro de esta pestaña están bloqueadas (Fig. 4.50).

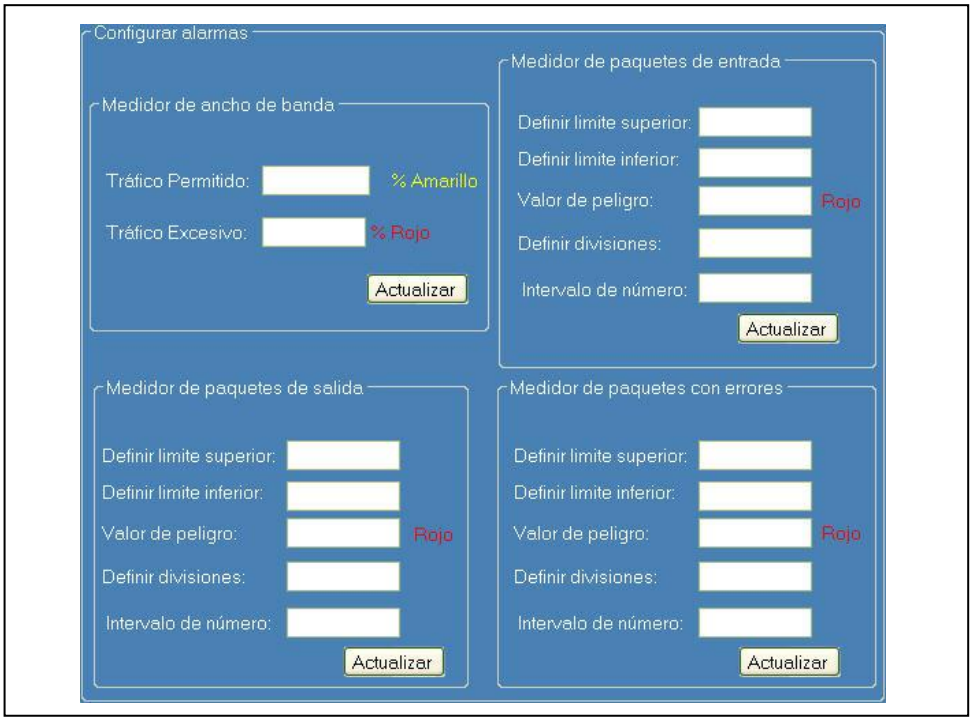


Figura 4.50 - Pestaña de administración

En la figura 4.50 se observa como las funciones para edición de nodo y configuración están bloqueadas, se diseño de esta forma para que solo el usuario con los privilegios de administrador pueda manejar esta pestaña, con esto se disminuye el riesgo que un usuario con poca experiencia realice modificaciones que puedan afectar la administración de red.

2) Prueba de activación de las funciones al acceder como usuario con privilegios de administrador (Fig 4.51).

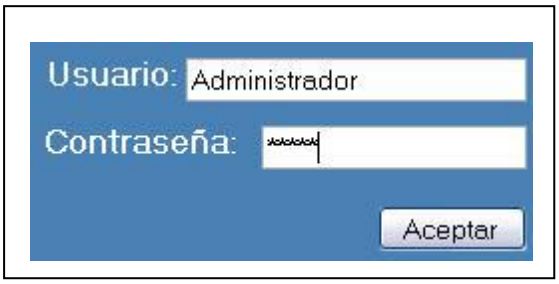


Figura 4.51 – Sección de acceso del administrador.

La figura 4.51 presenta la forma en como el usuario con privilegios de administración escribe su usuario y su contraseña, si esta es correcta se activan

las funciones de la pestaña de administración (fig.4.52) de los contrario permanecerán bloqueadas.

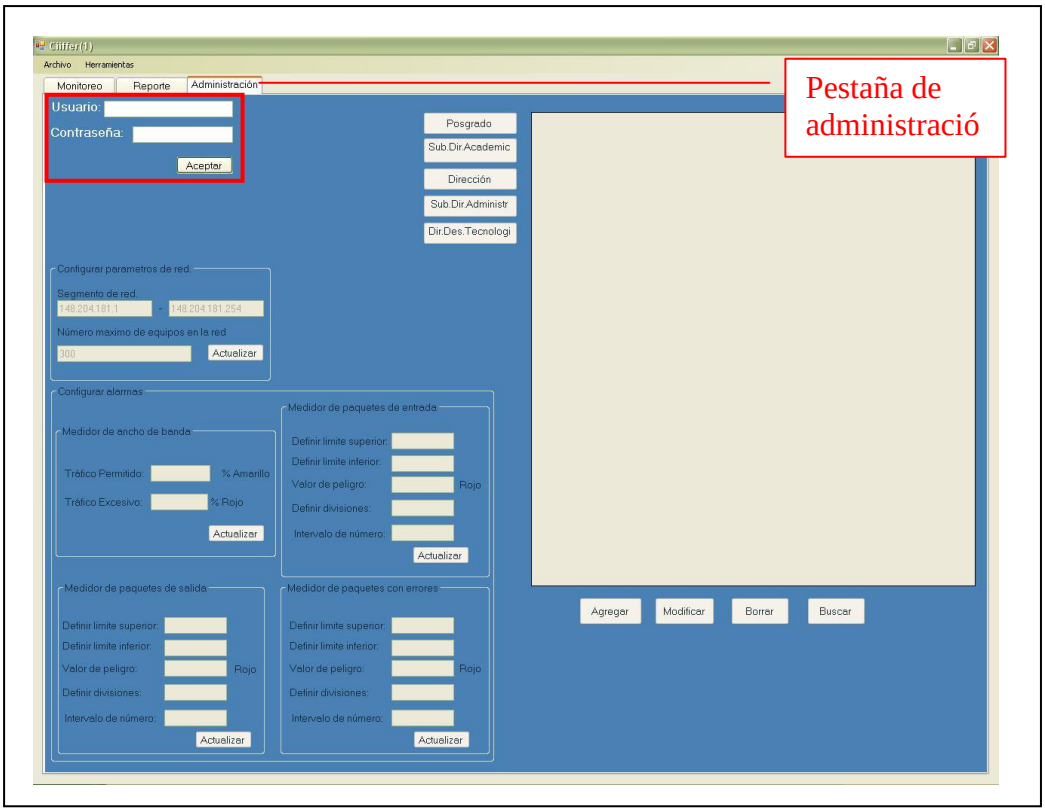


Figura 4.52 - Pestaña de administración activada

En la figura 4.52 se puede observar encerrada en un recuadro de color rojo la sección de acceso para el administrador, cuando el usuario y la contraseña son correctas se bloquea esta sección indicando que el administrador tiene el control del programa, también se observa que las funciones de edición de nodo y configuración de alarmas se encuentran activas.

3) Prueba de despliegue de información como dirección IP, dirección MAC, nombre de usuario, número de nodo, tipo de equipo y estado de cada uno de los departamentos (Fig 4.53).

Figura 4.53 - Botones de selección para el despliegue de información.

La figura 4.53 muestra los botones con los cuales el usuario puede observar la información de cada departamento, para ver la información contenida en posgrado solo se tiene que activar el botón "Posgrado" para desplegar la dirección IP, dirección MAC, nombre de usuario, número de nodo, tipo de equipo y estado (fig 4.54), para despegar la misma información pero de otro departamento solo se tiene que activar el botón se que dese.

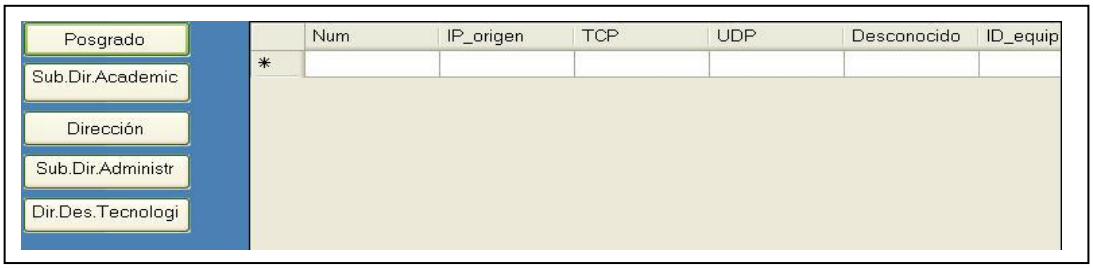


Figura 4.54 – Resultado de la consulta a la tabla posgrado.

En la figura 4.54 presenta el resultado del despliegue de información del nodo de red, se puede observar que no se tiene ninguna información en esta tabla, esto es porque aun no se a editado ningún nodo, en los siguientes puntos se realizaran las pruebas para agregar, modificar, buscar y eliminar nodos.

4) Prueba de inserción de un nuevo nodo en posgrado, para esta prueba se tiene que activar el botón agregar de la pestaña de administración para tener una ventana igual que la figura 4.55, en esta se tienen que llenar todos los campos porque no se pueden insertar campos vacios en la base de datos.

Figura 4.55 – Ventana Agregar nodo.

En la figura 4.55 se observa que ya se llenaron los campos con información del nodo que se insertara en la tabla posgrado, una vez completado el llenado de los campos se activa el botón insertar para agregar un nuevo registro, después se realiza una consulta a la tabla activando el botón posgrado y se desplegara la información como lo muestra la figura 4.56.

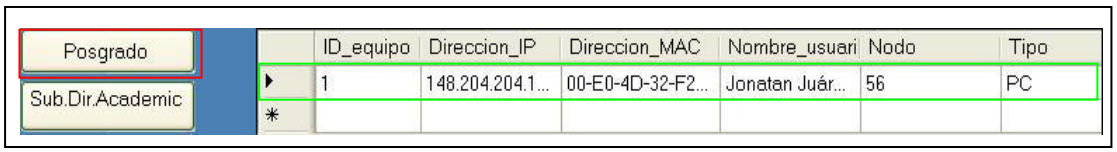


Figura 4.56 – Resultado de la inserción de un nuevo registro en la tabla posgrado.

La figura 4.56 muestra el resultado de la inserción del nuevo registro (encerrado en un recuadro de color verde)cuando se activa el botón posgrado (encerrado en un recuadro de color rojo), este registro ya se encuentra dentro de la base de datos en donde permanecerá hasta que el administrador considere necesario.

5) Prueba de modificación del nodo creado en el punto anterior, en esta prueba se tiene que activar el botón modificar y aparecerá en pantalla una venta, en donde puede seleccionar el área donde se encuentra el nodo a modificar, en este punto

se modifica la dirección IP, el nombre de usuario y el numero de puesto como lo muestra la figura 4.57.

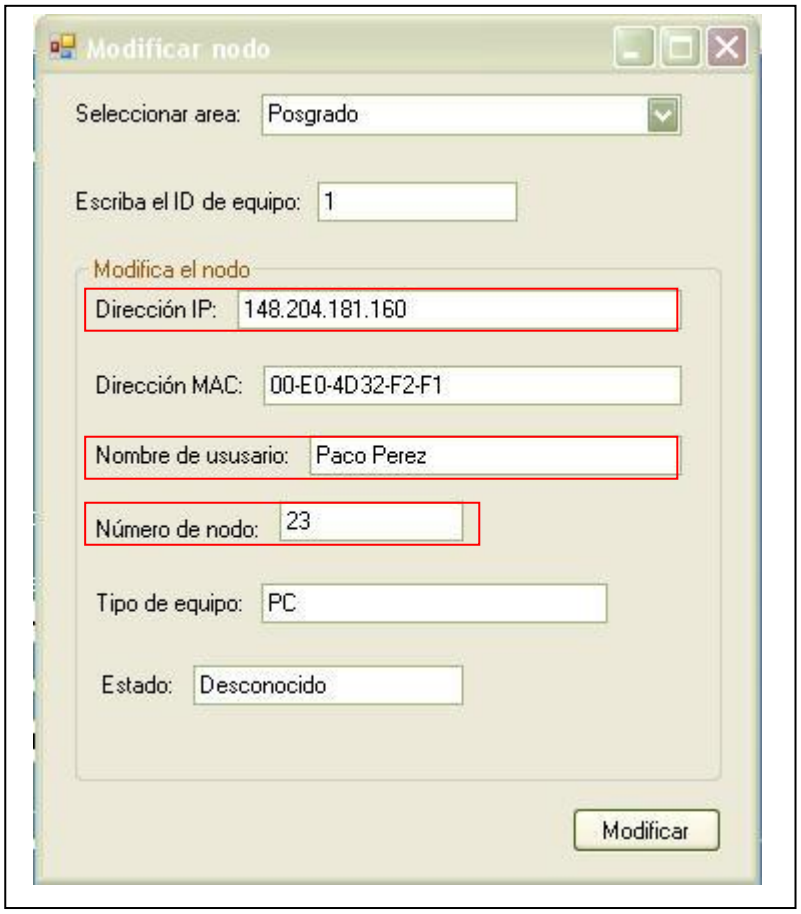


Figura 4.57 – Ventana modificar nodo

En la figura 4.57 aparecen encerrados en recuadros de color rojo los campos que serán modificados dentro de la tabla posgrado, una vez que se llenaron todos los campos se activa el botón modificar y aparecerá un mensaje indicando que la modificación se realizo satisfactoriamente (fig 4.58)

Figura 4.58 – Mensaje de modificación satisfactoria

La figura 4.58 muestra el mensaje de modificación satisfactoria esto quiere decir que se realizo una actualización en el registro de la tabla posgrado, para revisar que la actualización del registro se activa el botón posgrado de la pestaña de administración y se obtendrá el resultado mostrado en la figura 4.69.

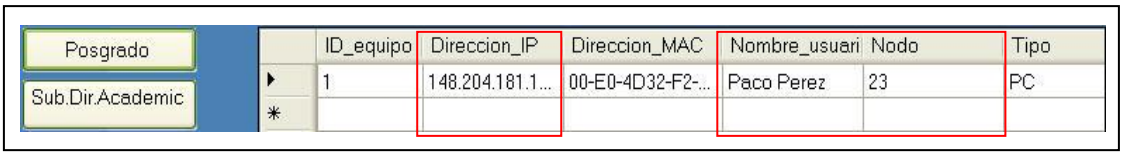


Figura 4.59 – Resultado de la modificación del registro de la tabla posgrado

En la figura 4.59 se observa en cerrados en un recuadro de color rojo los campos que son actualizados dentro de la tabla posgrado.

6) Prueba de búsqueda del nodo creado en posgrado, para esta prueba se tiene que activar el botón buscar dentro de la pestaña de administración para poder observar la ventana de búsqueda, en donde se tiene que seleccionar el área donde se desea buscar el nodo, después se tiene que escribir el valor de búsqueda, a continuación se selecciona el campo y finalmente se activa el botón buscar (fig 4.60).

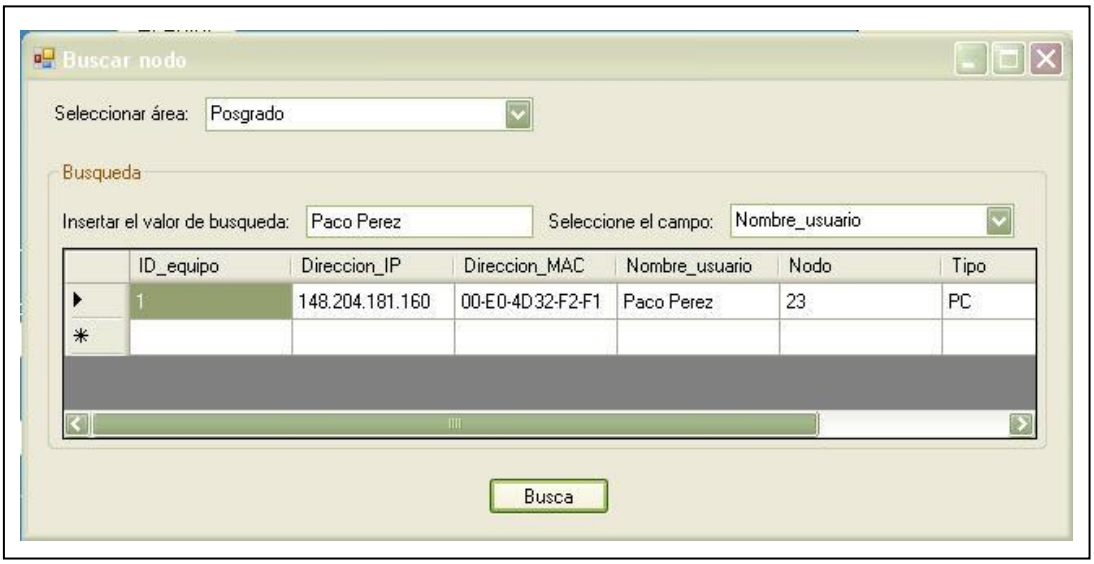


Figura 4.60 – Resultado de la búsqueda de un registro dentro de la base de datos.

La figura 4.60 presenta el resultado de la búsqueda de un registro dentro de la tabla posgrado con el valor de búsqueda Paco Pérez dentro del campo de nombres de usuario, al activar el botón búsqueda se obtiene la información del tipo de equipo que utiliza, cual es su dirección IP, dirección MAC el numero de nodo y el estado.

7) Prueba de borrado del nodo creado en posgrado, en esta prueba se selecciona el área donde se encuentra el nodo que se desea borrar, se escribe el identificador de elemento y se activa el botón eliminar, aparece un cuadro de dialogo indicando que la eliminación del registro es satisfactoria como lo muestra la figura 4.61.

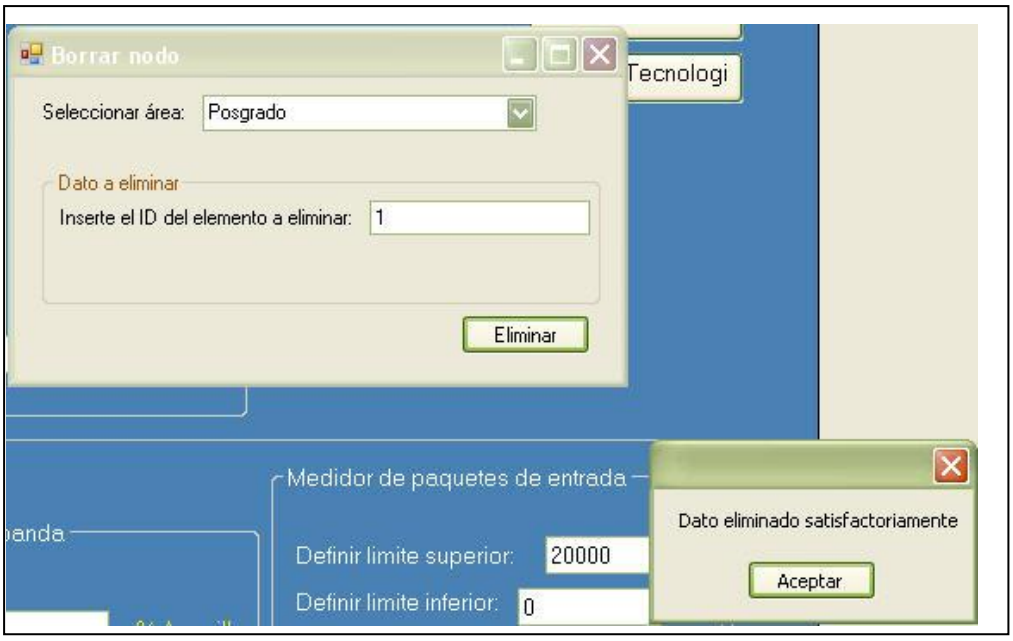


Figura 4.61 – Resultado de la prueba de borrado de un registro dentro de la base de datos.

La figura 4.61 presenta el resultado de borrado del registro con número de identificador 1, se diseño de esta forma por que el identificador es un campo único y se puede diferenciar de los demás campos de la tabla posgrado, también es mas fácil para el usuario recordar solo el área y el número de registro que se desea borrar. Para verificar que el registro ya no se encuentra dentro de la tabla

se activa el botón posgrado de la pestaña administración, se obtendrá la consulta a esta tabla como lo muestra la figura 4.62.

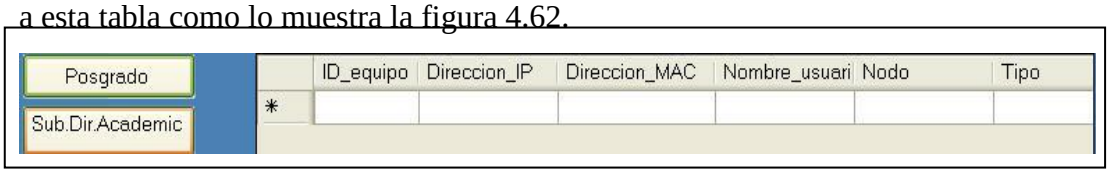


Figura 4.62 – Resultado de la consulta a la tabla posgrado.

En la figura 4.62 se observa que ya no se encuentra el registro insertado en el punto 5 y actualizado en el punto 6.

8) Prueba de configuración de los parámetros de red, en este punto el administrador configura el segmento y el número de equipos que componen la red (fig 4.63).

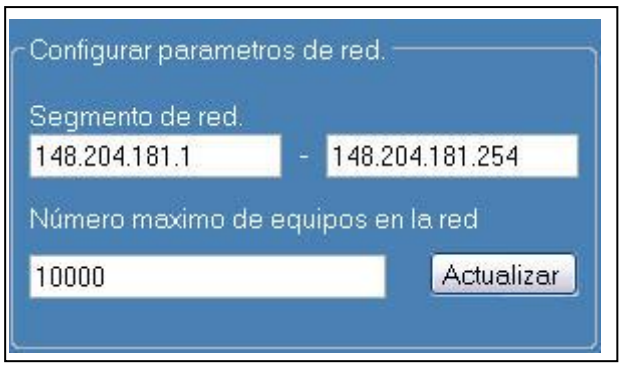


Figura 4.63 Sección de configuración de parámetros de red.

En la figura 4.63 el usuario puede configurar el segmento de red y el numero máximo de equipos con los que cuente su red, estos datos son tomados por el programa cuando se activa el botón “Actualizar” y sirven para el despliegue del porcentaje de equipos activos en red y para llevar la cuenta de los equipos activos que se encuentran dentro del segmento.

9) Prueba de configuración de alarmas, en este punto el administrador configura el porcentaje de tráfico permitido que el considere pertinente y el máximo permitido para generar un evento de alerta (fig 4.64).

Figura 4.64 – Sección medidor de ancho de banda.

En la figura 4.64 se puede observar dentro de la sección de configuración de alarmas, la configuración del medidor de tráfico de red en donde se sustituye los valores predeterminados de tráfico permitido por un valor del 10 y el de tráfico excesivo por un valor de 25, se activa el botón actualizar y el resultado se presenta en la figura 4.65.

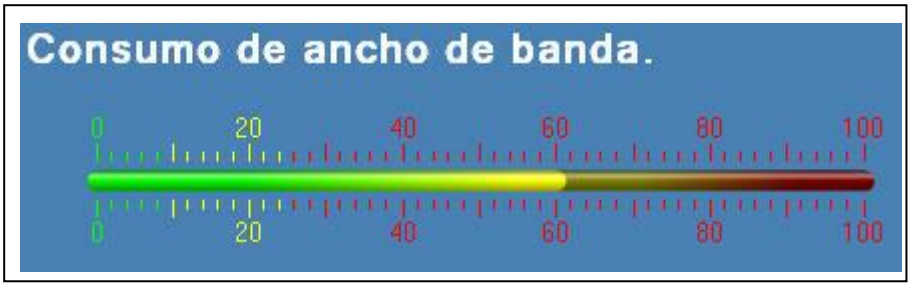


Figura 4.65 – Resultado de la edición de alarma en el medidor de consumo de ancho de banda del medio de transmisión.

La figura 4.65 presenta el resultado de la edición de alarmas del consumo de ancho de banda se puede observar que el nivel bajo de consumo de ancho de banda se encuentra en el intervalo de 0% al 9%, el nivel permitido se encuentra en el rango del 10% al 24% y el nivel de alarma se encuentra del 25% hasta el 100%, se observa en la en la figura 4.65 que al sobrepasa el nivel de alarma, esto dispara un evento que muestra el mensaje como lo muestra la figura 4.66.

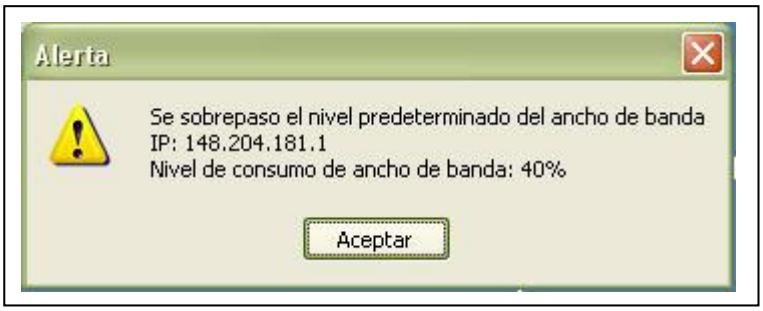


Figura 4.66 – Resultado de la detección del evento de consumo de ancho de banda

En la figura 4.66 se observa el mensaje de alarma cuando se sobrepasa el nivel predeterminado del porcentaje del consumo de ancho de banda, éste indica el porcentaje y la IP que está generando dicho consumo, de igual forma se editan las alarmas para los medidores de paquetes de entrada (fig. 4.66), salida y con errores.

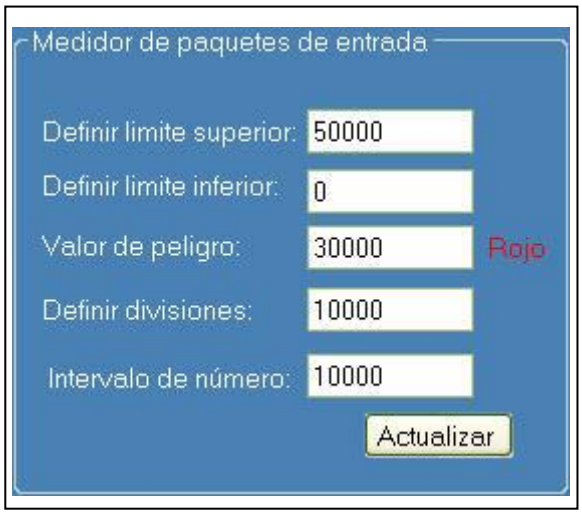


Figura 4.67 – Sección medidor de paquetes de entrada

En la figura 4.67 se muestra la modificación de los valores predeterminados del medidor de paquetes de entrada, el nivel superior indica el valor máximo que tendrá el medidor, el nivel inferior es mínimo valor que se puede alcanzar (Este valor debe ser cero según el fabricante), el valor de peligro es el valor en donde se genera el evento de alarma dentro del medidor, la definición de divisiones es el número de divisiones que aparecerán dentro de la escala del medidor y el intervalo es número que tendrá la escala en cada división dentro del medidor (fig 4.68).

Figura 4.68 – Resultado de la edición de alarma del medidor paquetes de entrada.

La figura 4.68 presenta el resultado de la modificación del medidor de paquetes de entrada, se puede observar que el medidor ya se encuentra alarmado por que el numero de paquetes de entrada son mas que los que se puede medir en la escala predefinida enviando un mensaje de alerta junto con un sonido para llamar la atención del administrador (fig 4.69).

Figura 4.69 – Resultado del la detección del evento de paquetes de entrada.

En la figura 4.69 se observa el mensaje que dispara un evento cuando se sobrepasa el nivel predeterminado del medidor de paquetes de entrada, el mensaje de alarma le indica al usuario la cantidad de paquetes y la dirección IP que los esta originando, la tabla 4.11 presenta los resultados de las pruebas de la interfaz gráfica del tablero de control.

Resultados de las pruebas del tablero de control		
Parámetro	Cumple	No cumple
Detección de trafico de red	✓	
Detección del numero de paquetes de TCP	✓	
Detección del numero de paquetes de UDP	✓	
Detección del numero de paquetes de desconocido	✓	
Detección del numero de computadoras activas en la red	✓	
Detección del consumo de ancho de banda	✓	
Despliegue de la velocidad de transmisión	✓	
Despliegue del porcentaje de equipos conectados en la red	✓	
Detección de la cantidad de paquetes de entrada	✓	
Detección de la cantidad de paquetes de salida	✓	
Detección de la cantidad de paquetes de errores	✓	
Detección de equipos por departamento	✓	
Edición de características de los equipos por departamento		
Agregar nodo	✓	
Modificar nodo	✓	
Borrar nodo	✓	
Buscar nodo	✓	
Configuración de alarmas		
Medidor de ancho de banda	✓	
Medidor de paquetes de entrada	✓	
Medidor de paquetes de salida	✓	
Medidor de paquetes con errores	✓	
Envío de mensaje de alarma	✓	
Sonido de mensaje de alarma	✓	
Comprueba que la dirección IP se encuentra dentro del segmento de red	✓	
Despliega las características de equipos por departamento	✓	
Almacena la información en una base de datos	✓	

Tabla 4.11 – Resultados de las pruebas de los módulos del tablero de control

Capítulo 5

Conclusiones y trabajo a futuro

5.1 Conclusiones.

Se desarrolló una aplicación gráfica, para el análisis de tráfico de red que permite tener un monitoreo continuo del consumo del ancho de banda en una Red de Área Local (LAN), la aplicación genera alertas al ser rebasado el nivel predeterminado del consumo del ancho de banda, de paquetes de entrada, salida y con errores. Asimismo genera un reporte del comportamiento de red en un periodo de tiempo.

Se utilizó la aplicación DNS y los protocolos TCP, UDP e IP para la construcción de un programa de análisis de tráfico de red, que permite obtener información de los paquetes de datos y de la información contenida dentro del paquete. Este análisis es primordial para generar una herramienta de administración de red que sea fácil de manejar y que permita a los usuarios interactuar con la aplicación.

Se realiza una conexión a la base de datos para almacenar el comportamiento de red y adicionalmente para almacenar la información que el usuario ingresa para la edición de nodos de red y configuración de alarmas. El contar con una base de datos permite generan reportes del comportamiento de red mediante una consulta a la base de datos, en ella se puede almacenar información por día, mese o año, permitiendo al administrador de red tener un historial del comportamiento de red y así poder predecir eventos que puedan degradar la red, este reporte puede ser impreso o almacenado en formato PDF en un directorio predeterminado.

Se detecto el consumo de ancho de banda utilizando el tamaño del paquete de datos contenido dentro del encabezado de IP, esto genera alertas audibles y visibles por medio de mensajes cuando se rebasa el nivel predeterminado del consumo de ancho de banda.

Finalmente el estar en la maestría en tecnología avanzada me permitió obtener conocimientos muy específicos en cuanto a la programación, las redes de computadoras, bases de datos y la formación como un recurso humano de alto desempeño creativo listo para enfrentar los problemas utilizando el razonamiento para obtener una solución.

5.2 Trabajo a futuro.

El trabajo que se puede realizar para mejorar las prestaciones de la aplicación se describen en los siguientes puntos:

- Mejorar el rendimiento de las alarmas para que sean lo mas oportunas posibles.
- Incrementar el número de eventos de alarmas.
- Detección de accesos no autorizados para el control de envío y recepción de información en los correos electrónicos, servidores y páginas de Internet.
- Generar estadísticas de las situaciones anómalas que se presentan en la red, con la finalidad de predecir comportamientos en la red que puedan causar su degradación.
- Consultar la aplicación vía Internet para que la supervisión, consultas y en un momento dado la corrección de los problemas en la red se resuelvan vía remota.
- Aumentar la cantidad de protocolos como ARP, IGMP, etc.
- Utilizar el protocolo SNMP para el envío de alarmas y reportes por correo electrónico, el protocolo 802.1 para monitorear diferentes redes que comparten el mimo medio físico.
- Efectuar el monitoreo de VLAN.

- Conexión de escritorio remoto para solucionar problemas a distancia en los equipos instalados en la red.
- Envío de alerta a teléfonos móviles para que el administrador este informado del comportamiento de la red.

Bibliografía

[1] Andrew S. Tanenbaum; "Redes de Computadora"; Tercera edición, Ed. Prentice Hall; 1997.

[2] Fiach Reid, Network Programming in .NET with C# and Visual Basic.NET, Software development, Ed. El Sevier digital press, 2004.

[3] Abraham Silverschatz, Henry F. Korth, S.Sudarshan, Fundamentos de bases de datos, Quinta edición, Ed. Mc Graw Hill, 2006.

[4] Michael Halvorson, Microsoft Visual Basic .NET aprenda ya, Ed. Mc Graw Hill, 2002.

[5] Francisco Charte Ojeda, Programación Visual Basic 2008, Ed. Anaya, 2009.

[6] BLACK, U. Voice over IP. New Jersey: Prentice Hall PTR, (1999).

[7] DOUSKALIS, B. IP telephony: the integration of robust VoIP services. New Jersey: Prentice Hall PTR, (2000).

[8] Service Interworking for PSTN and IP Networks. IEEE Communication Magazine, Mayo 1999, pags. 104-111.

[9] Herrera. Tecnologías y Redes de Transmisión de Datos, Ed. Limusa, 2003.

[10] El Modelo OSI y los Protocolos de Red, Capitulo 2

[11] Rodrigo Ancavil del Pino, Captura de tramas Ethernet pyEtherIP, verción 1.0

[12] Leandro Martínez Beiro, Norma Soubal Peralta, Estudio Estadístico del comportamiento de una red, Centro de investigación y desarrollo para la informática (CIDET - ICIMAF), Cuba.

[13] Lic. Adrián Estrada Corona, Protocolo TCP/IP de Internet, vol.5 , número 8, Revista Digital Universitaria (UNAM), 10 de septiembre de 2004.

[14] Ing. Rosales Briceño Caryuly, Protocolo SNMP(Protocolo Sencillo de Administración de Redes), Telematique, ene – jun, vol.3, número 001, Universidad Rafael Belloso Chacin, Zulia, Venezuela pp.94-106, 2004

[15] Diego A. López García, Manuel Sánchez Raya, Fundamentos de comunicaciones y redes de datos, Escuela Politécnica Superior Universidad de Huelva, Departamento de Ing. Electrónica, Sistemas informáticos y Automática, Versión 2, 1 de Octubre de 2006.

[16] Santos Ferreras, Javier , Directores: Sans Bobi, Miguel Angel, Castro Ponce, Mario, Sistemas Distribuido de detecciones en redes de ordenadores basados en agentes inteligentes, Universidad Pontifica Comillas – ICAI.

[17] Armando José Urdaneta Montiel, Análisis de trafico en una red LAN aplicando la tecnología de redes neuronales, Teñematique, vol.5, número 001, Universidad Rafael Belloso Chacin, Zulia, Venezuela, 2006.

[18] Magallanes Frabrici, Canepa Daniel, Desarrollo de un tablero de control directivo para análisis de evaluación de productividad en una red de atención provincial, NEC Argentina S.A, Departamento de producción de Software, 2005

[19] Comparativa de seguridad en navegadores de Internet en entornos Windows vista y Windows 7, Informática 64, versión 1.1, 19 de abril de 2010

[20] CUERVO, F., GREENE, N., HUITEMA, C., RAYHAN, A., ROSEN, B. y SEGERS, J. (2000). Megaco Protocol versión 0.8. RFC 2885, Agosto 2000.

[21] DARPA Information Processing Techniques Office, Information Sciences Institute University of Southern California 4676 Admiralty Way Marina del Rey, California, Internet Protocol, RFC 791, Septiembre 1981.

[22] DARPA Information Processing Techniques Office, Information Sciences Institute University of Southern California 4676 Admiralty Way Marina del Rey, California, Transmission Control Protocol, RFC 793, Septiembre 1981.
[23] Domingo Sánchez Ruiz, Traducción al castellano en diciembre de 1999, Protocolo de datagramas de usuario, RFC 768, 28 de Agosto de 1981.
[24] P. Mockapetris, Network Working Group, DOMAIN NAMES - IMPLEMENTATION AND SPECIFICATION, RFC 1035, Noviembre de 1987.
[25] Microsoft, Introducción a las características y herramientas (SQL Server 2008),Microsoft Corporation [en línea] 2010,[consultado en abril de 2010] disponible en <http://msdn.microsoft.com/eses/library/bb500397%28v=SQL.105%29.aspx>
[26] Microsoft, Top características de Microsoft Beld 3, Microsoft Corporation [en linea] 2010, [consultado en abril 2010], disponible en <http://www.microsoft .com/spain/expression/ products/Blend_Features.aspx>
[27] Microsoft, Información General sobre Silverlight, Microsoft Corporation [en linea] 2010, [consultado en abril 2010], disponible en <http://msdn.microsoft .com/es-es/library/bb404700%28v=VS.95%29.aspx>
[28] GREENE, N., RAMALHO, M. y ROSEN, B. (2000). Media Gateways Control Protocol Architecture and Requeriments. RFC 2805, Abril 2000.
[29] HERSENT,O., GURLE, D. y PETIT, J.P. (2000). IP telephony: packet – based multimedia communication systems. Great Britain: Addison – Wesley.
[30] Mundo Cisco, ¿Qué es un sniffer?, Cisco Systems,2009
[31] Julio-Linux, Analizando red con Ettercap, 3 de mayo 2009
[32] Wireshark, wikiwireshark [en línea] 2010 [consultado el Lunes 8 de febrero de 2010], disponible < http://wiki.wireshark.org/CaptureSetup >
[33] Diego Gonzáles Gómez, Sistema de Detección de Intrusiones, versión 1.01,[en línea] Julio 2003 [consultado 24 marzo de 2010], disponible <http://www.dgonzalez.net/pub/ids/html/cap06.htm>
[34] Roesch, Marty et al. *Snort.org*. [en línea]. Actualizado semanalmente [consultado en Marzo de 2010]. Disponible en http://www.snort.org
[35] GFI Software Ltd. *GFI Security Event Log Monitor*. [en línea]. Fecha no disponible [consultado en Marzo de 2010]. Disponible en http://www.gfi.com/
[36] Tripwire, Inc. *Tripwire*. [en línea]. Fecha no disponible [consultado en Marzo de 2010]. Disponible en http://www.tripwire.com/
[37] Vern, Paxon. *Bro: A System for Detecting Network Intruders in Real-Time*. Lawrence Berkeley National Laboratory, Berkeley, CA and AT&T Center for Internet Research at ICSI, Berkeley, CA. [en línea]. 14 de diciembre de 1999 [consultado en marzo de 2010]. Disponible desde Internet en <http://www.icir.org/vern/bro-info.html>
[38] Leach, John and Gianni Tedesco. *Firestorm*. [en línea]. 2002 [consultado en Marzo, 2010]. Disponible en <http://www.scaramanga.co.uk/firestorm/ index.html>
[39] Symantec Corporation, La última generación de tecnología antivirus de Symantec, Seguridad de puntos finales, Ed. Symantec, 2007.
[40] HAMDI, M., VERSCHEURE, O., HUBAUX, J-P., DALGIC, I. y WANG, P. (Mayo, 1999).Voice
[41] Juan R. Hernandez Gambay, Maria H. Almager Cantú, Rubén A. Gonzalez García, Cálculo de ancho de banda necesario para una empresa, Universidad Juárez autónoma de tabasco, volumen: 4, Número: 2, Diciembre de 2005

[42] Kaythik Lakshminarayanan, Venkata N. Padmananbhan, Jitendra Padhye, Estimación del ancho de banda en redes de banda ancha, University of California Berkeley, Microsoft Research.
[43] Verónica Medina, Francisco Pérez, Sergio Martín, Jaime Benjumea y Daniel Carretero, Diseño de una librería en lenguaje "C" para manejar tarjetas Ethernet en las prácticas de redes de computadores, Departamento: Tecnología Electrónica.
[44] Redes y Tecnología S.A, Optimizador ET (Administración de su conexión a Internet), Redes y Tecnología, S.A. (Guatemala), disponible en: <http://www.redytec.net>
[45] Cabezas Ayala Rocío y García Quispe Ricardo. 21 de junio de 2007. Pronostico del consumo de ancho de banda utilizando redes neuronales en una empresa de tecnología de la información. Tesis.
[46] Tapia Jardinez Raul, Sanchez Ruiz y David Salvador. 25 de noviembre de 2009. Propuesta de un sistema de monitoreo para La red de ESIME ZACATENCO utilizando el protocolo SNMP y software libre. Tesis de nivel superior. Escuela Superior de Ingeniería Mecánica y Eléctrica unidad Zacatenco, disponible en: <http://hdl.handle.net/123456789/5456>
[47] Bravo Albarrán Gladys y Gómez Garduño Noa. 2010. Arquitectura de monitoreo en tiempo real de una red. Tesis de nivel superior. Escuela Superior de Ingeniería Mecánica y Eléctrica unidad Culhuacan, disponible en: http://hdl.handle.net/ 123456789/5456
[48] Ing. Calos Alberto Vicente Altamirano. [en línea] Junio 2005. Monitoreo de recursos de red. Dirección General de Servicios de Cómputo Académico. Universidad Nacional Autónoma de México, [consultado en Febrero, 2011]. Disponible en http://www.seguridad.unam.mx/eventos/admin-unam/Monitoreo.pdf
[49] Lic. Carlos A. Rincón C, Modelo matemático para la estimación del rendimiento de una red Ethernet, Universidad Rafael Belloso Chacín Venezuela, volumen: 3, Número de edición: 2, año 2004
[50] BOGGS David, KENT Christopher, MOGUL Jeffrey. 1988. Measured Capacity of an Ethernet: Myths and Reality. Computer Communication Reviews. Volumen: 18. Número de edición: 4. 1988
[51] Copyright © 1997, 2010, Oracle and/or its affiliates. All rights reserved, MySQL 5.0 Reference Manual.

Glosario

- **TCP**(Protocolo de Control de Transmisión).- Es el encargado de añadir las funciones necesarias para que la comunicación entre sistemas sea libre de errores, sin perdidas y con seguridad.

- **IP**(Protocolo de Internet).- Este protocolo únicamente proporciona seguridad de sus cabeceras y no de sus datos transmitidos.

- **UDP**(Protocolo de Datagramas de Usuario).- Es un protocolo de transporte sin conexión que proporciona servicios en colaboración con el TCP.

- **ARP** (Protocolo de Resolución de Direcciones).- Es el responsable de obtener las direcciones físicas de una dirección IP.

- **SMTP** (Protocolo Simple de Transferencia de Correo).- Proporciona servicios de correo electrónico en las redes Internet e IP.

- **ICMP** (Protocolo de Mensajes de Control de Internet).- es un subprotocolo del IP su principal función es enviar mensajes de error.

- **IGMP** (Protocolo de Manejos de Grupos de Internet).- permite el intercambio de información entre enrutadores de IP.

- **IPv4** es la versión 4 del protocolo de Internet.

- **802.1** Es un protocolo que permite a múltiples redes compartir el mismo medio físico.

- **IrDA** (Asociación de Datos Infrarrojos).- Es estándar físico en la forma de transmisión y recepción de datos infrarrojos.

- **IPX** (Protocolo de Intercambio de Paquetes entre redes).- Este protocolo se maneja en la capa de transporte no orientado a la conexión que gestiona el direccionamiento de los datos en una red Novell Netware.

- **PING**(Rastreador de paquetes de Internet).- Es una utilidad de diagnostico para la comprobación de conectividad entre computadoras por medio del envío de paquetes ICMP de solicitud y respuesta

- **ECO**.- Es el número de solicitudes a enviar de un equipo origen al destino.

- **Tamaño del buffer**.- Cantidad de Bytes enviados.

- **Tiempo de vida**.- Permite conocer la cantidad de ruteadores por los que pasa el paquete mientras viaja de una maquina a otra.

Apéndice “A”

El modelo OSI

A la par de la creación del protocolo TCP/IP también se fue desarrollando un modelo mas completo y complejo de comunicación, desarrollado por la Organización Internacional para la Estandarización (ISO, por sus siglas en ingles) conocido como el Modelo de Referencia de Interconexión de Sistemas Abiertos (OSI, *Open System Interconnection Reference Model*) el cual esta mas apegado a la forma de comunicación entre locutor y receptor, precisamente por estas razones es que tardó mas tiempo en terminarse, pero en 1984 [10] este modelo pasó a ser el estándar internacional para las comunicaciones en red, el modelo OSI esta compuesto por siete capas que son aplicación, presentación, sesión, transporte, red, enlace de datos y física que se encuentran numerados de abajo hacia arriba como lo muestra la figura.A.1.

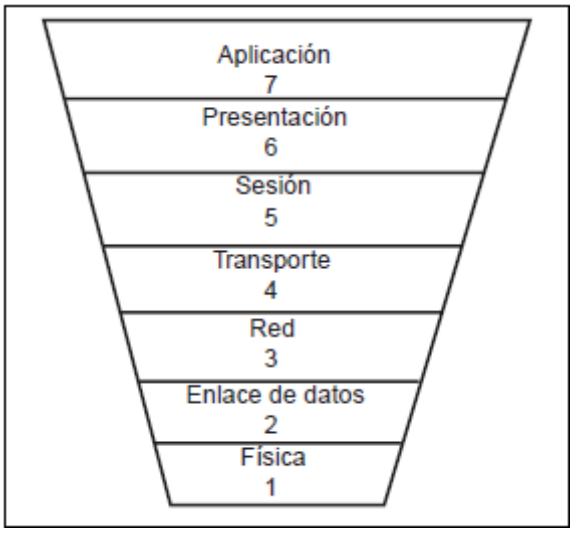


Figura.A.1 Pila de protocolos del modelo OSI. [10]

Cada una de las capas describe el proceso de trasmisión de datos entre computadoras dentro de la red, esto es mediante la colocación de encabezados que coloca cada una de las capas al pasar la información, así cuando los datos son transmitidos por el medio físico y llegan a la computadora receptora, cada una de las capas toma la información

que le corresponde para finalmente presentarla en pantalla como lo muestra la figura A.2.

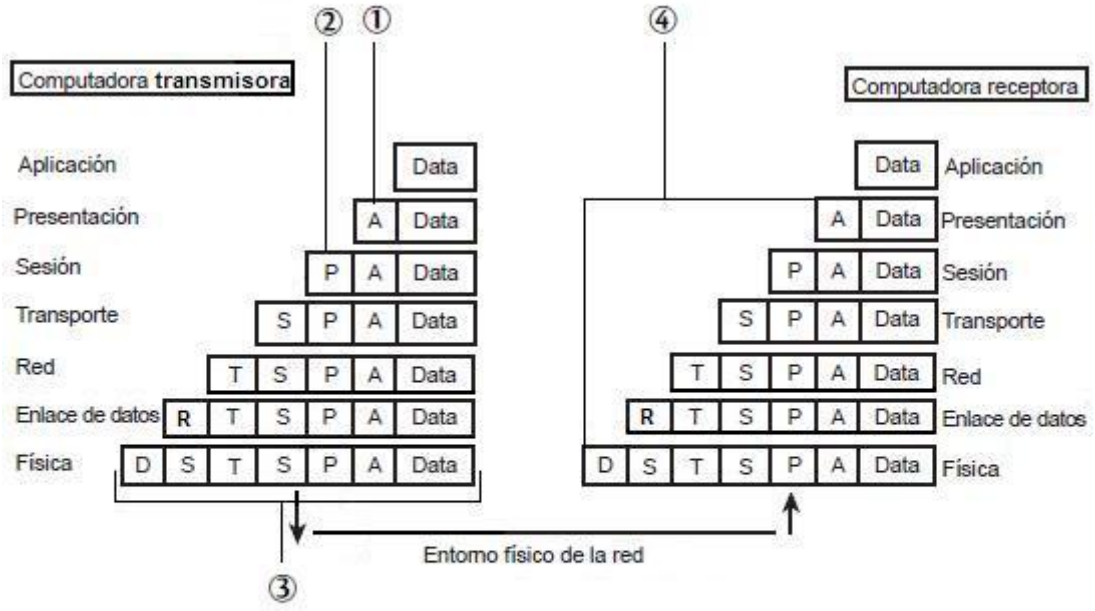


Figura A.2 Forma de comunicación de dos computadoras. [10]

Función de las capas del modelo OSI.

La capa de aplicación.

Es la capa con la cual la mayoría de los usuarios tiene contacto ya que proporciona la interfaz y servicios que soportan las aplicaciones de usuario. También se encarga de ofrecer acceso general a la red, ofrece los servicios relacionados con aplicaciones de usuario en la red, a demás de gestionar aplicaciones Web y servicios de correo electrónico (Eje. Protocolo Simple de Transferencia de Correo, comúnmente conocido como SMTP – *Simple Mail Tranfer Protocol* – incluido en TPC/IP), así como aplicaciones especiales de bases de datos cliente / servidor.

La capa de presentación.

Se puede considerarse el traductor del modelo OSI. Por que toma los paquetes se ocupa de la sintaxis y la semántica de la información de la capa de aplicación y los convierte a un formato genérico que pueden leer todas las computadoras.

También se encarga de cifrar los datos así como de comprimirlos para reducir su tamaño. Contiene los datos prácticamente con el formato con el que viajarán por las restantes capas de la pila OSI.

La capa de sesión.

Es la encargada de establecer el enlace de comunicaciones o sesión entre las computadoras emisora y receptora. Gestiona la sesión que se establece entre ambos nodos.

Una vez establecida la sesión pasa a encargarse de ubicar puntos de control en la secuencia de datos de modo que después de cada interrupción solo se deban repetir los datos que se transfirieron después del último punto de control. Los protocolos que operan en la capa de sesión pueden proporcionar comunicación orientada a la conexión y la comunicación sin conexión[82].

Los protocolos orientados a la conexión que operan en la capa de sesión proporcionan un entorno donde las computadoras conectadas se ponen de acuerdo sobre los parámetros relativos a la creación de los puntos de control en los datos. El funcionamiento de los protocolos sin conexión se parece más bien a un sistema de correo regular.

La Capa de Transporte.

Es la encargada de controlar el flujo de datos entre los nodos que establecen una comunicación, evalúa el tamaño de los datos con el fin de que estos tengan el tamaño

[82] La comunicación orientada a la conexión devuelve una respuesta cuando la información de transmisor a receptor llego sin errores, mientas que la comunicación sin conexión no devuelve ningún tipo de respuesta cuando se envía la información.

requerido por las capas inferiores del conjunto de protocolos, si el tamaño de los datos no es el adecuado los divide en unidades mas pequeñas para pasarlos a la capa de red para asegurar que todos los datos lleguen al otro extremo.

La Capa de Red.

Encamina los paquetes además de ocuparse de entregarlos, determina la ruta que deben seguir los datos, es en esta capa es donde las direcciones lógicas (direcciones IP de una computadora de red) pasan a convertirse en direcciones físicas (direcciones de hardware Tarjeta de Interfaz de Red, NIC, *Network Interface Card*).

La Capa de Enlace de Datos.

Una vez que los datos llegan a la capa de enlace, éstos pasan a ubicarse en tramas (unidades de datos, figura A.3), se encarga de desplazar los datos por el enlace físico de comunicación hasta el nodo receptor, e identifica cada computadora incluida en la red de acuerdo con su dirección de hardware.

Se asegura de que las tramas enviadas por el enlace físico se reciben sin error alguno ya que los protocolos que operan en esta capa adjuntarán un Chequeo de Redundancia Cíclica (CRC) al final de cada trama, básicamente es un valor que se calcula tanto en la computadora emisora como en la receptora. Si coinciden, significa que la trama se recibió correctamente e íntegramente, y no sufrió error alguno durante su transferencia, en el caso que los valores calculados no coincidan, quiere decir que no se recibió la trama o que si llego pero con errores, por lo que se enviará un mensaje a la computadora transmisora para que vuelva a enviar la trama de datos.

Figura A.3 Trama de datos Ethernet. [10]

En la tabla A.1 se describe cada uno de los componentes de la trama de datos, la difusión tiene una consideración adicional en la capa de enlace de datos: como controlar el acceso al canal compartido [1]. Por lo cual existe una subcapa encargada de controlar estos procesos, conocidas como las subcapas de enlace de datos, por lo que algunos puntos de la tabla.1 se abordaran en ese tema.

Segmento	Función
Preámbulo	Bits de alternación (1 y 0) que indican que se ha enviado una trama
Destino	Dirección de destino.
Fuente	Dirección de origen.
Longitud	Especifica el número de bytes de datos incluidos en la trama
DSAP	Punto de acceso al servicio de destino (*Destination Service Acces Point*), indica a la tarjeta de red de la computadora receptora dónde tiene que ubicar la trama dentro de la memoria intermedia.
SSAP	Proporciona el punto de acceso al servicio (*Service Access Point*), para la trama de datos.
CTRL	Control lógico de enlace (Se explicara mas a detalle en la subcapa de enlace de datos).
Datos	Mantiene la información que se ha enviado.
FCS	Secuencia de comportamiento de trama (*Frame Check Sequences*) contiene el valor de chequeo de redundancia cíclica (CRC) para la trama

Tabla A.1 Componentes de trama Ethernet. [10]

La Capa Física.

Aquí llegan las tramas de la capa de enlace de datos y se convierten en una secuencia única de bits que pueden transferirse por el entorno físico de la red (medio de transmisión[83]). Determina los aspectos físicos sobre la forma transmisión de la información de la tarjeta de red (NIC, *Network Interface Card,*) de la computadora al entorno físico.

[83] Los medios de transmisión puede ser por cable partensado, coaxial, etc.

Las subcapas del enlace de datos.

Algunas especificaciones desarrolladas por el IEEE[84] para la capa de enlace de datos del modelo OSI la dividen en dos subcapas, el Control Lógico del Enlace (*Logical Link Control* o LLC) y el Control de Acceso al Medio (*Media Access Control* o MAC).

La subcapa de Control Lógico del Enlace establece y mantiene el enlace entre las computadoras emisora y receptora cuando los datos se desplazan por el entorno físico de la red, también proporciona puntos de acceso al servidor (SAP) que son puntos de referencia a computadoras que envían información con las capas superiores del conjunto de protocolos OSI dentro de un determinado nodo receptor. La IEEE define a la capa LLC en el estándar 802.2[85].

La subcapa de Control de Acceso al Medio determina la forma en que las computadoras se comunican dentro de la red, como y dónde una computadora puede acceder al entorno físico y enviar datos. La especificación 802 divide a su vez la subcapa MAC en una serie de categorías relacionadas con la arquitectura de red como Ethernet y Token Ring (Fig.4).

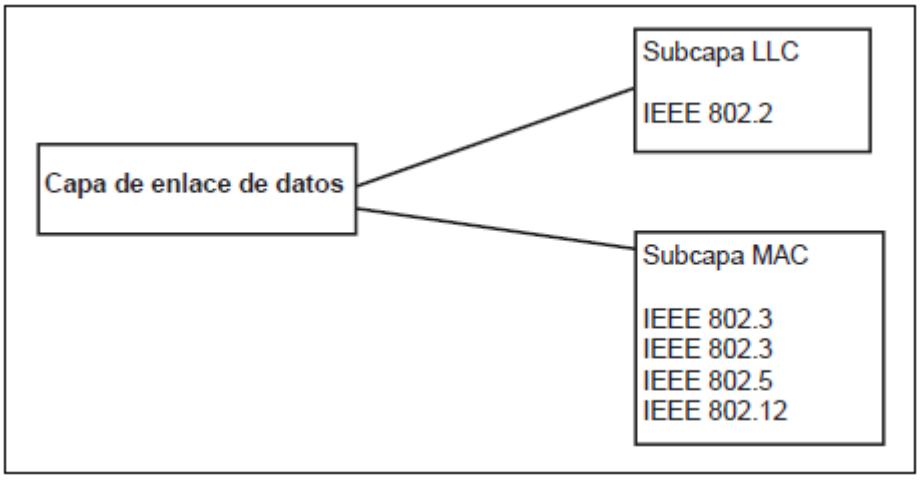


Fig.4 División de la subcapa de enlace de datos. [10]

[84] Instituto de Ingenieros electricistas y electrónicos (IEEE por sus siglas en ingles, *Institute of Electrical and Electronics Engineers*).
[85] Control de enlace lógico.

Apéndice “B”

Adaptador de red

El adaptador de red o tarjeta de interfaz de red (NIC por sus siglas en ingles, *Network Interface Card*) funciona como interface entre el punto físico de conexión de la red (nodo) y la computadora; la tarjeta de red proporciona una conexión con la red por un medio de transmisión cableado o inalámbrico utilizando conectores RJ45[86] para una conexión punto a punto o por un canal de difusión[87] para una conexión inalámbrica a un punto de acceso de dispositivos de comunicación inalámbrica (WAP o AP[88] por sus siglas en inglés , *Wireles Acces Point*).

La tarjeta de interfaz de red esta compuesto por dos componentes:

1) Una interfaz de BUS (sistema digital[89] de transferencia de datos) de la computadora (fig.B.1).
2) Una interfaz al enlace por medio de un cable o antena (fig.B.1).

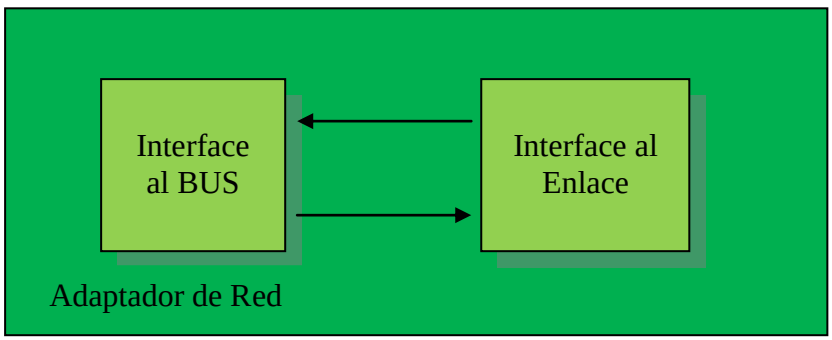


Figura.B.1. – Componentes del adaptador de red.

La figura.B.2 muestra los bloques de los componentes de la tarjeta de red, también se observa la conexión entre el bus de entrada y salida (E/S) de la computadora

[86] Registro toma 45 (RJ45 por sus siglas del ingles, *registered jack 45*) Interface que se usa para conectar redes de cableado estructurado.
[87] Medio de comunicación por donde viaja la forma de onda de la señal (portadora de la información) del transmisor al receptor.
[88] Dispositivo que se encuentra conectado por un medio de transmisión a la red.
[89] Es un sistema de transmisión o procesamiento de información en el cual la información se encuentre representada por medio de cantidades físicas (señales).

y la tarjeta de interfaz de red (NIC) [44] y los diagramas a bloques de los componentes de la tarjeta de red, en donde la interfaz al enlace contienen los registros de configuración de los *sockets* o conector, también llamados registros de descripción [2].

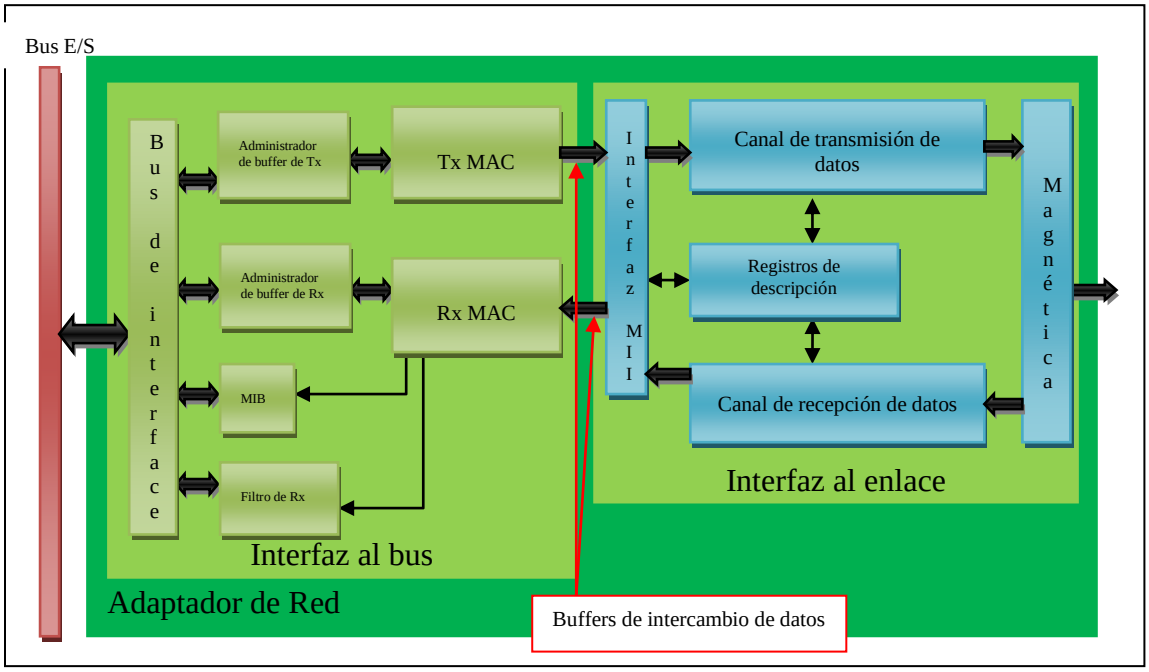


Figura.B.2. Arquitectura de los componentes de la tarjeta de red[90].

En la figura.B.3 muestra el proceso de comunicación entre el adaptador de red, el bus de entrada / salida, la memoria de acceso aleatorio (RAM por sus siglas en ingles, *Random Acces Memory*), la memoria cache, la unidad central de proceso (CPU por sus siglas en ingles, *Central Prossesing Unit*) y el enlace a la red.

[90] Para mayor información de cada bloque contenido en los componentes de la NIC puede consultar las hojas de datos del DP38820 10/100/1000 Mb/s PCI Ethernet Network Interface Control, DP83847 DsPHYTER II – Single 10/100 Ethernet Transcervier y DP83865 Gig PHYTER 10/100/1000 Ethernet Physical Layer de National Semiconductor

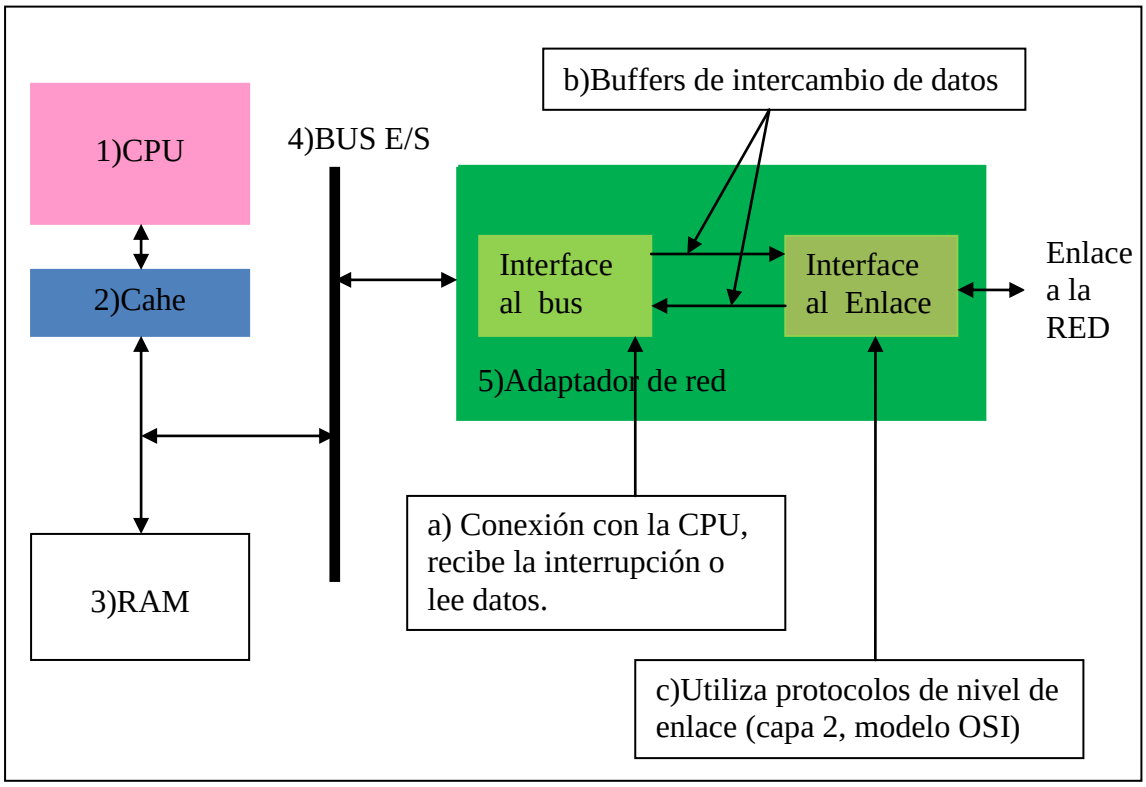


Figura.B.3 - Arquitectura de comunicación entre la tarjeta de red y la CPU.

El proceso de comunicación tiene los siguientes pasos:

1) Interrupción[91] de la CPU para el envió o recepción de datos.
2) Se envía una copia de los datos a enviar a la memoria caché para aumentar la velocidad de acceso a los datos.
3) Los datos se almacenan temporalmente en la memoria RAM.
4) Los datos son enviados por el bus de entrada / salida.
5) Los datos llega al adaptador de red donde:
    a. La interfaz al BUS toma la interrupción y recibe o envía los datos de la CPU a la red o viceversa.
    b. Los datos pasan al buffer de intercambio de datos que se encarga de enviar o recibir datos.
    c. Los datos pasan a la interfaz de enlace que envía los datos a la red o que recibe los datos que provienen de la red.

[91] Es una petición de envió o recepción de datos que hace la CPU, e indica que se debe interrumpir algún proceso que se este haciendo y ejecutar la petición actual.

La tarjeta de interfaz de red requiere para su comunicación con el sistema operativo un manejador lógico que es un software de configuración (driver) que permite realizar las siguientes funciones:

1) Inicialización de la tarjeta: Establece comunicación con el adaptador de red, verifica la compatibilidad y su estado.
2) Servicio de interrupción (IRQ por sus siglas en ingles, *Interrupt Request*): Envía un mensaje al sistema y al adaptador de red que se producirá un evento de comunicación entre ellos
3) Transmisión y recepción de tramas de datos: realiza accesos directos a memoria (DMA por sus siglas en ingles, *Direct Memory Access*) para el envió y recepción de datos.

Apéndice “C”

Bases de datos relacionales

C.1 Consultas con algebra relacional.

Una consulta es una solicitud de información a la base de datos, se realiza por medio de expresiones de un lenguaje especializado de consultas como es el algebra relacional.

El algebra relacional consiste en un conjunto de operaciones finitas que toman una o dos relaciones como entrada y generan otra relación como resultado, las operaciones utilizadas para realizar las consultas en la base de datos del tablero de control utilizando el algebra relacional son:

1) Selección: Selecciona tuplas[92] que satisfacen un predicado dado. Se usa la letra griega sigma minúscula (σ) para denotar la selección. El predicado aparece como subíndice de σ, la relación se coloca entre paréntesis a continuación de sigma, su expresión es la siguiente:

$$\sigma_P(r)$$

Donde:

   a. σ: es la selección
   b. P: es el predicado
   c. r: es la relación

2) Proyección: Devuelve relaciones entre los argumentos, se denota por la letra griega mayúscula pi (∏). Se crea una lista de atributos que se desea que aparezcan en el resultado como subíndices de ∏, su único argumento es una relación y se escribe a continuación entre paréntesis, su expresión es la siguiente:

$$\prod_{(A_1, A_2, \ldots, A_n)}(r)$$

Donde:

[92] En términos de de bases de datos una tupla es una fila.

a. ∏: es la proyección
b. A: son los atributos
c. r: es la relación

3) Producto cartesiano: Se denota por un aspa (×), permite combinar información de cualquier relación.
4) Reunión natural: Forma un producto cartesiano de dos argumentos, realiza una selección forzando la igualdad de los atributos que aparecen en ambos esquemas de relación y elimina los atributos duplicados, se denota por el símbolo "⋈".

Para realizar modificaciones en la base de datos del tablero de control se utilizan las operaciones borrar, insertar y actualizar que utilizan el operador asignación (←) para realizar las modificaciones, sus expresiones son las siguientes:

a) Borrado de una tupla.

$$r \leftarrow r - E$$

Donde:

r: es una relación

E: es una consulta algebraica

b) Inserción de tupla.

$$r \leftarrow r \cup E$$

Donde:

r: es una relación

E: es una consulta algebraica

c) Actualización de tupla.

$$r \leftarrow \prod_{F_1, F_2, \ldots, F_n} (r)$$

Donde:

r: es una relación

∏: es la proyección

F: es un atributo de r

C.2 Consultas de SQL del algebra relacional.

El lenguaje estructurado de consultas o SQL (por sus siglas en ingles, *Structured Query Lenguage*) [3], es un lenguaje que contiene los siguientes componentes:

- Lenguaje de definición de datos (LDD): Proporciona comandos para la definición de esquemas de relación, borrado de relaciones y modificación de esquemas de relación [3].
- Lenguaje interactivo de manipulación de datos (LMD): Proporciona un lenguaje de consultas basado en el algebra relacional como en el calculo relacional de tuplas, también contiene comandos para insertar, borrar y modificar tuplas [3].
- Definición de vistas: El LDD de SQL incluye comandos para la definición de vistas [3].

En SQL las relaciones se definen mediante el comando **create table** su sintaxis es la siguiente:

create table $r(A_1D_1,A_2D_2,\ldots,A_nD_n,$
$\langle restricción - integridad_1\rangle,$
$\ldots,$
$\langle restricción - integridad_k\rangle,)$

Donde:
r: es el nombre de la relación
A_i: es el nombre del atributo del esquema de la relación r
D_i: es el tipo de dominio de los valores del dominio del atributo A_i

Para la especificación de la clave primaria ninguna tupla puede tener un valor nulo (no nulo) y ningún par de tuplas de la relación puede ser igual que todos los atributos de la clave primaria (únicos) [3].

La estructura básica de una expresión SQL consta de tres cláusulas importantes:

- La cláusula **select** corresponde con la operación proyección del álgebra relacional. Se usa para obtener una relación de los atributos deseados en el resultado de una consulta.
- La clausula **from** corresponde con la operación producto cartesiano del álgebra relacional. Genera una lista de las relaciones que deben ser analizadas en la relación de la expresión.
- La clausula **where** corresponde con el predicado selección del álgebra relacional. Es un predicado que engloba a los atributos de las relaciones de la clausula **from**.

Las consultas habituales de SQL tienen la siguiente sintaxis:

select $A_1, A_2, \ldots, A_n$
from $r_1, r_2, \ldots, r_n$
where P

Donde:
A_i: representa un atributo
r_i: representa una relación
P: es un predicado

La consulta equivalente a la expresión del álgebra relacional es la siguiente:

$$\prod_{A_1, A_2, \ldots, A_n} \left(\sigma_p \left(r_1 \times r_2 \times \ldots \times r_m \right) \right)$$

Para realizar modificaciones en la base de datos del tablero de control se utilizan los comandos **deleted** (borrar), **insert** (insertar) y **update** (actualizar). Las solicitudes de borrado se expresan casi igual que la consulta y sólo se pueden borrar tuplas completas; no se pueden borrar sólo valores de atributos concretos. La sintaxis de SQL para el borrado de una tupla es:

deleted from r
where P

Donde:

P: representa un predicado
r: representa una relación

Para insertar datos en una relación se ocupa el comando **insert**, se especifica la tupla que se desea insertar o se formula una consulta cuyo resultado sea un conjunto de tuplas a insertar. Los valores de los atributos de las tuplas que se inserten deben pertenecer al dominio[93] de los atributos. Los valores se especifican en el mismo orden en que aparecen los atributos correspondientes en el esquema de la relación.

Para actualizar un valor dentro de una tupla sin cambiar los demás valores se utiliza la instrucción **update**. Al igual que ocurre con **insert** y **deleted**, se pueden elegir las tuplas que se van a actualizar.

[93] Son conjuntos de valores permitidos como: char, varchar, int, smallint, numeric, real, float, etc.

Apéndice “D”

Código husmeador de paquetes o sniffer

Éste apéndice contiene el código fuente de cada uno de los módulos que componen al husmeador de paquetes, junto con su ventana de windows form, el lenguaje de programación que se utiliza es visual basic.NET.

```
'Código fuente husmeador de paquetes de red
'Ing. Jonatan Juárez Hinojosa
'Programa que detecta y analiza el trafico de red,
'analiza la información de los encabezados de IP,TCP,UDP y DNS
'-------------------------------------------------------
Imports System
Imports System.Collections.Generic
Imports System.ComponentModel
Imports System.Data
Imports System.Drawing
Imports System.Text
Imports System.Windows.Forms
Imports System.Net.Sockets
Imports System.Net


Public Enum Protocolo 'Protocolos a manejar
    TCP = 6
    UDP = 17
    Unknown = -1
End Enum

Public Class PlizSnifferForm
    Private mainSocket As Socket  'El socket que captura todos los paquetes
    Dim buf_recend As Integer = 4069
    Private ByteDato() As Byte = New Byte(buf_recend) {}
    'Private ByteDato(0 To 4096 - 1) As Byte
```

```
   Private bContinuaCap As Boolean = False 'Una vandera de chequeo, si los paquetes
se capturan o no
   Private Delegate Sub AddTreeNode(ByVal node As TreeNode)


   Public Sub PlizSnifferForm()
      InitializeComponent()
   End Sub

   Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
      If ComboBox1.Text = "" Then
         MessageBox.Show("Seleccione una interface de captura de paquetes.",
"Usmeador", MessageBoxButtons.OK, MessageBoxIcon.Error)
         Return
      End If
      Try
         If Not bContinuaCap Then

            'Comienzo de captura de paquetes

            Button1.Text = "&Alto"
            bContinuaCap = True

            'Para la captura de paquetes se tiene que haber sobre un Socket
            'libre, con la familia de direcciones  que deben ser del tipo de red
            'y el protocolo IP
            mainSocket = New Socket(AddressFamily.InterNetwork, SocketType.Raw,
                                    ProtocolType.IP)

            'Amarra el socket a la dirección IP seleccionada
            mainSocket.Bind(New IPEndPoint(IPAddress.Parse(ComboBox1.Text), 0))

            'Conjunto de opciones del socket
            mainSocket.SetSocketOption(SocketOptionLevel.IP,
SocketOptionName.HeaderIncluded,
                                       True)
            'Solo  pilas  de paquetes se incluyen en el conjunto del cabezal con la opción
verdadera

            Dim byIn As Byte() = New Byte() {1, 0, 0, 0}
            Dim byOut As Byte() = New Byte() {1, 0, 0, 0} 'Captura de salida de
paquetes

            'Socket.IOControl es analogo a el método WSAIoctl de Winsock 2
            'equivalente  a la constante SIO_RCVALL de Winsock 2
            mainSocket.IOControl(IOControlCode.ReceiveAll, byIn, byOut)

            'Se activa la recepción asincrona de paquetes
```

```
            mainSocket.BeginReceive(ByteDato, 0, ByteDato.Length, SocketFlags.None,
_
                                    New AsyncCallback(AddressOf RecepciON), Nothing)
        Else
            Button1.Text = "&Iniciar"
            bContinuaCap = False
            'Se detiene la captura de paquetes y se
            'cierra el socket
            mainSocket.Close()
        End If
    Catch ex As Exception
        MessageBox.Show(ex.Message, "Esniffer(1)", MessageBoxButtons.OK, _
                        MessageBoxIcon.Error)
    End Try
End Sub

Private Sub RecepciON(ByVal ar As IAsyncResult)
    Try
        Dim nRecep As Integer = mainSocket.EndReceive(ar)

        'Analiza los bytes recibidos

        ParseData(ByteDato, nRecep)
        If bContinuaCap Then
            ReDim ByteDato(0 To 4069 - 1)
            'Se realiza otra llamada a BeginReceive de modo que se continua recibiendo
            'los paquetes
            mainSocket.BeginReceive(ByteDato, 0, ByteDato.Length, SocketFlags.None,
_
                                    New AsyncCallback(AddressOf RecepciON), Nothing)
        End If
    Catch ex As Exception
        MessageBox.Show(ex.Message, "Esniffer(2)", MessageBoxButtons.OK, _
                        MessageBoxIcon.Error)
    End Try
End Sub
Private Sub ParseData(ByVal ByteData As Byte(), ByVal nReceived As Integer)

    Dim NodoRaiz As New TreeNode

    'Ya que todos los paquetes de protolos son encapsulados en el datagrama IP
    'entonces comensamos por análizar el encabesado de IP y vemos que datos de
    'protocolos estan siendo llevados por ellos.

    Dim ipHeader As IPHeader = New IPHeader(ByteData, nReceived)
    Dim ipNode As TreeNode = MakeIPTreeNode(IPHeader)
    NodoRaiz.Nodes.Add(ipNode)

    'Ahora segun el protocolo que es llevado por el datagrama de IP,
    'análizamos el campo de datos del datagrama.
```

```
        Select Case ipHeader.ProtocoloType
            Case Protocolo.TCP
                Dim tcpHeader As New TCPHeader(ipHeader.Dato,
ipHeader.MessageLength)
                Dim tcpNode As TreeNode = MakeTCPTreeNode(TCPHeader)
                NodoRaiz.Nodes.Add(tcpNode)

                'Si el puerto es equivalente a 53 entonces el protocolo subrallado es DNS
                'Nota: DNS puede usar TCP o UDP por que la comprobación es hecha dos
veces
                If tcpHeader.DestinationProt = "53" OrElse tcpHeader.SourcePort = "53"
Then
                    Dim dnsNode As TreeNode = MakeDNSTreeNode(TCPHeader.Data, _
                                              CInt(TCPHeader.MessageLength))
                End If
            Case Protocolo.UDP
                Dim udpHeader As New UDPHeader(ipHeader.Dato,
CInt(ipHeader.MessageLength))
                Dim udpNode As TreeNode = MakeUDPTreeNode(UDPHeader)
                NodoRaiz.Nodes.Add(udpNode)

                'Si el puerto es igual a 53 entonces el protocolo subyacente es DNS
                'Nota: DNS puede usar TCP o UDP por que la comprobación se hace dos
veces
                If udpHeader.DestinatinationPort = "53" OrElse udpHeader.SourcePort =
"53" Then
                    'La Longitud del encabezado UDP es siempre de 8 bytes entonces restamos
el total
                    'de la longitud para encontrar la longitud de los datos
                    Dim dnsNode As TreeNode = MakeDNSTreeNode(UDPHeader.Data, _
                                              Convert.ToInt32(UDPHeader.Lengh) - 8)
                    NodoRaiz.Nodes.Add(dnsNode)
                End If
            Case Protocolo.Unknown
        End Select
        Dim addTreeNode As New AddTreeNode(AddressOf OnAddTreeNode)
        NodoRaiz.Text = ipHeader.SourceAddress.ToString() & "-" & _
                        ipHeader.DestinationAddress.ToString()
        'Caja fuerte de los hilos adicionados en los nodos
        TreeView1.Invoke(addTreeNode, New Object(), NodoRaiz)
    End Sub
    'La función de ayuda que devuelve la información contenida
    'en el cabezal de IP como un nodo de árbol
    Private Function MakeIPTreeNode(ByVal ipHeader As IPHeader) As TreeNode
        Dim ipNode As New TreeNode()
        ipNode.Text = "IP"
        ipNode.Nodes.Add("Ver: " & ipHeader.Version)
        ipNode.Nodes.Add("Longitud de encabezado: " & ipHeader.HeaderLength)
        ipNode.Nodes.Add("Differntiated Services: " & ipHeader.DifferentiatedServices)
        ipNode.Nodes.Add("Longitud total: " & ipHeader.TotalLength)
```

```
        ipNode.Nodes.Add("Identificación: " & ipHeader.Identification)
        ipNode.Nodes.Add("Banderas: " & ipHeader.Flags)
        ipNode.Nodes.Add("Compensación de Fragmentación: " &
ipHeader.FragmentationOffset)
        ipNode.Nodes.Add("Tiempo de vida: " & ipHeader.TTL)

        Select Case ipHeader.ProtocoloType
            Case Protocolo.TCP
                ipNode.Nodes.Add("Protocolo: " & "TCP")
            Case Protocolo.UDP
                ipNode.Nodes.Add("Protocolo: " & "UDP")
            Case Protocolo.Unknown
                ipNode.Nodes.Add("Protocolo: " & "Desconocido")
        End Select

        ipNode.Nodes.Add("Suma de comprobación: " & ipHeader.Checksum)
        ipNode.Nodes.Add("Fuente: " & ipHeader.SourceAddress.ToString())
        ipNode.Nodes.Add("Destino: " & ipHeader.DestinationAddress.ToString())

        Return ipNode
    End Function

    'La función de ayuda que devuelve la información contenida
    'en el encabezado de TCP como un nodo árbol
    Private Function MakeTCPTreeNode(ByVal tcpHeader As TCPHeader) As
TreeNode
        Dim tcpNode As New TreeNode()
        tcpNode.Text = "TCP"
        tcpNode.Nodes.Add("Puerto origen: " & tcpHeader.SourcePort)
        tcpNode.Nodes.Add("Puerto destino: " & tcpHeader.DestinationProt)
        tcpNode.Nodes.Add("Numero de secuencia: " & tcpHeader.SequenceNumber)

        If tcpHeader.AcknowledgementNumber <> "" Then
            tcpNode.Nodes.Add("Número de acuse (ACK): " &
tcpHeader.SequenceNumber)
            'Acknowledgement (ACK).- Acuse de recibo. Un tipo de mensaje que se envia
            'para indicar que un bloque de datos a llegado a su destino legible y sin
            'errores.
        End If

        tcpNode.Nodes.Add("Longitud de encabezado: " & tcpHeader.HeaderLength)
        tcpNode.Nodes.Add("Banderas: " & tcpHeader.Flags)
        tcpNode.Nodes.Add("Suma de comprobación: " & tcpHeader.Checksum)

        If tcpHeader.UrgentPoint <> "" Then
            tcpNode.Nodes.Add("Indicador de urgencia: " & tcpHeader.UrgentPoint)
        End If

        Return tcpNode
    End Function
```

```
    'Función de ayuda que devuelve la infromación contenida
    'en el cabezal UDP como un nodo árbol
    Private Function MakeUDPTreeNode(ByVal udpHeader As UDPHeader) As
TreeNode

        Dim udpNode As New TreeNode()
        udpNode.Text = "UDP"
        udpNode.Nodes.Add("Puerto de origen: " & udpHeader.SourcePort)
        udpNode.Nodes.Add("Puerto destino: " & udpHeader.DestinatinationPort)
        udpNode.Nodes.Add("Longitud: " & udpHeader.Lengh)
        udpNode.Nodes.Add("Suma de comprobación: " & udpHeader.Checksum)

        Return udpNode
    End Function

    'La función de ayuda que devuelve la información contenida
    'en el DNS del nodo árbol
    Private Function MakeDNSTreeNode(ByVal byteData() As Byte, ByVal nLength As
Integer) As TreeNode
        Dim dnsHeader As New DNSHeader(byteData, nLength)
        Dim dnsNode As New TreeNode()
        dnsNode.Nodes.Add("Identificador: " & dnsHeader.Identification)
        dnsNode.Nodes.Add("Banderas: " & dnsHeader.Flags)
        dnsNode.Nodes.Add("Consulta: " & dnsHeader.TotalQuestion)
        dnsNode.Nodes.Add("Respuesta RRS: " & dnsHeader.TotalAnswerRRs)
        dnsNode.Nodes.Add("Autoridad RRS: " & dnsHeader.TotalAuthorityRRs)
        dnsNode.Nodes.Add("Adicional RRS: " & dnsHeader.TotalAdditionalRRs)

        Return dnsNode
    End Function

    Private Sub OnAddTreeNode(ByVal node As TreeNode)
        TreeView1.Nodes.Add(node)
    End Sub

    Private Sub SnifferForm_Load(ByVal sender As Object, ByVal e As EventArgs)
Handles MyBase.Load
        Dim strIP As String = Nothing
        Dim HosyEntry As IPHostEntry = Dns.GetHostEntry((Dns.GetHostName()))
        If HosyEntry.AddressList.Length > 0 Then
            For Each ip As IPAddress In HosyEntry.AddressList
                strIP = ip.ToString()
                ComboBox1.Items.Add(strIP)
            Next
        End If
    End Sub

    Private Sub SnifferForm_FormClosing(ByVal sender As System.Object, ByVal e As
System.Windows.Forms.FormClosingEventArgs) Handles MyBase.FormClosing
```

```
        If bContinuaCap Then
            mainSocket.Close()
        End If
    End Sub
End Class
```

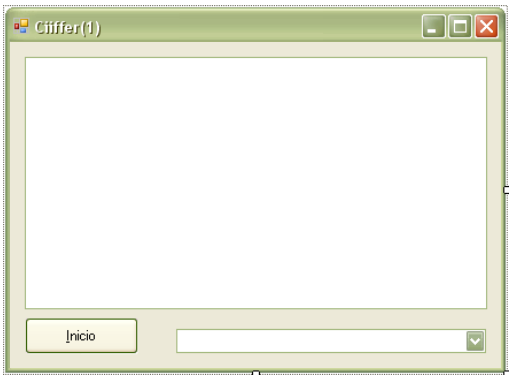


Figura D.1 Aplicación de windows form del husmeador de paquetes.

```
'Código IPHeader
'Ing. Jonatan Juárez Hinojosa
'Función que obtiene la información del enbezado IP
'-------------------------------------------------------
Imports System.Net
Imports System.Text
Imports System
Imports System.IO
Imports System.Windows.Forms

Public Class IPHeader

    'Archivos de cabesera IP
    Private byVersionAndHeaderLength As Byte    'bit alto de la versión y longitud del
encabesado
    Private byDifferentiatedServices As Byte    'bit alto para diferenciar los servicios
(TOS)
    Private usTotalLength As UShort             '16 bits para la longitud total de el
datagrama (encabesado + mensaje)
    Private usIdentification As UShort          '16 bits para identificación
    Private usflagsAndOffset As UShort          'bit alto para bandera y fracmentación de
desvalance
    Private byTTL As Byte                      'bit alto para el subrrallado del protocolo
    Private byProtocol As Byte                  'contenido de 16 bits en el checksum de el
encabesado
    Private sChecksum As Short                  '(el checksum puede ser negativo tanto
tomado como corto)
    Private uiSourceIPAddress As UInteger       'Origen de dirección IP de 32 bits
    Private uiDestinationIPAddress As UInteger  'Destino de dirección IP de 32 bits
    'Fin de archivo de cabeseras IP
    Private byHeaderLength As Byte              'longitud del encabesado
    Dim recend As Integer = 4069                'carga de datos por el datagrama
```

```
    Private byIPDato() As Byte = New Byte(recend) {}
    Public Sub New(ByRef byBuffer As Byte(), ByRef nReceived As Integer)
        Try
            'Creación de MemoryStream salida de la recepción de bytes
            Dim memoryStream As New MemoryStream(byBuffer, 0, nReceived)

            'Después creamos un BinaryReader del MemoryStream
            Dim binaryReader As New BinaryReader(memoryStream)

            'Los ocho primeros bits del encabesado de IP contienen la versión
            'y la longitud del encabesado entonces los leemos
            byVersionAndHeaderLength = binaryReader.ReadByte()

            'Los ocho siguientes bits contienen los servicios Diferenciados
            byDifferentiatedServices = binaryReader.ReadByte()

            'Después ocho bits sostienen la longitud  total del datagrama
            usTotalLength =
CUShort(IPAddress.NetworkToHostOrder(binaryReader.ReadInt16()))

            'Los siguientes dieciséis bytes tienen los identificadores
            usIdentification =
CUShort(IPAddress.NetworkToHostOrder(binaryReader.ReadInt16()))

            'Los siguientes dieciéis bits contienen la compensación de fragmnetaión y las
banderas
            usflagsAndOffset =
CUShort(IPAddress.NetworkToHostOrder(binaryReader.ReadInt16()))

            'Los siguientes ocho bits tienen el valor TTL
            byTTL = binaryReader.ReadByte()

            'Los siguientes ocho representan el protocolo encapsulado en el datagrama
            byProtocol = binaryReader.ReadByte()

            'Los siguientes diesies bits contienen el ancabesaso de checksum
            sChecksum = IPAddress.NetworkToHostOrder(binaryReader.ReadInt16())

            'Los siguientes 32 bits contienen la IP origen
            uiSourceIPAddress = CUInt(binaryReader.ReadInt32())

            'Los siguientes 32 bits contienen la IP destino
            uiDestinationIPAddress = CInt(binaryReader.ReadInt32())


            'Ahora el calculo de la lomgitud de encabesado

            byHeaderLength = byVersionAndHeaderLength
```

```
        'Los cuatro últimos bits de la versión y el campo de longitud de el encabezado
contienen
        'la longitud del encabezado, realizamos una operación aritmetica binario simple
para extraerlos.

        byHeaderLength <<= 4
        byHeaderLength >>= 4

        'multiplicamos por 4 para obtener esactamente la longitud del encabezado
        byHeaderLength *= 4
        'Copia los datos llevados por el gramo de datos en otra serie de modo que
        'según el protocolo es llevado en el datagrama IP
        Array.Copy(byBuffer, _
                byHeaderLength, _
                byIPDato, 0, _
                usTotalLength - byHeaderLength)
      Catch ex As Exception
        MessageBox.Show(ex.Message, "Sniffer", MessageBoxButtons.OK,
MessageBoxIcon.Error)
      End Try
   End Sub
   Public ReadOnly Property Version() As String
      Get
        'calcula la verción de IP
        'el cuarto bit de el encabezado de IP contiene la verción de IP
        If (byVersionAndHeaderLength >> 4) = 4 Then
          Return "IP v4"
        ElseIf (byVersionAndHeaderLength >> 4) = 6 Then
          Return "IP v6"
        Else
          Return "Unknown"
        End If
      End Get
   End Property
   Public ReadOnly Property HeaderLength() As String
      Get
        Return byVersionAndHeaderLength.ToString()
      End Get
   End Property
   Public ReadOnly Property MessageLength() As UShort
      Get
        'Longitud de mensaje = Total de la longitud de el datagrama - la longitud del
encabezado
        Return usTotalLength - byHeaderLength
      End Get
   End Property
   Public ReadOnly Property DifferentiatedServices() As String
      Get
        'regresa los diferentes servicios en formato hexadecimal
        Return String.Format("0x{0:x2}({1})", byDifferentiatedServices, _
```

```
                    byDifferentiatedServices)
    End Get
  End Property
  Public ReadOnly Property Flags() As String
    Get
      'Los tres primeros bits de las banderas y el campo de fracmentación
      'representan las banderas (que indican si los datos son
      'fragmentados o no)
      Dim nFlags As Integer = usflagsAndOffset >> 13
      If nFlags = 2 Then
        Return "No fragmentado"
      ElseIf nFlags = 1 Then
        Return "Mas fragmentos en camino"
      Else
        Return nFlags.ToString()
      End If
    End Get
  End Property
  Public ReadOnly Property FragmentationOffset() As String
    Get
      'Los trece últimos bits de las banderas y el campo
      'de fragmentación contienen la compensación de fragmentación
      Dim nOffset As Integer = usflagsAndOffset << 3
      nOffset >>= 3
      Return nOffset.ToString()
    End Get
  End Property
  Public ReadOnly Property TTL() As String
    Get
      Return byTTL.ToString()
    End Get
  End Property
  Public ReadOnly Property ProtocoloType() As Protocolo
    Get
      'El campo de protocolo representa
      'el protocolo en la parte de datos del datagrama
      If byProtocol = 6 Then
        Return Protocolo.TCP
      ElseIf byProtocol = 17 Then
        Return Protocolo.UDP
      Else
        Return Protocolo.Unknown
      End If
    End Get
  End Property
  Public ReadOnly Property Checksum() As String
    Get
      'Regresa el checksum en formato hexadecimal
      Return String.Format("0x{0:x2}", sChecksum)
    End Get
```

```
    End Property
    Public ReadOnly Property SourceAddress() As IPAddress
        Get
            Return New IPAddress(uiSourceIPAddress)
        End Get
    End Property
    Public ReadOnly Property DestinationAddress() As IPAddress
        Get
            Return New IPAddress(uiDestinationIPAddress)
        End Get
    End Property
    Public ReadOnly Property TotalLength() As String
        Get
            Return usTotalLength.ToString()
        End Get
    End Property
    Public ReadOnly Property Identification() As String
        Get
            Return usIdentification.ToString()
        End Get
    End Property
    Public ReadOnly Property Dato() As Byte()
        Get
            Return byIPDato
        End Get
    End Property
End Class
```

```
'Código TCPHeader
'Ing. Jonatan Juárez Hinojosa
'Función que obtiene la información del enbezado TCP
'-------------------------------------------------------
Imports System.Net
Imports System.Text
Imports System
Imports System.IO
Imports System.Windows.Forms

Public Class TCPHeader
    'Archivos de cabecera TCP
    Private usSourcePort As UShort                  '16 bits para el numero de puerto fuente
    Private usDestinationPort As UShort             '16 bits para el numero de puerto
destino
    Private uiSequenceNumber As UInteger = 555      '32 bits para el número de
secuencia
    Private uiAcknoledgementNumber As UInteger = 555    '32 bits para el número de
acknowledgement
    Private usDataOffsetEndFlags As UShort = 555    '16 bits para la bandera y datos
de desvalance
    Private usWindows As UShort = 555               '16 bits para el espacio de ventana
    Private sChecksum As Short                      '16 bits para el checksum
    '(checksum puede ser negativo o corto)
    Private usUrgentPoniter As UShort               '16 bits para el punto de urgencia
    'Fin de los archivos de cabecera TCP

    Private byHeaderLength As Byte                  'Longitud del ancabezado
    Private usMessageLength As UShort               'Longitud del dato que comienza el
acarreo
    Private byTCPData(0 To 4096 - 1) As Byte        'Dato acarreado por el paquete de
TCP
    Public Sub New(ByVal byBuffer() As Byte, ByVal nReceiver As Integer)
        Try
            Dim memoryStream As New MemoryStream(byBuffer, 0, nReceiver)
            Dim binaryReader As New BinaryReader(memoryStream)

            'Los primeros 16 bits contienen el puerto origen
            usSourcePort =
CUShort(IPAddress.NetworkToHostOrder(binaryReader.ReadInt16()))

            'Los siguientes 16 contienen el puerto destino
            usDestinationPort =
CUShort(IPAddress.NetworkToHostOrder(binaryReader.ReadInt16()))

            'Los siguientes 32 contienen el número de secuencia
            uiSequenceNumber =
CUInt(IPAddress.NetworkToHostOrder(binaryReader.ReadInt32()))

            'Los siguientes 32 contienen el número de acknowledgement
```

```
            uiAcknoledgementNumber =
CUInt(IPAddress.NetworkToHostOrder(binaryReader.ReadInt32()))

            'Los siguientes 16 bits sostienen las banderas de compensación de datos
            usDataOffsetEndFlags =
CUShort(IPAddress.NetworkToHostOrder(binaryReader.ReadInt16()))

            'Los siguientes 16 bits contienen el espacio de ventana
            usWindows =
CUShort(IPAddress.NetworkToHostOrder(binaryReader.ReadInt16()))

            'En los siguientes 16 es donde se tiene el checksum
            sChecksum =
CUShort(IPAddress.NetworkToHostOrder(binaryReader.ReadInt16()))

            'Los 16 siguientes contienen el punto de urgencia
            usUrgentPoniter =
CUShort(IPAddress.NetworkToHostOrder(binaryReader.ReadInt16()))

            'El dato de compensación indica donde los datos comienzan,
            'entonces se utilizan estos para calcular la longitud de la cabeza
            byHeaderLength = CByte((usDataOffsetEndFlags >> 12))
            byHeaderLength *= 4

            'Longitud de mensaje = longitud total de el paquete TCP - longitud del
ancabezado
            usMessageLength = CUShort(nReceiver - byHeaderLength)

            'Copia el dato TCP dentro del dato de buffer
            Array.Copy(byBuffer, byHeaderLength, byTCPData, 0, nReceiver -
byHeaderLength)

        Catch ex As Exception
            MessageBox.Show(ex.Message, "Usmeador TCP" & (nReceiver),
MessageBoxButtons.OK, MessageBoxIcon.Error)
        End Try
    End Sub
    Public ReadOnly Property SourcePort() As String
        Get
            Return usSourcePort.ToString()
        End Get
    End Property
    Public ReadOnly Property DestinationProt() As String
        Get
            Return usDestinationPort.ToString()
        End Get
    End Property
    Public ReadOnly Property SequenceNumber() As String
        Get
            Return uiSequenceNumber.ToString()
```

```
    End Get
End Property
Public ReadOnly Property AcknowledgementNumber() As String
    Get
        'Si la bandera de ACK es puesta entonces sólo tenemos un valor
        'válido en el campo de reconocimiento, entonces la comprobación
        'no regresa nada.
        If (usDataOffsetEndFlags And &H10) <> 0 Then
            Return uiAcknoledgementNumber.ToString()
        Else
            Return ""
        End If
    End Get
End Property
Public ReadOnly Property HeaderLength() As String
    Get
        Return byHeaderLength.ToString()
    End Get
End Property
Public ReadOnly Property WindowsSize() As String
    Get
        Return usWindows.ToString()
    End Get
End Property
Public ReadOnly Property UrgentPoint() As String
    Get
        'Si la bandera de URG es puesta entonces sólo tenemos un valor
        'válido en el campo de indicadores urgente,
        'entonces la comprobación en el devuelve un resultado
        If (usDataOffsetEndFlags And &H20) <> 0 Then
            Return usUrgentPoniter.ToString()
        Else
            Return ""
        End If
    End Get
End Property
Public ReadOnly Property Flags() As String
    Get
        'Los seis últimos bits de compensación de datos y
        'banderas contienen bits de control

        'Primer extremo de las banderas
        Dim nFlags As Integer = usDataOffsetEndFlags And &H3F

        Dim strFlags As String = String.Format("0x{0:x2} (", nFlags)

        'Ahora comenzamos a mirar cada uno de los bits si son colocados o no
        If (nFlags And &H1) <> 0 Then
            strFlags += "FIN, "
        End If
```

```
            If (nFlags And &H2) <> 0 Then
                strFlags += "SYN, "
            End If
            If (nFlags And &H4) <> 0 Then
                strFlags += "RST, "
            End If
            If (nFlags And &H8) <> 0 Then
                strFlags += "PSH, "
            End If
            If (nFlags And &H10) <> 0 Then
                strFlags += "ACK, "
            End If
            If (nFlags And &H20) <> 0 Then
                strFlags += "URG"
            End If
            strFlags += ")"
            If strFlags.Contains("()") Then
                strFlags = strFlags.Remove(strFlags.Length - 3)
            ElseIf strFlags.Contains(", )") Then
                strFlags = strFlags.Remove(strFlags.Length - 3, 2)
            End If
            Return strFlags
        End Get
    End Property
    Public ReadOnly Property Checksum() As String
        Get
            'Regresa el checksum en formato hexadecimal
            Return String.Format("0x{0:x2}", sChecksum)
        End Get
    End Property
    Public ReadOnly Property Data() As Byte()
        Get
            Return byTCPData
        End Get
    End Property
    Public ReadOnly Property MessageLength() As UShort
        Get
            Return usMessageLength
        End Get
    End Property
End Class
```

```
'Código UDPHeader
'Ing. Jonatan Juárez Hinojosa
'Función que obtiene la información del enbezado UDP
'-------------------------------------------------
Imports System.Net
Imports System.Text
Imports System
Imports System.IO
Imports System.Windows.Forms

Public Class UDPHeader
   'Archivos de cabecera UDP
   Private usSourcePort As UShort          '16 bits para la número del puerto de origen
   Private usDestinatinationPort As UShort '16 bits para el número de puerto de destino
   Private usLength As UShort              'Longitud del encabezado de UDP
   Private SChecksum As UShort             '16 bits para el checksum (checksum puede ser
negativo entonces tambien pequeño)
   'Fin de archivos de cabecera UDP

   Private byUDPData(0 To 4096 - 1) As Byte 'Dato de acarreo por el paquete de UDP
   Public Sub New(ByVal byBuffer() As Byte, ByVal nReceived As Integer)
      Dim memoryStream As New MemoryStream(byBuffer, 0, nReceived)
      Dim binaryReader As New BinaryReader(memoryStream)

      '16 primeros bits contiene el puerto de origen
      usSourcePort =
CUShort(IPAddress.NetworkToHostOrder(binaryReader.ReadInt16()))

      'Los siguientes 16 bits contienen el puerto de destino
      usDestinatinationPort =
CUShort(IPAddress.NetworkToHostOrder(binaryReader.ReadInt16()))

      'Los siguientes 16 bits contienen la longitud de elpqute de UDP
      usLength = CUShort(IPAddress.NetworkToHostOrder(binaryReader.ReadInt16()))

      'Los siguientes 16 bits contienen el checksum
      SChecksum = IPAddress.NetworkToHostOrder(binaryReader.ReadInt16())

      'Copia el dato acarreado por el paquete de UDP dentro del dato  de buffer
      'El cabezal de UDP es de 8 Bytes entonces comenzamos a copiar cada uno de ellos
      Array.Copy(byBuffer, 8, byUDPData, 0, nReceived - 8)
   End Sub
   Public ReadOnly Property SourcePort() As String
      Get
         Return usSourcePort.ToString()
      End Get
   End Property
   Public ReadOnly Property DestinatinationPort() As String
      Get
         Return usDestinatinationPort.ToString()
```

```
        End Get
    End Property
    Public ReadOnly Property Lengh() As String
        Get
            Return usLength.ToString()
        End Get
    End Property
    Public ReadOnly Property Checksum() As String
        Get
            'Regresa el checksum en formato hexadecimal
            Return String.Format("0x{0:x2}", SChecksum)
        End Get
    End Property
    Public ReadOnly Property Data() As Byte()
        Get
            Return byUDPData
        End Get
    End Property
End Class
```

```
'Código DNSHeader
'Ing. Jonatan Juárez Hinojosa
'Función que obtiene la información del enbezado DNS
'-------------------------------------------------
Imports System.Net
Imports System.Text
Imports System
Imports System.IO
Imports System.Windows.Forms
Imports System.Collections.Specialized
Imports System.Collections
Imports System.Collections.Generic

Public Class DNSHeader
   'Campo de cabecera DNS
   Private usIdentification As UShort      '16 bits para identificación
   Private usFlags As UShort               '16 bits para banderas DNS
   Private usTotalQuestions As UShort      '16 bits para idicar el numero de entradas
   'en la lista de consultas
   Private usTotalAnswerRRs As UShort      '16 bits para indicar el número de entradas
   'en la respuesta de entrada en el registro de la lista
   Private usTotalAuthorityRRs As UShort   '16 bits para indicar el número de entradas
   'en la autorización de entrada de la fuente en el registro de lista
   Private usTotalAdditionalRRs As UShort  '16 bits para indicar el numero de entradas
   'entradas en el registro adicional de la lista
   'Fin de campo cabecera DNS
   Public Sub New(ByVal byBuffer() As Byte, ByVal nReceived As Integer)
      Dim memoryStream As New MemoryStream(byBuffer, 0, nReceived)
      Dim binaryReader As New BinaryReader(memoryStream)

      'Los primeros 16 bits son para la identificación
      usIdentification =
CUShort(IPAddress.NetworkToHostOrder(binaryReader.ReadInt16()))

      'Los siguientes 16 contienen las banderas
      usFlags = CUShort(IPAddress.NetworkToHostOrder(binaryReader.ReadInt16()))

      'Lee los números totales de consultas en la lista de consulta
      usTotalQuestions =
CUShort(IPAddress.NetworkToHostOrder(binaryReader.ReadInt16()))

      'Lee el número total de de respuestas en la lista de respuestas
      usTotalAnswerRRs =
CUShort(IPAddress.NetworkToHostOrder(binaryReader.ReadInt16()))

      'Lee el número total de la lista de autorizaciones
      usTotalAuthorityRRs =
CUShort(IPAddress.NetworkToHostOrder(binaryReader.ReadInt16()))

      'Lee el número de entradas en la fuente adicional de la lista de registros
```

```
        usTotalAdditionalRRs =
CUShort(IPAddress.NetworkToHostOrder(binaryReader.ReadInt16()))
    End Sub
    Public ReadOnly Property Identification() As String
        Get
            Return String.Format("0x{0:x2}", usIdentification)
        End Get
    End Property
    Public ReadOnly Property Flags() As String
        Get
            Return String.Format("0x{0:x2}", usFlags)
        End Get
    End Property
    Public ReadOnly Property TotalQuestion() As String
        Get
            Return usTotalQuestions.ToString()
        End Get
    End Property
    Public ReadOnly Property TotalAnswerRRs() As String
        Get
            Return usTotalAnswerRRs.ToString()
        End Get
    End Property
    Public ReadOnly Property TotalAuthorityRRs() As String
        Get
            Return usTotalAuthorityRRs.ToString()
        End Get
    End Property
    Public ReadOnly Property TotalAdditionalRRs() As String
        Get
            Return usTotalAdditionalRRs.ToString()
        End Get
    End Property
End Class
```

Apéndice “E”

Código detección de paquetes de entrada / salida

Éste apéndice contiene el código fuente de cada uno de los módulos estadísticas, junto con su ventana de windows form, el lenguaje de programación que se utiliza es visual basic.NET.

```
'Código Estadísticas
'Ing. Jonatan Juárez Hinojosa
'Obtiene el número de paquetes de entrada, salida, con errores, etc...
'-------------------------------------------------------
Imports System.Net
Imports System.Net.NetworkInformation
Imports System.Net.Sockets

Public Class Estadistica

    Private Sub Estadistica_load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
        Dim Datos() As String
        With IPGlobalProperties.GetIPGlobalProperties()
            With .GetIPv4GlobalStatistics
                Datos = Split( _
                        "Paquetes de entrada:" & Format(.ReceivedPackets) & " " & _
                        "Paquetes de salida:" & Format(.OutputPacketRequests) & " " & _
                        "Intervalo maximo de fragmento de IP:" &
Format(.PacketReassemblyTimeout) & " " & _
                        "Paquete entregados:" & Format(.ReceivedPacketsDelivered) & " " &
_
                        "Paquete recibidos y descartados:" &
Format(.ReceivedPacketsDiscarded) & " " & _
                        "Paquete errores de dirección:" &
Format(.ReceivedPacketsWithAddressErrors) & " " & _
                        "Paquetes de errores de encabezado:" &
Format(.ReceivedPacketsWithHeadersErrors))
                For Each Cadena As String In Datos
                    ListBox1.Items.Add(Cadena)
                Next
            End With
```

```
        End With
    End Sub
End Class
```

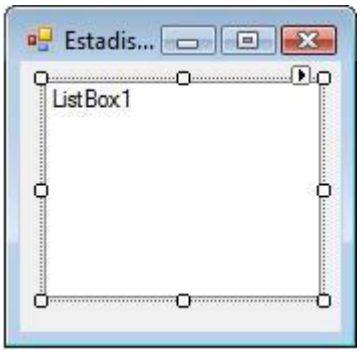


Figura E.1 – Ventana de windows form del programa estadísticas.

Apéndice “F”

Código de comprobación de conexión (PING)

Éste apéndice contiene el código fuente de cada uno de los modulos HacerPING, junto con su ventana de windows form, el lenguaje de programación que se utiliza es visual basic.NET.

```
'Código Hacer PING
'Ing. Jonatan Juárez Hinojosa
'Envía un mensaje de solicitud de conexión a un equipo remoto utilizando el protocolo ICMP
'-------------------------------------------------------
Imports System.Net.NetworkInformation
Imports System.Net
Imports System.Net.IPHostEntry

Public Class HacerPing
    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click
        With New NetworkInformation.Ping ' creación del objeto ping
            AddHandler .PingCompleted, AddressOf Pingcompletado 'asociación al método PingCompleted, con Pingcompletado
            .SendAsync(TextBox1.Text, TextBox1.Text) 'Comprobación de manera cincrona
        End With
    End Sub
    Public Sub Pingcompletado(ByVal sender As Object, ByVal e As PingCompletedEventArgs) Handles ListView1.SelectedIndexChanged
        Dim l, w, I As Integer 'declaración de variables
        l = NumericUpDown2.Value 'l= tamaño del buffer
        w = NumericUpDown3.Value 'w= Tiempo de espera
        For I = 1 To NumericUpDown1.Value 'Resaliza el numero de solicitudes de conexión
            Try                         'Intrucción para realizar esepciones
                If NumericUpDown1.Value > 4 Then 'Si el numero de echo es mayor a 4, se genera los echos del NumericUpDown1
                    With ListView1.Items.Add(e.UserState.ToString) 'Entrega nombre o dirección IP
                        .SubItems.Add(e.Reply.Address.ToString)    'Entrega la dirección IP
                        .SubItems.Add(e.Reply.Status.ToString)     'Entrega suficiente o tiempo terminado de conexión
```

```
                .SubItems.Add(e.Reply.Buffer.Length.ToString(l)) 'Entrega el numero
de bytes del buffer
                .SubItems.Add(e.Reply.RoundtripTime.ToString(w)) 'Entrega el tiempo
de conexión
                .SubItems.Add(e.Reply.Options.Ttl)
            End With
          Else              'De lo contrario solo ejecuta el ping de forma estandar
            With ListView1.Items.Add(e.UserState.ToString)
              .SubItems.Add(e.Reply.Address.ToString)
              .SubItems.Add(e.Reply.Status.ToString)
              .SubItems.Add(e.Reply.Buffer.Length.ToString)
              .SubItems.Add(e.Reply.RoundtripTime.ToString)
              .SubItems.Add(e.Reply.Options.Ttl)
            End With
          End If
        Catch                'Si no encuentra el equipo activo
          If I < 4 Then      'Realiza tres intentos de conexción
            MsgBox("Host de destino inaccesible.")
          Else               'Si el equipo no respondio a la conexión
            MsgBox("Sea terminado el tiempo de espera del host.")
          End If
        End Try
      Next
    End Sub
End Class
```

Figura F.1 Ventana de Windows from del programa hacer PING.

Apéndice “G”

Código de adaptadores de red

Éste apéndice contiene el código fuente de cada uno de los módulos detección de interfaces de red, junto con su ventana de windows form, el lenguaje de programación que se utiliza es visual basic.NET.

```
'Código Detección de interfaces de red
'Ing. Jonatan Juárez Hinojosa
'Detecta las características de cada una de las interfaces de red instaladas dentro de la computadora
'como el identificador, descripción, nombre del adaptador, tipo y estado.
'-----------------------------------------------------------
Imports System.Net.NetworkInformation

Public Class AdapRed

    Private Sub ListView1_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load
        'Recorremos la colección de adaptadores
        For Each Adaptador As NetworkInterface In NetworkInterface.GetAllNetworkInterfaces
            'Añadiendo por cada uno de ellos una fila a la tabla de listview
            With ListView1.Items.Add(Adaptador.Id.ToString)
                'con el nombre del adaptador
                .SubItems.Add(Adaptador.Name)
                'su descripción
                .SubItems.Add(Adaptador.Description)
                'tipo de adaptador
                .SubItems.Add(Adaptador.NetworkInterfaceType.ToString)
                'velocidad en kilo bits por segundo
                .SubItems.Add(Format(Adaptador.Speed / 1000, "#,## Kbits/s"))
                'dirección MAC
                .SubItems.Add(Adaptador.GetPhysicalAddress().ToString)
                Dim Direcciones As String = ""
                'obtención de la dirección IP física
                For Each Direccion As MulticastIPAddressInformation In _
                Adaptador.GetIPProperties().MulticastAddresses
                    'Las concatenamos en una cadena
                    Direcciones &= Direccion.Address.ToString & " - "
                Next
                .SubItems.Add(Direcciones)
```

```
            'Añadimos el estado operacional
            .SubItems.Add(Adaptador.OperationalStatus.ToString)
        End With
    Next
  End Sub
End Class
```

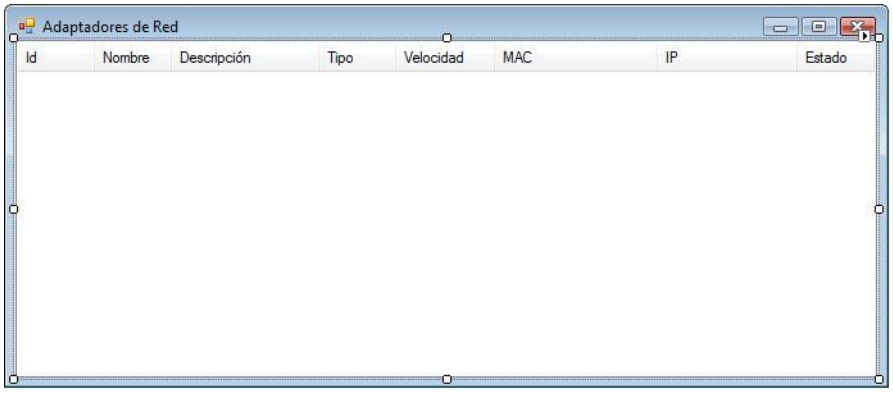


Figura G.1 – Ventana de windows form del programa detección de interfaces de red.

Apéndice “H”

Código genera datos

En éste apéndice se encuentra el código fuente de la aplicación “Generar datos”, el cual se muestra a continuación:

```
Public Class Form1
Private textDialog As New SaveFileDialog
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
        Dim Cont1 As Integer
        Dim Cont2 As Integer = 0
        Dim Cont3 As Integer = 0
        Dim Resultado, Mensaje As String

        textDialog.Filter = "Archivo de texto (*.txt)|*.txt"
'Espesifica el tipo de archivo a crear
        textDialog.ShowDialog()  'Abre el cuadro de dialogo para
colocar la ruta donde se guarda el archivo
        For Cont1 = 1 To 5000000
        Resultado = Resultado & vbTab & "1" & vbTab & "1" & vbTab &
        "1" & vbCrLf
        Cont2 += 1
        If Cont2 = 500000 Then
           If textDialog.FileName <> "" Then
               FileOpen(1, textDialog.FileName, OpenMode.Output)
               PrintLine(1, Resultado)'copia cada linea del texto
                FileClose(1)
            End If
            Cont3 += Cont2
            Label1.Text = "Número de registros: " & Cont3
            MsgBox("Registros escritos")
            Cont2 = 0
        End If
    Next
    Mensaje = "Generación de datos Finalizada"
    MsgBox(Mensaje)
 End Sub
End Class
```

Apéndice I

Manual de usuario

1) Se activa la aplicación del tablero de control de análisis de red.
2) Aparece en pantalla la aplicación como lo muestra la imagen I.1

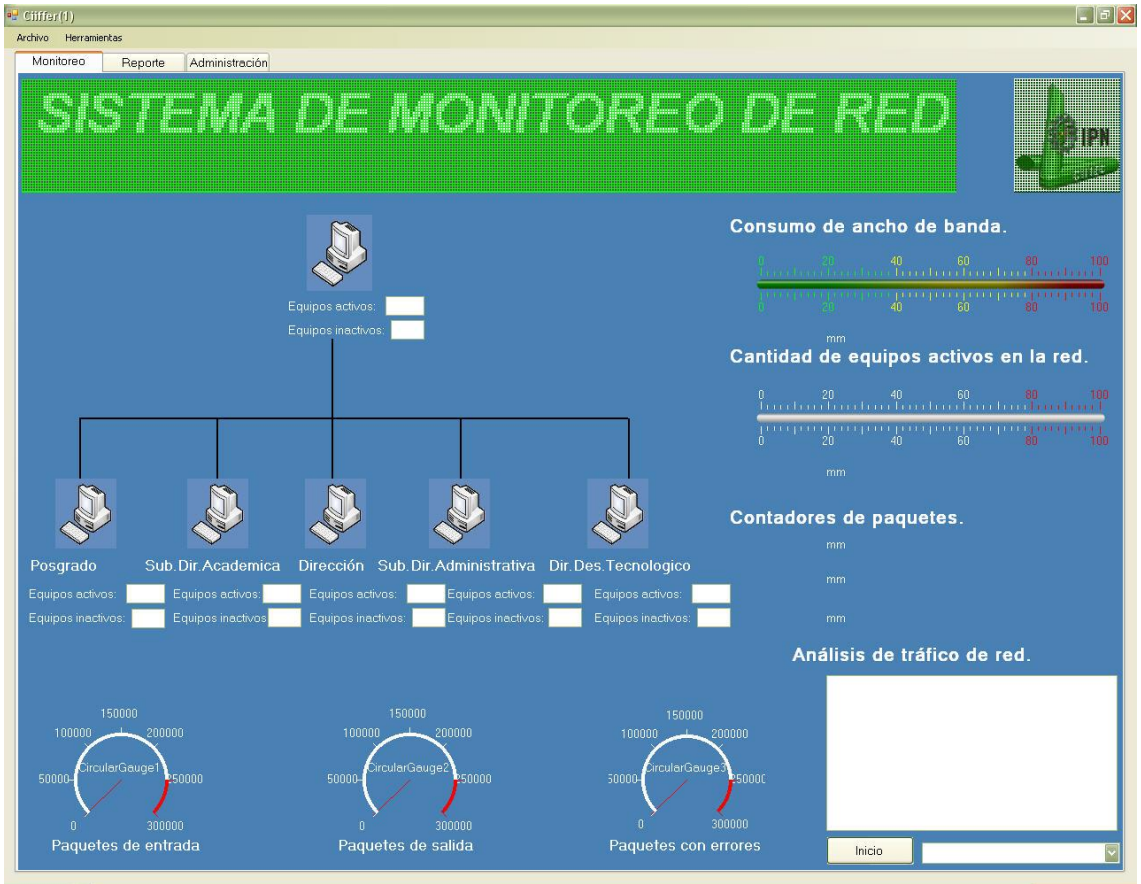


Figura I.1 ventana principal del tablero de control

Las pestañas de reporte y administración se encuentran bloqueadas.

3) Active la pestaña “Administración” (Fig I.2) e inicie cesión como administrador, para desbloquear la pestaña.

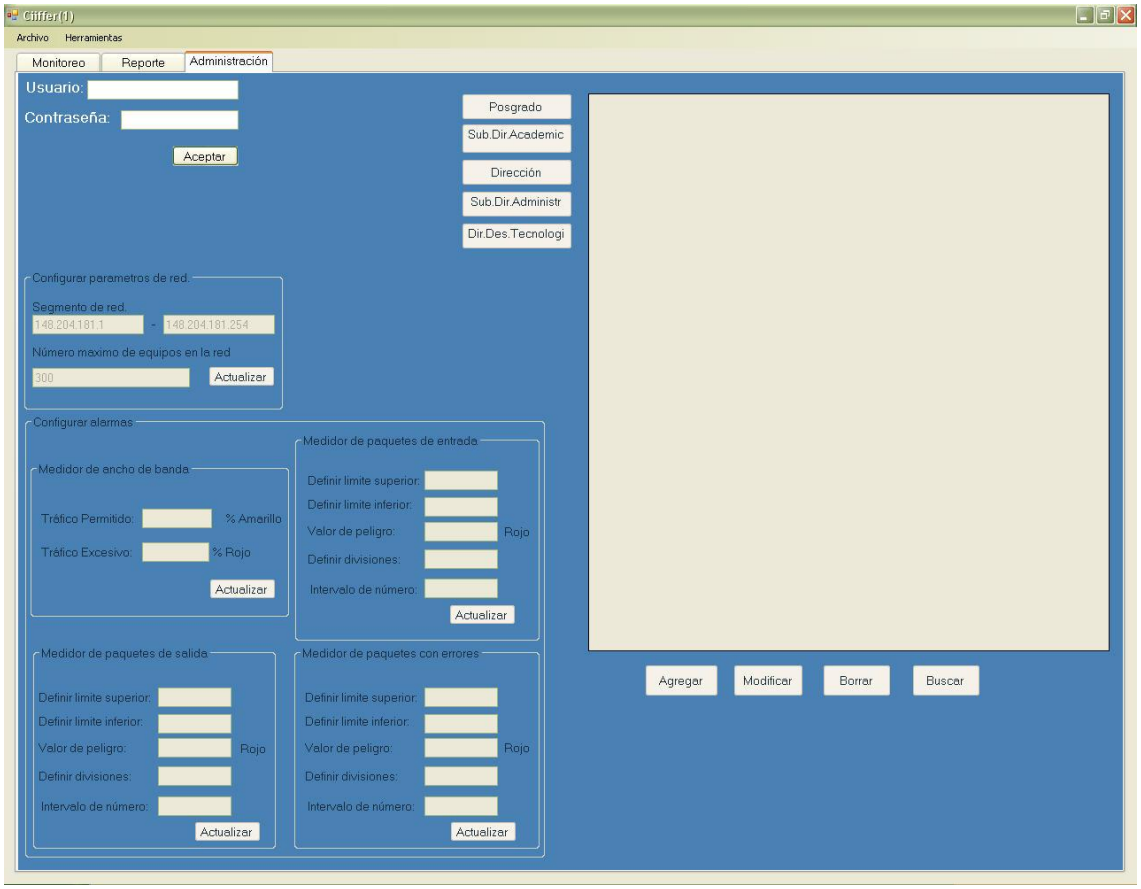


Figura I.2 Pestaña administración

- Usuario: Administrador
- Contraseña: Admin

4) Una vez que se activaron las funciones de la pestaña de administración configure el segmento y el número de computadoras de red a monitorear (Fig I.3) y active el botón actualizar.

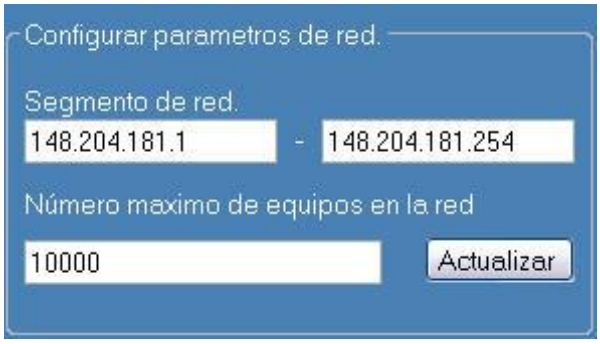


Figura I.3 Sección configurar parámetros de red

5) Active la pestaña "Monitoreo"

6) Dentro de la pestaña "Monitoreo" en la seleccione análisis de tráfico de red, seleccione una interfaz de red (Fig I.4) y active el botón "Inicio", este botón cambia de estado a "alto" y activa la pestaña reporte, después la aplicación comienza a detectar el tráfico de red.

Figura I.4 Sección análisis de tráfico de red.

7) Se empieza a detectar el tráfico de de red como lo muestra la figura I.5.

Figura I.5 Detección del tráfico de red

8) Se empieza a detectar el tipo de protocolos y el número de veces que se utiliza como lo muestra la figura I.6

Figura I.6 Sección contadores de paquetes

9) Se detecta la cantidad de paquete de entrada, salida y errores (Fig. I.7)

Figura I.7 Sección de medidores de paquetes

10) Se detecta el consumo de ancho de banda y la velocidad con la que viajan los paquetes de datos como lo indica la figura I.8.

Figura I.8 Medidor de consumo de ancho de banda

11) SE detecta el número y el porcentaje de equipos activos en la red (Fig I.9).

Figura I.9 Medidor de equipos activos en el segmento de red.

12) Haga clic sobre la imagen con una computadora para visualizar la cantidad de equipos activo e inactivos dentro cada departamento (fig. I.10)

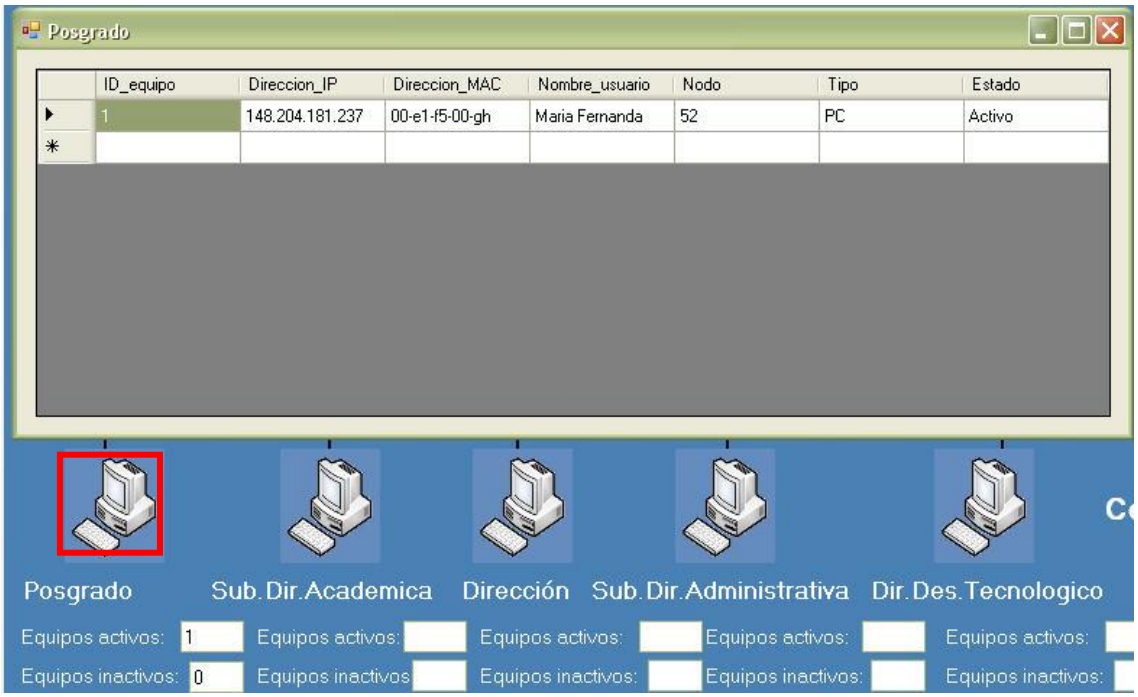
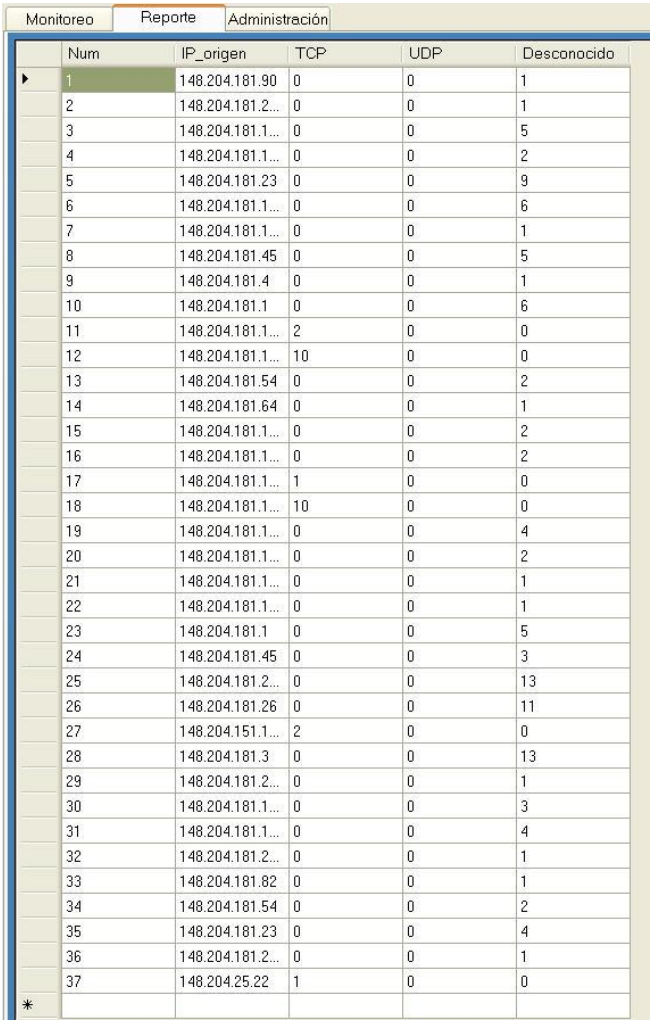


Figura I.10 Detección de equipos activos por departamento

13) Active la pestaña “reporte”, para mostrar la información del comportamiento de red (Fig I.11)



Num	IP_origen	TCP	UDP	Desconocido
1	148.204.181.90	0	0	1
2	148.204.181.2...	0	0	1
3	148.204.181.1...	0	0	5
4	148.204.181.1...	0	0	2
5	148.204.181.23	0	0	9
6	148.204.181.1...	0	0	6
7	148.204.181.1...	0	0	1
8	148.204.181.45	0	0	5
9	148.204.181.4	0	0	1
10	148.204.181.1	0	0	6
11	148.204.181.1...	2	0	0
12	148.204.181.1...	10	0	0
13	148.204.181.54	0	0	2
14	148.204.181.64	0	0	1
15	148.204.181.1...	0	0	2
16	148.204.181.1...	0	0	2
17	148.204.181.1...	1	0	0
18	148.204.181.1...	10	0	0
19	148.204.181.1...	0	0	4
20	148.204.181.1...	0	0	2
21	148.204.181.1...	0	0	1
22	148.204.181.1...	0	0	1
23	148.204.181.1	0	0	5
24	148.204.181.45	0	0	3
25	148.204.181.2...	0	0	13
26	148.204.181.26	0	0	11
27	148.204.151.1...	2	0	0
28	148.204.181.3	0	0	13
29	148.204.181.2...	0	0	1
30	148.204.181.1...	0	0	3
31	148.204.181.1...	0	0	4
32	148.204.181.2...	0	0	1
33	148.204.181.82	0	0	1
34	148.204.181.54	0	0	2
35	148.204.181.23	0	0	4
36	148.204.181.2...	0	0	1
37	148.204.25.22	1	0	0

Figura I.11 Vista del comportamiento de red.

14) Active el botón generar reporte (fig I.12) para crear un formato de reporte el cual se puede ser impreso o almacenado como un archivo (fig I.13).

Figura I.12 Botón generar reporte

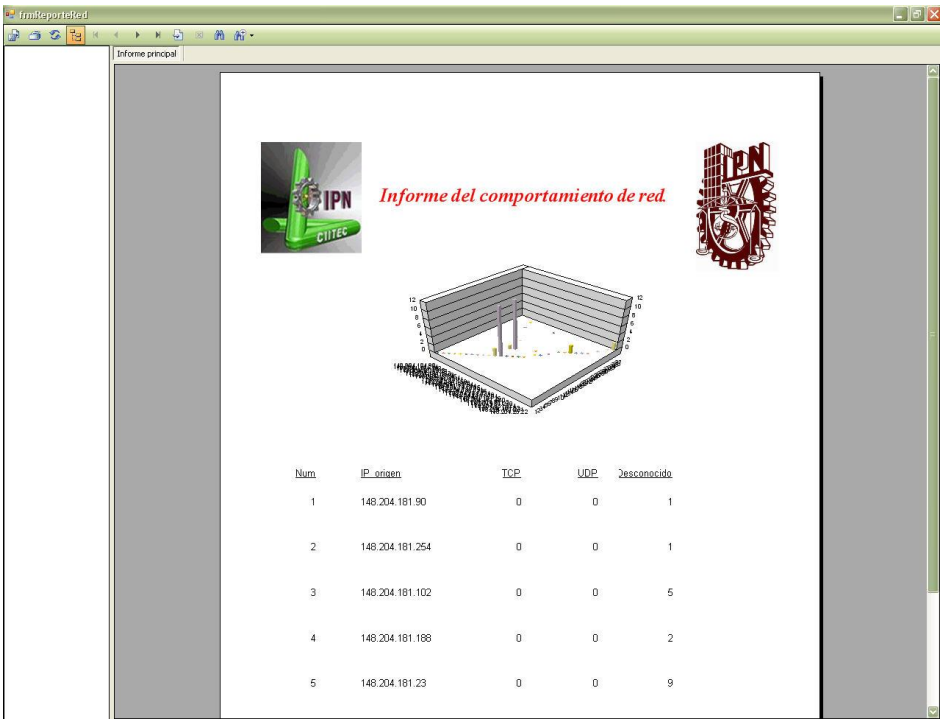


Figura I.13 Reporte del comportamiento de red

15) Active nuevamente la pestaña de administración (fig I.14), en esta sección se permite agregar un nuevo nodo en la red (fig I.15), modificarlo (Fig I.16), buscarlo (fig I.17) o Borrarlo (Fig I.18) activando cada uno delos botones en la pestaña de administración.

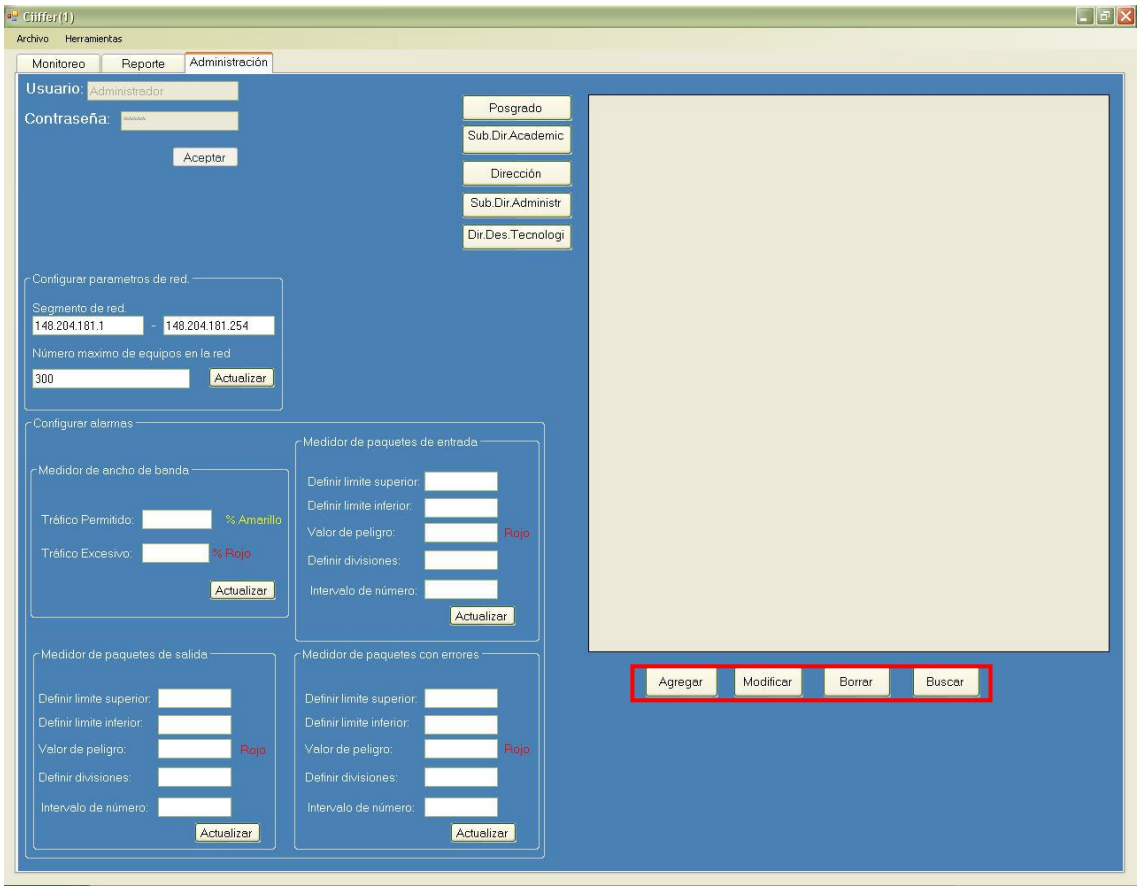


Figura I.14 Pestaña de administración

Figura I.15 Ventana agregar nodo

Figura I.16 Ventana modificar nodo

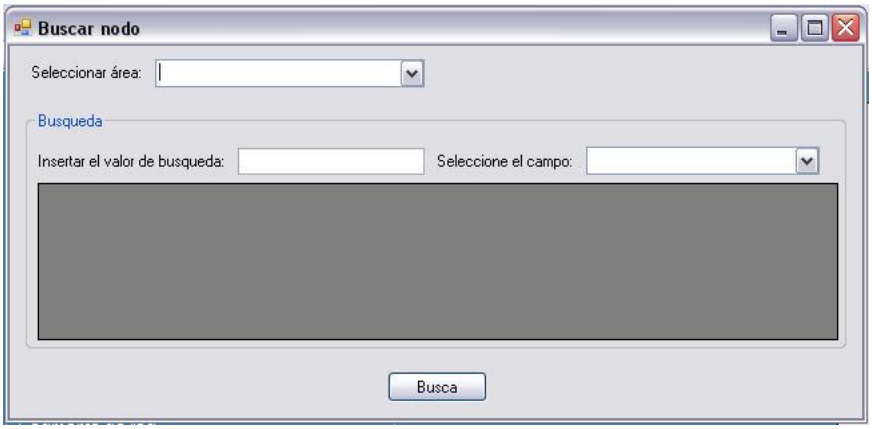


Figura I.17 Ventana buscar nodo

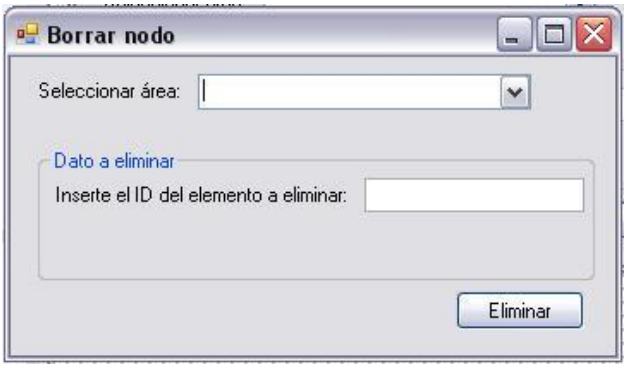


Figura I.18 Ventana borrar nodo

16) Se puede configurar el nivel de alarma del medidor de ancho de banda y de los medidores de paquetes de

datos de entrada, salida y errores en la sección configurar a alarmas (Fig I.19).

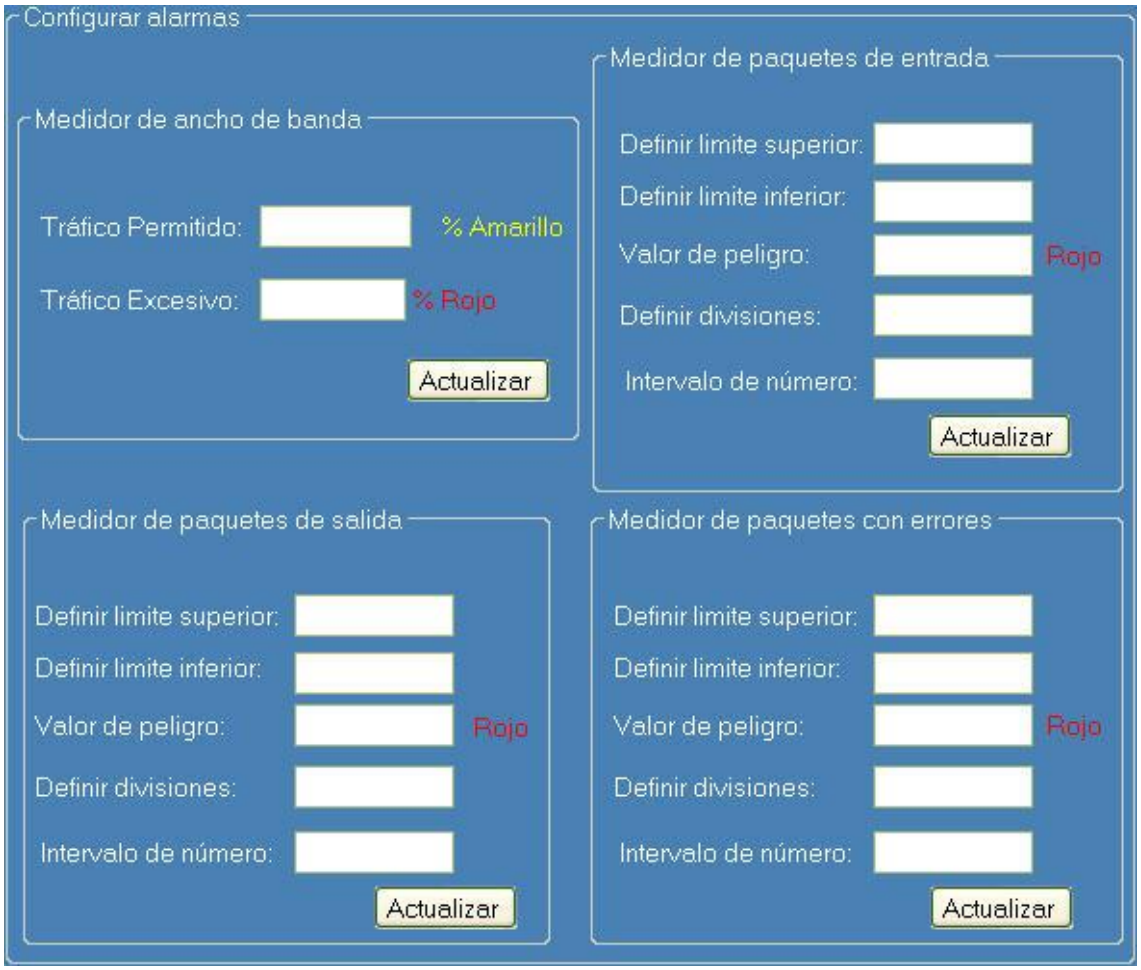


Figura I.19 Sección configurar alarmas dentro de la pestaña de administración

17) De la barra de menú active herramientas (fig I.20) para poder acceder a las herramientas de detección de interfaces de red (fig I.21), comprobación de conexión PING (fig I.22) y estadísticas de paquetes (fig I.23).

Figura I.20 Menú herramientas.

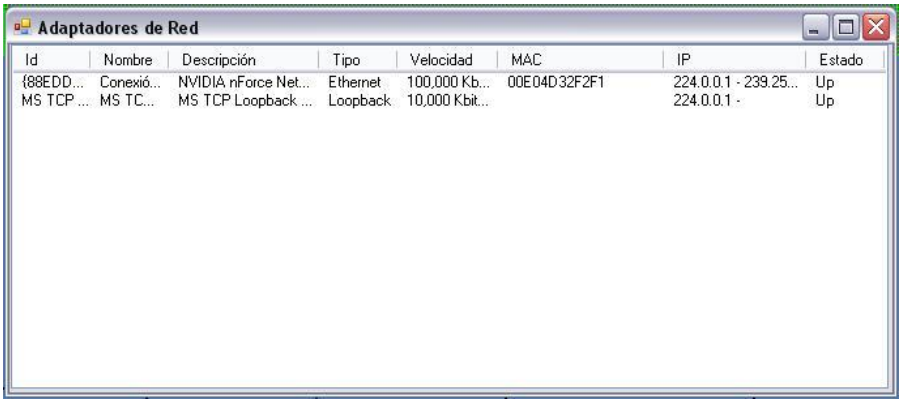


Figura I.21 Aplicación adaptadores de red.

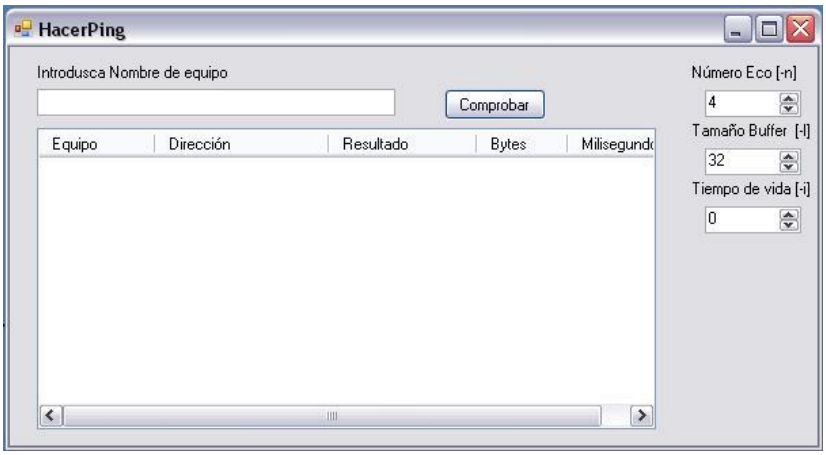


Figura I.22 Aplicación comprobación de conexión (HacerPing)

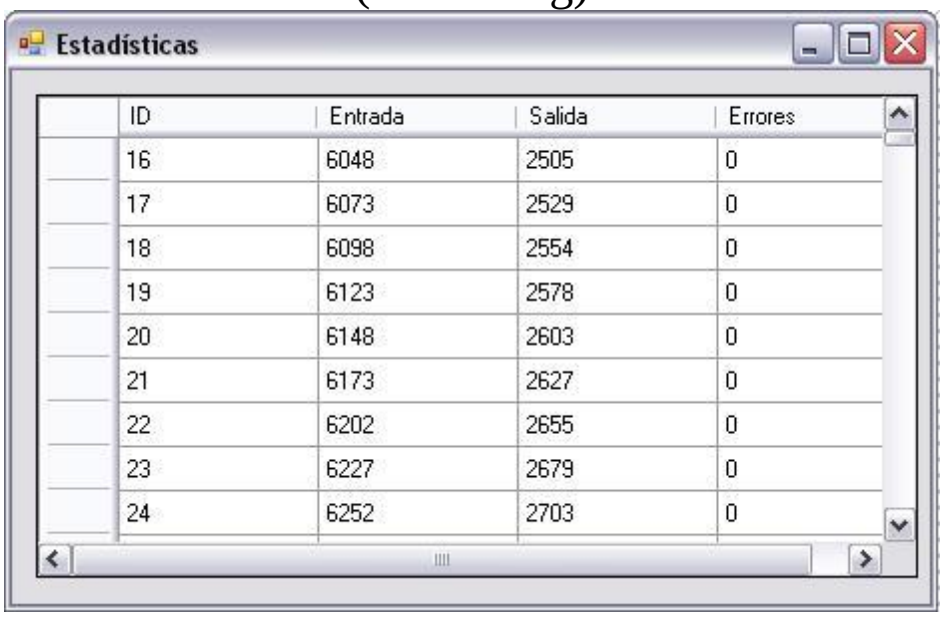


Figura I.23 Aplicación estadísticas de paquetes

18) Active el botón alto para detener la ejecución de la aplicación.

19) Active el menú archivo y seleccione la opción “salir” para cerrar la aplicación (fig I.24).

Figura I.24 Menú archivo con la opción salir.

Apéndice “J”

Esquemas de la familia de direcciones

Esquemas de direcciones	Usos
AddressFamily.AppleTalk	Dirección AppleTalk, usada para comunicaciones con ordenadores de Apple Macintosh.
AddressFamily.Atm	Modo de transferencia nativo asincrónico (ATM) dirección de servicios.
AddressFamily.Banyan	Direcciones Banyan VIDES (Sistema Virtual Conectado a una red).
AddressFamily.Ccitt	Direcciones para protocolo con X.25.
AddressFamily.Chaos	Protocolos de direcciones CHAOS, en formato 007.x.y.z.
AddressFamily.Cluster	Dirección para productos de cluster de Microsoft, como MSCS.
AddressFamily.Datakit	Dirección para protocolo Datakit, como protocolo de receptor universal.
AddressFamily.DataLink	Enlace de transmisión directo (MAC) dirección de interfaz.
AddressFamily.DecNet	Dirección DECnet, diseñada para miniordenadores DEC.
AddressFamily.Ecma	Dirección de Asociación de Fabricantes de Ordenador Europea (ECMA), usada para control de cambio de llamada por circuito.
AddressFamily.FireFox	Dirección IP, se ejecuta sobre TCP 1689
AddressFamily.HyperChannel	Dirección de hipercanal NSC, definida en el RFC 1044.
AddressFamily.Ieee12844	IEEE 1284.4 dirección de grupo de trabajo, comúnmente sabida como DOT4 y usado por impresoras de HP.
AddressFamily.ImpLink	ARPANET dirección de mensaje de proceso de interfaz (MAC).
AddressFamily.InterNetwork	Dirección IPv4, el más comúnmente usada para transferencias de Internet.
AddressFamily.InterNetworkV6	Dirección IPv6, usada para la siguiente versión de IP.
AddressFamily.Ipx	IPv6 dirección, usada para la siguiente versión de IP.
AddressFamily.Irda	Dirección de asociación de datos infrarroja.
AddressFamily.Iso	Dirección para protocolo de ISO, como ISO-IP.

AddressFamily.Lat	Dirección de protocolo de transporte de área local, usada con DEC de miniordenadores.
AddressFamily.Max	Máximo de direcciones.
AddressFamily.NetBios	Dirección NetBios, usada para compartir archivo de Windows e impresora.
AddressFamily.NetworkDesigners	Dirección para Diseñadores de Red OSI protocolos permitidos de entrada.
AddressFamily.NS	Dirección para Xerox con protocolo NS, como IDP.
AddressFamily.Pup	Dirección para paquete (PUP) universal de protocolos PARC.
AddressFamily.Sna	Dirección de Arquitectura de Red de Sistemas de IBM.
AddressFamily.Unix	Dirección local de maquina UNIX
AddressFamily.VoiceView	Dirección VoiceView, usada en voz y telefonía de datos.

Tabla.C.1 - Esquemas de dirección soportados por Socket

Esquema de dirección	Usos
ProtocolType.Ggp	Entrada a protocolo de entrada (GGP), usado para comunicaciones de interencaminador.
ProtocolType.Icmp	Protocolo de mensaje de control de Internet (ICMP), también sabido como ping y es usado para los reportes de error en la red.
ProtocolType.Idp	Protocolo de datagrama de Internet (IDP), el transporte subyacente para los protocolos Xerox conectados a una red.
ProtocolType.Igmp	Protocolo de dirección de grupo de Internet (IGMP), usado en multibastidor.
ProtocolType.IP	Protocolo de Internet (IP), el transporte subyacente para todas las comunicaciones sobre la Internet.
ProtocolType.Ipx	Cambio de paquetes de interconexión (IPX), la implementación de Novell de IDP.
ProtocolType.ND	Especifica un protocolo no oficial el disco llamado net (ND).
ProtocolType.Pup	Protocolo de paquete universal PARC (PUP), un precursor de encaminar protocolo de la información (RIP).
ProtocolType.Raw	Datos de raw socket; excluye cabezales de marco.
ProtocolType.Spx	Cambio secuencial de paquetes (SPX), la capa de protocolos de transporte de Novell proveedores deliberadamente los servicios de paquetes.
ProtocolType.SpxII	Cambio de paquete secuencial 2 (SPX2), una implementación moderna de SPX.
ProtocolType.Tcp	Protocolo de control de transmisión (TCP), el protocolo más común para transferencia de datos de Internet.
ProtocolType.Udp	El protocolo de datagrama de usuario (UDP), usado para alta velocidad, los datos de integridad baja se trasladan sobre la Internet.

Tabla.C.2 - Tipos de protocolos soportados por socket.